A PROJECT ON
Online Pizza Delivery System PizzaHaven

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY
Hinjawadi

SUBMITTED BY:

Roshan Kumar Pilore,

Yash Awatade,

Swami Mahale,

Aniket Mahajan

UNDER THE GUIDENCE OF:

Mr. Snehal Jadhav

Faculty Member

Sunbeam Institute of Information Technology, Pune

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT ,Pune) and Mr. Snehal Jadhav (Project mentor).

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Roshan Kumar Pilore,

Swami Mahale,

Yash Awatade,

Aniket Mahajan

0324 PG-DAC

SIIT Pune

A PROJECT ON

"Online Pizza Delivery System PizzaHaven"

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT FOR
THE COURSE OF
DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY
Hinjewadi

SUBMITTED BY:

Roshan Kumar Pilore,
Swami Mahale,
Yash Awatade,
Aniket Mahajan

UNDER THE GUIDENCE OF:

Mr. Snehal Jadhav
Faculty Member
Sunbeam Institute of Information Technology, PUNE.

CERTIFICATE

This is to certify that the project work under the title 'Online Pizza Delivery System PizzaHaven' is done by Roshan Kumar Pilore, Swami Mahale, Yash Awatade, Aniket Mahajan in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

Mr. Yogesh Kolhe

Project Guide

Course Co-Coordinator

Date: 11-02-2025

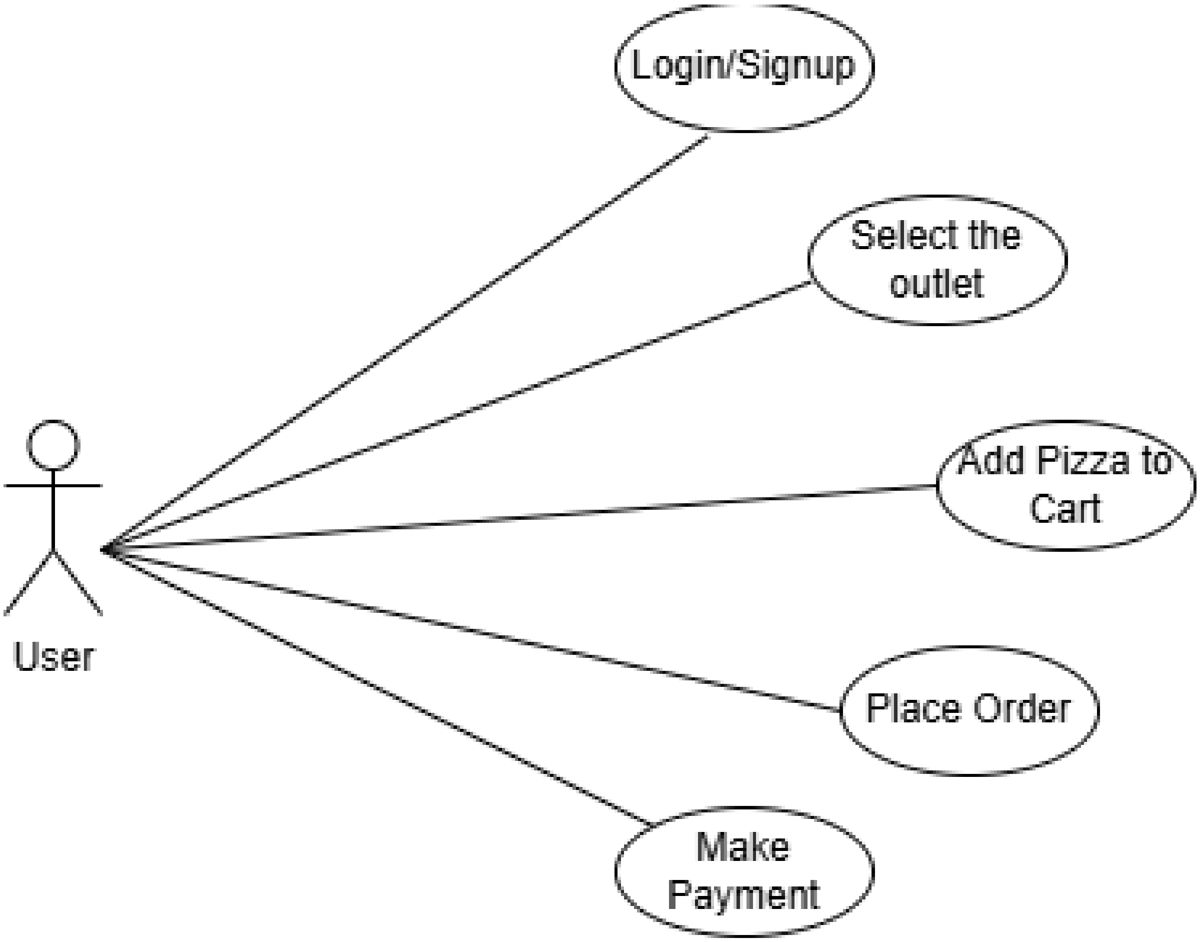
Table of Contents:

1.INTRODUCTION TO PROJECT

The **Online Pizza Delivery System** is a web-based application that enables customers to order pizzas from the nearest available outlet efficiently. It integrates **location-based outlet selection, secure payment processing**, and **role-based authentication** to enhance the user experience. Developed using **React.js for the frontend** and **Spring Boot for the backend**, the system ensures seamless ordering, secure transactions, and optimized delivery.

2.REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS



2.1 Customer Flow

2.1.1 Home Page

- **Objective:** Display a list of vendors.
- **Features:**
- View a list of available vendors with search and filter options.

2.1.2 Outlet Selection

- **Objective:** Select a Outlet to view their menu.
- **Features:**
- Click on a Outlet to see their menu items.

2.1.3 Menu Items

- **Objective:** Allow customers to view and manage menu items.
- **Features:**
- View a list of menu items for the selected vendor.
- Add items to the cart.
- Update quantities and remove items from the cart.

2.1.4 Cart Interaction

- **Objective:** Manage cart and checkout process.
- **Features:**
- **If Logged In:**
- Navigate to the cart page.
- **If Not Logged In:**
- Display a login prompt.
- If not registered, prompt registration before login.

2.1.5 Cart Page

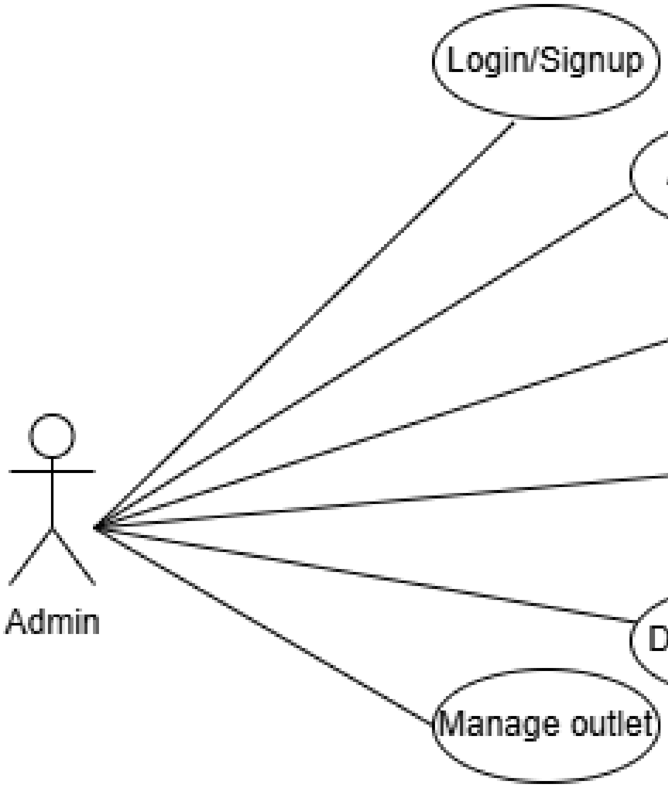
- **Objective:** Finalize the purchase process.
- **Features:**
- View cart items.
- Choose a delivery address.
- Place the order.
- If no delivery address is selected, prompt the user to choose one.
- Redirect to the home page after placing the order

2.1.6 Profile Page

Objective: Manage user profile and view order history.

Features:

- **If Not Logged In:**
- Display a sign-in prompt.
- **If Logged In:**
- View and update profile details.
- View order history and addresses.
- Submit reviews for orders
- Logout.


Login/Signup
Add Pizza
See order data
Edit Pizza
Delete Pizza
Manage outlet
Admin

2.2 Admin Flow

2.2.1 Home Page

- **Objective:** Admin access to system functionalities.
- **Features:**
- Sign in to access admin functionalities.

2.2.2 Admin Home Page

- **Objective:** Oversee and manage system operations.
- **Features:**
- View counts of customers, Outlets, and Handle the menu.
- View Lists of Customers,Outlets and menu

2.2.3 Admin Order Tracking

- **Objectives:** Oversees and manages all the order
- **Features:**
- View the username of the ordered person and amount can also change the status of the order

2.2.4 Admin Menu Page

- **Objectives:** Can oversees all the menu options available
- **Features:**
- Can change the menu
- Can change the status of the item out of stock/in stock
- Can Add or delete new items to the menu.

3. Non-Functional Requirements

3.1 Interface

- User interfaces must be intuitive and user-friendly. Detailed designs are provided in Appendix B.

3.2 Performance

- **Number of Concurrent Users:** The system should handle at least 1000 transactions/inquiries per second.
- **System Resilience:** The application should be resilient to temporary server failures.

3.3 Constraints

- The system should maintain performance standards of handling 1000 transactions/inquiries per second.

3.4 Other Requirements

3.4.1 Hardware Interfaces

Requirements: Intel Core i5 or higher (or AMD equivalent), 8 GB RAM, 512 GB SSD or larger.

□

3.4.2 Software Interfaces

- **Operating Systems:** MS Windows 13, Ubuntu 22.04.
- **Database:** MySQL.
- **Server:** Embedded Tomcat.
- **Browsers:** Compatible with modern web browsers.

4. System Design

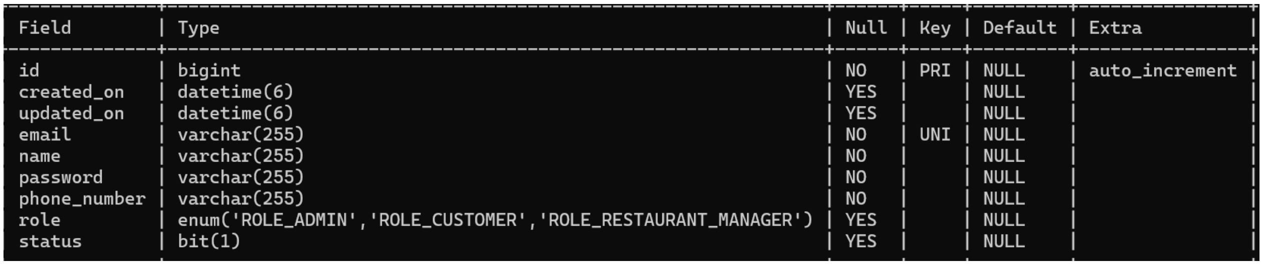
4.1 Architecture

- **Front-End:** Developed using React.js and Redux for state management.
- **Back-End:** Built with Spring Boot for server-side logic.
- **Database:** MySQL for storing user data, orders, and other system information.
- **Server:** Embedded Tomcat for hosting the application.

4. DESIGN

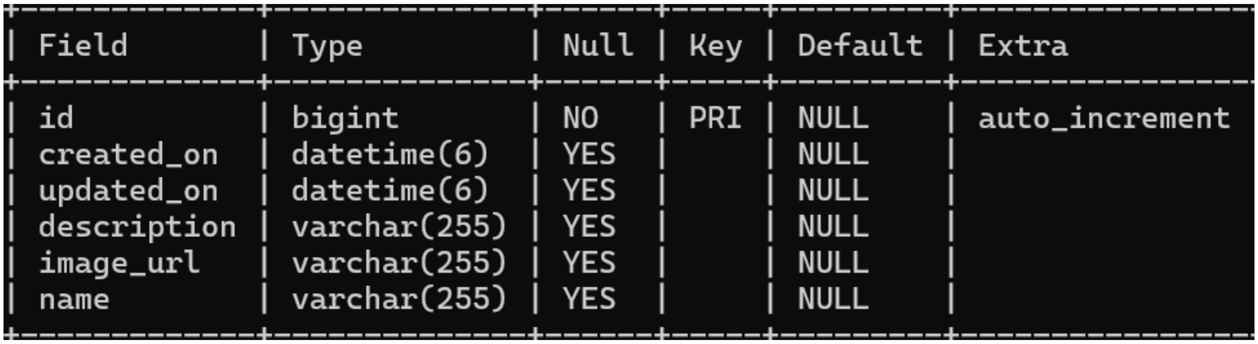
4.1 Database Design

Table 1:Users



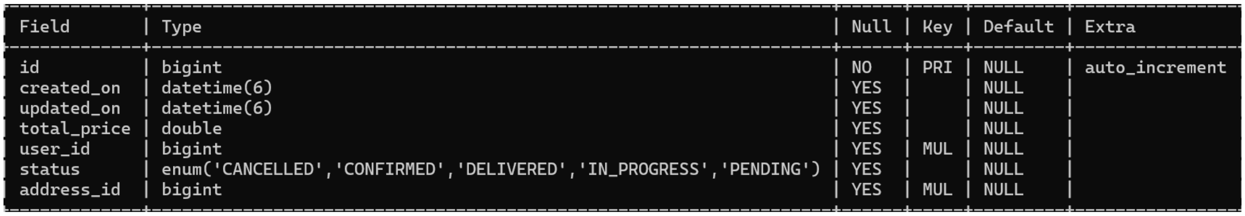
```
+--------------+---------------------------------------------------------------+------+-----+---------+----------------+
 Field        | Type                                                          | Null | Key | Default | Extra          |
+--------------+---------------------------------------------------------------+------+-----+---------+----------------+
 id           | bigint                                                        | NO   | PRI | NULL    | auto_increment |
 created_on   | datetime(6)                                                   | YES  |     | NULL    |                |
 updated_on   | datetime(6)                                                   | YES  |     | NULL    |                |
 email        | varchar(255)                                                  | NO   | UNI | NULL    |                |
 name         | varchar(255)                                                  | NO   |     | NULL    |                |
 password     | varchar(255)                                                  | NO   |     | NULL    |                |
 phone_number | varchar(255)                                                  | NO   |     | NULL    |                |
 role         | enum('ROLE_ADMIN','ROLE_CUSTOMER','ROLE_RESTAURANT_MANAGER') | YES  |     | NULL    |                |
 status       | bit(1)                                                        | YES  |     | NULL    |                |
```

Table 2: Items



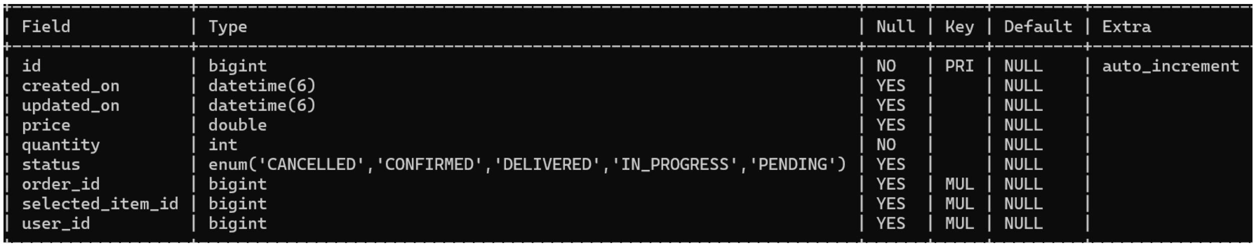
```
+-------------+--------------+------+-----+---------+----------------
| Field       | Type         | Null | Key | Default | Extra
+-------------+--------------+------+-----+---------+----------------
| id          | bigint       | NO   | PRI | NULL    | auto_increment
| created_on  | datetime(6)  | YES  |     | NULL    |
| updated_on  | datetime(6)  | YES  |     | NULL    |
| description | varchar(255) | YES  |     | NULL    |
| image_url   | varchar(255) | YES  |     | NULL    |
| name        | varchar(255) | YES  |     | NULL    |
+-------------+--------------+------+-----+---------+----------------
```

Table 3: Orders



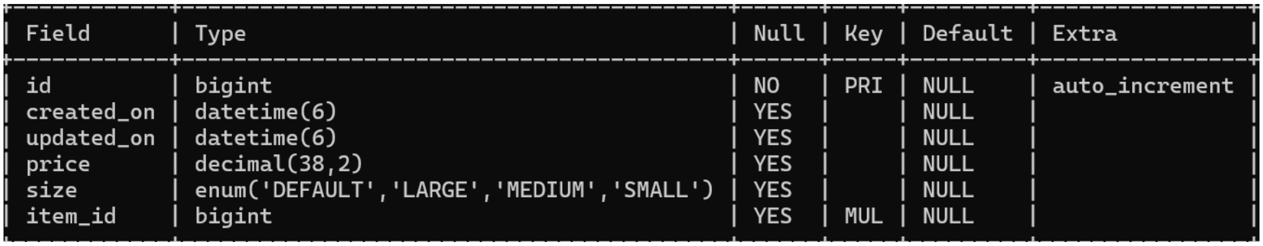
```
+-------------+-------------------------------------------------------------------+------+-----+---------+----------------
| Field       | Type                                                              | Null | Key | Default | Extra
+-------------+-------------------------------------------------------------------+------+-----+---------+----------------
| id          | bigint                                                            | NO   | PRI | NULL    | auto_increment
| created_on  | datetime(6)                                                       | YES  |     | NULL    |
| updated_on  | datetime(6)                                                       | YES  |     | NULL    |
| total_price | double                                                            | YES  |     | NULL    |
| user_id     | bigint                                                            | YES  | MUL | NULL    |
| status      | enum('CANCELLED','CONFIRMED','DELIVERED','IN_PROGRESS','PENDING') | YES  |     | NULL    |
| address_id  | bigint                                                            | YES  | MUL | NULL    |
+-------------+-------------------------------------------------------------------+------+-----+---------+----------------
```

Table 4: Order Items



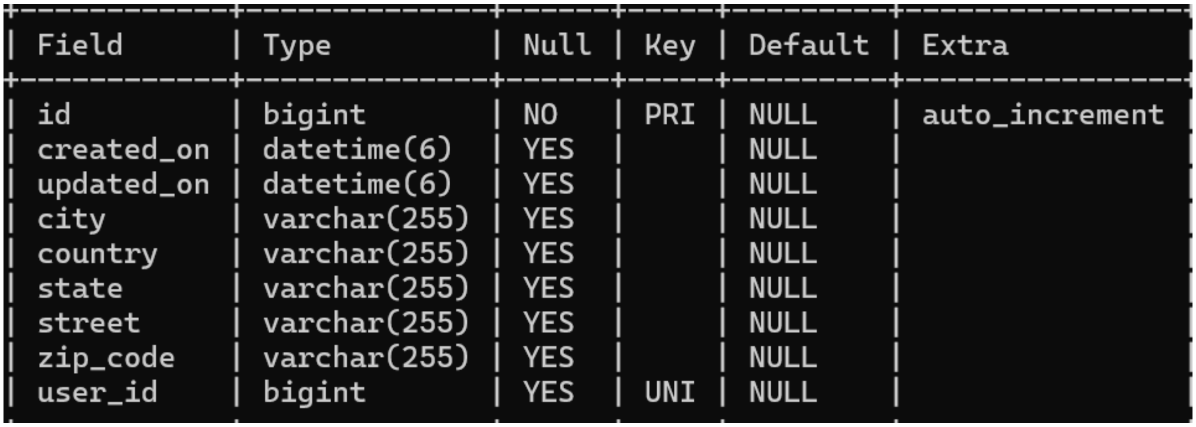
```
+------------------+-------------------------------------------------------------------+------+-----+---------+----------------
| Field            | Type                                                              | Null | Key | Default | Extra
+------------------+-------------------------------------------------------------------+------+-----+---------+----------------
| id               | bigint                                                            | NO   | PRI | NULL    | auto_increment
| created_on       | datetime(6)                                                       | YES  |     | NULL    |
| updated_on       | datetime(6)                                                       | YES  |     | NULL    |
| price            | double                                                            | YES  |     | NULL    |
| quantity         | int                                                               | NO   |     | NULL    |
| status           | enum('CANCELLED','CONFIRMED','DELIVERED','IN_PROGRESS','PENDING') | YES  |     | NULL    |
| order_id         | bigint                                                            | YES  | MUL | NULL    |
| selected_item_id | bigint                                                            | YES  | MUL | NULL    |
| user_id          | bigint                                                            | YES  | MUL | NULL    |
```

Table 5: Item Size



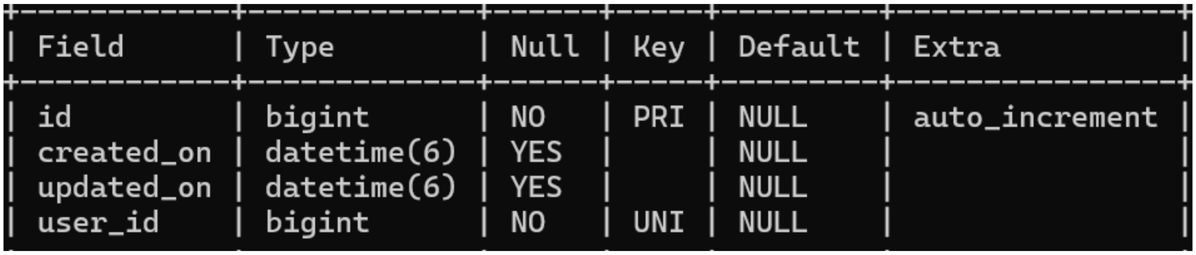
```
+------------+---------------------------------------------+------+-----+---------+----------------+
| Field      | Type                                        | Null | Key | Default | Extra          |
+------------+---------------------------------------------+------+-----+---------+----------------+
| id         | bigint                                      | NO   | PRI | NULL    | auto_increment |
| created_on | datetime(6)                                 | YES  |     | NULL    |                |
| updated_on | datetime(6)                                 | YES  |     | NULL    |                |
| price      | decimal(38,2)                               | YES  |     | NULL    |                |
| size       | enum('DEFAULT','LARGE','MEDIUM','SMALL')    | YES  |     | NULL    |                |
| item_id    | bigint                                      | YES  | MUL | NULL    |                |
+------------+---------------------------------------------+------+-----+---------+----------------+
```

Table 6: Adresses



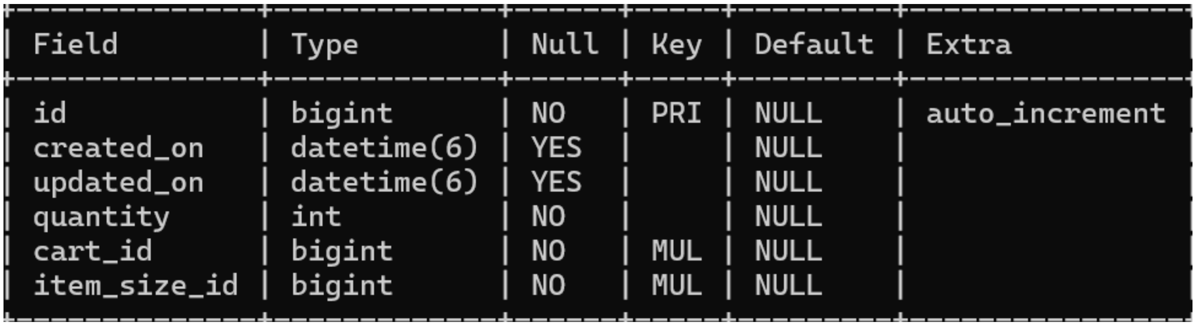
```
+------------+--------------+------+-----+---------+----------------
| Field      | Type         | Null | Key | Default | Extra
+------------+--------------+------+-----+---------+----------------
| id         | bigint       | NO   | PRI | NULL    | auto_increment
| created_on | datetime(6)  | YES  |     | NULL    |
| updated_on | datetime(6)  | YES  |     | NULL    |
| city       | varchar(255) | YES  |     | NULL    |
| country    | varchar(255) | YES  |     | NULL    |
| state      | varchar(255) | YES  |     | NULL    |
| street     | varchar(255) | YES  |     | NULL    |
| zip_code   | varchar(255) | YES  |     | NULL    |
| user_id    | bigint       | YES  | UNI | NULL    |
```

Table 7: Carts



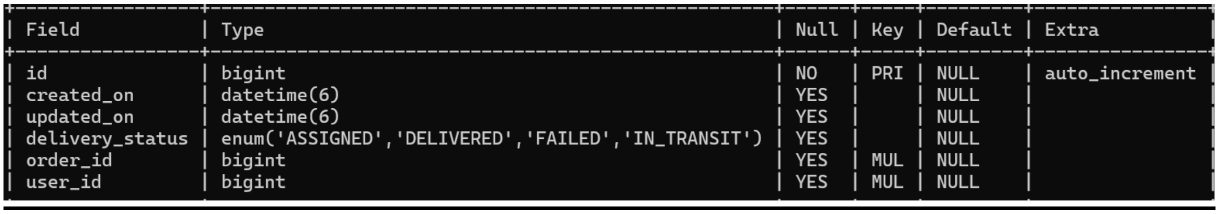
```
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| id         | bigint      | NO   | PRI | NULL    | auto_increment |
| created_on | datetime(6) | YES  |     | NULL    |                |
| updated_on | datetime(6) | YES  |     | NULL    |                |
| user_id    | bigint      | NO   | UNI | NULL    |                |
```

Table 8: Cart Items



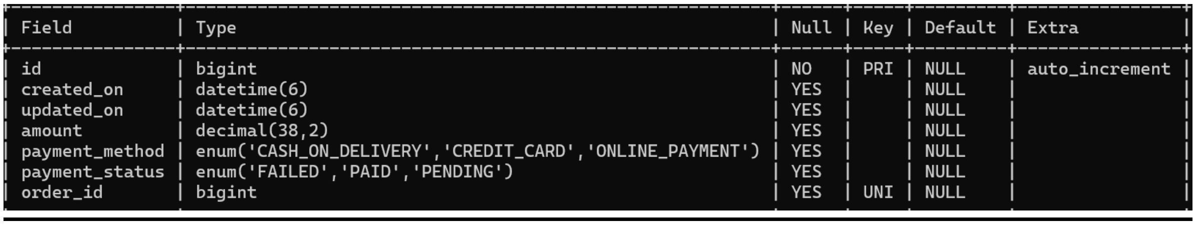
```
+--------------+-------------+------+-----+---------+----------------
| Field        | Type        | Null | Key | Default | Extra
+--------------+-------------+------+-----+---------+----------------
| id           | bigint      | NO   | PRI | NULL    | auto_increment
| created_on   | datetime(6) | YES  |     | NULL    |
| updated_on   | datetime(6) | YES  |     | NULL    |
| quantity     | int         | NO   |     | NULL    |
| cart_id      | bigint      | NO   | MUL | NULL    |
| item_size_id | bigint      | NO   | MUL | NULL    |
+--------------+-------------+------+-----+---------+----------------
```

Table 9: Delivery



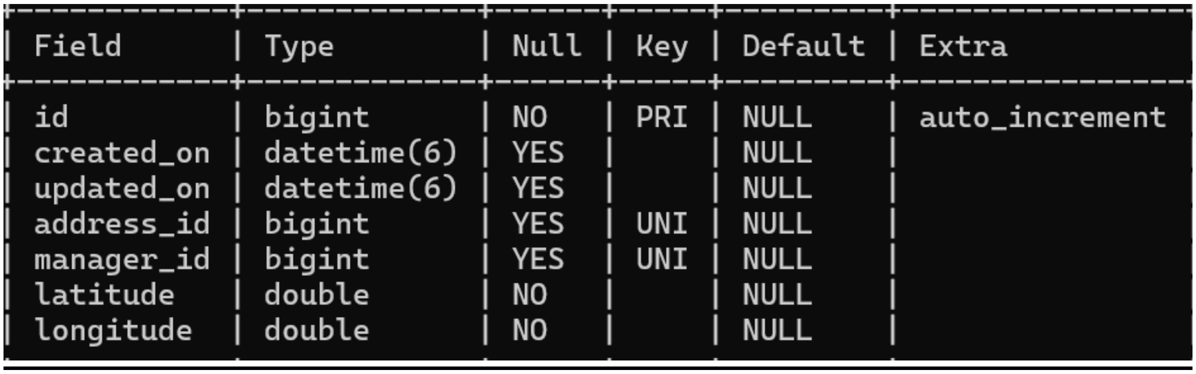
```
+-----------------+----------------------------------------------------+------+-----+---------+----------------+
| Field           | Type                                               | Null | Key | Default | Extra          |
+-----------------+----------------------------------------------------+------+-----+---------+----------------+
| id              | bigint                                             | NO   | PRI | NULL    | auto_increment |
| created_on      | datetime(6)                                        | YES  |     | NULL    |                |
| updated_on      | datetime(6)                                        | YES  |     | NULL    |                |
| delivery_status | enum('ASSIGNED','DELIVERED','FAILED','IN_TRANSIT') | YES  |     | NULL    |                |
| order_id        | bigint                                             | YES  | MUL | NULL    |                |
| user_id         | bigint                                             | YES  | MUL | NULL    |                |
```

Table 10: Payment



```
+----------------+---------------------------------------------------------+------+-----+---------+----------------+
| Field          | Type                                                    | Null | Key | Default | Extra          |
+----------------+---------------------------------------------------------+------+-----+---------+----------------+
| id             | bigint                                                  | NO   | PRI | NULL    | auto_increment |
| created_on     | datetime(6)                                             | YES  |     | NULL    |                |
| updated_on     | datetime(6)                                             | YES  |     | NULL    |                |
| amount         | decimal(38,2)                                           | YES  |     | NULL    |                |
| payment_method | enum('CASH_ON_DELIVERY','CREDIT_CARD','ONLINE_PAYMENT') | YES  |     | NULL    |                |
| payment_status | enum('FAILED','PAID','PENDING')                         | YES  |     | NULL    |                |
| order_id       | bigint                                                  | YES  | UNI | NULL    |                |
```

Table 11: Outlet



```
+------------+-------------+------+-----+---------+----------------
| Field      | Type        | Null | Key | Default | Extra
+------------+-------------+------+-----+---------+----------------
| id         | bigint      | NO   | PRI | NULL    | auto_increment
| created_on | datetime(6) | YES  |     | NULL    |
| updated_on | datetime(6) | YES  |     | NULL    |
| address_id | bigint      | YES  | UNI | NULL    |
| manager_id | bigint      | YES  | UNI | NULL    |
| latitude   | double      | NO   |     | NULL    |
| longitude  | double      | NO   |     | NULL    |
```

E-R Diagram,Dataflow diagram and Class Diagram:

Go to Appendix A

5. CODING STANDARDS IMPLEMENTED

Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

Identifier	Case	Examples	Additional Notes
Class	Pascal	User, Order, UserController	Class names should be based on "objects" or "real things" and should generally be **nouns**. No '_' signs allowed. Do not use type prefixes like 'C' for class.
Method	Camel	SignUp, SignIn, addReviews	Methods should use **verbs** or verb phrases.
Parameter	Camel	firstName, lastName, email, password	Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios.
Interface	Pascal with "I" prefix	UserRepository, OrderRepository , MenuRepository	Do not use the '_' sign
Annotation	Pascal	SpringBootApplication	Use @ at start of annotation

DTOs	Camel	ApiResponseDTO, SignUpReqDTO, OrderDetailsResDTO	Use to transfer data between the processes
Exception Class	Pascal with "Exception" suffix	ResourceNotFoundException	

Comments

- Comment each type, each non-public type member, and each region declaration.
- Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.
- Separate comments from comment delimiters (apostrophe) or // with one space.
- Begin the comment text with an uppercase letter.
- End the comment with a period.
- Explain the code; do not repeat it.

6. TEST REPORT

Another group called Linux did the testing and the report of the testing is given hereunder.

GENERAL TESTING:

SR-NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
1	SignUp Page	Signup successfully message	OK	Nothing
2	SignIn Page	Pop-up will come	Ok	Please enter username and password again .
3	HomePage	Outlet List fetched from db	Ok	Failed to fetched Outlet list
4	Menu List page	Gives all menu item for selected Outlet	Ok	Nothing
5	Cart Page	Give a list of added item in the cart	Ok	Please log in first
6	Customer Order history	Order history list render successfully	Ok	Failed to fetched orders
7	Add Menu Item	Menu added successfully	Ok	Nothing
8	See placed orders by vendor	Placed order list view	Ok	No orders is placed
9	Generate Token after first login	Token generated successfully	Ok	Failed to generate token
10	View all users list by admin	Seeing the list of all users	Ok	Nothing
11	Placing Order by customer	Order placed successfully and redirect to home page	Ok	Nothing

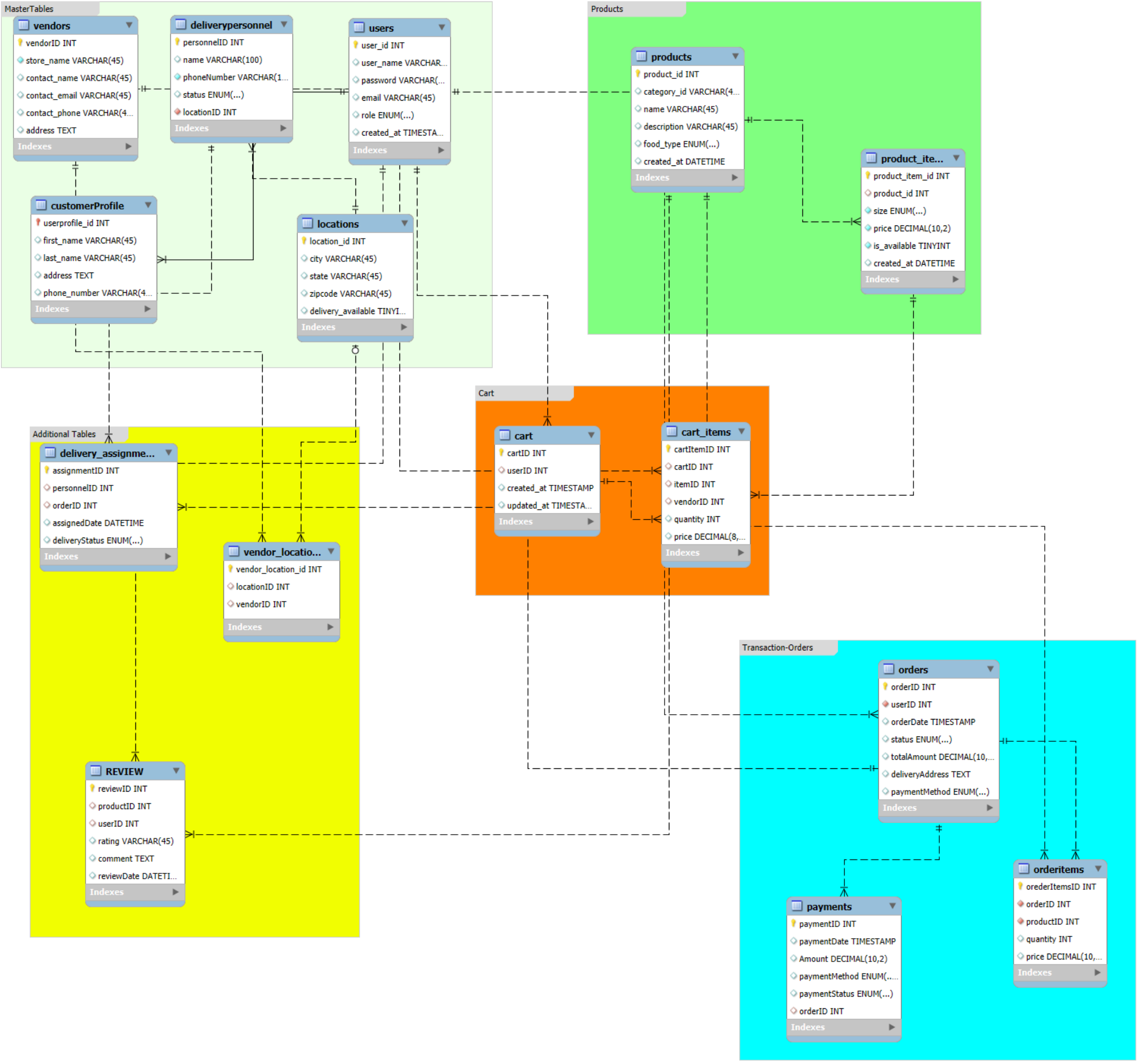
12	Logout	It will logout from user profile.	Ok	Nothing
	STATIC TESTING			
SR-NO	**Deviation**	**Program**		
1	Commenting not followed	All Web Application		

6. PROJECT MANAGEMENT RELATED STATISTICS

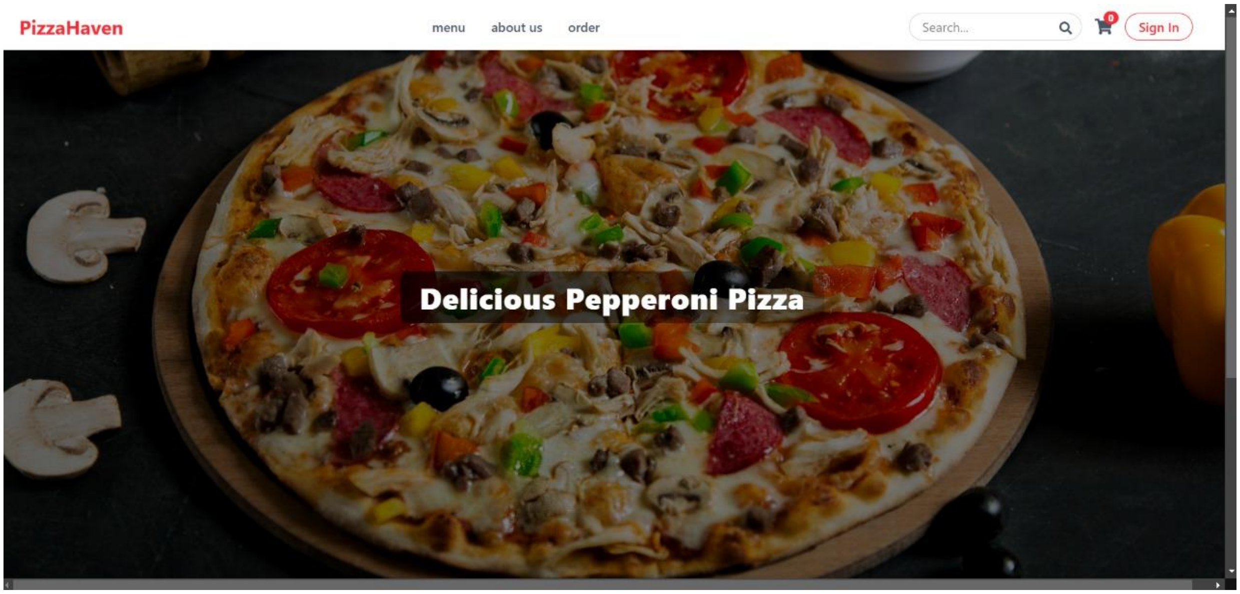
DATE	WORK PERFORMED	SLC PHASE	Additional Notes
October 2, 2024	Project Allotment and User Requirements Gathering	Feasibility Study	Our team met the client Mr. Nitin Kudale (CEO, SIIT Pune) to know his requirements.
October 12, 2024	Initial SRS Document Validation and Team Structure Decided	Requirement Analysis (Elicitation)	The initial SRS was presented to the client to understand his requirements better.
October 30, 2024 2024	Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram, and User Interfaces	Requirement Analysis & Design Phase	Database Design completed.
Nov 10, 2024	Business Logic Component Design Started	Design Phase	----------------------------
Nov 20, 2024	Coding Phase Started	Coding Phase	70% of Class Library implemented.
Nov 27, 2024	Implementation of Web Application and Window Application Started	Coding Phase	Class Library Development going on.
Dec 15, 2024	Implementation of Web Application and Window Application Continued	Coding Phase	POJOs and Class Development

Dec 25, 2024	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	Class Library Modified as per the need.
Jan 12, 2025	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	--------------------------- -
Jan 26, 2025	After Ensuring Proper Functioning the Required Validations were Implemented	Coding Phase and Unit Testing	Module Integration was done by the Project Manager
Jan 26, 2025	The Project was Tested by the respective Team Leaders and the Project Manager	Testing Phase (Module Testing)	--------------------------- -
Jan 26, 2025	The Project was Submitted to Other Project Leader of Other Project Group For Testing	Testing Phase (Acceptance Testing)	The Project of Other Team was Taken up by the Team for Testing
Feb 05, 2025	The Errors Found were Removed	Debugging	The Project was complete for submission
Feb 11, 2025	Final Submission of Project	------------- -- --	-

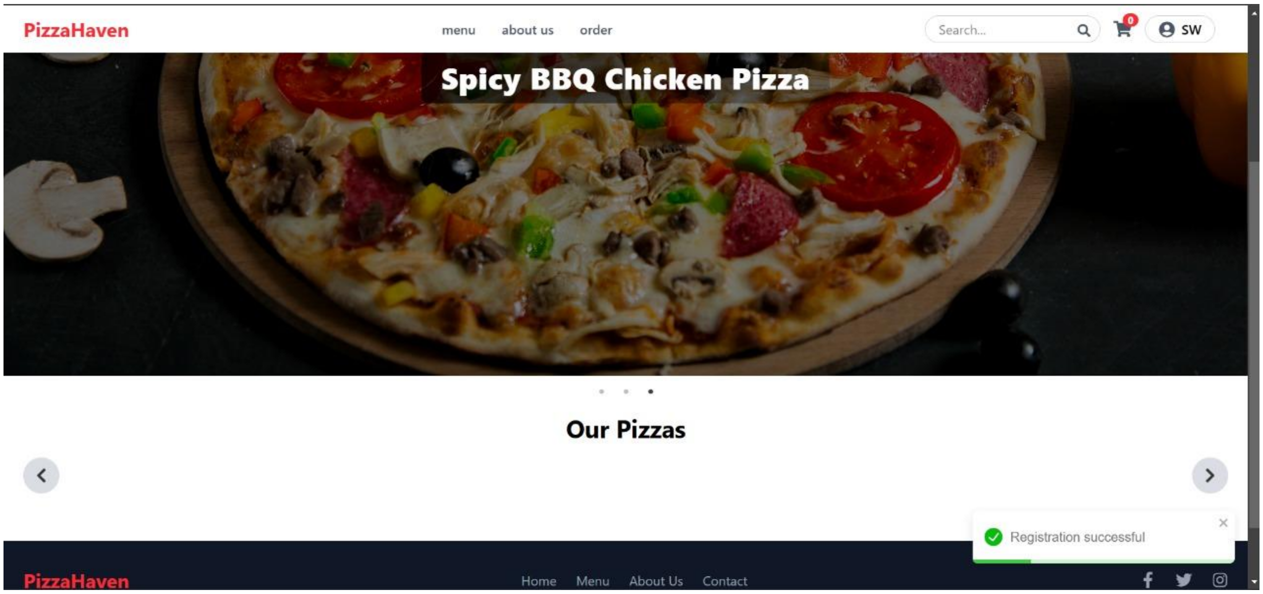
Appendix A
Entity Relationship Diagram



Landing Page (Carousel)



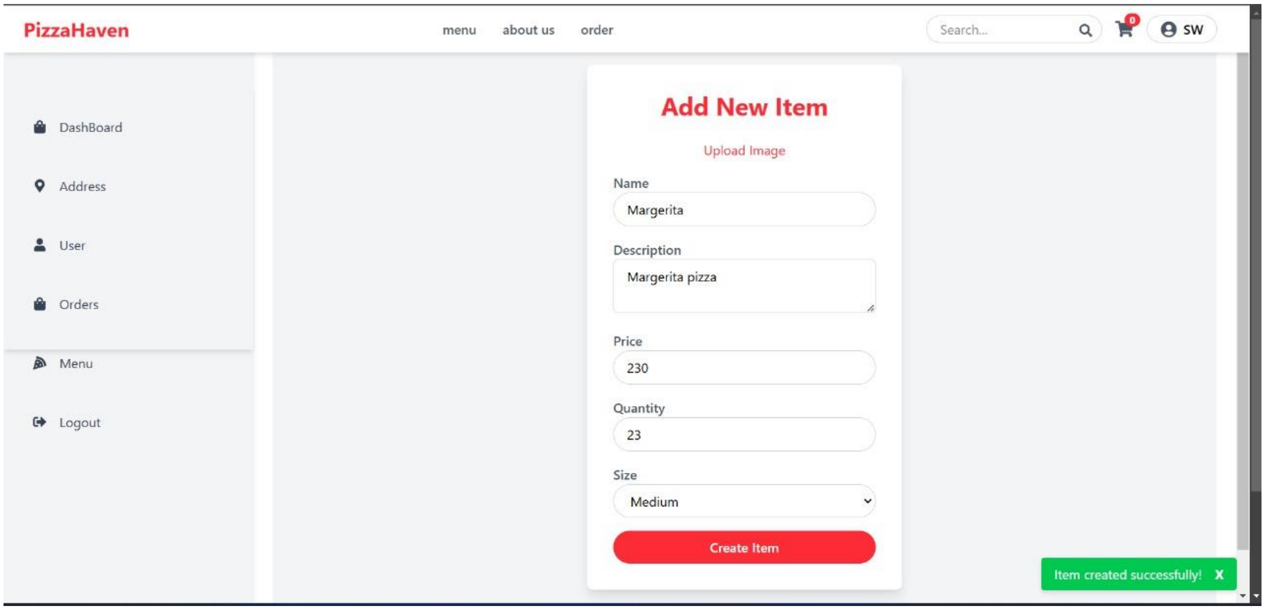
Pizza Options Displayed On the Landing page



Registration successful

Registration Successful displayed using the using toastify

Admin page to add new pizza to the menu



Add New Item

Name: Margerita

Description: Margerita pizza

Price: 230

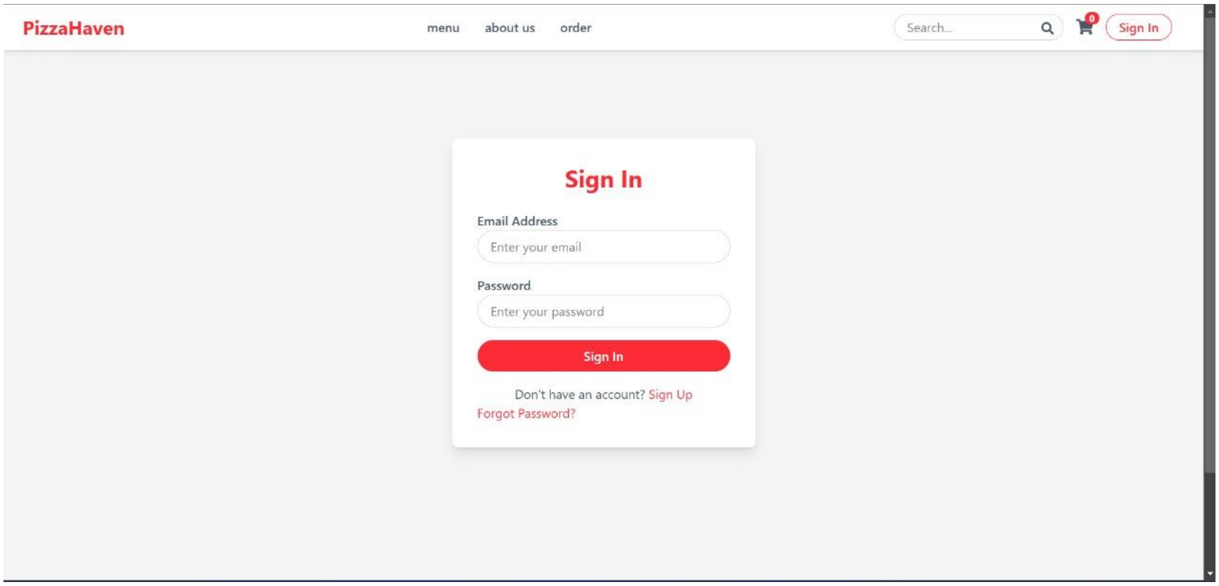
Quantity: 23

Size: Medium

Item created successfully! X

Dashboard UI also Displayed by the side for admin interface

Sign in Window

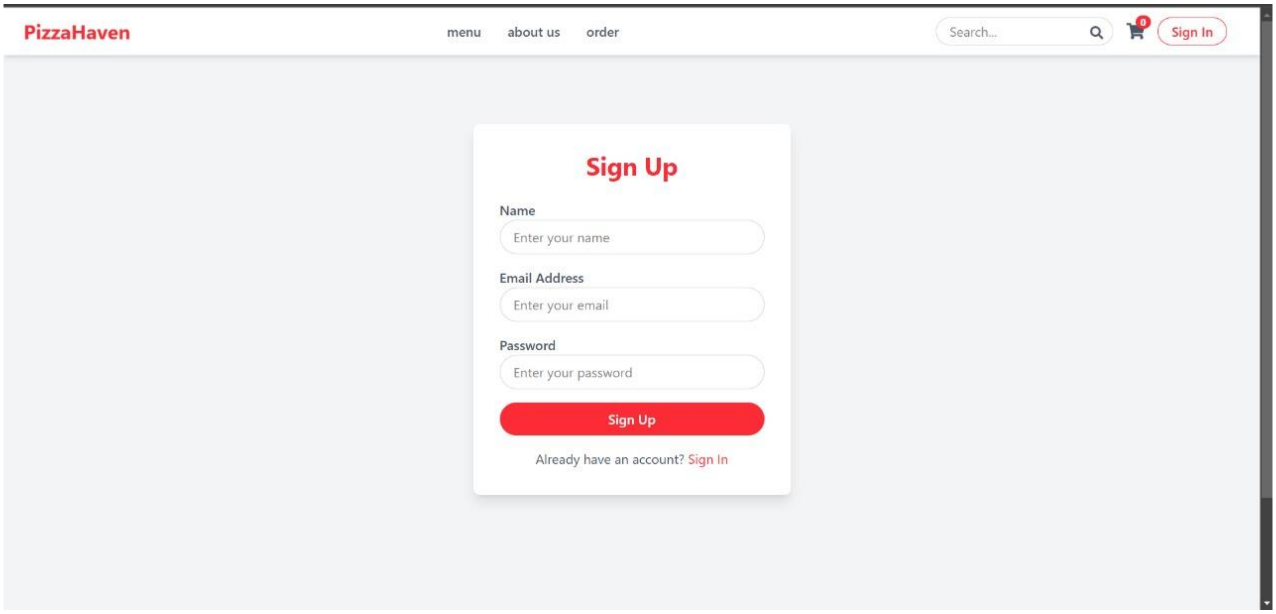


Sign In

Email Address: Enter your email

Password: Enter your password

Sign Up Window



Sign Up

Name

Enter your name

Email Address

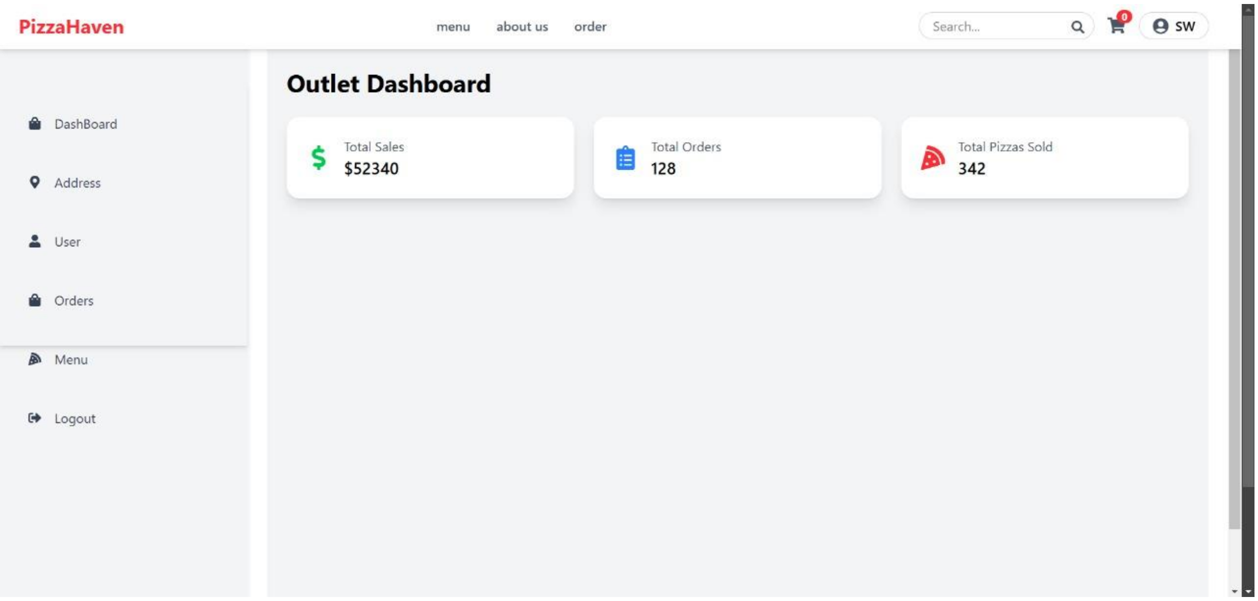
Enter your email

Password

Enter your password

Already have an account? Sign In

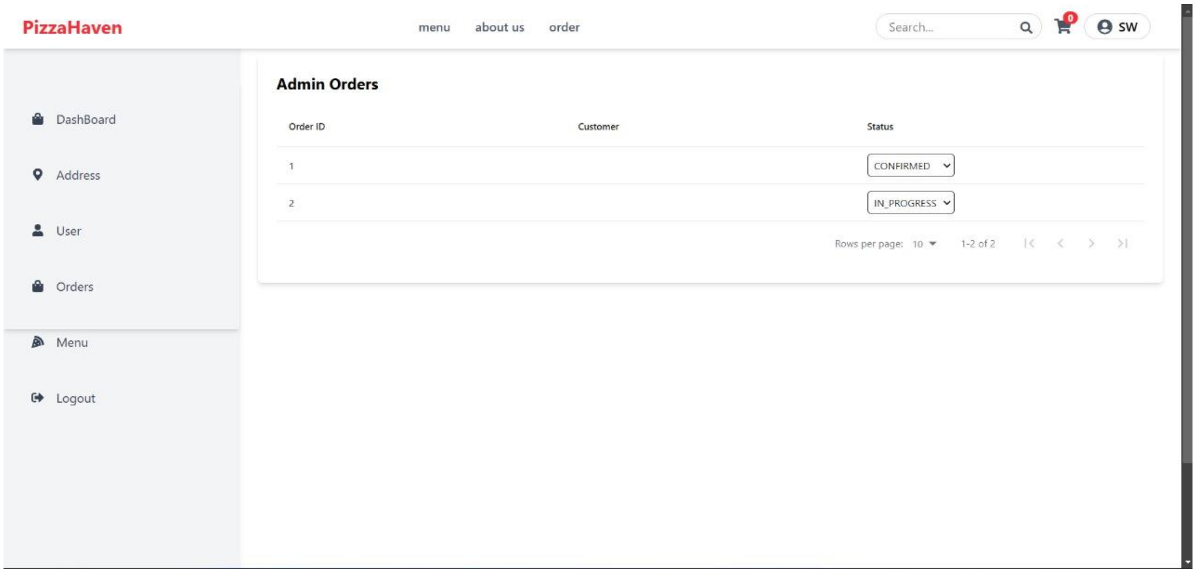
Outlet Dashboard



Outlet Dashboard

Total Sales	Total Orders	Total Pizzas Sold
$52340	128	342

Admin Dashboard



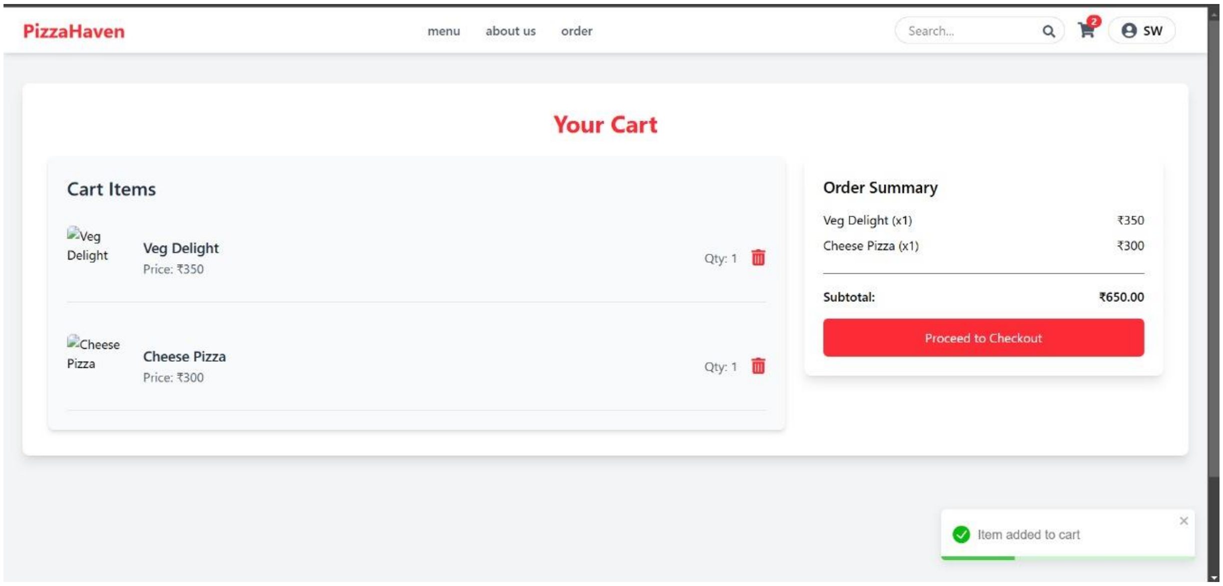
Admin Orders

Order ID	Customer	Status
1		CONFIRMED
2		IN_PROGRESS

Rows per page: 10 1-2 of 2

Admin can track orders and change statuses

Cart window



Your Cart

Cart Items

Veg Delight
Veg Delight
Price: ₹350
Qty: 1

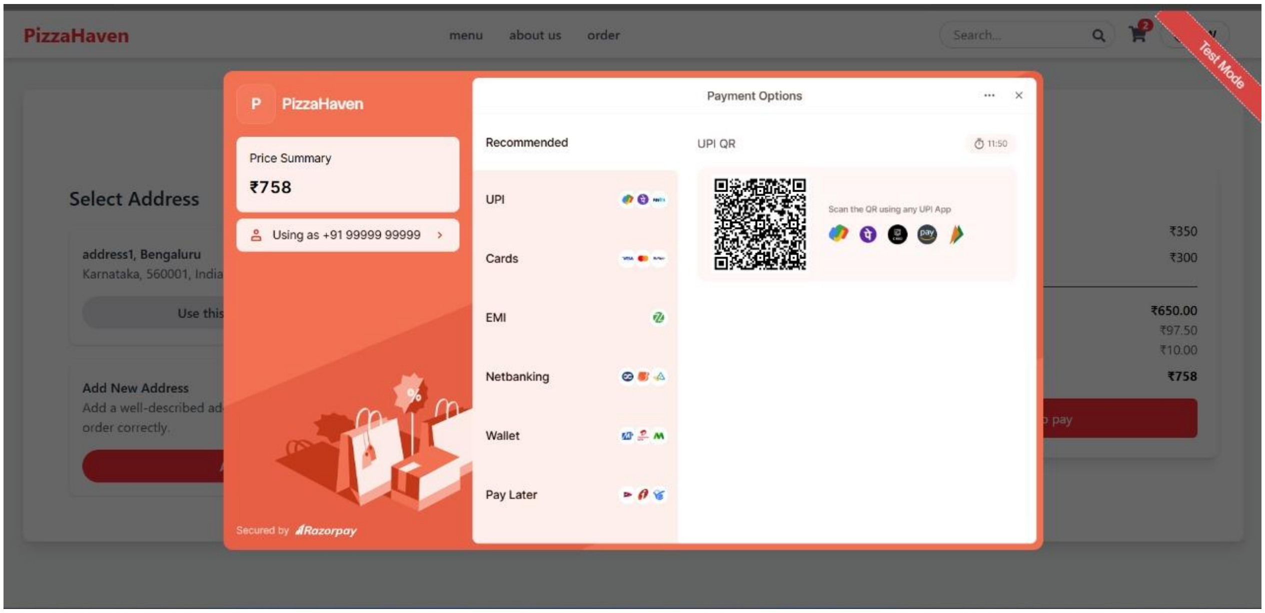
Cheese Pizza
Cheese Pizza
Price: ₹300
Qty: 1

Order Summary

Veg Delight (x1)	₹350
Cheese Pizza (x1)	₹300
Subtotal:	**₹650.00**

Item added to cart

Razor Pay Gateway



Price Summary
₹758

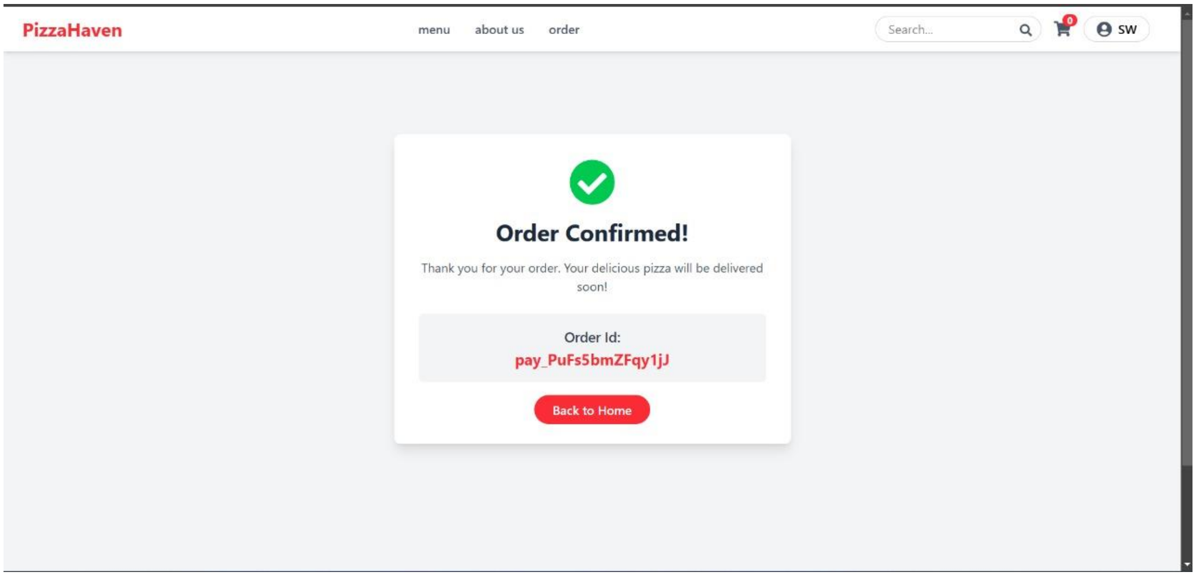
Using as +91 99999 99999

address1, Bengaluru
Karnataka, 560001, India

Add a well-described ad… order correctly.

₹350
₹300
₹650.00
₹97.50
₹10.00
₹758

Order Confirmation



Order Confirmed!

Thank you for your order. Your delicious pizza will be delivered soon!

Order Id:
pay_PuFs5bmZFqy1jJ

CONCLUSION

In conclusion, the Online Pizza Delivery website developed using React and Spring Boot, implementing microservices architecture and security features, has successfully met the goals of providing a seamless, secure, and efficient platform for users to place and manage their pizza orders. The integration of modern technologies ensures that the platform is scalable, secure, and user-friendly, addressing the key requirements of a modern online ordering system. Throughout the development process, we have gained valuable insights into the implementation of microservices and security protocols, and the experience of working with React and Spring Boot has enhanced our technical proficiency. The project has also allowed us to explore the challenges and solutions involved in building scalable and secure web applications.

We would like to express our deepest gratitude to Snehal Sir for his constant guidance, support, and invaluable advice throughout this journey. His expertise and encouragement have played a crucial role in the successful completion of this project. His suggestions and feedback have been instrumental in refining our approach, ensuring the quality and security of the system.

We are proud of what we have achieved and grateful for the opportunity to apply our learning to a real-world project. This experience has not only strengthened our technical skills but has also provided us with a deeper understanding of how to build robust and secure applications.

8. REFERENCES

1. **Spring Boot Documentation**
   URL: https://spring.io/projects/spring-boot

2. **React.js Documentation**
   URL: https://reactjs.org/docs/getting-started.html

3. **Redux Documentation**
   URL: https://redux.js.org

4. **Java Programming Language**
   URL: https://www.oracle.com/java/

5. **MySQL Workbench Documentation**
   URL: https://dev.mysql.com/doc/workbench/en/

6. **Spring Boot with React and Redux**
   URL: https://www.baeldung.com/spring-boot-react-and-redux

7. **Java Persistence API (JPA) Documentation**
   URL: https://www.eclipse.org/eclipselink/documentation/2.7/

8. **Swagger Documentation for Spring Boot** URL:
   https://springdoc.org/

9. **MDN Web Docs**
   URL: https://developer.mozilla.org/

10. **React Redux Integration Guide**
    URL: https://react-redux.js.org/