Web Developement

PROJECTS

ZIDIO DEVELOPMENT

PROJECTS

Project name	**Location Tracker Website**
Difficulty level	Intermediate
Tools	**Frontend:** • HTML, CSS, JavaScript (React) **Backend:** • Node.js, Express.js **Database:** • MySQL or MongoDB **Geolocation API:** • Browser Geolocation API or third-party services

Features

User Authentication:
- Allow users to register, log in, and securely authenticate their identity.
- Ensure that each user has a unique profile tied to their account.

Real-time Location Tracking:
- Utilize the browser's Geolocation API or a third-party service to capture and display the user's real-time location on a map.
- Implement an updating mechanism to continuously track and refresh the user's position.

Map Integration with React:
- Integrate a mapping library, such as React Leaflet or Mapbox, to visualize the user's current location in a dynamic map component.
- Customize the map interface for a smooth and responsive user experience.

Location History and MySQL Database:
- Set up a MySQL database to store the user's location history.
- Implement a backend (Node.js with Express) to handle data storage and retrieval.
- Allow users to view their historical location data with timestamps.

User Dashboard with React:
- Create a React-based user dashboard where users can manage their account settings, track preferences, and view location history.
- Use React Router for seamless navigation within the application.

Notifications and Alerts:
- Implement notifications or alerts for significant location changes, arrivals, or departures.
- Use React state management to handle and display real-time alerts.

Security and Privacy Measures:
- Implement secure authentication practices using bcrypt for password hashing.
- Ensure encrypted communication between the client and server.
- Regularly update and patch security vulnerabilities.

PROJECTS

Project name	**Personal Finance Tracker Web Application**
Difficulty level	**Advance**
Tools	**Frontend:** • HTML, CSS, JavaScript (React) **Backend:** • Node.js, Express.js **Database:** • MySQL **Charting Library:** • Chart.js

Features

User Authentication and Profile Management:
- Allow users to register, log in, and manage their profiles.
- Implement secure authentication practices and password hashing.
- Enable users to update their personal information.

Expense Entry and Categorization:
- Create a user-friendly interface for adding, editing, and categorizing expenses.
- Implement expense categories (e.g., groceries, utilities, entertainment) to help users organize their spending.

Monthly Budgeting:
- Enable users to set monthly budgets for different expense categories.
- Provide visual feedback on budget utilization, with color-coded indicators.

Expense Analytics and Reports:
- Utilize charts and graphs (using Chart.js or a similar library) to visualize expense patterns and trends.
- Generate reports summarizing monthly spending, category-wise breakdowns, and budget adherence.

Data Persistence with MySQL:
- Set up a MySQL database to store user profiles, expense entries, and related data.
- Implement backend APIs (Node.js with Express) for CRUD operations on user and expense data.

Responsive Design with React:
- Develop the frontend using React to ensure a responsive and seamless user experience.
- Optimize the application for various screen sizes and devices.

Notification System:
- Implement a notification system to alert users when they are approaching or exceeding their budget limits.
- Allow users to customize notification preferences.

RULES to FOLLOW

Collaboration

- Organize a collaborative development team, with roles for frontend and backend developers, a database administrator, and a UI/UX designer.

Submission

- Submit the project code, documentation, and a detailed report covering the chosen technologies, features, challenges encountered, and potential improvements for the future.