**Task Specification: To-Do List Application**

**Description:**

You are required to build a basic To-Do List application using the MEAN stack (MongoDB, Express.js, Angular, and Node.js). The application should allow users to create, read, update, and delete tasks.

**Requirements:**

1. **Backend (Node.js and Express):**
   * Set up a Node.js server using Express.js.
   * Implement RESTful APIs for CRUD operations on tasks.
   * Use MongoDB for storing tasks.
2. **Frontend (Angular):**
   * Create a user interface for the To-Do List using Angular.
   * Display a list of tasks with options to add, edit, and delete tasks.
   * Implement forms for adding and editing tasks.
   * Use Angular's HttpClient to communicate with the backend.
3. **Functionality:**
   * Users should be able to view all tasks.
   * Users should be able to add a new task.
   * Users should be able to edit existing tasks.
   * Users should be able to mark tasks as completed or delete tasks.
   * Tasks should persist even after refreshing the page.

**Implementation Guidelines:**

* **Backend:**
  + Use Express.js for handling HTTP requests.
  + Use Mongoose for MongoDB object modeling.
  + Implement CRUD operations for tasks (GET, POST, PUT, DELETE).
* **Frontend:**
  + Use Angular CLI to generate components, services, and modules.
  + Create separate components for displaying tasks, adding/editing tasks, and individual task items.
  + Use Angular forms for adding and editing tasks.
  + Implement services to interact with backend APIs using HttpClient.
* **Database:**
  + Use MongoDB for storing tasks.
  + Define a schema for tasks and interact with the database using Mongoose.

**Additional Features (Optional):**

* Authentication: Implement user authentication using JWT.
* Task Filtering: Allow users to filter tasks by status (completed/incomplete).
* Task Sorting: Allow users to sort tasks by priority, due date, etc.
* Task Categories: Allow users to categorize tasks into different categories (e.g., work, personal, etc.).