Front-End Developer Internship Assessment



**PROJECT TITLE:** Restaurant Management System

**Applicant Name:** Sameer Parajuli

**Emali:** [Parajulisameer2061@gmail.com](mailto:Parajulisameer2061@gmail.com)

**Assignment Due Date:** 10th December 2024

**Assignment Submission Date:** 10th December 2024

**Word Count:** 1092

**Table of Contents**

[1. Introduction 3](#_Toc184659506)

[2. Technologies Used 3](#_Toc184659507)

[3. Features: 4](#_Toc184659508)

[4. Components: 5](#_Toc184659509)

[5. How the system works 8](#_Toc184659510)

[6. Visual Overview 9](#_Toc184659511)

[7. Challenges 11](#_Toc184659512)

[8. Error-handling 11](#_Toc184659513)

[9. Conclusion 12](#_Toc184659514)

# 

# **Introduction**

The project involves designing a multi-step sign-up process that includes an OTP verification page and a form page for collecting additional details. A dynamic progress bar tracks the user's progress throughout the process in the form page. Key features of the project include conditional react routing, a functional progress bar, working buttons and drag-and-drop functionality for enhanced user interaction

# **Technologies Used**

* **React.js**: A JavaScript library for building user interfaces, used for creating dynamic, component-based UIs.
* **React Router**: A library for handling routing in React applications. It allows navigation between different components based on URL paths.
* **Tailwind CSS**: A utility-first CSS framework that allows for easy and customizable styling.
* **JavaScript**: The primary programming language used for logic implementation.

# **Features:**

* **Multi-Step Registration**:
  + Users can progress through different steps in the registration process.
  + A progress bar visually indicates the user's progress.
  + Clear "Next" and "Previous" buttons are provided for easy step navigation, ensuring users can effortlessly move forward or back.
* **Dynamic Forms**:
  + The form fields are dynamically rendered based on constants (i.e., formFields), making it easy to manage and add new fields.
* **Drag-and-Drop File Upload**:
  + Users can upload files by dragging and dropping them.
* **OTP Verification**:
  + Users need to input a 6-digit OTP to verify before moving forward to registration from.
* **Navigation Using React Router DOM:**
* The application integrates seamless navigation powered by **React Router DOM**, allowing users to switch between different pages dynamically.
* **Responsive Design**:
  + The application is designed to be responsive, for browsers.

# **Components:**

1. **App**

* **Description**: Acts as the root component that routes the application. In this, routing is possible between the Signup page and SteeperLayout by using React Router.
* **Features:**
  + Routing is done using BrowserRouter.
  + Route has been defined for, the Signup page, and SteeperLayout which is a multi-step registration process.

1. **Signup**

* **Description:** This page is the main page that has OTP verification field with cardlayout rendered.
* **Features:**
  + The list items of different registration options.
  + Renders OTP verification form for verification.

1. **StreeperLayout**

* **Description:** It is a multi-step registration layout that includes a progress bar, with buttons to proceed to the next step or go back to any previous step.
* **Features:**
  + A dynamic progress bar that automatically resizes itself according to the user's step completion.
  + It allows going forward or backward among the various steps of registration by clicking 'Next' and 'Previous' buttons.
  + It displays the content of each step by measuring the ActiveComponent function.
  + It permits the user to traverse through a series of predefined steps stored in registration\_steps.

1. **Drag-drop**

* **Description:** The FormLayout component is a reusable form rendering utility designed to dynamically display form fields based on the input configuration.
* **Features:**
  + Layout of the drag and drop section on form
  + Functional drag and drop function.

1. **FormLayout**

* **Description:** Layout of the from
* **Features:**
  + Conditionally renders fields based on the type property of the field object.
  + Dynamically populates options for dropdown fields based on the options array provided in the field object.

1. **BusinessForm**

* **Description:** This is the business details form, which has been added to a multi-step process.
* **Features:**
  + Dynamically renders the form fields along with the drag and drop.

1. **Cardlayout**

* **Description:** Organizes cards in a grid and includes business-related registration data.
* **Features:**
* Props like ImageURL, Title, Location can be taken.
* A card developed by it will contain a stylized layout within a gradient background.

1. **OtpLayout**

* **Description:** It provides an OTP input field for validation.
* **Features:**
  + 6-digit OTP integration; focus transitions automatically between inputs.
  + Ensure that only numeric input allows.
  + Transitions to another page after clicking verify button.

1. **Data Import and Configuration (index.js)**

* **Description:** The file is used to manage and export essential data and assets for the application. It imports images and configurations and then exports them as constants that are used across various components. This modular approach allows for the reuse of data and assets in multiple places throughout the app, ensuring consistency and maintainability.
* **Features:**
  + The file contains dynamic data structures for cards and form fields, allowing easy updates and additions without altering individual components.
  + The data is stored in arrays (such as cards, formFields, and registration\_steps), enabling dynamic rendering of content across various components.
  + All key assets and data are imported and exported from this file, providing a single source ford data used throughout the application.

# **How the system works**

1. **OTP Verification:**

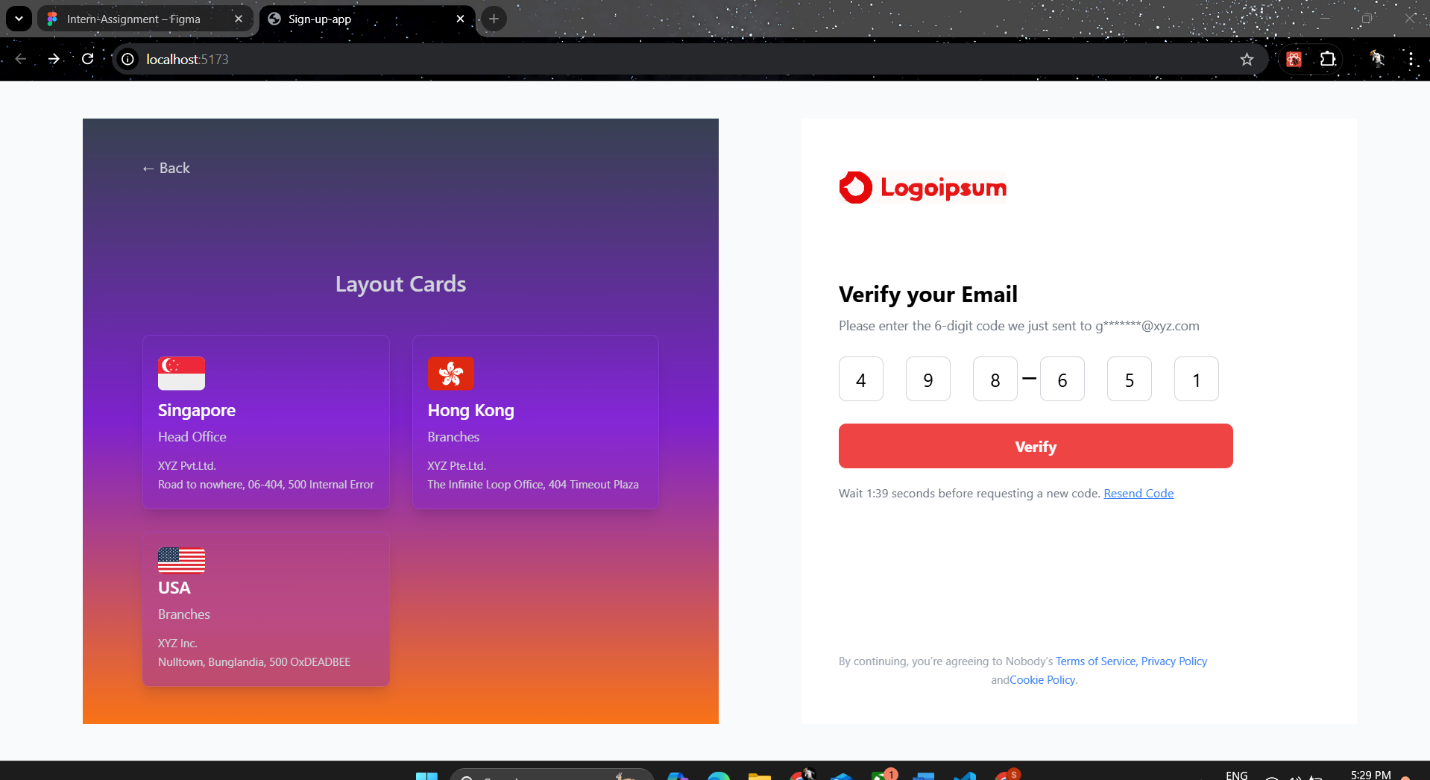
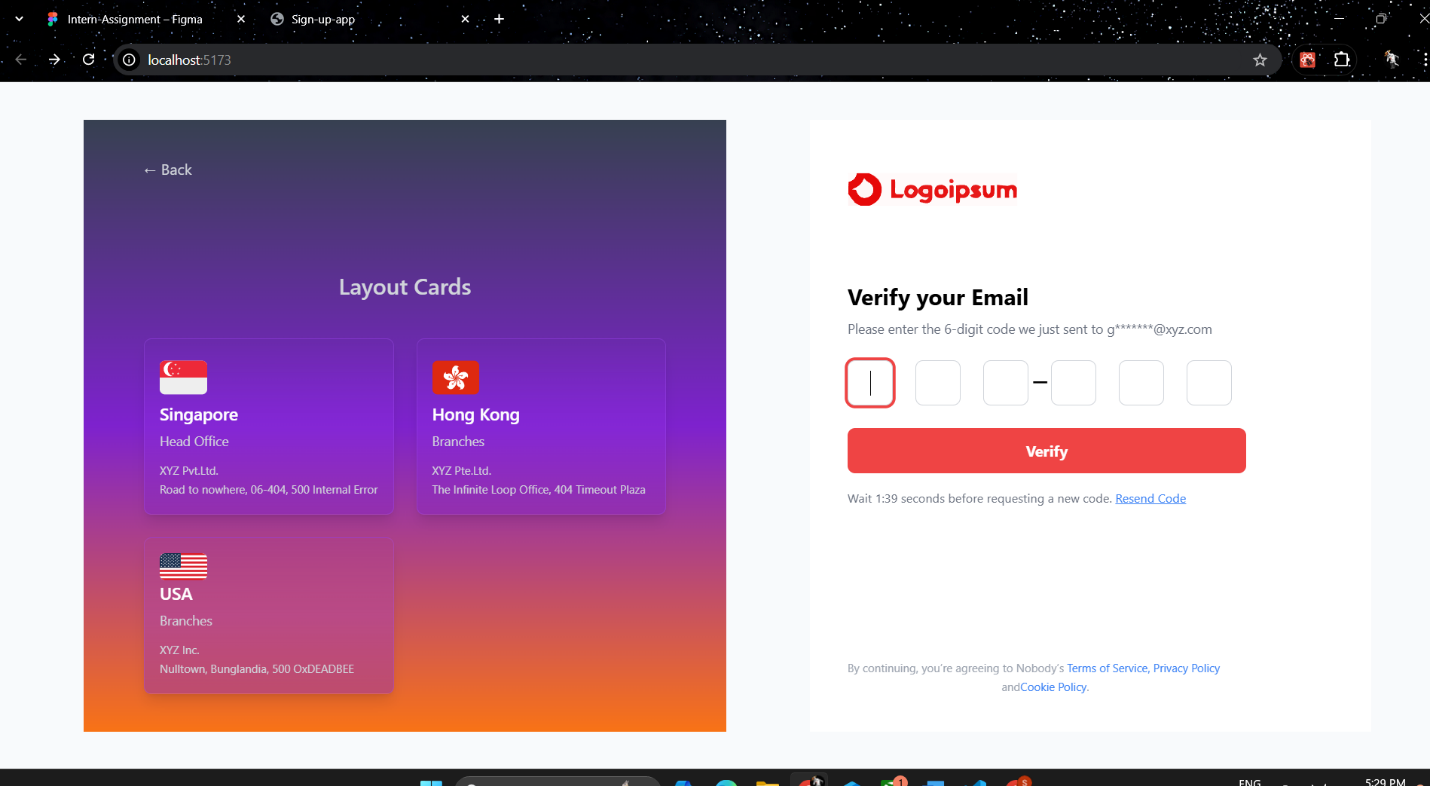
* The user starts by entering a 6-digit OTP (One-Time Password) in the provided OTP field.
* Once the OTP is entered, the user clicks the "Verify" button.
* Upon successful OTP verification, the user is redirected to the next page containing the multi-step registration form.

1. **Multi-Step Registration Form:**

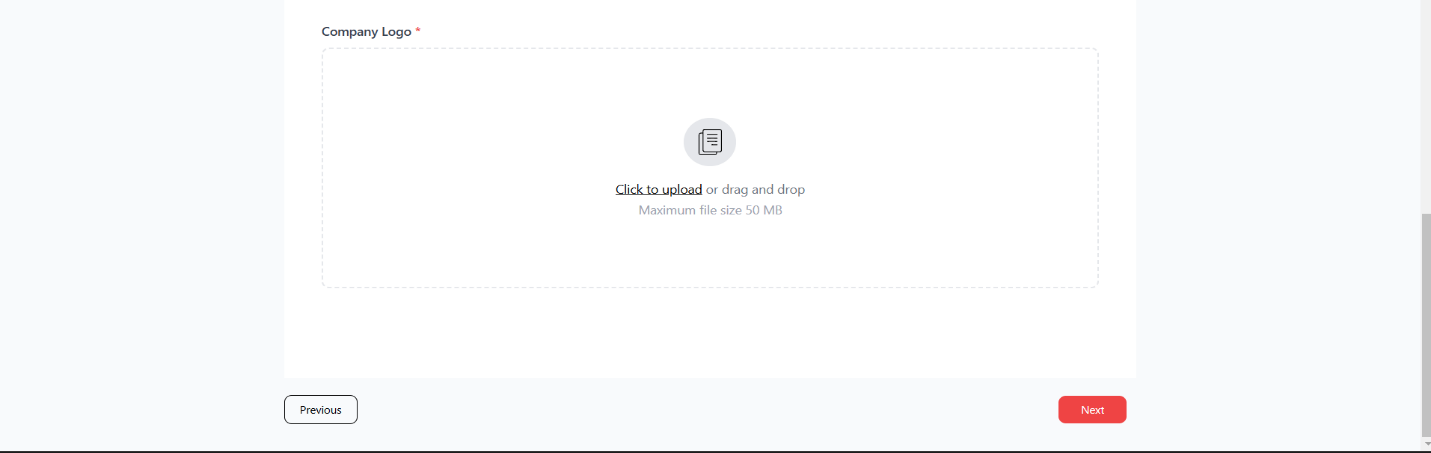
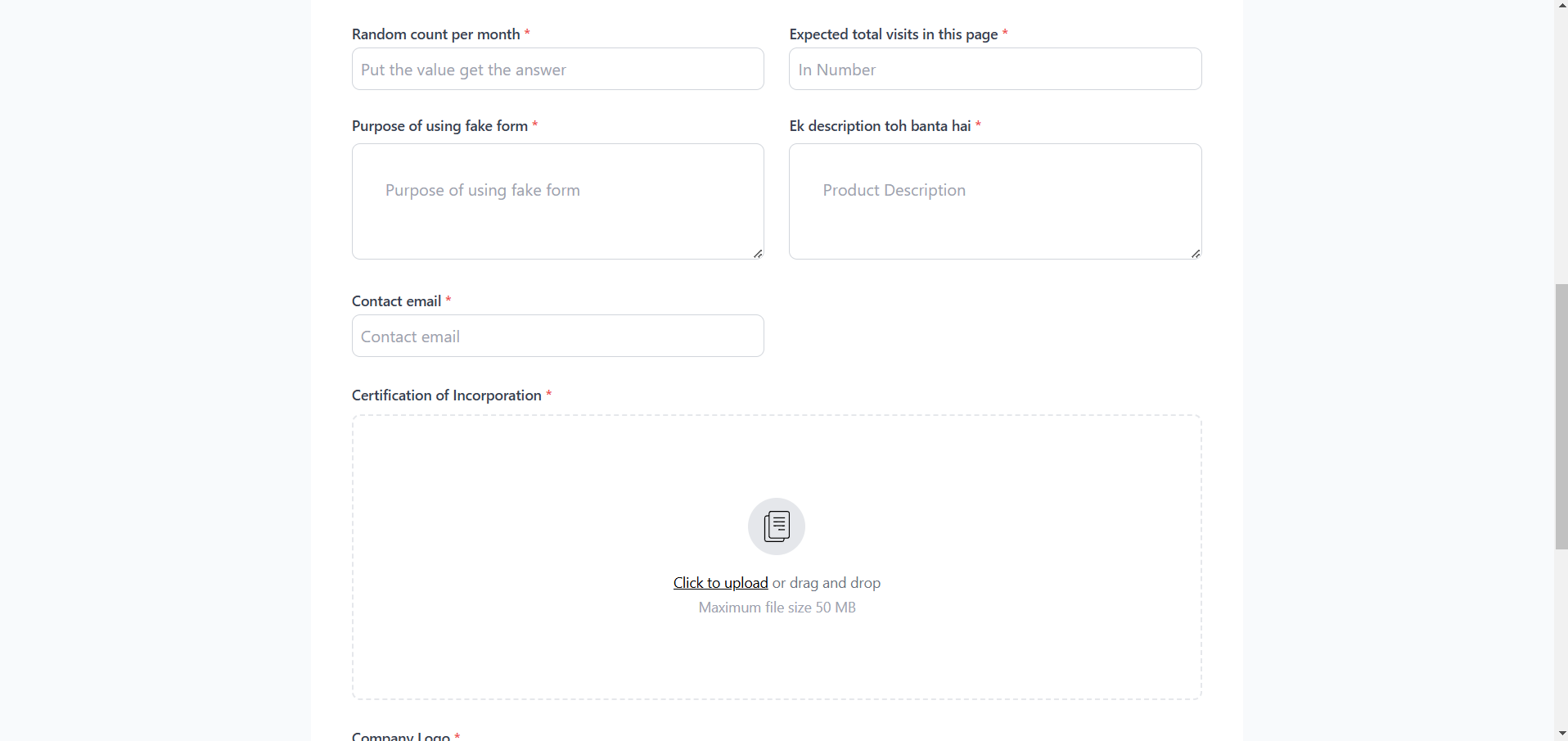
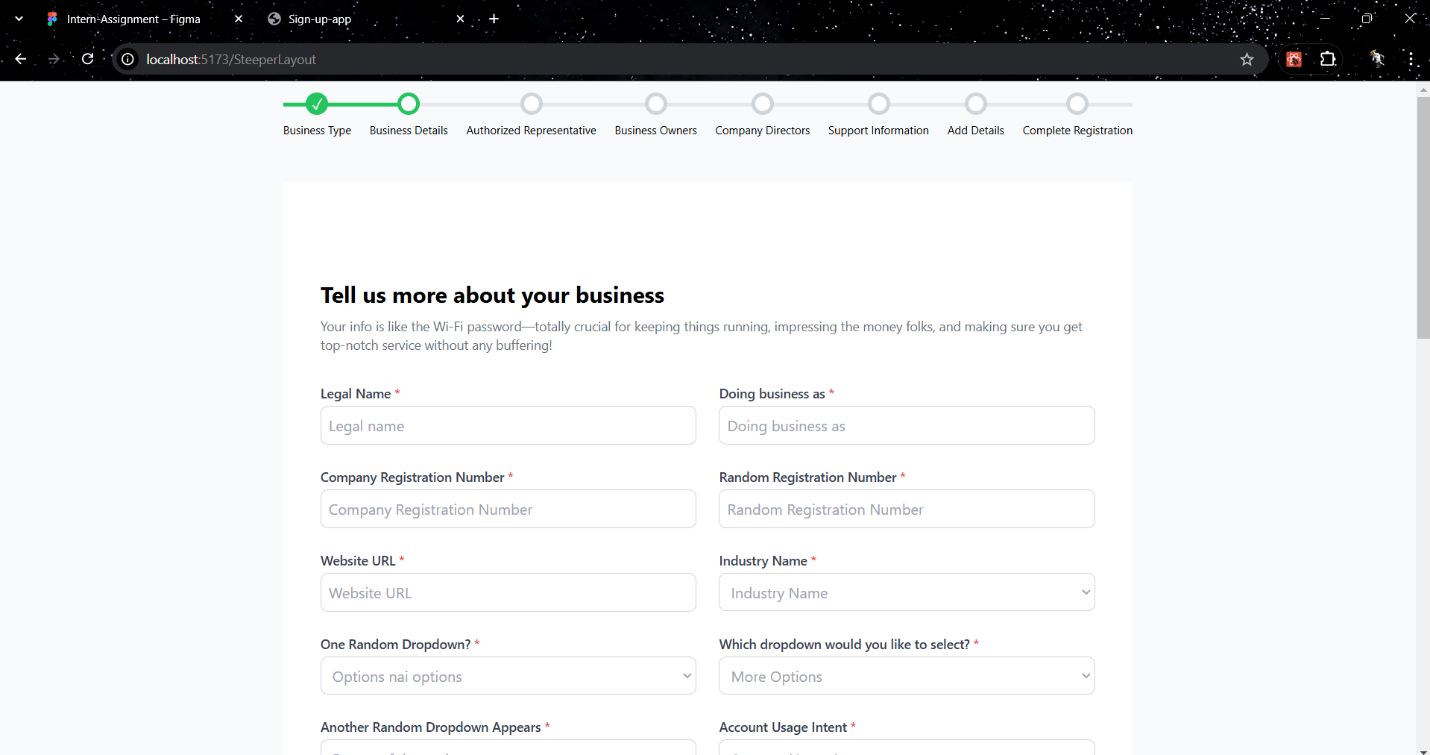
* After OTP verification, the user is presented with a multi-step form.
* This form is divided into several steps, each with specific fields for input.
* A progress bar at the top of the page visually tracks the user’s progress through each step.
* Users can navigate through the steps by clicking "Next" or "Previous" buttons, allowing them to review and update their information as needed.
* Each step is dynamically rendered based on the user’s current progress.

.

# **Visual Overview**



**Figure 1: OTP page before and after entering OTP**



**Figure 2: Multi-Step From after clicking verify**

# **Challenges**

During the development of the project, several challenges were encountered and successfully addressed. The main challenges include:

* **Card Layout Alignment:** Ensuring that the card layout matched the provided UI/UX design was a complex task. The card layout was not replicated as in the provided design.
* **Implementing a Dynamic Progress Bar:** Developing a functional and visually appealing progress bar was challenging. The progress bar had to dynamically update as users moved through various steps of the multi-step form while ensuring it was responsive and accurately represented the user's progress.
* **Building a Functional Drag-and-Drop Feature:** Integrating a smooth and reliable drag-and-drop functionality presented its own set of challenges.

# **Error-handling**

**Empty OTP Field: If the user attempts to proceed without filling in the OTP field, an alert message is displayed, prompting them to enter the OTP. The form will not proceed until the field is properly filled.**

# **Conclusion**

In conclusion, this project successfully implements multistep sign-up processes with OTP verification, dynamic form rendering, and user-friendly features like a progress bar and drag-and-drop functionality. Implementing modern technologies like React.js, React Router, and Tailwind CSS in a manner that ensures responsiveness, efficiency, and a visually appealing layout will allow for the best user experience.