

Foodie Application

<https://foodie-app-silk.vercel.app/>

https://github.com/vijenderchimma/Foodie_App.git

Table of Contents

| | |
|--------------------------|---|
| 1. UserLandingPage | 2 |
| Approach | 2 |
| 2. Navbar | 3 |
| Approach | 3 |
| 3. ItemsDisplay..... | 4 |

| | |
|--------------------------|---|
| Approach | 4 |
| 4. Chains..... | 4 |
| Approach | 4 |
| 5. FirmCollections | 5 |
| Approach | 5 |
| Data Flow..... | 5 |
| Error Handling | 6 |
| User Interactions | 6 |
| API Integration | 6 |
| Styling..... | 6 |
| Dependencies | 6 |

This document outlines the approach taken to develop the UserLandingPage and its associated sub-components (Navbar, ItemsDisplay, Chains, FirmCollections) in a foodie application. It emphasizes personal insights, the creative process, and the reasoning behind various decisions made during development.

1. UserLandingPage

Approach

Objective: Create an engaging and informative landing page that showcases various food items, restaurant chains, and collections available for users.

Steps:

1. **Component Integration:** Plan the layout to integrate Navbar, ItemsDisplay, Chains, and FirmCollections.
2. **Responsive Design:** Ensure the landing page is responsive and user-friendly across different devices.
3. **User Experience:** Focus on a smooth and intuitive user experience by arranging the components in a logical order.

Personal Insights:

- The landing page is the first impression for users, so it should be visually appealing and easy to navigate.
- Prioritize sections based on what users are likely most interested in: new items, popular chains, and curated collections.

2. Navbar

Approach

Objective: Provide users with easy navigation, search functionality, and quick access to authentication options.

Steps:

4. **Basic Layout:** Include a logo, search bar, and login/signup links.
5. **Styling:** Use a clean and minimalistic design to ensure the navbar is not intrusive but easily accessible.
6. **Search Functionality:** Implement a functional search bar to allow users to quickly find items.

Personal Insights:

- A well-designed navbar is crucial for user retention as it facilitates easy navigation.
- Keeping the design simple helps in maintaining focus on the main content of the page.

3. ItemsDisplay

Approach

Objective: Display a collection of food items attractively.

Steps:

7. **State Management:** Use useState to manage the list of items to be displayed.
8. **Mapping Items:** Map through the items array to dynamically generate the item list.
9. **Styling:** Ensure each item is presented with an image and name for visual appeal.

Personal Insights:

- Visual representation of food items is key in a foodie application to entice users.
- Use high-quality images and maintain consistency in presentation.

4. Chains

Approach

Objective: Showcase popular restaurant chains with an interactive and dynamic component.

Steps:

10. **API Integration:** Fetch data from the backend API to get the list of restaurant chains.
11. **Loading State:** Implement a loading state to handle data fetching delays.
12. **Scrollable List:** Enable horizontal scrolling to allow users to browse through different chains.

Personal Insights:

- Popular chains attract users, so highlighting them can drive user engagement.
- Interactivity, such as horizontal scrolling, makes the component more engaging.

5. FirmCollections

Approach

Objective: Display collections of restaurants that offer online food delivery.

Steps:

13. **API Integration:** Fetch data from the backend API to get the list of firms.
14. **Filtering:** Implement filtering options to allow users to view firms by cuisine or region.
15. **Linking:** Provide links to detailed views of each firm.

Personal Insights:

- Providing filters enhances the user experience by allowing them to quickly find what they are interested in.
- Including offers and detailed information can incentivize users to explore more.

Data Flow

16. **UserLandingPage:**

- Acts as a container for all sub-components, providing the overall layout and structure.

17. **Navbar:**

- Facilitates navigation and search, ensuring users can quickly find what they need.

18. **ItemsDisplay:**

- Presents a dynamic list of food items, updated based on the state.

19. **Chains:**

- Fetches and displays data from the API, with a focus on popular restaurant chains.

20. **FirmCollections:**

- Fetches and displays data from the API, offering filtering options to refine the display based on user preferences.

Error Handling

- **Chains:** Displays an alert if data fetching fails, ensuring users are informed of the issue.
- **FirmCollections:** Displays an alert if data fetching fails, maintaining transparency with users.

User Interactions

- **Navbar:** Users can navigate to different parts of the application, search for items, and access authentication.
- **ItemsDisplay:** Users can view detailed information about each item.
- **Chains:** Users can scroll through a list of restaurant chains, enhancing engagement.
- **FirmCollections:** Users can filter and explore different restaurant collections, leading to a personalized experience.

API Integration

- **Chains:** Integrates with the backend API to fetch vendor data.
- **FirmCollections:** Integrates with the backend API to fetch firm data.

Styling

Each component has specific CSS classes that ensure a consistent and visually appealing design. The styling focuses on clarity, responsiveness, and user engagement.

Dependencies

- **React Router:** Used for navigation (Link component).

- **Axios:** Used for making API requests.
- **React Icons:** Used for displaying icons.
- **React Loader Spinner:** Used for displaying loading indicators.

This detailed approach document outlines the thought process, creative insights, and steps taken to develop a comprehensive landing page and associated components for the foodie application.