**BQuick**

**A PROJECT REPORT**

**for**

**Mini Project (KCA353)**

**Session (2024-25)**

**Submitted by**

**Sneha Rani**

**( 2300290140184 )**

**Sunny Kushwaha**

**(2300290140189 )**

**Tanu**

**( 2300290140191 )**

**Upasana Chaudhary**

**( 2300290140197 )**

**Submitted in partial fulfilmentof the**

**Requirementsfor the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Dr. Amit Kumar**

### Assistant Professor



**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(December 2024)**

**CERTIFICATE**

Certified that **Sneha Rani (2300290140184), Sunny Kushwaha (2300290140189), Tanu (2300290140191), Upasana Chaudhary (2300290140197)**has/ have carried out the project work having “**BQuick**” (**Mini-Project-KCA353**) for **Master of Computer Application**fromDr.A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU),Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Dr. Amit Kumar Dr. Arun Tripathi**

**Assistant Professor Head**

**Department of Computer Applications Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad KIET Group of Institutions, Ghaziabad**

**BQuick-Team**

**ABSTRACT**

**“**BQuick**”**is an application designed to streamline restaurant operations during peak hours, enhancing the dine-in experience for both customers and staff. This solution eliminates the need for waiters to manually take orders, reducing service delays and errors.

During peak hours, restaurants often face challenges in managing customer service effectively. It becomes difficult for a single waiter to attend to every visiting customer promptly, leading to long wait times. Additionally, some customers take an extended time to decide on their orders, further delaying the overall service workflow. This can negatively impact the dining experience and operational efficiency.

To address these challenges, this project introduces a MERN stack-based application that streamlines the ordering process, enhancing both customer satisfaction and staff productivity. The application allows customers to place their orders directly from their tables by scanning a QR code. Once an order is placed, it is confirmed and displayed on a kitchen screen, enabling chefs to view and prepare the orders without delay.

Customers can conveniently place their orders by scanning a QR code on their table using their smartphones. Once the order is placed, it is confirmed and seamlessly transmitted to a kitchen display system, where chefs can view the details and prepare the dishes accordingly. This workflow not only speeds up the ordering process but also ensures accurate communication between customers and the kitchen.

This approach eliminates unnecessary delays, reduces the workload on waitstaff, and ensures efficient communication between customers and the kitchen. By streamlining the ordering process, the solution enhances customer satisfaction, improves service speed, and provides a more enjoyable and seamless dining experience, particularly during busy hours.

**ACKNOWLEDGEMENTS**

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Dr. Amit Kumar** for his/ her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work wouldnot have been possible in time. They keep my life filled with enjoyment and happiness.

**Sneha Rani**

**Sunny Kushwaha**

**Tanu**

**Upasana Chaudhary**

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Certificate | | ii |
|  | Abstract | | iii |
|  | Acknowledgements | | iv |
|  | Table of Contents | |  |
|  |  | |  |
|  |  | |  |
|  |  | | ix |
| 1 | Introduction | |  |
|  | 1.1 | Overview | 6 |
|  | 1.2 | Motivation | 6-7 |
|  | 1.3 | Problem Statement | 7-8 |
|  | 1.4 | Expected Outcome | 8 |
| 2 | Literature Survey | | 9 |
| 3 | Proposed Work | |  |
|  | 3.1 Technology Description | | 10 |
|  | 3.2 Approach Used | | 10-11 |
|  | 3.3 Modules Required | | 12-13 |
|  | 3.4 Algorithms | | 13-14 |
| 4 | Discussions | |  |
|  | 4.1 Performance | | 15-16 |
|  | 4.2 Limitations of the system | | 16 |
|  | 4.3 Future Research Direction | | 16-17 |
| 5 | Screenshots | | 18-29 |
| 6 | codes | | 30-31 |
| 7 | conclution | | 32 |
| 8 | Reference | | 33 |
| 8 | Bibliography | | 33 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 OVERVIEW**

“BQuick” is an innovative solution designed to streamline customer service during peak hours, ensuring a seamless dining experience. This project addresses the common challenges faced by restaurants, where few single waiters struggle to attend to multiple tables promptly, leading to long wait times and service inefficiencies.

The primary goal of the project is to enhance the ordering process by empowering customers to place orders directly from their tables using a QR code. By eliminating the need for manual order-taking, this solution minimizes delays, reduces staff workload, and ensures faster communication between customers and the kitchen.

The project unfolds in three main phases: conceptualization, development, and execution. During the conceptualization phase, the team identified the key pain points in traditional restaurant workflows and envisioned a customer-centric solution to address them. In the development phase, the team designed a user-friendly interface for customers to view the menu, select their items, and place orders seamlessly. A backend system was developed to transmit confirmed orders directly to a kitchen display, enabling chefs to view and prepare them promptly. Finally, in the execution phase, the system underwent rigorous testing to ensure smooth operation during high customer volumes, addressing potential issues and optimizing the solution for real-world application.

The project's report outlines the problem statement, solution design, key features, challenges encountered, and how the system effectively addresses them. By streamlining the order management process, this system has the potential to transform the dining experience, offering significant advantages in service efficiency, customer satisfaction, and operational management during busy restaurant hours.

**1.2 MOTIVATION**

The inspiration behind “BQuick” comes from a desire to enhance the dining experience and address challenges restaurants face during peak hours. Long wait times and overburdened staff often lead to customer dissatisfaction. This project aims to transform order management by enabling customers to place orders directly from their tables via QR codes, reducing delays and improving communication with the kitchen.

Our motivation lies in creating a seamless dining experience that benefits both customers and staff. By integrating technology into traditional workflows, the system simplifies operations, enhances efficiency, and ensures faster service during busy hours.

We believe this solution has the potential to make a lasting impact on the restaurant industry, showcasing how innovation can solve real-world problems and inspire similar advancements in hospitality management.

## **1.3 PROBLEM STATEMENT**

Restaurants often face significant challenges in maintaining efficient operations and providing a seamless customer experience during peak hours. These issues are particularly pronounced when a limited number of waitstaff are tasked with attending to a large volume of customers, leading to delays, errors, and dissatisfaction.

One major issue arises from the dependency on waitstaff to take orders manually. During busy hours, a single waiter managing multiple tables struggles to attend to all customers promptly. This often results in long wait times for customers, who may have to wait for their turn to place an order. Such delays not only impact customer satisfaction but also reduce table turnover, affecting the restaurant’s revenue potential.

Another challenge is the unpredictability of customer behavior. Some customers take considerable time to decide on their orders, further compounding delays for others. This creates a bottleneck in the workflow, slowing down the ordering process and placing additional pressure on the staff to manage expectations and maintain service quality.

In addition to these challenges, the manual nature of the ordering process increases the likelihood of errors in order communication. Miscommunication between waitstaff and the kitchen can result in incorrect or incomplete orders, leading to wasted resources, additional costs, and frustrated customers.

The traditional approach to order management also places a heavy burden on restaurant staff, who must multitask to manage tables, take orders, and ensure timely service. This can lead to burnout among staff and lower overall efficiency during peak hours.To address these challenges, a more streamlined and automated approach to order management is needed. The proposed solution introduces a digital ordering system where customers can place their orders directly from their tables using a QR code. This system eliminates the dependency on waitstaff for order-taking, significantly reducing wait times and ensuring accurate order communication.

By transmitting orders directly to a kitchen display system, the solution minimizes errors, enhances operational efficiency, and allows staff to focus on delivering better customer service. Customers benefit from a faster and more convenient dining experience, while the restaurant can handle high volumes more effectively, improving both satisfaction and profitability.Thisinnovative solution addresses the core inefficiencies in traditional restaurant workflows, providing a modern approach to enhancing customer experiences and optimizing operational management during crowded hours.

## **1.4EXPECTED OUTCOME**

* **Reduced Staff Workload:**The QR code-based ordering system eases the burden on waitstaff, especially during peak hours. By allowing customers to place orders directly, staff can focus on delivering personalized service and managing other essential tasks. This streamlined workflow minimizes errors and improves overall efficiency, enabling the restaurant to handle higher customer volumes without additional staff.
* **Enhanced Dine-In Experience:**With instant access to the digital menu through QR codes, customers can browse and order at their own pace without waiting for a waiter. The direct communication between customers and the kitchen ensures faster service and fewer errors, creating a seamless and enjoyable dining experience. This modern approach appeals to tech-savvy diners and adds a touch of convenience to traditional dining.
* **Increased Customer Satisfaction:**By reducing wait times and ensuring accurate order processing, the system enhances customer satisfaction. Diners appreciate the control and convenience of placing their orders directly, leading to a smoother experience. Satisfied customers are more likely to leave positive feedback, return for future visits, and recommend the restaurant to others, boosting the restaurant’s reputation.
* **Faster Table Turnover:**With orders being placed and processed more efficiently, the time customers spend waiting for service is significantly reduced. This results in quicker meal preparation and faster table turnover, enabling the restaurant to accommodate more guests during peak hours, ultimately increasing revenue potential.
* **Cost Efficiency:**The system reduces reliance on manual processes and minimizes the need for additional staff during busy periods. By streamlining operations and reducing errors, restaurants can save on labor costs and wastage, making the solution both practical and cost-effective.
* **Improved Order Accuracy:**Manual order-taking is prone to errors due to miscommunication or misinterpretation. The digital ordering system ensures precise communication between customers and the kitchen, reducing the chances of incorrect or incomplete orders. This not only saves time but also avoids customer dissatisfaction caused by order mix-ups.

**CHAPTER 2**

**LITERATURE SURVEY**

The study titled "Technology in Enhancing Customer Service Efficiency in Hospitality" examines the growing role of technological solutions in streamlining operations within the hospitality industry. It highlights how innovations like digital ordering systems significantly reduce wait times and improve workflow efficiency. The research underscores the positive impact of such technologies on both customer satisfaction and staff productivity, demonstrating their potential to address common challenges in service-oriented environments.

Another relevant work, "Self-Service Technologies in Restaurants: A Consumer Perspective," explores the adoption of self-service technologies, such as mobile ordering and kiosks, from the customer's point of view. This study identifies critical factors that influence consumer behavior, including the convenience, ease of use, and perceived improvement in service speed. The findings emphasize that customers are more likely to embrace these solutions when they align with their expectations of a modern dining experience.

"Queue Management Systems for Service-Oriented Businesses" delves into the challenges of managing high customer volumes in service industries. The research focuses on the use of digital queue management and order processing systems to address bottlenecks and enhance overall service efficiency. It provides evidence of how such systems can transform traditional workflows, ensuring faster and more accurate service delivery.

Additionally, "Customer-Centric Innovations in Restaurant Management" investigates emerging trends in restaurant technology aimed at improving customer experiences. This study examines digital tools that empower customers to interact with menus and place orders independently, reducing the need for staff intervention while maintaining a high standard of service.

Together, these studies provide a robust foundation for understanding the benefits and practical applications of digital solutions in the restaurant industry. They validate the relevance of the proposed Restaurant Order Management System as a timely and effective innovation to address peak-hour challenges, enhance the dining experience, and increase customer satisfaction.“BQuick” is an innovative solution designed.

**CHAPTER 3**

**PROPOSED WORK**

**3.1 TECHNOLOGY DESCRIPTION**

* **QR Code Technology:** Used to allow customers to scan codes from their tables and place orders directly from their devices, reducing wait times and enhancing convenience.
* **Backend Technologies:** We used RESTful APIs and JSON to manage and transfer order data between the user interface and the kitchen display system.
* **Database:** The system leverages a lightweight, non-database approach, as the dataset is small enough to be handled efficiently without the need for a full database management system.

**3.2 APPROACH USED**

The development of “BQuick” leverages modern technological solutions to streamline restaurant operations and enhance the customer dining experience. This approach integrates QR code-based ordering and kitchen display systems, focusing on reducing wait times and improving service efficiency, particularly during peak hours.

The system begins with the customer scanning a QR code at their table to access the menu and place an order directly from their mobile device. This eliminates the need for waitstaff to take orders manually, allowing for faster and more accurate order placement. The order is instantly transmitted to the kitchen display system, which shows the order details in real time, allowing chefs to begin preparation without delay.

The integration of this technology reduces the reliance on staff, freeing them up to attend to other tasks, thereby reducing workload and increasing overall operational efficiency. As a result, wait times are minimized, improving the dine-in experience for customers. Additionally, the seamless flow of information from the customer’s device to the kitchen ensures that the orders are processed accurately and efficiently.

This system also benefits restaurants by increasing customer satisfaction. Customers are empowered to place their orders at their convenience, and the real-time updates ensure that their orders are prepared on time, leading to fewer mistakes and faster service. Furthermore, by enabling quicker table turnover, restaurants can serve more customers during busy hours, optimizing their resources and increasing revenue.

Overall, the integration of QR code ordering and kitchen display systems presents a transformative solution that enhances customer experience, reduces staff workload, and optimizes restaurant operations for improved efficiency and customer satisfaction.

**3.3 MODULES REQUIRED**

* **Customer Interaction Module:**enhances the dining experience by allowing customers to scan a QR code placed on their table to access the menu directly from their mobile devices. This eliminates the need for traditional paper menus and waiting for a server, enabling customers to browse the menu, view detailed item descriptions, and place their orders at their own pace. Customizations and special instructions can be easily added, and once the order is placed, it is sent instantly to the kitchen. By streamlining the ordering process, this module reduces wait times, improves efficiency, and allows waitstaff to focus on other tasks, ultimately leading to a faster and more satisfying dining experience.
* **Order Management Module:**efficiently handles order submissions from customers by receiving and processing the orders placed through the **Customer Interaction Module**. Once an order is submitted, this module updates its status in real-time, keeping both the customer and the staff informed about the progress. The status is updated at each stage of the order’s lifecycle, such as "pending,""in progress," or "completed," allowing for easy tracking. Additionally, the module ensures that the order details are sent to the kitchen for preparation without delay. This seamless flow of information ensures accuracy in order fulfilment, minimizes errors, and enhances communication between the front-of-house and kitchen staff, ultimately contributing to a smoother and more efficient service.
* **Kitchen Display Module:**plays a crucial role in ensuring smooth communication between the front-of-house and kitchen staff. When an order is placed through the Order Management Module, it is instantly transmitted to the kitchen display, where the kitchen staff can view detailed information about each order. This includes the items ordered, any special instructions or customizations, and the order’s status. By displaying this information in real time, the module helps the kitchen staff prepare the orders accurately and efficiently. It eliminates the need for paper order tickets, reducing the chances of miscommunication or errors. The Kitchen Display Module ensures that orders are processed correctly and timely, contributing to improved kitchen workflow and faster service for customers.
* **Payment and Billing Module:**manages the entire payment process, ensuring a smooth and secure transaction for customers. Once the customer has finished their meal and is ready to pay, this module integrates with payment gateways to facilitate a variety of payment methods, including credit/debit cards, mobile wallets, and online payment systems. The module processes the payment securely, verifying the transaction and confirming the payment status before completing the order. After the payment is successfully processed, the module generates a digital receipt that is automatically sent to the customer’s device via email or SMS. This digital receipt includes all relevant details, such as the order summary, payment amount, and transaction reference, ensuring a paperless, efficient, and environmentally friendly billing process.

**3.4 ALGORITHMS**

* **QR Code Scanning Algorithm:**designed to facilitate the seamless interaction between customers and the restaurant’s online menu. Using a QR code scanning library, the algorithm enables customers to scan the QR code placed at their table using their mobile device. Once the QR code is captured, the algorithm decodes the embedded URL, which directs the customer’s device to the restaurant’s digital menu. This process eliminates the need for physical menus, allowing customers to quickly access the menu on their smartphones. By providing a quick and efficient way to view and select items, the algorithm enhances the customer experience, reduces wait times, and streamlines the ordering process.
* **Order Processing Algorithm:**ensures the accuracy and efficiency of the order submission process. Once a customer places an order through the online menu, the algorithm first validates the order by checking for correct item selection, quantities, and any special instructions. It ensures that the order is complete and that all selections are valid, preventing errors such as invalid items or quantities. Once the order is validated, the algorithm sends the order data to the kitchen display system using API calls, providing the kitchen staff with real-time, accurate details about the order. This streamlined process helps maintain smooth communication between the front-of-house and kitchen staff, ensuring that orders are processed quickly and correctly.
* **Order Status Update Algorithm:**efficiently tracks the progress of each order as it moves through different stages, such as "pending,""in progress," and "completed." As the order advances, the algorithm updates the system in real time, ensuring that the status is always current. Additionally, it notifies customers via their devices, keeping them informed about their order's status throughout the process. These updates provide transparency and help manage customer expectations, reducing uncertainty and enhancing the overall dining experience. By automating these status updates, the algorithm ensures smooth communication between the restaurant and the customers, improving service efficiency and satisfaction.
* **Payment Processing Algorithm:**ensures secure and seamless payment transactions by integrating with third-party payment gateways. When a customer initiates payment, the algorithm securely communicates with the gateway to process the transaction, using encryption protocols to protect sensitive payment information. It validates the transaction by confirming the payment details and ensuring the transaction is successful. Once the payment is verified, the algorithm confirms receipt of the payment before finalizing the order, allowing the system to update the order status and proceed with the next steps. This process ensures a smooth and secure payment experience for customers, minimizing errors and ensuring the accuracy of financial transactions.

**CHAPTER 4**

**DISCUSSIONS**

The integration of advanced technologies played a crucial role in shaping the overall experience of the Restaurant Order Management System. The use of React and Node.js facilitated the development of a seamless, user-friendly interface and a robust backend, allowing customers to place orders efficiently and enabling restaurant staff to manage them effectively. The system's design focuses on reducing wait times, streamlining the order process, and enhancing customer satisfaction, contributing significantly to a positive dining experience. The mobile-friendly interface makes it convenient for customers to interact with the system via QR code scanning, which not only modernizes the customer experience but also ensures speed and ease in the ordering process. This integration of web technologies sets a foundation for improving operational efficiency and customer engagement.

**5.1 Performance**

* **System Efficiency and Optimization:** The Restaurant Order Management System is designed to operate smoothly across multiple devices, ensuring optimal performance for both customers and staff. React's component-based architecture offers a responsive and fast interface, while Node.js handles concurrent server requests efficiently. The system ensures real-time updates, providing both customers and restaurant staff with the most up-to-date order status. Optimization strategies, such as caching frequently accessed data and optimizing database queries, are utilized to maintain smooth performance even during high-traffic hours.
* **Cross-Device Compatibility:** The system is compatible across a range of devices, including smartphones, tablets, and desktops, ensuring customers can easily place orders regardless of their device preferences. The flexible web interface adapts to various screen sizes, offering a seamless experience across devices. The application is also optimized for different browsers, ensuring accessibility for all users.
* **Payment Gateway Integration:** Payment processing is integrated with secure third-party payment gateways, ensuring fast and safe transactions. The system supports various payment methods, offering flexibility to customers and providing secure, encrypted transaction processing. The integration ensures quick confirmation of payments and seamless order completion, further enhancing customer satisfaction.

**5.2 Limitations of the System**

* **Internet Dependency:** One of the primary limitations of the Restaurant Order Management System is its dependency on an active internet connection. While the system is designed to function efficiently in most settings, interruptions in the internet service can impact the real-time functionality, affecting order processing and updates. Offline functionality could be a potential enhancement in the future.
* **Hardware Requirements:** The system relies on devices with a minimum RAM of 4GB and modern processors to run smoothly. While the web application is designed to be lightweight, older devices with limited resources may experience performance issues. Future optimization techniques and lower resource requirements could enhance system accessibility for a broader audience.
* **Platform Accessibility:** While the system is platform-independent, it may not support legacy systems or certain outdated browsers, limiting accessibility for some users. Future improvements could involve ensuring compatibility with older devices or operating systems to expand its reach.

**5.3 Future Research Directions**

* **Offline Functionality:** Research could explore the development of offline capabilities, allowing customers to place orders and manage their data when not connected to the internet. This could enhance system reliability and usability during network outages or in locations with limited connectivity.
* **Improved Analytics and Insights:** Future iterations could incorporate advanced analytics and machine learning algorithms to better understand customer behavior and optimize order patterns. By analysing data on customer preferences, peak dining times, and popular menu items, the system could provide more personalized experiences for customers and assist restaurants in improving their operations.
* **Voice Integration:** As voice assistants become more popular, integrating voice commands for placing orders could be a valuable addition to the system, making the ordering process more hands-free and accessible.
* **Extended Payment Integration:** Research into expanding payment gateway support, including cryptocurrencies and regional payment systems, could broaden the system's global reach and customer base. Additionally, the implementation of recurring billing for repeat customers could improve customer retention.
* **Advanced AI Features:** The integration of AI to predict customer preferences and suggest menu items based on previous orders could further enhance the customer experience. This predictive ordering could help restaurants optimize inventory and reduce food waste.
* **Accessibility Features for Diverse Users:** Research into accessibility features such as speech-to-text, screen readers, and high-contrast modes could ensure the system is inclusive for users with disabilities, enhancing usability and customer satisfaction.

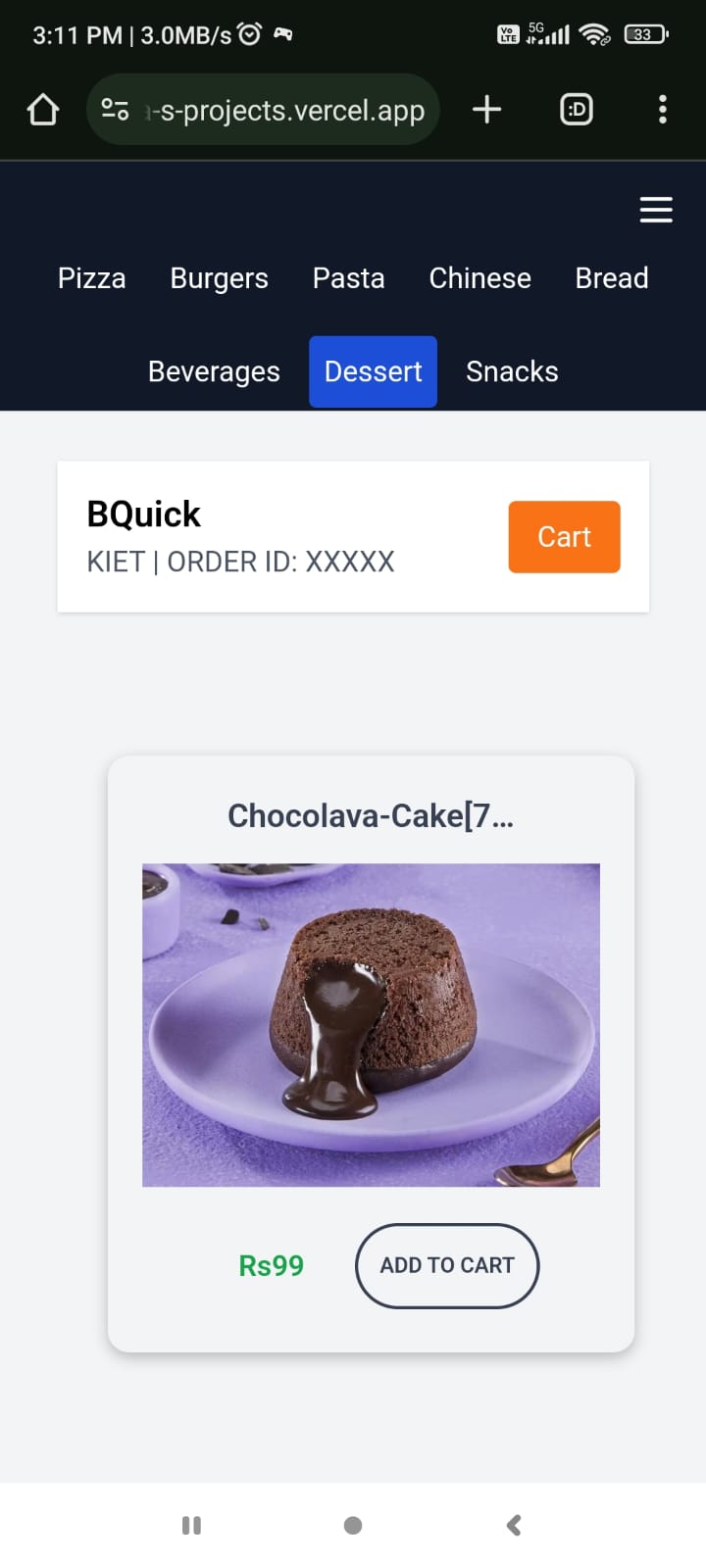
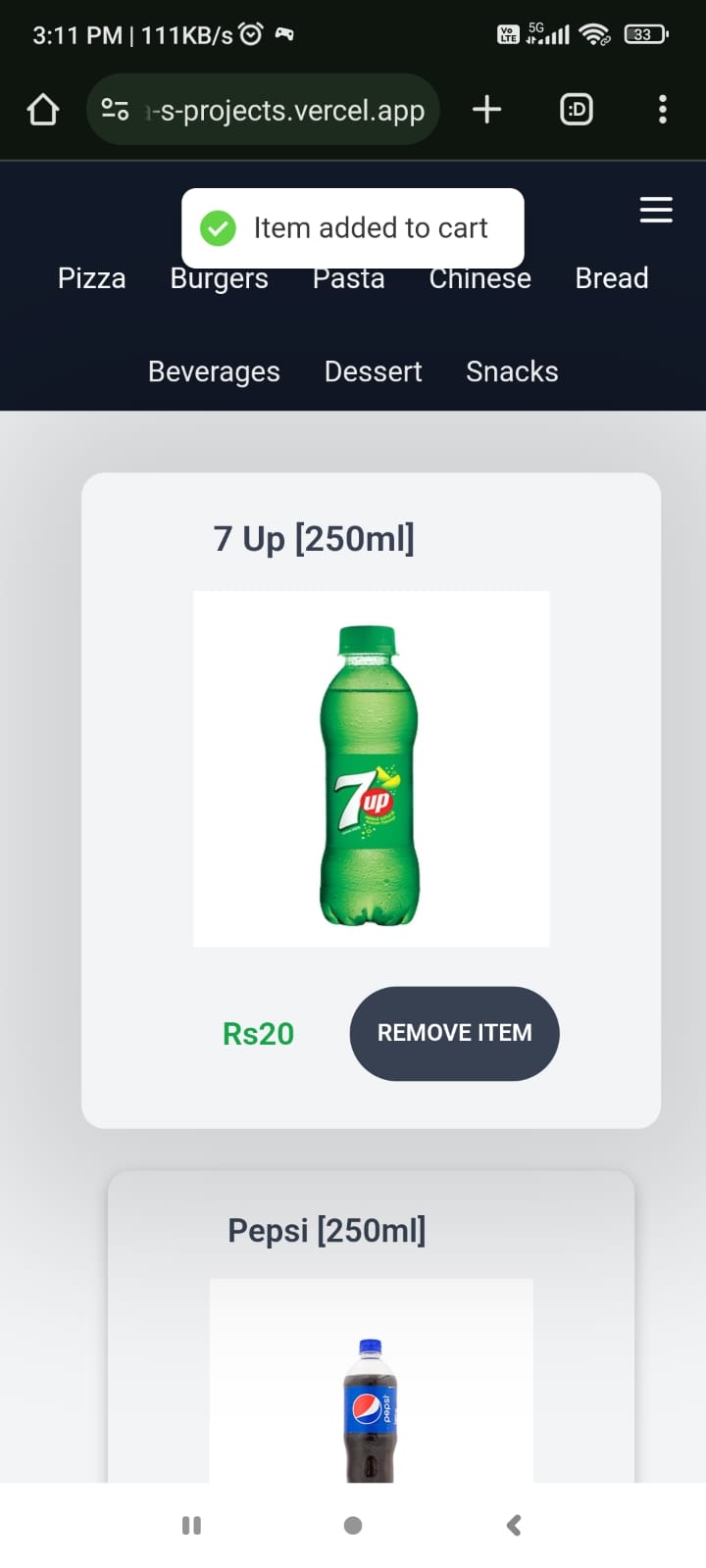
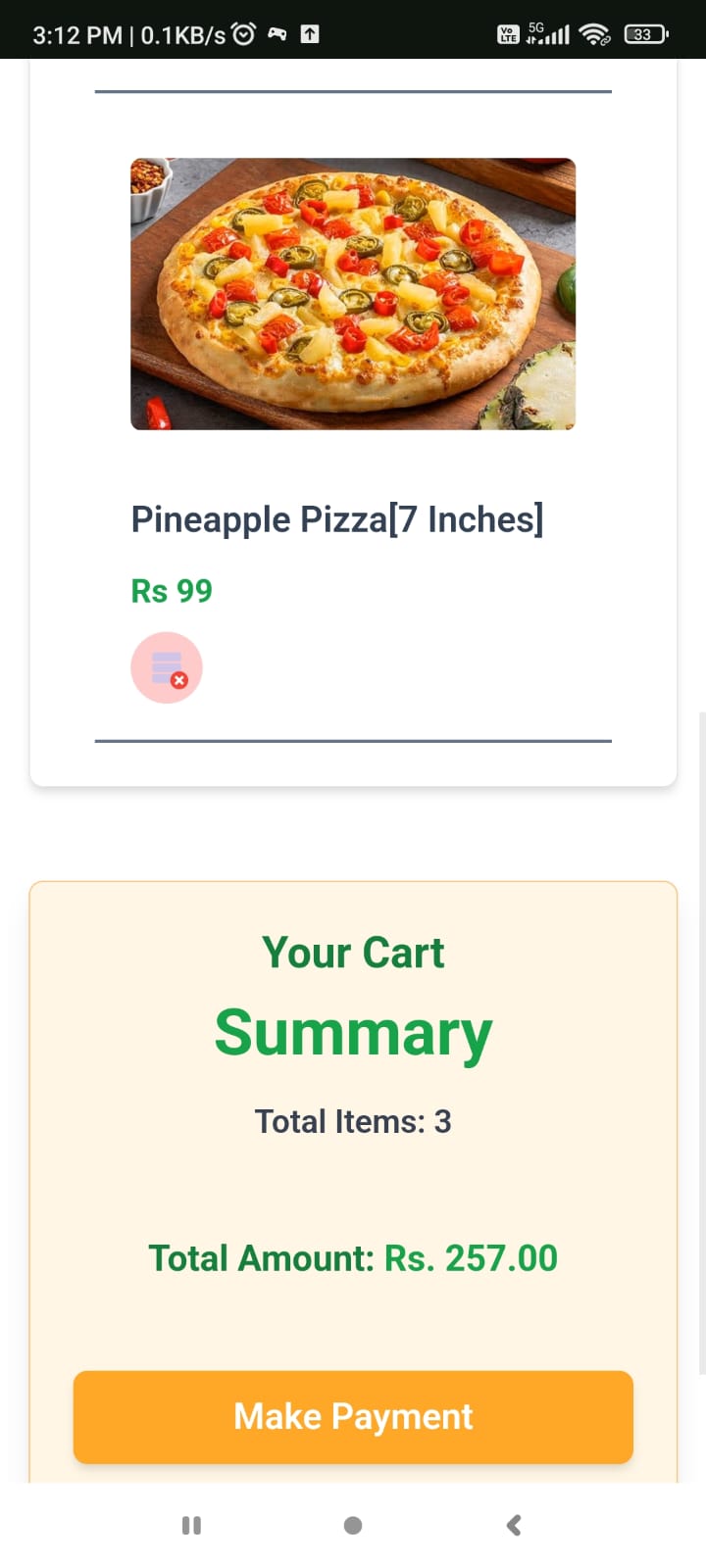
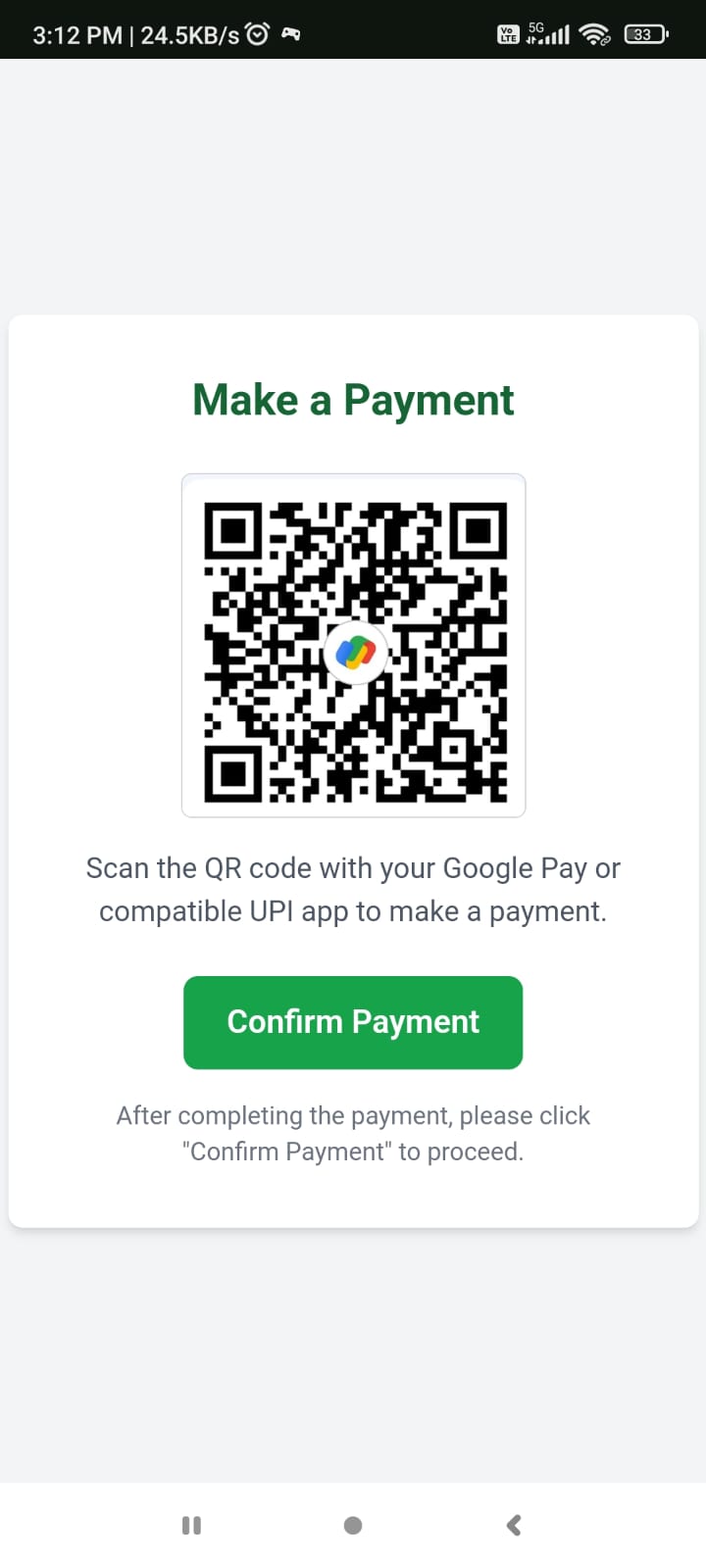
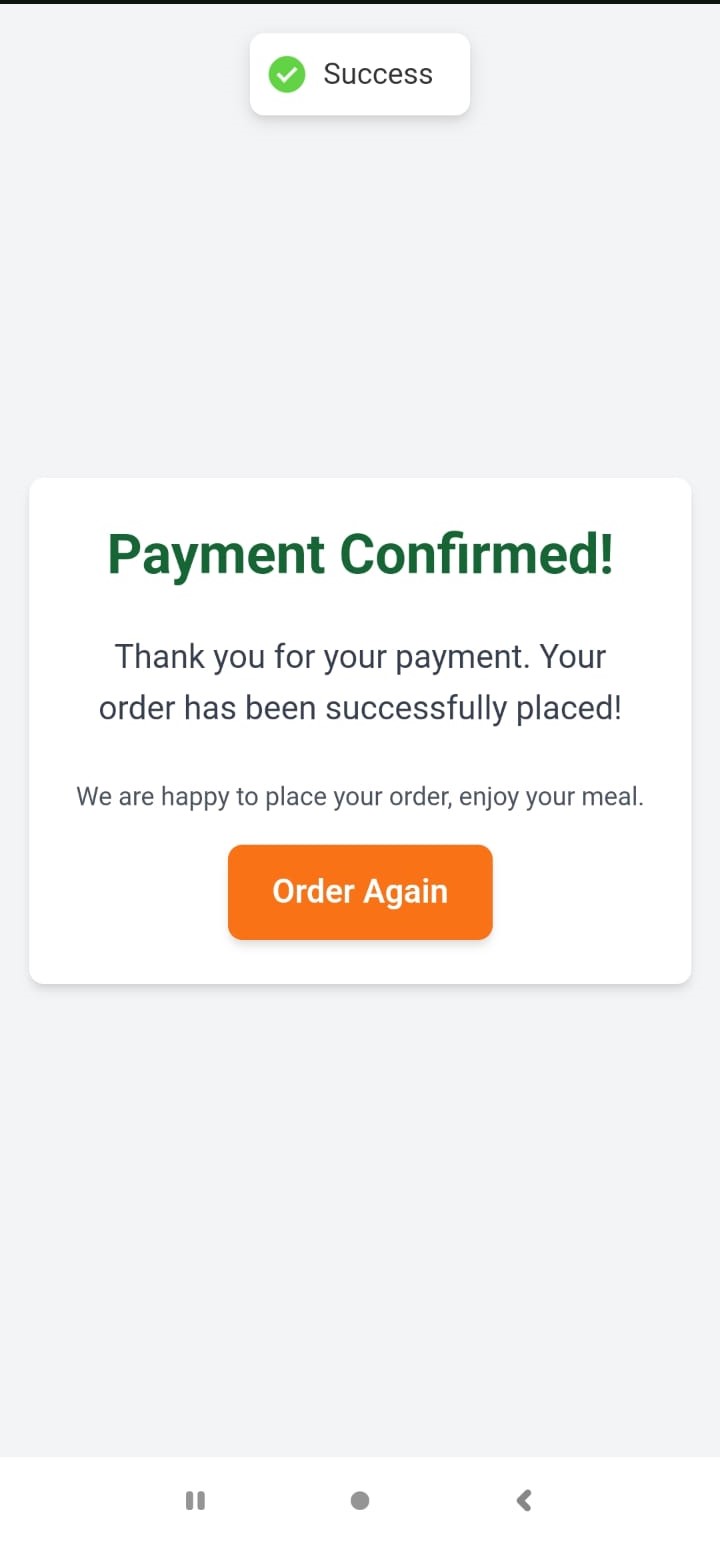
By exploring these future directions, the Restaurant Order Management System can continue to evolve, providing even more efficient and immersive solutions for both customers and restaurants, while pushing the boundaries of what’s possible in modern dining technology.

**CHAPTER 5**

**SCREENSHOTS**

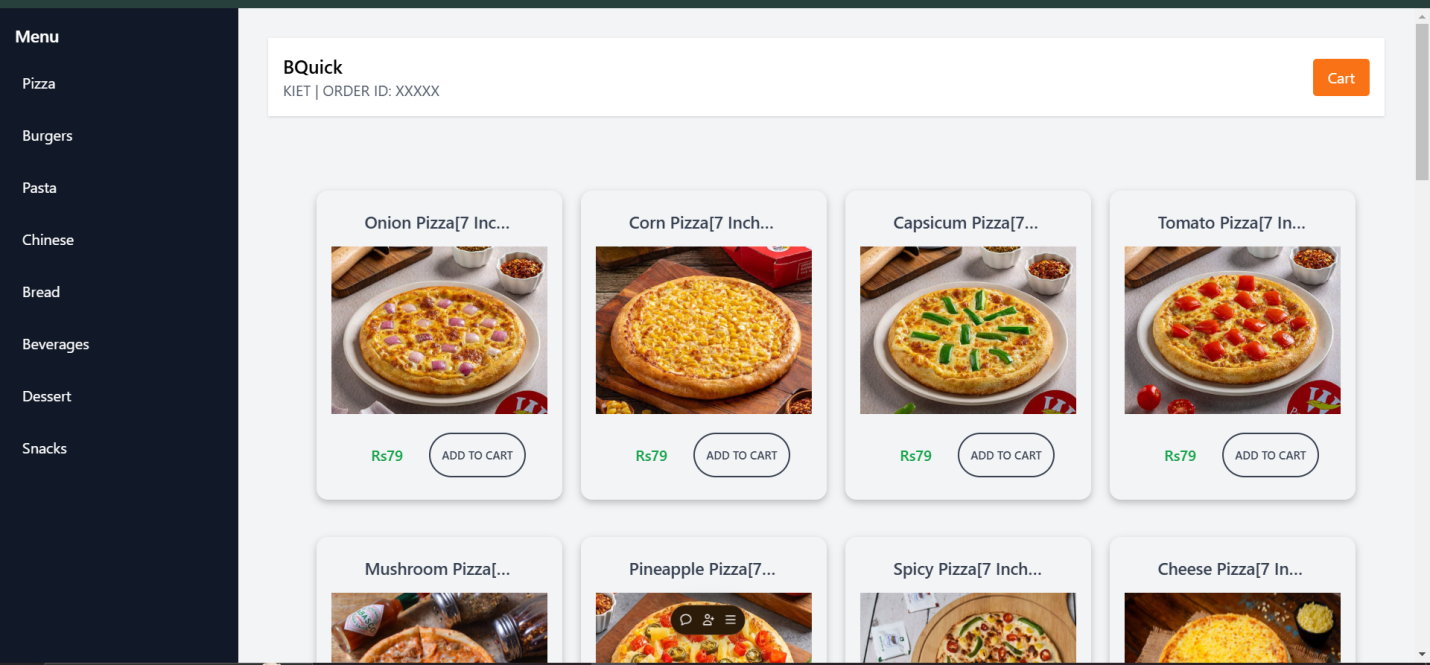
**MOBILE-VIEW**

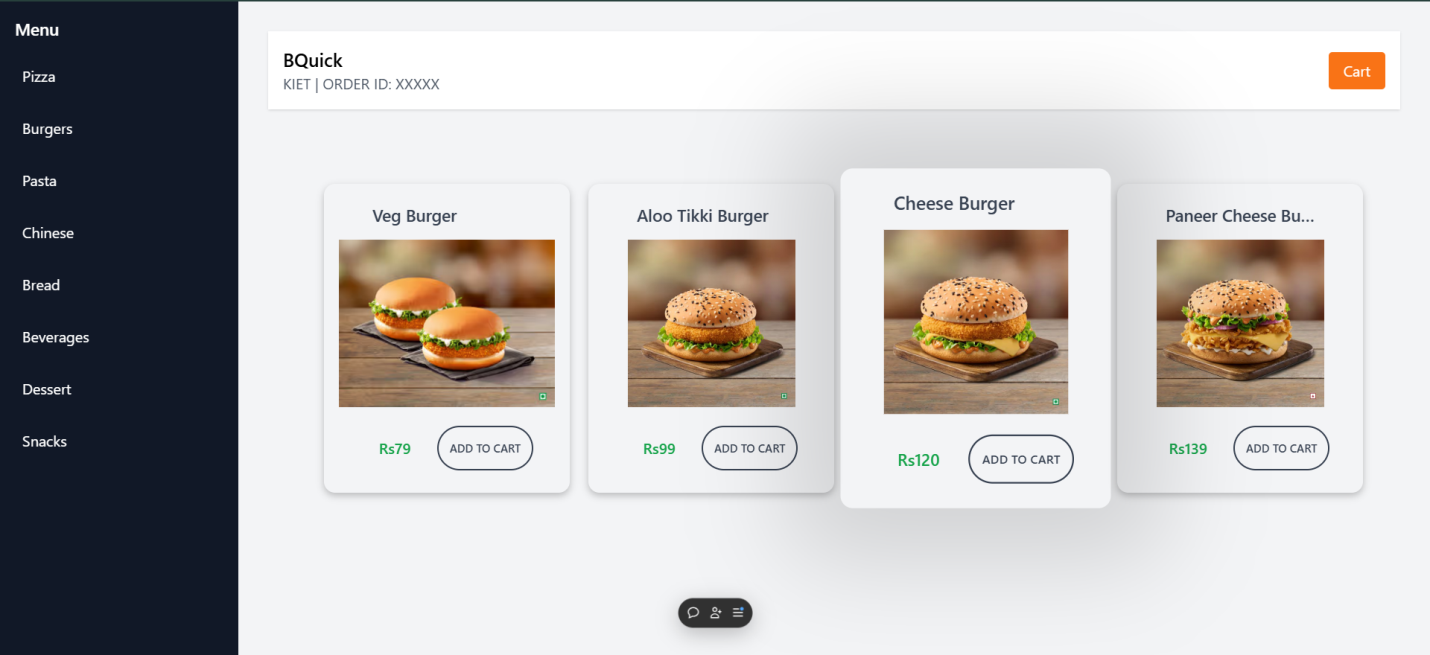
****

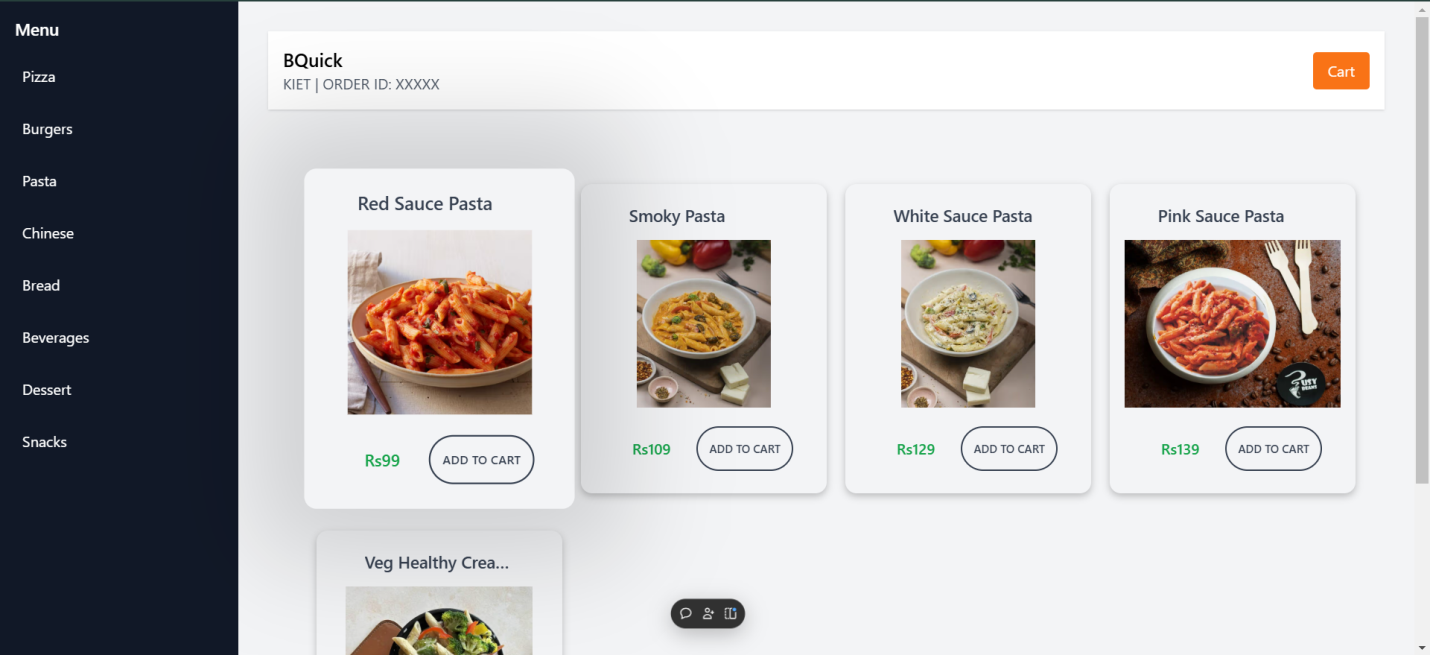


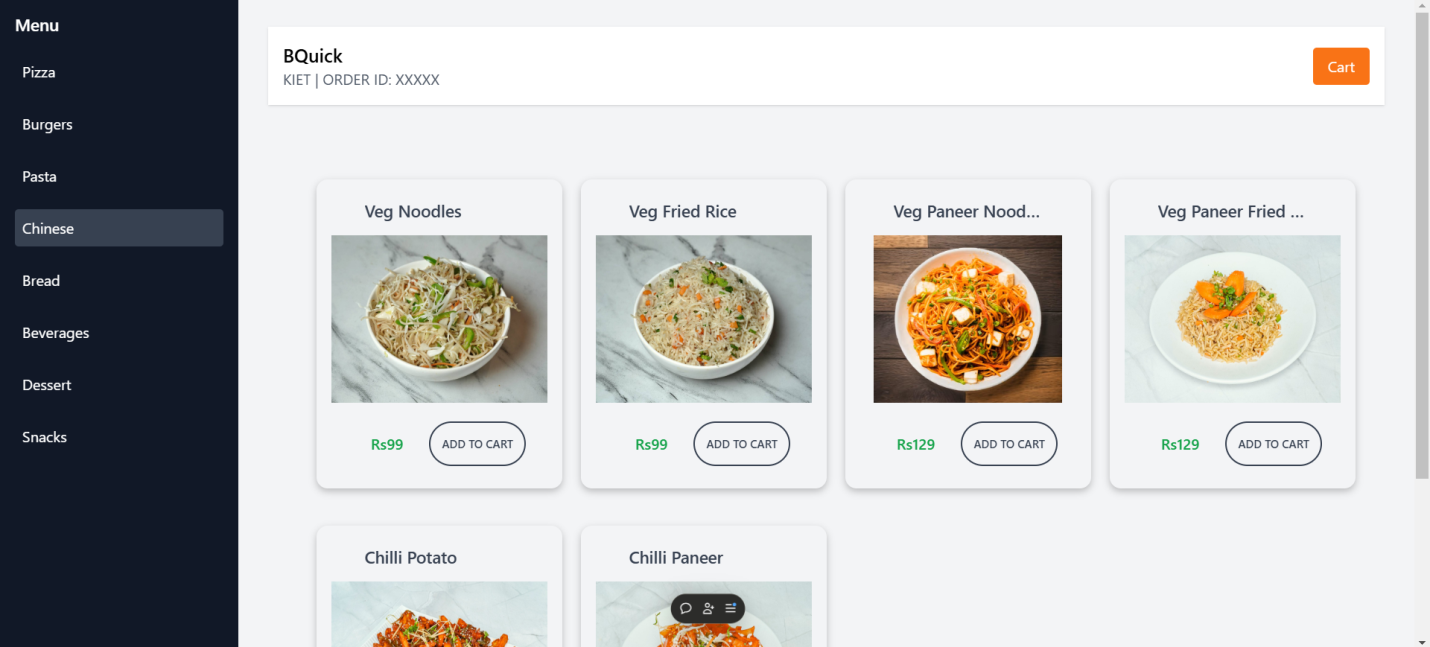
DESKTOP-VIEW

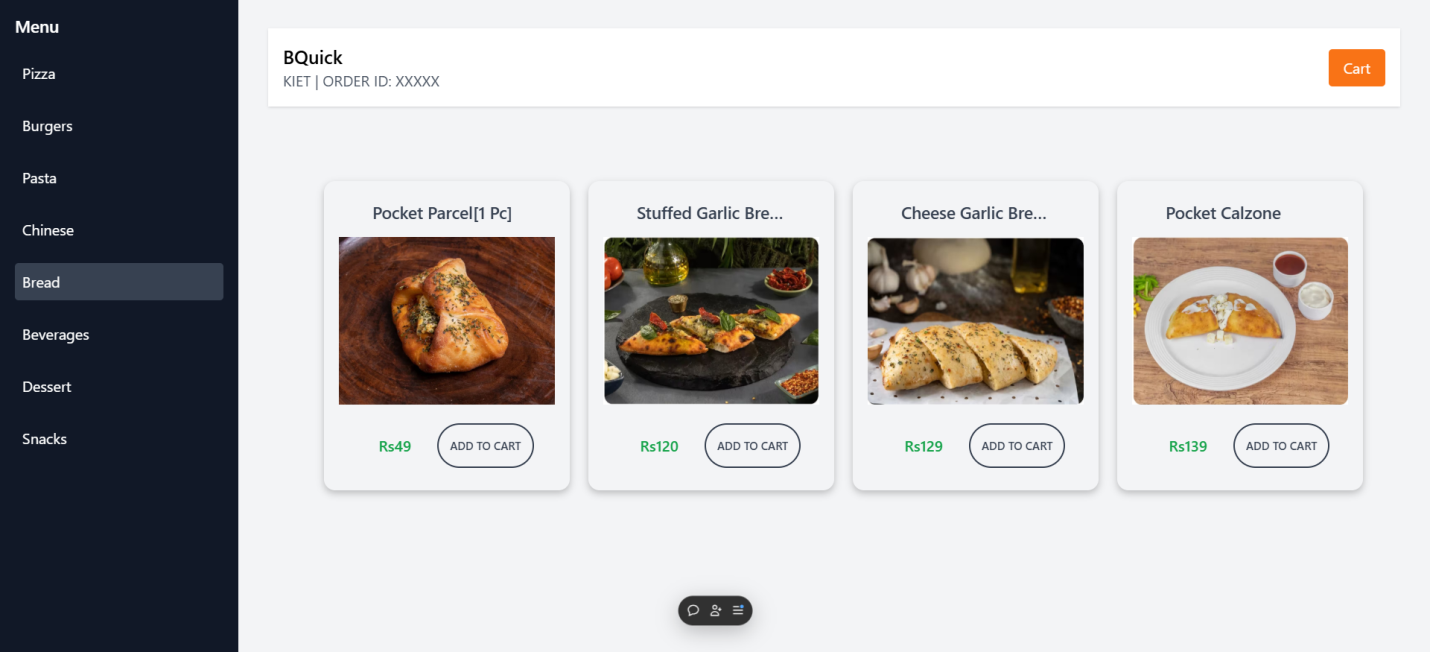


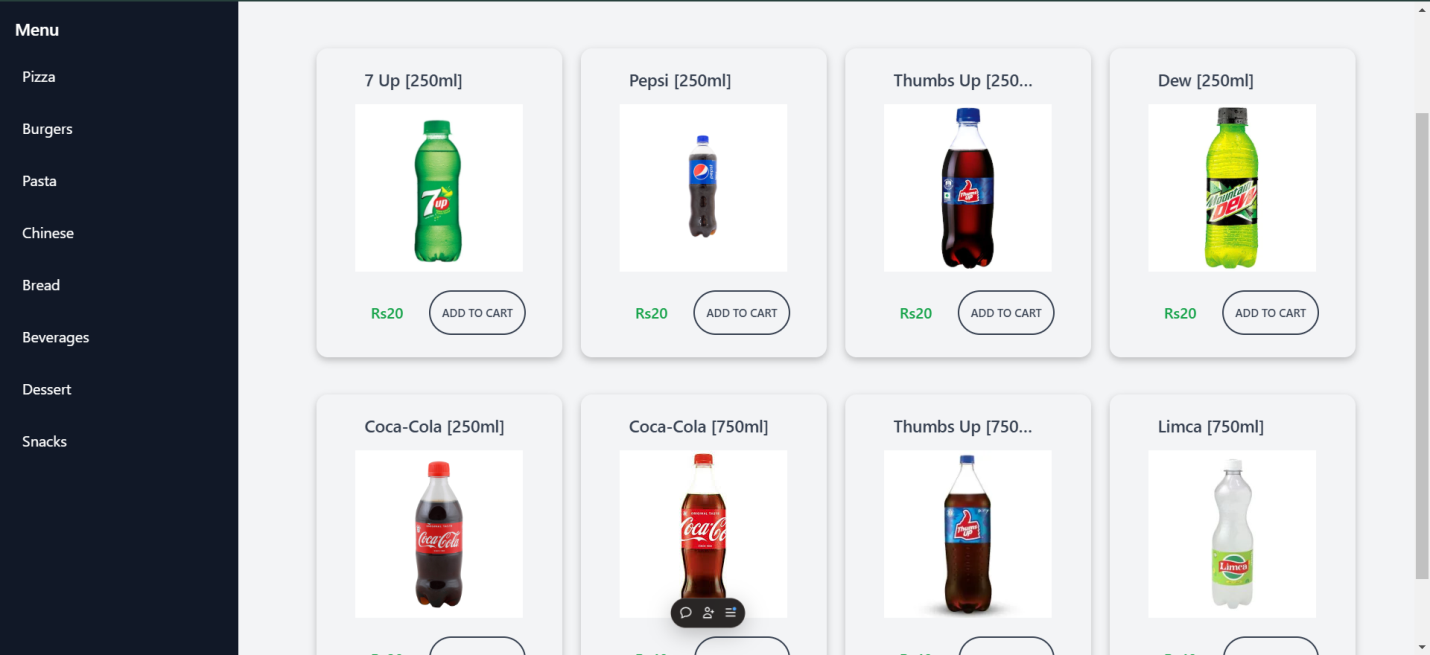


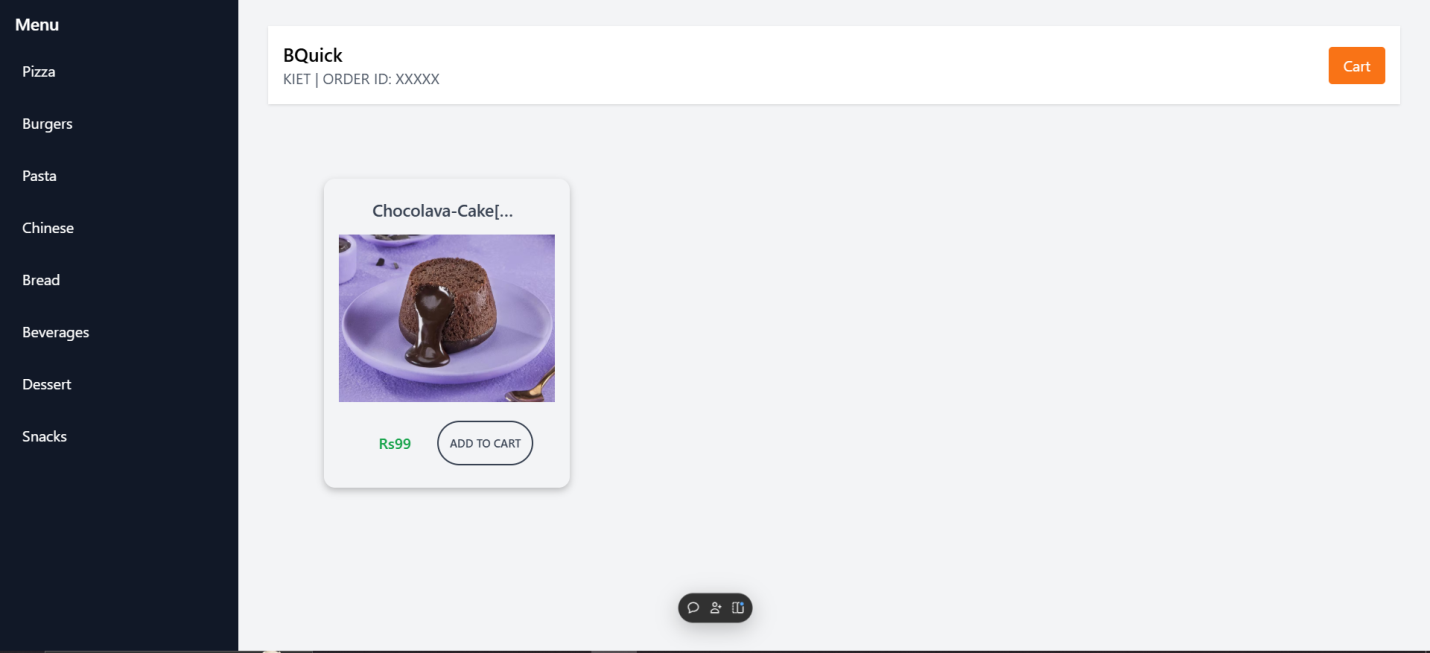


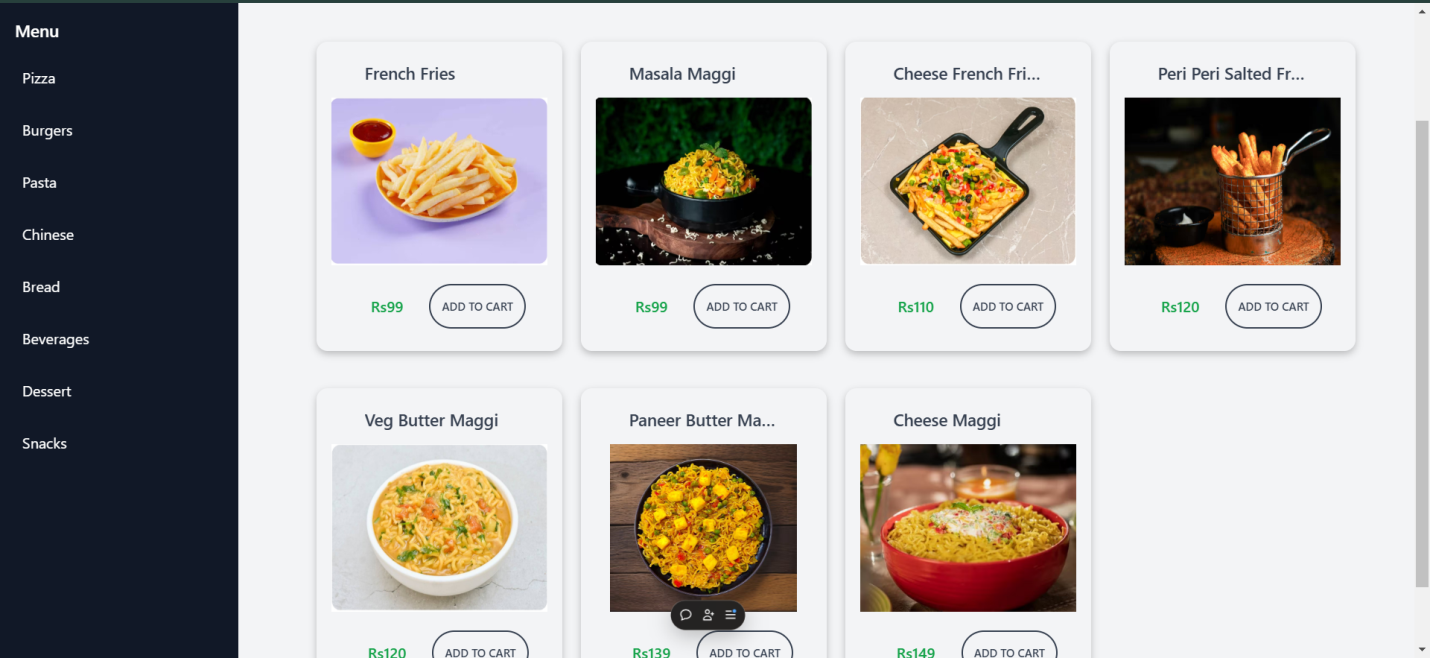


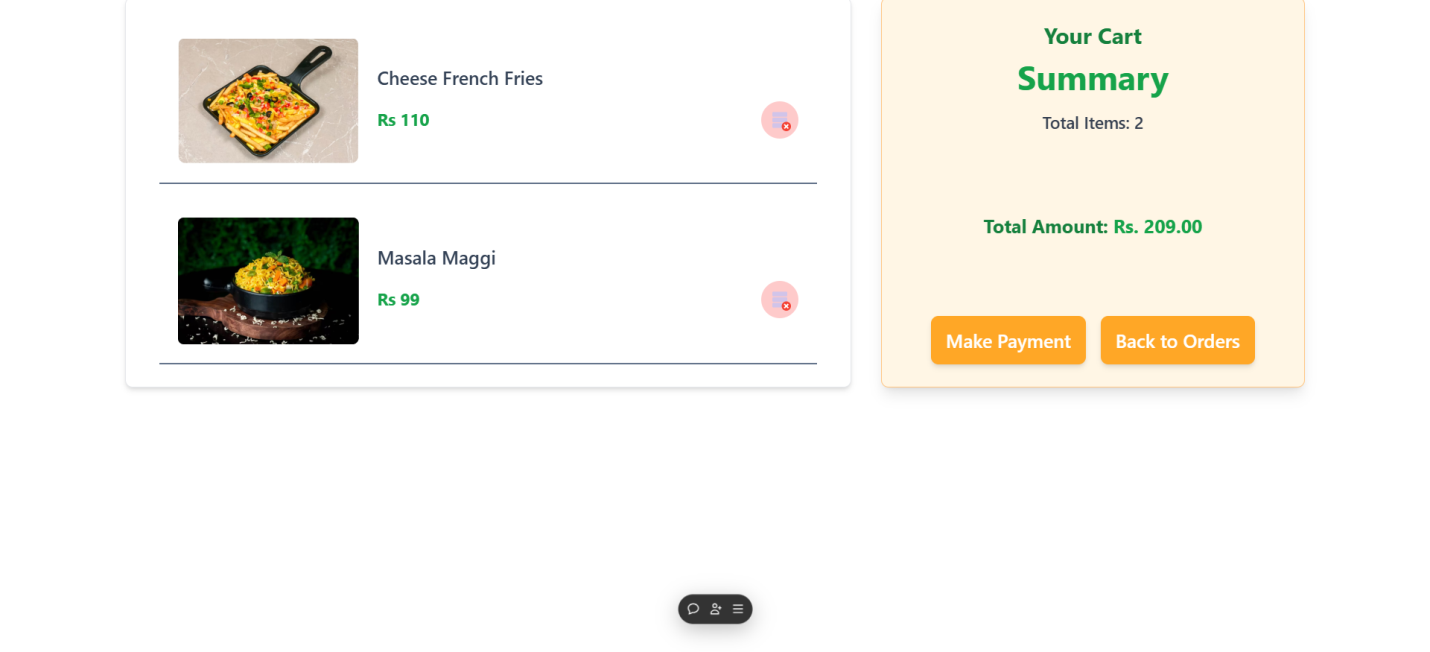


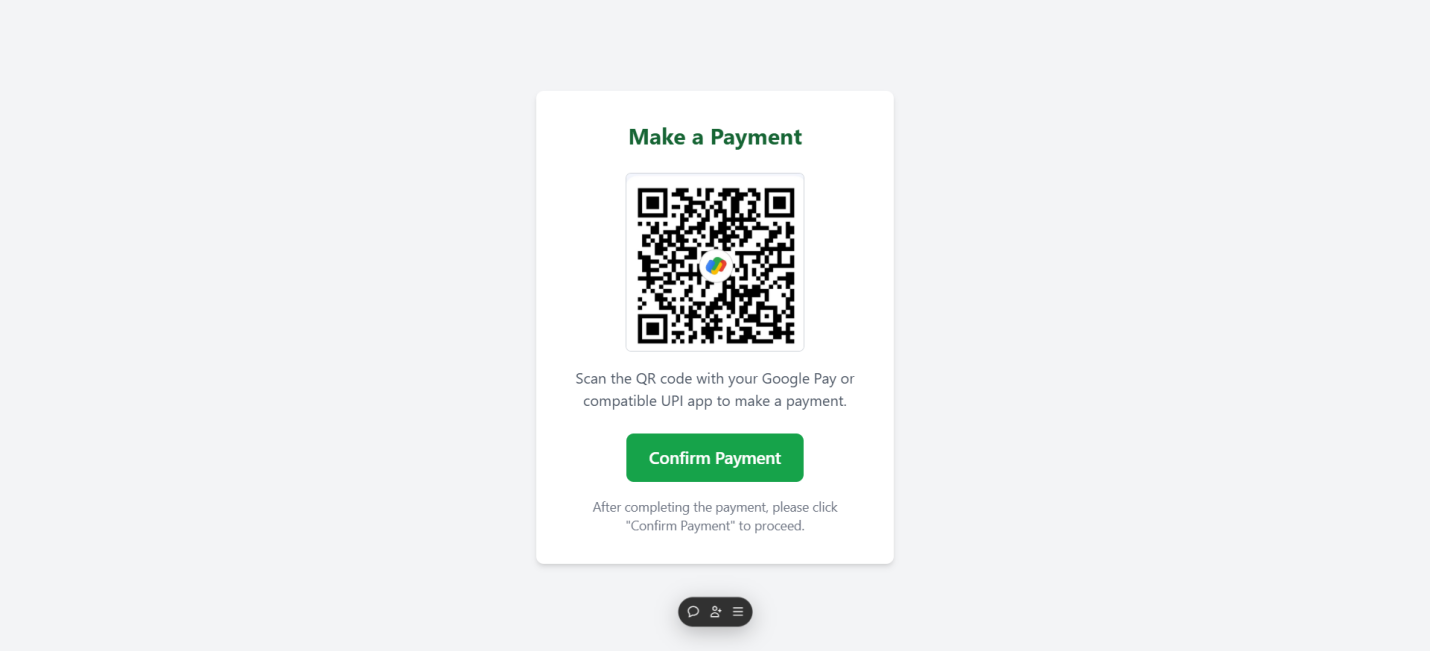


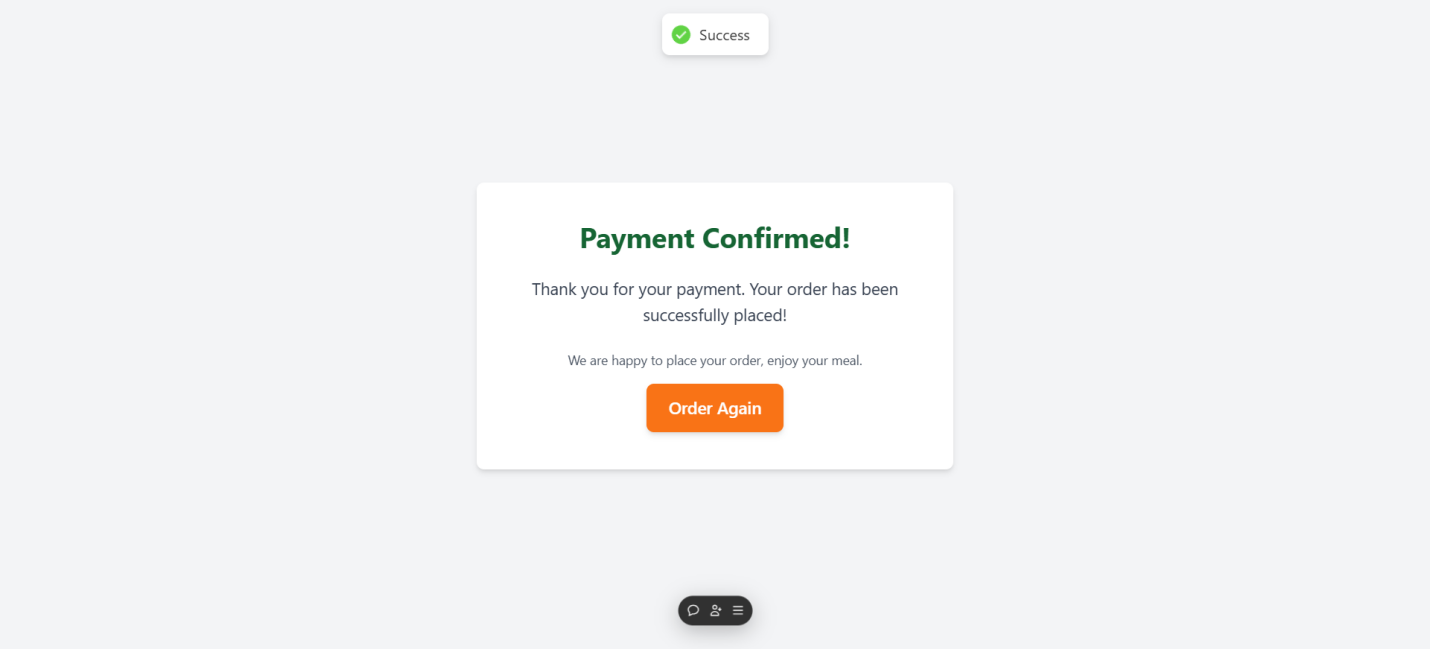












Codes

**Frontend**

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import { BrowserRouter } from 'react-router-dom';

import { store } from "./redux/slice/store";

import { Provider } from "react-redux";

import { Toaster } from "react-hot-toast";

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

    <BrowserRouter>

   <Provider store={store}>

      <App />

      <Toaster />

    </Provider>

  </BrowserRouter>

);

App.js

import React from 'react'

import {Routes,Route} from 'react-router-dom'

import Welcome from './pages/welcomepage'

import OrderManagement from './pages/ordermanagement';

import Cart from '../src/pages/cart'

import PaymentPage from './pages/paymentpage'

import OrderSummary from './pages/ordersummary'

function App() {

  return (

    <div>

    {/\* ............................routes..................... \*/}

<Routes>

  {/\* ..........doo page hai eisliye dooroute ........... \*/}

  <Route path="/" element={<Welcome/>}/>

  <Route path="/cart" element={<Cart/>}/>

  <Route path="/welcome" element={<Welcome/>}/>

  <Route path="/payment" element={<PaymentPage/>}/>

  <Route path="/order-summary" element={<OrderSummary/>}/>

  <Route path="/order" element={<OrderManagement/>}/>

</Routes>

    </div>

  );

}

export default App;

welcomepage.js

import React, { useState } from 'react';

import OrderManagement from './ordermanagement';  // Adjust path as necessary

import { NavLink } from "react-router-dom";

import PartyPopers from '../partypopers/party'

const App = () => {

  // useState hook to track the current screen

  const [hasOrdered, setHasOrdered] = useState(false);

  // Handler for button click to change the state

  return (

    <div className="flex items-center justify-center min-h-screen bg-gradient-to-r from-orange-100 via-orange-200 to-orange-300">

      <PartyPopers/>

      {!hasOrdered ? (

        <div className="text-center space-y-8">

          {/\* Title \*/}

          <h1 className="text-7xl font-extrabold text-gray-800 drop-shadow-lg">BQuick</h1>

          {/\* Order Button \*/}

          <NavLink to="/order">

          <button

            className="px-8 py-4 bg-orange-600 text-white text-2xl rounded-lg shadow-lg hover:bg-orange-700 transition-all duration-300 ease-in-out transform hover:scale-105"

          >

            Order

          </button>

          </NavLink>

        </div>

      ) : (

        <OrderManagement hasOrdered={hasOrdered} setHasOrdered={setHasOrdered} />

      )}

    </div>

  );

};

export default App;

cart.js

import React, { useState, useEffect } from "react";

import { useSelector } from "react-redux";

import { NavLink } from "react-router-dom";

import CartItem from "../components/cartitem";

function Cart() {

  const { cart } = useSelector((state) => state);

  const [totalAmount, setTotalAmount] = useState(0);

  // Calculate the total amount of items in the cart

  useEffect(() => {

    const calculateTotalAmount = () => {

      const total = cart.reduce((acc, curr) => acc + parseFloat(curr.price || 0), 0);

      setTotalAmount(total);

    };

    calculateTotalAmount();

  }, [cart]);

  return (

    <div className="mb-10 px-4">

      {cart.length > 0 ? (

        <div className="flex flex-col lg:flex-row flex-wrap justify-center max-w-[1300px] mx-auto gap-8">

          {/\* Cart Items \*/}

          <div className="w-full lg:w-[60%] flex flex-col p-4 bg-white shadow-md rounded-lg border border-gray-200">

            {cart.map((item, index) => (

              <CartItem key={item.id} item={item} itemindex={index} />

            ))}

          </div>

          {/\* Summary Section \*/}

          <div className="w-full lg:w-[35%] mt-5 lg:mt-0 flex flex-col bg-[#FFF6E5] shadow-lg rounded-lg p-6 border border-[#FFC480]">

            <div className="flex flex-col h-full justify-between gap-6">

              <div className="text-center">

                <h2 className="font-bold text-2xl text-green-700">Your Cart</h2>

                <h3 className="font-bold text-4xl text-green-600 mt-2">Summary</h3>

                <p className="text-gray-700 mt-4">

                  <span className="font-semibold text-lg">Total Items: {cart.length}</span>

                </p>

              </div>

              <div className="mt-6 text-center">

                <p className="text-xl font-bold text-green-700">

                  Total Amount: <span className="text-green-600">Rs. {totalAmount.toFixed(2)}</span>

                </p>

              </div>

              <div className="flex flex-col sm:flex-row gap-4 mt-6 justify-center">

                <NavLink to="/payment">

                  <button className="bg-[#FFA726] hover:bg-[#FB8C00] text-white px-4 py-3 rounded-lg font-semibold text-xl transition duration-300 shadow-md hover:shadow-lg w-full sm:w-auto">

                    Make Payment

                  </button>

                </NavLink>

                <NavLink to="/order">

                  <button className="bg-[#FFA726] hover:bg-[#FB8C00] text-white px-4 py-3 rounded-lg font-semibold text-xl transition duration-300 shadow-md hover:shadow-lg w-full sm:w-auto">

                    Back to Orders

                  </button>

                </NavLink>

              </div>

            </div>

          </div>

        </div>

      ) : (

        <div className="min-h-[80vh] flex flex-col justify-center items-center">

          <h1 className="text-gray-700 font-semibold text-2xl mb-4">

            Your cart is empty!

          </h1>

          <NavLink to="/">

            <button className="bg-[#FFA726] hover:bg-[#FB8C00] text-white py-3 px-10 rounded-lg font-semibold text-lg tracking-wider transition duration-300 shadow-md hover:shadow-lg">

              Order Again

            </button>

          </NavLink>

        </div>

      )}

    </div>

  );

}

export default Cart;

ordermanagement.js

import React, { useState ,useEffect} from "react";

import ItemListCard from '../components/itemlistcard'

import { NavLink } from 'react-router-dom';

import { MenuIcon} from 'lucide-react';

const OrderManagement = () => {

  const [listItem , setlistItme] = useState('pizza/get-pizza');

  const [openTopNav , setOpenTopNav] = useState(false);

  const URL = `https://bquick-backend.onrender.com/api/v1/${listItem}`;

  const [items, setitems] = useState([]);

//   ----------------fetch-data-function-----------

async function fetchItemsData() {

  try {

    const res = await fetch(URL , {method: 'GET'});

    const data = await res.json();

    console.log(data?.data);

    setitems(data?.data);

  } catch (error) {

    console.log("error aaye hai home page ke fet function me ");

    setitems([]);

  }

}

//   -----------to-run-fetch-function-----------------

useEffect(() => {

  console.log("Item is : " , listItem);

  fetchItemsData();

}, [listItem]);

  return (

    <div className={`flex min-h-screen bg-gray-100`}>

      {/\* Sidebar \*/}

      <aside className="w-64 bg-gray-900 text-white flex flex-col max-lg:hidden fixed h-full">

        <div className="p-4 text-lg font-semibold">Menu</div>

        <nav className="flex-1">

          <ul className="space-y-4 px-4">

            <li onClick={()=>setlistItme('pizza/get-pizza')} className="hover:bg-gray-700 p-2 rounded cursor-pointer">

              Pizza

            </li>

            <li onClick={()=>setlistItme('burger/get-burger')} className="hover:bg-gray-700 p-2 rounded cursor-pointer">

              Burgers

            </li>

            <li onClick={()=>setlistItme('pasta/get-pasta')} className="hover:bg-gray-700 p-2 rounded cursor-pointer">

              Pasta

            </li>

            <li onClick={()=>setlistItme('chinese/get-chinese')} className="hover:bg-gray-700 p-2 rounded cursor-pointer flex justify-between">

              Chinese

            </li>

            <li onClick={()=>setlistItme('bread/get-bread')} className="hover:bg-gray-700 p-2 rounded cursor-pointer">

              Bread

            </li>

            <li onClick={()=>setlistItme('beverage/get-beverage')} className="hover:bg-gray-700 p-2 rounded cursor-pointer">

              Beverages

            </li>

            <li onClick={()=>setlistItme('dessert/get-dessert')} className="hover:bg-gray-700 p-2 rounded cursor-pointer">

              Dessert

            </li>

            <li onClick={()=>setlistItme('snack/get-snack')} className="hover:bg-gray-700 p-2 rounded cursor-pointer">

              Snacks

            </li>

          </ul>

        </nav>

      </aside>

      <div className="hidden max-lg:block bg-gray-900 h-12 w-screen fixed top-0">

        <MenuIcon onClick={() => setOpenTopNav(!openTopNav)} className="fixed right-0 m-4" color="white"/>

      </div>

      {

        openTopNav &&

        <div className="hidden gap-2 fixed top-11 h-24 max-lg:flex w-full bg-gray-900 text-white flex-wrap justify-center items-center">

            <li onClick={()=>setlistItme('pizza/get-pizza')} className="hover:bg-blue-700 p-2 list-none rounded cursor-pointer">

              Pizza

            </li>

            <li onClick={()=>setlistItme('burger/get-burger')} className="hover:bg-blue-700 p-2 list-none rounded cursor-pointer">

              Burgers

            </li>

            <li onClick={()=>setlistItme('pasta/get-pasta')} className="hover:bg-blue-700 p-2 list-none rounded cursor-pointer">

              Pasta

            </li>

            <li onClick={()=>setlistItme('chinese/get-chinese')} className="hover:bg-blue-700 p-2 rounded list-none cursor-pointer flex justify-between">

              Chinese

            </li>

            <li onClick={()=>setlistItme('bread/get-bread')} className="hover:bg-blue-700 p-2 rounded list-none cursor-pointer">

              Bread

            </li>

            <li onClick={()=>setlistItme('beverage/get-beverage')} className="hover:bg-blue-700 p-2 list-none rounded cursor-pointer">

              Beverages

            </li>

            <li onClick={()=>setlistItme('dessert/get-dessert')} className="hover:bg-blue-700 p-2 list-none rounded cursor-pointer">

              Dessert

            </li>

            <li onClick={()=>setlistItme('snack/get-snack')} className="hover:bg-blue-700 p-2 list-none rounded cursor-pointer">

              Snacks

            </li>

        </div>

      }

      {/\* Main Content \*/}

      <div className="flex-1 p-8 ml-64 max-lg:ml-0 max-lg:mt-20">

        {/\* Top Bar \*/}

        <header

  className={`flex justify-between items-center bg-white shadow p-4 mb-8 transition-all duration-300 ${

    openTopNav ? "mt-14" : "mt-0"

  }`}

>

  <div>

    <h1 className="text-xl font-semibold">BQuick</h1>

    <p className="text-gray-600">KIET | ORDER ID: XXXXX</p>

  </div>

  <NavLink to="/cart">

    <button className="bg-orange-500 text-white px-4 py-2 rounded">

      Cart

    </button>

  </NavLink>

</header>

         {/\* -----------------items listed here--------------- \*/

        items ? (

        <div

        className="grid md:grid-cols-3 sm:grid-cols-2 sx:grid-cols-1 lg:grid-cols-4 max-w-6xl p-2 mx-auto space-y-10 space-x-5

        min-h-[80px]"

        >

        {

              items.map((item) => (

                <ItemListCard key={item.id} item={item}/>) )

        }

        </div>

      ) : (

        <div

        className="flex justify-center items-center"

        >

          <p>No items Found</p>

        </div>

      )}

      </div>

    </div>

  );

};

**backend**

index.js

import express from "express";

import dotenv from "dotenv";

import colors from "colors";

import pizzaRoutes from "./routes/pizzaRoutes.js"

import beverageRoutes from "./routes/beveragesRoutes.js"

import burgerRoutes from "./routes/burgerRoutes.js"

import breadRoutes from "./routes/breadRoutes.js"

import dessertRoutes from "./routes/dessertRoutes.js"

import pastaRoutes from "./routes/pastaRoutes.js"

import snackRoutes from "./routes/snackRoutes.js"

import chineseRoutes from "./routes/chineseRoutes.js"

import cors from "cors";

var app = express();

dotenv.config();

//middlewares

app.use(cors());

app.use(express.json())

// routes

app.use('/api/v1/pizza', pizzaRoutes);

app.use('/api/v1/beverage', beverageRoutes);

app.use('/api/v1/burger', burgerRoutes);

app.use('/api/v1/bread', breadRoutes);

app.use('/api/v1/chinese', chineseRoutes);

app.use('/api/v1/dessert', dessertRoutes);

app.use('/api/v1/pasta', pastaRoutes);

app.use('/api/v1/snack', snackRoutes);

//rest api

app.get('/', (req, res) => {

    res.send("<h1>Welcome to BQuick app</h1>")

})

//PORT

const PORT = process.env.PORT || 8080;

var server = app.listen(PORT, () => {

    console.log(`Server running on ${process.env.DEV\_MODE} mode port ${PORT}`.bgCyan.white);

})

Routes.js

import express from 'express';

import { getAllBeverage, getBeverageByName } from '../controllers/beverageController.js';

const router = express.Router();

//get all pizza

router.get('/get-beverage', getAllBeverage);

//single product

router.get('/get-beverage/:name', getBeverageByName);

export default router;

beveragescontroller.js

import fs from "fs";

import path from "path";

import { fileURLToPath } from 'url';

import { dirname } from 'path';

// import slugify from "slugify";

//getting details of beverage

export const getAllBeverage = (req, res) => {

    try {

        const \_\_filename = fileURLToPath(import.meta.url);

        const \_\_dirname = dirname(\_\_filename);

        const beverageFilePath = path.resolve(\_\_dirname, "../data/beverages.json");

        console.log("Resolved beverage file path:", beverageFilePath); // Log the resolved path

        if (!fs.existsSync(beverageFilePath)) {

            throw new Error(`File not found at path: ${beverageFilePath}`);

        }

        const beverageData = fs.readFileSync(beverageFilePath, "utf-8");

        const beverage = JSON.parse(beverageData);

        console.log(beverage);

        res.status(200).send({

            success: true,

            data: beverage,

        });

    } catch (err) {

        // Log full error stack for debugging

        console.error("Error fetching beverage data:", err.stack);

        console.log(err);

        res.status(500).send({

            success: false,

            message: "Error in fetching details of all products",

            err,

        })

    }

}

// getting single beverage by name

export const getBeverageByName = (req, res) => {

    console.log({ slug: req.params.name });

    try {

        const \_\_filename = fileURLToPath(import.meta.url);

        const \_\_dirname = dirname(\_\_filename);

        const beverageName = req.params.name; // get beverage name from URL parameters

        console.log(beverageName)

        // Resolve the beverage.json file path

        const beverageFilePath = path.resolve(\_\_dirname, "../data/beverages.json");

        if (!fs.existsSync(beverageFilePath)) {

            throw new Error(`File not found at path: ${beverageFilePath}`);

        }

        // Read the beverage JSON file

        const beverageData = fs.readFileSync(beverageFilePath, "utf-8");

        const beverageArray = JSON.parse(beverageData);

        // Find beverage by name (case-insensitive comparison)

        const beverage = beverageArray.find(p => p.name.toLowerCase() === beverageName.toLowerCase());

        if (!beverage) {

            return res.status(404).send({

                success: false,

                message: `Beverage with name "${beverageName}" not found`,

            });

        }

        res.status(200).send({

            success: true,

            data: beverage,

        });

    } catch (err) {

        console.error("Error fetching beverage by name:", err.stack);

        res.status(500).send({

            success: false,

            message: "Error in fetching beverage details",

            error: err.message,

        });

    }

};

Data.json

[

    {

        "id": "Beverages01",

        "name": "7UP",

        "image": "Onion Pizza.jpg",

        "price": [

            "(20)",

            "(40)",

            "(50)"

        ],

        "size": [

            "(250ml)",

            "(750ml)",

            "(1L)"

        ]

    },

    {

        "id": "Beverages02",

        "name": "Pepsi",

        "image": "Onion Pizza.jpg",

        "price": [

            "(20)",

            "(40)",

            "(50)"

        ],

        "size": [

            "(250ml)",

            "(750ml)",

            "(1L)"

        ]

    },

    {

        "id": "Beverages03",

        "name": "Thumbsub",

        "image": "Onion Pizza.jpg",

        "price": [

            "(20)",

            "(40)",

            "(50)"

        ],

        "size": [

            "(250ml)",

            "(750ml)",

            "(1L)"

        ]

    },

    {

        "id": "Beverages04",

        "name": "Dew",

        "image": "Onion Pizza.jpg",

        "price": [

            "(20)",

            "(40)",

            "(50)"

        ],

        "size": [

            "(250ml)",

            "(750ml)",

            "(1L)"

        ]

    },

    {

        "id": "Beverages05",

        "name": "Coca-Cola",

        "image": "Onion Pizza.jpg",

        "price": [

            "(20)",

            "(40)",

            "(50)"

        ],

        "size": [

            "(250ml)",

            "(750ml)",

            "(1L)"

        ]

    }

**CHAPTER 6**

**CONCLUSION**

In conclusion, “BQuick” represents a significant advancement in the field of restaurant technology, seamlessly integrating modern web technologies, efficient order processing, and real-time customer interaction to enhance the dining experience. The system's use of React for a responsive, user-friendly interface and Node.js for scalable backend processing has resulted in a robust and efficient platform that streamlines the order process and reduces wait times, benefiting both customers and restaurant staff. The innovative approach of QR code scanning for menu access and order placement sets a new standard for customer convenience, while the real-time order management system ensures smooth communication between customers, staff, and kitchen personnel.

Moreover, the project highlights the importance of accessibility and inclusivity in modern technology systems. By offering mobile and cross-platform compatibility, the system ensures a wide reach, allowing a variety of customers to interact with the system with ease. While there are areas for further enhancement, such as improving system performance on lower-end devices and expanding offline functionality, the project lays a strong foundation for future innovations in restaurant management. With continuous optimization and the incorporation of additional features such as AI-driven suggestions and enhanced payment integrations, the system is poised for further growth.

Looking ahead, there are numerous possibilities for expanding and enhancing the system, such as integrating advanced analytics to optimize menu offerings, incorporating voice commands for hands-free order placement, and improving accessibility features for a more inclusive experience. Research into the integration of emerging technologies, such as virtual reality or augmented reality, could offer innovative ways to further enrich customer interaction and engagement. Additionally, as the system expands, future enhancements could focus on optimizing the payment process, enhancing security features, and offering personalized experiences based on customer preferences.

Ultimately, “BQuick” stands as a testament to the dedication and innovative spirit of the development team in creating a cutting-edge solution that modernizes the restaurant experience. Through careful attention to user experience, technical performance, and scalability, the system has set new standards in the restaurant industry. As we continue to evolve and refine the system, we look forward to delivering more efficient, engaging, and customer-centric solutions that shape the future of dining technology.

In essence, “BQuick” is not just a tool for managing orders—it is a step forward in transforming the dining experience. It serves as an example of how technology, when applied thoughtfully, can improve operational efficiency, enhance customer satisfaction, and redefine industry standards.

**REFERENCES**

* Technology in Enhancing Customer Service Efficiency in Hospitality. (2022). Retrieved from https://hospitalitytechjournal.org.
* Self-Service Technologies in Restaurants: A Consumer Perspective. (2021). Journal of Modern Dining Experiences, 34(2), 45–56.
* Smith, J. (2020). Queue Management Systems for Service-Oriented Businesses. TechPress Publications, Boston.
* Kumar, A., & Tripathi, A. (2024). Customer-Centric Innovations in Restaurant Management. Pearson Education.

**BIBLIOGRAPHY**

- Digital Queue Management Systems: https://queuemanagementsystems.org.

Reference Books:

- Smith, J. (2020). Queue Management Systems for Service-Oriented Businesses. TechPress Publications, Boston.

- Kumar, A., & Tripathi, A. (2024). Customer-Centric Innovations in Restaurant Management. Pearson Education.

- HospitalityTech Journal: https://hospitalitytechjournal.org.