Hackathon 3

Day-4

Objective:

Designing and creating dynamic frontend elements for the FoodTuck marketplace was the main emphasis of day four. The objective was to use Sanity CMS or APIs to dynamically retrieve and display data while making sure that all of the components were responsive, reusable, and modular.

Key Learning Outcomes:

1. Dynamic frontend components were created to retrieve and present marketplace data.
2. Components were designed to be modular and reusable for scalability.
3. A better comprehension of state management strategies.
4. Improved user experience through the application of responsive design concepts.
5. Modeled real-world development processes by adhering to professional workflows.

Components Developed:

1. <u>Product Listing Component</u>

- **Functionality:** Displayed a grid of products dynamically fetched from the backend.
- **Fields Included:** Product Name, Price and Image.
- **Example:** Grid layout with product cards.

2. <u>Product Detail Component</u>

- **Functionality:** Created detailed pages for each product using dynamic routing.
- **Fields Included:** Description, Price, Available .

3. <u>Category Component</u>

- **Functionality:** Dynamically displayed product categories and enabled filtering.

4. <u>Search Bar</u>

- **Functionality:** Allowed users to search for products by name or id.

5. **Cart Component**

- **Functionality:** Displayed selected items, quantities, and total price.
- **State Management:** Tracked cart items using React Context.

6. **Filter Panel Component**

- **Functionality:** Allowed users to filter products by Category.

7. **Responsive Header and Footer**

- **Functionality:** Included navigation links, branding, and key contact details.
- **Responsiveness:** Adjusted design dynamically for desktop, tablet, and mobile views.

Frontend Best Practices Followed:

1. **Reusable Components**
   - Designed modular components like `ProductCard` and `CategoryFilter` to be used across pages.
   - Passed data via props for flexibility.
2. **State Management**
   - Used React state and context for efficient data management.
3. **Styling**
   - Implemented responsive designs using Tailwind CSS.
   - Ensured consistency across desktop and mobile views.
4. **Performance Optimization**
   - Lazy-loaded images and assets to improve load time.
   - Implemented pagination for large datasets.

Challenges Faced:

1. **Dynamic Routing Issues**
   - Initial challenges with setting up product detail pages using Next.js dynamic routing.
   - **Solution:** Reviewed and corrected file structure to enable correct routing.
2. **API Data Fetching**
   - Encountered delays in rendering data due to API response time.
   - **Solution:** Used loading states to improve user experience.
3. **Responsive Design Adjustments**
   - Ensuring the UI adapted perfectly on smaller screens like the iPhone 12.
   - **Solution:** Adjusted breakpoints in Tailwind CSS for seamless responsiveness.

Next Steps:

1. Implement advanced features like wishlist and order tracking.
2. Improve performance by optimizing the search bar's functionality.
3. Check for cross-browser compatibility with components.
4. Keep a record of every step and difficulty for future use.

Conclusion:

The development of a dynamic and polished interface for the FoodTuck marketplace made great strides on day four. Now that the components are responsive, scalable, and modular, a solid basis for future advancement and practical implementation has been established.