**Project Management & Planning**

**1. Project Proposal**

**Overview**

The **Fitness Tracker Dashboard** is a React-based web application that enables users to track their fitness activities, monitor progress, and visualize data using interactive charts.

**Objectives**

Develop a **responsive** and **interactive** fitness dashboard.  
 Implement **real-time data tracking** (e.g., steps, calories, heart rate).  
 Use **charts and graphs** to display progress analytics.  
 Ensure a smooth **user experience** with a clean UI.

**Scope**

* **Frontend:** React (UI/UX, Charts, State Management).
* **Backend:** Firebase or Node.js (Data storage, Authentication).
* **Libraries:** Chart.js or Recharts for data visualization.
* **Users:** Fitness enthusiasts, gym-goers, trainers.

**2. Project Plan**

**Phases & Duration :**

| **Phase** | **Task** | **Duration** |
| --- | --- | --- |
| **Phase 1** | Project Setup & UI/UX Design | **1 Week** |
| **Phase 2** | Develop Core Components (Dashboard, Activity Tracker, Graphs) | **2 Weeks** |
| **Phase 3** | Backend Integration & API Development | **2 Weeks** |
| **Phase 4** | Testing & Bug Fixes | **1 Week** |
| **Phase 5** | Deployment & Final Optimization | **1 Week** |

**🛠 Milestones**

**After Week 1:** Finalize UI/UX design & React project setup.  
 **After Week 3:** Core frontend components completed.  
 **After Week 5:** Backend APIs connected & tested.  
 **After Week 6:** Bug fixes & optimization done.  
 **After Week 7:** Final deployment & project launch.

**📌 Deliverables**

* **Week 1:** UI Design Wireframes & Initial Project Setup.
* **Week 3:** Functional Fitness Dashboard UI with Charts.
* **Week 5:** Backend Integration with Authentication & Data Storage.
* **Week 6:** Tested & Optimized Application.
* **Week 7:** Final Deployment Ready for Users.

**3. Task Assignment & Roles**

| **Role** | **Responsibility** |
| --- | --- |
| **Frontend Developer** | Build UI using React, state management, data visualization. |
| **Backend Developer** | Create APIs for data storage, user authentication. |
| **UI/UX Designer** | Design dashboard layout, ensure smooth UX. |

**4. Risk Assessment & Mitigation Plan**

| **Risk** | **Mitigation Plan** |
| --- | --- |
| Performance issues | Optimize API calls, use caching & lazy loading. |
| Data loss/corruption | Regular backups, use Firebase/NoSQL. |
| UI responsiveness issues | Use CSS frameworks (Tailwind, MUI) & test on devices. |
| Delayed API responses | Optimize database queries & implement pagination. |

**5. KPIs (Key Performance Indicators)**

**User Engagement** → Increase active users tracking fitness.  
 **System Uptime** → Ensure **99%+ uptime** for smooth tracking.  
 **Response Time** → API response time should be **< 500ms**.  
 **Data Accuracy** → Ensure correct logging of user activities.