ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Σύστημα Διαχείρισης για HTML Scraping»

Του φοιτητή
Γκόλτση Παντελεήμων
Αρ. Μητρώου: 144306

Επιβλέπων
Ονοματεπώνυμο Μιχαήλ
Σαλαμπάσης
Βαθμίδα Καθηγητής

Ημερομηνία 25/5/2023

Τίτλος Δ.Ε. Σύστημα Διαχείρισης για HTML Scraping
Κωδικός Δ.Ε. 21112
Ονοματεπώνυμο φοιτητή. Γκόλτσης Παντελεήμων
Ονοματεπώνυμο εισηγητή. Μιχαήλ Σαλαμπάσης
Ημερομηνία ανάληψης Δ.Ε. 09/10/2021
Ημερομηνία περάτωσης Δ.Ε. 25/5/2023

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της *πτυχιακής* εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Η απογοήτευση έρχεται πάντα πριν από το επίτευγμα»

Πρόλογος

Ο σημερινός κόσμος περιτριγυρίζεται από δεδομένα αφού αυτά διακατέχουν εξαιρετικά εκτεταμένη αξία και οδηγούν την πρόοδο. Τα δεδομένα επιτρέπουν στις επιχειρήσεις να λάβουν ορθές αποφάσεις, ενώ σε προσωπικό επίπεδο μας βοηθούν να κατανοήσουμε καλύτερα το αντικείμενο που μας ενδιαφέρει. Στην σημερινή ψηφιακή εποχή η εξάρτηση από δεδομένα συνεχώς αυξάνεται.

Πολλά από τα δεδομένα αυτά βρίσκονται διαθέσιμα για πρόσβαση από τους χρήστες στις ιστοσελίδες του παγκόσμιου ιστού.

Οι υπολογιστές είναι ικανοί να εκτελούν εξαιρετικά γρήγορα αυτόματες και επαναλαμβανόμενες ενέργειες.

Μπορούμε να αντιληφθούμε λοιπόν ότι η δυνατότητα συλλογής των δεδομένων από τον παγκόσμιο ιστό με έναν αυτόματο τρόπο, με την χρήση ενός προγράμματος, κρίνεται ιδιαίτερης σημασία.

Αυτό το παραπάνω μπορεί να το υλοποιήσει ο μηχανισμός του web scraping. Να συλλέξει δηλαδή δεδομένα που βρίσκονται ήδη διαθέσιμα στο διαδίκτυο, δημιουργώντας μεγάλες βάσεις δεδομένων απο δεδομένα που μας ενδιαφέρουν δίνοντας μας την δυνατότητα να εξάγουμε χρήσιμες πληροφορίες. Βέβαια για να γίνει αυτό απαιτείται σχεδόν πάντα ο καθαρισμός, η διόρθωση και η σωστή οργάνωση των εξαγόμενων δεδομένων.

Για τον λόγο αυτό είμαι ιδιαίτερα ικανοποιημένος που η εργασία μου αφορά το web scraping, δίνοντας μου την δυνατότητα να μελετήσω έναν από τους σημαντικούς τομείς του προγραμματισμού, να κατανοήσω τις επιπτώσεις που ενδεχομένως να έχει και εκτος από αυτό: πως να δημιουργηθεί ένα σύστημα που μπορεί να επιτηρεί την όλη διαδικασία του web scraping.

Περίληψη

Η εργασία βασίζεται στην συνέχεια της προηγούμενης πτυχιακής που έκανε χρήση του scraping για την εισαγωγή θεατρικών παραστάσεων και όλων των πληροφοριών που την αφορούν σε μία βάση δεδομένων.

Παρ' όλα αυτά παρουσιάζεται η ανάγκη βελτίωσης κάποιων από τις ατέλειες της προηγούμενης εφαρμογής (όπως κάθε εφαρμογή έχει τέτοιες) καθώς και την υλοποίηση ενός προγράμματος που επιτρέπει σε έναν τρίτο χρήστη να μπορεί να τελεί τον ρόλο τους διαχειριστή του συστήματος, όντας υπεύθυνος για την επιτήρηση και την επίβλεψη του scraping αλλά και της ποιότητας των δεδομένων που υπάρχουν στην βάση. Η εργασία αυτή αποτελεί ένα μέρος του συστήματος των θεατρικών παραστάσεων αφού έχουν υλοποιηθεί και άλλες εργασίες για το scraping, την εμφάνιση και την διαχείριση του backend συστήματος των θεατρικών παραστάσεων.

Με την χρήση αυτού του συστήματος, δίνεται η δυνατότητα στον διαχειριστή να επιβλέπει την διαδικασία του scraping, να διορθώνει και να προσθέτει με εύκολο και απλό τρόπο, τα στοιχεία των παραστάσεων που ενδεχομένως να μην έγιναν ορθά scraping.΄

Επίσης υπάρχει η δυνατότητα παραμετροποίησης του συστήματος ως προς τους κόμβους που γίνονται επιλογή, δεδομένου ότι ο διαχειριστής μπορεί να αλλάξει την ονομασία των κόμβων που αντιστοιχούν στην ιδιότητα του κάθε στοιχείου. Δηλαδή αν η κλάση βάση της οποίας γίνεται επιλογή το όνομα της θεατρικής παράστασης αλλάξει, ο διαχειριστής μπορεί να επέμβει και να δώσει το σωστό πλεον XPath. Αυτό το χαρακτηριστικό διευκολύνει την λειτουργία του συστήματος αφου δεν χρειάζεται επέμβαση από τον προγραμματιστή.

Εκτός από αυτό ο διαχειριστής μπορεί να ορίσει το πότε θα γίνονται scrap οι παραστάσεις και πόσες θα εισάγονται κάθε φορά. Επίσης η ύπαρξη αναλυτικού καταγραφικού δίνει την δυνατότητα ανακάλυψης τυχόν θεμάτων που χρήζουν αντιμετώπιση.

Παρατηρούμε λοιπόν ότι η αποτελεσματική χρήση αυτού του συστήματος scraping επιδρά και σε άλλες εργασίες που έχουν υλοποιηθεί από άλλος φοιτητές και ανήκουν στην ομάδα των πτυχιακών εργασιών που αναπτύσσουν συνεργατικά εφαρμογές για ένα καλλιτεχνικό portal

«Management System for HTML Scraping»

«Panteleimon Gkoltsis»

Abstract

The thesis is based on the continuation of the previous thesis that used scraping to enter theatrical performances and all the information related to them in a database.

Nevertheless, there is a need to improve some of the shortcomings of the previous application (as every application has such) as well as the implementation of a program that allows a third user to be able to play the role of system administrator, being responsible for monitoring and the supervision of the scraping as well as the quality of the data present in the database. This work is a part of the theater system since other tasks have been implemented for the scraping, display and management of the backend system of the theater performances.

By using this system, the administrator is given the possibility to supervise the scraping process and to correct and add in an easy and simple way, the elements of the shows that may not have been properly scraped.'

There is also the possibility of parameterizing the system in terms of the nodes that are selected, given that the administrator can change the name of the nodes that correspond to the property of each element. That is, if the base class from which the play name is selected changes, the administrator can step in and provide the correct XPath selector. This feature facilitates the operation of the system since it does not need any intervention from the programmer.

In addition to this, the administrator can define when the performances will be scrapped and how many will be imported each time. Also, the existence of an analytical log gives the possibility of discovering any issues that need to be addressed

So we notice that the effective use of this scraping system also affects other works that have been implemented by other students and belong to the group of degree works that develop collaborative applications for an artistic portal.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένεια μου, την γιαγιά μου και τον θείο μου, που μου έδωσαν την δυνατότητα να σπουδάσω, κάνοντας με επαρκή να εργαστώ στον τομέα του προγραμματισμού, καθώς και για την συμπαράσταση που μου παρείχαν σε όλες τις στιγμές που ήταν μη ρόδινες. Επίσης οφείλω να εκφράσω τις θερμές μου ευχαριστίες, προς τον επιβλέποντα της εργασίας, κ. Μιχάλη Σαλαμπάση, για την εμπιστοσύνη που μου έδειξε στην ανάθεση της πτυχιακής εργασίας, δίνοντας μου την δυνατότητα να εφαρμόσω τις γνώσεις που απέκτησα από τις σπουδές μου, καθώς επίσης και για την κατανόηση και την καθοδήγηση του

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
CSV	Comma-Separated Values
Captcha	Completely Automated Public Turing test to tell Computers and Humans Apart
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
DOM	Document Object Model
GDPR	General Data Protection Regulation
XML	Extensive Markup Language
XPATH	XML Path Language
XSLT	Extensible Stylesheet Language Transformations
API	Application Programming Interface
AJAX	Asynchronous JavaScript And XML
WYSIWYG	What You See Is What You Get
IDE	Integrated Development Environment
XAML	eXtensible Application Markup Language
WPF	Windows Presentation Foundation

Κεφάλαιο 1ο: Web Scraping

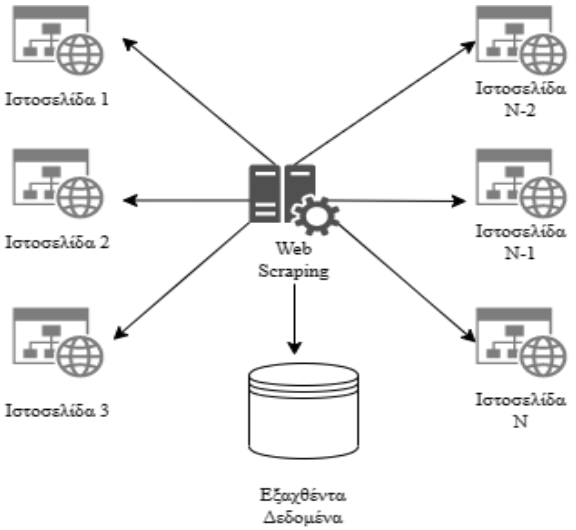
1.1 Εισαγωγή

Στην σημερινή εποχή οι υπολογιστές μπορούν να εκτελούν δισεκατομμύρια εντολές το δευτερόλεπτο, η ύπαρξη μεγάλων βάσεων δεδομένων με ορθή πληροφορία μπορεί να οδηγήσει τις επιχειρήσεις σε σωστές αποφάσεις, καλύτερη στρατηγική και επίγνωση του τομέα, όπου είναι ζωτικής σημασίας για την επιτυχία της επιχείρησης. Σε αυτόν τον τομέα το Web Scraping μπορεί να βοηθήσεις τις διάφορες επιχειρήσεις/οργανισμούς να συλλέξουν από το διαδίκτυο τα δεδομένα που χρειάζονται με σκοπό να τα επεξεργαστούν ώστε να εξάγουν σημαντικές πληροφορίες, προκειμένου να έχουν επίγνωση του περιβάλλοντος που τους απασχολεί.

1.2 Web Scraping

Το Web Scraping είναι η αυτοματοποιημένη διαδικασία ανάκτησης δομημένων δεδομένων από διάφορες ιστοσελίδες. Επιτυγχάνεται με την χρήση λογισμικού όπου εξάγεται το δεδομένο από την ιστοσελίδα και αποθηκεύεται με δομημένο τρόπο σε βάσεις δεδομένων για περαιτέρω ανάλυση ώστε να παρθούν πολύ χρήσιμα συμπεράσματα ή πληροφορίες. Με την χρήση του Web Scraping μπορούν να δημιουργηθούν πολύ μεγάλες βάσεις δεδομένων με πληροφορίες από πολλαπλές ιστοσελίδες. Για παράδειγμα, μια εταιρεία που έχει μια ιστοσελίδα πώλησης προϊόντων, μπορεί να χρησιμοποιήσει το Web Scraping και να εξάγει πληροφορίες από ανταγωνιστικές ιστοσελίδες, έτσι ώστε να γνωρίζει τι τιμές έχει το κάθε προϊόν σε μια συγκεκριμένη χρονική στιγμή. Με τον τρόπο αυτό θα είναι σε θέση να αποφασίζει σε τιμή την συμφέρει να πουλήσει το εκάστοτε προϊόν όπου παίζει βιοτικό ρόλο για την επιτυχία της και την ανάπτυξη της. Επομένως, είναι σε θέση να έχει επαρκή γνώση για τους ανταγωνιστές της.

Η πρακτική της χρήσης του Web Scraping αυξάνεται σημαντικά τα τελευταία έτη λόγω της εκθετικής αύξησης των διαθέσιμων πληροφοριών στο παγκόσμιο ιστό. Μπορεί να πραγματοποιηθεί από ένα πλήθος τεχνικών υλοποίησης Web Scraping που αποτελείται από εργαλεία Scraping, γλώσσες προγραμματισμού όπως Python, C#, Ruby, R και Javascript αλλά και από διάφορα εμπορικά προγράμματα που υπάρχουν διαθέσιμα



Σχήμα 1.1: Web Scraping

1.3 Web Crawling

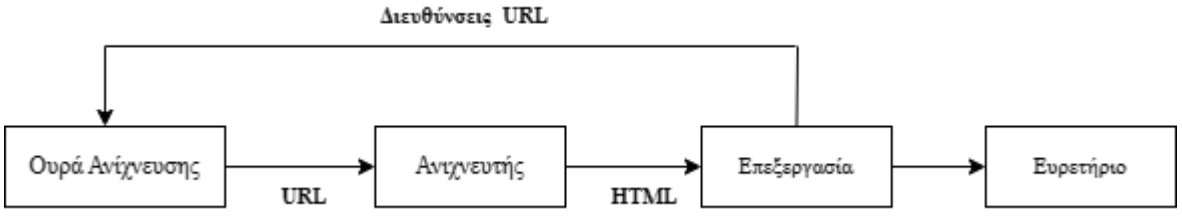
Το Web Crawling είναι η διαδικασία ανακάλυψης και ευρετηρίασης του παγκόσμιου ιστού συλλέγοντας δεδομένα από διάφορες ιστοσελίδες. Ένας web crawler που συχνά αποκαλείται και spider συνήθως χρησιμοποιείται απο τις μηχανές αναζήτησης με σκοπό την ευρετηρίαση του Web. Οι crawlers, επισκέπτονται ιστοσελίδες, ανακαλύπτουν τις επιμέρους ιστοσελίδες που αποτελούν τον ιστότοπο και συλλέγουν πληροφορίες για τις σελίδες που ανακάλυψαν. Οι πληροφορίες αυτές μπορεί να είναι κείμενο, εικόνες, βίντεο, μεταδεδομένα κλπ.

Η ιστορία του Web Crawling ξεκίνησε τον Ιούνιο του 1993 όπου δημιουργήθηκε το πρώτο web robot, το οποίο θεωρείται ως το πρώτο Web Crawler, το World Wide Web Wanderer, σκοπός του οποίου ήταν να ευρετηριάσει τον Παγκόσμιο Ιστό όπου εκείνη την χρονική περίοδο αποτελούνταν από μερικές χιλιάδες ιστοσελίδες. Το wanderer ήταν γραμμένο με την χρήση της γλώσσας προγραμματισμού Perl. Αρχικά μετρούσε μόνο διακομιστές ιστού και έπειτα προστέθηκε η δυνατότητα να ανακτα τα URL των ιστοσελίδων.

Το web crawling συνήθως αρχίζει από την εξερεύνηση μιας λίστας γνωστών ιστοσελίδων και σταδιακά μεγαλώνει τα ευρετήρια του καθώς ανακαλύπτουν περισσότερες υποσελίδες αλλά και νέες σελίδες μέσω των υπερσυνδέσμων που περιέχονται σε κάθε ιστοσελίδα. Τα δεδομένα που συλλέγονται συχνά αποθηκεύονται σε βάσεις δεδομένων.

Προκειμένου να εξασφαλιστεί το αποτελεσματικό Web Crawling, οι crawlers πρέπει να σχεδιαστούν με τέτοιο τρόπο ώστε να αντιμετωπίζουν διάφορες τεχνικές προκλήσεις. Μια κοινή πρόκληση είναι η αντιμετώπιση μεγάλων και πολύπλοκων ιστοσελίδων που μπορεί να έχουν εκατομμύρια σελίδες ή περιεχόμενο που δημιουργείται δυναμικά. Για να αντιμετωπιστεί αυτό το φαινόμενο, οι ανιχνευτές ιστού μπορεί να χρειαστεί να χρησιμοποιήσουν τεχνικές όπως η προσαρμοστική ανίχνευση, η οποία προσαρμόζει δυναμικά τον ρυθμό και το βάθος ανίχνευσης με βάση το μέγεθος και την πολυπλοκότητα του ιστότοπου, προκειμένου να γίνουν crawl περισσότερες ιστοσελίδες διαφορετικού περιεχομένου και προέλευσης.

Καθώς το περιεχόμενο στον παγκόσμιο ιστό όλο και αυξάνεται, τόσο και θα υπάρχει μεγαλύτερη ανάγκη για πιο αποδοτικές τεχνικές web crawling που θα απαιτούν την λιγότερη δυνατή υπολογιστική ισχύ και θα πραγματοποιούνται στην μικρότερη δυνατή χρονική διάρκεια.



Σχήμα 1.2: Web Crawling

1.4 Διαδικασία Web Scraping

Το επιθυμητό τελικό αποτέλεσμα ενός αποτελεσματικού Scraping είναι να εξάγουμε τα ορθά δεδομένα που επιθυμούμε από μια ή περισσότερες ιστοσελίδες και να τα αποθηκεύσουμε στην δική μας βάση δεδομένων.

Αρχικά θα πρέπει να επιλεχθεί η ιστοσελίδα από την από οποία θα κάνουμε parse όλα τα δεδομένα. Αυτό το βήμα είναι πολύ κρίσιμο καθώς η ιστοσελίδα θα πρέπει να έχει όσα περισσότερα δεδομένα χρειαζόμαστε καθώς και να ανανεώνεται συχνά. Με αυτόν τον τρόπο θα είναι δυνατό να να υπάρχουν σωστά δεδομένα.

Έπειτα, ακολουθεί το βήμα της ανάλυσης της ιστοσελίδας, ποιά διεύθυνση έχει ποιά δεδομένα. Αφού βρεθούν αυτές οι διευθύνσεις, θα πρέπει να αναλυθεί το DOM της κάθε σελίδας, Το Document Object Model ορίζει την δομή του HTML εγγράφου και μέσω αυτού η βιβλιοθήκη scraping που θα είναι σε θέση να γνωρίζει ποιόν κόμβο να επιλέξει έτσι ώστε να ανακτήσει το επιθυμητό δεδομένο.

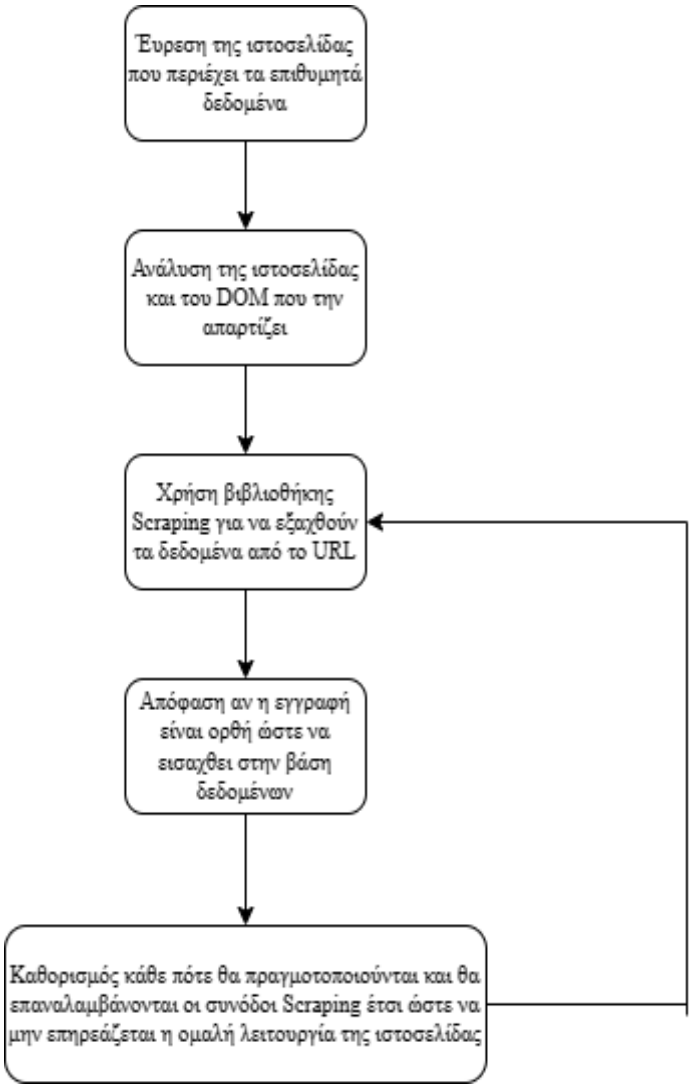
Αφού έχει αναλυθεί η ιστοσελίδα και υπάρχει γνώση από ποιόν κόμβο θα γίνει ανάκτησει της ιστοσελίδας, ακολουθεί η υλοποίηση του Scraping. Αυτό μπορεί να γίνει μέσω επιλογής μιας βιβλιοθήκης που το υλοποιεί όπως BeautifulSoup, Scrapy, ή Selenium. Αυτές οι βιβλιοθήκες βοηθούν να περιηγηθούμε στην ιστοσελίδα και να εξάγουμε τα δεδομένα χωρίς να απαιτείται ή χειροκίνητη παρέμβαση από τον χρήστη.

Με την χρήση της βιβλιοθήκης και αφού έχουμε εξάγει τα επιθυμητά δεδομένα, έρχεται το βήμα της αποθήκευσης των δεδομένων σε μια βάση δεδομένων ή σε ένα αρχείο JSON ή CSV.

Το επόμενο βήμα έχει να κάνει με τον καθαρισμό των δεδομένων από διπλότυπα ή από δεδομένα που δεν πρέπει να υπάρχουν στην βάση καθώς δεν έχουν κάποια εννοιολογική σημασία και θεωρούνται σκουπίδια. Να σημειωθεί ότι αυτό το βήμα μπορεί να γίνει και σε συνδυασμό με το προηγούμενο βήμα καθώς πριν εισαχθεί κάποιο δεδομένο στην βάση, θα πρέπει να ελέγχεται η ορθότητα του και αν ανήκει πραγματικά στο σύνολο των δεδομένων που θέλουμε να κάνουμε Scraping. Είναι ένα απαιτητικό βήμα αφού αν επιλεχθεί να γίνει Scraping σε περισσότερες από μια ιστοσελίδες σε μια κοινή βάση θα εμφανιστούν πολλαπλές διπλότυπες εγγραφές που θα πρέπει να ελεγχθούν και να διαγραφούν.

Τέλος το σύστημα Scraping θα πρέπει να επαναλαμβάνεται σε διαφορετικές συνόδους με απώτερο σκοπό όταν ανανεωθεί με νέο υλικό η ιστοσελίδα το σύστημα Scraping να βρει τα νέα δεδομένα και να τα εισάγει στην βάση δεδομένων. Αποτελεί ένα πολύ σημαντικό βήμα καθώς θα πρέπει να ακολουθούνται κάποια “βήματα ευγένειας” προς την ιστοσελίδα της οποίας γίνεται το Scraping. Για παράδειγμα δεν θα πρέπει να στέλνονται HTTP Requests σε ώρες αιχμής καθώς και πολλαπλά Requests.

Γενικά η υλοποίηση ενός συστήματος Scraping είναι μια απαιτητική διαδικασία που απαιτεί έναν συνδυασμό ικανοτήτων web όπως η ανάλυση του DOM και η χρήση των κατάλληλων selectors με στόχο να εξαχθεί το σωστό δεδομένο, καθώς και πολύ καλή γνώση της επεξεργασίας και του καθαρισμού των δεδομένων που έχουν εξαχθεί. Ο τελικός στόχος είναι να υπάρχουν δεδομένα σωστά, χωρίς να υπάρχουν διπλότυπα ή ασυνεπές πληροφορίες, πράγμα που καθορίζει και την επιτυχία της διαδικασίας του Scraping, ένα σύστημα που είναι σε θέση να προσθέτει νέα δεδομένα και να τα έχει ανανεωμένα.



Σχήμα 1.3: Διαδικασία Scraping

1.5 Θέματα

Όσο δυνατή διαδικασία και είναι το scraping, υπάρχει ένα πλήθος θεμάτων που θα πρέπει να μελετηθούν και να διευθετηθούν πριν ξεκινήσει η διαδικασία του Scraping.

1.5.1 Νομικά

Κάποιος που ξεκινάει το Scraping θα μπορούσε να αναρωτηθεί αν είναι νόμιμο. Η σύντομη απάντηση είναι για τα δημόσια διαθέσιμα δεδομένα είναι ναι, εκτός και αν τα δεδομένα περιέχουν πληροφορίες όπως

- Πνευματικά Δικαιώματα
- Προσωπικές Πληροφορίες

Τα πνευματικά δικαιώματα είναι οποιαδήποτε δεδομένα ή πληροφορίες προστατεύονται από τον νόμο, δίνουν αποκλειστικά δικαιώματα στον δημιουργό για το πώς αυτός θα τα διανέμει αλλά και τον τρόπο με τον οποία θα χρησιμοποιηθούν.

Ο τύπος αυτών των δεδομένων μπορεί να εμπεριέχει εικόνες, βίντεο, τραγούδια, άρθρα αλλα και βιβλία.

Για παράδειγμα, ένα άρθρο από μια ιστοσελίδα που μπορεί να γίνει μόνο προσβάσιμο με συνδρομή, είναι παράνομο να γίνει scrap και να διατεθεί δωρεάν χωρίς την άδεια του αρθρογράφου.

Επομένως περιεχόμενο που απαιτεί πρόσβαση μέσω συνδρομής καλό θα είναι να αποφεύγεται να γίνεται scraping.

Κάνοντας μια αναδρομή στο παρελθόν θα παρατηρήσουμε διάφορες αγωγές εταιρειών, αφού τις περισσότερες φορές, οι εταιρείες δεν θέλουν να γίνονται scrap

eBay εναντίον Bidder's Edge: Ήταν μία από τις πρώτες αγωγές εναντίον ενός web crawler και έγινε το 2000. Το Bidder's Edge ήταν ένας crawler όπου έκανε περίπου 100.000 requests την ημέρα προς την ιστοσελίδα του eBay για να έχει πρόσβαση στις δημοπρασίες και στην συνέχεια δημοσίευσε τις πληροφορίες στην ιστοσελίδα του. Το δικαστήριο αποφάσισε να απαγορεύσει στο Bidder's Edge να κάνει crawl το eBay αφού οι διακομιστές του τελευταίου εξασθενόντουσαν απο τις πολλές HTTP κλήσεις του crawler.

Facebook vs Power Ventures: Η Power Ventures (power.com) ήταν μια ιστοσελίδα κοινωνικής δικτύωσης όπου εμφάνιζε πληροφορίες για χρήστες που είχαν διαμοιραστεί από τους ίδιους σε διάφορα κοινωνικά δίκτυα όπως Facebook, Myspace, Twitter, LinkedIn, Aol κλπ. Το δικαστήριο ενέκρινε την αγωγή του Facebook καθώς αποφάσισε ότι βλάπτεται η εταιρεία και ότι ο κατηγορούμενος είχε παραβιάσει τους όρους παροχής υπηρεσιών του Facebook και τις πολιτικές απορρήτου των χρηστών

Linkedin vs hiQ Labs: Είναι η πιο τωρινή (2019) και ίσως και η πιο σημαντική. Η hiQ Labs είναι μια εταιρεία όπου διαθέτει ανάλυση δεδομένων σε επιχειρήσεις, χρησιμοποιώντας δημόσια διαθέσιμα δεδομένα από την ιστοσελίδα του Linked in. Αυτή χρησιμοποίησε το scraping για να συλλέξει δεδομένα όπως δημόσια προφίλ αλλά και πληροφορίες για του υπαλλήλους τους, που το LinkedIn υποστήριξε ότι είναι παραβίαση των όρων χρήσης του. Η υπόθεση αυτή ήταν πολύ σημαντική διότι καθιέρωσε ότι το web scraping δημόσιων διαθέσιμων δεδομένων μπορεί να είναι νόμιμο και ότι οι ιστοσελίδες δεν μπορούν να χρησιμοποιούν τους όρους χρήσης ως μέθοδο για να απαγορεύσουν την πρόσβαση σε αυτά τα διαθέσιμα δεδομένα. Αυτή η υπόθεση υπέδειξε πόσο σημαντική είναι η προστασία των δεδομένων και την υπευθυνότητα που πρέπει να δείχνουν οι εταιρείες ώστε να προστατεύουν τα δεδομένα των χρηστών τους.

Τα **Προσωπικά στοιχεία ταυτοποίησης** (Personally Identifiable Information) προστατεύονται από τον νόμο GDPR που άρχισε να ισχύει από την 25 Μαΐου του 2018 στην Ευρωπαϊκή Ένωση. Ο νόμος αυτός αντικατέστησε προηγούμενους νόμους και στοχεύει να έχει μεγαλύτερο έλεγχο όσον αναφορά την προστασία των προσωπικών δεδομένων των πολιτών που ανήκουν μέσα στην Ευρωπαϊκή Οικονομική ζώνη. Υπό τον νόμο αυτόν, προσωπικά δεδομένα καθορίζονται οποιεσδήποτε πληροφορίες που μπορούν έμμεσα ή άμεσα να ταυτοποιήσουν κάποιον όπως:

- Ονοματεπώνυμο
- Ηλεκτρονική Διεύθυνση
- Αριθμός τηλεφώνου
- Διεύθυνση IP
- Δεδομένα τοποθεσίας
- Βιομετρικά δεδομένα

Το GDPR έχει σημαντικές επιπτώσεις για το scraping καθώς απαγορεύει την συλλογή και επεξεργασία των προσωπικών δεδομένων χωρίς την ρητή συγκατάθεση των ατόμων που τους αφορά.

1.5.2 Τεχνικά

Εκτός από τα νομικά θέματα, υπάρχουν και τα τεχνικά θέματα που θα πρέπει να αντιμετωπιστούν αν εμφανιστούν κατά την διαδικασία του scraping. Τέτοια θέματα μπορεί να είναι να είναι η αλλαγή της δομής της ιστοσελίδας, η εμφάνιση Captcha, η αποτροπή πρόσβασης στην ιστοσελίδα μέσω αποκλεισμού της IP, το δυναμικό περιεχόμενο καθώς και η μορφοποίηση και ο καθαρισμός των δεδομένων.

Η **δομή της ιστοσελίδας** και η ανάλυση του DOM της αποτελεί ίσως το πιο βασικό βήμα κατά την διαδικασία του scraping, αφού μέσω αυτού μπορεί να γίνει η επιλογή της πληροφορίας που επιθυμούμε να εξάγουμε. Ας υποθέσουμε ότι θέλουμε να κάνουμε scraping για πληροφορίες βιβλίων μέσω ενός ιστότοπου και ότι ο τίτλος ενός βιβλίου βρίσκεται μέσα στο tag

```
<h1 class="booktitle"> The art of war </h1>
```

Ένας τρόπος ώστε να έχουμε πρόσβαση στον εκάστοτε τίτλο του βιβλίου είναι να χρησιμοποιήσουμε έναν css selector h1.booktitle και να πάρουμε τον τίτλο του βιβλίου. Με αυτόν τον τρόπο επισκέπτοντας την διεύθυνση του καθενός βιβλίου στον ιστότοπο θα μπορούμε μέσω του παραπάνω css selector να έχουμε όλους τους τίτλους των βιβλίων.

Παρόλα αυτά, ο διαχειριστής του ιστότοπου θα μπορούσε πολύ εύκολα να αποτρέψει προσωρινά τον τρόπο ανάκλησης του τίτλου του βιβλίου αλλάζοντας την δομή του παραπάνω tag. Για παράδειγμα, αν το tag γινόταν:

```
<h1 class="titleofthebook"> The art of war </h1>
```

ο τωρινός css selector δεν θα έφερνε τον τίτλο των βιβλίων με αποτέλεσμα να μην είναι δυνατή η διαδικασία του scraping. Θα έπρεπε δηλαδή ο προγραμματιστής να εντοπίσει το σφάλμα που προέκυψε και να να αλλάξει τον css selector χειροκίνητα ώστε να συνεχιστεί η ορθή διαδικασία του scraping.

Με αυτόν τον τρόπο οι διαχειριστές των ιστοσελίδων μπορούν να έχουν έναν προσωρινό τρόπο ώστε να αποτρέψουν το web scraping για μια συγκεκριμένη χρονική περίοδο μέχρι ο προγραμματιστής να εντοπίσει την αλλαγή και να προσαρμόσει ξανά το scraping σύμφωνα με την απαιτούμενη αλλαγή.

Η **εμφάνιση των Captcha** μπορεί να προκαλέσει ανησυχία σε πολλούς προγραμματιστές αφού είναι μια κοινή τακτική που χρησιμοποιείται για την αποφυγή του scraping, αφού αυτά είναι δημιουργημένα έτσι ώστε να αντιλαμβάνονται αν ο χρήστης είναι άνθρωπος ή κάποιο είδος αυτοματοποιημένου εργαλείου.

Σχήμα 1.4: Παράδειγμα ενός Captcha

Αυτά μπορούν να κατηγοριοποιηθούν στις εξής κατηγορίες: Αναγνώρισης χαρακτήρα, αναγνώρισης ήχου, μαθηματικών προβλημάτων, αναγνώρισης εικόνας αλλά και σύνδεση μέσω κοινωνικής δικτύωσης.

Έτσι μπορεί να αναγνωριστεί εάν το σύστημα αλληλεπιδρά με άνθρωπο ή με κάποιο άλλο σύστημα.

Κάποιοι από τους τρόπους αντιμετώπισης των Captcha είναι η χρήση των Headless browsers, υπηρεσίες λύσης captcha, αλλά και αλλαγή των τεχνικών που πραγματοποιείται το scraping.

Οι Headless Browsers είναι οι φυλλομετρητές που δεν χρησιμοποιούν γραφική διεπιφάνεια. Χρησιμοποιούνται κυρίως από software test μηχανικούς, καθώς αυτού οι φυλλομετρητές μπορούν να εκτελέσουν τις διαδικασίες που καλούνται να εκτελέσουν ταχύτερα από τους παραδοσιακούς, δεδομένου ότι δεν καταναλώνουν λιγότερους πόρους λόγω της έλλειψης της γραφικής διεπιφάνειας. Ένα μεγάλο τους πλεονέκτημα είναι ότι μπορούν να τρέξουν σε διακομιστές που και αυτοί δεν διαθέτουν υποστήριξη γραφικής διεπιφάνειας

Η αποφυγή της επίλυσης του Captcha μπορεί επίσης να γίνει μέσω της ανάθεσης της επίλυσης σε μια **εξωτερική υπηρεσία** η οποία παρέχει υπηρεσίες επίλυσης του Captcha. Με την ενσωμάτωση αυτού του service στον αλγόριθμο του scraping μπορεί να ξεπεραστεί το παραπάνω πρόβλημα.

Μερικές από αυτές τις υπηρεσίες επιτυγχάνουν την λειτουργίας τους με την ανάθεση της επίλυσης του Captcha σε έναν άνθρωπο ο οποίος το λύνει και στην συνέχεια επιστρέφει την λύση. Επομένως η χρήση τέτοιων υπηρεσιών μπορεί να είναι ένας αποδοτικός τρόπος για την αντιμετώπιση του προβλήματος, ειδικότερα για πιο σύνθετα Captcha, παρόλα αυτά υπάρχουν αρκετά σημαντικά μειονεκτήματα για την χρήση αυτών των υπηρεσιών.

Το κόστος τέτοιων υπηρεσιών μπορεί να είναι αρκετά μεγάλο, ειδικότερα για μεγάλης κλίμακας πρότζεκτ. Η κοστολόγηση τους συνήθως γίνεται ανά μήνα ή ανά πλήθος των λυμένων Captcha. Επίσης η ακρίβεια και η αξιοπιστία αυτών των υπηρεσιών μπορεί να διαφέρει από υπηρεσία σε υπηρεσία. Κάποιες υπηρεσίες μπορεί να προσφέρουν λύση μόνο για προβλήματα αναγνώρισης εικόνα, ενώ αλλα σε αναγνώριση ήχου, με αποτέλεσμα το αποτέλεσμα της λύσης να είναι έγκυρο ή άκυρο ανάλογα με το τι τύπου Captcha εμφανιστεί. Τέλος, ίσως και το πιο σημαντικό μειονέκτημα είναι ότι οι υπηρεσίες αυτές προκειμένου να τα λύσουν χρησιμοποιούν συγκεκριμένα χαρακτηριστικά που μπορούν να χαρακτηριστούν ως ρίσκα ασφάλειας, όπως η αποστολή του URL που γίνεται scrap.

Τέλος, ένας τρόπος αποφυγής των Captcha είναι η επιλογή ιστοσελίδων που θα γίνουν scrap που δεν τα έχουν ενσωματώσει στις ιστοσελίδες τους.

1.5.3 Ποιοτικά

Ένα θέμα που θα εμφανιστεί σε κάθε περίπτωση scraping είναι αυτό των ποιοτικών χαρακτηριστικών των δεδομένων. Το τελικό επιθυμητό αποτέλεσμα είναι τα δεδομένα τα οποία εισάγονται στην βάση να είναι υψηλής ποιότητας. Επομένως, θα πρέπει να διασφαλίζεται το χαρακτηριστικό αυτο το δεδομένων. Κάποια από τα θέματα ποιότητας δεδομένων που μπορούν να εμφανιστούν είναι τα παρακάτω:

Ένα από τα θέματα που μπορεί να εμφανιστεί είναι τα δεδομένα να έχουν **ελλιπή δεδομένα.** Για παράδειγμα, επιθυμούμε να εξάγουμε τους τίτλους από όλα τα βιβλία που υπάρχουν σε μια ιστοσελίδα. Τα βιβλία συνήθως θα εμφανίζονται ανά συγκεκριμένο Ν αριθμό και με χρήση pagination ή lazy loading εμφανίζονται τα επόμενα Ν βιβλία. Η υλοποίηση του scraping θα μπορούσε να μην διαχειρίζεται σωστά το pagination που έχει υλοποιηθεί με αποτέλεσμα να μην εισάγονται όλοι οι

τίτλοι των βιβλίων στην βάση μας, γεγονός που κατατάσει το σύνολο των δεδομένων της βάσης να είναι ελλιπές. Παρατηρούμε λοιπόν, ότι θα πρέπει να εφαρμοστούν οι σωστές τεχνικές scraping για την ορθή προσπέλαση όλων των στοιχείων.

Τα **ανακριβή δεδομένα** είναι μια ακόμη μορφή ποιοτικών χαρακτηριστικών που υπάρχει συχνα. Οι ιστοσελίδες δέχονται συχνά αλλαγή στην δομή τους με αποτέλεσμα να μην προσπελάσονται τα στοιχεία σωστά. Μια αλλαγή στο DOM της σελίδας HTML μπορεί να αποφέρει αυτό το αποτέλεσμα. Εκτός απο αυτό, η αλλαγή στην μορφή ενός URL μπορεί να αποφέρει το ίδιο αποτέλεσμα. Για παράδειγμα το URI /book_title=the%20art%20of%war και μέσω αυτού παίρναμε τα χαρακτηριστικά του βιβλίου με τον συγκεκριμένο τίτλο. Αν αυτό το URI αλλάξει και γίνει /title_of_the_book=the%20art%20of%war τότε η παραπάνω διεύθυνση δεν θα μπορέι να προσπελαστεί με αποτέλεσμα να υπάρχει πρόβλημα στο scraping

Τα **διπλότυπα δεδομένα** αποτελούν ένα μεγάλο θέμα συζήτησης. Ας πάρουμε το προηγούμενο παράδειγμα, πόσα βιβλία the art of war πρέπει να έχουμε στην βάση δεδομένων μας; Πώς θα διακρίνουμε ότι ο οίκος εκδόσεων βιβλίων Arcturus ήδη υπάρχει και δεν πρέπει να ξανα εισαχθεί στην βάση; Αυτό το φαινόμενο είναι πολύ συχνό και θα πρέπει να αντιμετωπίζεται από τον προγραμματιστή έχοντας ρυθμίσεις σωστά τον scraper του για να διακρίνει τις διπλότυπες εγγραφές που πιθανόν θα υπάρξουν.

Ένα ακόμα θέμα που εμφανίζεται είναι τα **ξεπερασμένα/ανεπίκαιρα δεδομένα.** Τα δεδομένα μπορούν να θεωρηθούν ξεπερασμένα αν η ιστοσελίδα δεν τα εμφανίζει ανά τακτά χρονικά διαστήματα ή αν ο scraper δεν εκτελει την διαδικασία του scraping τις χρονικές στιγμές που πρέπει να ανανεωθούν τα δεδομένα. Ας φέρουμε ως παράδειγμα το scraping ενός ιστότοπου με τις τιμές μια μετοχής. Αν η τιμή της μετοχής αναφέρεται στην προηγούμενη μέρα διότι η ιστοσελίδα δεν έχει ανανεωθεί τότε αυτή η τιμή μπορεί να θεωρηθεί ξεπερασμένη διότι η εφαρμογή μας απαιτεί την τιμή της μετοχής της τωρινής μέρας.

Παρατηρούμε λοιπόν ότι ο υπεύθυνος του scraping θα πρέπει να διευθετήσει αρκετά θέματα όσον αφορά την ποιότητα των δεδομένων, αφού και η ποιότητα των δεδομένων και η μεγάλη τους ποσότητα είναι ο τελικός στόχος της διαδικασίας του scraping.

1.5.4 Συντήρησης

Το web scraping μπορεί να απαιτήσει συνεχή συντήρηση και αναβαθμίσεις για να είναι σε θέση να συμβαδίζει με τις αλλαγές που μπορεί να γίνονται στην ιστοσελίδα ή και να διορθωθούν ήδη υπάρχοντα λάθη του κώδικα. Αυτή η συντήρηση μπορεί να διαρκέσει αρκετό χρόνο και να απαιτεί ιδιαίτερη τεχνική εμπειρία από τον προγραμματιστή.

Οι ιστοσελίδες από τις οποίες εξάγονται τα δεδομένα μέσω scraping συχνά κάνουν ανανεώσεις και στην δομή του HTML περιεχομένου, για τον λόγο αυτό θα πρέπει να ο προγραμματιστής να είναι σε ετοιμότητα έτσι ώστε να επέμβει στο κώδικα και να κάνει τις απαραίτητες αλλαγές που απαιτούνται ώστε να συνεχίζεται ομαλά η ροή του scraping.

Η παρακολούθηση της απόδοσης του scraping είναι ακόμα ένας κρίσιμος παράγοντας. Θα πρέπει να ελέγχεται συχνά ότι το πρόγραμμα δεν επιβραδύνει τον διακομιστή web στο οποίο φιλοξενείται η ιστοσελίδα. Εκτός από αυτό, η χρήση καταγραφικού ως προς τις ενέργειες που κάνει το πρόγραμμα είναι ένας πολύ σημαντικός παράγοντας. Με την χρήση του λοιπόν, θα μπορεί να υπάρχει μια σαφής εικόνα για το αν παρουσιάζονται σφάλματα στον κώδικα και σε ποιό σημείο, παράγοντας που θα επηρεάσει σημαντικά την ταχύτητα και την ευκολία στην προσπάθεια της αποσφαλμάτωσης του

κώδικα που σκοπό έχει να λυθούν τα προβλήματα/θέματα που ενδεχομένως θα εμφανιστούν στον κώδικα. Επίσης, ένας κώδικας που έχει πολλά σχόλια και καλή τεκμηρίωση θα βοηθήσει και μελλοντικούς νέους προγραμματιστές που ίσως αναλάβουν την συντήρηση του προγράμματος που υλοποιεί το scraping.

1.6 Τρόποι αποτροπής Web Scraping

Ο ιδιοκτήτης ενός ιστότοπου σε κάποιες περιπτώσεις θα θελήσει να αποτρέψει ή να περιορίσει τις εφαρμογές scraping από το να έχουν πρόσβαση στην ιστοσελίδα του. Οι λόγοι αυτοί μπορεί να είναι διότι επηρεάζεται η ταχύτητα του ιστότοπου λόγω των πολλαπλών αιτήσεων που πραγματοποιούν τα bots, με αποτέλεσμα οι πραγματικοί χρήστες να νιώθουν την έλλειψη της ταχύτητας στον ιστότοπο και να σταματούν την περιήγηση στην ιστοσελίδα. Οι ιδιοκτήτες επίσης μπορεί να θέλουν να αποτρέψουν τα προγράμματα διότι δεν επιθυμούν την μη εξουσιοδοτημένη χρήση, διανομή και ανάλυσης του περιεχομένου τους. Αρκετοί crawlers έχουν αναπτυχθεί με σκοπό την ανάλυση των τιμών από πολλαπλές ιστοσελίδες ηλεκτρονικού εμπορίου. Με την ανάλυση των τιμών αυτών οι εταιρείες που πραγματοποιούν το scraping και έχουν στην κατοχή τους αυτά τα δεδομένα, μπορούν να εξάγουν συμπεράσματα για το πως επηρεάζονται οι τιμές αλλά και ποιά τιμή είναι η πιο συμφέρουσα για αυτούς. Έτσι λοιπόν, παρατηρούν τις κινήσεις και τις αποφάσεις των ανταγωνιστών τους. Για τους παραπάνω λόγους λοιπόν, οι ιδιοκτήτες των ιστοσελίδων ενσωματώνουν κάποια χαρακτηριστικά στους διακομιστές τους ως μέτρα για την αποφυγή του web scraping.

Μπορεί να γίνει εν μέρη ταυτοποίηση των scrapers μέσω του **user-agent** που βρίσκεται στην κεφαλίδα αίτησης του HTTP. Η τιμή της συμβολοσειράς περιέχει στοιχεία που αναφέρουν τον φυλλομετρητή και το λειτουργικό σύστημα του αποστολέα Η τιμή του user-agent έχει μορφή όπως παρακάτω:

```
user-agent: Mozilla/5.0 (<system-information>) <platform> (<platform-details>) <extensions>
```

Δεδομένου ότι αρκετοί scrapers έχουν προκαθορισμένη την τιμή του user-agent σε κάθε αίτηση προς τον διακομιστή της ιστοσελίδας, οι διακομιστές είναι σε θέση να γνωρίζουν ότι η HTML αίτηση προέρχεται από πρόγραμμα scraping

Ένας ακόμη τρόπος αποφυγής του scraping είναι μέσω της διεύθυνσης **IP**. Αυτή η διεύθυνση είναι ένα μοναδικό αναγνωστικό και αποδίδεται σε κάθε συσκευή που συνδέεται στο διαδίκτυο. Επειδή ο διακομιστής γνωρίζει την διεύθυνση των υπολογιστών που του στελνουν αιτήσεις μπορεί να αποκλείσει αιτήσεις που προέρχονται από μια συγκεκριμένη διεύθυνση IP. Εκτός του αποκλεισμού μιας συγκεκριμένης διεύθυνσης όμως, μια ακόμα τακτική που χρησιμοποιείται είναι ο περιορισμός αιτήσεων για μια διεύθυνση IP, όπου η πρόσβαση επιτρέπεται αλλά για μόνο ένα αριθμό αιτήσεων ανα χρονική περίοδο.

Επίσης οι scrapers μπορούν να διακριθούν εύκολα από τους πραγματικούς χρήστες από την συμπεριφορά τους κατά την αλληλεπίδραση που έχουν με τον διακομιστή. Ένα τέτοιο πρόγραμμα, κάνει συγκεκριμένες και πολλαπλές αιτήσεις με πολύ μικρή καθυστέρηση από αίτηση σε αίτηση. Έτσι όταν γίνεται scraping ενός ιστότοπου με βιβλία, συνήθως θα υπάρχουν πολλές αιτήσεις για βιβλία μιας συγκεκριμένης κατηγορίας και αφού τελειώσουν τα βιβλία της κατηγορίας αυτής, θα ξεκινήσει η διαδικασία του scrap για βιβλία μιας άλλης κατηγορίας, Επομένως αν οι διακομιστές αναλύσουν την συμπεριφορά των αιτήσεων που δέχονται, θα μπορούν να έχουν μια καλή ιδέα για το αν αυτές προέρχονται από αυτοματοποιημένο εργαλείο scraping και να κάνουν χρήση του αποκλεισμού της συγκεκριμένης διεύθυνσης IP.

Τέλος, ένα βήμα που μπορεί να κάνει πιο δύσκολη την διαδικασία του web scraping και έχει ήδη αναλυθεί είναι η χρήση των **Captcha**, αφού θα εμφανίσει περαιτέρω εμπόδια στην όλη διαδικασία και ο προγραμματιστής που αναλάβει να αναπτύξει το πρόγραμμα scraping ίσως δεν το συμπεριλάβει στην λίστα των ιστοσελίδων που θα κάνει scrap.

1.7 Επίλογος

Μπορούμε λοιπόν να καταλήξουμε ότι η συγκομιδή δεδομένων (web scraping) εμπεριέχει τις μεθόδους με τις οποίες γίνεται δυνατή η εξαγωγή των δεδομένων από μια ιστοσελίδα με σκοπό την αποθήκευση τους σε μια βάση δεδομένων η σε αρχεία μορφής CSV, XML κλπ. Είναι ένα πολύ σημαντικό εργαλείο για τις επιχειρήσεις, τους ερευνητές αλλά και για άτομα που θέλουν να έχουν δεδομένα για έναν τομέα που τους απασχολεί.

Μια έρευνα του περιοδικού Forbes έδειξε ότι το το 2018 2,500,000,000,000,000 bytes πληροφοριών παράγονται καθημερινά. Στον σημερινό κόσμο που είναι εν μέρη οδηγούμενος από δεδομένα η δυνατότητα να εξάγονται δεδομένα από τις ιστοσελίδες καθιστά το web scraping ένα πολύ σημαντικό εργαλείο που μπορεί να χρησιμοποιείται από όλες τις επιχειρήσεις ώστε να έχουν δεδομένα που η ανάλυση τους θα τους φέρει σε πλεονεκτική θέση για τις αποφάσεις και την στρατηγική που θα ακολουθήσουν. Παρατηρούμε δηλαδή ότι, η ποιότητα των δεδομένων που θα εξαχθούν είναι άκρως σημαντική

Παρόλα αυτά τα προγράμματα scraping θα πρέπει να είναι προγραμματισμένα έτσι ώστε να σέβονται τις ιστοσελίδες από τις οποίες κάνουν συγκομιδή των δεδομένων μην υπερφορτώνοντας τους διακομιστές της ιστοσελίδας, να σέβονται τα προσωπικά δεδομένα των χρηστών και να μην εξάγουν περιεχόμενο που ενδεχομένως να προστατεύεται από το νόμο.

Κεφάλαιο 2ο: Μέθοδοι Υλοποίησης Scraping

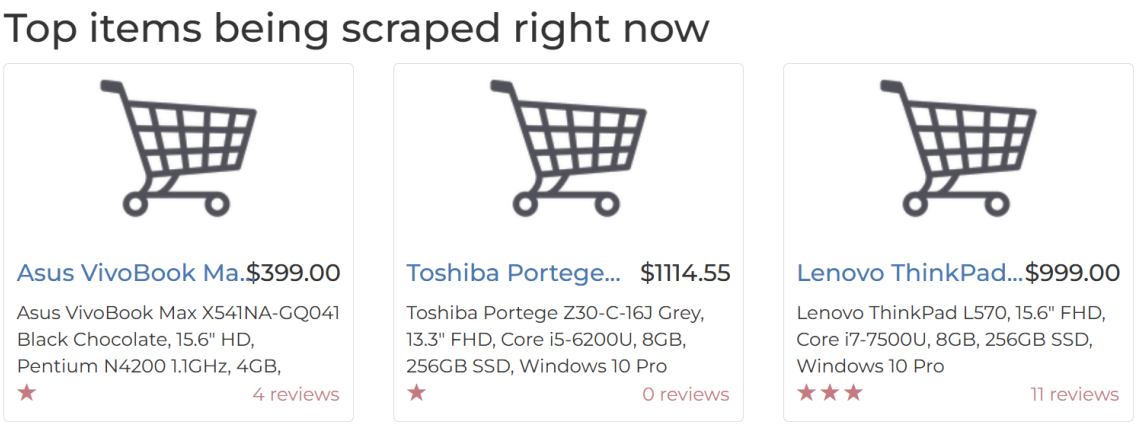
2.1 Εισαγωγή

Η συγκομιδή των δεδομένων μπορεί να πραγματοποιηθεί είτε μέσω την χρήση συγκεκριμένων βιβλιοθηκών scraping που χρησιμοποιούνται σε διάφορες γλώσσες προγραμματισμού είτε μέσω αυτοποιημένων εργαλείων που επιτρέπουν στους χρήστες να αντλήσουν τα δεδομένα που τους ενδιαφέρουν από ιστοσελίδες, χωρίς να έχουν καθόλου γνώσεις προγραμματισμού, αλλά χρησιμοποιώντας την γραφική διεπιφάνεια που τους παρέχει το εκάστοτε εργαλείο. Να σημειωθεί ότι με την χρήση γλωσσών προγραμματισμού δεν υπάρχουν περιορισμοί μέσα στις δυνατότητες που παρέχει το εργαλείο και ότι ο προγραμματιστής έχει περισσότερη ευελιξία στην ανάπτυξη του τελικού εργαλείου scraping

2.2 Scraping μέσω διαφόρων γλωσσών προγραμματισμού

Καθώς οι δυνατότητες του web scraping ανακαλύπτονται από όλο και περισσότερες εταιρείες, οργανισμούς αλλά και άτομα, τόσο η χρήση του και αυξάνεται. Για τον λόγο αυτό αναπτύσσονται όλο και περισσότερες βιβλιοθήκες που προσφέρουν έτοιμες λειτουργίες για τους προγραμματιστές που θα τους βοηθάνε στην πιο γρήγορη ανάπτυξη του scraping. Τέτοιες λειτουργίες μπορεί να είναι να ανακτήσουν τον τίτλο της ιστοσελίδας, να εξάγουν όλους τους συνδέσμους αλλά και η πιο σημαντική λειτουργία που είναι η επιλογή ενός συγκεκριμένο κόμβου απο το HTML DOM με σκοπό την άντληση του κείμενο/πληροφορίας που τον απαρτίζει.

Για την επίδειξη του πως υλοποιείται το Scraping στις διάφορες γλώσσες που θα αναφέρουμε, θα εκτελέσουμε το ίδιο παράδειγμα με χρήση των C#, Python, Ruby και R. Θα ανακτήσουμε απο την παρακάτω ιστοσελίδα τα ονόματα, την τιμή και το πλήθος των αξιολογήσεων των προϊόντων



Σχήμα 2.1: Παράδειγμα ιστοσελίδας που θα κάνουμε Scrap
https://webscraper.io/test-sites/e-commerce/allinone

2.2.1 C#

Η C# είναι μια σύγχρονη, αντικειμενοστρεφής γλώσσα προγραμματισμού που κυκλοφόρησε πρώτη φορά το 2001 και είναι μέρος του .NET framework και προσαρμόστηκε εξ αρχής ώστε να το υποστηρίζει. Αναπτύχθηκε για είναι μια απλή αλλά συνάμα και ισχυρή γλώσσα προγραμματισμού. Δανειζόμενη αρκετά χαρακτηριστικά από την Java και παρόμοια σύνταξη με την Java καθιστά την εκμάθηση της εύκολη. Αρκετά χαρακτηριστικά που διαθέτει βοηθούν τους προγραμματιστές να αναπτύξουν ισχυρές και ανθεκτικές εφαρμογές σε σφάλματα. Η αυτόματη διαχείριση μνήμης μέσω του Garbage Collection που παρέχεται, βοηθάει στην ελευθέρωση μνήμης που δεν χρησιμοποιείται από το πρόγραμμα και βοηθάει τον προγραμματιστή από προβλήματα υπερβολικής χρήσης μνήμης που μπορει να καταστήσει το πρόγραμμα αργό αλλά και τον υπολογιστή στον οποίο τρέχει αφού θα καταλαμβάνει μεγάλο ποσοστό των διαθέσιμων πόρων. Επίσης ένα ακόμα σημαντικό χαρακτηριστικό είναι η χρήση των nullable types όπου δίνεται η δυνατότητα για την αποφυγή σφαλμάτων όταν εμφανίζεται μια μεταβλητή με τιμή null, είναι δηλαδή ένας τύπος δεδομένων που μπορεί να έχει μια τιμή ή να είναι null. Οι τύποι αυτοί αναπαραριστούνται με την χρήση του συμβόλου ? μετά τον τύπο του δεδομένου (π.χ double?)

Μια από τις πιο γνωστές βιβλιοθήκες για την υλοποίηση του web scraping μέσω της C# είναι το Html Agility Pack (HAP) που παρέχει στους προγραμματιστές ευκολίες ως προς την ανάλυση του HTML και το διάβασμα/εγγραφή του DOM με εγγενής υποστήριξη του XPATH ή XSLT (μπορεί να υποστηρίξει και CSS Selector με την χρήση τρίτων βιβλιοθηκών).

Με την χρήση του HTML Agility Pack οι προγραμματιστές είναι σε θέση να:

- Φορτώσουν ένα HTML έγγραφο από ένα αρχείο ή ένα URL.
- Επιλέξουν τους κόμβους από το DOM που επιθυμούν με την χρήση XPATH, XSLT ή LINQ.
- Να διασχίσουν τους κόμβους.
- Τροποποιήσουν ή να πάρουν τις τιμές από τον κόμβο που έγινε επιλογή.

Ένα από τα πλεονεκτήματα του HAP είναι ότι έχει ενσωματωμένο parser που σημαίνει ότι ο προγραμματιστής δεν χρειάζεται να χρησιμοποιήσει άλλη βιβλιοθήκη για να ανακτήσει την ιστοσελίδα

Τύπος	Περιγραφή
Από αρχείο	Φόρτωση ενός HTML εγγράφου που είναι αποθηκευμένο τοπικά σε ένα αρχείο
Από συμβολοσειρά	Φόρτωση ενός HTML εγγράφου που είναι αποθηκευμένο σε μια μεταβλητή τύπου String
Από URL	Φόρτωση ενός HTML εγγράφου με την χρήση του URL στο οποίο βρίσκεται η σελίδα
Από φυλλομετρητή	Φόρτωση ενός HTML εγγράφου με την ενός φυλλομετρήτη

Πίνακας 2.1: Τρόποι ανάκτησης ιστοσελίδας

Όπως παρατηρούμε υπάρχει μεγάλη ευελιξία για τον προγραμματιστή για το πως θα γίνει ανάκτηση του HTML εγγράφου για την περαιτέρω ανάλυση του. Αξίζει να σημειωθεί ότι καλό θα είναι όταν γίνονται οι δοκιμές του scraping να αποθηκεύεται η ιστοσελίδα τοπικά και να φορτώνεται ως αρχείο αφού θα περιορίσει τις περιττές αιτήσεις προς τον διακομιστή όπου φιλοξενείται η ιστοσελίδα.

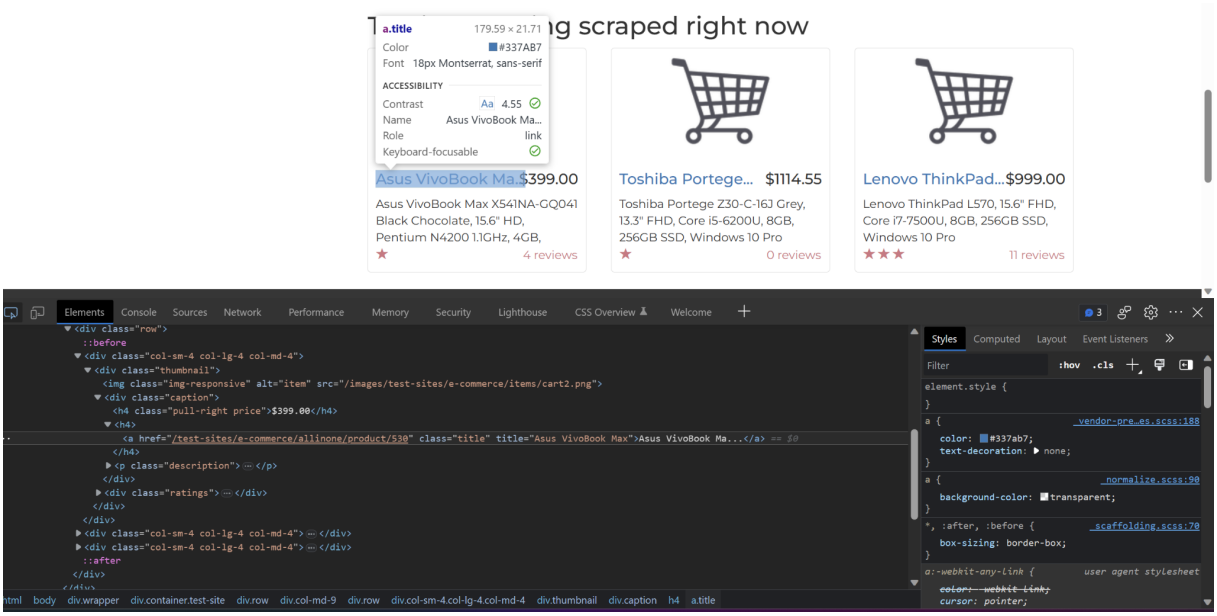
```
using System;
using System.Linq;
using HtmlAgilityPack;

class Program
{
   static void Main(string[] args)
   {
      //Ανάκτηση της HTML ιστοσελίδες και αποθήκευση της στην doc μεταβλητη
      var url = "https://webscraper.io/test-sites/e-commerce/allinone";
      var web = new HtmlWeb();
      var doc = web.Load(url);

      //Ανακτούμε τα ονόματα, την τιμή και τις αξιολογήσεις των προϊόντων με χρήση του XPath
      var productNodes = doc.DocumentNode.SelectNodes("//div[@class='thumbnail']");
      var products = productNodes.Select(node => new
      {
         Name = node.SelectSingleNode(".//a[@class='title']")?.InnerText.Trim(),
         Price = node.SelectSingleNode(".//h4[@class='pull-right price']")?.InnerText.Trim(),
         Reviews = node.SelectSingleNode(".//p[@class='pull-right']")?.InnerText.Trim(),
      }).ToList();

      // Εμφάνιση των αποτελεσμάτων
      Console.WriteLine("Χαρακτηριστικά προϊόντων:");
      foreach (var product in products)
      {
         Console.WriteLine($"Όνομα: {product.Name}");
         Console.WriteLine($"Τιμή: {product.Price}");
         Console.WriteLine($"Αξιολογήσεις: {product.Reviews}");
         Console.WriteLine();
      }
   }
}
```

Σχήμα 2.2: Παράδειγμα Scraping με χρήση C#

Στο παραπάνω παράδειγμα κώδικα παρουσιάζεται η ανάκτηση των πληροφοριών για το όνομα , την τιμή και το πλήθος των αξιολογήσεων από μια ιστοσελίδα με χρήση C#.

Αρχικά γίνεται η φόρτωση της ιστοσελίδας με την χρήση του URL ως παράμετρο στο αντικείμενο HtmlWeb. Επόμενο στάδιο είναι η επιλογή του κατάλληλου HTML κόμβου με σκοπό την ανάκτηση των στοιχείων που αναζητούμε. Παρατηρώντας την δομή του ανακαλύπτεται ότι όλα τα στοιχεία του προϊόντος βρίσκονται σε ένα html div που το όνομα της κλάσης του είναι “thumbnail”. Με την μέθοδο SelectNodes και ως παράμετρο της το κατάλληλο xpath που περιέχει την σωστή κλάση θα επιστραφούν όλοι οι αντίστοιχοι κόμβοι από την ιστοσελίδα. Έτσι, ο προγραμματιστής είναι σε θέση να προσπελάσει τους κόμβους έναν προς έναν με αποτέλεσμα να πάει από κάτι πιο γενικό (πατέρας κόμβος DOM που περιέχει όλα τα στοιχεία) σε κάτι πιο ειδικό (παιδί κόμβος που περιέχει την τιμή).

Από αυτό το σημείο ο προγραμματιστής μπορεί να επικεντρωθεί στις λεπτομέρειες που επιθυμεί να ανακτήσει, με την χρήση της συνάρτησης InnerText() είναι σε θέση να πάρει το κείμενο του κόμβου με αποτέλεσμα να έχει το όνομα του προϊόντος. Να σημειωθεί ότι ένας τρόπος που διευκολύνει την διαδικασία του scraping είναι η χρήση του inspector του εκάστοτε φυλλομετρητή όπου πηγαίνοντας το ποντίκι στον αντίστοιχο κόμβο μας εμφανίζεται το όνομα της κλάσης του DOM αλλά και την δομή του και διάφορα άλλα χαρακτηριστικα που βοηθούν στην καλύτερη αντίληψη του πως δομείται το DOM.



Σχήμα 2.3: Χρήση του inspector του φυλλομετρητή

Παρατηρούμε ότι υπάρχουν περιπτώσεις όπου αφού επιλέξουμε έναν κόμβο μπορούμε να χρησιμοποιήσουμε τις συναρτήσεις της βιβλιοθήκης scraping έτσι ώστε να περιορίσουμε την περιοχή αναζήτησης και να αναζητήσουμε μόνο κόμβους παιδιά του συγκεκριμένου στοιχείου.

Με την εντολή node.SelectSingleNode(".//a[@class='title']")?.InnerText.Trim() επιλέγουμε έναν κόμβο <a> με κλάση title που είναι παιδί του node που επιλέχθηκε νωρίτερα. Με το InnerText παίρνουμε το κείμενο του κόμβου και με την εντολή Trim αφαιρούμε περιττά κενά διαστήματα που ενδεχομένως να υπάρχουν στον κόμβο. Επομένως αποτελεί καλή πρακτική οι εντολες InnerText.Trim() να συνδυάζονται ώστε να μην υπάρχουν περιττά κενά αλλα και άλλες ιδιότητες html που μπορεί να περιέχονται στο περιεχόμενο του κόμβου.

Τέλος μπορούμε να καταλήξουμε ότι η επιλογή της C# μαζί με την βιβλιοθήκη Html Agility Pack αποτελεί μια πολύ καλή επιλογή για κάποιον που θέλει να αναλάβει την υλοποίηση του Scraping. Η C# έχει αρκετά πλεονεκτήματα διότι είναι φιλική και ως περιβάλλον ανάπτυξης το Microsoft Visual Studio μπορεί να βοηθήσει τον προγραμματιστή κατά την αποσφαλμάτωση του προγράμματος του. Αξίζει να αναφερθεί ότι ένα αρνητικό της HAP είναι ότι δεν έχει εγγενής υποστήριξη για επιλογή κόμβου με CSS Selector, αν και στην ιστοσελίδα τους αναφέρεται ότι θα υλοποιηθεί και ενσωματωθεί σε επόμενες εκδόσεις της.

2.2.2 Python

Η Python (προέρχεται από την αρχαιοελληνική λέξη πύθων) είναι μια ανοιχτού κώδικαγλώσσα προγραμματισμού υψηλού επιπέδου, γενικού σκοπού και διερμηνευόμενη (interpreted). Δημιουργήθηκε από τον Gudio van Rossum και έγινε διαθέσιμη για πρώτη φορά στις 20 Φεβρουαρίου του 1991 παίρνοντας το όνομα της από την τηλεοπτική σειρά του BBC Monty Python's Flying Circus. Είναι γνωστή για την ευκολία διαβάσματος της, την απλότητα της και την ευκολία χρήσης της που την καταστεί μια από τις πιο πολυχρησιμοποιημένες γλώσσες προγραμματισμού σήμερα.

Κάποιοι από τους λόγους για να μάθει κάποιος την Python είναι

- Ευκολία εκμάθησης: Η python είναι μια από τις πιο εύκολες γλώσσες προγραμματισμού λόγω της απλότητας της σύνταξης της και των πλούσιων συναρτήσεων που περιέχονται στις βιβλιοθήκες της. Αυτά τα δύο κύρια χαρακτηριστικά την καθιστούν μια γλώσσα ιδανική για αρχάριους προγραμματιστές ώστε να την μάθουν και να αναπτύξουν προγράμματα με αυτή σε διάφορα είδη πεδίων.
- Ύπαρξη μεγάλης κοινότητας: Το γεγονός ότι η python είναι ευρέως χρησιμοποιούμενη την καθιστά να έχει πολλά αποτελέσματα στο διαδίκτυο για τις ερωτήσεις που θα εμφανιστούν στον προγραμματιστή καθώς αναπτύσει τα προγράμματα, που κάνει την διαδικασία εύρεσης απαντήσεων και πληροφοριών πιο γρήγορη από μια γλώσσα που δεν είναι τόσο διάσημη.
- Ανοιχτού κώδικα: Η Python είναι μια γλώσσα προγραμματισμού ανοιχτού κώδικα που σημαίνει ότι είναι δωρεάν για χρήση και δεν επιβαρύνει τον προγραμματιστή με επιπλέον κόστη για αγορά άδειας χρήσης της ή ανανέωσης της όπως συμβαίνει με άλλες γλώσσες προγραμματισμού.
- Ευελιξία: Η Python είναι πολύ ευέλικτη αφου χρησιμοποιείται σε πολλούς τομείς όπως σε ανάπτυξη εφαρμογών web, ανάλυσης δεδομένων, τεχνητής νοημοσύνης κλπ. Έτσι ένας προγραμματιστής αφού την γνωρίσει, βρίσκεται σε θέση να μελετήσει και άλλους τομείς πάνω σε μια γλώσσα που ήδη έχει μάθει.
- Ευκαιρίες καριέρας: Το γεγονός ότι η Python είναι μια από τις πιο πολυχρησιμούποιημενες γλώσσες προγραμματισμού την κάνει μια απο τις πιο σημαντικές γλώσσες για να βρει κάποιος δουλειά.

Δύο από τα πιο γνωστά πακέτα Python για την υλοποίηση web scraping με την χρήση python είναι οι BeautifulSoup και Scrapy.

Το Scrapy είναι ένα framework για την εξαγωγή δεδομένων από τις ιστοσελίδες με έναν γρήγορο, απλό αλλά και επεκτάσιμο τρόπο και χρησιμοποιείται κυρίως ως web crawler. Όπως αναφέρθηκε σχεδιάστηκε ώστε να είναι επεκτάσιμο καθώς υποστηρίζοντας ασύγχρονη επεξεργασία με χρήση του Twisted μπορεί να στείλει πολλαπλές αιτήσεις την ίδια χρονική στιγμή, κάνοντας το ιδανικό για την εξαγωγή πολλών δεδομένων. Υποστηρίζει ένα μεγάλο εύρος φορμάτ δεδομένων όπως HTML, XML αλλά και JSON. Αν και το Scrapy αρχικά σχεδιάστηκε για web scraping μπορεί να εξάγει δεδομένα και με την χρήση των API. Έχει ενσωματωμένη υποστήριξη για παραμετροποίηση των proxies και των user-agents, δίνοντας την δυνατότητα στον προγραμματιστή να βρει τρόπους ώστε να μην αποκλειστεί το πρόγραμμα του απο την πρόσβαση στην ιστοσελίδα.

Η BeautifulSoup είναι μια βιβλιοθήκη για την εξαγωγή δεδομένων από αρχεία HTML και XML. Μπορεί να χρησιμοποιηθεί με αρκετούς parsers όπως τους lxml, html5lib, and html.parser. Διαθέτει συναρτήσεις για την μετακίνηση μεταξύ των κόμβων αλλά και τρόπους επιλογής των κόμβων με βάση τον τύπο του, το id αλλά και την κλάση. Επίσης μπορεί να αναζητήσει κόμβους με βάση τις ιδιότητες τους, το περιεχόμενο του κειμένου τους αλλά και την θέση τους. Υποστηρίζει την τροποποίηση του

περιεχομένου καθώς μπορεί να προσθέσει, να διαγράψει ή να τροποποιήσει τα στοιχεία που το απαρτίζουν.

```
import requests
from bs4 import BeautifulSoup

# Ανάκτηση της HTML ιστοσελίδας και αποθήκευση της στην response μεταβλητη
url = "https://webscraper.io/test-sites/e-commerce/allinone"
response = requests.get(url)

# Κάνουμε parse το περιεχόμενο με χρήση του parser html.parser
soup = BeautifulSoup(response.content, "html.parser")

# Παίρνουμε τα χαρακτηριστικά των προϊόντων με χρήση των  CSS selectors
product_containers = soup.select("div.thumbnail")
products = []
for container in product_containers:
    product = {
        "Name": container.select_one("a.title").get_text().strip(),
        "Price": container.select_one("h4.pull-right.price").get_text().strip(),
        "Reviews": container.select_one("p.pull-right").get_text().strip(),
    }
    products.append(product)

# Εμφάνιση των αποτελεσμάτων
print("Χαρακτηριστικα προιοντων::")
for product in products:
    print(f"Ονομα: {product['Name']}")
    print(f"Τιμη: {product['Price']}")
    print(f"Αξιολογήσεις: {product['Reviews']}")
```

Σχήμα 2.4: Παράδειγμα Scraping με χρήση Python και Beautiful Soup

Στο παραπάνω παράδειγμα κώδικα παρουσιάζεται η ανάκτηση των πληροφοριών για το όνομα , την τιμή και το πλήθος των αξιολογήσεων από μια ιστοσελίδα με χρήση της Python.

Αρχικά γίνεται ανάκτηση της HTML ιστοσελίδας με χρήση της βιβλιοθήκης requests, μια αναφορά με την HAP της C# είναι ότι η Beautifulsoup δεν διαθέτει ενσωματωμένη λειτουργία για ανάκτηση της ιστοσελίδας. Στην συνέχεια γίνεται parse το περιεχόμενο της ιστοσελίδας στο αντικείμενο soup με χρήση του parser html.parser (θα μπορούσε να είναι οποιοσδήποτε parser από τους lxml, html5lib, and html.parser). Έπειτα επιλέγουμε τα divs που περιέχουν τα προϊόντα με την χρήση του .select. Εδώ παρατηρούμε ένα πλεονέκτημα της BeautifulSoup σε σχέση με την C# που είναι ότι υποστηρίζει CSS Selectors ενώ με το HAP χρησιμοποιήθηκε το XPath. Κάνοντας την χρήση βρόχου επεξεργαζόμαστε όλα τα στοιχεία και επιλέγουμε τις ειδικότερες λεπτομέρειες όπως όνομα, τιμή και πλήθος αξιολογήσεων και τα προσθέτουμε στην λίστα products. Οι εντολές get_text().strip() είναι οι αντίστοιχες .InnerText.Trim() της C# που εξηγήθηκαν νωρίτερα. Τέλος εμφανίζουμε τα στοιχεία που εξάγαμε από την ιστοσελίδα.

Παρατηρούμε ότι το κύριο πλεονέκτημα της BeautifulSoup είναι ότι διαθέτει υποστήριξη για επιλογή των κόμβων με χρήση των CSS Selectors σε αντίθεση με το HAP που υποστηρίζει XPath που είναι λίγο πιο δύσκολο στην χρήση σε σχέση με την απλότητα που διαθέτουν τα CSS Selectors. Μια ακόμη διαφορά είναι ότι για την ανάκτηση του HTML περιεχομένου στην BeautifulSoup χρησιμοποιήσαμε εξωτερική βιβλιοθήκη ενώ στην HAP είναι ενσωματωμένη η λειτουργία.

2.2.3 Ruby

Η Ruby είναι μια ανοιχτού κώδικα γλώσσα υψηλού επιπέδου, διερμηνευόμενη και αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε από τον Ιάπωνα Yukihiro Matsumoto και είχε την πρώτη της διαθέσιμη έκδοση στο κοινό το 1996 αλλά η αναγνώριση της και η δημοτικότητα της αυξήθηκε από το 2006. Σύμφωνα με τον Yukihiro Matsumoto ήθελα να φτίαξει την Ruby ωστε “να είναι πιο ισχυρή απο την Perl και πιο αντικειμενοστρεφής απο την Python. Είναι γνωστή για την απλότητα της και την ευκολία χρήση της και αποτελείται από μια μεγάλη κοινότητα προγραμματιστών που συνεισφέρουν στην περαιτέρω ανάπτυξή της. Η πιο συνήθης χρήση της είναι η ανάπτυξη web εφαρμογών με την χρήση του framework Ruby on Rails αλλά επειδή είναι μια γλώσσα γενικού σκοπού μπορεί να χρησιμοποιηθεί σε πολλούς τομείς.

Κάποια από τα κύρια χαρακτηριστικά της Ruby είναι:

- Αντικειμενοστρεφής: Σύμφωνα με την ιστοσελίδα της Ruby όλα είναι ένα αντικείμενο και κάθε είδος στοιχείου και κώδικα μπορεί να έχει τις δικές τις ιδιότητες.
- Ευελιξία: Θεωρείται ευέλικτη γλώσσα αφού επιτρέπει στους χρήστες της να τροποποιήσουν τον κώδικα της. Σε μια προσπάθεια να μην περιορίζει τον προγραμματιστή, του επιτρέπει να αλλάξει, αφαιρέσει κύρια χαρακτηριστικά της.
- Χρήση των blocks: Ένα block αποτελείται από γραμμές κώδικα που είναι ανάμεσα σε {}, έχει όνομα και εκτελείται μέσα από μια συνάρτηση που έχει το ίδιο όνομα με τον block με την χρήση της εντολής yield.
- Αυτόματη διαχείριση μνήμης: Η Ruby έχει ενσωματωμένο Garbage collector που αναλαμβάνει την διαχείριση της μνήμης και δεν χρειάζεται ο προγραμματιστής να ασχολείται με την ελευθέρωση της μνήμης.
- Υποστήριξη μεταπρογραμματισμού: Η Ruby υποστηρίζει μεταπρογραμματισμού που σημαίνει ότι μπορεί να γραφτεί κώδικας που αυτός με την σειρά του θα μπορεί να γράψει κώδικα κατα την εκτέλεση του.

Μια απο τις πιο δημοφιλές βιβλιοθήκες για την χρήση Scraping με Ruby είναι η Nokogiri. Βασίζεται στην λογική ότι κάθε έγγραφο δεν είναι έμπιστο με αποτέλεσμα να είναι πιο ασφαλές. Παρέχει λειτουργίες για ανάγνωση, προσθήκη, επεξεργασία, διαγραφή αλλά και εύρεσης κόμβων HTML ή XML. Υποστηρίζει επιλογές κόμβων με CSS Selectors αλλά και XPath. Χρησιμοποιεί τον parser libxml2 που είναι σε θέση να αναλύει XML έγγραφα γρήγορα. Έχοντας μεθόδους για την εξαγωγή του κειμένου, χαρακτηριστικών και άλλων ιδιοτήτων των κόμβων παρέχει την ευκολία στον προγραμματιστή να αφοσιωθεί στην ανάλυση της δομής του εγγράφου και να μην ανησυχεί πως θα πάρει το οποιοδήποτε χαρακτηριστικό/ιδιότητα του κόμβου. Επίσης υπάρχει η δυνατότητα της διαγραφής των υπάρχων κόμβων αλλά και τροποποίησης τους προσθέτοντας νέες ιδιότητες όπως κλάσεις, id αλλά και αλλαγή του κειμένου από το οποίο απαρτίζεται.

```ruby
require 'nokogiri'
require 'open-uri'

# Ανάκτηση της HTML ιστοσελίδας
url = "https://webscraper.io/test-sites/e-commerce/allinone"
doc = Nokogiri::HTML(URI.open(url))

# Παίρνουμε τα χαρακτηριστικά των προϊόντων με χρήση των  CSS selectors
product_containers = doc.css("div.thumbnail")
products = []
product_containers.each do |container|
  product = {
    "Name" => container.css("a.title").text.strip,
    "Price" => container.css("h4.pull-right.price").text.strip,
    "Reviews" => container.css("p.pull-right").text.strip,
  }
  products << product
end

#Εμφάνιση των αποτελεσμάτων
puts "Χαρακτηριστικά προιοντων::"
products.each do |product|
  puts "Ονομα: #{product['Name']}"
  puts "Τιμη: #{product['Price']}"
  puts "Αξιολογησεις: #{product['Reviews']}"
  puts
end
```

Σχήμα 2.5: Παράδειγμα Scraping με χρήση Ruby και Nokogiri

Στο παραπάνω σχήμα παρουσιάζεται το γνωστό παράδειγμα με χρήση Ruby και της βιβλιοθήκης Nokogiri. Παρατηρούμε ότι η βιβλιοθήκη δεν υποστηρίζει την ανάκτηση της ιστοσελίδας όπως η HAP που έχει ως αποτέλεσμα την αναγκαστική χρήση της βιβλιοθήκης open-uri για την ανάκτηση του HTML περιεχομένου του παραπάνω συνδέσμου με χρήση της εντολής URI.open(). Αυτό το περιεχόμενο περνάει ως παράμετρο για δημιουργία του doc αντικειμένου της βιβλιοθήκης που με βάση αυτό θα μπορούμε να επιλέξουμε τους κόμβους που χρειαζόμαστε. Με την χρήση της εντολής .css() περνάμε ως παράμετρο τον CSS Selector για την επιλογή του επιθυμητού κόμβου. Αν κάποιος θέλει να να κάνει επιλογή κόμβου με χρήση του αντίστοιχου XPath αυτό είναι δυνατό με χρήση της εντολής .xpath().

Ακολουθεί η ίδια λογική για την ανάκτηση των χαρακτηριστικών του ονόματος, της τιμή και του πλήθους των αξιολογήσεων που χρησιμοποιήθηκε και στις προηγούμενες γλώσσες. Ένα χαρακτηριστικό της Nokogiri είναι ότι η εντολή .css μπορεί να φέρει έναν ή περισσότερους κόμβους, σε αντίθεση με τις άλλες βιβλιοθήκες που πρέπει να χρησιμοποιηθεί η αντίστοιχη εντολή ανάλογα αν θέλουμε έναν ή περισσότερους κόμβους.

2.2.4 R

Η R είναι μια γλώσσα προγραμματισμού και περιβάλλον που χρησιμοποιείται κυρίως για στατιστικές αναλύσεις, δημιουργία γραφημάτων. Είναι ανοιχτού κώδικα και παρέχεται δωρεάν. Έγινε διαθέσιμη για πρώτη φορά 18 Απριλίου του 1995 απο τους Ross Ihaka και Robert Gentleman. Η Γλώσσα R έχει πάνω από 10000 πακέτα αποθηκευμένα στο CRAN και ο αριθμός αυτών των πακέτων όλο και αυξάνεται. Μπορεί να τρέξει σε οποιοδήποτε λειτουργικό σύστημα, οπότε οι προγραμματιστές δεν χρειάζεται να ανησυχούν σε τι μηχάνημα θα τρέξει το πρόγραμμα που φτιάξανε στην R. Μπορεί να υποστηρίξει το διάβασμα JSON, CSV, XML, Excel αρχείων αλλά και να συνδεθεί σε βάσεις δεδομένων. Αποτελεί την πιο χρησιμοποιημένη γλώσσα προγραμματισμού για στατιστικές αναλύσεις και χρησιμοποιείται σε πολλές επιχειρηματικές εφαρμογές. Η γνωση της R μπορεί να δώσει στον προγραμματιστή πολλές ευκαιρίες καριέρας για την εύρεση εργασίας για δουλειές που σχετίζονται με Ανάλυση Δεδομένων.

Το γεγονός ότι η R είναι κατάλληλη γλώσσα προγραμματισμου για στατιστικές αναλύσεις την κάνει έναν παράγοντα για να την επιλέξει κάποιος ώστε να κάνει το scraping με χρήση της R. Δύο από τα πιο δημοφιλές πακέτα για web scraping είναι το rvest και RSelenium.

```
library(rvest)

# Ανάκτηση της ιστοσελίδας
url <- "https://webscraper.io/test-sites/e-commerce/allinone"
page <- read_html(url)

# Ανάκτηση των στοιχείων με χρήση CSS Selectors
product_containers <- html_nodes(page, "div.thumbnail")
products <- list()
for (container in product_containers) {
  product <- list(
    Name = html_text(html_nodes(container, "a.title")),
    Price = html_text(html_nodes(container, "h4.pull-right.price")),
    Reviews = html_text(html_nodes(container, "p.pull-right"))
  )
  products <- c(products, product)
}

# Εμφάνιση των αποτελεσμάτων
cat("Στοιχεία προϊόντων:\n")
for (product in products) {
  cat(product, "\n")
}
```

Σχήμα 2.6: Παράδειγμα Scraping με χρήση R και rvest

Το rvest επιτρέπει στον προγραμματιστή να ανακτήσει την ιστοσελίδα με χρήση του URL ως παράμετρο και στην συνέχεια να επιλέξει τον κατάλληλο κόμβο που χρειάζεται με χρήση των CSS Selectors. Με την χρήση της εντολης read_html() μπορούμε να ανακτήσουμε το περιεχόμενο της HTML ιστοσελίδας. Αξίζει να αναφερθεί ότι με την χρήση της παραμέτρου encoding μπορούμε να αναφέρουμε την κωδικοποίηση του εγγράφου. Το rvest μάλιστα υποστηρίζει και την επιλογή

συγκεκριμένου user-agent - της κεφαλίδας της αίτησης HTML που θα σταλεί στον διακομιστή που φιλοξενεί στην ιστοσελίδα - μέσω των εντολών httr::set_config() και httr::user_agent(). Έτσι, δίνεται η δυνατότητα στον προγραμματιστή να αποτρέψει ενδεχόμενο αποκλεισμό από την πρόσβαση στην ιστοσελίδα που μπορεί να σχετίζεται με την τιμη του user-agent.

Στο παραπάνω σχήμα παρατηρούμε το rvest εμπεριέχει μέθοδο για την ανάκτηση του περιεχομένου της ιστοσελίδας, εν αντιθέση με την βιβλιοθήκη BeautifulSoup της Python. Επίσης υποστηρίζει και την επιλογή των κόμβων με βάση τους CSS Selector σε αντίθεση με το HAP της C#.

2.3 No Code - Low Code Scraping

Ο Προγραμματισμός χωρίς κώδικα αναφέρεται στην διαδικασία της δημιουργίας εφαρμογών χωρίς την ανάγκη της σύνταξης γραμμών κώδικα από τον χρήστη. Η διαδικασία αυτή συμπεριλαμβάνει γραφικές επιφάνειες, εργαλεία drag and drop που βοηθούν/οδηγούν τον χρήστη στην ανάπτυξη του τελικού προγράμματος που επιθυμεί. Συνήθως αυτές οι πλατφόρμες έχουν στοιχεία που μπορούν να "ενωθούν" για να φτιαχτούν τα λεγόμενα building blocks που μπορούν να είναι διαγράμματα, πίνακες, αλλά και φόρμες με στοιχεία εισαγωγης. Αυτές οι φόρμες μπορούν να συνδέονται με κάποια βάση δεδομένων που θα εισάγει ο χρήστης τα στοιχεία της και θα αποθηκεύονται τα στοιχεία εκει. Επίσης τα στοιχεία θα μπορούν να αποθηκεύονται σε αρχεία με μορφή CSV και να είναι διαθέσιμα για λήψη από τον χρήστη. Η λογική του προγραμματισμού χωρίς κώδικα είναι δηλαδή να παρέχουν στον χρήστη εύχρηστες διεπιφάνειες που θα μπορεί να καθορίσει την λογική και την ροή του προγράμματος με έναν τρόπο που θα είναι εύκολα κατανοητός από αυτόν χωρίς να έχει γνώσεις προγραμματισμού

Για την ανάγκη της δημιουργίας μιας HTML ιστοσελίδας κάποιος που δεν έχει γνώσεις προγραμματισμού , θα μπορούσε να σκεφτεί ότι θα πρέπει να την αναθέσει σε έναν προγραμματιστή που διαθέτει γνώσεις ανάπτυξης εφαρμογών σε Web. Παρόλα αυτά, με χρήση προγραμμάτων no code αυτό δεν είναι αναγκαίο και δίνει την δυνατότητα σε απλούς χρήστες να δημιουργήσουν την ιστοσελίδα που αυτοί επιθυμούν. Το 1995 κυκλοφόρησε το GeoCites όπου με την χρήση ενός WYSIWYG εργαλείου έδινε την δυνατότητα για πρώτη φορά στους χρήστες να κάνουν drag-and-drop στοιχεία με σκοπό τον σχεδιασμό της διάταξης της ιστοσελίδας.

Έπειτα ακολούθησε το Dreamweaver όπου είχε έναν WYSIWYG editor για αυτούς που επιθυμούσαν να προγραμματίσουν χωρίς την χρήση κώδικα ενώ παράλληλα υπήρχε και ο παραγόμενος κώδικας που μπορούσαν να αλλάξουν οι χρήστες που είχαν γνώσεις HTML και ήθελαν να κάνουν κάτι πιο προχωρημένο. Το blogger που αγοράστηκε από την Google το 2003 επέτρεπε στους χρήστες να φτιάξουν το δικό τους blogspot και να συντάξουν το περιεχόμενο που επιθυμούν.

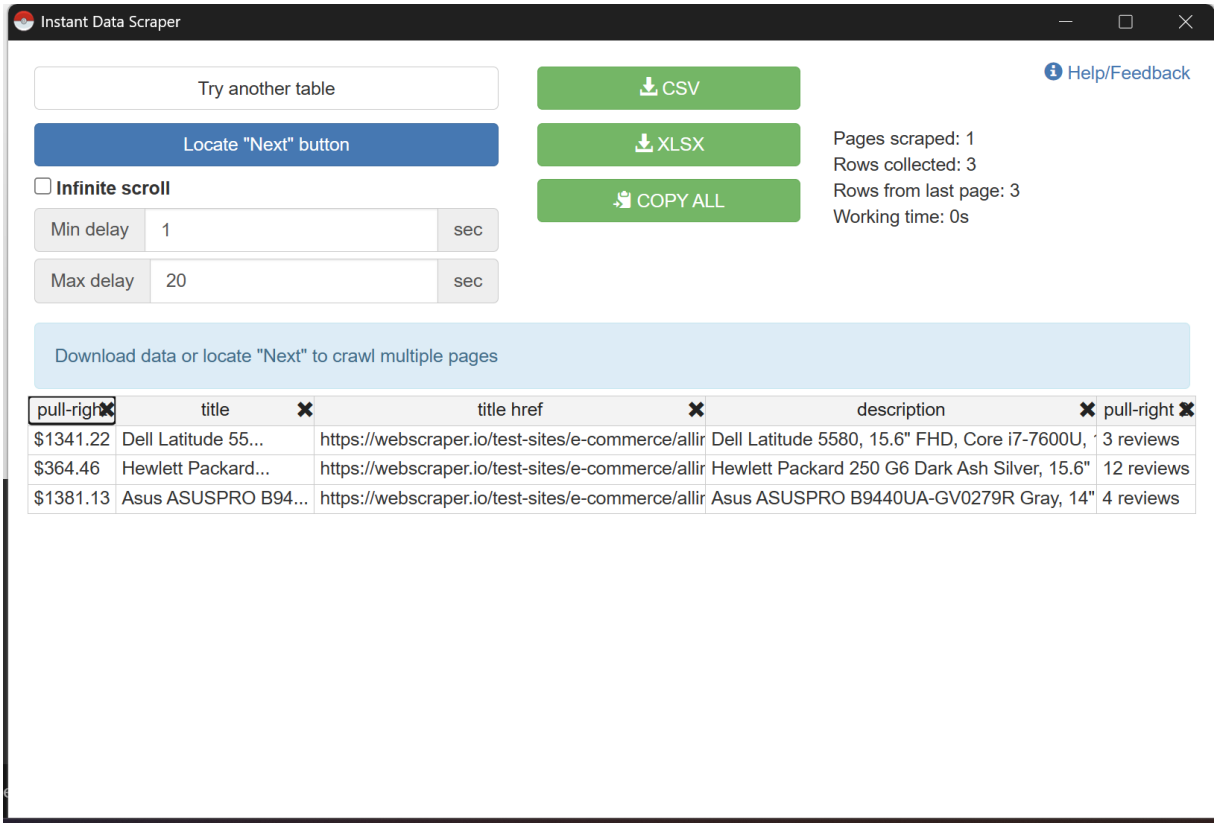
Σήμερα τα πιο γνωστά εργαλεία για την δημιουργία ιστοσελίδων χωρίς ή με λίγο κώδικα είναι τα Wix και Squarespace όπου με την βοήθεια των διαφημιστικών καμπανιών που κάνουν έχουν γίνει γνωστά σε πολλούς χρήστες ανα τον κόσμο.

Όσον αφορά το web scraping, υπάρχει μια πληθώρα εργαλείων που βοηθούν τους χρήστες να εκτελέσουν την διαδικασία του web scraping δίχως να γράψουν ουδεμία γραμμή κώδικα. Αυτό η διαδικασία συνήθως πραγματοποιείται με τρία στάδια. Αρχικά ο χρήστης εισάγει το URL που επιθυμεί να κάνει εξαγωγή των δεδομένων, έπειτα με την βοήθεια γραφικής επιφάνειας κάνει κλικ στις περιοχές που επιθυμεί να πάρει τα στοιχεία και στο τέλος βλέπει τα αποτελέσματα κάνοντας τις ανάλογες αλλαγές ανάλογα με το τι θα του επιστρέψει ως αποτέλεσμα η διεπιφάνεια.

Ένα από τα πιο δημοφιλές εργαλεία no code scraping είναι το Octoparse. Βοηθάει τους χρήστες να εξάγουν τα δεδομένα από την ιστοσελίδα που επιθυμούν απλώς κάνοντας κλικ στην περιοχή που

βρίσκονται τα δεδομένα μέσα από την διεπιφάνεια, Υπάρχει υποστήριξη για εξαγωγή δεδομένων από ιστοσελίδες που έχουν infinite scrolling, σελίδες που απαιτούν σύνδεση, η με σελίδες που φορτώνουν δυναμικά τα δεδομένα τους με χρήση AJAX. Επίσης υποστηρίζει χρονοπρογραμματιστές όπου ο χρήστης έχει την δυνατότητα να τρέχει το scraping όποτε επιθυμεί, δηλαδή ποιά μέρα βδομάδα, ποιά μέρα και ποιά ώρα. Εκτός από αυτό υπάρχει η δυνατότητα για αλλαγή των IP αυτόματα ώστε να μην αποκλείεται ο scraper από την πρόσβαση στην ιστοσελίδα. Το Octoparse διαθέτει δωρεάν δοκιμή και η τιμολόγηση του ξεκινά από τα 90 δολάρια τον μήνα.

Ένα δωρεάν addon για web scraping απο το Chrome web store είναι το Instant Data Scraper όπου εξάγει δεδομένα από τους πίνακές της ιστοσελίδας σε μορφή CSV ή XLSX. Χρησιμοποιεί τεχνητή νοημοσύνη για να προβλέψει ποια δεδομένα είναι αυτά που θα επιθυμεί να εξάγει ο χρήστης, αν τα δεδομένα δεν είναι αυτά που επιθυμεί ο χρήσης, τότε υπάρχει το κουμπί Try another table όπου κάνει την επόμενη πρόβλεψη για το που είναι τα δεδομένα που επιθυμεί να αποθηκεύσει ο χρήστης. Διαθέτικα χαρακτηριστικά όπου μπορεί να καταλάβει πότε ολοκληρώνεται η φόρτωση των δυναμικών δεδομένων, να φορτώσει την επόμενη σελίδα για περιεχόμενο που εμφανίζεται με σελιδοποίηση (pagination), υποστήριξη για infinite scrolling αλλά και μετονομασία των στηλών αφού η προκαθορισμένη ονομασία που αποδίδει είναι το όνομα της κλάσης.



Σχήμα 2.7: Παράδειγμα Scraping με χρήση no code addon Instant Data Scraper

2.4 Επίλογος

Παρατηρούμε ότι το Scraping μπορεί να υλοποιηθεί με διάφορες γλώσσες προγραμματισμού, βιβλιοθήκες αλλά και με εφαρμογές που βοηθούν τους χρήστες να υλοποιήσουν την διαδικασία του Scraping δίχως την γνώση προγραμματισμού

Ένα από τα μειονεκτήματα της υλοποίησης scraping με χρήση της βιβλιοθήκης HTML Agility Pack είναι η μη υποστήριξη -προς ώρας- της επιλογής κόμβων με χρήση CSS Selectors αλλά απαιτείται η χρήση του XPath που είναι λίγο πιο δύσκολο από το πρώτο. Παρόλα αυτά το HAP υποστηρίζει την ανάκτηση του περιεχομένου της HTML ιστοσελίδας με δική της συνάρτηση και δεν χρειάζεται εξωτερική βιβλιοθήκη ώστε να υλοποιήσει την λειτουργία της ανάκτησης του περιεχομένου του HTML.

Ένα σημαντικό χαρακτηριστικό των βιβλιοθήκων BeautifulSoup της Python, nokogiri της Ruby και rvest της R είναι ότι υποστηρίζουν την χρήση των CSS Selector κάνοντας την διαδικασία επιλογής κόμβων λίγο πιο εύκολη. Από αυτές τις τρεις βιβλιοθήκες μόνο η R δεν χρειάζεται εξωτερική βιβλιοθήκη για ανάκτηση της HTML ιστοσελίδας.

Απο την άλλη πλευρά είναι και τα εργαλεία χωρίς κώδικα που βοηθούν τους χρήστες να εξάγουν δεδομένα από την ιστοσελίδα που επιθυμουν και να τα αποθηκεύσουν τοπικά σε μορφή αρχείων. Αυτά τα εργαλεία μπορούν να διευκολύνουν την διαδικασία του scraping και να την κάνουν πιο προσιτή σε χρήστες που δεν έχουν γνώσεις προγραμματισμού ή για κάποιους που θα ήθελαν να έχουν τα δεδομένα ιστοσελίδων γρήγορα και χωρίς να πληρώσουν για το κόστος υλοποίησης ενός προγράμματος. Το αρνητικό με αυτά τα εργαλεία είναι ότι μπορούν να περιορίσουν τους χρήστες ανάλογα με το τι χαρακτηριστικά προσφέρουν. Όμως συνίστανται για απλές δομές και για χρήση από απλούς χρήστες έτσι ώστε να γνωρίσουν το Scraping.

Απο τις γλώσσες προγραμματισμού, θα μπορούσαμε να πούμε ότι σχετικά όλες οι βιβλιοθήκες έχουν τις βασικές ικανότητες επιλογής κόμβων, διάσχισης των κόμβων και ανάκτηση των χαρακτηριστικών που επιθυμεί ο χρήστης. Θα μπορούσαμε να πούμε ότι η επιλογή της γλώσσας προγραμματισμού που θα υλοποιηθεί το scraping είναι προσωπική επιλογή ως προς το ποιά γλώσσα προτιμάει ο προγραμματιστής. Όμως μια απλή αναζήτηση στο LinkedIn για εύρεση εργασίας που αφορά το scraping θα μας δείξει ότι η πλειοψηφία των δουλειών αφορά scraping με την χρήση του Python. Για τον λόγο αυτό θα προτεινόταν η Python για να εξοικειωθεί κάποιος με τον κόσμο του Scraping.

Κεφάλαιο 3ο: Microsoft Visual Studio

3.1 Εισαγωγή

Το Microsoft Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης κώδικα (IDE) που έχει δημιουργηθεί και ανήκει στην Microsoft. Μέσα από το περιβάλλον αυτό δίνει την δυνατότητα στου χρήστες να αναπτύξουν εφαρμογές για υπολογιστές, κινητά, δημιουργία διαδικτυακών εφαρμογών αλλά και βιντεοπαιχνίδια. Περιλαμβάνει μια πληθώρα από εργαλεία που επιτρέπουν στους χρήστες του να αναπτύξουν εφαρμογές σε διάφορες γλώσσες προγραμματισμού όπως C#, C++, Visual Basic, F#, JavaScript, και Python. Εκτός από τα Windows, μπορεί να χρησιμοποιηθεί και από χρήστες macOS αφού για πρώτη φορά κυκλοφόρησε για τα mac τον Νοέμβριο του 2016 όπου και έδωσε στην Microsoft την ευκαιρία να επεκτείνει το Visual Studio για σε χρήστες εκτός από του λειτουργικού συστήματος των Windows.

Εκτός από την σύνταξη του κώδικα ένα πολύ σημαντικό χαρακτηριστικό του Visual Studio είναι η ύπαρξη του IntelliSense που καθώς ο προγραμματιστής συντάσει τον κώδικα αυτό του προτείνει προτροπές με συμπλήρωση του κώδικα, υποδείξεις για το τι παραμέτρους παίρνει ή ενδεχομένως τι τιμή επιστρέφει μια συνάρτηση αλλά και γενικές πληροφορίες για την περιγραφή της χρήσης της συνάρτησης. Αυτό το χαρακτηριστικό επιτρέπει στον χρήστη να σώσει πολύτιμο χρόνο αφού δεν χρειάζεται να ανατρέχει στο documentation τις κάθε βιβλιοθήκης και επίσης αν δεν είναι σε θέση να ανακαλέσει το όνομα μια ιδιότητας ενός αντικειμένου τότε το IntelliSense θα του το υποδείξει φέρνοντας στην οθόνη του όλες τις πιθανές ιδιότητες ή συναρτήσεις που είναι ενσωματωμένες στο αντικείμενο.

Η ενσωμάτωση του Github Copilot προτείνει κώδικα που ίσως ενδιαφέρει το χρήστη ανάλογα με το τι συντάσει. Χρησιμοποιώντας το OpenAI Codex που έχει εκπαιδευτεί σε δισεκαττομύρια γραμμές κώδικα είναι σε θέση να προτείνει στον προγραμματιστή ολόκληρες συναρτήσεις ή μια γραμμή κώδικα. Εμπεριέχει ενσωματωμένο σύστημα ελέγχου έκδοσης για το Github όπου κάνει τις διαδικασίες του Github αρκετά πιο εύκολες για χρήστες που δεν είναι τόσο εξοικειωμένοι με την χρήση του Github.

Περιέχει χαρακτηριστικά που επίσης βοηθούν αρκετά στην επίπονη διαδικασία της αποσφαλμάτωσης του κώδικα, δείχνοντας στο χρήστη πόση μνήμη καταναλώνει το πρόγραμμα ανά τις χρονικές περιόδους καθώς επίσης και το ποσοστό της χρήσης του CPU από το πρόγραμμα.

Επίσης επιτρέπει στον χρήστη να διαμορφώσει την διεπιφάνεια του Visual Studio με χρώματα και θέματα όπως αυτός επιθυμεί, δίνοντας την επιλογή για επιλογή θέματος από τα ήδη δημιουργημένα της Microsoft ή άλλων εξωτερικών θεμάτων δημιουργημένα από άλλους χρήστες που βρίσκονται στο Visual Studio. Αυτό το χαρακτηριστικό δίνει την δυνατότητα στον προγραμματιστή να φέρει το περιβάλλον όσο πιο κοντά γίνεται στις χρωματικές επιλογές και τις προτιμήσεις του που βοηθάει στην αύξηση της παραγωγικότητας, διότι όσο πιο οικεία νιώθει κάποιος κατά την αλληλεπίδραση του με το πρόγραμμα τόσο αυξάνεται και η παραγωγικότητα του. Εκτός από αυτό δίνεται η δυνατότητα στον χρήστη να προσθέσει επεκτάσεις που βρίσκονται στο marketplace που βοηθάει σε περαιτέρω ευκολίες ώστε ο προγραμματιστής να “φέρει το Visual Studio στα μέτρα του”. Για παράδειγμα υπάρχουν επεκτάσεις που υλοποιούν χαρακτηριστικά που βρίσκονται σε άλλα ολοκληρωμένα περιβάλλοντα ανάπτυξης κώδικα και τα οποία η Microsoft δεν έχει ενσωματώσει ακόμα στο Visual Studio. Έτσι υπάρχει η ευχέρεια να ανακαλύψει ο προγραμματιστής επεκτάσεις και να τις προσθέσει στο περιβάλλον του.

3.2 Εκδόσεις

Το Microsoft Visual Studio είναι διαθέσιμο σε τρεις διαφορετικές κύριες εκδόσεις, τις Visual Studio Community, Visual Studio Professional και Visual Studio Enterprise.

Supported Features	Visual Studio Community Free download	Visual Studio Professional Buy	Visual Studio Enterprise Buy
⊕ Supported Usage Scenarios	●●●○	●●●●	●●●●
Development Platform Support [2]	●●●●	●●●●	●●●●
⊕ Integrated Development Environment	●●●○	●●●○	●●●●
⊕ Advanced Debugging and Diagnostics	●●○○	●●○○	●●●●
⊕ Testing Tools	●○○○	●○○○	●●●●
⊕ Cross-platform Development	●●○○	●●○○	●●●●
⊕ Collaboration Tools and Features	●●●●	●●●●	●●●●

Σχήμα 3.1: Οι εκδόσεις του Microsoft Visual Studio

Το Community edition είναι απολύτως δωρεάν και είναι για ανεξάρτητους προγραμματιστές που επιθυμούν να αναπτύξουν μόνοι τους την εφαρμογή και μπορεί να χρησιμοποιηθεί για την ανάπτυξη και πόλωση των εφαρμογών που έφτιαξαν μέσα από αυτό. Επίσης είναι η επιλογή για προγραμματιστές ή μαθητές που επιθυμούν να κάνουν τα πρώτα τους βήματα σε μια γλώσσα προγραμματισμού χρησιμοποιώντας ένα από το πιο ισχυρά ολοκληρωμένα περιβάλλοντα ανάπτυξης κώδικα. Τέλος είναι κατάλληλο και για οργανισμούς που τηρούν τις παρακάτω προϋποθέσεις:

- Ο οργανισμός θα πρέπει να διαθέτει λιγότερους από 250 υπολογιστές στην κατοχή του
- Τα ετήσια έσοδα του θα πρέπει να είναι λιγότερα από 1 εκατομμύριο δολάρια.
- Επιτρέπεται η εγκατάσταση του σε μέχρι 5 υπολογιστές

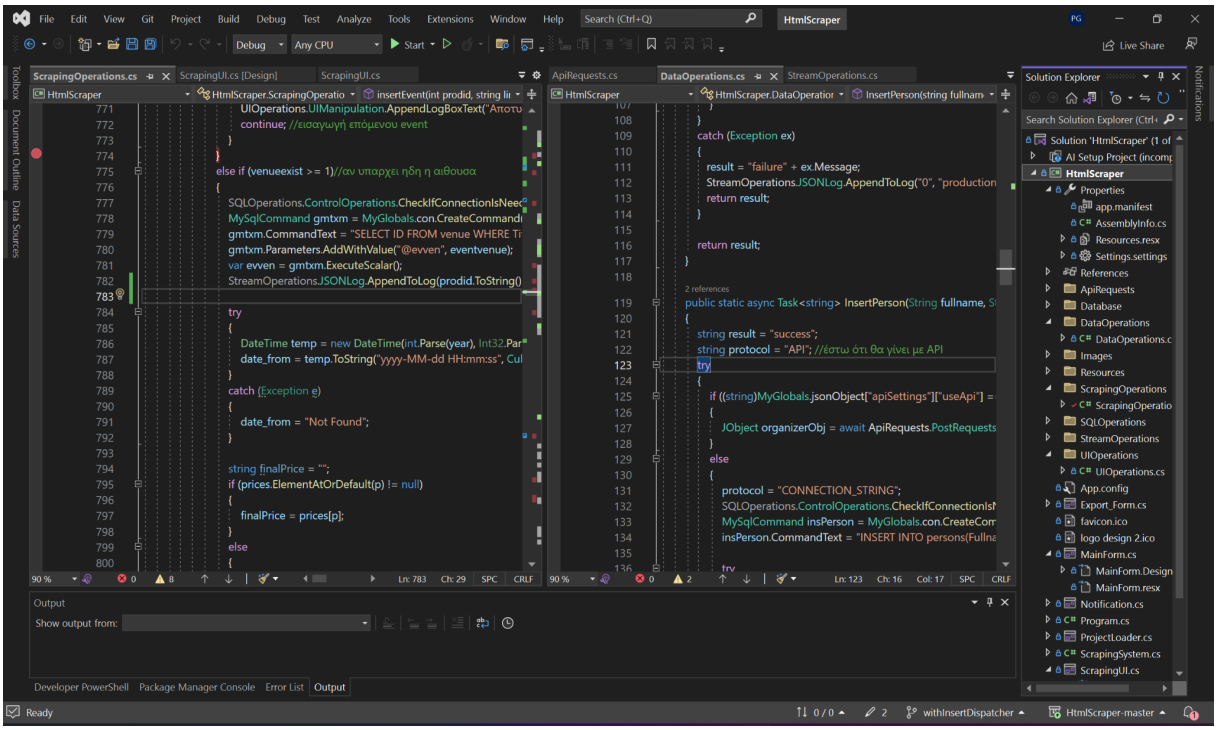
Σε οποιαδήποτε περίπτωση ανήκει κάποιος θα πρέπει να συνδέσει το IDE του με τον λογαριασμό Microsoft που διαθέτει.

Η τιμή του Professional είναι 45 δολάρια ανα μήνα και του Enterprise 250 δολάρια ανα μήνα. Η διαφορά αυτών των δύο μπορεί να φανεί απο το τι δεν μπορεί να κάνει το Professional Edition σε σχέση με το Enterprise και εν τέλη ο οργανισμός/προγραμματιστής να αποφασίσει ποιά έκδοση επιθυμεί να αγοράσει. Όσον αφορά την ανάπτυξη εφαρμογών για πολλά λειτουργικά συστήματα που επιτυγχάνεται με το Xamarin, ως προς αυτό η Enterprise διαθέτει Profiler που βοηθάει στην δυναμική ανάλυση του προγράμματος καθώς αυτό τρέχει και παρέχει κρίσιμες πληροφορίες για τα κομμάτια κώδικα που εκτελούνται τις περισσότερες φορές, βοηθώντας να δείξει στους προγραμματιστές που πρέπει να συγκεντρώσουν την προσοχή τους ώστε να βελτιώσουν τον πρόγραμμα τους. Επίσης παρέχει πληροφορίες για την χρονική πολυπλοκότητα των διαδικασιών αλλά και για τον τρόπο που επιτυγχάνεται η κατανομή της μνήμης. Εκτός από αυτό η Enterprise έκδοση παρέχει μια πληθώρα εργαλείων που βοηθούν στο testing της εφαρμογής.

3.3 Διεπαφή

Η Διεπαφή του Microsoft Visual Studio είναι πλήρως παραμετροποιήσιμη δίνοντας την δυνατότητα στον χρήστη να ενσωματώσει και να οργανώσει τα αγαπημένα του στοιχεία παραθύρων με τον τρόπο

με τον οποίον αυτός επιθυμεί να διατάξει το ολοκληρωμένο περιβάλλον ανάπτυξης κώδικα. Αποτελείται στην κορυφή του απο την μπάρα του μενού η οποία διαθέτει πολλά στοιχεία υπομενού.



Σχήμα 3.2: Η Διεπαφή του Microsoft Visual 2022

Με την επιλογή File δίνει την δυνατότητα στον χρήστη να ανοίξει νέα αρχεία, να αποθηκεύσει αλλά και να ανοίξει αρχεία που ενδεχομένως να έκλεισε καταλάθος. Ένα πολύ σημαντικό χαρακτηριστικό που βρίσκεται σε αυτό το μενού είναι το live share session όπου δίνει την δυνατότητα στους χρήστες να κάνουν ταυτόχρονη επεξεργασία του κώδικα της βλέποντας ζωντανά τι αλλαγές κάνει ο καθένας. Στην εποχή της τηλεργασίας και όπου ο προγραμματισμός είναι ένας από τους τομείς που μπορεί να επιτευχθεί πιο εύκολα η τηλεργασία, αυτό το χαρακτηριστικό του Visual Studio μπορεί να βοηθήσει τους προγραμματιστές να συνεργαστούν ακόμα καλύτερα και πιο άμεσα. Το live share μπορεί να χρησιμοποιηθεί ανεξάρτητα της γλώσσας προγραμματισμού, είδους της εφαρμογής ή λειτουργικού συστήματος του χρήστη. Επίσης καθώς οι χρήστες κάνουν αλλαγές για παράδειγμα σε μια διαδικτυακή εφαρμογή που τρέχει τοπικά στο μηχάνημα του τοπικού υπολογιστή, καθώς γίνεται αυτή η αλλαγή θα εμφανιστεί αυτόματα και στα δύο παράθυρα του φυλλομετρητή των χρηστών, δηλαδή και του τοπικού αλλά και του απομακρυσμένου υπολογιστή. Αυτό όπως καταλαβαίνουμε αυξάνει την παραγωγικότητα αφού δεν σπαταλάται χρόνος για την μεταφορά του project και των αρχείων που το αποτελούν από υπολογιστή σε υπολογιστή. Τέλος το live share δίνει την δυνατότητα αποστολής μηνυμάτων κατά την διάρκεια σύνταξης του κώδικα ή για την προσθήκη σημειώσεων για τις διάφορες συναρτήσεις ή γραμμές κώδικα που υπάρχουν.

Η επιλογές του μενού View ίσως είναι από τις πιο σημαντικές αφού δίνουν την δυνατότητα στον χρήστη να εμφανίσει παράθυρα που είτε δεν είναι ανοιγμένα κατά την έναρξη του Visual Studio αλλά και είτε τα έκλεισε κατά λάθος ο χρήστης. Αφού ανοίξει ένα υπο-παράθυρο ο χρήστης μπορεί να το σύρει προς ένα σημείο ώστε να μείνε σε εκείνο το συγκεκριμένο σημείο. Για παράδειγμα θα μπορεί να ανοίξει το υποπαράθυρο των Properties.

Απο το μενού Git και Debug μπορεί να εκτελέσει ενέργειες που σχετίζονται με το Github και την αποσφαλμάτωση του κώδικα και οι οποίες θα αναλυθούν στα επόμενα υποκεφάλαια.

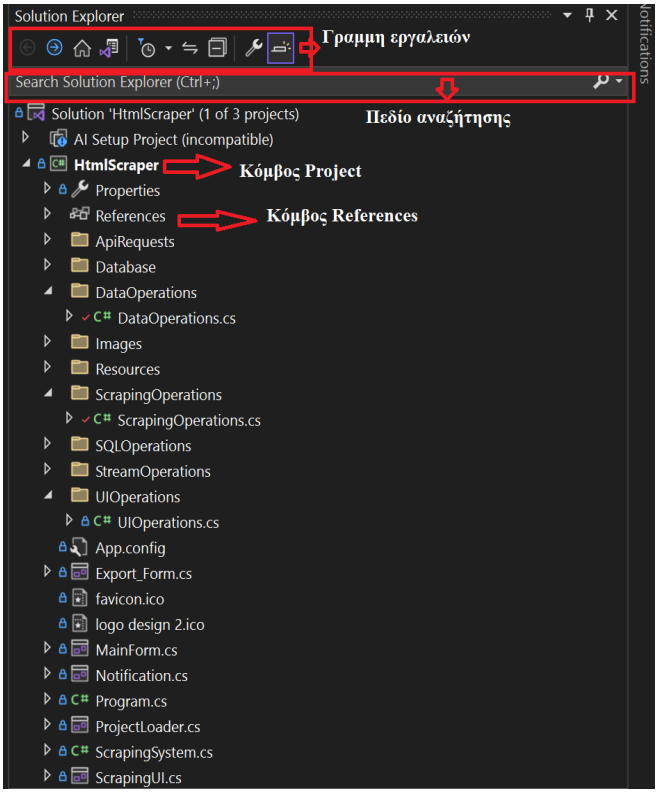
Απο το παράθυρο Windows να αποθηκεύσει την διάταξη των παραθύρων όπως αυτός τα έχει παραμετροποιήσει, με αποτέλεσμα την επόμενη φορά που θα ανοίξει το ολοκληρωμένο περιβάλλον ανάπτυξης κώδικα να εκκινήσει με τις αλλαγές όπως αυτός της αποθήκευσε.

Ένα ακόμα στοιχείο του IDE είναι οι πολλές γραμμές εργαλείων που καθιστούν τον χρήστη σε θέση να κάνει γρήγορη επιλογή της ενέργειας που επιθυμεί να εκτελέσει.

Γενικά η διεπαφή του Microsoft Visual Studio δίνει την δυνατότητα στους χρήστες να την παραμετροποιήσουν όπως αυτοί επιθυμούν και να έχουν τα παράθυρα τα οποία αυτοί χρησιμοποιούν πιο πολύ μπροστά τους.

3.3.1 Solution Explorer

Το Solution Explorer του Microsoft Visual Studio δείχνει στον χρήστη την δυνατότητα να δει, να ανοίξει και να επεξεργαστεί όλα τα αρχεία που σχετίζονται με το solution στο οποίο δουλεύει. Το IDE έρχεται προκαθορισμένο έτσι ώστε να ανοίγει το Solution Explorer στο δεξί μέρος του. Αν δεν εμφανίζεται αυτό, τότε υπάρχουν δύο τρόποι για να το ανοίξει ο χρήστης. Ο πρώτος είναι από το μενού επιλέγοντας το View και στην συνέχεια κάνοντας επιλογή το Solution Explorer. Ένας δεύτερος τρόπος να το ανοίξουμε πατόντας την τον συνδυασμό των πλήκτρων Ctrl + Alt + L που αποτελεί και την συντόμευση του. Αφού ανοίξει μπορούμε να το σύρουμε (όποιος και οποιοδήποτε παράθυρο του Visual Studio) και να το τοποθετήσουμε εκεί που επιθυμούμε.



Σχήμα 3.3: Microsoft Visual 2022 Solution Explorer

Αποτελείται από την γραμμή εργαλείων που κατα κύριο λόγο δίνει επιλογές για την εμφάνιση των αρχείων, την μπάρα αναζήτησης μέσω αυτής δίνεται η δυνατότητα στο χρήστη της γρήγορης αναζήτησης των αρχείων και τύπων των αρχείων που σχετίζονται με το ανάλογο solution. Το κύριο παράθυρο δείχνει όλα τα αρχεία του solution. Πολύ σημαντικό χαρακτηριστικό είναι η ύπαρξη του κόμβου References που δείχνει ποιες βιβλιοθήκες χρησιμοποιούνται.

Με δεξί κλικ στον κόμβο του Project εμφανίζεται ένα μενού όπου μπορούμε να πραγματοποιήσουμε ενέργειες που σχετίζονται με αυτό. Κάνοντας επιλογή το Add μπορούμε να προσθέσουμε ένα νέο ή και ένα υπάρχων αρχείο, εκτός από αυτό μπορεί να γίνει και προσθήκη μιας επιπλέον Windows Form όπου θα σχεδιαστεί και θα συνδεθεί με την λειτουργία του κώδικα ώστε να εμφανίζεται.

Επίσης μπορεί να γίνει και η διαχείριση των πακέτων NuGet. Το NuGet είναι ένας διαχειριστής πακέτων για εφαρμογές .NET όπου επιτρέπει στους χρήστες να βρουν, να διαχειριστούν και να εγκαταστήσουν πακέτα τα οποία θα χρησιμοποιήσουν στον κώδικα τους μέσα από το Visual Studio. Μόλις γίνει εγκατάσταση ενός πακέτου, τότε ο κώδικας του καθώς και όλες οι βιβλιοθήκες απο τις οποίες εξαρτάταται, γίνονται και αυτά εγκατάσταση και προστίθενται στο project. Μέσα από τον κόμβο References ο χρήστης μπορεί να δει και τα πακέτα τα οποία έχουν εγκατασταθεί απο το NuGet Manager. Καταλήγουμε λοιπόν ότι το NuGet είναι ένα σημαντικό εργαλείο το οποίο μπορεί να βοηθήσεις τον χρήστη να εγκαταστήσει πακέτα εύκολα, αλλά κυρίως ένα μέρος για να αναζητήσεις πακέτα που θα τον βοηθήσουν να αυξήσει την παραγωγικότητα του, καθώς δεν θα χρειάζεται να υλοποιήσει τις λειτουργίες που ήδη υλοποιεί ένα πακέτο το οποίο μπορεί να εγκαταστήσει χωρίς χρέωση.

3.3.2 Toolbox

Το παράθυρο Toolbox εμφανίζει στον χρήστη όλα τα στοιχεία που μπορεί να προσθέσει ο χρήστης στην Windows Form ή XAML αρχείου, του κάνοντας drag-and-drop του στοιχελιου στην φορμα ή κάνοντας διπλό κλικ στο στοιχεία και αυτό θα εμφανιστεί αυτόματα στην φόρμα. Να σημειωθεί οτι το Toolbox εμφανίζεται μόνο σε Designer View και αν το αρχείο είναι σχετικό με ένα αρχείο XAML ή Windows Forms App καθώς επίσης και εμφανίζει μόνο τα στοιχεία τα οποία μπορούν να χρησιμοποιηθούν. Ο χρήστης μπορεί να ανοίξει το παράθυρο του Toolbox μέσω του μενού επιλέγοντας View > Toolbox καθώς και με την χρήση της συντόμευσης Ctrl + Alt + X.

Απο το πεδίο αναζήτησης του μπορεί ο χρήστης να βρει γρήγορα το στοιχείο που επιθυμεί. Με δεξι κλικ σε κάποιο στοιχείου του toolbox ο χρήστης μπορεί να εκτελέσει ενέργειες όπως: διαγραφή στοιχείου, μετονομασία στοιχείου, μετακίνηση ενός επιπέδου πάνω ή κάτω, ταξινόμηση των στοιχείων αλφαβητικά. Επίσης μπορεί με drag-and-drop να μετακινήσει τα στοιχεία μέσα στην περιοχή του toolbox και να τα διατάξει με τον τρόπο που ο χρήστης επιθυμεί, όπως ανάλογα με την σειρά της πιθανότητας χρήσης τους, πράγμα που βοηθάει τον χρήστη να βρίσκει γρήγορα τα στοιχεία τα οποία θέλει να χρησιμοποιήσει. Τέτοια απλά αλλά σημαντικά χαρακτηριστικά του Microsoft Visual Studio του δίνουν την δυνατότητα να παραμετροποιηθεί πλήρως όπως ο χρήστης το φαντάζεται και να το φέρει στα μέτρα του. Επίσης ο χρήστης μπορεί να δημιουργήσει τα δικά του custom στοιχεία και να τα προσθέσει στο toolbox μέσω της χρήσης του template του WPF και αυτό θα προστεθεί αυτόματα στο toolbox του χρήστη. Βεβαίως, κατά την διαδικασία της παραμετροποίησης του Toolbox κάτι μπορεί να πάει στραβά και ο χρήστης να διαγράψει κάποια στοιχεία ή να μην του αρέσει η ταξινόμηση των στοιχείων, γι αυτό και δίνεται η δυνατότητα αναίρεσης των αλλαγών κάνοντας δεξί κλικ σε ένα στοιχείο και επιλέγοντας Reset Toolbox.

3.3.3 Properties

Η χρήση του παραθύρου Properties δίνει την δυνατότητα στον χρήστη να δει αλλά και να αλλάξει όλες τις ιδιότητες που σχετίζονται με το επιλεγμένο στοιχείο που έκανε κλικ στο design view. Αν το παράθυρο δεν εμφανιστεί αυτόματα κατά την επιλογή ενός στοιχείου τότε υπάρχουν δύο επιλογές για την εμφάνιση του. Η πρώτη είναι από το μενού, επιλέγοντας View και στην συνεχεια το Properties window που βρίσκεται προτελευταίο στην λίστα, ενώ η δεύτερη επιλογή είναι με την χρήση της συντόμευσης που είναι το πλήκτρο F4. Το παράθυρο των ιδιοτήτων εμφανίζει διαφορετικες ιδιότητες ανάλογα με τον τύπο του στοιχείου που επιλέχθηκε. Εκτός από τις ιδιότητες ο χρήστης μπορεί να δει και να αλλάξει τα events που σχετίζονται με τα components.

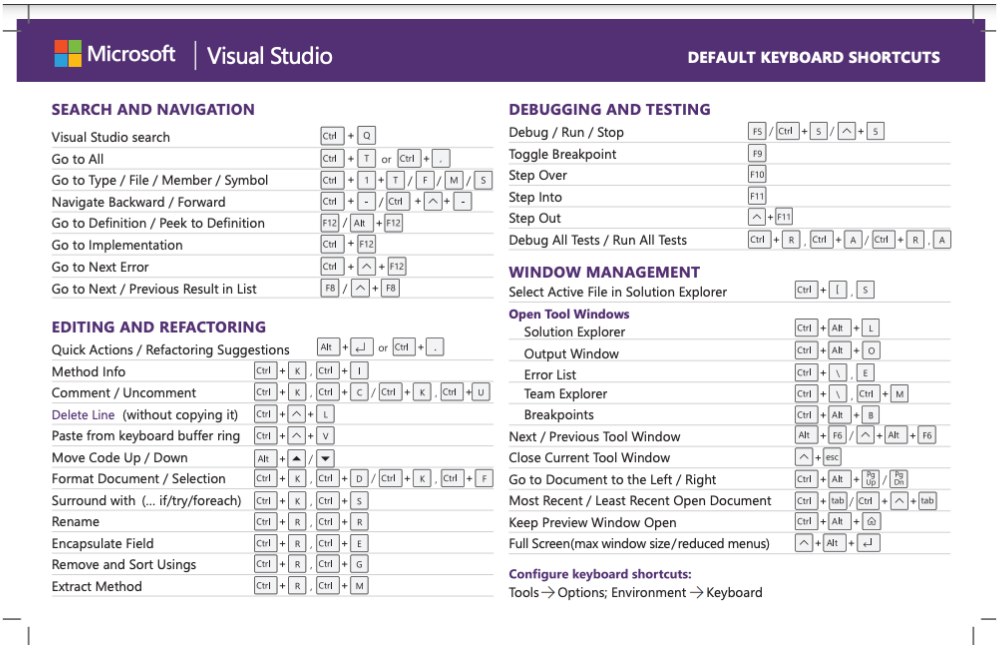
Κάποιες από τις πιο χρήσιμες ιδιότητες των components των Windows Forms και η χρήση τους είναι:

Όνομα	Περιγραφή
Name	Το όνομα του Control. Η προκαθορισμένη του τιμή είναι “”. Χρησιμοποιείται κυρίως για εύρεση του στοιχείου και το όνομα πρέπει να είναι μοναδικό
Parent	Χρησιμοποιείται κυρίως για να αναφερθεί στον πατέρα του Control ή για την διάταξη των γραφικών στοιχείων ώστε να εμφανιστεί σε ένα συγκεκριμένο control του UI
Location	Η τιμή του είναι ένα Point (X,Y) και αναπαριστά την απόσταση του στοιχείου από πάνω και δεξιά σε σχέση με το Control του πατέρα του.
Size	Αναπαριστά την τιμή του ύψους και του πλάτους του Control
Enabled	Αν η τιμή του είναι true τότε το Control θα αλληλεπιδράσει στις ενέργειες του χρήστη
Visible	Αν η τιμή του είναι true τότε το Control θα εμφανίζεται στον χρήστη
Tag	Η τιμή του είναι ένα αντικείμενο. Αρκετά χρήσιμο αν θέλουμε να αποθηκεύσουμε πληροφορίες σε ένα Control και να τις ανακτήσουμε σε δεύτερο χρόνο
AccessibleDescription	Η τιμή του είναι μια συμβολοσειρά που περιγράφει το Control. Χρησιμοποιείται κυρίως για να δώσει πληροφορίες σε άτομα με δυσκολίες όρασης.
Dock	Η τιμή του είναι ένα DockStyle. Καθορίζει τον τρόπο με τον οποίο εμφανίζεται στο Container του
TabIndex	Η τιμή του είναι ένας ακέραιος και προσδιορίζει το πως γίνονται focus τα Controls όταν πατιέται το κουμπί Tab
CanFocus	Αν η τιμή του είναι true τότε το Control μπορεί να δεχθεί focus
AutoSize	Αν η τιμή του είναι true, τότε το μέγεθος του Control θα τροποποιηθεί ανάλογα με το κείμενο του
RightToLeft	Αν η τιμή του είναι true, τότε η διάταξη του κειμένου θα εμφανιστεί από τα δεξιά προς τα αριστερά
Cursor	Η τιμή του είναι ένα Cursor και ανάλογα με την τιμή του θα αλλάξει και ο δείκτης του ποντικιού

Πίνακας 3.1: Χρήσιμα Properties των components των Windows Form

3.3.4 Συντομεύσεις

Οι συντομεύσεις πληκτρολογίου είναι το πάτημα κουμπιών ή συνδυασμού κουμπιών που όταν πατηθούν μαζί εκτελούν μια ενέργεια σε ένα συγκεκριμένο πρόγραμμα. Οι συντομεύσεις είναι ένας γρήγορος και αποτελεσματικός τρόπος με τον όποιον οι χρήστες αλληλεπιδρούν με πρόγραμμα και βοηθούν στην αύξηση της ταχύτητας της εκτέλεσης λειτουργιών και επίσης βοηθόντας χρήστες με προβλήματα όρασης που έχουν προβλήματα με την όραση του ποντικιου. Επομένως θα μπορούσαμε να πούμε ότι οι συντομεύσεις αυξάνουν την παραγωγικότητα του χρήστη. Κάποιες απο τις συντομεύσεις για το Microsoft Visual Studio στο λειτουργικό σύστημα των Windows φαίνονται στο παρακάτω σχήμα



Σχήμα 3.4: Συντομεύσεις του Microsoft Visual Studio

Γενικά οι συντομεύσεις πληκτρολογίου μπορούν να αποτελέσουν ένα πολύτιμο εργαλείο που για την αύξηση της αποδοτικότητας του χρήστη και αύξησης της ταχύτητας με την οποία εκτελεί συγκεκριμένες διαδικασίες που επαναλαμβάνονται συνεχώς. Επίσης δίνουν σημαντική βοήθεια για άτομα με προβλήματα όρασης. Ένα από τα πλεονεκτήματα του Microsoft Visual Studio είναι η πολύ καλή τεκμηρίωση του προγράμματος και αυτό δεν πάει πίσω ακόμα και στις συντομεύσεις.

3.4 Version Control μέσω Github

Το Github είναι μια πλατφόρμα που βοηθάει τους προγραμματιστές ως προς τον έλεγχο της έκδοσης του κώδικα αλλά και της βελτίωσης της συνεργασίας μεταξύ αυτών. Κυκλοφόρησε για πρώτη φορά το 2008 και αγοράστηκε από την Microsoft το 2018 για 7.5 δισεκατομμύρια δολάρια. Επιτρέπει στους προγραμματιστές να ανεβάζουν τον κώδικα τους στους διακομιστές του GitHub, που βοηθάει ιδιαίτερα όταν οι χρήστες δουλεύουν από διαφορετικούς υπολογιστές να πάρουν τον κώδικα ή την νέα έκδοση του κώδικα και σε άλλους υπολογιστές. Επίσης παρέχει δυνατότητες για την συνεργασία πολλών προγραμματιστών σε μια ενιαία και ενημερωμένη βάση κώδικα, επιτρέποντας τους να βλέπουν τις αλλαγές που έχουν πραγματοποιηθεί στις διάφορες κλάσεις/αρχεία αλλά και να προσθέτουν σχόλια για τις αλλαγές που έκαναν ή που πρέπει να κάνουν.

Κάποια από τα πλεονεκτήματα του Github είναι ότι βοηθούν τις ομάδες ανάπτυξης κώδικα να οργανωθούν αφού τους παρέχει όλα τα εργαλεία για την διαχείριση του έργο τους όπως issue tracking, έλεγχο των αλλαγών του κώδικα αλλά και ασφαλείς αλλαγές κώδικα αφού δίνει την δυνατότητα στους χρήστες να κάνουν αλλαγές στον κώδικα χωρίς να επηρεάζουν τα επίσημα αρχεία κώδικα (μέσω των branches που θα αναλυθούν παρακάτω) που ενδεχομένως να χρησιμοποιούνται για τις εκδόσεις του προγράμματος που αναπτύσουν οι προγραμματιστές και είναι η επίσημη.

Το Github βασίζεται στο Git. Το Git είναι ένα ανοιχτού κώδικα εργαλείο που σχεδιάστηκε το 2005 απο τον Linus Torvalds με σκοπό να ιχνηλατεί τις αλλαγές που γίνονται στον κώδικα.

3.4.1 Branch και Repository

Ένα repository στο Github είναι μια κεντρική τοποθεσία όπου οι προγραμματιστές μπορούν να αποθηκεύσουν, να διαχειριστούν αλλά και να μοιράσουν των κώδικα τους και όλα τα αρχεία που σχετίζονται με αυτό με άλλους. Ένα repository μπορεί είτε να είναι δημόσιο είτε ιδιωτικό. Τα δημόσια είναι διαθέσιμα για οποιονδήποτε στο διαδίκτυο, ενώ στα ιδιωτικά επιτρέπεται η πρόσβαση μόνο στον δημιουργό αλλά και στους χρήστες που ο δημιουργός θα δώσει τα δικαιώματα πρόσβασης.

Στο Github ένα repository μπορεί να είναι είτε τοπικό είτε απομακρυσμένο. Τα τοπικά είναι αποθηκευμένα στον υπολογιστή του χρήστη ενώ τα απομακρυσμένα είναι αποθηκευμένα στους διακομιστές του Github και είναι διαθέσιμα για άλλους χρήστες οι οποίοι έχουν πρόσβαση στο συγκεκριμένο repository. Κατά την δημιουργία ενός repository στο GitHub, αυτόματα τα περιεχόμενα του πηγαίνουν και στο απομακρυσμένο repository στους διακομιστές. Αυτό επιτρέπει στους προγραμματιστές που έχουν πρόσβαση να μπορούν να το κάνουν clone και να το έχουν στον δικό τους υπολογιστή τοπικά. Επομένως, κατά την πρώτη δημιουργία του, όλα τα περιεχόμενα του μεταφέρονται στους διακομιστές του Github. Έπειτα οι προγραμματιστές μπορούν να συνεχίσουν την δουλειά τους τροποποιόντας τον κώδικα τους, γνωρίζοντας ότι δεν θα χάσουν την λειτουργικότητα και τα αρχεία του απομακρυσμένου repository. Αφού ο χρήστης είναι ευχαριστημένος με την δουλειά που έκανε μπορει να **commit** τις αλλαγές του. Κατά την φάση του commit όλες οι αλλαγές που έχουν γίνει από τον προγραμματιστή πηγαίνουν στο τοπικό repository του GitHub στον υπολογιστή του χρήστη. Αν ο χρήστης κάνει **push** τότε όλες αυτές οι αλλαγές μεταφέρονται στο απομακρυσμένο repository.

Μπορούμε να παρατηρήσουμε ότι τα repositories βοηθούν στην συνεργασία μεταξύ των χρηστών παρέχοντας τους ένα μέρος όπου μπορούν όλοι όσοι έχουν πρόσβαση σε αυτό να έχουν έναν ενιαίο κώδικα, γεγονός που μηδενίζει προβλήματα όπου οι προγραμματιστές δεν είχα την τελευταία έκδοση του κώδικα και προγραμμάτισαν πάνω σε αυτήν. Έτσι μπορούν να βλέπουν τι αλλαγές έχει κάνει ο κάθε χρήστης και να αναλύουν τις αλλαγές που έχουν πραγματοποιηθεί. Εκτός από αυτό βοηθούν στον έλεγχο της έκδοσης του κώδικα, αφού κάθε commit εμπεριέχει και μια περιγραφή από τον συντάκτη που έκανε τις αλλαγές που παρέχει μια περιγραφή ως προς το τι αλλαγές έκανε και για ποιόν λόγο. Αυτή η διαδικασία μπορεί να βοηθήσει πάρα πολύ στην αποσφαλμάτωση του κώδικα, αφού κατά την προσπάθεια υλοποίησης νέων χαρακτηριστών στα προγράμματα γίνονται αλλαγές που μπορεί να επηρεάσουν και να φέρουν προβλήματα σε ήδη υπάρχοντα λειτουργικά χαρακτηριστικά. Με αυτόν τον τρόπο μπορούν να βρεθούν οι αλλαγές και να περιοριστεί η περιοχή του προβλήματος, βοηθώντας τους προγραμματιστές και κάνοντας πολύ πιο εύκολη την επίπονη διαδικασία της αποσφαλμάτωσης του κώδικα.

Κάθε repository κατά την δημιουργία του έχει ένα αυτόματα δημιουργημένο προεπιλεγμένο branch με την ονομασία main και κάθε repository μπορεί να έχει περισσότερα από ένα branches.

Αυτά είναι ένα πολύ καλός τρόπος έτσι ώστε να γίνονται αλλαγές χωρίς να επηρεάζεται ο επίσημος κώδικας. Για παράδειγμα αν κάποιος χρήστης αναλάβει την υλοποίηση ενός feature που μπορεί να επηρεάσει πολλές περιοχές κώδικα, τότε μπορεί να δημιουργήσει ένα νέο branch όπου θα είναι αντίγραφο του κύριου branch και ότι αλλαγές κάνει commit θα μεταφέρονται στο καινούριο branch. Έτσι δίνεται η δυνατότητα στους προγραμματιστές να αναπτύσουν νέα χαρακτηριστικά δίχως να έχουν τον φόβο ότι οι αλλαγές που κάνουν θα χαλάσουν την ολική λειτουργικότητα του προγράμματος, αφού η δημιουργία νέου branch τους δίνει την δυνατότητα να κάνουν τις αλλαγές τους σε απομονωμένη ανάπτυξη χωρίς να τροποποιούν το κύριο branch.

Ένα άλλο σημαντικό χαρακτηριστικό των branches είναι ότι βοηθούν την παράλληλη ανάπτυξη κώδικα από πολλαπλούς χρήστες. Έτσι κάθε προγραμματιστής που δουλεύει σε ένα συγκεκριμένο repository μπορεί να δημιουργήσει το δικό του branch με την ονομασία που θα το δώσει και να κάνει τις αλλαγές του χωρίς να επηρεάζουν αυτές τις αλλαγές άλλων προγραμματιστών που έχουν ανατεθεί να αναπτύξουν νέα χαρακτηριστικά και δουλεύουν στο δικό τους απομονωμένο branch.

Εάν όλοι οι προγραμματιστές έχουν ολοκληρώση την ανάπτυξη των χαρακτηριστικών που υλοποιήσουν στο branch τους, μπορούν να κάνουν **merge** το branch τους με το main branch.

Οι διαχειριστές των repositories μπορούν να παρέχουν προστασία σε ένα branch χαρακτηρίζοντάς το ως protected. Αν κάποιο έχει αυτόν τον χαρακτηρισμό, τότε δεν μπορεί να διαγραφεί

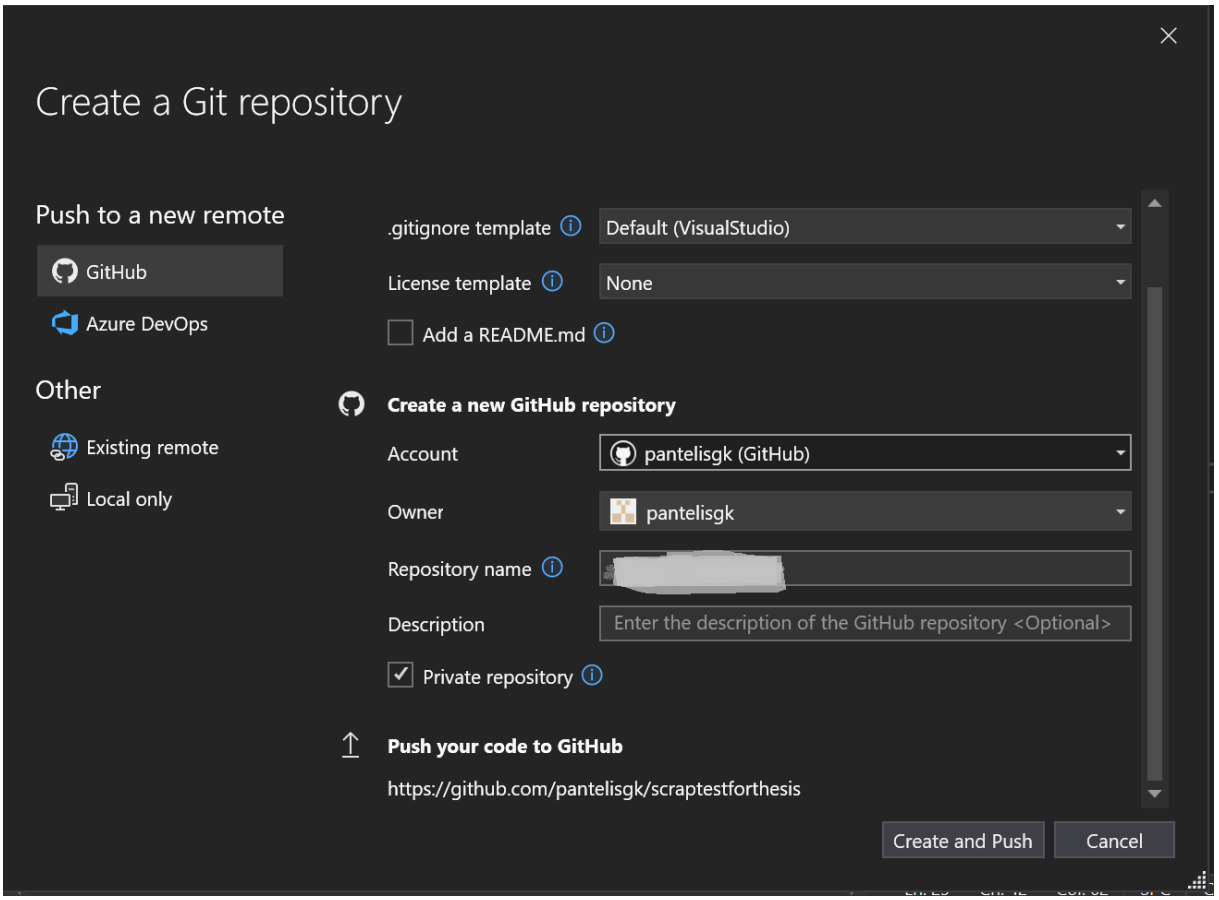
3.4.2 Σύνδεση του Github μέσα από το Visual Studio

Το Microsoft Visual Studio έχει ενσωματώσει τις λειτουργίες του Github μέσα στο ολοκληρωμένο περιβάλλον ανάπτυξης κώδικα, πράγμα που σημαίνει ότι οι χρήστες του δεν χρειάζεται να εκτελούν τις λειτουργίες του Github μέσα από την εκτέλεση των εντολών από το command line ή μέσω του προγράμματος Github desktop ή άλλου εργαλείου που κάνει την αντίστοιχη δουλειά. Έτσι οι προγραμματιστές είναι σε θέση να εκτελούν όλες τις ενέργειες που σχετίζονται με τον Github μέσα από το περιβάλλον IDE που έχουν εξοικειωθεί, κάνοντας και το Github πιο προσβάσιμο σε νέους χρήστες που θέλουν να γνωρίσουν και να εγκλιματιστούν με αυτό. Επομένως αποτελεί ένα πολύ καλό τρόπο γνωριμίας και εξοικειωσης νέων χρηστών κατά την ανάπτυξη μέσα από το Visual Studio.

Αρχικά ο χρήστης θα πρέπει να έχει έναν λογαριασμό στο Github ώστε μπορεί να συνδέσει το IDE του με τον λογαριασμό του στο Github. Αφού ανοίξει το project του στο Microsoft Visual Studio κάτω δεξιά θα πρέπει να επιλέξει την επιλογή Add to Source Control και απο την λίστα που θα εμφανιστεί την επιλογή Git. Στην συνέχεια θα επιλέξει την επιλογή GitHub που βρίσκεται κάτω από την προτροπή Push to a New Remote. Η επιλογή License Template αναφέρεται σε ήδη προκαθορισμένες άδειες που αναφέρουν τι δικαιώματα έχουν οι χρήστες που θα δουν τον κώδικα ως προς την διανομή αλλά και την χρήση του. Στην επιλογή Account ο χρήστης θα ακολουθήσει τα βήματα και να πραγματοποιήσει την σύνδεση στο GitHub και στην συνέχεια θα συνδεθεί το IDE με τον λογαριασμό που έγινε η σύνδεση. Στο Repository Name θα δοθεί η ονομασία που θα εμφανίζεται στον GitHub.

Όπως αναφέρθηκε προηγουμένως ένα repository μπορεί να είναι ιδιωτικό ή δημόσιο, επομένως αν ο χρήστης επιλέξει την επιλογή για Private Repository τότε αυτό αυτόματα θα γίνει ιδιωτικό και διαθέσιμο μόνο για τον χρήστη.

Με την επιλογή Create and Push θα πραγματοποιηθεί η δημιουργία του repository στο GitHub και θα γίνει αυτόματα push στους διακομιστές του Github.



Σχήμα 3.5: Δημιουργία repository στο Github μέσω του Microsoft Visual Studio

Αφού ολοκληρωθεί η διαδικασία κάτω δεξιά θα δούμε ότι αυτόματα έχει δημιουργηθεί το branch master και ότι είμαστε στο repository που φτιάξαμε. Επιλέγοντας πάνω στο branch δίνεται η δυνατότητα δημιουργίας νέων branch.

Για την δημιουργία νέου branch το μόνο που έχει να κάνει ο χρήστης είναι να επιλέξει το branch, και να κάνει επιλογή του Create branch. Θα του εμφανιστεί ένα παραθύρο με δύο πεδία εισαγωγής, το πρώτο είναι για την ονομασία του νέου branch και το δεύτερο είναι ποιού αντίγραφου να είναι το νέο branch. Αφού πατήσει create θα δει ότι κάτω δεξιά στην οθόνη του είναι αυτόματα επιλεγμένο το νέο branch που μόλις δημιουργήθηκε. Από αυτό το σημείο ότι ενέργειες κάνει θα σχετίζονται με το branch που εμφανίζεται στην οθόνη.

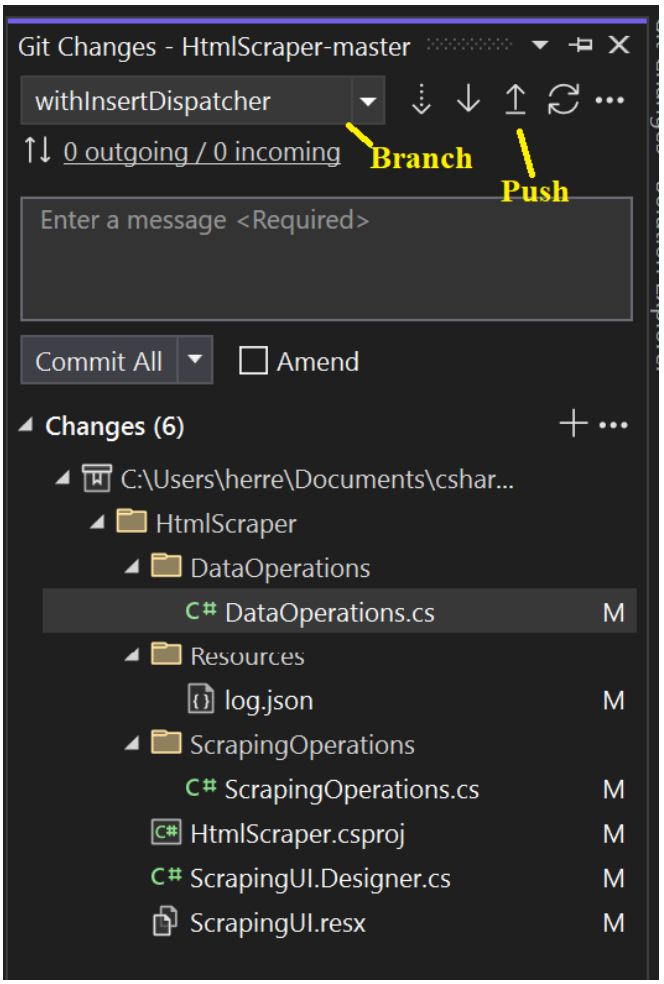
Επίσης αν ο χρήστης έχει δημιουργήσει και άλλα repositories μέσα από το Visual Studio στις επιλογές των repositories μπορεί να τα επιλέγει και να τα αλλάζει πολύ γρήγορα.

Από αυτό το σημείο και μετά κάθε φορά που ο χρήστης θα ανοίγει το συγκεκριμένο project θα είναι αυτόματα συνδεδεμένο με το Github και το repository που δημιουργήθηκε.

Ένας τρόπος να βλέπει ο χρήστης όλες τις αλλαγές που έγιναν στο συγκεκριμένο branch είναι επιλέγοντας το View Branch History που θα το βρει στις επιλογές το μενού Git. Αυτή η οθόνη θα του εμφανίσει όλες τις αλλαγές που έγιναν στο branch μέσα από τα push, δείχνοντας του σε έναν πίνακα ποιός χρήστης το πραγματοποίησε, την ημερομηνία και το σχόλιο που άφησε. Κάνοντας διπλό κλικ στην γραμμή θα εμφανιστεί ο κώδικας όπως ήταν πριν στα αριστερά και τα δεξιά η αλλαγή που έκανε στον κώδικα γύρω από πράσινο φόντο, δίνοντας μια πολύ καλή περίληψη στον χρήστη για τις αλλαγές που έγιναν.

3.4.3 Βασικές Λειτουργίες

Οι ενέργειες για commit και push για το GitHub μπορούν να γίνουν πολύ εύκολα και γρήγορα με την χρήση του Microsoft Visual Studio. Αφού ο προγραμματιστής κάνει την ανάπτυξη του κώδικα και θεωρήσει ότι έφτασε σε σημείο ώστε να να προωθήσει τις αλλαγές στο GitHub θα πρέπει να ακολουθήσει την παρακάτω διαδικασία. Αρχικά, θα πρέπει να ανοίξει το παράθυρο Git Changes. Για να γίνει θα πρέπει να το ανοίξει απο το View -> Git Changes ή με την χρήση της συντόμευσης Ctrl + 0, Ctrl + G.



Σχήμα 3.6: Παράθυρο Git Changes του Microsoft Visual Studio

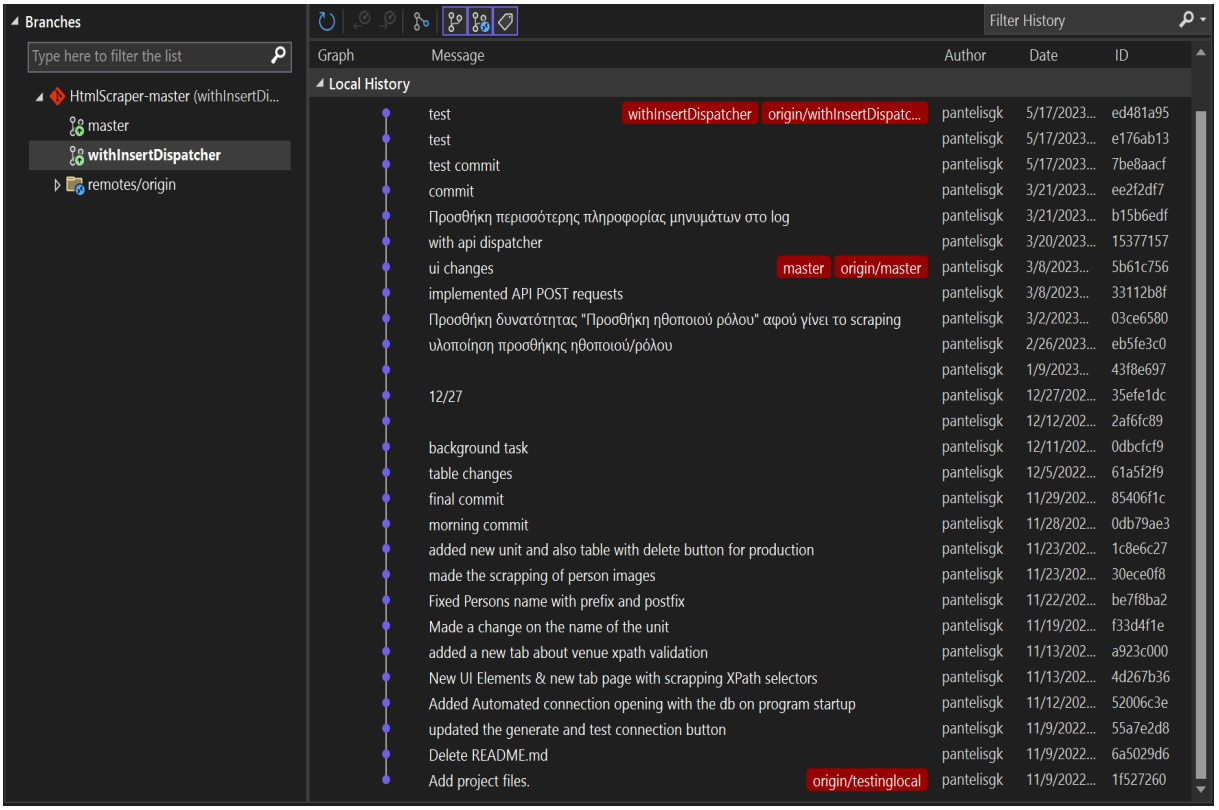
Στον κόμβο Changes υπάρχει η ιεραρχική δομή των αρχείων στο repository και εμφανίζονται μόνο τα αρχεία στα οποία έκανε ο χρήστης αλλαγές από την τελευταία φορά που έγινε το push. Με διπλό κλικ σε ένα από αυτά τα αρχεία ο χρήστης θα δει στο αριστερό του παράθυρο πως ήταν πριν τις αλλαγές και στο δεξί του παράθυρο πως είναι αυτή την χρονική στιγμή, ενώ με πράσινο θα είναι οι γραμμές κώδικα που προσέθεσε στο αρχείο. Επίσης, με δεξί κλικ και View History θα του εμφανιστεί ένας πίνακας με τις αλλαγές που έγιναν με βάση τα pushes που έκανε, ενώ μπορεί να δει και πως ήταν το αρχείο εκείνη την χρονική στιγμή.

Για να πραγματοποιηθεί η λειτουργία του push, αρχικά ο χρήστης θα πρέπει να γράψει κάποιο σχόλιο για τις αλλαγές που πραγματοποίησε στον κώδικα, πράγμα που θα βοηθάει κατά τον έλεγχο του ιστορικού των αλλαγών αφου θα φαίνεται τι είδους ήταν οι αλλαγές που πραγματοποιήθηκαν. Αφού γραφτεί το σύντομο σχόλιο, πάνω από αυτό το πεδίο θα επιλέξει το branch που επιθυμεί να περάσουν οι αλλαγές

Στην συνέχεια ακολουθεί το βήμα του Commit πατώντας το κουμπί Commit All. Αφού γίνει αυτό το βήμα, κάτω από την επιλογή του branch θα φαίνεται πόσα αρχεία είναι έτοιμα να γίνουν push από την ένδειξη N outgoing / N incoming. Όταν ο χρήστης είναι έτοιμος να πραγματοποιήσει το push θα το κάνει με την επιλογή του κουμπιού με το βέλος επάνω όπως φαίνεται στο Σχήμα 3.6.

Πολλές φορές κατά τον προγραμματισμό αλλάζουμε μια συνάρτηση για να υλοποιήσουμε νέα χαρακτηριστικά και να βελτιώσουμε το πρόγραμμα. Παρόλα αυτά, οι αλλαγές δεν βαίνουν πάντα όπως τις επιθυμούμε και πρέπει να γυρίσουμε σε προηγούμενη έκδοση του συγκεκριμένου κομματιού του κώδικα. Επειδή το GitHub είναι ένα εργαλείο ελέγχου έκδοσης κώδικα, μπορεί να μας βοηθήσει σε αυτή την λειτουργία με μόνο προαπαιτούμενο ότι όλες τις αλλαγές τις είχαμε κάνει Commit στο branch του.

Η όψη αρχείων από προηγούμενη έκδοση κώδικα μέσα από το Microsoft Visual Studio με την χρήση του GitHub είναι μια πολύ εύκολη διαδικασία. Αυτό μπορεί να γίνει από την επιλογή Git Repository. Για να γίνει θα πρέπει να το ανοίξει απο το View -> Git Repository ή με την χρήση της συντόμευσης Ctrl + 0, Ctrl + R.



Σχήμα 3.7: Ιστορικό Git Repository

Στο παράθυρο που θα εμφανιστεί στα αριστερά εμφανίζονται τα branches που έχουν δημιουργηθεί και στα δεξιά η ιστορία των αλλαγών. Επιλέγοντας ένα branch, στα δεξια ανανεώνεται η οθόνη με το ιστορικό όλων των αλλαγων των commit που έχουν γίνει. Με την επιλογή μιας εξ αυτών των γραμμων θα εμφανιστεί ένα νέο παράθυρο με τρια tab όπου το ένα tab θα εμφανίζει πως ήταν το αρχείο, το δεύτερο πως έγινε και το τρίτο όλα τα αρχεία που έγιναν commit, ενώ επιλέγοντας αυτό το αρχείο θα εμφανιστούν οι αντίστοιχες αλλαγές του αρχείου αυτού.

Αναλύοντας όλες αυτές τις δυνατότητες μπορούμε να διατυπώσουμε την άποψη ότι το GitHub μπορεί να αποτελέσει ένα πολύ δυνατό εργαλείο για κάθε προγραμματιστή αλλά και ομάδα προγραμματιστών.

Μέσα από την έκδοση ελέγχου του κώδικα δίνεται η δυνατότητα στους προγραμματιστές να ελέγχουν τις αλλαγές που έχουν κάνει, ενώ με την παράλληλη χρήση των branches είναι σε θέση να αναπτύσουν πολλά χαρακτηριστικά εν παραλλήλω χωρίς να επηρεάζονται οι διάφορες περιοχές που κάνει ο καθένας, ενώ στο τέλος της ανάπτυξης αυτών μπορούν να κάνουν merge και να ενώσουν τις αλλαγές που έκαναν χωρίς να υπάρχουν οι συνήθεις συγκρούσεις που ενδεχομένως να γινόντουσαν αν δεν υπήρχε η βοήθεια του Github.

Το GitHub αποτελείται από μια κοινότητα ανοιχτού κώδικα που μπορούν οι χρήστες να συνεισφέρουν στα διάφορα projects που υπάρχουν σε αυτό. Αυτό δίνει την δυνατότητα στους προγραμματιστές να δουν τον κώδικα άλλων και να πάρουν ιδέες και να βελτιώσουν τις γνώσεις τους. Επίσης μέσα από γνωριμίες που γίνονται απο την συνεισφορά σε τέτοια έργα ανοίγουν οι ορίζοντες των προγραμματιστών, διευρύνεται ο κύκλος του και γίνονται συζητήσεις για διάφορα θέματα που μπορούν να αποφέρουν γνώσεις οι οποίες θα βοηθήσουν σε νέα έργα ή προκλήσεις κώδικα που θα αναλάβει ο προγραμματιστής να υλοποιήσει.

Τέλος θα μπορούσε κάποιος να πει ότι το βιογραφικό κάποιου προγραμματιστή είναι ο κώδικας και τα εργαλεία που έχει φτιάξει, έτσι λοιπόν καταλήγουμε ότι το GitHub μπορεί να λειτουργήσει ως ένα βιογραφικό αφού εκεί ο προγραμματιστής μπορεί να επιδείξει όλες τις εφαρμογές που έχει φτιάξει περιγράφοντας τις μεθόδους και τις τεχνολογίες που έχει υλοποιήσει. Εκτός από αυτό στο GitHub φαίνεται η συχνότητα με την οποία κάνει commit ο προγραμματιστής ανα τις χρονικές περιόδου. Μέσα από αυτό μπορεί να φανεί η συνέπεια ενός προγραμματιστή αν υποθέσουμε ότι κάθε μέρα που δουλεύει χρησιμοποιεί το GitHub και εκτελεί τα αντίστοιχα commits.

3.5 Debugging

Η αποσφαλμάτωση του κώδικα είναι η μια σύνθετη διαδικασία που θα μπορούσαμε να πούμε ότι αποτελείται από τρεις επιμέρους διαδικασίες: την διαδικασία της αναγνώρισης, έπειτα της ανάλυσης και εν τέλη της διόρθωσης που προκαλεί το λάθος στον κώδικα. Αρχικά θα πρέπει ο προγραμματιστής να είναι σε θέση να αναπαράξει το πρόβλημα και έπειτα να το ταυτοποιήσει έτσι ώστε να γνωρίζει τι επιμέρους αλλαγές πρέπει να πραγματοποιήσει. Αφού κάνει αυτά τα δύο πράγματα έρχεται η επίλυση του κώδικα.

Το Microsoft Visual Studio παρέχει προηγμένα εργαλεία για να βοηθήσουν τον προγραμματιστή σε αυτήν την διαδικασία της αποσφαλμάτωσης του κώδικα. Για να υλοποιηθεί αυτή η διαδικασία ο χρήστης θα πρέπει να ανοίξει το αρχείου του κώδικα που επιθυμεί να αποσφαλματώσει και στην συνέχεια να κάνει δεξί κλικ αριστερότερα απο τον αριθμό της γραμμής του κώδικα και να εμφανιστεί μια κόκκινη βούλα. Όταν συμβεί αυτό ο χρήστης έχει θέσει ένα breakpoint στην Ν γραμμή του κώδικα και όταν τρέξει τον κώδικα σε Debug Mode θα σταματήσει ο κώδικας να εκτελείται σε όλες τις γραμμές που έχει τεθεί breakpoint.

Οι πιο βασικές λειτουργίες είναι το Step Into (F11) όπου μπαίνει μέσα στην συνάρτηση που είναι να εκτελεστεί, Step Over (F10) όπου πηγαίνει στην αμέσως επόμενη γραμμή κώδικα, Continue (F5) που μας πηγαίνει στο επόμενο breakpoint (αν υπάρχει) που θα βρεθεί κατά την εκτέλεση του κώδικα και τέλος το Quick Watch (Shift + F9) που με το άνοιγμα αυτού του παραθύρου μπορεί ο χρήστης να

γράψει την ονομασία μιας οποιοδήποτε μεταβλητής ή αντικειμένου του κώδικα και ο evaluator με την σειρά θα εμφανίσει την τιμή του ή τις ιδιότητες και τις αντίστοιχες τιμές του αν είναι αντικείμενο.

Ένα ακόμα χαρακτηριστικό που παρέχει το Microsoft Visual Studio και λείπει από μερικά IDE είναι ότι κατα την χρήση του Quick Watch καθώς ο χρήστης γράφει ενεργοποιείται το IntelliSense και εμφανίζει τις ιδιότητες των αντικειμένων, που είναι πολύ χρήσιμο διότι μερικά αντικείμενα έχουν πάρα πολλές ιδιότητες που είναι αδύνατον να θυμάται ο προγραμματιστής και ιδιαίτερα σε μεγάλα προγράμματα όπου υπάρχουν πάρα πολλά αντικείμενα.

Τέλος η προσθήκη των watches επιτρέπει στον προγραμματιστή να έχει μια γρήγορη σύνοψη με όλες τις τιμές των μεταβλητών που τον ενδιαφέρουν.

Επίσης πέρα από την αποσφαλμάτωση του κώδικα για την λύση των προβλημάτων που παρουσιάζονται στον κώδικα, γίνεται και αποσφαλμάτωση για την βελτίωση της χρήσης πόρων του υπολογιστή ώστε το κώδικα να γίνει πιο γρήγορα αλλά και πιο ελαφρύ στην χρήση του χωρίς να καταλαμβάνει πόρους από τον υπολογιστή που θα μπορούσαν να αποφευχθούν μέσα από την βελτίωση του κώδικα.

Αυτό επιτυγχάνεται μέσα από την παρακολούθηση των Diagnostic Tools που το παράθυρο του ανοίγει αυτόματα κατά την έναρξη του Debug Mode. Έτσι έχει την δυνατότητα ο χρήστης να δει πόση μνήμη καταλαμβάνει η διεργασία και να ανακαλύψει πιθανά memory leaks ή άλλα προβλήματα που σχετίζονται με την κατανομή της μνήμης. Επίσης μπορεί να δει και το ποσοστό χρήσης CPU που καταλαμβάνει η διεργασία που μπορεί επίσης να βοηθήσει να βρεθούν πιθανά προβλήματα ή συναρτήσεις που καταλαμβάνουν αρκετούς πόρους από την CPU.

Εκτός από αυτό μετά από κάθε χρήση του Step Over ενεργοποιείται το PerfTips που δείχνει πόσο χρόνος χρειάστηκε για να εκτελεστεί το προηγούμενο βήμα, που μπορεί να δώσει μια ιδέα για το πόσο επηρεάζει την εκτέλεση του κώδικα η γραμμή που μόλις εκτελέστηκε.

3.6 Επίλογος

Το Microsoft Visual Studio αποτελεί ένα σύγχρονο ολοκληρωμένο περιβάλλον ανάπτυξης κώδικα που μπορεί να βοηθήσει τον προγραμματιστή να επικεντρωθεί στην ανάπτυξη του κώδικα, την αποσφαλμάτωση του και τον έλεγχο έκδοσης του κώδικα ενώ βρίσκεται συνεχώς στο ίδιο περιβάλλον που θα του έχει γίνει οικείο. Υποστηρίζει 36 γλώσσες προγραμματισμού και το μεγάλο ποσοστό των προγραμματιστών μπορούν να το χρησιμοποιήσουν αφού είναι πολύ πιθανόν να υπάρχει η υποστήριξη της γλώσσας που χρησιμοποιούν.

Ως προς την υλοποίησης μιας διαδικασίας Scraping η επιλογή του IDE μπορεί να βοηθήσει ώστε μερικά πράγματα να γίνουν πιο απλά. Τα χαρακτηριστικά της αποσφαλμάτωσης που περιέχει το Visual Studio βάζουν τον προγραμματιστή σε πλεονεκτική θέση, αφου η χρήση του Quick Watch με την υποστήριξη του IntelliSense, μπορεί να δώσει μια εικόνα στο τι εντολές κώδικες μπορούν να χρησιμοποιηθούν. Πολλές φορές το scraping έχει να κάνει με την τροποποίηση των συμβολοσειρών ώστε να έρθουν στην κατάλληλη μορφή για την εισαγωγή τους στην βάση. Έτσι η δυνατότητα να θέσει ο προγραμματιστής breakpoint σε ένα συγκεκριμένο σημείο και με την χρήση του Quick Watch να δοκιμάζει εντολές ως προς το ποιά είναι η πιο κατάλληλη και να βλέπει ζωντανά το αποτέλεσμα, μπορεί να τον βοηθήσει να σώσει χρόνο και να υλοποιήσει την διαδικασία πιο γρήγορα.

Επίσης η ενσωμάτωση του Github τον βοηθάει να είναι στο ίδιο περιβάλλον όλο την χρονική περίοδο του προγραμματισμού και να συγκεντρωθεί πιο εύκολο, εκτός από αυτό, θα του υπενθυμίζεται για πιο συχνά commit με αποτέλεσμα να έχει καλύτερο ιστορικό όσον αφορά τον έλεγχο έκδοσης του κώδικα.

Περαιτέρω με την χρήση του NuGet μπορεί να ανακαλύψει βιβλιοθήκες που υλοποιούν διαδικασίες που διαφορετικά δεν θα ήξερε ότι υπάρχουν έτοιμες και θα τις υλοποιούσε από την αρχή

Παρατηρούμε λοιπόν ότι το Microsoft Visual Studio δίνει όλα τα εργαλεία που είναι απαραίτητα σε έναν προγραμματιστή και είναι ένα ίσως το πιο σύγχρονο και δωρεάν ολοκληρωμένο περιβάλλον ανάπτυξης κώδικα.

Κεφάλαιο 4ο: Το σύστημα Διαχείρισης Scraping

4.1 Εισαγωγή

Το σύστημα διαχείρισης Scraping που υλοποιήθηκε για τους λόγους της πτυχιακής αποτέλεσε μια προσπάθεια ώστε να υπάρχει καλύτερη επίγνωση για τα δεδομένα που γίνονται Scraping και εισέρχονται στην βάση δεδομένων. Η ιστοσελίδα από την οποία εξάγονται τα δεδομένα είναι το viva.gr. Αποτελεί μια ιστοσελίδα για ηλεκτρονικές κρατήσεις εισιτηρίων που οι κατηγορίες αυτών των εισιτηρίων είναι ακτοπλοϊκών, μουσικών, θεατρικών εκδηλώσεων κλπ. Η πτυχιακή αυτή ασχολείται με την εξαγωγή πληροφοριών που σχετίζονται με θεατρικές παραστάσεις. Οι πληροφορίες αυτές περιλαμβάνουν τίτλο παράστασης, περιγραφή, εικόνα παράστασης, συντελεστές και πληροφορίες εισιτηρίων.

Κατα την υλοποίηση της εφαρμογής και την μελέτη του αντικειμένου του Scraping δόθηκαν περεταίρω ιδέες για το πως αυτό το σύστημα θα μπορούσε να βελτιώσει γενικά την διαδικασία του Scraping πέρα από τις επιμέρους κλασικές ενέργειες που αφορούν το scraping.

Το σύστημα αυτό μπορεί να δουλέψει αρκετά καλά με την επίβλεψη του χρήστη όπου κατά την διενέργεια της διαδικασίας του scraping εμφανίζονται ζωντανά όλες οι πληροφορίες που αφορούν την εισαγωγή των παραστάσεων, ηθοποιών ρόλων κλπ. Έτσι ο χρήστης βλέπει ζωντανά τι εισήχθει και τι όχι στην βάση δεδομένων, μπορεί να τροποποιήσει τις πληροφορίες που εισάγονται (π.χ όνομα ηθοποιού ρόλου) αλλά και να διαγράψει περιεχόμενο που εισήχθει αλλά μπορεί να μην κρίθηκε σωστό.

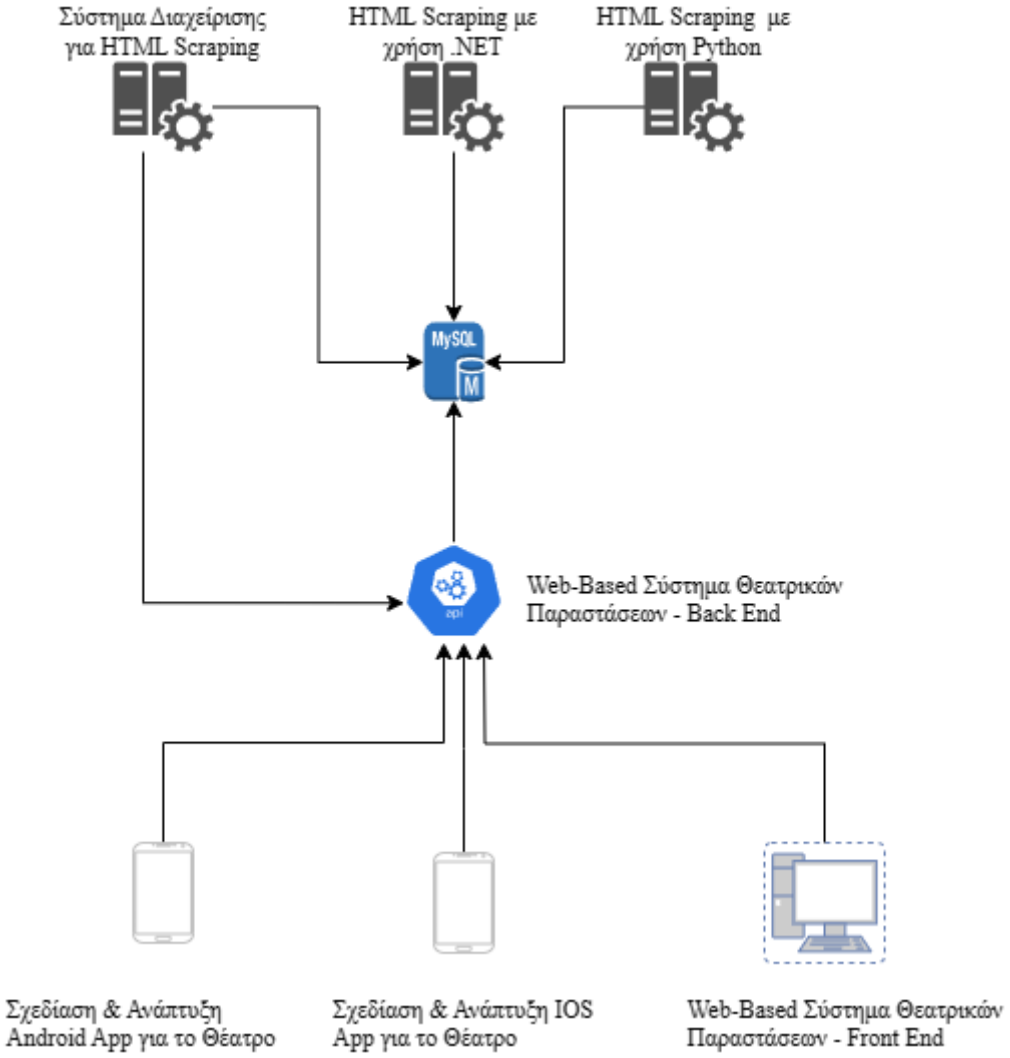
Αυτή η διαδικασία της τροποποίησης των εισαχθέντων δεδομένων φυσικά δεν είναι ανάγκη να γίνουν κατά την διάρκεια της εκτέλεσης του scraping αλλά και αφού γίνει διότι κρατιέται αναλυτικό καταγραφικό για όλες τις ενέργειες που έγιναν κατά την εκτέλεση της διαδικασίας του Scraping.

Ένα από τα μεγάλα προβλήματα που εμφανίζονται στις διαδικασίες του scraping είναι η αλλαγή της δομής της ιστοσελίδας (που αναλύθηκε στα τεχνικά χαρακτηριστικά στο κεφάλαιο 1.5.2) δηλαδή του DOM με αποτέλεσμα όλες οι επιλογές κόμβων με βάση στο οποίο στηρίζεται το scraping να μην δουλεύουν πια. Έτσι το αποτέλεσμα είναι η μη εύρεση των κατάλληλων κόμβων που καταλήγει στην μη εισαγωγή των δεδομένων στην βάση τους. Έτσι υλοποιήθηκε ένα σύστημα που αποθηκεύονται οι πληροφορίες επιλογής των κόμβων σε ένα αρχείο με σκοπό ο διαχειριστής του συστήματος να μπορεί να τους αλλάζει. Αυτό δίνει την δυνατότητα το σύστημα να μπορεί να είναι παραμετροποιήσιμο που σημαίνει ότι μπορεί να είναι δυναμικό. Έτσι αν η ιστοσελίδα viva.gr αποφασίσει να αλλάξει την κλάση του κόμβου που περιέχει την ονομασία της παράστασης ο διαχειριστής θα μπορεί να εντοπίσει ποιά είναι η σωστή κλάση του κόμβου και να εισάγει το αντίστοιχο νέο XPath, που αντιστοιχεί στον κόμβο και το σύστημα να χρησιμοποιεί τον νέο σωστό επιλογέα κόμβου. Με αυτή την υλοποίηση λοιπών λύνεται ένα από τα προβλήματα που αντιμετωπίζουν οι εφαρμογές scraping που είναι της αλλαγής της δομής του DOM της ιστοσελίδας.

Εκτός από αυτό ένα ακόμα από τα θέματα του scraping που αντιμετωπίζει το σύστημα διαχείρισης που υλοποιήθηκε είναι το θέμα της “ευγένειας” ως προς το πότε και πόσες αιτήσεις γίνονται προς την ιστοσελίδα. Ο διαχειριστής του συστήματος μπορεί να θέσει ποιες μέρες αλλα και ποιες ώρες θα πραγματοποιηθεί η διαδικασία του scraping. Εκτός από αυτό, υπάρχει και δυνατότητα ρύθμισης του αριθμού των παραστάσεων που θα εισαχθούν σε κάθε σύνοδο που πραγματοποιείται. Με αυτόν τον τρόπο υπάρχει έλεγχος του πλήθους των αιτήσεων που γίνονται που αποσκοπεί να μην επηρεάζεται η λειτουργία της ιστοσελίδας της κρίσιμες ώρες.

Τα δεδομένα αυτά εισέρχονται στην βάση δεδομένων μέσω του API που έχει υλοποιηθεί από τον Αριστείδη Τσαχλάρη και περιγράφεται στην πτυχιακή του (Web-Based Σύστημα Θεατρικών Παραστάσεων - Back End) είτε απευθείας με την εισαγωγή τους στην βάση που έγινε ως ένας προγενέστερος, αρχικός τρόπος. Ο τελικός σκοπός των δεδομένων είναι η επεξεργασία και η εμφάνιση τους και αυτό γίνεται από την αναπαράσταση τους μέσα από εφαρμογή web που υλοποιήθηκε από τον Δημήτριο Αναστασιάδη (Web-Based Σύστημα Θεατρικών Παραστάσεων - Front End), την εφαρμογή IOS του Ιωάννη Σκαρλή (Σχεδίαση & Ανάπτυξη IOS App για το Θέατρο) και την εφαρμογή Android της Αθηνάς Παπαχρήστου (Σχεδίαση & Ανάπτυξη Android App για το Θέατρο).

Η εφαρμογή μου βασίστηκε και αποτελεί συνέχεια της υλοποίησης του Scraping του Κωνσταντίνου Γεωργιάδη (HTML Scrapping με χρήση .NET). Επίσης scraping θεατρικών παραστάσεων έγινε και από τον Παναγιώτη Φωτίου (HTML Scrapping με χρήση Python). Να αναφερθεί ότι αυτές οι δύο πτυχιακές έγιναν σε προγενέστερο χρόνο και δεν είχε υλοποιηθεί τότε το API και κάνουν εισαγωγή των δεδομένων απευθείας στην βάση.

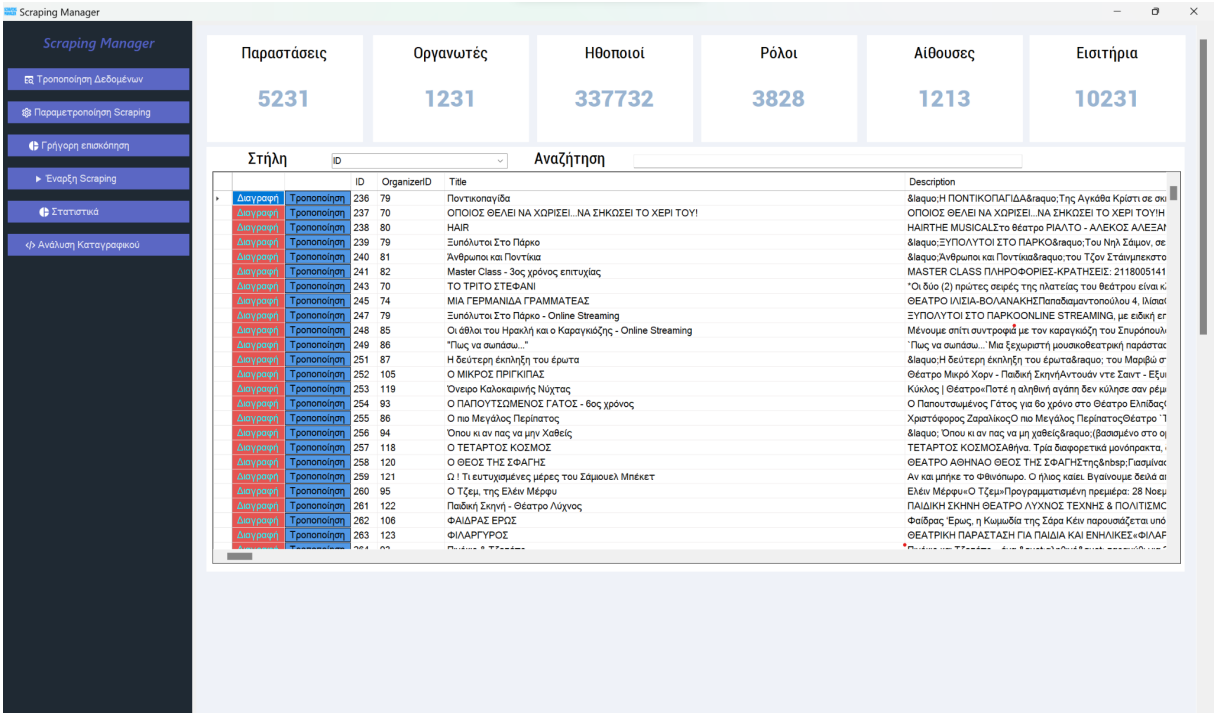


Σχήμα 4.1: Πτυχιακές που συμμετείχαν στο καλλιτεχνικό portal

Η εφαρμογή κάνει χρήση της γλώσσας προγραμματισμού C# με την βοήθεια των WPF για την υλοποίησης της διεπαφής του χρήστη και των βιβλιοθηκών HTML Agility Pack και Selenium που χρησιμοποιήθηκαν για την υλοποίηση του scraping.

4.2 Περιήγηση στην εφαρμογή

Η εφαρμογή αποτελείται από κάποιες βασικές οθόνες. Η αρχική οθόνη που ονομάζεται και τροποποίηση δεδομένων δίνει μια γρήγορη σύνοψη των δεδομένων που υπάρχουν δείχνοντας πόσες παραστάσεις, οργανωτές, ηθοποιοί, ρόλοι, αίθουσες και εισιτήρια υπάρχουν στην βάση δεδομένων. Με επιλογή ενός από τις παραπάνω επιλογές εμφανίζονται στον πίνακα (GridView) που υπάρχει δυναμικά τα δεδομένα του πίνακα που μας ενδιαφέρουν. Υπάρχει η δυνατοτητα για αναζήτηση με βάση της στήλης του πίνακα με επιλογή του column από το πεδίο Στήλη και έπειτα αναζήτηση του στοιχείου που ψάχνουμε με βάση το κείμενο από το πεδίο στήλη. Τέλος υπάρχει η δυνατότητα τροποποίησης των δεδομένων ή διαγραφής τους με επιλογή του κομβίου Διαγραφή/επεξεργασία που υπάρχει σε κάθε γραμμή του αντίστοιχου GridView.



Σχήμα 4.2: Οθόνη Τροποποίηση Δεδομένων

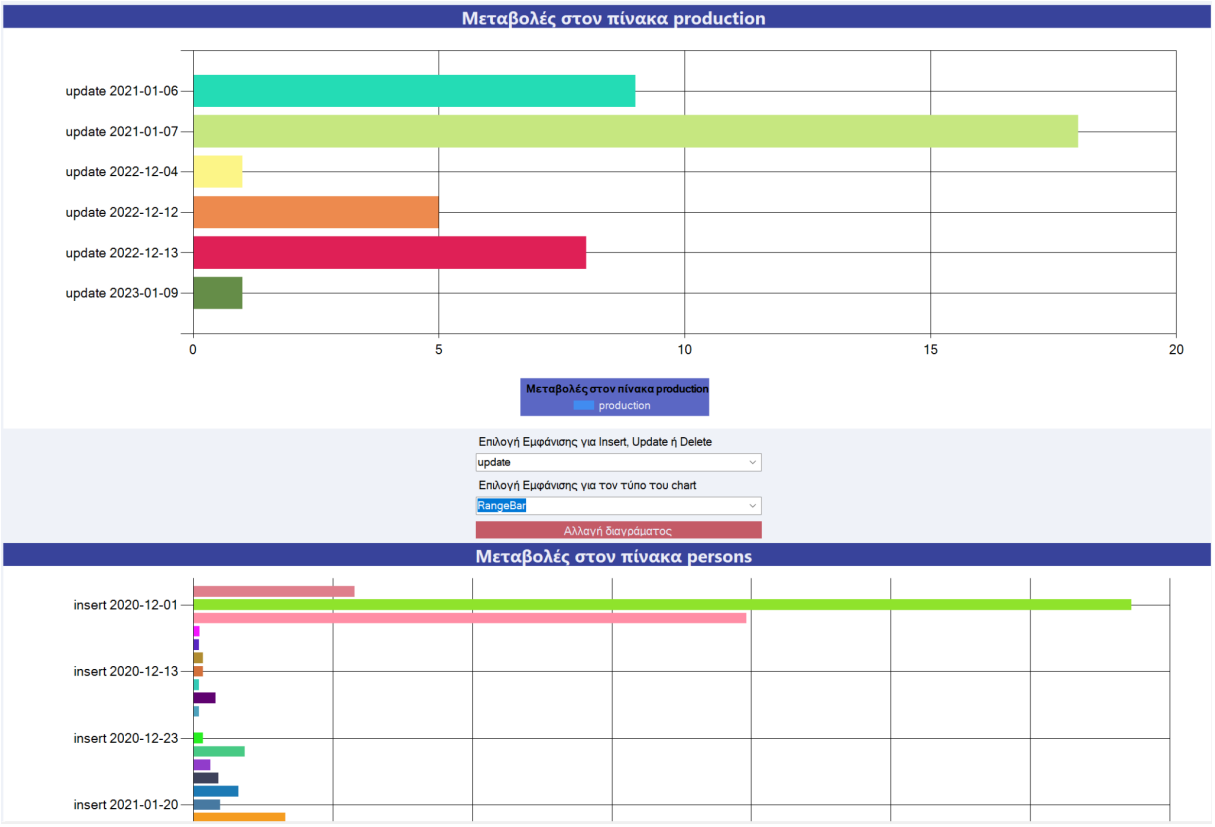
Αυτή η βασική οθόνη δίνει την δυνατότητα στον διαχειριστή του συστήματος να ελέγξει την ορθότητα των παραστάσεων και των επιμέρους δεδομένων που την αποτελούν και να κάνει τις αντίστοιχες ενέργειες της διόρθωσης των δεδομένων αν αυτός το κρίνει απαραίτητο. Είναι δηλαδή η οθόνη που μπορεί να επηρεάσει τις πληροφορίες που εμφανίζονται στις υπόλοιπες πτυχιακές που κάνουν την εμφάνιση των δεδομένων και την ανάλυση των δεδομένων μέσω της χρήσης του API από το οποίο αντλούν τα δεδομένα από την βάση.

Κατή την επιλογή του πίνακα που θέλουμε να εμφανίσουμε ενεργοποιείται η συνάρτηση labelPreviewClick(object sender, EventArgs e) όπου με βάση τον Sender παίρνουμε το Panel που αυτό περιέχει το όνομα του πίνακα στην ιδιότητα tag. Για παράδειγμα αν γίνει επιλογή στο text Παραστάσεις τότε το tag θα είναι production.

Αφού έχουμε το όνομα του πίνακα μπορούμε να παραμετροποιήσουμε το GridView που βρίσκεται ακριβώς από κάτω και αντιστοιχεί στην μεταβλητή dataGridViewMain. Αυτή η διαδικασία υλοποιείται στην συνάρτηση InitMainGridView(string tableName) όπου φτιάχνεται το sql string επιλογής του πίνακα δυναμικά και έπειτα καλείται η GenerateDataGridView(String sql,String tableName,bool generateDeleteButton) που δημιουργεί το DataGrid και το επιστρέφει στην συνάρτηση έτσι ώστε αυτό να εμφανιστεί.

Έπειτα υπάρχει η οθόνη Παραμετροποίηση Scraping που εκεί βρίσκεται όλη η λογική του συστήματος διαχείρισης scraping που υλοποιήθηκε και θα αναλυθεί σε επόμενο υποκεφάλαιο.

Η οθόνη Γρήγορη επισκόπηση αφορά τις ενέργειες που έγιναν στους πίνακες (insert,update,delete) με βάση τα περιεχόμενα του πίνακα changeLog και δίνει την δυνατότητα στον διαχειριστή να δει πότε έγιναν οι συγκεκριμένες ενέργειες και σε τι ποσότητα



Σχήμα 4.3: Οθόνη Γρήγορη Επισκόπηση

Ο διαχειριστής έχει την δυνατότητα για κάθε πίνακα να επιλέξει του τί είδος ενέργειες θέλει να εμφανιστεί μέσα από την επιλογή του πεδίου select (insert/update/delete) και επίσης του τι είδος επιθυμεί να είναι το διάγραμμα (διαθέσιμες επιλογές Area, Bar, BoxPlot Bubble, CandlestickColumn, Doughnut, ErrorBar, FastLine, FastPoint, Funnel, Kagi, Line, Pie, Point, PointAndFigure, Pola, Pyramid, Radar, Range, RangeBar, RangeColumn, Renko, Spline, SplineArea, SplineRange,StackedArea, StackedArea100, StackedBar, StackedColumn, StackedColumn100, StepLineStock, ThreeLineBreak).

Με αυτό τον τρόπο ο διαχειριστής έχει την δυνατότητα να δει τις ενέργειες που σχετίζονται με την βάση δεδομένων σε οποιοδήποτε διάγραμμα αυτός επιθυμεί.

Επόμενη οθόνη είναι η Έναρξη Scraping όπου δείχνει την προσομοίωση της διαδικασίας του Scraping με την δυνατότητα τροποποίησης των εισαχθέντων δεδομένων.

Η οθόνη Στατιστικά δίνει μια γενική εικόνα στον διαχειριστή ως προς τις παραστάσεις, τους ηθοποιούς κλπ για να μπορεί να έχει μια γενική ιδέα για τα στοιχεία των παραστάσεων.

Τέλος υπάρχει και η οθόνη Ανάλυση καταγραφικού όπου ο διαχειριστής μπορεί να βλέπει αναλυτικά όλες τις ενέργειες που έγιναν και δεν έγιναν κατά την διαδικασία του scraping.

Οι οθόνες Παραμετροποίηση Scraping, Έναρξη Scraping, Στατιστικά και ανάλυση καταγραφικού θα περιγραφούν αναλυτικά στα ακόλουθα υποκεφάλαια.

4.3 Το σχήμα της βάσης δεδομένων της εφαρμογής

Η MySQL είναι η πιο διάσημη ανοιχτού κώδικα SQL σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που έχει αναπτυχθεί, διανεμηθεί και υποστηρίζεται από την Oracle.

Χρησιμοποιείται κυρίως για την διαχείριση και την αποθήκευση των δεδομένων. Αποτελεί μια από τις πιο συνηθισμένες επιλογές των προγραμματιστών για την χρήση της ως βάση δεδομένων κατά την ανάπτυξη των εφαρμογών τους λόγω της ευκολίας χρήσης της, επεκτασιμότητα της αλλά και σταθερότητας της. Κάποια από τα χαρακτηριστικά της είναι:

- Σχεσιακή Βάση: Η MySQL ακολουθεί το σχεσιακό μοντέλο βάσεων δεδομένων, όπου τα δεδομένα οργανώνονται σε πίνακες που περιέχουν γραμμές και στήλες και αποτελεί μια συλλογή πληροφοριών που οργανώνει τα δεδομένα με προκαθορισμένες σχεσιακές σχέσεις μεταξύ τους που βοηθούν στην εύκολη πρόσβαση τους. Οι πίνακες, τα indexes και οι όψεις κρατιούνται σε ξεχωριστό μέρος από τα φυσικά δεδομένα με αποτέλεσμα οι διαχειριστές της βάσης να μπορούν να αλλάξουν τα φυσικά δεδομένα χωρίς να επηρεάσουν την λογική δομή των δεδομένων
- Αρχιτεκτονική πελάτη-διακομιστή: Η MySQL στον διακομιστή τρέχει ως μια ξεχωριστή διαδικασία και υποδέχεται και εκτελεί όλες τις διαδικασίες που σχετίζονται με την βάση. Ο πελάτης (ουσιαστικά δηλαδή η υλοποίηση της MySQL στην εκάστοτε γλώσσα προγραμματισμού) συνδέεται με τον διακομιστή και του στέλνει τις αιτήσεις για την εκτέλεση των ερωτημάτων
- Ανοιχτού Κώδικα: Η MySQL είναι ανοιχτού κώδικα λογισμικό, που σημαίνει ότι μπορεί να χρησιμοποιηθεί και να διανεμηθεί χωρίς χρέωση. Επίσης το γεγονός ότι ο κώδικας είναι ανοιχτός και μπορεί να τροποποιηθεί και από τους προγραμματιστές, κατέστησε την MySQL να έχει συνεχώς βελτιώσεις και να έχει μεγάλη σταθερότητα
- Ασφάλεια: Υποστήριξη χαρακτηριστικών ασφαλείας όπως αυθεντικοποίηση χρήστη που κάθε χρήστης μπορεί να έχει διαφορετικά δικαιώματα ως προς το τι εντολές SQL είναι σε θέση να εκτελέσει. Επίσης υποστηρίζει κρυπτογραφημένες συνδέσεις προς τον διακομιστή με αποτέλεσμα την προστασία των δεδομένων από τρίτους
- Περιορισμού ακεραιότητας: Η MySQL υποστηρίζει την χρήση των περιορισμών που αποτελούν μηχανισμό για τον έλεγχο της συνέπειας των δεδομένων και εξασφαλίζουν ότι η βάση δεν θα βρεθεί σε ασυνεπή κατάσταση με την χρήση των προτεύοντων και ξένων κλειδιών.
- Επεκτασιμότητα: Η MySQL έχει δημιουργηθεί ώστε να μπορεί να υποστηρίζει εφαρμογές που θα έχουν από μικρό εύρος δεδομένων ως ένα πολύ μεγάλο. Εφαρμογές όπως το GitHub, WePay, Spotify, Youtube, Twitter και Uber κάνουν χρήση της MySQL.

- Cross-Platform Συμβατότητα: Η MySQL μπορεί να εκτελεστεί σε διαφορετικά λειτουργικά συστήματα όπως Windows, Linux, macOS και διαφορετικές γλώσσες προγραμματισμού όπως C#, Java, Python, C++, Delphi κλπ.

Ο τύπος της βάσης δεδομένων που χρησιμοποιήθηκε κατά την υλοποίηση του θεατρικού συστήματος είναι MySQL και αποτελείται από 13 πίνακες και 32 indexes που φαίνονται στο παρακάτω σχήμα.

Ο πίνακας production περιέχει όλες τις λεπτομέρειες της θεατρικής παράστασης όπως όνομα, περιγραφή, σύνδεσμος της παράστασης στο viva.gr, όνομα διοργανωτή και έναν σύνδεσμο μιας εικόνας που χρησιμοποιείται ως εξώφυλλο.

Ο πίνακας roles περιέχει τα ονόματα των απο τους ρόλους των ηθοποιών, όπως το ίδιο και ο πίνακας persons που περιέχει τα ονόματα των ηθοποιών ενώ ο πίνακας images κρατάει πληροφορίες που έχουν τον σύνδεσμο URL όπου υπάρχει η εικόνα του ηθοποιού.

Ο πίνακας organizer εμπεριέχει όλες τις λεπτομέρειες του διοργανωτή της παράστασης όπως όνομα, διεύθυνση, πόλη, ταχυδρομικός κώδικας, τηλέφωνο, email, αριθμός ΔΟΥ και ΑΦΜ.

Ο πίνακας contributions αφορά όλους τους ανθρώπους, ρόλους που συμμετέχουν σε μια θεατρική παράσταση.

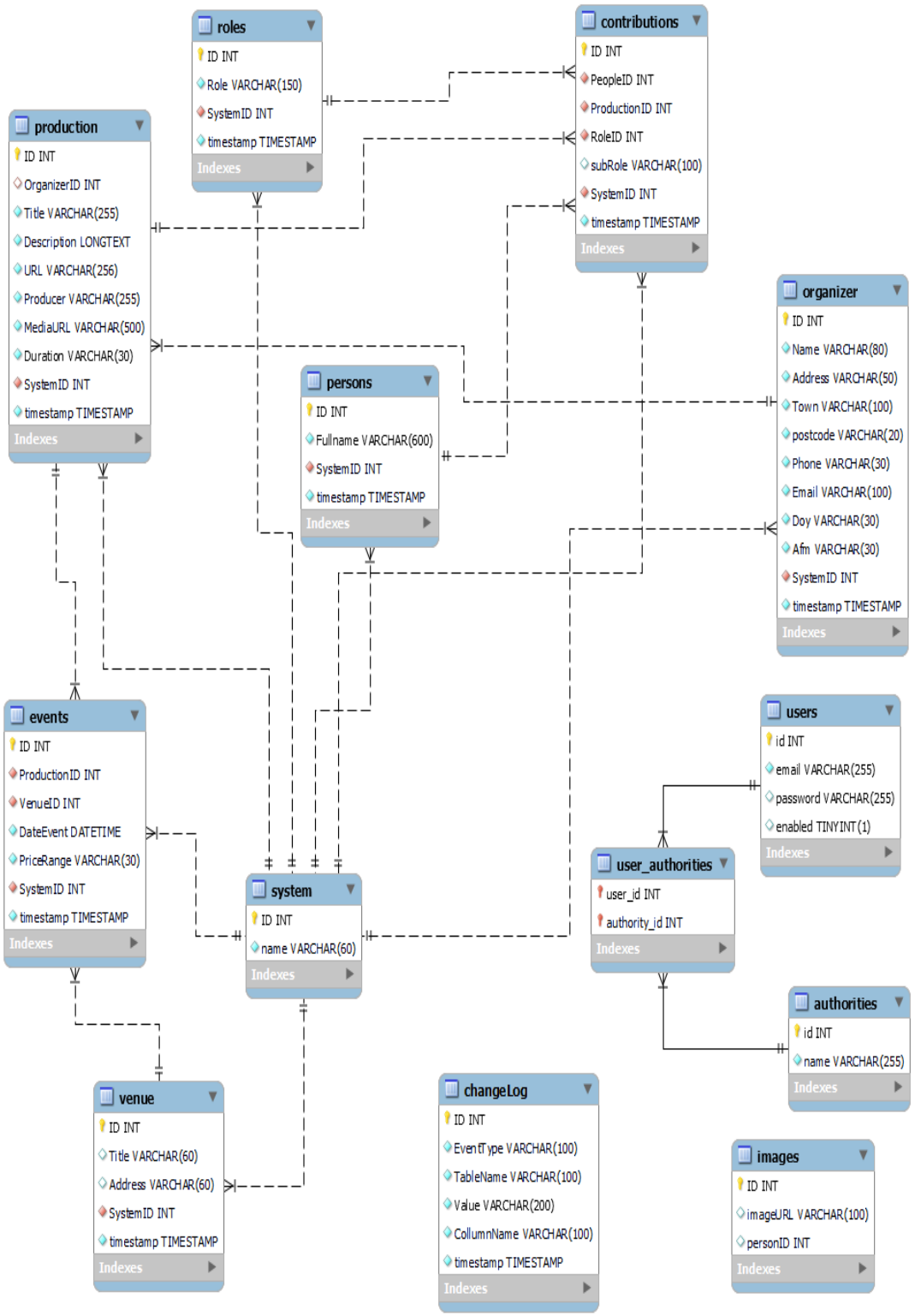
Οι πίνακες events και venues σχετίζονται με την ημερομηνία της διενέργειας της θεατρικής παράστασης, την τιμή του εισιτηρίου αλλά και σε ποιά αίθουσα θα προβληθούν.

Ο πίνακας System περιέχει πληροφορίες για το ποιά εφαρμογή έκανε το scraping, για παράδειγμα το SystemId για την python είναι 2 και για την C# (η εφαρμογή που περιγράφεται) είναι το 3.

Ο πίνακας changeLog έχει πληροφορίες που αφορούν ενέργειες όπως insert, update ή delete για τους ανάλογους πίνακες. Είναι ένας πολύ σημαντικός πίνακας που παρόμοιος πίνακας πρέπει να υπάρχει σε όλες τις βάσεις δεδομένων. Αυτός ο πίνακας μας δίνει μια γρήγορη και αναλυτική επισκόπηση που αφορά όλες τις ενέργειες που έγιναν στην βάση δεδομένων. Το πεδίο EventType αφορά την ενέργεια που διενεργήθηκε, το tableName το όνομα του πίνακα που τον αφορά, το πεδίο value την τιμή που ανατέθηκε/είχε/τροποποιήθηκε ανάλογα με την ενέργεια, το collumnName την στήλη που την αφορά και τέλος το timestamp την ημερομηνία που έγινε αυτή η ενέργεια.

Τέλος οι πίνακες users, user_authorities, authorities έχουν να κάνουν με την χρήση του API όπου λειτουργεί με το μοντελο JSON Web token το οποίο είναι ένα ασφαλές μοντέλο για την ασφαλή μεταφορά πληροφοριών μεταξύ απομακρυσμένων πόρων με την μορφή JSON Object. Ο πίνακας users έχει στήλες για το email και τον κωδικό του που αποθηκεύεται σε κρυπτογραφημένη μορφή ώστε να μην γνωρίζουν οι χρήστες που έχουν πρόσβαση στην βάση δεδομένων τον κωδικό του χρήστη. Επίσης περιέχει το πεδίο enabled που υποδεικνύει αν ο λογαριασμός είναι ενεργός. Ο πίνακας user_authorities περιέχει το επίπεδο των λειτουργιών που μπορεί να κάνει ο χρήστης του πίνακα users. Υπάρχουν άλλα δικαιώματα για απλό select των δεδομένων που χρησιμοποιούνται κυρίως από τις εφαρμογές που εμφανίζουν δεδομένα και δεν επιθυμούμε αυτές οι εφαρμογές να εκτελούν ενέργειες που εισάγουν δεδομένα στην βάση. Ενώ οι scrapers που χρησιμοποιούνται για την εξαγωγή των δεδομένων από την ιστοσελίδα πρέπει να έχουν δικαιώματα για την εισαγωγή των παραστάσεων, των ηθοποιών των ρόλων και λοιπά.

Παρατηρούμε λοιπόν ότι το API περιέχει διαβάθμιση των δικαιωμάτων των χρηστών ώστε να μην μπορεί να προκληθεί εισαγωγή ασαφών ή λάθος δεδομένων μέσω την χρήση του από εφαρμογές που δεν πρέπει να έχουν αυτά τα δικαιώματα.



Σχήμα 4.4: Διάγραμμα EER της βάσης δεδομένων

4.4 Η διαδικασία του Scraping των Παραστάσεων

Η παραπάνω διαδικασία έχει ως στόχο την εισαγωγή των παραστάσεων που δεν είναι στην βάση δεδομένων. Μια θεατρική παράσταση ως οντότητα αποτελείται από τον οργανωτή, τους ηθοποιούς και τους ρόλους τους αλλά και τις αίθουσες, την ώρα και ημερομηνία και την τιμή της θεατρικής παράστασης. Η διαδικασία αυτή υλοποιείται από την συνάρτηση insertProductions

Αρχικά σε προγραμματιστικούς όρους θα μπορούσαμε να εξισώσουμε μια θεατρική παράσταση ως έναν σύνδεσμο URL που αναφέρεται στην αυτήν και περιέχει όλες τις απαραίτητες λεπτομέρειες που χρειάζονται για την εισαγωγή της. Για αυτό τον λόγο αρχικά γίνεται η συλλογή όλων των θεατρικών παραστάσεων από το viva.gr όπου δεν έχουν εισαχθεί στην βάση δεδόμενων. Αυτό γίνεται μέσω του ελέγχου αν το URL της παράστασης υπάρχει στον πίνακα productions. Έτσι, υπάρχει μια λίστα με όλα τα διαθέσιμα URLs προς εισαγωγή.

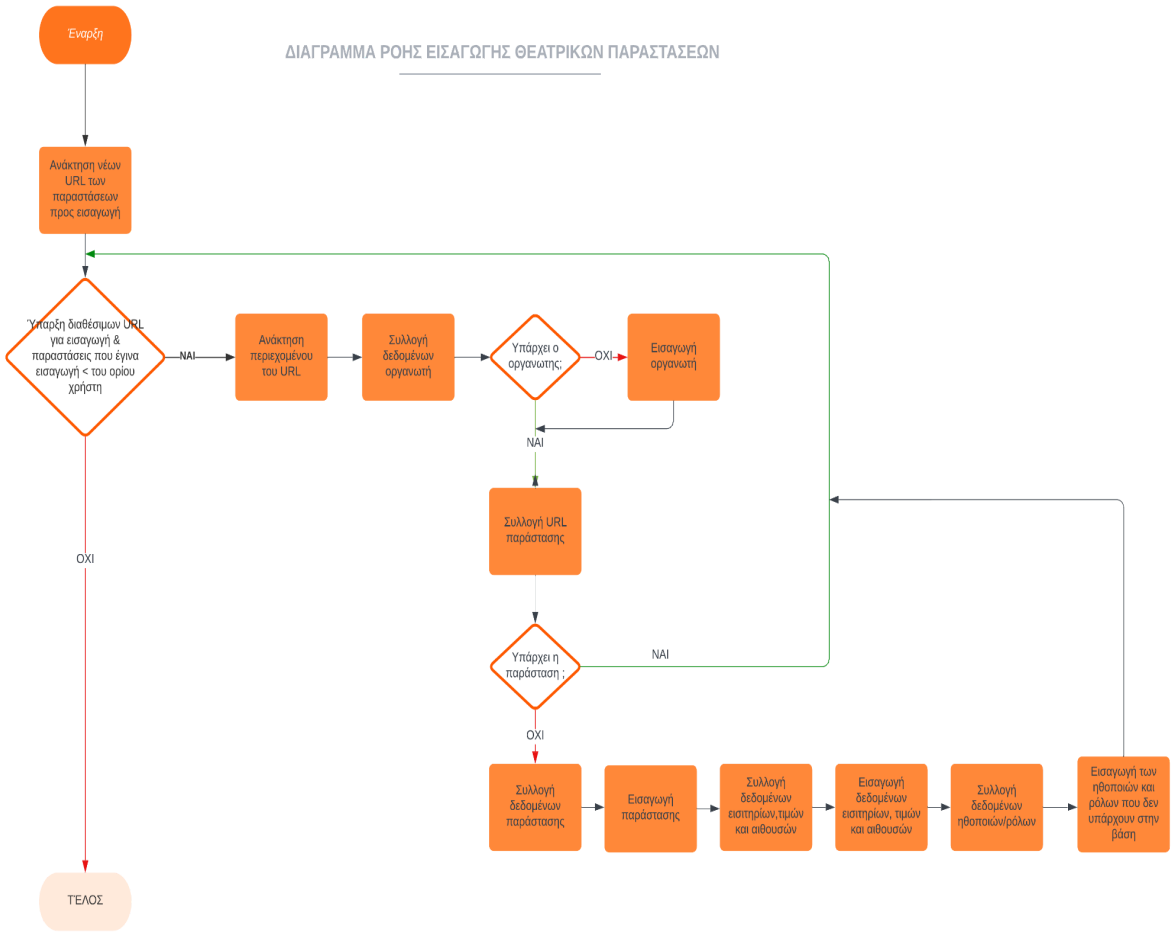
Για κάθε θεατρική παράσταση που δεν υπάρχει στην βάση, κάνουμε χρήση του Html Agility Pack ώστε να φορτώσουμε το περιεχόμενο του συνδέσμου σε ένα HtmlDocument. Αφού γίνει αυτό ουσιαστικά ξεκινάει η διαδικασία της εισαγωγής την παράστασης. Με την χρήση των επιλογέων XPath συλλέγουμε το όνομα του οργανωτή. Αν το όνομα του οργανωτή δεν υπάρχει στην βάση, τότε εξάγουμε όλες τις πληροφορίες του και στην συνέχεια εισάγουμε στην βάση τα στοιχεία του.

Να σημειωθεί ότι όλες οι εισαγωγές στην βάση γίνονται είτε μέσω της χρήσης του API είτε μέσω απευθείας σύνδεσης στην βάση, ανάλογα τι έχει επιλέξει ο διαχειριστής του scraping απο την σελίδα πραματεροποίηση scraping.

Ακολουθεί η συλλογή των χαρακτηριστικών της παράστασης όπως το όνομα της, η περιγραφή της, η εικόνας της και η διάρκεια της και γίνεται η εισαγωγή της. Για την εισαγωγή των εισιτηρίων η χρήση του Html Agility Pack δεν καθιστά δυνατή την συλλογή των απαραίτητων στοιχείων διότι η φόρτωση αυτών των στοιχείων στην σελίδα γίνεται δυναμικά μετά την αποδοχή των cookies και για τον λόγο αυτό χρησιμοποιείται η βιβλιοθήκη Selenium που επιτρέπει στον προγραμματιστή να εκτελεί ενέργειες που αφορούν την αλληλεπίδραση του με την ιστοσελίδα. Έτσι με το Selenium επιλέγουμε την αποδοχή των cookies και έπειτα συλλέγουμε το όνομα της αίθουσας, την ημερομηνία και την ώρα αλλά και την τιμή που αποτελούν την παράσταση. Τέλος εισάγονται τα δεδομένα στην βάση.

Τελευταίο βήμα είναι η εισαγωγή των ηθοποιών και των ρόλων της παράστασης. Ένα από τα θέματα που παρουσιάζεται σε αυτή την συλλογή των στοιχείων είναι ότι η σελίδα δεν έχει πάντα συνεπής δομή ως προς την εμφάνιση των συντελεστών. Σε μερικές παραστάσεις μια γραμμή αποτελεί τον ρόλο ακολουθούμενο από την ονομασία του συντελεστή, ενώ σε άλλες όλοι οι συντελεστές είναι γραμμένοι σε μια γραμμή, εκτός από αυτό παρουσιάστηκε και παράσταση όπου οι συντελεστές και οι ρόλοι τους ήταν σε μορφή πίνακα. Ένα από τα θέματα του scraping που αντιμετωπίζει ο προγραμματιστής είναι η εύρεση συνεπής δομή του DOM έτσι ώστε να γίνει σωστά η συλλογή των δεδομένων. Έτσι δώθηκε η καλύτερη δυνατή προσπάθεια για την επεξεργασία των συμβολοσειρών που περιέχουν τις λεπτομέρειες των ρόλων και των συντελεστών τους.

Τέλος αφού πραγματοποιηθούν τα βήματα της εισαγωγής της παράστασης, του οργανωτή, των συντελεστών και των ρόλων τους αλλα και τις πληροφορίες για τα εισιτήρια η διαδικασία του scraping περατώνεται όταν εξαντληθούν τα διαθέσιμα URL προς εισαγωγή ή όταν φτάσουμε στο όριο των εισαχθέντων παρασταστάσεων ανά σύνοδο που μπορεί να θέσει ο διαχειριστής όπως θα αναφερθεί στο επόμενο υποκεφάλαιο.



Σχήμα 4.5: Διάγραμμα ροής εισαγωγής θεατρικών παραστάσεων

Στο παραπάνω διάγραμμα γίνεται μια προσπάθεια απλοποίησης της αναπαράστασης της διαδικασίας της εισαγωγής των θεατρικών παραστάσεων σε διάγραμμα ροής. Το Microsoft Visual Studio μπορεί να μας δώσει στοιχεία για την κάθε υλοποιημένη συνάρτηση μέσω του Calculate Code Metrics που ενεργοποιείται από την επιλογή του μενού Analyze.

Όνομα μέτρησης	Τιμή
Maintainability Index	26
Cyclomatic Complexity	20
Class Coupling	33
Lines of Source Code	199
Lines of Executable Code	112

Πίνακας 4.1: Μετρήσεις για την συνάρτηση insertProductions

Η τιμή Maintainability Index κάνει μια προσπάθεια χαρακτηρισμού της συντηρησιμότητας της συνάρτησης και η τιμή της εξαρτάται από παράγοντες όπως μέγεθος κώδικα, γραμμές σχολίων , πολυπλοκότητας αλλά και coupling μεταξύ των κλάσεων. Η τιμή αυτή μπορεί να έχει τιμές από 0 μέχρι 100. Το Visual Studio χαρακτηρίζει την τιμή αυτή απο 0-9 ως κόκκινο , 10-19 κίτρινο, και 20-100 με πράσινο.

Η τιμή Cyclomatic Complexity μετράει την δομική πολυπλοκότητα του κώδικα και αναπαριστα όλες τις πιθανές ροές που μπορεί να έχει ο κώδικας ανάλογα με τις πιθανές ροές που σχετίζονται με τις πιθανές τιμες των μεταβλητών.

Η τιμή Class Coupling μετράει την σύζευξη σε μοναδικές κλάσεις ή το πόσο είναι εξαρτημένες από άλλες κλάσεις. Υψηλή τιμή αυτής της μέτρησης δείχνει δεν υπάρχει η απαραίτητη απομόνωση από άλλες συναρτήσεις κλάσεις και όσο μικρότερη είναι αυτή η τιμή τόσο πιο συντηρίσημη είναι.

Τέλος, η τιμή Lines of Source Code αναφέρεται στο σύνολο των γραμμών του κώδικα, συμπεριλαμβάνοντας και τα σχόλια αλλά και τις κενές γραμμές, ενώ η τιμή Lines of Executable Code αναφέρεται αποκλειστικά στις γραμμές που θα εκτελεστούν.

4.5 Παραμετροποίηση Scraping

Ένας από τους στόχους της παρούσας πτυχιακής ήταν η διαδικασία του Scraping να είναι όσο το δυνατόν πιο παραμετροποίησιμη γίνεται. Αυτό σημαίνει ο διαχειριστής να μπορεί να τροποποιεί τις τιμές των selectors του DOM που αντιμετωπίζει το φαινόμενο της αλλαγής της δομής της ιστοσελίδας. Για παράδειγμα στην ιστοσελίδα viva.gr ο κόμβος της ονομασίας της παράστασης είναι `<h1 id="playTitle" class="title-fix-font-size title-fix-font-size" itemprop="name"></h1>`

και η επιλογή του γίνεται με το XPath //h1[@id='playTitle'].

Όπως καταλαβαίνουμε αν αλλάξει το ID του κόμβου η ο τύπος του απο h1 σε h2 η επιλογή του κόμβου δεν θα γίνει με αποτέλεσμα όλη η διαδικασία του scraping να βγει εκτός και το πρόγραμμα να μην δουλεύει και ο διαχειριστής του συστήματος να επικοινωνήσει με τον προγραμματιστή ότι χρειάζεται η αλλαγή του κώδικα και να του στείλουν το νέο executable με τροποποιημένο τον κώδικα που περιέχει τον νέο σωστό XPath selector.

```
"production": {
   "title": "//h1[@id='playTitle']",
   "description": "//div[@itemprop='description']",
   "duration": "//li[@class='ui-duration']",
   "media": {
     "url-cookies": "//a[contains(@class,'cc-btn--accept')]",
     "openMedia": "openMedia",
     "imageMedia": "PageContent_PlayDetails_PlayImage",
     "imageMedia2": "//img[@id='PageContent_PlayDetails_PlayImage']"
   }
 }
```

Σχήμα 4.6: Παράδειγμα δομής αρχείου scrap-viva.json

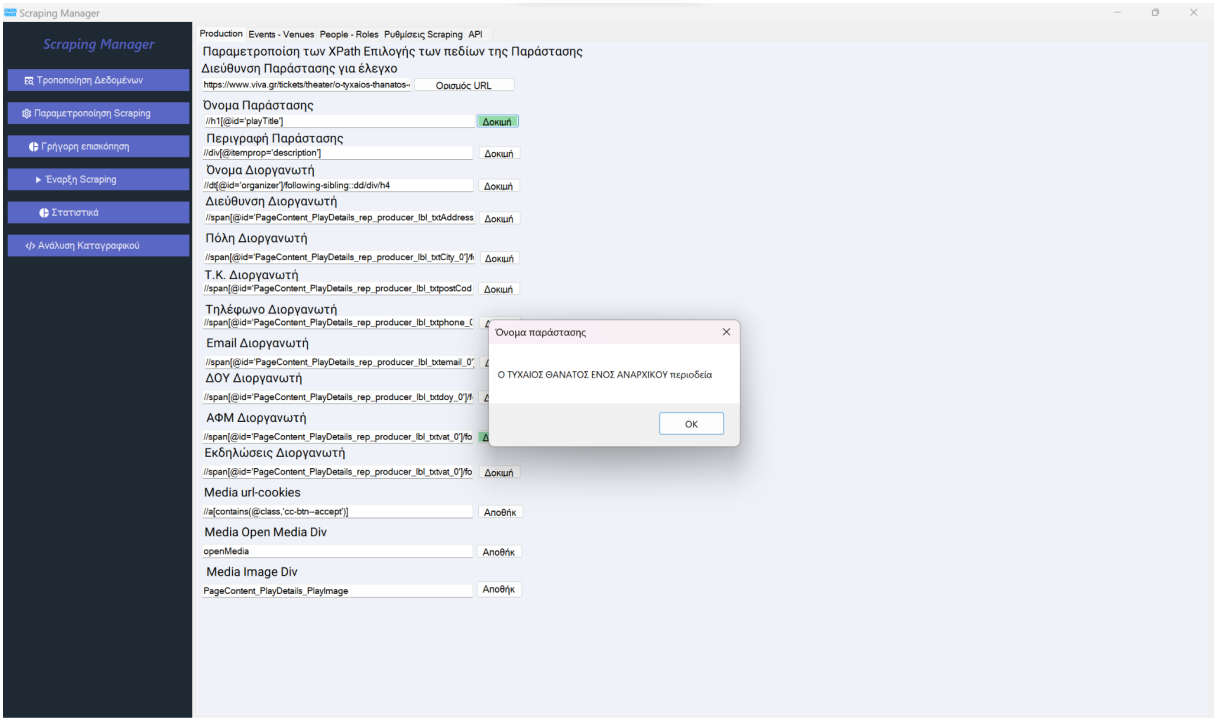
Αυτή η διαδικασία είναι χρονοβόρα και εκτός από αυτό απαιτεί και μεγαλύτερο κόστος συντήρησης. Με αυτόν τον τρόπο και εμπνεύστηκε η υλοποίηση της τροποποίησης του scraping. Αυτή η

υλοποίηση γίνεται με την βοήθεια ενός JSON αρχείο που ονομάζεται scrap-viva.json και βρίσκεται στον φάκελο Resources και με την χρήση ενός JObject με το όνομα jsonObject. Ο τύπος JObject ανήκει στην βιβλιοθήκη Newtonsoft.Json και αναπαριστά ένα αντικείμενο JSON με βάση το οποίο δίνεται η δυνατότητα στον χρήστη να έχει πρόσβαση και να αλλάζει τις τιμές των ιδιοτήτων του.

Αυτό το αντικείμενο μας δίνει μεγάλες δυνατότητες ώστε να κρατάμε τοπικά ένα αρχείο της μορφής JSON αλλάζοντας τις τιμές του και τέλος αποθηκεύοντας το. Κατά την έναρξη της εφαρμογής τρέχει η συνάρτηση LoadJSONToMemory και με αυτό τον τρόπο φορτώνει το αρχείο στην μεταβλητή jsonObject, ενώ αν το αρχείο δεν υπάρχει δημιουργεί ένα αρχείο με προκαθορισμένες τιμές.

Δίνεται η δυνατότητα δηλαδή κατά την εκτέλεση της insertProductions που είδαμε νωρίτερα να έχουμε πρόσβαση στις τιμές του αντικειμένου, για παράδειγμα τον τίτλο της παράστασης θα τον παίρναμε με την εκτέλεση της εντολής doc?.DocumentNode?.SelectSingleNode((string)MyGlobals.jsonObject["production"]["title"])

Επίσης υπάρχει η δυνατότητα ο χρήστης να ελέγχει ένα συγκεκριμένο σύνδεσμο παράστασης και βάζει το XPath και να βλέπει αν του επιστρέφει αποτέλεσμα. Όπως φαίνεται στο σχήμα 4.7 ο χρήστης εισάγει τον σύνδεσμο και να ελέγξει το αποτέλεσμα που έρχεται με βάση το XPath. Με την επιλογή του κουμπιού Ορισμός URL φορτώνεται το περιεχόμενο της σελίδας και μέσα από κάθε πεδίο όπως όνομα παράστασης, περιγραφή παράστασης ο χρήστης κάνει χρήση των πεδίων και επιλέγοντας το αντίστοιχο κουμπί δοκιμή του εμφανίζεται το αποτέλεσμα, έπειτα υπάρχει η δυνατότητα αποθήκευσης της επιλογής και με αυτό τον τρόπο το XPath που δόθηκε θα είναι το προκαθορισμένο κατά την υλοποίηση της διαδικασίας του scraping για την επιλογή του αντίστοιχου κόμβου.



Σχήμα 4.7: Οθόνη παραμετροποίησης scraping για την θεατρική παράσταση

Αν η τιμή του XPath φέρει αποτέλεσμα τότε το αντίστοιχο κουμπί που βρίσκεται δίπλα γίνεται πράσινο, ενώ αν δεν επιστραφεί τιμή που σημαίνει ότι το XPath είναι λάθος τότε το κουμπί θα γίνει κόκκινο, δίνοντας μια αίσθηση περαιτέρω ανάδρασης του συστήματος με τον χρήστη.

4.5.1 Παραμετροποίηση συνόδων

Όπως αναφέρθηκε σε προηγούμενα κεφάλαια τα προγράμματα που υλοποιούν το scraping θα πρέπει να εκτελούν τις αιτήσεις προς τον διακομιστή της ιστοσελίδας με φειδώ, ως ένδειξη σεβασμού προς την ομαλή λειτουργία της ιστοσελίδας, προκειμένου να μην υπερ φορτώσουν τον διακομιστή και να τελούν πολλές αιτήσεις και κυρίως σε ώρες αιχμής.

Η παραμετροποίηση των συνόδων του συστήματος έρχεται για να αντιμετωπίσει αυτό το πρόβλημας, της συμπεριφοράς του scraping με “ευγένεια” προς την ιστοσελίδα. Απο την οθόνη Παραμετροποίηση Scraping και επιλέγοντας το tab Ρυθμίσεις Scraping δίνονται δυνατότητες ρύθμισης του συστήματος από τον διαχειριστή.

Στο πεδίο μέρες που θα γίνεται το Scraping ο διαχειριστής έχει την δυνατότητα να επιλέξει τις μέρες που θα εκκινείται η διαδικασία. Για παράδειγμα μπορεί να θέλει να γίνεται scraping μονο μια φορά την εβδομάδα γιατί η ιστοσελίδα μπορεί να γίνεται ανανέωση μόνο μια φορά. Εκτός από αυτό μπορεί να επιλέξει να γίνεται scraping μόνο τα σαββατοκύριακα. Η τιμή 0 είναι η μέρα Κυριακή, η τιμή 1 είναι η Δευτέρα και ούτω καθεξής

Στο πεδίο ώρες ορίζεται οι ώρες που γίνεται το Scraping, η τιμή 1 σημαίνει 01:00 ή 1 π.μ. Αξίζει να σημειωθεί ότι καλό θα είναι το scraping να γίνεται ώρες μη αιχμής όπως μετά τα μεσάνυχτα, αφου δεν θα επηρεαστεί η ομαλή λειτουργία της ιστοσελίδας δεδομένου ότι οι άνθρωποι εκείνες τις ώρες κοιμούνται και δεν θα ψάχνουν για κρατήσεις εισιτηρίων θεατρικών παραστάσεων.

Τα πεδία διάλειμμα μεταξύ συνόδων scraping και παραστάσεις που γίνονται scrap ανά σύνοδο θα μπορούσαμε να πούμε ότι πηγαίνουν μαζί. Το πρώτο αναφέρεται στο χρονικό διάστημα που θα κάνει το σύστημα να κάνει τον έλεγχο αν υπάρχει το δικαίωμα να ξαναξεκινήσει το scraping. Δηλαδή μια τιμή 20, που σημαίνει 20 λεπτά, δείχνει ότι το διάστημα της επομένης συνόδου θα μπορεί να είναι 20 λεπτά μετα την περάτωση της πρώτης συνόδου. Αυτό φυσικά συμπεριλαμβάνει το γεγονός ότι βρισκόμαστε μέσα στα χρονικά διαστήματα μέρας και ώρας που περιγράφηκαν νωρίτερα. Το πεδίο παραστάσεις που γίνονται scrap ανά σύνοδο δείχνει πόσες παραστάσεις θα προσπαθήσει να εισάγει το σύστημα κάθε φορά που θα εκτελείται η διαδικασία scraping. Αυτό το πεδίο είναι σχετικά κρίσιμο διότι αν δοθεί μεγάλος αριθμός θα εκτελούνται πολλές αιτήσεις προς την ιστοσελίδα σε μικρό χρονικό διάστημα με αποτέλεσμα την κατανάλωση πόρων από τον διακομιστή της ιστοσελίδας.

Το πεδίο διαδρομή φακέλου chrome driver και data log είναι η διαδρομή του φακέλου που κρατούνται τα αρχεία chromedriver.exe και log.json. Το αρχείο chromedriver.exe πρέπει να ενημερώνεται από τον διαχειριστή ώστε να μπορεί να εκτελεστεί η διαδικασία του scraping. Το αρχείο log.json χρησιμοποιείται για τις ανάγκες του καταγραφικού και θα περιγραφεί στο κεφάλαιο 5.

Στο πεδίο connection string εισάγεται η τιμή του. Αυτή η τιμή θα χρησιμοποιηθεί αν ο διαχειριστής επιθυμεί να παρακάμψει την χρήση του API (π.χ. η λειτουργία του είναι εκτός) και πρέπει να γίνει η εισαγωγή απευθείας στην βάση δεδομένων. Τέλος, το SystemId είναι το ID από το τον πίνακα System που περιγράφηκε νωρίτερα.

Στο tab API έχουμε τα πεδία που σχετίζονται με την χρήση του. Το πρώτο checkbox αν το επιλέξει ο διαχειριστής θα γίνεται χρήση του της εισαγωγής των δεδομένων μέσα από την χρήση του API και θα παρακάμπτεται η τιμή του Connection String.

Η διεύθυνση API είναι ο σύνδεσμος μέσα στον οποίο συμπεριλαμβάνονται όλα τα επιμέρους URL για την εισαγωγή των ηθοποιών, της παράστασης, των ρόλων κ.λπ.

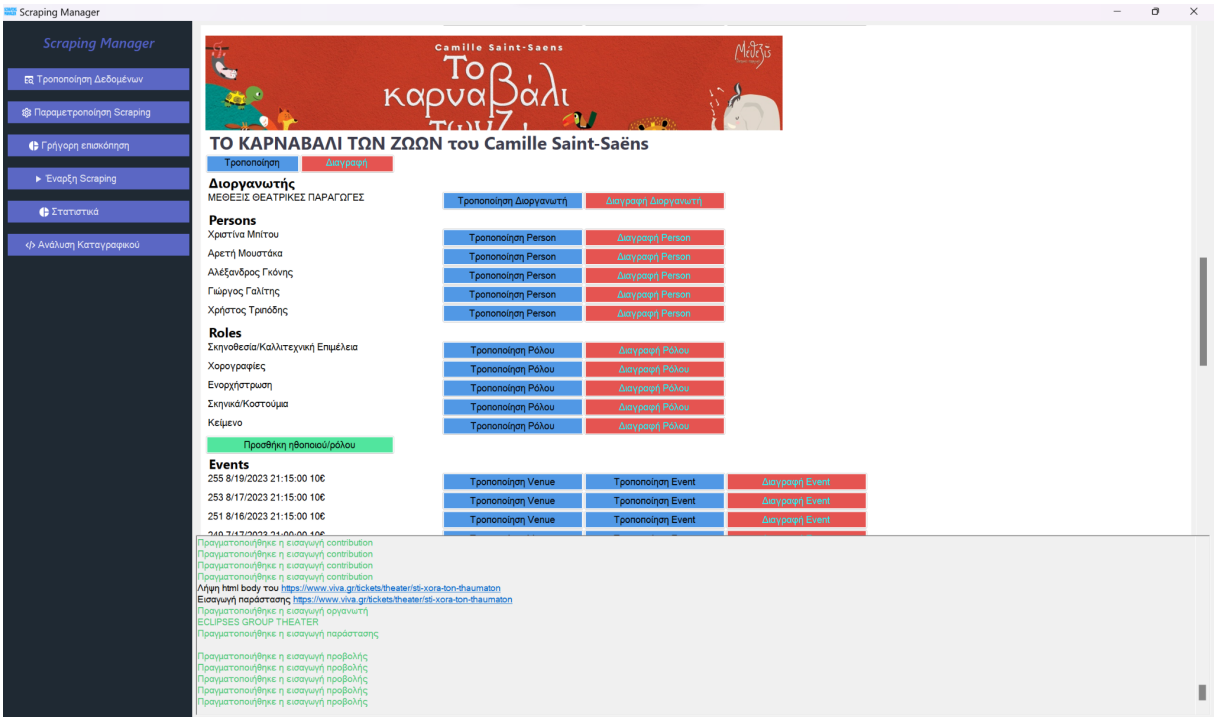
Τέλος τα πεδία όνομα χρήστη και κωδικός είναι του χρήστη με βάση του οποίου θα πραγματοποιουνται οι αιτήσεις για την εισαγωγή των δεδομένων. Να σημειωθεί ότι αυτός ο χρήστης θα πρέπει να έχει δικαιώματα POST και όχι μόνο GET, γιατί αλλιώς δεν θα υπάρχει η δυνατότητα χρήσης των κατάλληλων αιτήσεων που εισάγουν τα δεδομένα.

Μέσα από όλες αυτές τις ρυθμίσεις παρατηρούμε ότι ο διαχειριστής του συστήματος μπορεί να τροποποιήσει τις λειτουργίες εκτέλεσης του scraping και να επηρεάσει την απόδοση και την σωστή λειτουργία του συστήματος, πράγμα που μπορεί να δώσει ορθός τον όρο σύστημα διαχείρισης scraping.

4.6 Επικύρωση των δεδομένων που έγιναν Scraping κατά την σύνοδο

Ένα από τα πλεονεκτήματα του συστήματος είναι ότι κατά την διαδικασία του scraping ο χρήστης μπορεί να κάνει τροποποίηση των δεδομένων που έγινα scrap ή να διαγράφει στοιχεία που εισήχθησαν στην βάση και αυτά δεν θα έπρεπε να εισαχθούν, έτσι υπάρχει η δυνατότητα “επίβλεψης” του scraping. Με αυτόν τον τρόπο ο διαχειριστής μπορεί να επέμβει και διορθώσει τα τυχόν λάθη που έχουν πραγματοποιηθεί.

Υπάρχει δυνατότητα για τροποποίηση των στοιχείων της παράστασης, του οργανωτή, των ηθοποιών και των ρόλων τους και των εισιτηρίων αν ο διαχειριστής του συστήματος κρίνει ότι αυτά τα στοιχεία δεν είναι σωστά.



Σχήμα 4.8: Οθόνη Έναρξη Scraping

Με επιλογή του κουμπιού Έναρξη Scraping θα ξεκινήσει η σύνοδος του scraping σύμφωνα με τα στοιχεία που έχουν δοθεί από την οθόνη Παραμετροποίηση Scraping και σύμφωνα με το διάγραμμα ροής του σχήματος 4.5. Θα ξεκινήσει δηλαδή η διαδικασία του scraping για όσες θεατρικές παραστάσεις δεν είναι στην βάση δεδομένων και θα επεξεργαστούν και θα εισαχθούν από αυτές όσες έχουν οριστεί από τον διαχειριστή του συστήματος με βάση το πεδίο Παραστάσεις που γίνονται scrap ανά σύνοδο.

Όσο εισάγονται παραστάσεις μέσα από την διαδικασία, τόσο και θα εμφανίζονται στοιχεία που αφορούν τις παραστάσεις στην οθόνη. Για κάθε θεατρική παράσταση έχουμε την εικόνα της και απο κάτω στοιχεία για τροποποίηση των δεδομένων της παράστασης. Για κάθε επιλογή όπως θεατρική παράσταση, οργανωτής, ηθοποιοί, ρόλοι και πληροφορίες εισιτηρίων και αίθουσας υπάρχει η δυνατότητα επεξεργασίας των δεδομένων ή διαγραφής τους.

Αν ο χρήστης επιλέξει τροποποίηση τότε θα του εμφανιστεί ένα παράθυρο (όπως φαίνεται στο παρακάτω σχήμα) που έχει όλες τις πληροφορίες για την παράσταση μαζί με τις τιμές του και θα μπορεί να γίνει τροποποίηση των τιμών εκείνη την χρονική στιγμή.

Σχήμα 4.9: Παράθυρο τροποποίησης παράστασης

Με αυτόν τον τρόπο ο διαχειριστής του συστήματος είναι σε θέση να κάνει αλλαγές αν παρατηρήσει ότι κάποια από τα δεδομένα που συλλέχθηκαν και εισήχθησαν στην βάση δεν είναι έγκυρα.

Για να εμφανιστεί το παραπάνω popup παράθυρο για οποιαδήποτε τροποποίηση όπως παράσταση, οργανωτη, ρόλου κ.λπ. ενεργοποιείται η συνάρτηση ModifyButton_Click και γίνεται δυναμικά χωρίς προκαθορισμένες τιμές. Μέσα από αυτή την συνάρτηση κατα την δημιουργία του αντίστοιχου κουμπιού, στην ιδιότητα name του έχει αποδοθεί η τιμή id + “%^*%$#” + tableName. Με αυτόν τον τρόπο ανακτώντας την ιδιότητα name έχουμε το μοναδικό id της εγγραφής και τον πίνακα. Επειδή από τους πίνακες τα πρωτεύοντα κλειδιά τους έχουν όνομα στήλη ID μπορούμε πολύ εύκολα να επιλέξουμε την γραμμή τους με το παρακατω SQL ερώτημα SELECT * FROM tableName WHERE id = id.

Αφού έχουμε όλες τις τιμές θα χρειαστούμε και όλα τα στοιχεία του πίνακα. Αυτό μπορεί να γίνει εκτελώντας ενα SQL ερώτημα του τύπου DESCRIBE tableName και μέσα από αυτό θα πάρουμε το όνομα της στήλης (που αντιστοιχεί στο prompt του πεδίου) και τον τύπο της στήλης του πίνακα (που αντιστοιχεί στον τύπο του input πεδίου). Η υλοποίηση του παραπάνω αντιστοιχεί στην συνάρτηση GetTableFields(tableName,id);

Τώρα που έχουμε τα στοιχεία που αναφέρουν τί τύπου είναι ένα πεδίο, την ονομασία του και την τιμή του, είμαστε σε θέση να τα εμφανίσουμε στο popup modal. Για κάθε στοιχείο από τα παραπάνω γίνεται δημιουργία ενός TextBox και ενός Label με DockStyle.Top και βάζοντας ως πατέρα το Panel είναι δυνατό να εμφανιστεί το παράθυρο του σχήματος 4.9

4.7 Επίλογος

Παρατηρούμε ότι ένα πρόγραμμα scraping για να μπορεί να θεωρηθεί σύστημα διαχείρισης θα πρέπει να βοηθάει αυτόν που το διαχειρίζεται να ελέγχει την ορθότητα του συστήματος και την ροή της διαδικασίας του scraping. Αυτό επιτυγχάνεται σε μεγάλο βαθμό από τον έλεγχο του scraping κατά την διαδικασία της εκτέλεσης του βοηθώντας τον διαχειριστή να βλέπει ακριβώς και σε ζωντανό χρόνο τις ενέργειες που γίνονται όσο αναφορά την εισαγωγή των στοιχείων στην βάση.

Η ύπαρξη οθονών για την αναζήτηση, την τροποποίηση αλλά και διαγραφή εγγραφών από τους διάφορους πίνακες των θεατρικών παραστάσεων, ρόλων, ηθοποιών κ.λπ. είναι κάτι πολύ απλό προς την υλοποίηση του, όμως οι δυνατότητες που δίνει στον διαχειριστή είναι αρκετά μεγάλα αφού μπορεί να συμβάλει και αυτός κάνοντας έναν καθαρισμό δεδομένων ανά τακτά χρονικά διαστήματα.

Με αυτόν τον τρόπο ο διαχειριστής μπορεί να έχει έναν καλύτερο έλεγχο των δεδομένων που συμβάλλει σε καλύτερη ποιότητα των δεδομένων που εισάγονται στην βάση.

Εκτός από αυτό η παραμετροποίηση των ιδιοτήτων του scraping μπορεί να δώσει μακρά ζωή στην λειτουργία του συστήματος αφού αν αλλάξει σε κάποιο βαθμό η δομή της ιστοσελίδας, ο διαχειριστής θα μπορεί να βρει ποιά στοιχεία δεν εισάγονται σωστά, να βρει το νέο XPath και να το εισάγει ως προκαθορισμένη τιμή με αποτέλεσμα να μην χρειάζεται αλλαγή πηγαίου κώδικα αλλά μόνο αλλαγή ρυθμίσεων του συστήματος.

Τέλος η ύπαρξη καταγραφικού συστήματος που θα περιγραφεί στο αμέσως επόμενο κεφάλαιο είναι ζωτικής σημασίας για τον έλεγχο της ορθής λειτουργίας του scraping.

Όλα αυτά θα μπορούσαμε να πούμε ότι καθιστούν το πρόγραμμα που υλοποιήθηκε ένα σύστημα διαχείρισης για HTML Scraping

Κεφάλαιο 5ο:Καταγραφικό του Συστήματος Διαχείρισης Scraping

5.1 Εισαγωγή

Η ύπαρξη ενος καταγραφικού συστήματος για οποιαδήποτε εφαρμογή αποτελεί έναν ζωτικής σημασίας ρόλο για την ανάγνωση των σφαλμάτων και προβλημάτων που παρουσιάζονται στις εφαρμογές. Το αρχικά στάδια εκμάθησης προγραμματισμού αποτελούνται από την εμφάνιση απλών μεταβλητών στην οθόνη του υπολογιστή και καθώς προχωράνε τα στάδια εκμάθησης, ο μελλοντικός προγραμματιστής εμφανίζει όλο και πιο πολύπλοκες μεταβλητές που οι τιμές ανατίθενται από όλο και πιο σύνθετες επεξεργασίες.

Καθώς το πρόγραμμα γίνεται όλο και συνθετότερο η ανάγκη της ύπαρξης κάποιου μέρους να εμφανίζονται μεταβλητές και ενέργειες και διάφορα άλλα χαρακτηριστικά για ενέργειες που πραγματοποιήθηκαν κρίνεται αναγκαία. Με αυτό τον τρόπο ο προγραμματιστής μπορεί να βρίσκει τα λάθη και ίσως αν τα δεδομένα που βλέπει του το επιτρέπουν να βρει και λάθος από το οποίο προκλήθηκε.

Για τον λόγο αυτό πολλές εφαρμογές υλοποιούν ένα αρχείο καταγραφής των ενεργειών του συστήματος. Το καταγραφικό αυτό που έχει υλοποιηθεί για τις ανάγκες αυτού του συστήματος παίζει έναν σημαντικό ρόλο για την αποτελεσματικότητα του scraping. Πρώτα από όλα βοηθά τον διαχειριστή του συστήματος να βλέπει αναλυτικά όλες τις ενέργειες που έκανε το σύστημα κατά την διάρκεια του scraping των παραστάσεων. Αυτές οι ενέργειες αναφέρουν ποιον πίνακα και ποία τιμή του αφορούν και το αποτέλεσμα της ενέργειας που εκτελέστηκε.

Έτσι λοιπόν το καταγραφικό μπορεί να βοηθήσει τον διαχειριστή του συστήματος να έχει την γενική εικόνα για την ροή της εκτέλεσης του scraping και να βρίσκει αν κάποια εγγραφή στην βάση δεδομένων δεν έγινε και τον λόγο για τον οποίο δεν πραγματοποιήθηκε. Με αυτόν τον τρόπο του δίνει την δυνατότητα να επέμβει και να εντοπίσει τυχόν λάθη και αν του το επιτρέπουν η δυνατότητες της παραμετροποίησης του scraping να τα διορθώσει.

5.2 Δομή του συστήματος του καταγραφικού

Το καταγραφικό αποτελείται από ένα JSON αρχείο (log.json) που είναι αποθηκευμένο στον υπολογιστή του συστήματος του scraping και εκει πέρα αποθηκεύονται όλες οι ενέργειες του καταγραφικού.

Κατα την διάρκεια της εκτέλεσης της διαδικασίας του scraping καταγράφονται οι ενέργειες στο παραπάνω αρχείο για τα διάφορα συμβάντα που πραγματοποιούνται. Αυτές οι καταγραφές έχουν προστεθεί σε όλες τις επιμέρους συναρτήσεις που υλοποιούν τις εισαγωγές στην βάση δεδομένων αλλά και καταγραφές που αφορούν τα επιμέρους στοιχεία της επιλογής των κόμβων που επιτυγχάνουν την συλλογή των δεδομένων από το περιεχόμενο της ιστοσελίδας.

Για να πραγματοποιηθεί αυτή η εγγραφή των στοιχείων υλοποιήθηκε η συνάρτηση

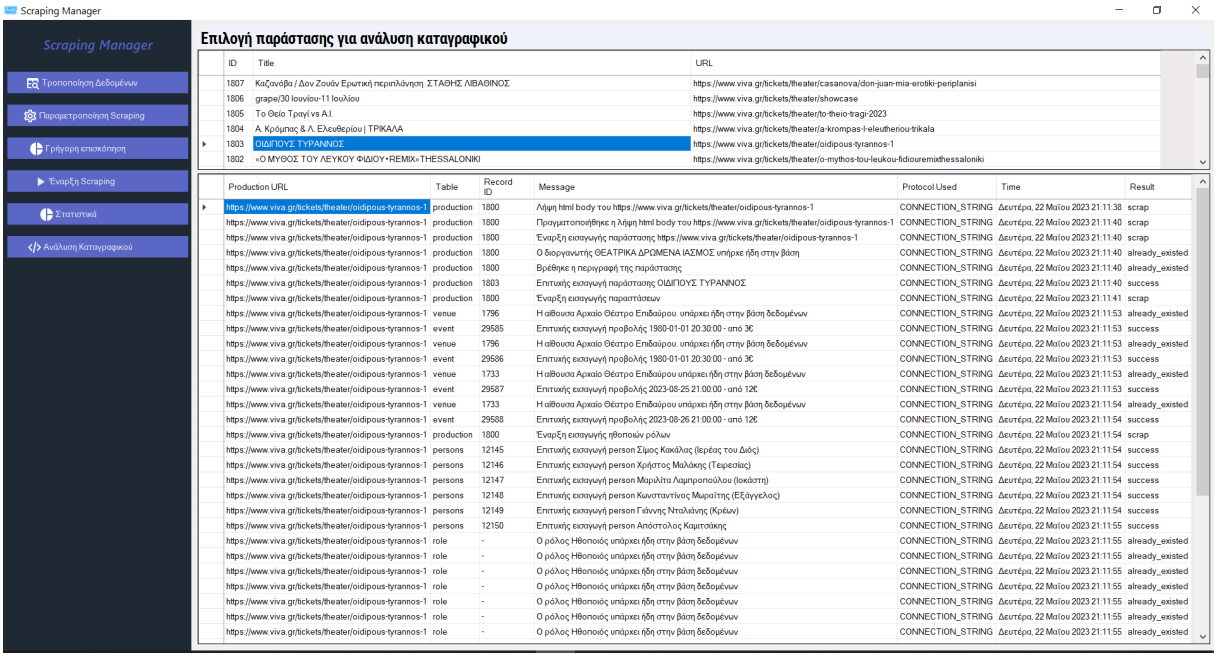
```
AppendToLog(string prodid, string table,string recordid,string
message,string protocol,string time,string result)
```

Οι παράμετροι που δέχεται αφορούν τον πίνακα για τον οποίο πρόκειται να γίνει η εγγραφή, το id της εγγραφής αν αυτή πραγματοποιήθηκε, το μήνυμα που θα εμφανιστεί, η χρονική στιγμή την οποία έγινε το συγκεκριμένο γεγονός και τέλος το αποτέλεσμα της ενέργειας.


```
    "url":
"https://www.viva.gr/tickets/theater/nea-epitheorisi-theatro-alsos-kalokairi-2023",
    "table": "venue",
    "recordid": "1887",
    "message": "Επιτυχής εισαγωγή Αίθουσας Θέατρο Άλσος",
    "protocol": "CONNECTION_STRING",
    "time": "Δευτέρα, 22 Μαΐου 2023 12:26:49",
    "result": "success"
```


Σχήμα 5.1: Παράδειγμα δομής αρχείου log.json

Η λογική της συνάρτησης AppendToLog είναι να δημιουργεί ένα αντικείμενο από τις παραπάνω παραμέτρους και προσθέτει το παραπάνω αντικείμενο στο αρχείο log.json



Σχήμα 5.2: Οθόνη Ανάλυσης Καταγραφικού

Όπως παρατηρούμε απο την παραπάνω εικόνα η οθόνη ανάλυσης καταγραφικού αποτελείται απο δύο GridView. Το πάνω εμφανίζει όλες τις παραστάσεις που υπάρχουν στην βάση δεδομένων και το απο κάτω εμφανίζει πληροφορίες που έχουν εγγραφεί στο καταγραφικό, όταν γίνει επιλογή της παράστασης από το πάνω GridView (ενεργοποίηση της συνάρτησης DataGridTop_CellClick).

Η λογική της προαναφερθείσας συνάρτησης είναι να πάρει την τιμή του url από τον κορυφαίο πίνακα, να κάνει αναζήτηση του αρχείου log.json και να εμφανίζει τα αποτελέσματα που αφορούν μόνο την συγκεκριμένη παράσταση.

Η αναζήτηση γίνεται με βάση το URL αφού αυτό είναι μοναδικό και με βάση αυτό μπορούν να ανακτηθούν όλες οι πληροφορίες που σχετίζονται μόνο με την συγκεκριμένη παράσταση.

5.3 Ανίχνευση των σφαλμάτων

Production URL	Table	Record ID	Message	Protocol Used	Time	Result
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	production	1800	Λήψη html body του https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:38	scrap
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	production	1800	Πραγματοποιήθηκε η λήψη html body του https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:40	scrap
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	production	1800	Έναρξη εισαγωγής παράστασης https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:40	scrap
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	production	1800	Ο διοργανωτής ΘΕΑΤΡΙΚΑ ΔΡΩΜΕΝΑ ΙΑΣΜΟΣ υπήρχε ήδη στην βάση	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:40	already_existed
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	production	1800	Βρέθηκε η περιγραφή της παράστασης	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:40	already_existed
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	production	1803	Επιτυχής εισαγωγή παράστασης ΟΙΔΙΠΟΥΣ ΤΥΡΑΝΝΟΣ	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:40	success
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	production	1800	Έναρξη εισαγωγής παραστάσεων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:41	scrap
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	venue	1796	Η αίθουσα Αρχαίο Θέατρο Επιδαύρου. υπάρχει ήδη στην βάση δεδομένων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:53	already_existed
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	event	29585	Επιτυχής εισαγωγή προβολής 1980-01-01 20:30:00 - από 3€	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:53	success
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	venue	1796	Η αίθουσα Αρχαίο Θέατρο Επιδαύρου. υπάρχει ήδη στην βάση δεδομένων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:53	already_existed
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	event	29586	Επιτυχής εισαγωγή προβολής 1980-01-01 20:30:00 - από 3€	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:53	success
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	venue	1733	Η αίθουσα Αρχαίο Θέατρο Επιδαύρου υπάρχει ήδη στην βάση δεδομένων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:53	already_existed
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	event	29587	Επιτυχής εισαγωγή προβολής 2023-08-25 21:00:00 - από 12€	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:53	success
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	venue	1733	Η αίθουσα Αρχαίο Θέατρο Επιδαύρου υπάρχει ήδη στην βάση δεδομένων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:54	already_existed
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	event	29588	Επιτυχής εισαγωγή προβολής 2023-08-26 21:00:00 - από 12€	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:54	success
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	production	1800	Έναρξη εισαγωγής ηθοποιών ρόλων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:54	scrap
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	persons	12145	Επιτυχής εισαγωγή person Σίμος Κακάλας (Ιερέας του Διός)	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:54	success
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	persons	12146	Επιτυχής εισαγωγή person Χρήστος Μαλάκης (Τειρεσίας)	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:54	success
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	persons	12147	Επιτυχής εισαγωγή person Μαριλίτα Λαμπροπούλου (Ιοκάστη)	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:54	success
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	persons	12148	Επιτυχής εισαγωγή person Κωνσταντίνος Μωραΐτης (Εξάγγελος)	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:54	success
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	persons	12149	Επιτυχής εισαγωγή person Γιάννης Νταλιάνης (Κρέων)	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:54	success
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	persons	12150	Επιτυχής εισαγωγή person Απόστολος Καμιτσάκης	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:55	success
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	role	-	Ο ρόλος Ηθοποιός υπάρχει ήδη στην βάση δεδομένων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:55	already_existed
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	role	-	Ο ρόλος Ηθοποιός υπάρχει ήδη στην βάση δεδομένων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:55	already_existed
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	role	-	Ο ρόλος Ηθοποιός υπάρχει ήδη στην βάση δεδομένων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:55	already_existed
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	role	-	Ο ρόλος Ηθοποιός υπάρχει ήδη στην βάση δεδομένων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:55	already_existed
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	role	-	Ο ρόλος Ηθοποιός υπάρχει ήδη στην βάση δεδομένων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:55	already_existed
https://www.viva.gr/tickets/theater/oidipous-tyrannos-1	role	-	Ο ρόλος Ηθοποιός υπάρχει ήδη στην βάση δεδομένων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:11:55	already_existed

Σχήμα 5.3: Ανίχνευση σφαλμάτων καταγραφικού (1ο παράδειγμα)

Όπως παρατηρούμε στην παραπάνω εικόνα υπάρχει αναλυτικό καταγραφικό για τις ενέργειες του συστήματος. Αρχικά βλέπουμε ότι για την παράσταση “ΟΙΔΙΠΟΥΣ ΤΥΡΑΝΝΟΣ” η έναρξη του scraping της ξεκίνησε την ημερομηνία Δευτέρα 22 Μαΐου 2023 21:11:36 με την λήψη του περιεχομένου της ιστοσελίδας και 4 δευτερόλεπτα αργότερα ολοκληρώθηκε το κατέβασμα του περιεχομένου.

Κατά την διάρκεια του ίδιου δευτερολέπτου ελέγχθηκε η ύπαρξη του διοργανωτή ΘΕΑΤΡΙΚΑ ΔΡΩΜΕΝΑ ΙΑΣΜΟΣ και βρέθηκε η ύπαρξη του οπότε δεν χρειάστηκε να συλλεχθούν τα δεδομένα του και να εισαχθεί στην βάση. Αυτό γίνεται εμφανές από το μήνυμα “Ο διοργανωτής ΘΕΑΤΡΙΚΑ ΔΡΩΜΕΝΑ ΙΑΣΜΟΣ υπήρχε ήδη στην βάση” και από το Αποτέλεσμα “already_existed”.

Στην διάρκεια των δευτερολέπτων 41 - 53 γινόταν συλλογή των δεδομένων της παράστασης και τέλος πραγματοποιήθηκε η εισαγωγή της παράστασης. Στην συνέχεια ακολουθεί η εισαγωγή των αιθουσών που θα γίνει η παράσταση και των εισιτηρίων τους. Βλέπουμε ότι οι παραστάσεις γίνονται στο Αρχαίο Θέατρο Επιδαύρου και επειδή υπάρχει ήδη στο venue δεν χρειάζεται να γίνει εισαγωγή στην βάση, μήνυμα “already_existed”.

Στην συνέχεια ακολουθούν σημαντικα ευρήματα για τον διαχειριστή του συστήματος. Όπως παρατηρούμε γίνεται εισαγωγή των ηθοποιων Σίμος Κακάλας (Ιερέας του Διός), Χρήστος Μαλάκης (Τειρεσίας), Μαριλίτα Λαμπροπούλου (Ιοκάστη), Κωνσταντίνος Μωραΐτης (Εξάγγελος), Γιάννης Νταλιάνης (Κρέων), Απόστολος Καμιτσάκη. Απο τα παραπάνω ονόματα μπορούμε να καταλήξουμε ότι δεν είναι πλήρως σωστά αφου σε πέντε εξ αυτών μέσα σε παρένθεση είναι ρόλος του ήρωα που υποδύεται ο ηθοποιός Γιάννης Νταλιάνης και δεν θα έπρεπε να υπάρχει. Αυτό δεν είναι εύκολο να εντοπιστεί από το σύστημα διαγράφοντας την παρένθεση, διότι θα μπορούσε κάλλιστα να είναι η ονομασία Γιάννης Νταλιάνης (Ιωάννης).

Για τον λόγο αυτό ο διαχειριστής του συστήματος, μπορεί μέσα από την εφαρμογή να πάει στην αρχική οθόνη και να ανοίξει το DataGridView των ηθοποιών να βρει τους παραπάνω ηθοποιούς και να διορθώσει τα ονόματα τους βγάζοντας την παρένθεση.

Production URL	Table	Record ID	Message	Protocol Used	Time
https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	production	1804	Λήψη html body του https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:12:29
https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	production	1804	Πραγματοποιήθηκε η λήψη html body του https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:12:29
https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	production	1804	Έναρξη εισαγωγής παράστασης https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:12:29
https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	production	1804	Ο διοργανωτής NG.ART.PRODUCTIONS E.E υπήρχε ήδη στην βάση	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:12:29
https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	production	1804	Βρέθηκε η περιγραφή της παράστασης	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:12:29
https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	production	1804	Επιτυχής εισαγωγή παράστασης Α. Κρόμπας & Λ. Ελευθερίου | ΤΡΙΚΑΛΑ	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:12:30
https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	production	1804	Έναρξη εισαγωγής παραστάσεων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:12:30
https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	venue	754	Η αίθουσα Πνευματικό Κέντρο Τρικάλων υπάρχει ήδη στην βάση δεδομένων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:12:46
https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	event	29589	Επιτυχής εισαγωγή προβολής 2023-05-23 21:30:00 - 13€	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:12:46
https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	production	1804	Έναρξη εισαγωγής ηθοποιών ρόλων	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:12:46
https://www.viva.gr/tickets/theater/a-krompas-l-eleutheriou-trikala	production	1804	Πραγματοποιήθηκε η εισαγωγή παράστασης	CONNECTION_STRING	Δευτέρα, 22 Μαΐου 2023 21:12:47

Σχήμα 5.4: Ανίχνευση σφαλμάτων καταγραφικού (2ο παράδειγμα)

Απο την παραπάνω εικόνα βλέπουμε ότι ενώ εισήχθηκε κανονικά η παράσταση, ο οργανωτής ήδη υπήρχε στην βάση και η προβολή και αυτή με την σειρά της έγινε εισαγωγή στην βάση δεν βρέθηκαν ηθοποιοί.

Αυτό γίνεται διότι δεν υπήρχε το tab της ιστοσελίδας με τους συντελεστές και επομένως η δομή της δεν ήταν επαρκής για να επιλέξει τους κατάλληλους κόμβους το σύστημα και να τους εισάγει στην βάση. Αυτοί οι δύο ηθοποιοί όμως υπάρχουν μέσα στην περιγραφή της παράστασης, επομένως ο διαχειριστής του συστήματος θα μπορούσε να τους προσθέσει και να τους συνδέσει με την θεατρική παράσταση.

Παρατηρούμε λοιπόν ότι το καταγραφικό σε αυτές τις δύο περιπτώσεις που περιγράφηκαν είχε αρκετή πληροφορία και σωστά δομημένη έτσι ώστε ο διαχειριστής του συστήματος να βρεί να τα λάθη και να επέμβει ώστε να υπάρχει καλύτερη συνέπεια των δεδομένων που ανήκουν στην βάση.

5.4 Επίλογος

Καταλήγοντας, είδαμε δύο περιπτώσεις όπου μέσα από το διάβασμα του καταγραφικού μπόρεσε να εξαχθεί το συμπέρασμα ότι το scraping δεν ήταν απόλυτο σωστό ως αποτέλεσμα της μη σωστής δομής της ιστοσελίδας. Παρόλα αυτά το σύστημα δίνει την δυνατότητα της διόρθωσης αυτών των θεατρικών παραστάσεων που δεν έχουν συνεπή δομή στον σύνδεσμο URL τους. Έτσι ο διαχειριστής μετά από τον έλεγχο και την παρέμβαση του μπορεί να φέρει τα δεδομένα της παράστασης σε πιο συνεπή μορφή.

Με αυτό τον τρόπο μπορούμε να πούμε με βεβαιότητα ότι η ανάλυση του καταγραφικού από τον διαχειριστή του συστήματος μπορεί να οδηγήσει σε εύρεση ασαφειών στα δεδομένα και την υποστήριξη των διορθώσεων τους.

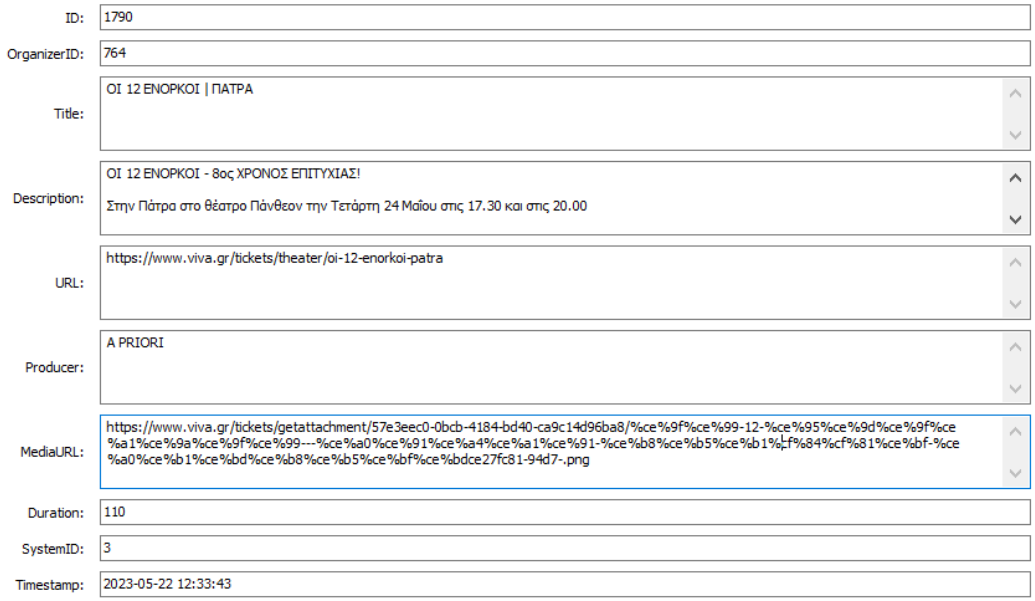
Κεφάλαιο 6ο: Εξέταση αποτελεσμάτων

Τα αποτελέσματα ενός συστήματος scraping είναι τα δεδομένα που συλλέγονται από την ιστοσελίδα και στην συνέχεια εισάγονται στο σύστημα. Τα δεδομένα αυτά πρέπει να έχουν συνεπή μορφή, να μην έχουν πολλά κενά διαστήματα και να μην περιέχουν ακατάλληλους χαρακτήρες όπως html entities που ήταν ένα από τα φαινόμενα που παρατηρήθηκε και διορθώθηκε από την προηγούμενη υλοποίηση του scraping.

Παρακάτω θα εξεταστεί μια παράσταση που έγινε scrap. Θα συγκρίνουμε τα αποτελέσματα ως προς του πως εμφανίζονται στο viva.gr και πως εισήχθησαν αυτά στην βάση δεδομένων. Η παράσταση που θα ελεγχθεί είναι η "Οι 12 Ένορκοι" και η ιστοσελίδα της παράστασης στο viva.gr μας δείχνει τα παρακάτω ως προς την παράσταση

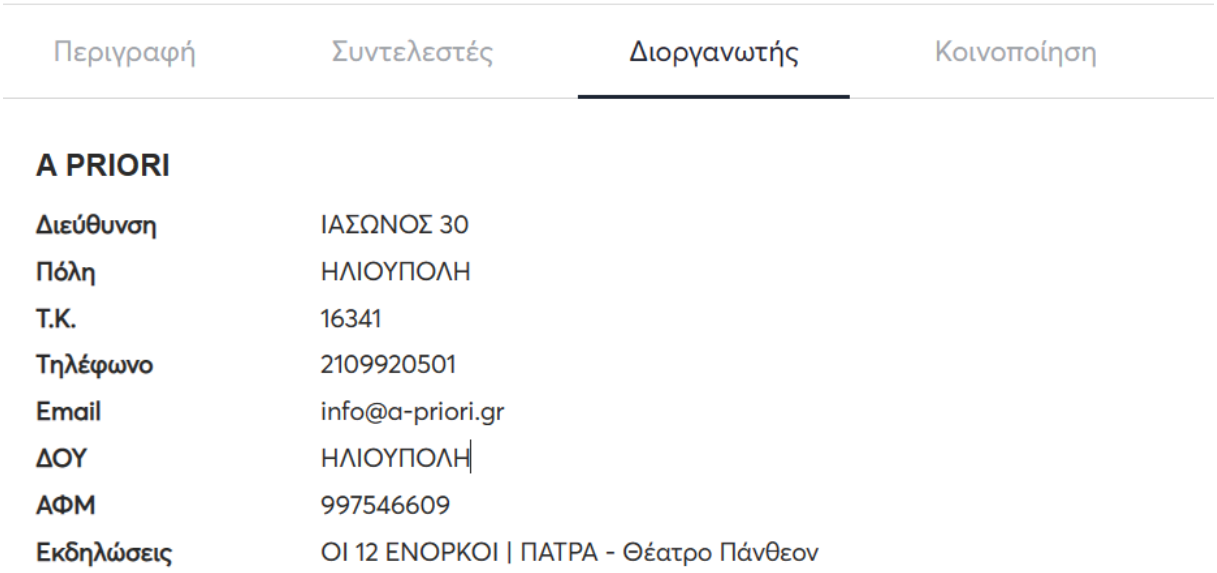
Σχήμα 6.1: Παράδειγμα Παράστασης viva.gr (Περιγραφή)

Αυτά που εισήχθησαν στον πίνακα production της βάσης από το σύστημα είναι



Σχήμα 6.2: Δεδομένα παράστασης που έγιναν εισαγωγή (Περιγραφή)

Παρατηρούμε λοιπόν ότι το σύστημα scraping βρήκε όλα τα απαραίτητα αποτελέσματα που σχετίζονται με την περιγραφή της παράστασης.



Σχήμα 6.3: Παράδειγμα Παράστασης viva.gr (Διοργανωτής)

Σχήμα 6.4: Δεδομένα παράστασης που έγιναν εισαγωγή (Διοργανωτής)

Αξίζει να αναφερθεί ότι οι σελίδες της παράστασης του διοργανωτή και των εισιτηρίων είναι οι πιο εύκολες στην υλοποίηση του scraping διότι τα δεδομένα είναι πάντα σωστά δομημένα με τους αντίστοιχους κόμβους τους και δεν υπάρχει πρόβλημα συλλογής των στοιχείων.

Το φαινόμενο αυτό της μη εύκολης δόμησης των στοιχείων βρίσκεται στην σελίδα των ηθοποιών και των ρόλων που εκεί πέρα χρειάζεται αρκετή επεξεργασία του κειμένου και επίσης δεν έχουν πάντα την ίδια δομή (μερικές φορές είναι απλώς κείμενο, άλλες φορές σε μορφή λίστας κουκίδων και μερικές φορέ σε μορφή πίνακα, πράγμα που δυσκολεύει την διαδικασία της ορθης συλλογής

Κείμενο: Reginald Rose

Μετάφραση/Σκηνοθεσία: Κωνσταντίνα Νικολαΐδη

Δραματουργική επεξεργασία: Κωνσταντίνα Νικολαΐδη & Νότης Παρασκευόπουλος

Σκηνικά: David Negrin

Κοστούμια: Κική Μήλιου

Πρωτότυπη μουσική: Γιώργος Περού

Κίνηση: Χριστίνα Φωτεινάκη

Σχεδιασμός φωτισμών: Αλέξανδρος Αλεξάνδρου

Υλοποίηση φωτισμών: Μανώλης Μπράτσης

Βοηθοί σκηνοθέτη: Μαγδαληνή Παλιούρα, Κατερίνα Κωνσταντέλλου

Επικοινωνία: Άντζυ Νομικού

Creative Agency: GRID FOX

Παραγωγή: A PRIORI

Οι 12 ένορκοι (αλφαβητικά)

Τάσος Γιαννόπουλος, Δημήτρης Δεγαΐτης, Μάνος Ζαχαράκος, Αλέξανδρος Καλπακίδης, Θανάσης Κουρλαμπάς, Βαγγέλης Κρανιώτης, Νίκος Μέλλος, Κωνσταντίνος Μπάζας, Γιώργος Μπινιάρης, Τάσος Παπαδόπουλος, Ορέστης Τρίκας, Βασίλης Φακανάς

Στον ρόλο του φύλακα ο Αλέξης Σταυριανός

Σχήμα 6.5: Παράδειγμα Παράστασης viva.gr (Συντελεστές)

ID	fullname	Role
19653	Reginald Rose	Κείμενο
19654	Κωνσταντίνα Νικολαΐδη	Μετάφραση/Σκηνοθεσία
19655	Κωνσταντίνα Νικολαΐδη & Νότης Παρ...	Δραματουργική επεξεργασία
19656	David Negrin	Σκηνικά
19657	Κική Μήλιου	Κοστούμια
19658	Γιώργος Περού	Πρωτότυπη Μουσική
19659	Χριστίνα Φωτεινάκη	Κίνηση
19660	Αλέξανδρος Αλεξάνδρου	Σχεδιασμός Φωτισμών
19661	Μανώλης Μπράτσης	Ηθοποιός
19662	Μαγδαληνή Παλιούρα	Βοηθοί σκηνοθέτη
19663	Κατερίνα Κωνσταντέλλου	Βοηθοί σκηνοθέτη
19664	Άντζυ Νομικού	Επικοινωνία
19665	Grid Fox	Creative Agency
19666	Τάσος Γιαννόπουλος	Ηθοποιός
19667	Δημήτρης Δεγαΐτης	Ηθοποιός
19668	Μάνος Ζαχαράκος	Ηθοποιός
19669	Αλέξανδρος Καλπακίδης	Ηθοποιός
19670	Θανάσης Κουρλαμπάς	Ηθοποιός
19671	Βαγγέλης Κρανιώτης	Ηθοποιός
19672	Νίκος Μέλλος	Ηθοποιός
19673	Κωνσταντίνος Μπάζας	Ηθοποιός
19674	Γιώργος Μπινιάρης	Ηθοποιός
19675	Τάσος Παπαδόπουλος	Ηθοποιός
19676	Ορέστης Τρίκας	Ηθοποιός
19677	Βασίλης Φακανάς	Ηθοποιός

Σχήμα 6.6: Δεδομένα παράστασης που έγιναν εισαγωγή (Συντελεστές)

Απο τα παραπάνω δεδομένα βλέπουμε ότι σε γενικές γραμμές έχει γίνει αρκετά καλό scraping. Το σύστημα κατάφερε να εντοπίσει ότι οι συντελεστές που είναι διαχωρισμένοι με κόμμα έχουν τον ρόλο του ηθοποιού και στους υπόλοιπους τους έδωσε το σωστό ρόλο που τους αναλογεί.

Στην γραμμή Βοηθοί σκηνοθέτη: Μαγδαληνή Παλιούρα, Κατερίνα Κωνσταντέλλου βλέπουμε ότι κατάφερε να τους ξεχωρίσει και να τους προσθέσει ορθώς ξεχωριστά στην βάση. Ενώ για την γραμμή Δραματουργική επεξεργασία: Κωνσταντίνα Νικολαΐδη & Νότης Παρασκευόπουλος τους εισήγαγε στην βάση ως μια οντότητα καθώς δεν αναγνώρισε το & ως διαχωριστικό όπως το κόμμα. Όμως δεν εισήγαγε την γραμμή Στον ρόλο του φύλακα ο Αλέξης Σταυριανός.

Παρατηρούμε ότι το σύστημα συλλέγει με πάρα πολύ μεγάλη ακρίβεια τα δεδομένα που είναι ορθώς δομημένα όπως περιγραφή παράστασης, διοργανωτής, εισιτήρια κ.λπ, ενώ για την σελίδα των συντελεστών ουσιαστικά δεν είναι τόσο scraping αλλα επεξεργασία συμβολοσειρών, παρόλα αυτά έκανε εισαγωγή των συντελεστών σε αρκετά ικανοποιητικό βαθμό.

Τέλος αξίζει να αναφερθεί ότι το παραπάνω θέμα μπορεί να διορθωθεί αφού το σύστημα υποστηρίζει επεξεργασία των ονομάτων των ηθοποιών και προσθήκη ρόλων/ηθοποιών για μια συγκεκριμένη παράσταση σε δεύτερο χρόνο από τον διαχειριστή του συστήματος. Μπορούμε να πούμε δηλαδή με σιγουριά ότι ο διαχειριστής μπορεί με ευκολία να επέμβει από το σύστημα και να προσθέσει τα δεδομένα που λείπουν όπως αναφέρθηκε στο κεφάλαιο 4.

Κεφάλαιο 7ο: Συμπεράσματα και προτάσεις βελτίωσης

Όπως κάθε εφαρμογή και κάθε τι στον πραγματικό κόσμο μπορεί να βελτιωθεί έτσι και το σύστημα αυτό θα μπορούσε να βελτιωθεί σε κάποιους τομείς.

Όσον αφορά το scraping θα ήταν σημαντικό να προστεθεί curation για τις ονομασίες των ηθοποιών ώστε αυτό το σύστημα να κρίνει αν η συμβολοσειρά αποτελεί όνομα ή όχι, το σύστημα έχει υλοποίηση μια συνάρτηση IsAPersonName όμως δεν ανταποκρίνεται σε πιο σύνθετες ανάγκες παρά μόνο σε απλές ονομασίες. Με τον τρόπο αυτό δεν θα εισάγονται ονόματα που δεν είναι ονόματα και ο διαχειριστής του συστήματος δεν θα χρειάζεται να κάνει τόσες διορθώσεις.

Επίσης στην οθόνη της παραμετροποίησης του Scraping θα μπορούσαν να προστεθούν και άλλα χαρακτηριστικά όπως αποστολή με email μιας σύνοψης όλων των παραστάσεων που έγιναν εισαγωγή στην βάση στην ηλεκτρονική διεύθυνση του διαχειριστή του συστήματος, με αποτέλεσμα να μην υπάρχει η ανάγκη ο διαχειριστής να είναι συνεχώς πάνω από το πρόγραμμα για να έχει την επιτήρηση του συστήματος.

Εκτός από αυτό θα μπορούσαν να προστεθούν οθόνες που δείχνουν στον διαχειριστή ποιές παραστάσεις έχουν ασαφή δεδομένα όπως για παράδειγμα έλλειψη της χρονικής διάρκειας της παράστασης ή παραστάσεις που δεν έχουν συντελεστές έτσι ώστε να βοηθούν τον διαχειριστή του συστήματος στην επίβλεψη του.

Κλείνοντας μπορούμε να καταλήξουμε ότι το σύστημα που υλοποιήθηκε δίνει όλες τις απαραίτητες δυνατότητες στον διαχειριστή να τροποποιήσει τις πληροφορίες που εισήχθησαν στην βάση δεδομένων μέσα από το σύστημα χωρίς να έχει γνώσεις SQL αλλά με απλή του αλληλεπίδραση με το σύστημα. Με αυτόν τον τρόπο δίνεται η δυνατότητα να υπάρχει μια συνεπής βάση δεδομένων με ορθά αποτελέσματα. Όλα τα παραπάνω μπορούν να γίνει είτε αφού δει τα αποτελέσματα των θεατρικών παραστάσεων που έγιναν εισαγωγή πατώντας το κουμπί “Εναρξη Scraping” και να δει αν κάποιες πληροφορίες λείπουν και αν ενδεχομένως λείπουν να τις προσθέσει. Για παράδειγμα μπορεί το σύστημα να μην έφερε σωστά την διάρκεια της παράστασης ή να μην πρόσθεσε κάποιον ηθοποιό ως συντελεστή της παράστασης. Αυτή η διαδικασίας διόρθωσης φαίνεται πως μπορεί να γίνει στο σχήμα 4.8. Έτσι υπάρχει η δυνατότητα του διαχειριστή να επέμβει και να κάνει αλλαγές.

Επίσης η ύπαρξη του καταγραφικού δίνει την δυνατότητα στον διαχειριστή να εντοπίσει ποιές παραστάσεις έγιναν scraping και πότε δίνοντας του την πλήρη εικόνα για το πως εξελίσσεται η διαδικασία του scraping και ποιές εγγραφές κρίθηκαν ότι θα πρέπει να εισαχθούν στην βάση και ποιές όχι. ‘Ετσι δίνεται η δυνατότητα ελέγχου του scraping από τον διαχειριστή του συστήματος αφού του δίνει την δυνατότητα να επιβλέπει την όλη διαδικασία.

Εκτός από αυτό ο διαχειριστής είναι σε θέση να ρυθμίσει σε ποιά βάση γίνονται εισαγωγή τα δεδομένα, αν επιθυμεί να χρησιμοποιήσει το API που έχει δημιουργηθεί, να ρυθμίσει το σύστημα πότε θα εκτελεί τις διαδικασίες του Scraping και πόσες παραστάσεις θα εισαχθούν ανά σύνοδο.

Καταλήγουμε λοιπόν ότι η υλοποίηση αυτού του συστήματος scraping δίνει όλα τα απαραίτητα εργαλεία στον διαχειριστή του scraping ώστε να μπορεί να συμβάλει, τροποποιόντας ασαφείς πληροφορίες (όπως για παράδειγμα κάποιο όνομα ηθοποιού), να διαγράφει λάθος πληροφορίες (κάποιος ηθοποιός που κρίθηκε λάθος από το σύστημα ότι δεν είναι ηθοποιός) και να προσθέτει νέες πληροφορίες (προσθήκη κάποιου συντελεστή μαζί με τον ρόλο του στην παράσταση ή προσθήκη ενός εισιτηρίου που ενδεχομένως να μην έγινε σωστά scrap).

Έτσι συμπεραίνουμε ότι το σύστημα που υλοποιήθηκε για τις ανάγκες της εργασίας δίνει την δυνατότητα στον διαχειριστή του συστήματος να έχει το scraping υπό την επίβλεψη του και να συμμετέχει στην διαδικασία συντήρησης της όλο και επεκτεινόμενης βάσης δεδομένων με σαφή δεδομένα.

Προσωπικά η εργασία αυτή μου έδωσε την δυνατότητα να γνωρίσω τον κόσμο του scraping και να αντιληφθώ πόσο σημαντικό είναι να υπάρχει η ικανότητα σε έναν προγραμματιστή να είναι σε θέση να συλλέξει δεδομένα από διάφορες ιστοσελίδες και να αντιμετωπίζει θέματα διαφορετικής δομής όπου η κάθε μια παρουσιάζει δυσκολίες για την τέλεση του scraping είτε αυτές οι δυσκολίες είναι λόγω δομής ή περιορισμών που υπάρχουν από την ιστοσελίδα για την αποφυγή του scraping (IP Blocking, Captcha κ.λπ.).

Εκτός από αυτό μου έδωσε την δυνατότητα να χρησιμοποιήσω τις γνώσεις που πήρα από την σχολή για την υλοποίηση ενός συστήματος διαχείρισης scraping εμφανίζοντας δυναμικά δεδομένα και καταγράφοντας τα δίνοντας σε έναν τρίτο χρήστη την δυνατότητα να διαχειριστει και να παραμετροποιήσει την λειτουργία του συστήματος που δημιουργήθηκε για τις ανάγκες της εργασίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία

[1] C#: 2 books in 1 - The Ultimate Beginner & Intermediate Guides to Mastering C# Programming Quickly (Computer Programming)

[2] Web Scraping Basics for Recruiters: Learn How to Extract and Scrape Data from the Web

[3] Web Scraping with Python: Collecting More Data from the Modern Web

Internet Sites

[1] Web Scraping | What is Web Scraping [Online]. Διαθέσιμο: https://www.zyte.com/learn/what-is-web-scraping

[2] What is a web crawler? | [Online]. Διαθέσιμο: https://www.cloudflare.com/learning/bots/what-is-a-web-crawler/

[3] Is Web Scraping Legal? Ethical Web Scraping Guide in 2023. [Online]. Διαθέσιμο: https://research.aimultiple.com/web-scraping-ethics/

[4] Web Scraping Laws. [Online]. Διαθέσιμο: https://www.termsfeed.com/blog/web-scraping-laws/

[5] Web Scraping Protection: How to Prevent Scraping & Crawler Bots. [Online]. Διαθέσιμο: https://datadome.co/learning-center/scraper-crawler-bots-how-to-protect-your-website-against-intensive-scraping/

[6] The importance of Data Analytics and Web Scraping [Online]. Διαθέσιμο: https://scrapingrobot.com/blog/importance-of-data/

[7] How Much Data Is Generated Every Minute? [Online]. Διαθέσιμο: https://www.socialmediatoday.com/news/how-much-data-is-generated-every-minute-infographic-1/525692/

[8] A tour of C# language [Online]. Διαθέσιμο: https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/

[9] About Python [Online]. Διαθέσιμο: https://pythoninstitute.org/about-python

[10] About Ruby [Online]. Διαθέσιμο: https://www.ruby-lang.org/en/about/

[11] Nokogiri [Online]. Διαθέσιμο: https://nokogiri.org/#features-overview

[12] R - Overview [Online]. Διαθέσιμο: https://www.tutorialspoint.com/r/r_overview.htm

[13] Visual Studio: IDE and Code Editor [Online]. Διαθέσιμο: https://visualstudio.microsoft.com/

[14] Visual Studio Enterprise vs. Professional: Essential Differences in 2023 [Online]. Διαθέσιμο: https://blog.ndepend.com/visual-studio-enterprise-vs-professional/

[15] Visual Studio IDE Documentation [Online]. Διαθέσιμο: https://learn.microsoft.com/en-us/visualstudio/ide

[16] Github Docs [Online]. Διαθέσιμο: https://docs.github.com/

[17]What is debugging? [Online]. Διαθέσιμο:
https://www.techtarget.com/searchsoftwarequality/definition/debugging

[18] What is MySQL? [Online]. Διαθέσιμο:
https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

[19] What is Log Management? [Online]. Διαθέσιμο:
https://www.crowdstrike.com/cybersecurity-101/observability/log-management/