Capstone Project: Software Engineering

Project Overview:

You will design and build a full-stack web application for a **task management system** called **TaskMaster**. This project will integrate various concepts such as Node.js, Express.js, databases, APIs, and front-end development.

Project Steps:

1. **Project Planning:**
    - **Objective**: Build a task management system that allows users to create, update, and delete tasks. Users can also organize tasks by priority and set deadlines. Your system should be scalable, secure, and optimized for performance.
    - **Tech Stack**:
        - **Backend**: Node.js, Express.js
        - **Database**: MongoDB or PostgreSQL
        - **Frontend**: HTML, CSS and JavaScript
2. **Core Functionalities:**
    - **User Registration and Authentication**:
        - Implement a secure registration and login system using **JWT (JSON Web Tokens)** or **sessions**.
        - Include password hashing using **bcrypt**.
    - **Task Management**:
        - Users can create tasks with attributes such as title, description, deadline, and priority (low, medium, high).
        - Users can update or delete tasks.
    - **Task Filtering**:
        - Implement a filtering system where users can filter tasks based on priority or due date.
    - **Search Functionality**:
        - Implement a search bar to search for tasks based on keywords in the title or description.
3. **Backend Development:**
    - Set up a **Node.js** server with **Express** to handle API requests for task management.
    - Use **RESTful API principles** to build routes for creating, reading, updating, and deleting tasks.
    - Implement **CRUD operations** with database integration (MongoDB/PostgreSQL).
4. **Database Design:**
    - Design a **task management schema** that stores user information and tasks. Ensure relationships between users and tasks are well-defined.

    - Use **NoSQL** (MongoDB) or **SQL** (PostgreSQL) for storing and retrieving data. Create collections/tables for users and tasks.
5. **Frontend Development:**
    - Design a user-friendly interface that allows users to manage their tasks efficiently.
    - Implement **AJAX** or **Fetch API** calls to communicate with the backend, ensuring asynchronous task management.
    - Use **event listeners** for buttons and input fields to handle user actions (e.g., adding or deleting tasks).
6. **Asynchronous Operations & Error Handling**:
    - Use **promises and async/await** for handling asynchronous operations like task creation and updating.
    - Implement proper **error handling** on both the client and server sides (e.g., validation errors, server errors).
7. **Testing & Deployment**:
    - Write **unit tests** for key functions, such as task creation and user authentication.
    - Deploy your API on **https://fly.io/ and frontend on Vercel or Netlify,** ensuring that the app is accessible online. Use **Git** for version control.
    - Implement **basic security features** such as input validation and protection against SQL injection or XSS attacks.
8. **Final Presentation**:
    - Present your application in a **live demo** where you walk through the core features.
    - Prepare a **2-minute video** showcasing how TaskMaster solves the problem of task organization.

Submission Requirements:

- Deploy and submit the link to your live TaskMaster app (for the Software Engineering Capstone).
- Include code repositories, final reports, and presentations as part of your submission.