Ministry of Communication and Information Technology

Digital Egypt Pioneers Initiative (DEPI)

YAT Learning Center

Fitness Tracker Dashboard with React

Group 1

DONE BY:

Mai Mohamed Ali

Lamiaa Ezzat Ahmed

Mazen Mohamed Abd El Maksoud

Abdelrhman Shoukry Abd El Ghaffar

Tarek Gamal Abd El Qawy

Instructor: Ashraf Sadek

Content List

Overview

Objective

Build a fitness tracking dashboard to display real-time statistics and goals.

Description

Develop a fitness dashboard where users can log daily workout activities, set fitness goals, and track progress over time. The application will use React for the front end, integrated with a Node.js backend to manage user data. Incorporate charts and graphs (e.g., using Chart.js or D3.js) to visualize users' progress, and implement user authentication to allow individuals to manage their own fitness records.

Technologies to use

React, NodeJS, HTML, CSS, JavaScript, Chart.js, API Integration, and Frontend Development.

Project Proposal

Overview:

Built with React and Node.js, the fitness tracking dashboard will provide an intuitive interface for users to track their workout history and progress. With API integration and real-time data visualization, the platform will offer user-friendly experience, making fitness tracking efficient and engaging.

Objective:

Develop a fitness tracking dashboard that enables users to log daily workouts, set fitness goals, and monitor progress through interactive charts. The platform will provide seamless experience with real-time statistics, helping users stay motivated and achieve their fitness targets.

Scope:

The application will feature user authentication, allowing individuals to manage personal fitness records securely. It will integrate a Node.js backend for data management and utilize Chart.js for visualizing progress. The dashboard will support activity logging, goal setting, and insightful analytics for better fitness tracking.

Project Plan

Week 1: Setup and Wireframes

Task	Timeframe	Progress
Set up React, NodeJS, and MongoDB	Day 1-2	Completed
Design wireframes for UI	Day 3-4	Completed
Build basic frontend layout	Day 5-7	Completed

Week 2: Core Features & Backend

Task	Timeframe	Progress
Create fitness data logging forms	Day 1-2	Completed
Develop backend API with MongoDB	Day 3-4	Completed
Integrate Redux for state management	Day 5-7	Completed

Week 3: Visualizations & Testing

Task	Timeframe	Progress
Implement Chart.js/D3.js graphs	Day 1-3	Completed
Conduct unit testing	Day 4-5	Completed
Ensure full responsiveness	Day 6-7	Completed

Week 4: Final Enhancements & Deployment

Task	Timeframe	Progress
Implement user authentication	Day 1-2	Completed
Final testing & bug fixes	Day 3-5	Completed
Deploy app & write documentation	Day 6-7	Completed

Task Assignment & Roles

Mazen →page 1 & 12 {Home & Help}

Tarek→page 8 {Dashboard}

Abdelrhman→ page 9 & 10{Training & activities & Sidebar}

Lamiaa→page 2, 3, 4 & 5{Forms & Daily Report}

Mai→page 6, 7, 8 &11{Forms & Profile}

Identifying Risks and Solutions

Risk	Solution
Delays in backend API development	Prioritize API setup early and use Postman for testing.
State management complexity	Use Redux for global state management and maintain clear documentation.
Performance issues with data visualization	Optimize Chart.js/D3.js rendering and use lazy loading where needed.
Security concerns in authentication	Implement JWT authentication and secure API endpoints.
Cross-browser compatibility issues	Conduct thorough testing on multiple browsers and devices.
Deployment challenges	Use cloud-based deployment (e.g., Vencel, Heroku) with CI/CD integration.

Feedback & Evaluation – Lecturer's Assessment

The project demonstrates a solid understanding of full-stack development, incorporating key technologies such as React, Node.js, and MongoDB. The structured timeline ensures a well-paced development process, and the inclusion of Redux for state management enhances efficiency. The use of Chart.js or D3.js for data visualization adds significant value by making progress tracking more interactive.

Key Strengths:

✅ Well-defined project scope and timeline.

✅ Strong use of modern frontend and backend technologies.

✅ Thoughtful integration of authentication and security measures.

✅ Clear risk assessment with practical solutions.

Areas for Improvement:

🔷 Consider improving UX/UI design for better user engagement.

🔷 Ensure accessibility best practices for a broader audience reach.

🔷 Implement automated testing to streamline quality assurance.

Overall, this project is well-planned and has the potential to be an effective fitness tracking solution with strong real-world application. Keep refining and testing for optimal performance!

Requirements Gathering

Functional Requirements

List of features and functionalities:

- User registration and authentication (Login/Signup)
- Ability to log fitness activities (Workouts, Steps, Calories, etc.)
- Dashboard displaying progress statistics with charts
- Goal-setting functionality for users
- Integration with external APIs (e.g., fitness tracking apps, health APIs)
- Profile management and activity history

Non-functional Requirements

Performance, security, usability, and reliability criteria:

- **Performance:** Ensure fast load times and real-time data updates.
- **Security:** Implement JWT authentication and encrypted storage for user data.
- **Usability:** Responsive and intuitive UI for all device types.
- **Reliability:** Ensure uptime and backup mechanisms to prevent data loss.

Use Case Diagram

Functional & Non-Functional Requirements

Functional Requirements

- User authentication (Login/Signup).
- Log fitness activities (Workouts, Steps, Calories, etc.).
- Dashboard with progress charts.
- Goal-setting functionality.
- Profile management & activity history.

Non-functional Requirements

Performance, Security, Usability, Reliability

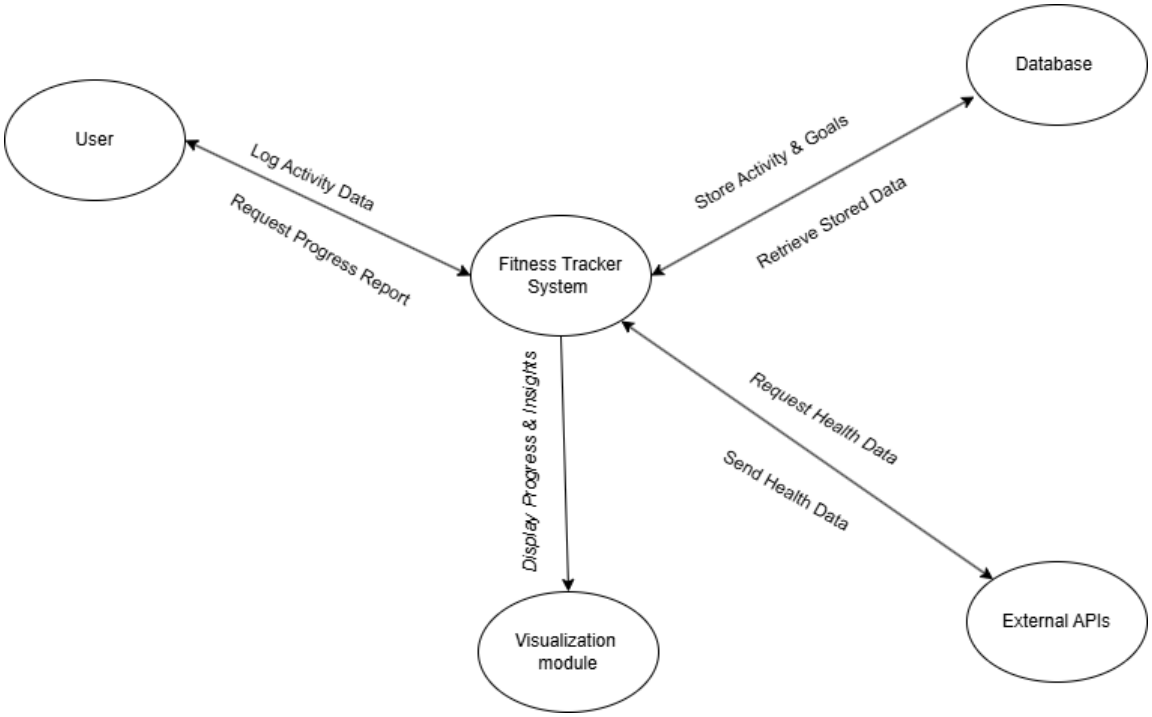
Software Architecture

The application follows **Component-Based Architecture** using React. The system is structured as follows:

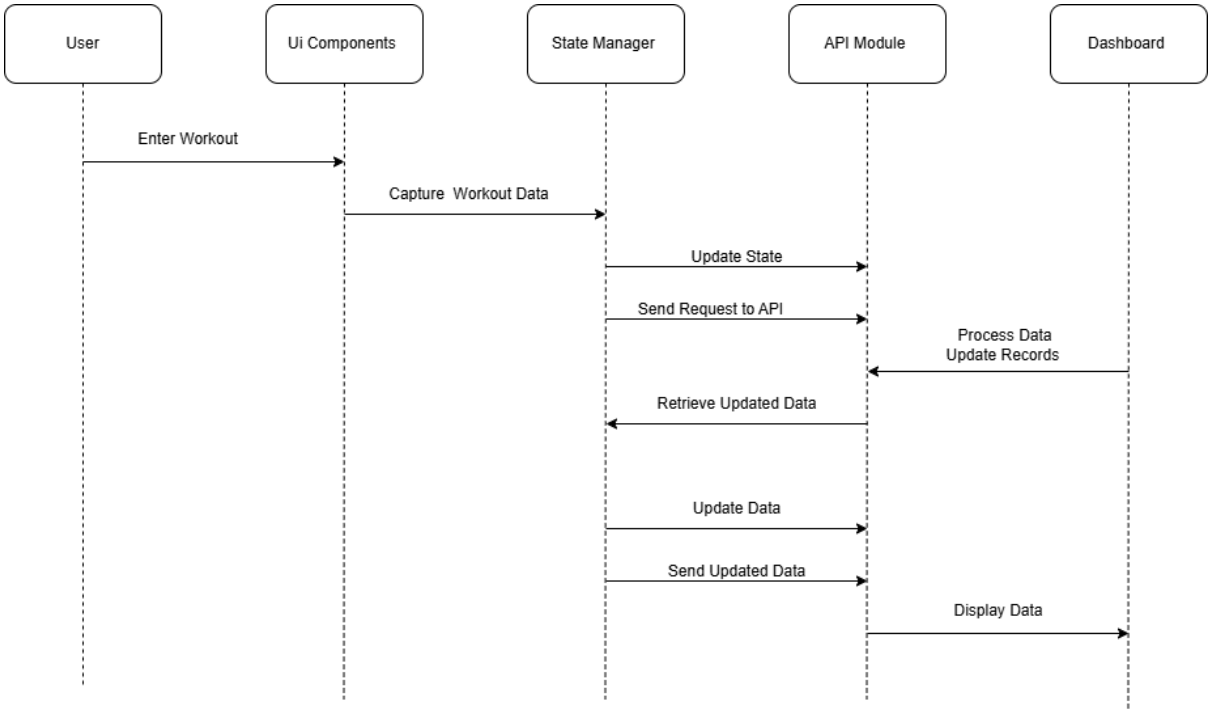
- **Presentation Layer (React Frontend):** Handles UI components, state management (Redux), and data visualization (Chart.js).
- **State Management:** Redux is used to manage global application state efficiently.
- **Responsive Design:** Ensuring compatibility across multiple devices.

Data Flow & System Behavior

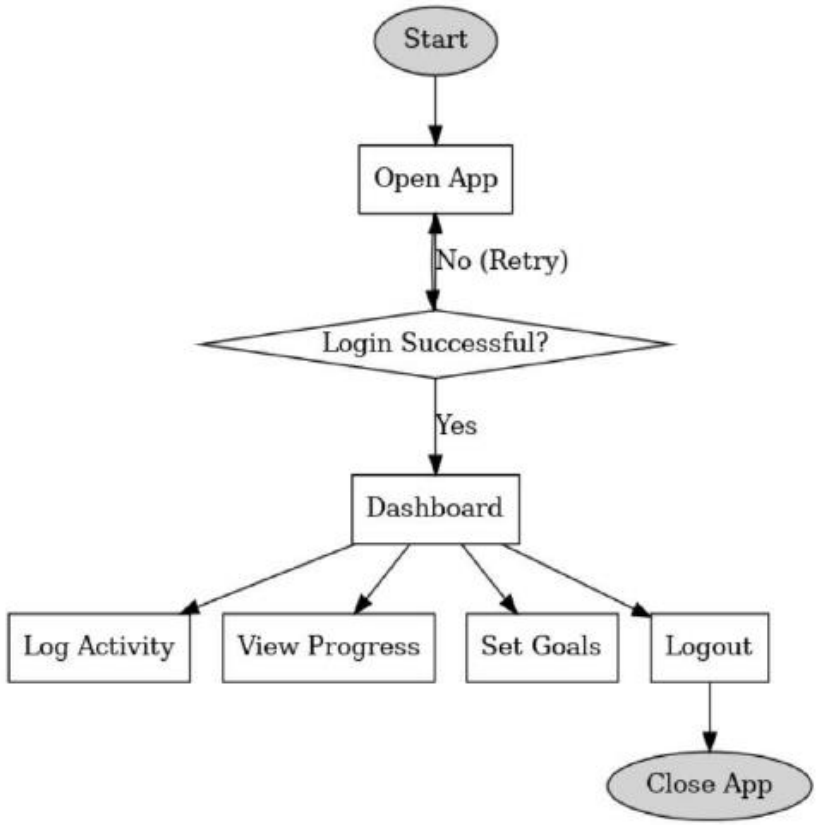
DFD (Data Flow Diagram)



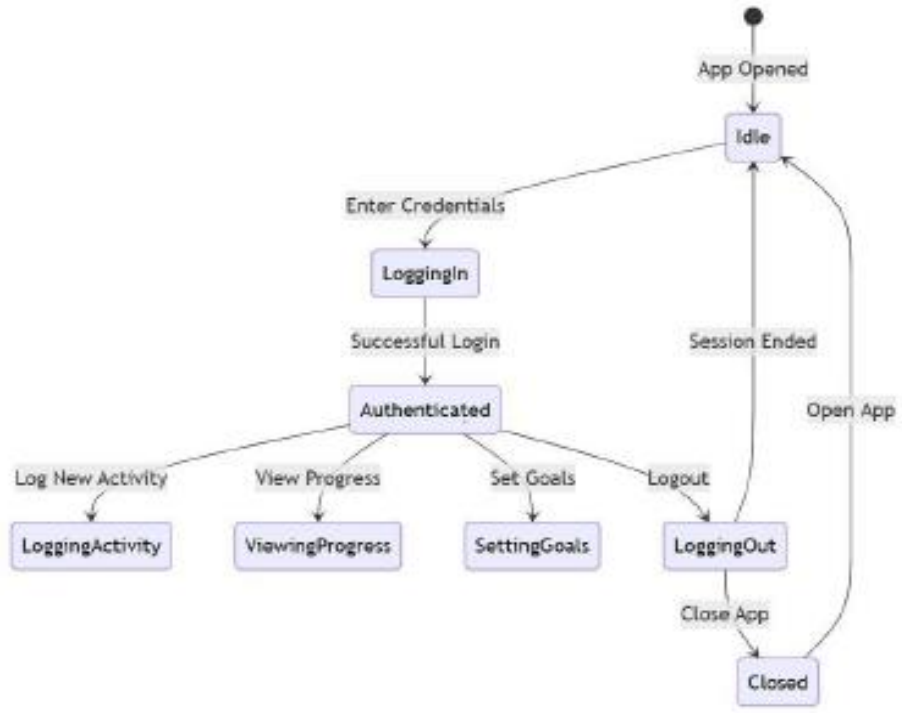
Sequence Diagrams



Activity Diagram

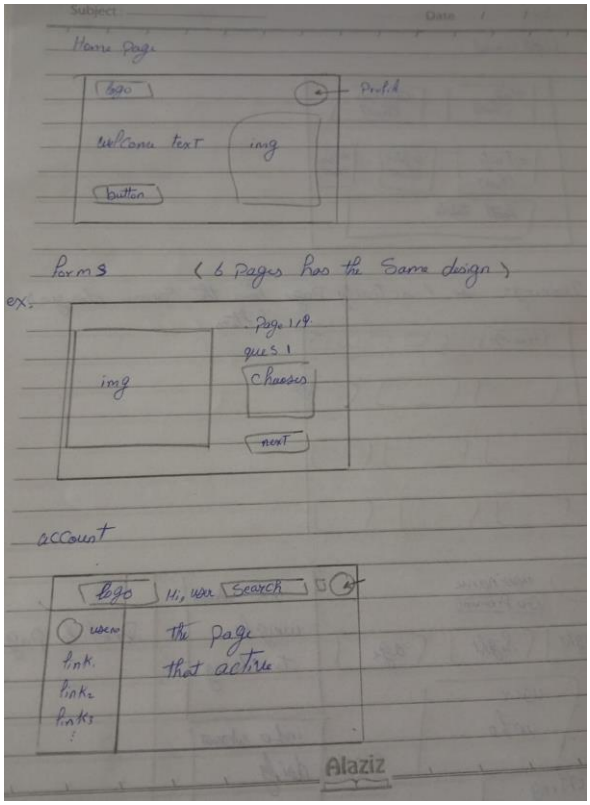


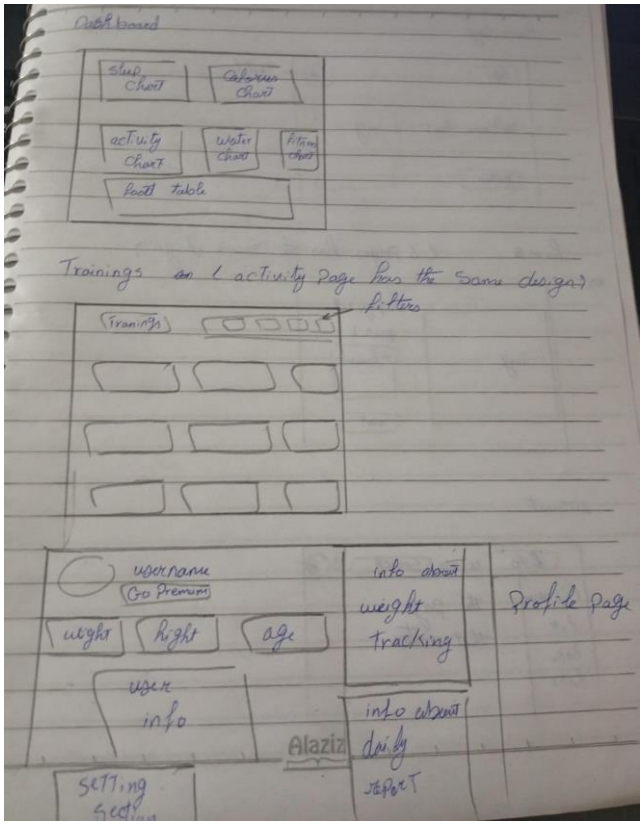
State Diagram



UI/UX Design & Prototyping

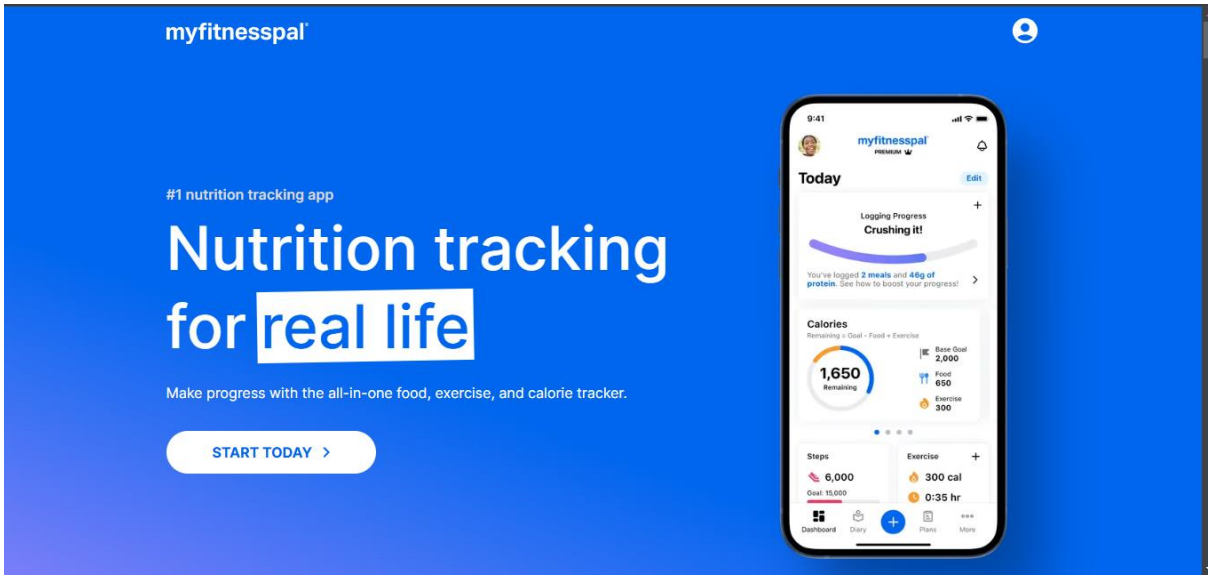
Wireframes



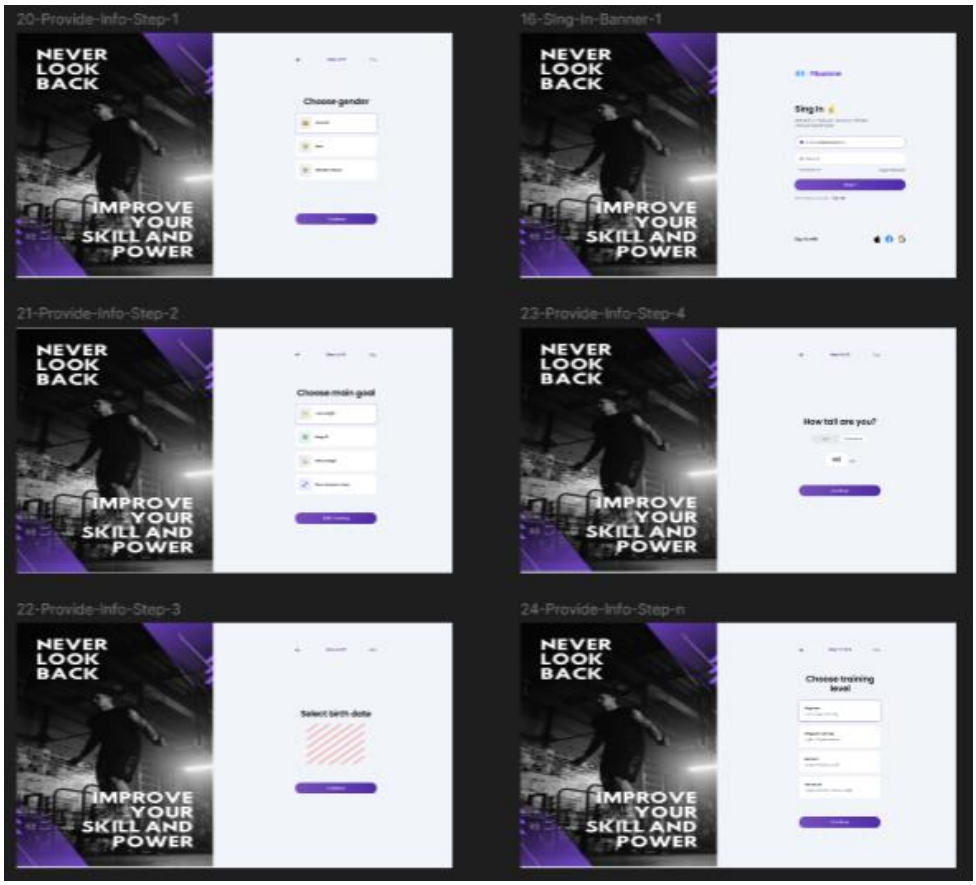


Mockups

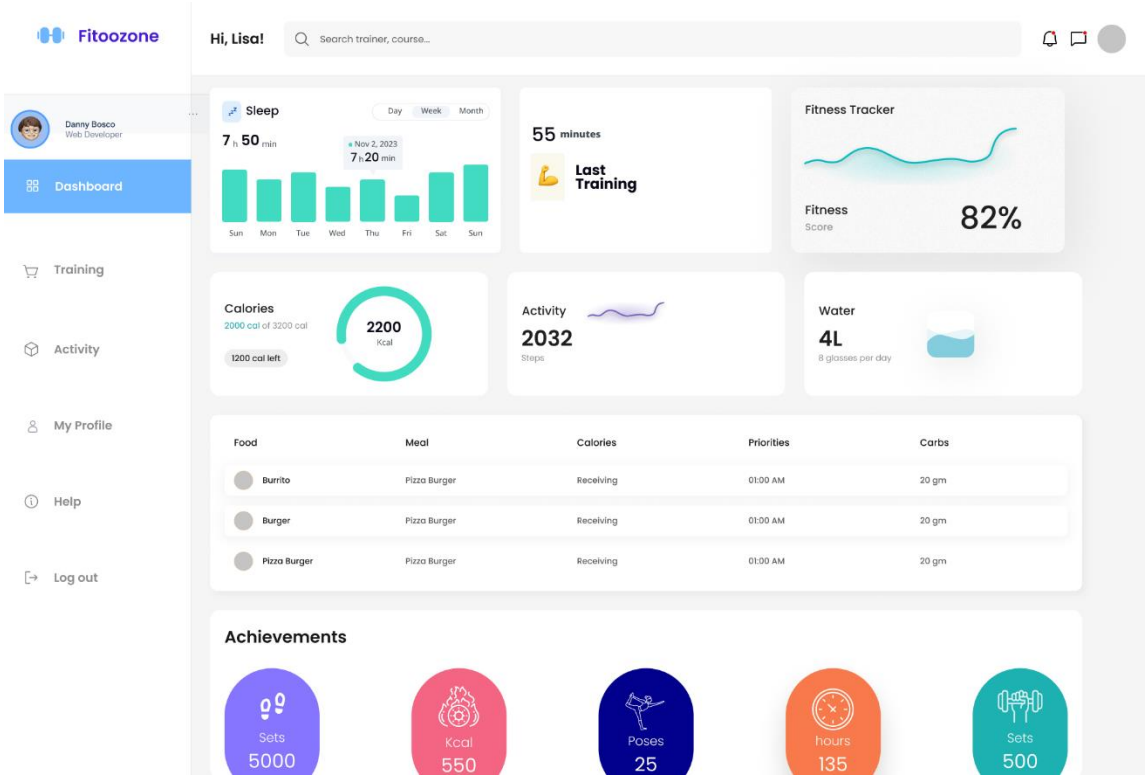
Home



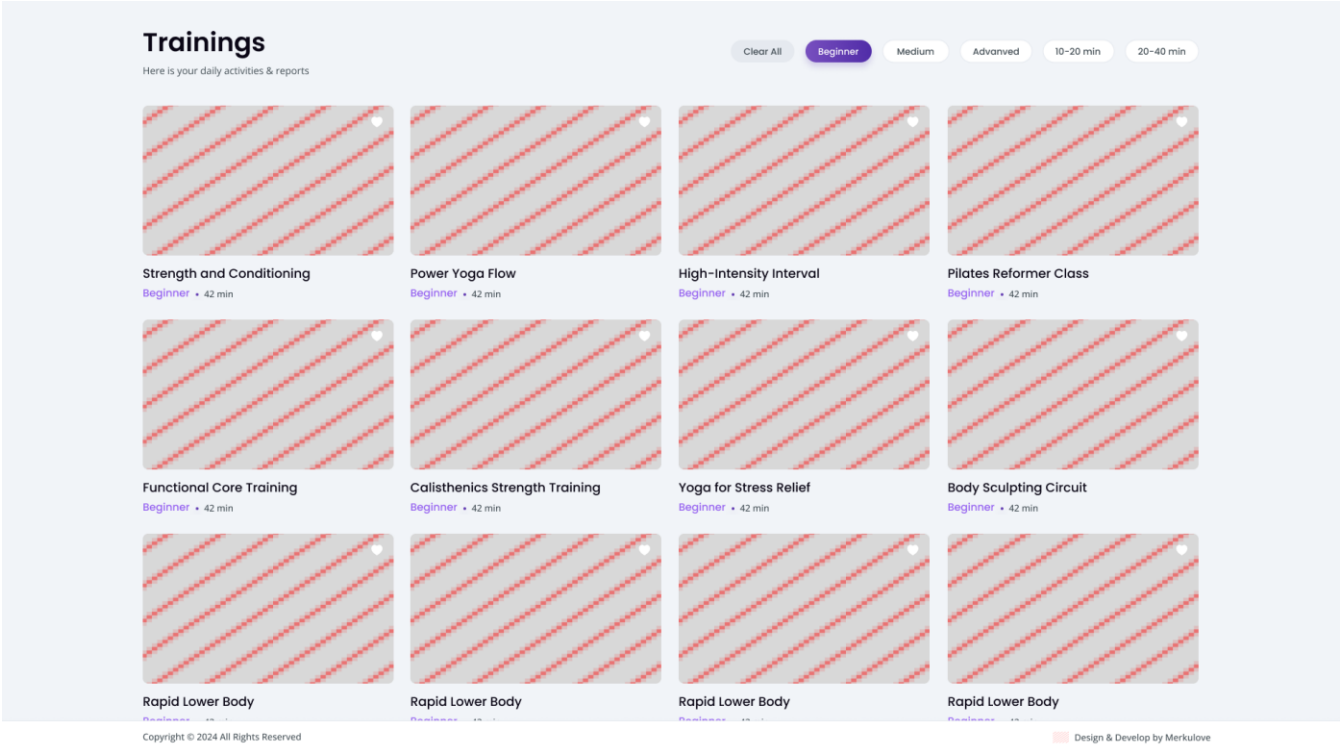
Forms



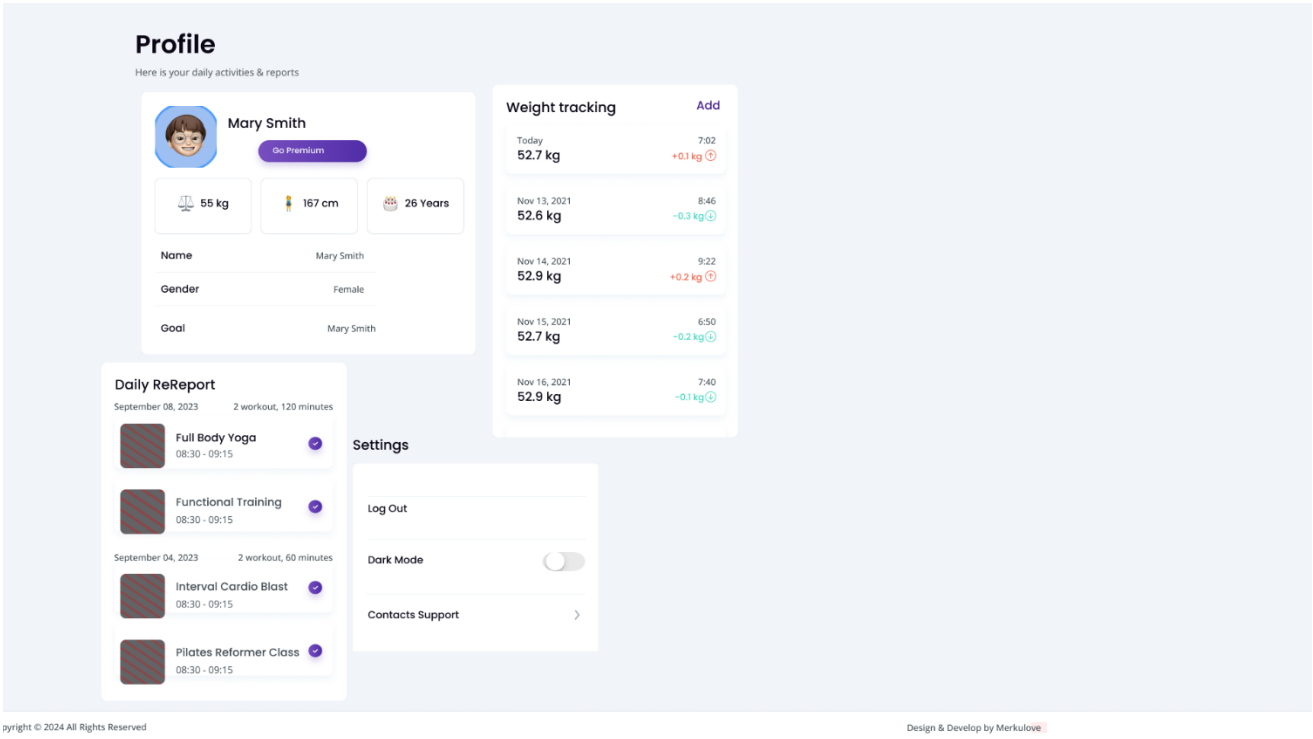
Account & Dashboard



Training & Activity



Profile



UI/UX Guidelines

1. Design Principles

- **Clarity & Simplicity:** A clean, minimalistic design for quick access to key info.
- **Consistency:** Unified styles for buttons, spacing, and layout.
- **User-Centric:** Intuitive navigation and easy fitness data access.
- **Visual Hierarchy:** Key elements emphasized using size, contrast, and spacing.
- **Responsive:** Optimized for various screen sizes.

2. Color Scheme

- **Primary Colors:** Vibrant hues (blue, green, orange) for motivation.
- **Secondary Colors:** Neutral tones for balance.
- **Accent Colors:** Highlights for key actions.
- **Contrast:** Ensures visibility and accessibility.

3. Typography

- **Font:** Modern sans-serif (Poppins, Inter, Roboto) for a clean look.
- **Hierarchy:** Bold for headings, lighter for body text.
- **Scaling:** Readable sizes with ample spacing.

Figma link:

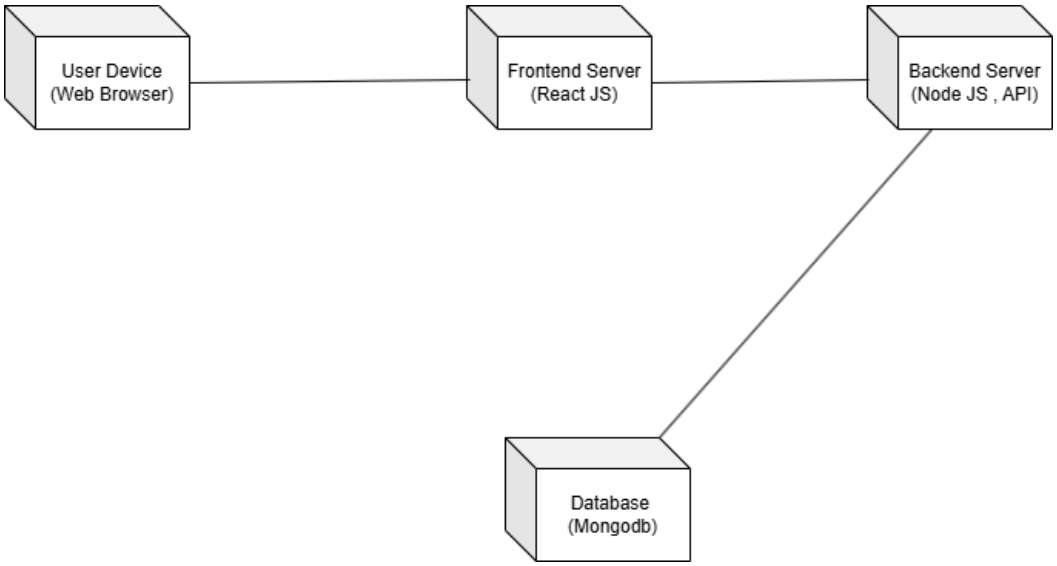
https://www.figma.com/design/afpo0RcH4EyHP33YSgcRmR/Fitness-Dashboard-UI-Kit-(No-Image)?node-id=0-1&t=s1qM3H154sLO4Pkq-1

System Deployment & Integration

Technology Stack

HTML5, CSS, JavaScript & React JS

Deployment Diagram



Component Diagram