## **Task Management System with React: Project Plan**

### **Phase 1: Project Planning & Requirements Gathering**

#### **1.1 Project Proposal**

* Overview: Develop a Task Management System to help users create, update, and delete tasks. Users can categorize tasks, set priority levels, and filter tasks by status.
* Objective:
  + Build a user-friendly task management tool with React and Redux for state management.
  + Enable users to create, assign, and manage tasks with features like categorization, priority levels, and due dates.
  + Implement filtering and search functionality for tasks.
* Scope:
  + Task creation, updating, and deletion.
  + Task categorization and priority levels.
  + Filtering and searching tasks by status (completed, in-progress).

#### **1.2 Stakeholder Analysis**

* Internal Stakeholders:
  + Team Members: Mohammed amir , Abdelrhman Khalid , Abdelrahman OKda, Mohamed zaghlol.
* External Stakeholders:
  + Users: Individuals or teams managing tasks.

#### **1.3 Requirements Gathering**

* Functional Requirements:
  + User authentication (optional for this project).
  + Task creation, updating, and deletion.
  + Task categorization and priority levels.
  + Filtering and searching tasks by status.
  + Data visualization (e.g., task progress charts).

### **Phase 2: System Design & Architecture**

#### **2.1 System Architecture**

* Frontend: React with Context API for state management.
* Data Flow:
  + Frontend sends requests to the backend API.
  + Backend sends responses back to the frontend.

#### **2.3 UI/UX Design**

* Wireframes: Design layouts for task lists, task creation, and filtering.
* Prototypes: Create interactive prototypes using Figma.
* Design Principles:
  + Consistency: Uniform design across all pages.
  + Simplicity: Avoid clutter and prioritize essential elements.
  + Feedback: Provide real-time updates and visual feedback.

### **Phase 3: Development**

#### **3.1 Frontend Development**

* Tasks:
  + Develop React components for task lists, task creation, and filtering.
  + Integrate Redux for global state management.
  + Implement data visualization (e.g., task progress charts).
* Deliverables:
  + Functional UI with responsive design.

### **Phase 5: Deployment**

#### **5.1 Hosting Environment**

* Frontend: Deploy React app to Netlify

#### **5.2 CI/CD Pipeline**

* Frontend: Automate deployment using Netlify

#### **6.1 Documentation**

* Update documentation with new features and changes.
* Provide user guides and API documentation.

## **Task Assignment**

| **Task** | **Assigned To** | | |
| --- | --- | --- | --- |
| Project Planning & Requirements | | Mohammed Ahmed Zaghloul & Mohamed Amir |
| UI/UX Design & Wireframes | | Abdelrahman Okda |
| Frontend Development (React) | | Abdelrhman Khaled & Mohamed Amir |
| Deployment & Documentation | | Mohamed Amir |

## **Timeline**

| **Week** | **Phase** | **Tasks** |
| --- | --- | --- |
| 1 | Planning & Requirements | Finalize project scope, gather requirements, conduct stakeholder analysis. |
| 2-3 | Design | Create system architecture, and UI/UX prototypes. |
| 4-6 | Development | Develop frontend integrate components |
| 7 | Deployment | Deploy application, set up CI/CD pipeline, and configure monitoring. |
|  |  |  |