

Project Guidelines

1. Form Validation with JavaScript Regex

- **Registration and Login Forms:**
 - Validate email formats, password strength (e.g., minimum length, inclusion of numbers/special characters).
- **Data Entry Forms:**
 - Ensure fields like dates, numbers (e.g., transaction amounts, flight numbers), and text inputs meet specified patterns.
- **Error Messaging:**
 - Provide real-time feedback to users on invalid inputs.

2. Asynchronous Operations with `async/await`

- **Fetch API:**
 - Use `async/await` for all client-server communication.
 - Handle loading states and errors gracefully.
- **File Operations on Server:**
 - Implement `async` functions for reading from and writing to JSON files.
 - Ensure data consistency and handle potential file access errors.

3. RESTful API Design

- **Consistent Naming Conventions:**
 - Use clear and consistent endpoint names (e.g., `/api/users` , `/api/tasks`).
- **HTTP Methods:**
 - Use GET for data retrieval, POST for creation, PUT/PATCH for updates, and DELETE for removals.
- **Status Codes:**
 - Return appropriate HTTP status codes for success and error scenarios.

4. Data Management

- **JSON File Structure:**
 - Organize data logically within JSON files (e.g., separate files for users, tasks, etc.).
- **Concurrency Handling:**

- Ensure that simultaneous file operations do not corrupt data (consider using file locks or atomic operations if applicable).

5. Security Considerations

- **Password Handling:**
 - Hash passwords before storing them (even if using JSON files).
- **Input Sanitization:**
 - Prevent injection attacks by sanitizing all user inputs.
- **Session Management:**
 - Securely manage user sessions or tokens to protect authenticated routes.

6. User Experience

- **Responsive Design:**
 - Ensure applications are accessible and functional across different devices and screen sizes.
- **Intuitive Navigation:**
 - Design clear navigation paths for users to access different features.
- **Feedback Mechanisms:**
 - Provide users with feedback on their actions (e.g., confirmations, error messages).

7. Documentation and Presentation

- **Code Documentation:**
 - Encourage commenting and clear code organization for maintainability.
- **Project Presentation:**
 - Prepare a demo showcasing key features, challenges faced, and solutions implemented.
- **README File:**
 - Include setup instructions, project descriptions, and any other relevant information.