Food Ordering System

Name	Reg No.	Email	Campus
Nibedita Karmakar	20BDS0229	nibedita.karmakar2020@vitstudent.ac.in	Vellore
Sai Chandu Pidikiti	20BDS0246	pidikitisai.chandu2020@vitstudent.ac.in	Vellore
Soumyajit Maity	20BCE7195	soumyajit.20bce7195@vitap.ac.in	AP

1.1 INTRODUCTION

The Food Ordering System project is a comprehensive application developed using Spring Boot, Angular, and MySQL. This system aims to streamline the process of ordering food from restaurants, providing an efficient and user-friendly platform for both customers and restaurant owners. By leveraging the power of modern technologies, this project offers a seamless experience for placing food orders, managing menus, and processing payments.

The backend of the system is built on Spring Boot, a popular Java framework known for its simplicity and productivity. Spring Boot provides a robust and scalable foundation for developing web applications, offering features such as dependency management, auto-configuration, and easy integration with databases like MySQL. The frontend of the application is developed using Angular, a powerful JavaScript framework. Angular provides a rich set of tools and components for building responsive and interactive user interfaces. MySQL, a widely-used open-source relational database management system, is employed to handle the data storage and retrieval requirements of the application.

Throughout this project, the Spring Boot, Angular, and MySQL technologies work in harmony to create a robust, scalable, and efficient food ordering system. The combination of these technologies enables seamless communication between the frontend and backend, ensuring a smooth user experience and efficient management of data.

1.2 PURPOSE

Creating a comprehensive and efficient platform for food ordering, leveraging the technologies of Spring Boot, Angular, and MySQL.

The project seeks to simplify the process of ordering food from restaurants, eliminating the need for manual methods and providing a user-friendly digital platform.

By leveraging Angular's capabilities, the project aims to create an intuitive and responsive user interface, ensuring a seamless and enjoyable experience for customers. The utilization of MySQL as the database management system ensures the integrity and security of the stored data, protecting sensitive information such as customer details and payment transactions.

The project strives to offer customers the convenience of browsing menus, placing orders, and making payments online from the comfort of their own devices, enhancing accessibility to food services. The project aligns with the growing trend of digitalization in the food industry, encouraging restaurants to adopt technology solutions for improved customer service and operational efficiency.

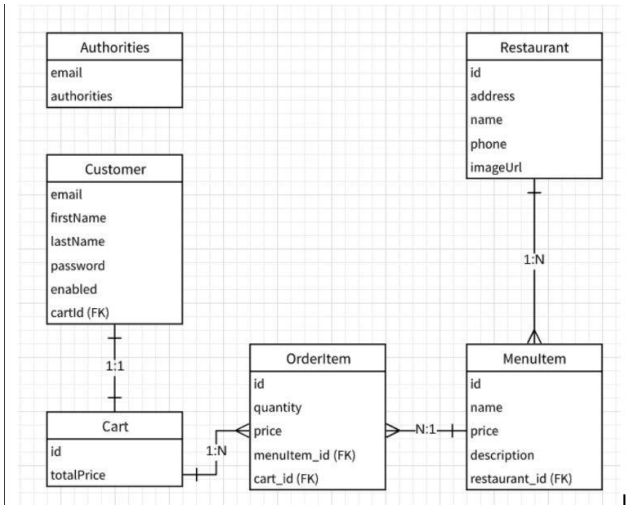
2. LITERATURE SURVEY

S.No.	Articles	Author	Problem	Proposed Solution
1	Improving Order Accuracy in Online Food Ordering Systems	Lee, J.	This article focuses on the problem of order accuracy in online food ordering systems. It highlights issues such as incorrect item selection, missing items, and miscommunication .	Implementing strategies to streamline order processing, such as optimizing backend systems, employing order batching techniques, or using predictive algorithms, can help reduce processing times and ensure timely delivery.
2	Analysis of Usability Issues in Online Food Ordering Systems	Johnson, R., Smith, A.	This paper explores the usability issues in online food ordering systems. It identifies problems such as complex user interfaces, limited customization options, and slow order processing.	Providing extensive customization options allows users to personalize their orders according to their preferences or dietary restrictions. This can include options for ingredient substitutions, portion sizes, or allergy alerts.
3	Integration Challenges in Food Ordering Systems: A Case Study	Chen, H., Wang, L.	This paper examines the integration challenges faced by food ordering	Integrating the food ordering system with other relevant systems, such as

			systems, particularly in integrating with external systems like payment gateways and delivery services. It discusses problems such as data discrepancies, manual workarounds, and inefficiencies.	payment gateways, inventory management, and delivery services, can help automate processes, reduce manual errors, and improve overall efficiency.
4	Enhancing Mobile User Experience in Food Ordering Apps	Garcia, M.	This article focuses on the challenges related to mobile user experience in food ordering apps. It discusses problems like difficult navigation, slow loading times, and inconsistent design.	Designing a user-friendly and intuitive interface can help simplify the ordering process. Clear and organized menus, visual cues, and straightforward navigation can improve the overall user experience.
5	Data Analytics for Performance Optimization in Food Ordering Systems	Zhang, Y., Liu, S.	This paper explores the use of data analytics for performance optimization in food ordering systems. It discusses problems related to slow order processing, peak-hour congestion, and delivery delays.	Leveraging data analytics can provide valuable insights into user behaviour, order patterns, and system performance. This information can be used to optimize the food ordering system, identify bottlenecks, and make data-driven improvements.

3.1 Theoretical Analysis

Block Diagram:



3.2 Software Requirements

- Visual Studio Code
- MongoDB
- MySQL Workbench
- Eclipse
- IDEA IntelliJ
- Spring boot Extension

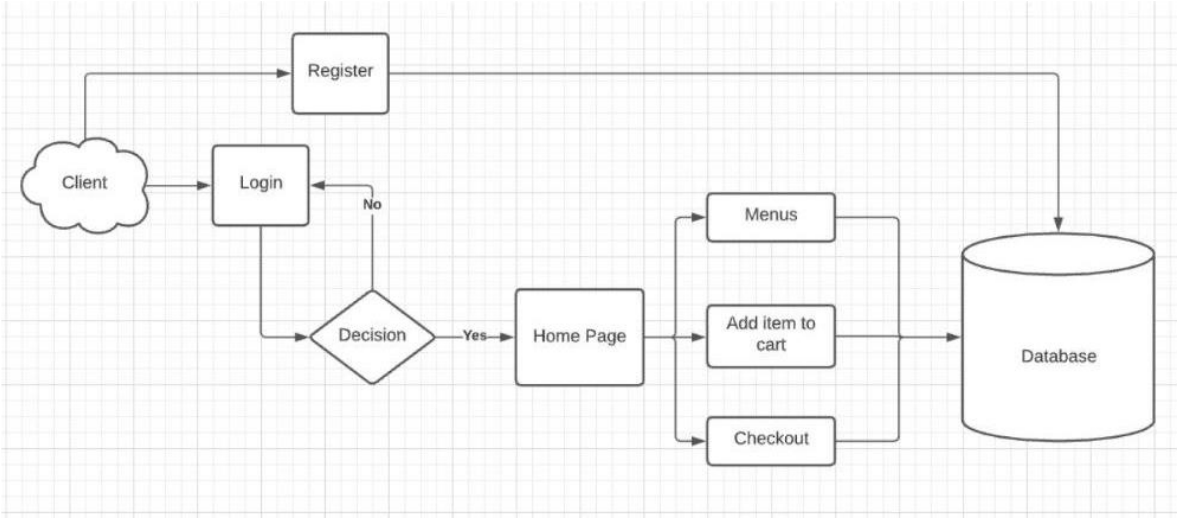
4. Experimental Investigations

During the development of the Food Ordering System project, various analyses and investigations are conducted to ensure an effective solution:

- User Requirements Analysis: Understanding customer and restaurant owner needs through surveys and research.
- System Architecture Analysis: Evaluating the scalability and performance of the chosen architecture.
- Security Analysis: Identifying potential threats and implementing security measures.
- Database Design and Optimization: Designing an efficient database schema for fast data retrieval.
- UI/UX Analysis: Testing and improving the user interface and experience based on feedback.

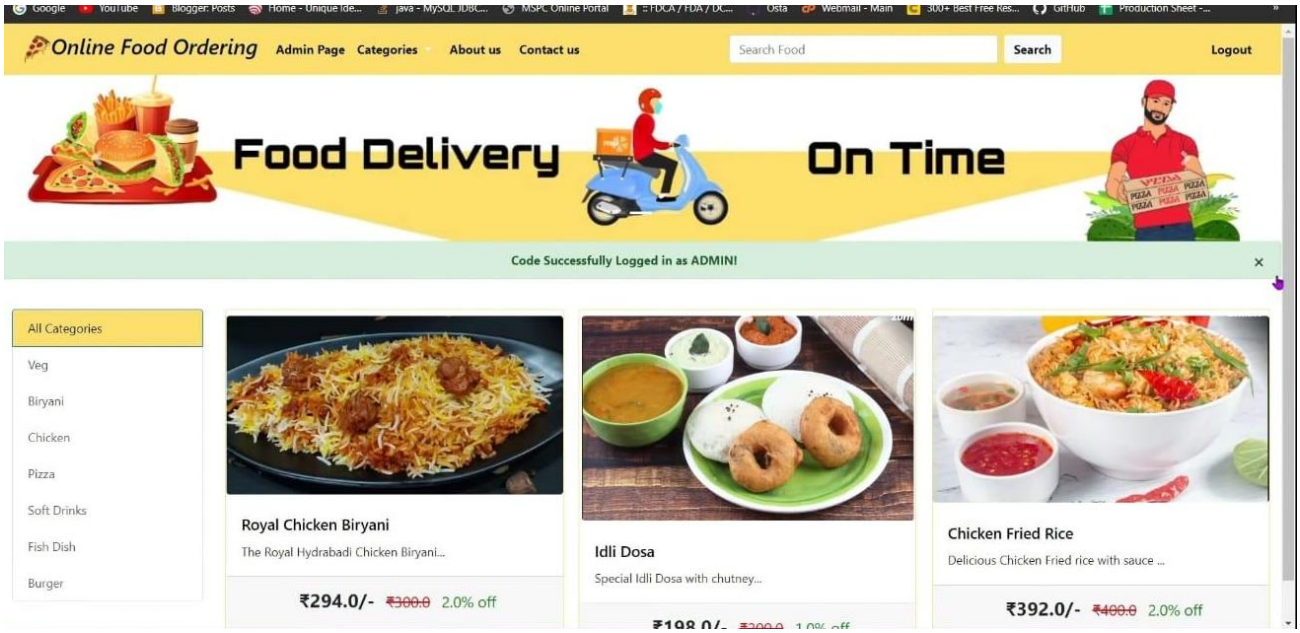
- Performance Testing: Evaluating system performance under different load conditions.
- Integration and Compatibility Analysis: Ensuring compatibility with browsers, operating systems, and external APIs.
- Continuous Improvement and Feedback Analysis: Collecting feedback and iteratively improving the solution.

5. Flowchart

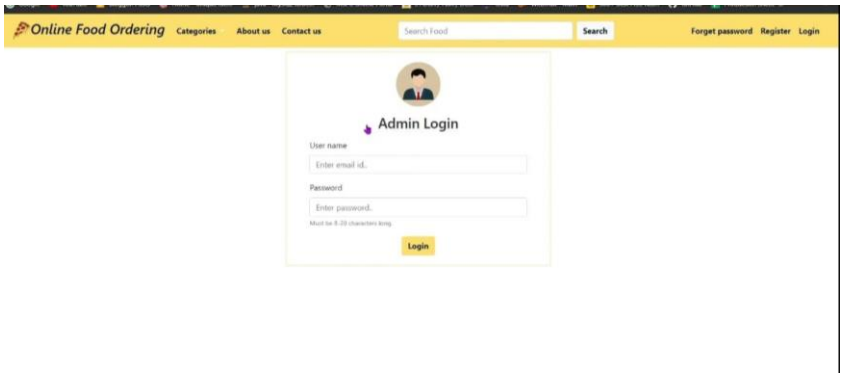


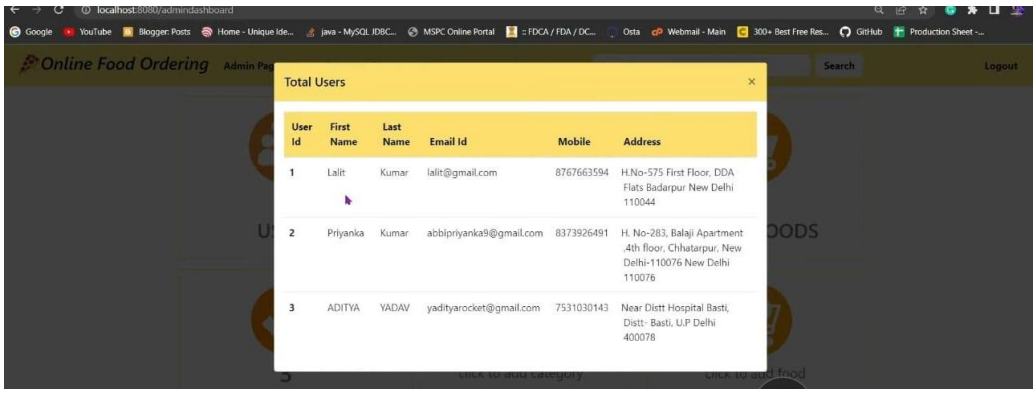
6. Result

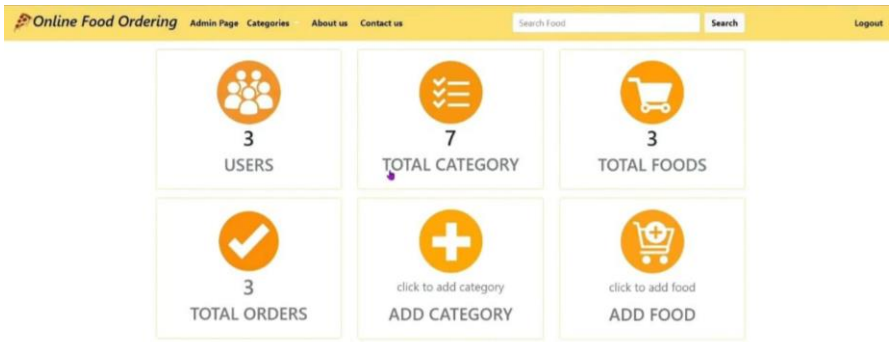
Website:

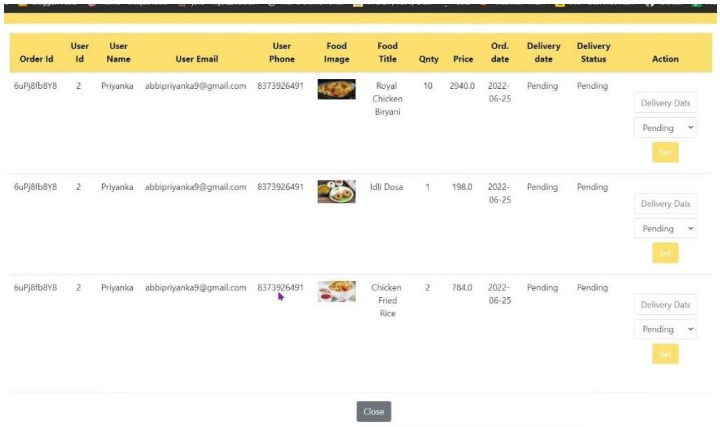


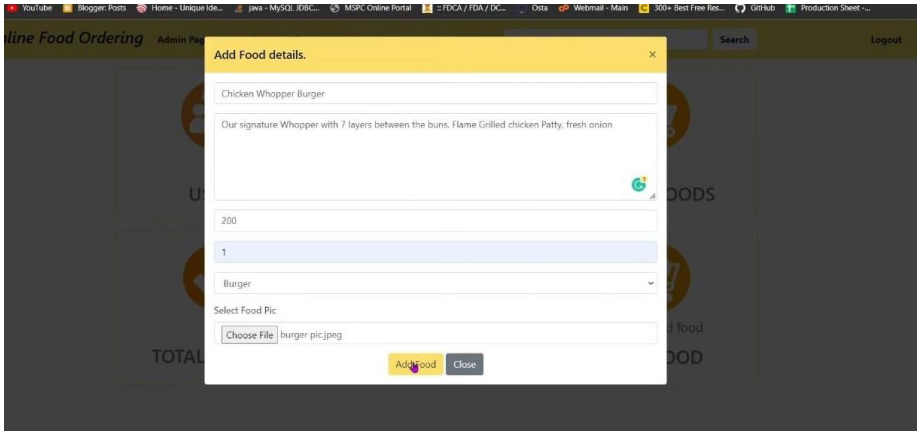
Admin Panel:

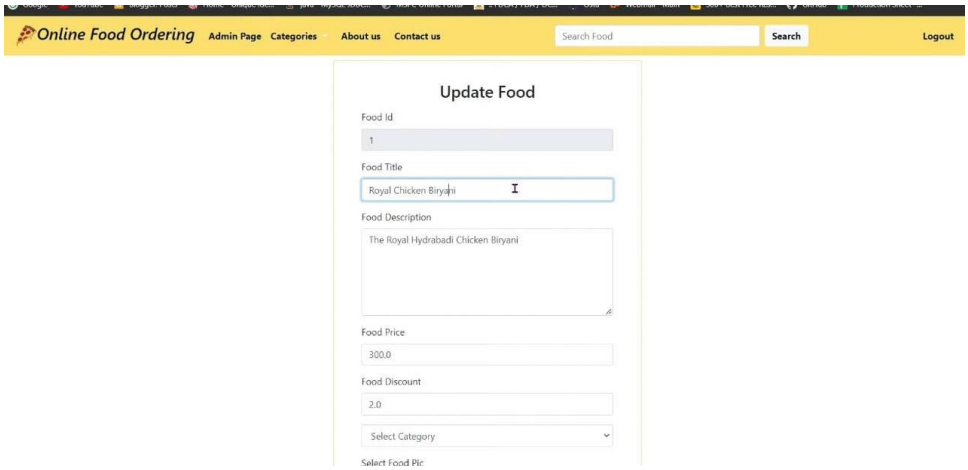

Online Food Ordering
Categories
About us
Contact us
Search Food
Search
Forget password
Register
Login
Admin Login
User name
Enter email id.
Password
Enter password.
Login


Online Food Ordering
Search
Logout
Total Users
User Id | First Name | Last Name | Email Id | Mobile | Address
1 | Lalit | Kumar | lalit@gmail.com | 8767663594 | H.No-575 First Floor, DDA Flats Badarpur New Delhi 110044
2 | Priyanka | Kumar | abbipriyanka9@gmail.com | 8373926491 | H. No-283, Balaji Apartment ,4th floor, Chhatarpur, New Delhi-110076 New Delhi 110076
3 | ADITYA | YADAV | yadityarocket@gmail.com | 7531030143 | Near Distt Hospital Basti, Distt- Basti, U.P Delhi 400078

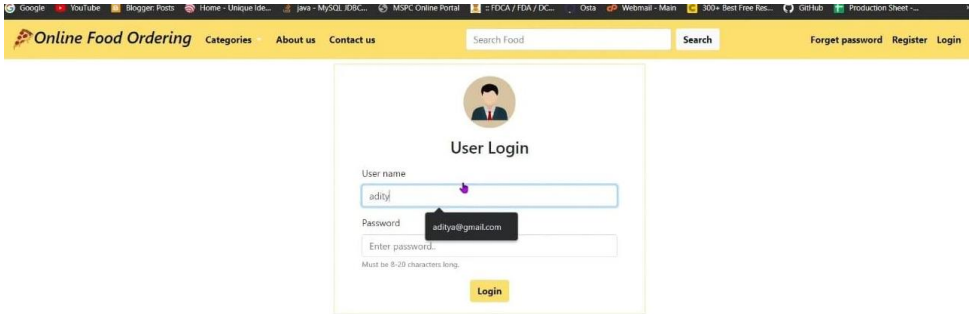

Online Food Ordering
Admin Page
Categories
About us
Contact us
Search Food
Search
Logout
3
USERS
7
TOTAL CATEGORY
3
TOTAL FOODS
3
TOTAL ORDERS
click to add category
ADD CATEGORY
click to add food
ADD FOOD

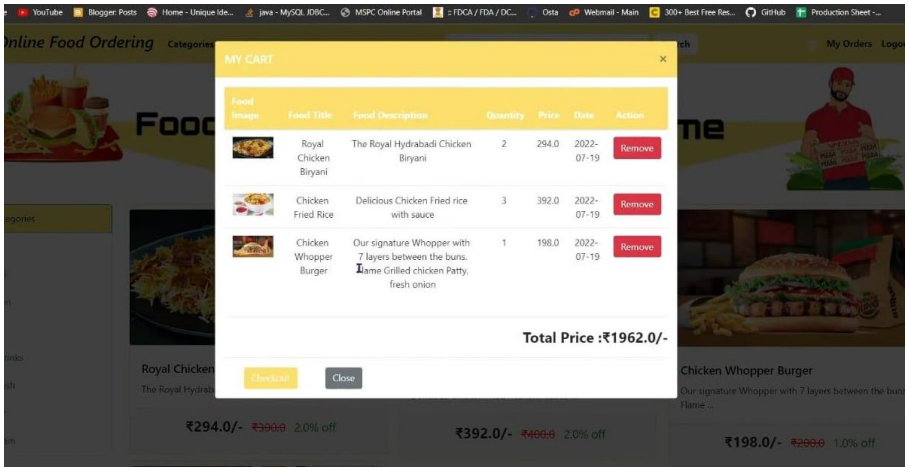

Order Id | User Id | User Name | User Email | User Phone | Food Image | Food Title | Qnty | Price | Ord. date | Delivery date | Delivery Status | Action
6uPj8fb8Y8 | 2 | Priyanka | abbipriyanka9@gmail.com | 8373926491 | Royal Chicken Biryani | 10 | 2940.0 | 2022-06-25 | Pending | Pending
6uPj8fb8Y8 | 2 | Priyanka | abbipriyanka9@gmail.com | 8373926491 | Idli Dosa | 1 | 198.0 | 2022-06-25 | Pending | Pending
6uPj8fb8Y8 | 2 | Priyanka | abbipriyanka9@gmail.com | 8373926491 | Chicken Fried Rice | 2 | 784.0 | 2022-06-25 | Pending | Pending
Delivery Date
Pending
Set
Close

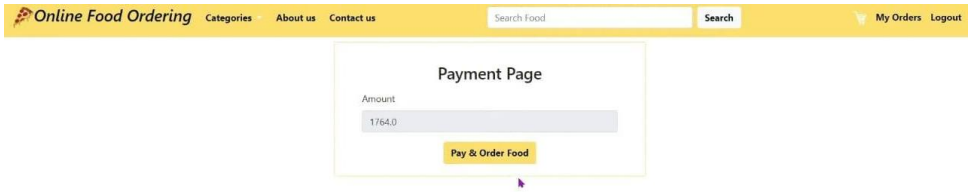

Add Food details.
Chicken Whopper Burger
Our signature Whopper with 7 layers between the buns, Flame Grilled chicken Patty, fresh onion
200
1
Burger
Select Food Pic
Choose File burger pic.jpeg
Add Food
Close

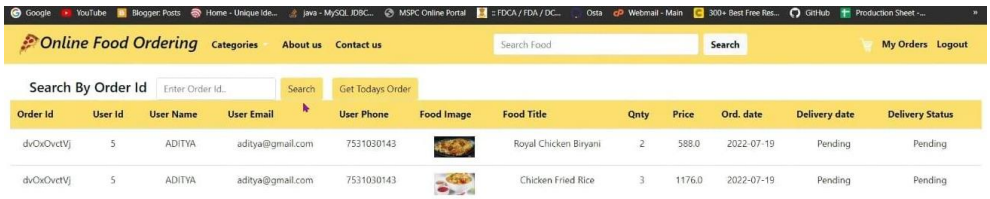

Online Food Ordering
Admin Page
Categories
About us
Contact us
Search Food
Search
Logout
Update Food
Food Id
1
Food Title
Royal Chicken Biryani
Food Description
The Royal Hydrabadi Chicken Biryani
Food Price
300.0
Food Discount
2.0
Select Category

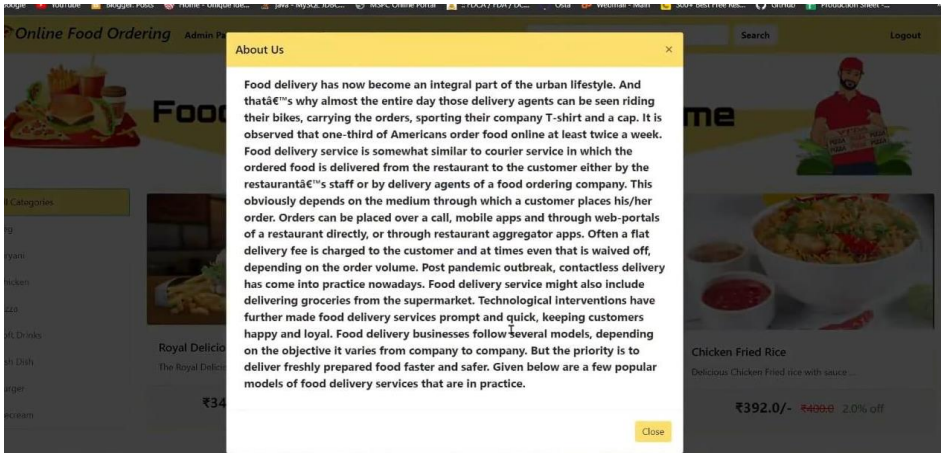
User Panel:


Online Food Ordering
Categories
About us
Contact us
Search Food
Search
Forget password
Register
Login
User Login
User name
adity
aditya@gmail.com
Password
Enter password..
Must be 8-20 characters long.
Login


Online Food Ordering
MY CART
Food Image
Food Title
Food Description
Quantity
Price
Date
Action
Royal Chicken Biryani
The Royal Hydrabadi Chicken Biryani
2
294.0
2022-07-19
Remove
Chicken Fried Rice
Delicious Chicken Fried rice with sauce
3
392.0
2022-07-19
Remove
Chicken Whopper Burger
Our signature Whopper with 7 layers between the buns. Flame Grilled chicken Patty, fresh onion
1
198.0
2022-07-19
Remove
Total Price :₹1962.0/-
Checkout
Close
Chicken Whopper Burger
₹294.0/-
₹392.0/-
₹198.0/-


Online Food Ordering
Categories
About us
Contact us
Search Food
Search
My Orders
Logout
Payment Page
Amount
1764.0
Pay & Order Food


Online Food Ordering
Categories
About us
Contact us
Search Food
Search
My Orders
Logout
Search By Order Id
Enter Order Id..
Search
Get Todays Order
Order Id | User Id | User Name | User Email | User Phone | Food Image | Food Title | Qnty | Price | Ord. date | Delivery date | Delivery Status
dvOxOvctVj | 5 | ADITYA | aditya@gmail.com | 7531030143 | Royal Chicken Biryani | 2 | 588.0 | 2022-07-19 | Pending | Pending
dvOxOvctVj | 5 | ADITYA | aditya@gmail.com | 7531030143 | Chicken Fried Rice | 3 | 1176.0 | 2022-07-19 | Pending | Pending


Online Food Ordering
Search
Logout
About Us
Food delivery has now become an integral part of the urban lifestyle. And thatâ€™s why almost the entire day those delivery agents can be seen riding their bikes, carrying the orders, sporting their company T-shirt and a cap. It is observed that one-third of Americans order food online at least twice a week. Food delivery service is somewhat similar to courier service in which the ordered food is delivered from the restaurant to the customer either by the restaurantâ€™s staff or by delivery agents of a food ordering company. This obviously depends on the medium through which a customer places his/her order. Orders can be placed over a call, mobile apps and through web-portals of a restaurant directly, or through restaurant aggregator apps. Often a flat delivery fee is charged to the customer and at times even that is waived off, depending on the order volume. Post pandemic outbreak, contactless delivery has come into practice nowadays. Food delivery service might also include delivering groceries from the supermarket. Technological interventions have further made food delivery services prompt and quick, keeping customers happy and loyal. Food delivery businesses follow several models, depending on the objective it varies from company to company. But the priority is to deliver freshly prepared food faster and safer. Given below are a few popular models of food delivery services that are in practice.
Close
Chicken Fried Rice
₹392.0/-

7. Advantages and Disadvantages

Advantages	Disadvantages
• Convenience	• Lack of personal interaction
• Time-saving-Saves time for both customers and restaurants, resulting in faster order processing and delivery.	• Technical issues: Online food ordering systems rely on technology, and any technical glitches or server downtime can disrupt the ordering process.
• Increased order accuracy	• Dependency on third-party platforms
• Access to a wider customer base	• Limited sensory experience
• Order customization	• Delivery challenges

8. Applications

- **Restaurants and Cafes:** Streamlines ordering process, enhances customer convenience, and improves operational efficiency.
- **Catering Businesses:** Simplifies catering process, enables online ordering, and improves communication with clients.
- **Grocery Stores and Supermarkets**: Allows customers to order groceries online and schedule convenient deliveries.
- **Cloud Kitchens and Virtual Restaurants**: Facilitates online ordering and delivery for establishments without a physical dining space.
- **Corporate Cafeterias:** Improves efficiency by allowing employees to pre-order meals and avoid long queues.
- **Food Delivery Platforms:** Connects customers with multiple restaurants for online ordering and delivery services.

9. Conclusion

The Food Ordering System project, developed using MySQL, Spring Boot, and Angular, offers a streamlined and user-friendly platform for customers to order food and for restaurant owners to manage their operations. The project ensures data integrity and security through MySQL, provides an efficient backend using Spring Boot, and delivers a responsive and intuitive frontend with Angular. Overall, it enhances the food ordering experience, promotes convenience and accessibility, and facilitates efficient management for restaurant owners.

10. Future Enhancements

- **Real-time Order Tracking:** Implement live order tracking for customers to monitor their deliveries.
- **Personalized Recommendations:** Provide customized menu suggestions based on customer preferences.
- **Loyalty and Rewards Program:** Introduce a loyalty program to incentivize customer retention.
- **Social Media Integration:** Allow customers to share orders and reviews on social media platforms.
- **Voice-Activated Ordering:** Enable voice-based ordering for hands-free convenience.
- **Payment Gateway Integration:** Expand payment options by integrating popular payment gateways.

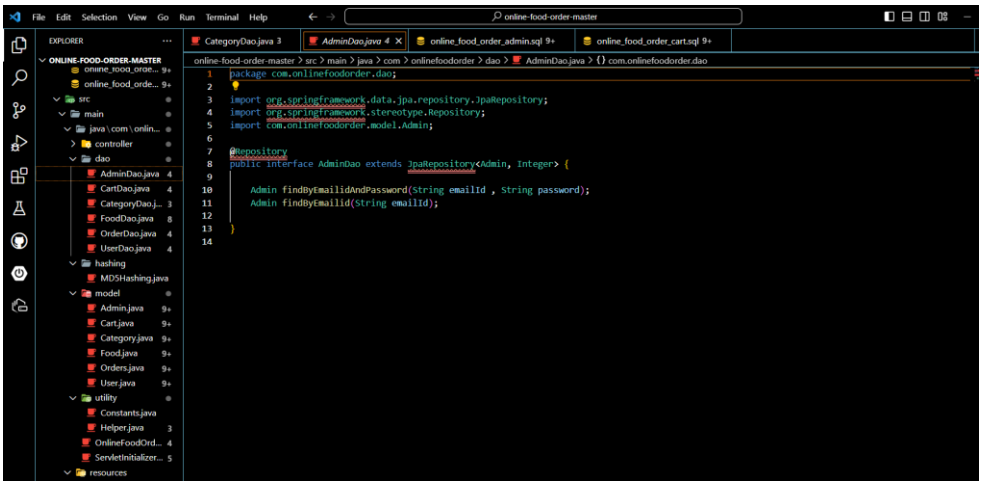
11. References

- Bhatia, A., & Tewari, R. (2019). Design and development of online food ordering system. International Journal of Advanced Research in Computer Science, 10(1), 100-104.
- Hidayanto, A. N., Wahyuni, E., & Setiawan, W. (2020). Implementation of online food ordering system using website and mobile application. Journal of Physics: Conference Series, 1441(1), 012027.
- Lam, S. S., Wong, D. S., & Lam, S. L. (2015). Exploring factors influencing the adoption of online food ordering and delivery services in enhancing customer experience. Procedia-Social and Behavioral Sciences, 195, 1350-1358.
- Oliveira, T., Thomas, M., & Espadanal, M. (2014). Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors. Information & Management, 51(5), 497-510.
- Patel, J., Varma, D., & Varma, N. (2016). Online food ordering system for restaurants. International Journal of Emerging Trends & Technology in Computer Science, 5(5), 325-328.
- Saranya, P., & Suryapriya, S. (2020). Design and implementation of online food ordering system using QR code. International Journal of Computer Sciences and Engineering, 8(5), 1-5.

- Sharma, A., & Gaba, A. (2020). Design and implementation of online food ordering system with artificial intelligence. In 2020 2nd International Conference on Advances in Science, Technology and Engineering (ICASTE) (pp. 1-5). IEEE.

SOURCE CODE

```
    private String pincode;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getFirstname() {
        return firstname;
    }

    public void setFirstname(String firstname) {
        this.firstname = firstname;
    }

    public String getLastname() {
        return lastname;
    }

    public void setLastname(String lastname) {
        this.lastname = lastname;
    }

    public String getEmailid() {
        return emailid;
    }

    public void setEmailid(String emailid) {
        this.emailid = emailid;
    }

    public String getPassword() {
        return password;
    }
```



```
package com.onlinefoodorder.dao;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
import com.onlinefoodorder.model.Admin;

@Repository
public interface AdminDao extends JpaRepository<Admin, Integer> {

    Admin findByEmailidAndPassword(String emailId , String password);
    Admin findByEmailid(String emailId);

}
```

```java
package com.onlinefoodorder.dao;

import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;
import com.onlinefoodorder.model.Food;

@Repository
public interface FoodDao extends JpaRepository<Food, Integer> {

    List<Food> findByNameContainingIgnoreCase(String name);
    List<Food> findByCategoryId(int categpryId);

    @Query(value="delete from Food f where f.categoryId = :categoryId")
    void deleteProductByCategoryId(@Param("categoryId") int categoryId);

}
```

```java
package com.onlinefoodorder.utility;

public class Constants {

    public enum DeliveryStatus {
        DELIVERED(status:"Delivered"),
        PENDING(status:"Pending"),
        ON_THE_WAY(status:"On the Way");

        private String status;

        private DeliveryStatus(String status) {
          this.status = status;
        }

        public String value() {
          return this.status;
        }

    }

}
```

```java
import java.util.List;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import com.onlinefoodorder.dao.FoodDao;
import com.onlinefoodorder.model.Cart;
import com.onlinefoodorder.model.Food;

public class Helper {

    @Autowired
    private FoodDao foodDao;

    public static String get10Words(String productDescription)
    {
        String[] str=productDescription.split(regex: " ");
        if(str.length>10)
        {
            String res="";
            for(int i=0;i<10;i++)
            {
                res=res+str[i]+" ";
            }

            return res+"...";
        }

        else
        {
            return productDescription+"...";
        }
    }

    public static double getFoodSellingPrice(double foodPrice,double foodDiscountPercent)
    {

        Double tenPercentValue=foodDiscountPercent*0.01;
```

```java
    }

    public static String getAlphaNumericOrderId(int n)
    {

        // chose a Character random from this String
        String AlphaNumericString = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
                                    + "0123456789"
                                    + "abcdefghijklmnopqrstuvxyz";

        // create StringBuffer size of AlphaNumericString
        StringBuilder sb = new StringBuilder(n);

        for (int i = 0; i < n; i++) {

            // generate a random number between
            // 0 to AlphaNumericString variable length
            int index
                = (int)(AlphaNumericString.length()
                        * Math.random());

            // add Character one by one in end of sb
            sb.append(AlphaNumericString
                      .charAt(index));
        }

        return sb.toString();
    }

    public double getProductsTotalSellingPrice(List<Cart> carts) {
        double totalPrice = 0;
        for(Cart c : carts) {
            Optional<Food> optinal = foodDao.findById(c.getFoodId());
            Food f = null;
            f = optinal.get();
            double foodPrice = getFoodSellingPrice(f.getPrice(),f.getDiscount());
            totalPrice = totalPrice + (foodPrice*c.getQuantity());
```

```java
package com.onlinefoodorder.hashing;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class MD5Hashing {

    //MD5 Hashing Algorithm
    public static String doHashing (String password)
    {
        StringBuilder sb = null;
        try {
         MessageDigest messageDigest = MessageDigest.getInstance(algorithm: "MD5");

         messageDigest.update(password.getBytes());

         byte[] resultByteArray = messageDigest.digest();

         sb = new StringBuilder();

         for (byte b : resultByteArray) {
          sb.append(String.format(format: "%02x", b));
         }


        } catch (NoSuchAlgorithmException e) {
         e.printStackTrace();
        }

        return sb.toString();
    }

}
```

```java
    public void setName(String name) {
        this.name = name;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public double getDiscount() {
        return discount;
    }

    public void setDiscount(double discount) {
        this.discount = discount;
    }

    public String getImagePath() {
        return imagePath;
    }

    public void setImagePath(String imagePath) {
        this.imagePath = imagePath;
    }
```

```java
package com.onlinefoodorder.dao;

import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.onlinefoodorder.model.Orders;

@Repository
public interface OrderDao extends JpaRepository<Orders , Integer> {

    List<Orders> findByOrderId(String orderId);
    List<Orders> findByUserId(int userId);
    List<Orders> findByOrderDate(String orderDate);
    List<Orders> findByOrderDateAndUserId(String orderDate, int userId);

}
```

[The entire code is attached in the GitHub Link]