**TASK MANAGEMENT SYSTEM — FULL PROJECT PLAN**

**1. PROJECT OVERVIEW**

The **Task Management System** is a full-stack web application that allows users to register, log in, and manage their daily tasks efficiently.  
The system supports multiple user roles (Admin and User), advanced features (Email notifications, Scheduler, Report Export, Pagination, Charts), and has a clean, responsive UI.

**2. TECHNOLOGIES USED**

| **Layer** | **Technology** | **Purpose** |
| --- | --- | --- |
| **Frontend** | React.js | Build dynamic, responsive UI |
| **Backend** | Spring Boot | Build RESTful APIs and manage logic |
| **Database** | MySQL | Store users, tasks, and roles |
| **Testing** | Postman | API testing and verification |
| **Styling** | Bootstrap 5 / Tailwind CSS | UI design and responsiveness |
| **Visualization** | Chart.js / Recharts | Admin dashboard graphs |
| **Mail Service** | Spring Boot Mail (JavaMailSender) | Send task reminders via email |
| **PDF Export** | iTextPDF / Apache POI | Generate reports |
| **Version Control** | Git & GitHub | Team collaboration and versioning |

## 3. BACKEND STRUCTURE (Spring Boot)

Backend handles:

* User authentication
* CRUD for tasks
* Role-based access (Admin/User)
* Business logic
* Database communication
* Advanced features (email, scheduler, reports)

**Backend Folder & File Structure**

**taskmanager-backend/**

**┣ src/main/java/com/taskmanager/**

**┃ ┣ 📁 controller/**

**┃ ┃ ┣ AuthController.java**

**┃ ┃ ┣ TaskController.java**

**┃ ┃ ┗ AdminController.java**

**┃ ┣ 📁 service/**

**┃ ┃ ┣ UserService.java**

**┃ ┃ ┣ TaskService.java**

**┃ ┃ ┣ EmailService.java**

**┃ ┃ ┗ ReportService.java**

**┃ ┣ 📁 repository/**

**┃ ┃ ┣ UserRepository.java**

**┃ ┃ ┗ TaskRepository.java**

**┃ ┣ 📁 model/**

**┃ ┃ ┣ User.java**

**┃ ┃ ┣ Task.java**

**┃ ┃ ┗ Role.java**

**┃ ┣ 📁 config/**

**┃ ┃ ┣ SecurityConfig.java**

**┃ ┃ ┗ SchedulerConfig.java**

**┃ ┣ 📁 exception/**

**┃ ┃ ┗ GlobalExceptionHandler.java**

**┃ ┗ TaskManagerApplication.java**

**┣ src/main/resources/**

**┃ ┗ application.properties**

**┗ pom.xml**

**Explanation of Main Backend Files**

| **File** | **Purpose** |
| --- | --- |
| **TaskManagerApplication.java** | Main Spring Boot starter file |
| **User.java / Task.java / Role.java** | Entity classes defining database tables |
| **UserRepository.java / TaskRepository.java** | Interfaces for database operations |
| **UserService.java** | Handles user registration, login logic |
| **TaskService.java** | Handles CRUD and business logic for tasks |
| **EmailService.java** | Sends email reminders for due tasks |
| **ReportService.java** | Generates PDF or CSV task reports |
| **AuthController.java** | Handles register/login endpoints |
| **TaskController.java** | Manages user task APIs (CRUD) |
| **AdminController.java** | Provides admin statistics endpoints |
| **SecurityConfig.java** | Sets up Spring Security (login, roles) |
| **SchedulerConfig.java** | Automates task checking using @Scheduled |
| **GlobalExceptionHandler.java** | Centralized error handler for APIs |
| **application.properties** | Contains DB credentials and configurations |

## 4. FRONTEND STRUCTURE (React.js)

Frontend handles:

* User interface and interaction
* API communication using Axios
* Displaying data from backend
* Task management (CRUD)
* Charts and reports for admin

**Frontend Folder & File Structure**

**taskmanager-frontend/**

**┣ src/**

**┃ ┣ 📁 components/**

**┃ ┃ ┣ Login.js**

**┃ ┃ ┣ Register.js**

**┃ ┃ ┣ Dashboard.js**

**┃ ┃ ┣ TaskList.js**

**┃ ┃ ┣ TaskForm.js**

**┃ ┃ ┣ Navbar.js**

**┃ ┃ ┗ AdminDashboard.js**

**┃ ┣ 📁 services/**

**┃ ┃ ┗ api.js**

**┃ ┣ 📁 context/**

**┃ ┃ ┗ AuthContext.js**

**┃ ┣ 📁 assets/**

**┃ ┃ ┣ css/**

**┃ ┃ ┗ images/**

**┃ ┣ App.js**

**┃ ┗ index.js**

**┣ package.json**

**Explanation of Main Frontend Files**

| **File** | **Purpose** |
| --- | --- |
| **App.js** | Main app structure with routing |
| **Navbar.js** | Navigation bar (Dashboard, Tasks, Admin, Logout) |
| **Login.js** | User login page connected to backend /api/auth/login |
| **Register.js** | User signup page connected to /api/auth/register |
| **Dashboard.js** | User overview: stats, completed/pending tasks |
| **TaskList.js** | Lists all tasks fetched from backend |
| **TaskForm.js** | Used to add or edit tasks |
| **AdminDashboard.js** | Displays charts, total users, and system reports |
| **api.js** | Central file for all Axios API calls |
| **AuthContext.js** | Stores user login state and authentication token |
| **css/** | Stylesheets for UI |
| **images/** | Static image assets (logos, icons, etc.) |

## 5. DATABASE STRUCTURE (MySQL)

### 🗄️ Tables:

1. **User**
   * id (PK)
   * username
   * password
   * email
   * role
2. **Task**
   * id (PK)
   * title
   * description
   * priority
   * status
   * due\_date
   * user\_id (FK)
3. **Role**
   * id (PK)
   * name (ADMIN / USER)

### **Relationships**

* **User ↔ Task** = One-to-Many  
  (One user can have many tasks)
* **Task ↔ User** = Many-to-One  
  (Each task belongs to one user)

**6. SPECIAL FEATURES**

|  | **Feature** | **Description** | **Tools Used** |
| --- | --- | --- | --- |
|  | **Email Notifications** | Sends reminders for tasks near due date | Spring Boot Mail |
|  | **Task Scheduler** | Automatically runs daily to check overdue tasks | @Scheduled |
|  | **Report Export** | Users can export tasks as PDF/CSV | iTextPDF / Apache POI |
|  | **Search & Pagination** | Filters and navigates tasks easily | Spring Data Pagination |
|  | **Admin Dashboard** | Displays charts of tasks/users | Chart.js / Recharts |

**7. TEAM ROLE DIVISION (8 MEMBERS)**

| **Member** | **Role** | **Main Responsibilities** |
| --- | --- | --- |
| **Member 1** | Backend Developer | Design MySQL tables, build entity classes (User, Task, Role), and repositories |
| **Member 2** | Backend Developer | Implement TaskService, CRUD, filtering, and email notification feature |
| **Member 3** | Backend Developer | Add Spring Security (login, roles), and implement task scheduler |
| **Member 4** | Backend Developer | Create ReportService for PDF/CSV export and connect it to controller |
| **Member 5** | Frontend Developer | Setup React project, implement routing & AuthContext.js |
| **Member 6** | Frontend Developer | Build Login & Register pages and connect them with backend |
| **Member 7** | Frontend Developer | Create Dashboard, TaskList, and TaskForm components |
| **Member 8** | Frontend Developer | Design Admin Dashboard (charts, user stats) and improve UI/UX with Bootstrap |

**8. PROJECT WORKFLOW**

**A screenshot of a computer screen

Description automatically generated**

**Example Flow:**

1. User registers on frontend → sent to /api/auth/register
2. Backend validates and saves user → MySQL
3. User logs in → gets authenticated (Spring Security)
4. Frontend dashboard loads tasks via /api/tasks
5. Scheduler checks for overdue tasks and triggers EmailService
6. Admin can view stats & export report

**9. TESTING PLAN**

| **Tool** | **Purpose** |
| --- | --- |
| **Postman** | Test all backend APIs |
| **Browser (React)** | Test UI functionality |
| **Console Logs** | Debug frontend/backend integration |
| **Database Checks** | Verify data saved correctly in MySQL |