Grenze International Journal of Engineering and Technology, January Issue

A Smart QR-based Food Ordering System with Dynamic Menu and Checkout Integration

Shabana Pathan[1], Simran Daultani[2], Om Dhoble[3] and Vrinda Pillai[4]
[1-4]St. Vincent Pallotti College of Engineering and Technology, Nagpur, India
Email: spathan@stvincentngp.edu.in, simran.daultani2345@gmail.com, dhobleom30@gmail.com, ajayvrinda381@gmail.com

***Abstract*— This paper presents the design and implementation of a web-based food ordering system that leverages modern web technologies and secure user authentication to streamline the dining experience. Customers can scan a static QR code to access a dynamic online menu, select food items, and add them to a cart. A unique, dynamic QR code is generated during checkout for order confirmation and payment processing. The system supports both dine-in and takeout options and ensures secure user handling using Clerk authentication. Technologies used include Next.js for frontend/backend integration, PostgreSQL for database management, and Clerk for secure user sessions.**

***Keywords*— PostgreSQL, QR Code, Food Ordering System, Web Application, Secure Authentication, Takeout, Dine-in, Dynamic Checkout.**

I. INTRODUCTION

The evolution of digital technology has significantly transformed the way services are delivered across various industries, and the restaurant sector is no exception. With the growing demand for contactless experiences and streamlined operations, traditional dining methods are being replaced by modern, tech-driven solutions. Customers today seek convenience, speed, and personalization, especially when it comes to ordering food in restaurants.

This paper presents the implementation of a web-based food ordering platform designed to enhance the dining and takeout experience through the use of QR codes and secure digital interfaces. The platform allows restaurant customers to scan a static QR code placed at their table (for dine-in) or at the counter (for takeout), instantly accessing a responsive digital menu. From there, users can browse food items, customize their order, and add selections to a virtual cart.

The system is built using Next.js, a React-based framework known for its performance and server-side rendering capabilities. PostgreSQL is used for database management, ensuring efficient handling of structured data such as user information, menu items, and orders. To guarantee user data security and smooth authentication, Clerk is integrated, providing secure sign-in, session management, and user profile handling.

One of the key features of this system is the dynamic QR code generation during checkout. This QR code is unique to each order and enables users to proceed to a secure payment gateway, eliminating the need for physical bills or manual confirmation. The system also caters to both dine-in and takeout scenarios, making it flexible and scalable for different restaurant setups.

Grenze ID: 01.GIJET.12.1.347_1

By leveraging modern web technologies and intuitive design, this platform aims to reduce wait times, improve order accuracy, and provide a seamless, user-friendly interface for both customers and restaurant staff. This paper outlines the technical approach, development process, and real-world applications of the system, highlighting its effectiveness and potential impact in the food service industry.

II. LITERATURE SURVEY

This research paper [1], "Dynamic QR Code Generation for Personalized Ordering Experiences" (2024), shows how restaurants can generate dynamic QR codes that reflect real- time menu updates, discounts, and special offers, enhancing user engagement and maximizing upselling opportunities.

Research paper [2], "Implementing Secure Authentication in Web-Based Ordering Systems Using Clerk API" (2023), focuses on integrating third-party authentication services like Clerk for enhancing security in online food ordering platforms. The authors discuss how Clerk's secure token handling and multi-factor authentication (MFA) improve data protection and user trust.

In study on paper [3],"Challenges in Developing Cross- Platform Food Ordering Applications" (2023), identifies difficulties in creating apps that function smoothly across Android, iOS, and web browsers. The paper recommends using frameworks like Next.js and React Native to ensure code reusability and a consistent user experience.

The investigation on paper [4],"Optimizing Checkout Processes in Online Food Ordering Applications" (2024), explores how streamlining checkout flows—such as fewer clicks, guest checkout, and QR-driven payment options— significantly boosts order conversion rates. The paper also stresses minimizing cognitive load during final purchasing steps.

This paper [5] on "Dynamic Menu Customization Based on User Behavior in Food Ordering Apps" (2024), investigates AI techniques are described which are used to track user preferences and display personalized menus. The paper shows how real-time adaptation increases order volume and improves the customer's dining experience.

The presentation of methods like server-side rendering (SSR), static site generation (SSG), and API route optimization in Next.js in paper [6], "Improving Performance of Next.js Web Applications for Food Services" (2023) to reduce load times and enhance user experience for web-based food ordering systems has been explained.

The study in paper [7], "Development of a Mobile Web- Based Food and Beverage Ordering Application in a Youth Cafe with QR Code Technology" (2025) emphasizes the importance of trendy designs, mobile responsiveness, and quick transaction flows to appeal to younger demographics the paper describes a youth-focused ordering system where users access digital menus via QR codes.

The research paper [8] on “Cafeteria Food Ordering System Using QR Code" (2023), proposes a digital ordering system where customers scan QR codes placed on cafeteria tables. The system automates the ordering process, minimizes human errors, and provides an easy-to-update menu management solution for cafeteria administrators.

The paper [9] "Design and Development of Groceries Ordering System with QR Code" (2023), introduces a groceries ordering platform leveraging QR codes for contactless product browsing and purchasing. The authors focus on enhancing shopping convenience, real-time inventory updates, and customer satisfaction by digitizing the traditional shopping experience.

This paper [10] "Responsive Web-Based Cafe Food Ordering System Using Bootstrap and QR Code" (2023), presents the development of a mobile-responsive food ordering system using QR code scanning. The research emphasizes reducing customer wait times and manual errors by integrating simple web technologies like Bootstrap, resulting in improved usability and cafe efficiency.

A digital grocery ordering solution aimed at simplifying shopping experiences has been explained in research paper[11] "Design and Development of a QR Code-Based Grocery Ordering System" (2024).Customers use QR codes to access real-time updated menus, improving purchasing efficiency and reducing in-store crowding.

The study as of how implementing QR code systems in dining establishments reduces wait times, minimizes errors, and increases overall customer satisfaction has been discussed in paper [12],“Impact of QR Code-Based Ordering Systems on Restaurant Efficiency" (2023). The paper uses case studies from small and medium-sized restaurants.

In the research paper [13], "Dynamic QR Code Generation for Personalized Ordering Experiences" (2024), investigates the creation of dynamic QR codes that can adjust based on time, day, or special promotions. Their findings show that personalized QR codes significantly enhance customer engagement and increase order values.

The paper [14] ,"The Evolution of Digital Menus: A UI/UX Perspective" (2023), discusses how static PDF menus have evolved into interactive digital experiences. It even emphasizes the importance of responsive design, vibrant food imagery, and intuitive navigation to drive more orders through mobile web platforms.
Multiple payment options, ensuring users complete their purchases without friction has been discussed in research paper [15], "Streamlining the Checkout Experience in QR- Based Ordering Systems" (2024).It also focuses on Designing simplified and intuitive checkout processes for food ordering apps, their research highlights the importance of fast checkouts, minimal forms, and
In paper [16] on,"Role of UI Aesthetics in Customer Loyalty for Food Ordering Websites" (2023), explores how the visual design of food ordering platforms directly impacts customer loyalty. Their research highlights that clean layouts, attractive imagery, and user-friendly interfaces lead to improved retention rates and repeat purchases among users.
The research study in paper [17] on, "Use of PostgreSQL for Efficient Backend Management in Ordering Systems" (2023),talks about details how PostgreSQL's performance tuning, indexing, and advanced query optimization support high-volume transaction handling for restaurant ordering systems, ensuring smooth operation during peak customer hours.
This research paper [18] on, "Enhancing User Experience through Responsive Web Design in Online Food Ordering Applications" (2023), analyzes how responsive layouts improve usability for food delivery apps. Their research shows that adaptive interfaces lead to higher user retention, especially when customers switch between devices like mobiles and tablets.

III. Problem Statement

In traditional restaurant settings, the process of ordering food often involves several manual steps that can lead to inefficiencies and poor customer experiences. Customers typically wait for staff to provide a physical menu, place their order verbally, and later wait again for the bill and payment process. This manual approach can lead to common problems such as delayed service, order inaccuracies, billing errors, and increased workload for restaurant staff. Moreover, during peak hours, these issues are amplified, resulting in customer dissatisfaction and potential loss of business.
Customers now expect faster, safer, and more self-reliant ways to interact with restaurants. However, many existing digital solutions either lack customization for dine-in and takeout scenarios or fail to provide secure, user-friendly interfaces. Additionally, data security and user privacy have become major concerns with digital platforms. Many restaurant apps and systems either store sensitive data insecurely or use outdated authentication methods, posing risks to both users and businesses.
Another gap in existing systems is the limited integration of QR code technology beyond basic menu access. While some restaurants use static QR codes to display digital menus, few offer full end-to-end functionality such as dynamic cart management, real-time checkout, and QR code-based payment initiation. This project aims to address these issues by building a modern web-based food ordering system that integrates QR-based interactions, secure authentication using Clerk, dynamic order management, and a streamlined checkout process—all built using robust technologies like Next.js and PostgreSQL.

A. Proposed Approach

To address the challenges identified in the traditional food ordering process, we propose a web-based food ordering system that leverages modern technologies and seamless user experiences. The solution is designed to be efficient, user- friendly, secure, and adaptable for both dine-in and takeout service models. The proposed approach revolves around five core components: digital menu access via QR code, secure user authentication, dynamic cart and order management, dynamic QR code generation for payment, and a robust backend for data handling and communication.

IV. Proposed System

Figure 1 represents an Smart food ordering System that integrates various components across different stages ofeach table or takeout area within the restaurant will have a static QR code displayed. When scanned using a smartphone, this QR code directs the customer to a dynamic web page tailored either for dine-in or takeout services. This eliminates the need for physical menus and minimizes direct interactions with staff, promoting a contactless environment. Upon accessing the system, users are prompted to either authenticate themselves through Clerk or continue as guests, depending on the restaurant’s preferences. Clerk provides a secure authentication framework, allowing users to sign up or log in through email, phone, or third-party providers such

as Google or Apple. Secure authentication ensures that user data is protected, sessions are managed efficiently, and features like order tracking, loyalty programs, and personalized service can be implemented. Once authenticated, users can access the restaurant's dynamic digital menu, which is maintained in a PostgreSQL database. The menu is organized into categories and includes item details such as descriptions, images, and prices. Customers can search for items, customize their orders based on available options, and add selections to an interactive cart. The cart updates in real-time, allowing users to review, modify, or remove items before proceeding to checkout. This dynamic cart functionality ensures complete transparency and flexibility in the ordering process.When users are ready to place their order, the system generates a unique dynamic QR code linked to their order details. This QR code can either be scanned by restaurant staff to retrieve the order or used by customers to proceed directly to a secure payment page. Payment is handled via trusted payment gateways like Stripe, Razorpay, or PayPal. Upon successful payment, the order status is updated in the backend database, and the customer receives a confirmation, while the kitchen staff can be notified immediately to begin preparation.

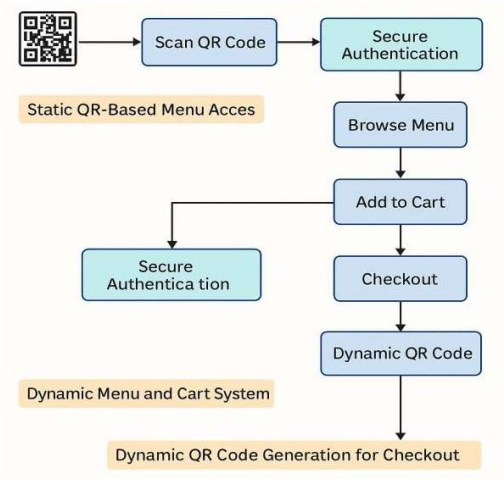

Fig.1 QR-Based food ordering System

V. PROPOSED METHODOLOGY

The image fig 2 the methodology for this food ordering website project follows a circular development process, emphasizing continuous improvement and user-centric design. The process begins with Requirement Gathering, where a detailed analysis is conducted to understand the restaurant's needs. It is identified that customers should be able to view the menu digitally, add food items to their cart, select between dine-in and takeaway options, and make payments via a dynamically generated QR code. Additional requirements include easy authentication, cart management, table number input for dine-in orders, and real-time order tracking.

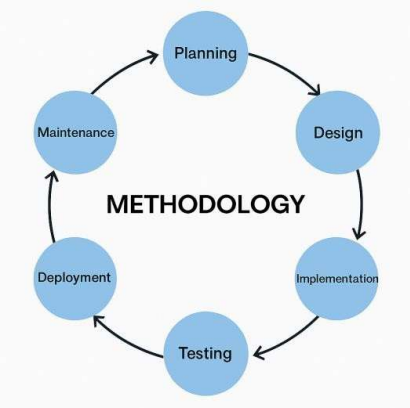

Fig 2. Methodology of Application

Following requirement gathering, the Planning phase structures the project roadmap. Decisions are made regarding the technology stack, selecting Next.js for fast and efficient front-end development, PostgreSQL for reliable backend database management, and Clerk for modern, secure authentication handling. User journeys are mapped out to ensure that moving between menu browsing, cart management, and checkout/payment is intuitive. During the Design phase, mockups and wireframes are created to visualize the user interface and experience. Focus is given to creating a minimalistic, clean design where users can easily navigate to the cart page, see a breakdown of items with their quantities and prices, and proceed smoothly to checkout. Responsive layouts are incorporated to ensure functionality on both mobile and desktop devices.

The platform presents a modular web-based ecosystem intended to simplify interactions between farmers, buyers, and suppliers in order to improve the innovative element of the Smart Agricultural Marketplace. This system incorporates intelligent automation, adaptive user interfaces, and real-time decision-making to maximize user engagement, in contrast to traditional agricultural platforms that only concentrate on product lists. The dashboard's ability to be customized and tailor recommendations according to user responsibilities is a significant advance.

Through offline caching and background synchronization, the platform also incorporates progressive web application (PWA) capabilities, which enable users to access services even in places with poor connectivity. Predictive demand analysis is also made possible by an automated supply chain management module, which guarantees that farmers can foresee market movements and maximize stock availability. Role-based access control (RBAC) and multi-factor authentication (MFA) protect user data and transactions to improve security [22].

Additionally, the platform uses cloud functionalities to construct serverless architecture, which ensures scalability and cost effectiveness while handling large transaction loads with ease. Together, these technological developments make the Smart Agricultural Marketplace a scalable, intelligent, and user- focused platform that connects modern agriculture and technology.

VI. IMPLEMENTATION

The implementation of the QR code-based food ordering system follows a modular and systematic approach, ensuring that each functional component is developed, tested, and integrated efficiently. Initially, the project setup is done using Next.js for both the frontend and backend development, providing server-side rendering capabilities, faster performance, and a unified framework. The frontend is designed to be fully responsive and mobile-first, allowing users to easily browse menus, manage carts, and complete checkouts across all device types. The backend services are handled using Next.js API routes, which simplify the creation of secure server-side endpoints for user management, menu data operations, order processing, and payment integration.

The PostgreSQL database is configured to manage structured data such as user profiles, menu items, cart details, and order transactions.

For secure user authentication and session handling, Clerk is integrated, allowing users to log in or register via email, phone, or third- party services like Google or Apple ID, ensuring a secure and seamless onboarding experience. Static QR codes are generated for each dine-in table and takeout point, linking directly to a unique digital menu URL hosted on the platform.

Menu details are fetched dynamically from the PostgreSQL database, ensuring that updates to prices, descriptions, or item availability reflect instantly on the user's interface.

A real-time cart management system is developed, allowing users to add, edit, or remove food items, with automatic updates to cart totals. For checkout, a dynamic QR code generation system is implemented, where each checkout action produces a unique QR code that links the customer's current cart and order. Payment integration is carried out using secure payment gateways like Stripe or Razorpay, ensuring that transactions are quick, reliable, and protected. Upon successful payment confirmation, the system updates the corresponding order status in the PostgreSQL database and simultaneously notifies the restaurant’s kitchen staff for order preparation.

Throughout the development, extensive testing is performed, including unit testing for individual functions, API testing for backend endpoints, and user acceptance testing (UAT) to ensure a smooth, bug-free user experience. Finally, after validation, the platform is deployed on Vercel for hosting, leveraging its serverless capabilities for performance optimization. Post-deployment, regular performance monitoring, security checks, and maintenance activities are scheduled to ensure smooth and scalable operations. Future enhancements such as real-time order tracking, customer feedback systems, and loyalty rewards programs are also planned for continuous improvement of the platform.

VII. EXPECTED OUTCOME

The home page of the QR code-based food ordering system serves as the first point of interaction for the customer. As shown in the screenshot, the design is intentionally kept clean, minimalistic, and user-friendly to ensure easy navigation, especially on mobile devices where users are most likely scanning the QR code. At the top, the navigation bar displays two key menu options — Menu and Cart. These links allow users to quickly view the available food items or check the items they have added to their cart. This ensures a seamless browsing and ordering experience. The central focus of the homepage is the QR Code displayed prominently with the instruction "Scan to Open Menu." This static QR code acts as a gateway for customers — when scanned using any mobile camera or QR code scanner, it redirects users to the digital food menu specific to the restaurant (and optionally, even table-specific menus if multiple codes are configured). This scanning step removes the need for printed menus and speeds up the ordering process.

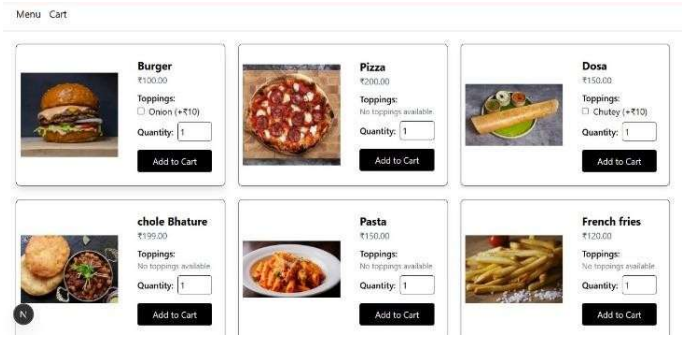

Fig.4 Menu Page

After scanning the QR code on the home page, the user is redirected to the digital menu page, where various food items are displayed in an organized card format. Each card contains an image of the dish, its name, price, available toppings (if any), and a quantity selector. This user-friendly layout ensures a smooth and intuitive browsing experience. Customers can customize their orders by selecting additional toppings, adjusting the quantity, and then adding the item to their cart using the "Add to Cart" button. The menu is dynamic, meaning it fetches real-time information from the PostgreSQL database and is rendered efficiently using the Next.js frontend. The top navigation bar continues to show easy access to the "Menu" and "Cart" pages, allowing customers to either continue exploring more dishes or proceed to checkout at any time. This system promotes a faster, contactless, and highly convenient way for users to place their food orders in the restaurant without the need for physical menus or waiter intervention whwn user click the add to cart button he will be redirect to the cart page.

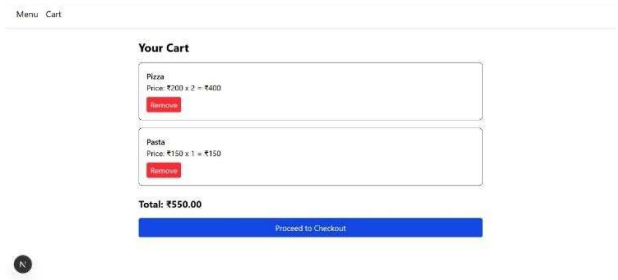

Fig.5 Cart Page

After the user selects food items from the menu, they are redirected to the Cart Page, where they can review and manage their order before checkout. Each item added to the cart is displayed with its name, quantity, individual price, and the total price for that item (calculated as quantity multiplied by unit price). Users have the flexibility to adjust their orders — if they want to remove any item, they can simply click the red "Remove" button beside each dish. Below the list of selected items, the grand total of the order is clearly shown, making it easy for users to understand the final amount they will be charged. The interface remains clean and intuitive, with a prominent "Proceed to Checkout" button, encouraging users to move toward payment. This functionality is managed

through the frontend built using Next.js, and the cart data is dynamically stored and managed, ensuring a seamless and interactive experience. When user click the proceed to check out button it will redirect to the checkout page

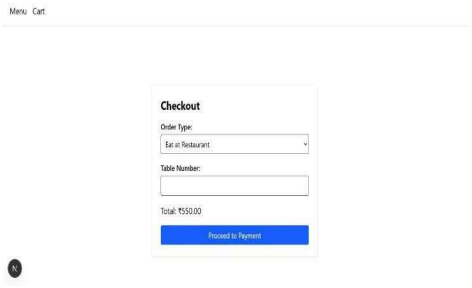

Fig.6 Checkout Page

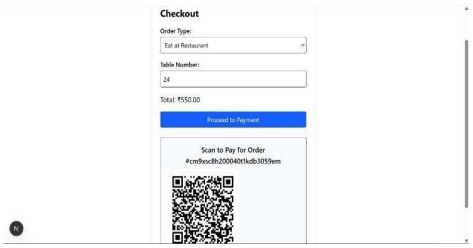

Fig.7 Payment Page

The Checkout Page to finalize their order details. Here, users can choose the Order Type from a dropdown menu, selecting either "Eat at Restaurant" (for dine-in) or "Takeaway" (for picking up the order). If the user selects "Eat at Restaurant," they are required to enter their Table Number to ensure the order is delivered correctly to their table. The page clearly displays the Total Amount based on the items added to the cart. Once the necessary details are filled in, users can click on the prominent "Proceed to Payment" button, which will lead them to the next step of payment. This step is crucial as it links the customer's order with the correct table for dine-in service or marks it for pickup if selected.

After selecting the Order Type (for example, "Eat at Restaurant") and entering the Table Number (like 24), the user proceeds to the final stage of the ordering process — the Payment Section. Once the user clicks on "Proceed to Payment," a dynamically generated QR code appears on the screen. This QR code is linked to a unique Order ID (e.g., #cm9xsc8h200040t1kdb3059em), ensuring that the restaurant system can correctly identify and process the payment for that particular table and order. The QR code can be scanned using any standard UPI or digital payment app, allowing the customer to quickly complete their payment securely. This seamless transition from checkout to payment enhances the user experience by making the dine-in or takeaway process highly efficient and contactless. The backend, powered by Next.js and PostgreSQL, updates the order status once the payment is successful, while Clerk authentication ensures the user's session and order details are secure throughout.

VIII. Conclusion

In conclusion, the development of the Intelligent Agriculture Marketplace successfully demonstrates how technology can bridge the gap between farmers and modern e-commerce platforms. By integrating features such as user authentication (login and registration), a streamlined product catalog with essential agricultural items, a dynamic shopping cart, and an intuitive order confirmation system, this platform simplifies agricultural trade for farmers and buyers alike. The user- friendly interface, secure order processing, and informative product listings ensure ease of use and reliability. This solution empowers farmers by giving them direct access to a wider market, reduces the need for intermediaries, and promotes fair pricing and transparency. Overall, the project fulfills its objective of creating a digital platform tailored to the unique needs of the agricultural community, paving the way for smarter, tech-driven farming commerce in the future.The Agricultural E-commerce System represents a comprehensive approach to managing agriculture from pre- production to post-production. By leveraging multiple technologies, it streamlines the flow of resources, information, and products, making agriculture more efficient and accessible. This integration ultimately aims to improve productivity, reduce transaction costs, and enhance market accessibility for agricultural products.

References

[1] Mohamed, S. A.'s study, "Dynamic QR Code Generation for Personalized Ordering Experiences" (2024)
[2] Chatterjee, P., and Ghosh, A.'s study, "Implementing Secure Authentication in Web-Based Ordering Systems Using Clerk API" (2023)
[3] Raj, S., and Sinha, P.'s study, "Challenges in Developing Cross- Platform Food Ordering Applications" (2023)
[4] Nguyen, T. H., and Le, M. Q.'s study, "Optimizing Checkout Processes in Online Food Ordering Applications" (2024)
[5] Sato, M., and Nakamura, T.'s study, "Dynamic Menu Customization Based on User Behavior in Food Ordering Apps" (2024)

[6] Brown, L., and Williams, E.'s study, "Improving Performance of Next.js Web Applications for Food Services" (2023)
[7] Sagaf, A. F. A., and Ibrahim, I.'s study, "Development of a Mobile Web-Based Food and Beverage Ordering Application in a Youth Cafe with QR Code Technology" (2025)
[8] Nikose, A., Hatwar, A., Nikose, A., Adikane, D., and Gaharwar, K.'s study, "Cafeteria Food Ordering System Using QR Code" (2023)
[9] Othman, M., Mohamad Sakri, I. N., Muhd Zain, N., Paidi, Z., Che Jan, N. Y., and Kamarudin, N. K.'s study, "Design and Development of Groceries Ordering System with QR Code" (2023)
[10] Parsikun, S. N., and Sedek, K. A.'s study, "Responsive Web-Based Cafe Food Ordering System Using Bootstrap and QR Code" (2023)
[11] Christopher, G., Hoo, L. C., and Kaarthik, K.'s study, "Design and Development of a QR Code-Based Grocery Ordering System" (2024)
[12] Pereira, F., and Santos, R.'s study, "Impact of QR Code-Based Ordering Systems on Restaurant Efficiency" (2023)
[13] Srinivasan, P., and Iyer, R.'s study, "Dynamic QR Code Generation for Personalized Ordering Experiences" (2024)
[14] Krishnamurthy, S., and Venkatesan, R.'s study, "The Evolution of Digital Menus: A UI/UX Perspective" (2023)
[15] Bhandarkar, M., and Choudhury, A.'s study, "Streamlining the Checkout Experience in QR-Based Ordering Systems" (2024)
[16] Sharma, K., and Verma, R.'s study, "Role of UI Aesthetics in Customer Loyalty for Food Ordering Websites" (2023
[17] Patel, A., and Mehta, P.'s study, "Use of PostgreSQL for Efficient Backend Management in Ordering Systems" (2023)