# Project Requirements: Task Management System

## Overview:

Build a Task Management System that allows users to manage their daily tasks. The application should provide functionalities to create new tasks, list all tasks, update task details, and delete tasks. Users should be able to filter tasks based on their status (e.g., completed, pending).

## Functional Requirements:

**User Authentication**

* **Sign Up**: Users should be able to sign up with an email and password.
* **Login**: Users should be able to log in with their email and password.
* **Logout**: Users should be able to log out of their account.

**Task Management**

* **Create Task**:
  + Users can create a new task with a title, description, due date, and status (e.g., pending, completed).
* **Read Tasks**:
  + Users can view a list of all their tasks.
  + Tasks should be displayed with their title, description, due date, and status.
* **Update Task**:
  + Users can update the details of a task (title, description, due date, status).
* **Delete Task**:
  + Users can delete a task.

**Task Filtering**

* **Filter by Status**:
  + Users can filter tasks based on their status (e.g., show only completed tasks).

**Task Search**

* **Search by Title/Description**:
  + Users can search for tasks based on the task title or description.

**User Profile**

* Allow users to update their profile information (e.g., name, email).

**Task Priority**

* Add a priority field to tasks (e.g., low, medium, high) and allow users to sort tasks by priority.

**Task Notifications**

* Implement notifications to remind users of upcoming due dates for tasks.

**Collaborative Tasks**

* Allow users to share tasks with other users and collaborate on task completion.

## Technical Requirements:

**1. Project Setup**

* Use Angular CLI to set up the project.
* Structure the project into modules (e.g., AuthModule, TaskModule).

**2. Components**

* **HeaderComponent**: Displays navigation links (e.g., Home, Tasks, Login, Logout).
* **TaskListComponent**: Displays a list of tasks.
* **TaskItemComponent**: Displays individual task details.
* **TaskFormComponent**: Form for creating and updating tasks.
* **LoginComponent**: Form for user login.
* **SignupComponent**: Form for user registration.

**3. Services**

* **AuthService**: Handles user authentication (sign up, login, logout).
* **TaskService**: Handles CRUD operations for tasks (create, read, update, delete).

**4. State Management**

* Use a state management solution like NgRx or a simple service with BehaviorSubject to manage the application state.

**5. Routing**

* **Home Route**: /home - Displays a welcome message or dashboard.
* **Task List Route**: /tasks - Displays the task list.
* **Task Detail Route**: /tasks/:id - Displays task details and allows for editing.
* **Login Route**: /login - Displays the login form.
* **Signup Route**: /signup - Displays the sign-up form.

**6. Forms**

* Use Reactive Forms for handling user inputs in task creation and update forms.

**7. Validation**

* Implement form validation for user inputs (e.g., required fields, email format).

**8. HTTP Requests**

* Use Angular's HttpClient to communicate with a mock backend (e.g., JSON server or Firebase) for task and user data.

**9. Error Handling**

* Implement error handling for HTTP requests and display user-friendly error messages.

**10. Unit Testing**

* Write unit tests for components, services, and pipes using Jasmine and Karma.
* Ensure high test coverage for critical parts of the application.