



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη εξατομικευμένης εφαρμογής γυμναστικής βασισμένη  
στην τεχνητή νοημοσύνη**

**Νικόλαος Καλόφωνος**  
**A.M. 19390074**

Εισηγητής: Επ. Καθηγητής Χρήστος Τρούσσας



**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη εξατομικευμένης εφαρμογής γυμναστικής βασισμένη στην τεχνητή νοημοσύνη**

**Νικόλαος Καλόφωνος  
Α.Μ. 19390074**

**Εισηγητής:**

**Χρήστος Τρούσσας, Επ. Καθηγητής**

**Εξεταστική Επιτροπή:**

**Χρήστος Τρούσσας, Επίκουρος Καθηγητής**

**Ακριβή Κρούσκα, Μέλος ΕΔΙΠ**

**Φοίβος Μυλωνάς, Αναπληρωτής Καθηγητής**

**Ημερομηνία εξέτασης: Σεπτέμβριος, 2024**



## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Νικόλαος Καλόφωνος του Δημητρίου, με αριθμό μητρώου 19390074 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

«Βεβαιώνω ότι είμαι συγγραφέας της παρούσας διπλωματικής εργασίας και ότι έχω αναφέρει ή παραπέμψει σε αυτή, ρητά και συγκεκριμένα, όλες τις πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, προτάσεων ή λέξεων, είτε αυτές μεταφέρονται επακριβώς (στο πρωτότυπο ή μεταφρασμένες) είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για την συγκεκριμένη διπλωματική εργασία»

Ο Δηλών



Νικόλαος Καλόφωνος



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα να αφιερώσω αυτή τη διπλωματική εργασία την οικογένειά μου και τους φίλους μου για τη στήριξη και τη συμπαράσταση τους κατά τη διάρκεια των σπουδών μου. Επίσης θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου για την καθοδήγησή του.





## ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία παρουσιάζει την ανάπτυξη μιας εξατομικευμένης εφαρμογής γυμναστικής που έχει σχεδιαστεί για να βοηθήσει στην αποκατάσταση τραυματισμών μέσω προσαρμοσμένων εβδομαδιαίων πλάνων προπόνησης που δημιουργούνται από ένα τεχνητό νευρωνικό δίκτυο (ANN). Το ANN εκπαιδεύεται στην καταλληλότητα των ασκήσεων για διαφορετικούς χρήστες, εξασφαλίζοντας ακριβείς εξατομικευμένες συστάσεις.

Κατασκευασμένη με React Native και Django, η εφαρμογή παρέχει τις κατάλληλες ασκήσεις με βάση τα δεδομένα του χρήστη, όπως ο τύπος τραυματισμού, το επίπεδο φυσικής κατάστασης ο δείκτης μάζας σώματος, η ηλικία και ο διαθέσιμος εξοπλισμός. Τα βασικά χαρακτηριστικά της εφαρμογής περιλαμβάνουν τη δημιουργία εξατομικευμένων προπονήσεων και την ανατροφοδότηση χρήστη με βάση τις επιδόσεις τού για συνεχείς προσαρμογές του πλάνου.

Η διεπαφή χρήστη (UI) της εφαρμογής είναι σχεδιασμένη έτσι ώστε να δίνει προτεραιότητα στην προσβασιμότητα και την ευκολία χρήσης. Αυτός ο στόχος επιτυγχάνεται μέσω του απλοϊκού σχεδιασμού της διεπαφής ώστε να γίνεται κατανοητή από κάθε χρήστη, καθώς και μέσω της ενσωμάτωσης εικόνων και αναλυτικών περιγραφών για κάθε άσκηση που συμπεριλαμβάνεται.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Ανάπτυξη Ανθρωποκεντικού Λογισμικού

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Εξατομίκευση, Αποκατάσταση τραυματισμών, Νευρωνικά δίκτυα, Ανατροφοδότηση Χρήστη

## **ABSTRACT**

This thesis presents the development of a personalized fitness application designed to aid injury rehabilitation through customized weekly training plans generated by an artificial neural network (ANN). The ANN is trained on the suitability of exercises for different users, ensuring accurate personalized recommendations.

Built with React Native and Django, the app provides appropriate exercises based on user data such as injury type, fitness level, body mass index, age and available equipment. Key features of the app include the creation of personalized workouts and performance-based user feedback for continuous plan adjustments based on the user's performance.

The app's user interface (UI) is designed to prioritize accessibility and ease of use. This goal is achieved through the simplistic design of the interface so that it can be understood by any user, as well as through the inclusion of images and detailed descriptions for each exercise included.

SCIENTIFIC FIELD: Human Centric Software Development

KEYWORDS: Personalization, Injury rehabilitation, Neural networks, User feedback

## **Περιεχόμενα**

|   |    |
|---|----|
| Κεφάλαιο 1: Εισαγωγή.....   | 12 |
| 1.1 Σκοπός και Στόχοι.....  | 12 |
| 1.2 Ιστορική Αναδρομή.....  | 12 |
| 1.3 Καινοτομία.....   | 14 |
| 1.4 Μεθοδολογία.....  | 14 |
| Κεφάλαιο 2: Αποκατάσταση τραυματισμών και συλλογή δεδομένων για νευρωνικά δίκτυα..... | 17 |
| 2.1 Επισκόπηση των τραυματισμών.....  | 17 |
| 2.2 Συλλογή και κατηγοριοποίηση δεδομένων.....  | 20 |
| Κεφάλαιο 3: Νευρωνικά Δίκτυα για Εξατομικευμένη Αποκατάσταση Τραυματισμού.....        | 23 |
| 3.1 Εισαγωγή στα Νευρωνικά Δίκτυα.....  | 23 |
| 3.2 Δημιουργία και Εκπαίδευση του ANN για την εφαρμογή.....                           | 25 |
| 3.2.1 Συλλογή δεδομένων και αξιολογισή καταλληλότητας.....                            | 26 |
| 3.2.2 Αρχιτεκτονική νευρωνικού δικτύου και διαδικασία εκπαίδευσης.....                | 27 |
| 3.2.3 Παραγωγή πλάνων γυμναστικής.....  | 32 |
| Κεφάλαιο 4: Σχεδίαση και Υλοποίηση του Συστήματος.....                                | 34 |
| 4.1 Επισκόπηση αρχιτεκτονικής.....  | 34 |
| 4.2 Ανάλυση του Front-End.....  | 35 |
| 4.2 Ανάλυση του Back-End.....   | 43 |
| Κεφάλαιο 5: Συμπεράσματα.....   | 48 |
| 5.1 Περιπτώσεις χρήσης.....   | 48 |
| 5.2 Μελλοντικές Επεκτάσεις.....   | 49 |
| 5.3 Συμπεράσματα.....   | 50 |
| Κεφάλαιο 6: Βιβλιογραφία-Αναφορές-Διαδικτυακές Πηγές.....                             | 52 |
| 6.1 Αναφορές που χρησιμοποιούνται στη διατριβή.....                                   | 52 |
| 6.2 Πηγές που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.....                    | 52 |



## **Κεφάλαιο 1: Εισαγωγή**

### **1.1 Σκοπός και Στόχοι**

Η ραγδαία ανάπτυξη της τεχνολογίας υγείας και το αυξανόμενο ενδιαφέρον για εφαρμογές γυμναστικής έχουν ανοίξει νέους δρόμους για την υποστήριξη της αποκατάστασης τραυματισμών. Η παρούσα διπλωματική εργασία επικεντρώνεται στο σχεδιασμό και την ανάπτυξη μιας εφαρμογής γυμναστικής που αξιοποιεί τεχνικές μηχανικής μάθησης για τη δημιουργία σχεδίων προπόνησης προσαρμοσμένων στις ανάγκες των ατόμων που αναρρώνουν από τραυματισμούς. Το πρόβλημα που αντιμετωπίζεται είναι η έλλειψη προσβάσιμων, φιλικών προς τον χρήστη εργαλείων που μπορούν να προσαρμόζουν δυναμικά τα προγράμματα άσκησης με βάση τις ατομικές ανάγκες και την πρόοδο κατά την διαδικασία της αποκατάστασης. Με την ενσωμάτωση συστάσεων με βάση την τεχνητή νοημοσύνη, η εφαρμογή αυτή στοχεύει στη βελτίωση της αποτελεσματικότητας των προγραμμάτων αποκατάστασης της φυσικής κατάστασης, καθιστώντας το ένα σχετικό θέμα στον αναπτυσσόμενο τομέα της τεχνολογίας υγείας.

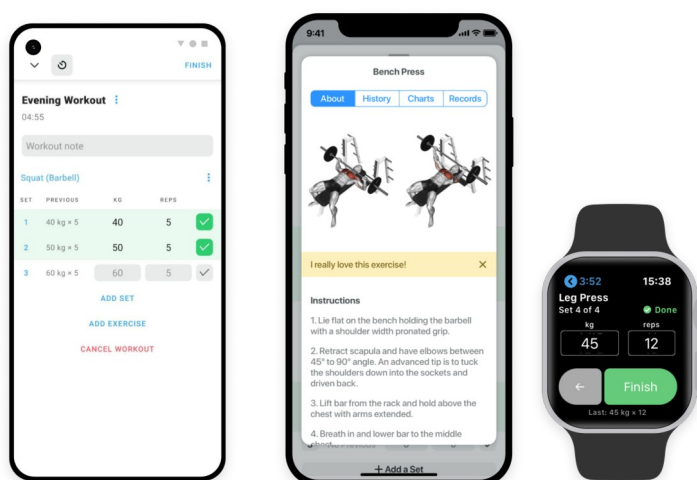
### **1.2 Ιστορική Αναδρομή**

Η διασταύρωση της τεχνολογίας και του fitness έχει υπάρξει θέμα πολλών συζητήσεων τις τελευταίες δεκαετίες, λόγω των ραγδαίων εξελίξεων στις κινητές συσκευές, την φορητή wearable τεχνολογία και την τεχνητή νοημοσύνη. Στις αρχές της δεκαετίας του 2000, οι εφαρμογές γυμναστικής επικεντρώνονταν κυρίως σε στατικά σχέδια προπόνησης και στη βασική παρακολούθηση σωματικών δραστηριοτήτων όπως το τρέξιμο ή η ποδηλασία. Απέκτησαν αρχικά δημοτικότητα ως βασικά εργαλεία που χρησιμοποιούσαν smartphones για την παρακολούθηση φυσικών δραστηριοτήτων όπως το περπάτημα και το τρέξιμο. Αυτές οι πρώτες εφαρμογές μετρούσαν τα βήματα και υπολόγιζαν τις θερμίδες που καίγονταν, προσφέροντας στους χρήστες έναν απλό τρόπο παρακολούθησης των καθημερινών επιπέδων δραστηριότητάς τους. Ωστόσο, παρείχαν μια περιορισμένη όψη της φυσικής κατάστασης, εστιάζοντας κυρίως στην ποσοτικοποίηση της σωματικής δραστηριότητας χωρίς να προσφέρουν λεπτομερείς πληροφορίες ή εξατομικευμένη καθοδήγηση για τη συνολική βελτίωση της φυσικής κατάστασης. Αυτές οι

πρώιμες λύσεις στερούνταν προσαρμοστικότητας, προσφέροντας λύσεις ενός μεγέθους για όλους. Ωστόσο, με την άνοδο των smartphones και την αυξημένη πρόσβαση σε προσωπικά δεδομένα υγείας μέσω wearables όπως το Fitbit και το Apple Watch, η ανάγκη για πιο προσαρμοσμένες λύσεις γυμναστικής έγινε εμφανής.

Μέχρι τα μέσα της δεκαετίας του 2010, η ενσωμάτωση της τεχνητής νοημοσύνης και της μηχανικής μάθησης στις εφαρμογές υγείας και γυμναστικής άρχισε να αυξάνεται, επιτρέποντας πιο δυναμικές και εξατομικευμένες εμπειρίες χρηστών. Οι συστάσεις με τεχνητή νοημοσύνη έγιναν πιο συνηθισμένες σε τομείς όπως η διατροφή, τα σχέδια προπόνησης και η ψυχική υγεία. Συγκεκριμένα, άρχισαν να εφαρμόζονται μοντέλα μηχανικής μάθησης για την ανάλυση δεδομένων χρηστών και την πρόβλεψη βέλτιστων σχεδίων προπόνησης με βάση τους ατομικούς στόχους και τις φυσικές συνθήκες.

Σήμερα, η αποκατάσταση μέσω εξατομικευμένων προγραμμάτων άσκησης, ιδίως μετά από τραυματισμό, έχει γίνει ένας κρίσιμος τομέας όπου η TN μπορεί να προσφέρει σημαντικές βελτιώσεις. Η παρούσα εργασία βασίζεται σε αυτή την τάση, εξερευνώντας τις δυνατότητες μίας lightweight προσέγγισης για την διευκόλυνση της εξυπηρέτησης διαφόρων αναγκών που μπορεί να προκύπτουν. Ακολουθώντας αυτές τις τάσεις, το μέλλον της τεχνολογίας γυμναστικής υπόσχεται μια πιο επικεντρωμένη στην υγεία προσέγγιση για την επίτευξη των στόχων σε θέματα γυμναστικής και ευεξίας. Η άνοδος αυτών των εφαρμογών αντιπροσωπεύει μια σημαντική αλλαγή στον τρόπο με τον οποίο οι άνθρωποι προσεγγίζουν τη φυσική κατάσταση. Ανεξάρτητα από το υπόβαθρο του καθενός, αυτές οι εξελίξεις θα ενισχύσουν την προσβασιμότητα και την απόλαυση της διατήρησης ενός υγιούς και ενεργού τρόπου ζωής.



Εικόνα 1.1: Παράδειγμα ενός fitness app σε συνεργασία με smartwatch (<https://www.strong.app>)

### 1.3 Καινοτομία

Η καινοτομία αυτού του project έγκειται στην ανάπτυξη ενός fitness app με τον συνδυασμό μηχανικής μάθησης και αποκατάστασης με βάση τον τραυματισμό. Ενώ υπάρχουν πολυάριθμες εφαρμογές γυμναστικής, λίγες είναι προσαρμοσμένες στην αποκατάσταση συγκεκριμένων τραυματισμών και ακόμη λιγότερες χρησιμοποιούν νευρωνικά δίκτυα για την εξατομίκευση της εμπειρίας του χρήστη. Αυτό το έργο διακρίνεται από το γεγονός ότι δεν εστιάζει μόνο σε γενικούς στόχους γυμναστικής, αλλά και στις συγκεκριμένες ανάγκες των χρηστών που βρίσκονται στα στάδια της ανάρρωσης.

Τα παραδοσιακά προγράμματα αποκατάστασης είναι συχνά στατικά και γενικευμένα, σχεδιασμένα με βάση κλινικές κατευθυντήριες γραμμές για ευρείες ομάδες ασθενών. Αντίθετα, η συγκεκριμένη εφαρμογή συλλέγει δεδομένα από την ανατροφοδότηση και τη σωματική απόδοση του χρήστη για να προσαρμόζει τη δυσκολία και τον τύπο των συνιστώμενων ασκήσεων. Αυτή η δυναμική προσέγγιση βοηθά να διασφαλιστεί ότι ο χρήστης θα προσαρμόσει τη δυσκολία που επιθυμεί με βάση τη δική του εμπειρία με τα προτεινόμενα προγράμματα.

Επομένως, το έργο αυτό συμβάλλει σε ένα αυξανόμενο σύνολο εφαρμογών που επιδιώκουν να ενσωματώσουν την τεχνητή νοημοσύνη στην υγεία και τη φυσική κατάσταση με ουσιαστικό τρόπο. Προσφέρει μια πρακτική και επικεντρωμένη στον χρήστη εφαρμογή για τη βελτίωση των αποτελεσμάτων της αποκατάστασης για συγκεκριμένους τραυματισμούς.

### 1.4 Μεθοδολογία

Η ανάπτυξη αυτής της εφαρμογής ακολουθεί μια δομημένη προσέγγιση που συνδυάζει τεχνολογίες frontend και backend. Η βασική τεχνολογική στοίβα αποτελείται από **React Native** για το frontend, **Django** για το backend και **TensorFlow** για το μοντέλο μηχανικής μάθησης.

Το **React Native** είναι ένα ευρέως διαδεδομένο πλαίσιο για την κατασκευή εφαρμογών για κινητά, ιδιαίτερα κατάλληλο για το συγκεκριμένο έργο λόγω των cross-platform δυνατοτήτων του που βασίζεται στην JavaScript. Χρησιμοποιείται από διάφορες εταιρείες τεχνολογίας όπως οι Netflix, Facebook, Instagram, Skype, Facebook Ads και Tesla. Αυτό επιτρέπει την ανάπτυξη μιας ενιαίας βάσης κώδικα που μπορεί να αναπτυχθεί τόσο σε iOS όσο και σε Android, μεγιστοποιώντας την προσβασιμότητα για τους χρήστες ανεξάρτητα από τη συσκευή τους. Το React Native παρέχει επίσης ένα πλούσιο οικοσύστημα βιβλιοθηκών που βελτιώνει τη δημιουργία

μιας φιλικής προς το χρήστη και ευέλικτης διεπαφής. Επιπλέον, ξεχωρίζει στην ανάπτυξη εφαρμογών για κινητά λόγω της ικανότητάς του να αξιοποιεί τις native δυνατότητες των συσκευών Android και iOS, όπως η κάμερα, το GPS και ο τοπικός αποθηκευτικός χώρος. Ένα άλλο πλεονέκτημά του είναι το ζωντανό οικοσύστημα της κοινότητάς του. Το React Native επωφελείται από μια μεγάλη, ενεργή κοινότητα που συνεισφέρει συνεχώς σε μια εκτεταμένη βιβλιοθήκη από open-source packages, καθιστώντας ευκολότερη την υλοποίηση χαρακτηριστικών χωρίς την κατασκευή τους από το μηδέν.

Για το backend, χρησιμοποιείται το **Django**, ένα ισχυρό και κλιμακούμενο web framework. Γραμμένο σε Python, το Django προσφέρει ένα high-level framework που ενθαρρύνει την ταχεία ανάπτυξη και τον καθαρό σχεδιασμό. Επιλέχθηκε για αυτό το έργο λόγω της ευκολίας ενσωμάτωσης με μοντέλα μηχανικής μάθησης, των χαρακτηριστικών user authentication και της ικανότητάς του να χειρίζεται τις πολύπλοκες λειτουργίες βάσης δεδομένων που απαιτούνται για την αποθήκευση πληροφοριών χρηστών, δεδομένων άσκησης και πλάνων προπόνησης. Επιπλέον, το πλαίσιο REST του Django (DRF) υποστηρίζει την ανάπτυξη APIs (Application Programming Interface) που επιτρέπουν την απρόσκοπτη επικοινωνία μεταξύ του frontend του κινητού και του backend. Το Django ακολουθεί την αρχιτεκτονική model-template-view (MTV) και είναι ιδανικό για τον ταχύ κύκλο ζωής της ανάπτυξης λογισμικού, καθώς ακολουθεί την προσέγγιση DRY (Don't Repeat Yourself). Χρησιμοποιείται από οργανισμούς όπως οι Disqus, YouTube, Instagram, Spotify και πολλοί άλλοι. Το admin interface του Django είναι ένα από τα χαρακτηριστικά που ξεχωρίζουν, προσφέροντας μια έτοιμη προς χρήση πλατφόρμα για τη διαχείριση των δεδομένων της εφαρμογής. Χωρίς να χρειάζεται να γράψουν πρόσθετο κώδικα, οι προγραμματιστές μπορούν να έχουν πρόσβαση σε μία πλήρως εξοπλισμένη οθόνη για να προβάλλουν, να δημιουργούν, να ενημερώνουν και να διαγράφουν εγγραφές στη βάση δεδομένων. Αυτή η διεπαφή είναι εξαιρετικά προσαρμόσιμη, επιτρέποντας στους προγραμματιστές να ρυθμίζουν ποια μοντέλα και πεδία είναι ορατά και πώς παρουσιάζονται.

Το **TensorFlow** είναι ένα open-source framework μηχανικής μάθησης που αναπτύχθηκε από την Google και έχει σχεδιαστεί τη διευκόλυνση της δημιουργίας, εκπαίδευσης και ανάπτυξης μοντέλων μηχανικής μάθησης. Παρέχει ισχυρά εργαλεία για την εργασία με νευρωνικά δίκτυα, επιτρέποντας τη βελτιστοποίηση μοντέλων που μπορούν να επιλύουν σύνθετα προβλήματα, όπως η αναγνώριση εικόνων, η επεξεργασία φυσικής γλώσσας και, στην προκειμένη περίπτωση, οι εξατομικευμένες συστάσεις γυμναστικής. Το TensorFlow είναι επίσης εξαιρετικά προσαρμόσιμο, προσφέροντας API για διάφορες γλώσσες προγραμματισμού, όπως η Python, η C++ και η JavaScript, καθιστώντας το μια ευέλικτη επιλογή για ανάπτυξη πολλαπλών πλατφορμών. Σε αυτό το έργο, το TensorFlow Keras - ένα high-level API του TensorFlow - διευκολύνει την ανάπτυξη του τεχνητού νευρωνικού δικτύου (ANN) που τροφοδοτεί τις συστάσεις ασκήσεων στην



εφαρμογή. Το Keras έχει σχεδιαστεί για γρήγορο πειραματισμό, καθιστώντας το ιδανικό για προγραμματιστές και ερευνητές. Το ANN σε αυτό το έργο κατασκευάζεται χρησιμοποιώντας το Sequential API του Keras, το οποίο είναι κατάλληλο για τη δημιουργία μοντέλων στρώμα προς στρώμα. Αυτή η προσέγγιση επιτρέπει την απλή διαμόρφωση στρωμάτων όπως τα πλήρως συνδεδεμένα dense layers και τα dropout layers, που συνεισφέρει στη μείωση του overfitting και τη βελτίωση της γενίκευσης του μοντέλου. Τέλος, υποστηρίζει εργαλεία για την προεπεξεργασία (preprocessing) δεδομένων και την αξιολόγηση μοντέλων. Χαρακτηριστικά όπως η ηλικία, το BMI και οι τύποι τραυματισμών τυποποιούνται και κωδικοποιούνται με τη χρήση των βοηθητικών προγραμμάτων προεπεξεργασίας του TensorFlow πριν τροφοδοτηθούν στο νευρωνικό δίκτυο.

## Κεφάλαιο 2: Αποκατάσταση τραυματισμών και συλλογή δεδομένων για νευρωνικά δίκτυα

### 2.1 Επισκόπηση των τραυματισμών

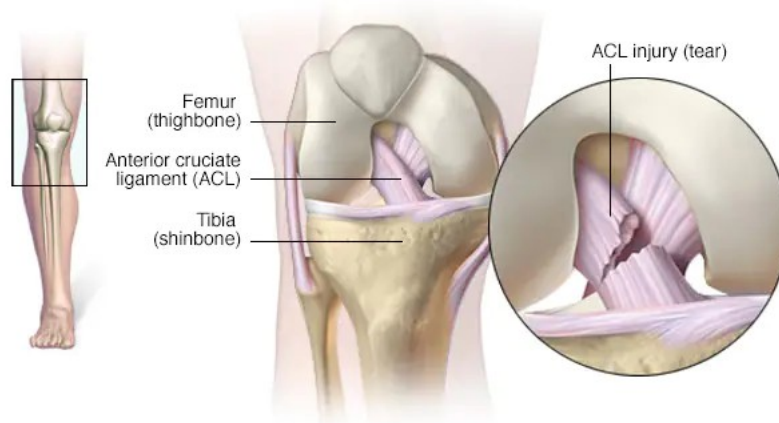
Οι τρεις τραυματισμοί για τους οποίους η εφαρμογή εξυπηρετεί την αποκατάστασή τους είναι η ζημιά/ρήξη **πρόσθιου χιαστού συνδέσμου (ACL)**, **μηνίσκου (Meniscus)** και **αχιλλείου τένοντα (Achilles Tendon)**. Αυτοί οι τραυματισμοί δεν είναι κοινοί μόνο στους αθλητές αλλά και στον γενικό πληθυσμό, με τα πρωτόκολλα αποκατάστασης να ποικίλλουν ανάλογα με τη σοβαρότητα του τραυματισμού, τα ατομικά επίπεδα φυσικής κατάστασης και τους ρυθμούς προόδου. Η παραδοσιακή αποκατάσταση περιλαμβάνει φάσεις ξεκούρασης, σταθεροποίησης του ποδιού, προοδευτικής προπόνησης δύναμης, ασκήσεων ευλυγισίας και νευρομυϊκής επανεκπαίδευσης για την αποκατάσταση της κίνησης και την πρόληψη επανατραυματισμού.

Η εφαρμογή έχει σχεδιαστεί για να βοηθήσει τους χρήστες στη διαδικασία αποκατάστασης των τραυματισμών τους, αλλά δεν προορίζεται να αντικαταστήσει τις επαγγελματικές ιατρικές συμβουλές. Πριν από τη χρήση της εφαρμογής, οι χρήστες θα πρέπει να συμβουλευούνται το γιατρό τους για να λάβουν εξατομικευμένη διάγνωση, ιδίως στα πρώτα στάδια της αποκατάστασης. Είναι απαραίτητο οι χρήστες να ανακτήσουν πρώτα τη βασική κινητικότητα στο τραυματισμένο πόδι τους υπό ιατρική επίβλεψη πριν ξεκινήσουν τις ασκήσεις που προτείνει η εφαρμογή. Αυτό διασφαλίζει ότι είναι σωματικά προετοιμασμένοι για πιο προχωρημένες κινήσεις, μειώνοντας τον κίνδυνο περαιτέρω τραυματισμού ή επιπλοκών.

- Ο τραυματισμός του **πρόσθιου χιαστού συνδέσμου** περιλαμβάνει ρήξη ή διάστρεμμα του πρόσθιου χιαστού συνδέσμου - μιας από τις ισχυρές ζώνες ιστού που βοηθούν στη σύνδεση του μηριαίου οστού με το οστό της κνήμης. Οι τραυματισμοί του χιαστού συνδέσμου συμβαίνουν συχνότερα κατά τη διάρκεια αθλημάτων που περιλαμβάνουν απότομες στάσεις ή αλλαγές κατεύθυνσης, άλματα και προσγειώσεις - όπως το ποδόσφαιρο, το μπάσκετ και το σκι. Πολλοί άνθρωποι αισθάνονται ένα «σκάσιμο» στο γόνατο όταν υπάρχει τραυματισμός του πρόσθιου χιαστού. Το γόνατό μπορεί να πρηστεί, να αισθάνεται ασταθές και να είναι πολύ επώδυνο για να σηκώσει βάρος. Η αποκατάσταση

περιλαμβάνει ασκήσεις εύρους κίνησης, προοδευτική προπόνηση δύναμης για τους τετρακέφαλους και τους οπίσθιους μηριαίους και προπόνηση ισορροπίας. Η πλήρης αποκατάσταση μπορεί να διαρκέσει 6-12 μήνες, ανάλογα με την τήρηση του πρωτοκόλλου αποκατάστασης και την ατομική πρόοδο.

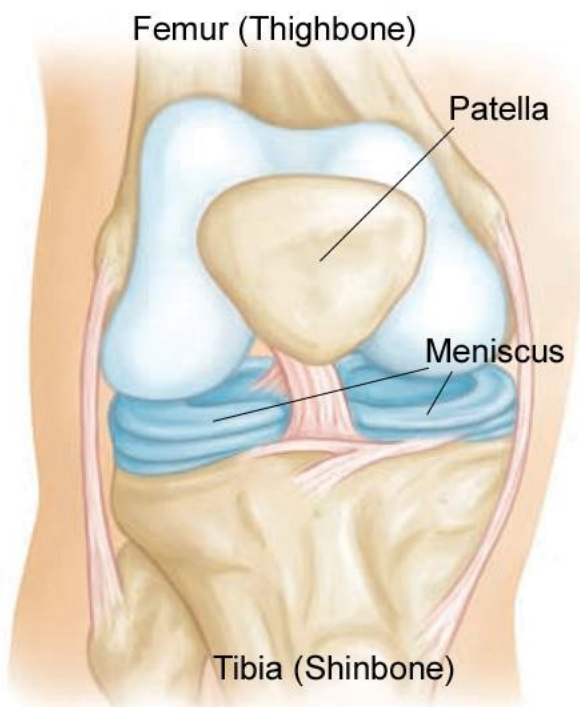
- Οι **μηνίσκοι** βρίσκονται μεταξύ της κνήμης και του μηριαίου οστού και προστατεύουν το κάτω μέρος του ποδιού από τους κραδασμούς που προκαλούνται από το βάρος του σώματός μας. Οι ρήξεις μηνίσκων συνήθως λαμβάνουν χώρα όταν ένας αθλητής στρίβει ή στρέφει το άνω μέρος του ποδιού του ενώ το πόδι του είναι πατημένο και το γόνατό του λυγισμένο. Περιστασιακά οι μηνίσκοι μπορεί να αναπτυχθούν σε σχήμα μπλοκ ή δίσκου, το οποίο ονομάζεται δισκοειδής μηνίσκος. Ένας δισκοειδής μηνίσκος είναι πιο πιθανό να σχιστεί και συνήθως παρουσιάζεται στην παιδική ηλικία. Τα συμπτώματα της ρήξης μηνίσκου μπορεί να είναι διαφορετικά για κάθε άτομο, αλλά μερικά από τα πιο συνηθισμένα συμπτώματα περιλαμβάνουν πόνο στην άρθρωση του γόνατος (συνήθως στο εσωτερικό (έσω), στο εξωτερικό (πλάγιο) ή στο πίσω μέρος του γόνατος), πρήξιμο, κλείδωμα της άρθρωσης του γόνατος και αδυναμία πλήρους έκτασης ή κάμψης της άρθρωσης του γόνατος. Η θεραπεία μπορεί να περιλαμβάνει παγοποίηση της περιοχής, φαρμακευτική αγωγή όπως ιβουπροφαίνη, ασκήσεις μυϊκής ενδυνάμωσης και αρθροσκοπική χειρουργική επέμβαση. Όπως και με τον πρόσθιο χιαστό, η ενδυνάμωση των τετρακεφάλων και των οπίσθιων μηριαίων, αλλά και των μυών των ισχύων είναι σημαντική για την αποκατάσταση.
- Ο **αχίλλειος τένοντας** είναι μια ινώδης ζώνη ιστού που συνδέει τους μύες της γάμπας με τη φτέρνα. Η δύναμη και η ευελιξία αυτού του τένοντα είναι σημαντικές για τα άλματα, το τρέξιμο και το περπάτημα. Ο αχίλλειος τένοντας σας δέχεται μεγάλη πίεση και καταπόνηση κατά τη διάρκεια των καθημερινών δραστηριοτήτων, καθώς και κατά τη διάρκεια αθλητικών και ψυχαγωγικών παιχνιδιών. Οι δύο κύριοι τραυματισμοί αχίλλειου τένοντα είναι η τενοντίτιδα και η ρήξη. Η τενοντίτιδα μπορεί να οφείλεται σε υπερβολική χρήση ή σε βλάβη της περιοχής. Μπορεί να προκαλέσει πόνο στο πίσω μέρος του ποδιού σας και γύρω από τη φτέρνα, με τμήματα του τένοντά να γίνονται πιο παχιά και να σκληραίνουν. Οι ρήξεις στις ίνες του τένοντα μπορεί να προκαλέσουν πλήρη ή μερική ρήξη του τένοντα. Οι τραυματισμοί του αχίλλειου τένοντα αποτελούν πρόκληση λόγω της αργής επούλωσής του. Η αποκατάσταση περιλαμβάνει ασκήσεις έκκεντρης ενδυνάμωσης της γάμπας, ασκήσεις ισορροπίας και, τελικά, αθλητικές δραστηριότητες για την πρόληψη περαιτέρω βλάβης και τη διευκόλυνση της επιστροφής στη σωματική δραστηριότητα.



© MAYO FOUNDATION FOR MEDICAL EDUCATION AND RESEARCH. ALL RIGHTS RESERVED.

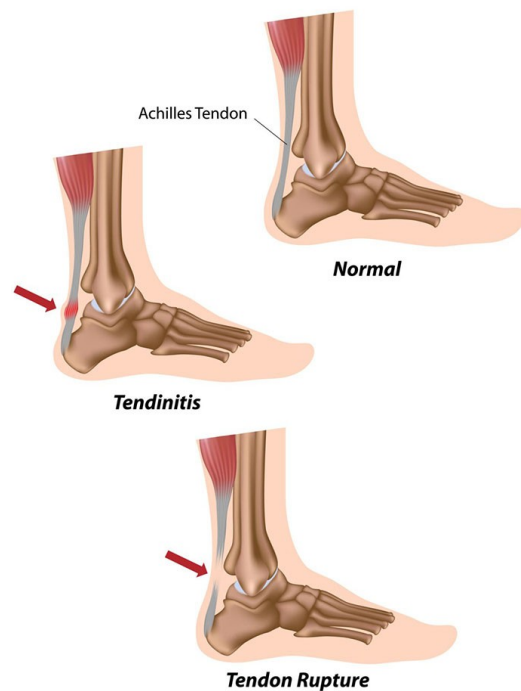
**Εικόνα 2.1:** Ανατομία πρόσθιου χιαστού συνδέσμου και τραυματισμός

(<https://www.mayoclinic.org/diseases-conditions/acl-injury/symptoms-causes/syc-20350738>)



**Εικόνα 2.2:** Ανατομία μηνίσκου

(<https://orthoinfo.aaos.org/en/diseases--conditions/meniscus-tears/>)



**Εικόνα 2.3:** Ανατομία αχίλλειου τένοντα και τραυματισμοί

(<https://www.alilamedicalmedia.com>)

Τραυματισμοί όπως οι παραπάνω, ειδικά σε περίπτωση ρήξης, είναι σύνηθες να απαιτούν χειρουργική επέμβαση για πλήρη αποκατάσταση, ιδίως στις μικρότερες ηλικιακές ομάδες. Μετά τη χειρουργική επέμβαση, οι ασθενείς συνήθως περνούν μια περίοδο ακινητοποίησης και

φυσικοθεραπείας υπό την καθοδήγηση επαγγελματιών υγείας. Κατά τη διάρκεια αυτού του κρίσιμου αρχικού σταδίου, το σώμα πρέπει να επουλωθεί σωστά πριν από την άσκηση οποιασδήποτε έντονης δραστηριότητας. Η εφαρμογή δεν προορίζεται για χρήση κατά τη διάρκεια αυτής της αρχικής φάσης ανάρρωσης. Ωστόσο, μόλις περάσει η πρώιμη μετεγχειρητική φάση ή σε περιπτώσεις όπου δεν απαιτήθηκε χειρουργική επέμβαση αλλά υπάρχει βλάβη, η εφαρμογή μπορεί να αποτελέσει χρήσιμο εργαλείο για την καθοδήγηση των χρηστών. Βοηθά τους χρήστες να ανακτήσουν τη δύναμη, την κινητικότητα και τη λειτουργικότητα στην τραυματισμένη περιοχή, ενώ διασφαλίζει ότι οι ασκήσεις είναι εξατομικευμένες ανάλογα με το επίπεδο φυσικής κατάστασης, τον τύπο τραυματισμού και την πρόοδό τους. Οι χρήστες θα πρέπει πάντα να συμβουλευονται τους γιατρούς τους πριν χρησιμοποιήσουν την εφαρμογή για να διασφαλίσουν ότι είναι έτοιμοι για τις ασκήσεις αποκατάστασης που παρέχονται.

## 2.2 Συλλογή και κατηγοριοποίηση δεδομένων

Προκειμένου να δημιουργηθεί μια λειτουργική εφαρμογή αποκατάστασης τραυματισμών, η διαδικασία συλλογής δεδομένων και η οργάνωσή τους κρίνεται απαραίτητη. Η εφαρμογή οφείλει να αντλεί πληροφορίες από αξιόπιστες πηγές, όπως ιατρικά περιοδικά και οδηγίες φυσιοθεραπείας, αλλά και να δομεί τα δεδομένα με τρόπο που να είναι εύκολα αφομοιώσιμα τόσο για τους χρήστες όσο και για το Τεχνητό Νευρωνικό Δίκτυο (ANN) που τροφοδοτεί τις συστάσεις.

Τα δεδομένα που συλλέγονται σχετικά με τις ασκήσεις αφορούν την καταλληλότητά τους για διαφορετικές περιπτώσεις χρηστών. Μία άσκηση μπορεί να εξυπηρετεί πολλές διαφορετικές περιπτώσεις τραυματισμού, καθιστώντας την απαραίτητη για όλους τους χρήστες, ενώ άλλες ασκήσεις μπορεί να ταιριάζουν μόνο σε πολύ ειδικές καταστάσεις όσων αφορά τον τραυματισμό, την ηλικία, τα φυσικά χαρακτηριστικά και το αθλητικό επίπεδο. Για τους τραυματισμούς με τους οποίους ασχολούμαστε τα δεδομένα αντλούνται από:

- **Άρθρα ιατρικής έρευνας**, τα οποία παρέχουν πληροφορίες σχετικά με τα πρωτόκολλα αποκατάστασης για κάθε τύπο τραυματισμού.
- **Κατευθυντήριες γραμμές φυσιοθεραπείας**, οι οποίες χρησιμεύουν ως βάση για την ανάπτυξη ασφαλών και αποτελεσματικών ρουτινών άσκησης κατά τη διάρκεια διαφορετικών σταδίων αποκατάστασης.
- **Διαδικτυακές κοινότητες γυμναστικής**, που περιλαμβάνουν εμπειρίες πραγματικού κόσμου και σχόλια που λαμβάνονται επίσης υπόψη για τον εμπλουτισμό του περιεχομένου της εφαρμογής με ποικίλες προσεγγίσεις αποκατάστασης.

Οι ασκήσεις που συλλέγονται από αυτές τις πηγές κατηγοριοποιούνται με βάση τους εξής βασικούς παράγοντες:

- **Τραυματισμός:** Ο βασικότερος παράγοντας, δημιουργούνται διαφορετικά σετ ασκήσεων, τα οποία επικεντρώνονται σε συγκεκριμένους μύες ή μέρη του σώματος.
- **Επίπεδο φυσικής κατάστασης:** Κάθε άσκηση είναι χαρακτηρισμένη ως κατάλληλη για τα πρώιμα, μεσαία ή αργότερα στάδια της ανάρρωσης, διασφαλίζοντας ότι οι χρήστες εκτελούν μόνο κινήσεις κατάλληλες για τη διαδικασία επούλωσής τους. Το αθλητικό επίπεδο του χρήστη επίσης παίζει ρόλο στην καταλληλότητα στις ασκήσεις.
- **Ηλικία:** Χρήστες προχωρημένων ηλικιών έχουν περιορισμούς στις ασκήσεις που μπορούν να κάνουν, καθώς ο ρυθμός αποκατάστασής τους είναι πιο αργός με αποτέλεσμα ορισμένες ασκήσεις να επιβαρύνουν την κατάστασή τους παρά να την βοηθούν.
- **Δείκτης μάζας σώματος (BMI):** Συνιστώνται διαφορετικές ασκήσεις με βάση το δείκτη μάζας σώματός τους, ώστε να λαμβάνονται υπόψη οι πιθανοί περιορισμοί των αρθρώσεων ή των κινήσεων.
- **Απαραίτητος εξοπλισμός:** Οι ασκήσεις κατηγοριοποιούνται περαιτέρω με βάση το αν απαιτείται εξοπλισμός όπως λάστιχα γυμναστικής ή εξοπλισμός γυμναστηρίου, διασφαλίζοντας ότι οι χρήστες έχουν τους απαραίτητους πόρους για τις ασκήσεις που τους ανατίθενται.

Όλα αυτά τα δεδομένα οργανώνονται σε ένα δομημένο φύλλο Excel, το οποίο χρησιμεύει ως κεντρική βάση δεδομένων τόσο για τη διεπαφή της εφαρμογής όσο και για το backend. Κάθε γραμμή αντιστοιχεί σε μια συγκεκριμένη άσκηση, ενώ οι στήλες αντιπροσωπεύουν τα διάφορα χαρακτηριστικά που αναφέρθηκαν προηγουμένως. Επιπλέον, περιλαμβάνονται και περαιτέρω πληροφορίες για την άσκηση με σκοπό την κατανόησή της από τον χρήστη, όπως ο τύπος άσκησης (Δύναμη/Ευλυγισία), οι στοχευμένοι μύες και η περιγραφή της σωστής εκτέλεσής της.

## Διπλωματική Εργασία – Νικόλαος Καλόφωνος

| ID | Name                                   | Difficulty   | Age      | Injury                                 | Muscle Group            | Type        | Equipment       | Description   | BMI Range |
|----|--|--------------|----------|--|-------------------------|-------------|-----------------|---|-----------|
| 2  | Straight Leg Raises                    | Beginner     | All      | Torn ACL, Torn Meniscus, Torn Achilles | Quadriceps, Hip Flexors | Strength    | None            | Lie on your back with both legs straight. Lift one leg slowly while keeping it straight, targeting the quadriceps and hip flexors. Perform 10-15 reps, 3 sets.  | All       |
| 16 | Seated Hamstring Curls                 | Intermediate | Under 60 | Torn Meniscus                          | Hamstrings, Glutes      | Strength    | Gym             | Sit on the machine with your back against the pad and adjust the thigh support so your knees are aligned with the pivot point. Curl your legs down towards your glutes by contracting your hamstrings, then slowly return to the starting position. Perform 10-15 reps, 3 sets. | < 35      |
| 23 | Lunges                                 | Advanced     | Under 50 | Torn ACL                               | Quadriceps, Glutes      | Strength    | None            | Stand with feet together. Step forward deeply into a lunge to strengthen the quadriceps and glutes. Use dumbbells if comfortable. Perform 10-12 reps per leg, 3 sets.   | < 30      |
| 50 | Reclining Hand-to-Big-Toe Stretch      | Intermediate | All      | Torn ACL, Torn Meniscus                | Hamstrings              | Flexibility | Resistance Band | Lie on your back, extend one leg towards the ceiling, and hold your big toe with your hand or a strap to stretch the hamstrings and lower back. Perform 20-30 seconds, 2-3 sets per side.   | < 35      |
| 61 | Seated Resistance Band Ankle Inversion | Intermediate | All      | Torn Achilles                          | Calves                  | Strength    | Resistance Band | Sit with legs extended, place a resistance band around the foot, and pull the foot inward, towards the other foot against resistance. Perform 10-15 reps, 3 sets per leg.   | All       |

**Σχήμα 2.1:** Παραδείγματα καταχωρημένων ασκήσεων

Αυτό το αρχείο εξυπηρετεί τους παρακάτω σκοπούς:

- **Backend Management:** Οι ασκήσεις αποθηκεύονται σε ένα database στο οποίο η εφαρμογή ανατρέχει ώστε να εμφανίσει τις κατάλληλες πληροφορίες στον χρήστη.
- **ANN Training:** Η δομημένη μορφή είναι ιδανική για την τροφοδότηση του ANN. Το νευρωνικό δίκτυο χρησιμοποιεί αυτά τα δεδομένα ως χαρακτηριστικά εισόδου, μαθαίνοντας να συσχετίζει τους τύπους τραυματισμών, τα επίπεδα φυσικής κατάστασης και άλλους παράγοντες με τις συστάσεις ασκήσεων.

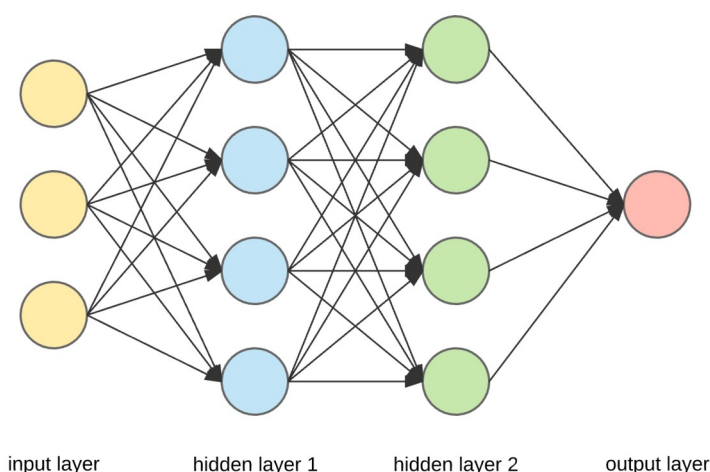
Διατηρώντας αυτή τη δομημένη προσέγγιση στη συλλογή και κατηγοριοποίηση δεδομένων, η εφαρμογή μπορεί να παρουσιάσει αυτά τα δεδομένα με έναν φιλικό και κατανοητό τρόπο προς τον χρήστη, αλλά επίσης μπορεί να βελτιώνεται συνεχώς, αφομοιώνοντας καινούργιες καταχωρήσεις ασκήσεων άμεσα και τροφοδοτώντας τες στο ANN.

## Κεφάλαιο 3: Νευρωνικά Δίκτυα για Εξατομικευμένη Αποκατάσταση Τραυματισμού

### 3.1 Εισαγωγή στα Νευρωνικά Δίκτυα

Τα νευρωνικά δίκτυα είναι ένα υποσύνολο της τεχνητής νοημοσύνης που μοντελοποιεί την ικανότητα του ανθρώπινου εγκεφάλου να επεξεργάζεται δεδομένα και να αναγνωρίζει μοτίβα. Αποτελούμενα από διασυνδεδεμένους νευρώνες (neurons), τα συστήματα αυτά χρησιμεύουν στην εύρεση υποκείμενων δομών στα δεδομένα, καθιστώντας τα ιδανικά για εργασίες όπως η ταξινόμηση, η πρόβλεψη και η αναγνώριση προτύπων. Τα νευρωνικά δίκτυα χρησιμοποιούνται όλο και περισσότερο σε όλους τους κλάδους, συμπεριλαμβανομένης της υγειονομικής περίθαλψης, για την ικανότητά τους να χειρίζονται πολύπλοκες, μη γραμμικές σχέσεις μεταξύ εισόδων και εξόδων.

Η βασική έννοια ενός νευρωνικού δικτύου περιστρέφεται γύρω από στρώματα νευρώνων: το στρώμα εισόδου (**input layer**) όπου τα δεδομένα τροφοδοτούνται στο δίκτυο, ένα ή περισσότερα κρυφά στρώματα (**hidden layers**) όπου γίνεται η επεξεργασία και η αναγνώριση προτύπων και το στρώμα εξόδου (**output layer**) το οποίο δίνει το αποτέλεσμα, όπως μια ταξινόμηση ή πρόβλεψη. Κάθε νευρώνας επεξεργάζεται τις εισόδους χρησιμοποιώντας μια μαθηματική συνάρτηση και μεταβιβάζει το αποτέλεσμα στο επόμενο στρώμα.

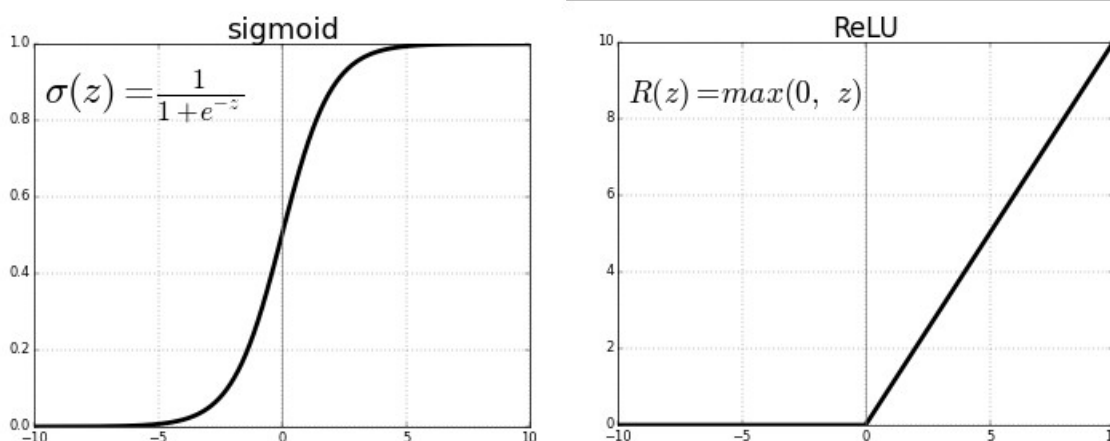


**Εικόνα 3.1:** Αναπαράσταση των layers ενός νευρωνικού δικτύου

(<https://blog.stevengong.co/how-does-a-neural-network-work-intuitively-in-code-f51f7b2c1e3f>)



Ο πιο συνηθισμένος τύπος νευρωνικού δικτύου που χρησιμοποιείται σε εφαρμογές όπως η δική μας είναι το feedforward neural network. Σε αυτόν τον τύπο, τα δεδομένα ρέουν προς μία κατεύθυνση, από την είσοδο στην έξοδο, χωρίς επιστροφή. Τα κρυφά στρώματα εφαρμόζουν συναρτήσεις ενεργοποίησης. Μια συνάρτηση ενεργοποίησης (**activation function**) αποφασίζει αν ένας νευρώνας πρέπει να ενεργοποιηθεί ή όχι. Αυτό σημαίνει ότι θα αποφασίσει αν η είσοδος του νευρώνα στο δίκτυο είναι σημαντική ή όχι στη διαδικασία της πρόβλεψης χρησιμοποιώντας απλούστερες μαθηματικές πράξεις. Η **ReLU** (Rectified Linear Unit) είναι μία χαρακτηριστική συνάρτηση ενεργοποίησης, η οποία εισάγει μη γραμμικότητα στο μοντέλο, επιτρέποντας στο δίκτυο να αποτυπώσει πολύπλοκες σχέσεις μεταξύ των χαρακτηριστικών εισόδου. Έχει γίνει η προεπιλεγμένη συνάρτηση ενεργοποίησης για πολλούς τύπους νευρωνικών δικτύων, επειδή ένα μοντέλο που τη χρησιμοποιεί είναι ευκολότερο στην εκπαίδευση και συχνά επιτυγχάνει καλύτερες επιδόσεις. Μια άλλη κοινή συνάρτηση ενεργοποίησης είναι η **Sigmoid**. Ο κύριος λόγος για τον οποίο χρησιμοποιούμε τη Sigmoid συνάρτηση είναι επειδή κυμαίνεται μεταξύ (0 έως 1). Επομένως, χρησιμοποιείται ιδιαίτερα για μοντέλα όπου πρέπει να προβλέπουν την πιθανότητα ως έξοδο.



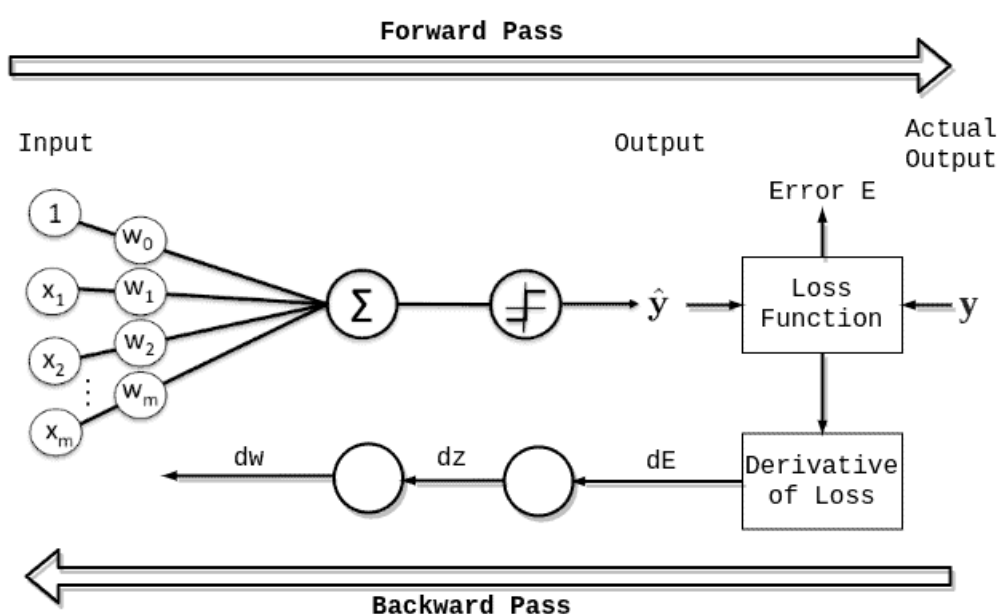
**Εικόνα 3.2:** Γραφική αναπαράσταση των συναρτήσεων ενεργοποίησης

(<https://medium.com/@srivastavashivansh8922/understanding-the-difference-between-relu-and-sigmoid-activation-functions-in-deep-learning-33b280fc2071>)

Τα νευρωνικά δίκτυα μαθαίνουν μέσω μιας διαδικασίας που ονομάζεται εκπαίδευση (**training**), κατά την οποία τροφοδοτούνται με δεδομένα και προσαρμόζουν τις εσωτερικές τους παραμέτρους, ή βάρη (**weights**), με βάση το πόσο καλά προβλέπουν το αποτέλεσμα. Αυτή η διαδικασία εκμάθησης διέπεται από έναν αλγόριθμο βελτιστοποίησης όπως το **backpropagation**, ο οποίος βοηθά το μοντέλο να ελαχιστοποιήσει το σφάλμα με τη λεπτομερή ρύθμιση των βαρών του. Ο backpropagation είναι ο πιο συνηθισμένος αλγόριθμος εκπαίδευσης για τα νευρωνικά δίκτυα. Κάνει την κάθοδο κλίσης εφικτή για νευρωνικά δίκτυα πολλαπλών επιπέδων. Πολλές βιβλιοθήκες κώδικα μηχανικής μάθησης (όπως το **Keras**) χειρίζονται την οπισθοδιάδοση αυτόματα.

Ένα **loss function** είναι μια συνάρτηση που συγκρίνει τις τιμές στόχου και τις προβλεπόμενες τιμές εξόδου μετρώντας το πόσο καλά το νευρωνικό δίκτυο μοντελοποιεί τα δεδομένα εκπαίδευσης. Κατά την εκπαίδευση, στοχεύουμε στην ελαχιστοποίηση αυτής της απώλειας μεταξύ των προβλεπόμενων και των στόχων εξόδου.

Η εκπαίδευση συνήθως περιλαμβάνει την επανάληψη ενός συνόλου δεδομένων πολλές φορές, ή εποχές (**epochs**), μέχρι το δίκτυο να επιτύχει ικανοποιητική ακρίβεια. Μια εποχή σημαίνει εκπαίδευση του νευρωνικού δικτύου με όλα τα δεδομένα εκπαίδευσης για έναν κύκλο. Σε ένα epoch χρησιμοποιούνται όλα τα δεδομένα ακριβώς μία φορά. Ένα forward pass και ένα backward pass μαζί υπολογίζονται ως ένα πέρασμα.



Εικόνα 3.3: Η διαδικασία ενός epoch

(<https://www.baeldung.com/cs/epoch-neural-networks>)

## 3.2 Δημιουργία και Εκπαίδευση του ANN για την εφαρμογή

Αυτό το κεφάλαιο περιγράφει λεπτομερώς τη διαδικασία σχεδιασμού, εκπαίδευσης και χρήσης του νευρωνικού δικτύου που χρησιμοποιήθηκε για την παραγωγή των πλάνων γυμναστικής. Θα αναλυθούν τα απαραίτητα αρχεία, οι συναρτήσεις, οι βιβλιοθήκες και οι προγραμματιστικές τεχνικές που χρησιμοποιήθηκαν ώστε να έρθει εις πέρας αυτή η διαδικασία.

### 3.2.1 Συλλογή δεδομένων και αξιολογηση καταλληλότητας

Έχοντας διαθέσιμο το dataset των ασκήσεων που αναφέρθηκε στο κεφάλαιο 2.2, το πρώτο βήμα για την εκπαίδευση του μοντέλου ήταν η δημιουργία ενός αρχείου excel με όνομα **user\_examples.xlsx** που περιέχει παραδείγματα χρηστών με τις απαραίτητες πληροφορίες (UserID, Injury, Age, Skill, BMI, Equipment). Σκοπός αυτού του αρχείου είναι η διασταύρωσή του με το dataset των ασκήσεων με τέτοιο τρόπο ώστε να εκπονηθεί ένα τρίτο αρχείο το οποίο θα περιλαμβάνει μία σύζευξη των προηγούμενων, προσθέτοντας ένα πεδίο 'Suitable' που παίρνει τιμές 1 (άν η άσκηση είναι κατάλληλη για τον χρήστη) ή 0 (αν η άσκηση δεν είναι κατάλληλη για τον χρήστη). Για τον σκοπό αυτό δημιουργήθηκε ένα αρχείο ονόματι **suitability\_script.py**, το οποίο συγκρίνει τα δεδομένα από το dataset ασκήσεων με τα δεδομένα από το dataset χρηστών και παράγει το αρχείο που αναφέρθηκε παραπάνω.

```
def is_suitable(user, exercise):
    if user['Injury'] not in exercise['Injury']:
        return False

    skill_mapping = {'Beginner': 0, 'Intermediate': 1, 'Advanced': 2}
    if skill_mapping[user['Skill']] < skill_mapping[exercise['Difficulty']]:
        return False

    if exercise['Age'] != 'All':
        age_limit = int(exercise['Age'].replace('Under ', ''))
        if user['Age'] >= age_limit:
            return False

    equipment_hierarchy = {'None': 0, 'Resistance Band': 1, 'Gym': 2}

    user_equipment = user['Equipment'] if pd.notna(user['Equipment']) else 'None'
    exercise_equipment = exercise['Equipment'] if pd.notna(exercise['Equipment']) else 'None'

    user_equipment_level = equipment_hierarchy[user_equipment]
    exercise_equipment_level = equipment_hierarchy[exercise_equipment]

    if user_equipment_level < exercise_equipment_level:
        return False

    if exercise['BMI Range'] != 'All':
        bmi_limit = int(exercise['BMI Range'].replace('< ', ''))
        if user['BMI'] >= bmi_limit:
            return False

    return True
```

**Σχήμα 3.1:** Συνάρτηση is\_suitable του αρχείου suitability\_script.py

Αυτή η συνάρτηση αξιολογεί αν μια άσκηση είναι κατάλληλη για έναν συγκεκριμένο χρήστη αναλύοντας διάφορα κριτήρια. Μόλις ολοκληρωθεί η διαδικασία αντιστοίχισης, δημιουργείται ένας νέος πίνακας που συνοψίζει ποιες ασκήσεις είναι κατάλληλες για ποιους χρήστες. Αυτό το σύνολο δεδομένων καταλληλότητας αποθηκεύεται σε αρχείο excel και αποτελεί τα δεδομένα εκπαίδευσης για το ANN.

| User ID | Skill        | Age | Injury        | Equipment       | BMI | Exercise ID | Suitable |
|---------|--------------|-----|---------------|-----------------|-----|-------------|----------|
| 1       | Intermediate | 52  | Torn ACL      | None            | 23  | 5           | 1        |
| 1       | Intermediate | 52  | Torn ACL      | None            | 23  | 6           | 0        |
| 1       | Intermediate | 52  | Torn ACL      | None            | 23  | 7           | 0        |
| ...     | ...          | ... | ...           | ...             | ... | ...         | ...      |
| 2       | Beginner     | 32  | Torn Meniscus | Resistance Band | 31  | 7           | 1        |
| 2       | Beginner     | 32  | Torn Meniscus | Resistance Band | 31  | 8           | 1        |
| 2       | Beginner     | 32  | Torn Meniscus | Resistance Band | 31  | 9           | 0        |
| ...     | ...          | ... | ...           | ...             | ... | ...         | ...      |
| 27      | Advanced     | 24  | Torn Achilles | Gym             | 24  | 20          | 1        |
| 27      | Advanced     | 24  | Torn Achilles | Gym             | 24  | 21          | 1        |
| 27      | Advanced     | 24  | Torn Achilles | Gym             | 24  | 22          | 1        |
| ...     | ...          | ... | ...           | ...             | ... | ...         | ...      |

Σχήμα 3.2: Αποσπάσματα από το αρχείο εξόδου του suitability\_script.py

### 3.2.2 Αρχιτεκτονική νευρωνικού δικτύου και διαδικασία εκπαίδευσης

Σε αυτή την ενότητα, θα καλυφθεί σε λεπτομέρεια η ανάπτυξη και η εκπαίδευση του ANN, τα οποία πραγματοποιήθηκαν μέσω του αρχείου **train\_ann.py**. Το νευρωνικό δίκτυο σχεδιάστηκε με τη βοήθεια του **TensorFlow**, μίας βιβλιοθήκης μηχανικής μάθησης ανοικτού κώδικα, αξιοποιώντας το **Keras**, ένα high-level API που απλοποιεί τη διαδικασία της εκπαίδευσης.

Πριν από το χτίσιμο και την εκπαίδευση του νευρωνικού δικτύου, είναι απαραίτητη η προετοιμασία των δεδομένων που θα δωθούν ως είσοδος. Οι μεταβλητές εισόδου χωρίστηκαν σε κατηγορικές (αθλητικό επίπεδο, τραυματισμός, εξοπλισμός) και σε αριθμητικές (ηλικία, BMI). Έτσι, η προεπεξεργασία γίνεται ως εξής:

- **Categorical Features:** Οι κατηγορικές μεταβλητές πρέπει να μετατραπούν σε αριθμητική μορφή που να μπορεί να επεξεργαστεί το μοντέλο. Αυτό επιτεύχθηκε με τη χρήση του **One-Hot Encoding**, μιας μεθόδου που μετατρέπει κάθε κατηγορία σε δυαδική διανυσματική αναπαράσταση. Για παράδειγμα, ο τραυματισμός ενός χρήστη θα μπορούσε να αναπαρασταθεί ως διάνυσμα όπου μία θέση είναι «1» (υποδεικνύοντας τον συγκεκριμένο τραυματισμό) και όλες οι άλλες είναι «0». Για παράδειγμα:
  - ACL: [1,0,0]
  - Meniscus: [0,1,0]
  - Achilles: [0,0,1]

- **Numerical Features:** Η ηλικία και το BMI, ως συνεχείς μεταβλητές, τυποποιήθηκαν με τη χρήση του StandardScaler. Η τυποποίηση των αριθμητικών χαρακτηριστικών βελτιώνει την αποτελεσματικότητα της διαδικασίας εκμάθησης διασφαλίζοντας ότι οι τιμές εμπίπτουν σε ένα παρόμοιο εύρος. Σκοπός της τυποποίησης είναι η κλιμάκωση των δεδομένων κατά τρόπο ώστε ο μέσος όρος κάθε χαρακτηριστικού να έχει κέντρο γύρω από το 0, με τυπική απόκλιση 1. Αυτό επιτυγχάνεται με την αφαίρεση του μέσου όρου από κάθε χαρακτηριστικό και τη διαίρεση με την τυπική απόκλιση.

```
categorical_features = ['Skill', 'Injury', 'Equipment', 'Exercise ID']
numerical_features = ['Age', 'BMI']

preprocessor = ColumnTransformer(
    transformers=[
        ('cat', Pipeline(steps=[
            ('imputer', SimpleImputer(strategy='constant',
fill_value='None')),
            ('onehot', OneHotEncoder())
        ]), categorical_features),
        ('num', StandardScaler(), numerical_features)
    ]
)
```

**Σχήμα 3.3:** Η διαδικασία της προεπεξεργασίας

Για να αξιολογηθεί η απόδοση του μοντέλου και να αποφευχθεί το overfitting, το dataset χωρίστηκε σε ένα set εκπαίδευσης και ένα set δοκιμής. Το set εκπαίδευσης χρησιμοποιήθηκε για την εκπαίδευση του νευρωνικού δικτύου, ενώ το set δοκιμής κρατήθηκε πίσω για να αξιολογηθεί πόσο καλά το μοντέλο γενικεύεται σε νέα δεδομένα.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
X_train = preprocessor.fit_transform(X_train)
X_test = preprocessor.transform(X_test)
```

**Σχήμα 3.4:** Train και Testing sets

Στην περίπτωση αυτή, το 20% των δεδομένων προορίζεται για δοκιμή, ενώ το υπόλοιπο 80% χρησιμοποιείται για εκπαίδευση. Το random\_state=42 εξασφαλίζει ότι η διαδικασία διαχωρισμού των δεδομένων είναι ντετερμινιστική και ότι τα αποτελέσματα είναι αναπαραγώγιμα σε διαφορετικές εκτελέσεις ή περιβάλλοντα.

Το δίκτυο σχεδιάστηκε ως ένα πλήρως συνδεδεμένο **feedforward** neural network. Αποτελείται από τρία hidden layers, καθένα από τα οποία χρησιμοποιεί τη συνάρτηση

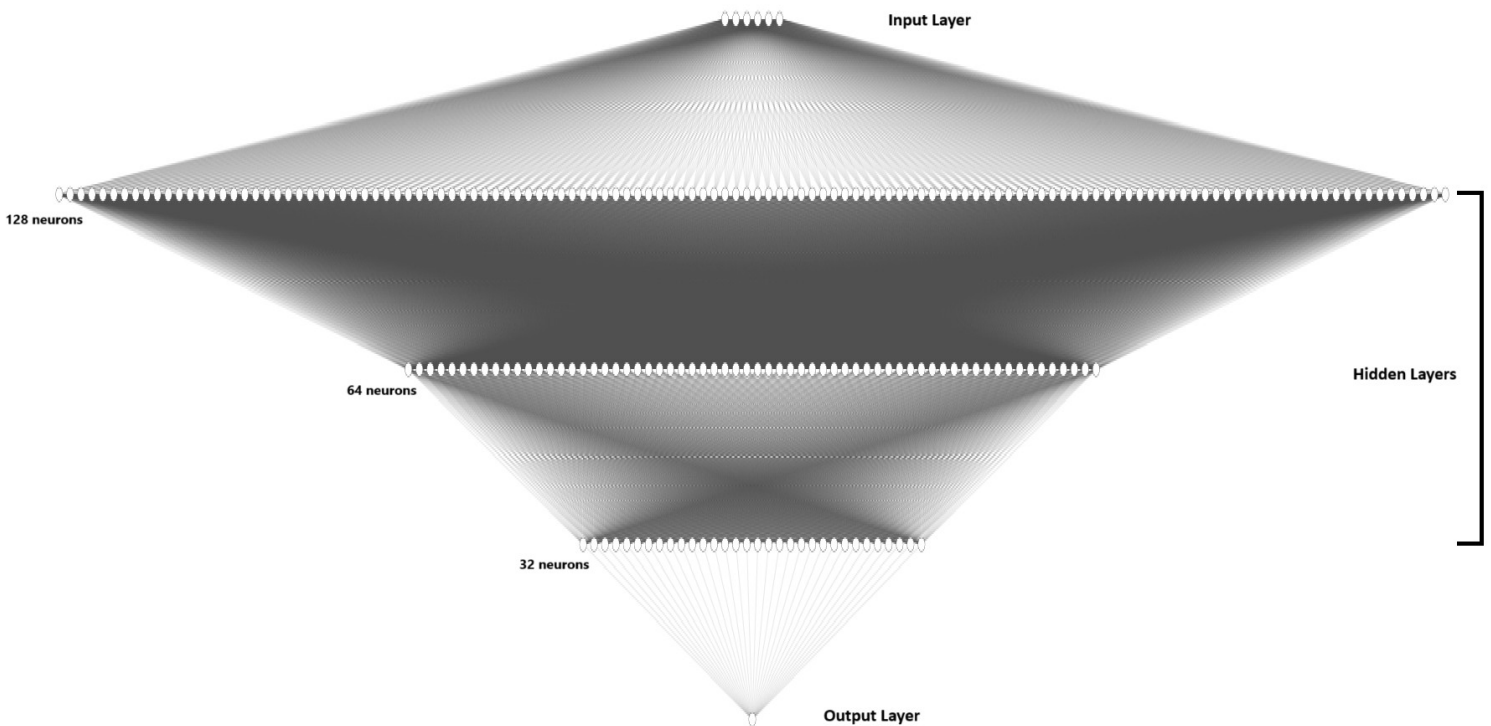
ενεργοποίησης **ReLU** (Rectified Linear Unit), η οποία είναι αποτελεσματική για μη γραμμικά δεδομένα και βοηθά στην αποφυγή του προβλήματος της εξαφανιζόμενης κλίσης.

Τα βασικά στοιχεία της αρχιτεκτονικής των στρωμάτων είναι:

- **Input Layer:** Το στρώμα εισόδου λαμβάνει τα επεξεργασμένα χαρακτηριστικά, τα οποία έχουν μετατραπεί σε αριθμητική μορφή μέσω των βημάτων προεπεξεργασίας.
- **Hidden Layers:** Υπάρχουν τρία κρυφά στρώματα, το καθένα με φθίνοντα αριθμό νευρώνων (128, 64 και 32). Η συνάρτηση ενεργοποίησης ReLU εφαρμόζεται σε κάθε κρυφό στρώμα για την εισαγωγή μη γραμμικότητας στο μοντέλο.
- **Dropout Layers:** Μετά από κάθε κρυφό στρώμα χρησιμοποιείται το Dropout για να αποφευχθεί το overfitting. Απενεργοποιεί τυχαία ένα κλάσμα νευρώνων (30% σε αυτή την περίπτωση) κατά τη διάρκεια κάθε επανάληψης εκπαίδευσης, καθιστώντας το μοντέλο πιο συμπαγές.
- **Output Layer:** Το τελικό στρώμα χρησιμοποιεί μια sigmoid συνάρτηση ενεργοποίησης, η οποία εξάγει μια πιθανότητα μεταξύ 0 και 1, υποδεικνύοντας αν μια συγκεκριμένη άσκηση είναι κατάλληλη για τον χρήστη.

```
model = Sequential([
    Dense(128, input_shape=(X_train.shape[1],), activation='relu'),
    Dropout(0.3),
    Dense(64, activation='relu'),
    Dropout(0.3),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')
])
```

Σχήμα 3.5: Αρχιτεκτονική του Νευρωνικού Δικτύου



Σχήμα 3.6: Γραφική αναπαράσταση του Νευρωνικού Δικτύου με NN SVG (<https://alexlenail.me/NN-SVG/>)

Αφού καθορίστηκε η αρχιτεκτονική, το μοντέλο συντάχθηκε και εκπαιδεύτηκε χρησιμοποιώντας το Adam optimizer, το οποίο είναι γνωστό για τους αποτελεσματικούς και προσαρμοστικούς ρυθμούς μάθησης. Το Adam, συντομογραφία των λέξεων «Adaptive Moment Estimation», είναι ένας επαναληπτικός αλγόριθμος βελτιστοποίησης που χρησιμοποιείται για την ελαχιστοποίηση της loss function κατά τη διάρκεια της εκπαίδευσης. Χρησιμοποιήθηκε η συνάρτηση απώλειας binary cross-entropy, καθώς λύνουμε είναι ένα δυαδικό πρόβλημα ταξινόμησης (αν μια άσκηση είναι κατάλληλη ή όχι).

Βασικές πτυχές του training:

- **Epochs:** Το μοντέλο εκπαιδεύτηκε για 50 epochs, δηλαδή το νευρωνικό δίκτυο “πέρασε” ολόκληρο το σύνολο δεδομένων 50 φορές. Αυτό βοήθησε το μοντέλο να μάθει σταδιακά και να βελτιώσει την ικανότητά του να προβλέπει τις κατάλληλες ασκήσεις.
- **Validation:** Μετά από κάθε epoch, η απόδοση του μοντέλου αξιολογήθηκε στο σύνολο δοκιμών. Αυτή η επικύρωση βοήθησε στην παρακολούθηση της υπερπροσαρμογής, διασφαλίζοντας ότι το μοντέλο θα γενικευόταν καλά σε νέους χρήστες.

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])  
history = model.fit(X_train, y_train, epochs=50, validation_data=(X_test, y_test))
```

Σχήμα 3.7: Εκπαίδευση του μοντέλου

## Διπλωματική Εργασία – Νικόλαος Καλόφωνος

```
Epoch 1/50
173/173 2s 3ms/step - accuracy: 0.6406 - loss: 0.6202 - val_accuracy: 0.8145 - val_loss: 0.4579
Epoch 2/50
173/173 0s 2ms/step - accuracy: 0.7977 - loss: 0.4536 - val_accuracy: 0.8819 - val_loss: 0.3016
Epoch 3/50
173/173 0s 2ms/step - accuracy: 0.8662 - loss: 0.3199 - val_accuracy: 0.9362 - val_loss: 0.1829
Epoch 4/50
173/173 0s 2ms/step - accuracy: 0.9154 - loss: 0.2189 - val_accuracy: 0.9674 - val_loss: 0.1122
Epoch 5/50
173/173 0s 2ms/step - accuracy: 0.9437 - loss: 0.1425 - val_accuracy: 0.9688 - val_loss: 0.0974
Epoch 6/50
173/173 0s 2ms/step - accuracy: 0.9545 - loss: 0.1119 - val_accuracy: 0.9754 - val_loss: 0.0807
Epoch 7/50
173/173 0s 2ms/step - accuracy: 0.9659 - loss: 0.0915 - val_accuracy: 0.9790 - val_loss: 0.0719
Epoch 8/50
173/173 0s 2ms/step - accuracy: 0.9679 - loss: 0.0790 - val_accuracy: 0.9848 - val_loss: 0.0654
Epoch 9/50
173/173 0s 2ms/step - accuracy: 0.9770 - loss: 0.0616 - val_accuracy: 0.9841 - val_loss: 0.0614
Epoch 10/50
173/173 0s 2ms/step - accuracy: 0.9856 - loss: 0.0441 - val_accuracy: 0.9790 - val_loss: 0.0609
Epoch 11/50
173/173 0s 2ms/step - accuracy: 0.9823 - loss: 0.0497 - val_accuracy: 0.9826 - val_loss: 0.0608
Epoch 12/50
173/173 0s 2ms/step - accuracy: 0.9841 - loss: 0.0475 - val_accuracy: 0.9812 - val_loss: 0.0541
Epoch 13/50
173/173 0s 2ms/step - accuracy: 0.9889 - loss: 0.0329 - val_accuracy: 0.9841 - val_loss: 0.0530
Epoch 14/50
173/173 0s 2ms/step - accuracy: 0.9873 - loss: 0.0315 - val_accuracy: 0.9819 - val_loss: 0.0658
Epoch 15/50
173/173 0s 2ms/step - accuracy: 0.9870 - loss: 0.0325 - val_accuracy: 0.9790 - val_loss: 0.0674
Epoch 16/50
173/173 0s 2ms/step - accuracy: 0.9883 - loss: 0.0344 - val_accuracy: 0.9877 - val_loss: 0.0539
Epoch 17/50
173/173 0s 2ms/step - accuracy: 0.9933 - loss: 0.0200 - val_accuracy: 0.9855 - val_loss: 0.0567
Epoch 18/50
173/173 0s 2ms/step - accuracy: 0.9900 - loss: 0.0281 - val_accuracy: 0.9833 - val_loss: 0.0576
Epoch 19/50
173/173 0s 2ms/step - accuracy: 0.9900 - loss: 0.0233 - val_accuracy: 0.9870 - val_loss: 0.0600
Epoch 20/50
173/173 0s 2ms/step - accuracy: 0.9925 - loss: 0.0201 - val_accuracy: 0.9884 - val_loss: 0.0598
Epoch 21/50
173/173 0s 2ms/step - accuracy: 0.9908 - loss: 0.0239 - val_accuracy: 0.9855 - val_loss: 0.0709
Epoch 22/50
173/173 0s 2ms/step - accuracy: 0.9927 - loss: 0.0181 - val_accuracy: 0.9848 - val_loss: 0.0655
Epoch 23/50
173/173 0s 2ms/step - accuracy: 0.9918 - loss: 0.0204 - val_accuracy: 0.9870 - val_loss: 0.0610
Epoch 24/50
173/173 0s 2ms/step - accuracy: 0.9915 - loss: 0.0188 - val_accuracy: 0.9891 - val_loss: 0.0589
Epoch 25/50
173/173 0s 2ms/step - accuracy: 0.9940 - loss: 0.0140 - val_accuracy: 0.9877 - val_loss: 0.0555
Epoch 26/50
173/173 0s 2ms/step - accuracy: 0.9940 - loss: 0.0139 - val_accuracy: 0.9884 - val_loss: 0.0663
Epoch 27/50
173/173 0s 2ms/step - accuracy: 0.9937 - loss: 0.0182 - val_accuracy: 0.9862 - val_loss: 0.0755
Epoch 28/50
173/173 0s 2ms/step - accuracy: 0.9925 - loss: 0.0185 - val_accuracy: 0.9870 - val_loss: 0.0738
Epoch 29/50
173/173 0s 2ms/step - accuracy: 0.9960 - loss: 0.0123 - val_accuracy: 0.9862 - val_loss: 0.0654
Epoch 30/50
173/173 0s 2ms/step - accuracy: 0.9932 - loss: 0.0159 - val_accuracy: 0.9884 - val_loss: 0.0758
Epoch 31/50
173/173 0s 2ms/step - accuracy: 0.9941 - loss: 0.0162 - val_accuracy: 0.9870 - val_loss: 0.0753
Epoch 32/50
173/173 0s 2ms/step - accuracy: 0.9953 - loss: 0.0132 - val_accuracy: 0.9870 - val_loss: 0.0678
Epoch 33/50
173/173 0s 2ms/step - accuracy: 0.9963 - loss: 0.0104 - val_accuracy: 0.9891 - val_loss: 0.0732
Epoch 34/50
173/173 0s 2ms/step - accuracy: 0.9979 - loss: 0.0086 - val_accuracy: 0.9870 - val_loss: 0.0802
Epoch 35/50
173/173 0s 2ms/step - accuracy: 0.9968 - loss: 0.0085 - val_accuracy: 0.9884 - val_loss: 0.0890
Epoch 36/50
173/173 0s 2ms/step - accuracy: 0.9959 - loss: 0.0126 - val_accuracy: 0.9899 - val_loss: 0.0671
Epoch 37/50
173/173 0s 2ms/step - accuracy: 0.9954 - loss: 0.0103 - val_accuracy: 0.9891 - val_loss: 0.0743
Epoch 38/50
173/173 0s 2ms/step - accuracy: 0.9947 - loss: 0.0145 - val_accuracy: 0.9899 - val_loss: 0.0762
Epoch 39/50
173/173 0s 2ms/step - accuracy: 0.9968 - loss: 0.0098 - val_accuracy: 0.9906 - val_loss: 0.0807
Epoch 40/50
173/173 0s 2ms/step - accuracy: 0.9976 - loss: 0.0078 - val_accuracy: 0.9913 - val_loss: 0.0710
Epoch 41/50
173/173 0s 2ms/step - accuracy: 0.9980 - loss: 0.0069 - val_accuracy: 0.9906 - val_loss: 0.0671
Epoch 42/50
173/173 1s 2ms/step - accuracy: 0.9987 - loss: 0.0058 - val_accuracy: 0.9899 - val_loss: 0.0842
Epoch 43/50
173/173 0s 2ms/step - accuracy: 0.9975 - loss: 0.0098 - val_accuracy: 0.9899 - val_loss: 0.0825
Epoch 44/50
173/173 0s 2ms/step - accuracy: 0.9936 - loss: 0.0140 - val_accuracy: 0.9855 - val_loss: 0.0973
Epoch 45/50
173/173 0s 2ms/step - accuracy: 0.9973 - loss: 0.0112 - val_accuracy: 0.9884 - val_loss: 0.0762
Epoch 46/50
173/173 0s 2ms/step - accuracy: 0.9972 - loss: 0.0076 - val_accuracy: 0.9870 - val_loss: 0.0831
Epoch 47/50
173/173 0s 2ms/step - accuracy: 0.9974 - loss: 0.0071 - val_accuracy: 0.9862 - val_loss: 0.0811
Epoch 48/50
173/173 0s 2ms/step - accuracy: 0.9955 - loss: 0.0111 - val_accuracy: 0.9870 - val_loss: 0.0949
Epoch 49/50
173/173 0s 2ms/step - accuracy: 0.9967 - loss: 0.0077 - val_accuracy: 0.9899 - val_loss: 0.0806
Epoch 50/50
173/173 0s 2ms/step - accuracy: 0.9994 - loss: 0.0039 - val_accuracy: 0.9906 - val_loss: 0.0768
```

Σχήμα 3.8: Αναπαράσταση της σταδιακής αύξησης της ακρίβειας μέσω των epochs



Η απόδοση του μοντέλου παρακολουθήθηκε με τη χρήση της ακρίβειας, η οποία δείχνει το ποσοστό των σωστών προβλέψεων, και της απώλειας, η οποία αντικατοπτρίζει πόσο καλά οι προβλέψεις του μοντέλου ταιριάζουν με τις πραγματικές ετικέτες. Ο στόχος ήταν η ελαχιστοποίηση της απώλειας και η μεγιστοποίηση της ακρίβειας. Στο τέλος της διαδικασίας εκπαίδευσης, το μοντέλο αξιολογήθηκε στο σύνολο δοκιμών για να διασφαλιστεί ότι δεν έκανε overfit στα δεδομένα εκπαίδευσης και ότι μπορούσε να γενικευτεί σε νέους χρήστες. Η ακρίβεια που επιτυγχάνεται στο σύνολο δοκιμών καθορίζει πόσο αξιόπιστο είναι το μοντέλο στη σύσταση ασκήσεων. Τέλος, το μοντέλο και ο προεπεξεργαστής που δημιουργήθηκαν αποθηκεύονται για την χρήση τους στην επόμενη διαδικασία.

### 3.2.3 Παραγωγή πλάνων γυμναστικής

Μόλις ολοκληρωθεί η εκπαίδευση του μοντέλου, μπορεί να χρησιμοποιηθεί για τη δημιουργία εξατομικευμένων πλάνων γυμναστικής για τους χρήστες. Το αρχείο `workout_plan.py` χρησιμοποιεί το εκπαιδευμένο μοντέλο για να προβλέψει την καταλληλότητα των ασκήσεων για έναν νέο χρήστη, με βάση τις πληροφορίες του χρήστη τις οποίες ανακτά από τη βάση δεδομένων της εφαρμογής. Σχηματίζεται ένα dataframe αναμειγνύοντας δεδομένα του χρήστη με το dataset των ασκήσεων και γίνεται προεπεξεργασία σε αυτό με βάση το αρχείο που παράχθηκε στο στάδιο της εκπαίδευσης. Το μοντέλο προβλέπει μια βαθμολογία καταλληλότητας για κάθε άσκηση, όπου οι ασκήσεις με βαθμολογία πάνω από ένα ορισμένο όριο (στην προκειμένη περίπτωση 0.5) θεωρούνται κατάλληλες για τον χρήστη. Εφαρμόζεται επίσης ένα μπόνους εάν το αθλητικό επίπεδο του χρήστη ταιριάζει με το επίπεδο δυσκολίας της άσκησης, εξασφαλίζοντας μια πιο ακριβή επιλογή.

```
suitability_scores = model.predict(X_new_user)

bonus = 0.1
user_exercise_df['Suitability'] = suitability_scores.flatten()
for i, row in user_exercise_df.iterrows():
    exercise_difficulty = row['Difficulty_Level']
    if exercise_difficulty == user_skill_level:
        user_exercise_df.at[i, 'Suitability'] += bonus
    elif (user_skill_level - exercise_difficulty) == 1:
        user_exercise_df.at[i, 'Suitability'] += bonus / 2
    elif (user_skill_level - exercise_difficulty) == 2:
        user_exercise_df.at[i, 'Suitability'] += bonus / 3

suitability_threshold = 0.5

valid_exercises = user_exercise_df[user_exercise_df['Suitability'] >= suitability_threshold]
valid_exercises = valid_exercises.sort_values(by='Suitability', ascending=False)
```

**Σχήμα 3.9:** Αξιολόγηση των ασκήσεων σχετικά με τον χρήστη και εύρεση των καταλληλότερων

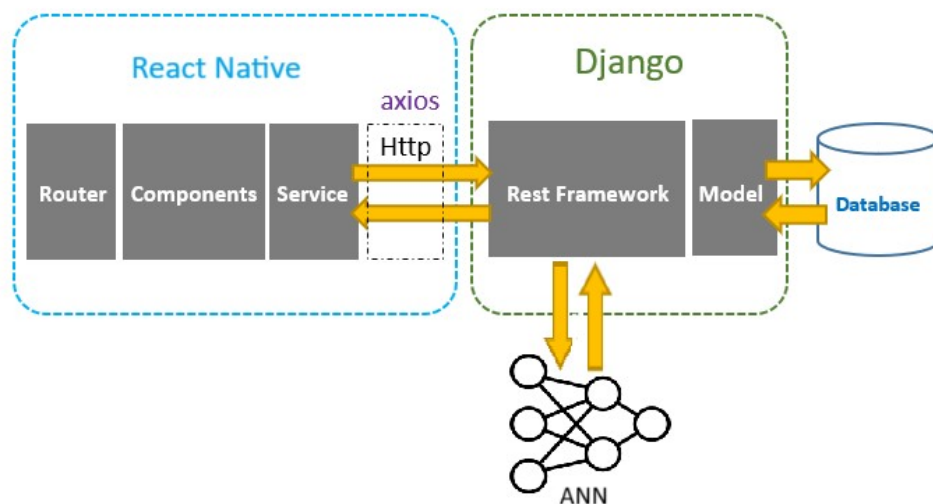
Αφού βρεθούν οι καταλληλότερες ασκήσεις για τον χρήστη ξεκινάει η διαδικασία του σχηματισμού του πλάνου γυμναστικής. Με βάση τις προπονήσεις που έχει καταχωρηθεί στη βάση ότι θέλει ο χρήστης, υπολογίζεται ο αριθμός των ασκήσεων που περιλαμβάνονται για την εβδομάδα (προπονήσεις \* 4 ασκήσεις ανα ημέρα). Στην περίπτωση που οι έγκυρες ασκήσεις είναι λίγες λόγω των ιδιομορφιών της κατάστασης κάποιου χρήστη, το πρόγραμμα αναπαράγει κάποιες ασκήσεις από αυτές που προβλέπηκαν για να συμπληρωθεί ο απαιτούμενος αριθμός, ενώ παράλληλα όλες οι ασκήσεις είναι κατάλληλες. Τέλος, δημιουργείται μία λίστα που αποτελείται από “Day x” και μετά τις ασκήσεις της συγκεκριμένης προπόνησης και αποθηκεύεται στη βάση δεδομένων.

## **Κεφάλαιο 4: Σχεδίαση και Υλοποίηση του Συστήματος**

### **4.1 Επισκόπηση αρχιτεκτονικής**

Η εφαρμογή ακολουθεί ένα μοντέλο client-server. Το React Native διαχειρίζεται το frontend, ενώ το Django χειρίζεται το backend και αλληλεπιδρά με το μοντέλο νευρωνικού δικτύου. Αυτή η αρχιτεκτονική εξασφαλίζει αποτελεσματική ροή δεδομένων και προσαρμογές του προγράμματος προπόνησης σε πραγματικό χρόνο με βάση την ανατροφοδότηση του χρήστη. Η ροή των δεδομένων λειτουργεί ως εξής:

- **Εισαγωγή δεδομένων:** Ο χρήστης αλληλεπιδρά με την εφαρμογή μέσω της διεπαφής React Native. Μπορεί να υποβάλει δεδομένα, όπως προτιμήσεις, επιλογές ή πληροφορίες που πρέπει να υποβληθούν σε επεξεργασία. Τα δεδομένα αυτά συλλέγονται και αποστέλλονται από το Axios HTTP client στο backend μέσω ενός αιτήματος API του πλαισίου Django REST. Το DRF επιτρέπει στο API να χειρίζεται τη serialization και το validation των δεδομένων των χρηστών, διασφαλίζοντας ότι πληρούν την απαιτούμενη μορφή και δομή πριν από την περαιτέρω επεξεργασία.
- **Επεξεργασία backend:** Μετά την επικύρωση, το backend επεξεργάζεται την είσοδο. Το Django ORM (Object-Relational Mapping) ανακτά υπάρχουσες εγγραφές από τη βάση δεδομένων ή την ενημερώνει με νέα δεδομένα. Εάν τα δεδομένα προορίζονται για το νευρωνικό δίκτυο, το Django τα προεπεξεργάζεται και τα περνά ως εισόδους
- **Πρόβλεψη νευρωνικού δικτύου:** Το μοντέλο νευρωνικού δικτύου αναλύει τα δεδομένα χρήστη. Χρησιμοποιώντας τις προεπεξεργασμένες εισόδους, προβλέπει τις καταλληλότερες ασκήσεις και φτιάχνει το σχέδιο γυμναστικής για τον χρήστη. Αυτές οι προβλέψεις επιστρέφονται στο backend για επεξεργασία.
- **Απόκριση backend (δεδομένα νευρωνικού δικτύου):** Μετά την επεξεργασία των προβλέψεων του νευρωνικού δικτύου, το Django χρησιμοποιεί το πλαίσιο Django REST για να δομήσει τα δεδομένα σε μια απάντηση JSON. Στη συνέχεια, η απόκριση αποστέλλεται πίσω στο frontend μέσω του αιτήματος Axios που χρησιμοποιήθηκε νωρίτερα.
- **Έξοδος frontend:** Το frontend λαμβάνει την απάντηση JSON και ενημερώνει το UI με βάση τα επιστρεφόμενα δεδομένα.



Σχήμα 4.1: Γραφική αναπαράσταση της αρχιτεκτονικής του συστήματος

## 4.2 Ανάλυση του Front-End

Σε αυτή την ενότητα θα αναλυθεί η δόμηση του front-end της εφαρμογής. Όπως έχει προαναφερθεί, το front-end κομμάτι της εφαρμογής αναπτύχθηκε με React Native. Τα βασικά στοιχεία της εφαρμογής περιλαμβάνουν έναν stack navigator, έναν drawer navigator και context providers, οι οποίοι μαζί διευκολύνουν την πλοήγηση και το state management. Τα αρχεία τα περιέχουν αυτή τη λειτουργικότητα είναι τα App.js, stack.js και drawer.js.

Το αρχείο **App.js** χρησιμεύει ως σημείο εισόδου για την εφαρμογή και ενσωματώνει βασικά στοιχεία πλοήγησης και state management. Δημιουργεί έναν RootStack navigator, ο οποίος είναι ένας πλοηγός στοίβας που διαχειρίζεται την πρωταρχική ροή του navigation της εφαρμογής.

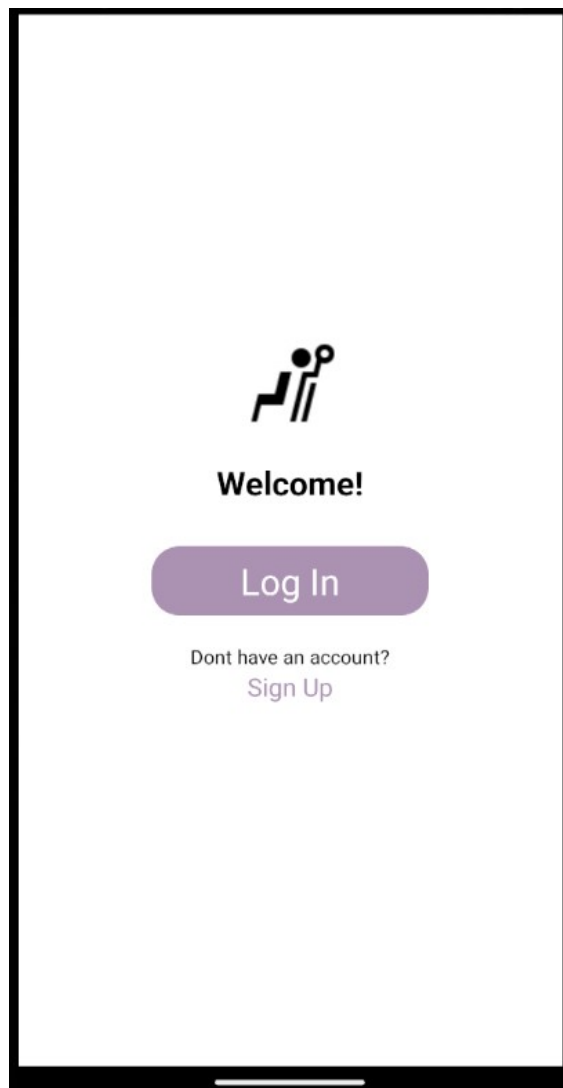
Το αρχείο **stack.js** ορίζει πολλαπλούς stack navigators, καθένας από τους οποίους διαχειρίζεται ένα διαφορετικό σύνολο οθονών:

- StartStack: με οθόνες για την εκκίνηση της εφαρμογής, το Log In και το Sign Up.
- ProfileStack: Διαχειρίζεται τις οθόνες που σχετίζονται με το προφίλ του χρήστη. Η οθόνη UserProfileScreen είναι η κύρια οθόνη σε αυτή τη στοίβα, παρέχοντας πρόσβαση σε πληροφορίες του προφίλ του χρήστη.
- HomeStack: Περιλαμβάνει οθόνες που σχετίζονται με τις κύριες λειτουργίες της εφαρμογής:
  - HomeScreen: Το κεντρικό ταμπλό ή η αρχική σελίδα.
  - UserInfoForm: Μια φόρμα για τη συλλογή ή την ενημέρωση πληροφοριών χρήστη.
  - PlanDayExercises: Διαχειρίζεται τα σχέδια άσκησης για συγκεκριμένες ημέρες.

- LibraryStack: Επικεντρώνεται στο περιεχόμενο που σχετίζεται με την βιβλιοθήκη ασκήσεων:
- SettingsStack: Παρέχει μια οθόνη για τη διαχείριση των ρυθμίσεων λογαριασμού.

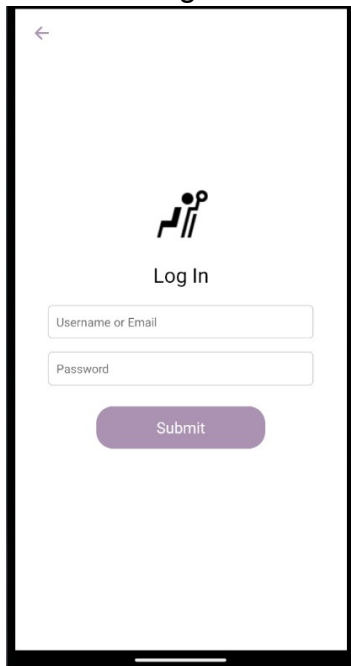
Τα παραπάνω καθιστούν δυνατή την εύρυθμη λειτουργία του front-end, παρουσιάζοντας ένα λειτουργικό User Interface. Παρακάτω ακολουθεί η ανάλυση του UI, με σκοπό την εξήγηση όλων των δυνατοτήτων της εφαρμογής και το πώς παρουσιάζονται στον χρήστη.

Στο άνοιγμα της εφαρμογής, ο χρήστης αντικρίζει την οθόνη εκκίνησης:



**Εικόνα 4.1:** Οθόνη εκκίνησης

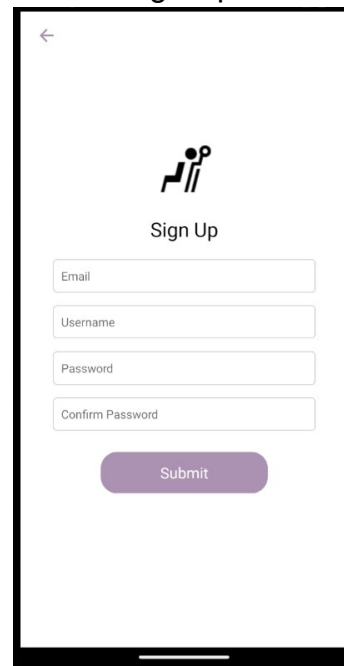
Log In:



A mobile app screen for logging in. It features a back arrow in the top left corner, a fitness icon (two figures running) in the center, and the text "Log In" below it. There are two input fields: "Username or Email" and "Password". A purple "Submit" button is at the bottom.

Εικόνα 4.2: Log In

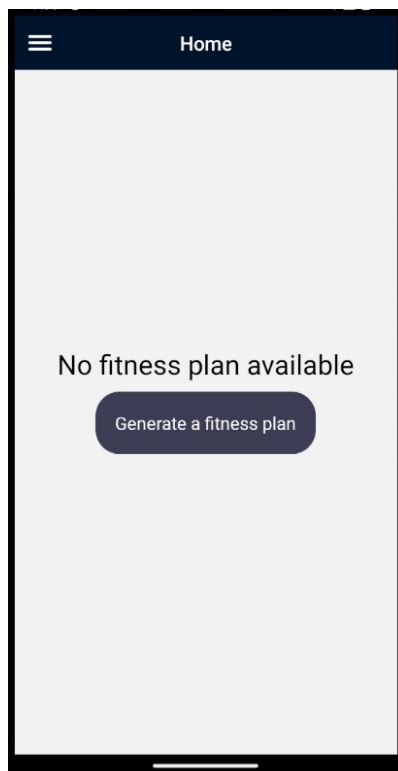
Sign Up:



A mobile app screen for signing up. It features a back arrow in the top left corner, a fitness icon (two figures running) in the center, and the text "Sign Up" below it. There are four input fields: "Email", "Username", "Password", and "Confirm Password". A purple "Submit" button is at the bottom.

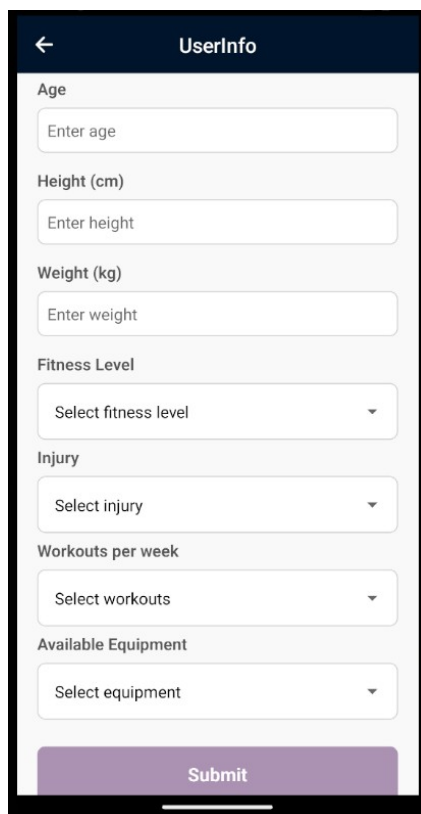
Εικόνα 4.3: Sign Up

Αφού εγγραφεί και ταυτοποιηθεί ο χρήστης, εισέρχεται στην αρχική οθόνη:



Εικόνα 4.3: Αρχική οθόνη

Πατώντας το κουμπί “**Generate a fitness plan**” παρουσιάζεται μία φόρμα συμπλήρωσης στοιχείων, όπου ο χρήστης δίνει όλα τα απαραίτητα στοιχεία ώστε να παραχθεί το εξατομικευμένο πλάνο:



Εικόνα 4.4: Φόρμα πληροφοριών χρήστη

Οι διαθέσιμες επιλογές για το επίπεδο φυσικής κατάστασης του χρήστη είναι **Beginner**, **Intermediate** και **Advanced**. Η κάθε επιλογή περιλαμβάνει μία εξήγηση ώστε ο χρήστης να επιλέξει το επίπεδο που του ταιριάζει:

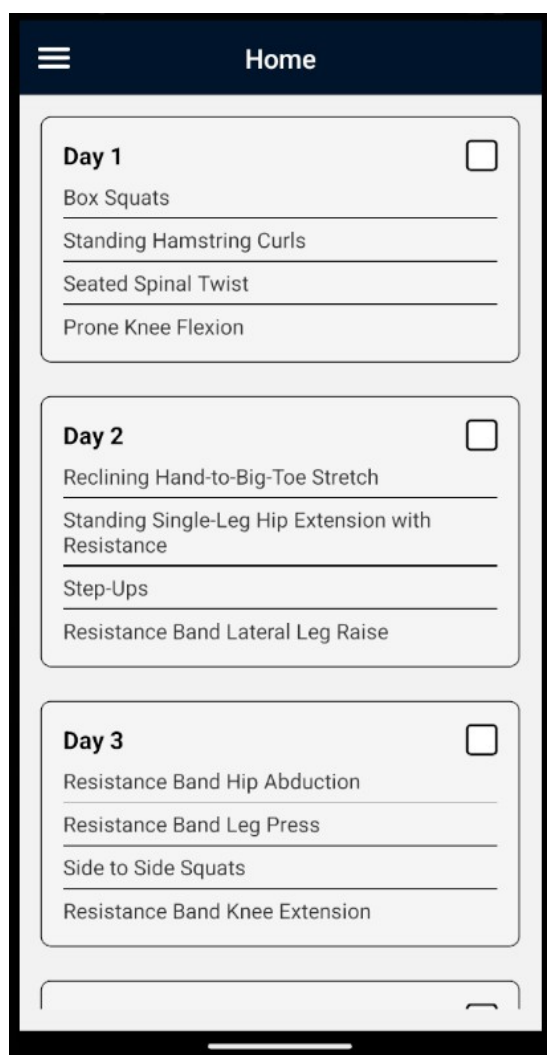
| Fitness Level   | Fitness Level   | Fitness Level  |
|---|---|--|
| <input type="text" value="Beginner"/>   | <input type="text" value="Intermediate"/>   | <input type="text" value="Advanced"/>  |
| <i>This level is suited for users in the early stages of rehabilitation or those with limited prior athletic experience. If you are recovering from a recent injury or surgery, have restricted movement, or are experiencing significant muscle weakness, this is the right level.</i> | <i>This level is designed for users in the mid-stage of rehabilitation or those who have some athletic background but are not yet ready for high-intensity exercise. If you've made progress in your recovery but still need to build strength, endurance, and flexibility, this is your level.</i> | <i>This level is intended for users in the late stages of rehabilitation or those with a high level of previous athletic experience. If you are nearing full recovery and looking to restore athletic performance or resume intense physical activities, this is the right choice.</i> |

Εικόνα 4.5: Εξήγηση επιπέδων

- Οι επιλογές τραυματισμού είναι αυτές που προαναφέρθηκαν: **ACL, Meniscus, Achilles**
- Οι επιλογές προπονήσεων ανά εβδομάδα είναι 3,4 ή 5
- Οι επιλογές εξοπλισμού είναι: **None, Resistance Bands, Gym Subscription**

Αφού συμπληρωθεί η φόρμα, ο χρήστης πατάει το κουμπί “**Submit**” και τα δεδομένα στέλνονται στο back-end για να υποστούν επεξεργασία και να παραχθεί το πλάνο γυμναστικής.

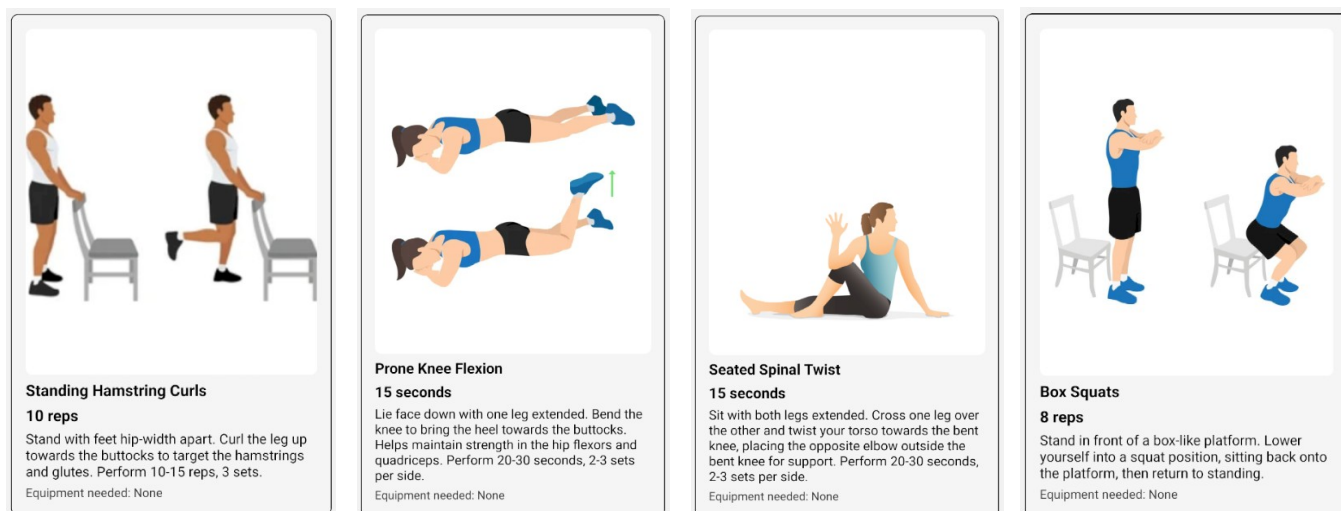
Αφού ολοκληρωθεί η διαδικασία της δημιουργίας του πλάνου, αυτό παρουσιάζεται στην αρχική οθόνη με κατανοητή και φιλική προς τον χρήστη μορφή:



**Εικόνα 4.6:** Αρχική οθόνη με πλάνο γυμναστικής

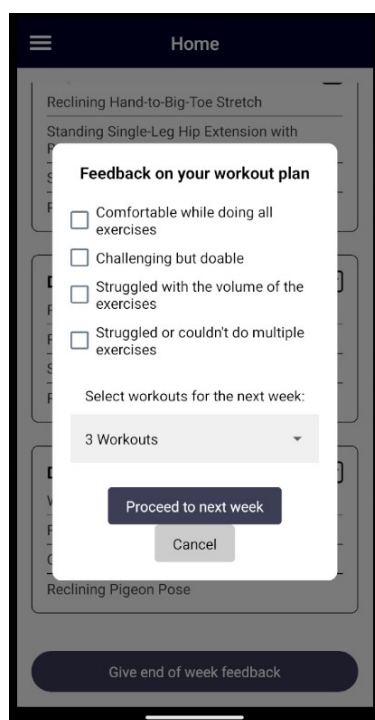
Πατώντας το κουτάκι που βρίσκεται πάνω δεξιά στο πλαίσιο της κάθε μέρας, ο χρήστης κάνει “τικ” σε όποια προπόνηση έχει ολοκληρώσει. Για την αναλυτική προβολή της κάθε προπόνησης ο χρήστης πατάει επάνω στο πλαίσιο της κάθε ημέρας, όπου η παρουσίαση της κάθε άσκησης περιλαμβάνει εικόνα, επαναλήψεις, περιγραφή και απαραίτητο εξοπλισμό:





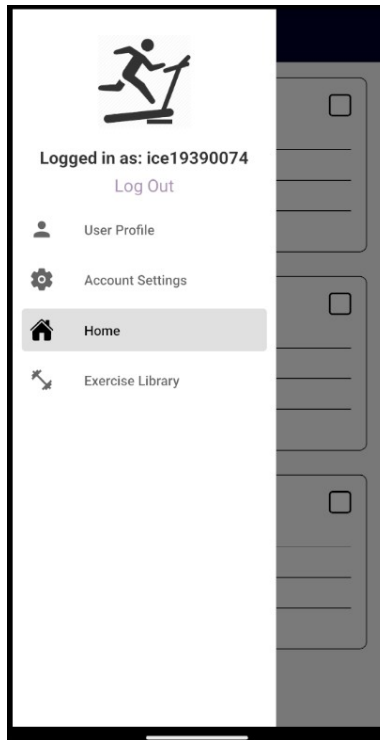
Εικόνα 4.7: Παραδείγματα format επεξήγησης άσκησης

Αφού ο χρήστης σημειώσει ότι ολοκλήρωσε όλες τις προπονήσεις της εβδομάδας, εμφανίζεται το κουμπί “Give end of week feedback” στο κάτω μέρος της οθόνης. Πατώντας το, εμφανίζεται ένα παράθυρο στο οποίο ο χρήστης καλείται να σχολιάσει το πλάνο με βάση τη δυσκολία του ώστε να προσαρμοστεί ανάλογα για την επόμενη εβδομάδα. Ο χρήστης μπορεί επίσης να επιλέξει εκ νέου τον αριθμό των ημερών προπόνησης για το επόμενο πλάνο. Αφού επιλέξει το κουμπί “Proceed to next week”, το πλάνο αναπροσαρμόζεται και η εφαρμογή επιστρέφει ξανά στην αρχική οθόνη.



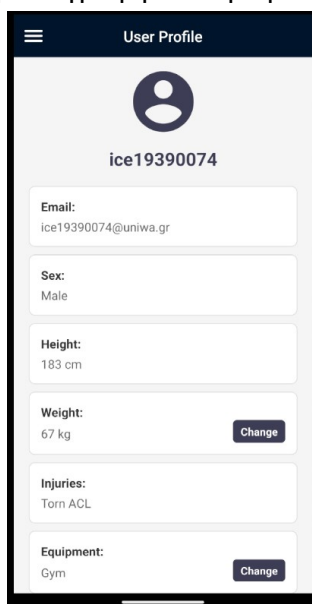
Εικόνα 4.8: Παράθυρο feedback χρήστη

Το κουμπί στο πάνω αριστερά μέρος της οθόνης υπάρχει το κουμπί που ενεργοποιεί το drawer menu. Εκεί περιέχονται οι υπόλοιπες λειτουργίες της εφαρμογής:

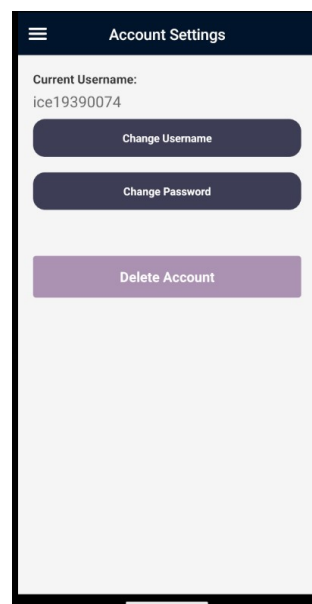


Εικόνα 4.9: Drawer menu

Το User Profile περιέχει τις καταχωρημένες πληροφορίες του χρήστη, επιτρέποντάς του να αλλάξει τις μεταβλητές πληροφορίες που έχουν καταχωρηθεί στο σύστημα όπως το βάρος και ο εξοπλισμός, ενώ το Account Settings περιέχει ρυθμίσεις σχετικά με την αλλαγή username και password, ή την διαγραφή του προφίλ.

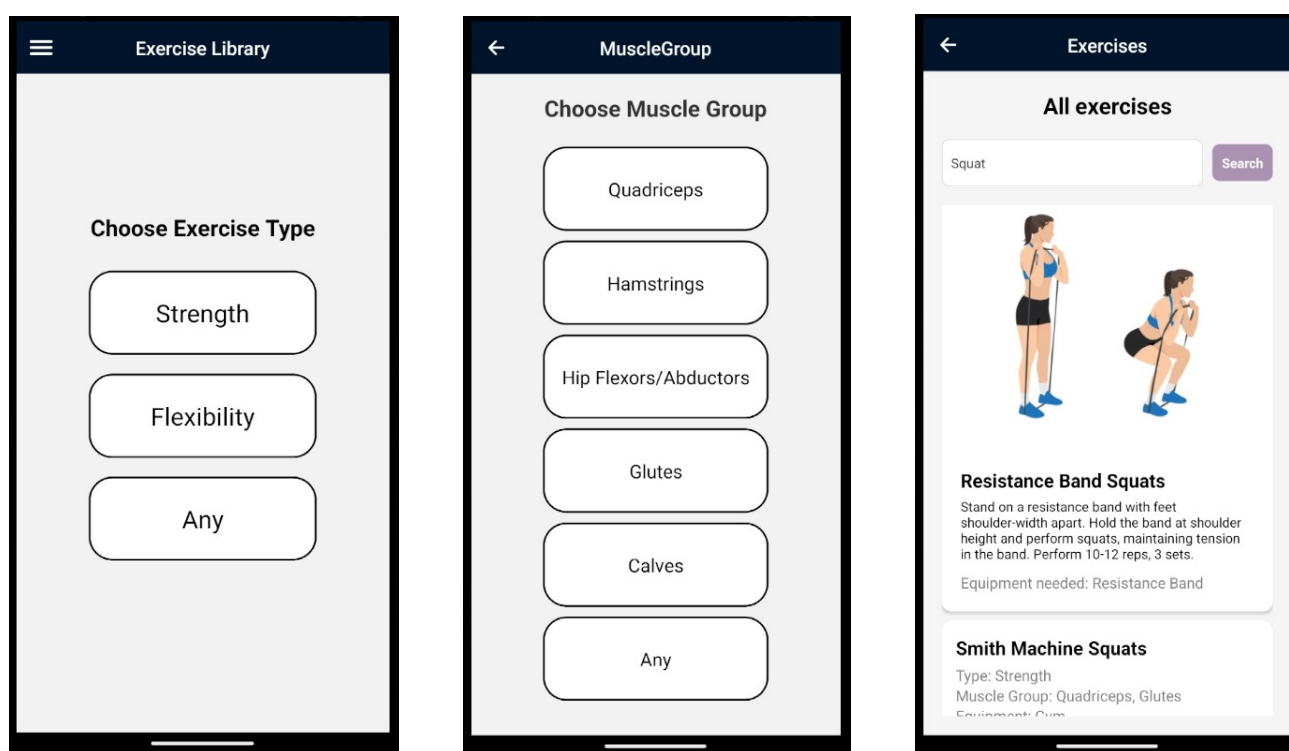


Εικόνα 4.10: User Profile



Εικόνα 4.11: Account Settings

Η επιλογή **Exercise Library** περιέχει την αρχειοθέτηση όλων των ασκήσεων που συμπεριλαμβάνονται στην εφαρμογή και είναι διαθέσιμες για να προταθούν σε κάποιο πιθανό πλάνο γυμναστικής. Κατά την είσοδο σε αυτή, ο χρήστης επιλέγει αν θέλει να φιλτράρει με βάση τον τύπο άσκησης (Strength/Flexibility) και ύστερα με βάση την μυϊκή ομάδα που στοχεύεται κατά την άσκηση. Η βιβλιοθήκη εμφανίζει τις ασκήσεις που απομένουν μετά από το φιλτράρισμα σε ένα πύο συνοπτικό format, ενώ αν ο χρήστης επιθυμεί να δει το πιο λεπτομερές format πρέπει να πατήσει επάνω στην άσκηση. Τέλος υπάρχει μία μπάρα αναζήτησης στο πάνω μέρος της οθόνης για περαιτέρω φιλτράρισμα των ασκήσεων.



Εικόνα 4.12: Exercise Library

Συνοψίζοντας, η εφαρμογή προσφέρει ένα φιλικό προς το χρήστη user interface και αποτελεί εύκολα προσβάσιμη λύση για το θέμα με το οποίο ασχολείται, παρέχοντας εξατομικευμένα σχέδια προπόνησης που υποστηρίζονται από ένα αξιόπιστο νευρωνικό δίκτυο. Αυτό το ANN επιλέγει τις ασκήσεις με βάση τις πληροφορίες του χρήστη και προσαρμόζει τη διεπαφή ανάλογα. Επιπλέον, δίνεται η ευκαιρία στον χρήστη να εξερευνήσει ασκήσεις πέρα από αυτές που προτείνονται στο πρόγραμμά του, παρέχοντας τις απαραίτητες πληροφορίες για αυτές ώστε να διευκολυνθεί η εκπαίδευσή του σε θέματα φυσικής κατάστασης. Τέλος, ο χρήστης έχει τη δυνατότητα να επεξεργαστεί τις πληροφορίες του προφίλ του με όποιον τρόπο επιθυμεί, ώστε να αναπροσαρμόζεται ανάλογα το state της εφαρμογής. Έτσι, δημιουργείται μια ευέλικτη εμπειρία που

εξελισσεται ανάλογα με τις ανάγκες του χρήστη, ενισχύοντας τόσο την ευκολία χρήσης όσο και την αποτελεσματικότητα.

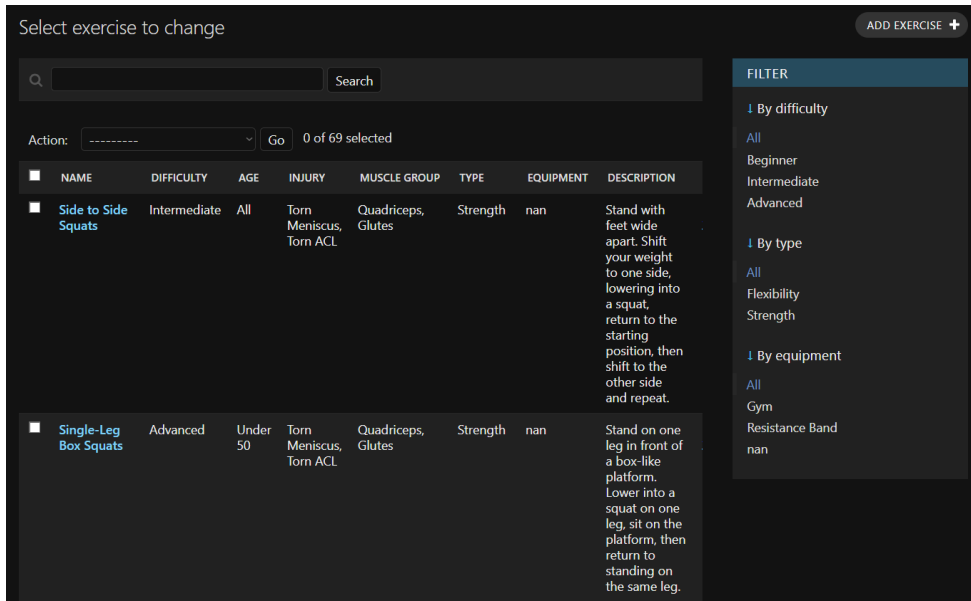
## 4.2 Ανάλυση του Back-End

Το Django backend αποτελεί τον πυρήνα της επεξεργασίας δεδομένων της εφαρμογής, χειριζόμενο τις αλληλεπιδράσεις μεταξύ της βάσης δεδομένων, του API και του UI. Αξιοποιεί την default βάση δεδομένων SQLite που παρέχει το Django framework, ένα ελαφρύ και file-based σχεσιακό σύστημα βάσεων δεδομένων, το οποίο είναι ιδανικό για ανάπτυξη λόγω της απλότητας και της ευκολίας ρύθμισής του.

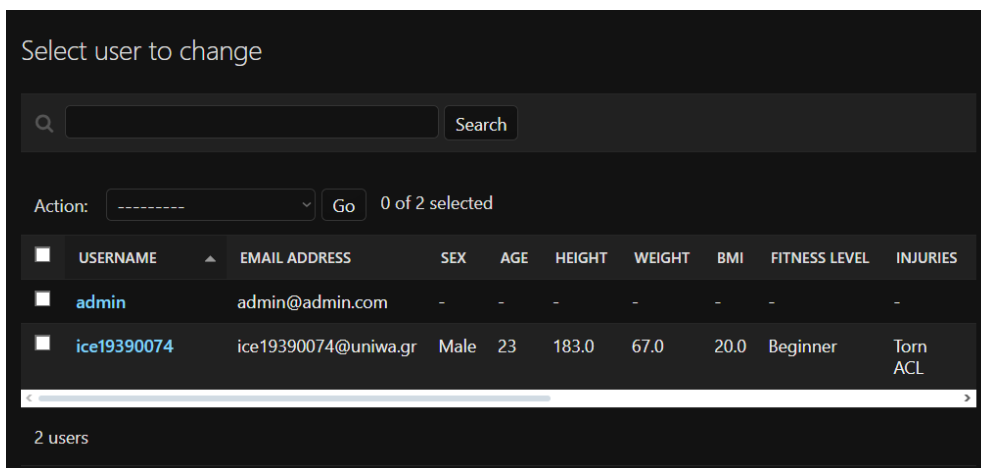
Η καρδιά της αλληλεπίδρασης του Django με τη βάση δεδομένων είναι το αρχείο **models.py**. Τα μοντέλα είναι κλάσεις Python που αναπαριστούν πίνακες στη βάση δεδομένων. Κάθε χαρακτηριστικό της κλάσης αντιστοιχεί σε μια στήλη της βάσης δεδομένων, ενώ κάθε entry του μοντέλου αναπαριστά μια γραμμή. Όταν οριστεί ένα μοντέλο στο models.py, το Django δημιουργεί αυτόματα τους απαραίτητους πίνακες της βάσης δεδομένων και χειρίζεται όλα τα SQL queries στο παρασκήνιο. Η βάση δεδομένων επεξεργάζεται τα queries χρησιμοποιώντας το ORM (Object-Relational Mapping) του Django, καθιστώντας τη διαχείριση δεδομένων αποτελεσματική και εύκολη. Τα τρία βασικά μοντέλα που χρησιμοποιούνται για την διαχείριση των δεδομένων της βάσης είναι το **CustomUser**, το οποίο αποθηκεύει όλες τις απαραίτητες πληροφορίες για τον χρήστη, όπως username, password, session token και τις υπόλοιπες πληροφορίες που καθιστούν το πλάνο γυμναστικής, το **Exercise**, το οποίο αποθηκεύει πληροφορίες για την κάθε άσκηση και το **WorkoutPlan**, το οποίο αποθηκεύει ένα JSON που περιέχει τις ημέρες προπόνησης ακολουθούμενες από τα ονόματα των ασκήσεων της κάθε μέρας για κάθε χρήστη που έχει πλάνο. Αφού οριστούν τα μοντέλα, το Django χρησιμοποιεί migrations για να εφαρμόσει αυτόματα τις αλλαγές στο schema της βάσης δεδομένων. Αυτό εξασφαλίζει ότι η δομή της βάσης δεδομένων παραμένει συνεπής με τα μοντέλα καθώς η εφαρμογή εξελίσσεται.

Ένα από τα χρησιμότερα χαρακτηριστικά του Django είναι το interactive **admin interface**, το οποίο επιτρέπει στους προγραμματιστές να αλληλεπιδρούν απευθείας με τη βάση δεδομένων χωρίς να χρειάζεται να γράψουν κώδικα και queries. Το admin panel παρέχει ένα περιβάλλον για το create, read, update, και delete (CRUD) δεδομένων στη βάση δεδομένων SQLite. Αυτή η διεπαφή δημιουργείται αυτόματα για κάθε μοντέλο που ορίζεται στο models.py και περιλαμβάνεται στο αρχείο **admin.py**, μαζί με τα fields του μοντέλου που επιθυμεί ο admin να εμφανίζονται. Στην προκειμένη περίπτωση, τα δεδομένα της βάσης εμφανίζονται ως εξής:

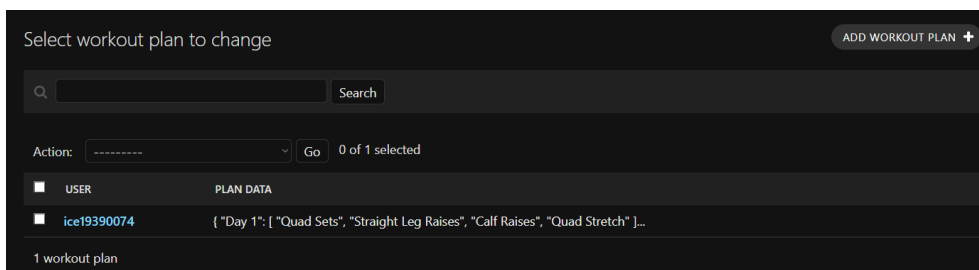
## Διπλωματική Εργασία – Νικόλαος Καλόφωνος



Εικόνα 4.13: Μοντέλο Exercise στο Admin Interface



Εικόνα 4.14: Μοντέλο CustomUser στο Admin Interface



Εικόνα 4.15: Μοντέλο WorkoutPlan στο Admin Interface

Το Django περιλαμβάνει επίσης ένα σύστημα middleware που λειτουργεί μεταξύ των κύκλων request και response. Το middleware χρησιμοποιείται για να χειριστεί εργασίες όπως το authentication του χρήστη, το session management, η ασφάλεια και το data validation.

Στην αρχιτεκτονική του Django, το **views.py** χειρίζεται τη λογική της εφαρμογής και επεξεργάζεται τα αιτήματα από το frontend. Κάθε view είναι ένα class που περιλαμβάνει λογική και συνδέεται με ένα πρότυπο URL που ορίζεται στο **urls.py**, το οποίο αντιστοιχίζει τα εισερχόμενα αιτήματα HTTP στις αντίστοιχες view functions. Αυτή η αντιστοίχιση είναι σημαντική για τη δρομολόγηση των αιτημάτων API από το frontend του React Native στην κατάλληλη λογική του backend. Για παράδειγμα, το αρχείο **api.js** στην πλευρά του React Native χρησιμοποιεί μεθόδους HTTP (όπως GET, POST, PUT, DELETE) για την αποστολή αιτημάτων σε συγκεκριμένα endpoints στο backend. Αυτά τα αιτήματα λαμβάνονται από τις view functions ή τα API που ορίζονται στο αρχείο views.py, τα οποία επεξεργάζονται τα δεδομένα, αλληλεπιδρούν με τη βάση δεδομένων (μέσω των μοντέλων) και επιστρέφουν μια απάντηση (συνήθως σε μορφή JSON). Αυτή η αλληλεπίδραση αποτελεί τη ραχοκοκαλιά της ροής δεδομένων της εφαρμογής, επιτρέποντας στο frontend να εμφανίζει και να τροποποιεί τα δεδομένα στη βάση δεδομένων.

Η εφαρμογή επίσης ενσωματώνει το νευρωνικό δίκτυο μέσω των views ενεργοποιώντας την αλληλεπίδρασή του με το front-end. Συγκεκριμένα για τη συνάρτηση διαχείρισης της φόρμας υποβολής του χρήστη, το αρχείο api.js χειρίζεται την επικοινωνία από την πλευρά του πελάτη με το backend του Django στέλνοντας ένα αίτημα POST που περιέχει δεδομένα χρήστη μέσω της συνάρτησης userInfo. Αυτή η συνάρτηση κάνει μια αίτηση HTTP στο /userInfo/ endpoint χρησιμοποιώντας το Axios, για να επικοινωνήσει με το API του Django. Στο backend, το αίτημα συλλαμβάνεται από το αρχείο urls.py του Django, το οποίο αντιστοιχίζει τη διαδρομή /userInfo/ στο αντίστοιχο UserInfoView στο αρχείο views.py. Μέσα στη μέθοδο post της κλάσης UserInfoView, γίνεται επεξεργασία των δεδομένων του αιτήματος. Στη συνέχεια, το backend ενημερώνει το μοντέλο CustomUser, αποθηκεύοντας το ενημερωμένο προφίλ χρήστη στη βάση δεδομένων. Μόλις αποθηκευτούν τα δεδομένα του χρήστη, το UserInfoView ενεργοποιεί το ANN περνώντας τα ενημερωμένα δεδομένα στη συνάρτηση generate\_plan. Αυτή η συνάρτηση χρησιμοποιεί το προφίλ φυσικής κατάστασης του χρήστη για να δημιουργήσει το εξατομικευμένο σχέδιο προπόνησης. Το παραγόμενο σχέδιο επιστρέφεται στη συνέχεια στο API response, επιβεβαιώνοντας ότι το προφίλ του χρήστη έχει ενημερωθεί και ότι έχει δημιουργηθεί επιτυχώς ένα σχέδιο προπόνησης. Παρακάτω επιδεικνύεται η διασύνδεση των αρχείων api.js, urls.py και views.py ώστε να έρθει εις πέρας η διαδικασία:

```
//api.js
export const userInfo = (username, sex, age, height, weight, fitness_level, injuries,
preferred_workout_times, available_equipment, hasPlan, planWeek) => {
  return axios.post(`${API_URL}/userInfo/`, { username, sex, age, height, weight,
fitness_level, injuries, preferred_workout_times, available_equipment, hasPlan, planWeek });
};
```

Σχήμα 4.2: api.js

```
#urls.py
urlpatterns = [
...
  path('api/userInfo/', UserInfoView.as_view(), name='userInfo'),
...
]
```

Σχήμα 4.3: urls.py

```
#views.py
class UserInfoView(APIView):
    def post(self, request):
        username = request.data.get('username')
        sex = request.data.get('sex')
        age = request.data.get('age')
        height = request.data.get('height')
        weight = request.data.get('weight')
        fitness_level = request.data.get('fitness_level')
        injuries = request.data.get('injuries')
        preferred_workout_times = request.data.get('preferred_workout_times')
        available_equipment = request.data.get('available_equipment')
        has_plan = request.data.get('hasPlan')
        plan_week = request.data.get('planWeek')

        try:
            user = CustomUser.objects.get(username=username)
        except CustomUser.DoesNotExist:
            return Response({'error': 'User not found'}, status=status.HTTP_404_NOT_FOUND)

        user.sex = sex
        user.age = age
        user.height = float(height) if height else None
        user.weight = float(weight) if weight else None
        user.fitness_level = fitness_level
        user.injuries = injuries
        user.preferred_workout_times = preferred_workout_times
        user.available_equipment = available_equipment
        user.has_plan = has_plan
        user.plan_week = plan_week
        user.save()

        user_data = {
            'username': username,
            'fitnessLevel': fitness_level,
            'age': age,
            'injuries': injuries,
            'availableEquipment': available_equipment,
            'height': float(height),
            'weight': float(weight),
            'preferredWorkoutTimes': preferred_workout_times,
        }

        plan = generate_plan(user_data)
        return Response({"message": "User profile updated and plan generated successfully", "plan": plan},
status=status.HTTP_200_OK)
```

Σχήμα 4.3: views.py

Με παρόμοια λογική δομούνται και οι υπόλοιπες λειτουργίες της εφαρμογής.

Μόλις τα δεδομένα του χρήστη φτάσουν στο backend, το `serializers.py` είναι υπεύθυνο για τη μετατροπή πολύπλοκων τύπων δεδομένων, όπως τα μοντέλα του Django, σε μορφή JSON, ώστε τα δεδομένα να μπορούν να μεταδοθούν εύκολα μέσω του δικτύου. Αυτή η μετατροπή διασφαλίζει τη συμβατότητα μεταξύ του frontend (βασισμένου στη JavaScript) και του backend (βασισμένου στην Python). Το **`serializers.py`** χειρίζεται επίσης το deserialization, μετατρέποντας τα εισερχόμενα δεδομένα JSON σε αντικείμενα Python, τα οποία μπορεί να επεξεργαστεί το backend. Σε αυτή την περίπτωση, θα μετατρέψει τα δεδομένα POST σε πεδία που μπορούν να αποθηκευτούν στο μοντέλο `CustomUser`.

Το backend περιλαμβάνει όλα τα απαραίτητα αρχεία για τη σωστή λειτουργία του νευρωνικού δικτύου, όπως είναι τα αρχεία `excel` που περιέχουν τα `datasets` που χρησιμοποιούνται για το γέμισμα της βάσης αλλά και για την εκπαίδευση του μοντέλου, και τα απαραίτητα `scripts` για τις μικρές εργασίες που καθιστούν εύκολη την αναπροσαρμογή του μοντέλου.

Τέλος, το **`settings.py`** παίζει κρίσιμο ρόλο συνδέοντας τα πάντα μεταξύ τους. Για παράδειγμα, διαχειρίζεται τις ρυθμίσεις για τη χρήση του SQLite ως βάση δεδομένων και ορίζει εγκατεστημένες εφαρμογές όπως το REST Framework του Django, το οποίο απλοποιεί τη δημιουργία και σειριοποίηση API. Αυτή η ρύθμιση εξασφαλίζει την ροή δεδομένων μεταξύ των `api.js`, `views.py` και του μοντέλου μηχανικής μάθησης (ANN), διατηρώντας παράλληλα μια ασφαλή και λειτουργική δομή εφαρμογής.

Η συνεργασία όλων των παραπάνω συνιστούν ένα ισχυρό backend σύστημα το οποίο είναι φιλικό προς τον προγραμματιστή, παρέχοντάς του όλα τα εργαλεία ώστε να διαχειριστεί με ευκολία τα δεδομένα της βάσης, αλλά και τις κλάσεις και συναρτήσεις οι οποίες τα τροποποιούν.



## **Κεφάλαιο 5: Συμπεράσματα**

### **5.1 Περιπτώσεις χρήσης**

Η εφαρμογή έχει σχεδιαστεί για να προσφέρει εξατομικευμένα σχέδια προπόνησης που προσαρμόζονται στις ατομικές ανάγκες. Είναι ιδιαίτερα χρήσιμη για τους χρήστες που αναρρώνουν από τραυματισμούς ή διατηρούν τη φυσική τους κατάσταση ενώ διαχειρίζονται παλιούς τραυματισμούς. Παρακάτω παρουσιάζονται οι κύριες περιπτώσεις χρήσης:

- **Αποκατάσταση τραυματισμών που προκλήθηκαν πρόσφατα**

Ένας χρήστης που έχει υποστεί πρόσφατα έναν τραυματισμό σχετικό με τους τρεις που περιλαμβάνονται (Πρόσθιος χιαστός, Μηνίσκος, Αχίλλειος τένοντας), είτε είναι μικρότερη ζημιά, ή περιλαμβάνει ρήξη που απαιτεί χειρουργείο, χρειάζεται ένα σχέδιο προπόνησης που να υποστηρίζει την αποκατάσταση, ελαχιστοποιώντας παράλληλα τον κίνδυνο επιδείνωσης του τραυματισμού, πάντα με τη συναίνεση του γιατρού του. Εισάγοντας λεπτομέρειες για το φυσικό του προφίλ, ο χρήστης αξιοποιεί το σύστημα το οποίο δημιουργεί ένα εξατομικευμένο σχέδιο προπόνησης που περιλαμβάνει ασφαλείς ασκήσεις που προάγουν την αποκατάσταση. Καθώς ο χρήστης προχωράει, η εφαρμογή τροποποιεί το σχέδιο ώστε να εισάγει ασκήσεις σε υψηλότερες εντάσεις, διασφαλίζοντας παράλληλα την ασφαλή αποκατάσταση.

- **Διαχείριση ενός παλιού τραυματισμού**

Ένας χρήστης με προηγούμενο τραυματισμό, ο οποίος πέφτει στις κατηγορίες που αναφέρθηκαν προηγουμένως, που θέλει να διατηρήσει τη φυσική του κατάσταση και παράλληλα να αποτρέψει την επανάληψη του τραυματισμού μπορεί να χρησιμοποιήσει την εφαρμογή. Εάν κάποιο άτομο αρχίζει να ξανανιώθει πόνο σε σημείο που είχε τραυματίσει παλιότερα και έχει συμβουλευτεί τον γιατρό του, ή έχει αναρρώσει πλήρως απο τον τραυματισμό αλλά θέλει να διασφαλίσει ότι μένει σε καλή μυϊκή κατάσταση, μπορεί να βασιστεί στην εφαρμογή. Το πλάνο που παράγεται επικεντρώνεται στην ενδυνάμωση των υποστηρικτικών μυών και στη βελτίωση της ευελιξίας, διατηρώντας παράλληλα τη συνολική φυσική κατάσταση, εξασφαλίζοντας μακροπρόθεσμη ασφάλεια.

Σε οποιαδήποτε περίπτωση, η προσαρμοστικότητα της εφαρμογής επιτρέπει στον χρήστη να αξιοποιήσει τις δυνατότητες με όποιον τρόπο επιθυμεί. Ένας χρήστης μπορεί να βρίσκει ορισμένες ασκήσεις άβολες ανάλογα με τον τραυματισμό του ή να τις θεωρεί αναποτελεσματικές για την αποκατάσταση. Εάν αναφερθεί οποιαδήποτε δυσφορία το πλάνο ξαναχτίζεται με βάση την ανατροφοδότηση του χρήστη.

## 5.2 Μελλοντικές Επεκτάσεις

Η παρούσα εφαρμογή έχει μεγάλες δυνατότητες ανάπτυξης, με πολλές βελτιώσεις που έχουν προγραμματιστεί για την καλύτερη εξυπηρέτηση των χρηστών και τη βελτίωση της συνολικής εμπειρίας. Για τους χρήστες που αναρρώνουν από τραυματισμούς, η εφαρμογή θα μπορούσε να εισαγάγει μια λειτουργία αποκατάστασης που θα ενσωματώνει ιατρικές συμβουλές, προσφέροντας δομημένη καθοδήγηση από φυσιοθεραπευτές παράλληλα με τις τρέχουσες συστάσεις προπόνησης. Αυτό θα ενέπνεε ακόμα μεγαλύτερο επίπεδο εμπιστοσύνης στους χρήστες, καθώς οι έγκυρες προτάσεις κατάλληλων ασκήσεων θα συνοδεύονταν από επαγγελματικές απόψεις ειδικών. Επίσης, καθιστάται δυνατή η ενσωμάτωση της εφαρμογής σε γενικά πλάνα φυσικοθεραπείας, αφού ένα τέτοιο βοήθημα θα μπορούσε να αποτελέσει χρήσιμο εργαλείο στα χέρια ενός εξειδικευμένου ιατρού που βοηθάει τους ασθενείς του να ξεπεράσουν τραυματισμούς. Έτσι, η χρήση της παρούσας εφαρμογής θα μπορούσε να αναβαθμιστεί από μία casual λύση στα χέρια απλών χρηστών σε ένα πιο εξειδικευμένο εργαλείο που ελέγχεται από ιατρούς και κέντρα φυσικοθεραπείας, ώστε να διευκολυνθεί η αποτελεσματικότητα του recovery των ασθενών τους.

Μία χρήσιμη προσθήκη θα ήταν η πιο λεπτομερής χειροκίνητη προσαρμογή του προγράμματος από τον χρήστη. Αυτό θα περιλάμβανε την εισαγωγή συγκεκριμένων ασκήσεων που περιλαμβάνονται στην βιβλιοθήκη, αντικαθιστώντας αυτές με ορισμένες ασκήσεις στο ήδη υπάρχον εβδομαδιαίο πλάνο. Στην παρούσα φάση, η εξατομίκευση της εφαρμογής εξαρτάται καθαρά από τα στοιχεία που εισάγει ο χρήστης για τη φυσική του κατάσταση, τα οποία διαχειρίζεται το νευρωνικό δίκτυο. Αυτό θα προσέθετε άλλο ένα επίπεδο εξατομίκευσης της εμπειρίας, αυξάνοντας την ελευθερία της προσαρμογής του πλάνου γυμναστικής. Αυτή η λειτουργία, όμως, θα πρέπει να γίνεται με την ευθύνη του χρήστη, καθώς παρότι θα προτεινόταν σχετικές εναλλακτικές, ο χρήστης μπορεί να εισάγει κάποια άσκηση η οποία δεν είναι απολύτως κατάλληλη για την περίπτωσή του.

Επιπλέον, η ενσωμάτωση περισσότερων ασκήσεων στο σύστημα θα είχε ως αποτέλεσμα την υποστήριξη την αποκατάστασης κι άλλων τραυματισμών. Η αποτελεσματική δημιουργία πλάνων γυμναστικής απαιτεί μία ευρεία γκάμα ασκήσεων γυμναστικής καθώς είναι απαραίτητη η κλιμάκωση της δυσκολίας αυτών για να υπάρχει εξέλιξη της φυσικής κατάστασης του χρήστη. Ως επέκταση αυτού, η ενσωμάτωση περισσότερων ασκήσεων θα μπορούσε να έχει ως αποτέλεσμα την δημιουργία πιά γενικών πλάνων γυμναστικής, όπως για περιπτώσεις που ένας χρήστης θέλει να βελτιώσει τη φυσική κατάστασή του ανεξάρτητα από κάποιον τραυματισμό, είτε είναι μέσω της αύξησης της μυϊκότητας, ή μέσω της απώλειας βάρους. Αυτές οι δύο επιλογές θα μπορούσαν να περιλαμβάνουν και εξατομικευμένα πλάνα διατροφής, προσαρμοσμένα στις ανάγκες και τις προτιμήσεις του κάθε χρήστη.

Τέλος, η υποστήριξη μίας λειτουργίας progress tracking θα μπορούσε να βελτιώσει το user engagement. Ο χρήστης θα έχει τη δυνατότητα να παρακολουθεί την πρόοδο του μέσω γραφημάτων αναπαράστασης, βλέποντας πως κυμαίνονται ανά τον χρόνο τα κιλά του, η επίδοσή του στις ασκήσεις και η αποκατάστασή του.

### **5.3 Συμπεράσματα**

Το παρόν fitness app αποτελεί ένα βήμα στην εξέλιξη της εξατομίκευσης των πλάνων γυμναστικής με χρήση τεχνητής νοημοσύνης με συγκεκριμένο σκοπό την αποκατάσταση τραυματισμών. Προσφέροντας σχέδια προπόνησης προσαρμοσμένα με βάση το προφίλ του χρήστη, παρέχει μια εμπειρία που όχι μόνο υποστηρίζει τους γενικούς στόχους γυμναστικής, αλλά ανταποκρίνεται και στις μοναδικές ανάγκες των ασθενών. Οι χρήστες μπορούν να επιστρέψουν με ασφάλεια στη σωματική δραστηριότητα μετά από πρόσφατους τραυματισμούς ή να διατηρήσουν τα επίπεδα της φυσικής τους κατάστασης παρά τα χρόνια προβλήματα, ελαχιστοποιώντας παράλληλα τον κίνδυνο νέου τραυματισμού. Μέσω της φιλικής προς το χρήστη διεπαφής, της ισχυρής υποστήριξης των νευρωνικών δικτύων και των προσαρμογών σε πραγματικό χρόνο με βάση την ανατροφοδότηση του χρήστη, η εφαρμογή δίνει τη δυνατότητα στους χρήστες να αναλάβουν τον έλεγχο του ταξιδιού ανάκτησης της παλιάς τους φόρμας.

Όσον αφορά τον μακροπρόθεσμο αντίκτυπο, αυτή η εφαρμογή θα μπορούσε να φέρει επανάσταση στο μέλλον της αποκατάστασης τραυματισμών, προσφέροντας οικονομικά προσιτές και προσβάσιμες λύσεις σε ένα πολύ ευρύτερο κοινό. Έχει τη δυνατότητα να μειώσει την εξάρτηση από προσωπικές συνεδρίες φυσιοθεραπείας, παρέχοντας δομημένα, επιστημονικά τεκμηριωμένα προγράμματα άσκησης που προσαρμόζονται σε πραγματικό χρόνο με βάση την πρόοδο του χρήστη. Αντίθετα με αυτό όμως, η εξέλιξη αυτής της τεχνολογίας θα μπορούσε να λειτουργήσει ως

ένα πολύ χρήσιμο συμπλήρωμα στην διάθεση των εξειδικευμένων επαγγελματιών που βοηθούν τους ασθενείς τους. Εν τέλει, αυτό θα μπορούσε να οδηγήσει σε ταχύτερες, πιο αποτελεσματικές ανακάμψεις, μειώνοντας τις δαπάνες υγειονομικής περίθαλψης. Με την ενσωμάτωση χαρακτηριστικών όπως οι συμβουλές εμπειρογνομόνων και η διεύρυνση των διαθέσιμων στόχων για τους χρήστες, η εφαρμογή θα μπορούσε να οδηγήσει σε ένα μέλλον όπου τα εξατομικευμένα προγράμματα αποκατάστασης θα είναι ευρύτερα διαθέσιμα, βελτιώνοντας τη συνολική υγεία των χρηστών παγκοσμίως.

## **Κεφάλαιο 6: Βιβλιογραφία-Αναφορές-Διαδικτυακές Πηγές**

### **6.1 Αναφορές που χρησιμοποιούνται στη διατριβή**

- Johns Hopkins Medicine. (n.d.). Torn Meniscus. Retrieved from <https://www.hopkinsmedicine.org/health/conditions-and-diseases/torn-meniscus>
- Mayo Clinic. (n.d.). ACL Injury Symptoms & Causes. Retrieved from <https://www.mayoclinic.org/diseases-conditions/acl-injury/symptoms-causes/syc-20350738>
- Johns Hopkins Medicine. (n.d.). Achilles Tendon Injuries. Retrieved from <https://www.hopkinsmedicine.org/health/conditions-and-diseases/achilles-tendon-injuries>
- Towards Data Science. (2021). Activation Functions in Neural Networks. Retrieved from <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
- V7 Labs. (2021). Neural Networks Activation Functions. Retrieved from <https://www.v7labs.com/blog/neural-networks-activation-functions>
- Google Developers. (n.d.). Neural Networks Backpropagation. Retrieved from <https://developers.google.com/machine-learning/crash-course/neural-networks/backpropagation>
- Baeldung. (n.d.). Epoch in Neural Networks. Retrieved from <https://www.baeldung.com/cs/epoch-neural-networks>
- Nielsen, M. (2015). Neural Networks and Deep Learning. Available at: <http://neuralnetworksanddeeplearning.com>

### **6.2 Πηγές που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής**

- ExRx.net. (n.d.). Retrieved from <https://exrx.net>
- Verywell Health. (n.d.). ACL Rehab Exercises. Retrieved from <https://www.verywellhealth.com/acl-rehab-exercises-3119304>
- ACL Tear Info. (n.d.). ACL Rehabilitation Phase 3. Retrieved from <https://acltear.info/anterior-cruciate-ligament-rehabilitation/acl-rehabilitation-phase-3/>
- Medical News Today. (n.d.). Meniscus Tear Exercises. Retrieved from <https://www.medicalnewstoday.com/articles/meniscus-tear-exercises#summary>
- JOI Online. (n.d.). Meniscus Tear Exercises. Retrieved from <https://www.joionline.net/library/show/meniscus-tear-exercises/#gsc.tab=0>

- The Life Sciences Magazine. (n.d.). 20 Exercises for Meniscus Tear Rehab. Retrieved from <https://thelifesciencesmagazine.com/20-exercises-for-meniscus-tear-rehab/>
- Sports Injury Physio. (n.d.). Exercises for Meniscus Tear Rehab. Retrieved from <https://www.sports-injury-physio.com/post/exercises-for-meniscus-tear-rehab>
- Surrey Physio. (n.d.). Top 5 Exercises for Achilles Rupture. Retrieved from <https://www.surreyphysio.co.uk/top-5/top-5-exercises-for-achilles-rupture/>
- MyHealth Alberta. (n.d.). Aftercare Information: Achilles Rupture. Retrieved from <https://myhealth.alberta.ca/Health/aftercareinformation/pages/conditions.aspx?hwid=ad1461>
- Shutterstock. (n.d.). Retrieved from [www.shutterstock.com](http://www.shutterstock.com)