



ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σύστημα ασφαλείας web camera με χρήση του Raspberry Pi και δυνατότητα παρακολούθησης μέσω site

Βασίλειος Γεώργιος Γκόγκος

A.M.: 513033

Επιβλέπων

Άγγελος Γιακουμής (Επικουρος Καθηγητής)

06/2025

ΔΗΛΩΣΗ ΤΗΡΗΣΗΣ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ

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 παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Γκόγκου Βασιλείου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

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

«Αφιέρωση»

Πρόλογος

Περίληψη

Η Παρούσα πτυχιακή εργασία εστιάζει στις σύγχρονες τεχνολογίες του *iot* (*internet of things*) και των δυνατοτήτων που αυτές προσφέρουν μέσω εφαρμογών οι οποίες θα βελτιώσουν και θα απλοποιήσουν την καθημερινότητα του κοινωνικού συνόλου.

Οι κύριοι άξονες της πτυχιακής αυτής είναι η δημιουργία και η υλοποίηση ενός εύχρηστου και αποτελεσματικού συστήματος κάμερας ασφαλείας με την χρήση του μικροϋπολογιστή *Raspberry Pi*.

Το σύστημα έχει ως στόχο την άμεση παρακολούθηση του χώρου μέσω του διαδικτύου και βασίζεται σε τεχνολογίες όπως η ανίχνευση κίνησης μέσω της *opencv* βιβλιοθήκης.

Για την βασική δομή του προγράμματος χρησιμοποιήθηκε η *Java*. Ενώ για την δημιουργία της ιστοσελίδας χρησιμοποιήθηκαν η *html* καθώς και η *javascript* για περισσότερη διαδραστικότητα.

Τέλος η επικοινωνία μεταξύ πελάτη και διακομιστή παρέχεται με την εφαρμογή του πρωτοκόλλου επικοινωνίας *websocket* το οποίο είναι και υπεύθυνο για την σύνδεση με το δίκτυο και την ενημέρωση σε πραγματικό χρόνο.

Η διαδικασία υλοποίησης του συστήματος αποτελείται από την εγκατάσταση και παραμετροποίηση των βιβλιοθηκών την μορφοποίηση και δημιουργία ιστοσελίδας, την δημιουργία *web* σύνδεσης άλλα και την αυτοματοποίηση και διαδραστικότητα μέσω χρήση *scripts*.

Στο παρακάτω κείμενο αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν για την επίτευξη του συστήματος αυτού.

Επιπλέον αναλύεται το σκεπτικό της επιλογής των εργαλείων που χρησιμοποιήθηκαν.

Συνολικά το τελικό αποτέλεσμα μετά την υλοποίηση του συστήματος αυτού αποδεικνύει ότι μπορεί να πραγματοποιηθεί ένα σύγχρονο σύστημα ασφαλείας με χαμηλό κόστος το οποίο μάλιστα μπορεί να ενσωματώσει και τεχνολογίες *AI* όπως αναγνώριση προσώπων *cloud sharing* αλλά και να *smart home projects* και άλλες ποικίλες εφαρμογές του *iot*.

Abstract

This study focuses on modern IOT Technologies and the possibilities they offer through applications that will improve and simplify the everyday life of the society.

The main axes of this study are the creation and implementation of an easy to use and effective security camera system using the Raspberry pi microcomputer.

The system aims to directly monitor the area via the internet and is based on technologies such as motion detection through opencv library. Java was used for the basic program structure. While HTML and javascript were used to create the interactive website.

Finally, communication between client and server is provided by implementing the websocket communication protocol which is also responsive for connecting to the network and updating notifications in real time.

The system implementation process consists of installing and configuring libraries, formatting and creating a website, creating web connection (server), and automation and interactivity through the use of scripts.

The following text analyzes the technologies used to achieve this system. In addition, the rationale for choosing the tools used is analyzed.

Overall, the final result after implementing this system proves that a modern security camera system can be implemented at low cost, which can also integrate AI Technologies such as face recognition, cloud sharing, smart home projects and various IOT applications.

Ευχαριστίες.

Περιεχόμενα

Πίνακας περιεχομένων

ΔΗΛΩΣΗ ΤΗΡΗΣΗΣ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ.....	2
Περίληψη.....	6
Abstract.....	7
Ευχαριστίες.....	8
Περιεχόμενα.....	9
Κατάλογος Σχημάτων.....	10
Κατάλογος Πινάκων.....	11
Κεφάλαιο 1ο: Ιστορική αναδρομή στους υπολογιστές και τα λειτουργικά συστήματα.....	14
1.1 Ιστορική αναδρομή στους Υπολογιστές.....	14
1.2. The First Generation (1945-1955): Vacuum Tubes.....	15
1.2.1 The Second Generation (1955–65): Transistors and Batch Systems.....	16
1.2.2 Η τρίτη γενιά (1965-1980): CCD και προγραμματισμός.....	17
1.2.3 The Fourth Generation (1980–σήμερα): Προσωπικοί Υπολογιστές.....	18
1.3 Ιστορία λειτουργικών συστημάτων.....	18
1.4 Η πέμπτη γενιά Κινητοί υπολογιστές (1990–σήμερα).....	19
Κεφάλαιο 2ο: SBC.....	21
2.1 SBC Single Board Computer.....	21
2.2 Raspberry Pi.....	21
2.3 Μοντέλα Raspberry Pi.....	22
2.4 Λογισμικό Raspberry Pi.....	22
2.5 Εφαρμογές Raspberry Pi.....	23
Κεφάλαιο 3ο: Internet Of Things.....	25
3.1 IoT (Internet Of things).....	25
3.2 SBC Και IOT.....	26
3.3 .SBC Και Μηχανική όραση.....	26
Κεφάλαιο 4ο Κάμερες.....	27
4.1 Κάμερες και ιστορική αναδρομή.....	27
4.2 Κάμερες ασφαλείας.....	29
4.3 Κατηγορίες Καμερών ασφαλείας Σήμερα.....	29
4.4 Εφαρμογές καμερών ασφαλείας σήμερα.....	30
Κεφάλαιο 5ο: 5 Γλώσσες Προγραμματισμού (Programming languages).....	31
Γλώσσες Προγραμματισμού.....	31
5.1 Γλώσσες Προγραμματισμού (Programming language).....	31
Κεφάλαιο 5ο.....	32
Κεφάλαιο 6ο Υλοποίηση Συστήματος – Βιβλιοθήκες και Εργαλεία.....	34
6.1 JAVA.....	35
6.1.1: Πλεονεκτήματα μειονεκτήματα Της JAVA.....	35
6.1.2. JAVA και εφαρμογές.....	37
6.1.3 Χρήση της Java στο σύστημα ασφαλείας.....	38
6.2 .Βιβλιοθήκη OpenCV.....	38
6.2.1 Java και OpenCV.....	39
6.2.2 Δομή της OpenCV.....	40
6.2.3 Gaussian Blur.....	41
6.2.4 Χρήση Opencv και Gaussian Blur στο σύστημα ασφαλείας.....	42
6.3 HTML.....	43

6.4 JavaScript.....	46
6.4.1 Εφαρμογές Javascript.....	46
6.4.2 Χρήση JavaScript στο Σύστημα Ασφαλείας.....	47
6.4.3 Emailjs.....	48
6.5 Communication protocol (Πρωτόκολλα Επικοινωνίας).....	49
6.5.1. IOT και communication protocol.....	49
6.5.2. Κατηγορίες Πρωτοκόλλων Ασφαλείας.....	52
6.6 Websocket Protocol.....	54
6.6.1 Websocket vs HTTP.....	55
6.6.2 Χρήση websocket στο σύστημα ασφαλείας.....	56
6.7 Spark.....	57
Κεφάλαιο 7ο Συμπεράσματα και Προτάσεις βελτιώσης.....	59
7.1 Συμπεράσματα.....	59
7.2 Προτάσεις Βελτίωσης.....	60
Κεφάλαιο 8ο.....	64
Παράρτημα-Πειραματικό μέρος - Επεξήγηση Κώδικα.....	64
Διάγραμμα Ροής Εργασίας.....	67
Παράρτημα Επεξήγησης Κώδικα.....	79

Κατάλογος Σχημάτων

Ευρετήριο Εικόνων

<u>Εικόνα 1 ENIAC First Computer.....</u>	<u>12</u>
<u>Εικόνα 2: Μέθοδος προγραμματισμού σε ένα σύστημα όπως ο IBM7094.....</u>	<u>14</u>
<u>Εικόνα 3: IBM7094.....</u>	<u>14</u>
<u>Εικόνα 4: Raspberry Pi 3 model B.....</u>	<u>18</u>
<u>Εικόνα 5 :Κλάδοι εφαρμογής Raspberry Pi.....</u>	<u>20</u>
<u>Εικόνα 6: Εξέλιξη καμερών.....</u>	<u>25</u>
<u>Εικόνα 7: Δημοφιλέστερες γλώσσες προγραμματισμού 0-100% ανά εξάμηνο(2015-2024)....</u>	<u>30</u>
<u>Εικόνα 8: Εφαρμογή Gaussian Blur μοντέλου σε μια εικόνα.....</u>	<u>38</u>
<u>Εικόνα 9: εισαγωγή κωδικού μέσω script στο HTML αρχείο.....</u>	<u>44</u>
<u>Εικόνα 10Κώδικας για εισαγωγή κωδικού(javascript).....</u>	<u>46</u>
<u>Εικόνα 11Websocket Protocol (server-communication).....</u>	<u>52</u>
<u>Εικόνα 12Websocket Vs HTTP Protocol.....</u>	<u>53</u>
<u>Εικόνα 13Διαγραμμα ροής εργασίας.....</u>	<u>63</u>
<u>Εικόνα 14Emailjs User Interface.....</u>	<u>70</u>

Κατάλογος Πινάκων

Ευρετήριο Εικόνων

<u>Πίνακας 1: Raspberry Pi Models Per Year.....</u>	<u>20</u>
<u>Πίνακας 2: OpenCV Εντολές (σύστημα ασφαλείας).....</u>	<u>34</u>
<u>Πίνακας 3: Εντολές χρήσης Gaussian Blur.....</u>	<u>38</u>
<u>Πίνακας 4: HTML εκδόσεις ανα έτος.....</u>	<u>40</u>
<u>Πίνακας 5: HTML Βασικές εντολές.....</u>	<u>41</u>
<u>Πίνακας 6: Websocket Method.....</u>	<u>44</u>
<u>Πίνακας 7: Βασικές κατηγορίες Πρωτοκόλλων ασφαλείας.....</u>	<u>47</u>
<u>Πίνακας 8: Spark Framework.....</u>	<u>54</u>

Συντομογραφίες

<u>Συντομογραφία</u>	<u>Πλήρης όρος</u>
AI	Artificial Intelligence
IOT	Internet of Things
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
SMTP	Simple Mail Transfer Protocol
HTML	HyperText Markup Language
API	Application Programming Interface
SBC	Single Board Computer
CSS	Cascading Style Sheets
FTP	File Transfer Protocol
CCTV	Closed-Circuit Television
URL	Uniform Resource Locator
CPU	Central Process Unit
RAM	Random Access Memory

Κεφάλαιο 1ο: Ιστορική αναδρομή στους υπολογιστές και τα λειτουργικά συστήματα

1.1 Ιστορική αναδρομή στους Υπολογιστές

Ένας υπολογιστής είναι μια ηλεκτρονική μηχανή που δέχεται πληροφορίες, τις αποθηκεύει, τις επεξεργάζεται σύμφωνα με τις οδηγίες που παρέχονται από έναν χρήστη και στη συνέχεια επιστρέφει το αποτέλεσμα. Οι προσπάθειες των ανθρώπων να αναπτύξουν ένα εργαλείο χειρισμού δεδομένων χρονολογούνται από το 2600 π.Χ., όταν οι Κινέζοι βρήκαν τον άβακα. Ο άβακας δεν ήταν κάτι περισσότερο από μια αριθμομηχανή που χρησιμοποιούνταν για την εκτέλεση των βασικών πράξεων όπως πρόσθεση, αφαίρεση και πολλαπλασιασμό.

Από τότε η τεχνολογία είχε μείνει σε ένα πρωτόγονο στάδιο εξέλιξης ως που το 1801 ο Joseph-Marie Masquard ήρθε να αλλάξει την τεχνολογική πρόοδο με την ανακάλυψη των διάτρητων καρτών. Οι διάτρητες κάρτες ήταν ένα μέσο αποθήκευσης δεδομένων που χρησιμοποιήθηκε στην πρώιμη υπολογιστική και επεξεργασία δεδομένων. Οι κάρτες ήταν συνήθως κατασκευασμένες από ανθεκτικό χαρτί και είχαν ορθογώνιες τρύπες σε συγκεκριμένες θέσεις για να αναπαραστήσουν δεδομένα. Οι διάτρητες κάρτες χρησιμοποιήθηκαν για την αποθήκευση δεδομένων όπως κείμενο, αριθμητικές τιμές και άλλες πληροφορίες και υποβλήθηκαν σε επεξεργασία από μηχανήματα που ονομάζονταν συσκευές ανάγνωσης καρτών.

Οι συσκευές ανάγνωσης καρτών μπορούσαν να διαβάσουν το μοτίβο των διάτρητων οπών στις κάρτες και να χρησιμοποιήσουν αυτές τις πληροφορίες για να εκτελέσουν διάφορες εργασίες, όπως ταξινόμηση, ταξινόμηση και υπολογισμό δεδομένων.

Στην πράξη η πρώτη υλοποίηση της χρήσης διάτρητων σε κατασκευή έγινε το 1835 από τον Άγγλο μαθηματικό Charles Babbage ο οποίος συνδύασε τις διάτρητες κάρτες με μια ατμομηχανή για να εφεύρει αυτό που ονόμασε «αναλυτική μηχανή».

Δημιούργησε μια αναλυτική μηχανή, η οποία μπορούσε να προγραμματιστεί με διάτρητες κάρτες για να πραγματοποιήσει υπολογισμούς. Η μεγαλύτερη και σημαντικότερη διαφορά σε σχέση με προηγούμενες συσκευές ήταν ότι η μηχανή του Charles Babbage ήταν σε θέση να λαμβάνει αποφάσεις με βάση τους δικούς της υπολογισμούς. Σχεδόν όλοι οι υπολογιστές που χρησιμοποιούνται σήμερα ακολουθούν αυτή τη βασική ιδέα που διατύπωσε ο Babbage, γι' αυτό και αναφέρεται συχνά ως «ο πατέρας των υπολογιστών».

[4]Tanenbaum, A. S. (2013). Modern Operating Systems (3rd Edition), Pearson.

1.2. The First Generation (1945-1955): Vacuum Tubes

Μετά τις ανεπιτυχείς προσπάθειες του Babbage σημειώθηκε μικρή πρόοδος στην κατασκευή ψηφιακών υπολογιστών μέχρι την περίοδο του Β' Παγκοσμίου Πολέμου, η οποία προκάλεσε μια εκρηκτική δραστηριότητα. Ο καθηγητής John Atanasoff και ο μεταπτυχιακός φοιτητής του Clifford Berry έχτισαν αυτό που θεωρείται τώρα ως ο πρώτος λειτουργικός ψηφιακός υπολογιστής στο Πολιτειακό Πανεπιστήμιο της Αϊόβα. Χρησιμοποιούσε 300 λυχνίες κενού(vacuum tubes).

Την ίδια περίπου εποχή και λόγω της έκτασης του πολέμου σε παγκόσμιο επίπεδο οι χώρες προσπαθούσαν να αποσπάσουν δεδομένα μέσω της τεχνολογίας και των υπολογιστών έτσι την ίδια χρονική περίοδο οι Γερμανοί με τον Konrad Zuse κατασκεύασαν τον Z3 .Ταυτόχρονα όμως οι Βρετανοί καθώς και οι Αμερικάνοι κατασκεύασαν τον υπολογιστή Colossus και τον ENIAC αντιστοίχως για να σπάσουν τους κρυπτογραφημένους γερμανικούς κώδικες.

Οι υπολογιστές αυτοί ήταν ογκώδεις και ζύγιζαν τόνους.

Οι πρώτες αριθμομηχανές εμφανίστηκαν τη δεκαετία του 1930.Μετά την εξέλιξη των υπολογιστικών αυτών μηχανών και κατά της ανάπτυξη της τεχνολογίας στο β παγκόσμιο πόλεμο η εποχή της ψηφιοποίησης άρχισε να ακμάζει.

Ο πρώτος υπολογιστής που κατασκευάστηκε για τις μάζες ήταν το UNIVAC, που κατασκευάστηκε από τον Remington Rand το 1951.Η εταιρεία IBM αναπτύχθηκε μετά από την εφεύρεση του tabulator, που δημιουργήθηκε από τον Herman Hollerith στα τέλη της δεκαετίας του 1880 και παρουσίασε τον πρώτο κεντρικό υπολογιστή της 1952. Οι πρώτοι υπολογιστές Remington είχαν πολύ υψηλό κόστος το οποίο άγγιζε το 1 εκατομμύριο ανά μηχανήμα αλλά η IBM έφτιαξε μικρότερα, πιο προσιτά μηχανήματα που έγιναν αργότερα πιο δημοφιλή. [1]

Το 1954 η IBM ανέπτυξε την Fortran, μια από τις αρχικές γλώσσες προγραμματισμού υπολογιστών, βασισμένη σε μεγάλο βαθμό στα μαθηματικά. Την ίδια δεκαετία, οι εξελίξεις του τρανζίστορ, των ολοκληρωμένων κυκλωμάτων και του μικρο προγραμματισμού οδήγησαν στη μείωση του μεγέθους του υπολογιστή. Εν τω μεταξύ, οι CPU αύξησαν την ταχύτητα επεξεργασίας του υπολογιστή και η μνήμη βελτίωσε την αποθήκευση δεδομένων.

Η άφιξη των μικροεπεξεργαστών που εισήγαγαν η Texas Instruments και η Intel στις αρχές της δεκαετίας του 1970 άνοιξε το δρόμο για μικροεπεξεργαστές ακόμα πιο ισχυρούς υπολογιστές.



1.2.1 The Second Generation (1955–65): Transistors and Batch Systems

Το τρανζίστορ είναι ένας κρύσταλλος με τρεις περιοχές εμπλουτισμένες με προσμίξεις. Η εισαγωγή του τρανζίστορ στα μέσα της δεκαετίας του 1950 άλλαξε ριζικά την πορεία της τεχνολογικής προόδου.

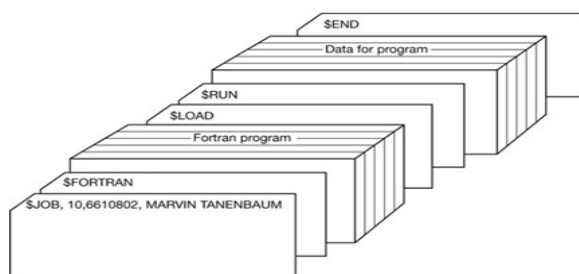
Οι υπολογιστές έγιναν αρκετά αξιόπιστοι ώστε να μπορούν να κατασκευαστούν και να πωληθούν να πληρώνουν πελάτες με την προσδοκία ότι θα συνεχίσουν να λειτουργούν για μεγάλο χρονικό διάστημα σε αντίθεση με τους υπολογιστές λυχνίας οι οποίοι θέλαν και αντικατάσταση ανά τακτά χρονικά διαστήματα.

Για πρώτη φορά, υπήρξε σαφής διαχωρισμός μεταξύ σχεδιαστών, κατασκευαστών, χειριστών, προγραμματιστών και προσωπικού συντήρησης. Αυτά τα μηχανήματα, που τώρα ονομάζονται mainframes, ήταν κλειδωμένα σε μεγάλα, ειδικά κλιματιζόμενα δωμάτια υπολογιστών, με επιτελεία επαγγελματιών χειριστών για τη λειτουργία τους.

Τα τρανζίστορ όχι μόνο ήταν μικρότερα και απαιτούσαν τη χρήση λιγότερης θερμότητας, ήταν επίσης λιγότερο επιρρεπή σε αστοχίες από τους υπολογιστές που κατασκευάζονταν με tubes. Ως αποτέλεσμα, τα τρανζίστορ επέτρεψαν στους υπολογιστές να γίνουν μικρότεροι, ταχύτεροι και λιγότερο ακριβοί από τους υπολογιστές πρώτης γενιάς. Παρόλα αυτά συνέχισαν να είναι προσιτοί μόνο σε μεγάλες εταιρείες πανεπιστήμια ή κρατικούς φορείς. Διότι το κόστος τους ανέρχονταν σε εκατομμύρια δολάρια για ένα μηχάνημα.

Μετα την ένταξη της Fortran από την IBM ξεκίνησε και η ενσωμάτωση των προγραμμάτων στους υπολογιστές. Η Διαδικασία άρχιζε με τον προγραμματιστή ο οποίος έγραφε τον κώδικα σε χαρτί σε (Fortran ή assembly). Και την έδινε σε έναν χειριστή που βρισκόταν στο δωμάτιο ο οποίος έκανε και την σύνδεση στο μηχάνημα.[1]

Τέλος ο έλεγχος του προγράμματος διαρκούσε πολλές ώρες και ήθελε πολύ εργατικό δυναμικό για την πραγματοποίησή του. Μεγάλοι υπολογιστές δεύτερης γενιάς χρησιμοποιήθηκαν κυρίως για επιστημονικούς και μηχανικούς υπολογισμούς, όπως η επίλυση των μερικών διαφορικών εξισώσεων που εμφανίζονται συχνά στη φυσική και τη μηχανική.



Ένα παράδειγμα υπολογιστή εκείνης της εποχής θα αποτελούσε ο IBM 1401. Ο IBM 1401 ήταν ένας υπολογιστής μεσαίας κατηγορίας που παρουσιάστηκε από την IBM το 1959. Ήταν ένας από τους πιο επιτυχημένους υπολογιστές της εποχής του και έπαιξε σημαντικό ρόλο στην επανάσταση των υπολογιστών της δεκαετίας του 1960. Σχεδιάστηκε για επαγγελματική χρήση και χρησιμοποιήθηκε ευρέως για επεξεργασία δεδομένων και άλλες επιχειρηματικές εφαρμογές. Η IBM 1401 είχε μέγιστη μνήμη 16.000 χαρακτήρων και ταχύτητα επεξεργασίας έως και 85.000 εντολές ανά δευτερόλεπτο. Χρησιμοποιούσε μνήμη μαγνητικού πυρήνα και οι συσκευές εισόδου/εξόδου του περιελάμβαναν συσκευές ανάγνωσης καρτών με διάτρηση και διατρήσεις, οδηγούς ταινίας και εκτυπωτές γραμμής. Χρησιμοποιούσε τον εκτυπωτή IBM 1403, ο οποίος ήταν ένας αλυσιδωτός εκτυπωτής υψηλής ταχύτητας ικανός να εκτυπώνει έως και 600 γραμμές ανά λεπτό.

1.2.2 Η τρίτη γενιά (1965-1980): CCD και προγραμματισμός

Στις αρχές της δεκαετίας του 1960, οι περισσότεροι κατασκευαστές υπολογιστών είχαν δύο ξεχωριστές, ασύμβατες, σειρές προϊόντων. Από τη μία πλευρά, υπήρχαν οι επιστημονικοί υπολογιστές μεγάλης κλίμακας με γνώμονα τις λέξεις, όπως ο 7094, που χρησιμοποιήθηκαν για υπολογισμούς βιομηχανικής αντοχής στην επιστήμη και τη μηχανική. Από την άλλη, υπήρχαν οι προσανατολισμένοι 1401 που ήταν κυρίως για βιομηχανικές εφαρμογές. Η ανάπτυξη και η διατήρηση δύο εντελώς διαφορετικών σειρών προϊόντων ήταν μια δαπανηρή πρόταση για τους κατασκευαστές. Επιπλέον, πολλοί νέοι πελάτες υπολογιστών χρειάζονταν αρχικά ένα μικρό μηχάνημα.

Η IBM προσπάθησε να λύσει και τα δύο αυτά προβλήματα με μία μόνο διαδρομή, εισάγοντας το Systems/360. Το 360 ήταν μια σειρά μηχανών συμβατών με λογισμικό που κυμαίνονταν από μοντέλα μεγέθους 1401 έως πολύ μεγαλύτερα, πιο ισχυρά από τα πανίσχυρα 7094. Όλη η σειρά των υπολογιστών αυτών είχε το ίδιο σύνολο αρχιτεκτονικής και εντολών με μόνες διαφορές ανάλογα με την τιμή να αποτελούν (μέγιστη μνήμη, ταχύτητα επεξεργαστή, αριθμός επιτρεπόμενων συσκευών εισόδου/εξόδου). Το επίσης θετικό σημείο ήταν ότι τα γραμμένα προγράμματα για ένα μηχάνημα θα μπορούσαν τρέχουν με όλα τα άλλα.

Ο 360 σχεδιάστηκε για να χειρίζεται τόσο επιστημονικούς (δη. Αριθμητικούς) όσο και εμπορικούς υπολογιστές, μια ενιαία οικογένεια μηχανών θα μπορούσε να ικανοποιήσει τις ανάγκες όλων των πελατών. Το IBM 360 ήταν η πρώτη μεγάλη σειρά υπολογιστών που χρησιμοποίησε (μικρής κλίμακας) CCD (integrate Circuit), παρέχοντας έτσι ένα σημαντικό πλεονέκτημα τιμής/απόδοσης έναντι του μηχανές δεύτερης γενιάς, οι οποίες κατασκευάστηκαν από μεμονωμένα τρανζίστορ.

Τα επόμενα χρόνια, η IBM κυκλοφόρησε παρόμοιας αρχιτεκτονικής ηλεκτρονικούς υπολογιστές βασισμένους στον 360, χρησιμοποιώντας πιο σύγχρονη τεχνολογία, γνωστή ως 370, 4300, 3080 και 3090. Η zSeries είναι ο πιο πρόσφατος απόγονος αυτής της γραμμής, αν και έχει αποκλίνει σημαντικά από το πρωτότυπο.

1.2.3 The Fourth Generation (1980–σήμερα): Προσωπικοί Υπολογιστές

Με την ανάπτυξη των κυκλωμάτων LSI (Large Scale Integration) τσιπ που περιέχουν χιλιάδες τρανζίστορ σε ένα τετραγωνικό εκατοστό πυριτίου η ιδέα του προσωπικού υπολογιστή άρχισε να γίνεται πραγματικότητα.

Το 1974 η Intel κυκλοφόρησε τον πρώτο μικροεπεξεργαστή (GPU) γενικής χρήσης 8-bit τον 8080. Παρόλα αυτά η χρήση ενός λειτουργικού συστήματος ήταν απαραίτητη έτσι ώστε ο 8080 να μπορεί να δοκιμαστεί. Για αυτό τον λόγο η Intel ζήτησε από έναν από τους συμβούλους της, τον Gary Kildall, να δημιουργήσει ένα λειτουργικό σύστημα. Αυτό και έγινε λοιπόν ο Gary Kildal έγραψε ένα λειτουργικό σύστημα βασισμένο σε δίσκο που ονομάζεται CP/M (ελεγχόμενο Πρόγραμμα για μικροϋπολογιστές). Το 1977, η Digital Research ξαναέγραψε το CP/M για να το κάνει κατάλληλο για εκτέλεση σε πολλούς μικροϋπολογιστές που χρησιμοποιούν το 8080.

Το CP/M (Πρόγραμμα Ελέγχου για Μικροϋπολογιστές) ήταν ένα δημοφιλές λειτουργικό σύστημα για πρώιμους μικροϋπολογιστές και κυκλοφόρησε για πρώτη φορά το 1974 για τον μικροεπεξεργαστή Intel 8080 8-bit. Το CP/M παρείχε μια τυπική πλατφόρμα για την εκτέλεση λογισμικού σε μια ποικιλία διαφορετικών μικροϋπολογιστών και χρησιμοποιήθηκε ευρέως σε επιχειρηματικούς και εκπαιδευτικούς χώρους. Σχεδιάστηκε για να είναι ένα απλό και ευέλικτο λειτουργικό σύστημα που θα μπορούσε να λειτουργεί σε μια ποικιλία πλατφορμών υλικού.

Παρείχε βασικές δυνατότητες όπως διαχείριση αρχείων, μορφοποίηση δίσκου και υποστήριξη για εκτύπωση και είσοδο/έξοδο τερματικού. Ένα από τα βασικά χαρακτηριστικά του CP/M ήταν η υποστήριξή του για ένα ευρύ φάσμα εφαρμογών λογισμικού, οι οποίες μπορούσαν να γραφτούν σε διάφορες γλώσσες προγραμματισμού. Το CP/M έπαιξε σημαντικό ρόλο στην πρώιμη ιστορία των μικροϋπολογιστών και άνοιξε το δρόμο για την ανάπτυξη μεταγενέστερων λειτουργικών συστημάτων όπως το Unix και το Linux .Τέλος κατείχε το σύνολο των αγορών για περίπου 5 χρόνια.

1.3 Ιστορία λειτουργικών συστημάτων

Το λειτουργικό σύστημα (OS) είναι ένα πρόγραμμα λογισμικού που διαχειρίζεται πόρους υλικού υπολογιστή και παρέχει κοινές υπηρεσίες για προγράμματα υπολογιστών. Αποτελεί τη ραχοκοκαλιά ενός συστήματος υπολογιστών και επιτρέπει στα στοιχεία υλικού και τις εφαρμογές λογισμικού να επικοινωνούν μεταξύ τους.

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο όλα άρχισαν την Δεκαετία του 1950. Τα πρώτα λειτουργικά συστήματα αναπτύχθηκαν για μεγάλους υπολογιστές (mainframe), όπως το UNIVAC και το IBM 701. Αυτά τα πρώιμα λειτουργικά συστήματα ήταν απλά και εκτελούσαν βασικές λειτουργίες όπως φόρτωση και εκτέλεση προγραμμάτων .Την Δεκαετία του 1970 και με την εμφάνιση των προσωπικών υπολογιστών λειτουργικά συστήματα όπως το CP/M και το MS-DOS αναπτύχθηκαν για τη διαχείριση και την επικοινωνία των τμημάτων των υπολογιστών .Σημαντικό λειτουργικό σύστημα εκείνης της δεκαετίας αποτέλεσε και το Unix , το οποίο αποτελεί και την βάση για πολλά σύγχρονα λειτουργικά συστήματα..

Στις αρχές της δεκαετίας του 1980, η IBM σχεδίασε το IBM PC και έψαξε τριγύρω για λογισμικό να τρέξει πάνω του. Άτομα της IBM επικοινωνήσαν με τον Μπιλ Γκέιτς για να λάβουν άδεια για να χρησιμοποιήσουν διερμηνέα BASIC του. Ο Μπιλ Γκέιτς είναι ένας Αμερικανός επιχειρηματίας που ίδρυσε την Microsoft Corporation το 1975. Τον ρώτησαν επίσης αν γνώριζε κάποιο λειτουργικό σύστημα που να τρέχει στον υπολογιστή .Ο Γκέιτς πρότεινε στην IBM να επικοινωνήσει με την Digital Research, (τότε η κυρίαρχη εταιρεία λειτουργικών συστημάτων στον κόσμο), Ο Κίνταλ αρνήθηκε να συναντηθεί με την IBM, στέλνοντας έναν υφιστάμενο. Κατά συνέπεια, η IBM επέστρεψε στο Μπιλ Γκέιτς ρωτώντας τον αν θα

μπορούσε να τους παράσχει ένα λειτουργικό σύστημα. Ο Gates συνειδητοποίησε ότι ένας τοπικός κατασκευαστής υπολογιστών, Η Seattle Computer Products, διέθετε κατάλληλο λειτουργικό σύστημα, το DOS (Disk Operating System). Τους πλησίασε και ζήτησε να το αγοράσει. Πρόταση την οποία η εταιρία αποδέχτηκε πρόθυμα. Στη συνέχεια, ο Gates πρόσφερε στην IBM ένα πακέτο DOS/BASIC, την οποία αποδέχτηκε η IBM. Η IBM ήθελε ορισμένες τροποποιήσεις, έτσι ο Γκέιτς προσέλαβε το άτομο που έγραψε το DOS, τον Tim Paterson, για να τα φτιάξει. Το αναθεωρημένο σύστημα μετονομάστηκε σε MS-DOS (MicroSoft Disk Operating System) και γρήγορα κυριάρχησε στην αγορά Η/Υ της IBM.

Η αρχική έκδοση του MS-DOS ήταν αρκετά πρωτόγονη, οι επόμενες εκδόσεις περιλάμβαναν περισσότερα προηγμένα χαρακτηριστικά, συμπεριλαμβανομένων πολλών από το UNIX. Τα CP/M και MS-DOS και άλλα παρόμοια λειτουργικά συστήματα για πρώιμους μικροϋπολογιστές ήταν βασισμένα σε εντολές που χρησιμοποιούσαν οι χρήστες από το πληκτρολόγιο.

Αυτό τελικά άλλαξε λόγω της έρευνας που έγινε από τον Doug Engelbart στο Stanford Research Institute την δεκαετία του 1960. Ο Engelbart εφεύρε τη γραφική διεπαφή χρήστη, πλήρης με παράθυρα, εικονίδια, μενού και ποντίκι. Αυτές οι ιδέες υιοθετήθηκαν από ερευνητές της εταιρείας Xerox Parc και ενσωματώθηκαν στους υπολογιστές της.

Την Συνέχεια έκανε ο Steve Jobs εφευρέτης του υπολογιστή Apple. Ο Steve Jobs μετά από επίσκεψη στην εταιρεία PARC είδε ένα (GUI) υπολογιστή με γραφικές διεπαφές χρήστη με το περιβάλλον. Τότε συνειδητοποίησε αμέσως την πιθανή του αξία του (GUI) και ξεκίνησε την ενσωμάτωση της τεχνολογίας αυτής σε Apple υπολογιστές. Αυτό το έργο οδήγησε στη Λίζα, η οποία ήταν πολύ ακριβό και απέτυχε εμπορικά. Η δεύτερη προσπάθεια του Jobs, το Apple Macintosh, ήταν τεράστια επιτυχία, όχι μόνο επειδή ήταν πολύ φθηνότερο από το Lisa, αλλά και επειδή ήταν φιλικό προς το χρήστη, δηλαδή προοριζόταν για χρήστες που όχι μόνο δεν ήξερε τίποτα για τους υπολογιστές αλλά επιπλέον

δεν είχε καμία απολύτως πρόθεση να μάθει. Τέλος υιοθέτησε Το OS X ένα λειτουργικό σύστημα που βασίζεται στο UNIX με διαδραστικό γραφικό περιβάλλον εύκολο στην χρήση.

Στα τέλη του 90 σειρά της είχε η Microsoft η οποία μετά την επιτυχία του Macintosh δημιούργησε έναν διάδοχο του του MS-DOS ονομαζόμενο windows αρχικά λειτουργούσε πάνω από το MS-DOS στην συνέχεια όμως αναπτύχθηκε ανεξάρτητη έκδοση, τα Windows 95, τα οποία ενσωμάτωσαν πολλές από τις παλιές λειτουργίες μόνο για εκκίνηση και εκτέλεση παλαιών προγραμμάτων MS-DOS. Το 1998, κυκλοφόρησε μια ελαφρώς τροποποιημένη έκδοση αυτού του συστήματος, που ονομάζεται Windows 98. Η windows συνέχισε με πολλές διαφορετικές εκδόσεις (2000,.xp,visa,10,11). Και αποτελεί την κορυφαία εταιρεία λειτουργικών συστημάτων ως σήμερα.

1.4 Η πέμπτη γενιά Κινητοί υπολογιστές (1990-σήμερα).

Μετά την δημιουργία των προσωπικών υπολογιστών και την μεγάλη ανάπτυξη των Ισι και των ολοκληρωμένων κυκλωμάτων(integrated circuit). Στα μέσα τις δεκαετίας του 90 η Nokia κυκλοφορεί για πρώτη φορά το N9000, το οποίο συνδύαζε κυριολεκτικά δύο, ως επί το πλείστον ξεχωριστές συσκευές: ένα τηλέφωνο και ένα PDA (Personal Digital Assistant).

Το 1981 κυκλοφόρησε ο Osborne 1, ο οποίος θεωρείται ευρέως ως ο πρώτος εμπορικά επιτυχημένος φορητός υπολογιστής. Ζύγιζε 24 κιλά και είχε οθόνη 5 ιντσών, αλλά ήταν ακόμα σημαντικά μικρότερος. Η δεκαετία του 1990 είδε την άνοδο του φορητού υπολογιστή, με εταιρείες όπως η IBM, η Compaq και η

Toshiba να κυκλοφορούν όλο και πιο ισχυρά και συμπαγή μοντέλα. Αυτοί οι πρώτοι φορητοί υπολογιστές εξακολουθούσαν να είναι σχετικά βαρείς και ακριβοί, αλλά γρήγορα έγιναν δημοφιλείς στους επιχειρηματίες οι οποίοι εργάζονταν εν κινήσει.

Στα τέλη της δεκαετίας του 1990 παρατηρήθηκε επίσης η άνοδος των προσωπικών ψηφιακών βοηθών(PDA), οι οποίοι ήταν μικρές συσκευές χειρός που μπορούσαν να χρησιμοποιηθούν για βασικές εργασίες υπολογιστών και οργάνωση πληροφοριών. Εταιρείες όπως η Palm και η Apple ήταν απο τις πιο βασικές εκείνη την περίοδο λαμβάνοντας μεγαλύτερο μερίδιο της αγοράς.

Στη δεκαετία του 2000, η εμφάνιση των smartphones και των υπολογιστών tablet έφερε επανάσταση στη βιομηχανία φορητών υπολογιστών. Το πρώτο smartphone, το IBM Simon, κυκλοφόρησε το 1993, αλλά μόνο όταν κυκλοφόρησε το iPhone το 2007, τα smartphone απογειώθηκαν. Τα tablet, τα οποία είναι ουσιαστικά μεγαλύτερες εκδόσεις smartphone, παρουσιάστηκαν για πρώτη φορά στις αρχές της δεκαετίας του 2000, αλλά μόνο όταν κυκλοφόρησε το iPad το 2010 έγιναν πραγματικά δημοφιλή. Σήμερα, ο φορητός υπολογιστής είναι πανταχού παρών, με smartphone και tablet που χρησιμοποιούνται από ανθρώπους σε όλο τον κόσμο για τα πάντα, από ψυχαγωγία και κοινωνικές συναναστροφές μέχρι εργασία και εκπαίδευση. Η τεχνολογία συνεχίζει να εξελίσσεται, με καινοτομίες όπως τα πτυσσόμενα smartphone και η επαυξημένη πραγματικότητα να προσφέρουν νέες δυνατότητες για το μέλλον των φορητών υπολογιστών.[1]

Κεφάλαιο 2ο:SBC

2.1 SBC Single Board Computer

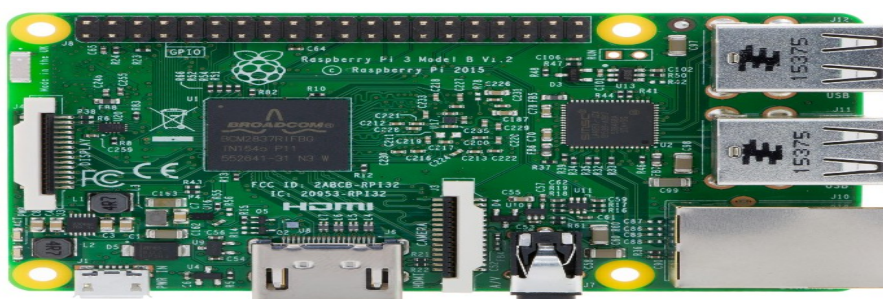
[2] 2022 2nd International Conference on Advanced Electrical Engineering (ICAEE)

Όπως Αναφέρθηκε στην προηγούμενη ενότητα η όλο ένα και μεγαλύτερη πρόοδος της τεχνολογίας καθώς και η εύρεση υλικών και τεχνογνωσίας άλλα και η τάση των εταιριών να εστιάζει σε μικρότερες πιο πρακτικές άλλα και πιο οικονομικές συσκευές οδήγησε στα SBC (Single Board Computer) .Ένας υπολογιστής μονής πλακέτας (SBC) είναι ένας πλήρης υπολογιστής κατασκευασμένος σε μία πλακέτα κυκλώματος, με μικροεπεξεργαστές, μνήμη, διεπαφές εισόδου/εξόδου και άλλα χαρακτηριστικά που απαιτούνται για έναν λειτουργικό υπολογιστή. Τα SBC χρησιμοποιούνται συχνά για εκπαιδευτικές, χομπίστες ή ενσωματωμένες εφαρμογές υπολογιστών. Παραδείγματα δημοφιλών SBC περιλαμβάνουν τα Raspberry Pi, BeagleBone και Arduino. Οι σημερινοί υπολογιστές Single Board (SBC) είναι αρκετά ισχυροί για να χειρίζονται κοινά λειτουργικά συστήματα και φόρτους εργασίας. Κυρίως λόγω του χαμηλού κόστους, του μικρού μεγέθους, της φορτικότητας και των χαμηλών απαιτήσεων ψύξης. Οι εφαρμογές SBC έχουν αυξηθεί τα τελευταία χρόνια και εξακολουθούν να έχουν συντριπτικό ενδιαφέρον λόγω της αναλογίας κόστους/απόδοσης. Αυτό κατέστησε αυτά τα συμπλέγματα SBC πιο κατάλληλα από τα παραδοσιακά γνωστά μηχανήματα συμπλέγματος σε διάφορες εφαρμογές τομέα, όπως το Edge Computing (EC), την Τεχνητή Νοημοσύνη (AI), το Internet of Things (IoT) και την Υγεία. [2]

2.2 Raspberry Pi

Το Raspberry pi αποτελεί βασικό πυλώνα των SBC (βλ ΚΕΦ2.1.1) .Ουσιαστικά αποτελεί ένα πλήρη υπολογιστή στο μέγεθος μιας πιστωτικής κάρτας .Το Raspberry Pi σχεδιάστηκε και αναπτύχθηκε από το Ίδρυμα Raspberry Pi Το έργο της υλοποίησης του Raspberry Pi ξεκίνησε το 2006 από τους Eben Urton, Rob Mullins, Jack Lang και Alan Mycroft στο Εργαστήριο Υπολογιστών του Πανεπιστημίου του Cambridge. Βασίζεται σε Linux λογισμικό και κυκλοφόρησε για πρώτη φορά στην αγορά το 2012 από τότε, έχει γίνει μια δημοφιλής συσκευή για διάφορες εφαρμογές, όπως εκπαίδευση, έργα χόμπι, οικιακούς αυτοματισμούς και βιομηχανικές χρήσεις. Λόγο του χαμηλού κόστους άλλα και της ευελιξίας του. Το Raspberry Pi έχει γίνει ένα σημαντικό εργαλείο για την εκμάθηση, τη δημιουργία καινοτόμων εφαρμογών. Το Raspberry Pi χάρη στο λειτουργικό τους σύστημα μπορεί να προγραμματιστεί χρησιμοποιώντας διάφορες γλώσσες προγραμματισμού, όπως Python, Java και C++. Οι πλακέτες Raspberry Pi διαθέτουν μια σειρά επιλογών εισόδου και εξόδου, συμπεριλαμβανομένων των ακίδων GPIO (Είσοδος/Εξόδος Γενικής Χρήσης), HDMI, USB και Ethernet..

Εικόν



α 4: Raspberry Pi 3 model B

SBC Single Board Computer

2.3 Μοντέλα Raspberry Pi

Το 2012 κυκλοφόρησε το πρώτο Raspberry Pi Model B. Είχε ένα Broadcom BCM2835 system-on-chip (SoC) με επεξεργαστή 700 MHz ARM11 και 512 MB RAM. Το Modem Ή είχε διάφορες επιλογές εισόδου και εξόδου, συμπεριλαμβανομένων των ακίδων USB, Ethernet, HDMI και GPIO.

Η πρώτη πώληση έγινε έναντι 35 \$ και γρήγορα έγινε δημοφιλής στους χομπίστες και τους εκπαιδευτικούς. Έκτοτε, το Raspberry Pi Foundation έχει κυκλοφορήσει πολλά μοντέλα του Raspberry Pi, συμπεριλαμβανομένων των Model A, Model B+, Model 2, Model 3 και Model 4. Κάθε μοντέλο έχει βελτιωμένες προδιαγραφές και χαρακτηριστικά, όπως ταχύτερους επεξεργαστές, περισσότερη μνήμη RAM, και πρόσθετες επιλογές εισόδου και εξόδου.(Βλ πίνακα 1).Στο συγκεκριμένο πρότζεκτ χρησιμοποιήθηκε το RaspberryPi 3 Model B+(2018) .

Το Raspberry Pi 3Model B επιλέχθηκε διότι σε σύγκριση με τα προηγούμενα μοντέλα προσφέρει ταχύτερο επεξεργαστή (ARMv8 quad core με συχνότητα λειτουργίας στα 1.4GHz) και διπλάσια μνήμη, που φτάνει συνολικά το 1 GB. Επιπλέον το Raspberry Pi 3 Model B παρέχει δυνατότητα ασύρματης σύνδεσης μέσω Bluetooth και WLAN. Λόγω του επεξεργαστή που διαθέτει .Η δυνατότητα της ασύρματης σύνδεσης στο διαδίκτυο (WLAN το κάνει ιδανικό για χρήση σε εφαρμογές όπως αυτοματισμούς σπιτιού απομακρυσμένες παρακολουθήσεις χώρων Και γενικά Internet εφαρμογές.

2.4 Λογισμικό Raspberry Pi

Κατά τον σχεδιασμό του sbc Raspberry pi απαραίτητη ήταν η δημιουργία ενός λογισμικού το οποίο θα ήταν ελαφρύ για να μπορεί να λειτουργήσει σωστά βάση των προδιαγραφών του πρώτο μοντέλου το οποίο δεν διακατείχε μεγάλες αποδόσεις στο hardware κομμάτι .Έτσι το 2012 δημιουργήθηκε το Raspbian.

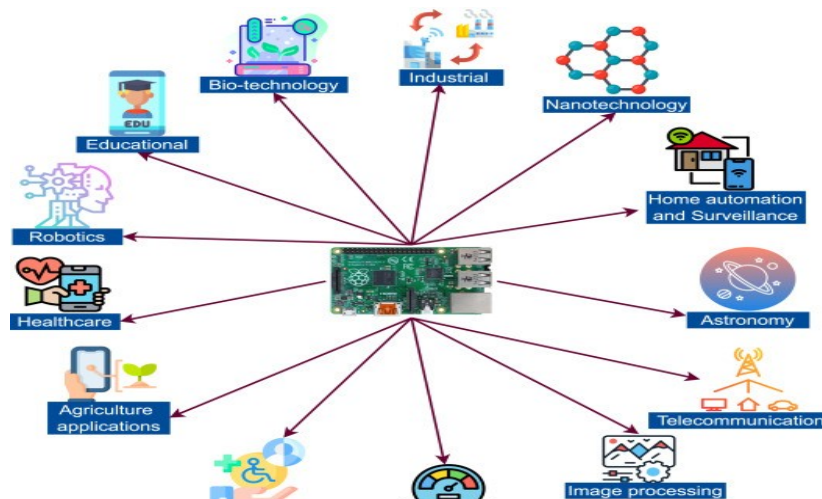
Το Raspbian είναι ένα δωρεάν λειτουργικό σύστημα που βασίστηκε στη διανομή Debian Linux που τροποποιήθηκε για να είναι πιο αποδοτική για υπολογιστές μίας πλακέτας Raspberry Pi. Είναι χτισμένο πάνω στο Debian, μια από τις πιο δημοφιλείς και σταθερές μορφές Linux .Το οποίο παρέχει πρόσβαση σε μια τεράστια συλλογή από προ μεταγλωτισμένα πακέτα λογισμικού. Τέλος διαθέτει ένα φιλικό προς το χρήστη περιβάλλον επιφάνειας εργασίας που ονομάζεται PIXEL (Pi Improved Xwindows Environment Lightweight).

Το PIXEL παρέχει μια οικεία γραφική διεπαφή(GUI), γραμμή εργασιών, μενού έναρξης και διάφορες εφαρμογές, καθιστώντας το εύκολο στη χρήση .Παρόλα αυτά η εξέλιξη του Raspberry pi και η αναβάθμιση των μοντέλων στο κομμάτι του hardware οδήγησαν στην αντικατάσταση του Raspbian από το Raspberry Pi OS το οποίο παρότι βασίζεται στην Debian όπως και το Raspbian, διαθέτει ορισμένες ενημερώσεις και βελτιώσεις διατηρώντας όμως όλα τα θετικά του Raspbian.

2.5 Εφαρμογές Raspberry Pi

Το Raspberry Pi Λόγο του μικρού μεγέθους και κατανάλωσης του σε ενέργεια καθώς και με την χρήση του ελαφριού linux based λειτουργικού του συστήματος μπορεί να χρησιμοποιηθεί σε εφαρμογές όπως

- Συστήματα Ασφαλείας (smart θυροτηλέφωνο χρήση προσώπου, ανίχνευση κίνησης, κάμερα ασφαλείας)
- Δίκτυα και Servers (VPN server, FTP server, Web server)
- Home automation (έλεγχος φωτισμού υγρασίας καθώς και άρδευση κήπου)
- Ρομποτική και Projects (αυτόνομα οχήματα, χειρισμός ρομπότ και drone)
- Εκπαίδευση και Προγραμματισμός
- Σταθμοί Μέτρησης και Καταγραφής (Μετεωρολογικός σταθμός, βιομηχανικός μετρητής)
- Άλλες Εφαρμογές και αυτοματισμοί



Κεφάλαιο 2ο

Πίνακας 1: Raspberry Pi Models Per Year

Μοντέλο / Έτος	USB Ports	CPU	RAM	Bluetooth	WiFi	HDMI Port	SD/MMC	Ethernet
RaspberryPi Model B (2012)	1 2x USB 2.0	700MHz	512MB	No	No	HDMI	SD	Yes
RaspberryPi Model A+ (2014)	1 1x USB	700MHz	512MB	No	No	HDMI	micro SD	Yes
RaspberryPi (2015)	2 4x USB	900MHz	1GB	No	No	HDMI	micro SD	Yes
Raspberry Pi Zero (2015)	1x Micro USB	1GHz	512MB	4.1	No	Mini HDMI	micro SD	NO
Raspberry Pi ModelB (2016)	3 4x USB	1.2GHz	1GB	4.1	Yes	HDMI	micro SD	Yes
RaspberryPi Zero W (2017)	1xMicro USB	1GHz	512MB	4.2, BLE	Yes	HDMI	micro SD	NO
RaspberryPi3 ModelB+ (2018)	4x USB 2.0	1.4GHz	1GB	4.2, BLE	Yes	HDMI	micro SD	Yes
RaspberryPi ModelA+ (2019)	3 1x USB 2.0	1.4GHz	512MB	5.0, BLE	Yes	2x micro HDMI	micro SD	Yes
RaspberryPi 400 (2021)	400 2x USB 3.0, 2x USB 2.0	1.8GHz	4GB	5.0, BLE	Yes	2x micro HDMI	micro SD	Yes

Κεφάλαιο 3ο: Internet Of Things

3.1 IoT (Internet Of things)

Το IoT σημαίνει «Διαδίκτυο των πραγμάτων». Είναι μια έννοια που αναφέρεται στη διασύνδεση καθημερινών συσκευών και αντικειμένων μέσω του διαδικτύου. Στο IoT, διάφορες συσκευές συνδέονται μεταξύ τους και μπορούν να επικοινωνούν, να συλλέγουν και να ανταλλάσσουν δεδομένα χωρίς ανθρώπινη παρέμβαση. Το IoT έχει τη δυνατότητα να φέρει επανάσταση στον τρόπο που ζούμε και εργαζόμαστε επιτρέποντας ένα ευρύ φάσμα εφαρμογών. Όπως έξυπνα σπίτια, έξυπνες πόλεις, βιομηχανικοί αυτοματισμοί αλλά και αυτόνομη οδήγηση.

Βασικά στοιχεία του IoT περιλαμβάνουν συσκευές/αισθητήρες, συνδεσιμότητα, υποδομή cloud και ανάλυση δεδομένων. Οι συσκευές/αισθητήρες IoT είναι τα «πράγματα» που συνδέονται στο διαδίκτυο και συλλέγουν δεδομένα από το περιβάλλον τους. Οι αισθητήρες αυτοί μπορούν να συνδέονται ασύρματα στο διαδίκτυο παντού. Αυτοί οι αισθητήρες μπορούν να χρησιμοποιήσουν διάφορους τύπους τοπικών συνδέσεων όπως RFID (Radio-Frequency Identification), NFC (Near-Field Communication), Wi-Fi, Bluetooth και Zigbee. Οι αισθητήρες βεβαίως μπορούν να έχουν ευζωνική σύνδεση όπως GSM, GPRS, 3G και LTE(4G). Η πτυχή της συνδεσιμότητας επιτρέπει σε αυτές τις συσκευές να επικοινωνούν μεταξύ τους και με την υποδομή cloud όπου αποθηκεύονται και αναλύονται τα δεδομένα. Η υποδομή cloud παρέχει τους απαραίτητους πόρους υπολογιστών και αποθήκευσης για την επεξεργασία και την ανάλυση των τεράστιων ποσοτήτων δεδομένων που παράγονται από συσκευές IoT. Η ανάλυση δεδομένων είναι η διαδικασία εξαγωγής πληροφοριών και γίνεται με την βοήθεια το συσκευών που είναι στο διαδίκτυο.[3]

Το IoT έχει τη δυνατότητα να μεταμορφώσει τις βιομηχανίες και να βελτιώσει την ποιότητα ζωής των ανθρώπων σε όλο τον κόσμο. Για παράδειγμα, στα έξυπνα σπίτια, οι συσκευές IoT μπορούν να ελέγχουν και να παρακολουθούν τη θερμοκρασία, τον φωτισμό, την ασφάλεια και άλλες πτυχές ενός σπιτιού για να βελτιώσουν την άνεση και την ενεργειακή απόδοση. Στην υγειονομική περίθαλψη, οι συσκευές IoT μπορούν να χρησιμοποιηθούν για την απομακρυσμένη παρακολούθηση των συνθηκών υγείας των ασθενών και την παροχή ειδοποιήσεων σε πραγματικό χρόνο στους παρόχους υγειονομικής περίθαλψης εάν εντοπιστούν ανωμαλίες. Στον βιομηχανικό αυτοματισμό, οι συσκευές IoT μπορούν να βελτιστοποιήσουν τις διαδικασίες παραγωγής, να μειώσουν το χρόνο διακοπής λειτουργίας και να βελτιώσουν τη συνολική απόδοση.

Με μια γενική εικόνα θα μπορούσαμε να χαρακτηρίσουμε το internet of things σαν ένα αξεδιάλυτο μείγμα hardware, software, δεδομένων και υπηρεσιών από συσκευές που συλλέγουν δεδομένα για να βοηθήσουν ήδη υπάρχουσες τεχνολογίες και να διανέμουν τα δεδομένα ανάμεσα σε πλήθος συσκευών.

3.2 SBC Και IOT

Από τεχνική άποψη το διαδίκτυο των πραγμάτων είναι η σύνδεση και η επικοινωνία συσκευών που είναι συνδεδεμένες στο δίκτυο για την επίτευξη ενός στόχου .Οι συσκευές αυτές επικοινωνούν μεταξύ τους με ένα σύστημα ανάδρασης για να επιτευχθεί ο έλεγχος του αποτελέσματος.[4]

Τα SBC μπορούν λόγω του μικρού τους μεγέθους,της χαμηλής κατανάλωσης άλλα και των ισχυρών επεξεργαστών που διαθέτουν να εργαστούν αποτελεσματικά σε εφαρμογές του IOT. Τα SBC έχουν την δυνατότητα να εκτελούν παράλληλες εργασίες και να επεξεργάζονται δεδομένα σε πραγματικό χρόνο .

3.3 .SBC Και Μηχανική όραση

Η μηχανική όραση η αλλιώς όραση του υπολογιστή είναι η απόκτηση η επεξεργασία και η ανάλυση οπτικών δεδομένων για την κατανόηση πραγματικών ερεθισμάτων όπως αυτά γίνονται αντιληπτά από την ανθρώπινη όραση .Μέσω της χρήσης οπτικών αισθητήρων όπως κάμερες και με την βοήθεια του προγραμματισμού ο υπολογιστής είναι σε θέση να αναγνωρίσει και να εκτιμάει τα αντικείμενα γύρω του.

Τα SBC αποτελούν βασικά εργαλεία για εφαρμογές μηχανικής όρασης διότι.

- Η κατανάλωσης τους που δεν ξεπερνά τα 5W κατά μέσο όρο.
- Το μικρό μέγεθος τα κάνει κατάλληλα για κινητές συσκευές.
- Δεν χρειάζονται σύστημα ψύξης πράγμα το οποίο τα καθιστά αθόρυβα.
- Διαθέτουν δυνατούς επεξεργαστές.
- Επεξεργάζονται τα δεδομένα σε πραγματικό χρόνο.

Κεφάλαιο 4ο Κάμερες

4.1 Κάμερες και ιστορική αναδρομή.

Οι κάμερες είναι το βασικό μέσο για την καταγραφή βίντεο εικόνων και οπτικών ερεθισμάτων από τον υπολογιστή. Η κάμερα είναι μια συσκευή με φακό, κλείστρο και αισθητήρα που συλλαμβάνει και καταγράφει οπτικές πληροφορίες. Η πρώτη κάμερα που είναι γνωστή στην ιστορία είναι η camera obscura. Εννοιολογικές περιγραφές του camera obscura μπορούν να βρεθούν σε κινεζικά κείμενα από το 400 π. Χ. και στα γραπτά του Αριστοτέλη, γύρω στο 330 π. Χ. Περίπου το 1000 μ. Χ., η έννοια του camera obscura διατυπώθηκε από τον Άραβα μελετητή Ibn Al-Haytham. Μια κάμερα obscura δεν τραβάει φωτογραφίες, αλλά μάλλον εστιάζει το φως μέσω ενός φακού (τεχνικά μια μικρή τρύπα) και το προβάλλει σε μια οθόνη. Το 1685, ο Γερμανός συγγραφέας Johann Zahn πρόσφερε ένα σχέδιο για αυτό που είναι γνωστό ως reflex κάμερα χειρός. Ωστόσο, κανένας εφευρέτης δεν συνειδητοποίησε την χρησιμότητα της μέχρι που ο Γάλλος Joseph Nicéphore Niépce τα κατάφερε το 1806.

Ο Niépce ανέπτυξε φωτογραφικές εικόνες σε χαρτί επενδυμένο με χλωριούχο ασήμι και μια φωτογραφία που έβγαλε περίπου το 1826 είναι η παλαιότερη σωζόμενη φωτογραφία στο Πανεπιστήμιο του Τέξας στο Όστιν. Τις επόμενες δεκαετίες και με την πάροδο των χρόνων αναπτύχθηκαν πολλά είδη καμερών όπως αναφέρονται παρακάτω.

- **Daguerreotypes:** Ο Louis Daguerre δημιούργησε ένα πιο πρακτικό μοντέλο κάμερας το 1829. Οι εικόνες που τραβήχτηκαν από την κάμερα του Daguerre ήταν γνωστές ως daguerreotypes και η μέθοδος του καθόρισε τις κάμερες στα μέσα του δέκατου ένατου αιώνα. Η διαδικασία περιλαμβάνει την επικάλυψη μιας πλάκας χαλκού με ασήμι, την τοποθέτηση της σε ιώδιο και στη συνέχεια την ανάπτυξη της πάνω από ζεστό υδράργυρο.
- **The mirror camera** (Η κάμερα καθρέφτη): Το πρόβλημα με το σύστημα κάμερας του Daguerre ήταν ότι οι εικόνες ξεθώριαζαν γρήγορα. Αυτό διορθώθηκε από τον Αμερικανό εφευρέτη Alexander S. Wolcott, ο οποίος δημιούργησε αυτό που έγινε γνωστό ως η κάμερα καθρέφτη. Αυτή η κάμερα παρήγαγε μια θετική εικόνα αντί για μια αρνητική με αντίστροφα χρώματα.
- **Instantaneous exposures** (Στιγμιαίες εκθέσεις): Στη συνέχεια, το 1871, ο Richard Leach Maddox εφηύρε μια ξηρή πλάκα ζελατίνης που παρήγαγε στιγμιαίες εκθέσεις - λειτουργώντας ως ένα είδος προδρόμου για τις κάμερες Polaroid του εικοστού αιώνα.
- **Kodak:** Η αναλογική φωτογραφία δεν έφτασε στο ζενίθ της μέχρι που ο Αμερικανός Τζορτζ Ίστμαν πρωτοστάτησε στη χρήση φωτογραφικών μηχανών roll film. Ξεκινώντας με χάρτινο φιλμ αλλά γρήγορα μεταβαίνοντας στο seliloid υλικό με βάση την σελλιζόλη με ελαστικές ιδιότητες, ο Eastman άρχισε να πουλά μια κάμερα box που την ονόμασε Kodak το 1888. Το 1901, αυτές οι αρχικές κινηματογραφικές μηχανές έδωσαν τη θέση τους στις κάμερες Brownie της Kodak, μια φθηνότερη παραλλαγή.
- **35mm film camera** (Κάμερα φιλμ 35 mm): Μεταξύ 1905 και 1913, οι εταιρείες φωτογραφικών μηχανών παρουσίασαν αυτόνομα ρολά φιλμ 35 mm που μπορούσαν να εισαχθούν και να αφαιρεθούν από την κάμερα του ίδιου του χρήστη. Ο Όσκαρ Μπάρνακ, ένας Γερμανός εφευρέτης

Κεφάλαιο 4ο

και φωτογράφος, πιστώνεται γενικά για την εμφάνιση των καμερών με φιλμ 35 χιλιοστών, ξεκινώντας από τη Leica, την οποία δημιούργησε για την εταιρεία Leitz. Ωστόσο, η Kodak θα γινόταν γρήγορα ο κορυφαίος πάροχος φωτογραφικών φιλμ για κάμερες 35 χιλιοστών στον κόσμο, με άλλες εταιρείες όπως η Fujifilm να παρέχουν αργότερα ισχυρό ανταγωνισμό.

- **Twin-reflex camera TLR** (Κάμερα διπλής αντανάκλαστικής): Η κάμερα γνώρισε σημαντική πρόοδο με την εφεύρεση των καμερών αντανάκλαστικού φακού, οι οποίες εισήγαγαν χαρακτηριστικά όπως σκόπευτρα, μεταβλητές ταχύτητες κλείστρου και αποσπώμενους φακούς. Οι παλαιότερες από αυτές τις κάμερες ήταν κάμερες διπλού φακού reflex (ή TLR για συντομία). Οι κάμερες TLR αντικαταστάθηκαν γρήγορα από κάμερες αντανάκλαστικού μονού φακού (ή SLR).
- **Digital SLR camera** (Ψηφιακή φωτογραφική μηχανή SLR): Η πρώτη φωτογραφική μηχανή TLR κατασκευάστηκε το 1999 και μετά από λίγα μόνο χρόνια τεχνολογικών βελτιώσεων, αντικατέστησε σε μεγάλο βαθμό τις αντανάκλαστικές κάμερες ενός φακού. Μια ψηφιακή αντανάκλαστική κάμερα ενός φακού (TLR ή ψηφιακή SLR) είναι ένας τύπος φωτογραφικής μηχανής που προσφέρει ποιότητα εικόνας υψηλής ποιότητας και χρησιμοποιείται ευρέως από ερασιτέχνες και επαγγελματίες. Μια κάμερα TLR σας επιτρέπει να βλέπετε την ακριβή εικόνα που τραβάτε απευθείας μέσω του σεισμσκοπίου, επιτρέποντάς σας να εστιάζετε και να καταγράφετε καλύτερα τις σκηνές σας.
- **Mirror-less Camera** (Κάμερα χωρίς καθρέφτη): Το 2004, η Epson κυκλοφόρησε την πρώτη κάμερα χωρίς καθρέφτη, έναν τύπο κάμερας που λειτουργεί χωρίς αντανάκλαστικό καθρέφτη. Το φως περνάει μέσω του φακού απευθείας στον ψηφιακό αισθητήρα, ο οποίος στη συνέχεια εμφανίζει την εικόνα σας στην οθόνη LCD της κάμερας, επιτρέποντάς σας να προσαρμόσετε τις ρυθμίσεις και να κάνετε προ επισκόπηση της εικόνας πριν από τη λήψη της.

Στην ψηφιακή φωτογραφία, φωτόνια φως περνούν μέσα από τον φακό και πάνω σε έναν ψηφιακό οπτικό αισθητήρα. Αυτοί οι αισθητήρες αποτελούνται από εκατομμύρια φωτοδιόδους, οι οποίες, συνδυαστικά, δημιουργούν μια φωτογραφική εικόνα.



4.2 Κάμερες ασφαλείας.

Με την πάροδο των ετών καθώς και με τον ερχομό των δύο παγκοσμίων πολέμων η ανάγκη για ασφάλεια και επιτήρηση έγινε όλο και μεγαλύτερη αυτό οδήγησε στην τεχνολογική ανάπτυξη καμερών ασφαλείας. Στην αρχή οι κάμερες ασφαλείας λειτουργούσαν κυρίως στον στρατιωτικό τομέα και έπειτα επεκτάθηκαν και στη βιομηχανία τις επαγγελματικές στέγες καθώς και την οικιακή χρήση τους σε κατοικίες.

- 1920: Ένας Ρώσος εφευρέτης ο Leon Theremin δημιούργησε μια συσκευή ραδιοκυμάτων για να κρυφακούει τις συνομιλίες η εφεύρεση του αυτή άνοιξε τις πύλες για την ανάπτυξη των τεχνολογιών ασφαλείας.
- 1942: Το 1942 Γερμανοί στρατιωτικοί ερευνητές δημιουργούν για πρώτη φορά ένα κλειστό κύκλωμα παρακολούθησης CCTV. Αυτό το σύστημα παρακολουθούσε στρατιωτικές επιχειρήσεις όπως εκτοξεύσεις πυραύλων αλλά και τον έλεγχο των τμημάτων των βάσεων και υποδομών κατά την διάρκεια της νύχτα.
- 1949: Μετά τον Β Παγκόσμιο πόλεμο οι Αμερικανοί άρχισαν να προσαρμόζουν τα συστήματα CCTV σε βιομηχανικούς χώρους για την αποτροπή κακόβουλων ενεργειών αυτή η ανάπτυξη οδήγησε και στην ανάπτυξη του συστήματος αυτού και στις κατοικίες.
- 1966: Η Marie Van Britta Brown δημιουργεί το πρώτο οικιακό σύστημα παρακολούθησης τηλεόρασης CCTV. Το σύστημα αυτό διέθετε πολλές κάμερες σε διαφορετικά ύψη οθόνη προβολής αμφίδρομη επικοινωνία σύστημα απομακρυσμένου ξεκλειδώματος άλλα και κουμπί συναγερμού.
- 1968: Μετά την εφεύρεση της Marie η αστυνομία εφάρμοσε και προσάρμοσε το σύστημα αυτό για την καλύτερη επίβλεψη στους δρόμους.
- 1970: Η ανάπτυξη της τεχνολογίας CCD βελτίωσε την ποιότητα εικόνας σε συνθήκες χαμηλού φωτισμού Τα VCR διευκόλυναν την καταγραφή των δεδομένων.
- 1990: Η τεχνολογία των VCR αντικαταστάει με τα DVR και έχουμε και την δημιουργία της πρώτης IP Camera ασφαλείας από την εταιρία AXIS Communications.
- 2000-σήμερα: Η τεχνολογική εξέλιξη έχει φέρει και άλλες κατηγορίες καμερών όπως.
- Web cameras: Οι web κάμερες χρησιμοποιούνται κυρίως για βιντεοκλήσεις και καταγραφή βίντεο μέσω υπολογιστή. Είναι μικρές, προσιτές και εύχρηστες, καθιστώντας τις ιδανικές για προσωπική χρήση και απλές ανάγκες βιντεοσκόπησης.
- Κάμερες Δράσης (Action Cameras): Οι κάμερες δράσης είναι σχεδιασμένες για σκληρές συνθήκες χρήσης και καταγραφή ακραίων δραστηριοτήτων, όπως καταδύσεις, αθλήματα δράσης κ.ά. Είναι μικρές, αδιάβροχες και ανθεκτικές στους κραδασμούς, με την πιο γνωστή μάρκα να είναι η GoPro.
- Κάμερες Θερμικής Απεικόνισης: Αυτές οι κάμερες ανιχνεύουν την υπέρυθη ακτινοβολία (θερμότητα) και χρησιμοποιούνται κυρίως σε περιπτώσεις όπου η οπτική απεικόνιση δεν είναι δυνατή, όπως σε νυχτερινή παρακολούθηση ή για τη μέτρηση θερμοκρασιών.

4.3 Κατηγορίες Καμερών ασφαλείας Σήμερα.

- Κάμερες IP (Internet Protocol): Χρησιμοποιούν το δίκτυο για τη μετάδοση εικόνων και βίντεο, επιτρέποντας την απομακρυσμένη παρακολούθηση μέσω διαδικτύου

Κεφάλαιο 4ο

- **Κάμερες CCTV (Closed Circuit Television):** Συμβατικές κάμερες που χρησιμοποιούνται σε κλειστά κυκλώματα τηλεόρασης. Συχνά συνδέονται με DVR (Digital Video Recorder) για την καταγραφή του υλικού.
- **Ασύρματες κάμερες:** Λειτουργούν μέσω Wi-Fi, εξαλείφοντας την ανάγκη για καλώδια, καθιστώντας την εγκατάσταση πιο εύκολη.
- **PTZ (Pan-Tilt-Zoom):** Κάμερες που μπορούν να περιστρέφονται οριζόντια, κατακόρυφα και να ζουμάρουν, επιτρέποντας την κάλυψη ευρύτερων περιοχών.

4.4 Εφαρμογές καμερών ασφαλείας σήμερα.

AI και Αναγνώριση Προσώπου

Οι κάμερες ασφαλείας χρησιμοποιούν όλο και περισσότερο την τεχνητή νοημοσύνη (AI) για την αναγνώριση προσώπων και αντικειμένων σε πραγματικό χρόνο. Για παράδειγμα, σε περιοχές υψηλής κυκλοφορίας, οι κάμερες μπορούν να αναγνωρίσουν άμεσα άτομα από μια βάση δεδομένων, βελτιώνοντας την ασφάλεια και την παρακολούθηση.

Cloud Storage και Remote Access

Οι κάμερες θα συνεχίσουν να μετατοπίζονται προς την αποθήκευση cloud, επιτρέποντας στους χρήστες να έχουν πρόσβαση στο υλικό από απόσταση. Αυτή η τεχνολογία διευκολύνει τη διαχείριση και την ανάκτηση πλάνα ασφαλείας από οπουδήποτε, ανά πάσα στιγμή.

Ενοποίηση με Smart Home Systems

Οι μελλοντικές κάμερες θα ενσωματωθούν απρόσκοπτα με έξυπνες οικιακές συσκευές, όπως κάμερες με κουδούνια πόρτας, φώτα και θερμοστάτες. Για παράδειγμα, μια κάμερα ασφαλείας θα μπορούσε να ενεργοποιήσει τα μπροστινά φώτα για να ανάψουν αυτόματα όταν ανιχνευτεί κίνηση.

Βελτιωμένη νυχτερινή όραση και απόδοση σε χαμηλό φωτισμό

Οι εξελίξεις στην τεχνολογία αισθητήρων ενισχύουν τις δυνατότητες νυχτερινής όρασης, επιτρέποντας καθαρότερες εικόνες σε συνθήκες χαμηλού φωτισμού. Αυτή η βελτίωση είναι ιδιαίτερα χρήσιμη για τη νυχτερινή επιτήρηση σε οικιακούς και εμπορικούς χώρους.

Παρακολούθηση μέσω Drone

Παρακολούθηση γρηγορότερα σε πραγματικό χρόνο εκτάσεις που υπάρχει κίνδυνος πυρκαγιάς κατά την καλοκαιρινή περίοδο κ.α.

Κεφάλαιο 5ο:5 Γλώσσες Προγραμματισμού(Programming languages).

5.1 Γλώσσες Προγραμματισμού(Programming language)

Μια γλώσσα προγραμματισμού είναι μια τεχνητή γλώσσα που έχει σχεδιαστεί για να μεταδίδει οδηγίες σε έναν υπολογιστή. Οι γλώσσες προγραμματισμού μπορούν να χρησιμοποιηθούν για τη δημιουργία προγραμμάτων που ελέγχουν τη συμπεριφορά ενός υπολογιστή και για την έκφραση αλγορίθμων. Υπάρχουν πολλές διαφορετικές γλώσσες προγραμματισμού. Οι περισσότερες γλώσσες προγραμματισμού περιγράφουν τον υπολογισμό με ένα στυλ επιτακτικής, δηλαδή ως μια ακολουθία εντολών, αν και ορισμένες γλώσσες, όπως αυτές που υποστηρίζουν λειτουργικό προγραμματισμό ή λογικό προγραμματισμό, χρησιμοποιούν εναλλακτικές μορφές περιγραφής. Οι γλώσσες προγραμματισμού χρησιμοποιούνται για να διευκολύνουν την οργάνωση και διαχείριση πληροφοριών, αλλά και για την ακριβή διατύπωση αλγορίθμων. Αποτελούν βασικό πυλώνα επικοινωνίας του ανθρώπου με έναν υπολογιστή διότι δημιουργούν έναν δίαυλο επικοινωνίας μεταξύ γλώσσα ανθρώπου και μηχανής.[1]

Ιστορικά μιλώντας Στη δεκαετία του 1940, δημιουργήθηκαν οι πρώτοι αναγνωρίσιμοι σύγχρονοι ηλεκτροκίνητοι υπολογιστές(βλ κεφ1). Η περιορισμένη ταχύτητα και η περιορισμένη χωρητικότητα μνήμης ανάγκασαν τους προγραμματιστές να γράφουν προγράμματα γλώσσας συναρμολόγησης με το χέρι assembly. Τελικά έγινε αντιληπτό ότι ο προγραμματισμός σε γλώσσα assembly απαιτούσε μεγάλο κόπο και πολύ χρόνο για την υλοποίηση ενός έργου. Μια πρώτη πρόταση για μια γλώσσα προγραμματισμού υψηλό επίπεδο ήταν η Plankalkül, που αναπτύχθηκε από τον Konrad Zuse για τον υπολογιστή του Z1 μεταξύ 1942 και 1945 αλλά δεν εφαρμόστηκε εκείνη την εποχή. Οι πρώτες λειτουργικές γλώσσες προγραμματισμού που σχεδιάστηκαν για να επικοινωνήσουν οδηγίες σε έναν υπολογιστή γράφτηκαν στις αρχές της δεκαετίας του 1950. Ο Σύντομος Κώδικας του John Mauchly, που προτάθηκε το 1949, μία από τις πρώτες γλώσσες υψηλού επιπέδου που αναπτύχθηκαν ποτέ για ηλεκτρονικό υπολογιστή.[5]

Σε αντίθεση με τον κώδικα μηχανής, οι δηλώσεις σύντομου κώδικα αντιπροσώπευαν μαθηματικές εκφράσεις σε κατανοητή μορφή [5]

Το 1954 εφευρέθηκε η FORTRAN της εταιρείας IBM από μια ομάδα με επικεφαλής τον John Backus. Ήταν η πρώτη ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού γενικού σκοπού υψηλού επιπέδου που είχε λειτουργική υλοποίηση, σε αντίθεση με απλώς ένα σχέδιο σε χαρτί. MATIC. Η Flow-Matic είχε μεγάλη επιρροή στο σχεδιασμό της COBOL, Τις δεκαετίες του 1960 και του 1970 δομήθηκαν γλώσσες όπως η Pascal και η BASIC.Μετα την επιτυχία Του GUI και της δημιουργίας προσωπικών υπολογιστών οι γλώσσες προγραμματισμού έγιναν ένα απαραίτητο κομμάτι της τεχνολογικής εξέλιξης της κοινωνίας με αποτέλεσμα να δημιουργηθούν και άλλες γλώσσες προγραμματισμού . Κάποιες από αυτές αναφέρονται παρακάτω.

- **1970/Pascal:** Πήρε το όνομά του από τον Γάλλο μαθηματικό Blaise Pascal, ο Niklaus Wirth ανέπτυξε τη γλώσσα προγραμματισμού προς τιμήν του. Αναπτύχθηκε ως εργαλείο εκμάθησης για

Κεφάλαιο 5ο

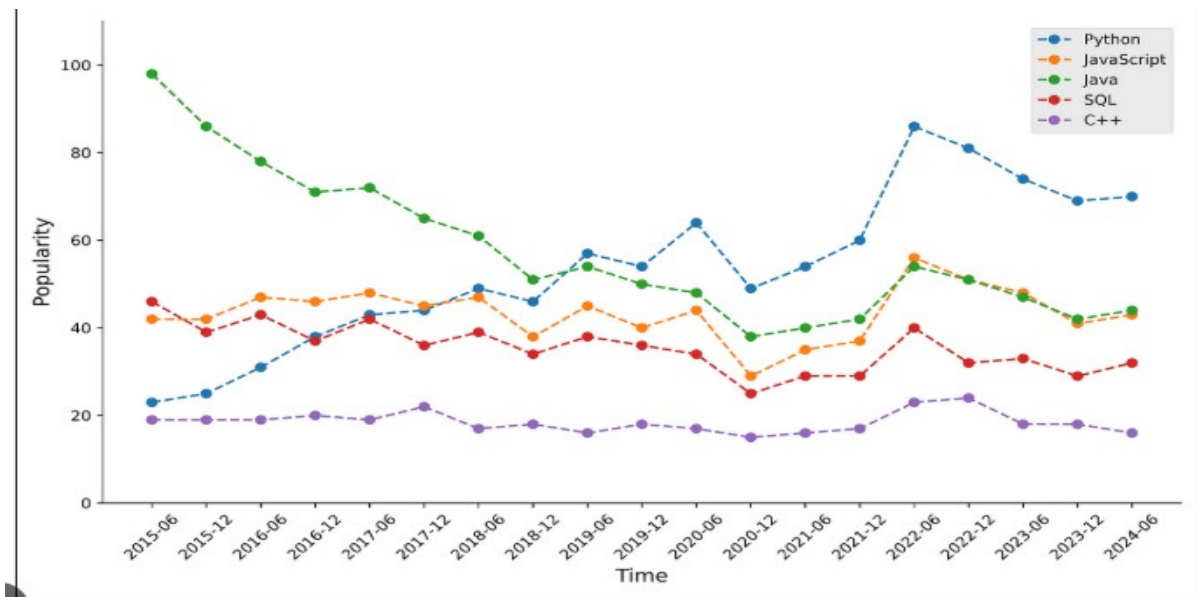
προγραμματισμό υπολογιστών που σήμαινε ότι ήταν εύκολο στην εκμάθηση. Ευνοήθηκε από την Apple στις πρώτες μέρες της εταιρείας, λόγω της ευκολίας χρήσης και της ισχύος του.

- **1972/Smalltalk:** Αναπτύχθηκε στο Ερευνητικό Κέντρο Xerox Palo Alto από τους Alan Kay, Adele Goldberg και Dan Ingalls, το Smalltalk επέτρεψε στους προγραμματιστές υπολογιστών να τροποποιούν τον κώδικα εν κινήσει. Εισήγαγε μια ποικιλία πτυχών γλωσσών προγραμματισμού που είναι ορατές γλώσσες του σήμερα όπως η Python, η Java και η Ruby. Εταιρείες όπως η Leafly, η Logitech και η CrowdStrike δηλώνουν ότι χρησιμοποιούν το Smalltalk στις στοίβες τεχνολογίας τους.
- **1972/C:** Αναπτύχθηκε από τον Dennis Ritchie στα Bell Telephone Laboratories για χρήση με το λειτουργικό σύστημα Unix. Ονομάστηκε C επειδή βασίστηκε σε μια προηγούμενη γλώσσα που ονομαζόταν «B». Πολλές από τις τρέχουσες κορυφαίες γλώσσες είναι παράγωγα του C, συμπεριλαμβανομένων: C#, Java, JavaScript, Perl, PHP και Python. Επίσης, χρησιμοποιείται/ακόμη χρησιμοποιείται από τεράστιες εταιρείες όπως η Google, το Facebook και η Apple.
- **1972/SQL (SEQUEL εκείνη την εποχή):** Η SQL αναπτύχθηκε για πρώτη φορά από τους ερευνητές της IBM Raymond Boyce και Donald Chamberlain. Το SEQUEL (όπως αναφερόταν τότε), χρησιμοποιείται για την προβολή και την αλλαγή πληροφοριών που είναι αποθηκευμένες σε βάσεις δεδομένων. Σήμερα η γλώσσα είναι ένα αρκτικόλεξο – SQL, το οποίο σημαίνει δομημένη γλώσσα ερωτημάτων. Υπάρχει μια πληθώρα εταιρειών που χρησιμοποιούν SQL και μερικές από αυτές περιλαμβάνουν τη Microsoft και την Accenture.
- **1981/Ada:** Το Ada σχεδιάστηκε αρχικά από μια ομάδα με επικεφαλής τον Jean Ichbiah της CUU Honeywell Bull με σύμβαση με το Υπουργείο Άμυνας των Ηνωμένων Πολιτειών. Πήρε το όνομά της από τη μαθηματικό Ada Lovelace των μέσων του 19ου αιώνα, η Ada είναι μια δομημένη, στατικά δακτυλογραφημένη, επιτακτική, ευρέως φάσματος και αντικειμενοστραφή γλώσσα προγραμματισμού υψηλού επιπέδου. Η Ada επεκτάθηκε από άλλες δημοφιλείς γλώσσες προγραμματισμού εκείνη την εποχή, όπως η Pascal. Το Ada χρησιμοποιείται για συστήματα διαχείρισης εναέριας κυκλοφορίας σε χώρες όπως η Αυστραλία, το Βέλγιο και η Γερμανία, καθώς και μια σειρά από άλλα έργα μεταφορών και διαστήματος.
- **1983/C++:** Ο Bjarne Stroustrup τροποποίησε τη γλώσσα C στα Bell Labs, η C++ είναι μια επέκταση της C με βελτιώσεις όπως κλάσεις, εικονικές συναρτήσεις και πρότυπα. Περιλαμβάνεται στις 10 κορυφαίες γλώσσες προγραμματισμού από το 1986 και έλαβε το καθεστώς Hall of Fame το 2003. Η C++ χρησιμοποιείται σε MS Office, Adobe Photoshop, μηχανές παιχνιδιών και άλλο λογισμικό υψηλής απόδοσης.
- **1991/Python:** Πήρε το όνομά του από τον βρετανικό θίασο κωμωδίας «Monty Python», ο Python αναπτύχθηκε από τον Guido Van Rossum. Είναι μια γλώσσα προγραμματισμού γενικής χρήσης, υψηλού επιπέδου που δημιουργήθηκε για να υποστηρίζει μια ποικιλία από στυλ προγραμματισμού και να είναι διασκεδαστική στη χρήση (ορισμένος από τα μαθήματα, τα δείγματα και τις οδηγίες συχνά περιέχουν αναφορές Monty Python). Η Python είναι, μέχρι σήμερα, μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού στον κόσμο που χρησιμοποιείται από εταιρείες όπως η Google, το yahoo και το Spotify.[5]
- **1991/Visual Basic:** Αναπτύχθηκε από τη Microsoft, η Visual Basic επιτρέπει στους προγραμματιστές να χρησιμοποιούν ένα στυλ μεταφοράς και απόθεσης για την επιλογή και την αλλαγή προεπιλεγμένων τμημάτων κώδικα μέσω μιας γραφικής διεπαφής χρήστη (GUI). Η γλώσσα

Γλώσσες Προγραμματισμού

δεν χρησιμοποιείται υπερβολικά αυτές τις μέρες, ωστόσο η Microsoft έχει χρησιμοποιήσει τμήματα της Visual Basic σε ορισμένες από τις εφαρμογές της όπως το Word, το Excel και την Access.

- **1993/Ruby:** Δημιουργήθηκε από τον Yukihiro Matsumoto, η Ruby είναι μια ερμηνευμένη γλώσσα προγραμματισμού υψηλού επιπέδου. Μια γλώσσα διδασκαλίας που επηρεάστηκε μεταξύ άλλων από τους Perl, Ada, Lisp και Smalltalk. Οι κύριες χρήσεις του Ruby είναι για την ανάπτυξη διαδικτυακών εφαρμογών και το Ruby on Rails. Το Twitter, το Hulu και το Groupon είναι μερικά γνωστά παραδείγματα εταιρειών που
- **1995/Java:** Η Java είναι μια γλώσσα γενικής χρήσης, υψηλού επιπέδου που δημιουργήθηκε από τον James Gosling για ένα διαδραστικό τηλεοπτικό έργο. Έχει λειτουργικότητα πολλαπλών πλατφορμών και είναι σταθερά μεταξύ των κορυφαίων των πιο δημοφιλών γλωσσών προγραμματισμού στον κόσμο.
- **1995/PHP:** Παλαιότερα γνωστή ως «Personal Home Page» που τώρα σημαίνει «Hypertext Preprocessor», η PHP αναπτύχθηκε από τον Rasmus Lerdorf. Οι κύριες χρήσεις του περιλαμβάνουν τη δημιουργία και τη συντήρηση δυναμικών ιστοσελίδων, καθώς και ανάπτυξη από την πλευρά του διακομιστή. Μερικές από τις μεγαλύτερες εταιρείες από όλο τον κόσμο χρησιμοποιούν PHP, συμπεριλαμβανομένων των Facebook, Wikipedia, wordpress
- **1995/JavaScript:** Η JavaScript δημιουργήθηκε από τον Brendan Eich, αυτή η γλώσσα χρησιμοποιείται κυρίως για δυναμική ανάπτυξη ιστού, έγγραφα PDF, προγράμματα περιήγησης ιστού και γραφικά στοιχεία επιφάνειας εργασίας. Σχεδόν κάθε σημαντικός ιστότοπος χρησιμοποιεί JavaScript. Το Gmail, το Adobe Photoshop και ο Mozilla Firefox περιλαμβάνουν μερικά γνωστά παραδείγματα.[5][6].



Κεφάλαιο 6ο Υλοποίηση Συστήματος – Βιβλιοθήκες και Εργαλεία

6 Ανάλυση Υλοποίησης Συστήματος.

Η εφαρμογή που θα ενσωματωθεί στο Raspberry pi θα είναι μια online εφαρμογή μεταφοράς δεδομένων εικόνας σε πραγματικό χρόνο μέσω της [Java](#).

Η επιλογή του pi ήταν χαμηλή κατανάλωση η γρήγορη επεξεργασία δεδομένων καθώς και η δυνατότητα ασύρματης σύνδεσης με wifi. της συσκευής Το πρόγραμμα το οποίο έγινε στο inteliJ χωρίζεται σε δύο μέρη το frontend και το backend.

Το frontend είναι το μέρος του προγράμματος το οποίο επικοινωνεί με το χρήστη δίνοντας του ερεθίσματα και εμφανίζει τα δεδομένα της κάμερας στην σελίδα. Η HTML η CSS οπως και η java script αποτελούν τα στοιχεία του frontend προγράμματος μας όπου:

HTML Η HTML είναι υπεύθυνη για την εμφάνιση της εικόνας και το πληροφοριών στο site καθώς και τον τίτλο της σελίδας

CSS Μορφοποιεί και ορίζει το στυλ της σελίδας την θέση της εικόνας την εμφάνιση του κειμένου πάνω στην εικόνα καθώς και τους χρωματισμούς

Javascript Σαν κύρια λειτουργία έχει την σύνδεση με τον websocketserver και την λήψη δεδομένων σε πραγματικό χρόνο στο(ws://<hostname>:<port>/socket).

Επιπλέον είναι αυτή που χρησιμοποιείται για την εισαγωγή του κωδικού εισόδου αλλά να ενημερώνει την ιστοσελίδα για την νέα εικόνα (σε μορφή Base64 String) και να αναπαράγει τον ήχο.

Αντιστοίχως σε επίπεδο backend το οποίο είναι και το τμήμα του προγράμματος το οποίο δεν είναι άμεσα εμφανή στον χρήστη έχουμε εργαλεία όπως το spark framework για την δημιουργία του server την βιβλιοθήκη openCV για την διαχείριση της κάμερας.

Spark είναι το framework το οποίο αναλύει τα δεδομένα και διαχειρίζεται τα δεδομένα και ουσιαστικά δημιουργεί και τον web socket server.

Ανοίγει την πόρτα 8080 και αποστέλλει τα δεδομένα της κάμερας

websocket είναι ένα πρωτόκολλο επικοινωνίας το οποίο διαχειρίζεται τις συνδέσεις και αποσυνδέσεις των clients και κάθε 300ms στέλνει εικόνα και ανιχνεύτηκε κίνηση.

Στην συνέχεια η βιβλιοθήκη openCV με κάποια εργαλεία όπως το μοντέλο Gaussian blur .Αρχικά κάνει δύο λήψεις από την κάμερα και επεξεργάζεται τα πλάνα για ανίχνευση κίνησης .Με την βοήθεια της τεχνικής Gaussian blur και με την χρήση διαφόρων effect όπως μετατροπή της εικόνας σε γκρι () καθώς και την μείωση του θορύβου της εικόνας για ευκολότερη επεξεργασία.

Τέλος με την βοήθεια της υπηρεσίας emailjs μπορούμε να αποστέλλουμε το email με την φωτογραφία όταν γίνει η ανίχνευση κίνησης

- **Frontend:** Παρουσιάζει την εικόνα από την κάμερα στον χρήστη και αναπαράγει τον ήχο του συναγερμού αν ανιχνευτεί κίνηση.
- **Backend:** Διαχειρίζεται την κάμερα, την ανίχνευση κίνησης και την αποστολή εικόνας μέσω

WebSocket στον client.

- **Επικοινωνία:** Ο frontend και backend επικοινωνούν μέσω WebSocket για να μεταφέρουν εικόνες και ειδοποιήσεις (κίνηση/ήχος).

6.1 JAVA

Η Java αποτελεί μια από τις πιο διαδεδομένες γλώσσες προγραμματισμού παγκοσμίως. Η αρχή της Java ξεκίνησε το 1991 από την εταιρεία Sun Microsystems και επικεφαλής τον James Gosling η αρχική της ονομασία ήταν oak και είχε σχεδιαστεί για υπολογιστές.

Το 1995 μετονομάστηκε σε java. Και ο βασικός της στόχος ήταν να μπορεί να εργάζεται σε οποιαδήποτε πλατφόρμα και λειτουργικό σύστημα.

Τέλος το 2009 αγοράστηκε από την εταιρεία Oracle corporation η οποία ανέλαβε και την ανάπτυξη της γλώσσας αυτής.

Η Java αποτελεί μια γλώσσα υψηλού επιπέδου γλώσσα προγραμματισμού διότι επιτρέπει τους προγραμματιστές να γράφουν κώδικα σε συσχέτιση με την ανθρώπινη γλώσσα και όχι σε γλώσσα μηχανής. Επιπλέον είναι κυρίως μια αντικειμενοστραφή γλώσσα προγραμματισμού (OOP) εστιάζει στη δημιουργία αντικειμένων που ενσωματώνουν δεδομένα και συμπεριφορά.

Τέλος είναι μια δημοφιλής γλώσσα προγραμματισμού που χρησιμοποιείται ευρέως για την ανάπτυξη ενός ευρέος φάσματος εφαρμογών, από λογισμικό υπολογιστή (computer software) έως εφαρμογές ιστού (web applications), εφαρμογές για κινητά (mobile apps) και εταιρικά συστήματα.

6.1.1: Πλεονεκτήματα μειονεκτήματα Της JAVA

Πλεονεκτήματα της Java.

- Ένας από τους μεγαλύτερους λόγους για τους οποίους η Java είναι τόσο δημοφιλής είναι η ανεξαρτησία της πλατφόρμας. Τα προγράμματα μπορούν να εκτελεστούν σε διάφορους τύπους υπολογιστών, εφόσον διαθέτουν το Java Runtime Environment (JRE). Οι περισσότεροι τύποι υπολογιστών είναι συμβατοί με ένα JRE, συμπεριλαμβανομένων υπολογιστών που λειτουργούν με Windows, Macintosh, Unix, Linux καθώς και κινητά τηλέφωνα.
- Προσφέρει υψηλή ασφάλεια λόγω ότι ο κώδικας εκτελείται μέσω της JVM μειώνοντας έτσι τα κακόβουλα λογισμικά
- Δεδομένου ότι υπάρχει εδώ και πολύ καιρό, ορισμένοι από τους μεγαλύτερους οργανισμούς

Υλοποίηση Συστήματος - Βιβλιοθήκες Εργαλεία

στον κόσμο έχουν κατασκευαστεί χρησιμοποιώντας την Java. Για παράδειγμα, πολλές τράπεζες, έμποροι λιανικής, ασφαλιστικές εταιρείες, επιχειρήσεις κοινής ωφέλειας και κατασκευαστές .

- Η σύνταξη Java είναι παρόμοια με τη C και τη C++, διευκολύνοντας τους προγραμματιστές που είναι εξοικειωμένοι με αυτές τις γλώσσες να μάθουν Java. Ο κώδικας είναι οργανωμένος σε κλάσεις, οι οποίες περιέχουν πεδία (μεταβλητές) και μεθόδους (συναρτήσεις).
- Η Java παρέχει μια ολοκληρωμένη τυπική βιβλιοθήκη, γνωστή ως Java Class Library (JCL), η οποία προσφέρει ένα πλούσιο σύνολο προκατασκευασμένων κλάσεων και μεθόδων για κοινές εργασίες προγραμματισμού. Το JCL καλύπτει τομείς όπως η δικτύωση, η είσοδος/εξαγωγή, οι δομές δεδομένων και άλλα.
- Διαθέτει επίσης το Java Development Kit (JDK) το οποίο είναι ένα κιτ ανάπτυξης λογισμικού που περιλαμβάνει εργαλεία για την ανάπτυξη, τον εντοπισμό σφαλμάτων και την εκτέλεση εφαρμογών.
- Παρέχει μια μεγάλη γκάμα πλαισίων και βιβλιοθηκών που ενισχύουν τις δυνατότητές της για συγκεκριμένους σκοπούς. Μερικά δημοφιλή πλαίσια Java περιλαμβάνουν τα Spring, Hibernate, JavaFX και Apache Struts.
- Τέλος στο θέμα της υποστήριξης η Java έχει μια μεγάλη και ενεργή κοινότητα προγραμματιστών που παρέχει υποστήριξη, πόρους και βιβλιοθήκες ανοιχτού κώδικα. Ακόμη, ο κώδικας που είναι γραμμένος 15 χρόνια πριν, θα μπορεί να εκτελείται στα πιο ενημερωμένα JVM (java Virtual Machine) και θα αποκτήσει το πλεονέκτημα ταχύτητας .

Μειονεκτήματα της Java.

- Το Πρόγραμμα ενδέχεται να καθυστερήσει σε σχέση με άλλες γλώσσες
- Ο κώδικας πολλές φορές καταναλώνει μεγαλύτερη ποσότητα μνήμης λόγω της χρήσης της JVM.
- Η σύνταξη της μπορεί να είναι πιο σύνθετη από άλλες γλώσσες υψηλού επιπέδου όπως (Python).

6.1.2. JAVA και εφαρμογές .

Λόγο της ευελιξίας της ανεξαρτησίας της πλατφόρμας και την ισχυρή της σύσταση η Java είναι μια δημοφιλής επιλογή για την κατασκευή ενός ευρέος φάσματος εφαρμογών όπως.

- **Ανάπτυξη εφαρμογών Android.** Οι περισσότερες εφαρμογές Android είναι γραμμένες σε Java με την χρήση του Google Android API. 25
- **Ανάπτυξη επιστημονικών εφαρμογών.** Η Java είναι συχνά η προεπιλεγμένη γλώσσα για επιστημονικές εφαρμογές. Είναι ασφαλής, φορητή και διαθέτει καλύτερα εργαλεία υψηλού επιπέδου από οποιαδήποτε άλλη γλώσσα.
- **Ανάπτυξη web εφαρμογών.** Υπάρχουν πολλές και σημαντικές web εφαρμογές στην καθημερινότητα μας. Ένα σημαντικό παράδειγμα είναι το Gmail της Google.
- **Ανάπτυξη εργαλείων λογισμικού.** Πολλά λογισμικά και εργαλεία ανάπτυξης λογισμικού είναι γραμμένα σε Java. Μερικά από αυτά, IntelliJ IDEA, Eclipse, NetBeans IDE.
- **Ανάπτυξη Ιστού:** Η Java χρησιμοποιείται ευρέως για τη δημιουργία διαδικτυακών εφαρμογών από την πλευρά του διακομιστή. Προσφέρει πλαίσια όπως το Spring και το JavaServer Faces (JSF) για την ανάπτυξη ισχυρών και επεκτάσιμων εφαρμογών ιστού.
- **Εφαρμογές επιφάνειας εργασίας:** Η Java μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών επιτραπέζιου υπολογιστή χρησιμοποιώντας πλαίσια(Frameworks) όπως το JavaFX και το Swing. Αυτά τα πλαίσια παρέχουν εργαλεία και βιβλιοθήκες για τη δημιουργία εμπλουτισμένων γραφικών διεπαφών χρήστη (GUI) και συμβατότητας μεταξύ πλατφορμών.
- **Ανάπτυξη και Επεξεργασία ανάλυσης δεδομένων:** Παρέχει βιβλιοθήκες όπως το Apache Commons Math και το Jama για την εκτέλεση πολύπλοκων επιστημονικών και αριθμητικών υπολογισμών. Καθώς διαθέτει και τα Apache Hadoop, Apache Spark (πλαίσια) frameworks για την κατανεμημένη επεξεργασία μεγάλου συνόλου δεδομένων.
- **Ανάπτυξη εφαρμογών Internet of Things (IoT):** Η Java χρησιμοποιείται για την ανάπτυξη εφαρμογών στον τομέα IoT. Παρέχει βιβλιοθήκες όπως το Eclipse IoT, το οποίο προσφέρει πλαίσια για τη δημιουργία πυλών IoT, υπολογιστών αιχμής και συστημάτων διαχείρισης αισθητήρων.

- **Ανάπτυξη παιχνιδιών:** Η Java διαθέτει βιβλιοθήκες όπως το LibGDX και το jMonkeyEngine που έχουν σχεδιαστεί ειδικά για την ανάπτυξη παιχνιδιών. Αυτές οι βιβλιοθήκες παρέχουν εργαλεία για το χειρισμό γραφικών, φυσικής, ήχου και εισαγωγής σε παιχνίδια που βασίζονται σε Java.

6.1.3 Χρήση της Java στο σύστημα ασφαλείας.

Η Java χρησιμοποιήθηκε στο σύστημα ασφαλείας σε επίπεδο backend.

Αποτελεί την αρχή και το τέλος προγράμματος μας .Επιπλέον εντάσσει τις βιβλιοθήκη opencv για να μπορέσει να γίνει δυνατή η χρήση των δυνατοτήτων που αυτή προσφέρει στην ανάλυση πληροφοριών μέσω της κάμερας .Επιπλέον προσθέτει μέσω του spark framework την χρήση του websocket server.

Οι βασικότερες εντολές java που χρησιμοποιήθηκαν είναι.

<pre>Import org.bytedeco.javacpp.Loader; import org.bytedeco.opencv.opencv_java; import spark.Spark;</pre>	Εισαγωγή εργαλείων όπως opencv spark framework
<pre>Loader.load(opencv_java.class);</pre>	Ανοίγει την βιβλιοθήκη opencv μέσω της java
<pre>Camera.openCamera();</pre>	Μέσω της βιβλιοθήκης opencv ανοίγει την κάμερα

Πίνακας 2: OpenCV Εντολές (σύστημα ασφαλείας)

6.2 .Βιβλιοθήκη OpenCV

Το OpenCV (Open Source Computer Vision Library) είναι μια βιβλιοθήκη λογισμικού ανοιχτού κώδικα όρασης υπολογιστή και μηχανικής εκμάθησης. Το OpenCV κατασκευάστηκε από την Intel το 1999 για να παρέχει μια κοινή υποδομή για εφαρμογές υπολογιστικής όρασης και να επιταχύνει τη χρήση της αντίληψης μηχανών στα εμπορικά προϊόντα. Ως προϊόν με άδεια χρήσης Apache 2, το OpenCV διευκολύνει τις επιχειρήσεις να χρησιμοποιούν και να τροποποιούν τον κώδικα.

Η βιβλιοθήκη διαθέτει περισσότερους από 2500 βελτιστοποιημένους αλγόριθμους, οι οποίοι περιλαμβάνουν ένα ολοκληρωμένο σύνολο κλασικών και υπερσύγχρονων αλγορίθμων υπολογιστικής όρασης και μηχανικής μάθησης. Αυτοί οι αλγόριθμοι μπορούν να χρησιμοποιηθούν για:[6]

- Ανίχνευση και την αναγνώριση προσώπων.
- Αναγνώριση αντικειμένων.
- Καταγραφή ανθρώπινων ενεργειών σε βίντεο.
- Την Ανίχνευση κινήσεων της κάμερας.
- Την εύρεση και σύγκριση παρόμοιων εικόνων από μια βάση δεδομένων.
- Τη βοήθεια ρομπότ αλλά και οχημάτων κατά την πλοήγηση.

Για τους παραπάνω λόγους το OpenCV είναι ευρέως διαδεδομένο και έχει περισσότερους από 47 χιλιάδες χρήστες και εκτιμάται ότι ο αριθμός των λήψεων υπερβαίνει τα 18 εκατομμύρια.

Η Βιβλιοθήκη χρησιμοποιείται απο κολοσούς όπως (Google,Microsoft,IBM,Sony,Toyota κ.α)

Τέλος Πολύ σημαντικό εργαλείο είναι οτι διαθέτει διεπαφές C++, Python, Java και MATLAB και υποστηρίζει Windows, Linux, Android και MacOS.

6.2.1 Java και OpenCV

Η Java είναι άρρητα δεμένη με την βιβλιοθήκη OpenCV. Και αυτό γίνεται διότι η java είναι ευρέως διαδεδομένη για την ανάπτυξη εφαρμογών σε περιβάλλοντα web και επιχειρησιακά συστήματα OpenCV από την άλλη προσφέρει τα Εργαλεία που χρειάζονται για την επεξεργασία μιας εικόνας όπως και την ανάλυση βίντεο,Καθιστώντας έτσι την Java μιας από τις κύριες γλώσσες που δουλεύουν με το OpenCV.

Πλεονεκτήματα χρήσης OpenCV με Java:

- **Εύκολη και ευέλικτη χρήση:** Το OpenCV μπορεί να ενσωματωθεί εύκολα σε Java-based συστήματα και εφαρμογές, επιτρέποντας τη χρήση του σε Web περιβάλλοντα, Mobile εφαρμογές (π.χ., Android), και λογισμικό επιχείρησης.

- **Επεκτασιμότητα:** Η Java είναι ιδανική για ανάπτυξη κλιμακούμενων εφαρμογών. Σε συνδυασμό με το OpenCV, μπορεί να δημιουργηθεί ένα ισχυρό και ευέλικτο σύστημα υπολογιστικής όρασης που επεκτείνεται ανάλογα με τις απαιτήσεις.
- **Real-time επεξεργασία:** Με τη χρήση τεχνολογιών όπως WebSocket μπορούν να δημιουργηθούν real-time εφαρμογές.

Εφαρμογές Java Με OpenCV :

- **Συστήματα ασφαλείας:** Παρακολούθηση καμερών, ανίχνευση κίνησης, και αποστολή ειδοποιήσεων.
- **Ανάλυση εικόνας και βίντεο:** Ανίχνευση προσώπων, αντικειμένων, ή καταστάσεων σε ζωντανή ροή βίντεο.
- **Βιομηχανικές εφαρμογές:** Χρήση σε αυτοματισμούς, παρακολούθηση παραγωγικών διαδικασιών, και ανάλυση δεδομένων σε πραγματικό χρόνο.

6.2.2 Δομή της OpenCV

Η Βασική δομή στην OpenCV Η εικόνα στην OpenCV αναπαρίσταται με τη δομή `cv::Mat`. Αυτή η δομή αυτή δεν είναι τίποτα άλλο από ένα πίνακα δεδομένων που περιέχει τα pixel της εικόνας τα οποία μπορεί να έχουν διάφορα χαρακτηριστικά όπως είναι η φωτεινότητα τα χρώματα RGB Και στην συνέχεια αποθηκεύονται με την μορφή πινάκων δυο οι τριών διαστάσεων 2D Η 3D (ανάλογα με τον τύπο εικόνας).

Έπειτα η εικόνα αναλύεται και μπορούμε να πάρουμε σημαντικές πληροφορίες (όπως το πλάτος, ύψος), ο αριθμός των καναλιών χρώματος (π.χ., 3 για RGB ή 1 για γκρι κλίμακα) και ο τύπος των δεδομένων (π.χ., ακέραιοι 8-bit ή 32-bit). Με την βοήθεια των εργαλείων και φίλτρων (blurring, sharpening κ.α) που μας προσφέρει η βιβλιοθήκη αυτή μπορούμε να αντλήσουμε γρηγορότερα τις πληροφορίες της εικόνας ώστε αυτές να μετατραπούν σε δεδομένα για την μετέπειτα χρήση τους. Στην περίπτωση που έχουμε βίντεο το οποίο προέρχεται από αρχείο ή από κάμερα. Το αρχείο αναπαρίσταται ως μια σειρά από εικόνες το δευτερόλεπτο (frames/sec).

Παράλληλα επιτρέπεται η χρήση και εφαρμογή Φίλτρων για την εύκολο επεξεργασία των δεδομένων. Τέλος μετά την επεξεργασία των δεδομένων υπάρχει η δυνατότητα σύγκρισης των χαρακτηριστικών μέσω των αλγορίθμων μηχανικής όρασης η οποία θα βοηθήσει για την ασφάλεια και σωστή επιλογή των εργαλείων που θα χρησιμοποιήσουμε στην εφαρμογή μας.

6.2.3 Gaussian Blur

Το Gaussian Blur αποτελεί ουσιαστικά ένα μαθηματικό μοντέλο το οποίο εφαρμόζεται στην επεξεργασία της εικόνας. Το χαρακτηριστικό θολώματος Γκαουσιανή μεθόδου εφαρμόζεται θολώνοντας (εξομαλύνοντας) μια εικόνα χρησιμοποιώντας μια Γκαουσιανή συνάρτηση για τη μείωση του επιπέδου θορύβου.

Το Gaussian Blur θεωρείται ένας τύπος φίλτρου χαμηλής διέλευσης, που σημαίνει ότι μειώνει τις λεπτομέρειες υψηλής συχνότητας και τον θόρυβο εξομαλύνοντας τις ανομοιόμορφες τιμές των pixel. Ο αλγόριθμος υπολογίζει την ένταση κάθε pixel με βάση τα pixel που το περιβάλλουν, δίνοντας περισσότερο βάρος στα κοντινά pixel και μειώνοντας σταδιακά το βάρος καθώς αυξάνεται η απόσταση. Αυτό έχει ως αποτέλεσμα μια ομαλή ανάμειξη χρωμάτων και ένα εφέ απαλότητας στην εικόνα.

Το Gaussian blur έχει τη δυνατότητα να:

- Εξομαλύνει τις έντονες μεταβάσεις και μειώνει την εμφάνιση θορύβου.
- Βελτιώνει τα φόντα.
- Μαλακώνει τις έντονες άκρες.
- Μειώνει τις λεπτομέρειες απλοποιεί τις σύνθετες υφές και τα μοτίβα.
- Προετοιμάζει για άλλες επεξεργασίες χρησιμοποιείται συχνά ως βήμα προ επεξεργασίας σε προηγμένο χειρισμό εικόνας, όπως η ανίχνευση και η κάλυψη άκρων.



Το Gaussian Blur δεν είναι μόνο ένα δημοφιλές εργαλείο στην επεξεργασία εικόνας, αλλά και μια έννοια βαθιά ριζωμένη στα μαθηματικά και σε άλλους κλάδους. Οι Gaussian συναρτήσεις χρησιμοποιούνται στη στατιστική για την περιγραφή κανονικών κατανομών, στην επεξεργασία σήματος για τον ορισμό Gaussian φίλτρων και στα μαθηματικά για την επίλυση εξισώσεων θερμότητας και διάχυσης. Αυτές οι σχετικές έννοιες υπογραμμίζουν την ευέλικτη και θεμελιώδη φύση

της Gaussian συνάρτησης πέρα από την εφαρμογή της στην επεξεργασία εικόνας.[7]

6.2.4 Χρήση OpenCV και Gaussian Blur στο σύστημα ασφαλείας.

Η openCV χρησιμοποιεί το Gaussian Blur ως βασικό φίλτρο επεξεργασίας εικόνας. Στο σύστημα ασφαλείας που υλοποιήθηκε η openCV συνέβαλε στην αναγνώριση της κάμερας και την ανίχνευση κίνησης μέσω της επεξεργασίας και σύγκρισης των συνεχόμενων εικόνων που δίνει η κάμερα.

Ενώ το Gaussian blur χειρίζεται την εξομάλυνση θορύβου και την μείωση διαφορών για περισσότερη ακρίβεια ακόμα και σε συνθήκες χαμηλού φωτισμού και με όχι τόσο καλή κάμερα.

.Στην main τοποθετήθηκαν οι βιβλιοθήκες και έπειτα στην java class με όνομα camera χρησιμοποιήθηκαν οι λειτουργίες της openCV βιβλιοθήκης η κάμερα συγκρίνει απλά δυο frames εικόνων από τον χώρο που θα τοποθετηθεί η κάμερα και αν υπάρχει διαφορά σημαίνει ότι ανιχνεύτηκε η κίνηση

Οι εντολές οι οποίες χρησιμοποιήθηκαν φαίνονται στον παρακάτω πίνακα.

OpenCV Εντολή	Επεξήγηση
<pre>public static void openCamera() { videoCapture.open(0); }</pre>	Ανοίγει μέσω της openCV την camera του υπολογιστή.
<pre>Mat frame1 = new Mat();</pre>	Αναγνώριση καρτέ 1 και 2.
<pre>Imgproc.GaussianBlur</pre>	Εφαρμόζω στην εικόνα 1 και 2 gaussian blur φίλτρο (εξομάλυνση θορύβου)
<pre>Imgproc.cvtColor</pre>	Μετατροπή σε ασπρόμαυρες τις δυο εικόνες(gray scale)
<pre>Core.absdiff(first, second, diff);</pre>	Βρίσκει την διαφορά από τις δύο εικόνες
<pre>Imgproc.threshold(diff, diff, 50, 255, Imgproc.THRESH_BINARY); //binarization</pre>	Μετατρέπει την διαφορά σε δυαδική μορφή για γρηγορότερη επεξεργασία

Πίνακας 3: Εντολές χρήσης Gaussian Blur

6.3 HTML

Η HTML(HyperText Markup Language) δηλαδή Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου και βασίζεται στη γλώσσα SGML, Standard Generalized Markup Language, που είναι ένα από τα μεγαλύτερο σύστημα επεξεργασίας εγγράφων και είναι η βασική γλώσσα με την οποία πραγματοποιείται η δόμηση σελίδων του Παγκόσμιου Ιστού.

Η HTML αναπτύχθηκε για πρώτη φορά το 1990 από τον Tim Berners-Lee από το CERN, σε συνεργασία μια επιστημονική ομάδα από το Ευρωπαϊκό Εργαστήριο Μοριακής Φυσικής της Γενεύης, δημιούργησαν ένα νέο πρωτόκολλο, με το οποίο μπορούσε να μεταφερθεί κάθε είδους πληροφορία μέσα στο Διαδίκτυο.

Το πρωτόκολλο αυτό ονομάστηκε HTTP (HyperText Transfer Protocol) και σηματοδότησε την αρχή του Παγκόσμιου Ιστού.

Η HTML είναι η πιο διαδεδομένη μορφή δημιουργίας ιστοσελίδων διότι μπορεί να τρέξει στα περισσότερα προγράμματα περιήγησης (broswer).Παράλληλα όμως για την εγγραφή αρχείων τύπου HTML χρειαζόμαστε απλούς συντάκτες κειμένων όπως το σημειωματάριο για windows η το Geany για Linux.

Χρήσεις HTML:

- **Δημιουργία ιστοσελίδων:** Το HTML είναι η γλώσσα που χρησιμοποιείται για την ανάπτυξη της δομής μιας ιστοσελίδας. Σε συνδυασμό με το CSS (για τη μορφοποίηση) και τη JavaScript (για τη λειτουργικότητα), συνθέτει το front-end (προσκήνιο) των περισσότερων ιστοσελίδων και web εφαρμογών.
- **Εφαρμογές Ιστού (Web Apps):** Οι σύγχρονες web εφαρμογές, όπως οι υπηρεσίες ηλεκτρονικού ταχυδρομείου και οι πλατφόρμες κοινωνικής δικτύωσης, χρησιμοποιούν HTML για την απόδοση του περιεχομένου τους στις συσκευές των χρηστών.
- **Εφαρμογές κινητών:** Μέσω τεχνολογιών όπως το PWA (Progressive Web Apps), το HTML συνδυάζεται με άλλες τεχνολογίες και χρησιμοποιείται για την ανάπτυξη εφαρμογών που μπορούν να λειτουργούν σε κινητές συσκευές χωρίς τη χρήση native frameworks.

Υλοποίηση Συστήματος - Βιβλιοθήκες Εργαλεία

Η αρχική έκδοση της HTML δημιουργήθηκε το 1991. Η αρχική έκδοση περιείχε λίγες ετικέτες για την οργάνωση απλών ιστοσελίδων. Έπειτα ακολούθησαν και άλλες εκδόσεις όπως απεικονίζονται στο παρακάτω πίνακα:

Πίνακας 4: HTML εκδόσεις ανά έτος.

Έκδοση HTML	Χρονολογία	Διαφοροποιήσεις
HTML	1991	Πρώτη έκδοση και δημιουργία βασικών ετικετών όπως <p>, <a>,).
HTML 2.0	1995	Επίσημη τυποποίηση από το IETF. Υποστήριξη βασικών φορμών και βελτίωση διασυνδέσεων.
HTML 3.2	1997	Εισαγωγή Βασικών ετικετών όπως πινάκων <table> για την βελτιστοποίηση και καλύτερη οργάνωση του ιστού
HTML 4.01	1999	Εισαγωγή του μοντέλου CSS για τον καλύτερο διαχωρισμό δομής και στυλ. Παράλληλα διορθώθηκαν ατέλειες
HTML 5	2014	Για πρώτη φορά γίνεται υποστήριξη πολυμέσων (video, audio) χωρίς εξωτερικά plugins. Υποστήριξη μοντέλων όπως τα WebSockets.
HTML 5.1	2016	Βελτίωση της υπάρχουσας προσθήκες κάποιων ετικετών
HTML 5.2	2017	Νέα api authentication API και άλλες μικρές βελτιώσεις στη δομή και τα χαρακτηριστικά ασφαλείας.

Η HTML δεν είναι μια γλώσσα προγραμματισμού αλλά μια περιγραφική γλώσσα, δηλαδή ένας ειδικός τρόπος γραφής κειμένου. Ορίζει ένα σύνολο κοινών στυλ για τις Web σελίδες, όπως τίτλοι (titles), επικεφαλίδες (headings), παράγραφοι (paragraphs), λίστες (lists) και πίνακες (tables).

Κάθε στοιχείο έχει ένα όνομα και περιέχεται μέσα στα σύμβολα < >, που αποκαλούνται ετικέτες (tags).

Οι ετικέτες(tags) HTML είναι τα θεμελιώδη δομικά στοιχεία της ανάπτυξης Ιστού, παρέχοντας τη δομή και την οργάνωση που είναι απαραίτητη για τη δημιουργία ιστοσελίδων. Περιλαμβάνουν ετικέτες για επικεφαλίδες, παραγράφους, συνδέσμους, εικόνες και άλλα. Ετικέτες που χρησιμοποιούνται συνήθως όπως <html>, <head> και <body> είναι απαραίτητες για τη δημιουργία ενός εγγράφου υπερκειμένου

αυτές οι ετικέτες παρέχουν το πλαίσιο για τη δημιουργία φιλικών προς τον χρήστη και προσβάσιμοι ιστοσελίδων.

Κεφάλαιο 6ο

Κάποια από τα πιο συνηθισμένα tags που χρησιμοποιούνται στην HTML ακολουθούν στους παρακάτω πίνακες:

Πίνακας 5: HTML Βασικές εντολές

HTML (Labs)	Σύνταξη	Περιγραφή ετικέτας
<div>	<div>...</div>	Ορίζει μια διαίρεση ή μια ενότητα σε ένα έγγραφο HTML.
<p>	<p>...</p>	Ορίζει μια παράγραφο.
<a>	...	Ορίζει μια υπερσύνδεση.
		Ενσωματώνει μια εικόνα.
	...	Καθορίζει ένα στοιχείο λίστας.
<table>	<table>...</table>	Ορίζει έναν πίνακα.
<form>	<form>...</form>	Καθορίζει μια φόρμα HTML για εισαγωγή χρήστη.
<input>	<input type="...">	Καθορίζει ένα στοιχείο ελέγχου εισόδου σε μια φόρμα.
<h1>-<h6>	<h1>...</h1>	Ορίστε επικεφαλίδες διαφορετικών επιπέδων.
<label>	<label for="...">...</label>	Καθορίζει μια ετικέτα για ένα στοιχείο εισόδου
<meta>	<meta name="name" content="value">	Παρέχει πληροφορίες σχετικά με τη διάταξη της σελίδας
<script>	<script src="script.js"></script>	Ενσωματώνει ή συνδέει αρχεία JavaScript για τη λειτουργία της ιστοσελίδας.
<!DOCTYPE html>	<!DOCTYPE html>	Ορίζει την έκδοση του HTML ως HTML5 έγγραφο.

Η χρήση της HTML έγινε για τον σχεδιασμό της ιστοσελίδας και την μορφοποίηση των τίτλων και όλου του περιεχομένου σε frontend επίπεδο(αυτό που βλέπει ο χρήστης).

Η HTML ήταν απαραίτητη για την ενσωμάτωση του web server την λήψη και την αποστολή καρτέ με μέσω mail καθώς και την εισαγωγή scripts στον κώδικα (κωδικός ήχος).

Ο συνδυασμός HTML με JavaScript είναι πολύ σημαντικός, καθώς το HTML παρέχει τη δομή της ιστοσελίδας, ενώ η JavaScript προσθέτει διαδραστικότητα.

6.4 JavaScript

Η javascript αποτελεί μία από τις πιο ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού παγκοσμίως .Δημιουργήθηκε το 1985 από τον Brendan Eich και τη εταιρείας Netscape. Η javascript είναι μια γλώσσα σεναρίων.

Ο κώδικας της εισάγεται μέσα σε σελίδες html και εκτελείται στον browser του χρήστη οι πιο γνωστοί browsers(Firefox Chrome Opera) χρησιμοποιούν διερμηνευτή javascript.

Η javascript είναι μια αντικειμενοστραφής αλλά και συναρτησιακή γλώσσα. Διοτι μπορεί να τρέχει και στον πελάτη (client) μέσω του webbrowser όσο και στον διακομιστή (server) με την χρήση διαφόρων περιβαλλόντων όπως το Node.js.

Κάποια από τις βασικές χρήσεις της javascript είναι :

- Η αλληλεπίδραση με τον χρήστη με την βοήθεια συμβάντων όπως το πάτημα ενός κουμπιού η εισαγωγή δεδομένων η το scroll down σε μια σελίδα.
- Παρέχει την δυνατότητα μεταβολή του περιεχομένου HTML και CSS μέσω της διεπαφής DOM(Document Object Model).
- Ο έλεγχος εισαγωγής σε φόρμες αλλά και η εναλλαγή των εικόνων.
- Η ευκολότερη χρήση στο browser διότι δεν απαιτεί εγκατάσταση εργαλείων η μεταγλωττιστή και υποστηρίζεται από τους περισσότερους browser.

Η χρήση της είναι απλή και αρκεί ένα απλό σημειωματάριο για τον προγραμματισμό των εντολών επίσης επειδή είναι γλώσσα σεναρίων μπορεί να χρησιμοποιηθεί οπουδήποτε μέσα στο html αρχείο ενός προγράμματος.

6.4.1 Εφαρμογές Javascript

Η Javascript λόγω της ανεξαρτησίας του browser (αφού τρέχει στους περισσότερους) είναι ιδανική για και για front-end αλλά και για back-end εφαρμογές αντιστοίχως. Κάποιες από τις οποίες είναι:

- Διαδραστικά κουμπιά sliders εικόνων
- Live Data (όπως η άμεση ενημέρωση του καιρού)
- Real time εφαρμογές
- Web servers και websocket servers
- Επεξεργασία αρχείων.
- Mobile εφαρμογές παχνίδια αλλά και iot εφαρμογές.

6.4.2 Χρήση JavaScript στο Σύστημα Ασφαλείας

Στο πλαίσιο της παρούσας πτυχιακής εργασίας, η JavaScript χρησιμοποιείται για την υλοποίηση της διεπαφής του χρήστη. Μέσα από τη σελίδα HTML, η JavaScript ελέγχει δυναμικά την εμφάνιση της εικόνας από την κάμερα, την ενεργοποίηση καταγραφής, την εισαγωγή κωδικού πρόσβασης, και την προβολή ειδοποιήσεων σε πραγματικό χρόνο μέσω WebSocket.

Επιπλέον, χρησιμοποιείται για την αποστολή αιτημάτων προς το backend server, για την ενεργοποίηση λειτουργιών όπως η αποστολή email ειδοποίησης σε περίπτωση ανίχνευσης κίνησης. Η σύνδεση HTML JavaScript επιτρέπει στον χρήστη να αλληλεπιδρά άμεσα με το σύστημα παρακολούθησης μέσω ενός απλού και λειτουργικού γραφικού περιβάλλοντος. Με την χρήση scripts τοποθετούμε τον κωδικό εισόδου. Ένα παράδειγμα απο javascript αρχείο του κώδικα του συστήματος απεικονίζεται στο παρακάτω γράφημα.

```
<script>
  // Έλεγχος κωδικού πρόσβασης
  document.addEventListener("DOMContentLoaded", function() {
    var password = prompt("give password:");
    if (password === "1234") { // Σωστός κωδικός
      document.querySelector('.container').style.display = 'block'; // Εμφάνιση περιεχομένου
    } else {
      alert("wrong password!");
      window.location.href = "https://www.google.com"; // Ανακατεύθυνση αν είναι λάθος
    }
  })

```

Ταυτόχρονα μέσω της javascript αντλούνται τα δεδομένα μέσω του websocket server εμφανίζεται η ζωντανή ροή των βίντεο στην σελίδα καθώς και γίνεται η αποστολή των email μέσω της υπηρεσίας emailjs . Ακολουθεί πίνακας των επιπλέον εντολών javascript που χρησιμοποιήθηκαν στο σύστημα

ασφαλείας.

<pre>const ws = new WebSocket("ws://" + location.hostname + ":8080/socket");</pre>	+	Καθορίζει το path και επιτρέπει την επικοινωνία στην πύλη με διεύθυνση http://localhost:8080/
<pre>ws.onmessage=function(event){const data= JSON.parse(event.data); const base64Image=data.image;</pre>		Παίρνει τα data από τον websocket server σε περίπτωση ανίχνευσης κίνησης και τα στέλνει με την μορφή κωδικοποίησης σαν json object για την επεξεργασία και προβολή τους από τον χρήστη.
<pre>emailjs.init("lrCFbRNezw4pS1Q6D");</pre>		Ορίζει την έναρξη της χρήσης της υπηρεσίας Emailjs.

Πίνακας 6: Websocket Method

6.4.3 Emailjs

Η Emailjs είναι μια δωρεάν online υπηρεσία με την οποία δίνεται η δυνατότητα στον χρήστη να αποστέλλει email χωρίς την χρήση SMTP server ή κάποιας άλλης backend εφαρμογής

Η ρύθμιση περιλαμβάνει τη δημιουργία ενός email template (HTML μορφή) στην πλατφόρμα του EmailJs, στο οποίο προσαρμόζονται δυναμικά μεταβλητές όπως το περιεχόμενο του μηνύματος και η διεύθυνση email καθώς και ένα username το οποίο είναι μοναδικό για κάθε χρήστη.

Η συγκεκριμένη εφαρμογή ξεκινάει με ανίχνευση των δύο διαφορετικών frames από την camera και την σύγκριση των εικόνων αυτών μέσω της opencv. Στην συνέχεια μέσω του websocket server η εικόνα στέλνει σήμα στο frontend κομμάτι το οποίο με την βοήθεια της γλώσσας σεναρίων javascript και την εντολή που απεικονίζεται στην παρακάτω εικόνα αποστέλλεται την εικόνα μετά την ανίχνευση κίνησης στο email του χρήστη.

```
emailjs.send("service_x7ezqyf", "template_ech0tz9", {
  message: "Ανιχνεύθηκε εικόνα από το σύστημα ασφαλείας.",
  image_base64: compressedImage
}).then(function(response) {
  console.log("📧 Email sent successfully:", response.status);
}, function(error) {
  console.error("Email failed:", error);
});
```

6.5 Communication protocol (Πρωτόκολλα Επικοινωνίας)

Για την επίτευξη και υλοποίηση του projects και επειδή αφορά μια online Bell time εφαρμογή με το raspberrypi απαραίτητη είναι η χρήση κάποιου πρωτοκόλλου επικοινωνίας (communication protocol) για να μπορέσει να υπάρξει η σύνδεση στο διαδίκτυο

Τα πρωτόκολλα επικοινωνίας αποτελούν σύνολα κανόνων και προτύπων που βοηθούν στην ανταλλαγή δεδομένων μεταξύ συσκευών ή συστημάτων. Καθορίζουν τη μορφή, το χρονοδιάγραμμα, την αλληλουχία και τους μηχανισμούς ελέγχου σφαλμάτων που είναι απαραίτητοι για αξιόπιστη επικοινωνία. Τα πρωτόκολλα επικοινωνίας επιτρέπουν στις συσκευές να κατανοούν και να ερμηνεύουν τα δεδομένα που μεταδίδονται, διασφαλίζοντας τη συμβατότητα και τη σωστή και άρτια λειτουργία του συστήματος.

Τα χρησιμοποιούμε σχεδόν καθημερινά, αν και οι περισσότεροι χρήστες δεν το γνωρίζουν και δεν καταλαβαίνουν πώς λειτουργούν. Αυτά τα πρωτόκολλα επιτρέπουν τη μεταφορά δεδομένων από τις συσκευές μας, ώστε να μπορούμε να περιηγηθούμε σε ιστότοπος, να στείλουμε email, να ακούσουμε μουσική στο διαδίκτυο κλπ.

[8]

6.5.1. IOT και communication protocol

Το Internet of Things (IoT) είναι μια νέα τεχνολογία που αναπτύχθηκε τα τελευταία χρόνια όπου γίνεται η διασύνδεση μεταξύ του Διαδικτύου και των φυσικών αντικειμένων. Οι εφαρμογές IoT καλύπτουν πληθώρα πεδίων, που κυμαίνονται από μεγάλες βιομηχανικές εφαρμογές έως μικρές καθημερινές χρήσεις.

Αυτές οι εφαρμογές συνδέονται σε πραγματικό χρόνο και μεταφέρουν διαφορετικές πληροφορίες. Επομένως, υπάρχει γύρω μας μια σειρά από αντικείμενα ικανά να συλλέγουν, να στέλνουν και να επεξεργάζονται δεδομένα και να επικοινωνούν μεταξύ τους χρησιμοποιώντας διάφορες τεχνολογίες, συγκεκριμένα: Big data, cloud computing, πρωτόκολλα κ.λπ. Έτσι, μία από τις κύριες πτυχές του IoT είναι η επικοινωνία μεταξύ των διαφορετικών στοιχείων του παγκόσμιου συστήματος χάρη σε ένα σύνολο πρωτοκόλλων δικτύου, έτσι ώστε να υπάρχει μια ποικιλία προτύπων και πρωτοκόλλων που επιτρέπουν στα φυσικά αντικείμενα να αλληλεπιδρούν μεταξύ τους, άρα ασφαλής μεταφορά πληροφοριών από συσκευές IoT .

Ωστόσο, η ποικιλία των πρωτοκόλλων δικτύου που προσφέρονται στην αγορά από οργανισμούς και ερευνητές απαιτεί προσεκτική μελέτη της λειτουργίας των πρωτοκόλλων που θα προσαρμόσουμε στις εφαρμογές μας.[9]

Κάθε πρωτόκολλο πρέπει να μπορεί να πληροί ορισμένες αυστηρές απαιτήσεις. Θα πρέπει να είναι σε θέση να παρέχει λειτουργικότητα στη σωστή Διαδικασία. Υπάρχουν πολλές απαιτήσεις που μας επιτρέπουν να αξιολογήσουμε ένα πρωτόκολλο. Πρώτον, το πρωτόκολλο πρέπει επομένως να έχει

Υλοποίηση Συστήματος - Βιβλιοθήκες Εργαλεία

λειτουργικότητα που να επιτρέπει τη διαχείριση της μορφής δεδομένων του. Δεύτερον, κατά τη διαδικασία μετάδοσης δεδομένων, θα ήταν απαραίτητο να διαχειριστείτε τη μορφή διεύθυνσης και τη δρομολόγηση δεδομένων. Τρίτον, ένα πρωτόκολλο θα πρέπει να μπορεί να ανιχνεύει αυτά τα σφάλματα. Επομένως, για να διαχειριστείτε την απώλεια πληροφοριών.

Τέλος, το πρωτόκολλο πρέπει να διασφαλίζει την κατεύθυνση της ροής πληροφοριών και τον έλεγχο της ακολουθίας. [9]

Η ανάγκη για μια αρχιτεκτονική πρωτοκόλλων αναπτύχθηκε διότι όταν ένας ηλεκτρονικός υπολογιστής όπως και άλλες συσκευές επεξεργασίας δεδομένων ανταλλάσσουν δεδομένα ,οι διαδικασίες μπορεί να είναι πολύ περίπλοκες καθώς απαιτείται μεγάλος βαθμός συνεργασίας μεταξύ των δύο συστημάτων για αυτό τον λόγο λοιπόν αντί η όλη διαδικασία να υλοποιείται ως ένα ενιαίο κομμάτι χωρίζεται και διαμορφώνεται ξεχωριστά σε επιμέρους τμήματα .

Με αποτέλεσμα να έχουμε μια αρχιτεκτονική πρωτοκόλλου η οποία οργανώνει τα τμήματα (modules) σε στοίβες .Οι οποίες με την σειρά τους εκτελούν ένα μέρος σχετικών λειτουργιών για την επικοινωνία των συστημάτων μεταξύ τους.

Υπάρχουν πολλές διαφορετικές κατηγορίες πρωτοκόλλων επικοινωνίας οι οποίες κατηγοριοποιούνται βάση την διαφορετική εφαρμογή και τα χαρακτηριστικά τους αναφορικά κάποιες από τις βασικές κατηγορίες φαίνονται στον παρακάτω πίνακα(Πίνακας 2).[10]

Network Protocols	Web Protocols	File Transfer Protocols	Wireless Communication Protocols	Security Protocols	Industrial Protocols	Email Protocols:
Internet Protocol (IP)	Hypertext Transfer Protocol (HTTP)	File Transfer Protocol (FTP)	Bluetooth	Secure Shell (SSH)	Modbus	Simple Mail Transfer Protocol (SMTP)
Transmission Control Protocol (TCP)	Hypertext Transfer Protocol Secure (HTTPS)	Secure File Transfer Protocol (SFTP)	Wi-Fi (IEEE 802.11)	Secure Socket Layer(SSL)	Profibus	PostOffice Protocol (POP)

Κεφάλαιο 6ο

User Datagram Protocol (UDP)	Simple Object Access Protocol (SOAP)	Trivial File Transfer Protocol (TFTP)	Zigbee	IPsec (internet protocol Security)	Devicenet	Internet Message Access Protocol (IMAP)
Internet Control Message Protocol (ICMP)	Representational State Transfer (REST)	Trivial File Transfer Protocol (TFTP)	Near Field Communication (NFC)	Secure File Transfer Protocol (SFTP)	CAN (Controller Area Network)	
Routing Information Protocol	Websocket	Network File System (NFS)	Long Term Evolution (LTE)	Kerberos		

Πίνακας 7: Βασικές κατηγορίες Πρωτοκόλλων ασφαλείας

Αυτές οι κατηγορίες καλύπτουν ένα ευρύ φάσμα πρωτοκόλλων επικοινωνίας που χρησιμοποιούνται στη δικτύωση, τις εφαρμογές Ιστού, τα συστήματα email, τις μεταφορές αρχείων, την ασύρματη επικοινωνία, την ανταλλαγή μηνυμάτων, την ασφάλεια και τον βιομηχανικό αυτοματισμό. Κάθε κατηγορία έχει συγκεκριμένα πρωτόκολλα σχεδιασμένα για τους αντίστοιχους σκοπούς και τεχνολογίες της.

6.5.2. Κατηγορίες Πρωτοκόλλων Ασφαλείας

Σε αυτό το κεφάλαιο θα δοθούν επιγραμματικά κάποιες από τις πιο βασικές κατηγορίες των πρωτοκόλλων ασφαλείας.

- **Zigbee:**

Το Zigbee είναι ένα πρωτόκολλο ραδιοεπικοινωνίας που βασίζεται στο 802.14.4, που θεωρείται σημαντική τεχνολογία για εφαρμογές IoT οικιακού αυτοματισμού. Το 2005, η γέννησή του πρόκειται να δεχτεί για το IoT.14 Δεκεμβρίου 2004, επικυρώθηκε η προδιαγραφή IEEE 802.15.4-2003 Zigbee. Με μέση εμβέλεια 10 μέτρα. Χρησιμοποιεί χαμηλό εύρος ζώνης, επομένως είναι ιδανικό για μεταφορά δεδομένων.

- **Bluetooth:**

Το Bluetooth εφευρέθηκε το 1994 από τη σουηδική εταιρεία Ericsson. Αναπτύσσεται από την Special Interest Group (SIG) όπου έχει δημοσιεύσει το Bluetooth 5 προκειμένου να προσαρμόσει αυτά τα πρωτόκολλα στο IoT. Αυτή η τεχνολογία είναι μέρος των ασύρματων επικοινωνιών. Το πρωτόκολλο Bluetooth επιτρέπει αμφίδρομη ασύρματη μεταφορά και ανταλλαγή δεδομένων σε πολύ μικρή απόσταση χρησιμοποιώντας ραδιοκύματα UHF σε ζώνη συχνοτήτων 2,4 GHz. Χρησιμοποιεί χαμηλό εύρος ζώνης. Κατά συνέπεια, το Bluetooth επιτρέπει τη μεταφορά μικρής ποσότητας δεδομένων σε μικρές αποστάσεις. Επιπλέον, προσφέρει μια λύση χαμηλής ενέργειας και χαμηλό κόστος για ραδιομεταδόσεις UHF σε μικρή απόσταση. Το Bluetooth χρησιμοποιείται σε πολλές εφαρμογές και συσκευές IoT: έξυπνο ρολόι, αυτοματοποιημένα έξυπνα σπίτια, κινητά τηλέφωνα, αυτοκίνητα κ.λπ. Το Bluetooth είναι εύκολο στη χρήση, εγκατάσταση και προσαρμογή, αλλά έχει χαμηλή ασφάλεια.

- **Επικοινωνία κοντινού πεδίου NFC :**

Το πρωτόκολλο NFC είναι μια επέκταση του ISO / IEC 14443 που τυποποιεί τις κάρτες εγγύτητας χρησιμοποιώντας RFID. Έτσι, βασίζεται στην τεχνολογία αναγνώρισης ραδιοσυχνοτήτων. Το NFC είναι ένα πρότυπο για ασύρματη επικοινωνία ραδιοσυχνοτήτων μικρής εμβέλειας και υψηλής συχνότητας με μικρή ποσότητα. Ως εκ τούτου, έχει μια ετικέτα μόνο για ανάγνωση. Αυτή η ετικέτα περιέχει μια μικρή ποσότητα δεδομένων, η οποία μπορεί να αλλάξει αργότερα από το αντικείμενο. Επιτρέπει την ανταλλαγή πληροφοριών μεταξύ συσκευών μέχρι απόσταση περίπου 10 cm. Επιπλέον, το NFC παρέχει αναγνώριση, απλή και αμφίδρομη επικοινωνία μεταξύ δύο ηλεκτρονικών αντικειμένων που είναι εξοπλισμένα με τσιπ RFID. Αυτά τα αντικείμενα έχουν ετικέτα και αναγνωρίζονται αυτόματα από τη ραδιοσυχνότητα. Το NFC χρησιμοποιείται ευρέως για τον έλεγχο συσκευών IoT σε διαφορετικά περιβάλλοντα όπως: έξυπνο σπίτι, σήματα για εμπορικούς χώρους, εργοστασιακές και βιομηχανικές εφαρμογές.

- **Transmission Control Protocol (TCP): TCP:**

Η αρχιτεκτονική TCP/IP είναι το αποτέλεσμα έρευνας η οποία χρηματοδοτήθηκε από την DARPA (Defense Advanced Research projects Agency) των ΗΠΑ, Το TCP είναι ένα αξιόπιστο πρωτόκολλο προσανατολισμένο στη σύνδεση που χρησιμοποιείται για τη μετάδοση δεδομένων μέσω του Διαδικτύου. Διασπά τα δεδομένα σε πακέτα, διασφαλίζει ότι παραδίδονται με τη σωστή σειρά και εκτελεί έλεγχο σφαλμάτων για να διασφαλίσει την ακεραιότητα των δεδομένων. Εφαρμόζεται στα τερματικά συστήματα και εξασφαλίζει ότι τα δεδομένα θα είναι αξιόπιστα στην κατάλληλη εφαρμογή

- **Internet Protocol (IP) Πρωτόκολλο Διαδικτύου (IP):**

Το IP είναι ένα πρωτόκολλο επιπέδου δικτύου που παρέχει τον μηχανισμό διεύθυνσης και δρομολόγησης για πακέτα δεδομένων στο Διαδίκτυο. Αυτό το πρωτόκολλο είναι υλοποιημένο όχι μόνο στα τερματικά αλλά και στους δρομολογητές (router) δηλαδή τους επεξεργαστές που η κύρια τους λειτουργία είναι να μεταδίνουν δεδομένα από το ένα δίκτυο στο άλλο. Ορίζει μοναδικές διευθύνσεις IP για συσκευές και επιτρέπει τη δρομολόγηση πακέτων μεταξύ δικτύων.[10]

- **Hypertext Transfer Protocol Πρωτόκολλο μεταφοράς υπερκειμένου (HTTP):**

Το HTTP είναι το θεμελιώδες πρωτόκολλο του παγκόσμιου ιστού και χρησιμοποιείται για την μεταφορά πληροφοριών με την χρήση υπερκειμένων. Τα δεδομένα που μεταφέρονται από το πρωτόκολλο μπορεί να είναι κείμενο, εικόνα ήχος ή οποιαδήποτε πληροφορία προσβάσιμη μέσω διαδικτύου. Το HTTP αποτελεί ένα πρωτόκολλο προσανατολισμένο στις συναλλαγές πελάτη-διακομιστή (server-client). Η πιο συνηθισμένη χρήση του HTTP είναι ανάμεσα σε ένα λογισμικό πλοήγησης web browser και ένα διακομιστή ιστού web server ενώ για την αξιοπιστία μετάδοσης των δεδομένων βασίζεται στο TCP. Ωστόσο στο HTTP η κάθε συναλλαγή αντιμετωπίζεται ξεχωριστά. Κατά συνέπεια για κάθε νέα συναλλαγή μεταξύ πελάτη και διακομιστή δημιουργεί μια νέα σύνδεση η οποία μετά την διεκπεραίωση της τερματίζεται.

- **Το πρωτόκολλο μεταφοράς αρχείων File Transfer Protocol (FTP):**

Το πρωτόκολλο αυτό χρησιμοποιείται για να αποστέλλονται αρχεία από ένα σύστημα σε ένα άλλο βάση των εντολών που έδωσε ο χρήστης διαμέσω διαδικτύου. Εξυπηρετεί αρχεία κειμένου όπως και δυαδικά αρχεία. Και ουσιαστικά χρησιμοποιεί μια σύνδεση TCP με το σύστημα προορισμού για να διασφαλίσει την σωστή και ακέραια μεταφορά των δεδομένων.

- **Simple Mail Transfer Protocol Απλό πρωτόκολλο μεταφοράς email (SMTP):**

Το συγκεκριμένο πρωτόκολλο παρέχει μια στοιχειώδη υπηρεσία ηλεκτρονικού ταχυδρομείου. Δηλαδή τον μηχανισμό για την μεταφορά μηνυμάτων μεταξύ απομακρυσμένων υπολογιστών. Μερικά από τα

χαρακτηριστικά του SMTP είναι οι λίστες ταχυδρομείου (mail listings) η προώθηση καθώς και η παραλαβή μηνυμάτων .Το SMTP καθορίζει τον τρόπο με το οποίο θα δημιουργηθούν τα μηνύματα .Για τη δημιουργία των μηνυμάτων σε κάθε υπολογιστή απαραίτητη προϋπόθεση αποτελεί η χρήση υπηρεσίας ηλεκτρονικού ταχυδρομείου .

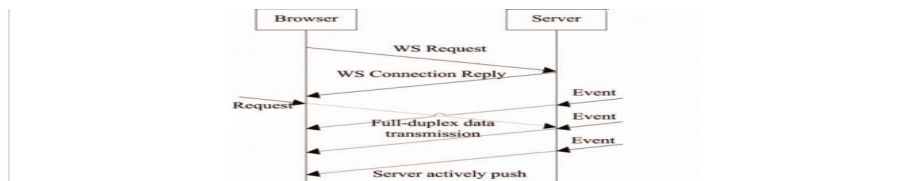
- **Secure Shell (SSH): Το SSH** είναι ένα κρυπτογραφικό πρωτόκολλο δικτύου που παρέχει ασφαλή απομακρυσμένη πρόσβαση και ασφαλή μεταφορά αρχείων. Επιτρέπει στους χρήστες να συνδέονται με ασφάλεια σε απομακρυσμένα συστήματα και να εκτελούν εντολές ή να μεταφέρουν αρχεία με ασφάλεια.
- **Wi-Fi:** Το Wi-Fi είναι ένα δημοφιλές πρωτόκολλο ασύρματης επικοινωνίας που χρησιμοποιείται για τοπική δικτύωση. Επιτρέπει στις συσκευές να συνδέονται σε ασύρματο δίκτυο, συνήθως χρησιμοποιώντας τα πρότυπα IEEE 802.11, επιτρέποντας πρόσβαση στο Διαδίκτυο και μεταφορά δεδομένων χωρίς φυσικά καλώδια.
- **Universal Serial Bus (USB):** Το USB είναι ένα κοινό πρωτόκολλο για τη σύνδεση περιφερειακών συσκευών με υπολογιστές και άλλες συσκευές υποδοχής. Επιτρέπει τη μεταφορά δεδομένων και ενέργειας μεταξύ συσκευών, όπως σύνδεση πληκτρολογίου, ποντικιού ή εκτυπωτή σε υπολογιστή.[11]
- **Websocket:**Το Websocket είναι ένα πρωτόκολλο επικοινωνίας, το οποίο ορίστηκε από την Hyper Text Markup Language 5 (HTML5) το 2008 και στη συνέχεια έγινε το διεθνές πρότυπο το 2011. Όλα τα προγράμματα περιήγησης ιστού υποστηρίζουν αυτήν την τεχνολογία push server τώρα. Το μεγαλύτερο πλεονέκτημα του websocket είναι ότι τόσο ο διακομιστής όσο και ο πελάτης μπορούν να στέλνουν πληροφορίες ο ένας στον άλλον, γεγονός που το καθιστά μια πλήρη αμφίδρομη επικοινωνία μέσω μιας ενιαίας σύνδεσης TCP. Είναι μια πολλά υποσχόμενη τεχνική για τη δημιουργία ενός καναλιού αμφίδρομης επικοινωνίας χαμηλού κόστους σε πραγματικό χρόνο, που υποστηρίζει αυθαίρετα πρωτόκολλα επιπέδου εφαρμογής.[12]

6.6 Websocket Protocol

Το πρωτόκολλο websocket είναι ένα πρωτόκολλο επικοινωνίας διπλής κατεύθυνσης για την ασύγχρονη ανταλλαγή δεδομένων .Το πρωτόκολλο αυτό βασίζεται σε μια σύνδεση TCP και λόγω της συνεχούς ανταλλαγής δεδομένων με τον client δίνει στο κύκλωμα ασφαλείας μας μια γρήγορη ανταπόκριση στην επικοινωνία καθώς και μειώνει τον όγκο του φορτίου του δικτύου.

Κεφάλαιο 6ο

Το WebSocket παρέχει έναν πιο αποτελεσματικό μηχανισμό δυαδικής μεταφοράς δεδομένων σε σύγκριση με πρωτόκολλα που βασίζονται σε κείμενο όπως το HTTP επομένως όταν η εφαρμογή ασχολείται με μεγάλες ποσότητες δεδομένων ή απαιτεί επικοινωνία χαμηλής καθυστέρησης το WebSocket προσφέρει καλύτερη απόδοση. Προς το παρόν, τα περισσότερα προγράμματα περιήγησης υποστηρίζουν ήδη το WebSocket, όπως Chrome, Safari, Opera και FireFox6 κλπ, αλλά το IE9 δεν υποστηρίζεται

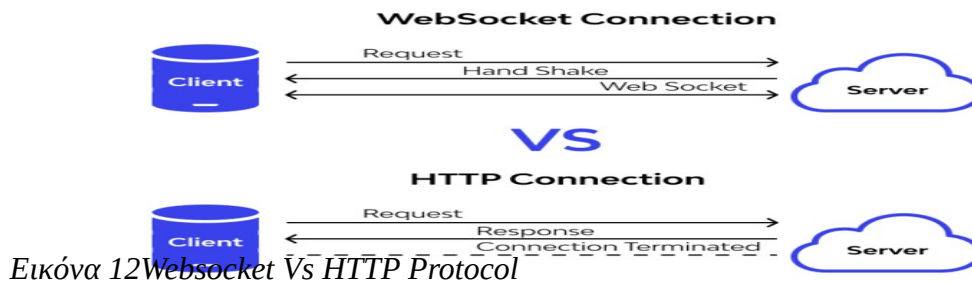


6.6.1 Websocket vs HTTP.

Το WebSocket όπως και το HTTP (βλ ΚΕΦ) πρωτόκολλο αποτελούν δύο πρωτόκολλα επικοινωνίας που χρησιμοποιούνται κατά την σύνδεση πελάτη διακομιστή (server-client)

Ή HTTP είναι μονής κατεύθυνσης όπου ο πελάτης στέλνει το αίτημα και ο διακομιστής στέλνει την απάντηση. Ο χρήστης στέλνει ένα αίτημα στον διακομιστή με τη μορφή HTTP ή HTTPS ο διακομιστής λαμβάνει ένα αίτημα και στέλνει απάντηση στον πελάτη. Κάθε αίτημα σχετίζεται με μια αντίστοιχη απάντηση, μετά την αποστολή της απάντησης η σύνδεση κλείνει.

Εάν ο πελάτης στείλει 10 αιτήματα στον διακομιστή, θα ανοίξουν 10 ξεχωριστές συνδέσεις TCP. Το WebSocket είναι αμφίδρομο, ένα πλήρες αμφίδρομο πρωτόκολλο που χρησιμοποιείται και αυτό για την επικοινωνία πελάτη-διακομιστή, σε αντίθεση με το HTTP το Websocket ξεκινά από ws:// ή wss://. Είναι ένα πρωτόκολλο το οποίο εξασφαλίζει ότι η σύνδεση μεταξύ πελάτη και διακομιστή θα παραμείνει ζωντανή έως ότου τερματιστεί από οποιοδήποτε μέρος (πελάτη ή διακομιστή). Ή επικοινωνία γίνεται δια μέσω μιας χειραψίας (hand shake) μεταξύ client και server. Μετά το κλείσιμο της σύνδεσης από κάποιον πελάτη(client) ή διακομιστή(sever), η σύνδεση τερματίζεται και από τα δύο άκρα.



Εικόνα 12 WebSocket Vs HTTP Protocol

Το HTTP μπορεί να χρησιμοποιηθεί εάν η εφαρμογή ακολουθεί ένα παραδοσιακό μοτίβο αιτήματος-απόκρισης, όπου οι πελάτες ανακτούν ή υποβάλλουν πληροφορίες κυρίως σε έναν διακομιστή και δεν απαιτούν ενημερώσεις σε πραγματικό χρόνο επιπλέον υποστηρίζεται από όλα τα προγράμματα περιήγησης ιστού οπότε σε περίπτωση που η εφαρμογή απαιτεί ευρύ φάσμα εφαρμογών και συσκευών το HTTP αποτελεί λύση.

Το WebSocket λόγω της αμφίδρομης και συνεχόμενης επικοινωνίας πελάτη διακομιστή είναι καλύτερο να χρησιμοποιηθεί σε εφαρμογές πραγματικού χρόνου λόγω του ότι επιτρέπει την αποτελεσματική και χαμηλής καθυστέρησης ροή δεδομένων μεταξύ του πελάτη και του διακομιστή, επιτρέποντας ενημερώσεις σε πραγματικό χρόνο, ειδοποιήσεις, λειτουργικότητα.

Τέλος εάν η εφαρμογή χρειάζεται να διατηρήσει μια μακροχρόνια σύνδεση μεταξύ του πελάτη και του διακομιστή για να επιτρέψει τη συνεχή επικοινωνία χωρίς την επιβάρυνση της δημιουργίας νέων συνδέσεων για κάθε αλληλεπίδραση το WebSocket είναι ο καλύτερος τρόπος .

.[13]

6.6.2 Χρήση websocket στο σύστημα ασφαλείας

Σκοπός της χρήσης του websocket είναι η επικοινωνία και η μετάδοση της εικόνας στον browser.

<pre>Spark.webSocket("/socket", WebSocketHandler.class);</pre>	<p>Μέσω του spark και της κλάσης websockethandler. Γίνεται η χρήση του websocket.</p>
<pre>@OnWebSocketConnect</pre>	<p>Όταν ανοίξει την σελίδα στο browser ο χρήστης ενεργοποιείται και η websocket σύνδεση</p>

Η websocket σύνδεση μέσω της κλάσης websockethandler αποτελεί τον βασικό πυρήνα του server ο οποίος διαχειρίζεται την livestreaming μετάδοση των εικόνων μέσω **startService** μεθόδου κατά την

οποία ορίζεται ένας timer ο οποίος κάθε 300ms (0.3 sec) ο οποίος ανανεώνει την εικόνα.

Έπειτα μέσω της μεθόδου `broadcastMessage()` λαμβάνει τα καρτέτα κωδικοποιεί σε μορφή base64 και τα αποστέλλει με μορφή JSON στον websocket server.

6.7 Spark

Το Spark δεν είναι τίποτα παραπάνω από ένα Framework δηλαδή ένα εργαλείο λογισμικού ή πλατφόρμα που παρέχει έναν δομημένο τρόπο ανάπτυξης εφαρμογών καθώς περιλαμβάνει βιβλιοθήκες, πρότυπα και οδηγίες που βοηθούν τον προγραμματισμό παρέχοντας μια βάση και μια βασική δομή.

Τα frameworks προσφέρουν συχνά ένα σύνολο επαναχρησιμοποιήσιμων λειτουργιών που επιλύουν κοινές προκλήσεις ανάπτυξης, όπως χειρισμό συνδέσεων βάσεων δεδομένων, διαχείριση επαφών χρήστη ή την διασφάλιση χρήσης κάποιων αλγορίθμων.

Συνολικά, τα frameworks έχουν σχεδιαστεί για να απλοποιούν και να επιταχύνουν τη διαδικασία ανάπτυξης παρέχοντας ένα δομημένο περιβάλλον και κοινά δομικά στοιχεία.

Το Spark είναι ένα ανοιχτού κώδικα framework. Το Apache Spark ξεκίνησε το 2009 ως ένα ερευνητικό έργο στο AMPLab του CCC Berkley. Ήταν ουσιαστικά μια συνεργασία μεταξύ φοιτητών και καθηγητών η επικεντρώθηκε σε τομείς εφαρμογών με μεγάλο φόρτο δεδομένων. Ο στόχος του Spark projects ήταν να δημιουργήσει ένα νέο framework, βελτιστοποιημένο για γρήγορη επαναληπτική επεξεργασία και ανάλυση μεγάλων δεδομένων. Όπως η μηχανική εκμάθηση και η διαδραστική ανάλυση δεδομένων. Η βελτιστοποιημένη μηχανή εκτέλεσης του Spark επιτρέπει επίσης την αποτελεσματική παράλληλη επεξεργασία σε καταναμημένα συμπλέγματα (distributed clusters.).

Συνολικά το Spark χρησιμοποιείται ευρέως σε εφαρμογές ανάλυσης και επεξεργασίας μεγάλων δεδομένων.

Η δυνατότητα του να υποστίζεται από πολλές γλώσσες προγραμματισμού όπως Scala, Java, Python και η ταχύτητα στην επεξεργασία δεδομένων που επιτυγχάνεται μέσω της δυνατότητας αποθήκευσης δεδομένων στη μνήμη. Εκτελώντας έτσι τις λειτουργίες πολύ πιο γρήγορα από τα παραδοσιακά συστήματα που βασίζονται σε δίσκο.

Σκοπός της χρήσης του spark framework είναι η ενσωμάτωση αρχείων όπως HTML και javascript στον websocket server του συστήματος.

<code>import spark.</code>	Spark;	Εισαγωγή της κλάσης του spark .
<code>Spark.port(8080);</code>		Ορίζει την πόρτα που τρέχει ο server .
<code>Spark.staticFiles.location("/web");</code>		Ορίζει τον φάκελο που περιέχει αρχεία ήχου εικόνας,HTML,Javascript.

Υλοποίηση Συστήματος - Βιβλιοθήκες Εργαλεία

<code>Spark.staticFiles.expireTime(600);</code>	Ορίζει τον χρόνο που κρατάει την εικόνα στα 10 λεπτά μετά κάνει request ξανά.
<code>Spark.init();</code>	Εκκινεί τον webserver αρχή των πάνω εντολών.

Πίνακας 8: *Spark Framework*

Κεφάλαιο 7ο Συμπεράσματα και Προτάσεις βελτιώσης

7.1 Συμπεράσματα

Η ανάπτυξη του συστήματος ασφαλείας με την χρήση κάμερας και την παρακολούθηση μέσω site αποτελεί μια σύγχρονη εφαρμογή που βαδίζει στα πλαίσια των IOT εφαρμογών.

Η ανάπτυξη του συστήματος αυτού έγινε με την βοήθεια της γλώσσας προγραμματισμού java σε βασικό επίπεδο backend προγράμματος. Άλλα και με την χρήση εργαλείων όπως openCV βιβλιοθήκης της javascript CSS και HTML. Επιπλέον για την επικοινωνία με τον server έγινε η χρήση websocket.

Εν κατακλείδι καταλήγουμε ότι το σύστημα μας είναι πλήρως λειτουργικό και καλύπτει τις απαιτήσεις ενός συστήματος ασφαλείας διότι:

- Είναι εύκολα προσβάσιμο στο χρήστη από οποιαδήποτε εφαρμογή browser.
- Λόγο του ότι έχει χρησιμοποιηθεί java μας διευκολύνει στην χρήση του σε διαφορετικά λειτουργικά συστήματα (ανεξαρτησία πλατφόρμας).
- Συλλέγει Μέσω της βιβλιοθήκης openCV τα καρτέ και τα συγκρίνει με εργαλεία απλοποίησης και προ επεξεργασίας για την ευκολότερη ανάλυση δεδομένων.
- Κάνει live streaming παρακολούθηση και ανιχνεύει την κίνηση στον χώρο.
- Σε περίπτωση ανίχνευσης κίνησης στέλνει email στον χρήστη με φωτογραφία και οτι ανιχνεύτηκε κίνηση.
- Τέλος με την χρήση του πρωτοκόλλου websocket επιτυγχάνεται η εύκολη και γρήγορη ανταλλαγή δεδομένων server client καθώς επιτρέπεται η αποστολή δεδομένων χωρίς την απαίτηση αιτήματος από τον χρήστη

Το συμπέρασμα είναι ότι το σύστημα αυτό μας δείχνει πως μπορούν να εφαρμοστούν σύγχρονες τεχνολογίες που μέσω αυτών μπορεί να υλοποιηθεί ένα οικονομικό σύστημα ασφαλείας μέσω κάμερας χαμηλού κόστους υλικών αλλά και κατανάλωσης το οποίο μπορεί εύκολα να αναπτυχθεί σε προγραμματιστικό επίπεδο και να αποδώσει πολλές βελτιώσεις για ακόμα πιο καινοτόμες και χρήσιμες εφαρμογές του IOT.

7.2 Προτάσεις Βελτίωσης

Παρότι το σύστημα ασφαλείας που υλοποιήθηκε αποτελεί μια σύγχρονη εφαρμογή θα μπορούσε να βελτιωθεί και να αναπτυχθεί με ποικίλους τρόπους

1. Στο κομμάτι ασφαλείας θα μπορούσε να προστεθεί αντί της χρήσης της υπηρεσίας emails η backend java βιβλιοθήκη java-mail μέσω τις οποίας θα γίνεται καλύτερη και ασφαλέστερη απόκρυψη του ονόματος χρήστη και των λοιπών κωδικών API για την αποτροπή τυχόν αποστολής κακόβουλων email στον χρήστη της εφαρμογής.
2. Επιπλέον για την καλύτερη απόδοση του συστήματος και την αποφυγή λανθασμένων ειδοποιήσεων στον χρήστη μπορεί να προστεθεί τεχνολογία AI για την ανίχνευση προσώπων αντικειμένων η ακόμα και κατοικίδιων .
3. Για την καλύτερη παρακολούθηση περισσότερων χώρων θα μπορούσε να γίνει η τοποθέτηση πολλαπλών καμερών στο σύστημα ασφαλείας οι οποίες θα εμφανίζονται στην ίδια σελίδα. Με την προσθήκη αισθητήρων μπορεί να γίνει μια επιπλέον κλάση για την ανίχνευση κίνησης μέσω PIR sensors και έτσι την καλύτερη παρακολούθηση της εισόδου καθώς και των παραθύρων του σπιτιού.
4. Ταυτόχρονα μπορεί να δημιουργηθεί εύκολα μια εφαρμογή σε ios και android για την χρήση της εφαρμογής σε smartphome και tablet.
5. Τέλος η εφαρμογή μπορεί να υποστηρίξει σύνδεση της εφαρμογής με το cloud για ανταλλαγή στιγμιότυπων φωτογραφιών μέσω εφαρμογών πλατφόρμων όπως το google drive για την καλύτερη αποθήκευση των δεδομένων.

Κεφάλαιο 7ο
ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία

ISBN-13: 978-0-13-359162-0 Modern Operating Systems

ISBN 978-960-418-549-8 ασύρματες επικοινωνίες ΚΑΙ δίκτυα Corry Beard

978-0-12-817736-5 Machine Learning for Subsurface Characterization

journal Articles

[1] *Tanenbaum, A. S. (2013). Modern Operating Systems (3rd Edition), Pearson. ISBN-13: 978-0-13-359162-0*

[2] *2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*

[3] *2022 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*

[4] *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*

[5] *13th International Workshop on Program Comprehension (IWPC'05)*

[6] *opencv2014 International Carnahan Conference on Security Technology (ICCST)*

[7] *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*

[8] *2013 IEEE International Conference on **Image** Processing*

[9] *2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*
ISBN:978-1-6654-2222-2

ISBN:978-1-6654-2222-2

[10] *2022 5th International Conference on Energy, Electrical and Power Engineering (CEEPE) :ISBN:978-1-6654-7906-6*

[11] *ISBN 978-960-418-549-8 CORY BEARD*

[12] *2019 Chinese Control Conference (CCC) :ISBN:978-9-8815-6397-2*

[13] *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC) ISBN:978-1-4799-2565-0*

Ιστοσελίδες .

<https://opencv.org/>

<https://www.raspberrypi.com/>

<https://www.jetbrains.com/>

<https://projects.iq.harvard.edu/photographpreservationprogram/harvards-history-photography-timeline-text-only>

<https://www.w3schools.com/>

<https://www.java.com/en/>

<https://www.researchgate.net/>

<https://www.sciencedirect.com/>

Κεφάλαιο 8ο

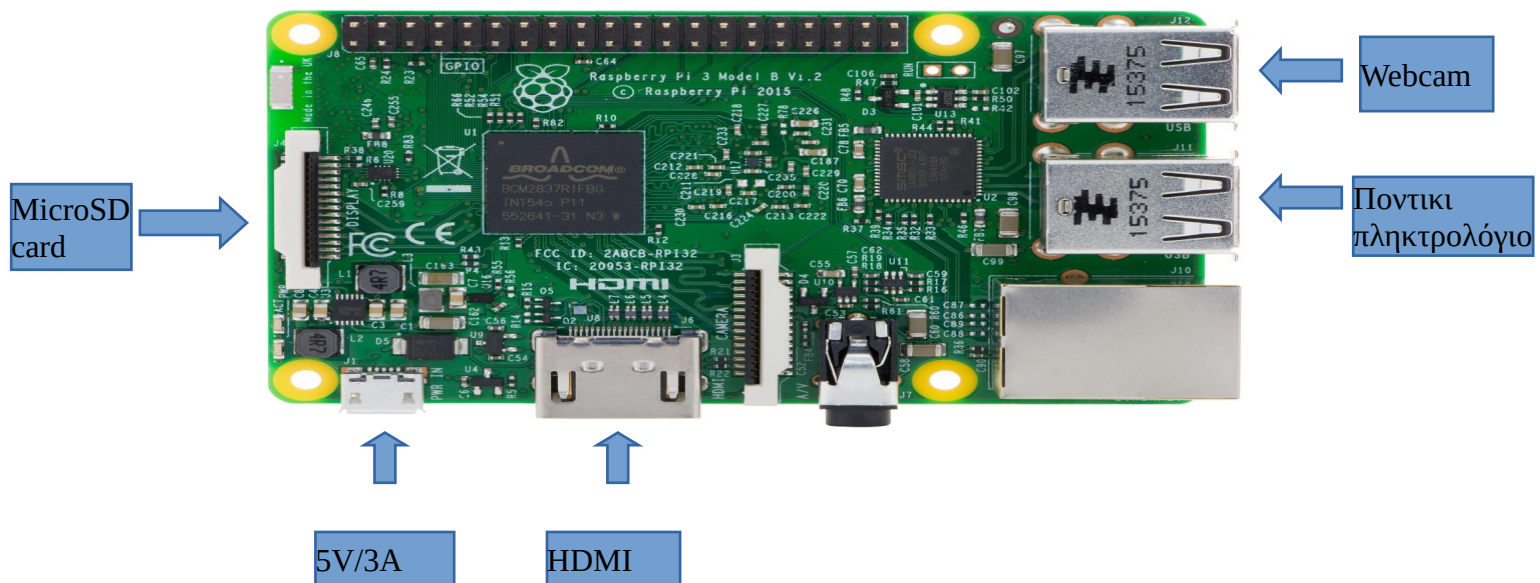
Παράρτημα-Πειραματικό μέρος - Επεξήγηση Κώδικα

Hardware Κομμάτι :Για την υλοποίηση του πρότζεκτ θα χρειαστεί να δοθεί έμφαση τόσο hardware όσο και στο Software κομμάτι Από Hardware θα χρησιμοποιηθεί το RaspberryPi 3 Model B+(2018) το οποίο θα λειτουργεί σαν τον σερβερ αλλά και τον υπολογιστή όπου θα γίνει η επεξεργασία των δεδομένων ν.Μια web camera στην προκειμένη περίπτωση θα είναι η Xiaomi Mi WebCam Imilab Με κόστος τα 12ευρ.Ένας Μετατροπέας τάσης 5V/3A .Μαζί με μια κάρτα microSD η οποία στην προκειμένη περίπτωση αυτή είναι η Samsung evo plus 32GB.Η οποία θα λειτουργήσει σαν σκληρός δίσκος για το πρόγραμμα μας καθώς και το Raspberry pi Linux based Raspbian καθώς και ο κώδικας σε java.

Ακολουθεί εικόνα της σύνδεσης των hardware που αποτελείται από.

1. Μετατροπέας 5v/3a
2. Micro SD Card για το λογισμικό και κώδικα
3. HDMI Cable για χρήση με οθόνη
4. Ποντίκι,πληκτρολόγιο για τις εντολές που θα χρησιμοποιηθούν
5. Web camera για την καταγραφή του χώρου

Μετά την εγκατάσταση του κώδικα και την εκκίνηση του προγράμματος το ποντίκι και το πληκτρολόγιο δεν είναι απαραίτητα.



Software Κομμάτι :

Επεξήγηση του προγράμματος (Περιληπτικά)

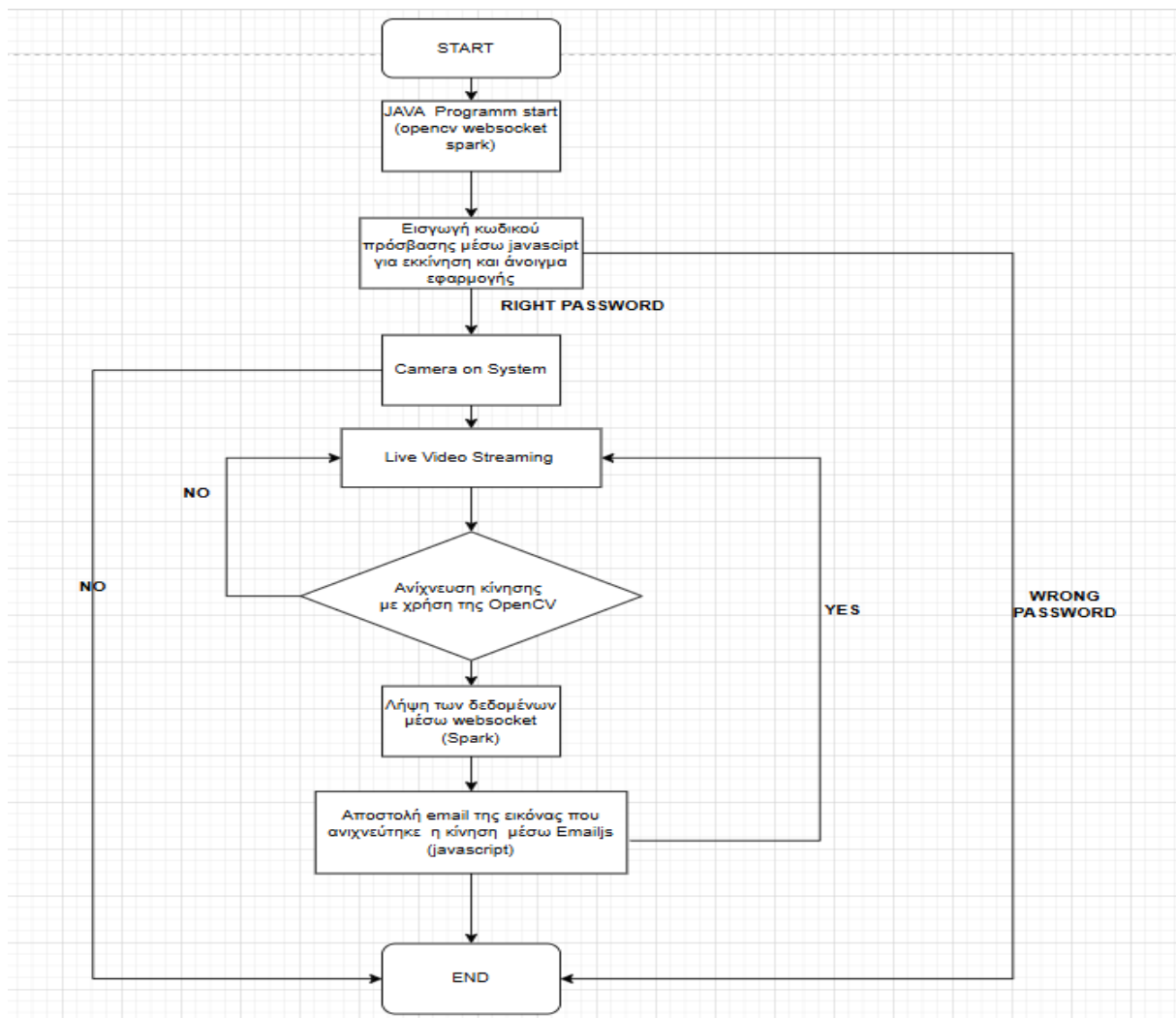
1. Το πρόγραμμα μας ξεκινάει με την main.java ουσιαστικά γίνεται η εκκίνηση του spark server στην θέση port 8080.Έπειτα με την βοήθεια της openen βιβλιοθήκης ανοίγουμε την κάμερα μέσω της μεθόδου Camera.openCamera().Ορίζουμε την τοποθέτηση των στατικών στοιχείων (όπως το HTML αρχείο και τα σχετικά αρχεία JavaScript) στο φάκελο /web.Ουσιαστικά εδώ με την βοήθεια του spark framework αρχικοποιούμε τον server και ανοίγουμε την κάμερα για να έχουμε την ζωντανή λήψη video μέσω websocket.
2. Στην συνέχεια καλούμε την κλάση WebSocketHandler η οποία διαχειρίζεται τη σύνδεση websocket .Και όταν ο χρήστης (client) συνδέεται καλείται η μέθοδος connected(Session session).Επιπλέον ορίζεται ένας timer ο οποίος στέλνει τα δεδομένα σε τακτά χρονικά διαστήματα μέσω του websocket (300ms) και με την μέθοδο broadcastMessage() στέλνουμε τα δεδομένα που έχουν ληφθεί από την κάμερα στην μορφή JSON.
3. Καλούμε την μέθοδο opencamera() η οποία με την βοήθεια της openen ανοίγει την κάμερα του υπολογιστή μας.Η μέθοδος encodeWebcamFrame() τραβάει δύο διαδοχικά πλάνα από την κάμερα τα συγκρίνει και μας δείχνει αν υπάρχει κάποια κίνηση στον χώρο στην συνέχεια κωδικοποιούμε την εικόνα σε μορφή base64 ώστε να μπορέσει να σταλεί μέσω της websocket σύνδεσης. Τέλος αν ανιχνευτεί κίνηση παίζει τον ήχο και ειδοποιεί για την ανίχνευση συναγερμού. Ουσιαστικά ενσωματώνει την επεξεργασία εικόνας και αν ανιχνευτεί κίνηση στέλνει τα δεδομένα ήχου και εικόνας.
4. Με την βοήθεια της javascript δημιουργούμε μια σύνδεση με τον websocket server ("**ws://" + location.hostname + ":" + location.port + "/socket"**).Όταν ο πελάτης (client) λάβει ένα μήνυμα από τον server το μήνυμα προβάλλεται στην ιστοσελίδα σε πραγματικό χρόνο και αν ανιχνευτεί κίνηση ενσωματώνει και τον ήχο.
5. Με την χρήση της HTML σελίδας μορφοποιούμε αυτό που θα βλέπει ο χρήστης όταν συνδέεται μέσω του browser στην σελίδα localhost:8080 την γενική μορφή δηλαδή της ιστοσελίδας μέσω της οποίας ο χρήστης θα μπορεί να παρακολουθεί live τον χώρο που έχει τοποθετήσει την κάμερα ασφαλείας.
6. Τέλος με την χρήση της υπηρεσίας emailjs και την βοήθεια javascript εισάγουμε στο html αρχείο τον κωδικό πρόσβασης και την δυνατότητα αποστολής email στον χρήστη όταν ανιχνευτεί κίνηση

Κεφάλαιο 8ο

Σύνοψη Ροής Εργασίας

1. **Η εκκίνηση του server** (μέσω Spark και WebSocket) δημιουργεί τον server που ακούει για συνδέσεις.
2. Όταν ένας client συνδέεται στο WebSocket, το σύστημα αρχίζει να στέλνει τα δεδομένα της κάμερας (επεξεργασμένα σε εικόνες και ανιχνεύσεις κίνησης) μέσω του WebSocket.
3. Ο client λαμβάνει τις εικόνες σε **Base64 String** και τις εμφανίζει στην ιστοσελίδα του, ενώ αν ανιχνευτεί κίνηση, αναπαράγεται ο ήχος του συναγερμού.
4. Επιπλέον πριν την είσοδο του και για την έναρξη του προγράμματος πρέπει να τοποθετηθεί κωδικός εισόδου τον οποίο ο χρήστης ορίζει μέσω javascript
5. Το πρόγραμμα λειτουργεί σε πραγματικό χρόνο, προσφέροντας **ζωντανή παρακολούθηση και ανίχνευση κίνησης** μέσω του συστήματος κάμερας και ειδοποιήσεων email μέσω της υπηρεσίας emailjs.

Διάγραμμα Ροής Εργασίας



Η εγκατάσταση του λογισμικού Raspberry Pi εξηγείτε στα παρακάτω βήματα

- 1) Μπαίνουμε στην σελίδα <https://www.raspberrypi.com/software/> και κατεβάζουμε το αρχείο Raspberry Pi Imager στον υπολογιστή μας αναλόγως το λειτουργικό σύστημα που διαθέτουμε

Κεφάλαιο 8ο

Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. [Watch our 45-second video](#) to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

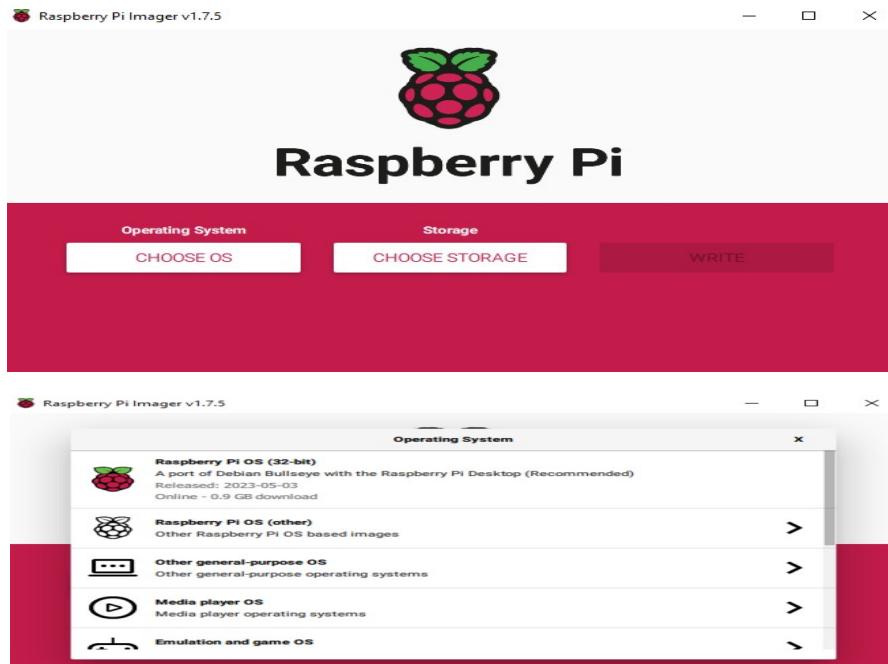
[Download for Windows](#)

[Download for macOS](#)

[Download for Ubuntu for x86](#)

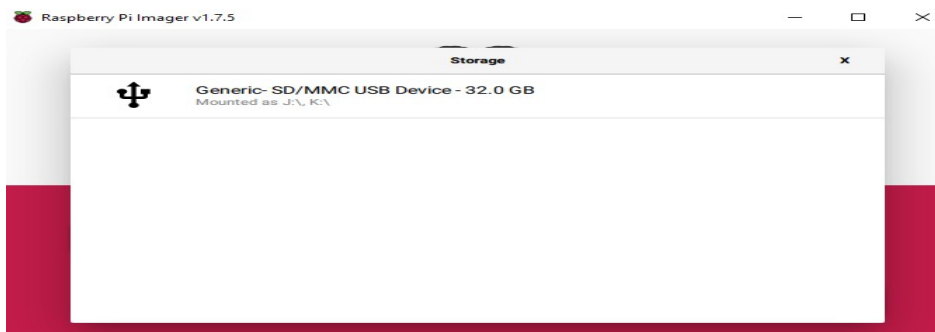
To install on **Raspberry Pi OS**, type `sudo apt install rpi-imager` in a Terminal window.

- 2) Αφου τρέξουμε το παραπάνω αρχείο στον υπολογιστή μας επιλέγουμε πιο version λογισμικό μας ενδιαφέρει ανάλογα με την εφαρμογή που θέλουμε να τρεξουμε για παραδειγμα:



Παράρτημα-Πειραματικό μέρος - Επεξήγηση Κώδικα

- 3) Έπειτα αφού επιλέξουμε το OS operating system που μας ταιριάζει κάνουμε κλικ στο choose storage και επιλέγουμε την θέση της micro SD Card που χρησιμοποιείται και σαν σκληρός δίσκος για το OS .



- 4) αφού τελειώσει και η επιλογή της θέσης που θα εγκατασταθεί λογισμικό δεν μένει παρά να πατήσουμε το κουμπί write



- 5) Μετά την επιτυχή εγγραφή εμφανίζεται το παρακάτω μήνυμα .

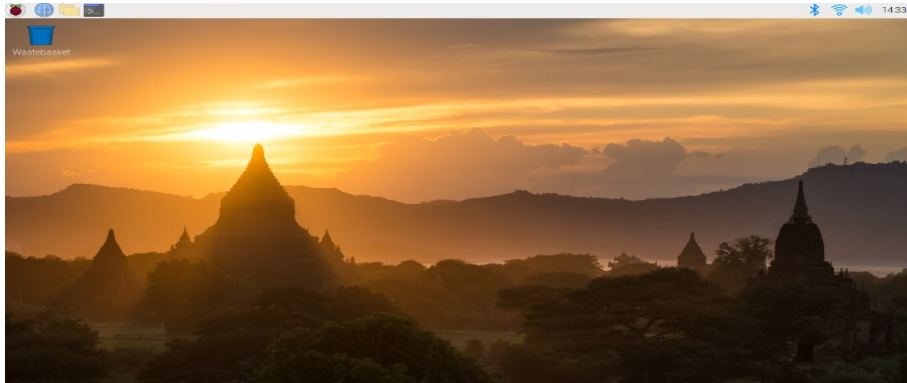
Κεφάλαιο 8ο



- 6) Αφού ολοκληρωθεί η διαδικασία εγκατάστασης του λογισμικού στην κάρτα μνήμης δεν μένει παρά να αφαιρέσουμε την κάρτα και την τοποθετήσουμε στην θέση (socket) του Raspberry pi.
- 7) Μετά την τοποθέτηση της κάρτας SD συνδέουμε τα περιφερειακά (ποντίκι οθόνη πληκτρολόγιο webcam) και συνδέουμε τον Adapter 5V/2A στο pi για εκκίνηση.
- 8) Με την ολοκλήρωση της διαδικασίας εκκίνησης. Το Raspberry Pi OS θα δημιουργήσει έναν οδηγό διαμόρφωσης συστήματος. Το Raspberry Pi OS θα ζητήσει να ρυθμιστούν πολλές διαμορφώσεις. Θα πρέπει να επιλέξετε τις προτιμώμενες ρυθμίσεις για την επιλογή Γλώσσα τις ρυθμίσεις του πληκτρολογίου, του δικτύου, την ζώνη ώρας και τον λογαριασμό χρήστη σύνδεσης.
- 9) Με την ολοκλήρωση της διαδικασίας ρύθμισης, το σύστημα θα ζητήσει επανεκκίνηση για να ολοκληρωθεί η διαδικασία. Αφού γίνει η επανεκκίνηση εμφανίζεται η παρακάτω οθόνη που μας δηλώνει ότι το λογισμικό έτρεξε σωστά.



- 10) Τέλος μας ανοίγει η επιφάνεια εργασίας του pi Os η οποία μοιάζει αρκετά με μια windows επιφάνεια εργασίας (desktop)



Εγκατάσταση προγράμματος στο raspberry pi

Αφού δημιουργήσουμε το πρόγραμμα μας και εγκαταστήσουμε το λογισμικό raspbian

Το μόνο που έχουμε είναι να εγκαταστήσουμε της βιβλιοθήκες που χρησιμοποιήσαμε κατά την gradle δημιουργία του αρχείου καθώς και την έκδοση της java που χρησιμοποιήσαμε.

Αυτό γίνεται μέσω του εικονιδίου του τερματικού bash (4ο εικονίδιο αριστερά πάνω φωτογραφία).

Εκεί πληκτρολογούμε .

1. Download βιβλιοθήκης

```
wget https://download.oracle.com/java/18/archive/jdk-18.0.2.1\_linux-aarch64\_bin.tar.gz (sha256)
```

2. Αποσυμπίεση του αρχείου

```
tar -xvzf jdk-18.0.2.1_linux-aarch64-linux.jdk.tar.gz
```

3. `sudo mv jdk-18.0.2.1* /opt/ jdk-18.0.2.1`

4. Εγκατάσταση της βιβλιοθήκης jdk-18.0.2.1 (default) ως με τις εντολές

```
sudo update-alternatives --install /usr/bin/java java /opt/jdk-18.0.2.1/bin/java 1
```

```
sudo update-alternatives --install /usr/bin/javac javac /opt/jdk-18.0.2.1/bin/javac 1
```

Κεφάλαιο 8ο

```
sudo update-alternatives --set java /opt/jdk-18.0.2.1/bin/java
```

```
sudo update-alternatives --set javac /opt/jdk-18.0.2.1/bin/javac
```

5. Δημιουργούμε ένα φάκελο movement detection master και εισάγουμε το usb stick με το πρόγραμμα σε μορφή τύπου zip(συμπιεσμένο).Μέσα στον φάκελο που δημιουργήσαμε κάνουμε έναν υποφάκελο και βάζουμε τις ακόλουθες εξαρτήσεις (gradle).

```
spark-core-2.9.4.jar
```

```
javacv-platform-1.5.8.jar
```

```
json-20220924.jar
```

```
slf4j-simple-2.0.6.jar
```

6. `cd ~/Desktop unzip movement detection master`
7. `cd ~/Desktop/movement detection master java -cp "libs/*" -Djava.library.path=. MainClass`
8. Τρέχει την main

```
java -cp "movement detection master.jar:libs/*" Main
```

9. `@reboot java -cp "/home/pi/Desktop/movement detection master/libs*/:/home/pi/Desktop/movement detection master/movement detection master.jar" -Djava.library.path=/home/pi/Desktop/ movement detection master com.example.Main`

EmailJs

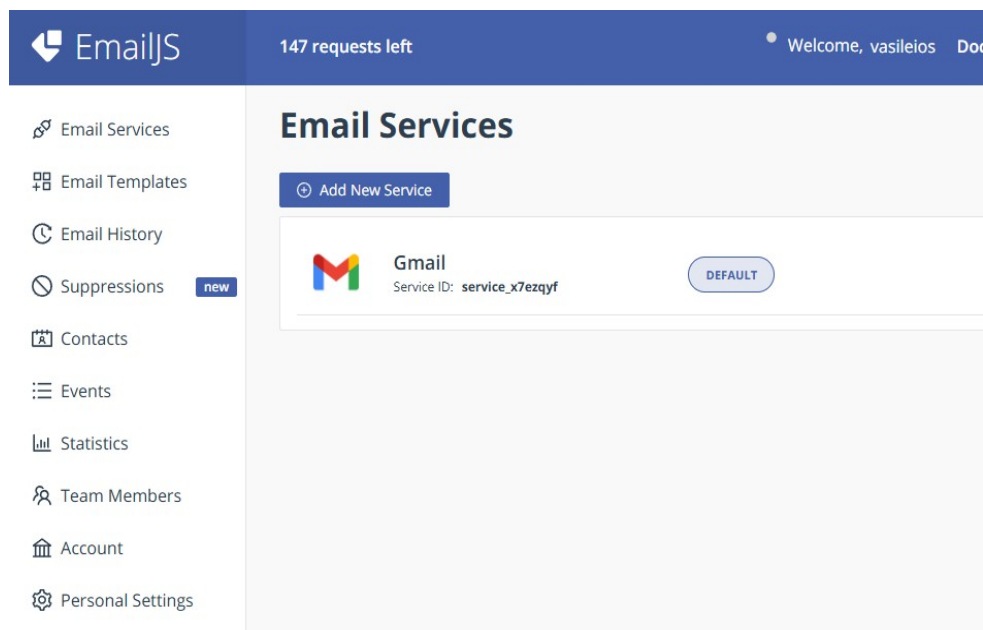
1. Για την χρήση της υπηρεσίας emailjs ο χρήστης πρέπει απλα να ακολουθήσει τον σύνδεσμο: <https://www.emailjs.com/>

2. Στην συνέχεια αφού βάλει το email εγγραφής και επιβεβαιώσει με σύνδεσμο στο email του του εισάγει τα username και password που έδωσε κατά την εγγραφή του στην πλατφόρμα κάνει είσοδο στον σύνδεσμο <https://dashboard.emailjs.com/admin>
3. Το interface της emailjs ακολουθεί σε εικόνα
4. Αφού την αριστερή καρτέλα όπως εμφανίζεται στην παρακάτω εικόνα ο χρήστης πρέπει να ρυθμίσει το email tamplate και το email service. Αυτό το κάνει ο χρήστης από την αριστερή καρτέλα και πατάει

email services→ add new service

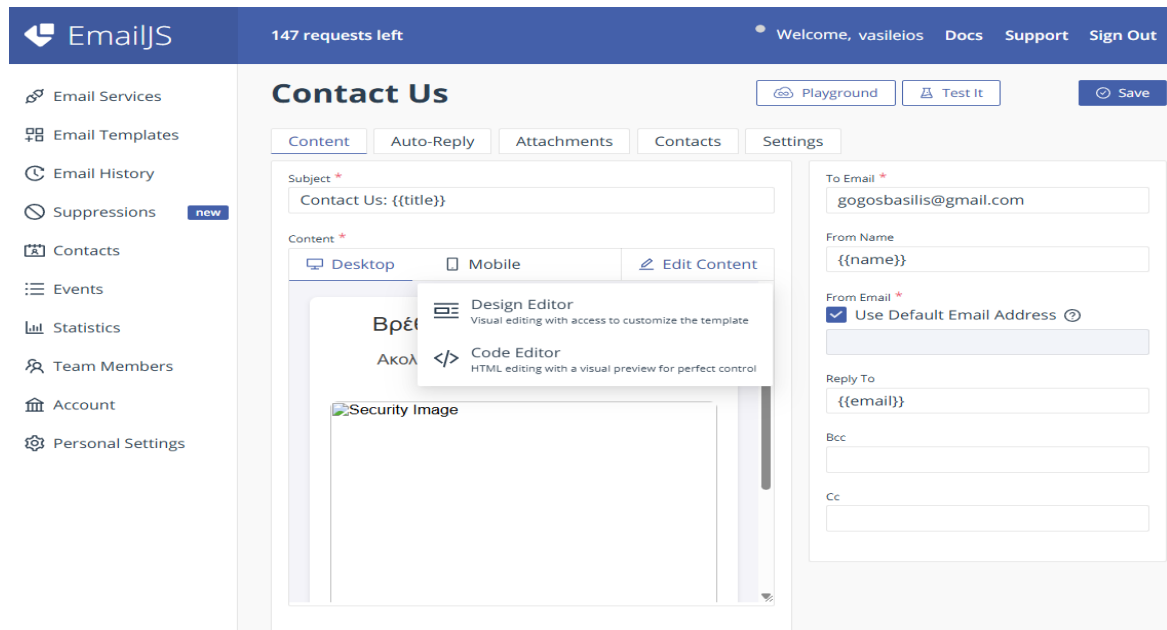
email template→ create new template

5. Αφού εισάγει τον τύπο του email που χρησιμοποιεί(gmail,icloud yahoo κλπ) καθώς και την μορφή του email .Ο χρήστης θα λάβει δύο κωδικούς το service ID και το Template ID που εμφανίζονται στο κάτω μέρος όπως φαίνεται στην εικόνα.



Κεφάλαιο 8ο

- Μετά την δημιουργία του template της μορφής δλδ που λάβει ο χρήστης στο mail του πρέπει να τροποποιηθεί το εσωτερικό αρχείο για καλύτερη εμφάνιση του email ο κώδικας γράφεται σε HTML .Η διαδικασία γίνεται κάνοντας κλικ μέσα στο template που δημιουργήσαμε στην συνέχεια κάνουμε κλικ στο edit content και πάμε code editor οπου και γράφουμε τον κώδικα μας



- Ο κώδικας του template ΣΕ μορφή HTML απεικονίζεται στις παρακάτω φωτογραφίες

```
1 <!DOCTYPE html>
29     font-size: 18px;
30     color: #555;
31     margin-bottom: 20px;
32 }
33 .image {
34     width: 100%;
35     max-width: 100%;
36     height: auto;
37     border-radius: 8px;
38 }
39 .footer {
40     text-align: center;
41     font-size: 14px;
42     color: #777;
43     margin-top: 20px;
44 }
45 </style>
46 </head>
47 <body>
48     <div class="container">
49         <div class="title">Βρέθηκε κίνηση στο σαλόνι!</div>
50         <div class="message">Ακολουθεί η εικόνα από το σύστημα ασφαλείας:</div>
51         
52         <div class="footer">Αυτό το μήνυμα στάλθηκε από το σύστημα ασφαλείας του σπιτιού σας.
53     </div>
54 </body>
55 </html>
56
```

Παράρτημα-Πειραματικό μέρος - Επεξήγηση Κώδικα

1: Εκκίνηση αρχείου html

2: Ορισμός αγγλικής σαν γλώσσα επιλογής

3: Αρχή τίτλου mail

6-45: CSS μορφοποίηση κειμένου γραμματοσειράς κλπ χρώατος φόντου κλπ.

46-55: δηλώνεται τιο μήνυμα που θα γράφει το email μας Τελος για την χρήση της emailjs υπηρεσίας πρέπει να κρατήσω τρεις κωδικούς

1. To user ID
2. To Template ID
3. To service ID

Τα δυο πρώτα επεξηγήθηκαν στα παραπάνω βήματα για το user-id πάω στην main interface της emailjs και πατάω accounts στην πρώτη σελίδα το public key code είναι αυτό που θα χρειαστώ.

Οι τρεις κώδικες που θα ενταχθούν στο HTML αρχείο είναι χρήσιμοι για να μπορέσει να σσταλεί το email (ο html κώδικας εξηγείται παρακάτω).

Αναλυτική εγκατάσταση λογισμικού και κώδικα.

Το πρώτο βήμα είναι να κατεβάσουμε ένα πρόγραμμα για να μπορέσεις να τρέξει ο κώδικας. Στην συγκεκριμένη περίπτωση χρησιμοποιήθηκε το intellij.

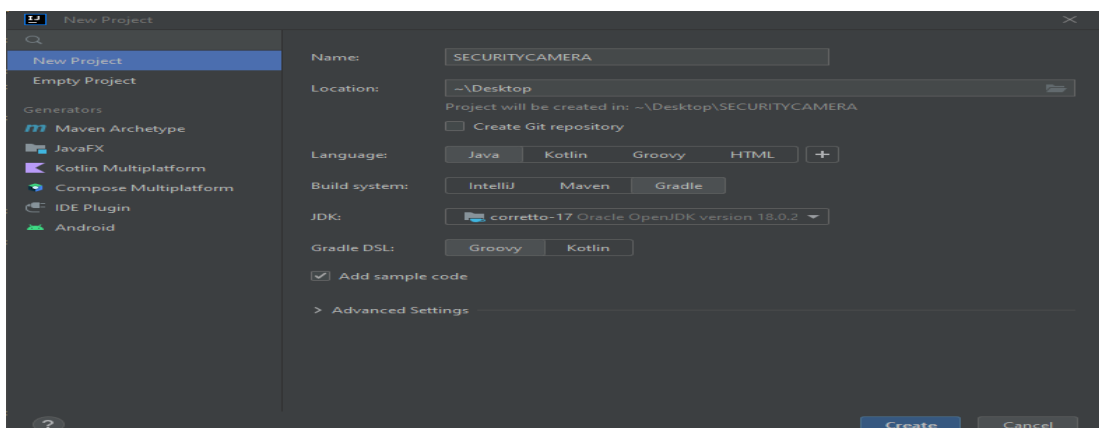
1. Για την εγκατάσταση του προγράμματος μπαίνουμε στο link :

<https://www.jetbrains.com/idea/download/?section=windows>

Κατεβάζουμε το αρχείο ανάλογα με το λογισμικό που χρησιμοποιεί ο υπολογιστή μας.

2. Δημιουργία projects με gradle :

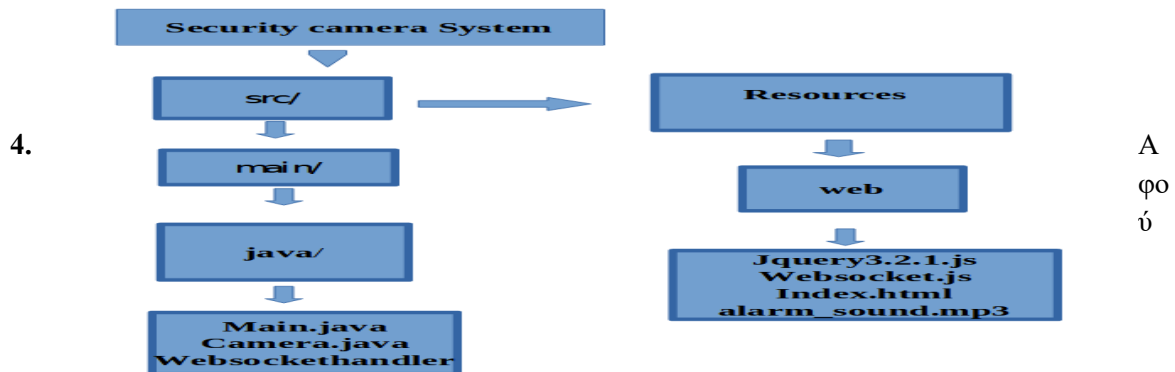
File→ new→ projects και επιλέγουμε την βιβλιοθήκη JDK (oracle openJDK version 18.0.2) όπως φαίνεται στην παρακάτω εικόνα.



3.

Κεφάλαιο 8ο

Έπειτα δημιουργώ την δομή του προγράμματος και τις θέσεις των φακέλων κάνοντας αριστερό κλικ για να δημιουργήσω τις java classes η ακολουθία των φακέλων φαίνεται στο παρακάτω διάγραμμα ροής.



εγκαταστήσουμε το πρόγραμμα στο φάκελο που επιθυμούμε κάνουμε μέσα στον υποφάκελο src ένα ακόμα φάκελο τον web που θα βάλουμε τα αρχεία που θα χρησιμοποιηθούν για την υλοποίηση της σελίδας(html,scripts) καθώς και το αρχείο του ήχου σε μορφή mp3.

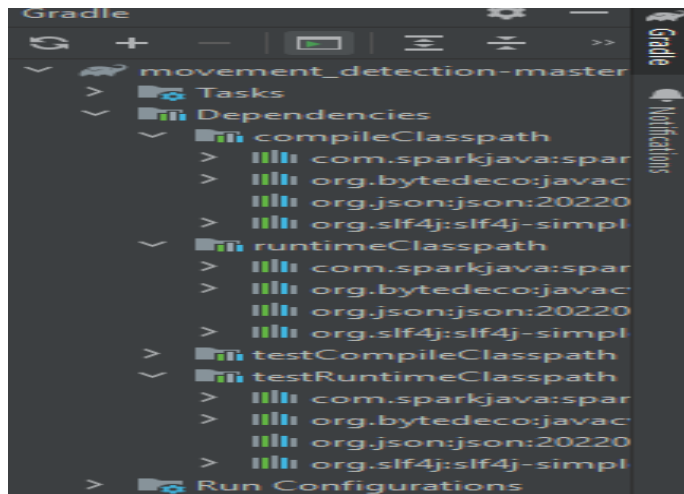
5. Ρύθμιση build.gradle: Ανοίγουμε το gradle αρχείο και προσθέτουμε τις ακόλουθες βιβλιοθήκες:

```
1  plugins {
2      id 'java'
3  }
4
5  group 'org.example'
6  version '1.0-SNAPSHOT'
7
8  repositories {
9      mavenCentral()
10 }
11
12 dependencies {
13     implementation 'org.bytedeco:opencv-platform:4.5.5-1.5.7'
14     implementation 'com.sparkjava:spark-core:2.9.3'
15     implementation 'org.eclipse.jetty.websocket:websocket-server:9.4.44.v20210927'
16     implementation 'javax.mail:mail:1.4.7'
17 }
18
19 test {
20     useJUnitPlatform()
21 }
```

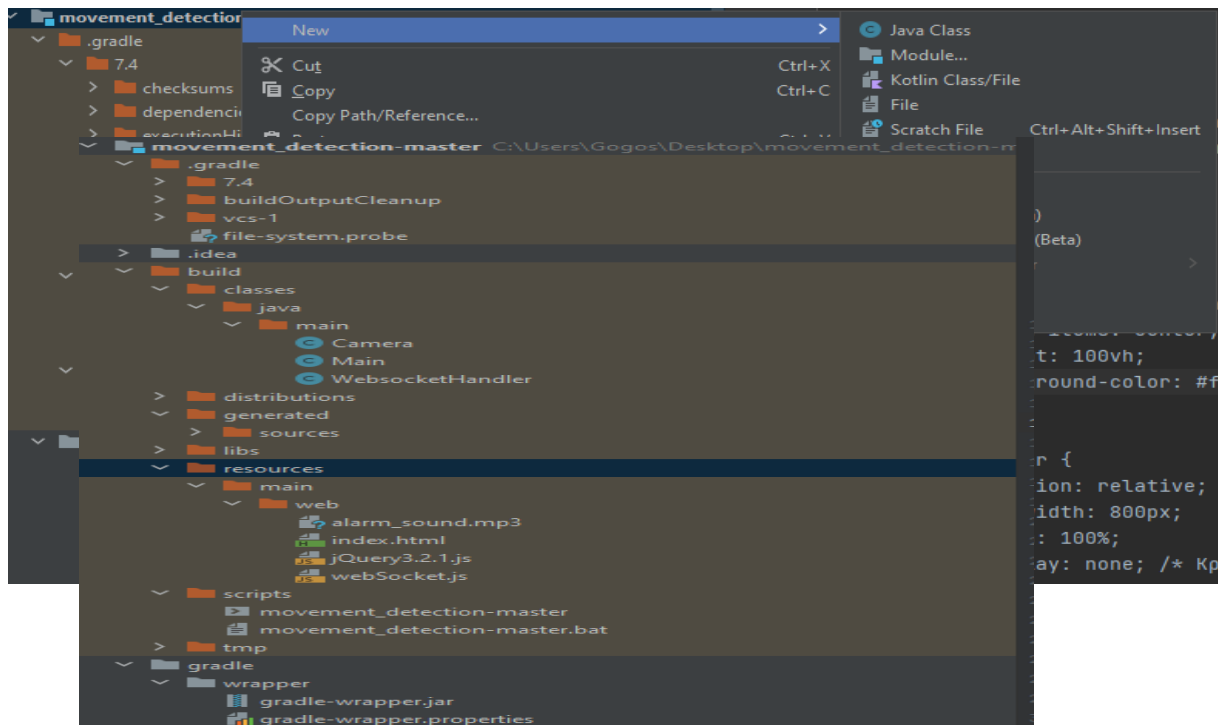
Ανοίγουμε το gradle αρχείο και προσθέτουμε τις ακόλουθες βιβλιοθήκες:

Αφού προσθέσουμε αυτά μέσα στο gradle και κάνουμε κλικ στο εικονίδιο reload all gradle πάνω αριστερά στην φωτογραφία

Παράρτημα-Πειραματικό μέρος - Επεξήγηση Κώδικα



6. Στην συνέχεια δημιουργείται η main και κάνοντας δεξί κλικ πάνω στον φάκελο του προγράμματος μπορούμε να δημιουργήσουμε τα αρχεία html java και js όπως αυτά φαίνονται στις παρακάτω φωτογραφίες



Κεφάλαιο 8ο

7. Τέλος Θα δημιουργήσουμε με την παραπάνω μέθοδο τα παρακάτω προγράμματα σε HTML Javascript και java με τα ονόματα.
8. Και θα δημιουργήσουμε τις ακόλουθες κλάσεις όπου και θα εισάγουμε τον κώδικα μας
 1. Main.java()
 2. WebsocketHandler.java()
 3. Camera.java()
 4. webSocket.js()
 5. index.html()

Παράρτημα Επεξήγησης Κώδικα

Main. Java Code

```
1 import org.bytedeco.javacpp.Loader;
2 import org.bytedeco.opencv.opencv_java;
3 import spark.Spark;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         Spark.port(8080);
9         Loader.load(opencv_java.class);
10        Camera.openCamera();
11        Spark.staticFiles.location( folder: "/web");//
12        Spark.staticFiles.expireTime( seconds: 600);
13        Spark.webSocket( path: "/socket", WebSocketHandler.class);
14        Spark.init();
15    }
16 }
17
```

1-3:Στις γραμμές αυτές του κώδικα εισάγουμε τις βιβλιοθήκες Opencv μέσω του JavaScript της opencv_java κλάσης καθώς και της εισαγωγής του spark framework για την δημιουργία του server.

5:Εναρξη main.

7:Εκκίνηση του java προγράμματος.

8:Δηλώνουμε Την Θέση port που θα τρέξει ο server.

Κεφάλαιο 8ο

9: Φορτώνουμε την βιβλιοθήκη `opencv`

10: Ανοίγει την κάμερα με την μέθοδο `openCamera()`;

11-13: Ορίζει τη διαδρομή των στατικών αρχείων για το Site. Τοποθετεί το `WebSocket` στο μονοπάτι `/socket`, που αντιστοιχεί με την κλάση `WebSocketHandler`.

14: Έναρξη `spark Server`.

`Websockethandler.java` code

Παράρτημα-Πειραματικό μέρος - Επεξήγηση Κώδικα

```
1  import org.eclipse.jetty.websocket.api.Session;
2  import org.eclipse.jetty.websocket.api.annotations.OnWebSocketClose;
3  import org.eclipse.jetty.websocket.api.annotations.OnWebSocketConnect;
4  import org.eclipse.jetty.websocket.api.annotations.WebSocket;
5  import java.io.IOException;
6  import java.nio.ByteBuffer;
7  import java.util.Timer;
8  import java.util.TimerTask;
9
10  1 usage
11  @WebSocket
12  public class WebSocketHandler {
13
14      3 usages
15      private Session session;
16      @OnWebSocketConnect
17      public void connected(Session session){
18          System.out.println("Socket opened");
19          this.session = session;
20          startService(); // We start sending data through the websocket
21      }
22
23      1 usage
24      private void startService() {
25          Timer timer = new Timer( name: "myTimer", isDaemon: true);
26          timer.scheduleAtFixedRate(() -> {
27              try {
28                  broadcastMessage();
29              } catch (IOException e) {
30                  timer.cancel();
31              }
32          }, delay: 10, period: 300);
33      }
34
35      1 usage
36      private void broadcastMessage() throws IOException {
37          session.getRemote().sendString(String.valueOf(Camera.encodeWebcamFrame()));
38      }
39
40      @OnWebSocketClose
```

1-8:Εισάγει τις κλάσεις για τη χρήση WebSocket από το Jetty WebSocket API. Χρησιμοποιούνται κλάσεις όπως Session για τη διαχείριση των συνδέσεων. Καθώς και το timer για την εισαγωγή timer στον κώδικα

10-13:Η σημείωση @WebSocket υποδηλώνει ότι αυτή η κλάση χρησιμοποιείται για να χειριστεί

Κεφάλαιο 8ο

WebSocket επικοινωνία. Μέσα σε αυτήν την κλάση, χρησιμοποιείται η κλάση Session για να κρατάει πληροφορίες σχετικά με τη σύνδεση του WebSocket.

14-19: Η μέθοδος `connected()` καλείται όταν ανοίγει το WebSocket. Εμφανίζει μήνυμα `socket opened` και ξεκινά την υπηρεσία που στέλνει δεδομένα συνεχώς μέσω του `WebSocket -startService()`.

21-36: Η μέθοδος `startService()` ξεκινά έναν timer που στέλνει δεδομένα κάθε 300 ms μέσω του `WebSocket` χρησιμοποιώντας την `broadcastMessage()`.

`BroadcastMessage()`. Αυτή η μέθοδος στέλνει δεδομένα στον client μέσω του `WebSocket`. Χρησιμοποιεί τη `session` που δημιουργήθηκε στη μέθοδο `connected` για να στείλει ένα `string` μήνυμα στον client (πελάτη). Συγκεκριμένα, στέλνει το `string` που παράγεται από τη μέθοδο `Camera.encodeWebcamFrame()`, το οποίο περιέχει την εικόνα της κάμερας και την ένδειξη αν ανιχνεύτηκε κίνηση, σε μορφή `JSON`.

```
38      @OnWebSocketClose
39      //οταν κλεινει το socket san logout
40      public void close(int statusCode, String reason) {
41          System.out.println("Socket closed");
42          this.session = null;
43      }
44  }
45
```

38: Στην γραμμή αυτή του κώδικα με την σημείωση `@onWebSocketClose` δηλώνουμε ότι κλείνει η `websocket` σύνδεση

40: Η μέθοδος `close(int statusCode, String reason)` καλείται αυτόματα όταν η σύνδεση του `WebSocket` κλείνει

41: Εκτυπώνει στο τερματικό ότι το `socket` έκλεισε (`socket closed`)

42: Θέτει τη μεταβλητή `session` σε `null`, πράγμα που σημαίνει ότι πλέον δεν υπάρχει ενεργή σύνδεση για αυτόν τον client (πελάτη).

Camera.java Code

Παράρτημα-Πειραματικό μέρος - Επεξήγηση Κώδικα

```
1 import org.json.JSONObject;
2 import org.opencv.core.*;
3 import org.opencv.imgcodecs.Imgcodecs;
4 import org.opencv.imgproc.Imgproc;
5 import org.opencv.videoio.VideoCapture;
6
7 import java.util.AbstractMap;
8 import java.util.Base64;
9 import java.util.Map;
10
11 public class Camera {
12
13     private static final VideoCapture videoCapture = new VideoCapture();
14
15     /** Opens the computer's default webcam */
16     public static void openCamera() { videoCapture.open(0); }
17
18     /** [We want to show images in HTML which are provided from the Server. ...*/
19     public static JSONObject encodeWebcamFrame() {
20         Mat frame1 = new Mat();
21         Mat frame2 = new Mat();
22         videoCapture.read(frame1);
23         videoCapture.read(frame2);
24         Map.Entry<Mat, Boolean> framePair = detectMotion(frame1, frame2);
25         MatOfByte matOfByte = new MatOfByte();
26         Imgcodecs.imencode("jpg", framePair.getKey(), matOfByte);
27         String frameData = "data:image/jpeg;charset=utf-8;base64," + Base64.getEncoder().encodeToString(matOfByte);
28         JSONObject jsonObject = new JSONObject();
29         jsonObject.put("image", frameData);
30         jsonObject.put("sound", framePair.getValue());
31         return jsonObject;
32     }
33
34     /** Detects movement between two images based on their absolute difference. ...*/
35     public static Map.Entry<Mat, Boolean> detectMotion(Mat first, Mat second) {
```

1-9:Στις γραμμές αυτές του κώδικα έχουμε την εισαγωγή βιβλιοθηκών OpenCV για την επεξεργασία εικόνας το JSONObject για αποστολή δεδομένων σε μορφή Json, και τις υπόλοιπες βιβλιοθήκες για χειρισμό αρχείων εικόνας και Base64 κωδικοποίηση.

11-13:Δημιουργώ μια νέα κλάση την camera και με την μεταβλητή videoCapture δίνω εντολή στον υπολογιστή να κάνει λήψη ακολουθίας εικόνων από την κάμερα.

18:Ανοίγει την κάμερα χρησιμοποιώντας την κλάση VideoCapture το 0 είναι η θέση της κάμερας του υπολογιστή.

34-39:Δημιουργεί δύο καρτέ (frame1 και frame2) που λαμβάνει από την κάμερα και με την μέθοδο detectMotion () συγκρίνει αν υπάρχει διαφορά ανάμεσα στα καρτέ και έτσι να ανιχνεύσουμε κίνηση.

40-42:Κωδικοποιεί το καρτέ που περιέχει κίνηση σε JPEG και το μετατρέπει σε string μορφή Base64.Η μορφή Base64 χρησιμοποιείται για την αποστολή της εικόνας μέσω WebSocket.

Κεφάλαιο 8ο

43-47: Δημιουργεί ένα αντικείμενο JSON που περιέχει την εικόνα και μια ένδειξη αν υπάρχει ανιχνευμένη κίνηση έτσι ώστε να μας δώσει ήχο.

```
56 @ public static Map.Entry<Mat, Boolean> detectMotion(Mat first, Mat second) {
57     Mat painted = new Mat();
58     second.copyTo(painted);
59     //Imgproc.resize(first,first,new Size(300,300));
60     //Imgproc.resize(second,second,new Size(300,300));
61     Imgproc.GaussianBlur(first, first, new Size( width: 15, height: 15), sigmaX: 0); // reducing noise
62     Imgproc.GaussianBlur(second, second, new Size( width: 15, height: 15), sigmaX: 0);
63     Imgproc.cvtColor(first, first, Imgproc.COLOR_RGB2GRAY); // greyscale
64     Imgproc.cvtColor(second, second, Imgproc.COLOR_RGB2GRAY);
65     Mat diff = new Mat();
66     Core.absdiff(first, second, diff);
67     Imgproc.threshold(diff, diff, thresh: 50, maxval: 255, Imgproc.THRESH_BINARY); //binarization
68     Imgproc.dilate(diff, diff, Imgproc.getStructuringElement(Imgproc.MORPH_RECT, new Size( width: 35, height: 35)));
69
70
71     Mat stats = new Mat();
72     int numberOfLabels = Imgproc.connectedComponentsWithStats(diff, new Mat(), stats, new Mat(), connectivity: 8, Cv
73
74     boolean movementDetected = false;
75     // Label 0 is considered to be the background label, so we skip it
76     for (int i = 1; i < numberOfLabels; i++) {
77         if (stats.get(i, col: 4)[0] < 1000) continue; // EMDADON ignore really small detected areas
78         movementDetected = true;
79         // stats columns; [0-4] : [left top width height area]
80
81         // PAINTING
82         Rect textBlock = new Rect(new Point(stats.get(i, col: 0)[0], stats.get(i, col: 1)[0]),
83             new Size(stats.get(i, col: 2)[0], stats.get(i, col: 3)[0]));
84         Imgproc.rectangle(painted, new Point(textBlock.x, textBlock.y),
85             new Point( x: textBlock.x + textBlock.width, y: textBlock.y + textBlock.height), new Scalar( v0: 25
86     }
87     return new AbstractMap.SimpleEntry<>(painted, movementDetected);
88 }
89 }
```

56-66: Εισάγουμε την μέθοδο detectmotion με την οποία συγκρίνουμε τις εικόνες mat first και mat Second. Στην συνέχεια εισάγουμε το GaussianBlur μια μαθηματική συνάρτηση η οποία μας βοηθάει στην απλοποίηση επεξεργασίας της εικόνας. Οι εικόνες στην αρχή θολώνονται και στην συνέχεια μετατρέπονται σε ασπρόμαυρες grayscale. Τέλος η διαφορά τους εάν υπάρχει υπολογίζεται από την Core.absdiff(first,second,diff).

67-74: Με την συνάρτηση Imgproc.threshold ορίζουμε την χρήση κατωφλίου. Το πρώτο όρισμα είναι η εικόνα πηγής, η οποία θα πρέπει να είναι εικόνα σε κλίμακα του γκρι όπως μετατράπηκε πριν. Το

Παράρτημα-Πειραματικό μέρος - Επεξήγηση Κώδικα

δεύτερο όρισμα είναι η τιμή κατωφλίου που χρησιμοποιείται για την ταξινόμηση των τιμών των εικονοστοιχείων. Το τρίτο όρισμα είναι η μέγιστη τιμή που εκχωρείται σε τιμές pixel που υπερβαίνουν το όριο THRESH_BINARY. Ουσιαστικά λοιπόν εφαρμόζουμε ένα κατώφλι στην εικόνα ώστε να γίνει δυαδική ασπρόμαυρη. Χρησιμοποιούμε dilation για να μεγαλώσουμε τις περιοχές κίνησης. Και με την βοήθεια της `connectedComponentsWithstats` μετρά τις περιοχές και υπολογίζει τα στατιστικά τους.

76-88: Ξεκινάμε δημιουργώντας μια `if` η οποία εφαρμόζεται στην περιοχή της οποίας έχουν αναλυθεί τα στατιστικά και αν το μέγεθος αλλαγής είναι κάτω από 1000 pixels τότε η `if` δεν τρέχει. Διαφορετικά εντοπίζεται η ανιχνεύσιμη κίνηση η οποία έχει οριστεί και σχεδιάζει ένα ορθογώνιο γύρω από την περιοχή κίνησης. Τέλος στην γραμμή 87 επιστρέφει την εικόνα με το ορθογώνιο και αν βρέθηκε κίνηση.

Websocketjs

```
1  var websocket;  
2  var mp3_url = './alarm_sound.mp3';  
3  var audio = new Audio(mp3_url);  
4  audio.loop = false;  
5  
6  
7  $(document).ready(function(){  
8      websocket = new WebSocket("ws://" + location.hostname + ":" + location.port + "/socket");  
9      websocket.onmessage = function (msg) {onMessage(msg)};  
10     // above event will be triggered by 'WsHandler.broadcastMessage()'   
11  });  
12  
13  function onMessage(msg) {  
14      var data = JSON.parse(msg.data); // retrieving json data from websocket  
15      $("#frame").attr("src", data.image); // shows the image in index.html  
16      if (data.sound == true){  
17          audio.play();  
18      }  
19  }  
20  
21
```

1: Ορίζουμε μια μεταβλητή για την αποθήκευση του web-socket αντικειμένου.

2: Το `mp3_url` περιέχει τη διαδρομή του αρχείου ήχου ('./alarm_sound.mp3')

3: Δημιουργούμε ένα νέο αρχείο ήχου (`audio`).

4: Ορίζει την ιδιότητα `loop` του `audio` `false` (σε περίπτωση που θέλαμε να παίζει όλη την ώρα ο ήχος θα βάζαμε `true`)

7: Διασφαλίζουμε ότι όλος ο κώδικας μέσα στη συνάρτηση θα εκτελεστεί όταν φορτωθεί πλήρως η σελίδα (το DOM).

8: `websocket = new WebSocket("ws://" + location.hostname + ":" + location.port + "/socket");`:

Κεφάλαιο 8ο

Δημιουργεί μια νέα σύνδεση WebSocket προς το server. Η διεύθυνση του server είναι δυναμική και βασίζεται στο όνομα του hostname και του port της τρέχουσας σελίδας

9: Ορίζει τη συνάρτηση `onMessage()` που θα εκτελείται κάθε φορά που λαμβάνεται ένα μήνυμα από το WebSocket.

13: Αυτή η συνάρτηση ενεργοποιείται κάθε φορά που λαμβάνεται ένα μήνυμα από το WebSocket. Το μήνυμα (`msg`) περιέχει τα δεδομένα που στέλνονται από το server.

14: Μετατρέπει τα δεδομένα από το μήνυμα WebSocket (που είναι σε μορφή string) σε JSON αντικείμενο.

15: Αλλάζει το `src` του HTML στοιχείου με `id frame` στην εικόνα που περιλαμβάνεται στο `data.image`. Αυτό εμφανίζει την εικόνα στη σελίδα.

16: Αν το `data.sound` είναι αληθές.

17: Παίζει τον ήχο συναγερμού (`audio.play()`).

Index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Security Alarm System</title>
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <script src="https://cdn.jsdelivr.net/npm/@emailjs/browser@3/dist/email.min.js"></script>
8   <style>
9     body {
10      margin: 0;
11      font-family: Arial, sans-serif;
12      display: flex;
13      justify-content: center;
14      align-items: center;
15      height: 100vh;
16      background-color: #f0f0f0;
17    }
18    .container {
19      position: relative;
20      max-width: 800px;
21      width: 100%;
22      display: none;
23      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
24      border-radius: 10px;
25      overflow: hidden;
26    }
27    .container img {
28      width: 100%;
29      border-radius: 10px;
30    }
31    .text-block {
32      position: absolute;
33      top: 20px;
34      left: 20px;
35      background-color: rgba(0, 0, 0, 0.6);
36      color: white;
37      padding: 10px 20px;
38      border-radius: 8px;
39    }
40    h4 {
```

- 1: Δηλώνει ότι το έγγραφο θα είναι μορφής html5
- 2: Ορίζουμε σαν κύρια γλώσσα τα αγγλικά
- 3: Ξεκινά το τμήμα κεφαλίδας όπου δηλώνονται ρυθμίσεις, τίτλος, scripts κ.ά.
- 4: Προσθέτουμε το encoding χαρακτήρων utf-8 για την υποστήριξη ελληνικών.

Κεφάλαιο 8ο

5:Κάνει το site responsive σε κινητές συσκευές

6:Τίτλος που εμφανίζεται στο browser

7:Εισάγουμε την Βιβλιοθήκη emailjs για αποστολή αρχείων email.

8-53:έναρξη CSS μορφοποίησης Διαμόρφωση στυλ σελίδας (κεντράρει το φόντο γκριζάρει, κάνει το background της σελίδας στρογγύλεμα γωνιών εφαρμόζει σκίαση και άλλα.)

```
51 <body>
52
53 <script src="jquery3.2.1.js"></script>
54
55 <div class="container">
56 <img id="frame" src="" alt="Live Feed">
57 <div class="text-block">
58 <h4>Security Alarm System</h4>
59 <p>Live view & real time image capture</p>
60 </div>
61 </div>
62
63 <script>
64 // Αρχικοποίηση EmailJS
65 emailjs.init("lrCFbRNezw4pS1Q6D"); // Αντικατάστησε με το δικό σου PUBLIC KEY
66
67 const frameImg = document.getElementById("frame");
68 let sentRecently = false;
69
70 const ws = new WebSocket("ws://" + location.hostname + ":8080/socket");
71
72 ws.onmessage = function(event) {
73   const data = JSON.parse(event.data);
74   const base64Image = data.image;
75
76   // Εμφάνιση live εικόνας
77   frameImg.src = base64Image;
78
79   // Συμπίεση της εικόνας πριν την αποστολή
80   const compressedImage = resizeImage(base64Image);
81
82   // Αποστολή email μία φορά ανά 30''
83   if (!sentRecently) {
84     sentRecently = true;
85
86     emailjs.send("service_x7ezqyf", "template_ech0tz9", {
87       message: "Ανιχνεύθηκε εικόνα από το σύστημα ασφαλείας.",
88       image_base64: compressedImage
89     }).then(function(response) {
90       console.log("📧 Email sent successfully:", response.status);
91     });
92   }
93 }
94 </script>
95 </body>
```

55-61: Δημιουργείται το container που περιέχει τον τίτλο του συστήματος που θα φαίνεται στην σελίδα

63: Τέλος του head

Κεφάλαιο 8ο

```
104     document.querySelector('.container').style.display = 'block';
105   } else {
106     alert("wrong password!");
107     window.location.href = "https://www.google.com";
108   }
109 });
110
111 // Συνάρτηση για συμπίεση εικόνας
112 function resizeImage(base64Image) {
113   const img = new Image();
114   img.src = base64Image;
115
116   return new Promise((resolve, reject) => {
117     img.onload = function() {
118       const canvas = document.createElement('canvas');
119       const ctx = canvas.getContext('2d');
120
121       // Επέλεξε το μέγεθος που θέλεις να έχει η εικόνα
122       const maxWidth = 800; // Μέγιστο πλάτος εικόνας
123       const scaleFactor = maxWidth / img.width;
124       canvas.width = maxWidth;
125       canvas.height = img.height * scaleFactor;
126
127       ctx.drawImage(img, 0, 0, canvas.width, canvas.height);
128
129       // Επιστροφή του νέου base64 string της συμπιεσμένης εικόνας
130       const resizedImage = canvas.toDataURL("image/jpeg", 0.7); // Χρησιμοποιώ
131       resolve(resizedImage);
132     };
133     img.onerror = function(error) {
134       reject(error);
135     };
136   });
137 }
138 </script>
139
140
141
142
143 </body>
html > body > script
```

69: Ανοίγει μέσω websocket την πύλη **192.168.1.5:8080/socket**

72-97 :εισάγει την μορφή websocket

82-99:Χρήση της υπηρεσίας emailjs και αποστολή email σε περίπτωση ανίχνευσης κίνησης.

99-108: Εισαγωγή Κωδικού πρόσβασης

112-137: Συμπίεση αρχείου εικόνας που λήφθηκε για ευκολότερη αποστολή της εικόνας μέσω mail.

143 :Τέλος html