SIMPLELEARN Full Stack Developer - MERN Stack Masters Program

Course – 2 Build a strong MERN Foundation

Assessment – Writeup

Student Name : Vedaang Sharma

Online Food Ordering Application – Project Documentation

1. Project Overview

The **Online Food Ordering Application** is a web-based system designed to simulate a food delivery platform similar to Swiggy. It enables users to browse food items, filter them by category, add items to a cart, and complete a checkout process. The application provides a responsive and user-friendly interface, making it suitable for desktop and mobile devices.

Purpose:

- Enable restaurants to showcase their menu online.
- Allow users to place orders conveniently.
- Provide an interactive and visually appealing interface for an enhanced user experience.

---

2. Features

User Authentication

- **Registration:** Users can register with name, email, and password.
- **Login:** Users can log in using registered credentials.
- **Logout:** Users can securely log out at any time.

Food Browsing

- **Food Categories:** Menu items are categorized into **Indian**, **Fast Food**, and **Beverages**.
- **Filter Functionality:** Users can filter food items by category using filter buttons.

Cart Functionality

- **Add to Cart:** Users can add items to the cart.

- **Item Count:** Each item shows how many times it has been added to the cart.
- **Cart Overview:** Users can view all added items, remove items, and see the total count.
- **Cart Animation:** Adding items triggers a visual animation on the cart icon.

Checkout Flow

- **Step 1 – Delivery Details:** Users enter address and contact number.
- **Step 2 – Payment:** Users enter credit card details including card number, expiry date, and CVV. Input fields are formatted for usability (e.g., card number grouped every 4 digits, expiry as MM/YY).
- **Step 3 – Confirmation:** After successful checkout, users receive an order confirmation message.

Responsive Design

- Uses **Bootstrap** to ensure proper layout on various devices.
- Cards, buttons, and input fields are styled professionally.
- Sticky navbar remains visible while scrolling.

---

3. Implementation Details

Frontend Technologies

- **HTML5:** Structure of the web pages and forms.
- **CSS3 & Bootstrap 5:** Styling, layout, responsive design, card components, buttons, navbar, badges, and spacing.
- **JavaScript (ES6):** Dynamic functionality including:
  - Cart management
  - Item-specific count updates
  - Filter functionality
  - Checkout flow (step navigation)
  - Input formatting for card and expiry fields
  - DOM manipulation and animations

Key JavaScript Components

1. **Data Structures**

```
let users = [];
let currentUser = null;
let cart = [];
let itemCounts = {};
```

2. **Add to Cart Functionality**

- Adds an item object {name, price} to cart.
- Updates global cart count and item-specific count displayed on buttons.

```
function addToCart(name, price) {
  // Add to cart array
  cart.push({ name, price });

  // Update global cart count
  updateCartCount();
  animateCart();

  // Update item-specific count
  if (!itemCounts[name]) itemCounts[name] = 0;
  itemCounts[name]++;
  const countSpan = document.getElementById(`count-${name}`);
  if (countSpan) countSpan.innerText = itemCounts[name];

  console.log(cart); // debug
}
```

3. **Cart Rendering**

- Displays all items in the cart with Remove buttons.
- Updates counts dynamically when items are removed.

4. **Filtering Food Items**

- Static HTML cards use data-category attributes.
- Filter buttons show/hide cards based on category:

```
function filterItems(category) {
  // Select all food cards
  const items = document.querySelectorAll(".food-item");

  items.forEach(item => {
    if (category === "all") {
      item.classList.remove("d-none"); // show all
    } else {
      if (item.dataset.category === category) {
        item.classList.remove("d-none"); // show matching category
      } else {
        item.classList.add("d-none"); // hide non-matching
      }
    }
  });
}
```

5. **Checkout Flow**

- Multi-step form:
    - Step 1: Address & contact
    - Step 2: Payment (formatted card input)
    - Step 3: Order confirmation
- Only one step visible at a time using classList.add("d-none") / remove("d-none").

6. **Input Formatting**

- Credit card number grouped every 4 digits.
- Expiry date formatted as MM/YY.

7. **Animations**

- Cart icon has a “bump” animation when an item is added.

---

4. Workflow / Working of the Application

1. **User Registration/Login**
    - New users register and then log in.
    - Returning users log in directly.
2. **Browsing Food Items**
    - Users see all food items categorized by type.

    - Filters allow displaying specific categories only.

3. **Adding Items to Cart**
    - Clicking "Add to Cart" adds the item to the cart.
    - The button displays the count for that item.
    - Cart icon updates total item count and shows an animation.
4. **Viewing Cart**
    - Users can view all selected items.
    - Items can be removed individually.
5. **Checkout**
    - Step 1: Enter delivery details.
    - Step 2: Enter payment information.
    - Step 3: Order confirmation displayed.
    - After confirmation, cart is cleared.
6. **Logout**
    - Users can log out and return to the login/registration page.

---

5. Features Summary

Feature	**Description**
User Registration & Login	Secure authentication and account management
Responsive Food Menu	Shows items categorized with filters
Dynamic Cart Functionality	Add/remove items, item count, animations
Checkout Flow	Multi-step form: address → payment → confirmation
Input Formatting	Card number and expiry date formatted automatically
Sticky Navbar	Always visible for quick access to cart and filters
Professional UI	Bootstrap cards, buttons, badges, and layout

---

6. Project Deliverables

I am submitting a .zip file of the source code developed by me in the project. The project is also hosted on my GitHub profile and deployed for live viewing through a hosting platform.

Links :

- GitHub Repository : https://github.com/gtathelegend/food-ordering-app
- Live Project : https://food-ordering-app-50hd.onrender.com
- Google Drive Link : https://drive.google.com/drive/folders/1-CJhDHp_4pP1rm49YbUq4nTS4YZb7H8c?usp=sharing

---

7. Conclusion

This project demonstrates how to build a **dynamic and responsive online food ordering system** using HTML, CSS, Bootstrap, and JavaScript. It covers core concepts such as:

- DOM manipulation
- Event handling
- Dynamic updates to UI
- Multi-step forms
- Input formatting
- Cart management

This project simulates real-world food delivery applications and can be further extended with backend integration, database storage, and payment gateway integration for production use.