

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

ΕΞΥΠΝΟΣ ΑΝΕΛΚΥΣΤΗΡΑΣ ΜΕ ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΣΩΠΩΝ ΚΑΙ ΟΜΙΛΙΑΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αστέριος Σελήνης ΤΠ5058

Επιβλέπων: Σπυρίδων Παναγιωτάκης

ΗΡΑΚΛΕΙΟ 2024

Περίληψη

Στην εποχή μας, οι τεχνολογίες της Τεχνητής Νοημοσύνης (AI) και του Διαδικτύου των Πραγμάτων (IoT) ενσωματώνονται ολοένα και περισσότερο στην καθημερινότητα, με στόχο τη βελτίωσή της. Οι ανελκυστήρες, δεν αποτελούν εξαίρεση σε αυτήν την τάση. Με την εξέλιξη της τεχνολογίας, οι ανελκυστήρες δεν είναι πλέον απλά μέσα μεταφοράς ανάμεσα σε ορόφους, αλλά μπορούν να μετατραπούν σε πολύπλοκα συστήματα που ενισχύουν την εμπειρία των χρηστών. Η ενσωμάτωση των τεχνολογιών της αναγνώρισης προσώπων και ομιλίας στους ανελκυστήρες δίνει πολλές δυνατότητες για την καινοτομική περαιτέρω ανάπτυξή τους.

Η παρούσα εργασία εστιάζει στην ενσωμάτωση προηγμένων τεχνολογιών στον σχεδιασμό και τη λειτουργία των ανελκυστήρων. Αναλύονται τα κύρια συστήματα και οι τεχνολογίες που χρησιμοποιήθηκαν, όπως το Διαδίκτυο των Πραγμάτων (IoT), η αναγνώριση προσώπων, η αναγνώριση ομιλίας, η ψηφιακή αναπαραγωγή φωνής, καθώς και ο τρόπος με τον οποίο αυτές οι τεχνολογίες ενσωματώθηκαν στο σύστημα του έξυπνου ανελκυστήρα. Το συνολικό σύστημα έχει αναπτυχθεί υπό τη μορφή διαδικτυακής εφαρμογής, χρησιμοποιώντας αποκλειστικά αντίστοιχες τεχνολογίες του διαδικτύου.

Η απόφαση να αναπτυχθεί ένας έξυπνος ανελκυστήρας αποκλειστικά μέσω διαδικτυακών εφαρμογών πάρθηκε, αφενός για να επωφεληθεί από τη χρήση διαθέσιμων ανοικτών APIs για αναγνώριση ομιλίας, προσώπων, text to speech, κλπ, και αφετέρου για να αξιολογηθεί το τελικό αποτέλεσμα ως προς τη λειτουργικότητά του. Τα APIs προσφέρουν ευκολία στην τροποποίηση και ενσωμάτωση διαφόρων τεχνολογιών, προσαρμόζοντας την εφαρμογή σε διαφορετικές ανάγκες και συνθήκες χρήσης.

Επιπρόσθετα, στην πτυχιακή τονίζεται η σημασία της αυτοματοποίησης για τη βελτίωση της εμπειρίας του χρήστη, καθώς και οι προκλήσεις και τα οφέλη που συνεπάγονται από την εφαρμογή αυτών των τεχνολογιών στο τομέα των ανελκυστήρων.

Abstract

In our era, the technologies of Artificial Intelligence (AI) and the Internet of Things (IoT) are increasingly integrated into everyday life, with the aim of improving it. Elevators are no exception to this trend. With the development of technology, elevators are no longer simple means of transportation between floors, but can be transformed into complex systems that enhance the user experience. The integration of face and speech recognition technologies in elevators gives many possibilities for their innovative further development.

This work focuses on the integration of advanced technologies in the design and operation of elevators. The main systems and technologies used, such as the Internet of Things (IoT), face recognition, speech recognition, interactive voice response, and how these technologies were integrated into the smart elevator system are analyzed. The overall system has been developed in the form of a web application, using exclusively corresponding web technologies.

The decision to develop a smart elevator exclusively through web applications was made, on the one hand, to benefit from the use of available open APIs for speech and face recognition, text to speech, etc., and on the other hand, to evaluate the result in terms of its functionality. APIs offer ease of modification and integration of various technologies, adapting the application to different needs and usage conditions.

Additionally, the thesis highlights the importance of automation to improve the user experience, as well as the challenges and benefits of implementing these technologies in the elevator sector.

Πίνακας Περιεχομένων

Περίληψη....2
Abstract....3
Εικόνες....5
Κεφάλαιο 1 - Εισαγωγή....6
1.1 Κίνητρα....6
1.2 Αντικείμενο της Πτυχιακής....10
1.3 Δομή της Εργασίας....12
Κεφάλαιο 2 – Συναφείς Τεχνολογίες....13
2.1 Διαδίκτυο των Πραγμάτων (IoT)....13
2.2 Έξυπνοι Ανελκυστήρες....17
2.3 Αναγνώριση Προσώπου....20
2.4 Αναγνώριση Ομιλίας....22
2.5 Διαδίκτυο των Πραγμάτων (IoT) και Web....24
Κεφάλαιο 3 - Σχεδιασμός Συστήματος Έξυπνου Ανελκυστήρα....26
3.1 Ανάλυση απαιτήσεων....26
3.2 Κατηγορίες χρηστών....27
3.3 Περιπτώσεις Χρήσης Συστήματος....28
3.4 Περιβάλλον λειτουργίας....28
3.5 Σχεδιαστική Προσέγγιση....29
Κεφάλαιο 4 - Υλοποίηση της Διαδικτυακής Εφαρμογής του Έξυπνου Ανελκυστήρα....31
4.1 Τεχνολογίες που Χρησιμοποιήθηκαν....31
4.2 Δομή Φακέλων....32
4.3 Ενσωμάτωση του Teachable Machine....32
4.4 Ενσωμάτωση του Web Speech Api....34
4.5 Ενσωμάτωση του Responsive Voice....34
4.6 Εξήγηση του Κώδικα Υλοποίησης....34
Κεφάλαιο 5 - Χρήση του Συστήματος....53
5.1 Ιστοσελίδα χρήστη....53
5.2 Ιστοσελίδα του Διαχειριστή....57
Κεφάλαιο 6 - Αξιολόγηση Συστήματος....64
Κεφάλαιο 7 - Συμπεράσματα και Μελλοντικές Επεκτάσεις....65
Βιβλιογραφία....66

Εικόνες

Εικόνα 1 Facial Recognition Market - Grand View Reasearch 7
Εικόνα 2 Facial Recognition - Market Research Future 8
Εικόνα 3 Voice & Speech Recognition Market - Grand View Research 8
Εικόνα 4 Voice & Speech Recognition Market - Fortune Business Insights 9
Εικόνα 5 Four Stage IoT Architecture - InterviewBit 13
Εικόνα 6 Σχεδιαστική Προσέγγιση 29
Εικόνα 7 Teachable Machine train Image Model 33
Εικόνα 8 Teachable Machine Export Model 33
Εικόνα 9 Αρχική οθόνη του συστήματος 53
Εικόνα 10 Κατάσταση του ανελκυστήρα 54
Εικόνα 11 Όροφος του ανελκυστήρα 54
Εικόνα 12 Φωνητική εντολή 55
Εικόνα 13 Ενημερωτικές Πληροφορίες 55
Εικόνα 14 Πίνακας ενοίκων 56
Εικόνα 15 Αναγνώριση Προσώπου 57
Εικόνα 16 Σύνδεση 57
Εικόνα 17 Admin Logout 58
Εικόνα 18 Floor Data 58
Εικόνα 19 Add Floor 59
Εικόνα 20 Success message 59
Εικόνα 21 Failed Message 60
Εικόνα 22 Edit Floor 60
Εικόνα 23 Delete Floor 61
Εικόνα 24 Tenants Data 61
Εικόνα 25 Add Tenant 62
Εικόνα 26 Edit Tenant 62
Εικόνα 27 Delete Tenant 63

Κεφάλαιο 1 - Εισαγωγή

Η εισαγωγή της πτυχιακής θέτει τις βάσεις για την ανάλυση του επιλεγμένου θέματος. Εξετάζοντας τα κίνητρα, το αντικείμενο της πτυχιακής και την δομή της εργασίας.

1.1 Κίνητρα

Ζούμε σε μια εποχή όπου παρακολουθούμε μια νέα επανάσταση στη βιομηχανία, γνωστή ως Industry 4.0 (Βιομηχανία 4). Το Industry 4.0 επηρεάζει σημαντικά τα επιχειρηματικά μοντέλα, αλλάζοντας τον τρόπο παραγωγής, βελτιστοποίησης και διανομής προϊόντων από τις εταιρίες. Αυτό το καθιστά ένα ζήτημα που χρειάζεται ιδιαίτερη προσοχή, καθώς επηρεάζει την οικονομία, την κοινωνική ανάπτυξη αλλά και το περιβάλλον.

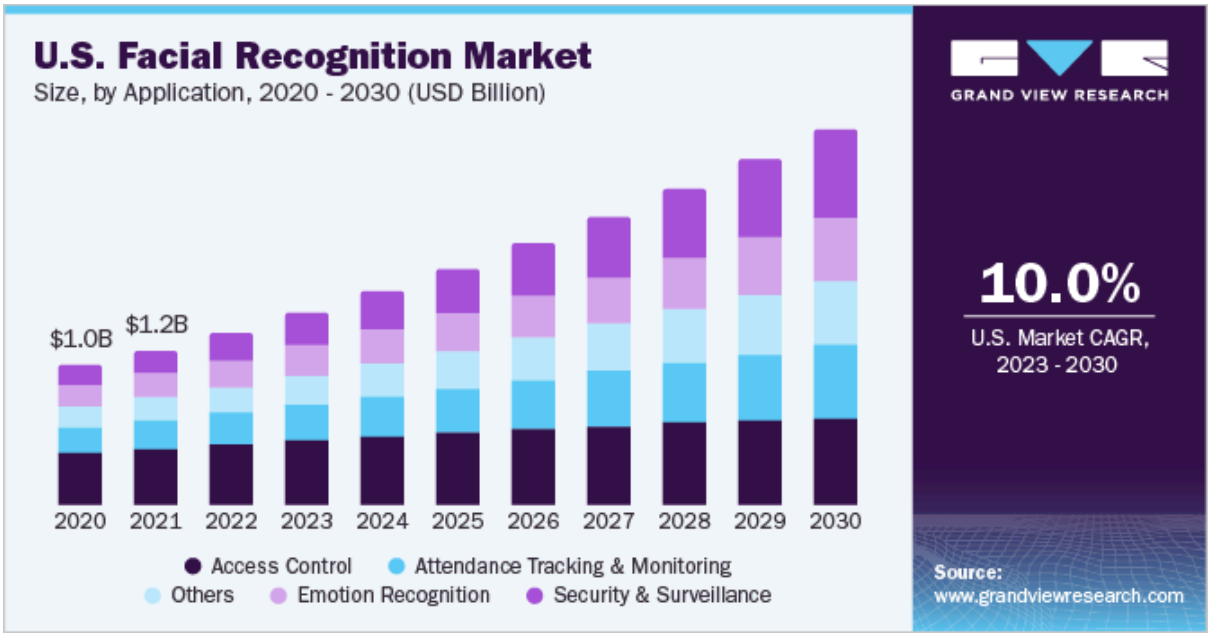
Οι εταιρίες εξελίσσονται και προσπαθούν να προσαρμόσουν τις τρέχουσες τεχνολογίες που έχουν να κάνουν με αυξημένη τάση προς την αυτοματοποίηση και την ανταλλαγή δεδομένων. Κάποιες από αυτές τις τεχνολογίες είναι το «Διαδίκτυο των Πραγμάτων» (Internet of things - IoT), το «Βιομηχανικό Διαδίκτυο των Πραγμάτων» (Industrial internet of things - IIoT), τα «Κυβερνοφυσικά Συστήματα» (Cyber-physical systems - CPS), την «Έξυπνή Κατασκευή» (Smart manufacture), τα «Έξυπνα Εργοστάσια» (Smart factories), το «Υπολογιστικό Νέφος» (Cloud computing), τη «Γνωστική Υπολογιστών» (Cognitive computing), την «Τεχνική Νοημοσύνη» (Artificial intelligence - AI). Για να επιτύχουν τις βέλτιστες αποφάσεις τους, οι εταιρίες εξοπλίζονται με υλικό και λογισμικό που συλλέγουν και αναλύουν δεδομένα. Αυτή η τάση έχει ως αποτέλεσμα την ανάδειξη νέων τομέων.

Τα έξυπνα εργοστάσια χρησιμοποιούν υψηλές τεχνολογικές συσκευές IoT για τη μεγιστοποίηση της παραγωγικότητας και τη βελτίωση της ποιότητας. Η αντικατάσταση των χειροκίνητων εργασιών επιθεώρησης με τεχνικές που βασίζονται στην τεχνητή νοημοσύνη μειώνει τα σφάλματα παραγωγής και εξοικονομεί χρήματα και χρόνο. Το προσωπικό συντήρησης μπορεί να έχει ένα έξυπνο κινητό συνδεδεμένο στο νέφος με την δυνατότητα να παρακολουθεί τις διαδικασίες από οπουδήποτε. Επιπλέον, με την βοήθεια των αλγόριθμων μηχανικής μάθησης, οι κατασκευαστές μπορούν να εντοπίσουν τα σφάλματα αμέσως, εξοικονομώντας τους χρήματα και χρόνο.

Μία από τις σημαντικές τεχνολογίες του Industry 4.0 είναι η αναγνώριση ομιλίας και η αναγνώριση του προσώπου. Αυτό μπορούμε να το διακρίνουμε από διάφορες πηγές που

πιστεύουν την επέκταση και την ανάπτυξη αυτών τον δύο τεχνολογιών στην αγορά και τονίζουν την αύξηση της ζήτησης τους σε ποικίλες εφαρμογές.

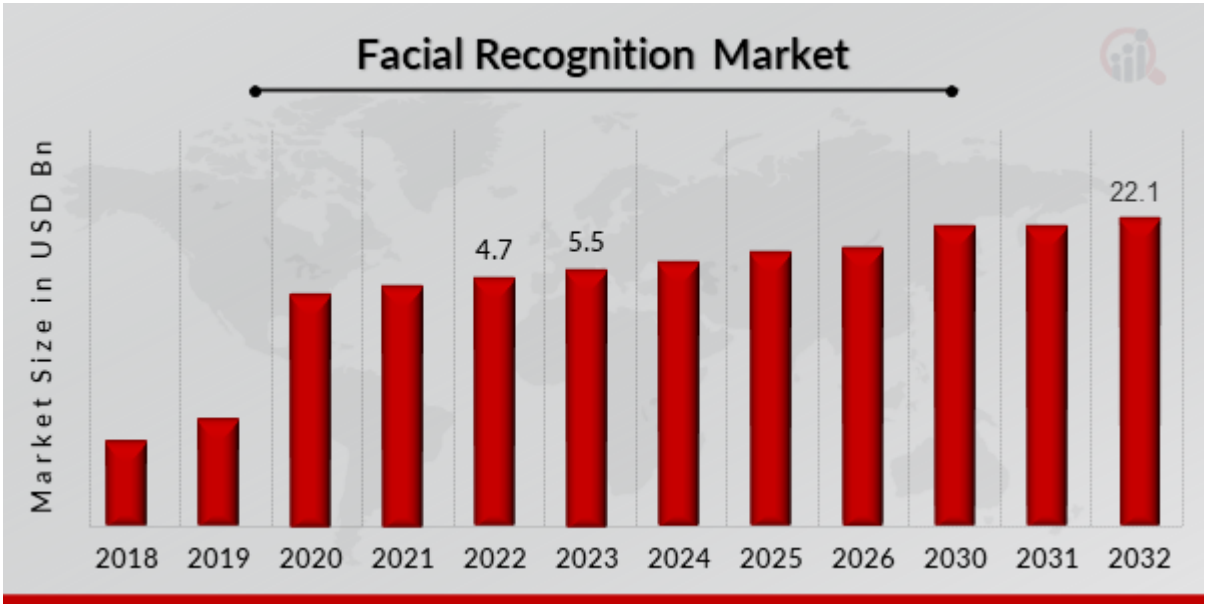
Η Grand View Research υπογραμμίζει την ταχεία ανάπτυξη της αγοράς αναγνώρισης προσώπου. Δείχνει ότι η αγορά είχε αξία 5,15 δισεκατομμύρια δολάρια το 2022. Αναμένεται να αυξηθεί με ρυθμό 14,9% ετησίως από το 2023 έως το 2030, φτάνοντας τα 15,84 δισεκατομμύρια δολάρια. Η αύξηση αυτή οφείλεται στην αυξημένη χρήση της τεχνολογίας σε τομείς όπως ασφάλεια, υγεία, λιανική πώληση και ηλεκτρονικά είδη.



Εικονα 1 Facial Recognition Market - Grand View Reasearch

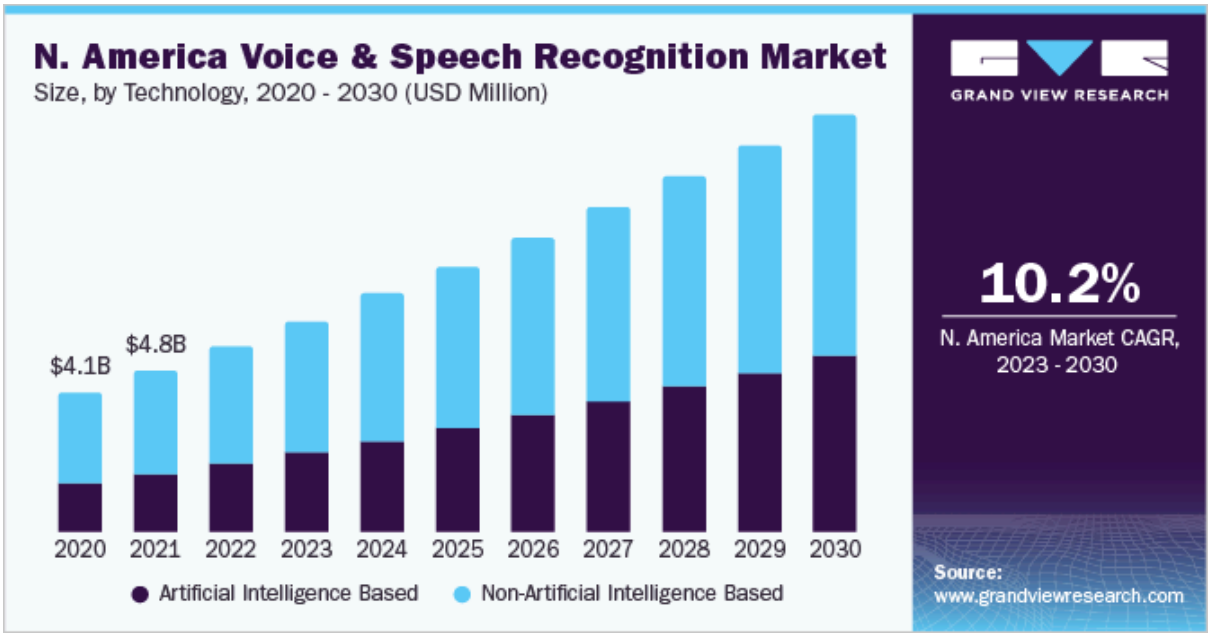
Η Mordor Intelligence τονίζει τις σημαντικές εξελίξεις στην αγορά αναγνώρισης προσώπου, με έμφαση στην αυξανόμενη σημασία της τεχνητής νοημοσύνης και της μηχανικής μάθησης για τη βελτίωση αυτών των τεχνολογιών. Η αγορά εκτιμάται στα 5,69 δισεκατομμύρια δολάρια το 2023 και αναμένεται να φτάσει τα 12,05 δισεκατομμύρια δολάρια έως το 2028 με ετήσια αύξηση 16,20%. Η Mordor Intelligence προτείνει διάφορες εφαρμογές, όπως ασφάλεια έως προσωποποιημένες εμπειρίες πελατών, και τονίζει την αυξανόμενη ένταξη της αναγνώρισης προσώπου σε διάφορους τομείς.

Η Market Research Future προβλέπει σημαντική ανάπτυξη για την αγορά της αναγνώρισης προσώπου με CARG 18,80 από το 2022 έως το 2032. Αποδίδει αυτή την ανάπτυξη στις αυξημένες κρατικές επενδύσεις σε αυτές τις τεχνολογίες και στην αυξημένη ζήτηση για τα προηγμένα συστήματα παρακολούθησης βίντεο.



Εικόνα 2 Facial Recognition - Market Research Future

Η Grand View Research τονίζει ότι η αγορά της αναγνώρισης φωνής αναδεικνύει μια σημαντική ανάπτυξη στον τομέα. Η παγκόσμια αγορά της αναγνώρισης φωνής και ομιλίας έχει μια αξία 17,17 δισεκατομμύρια δολάρια το 2022 και αναμένεται να αναπτυχθεί με ετήσιο ρυθμό αύξησης 14,9% από το 2023 έως το 2030. Η εντυπωσιακή αυτή ανάπτυξη οφείλεται στην ενίσχυση των εφαρμογών της αναγνώρισης φωνής σε διάφορους τομείς, όπως η υγεία, το αυτοκίνητο, οι καταναλωτικές συσκευές και η εκπαίδευση.

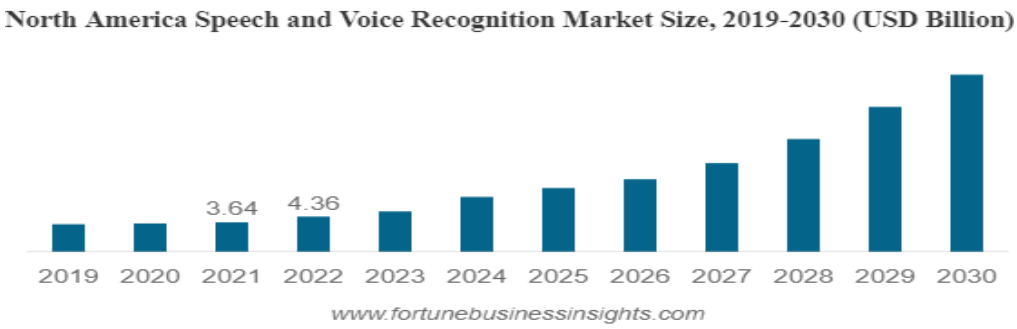


Εικόνα 3 Voice & Speech Recognition Market - Grand View Research

Η αναγνώριση προσώπου αποτελεί μια τεχνολογία που επιτρέπει τον εντοπισμό και την αναγνώριση του προσώπου ενός ατόμου μέσω μιας φωτογραφίας, ενός βίντεο που έχει καταγραφεί ή ακόμη και σε πραγματικό χρόνο. Αυτή η τεχνολογία χρησιμοποιείται για διάφορος λόγους, όπου ένας είναι η αποτελεσματική ασφάλεια σε κτίρια και κινητά. Η αναγνώριση του προσώπου σε αυτές τις περιπτώσεις εκτός από ασφάλεια προσφέρει μια βολική και γρήγορη εναλλακτική από τις άλλες βιομετρικές τεχνολογίες. Άλλη χρήση του είναι σε νοσοκομεία δίνοντας την δυνατότητα στους γιατρούς να έχουν πρόσβαση στα αρχεία του ασθενή μέσα από την αναγνώριση προσώπου. Ακόμη, μπορεί να χρησιμοποιηθεί στα συστήματα τις τράπεζας προσφέροντας άλλο ένα επίπεδο ασφάλειας και ταυτοποιήσεις. Μια, ανάληψη ποσού εκτός από την χρήση του μυστικού κωδικού μπορεί να έχει και την αναγνώριση προσώπου.

Η Markets And Markets υποστηρίζει ότι η αγορά της αναγνώρισης ομιλίας και φωνής προβλέπει σημαντική ανάπτυξη από το 2022 έως το 2027, με το μέγεθος της αγοράς να αναμένεται να αυξηθεί από 9,4 δισεκατομμύρια δολάρια σε 28,1 δισεκατομμύρια δολάρια. Αυτή η αυξημένη ζήτηση οφείλεται στην υγειονομική περίθαλψη όπου χρειάζεται την αποτελεσματικότητα και στην αυξημένη χρήση των έξυπνων συσκευών. Κορυφαίες εταιρείες στον συγκεκριμένο τομέα συμπεριλαμβάνουν την Apple, την Microsoft, την IBM, την Alphabet, την Amazon, την Baidu και την IFytek.

Η Fortune Business Insights υποστηρίζει ότι η αγορά της Αναγνώρισης Ομιλίας και Φωνής προβλέπει σημαντική ανάπτυξη από το 2023 έως το 2030. Η αγορά αποτιμάται σε 10,42 δισεκατομμύρια δολάρια το 2022 και αναμένεται να φτάσει τα 59,62 δισεκατομμύρια δολάρια έως το 2030, αυξάνοντας με CAGR 24,8%. Αυτή η ανάπτυξη οφείλεται στις εξελίξεις σε τεχνολογίες όπως η Επεξεργασία Φυσικής Γλώσσας (NLP), τη Μηχανική Εκμάθηση (ML) και την Αυτοματοποιημένη Αναγνώριση Ομιλίας (ASR). Ακόμη, οι τάσεις δείχνουν την ενσωμάτωση της Τεχνητής Νοημοσύνης και της Μηχανική Μάθησης σε διάφορους κλάδους.



Εικόνα 4 Voice & Speech Recognition Market - Fortune Business Insights

Η αναγνώριση ομιλίας είναι η τεχνολογία που έχει την δυνατότητα να ερμηνεύει, να κατανοεί την ανθρώπινη ομιλία. Εκτός, από το να κατανοεί και να ερμηνεύει μπορεί και να εκτελεί φωνητικές εντολές. Η αναγνώριση της ομιλίας έχει αναπτυχθεί αρκετά με την άνοδο της τεχνικής νοημοσύνης. Η πιο γνωστή χρήση του είναι ο φωνητικός βοηθός και κάποιες από τις πιο γνωστές εφαρμογές είναι η Alexa της Amazon , η Siri της Apple και η Google Assistant. Η αναγνώριση ομιλίας προσφέρει στον άνθρωπο κάνει πολλές εργασίες ταυτόχρονα και εξοικονομεί χρόνο. Βελτιώνει για όλους την προσβασιμότητα στην τεχνολογία. Είναι ένα ζωντανό παράδειγμα που η τεχνολογία μας κάνει πιο εύκολη και αποτελεσματική την καθημερινότητα. Αυτή η τεχνολογία είναι προσβάσιμη και για άτομα με ειδικές ανάγκες όπως περιορισμένη κινητικότητα ή όραση. Οι φωνητικοί βοηθοί και η αναγνώριση ομιλίας καθιστούν δυνατή την αλληλεπίδραση τους με τις εφαρμογές χωρίς την ανάγκη για χρήση του ποντικιού, πληκτρολογίου ή οθόνη αφής.

1.2 Αντικείμενο της Πτυχιακής

Η πρόοδος στον τομέα της τεχνολογίας και η αυξανόμενη αστικοποίηση έχουν σημαντική επίδραση στην αύξηση της χρήσης των ανελκυστήρων. Αυτό φαίνεται καθημερινά στη ζωή μας, καθώς τα πολυώροφα κτίρια απαιτούν συχνά τη χρήση του ανελκυστήρα ως μέσο μετακίνησης. Οι εξελίξεις στην τεχνολογία του Διαδίκτυού των Πραγμάτων (IoT) επιτρέπει τους ανελκυστήρες να γίνουν πιο έξυπνοι βελτιώνοντας τις λειτουργίες και το πρωτόκολλα τους.

Η παγκόσμια αγορά εκτιμάει ότι η ζήτηση για έξυπνους ανελκυστήρες που χρησιμοποιούν την τεχνολογία Διαδίκτυού των Πραγμάτων (IoT) αναμένεται να αυξηθεί. Κάποιες από τις βασικές εταιρίες που δουλεύουν στην ανάπτυξη της είναι η Fujitec Co, η Hitachi Ltd, η Toshiba Elevators, η Schindler Group, η ThyssenKrupp AG, η Kone

Corporation, η Otis Elevator Company, η Hyundai Elevator Co. Ltd και η Mitsubishi Electric Corporation.

Οι έξυπνοι ανελκυστήρες φέρνουν στην αγορά αρκετά οφέλη. Η ασφάλεια είναι ένα από αυτά. Ο ανελκυστήρας μπορεί να χρησιμοποιήσει βιομετρικές τεχνολογίες όπως αναγνώριση δαχτυλικού αποτυπώματος και αναγνώριση προσώπου. Ακόμη μπορεί να χρησιμοποιεί έξυπνες κάρτες που θα παρέχουν πρόσβαση στα εξουσιοδοτημένα άτομα. Άλλο ένα όφελος, είναι ότι δίνει την δυνατότητα να πλοηγηθούν στους ορόφους μόνο με φωνητικές εντολές και χωρίς να ακουμπήσεις κουμπιά ή οθόνη, που το καθιστά προσβάσιμο για όλους τους ανθρώπους.

Το αντικείμενο της πτυχιακής είναι η κατασκευή ενός ανελκυστήρα που θα ενσωματώνει τεχνολογίες όπως Internet Of Things (IoT) και μέσα σε αυτό το πλαίσιο θα εντάξουμε την αναγνώριση ομιλίας και του προσώπου.

Ο κύριος σκοπός την πτυχιακής είναι να αξιοποιηθούν οι τεχνολογίες του διαδικτύου και να εξεταστεί η εφικτότητα της χρήσης έτοιμων λειτουργιών που προσφέρονται μέσω αντίστοιχων ανοικτών διεπαφών API. Στόχος είναι η αξιολόγηση της αποδοτικότητας των υπάρχοντων εργαλείων, η εξέταση των περιορισμών τους, με σκοπό να κατανοηθεί πόσο πρακτικές και χρήσιμες μπορούν να αποδειχθούν σε πραγματικές εφαρμογές.

Ζητούμενο της πτυχιακής είναι να κατασκευαστεί μια διαδικτυακή εφαρμογή ώστε μέσω μιας ιστοσελίδας να γίνεται αναγνώριση του προσώπου ενός ενοίκου όταν μπαίνει στον ανελκυστήρα. Να γίνεται αναγνώριση ομιλίας, όταν μιλάει ο χρήστης, ώστε να έχει την δυνατότητα να μεταφερθεί στους ορόφους μόνο με φωνητικές εντολές. Επίσης, να υπάρχει φωνητική απάντηση από το σύστημα για να αλληλοεπιδρά με τους χρήστες σε τρία βασικά σημεία της λειτουργίας: α) κατά την είσοδο του χρήστη στον ανελκυστήρα, για να τον καλωσορίζει, β) πριν ξεκινήσει ο ανελκυστήρας για τον προορισμό του, για να τον ενημερώνει, και γ) κατά την άφιξη στον προορισμό. Η αναγνώριση ομιλίας και οι φωνητικές απαντήσεις, εμφανίζονται σε οθόνη για να τις βλέπει ο χρήστης και όχι μόνο να τις ακούει. Παράλληλα, ο ανελκυστήρας εμφανίζει στην οθόνη του διάφορες ενημερωτικές πληροφορίες όπως ο καιρός, αλλά και που βρίσκεται ο ανελκυστήρας και που κατευθύνεται, ώστε οι χρήστες να έχουν συνεχή ενημέρωση για την διαδρομή τους. Τέλος, στην οθόνη υπάρχει δυνατότητα να εμφανίζονται πληροφορίες για τους ένοικους που έχουν δώσει τα στοιχεία τους, ώστε οι χρήστες να μπορούν εύκολα να βρίσκουν τον όροφο που θέλουν να πάνε.

Ακόμη ζητούμενο είναι να υπάρχει η ιστοσελίδα του διαχειριστή, για την οποία υπάρχει επιλογή σύνδεσης και αποσύνδεσης. Ο διαχειριστής θα μπορεί να κάνει προσθήκη, επεξεργασία, και διαγραφή τον αριθμό των ορόφων που θα υποστηρίζει ο ανελκυστήρας αλλά και τις λέξεις που συνδέονται με αυτόν τον όροφο. Για παράδειγμα, άμα ο χρήστης πει πρώτος όροφος, το σύστημα θα αντιστοιχήσει τη λέξη με τον πρώτο όροφο και θα ξεκινήσει η μετακίνηση. Ακόμη, θα μπορεί να κάνει προσθήκη, επεξεργασία και διαγραφή των πληροφοριών που έχουν δώσει οι ένοικοι.

1.3 Δομή της Εργασίας

Η πτυχιακή αυτή αναλύεται σε εφτά κεφάλαια τα οποία είναι τα ακόλουθα:

- Κεφάλαιο 1: είναι η εισαγωγή που αναλύει τα κίνητρα τις πτυχιακής και μας περιγράφει τις βασικές ιδέες που υλοποιήθηκαν.
- Κεφάλαιο 2: παρουσιάζονται οι σχετικές τεχνολογίες που χρησιμοποιήθηκαν σε αυτή την πτυχιακή.
- Κεφάλαιο 3: είναι το κεφάλαιο του σχεδιασμού που αναλύει τις απαιτήσεις και την σχεδίαση προσέγγιση του συστήματος.
- Κεφάλαιο 4: περιγράφεται η υλοποίηση του συστήματος, οι τεχνολογίες που χρησιμοποιήθηκαν και ο κώδικας υλοποίησης.
- Κεφάλαιο 5: παρουσιάζονται σενάρια χρήσης του συστήματος.
- Κεφάλαιο 6: είναι το κεφάλαιο που γίνεται η αξιολόγηση του συστήματος
- Κεφάλαιο 7: είναι το κεφάλαιο που αναλύονται τα συμπεράσματα και οι μελλοντικές επεκτάσεις.

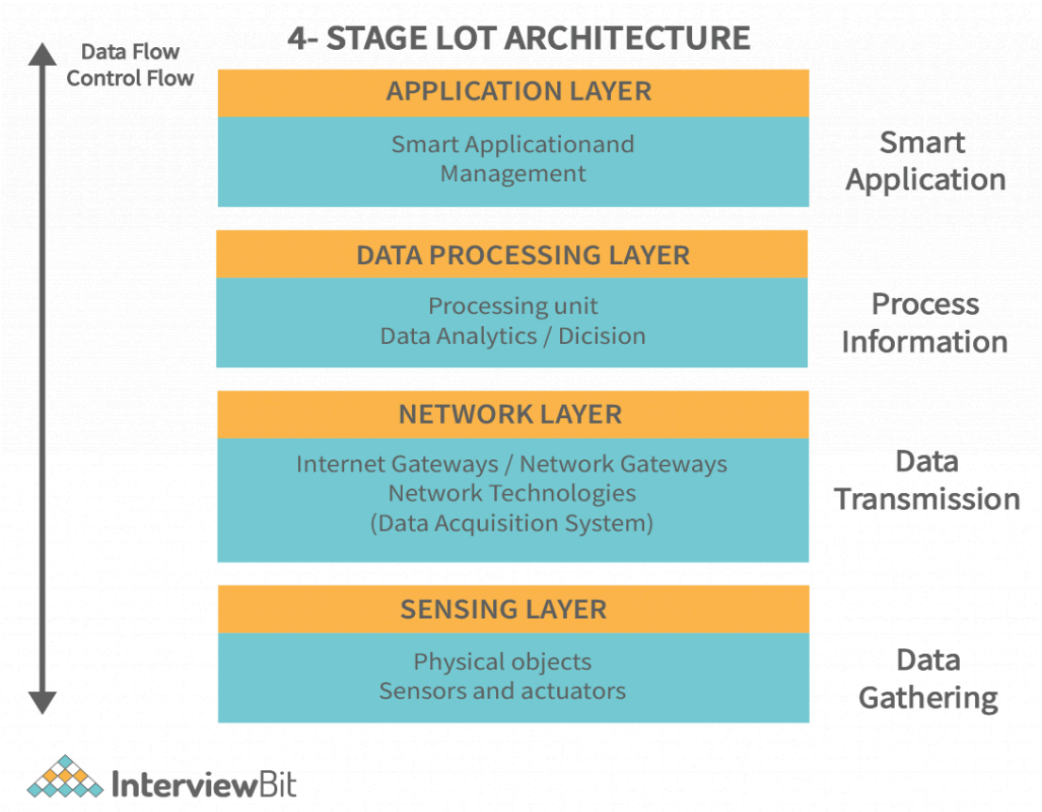
Κεφάλαιο 2 – Συναφείς Τεχνολογίες

Στο δεύτερο κεφάλαιο, θα εξεταστούν οι συναφείς τεχνολογίες που χρειάζονται στην ανάπτυξη και την λειτουργία του έξυπνου ανελκυστήρα.

2.1 Διαδίκτυο των Πραγμάτων (IoT)

Το Industry 4.0 είναι η τέταρτη βιομηχανική επανάσταση, η οποία βασίζεται στην ενσωμάτωση νέων ψηφιακών τεχνολογιών στη βιομηχανία. Αυτή η επανάσταση φέρνει αυξημένη αυτοματοποίηση και συντήρηση, βασιζόμενη σε προβλέψεις, και δίνει τη δυνατότητα για βελτίωση της καθημερινότητας και της εξυπηρέτησης των πελατών. Μια από τις τεχνολογίες που χρησιμοποιούνται σε αυτό το πλαίσιο είναι το Διαδίκτυο των Πραγμάτων (IoT).

Στο Διαδίκτυο των Πραγμάτων (IoT), η αρχιτεκτονική αποτελείται από τέσσερα ενσωματωμένα επίπεδα, το καθένα με την ξεχωριστή του λειτουργικότητα και τον ρόλο του. Αυτά τα επίπεδα αποτελούν τη ραχοκοκαλιά των συστημάτων IoT, επιτρέποντας την ενσωμάτωση και την αλληλεπίδραση μεταξύ του φυσικού και του ψηφιακού κόσμου.



Εικονα 5 Four Stage IoT Architecture - InterviewBit

Το πρώτο επίπεδο είναι γνωστό ως Επίπεδο Αντίληψης ή Ανίχνευσης (Perception/Sensing Layer). Αυτό το επίπεδο είναι εξοπλισμένο με συσκευές τελικού σημείου (endpoints) όπως αισθητήρες και ενεργοποιητές. Αυτές οι συσκευές παίζουν καθοριστικό ρόλο στην συλλογή και επεξεργασία δεδομένων από το περιβάλλον τους. Είναι ευέλικτα ως προς τη συνδεσιμότητά τους, ικανά να συνδέονται τόσο με ασύρματα όσο και με ενσύρματα μέσα. Ο σχεδιασμός αυτού του στρώματος είναι ευέλικτος, επιτρέποντας διάφορους τύπους εξαρτημάτων και την χρήση τους σε διαφορετικές τοποθεσίες.

Το δεύτερο επίπεδο είναι γνωστό ως Επίπεδο Δικτύου (Network Layer). Αυτό το επίπεδο είναι ζωτικής σημασίας για τη διαχείριση της ροής δεδομένων σε όλα τα συστήματα. Αποτελείται από στοιχεία όπως Συστήματα Απόκτησης Δεδομένων (DAS) και πύλες δικτύου (gateways), που διευκολύνουν τη συγκέντρωση δεδομένων από τους αισθητήρες. Επιπλέον, αυτό το επίπεδο είναι υπεύθυνο για τη μετατροπή των αναλογικών δεδομένων σε ψηφιακή μορφή. Όμως, η κύρια λειτουργία του είναι να διασφαλίζει την ομαλή μετάδοση και επικοινωνία δεδομένων μεταξύ αισθητήρων και άλλων συσκευών, συμπεριλαμβανομένων των διακομιστών και των έξυπνων συσκευών.

Το τρίτο επίπεδο είναι γνωστό ως το Επίπεδο Επεξεργασίας (Processing Layer) και συχνά αποκαλείται ο εγκέφαλος του οικοσυστήματος IoT. Σε αυτό το επίπεδο, τα δεδομένα όχι μόνο αναλύονται και προεπεξεργάζονται αλλά αποθηκεύονται και για μελλοντική χρήση. Είναι ένα κρίσιμο σημείο όπου τα δεδομένα είναι προσβάσιμα από τις διάφορες εφαρμογές λογισμικού, που τα διαχειρίζονται και τα παρακολουθούν.

Το τέταρτο επίπεδο είναι γνωστό ως το Επίπεδο Εφαρμογών (Application Layer). Αντιπροσωπεύει τη διεπαφή όπου οι χρήστες αλληλεπιδρούν με το σύστημα IoT. Αυτό το επίπεδο είναι γνωστό για την παροχή υπηρεσιών στους χρήστες. Για παράδειγμα, σε ένα έξυπνο σπίτι οι χρήστες ενδέχεται να αλληλεπιδράσουν με μία εφαρμογή για τον έλεγχο των οικιακών συσκευών. Εκεί φαίνεται η ευελιξία του IoT, επιδεικνύοντας τη χρησιμότητα του σε διάφορες τομείς όπως τα έξυπνα σπίτια, τις έξυπνες πόλεις και έξυπνη υγεία.

Αυτά τα επίπεδα συνεργάζονται για να δημιουργήσουν ένα ευέλικτο και πολύπλοκο σύστημα IoT, αλλάζοντας θετικά τον τρόπο με τον οποίο χρησιμοποιούμε την τεχνολογία στην καθημερινότητα μας.

Το Διαδίκτυο των Πραγμάτων (IoT) αναφέρεται στην σύνδεση και την επικοινωνία πολλαπλών συσκευών μέσου του διαδικτύου. Σε αυτόν τον τομέα, οι συσκευές διαθέτουν

αισθητήρες, επεξεργαστές και δυνατότητα επικοινωνίας, επιτρέποντας την συλλογή, την ανάλυση και την ανταλλαγή δεδομένων με άλλες συσκευές και συστήματα.

Αυτές οι συσκευές, γνωστές και ως έξυπνες συσκευές, όπου συνδέονται συνήθως στο διαδίκτυο και δημιουργούν ένα δίκτυο με τις άλλες συνδεδεμένες συσκευές. Μέσα σε αυτό το δίκτυο, οι συσκευές μπορούν να ανταλλάσσουν δεδομένα και να εκτελούν αυτόματα εργασίες. Αυτό μπορεί να προσαρμοστεί άμεσα σε διάφορους τομείς, όπως σε αγρόκτημα που να παρακολουθεί τις μετεωρολογικές παραμέτρους όπως θερμοκρασία, υγρασία και βροχή. Επίσης, μπορεί να προσαρμοστεί για διαχείριση μοτίβων κυκλοφορίας αλλά και σε έλεγχο μηχανημάτων εργοστασίου.

Επιπλέον, η συλλογή και η ανάλυση δεδομένων από τις έξυπνες συσκευές δίνει την δυνατότητα στις επιχειρήσεις να λαμβάνουν προληπτικά μέτρα και να βελτιώνουν διαδικασίες τους, προσφέροντας έτσι καλύτερες υπηρεσίες στους πελάτες τους. Αυτό τις καθιστά πιο ανταγωνιστικές, ενισχύοντας την θέση τους στην αγορά.

Το Διαδίκτυο των Πραγμάτων (IoT) τα τελευταία χρόνια γίνεται μία από τις πιο σημαντικές τεχνολογίες της εποχής μας. Σχετίζεται με την σύνδεση καθημερινών συσκευών όπως το ψυγείο ή αυτοκίνητο, στο διαδίκτυο μέσω ειδικών ενσωματωμένων τεχνολογιών. Αυτή η σύνδεση δίνει την δυνατότητα σε αυτές τις συσκευές να επικοινωνούν μεταξύ τους αλλά και με τον άνθρωπο. Η εύκολη πρόσβαση σε υπολογιστές, αποθήκευση σε cloud και κινητά καθιστά όλα αυτά εφικτά. Το Διαδίκτυο των Πραγμάτων (IoT) ενώνει τον φυσικό κόσμο και τον ψηφιακό κόσμο με απώτερο σκοπό να κάνει την ζωή μας και τις επιχειρήσεις καλύτερες.

Ένα από τα σημαντικά πράγματα που κάνει το Διαδίκτυο των Πραγμάτων (IoT) είναι να λειτουργούν σωστά οι συσκευές και να εξοικονομεί χρόνο. Για παράδειγμα, οι αισθητήρες σε συσκευές IoT μπορούν να παρακολουθούν πως λειτουργούν οι μηχανές και όταν πρόκειται να χαλάσουν να ειδοποιούν. Αυτό σημαίνει ότι έξοδα για επισκευές μειώνονται, καθώς και ο χρόνος που τα μηχανήματα δεν λειτουργούν, κάτι που είναι καλό για τις επιχειρήσεις.

Ένα άλλο σημαντικό όφελος του IoT είναι η συγκεντρώσει δεδομένων. Αυτά τα δεδομένα είναι κρίσιμα για να παρθούν καλύτερες αποφάσεις. Δηλαδή μπορούν να καταλάβουν τι αρέσει στου πελάτες, τι είναι δημοφιλές στην αγορά και πόσο καλά τα πάει η επιχείρηση.

Πολλές τεχνολογίες συνδυάζονται για να κάνουν εφικτό το Διαδίκτυο των Πραγμάτων (IoT). Μία από αυτές είναι οι αισθητήρες (sensors) και οι ενεργοποιητές (actuators). Οι αισθητήρες είναι συσκευές σχεδιασμένες να ανιχνεύουν αλλαγές στο περιβάλλον, συμπεριλαμβανομένων την θερμοκρασία, την υγρασία, τη φωτεινότητα, την κίνηση και την πίεση. Από την άλλη πλευρά, οι ενεργοποιητές είναι συσκευές που μπορούν να αλλάξουν το φυσικό περιβάλλον, όπως άνοιγμα ή κλείσιμο μιας βαλβίδας, ενεργοποίηση ενός κινητήρα ή και άνοιγμα μιας λάμπας. Αυτές οι συσκευές μπορούν να θεωρηθούν και ως τα θεμελιώδη δομικά στοιχεία του IoT, διότι βοηθούν την αλληλεπίδραση των μηχανών με τον φυσικό κόσμο.

Οι συσκευές IoT βασίζονται σε ένα δίκτυο τεχνολογιών συνδεσιμότητας (Connectivity technologies) για την μεταφορά δεδομένων από αισθητήρες και ενεργοποιητές στο cloud. Για να επιτευχθεί αυτό είναι απαραίτητο οι συσκευές να είναι συνδεμένες στο διαδίκτυο. Το οικοσύστημα του IoT έχει διάφορες τεχνολογίες συνδεσιμότητας και οι βασικές τους είναι το Wi-Fi, το Bluetooth, Cellular, Zigbee και LoRaWAN, όπου καθεμία από αυτές καλύπτει συγκεκριμένες ανάγκες και απαιτήσεις.

Το Υπολογιστικό Νέφος (Cloud Computing) είναι ένας από τους χώρους που αποθηκεύονται οι τεράστιοι όγκοι δεδομένων που παράγονται από της έξυπνες συσκευές του IoT. Οι πλατφόρμες του Υπολογιστικό Νέφος συνήθως προσφέρουν εργαλεία που είναι απαραίτητα για την διαχείριση και την ανάλυση των δεδομένων. Επιπλέον δίνουν την δυνατότητα στους προγραμματιστές να δημιουργούν και να αναπτύσσουν τις IoT εφαρμογές τους.

Ο τεράστιος όγκος των δεδομένων που δημιουργείται από τις IoT εφαρμογές χρειάζεται χρήση εξελιγμένων εργαλείων ανάλυσης μεγάλων δεδομένων (Big Data Analytics). Οι εταιρείες συνηθίζουν να χρησιμοποιούν εργαλεία όπως μηχανική μάθηση και οπτικοποίηση δεδομένων.

Με την αυξημένη ανάπτυξη του Διαδικτύου Των Πραγμάτων (IoT), το ζήτημα της ασφάλειας και του απορρήτου είναι αναγκαίο να αποκτούν μεγαλύτερη σημασία. Πρέπει να υπάρχει προτεραιότητα στην προστασία των συσκευών του IoT και των δεδομένων τους Χρησιμοποιούνται διάφορες τεχνολογίες για την ενίσχυση της ασφάλειας του IoT, μια από αυτές είναι η κρυπτογράφηση. Οι έλεγχοι πρόσβασης (Access Control) απαγορεύουν την πρόσβαση σε μη εξουσιοδοτημένη άτομα, ενώ τα συστήματα ανίχνευσης εισβολής χρησιμοποιούνται για την παρακολούθηση και τον εντοπισμό ύποπτων δραστηριοτήτων. Με

τη εφαρμογή αυτών των τεχνολογιών ασφαλείας και απορρήτου διασφαλίζεται η αξιοπιστία των εφαρμογών του IoT.

Βλέπουμε πολλά παραδείγματα ότι οι εφαρμογές του Διαδικτύου των Πραγμάτων (IoT) να έχουν ενσωματωθεί και να λαμβάνουν ένα κρίσιμο ρόλο σε αρκετούς τομείς στην καθημερινότητα μας. Ένας από αυτούς είναι ο τομέας της υγειονομικής περίθαλψης. Το IoT επιτρέπει την απομακρυσμένη παρακολούθηση ασθενών και τη συλλογή δεδομένων σε πραγματικό χρόνο. Αυτά τα δεδομένα μπορούν να αναλυθούν για τον εντοπισμό πιθανών προβλημάτων υγείας, ενισχύοντας έτσι την πρόληψη.

Στη γεωργία, οι συσκευές του IoT παίζουν και εκεί σημαντικό ρόλο. Χρήσεις του φαίνονται στην παρακολούθηση του εδάφους και των καιρικών φαινομένων. Δηλαδή, υπάρχουν αισθητήρες μέτρησης υγρασίας του εδάφους βοηθώντας έτσι στην επιλογή του σωστού χρόνου άρδευσης της καλλιέργειας. Επιπλέον οι συσκευές του IoT μπορούν να χρησιμοποιηθούν για την επίβλεψη της υγείας των ζώων και την παρακολούθηση της κατάστασης των γεωργικών μηχανημάτων, ενισχύοντας πάλι στην πρόληψη.

Στον τομέας της μεταφοράς, είναι και εκεί φανερή η επίδραση των συσκευών του IoT. Η παρακολούθηση της κατάστασης του οχήματος αλλά και η βελτιστοποίηση της διαδρομής δείχνουν τον σημαντικό ρόλο αυτών των συσκευών. Για παράδειγμα, οι αισθητήρες χρησιμοποιούνται για την παρατήρηση της απόδοσης του καυσίμου, οδηγώντας στην εξοικονόμηση κόστους στα καύσιμα και συμβάλλοντας θετικά στο περιβάλλον.

Επιπλέον, οι φυσικοί υπολογιστές (Physical Computing) έχουν παίξει σημαντικό ρόλο στην ανάπτυξη του Διαδικτύου των Πραγμάτων (IoT). Οι φυσικοί υπολογιστές περιλαμβάνουν ενσωματωμένα συστήματα που βασίζονται σε μικροελεγκτές, που είναι ικανά να αλληλοεπιδρούν με το περιβάλλον τους, ελέγχοντας εξόδους όπως φώτα, οθόνες και κινητήρες. Οι διάφορες φυσικές υπολογιστικές συσκευές που υπάρχουν στην αγορά όπως τον Arduino, το Raspberry Pi, το Circuit Playground και το BBC micro:bit, έχουν βοηθήσει μαθητές, εκπαιδευτικούς και χομπίστες για να εκπαιδεύονται και να δημιουργούν καινοτόμες εφαρμογές του IoT.

2.2 Έξυπνοι Ανελκυστήρες

Οι ανελκυστήρες, είναι ένα γνωστό μέσο μεταφοράς που χρησιμοποιείται καθημερινά από ανθρώπους για γρήγορη μετακίνηση τους στους ορόφους κτηρίων. Είναι φυσικό οι

ανελκυστήρες να εξελιχθούν και το βλέπουμε μέσα από αναβαθμίσεις σε προηγμένα υλικά, όπως σχοινιά από ανθρακονήματα και πιο ανθεκτικούς μηχανισμούς για τη βελτίωση της λειτουργικότητας τους. Τώρα, όμως, η σχεδίαση προχωράει ένα βήμα παραπέρα με τις τεχνολογίες του Διαδικτύου των Πραγμάτων (IoT).

Η ενσωμάτωση του IoT αναδιαμορφώνει τους ανελκυστήρες, επεκτείνοντας τις λειτουργίες τους πέρα από την απλή μεταφορά. Αυτοί οι έξυπνοι ανελκυστήρες διευκολύνουν την κίνηση, αλλά και συλλέγουν δεδομένα, παρακολουθούν το ροή κίνησης και λειτουργούν και ως εργαλεία διαφήμισης.

Οι λύσεις Elevator IoT της Robustel απευθύνονται σε κατασκευαστές ανελκυστήρων και εταιρείες συντήρησης, δίνοντας τους τη δυνατότητα να επιβλέπουν αποτελεσματικά την απόδοση του μηχανήματος τους. Η ενσωμάτωση μιας συσκευής Robustel Cloud Management Service (RCMS) διασφαλίζει απλοποιημένη διαχείριση του δικτύου, της συνδεσιμότητας και της παροχής.

Τα πλεονεκτήματα της εφαρμογής ενός ανελκυστήρα IoT είναι ποικίλα. Δίνουν στους χειριστές νέες δυνατότητες για πιο αποτελεσματική παρατήρηση, ανάλυση και απόκριση. Τα οφέλη των λύσεων ανελκυστήρων IoT είναι πολλά. Μερικά από αυτά είναι τα ενισχυμένα μέτρα ασφαλείας, η εξοικονόμηση χρόνου και οι οικονομικές αναβαθμίσεις, η μείωση του χρόνου αναμονής για τους χρήστες, οι άμεσες απαντήσεις έκτακτης ανάγκης, η απομακρυσμένη διάγνωση προβλημάτων και η εφαρμογή ενεργειακών αποδοτικών συστημάτων. Ακόμη με την τεχνολογία Smart Elevator υπάρχει βελτιστοποίηση των λειτουργιών, τα συστήματα διαχείρισης κτιρίων είναι παρόμοια και οι λύσεις IoT στοχεύουν κυρίως στην παρακολούθηση των λειτουργικών συνθηκών. Αξιοποιώντας τεχνολογίες όπως αισθητήρες του ανελκυστήρα και συλλογή δεδομένων. Υπάρχουν αρκετά είδη δεδομένων που μπορούν να συγκεντρωθούν όπως η μέτρηση φορτίου, τα κρίσιμα κυκλώματα ασφαλείας, την μέτρηση καθημερινών δρομολογίων, τις ώρες αναμονής, τις ώρες αιχμής και την ανάλυση διαδρομών (επιτάχυνση, τραντάγματα, δονήσεις).

Αυτές οι πληροφορίες εξοπλίζουν τα συστήματα διαχείρισης των ανελκυστήρων για τη λήψη καλύτερων αποφάσεων σχετικά με τα χρονοδιαγράμματα λειτουργίας, την υποδοχή εκδηλώσεων ή τον προγραμματισμό γύρω από τις ώρες άφιξης τρένων.

Ένα από τα μεγαλύτερα πλεονεκτήματα της σύνδεσης του IoT με τον ανελκυστήρα είναι η βελτιστοποίηση του χρονοδιαγράμματος για την συντήρηση του, εξοικονομώντας έτσι κρίσιμο χρόνο. Οι συσκευές συνηθίζεται να παρακολουθούν τις λειτουργικές αλλαγές,

όπως διακυμάνσεις θερμότητας ή θορύβου και έτσι προβλέπεται η ανάγκη της συντήρησης. Συνήθως, η συντήρηση του ανελκυστήρα γίνεται με βάση σταθερού χρονοδιαγράμματος, αλλά με την βοήθεια του IoT, υιοθετείται μια προληπτική προσέγγιση. Ακόμη η απομακρυσμένη παρακολούθηση διευκολύνει τον προγραμματισμό και την εκτέλεση της συντήρησης σε ώρες εκτός αιχμής.

Συλλέγοντας δεδομένα για κάθε εξάρτημα του ανελκυστήρα, ακόμη και σε σημεία όπως αισθητήρες πόρτας του ανελκυστήρα, η προληπτική ανάλυση επιτρέπει την επίλυση προβλημάτων προτού γίνουν, αποτρέποντας έτσι πιθανές διακοπές.

Δεδομένα συλλέγονται από όλα τα εξαρτήματα ενός ανελκυστήρα. Αυτά τα δεδομένα μπορούν να χρησιμοποιηθούν για την πρόβλεψη και επίλυση προβλημάτων πριν αυτά συμβούν, αποτρέποντας έτσι πιθανές διακοπές λειτουργίας.

Με την βοήθεια του IoT δηλαδή την δυνατότητα της απομακρυσμένης παρακολούθησης επιτρέπει στους τεχνικούς να μην είναι φυσικά παρόντες ώστε να διαγνώσουν τις βλάβες, με αποτέλεσμα να μειωθούν οι καθυστερήσεις στις παραγγελίες ανταλλακτικών. Η ενοποίηση του IoT με τους ανελκυστήρες επιτρέπει στους τεχνικούς να εντοπίζουν άμεσα τα προβλήματα εξ αποστάσεως, επιταχύνοντας την διαδικασία αντιμετώπισης προβλημάτων.

Το Dusun DSGW-089 IIoT Gateway έχει σχεδιαστεί ειδικά για την βιομηχανία ανελκυστήρων IoT. Χρησιμοποιεί την τεχνολογία i.MX8M mini SoC, παρέχοντας υψηλή απόδοση και αποτελεσματικότητα στην κατανάλωση ενέργειας. Είναι εξοπλισμένο με τετραπύρηνο επεξεργαστή, μνήμη RAM τύπου DDR4, αποθηκευτικό χώρο eMMC, επιπλέον αυτή η πύλη υποστηρίζει διάφορα πρωτόκολλα επικοινωνίας όπως RS485, CAN, Ethernet και Wi-Fi.

Με τις πολλαπλές διεπαφές I/O, η πύλη DSGW-089 IIoT επιτρέπει την εύκολη ενσωμάτωση με τα υπάρχοντα συστήματα ελέγχου των ανελκυστήρων. Επιτρέπει την παρακολούθηση, την ανάλυση και τον αυτοματισμό σε πραγματικό χρόνο, καθιστώντας την μια καλή επιλογή για τη συλλογή και ανάλυση δεδομένων από αισθητήρες και ελεγκτές σε συστήματα ανελκυστήρων IoT.

Το Dusan DSGW-089 IIoT Gateway έχει σχεδιαστεί για να βελτιστοποιεί τα δεδομένα (optimize data), να παρέχει απόκριση σε πραγματικό χρόνο (real-time response) και να επιτρέπει την έξυπνη ανάλυση στο IoT edge. Μειώνει σημαντικά τη ροή δεδομένων μεταξύ τοποθεσιών πεδίου (filed sites) και κέντρων δεδομένων (data centers), αποφεύγοντας

τα σημεία συμφόρησης στο cloud-end computing. Έχει συμπαγή και ανθεκτικό σχεδιασμό, μαζί με χαμηλή κατανάλωση ενέργειας που το κάνουν κατάλληλο για συστήματα ανελκυστήρων IoT.

Συνοπτικά, το Dusan DSGW-089 IIoT Gateway αποτελεί μια συνιστώμενη λύση για την ανάπτυξη συστημάτων ανελκυστήρων στο Διαδίκτυο των Πραγμάτων (IoT). Οι προηγμένες δυνατότητες, η υποστήριξη πολλαπλών πρωτοκόλλων και οι δυνατότητες επεξεργασίας δεδομένων σε πραγματικό χρόνο το καθιστούν ιδανική επιλογή για αποτελεσματική και αξιόπιστη επικοινωνία στην βιομηχανία ανελκυστήρων IoT.

2.3 Αναγνώριση Προσώπου

Η τεχνολογία αναγνώριση προσώπου αποτελεί ένα τομέα της τεχνητής νοημοσύνης (AI) που προσδιορίζει ανθρώπινα πρόσωπα σε ψηφιακά μέσα, όπως εικόνες και βίντεο. Αυτή η τεχνολογία εφαρμόζεται σε διάφορους τομείς, όπως της ασφάλειας, της βιομετρίας, της ψυχαγωγίας και των μέσων κοινωνικής δικτύωσης. Η ανάλυση προσώπων παίζει κεντρικό ρόλο στην παρακολούθηση, καθώς είναι απαραίτητη για τη δημιουργία και τη σύγκριση των αποτυπωμάτων των προσώπων.

Τα συστήματα ανίχνευσης προσώπου χρησιμοποιούν τεχνητή νοημοσύνη, μηχανική μάθηση και στατιστική ανάλυση για να διακρίνουν ανθρώπινα πρόσωπα μέσα σε ευρύτερες εικόνες, ξεχωρίζοντας τα πρόσωπα από αντικείμενα. Η διαδικασία ξεκινά με την προεπεξεργασία των δεδομένων με σκοπό την βελτίωση της ποιότητας και την μείωση πιθανών παρεμβολών. Οι αλγόριθμοι αρχικά αναζητούν χαρακτηριστικά εύκολα στην εντόπιση όπως τα μάτια, φρύδια, στόμα και τα λοιπά. Όταν εντοπίσουν μια περιοχή του προσώπου τότε γίνονται πρόσθετες δοκιμές με σκοπό να επιβεβαιωθεί ότι έχει εντοπιστεί ένα πρόσωπο. Για να ενισχυθεί η ακρίβεια στην αναγνώριση προσώπων, οι αλγόριθμοι εκπαιδεύονται σε μεγάλα σύνολα δεδομένων που περιλαμβάνουν πολλές εικόνες.

Στην αναγνώριση προσώπου χρησιμοποιούνται διάφορες μέθοδοι και η καθεμία είναι ξεχωριστή. Μία τέτοια μέθοδος είναι ο αλγόριθμος Viola-Jones, ο οποίος βασίζεται σε ένα εκπαιδευμένο μοντέλο που μπορεί να ξεχωρίζει τα πρόσωπα με άλλα αντικείμενα. Αυτή η προσέγγιση εξακολουθεί να χρησιμοποιείται ευρέως για την αναγνώριση προσώπου σε πραγματικό χρόνο, δεν συνιστάται για να εντοπίσει με ακρίβεια πρόσωπα που είναι κρυμμένα ή δεν είναι στην κατάλληλη γωνία.

Ακόμη υπάρχει και η μέθοδος που βασίζεται σε κανόνες, τα πρόσωπα ορίζονται χρησιμοποιώντας ένα σύνολο προκαθορισμένων κανόνων. Από την άλλη πλευρά, υπάρχουν και μέθοδοι με προσεγγίσεις που βασίζονται στον εντοπισμό χαρακτηριστικών του προσώπου όπως τα μάτια ή οι μύτες και τα λοιπά. Μία άλλη μέθοδος, περιλαμβάνει τη σύγκριση εικόνων με αποθηκευμένα τυπικά μοτίβα προσώπων για την αναγνώριση προσώπων. Υπάρχουν και μέθοδοι που βασίζονται στην στατιστική ανάλυση και μηχανική μάθηση για τον προσδιορισμό των χαρακτηριστικών του προσώπου.

Πιο προηγμένες μέθοδοι περιλαμβάνουν την χρήση συνελικτικών νευρωνικών δικτύων (CNN), τα οποία χρησιμοποιούν δίκτυα βαθιάς μάθησης (deep learning) για την επεξεργασία εικόνας και τον εντοπισμό αντικειμένων. Επιπλέον, οι τεχνικές ανιχνευτή μονής λήψης (SSD), επιτρέπουν ταχύτερη ανίχνευση με την χρήση μόνο μιας εικόνας για την αναγνώριση πολλών αντικειμένων. Κάθε μια από αυτές τις μεθόδους συμβάλει στην ακρίβεια και την αποτελεσματικότητα της αναγνώρισης προσώπου.

Το Teachable Machine είναι ένα διαδικτυακό εργαλείο από την Google, που βοηθάει τους χρήστες να φτιάξουν τα δικά τους μοντέλα μηχανικής μάθησης. Είναι φιλικό προς το χρήστη, επομένως δεν είναι προαπαιτούμενες πολλές τεχνικές γνώσεις για να χρησιμοποιηθεί. Μπορεί να εκπαιδευτεί με εικόνες, ήχους ή κινήσεις και μαθαίνει γρήγορα να τα αναγνωρίζει. Βασίζεται σε μια μέθοδο που ονομάζεται μάθηση μεταφοράς (transfer learning), που σημαίνει ότι ξεκινά με ένα υπάρχον εκπαιδευμένο μοντέλο και προσαρμόζεται με τα νέα δεδομένα του χρήστη. Έτσι μειώνεται η ανάγκη για μεγάλα σύνολα δεδομένων και υπολογιστική ισχύ.

Το Teachable Machine χρησιμοποιεί το TensorFlow.js, μια βιβλιοθήκη μηχανικής μάθησης που βασίζεται στην JavaScript. Επιπλέον, δίνει τη δυνατότητα στους χρήστες να εκπαιδεύουν και να τρέχουν τα μοντέλα άμεσα στο πρόγραμμα περιήγησης (web browser).

Στο Teachable Machine, η διαδικασία ταξινόμησης εικόνων βασίζεται στο MobileNet, ένα προεκπαιδευμένο μοντέλο συνελικτικού νευρωνικού δικτύου (CNN), το οποίο έχει εκπαιδευτεί χρησιμοποιώντας τη μεγάλη βάση δεδομένων ImageNet. Αυτή η εκπαίδευση επιτρέπει στον classifier, να αναγνωρίζει με ακρίβεια αντικείμενα σε ψηφιακές εικόνες. Το θετικό του μοντέλου είναι η χρήση της μάθησης μεταφοράς (transfer learning), η οποία μας επιτρέπει να δημιουργούμε τα μοντέλα πιο γρήγορα και πιο αποτελεσματικά. Ακόμη με το MobileNet η εκπαίδευση του γίνεται πιο γρήγορα και με ακριβείς προβλέψεις,

καθιστώντας το ένα πολύτιμο τόσο στην επιστήμη των υπολογιστών όσο και στις καθημερινές εφαρμογές.

2.4 Αναγνώριση Ομιλίας

Η αναγνώριση ομιλίας είναι μια τεχνολογία που επιτρέπει σε ένα πρόγραμμα να μετατρέπει την ανθρώπινη ομιλία σε γραπτή μορφή. Σε αντίθεση με την αναγνώριση φωνής, η οποία προσδιορίζει την φωνή ενός συγκεκριμένου χρήστη. Δηλαδή, η αναγνώριση ομιλίας εστιάζει στην μετάφραση της ομιλίας από προφορική σε μορφή κειμένου.

Στον τομέα της επιστήμης των υπολογιστών, η αναγνώριση ομιλίας είναι ένας ιδιαίτερα περίπλοκος τομέας, που συνδυάζει κλάδους όπως η γλωσσολογία, τα μαθηματικά και η στατιστική. Η αναγνώριση ομιλίας αποτελείται από μερικά στοιχεία, όπως η είσοδος ομιλίας, η εξαγωγή χαρακτηριστικών, τα διανύσματα χαρακτηριστικών, ένας αποκωδικοποιητής και μία έξοδος λέξης. Ο αποκωδικοποιητής χρησιμοποιεί ακουστικά μοντέλα όπως ένα λεξικό προφοράς και μοντέλα γλώσσας για να καθορίσει το σωστό τελικό κείμενο.

Η αποτελεσματικότητα της τεχνολογίας αναγνώριση ομιλίας μετριέται συχνά από το ποσοστό ακρίβειας, γνωστό ως ποσοστό σφάλματος λέξης (WER), και την ταχύτητα επεξεργασίας της. Παράγοντες όπως η προφορά, η έμφαση, το ύψος και η ένταση της φωνής αλλά θόρυβοι που ακούγονται από το βάθος, μπορούν να επηρεάσουν το WER. Ο μακροχρόνιος στόχος στην ερευνά για την αναγνώριση της ομιλίας, είναι να πετύχει ποσοστό σφάλματος παρόμοιο με τις ανθρώπινες συνομιλίες.

Αρκετοί αλγόριθμοι και τεχνικές υπολογισμού χρησιμοποιούνται για την βελτίωση της ακρίβειας της αναγνώρισης ομιλίας. Μία από τις βασικές μεθόδους είναι η Επεξεργασία Φυσική Γλώσσας (NLP), όπου είναι ένας τομέας της τεχνητής νοημοσύνης. Εστιάζει στην αλληλεπίδραση μεταξύ ανθρώπων και μηχανών μέσω της γλώσσας. Έχει κρίσιμο ρόλο στην αναγνώριση ομιλίας, διευκολύνοντας τις φωνητικές αναζητήσεις και βελτιώνει τη προσβασιμότητα σε κινητές συσκευές.

Τα Hidden Markov Models (HMM) είναι μοντέλα που βασίζονται στην αλυσίδα Markov, όπου η πιθανότητα μιας κατάστασης εξαρτάται μόνο από την τρέχουσα κατάσταση και όχι από τις προηγούμενες καταστάσεις. Είναι ένα τρόπος για να προβλέπεται μια σειρά γεγονότων. Σκεφτείτε μια αλληλουχία από καταστάσεις, όπως λέξεις ή συλλαβές στην ομιλία. Στα HMM, η πρόβλεψη για την επόμενη κατάσταση βασίζεται μόνο στην τρέχουσα

κατάσταση, χωρίς να λαμβάνεται υπόψη οι προηγούμενες. Αυτό είναι χρήσιμο στην αναγνώριση ομιλίας, όπου τα HMM βοηθούν το σύστημα να αποφασίσει ποια λέξη ή συλλαβή ταιριάζει καλύτερα με την ομιλία που ανιχνεύεται.

Τα N-grams είναι μια τεχνική που αναλύσει τις πιθανότητες των προτάσεων ή φράσεων, βασισμένη σε ακολουθίες N λέξεων. Η μέθοδος αυτή συμβάλει στην βελτίωση της αναγνώρισης και της ακρίβειας σε συστήματα γλωσσικής επεξεργασίας, χρησιμοποιώντας την γραμματική και την πιθανότητα των λεκτικών ακολουθιών.

Τα νευρωνικά δίκτυα χρησιμοποιούνται σε αλγόριθμους βαθιάς μάθησης, αποτελούνται από στρώματα κόμβων που μιμούνται τη δομή του ανθρώπινου εγκεφάλου. Κάθε κόμβος επεξεργάζεται τα δεδομένα και, εάν η τιμή εξόδου του υπερβεί ένα συγκεκριμένο όριο τότε ενεργοποιεί το επόμενο επίπεδο στο δίκτυο. Αυτά τα δίκτυα μαθαίνουν μέσω καθοδηγούμενης μάθησης και βελτιώνονται βάσει μιας συνάρτησης απώλειας, χρησιμοποιώντας την τεχνικής τη καθοδικής ανάβασης (gradient descent). Ενώ προσφέρουν υψηλότερη ακρίβεια στην αναγνώριση ομιλίας, απαιτούν περισσότερους πόρους για την εκπαίδευση τους σε σύγκριση με παραδοσιακά μοντέλα.

Οι αλγόριθμοι Speaker Diarization (SD) ειδικεύονται στην αναγνώριση και τμηματοποίηση της ομιλία από διαφορετικούς ομιλητές, βοηθώντας στη διάκριση συγκεκριμένων ομιλητών. Αυτό είναι πολύ χρήσιμο για την αναγνώριση ομιλίας επειδή μπορεί να χρησιμοποιηθεί σε περιβάλλοντα όπου συμμετέχουν πολλοί ομιλητές.

Το Web Speech API είναι ένα εργαλείο που προσφέρει την επιλογή να ενταχθούν φωνητικά δεδομένα σε εφαρμογές του web. Αποτελείται από δύο κύρια στοιχεία, το SpeechSynthesis, το οποίο χρησιμοποιείται για μετατροπή κειμένου σε ομιλία και το SpeechRecognition, που χρησιμοποιείται για την ασύγχρονη αναγνώριση ομιλίας.

Το Responsive Voice API είναι μια υπηρεσία μετατροπής κειμένου σε ομιλία (TTS) που επιτρέπει την εύκολη ενσωμάτωση φωνητικών λειτουργιών σε εφαρμογές του web. Περιλαμβάνει διάφορες λειτουργίες και παρέχει υποστήριξη σε πάνω από 51 γλώσσες συμπεριλαμβανομένης και της ελληνικής. Ακόμη είναι δωρεάν για μη εμπορική χρήση, που το καθιστά κατάλληλη λύση για την προσθήκη δυναμικών λειτουργιών φωνής και βελτιώνει την προσβασιμότητα στο web.

2.5 Διαδίκτυο των Πραγμάτων (IoT) και Web

Στο τομέα του Διαδικτύου των Πραγμάτων (IoT), το διαδίκτυο έχει θεμελιώδη ρόλο ως πλατφόρμα αλληλεπίδρασης και ανταλλαγής δεδομένων μεταξύ των συσκευών IoT και των χρηστών. Οι τεχνολογίες του web επιτρέπουν την εύκολη σύνδεση, την επικοινωνία και τη κοινοποίηση δεδομένων, διευκολύνοντας την παρακολούθηση και τον έλεγχο σε πραγματικό χρόνο. Η ενσωμάτωση του IoT στην ανάπτυξη ιστοσελίδων βελτιώνει την εμπειρία του χρήστη, προσφέροντας δυναμικές και με γρήγορη ανταπόκριση διεπαφές που μπορούν να αλληλεπιδρούν με διάφορους αισθητήρες και συσκευές.

Τα Restful APIs αποτελούν σημαντικό κομμάτι στον τομέα του Διαδικτύου των Πραγμάτων (IoT), προσφέροντας ένα αποδοτικό τρόπο για την επικοινωνία των συσκευών μέσα από το διαδίκτυο. Ακόμη, χρησιμοποιούνται γνωστά πρωτόκολλα δικτύου όπως το HTTP, διευκολύνοντας την ανταλλαγή πληροφοριών μεταξύ διαφορετικών συσκευών του IoT. Αυτό, σε συνδυασμό με την ικανότητα τους να υποστηρίζουν διάφορες μορφές δεδομένων όπως το JSON ή το XML, καθιστά τα RESTful APIs μια δημοφιλή επιλογή για τις εφαρμογές IoT.

Η React.js, είναι γνωστή στις εφαρμογές του Διαδικτύου των Πραγμάτων (IoT) λόγου του αποτελεσματικού χειρισμού των διεπαφών χρήστη και των δεδομένων σε πραγματικό χρόνο. Τα βασικά χαρακτηριστικά που κάνουν την React.js κατάλληλη για το περιβάλλον του IoT είναι η δομή σύνθεσης για τη δημιουργία επαναχρησιμοποιήσιμων στοιχείων (components) και οι μέθοδοι κύκλοι ζωής (lifecycle) για την ενημέρωση δεδομένων σε πραγματικό χρόνο. Τα παραπάνω καθιστούν το React.js ένα αποτελεσματικό εργαλείο για τις εφαρμογές του IoT, ιδιαίτερα για εφαρμογές όπου η αλληλεπίδραση του χρήστη με τις διεπαφές αποτελεί προτεραιότητα.

Στην πτυχιακή χρησιμοποιούνται τα παρακάτω API: το Responsive Voice, το Open Weather Map, το Web Speech και το Navigator Geolocation. Η επιλογή να χρησιμοποιηθούν API έγινε για να ενισχύσει την ευελιξία της εφαρμογής και την ταχύτητα ανάπτυξης. Για παράδειγμα, με την χρήση API, η αναγνώριση ομιλίας και η αναγνώριση προσώπου μπορεί εύκολα να ενσωματωθεί και να αντικατασταθεί από άλλες υλοποιήσεις, ανάλογα με τις ανάγκες της εφαρμογής. Αυτό μειώνει τον απαιτούμενο χρόνο για την ανάπτυξη νέων λειτουργιών, καθώς δεν απαιτεί τη δημιουργία νέου κώδικα από την αρχή.

Ακόμη, το API επιτρέπει τη δημιουργία ενός modular συστήματος, δηλαδή ένα σύστημα δομημένο σε μικρότερες, ανεξάρτητες μονάδες ή ενότητες, κάθε μία από τις οποίες υποστηρίζει συγκεκριμένη λειτουργικότητα. Αυτές οι μονάδες μπορούν να προστεθούν ή να

τροποποιηθούν χωρίς να επηρεάσουν το υπόλοιπο σύστημα, καθιστώντας την διαχείριση, την ενημέρωση και την ανάπτυξη πιο ευέλικτες και αποδοτικές.

Επιπλέον στην πτυχιακή χρησιμοποιούνται κάποιες βιβλιοθήκες όπως η sweetalert2, ώστε υπάρχουν όμορφα προειδοποιητικά αναδυόμενα παράθυρα. Η React-data-table-component, για να έχουν οι πίνακες δεδομένων λειτουργίες ταξινόμησης, αλλαγής σελίδας και τα λοιπά. Το Nodemon χρησιμοποιήθηκε για την αυτόματη επανεκκίνηση του server, βοηθώντας την διαδικασία του debugging. Το Cors χρησιμοποιήθηκε για να επιτρέψει τις αιτήσεις προς το διακομιστή. Το Bcrypt χρησιμοποιήθηκε για την κρυπτογράφηση και την αποκρυπτογράφηση του κωδικού του admin. Το framework express χρησιμοποιήθηκε για την δημιουργία RESTful APIs με την Nodejs.

Κεφάλαιο 3 - Σχεδιασμός Συστήματος Έξυπνου Ανελκυστήρα

Σε αυτό το κεφάλαιο, αναλύεται ο σχεδιασμός του συστήματος του έξυπνου ανελκυστήρα. Η ανάλυση απαιτήσεων, οι κατηγορίες χρηστών, οι περιπτώσεις χρήσης του συστήματος, το περιβάλλον λειτουργίας και η σχεδιαστική προσέγγιση.

3.1 Ανάλυση απαιτήσεων

Οι σημαντικότερες απαιτήσεις που πρέπει να έχει το σύστημα συνοψίζονται παρακάτω ως εξής:

1. Αναγνώριση προσώπου ενός ένοικου
2. Αναγνώριση ομιλίας.
3. Φωνητική απάντηση.
4. Εμφάνιση αναγνώριση ομιλίας και φωνητικής απάντησης.
5. Εμφάνιση του τρέχον ορόφου.
6. Εμφάνιση κατάστασης του ανελκυστήρα.
7. Εμφάνιση της τοποθεσίας.
8. Εμφάνιση της ημέρας.
9. Εμφάνιση του καιρού.
10. Εμφάνιση των ενοίκων.
11. Κατάταξη των ενοίκων με αύξουσα ή φθίνουσα σειρά ανάλογα με το όνομα, το επίθετο, το επάγγελμα και τον όροφο.
12. Σύνδεση του διαχειριστή.
13. Αποσύνδεση του διαχειριστή.
14. Προσθήκη ορόφου από τον διαχειριστή.
15. Επεξεργασία ορόφου από τον διαχειριστή.
16. Διαγραφή ορόφου από τον διαχειριστή.
17. Προσθήκη, επεξεργασία και διαγραφή του ονόματος, του επιθέτου, του επαγγέλματος και του ορόφου ενός ένοικου από τον διαχειριστή.

Στην αναγνώριση προσώπου του ένοικου είναι να αναγνωρίζει τον ένοικο και το σύστημα (φωνητική απάντηση) να τον ρωτάει αν πάει στον όροφο που μας έχει δηλώσει. Η αναγνώριση ομιλίας πρέπει να αναγνωρίζει την ομιλία του χρήστη και το σύστημα να του δίνει την δυνατότητα να μεταφερθεί στους ορόφους, μόνο με την χρήση της αναγνώρισης ομιλίας. Η φωνητική απάντηση πρέπει να ενημερώνει τον χρήστη, για παράδειγμα στην έναρξη και άφιξη του ανελκυστήρα σε όροφο. Στην διεπαφή χρήστη πρέπει να φαίνεται η αναγνώριση ομιλίας και η φωνητική απάντηση. Ο χρήστης πρέπει να έχει την δυνατότητα να βλέπει σε ποιον όροφο είναι και την κατάσταση του ανελκυστήρα, δηλαδή αν είναι σταματημένος ή ανεβαίνει ή κατεβαίνει. Στην οθόνη πρέπει να εμφανίζονται ενημερωτικές πληροφορίες όπως η τοποθεσία, ο καιρός της περιοχής και η εμφάνιση των ενοίκων που μας έχουν δώσει τα στοιχεία τους. Ακόμη αυτά τα στοιχεία πρέπει να έχουν την δυνατότητα να ταξινομηθούν με αύξουσα ή φθίνουσα σειρά.

Επιπρόσθετα, ο διαχειριστής πρέπει να έχει μια διεπαφή ώστε να μπορεί να συνδεθεί και να αποσυνδεθεί. Να κάνει προσθήκη, επεξεργασία και διαγραφή των στοιχείων των ένοικων που έχουν καταχωρηθεί αλλά και την προσθήκη, επεξεργασία και διαγραφή των ορόφων και των λέξεων που συνδέονται με τους ορόφους. Για παράδειγμα ο διαχειριστής θα συμπληρώσει ότι ο Γιάννης Παπαδόπουλος είναι οδοντίατρος και έχει το ιατρείο του στον τρίτο όροφο. Οπότε όταν ο χρήστης πει μια από αυτές τις πληροφορίες πχ «Οδοντίατρος» τότε το σύστημα θα πρέπει να καταλάβει ότι χρήστης θέλει να πάει στον τρίτο όροφο που έχει το ιατρείο του ο οδοντίατρος και να ξεκινήσει ο ανελκυστήρας να κατευθύνεται προς τα εκεί. Επίσης, όταν ο χρήστης αναφέρει «πρώτος όροφος» τότε το σύστημα πρέπει να καταλάβει ότι ο χρήστης επιθυμεί να πάει στον πρώτο όροφο και να ξεκινήσει την διαδικασία.

3.2 Κατηγορίες χρηστών

Οι κατηγορίες χρηστών που μπορεί να έχει η πλατφόρμα είναι ο πελάτης, ο ένοικος και ο διαχειριστής.

Ο πελάτης διαθέτει την δυνατότητα να αξιοποιήσει τη λειτουργεία ταξινόμησης προκειμένου να εντοπίσει με άμεσο τρόπο σε ποιόν όροφο βρίσκεται ο επιθυμητός ένοικος. Επιπροσθέτως, μπορεί να επιλέξει να χρησιμοποιήσει τις φωνητικές εντολές όπως να αναφέρει την φράση «Γιώργος Παπαδόπουλος Οδοντίατρος» ή να αναφέρει Παπαδόπουλος.

Σε περίπτωση μοναδικής αντιστοίχισης, ο ανελκυστήρας θα ξεκινήσει αυτόματα και θα μεταβεί στον καθορισμένο όροφο.

Ο ένοικος έχει πρόσβαση σε όλες τις λειτουργίες ενός πελάτη, αλλά έχει επιπλέον την λειτουργία αναγνώριση προσώπου. Όταν μπαίνει στο ασανσέρ, η αναγνώριση προσώπου τον αναγνωρίζει και τον ρωτά αν θέλει να πάει στον όροφο που είναι δηλωμένος στο σύστημα. Σε περίπτωση θετικής απάντησης, ο ασανσέρ ξεκινά και πηγαίνει στον δηλωμένο όροφο. Σε περίπτωση αρνητικής απάντησης, ο ασανσέρ περιμένει μέχρι να δοθεί άλλη εντολή.

Ο διαχειριστής έχει πρόσβαση σε όλες τις λειτουργίες που διατίθενται σε πελάτες και ένοικους, αλλά η κύρια ευθύνη του είναι η διαχείριση των πληροφοριών του συστήματος. Υπάρχει μία ιστοσελίδα διαχείρισης που έχουν πρόσβαση μόνο οι διαχειριστές, όπου μπορούν να προσθέσουν, επεξεργαστούν και να διαγράψουν έναν όροφο. Επιπλέον, έχουν την δυνατότητα να προσθέσουν, να επεξεργαστούν και να διαγράψουν πληροφορίες σχετικά με τους ένοικους όπως το όνομα, το επίθετο, το όνομα και το επάγγελμα.

3.3 Περιπτώσεις Χρήσης Συστήματος

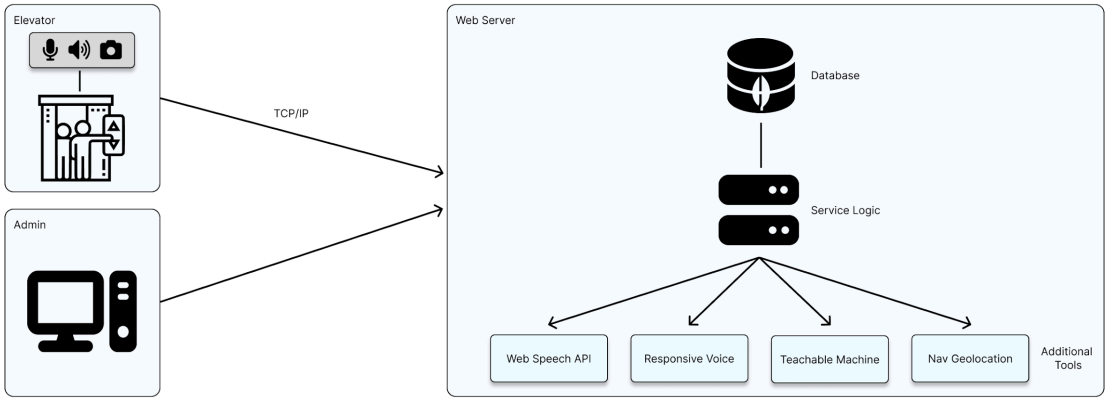
Εμπλεκόμενοι ρόλοι	Περιπτώσεις Χρήσης
Πελάτης	• Λειτουργία ταξινόμησης. • Αναγνώριση ομιλίας.
Ένοικός	• Λειτουργία ταξινόμησης. • Αναγνώριση ομιλίας. • Αναγνώριση προσώπου.
Διαχειριστής	• Λειτουργία ταξινόμησης. • Αναγνώριση ομιλίας. • Αναγνώριση προσώπου. • Σύνδεση στην σελίδα διαχείρισης. • Αποσύνδεση στην σελίδα διαχείρισης. • Προσθήκη, επεξεργασία και διαγραφή ορόφου από τη σελίδα διαχείρισης. • Προσθήκη, επεξεργασία και διαγραφή του ονόματος, του επιθέτου, του επαγγέλματος και του ορόφου ενός ενοίκου από τη σελίδα διαχείρισης.

3.4 Περιβάλλον λειτουργίας

Το περιβάλλον λειτουργίας που χρησιμοποιήθηκε για δοκιμές ήταν πρόγραμμα περιήγησης Google Chrome. Παρατηρήθηκε ότι οι άλλα προγράμματα περιήγησης, όπως ο Mozilla Firefox, δεν υποστηρίζουν πλήρως όλες τις λειτουργίες του Web Speech API, καθιστώντας την διαδικτυακή εφαρμογή μη πλήρως λειτουργική σε αυτά τα περιβάλλοντα.

Πρέπει στο πρόγραμμα περιήγησης να υπάρχουν δυο ιστοσελίδες, μια διεπαφή για τον ανελκυστήρα που θα βλέπει ο χρήστης και θα εμφανίζει τις πληροφορίες που αναφέρθηκαν παραπάνω και μια διεπαφή για τον διαχειριστή που θα μπορεί επεξεργάζεται τις πληροφορίες που αναφέρθηκαν. Αυτές οι δυο διεπαφές θα επικοινωνούν με τον web server όπου αυτός διαχειρίζεται τις πληροφορίες που θα τους δίνει. Όπως τα δεδομένα της βάσης δεδομένων, την επικοινωνία με τα APIs και την επικοινωνία με το Teachable Machine.

3.5 Σχεδιαστική Προσέγγιση



Εικόνα 6 Σχεδιαστική Προσέγγιση

Στη εικόνα 6 βλέπουμε την σχεδιαστική προσέγγιση της διαδικτυακής εφαρμογής.

1) Ανελκυστήρας: Περιέχει μία οθόνη που θα έχει μια ιστοσελίδα για την διεπαφή του χρήστη και ο ανελκυστήρας θα είναι εξοπλισμένος με μικρόφωνο, ηχείο και κάμερα.
2) Διαχειριστής: Έχει μια ιστοσελίδα για να διαχειρίζεται τα δεδομένα της διαδικτυακής εφαρμογής.
3) Επικοινωνία: Η επικοινωνία ανάμεσα στους clients και τον server γίνεται μέσω TCP/IP.
4) Web Server: Απαντάει στα αιτήματα του χρήστη και του διαχειριστή.

5) Βάση Δεδομένων: Χρησιμοποιείται για την αποθήκευση των δεδομένων της διαδικτυακής εφαρμογής.
6) Service Logic: Εξασφαλίζει την επικοινωνία με την βάση δεδομένων και των εξωτερικών εργαλείων.
7) Πρόσθετα Εργαλεία:
    a. Web Speech API: παρέχει την αναγνώριση της ομιλίας. (Speech To Text)
    b. Responsive Voice: προσφέρει την δυνατότητα φωνητικής απάντησης στα ελληνικά. (Text To Speech)
    c. Teachable Machine: Χρησιμοποιείται για την αναγνώριση εικόνων μέσω μοντέλου μηχανικής μάθησης.
    d. Navigation Geolocation: Χρησιμοποιείται για την εύρεση της τοποθεσίας.
        i.

Κεφάλαιο 4 - Υλοποίηση της Διαδικτυακής Εφαρμογής του Έξυπνου Ανελκυστήρα.

Στο τέταρτο κεφάλαιο αναλύεται η υλοποίηση της διαδικτυακής εφαρμογή για τον Έξυπνο Ανελκυστήρα, καλύπτοντας τις τεχνολογίες που χρησιμοποιήθηκαν, την δόμηση, την ενσωμάτωση των εργαλείων που χρησιμοποιήθηκαν και την εξήγηση του κώδικα υλοποίησης.

4.1 Τεχνολογίες που Χρησιμοποιήθηκαν

Ακολουθούν οι βασικές τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της διαδικτυακής εφαρμογής:

- Front-end:
  - React
  - Εξωτερικές βιβλιοθήκες
    - SweetAlert2
    - React-data-table-component:
- Back-end:
  - Nodejs
  - Express
  - Βιβλιοθήκες
    - Nodemon
    - Mongoose
    - Cors
    - Bcrypt
- Βάση δεδομένων:
  - MongoDB
- Πρόσθετα Εργαλεία:
  - Teachable Machine
  - Responsive Voice API
  - Open Weather Map API
  - Web API:
    - Web Speech API
    - Navigator Geolocation

4.2 Δομή Φακέλων

Στη συνέχεια παρουσιάζεται η δομή των φακέλων που αναπτύχθηκε η διαδικτυακή εφαρμογή. Κάθε φάκελος και υποφάκελος συνοδεύονται από μια σύντομη περιγραφή που εξηγεί τον σκοπό ύπαρξη τους:

Ο φάκελος "Client" περιέχει τους υποφακέλους:

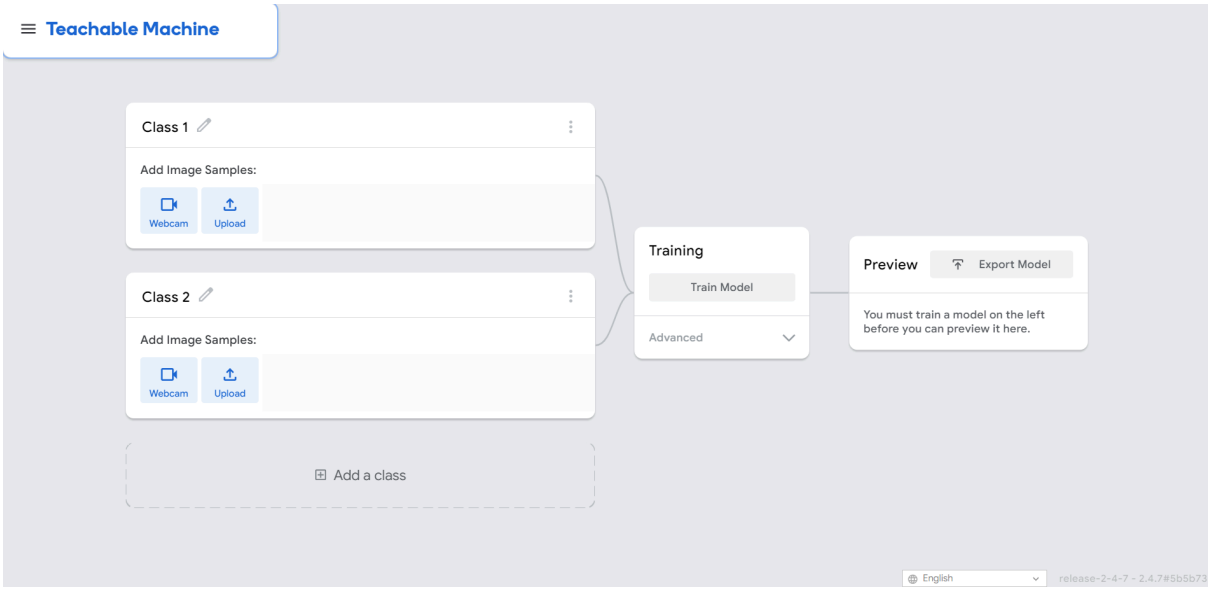
1. Pages: Περιέχει τα τρία αρχεία που αντιπροσωπεύουν τις ιστοσελίδες της διαδικτυακής εφαρμογή:
    a. Admin: Η ιστοσελίδα που ο διαχειριστής είναι υπεύθυνος για τα δεδομένα που θα εμφανίζονται στο Smart-Elevator.
    b. Elevator: Η ιστοσελίδα που εμφανίζεται στην οθόνη του ανελκυστήρα, περιλαμβάνει αναγνώριση προσώπου και ομιλίας.
    c. Login: Η ιστοσελίδα που συνδέεται ο διαχειριστής στην admin ιστοσελίδα.
2. Components: Περιέχει δύο υποφακέλους:
    a. Admin: Περιλαμβάνει τα components που αφορούν την admin ιστοσελίδα.
    b. Elevator: Περιλαμβάνει τα components που αφορούν την σελίδα του ανελκυστήρα.

Ο φάκελος "Server" περιέχει:

1. Models: Είναι φάκελος που περιέχει τα μοντέλα της βάσης.
2. ConnectToDb: Είναι αρχείο που αναλαμβάνει την σύνδεση με την βάση δεδομένων.
3. Server: Είναι το κύριο αρχείο του server που περιλαμβάνει τη λειτουργικότητα του server, συμπεριλαμβάνονται οι διαχειριστικές λειτουργίες και οι επικοινωνία με τη βάση δεδομένων, καθώς και όλα τα endpoints.

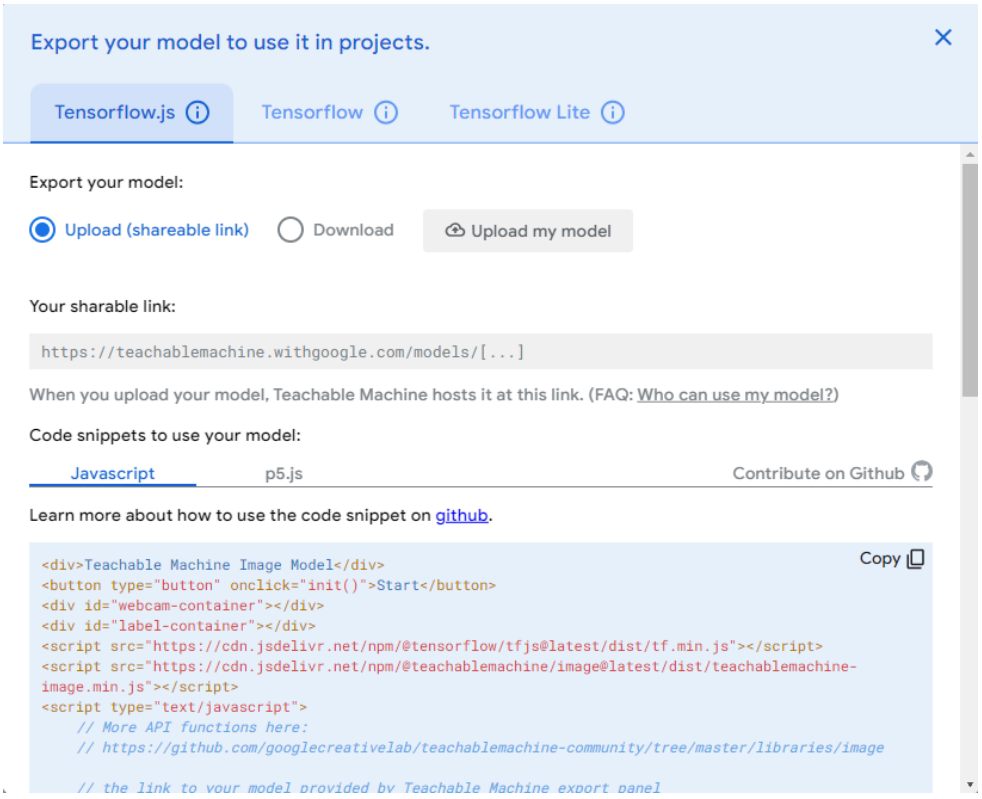
4.3 Ενσωμάτωση του Teachable Machine

Στην συνέχεια, παρουσιάζεται η ενσωμάτωση του Teachable Machine στην διαδικτυακή εφαρμογή. Αρχικά θα χρειαστεί η δημιουργία ενός μοντέλου για εικόνες από την ιστοσελίδα του Teachable Machine.



Εικόνα 7 Teachable Machine train Image Model

Υπάρχει η επιλογή να δοθούν ονόματα στις κλάσεις. Να γίνουν upload οι εικόνες ή να δημιουργηθούν οι εικόνες εκείνη την στιγμή μέσω της Webcam. Όταν γίνει αυτή η διαδικασία πρέπει να γίνει η εκπαίδευση του μοντέλο είτε με τις προεπιλεγμένες επιλογές είτε με τις Advanced ρυθμίσεις. Η εκπαίδευση γίνεται στο πρόγραμμα περιήγησης και όταν τελειώσει η διαδικασία υπάρχει η επιλογή να γίνει Export το μοντέλο.



Εικόνα 8 Teachable Machine Export Model

Στην εξαγωγή του μοντέλου προσφέρονται επιλογές όπως το ανέβασμα για πρόσβαση μέσω συνδέσμου ή το κατέβασμα για τοπική χρήση. Παρέχονται επίσης παραδείγματα κώδικα για ενσωμάτωση στην εφαρμογή.

4.4 Ενσωμάτωση του Web Speech Api

Για τη χρήση του Web Speech API, απαιτείται ένα πρόγραμμα περιήγησης που να το υποστηρίζει, όπως το Google Chrome. Στο εγχειρίδιο υπάρχουν αρκετά παραδείγματα κώδικα για την χρήση του. Η χρήση που έγινε στην υλοποίηση της πτυχιακής φαίνεται και στο κεφάλαιο του Εξήγηση του κώδικα υλοποίησης.

4.5 Ενσωμάτωση του Responsive Voice

Ακολουθεί η ενσωμάτωση του Responsive Voice στη διαδικτυακή εφαρμογή. Αρχικά, απαιτείται η δημιουργία ενός λογαριασμού στο Responsive Voice. Μετά την εγγραφή, παρέχεται ένα CDN, το οποίο συνιστάται να ενσωματωθεί πριν από το κλείσιμο του body tag στο HTML.

```
<script src="https://code.responsivevoice.org/responsivevoice.js?key=key"></script>
```

Στο εγχειρίδιο περιλαμβάνει παραδείγματα κώδικα για την χρήση του. Για παράδειγμα έτσι θα ακουστεί στα ελληνικά το «Γειά σου κόσμε».

```
window.responsiveVoice.speak("Γειά σου κόσμε", "Greek Female");
```

4.6 Εξήγηση του Κώδικα Υλοποίησης

Ακολουθεί μια συνοπτική ανάλυση και εξήγηση του κώδικα που χρησιμοποιείται για την υλοποίηση της διαδικτυακής εφαρμογής, εστιάζοντας στις πιο σημαντικές λειτουργίες της εφαρμογής.

Front-end

Αρχείο App.js:

Η συνάρτηση 'App', χρησιμοποιώντας την βιβλιοθήκη 'react-router-dom', ρυθμίζει την πλοήγηση της διαδικτυακής εφαρμογής με τον καθορισμό των διαδρομών για τα URL paths '/' και '/admin'. Στο path '/' καλείται το component 'Elevator', το οποίο φορτώνεται ως αρχική σελίδα, ενώ το path '/admin' καλεί το component 'Admin', προορισμένο για την διαχείριση της διαδικτυακής εφαρμογής.

```
function App(){

  return (
    <Routes>
      <Route path='/' element={<Elevator />} />
      <Route path='/admin' element={<Admin />} />
    </Routes>
  );
}

export default App;
```

Αρχείο Elevator.jsx:

Αυτό το αρχείο είναι το κομβικό για τον Έξυπνο Ανελκυστήρα. Σε αυτό το αρχείο, βλέπουμε ότι μετά το rendering του DOM, τραβάει τα απαραίτητα δεδομένα από την βάση δεδομένων, ανακτά πληροφορίες για τον καιρό της περιοχής, κάνει το setup της αναγνώριση ομιλίας και της αναγνώρισης προσώπου. Η αναγνώριση προσώπου λειτουργεί διαρκώς, αλλά τα αποτελέσματα χρησιμοποιούνται μόνο όταν ο ανελκυστήρας είναι σταματημένος. Η αναγνώριση ομιλίας επίσης λειτουργεί διαρκώς, όταν ανιχνεύσει τον αριθμό ορόφου ή οποιοδήποτε άλλο σχετικό στοιχείο για τους εγγεγραμμένους ένοικους, τότε δίνει την εντολή για την έναρξη της λειτουργίας του ανελκυστήρα. Ο ανελκυστήρας δεν διαθέτει κανένα υλικό εξοπλισμό, άρα η υλοποίηση του γίνεται αποκλειστικά μέσω λογισμικού και προσομοίωσης.

Παρακάτω υπάρχει ένα 'useEffect', είναι μια συνάρτηση της React και χρησιμοποιείται για την διαχείριση των side effects. Ο πίνακας '[]' στο τέλος του 'useEffect' καθορίζει πότε θα εκτελεστεί. Επειδή ο πίνακας είναι κενός, η συνάρτηση θα εκτελεστεί μια φορά μετά το rendering. Αρχικά η συνάρτηση 'fetchData' ανακτά δεδομένα από τα δύο endpoints, το '/floorWordMappings' και το '/tenants'. Στην συνέχεια, χρησιμοποιεί το geolocation, το οποίο αποτελεί μέρος του Web API, για να επιστρέψει την τοποθεσία. Εάν προκύψει κάποιο

σφάλμα, η προκαθορισμένη τιμή είναι οι συντεταγμένες του Ηρακλείου. Αυτές οι συντεταγμένες χρησιμοποιούνται στο Open Weather Map API για να παρέχουν πληροφορίες για τον καιρό της περιοχής, οι οποίες εμφανίζονται στην οθόνη του έξυπνου ανελκυστήρα. Έπειτα, βλέπουμε το 'setInterval' να καλεί το 'fetchData' κάθε μια ώρα με σκοπό την ενημέρωση των δεδομένων. Τέλος, το 'return' μέσα στο 'useEffect' εκτελείται όταν το component απενεργοποιείται ή ανανεώνεται, για παράδειγμά κατά τη διάρκεια ενός refresh της σελίδας. Χρησιμοποιείται για να διακόψει ομαλά τις διεργασίες που βρίσκονται σε εξέλιξη, όπως το 'setInterval'.

```
  useEffect(() => {
    const getLocation = () => new Promise((resolve, reject) => {
      // Check if geolocation is supported by the browser
      if (!navigator.geolocation) {
        console.error('Geolocation is not supported by this browser.');
        reject(new Error('Geolocation is not supported'));
      }
      // Attempt to get the current position
      navigator.geolocation.getCurrentPosition(
        position => {
          // Format the coordinates to two decimal places
          const latitude = position.coords.latitude.toFixed(2);
          const longitude = position.coords.longitude.toFixed(2);
          resolve({ lat: latitude, lon: longitude });
        },
        error => {
          console.error('Error getting location:', error);
          reject(error);
        }
      );
    });

    async function fetchData() {
      try {
        const floorWordMappingsResponse = await fetch('/floorWordMappings');
        const floorWordMappingsData = await floorWordMappingsResponse.json();

        const tenantsResponse = await fetch('/tenants');
        const tenantsData = await tenantsResponse.json();

        // Get geolocation or use default values (Heraklion location) on error
        const { lat, lon } = await getLocation().catch(() => { return { lat:
35.32, lon: 25.14 } });

        const weatherResponse = await
fetch(`https://api.openweathermap.org/data/2.5/weather?lat=${lat}&lon=${lon}&a
ppid=${process.env.OPEN_WEATHER_API_KEY}&units=metric&lang=el`);
        const weatherData = await weatherResponse.json();
```

```
        setFloorWordMappings(floorWordMappingsData);
        setTenants(tenantsData);
        setWeather(weatherData);
      } catch (error) {
        console.error(error);
      }
    }

    fetchData(); // Fetch data initially

    const intervalId = setInterval(() => {
      fetchData(); // Fetch data every hour
    }, 3600000); // 3600000 milliseconds = 1 hour

    return () => {
      clearInterval(intervalId);
    };

  }, []);
```

Παρακάτω βλέπουμε ένα ‘useEffect’ που αναλαμβάνει την αρχικοποίηση της αναγνώρισης της εικόνας. Ο κενός πίνακας ‘[]’ δηλώνει ότι θα εκτελεστεί μια φορά. Επιπλέον, ο κώδικας φορτώνει ένα μοντέλο αναγνώρισης εικόνας από την υπηρεσία Teachable Machine της Google, με την παροχή της διεύθυνσης URL. Στην συνέχεια, η κάμερα ενεργοποιείται για τη λήψη εικόνων σε πραγματικό χρόνο. Έπειτα, η συνάρτηση loop ανανεώνει τις εικόνες και το μοντέλο δίνει την πρόβλεψή του για την κατηγορία της εικόνας. Αυτή η διαδικασία επαναλαμβάνεται συνεχώς μέσω της αναδρομής που δημιουργεί η συνάρτηση requestAnimationFrame, επιτρέποντας τη διαρκεί ενημέρωση των προβλέψεων.

```
  useEffect(() => {
    async function init() {
      const URL = 'https://teachablemachine.withgoogle.com/models/CO7TKpjqW/';
      const modelURL = URL + 'model.json';
      const metadataURL = URL + 'metadata.json';

      const model = await window.tmImage.load(modelURL, metadataURL);
      modelRef.current = model;
      const maxPredictions = model.getTotalClasses();

      const webcam = new window.tmImage.Webcam(200, 200, true);
      await webcam.setup();
      await webcam.play();
      webcamRef.current = webcam;
      loop();
    }

    async function loop() {
```

```
      if (webcamRef.current) {
        webcamRef.current.update();
        const prediction = await
modelRef.current.predict(webcamRef.current.canvas);
        setPrediction(prediction);
        animationFrameId.current = window.requestAnimationFrame(loop);
      }
    }

    init();

    return () => {
      if (webcamRef.current) {
        webcamRef.current.stop();
      }
      if (animationFrameId.current) {
        window.cancelAnimationFrame(animationFrameId.current);
      }
    };
  }, []);
```

Το παρακάτω ‘useEffect’ διαχειρίζεται τα αποτελέσματα του μοντέλου αναγνώρισης εικόνας. Βλέπουμε ότι ο πίνακας των dependencies δεν είναι κενός [elevatorState, prediction, stopPrediction], οπότε αυτό σημαίνει ότι ο κώδικας θα ξανά εκτελεστεί κάθε φορά που θα αλλάξει κάποια τιμή του πίνακα. Ο κώδικας ξεκινά με δύο ελέγχους, ένας αν ο ανελκυστήρας είναι σταματημένος και ένας αν η πρόβλεψη έχει σταματήσει. Αν περάσει τους ελέγχους βλέπουμε ότι γίνεται ένας βρόχος με τα αποτελέσματα του μοντέλου της αναγνώρισης της εικόνας. Αν το αποτέλεσμα δεν είναι ο άδειος ανελκυστήρας, η πιθανότητα είναι μεγαλύτερη από 0,98, το ονοματεπώνυμο του ενοίκου υπάρχουν στην βάση και ο όροφος δεν υπάρχει στην υπάρχουσα διαδρομή του ανελκυστήρα τότε με την βοήθεια του Responsive Voice το σύστημα ρωτάει τον χρήστη αν θέλει να πάει σε κάποιον όροφο ή στο ισόγειο ανάλογα με το που βρίσκεται. Αυτή η ερώτηση εμφανίζεται επίσης στην οθόνη με το ‘setVoiceResponse’. Στη συνέχεια, γεμίζει την μεταβλητή ‘predictionFloor’ και τερματίζει την διαδικασία ορίζοντας την μεταβλητή ‘stopPredition’ σε true.

```
  useEffect(() => {
    if (elevatorState !== "stop") {
      setStopPrediction(false);
      return;
    }

    if (stopPrediction || startElevator) {
      return;
    }
```

```
    prediction.forEach(predictedTenant => {
      if (predictedTenant.className === "empty elevator" ||
predictedTenant.probability <= 0.98) {
        return;
      }

      const tenant = tenants?.find(tenant => `${tenant.name}
${tenant.surname}` === predictedTenant.className);
      if (!tenant) {
        return;
      }

      const isDestinationFloor = destinationFloors.some(floor_level =>
tenant.floor_level.includes(floor_level));
      if (isDestinationFloor) {
        return;
      }

      const text = tenant.floor_level > currentFloor ?
        `πάτε στον ${tenant.floor_level}ο όροφο;` :
        `πάτε στο ισόγειο;`;

      if (!window.responsiveVoice.isPlaying()) {
        window.responsiveVoice.speak(text, "Greek Female", {
            onend: () => {
                recognitionRef.current.abort();
            }
        });
      }
      setPredictionFloor(tenant.floor_level > currentFloor ?
tenant.floor_level : 0);
      setVoiceResponse({
        entityType: 'elevator',
        textResponse: text
      });
      setStopPrediction(true);
    });

  }, [elevatorState, prediction, stopPrediction]);
```

Ακολουθεί ένα ‘useEffect’ με έναν κενό πίνακα dependencies, πράγμα που σημαίνει ότι θα εκτελεστεί μία φορά μετά το rendering. Ο παρακάτω κώδικας, αναλαμβάνει την ενεργοποίηση και την διαχείριση της αναγνώρισης ομιλίας. Αρχικά παρατηρούμε την αρχικοποίηση το Web Speech API. Εντοπίζουμε την προσθήκη συγκεκριμένων λέξεων στο ‘SpeechGrammarList, βελτιώνοντας έτσι σημαντικά την απόδοση του συστήματος για αυτές

τις λέξεις. Επιπλέον, ρυθμίζεται η γλώσσα στα ελληνικά, η αναγνώριση ομιλίας ρυθμίζεται στο να σταματήσει στην πρώτη παύση της ομιλίας και ότι δεν θα προέρχονται προσωρινά αποτελέσματα κατά την διάρκεια της ομιλίας, αλλά μόνο το τελικό αποτέλεσμα. Έπειτα, βλέπουμε την χρήση διάφορων events. Η εντολή 'start' ενεργοποιεί την αναγνώριση ομιλίας σύμφωνα με τις προηγούμενες ρυθμίσεις. Το 'onresult' ενεργοποιείται όταν ληφθούν αποτελέσματα, καταγράφοντας στην μεταβλητή 'transcript' και προβάλλοντας τα στην οθόνη μέσω της 'setVoiceResponse'. Το 'onerror' ενεργοποιείται σε περίπτωση σφαλμάτων, εμφανίζοντας στην κονσόλα για διαγνωστικούς σκοπούς. Το 'onend' ενεργοποιείται όταν η αναγνώριση ομιλίας σταματάει να λειτουργεί, τότε την ξανά ενεργοποιεί, εξασφαλίζοντας έτσι την συνεχή λειτουργία. Το 'return' χρησιμοποιείται για να διακόψει ομαλά την αναγνώριση ομιλίας, για παράδειγμα κατά τη διάρκεια ενός refresh της σελίδας.

```
useEffect(() => {
    const SpeechRecognition = window.SpeechRecognition ||
window.webkitSpeechRecognition;
    const SpeechGrammarList = window.SpeechGrammarList ||
window.webkitSpeechGrammarList;

    const recognition = new SpeechRecognition();
    const words = ["ισόγειο","1ος όροφος", "2ος όροφος", "3ος όροφος", "4ος
όροφος", "5ος όροφος"];
    const grammar = `#JSGF V1.0; grammar words; public <word> = ${words.join('
| ')} ;`;
    const speechRecognitionList = new SpeechGrammarList();
    speechRecognitionList.addFromString(grammar, 1);

    recognition.grammars = speechRecognitionList;
    recognition.lang = 'el-GR';
    recognition.continuous = false;
    recognition.interimResults = false;

    recognitionRef.current = recognition;

    recognitionRef.current.start();

    recognitionRef.current.onresult = (event) => {
      const transcript = event.results[event.results.length -
1][0].transcript;
      setVoiceResponse({
        entityType: 'human',
        textResponse: transcript
      });
      setTranscript({ value: transcript, updated: Date.now() });
    };

    recognitionRef.current.onerror = (event) => {
```

```
      console.warn(event.error);
    };

    //This will be triggered when it stops listening.
    recognitionRef.current.onend = () => {
      console.warn("Onend: restart speech");
      if (recognitionRef.current) {
        recognitionRef.current.stop();
        recognitionRef.current.start();
      }
    };

    return () => {
      if (recognitionRef.current) {
        recognitionRef.current.stop();
      }
    };
  }, []);
```

Το παρακάτω ‘useEffect’ εκτελείται κάθε φορά που αλλάζει η τιμή του ‘transcript’, αυτό το καταλαβαίνουμε στην δήλωση [transcript]. Όπως είδαμε, η τιμή του ‘transcript’ ανανεώνεται όταν λαμβάνουμε κάποιο αποτέλεσμα από την αναγνώριση ομιλίας. Αυτό το ‘useEffect’ έχει αναλάβει την διαχείριση των αποτελεσμάτων από την αναγνώριση ομιλίας και αν περάσουν τους ελέγχους προσθέτει σε ένα πίνακα ‘destinationsFloors’ τον προορισμό των ορόφων. Το συγκεκριμένο ‘useEffect’ είναι υπεύθυνο για την διαχείριση των αποτελεσμάτων της αναγνώρισης ομιλίας. Αν τα αποτελέσματα περάσουν τους απαιτούμενους ελέγχους, τότε θα προστεθούν οι όροφοι στον πίνακα με την ονομασία ‘destinationsFloors’, ο οποίος καθορίζει τους ορόφους όπου θα μετακινηθεί ο ανελκυστήρας. Η διαδικασία ξεκινάει ελέγχοντας αν η μεταβλητή ‘transcript’ δεν είναι κενή. Έπειτα, καλείται η συνάρτηση ‘findFloorLevel’, όπου σε επιτυχία επιστρέφει τον όροφο και σε αποτυχία null. Σαν παράμετρο παίρνει το αποτέλεσμα της αναγνώρισης ομιλίας και το μετατρέπει σε πεζά γράμματα. Στην συνέχεια, πραγματοποιείται μια σειρά ελέγχων. Αν το αποτέλεσμα της αναγνώρισης ομιλίας περιέχει αρνητικές λέξεις, η διαδικασία διακόπτεται. Αν το αποτέλεσμα αντιστοιχεί σε κάποιο όροφο του ανελκυστήρα πχ πρώτος, ή 2ος Όροφος κτλ. τότε επιστρέφει τον αριθμό του ορόφου. Αν το αποτέλεσμα αντιστοιχεί στο όνομα ή στο επίθετο ή στο επάγγελμα των ένοικων που έχουν δώσει τα στοιχεία τους, τότε επιστρέφει τον αριθμό του ορόφου. Όπως είδαμε και παραπάνω, αν η αναγνώριση εικόνας αναγνώριση κάποιο πρόσωπο, τότε το σύστημα ρωτά αν επιθυμεί να πάει στον συγκεκριμένο όροφο, τότε εδώ βλέπουμε το έλεγχο αν έχει βρεθεί κάποιος από τους εγγεγραμμένους ένοικους και η απάντηση του είναι καταφατική τότε επιστρέφει η

συνάρτηση επιστρέφει τον όροφο που έχει δηλώσει. Τέλος, αν επιστραφεί κάποιος όροφος και δεν υπάρχει ήδη στο δρομολόγιο του ανελκυστήρα τότε τον προσθέτει.

```
   useEffect(() => {

    const findTenantFloorLevel = (sentence, tenantData) => {

      const match = tenantData.find(tenant => {
        const tenantName = tenant.name.toLowerCase();
        const tenantSurname = tenant.surname.toLowerCase();
        const tenantCategory = tenant.category.toLowerCase();

        return (
          sentence.includes(tenantName) ||
          sentence.includes(tenantSurname) ||
          sentence.includes(tenantCategory)
        );
      });

      return match ? match.floor_level : null;
    }


    const findFloorLevel = (speechResult) => {

      const lowercaseSentence = speechResult.toLowerCase();

      const aggreeWords = ["ναι", "ναι πάω"];

      const negationWords = ["όχι", "δεν πάω"];
      if (negationWords.some(negation =>
lowercaseSentence.includes(negation))) {
        return null;
      }

      const matchedFloor = floorWordMappings.find(item =>
        item.words.some(word =>
lowercaseSentence.includes(word.toLowerCase()))
      );
      if (matchedFloor) {
        return matchedFloor.floor_level;
      }


      const tenantFloorLevel = findTenantFloorLevel(lowercaseSentence,
tenants);
      if (tenantFloorLevel !== null) {
        return tenantFloorLevel;
      }
```

```
      if (stopPrediction && predictionFloor !== null) {
        if (aggreeWords.some(agreeWord =>
lowercaseSentence.includes(agreeWord))) {
          return predictionFloor;
        }
      }

      return null;

    }


    if (transcript?.value?.length > 0) {
      const floorLevel = findFloorLevel(transcript.value);
      if (floorLevel !== null && currentFloor !== floorLevel &&
!destinationFloors.includes(floorLevel)) {
        setDestinationFloors((prevDestinationFloors) =>
[...prevDestinationFloors, floorLevel]);
      }
    }

  }, [transcript]);
```

Το παρακάτω ‘useEffect’ εκτελείται κάθε φορά που αλλάζει η τιμή του ‘startElevator’ ή του ‘destinationsFloors’, αυτό το καταλαβαίνουμε από την δήλωση [startElevator, destinationFloors]. Έχει αναλάβει την έναρξη του ανελκυστήρα όταν είναι σταματημένος και ο πίνακας με του όροφος του δεν είναι άδειος. Ακόμη, με το ‘responsiveVoice’ ανακοινώνει ότι ξεκινάει ο ανελκυστήρας.

```
  useEffect(() => {

    if (startElevator === false && destinationFloors.length > 0) {
      setStartElevator(true);
      if (!window.responsiveVoice.isPlaying()) {
        window.responsiveVoice.speak("ξεκινάμε", "Greek Female", {
          onend: () => {
            recognitionRef.current.abort();
          }
        });
      }
    }

  }, [startElevator, destinationFloors]);
```

Ακολουθεί ένα ‘useEffect’ που θα εκτελεσθεί κάθε φορά που γίνεται αλλαγή τιμής στο ‘startElevator’ ή στο ‘resumeElevator’, αυτό το διακρίνουμε από τον πίνακα των dependencies [startElevator, resumeElevator]. Αυτό το ‘useEffect’ είναι η προσομοίωση, της εναλλαγής των ορόφων του ανελκυστήρα. Ο κάθε όροφος αλλάζει σε 3 δευτερόλεπτα, μετά την ανακοίνωση της άφιξης περιμένει επιπλέον 3 δευτερόλεπτα. Η συνάρτηση ‘changeFloor’ εκτελείται μόνο όταν η μεταβλητή ‘startElevator’ είναι true. Μέσα στην συνάρτηση βλέπουμε ένα setTimeout που έχει δηλωθεί ότι θα τρέξει μετά από 3 δευτερόλεπτα. Όταν τρέξει θα δούμε ότι υπάρχει μία δομή ελέγχου. Αν ο τωρινός όροφος ‘currentFloor’ είναι ο όροφος του πρώτου προορισμού ‘destinationFloors[0]’ τότε τον αφαιρεί από τον πίνακα, ενημερώνει την κατάσταση του ανελκυστήρα σε “stop”, το ‘responsiveVoice’ ανακοινώνει «φτάσαμε», το’setVoiceResponse’ θέτει την μεταβλητή για να εμφανιστή στην οθόνη, η μεταβλητή ‘startElevator’ γίνεται ‘false’ και το ‘await wait();’ περιμένει 3 δευτερόλεπτα ώστε να γίνει η προσομοίωση της στάσης για να κατέβει κάποιος. Αν ο τωρινός όροφος ‘currentFloor’ είναι μικρότερος από τον πρώτο όροφο του προορισμού ‘destinationFloors’ τότε αυξάνει τον τωρινό όροφο κατά 1, ενημερώνει την κατάσταση του ανελκυστήρα σε ‘up’ και έπειτα δηλώνει ότι θα συνεχιστεί η διαδρομή του ανελκυστήρα ‘setResumeElevator’ (ξανά καλείται το ‘useEffect’). Αν ο τωρινός όροφος ‘currentFloor’ είναι μεγαλύτερος από τον πρώτο όροφο του προορισμού ‘destinationFloors’ τότε μειώνει τον τωρινό όροφο κατά 1, ενημερώνει την κατάσταση του ανελκυστήρα σε ‘down’ και έπειτα δηλώνει ότι θα συνεχιστεί η διαδρομή του ανελκυστήρα ‘setResumeElevator’ (ξανά καλείται το ‘useEffect’). Τέλος, το ‘return’ καθαρίζει τα timeouts.

```
  useEffect(() => {
    let changeFloorTimeout;
    let waitTimeout;

    const wait = () => new Promise(resolve => {
      waitTimeout = setTimeout(resolve, 3000);
    });

    const changeFloor = () => {
      //change floor every 3 sec
      changeFloorTimeout = setTimeout(async () => {
        if (currentFloor === destinationFloors[0]) {
          //removes the first element
          setDestinationFloors(prevNextFloors => prevNextFloors.slice(1));
          setElevatorState('stop');
          let text = currentFloor === 0 ? "φτάσαμε στο ισόγειο" : (() => {
            const floorWordMap = floorWordMappings.find(row => row.floor_level
=== currentFloor);
            return `φτάσαμε στον ${floorWordMap.words[2]} όροφο`;
```

```
        })();
        if (!window.responsiveVoice.isPlaying()) {
          window.responsiveVoice.speak(text, "Greek Female", {
            onstart: () => {
              setVoiceResponse({
                entityType: 'elevator',
                textResponse: text
              });
            },
            onend: async() => {
                recognitionRef.current.abort();
                await wait();
                setStartElevator(false);
            }
          });
        }
      } else if (currentFloor < destinationFloors[0]) {
        setCurrentFloor(prevFloor => (prevFloor + 1));
        setElevatorState('up');
        setResumeElevator((prevCount) => (prevCount + 1) % 6);
      } else if (currentFloor > destinationFloors[0]) {
        setCurrentFloor(prevFloor => (prevFloor - 1));
        setElevatorState('down');
        setResumeElevator((prevCount) => (prevCount + 1) % 6);
      }

    }, 3000);
  }


  if (startElevator) {
    changeFloor();
  }

  return () => {
    clearTimeout(changeFloorTimeout);
    clearTimeout(waitTimeout);
  };

}, [startElevator, resumeElevator]);
```

Back-end

Αρχικά υπάρχουν τρία αρχεία το floorWordMappings.js, tenants.js και user.js που είναι μοντέλα της βάσης και στο κάθε αρχείο υπάρχει η δήλωση του σχήματος.

Αρχείο floorWordMappings.js:

```
const mongoose = require('mongoose');
```

```
const floorWordMappingsSchema = new mongoose.Schema({
  floor_level: Number,
  words: [String],
}, { versionKey: false });

const FloorWordMappings = mongoose.model('floor-word-mappings',
floorWordMappingsSchema);

module.exports = FloorWordMappings;
```

Αρχείο tenants.js:

```
const mongoose = require('mongoose');

const tenantsSchema = new mongoose.Schema({
    _id: { type: mongoose.Schema.Types.ObjectId, auto: true },
    name: String,
    surname: String,
    category: String,
    floor_level: Number,
    teachable_machine: Boolean
}, { versionKey: false });

const Tenants = mongoose.model('tenants', tenantsSchema);

module.exports = Tenants;
```

Αρχείο user.js

```
const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({
    username: String,
    password: String
});

const User = mongoose.model('users', userSchema);

module.exports = User;
```

Ακόμη υπάρχει το αρχείο που ένα αρχείο το connectToDb.js που αναλαμβάνει την σύνδεση με τη βάση.

Αρχείο connectToDb.js:

```
const mongoose = require('mongoose');

const connectToDb = async () => {
    try {
        await mongoose.connect('mongodb://127.0.0.1/smart-elevator, {
useNewUrlParser: true, useUnifiedTopology: true });
    } catch (err) {
        console.log(err);
    }
}

module.exports = connectToDb;
```

Αρχείο server.js

Αυτό το αρχείο είναι ένα από τα πιο σημαντικά στο φάκελο back-end. Αρχικά γίνονται δηλώσεις για τα πακέτα “express”, “bcrypt”, “mongoose” καιι “cors.

```
const express = require('express');
const bcrypt = require('bcrypt');
const mongoose = require('mongoose');

const cors = require('cors');
const app = express();
```

Μετά γίνονται οι δηλώσεις μεταβλητών με τα παραπάνω αρχεία.

```
const connectToDb = require('./connectToDb');

const User = require('./models/user');
const FloorWordMappings = require('./models/floorWordMappings');
const Tenants = require('./models/tenants');
```

Παρακάτω γίνεται η σύνδεση βάση MongoDB και τρέχει η Server.

```
connectToDb();

mongoose.connection.once('open', () => {
    console.log('Connected to MongoDB');
    app.listen(5000, () => { console.log("Server started on port 5000") });
})

mongoose.connection.on('error', err => {
    console.log(err);
})
```

Τέλος, ορίζονται διάφορα endpoints για την διαχείριση χρηστών, ορόφων και ενοικιαστών και περιλαμβάνουν λειτουργίες CRUD (Δημιουργία, Ανάγνωση, Ενημέρωση, Διαγραφή).

```
// Manage Floor Endpoints

app.get('/floorWordMappings', async (req, res) => {
    try {
        // Fetch all documents from the collection
        const mappings = await FloorWordMappings.find({});
        res.json(mappings);
    } catch (err) {
        console.error('Error fetching data:', err);
        res.status(500).send('Internal Server Error');
    }
});

app.post('/addFloorWordMappings', async (req, res) => {
    const { floor_level, words } = req.body;

    const newFloor = new FloorWordMappings({
        floor_level: floor_level,
        words: words
    });

    try {
        await newFloor.save();
        res.status(200).send('Floor added successfully');
    } catch (err) {
        console.error(err);
        res.status(500).send('Error saving Floor');
    }

})

app.put('/updateFloorWordMapping/:id', async (req, res) => {
    const tenantId = req.params.id;
    const updatedData = req.body;

    try {
        const updatedFloorWordMapping = await
FloorWordMappings.findByIdAndUpdate(tenantId, updatedData, {
            new: true,
        });

        if (!updatedFloorWordMapping) {
            res.status(404).send('Floor not found');
        } else {
            res.status(200).json(updatedFloorWordMapping);
        }
    } catch (err) {
        console.error(err);
```

```
        res.status(500).send('Error updating Floor');
    }
});


app.delete('/floorWordMapping/:id', async (req, res) => {
    const floorsId = req.params.id;

    try {
        const deletedFloor = await
FloorWordMappings.findByIdAndRemove(floorsId);

        if (!deletedFloor) {
            res.status(404).send('Floor not found');
        } else {
            res.status(200).send('Floor deleted successfully');
        }
    } catch (err) {
        console.error(err);
        res.status(500).send('Error deleting Floor');
    }
});


//Manage Tenants Endpoints

app.get("/tenants", async (req, res) => {
    try {
        const tenants = await Tenants.find({ category: { $ne: 'admin' } });
        res.json(tenants);
    } catch (err) {
        console.error('Error fetching data:', err);
        res.status(500).send('Internal Server Error');
    }
});

app.post('/addTenant', async (req, res) => {
    const { name, surname, category, floor_level, teachable_machine } =
req.body;

    const newTenant = new Tenants({
        name: name,
        surname: surname,
        category: category,
        floor_level: floor_level,
        teachable_machine: teachable_machine
    });

    try {
        await newTenant.save();
        res.status(200).send('Tenant added successfully');
```

```
    } catch (err) {
        console.error(err);
        res.status(500).send('Error saving Tenant');
    }
});

app.put('/updateTenant/:id', async (req, res) => {
    const tenantId = req.params.id;
    const updatedData = req.body;

    try {
        const updatedTenant = await Tenants.findByIdAndUpdate(tenantId,
updatedData, {
            new: true,
        });

        if (!updatedTenant) {
            res.status(404).send('Tenant not found');
        } else {
            res.status(200).json(updatedTenant);
        }
    } catch (err) {
        console.error(err);
        res.status(500).send('Error updating tenant');
    }
});

app.delete('/tenants/:id', async (req, res) => {
    const tenantId = req.params.id;

    try {
        const deletedTenant = await Tenants.findByIdAndRemove(tenantId);

        if (!deletedTenant) {
            res.status(404).send('Tenant not found');
        } else {
            res.status(200).send('Tenant deleted successfully');
        }
    } catch (err) {
        console.error(err);
        res.status(500).send('Error deleting tenant');
    }
});

// Login

app.post("/login", async (req, res) => {
    const { username, password } = req.body;

    try {
        // Find the user in the database
```

```
        const user = await User.findOne({ username });

        if (!user) {
            res.status(401).json({ error: 'Invalid credentials' });
        } else {
            // Compare the hashed password with the provided password
            const passwordMatch = await bcrypt.compare(password,
user.password);

            if (passwordMatch) {
                res.json({ message: 'Login successful' });
            } else {
                res.status(401).json({ error: 'Invalid credentials' });
            }
        }
    } catch (err) {
        console.error(err);
        res.status(500).json({ error: 'An error occurred' });
    }
});
```

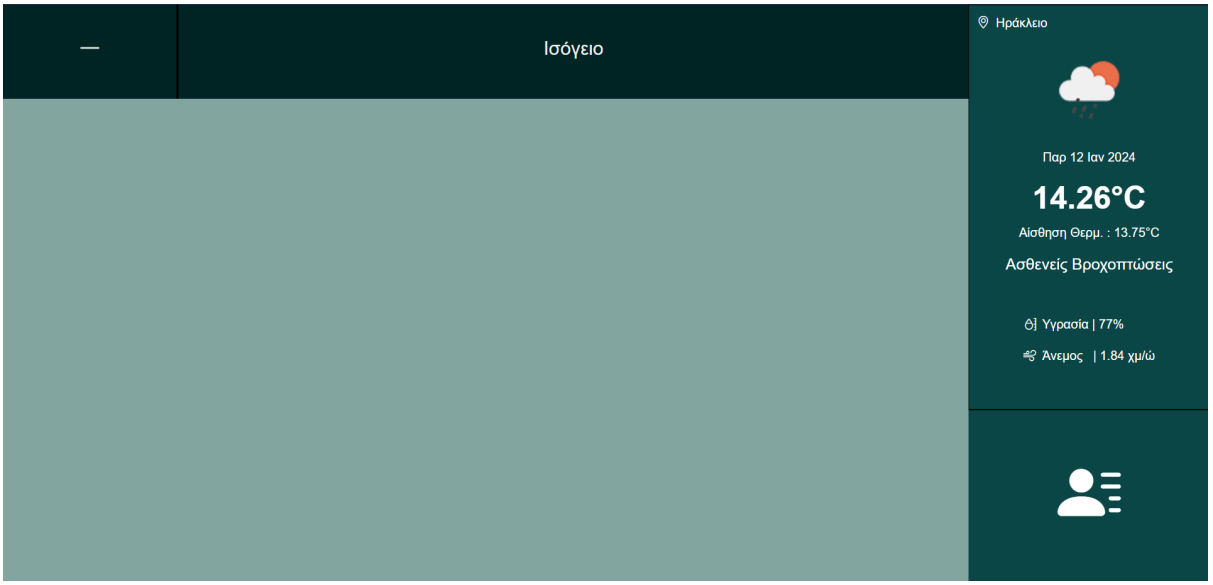
Κεφάλαιο 5 - Χρήση του Συστήματος

Σε αυτό το κεφάλαιο αναφέρονται όλα τα σενάρια συναντιούνται στην διαδικτυακή εφαρμογή του συστήματος.

5.1 Ιστοσελίδα χρήστη

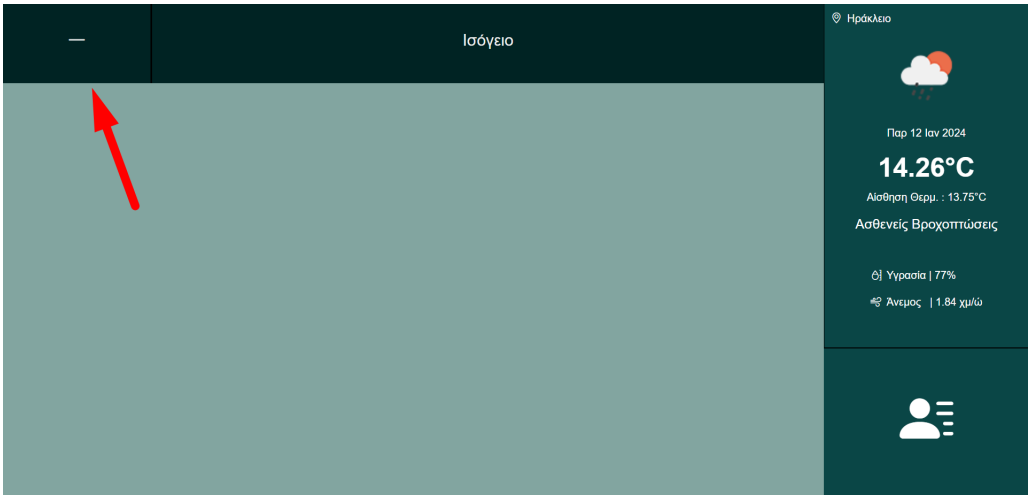
Παρακάτω είναι η ιστοσελίδα που θα βλέπει ο χρήστης όταν μπαίνει στο ανελκυστήρα.

Η Εικόνα 9 είναι η αρχική οθόνη του συστήματος και είναι ορατή σε κάθε χρήστη που εισέρχεται στον ανελκυστήρα.



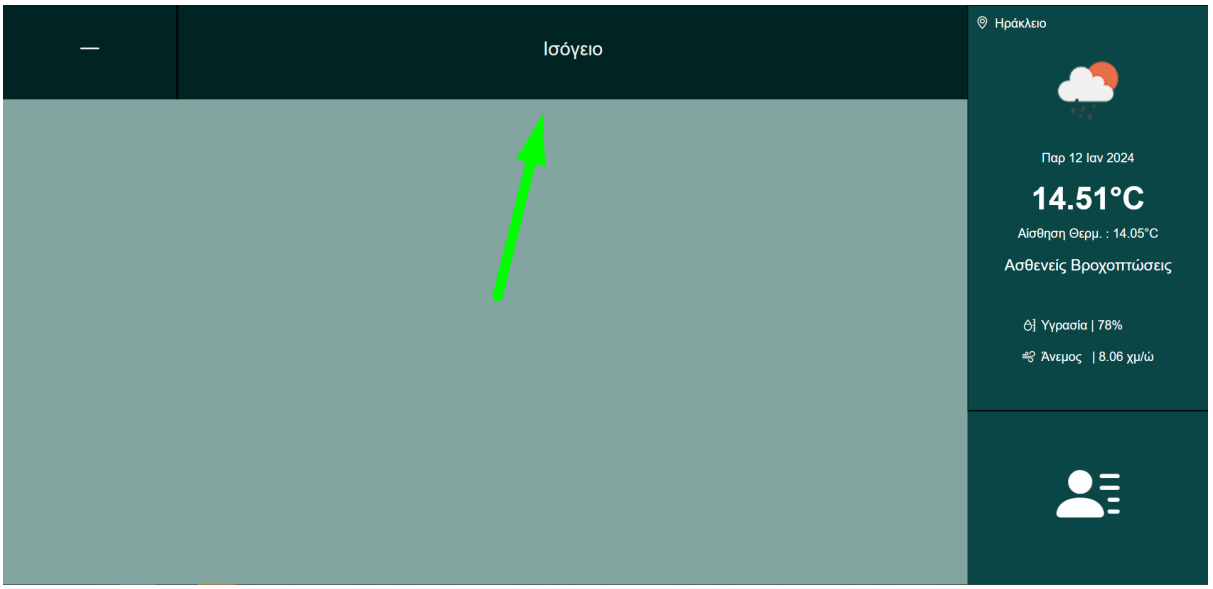
Εικόνα 9 Αρχική οθόνη του συστήματος

Η εικόνα 10 δείχνει ένα κόκκινο βέλος να υποδεικνύει ένα πλαίσιο που εμφανίζει την κατάσταση του ανελκυστήρα. Συγκεκριμένα, το σύμβολο «—» που είναι αναγραμμένο στο πλαίσιο σημαίνει ότι ο ανελκυστήρας είναι σταματημένος. Το σύμβολο «□» δηλώνει ότι ο ανελκυστήρας ανεβαίνει όροφο, ενώ το σύμβολο «□» ότι κατεβαίνει όροφο.



Εικόνα 10 Κατάσταση του ανελκυστήρα

Η εικόνα 11 υποδεικνύει με ένα πράσινο βέλος το πλαίσιο που εμφανίζει τον όροφο του ανελκυστήρα.



Εικόνα 11 Όροφος του ανελκυστήρα

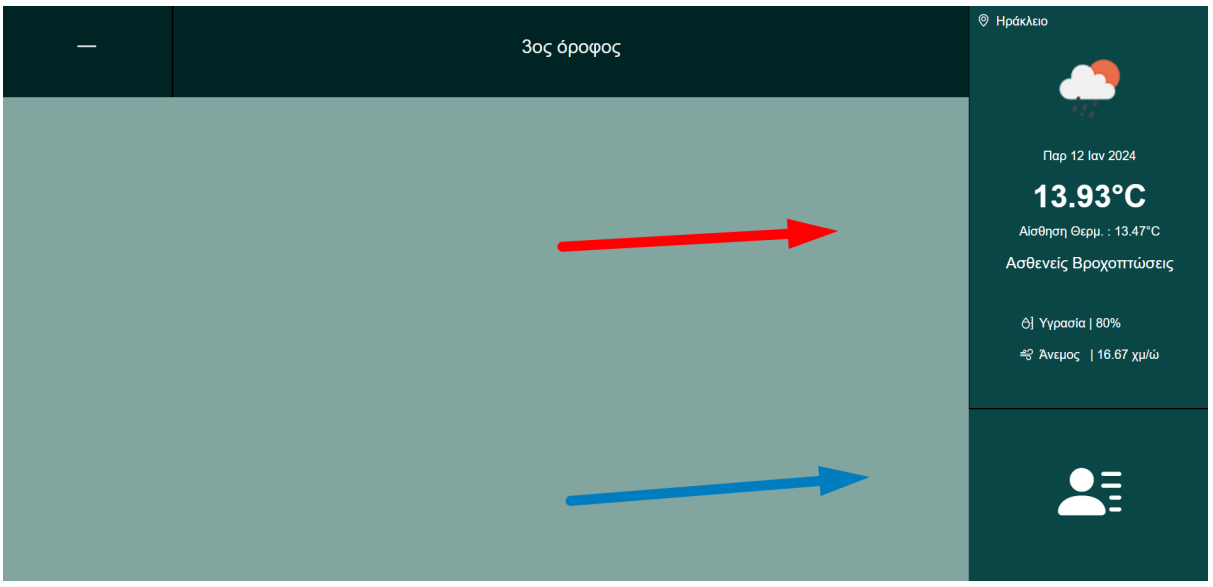
Η εικόνα 12 υποδεικνύει με κόκκινο βέλος το πλαίσιο που εμφανίζεται η φωνητική εντολή που έδωσε ο χρήστης. Ακόμη παρατηρούμε πάνω δεξιά ότι ο ανελκυστήρας ανεβαίνει.



Εικόνα 12 Φωνητική εντολή

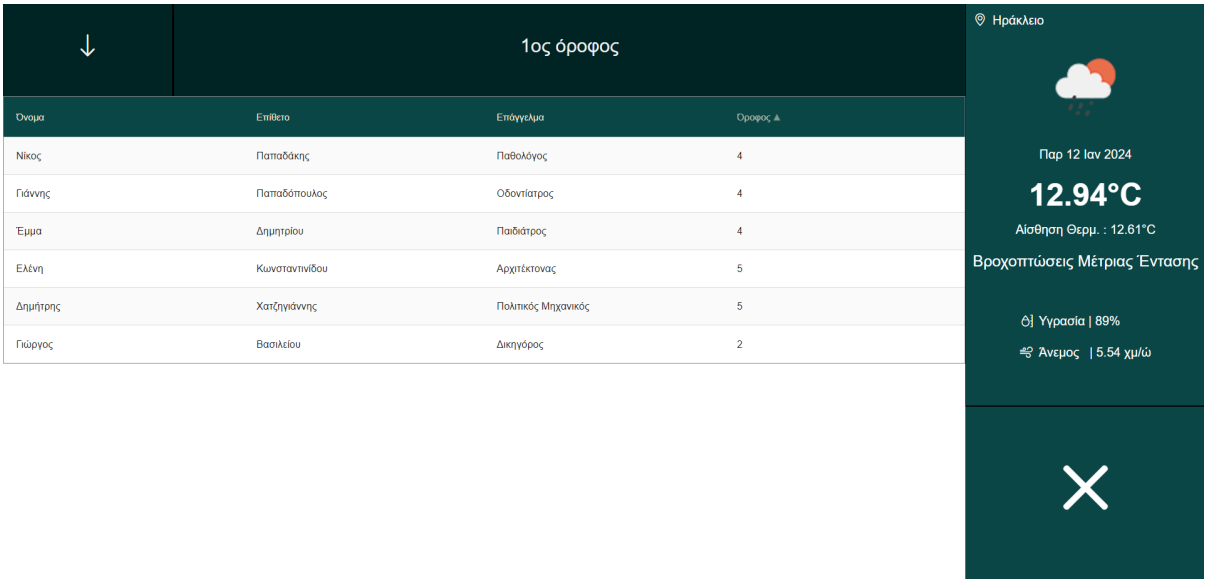
Η εικόνα 13 υποδεικνύει με το κόκκινο βέλος το ένα πλαίσιο που εμφανίζονται διάφορες πληροφορίες όπως η τοποθεσία του ανελκυστήρα, η τρέχουσα ημερομηνία, μια εικόνα που συμβολίζει την ημέρα και την βροχή, καθώς και επιπλέον στοιχεία για την θερμοκρασία που λαμβάνονται από το OpenWeatherMap API.

Το μπλε βέλος στην εικόνα υποδεικνύει το πλαίσιο που αποτελεί ένα κουμπί. Όπου όταν πατηθεί, εμφανίζεται ένας πίνακας με διάφορες πληροφορίες σχετικές με τους ένοικους.



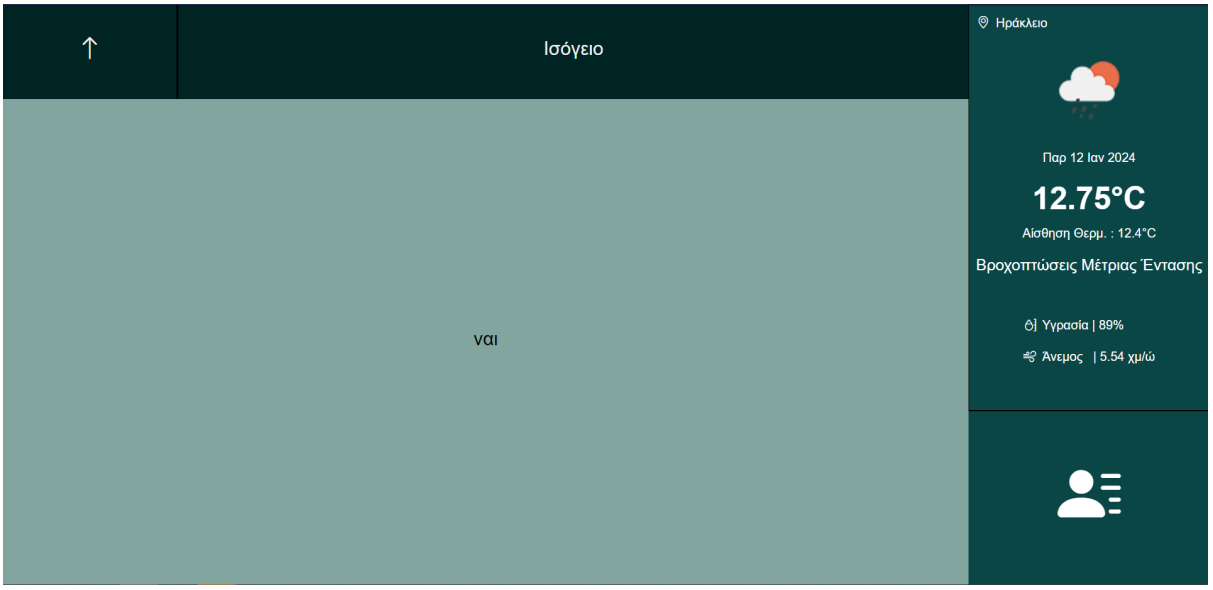
Εικόνα 13 Ενημερωτικές Πληροφορίες

Στην εικόνα 14, φαίνεται ο πίνακας με τους ενοίκους που έχουν δώσει τη συγκατάθεση τους, να παρουσιαστούν τα στοιχεία Όνομα, Επίθετο, Επάγγελμα και Όροφος. Αυτά τα πεδία είναι διαμορφωμένα ώστε να επιτρέπουν την ταξινόμηση των πληροφοριών, είτε σε αύξουσα είτε σε φθίνουσα σειρά Επιπλέον, όποιος θέλει να επιστρέψει στην αρχική θα πρέπει να πατήσει το Χ κουμπί. Ακόμη, στο επάνω δεξιό μέρος της εικόνας, φαίνεται ότι ο ανελκυστήρας κατεβαίνει όροφο.



Εικόνα 14 Πίνακας ενοίκων

Στην εικόνα 15 βλέπουμε την περίπτωση γίνεται η αναγνώριση του προσώπου ενός ένοικου. Τότε γίνεται από το σύστημα η ερώτηση άμα κατευθύνεται στο τάδε όροφο, ο χρήστης απαντάει καταφατικά (ναι) και ο ανελκυστήρας κατευθύνεται σε αυτό τον όροφο. Το βλέπουμε και από το βέλος πάνω αριστερά.

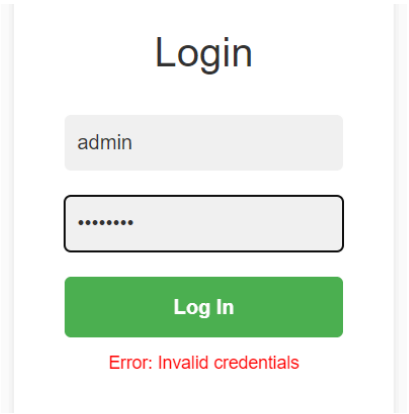


Εικονα 15 Αναγνωριση Προσωπου

5.2 Ιστοσελίδα του Διαχειριστή

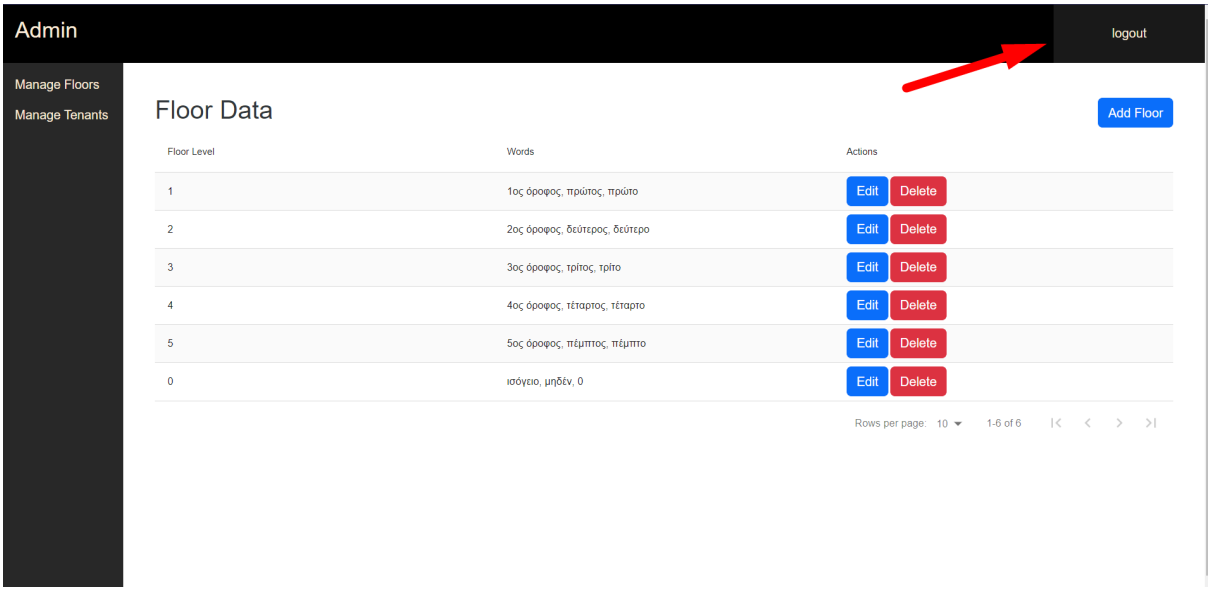
Ακολουθεί η ιστοσελίδα του διαχειριστή. Κατά την πληκτρολόγηση αυτού του URL /admin, ο χρήστης οδηγείται αυτόματα στη φόρμα σύνδεσης. Αφού εισάγει επιτυχώς τον κωδικό πρόσβασης, εμφανίζεται η διεπαφή του διαχειριστή, όπου μπορεί να επεξεργαστεί τις πληροφορίες του ανελκυστήρα.

Στην εικόνα 16 φαίνεται η διεπαφή σύνδεσης όπου οι διαχειριστές πραγματοποιούν την είσοδο στο σύστημα διαχείρισης.



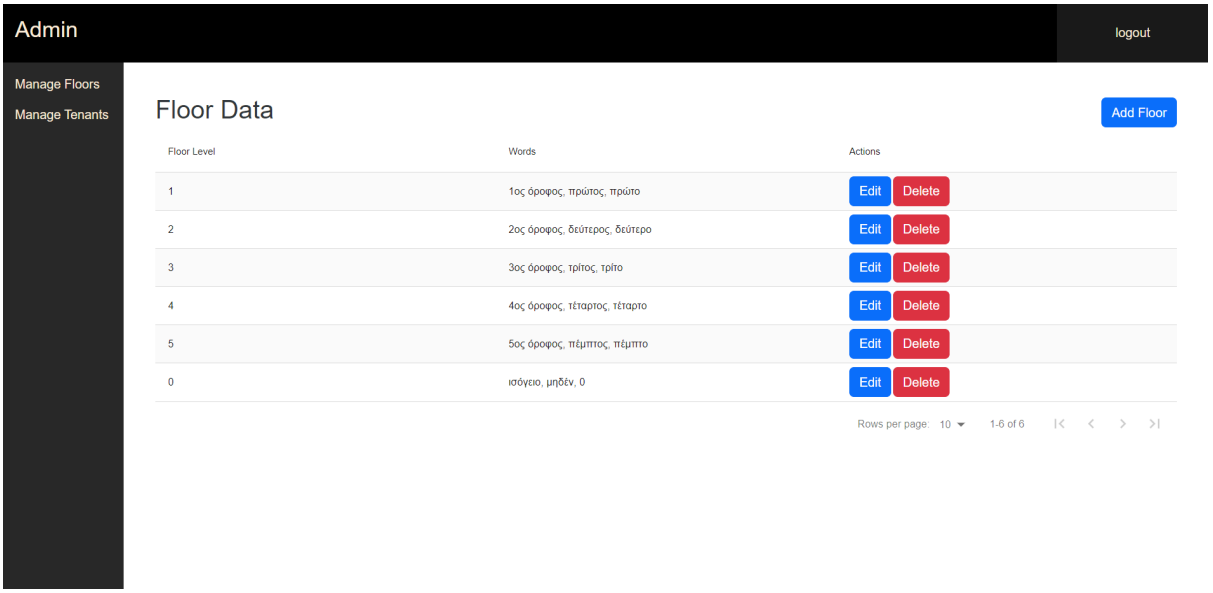
Εικονα 16 Συνδεση

Στην εικόνα 17 φαίνεται η διεπαφή του Admin και παρατηρούμε πάνω δεξιά ένα κουμπί για αποσύνδεση.



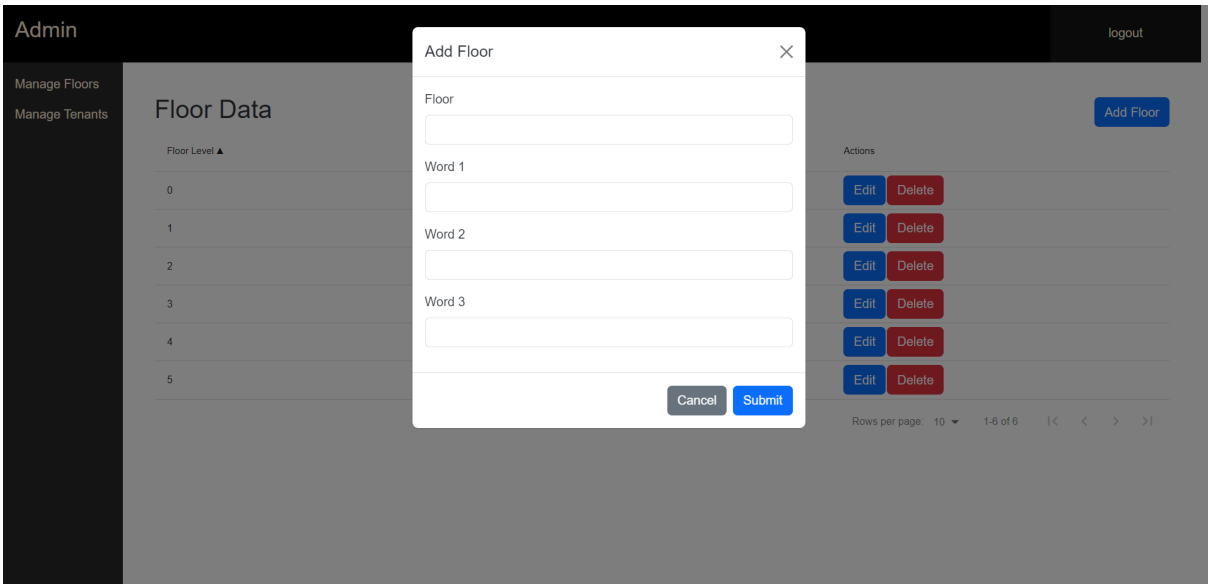
Εικόνα 17 Admin Logout

Στην εικόνα 18 βλέπουμε τον πίνακα των ορόφων με 3 στήλες. Η πρώτη δείχνει τους ορόφους της πολυκατοικίας, η δεύτερη δείχνει τις λέξεις που συνδέονται με τον όροφο. Όπου, οι παρακάτω λέξεις είναι αναγνωρίσιμες από το σύστημα φωνητικών εντολών και χρησιμοποιούνται για την αυτόματη μεταφορά στους επιθυμητούς ορόφους. Το τρίτο είναι οι ενέργειες που μπορείς να κάνεις.



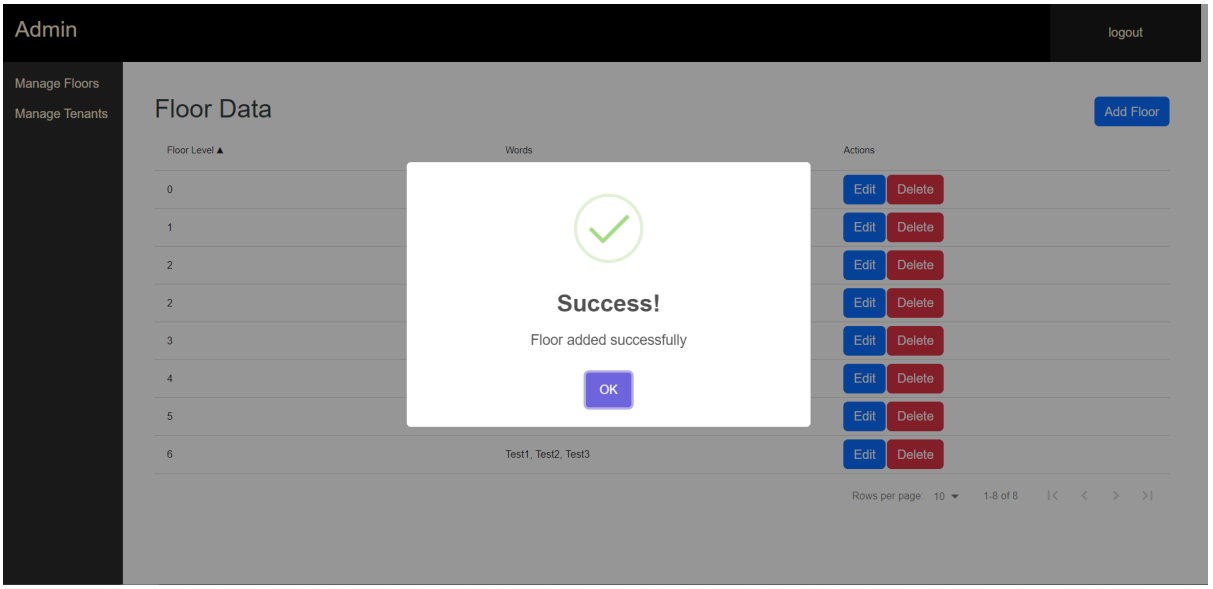
Εικόνα 18 Floor Data

Στην εικόνα 19 βλέπουμε ότι όταν πατήσεις το κουμπί Add Floor, εμφανίζεται ένα modal που περιέχει 4 πεδία (Όροφος, Word1, Word2, Word3) που ο διαχειριστής μπορεί να συμπληρώσει όταν χρειάζεται να προσθέσει ένα όροφο.

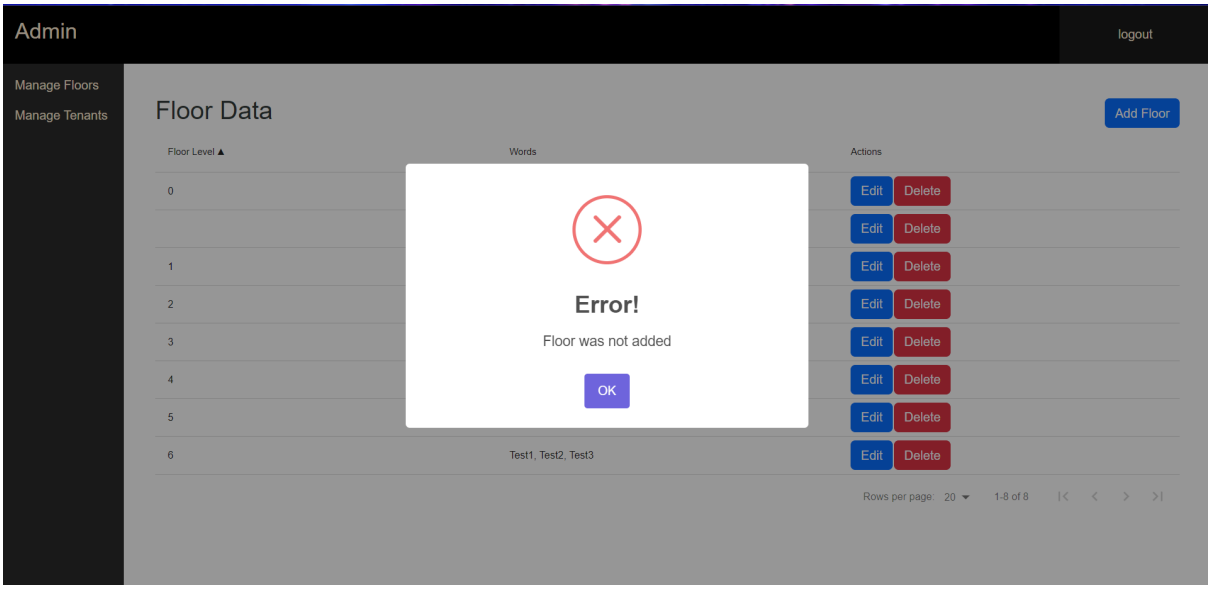


Εικόνα 19 Add Floor

Στην εικόνα 20 βλέπουμε το μήνυμα επιτυχίας και στην εικόνα 19 το μήνυμα αποτυχίας.

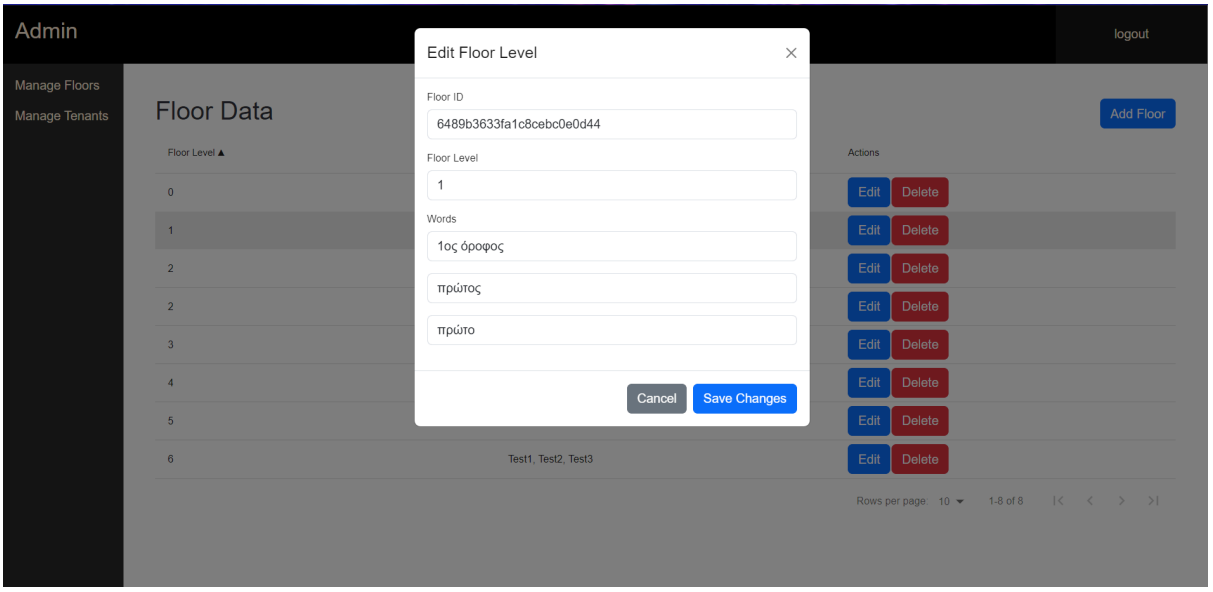


Εικόνα 20 Success message



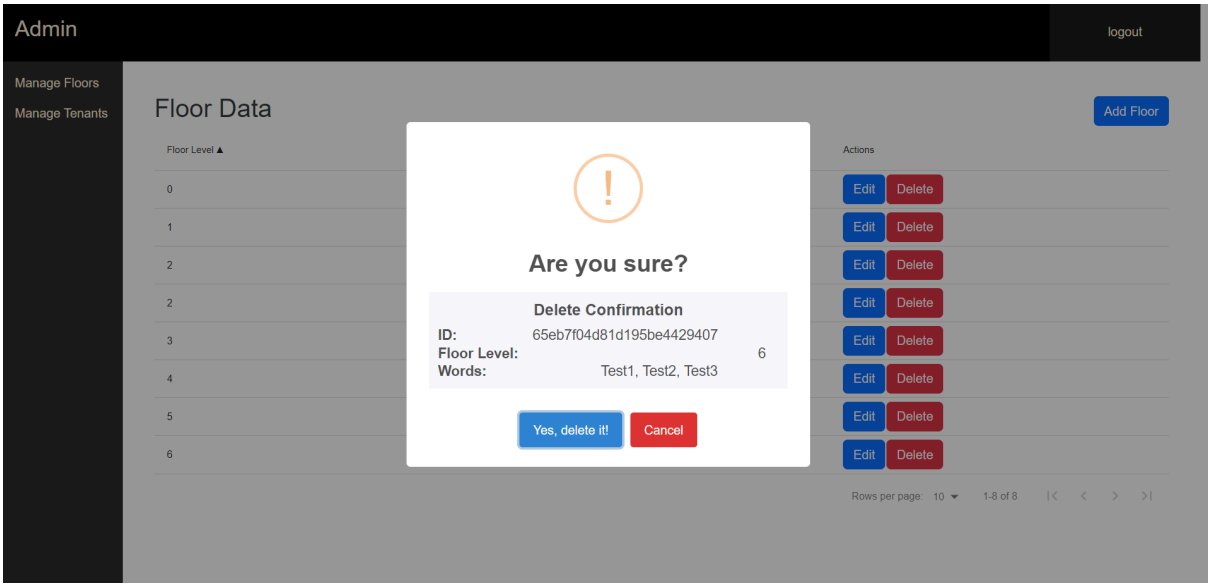
Εικόνα 21 Failed Message

Στην εικόνα 22 βλέπουμε την επεξεργασία ενός ορόφου.



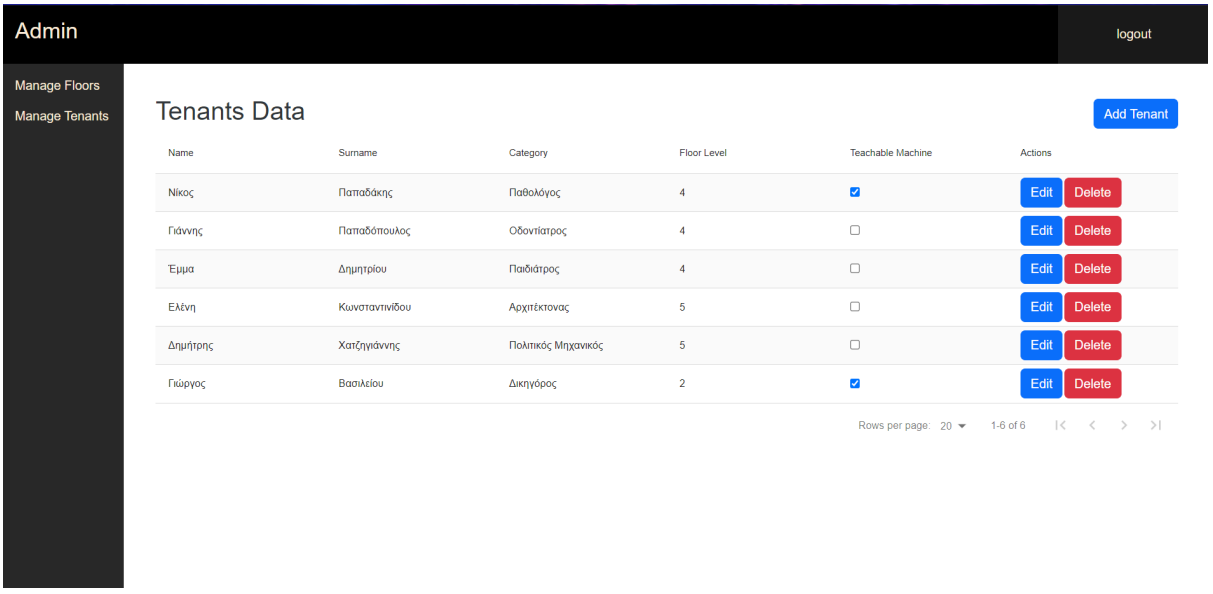
Εικόνα 22 Edit Floor

Στην εικόνα 23 βλέπουμε την διαγραφή ενός ορόφου.



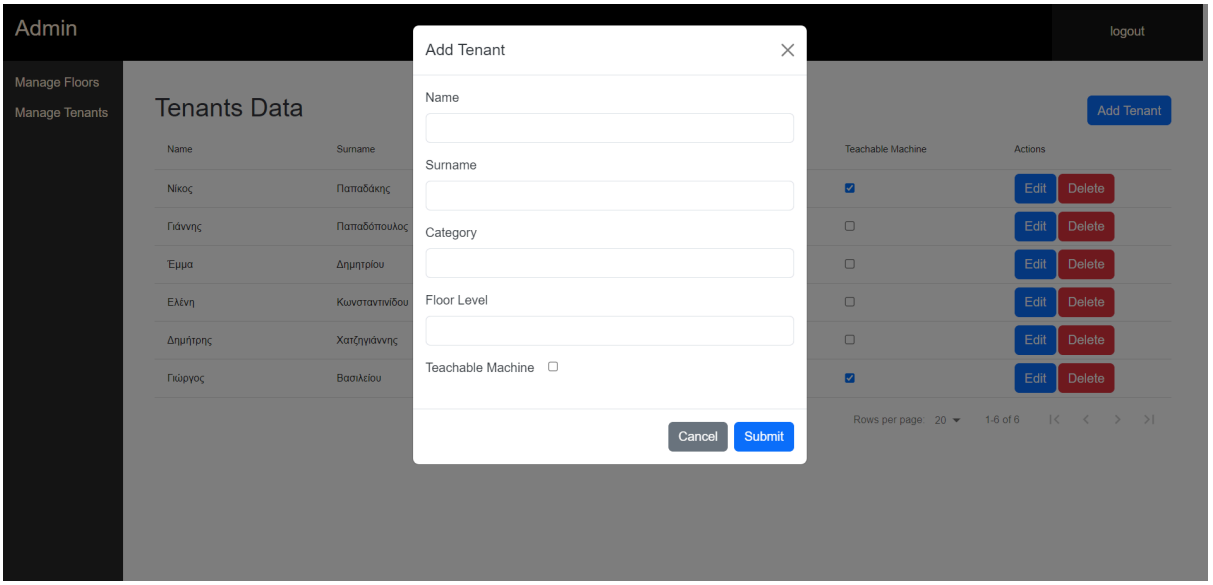
Εικόνα 23 Delete Floor

Στην εικόνα 24 βλέπουμε τον πίνακα των ενοίκων. Στις στήλες παρατηρούμε ότι υπάρχουν πληροφορίες για το όνομα, το επίθετο, το επάγγελμα τους, τον όροφο τους, και άμα έχουν πρόσβαση στην αναγνώριση προσώπου (Teachable Machine).



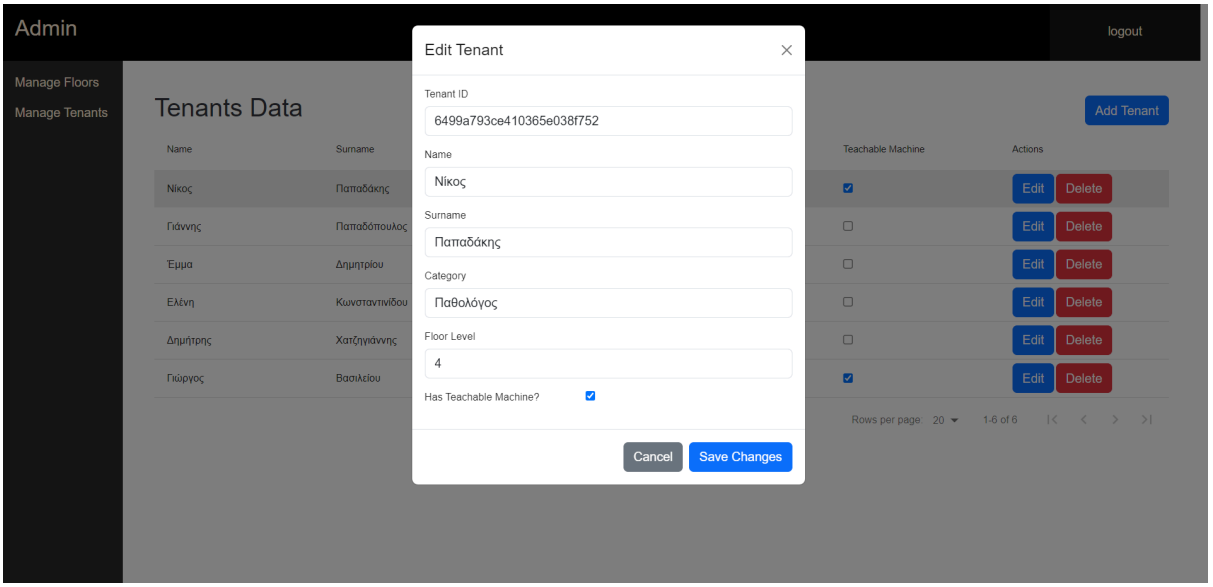
Εικόνα 24 Tenants Data

Στην εικόνα 25 βλέπουμε ότι υπάρχει η δυνατότητα για προσθήκη ένοικου.



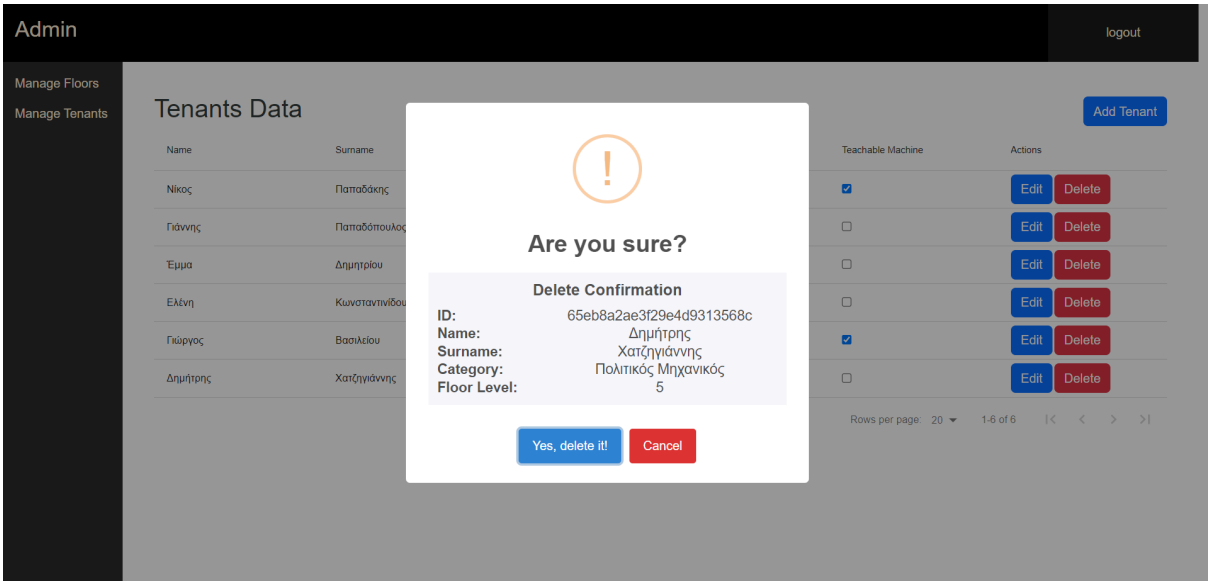
Εικόνα 25 Add Tenant

Στην εικόνα 26 βλέπουμε ότι υπάρχει η δυνατότητα για επεξεργασία του ενοίκου.



Εικόνα 26 Edit Tenant

Στην εικόνα 27 βλέπουμε ότι υπάρχει η δυνατότητα διαγραφής ενοίκου.



Εικόνα 27 Delete Tenant

Κεφάλαιο 6 - Αξιολόγηση Συστήματος

Σε αυτό το κεφάλαιο, θα πραγματοποιηθεί η αξιολόγηση του συστήματος. Η ταξινόμηση των εγγεγραμμένων ενοίκων εκτελείται με ακρίβεια. Ομοίως, η προσομοίωση της μετακίνησης των ορόφων μέσω του έξυπνου ανελκυστήρα αποδεικνύεται αποτελεσματική. Στο πεδίο της διαχείρισης, η σύνδεση και αποσύνδεση στην διαδικτυακή εφαρμογή δείχνει την αξιοπιστία του συστήματος. Επιπλέον, οι διαδικασίες ταξινόμησης, προσθήκης, επεξεργασίας και διαγραφής και αλλαγής σελίδας στο πίνακα στις ενότητες 'Manage Floors' και 'Manage Tenants' λειτουργούν αποτελεσματικά και δίνουν μεγάλη ευελιξία στους διαχειριστές για την διαχείριση των δεδομένων.

Έγιναν δοκιμές στην αναγνώριση ομιλίας του Web Speech API. Χρησιμοποιήθηκε ως δοκιμαστικό κείμενο ένα άρθρο από την Βικιπαίδεια. Στην πρώτη δοκιμή, από τις 76 λέξεις, το σύστημα κατάφερε να αναγνώριση επιτυχώς 62 λέξεις. Στην δεύτερη δοκιμή, από τις 119 λέξεις, αναγνώρισε επιτυχώς 100.

Το κείμενο της Βικιπαίδειας στις δοκιμές διαβάστηκε με την βοήθεια του google translate για την αποφυγή τυχών λαθών στην ανάγνωση. Τα αποτελέσματα της αναγνώρισης ομιλίας εμφανίστηκαν στην οθόνη και στο τέλος έγινε η σύγκριση με το αρχικό κείμενο και η μέτρηση της επιτυχίας.

Ακόμη, έγιναν δοκιμές με φράσεις που είναι πιθανόν να χρησιμοποιηθούν στο σύστημα ενός ανελκυστήρα, όπως:

- «Καλημέρα, θέλω να πάω στον πέμπτο όροφο.»
- «Καλό απόγευμα, με πας στο δεύτερο;»
- «Καλησπέρα, πηγαίνω στο ισόγειο.»
- «Πάω στον τρίτο όροφο.»
- «Καλημέρα, μπορείς να με πας στο τέταρτο;»
- «Πρέπει να ανέβω στον πρώτο.»

Το σύστημα ανταποκρίθηκε επιτυχώς σε όλες τις παραπάνω προτάσεις, καταφέρνοντας να κατευθύνει τον ανελκυστήρα στον επιθυμητό όροφο. Τέλος, έγιναν δοκιμές με θόρυβο βάθους (background noise), σε πραγματικές συνθήκες αλλά και με την βοήθεια ενός βίντεο με θορύβους βάθους (background noise). Αν και υπήρχε θόρυβος, η αναγνώριση ομιλίας λειτούργησε αποτελεσματικά. Περιστασιακά, ήταν απαραίτητο να αυξάνεται η ένταση της φωνής, κάτι που είναι αναμενόμενο υπό τις συνθήκες αυτές. Το σύστημα ομιλίας υποστηρίζει περισσότερους από δυο χρήστες, παρόλο που είναι πιο αποτελεσματικό όταν χρησιμοποιείται με ένα χρήστη.

Επιπρόσθετα, η αναγνώριση προσώπου με το Teachable Machine φαίνεται να έχει αρκετή ακρίβεια όταν χρησιμοποιούνται τα δεδομένα που εκπαιδεύτηκε. Ωστόσο, φαίνεται να μειώνεται η ακρίβεια όταν το μοντέλο εφαρμόζεται σε δεδομένα που διαφέρουν με αυτά της αρχικής εκπαίδευσης. Αυτό το βλέπουμε από την μειωμένη απόδοση του σε διαφορετικές συνθήκες όπως αλλαγές στον φωτισμό ή από ασυνήθιστες γωνίες λήψεις. Τέλος η ταχύτητα της επεξεργασίας είναι αρκετά καλή, κάτι που το καθιστά ιδανικό για εφαρμογές σε πραγματικό χρόνο.

Η αναγνώριση ομιλίας δεν μπορεί να λειτουργήσει χωρίς σύνδεση στο διαδίκτυο λόγω της χρήσης του Web Speech API και του Responsive Voice. Συγκεκριμένα, η αναγνώριση ομιλίας βασίζεται σε έναν μηχανισμό αναγνώρισης που υπάρχει στον διακομιστή. Όλα τα φωνητικά δεδομένα στέλνονται στον διακομιστή για την επεξεργασία, οπότε δεν θα λειτουργήσει χωρίς σύνδεση στο διαδίκτυο.

Κεφάλαιο 7 - Συμπεράσματα και Μελλοντικές Επεκτάσεις

Συμπεραίνουμε ότι ο έξυπνος ανελκυστήρας μπορεί να ενσωματωθεί με τις τεχνολογίες του διαδικτύου και οι λειτουργίες που προσφέρει βελτιώνουν την εμπειρία του χρήστη. Σαν αρχικό στάδιο, αυτές οι λειτουργίες μπορούν εύκολα να χρησιμοποιηθούν ως βοηθητικές στην λειτουργία του ανελκυστήρα.

Η παροχή πληροφοριών στον χρήστη, όπως ο καιρός, μπορεί να εμπλουτίσει την εμπειρία του. Επιπλέον, αυτό δείχνει ότι οι πληροφορίες μπορούν να προσαρμοστούν και να είναι κατάλληλες σε κάθε περίσταση, στοχεύοντας έτσι στο κοινό τους.

Η εμφάνιση πληροφορίων για τους ενοικιαστές που έχουν δώσει την συγκατάθεση τους, καθιστά μια απλή λειτουργία αλλά με την χρήση εργαλείων όπως ταξινόμηση μπορούν να τα κάνουν αρκετά χρήσιμα στην αναζήτηση του ορόφου από τους χρήστες

Η εμφάνιση εγκεκριμένων πληροφοριών για τους ενοικιαστές είναι μια απλή λειτουργία που, μέσω της ταξινόμησης, μπορεί να επιταχύνει σημαντικά την αναζήτηση του σωστού ορόφου από τους χρήστες. Για παράδειγμα, οι χρήστες μπορούν γρήγορα βρουν σε ποιον όροφο βρίσκεται ο γιατρός, ο δικηγόρος τους και τα λοιπά.

Η αναγνώριση ομιλίας μπορεί να επιλύσει καθημερινά προβλήματα και να προσφέρει σημαντική βοήθεια σε άτομα με ειδικές ανάγκες. Με την χρήση των φωνητικών εντολών μπορούν να μεταφερθούν αποτελεσματικά στο κτήριο. Επιπλέον, όταν οι ενοικιαστές έχουν παρέχει τα δεδομένα τους για την αναγνώριση, οι χρήστες μπορούν μέσω των φωνητικών εντολών να μεταφερθούν στον όροφο του ενοικιαστή.

Η αξιολόγηση της αναγνώρισης ομιλίας έδειξε ότι αν και το σύστημα υποστηρίζει πολλαπλούς χρήστες, η απόδοση του είναι καλύτερη όταν χρησιμοποιείται ένας χρήστης. Οπότε, ένας μελλοντικός στόχος είναι η βελτίωση της απόδοσης του για πολλαπλούς χρήστες. Παράλληλα, είναι σημαντικό να γίνει έρευνα για ενναλακτικές τεχνολογίες αναγνώρισης ομιλίας που να πληρούν τις απαιτήσεις του συστήματος. Την υποστήριξη της ελληνικής γλώσσας για την μετατροπή ομιλίας σε κείμενο και το αντίστροφο, η λειτουργία εκτός σύνδεση και η ακριβής επεξεργασία των λέξεων που χρειάζεται ο έξυπνος ανελκυστήρας.

Ακόμη, μπορούν να γίνουν βελτιώσεις στην αλληλεπίδραση του χρήστη με τον έξυπνο ανελκυστήρα. Μια βελτίωση θα ήταν η εκκίνηση του ανελκυστήρα να πραγματοποιείται μόνο μετά από την επιβεβαίωση του χρήστη, η οποία μπορεί να είναι είτε φωνητική είτε μέσω από πάτημα κουμπιού στην οθόνη. Η φωνητική επιβεβαίωση υποστηρίζεται μέσω της αναγνώρισης προσώπου, κάτι που διευκολύνει την ενσωμάτωση αυτή της λειτουργίας.

Η αξιολόγηση της αναγνώρισης προσώπου έδειξε ότι υπάρχουν περιθώρια βελτίωσης. Η εκπαίδευση του μοντέλου δεν περιλάμβανε διάφορες συνθήκες, όπως αλλαγές στον φωτισμό ή ασυνήθιστες γωνίες λήψης, οπότε αυτό αποτελεί ένα μελλοντικό στόχο. Επίσης μπορεί να γίνει διερεύνηση των δυνατοτήτων για ταυτόχρονη αναγνώριση πολλαπλών προσώπων και των απαραίτητων τεχνολογιών για την εφαρμογή τους.

Επιπλέον, μια μελλοντική επέκταση θα μπορούσε να είναι την αλλαγή της μεθόδου αποτίμησης FIFO, δεδομένου ότι ο έξυπνος ανελκυστήρας, εξελίσσεται για να υποστηρίζει αξιόπιστα πολλαπλούς χρήστες σε όλους τους τομείς, θα χρειαστεί να υιοθετήσει πιο αποδοτικούς αλγόριθμους, όπως αυτούς που χρησιμοποιούν οι σύγχρονοι ανελκυστήρες.

Επιπρόσθετα, μια μελλοντική επέκταση θα μπορούσε να περιλαμβάνει την ενσωμάτωση του υλικού για την αντικατάσταση των προσομοιώσεων με λογισμικό, κάνοντας έτσι την χρήση του πιο πραγματική. Τέλος, είναι δυνατόν να γίνουν βελτιώσεις στην σελίδα του διαχειριστή, ώστε να ενισχυθεί η εμπειρία του και να αυξηθεί η αποδοτικότητα του.

Βιβλιογραφία

1. IBM. (n.d.). What is Industry 4.0? Retrieved December 28, 2023, from https://www.ibm.com/topics/industry-4-0
2. Wikipedia. (n.d.). Fourth Industrial Revolution. Wikipedia. Retrieved December 28, 2023, from https://en.wikipedia.org/wiki/Fourth_Industrial_Revolution
3. TWI. (n.d.). What is Industry 4.0? Retrieved December 28, 2023, from https://www.twi-global.com/what-we-do/research-and-technology/technologies/industry-4-0
4. Grand View Research. (n.d.). Facial Recognition Market Size, Share & Trends Report, 2030. Retrieved December 28, 2023, from https://www.grandviewresearch.com/industry-analysis/facial-recognition-market
5. Mordor Intelligence. (n.d.). Facial Recognition Market - Size, Growth & Industry Research Report. Retrieved December 28, 2023, from https://www.mordorintelligence.com/industry-reports/facial-recognition-market
6. Market Research Future. (n.d.). Facial Recognition Market Research Report – Global Forecast till 2032. Retrieved December 28, 2023, from https://www.marketresearchfuture.com/reports/facial-recognition-market-1250
7. Grand View Research. (n.d.). Voice & Speech Recognition Market Size, Share Report 2030. Retrieved December 28, 2023, from https://www.grandviewresearch.com/industry-analysis/voice-recognition-market
8. Markets and Markets. (n.d.). Speech and Voice Recognition Market Size. Retrieved December 28, 2023, from https://www.marketsandmarkets.com/Market-Reports/speech-voice-recognition-market-202401714.html
9. Fortune Business Insights. (n.d.). Speech and Voice Recognition Market Growth Analysis, 2030. Retrieved December 28, 2023, from https://www.fortunebusinessinsights.com/industry-reports/speech-and-voice-recognition-market-101382
10. Interview Bit. (n.d.). IoT Architecture. Retrieved December 28, 2023, from https://www.interviewbit.com/blog/iot-architecture/
11. Amazon Web Services. (n.d.). What is Facial Recognition? Retrieved December 28, 2023, from https://aws.amazon.com/what-is/facial-recognition/

12. Shijakovski, D. (2022, April 20). Using Teachable Machine for Sound and Image Classification in AI Applications. Audio & Video Test Lab. Retrieved December 28, 2023, from https://audiovideotestlab.com/blog/using-teachable-machine-for-sound-and-image-classification-in-ai-applications/
13. Google. (n.d.). FAQ - Teachable Machine. Retrieved December 28, 2023, from https://teachablemachine.withgoogle.com/faq
14. Gillis, A. S. (n.d.). What is Voice Recognition? Definition from TechTarget. Retrieved December 28, 2023, from https://www.techtarget.com/searchcustomerexperience/definition/voice-recognition-speaker-recognition
15. Rec Faces. (n.d.). What is Voice Recognition? Retrieved December 28, 2023, from https://recfaces.com/articles/what-is-voice-recognition
16. University of Illinois Urbana-Champaign. (n.d.). Speech Accessibility Project. Beckman Institute for Advanced Science and Technology. Retrieved December 28, 2023, from https://speechaccessibilityproject.beckman.illinois.edu/
17. R, Kamal. (2022, July 18). A Guide On the IoT Solutions For Smart Elevator Management. Updated April 25, 2023. Retrieved December 28, 2023, from https://www.intuz.com/blog/guide-iot-solutions-for-smart-elevator-management
18. IBM. (n.d.). What is the Internet of Things (IoT)? IBM. Retrieved December 28, 2023, from https://www.ibm.com/topics/internet-of-things
19. Gillis, A. S. (n.d.). What is IoT (Internet of Things) and How Does it Work? Definition from TechTarget. Retrieved December 28, 2023, from https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT
20. Oracle. (n.d.). What Is the Internet of Things (IoT)?. Retrieved December 28, 2023, from https://www.oracle.com/internet-of-things/what-is-iot/
21. Robustel. (n.d.). IoT in Elevators: Application of Smart Elevator Management Solutions. Retrieved December 28, 2023, from https://www.robustel.com/smart-cities/elevator/
22. DusunIoT. (2023, June 7). Smart Elevator Management Based on IoT Solutions. Retrieved December 28, 2023, from https://www.dusuniot.com/blog/smart-elevator-management-based-on-iot-solutions

23. Barney, N., & Bernstein, C. (n.d.). What Is Face Detection and How Does It Work? TechTarget. Retrieved December 28, 2023, from https://www.techtarget.com/searchenterpriseai/definition/face-detection
24. IBM. (n.d.). What Is Speech Recognition? IBM. Retrieved December 28, 2023, from https://www.ibm.com/topics/speech-recognition
25. Geekflare. (n.d.). The Impact of IoT on Web Development. Retrieved December 28, 2023, from https://geekflare.com/iot-on-web-development/
26. Nabto. (2023). A Complete Guide to REST APIs in IoT [2023 update]. Retrieved December 28, 2023, from https://www.nabto.com/rest-api-iot-guide/
27. Kundariya, H. (2021, January 22). Why You Should Choose React.js for IoT Applications. Strapi. Retrieved December 28, 2023, from https://strapi.io/blog/why-you-should-choose-react-js-for-iot-applications
28. ResponsiveVoice. (n.d.). ResponsiveVoice Text To Speech - ResponsiveVoice.JS Text to Speech. Retrieved December 28, 2023, from https://responsivevoice.org/
29. Web Platform Incubator Community Group. (2020, August 18). Web Speech API. Retrieved December 28, 2023, from https://wicg.github.io/speech-api/
30. Mozilla Foundation. (2023, February 19). Web Speech API - Web APIs. MDN Web Docs. Retrieved December 28, 2023, from https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API
31. Google Creative Lab. (n.d.). Teachable Machine Community: TensorFlow.js Image Classification Snippets in JavaScript. Retrieved December 28, 2023, from https://github.com/googlecreativelab/teachablemachine-community/blob/master/snippets/markdown/image/tensorflowjs/javascript.md
32. React Community. (n.d.). React - React Reference. React Documentation. Retrieved June 7, 2024 from https://react.dev/reference/react
33. Wikipedia. (n.d.). Ιστορία – Ιστορία Reference. Wikipedia. Retrieved June 23, 2024, from https://el.wikipedia.org/wiki/Ιστορία
34. MDN Web Docs. (n.d.). Navigator geolocation. MDN Web Docs. Retrieved July 7, 2024, from https://developer.mozilla.org/en-US/docs/Web/API/Navigator/geolocation