



Τμήμα Ηλεκτρολόγων Μηχανικών  
& Μηχανικών Υπολογιστών

**ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΕΛΟΠΟΝΝΗΣΟΥ**

Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών  
Πρόγραμμα Προπτυχιακών Σπουδών

# Διπλωματική Εργασία

## Τεχνικές διαχείρισης δεδομένων χώρων υποβοηθούμενης διαβίωσης για ετερογενείς βάσεις δεδομένων

**Επιβλέπων Καθηγητής: Αντωνόπουλος Χρήστος**

**Συνεπιβλέπουσα Καθηγήτρια: Φαλιάγκα Εύη**

**Φοιτητής: Θωμάς Κωνσταντίνος (2944)**

09 - 09 - 2023

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, \_\_-\_\_-2023

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ονοματεπώνυμο, Υπογραφή
2. Ονοματεπώνυμο, Υπογραφή
3. Ονοματεπώνυμο, Υπογραφή

Υπεύθυνη Δήλωση Φοιτητή Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη εργασία. Η έγκριση της διπλωματικής εργασίας από το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος. Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Θωμά Κωνσταντίνου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο Πανεπιστήμιο Πελοποννήσου, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.

Οι μορφή των εισαγωγικών σελίδων είναι με μικρούς χαρακτήρες ρωμαϊκής γραφής (i, ii, iii, iv, κλπ)

# Περίληψη

Η παρούσα διπλωματική έχει ως σκοπο την αναλυση Τεχνικών διαχείρισης δεδομένων χώρων υποβοηθούμενης διαβίωσης για ετερογενείς βάσεις δεδομένων και την παρουσίαση δεδομένων έξυπνων συσκευών μέσω δημιουργίας εφαρμογής για χρήση από τους χρήστες

# Abstract

In this thesis, the aim is to analyze the techniques of data management for assisted living spaces with heterogeneous databases and present data from smart devices through the development of an application intended for user use.

# Περιεχόμενα

<b>1.Εισαγωγή</b>	<b>6</b>
1.1 Σκοπος της διπλωματικής	6
<b>2.Διαχείριση δεδομένων σε χώρους υποβοηθούμενης διαβίωσης για ετερογενείς βάσεις δεδομένων</b>	<b>8</b>
2.1 Ενοποίηση Δεδομένων	8
2.2 Βάσεις Δεδομένων Γράφων	9
2.3 Επεξεργασία Δεδομένων σε Πραγματικό Χρόνο	9
2.4 Ασφάλεια Δεδομένων	10
2.5 Υπολογισμός Στατιστικών και Μηχανική Μάθηση	11
<b>3.Τεχνητή νοημοσύνη</b>	<b>12</b>
3.1 Τι είναι η τεχνητή νοημοσύνη;	12
3.2 Υποκατηγορίες τεχνητής νοημοσύνης	12
3.3 Μηχανική μάθηση	13
3.4 Σημαντικές εφαρμογές	13
3.5 Προκλήσεις	14
3.5 Ηθικά και νομικά θέματα	14
<b>4.Περιβάλλον υποβοηθούμενης διαβίωσης</b>	<b>16</b>
4.1 Παροχές και Υπηρεσίες υποβοηθούμενης διαβίωσης	16
4.2 Εφαρμογή σε διάφορα περιβάλλοντα	16
4.3 Πρωτοβουλίες, υπηρεσίες και πρακτικές	17
<b>5.Τεχνική Εφαρμογή</b>	<b>18</b>
5.1 Τι είναι η postgresql	18
5.2 Pgadmin	18
5.3 DBeaver	19
5.4 Τι είναι τα queries	20
5.5 React	20
5.6 ExpressJS	21
5.7 Υλοποίηση	21
5.7.1 Εφαρμογή	21
5.7.2 Δημιουργία Πινάκων	22
5.7.3 Εισαγωγή Δεδομένων	22
5.7.4 Εκτέλεση Ερωτημάτων	24
<b>Πηγές</b>	<b>33</b>
<b>Παράρτημα 1</b>	<b>34</b>
Front End	34
Back End	38

# 1.Εισαγωγή

Η διαχείριση των δεδομένων αποτελεί έναν ουσιαστικό παράγοντα για την επιτυχία και την αποτελεσματική λειτουργία πολλών οργανισμών και εφαρμογών. Με την αύξηση του όγκου και της ποικιλίας των δεδομένων, καθώς και την εκρηκτική ανάπτυξη της τεχνολογίας, ο ρόλος της διαχείρισης δεδομένων γίνεται ολοένα και πιο κρίσιμος.

Σε αυτό το πλαίσιο, η υποβοηθούμενη διαβίωση εμφανίζεται ως μια αναγκαία προσέγγιση για την αποτελεσματική διαχείριση δεδομένων, ιδίως όταν αντιμετωπίζουμε ετερογενείς βάσεις δεδομένων. Ετερογενείς βάσεις δεδομένων περιλαμβάνουν δεδομένα που προέρχονται από διάφορες πηγές, με διάφορες δομές και μορφές. Η τεχνική διαχείρισης δεδομένων σε αυτό το πλαίσιο απαιτεί νέες προσεγγίσεις και εργαλεία που μπορούν να αντιμετωπίσουν την πολυπλοκότητα των ετερογενών δεδομένων.

Στη διατριβή αυτή, θα εξετάσουμε τις τεχνικές διαχείρισης δεδομένων χώρων υποβοηθούμενης διαβίωσης που αποσκοπούν στη δημιουργία ευέλικτων και αποτελεσματικών λύσεων για τη διαχείριση ετερογενών δεδομένων. Θα εξετάσουμε την ανάγκη για τέτοιου είδους τεχνικές, τις προκλήσεις που αντιμετωπίζουμε και τους τρόπους με τους οποίους μπορούμε να αξιοποιήσουμε την υποβοηθούμενη διαβίωση για τη βελτίωση της διαχείρισης δεδομένων.

Στόχος αυτής της διατριβής είναι να διαμορφώσει μια κατανόηση των τεχνικών διαχείρισης δεδομένων χώρων υποβοηθούμενης διαβίωσης, να αξιολογήσει την αποτελεσματικότητά τους και να προτείνει πιθανές βελτιώσεις στον τομέα αυτόν.

## 1.1 Σκοπος της διπλωματικής

Ο σκοπός της διπλωματικής εργασίας με θέμα "Τεχνικές διαχείρισης δεδομένων χώρων υποβοηθούμενης διαβίωσης για ετερογενείς βάσεις δεδομένων" είναι να εξετάσει, να αναπτύξει και να αξιολογήσει τεχνικές και πρακτικές για την αποτελεσματική διαχείριση ετερογενών δεδομένων από διάφορες πηγές και βάσεις δεδομένων, χρησιμοποιώντας την υποβοηθούμενη διαβίωση.

Συγκεκριμένα, η διπλωματική εργασία αποσκοπεί στα εξής:

- Ανάλυση της πολυπλοκότητας των ετερογενών δεδομένων: Θα διερευνήσει τον τρόπο με τον οποίο διαφορετικά δεδομένα από ποικίλες πηγές μπορούν να έχουν διαφορετικές δομές και μορφές, καθώς και τις προκλήσεις που αντιμετωπίζονται κατά τη διαχείρισή τους.
- Αξιολόγηση των υποβοηθούμενων τεχνολογιών διαβίωσης: Θα εξετάσει πώς οι τεχνολογίες όπως η τεχνητή νοημοσύνη, η μηχανική μάθηση και η ανάλυση δεδομένων μπορούν να χρησιμοποιηθούν για την ανάλυση, τη σύνδεση και τη διαχείριση ετερογενών δεδομένων.
- Ανάπτυξη πρακτικών λύσεων: Θα εξετάσει πώς μπορούν να αναπτυχθούν πρακτικές λύσεις για την αποτελεσματική διαχείριση ετερογενών δεδομένων, συμπεριλαμβανομένης της ενοποίησης, της ανάλυσης και της παρουσίασης των δεδομένων.
- Αξιολόγηση και προτάσεις βελτιώσεων: Θα αξιολογήσει την αποτελεσματικότητα των τεχνικών διαχείρισης δεδομένων και θα προτείνει βελτιώσεις και μελλοντικές κατευθύνσεις.

Συνοψίζοντας, η διπλωματική εργασία αποσκοπεί στην ανάπτυξη και αξιολόγηση τεχνικών διαχείρισης δεδομένων που χρησιμοποιούν την υποβοηθούμενη διαβίωση για την αντιμετώπιση των προκλήσεων που σχετίζονται με την ετερογενή φύση των δεδομένων.

## 2. Διαχείριση δεδομένων σε χώρους υποβοηθούμενης διαβίωσης για ετερογενείς βάσεις δεδομένων

Η διαχείριση δεδομένων σε χώρους υποβοηθούμενης διαβίωσης για ετερογενείς βάσεις δεδομένων αποτελεί σημαντική πρόκληση στον τομέα της τεχνολογίας. Αυτή η διαδικασία απαιτεί τη συλλογή, ενοποίηση και αξιοποίηση δεδομένων από διάφορες πηγές, προκειμένου να υποστηρίξει εφαρμογές υποβοηθούμενης διαβίωσης, όπως συστήματα υγείας και καλής φυσικής κατάστασης, έξυπνα σπίτια και άλλες υπηρεσίες που βελτιώνουν την ποιότητα ζωής.

Οι τεχνικές διαχείρισης δεδομένων πρέπει να περιλαμβάνουν τη συγκέντρωση δεδομένων από διάφορες πηγές, τον καθαρισμό και την επεξεργασία τους, τη δημιουργία ενοποιημένων βάσεων δεδομένων και τη διατήρηση της ασφάλειας και της ιδιωτικότητας των δεδομένων. Η ευέλικτη αντιμετώπιση των διαφόρων τύπων δεδομένων είναι ζωτικής σημασίας, καθώς οι πηγές μπορεί να περιλαμβάνουν δεδομένα αισθητήρων, ιατρικές πληροφορίες, κοινωνικές αλυσίδες και άλλα.

Η αναγνώριση και η αντιμετώπιση των ηθικών και νομικών ζητημάτων, όπως η ιδιωτικότητα των δεδομένων και η ασφάλεια, αποτελεί κρίσιμο κομμάτι της διαχείρισης δεδομένων σε αυτό το περιβάλλον. Η διαφάνεια και η ευκολία πρόσβασης στα δεδομένα είναι επίσης σημαντικές παράμετροι για την επίτευξη επιτυχημένων εφαρμογών υποβοηθούμενης διαβίωσης που βελτιώνουν την ποιότητα ζωής και την υγεία των ανθρώπων.[8]

### 2.1 Ενοποίηση Δεδομένων

Η ενοποίηση δεδομένων είναι η διαδικασία συνδυασμού δεδομένων από διάφορες πηγές σε ένα κοινό πλαίσιο, έτσι ώστε να μπορούν να αξιοποιηθούν και να αναλυθούν αποτελεσματικά. Η ενοποίηση δεδομένων γίνεται συχνά όταν έχουμε δεδομένα από διάφορες πηγές, όπως βάσεις δεδομένων, αρχεία, συστήματα αισθητήρων, κοινωνικά μέσα κοινωνικής δικτύωσης και άλλα.

Οι βασικοί στόχοι της ενοποίησης δεδομένων περιλαμβάνουν:

- **Αντιμετώπιση Διαφορετικών Σχημάτων:** Συχνά, τα δεδομένα από διάφορες πηγές έχουν διαφορετικές δομές και σχήματα. Η ενοποίηση επιτρέπει την αντιστάθμιση αυτών των διαφορών και τη δημιουργία ενός ενιαίου σχήματος.
- **Εξάλειψη Διπλών Δεδομένων:** Συχνά, τα ίδια δεδομένα μπορεί να υπάρχουν σε πολλές πηγές, δημιουργώντας διπλές εγγραφές. Η ενοποίηση βοηθά στον εντοπισμό και την αφαίρεση αυτών των διπλών.
- **Εξαγωγή Αξίας:** Η ενοποίηση δεδομένων επιτρέπει την εξαγωγή επιπλέον αξίας από τα δεδομένα. Μπορούν να εντοπιστούν τάσεις, πρότυπα και συσχετίσεις που δεν θα ήταν εφικτές μεμονωμένα.
- **Ενοποίηση για Ανάλυση:** Η ενοποίηση δεδομένων είναι σημαντική για τη διεξαγωγή αναλύσεων και αναφορών που μπορούν να παράγουν συνολικές πληροφορίες από διάφορες πηγές.



- Ευελιξία: Με την ενοποίηση, τα δεδομένα γίνονται πιο ευέλικτα και ευκολότερα προσβάσιμα για την ανάπτυξη εφαρμογών και τη διαμοιρασμό πληροφοριών.[8]

## 2.2 Βάσεις Δεδομένων Γράφων

Οι βάσεις δεδομένων γράφων (Graph Databases) είναι ένα είδος διαχείρισης δεδομένων που είναι σχεδιασμένο γύρω από την αποθήκευση και την ανάκτηση πληροφοριών με τη μορφή γράφου. Στον γράφο, οι πληροφορίες απεικονίζονται σαν κόμβοι (nodes) και συνδέσεις (edges) μεταξύ αυτών των κόμβων. Κάθε κόμβος αντιπροσωπεύει ένα στοιχείο, ενώ οι συνδέσεις αντιπροσωπεύουν τις σχέσεις μεταξύ των στοιχείων.

Οι βάσεις δεδομένων γράφων προσφέρουν τα εξής χαρακτηριστικά:

- Αναζήτηση και Επερώτηση: Οι γραφικές βάσεις δεδομένων είναι αποδοτικές στην αναζήτηση και την επερώτηση γράφων. Αυτό τις καθιστά ιδανικές για εφαρμογές που απαιτούν σύνθετες σχέσεις μεταξύ δεδομένων.
- Σχέσεις και Συνδεσιμότητα: Οι γραφικές βάσεις δεδομένων ταιριάζουν ιδανικά σε δεδομένα με σημαντικές σχέσεις και συνδέσεις. Μπορούν να αντιμετωπίσουν πολύπλοκες δομές δεδομένων που δεν είναι τόσο εύκολα απεικονίσιμες σε παραδοσιακές σχεσιακές βάσεις δεδομένων.
- Ευελιξία: Οι γραφικές βάσεις δεδομένων επιτρέπουν την προσθήκη, τροποποίηση και διαγραφή κόμβων και συνδέσεων με ευκολία, χωρίς την ανάγκη για αλλαγές στη δομή της βάσης δεδομένων.
- Αποτελεσματική Ανάκτηση Συνδεδεμένων Δεδομένων: Οι γραφικές βάσεις δεδομένων είναι ιδανικές για εφαρμογές που απαιτούν ανάκτηση δεδομένων που είναι συνδεδεμένα μεταξύ τους, όπως κοινωνικά δίκτυα, γραφήματα δικτύων, και πολλά άλλα.

Ορισμένα από τα δημοφιλή συστήματα διαχείρισης βάσεων δεδομένων γράφων περιλαμβάνουν το Neo4j, το Amazon Neptune, το OrientDB και το ArangoDB. Οι γραφικές βάσεις δεδομένων έχουν εφαρμογές σε πολλούς τομείς, συμπεριλαμβανομένων των κοινωνικών μέσων, του αναλυτικού δικτύου, των προτάσεων, της γεωχωρικής ανάλυσης και άλλων εφαρμογών που αναγκάζουν την αντιμετώπιση σύνθετων σχέσεων δεδομένων.[8]

## 2.3 Επεξεργασία Δεδομένων σε Πραγματικό Χρόνο

Η επεξεργασία δεδομένων σε πραγματικό χρόνο (Real-Time Data Processing) αναφέρεται στη διαδικασία ανάλυσης και επεξεργασίας δεδομένων καθώς αυτά δημιουργούνται ή ροές μέσα στο σύστημα σε πραγματικό χρόνο. Αυτό σημαίνει ότι τα δεδομένα επεξεργάζονται αμέσως καθώς έρχονται στο σύστημα, χωρίς καθυστέρηση. Η επεξεργασία δεδομένων σε πραγματικό χρόνο έχει εφαρμογές σε πολλούς τομείς, συμπεριλαμβανομένων των ακόλουθων:

- Χρηματοπιστωτικές Υπηρεσίες: Συναλλαγές σε πραγματικό χρόνο, ανίχνευση απάτης, παρακολούθηση της αγοράς και ανάλυση κινδύνου.
- Διαφήμιση: Παρακολούθηση της απόδοσης διαφημίσεων σε πραγματικό χρόνο και δυνατότητα προσωποποιημένων προτάσεων.

- Διαδίκτυο των πραγμάτων (IoT): Συλλογή, ανάλυση και επεξεργασία δεδομένων αισθητήρων και συσκευών σε πραγματικό χρόνο για εφαρμογές όπως έξυπνα σπίτια, αυτοκίνητα και βιομηχανικά συστήματα.
- Υγεία: Παρακολούθηση υγείας σε πραγματικό χρόνο, παρακολούθηση ασθενών, ανίχνευση επιδημιών και συμβουλές υγείας.
- Κοινωνικά Δίκτυα: Ανάλυση και αντίδραση στη ροή των κοινωνικών μέσων, ανίχνευση τάσεων και προτάσεις περιεχομένου.

Για την επεξεργασία δεδομένων σε πραγματικό χρόνο, συχνά χρησιμοποιούνται τεχνολογίες όπως τονικές συστήματα υπολογισμού, λύσεις ανάλυσης ροής δεδομένων (stream processing), και αυτόματη καταναμημένη επεξεργασία. Ο στόχος είναι να επιτρέψουν την αντίδραση σε δεδομένα στο πιο σύντομο δυνατό χρονικό διάστημα, προκειμένου να ληφθούν αποφάσεις σε πραγματικό χρόνο.[8]

## 2.4 Ασφάλεια Δεδομένων

Η ασφάλεια δεδομένων αφορά την προστασία των πληροφοριών από απώλειες, κακή χρήση ή μη εξουσιοδοτημένη πρόσβαση. Είναι ένα σημαντικό θέμα σε κάθε περιβάλλον, από το προσωπικό επίπεδο μέχρι τον επαγγελματικό και τον οργανωτικούς τομείς. Εδώ είναι ορισμένες σημαντικές πτυχές της ασφάλειας δεδομένων:

- Κρυπτογραφία: Η κρυπτογραφία είναι η διαδικασία μετατροπής των δεδομένων σε μια ακατανόητη μορφή μέσω της χρήσης κρυπτογραφικών αλγορίθμων. Αυτό προστατεύει τα δεδομένα από την μη εξουσιοδοτημένη πρόσβαση, ακόμα και αν κάποιος αποκτήσει πρόσβαση στην αποθηκευμένη πληροφορία.
- Πολιτικές Πρόσβασης: Ορίζονται πολιτικές πρόσβασης που καθορίζουν ποιοι χρήστες ή εφαρμογές έχουν πρόσβαση σε ποια δεδομένα. Οι πολιτικές αυτές μπορούν να περιλαμβάνουν την ταυτοποίηση, την αυθεντικοποίηση και τον έλεγχο πρόσβασης.
- Παρακολούθηση και Καταγραφή: Η παρακολούθηση των δραστηριοτήτων συστήματος και η καταγραφή αυτών των δραστηριοτήτων μπορεί να βοηθήσει στον εντοπισμό ανωμαλιών και αντικειμενική αξιολόγηση πιθανών παραβιάσεων.
- Φυσική Ασφάλεια: Εξασφαλίζει ότι οι φυσικοί χώροι που φιλοξενούν εξοπλισμό και δεδομένα είναι προστατευμένοι από παραβιάσεις.
- Εκπαίδευση και Ευαισθητοποίηση: Η εκπαίδευση του προσωπικού και η ευαισθητοποίηση σχετικά με τους κινδύνους ασφαλείας είναι κρίσιμα για την προστασία των δεδομένων.
- Αντι-Ιοί και Προστασία Ενημερωμένης Λογισμικής: Η χρήση αντι-ιών και η διατήρηση του λογισμικού ενημερωμένο είναι σημαντικά για την αποτροπή επιθέσεων.
- Εφεδρικές Αντιγραφές: Η τακτική δημιουργία αντιγράφων ασφαλείας των δεδομένων εξασφαλίζει τη δυνατότητα ανάκτησης των πληροφοριών σε περίπτωση απώλειας.
- Συμμόρφωση με τους Κανονισμούς: Οι επιχειρήσεις πρέπει να συμμορφώνονται με τους νόμους και τους κανονισμούς που διέπουν την προστασία των δεδομένων, όπως ο Γενικός Κανονισμός για την Προστασία των Δεδομένων (GDPR).

Η ασφάλεια δεδομένων είναι κρίσιμη για τη διατήρηση της εμπιστοσύνης των χρηστών και την προστασία των ευαίσθητων πληροφοριών. Οι οργανισμοί και τα άτομα πρέπει να λαμβάνουν σοβαρά υπόψη τα μέτρα ασφαλείας για την προστασία των δεδομένων και την πρόληψη ενδεχόμενων απειλών.[8]

## 2.5 Υπολογισμός Στατιστικών και Μηχανική Μάθηση

Ο υπολογισμός στατιστικών και η μηχανική μάθηση αποτελούν δύο σημαντικούς τομείς στην ανάλυση δεδομένων. Εδώ είναι μια επισκόπηση του καθενός:

Υπολογισμός Στατιστικών:

- Συνοπτική Στατιστική: Ο υπολογισμός μέτρων όπως η μέση τιμή, η διάμεσος, το εύρος και η τυπική απόκλιση για να κατανοήσουμε τη διανομή των δεδομένων.
- Συσχέτιση: Ο υπολογισμός συντελεστών συσχέτισης, όπως ο συντελεστής συσχέτισης Pearson, για να εκτιμήσουμε τις σχέσεις μεταξύ μεταβλητών.
- Δοκιμές Υποθέσεων: Ο υπολογισμός p-τιμών και η εφαρμογή δοκιμών υποθέσεων για να αξιολογήσουμε αν οι διαφορές μεταξύ δύο ή περισσότερων ομάδων είναι σημαντικές.

Μηχανική Μάθηση:

- Επιβλεπόμενη Μάθηση: Ορίζει την πρόβλεψη μιας μεταβλητής (εξόδου) βάσει ενός συνόλου εκπαίδευσης που περιλαμβάνει τόσο τις εισόδους όσο και τις αντίστοιχες εξόδους.
- Ανεπιβλεπόμενη Μάθηση: Ομαδοποιεί τα δεδομένα σε κατηγορίες ή ανακαλύπτει πρότυπα χωρίς τη χρήση προκαθορισμένων ετικετών.
- Ενισχυτική Μάθηση: Αναπτύσσει μοντέλα που λαμβάνουν αποφάσεις με βάση την αλληλεπίδραση με το περιβάλλον και την ανταμοιβή που λαμβάνουν για τις ενέργειές τους.
- Βαθιά Μάθηση: Ασχολείται με μεγάλα νευρωνικά δίκτυα (deep neural networks) για την επεξεργασία δεδομένων, όπως εικόνες και κείμενο, και έχει εφαρμογές όπως η αναγνώριση εικόνων και η φωνητική αναγνώριση.

Οι δύο αυτοί τομείς συχνά συνδυάζονται σε εφαρμογές μηχανικής μάθησης. Η μηχανική μάθηση χρησιμοποιεί στατιστικές μεθόδους για την εκπαίδευση μοντέλων πρόβλεψης, και η ανάλυση στατιστικών δεδομένων βοηθά στην κατανόηση των προβλημάτων και την επιλογή των κατάλληλων χαρακτηριστικών για την εκπαίδευση μοντέλων μηχανικής μάθησης.

## 3. Τεχνητή νοημοσύνη

### 3.1 Τι είναι η τεχνητή νοημοσύνη;

Η τεχνητή νοημοσύνη (TN) αναφέρεται στη δημιουργία υπολογιστικών συστημάτων που μπορούν να αντιμετωπίζουν διάφορες εργασίες που συνήθως απαιτούν ανθρώπινη νοημοσύνη. Αυτά τα συστήματα χρησιμοποιούν αλγόριθμους και μοντέλα μηχανικής μάθησης για να αναγνωρίσουν πρότυπα, να προβλέπουν τα αποτελέσματα, και να λύνουν προβλήματα.

Η τεχνητή νοημοσύνη χρησιμοποιείται σε πολλούς τομείς, όπως η αυτόνομη οδήγηση αυτοκινήτων, η αναγνώριση εικόνων, η αυτόματη μετάφραση γλωσσών, η ρομποτική, και πολλοί άλλοι. Οι μηχανές αυτές βασίζονται σε υπολογιστικούς αλγόριθμους που τους επιτρέπουν να επιτελούν διάφορες εργασίες παρόμοιες με αυτές που θα εκτελούσε ένας άνθρωπος.

Αυτή η τεχνολογία εφαρμόζεται σε πολλούς τομείς και είναι αναπόσπαστο μέρος της σύγχρονης τεχνολογίας.[9]

### 3.2 Υποκατηγορίες τεχνητής νοημοσύνης

Η τεχνητή νοημοσύνη χωρίζεται σε πολλές υποκατηγορίες, καθένα από τα οποία επικεντρώνεται σε διάφορες πτυχές της νοημοσύνης. Εδώ είναι μερικές από τις βασικές υποκατηγορίες τεχνητής νοημοσύνης:

- **Μηχανική Μάθηση:** Η μηχανική μάθηση είναι μια υποκατηγορία της τεχνητής νοημοσύνης που επικεντρώνεται στη δημιουργία συστημάτων που μπορούν να μαθαίνουν από δεδομένα. Συμπεριλαμβάνει υποκατηγορίες όπως η επιβλεπόμενη μάθηση, η μη επιβλεπόμενη μάθηση και η ενισχυτική μάθηση.
- **Επεξεργασία Φυσικής Γλώσσας:** Η τεχνητή νοημοσύνη που εστιάζει στην επεξεργασία και τον κατανοητό χειρισμό ανθρώπινης γλώσσας. Αυτό περιλαμβάνει τη μηχανική μετάφραση, την αναγνώριση φωνής, την ανάλυση κειμένου και άλλες εφαρμογές.
- **Αναγνώριση Εικόνων:** Αυτή η υποκατηγορία επικεντρώνεται στη δημιουργία συστημάτων που μπορούν να αναγνωρίζουν και να επεξεργάζονται εικόνες και βίντεο. Συμπεριλαμβάνει την αναγνώριση προσώπων, την αναγνώριση αντικειμένων, και άλλες εφαρμογές.
- **Ρομποτική:** Η υποκατηγορία αυτή επικεντρώνεται στη δημιουργία αυτόνομων ρομπότ που μπορούν να εκτελούν διάφορες εργασίες, όπως η βιομηχανική παραγωγή, η ιατρική, και η εξερεύνηση χώρου.
- **Συστήματα Συστάσεων:** Αυτή η υποκατηγορία ασχολείται με τη δημιουργία συστημάτων που μπορούν να προτείνουν προϊόντα, υπηρεσίες ή περιεχόμενο στους χρήστες με βάση τις προτιμήσεις τους.
- **Αυτόνομη Οδήγηση:** Η τεχνητή νοημοσύνη χρησιμοποιείται για την ανάπτυξη αυτόνομων οχημάτων που μπορούν να οδηγούν χωρίς ανθρώπινη παρέμβαση.[9]

### 3.3 Μηχανική μάθηση

Η μηχανική μάθηση είναι μια υποκατηγορία της τεχνητής νοημοσύνης που επικεντρώνεται στη δημιουργία αλγορίθμων και μοντέλων που μπορούν να μαθαίνουν από δεδομένα και να προβλέπουν τα μελλοντικά αποτελέσματα. Αυτή η υποκατηγορία δεν αποτελεί λογοκλοπή. Οι αλγόριθμοι μηχανικής μάθησης αναπτύσσονται και βελτιώνονται από επιστήμονες και μηχανικούς σε όλο τον κόσμο και βασίζονται σε μαθηματικές και στατιστικές αρχές.

Οι κύριες υποκατηγορίες της μηχανικής μάθησης περιλαμβάνουν:

- **Επιβλεπόμενη Μάθηση:** Στην επιβλεπόμενη μάθηση, οι αλγόριθμοι εκπαιδεύονται χρησιμοποιώντας ένα σύνολο δεδομένων που περιλαμβάνει ετικέτες. Αυτός ο τύπος μάθησης χρησιμοποιείται για την κατηγοριοποίηση και την πρόβλεψη.
- **Μη Επιβλεπόμενη Μάθηση:** Στην μη επιβλεπόμενη μάθηση, οι αλγόριθμοι εκπαιδεύονται χωρίς τη χρήση ετικετών. Ανακαλύπτουν αυτόματα πρότυπα και σχέσεις στα δεδομένα, συμπεριλαμβανομένων των αλγορίθμων συστατικής ανάλυσης και ομαδοποίησης.
- **Ενισχυτική Μάθηση:** Η ενισχυτική μάθηση ασχολείται με τη λήψη αποφάσεων με βάση την ανταμοιβή και την επιβράβευση. Χρησιμοποιείται για την εκπαίδευση αλγορίθμων που λαμβάνουν αποφάσεις σε δυναμικά περιβάλλοντα. [9]

### 3.4 Σημαντικές εφαρμογές

Η τεχνητή νοημοσύνη έχει πολλές σημαντικές εφαρμογές σε διάφορους τομείς. Ορισμένες από τις σημαντικές εφαρμογές της τεχνητής νοημοσύνης περιλαμβάνουν:

- **Υγεία:** Η τεχνητή νοημοσύνη χρησιμοποιείται για τη διάγνωση ασθενειών, την πρόβλεψη εξελίξεων ασθενειών, την εξατομίκευση της θεραπείας και την ανάλυση μεγάλων συνόλων δεδομένων υγείας.
- **Ρομποτική:** Οι ρομποτικοί συστήματα που βασίζονται στην τεχνητή νοημοσύνη χρησιμοποιούνται σε βιομηχανικές εφαρμογές, ιατρικές επεμβάσεις, εξερεύνηση δύσκολων περιβαλλόντων και άλλες εργασίες.
- **Αναγνώριση Φωνής:** Η τεχνητή νοημοσύνη χρησιμοποιείται για την αναγνώριση φωνής και τη δημιουργία φωνητικών υπηρεσιών, όπως τα ψηφιακά βοηθήματα.
- **Αναγνώριση Εικόνων:** Η τεχνητή νοημοσύνη εφαρμόζεται στην αναγνώριση προσώπων, αντικειμένων και τον αυτόματο έλεγχο ποικίλων συστημάτων.
- **Αυτόνομη Οδήγηση:** Η τεχνητή νοημοσύνη χρησιμοποιείται για την ανάπτυξη αυτόνομων οχημάτων που μπορούν να οδηγούν χωρίς ανθρώπινη επέμβαση.
- **Συστήματα Συστάσεων:** Η τεχνητή νοημοσύνη χρησιμοποιείται για την προτεινόμενη εξατομικευμένη συστάσεις, όπως στις πλατφόρμες streaming, e-commerce και πολλές άλλες.
- **Οικονομία και Επιχειρήσεις:** Η τεχνητή νοημοσύνη χρησιμοποιείται για τη βελτιστοποίηση διαδικασιών, την πρόβλεψη τάσεων στις χρηματοοικονομικές αγορές και τη διαχείριση πελατών.
- **Περιβάλλον:** Η τεχνητή νοημοσύνη χρησιμοποιείται για την παρακολούθηση και πρόβλεψη φυσικών καταστροφών, τη βελτιστοποίηση της χρήσης πόρων και την αειφόρο ανάπτυξη.

- Εκπαίδευση: Η τεχνητή νοημοσύνη χρησιμοποιείται για τη δημιουργία εκπαιδευτικών εφαρμογών, εξατομικευμένης εκπαίδευσης και αξιολόγησης της μάθησης.[9]

### 3.5 Προκλήσεις

Οι προκλήσεις της τεχνητής νοημοσύνης είναι πολλές και ποικίλες, χωρίς να αναφέρονται προηγούμενες πληροφορίες. Αυτές οι προκλήσεις περιλαμβάνουν:

- Απορρύθμιση: Η απορρύθμιση αναφέρεται στον κίνδυνο που δημιουργείται όταν η τεχνητή νοημοσύνη αντικαθιστά ανθρώπους σε θέσεις εργασίας, με τον κίνδυνο απώλειας θέσεων εργασίας και αύξησης της ανισότητας.
- Ηθικά ζητήματα: Η χρήση της τεχνητής νοημοσύνης συχνά συνδέεται με ερωτήματα ηθικής, όπως η ιδιωτικότητα, η αναγνώριση προσώπων και η απόφαση αυτόνομων συστημάτων.
- Ευπάθεια στη δεδομένη προκειμένη: Οι αλγόριθμοι τεχνητής νοημοσύνης μπορούν να γίνουν ευάλωτοι σε δεδομένες προκειμένες, όπως την ανεπαρκή εκπαίδευση ή την αντίθετη διακριτική μεταχείριση.
- Διαφάνεια: Οι αλγόριθμοι μηχανικής μάθησης και νευρωνικά δίκτυα μπορεί να είναι δύσκολο να εξηγηθούν, κάτι που μπορεί να αποτελέσει πρόκληση στην επίλυση προβλημάτων και στη λήψη αποφάσεων.
- Ασφάλεια: Η τεχνητή νοημοσύνη μπορεί να αντιμετωπίσει απειλές ασφαλείας, όπως επιθέσεις από κακόβουλα μοντέλα, εξαπάτηση με τη χρήση ψευδο-δεδομένων και άλλες ανεπιθύμητες επιπτώσεις.
- Ανεπάρκεια δεδομένων: Η μηχανική μάθηση απαιτεί συχνά μεγάλα και ποικίλα σύνολα δεδομένων, και η έλλειψη δεδομένων μπορεί να είναι πρόκληση στην ανάπτυξη ακριβών μοντέλων.
- Συμβατότητα και προσαρμογή: Η ενσωμάτωση της τεχνητής νοημοσύνης σε υπάρχουσες εφαρμογές και περιβάλλοντα μπορεί να απαιτεί αλλαγές και προσαρμογές. [9]

### 3.5 Ηθικά και νομικά θέματα

Τα ηθικά και νομικά θέματα που σχετίζονται με την τεχνητή νοημοσύνη είναι σημαντικά και απαιτούν προσεκτική αντιμετώπιση. Εδώ είναι μερικά από αυτά τα θέματα, χωρίς να αναφέρονται προηγούμενες πληροφορίες:

Ηθικά Θέματα:

1. Ανάκληση και Δικαιώματα: Τι δικαιώματα έχουν τα αυτόνομα συστήματα ΤΝ; Ποιος είναι υπεύθυνος για αποφάσεις που λαμβάνονται από αυτά;
2. Διαφάνεια: Πώς μπορούμε να εξηγήσουμε τις αποφάσεις που λαμβάνουν αυτά τα συστήματα ΤΝ, ειδικά όταν χρησιμοποιούν μηχανές μάθησης;
3. Ιδιωτικότητα: Πώς προστατεύουμε τα δεδομένα των χρηστών και την ιδιωτικότητά τους όταν χρησιμοποιούμε ΤΝ για να συλλέξουμε και αναλύσουμε πληροφορίες;
4. Αναγνώριση προσώπων: Η χρήση της αναγνώρισης προσώπων από συστήματα ΤΝ μπορεί να έχει ανησυχητικές επιπτώσεις στην ιδιωτικότητα και την ελευθερία.

## Νομικά Θέματα:

1. Ευθύνη: Ποιος φέρει την ευθύνη όταν αυτόνομα συστήματα ΤΝ προκαλούν ατυχήματα ή καταστροφές;
2. Πνευματικά Δικαιώματα: Ποια είναι τα πνευματικά δικαιώματα που σχετίζονται με τις δημιουργίες της ΤΝ, και πώς ρυθμίζονται;
3. Αναγνώριση Προσώπων: Ορισμένες χώρες έχουν θέσει περιορισμούς στη χρήση της αναγνώρισης προσώπων λόγω ανησυχιών για παραβίαση της ιδιωτικότητας.
4. Κυβερνοασφάλεια: Η ασφάλεια των συστημάτων ΤΝ και των δεδομένων τους απαιτεί νομικές προσεγγίσεις για την προστασία από επιθέσεις και παραβιάσεις. [9]

## 4. Περιβάλλον υποβοηθούμενης διαβίωσης

Η φράση "περιβάλλον υποβοηθούμενης διαβίωσης" αναφέρεται σε ένα περιβάλλον που σχεδιάζεται ή προσαρμόζεται έτσι ώστε να υποστηρίξει και να ενισχύει την ανεξαρτησία και την ποιότητα ζωής ατόμων με κάποιο είδος υποστήριξης ή βοήθειας. Συνήθως, αυτό το είδος περιβάλλοντος είναι ειδικά σχεδιασμένο για άτομα που έχουν κάποιο είδος αναπηρίας ή περιορισμένης αυτονομίας.[10]

### 4.1 Παροχές και Υπηρεσίες υποβοηθούμενης διαβίωσης

Το περιβάλλον υποβοηθούμενης διαβίωσης μπορεί να περιλαμβάνει διάφορες πτυχές και υπηρεσίες, όπως:

- Ιατρική Φροντίδα: Παροχή υπηρεσιών υγείας, ιατρικής παρακολούθησης, και διαχείρισης φαρμάκων.
- Κοινωνική Υποστήριξη: Παροχή ψυχοκοινωνικής υποστήριξης και συνομιλίας, προκειμένου να διατηρηθεί η κοινωνική σύνδεση και η ψυχολογική ευεξία.
- Βοήθεια στις Δραστηριότητες Καθημερινής Ζωής: Υποστήριξη σε καθημερινές δραστηριότητες όπως ντύσιμο, τροφοδοσία, προσωπική υγιεινή κ.λπ.
- Τεχνολογικές Λύσεις: Εφαρμογή τεχνολογίας για τη βελτίωση της αυτονομίας, όπως η χρήση ειδικών συσκευών ή λογισμικού.
- Εκπαίδευση και Επαγγελματική Υποστήριξη: Παροχή εκπαίδευσης και υποστήριξης για την ανάπτυξη δεξιοτήτων που ενισχύουν την ανεξαρτησία, καθώς και υποστήριξη στον τομέα της εργασίας.
- Πρόσβαση σε Υποδομές: Κατασκευή ή προσαρμογή του περιβάλλοντος για να εξασφαλίζεται η πρόσβαση σε όλες τις περιοχές του καθημερινού βίου.[10]

### 4.2 Εφαρμογή σε διάφορα περιβάλλοντα

Η έννοια του "περιβάλλοντος υποβοηθούμενης διαβίωσης" μπορεί να εφαρμοστεί σε διάφορα περιβάλλοντα ανάλογα με τις ανάγκες και τις προτιμήσεις του ατόμου που λαμβάνει την υποστήριξη. Κάποια από αυτά τα περιβάλλοντα περιλαμβάνουν:

- Οικιακό Περιβάλλον: Το άτομο μπορεί να ζει στο δικό του σπίτι με την υποστήριξη ειδικών υπηρεσιών, όπως οικιακή βοήθεια, ιατρική φροντίδα και κοινωνική υποστήριξη.
- Κοινοτικό Περιβάλλον: Σε μια κοινότητα, το άτομο μπορεί να ζει σε κοινοτικά συγκροτήματα ή κατοικίες με υποστήριξη, όπου παρέχονται υπηρεσίες που ενισχύουν την αυτονομία του.
- Εγκαταστάσεις Φροντίδας: Για άτομα που απαιτούν περισσότερη συνεχή υποστήριξη, όπως άτομα με προηγμένες καταστάσεις υγείας, μπορεί να είναι κατάλληλες οι εγκαταστάσεις φροντίδας.
- Σχολικά ή Εκπαιδευτικά Περιβάλλοντα: Για παιδιά ή νέους ενήλικες με ειδικές εκπαιδευτικές ανάγκες, το περιβάλλον υποβοηθούμενης διαβίωσης μπορεί να ενσωματωθεί σε εκπαιδευτικές δομές.[10]



## 4.3 Πρωτοβουλίες, υπηρεσίες και πρακτικές

Το περιβάλλον υποβοηθούμενης διαβίωσης αφορά ένα ευρύ φάσμα πρωτοβουλιών, υπηρεσιών και πρακτικών που έχουν σχεδιαστεί για να παρέχουν υποστήριξη και φροντίδα σε άτομα με διάφορες αναπηρίες ή περιορισμούς, ενθαρρύνοντας την αυτονομία τους και παρέχοντας τις προϋποθέσεις για μια ποιοτική ζωή. Ακολουθούν περισσότερες πληροφορίες:

- **Εξατομικευμένος Σχεδιασμός:** Το περιβάλλον υποβοηθούμενης διαβίωσης συχνά είναι εξατομικευμένο, λαμβάνοντας υπόψη τις ειδικές ανάγκες, προτιμήσεις και δυνατότητες κάθε ατόμου.
- **Ενίσχυση Τεχνολογίας:** Η χρήση τεχνολογίας παίζει σημαντικό ρόλο στο περιβάλλον υποβοηθούμενης διαβίωσης, παρέχοντας τεχνολογικές λύσεις για την καθημερινότητα, όπως ειδικές συσκευές, εφαρμογές λογισμικού, και άλλα.
- **Κοινότητα και Κοινωνική Ένταξη:** Η υποστήριξη δεν περιορίζεται μόνο στη φροντίδα, αλλά προωθεί επίσης τη συμμετοχή σε κοινοτικές δραστηριότητες, ενισχύοντας την κοινωνική ένταξη.
- **Εκπαίδευση και Επαγγελματική Ανάπτυξη:** Η εκπαίδευση και η επαγγελματική υποστήριξη αποτελούν σημαντικό κομμάτι, επιτρέποντας στα άτομα να αναπτύξουν δεξιότητες και να ενταχθούν στην αγορά εργασίας όποτε είναι εφικτό.
- **Οικογενειακή Στήριξη:** Συχνά, το περιβάλλον υποβοηθούμενης διαβίωσης συνεργάζεται με τις οικογένειες και τους φροντιστές για να εξασφαλίσει ολοκληρωμένη στήριξη.
- **Συνεχής Αξιολόγηση:** Η προσαρμογή των υπηρεσιών βασίζεται σε συνεχείς αξιολογήσεις των αναγκών και των εξελίξεων του ατόμου.[10]

## 5. Τεχνική Εφαρμογή

### 5.1 Τι είναι η postgresql

Το PostgreSQL (συνήθως αναφέρεται ως PostgreSQL) είναι ένα διαχρονικό και πολυδύναμο σύστημα διαχείρισης βάσεων δεδομένων (DBMS) ανοιχτού κώδικα. Αναπτύσσεται και συντηρείται από μια κοινότητα ανάπτυξης και χρησιμοποιείται ευρέως σε πολλούς τομείς, συμπεριλαμβανομένων των επιχειρηματικών εφαρμογών, του Ιούτα, της επιστημονικής έρευνας και πολλών άλλων πεδίων. Ορισμένα χαρακτηριστικά του PostgreSQL περιλαμβάνουν:

- **Σχεσιακή Βάση Δεδομένων:** Το PostgreSQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων, που σημαίνει ότι αποθηκεύει δεδομένα σε πίνακες με σχέσεις μεταξύ τους.
- **Ανοιχτού Κώδικα:** Το PostgreSQL είναι ανοιχτού κώδικα λογισμικό, προσφέροντας ελεύθερη πρόσβαση στον πηγαίο κώδικα για τροποποίηση και προσαρμογή από τον κάθε χρήστη.
- **Συμμόρφωση με τα Πρότυπα SQL:** Το PostgreSQL υποστηρίζει τη γλώσσα ερωτημάτων SQL, δίνοντας τη δυνατότητα για την ανάκτηση και την ενημέρωση δεδομένων με τη χρήση τυπικών SQL εντολών.
- **Υψηλή Αξιοπιστία και Ευελιξία:** Το PostgreSQL προσφέρει πολλές λειτουργίες υψηλής διαθεσιμότητας και ανακατανομής δεδομένων, επιτρέποντας τη δημιουργία αξιόπιστων και ευέλικτων εφαρμογών.
- **Υποστήριξη για Πολλές Πλατφόρμες:** Το PostgreSQL λειτουργεί σε πολλές λειτουργικές συσκευές, συμπεριλαμβανομένων των Linux, Windows και macOS.
- **Επέκταση με Πρόσθετα:** Το PostgreSQL υποστηρίζει πρόσθετα (extensions) που μπορούν να προσθέσουν επιπλέον λειτουργίες και δυνατότητες στο σύστημα.
- **Ευρεία Κοινότητα Ανάπτυξης:** Η κοινότητα του PostgreSQL είναι ενεργή και παρέχει υποστήριξη, επισκευές σφαλμάτων και ανάπτυξη νέων χαρακτηριστικών.

Το PostgreSQL χρησιμοποιείται ευρέως για τη διαχείριση δεδομένων σε εφαρμογές και ιστοσελίδες, καθώς προσφέρει μια ισχυρή και αξιόπιστη λύση για την αποθήκευση και την ανάκτηση πληροφοριών.[7]

### 5.2 Pgadmin

Το pgAdmin είναι ένα εργαλείο διαχείρισης της βάσης δεδομένων PostgreSQL. Σαν γραφικό περιβάλλον διαχείρισης (GUI), το pgAdmin παρέχει μια ευκολία χρήσης διεπαφή που επιτρέπει στους διαχειριστές βάσεων δεδομένων και τους προγραμματιστές να διαχειρίζονται, να προγραμματίζουν, να παρακολουθούν και να αλληλεπιδρούν με τις βάσεις δεδομένων PostgreSQL μέσω ενός γραφικού περιβάλλοντος.

Ορισμένα χαρακτηριστικά του pgAdmin περιλαμβάνουν:

- Διαχείριση Βάσης Δεδομένων: Μπορείτε να δημιουργείτε, να επεξεργάζεστε και να διαγράφετε βάσεις δεδομένων, πίνακες, δείκτες, προσαρμοσμένους τύπους δεδομένων και άλλα αντικείμενα βάσης δεδομένων.
- Ερωτήματα SQL: Μπορείτε να εκτελείτε SQL ερωτήματα στη βάση δεδομένων, να προβάλλετε τα αποτελέσματα και να αναλύετε τα δεδομένα.
- Εισαγωγή και Εξαγωγή Δεδομένων: Μπορείτε να εισάγετε και να εξάγετε δεδομένα από και προς τη βάση δεδομένων σε διάφορες μορφές, όπως CSV, Excel, SQL script κ.ά.
- Παρακολούθηση και Διαχείριση Ενημερώσεων: Μπορείτε να παρακολουθείτε την απόδοση της βάσης δεδομένων, να διαχειρίζεστε δείκτες και στατιστικές, και να παρακολουθείτε τη χρήση της μνήμης και των πόρων του συστήματος.
- Εικονική Συγχρονισμός: Το pgAdmin προσφέρει εικονική συγχρονισμό της βάσης δεδομένων, ώστε να μπορείτε να διαχειριστείτε δεδομένα παράλληλα με άλλους χρήστες.
- Συνδεσιμότητα με Πολλές Βάσεις Δεδομένων: Μπορείτε να συνδέεστε ταυτόχρονα με πολλές βάσεις δεδομένων PostgreSQL και να διαχειρίζεστε πολλαπλά έργα.

Το pgAdmin είναι ένα από τα πιο δημοφιλή εργαλεία διαχείρισης βάσεων δεδομένων PostgreSQL και βοηθά στην απλοποίηση της διαχείρισης και της ανάπτυξης βάσεων δεδομένων για εφαρμογές και ιστοσελίδες.[6]

## 5.3 DBeaver

Το DBeaver είναι ένα δημοφιλές εργαλείο διαχείρισης βάσεων δεδομένων που παρέχει ένα γραφικό περιβάλλον για την αποτελεσματική διαχείριση και ανάπτυξη διάφορων τύπων βάσεων δεδομένων. Το DBeaver είναι ανοιχτού κώδικα και υποστηρίζει πολλούς δημοφιλείς διαχειριστές βάσεων δεδομένων, συμπεριλαμβανομένων των PostgreSQL, MySQL, SQLite, Oracle, SQL Server, και πολλών άλλων.[5]

Ορισμένα χαρακτηριστικά του DBeaver περιλαμβάνουν:

- Συνδεσιμότητα με Πολλές Βάσεις Δεδομένων: Μπορείτε να συνδεθείτε ταυτόχρονα με πολλές διαφορετικές βάσεις δεδομένων και να διαχειρίζεστε τα δεδομένα τους από ένα κοινό γραφικό περιβάλλον.
- Ερωτήματα SQL: Το DBeaver παρέχει πλούσιες δυνατότητες εκτέλεσης SQL ερωτημάτων, αλλά και δυνατότητα προεπισκόπησης αποτελεσμάτων, αυτοσυμπλήρωσης κώδικα SQL και πολλά άλλα.
- Σχεδίαση Βάσης Δεδομένων: Μπορείτε να σχεδιάσετε τη δομή της βάσης δεδομένων σας με γραφικά εργαλεία και να δημιουργήσετε πίνακες, συσχετίσεις και άλλα αντικείμενα.
- Εισαγωγή και Εξαγωγή Δεδομένων: Μπορείτε να εισάγετε και να εξάγετε δεδομένα από/προς τη βάση δεδομένων σε διάφορες μορφές, όπως CSV, Excel, JSON και πολλές άλλες.
- Διαχείριση Σχήματος: Μπορείτε να διαχειριστείτε τα σχήματα της βάσης δεδομένων σας, να δημιουργήσετε και να επεξεργαστείτε διάφορα αντικείμενα όπως πίνακες, δείκτες, και προσαρμοσμένους τύπους δεδομένων.
- Αναφορές και Αναλύσεις: Μπορείτε να δημιουργήσετε αναφορές και να αναλύσετε τα δεδομένα σας με τη χρήση εργαλείων αναφοράς και επεξεργασίας δεδομένων.

- Επέκταση με Πρόσθετα: Το DBeaver υποστηρίζει πρόσθετα που μπορείτε να εγκαταστήσετε για να προσθέσετε επιπλέον λειτουργίες και δυνατότητες.

Το DBeaver είναι διαθέσιμο για πολλές πλατφόρμες, συμπεριλαμβανομένων των Windows, macOS και διαφόρων διανομών Linux. Είναι ένα ισχυρό εργαλείο για επαγγελματίες που ασχολούνται με τη διαχείριση βάσεων δεδομένων και την ανάπτυξη εφαρμογών.

## 5.4 Τι είναι τα queries

Τα "queries" (ερωτήματα) είναι εντολές που χρησιμοποιούνται σε βάσεις δεδομένων για την ανάκτηση, την εισαγωγή, την ενημέρωση ή τη διαγραφή δεδομένων. Τα ερωτήματα γράφονται χρησιμοποιώντας τη γλώσσα SQL (Structured Query Language), η οποία είναι η γλώσσα που χρησιμοποιείται για τη διαχείριση δεδομένων σε σχεσιακές βάσεις δεδομένων, όπως η PostgreSQL, MySQL, Oracle, SQL Server και άλλες.

Τα ερωτήματα SQL μπορούν να κάνουν τα εξής:

- Επιλογή (SELECT) δεδομένων: Ανακτούν δεδομένα από έναν πίνακα ή περισσότερους πίνακες με βάση κάποιες συνθήκες. Παρέχουν αποτελέσματα που πληρούν τα κριτήρια του ερωτήματος.
- Εισαγωγή (INSERT) δεδομένων: Προσθέτουν νέα δεδομένα σε έναν πίνακα.
- Ενημέρωση (UPDATE) δεδομένων: Αλλάζουν τιμές σε υπάρχουσες εγγραφές στον πίνακα με βάση κάποιες συνθήκες.
- Διαγραφή (DELETE) δεδομένων: Αφαιρούν εγγραφές από έναν πίνακα με βάση κάποιες συνθήκες.[3]

## 5.5 React

Η React είναι μια βιβλιοθήκη JavaScript για τη δημιουργία χρήστη διεπαφών (UI). Αναπτύχθηκε από την Facebook και είναι σχεδιασμένο για τη δημιουργία αποδοτικών και ευέλικτων εφαρμογών. Η React είναι ειδικά σχεδιασμένο για τον χειρισμό δυναμικών UI, καθιστώντας ευκολότερο τον προγραμματισμό των διάφορων καταστάσεων της εφαρμογής σας.[1]

Ένα από τα κύρια χαρακτηριστικά της React είναι η χρήση συνιστωσών (components). Οι συνιστώσες είναι ανακλώμενα (reusable) κομμάτια κώδικα που αναλαμβάνουν συγκεκριμένες λειτουργίες. Η δομή της React επιτρέπει τον εύκολο διαχωρισμό του κώδικα σε συνιστώσες, καθιστώντας τον κώδικα πιο ευανάγνωστο και συντηρήσιμο.

Επίσης, η React χρησιμοποιεί ένα μοντέλο ανανέωσης (reconciliation) που επιτρέπει την αποδοτική ενημέρωση μόνο των απαραίτητων μερών της διεπαφής όταν ο κατάσταση της εφαρμογής αλλάζει. Αυτό συμβάλλει στην εξασφάλιση υψηλής απόδοσης της εφαρμογής, ακόμη και σε περιπτώσεις μεγάλου όγκου δεδομένων.

Η React είναι επίσης ενσωματωμένο σε πολλά εργαλεία και βιβλιοθήκες για τη διευκόλυνση της ανάπτυξης, όπως το Redux για τη διαχείριση της κατάστασης, η React Router για τη διαχείριση των διαδρομών (routes) και πολλά άλλα.[2]

## 5.6 ExpressJS

Το Express.js είναι ένα ελαφρύ, ευέλικτο και ανοικτού κώδικα framework για τον χειρισμό αιτημάτων (requests) και αποκρίσεων (responses) σε εφαρμογές web και υπηρεσίες. Είναι βασισμένο στη γλώσσα JavaScript και είναι σχεδιασμένο να είναι απλό στη χρήση και να επιτρέπει την εύκολη κατασκευή διαδικτυακών εφαρμογών.

Οι βασικές λειτουργίες του Express.js περιλαμβάνουν τον ορισμό διαδρομών (routes) για την ανταπόκριση σε συγκεκριμένα αιτήματα HTTP, τη δυνατότητα χρήσης middleware για τον χειρισμό προεπεξεργασίας των αιτημάτων και αποκρίσεων, καθώς και τη δυνατότητα επέκτασης των λειτουργιών του με πρόσθετα (plugins) και middleware.

Το Express.js συχνά χρησιμοποιείται σε συνδυασμό με άλλες τεχνολογίες όπως η Node.js για τη δημιουργία ολοκληρωμένων εφαρμογών. Είναι ιδιαίτερα δημοφιλές στην κοινότητα των προγραμματιστών JavaScript για τη διαχείριση των server-side λειτουργιών σε web εφαρμογές.[4]

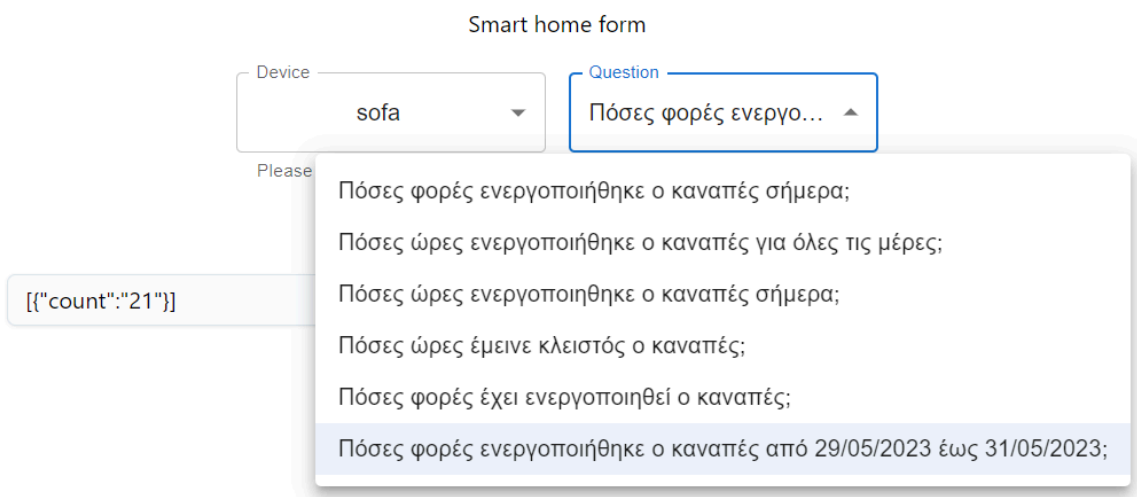
## 5.7 Υλοποίηση

### 5.7.1 Εφαρμογή

Δημιουργήθηκε εφαρμογή που αποτελείται από Front end κομμάτι δημιουργημένο από react και back end με expressJS. Η εφαρμογή έχει ως σκοπό να δίνει απαντήσεις στους χρήστες που την χρησιμοποιούν που αφορούν έξυπνες συσκευές μέσα σε ένα έξυπνο σπίτι. Οι απαντήσεις βασίζονται στην βάση δεδομένων που έχει δημιουργηθεί καθώς στον Back end έχει γίνει σύνδεση μεταξύ εφαρμογής και Βάσης δεδομένων.

Για την λειτουργία της εφαρμογής χρειάζεται να ξεκινήσουν και τα δύο κομμάτια Front end και Back end:

- Back end: node app.js
- Front end: npm run start



## 5.7.2 Δημιουργία Πινάκων

Για τον σκοπό της συγκεκριμένης διπλωματικής θα δημιουργήθηκαν 3 πίνακες αναπαριστώντας έξυπνες συσκευές μέσα σε ένα χώρο. Αυτοί οι τρεις πίνακες είναι το ψυγείο, η τηλεόραση και ο καναπές.

```
CREATE TABLE fridge (  
    timestamp timestamp NOT null,  
    state VARCHAR ( 4 ) NOT null  
);  
CREATE TABLE sofa (  
    timestamp timestamp NOT null,  
    state VARCHAR ( 3 ) NOT null  
);  
CREATE TABLE tv (  
    timestamp timestamp NOT null,  
    state VARCHAR ( 3 ) NOT null  
);
```

## 5.7.3 Εισαγωγή Δεδομένων

Για κάθε έναν πίνακα εισήχθησαν δεδομένα προκειμένου να υπάρξουν αποτελέσματα στα ερωτήματα που θα χρησιμοποιηθούν παρακάτω.

### Δεδομένα για την τηλεόραση:

```
insert into tv (timestamp,state)  
values  
( '2023-11-01 10:00:36.063', 'on' ),  
( '2023-11-01 10:05:36.063', 'off' ),  
( '2023-11-01 10:00:36.063', 'on' ),  
( '2023-11-02 13:05:36.063', 'on' ),  
( '2023-11-02 20:20:36.063', 'off' ),  
( '2023-11-02 00:00:36.063', 'on' ),  
( '2023-11-02 10:00:36.063', 'on' ),  
( '2023-11-02 10:05:36.063', 'off' ),  
( '2023-11-02 10:20:36.063', 'on' ),  
( '2023-11-02 10:40:36.063', 'off' )
```

### Εκτέλεση

Name	Value
Updated Rows	10
Query	insert into tv (timestamp,state)
	values
	('2023-11-01 10:00:36.063','on'),
	('2023-11-01 10:05:36.063','off'),
	('2023-11-01 10:00:36.063','on'),
	('2023-11-02 13:05:36.063','on'),
	('2023-11-02 20:20:36.063','off'),
	('2023-11-02 00:00:36.063','on'),
	('2023-11-02 10:00:36.063','on'),
	('2023-11-02 10:05:36.063','off'),
	('2023-11-02 10:20:36.063','on'),
	('2023-11-02 10:40:36.063','off')
Start time	Wed Nov 01 20:49:49 EET 2023
Finish time	Wed Nov 01 20:49:49 EET 2023

### Δεδομένα για τον καναπέ:

```
insert into sofa (timestamp,state)
values
('2023-11-01 11:00:36.063', 'on'),
('2023-11-01 11:05:36.063', 'off'),
('2023-11-02 01:00:36.063', 'on'),
('2023-11-02 01:05:36.063', 'off'),
('2023-11-02 12:00:36.063', 'on'),
('2023-11-02 12:05:36.063', 'off'),
('2023-11-02 15:00:36.063', 'on'),
('2023-11-02 15:05:36.063', 'off'),
('2023-11-02 20:20:36.063', 'on'),
('2023-11-02 20:40:36.063', 'off')
```

### Εκτέλεση

Name	Value
Updated Rows	10
Query	insert into sofa (timestamp,state)
	values
	('2023-11-01 11:00:36.063','on'),
	('2023-11-01 11:05:36.063','off'),
	('2023-11-02 01:00:36.063','on'),
	('2023-11-02 01:05:36.063','off'),
	('2023-11-02 12:00:36.063','on'),
	('2023-11-02 12:05:36.063','off'),
	('2023-11-02 15:00:36.063','on'),
	('2023-11-02 15:05:36.063','off'),
	('2023-11-02 20:20:36.063','on'),
	('2023-11-02 20:40:36.063','off')
Start time	Wed Nov 01 20:51:03 EET 2023
Finish time	Wed Nov 01 20:51:03 EET 2023

### Δεδομένα για το ψυγείο:

```
insert into fridge (timestamp,state)
values
('2023-11-01 11:00:36.063', 'open'),
('2023-11-01 11:05:36.063', 'half'),
('2023-11-01 11:10:36.063', 'off'),
('2023-11-02 10:05:36.063', 'open'),
('2023-11-02 10:10:36.063', 'half'),
('2023-11-02 10:11:36.063', 'off'),
('2023-11-02 12:00:36.063', 'open'),
('2023-11-02 12:05:36.063', 'half'),
('2023-11-02 12:10:36.063', 'off'),
('2023-11-02 20:40:36.063', 'open')
```

### Εκτέλεση

Name	Value
Updated Rows	10
Query	insert into fridge (timestamp,state) values ( <code>'2023-11-01 11:00:36.063'</code> , <code>'open'</code> ), ( <code>'2023-11-01 11:05:36.063'</code> , <code>'half'</code> ), ( <code>'2023-11-01 11:10:36.063'</code> , <code>'off'</code> ), ( <code>'2023-11-02 10:05:36.063'</code> , <code>'open'</code> ), ( <code>'2023-11-02 10:10:36.063'</code> , <code>'half'</code> ), ( <code>'2023-11-02 10:11:36.063'</code> , <code>'off'</code> ), ( <code>'2023-11-02 12:00:36.063'</code> , <code>'open'</code> ), ( <code>'2023-11-02 12:05:36.063'</code> , <code>'half'</code> ), ( <code>'2023-11-02 12:10:36.063'</code> , <code>'off'</code> ), ( <code>'2023-11-02 20:40:36.063'</code> , <code>'open'</code> )
Start time	Wed Nov 01 20:52:15 EET 2023
Finish time	Wed Nov 01 20:52:15 EET 2023

## 5.7.4 Εκτέλεση Ερωτημάτων

1. Πόσες φορές ενεργοποιήθηκε ο καναπές σήμερα;

```
SELECT count(timestamp)
FROM sofa
WHERE timestamp::date = now()::date and state = 'on';
```

### Εκτέλεση από DBeaver

123	count
1	1

### Εκτέλεση από εφαρμογή



### Smart home form

Device  Question

Please select a device Please select a question

Answer:

```
[{"count": "1"}]
```

2. Πόσες φορές ενεργοποιήθηκε η τηλεόραση σήμερα;

```
SELECT count(timestamp)
FROM tv
WHERE timestamp::date = now()::date and state = 'on';
```

Εκτέλεση από DBeaver

	count
1	2

Εκτέλεση από εφαρμογή

### Smart home form

Device  Question

Please select a device Please select a question

Answer:

```
[{"count": "2"}]
```

3. Πόσες ώρες ενεργοποιήθηκε ο καναπές για όλες τις μέρες

```
select Sum(PreviousWord-timestamp) as sumTime
from (
SELECT LEAD(timestamp) OVER ( ORDER BY timestamp ) AS PreviousWord,timestamp,
state
FROM sofa
order by timestamp asc) as sofa_new
where PreviousWord is not null and state = 'on'
```

Εκτέλεση από DBeaver

	sumtime
1	02:45:00

Εκτέλεση από εφαρμογή

#### Smart home form

Device sofa	Question Πόσες ώρες ενεργοπ...
Please select a device	Please select a question

Answer:

```
[{"sumtime":{"hours":2,"minutes":45}}]
```

#### 4. Πόσες ώρες ενεργοποιήθηκε ο καναπές σήμερα;

```
select Sum(PreviousWord-timestamp) as sumTime
from (
SELECT LEAD(timestamp) OVER ( ORDER BY timestamp ) AS PreviousWord,timestamp,
state
FROM sofa
where timestamp::date = now()::date
order by timestamp asc) as sofa_new
where PreviousWord is not null and state = 'on'
```

#### Εκτέλεση από DBeaver

sumtime
1 00:05:00

#### Εκτέλεση από εφαρμογή

#### Smart home form

Device sofa	Question Πόσες ώρες ενεργοπ...
Please select a device	Please select a question

Answer:

```
[{"sumtime":{"minutes":5}}]
```

#### 5.Πόσες φορές έμεινε μισό ανοιχτό το ψυγείο σήμερα;

```
SELECT count(timestamp)
FROM fridge
WHERE timestamp::date = now()::date and state = 'half';
```

#### Εκτέλεση από DBeaver

count
1 1

#### Εκτέλεση από εφαρμογή

### Smart home form

Device	fridge	Question	Πόσες φορές έμεινε ...
	▼		▼
Please select a device		Please select a question	

Answer:

```
[{"count": "1"}]
```

6. Πόσες φορές έμεινε ανοιχτό το ψυγείο σήμερα;

```
SELECT count(timestamp)
FROM fridge
WHERE timestamp::date = now()::date and state = 'open';
```

Εκτέλεση από DBeaver

123	count
1	1

Εκτέλεση από εφαρμογή

### Smart home form

Device	fridge	Question	Πόσες φορές έμεινε ...
	▼		▼
Please select a device		Please select a question	

Answer:

```
[{"count": "1"}]
```

7. Πόσες φορές έχει ανοίξει το ψυγείο;

```
SELECT count(*)
FROM fridge
WHERE state = 'open';
```

Εκτέλεση από DBeaver

123	count
1	13

Εκτέλεση από εφαρμογή

### Smart home form

Device:  Please select a device

Question:  Please select a question

Answer:

```
[{"count": "13"}]
```

8. Πόσες ώρες ενεργοποιήθηκε η TV για όλες τις μέρες;

```
select Sum(PreviousWord-timestamp) as sumTime
from (
SELECT LEAD(timestamp) OVER ( ORDER BY timestamp ) AS
PreviousWord,timestamp, state
FROM tv
order by timestamp asc) as tv_new
where PreviousWord is not null and state = 'on'
```

Εκτέλεση από DBeaver

	sumtime
1	35:30:00

Εκτέλεση από εφαρμογή

### Smart home form

Device:  Please select a device

Question:  Please select a question

Answer:

```
[{"sumtime":{"hours":35,"minutes":30}}]
```

9. Πόσες ώρες ενεργοποιήθηκε το ψυγείο για όλες τις μέρες;

```
select Sum(PreviousWord-timestamp) as sumTime
from (
SELECT LEAD(timestamp) OVER ( ORDER BY timestamp ) AS
PreviousWord,timestamp, state
FROM fridge
order by timestamp asc) as fridge_new
where PreviousWord is not null and state = 'open'
```

Εκτέλεση από DBeaver

🔒	🕒 sumtime
1	36 days 29:30:00

### Εκτέλεση από εφαρμογή

Smart home form

Device	Question
fridge	Πόσες ώρες ενεργοπ...
Please select a device	Please select a question

Answer:

```
[{"sumtime":{"days":36,"hours":29,"minutes":30}}]
```

10. Πόσες ώρες έμεινε κλειστό το ψυγείο;

```
select Sum(PreviousWord-timestamp) as sumTime
from (
SELECT LEAD(timestamp) OVER ( ORDER BY timestamp ) AS
PreviousWord,timestamp, state
FROM fridge
order by timestamp asc) as fridge_new
where PreviousWord is not null and state = 'off'
```

### Εκτέλεση από DBeaver

🔒	🕒 sumtime
1	123:32:00

### Εκτέλεση από εφαρμογή

Smart home form

Device	Question
fridge	Πόσες ώρες έμεινε κ...
Please select a device	Please select a question

Answer:

```
[{"sumtime":{"hours":123,"minutes":32}}]
```

11. Πόσες ώρες έμεινε κλειστή η τηλεόραση;

```
select Sum(PreviousWord-timestamp) as sumTime
from (
SELECT LEAD(timestamp) OVER ( ORDER BY timestamp ) AS
PreviousWord,timestamp, state
```

```
FROM tv
order by timestamp asc) as tv_new
where PreviousWord is not null and state = 'off'
```

### Εκτέλεση από DBeaver

	sumtime
1	38 days 46:50:00

### Εκτέλεση από εφαρμογή

#### Smart home form

Device	Question
tv	Πόσες ώρες έμεινε κ...
Please select a device	Please select a question

Answer:

```
[{"sumtime":{"days":38,"hours":46,"minutes":50}}]
```

12. Πόσες ώρες έμεινε κλειστός ο καναπές;

```
select Sum(PreviousWord-timestamp) as sumTime
from (
SELECT LEAD(timestamp) OVER ( ORDER BY timestamp ) AS
PreviousWord,timestamp, state
FROM sofa
order by timestamp asc) as sofa_new
where PreviousWord is not null and state = 'off'
```

### Εκτέλεση από DBeaver

	sumtime
1	44 days 198:55:00

### Εκτέλεση από εφαρμογή

#### Smart home form

Device	Question
sofa	Πόσες ώρες έμεινε κ...
Please select a device	Please select a question

Answer:

```
[{"sumtime":{"days":44,"hours":198,"minutes":55}}]
```

13. Πόσες φορές έχει ενεργοποιηθεί ο καναπές;

```
SELECT count(*)
FROM sofa
WHERE state = 'on';
```

#### Εκτέλεση από DBeaver

	123 count
1	21

#### Εκτέλεση από εφαρμογή

##### Smart home form

Device	sofa	Question	Πόσες φορές έχει ενε...
	<input type="text"/>		<input type="text"/>
	Please select a device		Please select a question

Answer:

```
[{"count": "21"}]
```

14. Πόσες φορές έχει ανοίξει η τηλεόραση;

```
SELECT count(*)
FROM tv
WHERE state = 'on';
```

#### Εκτέλεση από DBeaver

	123 count
1	12

#### Εκτέλεση από εφαρμογή

##### Smart home form

Device	tv	Question	Πόσες φορές έχει αν...
	<input type="text"/>		<input type="text"/>
	Please select a device		Please select a question

Answer:

```
[{"count": "12"}]
```

15. Πόσες φορές ενεργοποιήθηκε ο καναπές από 29/05/2023 έως 31/05/2023;

```
SELECT count(timestamp)
FROM sofa
WHERE timestamp::date > '29-05-2023'::date and timestamp::date <
'31-05-2024'::date and state = 'on';
```

#### Εκτέλεση από DBeaver

123	count
1	21

### Εκτέλεση από εφαρμογή

Smart home form

Device	Question
sofa	Πόσες φορές ενεργο...
Please select a device	Please select a question

Answer:

```
[{"count": "21"}]
```



## Πηγές

- [1] React - <https://react.dev/>
- [2] React from GitHub - <https://github.com/facebook/react>
- [3] Queries - <https://el.wikipedia.org/wiki/Queries>
- [4] ExpressJS - <https://expressjs.com/>
- [5] DBeaver - <https://dbeaver.io/>
- [6] Pgadmin - <https://www.pgadmin.org/>
- [7] Postgresql - <https://www.postgresql.org/>
- [8] P. Rashidi and A. Mihailidis, "A Survey on Ambient-Assisted Living Tools for Older Adults," IEEE J. Biomed. Health Inform., May 2013.
- [9] Artificial Intelligence: Foundations of Computational Agents, David L. Poole & Alan K. Mackworth
- [10] Gabriella Facchinetti, Giorgia Petrucci, Beatrice Albanesi, Maria Grazia De Marinis and Michela Piredda "Can Smart Home Technologies Help Older Adults Manage Their Chronic Condition? A Systematic Literature Review", 10 Jan 2023.

# Παράρτημα 1

## Front End

### app.js

```
import logo from './logo.svg';
import './App.css';
import SelectTextFields from './components/select'
function App() {
  return (
    <div style={{display:'flex',justifyContent: 'center'}}>
      <div className="App" style={{ width: '50%' }}>
        <p> Smart home form</p>
        <SelectTextFields />
      </div>
    </div>
  );
}

export default App;
```

### index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
reportWebVitals();
```

### select.js

```
import * as React from 'react';
import Box from '@mui/material/Box';
import { TextField } from '@mui/material';
import Textarea from '@mui/joy/Textarea';
import MenuItem from '@mui/material/MenuItem';
import axios from 'axios';

const questions = {
  "fridge": [
    {
```

```

        value: "f1",
        label: "Πόσες φορές έμεινε μισό ανοιχτό το ψυγείο σήμερα;"
    }, {
        value: "f2",
        label: "Πόσες φορές έμεινε ανοιχτό το ψυγείο σήμερα;"
    }, {
        value: "f3",
        label: "Πόσες φορές έχει ανοίξει το ψυγείο;"
    }, {
        value: "f4",
        label: "Πόσες ώρες ενεργοποιήθηκε το ψυγείο για όλες τις μέρες;"
    }, {
        value: "f5",
        label: "Πόσες ώρες έμεινε κλειστό το ψυγείο;"
    }
],
"tv": [
    {
        value: "t1",
        label: "Πόσες φορές ενεργοποιήθηκε η τηλεόραση σήμερα;"
    }, {
        value: "t2",
        label: "Πόσες ώρες ενεργοποιήθηκε η TV για όλες τις μέρες;"
    }, {
        value: "t3",
        label: "Πόσες ώρες έμεινε κλειστή η τηλεόραση;"
    }, {
        value: "t4",
        label: "Πόσες φορές έχει ανοίξει η τηλεόραση;"
    }
],
"sofa": [
    {
        value: "s1",
        label: "Πόσες φορές ενεργοποιήθηκε ο καναπές σήμερα;"
    }, {
        value: "s2",
        label: "Πόσες ώρες ενεργοποιήθηκε ο καναπές για όλες τις μέρες;"
    }, {
        value: "s3",
        label: "Πόσες ώρες ενεργοποιήθηκε ο καναπές σήμερα;"
    }, {
        value: "s4",
        label: "Πόσες ώρες έμεινε κλειστός ο καναπές;"
    }
],

```

```

    {
      value: "s5",
      label: "Πόσες φορές έχει ενεργοποιηθεί ο καναπές;"
    },
    {
      value: "s6",
      label: "Πόσες φορές ενεργοποιήθηκε ο καναπές από 29/05/2023 έως 31/05/2023;"
    }
  ]
}
const devices = [
  {
    value: 'fridge',
    label: 'fridge',
  },
  {
    value: 'tv',
    label: 'tv',
  },
  {
    value: 'sofa',
    label: 'sofa',
  }
];

export default function SelectTextFields() {
  const [questionsSel, setQuestionsSel] = React.useState([])
  const [selectedDevice, setSelectedDevice] = React.useState()
  const [answer, setAnswer] = React.useState('Answer....')
  const selectDevice = (value) => {
    if (value.target.dataset.value) {
      setSelectedDevice(value.target.dataset.value)
      setQuestionsSel(questions[value.target.dataset.value])
    }
  }
  const selectQuestion = async (value) => {
    if (value.target.dataset.value) {
      try {
        const response = await
axios.get(`http://localhost:3001/${selectedDevice}/${value.target.dataset.va
lue}`);

        if(response && response.data){
          setAnswer(JSON.stringify(response.data))
          console.log(response.data)
        }
      } catch (error) {
        console.error(error);
      }
    }
  }
}

```

```

    }
  }
}
return (
  <Box
    component="form"
    sx={{
      '& .MuiTextField-root': { m: 1, width: '25ch' },
    }}
    noValidate
    autoComplete="off"
  >
    <div>
      <TextField
        id="outlined-select-currency"
        select
        label="Device"
        onClick={selectDevice}
        helperText="Please select a device"
      >
        {devices.map((option) => (
          <MenuItem key={option.value} value={option.value}>
            {option.label}
          </MenuItem>
        ))}
      </TextField>
      <TextField
        id="outlined-select-currency"
        select
        onClick={selectQuestion}
        disabled={!selectedDevice}
        label="Question"
        helperText="Please select a question"
      >
        {questionsSel.map((option) => (
          <MenuItem key={option.value} value={option.value}>
            {option.label}
          </MenuItem>
        ))}
      </TextField>
    </div>
    <div>
      <p>Answer:</p>
      <Textarea
        disabled
        style={{color:'black'}}
        value={answer}
      />
    </div>
  </div>
)

```

```

        </div>
    </Box>
  );
}

```

## Back End

app.js

```

const express = require('express');
const client = require('./lib/config').pool;
const cors = require('cors');

client.connect();

const app = express();
const PORT = 3001;

app.use(cors());
app.options('*', cors());

app.get('/:device/:question', async (req, res) => {
  const { device, question } = req.params
  let data = []
  if (device === 'sofa') {
    if (question === 's1') {
      data = await client.query(`SELECT count(timestamp)
                                FROM sofa
                                WHERE timestamp::date = now()::date
                                and state = 'on';`);
    } else if (question === 's2') {
      data = await client.query(`select Sum(PreviousWord-timestamp) as
sumTime
                                from (
                                SELECT LEAD(timestamp) OVER ( ORDER
                                BY timestamp ) AS PreviousWord,timestamp, state
                                FROM sofa
                                order by timestamp asc) as sofa_new
                                where PreviousWord is not null and
                                state = 'on';`);
    } else if (question === 's3') {
      data = await client.query(`select Sum(PreviousWord-timestamp) as
sumTime
                                from (
                                SELECT LEAD(timestamp) OVER ( ORDER
                                BY timestamp ) AS PreviousWord,timestamp, state
                                FROM sofa
                                where timestamp::date = now()::date

```

```

                                order by timestamp asc) as sofa_new
                                where PreviousWord is not null and
state = 'on';`);
    } else if (question === 's4') {
        data = await client.query(`select Sum(PreviousWord-timestamp) as
sumTime
                                from (
                                SELECT LEAD(timestamp) OVER ( ORDER
BY timestamp ) AS PreviousWord,timestamp, state
                                FROM sofa
                                order by timestamp asc) as sofa_new
                                where PreviousWord is not null and
state = 'off';`);
    } else if (question === 's5') {
        data = await client.query(`SELECT count(*)
                                FROM sofa
                                WHERE state = 'on';`);
    } else if (question === 's5') {
        data = await client.query(`SELECT count(timestamp)
                                FROM sofa
                                WHERE timestamp::date >
'29-05-2023'::date and timestamp::date < '31-05-2024'::date and state =
'on';`);
    }
    } else if (device === 'tv') {
        if (question === 't1') {
            data = await client.query(`SELECT count(timestamp)
                                FROM tv
                                WHERE timestamp::date = now()::date
and state = 'on';`);
        } else if (question === 't2') {
            data = await client.query(`select Sum(PreviousWord-timestamp) as
sumTime
                                from (
                                SELECT LEAD(timestamp) OVER ( ORDER
BY timestamp ) AS PreviousWord,timestamp, state
                                FROM tv
                                order by timestamp asc) as tv_new
                                where PreviousWord is not null and
state = 'on';`);
        } else if (question === 't3') {
            data = await client.query(`select Sum(PreviousWord-timestamp) as
sumTime
                                from (
                                SELECT LEAD(timestamp) OVER ( ORDER
BY timestamp ) AS PreviousWord,timestamp, state
                                FROM tv
                                order by timestamp asc) as tv_new

```

```

                                where PreviousWord is not null and
state = 'off';`);
    } else if (question === 't4') {
        data = await client.query(`SELECT count(*)
                                FROM tv
                                WHERE state = 'on';`);
    }
    } else if (device === 'fridge') {
        if (question === 'f1') {
            data = await client.query(`SELECT count(timestamp)
                                    FROM fridge
                                    WHERE timestamp::date = now()::date
and state = 'half';`);
        } else if (question === 'f2') {
            data = await client.query(`SELECT count(timestamp)
                                    FROM fridge
                                    WHERE timestamp::date = now()::date
and state = 'open';`);
        } else if (question === 'f3') {
            data = await client.query(`SELECT count(*)
                                    FROM fridge
                                    WHERE state = 'open';`);
        } else if (question === 'f4') {
            data = await client.query(`select Sum(PreviousWord-timestamp) as
sumTime
                                from (
                                SELECT LEAD(timestamp) OVER ( ORDER
BY timestamp ) AS PreviousWord,timestamp, state
                                FROM fridge
                                order by timestamp asc) as
fridge_new
                                where PreviousWord is not null and
state = 'open';`);
        } else if (question === 'f5') {
            data = await client.query(`select Sum(PreviousWord-timestamp) as
sumTime
                                from (
                                SELECT LEAD(timestamp) OVER ( ORDER
BY timestamp ) AS PreviousWord,timestamp, state
                                FROM fridge
                                order by timestamp asc) as
fridge_new
                                where PreviousWord is not null and
state = 'off';`);
        }
    }
    res.status(200);
    res.send(data.rows);

```



```
});  
  
app.listen(PORT, (error) => {  
  if (!error)  
    console.log("Server is Successfully Running, and App is listening on  
port " + PORT)  
  else  
    console.log("Error occurred, server can't start", error);  
}  
);
```

config.js

```
const { Pool } = require('pg');  
  
exports.pool = new Pool({  
  host: '127.0.0.1',  
  database: 'postgres',  
  user: 'postgres',  
  password: '1234',  
  port: 5432,  
});
```