Project Based Learning – 1

Title: *Building a Task Management Application (MERN Stack)*

You have been assigned the task of building a task management application using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The application should allow users to create, update, and delete tasks, as well as mark tasks as completed. Implement the following steps to complete the case:

Step 1: Set up the project

- Set up a MongoDB database to store the task data.
- Create a Node.js backend using Express.js to handle API routes and interact with the database.
- Create a React.js frontend using Create React App to build the user interface.

Step 2: Backend Development

- Define the task data structure and create a MongoDB schema for tasks.
- Set up API routes for handling task creation, retrieval, update, and deletion.
- Implement the necessary API endpoints using Express.js, including route handlers and database interactions using a MongoDB driver or an ORM like Mongoose.

Step 3: Frontend Development

- Create a component for displaying the task list and rendering individual tasks.
- Implement functionality for fetching tasks from the backend API and displaying them in the UI.
- Create a form component for creating new tasks and update the task list upon submission.
- Add features to edit task details, mark tasks as completed, and delete tasks.

Step 4: Styling and UI Enhancements

- Apply CSS styles to improve the visual appearance of the task management application.
- Customize the UI to provide an intuitive user experience, such as using icons, colors, and layouts to enhance usability.
- Implement responsive design to ensure the application is accessible on different devices.

Step 5: Testing and Validation

- Test the application by creating, updating, and deleting tasks to ensure the desired functionality is working correctly.
- Verify that tasks and their details are correctly displayed and persisted in the database.
- Validate user input and handle potential errors or edge cases, such as empty task titles or invalid input.

This case study-based lab activity will help you practice building a complete MERN stack application, including backend development with Node.js and Express.js, frontend development with React.js, database integration with MongoDB, and deployment to a hosting platform. Customize and expand the features based on your learning goals and preferences. Happy coding!

Project Based Learning – 2

Title: *Building a Recipe Sharing Platform (MERN Stack)*

You have been assigned the task of building a recipe sharing platform using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The platform should allow users to create, view, and share recipes with others. Implement the following steps to complete the case:

Step 1: Set up the project

- Set up a MongoDB database to store recipe data.
- Create a Node.js backend using Express.js to handle API routes and interact with the database.
- Create a React.js frontend using Create React App to build the user interface.

Step 2: Backend Development

- Define the recipe data structure and create a MongoDB schema for recipes.
- Set up API routes for handling recipe creation, retrieval, update, and deletion.
- Implement the necessary API endpoints using Express.js, including route handlers and database interactions using a MongoDB driver or an ORM like Mongoose.

Step 3: Frontend Development

- Create a component for displaying the list of recipes and rendering individual recipe cards.
- Implement functionality for fetching recipes from the backend API and displaying them in the UI.
- Create a form component for users to submit new recipes and update the recipe list upon submission.
- Add features to view recipe details, such as ingredients, instructions, and user reviews.

Step 4: User Authentication and Authorization

- Implement user authentication using JWT (JSON Web Tokens) or a library like Passport.js.
- Allow users to register, log in, and log out.
- Restrict certain API endpoints and actions to authenticated users only.
- Implement authorization rules to ensure users can only edit or delete their own recipes.

Step 5: Styling and UI Enhancements

- Apply CSS styles to improve the visual appearance of the recipe sharing platform.
- Customize the UI to provide an intuitive and visually appealing user experience.
- Enhance the user interface with features such as image uploads for recipe photos and interactive elements for favoriting or rating recipes.

Step 6: Testing and Validation

- Test the application by creating, viewing, and interacting with recipes to ensure the desired functionality is working correctly.
- Verify that recipes and their details are correctly displayed and persisted in the database.
- Validate user input and handle potential errors or edge cases, such as empty fields or invalid data.

This case study-based lab activity will help you practice building a complete MERN stack application, including backend development with Node.js and Express.js, frontend development with React.js, database integration with MongoDB, user authentication and authorization, and deployment to a hosting platform. Customize and expand the features based on your learning goals and preferences. Happy coding!

Project Based Learning – 3

Title: *Building a Real-Time Chat Application (MERN Stack with Socket.io)*

You have been assigned the task of building a real-time chat application using the MERN (MongoDB, Express.js, React.js, Node.js) stack with Socket.io for real-time communication. The application should allow users to create chat rooms, join existing rooms, and exchange messages with other users. Implement the following steps to complete the case:

Step 1: Set up the project

- Set up a MongoDB database to store chat messages and user data.
- Create a Node.js backend using Express.js to handle API routes and interact with the database.
- Create a React.js frontend using Create React App to build the user interface.

Step 2: Backend Development

- Define the chat message data structure and create a MongoDB schema for chat messages.
- Set up API routes for user registration, login, and chat room creation.
- Implement authentication endpoints using libraries like Passport.js or JSON Web Tokens (JWT).
- Integrate Socket.io with your backend to enable real-time communication.

Step 3: Frontend Development

- Create a component for user registration and login.
- Implement functionality to authenticate users and store authentication tokens.
- Create a component for creating and joining chat rooms.
- Set up Socket.io on the frontend to establish a connection with the backend for real-time messaging.

Step 4: Chat Room and Messaging

- Create a component for displaying the list of available chat rooms.
- Implement functionality for users to join specific chat rooms.
- Create a chat component for displaying messages within a selected chat room.
- Implement the logic to send and receive messages in real-time using Socket.io.

Step 5: User Interface and Styling

- Apply CSS styles to improve the visual appearance of the chat application.
- Customize the user interface to provide an intuitive and visually appealing experience.
- Enhance the UI with features such as message timestamps, user avatars, and typing indicators.

Step 6: Testing and Validation

- Test the application by creating multiple user accounts and joining chat rooms.
- Verify that messages are exchanged in real-time and displayed correctly in the chat interface.
- Test the authentication and authorization features, ensuring that only authenticated users can join chat rooms and send messages.

This case study-based lab activity will help you practice building a real-time chat application using the MERN stack and Socket.io for real-time communication. It covers various aspects such as user authentication, chat room management, real-time messaging, and deployment. Customize and expand the features based on your learning goals and preferences. Happy coding!

Project Based Learning – 4

Title: *Building a Social Media Application (MERN Stack)*

You have been assigned the task of building a social media application using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The application should allow users to create profiles, post updates, follow other users, and engage with their posts. Implement the following steps to complete the case:

Step 1: Set up the project

- Set up a MongoDB database to store user profiles, posts, and interactions.
- Create a Node.js backend using Express.js to handle API routes and interact with the database.
- Create a React.js frontend using Create React App to build the user interface.

Step 2: Backend Development

- Define the user profile and post data structures and create MongoDB schemas for users and posts.
- Set up API routes for user registration, login, profile creation, post creation, and interactions (such as likes, comments, and shares).
- Implement authentication endpoints using libraries like Passport.js or JSON Web Tokens (JWT).

Step 3: Frontend Development

- Create components for user registration and login forms.
- Implement functionality to authenticate users and store authentication tokens.
- Create a profile creation form component to allow users to create their profiles.
- Implement a post creation form component to allow users to create posts.
- Create components to display user profiles, posts, and interactions (such as likes, comments, and shares).

Step 4: User Interaction

- Implement functionality to allow users to follow/unfollow other users.
- Create components to display posts from followed users on the user's feed.
- Implement features for liking, commenting on, and sharing posts.
- Implement a notification system to notify users about interactions on their posts or activities of followed users.

Step 5: User Interface and Styling

- Apply CSS styles to improve the visual appearance of the social media application.
- Customize the user interface to provide an intuitive and visually appealing experience.
- Enhance the UI with features such as user avatars, post previews, and dynamic content loading.

Step 6: Testing and Validation

- Test the application by creating user accounts, creating posts, and interacting with other users' posts.
- Verify that interactions, such as likes, comments, and shares, are correctly reflected in the UI and persisted in the database.
- Test the authentication and authorization features, ensuring that only authenticated users can create profiles, post updates, and engage with others' posts.

This case study-based lab activity will help you practice building a social media application using the MERN stack. It covers various aspects such as user authentication, profile management, post creation and interaction, and deployment. Customize and expand the features based on your learning goals and preferences. Happy coding!