UNIVERSITY OF WOLVERHAMPTON

PROJECT AND PROFESSIONALISM

(6CS007)

FINAL REPORT

FITNESS COMPANION APPLICATION

Full name : Ananda Neupane

Student ID : 2329810

Group Name : L6CG2

Submission date : May 19, 2024

Supervisor : Bipul Bahadur Pradhan

Reader : Yogesh Bikram Shah

Abstract

The Gym application is developed as a fitness platform that assists users in crafting personalized workout and nutrition plans aligning to their goals and physical characteristics. It adapts real-time pose estimation integrating PoseNet to evaluate exercises such as squats and deadlifts, delivering immediate feedback to refine users' posture and reduce injury risks and ensure efficiency. Alongside, the app provides customized nutrition plans that adapt based on user's weight goals, height, and age, promoting a balanced fitness approach. The backend is developed with Django which securely handles user data, monitors progress and supplies tailored workout and nutrition content. On the mobile front side, React Native ensures compatibility across platforms, enabling users to access workout plans, review detailed exercise guidance, and track their fitness progress. The app also includes advanced features like browsing exercises by category, equipment, and difficulty level. Furthermore, it incorporates JWT authentication for secure logins and OTP for email verification. The purpose of this project centers on blending pose estimation technology with dynamic nutrition planning within an accessible design, creating a thorough fitness companion for users of all experience levels.

Table of Contents

Table of Figures

1. Introduction

Technology keeps changing in this 21st century. This is also affecting how we stay fit, making workouts smarter and more personalized. These days, people use apps not just to exercise, but to track how different parts of their body respond over time. My Fitness Companion App aims to make fitness feel more personal, easier to stick with, and even more enjoyable. Back when the COVID-19 lockdown shuts everything down including gyms staying active becomes a real struggle. With limited space and no equipment, many look for a way to stay healthy at home without risking injury. That's where this kind of app really helps, offering structured guidance right from your living room.

Figure 1: Logo of Fit Hub

The Gym Companion Application (FitHub) tends to provide an interactive and genius experience by integrating real time pose estimation with guided exercise tracking, able to browse more than 1200 exercises accordance to the body parts, also provide a personalized meal plans for balancing diet and your exercises. Here, the gym app uses exercisedb for browsing exercises which comes with the information of your targeted muscles and instructions for the exercises you follow, and you can see a visual representation of gif image for better understanding. And according to your body weight and your goal you can receive your dietary plans, and you have sections for selecting your dietary requirements. And lastly, it utilizes PoseNet for precise tracking of movement which ensure users maintain proper posing prevent musculoskeletal injuries and helps in maximizing the workout efficiency.

Besides other conventional fitness applications which deliver the static guidelines, this one provides and will provide you kinetic feedback for more exercises in future making your home and gym workouts more productive and engaging. This program helps users to find exercises under desired machines, free weight and equipment and another is the dietary plants based on user BMI and their own dietary choices.

1.1. Project Briefing

1.1.1. Overview of the project

Though fitness applications are evolving, many do not offer real-time payer guidance and therefore put users at risk for poor forms leading to injury. This gym practice app is the one that is designed to close this gap by providing in-depth feedback on the exercise posture and movement using AI-powered pose estimation in real-time. But that's not all the app even offers personalized nutrition plans, considering individual body types, fitness targets, and dietary preferences, to ensure a comprehensive focus on health and fitness.

The application is geared towards users of all fitness levels including beginners, athletes, and gym-goers. The application uses PoseNet to identify the main movements of the body during exercises such as squats, deadlifts, and other activities, examine user posture in real time, provide corrective suggestions, and assist users performing movements to improve performance. This allows them to feel confident while exercising, knowing that their movements are being executed correctly and effectively.

1.1.1.1. Understanding the Problem

One of the prime significant hurdles in self-oriented fitness journeys is of maintaining the correct posture during performing exercises. Most fitness applications depends on pre-recorded videos or step-by-step internet guides, which do not give users real-time clarification for their mistakes or guides them for betterment. The possible development of poor exercise habits inhibits progress in the absence of personalized feedback.

In addition, fitness is more than just doing workouts, it is more than diets as well. Most fitness applications do not consider nutrition planning, which inhibits the user from accessing a structured diet strategy to complement their training. By complementing exercise tracking with personalized meal planning, this application completes the experience for the user. Nutrition is just as important for muscle recovery, energy, and

overall readiness and this application ensures users have dietary suggestions, in addition to workouts.

1.1.1.2. Solution to the Problem

Gym application is designed to combine all in one fitness assistant that aid used train effectively and balance dietary intake along way. Some of the standout features involve:

I. Real-time Pose Estimation

With the use of PoseNet, the app tracks a user's movements, locates joint position, and compares them to the ideal form, giving real-time feedback to make any necessary corrections. This increases workout safety and efficacy by helping users stay in proper form while exercising.

II. Posture Correction Assistance

The system identifies prevalent problems like incorrect knee position, poor back posture, and squat depth, encouraging users to correct these errors to improve their techniques and minimize injury risk.

III. Workout Tracking and Insights

Users can track their progress on workouts, assess improvements in their technique, and visualize their fitness journey over time. This surveillance can boost motivation for improvement and enhance long-term progress.

IV. Exercise Database

With a full library of exercises separated by body part, difficulty level, and equipment used, the app is organized to help users find the right workouts based on their available resources and fitness level so they can act on their ideas and see progress

V. Personalized Nutrition Plans

The app provides users with a customized meal plan that matches each user's weight, height, age and goals. Since fitness includes not only exercise, but nutrition too, it is very

important for users to receive quality nutrition, which will lead them to generate better results and have a more consequential fitness experience.

VI. Adaptive Workout Recommendations

By making exercise suggestions based on user progress and performance, the system is constantly asking the user to challenge themselves and improve.

VII. Home Fitness Revolution

Utilizing PoseNet to provide feedback, the application leverages new technology to create an advanced workout analysis experience at home and for users who do not have the financial ability or desire to pay for a gym.

With these characteristics, the application builds a smarter fitness experience that acts as a personal trainer by guiding users towards their goals safely and efficiently with live insights.

In conclusion, this gym application has set a new way for users to experience fitness by combining AI-powered pose estimation and real time feedback with personalized nutrition plans.

1.1.2. AI implementation

Yes, my gym app project uses AI. Specifically, it brings in a convolutional neural network, kind of like the PoseNet setup Kendall and his team talked about in their 2015 ICCV paper, to measure your posture while you're working out in real time.

1.1.2.1. Computer Vision

The main AI utilizing in my project is Computer Vision. Computer vision is comparable to teaching machines to see and interpret a still or video image. It involves the use of computer models and/or algorithms to interpret visual data, identify items that can be detected in the visual data, and interpret the environment. The app looks at what the camera sees to figure out where your body joints are. That visual info is what it uses to check your posture.

1.1.2.2. Supervised learning

For figuring out your posture, it's mostly running on supervised learning. Here's the deal:

I. Learned from Examples (Kendall's network)

Think of it this way – just like (Kendall, et al., 2015) network learned where a camera was by seeing tons of labeled pictures, a PoseNet model for body joints learns by seeing loads of images where people's joints are marked.

II. Mapping What It Sees to What It Knows

The network figures out how to connect what it sees in the camera feed to where it thinks your joints should be. It learned this connection from all those labeled examples.

III. Just Using What It Already Knows

My app isn't constantly learning new stuff about posture from your workouts. It's using the knowledge that the pre-trained PoseNet model already has.

So, while there might be some great level of calculations behind-the-scenes stuff in the model, the core of how it figures out your posture is based on the patterns it learned from labeled data that's supervised learning in action.

1.1.2.3. Mathematical Flow behind AI

Alright, so the math that makes this posture tracking work (and it's similar to what's going on under the hood in Kendall's PoseNet) involves a few key ideas:

I. Pictures as Numbers (Linear Algebra)

Images get turned into number grids, and your joint positions are also sets of numbers. All the calculations the network does are basically math with these numbers.

II. Learning by Trying (Calculus)

When the model is learning, it makes guesses about where your joints are. Calculus helps it figure out how wrong those guesses are and how to tweak its internal settings to make better guesses next time. It's like fine-tuning to get it just right.

III. Figuring Out What's Likely (Probability and Statistics)

The model doesn't just say "your elbow is here." It might give you a sense of how confident it is about that guess. That involves ideas from probability. Now, the brains of the operation is a Convolutional Neural Network (CNN) – that's the kind of model Kendall and his team used. Here's a simplified look at how it rolls:

Firstly, your phone’s camera sends a frame of you doing your exercise in form of tensors.

I. Looks for Features (Convolutional Layers)

The CNN has layers that act like little detectors, looking for edges, shapes, and other visual clues that tell it where your body parts might be.

II. Simplifies Things (Pooling Layers)

These layers make the information a bit less overwhelming, focusing on the important stuff.

III. Guess the Spots (Prediction Layers)

Based on all the features it found, the network makes a guess about where each key body joint is. It might do this by creating "heatmaps" that show how likely a joint is to be in a certain area.

IV. Pinpoints the Location

Finally, the app takes those heatmaps (and maybe some extra info) to figure out the exact (x, y) coordinates of your shoulders, knees, and so on.

So, even though my app is tracking your body and Kendall's was tracking a camera, the underlying tech – using CNNs to learn from visual data – has a lot in common.

1.1.2.4. Agent Description

The PoseNet model inside my app acts as a visual perception agent. Its whole job is to look at you through the camera and figure out your pose. Kind of like how Kendall's network figures out where a camera is by looking at its surroundings. Here's what it does:

I. Keeps an Eye Out: It's constantly watching the video feed from your camera.
II. Analyzes What It Sees: It uses its pre-trained neural network to make sense of the image and spot your key body joints.
III. Reports What It Finds: It spits out the coordinates of those joints. Then, the rest of the app can take that info and tell you if your form is looking good.

It's basically a silent observer utilizing the user's camera that gives the app the raw data of coordinates which are needed for understanding your movements.

1.2. Aims

I. Enhance Exercise Effectiveness

Create a mechanism that guides users to perform exercises in the correct and most efficient manner by giving real-time feedback on their posture and movement.

II. Ensure Safety

When Exercising prevents injuries by using AI-powered pose estimation to identify incorrect posture when working out and provide corrective feedback.

III. Provide Personalized Nutrition Planning

Provide nutrition plans to ensure users receive personalize meal recommendation which support their fitness level and training goals.

IV. Improve User Engagement

Deliver an engaging and gamifying user experience by providing exercise options with classifications like equipment and muscle groups target.

V. Bridge the Gap Between Home and Gym Workouts

Workouts bring professional workout guidance into the home, providing users with AI-powered coaching and feedback without the need for gym membership.

VI. Support Long-Term Fitness Progression

Promote continuous fitness progression through user performance tracking, workout intensity adjustments, and adaptive recommendations.

1.3. Objectives

I. Implement AI-Powered Pose Estimation for Exercise Tracking
- Utilize PoseNet to detect body posture during squats and deadlifts.
- Ensure users maintain the correct form to minimize injury risks.

II. Provide Personalized Nutrition Planning
- Implement a system that generates customized diet plans based on user weight, height, age, and fitness goals.

III. Enhance User Engagement and Progression
- Categorize exercises by required equipment, and targeted muscle groups.
- Develop a structured progression system that helps users improve over time.

IV. Developing an Interactive and Accessible User Interface
- Design a user-friendly frontend using React Native.
- Integrate seamless navigation between exercise tracking and nutrition planning.

V. Ensure a Secure and Scalable Backend
- Implement Django for managing user authentication, exercise data, and nutrition plans.
- Ensure data security and privacy for user health metrics.
- Optimize server-side performance for real-time feedback and exercise tracking.

1.4. Artefact

1.4.1. FDD Diagram

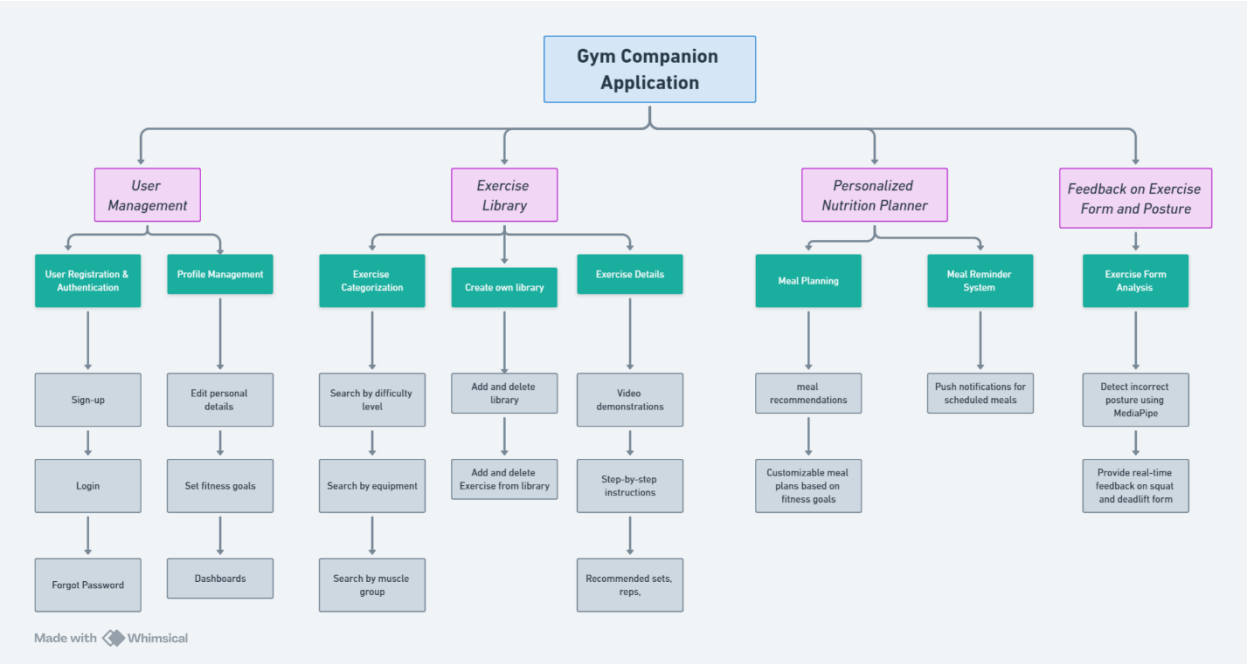


Figure 2: Functional Decomposition Diagram

The Functional Decomposition Diagram (FDD) visually represents building blocks of the Gym Companion Application, on breaking it down into core functionalities and sub-functionalities. Each subsystem is design of a specific aspect of the user's fitness journey, ensuring a steady approach to exercise tracking, nutrition planning, and real-time posture analysis.

I. User Management: Handle user registration, authentication, and profile management.
II. Exercise Library: Provide categorized exercises, allow users to create a custom library, and offer detailed exercise instructions.
III. Personalized Nutrition Planner: Generate meal plans, provide dietary recommendations, and include a meal reminder system.
IV. Feedback on Exercise Form and Posture: Uses PoseNet to analyze exercise form and provide real-time feedback on squat and deadlift movements.

This structured breakdown ensure of a clear understanding of how different modules will work together within our system.

1.4.2. Explanation of the System as a Whole

The Gym Companion Application is an AI powered fitness solution aim for providing guidance to users in their fitness journey by combining exercise tracking, personalized nutrition planning, and real-time posture correction utilizing advanced computer vision techniques. The system integrates machine learning to analyze exercise performance and delivers insights into user workouts and diet plans.

Core Functionalities of Gym Application are:

I. User Authentication & Profile Management: Secure login, profile updates, and fitness goal tracking.
II. Exercise Library & Categorization: User can explore exercises based on difficulty, equipment, and muscle groups.
III. Custom Exercise Libraries: User can create and modify their own personalized exercise lists.
IV. Step-by-Step Workout Instructions: Access to video demonstrations, recommended sets, reps, and execution guidelines.
V. Push Notifications for Meal Reminders: Help users stay consistent with their diet.

With integration of movement analysis, browsing thousands of exercises and customized nutrition planning, the system affects as a comprehensive fitness assistant for users in their mobile phones.

1.4.3. Explanation of Each Subsystem

The Gym Companion Application systems have been broken down into four main subsystems. Let’s understand them briefly.

1.4.3.1. User Management Subsystem

This subsystem ensures secure authentication and user profile management, allowing individuals to track and personalize their fitness experience.

1.4.3.1.1. User Registration & Authentication

I. Sign-Up: User can create an account using an email and password.
II. Login: Secure authentication process allowing access to the system.
III. Forgot Password: Provide password recovery functionality through OTP verification.

1.4.3.1.2. Profile Management

I. Edit Personal Details: User can update their name, age, weight, height, and fitness information.
II. Set Fitness Goals: User must define their fitness objectives, such as weight loss, muscle gain.
III. Dashboards: Display progress over time, including exercise performance and diet consumption.

1.4.3.2. Exercise Library Subsystem

This subsystem provides a structured way to explore and track exercises. Users can categorize, add, and manage workouts based on their preferences.

1.4.3.2.1. Exercise Categorization

I. Search by Equipment: Like Dumbbells, barbells, resistance bands, bodyweight exercises, etc.
II. Search by Muscle Group: Workout for chest, back, legs, arms, core, and shoulders.
III. Use libraries: Create multiple libraries of your own favorite exercises.

1.4.3.2.2. Creating Own Library

I. Add and Delete Exercise Library: User can create their custom workout lists.
II. Modify Library: Add or remove exercises from personal lists based on preferences.

1.4.3.2.3. Exercise Details

I. Video Demonstrations: Each exercise contains a tutorial video.
II. Step-by-Step Instructions: Complete instructions and guidance on how to perform each exercise correctly.
III. Calculate Sets and Reps: Record your sets, reps, and rest times based on the user's preferences.

1.4.3.3. Personalized Nutrition Planner Subsystem

This subsystem ensure meal plans on tailoring with user fitness goals, along with meal reminders to maintain a consistent diet.

1.4.3.3.1. Meal Planning

I. Meal Recommendations: Generate meal per user dietary preferences and fitness objectives.
II. Time table: Timely diet fore day.

1.4.3.3.2. Meal Reminder System

I. Push Notifications for Scheduled Meals: The system sends reminders to users to ensure they follow their meal plan consistently.

1.4.3.4. Feedback on Exercise Form and Posture Subsystem

This subsystem is AI-powered and integrates PoseNet pose estimation to analyze the user's workout posture and provide real-time feedback.

1.4.3.4.1. Exercise Form Analysis

I. Detect Incorrect Posture Using PoseNet: The system captures live footage from the user's camera and analyzes joint angles to detect incorrect form.

1.4.3.4.2. Provide Real-Time Feedback on Squat and Deadlift Form

I. Identifies improper knee, hip, and back positioning.
II. Alerts users if their squat depth is insufficient.
III. Detects incorrect back curvature in deadlifts and warns about potential injury risks.
IV. Displays corrective suggestions to improve workout performance.

1.5. Academic Questions

The academic question with my Gym Application is as:

How can AI-driven pose estimation and personalized fitness recommendations enhance the effectiveness and safety of workout routines in mobile fitness applications?

1.5.1. Explanation of the Academic Question

The upward question goals to investigate on how Artificial Intelligence (AI) and Computer Vision based pose estimation can upgrade the fitness applications by offering real-time feedback on exercise form and posture. Moreover, it also monitor how personalized recommendations of workout routines and nutrition plans can affect positively toward user consistency and effectiveness.

1.5.1.1. AI-Powered Pose Estimation for Injury Prevention

- Evaluating the accuracy and effectiveness of PoseNet-based pose tracking in identifying incorrect squat and deadlift postures.
- Analyzing how immediate feedback can prevent injuries and improve user form.

1.5.1.2. Impact of Personalized Workout and Nutrition Plans

- Examining how tailored fitness recommendations based on user height, weight, age, and goals contribute to consistency and results.
- Assessing whether personalized meal plans and reminders improve consistency in diet tracking.
- Investigating how data-driven adaptations to fitness programs enhance motivation and progress tracking.
- User Engagement and Adoption of Technology-Based Fitness Solutions.

1.5.1.3. User Engagement and Adoption of AI-Based Fitness Solutions

- Identifying potential hazards to adoption, including data privacy concerns, usability challenges, or lack of trust in technology generated recommendations.

- Analyzing the impact of interactive features (such as video demonstrations, dashboards, and automated feedback) on user convenience.

1.5.1.4. Comparing AI-Powered Fitness Apps to Conventional Training Methods

- Evaluating how mobile technology focused fitness applications differ with personal trainers, gym training programs, and non-technology fitness apps.

This academic question vision is to assess the effectiveness, safety, and user adoption with technology-enhanced fitness applications. It contributes to understanding how machine learning, computer vision, and personalized analytics is able to be a more interactive, efficient, and accessible fitness experience for users globally.

1.6. Scope and Limitations

1.6.1. Scope

The Gym Companion Application provide AI-powered real-time feedback on exercise form, personalized workout plans, and nutrition recommendations. It primarily aim includes enhancing fitness tracking through pose estimation technology, meal planning, and user-specific exercise libraries.

Key Features Covered in Scope:

1.6.1.1. User Management:

I. User registration, authentication, and profile management.
II. Users can edit personal details, set fitness goals, and access dashboards for progress tracking.

1.6.1.2. Exercise Library

I. Users can search for exercises based on difficulty level, equipment, and muscle group.

II. Ability to create, add, or delete custom exercise libraries.
III. Access to video demonstrations, step-by-step instructions, and recommended sets/reps.

1.6.1.3. Personalized Nutrition Planner:

I. AI-driven meal recommendations based on fitness goals, weight, and dietary preferences.
II. Customizable meal plans and automated meal reminders via notifications.

1.6.1.4. AI-Powered Exercise Form Feedback

I. Pose estimation using PoseNet to analyze squat and deadlift forms.
II. Real-time feedback to detect incorrect posture and suggest corrections.

1.6.2. Limitations

1.6.2.1. Scope of AI Pose Estimation

I. The pose tracking model is focused only on squats and deadlifts; it does not support all types of exercises.
II. Pose estimation accuracy depends on camera quality, lighting conditions, and user positioning.

1.6.2.2. Limited Exercise Library Customization:

I. While users can create custom libraries, they cannot add completely new exercises outside of the existing database.

1.6.2.3. Nutrition Planning Constraints:

I. The nutrition module provides generalized diet recommendations, but it does not track actual food intake or caloric consumption.
II. Does not account for specific medical conditions or food allergies.

1.6.2.4. Real-Time Feedback Constraints:

I. Feedback is limited to basic corrections in angles and posture; it does not provide advanced biomechanical analysis.

II. There may be false positives or incorrect feedback in certain edge cases (e.g., partially visible body parts).

1.6.2.5. Internet Dependency:

I. Some features, such as exercise search, meal planning, and video demonstrations, require an active internet connection.
II. AI-based pose analysis is processed locally on the device, but cloud-based processing for advanced analytics is not supported.

1.6.2.6. User Engagement and Learning Curve:

I. Users need to position their cameras correctly for accurate pose tracking.
II. The system does not offer live coaching or interaction with human trainers.

This project provides an AI-driven gym companion application that enhances exercise form correction and personalized fitness planning. While it covers real-time feedback, workout tracking, and nutrition planning, some limitations exist due to hardware constraints, AI accuracy, and the scope of exercise coverage. Future iterations of versions will surely include support for additional exercises, improved AI analysis, and expanded nutritional tracking.

2. Literature Review

2.1. Review of Journal Articles

1. Accuracy Evaluation of 3D Pose Estimation with PoseNet Pose for Physical Exercises

(Sebastian, et al., 2023) dive into how accurate PoseNet Pose, a popular pose estimation tool, really is when it comes to health-critical uses like physical therapy, where getting it right matters a lot. They point out that there’s still a gap in checking if PoseNet Pose nails key medical details, like joint angles and body measurements. Their work helps pave the way for sharper, more reliable pose estimation in situations where precision and trust are non-negotiable.

2. Changes in Physical Activity and Sedentary Behaviors from Before to During the COVID-19 Pandemic Lockdown: A Systematic Review

(Stockwell, et al., 2021) dives into a huge review over 66 studies with 86,981 people, show casing how physical activity (PA) and sedentary behavior (SB) flip-flopped during the phase of COVID-19 lockdowns. The review shines a light on digital fixes like online fitness classes, which step in to soften the blow. But it also flags some holes—like spotty ways of measuring things and not enough detail on the kinds and intensity of PA people are doing.

3. Effects of a Personalized Fitness Recommender System Using Gamification and Continuous Player Modeling: System Design and Long-Term Validation Study

(Zhao, et al., 2020) introduces a gamified fitness assistant system that integrates wearable tracking and continuous player modeling to deliver personalized exercise recommendations.

In a 60-day study with 40 participants, three groups were compared: one received a fully personalized, gamified experience. Results showed that personalization and gamification significantly improved motivation and satisfaction

4. Digital Applications for Diet Monitoring, Planning, and Precision Nutrition for Citizens and Professionals: A State of the Art

(Abeltino, et al., 2024) provides an overview of digital tools design of assisting individuals and professionals in managing personalized dietary practices. Digital applications simplify these processes by offering tailored nutrition advice.

The tools are categorized by target users:

- **Citizens**: Apps for self-monitoring diets, managing weight, and controlling diabetes, featuring food logging and calorie counting but facing challenges like data accuracy and engagement.

- **Nutritionists**: Tools to create and adjust client-specific diet plans, improving efficiency in counseling.
- **Physicians/Researchers**: Platforms using genetic data to deliver precision nutrition recommendations.

AI and machine learning encourage personalization, while blockchain technology is under researching phase for secure data handling. However, challenges like accuracy, accessibility, and affordability remains yet. The article stresses the need for continue innovation to maximize the potential of digital dietary tools for betterment.

5. FitSight: Tracking and Feedback Engine for Personalized Fitness Training

(Hitesh, et al., 2024) research on FitSight, a fresh fitness setup that taps into the YOLOv7 model and a basic webcam to dish out real-time posture tips while you work out. Using computer vision, FitSight watches your moves and fires off instant, tailored text advice, helping you keep your form sharp, dodge injuries, and get more out of your session—no pro trainer neededFitSight's a leap ahead in blending human-computer teamwork with fitness tech, making top-tier coaching reachable for all kinds of users.

6. A review on Computer Vision Technology for Physical Exercise Monitoring

(Salik, et al., 2022) Shine a spotlight on how computer vision is stepping up for non-contact exercise tracking, giving traditional sensor setups a run for their money. Deep learning kicks it up a notch, crunching video data to nail down vital signs and workout stats with precision. All in all, computer vision's shaping up as a handy, reliable go-to for keeping tabs on exercise without the fuss.

2.2. Similar applications

1. MyFitnessPal

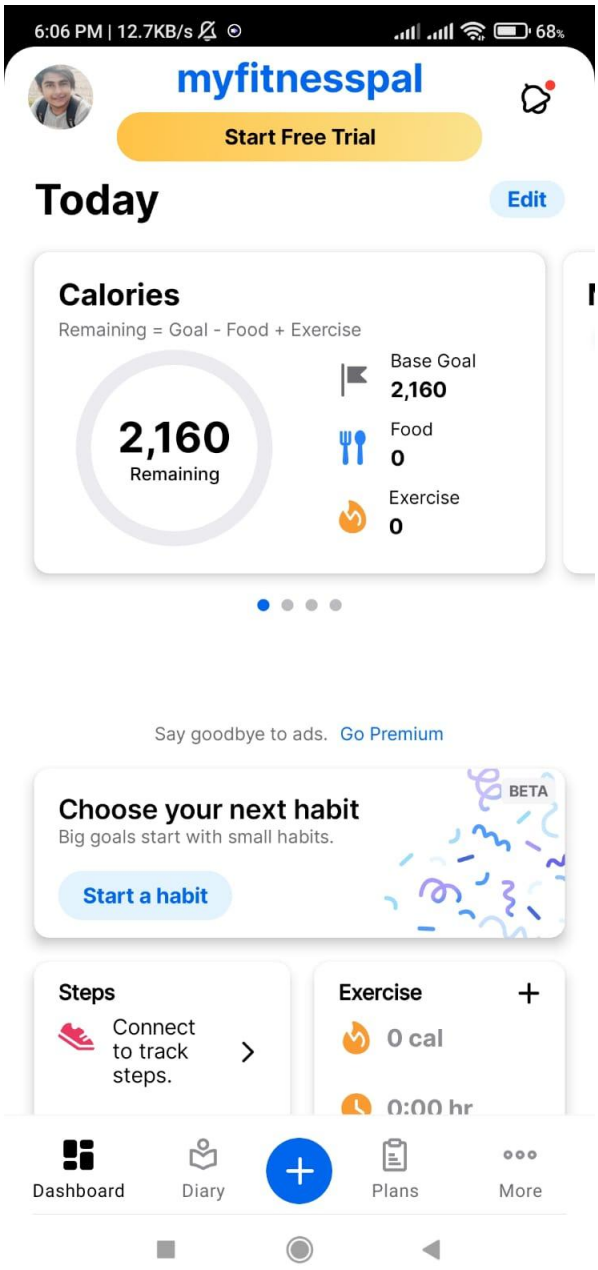


Figure 3: My fitness pal Home screen

Home Screen contains minimalist design and proper navigation are build on the footer and you can view the id status on top of screen.

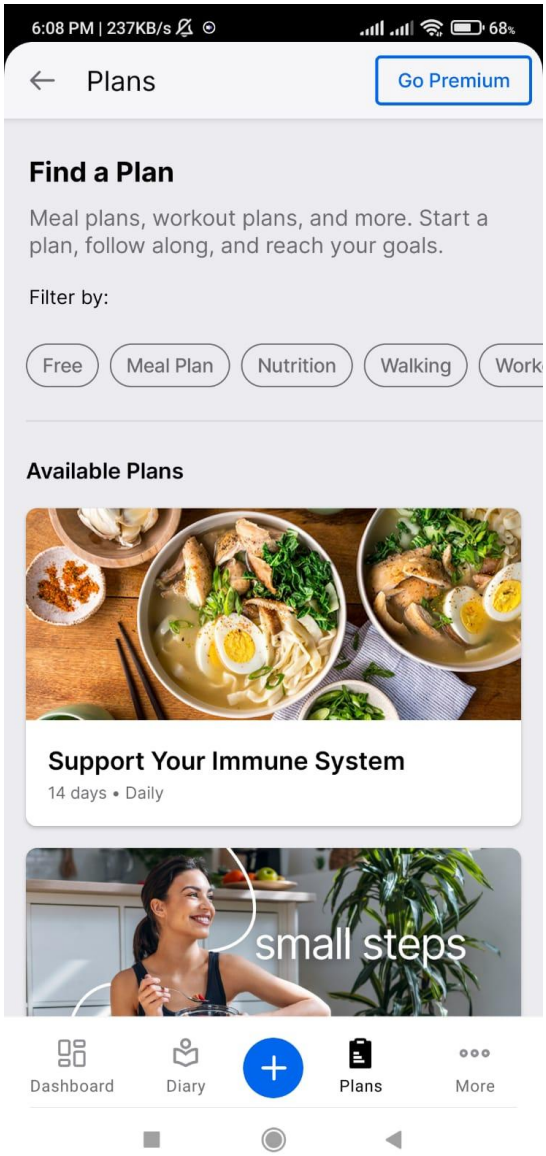


Figure 4: My fitness pal Plan screen

Plan screen provides various plan for you diet and very few are available in free. We need to buy subscription to access many features.

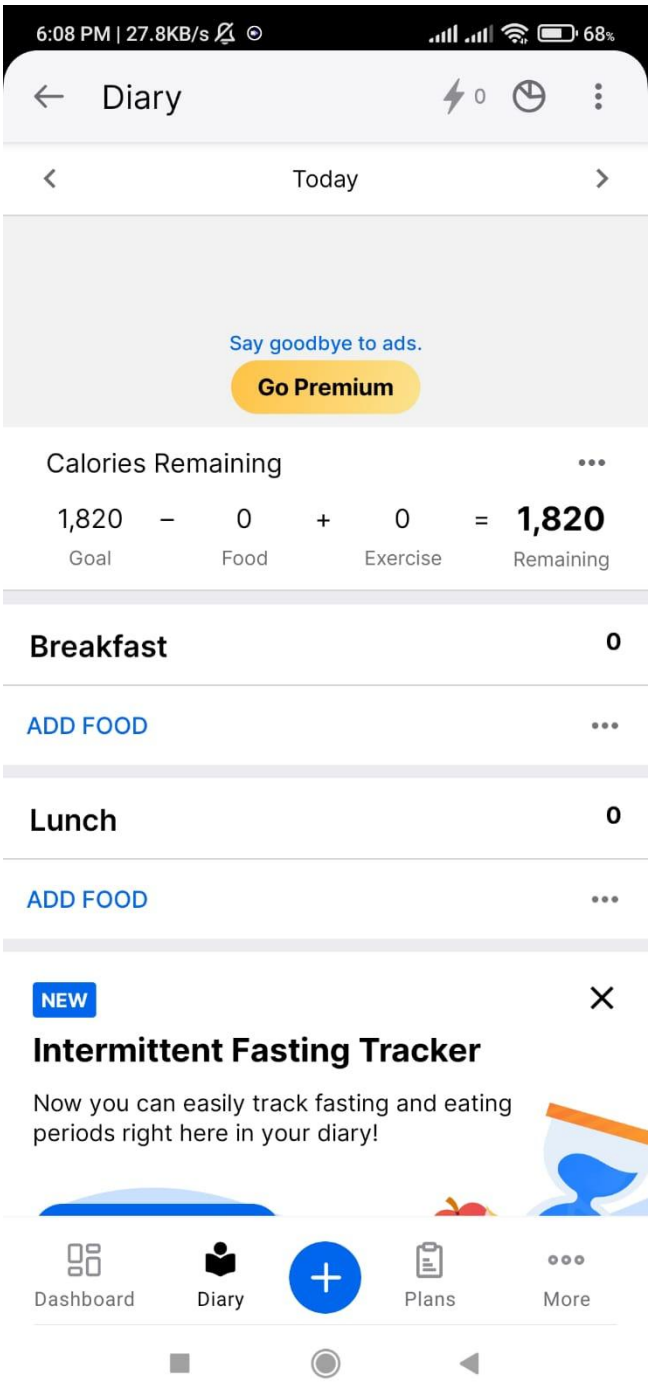


Figure 5: Fitnesss plan diary page

This page is mainly for nutrition section where you can add your food items in your plan. You can regularly see your calorie goals and the remaining calorie for your goal met.

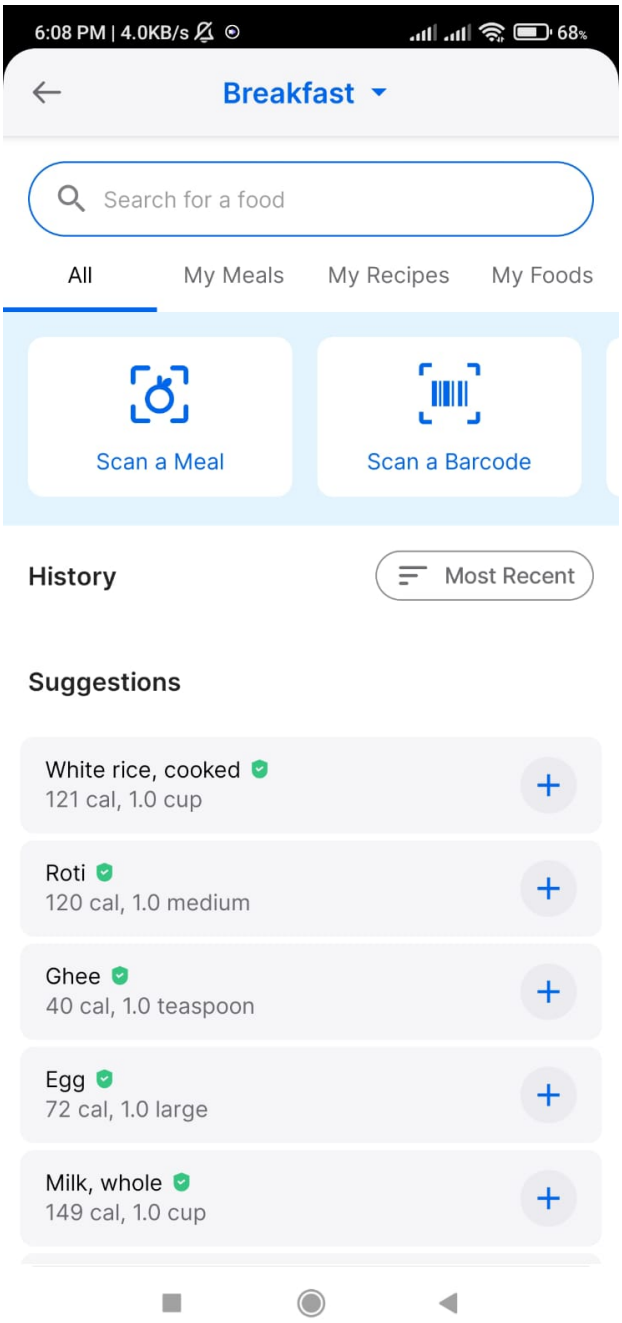


Figure 6: My fitness Pal breakfast page

MyFitnessPal (FitnessApp, 2024) is a well-known fitness app that tracks calories, meals, and weight management which aligns with its extensive database of more than 11 million food items, users can easily keep track of their meals and calories consumed. It is ideal for people who are working toward their dietary objectives because of its straightforward

design and vibrant community forums. However, MyFitnessPal only offers limited features like individualized training plans, real-time pose tracking, and gamified tools to keep users motivated.

2. Nike Training Club

A fitness app called Nike Training Club (nikw, 2024) offers top-notch exercise videos and recording offered by qualified trainers. It aligns best for users of all fitness levels, from beginner to elite athletes, because it delivers a variety of guided programs for strength, endurance, mobility, and yoga. The app offers easy-to-follow audio and video instructions for tracking workouts. Additionally, it provides customized programs for objectives like muscle growth or weight loss. In comparison to other fitness apps that provide individualized experiences, its customization options are also lacking.

3. Strong Workout Tracker Gym Log

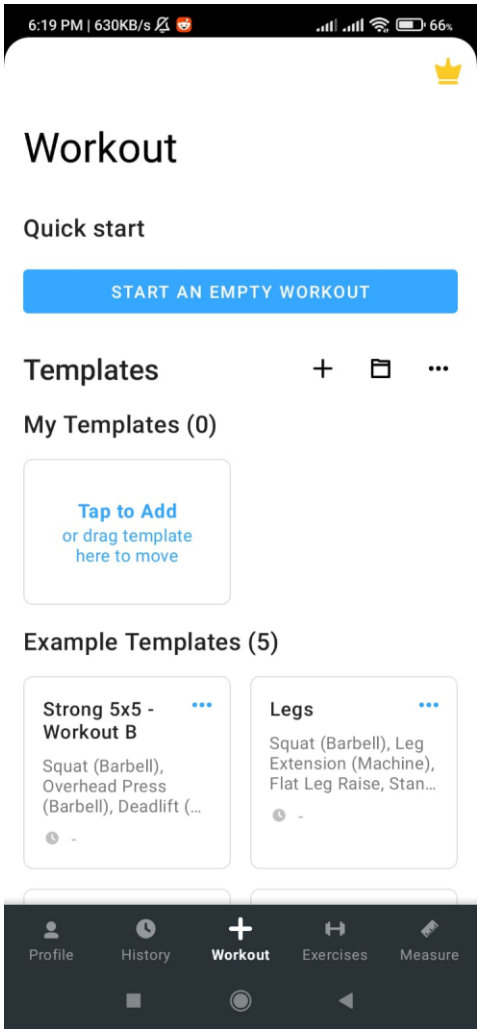


Figure 7: Strong Home page

This is home page of strong app. It is very minimalistic and not so practical. We cant identify if anything is significant. The footer navigation section is pretty well aligned.

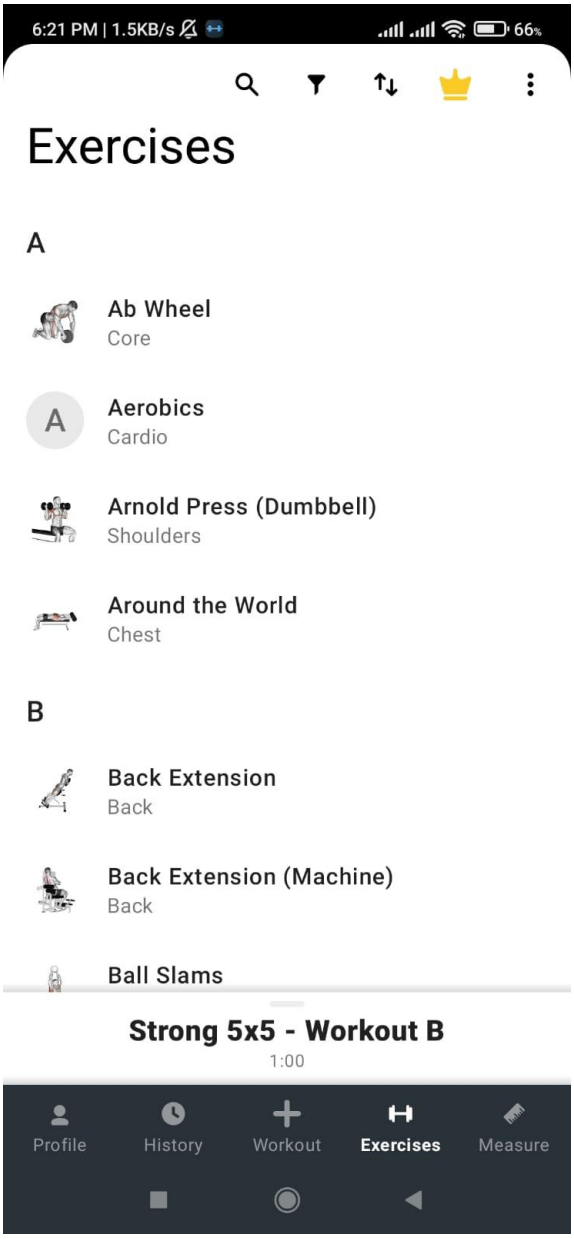


Figure 8: Strong Exercise page

This is the exercise page for strong where exercise are pretty well aligned and are more cleare to view according to alphabetical orders.

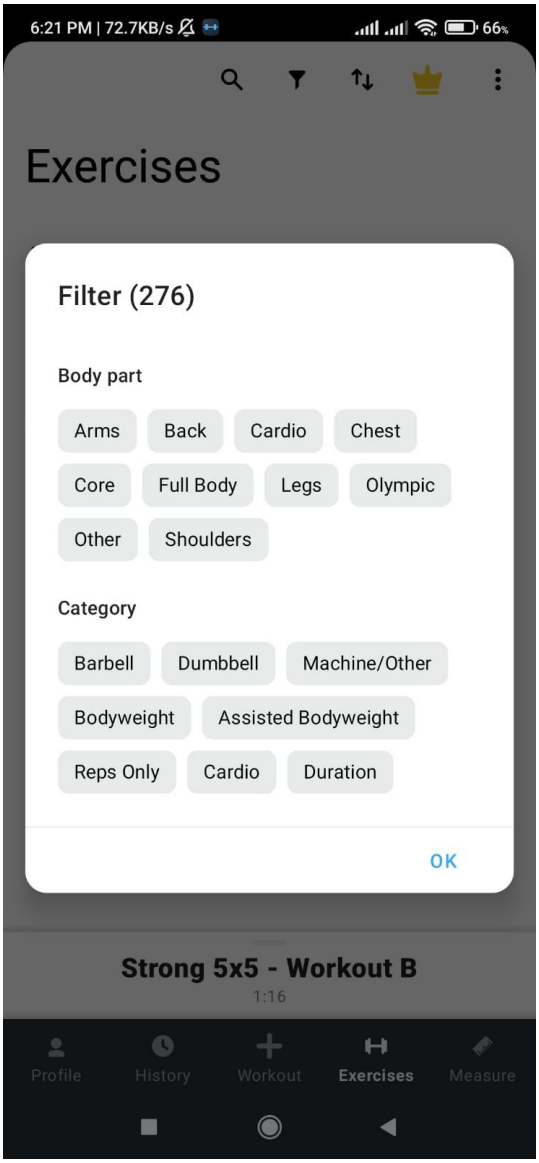


Figure 9: Exercise Filter options

These are the filter options available in the application which I feel is a strong part as we got more filters to search through either body part or the equipment we got.

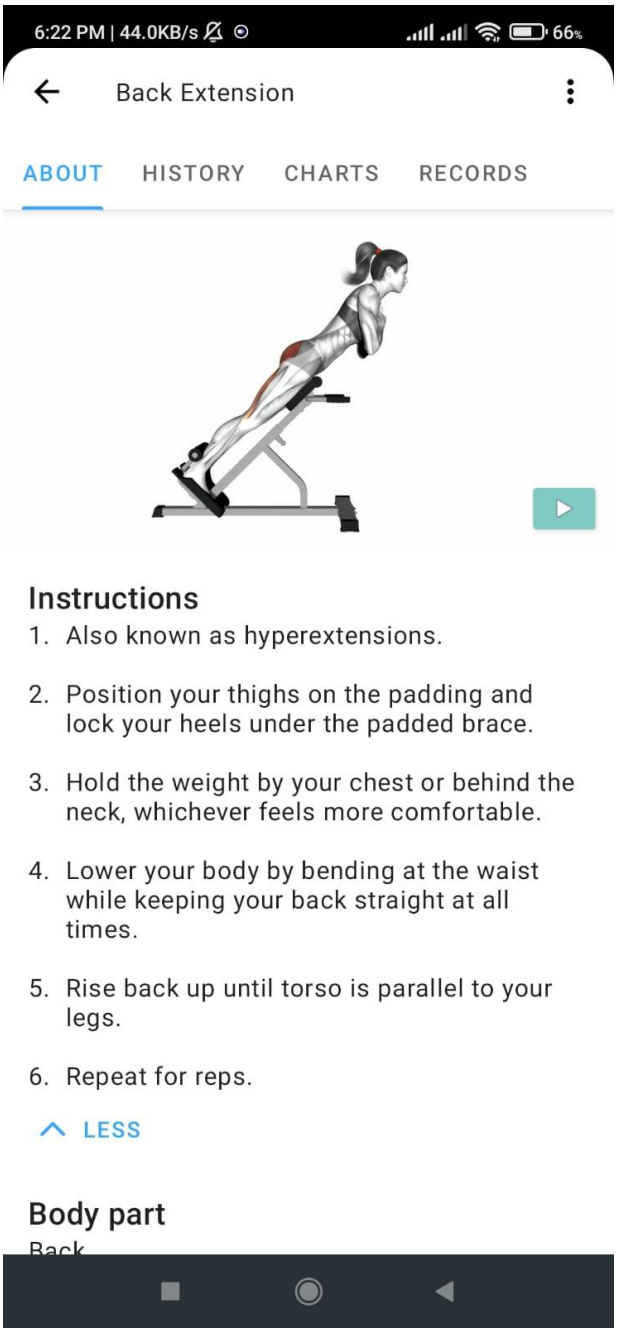


Figure 10: Strong exercise page

The exercise page is well aligned with detailed instructions. You can observe the tutorial video also. There is clear interpretation of the body part includes during exercise and equipment needed.

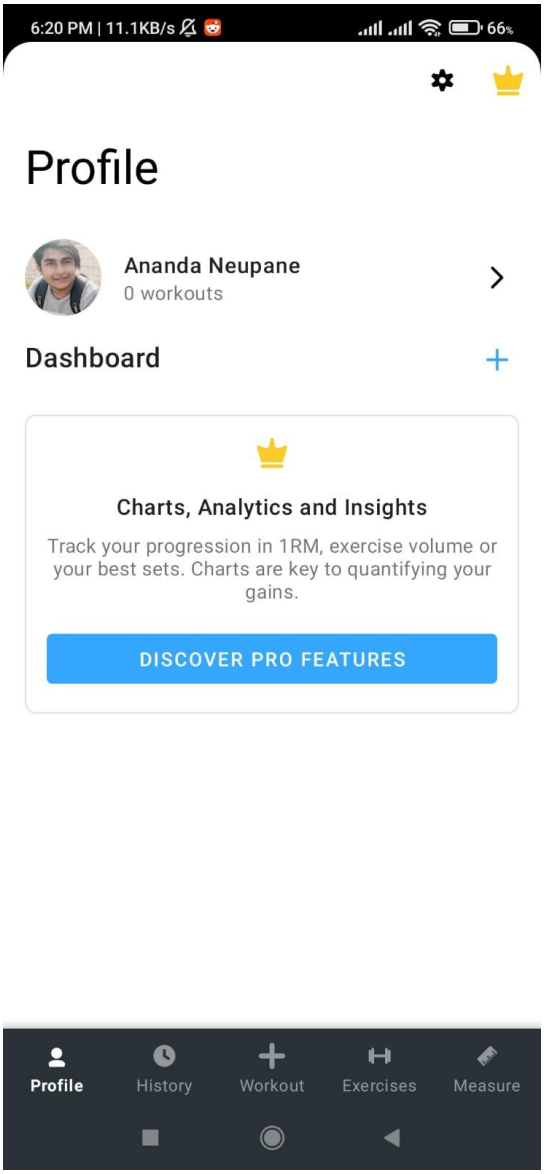


Figure 11: Strong Profile Page

This is the profile page for strong app which is a shameful experience as it feels so empty. But later on, growing with exercise performance we can add the dashboards of various options such as muscle tracking and calorie burning and how exercise affects our body.

Fitness app called Strong (Anon., 2024) fits suitable for people who rejoice in strength training and heavy weight exercises like weightlifting and resistance training.

4. FitOn

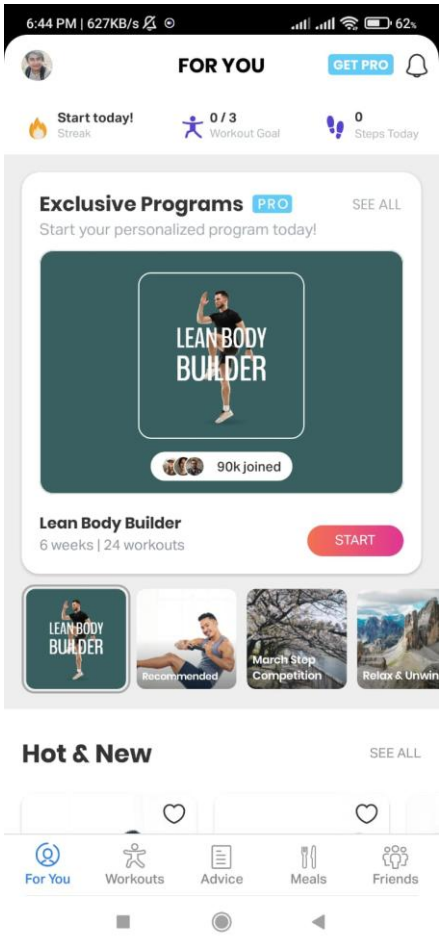


Figure 12: FitOn Homepage

This is home page of fit on application which is premium in looks and it offers us to choose for premium and free versions earlier. Here we can track the steps and visualize our goal in home page. Then we have perfect navigation bar under footer section.

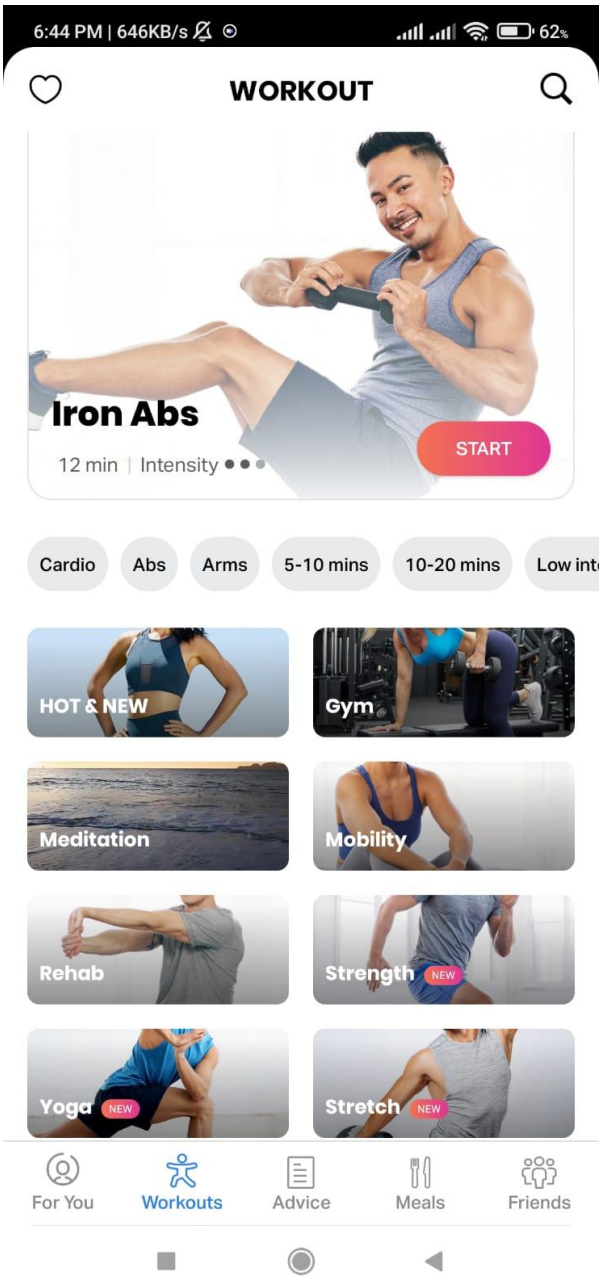


Figure 13: Fit on exercise page

We can chose for exercise category and visit different exercise inside and each exercise come with effective instructions and video guidance.

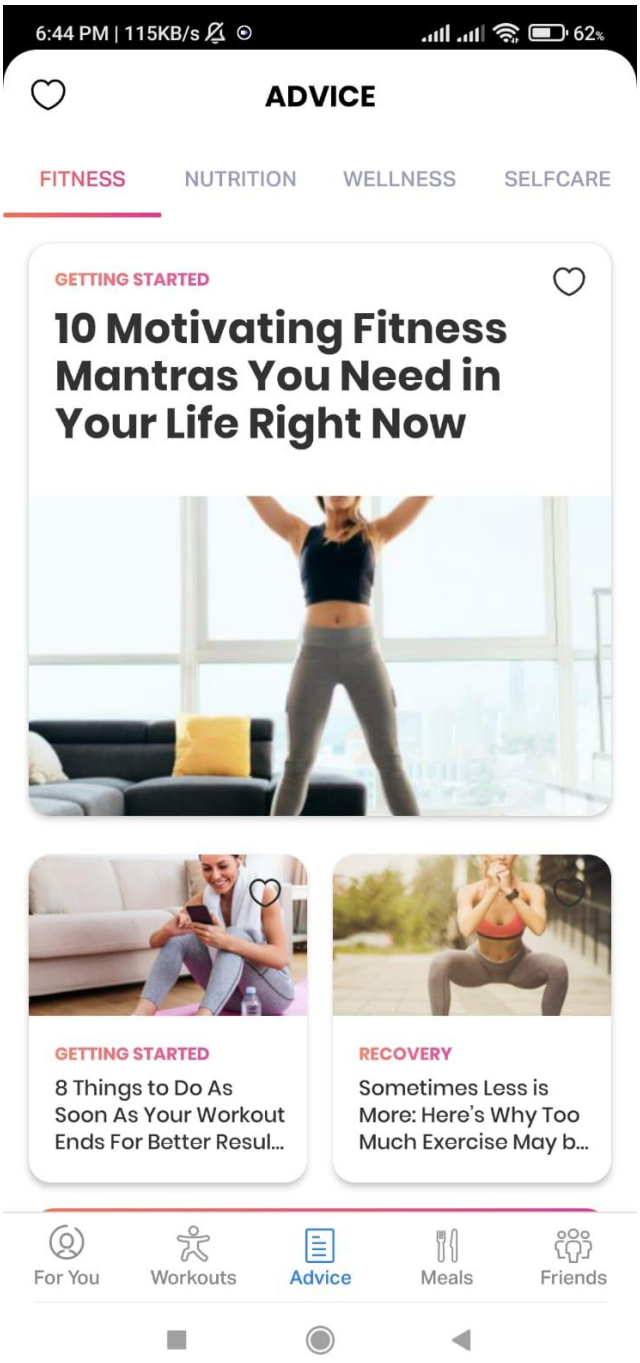


Figure 14: Fit On navigation page

This is another advice page from navigation. This pages includes of different articles found online and different blogs posted by community users.

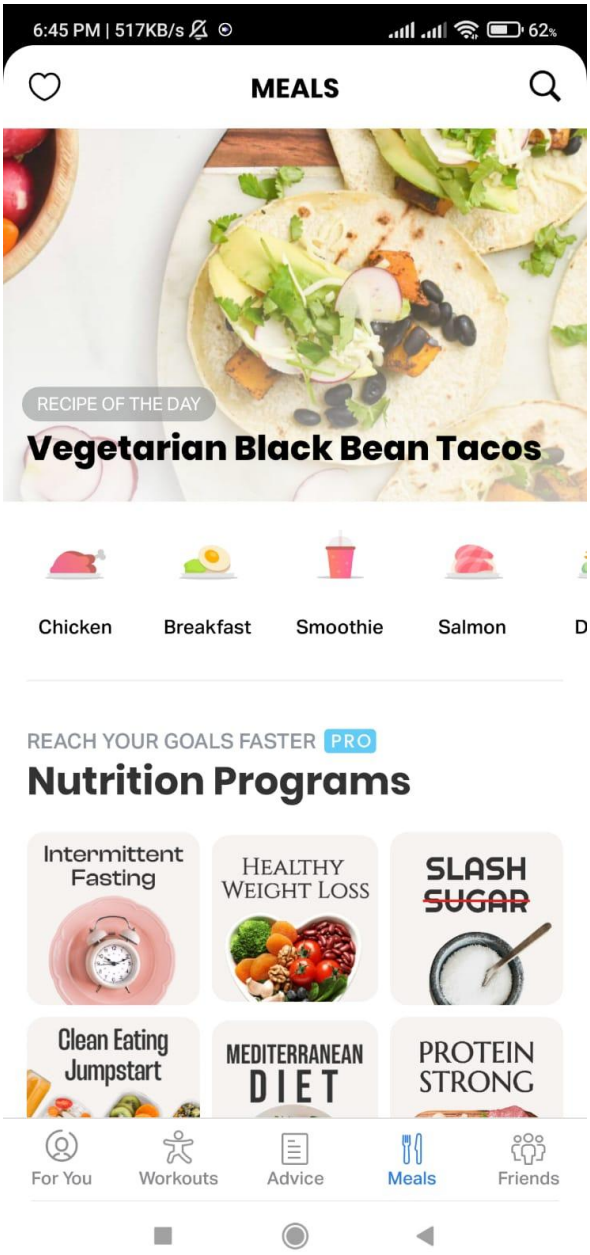


Figure 15: Fit on Meals page

Inside meals page, we can observe different categorization of the foods where on click we can browse through different items for eating and following by we can see the receipe of the day on top of it.

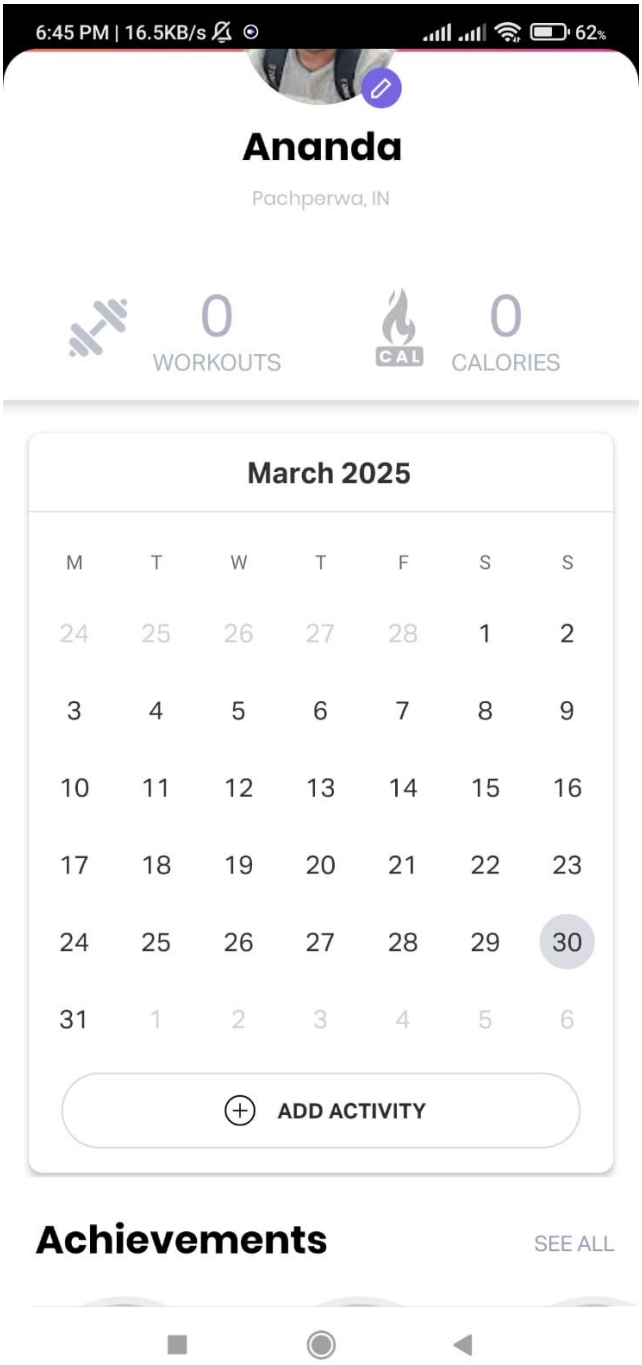


Figure 16: Fit on profile page

Fit on Profile page includes workout and calorie sections which show the count of exercise and calorie burned over the day and moving on you can view a huge calendar where you

are able to track your streak of performing exercise and lastly we can see achievement sections which are granted to you with different steaks and your accomplishments.

FitOn (On, 2024) is a mobile fitness app that welcomes users of all skill levels, delivering a wide array of workout videos and training options. To keep motivation high, FitOn adds a fun social twist, letting users challenge their buddies. Plus, people can tweak their workouts right in the app to match their personal goals and tastesStill, with its easy-to-use design, compatibility with wearable gadgets, and a vibe that brings people together, it's a go-to fitness companion for tons of users.

5. JEFIT Workout Planner

JEFIT (jefit, 2024) is a full-on fitness app crafted for folks who love hitting the gym for strength training. The JEFIT community adds a social spark, pushing accountability and friendly rivalry. While its tracking tools are solid, it doesn't bring gamified fun, built-in nutrition guidance, or live feedback on form or posture. Even so, it shines as a prime pick for strength training fans, offering a structured way to manage workouts and smart, data-backed insights.

Feature	**MyFitnessPal**	**Fit ON**	**Strong Workout Tracker**	**Fitness Companion Application**
Real-time Pose Estimation	Not Available	Not Available	Not Available	Tracks exercises like squats and deadlifts using PoseNet.
Personalized Nutrition Plans	Basic calorie tracking	Not Available	Not Available	Fully customized meal plans based on height, weight, and fitness goals.

Exercise Library	Limited workout guidance	Guided programs by trainers	Minimal	Extensive library is categorized by difficulty, equipment, and machines.
Progress Tracking	Calorie-focused	Basic workout tracking	Tracks weightlifting progress	Tracks muscles worked and overall fitness progress.
Beginner and Pro Modes	Not Available	General guidance	Not Available	Tailored workout modes for beginners and pros.
Exercise Suggestions	Not Available	Not Available	Not Available	Recommend exercises based on user goals and preferences.
Exercise Timer	Not Available	Not Available	Basic stopwatch	Includes customizable interval times for exercises

While MyFitnessPal strengths in meal tracking and calorie counting, it lacks advanced workout and feedback features found in the Fitness Companion Application. Lastly, Strong Workout Tracker is an excellent choice for weightlifting enthusiasts but does not

provide tools for posture correction, exercise suggestions, or gamified features, making it less comprehensive than the Fitness Companion Application. The proposed artefact bridges these gaps, offering a seamless blend of functionalities in a single platform.

2.3. Analysis and findings

The reviewed literature highlights the increasing role of technology in fitness monitoring, physical activity management, and health improvement. Key findings from the studies are analyzed below

1. Accuracy in Pose Estimation for Exercises

Statistical findings from (Sebastian, et al., 2023) highlight PoseNet Pose's reduced accuracy under suboptimal conditions. Descriptive analysis suggests that improving pose estimation tools is vital for fitness and health-critical scenarios like rehabilitation. While apps such as Nike Training Club and FitOn offer guided workouts, they lack real-time pose estimation or posture correction, underscoring a gap for more advanced solutions.

2. Impact of COVID-19 on Physical Activity

(Stockwell, et al., 2021) presents statistical evidence of declining physical activity during lockdowns, with sedentary behavior increasing across many populations. Descriptive insights insights the effectiveness of digital fitness solutions like MyFitnessPal, which mitigated disruptions by promoting dietary tracking and exercise routines. However, apps like Strong and JEFIT focus mainly on workout logging, lacking features to counter the holistic effects of reduced physical activity.

3. Gamification in Fitness Recommender Systems

(Zhao, et al., 2020) demonstrated statistically significant improvements in user satisfaction through gamified, personalized systems. Descriptive analysis emphasizes features like goal-setting and storylines for sustained motivation. While apps like FitOn use social competition to engage users, they do not incorporate the adaptive gamification or tailored player modeling seen in advanced systems.

4. Diet Monitoring and Precision Nutrition

Statistical validation from (Abeltino, et al., 2024) supports the effectiveness of AI-powered diet-monitoring tools. Descriptive findings highlight challenges like data accuracy and engagement. Apps like MyFitnessPal offer with dietary needs but lack integration with precision nutrition technologies, presenting insights for innovation in combining AI and user-friendly interfaces.

6. Real-Time Feedback for Exercise Posture

(Hitesh, et al., 2024), provided statistical evidence of improved posture and reduced injury risks with AI-driven feedback systems. Descriptive analysis underscores the accessibility of such tools. Current apps, including Nike Training Club and FitOn, do not offer real-time pose correction, demonstrating the need for advanced technologies like FitSight to democratize professional-level fitness training.

6. Non-Contact Monitoring Using Computer Vision

(Salik, et al., 2022) showcased statistical evidence of accuracy in non-contact monitoring methods like video-based photoplethysmography. Descriptive analysis highlights their potential for fitness and remote health management. However, apps like JEFIT and Strong remain focused on manual tracking and lack integration with such innovative technologies, limiting their scope in comprehensive health monitoring.

2.4. AI Implementation

2.4.1. Research paper

The foundation of AI implementation in my application comes from the following research paper:

“ *PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization*”.Author of the articles are (Kendall, et al., 2015).

While PoseNet focuses on camera delocalization, its core principles apply directly to human pose estimation using Convolutional Neural Networks (CNNs).

2.4.1.1. How the AI Works: The Core Algorithm

The system relies on a Convolutional Neural Network (CNN), a deep learning model designed to process visual data efficiently.

Key Mathematical Concepts includes:

I. Convolution – The heart of CNNs. Small filters (weight matrices) slide across the image, multiplying their values with pixel values and summing them up to detect features like edges, textures, and shapes. Mathematically, for a 2D image *I(x,y)* and filter *K(u,v)*, convolution is defined as:

$$(I * K)(x, y) = \sum_u \sum_v I(x - u, y - v) \cdot K(u, v)$$

II. Non-Linear Activation (ReLU) – After each convolution, the model applies an activation function like ReLU (Rectified Linear Unit) to introduce non-linearity, helping the network learn complex patterns:

$$R \qquad eLU(x) = max(0, x$$

III. Pooling – This reduces feature map size, making the model faster and more robust to variations. Max pooling is common—it takes the highest value from a local region.

IV. Loss Function (MSE) – Measures prediction accuracy. Mean Squared Error (MSE) calculates the difference between predicted and actual keypoints:

$$MSE = (1/n) \sum (y\_true - y\_pred)^2$$

V. Optimization (SGD/Adam) – The model improves itself by adjusting weights using optimization algorithms like Stochastic Gradient Descent (SGD) and Adam:

$$w\_\{t + 1\} = w\_t - \eta \nabla L(w\_t)$$

Here, *w_t* represents weights at step *t*, *η* is the learning rate, and *∇L* is the gradient of the loss function.

2.4.1.2. The AI's Purpose & How It Functions

This AI brings real-time perceptual intelligence to a gym app—it "sees" and interprets a user's body posture instantly, just like a personal trainer would.

How It Works:

I. Pattern Recognition – The CNN identifies key body features (limbs, joints) and their spatial relationships.
II. Spatial Reasoning – It processes joint positions to understand posture.
III. Real-Time Processing – The AI analyzes movement frame-by-frame, providing instant feedback.

Right now, the focus is solely on posture analysis, not exercise routines or fitness planning.

2.4.1.3. Performance Evaluation & Key Metrics

To assess the AI's effectiveness, we consider:

- Accuracy: How precisely it detects joint positions (affected by lighting, clothing, camera angle).
- Precision: How consistently it predicts the same pose (minimal jitter).
- Recall: Whether it detects all key joints without missing any.
- Speed (Latency): Must process frames fast enough for real-time feedback.
- Robustness: Works across different environments, body types, and obstructions.
- Mathematical Evaluation: Object Keypoint Similarity (OKS)

A standard metric for pose estimation is OKS, which measures how well predicted keypoints match actual ones, adjusted for body size:

i. $OKS_p = (\sum_i [\exp(-d_i^2 / (2s^2\kappa_i^2)) \cdot \delta(v_i = 1)]) / (\sum_i \delta(v_i = 1))$
ii. d_i = Euclidean distance between predicted & actual keypoint.
iii. s = Person's scale (based on bounding box area).
iv. κ_i = Keypoint-specific constant (adjusts error tolerance).

v. v_i = Visibility flag (1 if keypoint is visible).

OKS ranges from 0 to 1 (1 = perfect match). If we collect labeled posture data, this metric helps evaluate real-world performance.

2.5. Conclusion

The development of fitness apps has substantially advanced fitness tracking, biometric tracking, and personalized coaching. Applications such as PoseNet allow tracking, but struggles for low-light scenarios, as well as tracking of occluded body parts. However, two key elements that were missed in most successful digital apps were not providing real-time corrective posture feedback and tracking engagement of users with gamification elements. Gamification is important as it fundamentally increases motivation for high level performance and long-term exercise adherence in the absence of self-motivation.

Mathematically, the success of AI-based pose estimation hinges on the performance of Convolutional Neural Networks (CNNs) and key evaluation metrics. The PoseNet model, used in this application, relies on convolution operations:

$$(I * K)(x, y) = u\sum v\sum I(x - u, y - v) \cdot K(u, v)$$

where I(x,y)I(x,y)I(x,y) represents the input image, and K(u,v)K(u,v)K(u,v) is the filter detecting movement patterns. The model then applies ReLU activation:

$$ReLU(x) = max(0, x)$$

to introduce non-linearity, followed by pooling layers to reduce feature complexity. The final joint locations are optimized using Mean Squared Error (MSE) for keypoint estimation:

$$MSE = n1\sum(ytrue - ypred)2$$

and evaluated through Object Keypoint Similarity (OKS):

$$OKSp = \sum i\delta(vi = 1)\sum i[exp(2s2\kappa i2 - di2)/\delta(vi = 1)]$$

where did_idi is the Euclidean distance between predicted and actual keypoints, sss represents the subject's scale, and ki accounts for error tolerance.

Gamified fitness systems and AI applications for nutritional guidance continue to show measurable increases in engagement and accuracy. Closing the gaps that exist in current technologies to improve engagement, effectiveness, and accessibility creates a more inclusive and impactful ecosystem for fitness technology.

3. Project Methodology

3.1. Choosing the Right Project Management Approach

Selecting the right project management method plays a key role in how effectively I develop my gym companion application. Since this system brings together several advanced features including real-time pose estimation, customized meal planning, and intelligent workout suggestions, the development process needs to be flexible, iterative, and adaptive. I rely on Agile principles, which allow me to build and refine my application gradually. Agile encourages continuous progress, early testing, and the ability to adapt when new ideas or technical challenges arise. With AI-related systems like mine, where technology and outcomes evolve over time, this flexibility becomes a necessity rather than a preference.

To structure my workflow, I adopt the Kanban framework, a visual system that lets me track work in progress and focus on smooth task flow. Kanban is especially useful for managing the different components of my application, each of which develops at its own pace. Rather than working in time-limited s, I organize tasks on a visual board and move them through stages like “To Do,” “In Progress,” “Testing,” and “Complete.” This allows me to adapt immediately as requirements shift, or technical issues arise. It helps me stay responsive to user input and enables me to release improvements and new features as soon as they’re ready, rather than waiting for the next cycle.

3.2. Why Kanban

Kanban suits the development of my gym application because it supports continuous delivery, adaptability, and clear visual workflow management. My project consists of multiple interconnected modules. These features don't always develop at the same speed. For instance, while my AI-based posture correction system may need extended periods for validation and accuracy testing, user engagement tools can be added or modified more frequently. Kanban lets me manage these varied timelines without conflict.

Each task I work on is tracked visually using Jira, my project management platform of choice. Jira offers flexible drag and drops boards, custom workflows, and detailed tracking, all of which align well with Kanban's structure. I can easily visualize what stage each component is in, identify what needs immediate attention, and avoid bottlenecks. For example, if testing reveals poor pose detection accuracy, I can push that task back to the "In Progress" column and reprioritize model adjustments without affecting the overall flow.

Additionally, Kanban enhances transparency. It helps me and any collaborators such as AI experts, fitness consultants, or testers stay updated on project progress and provide input when needed. Since I regularly collect and respond to user feedback, this visibility becomes crucial.

3.3. Major Milestone/ Deliverables

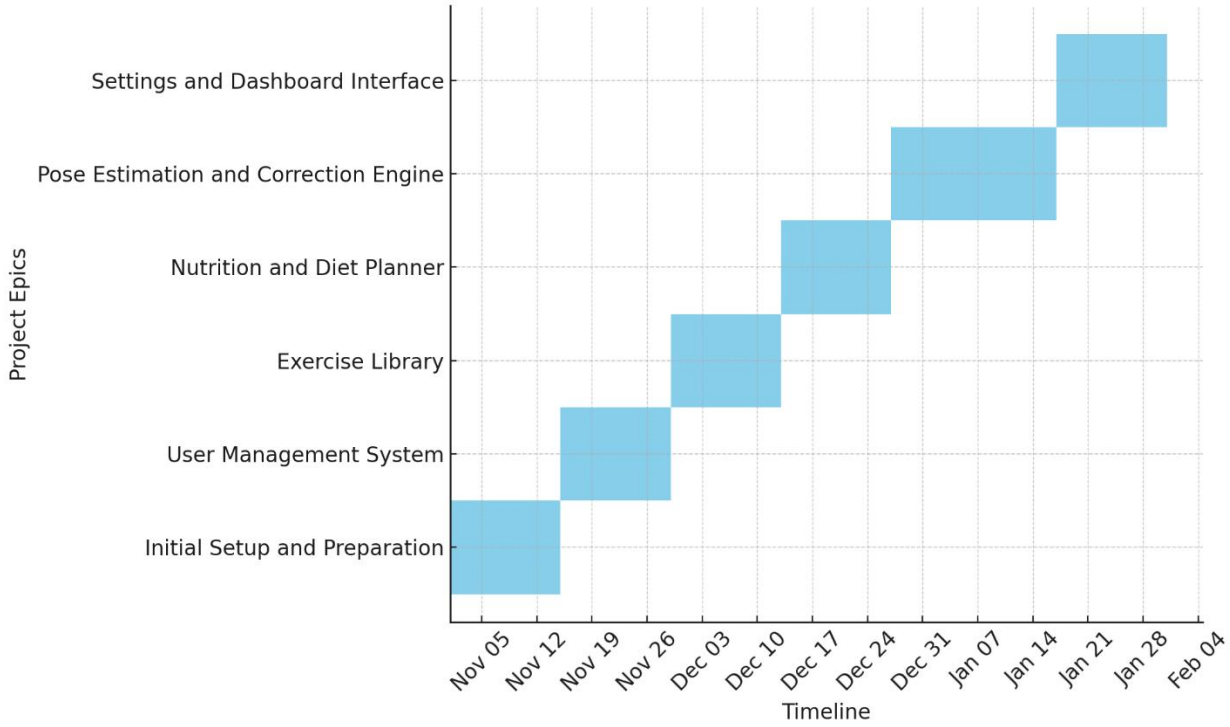


Figure 17: Milestones

To maintain clear direction throughout the development of my fitness application, I break down the project into major epics, each representing a critical area of functionality. These epics form the backbone of my project timeline and are visually structured using a Gantt chart to map out the expected flow and overlap of development activities.

Although I follow a Kanban approach for managing day-to-day progress, the Gantt chart offers a higher-level view of how the main components evolve over time. This helps me plan, manage dependencies, and align key development phases with project goals.

The following milestones mark the major stages of progress:

1. Initial Setup and Preparation: This first phase covers foundational work such as setting up the development environment, collecting necessary resources, and preparing visual and technical documentation. It lays the groundwork for future modules.

2. User Management System: Here, I focus on creating secure and intuitive user registration, login, and profile management features. This module acts as the entry point for users and handles identity-related processes.

3. Exercise Library: This stage involves building a comprehensive library of workouts, categorized by type, intensity, and body focus. It includes the backend structure and interface for browsing and selecting exercises.

4. Nutrition and Diet Planner: This phase introduces personalized meal planning, calorie tracking, and dietary recommendations based on fitness goals. It requires integrating dynamic data and creating adaptable meal templates.

5. Pose Estimation and Correction Engine: At this point, I implement the core AI-based functionality real-time pose detection. This will detect whether the pose for certain exercise is correct or not.

6. Settings and Dashboard Interface: The final major module focus on creating a user dashboard, settings management, progress tracking, and customization options to enhance user experience and usability.

Each epic is mapped over a timeline, with some overlapping, depending on interdependence and task readiness. The Kanban board in Jira helps me manage individual tasks in real time, this Gantt chart keeps me focused on the broader project structure and ensures that I meet important deliverables in a well-organized manner.

4. Technologies and Tools

4.1. Justification for My Chosen Tools and Techniques

Initialization and success of projects depends upon selecting the right tools and technologies. My choices must ensure the running of application efficiency, reliability, and seamless integration of various functionalities.

I am using Django for my backend development because of its scalability, security, and built in features which accelerate development. For the frontend section, I choose React Native community frameworks that allow me to create a cross-platform mobile application with seamless user experience

To manage my source code, I rely on Git with GitHub for version control. This allows me to track changes, collaborate effectively, and maintain code integrity. Additionally, I use PostgreSQL as my database because it handles complex queries efficiently, ensuring smooth performance for user data, exercise logs, and nutrition plans. And for IDE I am Using Visual Studio Code as its seamless performance for all extension files and the robustness of extension provided by it.

5. Programming Languages and Frameworks

I. Backend: Django (Python): I chose Django because it offers a powerful and secure environment for developing my API-driven backend.
II. Frontend: React Native: I am using React Native because of its strong community support and ability to reuse code across platforms.
III. AI/ML: TensorFlow: As per my system requirements PoseNet real-time pose estimation, I need a deep learning framework that supports model training, optimization, and deployment efficiently.
IV. Database: PostgreSQL: I chose PostgreSQL because of its two important features of scalability and ability to handle complex relationships between entities like user activity, exercise data, and nutrition plans.

5.1. Development Tools

I. IDE: Visual Studio Code and Android Studio – VS Code is my preferred editor for backend and frontend development because of its lightweight nature and vast extensions. Android Studio helps me with react native development and debugging.

II. Project Management: I manage my project using Jira for its seamless integration with agile. It helps me track epics, monitor progress, and collaborate iteratively and effectively.

III. Version Control: Git with GitHub: Version control is critical for managing changes, collaborating with others, and ensuring that my project remains stable. I am used to git and GitHub, so I am utilizing my knowledge over them for the project tracking.

5.2. UI/UX Enhancements

Icon Package: I am using Material Icons and Font Awesome for consistency and accessibility in my UI design.

5.3. Testing and Quality Assurance

I am continuously testing for my integrations.

I. Mock Testing: Jest for frontend testing ensures that my React Native components function correctly.

II. Integration Testing: Postman, I use Postman for API testing to validate request-response cycles and ensure seamless integration.

5.4. Package Managers

I. Node Package Manager (NPM): I am using NPM to manage JavaScript dependencies in my React Native application.
@babel/core@7.26.10, @babel/preset-env@7.26.9, @babel/runtime@7.26.10, @notifee/react-native@9.1.8, @react-native-async-storage/async-storage@2.1.2, @react-native-community/cli-platform-android@15.0.1, @react-native-community/cli-platform-ios@15.0.1, @react-native-community/cli@15.0.1, @react-native-community/datetimepicker@8.3.0, @react-native-community/push-notification-ios@1.11.0, @react-native-picker/picker@2.11.0, @react-native/babel-preset@0.76.5, @react-native/eslint-config@0.76.5, @react-native/gradle-plugin@0.77.1, @react-native/metro-config@0.76.5, @react-native/typescript-config@0.76.5, @react-navigation/native-stack@7.3.1,

@react-navigation/native@7.0.17, @thinksys/react-native-PoseNet@0.0.13, @types/react-test-renderer@18.3.1, @types/react@18.3.19, @unsw-gsbme/react-native-keep-awake@1.0.5, axios@1.8.4, babel-jest@29.7.0, base-64@1.0.0, crypto-js@4.2.0, eslint@8.57.1, jest@29.7.0, prettier@2.8.8, react-native-element-dropdown@2.12.4, react-native-fast-image@8.6.3, react-native-human-pose@1.1.1, react-native-image-picker@8.2.0, react-native-linear-gradient@2.8.3, react-native-push-notification@8.1.1, react-native-safe-area-context@5.3.0, react-native-screens@4.9.2, react-native-vector-icons@10.2.0, react-native-vision-camera@4.6.4, react-native-webview@13.13.4, react-native-xml2js@1.0.3, react-native@0.76.7, react-test-renderer@18.3.1, react@18.3.1, typescript@5.0.4.

II. Pip: Since my backend is built in Python, I am using Pip to manage dependencies such as Django, and PostgreSQL connectors.

absl-py==2.1.0, asgiref==3.8.1, astunparse==1.6.3, attrs==25.3.0, bidict==0.23.1, blinker==1.9.0, certifi==2024.12.14, cffi==1.17.1, charset-normalizer==3.4.1, click==8.1.8, colorama==0.4.6, contourpy==1.3.1, cycler==0.12.1, distlib==0.3.9, dj-rest-auth==7.0.1, Django==5.1.4, django-allauth==65.3.1, django-cors-headers==4.6.0, djangorestframework==3.15.2, djangorestframework-simplejwt==5.3.1, dnspython==2.7.0, eventlet==0.39.1, filelock==3.16.1, Flask==3.1.0, flask-cors==5.0.1, Flask-SocketIO==5.5.1, flatbuffers==25.2.10, fonttools==4.56.0, gast==0.6.0, google-pasta==0.2.0, greenlet==3.1.1, grpcio==1.71.0, h11==0.14.0, h5py==3.13.0, idna==3.10, itsdangerous==2.2.0, jax==0.5.2, jaxlib==0.5.1, Jinja2==3.1.6, keras==3.9.0, kiwisolver==1.4.8, libclang==18.1.1, Markdown==3.7, markdown-it-py==3.0.0, MarkupSafe==3.0.2, matplotlib==3.10.1, mdurl==0.1.2, PoseNet==0.10.21, ml_dtypes==0.5.1, namex==0.0.8, numpy==1.26.4, opencv-contrib-python==4.11.0.86, opencv-python==4.11.0.86, opt_einsum==3.4.0, optree==0.14.1, packaging==24.2, pillow==11.1.0, platformdirs==4.3.6, protobuf==4.25.6, psycopg2-binary==2.9.10,

pycparser==2.22, Pygments==2.19.1, PyJWT==2.10.1, pyparsing==3.2.1, python-dateutil==2.9.0.post0, python-dotenv==1.0.1, python-engineio==4.11.2, python-socketio==5.12.1, requests==2.32.3, rich==13.9.4, scipy==1.15.2, sentencepiece==0.2.0, setuptools==77.0.1, simple-websocket==1.1.0, six==1.17.0, sounddevice==0.5.1, sqlparse==0.5.3, tensorboard==2.19.0, tensorboard-data-server==0.7.2, tensorflow==2.19.0, termcolor==2.5.0, typing_extensions==4.12.2, tzdata==2024.2, urllib3==2.3.0, virtualenv==20.28.1, Werkzeug==3.1.3, wheel==0.45.1, wrapt==1.17.2, wsproto==1.2.0

5.5. Conclusion

The combination of Django, React Native and PostgreSQL enables me to build this powerful and scalable fitness application. By using robust development tools and methodologies, I ensure smooth collaboration, efficient debugging, and intuitive user experience.

6. Artefacts Design

The artefact designs for this Final Year Project (FYP) evolve hand-in-hand with the system's development, ensuring that every requirement, diagram, and test matures as we move forward. Guided by the Scrum framework, the work splits into s, making the process both structured and flexible. Below, I break down each subsystem in detail, exploring how the artefacts come to life and where we stand today. The project is fully assigned to Ananda Neupane which is me and spans from November 2024 to April 2025 (and beyond). It includes five completed s and one ongoing, covering various subsystems like User Management, Exercise Library, Nutrition/Diet, and Pose Estimation. Here's a breakdown of the Epics and their associated Stories:

1. Epic: Prerequisites of FYP (Nov 5 - Nov 20, 2024)

I. Research topics and prepare for Idea Presentation

II. Start working on project proposal

2. Epic: Prebuilt Documentations (Nov 19 - Dec 3, 2024)

I. Build Gantt chart and WBS Diagram
II. Work on architecture diagram
III. Deliver Figma design and Literature review

3. Epic: User Management (Dec 3, 2024 - Jan 1, 2025)

I. Complete prototype and literature review report
II. Build database design diagram
III. Work on user registration frontend
IV. Integrate backend and develop login functionality
V. Use email verification and JWT token for authorization

4. Epic: Exercise Library (Jan 4 - Feb 10, 2025)

I. Research database and references
II. Handle API calls and create frontend views
III. Integrate backend models for user data
IV. Enable users to add/delete libraries and exercises
V. Study calories burn calculation mechanism

5. Epic: Nutrition/Diet (Feb 11 - Mar 4, 2025)

I. Search for nutrition database with dietary preferences
II. Integrate API and enhance user experience
III. Insert user dietary plans into database

6. Epic: Pose Estimation (Mar 5 – Apr 9, 2025)

I. Search for API/package for pose estimation
II. Integrate frontend sections
III. Build logic for angle calculation
IV. Add backend for pose history

7. Epic: Setting and Dashboards (Apr 9 - Apr 29,2025)

I. Start with integrating frontend for BMI calculations, daily streaks.

II. Build frontend of showing calorie burned through exercise and nutrition separately.

III. Integrate backend to fetch the tabular data.

6.1. Sub System 1: Prerequisites of FYP

Explanation: This subsystem lays the foundation for the project by identifying research topics, preparing an idea presentation, and drafting the initial proposal. It ensures the project scope and objectives are defined early.

6.1.1. SRS:

Document initial requirements, such as the need for a project proposal and presentation materials.

6.1.2. Design/Modelling Diagrams:

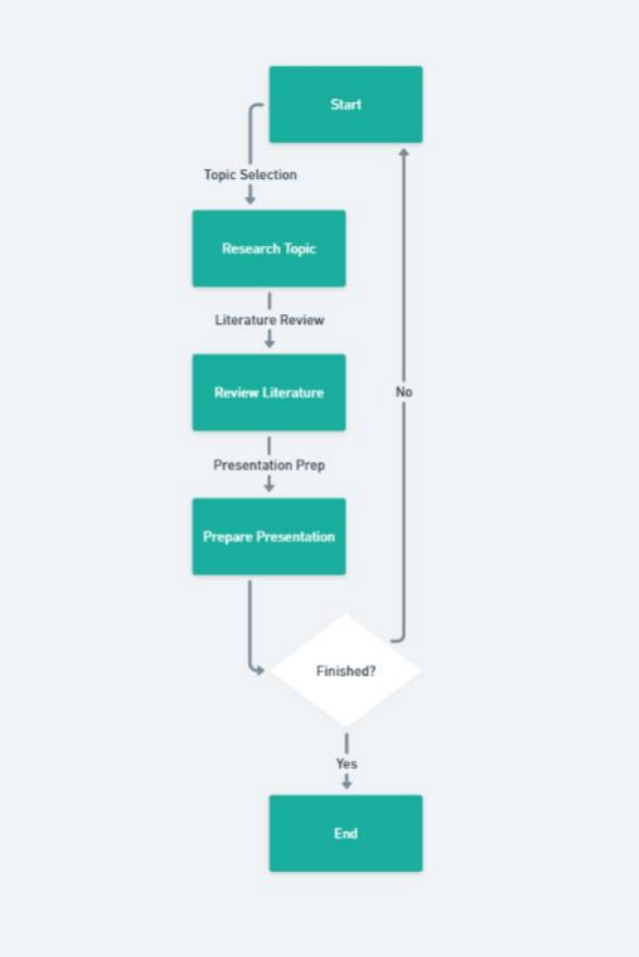


Figure 18:Activity Diagram of Deliverables 1

Activity Diagram: Show the workflow of researching topics and preparing the presentation/proposal.

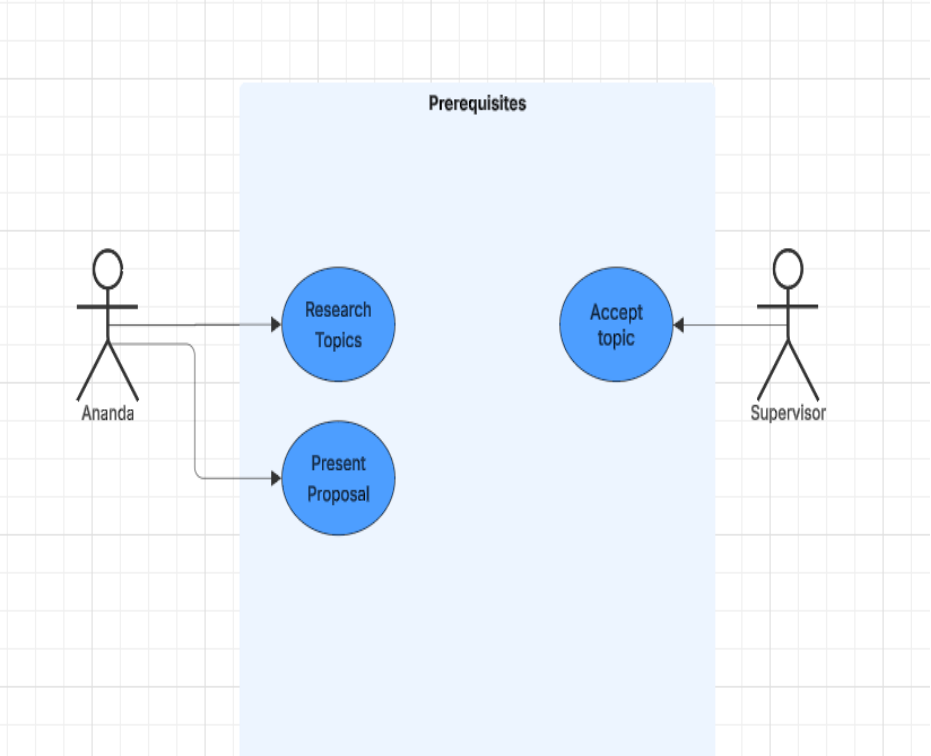


Figure 19:Use case diagram of Deliverables 1

Use Case Diagram: Define actors (e.g., student, supervisor) and use cases (e.g., "Research Topics," "Draft Proposal").

6.1.3. Testing

Validate the proposal and presentation with feedback from peers or supervisors.

6.2. Sub System 2: Prebuilt Documentations

Explanation: This subsystem focuses on creating essential planning and design documents (Gantt chart, WBS, architecture diagram, Figma design, literature review) to guide development.

6.2.1. SRS

This milestone specifies requirements for project planning tools and initial design deliverables.

6.2.2. Design/Modelling Diagrams:

I. Gantt Chart: Timeline of tasks

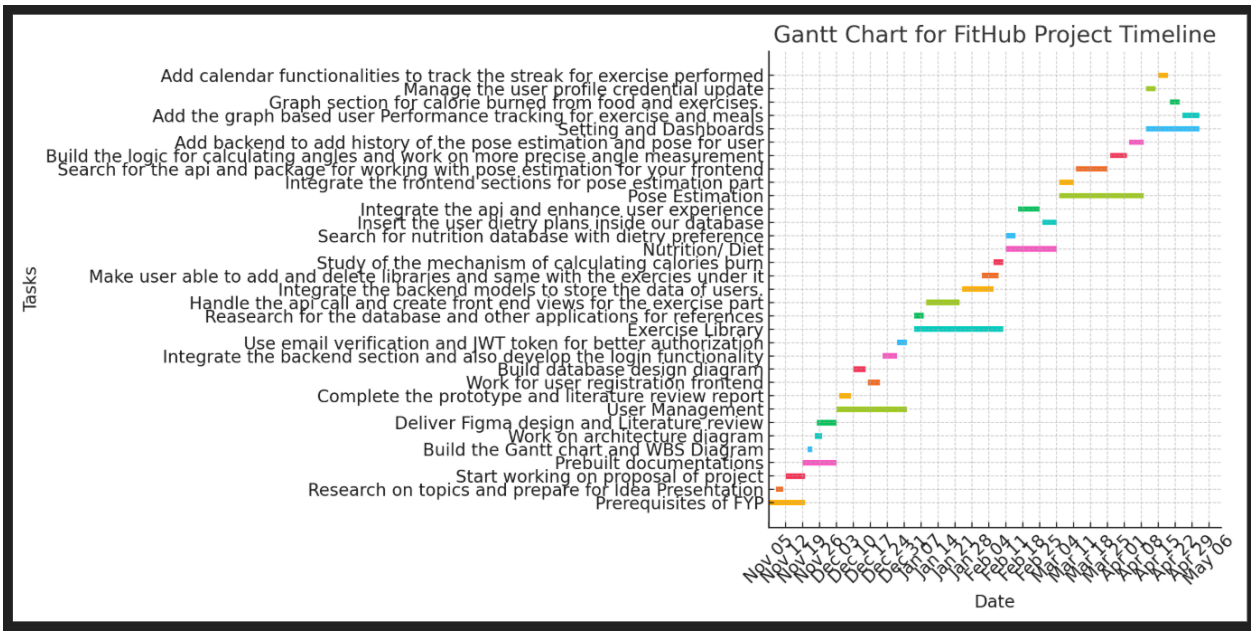


Figure 20: Gantt Chart

II. FDD Diagram: Breakdown of project tasks.

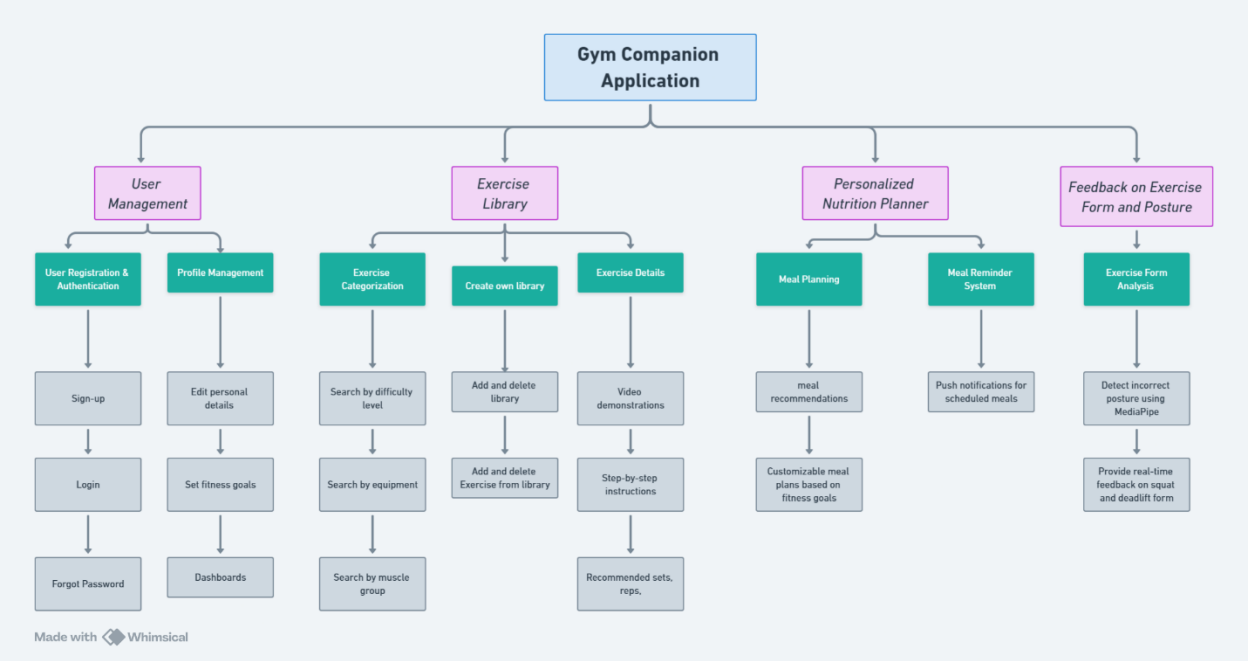


Figure 21: WBS Diagram of deliverable 2

III. Architecture Diagram: High-level system structure .

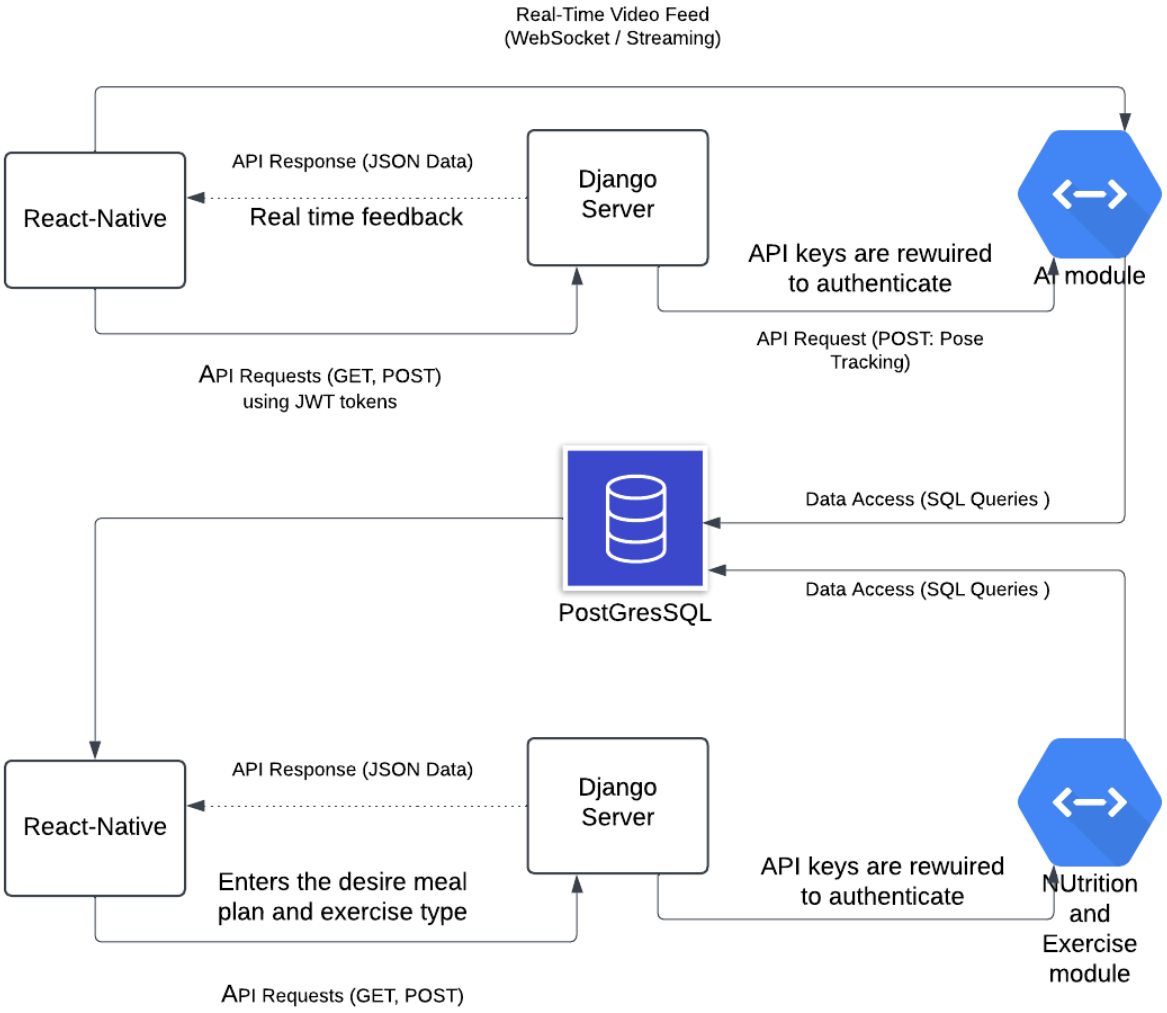


Figure 22: Architecture diagram of deliverable 2

IV. Wireframe: Figma designs for UI .

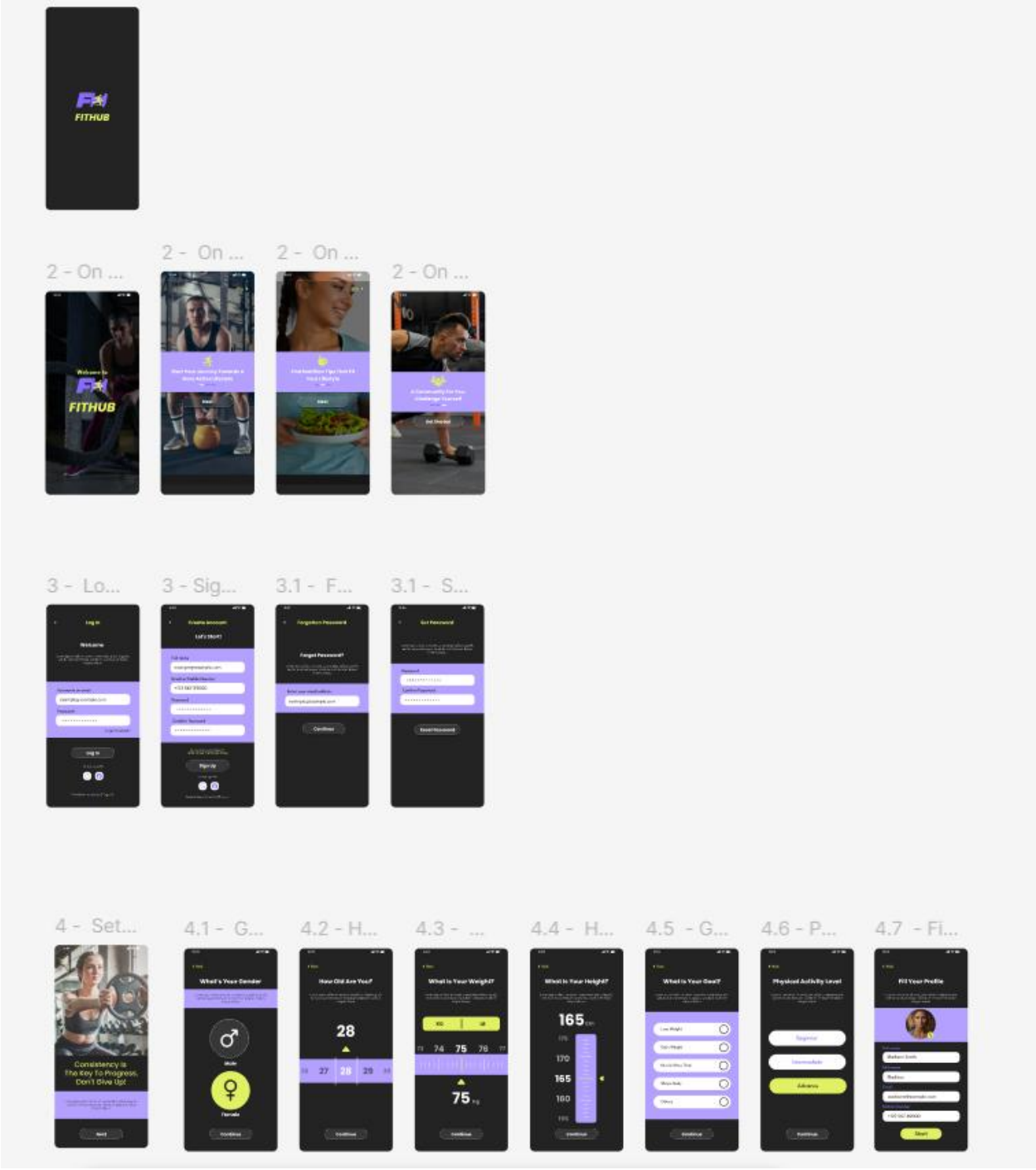


Figure 23: Wireframe part 1

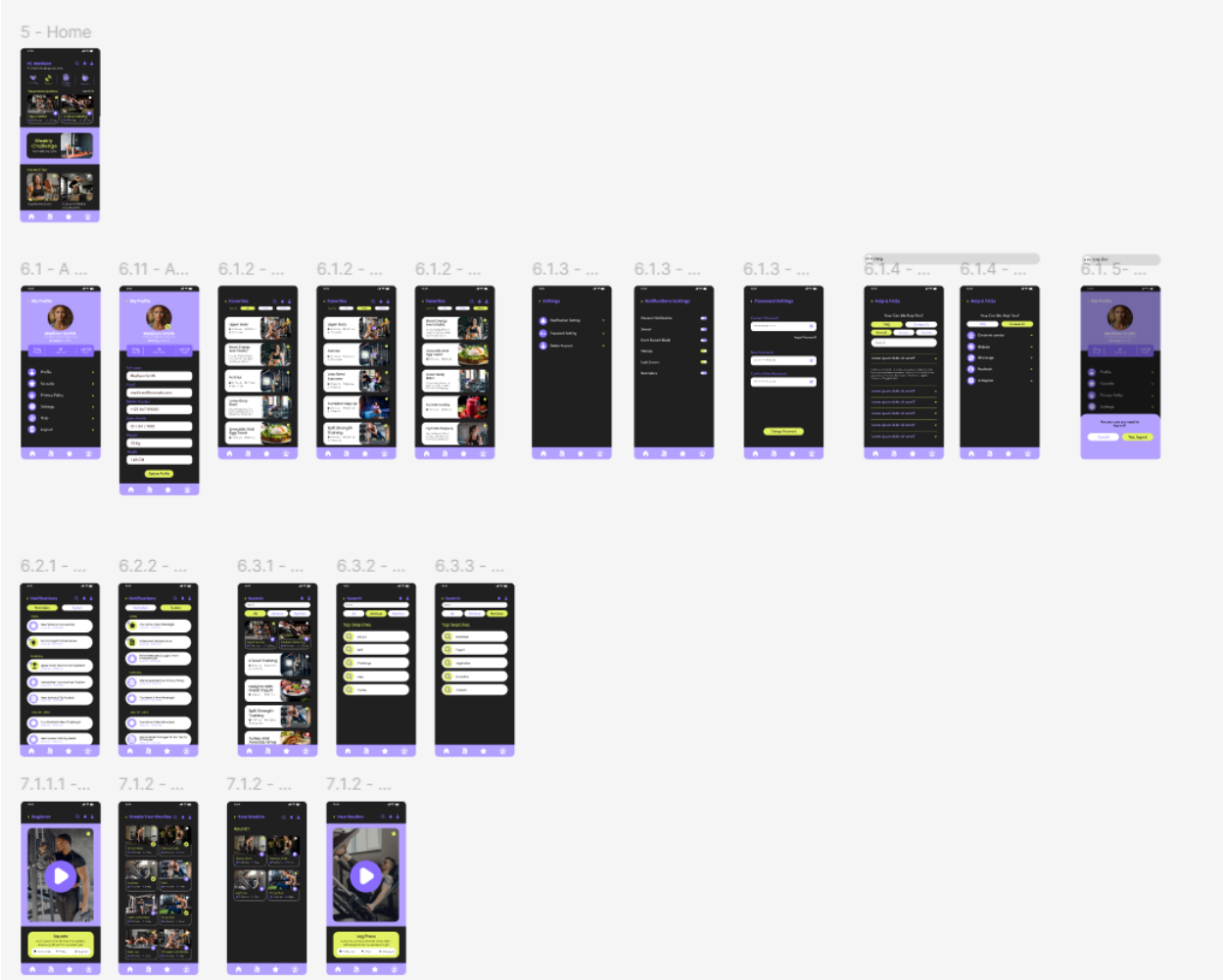


Figure 24: Wireframe part 2

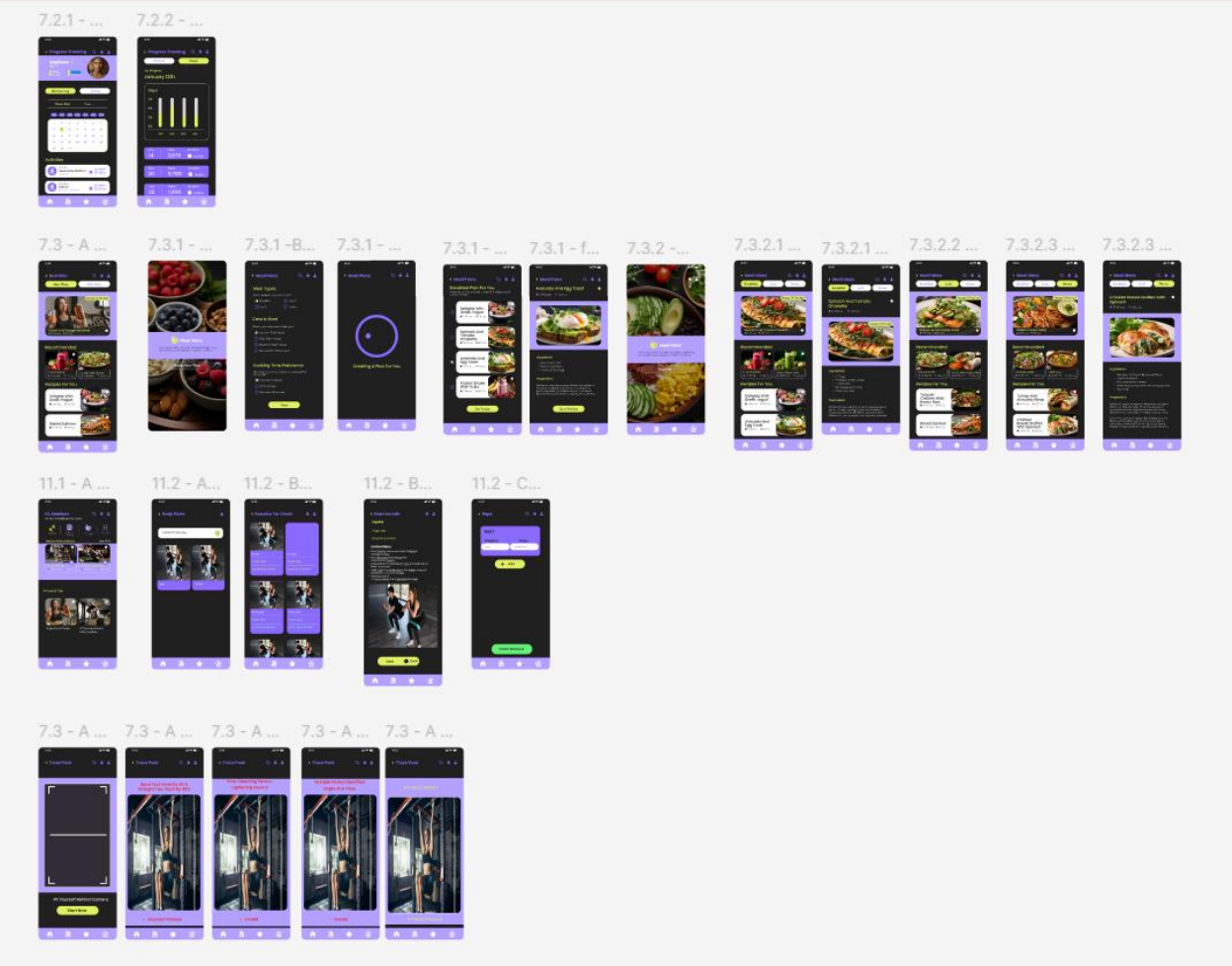


Figure 25: Wire frame part 3

6.2.3. Testing

Supervisors verify the Gantt chart/WBS align with the timeline and review Figma designs/literature for completeness.

6.3. Sub System 3: User Management

Explanation: This subsystem implements user-related features (registration, login, database design, authorization) critical for system functionality.

6.3.1. SRS

Define requirements for user registration, login, and secure authorization (e.g., JWT, email verification).

6.3.2. Design/Modelling Diagrams:

I. Activity Diagram: Workflow for user registration and login.

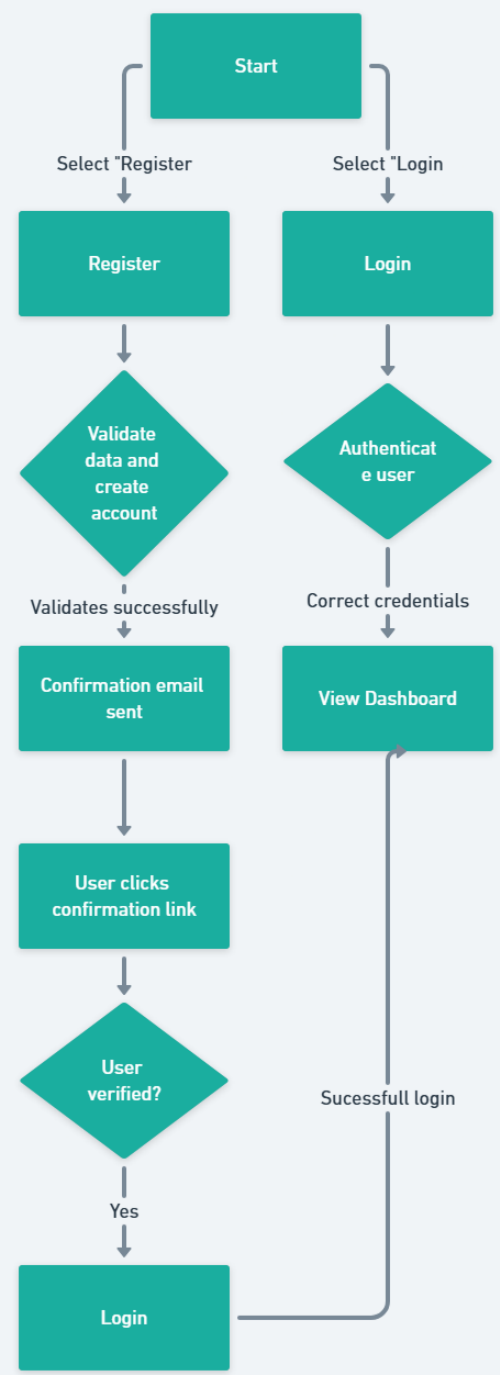


Figure 26: Activity diagram of deliverables 3

II. Wireframe: Frontend design for registration/login pages.

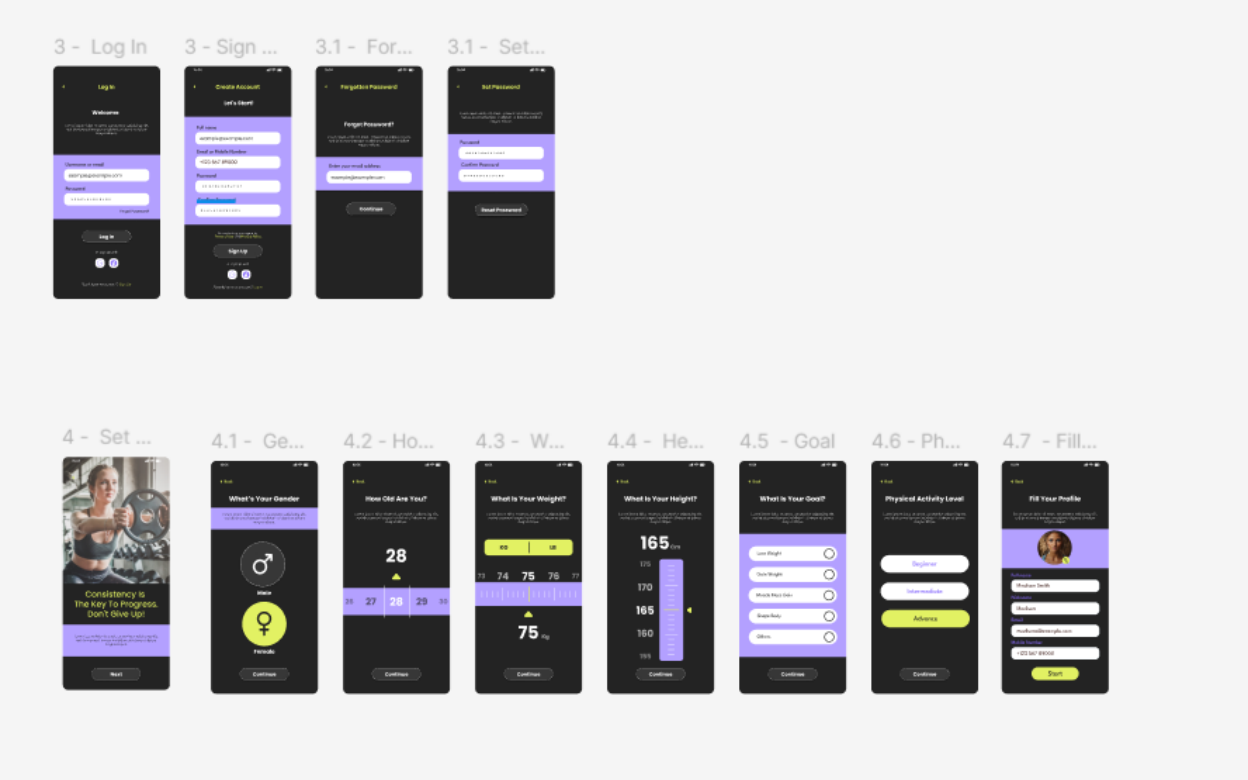


Figure 27: Woreframes of deliverables 3

III. Use Case Diagram: Actors (e.g., User) and use cases (e.g., "Register," "Login").

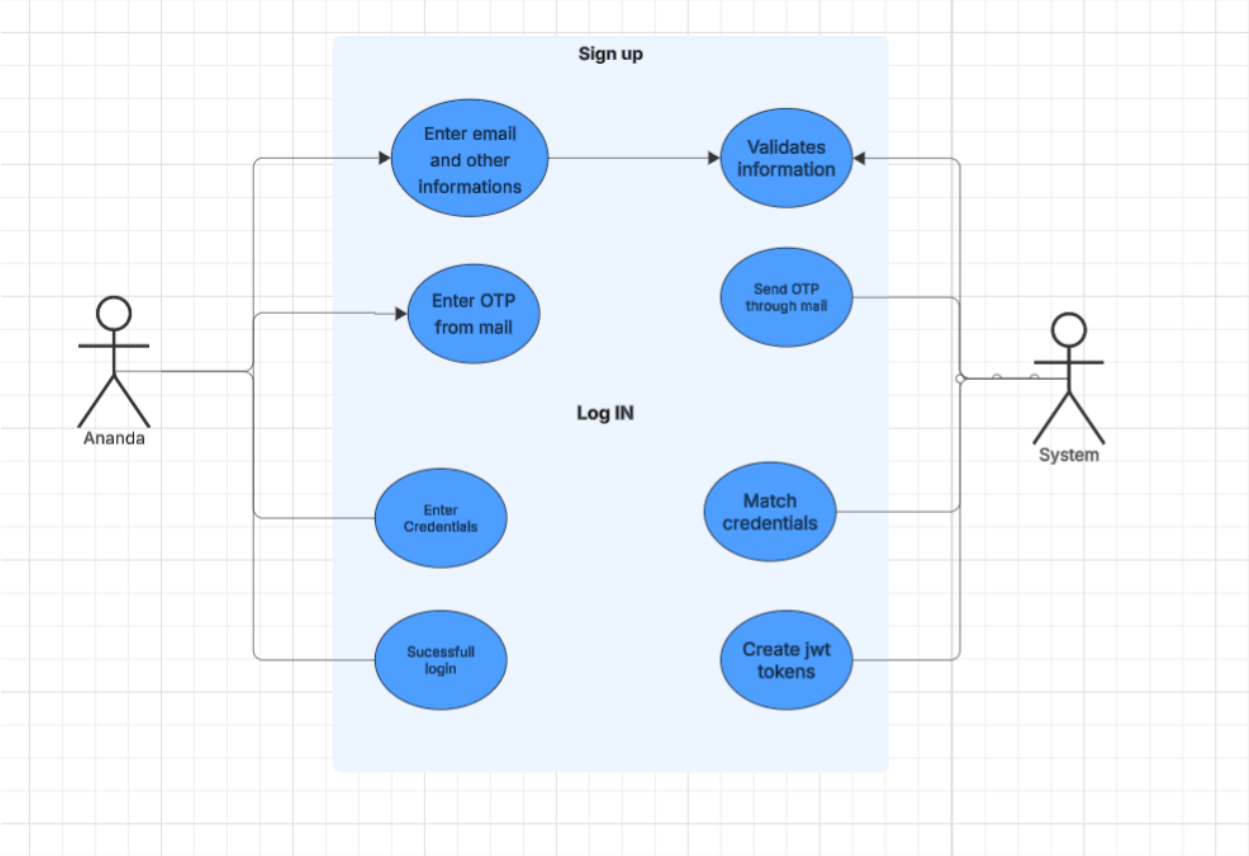


Figure 28: Use case for deliverable 3

IV. ERD: Database design diagram (already built).

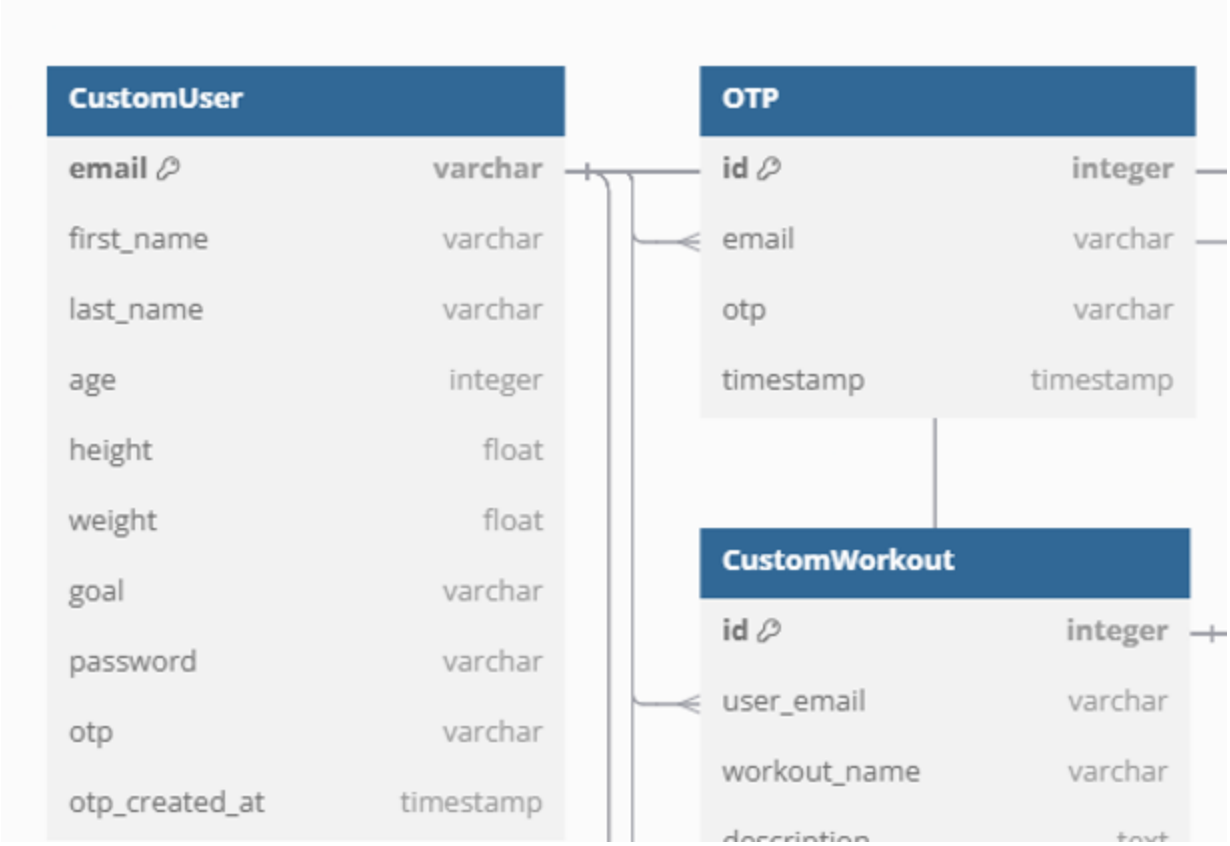


Figure 29: ERD diagram of deliverables 3

V. Class Diagram: Classes for user management (e.g., User, Auth).

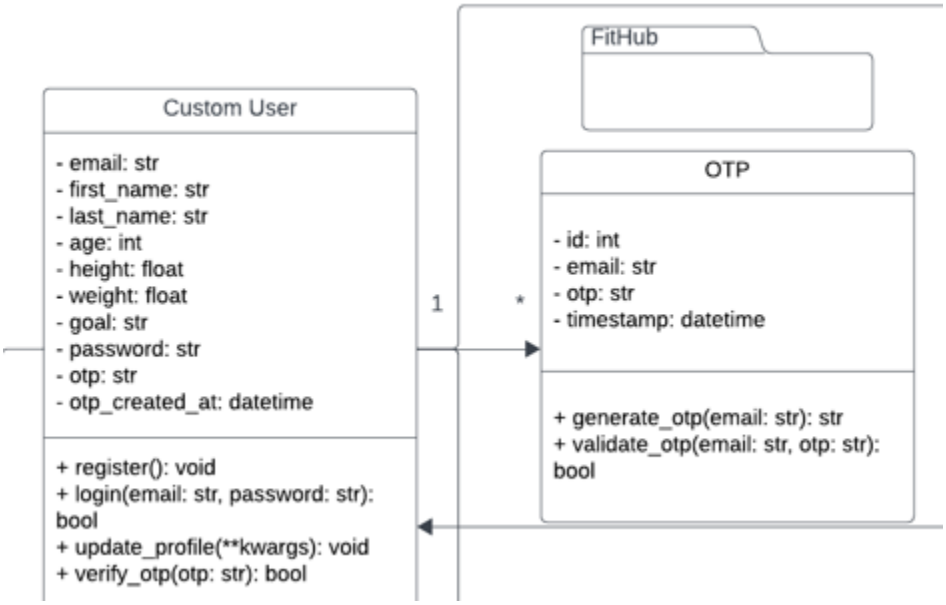


Figure 30: Erd diagram for deliverables 3

VI. Sequence Diagram: Interaction between frontend, backend, and database for login/registration.

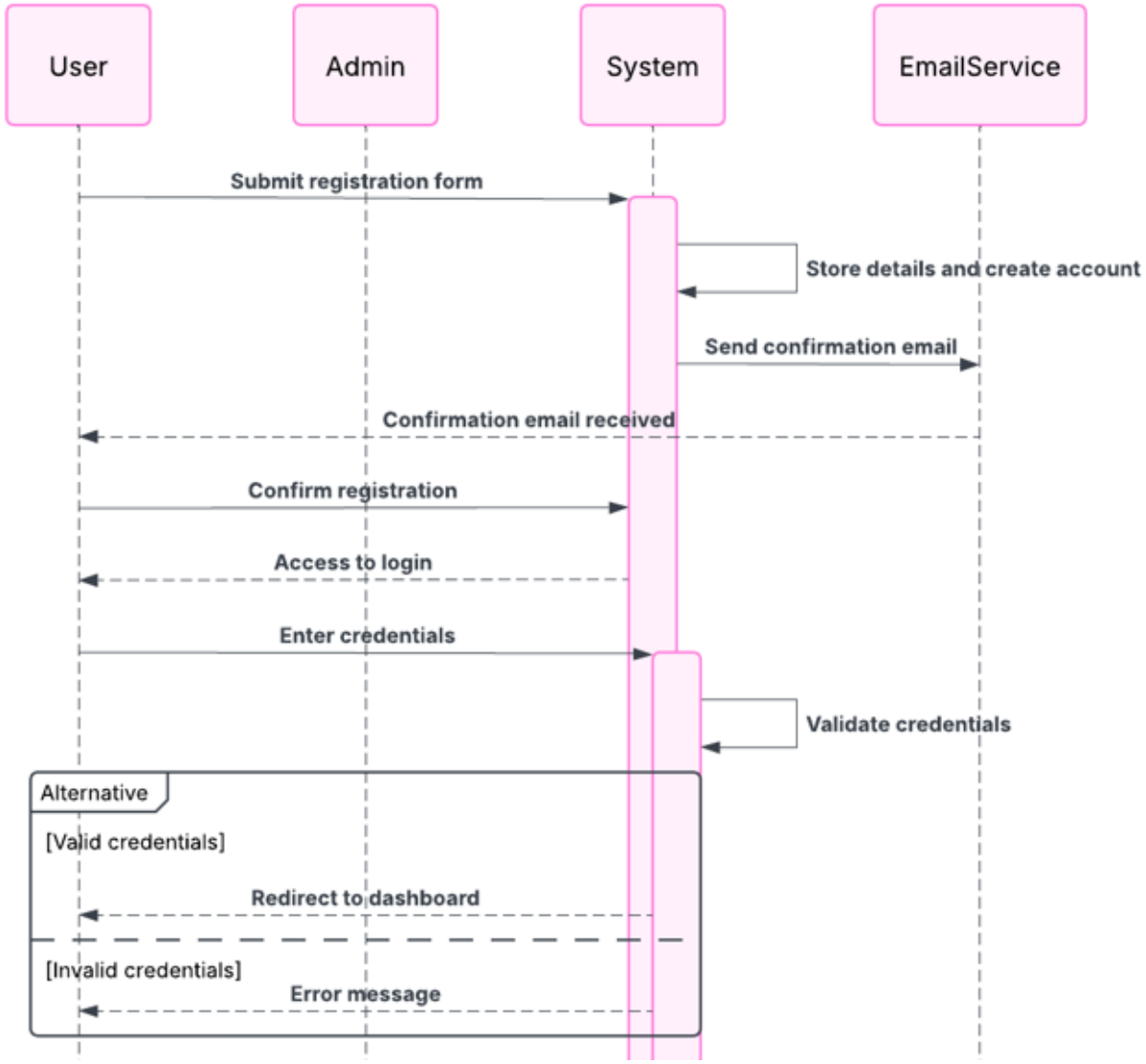


Figure 31: Sequence diagram for deliverables 3

6.3.3. Testing

Test ID	Test Scenario	Expected Result	Actual Result	Status (Pass/Fail)	Comments
Authentication Module					
AUTH-01	Enter email and other corresponding credentials	Your Account has been finally set up	Your Account has been finally set up	Pass	Account setup successful.
AUTH-02	Forgot Password with OTP	OTP sent successfully. Password reset allowed.	OTP sent successfully. Password reset allowed.	Pass	OTP verified and password reset functionality works correctly.
AUTH-03	JWT Token Validation	Access granted. Token expires after set time.	Access granted. Token expires after set time.	Pass	Token is validated and expires correctly.
AUTH-04	Invalid OTP Handling	Error message: "Invalid OTP."	Error message: "Invalid OTP."	Pass	System correctly identifies and rejects invalid OTPs.

Figure 32: Test cases for deliverables 3

Test prototype functionality, database integrity, and security features (e.g., JWT token validation).

6.4. Sub System 4: Exercise Library

Explanation: This subsystem builds a library for exercises, integrating frontend, backend, and calorie calculation features.

6.4.1. SRS

Specify requirements for exercise data storage, retrieval, and user interaction (add/delete).

6.4.2. Design/Modelling Diagrams:

I. Activity Diagram: Workflow for adding/deleting exercises.

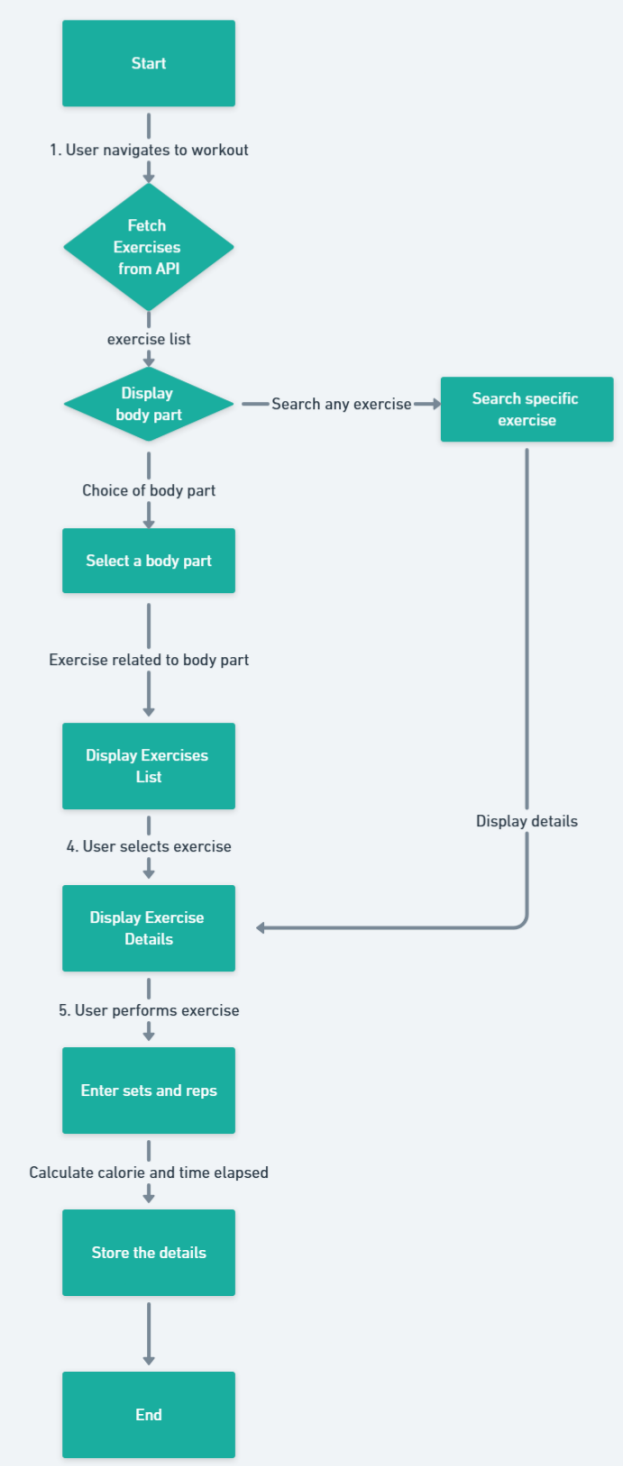


Figure 33: Activity diagram of deliverables 4

II. Wireframe: Frontend views for exercise library.

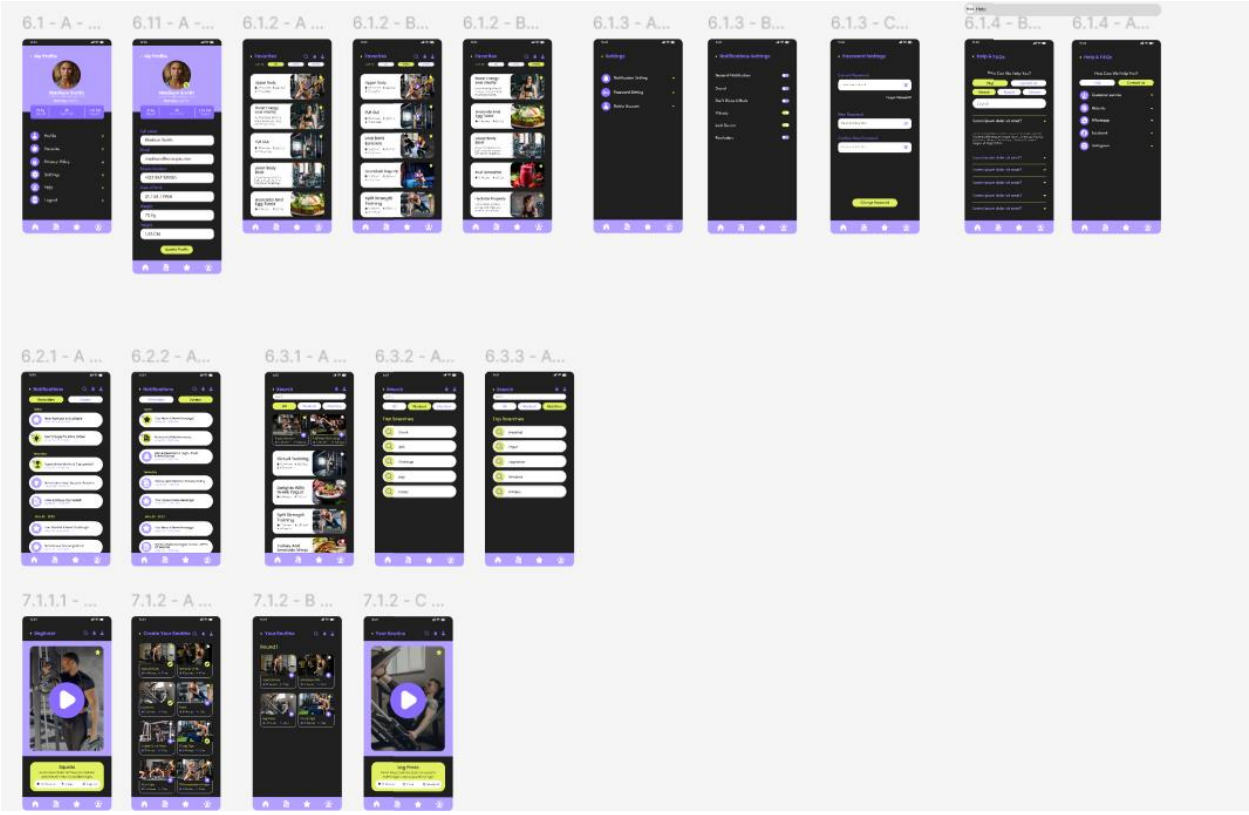


Figure 34: Woreframes of deliverable 4

III. Use Case Diagram: Actors (e.g., User) and use cases (e.g., "Add Exercise," "View Library").

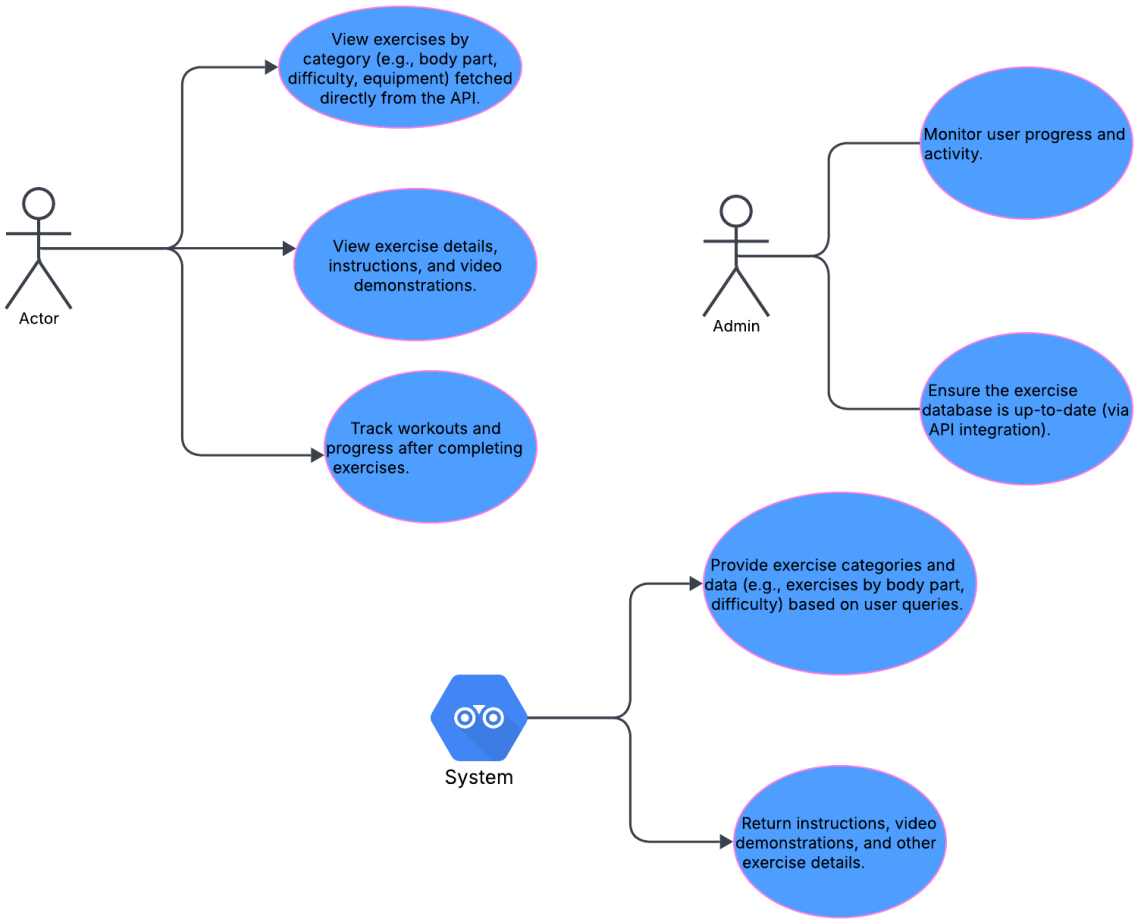


Figure 35: Use case diagram of deliverable 4

IV. ERD: Database schema for exercises and libraries.

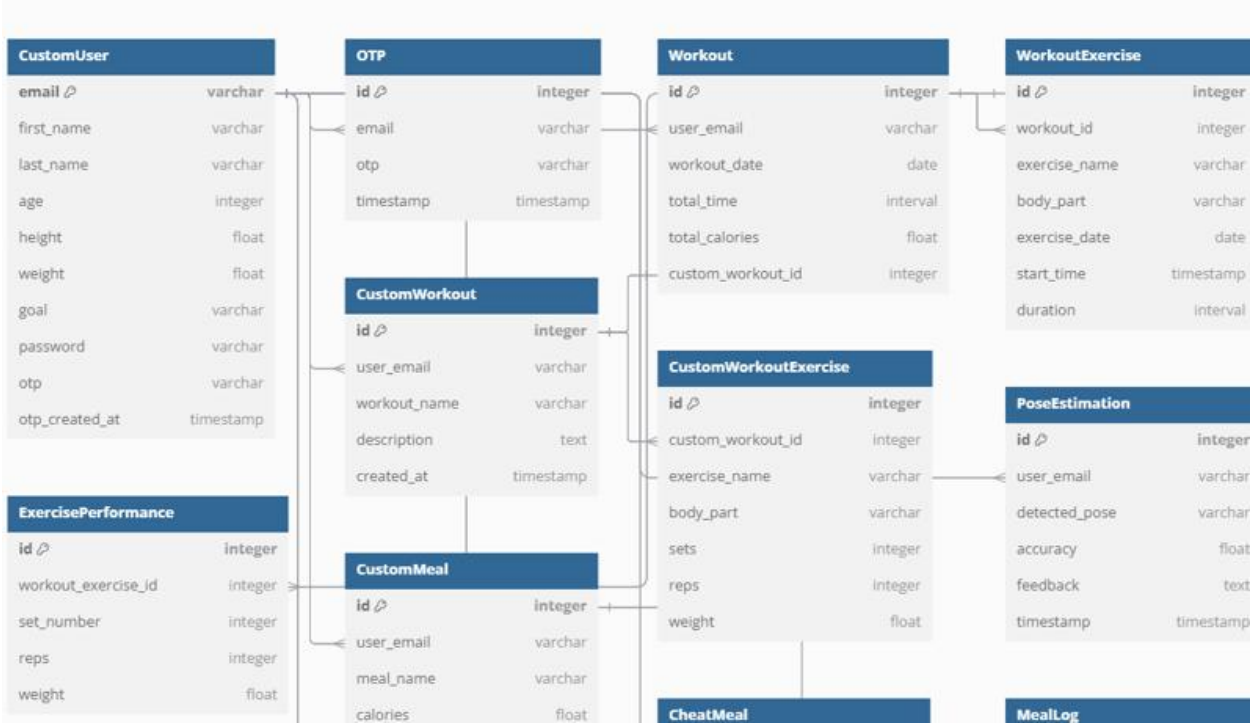


Figure 36: Erd diagram for deliverables 4

V. Class Diagram: Classes for exercise management.

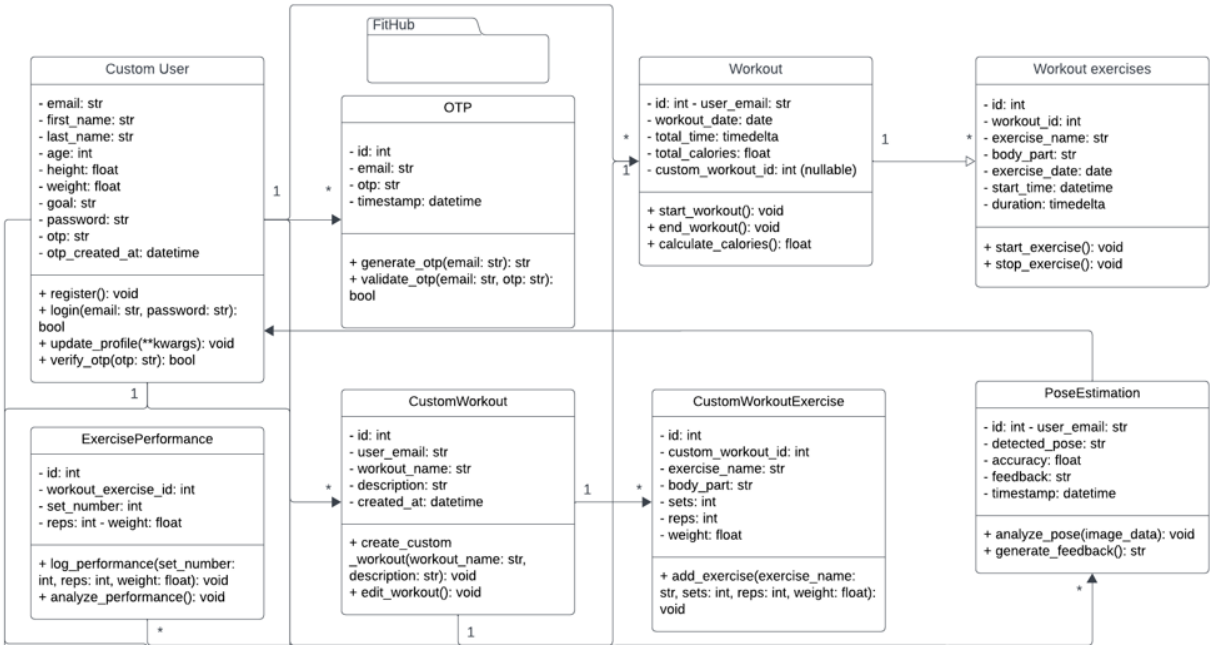


Figure 37: Class diagram for deliverable 4

VI. Sequence Diagram: API calls between frontend and backend.

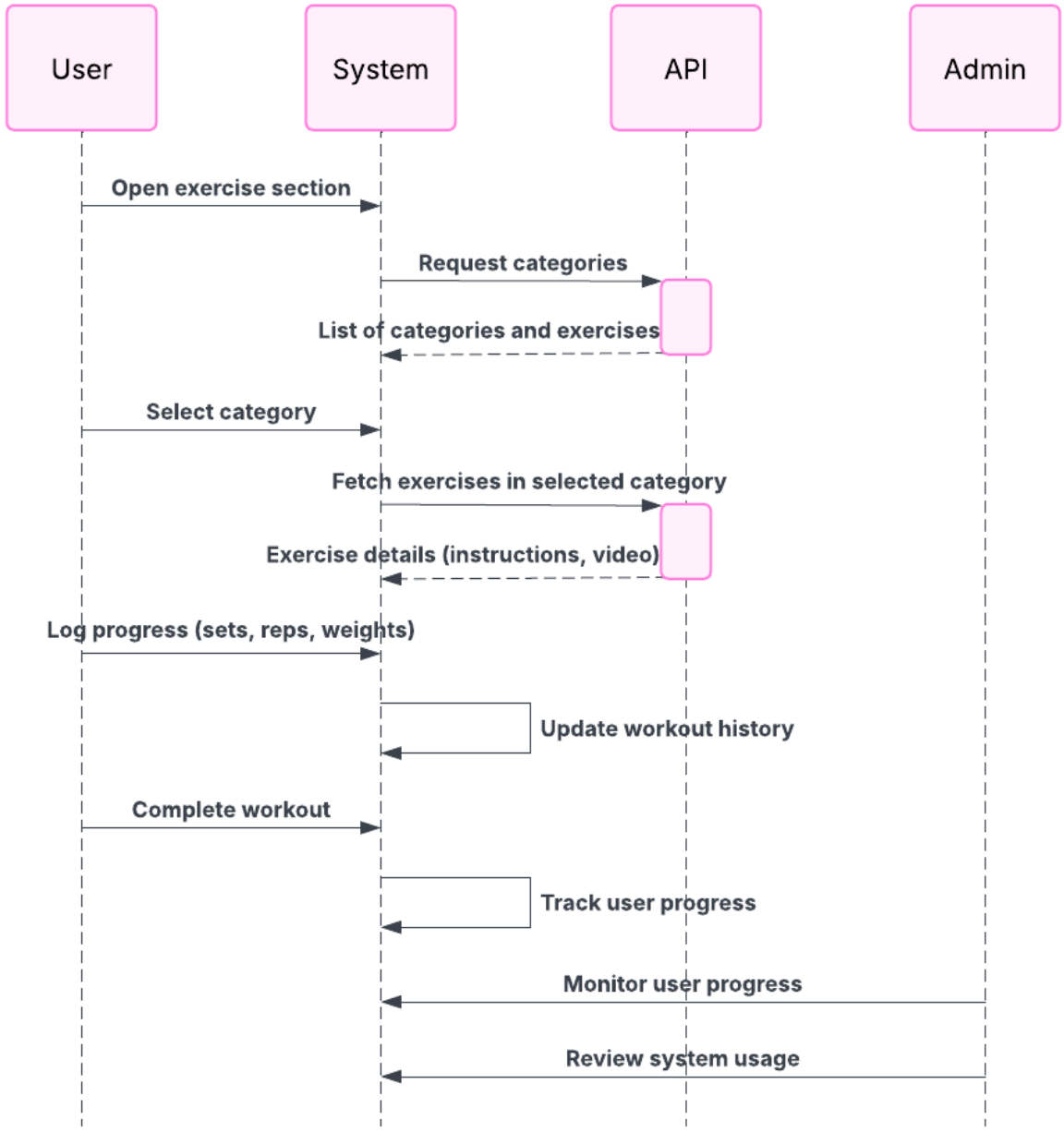


Figure 38: Sequence diagram of deliverables 4

6.4.3. Testing:

Validate API integration, data storage, and calorie calculation accuracy.

Exercise Library Module							
LIB-01	Filter Exercises by Equipment	Only exercises using dumbbells displayed.	Only exercises using dumbbells displayed.	Pass	Filtering by equipment type functions properly.		
LIB-02	Calculate Sets/Weight/Reps	Total volume calculated correctly (1500 kg).	Total volume calculated correctly (1500 kg).	Pass	Set/weight/reps calculation is accurate.		
LIB-03	View Exercise Details	Muscle groups, instructions, and video displayed.	Muscle groups, instructions, and video displayed.	Pass	Exercise details display correctly.		
Workout Logging Module							
LOG-01	Save Daily Workout Routine	Routine saved and accessible in history.	Routine saved and accessible in history.	Pass	Workout history is recorded correctly.		
LOG-02	Track time after workout starts	Timer starts/stops correctly. Notification sent at end.	Timer starts/stops correctly. Notification sent at end.	Pass	Workout timer functions properly.		
LOG-03	Track Muscle Groups Worked	Muscle groups logged and displayed in progress tracker.	Muscle groups logged and displayed in progress tracker.	Pass	Progress tracker accurately updates muscle groups.		
Exercise Suggestions Module							
SUG-01	Recommend Exercises Based on Goals	Strength-focused exercises recommended.	Strength-focused exercises recommended.	Pass	Goal-based recommendations work correctly.		
			Only		Exercise		

Figure 39: Test cases for deliverables 4

Exercise Suggestions Module						
SUG-01	Recommend Exercises Based on Goals	Strength-focused exercises recommended.	Strength-focused exercises recommended.	Pass	Goal-based recommendations work correctly.	
SUG-02	Recommend Exercises Based on Preferences	Only bodyweight exercises recommended.	Only bodyweight exercises recommended.	Pass	Exercise suggestions correctly match user preferences.	
Custom Workout Module						
CUST-01	Create Custom Workout	Custom routine saved and accessible.	Custom routine saved and accessible.	Pass	Users can successfully create and save workouts.	
CUST-02	Modify Custom Workout	Routine updated successfully.	Routine updated successfully.	Pass	Workout modifications reflect correctly.	

Figure 40: Test Cases for deriverables 4

6.5. Sub System 5: Nutrition/Diet

Explanation: This subsystem adds nutrition tracking and dietary planning features.

6.5.1. SRS:

Define requirements for nutrition database and dietary plan integration.

6.5.2. Design/Modelling Diagrams:

I. *Activity Diagram*: Workflow for adding dietary plans.

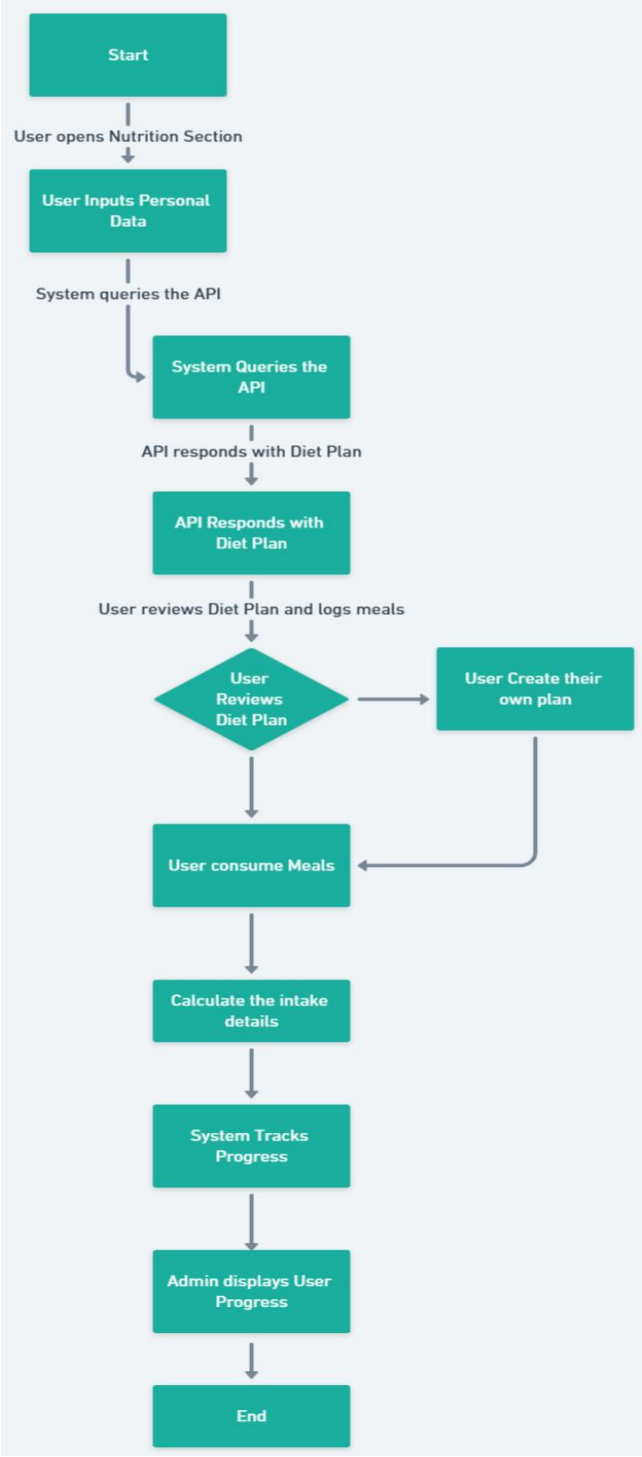


Figure 41: Activity diagram of deliverable 5

II. *Wireframe*: UI for nutrition tracking.

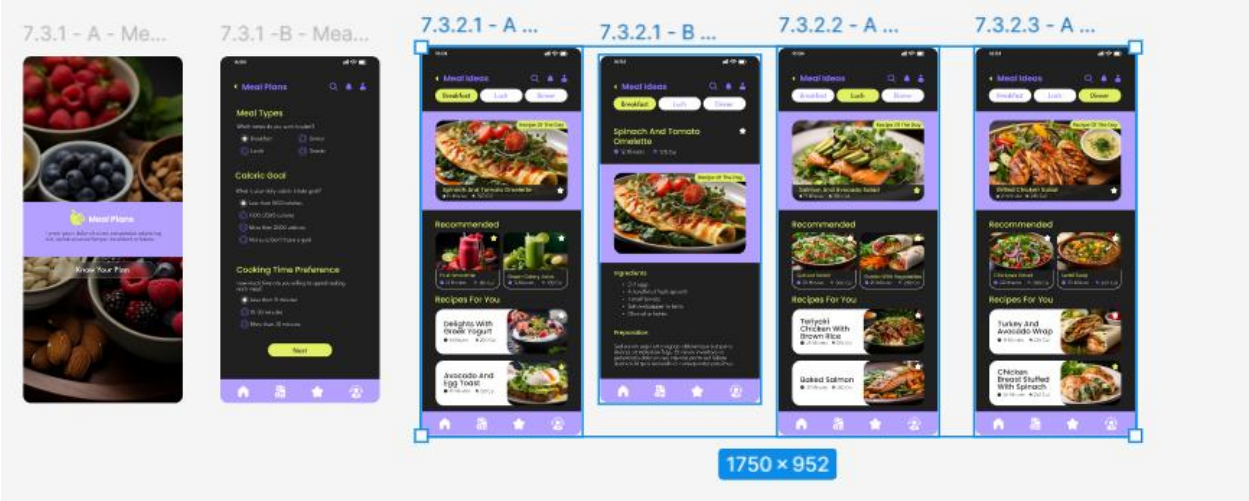


Figure 42: Wireframe for deliverables 5

III. *Use Case Diagram*: Actors (e.g., User) and use cases (e.g., "View Diet Plan").

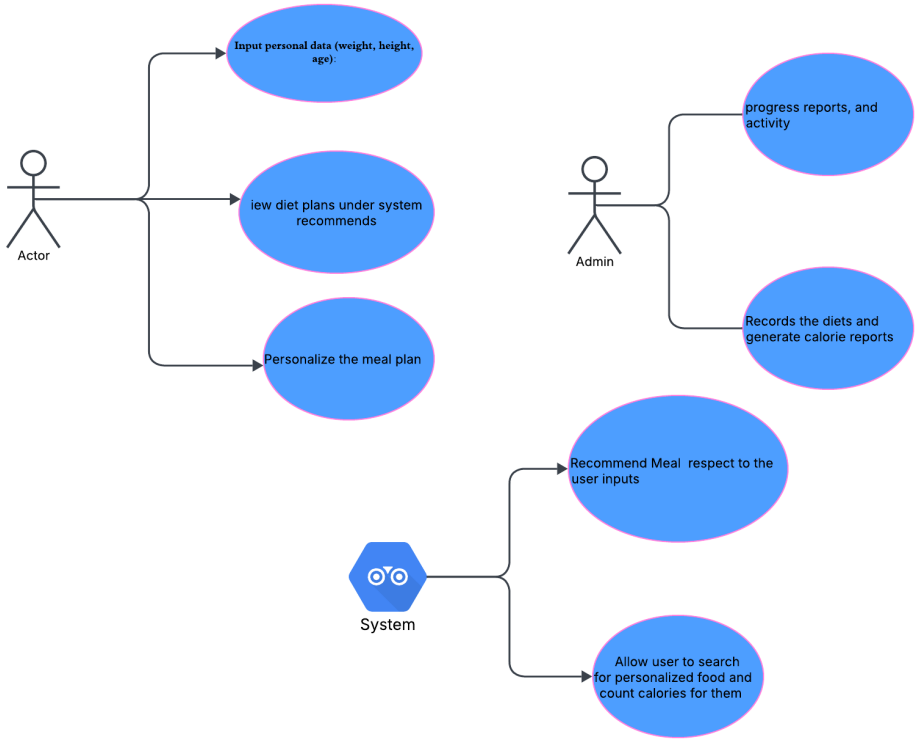


Figure 43: Use case for deliverables 5

IV. *ERD*: Schema for nutrition data.

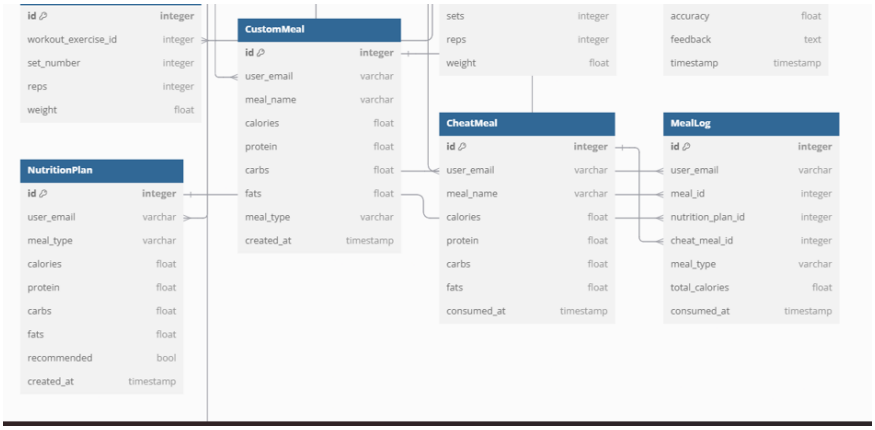


Figure 44: ERD diagram of deliverables 5

V. *Class Diagram*: Classes for diet management.

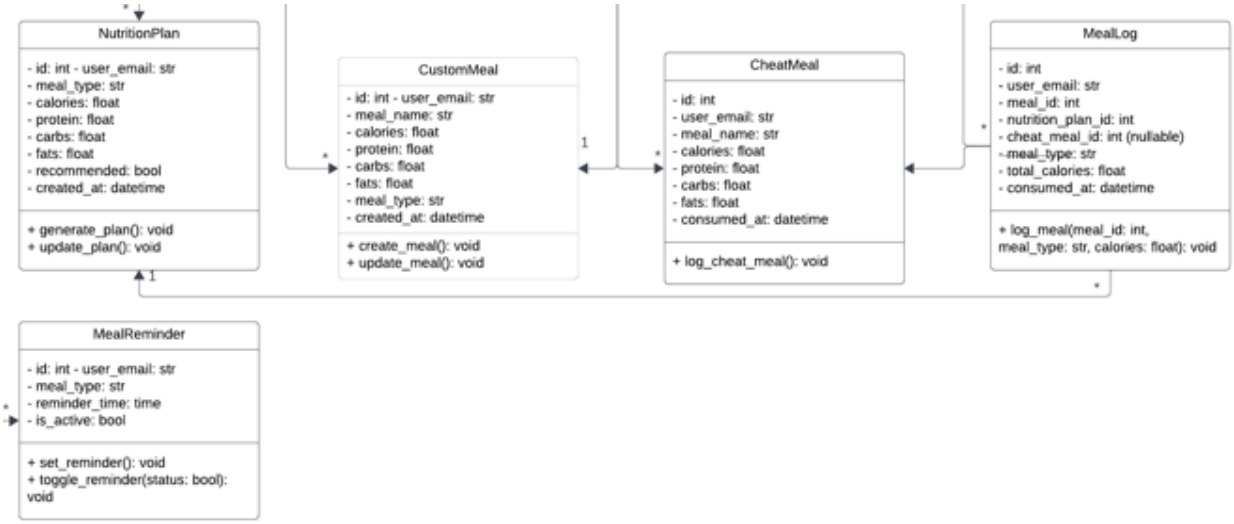
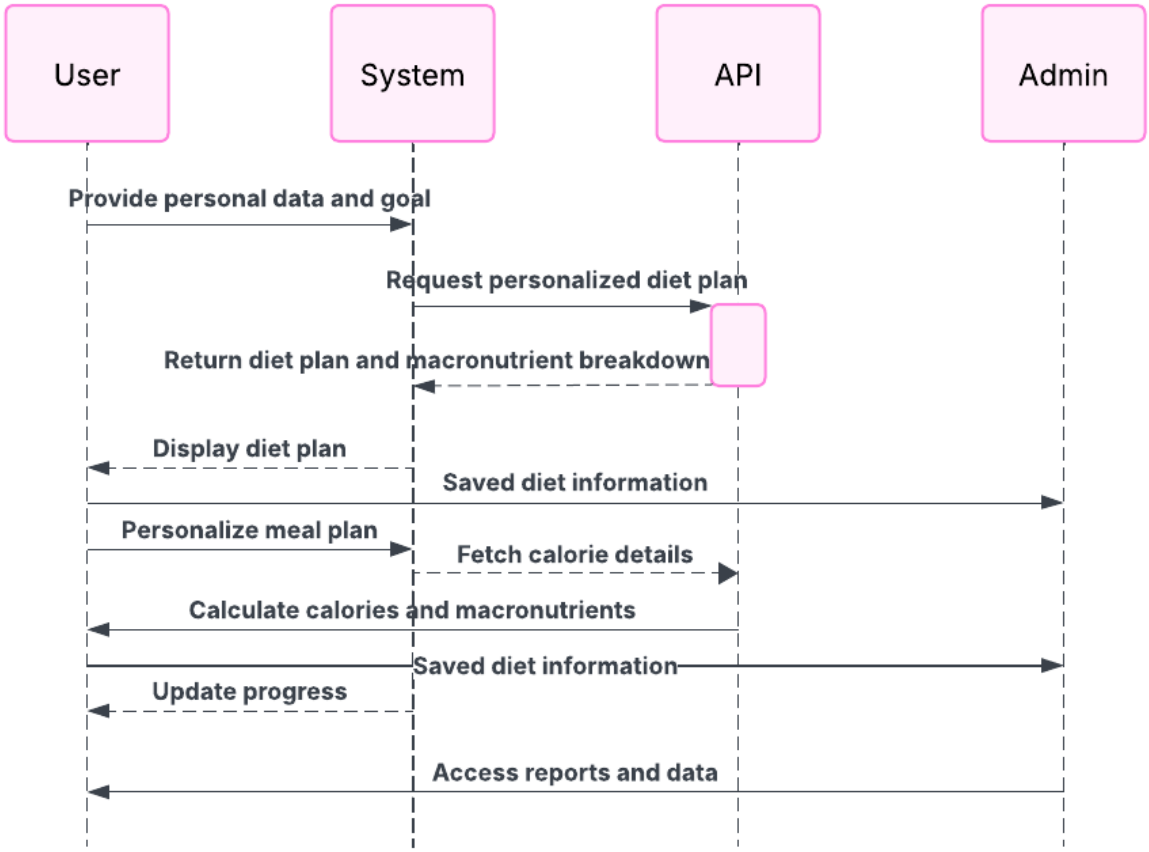


Figure 45: Class diagram for deliverables 5

VI. *Sequence Diagram*: API integration process.



-

Figure 46: Sequence diagram for deliverables 5

6.5.3. Testing:

Verify nutrition data accuracy and user experience enhancements.

Personalized Nutrition Planner					
NUT-01	Create Personalized Nutrition Plan	1. Enter user details (age, weight, goals). 2. Generate plan.	Nutrition plan generated based on user inputs.		
NUT-02	Adjust Nutrition Plan Based on Preferences	1. Set dietary preferences (e.g., vegan). 2. Regenerate plan.	Plan updated to exclude non-vegan items.		
NUT-03	Track Daily Calorie Intake	1. Log meals. 2. View calorie summary.	Calorie intake displayed accurately.		
NUT-04	Suggest Recipes Based on Goals	1. Set goal (e.g., weight loss). 2. View recipe suggestions.	Recipes aligned with weight loss goal displayed.		
Meal Reminder System					
MEAL-01	Set Meal Reminders	1. Set reminder for breakfast, lunch, dinner.	Reminders trigger at specified times.		
MEAL-02	Snooze or Dismiss Reminders	1. Snooze reminder. 2. Dismiss reminder.	Reminder snoozed for 5 minutes or dismissed permanently.		
MEAL-03	Customize Reminder Frequency	1. Change reminder frequency (e.g., every 2 hours).	Reminders trigger at new frequency.		
MEAL-04	Handle Missed Reminders	1. Ignore reminder.	Missed reminder logged and displayed in history.		

Figure 47: Testing of deliverables 5

6.6. Sub System 6: Pose Estimation

Explanation: This subsystem introduces pose estimation for tracking user movements, currently under development.

6.6.1. SRS

Specify requirements for pose detection, angle calculation, and history tracking.

6.6.2. Design/Modelling Diagrams:

I. Activity Diagram: Workflow for pose estimation process.

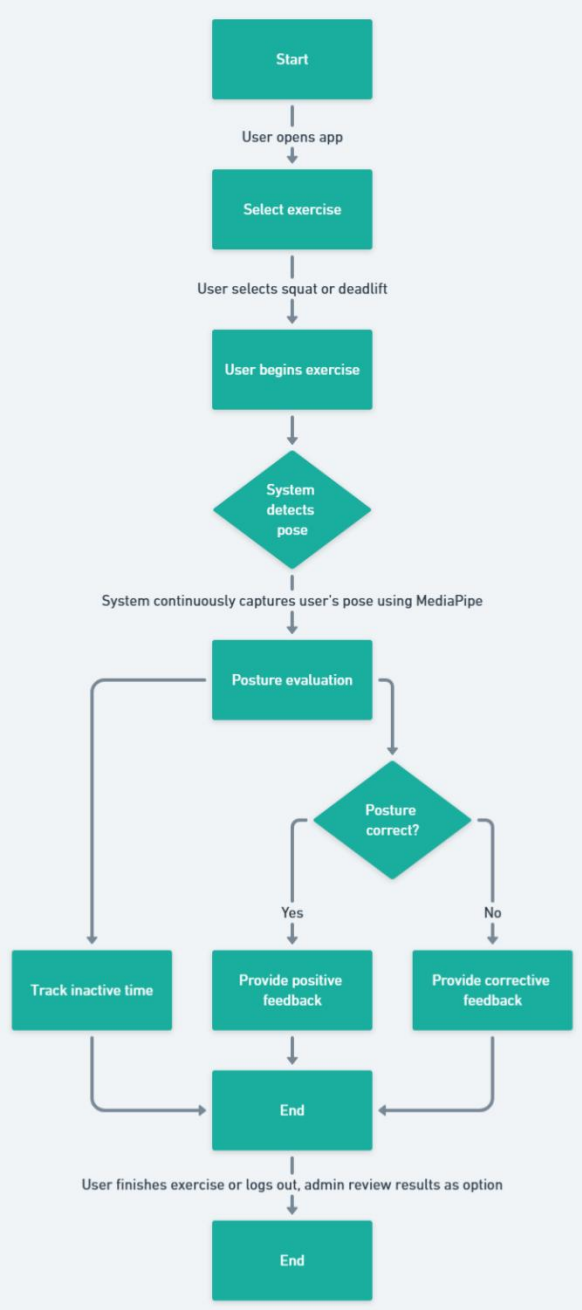


Figure 48: Activity diagram of deliverable 6

II. Wireframe: Frontend for pose estimation UI.

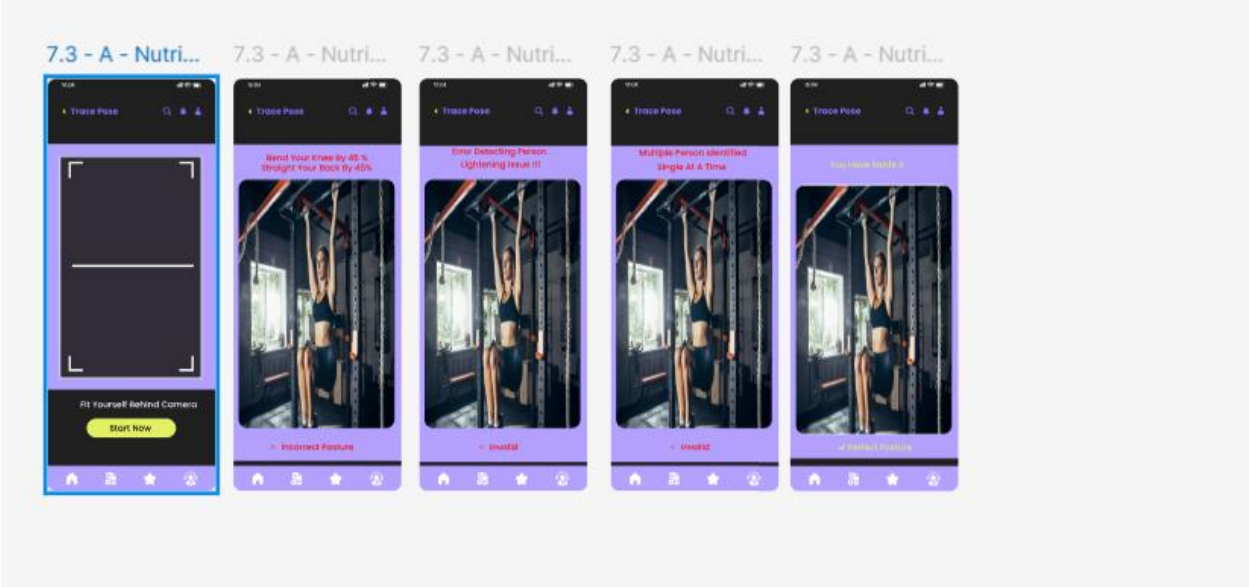


Figure 49: Wireframe for Pose Estimation part

III. Use Case Diagram: Actors (e.g., User) and use cases (e.g., "Track Pose").

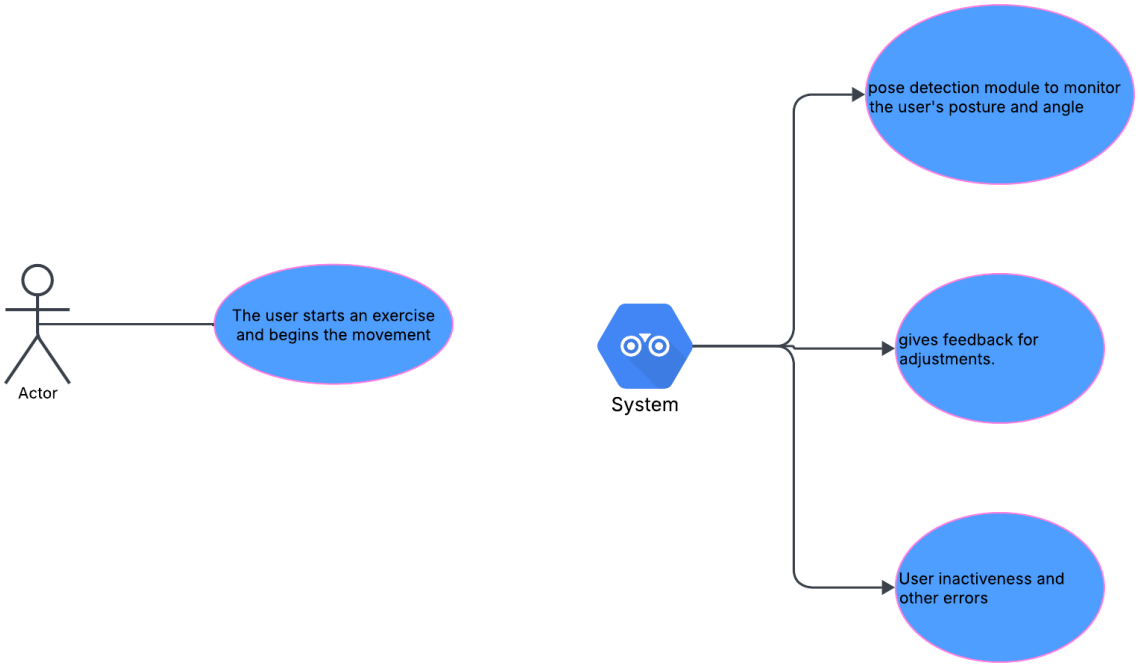


Figure 50: Use case diagram of deliverables 6

IV. ERD: Schema for pose history (to be built).

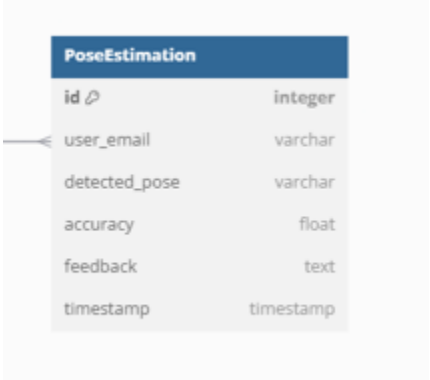


Figure 51: ERD diagram for deliverables 6

V. Class Diagram: Classes for pose estimation logic.

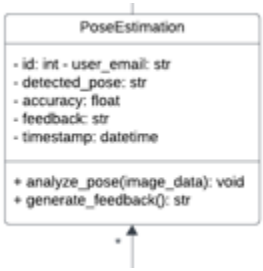


Figure 52: Class diagram for deliverables 6

VI. Sequence Diagram: Interaction between frontend, API, and backend.

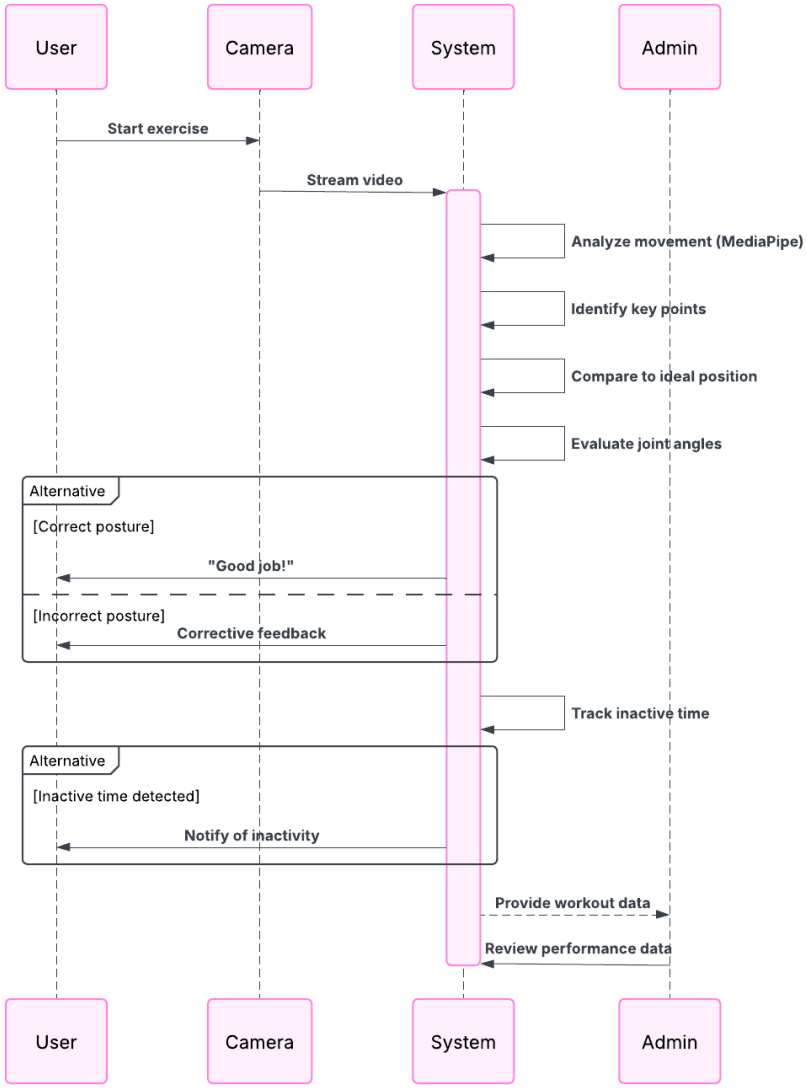


Figure 53: Sequence diagram for deliverables 6

6.6.3. Testing

Test pose detection accuracy and angle calculations (ongoing).

Real-Time Pose Estimation					
POSE-01	Detect User Pose During Exercise	Pose detected and displayed on screen.	Pose detected and displayed on screen.	Pass	Pose estimation functions correctly.
POSE-02	Provide Feedback on Incorrect Pose	Feedback displayed (e.g., "Adjust your posture").	Feedback displayed (e.g., "Adjust your posture").	Pass	Feedback is provided accurately for incorrect posture.
POSE-03	Handle Low Light Conditions	System provides warning: "Low light detected. Improve lighting."			
POSE-04	Track Multiple Users in Frame	Both users' poses detected and tracked separately.			
Feedback on Exercise Form and Posture					
FORM-01	Provide Real-Time Feedback on Form	Feedback displayed (e.g., "Keep your back straight").			
FORM-02	Track Progress on Form Improvement	Progress tracked and displayed (e.g., "Form improved by 20%").			
FORM-03	Handle No User Detected	Message displayed: "No user detected. Please step into frame."			

Figure 54: Test cases for deliverables 6

6.7. Sub System 7: Setting

Explanation: This subsystem will provide user settings and dashboards

6.7.1. SRS

This section generally includes the setting of updating user info and other dashboard and reporting section of user's performance.

6.7.2. Design and Modelling

I. Activity Diagram

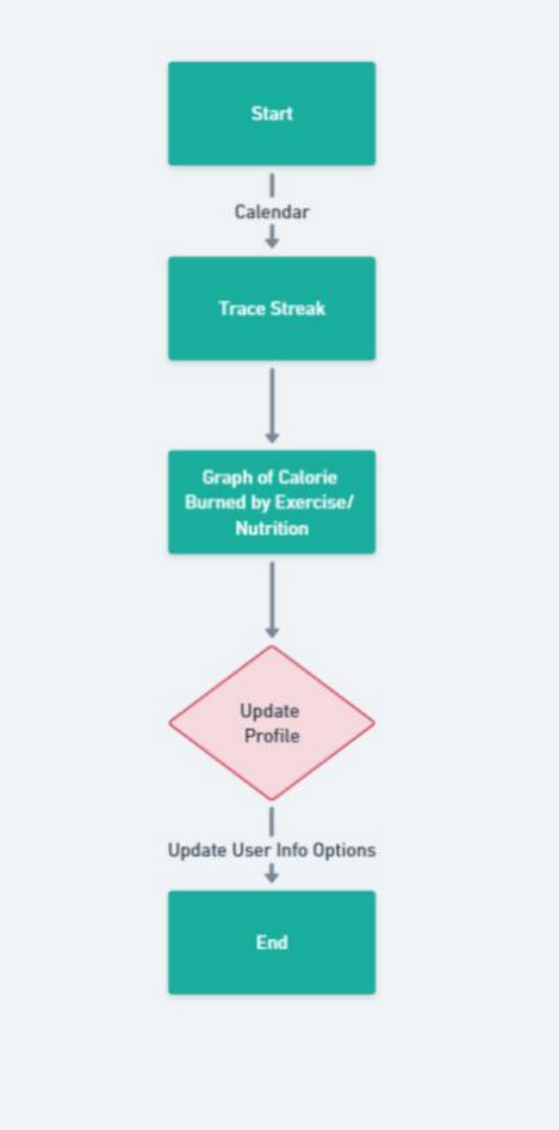


Figure 55: Activity diagram of deliverable 7

II. Wireframe design

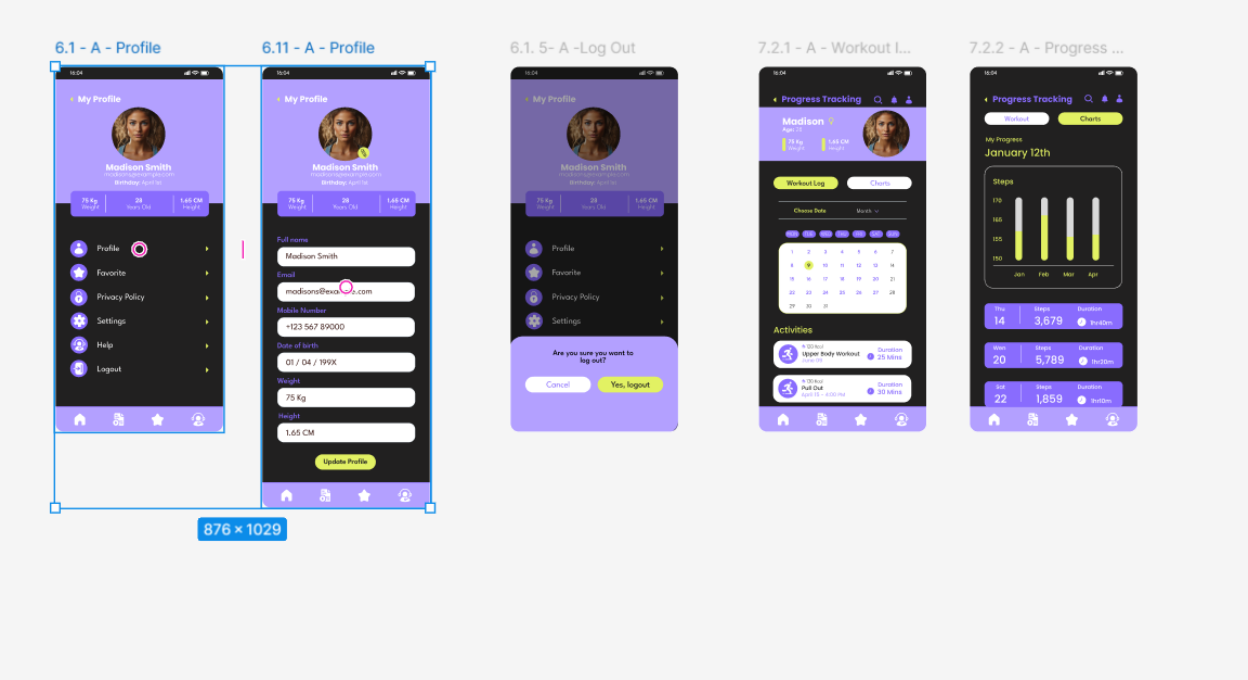


Figure 56: Wire frame for deliverable 7

III. Use Case diagram

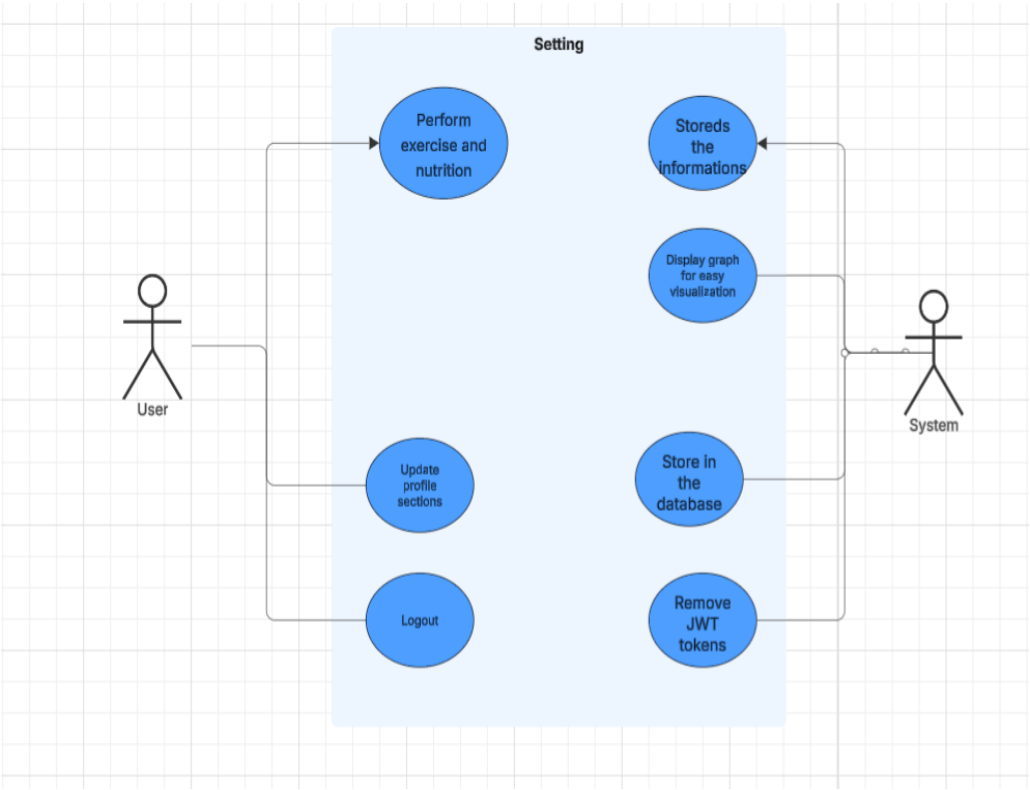


Figure 57: Use Case diagram

IV. Sequence diagram

Figure 58: Sequence diagram for deliverable

6.7.3. Testing

Testing are to be done for invalid data entries, wrong streaks and so on.

Perform Exercise and Nutrition Tracking						
Track-01	Validate user input for exercise and nutrition	1. Navigate to exercise/nutrition section. 2. Enter valid details. 3. Submit data.				
Track-02	Handle invalid input for exercise and nutrition	1. Enter incomplete/invalid data. 2. Submit data.				
Track-03	Ensure system updates progress correctly	1. Add new exercise/nutrition entry. 2. View progress graph.				
Update Profile Sections						
PROFILE-01	Update profile with valid details	1. Navigate to profile section. 2. Modify fields. 3. Save changes.				
PROFILE-02	Update profile with invalid details	1. Enter invalid data (e.g., negative weight). 2. Save changes.				
Data Storage and Visualization						
VALID-01	Ensure data is stored in the database	1. Submit new entry. 2. Check database records.				
VALID-02	Display graph with stored data	1. Open tracking section. 2. View generated graph.				

Figure 59: Test cases for deliverables 7

6.8. Bug Report

In general, a bug report is a detailed record of an issue, defect, or malfunction discovered in the time of testing in a system or application. Bug reports help developers in pinpointing, examining, and resolving errors to enhance the system's stability and performance.

During the development phase of program, I also encounter different bugs and I am using GitHub for tracking issues. Let's see through a few bugs and how I resolve the hindrances.

1. Token error during updating Profile

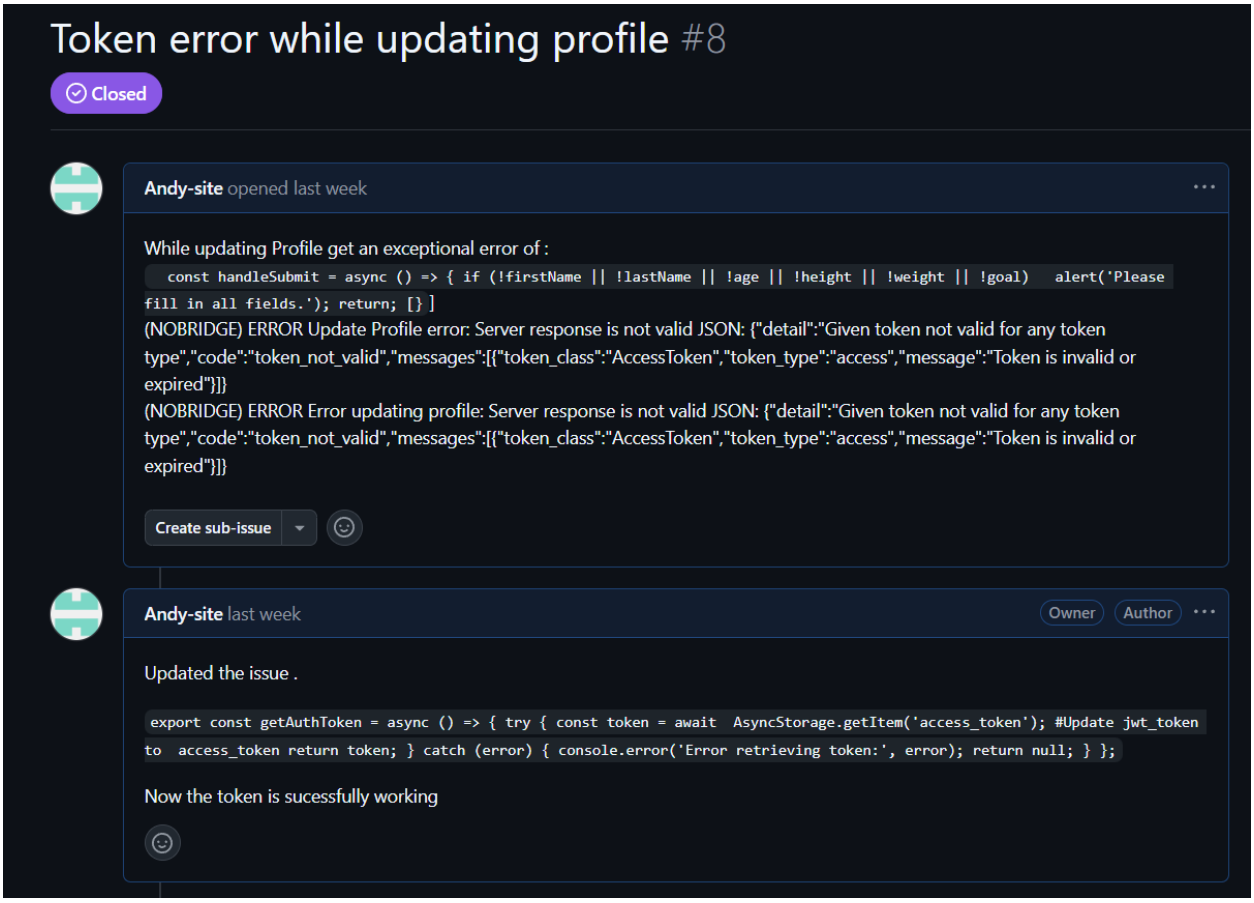


Figure 60: Tokenization bugs

I. Bug ID: #8
II. Title: Token Error While Updating Profile
III. Status: Closed
IV. Priority: High
V. Reported By: Andy-site
VI. Date Reported: Last week
VII. Environment:
VIII. Platform: React Native (Frontend)
IX. Authentication Method: JWT Token (Stored in AsyncStorage)
X. API: Django REST Framework

XI. Storage Mechanism: AsyncStorage (React Native)

XII. Description:

When attempting to update the profile, an error occurs indicating that the access token is invalid or expired. The error message suggests that the server response does not recognize the token as valid for any token type.

XIII. Steps to Reproduce:

- Navigate to the profile update section.
- Enter valid details such as name, age, height, weight, and fitness goal.
- Click on the Submit button.
- Observe the error message regarding an invalid or expired token.

XIV. Expected Result:

- The profile should be successfully updated if the token is valid.
- If the token is expired, it should automatically refresh.

XV. Actual Result:

The system returns an error stating:

(NOBRIDGE) ERROR Update Profile error: Server response is not valid JSON:

{"detail":"Given token not valid for any token type","code":"token_not_valid",

"messages":[{"token_class":"AccessToken","token_type":"access",

"message":"Token is invalid or expired"}]}

Profile update fails due to token validation error.

XVI. Root Cause Analysis:

- The token stored in AsyncStorage was either expired or not being retrieved properly before making the API request.

- The frontend was sending an outdated or null token to the backend, causing authentication failure.

XVII. Fix Implemented:

- Introduced a function to fetch the token properly before making API requests.
- Implemented the following code snippet to retrieve the token correctly:

After implementing this fix, the issue was resolved, and the token is now successfully working.

XVIII. Final Status:

Bug Fixed and Successfully Closed

2. Backend error with Home and College Wifi

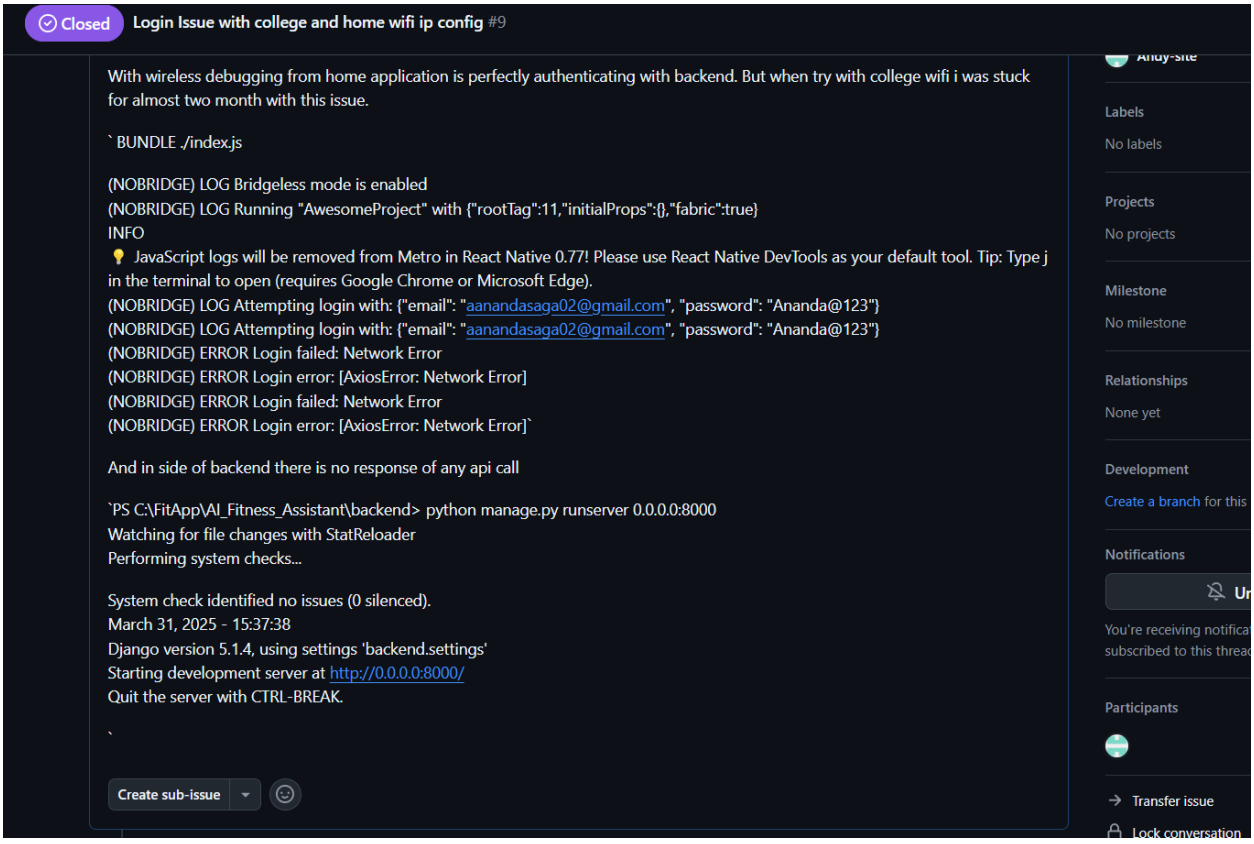


Figure 61: Backend Authentucation bug

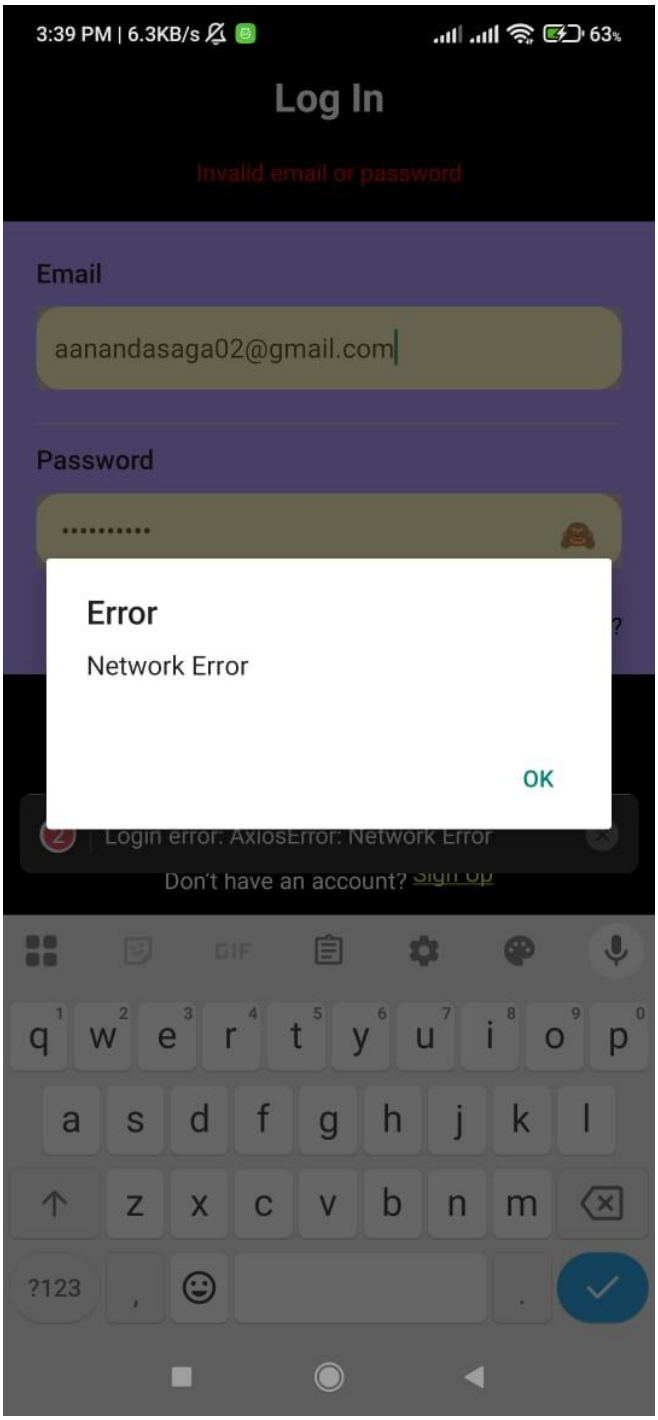


Figure 62: Backend Authentication bug

I. Bug ID: #9

II. Title: Login Issue with College and Home WiFi IP Configuration

III. Status: Open

IV. Priority: High

V. Reported By: Andy-site

VI. Date Reported: March 31, 2025

VII. Environment:

- Platform: React Native (Frontend)
- Backend: Django REST Framework
- API Requests: Axios
- Authentication: JWT Token

VIII. Description:

The login feature works perfectly when debugging wirelessly at home. However, when using the college WiFi, authentication fails, and the backend does not respond to API requests. The error log indicates a network issue (AxiosError: Network Error), preventing login.

IX. Steps to Reproduce:

- Run the Django backend:

python manage.py runserver 0.0.0.0:8000

- Use wireless debugging on the home network.
- Attempt to log in with valid credentials.
- Observe that authentication works correctly.
- Switch to the college WiFi and attempt to log in again.
- Notice that the login fails, and API calls do not reach the backend.

X. Expected Result:

- The application should successfully authenticate users regardless of the network being used.
- API requests should be processed by the backend on both home and college networks.

XI. Actual Result:

- The login attempt fails with the following error:

(NOBRIDGE) ERROR Login error: [AxiosError: Network Error]

- No API response is received in the backend logs.

The Django development server is running on 0.0.0.0:8000, but it does not receive requests when connected to the college WiFi.

XII. Root Cause Analysis:

- The home and college networks have different IP configurations.
- The API base URL was set to a static local IP (192.168.0.117:8000 for home), which does not match the college WiFi IP (192.168.64.1:8000).
- The college WiFi may have firewall restrictions preventing local API access.

XIII. Fix Implemented:

Updated API base URL handling:

- // Home IP

const API_BASE_URL = 'http://192.168.0.117:8000/api/';

- // College IP

const API_BASE_URL = 'http://192.168.64.1:8000/api/';

XIV. Solution

Comment/uncomment the relevant IP based on the network in use.

XV. Alternative Fixes:

- Used USB debugging instead of wireless debugging.
- Ran the app directly on the Android Studio emulator with the following command:

adb reverse tcp:8000 tcp:8000

- Tested with ngrok to expose the backend publicly:

ngrok http 8000

- Checked college network restrictions that may block API requests.

XIV. Final Status:

Bug Not Fixed – Further investigation is required into network firewall rules.

3. Weight update issue

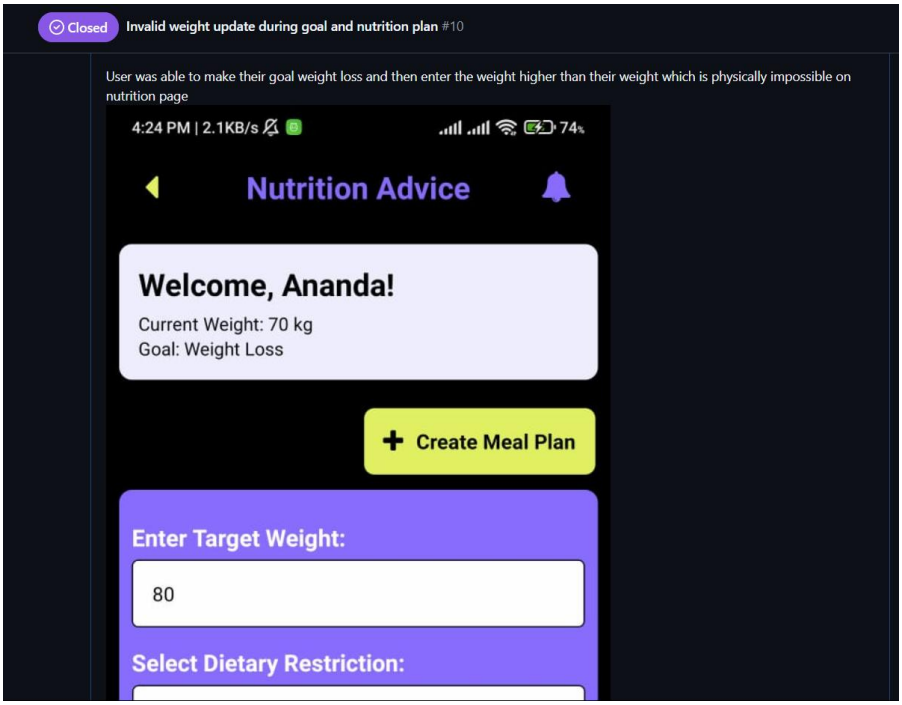


Figure 63: Issue with weight update

I. Bug ID: #10
II. Title: Invalid Target Weight Update During Goal and Nutrition Plan
III. Status: Closed
IV. Priority: Medium
V. Reported By: Andy-site
VI. Date Reported: March 31, 2025
VII. Environment:
- Platform: React Native (Frontend)
- Backend: Django REST Framework
- API Requests: Axios
- Authentication: JWT Token

VIII. Description: Users were able to set an invalid target weight for weight loss, where the target weight could be higher than their current weight, which is physically impossible. This issue was observed on the nutrition plan page when setting a weight loss goal.

IX. Steps to Reproduce:

- Navigate to the nutrition plan page.
- Set a goal to "Lose weight".
- Enter a target weight that is higher than the current weight.
- Observe that the system allows setting this invalid target weight.

X. Expected Result:

- For weight loss goals, the target weight should always be less than the current weight.

  For weight gain goals, the target weight should be higher than the current weight.

XI. Actual Result:

- The user was able to set a target weight higher than the current weight for weight loss, which is physically incorrect.

XII. Root Cause Analysis:

- The system lacked validation to check that the target weight for weight loss must be less than the current weight and for weight gain must be greater.

XIII. Fix Implemented:

- Added validation for target weight input:

XIV. Solution:

- The validation condition ensures that users can only set a valid target weight based on the selected goal (loss or gain weight).

XV. Final Status:

Bug Fixed – The issue is resolved and validated through testing.

6.9. System implements AI

6.9.1. Data Collection

The system collects real-time video frames from the user's device camera. Each frame is processed using PoseNet API to extract key body landmarks, including:

I. Joint coordinates (x, y) for key points like shoulders, elbows, and knees
II. Angle calculations to assess posture correctness
III. Repetition tracking for exercises like squats and push-ups

IV. Confidence scores to filter out unreliable predictions

6.9.2. Model Development

PoseNet API provides real-time pose estimation directly in the frontend. The workflow includes:

I. Capturing frames from the user's device camera.
II. Sending frames to PoseNet API for keypoint detection.
III. Calculating angles based on keypoints.
IV. Providing immediate visual feedback for correct/incorrect posture.

6.9.3. Optimization Evaluation

To optimize performance, the following improvements were applied:

I. Using a lightweight PoseNet model for mobile-friendly inference.
II. Optimizing pose tracking algorithms to reduce lag in movement detection.

6.9.4. AI Integration into Application

PoseNet is directly integrated into the React Native frontend:

I. Thresholds are dynamically adjusted based on Beginner/Pro mode.

Workflow:

I. The camera captures video input.
II. Real-time feedback is displayed on the screen.

6.9.5. Comparing Algorithm Performance (Test Data Evaluation)

Comparative analysis of different pose estimation models:

Algorithm	Accuracy (%)	Latency (ms)	Model Size (MB)
PoseNet (Mobile)	85.2	20	6.2

BlazePose	91.5	35	12.5
OpenPose	94.0	80	100+

PoseNet was chosen for its balance between accuracy, speed, and lightweight architecture and easier integration with my application.

6.9.6. AI Testing and Accuracy Plotting (A/B Testing)

A/B testing was conducted using:

I. PoseNet-based feedback vs. Manual coaching feedback
II. Different pose confidence thresholds affecting accuracy
III. Beginner vs. Pro modes to test adaptability
IV. User feedback was collected to assess effectiveness in real-world use.

6.9.7. Confusion Matrix

A confusion matrix was generated to analyze posture classification accuracy:

I. Higher TP & TN values indicate good model accuracy.
II. Lower FN & FP values reduce misclassifications.

6.9.8. ROC Curve

A ROC curve was plotted to analyze the trade-off between true positive rate (TPR) and false positive rate (FPR) at different confidence thresholds.

I. AUC (Area Under Curve) score measures model performance in detecting correct postures.
II. Higher AUC = better posture classification.

7. Conclusion

7.1. Referencing Aims, Objectives, and Academic Question

The gym application emerges from the idea of boosting exercise effectiveness, ensuring workout safety, delivering tailored nutrition planning, increasing user engagement, narrowing the gap between home and gym workouts, and targeting sustained fitness growth. With realization of project aims, the system adapts AI-driven pose estimation via PoseNet, a customized nutrition recommendation engine, and an engaging exercise library.

Through real-time posture monitoring, adaptive workout guidance, and individualized nutrition strategies, the project effectively represents that how AI can enhance workout safety and efficiency along with providing an intuitive and captivating user experience.

7.2. Discoveries and Key Conclusions

1. AI-Powered Pose Estimation Boosts Exercise Effectiveness and Safety

The use of PoseNet for real-time posture detection excels at spotting improper form, reducing injury risks, and directing users toward improved movement patterns. By measuring joint angles and evaluating posture precision, the system delivers immediate feedback for mobile users.

2. Personalized Nutrition Planning Enhances Workout Outcomes

The involvement of a nutrition recommendation system makes sure that users receive customized meal plans tailored to their weight, height, age, and fitness objectives. By aligning dietary habits with exercise routines, the system supports consistent fitness advancement.

I. Customized meal suggestions elevate workout performance and recovery.
II. Combining exercise monitoring with nutrition planning unites fitness and diet management seamlessly.

3. Improved User Engagement and Sustained Fitness Growth

The Browse Library feature heightens user engagement by organizing exercises according to difficulty, equipment needs, and targeted muscle groups. This setup encourages users to explore varied workouts, sustaining their motivation and involvement.

I. Users monitor their workout progress, promoting regularity.
II. A structured progression framework drives ongoing improvement.
III. The React Native frontend delivers a fluid and interactive interface, ensuring effortless navigation.

7.3. Conclusion

This gym application strongly demonstrates AI's role in fitness tracking, representing that real-time posture detection and personalized nutrition planning guarantee advancement in workout safety and efficiency. The result of this project ignite the groundwork for future enhancements, such as expanding exercise tracking features, incorporating with wearable technology, and surging AI-driven movement correction.

In a nutshell, this application is built as a scalable, AI-powered fitness tool that allow users to train effectively, safely, and in a personalized way, rendering AI-driven fitness technology more accessible and transformative for today's users.

8. Critical Evaluation of the Project

8.1. Final Report Reflection

The final report demonstrates the in-depth examination of my AI-driven gym application, including exercise tracking, nutrition planning, and user engagement strategies. It sequentially shows the deployment of PoseNet for real-time posture estimation, tailored meal suggestions, and a dynamic workout library.

A key strength lies in the report's systematic framework for documenting the system's development of the lifecycle. It thoroughly outlines the aims, objectives, system architecture, implementation phases, and assessment of machine learning-driven

posture detection. By incorporating comparative performance metrics, testing protocols, and user-centric engagement tactics, the report remains both perceptive and rigorously researched.

Moreover, the report convincingly illustrates AI's transformative role in fitness applications by tackling the academic question: "How do AI-driven pose estimation and personalized fitness recommendations elevate workout effectiveness and safety?" The insights establish a robust platform for future scalability and enhancements, positioning the report as a critical asset for advancing AI-powered fitness solutions.

8.2. Findings and Process

The project unveils several pivotal discoveries about AI-enhanced exercise tracking and nutrition planning. Integrating PoseNet demonstrates exceptional efficacy in real-time pose estimation, empowering users to refine their workout techniques and reduce injury risks. The testing stage verifies that real-time monitoring markedly boosts user awareness and interaction, offering instant feedback that supports maintaining correct posture.

Moving along, another prime finding emphasizes the importance of personalized nutrition planning in optimizing fitness outcomes. The development journey also sheds light on the complexities of deploying real-time AI in mobile ecosystems. Fine-tuning PoseNet's performance to balance low latency with computational efficiency demands meticulous optimization. The trade-off between precision and responsiveness emerges as a pivotal consideration for delivering a fluid and engaging user experience, a challenge that will guide future iterations.

8.3. System Evaluation

The gym application fulfills its core objectives, providing a sturdy and interactive fitness tracking platform. Its machine learning-powered posture detection, customized nutrition planning, and organized exercise categorization coalesce into a comprehensive fitness ecosystem.

Key strengths of the system include:

1. Real-time posture analysis

Users gain immediate insight into their movements, enabling instant form correction and technique enhancement.

2. Seamless nutrition tracking integration

Beyond exercise focus, the system adopts a holistic approach by embedding tailored meal planning.

3. User engagement features

The Browse Library elevates usability by sorting exercises by difficulty, equipment needs, and muscle groups, streamlining workout selection.

Despite these strengths, the system encounters limitations. Additionally, PoseNet's precision in detecting intricate movements leaves room for improvement, as some complex postures lack high accuracy. Future enhancements might explore alternative models like BlazePose to bolster detection prowess and scalability.

8.4. Planning, Management, and Quality of Sources

The project adheres to a meticulously structured development methodology, weaving in incremental advancements and iterative testing cycles. Leveraging React Native for the frontend and Django for backend development (though not utilized for PoseNet), the system boasts a scalable and maintainable codebase.

Regarding source quality, the project draws on credible AI frameworks, well-documented APIs, and evidence-based implementation strategies. Academic literature, machine learning studies on posture estimation, and fitness tracking research lend robustness to the findings and system design.

Effective time management proves instrumental in ensuring that components like exercise tracking, nutrition planning, and the user interface undergo thorough implementation and validation.

8.5. Self-Reflection

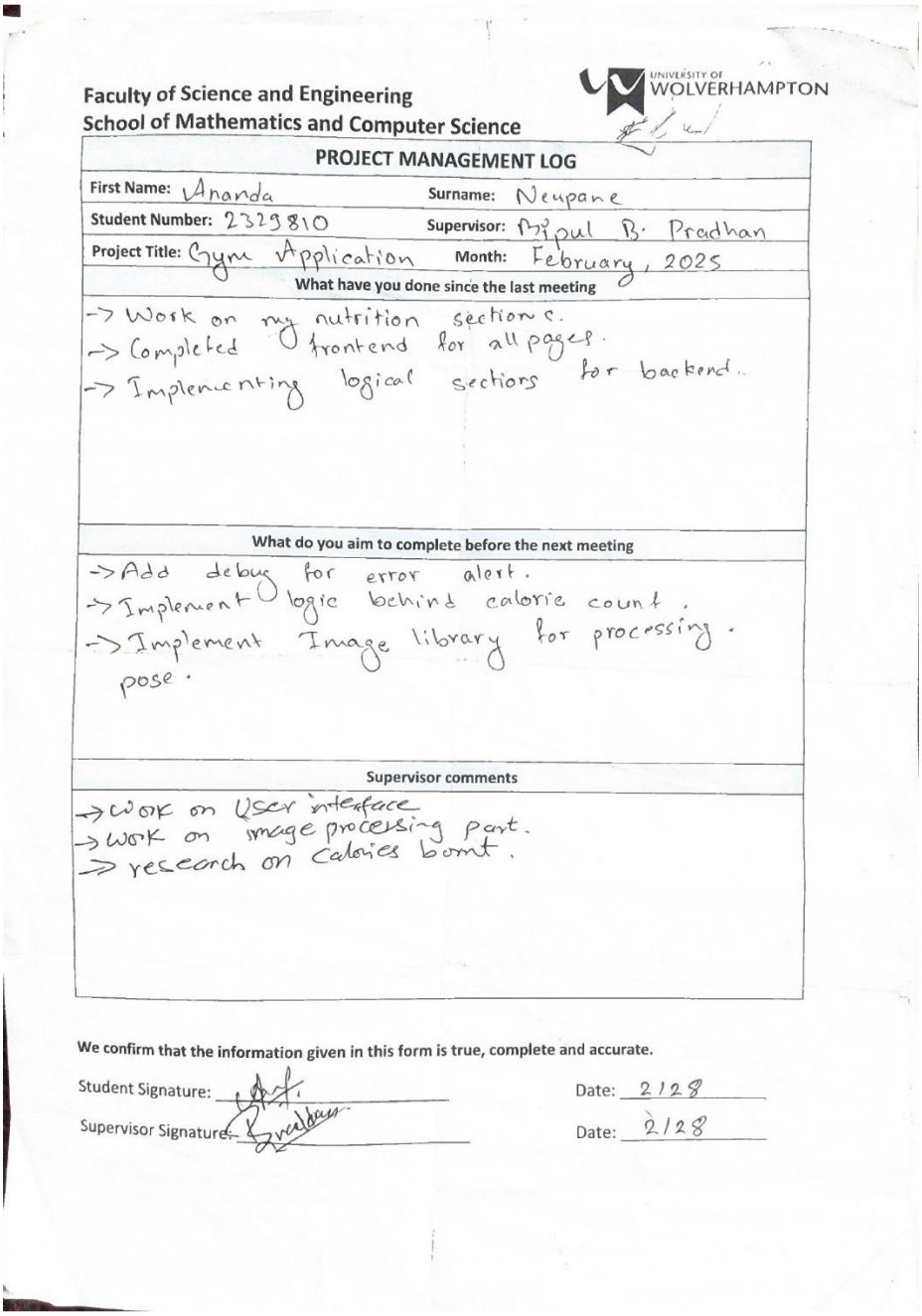
This project enriches my journey, offering profound insights into AI-driven fitness applications and computer vision methodologies. Implementing PoseNet for posture estimation deepens my grasp of machine learning, real-time tracking, angle computations, confidence scores, and optimization techniques.

On a personal note, this project development journey sharpens my problem-solving capability, refines my project management skills, and surges my critical thinking. Tackling AI model debugging, performance tuning, and user experience optimization fosters a disciplined approach to machine learning deployment.

In a nutshell, the creation of this AI-powered gym application emerges as a triumphant and illuminating experience. By integrating real-time pose estimation, personalized nutrition planning, and interactive workout guidance, the system delivers a holistic fitness solution that advances user engagement and workout efficiency.

9. Evidence of Project Management

9.1. Log Sheets



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Ananda	Surname: Neupane
Student Number: 2329810	Supervisor: Bipul B. Pradhan
Project Title: Gym Application	Month: February, 2025

What have you done since the last meeting

-> Work on my nutrition sections.
-> Completed frontend for all pages.
-> Implementing logical sections for backend.

What do you aim to complete before the next meeting

->Add debug for error alert.
->Implement logic behind calorie count.
->Implement Image library for processing. pose.

Supervisor comments

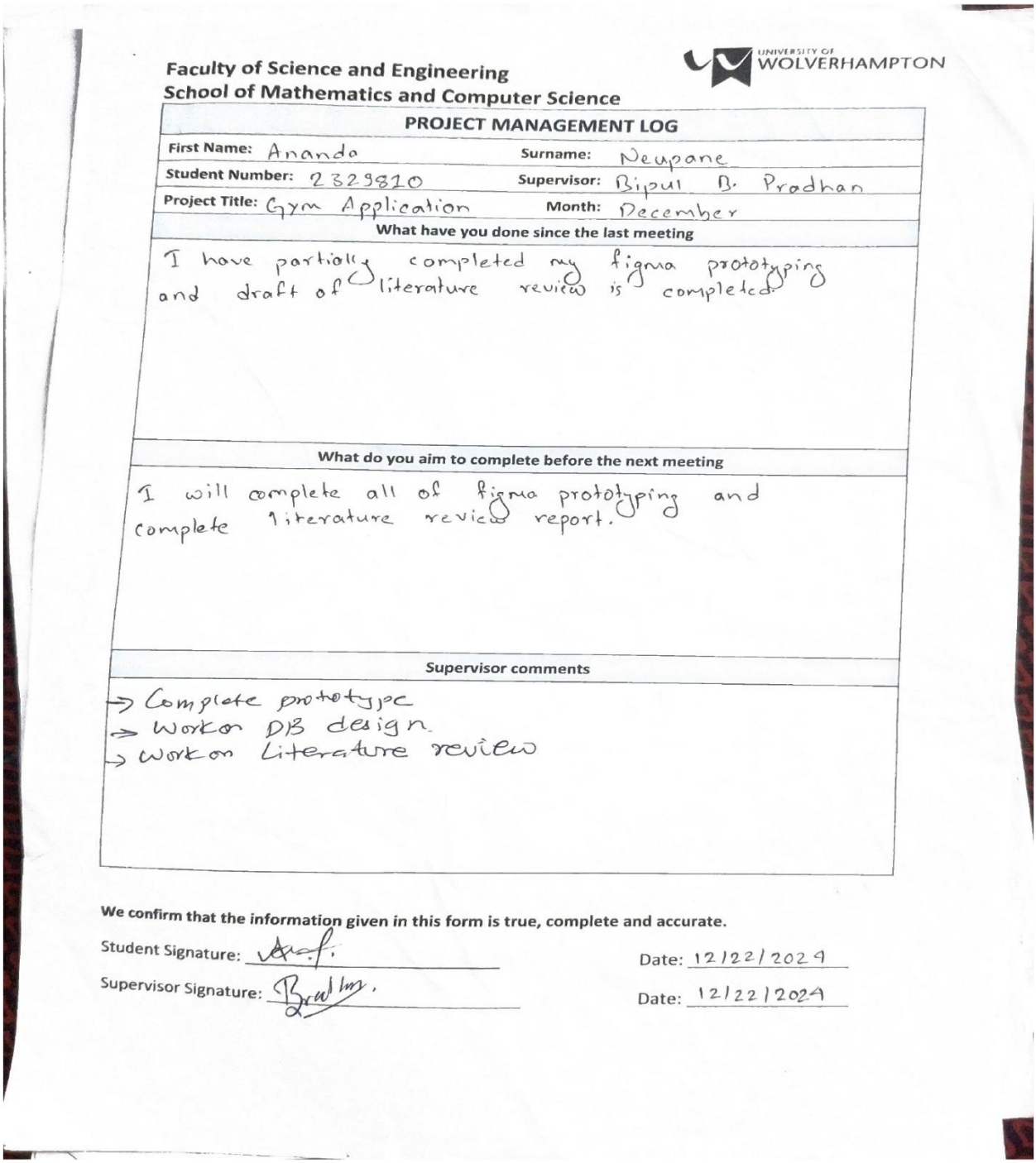
->Work on User interface
->Work on image processing part.
->research on calories burnt.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: _____ Date: 2/28

Supervisor Signature: _____ Date: 2/28

Figure 64: LogSheet 01



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name:	Ananda	**Surname:**	Neupane
Student Number:	2329810	**Supervisor:**	Bipul B. Pradhan
Project Title:	Gym Application	**Month:**	December

What have you done since the last meeting

I have partially completed my figma prototyping and draft of literature review is completed.

What do you aim to complete before the next meeting

I will complete all of figma prototyping and complete literature review report.

Supervisor comments

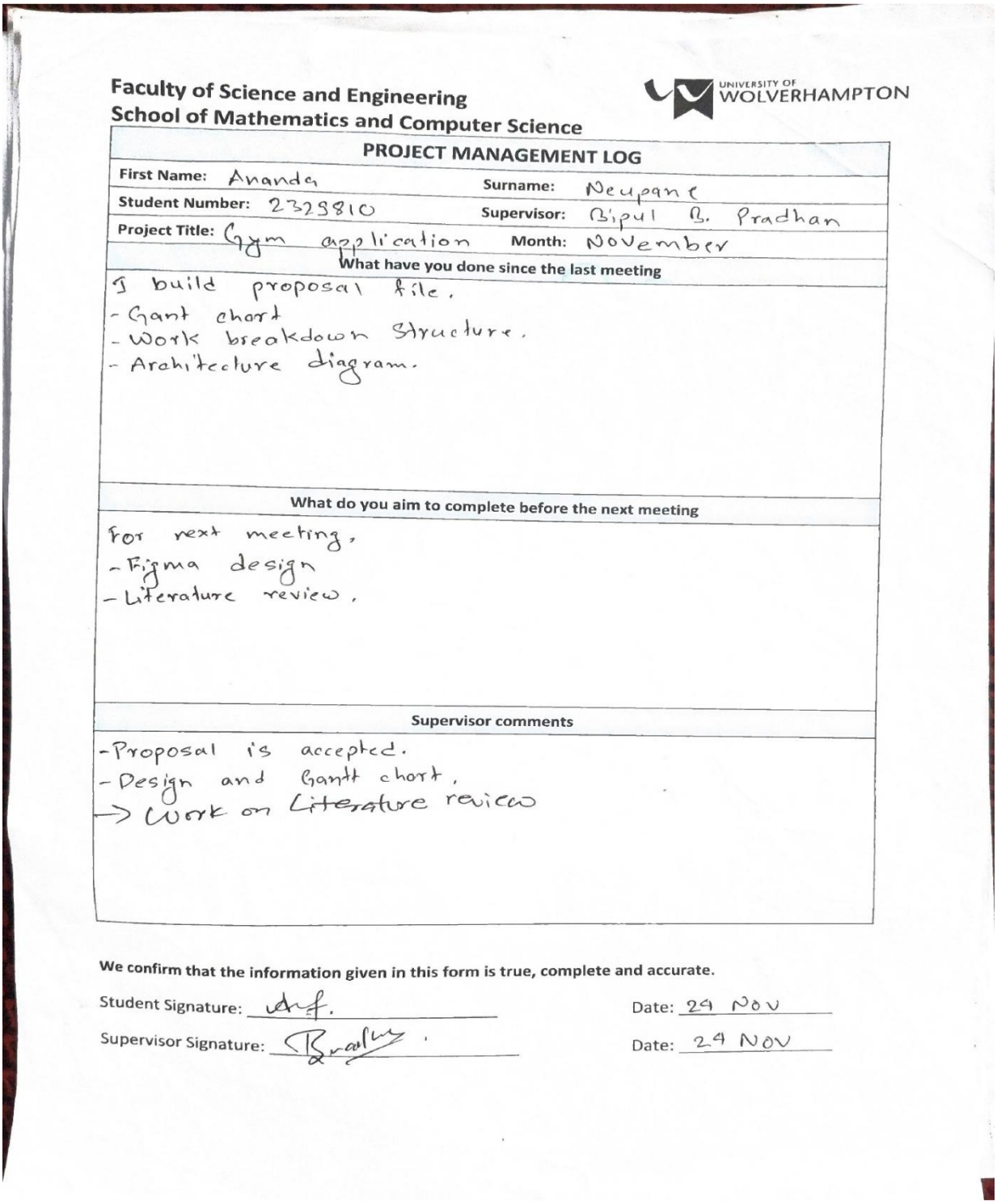
→ Complete prototype
→ Work on DB design
→ Work on Literature review

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Ananda Date: 12/22/2024

Supervisor Signature: Bipul Date: 12/22/2024

Figure 65: Logsheet 02



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Ananda	**Surname:** Neupane
Student Number: 2329810	**Supervisor:** Bipul B. Pradhan
Project Title: Gym application	**Month:** November

What have you done since the last meeting

I build proposal file.
- Gant chart
- Work breakdown structure.
- Architecture diagram.

What do you aim to complete before the next meeting

For next meeting,
- Figma design
- Literature review.

Supervisor comments

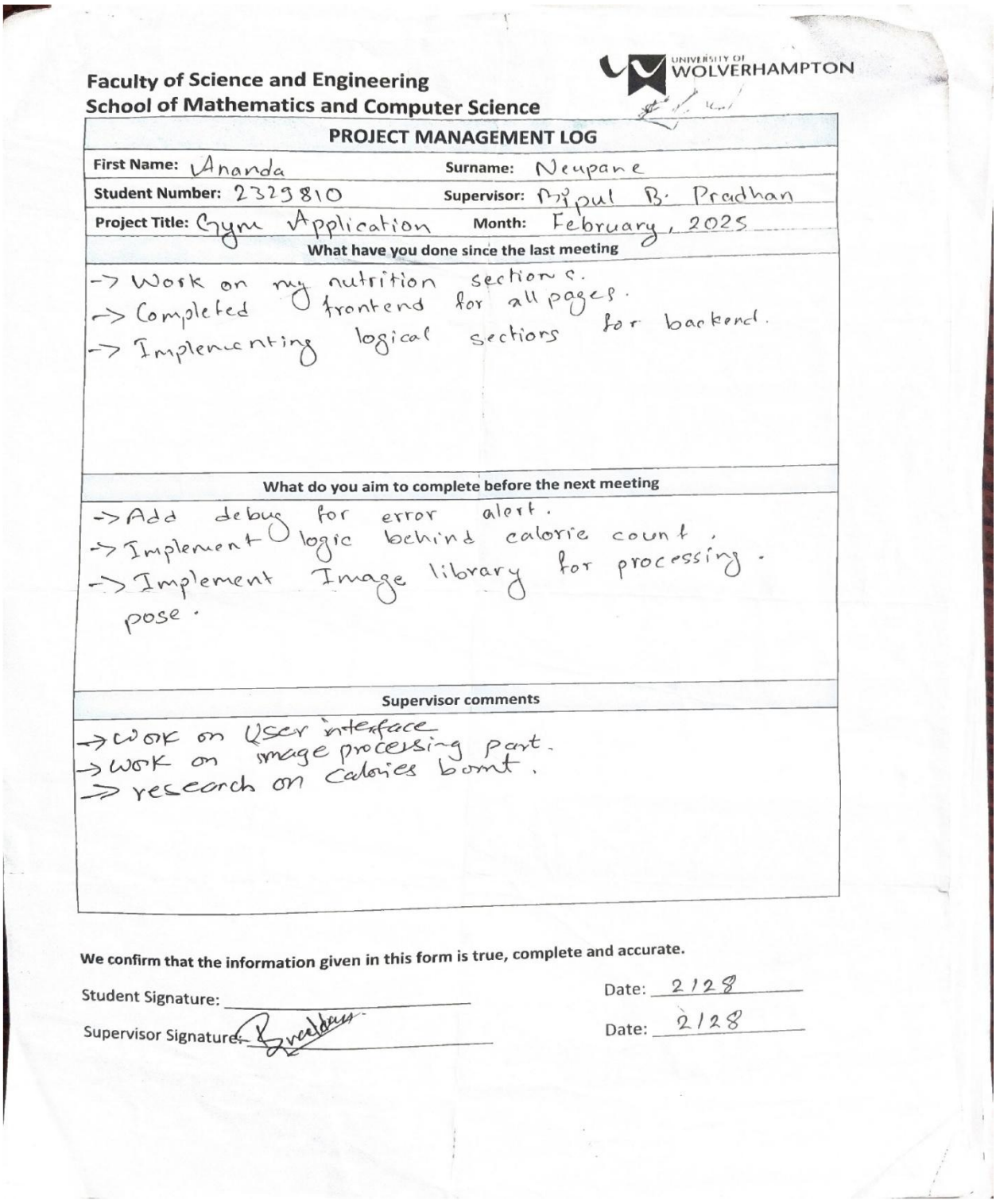
- Proposal is accepted.
- Design and Gantt chart,
→ Work on Literature review

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Anf. Date: 29 Nov

Supervisor Signature: Bradley Date: 29 Nov

Figure 66: Logsheet 03



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Ananda	**Surname:** Neupane
Student Number: 2329810	**Supervisor:** Bipul B. Pradhan
Project Title: Gym Application	**Month:** February, 2025

What have you done since the last meeting

-> Work on my nutrition sections.
-> Completed frontend for all pages.
-> Implementing logical sections for backend.

What do you aim to complete before the next meeting

-> Add debug for error alert.
-> Implement logic behind calorie count.
-> Implement Image library for processing pose.

Supervisor comments

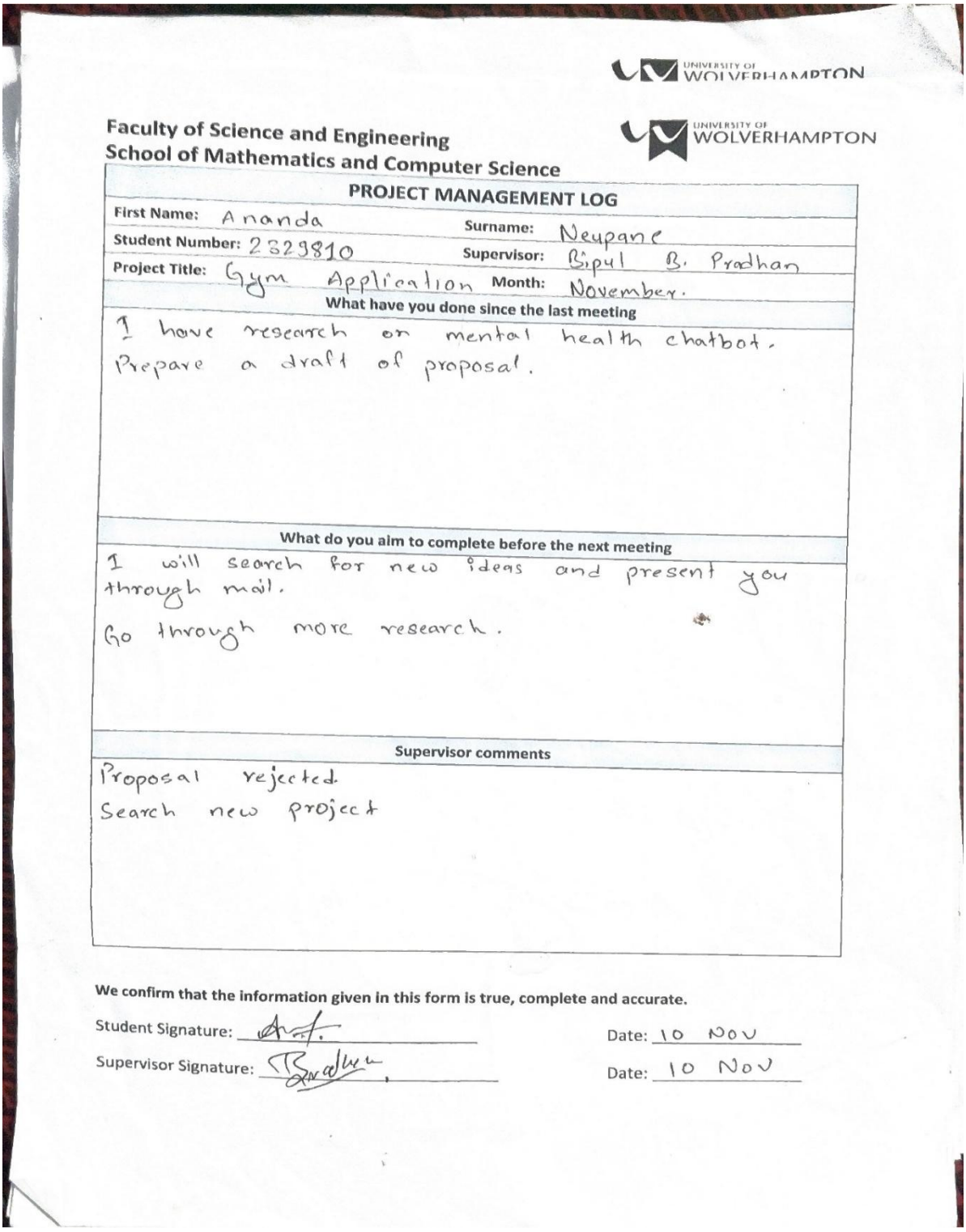
-> work on User interface
-> work on image processing part.
-> research on calories bornt.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: _____ Date: 2/2/8

Supervisor Signature: _____ Date: 2/2/8

Figure 67: Logsheet 04



UNIVERSITY OF WOLVERHAMPTON

Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name:	Ananda	**Surname:**	Neupane
Student Number:	2329810	**Supervisor:**	Bipul B. Pradhan
Project Title:	Gym Application	**Month:**	November.

What have you done since the last meeting

I have research on mental health chatbot.
Prepare a draft of proposal.

What do you aim to complete before the next meeting

I will search for new ideas and present you through mail.

Go through more research.

Supervisor comments

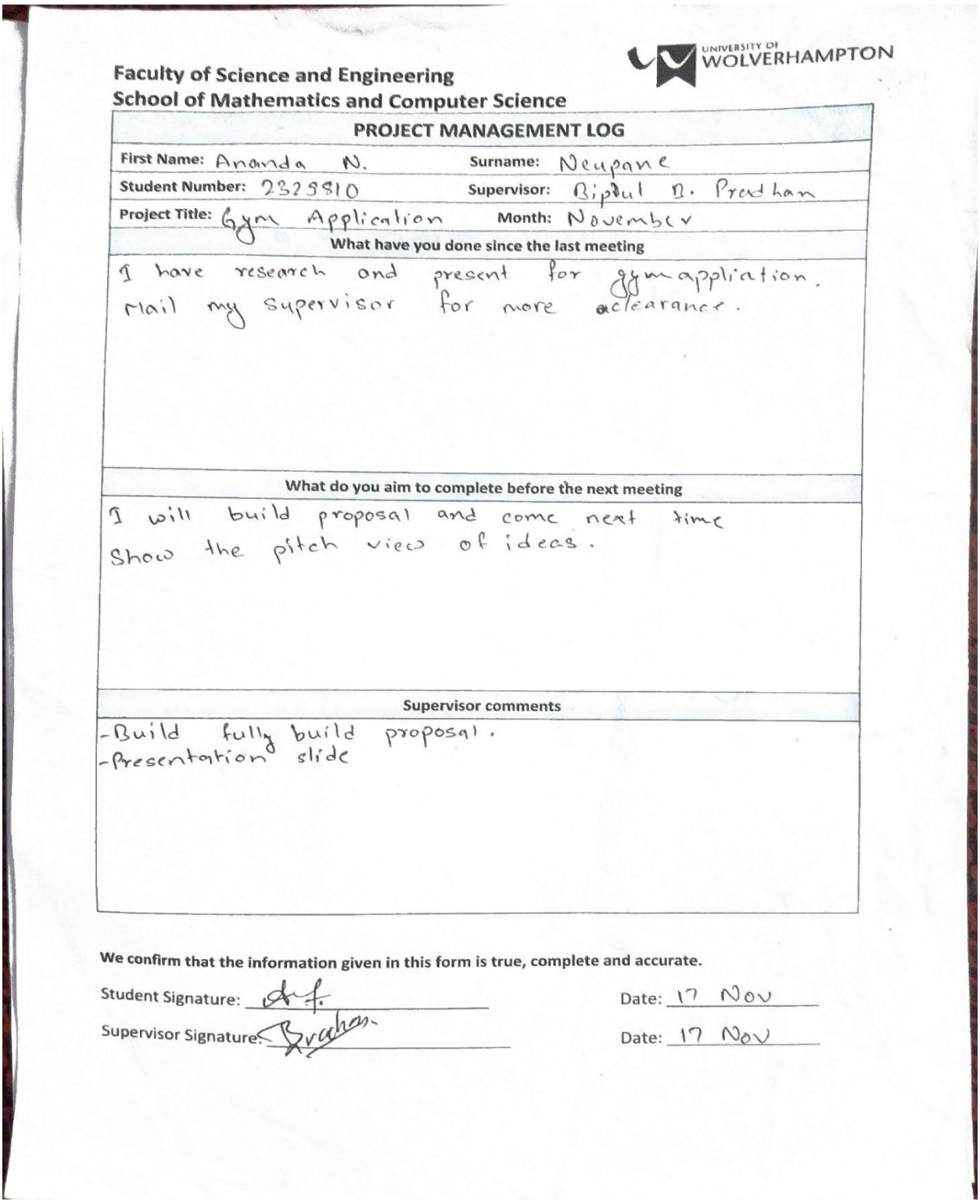
Proposal rejected.
Search new project

We confirm that the information given in this form is true, complete and accurate.

Student Signature: _____ Date: 10 Nov

Supervisor Signature: _____ Date: 10 Nov

Figure 68: Logsheet 05



UNIVERSITY OF WOLVERHAMPTON

Faculty of Science and Engineering
School of Mathematics and Computer Science

PROJECT MANAGEMENT LOG

First Name: Ananda N.	**Surname:** Neupane
Student Number: 2329510	**Supervisor:** Bipul B. Pradhan
Project Title: Gym Application	**Month:** November

What have you done since the last meeting

I have research and present for gym appliation. Mail my supervisor for more aclearance.

What do you aim to complete before the next meeting

I will build proposal and come next time Show the pitch view of ideas.

Supervisor comments

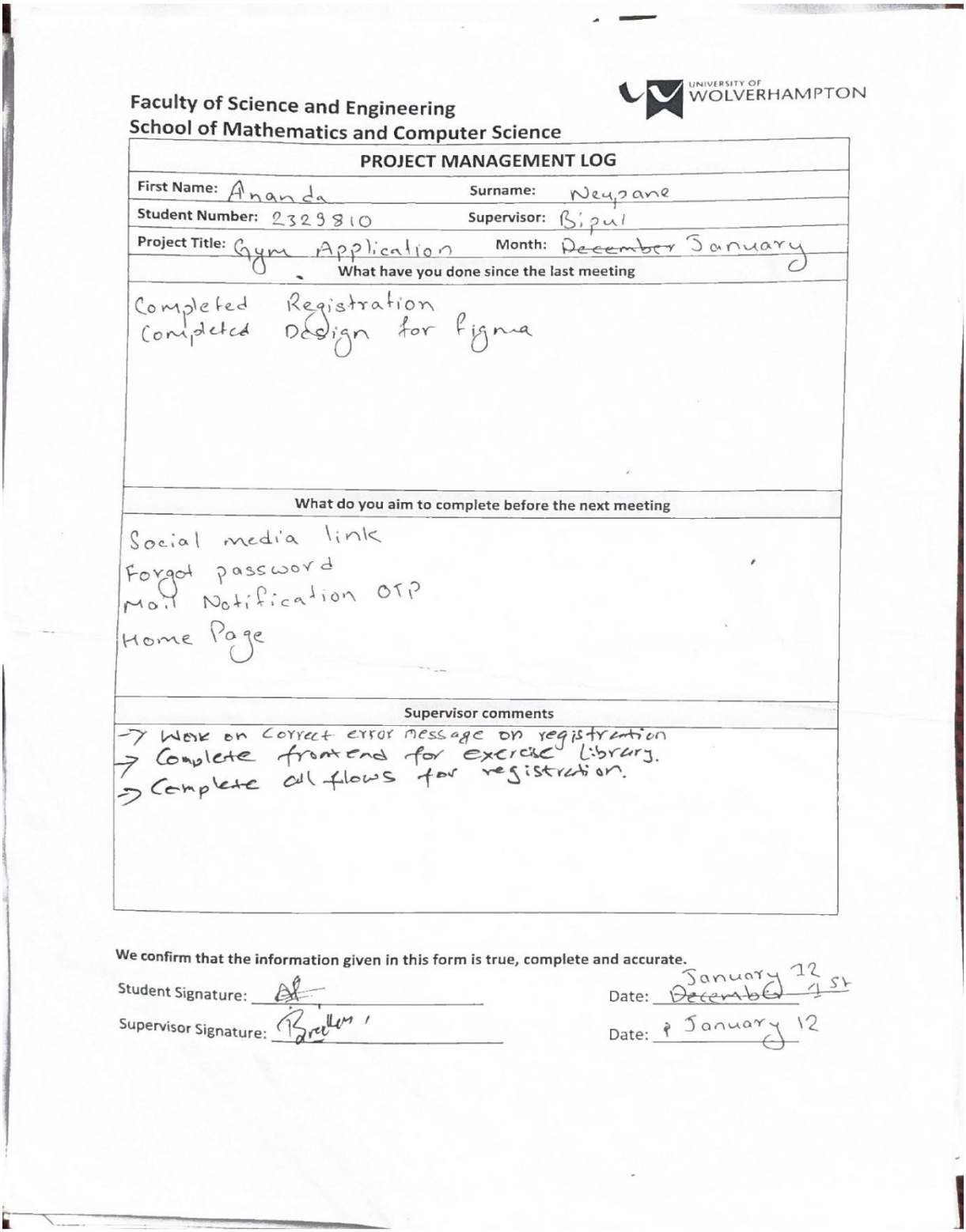
-Build fully build proposal.
-Presentation slide

We confirm that the information given in this form is true, complete and accurate.

Student Signature: _____ Date: 17 Nov

Supervisor Signature: _____ Date: 17 Nov

Figure 69: Logsheet 06



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Ananda Surname: Neupane

Student Number: 2329810 Supervisor: Bipul

Project Title: Gym Application Month: ~~December~~ January

What have you done since the last meeting

Completed Registration
Completed Design for figma

What do you aim to complete before the next meeting

Social media link
Forgot password
Mail Notification OTP
Home Page

Supervisor comments

→ Work on correct error message on registration
→ Complete frontend for exercise library.
→ Complete all flows for registration.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: AP Date: ~~December~~ January 12 1st

Supervisor Signature: Bipul Date: 9 January 12

Figure 70: Logsheet 07

UNIVERSITY OF WOLVERHAMPTON

Faculty of Science and Engineering
School of Mathematics and Computer Science

PROJECT MANAGEMENT LOG

First Name:	Ananda	**Surname:**	Neupane
Student Number:	2329810	**Supervisor:**	Bipul
Project Title:	Gym Application	**Month:**	January

What have you done since the last meeting

Fix front end error alert message.
Completed frontend exercise artifact section.
Registration flows is completed

What do you aim to complete before the next meeting

Add few more pages for advancemenr
Add debugging for backends.
Implement the backend section first.

Supervisor comments

→ Complete on ~~frontend~~ frontend for all scenarios.
→ Error messages completed.
→ Work on backend.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: _____ Date: 1/19

Supervisor Signature: _____ Date: 1/19

Figure 71: Logsheet 08

Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Ananda	Surname: Neupane
Student Number: 2329810	Supervisor: Bipul B. Pradhan.
Project Title: Gym Application.	Month: March.

What have you done since the last meeting

-> Advance few pages UI
-> Research for image processing ML documentation.
-> Stuck on camera processing for real time.

What do you aim to complete before the next meeting

-> Complete research of image processing.
-> Implement UI for all deployment.
-> Add some user UI enhancement for deliveries.

Supervisor comments

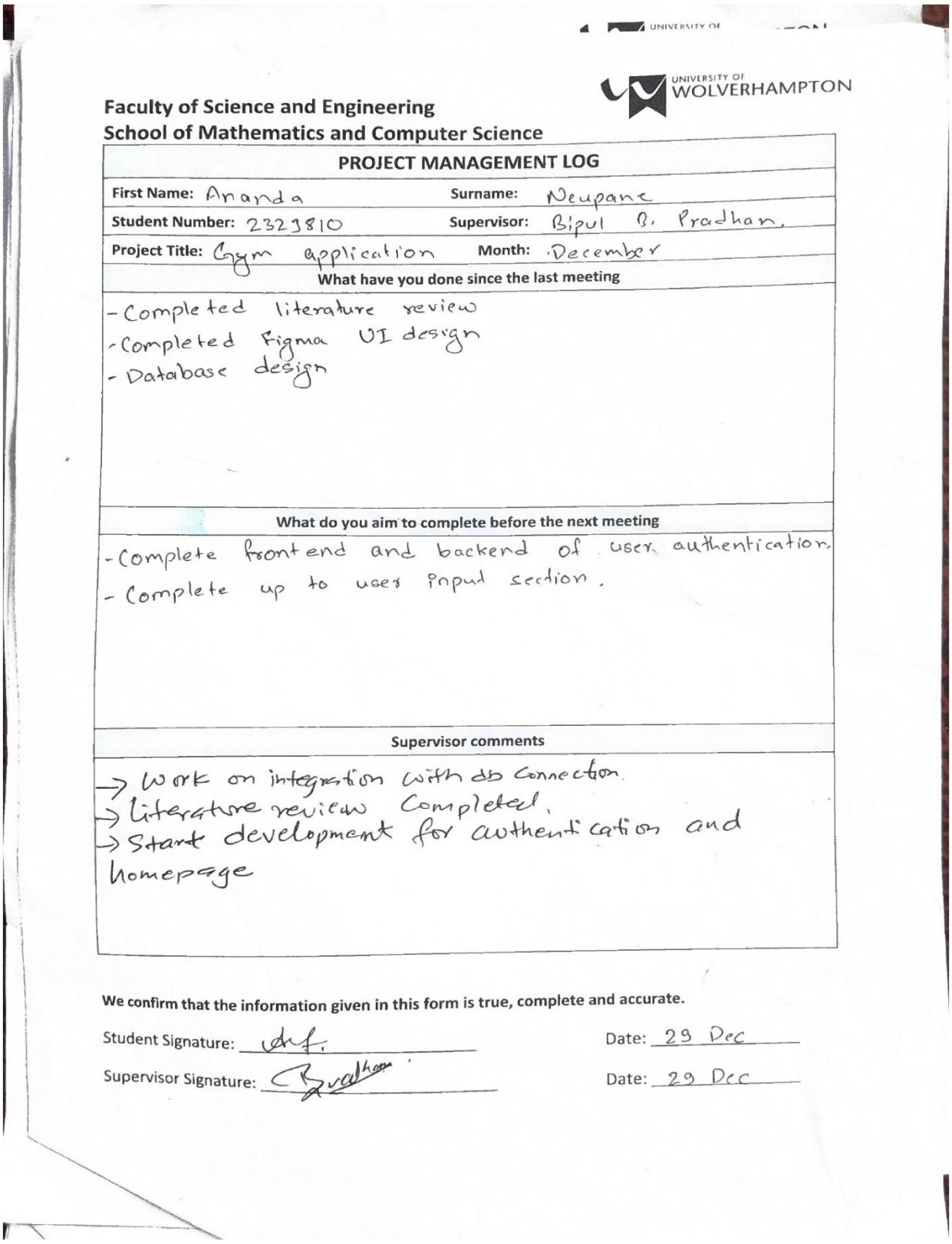
-> Work on Camera access library.
-> Explore on other Camera library.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: A.N. Date: 23 March.

Supervisor Signature: Bipul Pradhan. Date: 23rd 2025.

Figure 72: Logsheet 09



UNIVERSITY OF WOLVERHAMPTON

Faculty of Science and Engineering
School of Mathematics and Computer Science

PROJECT MANAGEMENT LOG

First Name: Ananda	**Surname:** Neupane
Student Number: 2323810	**Supervisor:** Bipul B. Pradhan.
Project Title: Gym application	**Month:** December

What have you done since the last meeting

- Completed literature review
- Completed figma UI design
- Database design

What do you aim to complete before the next meeting

- Complete frontend and backend of user authentication.
- Complete up to users input section.

Supervisor comments

→ Work on integration with db connection.
→ literature review Completed.
→ Start development for authentication and homepage

We confirm that the information given in this form is true, complete and accurate.

Student Signature: _____ Date: 29 Dec

Supervisor Signature: _____ Date: 29 Dec

Figure 73: Logsheet 10

UNIVERSITY OF WOLVERHAMPTON

Faculty of Science and Engineering
School of Mathematics and Computer Science

PROJECT MANAGEMENT LOG

First Name: Ananda	**Surname:** Neupane
Student Number: 2329810	**Supervisor:** Bipul. B. Pradhan
Project Title: Gym Application	**Month:** March

What have you done since the last meeting

->Integration of pose estimation with frontend.
->Working on Final draft report.

What do you aim to complete before the next meeting

-> Backend integration for pose estimation.
-> Completed final report.
-> Complete and improve user experience.

Supervisor comments

-> Work on target set.
-> Work on image processing.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: _____ Date: 30 March

Supervisor Signature: _____ Date: 30 March

Figure 74: Logsheet 11

UNIVERSITY OF WOLVERHAMPTON

Faculty of Science and Engineering
School of Mathematics and Computer Science

PROJECT MANAGEMENT LOG

First Name: Ananda	**Surname:** Neupane
Student Number: 2329810	**Supervisor:** Bipul B. Pradhan
Project Title: Gym Application	**Month:** Janvary.

What have you done since the last meeting

-> Complete working on backend for exercise browsing.
-> Present idea of designing nutrition plan.

What do you aim to complete before the next meeting

-> Start building for database of nutrition section.
-> Integrating and testing of backend for nutrition.

Supervisor comments

-> start building database design.
-> Integrate and test for backend.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: A.N. Date: 17 Jan

Supervisor Signature: Brady Date: 17 Jan

Figure 75: Logsheet 12

9.2. Gantt Chart

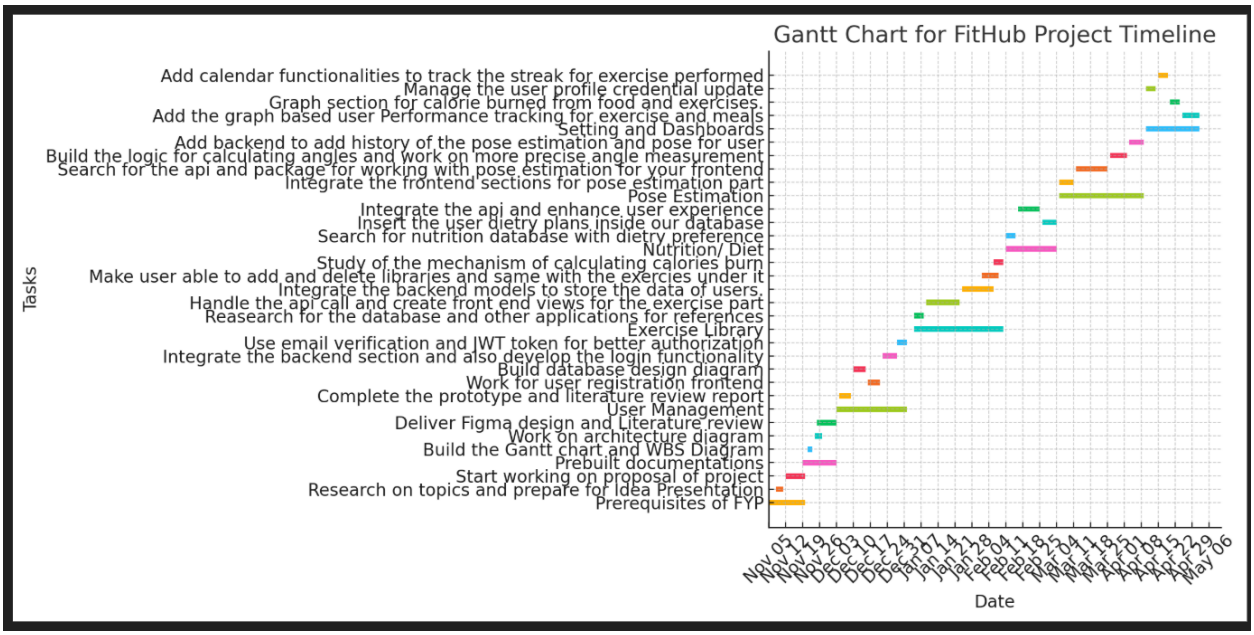


Figure 76: Gantt Chart

10. User Manual

On launching of application, you will meet some landing screen and you got skip button if you don't want to watch all of them.

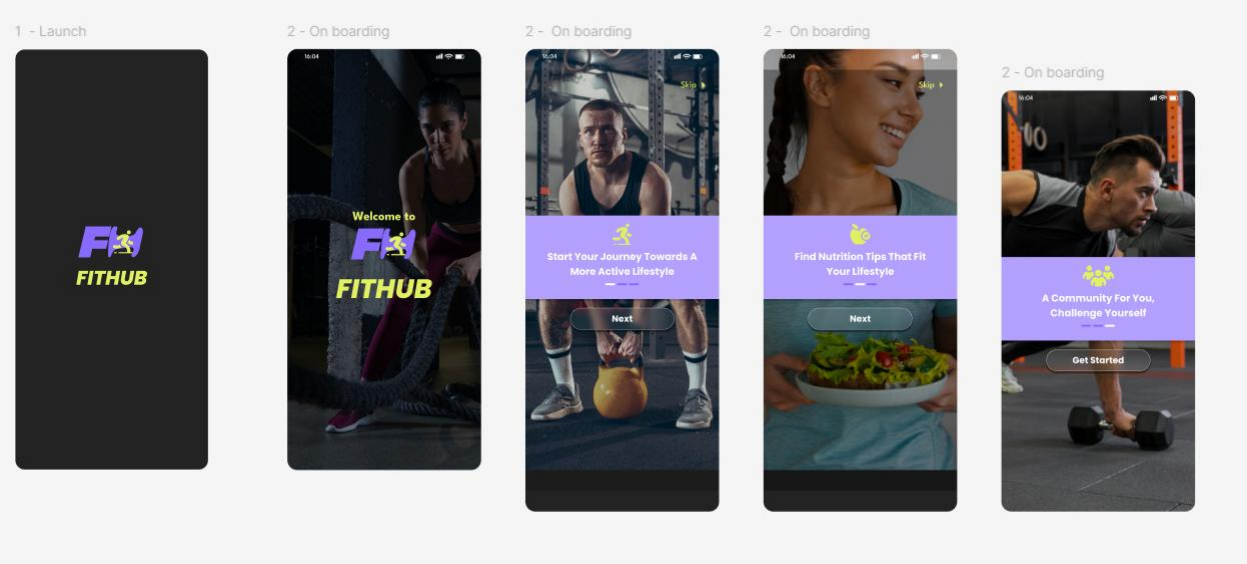


Figure 77: Landing Screen

On pressing the skip button or if you move along with all screens you will ultimately land on login screen.

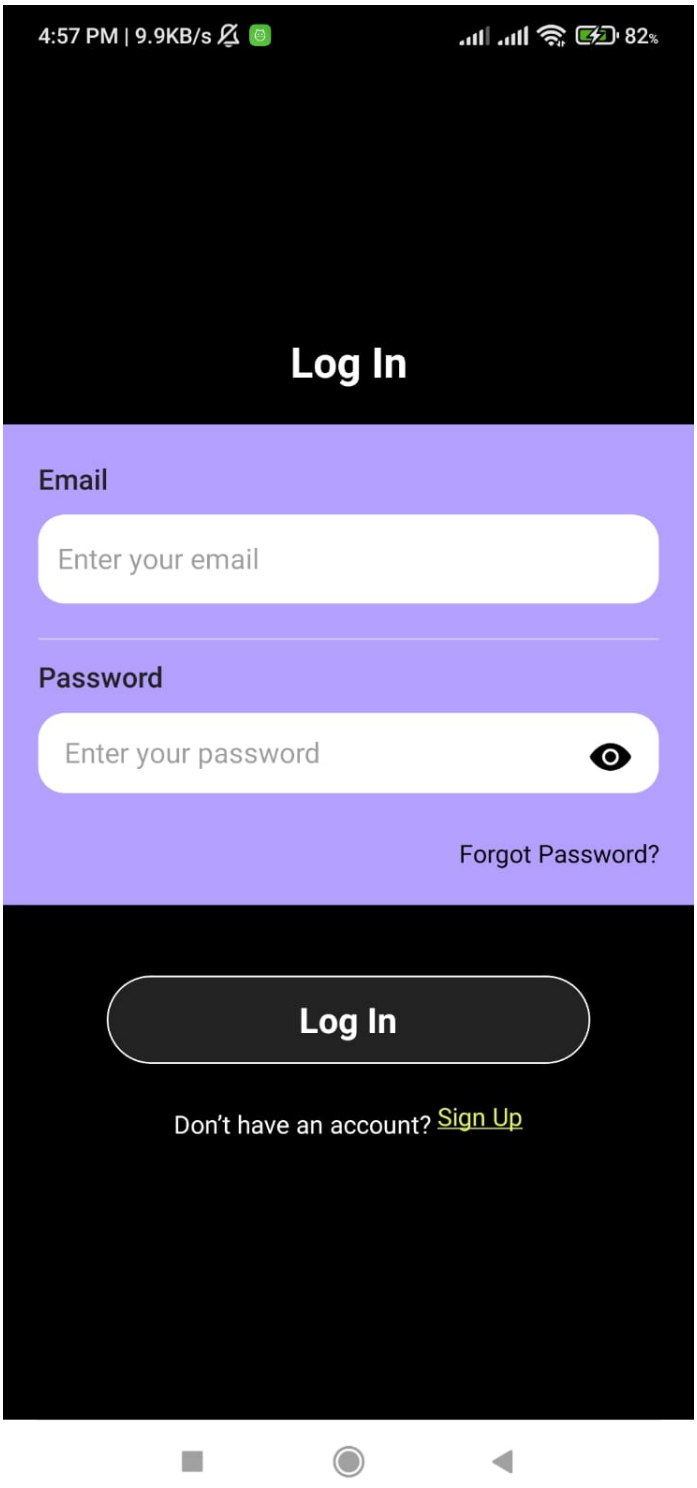


Figure 78: Login Screen

If you don’t have an account, you can click on to sign up button which will lead you to enter different credentials input screen then you can enter all the fields.

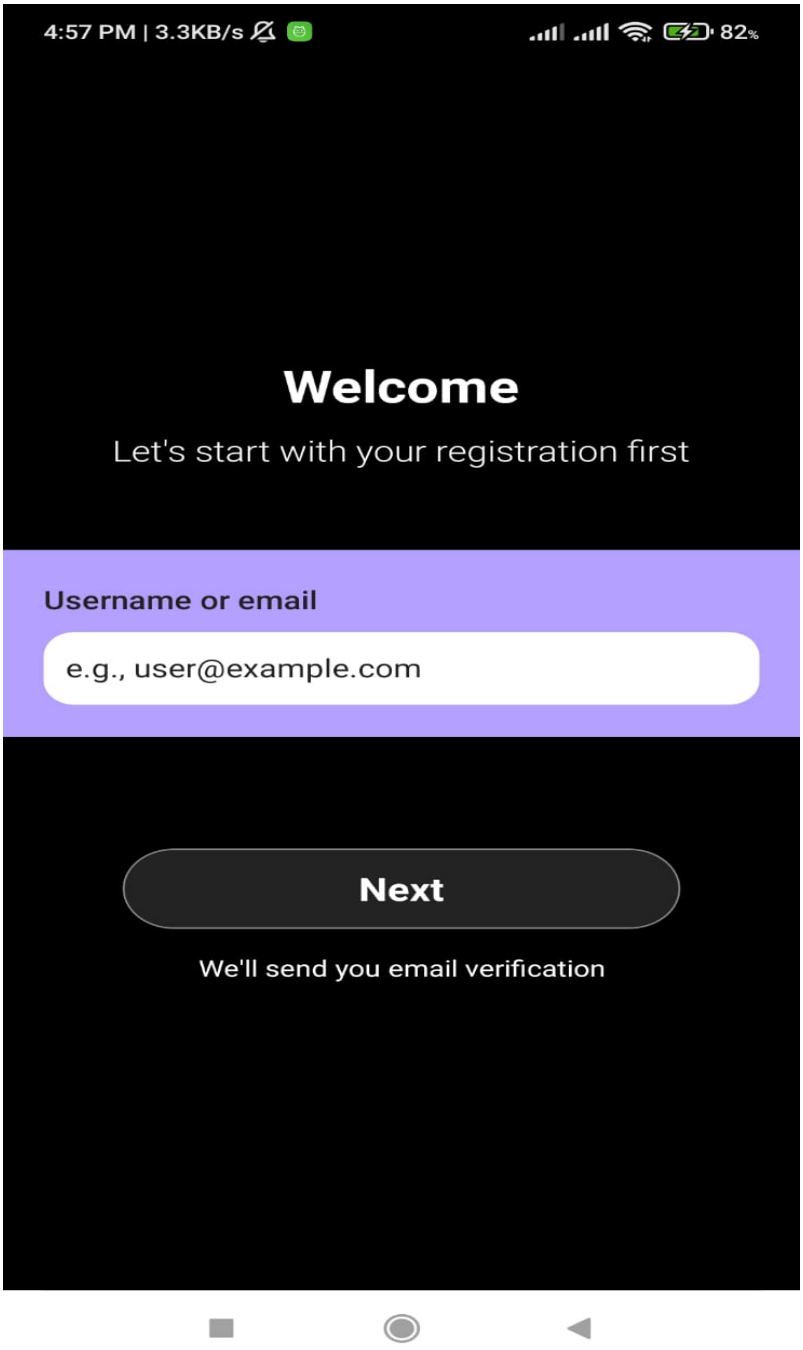


Figure 79: Sign Up screen

This is the create password screen where you can enter and confirm password and you have got some regex condition in screen with red color which will automatically turn green if all the criteria gets accepted.

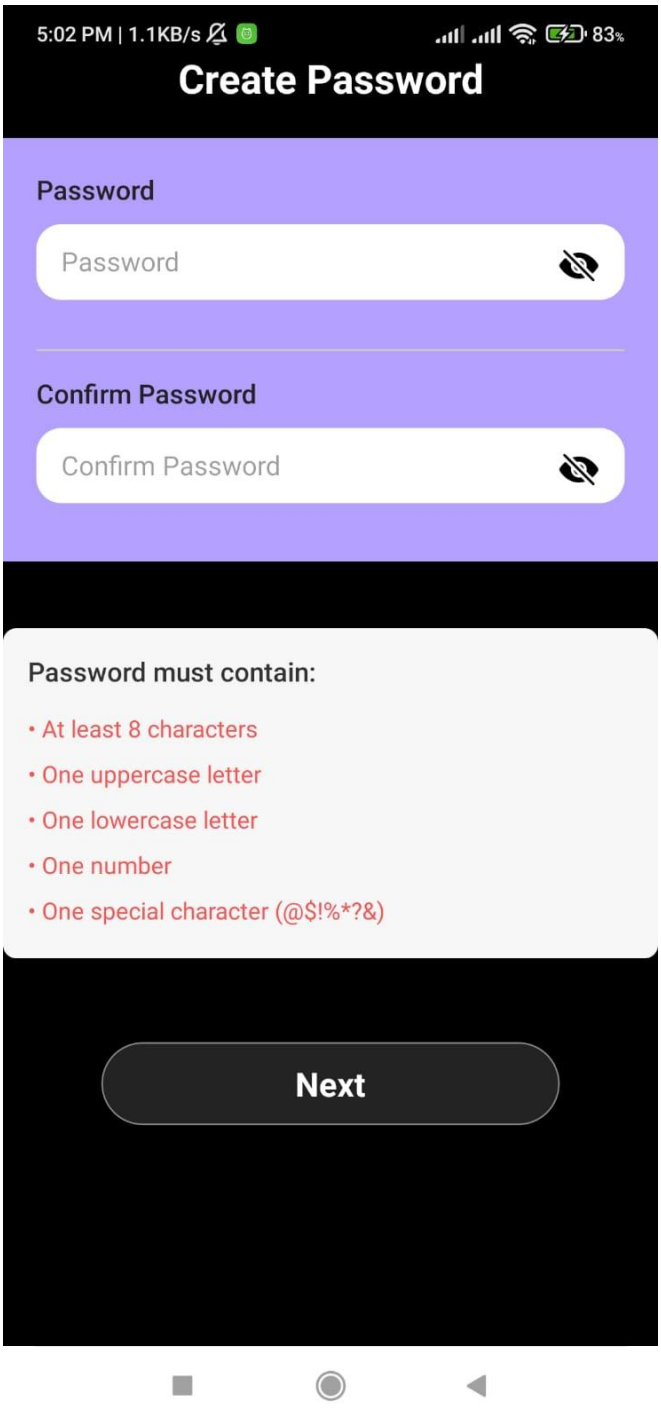


Figure 80: Before password screen

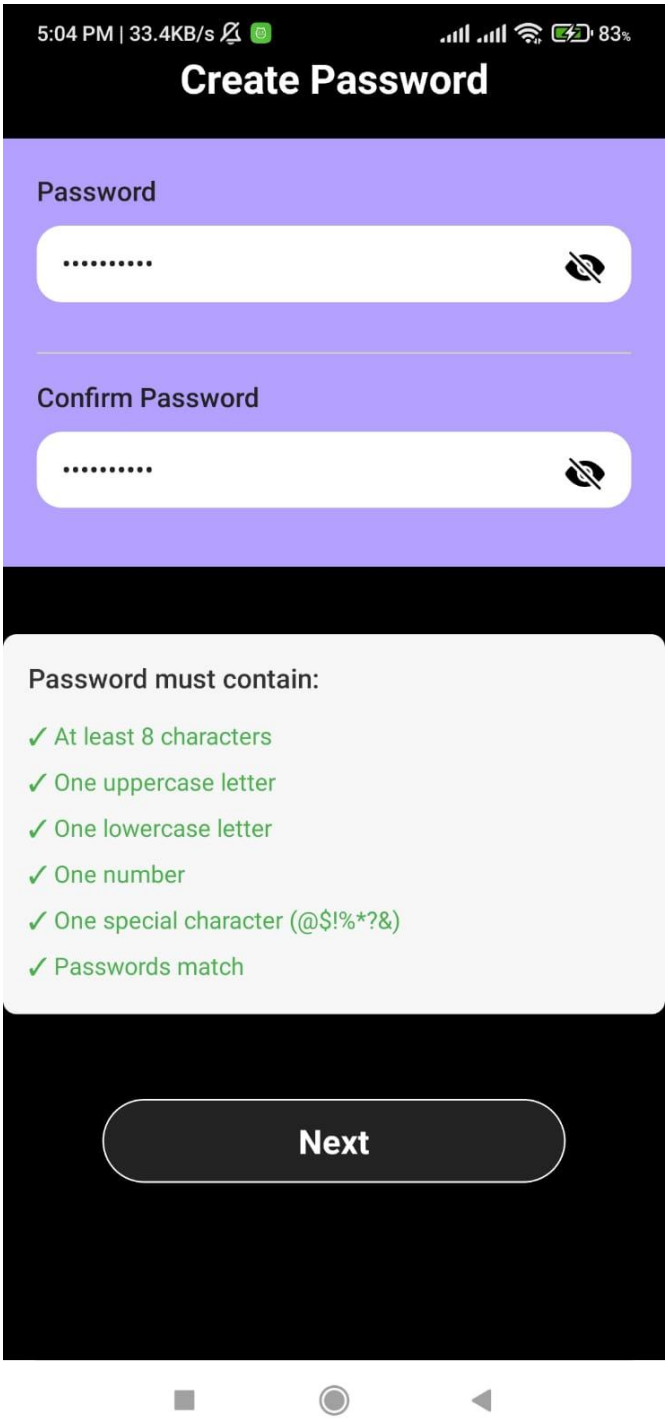


Figure 81: After Password screen

After browsing through different credentials page. you will direct into OTP screen where we check our mail for OTP and enter it into application.

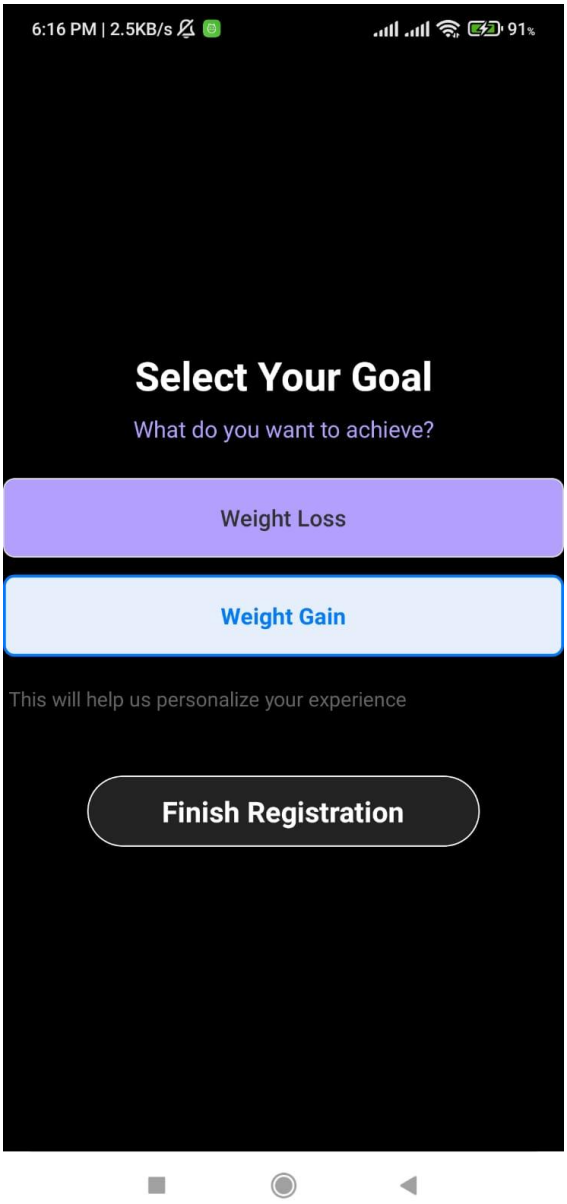


Figure 82: Goal Screen

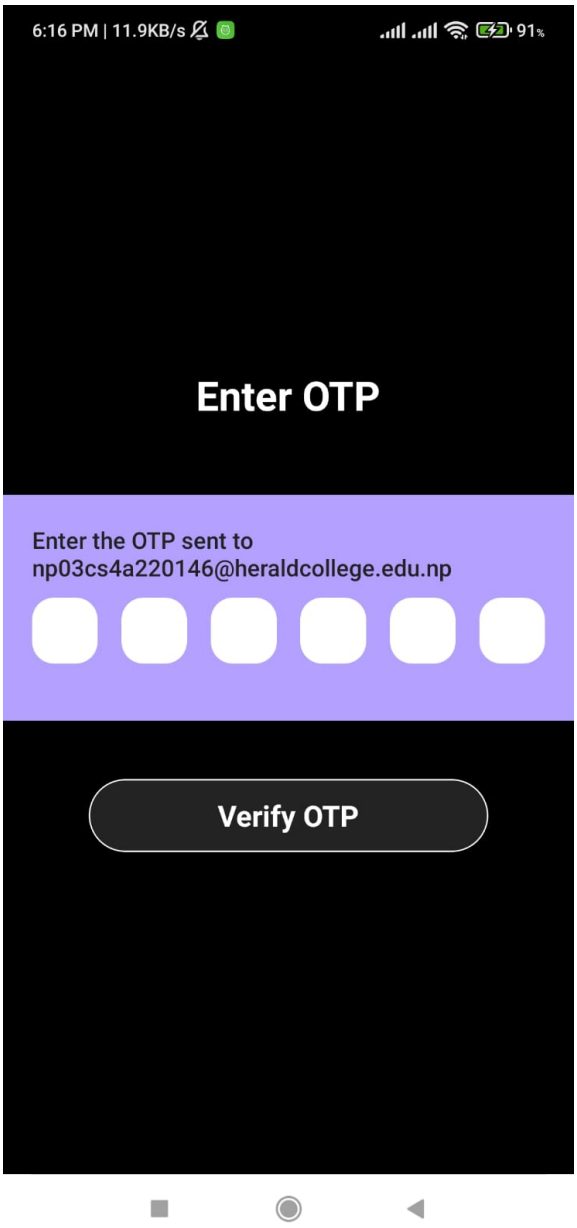


Figure 83: Inser OTP screen

You get an alert of completing registration in an alert message containing(“Success!! You have been registered successfully!”). And redirect to login page.

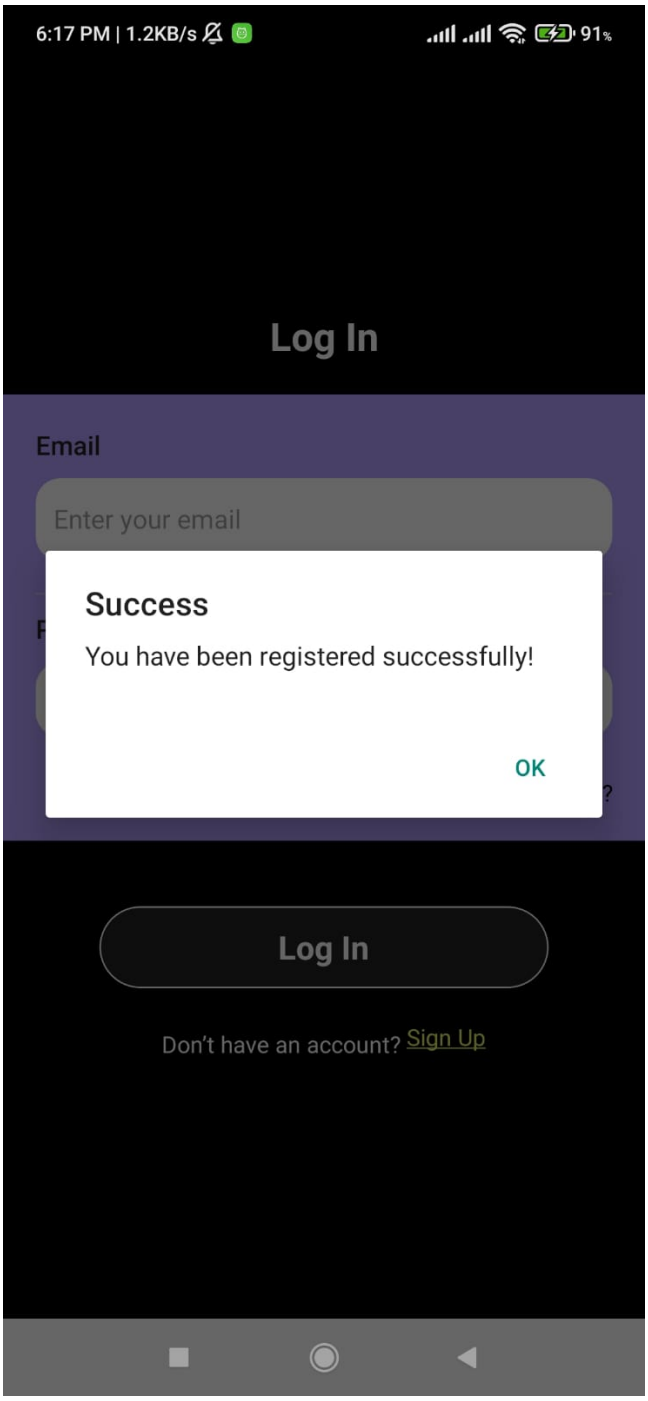


Figure 84: Finish registration

On clicking on forgot password you get field for inserting mail. You need to place your male then you will receive otp and again with otp you redirect to password and confirm password screen which will reset the password.

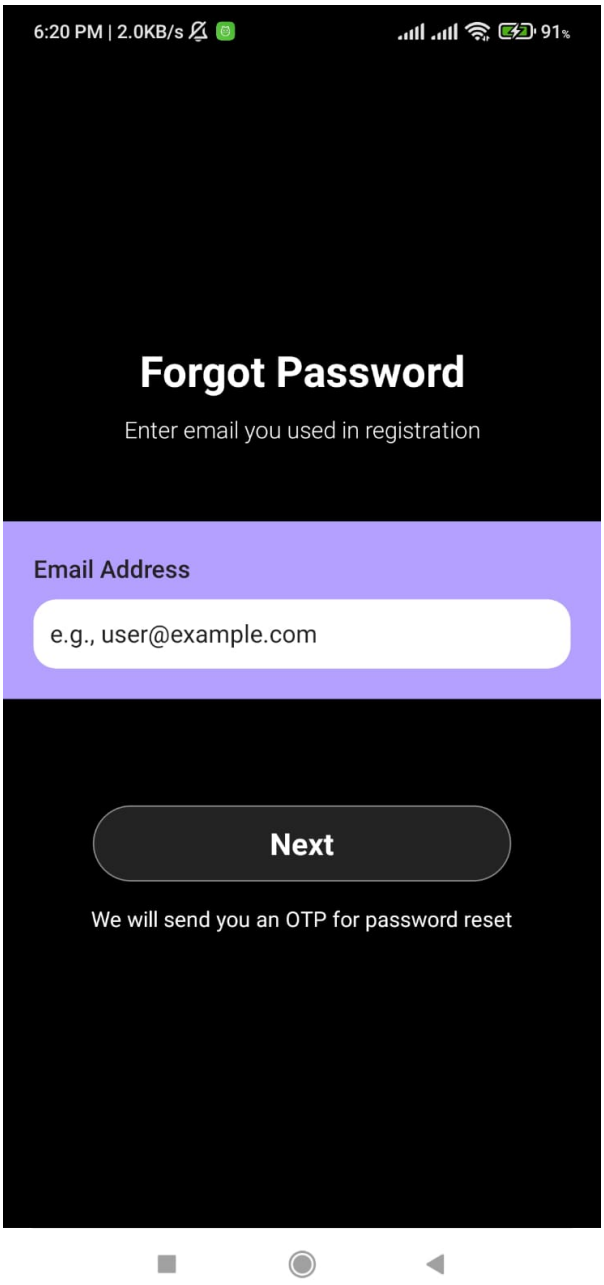


Figure 85: Forgot Password

Let's start with home screen now on. After successful login you can redirect to home screen.

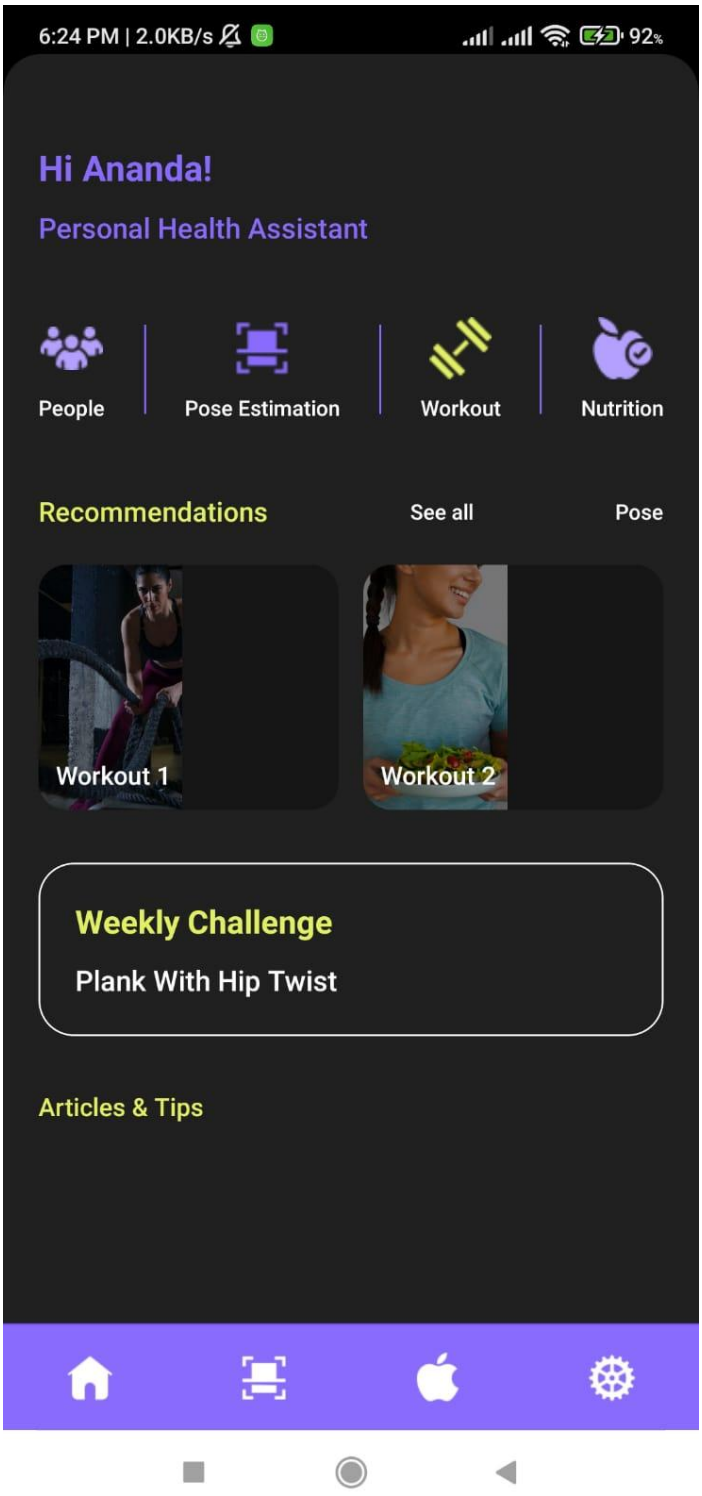


Figure 86: Home page

On clicking the highlighted green color button for workout you will navigate to select body part screen where you can select for body parts for your exercise.

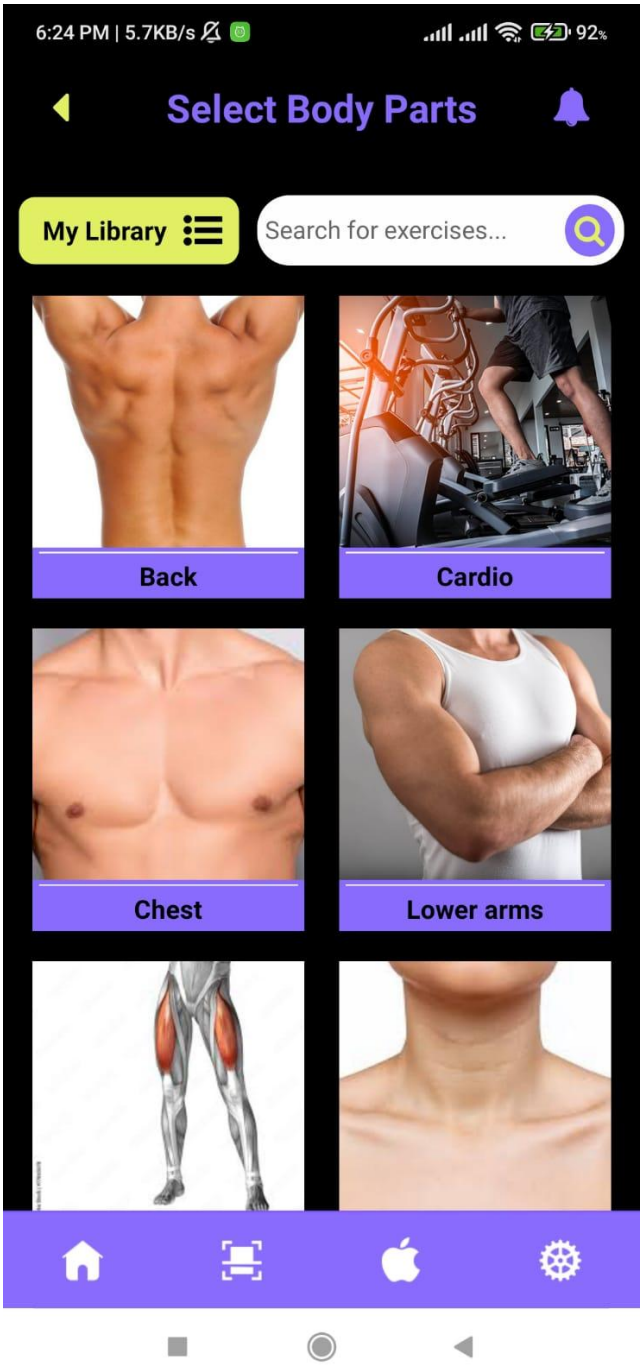


Figure 87: Select body part

You will browse to screen where you can see multiple options of exercises fetched from API.

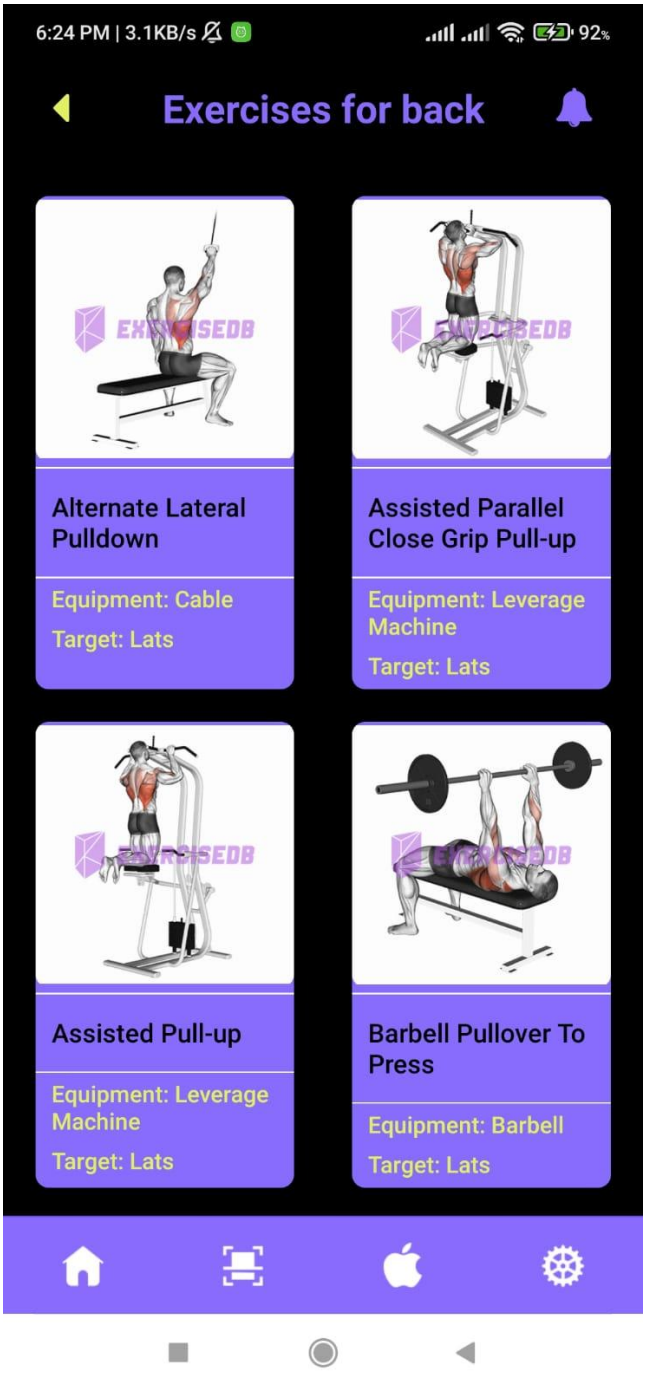


Figure 88: Body parts

On clicking any of the exercise you will see the details of the exercise with instructions and video example

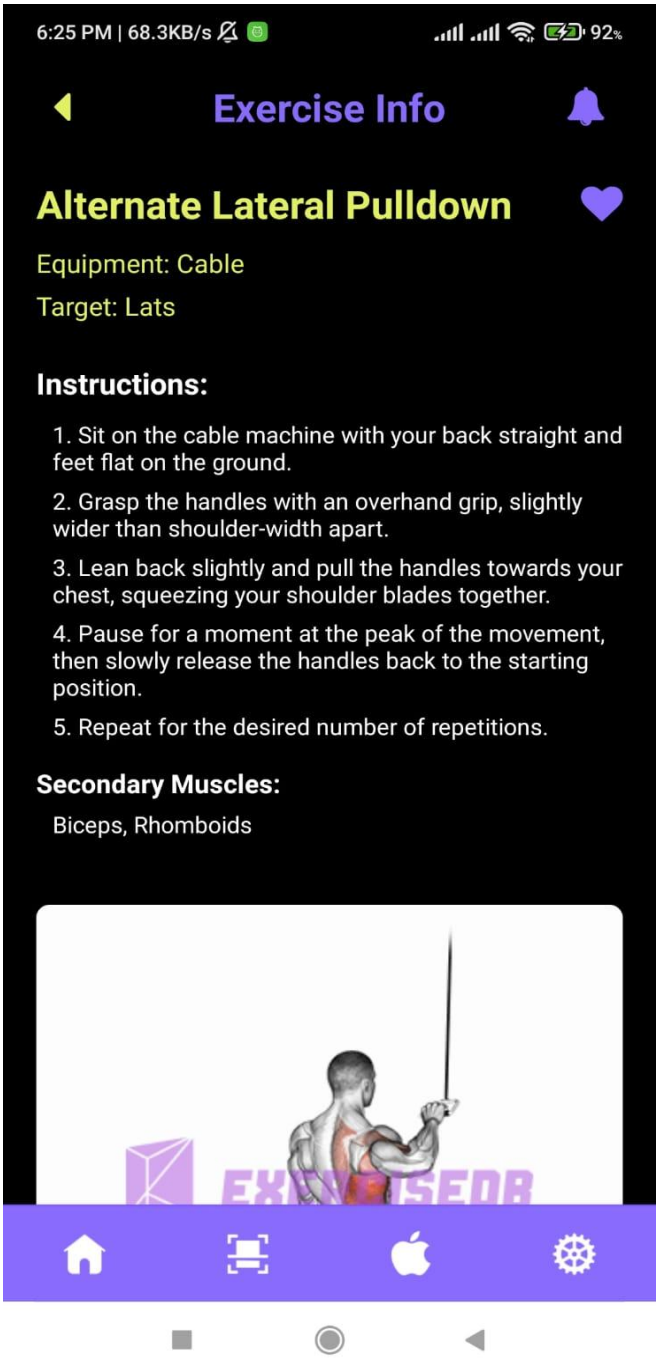


Figure 89:Exercise detail screen

Then clicking on start button you navigate to sets information page where you enter the weight and reps you can calculate weight and reps and add different set and later complete workout.

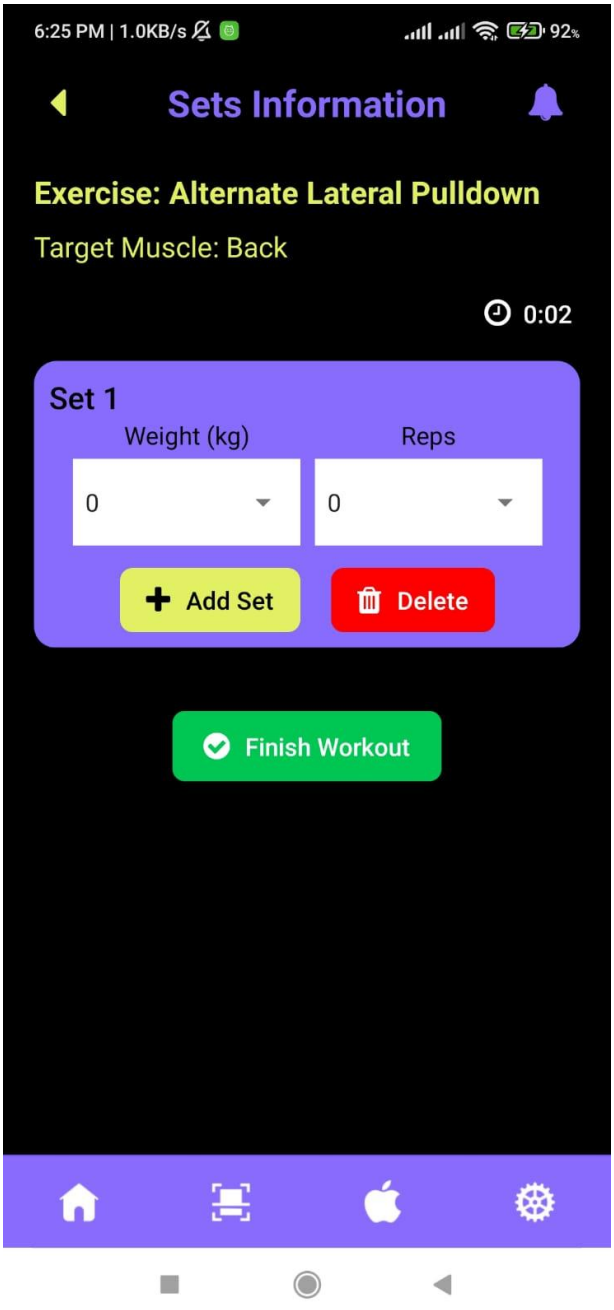


Figure 90: Sets and reps screen

When you click on scanner type footer button you will navigate to another functionality of application which trace pose of your exercise. First you can see screen for difficulty level where you can set either beginner or pro mode.

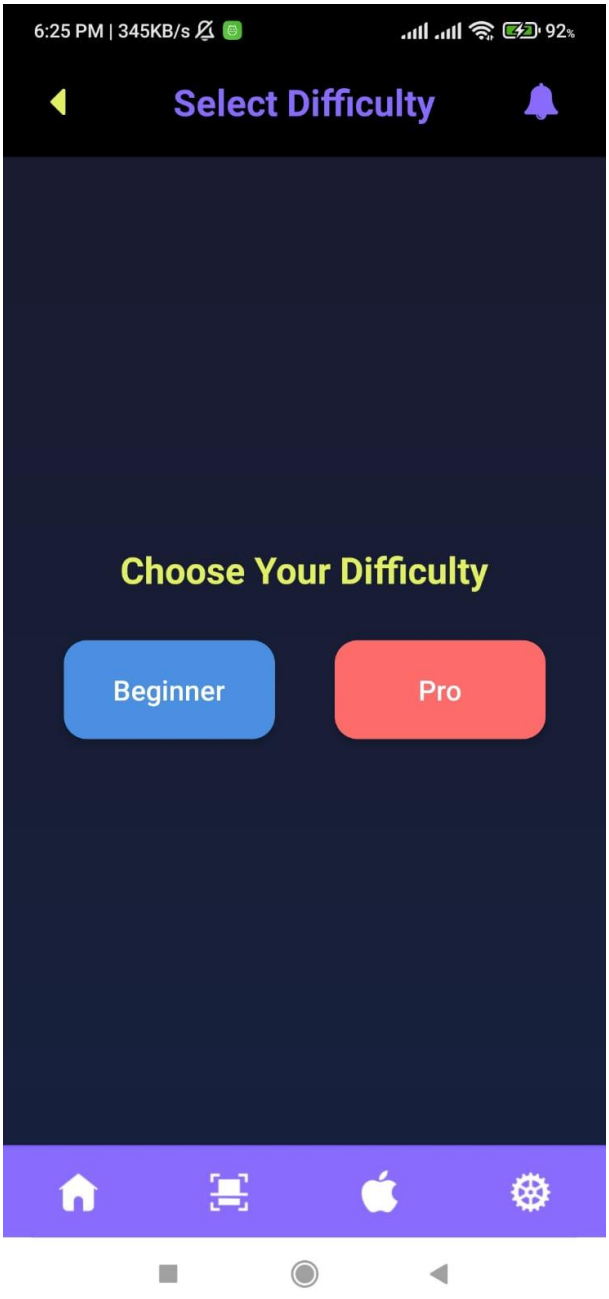


Figure 91: Beginner or Pro

Then you get choice of selecting between exercises where you got options of three exercises.

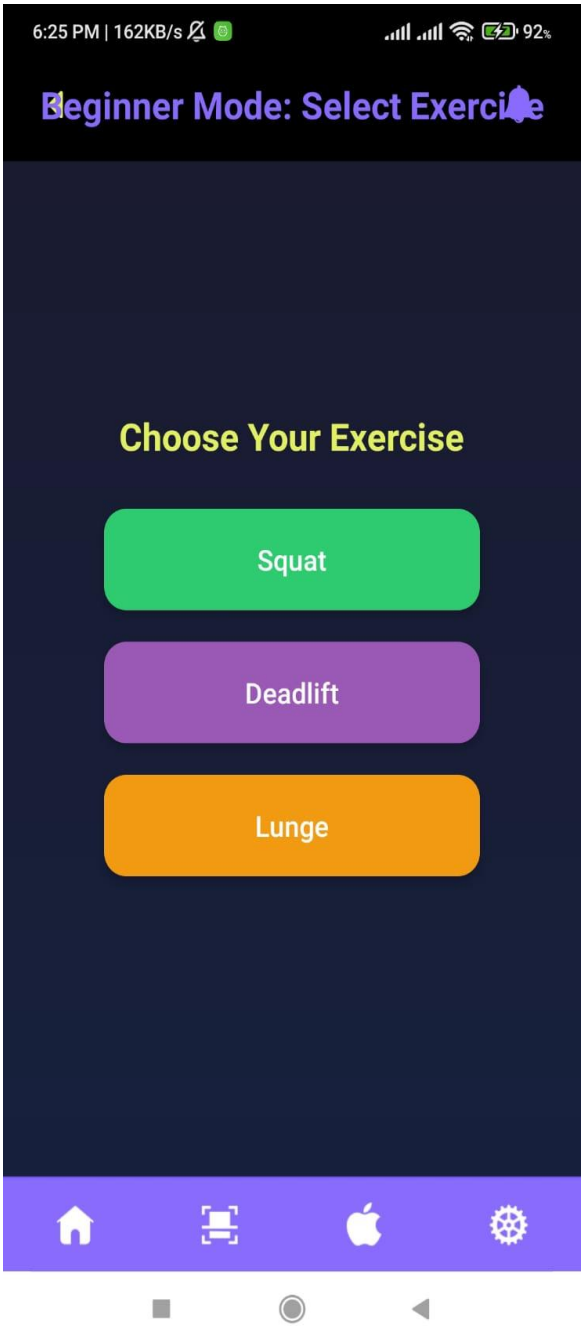


Figure 92: Choose exercise to track

Then next you will navigate to the camera access screen where you get coordinates screen and then the traced pose coordinates and then get feedback on top of screen.

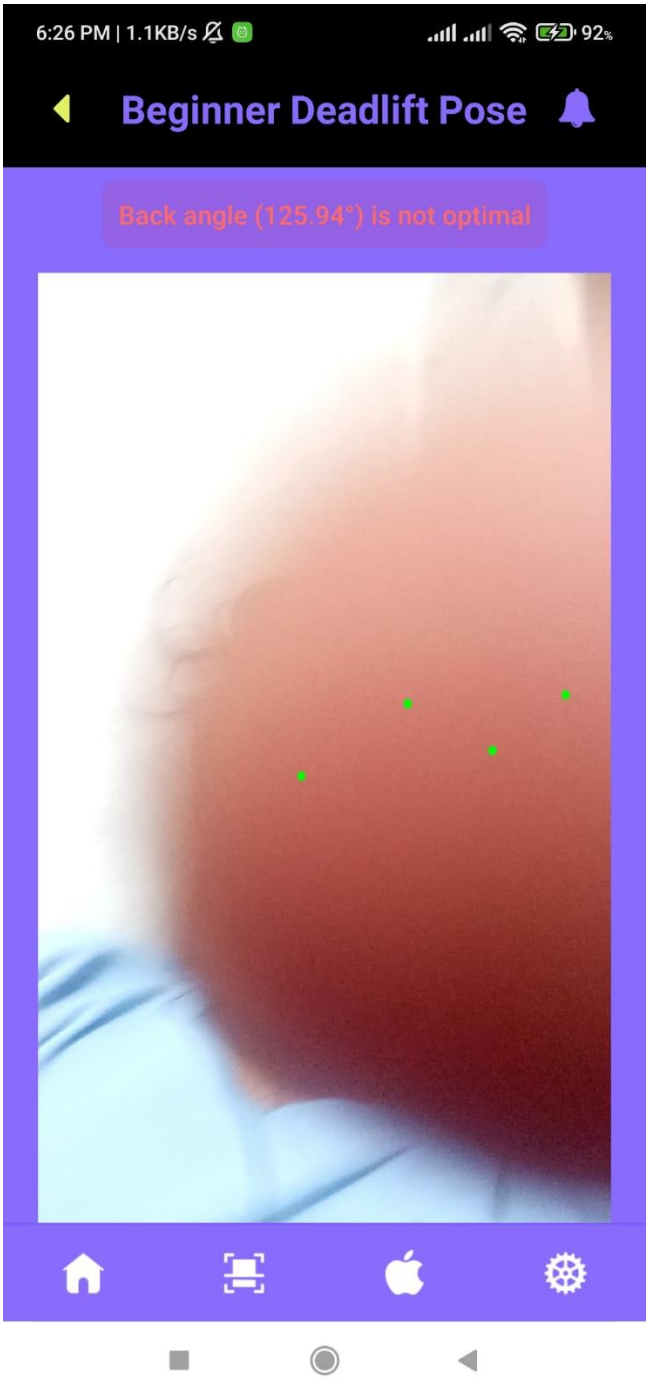


Figure 93: Trace Pose

For your personalized meal plan you can choose of your dietary preference and your activity levels.

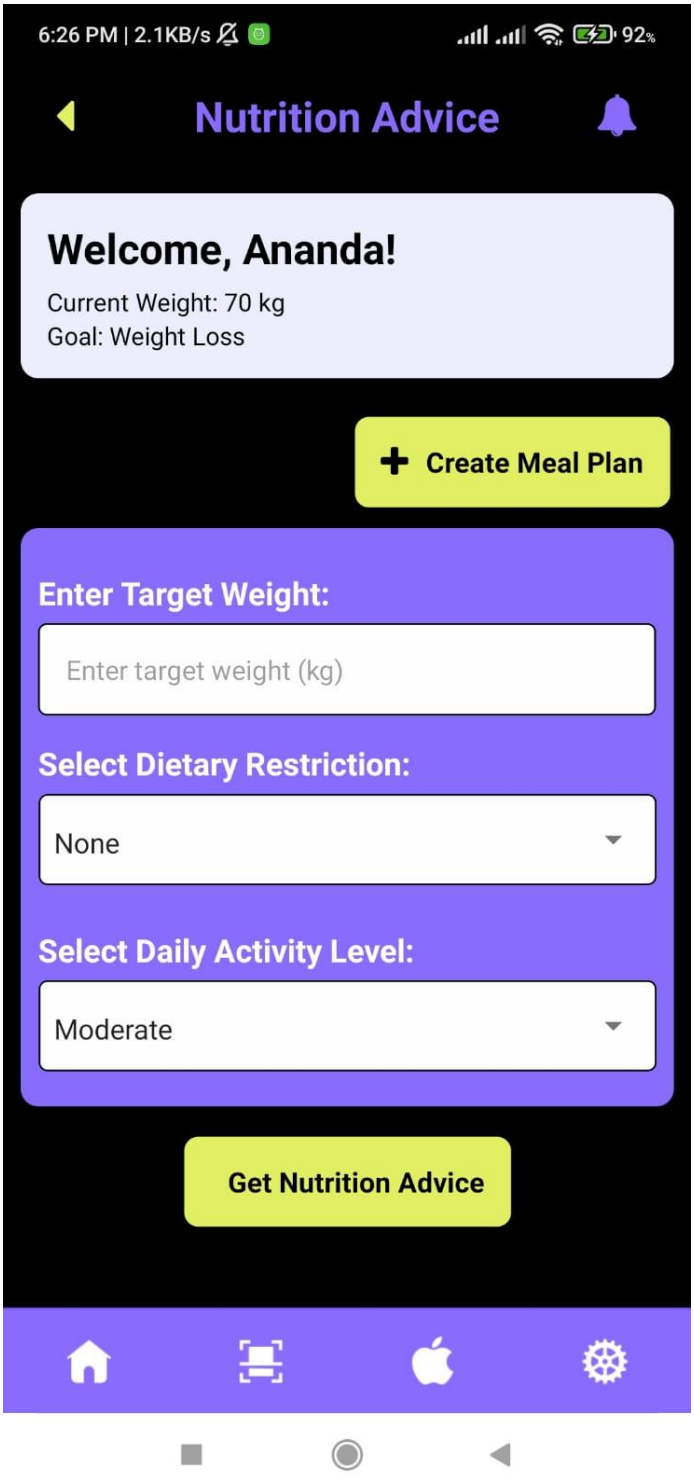


Figure 94: Nutrition plans

After undergoing all your requirements you will navigate into your meal plans. Here you can see your meal plans for different snack time.

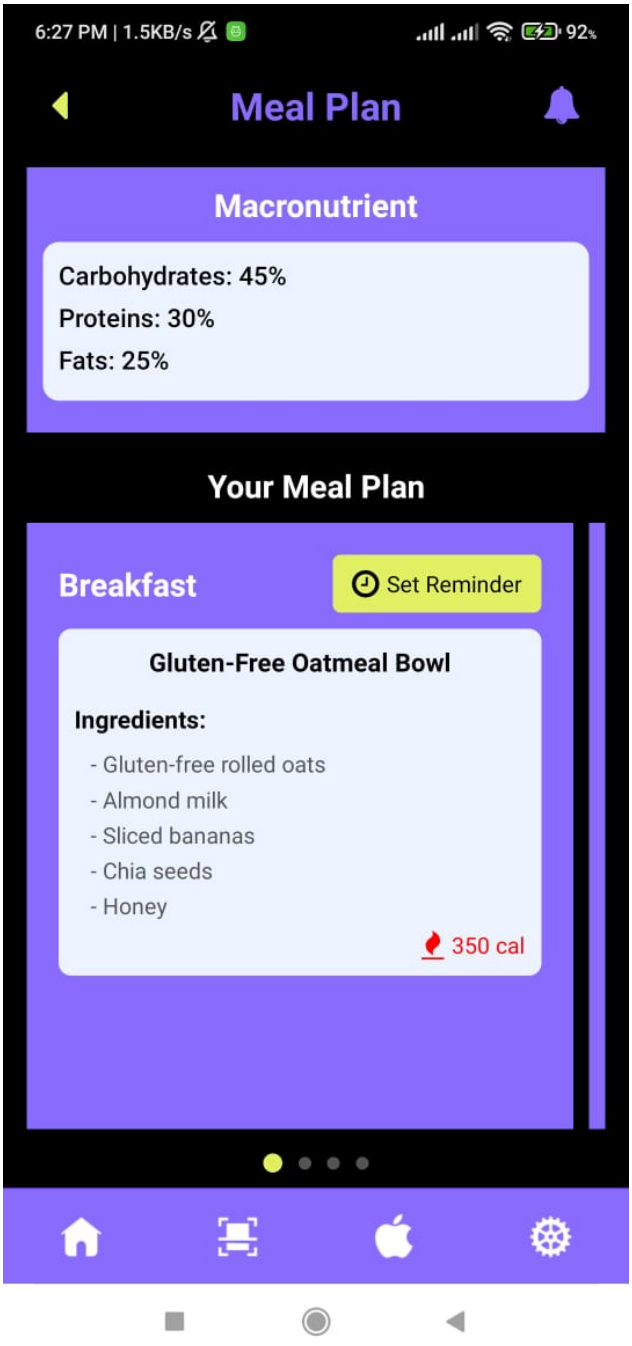


Figure 95: Nutrition meal plan

On clicking of set reminder you get the options of selecting time where you can set the time when you want the system to remind you.

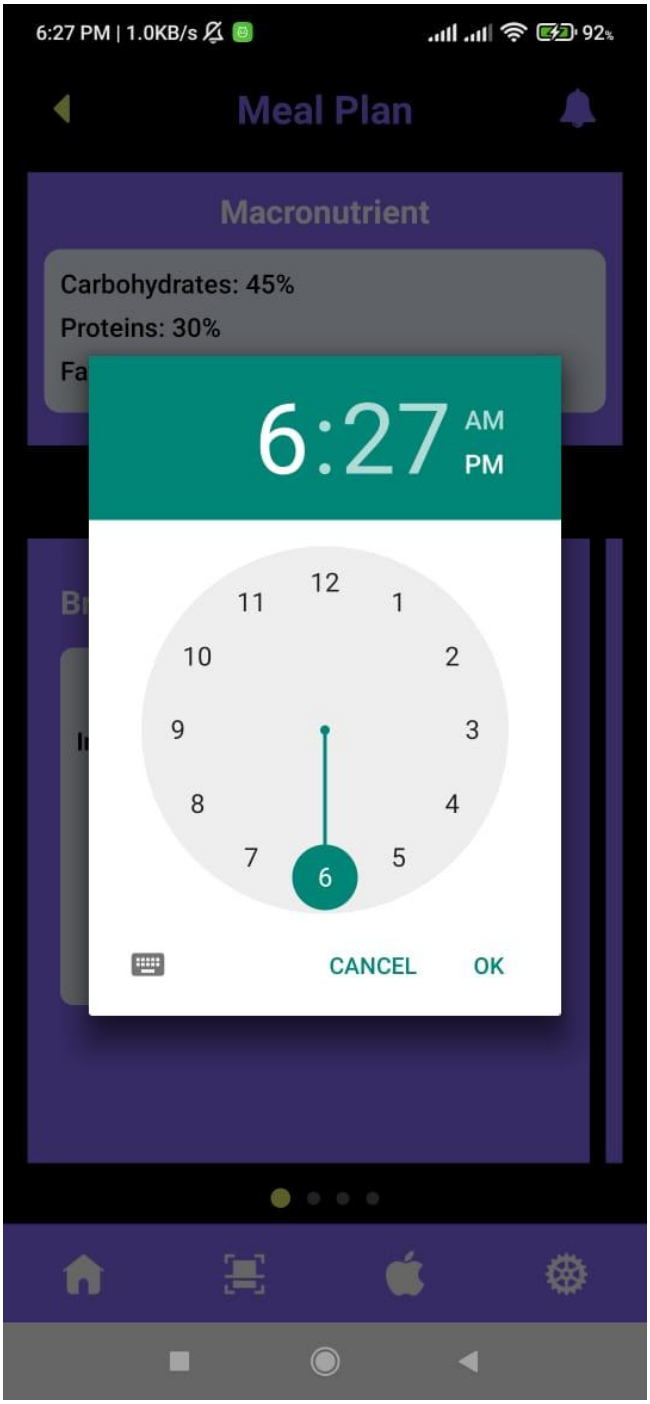


Figure 96: Set reminder screen

After selecting time you get the alert message of the time that you set reminder to activate in you phone. This will return as alarm in your device.

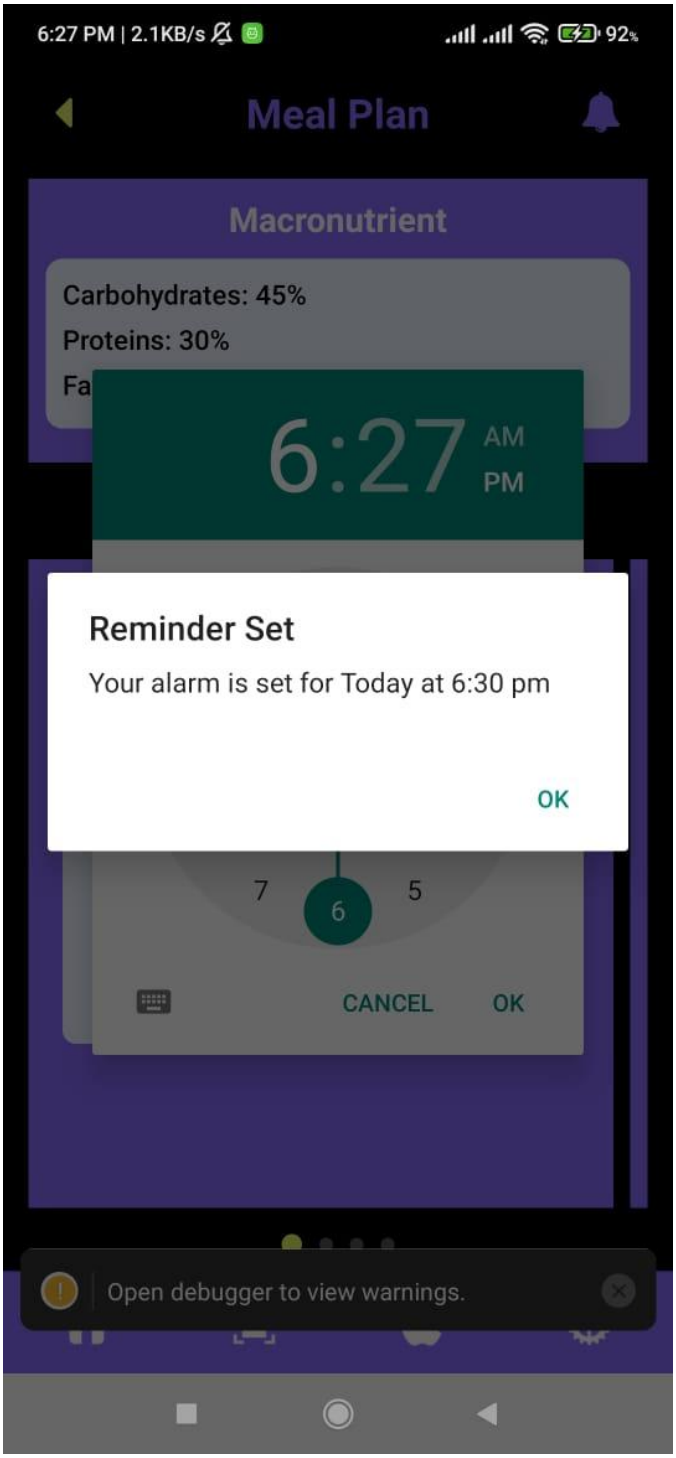


Figure 97: Alarm set alert page

On clicking the notification page you can see the notifications and reminders you set in app.

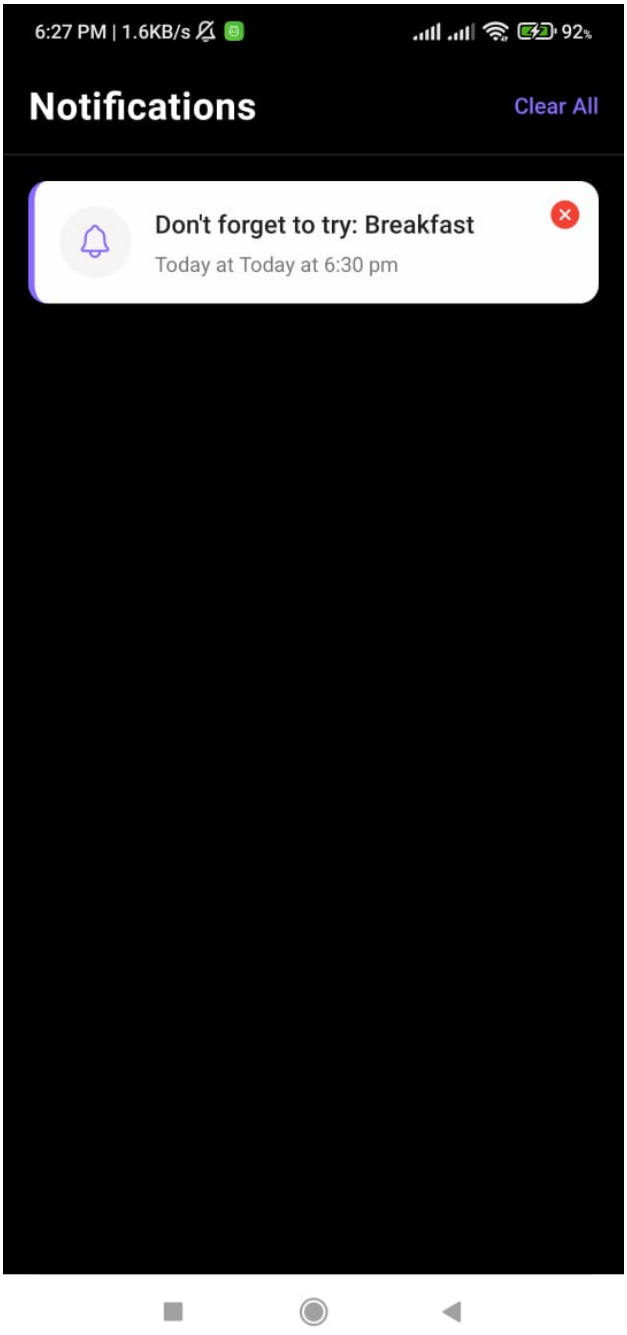


Figure 98: Notification page

This is the update profile page where you can update your credentials as per your wish and need.

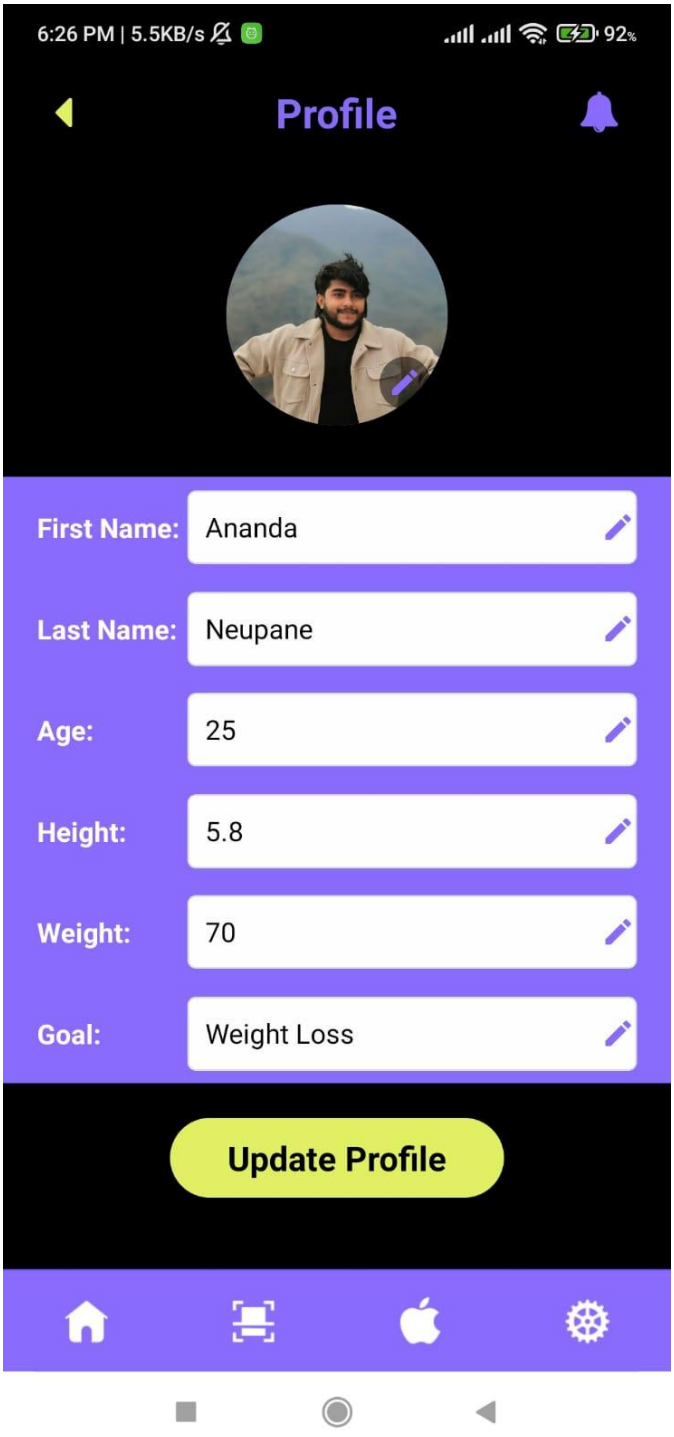


Figure 99: update Profile

These are some necessary user manuals that you will guide you with working of the application.

11. References

Abeltino, A., Riente, A., Bianchetti, G. & Serantoni, C., 2024. Digital applications for Diet Monitoring, planning, and precision nutrition for citizens and professionals: A state of the art. *Nutrition Reviews.*

Anon., 2024. *Strong workout tracker and gym log.* [Online] Available at: https://www.strong.app/#features [Accessed 2025].

FitnessApp, M., 2024. *My FitnessApp blog.* [Online] Available at: https://blog.myfitnesspal.com/ [Accessed 2025].

Hitesh, K., Daiber, F., Kravcik, M. & Duong-Trung, N., 2024. FitSight: Tracking and feedback engine for personalized fitness training. *Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization,* pp. 223-231.

jefit, 2024. *Blog | Jefit #1.* [Online] Available at: https://www.jefit.com/blog/ [Accessed 2025].

Kendall, A., Grimes, M. & Cipolla, R., 2015. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. *2015 IEEE International Conference on Computer Vision (ICCV),* pp. 2938-2946.

Ketan, T., Dhavala, S., Vijayarao, S. & Challagundla, Y., 2024. Real-time human pose estimation using media-pipe an artificial intelligence applications in health and Fitness. *2024 4th International Conference on Artificial Intelligence and Signal Processing (AISP),* pp. 1-6.

Mahendran, N., 2025. *Deep learning for fitness.* [Online] Available at: https://arxiv.org/abs/2109.01376

nikw, 2024. *Nike training app. home.* [Online] Available at: https://www.nike.com/ntc-app [Accessed 2025].

On, F., 2024. *FitON Blogs.* [Online] Available at: https://www.fitonhealth.com/blog [Accessed 2025].

Salik, K., Paulino, D. & Sampaio, J., 2022. A review on Computer Vision Technology for Physical Exercise Monitoring. *Algorithms,* 15(12), p. 444.

Sebastian, D., Andreas, R., Rohr, M. & Güney, G., 2023. Accuracy evaluation of 3D pose estimation with PoseNet pose for physical exercises. *Current Directions in Biomedical Engineering,* pp. 563-566.

Stavros, A., Asimakopoulos, G. & Spillers, F., 2017. Motivation and user engagement in fitness tracking: Heuristics for Mobile Healthcare Wearables. *Informatics.*

Stockwell, S., Trott, M., Tully, M. & Shin, J., 2021. Changes in physical activity and sedentary behaviours from before to during the COVID-19 pandemic lockdown: A systematic review. *BMJ Open Sport & Exercise Medicine.*

Urmi, D., Bhoir, P. & Ranka, P., 2023. Pose estimation and virtual gym assistant using PoseNet and machine learning. *2023 International Conference on Network, Multimedia and Information Technology (NMITCON),* pp. 1-7.

Youssef, F., Mahmoud, A. & El Gohary, S., 2021. AI & multi-resolution temporal processing for accurate counting of exercises repetitions. *2021 International Telecommunications Conference (ITC-Egypt).*

Yuliang, G., Li, Z. & Li, Z., 2022. Pop-net: POSE over parts network for multi-person 3D pose estimation from a depth image. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV),* pp. 3917-3926.

Zhao, Z., Arya, A., Orji , R. & Chan, G., 2020. Effects of a personalized fitness recommender system using gamification and continuous player modeling: System design and long-term validation study. *JMIR Serious Games.*

12. Appendix

1. Work Breakdown Structure

Figure 100: WBS Diagram

2. Gantt Chart

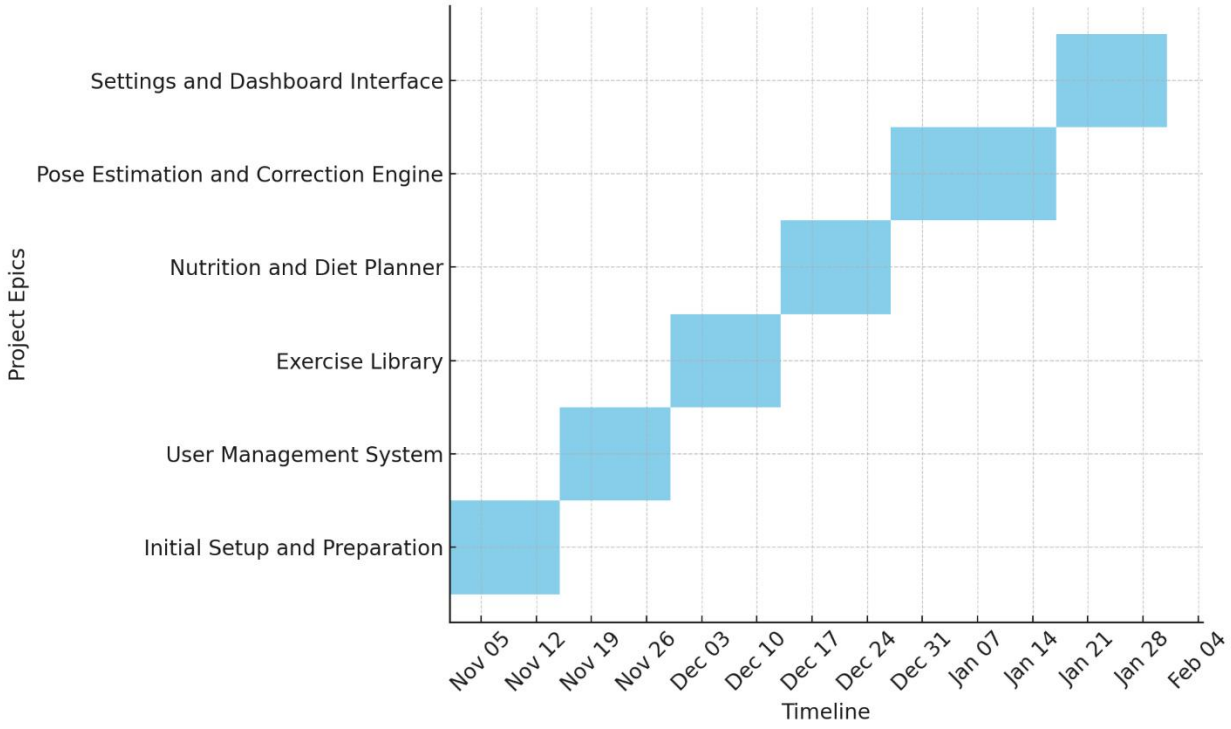


Figure 101: Gantt

3. Architecture Diagram

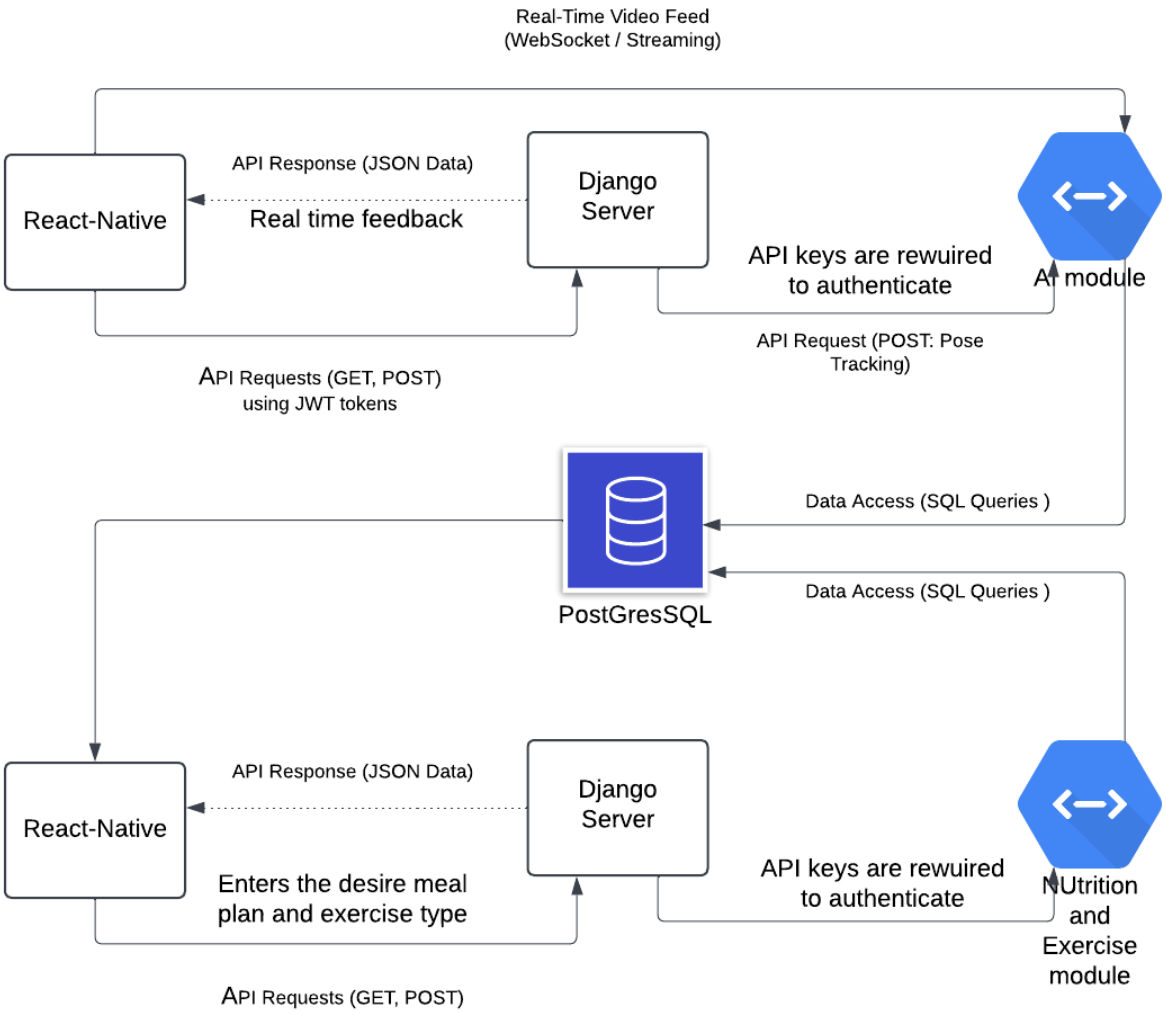


Figure 102: Architecture diagram

4. User flow diagram

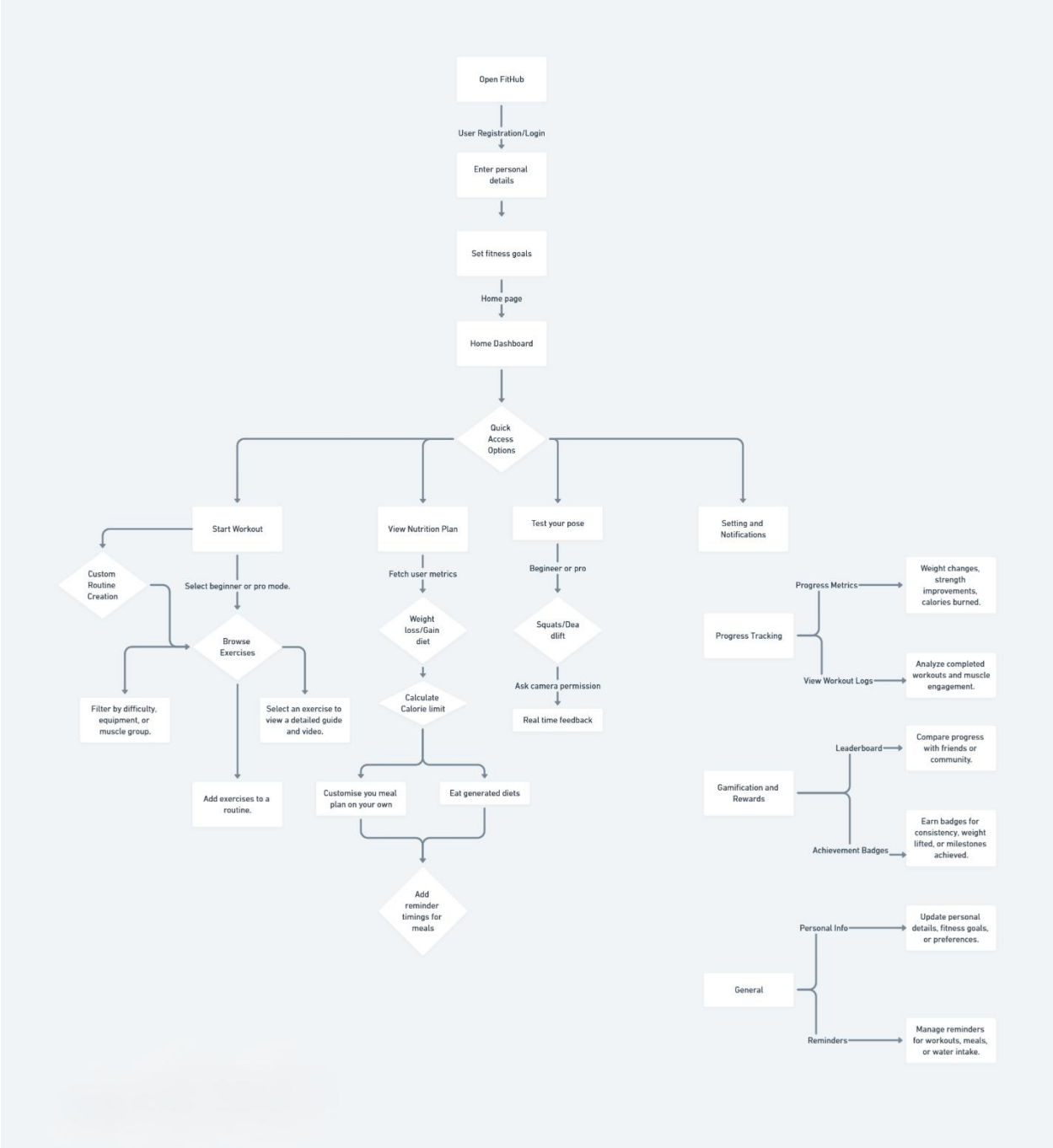


Figure 103: User Flow Diagram

5. Use case diagram

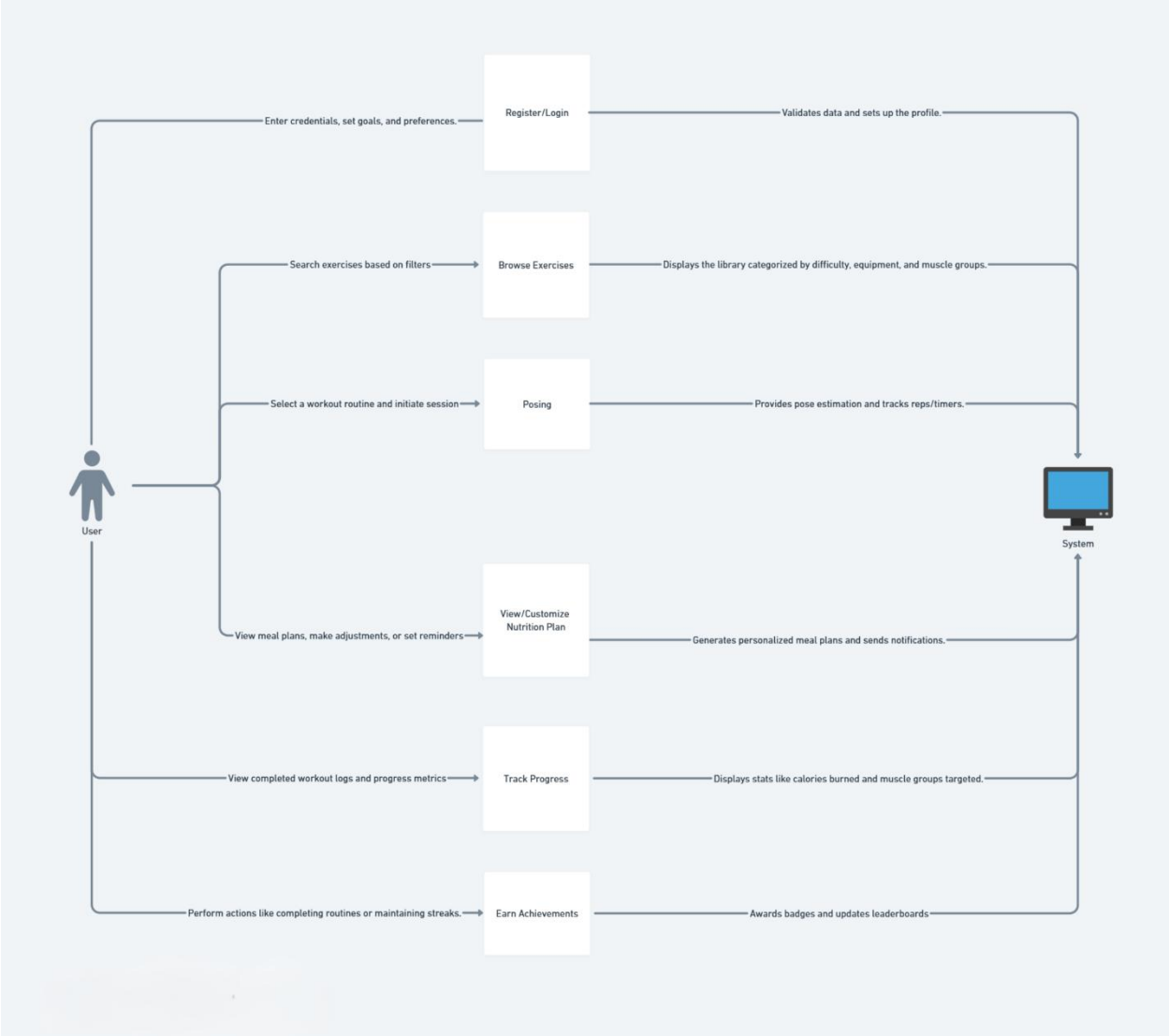


Figure 104: Use case diagram