ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Συστήματος για Δημιουργία Υποτίτλων στο YouTube»

Του φοιτητη
Σαλαμπάση Σαββα
Αρ. Μητρώου: 174960

Επιβλέπων
Ονοματεπώνυμο: Τεκτονίδης
Δημήτριος

Ημερομηνία 29/1/2024

Τίτλος Δ.Ε. Ανάπτυξη Συστήματος για Δημιουργία Υποτίτλων στο YouTube
Κωδικός Δ.Ε. 23217
Ονοματεπώνυμο φοιτητή/τών. Σαλαμπάσης Σάββας
Ονοματεπώνυμο εισηγητή. Τεκτονίδης Δημήτριος
Ημερομηνία ανάληψης Δ.Ε. 14/8/2023
Ημερομηνία περάτωσης Δ.Ε. 31/1/2024

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 διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής 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.

Περίληψη

Η Διπλωματική εργασία παρουσιάζει το σχεδιασμό, την ανάπτυξη και την αξιολόγηση μιας νέας διαδικτυακής εφαρμογής που στοχεύει στη βελτίωση της προσβασιμότητας και του User Experience για την δημιουργία διαδικτυακού περιεχομένου. Με τη διαρκώς αυξανόμενη εξάρτηση από τις ψηφιακές πλατφόρμες για την κατανάλωση πληροφοριών, η διασφάλιση της ένταξης διάφορων ομάδων χρηστών γίνεται πρωταρχικής σημασίας. Το επίκεντρο αυτής της εφαρμογής είναι τα YouTube βίντεο, μια πανταχού παρούσα πηγή διαδικτυακού περιεχομένου, και στόχος της είναι να αντιμετωπίσει την ανάγκη για βελτιωμένη προσβασιμότητα μέσω υποτίτλων σε οσες περισσότερες γλώσσες δυνατόν.

Η εφαρμογή έχει σχεδιαστεί για να δίνει τη δυνατότητα στους χρήστες να δημιουργούν, να επεξεργάζονται και να προσαρμόζουν υπότιτλους για βίντεο YouTube με φιλικό και διαισθητικό τρόπο. Αξιοποιώντας σύγχρονες web τεχνολογίες, οπως το Angular Framework, η εφαρμογή ενσωματώνεται άψογα με διάφορα APIs, επιτρέποντας στους χρήστες να εισάγουν και να επεξεργάζονται βίντεο στην εφαρμοφή απευθείας, χρησιμοποιώντας serverless solutions οπως το Firebase. Το User Interface δίνει προτεραιότητα στην απλότητα και την αποτελεσματικότητα, καθιστώντας τη διαδικασία υποτιτλισμού προσβάσιμη σε ένα ευρύ φάσμα χρηστών, συμπεριλαμβανομένων εκείνων με ελάχιστες τεχνολογικές δεξιότητες.

Συμπερασματικά, η παρούσα εργασία συμβάλλει στον τομέα της τεχνολογικής προσβάσιμοτητας, προσφέροντας μια πρακτική λύση για τον εύκολο υποτιτλισμό των βίντεο του YouTube μέσω μιας φιλικής προς τον χρήστη διαδικτυακής εφαρμογής.

«Application Development for YouTube Subtitling»

«Salampasis Savvas»

Abstract

This Thesis presents the development and design of a new web application which aims to improve accessibility for any language speaker through the improvement of YouTube video subtitling. With the continuously increasing reliance on digital platforms for content consumption, ensuring that a video is accessible to a wide variety of people is of critical importance. This application is focused on YouTube videos, an endless source of digital content, and its aim is to provide a platform that will make it easier for a creator to serve his content in as many languages as possible through subtitling.

This application is designed to give users the ability to create, edit and customize subtitles for their YouTube videos through a user-friendly and easy-to-use User Interface. It utilizes modern Web Technologies, like the Angular Framework, as a result the application integrates easily with multiple APIs, allowing users to easily store and edit their subtitles, using serverless solution, Firebase. The User Interface prioritizes simplicity and efficiency, making it accessible to a wide variety of users without requiring extensive technical skills.

In conclusion, this Thesis contributes to the accessibility domain, by providing a practical solution for easy YouTube video subtitling through a user-friendly interface.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Κεφάλαιο 1ο: Ο ρόλος του Βίντεο σήμερα

Εισαγωγή

Τα τελευταία χρόνια, το βίντεο έχει γίνει το πιο δημοφιλές μέσο σε όλο τον κόσμο [1]. Το βλέπουμε παντού - στην τηλεόραση, στα μέσα κοινωνικής δικτύωσης, στα γραπτά μηνύματα, στις διαδικτυακές διαφημίσεις, ακόμη και στα μενού εστιατορίων.

Επειδή τα βίντεο έχουν καταστεί ιδιαιτέρως σημαντικά για τη νεότερη γενιά, η δημιουργία οποιουδήποτε βίντεο δεν ήταν ποτέ πιο απλή. Πλέον, ο καθένας έχει άμεση πρόσβαση σε εργαλεία για την καταγραφή, επεξεργασία και δημοσίευση βίντεο, καθώς και πολλαπλά βοηθητικά εργαλεία για την ομαλή δημιουργία τους.

Η παραγωγή βίντεο έχει γίνει καθημερινό κομμάτι και τρόπος διαβίωσης πολλών ανθρώπων, είτε σαν hobby είτε επαγγελματικά. Καθημερινά, στην πλατφόρμα του YouTube μόνο, συνολικά ανεβαίνουν τοσα βίντεο τα οποία η συνολική διάρκεια τους αθροιζεται στις 400 ωρες. Για να παρακολουθήσει κανείς όλο το περιεχόμενο που ανεβαίνει μέσα σε μια μόνο ημέρα, θα χρειαστούν 66 χρόνια χωρίς διακοπή ύπνου. Επιπλέον, καθημερινά καταναλώνονται συνολικά πάνω από 1 δισεκατομμύριο ώρες βίντεο στο YouTube.

Τα βίντεο έχουν εξελιχθεί σε κεντρικό στοιχείο της καθημερινότητάς μας, καλύπτοντας πλατφόρμες όπως το YouTube, το Netflix, το Twitter, και το TikTok [2]. Η συνεχώς εξελισσόμενη τεχνολογία και ο ταχύτατος ρυθμός ζωής μας, έχουν οδηγήσει μεγάλο μέρος της κοινωνίας σε συχνές αποσπασης προσοχής, με τα βίντεο να αποτελούν την πρώτη επιλογή για την ενημέρωση και την εκμάθηση νέων πραγμάτων και δεξιοτήτων. Τα βίντεο ενεργοποιούν περισσότερα τμήματα του εγκεφάλου μας, ενώ με τα οπτικά και ακουστικά κίνητρα καταφέρνουν να διατηρούν την προσοχή μας με ευκολία. Παρόλα αυτά, τα βίντεο έχουν την ικανότητα να διαμορφώνουν ένα τεράστιο φάσμα συναισθημάτων στον θεατή. Είτε χρησιμοποιούνται για αφηγηματικούς σκοπούς, ψυχαγωγία ή για τη μετάδοση πληροφοριών, τα βίντεο μπορούν να δημιουργήσουν μια βαθιά σύνδεση με άτομα τόσο σε συναισθηματικό όσο και σε πνευματικό επίπεδο.

Επειδή το βίντεο αρχίζει σταδιακά να αντικαθιστά τον γραπτό λόγο, η κοινωνία έχει αρχίσει να κάνει την ερώτηση, "βίντεο έναντι κειμένου: ποιο είναι καλύτερο;". Αν και το βίντεο δεν θα αντικαταστήσει ποτέ πλήρως αυτά τα μέσα, σίγουρα θα συνεχίσει να επεκτείνεται ως «το πιο δημοφιλές περιεχόμενο που καταναλώνεται παγκοσμίως» [3]. Υπάρχουν αρκετοί λόγοι για τους οποίους το βίντεο είναι μια τόσο αποτελεσματική μορφή περιεχομένου, μάθησης, ψυχαγωγίας και γιατί πλατφόρμες διαμοίρασης βίντεο όπως το TikTok, το YouTube κ.λπ συνεχίζουν να εξελίσσονται.

1.1.1 Το βίντεο προβάλει την μη λεκτική επικοινωνία

Η γλώσσα του σώματος και ο τόνος της φωνής παίζουν τεράστιο ρόλο στη μετάδοση ενός μηνύματος. Το περιεχόμενο ενος κειμένου βασίζεται στην ακριβή και εύστοχη επιλογή λέξεων, στα σημεία στίξης και οπτικά χαρακτηριστικά, όπως emoticon, για να δημιουργήσει τον σωστό τόνο. Ωστόσο, με το βίντεο, οι θεατές είναι σε θέση να προσδιορίσουν τι ακριβώς προσπαθεί να περάσει ο ομιλητής παρατηρώντας τη γλώσσα του σώματος, τον λεκτικό τόνο και άλλες οπτικές ενδείξεις.

Όχι μόνο αυτό, αλλά τα βίντεο μπορούν να επεκταθούν ακόμη και πέρα από τη λεκτική και τη μη λεκτική επικοινωνία, συμπεριλαμβάνοντας οπτικά βοηθήματα όπως εικόνες και πλάνα, τα οποία ενισχύουν περαιτέρω τα αντικείμενα που πραγματεύονται.

1.1.2 Το βίντεο προσελκύει περισσότερο κοινό

Με τον συνδυασμό οπτικοακουστικού υλικού, δεν υπάρχει τίποτα που να μην μπορεί να κάνει το βίντεο. Επειδή είναι το πιο δημοφιλές μέσο, οι θεατές είναι έτοιμοι να ακούσουν, αρκεί να προσφέρετε ποιοτικό και συνοπτικό περιεχόμενο. Οι επιχειρήσεις μπορούν να το χρησιμοποιήσουν προς όφελός τους τα βίντεο, κάνοντας μάρκετινγκ με αυτά αντί για κείμενο και εικόνες.

Στην πραγματικότητα, οι καταναλωτές έχουν πάνω από 27 φορές περισσότερες πιθανότητες να κάνουν κλικ σε μια διαφήμιση βίντεο στο διαδίκτυο από μια τυπική banner διαφήμιση. Τα βίντεο που έχουν ενδιαφέρον, ελκυστικό περιεχόμενο και μια τελική παρότρυνση για δράση (Call to Action) είναι βέβαιο ότι θα έχουν επιτυχία, κρατώντας το ενδιαφέρον του κοινού.

1.1.3 Το βίντεο παρουσιάζει γρήγορο και πλούσιο περιεχόμενο

Δεν είναι μυστικό ότι η ανάγνωση διαρκεί πολύ περισσότερο από την παρακολούθηση ενός βίντεο. Ακόμη και τα podcast χρειάζονται περισσότερο χρόνο επειδή δεν έχουν οπτικά βοηθήματα. Το οπτικό περιεχόμενο είχε πάντα το πλεονέκτημα στη συμπύκνωση περιεχομένου. Η οπτική αισθητική σε συνδυασμό με τον ήχο επιτρέπει στις παρεχόμενες πληροφορίες να παραδίδονται πολύ πιο γρήγορα.

Για παράδειγμα, ένα γράφημα θα μπορούσε να παρουσιαστεί με έναν ομιλητή που επεξηγεί ταυτόχρονα τις πληροφορίες σε αυτό - αντί για ένα γράφημα που ακολουθείται από μια παράγραφο που το εξηγεί. Τα βίντεο είναι εγγενώς πλούσια σε περιεχόμενο, το οποίο είναι πιο ικανοποιητικό από την προβολή απλών εικόνων ή κουραστικών λέξεων

1.1.4 Το μέλλον του βίντεο

Το βίντεο αναδεικνύεται ως το αναμφισβήτητο μέσο του μέλλοντος. Ενώ τα άρθρα, τα podcast και τα γραφήματα παίζουν σημαντικό ρόλο στο εξελισσόμενο τοπίο, η ελκυστικότητα και ο αντίκτυπος του βίντεο δεν μπορούν να υποτιμηθούν. Οι άνθρωποι έλκονται εγγενώς από το περιεχόμενο των βίντεο και, τελικά, είναι η ανθρώπινη σύνδεση που έχει ύψιστη σημασία.

Για χρόνια, το βίντεο έχει δείξει τεράστιες δυνατότητες και τώρα είναι η κατάλληλη στιγμή για να συνειδητοποιήσει πλήρως τις δυνατότητές του. Η σημασία της ενσωμάτωσης του διαδικτυακού μάρκετινγκ βίντεο στη στρατηγική μιας επιχείρησης είναι απαράμιλλη. Η αξιοσημείωτη κοινωνική εμβέλεια των βίντεο δεν είναι τίποτα λιγότερο από εξαιρετική. Αν σκοπεύετε να συνδεθείτε με ένα παγκόσμιο κοινό, το μόνο που χρειάζεστε είναι το βίντεο.

Συστήματα Διαμοίρασης Βίντεο

1.1.5 Netflix

Σε όλη την ιστορία της ψυχαγωγίας, λίγες καινοτομίες έχουν αφήσει τόσο ανεξίτηλο σημάδι όσο η έλευση των πλατφόρμων streaming, με το Netflix να βρίσκεται στην πρώτη γραμμή αυτής της ψηφιακής επανάστασης.

Η κομβική στιγμή ήρθε το 2007 όταν το Netflix παρουσίασε την streaming υπηρεσία του, επιτρέποντας στους συνδρομητές να παρακολουθούν αμέσως μια τεράστια βιβλιοθήκη ταινιών και τηλεοπτικών εκπομπών στις προσωπικές συσκευές τους.

Για πρώτη φορά, οι πελάτες μπορούσαν να παρακολουθήσουν μια τηλεοπτική εκπομπή ή μια ταινία σε υπολογιστή, smart τηλεόραση, tablet, στο κινητό ή gaming συσκευη. Αυτό σηματοδότησε μια σεισμική

αλλαγή στον κλάδο, καθώς εξάλειψε την ανάγκη για φυσικά μέσα και εγκαινίασε την εποχή της on-demand παρακολούθησης και στιγμιαίας ψυχαγωγίας.

Την ίδια περίοδο, το Netflix άρχισε να ανεβάζει ολόκληρες σεζόν καθιερωμένων τηλεοπτικών σειρών απευθείας, δημιουργώντας ουσιαστικά την τάση του binge-watching, σε αντίθεση με το παραδοσιακό εώς τοτε μοντέλο, του ενός επεισοδίου την εβδομάδα της αντίστοιχης εκπομπής ή τηλεοπτικής σειράς.

Αυτος ο συνδυασμός καινοτομιών ισοπέδωσε τον κλάδο της ενοικίασης βίντεο με τα εώς τοτε πασίγνωστα video club. Σύντομα, οι καλωδιακές εταιρείες και τα τηλεοπτικά δίκτυα άρχισαν και αυτα να αναπτύσσουν τις δικές τους πλατφόρμες με σκοπό την προσφορά on-demand περιεχομένου.

Η εμβέλεια του Netflix εκτείνεται πολύ πέρα από την αμερικανική καταγωγή του. Η πλατφόρμα ξεκίνησε μια φιλόδοξη παγκόσμια επέκταση, καθιστώντας τις υπηρεσίες της διαθέσιμες σε πολλές χώρες. Αυτή η παγκόσμια προσέγγιση, σε συνδυασμό με το πολύγλωσσο περιεχόμενο, συνέβαλε στη θέση του Netflix ως μιας πραγματικά διεθνούς streaming δύναμης.

1.1.6 YouTube

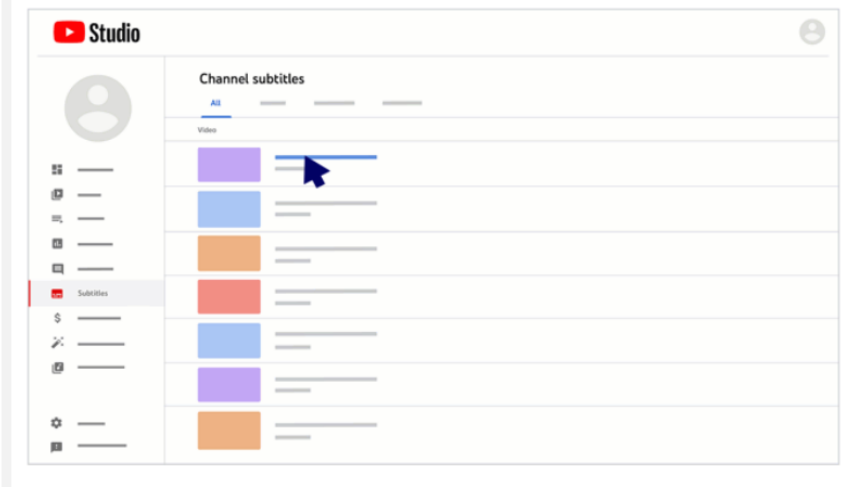
Το YouTube, μια ψηφιακή πλατφόρμα για δημοσίευση βίντεο και social interactions, είναι μια οντότητα υπό την ιδιοκτησία της Google. Κυκλοφόρησε παγκοσμίως στις 14 Φεβρουαρίου 2005, από τους πρώην υπαλλήλους της PayPal, Steve Chen, Chad Hurley και Jawed Karim. Είναι ο δεύτερος πιο δημοφιλής ιστότοπος παγκοσμίως, τον οποίο τον ξεπερνάει μονο το Google Search. Με περισσότερους από 2,5 δισεκατομμύρια μηνιαίους χρήστες, το YouTube καταγράφει συλλογικά άνω του ενός δισεκατομμυρίου ωρών παρακολούθησης ημερησίως.

Τον Οκτώβριο του 2006, η Google εξαγόρασε το YouTube, σηματοδοτώντας μια κομβική στιγμή που άλλαξε το επιχειρηματικό μοντέλο της πλατφόρμας. Αρχικά βασιζόμενο στα έσοδα από διαφημίσεις, το YouTube διαφοροποιήθηκε ώστε να περιλαμβάνει περιεχόμενο επί πληρωμή, όπως ταινίες και αποκλειστικές παραγωγές. Το YouTube Premium, μια συνδρομητική υπηρεσία που προσφέρει περιεχόμενο χωρίς διαφημίσεις, εμφανίστηκε ως μέρος αυτής της εξέλιξης. Επίσης, δόθηκε η ευκαιρία στους δημιουργούς να συμμετάσχουν στο πρόγραμμα AdSense της Google, με στόχο την ενίσχυση των εσόδων και για τα δύο μέρη. Συγκεκριμένα, τα ετήσια διαφημιστικά έσοδα του YouTube έφτασαν τα 28,8 δισεκατομμύρια δολάρια το 2021, σημειώνοντας αύξηση 9 δισεκατομμυρίων δολαρίων σε σχέση με το προηγούμενο έτος. Η πλατφόρμα ανέφερε συνολικά έσοδα 29,2 δισεκατομμυρίων δολαρίων το 2022.

Το περιεχόμενο των βίντεο στο YouTube καλύπτει διάφορες κατηγορίες, όπως μουσικά βίντεο, ειδήσεις, ταινίες μικρού μήκους, ντοκιμαντέρ και ζωντανές ροές (livestreams). Το μεγαλύτερο μέρος του περιεχομένου δημιουργείται από χρήστες της πλατφόρμας και περιλαμβάνει συνεργασίες μεταξύ μεμονωμένων δημιουργών και εταιρικών χορηγών. Καθιερωμένοι κολοσσοί όπως η Disney, η Paramount, η NBCUniversal και η Warner Bros. Discovery έχουν επίσης αξιοποιήσει τα εταιρικά κανάλια YouTube για να προσεγγίσουν ένα ευρύτερο κοινό.

1.1.6.1 Youtube Studio

Το YouTube Studio, παλαιότερα γνωστό ως YouTube Creator Studio, είναι ένα εργαλείο του YouTube που επιτρέπει στους χρήστες και τους δημιουργούς να διαχειρίζονται τα κανάλια τους, να επεξεργάζονται και να παρακολουθούν την απόδοση των βίντεό τους και να βλέπουν,να απαντούν σε σχόλια στα βίντεό τους και πιο σημαντικό/σχετικό στα πλαίσια του υποτιτλισμού και της μεταγλώττισης, είναι το εργαλείο που προσφέρουν για την δημιουργία υποτίτλων σε διάφορες γλώσσες.



Σχήμα 1.1: YouTube Studio

Το YouTube προσέφερε στο παρελθόν μια λειτουργία, που ονόμασε "Community Subtitles", όπου οι θεατές μπορούσαν να γράφουν και να υποβάλλουν υπότιτλους για δημόσια προβολή μετά την έγκριση από τον χρήστη που ανέβασε το βίντεο, αλλά αυτό καταργήθηκε τον Σεπτέμβριο του 2020 [5], ένα απο τα κενά τα οποία προσπαθεί να καλύψει η εφαρμογή στα πλαίσια της Δ.Ε

Μεταγλώττιση και Υποτιτλισμός

1.1.7 Γιατι

Ο υποτιτλισμός και η μεταγλώττιση κατέχουν κρίσιμο ρόλο στην πρόσβαση και κατανάλωση οπτικοακουστικού περιεχομένου από ένα παγκόσμιο κοινό. Λειτουργούν ως μέσα για την υπέρβαση των γλωσσικών φραγμών και των πολιτισμικών διαφορών, επιτρέποντας στους θεατές να αλληλεπιδράσουν με περιεχόμενο από οποιοδήποτε γλωσσικό υπόβαθρο και αν προέρχονται. Είτε πρόκειται για ταινίες, τηλεοπτικές εκπομπές ή ψηφιακό βίντεο, ο υποτιτλισμός και η μεταγλώττιση επιτρέπουν στους δημιουργούς περιεχομένου να διευρύνουν το κοινό τους και να βελτιώσουν τη συνολική εμπειρία τους.

Σε μια κοινωνία όπως είναι αυτή του σήμερα, ο υποτιτλισμός και η μεταγλώττιση καταρρίπτουν τα γλωσσικά εμπόδια που μπορεί να εμποδίσουν την επικοινωνία μεταξύ των ανθρώπων. Ο ρόλος του υποτιτλισμού και της μεταγλώττισης στο παγκόσμιο τοπίο των μέσων ενημέρωσης εκτείνεται πολύ πέρα από την απλή μετάφραση, συμβάλλοντας στη διασύνδεση των πολιτισμών και στον εμπλουτισμό της οπτικοακουστικής εμπειρίας ενός κοινού .

1.1.7.1 Engagement

Βελτιώνεται το User engagement κατά 80% προσθέτοντας υπότιτλους, σύμφωνα με μια μελέτη της PLYMedia [7]. Αυτό οφείλεται στο γεγονός ότι οι άνθρωποι έλκονται φυσικά στο να παρακολουθούν μαζί με την εικόνα, και το κείμενο, για καλύτερη κατανόηση του περιεχομένου που καταναλώνουν. Η ίδια μελέτη, διαπίστωσε επίσης ότι οι περισσότεροι χρήστες ενεργοποιούν τους υπότιτλους όταν παρακολουθούν ταινίες, τηλεοπτικές εκπομπές ή αλλα media, ακόμα κι αν είναι φυσικοί ομιλητές της γλώσσας.

1.1.7.2 Προσέγγιση διεθνούς κοινου

Σε έναν τόσο διασυνδεδεμένο κόσμο, το περιεχόμενο ταξιδεύει πέρα από τις αγγλόφωνες χώρες. Η βελτιστοποίηση των υπότιτλων των βίντεό σας με τη μητρική γλώσσα του target κοινού σας μπορεί να είναι ένας πολύ καλός τρόπος για να δημιουργήσετε πιο inclusive περιεχόμενο.

1.1.7.3 Sound-off κοινό

Κατά μέσο όρο, το 63% των θεατών παρακολουθεί βίντεο χωρίς ήχο. Αυτός ο αριθμός είναι πιθανώς ακόμη υψηλότερος τώρα που οι άνθρωποι εργάζονται από το σπίτι σε μικρότερους χώρους με άλλους γύρω τους.

Αυτό σημαίνει ότι είναι πιο σημαντικό από ποτέ για το κοινό να μπορεί να συνδεθεί με το περιεχόμενό σας χωρίς να χρειάζεται να ενεργοποιήσει τον ήχο του.

1.1.7.4 Δεν απαιτεί εξειδίκευση

Η προσθήκη υπότιτλων στα βίντεό σας δεν απαιτεί πλέον δεξιότητες επεξεργασίας βίντεο ή πολύ χρόνο, έτσι πολλοί δημιουργοί επιλέγουν να προσθέτουν υπότιτλους στο περιεχόμενό τους απλά επειδή μπορούν. Ποτέ δεν ήταν πιο εύκολο να τραβήξετε και να επεξεργαστείτε ένα βίντεο από ό,τι σήμερα.

Είναι αδιαμφισβήτητο ότι η προσθήκη υποτίτλων σε βίντεο μπορεί να είναι το βασικό χαρακτηριστικό που ξεχωρίζει το περιεχόμενό σας από το περιεχόμενο άλλων δημιουργών. Το να προσφέρετε έναν τρόπο αλληλεπίδρασης με το περιεχόμενό σας με περισσότερους από έναν τρόπους και να το κάνετε πιο προσιτό σε ευρύτερο κοινό δημιουργεί μια αξέχαστη εμπειρία που σχετίζεται με την επωνυμία σας. Και, το καλύτερο μέρος; Είναι απίστευτα εύκολο αυτές τις μέρες.

1.1.8 Προβλήματα

Παρά την τεράστια σημασία τους, ο υποτιτλισμός και η μεταγλώττιση αντιμετωπίζουν διάφορες προκλήσεις. Ένα πρωταρχικό ζήτημα είναι η διατήρηση της γλωσσικής και πολιτισμικής ακρίβειας. Η μετάφραση του διαλόγου διατηρώντας αποχρώσεις και ιδιωματισμούς είναι περίπλοκη και η εύρεση της σωστής ισορροπίας μεταξύ της κυριολεκτικής μετάφρασης και της γλωσσικής προσαρμογής είναι ζωτικής σημασίας.

Επιπλέον, ο συγχρονισμός των υποτίτλων με τις ενέργειες στην οθόνη και τις κινήσεις των χειλιών αποτελεί τεχνική πρόκληση, ιδιαίτερα σε σκηνές με γρήγορο ρυθμό. Ταυτόχρονα, κατά τη μεταγλώττιση η αντιστοίχιση των κινήσεων των χειλιών στην αρχική τους γλώσσα, διατειρώντας το επιδιωκόμενο νόημα αποτελεί και αυτό μια ιδιαίτερη πρόκληση. Η επιλογή κατάλληλων εκφράσεων, ο χειρισμός του χιούμορ και η προσαρμογή των πολιτισμικών αναφορών είναι ευαίσθητες εργασίες, οι οποίες χρειάζονται ιδιαίτερο χειρισμό για την παραγωγή του κατάλληλου μηνύματος.

1.1.8.1 Πολλαπλοί Ομιλητές

Μία από τις πιο σημαντικές προκλήσεις στον υποτιτλισμό προκύπτει όταν έχουμε να κάνουμε με πολλούς ομιλητές. Αυτή η πολυπλοκότητα είναι ιδιαίτερα εμφανής σε σκηνές με γρήγορους ρυθμούς συνομιλίας, λογομαχίες ή συζητήσεις που περιλαμβάνουν πολλούς χαρακτήρες. Οι υποτιτλιστες πρέπει να διασφαλίζουν ότι το αντίστοιχο κείμενο είναι ορατό καθώς μιλάνε οι χαρακτήρες ή άνθρωποι ανα πάσα στιγμή. Αυτή η πρόκληση γίνεται πιο έντονη σε σκηνές μεγάλου πλήθους ή τσακωμών, απαιτώντας έντονη συγκέντρωση από επαγγελματίες υποτιτλιστές για να καταγραφεί κάθε αντίδραση και λέξη με ακρίβεια.

1.1.8.2 Μήκος κειμένου

Οι διαφορές στον αριθμό των χαρακτήρων και των λέξεων μεταξύ των γλωσσών αποτελούν μεγάλο εμπόδιο στον υποτιτλισμό. Οι μεταφραστές πρέπει να συμπυκνώνουν προτάσεις με διαφορετικό μήκος, διατηρώντας παράλληλα την ακρίβεια των συμφραζομένων. Αυτή η πρόκληση είναι πιο εμφανής κατά τη μετάφραση στα αγγλικά, όπου επικρατούν συνοπτικές δηλώσεις. Οι αυστηροί κανόνες που διέπουν τη δημιουργία υποτίτλων, όπως ο περιορισμένος χώρος και η αναγνωσιμότητα, προσθέτουν πολυπλοκότητα σε αυτήν την εργασία.

1.1.8.3 Στυλ

Η διατήρηση του επιθυμητού στυλ και του νοήματος κατά τη μετάφραση είναι απαραίτητη για τον αποτελεσματικό υποτιτλισμό. Οι άμεσες μεταφράσεις λέξης προς λέξη μπορούν να θέσουν σε κίνδυνο το περιεχόμενο του αρχικού κειμένου. Ενώ ορισμένοι μεταφραστές μπορεί να καταφεύγουν σε αυτοματοποιημένα εργαλεία, γενικά είναι μια τεχνική η οποία αποθαρρύνεται. Μια ομάδα ικανών μεταφραστών, που γνωρίζουν καλά και τις δύο γλώσσες, διασφαλίζει ότι το περιεχόμενο με τους υπότιτλους παραμένει πιστό στο πρωτότυπο, προσαρμόζοντας στοιχεία όπως το μήκος και ο χρονισμός ώστε να έχει απήχηση στο κοινό. Ο στόχος είναι να διατηρηθεί η φυσική ροή, η αναγνωσιμότητα και η απόλαυση του περιεχομένου χωρίς να μειώνεται η ουσία του.

1.1.9 Ιδιαιτερότητες Ελληνικής Γλώσσας

Ο υποτιτλισμός για την ελληνική γλώσσα θέτει ένα μοναδικό σύνολο προκλήσεων λόγω των γλωσσικών και πολιτισμικών αποχρώσεων που είναι εγγενείς στη γλώσσα. Αυτές οι δυσκολίες μπορούν να αποδοθούν σε διάφορους παράγοντες και η κατανόησή τους είναι ζωτικής σημασίας για τη δημιουργία ακριβών και εύστοχων υπότιτλων.

Τα ελληνικά είναι γνωστά για το πλούσιο λεξιλόγιό τους και τις πολύπλοκες δομές προτάσεων. Η μετάφραση αυτών των περίπλοκων προτάσεων σε συνοπτικούς υπότιτλους διατηρώντας παράλληλα το νόημά τους μπορεί να αποδειχθεί ιδιαίτερα δύσκολη πρόκληση. Οι υποτιτλιστές αντιμετωπίζουν συχνά το δίλημμα της μετάδοσης της ουσίας του διαλόγου εντός των περιορισμών του χρόνου και του χώρου. Επίσης, Τα ελληνικά είναι μια γλώσσα που τείνει να είναι πιο λεκτική και εκφραστική σε σύγκριση με τα αγγλικά. Η επίτευξη ισοδύναμου συναισθηματικού και γλωσσικού αντίκτυπου στους υπότιτλους απαιτεί προσεκτική εξέταση στην επιλογή των λέξεων και την αναδιάρθρωση των προτάσεων. Η αποτύπωση των αποχρώσεων του χιούμορ, του σαρκασμού και των συναισθημάτων αποτελεί μια πολύ ευαίσθητη εργασία.

Ο ελληνικός πολιτισμός είναι βαθιά ριζωμένος στην ιστορία, τη μυθολογία και τις παραδόσεις. Οι υποτιτλιστές πρέπει να γνωρίζουν καλά την ελληνική κουλτούρα για να μεταφέρουν με ακρίβεια το πλαίσιο ορισμένων φράσεων, ιδιωμάτων ή αναφορών που μπορεί να μην έχουν άμεσο ισοδύναμο σε άλλες γλώσσες. Η αποτυχία ή αδυναμία να πετύχει κάποιος αυτο, μπορεί να οδηγήσει σε απώλεια νοήματος ή παρερμηνεία μιας πρότασης.

Η ελληνική, όπως και πολλές άλλες γλώσσες, έχει διακριτά επίπεδα ευγένειας και πολλαπλές μορφές αντωνυμιών. Η σωστή επιλογή των αντωνυμιών είναι καθοριστική για τη σωστή μετάδοση των σχέσεων μεταξύ των χαρακτήρων. Οι μεταφραστές πρέπει να περιηγηθούν με προσοχή και ακρίβεια σε αυτές τις λεπτές λεπτομέρειες για να διασφαλίσουν ότι οι υπότιτλοι αντικατοπτρίζουν πιστά την κοινωνική δυναμική που περιέχεται στο διάλογο.

Οι υπότιτλοι υπόκεινται σε τεχνικούς περιορισμούς, όπως η ταχύτητα ανάγνωσης και ο αριθμός των χαρακτήρων ανά γραμμή. Οι ελληνικές λέξεις μπορεί να είναι μεγαλύτερες από τις αντίστοιχες αγγλικές τους, γεγονός που μπορεί να οδηγήσει σε προβλήματα συγχρονισμού. Η επίτευξη ισορροπίας μεταξύ αναγνωσιμότητας και ακριβούς μετάφρασης είναι ζωτικής σημασίας.

Η Ελλάδα έχει διάφορες τοπικές διαλέκτους και προφορές, και αυτές μπορούν να επηρεάσουν τον τρόπο ομιλίας των χαρακτήρων. Οι υποτιτλιστές μπορεί να χρειαστεί να λάβουν αποφάσεις σχετικά με το πώς να αναπαραστήσουν αυτές τις παραλλαγές στη γραπτή μορφή χωρίς να θυσιάσουν τη σαφήνεια μιας πρότασης.

Συμπερασματικά, ο υποτιτλισμός για την ελληνική γλώσσα απαιτεί βαθιά κατανόηση των γλωσσικών περιπλοκών, του πολιτιστικού πλαισίου και της ικανότητας επίτευξης ισορροπίας μεταξύ της πιστότητας στο αρχικό περιεχόμενο και των περιορισμών του μέσου. Οι έμπειροι υποτιτλιστές πρέπει να ανταποκριθούν σε αυτές τις προκλήσεις για να παρέχουν μια αυθεντική και ευχάριστη εμπειρία προβολής για το κοινό σε όλο τον κόσμο.

1.1.10 Πως γίνεται ο υποτιτλισμός σήμερα

Το πρώτο βήμα στη διαδικασία υποτιτλισμού είναι η μεταφορά του ήχου από το βίντεο, σε γραπτή μορφή, είτε ηλεκτρονικά, είτε σε κάποιο φυσικό έγγραφο. Μόλις δημιουργηθεί αυτό το έγγραφο, ένας υποτιτλιστής θα εξετάσει πώς να χρησιμοποιήσει καλύτερα αυτό το κείμενο για να δημιουργήσει τους πιο ευστοχους υπότιτλους.

Οι επαγγελματίες υποτιτλιστές συνήθως αξιοποιούν κάποιο εξειδικευμένο λογισμικό και hardware όπου το βίντεο αποθηκεύεται ψηφιακά, καθιστώντας κάθε καρέ άμεσα προσβάσιμο. Εκτός από τη δημιουργία των υπότιτλων, ο υποτιτλιστής συνήθως προσδιορίζει στο λογισμικό του υπολογιστή τις ακριβείς θέσεις όπου κάθε υπότιτλος πρέπει να εμφανίζεται και να εξαφανίζεται. Το αποτέλεσμα είναι ένα αρχείο υποτίτλων που περιέχει τους πραγματικούς υπότιτλους και τους δείκτες θέσης που υποδεικνύουν πού πρέπει να εμφανίζεται και να εξαφανίζεται κάθε υπότιτλος. Αυτοί οι δείκτες βασίζονται συνήθως σε time formats και timestamps.

Οι υπότιτλοι μπορούν επίσης να δημιουργηθούν από άτομα που χρησιμοποιούν ελεύθερα διαθέσιμο λογισμικό δημιουργίας υποτίτλων όπως το Subtitle Workshop για Windows, το MovieCaptioner για Mac/Windows και το Subtitle Composer για Linux.

Ορισμένα προγράμματα και διαδικτυακό λογισμικό επιτρέπουν την δημιουργία αυτόματων υπότιτλων, κυρίως χρησιμοποιώντας speech-to-text μοντέλα.

Για παράδειγμα, στο YouTube, οι αυτόματοι υπότιτλοι είναι διαθέσιμοι στα Αγγλικά, Ολλανδικά, Γαλλικά, Γερμανικά, Ιταλικά, Ιαπωνικά, Κορεάτικα, Πορτογαλικά, Ρωσικά, Ινδονησιακά, Ισπανικά, Τουρκικά, Ουκρανικά και Βιετναμέζικα.

1.1.11 Προς ένα νέο Πιθανό μέλλον?

Με την ραγδαία ανάπτυξη που έχουν δει τεχνολογίες όπως το Artificial Intelligence (ChatGPT,Bard,Gemini etc…) και γενικότερα το Domain των LLM (Large Language Model), είναι πιο εύκολο από ποτέ να ξεκινήσει κάποιος την διαδικασία του υποτιτλισμού ακόμα και ερασιτεχνικά, χωρις κάποια επαγγελματική εκπαίδευση και εμπειρία. Εμφανίζονται συνεχώς νέες πλατφόρμες και λογισμικά, τα οποία μπορούν να βοηθήσουν έναν χρήστη να υποτιτλισει σε μια τεράστια γκάμα γλωσσών το περιεχόμενο του.

Στόχος της Διπλωματικής Εργασίας

Ο στόχος τον οποίο θέλει να πετύχει αυτή η Δ.Ε, είναι η δημιουργία μιας web εφαρμογής η οποία καλύπτει διάφορα features τα οποία λείπουν από άλλες μεγάλες πλατφόρμες υποτιτλισμού, κυρίως από την πλατφόρμα του YouTube Studio, μιας και ο στόχος της εφαρμογής ειναι η υποβοήθηση υποτιτλισμού για βίντεο στο YouTube. Παράδειγμα κάποιων χρήσιμων λειτουργιών που δεν είναι διαθέσιμες στο YouTube studio και προσφέρει η εφαρμογή που αναπτύχθηκε, είναι η γρήγορη δημιουργία ολων των χρονισμών των υποτίτλων, η απευθείας μετάφραση ενος υποτιτλου απο μια γλώσσα σε μια αλλη, το integration με το μοντέλο του ChatGPT, assignment των voice actor/ηθοποιών ανα διάλογο/υπότιτλο, Community Subtitling κ.α.

Κεφάλαιο 2ο: Τεχνολογίες Web Development που χρησιμοποιήθηκαν

Για το development της Web εφαρμογής χρησιμοποιήθηκε ενας συνδυασμός τεχνολογιών του Web framework οικοσυστήματος, το οποίο βρίσκεται σε ραγδαία εξέλιξη τα τελευταία χρόνια. Συγκεκριμένα για το Frontend κομμάτι της εφαρμογής, χρησιμοποιήθηκε το Angular Framework, ενα TypeScript-based framework το οποίο αναπτύσσεται από την Google. Για το Backend της εφαρμογής αξιοποιήθηκε το Firebase, η οποία είναι μια cloud πλατφόρμα της Google, που προσφέρει πολλαπλές υπηρεσίες για την ανάπτυξη μιας πλήρους Full-stack εφαρμογής, σε συνδυασμό με διάφορα διαθέσιμα API για την αξιοποίηση ήδη υπάρχον δεδομένων και υπηρεσιών (YouTube, Google, OpenAI etc.)

Angular

Η Angular είναι ένα open-source JavaScript framework, παλαιότερα γνωστό και ως AngularJS, γραμμένο σε TypeScript. Η Google το διατηρεί και πρωταρχικός σκοπός της είναι η ανάπτυξη single-page application (SPA). Ως framework, είναι γνωστό για την στιβαρή αρχιτεκτονική του, τις εκτεταμένες δυνατότητες του, και την δυνατότητα αποδοτικής ανάπτυξης λογισμικού.

2.1.1 Γιατι Angular?

Η JavaScript είναι η πιο διαδεδομένη και ευρέως χρησιμοποιούμενη scripting γλώσσα σε client-side web εφαρμογές. Ενσωματώνεται σε HTML documents και επιτρέπει την αλληλεπίδραση με μια ιστοσελίδα με πολλούς μοναδικούς τρόπους. Ως μια σχετικά εύκολη στην εκμάθηση γλώσσα με διάχυτη υποστήριξη, προτιμάται συχνα για την ανάπτυξη σύγχρονων εφαρμογών.

Είναι όμως η JavaScript ιδανική για την ανάπτυξη εφαρμογών που απαιτούν modularity, δυνατότητα testing και προάγουν την παραγωγικότητα του προγραμματιστή; Η απάντηση είναι πως συνήθως όχι.

Αυτές τις μέρες, έχουμε υπάρχει μια τεράστια ποικιλία από framework και βιβλιοθηκών που έχουν σχεδιαστεί για να παρέχουν εναλλακτικές λύσεις. Όσον αφορά την ανάπτυξη του front-end κομματιού μιας εφαρμογής, η Angular καταπολεμά πολλά, αν όχι όλα, τα προβλήματα που αντιμετωπίζουν οι προγραμματιστές όταν χρησιμοποιούν απλή JavaScript.

2.1.2 Typescript

Η TypeScript ορίζει ένα σύνολο τύπων στην JavaScript, το οποίο την μετατρέπει απο loosely-typed σε μια strongly-typed γλώσσα, το οποίο βοηθά τους προγραμματιστές να γράψουν κώδικα που είναι πιο κατανοητός και αναγνώσιμος.

Όλος ο κώδικας TypeScript μεταγλωττίζεται τελικα σε JavaScript και μπορεί να εκτελεστεί ομαλά σε οποιαδήποτε πλατφόρμα. Η TypeScript δεν είναι υποχρεωτική για την ανάπτυξη μιας εφαρμογής Angular. Ωστόσο, συνιστάται ιδιαίτερα καθώς προσφέρει καλύτερη συντακτική δομή — καθιστώντας παράλληλα το code base ευκολότερο στην κατανόηση και την διαχρονική διατήρηση και ανάπτυξη του.

2.1.3 Αρχιτεκτονική και Κύρια Χαρακτηριστικά

2.1.3.1 Components και Modules

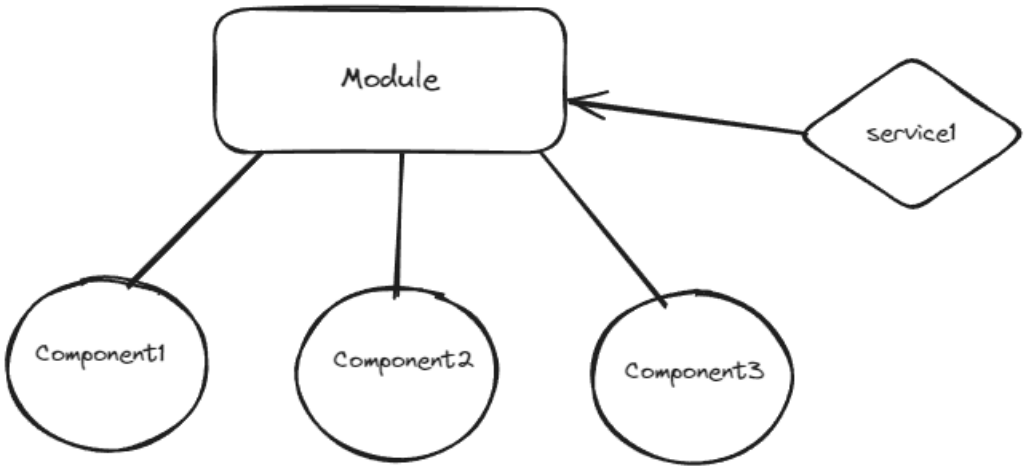
Η Angular βασίζεται στην έννοια των component [8], τα οποία είναι αυτόνομες, επαναχρησιμοποιήσιμες μονάδες που είναι υπεύθυνες για το χειρισμό συγκεκριμένων πτυχών του user interface. Τα component ενσωματώνουν τον κώδικα HTML, CSS και TypeScript που σχετίζεται με ένα συγκεκριμένο feature ή application logic. Κάθε component αποτελείται από:

- Ένα HTML template που δηλώνει τι αποτυπώνεται (rendered) στη σελίδα (*.component.html)
- Μια κλάση TypeScript που ορίζει τη συμπεριφορά (*.component.ts)
- Προαιρετικά, τα στυλ CSS εφαρμόζονται στο template (*.component.css)

Παρακάτω είναι ενα τυπικό component στο Angular Framework:

```
@Component({
  selector: 'app-dashboard',
  templateUrl: './dashboard.component.html',
  styleUrls: ['./dashboard.component.css'],
})
export class DashboardComponent implements OnInit {
  //code here
}
```

Τα modules, από την άλλη πλευρά, χρησιμεύουν ως δοχεία για την οργάνωση component, directive, services και άλλων δομικών block. Οι Angular εφαρμογές είναι συνήθως δομημένες ως μια συλλογή διασυνδεδεμένων module, κάθε ένα από τα οποία είναι υπεύθυνο για ένα ξεχωριστό χαρακτηριστικό ή λειτουργικότητα. Επίσης σημαντική είναι η έννοια των lazy loading module, που επιτρέπουν τη φόρτωση συγκεκριμένων τμημάτων μιας εφαρμογής κατά παραγγελία, συμβάλλοντας στην βέλτιστη απόδοση της εφαρμογής και γρηγορότερων loading times.



Σχήμα 2.1. Module Architecture

Τα NgModules διαμορφώνουν τον injector και τον compiler και βοηθούν στην οργάνωση σχετικών πραγμάτων μαζί. Ένα NgModule είναι ενα class που επισημαίνεται από τον decorator @NgModule. Το @NgModule παίρνει ένα αντικείμενο μεταδεδομένων που περιγράφει τον τρόπο compilation του template ενός component και τον τρόπο δημιουργίας ενός injector κατά το χρόνο εκτέλεσης.

Κάθε εφαρμογή Angular έχει τουλάχιστον ένα module, το root module. Για την εκτέλεση της εφαρμογής, πρέπει να γίνει bootstrap αυτο το module.

Το root module είναι το μόνο που χρειάζεται μια εφαρμογή με λίγα component. Καθώς η εφαρμογή μεγαλώνει, αναδιαμορφώνετε το root module σε feature modules, τα οποία είναι οργανωμένα ανάλογα με την λειτουργία που εκτελούν. Στη συνέχεια, αυτά τα feature modules εισάγωνται και αυτά στο Root module.

2.1.3.2 Change Detection

Η ανίχνευση αλλαγών (Change Detection) είναι η διαδικασία μέσω της οποίας h Angular ελέγχει εάν η κατάσταση της εφαρμογής έχει αλλάξει και εάν χρειάζεται ενημέρωση κάποιο σημείο στο DOM. Σε πιο high-level, η Angular διασχίζει τα component της εφαρμογής ένα προς ένα, αναζητώντας αλλαγές. Η Angular εκτελεί περιοδικά τον μηχανισμό ανίχνευσης αλλαγών, έτσι ώστε οι αλλαγές στο μοντέλο δεδομένων να αντικατοπτρίζονται στο view μιας εφαρμογής. Ο εντοπισμός αλλαγών μπορεί να ενεργοποιηθεί είτε manually είτε μέσω ενός ασύγχρονου συμβάντος (για παράδειγμα, μια αλληλεπίδραση χρήστη ή μια ολοκλήρωση XMLHttpRequest).

Η ανίχνευση αλλαγών είναι εξαιρετικά βελτιστοποιημένη και αποτελεσματική, αλλά μπορεί να προκαλέσει performance issues εάν η εφαρμογή την εκτελεί πολύ συχνά.

Η Angular χρησιμοποιεί μια “μονοκατευθυντική” ροή δεδομένων, που σημαίνει ότι οι αλλαγές στην κατάσταση της εφαρμογής διαδίδονται από τα δεδομένα του component προς το view template. Η διαδικασία του change detection περιλαμβάνει μια σειρά ελέγχων για τον εντοπισμό αλλαγών στα δεδομένα. Το framework χρησιμοποιεί στρατηγικές για τη βελτιστοποίηση της απόδοσης του change detection, όπως το default change detection, το on-push change detection και τη δυνατότητα εκτλεκεσης του change detection manually χρησιμοποιώντας το ChangeDetectorRef service.

Το change detection μπορεί να βελτιστοποιηθεί περαιτέρω αξιοποιώντας το Immutability και χρησιμοποιώντας το OnPush strategy. Οι immutable δομές δεδομένων βοηθούν την Angular να εντοπίζει τις αλλαγές πιο αποτελεσματικά, καθώς μπορεί να εντοπίσει γρήγορα τις αλλαγές συγκρίνοντας object references. Ο εντοπισμός αλλαγών OnPush, επιτρέπει στους προγραμματιστές να καθορίσουν ότι ένα component θα πρέπει να ελέγχεται για αλλαγές μόνο εάν αλλάξουν οι ιδιότητες Input του ή εάν ένας event listener υποδεικνύει μια πιθανή τροποποίηση.

2.1.3.3 Lifecycle & Hooks

Ένα component στην Angular έχει έναν κύκλο ζωής, μια σειρά από διαφορετικές φάσεις που περνά από την δημιουργία έως και την καταστροφή του. Μπορούμε να συνδεθούμε ή <<γατζωθούμε>>, εξού και η λέξη hook, σε αυτές τις διαφορετικές φάσεις για να αποκτήσουμε τον απόλυτο έλεγχο της εφαρμογής μας.

Για να γίνει αυτό, προσθέτουμε ορισμένες μεθόδους στην κλάση του component μας, οι οποίες καλούνται κατά τη διάρκεια καθεμιάς από αυτές τις φάσεις του κύκλου ζωής του component, αυτές οι μέθοδοι ονομάζονται hooks.

Αφού η εφαρμογή δημιουργήσει ένα component καλώντας τον constructor του, η Angular καλεί τις μεθόδους hook που κανει implement το component, στο κατάλληλο σημείο του κύκλου ζωής αυτου του component instance.

Η Angular εκτελεί τα παρακάτω βασικά hook methods που βλέπουν την πιο συχνή χρήση (στην πραγματικότητα εκτελούνται περισσότερα hooks) με την ακόλουθη σειρά [8]:

Πίνακας 2.1. Angular Lifecycle Hooks

Hook	Σκοπός	Run time
ngOnChanges()	Αντίδραση σε αλλαγές τιμής μεταβλητών. Η μέθοδος λαμβάνει ένα αντικείμενο SimpleChanges σαν argument με τις τρέχουσες και προηγούμενες τιμές των ιδιοτήτων.	Καλείται πριν το ngOnInit() hook (αν το component εχει data-bound properties/μεταβλητές) και κάθε φορά που αυτές αλλαζουν.
ngOnInit()	Initialization του component και αντίστοιχες ενέργειες που πρέπει να γίνουν με την δημιουργία του.	Καλείται μία φορά, μετά το πρώτο ngOnChanges(). Η ngOnInit() εξακολουθεί να καλείται ακόμα και όταν η ngOnChanges() δεν εκτελείται.
ngDoCheck()	Detection και αντίδραση σε αλλαγές που η Angular δεν μπορεί να κάνει catch/ αντιληφθεί από μόνη της.	Καλείται αμέσως μετά την ngOnChanges() σε κάθε εκτέλεση του change detection και αμέσως μετά την εκτέλεση ngOnInit(). Είναι ενα hook το οποίο εκτελείται πολύ συχνά και ειναι αρκετα resource-heavy.
ngOnDestroy()	Εκκαθάριση «σκουπιδιών» λίγο πριν η Angular καταστρέψει το component. Κατάργηση των subscription σε Observables για την αποφυγή των memory leak.	Καλείται αμέσως πριν η Angular καταστρέψει το component.

Για την αποτελεσματική ανταπόκριση σε διάφορα στάδια του κύκλου ζωής ενός component, μπορείτε να χρησιμοποιήσετε τα lifecycle hooks που παρέχονται από την βιβλιοθήκη Angular/core. Αυτές οι διεπαφές σάς δίνουν τη δυνατότητα να προβείτε σε συγκεκριμένες ενέργειες σε ένα component instance κατα τη δημιουργία, την ενημέρωση ή την καταστροφή του.

Κάθε interface περιγράφει τη δομή για μια hook μέθοδο, που προσδιορίζεται από ένα όνομα με πρόθεμα το "ng". Για παράδειγμα, το OnInit interface εισάγει μια μέθοδο hook που ονομάζεται ngOnInit(). Με την ενσωμάτωση αυτής της μεθόδου στο component, η Angular την καλεί αυτόματα λίγο μετά την πρώτη εξέταση των input properties για το συγκεκριμένο component.

```
@Component()
export class LifecycleComponent implements OnInit {
  constructor() { }
  // implement OnInit's `ngOnInit` method
  ngOnInit() {
    console.log('OnInit');
  }
}
```

Μόλις ολοκληρωθεί η δημιουργία του component, ο χρήστης θα δεί στην κονσόλα του browser το μήνυμα 'OnInit'.

2.1.3.4 Data Binding

Οπως αναφέρθηκε νωρίτερα η Angular βασίζεται στην συνύπαρξη και αλληλεπίδραση πολλών component οργανωμένα σε Module. Αυτο σημαινει οτι συνανταμε την έννοια των **Parent και Child components**, δηλαδή component τα οποία βρίσκονται εμφωλευμένα μεσα σε αλλα component και την σχέση μεταξύ τους.

Το Data-binding είναι από τα πιο βασικά χαρακτηριστικά της Angular που δημιουργεί μια ισχυρή σύνδεση μεταξύ των δεδομένων της εφαρμογής και του user interface, διασφαλίζοντας τον συγχρονισμό μεταξύ τους. Η Angular προσφέρει διάφορους τύπους data-binding, επιτρέποντας στους προγραμματιστές να δημιουργούν δυναμικές και διαδραστικές web εφαρμογές. Οι τρεις κύριοι τύποι data-binding είναι:

Property Binding: Το property-binding επιτρέπει τη δυναμική ρύθμιση ενός HTML element attribute με βάση τα δεδομένα του component. Επιτυγχάνεται δεσμεύοντας ενα HTML attribute σε μια μεταβλητή του component, χρησιμοποιώντας αγκύλες ([]). Για παράδειγμα:

```
<img [src]="imageUrl" alt="Angular Logo">
<button [disabled]="isButtonDisabled">Click me</button>
```

Εδώ, το **imageUrl** και το **isButtonDisabled** είναι μεταβλητές στο component και οι τιμές τους καθορίζουν το src attribute της εικόνας και το disabled attribute του button, αντίστοιχα.

Event Binding: Το event-binding επιτρέπει την εκτέλεση μεθόδων ως απόκριση στις διάφορες ενέργειες ενός χρήστη, όπως τα κλικ, πάτημα πλήκτρων ή άλλα custom event τα οποία μπορεί να δημιουργήσει ο developer. Συμβολίζεται με παρενθέσεις (()) και χρησιμοποιείται για τη σύνδεση μιας μεθόδου του component σε ένα συγκεκριμένο event στο template. Για παράδειγμα:

```
<button (click)="onButtonClick()">Click me</button>
```

Σε αυτήν την περίπτωση, η μέθοδος `onButtonClick` στο `component` θα καλείται κάθε φορα που κάνει ο χρήστης κλικ στο button.

Two-Way Data Binding: Το two-way data binding συνδυάζει το property και event binding για να βελτιστοποιήσει το συγχρονισμό μια μεταβλητή ενος component και ενός HTML element (input,select,textarea etc.). Υποδηλώνεται με την οδηγία [(ngModel)] και χρησιμοποιείται συχνά σε φόρμες για την αμφίδρομη ροή δεδομένων. Για παράδειγμα:

```
<input [(ngModel)]="userName" />
```

Εδώ, οι αλλαγές στο `input element` θα ενημερώσουν την μεταβλητή `username` στο `component` και το αντίστροφο.

Κάνοντας χρήση αυτων των τεχνικών επιτυγχάνεται και ενας απο τους τρόπους επικοινωνίας μεταξύ των διάφορων parent/child component. Πιο συγκεκριμένα χρησιμοποιώντας τους παραπάνω τρόπους:

Τα **@Input()** και **@Output() decorators** δίνουν σε ένα child και parent component εναν τρόπο επικοινωνίας μεταξύ τους. Το @Input() επιτρέπει σε ένα parent component να ενημερώνει δεδομένα στο child component. Αντίθετα, το @Output() επιτρέπει στο child component να στέλνει δεδομένα σε ένα parent component.

```
import { Component, Input } from '@angular/core'; // First, import Input
export class ItemDetailComponent {
  @Input() item = ''; // decorate the property with @Input()
}
```

Σε αυτήν την περίπτωση, το @Input() κανει decorate την ιδιότητα item, το οποίο είναι type string, ωστόσο, οι ιδιότητες @Input() μπορούν να έχουν οποιονδήποτε τύπο, όπως number, string, boolean ή object. Η τιμή για το item προέρχεται από το parent component.

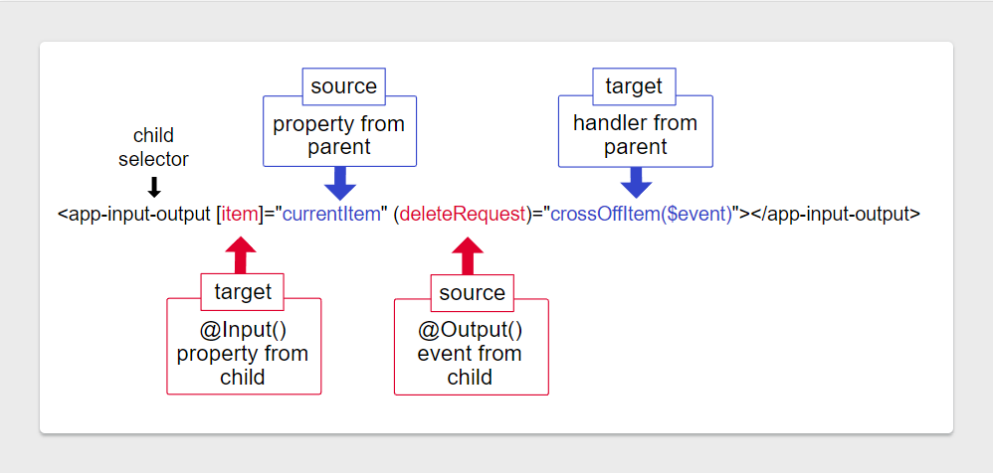
Το @Output() decorator επισημαίνει μια ιδιότητα σε ένα child component ως μια πόρτα μέσω της οποίας τα δεδομένα μπορούν να μεταφερθούν στο parent component.

Το child component χρησιμοποιεί την ιδιότητα @Output() για να δημιουργήσει ένα event για να ειδοποιήσει τον γονέα για κάποα αλλαγή. Για να γινει dispatch ένα event, μια @Output() ιδιότητα πρέπει να έχει τον τύπο EventEmitter, που είναι μια κλάση της Angular που χρησιμοποιείται για την εκπομπή custom event. Για παράδειγμα:

```
export class ItemOutputComponent {
  @Output() newItemEvent = new EventEmitter<string>();

  addNewItem(value: string) {
    this.newItemEvent.emit(value);
  }
}
```

Καθε φορα που τρέχει το addNewItem function (μέσω ενος event στο DOM ή οποιονδήποτε αλλο τρόπο), γίνεται emit το value που περνιέται σαν παράμετρος σε αυτο μέσω του custom event newItemEvent. Στην συνέχεια το parent component “ακούει” οτι αυτο το event εγινε trigger και τρέχει μια αντίστοιχη εντολή/function κάθε φορα εχοντας σαν πληροφορία το value που πέρασε το child component στην μορφη του $event.

```
<app-item-output (newItemEvent)="addItem($event)"></app-item-output>
```



Σχήμα 2.2. Input και Output Data flows ενός component

2.1.3.5 Services & Dependency Injection (DI)

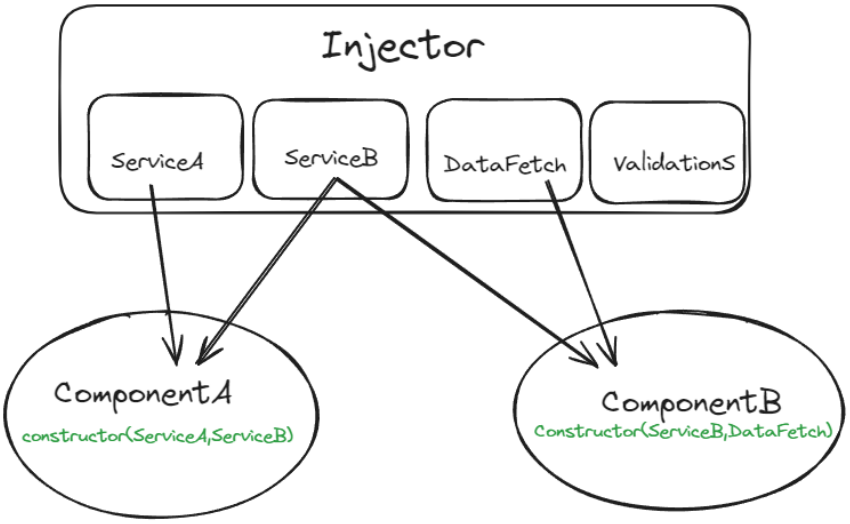
Ενα Service, στο πλαίσιο της Angular, περιλαμβάνει οποιαδήποτε τιμή, function ή χαρακτηριστικό είναι απαραίτητο για μια εφαρμογή. Συνήθως υλοποιείται ως μια κλάση με έναν συγκεκριμένο και καθορισμένο σκοπό, ενα service θα πρέπει να εκτελεί μία ή ένα σύνολο συγκεκριμένων εργασιών αποτελεσματικά.

Η Angular κάνει διάκριση μεταξύ ενός component και ενός service για τη βελτίωση της ευελιξίας και του reuseability των δυο αυτών κλασεων. Σε ένα ιδανικό σενάριο, ο πρωταρχικός ρόλος ενός component είναι να διευκολύνει και να βελτιώνει το user experience, προσφέροντας ιδιότητες και μεθόδους για τον συγχρονισμό και την διασύνδεση των δεδομένων μεταξύ του HTML template και την λογική που υλοποιεί το component.

Για εργασίες που δεν σχετίζονται με το view ή τη λογική της εφαρμογής, τα component θα πρέπει να αξιοποιούν τα services. Τα services διαπρέπουν στο χειρισμό λειτουργιών όπως η ανάκτηση δεδομένων από εναν server (fetch, http requests etc.), το user validation ή η απευθείας καταγραφή στην κονσόλα του browser. Ενσωματώνοντας τέτοιες εργασίες σε **Injectable service classes**, διασφαλίζετε η διαθεσιμότητά τους σε οποιοδήποτε component.

Η Angular δεν επιβάλλει αυτές τις αρχές, αλλά διευκολύνει την τήρηση τους παρέχοντας ένα πλαίσιο που καθιστά εύκολη την οργάνωση της λογικής που υλοποιεί εφαρμογή σε services. Μέσω του **dependency injection**, η Angular διασφαλίζει τη διαθεσιμότητα αυτών των υπηρεσιών σε components, προωθώντας μια modular και διατηρήσιμη δομή εφαρμογής.

Το **Dependency Injection (DI)** είναι το μέρος του Angular framework που παρέχει στα component πρόσβαση σε services και άλλους πόρους. Η Angular παρέχει τη δυνατότητα να εισάγετε ενα service σε ένα component, για να δώσετε σε αυτό το component πρόσβαση στην στις λειτουργίες του service.



Σχήμα 2.3. Angular Service Injector

Παράδειγμα ενος απλού service class και η χρήση του απο ενα component:

```
@Injectable({providedIn: 'root'})
export class HeroService {

  function funcA(): void {
    console.log('I'm a logger service');
  }

  function funcB(hero: Hero): void {
    console.log('Hello' + hero.name);
  }
}
```

Στην συνέχεια αυτο το service μπορεί να χρησιμοποιηθεί από **οποιοδήποτε** component της εφαρμογής κάνωντας initialize το service στον constructor του.

```
@Component
export class HeroComponent {
  constructor(private service: HeroService) { }

  ngOnInit(): void {
    let hero = new Hero({name:'Batman'});
    this.service.funcA();
    this.service.funcB(hero);
   }
}
```

2.1.3.6 Reactivity & Observables

Στην Angular, τα Observables είναι ένα κρίσιμο μέρος του reactive προγραμματισμού, ενός προγραμματιστικό πρότυπο που δίνει έμφαση στη διάδοση των αλλαγών που συμβαίνουν μέσω ροών δεδομένων. Τα Observables επιτρέπουν στους προγραμματιστές να χειρίζονται ασύγχρονα δεδομένα

και συμβάντα πιο αποτελεσματικά από τις παραδοσιακές τεχνικές όπως ειναι τα promises που συναντάμε στην απλή JavaScript.

Τα Observables είναι ένα ισχυρό εργαλείο για τη δημιουργία πολύπλοκων εφαρμογών που απαιτούν ενημερώσεις δεδομένων σε πραγματικό χρόνο και τον βελτιστό χειρισμό των event.

Τα Observables είναι ένα από τα βασικά χαρακτηριστικά της Angular. Ένα Observable είναι μια **ροή δεδομένων που μπορεί να παρατηρηθεί με την πάροδο του χρόνου**. Η Angular για την υλοποίηση των Observable interfaces χρησιμοποιεί natively την βιβλιοθήκη **RxJS** (Reactive Extensions for JavaScript), μια βιβλιοθήκη για reactive προγραμματισμό με χρήση Observable που διευκολύνει τη σύνθεση ασύγχρονου κώδικα ή κώδικα που βασίζεται σε callbacks.

Τα Observable είναι και αυτά μια δομή δεδομένων, αλλά με μερικές βασικές διαφορές:

- Τα Observable μπορούν να εκπέμπουν πολλαπλές τιμές με την πάροδο του χρόνου, ενώ οι πίνακες είναι για παράδειγμα είναι στατικοί και περιέχουν ένα σταθερό σύνολο τιμών.
- Τα Observable μπορούν να χειριστούν ασύγχρονες εργασίες όπως είναι το User input, HTTP requests και timers.
- Τα Observables μπορούν να συνδυαστούν, να μετασχηματιστούν και να συντεθούν με διάφορους τρόπους για να δημιουργήσουν πιο σύνθετες ροές δεδομένων.

Για να καταναλωθούν δεδομένα που εκπέμπονται από ένα observable, πρέπει να γίνει subscribe σε αυτο. Το subscribe σε ενα observable είναι παρόμοια με την εγγραφή ενός event listener και μας επιτρέπει να λαμβάνουμε και να χειριζόμαστε τιμές που εκπέμπονται από το observable. Παράδειγμα του πως γίνεται το subscribe σε ενα observable:

```
import { of } from 'rxjs';

const numbers = of(1, 2, 3);
numbers.subscribe(
 value => console.log(value),
 error => console.error(error),
 () => console.log('Completed')
);
```

Σε αυτό το παράδειγμα, δημιουργούμε ένα Observable χρησιμοποιώντας τον τελεστή of(), ο οποίος εκπέμπει ένα σταθερό σύνολο τιμών. Στη συνέχεια κανουμε subscribe στην Observable ιδιότητα numbers και παρέχουμε τρεις callback συναρτήσεις ως arguments:

- Η πρώτη συνάρτηση χειρίζεται κάθε τιμή που εκπέμπεται από το Observable και την καταγράφει στην κονσόλα.
- Η δεύτερη συνάρτηση χειρίζεται σφάλματα κατά τη ροή του Observable και τα καταγράφει στην κονσόλα.
- Η τρίτη συνάρτηση καλείται όταν το Observable ολοκληρωθεί και καταγράφει ένα μήνυμα ολοκλήρωσης στην κονσόλα.

Όταν εκτελεστεί αυτός ο κώδικας, θα πρέπει να εμφανιστούν οι τιμές 1, 2 και 3 καταγεγραμμένες στην κονσόλα, ακολουθούμενες από το μήνυμα ‘completed’.

Είναι σημαντικό να σημειωθεί ότι τα Observables είναι “lazy”, που σημαίνει ότι δεν εκπέμπουν τιμές μέχρι να γίνει subscribe σε αυτά. Επίσης ιδιαίτερα σημαντικό είναι η κατάργηση ενος subscription από ένα Observable, π.χ numbers.unsubscribe() στο παράδειγμα μας, όταν πλέον δεν χρειάζεται, για την αποτροπή memory leaks.

Μία από τις πιο συνηθισμένες περιπτώσεις χρήσης των Observables στην Angular είναι στα services, όπου μπορούν να χρησιμοποιηθούν για την ανάκτηση και τον χειρισμό δεδομένων από API, ή άλλες πηγές. Ακολουθεί ένα παράδειγμα μιας απλής υπηρεσίας που χρησιμοποιεί τα Observables για την ανάκτηση δεδομένων από ένα mock API:

```
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
 providedIn: 'root'
})
export class DataService {
 private apiUrl = '<https://jsonplaceholder.typicode.com/posts>';
  constructor(private http: HttpClient) {}
 getPosts(): Observable<any[]> {
   return this.http.get<any[]>(this.apiUrl);
 }
}
```

Σε αυτό το παράδειγμα, δημιουργούμε ένα service που ονομάζεται DataService που χρησιμοποιεί το ενσωματωμένο HttpClient της Angular για να υποβάλει ενα GET request σε ένα ψεύτικο API, το οποίο βρίσκεται στη διεύθυνση URL https://jsonplaceholder.typicode.com/posts. Ορίζουμε μια μέθοδο που ονομάζεται getPosts() που επιστρέφει ένα Observable τύπου any[], το οποίο αντιπροσωπεύει έναν πίνακα αντικειμένων JSON.

Για να χρησιμοποιήσουμε αυτήν την υπηρεσία σε ένα component, πρέπει απλώς να την εισάγουμε στον constructor του και να καλέσουμε τη μέθοδο getPosts():

```
export class AppComponent {
  posts: any[];
  constructor(private dataService: DataService) {}
  ngOnInit() {
    this.dataService.getPosts().subscribe(
      data => this.posts = data,
      error => console.error(error),
      () => console.log('Posts loaded')
      );
  }
}
```

Σε αυτο το component δημιουργούμε ενα instance του DataService στον constructor και στο ngOnInit hook καλούμε την μέθοδο getPosts() και κάνουμε subscribe για να ενημερωθούμε για την κατάσταση αυτής της ασύγχρονης διεργασίας(HTTP request). Όταν ολοκληρωθεί το Observable, ορίζουμε την

ιδιότητα posts του component με τα δεδομένα που ανακτήθηκαν από το GET request που έκανε το service νωρίτερα.

Τα Observables είναι ένα ισχυρό εργαλείο για τη διαχείριση της ασύγχρονης ροής δεδομένων μιας εφαρμογής. Με τη βοήθεια της βιβλιοθήκης RxJS, η δημιουργία και διαχείριση των Observable, για εργασίες όπως τα user inputs έως και τα HTTP request, γίνεται με μεγάλη ευελιξία και efficiency.

2.1.4 Angular CLI

Το Angular CLI είναι ένα command-line interface - εργαλείο εντολών που χρησιμοποιείται για την προετοιμασία, την ανάπτυξη, τη δημιουργία scaffolds και τη συντήρηση εφαρμογών Angular απευθείας από ένα command shell.

Το εργαλείο μπορεί να καλεστεί στο command line μέσω του εκτελέσιμου **ng**.

Για να δημιουργήσετε, να κανετε build και να τρέξετε ένα νέο Project στην Angular σε έναν local development server, μεταβείτε στο parent directory του χώρου εργασίας σας, και χρησιμοποιήστε τις ακόλουθες εντολές:

```
ng new my-project
cd my-project
ng serve
```

Στον browser, ανοίξτε το http://localhost:4200/ για να δείτε την εκτέλεση της εφαρμογής. Όταν χρησιμοποιείτε την εντολή **ng serve** για να δημιουργήσετε μια εφαρμογή και να την εξυπηρετήσετε τοπικά, ο server αναδημιουργεί αυτόματα την εφαρμογή και φορτώνει ξανά τη σελίδα όταν αλλάζετε οποιοδήποτε από τα source αρχεία.

Για να γίνει deploy μια Angular εφαρμογή σε εναν web server, αρκεί μονο ο developer να τρέξει την εντολη **ng build** στο CLI εντος του project, και αυτό θα παράξει ολα τα απαραίτητα αρχεία για να γίνει deploy μια web based εφαρμογή (index.html, css files, .js files etc…).

2.1.5 Testing

Οι περισσότεροι web developer σίγουρα θα συναντήσουν κάποια μορφή testing στην καριέρα τους. Ως developers, ως επιχειρηματίες ή οτιδήποτε άλλο, θέλουμε να γνωρίζουμε αν η εφαρμογή λειτουργεί για τον χρήστη ή τους πελάτες της. Επιτρέπει ο ιστότοπος στον χρήστη να ολοκληρώσει τις εργασίες του; Ο ιστότοπος εξακολουθεί να λειτουργεί μετά την εισαγωγή νέων λειτουργιών ή την ανακατασκευή εσωτερικών στοιχείων; Πώς αντιδρά ο ιστότοπος σε σφάλματα ή αποτυχία του συστήματος; Το testing δίνει απαντήσεις σε αυτές ακριβώς τις ερωτήσεις.

Όταν γράφετε tests, πρέπει να έχετε κατά νου τους στόχους της δοκιμής. Πρέπει να κρίνετε εάν ένα test είναι πολύτιμο σε σχέση με αυτούς τους στόχους.

Τα automated tests έχουν πολλά τεχνικά, οικονομικά και οργανωτικά οφέλη[9]. Τα automated tests προσπαθούν να λύσουν προβλήματα λογισμικού στην αρχή, προτού προκαλέσουν πραγματική ζημιά, όταν εξακολουθούν να είναι διαχειρίσιμα και υπό έλεγχο. Φυσικά, η διασφάλιση ποιότητας (QA) απαιτεί χρόνο και κοστίζει. Χρειάζεται όμως λιγότερο χρόνο και είναι φθηνότερο από το να αφήσετε σφάλματα να περάσουν σε ενα software release.

Όταν μια ελαττωματική εφαρμογή αποστέλλεται στον πελάτη, όταν οι χρήστες αντιμετωπίζουν σφάλματα, όταν τα δεδομένα μιας εφαρμογής χάνονται ή γίνονται corrupted, μπορεί να διακυβευτεί όλο το integrity της. Μετά από ένα τέτοιο περιστατικό, είναι ακριβό να αναλυθεί και να διορθωθεί το σφάλμα προκειμένου να ανακτηθεί η εμπιστοσύνη ενός χρήστη.

Η πολυπλοκότητα της Angular δεν μπορεί να γίνει κατανοητή χωρίς να ληφθεί υπόψη το automated testing. Γιατί μια εφαρμογή Angular είναι δομημένη σε Components, Services, Modules κ.λπ.; Γιατί τα μέρη είναι αλληλένδετα όπως είναι; Γιατί όλα τα μέρη μιας εφαρμογής Angular εφαρμόζουν τα ίδια μοτίβα;

Ένας σημαντικός λόγος είναι η δυνατότητα του testing. Η αρχιτεκτονική του Angular εγγυάται ότι όλα τα μέρη της εφαρμογής μπορούν να τεσταριστούν εύκολα με παρόμοιο τρόπο.

Η Angular παρέχει δυνατά εργαλεία δοκιμών απευθείας "out of the box". Όταν δημιουργείτε ένα project Angular χρησιμοποιώντας το CLI, συνοδεύεται από πλήρως λειτουργικες ρυθμίσεις για automated και end-to-end tests.

Η ομάδα της Angular έχει επιλέξει και χρησιμοποιεί natively το Jasmine ως το testing framework και το Karma ως test runner. Τα μέρη της εφαρμογής ελέγχονται συνήθως μέσα στο TestBed της Angular. Αυτή η ρύθμιση είναι μια αντιστάθμιση με δυνατά και αδύνατα σημεία. Δεδομένου ότι είναι μόνο ένας από τους πιθανούς τρόπους testing εφαρμογών Angular, μπορείτε να χρησιμοποιήσετε οποιοδήποτε αλλο testing toolset.

Ένα απλό unit test:

Ας υποθέσουμε ότι έχουμε ένα μικρό service - το UserService.

```
// user.service.ts
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class UserService {
  getUser(): string {
    return 'John Doe';
  }
}
```

Τώρα, μπορούμε να γράψουμε ενα unit test για αυτό το service χρησιμοποιώντας το Jasmine:

```
// user.service.spec.ts
import { TestBed } from '@angular/core/testing';
import { UserService } from './user.service';

describe('UserService', () => {
  let service: UserService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(UserService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });

  it('should return a user', () => {
    const user = service.getUser();
    expect(user).toBe('John Doe');
  });
});
```

Στην συνέχεια μέσω του Angular CLI, τρέχοντας το command - ng test εκτελείται το test και γίνεται launch το Karma test runner, το οποίο τρέχει στον browser και παρέχει feedback σχετικό με το test στο command line, δηλαδή μας ενημερώνει για το που και εαν υπάρχει κάποιο error στο test που μόλις έτρεξε.

Firebase

Το Firebase, μια application development πλατφόρμα της Google, χρησιμεύει ως μια εργαλειοθήκη για την ανάπτυξη, βελτίωση και επέκταση μιας εφαρμογής [10]. Παρέχει ένα ολοκληρωμένο σύνολο εργαλείων που ανακουφίζουν τους προγραμματιστές από το έργο της δημιουργίας διαφόρων υπηρεσιών που μπορεί να προτιμούν να μην χειρίζοντ αι ή δημιουργήσουν, επιτρέποντάς τους να επικεντρωθούν στη βελτίωση και περαιτέρω ανάπτυξη της εφαρμογής. Αυτές οι υπηρεσίες περιλαμβάνουν analytics, authentication, databases, διάφορα configurations, file storage, ακόμα και αποστολή push messages. Μιας και αυτές οι υπηρεσίες είναι cloud-hosted, κλιμακώνονται εύκολα χωρίς να απαιτούν σημαντικές προσπάθειες από τον προγραμματιστή ανάλογα με τις ανάγκες της εφαρμογής.

Όταν αναφερόμαστε στο "cloud-hosted", σημαίνει ότι η Google διατηρεί και διαχειρίζεται πλήρως το backend των προϊόντων Firebase. Τα client SDK από το Firebase αλληλεπιδρούν απευθείας με αυτά τα backend services, εξαλείφοντας την ανάγκη για οποιοδήποτε ενδιάμεσο λογισμικό (middleware) μεταξύ της εφαρμογής και των services. Στην περίπτωση χρήσης μιας βάσης δεδομένων στο Firebase, συνήθως ο κώδικας γράφετε απευθείας στην client εφαρμογή για να υποβληθούν queries στη βάση δεδομένων.

Αυτή η προσέγγιση διαφέρει από την παραδοσιακή ανάπτυξη εφαρμογών, όπου συνήθως γράφονται τόσο λογισμικό frontend όσο και backend. Στην παραδοσιακή ανάπτυξη μιας εφαρμογής, το frontend καλεί τα τελικά API endpoint που εκτίθενται από το backend, και το backend εκτελεί τις πραγματικές διεργασίες. Ωστόσο, τα προϊόντα Firebase παρακάμπτουν το παραδοσιακό backend, τοποθετώντας τον

φόρτο εργασίας στον client. Για αυτό τον λόγο, το Firebase χαρακτηρίζεται ως “Platform as a Service” ή “Backend as a Service”.

Πραγματικά δεν υπάρχει όριο στους τύπους εφαρμογών που μπορούν να βοηθηθούν από τα προϊόντα Firebase. Το iOS, το Android και οι Web εφαρμογές είναι οι κύριοι στόχοι για τα Firebase SDK και υπάρχει αυξανόμενη υποστήριξη για Unity και C++.

2.1.6 Authentication

Η αναγνώριση χρήστη (User Identification) είναι μια κρίσιμη πτυχή για πολλές εφαρμογές. Η κατανόηση της ταυτότητας ενός χρήστη επιτρέπει την ασφαλή αποθήκευση των δεδομένων του χρήστη στο cloud και διασφαλίζει μια συνεπή εξατομικευμένη εμπειρία σε όλες τις συσκευές του.

Το Firebase Authentication προσφέρει backend υπηρεσιες, φιλικά προς τον χρήστη SDK και pre-built UI βιβλιοθήκες που έχουν σχεδιαστεί για τον έλεγχο ταυτότητας των χρηστών για μια εφαρμογή. Διευκολύνει τον έλεγχο ταυτότητας μέσω διαφόρων μεθόδων, συμπεριλαμβανομένων κωδικών πρόσβασης, αριθμών τηλεφώνου και δημοφιλών “federated” παρόχων ταυτότητας όπως το Google, το Facebook και το Twitter.

Για να κάνει sign-in ενας χρήστης στην εφαρμογή, αρχικά λαμβάνονται τα credentials από τον χρήστη. Αυτά τα διαπιστευτήρια μπορεί να περιλαμβάνουν τη διεύθυνση email και τον κωδικό πρόσβασης του χρήστη ή ένα διακριτικό OAuth που λαμβάνεται από έναν federated πάροχο ταυτότητας (Facebook, Google, Twitter etc.). Στη συνέχεια, τα ληφθέντα διαπιστευτήρια προωθούνται στο Firebase Authentication SDK, όπου οι backend υπηρεσίες τα επαληθεύουν και στέλνουν μια απάντηση στον client.

Σχήμα 2.4. Firebase Federated Sign-in providers

Μετά από μια επιτυχημένη είσοδο, η εφαρμογή αποκτά πρόσβαση στις βασικές πληροφορίες του χρήστη και σε δεδομένα που είναι αποθηκευμένα σε άλλα Firebase SDK.

Όταν ένας χρήστης εγγραφεί ή συνδεθεί, αυτός ο χρήστης ορίζεται ως ο τρέχων χρήστης του current Auth instance. Αυτο το instance διατηρεί την κατάσταση του χρήστη, διασφαλίζοντας ότι η ανανέωση της σελίδας (σε browser) ή η επανεκκίνηση της εφαρμογής δεν έχει ως αποτέλεσμα την απώλεια των πληροφοριών του χρήστη.

Κατά την αποσύνδεση, το στιγμιότυπο Auth παύει να διατηρεί μια αναφορά στο αντικείμενο χρήστη και δεν διατηρεί πλέον την κατάστασή του, ως εκ τούτου, δεν υπάρχει τρέχων χρήστης.

Ο συνιστώμενος τρόπος παρακολούθησης της τρέχουσας κατάστασης του Auth instance είναι η χρήση listener (ονομάζονται επίσης "observers" στην JavaScript/Angular). Ένας Auth listener/observer

ειδοποιείται κάθε φορά που συμβαίνει κάτι σχετικό με το αντικείμενο Auth (sign-in, sign-out, token refresh etc.) και μπορεί να αντιδρά ανάλογα.

2.1.7 Firestore

Το Firestore αποτελεί μια ευέλικτη και επεκτάσιμη βάση δεδομένων που έχει σχεδιαστεί για την ανάπτυξη web και mobile εφαρμογών εντός του Firebase και Google Cloud οικοσυστήματος. Η βάση δεδομένων Firestore, διασφαλίζει τον συγχρονισμό των δεδομένων μεταξύ των client σε πραγματικό χρόνο μέσω των listener.

Το Firestore, είναι μια **NoSQL document-oriented database** που φιλοξενείται στο cloud και σε αντίθεση με μια βάση δεδομένων SQL, δεν υπάρχουν πίνακες ή σειρές. Τηρώντας το NoSQL μοντέλο δεδομένων του Cloud Firestore, οι πληροφορίες αποθηκεύονται σε documents με key - value ζευγάρια. Αυτά τα document είναι οργανωμένα σε collections, λειτουργώντας ως δοχεία για την οργάνωση δεδομένων και την κατασκευή query.

Υποστηρίζοντας μια ποικιλία τύπων δεδομένων, από απλά string και numbers έως περίπλοκα nested αντικείμενα, τα document επιτρέπουν τη δημιουργία subcollection και ιεραρχικών δομών δεδομένων που κλιμακώνονται ανάλογα με το μέγεθος της βάσης δεδομένων.

Όσον αφορά τα queries, το Firestore προσφέρει αποτελεσματικές και ευέλικτες επιλογές. Οι χρήστες μπορούν να δημιουργήσουν shallow ερωτήματα για να ανακτήσουν δεδομένα σε document επίπεδο χωρίς να ανακτήσουν ολόκληρη το collection ή τυχόν εμφωλευμένα subcollection. Η ενσωμάτωση των ordering, filtering και limit στα queries επιτρέπει την εύκολη σελιδοποίηση (pagination) των αποτελεσμάτων στον client.

2.1.7.1 Data Model

Στο Cloud Firestore, η μονάδα αποθήκευσης είναι το document. Ένα document είναι μια ελαφριά εγγραφή που περιέχει πεδία, τα οποία αντιστοιχίζονται σε τιμές. Κάθε document προσδιορίζεται με ένα όνομα.Ένα έγγραφο που αντιπροσωπεύει ένα χρήστη savvassal μοιάζει κάπως έτσι:

```
firstName : "Savvas"
lastName : "Salampasis"
born : 1999
```

Τα σύνθετα, nested αντικείμενα σε ένα document ονομάζονται maps. Για παράδειγμα, θα μπορούσατε να δομήσετε το όνομα του χρήστη από το παραπάνω παράδειγμα με ένα map, ως εξής:

```
name :
      first : "Savvas"
      last : "Salampasis"
born : 1999
```

Μπορεί να παρατηρήσετε ότι τα document μοιάζουν πολύ με JSON αντικείμενα. Στην πραγματικότητα είναι ακριβώς αυτο. Υπάρχουν ορισμένες διαφορές (για παράδειγμα, τα document υποστηρίζουν επιπλέον τύπους δεδομένων και έχουν περιορισμένο μέγεθος σε 1 MB), αλλά γενικά, μπορείτε να αντιμετωπίζετε τα document ως lightweight εγγραφές JSON.

Τα document ζουν σε collections, τα οποία είναι απλώς δοχεία για documents. Για παράδειγμα, θα μπορούσε υπάρχει ένα collection χρηστών (με όνομα - *users*) που να περιέχει τους διάφορους χρήστες μιας εφαρμογής, καθένας από τους οποίους αντιπροσωπεύεται από ένα document:

Το Firestore είναι τελείως “schemaless”, επομένως δίνει πλήρη ελευθερία ως προς τα πεδία που μπαίνουν σε κάθε document και τους τύπους δεδομένων που αποθηκεύονται σε αυτά τα πεδία. Τα document εντός του ίδιου collection μπορούν όλα να περιέχουν διαφορετικά πεδία ή να αποθηκεύουν διαφορετικούς τύπους δεδομένων σε αυτά τα πεδία. Ωστόσο, προτείνεται η χρήση των ίδιων πεδίων και τους ίδιους τύπους δεδομένων στα document, ώστε να μπορεί να γίνει αναζήτηση σε αυτά πιο εύκολα.

2.1.7.2 Security Rules

Οι κανόνες ασφαλείας (Security Rules) του Firestore επιτρέπουν τη ρύθμιση της πρόσβασης σε έγγραφα και συλλογές σε μια βάση δεδομένων. Η προσαρμόσιμη σύνταξη αυτών των κανόνων επιτρέπει τη δημιουργία κανονισμών που περιλαμβάνουν διάφορα σενάρια, που κυμαίνονται από τον έλεγχο όλων των εγγραφών σε ολόκληρη τη βάση δεδομένων έως τον καθορισμό λειτουργιών σε ένα συγκεκριμένο έγγραφο.

Η ευθύνη της δημιουργίας και της επίβλεψης των Κανόνων Ασφαλείας του Cloud Firestore έγκειται στην προσαρμογή τους σύμφωνα με το μοντέλο δεδομένων που έχει καθιερωθεί τόσο για την προεπιλεγμένη βάση δεδομένων όσο και για τυχόν πρόσθετες βάσεις δεδομένων εντός του project.

Κάθε σύνολο κανόνων ασφαλείας του Cloud Firestore περιλαμβάνει match statements, τα οποία αντιστοιχούν σε document εντός της βάσης δεδομένων, και επιτρέπουν (allow) εκφράσεις που διέπουν την πρόσβαση σε αυτά τα document. Για παράδειγμα:

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /<some_path>/ {
      allow read, write: if <some_condition>;
    }
  }
}
```

Κάθε αίτημα στην βάση δεδομένων απο εναν client αξιολογείται σύμφωνα με τους κανόνες ασφαλείας πριν από την ανάγνωση ή την εγγραφή οποιωνδήποτε δεδομένων. Εάν οι κανόνες απαγορεύουν την πρόσβαση σε οποιαδήποτε από τις καθορισμένες διαδρομές του document, ολόκληρο το αίτημα αποτυγχάνει.

2.1.8 Storage

Το Cloud Storage for Firebase σάς επιτρέπει να ανεβάζετε και να μοιράζεστε user-generated content, όπως εικόνες, βίντεο, text files κ.α. Τα δεδομένα αποθηκεύονται σε ένα Google Cloud Storage **bucket** — μια λύση αποθήκευσης αντικειμένων σε κλίμακα exabyte με υψηλή διαθεσιμότητα. Το Cloud Storage επιτρέπει το ανέβασμα αυτών των αρχείων με ασφάλεια απευθείας από κινητές συσκευές και web browsers.

Τα αρχεία σας αποθηκεύονται σε έναν Cloud Storage bucket. Τα αρχεία σε αυτό το bucket παρουσιάζονται σε μια ιεραρχική δομή, όπως το σύστημα αρχείων στον τοπικό σας σκληρό δίσκο. Δημιουργώντας ενα reference σε ένα αρχείο, η εφαρμογή σας αποκτά πρόσβαση σε αυτό. Αυτά τα reference μπορούν στη συνέχεια να χρησιμοποιηθούν για τη μεταφόρτωση, λήψη δεδομένων, την

ενημέρωση metadata ή τη διαγραφή του αρχείου. Ενα reference μπορεί να κάνει point σε ένα συγκεκριμένο αρχείο είτε σε έναν κόμβο υψηλότερου επιπέδου στην ιεραρχία.

Για να δημιουργήσετε ενα reference, λάβετε ενα instance του Storage service χρησιμοποιώντας την getStorage() και στη συνέχεια καλέστε την ref() με το service ως argument. Αυτό το reference δείχνει τη ρίζα του bucket σας στο Cloud Storage.

```
import { getStorage, ref } from "firebase/storage";

// Get a reference to the storage service, which is used to create references in your storage bucket

const storage = getStorage();

// Create a storage reference from our storage service

const storageRef = ref(storage);
```

Μπορείτε να δημιουργήσετε ενα reference σε μια τοποθεσία χαμηλότερα στην ιεραρχία/tree, για παράδειγμα "subtitles/greek.sbv" περνώντας αυτό το path ως argument όταν καλείτε την ref().

```
import { getStorage, ref } from "firebase/storage";

const storage = getStorage();

// Create a child reference

Const subtitlesRef = ref(storage, 'subtitles');

// subtitlesRef now points to 'images'

// Child references can also take paths delimited by '/'

const greekSubRef = ref(storage, subtitles/greek.sbv');

// greekSubRef  now points to "subtitles/greek.sbv"

// subtitlesRef still points to "subtitles"
```

Έχοντας πλέον reference σε κάποιο file, μπορούν να πραγματοποιηθούν ολες οι συνηθισμένες εργασίες που μπορεί να γίνουν σε αυτό, οπως edit, delete, download,upload κ.α.

YouTube iFrame API

Το IFrame API του YouTube είναι μια διεπαφή προγραμματισμού που επιτρέπει στους προγραμματιστές να ενσωματώνουν και να ελέγχουν απρόσκοπτα ενα YouTube video σε ιστοσελίδες. Χρησιμοποιώντας ένα ενσωματωμένο iframe element, το API παρέχει έναν τυποποιημένο τρόπο αλληλεπίδρασης με τον video player του YouTube. Οι προγραμματιστές μπορούν να φορτώνουν βίντεο, να προσαρμόζουν τις ρυθμίσεις του video player και να λαμβάνουν ειδοποιήσεις για event όπως αλλαγές στην κατάσταση (play/pause κ.α) αναπαραγωγής ή σφάλματα. Το API υποστηρίζει τόσο JavaScript όσο και HTML5, καθιστώντας το ευέλικτο για διάφορα σενάρια.

Αξιοποιώντας το YouTube Iframe API, οι προγραμματιστές μπορούν να ενσωματώσουν απρόσκοπτα περιεχόμενο βίντεο στις εφαρμογές τους και να ενισχύσουν την εμπειρία των χρηστών με την τεράστια αποθήκη βίντεο του YouTube.

Το να γίνει embed ενα YouTube video σε κάποια σελίδα είναι μια αρκετά απλή εργασία:

Αρχικά, πρέπει να συμπεριλάβετε το API στο source tag του αρχείο HTML:

```
<!-- Include the YouTube Iframe API script -->
<script src="https://www.youtube.com/iframe_api"></script>
```

Στη συνέχεια, δημιουργήστε ένα `container element` όπου θέλετε να ενσωματώσετε το βίντεο:

```
<!-- Create a container for the YouTube player -->
<div id="player"></div>
```

Στην συνέχεια, χρησιμοποιώντας JavaScript μπορείτε να κάνετε initialize τον video player για οποιοδήποτε video, ξέροντας απλα το ID του και περνώντας στο videoId field του YT.Player object.

```
// Initialize the player when the API script is loaded
function onYouTubeIframeAPIReady() {
  // Create a new YouTube player
  const player = new YT.Player('player', {
    height: '360',
    width: '640',
    videoId: 'your-video-id',
    playerVars: {
      autoplay: 1,
      controls: 1,
      showinfo: 0,
      rel: 0,
      modestbranding: 1
    },
    events: {
      'onReady': onPlayerReady,
      'onStateChange': onPlayerStateChange
    }
  });
}
```

Επίσης μπορείτε να ορίσετε και κάποια function τα οποία θα τρέχουν όταν συμβαίνουν ορισμένα event, οπως είναι το onStateChange και onReady.

```
// Callback function when the player is ready
function onPlayerReady(event) {
  // You can perform additional actions when the player is ready
  // For example, event.target.playVideo();
}

// Callback function when the player state changes
function onPlayerStateChange(event) {
  // You can handle different player states here
  // For example, pause the video when it ends
  if (event.data === YT.PlayerState.ENDED) {
    event.target.pauseVideo();
  }
}
```

Git

Το Git είναι ένα κατανεμημένο σύστημα version control που επιτρέπει στους προγραμματιστές να παρακολουθούν τις αλλαγές στο codebase τους, να συνεργάζονται απρόσκοπτα και να διαχειρίζονται διαφορετικές εκδόσεις των έργων τους [11]. Δημιουργήθηκε από τον Linus Torvalds το 2005, και πλέον το Git έχει γίνει το de facto πρότυπο για τον version control στην ανάπτυξη λογισμικού λόγω της ευελιξίας, της ταχύτητας και της αποτελεσματικότητάς του. Επιτρέπει στους προγραμματιστές να εργάζονται σε έργα συλλογικά παρέχοντας έναν τρόπο συγχώνευσης αλλαγών από πολλούς συνεισφέροντες, παρακολούθησης του ιστορικού των τροποποιήσεων κώδικα και εύκολης επαναφοράς σε προηγούμενες καταστάσεις εάν είναι απαραίτητο.

Το GitHub το οποίο χρησιμοποιήθηκε για το versioning της εφαρμογής, είναι μια διαδικτυακή πλατφόρμα που βασίζεται στο Git και παρέχει πρόσθετες δυνατότητες συνεργασίας και hosting [12]. Το GitHub, το οποίο ιδρύθηκε το 2008, επιτρέπει στους προγραμματιστές να αποθηκεύουν τα αποθετήρια Git τους στο cloud, διευκολύνοντας την συνεργασία μεταξύ των μελών μιας ομάδας ανεξάρτητα από τη φυσική τους τοποθεσία. Το GitHub περιλαμβάνει λειτουργίες όπως παρακολούθηση προβλημάτων, pull requests και wikis, βελτιώνοντας τη διαχείριση έργων και την επικοινωνία. Λειτουργεί επίσης ως κεντρικός κόμβος για open source repos, ενισχύοντας μια προσέγγιση με γνώμονα την κοινότητα στην ανάπτυξη λογισμικού.

Οι προγραμματιστές χρησιμοποιούν το Git τοπικά στους υπολογιστές τους για να διαχειριστούν τις αλλαγές και στη συνέχεια να προωθήσουν αυτές τις αλλαγές σε ένα απομακρυσμένο αποθετήριο που φιλοξενείται σε πλατφόρμες όπως το GitHub. Το GitHub έχει γίνει ένα κρίσιμο εργαλείο τόσο για έργα ανοιχτού κώδικα όσο και για ιδιωτικά έργα, παρέχοντας μια φιλική προς το χρήστη διεπαφή, παρακολούθηση των issues και του code collaboration. Ο συνδυασμός Git και GitHub έχει εξορθολογίσει σημαντικά τη διαδικασία ανάπτυξης λογισμικού, ενισχύοντας τη συνεργασία, τη διαφάνεια και την αποτελεσματικότητα στον κόσμο του προγραμματισμού.

2.1.9 GitHub Pages

Οι GitHub Pages είναι μια ισχυρήδυνατότητα που παρέχεται από το GitHub και επιτρέπει στους προγραμματιστές να φιλοξενούν και να δημοσιεύουν αβίαστα στατικές ιστοσελίδες απευθείας από τα αποθετήρια GitHub τους. Το GitHub Pages που κυκλοφόρησε το 2008 έχει γίνει μια δημοφιλής επιλογή για φιλοξενία προσωπικών ιστολογίων, projects και οποιουδήποτε άλλου στατικού περιεχομένου. Το feature αυτό χρησιμοποιήθηκε για το deployment της εφαρμογής στο public domain του GitHub. Παρακάτω είναι κάποια βασικά χαρακτηριστικά των GitHub Pages:

- Δωρεάν hosting: Το GitHub Pages προσφέρει δωρεάν φιλοξενία για στατικούς ιστότοπους, καθιστώντας το μια εξαιρετική επιλογή για προσωπικά project, και ιστότοπους μικρής κλίμακας. Οι χρήστες μπορούν να επωφεληθούν από αυτήν την υπηρεσία χωρίς να επιβαρυνθούν με επιπλέον κόστος.

- Ενσωμάτωση με το Git: Οι Σελίδες GitHub αξιοποιούν το υποκείμενο version control σύστημα Git, επιτρέποντας στους χρήστες να δημοσιεύουν τους ιστότοπούς τους κάνοντας "push" απλώς τον κώδικά τους σε ενα dedicated branch, που συνήθως ονομάζεται gh-pages ή main.

- Custom domains: Οι χρήστες μπορούν να συσχετίσουν ενα custom domain με τον ιστότοπο του GitHub Pages, παρέχοντας μια επαγγελματική και επώνυμη διεύθυνση ιστού. Αυτή η δυνατότητα επιτρέπει στους προγραμματιστές να χρησιμοποιούν τα δικά τους domain name αντί να βασίζονται στα προεπιλεγμένα subdomain του GitHub.

- Automatic Builds: Το GitHub Pages δημιουργεί αυτόματα και ενημερώνει τον δημοσιευμένο ιστότοπο κάθε φορά που οι αλλαγές γίνεται push στο σχετικό repository. Αυτό διασφαλίζει ότι η πιο πρόσφατη έκδοση του ιστότοπου είναι πάντα διαθέσιμη στους επισκέπτες χωρίς να χρειάζεται manual παρέμβαση απο τον maintainer του project.

Το GitHub Pages απλοποιεί τη διαδικασία φιλοξενίας στατικών ιστοτόπων, προσφέροντας μια λύση χωρίς προβλήματα με την ενσωμάτωσή του στο Git και την υποστήριξη προσαρμοσμένων τομέων. Είτε είστε προγραμματιστής, σχεδιαστής ή δημιουργός περιεχομένου, οι Σελίδες GitHub παρέχουν έναν απλό και αποτελεσματικό τρόπο για να μοιραστείτε την εργασία σας με τον κόσμο.

Κεφάλαιο 3ο: Γλωσσικές Τεχνολογίες/API που Χρησιμοποιήθηκαν

Google Translate API

Ως μία από τις πολλές μηχανές μετάφρασης, το Google Translate είναι μια πολύ δημοφιλής μηχανή και χρησιμοποιείται από ανθρώπους σε όλο τον κόσμο. Παρέχεται από την Google, που λειτουργεί ως μηχανή μετάφρασης που μεταφράζει τμήματα ενός κειμένου από μια γλώσσα σε άλλη. Κυκλοφόρησε το 2007 και βασίστηκε σε στατιστικά μοντέλα. Όσον αφορά την ποιότητα της απόδοσής του, το Google Translate αναγνωρίζει στον ιστότοπό του ότι ακόμη και η πιο εξελιγμένη μηχανή μετάφρασης δεν μπορεί ακόμη να προσεγγίσει την ποιότητα γλώσσας ενός φυσικού ομιλητή ή έχει ακόμη τις δεξιότητες ενός επαγγελματία μεταφραστή. Έτσι, το Google Translate αναφέρει επίσης ότι μπορεί να χρειαστεί πολύς χρόνος για να μπορέσουν να προσφέρουν μεταφράσεις με ποιότητα ανθρώπινης μετάφρασης. Είναι σε θέση να μεταφράζει λέξεις από εκατοντάδες γλώσσες και να μεταφράζει, όχι μόνο λέξη με λέξη, αλλά και πρόταση σε πρόταση ή ακόμα και παραγράφους και έγγραφα. Το Google Translate προσφέρει επίσης τη μετάφραση τυπωμένων λέξεων σε χαρτί, φωτογραφίζοντας το έγγραφο.

Η δυνατότητα μετάφρασης κειμένου έχει εξελιχθεί σημαντικά, καθώς τόσο το Google Translate όσο και το Cloud Translation API παρέχουν λύσεις για διαφορετικές ανάγκες. Το Google Translate, ως δημοφιλής μηχανή μετάφρασης, διακρίνεται για τη διαθεσιμότητα σε πολλές γλώσσες και τη δυνατότητα μετάφρασης προτάσεων και εγγράφων

Από την άλλη πλευρά, το Cloud Translation API διευρύνει τη δυνατότητα μετάφρασης σε δυναμικό επίπεδο, ανοίγοντας νέες προοπτικές για ιστότοπους και εφαρμογές. Χρησιμοποιώντας προεκπαιδευμένα ή προσαρμοσμένα μοντέλα μηχανικής μάθησης, το Cloud Translation ενισχύει την ακρίβεια των μεταφράσεων. Η συνεχής ενημέρωση του προεκπαιδευμένου μοντέλου, όπως του Neural Machine Translation (NMT), από την Google εξασφαλίζει ότι η ποιότητα παραμένει στο ύψος των προσδοκιών, αντικατοπτρίζοντας την εξέλιξη στον τομέα της γλωσσικής τεχνολογίας.

Το Cloud Translation επιτρέπει στους ιστότοπους και τις εφαρμογές να μεταφράζουν δυναμικά κείμενο μέσω ενός API. Το Cloud Translation χρησιμοποιεί ένα προεκπαιδευμένο από την Google ή ένα προσαρμοσμένο μοντέλο μηχανικής μάθησης για τη μετάφραση κειμένου. By default, το Cloud Translation χρησιμοποιεί ένα προεκπαιδευμένο μοντέλο της Google, το Neural Machine Translation (NMT), το οποίο ενημερώνει σε συχνό ρυθμό όταν διατίθενται περισσότερα δεδομένα εκπαίδευσης ή καλύτερες τεχνικές.

Η ενσωμάτωση του Cloud Translation API σε ιστότοπους και εφαρμογές αντιπροσωπεύει μια κομβική πρόοδο στον τομέα της δυναμικής μετάφρασης κειμένου. Αυτό το API προσφέρει μια ευέλικτη λύση, επιτρέποντας στους προγραμματιστές να ενσωματώνουν λειτουργίες μετάφρασης στις πλατφόρμες τους. Αξιοποιώντας ένα εκτεταμένο μοντέλο μηχανικής μάθησης, το Cloud Translation διασφαλίζει την απρόσκοπτη μετατροπή του κειμένου σε πολλές γλώσσες.

Καθώς το ψηφιακό τοπίο γίνεται ολοένα και πιο διασυνδεδεμένο και παγκοσμιοποιημένο, η ενσωμάτωση του Cloud Translation API αναδεικνύεται ως ένα κρίσιμο εργαλείο για εναν προγραμματιστή. Η ικανότητά του να μεταφράζει δυναμικά κείμενο μέσω προγραμματιστικών μέσων όχι μόνο αντικατοπτρίζει την πρόοδο στη μηχανική μάθηση, αλλά υπογραμμίζει επίσης τη σημασία της προώθησης της διαπολιτισμικής επικοινωνίας και της προσβασιμότητας στον ψηφιακό χώρο.

3.1.1 Χρήση του API

Για να μεταφράσετε κάποιο κείμενο, πρεπει να γίνει ένα αίτημα POST και να δοθεί ένα JSON στο σώμα του αιτήματος που προσδιορίζει τη γλώσσα προς μετάφραση (target) και το κείμενο προς μετάφραση (q). Μπορείτε να παρέχετε πολλά τμήματα κειμένου για μετάφραση, συμπεριλαμβάνοντας πολλαπλά πεδία q ή μια λίστα τιμών για το πεδίο q. Καθορίζετε τις γλώσσες - target χρησιμοποιώντας τους κωδικούς ISO-639 τους.

HTTP method and URL:

POST https://translation.googleapis.com/language/translate/v2

Request JSON body:

```
{
  "q": ["Hello world", "My name is Savvas Salampasis!"],
  "target": "de"
}
```

Στην συνέχεια εκτελόντας αυτο το HTTP request (curl, PowerShell, XMLHttpRequest etc.) θα λάβετε αυτό το response:

```
"data": {
  "translations": [
    {
      "translatedText": "Hallo Welt",
      "detectedSourceLanguage": "en"
    },
    {
      "translatedText": "Mein Name ist Savvas Salampasis",
      "detectedSourceLanguage": "en"
    }
  ]
}
```

Ο πίνακας translations περιέχει δύο πεδία μεταφρασμένου κειμένου με μεταφράσεις που παρέχονται στη γλώσσα-target που ζητήθηκε απο το API (de: Γερμανικά). Παρακάτω παρατίθεται παράδειγμα απο την εφαρμογή, του πως χρησιμοποιείται πρακτικα το API στην εφαρμογή για την μετάφραση μιας πρότασης ενος βίντεο:

```
translateSingleSubtitle(targetLanguage: {lang: string, id: number}): void {
 let translationObject: GoogleTranslateRequestObject =
{
  q: [this.form.get(targetLanguage.id + '-dialogBox').get('subtitles').value],
  target: targetLanguage.lang
};

this.google.translate(translationObject).subscribe(
(response: ResponseObject) => {
 if (response) {
   this.form.get(targetLanguage.id + '-
dialogBox').get('subtitles').setValue(response.data.translations[0].translated
Text);
 }})

}
```

ChatGPT

Από την πρώτη στιγμή της κυκλοφορίας του τον Νοέμβριο του 2022, το ChatGPT έχει συγκεντρώσει την παγκόσμια προσοχή ως ένα καινοτόμο chatbot τεχνητής νοημοσύνης, ικανό να κατανοεί οδηγίες φυσικής γλώσσας και να δημιουργεί απαντήσεις παρόμοιες με την ανθρώπινη συνομιλία σε μια ευρεία γκάμα θεμάτων.

Η εμφάνιση των Large Language Models (LLM) όπως το GPT-4 έχει εισαγάγει νέες δυνατότητες στην επεξεργασία της φυσικής γλώσσας. Με την εισαγωγή του ChatGPT API από την OpenAI, η απρόσκοπτη ενσωμάτωση και δημιουργία συνομιλιών μεσω της τεχνητής νοημοσύνης σε εφαρμογές έχει γίνει άμεσα εφικτή.

3.1.2 Τι είναι το GPT

Το GPT, ένα ακρωνύμιο του Generative Pre-trained Transformer, αντιπροσωπεύει μια σειρά γλωσσικών μοντέλων που αναπτύχθηκαν από την OpenAI. Με εξέλιξη από GPT-1 σε GPT-4, αυτά τα μοντέλα εκπαιδεύονται σε εκτεταμένα δεδομένα κειμένου και μπορούν να βελτιωθούν περαιτέρω για συγκεκριμένες γλωσσικές εργασίες. Διαπρέπουν στην παραγωγή συνεκτικού κειμένου προβλέποντας επόμενες λέξεις. Το ChatGPT, ένα συνομιλητικό AI που βασίζεται σε αυτά τα μοντέλα, εμπλέκεται σε αλληλεπιδράσεις φυσικής γλώσσας και έχει σχεδιαστεί για να είναι ασφαλές, αξιόπιστο και ενημερωτικό, με τις γνώσεις του να περιλαμβάνουν δεδομένα μέχρι και τον Μάρτιο του 2023.

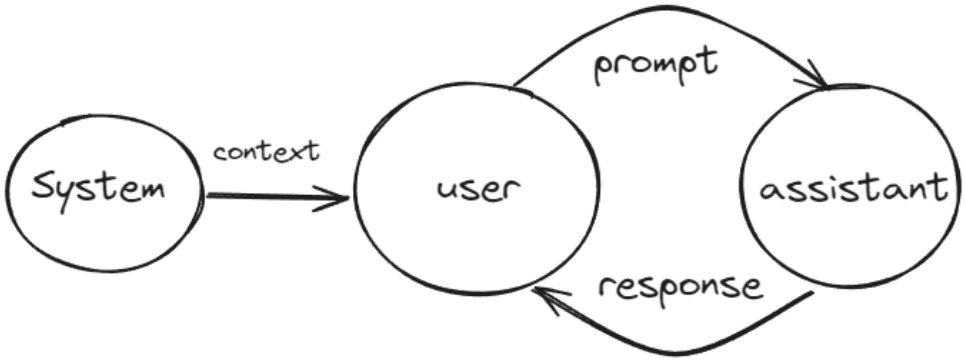
3.1.3 Τι είναι το ChatGPT API

Το ChatGPT API παρέχει πρόσβαση σε μοντέλα συνομιλίας AI της OpenAI, όπως το GPT-4, GPT-4 turbo, GPT-3 κ.λπ. Επιτρέπει την ενσωμάτωση αυτών των μοντέλων γλώσσας σε εφαρμογές μέσω κλήσεων API. Διάφορες περιπτώσεις χρήσης, που κυμαίνονται από τη δημιουργία chatbot και εικονικών βοηθών έως την αυτοματοποίηση των ροών εργασίας υποστήριξης πελατών και τη δημιουργία περιεχομένου όπως μηνύματα ηλεκτρονικού ταχυδρομείου και αναφορές, γίνονται εφικτές με την αξιοποίηση αυτών των API [13].

Το API υπερέχει στη δημιουργία όχι μόνο συνεκτικού αλλά και σχετικού κειμένου. Αυτό σημαίνει ότι το ChatGPT μπορεί να παρέχει απαντήσεις άψογα ευθυγραμμισμένες με τη ροή συνομιλίας, διατηρώντας τη συνάφεια με το context της συνομιλίας.

3.1.4 Διαμόρφωση Συμπεριφοράς του API

Οι τρεις κύριοι τύποι μηνυμάτων που διαμορφώνουν τη συμπεριφορά ενός chatbot είναι τα μηνύματα «system», «user» και «assistant». Τα μηνύματα system αντιπροσωπεύουν τις εσωτερικές διαδικασίες του chatbot, τα μηνύματα user είναι οι εισροές από ανθρώπους και τα μηνύματα assistant είναι οι απαντήσεις του chatbot.



Σχήμα 3.1. Διαμόρφωση συμπεριφοράς ChatGPT API & data flow

Τα μηνύματα system επιτρέπουν στο chatbot να παρακολουθεί την κατάσταση της συνομιλίας, να κατανοεί το content και να προσδιορίζει τις κατάλληλες απαντήσεις. Για παράδειγμα, τα μηνύματα system μπορεί να καταγράφουν το τρέχον θέμα συζήτησης, τη διάθεση του χρήστη ή προηγούμενες συνομιλίες με αυτόν τον χρήστη. Αυτά τα μεταδεδομένα διαμορφώνουν τον τρόπο με τον οποίο το chatbot ερμηνεύει τα μηνύματα user και δημιουργεί μηνύματα assistant.

Τα μηνύματα user παρέχουν την ακατέργαστη εισροή συνομιλίας που πρέπει να αναλύσει και να αντιδράσει το chatbot. Το chatbot χρησιμοποιεί επεξεργασία φυσικής γλώσσας για να εξάγει νόημα από αυτά τα μηνύματα και να προσδιορίσει την πρόθεση. Διαφορετικές φράσεις, μήκος, σημεία στίξης και περιεχόμενο μηνυμάτων user θα προκαλέσουν διαφορετικές απαντήσεις από το chatbot.

Τέλος, τα μηνύματα assistant αντιπροσωπεύουν τις απαντήσεις του chatbot που διαμορφώνονται από την ανάλυσή του για την κατάσταση του συστήματος και την εισαγωγή μηνυμάτων user. Ο τόνος, η προσωπικότητα και το περιεχόμενο πληροφοριών των μηνυμάτων assistant καθορίζουν τελικά την εμπειρία του χρήστη. Η προσεκτική επεξεργασία των κανόνων chatbot είναι το κλειδί για τη δημιουργία ελκυστικών και χρήσιμων διαλόγων.

3.1.5 Prompt Engineering

Το Prompt engineering είναι μια κρίσιμη πτυχή του natural language processing (NLP), ιδιαίτερα στο πλαίσιο των μοντέλων μηχανικής εκμάθησης όπως το GPT-3.5, τα οποία βασίζονται σε καλοσχεδιασμενες προτροπές για τη δημιουργία απαντήσεων. Το concept του prompt engineering, περιλαμβάνει το σχεδιασμό και τη διατύπωση προτροπών ή ερωτημάτων εισόδου με τρόπο που να καθοδηγεί αποτελεσματικά το μοντέλο να παράγει τα επιθυμητά αποτελέσματα. Στον τομέα του NLP, το prompt engineering είναι παρόμοια με τη δημιουργία μιας ερώτησης ή μιας εντολής που εξάγει τις επιθυμητές πληροφορίες ή συμπεριφορά από το γλωσσικό μοντέλο.

Τα τελευταία χρόνια, το prompt engineering έχει αποκτήσει εξέχουσα θέση λόγω του σημαντικού αντίκτυπού της στην απόδοση και την ευελιξία των γλωσσικών μοντέλων. Ερευνητές και επαγγελματίες έχουν αναγνωρίσει τη σημασία της προσεκτικής κατασκευής προτροπών για την προσαρμογή της συμπεριφοράς αυτών των μοντέλων για συγκεκριμένες εργασίες. Αυτή η διαδικασία περιλαμβάνει πειραματισμό με διαφορετικές δομές, στυλ και παραλλαγές εντολών για την εύρεση της βέλτιστης διαμόρφωσης που αποφέρει τα επιθυμητά αποτελέσματα. Είναι ένα μείγμα γλωσσικής διαίσθησης, τεχνογνωσίας στον τομέα και εμπειρικών δοκιμών.

Μία από τις προκλήσεις είναι η επίτευξη ισορροπίας μεταξύ ειδικότητας και γενικότητας. Μια προτροπή πρέπει να είναι αρκετά συγκεκριμένη ώστε να καθοδηγεί το μοντέλο με ακρίβεια προς την επιθυμητή εργασία, αλλά θα πρέπει επίσης να είναι γενικευμένη για να χειρίζεται μια σειρά ή γκαμα από εισόδους. Αυτό απαιτεί μια λεπτή κατανόηση των δυνατοτήτων του μοντέλου, των πιθανών προκαταλήψεων και των επιπλοκών των εργασιών που αναμένεται να εκτελέσει. Οι ερευνητές συχνά επαναλαμβάνουν πολλαπλά prompt, τελειοποιώντας και προσαρμόζοντάς τα για να επιτύχουν το επιθυμητό επίπεδο ακρίβειας και στιβαρότητας.

Συμπερασματικά, το prompt engineering είναι ένα δυναμικό και εξελισσόμενο πεδίο στην έρευνα του NLP. Καθώς τα μοντέλα μηχανικής μάθησης συνεχίζουν να εξελίσσονται, η τέχνη και η επιστήμη της δημιουργίας αποτελεσματικών προτροπών γίνονται όλο και πιο ζωτικής σημασίας. Οι ερευνητές πρέπει να βρουν τη λεπτή ισορροπία μεταξύ ειδικότητας και γενικότητας, ηθικών κριτηρίων και συνάφειας για να αξιοποιήσουν πλήρως τις δυνατότητες των γλωσσικών μοντέλων για διάφορες εφαρμογές.

3.1.6 Χρήση του API

Τα μοντέλα συνομιλίας λαμβάνουν μια λίστα μηνυμάτων ως είσοδο και επιστρέφουν ένα μήνυμα που δημιουργείται από το μοντέλο ως έξοδο.

Ένα παράδειγμα κλήσης του Chat Completions API έχει την εξής μορφή:

```
async function main() {
  const completion = await openai.chat.completions.create({
  messages: [
      {"role": "system", "content": "You are a helpful assistant."},
      {"role": "user", "content": "Who won the world series in 2020?"},
      {"role": "assistant", "content": "The Los Angeles Dodgers won the World
Series in 2020."},
      {"role": "user", "content": "Where was it played?"}
      ],
      model: "gpt-3.5-turbo",
    });
  }
```

Η κύρια είσοδος είναι η παράμετρος messages. Τα μηνύματα πρέπει να είναι μια σειρά αντικειμένων message, όπου κάθε αντικείμενο έχει έναν ρόλο (είτε "system", "user" ή "assistant") και περιεχόμενο. Οι συνομιλίες μπορεί να είναι τόσο σύντομες όσο ένα μήνυμα, αλλά και παραπάνω.

Συνήθως, μια συνομιλία μορφοποιείται πρώτα με ένα μήνυμα system, ακολουθούμενο από εναλλασσόμενα μηνύματα user και assistant. Εκτός απο τα message objects, υπάρχει μια μεγάλη πληθώρα παραμέτρων για την επιπλέον παραμετροποίηση της απάντησης που θα λάβει ο client απο το

API. Στην συνέχεια, με την εκτέλεση του request ο client λαμβάνει το παρακάτω response:


```
{
  "choices": [
    {
      "finish_reason": "stop",
      "index": 0,
      "message": {
        "content": "The 2020 World Series was played in Texas at Globe Life
Field in Arlington.",
        "role": "assistant"
      },
      "logprobs": null
    }
  ],
  "created": 1677664795,
  "id": "chatcmpl-7QyqpwdfhqwajicIEznoc6Q47XAyW",
  "model": "gpt-3.5-turbo-0613",
  "object": "chat.completion",
  "usage": {
    "completion_tokens": 17,
    "prompt_tokens": 57,
    "total_tokens": 74
  }
}
```


3.1.7 Διαχείριση Token

Τα language models διαβάζουν και γράφουν κείμενο σε κομμάτια που ονομάζονται tokens. Στα αγγλικά, ένα διακριτικό μπορεί να είναι τόσο σύντομο όσο ένας χαρακτήρας ή όσο μια λέξη (π.χ. a ή apple) και σε ορισμένες γλώσσες τα διακριτικά μπορεί να είναι ακόμη μικρότερα από έναν χαρακτήρα ή ακόμη και μεγαλύτερα από μία λέξη.

Για παράδειγμα, το string "ChatGPT is great!" κωδικοποιείται σε έξι token: ["Chat", "G", "PT", " is", " great", "!"].

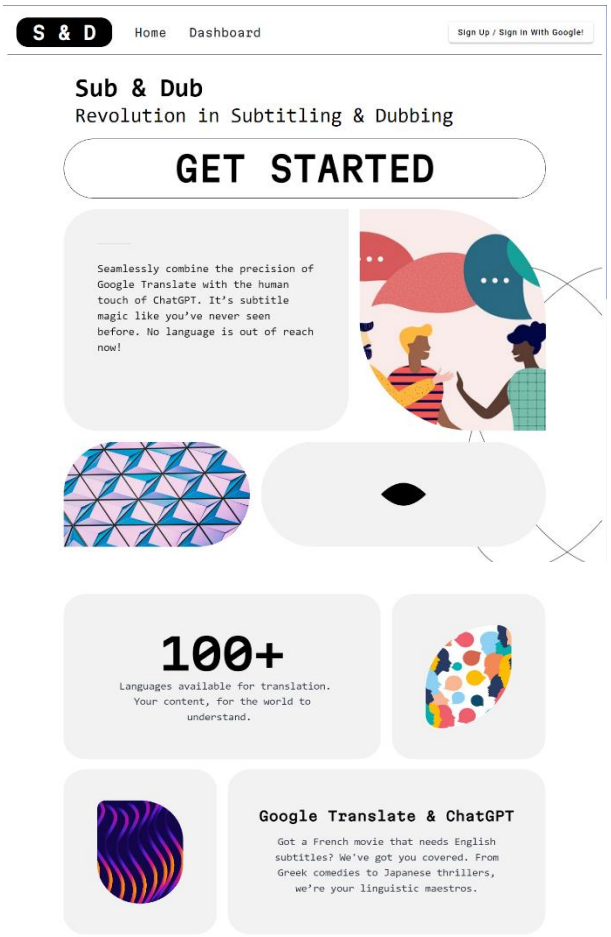
Ο συνολικός αριθμός των token σε ενα API call επηρεάζει:

- Πόσο κοστίζει το συγκεκριμένο API call, το οποίο πληρώνεται ανα token
- Πόσος χρόνος διαρκεί η κλήση API, καθώς η σύνταξη περισσότερων token απαιτεί περισσότερο χρόνο
- Εάν η API κλήση μπορεί να πραγματοποιηθεί, καθώς τα συνολικά token πρέπει να είναι κάτω από το μέγιστο όριο του μοντέλου (4097 μάρκες για gpt-3.5-turbo

Κεφάλαιο 4ο: Αναλυτική Περιγραφή της Εφαρμογής

Στο κεφάλαιο αυτό θα παρουσιαστεί αναλυτικά η εφαρμογή και οι λειτουργίες της, απο το πρώτο βήμα ενός χρήστη εώς και το τελευταίο. Αυτό περιλαμβάνει το homepage της εφαρμογής, την μέθοδο sign-in & sign-up, το dashboard, την διαχείριση των βίντεο ενος χρήστη, τους υπότιτλους που έχει δημιουργήσει, την επεξεργασία και δημιουργία του subtitle file, κ.α.

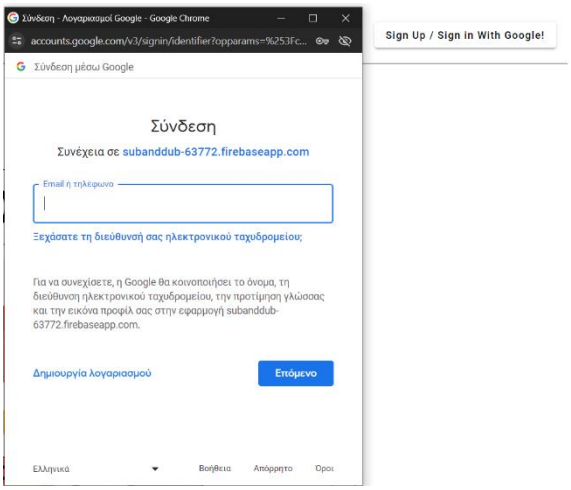
Homepage



Το Homepage της εφαρμογής, οπου γίνεται μια σύντομη περιγραφή των δυνατοτήτων της εφαρμογής, και παροτρείνει τον χρήστη να προχωρήσει στην πλατφόρμα κάνοντας sign-up. Το homepage, θα έλεγε κάποιος, οτι είναι το "lobby" της εφαρμογής, και ο πρωταρχικός στόχος της αρχικής σελίδας είναι να κάνει τον επισκέπτη να νιώσει ευπρόσδεκτος και στη συνέχεια να παρέχει πληροφορίες σχετικά με το τι μπορεί να βρεθεί στον ιστότοπο. Επέλεξα να κρατήσω το design αρκετα simplistic για να προβάλω στον χρήστη απευθείας τις πληροφορίες που θα περίμενε να βρει σε μια πλατφόρμα υποτιτλισμού

Σχήμα 4.1. Homepage

Sign-in & Sign-Up



Σχήμα 4.2. Sign In pop-up

Κάνοντας click στο κουμπί Sign Up/ Sign in with Google, ανοίγει το prompt για να κάνει login ένας χρήστης χρησιμοποιώντας τον Google λογαριασμό του, χωρις να χρειαστεί να δημιουργήσει νέο λογαριασμό από την αρχή. Οταν γίνει το authentication του χρήστη, σε αυτό το σημείο η εφαρμογή εντοπίζει αν είναι νέος χρήστης ή είναι repeat επισκέπτης της πλατφόρμας. Εάν είναι νέος χρήστης, απευθείας γίνεται μια νεα εγγραφή στην Firebase με τα unique identifiers του, διαφορετικά κάνει retrieve τα ήδη υπάρχον στοιχεία του και γίνεται απευθείας initialize ως ο τρέχον User. Να σημειωθεί επίσης οτι το connection του χρήστη στην εφαρμογή ειναι persistent, δηλαδή κλείνοντας την εφαρμογή τελείως, και με την επιστροφή του σε αυτήν, ο χρήστης παραμένει Logged in.

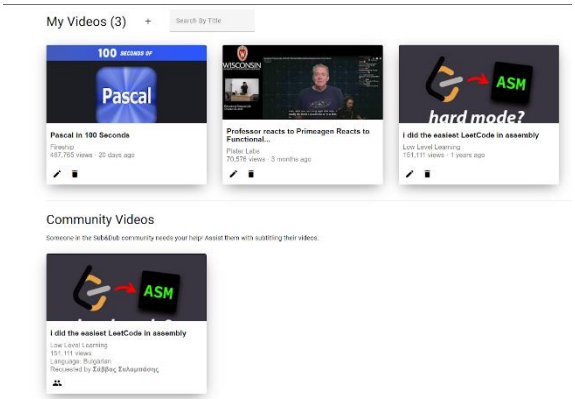
Για την καλύτερη διαχείριση των Authentication services γίνεται η χρήση της AngularFire, μιας wrapper βιβλιοθήκης για την πραγματοποίηση των διάφορων interaction με την Firebase. Παρακάτω είναι ένα code snippet που αφορά το sign-in/sign-up ενός user:

```
constructor(private fireAuth: AngularFireAuth,
            private firestore: AngularFirestore,
            private router: Router) {
      this.user$ = this.fireAuth.authState.pipe(
        switchMap((user) => {
          if (user) {
            this.loggedIn$.next(true);
            return
this.firestore.doc<GmailUser>(`users/${user.uid}`).valueChanges();
          } else {
            this.loggedIn$.next(false);
            return of(null);
          }
        })
      );
    }
```

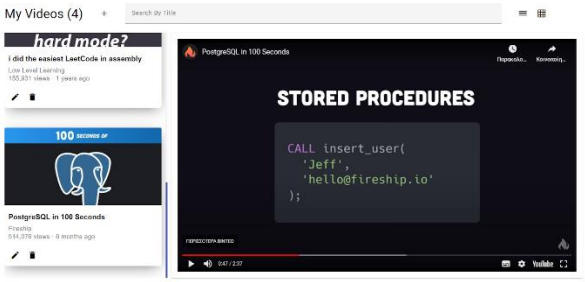
Με το initialize του auth.service.ts, το οποίο είναι το service που έχω δημιουργήσει και είναι υπεύθυνο για όλες τις authentication διεργασίες της εφαρμογής, ελέγχει το observable authState, το οποίο αποθηκεύει το current state του user, και ανάλογα με το αν υπάρχει τιμή/value για τον current user, τοτε περνάει στην BehaviorSubject μεταβλητή loggedIn$ (ενας τύπος Observable) την τιμή true ή false αντίστοιχα, έτσι ώστε η εφαρμογή να αναγνωρίζει αν ο χρήστης είναι logged in ή όχι. Βασικό να σημειωθεί επίσης ότι στην περίπτωση που ο user είναι true (logged in δηλαδή), γίνεται ενα subscription στο document του current user, μέσω του function this.firestore.doc<GmailUser>(`users/${user.uid}`).valueChanges() έτσι ώστε η εφαρμογή να ειδοποιείται για οποιαδήποτε αλλαγή συμβαίνει στον user και να λαμβάνει real-time updates.

Dashboard

Κάνοντας successful login στην εφαρμογή, ο χρήστης αποκτά πρόσβαση στο dashboard του.



Σχήμα 4.3. Dashboard Grid View



Σχήμα 4.4. Dashboard List View

Το Dashboard του χρήστη, όπου εκεί βρίσκονται ολα τα ήδη αποθηκευμένα βίντεο του, καθώς και άλλες λειτουργίες οπως το search, η δυνατότητα προσθήκης νέων βίντεο κ.α . Εκτός από τα βίντεο του χρήστη, υπάρχει και ενα section οπου ονομάζεται Community Videos, και εμφανίζει βίντεο τα οποία κάποιος άλλος χρήστης της εφαρμογής έχει ζητήσει βοήθεια από το community για τον υποτιτλισμό ενός δικου του βίντεο. Σε αυτά τα βίντεο ο current χρήστης της εφαρμογής μπορει να βοηθήσει κάποιον άλλον user, βοηθώντας τον να δημιουργήσει υπότιτλους για κάποια συγκεκριμένη γλώσσα. Αυτή η διαδικασία δημιουργεί την αίσθηση του community μεταξύ των χρηστών.

Στο Dashboard γίνονται αρκετέας διεργασίες για την σωστή παρουσίαση όλων των βίντεο ενος χρήστη και όλες τις επιμέρους πληροφορίες αυτών, οπως κλήσεις στην Firebase βάση, API calls στο YouTube κ.α. Παρακάτω είναι ένα snippet κώδικα που τρέχει την στιγμή που γίνεται initialize το dashboard:

```
ngOnInit(): void {
  this.user$ = this.auth.user;
  this.loading$.next(true);
  combineLatest([
    this.user$,
    this.user$.pipe(
      distinctUntilChanged(),
      switchMap(user => {
        if (user) {
          this.userId$.next(user.uid);
          this.userVideos$ = this.dashboardService.getVideos(user.uid);
          this.communityVideos$ = this.dashboardService.getCommunityVideos();
          return combineLatest([this.userVideos$, this.communityVideos$]);
        } else {
          // If there is no user, return an empty observable
          return of(null);
        }
      }))]).pipe(
    distinctUntilChanged(),
    switchMap(([user, [userVideos, communityVideos]]) => {
      const userVideoIds = userVideos.map(item => item.videoId);
      const communityVideoIds = communityVideos.map(item => item.videoId);

      // Using Set to ensure unique video IDs
      const uniqueVideoIds = new Set([...userVideoIds, ...communityVideoIds]);

      const allVideoIds = Array.from(uniqueVideoIds).join(',');
      return this.youtubeService.getVideoDetails(allVideoIds);
    })).subscribe((res: YoutubeVideoDetails[]) => {
      if (res) {
        this.youtubeVideoDetails = res;
        this.loading$.next(false);
      }
    });
```

Αναλυτικά το τι πραγματοποιεί ο παραπάνω κώδικας:

- this.user$ = this.auth.user; - Εκχωρεί το user observable απο το auth service στην τοπική μεταβλητή user$.
- this.loading$.next(true); - Ορίζει μια αρχική κατάσταση loading για να υποδείξει ότι η ανάκτηση δεδομένων βρίσκεται σε εξέλιξη και να ενημερώσει ανάλογα το UI με εναν loader.
- Χρησιμοποιεί το combineLatest για να συνδυάσει πολλαπλά observables σε ένα.
- χρησιμοποιεί το distinctUntilChanged και το switchMap για τη διαχείριση των αλλαγών ενος user και την ανάκτηση Video και Community Videos για τον current user.

Ανάκτηση Video:

- Αν υπάρχει ο user (if (user) { ... }):
  - Κάνει set το user ID στο userId$ observable.
  - Κάνει fetch τα video του χρήστη (userVideos$) και τα community videos (communityVideos$) χρησιμοποιώντας το dashboardService.
  - Χρησιμοποιεί το combineLatest για να περιμένει την ανάκτηση και των δύο κατηγοριών video (user & community).
- Απο τα user video και community videos, χρησιμοποιεί ένα Set για την αποθήκευση των unique video IDs.

YouTube API Call:

- Καλεί το function του YoutubeService, this.youtubeService.getVideoDetails() με τα concatenated video IDs για την ανάκτηση περισσότερων πληροφοριών για κάθε video(views, video owner, when it was uploaded etc..) απο το YouTube API.
- Κάνει update την τοπική μεταβλητή youtubeVideoDetails με τις πληροφορίες που επέστρεψε η κληση στο YouTube API.
- Κάνει set το loading σε false για να ενημερώσει ανάλογα το UI, εφόσον το request έχει πραγματοποιηθεί.

Αυτός ο κώδικας έχει σχεδιαστεί για την αποτελεσματική διαχείριση και ανάκτηση λεπτομερειών των βίντεο για το dashboard, λαμβάνοντας υπόψη τόσο τα βίντεο που αφορούν τον χρήστη όσο και τα community video, και στη συνέχεια ανακτά πρόσθετες λεπτομέρειες από το API του YouTube.

Video Details

Κάνοντας click σε ενα video card απο το dashboard, η εφαρμογή θα τον κάνει navigate στα details αυτού του video.

Σχήμα 4.5. Video Details view

Στο details view ενος video, ο χρήστης μπορεί να προσθέσει νέους υποτίτλους, βλέπει αναλυτικά τι υπότιτλους έχει ήδη δημιουργήσει για το συγκεκριμένο video, συγκεκριμένα εμφανίζει το unique όνομα ενος subtitle entity, το οποίο έχει κάποιο file format (sbv/srt), σε ποια γλώσσα ανήκει και ενα timestamp με το τελευταίο update που έχει συμβεί σε αυτό. Σε κάθε υπότιτλο δίνεται η επιλογή του action "request community help", το οποίο κάνει flag την συγκεκριμένη γλώσσα για το ίδιο βίντεο και την προσθέτει στο community videos του Dashboard. Στα αριστερα του view, στην κάρτα του video, εμφανίζονται πάλι κάποιες βασικές πληροφορίες και οι διαθέσιμοι υπότιτλοι που είναι ήδη uploaded για το συγκεκριμένο βίντεο (ASR ή user-generated).

Παρακάτω είναι μια μικρή επεξήγηση ενός μέρους του details-view component, το οποίο διαχειρίζεται την προσθήκη ενός νέου subtitle:

Ο χρήστης κάνοντας click στην γλώσσα την οποία θέλει να προσθέσει στους υπότιτλους, το ανοίγει ένα dialog prompt για να εισάγει το name και προαιρετικά το format του υποτίτλου:

```
openSubtitleDialog(language: Language): void {
    this.dialog.open(SaveSubtitleDialogComponent,{width:'500px',data:language.
name}).afterClosed().pipe(take(1)).subscribe(dialog => {
    if (dialog?.name) {
    this.detailsViewService.addSubtitle(this.videoId, language,
this.user$.value.uid, dialog.name, dialog.format);
    }})
}
```

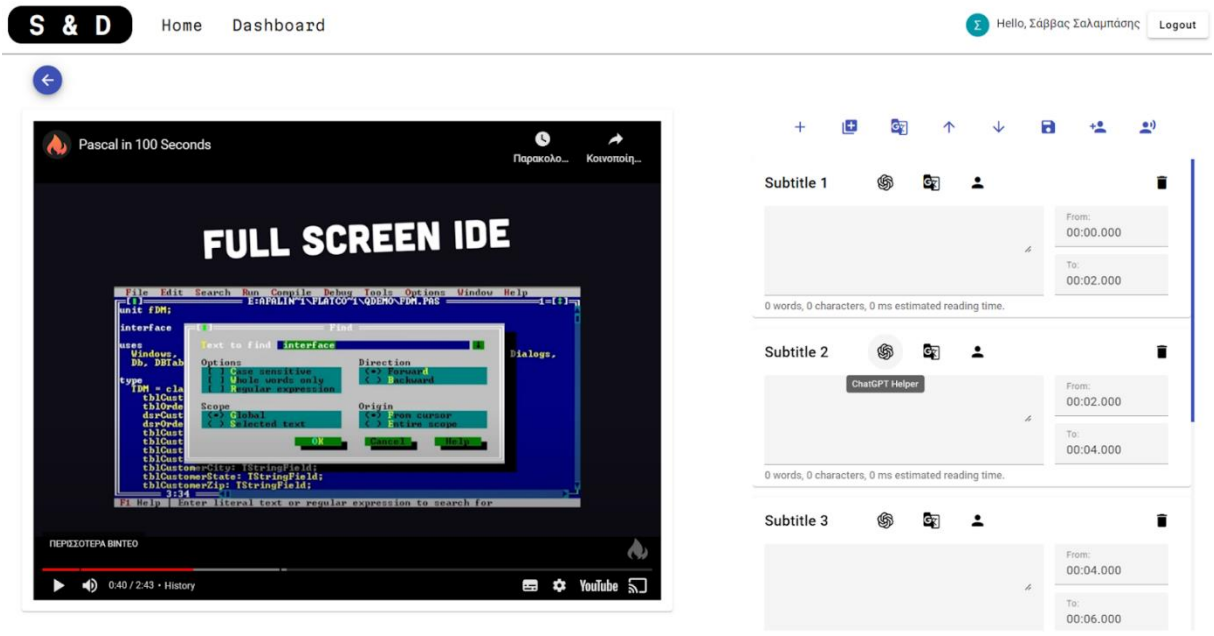
Μόλις ο χρήστης εισάγει τα παραπάνω στοιχεία και κάνοντας click στο Save button, τρέχει το service function addSubtitle(), το οποίο ξεκινάει την διαδικασία προσθήκης του υπότιτλου στο Firestore Database της εφαρμογής:

```
addSubtitle(videoId: string, language: Language, userUid: string, name:
string, format: SubtitleFormat): void {
const docRef: AngularFirestoreDocument =
this.firestore.doc(`users/${userUid}/videos/${videoId}/subtitleLanguages/${lan
guage.language}`);

const docData = {
    humanReadable: language.name,
    ISOcode: language.language,
}

docRef.set(docData).then(()=> {const subtitleRef: AngularFirestoreDocument
docRef.collection(`/subtitles`).doc(name);

const data = {
   lastUpdated: Date.now(),
   fileName: name,
   fullFileName: `${name}.${format}`,
   format: format,
   language: language.name,
   iso: language.language
}

subtitleRef.set(data);
    })
  }
}
```

Το function αρχικά παιρνει ενα referrence στο firestore document μέσω του path που περνάει στην μεταβλητή docRef, και στην συνέχεια εάν δεν υπάρχει ήδη κάποιο άλλο subtitle στην ίδια γλώσσα, δημιουργεί ένα νέο collection για αυτην, και στην συνέχεια δημιουργεί ένα subtitle reference και το προσθήκει σε αυτό το collection περνώντας ως argument το data object το οποίο περιέχει όλες τις αναγκαίες πληροφορίες.

Subtitling

Στο τέλος του flow της εφαρμογής, βρίσκεται το subtitling-container component, στο οποίο ο χρήστης θα πραγματοποιήσει τον υποτιτλισμό για το βίντεο του.



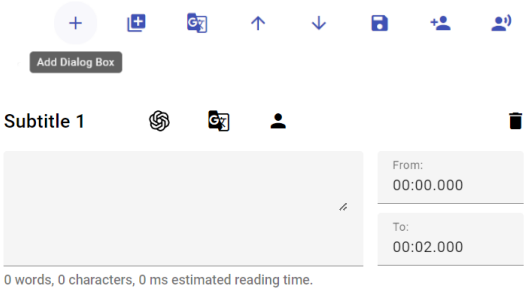
Σχήμα 4.6. Subtitling Container/Components

Σε αυτό εδω το page, ο χρήστης βρίσκει τις διάφορες λειτουργίες υποτιτλισμού που προσφέρει η εφαρμογή. Αυτό το σύνολο των component δίνει την δυνατότητα στον χρήστη να παράξει το τελικό SBV ή SRT αρχείο το οποίο στην συνέχεια θα μπορεί να το χρησιμοποιήσει για τις ανάγκες του. Εδώ βρίσκονται λειτουργίες όπως το Translation, ο ChatGPT helper, person assign, batch dialog creation, file download, Firebase uploads και file import. Τα βασικά component είναι ο Video player, ο οποίος έχει γινει embedded μέσω του YouTube IFrame API, και το σύνολο των dialogs μαζί με τα διάφορα options στα δεξιά του view.

Αρχικά, πρέπει να σημειωθεί οτι το implementation των dialog βασίζεται στο Reactive Form μοντέλο της Angular, το οποίο δίνει ένα εκτενές customization των διαφόρων input και άλλων fields τα οποία διαχειρίζονται το περιεχόμενο και timestamp των υποτίτλων.

Παρακάτω είναι μια επεξήγηση κάποιων λειτουργιών που βρίσκονται στο σύνολο αυτών των component.

4.1.1 Add Dialog Box



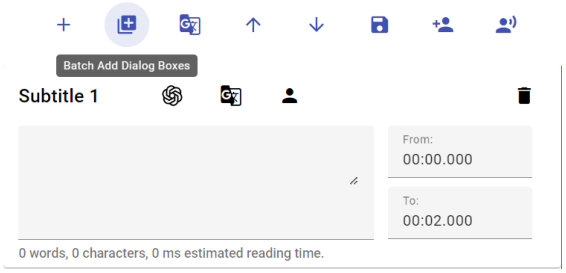
Προσθήκη ενός νέου dialog box/ υποτίτλου

Code snippet:

Το function αυτό προσθέτει ένα dialog box, δημιουργώντας ένα νέο control στο FormGroup που περιέχει το σύνολο των υποτίτλων, ενώ ταυτόχρονα προσθέτει και ενα νεο entry στο dialogBoxes array.

```
addDialogBox(value: ImportModel = null): void {
  this.dialogBoxId ++;
  this.form.addControl(((this.dialogBoxId + '-dialogBox')), this.fb.group({
  subtitles: this.fb.control((value?.subtitleText) ? value.subtitleText : ''),
  start_time: this.fb.control((value?.start_time) ? value.start_time :
this.setStartTimeControlValue()),
   end_time: this.fb.control((value?.end_time) ? value.end_time :
this.setEndTimeControlValue()),
    }));

  this.dialogBoxes.push({
    id: this.dialogBoxId,
    text: value?.subtitleText,
    start_time: value?.start_time,
    end_time: value?.end_time
  });
}
```

4.1.2 Batch add Dialogs



Διαχειρίζεται την δημιουργία όλων των dialog box, ανάλογα με το interval σε δευτερόλεπτα που θέτει ο χρήστης μέχρι το τέλος του video.

Code snippet:

Το function αυτό, μαζί με την συνεργασία μερικών ακόμα helper function για τον υπολογισμό των timestamp και πλήθος dialog που πρέπει να δημιουργήσει, καλει το function addDialogBox που περιγράφηκε και παραπάνω περνώντας τα κατάλληλα timestamps, start_time και end_time σαν arguments για καθε dialog.

```
batchAddDialogBox(): void {
  this.dialog.open(BatchDialogModalComponent).afterClosed().pipe(take(1))
  .subscribe((interval: number) => {
    if (interval > 0) {
      const seconds = this.parseISOtoSeconds(this.videoDuration);
      const controlNames = Object.keys(this.form.controls);
      let lastControlName = controlNames[controlNames.length - 1];
 for (let i = parseInt(lastControlName.split('-')[0]); i < seconds / interval;
i ++) {
    this.addDialogBox({
      start_time: this.form.get(i + '-dialogBox').get('end_time').value,
      end_time: this.intervalAddition(this.form.get(i + '-
dialogBox').get('end_time').value, interval),
       subtitleText: ''
       });
  };

    }
  });
}
```

4.1.3 Translate All Subtitles

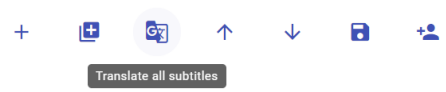
Μετάφραση ολων των περιεχομένων/κειμένων εντός των dialog boxes σε ένα νέο target language (menu selection).

Code snippet:

Η μέθοδος χτίζει το JSON object το οποίο θα σταλθεί σαν παράμετρος στο Translate API, περνώντας ολα τα υπάρχον κείμενα/dialogs στο array q και στο target την γλώσσα προς μετάφραση που επέλεξε ο χρήστης στην εφαρμογή. Στην συνέχεια στέλνει το JSON αυτό στο body του HTTP request και μετά την ολοκλήρωση του, κάνει parse το response και αντικαθιστά τα υπάρχον κείμενα με τα νέα translated κείμενα.

```
translateAllSubtitles(targetLanguage: string): void {
  let translationObject: GoogleTranslateRequestObject = {
    q: [],
    target: targetLanguage
  };

  let controllersToChange = {
    controlsName : []
  };

  Object.keys(this.form.controls).forEach(control=> {
    const controlValue = this.form.get(control).get('subtitles').value;
    if (controlValue) {
      translationObject.q.push(controlValue)
      controllersToChange.controlsName.push(control)
    }
  });

  if (translationObject.q) {
    this.google.translate(translationObject).subscribe((response:
GoogleTranslateResponse) => {
      this._translatedText$.next(response);
      let translationArray: GoogleTranslations[] =
this._translatedText$.value.data['translations'];

  if (translationArray) {
   for (let i = 0; i < controllersToChange.controlsName.length; i++) {
    const control =
this.form.get(controllersToChange.controlsName[i]).get('subtitles');
    control.setValue(translationArray[i].translatedText);
    }
  }});
  }
}
```

4.1.4 Import Subtitles



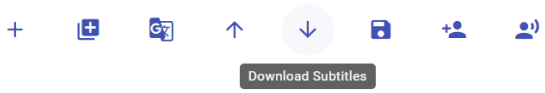
Παρέχεται η δυνατότητα του Import ενος ήδη υπάρχον sbv, srt ή txt format αρχείου. Κάνοντας import αυτό το αρχείο, η εφαρμογή το κάνει parse, και ανάλογα το source format, χτίζει ολα τα dialogs/subtitles που περιέχονται στο file.

Code Snippet:

Το παρακάτω function παίρνει σαν input το raw string του file, και το μετατρέπει σε ενα array απο object κάνοντας match ανα line για να εντοπίσει εαν το current line είναι Subtitle ή Timestamp βάσει ενος regex, το array object αυτό, μπορεί με ευκολία να το κάνει parse η εφαρμογή και να δημιουργήσει ολα τα dialog boxes.

```
cleanMultilineString(input: string): ImportModel[] {
  const lines = input.trim().replace(/(\r|\r)/gm, '').split('\n');

  const subtitleObjects = [];
  let currentSubtitle: ImportModel = {start_time: '', end_time:'',
subtitleText:''};
  for (let i = 0; i < lines.length; i++) {
    const line = lines[i].trim();
    const timestampRegex =
/^(\d{1,2}:\d{2}:\d{2}\.\d{3}),(\d{1,2}:\d{2}:\d{2}\.\d{3})$/;

    if (line.match(timestampRegex)) {
      const [startTime, endTime] = line.split(",");
      currentSubtitle = {
        start_time: this.formatTimestamp(startTime),
        end_time: this.formatTimestamp(endTime),
        subtitleText: ""
       };
      subtitleObjects.push(currentSubtitle);
    } else if (line !== "") {
      currentSubtitle.subtitleText += line + ' ';
    }
  }
  return subtitleObjects;
}
```

4.1.5 Download Subtitles



Κάνει download σε sbv ή srt format, ενα αρχείο το οποίο έχει το περιεχομενο ολων των υπάρχων dialog (text & timestamps).

Για την δημιουργία και download αυτού του αρχείου, πρέπει να γίνει construct ενα Blob object.

```
createSubtitleBlob(): Blob {
  let sbvContent = ''
  Object.keys(this.form.controls).forEach((control) => {
    const currentGroup = this.form.get(control);
    sbvContent += `${'00:'+ currentGroup.get('start_time').value},${'00:' +
currentGroup.get('end_time').value}\n${currentGroup.get('subtitles').value}\n\
n`;
  });
  const blob = new Blob([sbvContent], { type: 'text/sbv;charset=utf8'});
  return blob;
}
```

Σε αυτό το object, περνιέται όλο το σχετικό περιεχόμενο που περιέχει ενα Subtitle file (content & timestamps), και στην συνέχεια γίνεται download στο file system του χρήστη.

```
downloadSubtitle(): void {
  const blob = this.createSubtitleBlob()
  const url = window.URL.createObjectURL(blob);
  const a = document.createElement('a');
  a.href = url;
  a.download = 'subtitles.sbv';
  document.body.appendChild(a);
  a.click();
  window.URL.revokeObjectURL(url);
  document.body.removeChild(a);
}
```

4.1.6 Save Subtitles

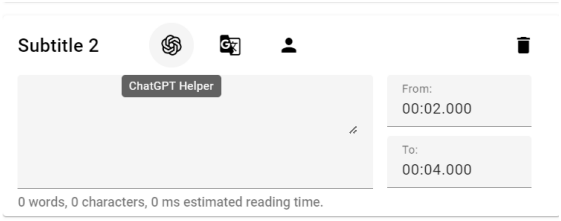
Αποθηκεύει το current state των subtitle στο Firebase Storage της εφαρμογής, για μελλοντική επεξεργασία ή προβολή.

Code Snippet:

Για να πραγματοποιηθεί το upload στο Storage, αρχικά πρέπει να δημιουργηθεί ενα reference στο path στο οποίο θα αποθηκευτεί αυτο το file, και στην συνέχεια εφόσον γίνει το upload successfully, αποθηκεύεται ενα νεο field στο Firestore Database document του χρήστη, το οποίο περιέχει το download path για το συγκεκριμένο file, έτσι ώστε να μπορεί ο εφαρμογή να το κάνει fetch σε περίπτωση του edit/view.

```
createFirestorageRef(storage: AngularFireStorage, language: string, subtitle:
Blob, videoId: string, filePath: string): void {
//sets up the Firestorage required references and receives required parameters
to complete the upload proccess
this.storageRef = storage;
let pathRef: AngularFireStorageReference;
this.authService.user.pipe(take(1)).subscribe(user => {
const pathString = `subtitles/${user.uid}/${videoId}/${language}/${filePath}`;
   pathRef = this.storageRef.ref(pathString);
   this.uploadToFirestorage(pathRef, subtitle, user.uid, videoId , language,
filePath);
   })
}

uploadToFirestorage(pathRef: AngularFireStorageReference, subtitle: Blob,
userUid: string, videoId: string, language: string, filePath:string): void {
   pathRef.put(subtitle).then(() => {
 //update firestore record of user with the download url for this uploaded
subtitle for future use
   const subRef: AngularFirestoreDocument =
this.firestore.doc(`users/${userUid}/videos/${videoId}/subtitleLanguages/${lan
guage}/subtitles/${filePath.split('.')[0]}`);
    pathRef.getDownloadURL().pipe(take(1)).subscribe((url: URL) => {
         subRef.update({
           storageUrl: url,
           lastUpdated: Date.now()
          });
    this.snackbar.open('Update Complete!','DISMISS', {duration:5000});
      })
     }, (reject) => {
          this.snackbar.open(reject.message);
     });
  }
```

4.1.7 ChatGPT Helper



Μία ακομα λειτουργία που είναι διαθέσιμη για τον χρήστη είναι η ενσωμάτωση του ChatGPT, για την επεξεργασία/transform μιας πρότασης με συγκεκριμένους τρόπους που έχουν οριστεί από την εφαρμογή, συγκεκριμένα:

- Να δώσει σε μια πρόταση πιο χαρούμενο τόνο
- Να δώσει σε μια πρόταση πιο λυπητερό τόνο

- Να επιμηκύνει μια πρόταση
- Να μικραίνει μια πρόταση
- Να χρησιμοποιήσει την λειτουργία μετάφρασης

Code Snippet:

Ανάλογα με την επιλογή του χρήστη μέσω του UI, εκτελεί το αντίστοιχο prompt που βρίσκεται στο promptMap, μέσω του ChatGPT Api και λαμβάνει μέσω του response την νέα αλλαγμένη πρόταση.

```
readonly promptMap = new Map([
  ['sad', 'Adjust the tone of the following sentence to make it sadder.'],
  ['joyful', 'Transform the following sentence to make it more joyful.'],
  ['shorter', 'Condense the following sentence while preserving its original
meaning.'],
  ['longer', 'Enlarge the following sentence without changing its original
meaning.']
]);

 getDataFromOpenAI(GPTaction: ChatGPTACtion): Observable<any> {
  const url = "https://api.openai.com/v1/chat/completions";
  const  httpHeaders = new HttpHeaders().set("Authorization",`Bearer
${openAIConfig.apiKey}`);

  let payload = {
    model: 'gpt-3.5-turbo',
    messages: [{ role: 'user', content: this.promptMap.get(GPTaction.action) +
GPTaction.text }],
  }

  return this.http.post(url, payload, {headers: httpHeaders} ).pipe(
    map((res: any) => {
      return res.choices[0].message.content
    })
  );
}
```

Κεφάλαιο 5ο: Συμπεράσματα & Βελτιώσεις

Αυτή η Δ.Ε διερεύνησε την ανάπτυξη και την αξιολόγηση μιας διαδικτυακής εφαρμογής προσαρμοσμένης για τον υποτιτλισμό βίντεο στο YouTube, με κύρια εστίαση στην προώθηση της προσβασιμότητας μέσω της βελτίωσης των διαδικασιών υποτιτλίσμού. Η έρευνα ξεκίνησε αναγνωρίζοντας την αυξανόμενη σημασία του ψηφιακού video content και την επιτακτική ανάγκη για διευκόλυνση στην κατανάλωσή του. Μέσω του σχεδιασμού και της υλοποίησης μιας φιλικής προς τον χρήστη διαδικτυακής εφαρμογής, στόχος της ήταν να αντιμετωπίσει τις διαφορετικές απαιτήσεις προσβασιμότητας των χρηστών, συμβάλλοντας τελικά σε ένα ψηφιακό περιβάλλον χωρίς γλωσσικά εμπόδια.

Αποδείχθηκε εξαιρετικά σημαντική η συμβολή των συγχρονων τεχνολογιών, του Angular Framework και το Firebase, που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής διευκολύνοντας το development process. Τα εργαλεία που προσέφερε η Angular οσον αφορά την διαχείριση των διαφόρων user input μεσω των ευκολόχρηστων βιβλιοθηκών της, η διαχείριση των ασύγχρονων δεδομένων μέσω των Observable και η εύκολη υλοποιήση services για ανάκτηση δεδομένων απο εξωτερικες πηγές (Firebase,YouTube etc.), έπαιξαν κρίσιμο ρόλο στην ομαλή ανάπτυξη της εφαρμογής.

Κομβικής σημασίας επίσης ήταν η χρήση του Firebase για το “backend” κομμάτι της εφαρμογής. Προσέφερε έξυπνες λύσεις οσον αφορα το authentication των user, την αποθήκευση δεδομένων μέσω του Firestore, μια NoSQL βάση δεδομένων, και το Cloud Storage για την εύκολη αποθήκευση και επεξεργασία όλων των απαραίτητων αρχείων ενός χρήστη.

Η web εφαρμογή έδειξε με επιτυχία τις ικανότητες της παρέχοντας στους χρήστες τη δυνατότητα να δημιουργούν, να επεξεργάζονται και να προσαρμόζουν τους υπότιτλους τους σε διάφορες γλώσσες με ευκολία. Η δυνατότητα συνεργασίας με άλλη μέλη της εφαρμογής ενθάρρυνε την αίσθηση της κοινότητας, επιτρέποντας στους χρήστες να συνεισφέρουν συλλογικά στη δημιουργία υποτίτλων.

Φυσικά, υπάρχουν ακόμα βελτιώσεις που μπορούν να γίνουν στην εφαρμογή οι οποίες θα διευκόλυναν τον χρήστη ακόμα παραπάνω στην διαδικασία του υποτιτλισμού αλλα και της μεταγλώτισσης.

Η κυριότερη βελτίωση που θα μπορούσε να γίνει, θα ήταν η προσθήκη του Voice Generation σε πολλαπλές γλώσσες (text-to-speech) για κάποιο βίντεο μέσω των μοντέλων της Google και της OpenAI. Στα πλαίσια αυτής της Δ.Ε, έγινε μια αρχική προσπάθεια ενσωμάτωσης αυτής της λειτουργίας χρησιμοποιώντας το μοντελο της Google – Speech και της Speech Synthesis Markup Language (SSML), μιας markup γλώσσας που δίνει περισσότερες επιλογές customization του ήχου [14], και εμφάνισε ιδιαίτερο ενδιαφέρον όσον αφορα τις δυνατότητες μιας τέτοιας λειτουργίας. Κάνοντας χρήση μιας τέτοιας λειτουργίας, ο χρήστης εκτός απο υπότιτλους σε οποιαδήποτε γλώσσα της επιλογής του, ταυτόχρονα θα είχε την δυνατότητα να δημιουργήσει και μεταγλωττίσεις για τα βίντεο του.

Μία επίσης σημαντική βελτίωση θα ήταν η χρήση και ενσωμάτωση ειδικά εκπαιδεύμενων GPT Agents [15], μια νεα cutting-edge τεχνολογία που αναπτύχθηκε απο την εταιρία OpenAI, η οποία δίνει την δυνατότητα δημιουργίας ειδικών GPT μοντέλων για την εκπλήρωση συγκεκριμένων εργασιών, στην περίπτωση της εφαρμογής, την δημιουργία υποτίτλων ή μεταγλωττίσεων. Αυτή οι Agents μπορούν να συνδέονται απευθείας στην εφαρμογή και να παρέχουν στον χρήστη real-time feedback για την διαδικασία για την οποία έχουν εκπαιδευτεί.

Επιπλέον, μια ακόμα βελτίωση η οποία θα μπορούσε να γίνει, θα ήταν στο interface στο οποίο ο χρήστης δημιουργεί τους υποτίτλους για το βίντεο του, όπου η προσθήκη ενος timeline το οποίο αναπαριστά με

οπτικό τρόπο τον συγχρονισμό των υποτίτλων (βλ. YouTube Studio) θα βελτίωνε σημαντίκα το User Experience (UX).

Συμπερασματικά, υπάρχει ενα μεγάλο φάσμα τεχνολογιών το οποίο συνεχώς εξελίσσεται που μπορεί να βελτιώσει σημαντικά την διαδικασία του υποτιτλισμού και της μεταγλώτισσης, είτε για επαγγελματίες αλλα και απλούς χρήστες της πλατφόρμας του YouTube, με ολο και μεγαλύτερη ευκολία.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] C.Evans, “Importance of video content on social media”, https://www.pictureperfectphoto.co.uk/ [Online]. Available: https://www.pictureperfectphoto.co.uk/importance-of-video-content-on-social-media/ (accessed Jan. 3, 2024).

[2] G.Sharma, “6 Strong Reasons Why Video Marketing is Important”, https://raindance.org/ [Online]. Available: https://raindance.org/6-strong-reasons-why-video-marketing-is-important/ (accessed Jan. 3, 2024).

[3] “Establishing the importance of video”, https://hihaho.com/ [Online]. Available: https://hihaho.com/blog/establishing-the-importance-of-video/ (accessed Jan. 3, 2024).

[4] “Video is king – The importance of video marketing.”, https://www.imsmarketing.ie/ [Online]. Available: https://www.imsmarketing.ie/news/video-is-king-the-importance-of-video-marketing/ (accessed Jan. 3, 2024).

[5] Google Inc, "Saying Goodbye to YouTube's Community Contributions feature after September 28, 2020". Google Inc. Retrieved June 8, 2022. [Online]. Available: https://support.google.com/youtube/thread/61967856

[6] B.Clarine, “11 Reasons Why Video is Better Than Any Other Medium”, https://www.advancedwebranking.com/ [Online]. Available: https://www.advancedwebranking.com/blog/11-reasons-why-video-is-better (accessed Jan. 3, 2024).

[7] H.Stephens, “12 Reasons to Add Subtitles to Your Content”, https://www.linkedin.com/ [Online]. Available: https://www.linkedin.com/pulse/12-reasons-add-subtitles-your-content-holly-stephens/ (accessed Jan. 5, 2024).

[8] Angular/ Google Inc., “Angular Documentation”, https://angular.io/ [Online]. Available: https://angular.io/docs (accessed Jan. 8, 2024).

[9] “Testing Angular A Guide to Robust Angular Applications”, https://testing-angular.com/ [Online]. Available: https://testing-angular.com/ (accessed Jan. 8, 2024).

[10] Google Inc., “Firebase Documentation”, https://firebase.google.com/ [Online]. Available: https://firebase.google.com/docs (accessed Jan. 8, 2024).

[11] Git , https://git-scm.com/ [Online]. Available: https://git-scm.com/ (accessed Jan. 8, 2024).

[12] GitHub , https://github.com/ [Online]. Available: https://docs.github.com/en (accessed Jan. 8, 2024).

[13] OpenAI , “ChatGPT API Documentation”, https://platform.openai.com/ [Online]. Available: https://platform.openai.com/docs/ (accessed Jan. 8, 2024).

[14] Google Inc., “Speech-to-Text”, https://cloud.google.com/ [Online]. Available: https://cloud.google.com/speech-to-text (accessed Jan. 16, 2024).

[15] OpenAI , Introducing GPTs", https://openai.com/ [Online]. Available: https://openai.com/blog/introducing-gpts (accessed Jan. 16, 2024).

ΠΑΡΑΡΤΗΜΑ Α : Πηγαίος Κώδικας

Ο πηγαίος κώδικας (source code) και το deployed application βρίσκονται διαθέσιμα στο GitHub Repository: https://github.com/savvas23/SubAndDub