



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
ΑΙΣΘΗΤΗΡΑΣ ΦΥΣΙΚΗΣ ΚΑΤΑΣΤΑΣΗΣ ΜΕ  
ΤΗΝ ΧΡΗΣΗ ESP32



Του φοιτητή  
Ευάγγελου Σκοτίδης  
Αρ. Μητρώου:517127

Επιβλέπων  
Δρ. Κυριάκος Τσιακμάκης

Ημερομηνία 14/1/2025

Τίτλος Δ.Ε. : Αισθητήρας φυσικής κατάστασης με την χρήση του esp32

Κωδικός Δ.Ε. 24158

Όνοματεπώνυμο φοιτητή : Σκοτίδης Ευάγγελος

Όνοματεπώνυμο εισηγητή: Κυριάκος Τσιακμάκης

Ημερομηνία ανάληψης Δ.Ε. 19/3/2024

Ημερομηνία περάτωσης Δ.Ε. 19/3/2026

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

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

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



## Πρόλογος

Η παρούσα Διπλωματική Εργασία (Δ.Ε.) στοχεύει στη δημιουργία ενός συστήματος για τη μέτρηση του καρδιακού παλμού ατόμων, με ιδιαίτερη έμφαση στους αθλητές. Ο κύριος σκοπός της επιλογής αυτής της θεματολογίας είναι να βοηθήσει τους αθλητές να κατανοήσουν τα όριά τους και να αποτρέψουν πιθανά προβλήματα υγείας, τόσο βραχυπρόθεσμα όσο και μακροπρόθεσμα. Επιπλέον, το σύστημα αυτό συμβάλλει στην επίτευξη των στόχων τους με τον πιο ασφαλή και αποτελεσματικό τρόπο. Η ανάγκη για τέτοια συστήματα αναδείχθηκε πρόσφατα, ειδικά μετά το περιστατικό στο Euro 2020, όπου ο Δανός ποδοσφαιριστής Κρίστιαν Έρικσεν κατέρρευσε κατά τη διάρκεια του αγώνα λόγω καρδιακών προβλημάτων. Αυτό το γεγονός ευαισθητοποίησε την αθλητική κοινότητα και το ευρύτερο κοινό σχετικά με την αναγκαιότητα παρακολούθησης της καρδιακής υγείας των αθλητών. Το σύστημα που θα μελετηθεί στην Δ.Ε, αποσκοπεί να προσφέρει ένα εργαλείο πρόληψης και διαχείρισης της υγείας, το οποίο είναι κρίσιμο για την ασφάλεια και την απόδοση των αθλητών, προλαμβάνοντας σοβαρά περιστατικά και προάγοντας τη γενική ευεξία τους. Επομένως, θα δημιουργηθεί ένα σύστημα μέτρησης και καταγραφής δεδομένων του καρδιακού παλμού ενός αθλητή, με την χρήση του ESP32 και ενός αισθητήρα παλμών.

## Περίληψη

Η διπλωματική εργασία επικεντρώνεται στην ανάπτυξη ενός μετρητή καρδιακού ρυθμού για αθλητές, χρησιμοποιώντας την πλατφόρμα ESP32.

Ένας πολύ σημαντικός δείκτης της φυσικής κατάστασης και της υγείας του αθλητή είναι ο καρδιακός ρυθμός και η συνεχής παρακολούθησή του είναι πολύ σημαντική για την απόδοση και την υγεία. Για να ολοκληρωθεί αυτή η εργασία χρησιμοποιήθηκε ο μικροελεγκτής ESP32, ο οποίος έχει ενσωματωμένη σύνδεση Wi-Fi και bluetooth. Μέσω αυτής της πλατφόρμας μπορεί να δημιουργηθεί ένα αξιόπιστο και ταυτόχρονα φορητό σύστημα παρακολούθησης καρδιακού ρυθμού.

Ο μετρητής καρδιακού ρυθμού λαμβάνει τα σήματα του παλμού από έναν αισθητήρα που τοποθετείται στο σώμα του αθλητή, στη συνέχεια τα δεδομένα επεξεργάζονται από την πλατφόρμα και εμφανίζονται σε smartphone, σε υπολογιστή ή σε κάποια σταθερή συσκευή.

Ο καρδιακός παλμός παρουσιάζεται σε πραγματικό χρόνο και μπορεί με αυτόν τον τρόπο να παρέχει στατιστικά στοιχεία σχετικά με την καρδιακή υγεία του ανθρώπου που χρησιμοποιεί τον μετρητή και να τον ειδοποιήσει σε περίπτωση που ο καρδιακός του ρυθμός υπερβαίνει κάποιο όριο, προσφέροντάς του έτσι μία προειδοποίηση σε περίπτωση προβλήματος.

Σε αυτή την εργασία παρουσιάζεται ένας αξιόπιστος και ταυτόχρονα φορητός τρόπος παρακολούθησης του καρδιακού ρυθμού για αθλητές μέσω της πλατφόρμας ESP32 και ο τρόπος καταγραφής του.

# «Physical fitness sensor, using ESP32»

«Evangelos Skotidis»

## **Abstract**

This thesis focuses on the development of a heart rate monitor for athletes using the ESP32 platform. Heart rate is a critical indicator of an athlete's physical condition and health, and continuous monitoring is essential for both performance and well-being. To complete this project, the ESP32 microcontroller was utilized due to its built-in Wi-Fi and Bluetooth connectivity. This platform enables the creation of a reliable yet portable heart rate monitoring system.

The heart rate monitor captures pulse signals from a sensor placed on the athlete's body. These signals are processed by the ESP32 and then displayed on various devices, such as smartphones, computers, or other fixed devices.

The system provides real-time heart rate data and can generate statistical insights regarding the user's cardiovascular health. Additionally, it can send alerts if the heart rate exceeds a predefined threshold, offering early warnings in case of potential issues.

This thesis presents an efficient and portable method for monitoring heart rate in athletes using the ESP32 platform and outlines the process of recording and analyzing the collected data.

## Ευχαριστίες

Θα ήθελα αρχικά να ευχαριστήσω πολύ τον καθηγητή μου, Κυριάκο Τσιακμάκη, για την καθοδήγηση και την στήριξη κατά την διάρκεια αυτής της Δ.Ε. Θα ήθελα να ευχαριστήσω την οικογένεια μου, και ειδικότερα τους γονείς μου, Σταύρο και Ιωάννα, για την στήριξη που μου δώσανε. Τέλος, ευχαριστώ πολύ όλους τους φίλους και συγγενείς που δέχτηκαν να συνεισφέρουν για να κάνω τις μετρήσεις του πειράματος.

## Περιεχόμενα

Πρόλογος.....	1
Περίληψη.....	1
Abstract.....	1
Ευχαριστίες.....	1
Περιεχόμενα.....	1
Λίστα Πινάκων.....	1
Λίστα Εικόνων.....	1
Κεφάλαιο 1 <sup>ο</sup> : Εισαγωγή.....	1
1.1 Εισαγωγή.....	1
1.2 Η δομή της εργασίας.....	1
Κεφαλαίο 2 <sup>ο</sup> : Καρδιά και καρδιακός παλμός.....	2
2.1 Ανασκόπηση της φυσιολογίας του καρδιακού παλμού.....	2
2.1.1. Δομή και λειτουργία του καρδιακού μυός.....	3
2.1.2 Παράγοντες που επηρεάζουν τον καρδιακό παλμό.....	4
2.1.3 Φυσιολογικοί παλμοί ανά ηλικία.....	6
2.1.4 Ηλεκτροκαρδιογράφημα.....	7
2.2 Η σημασία της μέτρησης του καρδιακού παλμού στον αθλητισμό.....	9
2.2.1 Τα οφέλη της γυμναστικής στον άνθρωπο.....	10
2.2.2 Παρακολούθηση της φυσικής κατάστασης.....	11
2.2.3 Βελτίωση της απόδοσης.....	12
2.2.4 Ανίχνευση προβλημάτων υγείας.....	12
Κεφάλαιο 3 <sup>ο</sup> : Υλικά.....	13
3.1 Η έννοια του ESP32 και ιστορική αναδρομή.....	13
3.1.1 Χαρακτηριστικά ,δυνατότητες, πλεονεκτήματα.....	13
3.1.2 Σύγκριση ESP32 και Arduino και επιλογή για την εργασία.....	14
3.1.3 Έκδοση ESP32 WROOM-32D.....	14
3.2 Αισθητήρας καρδιακού παλμού.....	16
3.2.1 Λειτουργία του αισθητήρα.....	17
Κεφάλαιο 4 <sup>ο</sup> : Δημιουργία και σχεδιασμός του μετρητή.....	18
4.1 Αρχιτεκτονική του συστήματος.....	18
4.2. Arduino IDE.....	19
4.2.1 Ιστορική αναδρομή.....	19
4.2.2 Βιβλιοθήκη για ESP32.....	20
4.3. Προγραμματισμός της μονάδας μέτρησης καρδιακού παλμού.....	21
Κεφάλαιο 5 <sup>ο</sup> : Υλοποίηση και πειράματα.....	25
5.1 Δημιουργία server.....	25
5.1.1 Σύνδεση με το Access Point.....	25
5.1.2 Front-end κώδικας.....	27
5.1.2.1 HTML-CSS.....	27
5.1.2.2 Javascript.....	31
5.1.3 Κώδικας My-Sql.....	35
5.1.3.1 Αποθήκευση στιγμιότυπου κυματομορφής.....	35
5.1.3.2 Κουμπί αποθήκευσης.....	36
5.2 Πειράματα και καταγραφή δεδομένων.....	40
5.2.1 Αποτελέσματα και πειράματα.....	41
5.2.2 Συμπεράσματα από τα αποτελέσματα.....	42
5.2.2.1 Προβλήματα και μειονεκτήματα.....	43

Κεφάλαιο 6° : Συμπεράσματα και Μελλοντικές Επεκτάσεις.....	43
6.1 Τελικά Συμπεράσματα.....	43
6.2 Μελλοντικές επεκτάσεις.....	44
Βιβλιογραφία.....	45
Παράρτημα.....	47

#### Λίστα Πινάκων

Πίνακας 1: Τα όρια του καρδιακού παλμού ανάλογα με την ηλικία

Πίνακας 2: Τα όρια του καρδιακού παλμού κατά την άσκηση, ανάλογα με την ηλικία

#### Λίστα Εικόνων

Εικόνα 1: Η δομή της καρδιάς του ανθρώπου

Εικόνα 2: Γραφική αναπαράσταση ενός ηλεκτροκαρδιογραφήματος

Εικόνα 3: Το μοντέλο ESP32-WROOM-32D

Εικόνα 4: Αισθητήρας Παλμών μοντέλου Pulse Sensor Amped

Εικόνα 5: Το κύκλωμα της Δ.Ε.

Εικόνα 6: Δημιουργία ενός νέου sketch στο Arduino IDE

Εικόνα 7: Η διαδικασία για την προσθήκη της βιβλιοθήκης ESP32

Εικόνα 8: Το διάγραμμα ροής για τον καρδιακό παλμό

Εικόνα 9: Το διάγραμμα ροής για το access point

Εικόνα 10: Το διάγραμμα ροής για την αποθήκευση δεδομένων

Εικόνα 11: Η ιστοσελίδα με την IP του ESP32

Εικόνα 12: Πείραμα για την πρώτη μέτρηση

Εικόνα 13: Πείραμα σε φίλο αθλητή

Εικόνα 14: Κυματομορφή δεύτερης προσπάθειας

Εικόνα 15: Πίνακας με τα αποθηκευμένα στοιχεία mysql

# Κεφάλαιο 1: Εισαγωγή

## 1.1 Εισαγωγή

Η συνεχιζόμενη και γρήγορη εξέλιξη της τεχνολογίας έχει επηρεάσει δραστικά πολλούς τομείς της καθημερινής μας ζωής, συμπεριλαμβανομένου και του χώρου της υγείας και του αθλητισμού. Η παρακολούθηση της φυσικής κατάστασης και η βελτίωση της απόδοσης έχουν γίνει ουσιαστικοί στόχοι για αθλητές κάθε επιπέδου, από τους ερασιτέχνες μέχρι τους επαγγελματίες.

Για το λόγο αυτό μία βασική παράμετρος παρακολούθησης είναι οι καρδιακοί παλμοί. Μέσω της παρακολούθησης αυτής φαίνεται η στιγμιαία κατάσταση της καρδιάς. Οι καρδιακοί παλμοί αποτελούν έναν σημαντικό δείκτη της υγείας του ανθρώπου.

Ειδικά στον αθλητισμό, το να γνωρίζει ο αθλητής και ο προπονητής τους καρδιακούς παλμούς, επιτρέπει την προσαρμογή της προπόνησης ανά πάσα στιγμή ανάλογα με τις ανάγκες και έτσι όχι μόνο φτάνει ο αθλητής στο μέγιστο των ικανοτήτων του, αλλά και προλαμβάνονται και πιθανά προβλήματα υγείας. Επιπλέον, η χρήση της τεχνολογίας των αισθητήρων και των ασύρματων δικτύων προσφέρει αξιόπιστες λύσεις για τη μέτρηση και την ανάλυση των καρδιακών παλμών.

Σε αυτή Δ.Ε θα γίνει ανάλυση της σχεδίασης ενός συστήματος μέτρησης των καρδιακών παλμών, πώς υλοποιείται και τελικά την αξιολόγησή του και την εφαρμογή του στην πράξη. Χρησιμοποιώντας την πλατφόρμα ESP32, θα υπάρξει μια αξιόπιστη καταγραφή των δεδομένων και την ασύρματη μεταφορά τους σε οθόνη για την παρακολούθησή τους.

## 1.2 Δομή της εργασίας

Αρχικά, στην παρούσα Δ.Ε. θα γίνει μια περιγραφή της λειτουργίας της καρδιάς και του καρδιακού παλμού, και θα διερευνηθούν τα ιδανικά όρια καρδιακού παλμού για αθλητές. Στην συνέχεια, θα υπάρξει εξέταση των υλικών που χρησιμοποιήθηκαν και θα διερευνηθούν οι ιδιαιτερότητες των μοντέλων που επιλέχθηκαν και η ανάλυση της κατασκευής τους κυκλώματος με τον αισθητήρα και το ESP-32. Επιπλέον, θα αναλυθεί ο κώδικας που δημιουργήθηκε στο Arduino ide και στο front-end και στο back-end, καθώς και η βάση sql, στην οποία θα αποθηκεύονται τα δεδομένα. Τέλος, θα μελετηθεί το πείραμα, θα βγούνε συμπεράσματα και πιθανές βελτιώσεις στο μέλλον.

## Κεφάλαιο 2<sup>ο</sup> : Καρδιά και καρδιακός παλμός

### 2.1 Ανασκόπηση της φυσιολογίας του καρδιακού παλμού

Η καρδιά είναι το βασικό όργανο του κυκλοφορικού συστήματος του ανθρώπου. Είναι ένα κοίλο, μυώδες όργανο, που δέχεται το αίμα που προέρχεται από τις φλέβες και το ωθεί στις αρτηρίες.

Σκοπός της καρδιακής λειτουργίας είναι η διατήρηση των ζωτικών λειτουργιών του οργανισμού, μέσω της μεταφοράς στα κύτταρα οξυγόνου και θρεπτικών ουσιών και την αποκομιδή άχρηστων ή βλαπτικών προϊόντων του μεταβολισμού και του διοξειδίου του άνθρακα, που παράγεται από τις καύσεις. [1]

Η καρδιά περιβάλλεται από ένα προστατευτικό περίβλημα που ονομάζεται περικάρδιο, το οποίο την προστατεύει από τραυματισμούς και την κρατά στη σωστή θέση μέσα στο στήθος. Τα τοιχώματα της καρδιάς αποτελούνται από τρία στρώματα: το ενδοκάρδιο (εσωτερικό στρώμα), το μυοκάρδιο (μεσαίο στρώμα) και το επικάρδιο (εξωτερικό στρώμα).

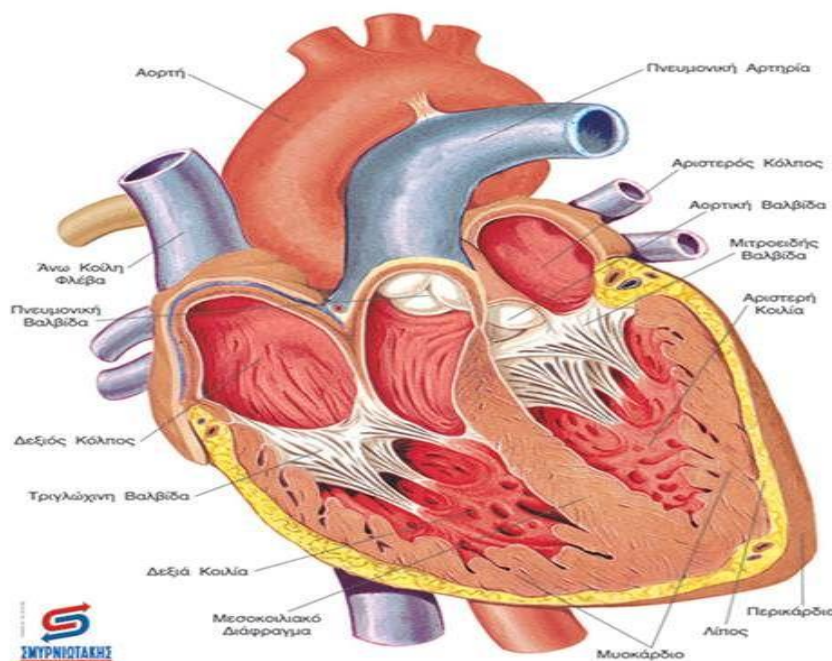
Η λειτουργία της καρδιάς ελέγχεται από έναν ενσωματωμένο ηλεκτρικό μηχανισμό, ο οποίος ονομάζεται φλεβοκόμβος. Ο φλεβοκόμβος παράγει ηλεκτρικά σήματα που προκαλούν τη σύσπαση της καρδιάς.

Καθώς το σήμα αυτό διαδίδεται μέσω συγκεκριμένων μονοπατιών, προκαλεί τη συγχρονισμένη συστολή των κόλπων και των κοιλιών, εξασφαλίζοντας την αποτελεσματική άντληση του αίματος. [2] [5]

### 2.1.1 Δομή και λειτουργία του καρδιακού μυός

Ξεκινώντας από τα βασικά, η καρδιά αποτελείται από δύο διαφορετικές αντλίες, την δεξιά και την αριστερή καρδιά. Η πρώτη στέλνει το αίμα από τους πνεύμονες στην καρδιά καθώς και το αντίστροφο, ενώ η δεύτερη στέλνει το αίμα από την καρδιά στους ιστούς και το αντίστροφο. Οι δύο αυτές λειτουργίες τις καρδιάς ονομάζονται πνευμονική και συστηματική αντίστοιχα.

## Η ΚΑΡΔΙΑ



Εικόνα 1: Η καρδιά του ανθρώπου

Πηγή: kardia--xlbook-smirniotakis

Ο φλεβόκομβος, που αναφέρθηκε παραπάνω, μεταδίδεται από τα περικολιακά κύτταρα ή τα μεταβατικά κύτταρα στον δεξιό κόλπο και στη συνέχεια στο υπόλοιπο ηλεκτρικό σύστημα αγωγής της καρδιάς, καταλήγοντας στη συστολή του μυοκαρδίου και τη διανομή του αίματος στο υπόλοιπο σώμα. [5]

Επίσης, η καρδιά συνεργάζεται, με άμεσο τρόπο, με το νευρικό σύστημα και το ενδοκρινικό. Το ενδοκρινικό σύστημα στέλνει ορμόνες, οι οποίες ειδοποιούν τα αιμοφόρα αγγεία να χαλαρώσουν ή να συμπιεστούν, κάτι το οποίο επηρεάζει την πίεση του αίματος. Ενώ το νευρικό σύστημα είναι αυτό που βοηθάει στον έλεγχο του παλμού, καθώς στέλνει συνεχώς σήματα όταν χρειάζεται να χτυπήσει χαμηλότερα, όπως σε κατάσταση χαλάρωσης, ή να χτυπήσει δυνατώτερα, όπως σε κατάσταση στρες. [1] [2]

Πιο συγκεκριμένα, ο καρδιακός παλμός ελέγχεται από δύο μέρη του νευρικού συστήματος, το συμπαθητικό(SNS) και το παρασυμπαθητικό(PNS). Το SNS είναι αυτό που ελευθερώνει τις ορμόνες επινεφρίνη και νορεπινεφρίνη. Κατά την διάρκεια ενεργοποίησης αυτής της λειτουργίας, παρατηρείται:

- 1) Αύξηση του οξυγόνου που μεταδίδεται στο σώμα και καλύτερη μεταφορά του στους πνεύμονες
- 2) Μεγέθυνση της κόρης του ματιού και βελτίωση της όρασης
- 3) Το συκώτι ενεργοποιεί τις αποθήκες ενέργειας του
- 4) Μείωση της ώρας χώνεψης φαγητού, ώστε η ενέργεια να μεταδοθεί αλλού [3]

Από την άλλη, το PNS, ελευθερώνει την ορμόνη ακετυλχολίνη, η οποία επιστρέφει τον οργανισμό σε μια κατάσταση χαλάρωσης, ισορροπώντας τις παρενέργειες του SNS. Συγκεκριμένα, κατά την διάρκεια ενεργοποίησης αυτής της λειτουργίας, παρατηρούνται:

- 1) Σμίκρυνση της κόρης του ματιού και πιο εύκολη παραγωγή περισσότερων δακρύων
- 2) Το στόμα παράγει περισσότερο σάλιο για να βοηθήσει στην κατάποση του φαγητού
- 3) Στενεύει το αναπνευστικό σύστημα και μειώνει την δουλειά των πνευμόνων
- 4) Ειδοποιεί το πάγκρεας να παράγει ινσουλίνη, διασπώντας την ζάχαρη που έχει ο οργανισμός [4] [6]

### **2.1.2 Παράγοντες που επηρεάζουν τον καρδιακό παλμό**

Ο καρδιακός παλμός είναι ένας από τους σημαντικότερους δείκτες της υγείας μας και επηρεάζεται από πολλούς παράγοντες, φυσικούς και ψυχολογικούς. Η καρδιά του ανθρώπου λειτουργεί ακατάπαυστα για να διοχετεύει αίμα σε όλο το σώμα, και η συχνότητα των παλμών της μπορεί να επηρεαστεί από ποικίλες συνθήκες.

Αρχικά, υπάρχουν οι φυσιολογικοί παράγοντες όπως η ηλικία και το φύλο. Όσον αφορά την ηλικία, είναι γνωστό ότι οι νεότεροι άνθρωποι τείνουν να έχουν υψηλότερο καρδιακό ρυθμό σε σύγκριση με τους μεγαλύτερους, καθώς η καρδιά των νέων είναι πιο ευαίσθητη στις μεταβολές των αναγκών του σώματος. Με την πάροδο του χρόνου, η καρδιά χάνει κάποια από την ελαστικότητά της, με αποτέλεσμα να αντλεί το αίμα πιο αργά, και συνεπώς ο καρδιακός ρυθμός μειώνεται. Και όσο αφορά το φύλο, ισχύει ότι οι γυναίκες τείνουν να έχουν ελαφρώς υψηλότερο καρδιακό ρυθμό σε σύγκριση με τους άνδρες, λόγω εν μέρει των διαφορών στη σωματική σύνθεση και την ορμονική λειτουργία. [7]

Δεύτερον, υπάρχουν οι ψυχολογικοί παράγοντες. Διάφορα συναισθήματα όπως ο θυμός, η χαρά ή ο φόβος επηρεάζουν την καρδιακή λειτουργία, αλλά κυρίως το άγχος και το στρες. Το

άγχος και οι συναισθηματικές καταστάσεις επηρεάζουν τον καρδιακό παλμό μέσω του αυτόνομου νευρικού συστήματος. Όταν βιώνουμε άγχος ή στρες, το σώμα απελευθερώνει αδρεναλίνη, η οποία αυξάνει τον καρδιακό ρυθμό. Αυτή η αντίδραση είναι γνωστή ως 'fight or flight response' και προετοιμάζει το σώμα για την αντιμετώπιση μιας απειλής. Είναι γεγονός πως η μακροχρόνια έκθεση σε άγχος μπορεί να οδηγήσει σε αυξημένο καρδιακό ρυθμό ακόμα και σε κατάσταση ηρεμίας, και αυτό μπορεί να είναι επικίνδυνο μακροπρόθεσμα για την καρδιαγγειακή υγεία. Η επίδραση του άγχους στην καρδιά μπορεί να οδηγήσει σε αυξημένο κίνδυνο για καρδιακά προβλήματα, όπως η υπέρταση και οι καρδιακές αρρυθμίες. [7]

Τρίτον, μπορεί οι παράγοντες να είναι περιβαλλοντικοί. Συγκεκριμένα η θερμοκρασία του περιβάλλοντος έχει σημαντική επίδραση στον καρδιακό παλμό καθώς σε υψηλές θερμοκρασίες, η καρδιά πρέπει να εργαστεί περισσότερο για να βοηθήσει το σώμα να διατηρήσει τη θερμοκρασία του μέσω της αυξημένης κυκλοφορίας του αίματος προς το δέρμα. Αυτό έχει ως αποτέλεσμα την αύξηση του καρδιακού ρυθμού. [7] Αντίθετα, σε χαμηλές θερμοκρασίες, ο καρδιακός ρυθμός μπορεί να μειωθεί καθώς το σώμα προσπαθεί να διατηρήσει τη θερμότητα του περιορίζοντας την κυκλοφορία του αίματος. Αλλά και επίσης, η υγρασία επηρεάζει τον καρδιακό ρυθμό. Σε συνθήκες υψηλής υγρασίας, το σώμα δυσκολεύεται να δροσιστεί μέσω της εφίδρωσης, και αυτό οδηγεί σε αυξημένο καρδιακό παλμό καθώς η καρδιά προσπαθεί να αντλήσει περισσότερο αίμα για να ρυθμίσει τη θερμοκρασία του σώματος. [8]

Τέλος, οι φυσικές δραστηριότητες παίζουν πολύ σημαντικό ρόλο στον καρδιακό παλμό, κάτι το οποίο θα αναλυθεί και παρακάτω. Κατά τη διάρκεια της άσκησης, ο καρδιακός ρυθμός αυξάνεται για να εξασφαλίσει την επαρκή παροχή οξυγόνου και θρεπτικών ουσιών στους μυς που εργάζονται. Η ένταση και η διάρκεια της άσκησης καθορίζουν το επίπεδο αύξησης του καρδιακού ρυθμού. Η καρδιά των αθλητών, ειδικά αυτών που κάνουν αερόβια άσκηση, συχνά λειτουργεί πιο αποτελεσματικά και έχει χαμηλότερο καρδιακό ρυθμό ηρεμίας. Είναι γεγονός ότι η σωματική άσκηση μακροπρόθεσμα μπορεί να βελτιώσει την καρδιακή λειτουργία, μειώνοντας τον καρδιακό παλμό σε κατάσταση ηρεμίας. Αυτή η μείωση του καρδιακού παλμού οφείλεται στο γεγονός ότι η καρδιά ενισχύεται και μπορεί να αντλήσει περισσότερο αίμα με κάθε παλμό, μειώνοντας έτσι την ανάγκη για αυξημένη συχνότητα παλμών. [7]

[8]

### 2.1.3 Φυσιολογικοί παλμοί ανά ηλικία

Όπως αναφέρθηκε παραπάνω, το φυσιολογικό όριο των παλμών της καρδιάς καθορίζεται με βάση παράγοντες όπως η φυσική κατάσταση, η ηλικία, τα επίπεδα του στρες και η ορμονική λειτουργία. Για τους περισσότερους ενήλικες, ιδανικοί θεωρούνται οι παλμοί μεταξύ 60 και 100 το λεπτό (bpm).

Το εύρος των παλμών άσκησης βοηθά στη διατήρηση της καρδιοαγγειακής υγείας, ενώ οι παλμοί σε χαλάρωση αποτελούν δείκτη φυσικής κατάστασης, με τιμές κοντά στα 60 bpm να υποδηλώνουν καλύτερη υγεία. Για να υπολογίσει κάποιος το ανώτατο φυσιολογικό όριο για τον εαυτό του, πρέπει να αφαιρέσει την ηλικία του από τον αριθμό 220. Κατά τη διάρκεια σωματικής άσκησης μέτριας έντασης, οι παλμοί πρέπει να κυμαίνονται στο 50% με 70% αυτού του αριθμού. Κατά τη διάρκεια πιο απαιτητικών δραστηριοτήτων, φυσιολογικοί θεωρούνται οι παλμοί που κυμαίνονται στο 70% με 85% του ανώτατου ορίου.

Για παράδειγμα, για ένα άτομο 40 ετών, το ανώτατο φυσιολογικό όριο των παλμών είναι 180. Κατά τη διάρκεια σωματικής άσκησης μέτριας έντασης οι παλμοί πρέπει να κυμαίνονται μεταξύ 90 και 126, ενώ κατά τη διάρκεια σωματικής άσκησης υψηλής έντασης μεταξύ 126 και 153. [9]

Ανάλογα με την ηλικία, οι ιδανικοί και οι μέγιστοι παλμοί αποτυπώνονται συνοπτικά στον παρακάτω πίνακα.

*Πίνακας 1 : Καρδιακοί παλμοί ανά ηλικία σε χαλάρωση, κατά τη δραστηριότητα, μέγιστοι*  
ΠΗΓΗ: <https://www.iatropedia.gr/ygeia/fysiologikoi-sfygmoi-oi-times-ana-ilikia-kai-pos-na-tous-metrisete/49901/>

ΗΛΙΚΙΑ (έτη)	ΦΥΣΙΟΛΟΓΙΚΟΙ ΠΑΛΜΟΙ ΣΕ ΧΑΛΑΡΩΣΗ(bpm)	ΦΥΣΙΟΛΟΓΙΚΟΙ ΠΑΛΜΟΙ 50-85%(bpm)	ΜΕΓΙΣΤΟΙ ΠΑΛΜΟΙ 100%(bpm)
20	60-100	100-170	200
30	60-100	95-162	190
40	60-100	93-157	185
45	60-100	90-153	180
50	60-100	88-149	175
55	60-100	85-145	170
60	60-100	83-140	165
65	60-100	80-136	160
70	60-100	78-132	155

Κατά τη διάρκεια της άσκησης, αλλά και της συνεχόμενης αθλητικής ενασχόλησης, συμβαίνουν αλλαγές στο σώμα των αθλητών. Μία από αυτές τις αλλαγές είναι ο χαμηλότερος καρδιακός ρυθμός σε κατάσταση ηρεμίας. Αυτό συμβαίνει γιατί η καρδιά είναι πιο αποτελεσματική, λόγω της συνεχόμενης εξάσκησης της και μπορεί να αντλήσει περισσότερο αίμα σε κάθε χτύπο.

Παρακάτω υπάρχει πίνακας που δείχνει τα εύρη των παλμών ανάλογα με την ηλικία για τους αθλητές.

Υπάρχουν μικρές διαφοροποιήσεις ανάλογα με το είδος της άσκησης και το επίπεδο της φυσικής κατάστασης του αθλητή. [9]

*Πίνακας 2: Εύρη καρδιακών παλμών ανάλογα με την ηλικία για αθλητές*

ΠΗΓΗ: <https://www.iatropedia.gr/ygeia/fysiologikoi-sfygmoi-oi-times-ana-ilikia-kai-pos-na-tous-metrisete/49901/>

ΗΛΙΚΙΑ(έτη)	ΚΑΡΔΙΑΚΟΙ ΠΑΛΜΟΙ (bpm)
6-10	70-100
11-14	60-100
15-19	50-90
20-29	40-80
30-39	40-80
40-49	40-80
50-59	40-80
60+	50-90

### 2.1.3 Ηλεκτροκαρδιογράφημα

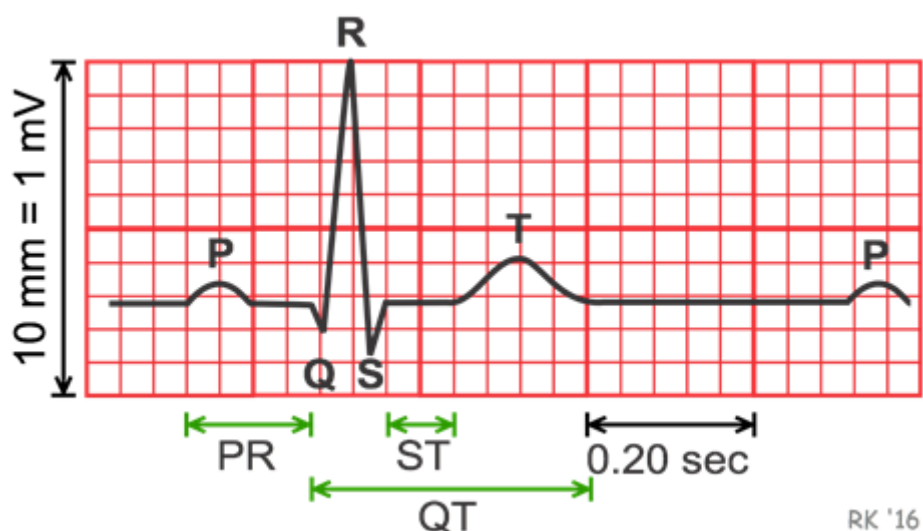
Το ηλεκτροκαρδιογράφημα (ΗΚΓ) είναι ένα από τα πιο σημαντικά διαγνωστικά εργαλεία στην καρδιολογία και στην γενική ιατρική. Το ΗΚΓ είναι μια μέθοδος καταγραφής της ηλεκτρικής δραστηριότητας της καρδιάς με τη χρήση ηλεκτροδίων που τοποθετούνται στο δέρμα του ασθενή. Με την καταγραφή αυτής της δραστηριότητας αποτυπώνεται η πορεία του ηλεκτρικού ρεύματος που διατρέχει την καρδιά, παρέχοντας σημαντικές πληροφορίες για τη λειτουργία της. Με αυτόν τον τρόπο, μπορεί να αποκαλύψει παθολογικές καταστάσεις, όπως αρρυθμίες, ισχαιμία, και διαταραχές στην αγωγιμότητα. [10] [11]

Η διαδικασία που χρησιμοποιείται για την εκτέλεση ενός ΗΚΓ ονομάζεται "ηλεκτροκαρδιογραφία", ενώ στα Αγγλικά είναι "Electrocardiogram(EKG)" [10]

Ακόμα, για να μελετηθεί το ΗΚΓ, πρέπει να είναι γνωστές η εκπόλωση και επαναπόλωση της καρδιάς. Η εκπόλωση αναφέρεται στην ταχεία μετακίνηση ιόντων νατρίου και ασβεστίου εντός των καρδιακών κυττάρων, προκαλώντας τη σύσπαση της καρδιάς. Η επαναπόλωση ακολουθεί, με την έξοδο ιόντων καλίου από τα κύτταρα, επιτρέποντας την επιστροφή της καρδιάς στην ηρεμία και την προετοιμασία για τον επόμενο καρδιακό παλμό. [12]

### 2.1.3.1 Γραφική αναπαράσταση και ανάλυση

Στη βασική του μορφή, το ΗΚΓ είναι μια γραφική αναπαράσταση της διαφοράς δυναμικού που καταγράφεται μεταξύ δύο σημείων στην επιφάνεια του σώματος με την πάροδο του χρόνου. Η γραφική αυτή απεικόνιση αποτελείται από κύματα, διαστήματα και τμήματα, τα οποία αντιστοιχούν σε συγκεκριμένες φάσεις του καρδιακού κύκλου. Οι βασικές συνιστώσες του ΗΚΓ είναι το κύμα P, το σύμπλεγμα QRS και το κύμα T. Το κύμα P αντιστοιχεί στην εκπόλωση των κόλπων, το σύμπλεγμα QRS αντιστοιχεί στην εκπόλωση των κοιλιών, και το κύμα T αντιστοιχεί στην επαναπόλωση των κοιλιών. Η λήψη ενός ΗΚΓ απαιτεί την τοποθέτηση ηλεκτροδίων σε συγκεκριμένα σημεία του σώματος, όπως στα άκρα και στο θώρακα. Αυτά τα ηλεκτρόδια συνδέονται με τον ηλεκτροκαρδιογράφο, ο οποίος καταγράφει τη διαφορά δυναμικού μεταξύ των σημείων αυτών. Οι καταγραφές αυτές παρουσιάζονται ως κυματομορφές σε μια οθόνη ή σε χαρτί. Η ανάλυση ενός ΗΚΓ γίνεται με την παρατήρηση και τη μέτρηση των διαφόρων κυμάτων και διαστημάτων. [13][14][12]



Εικόνα 2: Γραφική αναπαράσταση ενός ΗΚΓ  
ΠΗΓΗ: <https://www.caremedic.gr/cardiologos-kardiografima-ecg-ekg/>

Το πρώτο μισό του κύματος P αντιπροσωπεύει την εκπόλωση του δεξιού κόλπου και το δεύτερο μισό δείχνει την εκπόλωση του αριστερού κόλπου. Επιπλέον, πρέπει να έχει συγκεκριμένη μορφολογία και διάρκεια. Αν παρατηρηθούν ανωμαλίες, όπως μεγέθυνση ή ασυμμετρία του κύματος P, μπορεί να υποδηλώνουν παθολογικές καταστάσεις, όπως υπερτροφία των κόλπων. Το σύμπλεγμα QRS, το οποίο είναι πιο οξύ από το κύμα P, αντικατοπτρίζει τη διέγερση των κοιλιών. Η διάρκειά του, η μορφολογία του και οι μεταβολές σε διαφορετικές απαγωγές παρέχουν σημαντικές πληροφορίες για τη διάγνωση κοιλιακών διαταραχών. Το Q

αντιπροσωπεύει την εκπόλωση του μεσοκοιλιακού διαφράγματος, ενώ το S αντιπροσωπεύει την τελική αποπόλωση των ινών Purkinje. Το R, που είναι και το υψηλότερο κύμα, αντιπροσωπεύει το ηλεκτρικό ερέθισμα καθώς διέρχεται από τις κοιλίες κατά την εκπόλωση. [10]

Το κύμα T αντιπροσωπεύει την επαναπόλωση των κοιλιών. Οι μεταβολές στη μορφολογία του, όπως η ύψωση ή η αναστροφή του, μπορούν να υποδηλώνουν ισχαιμία ή άλλες μορφές καρδιακής δυσλειτουργίας. Η επαναπόλωση αυτή είναι ιδιαίτερα ευαίσθητη σε ηλεκτρολυτικές ανωμαλίες και μεταβολικές διαταραχές, κάνοντάς την σημαντικό δείκτη κλινικής σημασίας. Η εκτίμηση του διαστήματος PR, που είναι το διάστημα μεταξύ του έναρξης του κύματος P και του έναρξης του συμπλέγματος QRS, προσφέρει πληροφορίες για τη λειτουργία του κολποκοιλιακού κόμβου και της αγωγιμότητας του καρδιακού ερεθίσματος. Συνήθως είναι από 3 έως 5 μικρά τετράγωνα (120 έως 200 ms). [10]

Το παρατεταμένο διάστημα PR μπορεί να υποδηλώνει κολποκοιλιακό αποκλεισμό πρώτου βαθμού, ενώ βραχεία διάρκεια μπορεί να υποδηλώνει σύνδρομο Wolff-Parkinson-White, όπου υπάρχει παθολογική επικοινωνία μεταξύ των κόλπων και των κοιλιών.

[<https://www.ncbi.nlm.nih.gov/books/NBK549803>] Η καταγραφή του ηλεκτροκαρδιογραφήματος παρέχει επίσης τη δυνατότητα διάγνωσης οξείας ισχαιμίας του μυοκαρδίου, μέσω της παρατήρησης ανωμαλιών στο τμήμα ST και στην κυματομορφή T. Οι ανυψώσεις ή καταβυθίσεις του τμήματος ST είναι ισχυροί δείκτες ισχαιμίας ή εμφράγματος, ενώ η ανώμαλη κυματομορφή T μπορεί να υποδηλώνει ισχαιμικές αλλοιώσεις ή ηλεκτρολυτικές διαταραχές. [10]

### **2.1.3.2 Διάγνωση καρδιακών παθήσεων και αναταραχών**

Πέρα από την ανάλυση των βασικών κυμάτων και διαστημάτων, το ηλεκτροκαρδιογράφημα μπορεί να παράσχει πληροφορίες για τη συνολική λειτουργία της καρδιάς μέσω της αξιολόγησης της καρδιακής συχνότητας και του ρυθμού. Το κανονικό εύρος της καρδιακής συχνότητας είναι 60-100 παλμούς ανά λεπτό σε κατάσταση ηρεμίας. Αρρυθμίες, όπως η κολπική μαρμαρυγή ή η κοιλιακή ταχυκαρδία, μπορούν να διαγνωστούν μέσω της ανάλυσης των χαρακτηριστικών του ΗΚΓ. Η αναγνώριση αυτών των ανωμαλιών είναι κρίσιμη, καθώς οι αρρυθμίες μπορούν να προκαλέσουν σημαντικά αιμοδυναμικά προβλήματα ή ακόμα και να είναι θανατηφόρες αν δεν αντιμετωπιστούν έγκαιρα. Η εξέλιξη της τεχνολογίας επέτρεψε την ανάπτυξη ψηφιακών ηλεκτροκαρδιογράφων, που επιτρέπουν την καλύτερη ανάλυση και αποθήκευση των δεδομένων. Οι ψηφιακοί ηλεκτροκαρδιογράφοι προσφέρουν τη δυνατότητα επεξεργασίας και ανάλυσης των καταγραφών με τη χρήση ειδικών αλγορίθμων, οι οποίοι βοηθούν στην αυτόματη διάγνωση και τη μείωση της πιθανότητας λάθους. Ένα ακόμα σημαντικό σημείο που αξίζει να σημειωθεί είναι η δυνατότητα παρακολούθησης του ΗΚΓ σε

πραγματικό χρόνο, γεγονός που επιτρέπει την άμεση αξιολόγηση της κατάστασης του ασθενούς και τη λήψη άμεσων θεραπευτικών αποφάσεων. Αυτή η δυνατότητα είναι κρίσιμη σε περιπτώσεις επειγόντων περιστατικών, όπως σε ασθενείς με οξύ έμφραγμα του μυοκαρδίου, όπου η άμεση αντιμετώπιση μπορεί να σώσει ζωές.

Ένα από τα πιο εξελιγμένα χαρακτηριστικά του ΗΚΓ είναι η δυνατότητα της ανίχνευσης παθολογικών καταστάσεων που δεν σχετίζονται άμεσα με την καρδιά, όπως οι ηλεκτρολυτικές διαταραχές. Για παράδειγμα, η μεταβολή στα επίπεδα καλίου και ασβεστίου, μπορεί να επηρεάσει τη μορφολογία του κύματος T και αυτό να οδηγήσει σε επικίνδυνες αρρυθμίες. Το ΗΚΓ είναι έτσι ένα εργαλείο όχι μόνο για την καρδιολογία αλλά και για άλλους κλάδους της ιατρικής, όπως η εντατική θεραπεία και η νεφρολογία.[15] [11]

## **2.2 Η σημασία της μέτρησης του καρδιακού παλμού στον αθλητισμό**

Η μέτρηση του καρδιακού παλμού παίζει καθοριστικό ρόλο στον αθλητισμό, καθώς παρέχει πολύτιμες πληροφορίες για τη φυσική κατάσταση και την απόδοση του αθλητή. Η καταγραφή του καρδιακού παλμού επιτρέπει την παρακολούθηση αυτής της διαδικασίας και τη διασφάλιση ότι ο αθλητής προπονείται σε ασφαλή και αποδοτικά επίπεδα. Πιο συγκεκριμένα, η διατήρηση του καρδιακού παλμού σε μια συγκεκριμένη ζώνη, για παράδειγμα, όπως η ζώνη καύσης λίπους ή αντοχής, μπορεί να βοηθήσει τον αθλητή να πετύχει συγκεκριμένους στόχους, όπως την απώλεια βάρους ή την βελτίωση της αναπνευστικής ικανότητας. [16]

### **2.2.1 Τα οφέλη της γυμναστικής στον άνθρωπο**

Η τακτική σωματική άσκηση αποτελεί θεμέλιο για τη σωματική και ψυχική υγεία. Η συμμετοχή σε φυσικές δραστηριότητες ενισχύει το καρδιοαγγειακό σύστημα, βελτιώνει τη μυϊκή δύναμη και την ευλυγισία και συνεισφέρει στη διατήρηση ενός υγιούς βάρους. Συγκεκριμένα, η άσκηση βοηθά στη μείωση της αρτηριακής πίεσης, στον έλεγχο του σακχάρου στο αίμα, και στην ενίσχυση του ανοσοποιητικού συστήματος. Σε ψυχολογικό επίπεδο, η γυμναστική συνδέεται με υψηλότερα επίπεδα σεροτονίνης, γεγονός που συμβάλλει στη βελτίωση της διάθεσης και στη μείωση των συμπτωμάτων άγχους και κατάθλιψης [17] [18]

Η άσκηση επίσης ενισχύει τη γνωστική λειτουργία, ιδιαίτερα σε ηλικιωμένους, μέσω της βελτίωσης της κυκλοφορίας του αίματος στον εγκέφαλο, ενισχύοντας την ικανότητα μνήμης και συγκέντρωσης. Επιστημονικές μελέτες καταδεικνύουν ότι ακόμα και μικρές δόσεις άσκησης μπορούν να μειώσουν τις πιθανότητες εμφάνισης χρόνιων παθήσεων, όπως καρδιαγγειακών νοσημάτων, διαβήτη τύπου 2 και ορισμένων μορφών καρκίνου. [18]

Η γυμναστική έχει αναγνωριστεί ως ένας από τους πιο σημαντικούς παράγοντες για τη βελτίωση της υγείας και της ευεξίας. Πλήθος ερευνών έχει δείξει ότι η τακτική άσκηση βελτιώνει την καρδιακή λειτουργία, ενισχύει το ανοσοποιητικό σύστημα, μειώνει το άγχος και το στρες, προστατεύει το σώμα από την οστεοπόρωση και αυξάνει τις συνολικές επιδόσεις του οργανισμού. Επιπλέον, η γυμναστική συμβάλλει σημαντικά στη διαχείριση της χοληστερίνης, αυξάνοντας τα επίπεδα της καλής χοληστερίνης (HDL) και μειώνοντας την κακή χοληστερίνη (LDL) και τα τριγλυκερίδια, προάγοντας έτσι την καρδιαγγειακή υγεία. [19]

Η τακτική άσκηση έχει επίσης αποδειχθεί ότι μειώνει τα συμπτώματα της κατάθλιψης και ενισχύει την ψυχική υγεία, αυξάνοντας τις ενδορφίνες, τις ορμόνες της χαράς, που προκαλούν ευεξία και δίνουν ενέργεια στον οργανισμό. Παράλληλα, η γυμναστική συμβάλλει στην αύξηση της αυτοπεποίθησης και στον έλεγχο του βάρους, βοηθώντας τα άτομα να διατηρούν ένα υγιές σωματικό βάρος και να αισθάνονται καλύτερα με τον εαυτό τους. Επίσης, η γυμναστική προάγει την κοινωνικοποίηση, καθώς δίνει την ευκαιρία στα άτομα να αλληλεπιδρούν με άλλους, να δημιουργούν φιλίες και να συμμετέχουν σε κοινές δραστηριότητες. [17]

Η γυμναστική δεν είναι μόνο μέσο βελτίωσης της σωματικής και ψυχικής υγείας, αλλά και ένα ισχυρό εργαλείο για την πρόληψη και τη διαχείριση σοβαρών υγειονομικών προβλημάτων. Η τακτική άσκηση μπορεί να μειώσει τον κίνδυνο εγκεφαλικού επεισοδίου και να βοηθήσει στη διαχείριση της υψηλής αρτηριακής πίεσης, προσφέροντας έτσι προστασία έναντι καρδιαγγειακών παθήσεων. Επίσης, η γυμναστική παίζει σημαντικό ρόλο στην πρόληψη και διαχείριση του διαβήτη τύπου 2, βελτιώνοντας την ευαισθησία στην ινσουλίνη και βοηθώντας στον έλεγχο των επιπέδων σακχάρου στο αίμα. Για άτομα που πάσχουν από αρθρίτιδα, η άσκηση μπορεί να βελτιώσει την κινητικότητα και να μειώσει τον πόνο, ενώ για εκείνους που αντιμετωπίζουν προβλήματα ύπνου, η φυσική δραστηριότητα μπορεί να συμβάλει σε καλύτερο και πιο ποιοτικό ύπνο. Συνολικά, η γυμναστική αποτελεί έναν ανεκτίμητο σύμμαχο στην προαγωγή της υγείας, καθώς και στην ενίσχυση της ποιότητας ζωής. [17]

### **2.2.2 Παρακολούθηση της φυσικής κατάστασης**

Για την παρακολούθηση της φυσικής κατάστασης παλιότερα χρησιμοποιούνταν ένα απλό ημερολόγιο στο οποίο καταγραφόταν οι δραστηριότητες και οι χρόνοι του αθλητή, το πώς αισθανόταν και άλλοι παράγοντες που μπορούσαν να επηρεάσουν την απόδοσή του.

Αυτό το ημερολόγιο σιγά σιγά αντικαταστάθηκε από εφαρμογές στο κινητό τηλέφωνο που βοηθάνε στην παρακολούθηση των προπονήσεων και της προόδου.

Η παρακολούθηση της φυσικής κατάστασης μέσω της μέτρησης του καρδιακού παλμού βοηθά στην εξατομίκευση της άσκησης. Ο καρδιακός παλμός είναι ένας από τους βασικούς δείκτες φυσικής κατάστασης, καθώς παρέχει πληροφορίες για την ένταση της άσκησης και την απόκριση του σώματος σε αυτήν. Με την παρακολούθηση του καρδιακού παλμού, οι αθλητές μπορούν να διατηρούν την άσκησή τους σε ένα ασφαλές επίπεδο, αποφεύγοντας την υπερβολική καταπόνηση. [17]

Είναι επίσης σημαντική η κατανόηση της έντασης μέσω ζωνών καρδιακού παλμού (όπως η αερόβια και η αναερόβια ζώνη), οι οποίες επιτρέπουν την καλύτερη αξιοποίηση του χρόνου άσκησης για τη βελτίωση αντοχής και δύναμης.

### **2.2.3 Ανίχνευση προβλημάτων υγείας**

Οι αλλαγές στον καρδιακό παλμό μπορούν να χρησιμοποιηθούν ως ένδειξη πιθανών κινδύνων τραυματισμών αλλά και προβλημάτων υγείας. Υψηλοί παλμοί για μη αναμενόμενο διάστημα μπορεί να οδηγήσει σε κόπωση, τραυματισμούς και αυξημένο κίνδυνο καρδιαγγειακών συμβάντων. Χαμηλοί παλμοί μπορεί να υποδεικνύουν κόπωση ή υπερ-προπόνηση, που είναι παράγοντες που μπορούν να αυξήσουν τον κίνδυνο τραυματισμού. Επίσης οι ασυνήθιστες αλλαγές, όπως αρρυθμίες ή ξαφνική άνοδο ή πτώση του καρδιακού παλμού μπορεί να κρύβουν υποκείμενα καρδιαγγειακά προβλήματα. Επομένως, η συνεχής καταγραφή και ανάλυση του παλμού κατά τη διάρκεια της προπόνησης ή της ξεκούρασης επιτρέπει στους αθλητές να εντοπίζουν σημάδια υπερπροπόνησης, στρες ή υποκείμενων προβλημάτων που σχετίζονται με την καρδιακή λειτουργία. [20]

## Κεφάλαιο 3: Υλικά

### 3.1 Η έννοια του ESP32 και η ιστορική αναδρομή του

Το ESP32 είναι ένας μικροελεγκτής, εξαιρετικά ισχυρός, ευέλικτος και αποδοτικός, που αναπτύχθηκε από την εταιρεία Espressif Systems, μια κινέζικη εταιρεία με έδρα την Σανγκάη. Πρόκειται για μια πλατφόρμα που συνδυάζει δυνατότητες επεξεργασίας, δικτύωσης Wi-Fi και Bluetooth, καθιστώντας την ιδανική για εφαρμογές στο Διαδίκτυο των Πραγμάτων (IoT), αυτοματισμούς και άλλες εφαρμογές που απαιτούν συνδεσιμότητα και υπολογιστική ισχύ σε μικρό μέγεθος. Ο προκάτοχός του, το ESP8266, κατέκτησε την αγορά χάρη στη χαμηλή τιμή και τη δυνατότητα σύνδεσης Wi-Fi, ωστόσο, οι περιορισμένες δυνατότητες επεξεργασίας και συνδεσιμότητας του άφησαν περιθώριο για περαιτέρω ανάπτυξη. Το ESP32 εισήλθε στην αγορά το 2016 και κατάφερε να συνδυάσει αυξημένη ισχύ, χαμηλή κατανάλωση ενέργειας και προηγμένες δυνατότητες ασύρματης επικοινωνίας. Η πλατφόρμα έγινε ιδιαίτερα δημοφιλής λόγω της υποστήριξης από ανοιχτό κώδικα εργαλεία και τη δυνατότητα εύκολης ενσωμάτωσης σε πολλά έργα. [21]

Η ιστορική πορεία του ESP32 αποτελεί παράδειγμα της συνεχούς εξέλιξης της τεχνολογίας μικροελεγκτών. Η Espressif Systems, μέσω του ESP32, κατάφερε να προσφέρει μία ολοκληρωμένη λύση για κατασκευαστές και προγραμματιστές, επιτρέποντάς τους να δημιουργήσουν από απλές IoT συσκευές έως σύνθετα έξυπνα συστήματα. Η δυνατότητα συνδυασμού Wi-Fi και Bluetooth ήταν πρωτοποριακή τη στιγμή της κυκλοφορίας του και αποτέλεσε καθοριστικό σημείο για τη διεύρυνση του ESP32 στην αγορά IoT.[21]

#### 3.1.1 Χαρακτηριστικά, δυνατότητες και πλεονεκτήματα

Το ESP32 χαρακτηρίζεται από ένα σύνολο προηγμένων τεχνολογικών χαρακτηριστικών που το καθιστούν μοναδικό στον χώρο των μικροελεγκτών. Διαθέτει διπύρηνο επεξεργαστή Tensilica Xtensa LX6, ο οποίος παρέχει υψηλή απόδοση και μπορεί να φτάσει συχνότητα λειτουργίας έως και 240 MHz. Επιπλέον, ενσωματώνει μνήμη RAM 520 KB και δυνατότητα επέκτασης μέσω εξωτερικής μνήμης, προσφέροντας αρκετή χωρητικότητα για σύνθετες εφαρμογές. Η υποστήριξη τόσο Wi-Fi όσο και Bluetooth (Classic και BLE) εξασφαλίζει ευελιξία στη συνδεσιμότητα, ενώ οι πολλαπλές ψηφιακές και αναλογικές εισοδοί/έξοδοι, καθώς και πρωτόκολλα όπως SPI, I2C και UART, το καθιστούν εξαιρετικά ευπροσάρμοστο. [22]

Τα πλεονεκτήματα του ESP32 είναι ιδιαίτερα εμφανή σε εφαρμογές IoT. Η χαμηλή κατανάλωση ενέργειας μέσω των ενεργειακά αποδοτικών λειτουργιών του, όπως το "deep sleep mode," επιτρέπει τη χρήση του σε εφαρμογές με τροφοδοσία από μπαταρία. Παράλληλα, η υποστήριξη για ασφαλείς επικοινωνίες μέσω πρωτοκόλλων όπως TLS και τα ενσωματωμένα χαρακτηριστικά ασφαλείας, όπως το hardware encryption, προσφέρουν υψηλό επίπεδο προστασίας. Όλα αυτά τα χαρακτηριστικά καθιστούν το ESP32 κατάλληλο για ένα ευρύ φάσμα εφαρμογών, από έξυπνα σπίτια και φορητές συσκευές μέχρι βιομηχανικούς αυτοματισμούς. [23] [22]

### 3.1.2 Σύγκριση ESP32 και Arduino και επιλογή για την εργασία

Αρχικά, αξίζει να αναφερθεί ότι και οι δύο πλατφόρμες, ESP32 και Arduino, είναι αρκετά αξιόπιστες, με καλή ποιότητα κατασκευής και σταθερό λογισμικό. Συγκρίνοντας το ESP32 και το Arduino μπορεί κανείς να εντοπίσει σημαντικές διαφορές, ιδιαίτερα σε επίπεδο επιδόσεων και δυνατοτήτων. Το Arduino είναι γνωστό για την απλότητα και τη φιλικότητα του προς τον χρήστη, καθιστώντας το ιδανικό για αρχάριους. Ωστόσο, η πλατφόρμα αυτή, ιδιαίτερα στα βασικά της μοντέλα όπως το Arduino Uno, υπολείπεται σε επεξεργαστική ισχύ, δυνατότητες συνδεσιμότητας και υποστήριξη περιφερειακών συγκριτικά με το ESP32. Το Arduino διαθέτει επεξεργαστή AVR με συχνότητα που συνήθως δεν ξεπερνά τα 16 MHz, ενώ οι περιορισμένες μνήμες RAM και Flash καθιστούν δύσκολη την ανάπτυξη πολύπλοκων εφαρμογών. [24]

Αντίθετα, το ESP32 προσφέρει ενσωματωμένο Wi-Fi και Bluetooth, χαρακτηριστικά που απουσιάζουν από τα περισσότερα Arduino χωρίς την προσθήκη εξωτερικών module. Επιπλέον, η μεγαλύτερη ταχύτητα επεξεργασίας, η μεγαλύτερη μνήμη και η δυνατότητα χειρισμού πολλαπλών νημάτων καθιστούν το ESP32 ιδανικό για απαιτητικά έργα. Επίσης, ένα κοινό χαρακτηριστικό είναι ότι και οι δύο πλατφόρμες μπορούν να προγραμματιστούν μέσω του Arduino IDE, με το ESP32 να απαιτεί πρόσθετες βιβλιοθήκες, ανάλογα με τις απαιτήσεις του έργου. [24]

Ο λόγος που το ESP32 επιλέχθηκε για την εργασία, είναι επειδή προσφέρει μια πιο ολοκληρωμένη λύση, ιδανική για εφαρμογές IoT που απαιτούν ασύρματη συνδεσιμότητα, ευελιξία και ισχύ επεξεργασίας. Επίσης, η υποστήριξη από την κοινότητα και η συμβατότητα με το οικοσύστημα του Arduino IDE διευκολύνουν την ανάπτυξη λογισμικού, καθιστώντας την επιλογή του ESP32 όχι μόνο πρακτική αλλά και οικονομικά συμφέρουσα. [29]

### 3.1.3 Έκδοση ESP32 WROOM-32D

Η έκδοση ESP32 WROOM-32D αποτελεί μία από τις πιο διαδεδομένες παραλλαγές του ESP32, σχεδιασμένη για χρήση σε βιομηχανικές και καταναλωτικές εφαρμογές. Το WROOM-32D περιλαμβάνει ένα ολοκληρωμένο σύστημα με επεξεργαστή ESP32, μνήμη Flash 4 MB και κεραία υψηλής απόδοσης. Ένα από τα κύρια χαρακτηριστικά της είναι η μικρή φυσική της διάσταση, που την καθιστά ιδανική για έργα με περιορισμούς χώρου, όπως φορητές συσκευές και συστήματα IoT.

Η έκδοση αυτή προσφέρει σταθερή απόδοση, ακόμα και σε περιβάλλοντα με παρεμβολές Wi-Fi ή Bluetooth, χάρη στην ενσωματωμένη δυνατότητα αυτόματης ρύθμισης κεραίας. Επίσης, η δυνατότητα λειτουργίας σε θερμοκρασίες μεταξύ  $-40^{\circ}\text{C}$  και  $85^{\circ}\text{C}$  καθιστά το WROOM-32D κατάλληλο για βιομηχανικές εφαρμογές. Η υποστήριξη για πολλαπλά επίπεδα κατανάλωσης ενέργειας επιτρέπει στους προγραμματιστές να ρυθμίσουν την κατανάλωση ανάλογα με τις ανάγκες του έργου τους. [25]



Εικόνα 3: TO ESP32 WROOM 32D

ΠΗΓΗ: <https://linuxhint.com/esp32-cam-same-as-esp32/>

Βασικές προδιαγραφές:

- Τάση Τροφοδοσίας: 3.3V
- Πρότυπα Wi-Fi: 802.11 b/g/n
- Bluetooth: v4.2 BR/EDR και BLE
- Επεξεργαστής: Διπλός πυρήνας 32-bit LX6 μικροεπεξεργαστής
- Ταχύτητα Ρολογιού: Έως 240 MHz
- Μνήμη: 448 KB ROM, 520 KB SRAM

- Αποθηκευτικός Χώρος Flash: 4 MB έως 16 MB
- Θερμοκρασία Λειτουργίας: -40°C έως 85°C
- Διαστάσεις: 35.6 mm x 34.4 mm x 3.5 mm

[25]

Το ESP32 WROOM-32D προσφέρει έναν συνδυασμό υψηλής απόδοσης, χαμηλής κατανάλωσης ενέργειας και ευκολίας στην ανάπτυξη. Η ενσωμάτωσή του στην παρούσα εργασία βασίζεται στην αξιοπιστία του και στη δυνατότητα να υποστηρίξει πλήρως τις απαιτήσεις της εφαρμογής. Η συγκεκριμένη έκδοση υποστηρίζεται από εκτενή τεκμηρίωση και παραδείγματα από την κοινότητα, καθιστώντας την ιδανική επιλογή για γρήγορη υλοποίηση και ανάπτυξη προηγμένων συστημάτων IoT.

### 3.2 Αισθητήρας καρδιακού παλμού

Ο αισθητήρας καρδιακού ρυθμού για Arduino είναι ένα προηγμένο και αξιόπιστο εργαλείο σχεδιασμένο για την παρακολούθηση του καρδιακού ρυθμού και την ανίχνευση των παλμών. Καθώς αναπτύχθηκε για εύκολη ενσωμάτωση με την πλατφόρμα Arduino, αυτός ο αισθητήρας είναι μια εξαιρετική επιλογή για την παρακολούθηση της υγείας και της φυσικής κατάστασης. Προσφέρει μια αποδοτική λύση για την καταγραφή και ανάλυση δεδομένων καρδιακού ρυθμού σε πραγματικό χρόνο.

Βασικά χαρακτηριστικά:

- Υψηλή Ευαισθησία: Ο αισθητήρας παρέχει ακριβείς μετρήσεις παλμών, καθιστώντας τον κατάλληλο για εφαρμογές που απαιτούν ακριβή παρακολούθηση καρδιακού ρυθμού.
- Εύκολη Ενσωμάτωση: Σχεδιασμένος για να λειτουργεί άψογα με τις πλακέτες Arduino, ο αισθητήρας έρχεται με σαφείς οδηγίες και παραδείγματα κωδικών, απλοποιώντας τη διαδικασία ενσωμάτωσης.
- Συμπαγής Σχεδίαση: Το μικρό μέγεθος και το ελαφρύ του σχεδιασμό τον καθιστούν ιδανικό για φορητές εφαρμογές και έργα όπου ο χώρος είναι περιορισμένος.
- Χαμηλή Κατανάλωση Ενέργειας: Ο αισθητήρας λειτουργεί αποδοτικά, εξασφαλίζοντας ελάχιστη κατανάλωση ενέργειας, κάτι που είναι κρίσιμο για συσκευές που λειτουργούν με μπαταρία.

- Δεδομένα σε Πραγματικό Χρόνο: Προσφέρει δεδομένα παλμών σε πραγματικό χρόνο, τα οποία μπορούν εύκολα να υποβληθούν σε επεξεργασία και να απεικονιστούν χρησιμοποιώντας τον σειριακό plotter του Arduino ή εξωτερικές οθόνες.

Ο συγκεκριμένος αισθητήρας έχει τις εξής προδιαγραφές:

- Όνομα: Pulse Sensor Amped
- Τάση Λειτουργίας: 3V ή 5V
- Κατανάλωση Ρεύματος: 4mA στα 5V
- Μήκος Καλωδίου: 16mm με αρσενικές κεφαλίδες

[26]



Εικόνα 4: Pulse Sensor Amped

ΠΗΓΗ: Hellas Digital

### 3.2.1 Λειτουργία του αισθητήρα καρδιακού παλμού

Για να υπάρξει κατανόηση του πως λειτουργεί ο αισθητήρας, πρέπει να γίνει ανάλυση της τεχνολογίας της φωτοπληθυσμογραφίας (PPG), μια μη επεμβατικής μεθόδου που ανιχνεύει τις μεταβολές στον όγκο του αίματος μέσω της ανάλυσης του φωτός. Το κύκλωμα του αισθητήρα αποτελείται από έναν αισθητήρα και έναν φωτοανιχνευτή. Το LED εκπέμπει φως προς τον ιστό, όπως το δάχτυλο ή τον καρπό, και καθώς το αίμα απορροφά μέρος αυτού του φωτός, οι μεταβολές στη ροή του αίματος κατά τη διάρκεια των καρδιακών παλμών προκαλούν αλλαγές στην ποσότητα του φωτός που ανακλάται ή διαπερνά τον ιστό. Παράλληλα, ο φωτοανιχνευτής καταγράφει αυτές τις μεταβολές στην ένταση του φωτός, δημιουργώντας ένα σήμα που αντιπροσωπεύει τον καρδιακό παλμό. Το σήμα αυτό στη συνέχεια υποβάλλεται σε επεξεργασία μέσω ενισχυτών και φίλτρων για την απομάκρυνση θορύβου και την ενίσχυση της ακρίβειας.

Η ανάλυση του επεξεργασμένου σήματος επιτρέπει την εξαγωγή πληροφοριών όπως ο ρυθμός και η μεταβλητότητα του καρδιακού παλμού. [27] [28]

Το 1937, ο Hertzman ανακάλυψε ότι η ποσότητα του φωτός που ανιχνεύεται μέσω σκέδασης αφού φωτίσει το δέρμα, αλλάζει σημαντικά ανάλογα με τη δραστηριότητα της καρδιάς. Στην συνέχεια, μέσω του νόμου Beer–Lambert (1851), ορίζεται “η αποδυνάμωση της έντασης του φωτός ανάλογα με τον συντελεστή απόσβεσης, τη συγκέντρωση και το μήκος της οπτικής διαδρομής του φωτός μέσα από ένα μέσο”. [28]

Ο νόμος Beer–Lambert εκφράζεται ως εξής:

$$I=I_0e^{-\epsilon I c} \quad \text{ΠΗΓΗ: [28]}$$

όπου το  $I$  είναι η ένταση του φωτός που διαπερνά το μέσο, το  $I_0$  είναι η αρχική ένταση του φωτός, το  $\epsilon$  είναι ο συντελεστής απορρόφησης, το  $I$  είναι το μήκος της οπτικής διαδρομής και το  $c$  είναι η συγκέντρωση του μέσου. Το εκθετικό μέρος του νόμου ορίζεται ως απορρόφηση ( $A$ ), με τη σχέση:

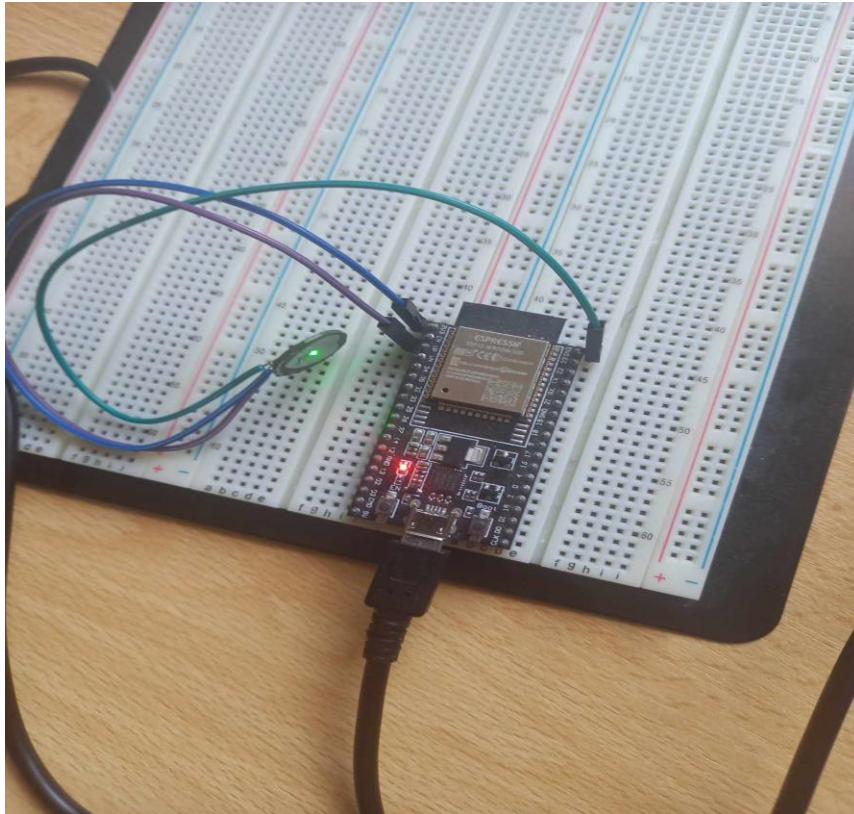
$$A= - \epsilon * I * c$$

Έτσι εξηγείται η δυνατότητα της PPG να καταγράφει την ποσότητα του φωτός που διαπερνά ή αντανακλάται από τις αλλαγές στη συγκέντρωση ουσιών στο αίμα και στην οπτική διαδρομή του φωτός λόγω των παλμών [28]

## Κεφάλαιο 4: Δημιουργία και σχεδιασμός του μετρητή

### 4.1 Η αρχιτεκτονική του συστήματος

Χρησιμοποιώντας τα δύο υλικά, το ESP32 και τον αισθητήρα, θα μελετηθεί η αρχιτεκτονική του συστήματος. Αρχικά, το ESP32 τοποθετείται στο ράστερ, διασφαλίζοντας την πρόσβαση στα ακροδέκτες του. Στην συνέχεια, θα συνδεθεί ο αισθητήρας, ο οποίος διαθέτει τρία καλώδια, το πράσινο, το μπλε και μωβ. Το πράσινο καλώδιο συνδέεται σε έναν ακροδέκτη ‘GND’ του ESP32, εξασφαλίζοντας κοινή γείωση μεταξύ των συσκευών. Το μπλε καλώδιο συνδέεται στον ακροδέκτη ‘3.3V’ του ESP32, παρέχοντας την απαιτούμενη τάση λειτουργίας. Το μωβ καλώδιο, το οποίο μεταφέρει το σήμα, συνδέεται σε έναν αναλογικό ακροδέκτη, συγκεκριμένα τον ακροδέκτη ‘VP’ για την ανάγνωση των δεδομένων από τον αισθητήρα. Με τον αισθητήρα συνδεδεμένο στο ESP32 σωστά, μπορούν να ξεκινήσουν οι μετρήσεις. [30]



Εικόνα 5: Το κύκλωμα

## 4.2 Arduino IDE

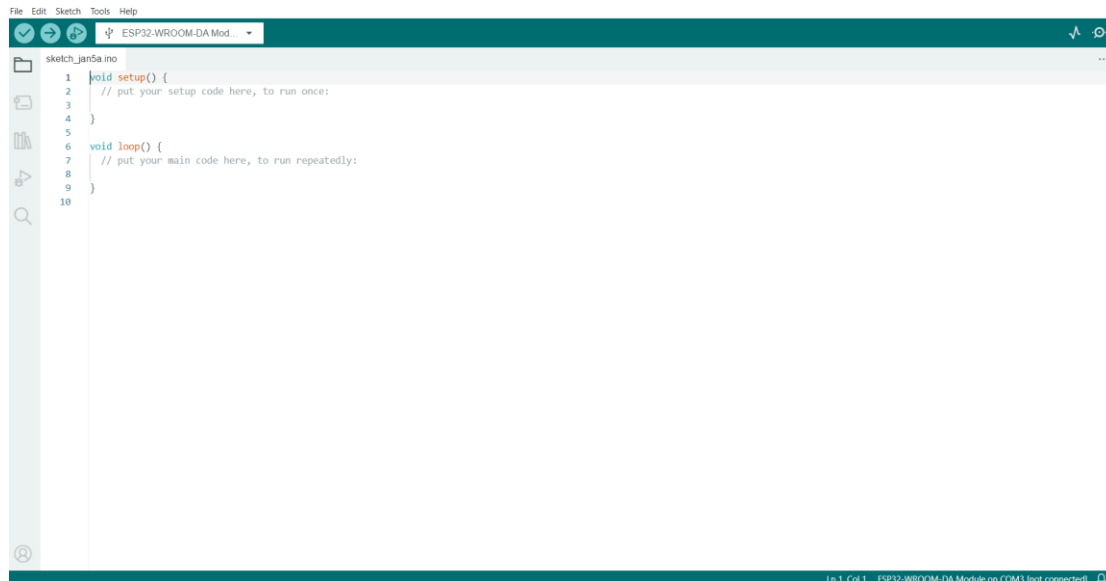
Για την συγκεκριμένη Δ.Ε, χρησιμοποιήθηκε το περιβάλλον ARDUINO IDE. Ένα περιβάλλον ανάπτυξης λογισμικού που χρησιμοποιείται για τον προγραμματισμό των πλακετών Arduino. Ξεκίνησε το 2005 από τους Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, και David Mellis στο πλαίσιο ενός πρότζεκτ στο πανεπιστήμιο της Ιταλίας, της Interaction Design Institute Ivrea. [31]

### 4.2.1 Χαρακτηριστικά

Η γλώσσα προγραμματισμού που χρησιμοποιείται στο Arduino IDE είναι η γλώσσα C/C++. Αν και η Arduino IDE παρέχει μια απλοποιημένη διεπαφή και δομές κώδικα για ευκολία χρήσης, οι βασικές αρχές και συντακτικές δομές της γλώσσας παραμένουν τα ίδια με αυτές της C/C++.

Μερικά από τα βασικά χαρακτηριστικά της γλώσσας προγραμματισμού που χρησιμοποιείται στην Arduino IDE περιλαμβάνουν τη δήλωση μεταβλητών, τις συναρτήσεις, τις δομές ελέγχου ροής όπως οι if-else, οι βρόχοι for και while, καθώς και δομές δεδομένων όπως πίνακες και δομές. Επίσης, η Arduino IDE παρέχει βιβλιοθήκες που διευκολύνουν την επικοινωνία με

αισθητήρες, κινητήρες, οθόνες και άλλες ηλεκτρονικές συσκευές, καθιστώντας τον προγραμματισμό τους απλούστερο. [31]



Εικόνα 6: Ένα νέο sketch στο Arduino IDE

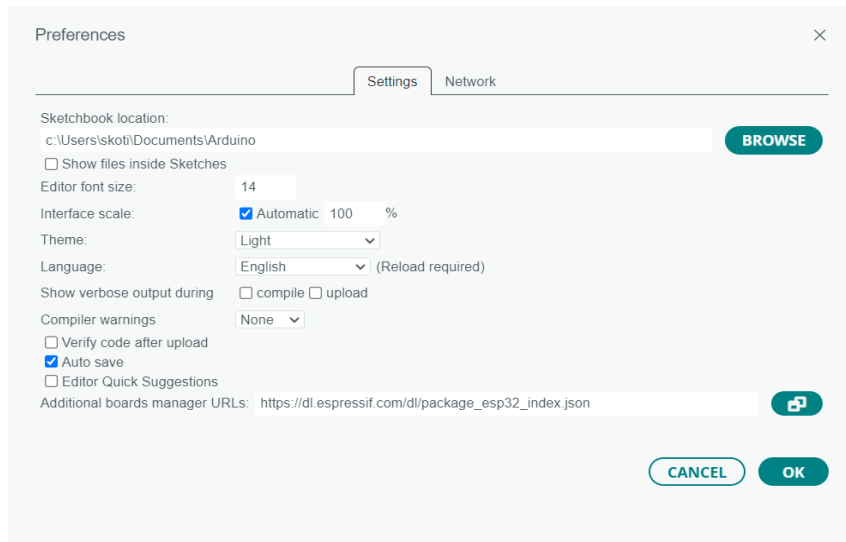
ΠΗΓΗ: ARDUINO IDE

#### 4.2.2 Βιβλιοθήκη ESP32

Η βιβλιοθήκη ESP32 παρέχει όλα τα απαραίτητα εργαλεία και παραδείγματα κώδικα για τη δημιουργία σύνθετων IoT εφαρμογών, αξιοποιώντας τις ασύρματες δυνατότητες του ESP32. Αρχικά, περιέχει όλες τις πρόσθετες δυνατότητες του esp32, όπως την σύνδεση στο WIFI και στο LOW-ENERGY-BLUETOOTH, την ανάγνωση αισθητήρων και την δημιουργία webserver, αλλά και την σύνδεση σε access point. Όλες αυτές οι δυνατότητες θα χρησιμοποιηθούν για την Δ.Ε.

Η διαδικασία για την εγκατάσταση της βιβλιοθήκης είναι πολύ απλή. Ακολουθώντας την διαδρομή File>Preferences, ανοίγει το παράθυρο Settings. Εκεί, στο πεδίο “Additional Board Manager URLs” προστίθεται η διεύθυνση:

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)



Εικόνα 7: Η διαδικασία για την προσθήκη της βιβλιοθήκης ESP32

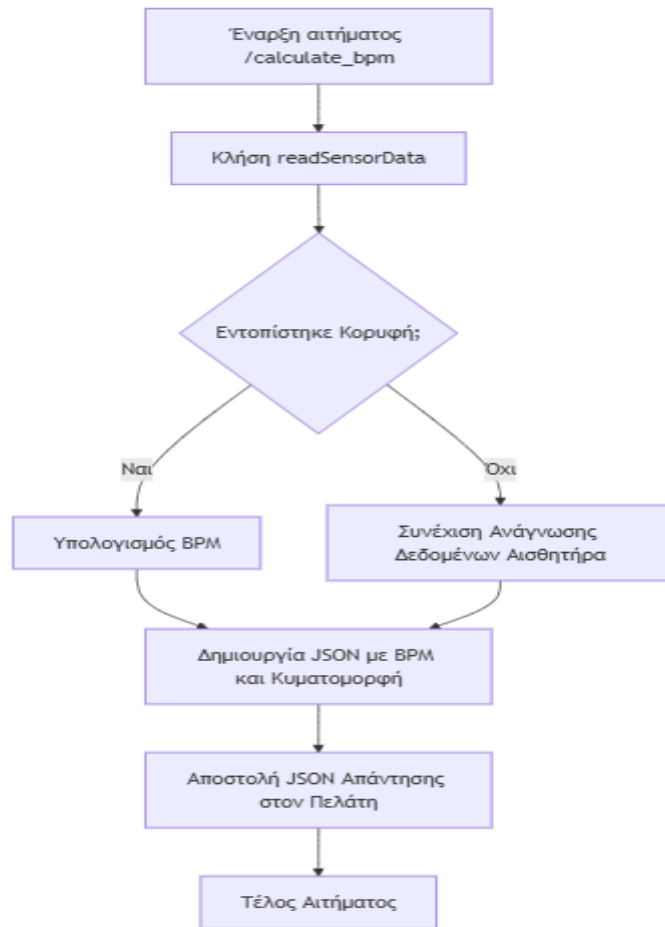
ΠΗΓΗ: ARDUINO IDE

Μετά από αυτή την διαδικασία, πρέπει να γίνει ακόμα ένα βήμα. Στο πεδίο Tools, επιλέγεται το Board και από εκεί το Board Manager (Tools > Board > Board Manager). Στην συνέχεια γίνεται αναζήτηση για την βιβλιοθήκη “ESP32 by Espressif Systems” και πραγματοποιείται η εγκατάσταση της με το κουμπί install.

Επομένως, πλέον η εγκατάσταση έχει γίνει αποτελεσματικά και μπορεί να ξεκινήσει το κομμάτι του κώδικα, και πιο συγκεκριμένα, η μέτρηση των καρδιακών παλμών.

### 4.3 Προγραμματισμός της μονάδας μέτρησης καρδιακού παλμού

Για να λειτουργήσει ο μετρητής όπως θέλουμε, πρέπει να υπολογιστεί ο καρδιακός παλμός ανά λεπτό (BPM), μέσω της τιμής που λαμβάνεται από τον αισθητήρα. Παρακάτω παρατίθεται το διάγραμμα ροής.



Εικόνα 8: Το διάγραμμα ροής για το bpm

Ο κώδικας για τον συγκεκριμένο υπολογισμό είναι ο εξής:

```

unsigned long lastBeatTime = 0;

unsigned long currentBeatTime = 0;

int Signal;

int threshold = 550;

bool inPeak = false;

int BPM = 0;

String waveform = "";
  
```

```

void loop() {
  server.handleClient();

  Signal = analogRead(PulseSensor_PIN);

  if (Signal > threshold && !inPeak) {

    inPeak = true;

    currentBeatTime = millis();

    if (lastBeatTime > 0) {

      unsigned long beatInterval = currentBeatTime - lastBeatTime;

      BPM = 60000 / beatInterval;

    }

    lastBeatTime = currentBeatTime;

  }

  if (Signal < threshold && inPeak) {

    inPeak = false;

  }

  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {

    previousMillis = currentMillis;

    Serial.print("Current BPM: ");

    Serial.println(BPM);

  }

  delay(20);

}

```

Αρχικά, στην πρώτη γραμμή ορίζεται μια συμβολική σταθερά με την τιμή 36, η οποία αντιστοιχεί στον αριθμό του pin στον οποίο είναι συνδεδεμένος ο αισθητήρας Pulse. Στις επόμενες δύο γραμμές δημιουργούνται μεταβλητές για την αποθήκευση του χρόνου του τελευταίου παλμού και του τρέχοντος παλμού.

Πρώτα δηλώνεται η μεταβλητή 'int Signal' για την αποθήκευση της τιμής του σήματος που διαβάζεται από τον αισθητήρα. Στην συνέχεια θέτουμε μια τιμή κατώφλιου για την ανίχνευση του παλμού. Εντολή 'inPeak' δηλώνεται ώστε να κρατάει την κατάσταση του παλμού σε σημείο κορυφής, όταν ανιχνευθεί ο παλμός, ενώ η εντολή με το όνομα 'BPM' και την αρχική τιμή 0, χρησιμοποιείται για να αποθηκεύσει τους καρδιακούς παλμούς ανά λεπτό που υπολογίζονται από τον κώδικα.

Στην συνέχεια, η setup() συνάρτηση αρχικοποιεί την επικοινωνία σειριακού καναλιού με την ταχύτητα 115200 bps και ορίζει την ανάλυση ανάγνωσης του ADC σε 10 bit.

Η loop() συνάρτηση εκτελείται επανειλημμένα. Αναλαμβάνει να διαβάσει τιμές από τον αισθητήρα, να ανιχνεύει παλμούς, να υπολογίζει τους καρδιακούς παλμούς ανά λεπτό (BPM) και να εκτυπώνει τις τιμές στην σειριακή θύρα, καθώς η εντολή analogRead, διαβάζει την τιμή από τον Pulse Sensor και την αποθηκεύει στη μεταβλητή Signal

Στο επόμενο κομμάτι του κώδικα, ελέγχεται άμα η τιμή έχει ξεπεράσει το κατώφλι και υπολογίζεται το bpm με τον τύπο:

$$BPM = \frac{60000}{beatInterval}$$

Ο τύπος αυτός υπολογίζει τους καρδιακούς παλμούς ανά λεπτό χρησιμοποιώντας το διάστημα μεταξύ δύο συνεχόμενων παλμών, το οποίο ορίστηκε ως "beatInterval", ενώ το 60000 αναπαριστά τον αριθμό των χιλιοστών του λεπτού, καθώς υπάρχουν 60 χιλιοστά του λεπτού σε ένα λεπτό. Με την διαίρεση αυτή, μετατρέπουμε τα χιλιοστά του λεπτού σε λεπτά. [32]

```
if (Signal < threshold && inPeak) {  
  inPeak = false;  
}
```

Αυτή η εντολή ελέγχει αν η τιμή του σήματος έχει ξεπεράσει το κατώφλι και αν δεν είναι ήδη σε κατάσταση peak, υπολογίζει το BPM. Και τέλος, η εντολή delay, προκαλεί μια μικρή καθυστέρηση 20 milliseconds για σταθεροποίηση της εκτέλεσης του κώδικα.

## ΚΕΦΑΛΑΙΟ 5<sup>ο</sup> : ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΠΕΙΡΑΜΑΤΑ

### 5.1 Δημιουργία του server

Στη σύγχρονη ανάπτυξη εφαρμογών, υπάρχουν πολλές επιλογές για τη δημιουργία ενός server, με τον πιο συνηθισμένο να είναι η χρήση web frameworks που υποστηρίζουν δυναμικά δεδομένα και επικοινωνία μέσω του πρωτοκόλλου HTTP. Ένα από τα πιο δημοφιλή εργαλεία για τη δημιουργία ενός web server είναι το Node.js, το οποίο χρησιμοποιεί τη γλώσσα προγραμματισμού JavaScript και το framework Express.js. Αυτός ο server επιτρέπει τη δημιουργία web εφαρμογών που ανταποκρίνονται σε αιτήματα μέσω της μεθόδου GET, POST κ.λπ. Όταν εκτελείται ένας τέτοιος server, καθορίζονται διαδρομές (routes) για να υπάρξει απάντηση σε αιτήματα που στέλνουν οι χρήστες. [33] [34]

Όταν δημιουργείται ένας server, συχνά χρησιμοποιείται JSON (JavaScript Object Notation) για την ανταλλαγή δεδομένων μεταξύ του server και του πελάτη (client). Το JSON είναι ένας ελαφρύς και ευανάγνωστος τρόπος αποθήκευσης και αποστολής δεδομένων, χρησιμοποιούμενος κυρίως για την επικοινωνία με web APIs. [33]

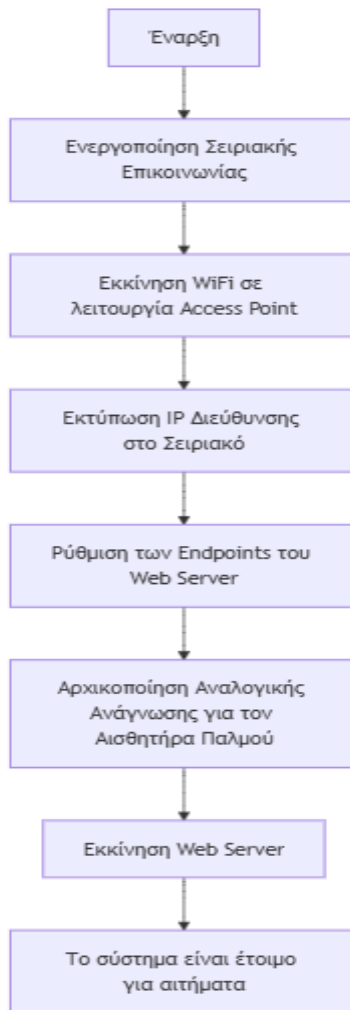
Πέρα από τη βασική λειτουργία του server, πολλές φορές χρειάζεται να αποθηκευτούν δεδομένα σε μια βάση δεδομένων, όπως θα γίνει στην παρούσα Δ.Ε. Αυτό μπορεί να γίνει με την αποστολή δεδομένων από τον client στον server μέσω HTTP αιτημάτων (π.χ., POST), τα οποία περιλαμβάνουν δεδομένα σε μορφή JSON. Στη συνέχεια, ο server επεξεργάζεται αυτά τα δεδομένα και τα αποθηκεύει σε μια βάση δεδομένων, όπως MySQL. [34]

#### 5.1.1 Esp32 Access Point

Όπως αναφέρθηκε και παραπάνω, το ESP32 διαθέτει ενσωματωμένη την λειτουργία Access Point (Σημείο Πρόσβασης), επιτρέποντας τη δημιουργία ενός τοπικού ασύρματου δικτύου στο οποίο άλλες συσκευές μπορούν να συνδεθούν απευθείας. Οι συσκευές που βρίσκονται εντός της εμβέλειας του δικτύου μπορούν να συνδεθούν στο δίκτυο αυτό χρησιμοποιώντας ένα SSID και έναν κωδικό πρόσβασης. Μόλις συνδεθούν, οι συσκευές λαμβάνουν αυτόματα μια διεύθυνση IP από τον ενσωματωμένο διακομιστή DHCP του ESP32. Έτσι, το ESP32 μπορεί να επικοινωνεί με όλες τις συνδεδεμένες συσκευές, ανταλλάσσοντας δεδομένα μέσω του πρωτοκόλλου TCP/IP.

Ο προγραμματισμός του ESP32 ως Access Point γίνεται εύκολα μέσω της Arduino IDE, χρησιμοποιώντας τη βιβλιοθήκη **WiFi.h**. Έτσι, ο χρήστης καθορίζει το SSID, το οποίο είναι το όνομα που εμφανίζεται στις διαθέσιμες ασύρματες συνδέσεις καθώς και έναν κωδικό πρόσβασης, που προστατεύει το δίκτυο από μη εξουσιοδοτημένες. Τέλος, αξίζει να σημειωθεί η συνάρτηση `WiFi.softAP()`, που ενεργοποιεί το access point στον κώδικα.

Παρακάτω ακολουθούν το διάγραμμα ροής και οι εντολές που χρησιμοποιήθηκαν σχετιζόμενες με το Access point, αλλά και η συνάρτηση setup στην οποία εκτελείται.



Εικόνα 9: Διάγραμμα ροής για το access point

```

const char* apSSID = "Esp32-Platform";
const char* apPassword = "123456789";

void setup() {
  Serial.begin(115200);
  WiFi.softAP("Esp32-Platform", "123456789");
  delay(1000);
  IPAddress IP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(IP);
}
  
```

Μετά από αυτό το βήμα και την εύρεση της IP του Esp32, ξεκινάει η δημιουργία του webserver.

### 5.1.2 Front-end κώδικας

Για να δημιουργήσουμε την σελίδα του webserver, χρησιμοποιούμε μια εντολή στο Arduino IDE, ώστε να προγραμματίσουμε σε γλώσσα html-css-javascript και να αποθηκευτεί στην μνήμη Flash του esp32. Η εντολή γράφεται ως εξής:

```
'const char index_html[] PROGMEM = R"rawliteral{
```

Αναλύοντας την εντολή, το `'const char index_html[]` δηλώνει έναν σταθερό πίνακα χαρακτήρων, δηλαδή μεταβλητή τύπου string με το όνομα `'index_html`, ενώ το `'PROGMEM` είναι η ειδική οδηγία που χρειάζεται για να αποθηκευτεί η html στον μικροελεγκτή μέσω της μνήμης Flash.

Υπάρχει και το `= R"rawliteral` δηλώνει ότι το περιεχόμενο μεταξύ των παρενθέσεων, θα διαβάζεται όπως είναι, χωρίς να χρειαστούν escape χαρακτήρες.

#### 5.1.2.1 HTML-CSS

Η χρήση της html ήταν απαραίτητη για την εργασία. Η Html χρησιμοποιείται για δημιουργία ιστοσελίδων και για την δομή και οργάνωσή τους.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Heart Rate Calculator</title>
</head>
<body>
  <div class="container">
    <h1>Heart Rate Calculator</h1>
    <div class="form-group">
      <label for="name">Enter your name:</label>
      <input type="text" id="name" name="name">
    </div>
    <div class="form-group">
      <label for="age">Enter your age (5-75):</label>
      <input type="number" id="age" name="age" min="5" max="75">
    </div>
    <button onclick="calculateHeartRate()">Calculate Heart Rate</button>
    <p id="heartRate"></p>
    <button onclick="saveBPM()">Save BPM</button>
    <canvas id="waveform" width="500" height="200"></canvas>
  </div>
</body>
```

Στον κώδικα φαίνεται η δημιουργία μιας σελίδας με τίτλο Heart Rate Calculator. Στην συνέχεια, υπάρχουν δύο πεδία. Ένα για τον χρήστη να συμπληρώσει το όνομα του και ένα για να συμπληρώσει την ηλικία του. Ο λόγος που υπάρχει το πεδίο του ονόματος, είναι για την αποθήκευση των στοιχείων του χρήστη που θα του γίνει η μέτρηση, ενώ ο λόγος που υπάρχει το πεδίο της ηλικίας είναι επειδή ανά ηλικία, ο μέσος καρδιακός παλμός διαφέρει, όπως εξηγήθηκε και παραπάνω. Στην συνέχεια δημιουργούνται δύο κουμπιά, ένα που λέγεται 'Calculate Heart Rate' και ένα που λέγεται 'Save Bpm'. Το τι κάνουν αυτά τα κουμπιά θα εξηγηθεί στο κομμάτι της javascript.

Κάποια πράγματα για την html είναι σημαντικό να αναφερθούν, όπως το head, που περιλαμβάνει πληροφορίες για την σελίδα, όπως τι είδους χαρακτήρες χρησιμοποιεί, μέσω του charset και τον τίτλο της σελίδας. Το body περιέχει το κύριο κείμενο της σελίδας, ενώ το div class συνδέεται με την css.

Η css είναι η γλώσσα για την διαμόρφωση ιστοσελίδας. Φροντίζει τα κενά, τα χρώματα και την μορφοποίηση των πεδίων. Περιέχεται μέσα στο head της html

```
<style>
  body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #f4f4f4;
  }
  .container {
    background-color: white;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    max-width: 400px;
    width: 100%;
  }
  h1 {
    text-align: center;
    margin-bottom: 20px;
  }
}
```

```
form-group label {    display: block;
    margin-bottom: 5px;
}
.form-group input, .form-group select {
    width: 100%;
    padding: 8px;
    box-sizing: border-box;
}
button {
    width: 100%;
    padding: 10px;
    background-color: #007bff;
    color: white;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}
button:hover {
    background-color: #0056b3;
}
#heartRate {
    margin-top: 20px;
    height: 50px;
}
#waveform {
    width: 100%;
    height: auto;
    border: 1px solid #000000;
    margin-top: 20px;
}
</style>
```

Η css ξεκινάει και τελειώνει με την εντολή style. Αυτό που κατορθώνει, είναι να δημιουργεί κλάσεις οι οποίες στην συνέχεια περιλαμβάνονται στο κομμάτι του body της html, ώστε να διαμορφωθεί η σελίδα.

Αρχικά, η κλάση `body` εφαρμόζεται στο κύριο στοιχείο της σελίδας και είναι υπεύθυνη για τη συνολική διάταξη και εμφάνιση του περιεχομένου της. Η βασική γραμματοσειρά που χρησιμοποιείται είναι η `Agial`, με εναλλακτική επιλογή τη `sans-serif`, εξασφαλίζοντας έτσι ένα καθαρό και ευανάγνωστο κείμενο σε διαφορετικούς τύπους συσκευών και προγραμμάτων περιήγησης. Το περιεχόμενο τοποθετείται στο κέντρο της οθόνης τόσο οριζόντια όσο και κάθετα, μέσω της χρήσης της διάταξης `Flexbox`. Αυτή η διάταξη επιτυγχάνεται με τις ιδιότητες `justify-content` και `align-items`, οι οποίες ευθυγραμμίζουν τα στοιχεία στο κέντρο του διαθέσιμου χώρου. Το ύψος του `body` ορίζεται στο 100% του ορατού τμήματος του παραθύρου του χρήστη, ώστε το περιεχόμενο να προσαρμόζεται πλήρως στην οθόνη χωρίς περιττά κενά. Η επιλογή της ιδιότητας `background-color` με απόχρωση `#f4f4f4` προσφέρει ένα απαλό γκρι φόντο, το οποίο κάνει τα κεντρικά στοιχεία της σελίδας να ξεχωρίζουν ευκολότερα. Το `margin` και το `padding` έχουν μηδενιστεί, ώστε να μην υπάρχει ανεπιθύμητος χώρος γύρω από το περιεχόμενο και η διάταξη να παραμένει σταθερή και συμπαγής.

Η κλάση `container` είναι υπεύθυνη για τη δημιουργία ενός καλοσχεδιασμένου πλαισίου στο κέντρο της ιστοσελίδας, το οποίο περιέχει το κύριο περιεχόμενο της εφαρμογής. Το φόντο της έχει οριστεί σε λευκό χρώμα, προσφέροντας αντίθεση με το γκρι φόντο της υπόλοιπης σελίδας και καθιστώντας το περιεχόμενο πιο ευδιάκριτο. Η χρήση `padding 20 pixel` προσθέτει εσωτερικό κενό γύρω από το περιεχόμενο, αποτρέποντας το κείμενο και τα στοιχεία να ακουμπούν στα άκρα του πλαισίου και διασφαλίζοντας μια πιο ευχάριστη διάταξη. Η ιδιότητα `border-radius` έχει οριστεί σε 8 pixel, δημιουργώντας στρογγυλεμένες γωνίες στο πλαίσιο, κάτι που προσδίδει μια μοντέρνα και φιλική προς τον χρήστη αισθητική. Επιπλέον, η σκιά που εφαρμόζεται μέσω της ιδιότητας `box-shadow` προσφέρει βάθος και οπτικό διαχωρισμό του `container` από το φόντο της σελίδας, κάνοντάς το να φαίνεται ανυψωμένο. Το πλάτος του `container` ρυθμίζεται δυναμικά μέσω των ιδιοτήτων `max-width` και `width`, ώστε να καλύπτει το 100% του διαθέσιμου χώρου σε μικρότερες οθόνες, χωρίς όμως να υπερβαίνει τα 400 pixel σε μεγαλύτερες οθόνες. Αυτή η προσέγγιση διασφαλίζει ότι το περιεχόμενο παραμένει ευανάγνωστο και καλοσχεδιασμένο ανεξάρτητα από τη συσκευή του χρήστη. Συνολικά, η κλάση `container` δημιουργεί ένα ευδιάκριτο, τακτοποιημένο και οπτικά ευχάριστο πλαίσιο για την παρουσίαση του περιεχομένου της ιστοσελίδας, διασφαλίζοντας τη λειτουργικότητα και την καλή εμπειρία χρήστη.

Η επικεφαλίδα `h1` κεντράρεται οπτικά με τη χρήση της ιδιότητας `text-align: center` και αποκτά απόσταση 20 pixel από τα υπόλοιπα στοιχεία μέσω της ιδιότητας `margin-bottom`. Τα στοιχεία της φόρμας οργανώνονται μέσω της κλάσης `form-group`, η οποία εξασφαλίζει απόσταση 15 pixel μεταξύ των πεδίων. Τα `label` τοποθετούνται ως `block` στοιχεία μέσω της ιδιότητας `display: block` και απέχουν 5 pixel από τα πεδία εισαγωγής για καλύτερη ορατότητα. Τα πεδία `input` και `select` προσαρμόζονται δυναμικά στο πλάτος του `container`, καλύπτοντας το 100% του διαθέσιμου χώρου μέσω της ιδιότητας `width: 100%`, και εξασφαλίζουν επαρκές `padding 8 pixel` για τη βελτίωση της εμπειρίας χρήστη.

Τα κουμπιά σχεδιάζονται ώστε να καλύπτουν ολόκληρο το πλάτος του `container` με τη χρήση της ιδιότητας `width: 100%` και να έχουν ομοιόμορφο `padding 10 pixel`. Το φόντο τους διαθέτει μία ζωντανή μπλε απόχρωση μέσω της ιδιότητας `background-color: #007bff`, ενώ η γραμματοσειρά τους είναι λευκή για υψηλή αντίθεση. Με την ιδιότητα `border-radius: 4px`, αποκτούν στρογγυλεμένες γωνίες, ενώ το περίγραμμα αφαιρείται για καθαρότερο σχέδιο. Κατά την αιώρηση του δείκτη, η ιδιότητα `background-color` αλλάζει το χρώμα του φόντου σε ένα πιο σκούρο μπλε (`#0056b3`), προσδίδοντας δυναμικότητα στη διάδραση.

Το στοιχείο με `id="heartRate"` εξασφαλίζει χώρο 50 pixel για την εμφάνιση πληροφοριών καρδιακού παλμού μέσω της ιδιότητας `height`, ενώ διαχωρίζεται από το περιβάλλον του με περιθώριο 20 pixel στην κορυφή. Το `canvas` με `id="waveform"` επεκτείνεται στο πλήρες πλάτος του `container` μέσω της ιδιότητας `width: 100%`, ενώ το ύψος του προσαρμόζεται αυτόματα. Το περίγραμμά του ορίζεται με μαύρη γραμμή πάχους 1 pixel (`border: 1px solid #000000`), ενώ περιθώριο 20 pixel στην κορυφή το διαχωρίζει από τα άλλα στοιχεία.

Μέσα στην κλάση `container` περιλαμβάνονται τα πεδία εισαγωγής δεδομένων, τα κουμπιά, το μήνυμα για τον καρδιακό παλμό αλλά και η κυματομορφή. Καταλήγοντας, η σελίδα εμφανισιακά είναι έτοιμη και στην συνέχεια θα μελετηθεί η `javascript`.

### 5.1.2.2 Javascript

Ο κώδικας της javascript γράφεται μέσα στο script, που βρίσκεται και αυτό στο head της html.

```
<script>

  let waveformInterval;

  function calculateHeartRate() {
    var age = document.getElementById('age').value;
    if (age >= 5 && age <= 75) {
      fetch('/calculate_bpm')
        .then(response => response.json())
        .then(data => {
          const bpm = data.bpm;
          const waveform = data.waveform;
          let message = `Heart Rate: ${bpm} bpm. `;
          if (age >= 5 && age <= 29) {
            if (bpm < 100 || bpm > 170) {
              message += "Your heart rate is not within the normal range for your age. Please
stop exercising.";
            } else {
              message += "Your heart rate is within the normal range.";
            }
          } else if (age >= 30 && age <= 39) {
            if (bpm < 95 || bpm > 162) {
              message += "Your heart rate is not within the normal range for your age. Please
stop exercising.";
            } else {
              message += "Your heart rate is within the normal range.";
            }
          }
        })
    }
  }
}
```

```

} else if (age >= 40 && age <= 49) {
    if (bpm < 93 || bpm > 157) {
        message += "Your heart rate is not within the normal range for your age. Please
stop exercising.";
    } else {
        message += "Your heart rate is within the normal range.";
    }
} else if (age >= 50 && age <= 59) {
    if (bpm < 88 || bpm > 149) {
        message += "Your heart rate is not within the normal range for your age. Please
stop exercising.";
    } else {
        message += "Your heart rate is within the normal range.";
    }
} else if (age >= 60 && age <= 69) {
    if (bpm < 83 || bpm > 140) {
        message += "Your heart rate is not within the normal range for your age. Please
stop exercising.";
    } else {
        message += "Your heart rate is within the normal range.";
    }
} else if (age >= 70 && age <= 75) {
    if (bpm < 78 || bpm > 132) {
        message += "Your heart rate is not within the normal range for your age. Please
stop exercising.";
    } else {
        message += "Your heart rate is within the normal range.";
    }
}

document.getElementById('heartRate').innerText = message;
drawWaveform(waveform);
})
.catch(error => console.error('Error:', error));
} else {
    document.getElementById('heartRate').innerText = "Please enter a valid age (5-
75).";
}
}

```

```

function drawWaveform(waveform) {
  const canvas = document.getElementById('waveform');
  const ctx = canvas.getContext('2d');
  ctx.clearRect(0, 0, canvas.width, canvas.height);
  ctx.beginPath();
  ctx.moveTo(0, canvas.height / 2 + 80);
  for (let i = 0; i < waveform.length; i++) {
    const y = (canvas.height / 2 + 80) - (waveform[i] / 4);
    ctx.lineTo(i * (canvas.width / waveform.length), y);
  }
  ctx.stroke();
}

function saveBPM() {
const name = document.getElementById('name').value;
const age = document.getElementById('age').value;
const bpm = document.getElementById('heartRate').innerText;
const screenshot = document.getElementById('waveform');
fetch('/save_bpm', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({ name, age, bpm, waveform_screenshot:
screenshot.toDataURL() }),
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error('Error:', error));
}
</script>

```

Ο κώδικας χρησιμοποιεί συναρτήσεις όπως το fetch() για να αντλήσει δεδομένα από τον διακομιστή, καθώς και τη μέθοδο σχεδίασης σε καμβά για τη δημιουργία δυναμικών γραφικών.

Η ανάλυση αυτή εξετάζει τον τρόπο λειτουργίας κάθε εντολής, τον τρόπο ελέγχου των ηλικιακών ορίων και την απεικόνιση της κυματομορφής.

Η συνάρτηση `calculateHeartRate()` είναι η κύρια συνάρτηση που υπολογίζει τον καρδιακό ρυθμό του χρήστη. Αρχικά, αντλεί την ηλικία από το πεδίο εισαγωγής της φόρμας μέσω της εντολής `document.getElementById('age').value`. Αν η ηλικία που έχει εισαχθεί βρίσκεται στο αποδεκτό εύρος από 5 έως 75 έτη, τότε η συνάρτηση προχωρά στην αποστολή αιτήματος προς τον διακομιστή μέσω της `fetch()`, ζητώντας δεδομένα `hrm` και την αντίστοιχη κυματομορφή. Η απόκριση από τον διακομιστή επιστρέφεται σε μορφή JSON και αποθηκεύεται σε μεταβλητές. Ο καρδιακός ρυθμός αποθηκεύεται στη μεταβλητή `hrm`, ενώ η κυματομορφή αποθηκεύεται στη μεταβλητή `waveform`. Το μήνυμα που θα εμφανιστεί στον χρήστη ξεκινά με την πληροφορία για τον καρδιακό ρυθμό, ενώ στη συνέχεια εξετάζονται οι ηλικιακές ομάδες και προστίθενται συγκεκριμένα μηνύματα ανάλογα με το αν ο καρδιακός ρυθμός είναι φυσιολογικός ή όχι για την ηλικία του χρήστη.

Η λειτουργία της ανάλυσης των ηλικιακών ορίων γίνεται μέσω συνθηκών `if-else if`. Για την ηλικιακή ομάδα 6-10 ετών, ο φυσιολογικός καρδιακός ρυθμός θεωρείται ότι κυμαίνεται από 70 έως 100 `hrm`. Αν οι χτύποι της καρδιάς του χρήστη είναι εκτός αυτών των ορίων, εμφανίζεται μήνυμα που προειδοποιεί να σταματήσει την άσκηση. Αντιστοίχως, για την ηλικιακή ομάδα 11-14 ετών, ο φυσιολογικός ρυθμός είναι από 60 έως 100 `hrm`, ενώ για τις ηλικίες 15-19 ετών το εύρος μειώνεται σε 50 έως 90 `hrm`. Για τις ηλικίες 20-59 ετών, το φυσιολογικό εύρος κυμαίνεται από 40 έως 80 `hrm`, ενώ για χρήστες άνω των 60 ετών το εύρος αυξάνεται σε 50 έως 90 `hrm`. Αν ο καρδιακός ρυθμός του χρήστη είναι εντός αυτών των ορίων, εμφανίζεται μήνυμα ότι είναι φυσιολογικός. Αν είναι εκτός ορίων, εμφανίζεται μήνυμα προειδοποίησης. Ο κώδικας με όλα τα ηλικιακά όρια βρίσκεται στο παράρτημα.

Η συνάρτηση `fetchWaveform()` λειτουργεί συμπληρωματικά με την `calculateHeartRate()` και καλεί τη συνάρτηση `drawWaveform()` για τη σχεδίαση της κυματομορφής στον καμβά. Η σχεδίαση της κυματομορφής πραγματοποιείται μέσω της συνάρτησης `drawWaveform(waveform)`, η οποία χρησιμοποιεί τον καμβά HTML5 (`canvas`) για να δημιουργήσει μια γραφική αναπαράσταση των παλμών. Αρχικά, μέσω της εντολής `document.getElementById('waveform')`, ανακτάται το στοιχείο του καμβά και δημιουργείται ένα πλαίσιο σχεδίασης μέσω της μεθόδου `getContext('2d')`. Η εντολή `ctx.clearRect(0, 0, canvas.width, canvas.height)` διαγράφει οτιδήποτε υπάρχει ήδη στον καμβά για να αποφευχθεί η επικάλυψη γραφικών από προηγούμενες κυματομορφές.

Η σχεδίαση της κυματομορφής ξεκινά με την εντολή `ctx.beginPath()`, που δηλώνει την έναρξη μιας νέας διαδρομής σχεδίασης. Το αρχικό σημείο της γραμμής καθορίζεται με την εντολή `ctx.moveTo(0, canvas.height / 2 + 80)`, η οποία τοποθετεί τη γραμμή στο μέσο του καμβά. Στη συνέχεια, η συνάρτηση χρησιμοποιεί έναν βρόχο `for` για να σχεδιάσει τη γραμμή της κυματομορφής, διατρέχοντας τα δεδομένα της μεταβλητής `waveform` και προσαρμόζοντας το ύψος της γραμμής ανάλογα με την τιμή του κάθε στοιχείου. Η εντολή `ctx.lineTo(i * (canvas.width / waveform.length), y)` προσθέτει σημεία στη γραμμή της κυματομορφής, με το κάθε σημείο να προσαρμόζεται στο ύψος του καμβά με βάση τις τιμές της κυματομορφής. Τέλος, η εντολή `ctx.stroke()` ολοκληρώνει τη σχεδίαση της γραμμής και την εμφανίζει στον καμβά.

Η συνάρτηση `saveBPM()` είναι υπεύθυνη για την αποθήκευση των δεδομένων του χρήστη, συμπεριλαμβανομένου του ονόματος, της ηλικίας, του καρδιακού ρυθμού και της εικόνας της κυματομορφής. Χρησιμοποιεί την `fetch()` για να στείλει αυτά τα δεδομένα στον διακομιστή σε μορφή JSON. Η εικόνα της κυματομορφής αποθηκεύεται μέσω της μεθόδου `toDataURL()`, η οποία μετατρέπει το περιεχόμενο του καμβά σε μια εικόνα με μορφή `base64`. Τα δεδομένα αποστέλλονται στον διακομιστή μέσω της μεθόδου `POST`, όπου αποθηκεύονται για μελλοντική χρήση.

### 5.1.3 Κώδικας MySQL

```
CREATE DATABASE HeartRateSensor;  
USE HeartRateSensor;  
  
CREATE TABLE HeartRateData (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    age INT NOT NULL,  
    bpm INT NOT NULL,  
    waveform_screenshot TEXT NOT NULL,  
    createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Ο κώδικας αρχικά δημιουργεί μια νέα βάση δεδομένων με το όνομα HeartRateSensor και στη συνέχεια ορίζει έναν πίνακα μέσα στη βάση δεδομένων με το όνομα HeartRateData.

Αρχικά, η εντολή CREATE DATABASE HeartRateSensor; χρησιμοποιείται για να δημιουργηθεί η βάση δεδομένων με το όνομα HeartRateSensor. Αυτή η βάση δεδομένων θα περιέχει έναν ή περισσότερους πίνακες με δεδομένα που σχετίζονται με τις μετρήσεις του καρδιακού παλμού. Στη συνέχεια, η εντολή USE HeartRateSensor; δηλώνει ότι όλες οι επόμενες ενέργειες που θα γίνουν με SQL εντολές θα αφορούν τη συγκεκριμένη βάση δεδομένων.

Η εντολή CREATE TABLE HeartRateData δημιουργεί έναν πίνακα με το όνομα HeartRateData, ο οποίος θα περιέχει τα δεδομένα που συλλέγονται από τον αισθητήρα καρδιακού παλμού. Ο πίνακας αυτός έχει διάφορες στήλες (πεδία), καθεμία από τις οποίες αντιπροσωπεύει μια συγκεκριμένη πληροφορία που αποθηκεύεται για κάθε μέτρηση. Πιο αναλυτικά, η πρώτη στήλη είναι το id, το οποίο έχει τον τύπο INT και ορίζεται ως PRIMARY KEY και AUTO\_INCREMENT. Αυτό σημαίνει ότι το id είναι ο μοναδικός αριθμός που ταυτοποιεί κάθε εγγραφή στον πίνακα και αυξάνεται αυτόματα με κάθε νέα εγγραφή.

Η δεύτερη στήλη είναι το name, που έχει τύπο VARCHAR(255). Αυτό σημαίνει ότι μπορεί να αποθηκεύσει μια συμβολοσειρά με μέγιστο μήκος 255 χαρακτήρες και είναι υποχρεωτικό πεδίο (NOT NULL), δηλαδή δεν μπορεί να παραμείνει κενό. Η τρίτη στήλη είναι το age, που έχει τύπο INT και αποθηκεύει την ηλικία του χρήστη. Και αυτή η στήλη είναι υποχρεωτική, κάτι που υποδηλώνεται από τη δήλωση NOT NULL.

Η επόμενη στήλη είναι το bpm, που επίσης έχει τύπο 'INT' και αποθηκεύει τους καρδιακούς παλμούς ανά λεπτό που έχουν καταγραφεί από τον αισθητήρα. Όπως και τα προηγούμενα πεδία, έτσι και αυτό δεν μπορεί να παραμείνει κενό. Η στήλη waveform\_screenshot έχει τύπο 'TEXT' και χρησιμοποιείται για να αποθηκεύσει δεδομένα σχετικά με το στιγμιότυπο της κυματομορφής που δημιουργήθηκε από την καταγραφή του παλμού. Αυτό το πεδίο

προορίζεται για την αποθήκευση μεγαλύτερων όγκων δεδομένων σε μορφή κειμένου και είναι επίσης υποχρεωτικό.

Τέλος, η στήλη `createdAt` έχει τύπο `'TIMESTAMP'` και χρησιμοποιείται για την αποθήκευση της ημερομηνίας και της ώρας κατά την οποία έγινε η καταγραφή. Η ιδιότητα `'DEFAULT CURRENT_TIMESTAMP'` διασφαλίζει ότι όταν προστίθεται μια νέα εγγραφή στον πίνακα, η τρέχουσα ημερομηνία και ώρα θα αποθηκεύονται αυτόματα σε αυτό το πεδίο.

Συνολικά, ο συγκεκριμένος πίνακας σχεδιάστηκε για να αποθηκεύει με δομημένο τρόπο πληροφορίες σχετικά με τους καρδιακούς παλμούς, όπως το όνομα του χρήστη, την ηλικία του, τον αριθμό των παλμών ανά λεπτό, την εικόνα της κυματομορφής και τον χρόνο καταγραφής. Ο σχεδιασμός του πίνακα εξασφαλίζει ότι κάθε καταγραφή είναι μοναδική και πλήρης, ενώ τα δεδομένα αποθηκεύονται με τρόπο που διευκολύνει τη μελλοντική επεξεργασία και ανάκτησή τους.

### 5.1.3.1 Στιγμιότυπο κυματομορφής

Θέλοντας να γίνει η αποθήκευση του στιγμιότυπου της κυματομορφής, στην παρούσα Δ.Ε, χρησιμοποιείται το `amazon s3 bucket`. Το Amazon S3 (Simple Storage Service) είναι μια υπηρεσία αποθήκευσης στο cloud που παρέχεται από την Amazon Web Services (AWS). Πρόκειται για έναν από τους πιο διαδεδομένους τρόπους αποθήκευσης και ανάκτησης δεδομένων στο διαδίκτυο. Η βασική του λειτουργία είναι να επιτρέπει στους χρήστες να αποθηκεύουν αρχεία κάθε τύπου, όπως εικόνες, βίντεο, αρχεία ήχου ή κειμένου. Όλα αυτά, τα αποθηκεύουν σε μια δομή που ονομάζεται `bucket`.

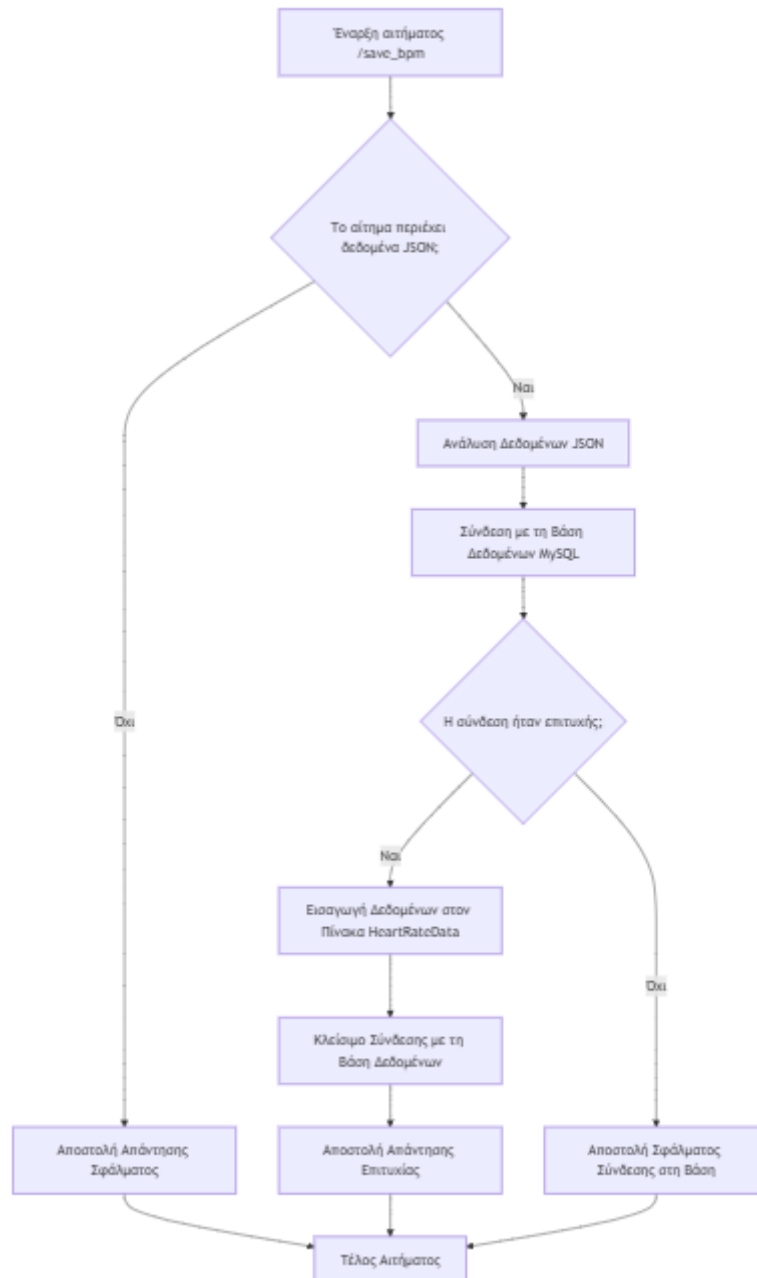
Ο βασικός λόγος που χρησιμοποιήθηκε είναι επειδή έχει υψηλή διαθεσιμότητα και ανθεκτικότητα, καθώς τα δεδομένα που αποθηκεύονται στο S3 αναπαράγονται σε πολλαπλά `data centers`, εξασφαλίζοντας ότι παραμένουν ασφαλή. Ένας ακόμα λόγος είναι η εύκολη διαχείριση δεδομένων, καθώς τα στιγμιότυπα μπορούν να οργανωθούν σε φακέλους και να έχουν διαφορετικά επίπεδα πρόσβασης. Επίσης μπορούν πολύ εύκολα να ανακτηθούν τα δεδομένα αυτόματα μέσω της εφαρμογής και έτσι ο χρήστης μπορεί να τα επεξεργαστεί και να τα αναλύσει με ευκολία.

Όταν δημιουργείται ένα στιγμιότυπο κυματομορφής, το αρχείο ανεβαίνει αυτόματα στο S3 bucket. Η επιλογή αποθήκευσης στο cloud εξασφαλίζει ότι τα δεδομένα είναι πάντοτε προσβάσιμα από τον χρήστη, ακόμα και αν βρίσκεται σε διαφορετική τοποθεσία ή συνδέεται από άλλη συσκευή.

### 5.1.3.2 Κουμπί αποθήκευσης δεδομένων

Το διάγραμμα περιγράφει τη διαδικασία χειρισμού ενός αιτήματος μέσω του `endpoint /save_bpm`, το οποίο χρησιμοποιείται για την αποθήκευση δεδομένων καρδιακού παλμού στην βάση δεδομένων MySQL που δημιουργήθηκε παραπάνω.

Παρατίθεται το διάγραμμα ροής και ο κώδικας για την αποθήκευση των δεδομένων



Εικόνα 10: Διάγραμμα ροής για την αποθήκευση δεδομένων

Αρχικά, η διαδικασία ξεκινά με την παραλαβή του αιτήματος (Start /save\_bpm Request) και την αξιολόγηση αν το αίτημα περιέχει έγκυρα δεδομένα JSON (Request Contains JSON Data?). Αν δεν περιέχει δεδομένα, αποστέλλεται μήνυμα σφάλματος στον πελάτη (Send Error Response) και η διαδικασία τερματίζεται. Εάν περιέχει δεδομένα JSON, αυτά αναλύονται (Parse JSON Data) και επιχειρείται σύνδεση με τη βάση δεδομένων MySQL (Connect to MySQL Database). Στη συνέχεια, ελέγχεται αν η σύνδεση ήταν επιτυχής (Connection Successful?). Αν η σύνδεση αποτύχει, αποστέλλεται μήνυμα σφάλματος για τη βάση δεδομένων (Send Database Connection Error) και η διαδικασία ολοκληρώνεται. Αν η σύνδεση είναι επιτυχής, τα δεδομένα (δηλαδή όνομα, ηλικία, BPM και waveform screenshot) εισάγονται στον πίνακα HeartRateData (Insert Data into HeartRateData Table), κλείνεται η σύνδεση με τη βάση (Close Database Connection), και αποστέλλεται μήνυμα επιτυχούς καταχώρησης (Send

Success Response). Τελικά, ανεξάρτητα από την πορεία, η διαδικασία καταλήγει στο τέλος (End Request).

```
void saveBPM(const char* name, int age, int bpm, const char* base64Screenshot) {
    IPAddress server_ip(192, 168, 1, 188);
    if (conn.connect(server_ip, 3306, user, password_db)) {
        char query[1024];
        sprintf(query,
            "INSERT INTO HeartRateData (name, age, bpm, waveform_screenshot) VALUES
            ('%s', %d, %d, %s)",
            name, age, bpm, base64Screenshot);
        MySQL_Cursor *cur = new MySQL_Cursor(&conn);
        cur->execute(query);
        delete cur;
        conn.close();
        Serial.println("Data saved successfully!");
    } else {
        Serial.println("Connection to database failed.");
    }
}

void handleSaveBPM() {
    if (server.hasArg("plain")) {
        String body = server.arg("plain");
        DynamicJsonDocument doc(2048);
        deserializeJson(doc, body);
        const char* name = doc["name"];
        int age = doc["age"];
        int bpm = doc["bpm"]; const char* screenshot = doc["waveform_screenshot"];
        saveBPM(name, age, bpm, screenshot);
        server.send(200, "application/json", "{\"status\":\"success\"}");
    } else {
        server.send(400, "application/json", "{\"status\":\"error\"}");
    }
}
```

Η πρώτη συνάρτηση `saveBPM` είναι υπεύθυνη για τη σύνδεση στη βάση δεδομένων, τη δημιουργία και την εκτέλεση της SQL εντολής που αποθηκεύει τα δεδομένα που συλλέγονται από τον αισθητήρα, ενώ η δεύτερη συνάρτηση `handleSaveBPM` διαχειρίζεται το POST αίτημα από τον web server, αναλύει τα δεδομένα που περιέχει και καλεί τη συνάρτηση `saveBPM` για την αποθήκευσή τους. Η λειτουργία αυτών των δύο συναρτήσεων εξασφαλίζει την ολοκλήρωση της διαδικασίας συλλογής, επεξεργασίας και αποθήκευσης των μετρήσεων του καρδιακού ρυθμού.

Η συνάρτηση `saveBPM` δέχεται ως παραμέτρους τέσσερις τιμές: το όνομα του χρήστη (`name`), την ηλικία του (`age`), τον αριθμό των χτύπων ανά λεπτό (`bpm`) και την κυματομορφή της καρδιακής δραστηριότητας σε μορφή Base64 (`base64Screenshot`). Αρχικά, γίνεται δήλωση της διεύθυνσης IP της MySQL βάσης δεδομένων μέσω της εντολής `IPAddress server_ip(192, 168, 1, 1)`, η οποία πρέπει να αντικατασταθεί με τη σωστή διεύθυνση της βάσης που χρησιμοποιείται στο δίκτυο. Η σύνδεση με τη βάση δεδομένων επιτυγχάνεται με τη χρήση της εντολής `conn.connect(server_ip, 3306, user, password_db, "heartratesensor")`, όπου ορίζονται η IP, η πόρτα σύνδεσης (3306), το όνομα χρήστη, ο κωδικός πρόσβασης και το όνομα της βάσης δεδομένων. Αν η σύνδεση είναι επιτυχής, εμφανίζεται μήνυμα στην σειριακή κονσόλα με την ένδειξη "Saving to db starting...".

Στη συνέχεια, δημιουργείται η SQL εντολή για την εισαγωγή των δεδομένων στη βάση. Η εντολή `sprintf(query, "INSERT INTO HeartRateData (name, age, bpm, waveform_screenshot) VALUES ('%s', %d, %d, '%s')", name, age, bpm, base64Screenshot)` μορφοποιεί τη δήλωση SQL ώστε να εισάγει τις παραμέτρους στην αντίστοιχη μορφή. Το όνομα, η ηλικία, ο αριθμός των bpm και το στιγμιότυπο της κυματομορφής, που εμφανίζεται στο amazon s3 bucket, προστίθενται στους αντίστοιχους πίνακες της βάσης δεδομένων. Για την εκτέλεση της SQL εντολής χρησιμοποιείται η κλάση `MySQL_Cursor`. Η εντολή `cur->execute(query)` εκτελεί την εισαγωγή των δεδομένων, και στη συνέχεια ο δείκτης `cur` διαγράφεται με τη χρήση της `delete cur` για να απελευθερωθεί η μνήμη. Αφού ολοκληρωθεί η αποθήκευση των δεδομένων, η σύνδεση με τη βάση δεδομένων κλείνει με την εντολή `conn.close()`, και εμφανίζεται μήνυμα επιτυχίας στη σειριακή κονσόλα με την ένδειξη "Data saved successfully!". Αν η σύνδεση αποτύχει, εμφανίζεται το μήνυμα "Connection to database failed."

Η συνάρτηση `handleSaveBPM` είναι υπεύθυνη για τη λήψη των δεδομένων από το POST αίτημα που στέλνει ο client στον web server. Αρχικά, ελέγχει αν υπάρχουν δεδομένα στο αίτημα μέσω της εντολής `if (server.hasArg("plain"))`. Αν υπάρχουν δεδομένα, ανακτώνται μέσω της εντολής `server.arg("plain")` και αποθηκεύονται στη μεταβλητή `body`. Στη συνέχεια, δημιουργείται ένα `DynamicJsonDocument` με μέγεθος 2048 bytes και τα δεδομένα JSON αναλύονται με τη χρήση της `deserializeJson(doc, body)`. Τα δεδομένα που περιέχονται στο JSON αντιστοιχίζονται στις μεταβλητές `name`, `age`, `bpm` και `screenshot`. Οι τιμές αυτές αντιστοιχούν στο όνομα του χρήστη, την ηλικία, τον αριθμό των bpm και το στιγμιότυπο της κυματομορφής αντίστοιχα.

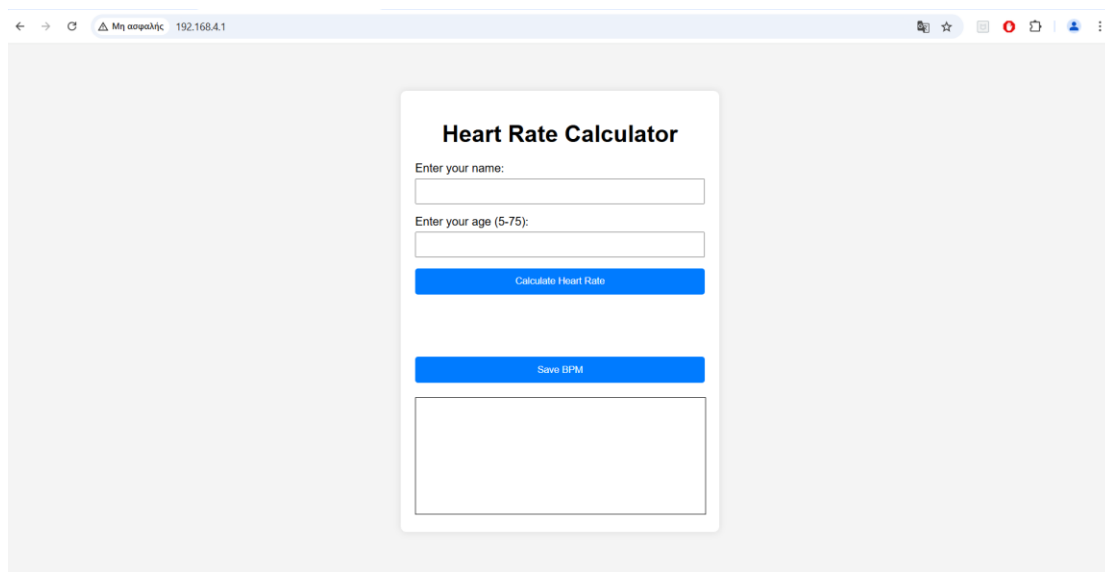
Αφού ανακτηθούν τα δεδομένα, η συνάρτηση `saveBPM` καλείται με τις αντίστοιχες παραμέτρους για να αποθηκεύσει τα δεδομένα στη MySQL βάση δεδομένων. Αν η διαδικασία ολοκληρωθεί επιτυχώς, ο web server στέλνει απάντηση στον client με την ένδειξη 200 (OK) και ένα JSON μήνυμα που δηλώνει την επιτυχία: `server.send(200, "application/json", "{\"status\":\"success\"}").` Αν το POST αίτημα δεν περιλαμβάνει δεδομένα, ο server επιστρέφει μήνυμα σφάλματος με την ένδειξη 400 (Bad Request) και το μήνυμα `{"status\":\"error\"}.`

Οι δύο αυτές συναρτήσεις συνεργάζονται για την ολοκλήρωση της διαδικασίας αποθήκευσης των δεδομένων του καρδιακού ρυθμού. Η `saveBPM` φροντίζει να εισαγάγει τα δεδομένα στη

βάση, ενώ η handleSaveBPM διαχειρίζεται τα αιτήματα από τον client και εξασφαλίζει την ορθή αποστολή των δεδομένων στον μικροελεγκτή. Η αποθήκευση της κυματομορφής σε μορφή Base64 επιτρέπει την αποθήκευση μιας εικόνας που αναπαριστά τη δραστηριότητα της καρδιάς του χρήστη, διασφαλίζοντας ότι τα δεδομένα μπορούν να ανακτηθούν και να παρουσιαστούν οπτικά σε μεταγενέστερο στάδιο.

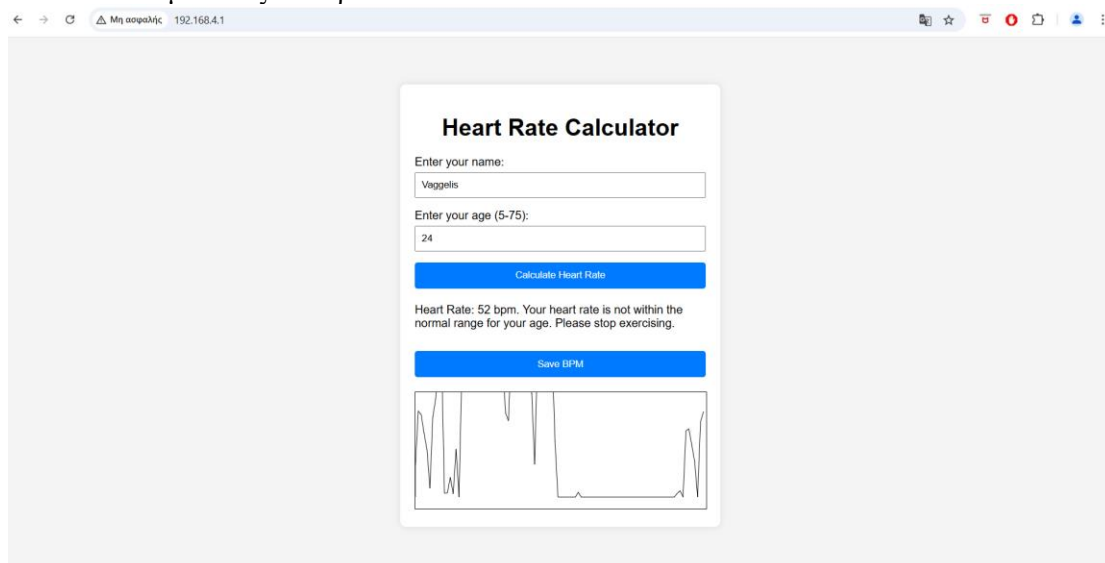
## 5.2 Πειράματα και καταγραφή δεδομένων

Μετά από την χρήση των γλωσσών html, css και javascript, το αποτέλεσμα της σελίδας είναι αυτό



Εικόνα 11: Η ιστοσελίδα με την IP του ESP32

Αρχικά συμπληρώνονται τα πεδία ονόματος και ηλικίας και στην συνέχεια πατιέται το κουμπί calculate heart rate, ενώ το δάχτυλο του χρήστη ακουμπάει τον αισθητήρα. Κατευθείαν, βγαίνει ένα αποτέλεσμα όπως το παρακάτω



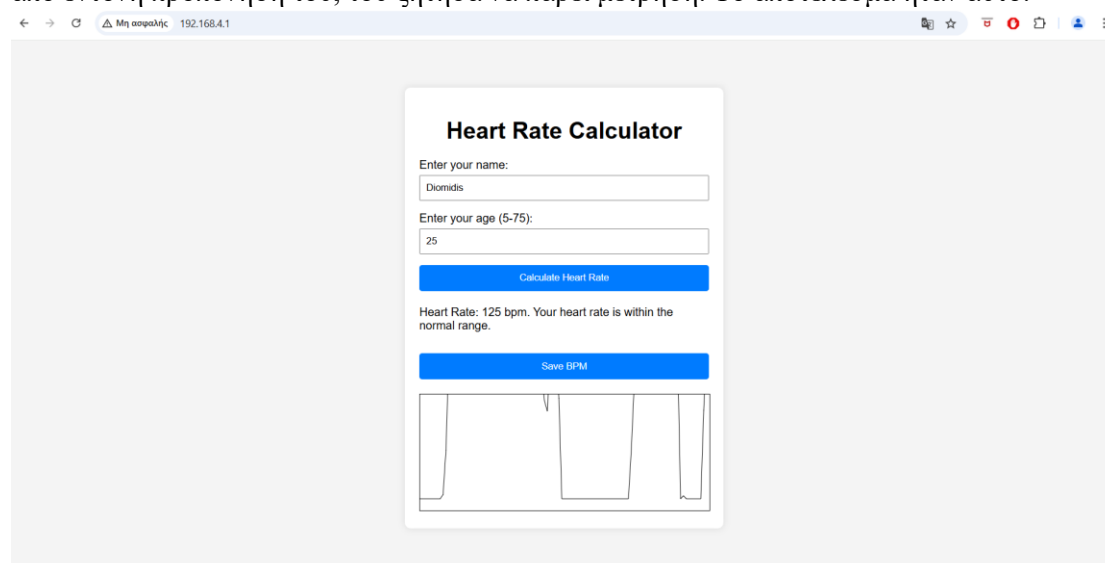
Εικόνα 12: Πείραμα για την πρώτη μέτρηση

Λόγω του ότι δεν είχα γυμναστεί την ώρα του πειράματος, ο καρδιακός παλμός βγήκε κάτω από τα όρια για την ηλικία μου και εμφανίστηκε το μήνυμα ότι το heart rate δεν είναι στο φυσιολογικό εύρος. Κάτω από το κουμπί Save Bpm, φαίνεται η κυματομορφή του ΗΚΓ μου την ώρα που πατήθηκε το Calculate Heart Rate.

Η κυματομορφή δείχνει τα peaks που ξεπερνάνε το threshold και αντιστοιχούν στους παλμούς της καρδιάς.

### 5.2.1 Αποτελέσματα και πειράματα

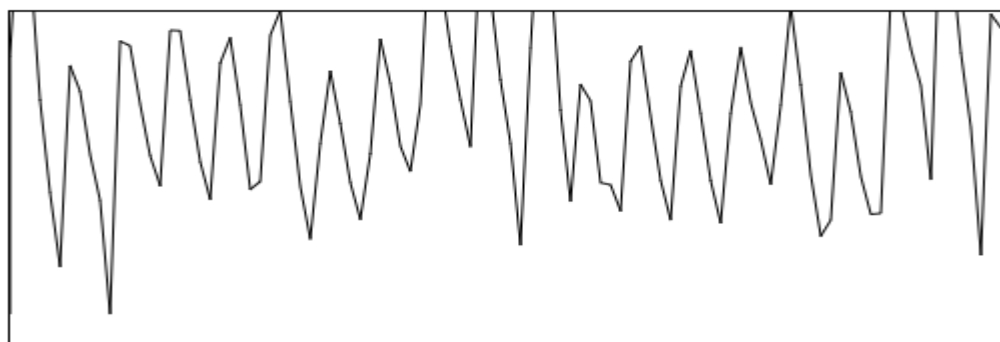
Αφού βεβαιώθηκα ότι το πείραμα λειτουργεί κανονικά, ξεκίνησα τα πειράματα σε άλλους. Αρχικά, ένας φίλος μου, ο Διόμηδης, που είναι μπασκετμπολίστας, δέχτηκε να το κάνει. Μετά από έντονη προπόνηση του, του ζήτησα να πάρει μέτρηση. Το αποτέλεσμα ήταν αυτό:



Εικόνα 13: Πείραμα σε φίλο αθλητή

Ο καρδιακός του παλμός ήταν μέσα στα όρια που είχα θέσει για την ηλικία του. Παρατηρείται επίσης ξεκάθαρα το πόσο πιο ψηλά είναι η κυματομορφή του ΗΚΓ του.

Την δεύτερη φορά στο ίδιο άτομο, η κυματομορφή βγήκε κάπως έτσι:



Εικόνα 14: Η κυματομορφή στην 2<sup>η</sup> προσπάθεια

Η κυματομορφή δείχνει πιο τακτικό σήμα σε σύγκριση με την προηγούμενη, με έντονες κορυφές και κοιλάδες που υποδηλώνουν πιο σταθερή καταγραφή του καρδιακού παλμού. Παρόλα αυτά, υπάρχουν μικρές διακυμάνσεις στο ύψος των κορυφών και ασυνέχειες σε ορισμένα σημεία, πιθανώς λόγω θορύβου ή μη ομαλής επαφής του αισθητήρα με το δέρμα. Ο ρυθμός των κορυφών φαίνεται επαναλαμβανόμενος, γεγονός που υποδεικνύει τη δυνατότητα εξαγωγής BPM.

Πατώντας το κουμπί Save Bpm έγινε η αποθήκευση στην mysql. Επομένως, σχηματίζεται ο πίνακας της mysql με το όνομα, την ηλικία, το bpm και το screenshot της κυματομορφής

Η τελική μορφή του πίνακα μετά τα διάφορα πειράματα σε φίλους και συγγενείς, αφού αθλήθηκαν, είναι τα εξής:

	id	name	age	bpm	waveform_screenshot	createdAt
	1	Vaggelis	24	83	data:image/png;base64,iVBORw0KGgoAAAANS...	2024-06-07 20:35:49
	2	Diomidis	25	125	data:image/png;base64,aGVsbG8gd29ybGQ=	2024-08-18 12:52:10
	3	Michalis	25	133	data:image/png;base64,c29tZSB3YXZlIGRhdGE=	2024-09-23 09:43:12
	4	Maria	22	94	data:image/png;base64,dGVzdCBkYXRhIGhlcmU=	2024-10-01 15:16:38
	5	Chris	21	88	data:image/png;base64,cmFuZG9tIHJmVlbnN...	2024-10-01 15:18:23
	6	Stavros	58	99	data:image/png;base64,Vrt93cmaGE13QmUTIO...	2024-10-03 20:54:31
	7	Ioanna	49	105	data:image/png;base64,R0lGODlhPQBEAPeoAJ...	2024-10-04 16:21:38
	8	Kostas	29	118	data:image/png;base64,iVBORw0KGgoAAAANS...	2024-11-14 22:32:05
	9	George	10	147	data:image/png;base64,PHN2ZyB4bWxucz0iaH...	2024-12-23 09:28:55
	10	Kyriakos	38	107	data:image/png;base64,YW5vdGhlciBzaG90IG9...	2025-01-03 18:23:41
✖		NULL	NULL	NULL	NULL	NULL

Εικόνα 15: Πίνακας με τα αποθηκευμένα στοιχεία

ΠΗΓΗ: MYSQL

Τα πειράματα έγιναν σε οικογένεια και φίλους, όλοι μετά από κάποιου είδους άσκηση, από έντονο περπάτημα μέχρι και άθληση.

### 5.2.2 Συμπεράσματα από τα αποτελέσματα

Από την ανάλυση των αποθηκευμένων δεδομένων στη βάση HeartRateData, παρατηρούνται σημαντικές διαφοροποιήσεις στις τιμές των καρδιακών παλμών σε διαφορετικές ηλικιακές ομάδες. Τα αποτελέσματα δείχνουν ότι οι νεότερες ηλικίες όπως η Μαρία και ο Γιώργος, παρουσιάζουν υψηλότερους καρδιακούς παλμούς σε σχέση με τις μεγαλύτερες ηλικίες, όπως ο Σταύρος. Ειδικότερα, οι υψηλότερες τιμές καρδιακού παλμού εμφανίζονται σε άτομα ηλικίας κάτω των 30 ετών, όπως του Γιώργου (147 bpm) και του Μιχάλη (133 bpm), γεγονός που μπορεί να αποδοθεί στην έντονη σωματική δραστηριότητα που είχαν λίγο πριν την στιγμή της μέτρησης. Αντίθετα, οι μεγαλύτερες ηλικίες τείνουν να έχουν τιμές bpm που βρίσκονται εντός φυσιολογικών ορίων, όπως ο Σταύρος (99 bpm) και η Ιωάννα (105 bpm). Παρατηρείται, επίσης, ότι οι εγγραφές συνοδεύονται από στιγμιότυπα κυματομορφών, τα οποία έχουν αποθηκευτεί σε μορφή base64, κάτι που επιβεβαιώνει την ολοκλήρωση της διαδικασίας αποθήκευσης τόσο των αριθμητικών δεδομένων όσο και των γραφικών δεδομένων. Τέλος, ορισμένες τιμές bpm, όπως του Γιώργου, πλησιάζουν κοντά στα όρια, οπότε μπορεί να

υποδεικνύουν αυξημένο καρδιακό ρυθμό που χρήζει περαιτέρω ανάλυσης και αξιολόγησης. Συνολικά, τα δεδομένα αναδεικνύουν τη χρησιμότητα της παρακολούθησης των καρδιακών παλμών κατά την άθληση.

### 5.2.2.1 Προβλήματα και μειονεκτήματα

Κατά την ανάπτυξη της εφαρμογής προέκυψε ένα τεχνικό πρόβλημα που σχετιζόταν με τη σύνδεση του server με τη βάση δεδομένων MySQL. Ειδικότερα, η εφαρμογή δεν μπορούσε να συνδεθεί σωστά στη βάση, με αποτέλεσμα οι μετρήσεις των καρδιακών παλμών να μην αποθηκεύονται. Το ζήτημα εντοπίστηκε στη ρύθμιση της διεύθυνσης IP του server και στη σωστή διαμόρφωση των username και password, για τη βάση δεδομένων. Η λύση δόθηκε με την ενημέρωση της διεύθυνσης IP στη συνάρτηση connect του ESP32. Επιπλέον, διασφαλίστηκε ότι η σύνδεση κλείνει σωστά μετά την εκτέλεση κάθε query μέσω της εντολής conn.close(), ώστε να αποφεύγονται προβλήματα υπερφόρτωσης του server.

Ένα ακόμα πρόβλημα που υπήρξε, αφορά την αποθήκευση των αποτελεσμάτων των μετρήσεων. Συγκεκριμένα, απορρίφθηκαν περίπου ένα στα τέσσερα αποτελέσματα που δεν αποθηκεύτηκαν στη βάση δεδομένων. Η αιτία του προβλήματος αυτού ήταν οι ελαττωματικές μετρήσεις που παρήγαγαν εξαιρετικά υψηλές τιμές bpm, συνήθως πάνω από 200 παλμούς ανά λεπτό. Ο λόγος που προκλήθηκαν αυτές οι μη ρεαλιστικές τιμές καρδιακού ρυθμού είναι οι παρεμβολές και τα σφάλματα του αισθητήρα κατά τη λήψη των δεδομένων.

## Κεφάλαιο 6<sup>ο</sup>: Συμπεράσματα και μελλοντικές επεκτάσεις

### 6.1 Συμπεράσματα

Το σύστημα μέτρησης καρδιακού παλμού που αναπτύχθηκε με τη χρήση του ESP32 πέτυχε τον στόχο του, παρέχοντας αξιόπιστες μετρήσεις των καρδιακών παλμών (BPM) και απεικονίζοντας την κυματομορφή της καρδιακής δραστηριότητας σε πραγματικό χρόνο. Οι μετρήσεις είναι αρκετά ακριβείς και ανταποκρίνονται άμεσα στις αλλαγές του καρδιακού ρυθμού, ενώ η οπτικοποίηση της κυματομορφής αποτυπώνει με συνέπεια τη φυσιολογική δραστηριότητα του παλμού, διευκολύνοντας την ανάλυση. Το ESP32 αποτέλεσε την ιδανική κεντρική μονάδα χάρη στις ισχυρές δυνατότητες επεξεργασίας και ασύρματης επικοινωνίας, ενώ η χρήση του πρωτοκόλλου JSON εξασφαλίζει ευκολία διασύνδεσης με εξωτερικές εφαρμογές και αποθήκευση δεδομένων σε server. Παράλληλα, το σύστημα είναι επεκτάσιμο και κατάλληλο για εφαρμογές σε αθλητικές δραστηριότητες, με δυνατότητα περαιτέρω ανάπτυξης για ανίχνευση αρρυθμιών ή ενσωμάτωση σε φορητές συσκευές (wearables). Ωστόσο, η ακρίβεια των μετρήσεων μπορεί να επηρεαστεί από εξωτερικούς παράγοντες, όπως η κίνηση του χρήστη ή παρεμβολές στον αισθητήρα, ενώ η κυματομορφή παρέχει μόνο βασική παρακολούθηση και δεν αντικαθιστά την ιατρική διάγνωση. Συνολικά, το σύστημα

αποδείχθηκε αποτελεσματικό και παρέχει μία αξιόπιστη βάση για περαιτέρω βελτιώσεις και εφαρμογές.

## **6.2 Μελλοντικές επεκτάσεις**

Στο πλαίσιο των μελλοντικών επεκτάσεων της διπλωματικής εργασίας, υπάρχουν τρεις ιδέες για βελτίωση της και αξιοποίησης της στο μέγιστο. Πρώτον, η αντικατάσταση του αισθητήρα καρδιακού παλμού με έναν πιο σύγχρονο και προηγμένο αισθητήρα θα βελτιώνε σημαντικά την ακρίβεια των μετρήσεων, δίνοντας πιο ακριβή αποτελέσματα για σωστή ανάλυση. Δεύτερον, θα ήταν πολύ χρήσιμη η ανάλυση του ΗΚΓ που αποθηκεύεται στο S3 bucket, η οποία θα ενσωματώσει τεχνικές μηχανικής μάθησης, με στόχο την ανίχνευση ανωμαλιών και τη δημιουργία προσωποποιημένων συστάσεων για τους χρήστες. Τέλος, η συμπερίληψη ενός συστήματος διαχωρισμού των ειδών γυμναστικής, όπως αερόβιες ασκήσεις, βάρη ή διαλειμματική προπόνηση, θα επέτρεπε τη δυναμική προσαρμογή του συστήματος στις ανάγκες του χρήστη, καθώς κάθε είδος άσκησης έχει διαφορετικές απαιτήσεις και επιδράσεις στο καρδιαγγειακό σύστημα.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Cleveland Clinic, “Heart: Anatomy, Function and Conditions,” Cleveland Clinic. Ανακτήθηκε από: <https://my.clevelandclinic.org/health/body/21704-heart>.
- [2] Δ. Κουτσούρης, Σ. Παυλόπουλος, Α. Πρέντζα, *Εισαγωγή στην βιοϊατρική τεχνολογία και ανάλυση ιατρικών σημάτων*. Θεσσαλονίκη: Εκδόσεις Τζιόλα, 2017.
- [3] Cleveland Clinic, “Sympathetic Nervous System (SNS): Fight or Flight,” Cleveland Clinic. Ανακτήθηκε από: <https://my.clevelandclinic.org/health/body/23262-sympathetic-nervous-system-sns-fight-or-flight>.
- [4] Cleveland Clinic, “Parasympathetic Nervous System (PSNS),” Cleveland Clinic. Ανακτήθηκε από: <https://my.clevelandclinic.org/health/body/23266-parasympathetic-nervous-system-psns>.
- [5] StatPearls, “Heart Physiology,” National Center for Biotechnology Information. Ανακτήθηκε από: <https://www.ncbi.nlm.nih.gov/books/NBK459238/>.
- [6] UC Davis Health, “Athletes and Heart Rate,” UC Davis Sports Medicine. Ανακτήθηκε από: <https://health.ucdavis.edu/sports-medicine/resources/heart-rate>. [7] Ελληνική Καρδιολογική Εταιρεία, “Οι 6 Παράγοντες που Επηρεάζουν την Καρδιά,” Greek Cardiology College. Ανακτήθηκε από: <https://greekcardiology.college/endaferoda-arthra/poioi-oi-6-paragontes-pou-epireazoun-tin-kardia/>.
- [8] Life Ergo, “Heart Rate Training,” Life Ergo. Ανακτήθηκε από: <https://www.lifergo.gr/hr-training/>.
- [9] Ιατροπαιδεία, “Καρδιά και Υγεία,” Ιατροπαιδεία. Ανακτήθηκε από: <https://www.iatropedia.gr/ygeia>.
- [10] StatPearls, “Cardiovascular System Anatomy and Physiology,” National Center for Biotechnology Information. Ανακτήθηκε από: <https://www.ncbi.nlm.nih.gov/books/NBK549803/>.
- [11] Cleveland Clinic, “Electrocardiogram (EKG),” Cleveland Clinic. Ανακτήθηκε από: <https://my.clevelandclinic.org/health/diagnostics/16953-electrocardiogram-ekg>.
- [12] StatPearls, “Cardiac Pacemaker Function,” National Center for Biotechnology Information. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK537194/>.
- [13] Καρδιολογία και Υγεία, “Προαθλητικός Καρδιολογικός Έλεγχος,” Ιπποκράτης. Ανακτήθηκε από: [https://ipokratis.gr/plugins/system/vm\\_multiupload\\_attachment/js/multiupload/server/uploads/182/ΠΕΠΙΕΧΟΜΕΝΑ.pdf](https://ipokratis.gr/plugins/system/vm_multiupload_attachment/js/multiupload/server/uploads/182/ΠΕΠΙΕΧΟΜΕΝΑ.pdf).
- [14] Care Medic, “Καρδιογράφημα (ECG),” Care Medic. Ανακτήθηκε από: <https://www.caremedic.gr/cardiologos-kardiografima-ecg-ekg/>.
- [15] Mayo Clinic, “Electrocardiogram (ECG),” Mayo Clinic. Ανακτήθηκε από: <https://www.mayoclinic.org/tests-procedures/ekg/about/pac-20384983>.
- [16] UPMC, “Athlete Heart Rate: What You Need to Know,” UPMC Health Beat. Ανακτήθηκε από: <https://share.upmc.com/2021/02/athlete-heart-rate/>.
- [17] Iatronet, “Καρδιολογικές Παθήσεις,” Iatronet. Ανακτήθηκε από: <https://iatronet.gr>.
- [18] The Healthy Plan, “Τα Οφέλη της Άσκησης,” The Healthy Plan. Ανακτήθηκε από: <https://thehealthyplan.gr/ta-ofeli-tis-askisis-stin-somatiki-kai-tin-psychiki-ygeia>.
- [19] Χαλάτσης, “Προαθλητικός Καρδιολογικός Έλεγχος,” Χαλάτσης. Ανακτήθηκε από: <https://halatsis.gr/προαθλητικός-καρδιολογικός-έλεγχος/>.
- [20] Espressif Systems, “ESP32 Data Sheet,” Espressif. Available: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf). [21] Flux AI, “Exploring ESP32: From History to Application,” Flux AI. Ανακτήθηκε από: <https://www.flux.ai/p/blog/exploring-esp32-from-history-to-application>.
- [22] Nabto, “Guide to IoT: ESP32,” Nabto. Ανακτήθηκε από: <https://www.nabto.com/guide-to-iot-esp-32>.

- [23] Flux AI, “ESP32 vs. Arduino: Why Choose One Over the Other,” Flux AI. Ανακτήθηκε από: <https://www.flux.ai/p/blog/esp32-vs-arduino-why-choose-one-over-the-other-in-2023>.
- [24] Modtronix, “ESP-32S IoT Development Board,” Modtronix. <https://modtronix.com/product/esp-32s/>.
- [25] Circuito.io, “Arduino Code Basics,” Circuito.io. Ανακτήθηκε από: <https://www.circuito.io/blog/arduino-code/>.
- [26] Last Minute Engineers, “Pulse Sensor Tutorial,” Last Minute Engineers. <https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/>.
- [27] J. Papaioannou, “Heart Rate Sensors in IoT Applications,” in *Proc. Polytechnic Conference on IoT Systems*, Athens, Greece, 2021, pp. 24-30.
- [28] News Medical. (n.d.). *Photoplethysmography (PPG)*. Ανακτήθηκε από: [https://www.news-medical.net/health/Photoplethysmography-\(PPG\).aspx](https://www.news-medical.net/health/Photoplethysmography-(PPG).aspx)
- [29] Wizatronics. (n.d.). *Γενικά για τους μικροελεγκτές*. Ανακτήθηκε από: <https://wizatronics.webnode.gr/news/%CE%B3%CE%B5%CE%BD%CE%B9%CE%BA%CE%AC-%CE%B3%CE%B9%CE%B1-%CF%84%CE%BF%CF%85%CF%82-%CE%BC%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AD%CF%82/>
- [30] IoT Design Pro. (2020). *IoT Heartbeat Monitoring System using Raspberry Pi*. Ανακτήθηκε από: <https://iotdesignpro.com/projects/iot-heartbeat-monitoring-system-using-raspberry-pi>
- [31] Circuito.io. (2020). *Arduino Code*. Ανακτήθηκε από: <https://www.circuito.io/blog/arduino-code/>
- [32] Last Minute Engineers. (2020). *Pulse Sensor Arduino Tutorial*. Ανακτήθηκε από: <https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/>
- [33] Tanenbaum, Wetherall, *ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ*. Εκδόσεις Κλειδάριθμος, σελ.39-40
- [34] Kurose, Ross, *Δικτύωση Υπολογιστών: Προσέγγιση από Πάνω προς τα Κάτω*, σελ.84-86

## ΠΑΡΑΡΤΗΜΑ Α : ΚΩΔΙΚΑΣ

```
#include <WiFi.h>
#include <WebServer.h>
#include <MySQL_Cursor.h>
#include <ArduinoJson.h>

#define PulseSensor_PIN 36

const char* apSSID = "Esp32-Platform";
const char* apPassword = "123456789";

char user[] = "arduino";           //Όνομα χρήστη MySQL
char password_db[] = "Password12!!";
WiFiClient client;
MySQL_Connection conn((Client *)&client);

WebServer server(80);

unsigned long lastBeatTime = 0;
unsigned long currentBeatTime = 0;
int Signal;
int threshold = 550;
bool inPeak = false;
int BPM = 0;
String waveform = "";

unsigned long previousMillis = 0;
const long interval = 20000; // 20 seconds

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Heart Rate Calculator</title>
<script>
  let waveformInterval;

  function calculateHeartRate() {
    var age = document.getElementById('age').value;
    if (age >= 5 && age <= 75) {
      fetch('/calculate_bpm')
        .then(response => response.json())
        .then(data => {
          console.log(data); // Debugging: log the received data
          const bpm = data.bpm;
          const waveform = data.waveform;
          let message = `Heart Rate: ${bpm} bpm. `;
          if (age >= 5 && age <= 75) {
            fetch('/calculate_bpm')
              .then(response => response.json())
              .then(data => {
                console.log(data); // Debugging: log the received data
                const bpm = data.bpm;
                const waveform = data.waveform;
                let message = `Heart Rate: ${bpm} bpm. `;

                // Age group 6-10
                if (age >= 6 && age <= 10) {
                  if (bpm < 70 || bpm > 100) {
                    message += "Your heart rate is not within the normal range for your age. Please stop exercising.";
                  } else {
                    message += "Your heart rate is within the normal range.";
                  }
                }
              }
            )
          }
          // Age group 11-14
          else if (age >= 11 && age <= 14) {
            if (bpm < 60 || bpm > 100) {
              message += "Your heart rate is not within the normal range for your age. Please stop exercising.";
            }
          }
        }
      )
    }
  }
}

```

```

    } else {
        message += "Your heart rate is within the normal range.";
    }
}

// Age group 15-19
else if (age >= 15 && age <= 19) {
    if (bpm < 50 || bpm > 90) {
        message += "Your heart rate is not within the normal range for your age. Please stop
exercising.";
    } else {
        message += "Your heart rate is within the normal range.";
    }
}

// Age group 20-29
else if (age >= 20 && age <= 29) {
    if (bpm < 40 || bpm > 80) {
        message += "Your heart rate is not within the normal range for your age. Please stop
exercising.";
    } else {
        message += "Your heart rate is within the normal range.";
    }
}

// Age group 30-39
else if (age >= 30 && age <= 39) {
    if (bpm < 40 || bpm > 80) {
        message += "Your heart rate is not within the normal range for your age. Please stop
exercising.";
    } else {
        message += "Your heart rate is within the normal range.";
    }
}

// Age group 40-49
else if (age >= 40 && age <= 49) {
    if (bpm < 40 || bpm > 80) {
        message += "Your heart rate is not within the normal range for your age. Please stop
exercising.";
    } else {
        message += "Your heart rate is within the normal range.";
    }
}

```

```

    }
  }
  // Age group 50-59
  else if (age >= 50 && age <= 59) {
    if (bpm < 40 || bpm > 80) {
      message += "Your heart rate is not within the normal range for your age. Please stop
exercising.";
    } else {
      message += "Your heart rate is within the normal range.";
    }
  }
  // Age group 60+
  else if (age >= 60) {
    if (bpm < 50 || bpm > 90) {
      message += "Your heart rate is not within the normal range for your age. Please stop
exercising.";
    } else {
      message += "Your heart rate is within the normal range.";
    }
  }

  document.getElementById('heartRate').innerText = message;
  drawWaveform(waveform);
})
.catch(error => console.error('Error:', error));
} else {
  document.getElementById('heartRate').innerText = "Please enter a valid age (5-75).";
}

```

```

function fetchWaveform() {
  fetch('/get_waveform')
    .then(response => response.json())
    .then(data => {
      console.log(data); // Debugging: log the received data
      const waveform = data.waveform;
      drawWaveform(waveform);
    })
    .catch(error => console.error('Error:', error));
}

```

```

}

function drawWaveform(waveform) {
  const canvas = document.getElementById('waveform');
  const ctx = canvas.getContext('2d');
  ctx.clearRect(0, 0, canvas.width, canvas.height);
  ctx.beginPath();
  ctx.moveTo(0, canvas.height / 2 + 80);
  for (let i = 0; i < waveform.length; i++) {
    const y = (canvas.height / 2 + 80) - (waveform[i] / 4);
    ctx.lineTo(i * (canvas.width / waveform.length), y);
  }
  ctx.stroke();
  ctx.beginPath();
}

function saveBPM() {
  const name = document.getElementById('name').value;
  const age = document.getElementById('age').value;
  const bpm = document.getElementById('heartRate').innerText;
  const screenshot = document.getElementById('waveform');

  // Send the data as JSON to the server
  fetch('/save_bpm', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({ name, age, bpm, waveform_screenshot: screenshot.toDataURL()
  })),
  })
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error('Error:', error));
}
</script>

```

```
<style>
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  background-color: #f4f4f4;
}
.container {
  background-color: white;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  max-width: 400px;
  width: 100%;
}
h1 {
  text-align: center;
  margin-bottom: 20px;
}
.form-group {
  margin-bottom: 15px;
}
.form-group label {
  display: block;
  margin-bottom: 5px;
}
.form-group input, .form-group select {
  width: 100%;
  padding: 8px;
  box-sizing: border-box;
}
button {
  width: 100%;
```

```

padding: 10px;
background-color: #007bff;
color: white;
border: none;
border-radius: 4px;
cursor: pointer;
}
button:hover {
background-color: #0056b3;
}
#heartRate {
margin-top: 20px;
height: 50px; /* Προσθήκη χώρου για τα μηνύματα */
}
#waveform {
width: 100%;
height: auto;
border: 1px solid #000000;
margin-top: 20px;
}
</style>
</head>
<body>
<div class="container">
<h1>Heart Rate Calculator</h1>

<!-- Name Input Field -->
<div class="form-group">
<label for="name">Enter your name:</label>
<input type="text" id="name" name="name">
</div>

<!-- Age Input Field -->
<div class="form-group">
<label for="age">Enter your age (5-75):</label>
<input type="number" id="age" name="age" min="5" max="75">
</div>

```

```

<button onclick="calculateHeartRate()">Calculate Heart Rate</button>
<p id="heartRate"></p>
<button onclick="saveBPM()">Save BPM</button>
<canvas id="waveform" width="500" height="200"></canvas>
</div>
</body>
</html>

```

```
)rawliteral";
```

```

void handleRoot() {
  server.send_P(200, "text/html", index_html);
}

```

```

void calculateBPM() {
  readSensorData();
  String json = "{\"bpm\":" + String(BPM) + ", \"waveform\":" + waveform + "}";
  server.send(200, "application/json", json);
  Serial.println(json); // Debugging: log the sent JSON data
}

```

```

void getWaveform() {
  readSensorData();
  String json = "{\"waveform\":" + waveform + "}";
  server.send(200, "application/json", json);
  Serial.println(json); // Debugging: log the sent JSON data
}

```

```

void readSensorData() {
  waveform = "";
  int peakCount = 0;
  for (int i = 0; i < 100; i++) {
    Signal = analogRead(PulseSensor_PIN);
    waveform += String(Signal) + ((i < 99) ? "," : "");
    delay(20);
    if (Signal > threshold && !inPeak) {

```

```

    inPeak = true;
    currentBeatTime = millis();
    if (lastBeatTime > 0) {
        unsigned long beatInterval = currentBeatTime - lastBeatTime;
        BPM = 60000 / beatInterval;
    }
    lastBeatTime = currentBeatTime;
    peakCount++;
}
if (Signal < threshold && inPeak) {
    inPeak = false;
}
}
if (peakCount == 0) {
    BPM = 0;
}
}

void saveBPM(const char* name, int age, int bpm, const char* base64Screenshot) {
    IPAddress server_ip(192, 168, 1, 1);
    Serial.println("Saving to db starting...");
    if (conn.connect(server_ip, 3306, user, password_db, "heartratesensor")) {
        char query[1024];
        sprintf(query,
            "INSERT INTO HeartRateData (name, age, bpm, waveform_screenshot) VALUES
            (%s', %d, %d, '%s')",
            name, age, bpm, base64Screenshot);
        MySQL_Cursor *cur = new MySQL_Cursor(&conn);
        cur->execute(query);
        delete cur;
        conn.close();
        Serial.println("Data saved successfully!");
    } else {
        Serial.println("Connection to database failed.");
    }
}

void handleSaveBPM() {

```

```

if (server.hasArg("plain")) {
  String body = server.arg("plain");
  DynamicJsonDocument doc(2048);
  deserializeJson(doc, body);
  const char* name = doc["name"];
  int age = doc["age"];
  int bpm = doc["bpm"];
  const char* screenshot = doc["waveform_screenshot"];
  saveBPM(name, age, bpm, screenshot);
  server.send(200, "application/json", "{\"status\":\"success\"}");
} else {

  server.send(400, "application/json", "{\"status\":\"error\"}");
}
}

void setup() {
  Serial.begin(115200);

  WiFi.softAP("Esp32-Platform", "123456789");

  delay(1000);

  IPAddress IP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(IP);

  // Ρύθμιση των endpoint για τον web server
  server.on("/", HTTP_GET, handleRoot);
  server.on("/calculate_bpm", HTTP_GET, calculateBPM);
  server.on("/get_waveform", HTTP_GET, getWaveform); // Προσθήκη νέου endpoint
  server.on("/save_bpm", HTTP_POST, handleSaveBPM);
  server.begin();

  analogReadResolution(10);

```

```

}

void loop() {
  server.handleClient();

  Signal = analogRead(PulseSensor_PIN);

  if (Signal > threshold && !inPeak) {
    inPeak = true;
    currentBeatTime = millis();
    if (lastBeatTime > 0) {
      unsigned long beatInterval = currentBeatTime - lastBeatTime;
      BPM = 60000 / beatInterval;
    }
    lastBeatTime = currentBeatTime;
  }

  if (Signal < threshold && inPeak) {
    inPeak = false;
  }

  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    Serial.print("Current BPM: ");
    Serial.println(BPM);
  }

  delay(20);
}

```





