### **Project Tracker - Service Charter Document**

### 

### **Purpose**

### To provide a centralized project management platform for teams. The platform helps:

### Track tasks status

### Monitor progress

### Manage deadlines

### Verify completed work

### Improve collaboration

### Manage documentation

### 

### **Objective**

### Simplify project management for cross-functional teams.

### Enable users to create, assign, and track tasks with real-time updates.

### Provide tools for progress monitoring (dashboards, timelines and virtual meetings).

### 

### **Demand / Opportunity**

### Responds to the increasing demand for simple yet effective project tracking tools in small-team environments.

### Fills the gap left by overly complex tools or those not designed for student needs or basic needs.

### Helps organizations and teams move away from inefficient solutions like spreadsheets or informal chats.

### 

### **Business Requirement**

#### **4.1 User Authentication**

### Login system based on email and password with JWT tokens for session handling.

#### **4.2 Project & Task Management**

### CRUD operations for projects, tasks, and milestones with proper role-based permissions.

#### **4.3 Progress Tracking**

### Dashboards, charts, and Gantt timelines to monitor task and milestone statuses.

#### **4.4 Real-Time Notifications**

### Email or in-app notifications for task updates, deadlines, or completion.

#### **4.5 Role Assignment**

### Users assigned as project creators, project completers; ability to comment and give feedback.

### **4.6 Real-time Communication and collaboration**

### Users can set up virtual meets to discuss progress and prevent miscommunication.

### **4.7 Download Project Report**

### Users can download detailed reports containing project information, milestones, tasks, team members and deadlines.

### 

### 

### **Technical Requirement**

#### **5.1 User Authentication**

### API routes for login, registration, and session handling using JWT.

### Password encryption and role-based access.

#### **5.2 Project Management**

### Backend logic for project/task CRUD operations.

### Role assignment and deadline tracking.

#### **5.3 Progress Visualization**

### Charts, timelines using libraries like Chart.js or D3.js.

#### **5.4 Notification**

### Integration with NodeMailer for real-time and email notifications.

### 

#### **5.5 Collaboration and communication**

### Integration with Google Calendar to automatically schedule virtual meetings.

### 

### **Technological Requirement**

#### **6.1 Frontend**

### React.js

### Tailwind CSS

### shadcn/ui

#### **6.2 Backend**

### Node.js with Express.js

#### **6.3 Database**

### MySQL

#### **6.4 Authentication**

### JWT

#### **6.5 Virtual meetings**

### Google Meet through Google Calendar (.ics invites)

#### **6.6 Tools**

### Visual Studio Code

### Git & GitHub

### Postman

### AWS Services (EC2, RDS, S3, etc.)

### 

### **Stakeholders**

| **Stakeholder** | **Name** | **Count** |
| --- | --- | --- |
| Teams/Users | NA | - |
| Developers | Dhanwantari, Viraj, Om, Shripad, Parth | 5 |
| Testers | Srivaths, Hanumant, Prajwal, Avishkar | 4 |
| Cloud Service Provider | AWS | 1 |
| Third-parties | React, Node.js, Tailwind Developers, etc. | - |

### 

### 

### **Resources Needed**

#### **8.1 Documentation**

### React.js

### Tailwind CSS

### MySQL

### Express.js + JWT

### Cloud Account

#### **8.2 Human Resource\***

| **Role** | **Count** | **Names** |
| --- | --- | --- |
| Frontend Developers | 2 | Parth, Dhanwantari |
| Backend Developers | 3 | Viraj, Shripad, Om |
| API Integration | 2 | Om, Dhanwantari, Viraj |
| DB Designer | 2 | Shripad, Parth |
| Documentation | 2 | Parth, Dhanwantari, Om |
| Testers | 4 | Srivaths, Hanumant, Prajwal, Avishkar |

### 

### 

### **PESTEL Analysis**

#### **9.1 Political**

### No direct impact; follows industrial policies.

#### **9.2 Economic**

### Cost-efficient, scalable. Reduces project delays.

#### **9.3 Social**

### Promotes structured collaboration and accountability in teams.

#### **9.4 Technological**

### Built on modern tech stack; supports scalability, extensibility, and cross-device use.

#### **9.5 Legal**

### Compliant with IT Act 2000 and GDPR policies for user data protection.

#### **9.6 Environmental**

### No environmental impact; hosted on cloud infrastructure.

### 

### **Risk Analysis**

| **Risk** | **Description** | **Mitigation Strategy** |
| --- | --- | --- |
| Data Breaches | Sensitive data can be exposed. | Use JWT, encrypt sensitive info, log access, apply MFA. |
| Downtime | Service unavailability due to AWS, DB or API outages. | Use multi-AZ deployment and load balancing. |
| Compliance | GDPR or institutional regulation violations. | Follow legal protocols; seek data protection officer’s review. |
| High Cost | Budget overrun due to third-party services. | Optimize code, use serverless APIs wherever possible. |
| UI/UX Confusion | Users might find it hard to navigate initially. | Conduct usability testing and regular UX audits. |
| Security Weakness | Improper access control or outdated tokens. | Use JWT with expiration and secure role checks in backend APIs. |
| Blank data exports | The exported PDF file might be empty due to missing data or export failure. | Add data validation before export; show users a warning if data is missing. |

### 

### **11. Timeline / Milestone**

| **Phase** | **Milestone Tasks** | **Timeline** |
| --- | --- | --- |
| 1. Planning & Wireframing | Requirement analysis, UI mockups | Week 1 |
| 2. Backend Setup | API creation, DB schema, auth system | Week 2 |
| 3. Frontend Development | Develop UI components, dashboard, timeline view | Week 2-4 |
| 4. API Creation and Integration | Connect frontend with backend, test validations | Week 3-5 |
| 5. Testing | UI/UX and functional testing, bug fixing | Week 5-6 |
| 6. Deployment | Host on AWS, CI/CD pipeline setup, monitoring | Week 6 |

### 

### 

### 

### 

### 

### 

### 

### **12. RACI Chart**

| **ACTIVITY / MEMBERS** | **Parth** | **Dhanwantari** | **Viraj** | **Om** | **Shripad** | **Jayesh Sir** | **Anuradha Mam** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Design Database** | **R** | **I** | **I** | **I** | **R & A** | **C** | **I** |
| **API Development** | **C** | **I** | **R & A** | **R** | **C** | **C** | **I** |
| **API Testing** | **I** | **R** | **R** | **R&A** | **C** | **I** | **I** |
| **UI Design / Development** | **R** | **R & A** | **C** | **I** | **I** | **C** | **I** |
| **API Integration** | **I** | **C** | **R** | **A** | **R** | **C** | **I** |
| **Project Management & Documentation** | **R & A** | **R** | **C** | **C** | **C** | **I** | **I** |
| **Project Deployment** | **C** | **I** | **R** | **R** | **R&A** | **C** | **I** |

### 

### 

### 

### 

### 

### **13. Budget**

| **Service** | **Monthly Cost (INR)** |
| --- | --- |
| AWS EC2 | 567.07 |
| AWS RDS (MySQL) | 7119.80 |
| AWS S3 + CloudWatch | 2000.00 |
| Development Cost | 20000 \* 5 = 100000 |
| Total Estimated Cost | 109686.87 INR |