

Governors State University

OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Fall 2022

Management System for a Restaurant

Prudveer Karne

Governors State University

Follow this and additional works at: <https://opus.govst.edu/capstones>

Recommended Citation

Karne, Prudveer, "Management System for a Restaurant" (2022). *All Capstone Projects*. 569.
<https://opus.govst.edu/capstones/569>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Capstone Project is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

MANAGEMENT SYSTEM FOR A RESTAURANT

By

Prudveer Karne

Bachelor of Science (MECs), Osmania University, 2017

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science



Governors State University
University Park, IL 60484

2022

ABSTRACT

The Management System for a Restaurant is a web-based application. The main aim is to provide better communication between customers and restaurants. By using this application, the user can order from any location. This application will reduce all the manual processes. So, users can view the restaurant menu from nearby locations, and can place an order and view the status of the order like cooking/packed/delivered.

There are three main actors in this application: Admin, Restaurant manager and User. The admin will create all menus with their corresponding price lists. Also, the admin will be able to view statistical reports for foods. Restaurant managers will receive orders from customers. Users/Customers can place orders using this application and they will be able to view available food items.

The system is designed as a 3-tier architecture. Each user must register with the system; after logging in successfully, they are able to perform operations. Separate login pages are provided for logging into the system. To develop this system, we use HTML, CSS, JavaScript, and bootstrap for the front end, MySQL database as the back end, and Java, JSP, and JDBC as middleware.

Table of Content

1	<i>Project Description</i>	1
1.1	Competitive Information.....	1
1.2	Relationship to Other Applications/Projects.....	1
1.3	Assumptions and Dependencies.....	1
1.4	Future Enhancements.....	2
1.5	Definitions and Acronyms.....	2
2	<i>Technical Description</i>	2
2.1	Project/Application Architecture.....	2
2.2	Project/Application Information flows.....	3
2.3	Interactions with other Applications.....	4
2.4	Capabilities.....	4
2.5	Risk Assessment and Management.....	4
3	<i>Project Requirements</i>	4
3.1	Identification of Requirements.....	4
3.2	Operations, Administration, Maintenance, and Provisioning (OAM&P).....	5
3.3	Security and Fraud Prevention.....	6
3.4	Release and Transition Plan.....	6
4	<i>Project Design Description</i>	7
5	<i>Project Internal/external Interface Impacts and Specification</i>	8
6	<i>Project Design Units Impacts</i>	16
6.1	Functional Area/Design Unit A.....	16
6.1.1	Functional Overview.....	16
6.1.2	Impacts.....	16
7	<i>Open Issues</i>	17
8	<i>Acknowledgements</i>	17
9	<i>References</i>	17
10	<i>Appendices</i>	17

1. Project Description

The Management System in a Restaurant is a web application that establishes communication between different restaurants and users. It consists of three main actors: Admin, Restaurant Manager, and User. The administrator is the owner of the site; it works under his supervision. Admin approves restaurants and he can view or delete them. Users are customers who search for restaurants, view products from restaurants and place orders from any location using the website's features. According to M. Mahaputra Hidayat, R. Dimas Adityo and Alek Siswanto [1], the design of a restaurant management system mainly depends on the number of items, orders, and payment details. In 2018, Singh Abhishek, R Adithya, Vaishnav Kanade and Salma Pathan [2] developed an advanced version of the online food ordering system. In which they added a notification module.

1.1. Competitive Information

There are many websites in the market that allow users to place food orders. Some of these sites include McDonald's, Starbucks, Door Dash, a food delivery service. However, our site is different from those existing ones. In our project multiple restaurants can add their products so that the user can search for items from a variety of restaurants and compare price of items. In our system users can view status of products and their previous orders. Users can also place orders without redirecting to third-party websites.

1.2. Relationship to Other Applications/Projects

As with other online food ordering projects available on the market, this project allows users to search and place orders. We developed this application using web technologies like HTML, CSS, JavaScript, JSP, JDBC, and to run the program we used a Tomcat web server.

1.3 Assumptions and Dependencies

Only registered users can access the system. The system provides the option for all users to register. Multiple users can access it simultaneously. The administrator is the owner of this application. His privileges include adding/removing restaurants and removing items. The application is developed using Java, so it is operating system-independent and server-independent.

1.4. Future Enhancements

The project can be updated soon if any new requirement arises because it is flexible in terms of expansion. In the future, we can develop search algorithms for fast retrieval of data. Furthermore, we could add a notification module where the user gets a mail or message whenever the restaurant adds a new item, and we could add real-time tracking using GPS. Using the recommendations of A. Bhargave, N. Jadhav, A. Joshi, P. Oke, and S. R. Lahane [3], we can receive orders digitally. In addition, users and restaurants can communicate using live chat.

1.5. Definitions and Acronyms

1) Admin: The administrator maintains the entire system. He can add or remove restaurants, he can view orders.

2) Restaurant Manager: The one who is responsible for adding/removing/deleting items, and he will accept or reject orders also.

3) User: The one who searches for products and can place orders.

4) View Items: Items added by the restaurants.

5) Ordered Items: Items ordered by the user.

2. Project Technical Description

The project contains three actors/categories. Each other is designed to perform particular operations specified in the proposal document. The actors are the admin, Restaurant Manager and Customer. The system architecture and technical architecture are explained in the following sections.

2.1. Application Architecture

The proposed system is a 3-tier application. We have three stakeholders in our web application: Admin, Restaurant Manager and Customer. In other words, System architecture can be described as the flow of application which is represented below in pictorial format.

Figure 1 shows System Architecture of a Restaurant Management System.

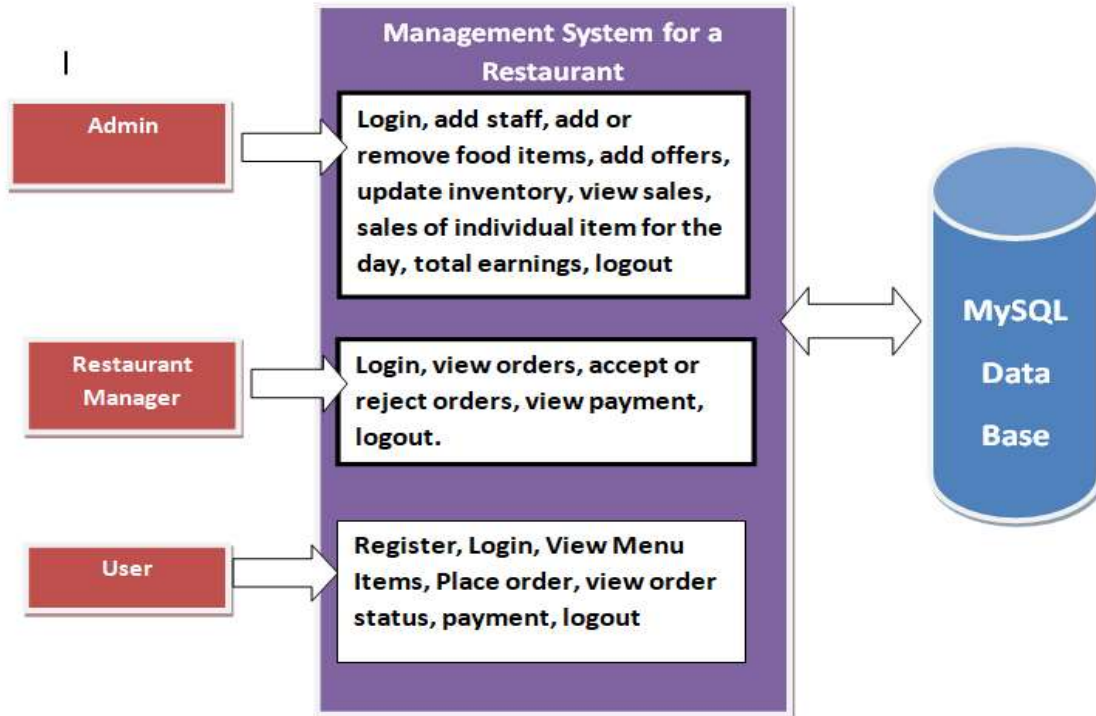


Fig. 1: System Architecture of Management System for a Restaurant

2.2. Technical Architecture:

The current application is being developed using a 3-tier architecture as a prototype. In this model, the web browser acts as a client. Tomcat server handles requesting an appointment and checking the status of visitors and a separate tier SQL server handles database functions like storing and viewing details. Figure 2 shows the technical architecture of our website.

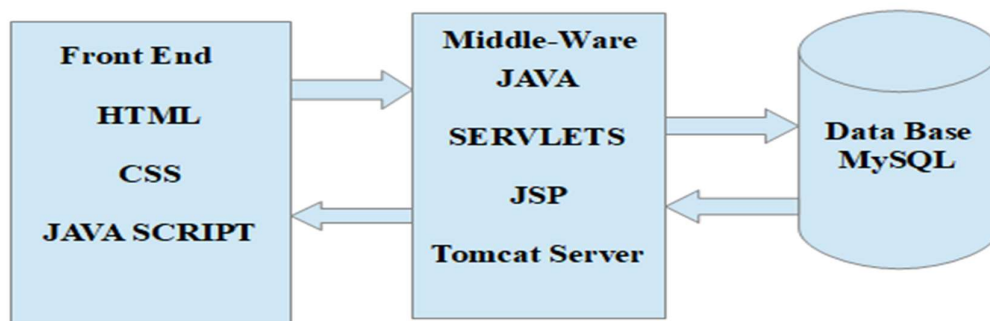


Fig. 2: Technical Architecture

2.3 Interactions with other applications

Customers can compare prices of products from different restaurants using the service, which can also interact with other online ordering systems like Door Dash. They can view restaurant details and feedback received from previous customers.

2.4. Capabilities

Various open-source software is involved in the development of this project. MySQL is used as a database and HTML; CSS and Bootstrap are utilized for application development. We can perform insert, delete, and update operations on the database dynamically. To retrieve item details, we use the select operation. Whenever a user registers with the system we perform an insert operation to store details. If the restaurant manager wants to change item details, we use the update operation. If the admin wants to delete a restaurant from our site, we call it a deletion operation.

2.5 Risk Assessment and Management

This project was implemented with high interest. If any risk arises, the development team will identify it quickly, analyse it, and monitor it until the end of the project. They will identify risks and mitigate them early to reduce the impact on the system, and they maintain a risk management log for future reference. Risks are divided into three types: high, medium, and low. All risks are handled according to priority, i.e., the highest priority is addressed first.

3. Project Requirements

3.1. Identification of requirements

The main aim is to provide better communication between customers and restaurants. By using this application, users can order from any location. It should be beneficial for both users and restaurants. Thus, I conducted a survey (online and offline) to note what various customers want in an online order portal. I met different restaurant owners and noted their priorities and basic requirements. Following the completion of the literature survey, I discussed the requirements with my professor, which we are planning to implement in this application.

1) User capability:

In our application, every user must register with the system before performing any operation. We have designed a user interface where the user can register, login and update his profile. To store details, we have used a MySQL database.

2) User Friendly:

To achieve and enhance the usability of the application, we have implemented HTML, CSS, and JavaScript for the user interface. HTML5 is used for the look and feel of the website. The application is created using HTML, so it is necessary to train its users on how to operate it.

3) User Experience:

This application is designed to handle heavy traffic. The application is running inside the server, and a database is maintained to store all the user data. To increase the user experience of the application we have used the most advanced technology to render the server-side, which is called JSP (Java Server Page). This application is quite fast and performs search operations very quickly.

3.2 Operations, Administrations, Maintenance, and Provisioning

The user could search for items and choose the most suitable one. The user is the main actor in this application, as the primary goal is to provide search and place orders online for users. Following are the operations performed by the user

- Register
- Login
- View Menu Items
- Place an Order
- Payment
- view order status
- View previous orders
- Logout

The other key role in the application is that of the restaurant manager. He is responsible for adding items, and accepting, or rejecting user orders. The following operations are performed by the restaurant manager

- Login
- add/remove food items
- View Order placed by Customer
- Accept/reject order requests
- view payment

- Logout

Admins in the application are responsible for the entire project. They monitor restaurants and users and take feedback from them to improve the application. Operations performed by the admin are as follows.

- Login
- Add restaurants
- Update/Delete items
- View Sales details
- View Payment details
- Logout

3.3. Security and Fraud Detection

Sensitive information is frequently requested on food delivery portals, which could be exploited to steal an applicant's identity or commit financial fraud. Data collected via an online form and then sent or kept insecurely may be exposed to exploitation. While information from paper applications can be used for unlawful reasons, online security flaws expose applicant information to a wider audience. We can detect any fraud operations in the following two steps.

- 1) Look out for the number of orders placed by the customer per day.
- 2) Check how many orders the restaurant delivers each day.

Suppose a customer is asked to pay an additional amount for a particular item, then that restaurant is considered a fraud restaurant. Even if the restaurant is legitimate and the offers are genuine and the payments are fair, if the charges for the item are not legal, you will be charged. You are not aware of it, e.g., if the restaurant charges additional amounts in the form of delivery charges, etc., you could be liable for consumer fraud.

3.4 Release and Transition Plan

The project is developed as a client-server model, and it is independent of operating systems, technologies, and servers so it is ready to deploy on servers or in operating systems. To run the

application all the required software is installed and ready to test. Based on user requirements, upcoming releases are scheduled. After successful deployment, there will be maintenance and upgrades every 3 months.

4. Project Design Description

The UI in the application has been divided into an administrative user interface and an operational user interface. The administrative user interface is responsible for maintaining the entire system and database and this UI is helpful for performing insert, delete and update operations on the database. User interfaces allow users to store data whenever they need it, and we can retrieve it whenever it is needed. Using the user interface, they can perform various DML operations on the database. After careful examination, the system was developed with 3 modules.

- 1) Admin
- 2) Restaurant Manager
- 3) User

The admin module will perform following operations

- Login
- Add restaurants
- Update/Delete items
- View Sales details
- View Payment details
- Logout

The Restaurant Manager will perform following operations

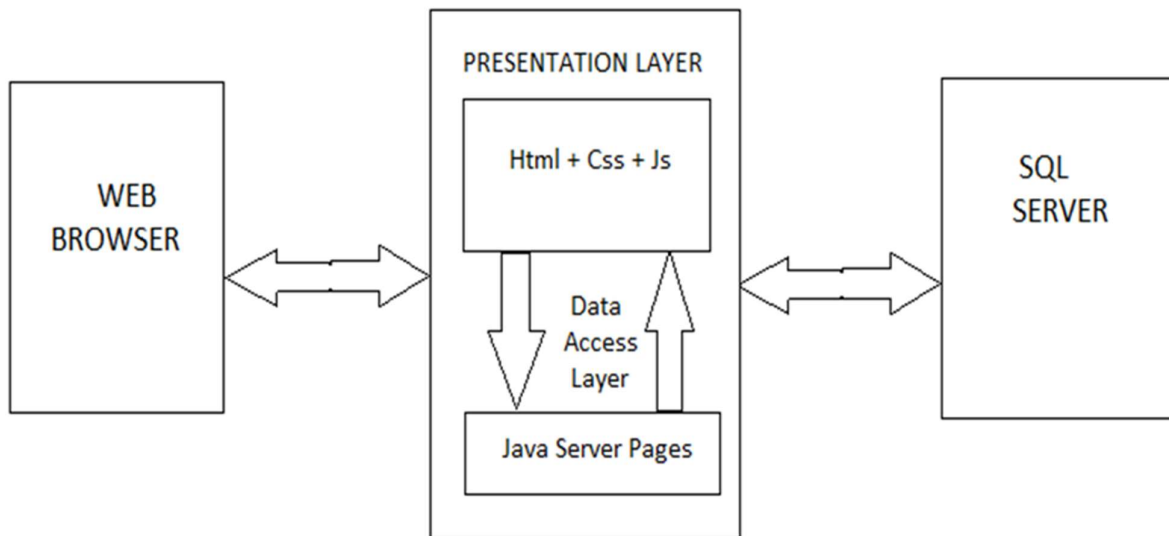
- Login
- add/remove food items
- View Order placed by Customer
- Accept/reject order requests
- view payment
- Logout

User will perform following operations

- Register
- Login
- View Menu Items
- Place an Order
- Payment
- view order status
- View previous orders
- Logout

5. Internal/external Interface Impacts and Specification

To design interfaces, we used different technologies that helped us create user-friendly interfaces. They are HTML, CSS, JavaScript, Java [5], JSP, JDBC and MySQL.



For User Interface development we used HTML, for styling we used CSS, and to perform client-side validations we relied on JavaScript. For better navigation we used Bootstrap[4].

MySQL is used as a back-end database. MySQL is simple and open-source software. We can easily perform various SQL operations

Java Server Pages (JSP) and Java Database Connectivity (JDBC) act as middleware. These are responsible for receiving requests from the front-end and handling database operations.

Tomcat server is used for deployment. Tomcat is an open-source web server designed to run Java-based web applications.

The waterfall model is used to develop our application because it is suitable for small, time-bound applications. The waterfall model is sequential. The following diagram shows the waterfall model used in this application.

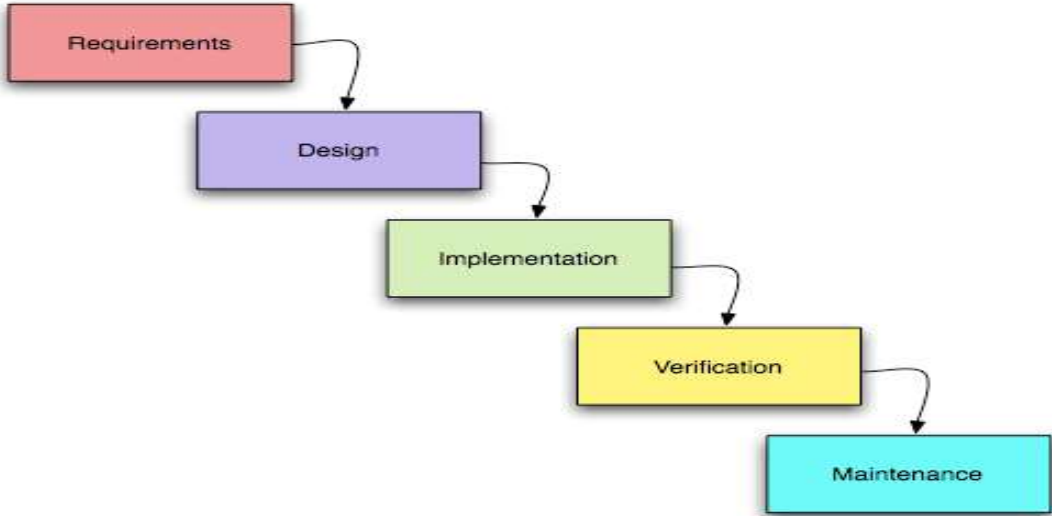


Fig 3. SDLC model for Management System in a Restaurant

Screen shots

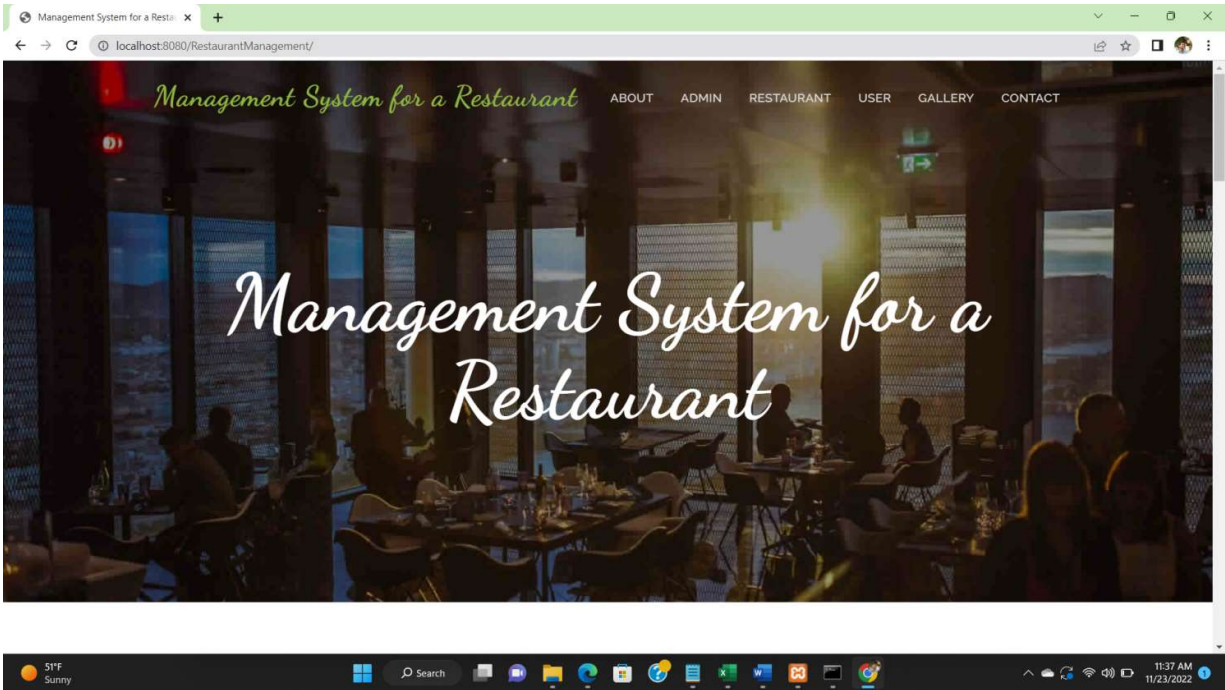


Fig. 4: Home Page

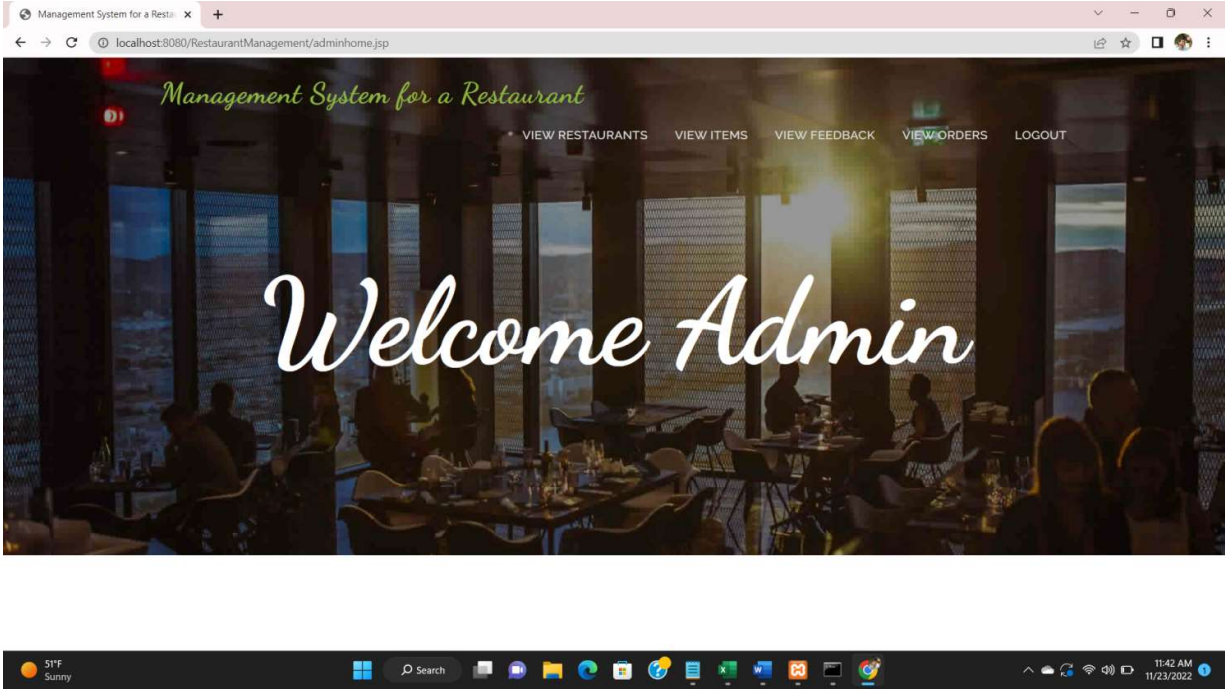


Fig. 5: Admin Home Page

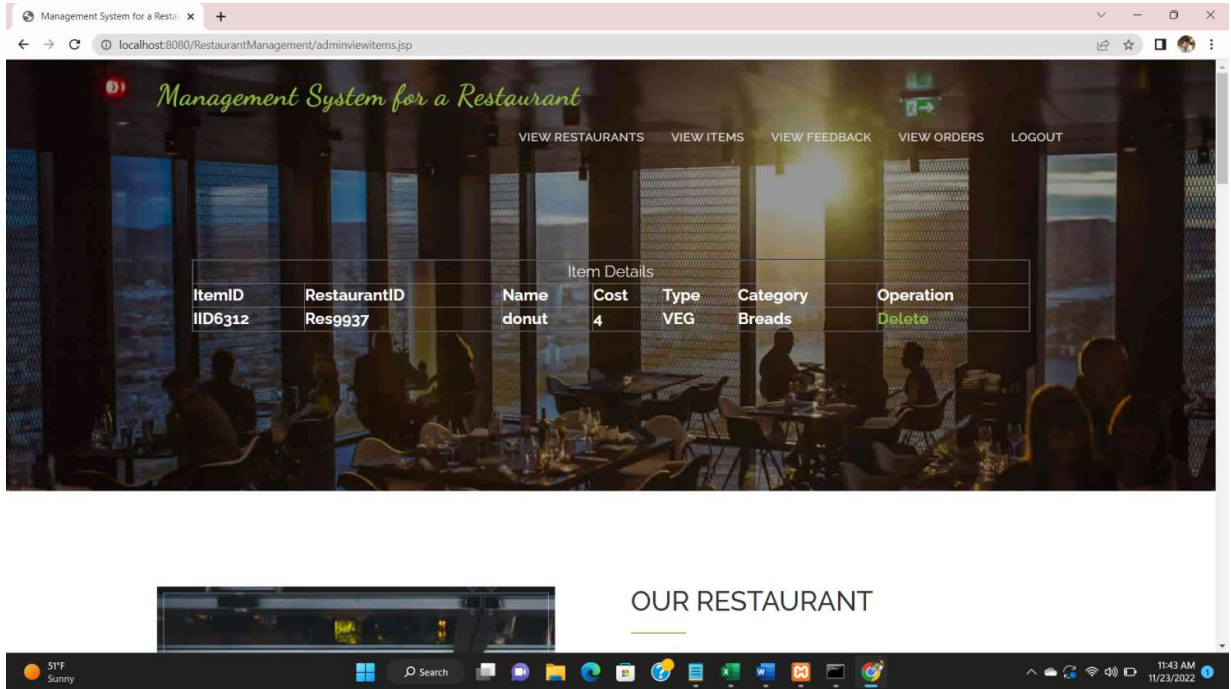


Fig. 6: Admin view Items

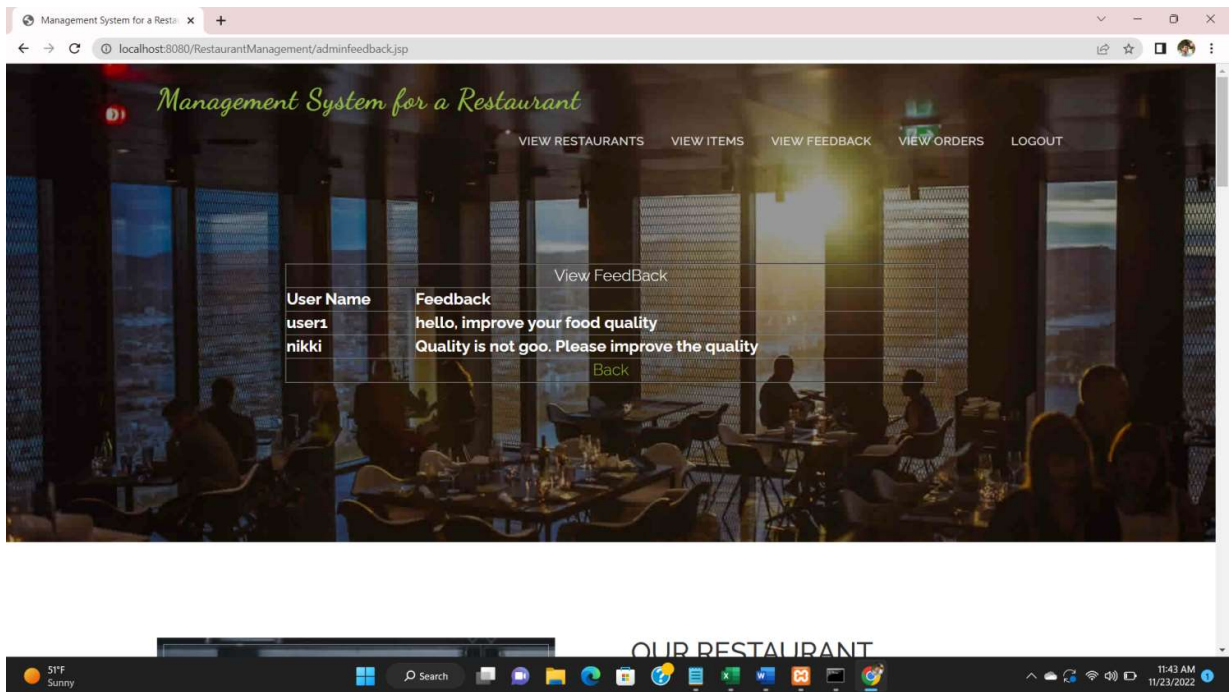


Fig. 7: Admin view User Feedback

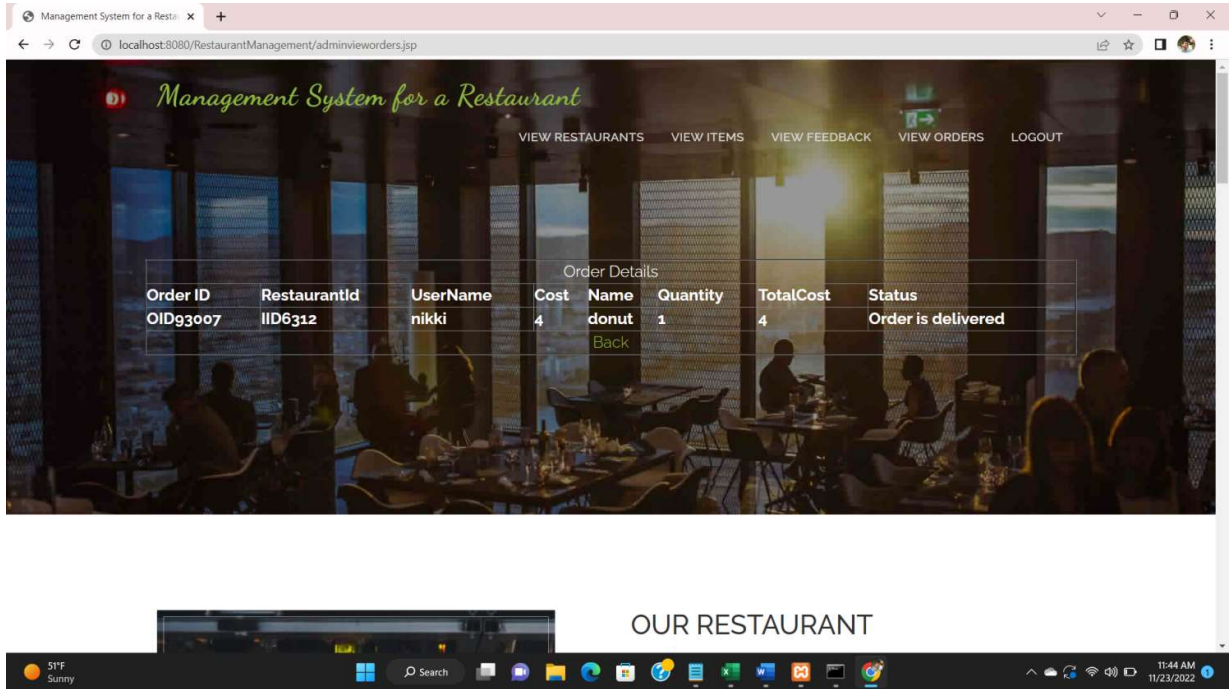


Fig. 8: Admin view Orders

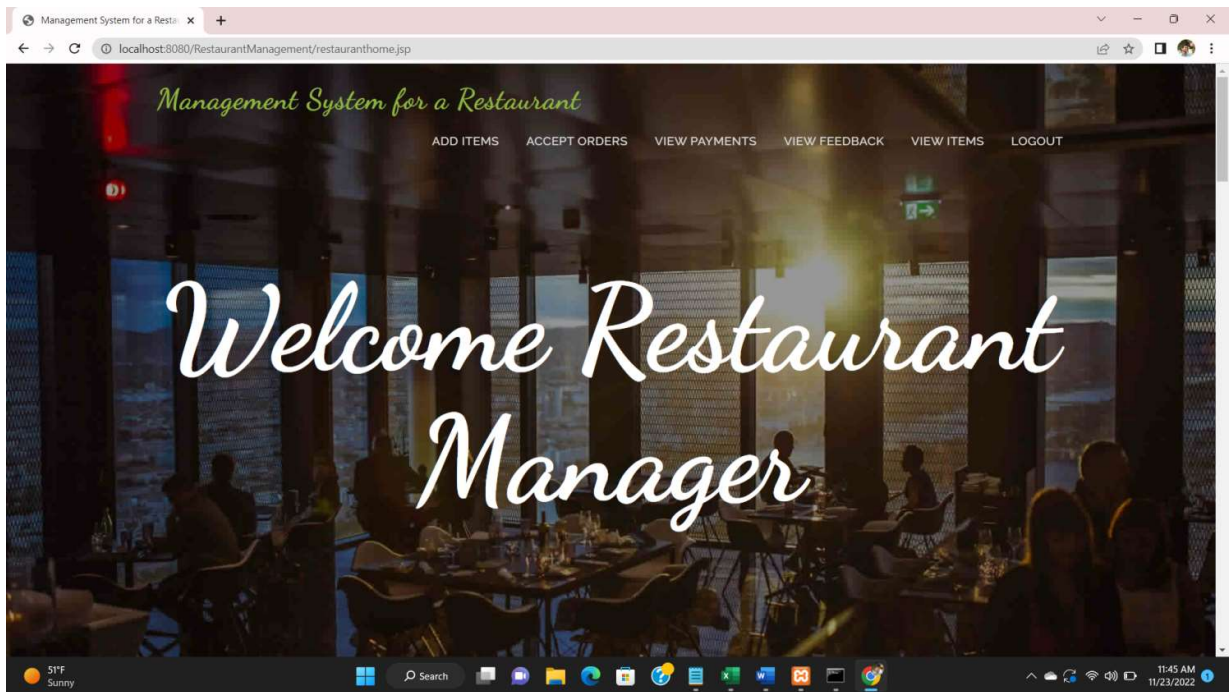


Fig. 9: Restaurant Manager Home page

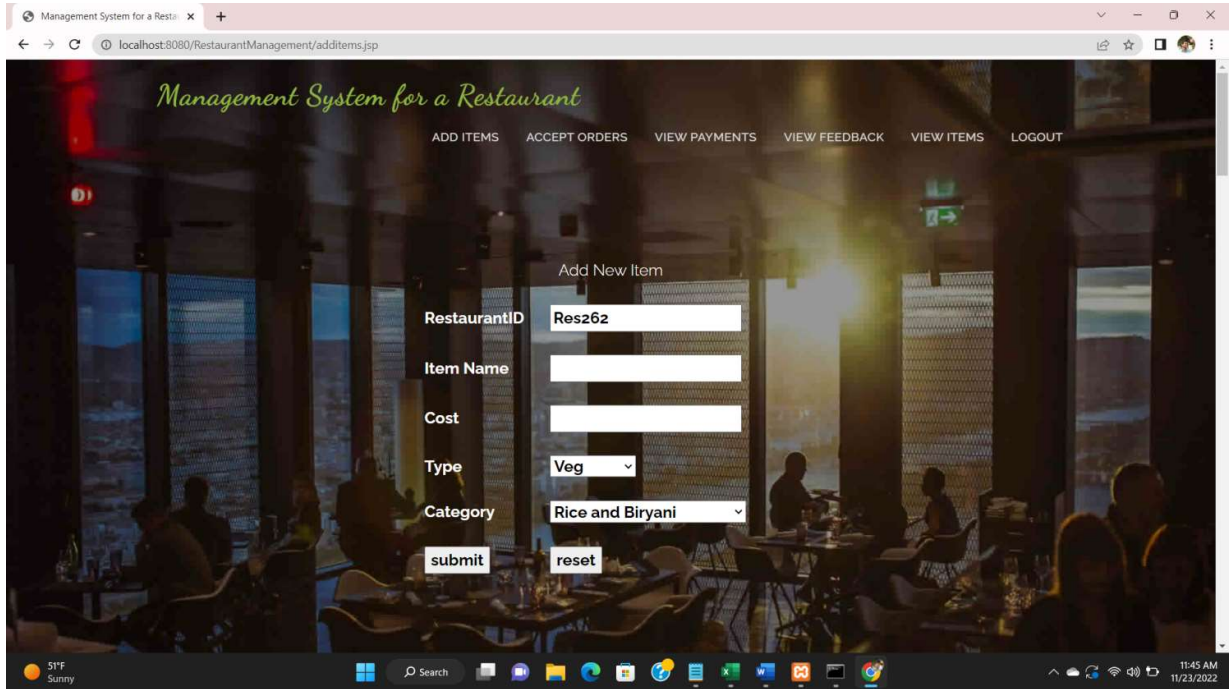


Fig. 10: Restaurant Manger add items page

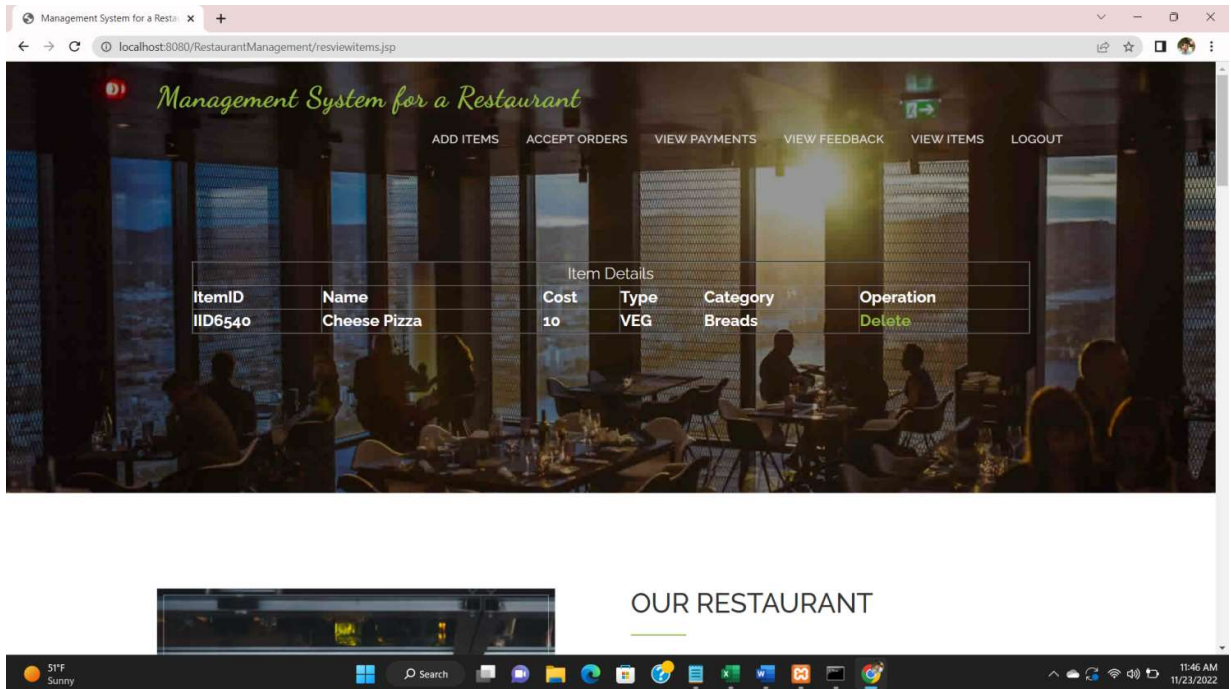


Fig. 11: Restaurant Manger view Items

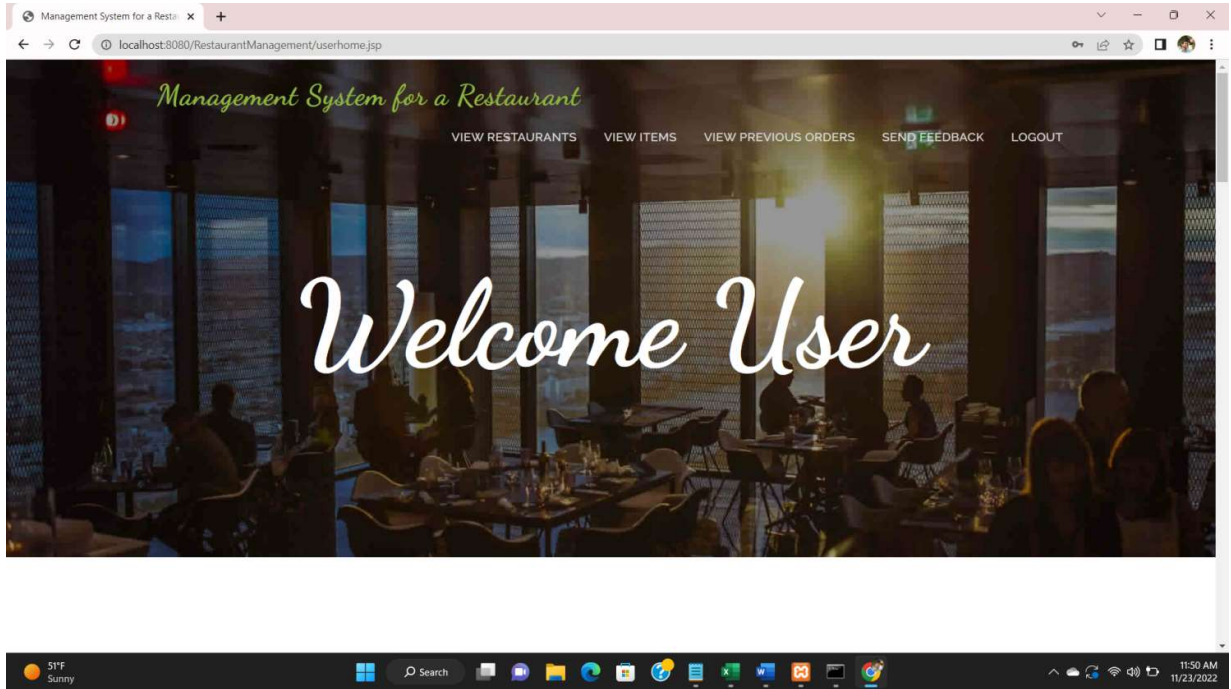


Fig. 12: User Home page

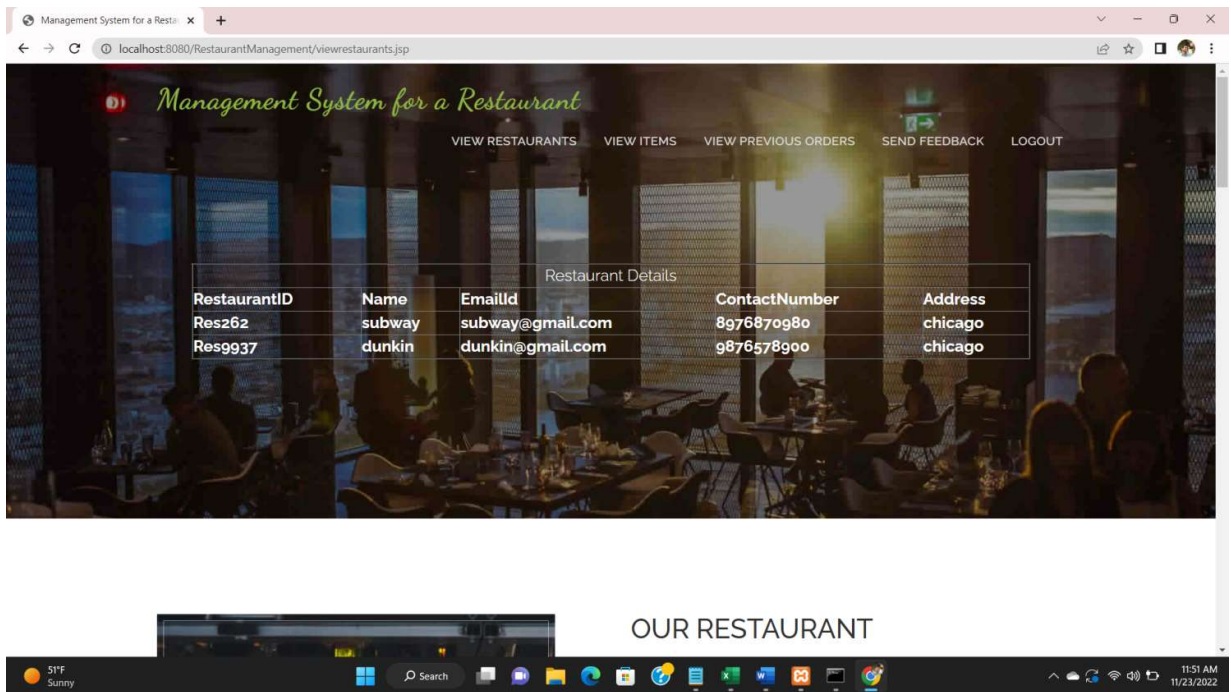


Fig. 13: User view Restaurant details

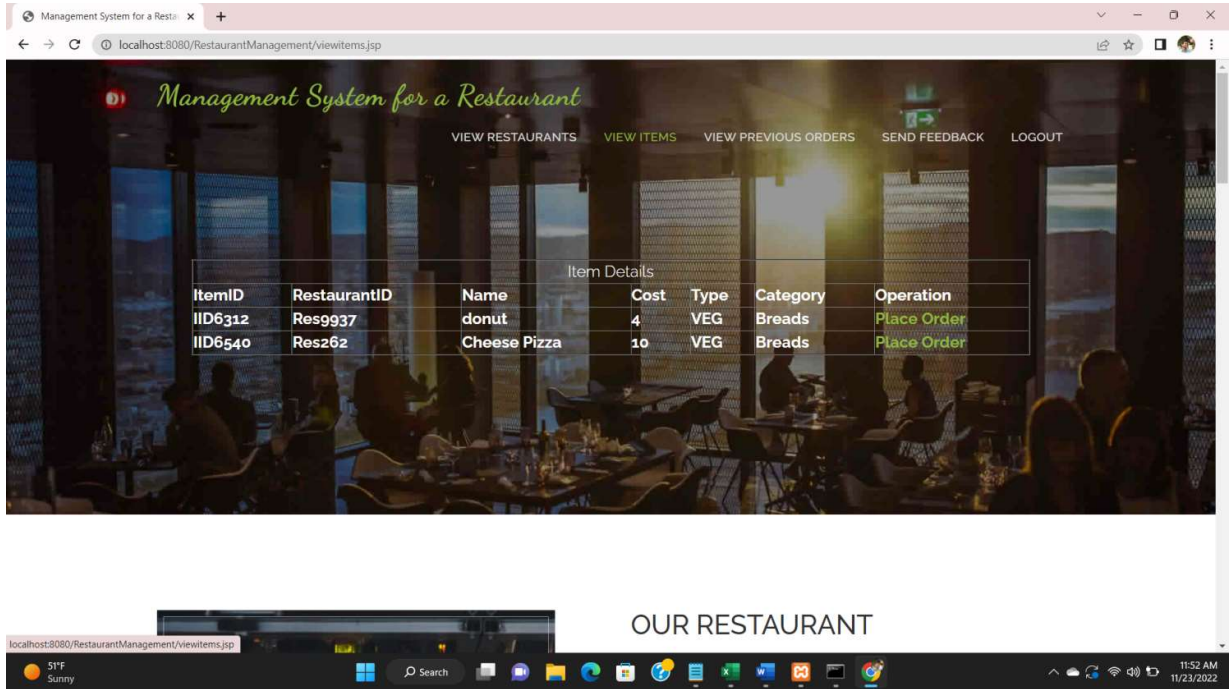


Fig. 14: User View Items

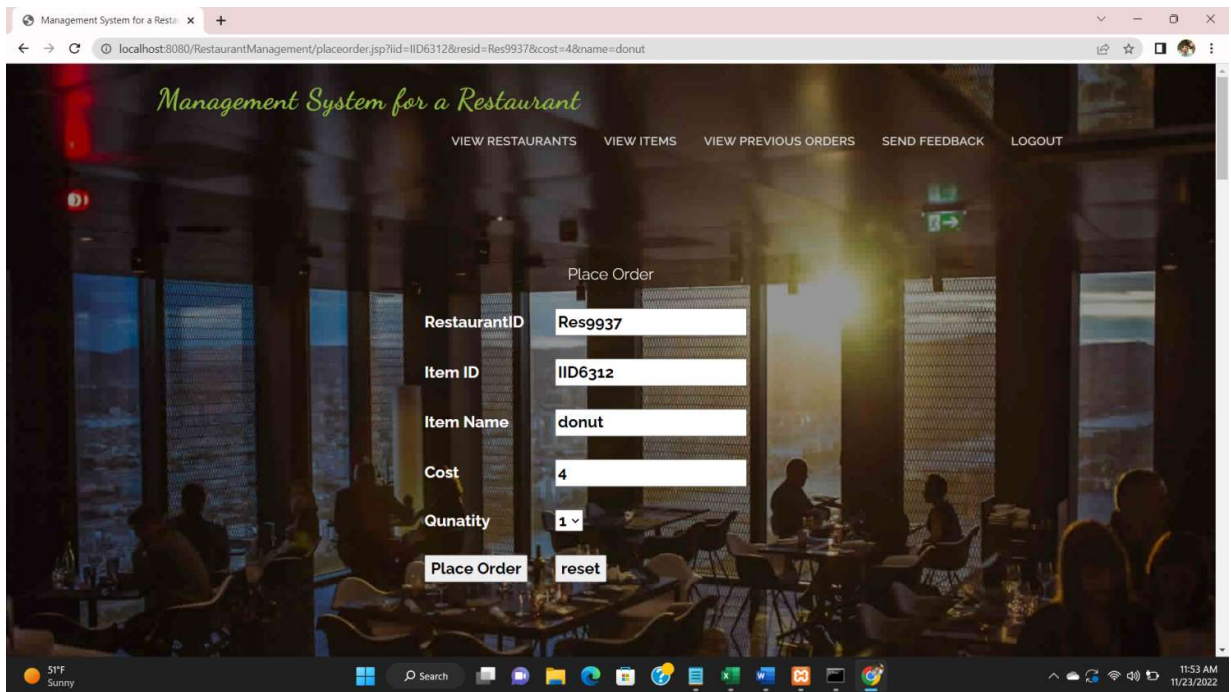


Fig. 15: User place orders Page

6. Design Units Impacts

Users can sign up as customers, can update profiles and can search for restaurants.

- Users can place an order from any location.
- Restaurant managers can register, login and can add multiple items.
- The restaurant manager can view orders and can accept or reject them, as well as view payment details.
- Only registered users are allowed to see restaurant details and can place orders.

6.1. Functional Area A/Design Unit A

6.1.1. Functional Overview

In software development, the requirements specification is one of the most important aspects. We have used many software and hardware technologies to develop this application. Following are the software and hardware technologies used in this application.

Software Requirements:

- Web Presentation : HTML, CSS, and Bootstrap
- Client – side Scripting : JavaScript
- Programming Language : Java
- Web based Technologies : Servlets, JSP
- Database Connectivity : JDBC
- Backend Database : MySQL
- Operating System : Window's
- Web Server : Tomcat
- IDE : Eclipse IDE

Deployment Requirements

- **Server:** Apache Tomcat
- **Database:** MySQL Database

6.1.2. Impacts

There is a huge impact in providing the required items to the user and making their search successful, but in terms of functional impacts, there are no wide area effects on the application. Later, the database

must be increased to take care of the huge number of users enrolled in the system. They will place multiple orders and manage the connection issues with the MySQL database.

7. Open Issues

In the modern era, users can access applications from remote locations so they may face some risks. There is a high risk of finding fraudulent restaurants, fraudulent users placing orders, delivering food to the wrong address, and payment fraud. To reduce the risk impact, a proper risk management team must be maintained.

8. Acknowledgments

I would like to thank my professor Yunchuan Liu. Without his assistance, this project could not have been completed. I also thank my mentors who helped to finish this project, guiding me to complete this project through all major decisions and believing in our capabilities.

9. Appendices

- 1) <https://www.tutorialspoint.com/jsp/index.htm>
- 2) <https://www.guru99.com/software-development-life-cycle-tutorial.html>
- 3) <https://www.geeksforgeeks.org/jdbc-tutorial/>
- 4) <https://html.com/>
- 5) <https://www.mysqltutorial.org/>

10. References

- 1) M. Mahaputra Hidayat, M. M., Dimas Adityo, R., & Siswanto, A. (2020, February 1). *Design of Restaurant Billing System (E Bill Resto) by Applying Synchronization of Data Billing in Branch Companies to Main Companies Based on Rest API*. IEEE Xplore.
<https://doi.org/10.1109/ICoSTA48221.2020.1570615039>

2) Adithya R, Abhishek Singh, Salma Pathan, & Vaishnav Kanade. (2017). Online Food Ordering System. *International Journal of Computer Applications*, 180(6), 22–24.

<https://www.ijcaonline.org/archives/volume180/number6/28805-2017916046>

3) Bhargave, A., Jadhav, N., Joshi, A., Oke, P., & Lahane. (2013). Digital Ordering System for Restaurant Using Android. *International Journal of Scientific and Research Publications*, 3(4).

<https://www.ijsrp.org/research-paper-0413/ijsrp-p1605.pdf>

4) Otto, M. (2019). *Introduction*. Getbootstrap.com. <https://getbootstrap.com/docs/4.3/getting-started/introduction/>

5) Arnold, K., Gosling, J., & Holmes, D. (2000). *The Java Programming Language*. Addison-Wesley Professional.