Partner With Us Form – Documentation

This documentation provides an overview of the "Partner With Us" multi-step form, detailing the design choices, component structure, state management, and features like OTP verification, image upload, and persistent form data using local storage.

Project Structure and Component Design

The project follows a modular structure, dividing each section of the form into individual components to maintain clean and reusable code. The major components for these two pages are as follows:

1.1 Side Navigation Component:

- A vertical navigation sidebar on the left side to display the form progress.
- Highlights the current step, allowing users to understand which part of the form they're on.
- Displays eight sections in total: Business Information, Owner & Manager Details, PAN/Aadhaar Details, Legal Documents, Bank Details, Service Info, Preview Document, and Reach Increased.

1.2 Form Components:

 Each page is split into form components based on the information collected.

 Page 1 (Business Information):

- **Business Name**: Text input for the name of the business.
- **Country, State, City**: Dropdowns and text inputs for geographical details.
- **Address**: Text input for the business address.
- **Opening and Closing Times**: Time inputs for operating hours.
- **Email and Mobile Number**: Inputs for email and mobile number with OTP verification.
- **Image Upload**: For uploading an image of the restaurant.
- **Next Step Button**: Advances to the next section, Owner & Manager Details.

 Page 2 (Owner & Manager Details):

- **Owner Details Section**:
    - **Full Name**: Text input for the owner's name.
    - **Profile Picture Upload**: Image upload functionality for owner’s profile.
    - **Country, State, City**: Dropdown and text inputs for location.
    - **Address**: Text input for owner’s address.
    - **Email and Mobile Number**: Inputs with OTP verification, option to auto-fill from business details.
    - **Manager Details Prompt**: Option to fill manager details separately.

Button Components:

- Reusable button components for actions like "Send OTP," "Upload Image," and navigation.

State Management (Redux Toolkit)

Redux Toolkit is used for state management to handle form data, OTP verification, and image uploads. The global state stores form data, enabling users to navigate back and forth without losing information.

- **Slices**:
    - formSlice: Manages form data across multiple pages.
    - otpSlice: Manages OTP generation, verification status, and any errors.
- **Async Actions**:
    - **OTP Verification**: Thunk actions are used to make API calls for OTP generation and verification.
    - **Mock API Data Retrieval**: Retrieves mock data for country and state dropdowns from an API (if provided) to demonstrate asynchronous data handling.

Persistent Data with Browser Storage

To improve user experience, local storage is used to save form data, so fields on Page 1 auto-populate when revisited. On component mount, the app checks local storage for existing data and fills the form accordingly.

User Experience Enhancements

- **OTP Verification**:
    - Includes "Send OTP" buttons next to email and mobile fields.
    - Handles OTP errors and displays feedback to guide users if OTP verification fails.
- **Image Upload**:
    - Enables users to upload an image of their restaurant and owner profile picture.
    - Handles image previews and validates image type and size for optimal user experience.
- **Field Validation and Error Handling**:
    - Each required field has validation rules to ensure data completeness.
    - Displays error messages if users attempt to proceed without filling mandatory fields.
- **Navigation and Progress**:
    - Step-by-step navigation aids users in tracking progress.
    - Each step in the sidebar becomes active when the user navigates to the respective form section.

Styling with Tailwind CSS

Tailwind CSS is used for quick styling, ensuring a responsive and visually appealing layout:

- **Responsive Layout**: Adjusts the layout and elements to fit various screen sizes.

- **Consistent Design**: Tailwind utility classes provide a consistent style across pages, especially for buttons, form inputs, and error messages.

Deployment

Deployment of the project (optional but recommended) can be done using platforms like Vercel or Netlify, which support React applications. If deployed, include the URL in the submission.

Suggestions for Improvement

- **Progress Save Feature**: Implement an autosave feature to save form data after each field entry.
- **Data Validation on the Backend**: Add backend validation for email and mobile number to verify their authenticity.
- **Form Accessibility Enhancements**: Ensure all form elements are accessible, with proper ARIA labels and keyboard navigability.

How to Run the Project

1. **Clone the Repository**: Clone the GitHub repository.
2. **Install Dependencies**: Run npm install to install project dependencies.
3. **Start the Development Server**: Run npm start to start the server and view the application at http://localhost:3000.