

6. Task: Build a Task Management Application (ASP.NET Core + Angular)

Situation: You need to build a simple task management application where users can create, assign, and track tasks.

Task Requirements:

- **Backend (ASP.NET Core):**
 - Create a Task model with fields like Title, Description, AssignedTo, Status (e.g., Pending, In Progress, Completed), and DueDate.
 - Implement the following API endpoints:
 - POST /tasks: Create a new task.
 - GET /tasks: Get a list of all tasks.
 - GET /tasks/{id}: Retrieve a task by ID.
 - PUT /tasks/{id}: Update a task's status or details.
 - DELETE /tasks/{id}: Delete a task.
 - Implement basic validation (e.g., required fields, proper date format).
 - **Frontend (Angular):**
 - Create a list of tasks and display their status.
 - Implement filters to view tasks by status or assigned user.
 - Create a form to add or edit tasks.
 - Use Angular Forms to handle task creation and editing.
 - Implement routing to navigate between task lists and task details.
-

7. Task: Build an Online Polling System (ASP.NET Core + React)

Situation: You need to create an online polling system where users can create polls and vote on them.

Task Requirements:

- **Backend (ASP.NET Core):**
 - Create a Poll model with fields like PollId, Question, Options (list of options), and CreatedAt.
 - Create an API endpoint to create a poll (POST /polls).
 - Create an endpoint to retrieve all polls (GET /polls).
 - Implement an endpoint to vote on a poll (POST /polls/{id}/vote).
 - Track votes and store the results (e.g., which option was selected).
- **Frontend (React):**
 - Display a list of polls and show the poll question and available options.
 - Allow users to vote on a poll and display the results.
 - Optionally, show a summary of voting results after a user votes.
 - Use Axios to send requests to the backend API.

8. Task: Build a Contact Management System (ASP.NET Core + Angular)

Situation: Your company needs a simple system to manage contacts, where users can add, update, and delete contact details.

Task Requirements:

- **Backend (ASP.NET Core):**
 - Create a `Contact` model with fields like `FirstName`, `LastName`, `Email`, `Phone`, and `Address`.
 - Implement the following API endpoints:
 - `POST /contacts`: Add a new contact.
 - `GET /contacts`: Get all contacts.
 - `GET /contacts/{id}`: Retrieve a contact by ID.
 - `PUT /contacts/{id}`: Update an existing contact.
 - `DELETE /contacts/{id}`: Delete a contact.
- **Frontend (Angular):**
 - Create a contact list component to display all contacts.
 - Implement a form to add and edit contact details.
 - Use Angular services to interact with the backend API.
 - Optionally, implement search or filtering functionality for contacts.