

A Project Report On

**“Cloud Kitchen website”**

**(Heaven’s kitchen)**

**SUBMITTED BY**

Aniket Gurav, Abhijeet Gund

Sanjana Patel, Mansi Zanjale.

(TY-BBA-CA)

**UNDER GUIDANCE**

PROF.MONALI MEGHAL

**SUBMITTED TO**

SURYADATTA COLLAGE OF MANAGEMENT INFORMATION

RESEARCH AND TECHNOLOGY (SCMIRT)

PUNE-411021

**IN FULLFILLMENT OF**

BATCHELOR OF BUSINESS ADMINISTRATION

(COMPUTER APPLICATION)

2024-25



**\*Acknowledgement\***

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the completion of our minor project work. Whatever we have done is only due to such guidance and assistance.

we would not forget to thank BBA-CA coordinator and Project guide Asst.Prof. Monali Meghal, we are extremely grateful to her for providing such a nice support and guidance though she had busy schedule managing the college dealings.

Head of BCA Department Asst.prof. Deepak Singh and all other Assistant professors of SCMIRT Department, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We are thankful and fortunate enough to get support and guidance from all Teaching staffs of Bachelor of Business Administration (Computer Application) Department which helped us in successfully completing our project work. Also, we would like to extend our sincere regards to all the non-teaching staff of Bachelor of Business Administration (Computer Application) Department for their timely support.

|  |  |
| --- | --- |
| **Sr.no** | **Index Name** |
| 1 | **Introduction.**   * MOTIVATION * PROBLEM STATEMENT * PURPOSE * LITERATURE SURVEY * PROJECT SCOPE AND LIMITION |
| 2 | * **SYSTEM ANALYSIS** * EXISTING SYSTEM * SCOPE & LIMITATION OF EXISTING SYSTEM * PROJECT PERSPECTIVE, FEATURE * STACKE HOLDERS * REQUREMENT ANALYSIS (Functional Requirement, Performance Requirement, Security Requirement) |
| 3 | **SYSTEM DESIGN**   * SYSTEM MODULE * DATA MODEL * USER INTERFACE |
| 4 | **IMPLEMENTATION DETAILS**  • SOTWARE / HARDWARE SPECIFICATION  **Outputs and report Testing**  Test plan, Black box Testing, White box Testing and Results |
| 6 | **CONCLUSION AND RECOMMENDATIONS** |
| 7 | **FUTURE SCOPE** |
| 8 | **REFERENCE** |

**INDEX**

5

***Heavens Kitchen***

Our website aims to provide a delightful online experience for patrons, offering a virtual glimpse into our cozy establishment. Through a user-friendly interface, visitors can explore our menu, learn about our commitment to quality ingredients and sustainability, and stay updated on upcoming events and promotions.

Explore our diverse range of offerings, from signature coffees brewed from the finest beans sourced globally, to freshly baked pastries and sandwiches prepared with locally sourced ingredients.

Each dish is a testament to our commitment to quality, flavour, and sustainability.

Ordering with us is simple and convenient. Navigate our user-friendly website to browse our menu, place your order with just a few clicks, and track it in real-time as it makes its way to you.

**Technology: -** HTML, CSS, JavaScript, PHP and MYSQL

**Team members: -** Aniket Gurav, Abhijeet Gund,

Sanjana patel, Mansi Zanjale.

Aniket Gurav Mansi Zanjale

Abhijeet Gund Sanjana Patel

**\*Introduction: \***

A cloud kitchen, also known as a virtual kitchen, is a commercial facility set up solely for the preparation of delivery-only meals. With the rise in demand for food delivery services, cloud kitchens have emerged as a cost-effective and efficient solution for restaurateurs to reach a broader customer base without the expense of a physical dining area.

A cloud kitchen operates without a dine-in facility, focusing exclusively on fulfilling online food orders. It leverages digital platforms for order management, customer interactions, and kitchen operations.

Technology is integral to the success of cloud kitchens. It allows seamless integration of customer orders, kitchen workflows, and delivery logistics. A well-designed website can enhance the operational efficiency of cloud kitchens by providing a direct line of communication between the kitchen and customers, managing inventory, and streamlining workflows.

Project Summary:

This project involves developing a cloud kitchen website to provide a digital platform where customers can easily order food. The website will include features like a menu display, and a backend system for managing orders and updating the menu. This platform will aim to enhance the operational efficiency of the kitchen while improving the user experience for customers

**\* Motivation: \***

Personal Motivation:

The rapid growth of cloud kitchens caught our attention as a developer and student of business. Seeing how this model has transformed the restaurant industry, I wanted to create a technological solution that would help small and medium-scale cloud kitchens manage operations and orders more efficiently.

Industry Trends:

The food delivery market is booming, fueled by customer demand for convenience and variety. Cloud kitchens are capitalizing on this trend by offering low overhead costs and greater reach through online platforms. This growing sector requires robust and scalable technology to manage the influx of orders, customer data, and delivery coordination.

Web development, cloud computing, and mobile app integration are key areas that can revolutionize cloud kitchens. With the right technology stack, a cloud kitchen can operate smoothly, automate many tasks, and cater to a growing customer base. The project aims to implement these advancements to solve the operational issues faced by cloud kitchens.

**\* Problem Statement: \***

Challenges for Traditional Restaurants: Traditional brick-and-mortar restaurants are limited by physical space, staff availability, and high operational costs. They also face competition from digital platforms that can deliver food more conveniently.

**Customer Convenience:**

Customers today demand more convenience when ordering food. Traditional models don't offer the level of convenience that cloud kitchens and online platforms can provide. Hence, there's a need for a streamlined online system where customers can easily browse, select, and order meals with minimal hassle.

**Cloud Kitchen Challenges:**

Despite their operational advantages, cloud kitchens often face challenges in managing online orders, customer feedback, and real-time inventory updates. Additionally, some cloud kitchens lack an efficient, dedicated online platform for handling all these aspects in a user-friendly way.

**\* Purpose/Objective and Goals: \***

**Purpose:**

The main purpose of this project is to develop a fully functional cloud kitchen website that simplifies online ordering for customers and enhances the operational efficiency of cloud kitchens.

**Goals:**

1. User-Friendly Interface: Create a customer-friendly website where users can view menus, select items, and place orders seamlessly.

2. Order Management System: Provide a backend system for kitchen staff to track orders, update inventory, and manage menu items.

4. Customer Accounts and Feedback: Enable users to create accounts, view order history, and provide feedback on their dining experience.

5. Mobile Responsiveness: Ensure that the website is responsive across various devices, offering the same level of convenience on smartphones as on desktops.

6. Payment Integration: Integrate secure payment gateways for online transactions.

**\* Literature Survey: \***

**Existing Solutions:**

Platforms like Swiggy, Zomato, and UberEATS dominate the cloud kitchen and food delivery markets, offering robust order management, payment integration, and customer interface. However, these platforms charge high commission fees, which can strain the profitability of smaller kitchens. Additionally, they don't allow kitchens full control over the customer experience.

**Technology Stack:**

The project will leverage modern web technologies:

**Frontend:** HTML and CSS, JS for an interactive user interface.

**Backend:** PHP for managing orders, user authentication and data storage.

**Database:** A relational database like MySQL for managing customer orders, inventory, and feedback.

**Research Studies:**

Studies have shown that cloud kitchens can reduce operating costs by up to 60% compared to traditional restaurants. The efficient use of digital platforms enhances customer experience while minimizing delays and order mismanagement. Research papers suggest that cloud kitchens are particularly well-suited for urban areas with high demand for quick, affordable meals

**\* Project Scope and Limitations: \***

**Scope:**

* The platform will allow customers to browse menus, place orders.
* Admins (kitchen staff) will have a backend dashboard to manage orders, update the menu, and track inventory.
* Secure payment gateways will be integrated to ensure seamless online transactions.
* The project will focus on a web-based solution but will include a responsive design to support mobile users.
* The architecture will be scalable, allowing for the addition of multiple kitchens or branches in the future.

**Limitations:**

* The initial version will focus on basic features like order management and menu display. Advanced features like delivery management (integrating third-party logistics) will be developed in future phases.
* The website will be developed initially for desktop and mobile web browsers, but a native mobile app is not part of the current scope.
* Real-time order tracking will be limited to notifications once the order is prepared and dispatched but won't include live delivery tracking in the initial release.

**\* System Analysis: \***

**Existing System:**

Current Food Delivery Solutions:

The existing systems, such as Swiggy, Zomato, UberEats, and other food delivery platforms, provide restaurants and cloud kitchens with a way to list their menus online and receive orders. However, these platforms take a significant commission from each order and often don't give kitchens full control over their customer relationships or operations. Additionally, many smaller cloud kitchens rely on manual processes (e.g., phone orders or third-party delivery platforms), leading to inefficiencies in order management and customer engagement.

Challenges in Current Systems:

- High commission fees for third-party delivery platforms.

- Limited control over customer data and experience.

- Inefficiencies in handling multiple online orders, leading to delays.

- Dependence on external delivery partners.

- Lack of real-time updates for customers and kitchen staff.

**\*Scope and Limitations of Existing Systems\***

\*Scope of Current Systems\*:

- Wide Customer Base: These systems offer cloud kitchens access to a vast customer base through third-party platforms.

- Logistics Support: They often handle logistics (delivery) so that cloud kitchens don't need to worry about the transportation aspect.

- Marketing and Promotions: Some platforms help with marketing efforts by listing restaurants in search results or offering promotions.

\*Limitations\*:

- High Fees: Third-party platforms charge high commissions (ranging from 15% to 30%) per order, impacting profitability.

- Limited Control: Cloud kitchens cannot fully control their branding, customer relationships, or user experience as they are dependent on the platform’s ecosystem.

- Scalability Issues: For smaller cloud kitchens, it may be difficult to scale using these platforms due to the overhead costs and lack of direct engagement with customers.

- Data Ownership: Cloud kitchens often do not have access to the full data on customer preferences, making it difficult to target their marketing efforts or improve services based on feedback.

\*Project Perspective\*

This cloud kitchen website project is designed to provide cloud kitchens with full control over their online presence, order management, and customer interactions. It addresses the gaps in existing solutions by offering an in-house platform that does not require reliance on third-party delivery services or charge high commission fees.

\*Key Perspectives\*:

- For Cloud Kitchens: A solution that enables easy menu management, real-time order updates, and customer feedback tracking, all under their control.

- For Customers: A seamless, user-friendly platform where they can browse the menu, place orders, and track their food in real-time without interacting with a third-party service.

- For Growth: A platform that can scale to support multiple kitchens, helping businesses expand with lower operational costs.

**\*Features\***

**Customer Features:**

- Menu Browsing: Users can easily browse the menu, filter items, and view details.

- Order Placement: Simple, secure order placement with customizations for meals.

- User Accounts: Customers can create accounts to save preferences and view order history.

- Feedback System: After each order, users can provide feedback and rate their experience.

**Admin Features (For Cloud Kitchen Staff) :**

- Order Management: View, update, and track all incoming orders in real time.

- Menu Management: Add, modify, or remove items from the menu with a simple interface.

- Inventory Tracking: Track inventory levels and receive alerts when ingredients are low.

- Reports and Analytics: Generate reports on sales, customer behaviour, and performance to optimize operations.

- Customer Relationship Management: Manage customer feedback and preferences to improve services and personalize marketing efforts.

**\*Stakeholders\***

- Customers: The end-users who will place food orders via the website.

- Cloud Kitchen Owners/Operators: The primary stakeholders who will use the platform to manage orders, inventory, and customer interactions.

- Admin/Staff: Kitchen staff who will handle order processing, updating menus, and managing the backend system.

- Delivery Personnel: If the cloud kitchen uses its own delivery service, delivery staff will need access to real-time order updates and tracking information.

- Payment Gateway Providers: Third-party providers like PhonePay or GooglePay who will handle payment processing securely

**\*Requirement Analysis\***

**Functional Requirements:**

- Customer Registration/Login: Allow users to create accounts and log in.

- Order Placement: Users should be able to place orders seamlessly and view available menu options.

- Payment Integration: Securely process payments via multiple payment gateways.

- Inventory Management: The kitchen should be able to track stock levels and manage them efficiently.

- Reporting and Analytics: The system should provide key insights into sales and customer behaviour.

**Non-Functional Requirement:**

- Scalability: The system should be able to handle an increasing number of users and orders as the business grows.

- Security: Sensitive data (e.g., payment details, user data) should be encrypted, and the system should adhere to security standards like PCI-DSS for payment processing.

- Usability: The website must be user-friendly, with a focus on simplicity for both customers and kitchen staff.

- Reliability: The system should be able to handle heavy traffic without crashing, and downtime should be minimal

**Security Requirements**

- Data Encryption: All sensitive data, such as customer payment information and personal details, should be encrypted using SSL/TLS encryption to ensure secure communication.

- User Authentication: Implement secure login mechanisms using industry standards like OAuth2 or JWT (JSON Web Token) to authenticate users.

- Role-Based Access Control: Ensure that only authorized kitchen staff/admins have access to sensitive backend features like inventory management or order tracking.

- Data Backup: Regularly back up all system data (orders, customer details, etc.) to prevent loss in case of system failures.

- Payment Security: Ensure that payment transactions comply with PCI-DSS standards and use trusted, secure payment gateways.

- Account Protection: Protect customer accounts using strong password policies and implement multi-factor authentication (MFA) for added security.

- Audit Logs: Maintain audit logs of admin activities (such as menu updates or order modifications) to trace any potential misuse of the system.

**\* System Model: \***

**DFD:**

Food order

Orders for food

Kitchen

1.0

Processing

Customer

of an order

Recepit

Sold

items

Inventory data

3.0

Update inventory file

2.0

Update sold items file

Formatted

Inventory database

Sold item data

Formatted

Database of sold items

data

Inventory

4.0

Generate mgtn report

Info. about daily sold items

Info. about daily inventory

depletion amount

and amount

Restaurant

manager

management

Report

**E R Diagram:**

View id

Email

PName

PID

FName

Password

Price

User mst

Products mst

Image

have

Quantity

have

CCV

Price

Order mst

Card

OID

Payment mst

has

PNmae

Email

PID

Bank

Email

**Use Case Diagram:**

Registration

Login

Navigate Menu

Customer

Select Item

Add Item

Admin

Remove Item

Review Item

Pay for Order

Update Menu

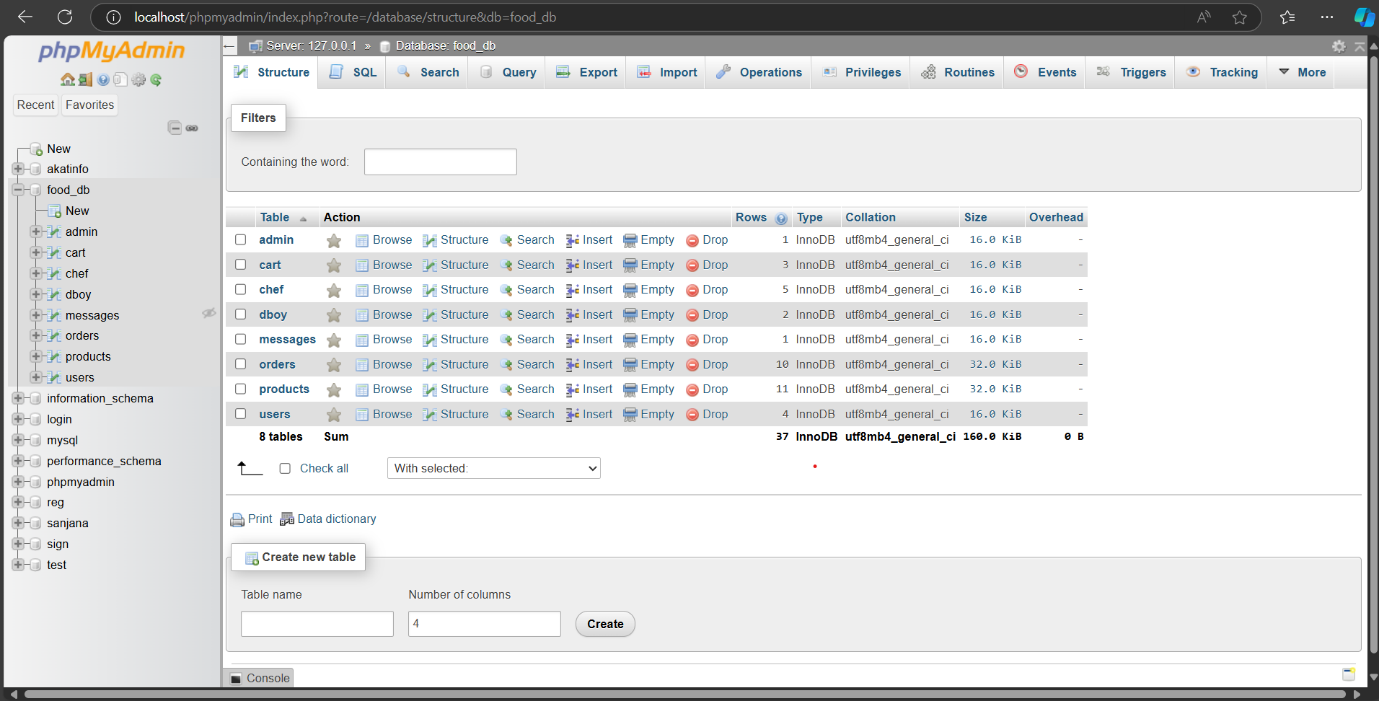
Employee

Receive for Order

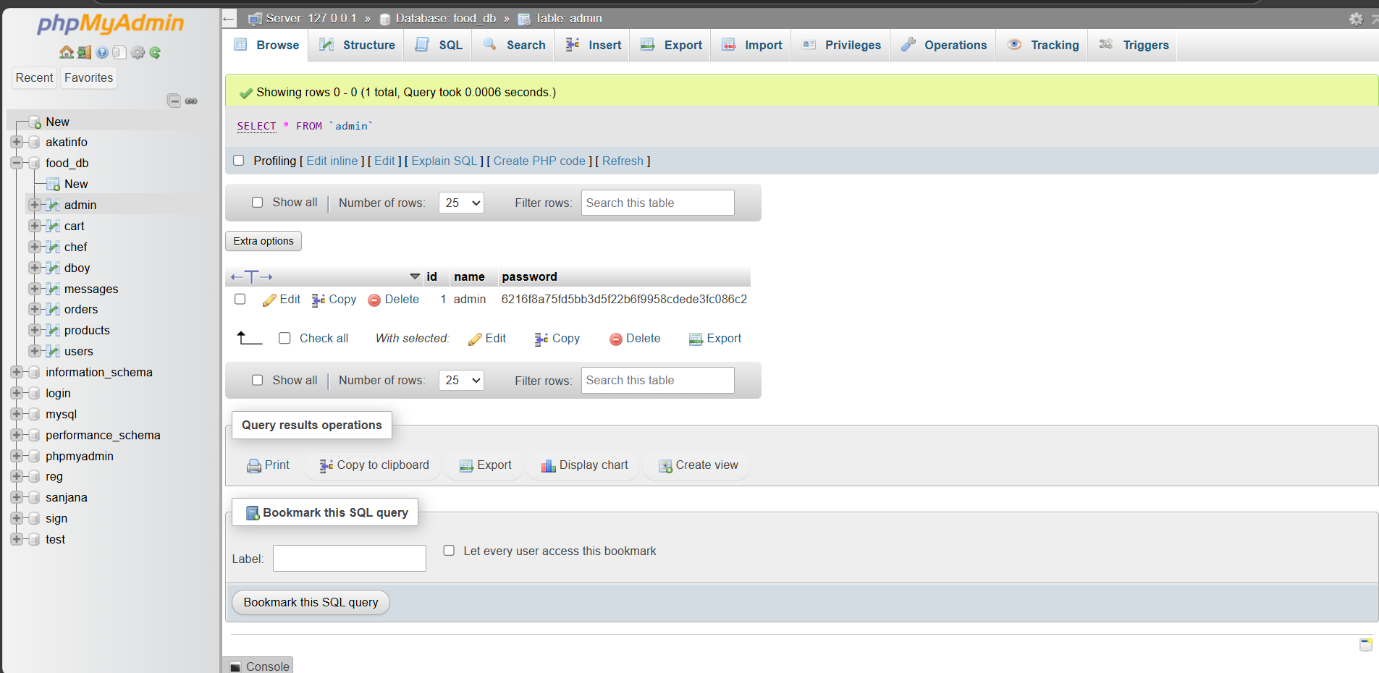
Check Out

**\* Data Model: \***

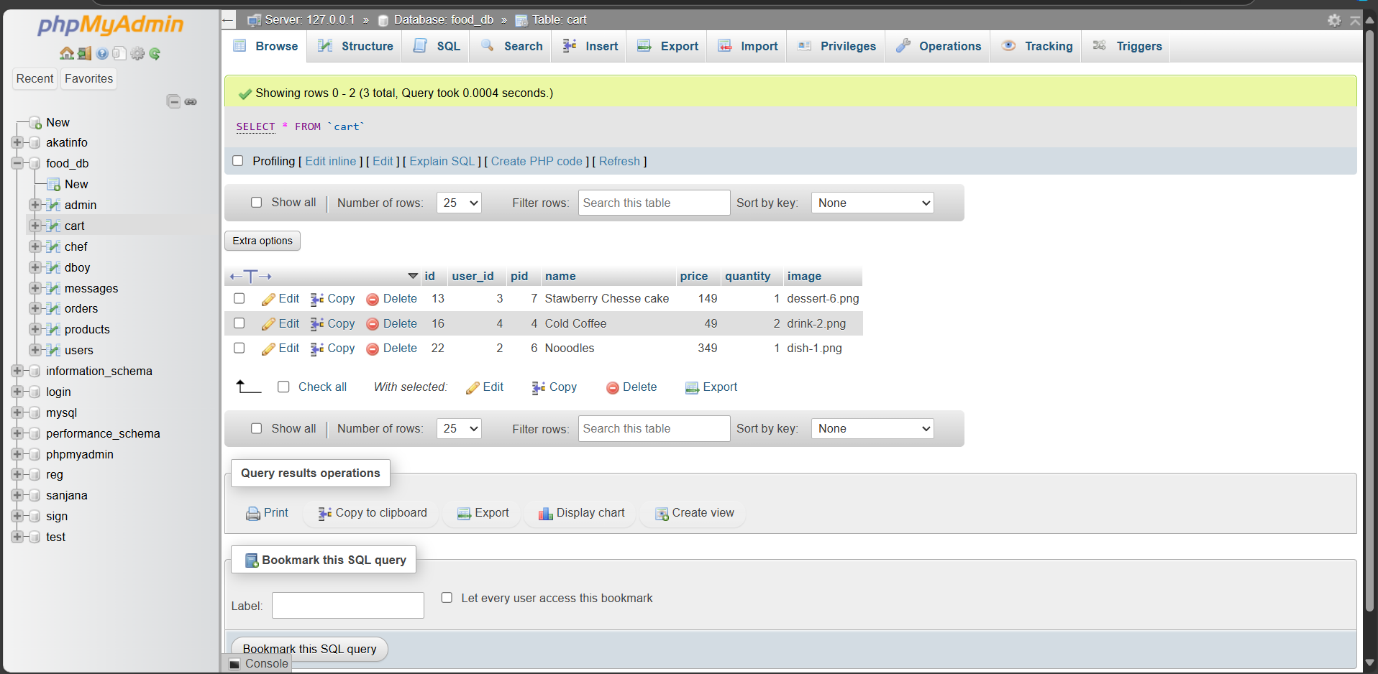
Database:



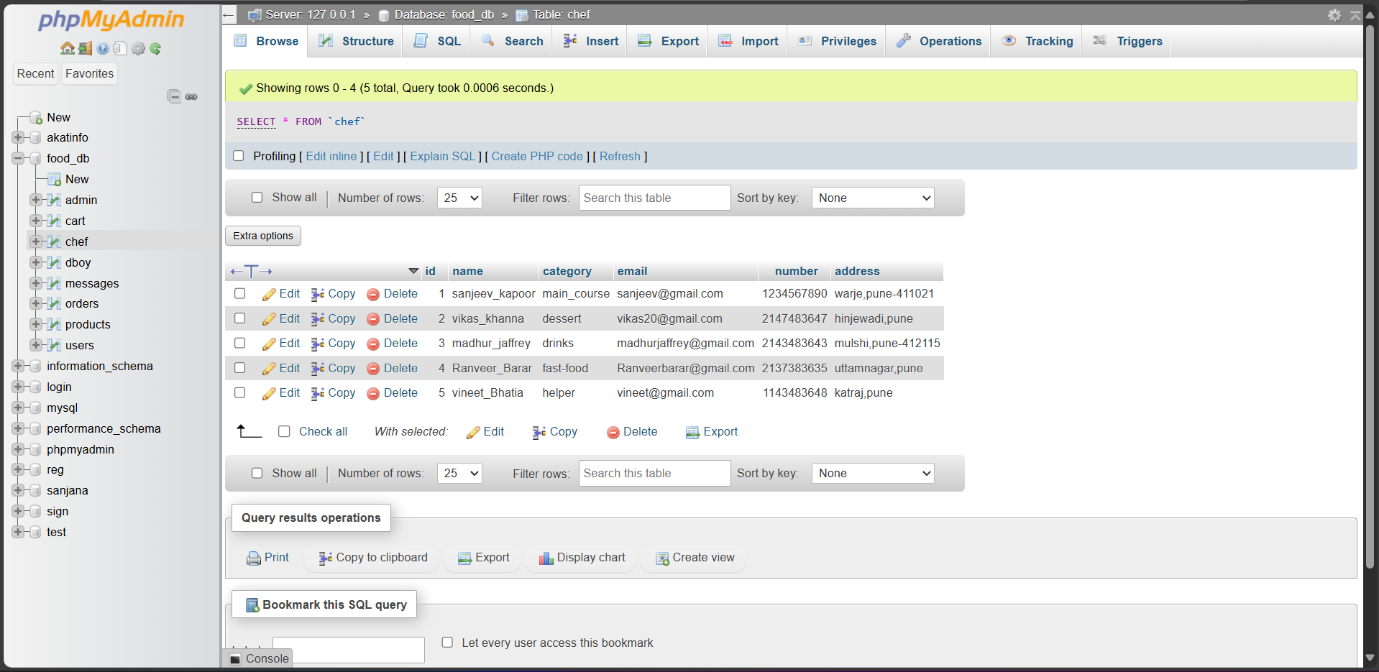
Admin Table:



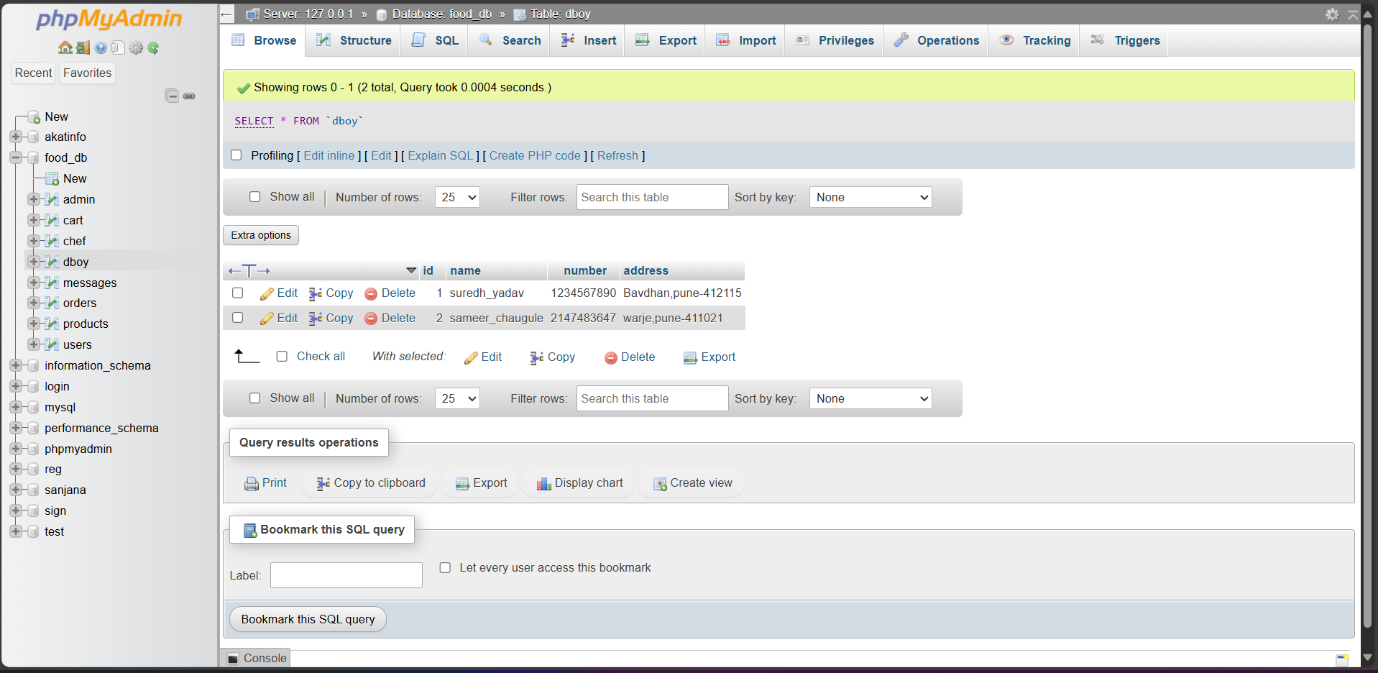
Cart Table:



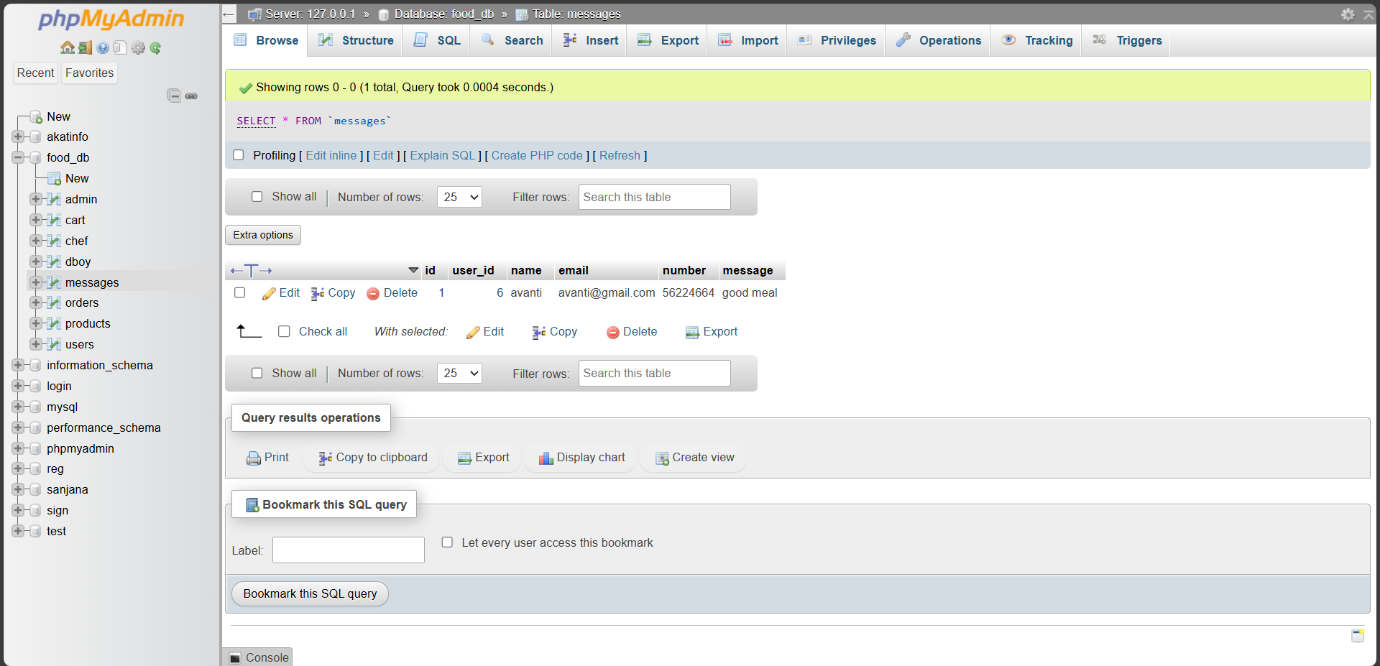
Chef Table:



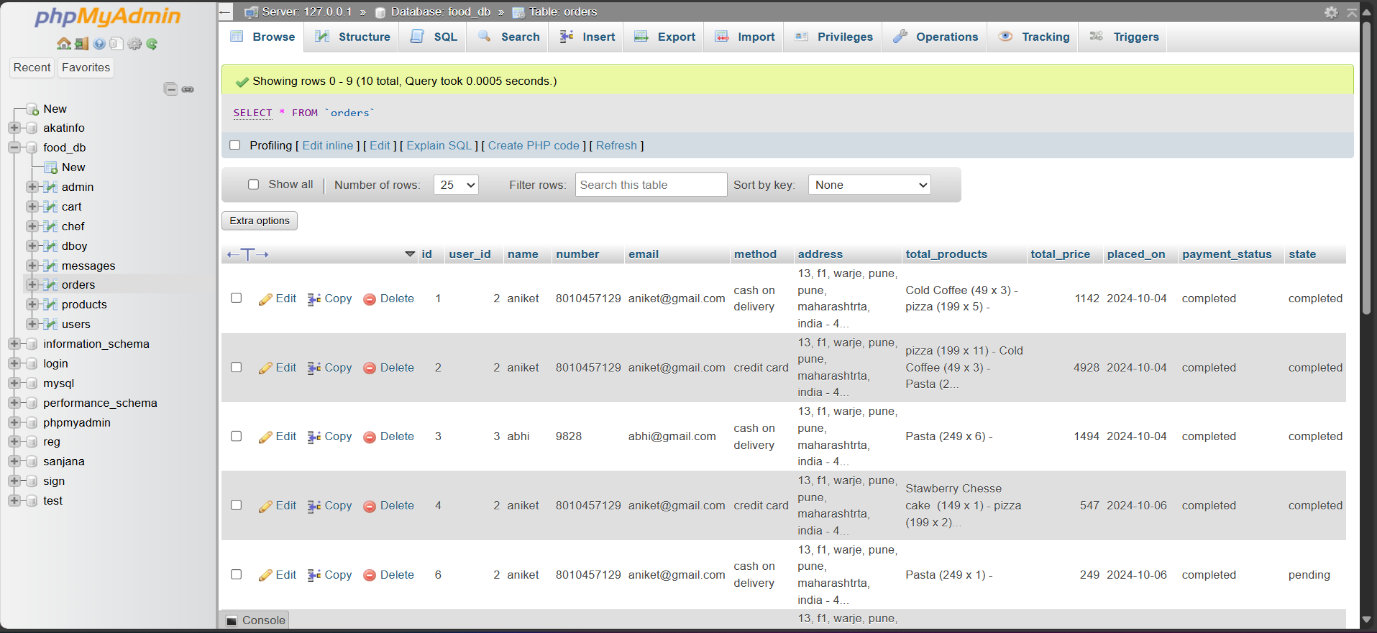
Dboy Table:



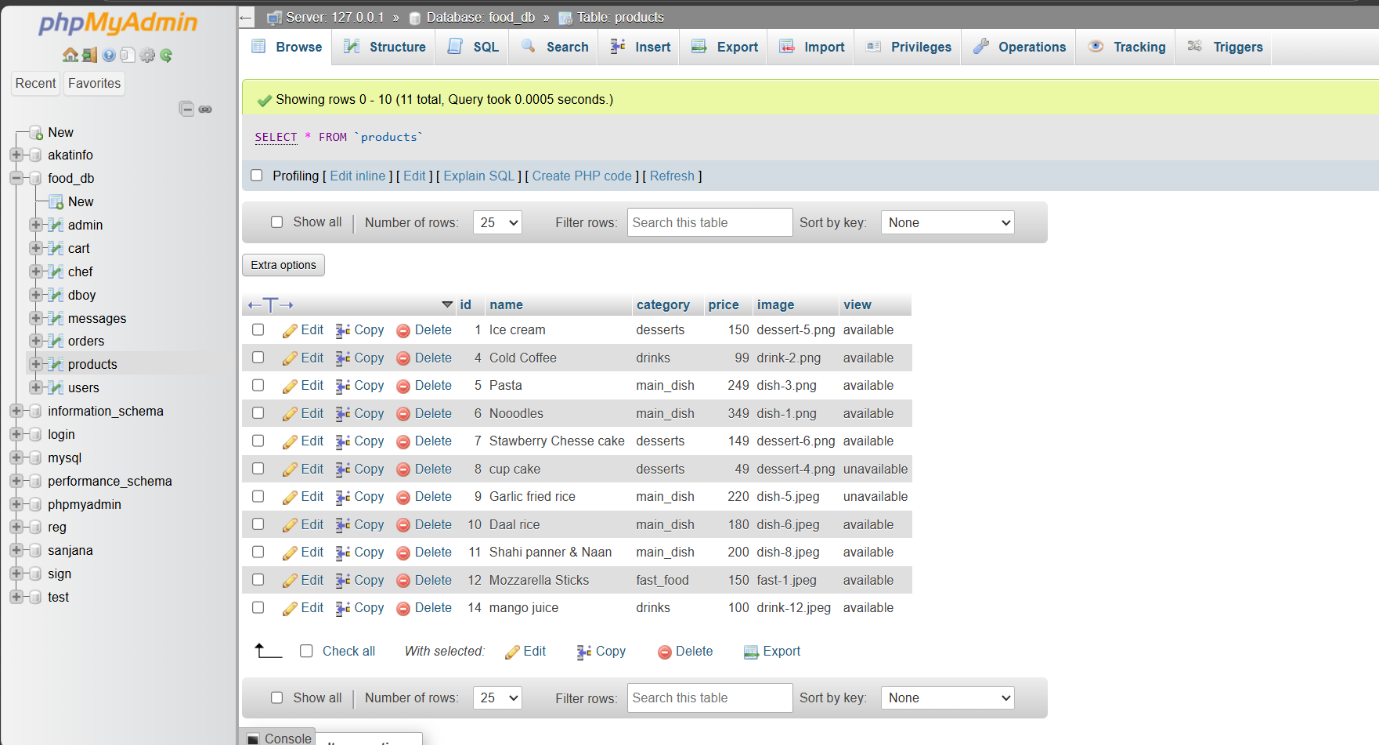
Messages Table:



Orders Table:



Products Table:

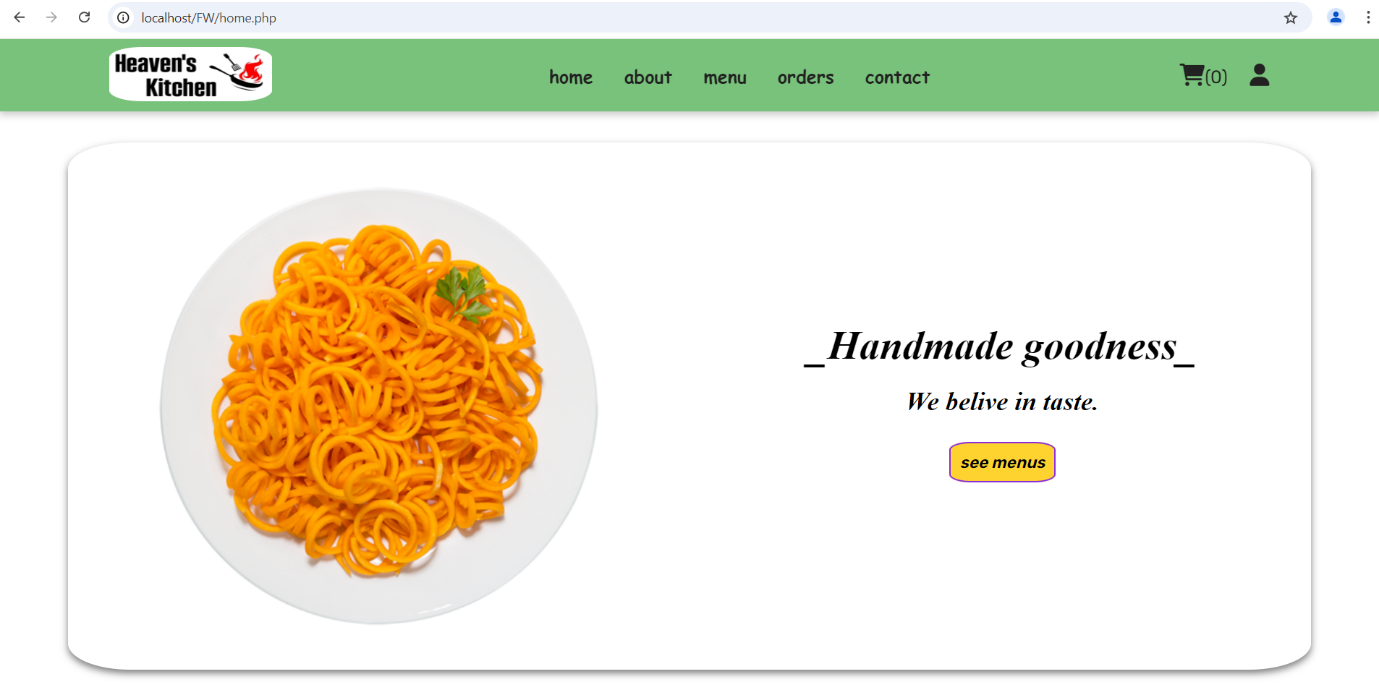


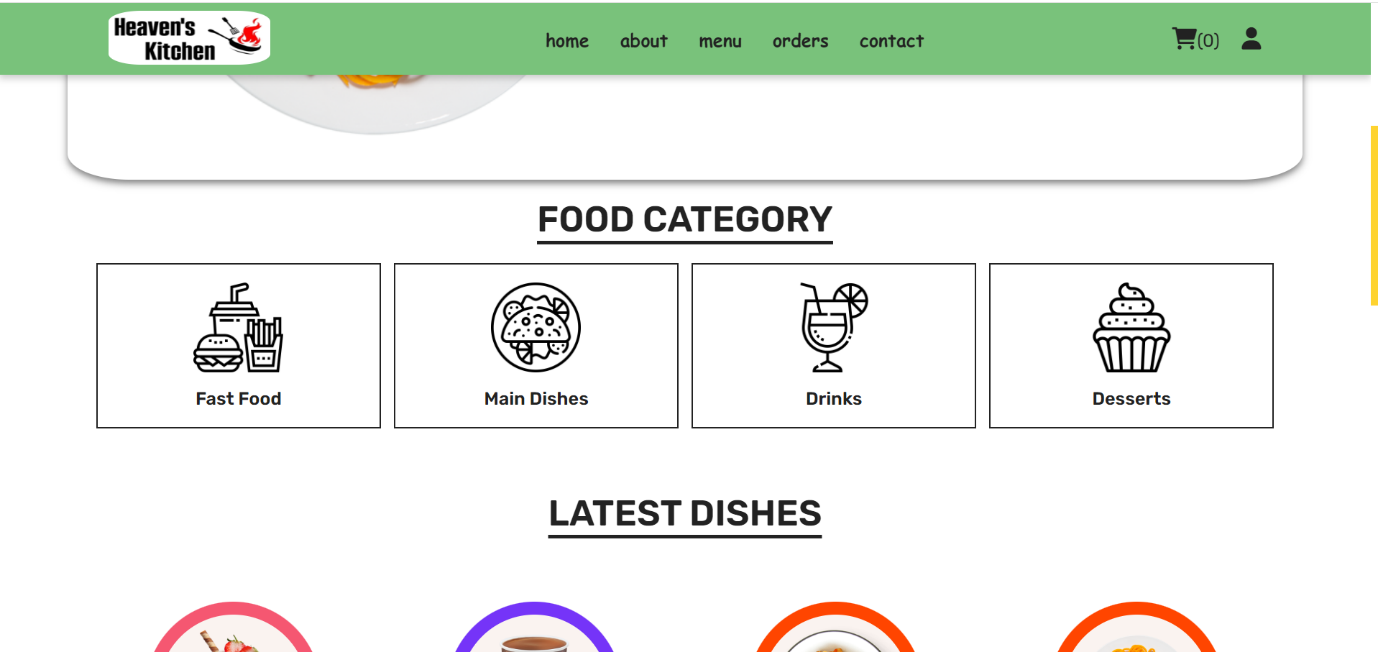
Users Table:



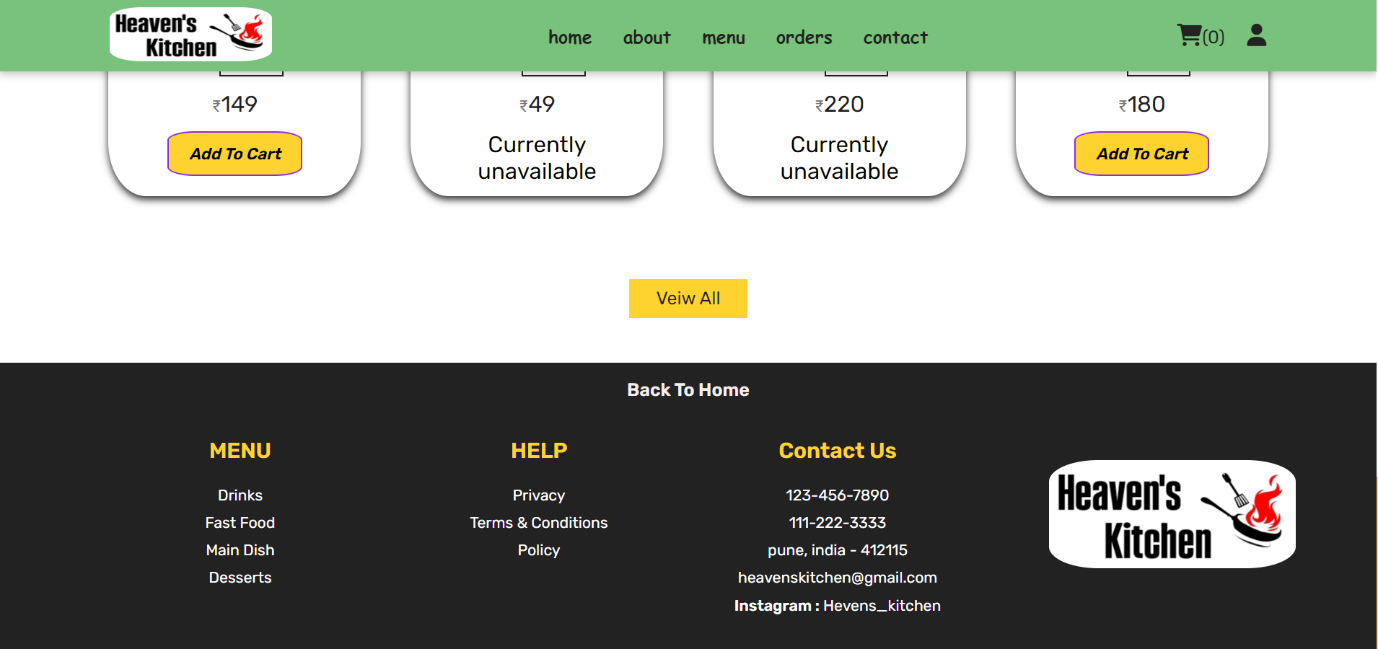
**\* User Interface: \***

Home Page:

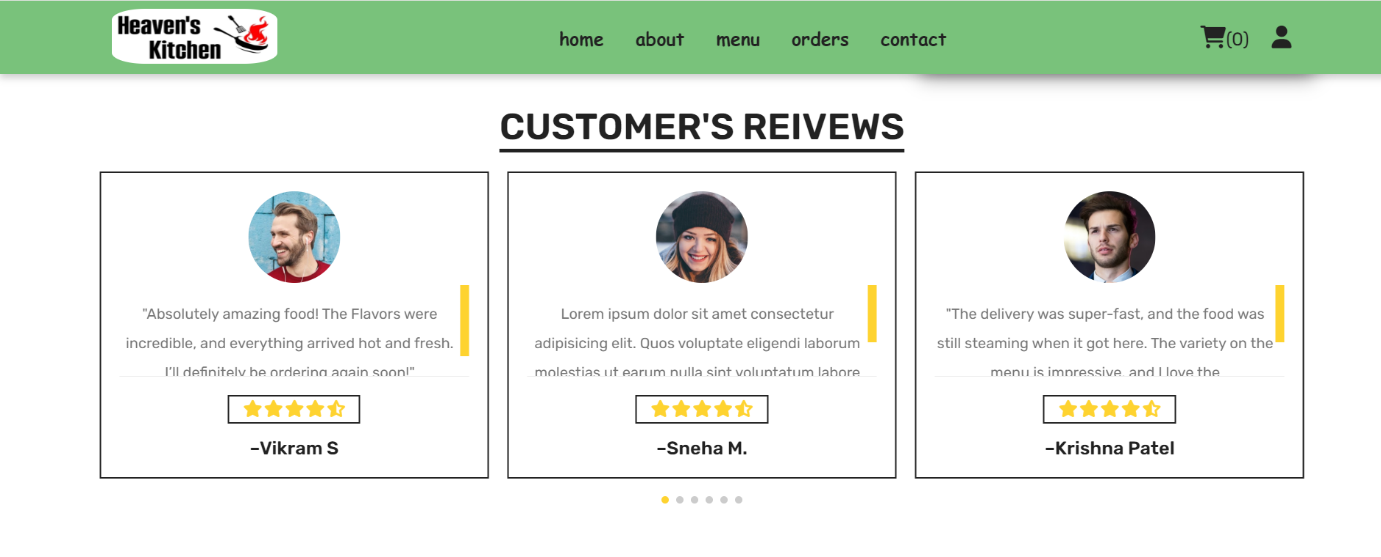




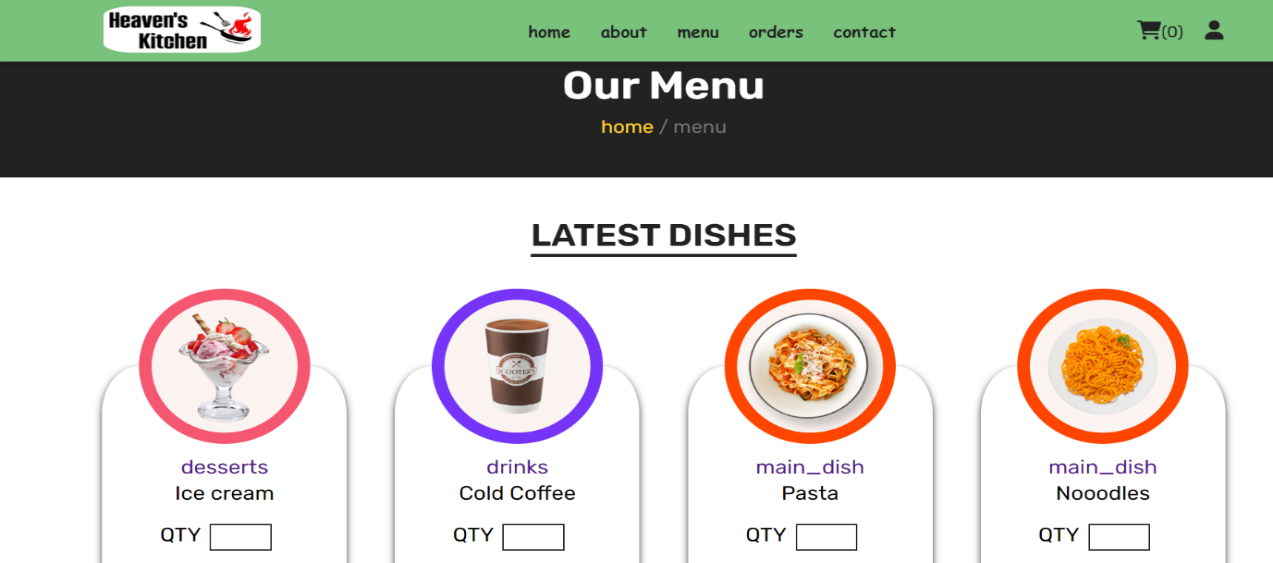




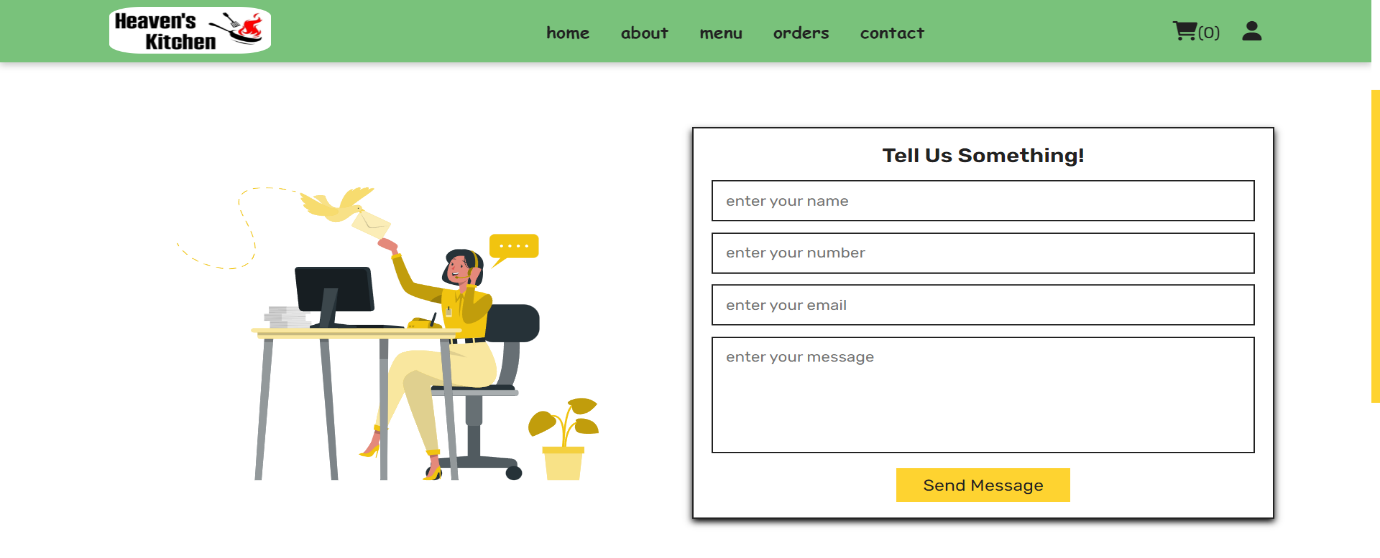
Review Section:



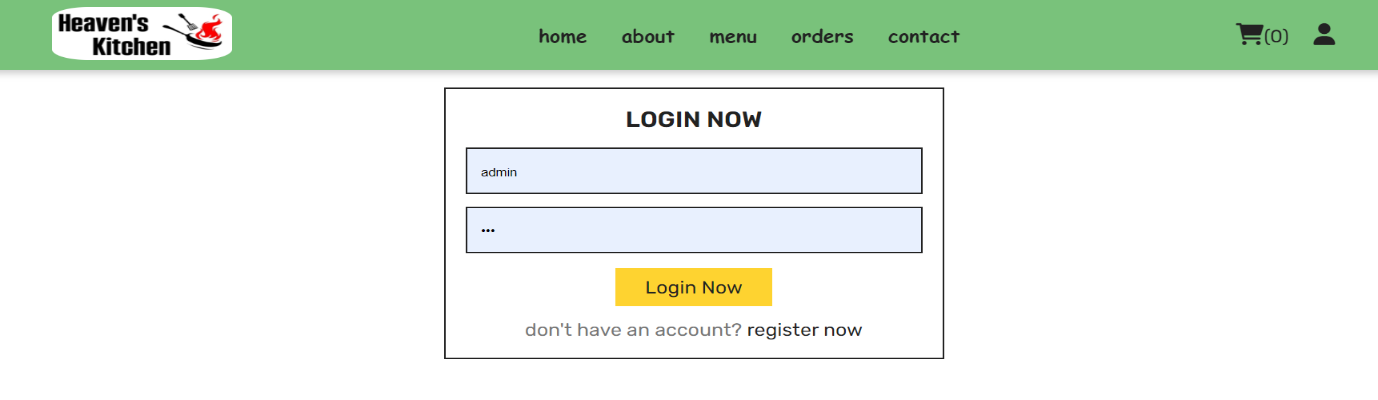
Menu Page:



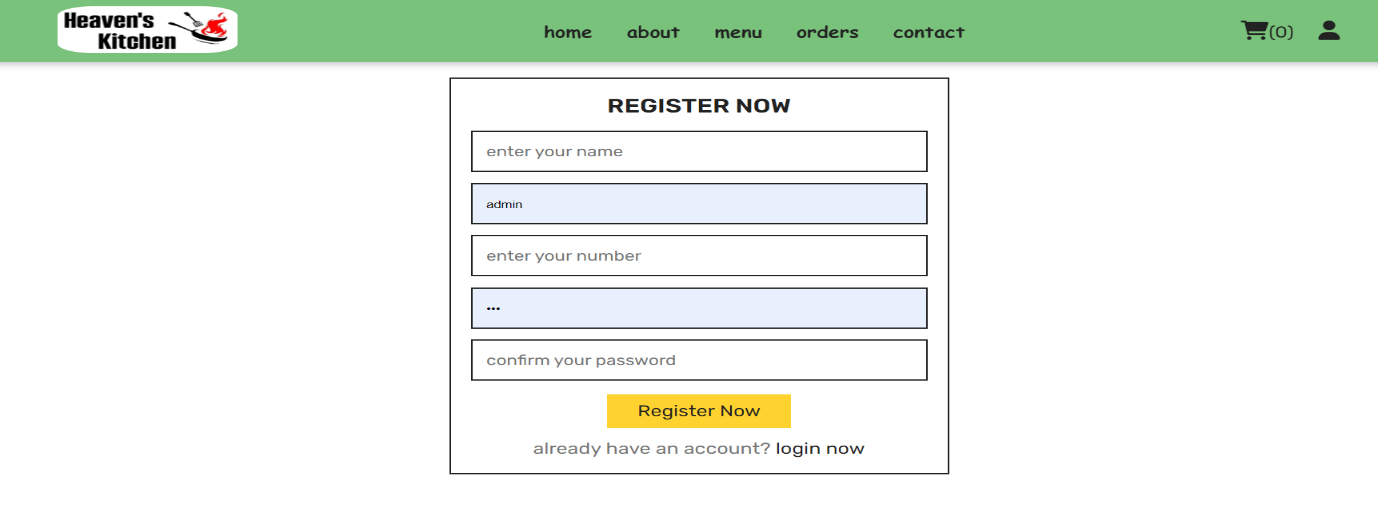
Contact Page:



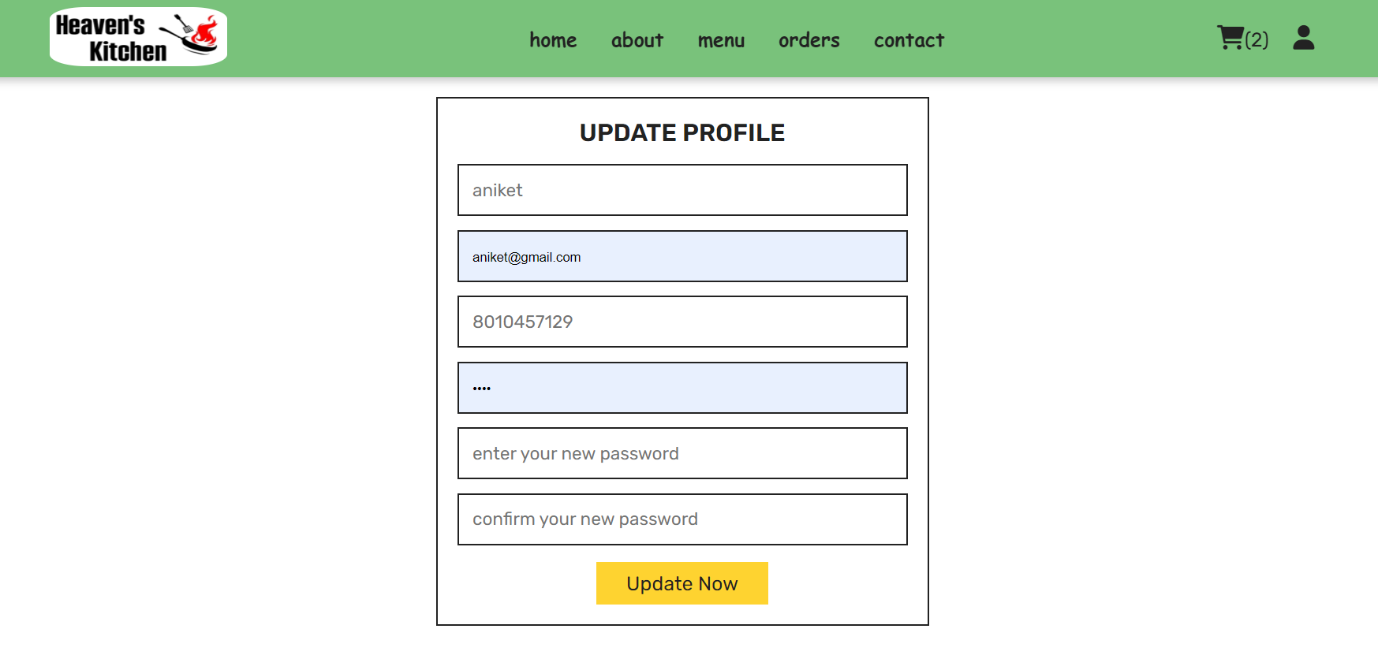
Login Page:

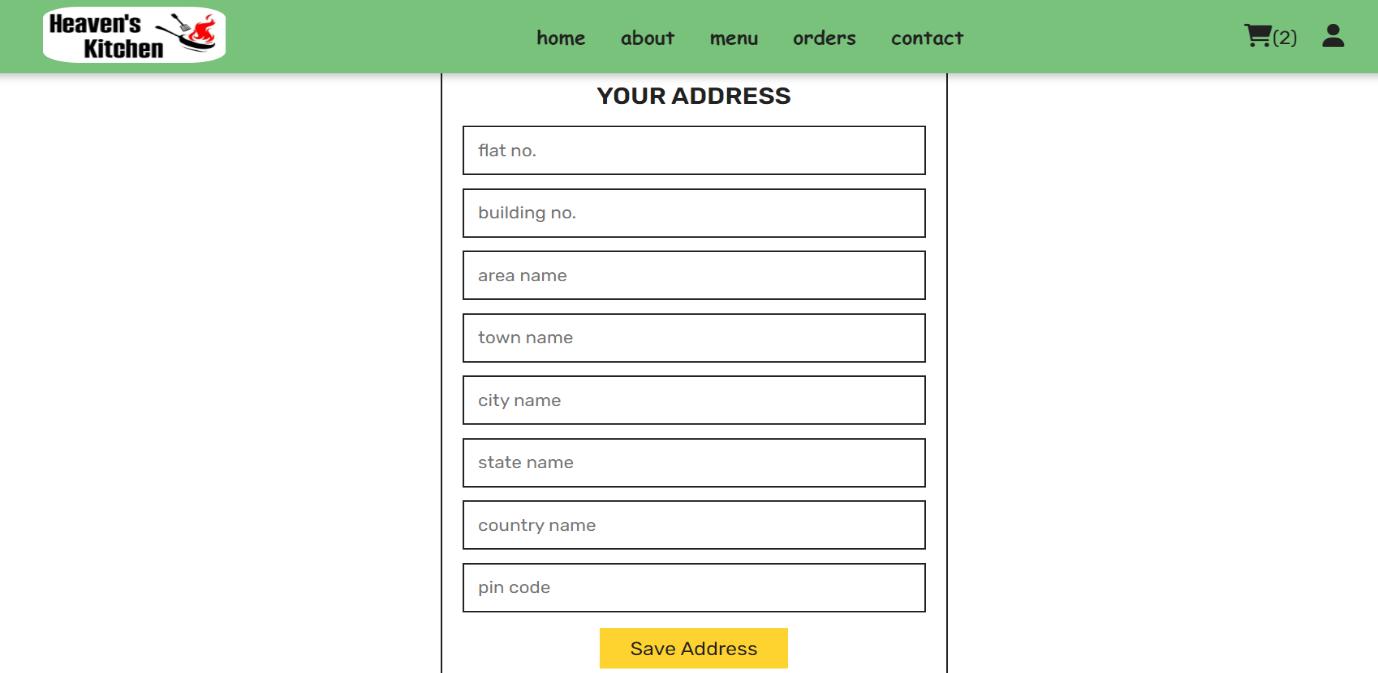


Register Page:

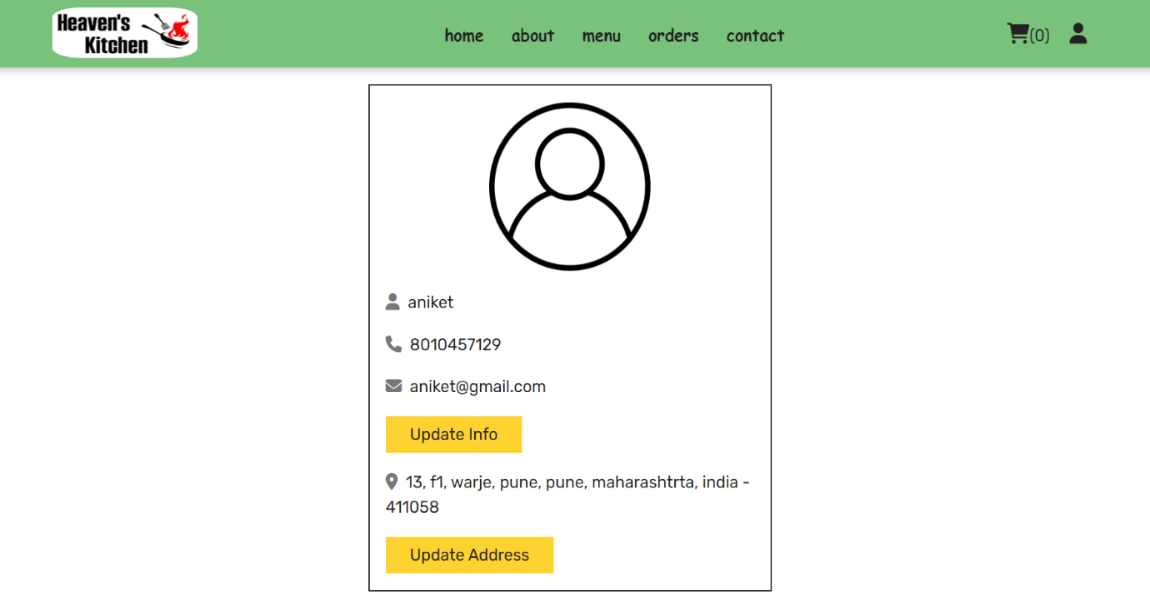


Update Profile:

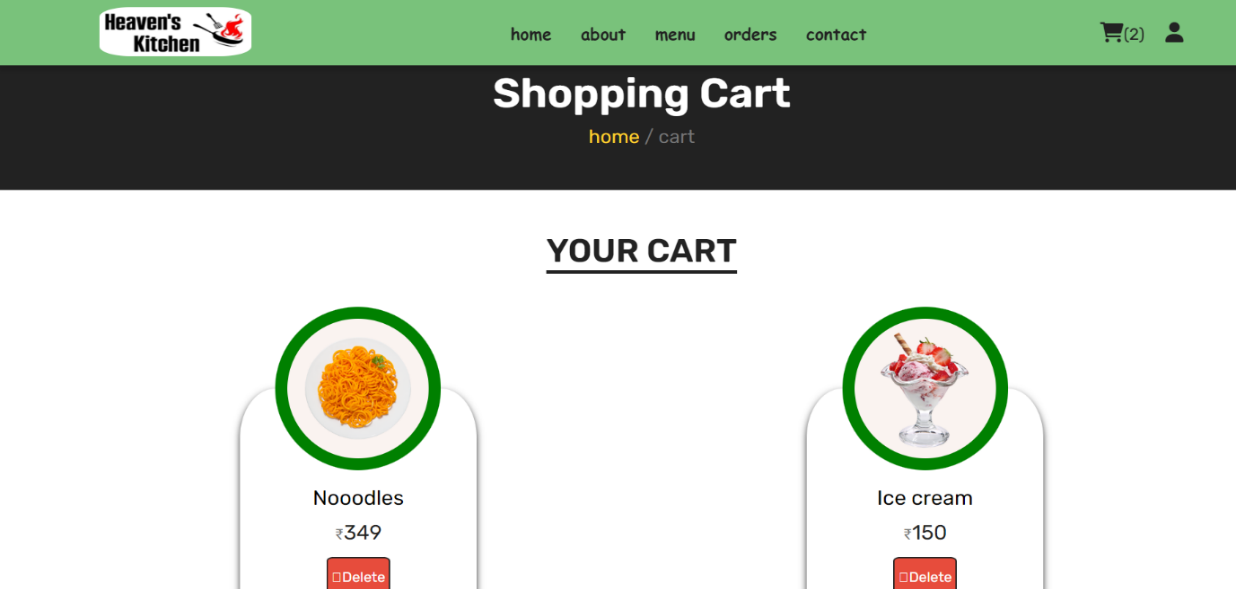


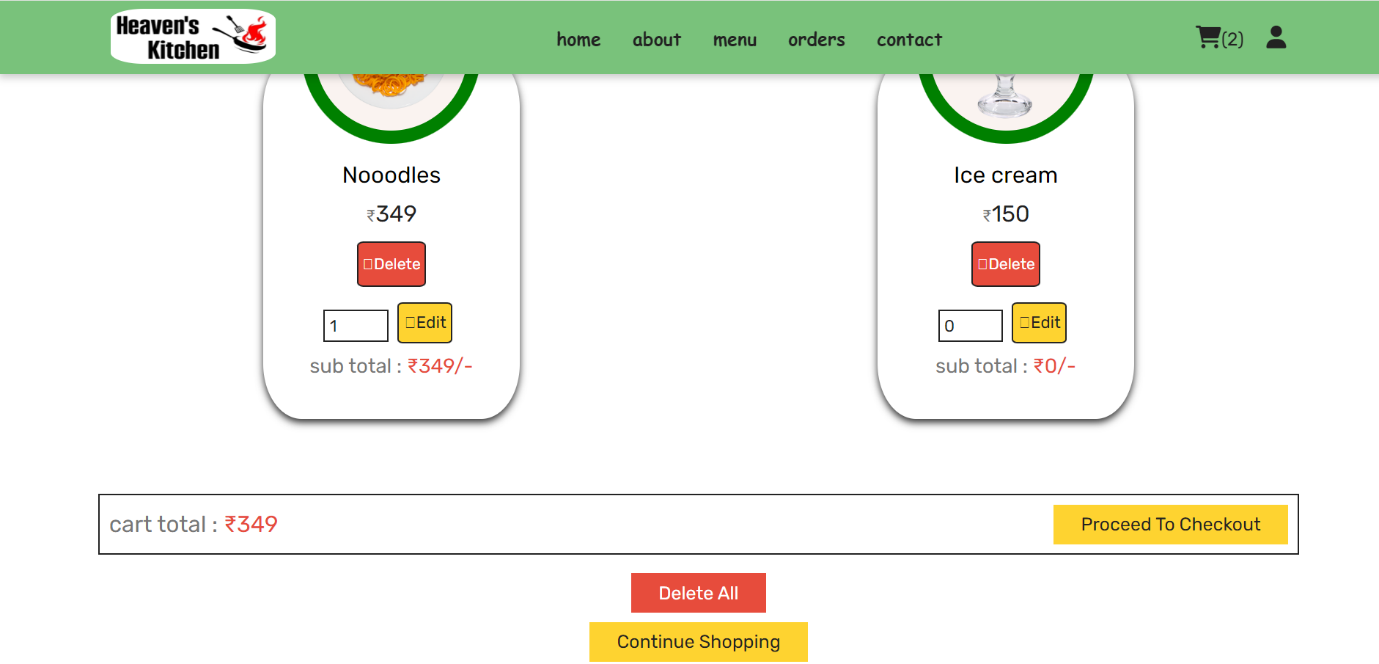


User Profile:

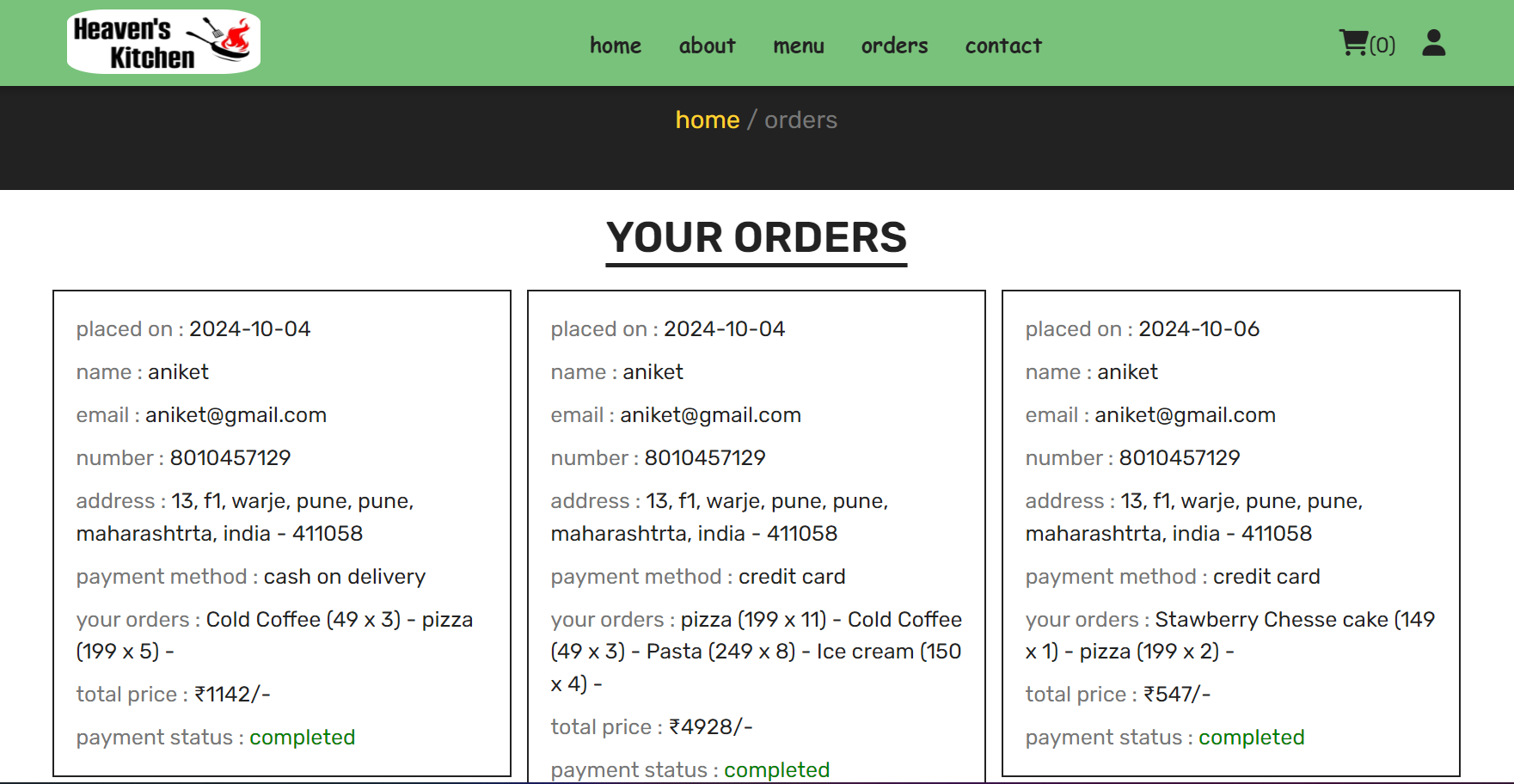


Shopping Cart:

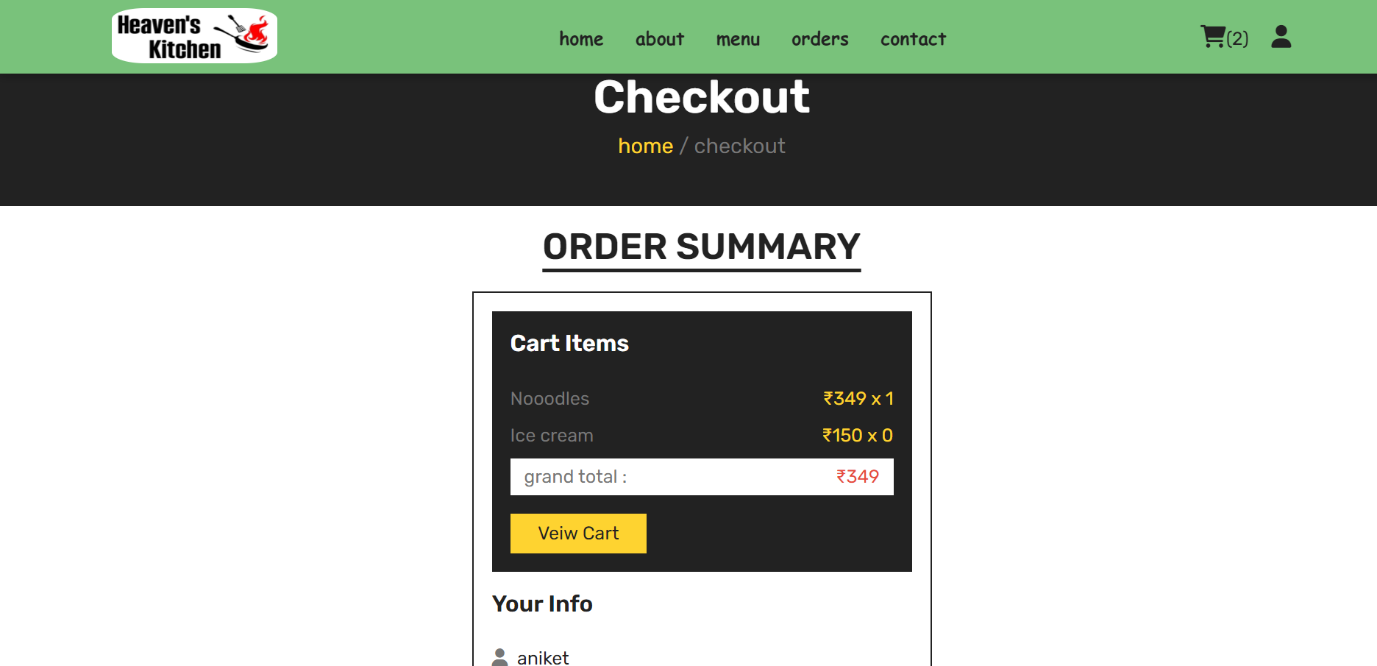


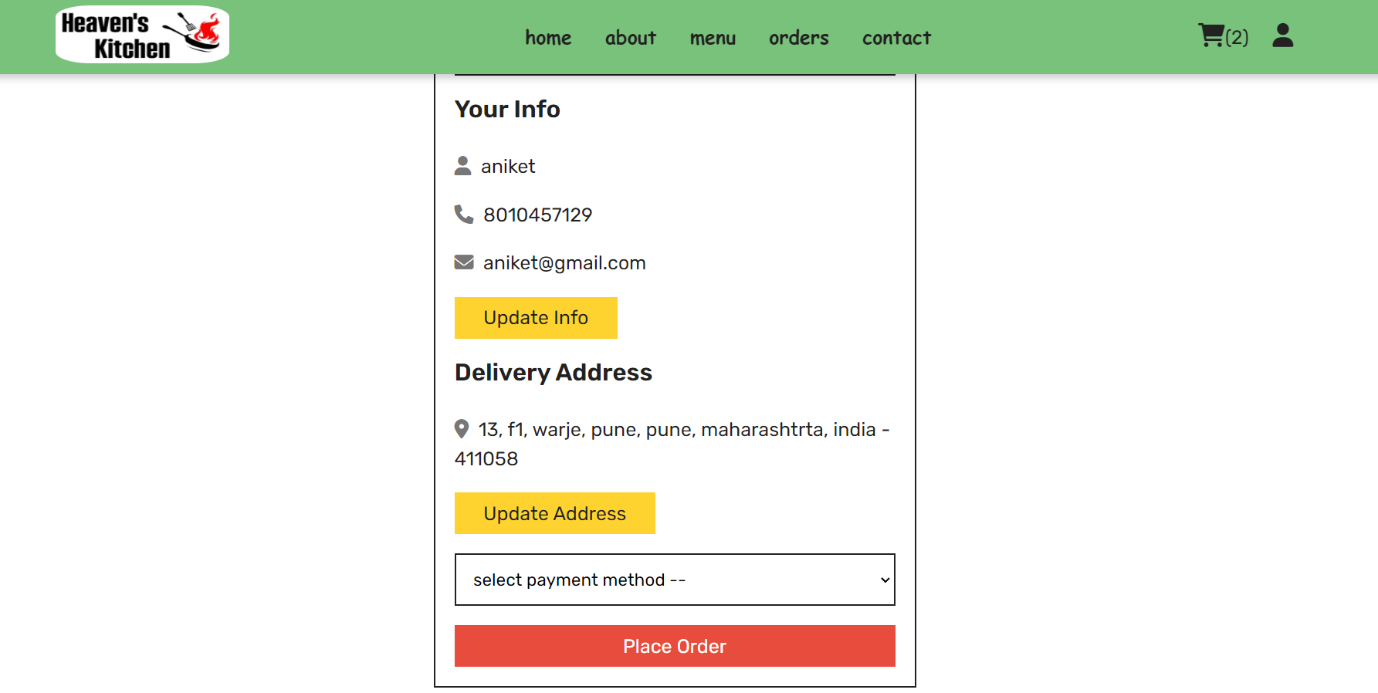


User Orders:

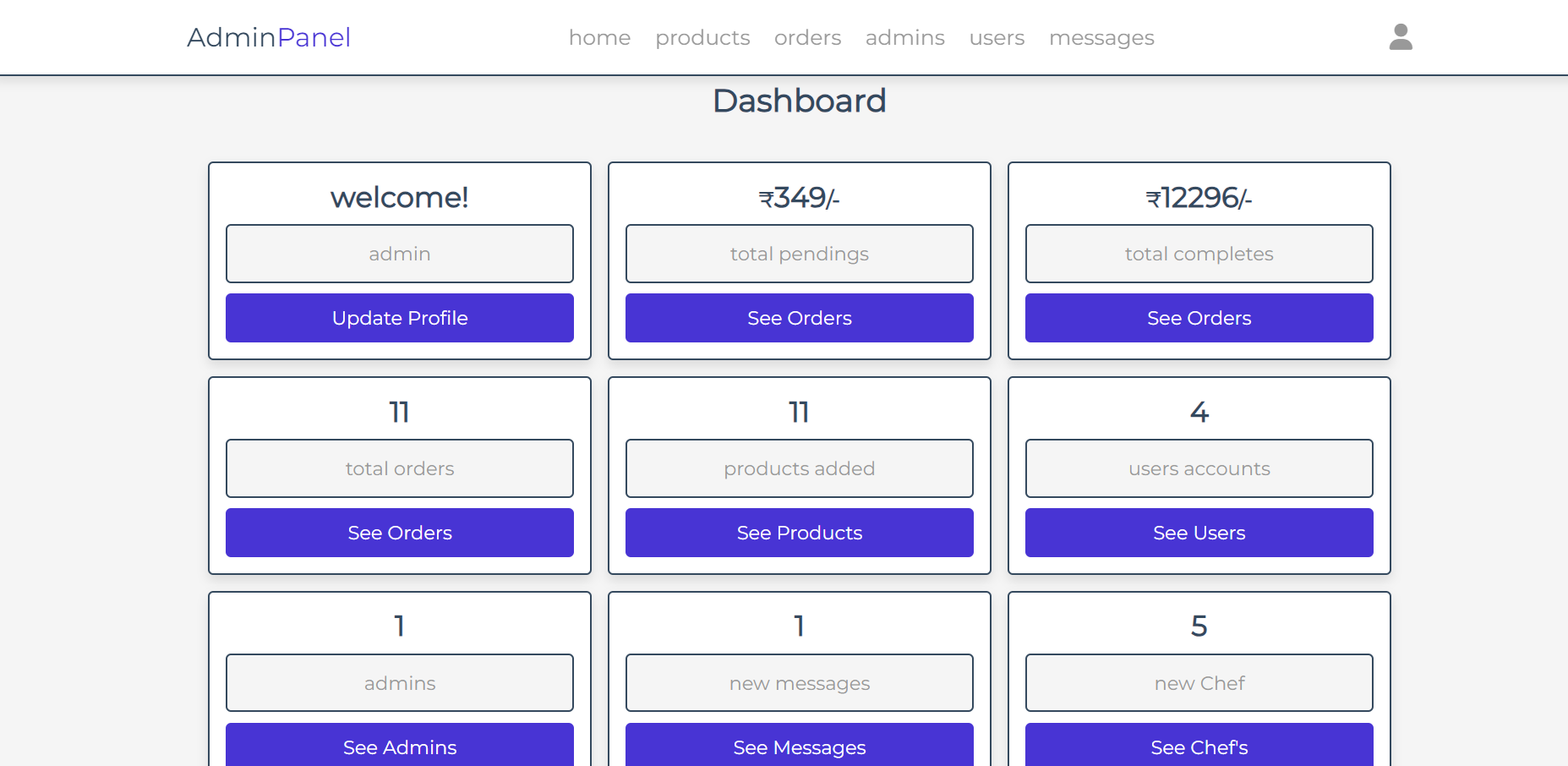


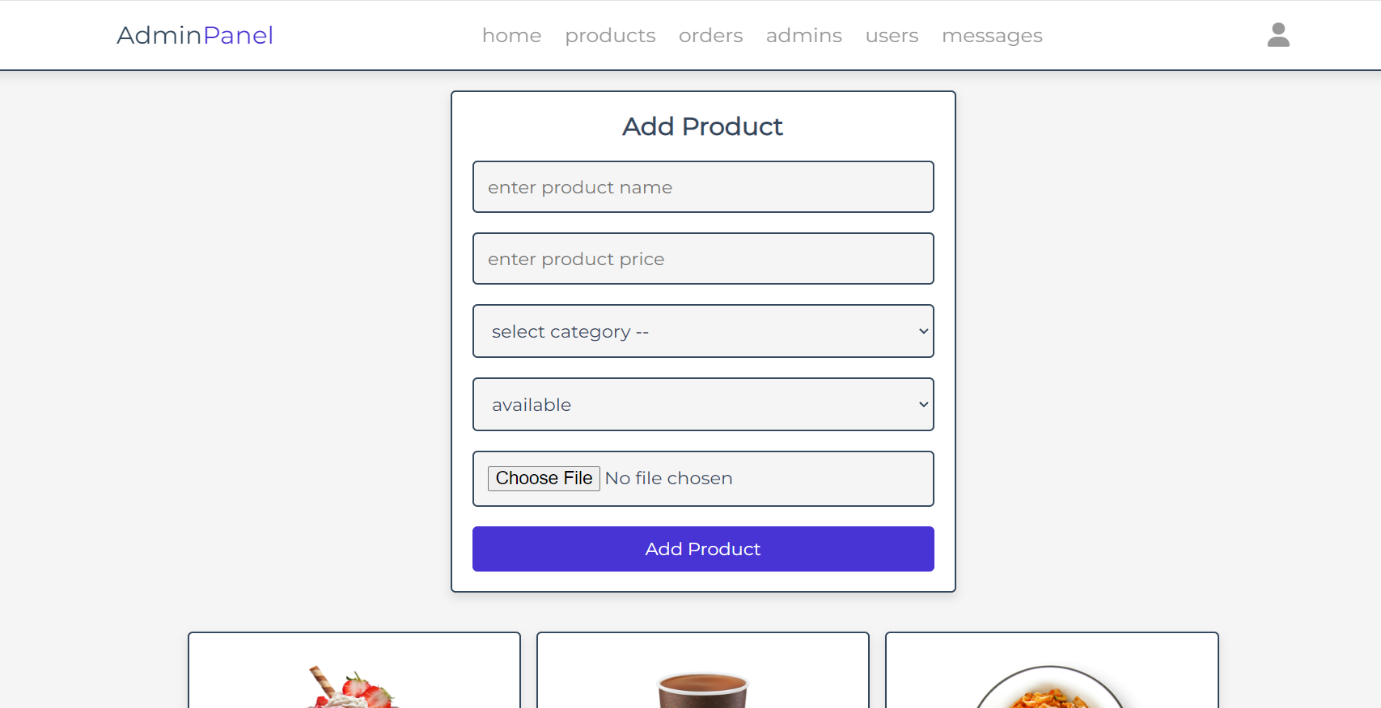
Checkout Process:

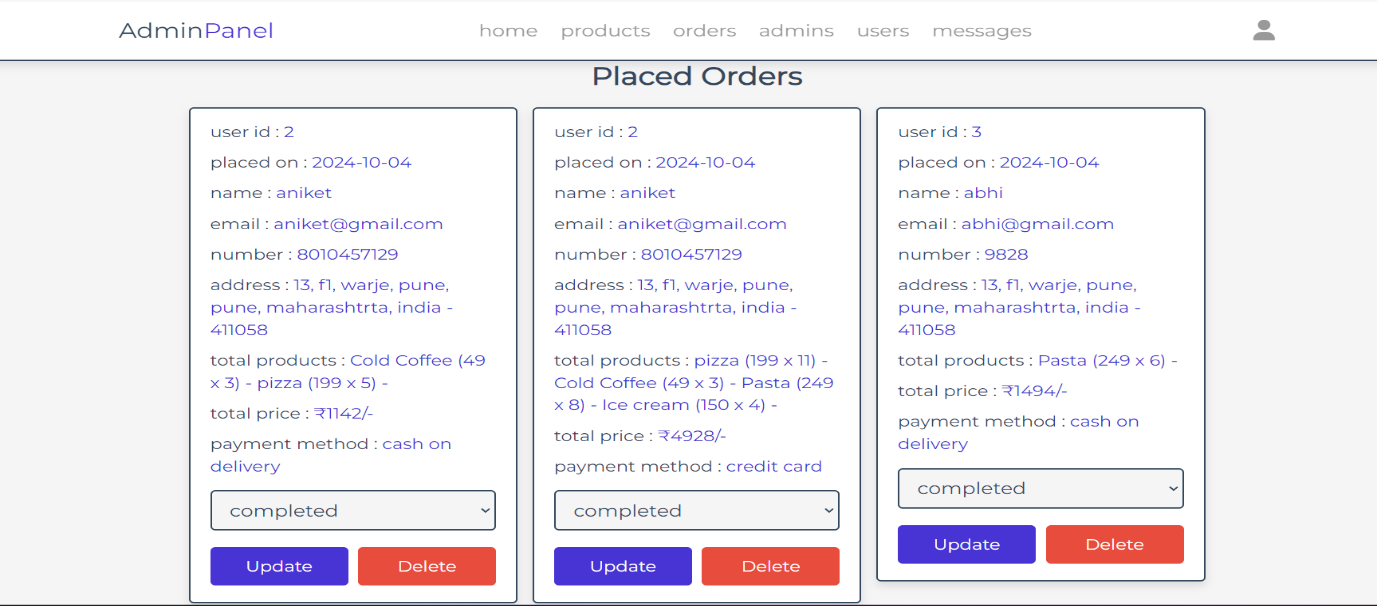


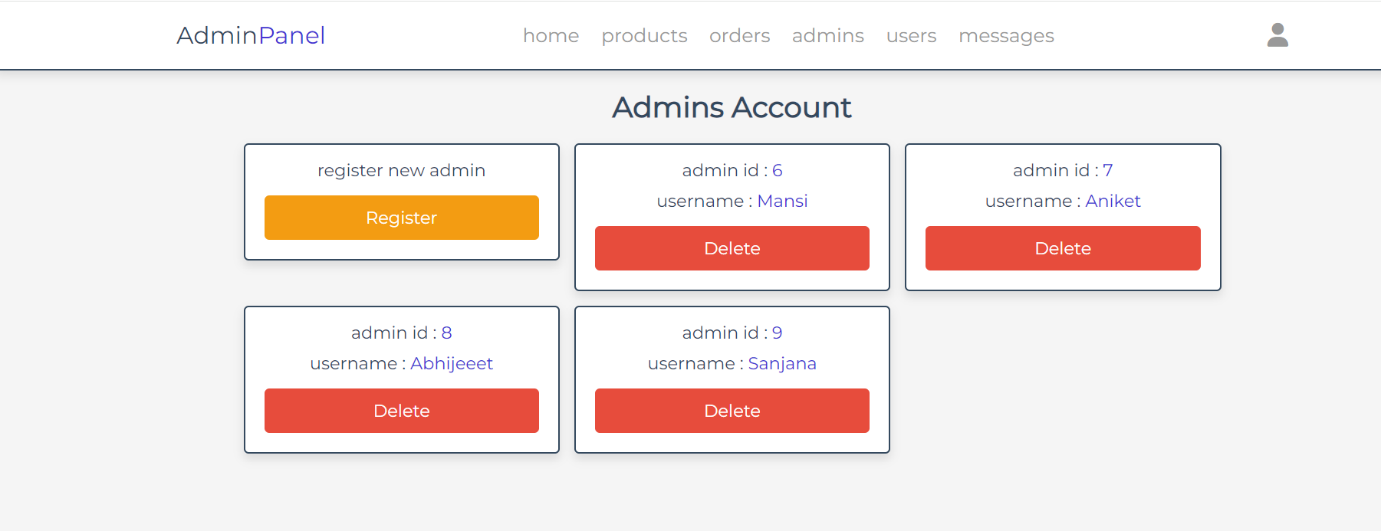


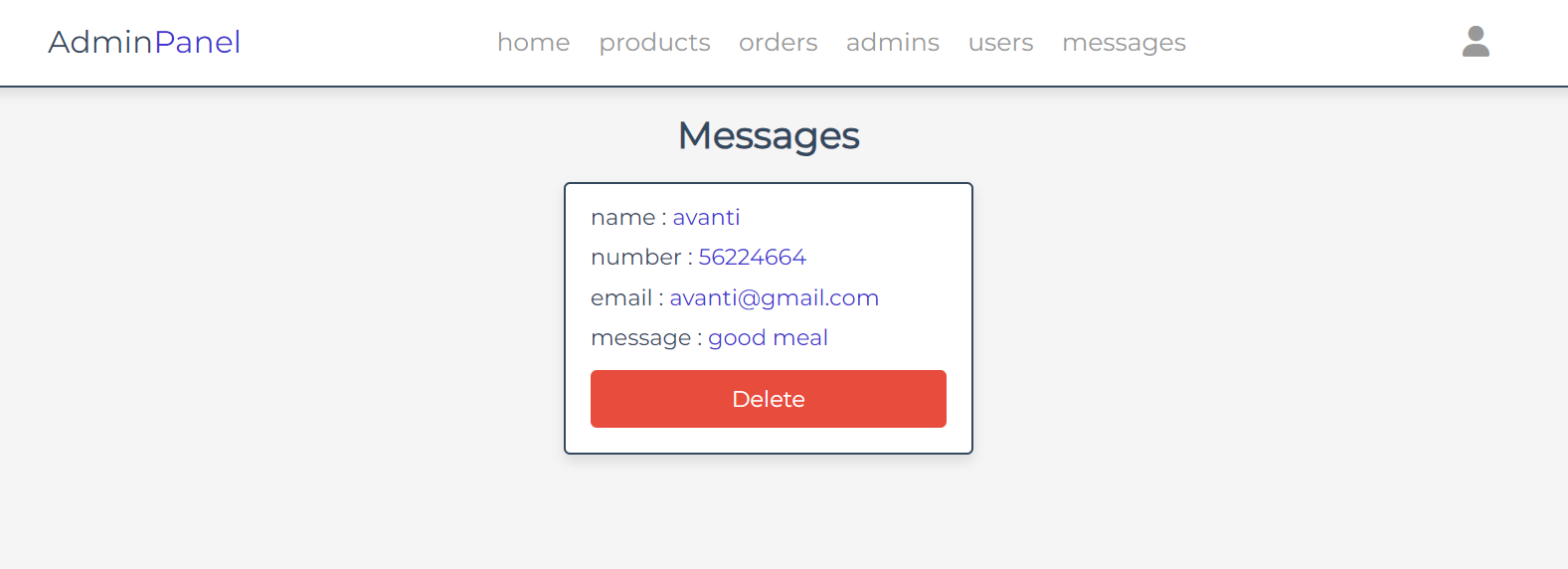
Admin Pannel:

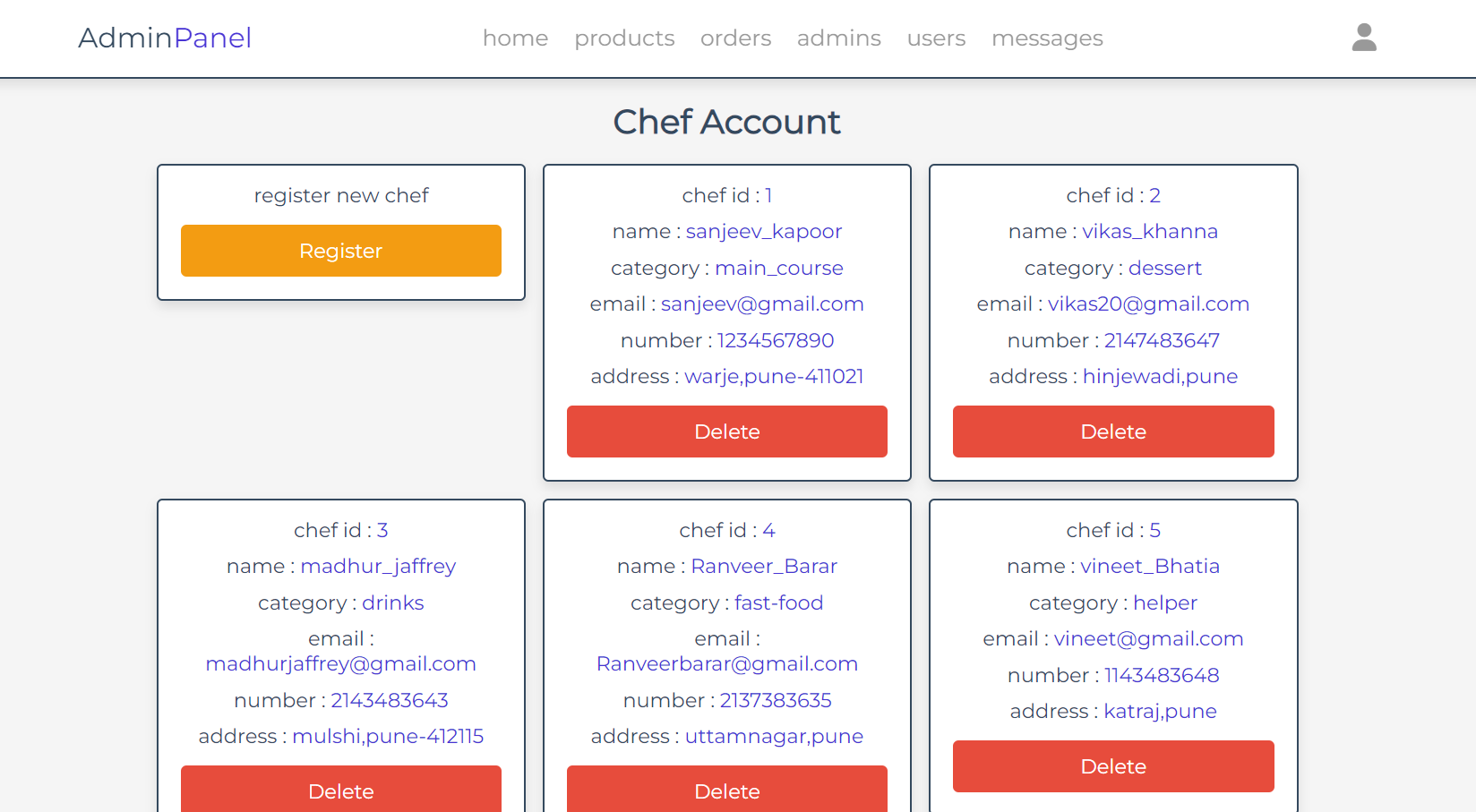














\***IMPLEMENTATION DETAILS: \***

SOFTWARE & HARDWARE SPECIFICATION

Operating System:

•Any modern operating system like Linux, Windows, or macOS can be used for development.

•Programming Languages:

•Frontend: HTML, CSS, JavaScript

•BACKEND: PHP

•RAM: MINIMUM 4GB

•Processor: At least dual-core, preferably quad-core.

•ALL TYPES OF BROWSERS

•Database Management System (DBMS): MySQL

**\*Outputs and Reports Testing\***

1. Test Plan

A test plan outlines the strategy, resources, scope, and schedule for the testing activities in the cloud kitchen website project. It defines the approach for verifying that the project meets all its requirements and functions as expected.

Test Plan Outline:

Objective:

To ensure that the cloud kitchen website is fully functional, bug-free, and meets all requirements as specified. The system should handle user interactions smoothly, provide accurate data, and process payments securely.

Scope:

This test plan covers the following areas:

1. User registration and authentication

2. Menu browsing and selection

3. Order placement and payment processing

4. Admin dashboard and management

5. Chef dashboard and menu management

6. Security testing (authentication, access control, data encryption)

Test Items:

1. User authentication and role management (customer and admin)

2. Order placement and payment gateway integration

3. Inventory management and order history

4. Frontend responsiveness and usability across devices

Test Environment:

Frontend: Chrome, Firefox, Safari (Mobile/Desktop)

Backend: PHP running on a test server

Database: MySQL test database

Payment Gateway: For now there are no payment gateway

Test Schedule:

Unit Testing: Ongoing during development

Integration Testing: After module completion

System Testing: After all modules are integrated

User Acceptance Testing (UAT): Before final deployment

Deliverables:

1. Test Cases

2. Bug Reports

3. Test Results

4. Acceptance Sign-Off

2. Black Box Testing

Black box testing focuses on testing the system from an external perspective, without looking at the internal code structure. This is useful for validating that the input-output functionality works as expected.

Test Cases for Black Box Testing:

Black Box Testing Goal:

To ensure that all user-facing functionality operates as intended, and that inputs from the user produce the expected outputs without system errors or failures.

3. White Box Testing

White box testing involves testing the internal logic, code structure, and flow of the application. This testing ensures that the code follows the correct logic paths and handles different scenarios appropriately.

Test Cases for White Box Testing:

White Box Testing Goal:

To verify that the code executes as expected and that all logical paths, conditions, loops, and error handling scenarios are covered and handled correctly.

4. Output and Test Results Summary

Output Summary:

After running the black box and white box tests, the following outcomes were observed:

User Functionality: The website functions correctly with valid input, allowing users to register, log in, browse the menu, place orders.

Admin Functionality: Admins can log in securely, manage the menu.

Security: The platform successfully blocks unauthorized access, securely handles user passwords, and prevents malicious input through input validation.

Payment Processing: Payments are processed correctly, with successful order placement and order status updates after payment confirmation.

Performance: The system performs well under normal traffic loads, with real-time updates working as expected.

Test Results:

Out of the test cases executed, most passed successfully, with a few edge cases requiring additional fixes. These edge cases mainly involved scenarios where invalid data inputs were not handled gracefully, which can be improved by refining input validation logic.

With these testing strategies and results in place, your cloud kitchen website should be ready for deployment, after addressing any identified bugs or vulnerabilities

**\*Recommendations\***

To ensure the success and continuous improvement of the cloud kitchen website, the following recommendations should be considered:

1. Implement Mobile Applications: Since many customers prefer ordering food via mobile devices, it’s recommended to develop native mobile applications for both Android and iOS. This will improve accessibility and customer engagement.

2. Advanced Analytics and Reporting: Integrating more advanced data analytics tools will help the cloud kitchen operators track customer preferences, peak order times, and inventory trends. This can lead to better decision-making, menu optimization, and demand forecasting.

3. AI-Powered Recommendations: Implementing AI-driven recommendations based on customer behaviour and previous orders can enhance the user experience by suggesting items customers are more likely to order.

4. Integration with Delivery Services: While the project focuses on internal delivery management, integrating with popular third-party delivery services can provide an additional option for kitchens without their own delivery fleet. This flexibility can attract more customers.

5. Customer Loyalty Program: To foster customer retention, consider adding a loyalty program where customers earn points or rewards for frequent orders. This can help build long-term relationships and encourage repeat business.

6. Real-Time Customer Support: Adding a live chat or chatbot feature for real-time support can help address customer queries quickly, improving overall service and reducing the chance of lost orders or customer dissatisfaction.

7. Continuous Security Audits: Since the platform deals with sensitive customer data and payments, regular security audits and vulnerability assessments should be performed to ensure the platform remains secure against evolving cyber threats.

8. Scalability for Multiple Kitchens: If the project is aimed at supporting multiple cloud kitchens under a single umbrella, it’s recommended to design the system to accommodate multiple kitchen locations, menus, and operational workflows with ease.

9. Multilingual and Multicurrency Support: To reach a broader audience, especially in diverse regions, adding support for multiple languages and currencies will enhance the platform’s usability and appeal to international customers.

10. Regular Feedback and Iteration: Establish a feedback loop with both customers and kitchen operators to regularly gather input on features, usability, and potential improvements. Continuous iteration based on real user feedback will help the platform stay competitive and user-friendly.

\*Future Scope of the Cloud Kitchen Website Project\*

The future scope of the cloud kitchen website project presents several growth opportunities and technological advancements that can enhance both functionality and scalability. These include:

1. Expansion to a Multi-Kitchen Platform:

The platform can be expanded to support multiple cloud kitchens, each with its own menu, staff, and operational workflows. This would allow for the creation of a marketplace where multiple cloud kitchens can collaborate under one system, increasing the variety for customers.

2. Integration with AI and Machine Learning:

Implementing AI and machine learning algorithms can personalize customer experiences by analyzing ordering patterns to suggest menu items, optimize kitchen workflows, and predict inventory needs. AI can also help in forecasting demand during peak hours and optimizing staffing accordingly.

3. Augmented Reality (AR) for Menu Visualization:

In the future, AR could be integrated into the platform, allowing customers to visualize how a dish looks in 3D before ordering. This could create a more immersive and engaging customer experience, helping users make informed choices.

4. Drone and Autonomous Delivery Integration:

As delivery technologies evolve, the system can integrate with drone or autonomous vehicle delivery services. This would significantly reduce delivery time, enhance customer satisfaction, and reduce reliance on traditional delivery methods.

5. Global Expansion with Multilingual and Multicurrency Support:

The platform can be scaled globally, enabling cloud kitchens from different countries to use the system. Supporting multiple languages and currencies will allow seamless operation across borders, tapping into international markets.

6. Subscription-Based Meal Plans:

The platform can introduce subscription models where customers sign up for weekly or monthly meal plans, offering regular meals from the cloud kitchen. This would create a steady revenue stream and offer customers convenience.

7. Health and Dietary Preferences:

In the future, the platform can be enhanced to provide personalized menu options based on users' dietary restrictions, health goals, or preferences (e.g., vegan, keto, gluten-free). Nutritional information and calorie tracking can also be integrated.

8. Sustainability and Eco-Friendly Options:

As sustainability becomes increasingly important, the platform could introduce features such as tracking and optimizing food waste, offering eco-friendly packaging options, and sourcing ingredients from sustainable suppliers. Customers could even filter by “green” kitchens.

9. Partnership with Grocery Suppliers for Ingredient Delivery:

The platform can extend to integrate grocery delivery for cloud kitchens. Kitchens can automatically order ingredients from suppliers based on real-time inventory data, ensuring freshness and reducing food waste.

10. Enhanced Customer Engagement through Gamification:

Gamification elements, such as rewarding users for frequent orders or completing specific challenges (e.g., trying new menu items), can be added to increase customer loyalty and interaction with the platform

**\*Bibliography and References\***

Books

* Web Development with HTML, CSS, and JavaScript – Smith John.
* Learning PHP, MYSQL & JavaScript with jQuery – Nixon Robin

Websites

* [www.techblog.com](http://www.techblog.com)
* [www.apnacollege.in](http://www.apnacollege.in)
* [www.killerphp.com](http://www.killerphp.com)
* [www.tutorialspoint.com](http://www.tutorialspoint.com)

AI Tools

* ChatGpt.AI
* Blackbox.AI
* Myninja.AI