ONLINE FOOD ORDERING SYSTEM

Project Report Submitted By

JAYALEKSHMI JAYAKUMAR

Reg. No: AJC17MCA-1022

In Partial fulfillment for the Award of the Degree Of

**INTEGRATED MASTER OF COMPUTER APPLICATIONS
(INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**

AMAL JYOTHI COLLEGE OF ENGINEERING, KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE,
Accredited by NAAC with "A" grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**ONLINE FOOD ORDERING SYSTEM"** is the bonafide work of **JAYALEKSHMI JAYAKUMAR (Reg.No:AJC17MCA-I022)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017 - 22.

Sr. Elsin Chakkalackal. S.H

Internal Guide

Rev.Fr.Dr.Rubin Thottupurathu Jose

Coordinator

Rev.Fr.Dr. Rubin Thottupurathu Jose

Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report **“ONLINE FOOD ORDERING SYSTEM”** is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2017-2022.

Date: **JAYALEKSHMI JAYAKUMAR**

KANJIRAPPALLY **Reg. No: AJC17MCA-I022**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department and project coordinator **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us and their valuable suggestions for overwhelming concern and guidance from the beginning to the end of the project.I would also like to express sincere gratitude to my guide**, Sr. Elsin Chakkalackal.S.H** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

JAYALEKSHMI JAYAKUMAR

ABSTRACT

The main objective of the Online Food Ordering System is to manage the details of Item Category, Food Delivery Address, Order, and Shopping Cart. The purpose of the project is to build an application program to reduce the manual work for managing the Item Category, Food, Customer, Delivery Address etc. The project Online Food ordering system is a web based application that allows the administrator to handle all the activities online quickly and safely. It overcomes the disadvantages of the traditional queuing system.

Our proposed system is a medium to order online food hassle free from restaurants as well as mess service. This system improves the method of taking the order from customer. The online food ordering system sets up a food menu online and customers can easily place the order as per their wish from their favorite Restaurants. Also with a food menu, customers can easily track the orders.

CONTENT

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The "Online Food Ordering System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user- friendly. Online Food Ordering System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources.

1.2 PROJECT SPECIFICATION

The main objective of the Project on Online Food Ordering System is to manage the details of Food Item, Category, Customer, Order, and Confirm Order. It manages all the information about Food Item, Payment, Confirm Order, and Food Item. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Food Item, Category, Payment, and Customer. It tracks all the details about the Customer, Order, and Confirm Order.

The system includes 4 modules. They are:

1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can view all the registered customers and restaurants, can able to approve or reject users and restaurants .

2. Customer Module

Customer can register and they can upload order variety of foods and do secure online payment.

3. Restaurant Module

Restaurant can register and they can add variety of foods and add and assign delivery boys. They can also view the payment reports daily.

4. Delivery Boy Module

Delivery boy can login to the site and they can view order details and can update the status of booking status.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information about the Online Food Ordering System to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals.

The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal. Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

In the existing system the route details for delivering the food are not done manually but in proposed system we have to computerize the route details using this application.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Lack of security of data.
- Needs route details

2.4 PROPOSED SYSTEM

- The aim of proposed system is to develop a system of improved facilities.
- The proposed system can overcome all the limitations of the existing system.
- The system provides proper security and reduces the manual work.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Ensure data accuracy’s.
- Proper control of the higher officials.
- Minimize manual data entry.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

After doing the project Online Food Ordering System, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time. Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

3.1.1 Economical Feasibility

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

- All hardware and software cost has to be borne by the organization.
- Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

3.1.2 Technical Feasibility

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platforms.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	- Intel core i3
RAM	- 4 GB
Hard disk	- 1 TB

3.2.2 Software Specification

Front End- HTML, CSS, Bootstrap

Back End- MYSQL

Client on PC -Windows 7 and above.

Technologies used - JS, HTML, J Query, PHP, CSS, Bootstrap, Ajax

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP Hypertext Preprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command- line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**
  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**
  A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**
  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**
  If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.
- **MySQL Server works in client/server or embedded systems.**
  The MySQL Database Software is a client/server system that consists of a multi- threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi- threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design.

After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service- oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their

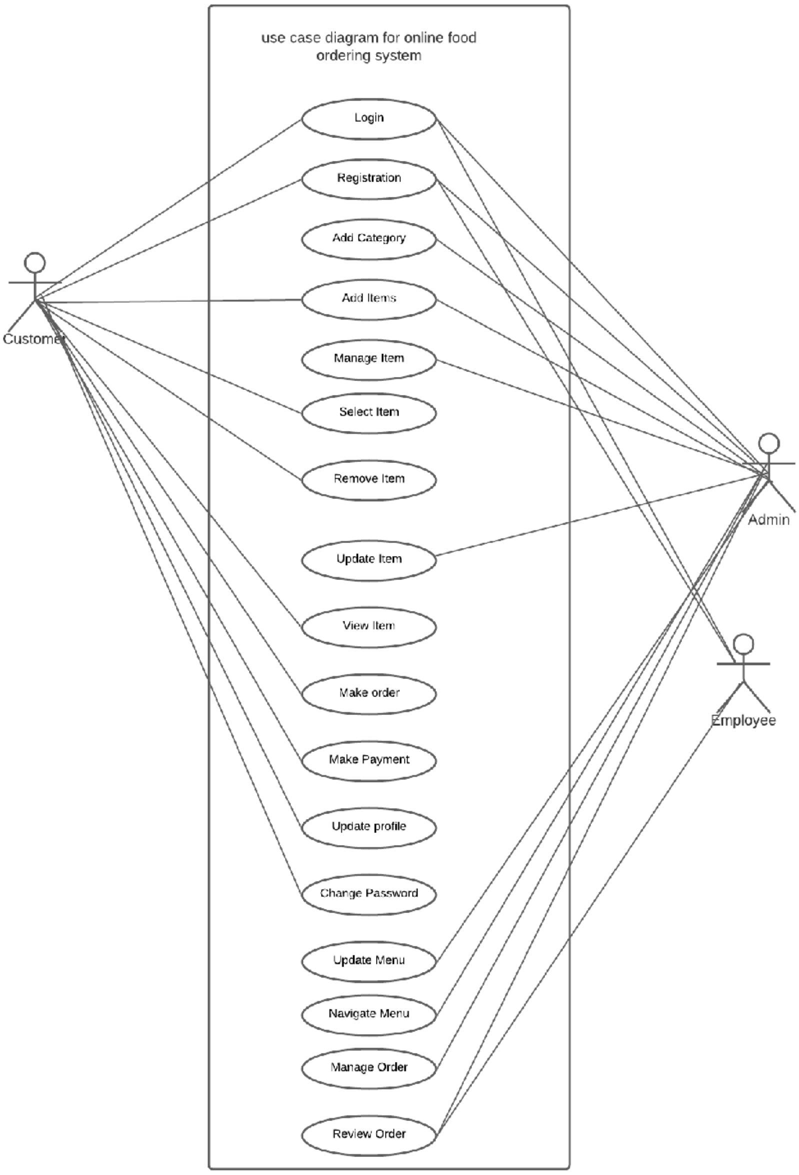
roles.

- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Use Case Diagram of Online Food Ordering System



4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order that is the order in which these interactions take place. We can also use the terms even diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

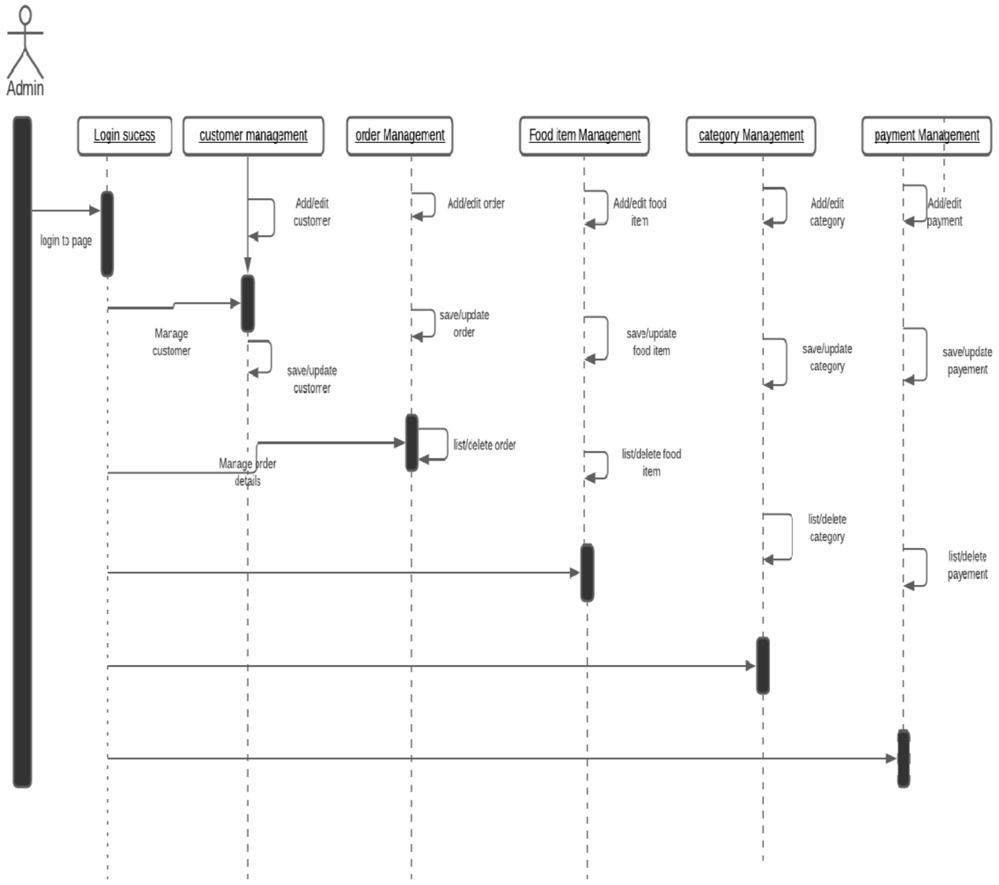
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message

iv. **Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

Sequence diagram for Online food ordering System

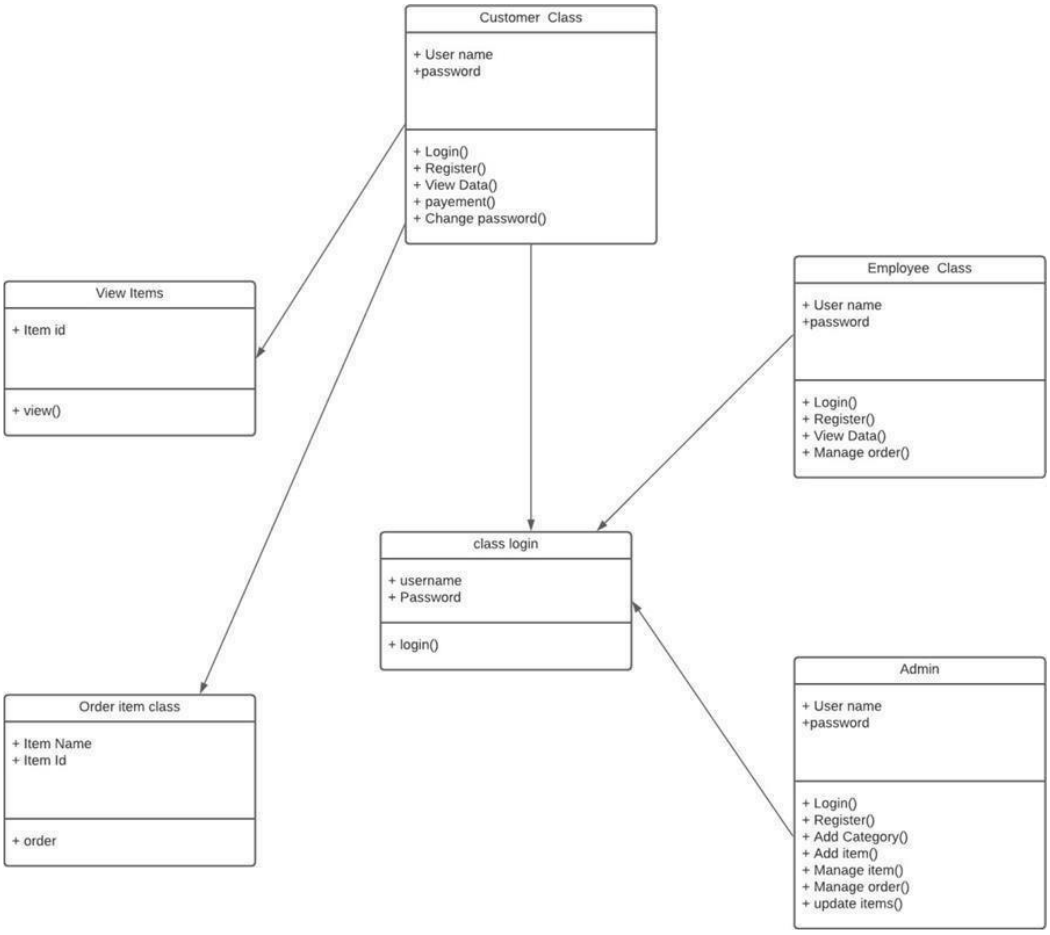


4.2.3 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

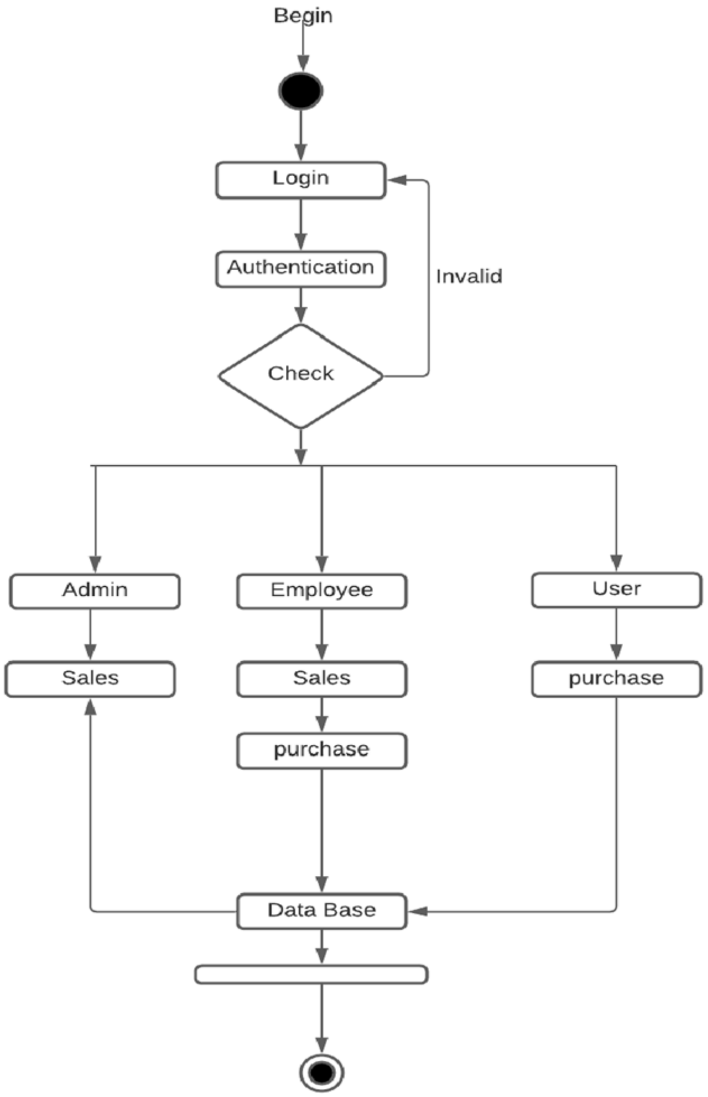
Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.



4.2.4 ACTIVITY DIAGRAM

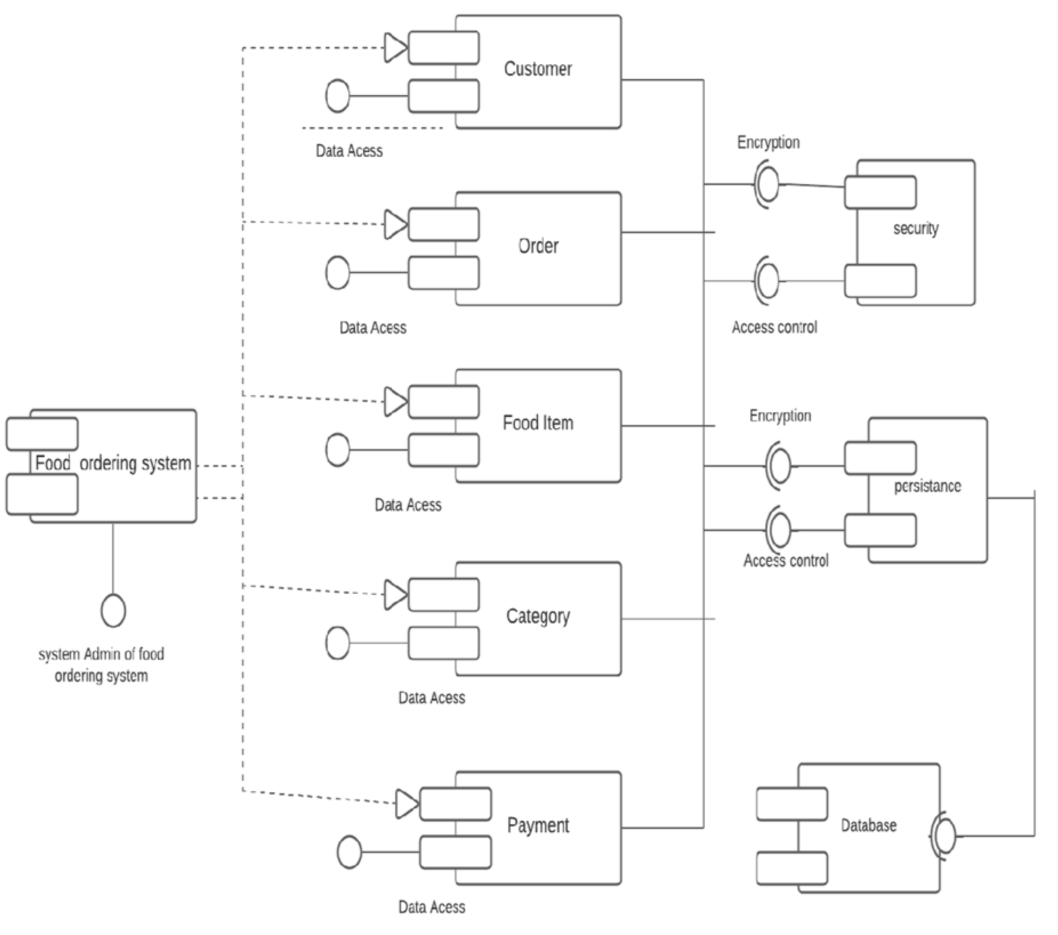
Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.
The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.



4.2.5 COMPONENTDIAGRAM

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executable, files, libraries, etc. that resides within the node.

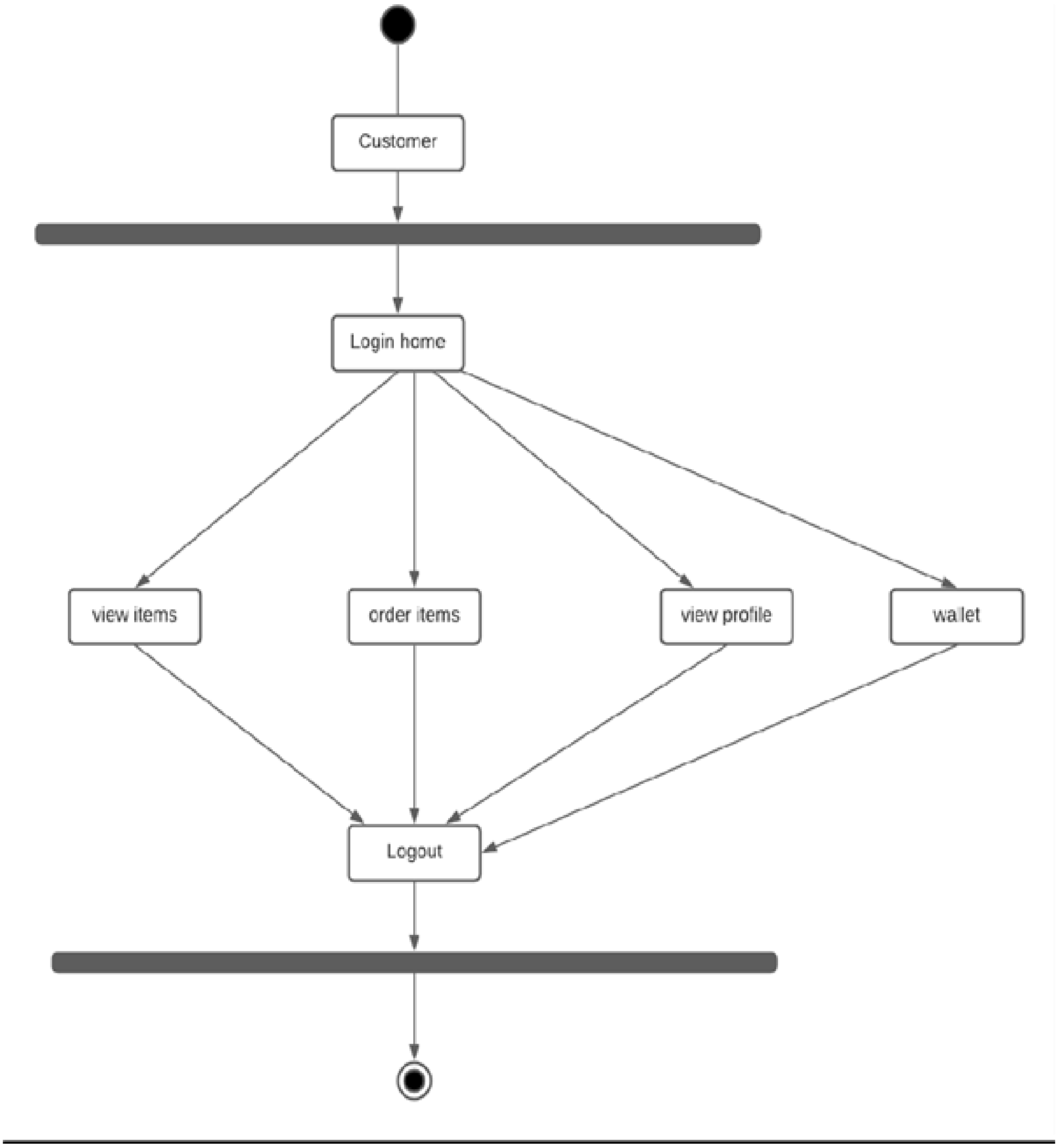
It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.



4.2.6 STATE CHART DIAGRAM (customer)

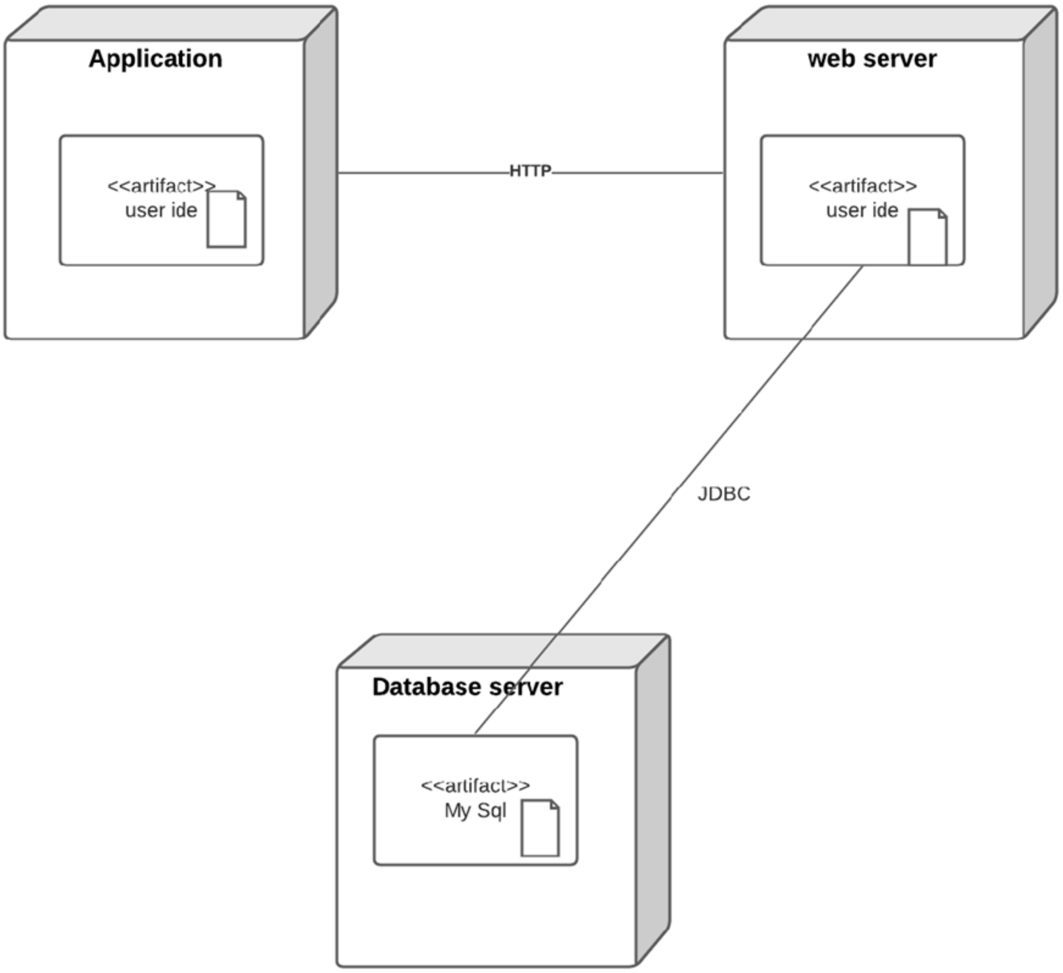
The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. Activity diagram explained in the next chapter, is a special kind of a State chart diagram. As State chart diagram defines the states, it is used to model the lifetime of an object.



4.2.7 DEPLOYMENT DIAGRAM

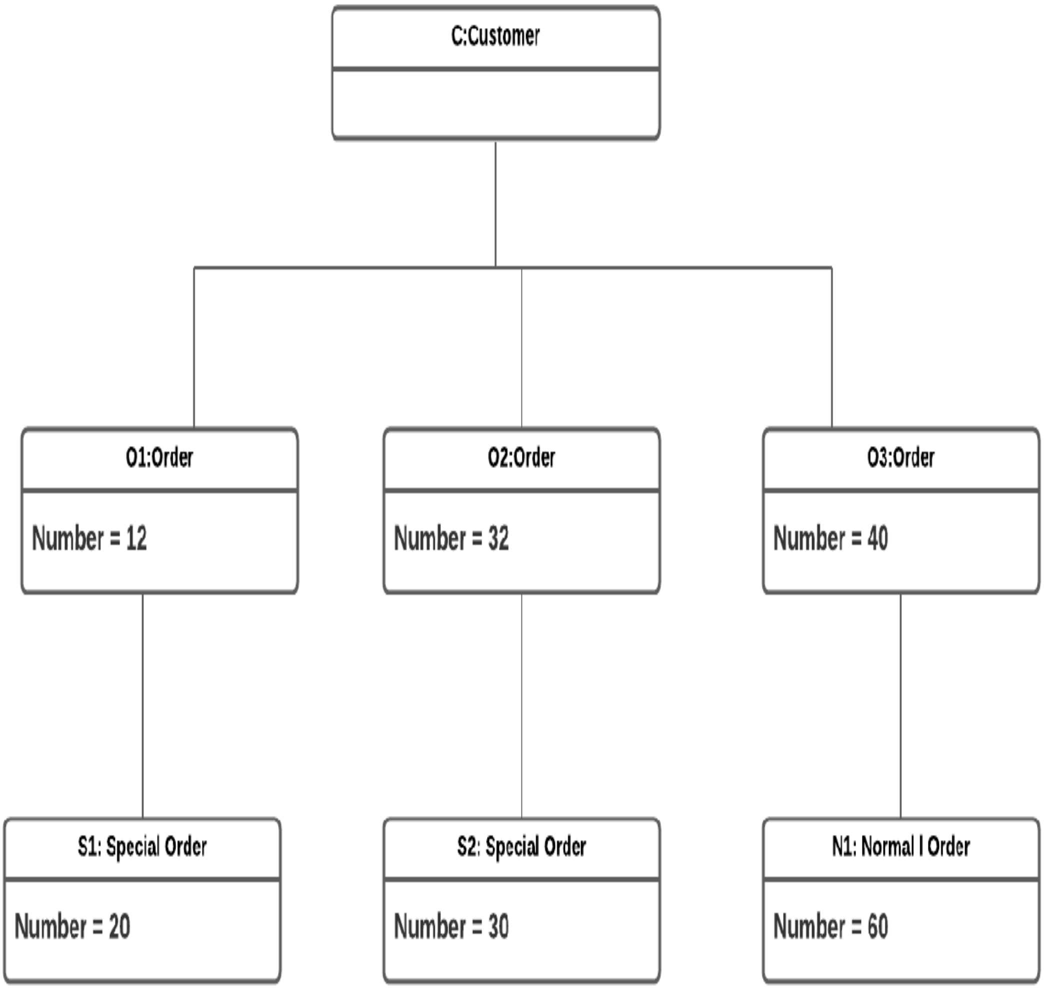
The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.



4.2.8 OBJECT DIAGRAM

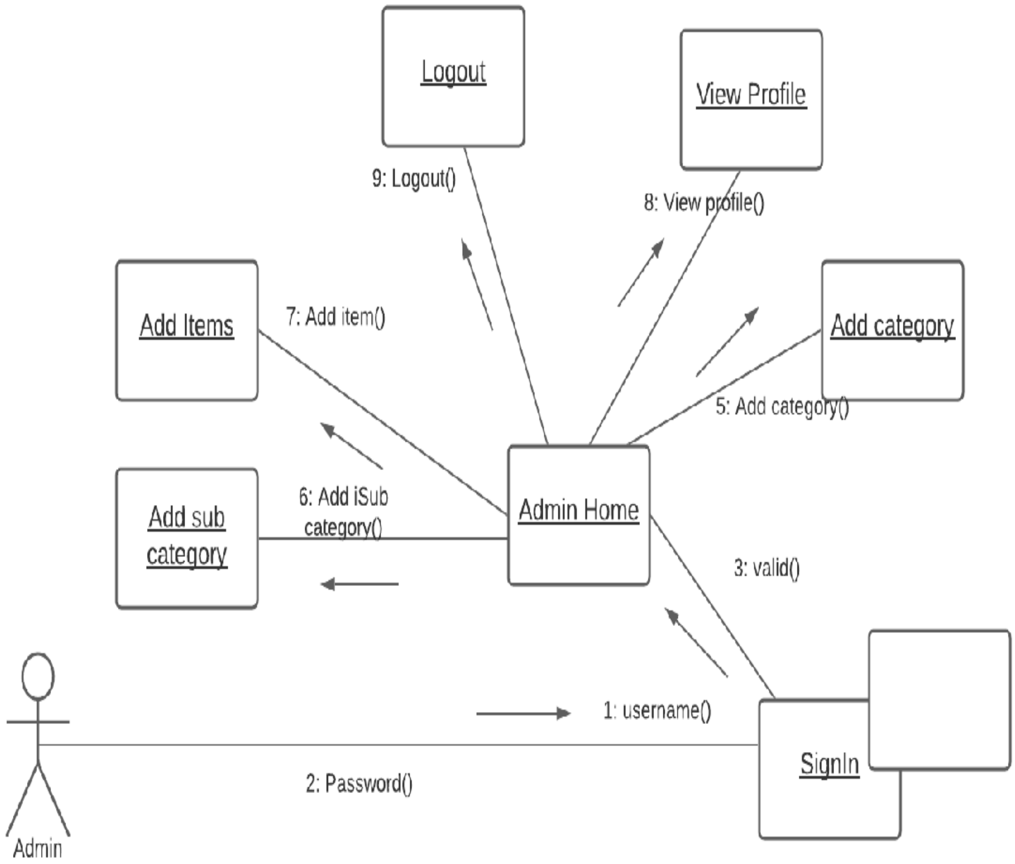
Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.

Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.



4.2.9 COLLABORATION DIAGRAM

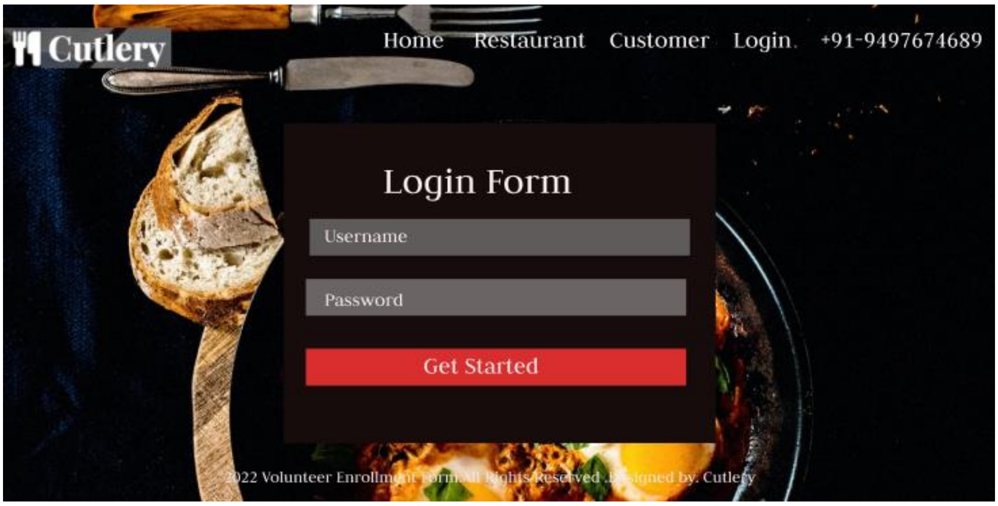
The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.



4.3 USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

Form Name: Login Page



Login Form

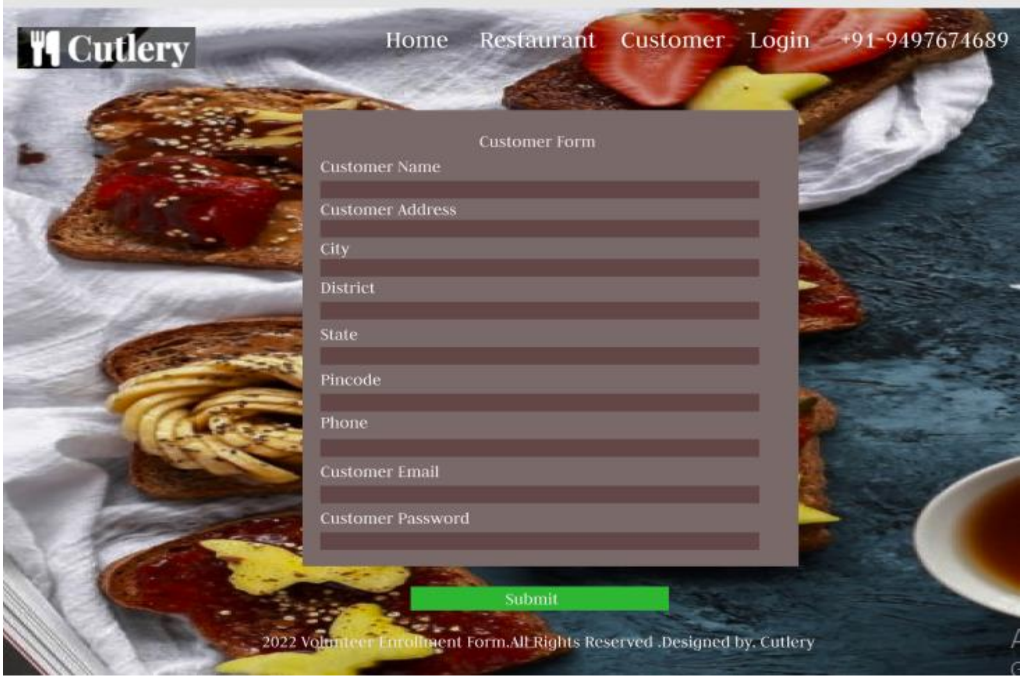
Username

Password

Get Started

2022 Volunteer Enrollment Form.All Rights Reserved .Designed by. Cutlery

Form Name : User Sign up



Customer Form

Customer Name

Customer Address

City

District

State

Pincode

Phone

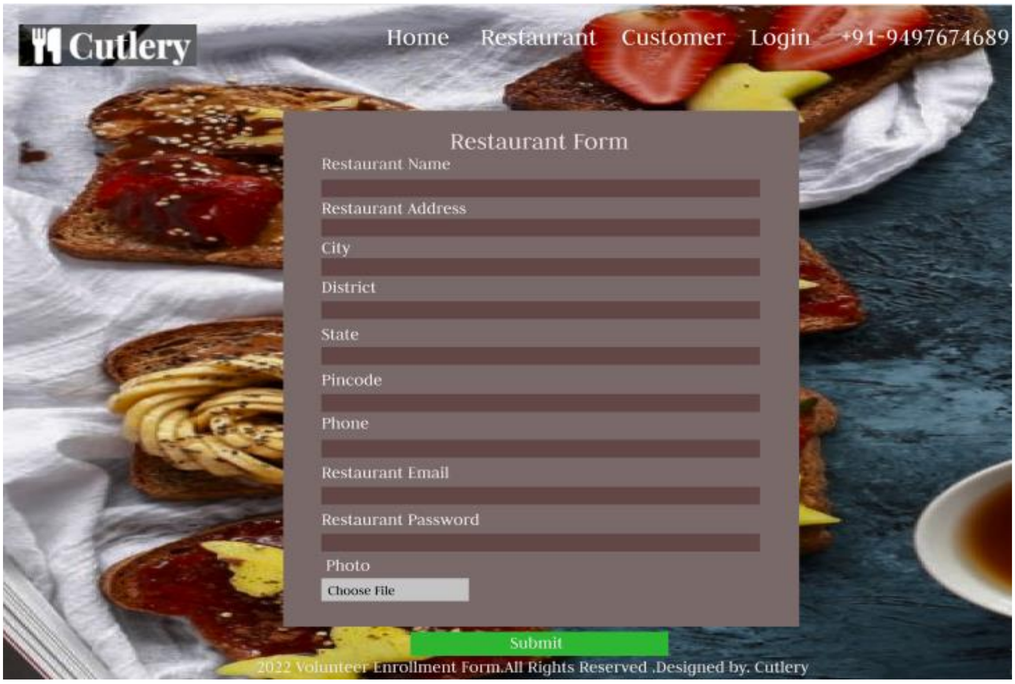
Customer Email

Customer Password

Submit

2022 Volunteer Enrollment Form.All Rights Reserved .Designed by. Cutlery

Form Name : Restaurant Sign up



Restaurant Form

Restaurant Name

Restaurant Address

City

District

State

Pincode

Phone

Restaurant Email

Restaurant Password

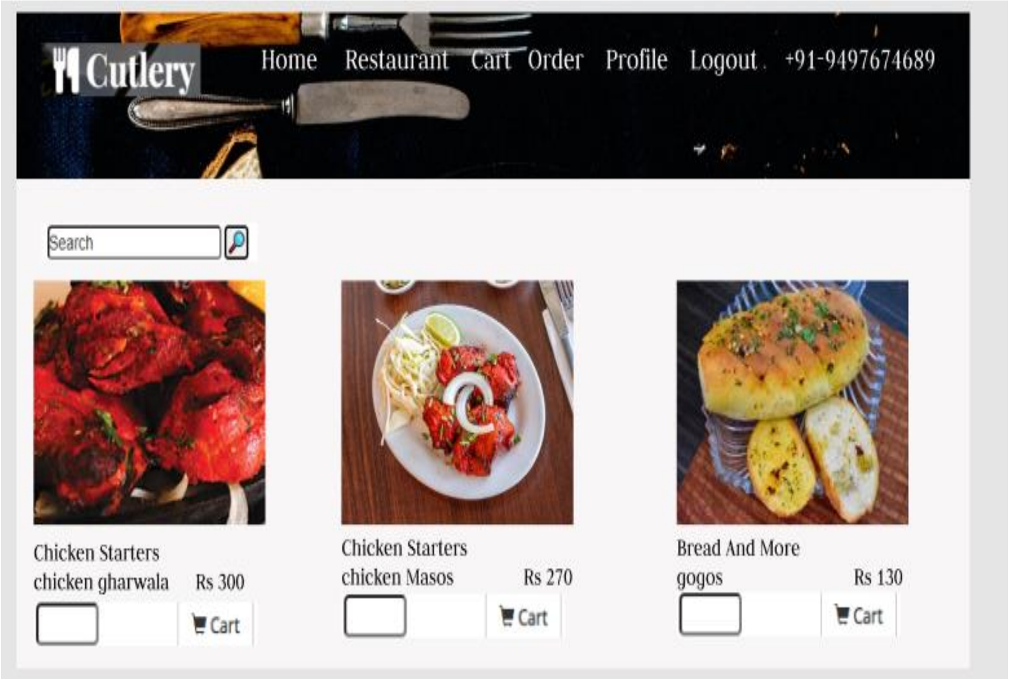
Photo

Choose File

Submit

2022 Volunteer Enrollment Form.All Rights Reserved .Designed by. Cutlery

Form Name: Menu List (Customer)

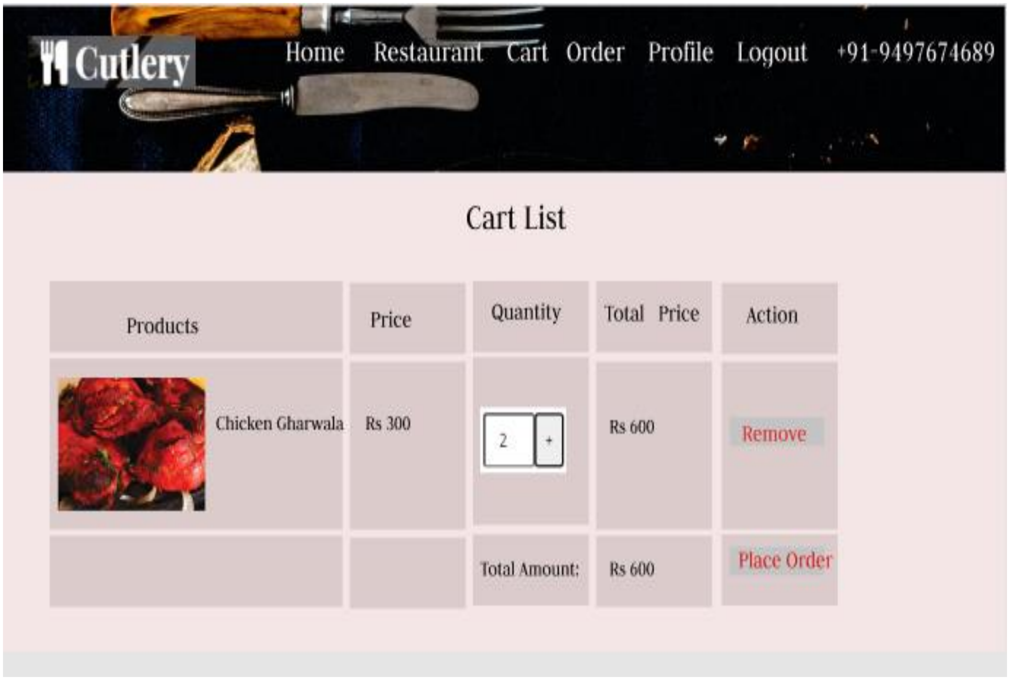


Chicken Starters
chicken gharwala Rs 300

Chicken Starters
chicken Masos Rs 270

Bread And More
gogos Rs 130

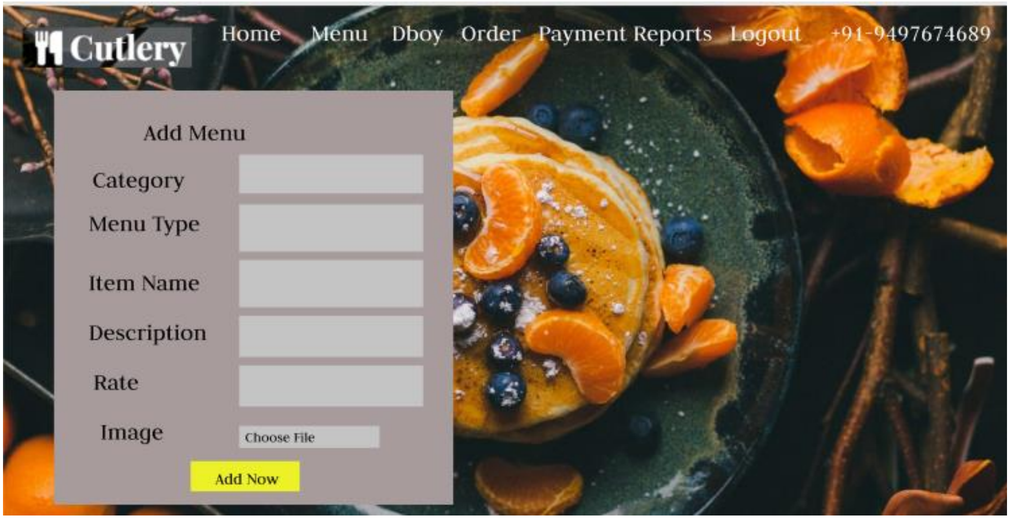
Form Name : Cart List



Cart List

Products	Price	Quantity	Total Price	Action
Chicken Gharwala	Rs 300	2 +	Rs 600	Remove
		Total Amount:	Rs 600	Place Order

Form Name : Add Menu



Add Menu

Category

Menu Type

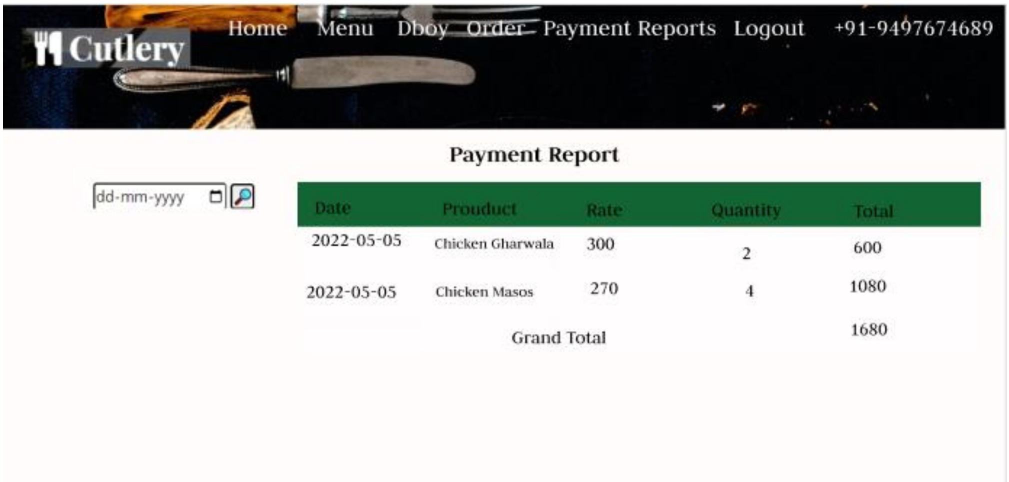
Item Name

Description

Rate

Image Choose File

Form Name : Payment Report



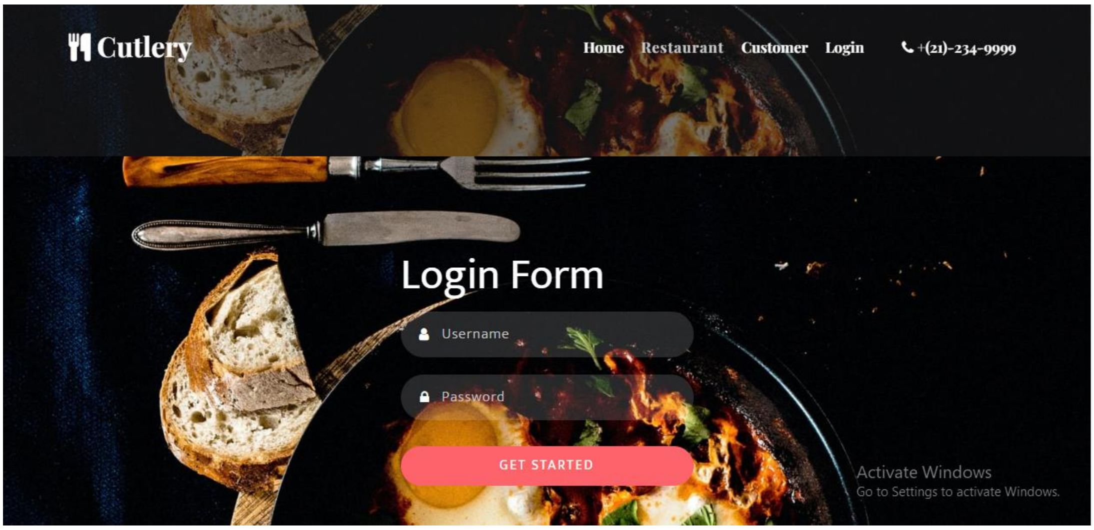
Payment Report

dd-mm-yyyy

Date	Prouduct	Rate	Quantity	Total
2022-05-05	Chicken Gharwala	300	2	600
2022-05-05	Chicken Masos	270	4	1080
	Grand Total			1680

4.3.2 OUTPUT DESIGN

User Login



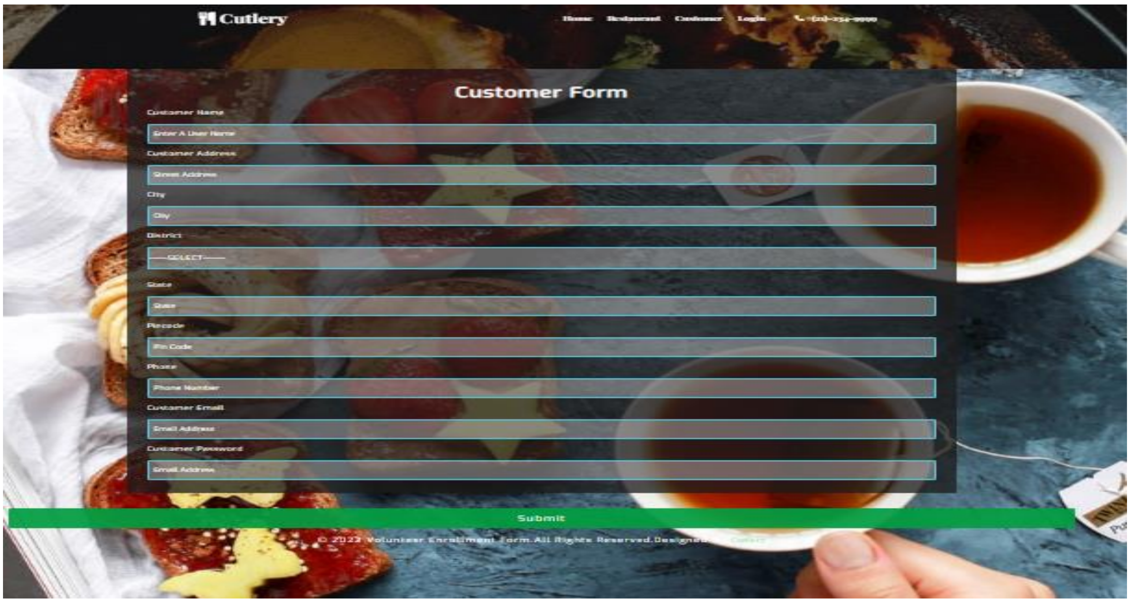
Login Form

Username

Password

GET STARTED

Customer Registration



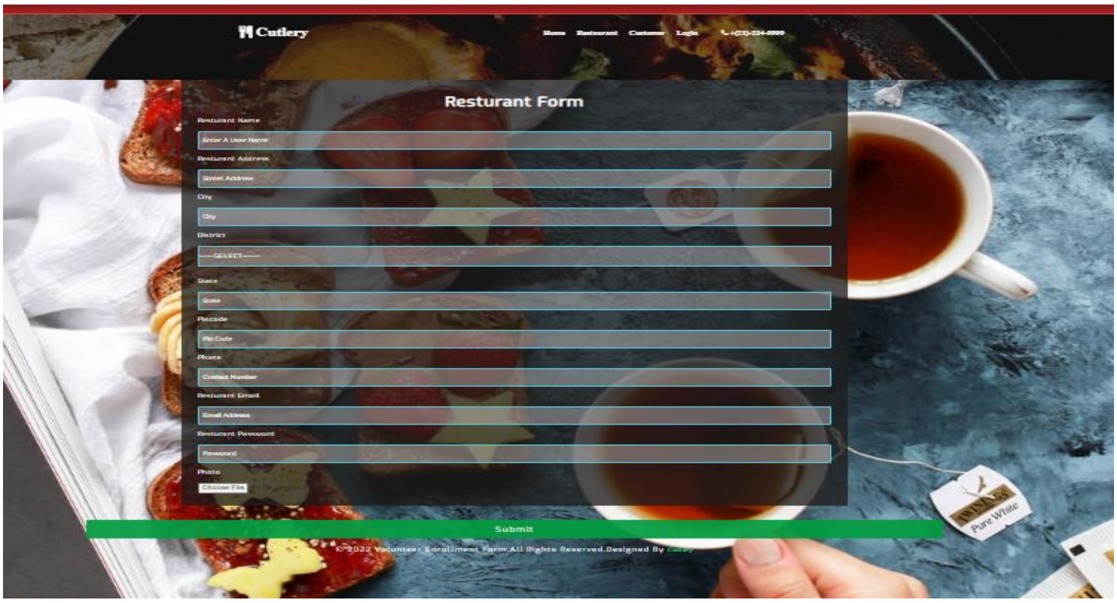
Customer Form

Customer Name
Enter A User Name
Customer Address
Street Address
City
City
District
——SELECT——
State
State
Pincode
Pin Code
Phone
Phone Number
Customer Email
Email Address
Customer Password
Email Address

Submit

© 2022 Volunteer Enrollment Form.All Rights Reserved.Designed

Restaurant Registration



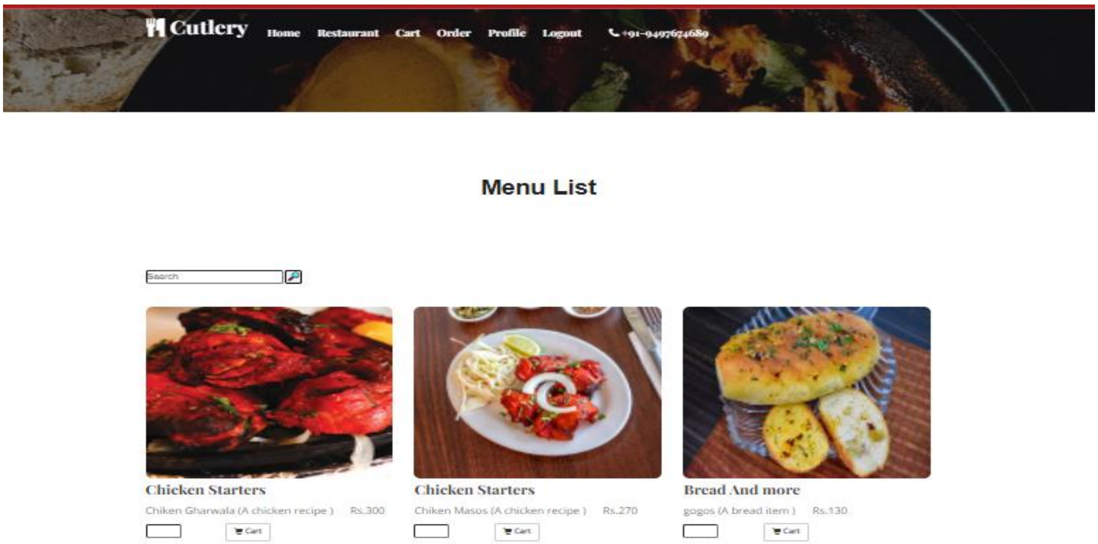
Resturant Form

Resturant Name
Enter A User Name
Resturant Address
Street Address
City
City
District
——SELECT——
State
State
Pincode
Pin Code
Phone
Contact Number
Resturant Email
Email Address
Resturant Password
Password
Photo
Choose File

Submit

© 2022 Volunteer Enrollment Form.All Rights Reserved.Designed By

Menu List(customer)



Menu List

Chicken Starters

Chiken Gharwala (A chicken recipe) Rs.300

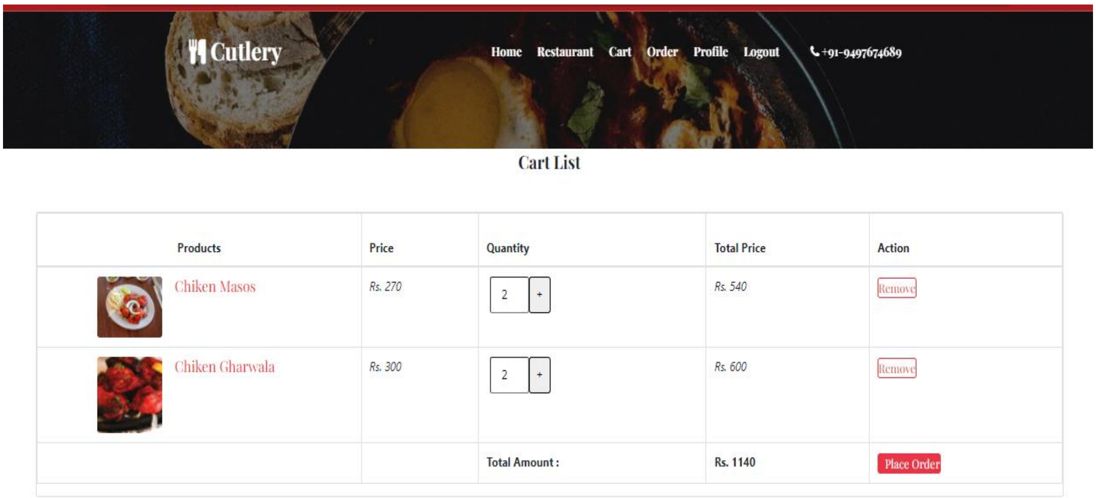
Chicken Starters

Chiken Masos (A chicken recipe) Rs.270

Bread And more

gogos (A bread item) Rs.130

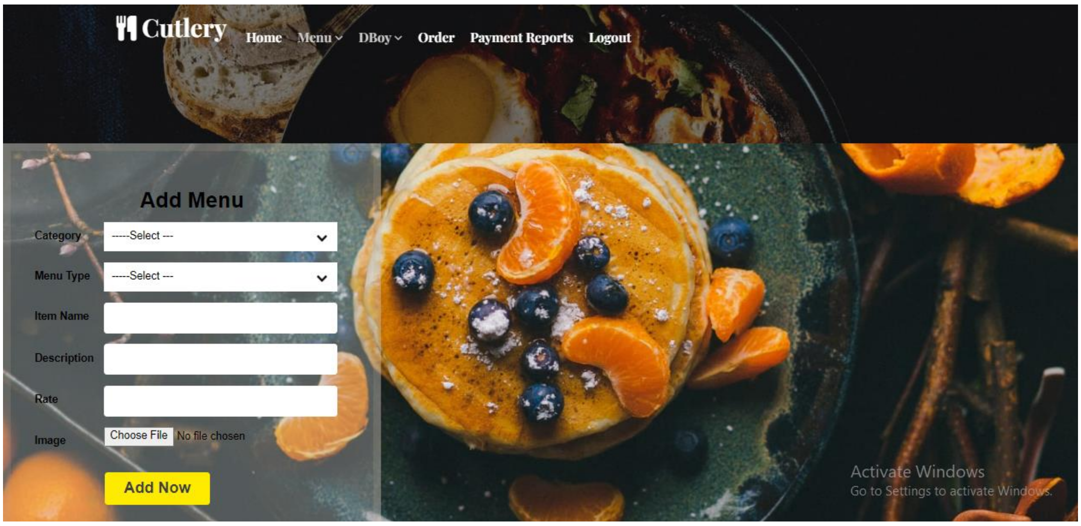
Cart List(customer)



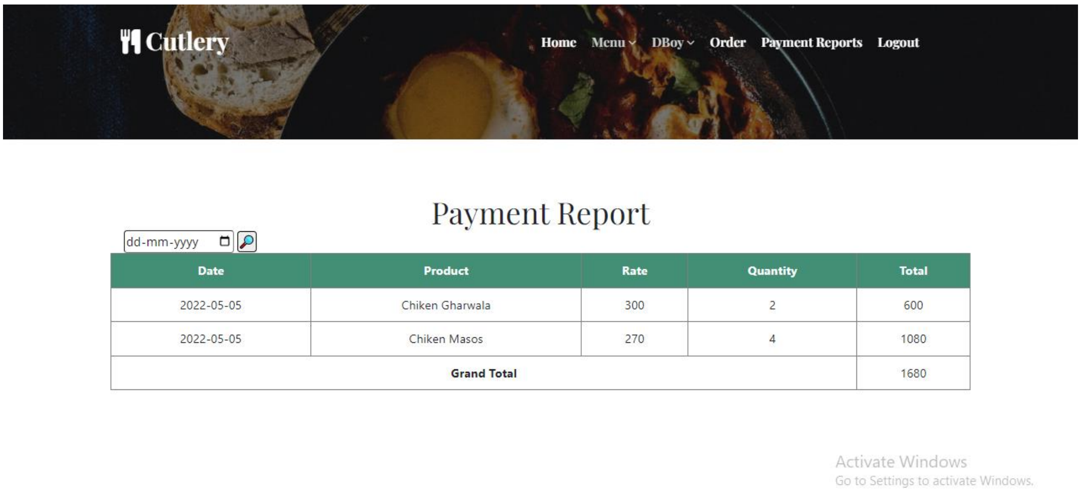
Cart List

Products	Price	Quantity	Total Price	Action
Chiken Masos	Rs. 270	2 +	Rs. 540	Remove
Chiken Gharwala	Rs. 300	2 +	Rs. 600	Remove
		Total Amount :	Rs. 1140	Place Order

Add Menu(Restaurants)



Payment Report(Restaurants)



Payment Report

Date	Product	Rate	Quantity	Total
2022-05-05	Chiken Gharwala	300	2	600
2022-05-05	Chiken Masos	270	4	1080
Grand Total				1680

4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

4.4.1 Data Integrity
4.4.2 Data independence

4.4.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Keycan have null values.
- Referential Integrity enforces that no Primary Key canhave null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.4.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modeling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalizedup to the thirdnormal form.

As the name implies, it denotes putting things in the normal form. The app lication developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows "relations within relations" or "relations as attribute values within tuples". The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

5. TABLE DESIGN

- **Table No:01**

  - ✓ **Table name: tblassigndboy**
  - ✓ **Primary Key: allocid**
  - ✓ **Table Description : To assign delivery boy**

Name	Type	Size	Constraints
Allocid	Int	11	Primary Key
Cmid	Int	11	Not Null
Dboyid	Int	11	Not Null
Status	Varchar	20	Not Null

- **Table No: 02**

  - ✓ **Table Name : tblcart**
  - ✓ **Primary Key: cartid**
  - ✓ **Table Description: To store cart details**

Name	Type	Size	Constraints
Cartid	int	11	Primary Key
Cmid	int	11	Not Null
cid	int	11	Not Null
Mid	int	11	Not Null
qty	int	11	Not Null
totalamt	int	11	Not Null
Cstatus	int	11	Not Null

- **Table No: 03**
  - ✓ **Table Name : tblcartmaster**
  - ✓ **Primary Key: cmid**
  - ✓ **Table Description: To store cart details**

Name	Type	Size	Constraints
Cmid	Int	11	Primary Key
cid	Int	11	Not Null
Rid	Int	11	Not Null
Totalprice	Int	11	Not Null
Status	varchar	20	Not Null

Table No: 04

- ✓ **Table Name : tblcustomer**
- ✓ **Primary Key: cid**
- ✓ **Table description: To store customer details**

Name	Type	Size	Constraints
cid	Int	11	Primary key
Cname	Varchar	50	Not Null
Caddress	Varchar	40	Not Null
Ccity	Varchar	40	Not Null
Cdistrict	Varchar	30	Not Null
Cstate	Varchar	30	Not Null
Cpincode	int	11	Not Null
Cphone	bigint	11	Not Null
Cemail	Varchar	40	Not Null
Status	int	11	Not Null

- **Table No: 05**
  - ✓ **Table Name : tbldboy**
  - ✓ **Primary Key: id**
  - ✓ **Table Description: To store delivery boy details**

Name	Type	Size	Constraints
Id	Int	11	Primary key
Rid	Int	11	Not Null
Name	Varchar	50	Not Null
Address	Varchar	50	Not Null
City	Varchar	50	Not Null
District	Varchar	50	Not Null
Gender	Varchar	40	Not Null
Dob	Int	11	Not Null
Phone	Bigint	50	Not Null
Email	Varchar	50	Not Null
Status	Int	30	Not Null

Table No:06

- ✓ **Table Name : tbllogin**
- ✓ **Table Description: To store login details**

Name	Type	Size	Constraints
username	Varchar	30	Not Null
password	Varchar	30	Not Null
usertype	Varchar	30	Not Null
Status	Varchar	10	Not Null

- **Table No:07**
  - ✓ **Table Name : tblmenu**
  - ✓ **Primary Key: mid**
  - ✓ **Table Description: To store menu details**

Name	Type	Size	Constraints
mid	Int	11	Primary key
rid	Int	11	Not Null
mcat	Varchar	50	Not Null
mtype	Varchar	50	Not Null
mname	Varchar	40	Not Null
mdes	Varchar	50	Not Null
mrate	Int	11	Not Null
mimage	Varchar	490	Not Null
status	Int	11	Not Null

- **Table No:08**
  - ✓ **Table Name : tblpayment**
  - ✓ **Primary Key: pid**
  - ✓ **Table Description: To store payment details**

Name	Type	Size	Constraints
Pid	int	11	Primary key
Cmid	int	11	Not Null
Cardname	Varchar	40	Not Null
cardnumber	Bigint	20	Not Null
Expdate	Date	0	Not Null
Amount	int	11	Not Null
Paydate	Date	0	Not Null

- **Table No:09**
  - ✓ **Table Name : tblrestaurant**
  - ✓ **Primary Key: pid**
  - ✓ **Table Description: To store restaurant details**

Name	Type	Size	Constraints
Rid	int	11	Primary key
Mame	Varchar	30	Not Null
raddress	Varchar	50	Not Null
Rcity	Varchar	30	Not Null
rdistrict	Varchar	30	Not Null
Rstate	Varchar	30	Not Null
rpincode	int	20	Not Null
Rphone	bigint	11	Not Null
Remail	Varchar	40	Not Null
Rphoto	Varchar	300	Not Null
Status	int	11	Not Null

CHAPTER 5

SYSTEM TESTING

6.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

6.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels oftesting include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

6.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during al steps in an algorithm’s execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

6.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

6.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that willfully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

6.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

6.2.4.1 Input Screen Designs,

6.2.4.2 Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

6.2.5 Test Case

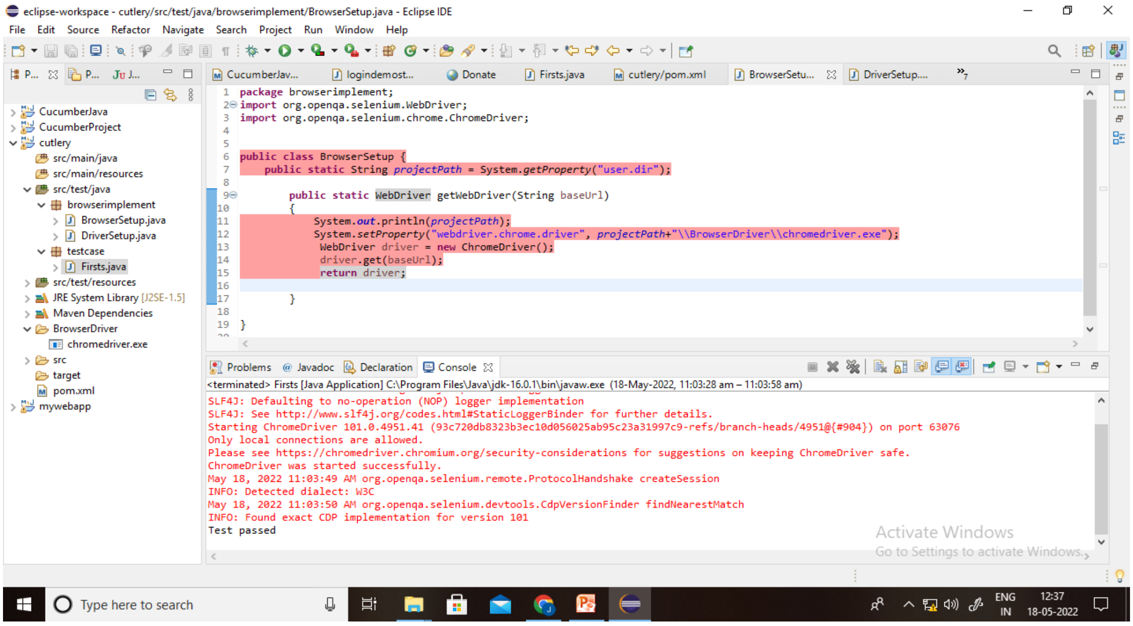
Test Case 1

Project Name: Online Food Ordering System	
Login Test Case	
Test Case ID: Fun_1	**Test Designed By:Jayalekshmi Jayakumar**
Test Priority(Low/Medium/High): High	**Test Designed Date:** 18-05-2022
Module Name: Login Screen	**Test Executed By : Sr. Elsin Chakkalackal. S.H**
Test Title : Verify login with validemail and password	**Test Execution Date:** 18-05-2022
Description: Test the Login Page	
Pre-Condition : User has valid email id and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation toLogin Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid Email Id	User Name: jk123@gmail.com	User should beable to Login	User Logged inand navigated to Subadmin Dashboard with records	Pass
3	Provide Valid Password	Password: jk@1234			
4	Click on Sign In button				
5	Provide Invalid Email Id or password	Email Id: user@gmail.Com Password: User1234	User shouldnot be able to Login	Message for enter valid email id or password displayed	Pass
6	Provide Null Email Id or Password	Email Id: null Password: null			
7	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account.The Account session details are logged in database					

Code

```
package testcase;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import browserimplement.DriverSetup;
public class Firsts {


public static WebDriver driver;

public static void main(String[] args) {
// TODO Auto-generated method stub

driver =  DriverSetup.getWebDriver("http://localhost/Restaurant/login.php");

//login-Invalid case
driver.findElement(By.name("username")).sendKeys("cinu@gmail.com");
driver.findElement(By.name("password")).sendKeys("cinu@123");
driver.findElement(By.name("submit")).click();

String actualUrl="http://localhost/Restaurant/custhome.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed"); } else { System.out.println("Test failed"); }
driver.quit();
}

}
```


eclipse-workspace - cutlery/src/test/java/browserimplement/BrowserSetup.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
CucumberJav... logindemost... Donate Firsts.java cutlery/pom.xml BrowserSetu... DriverSetup....
CucumberJava
CucumberProject
cutlery
src/main/java
src/main/resources
src/test/java
browserimplement
BrowserSetup.java
DriverSetup.java
testcase
Firsts.java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
BrowserDriver
chromedriver.exe
src
target
pom.xml
mywebapp
Problems Javadoc Declaration Console
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG IN 12:37 18-05-2022

```
package browserimplement;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;


public class BrowserSetup {
    public static String projectPath = System.getProperty("user.dir");

        public static WebDriver getWebDriver(String baseUrl)
        {
            System.out.println(projectPath);
            System.setProperty("webdriver.chrome.driver", projectPath+"\\BrowserDriver\\chromedriver.exe");
             WebDriver driver = new ChromeDriver();
             driver.get(baseUrl);
             return driver;

        }

}
```

```
<terminated> Firsts [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (18-May-2022, 11:03:28 am – 11:03:58 am)
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 63076
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 18, 2022 11:03:49 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 18, 2022 11:03:50 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

CHAPTER 7

IMPLEMENTATION

7.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

7.2IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system.

In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must beware of the benefits of using the new system.Their
- confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

7.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

7.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application.

This training may be different across different user groups and across different levels of hierarchy

7.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

8.1.1 FUTURE SCOPE

- Customers can able to do advanced search options
- Data security can be enhanced.

CHAPTER 9

BIBLIOGRAPHY

REFERENCES:

- Roger S Pressman, “*Software Engineering*”, 1994.
- PankajJalote, “So*ftware engineering*: a precise approach”, 2006.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html
- https://www.sqlite.org/index.html
- https://www.w3schools.com/html/

CHAPTER 10

APPENDIX

10.1 Sample Code

Login.php

```
<?php
session_start();
include 'commonbase.php'

?>
<!DOCTYPE html>
<html lang="zxx">


<head>

<meta name="viewport" content="width=device-width, initial-scale=1">
<meta charset="utf-8">
<meta name="keywords" content="Key Login Form a Responsive Web Template, Bootstrap
Web Templates, Flat Web Templates, Android Compatible Web Template, Smartphone
Compatible Web Template, Free Webdesigns for Nokia, Samsung, LG, Sony Ericsson,
Motorola Web Design">
<script>
addEventListener("load", function() {
setTimeout(hideURLbar, 0);
}, false);

function hideURLbar() {
window.scrollTo(0, 1);
}
</script>

<link rel="stylesheet" href="login/css/style.css" type="text/css" media="all">

<link rel="stylesheet" href="login/css/font-awesome.min.css" type="text/css" media="all">

<link href="//fonts.googleapis.com/css?family=Quattrocento+Sans:400,400i,700,700i"
rel="stylesheet">
<link href="//fonts.googleapis.com/css?family=Mukta:200,300,400,500,600,700,800"
rel="stylesheet">

</head>
<body>
<section class="main">
<div class="layer">
<div class="content-w3ls">
<div class="content-bottom">
<h1 style="color:white;font-size:50px">Login Form</h1>
<form action="#" method="post">
```

```
<div class="field-group">
<span class="fa fa-user" aria-hidden="true"></span>
<div class="wthree-field">
<input name="username" id="text1" type="text" value="" placeholder="Username" required>
</div>
</div>
<div class="field-group">
<span class="fa fa-lock" aria-hidden="true"></span>
<div class="wthree-field">
<input name="password" id="myInput" type="Password" placeholder="Password">
</div>
</div>
<div class="wthree-field">
<button type="submit" name="submit" class="btn">Get Started</button>
</div>
<ul class="list-login">

<li class="clearfix"></li>
</ul>
<ul class="list-login-bottom">
<li class="">
<a href="#" class="">Create Account</a>
</li>

<li class="clearfix"></li>
</ul>
</form>
</div>
</div>
<div class="bottom-grid1">
<div class="links">
<ul class="links-unordered-list">
<li class="">
<a href="#" class="">About Us</a>
</li>
<li class="">
<a href="#" class="">Privacy Policy</a>
</li>
<li class="">
<a href="#" class="">Terms of Use</a>
</li>
</ul>
</div>
<div class="copyright">
<p>© 2022 Key. All rights reserved | Design by
<a href="http://w3layouts.com">Cutlery</a>
</p>
</div>
</div>
</div>
</section>
```

```
</body>
</html>
<?php
include 'connection.php';
if (isset($_REQUEST['submit'])) {
$email = $_REQUEST['username'];
$pwd = $_REQUEST['password'];

$q = "select count(*) from tbllogin where username='$email'";
$s = mysqli_query($conn, $q);
$r = mysqli_fetch_array($s);
if ($r[0] == 0)    //to check whether the username exist
{
echo '<script>alert("Username doesnt exist")</script>';
} else {
$_SESSION['email'] = $email;    //creating a session variable
$q = "select * from tbllogin where username='$email'";
$s = mysqli_query($conn, $q);
$r = mysqli_fetch_array($s);
if ($r['password'] == $pwd)  //to check the password entered by user with the password in
database
{
if ($r['status'] == "1")  //to check the status of user
{
if ($r['usertype'] == "admin")  //to check the usertye/role of the user
{
echo '<script>location.href="adminhome.php"</script>';
} else if ($r['usertype'] == "restaurant") {
$s = "select * from tblrestaurant where remail='$email'";
$ss = mysqli_query($conn, $s);
$data = mysqli_fetch_array($ss);
$_SESSION['rid'] = $data[0];
echo '<script>location.href="resthome.php"</script>';
} else if ($r['usertype'] == "dboy") {
$s = "select * from tbldboy where email='$email'";
$ss = mysqli_query($conn, $s);
$data = mysqli_fetch_array($ss);
$_SESSION['did'] = $data[0];
echo '<script>location.href="dboyhome.php"</script>';
} else if ($r['usertype'] == "customer") {
$s = "select * from tblcustomer where cemail='$email'";
$ss = mysqli_query($conn, $s);
$data = mysqli_fetch_array($ss);
$_SESSION['cid'] = $data[0];
echo '<script>location.href="custhome.php"</script>';
}
} else {
echo '<script>alert("Your account is not valid")</script>';
}
} else {
echo '<script>alert("Incorrect password")</script>';
}
}
}
?>
</html>
```

Custcart.php

```
<?php
session_start();
if (!isset($_SESSION['email'])) {
header('location:index.php');
}
include 'connection.php';
include 'customerbase.php';

?>


<h4 class=" font-weight-bold mt-1 " style="text-align: center; ">Cart List</h4>
<div class="card" style="margin: 50px;">


<div class="table-responsive">

<table class="table table-stripped table-bordered">
<thead>
<tr style="height:70px; vertical-align: middle;">
<th style="width: 500px; text-align:center" scope="col">Products</th>
<th scope="col">Price</th>
<th scope="col">Quantity</th>
<th scope="col">Total Price</th>
<th scope="col">Action</th>
</tr>
</thead>
<?php
if (!empty($_SESSION['cid'])) {
$cid = $_SESSION['cid'];
// echo $custid;
```

```
$c = 0;

$sql_fetch_cm = "SELECT tblmenu.*,tblcart.* FROM tblmenu,tblcart WHERE
tblcart.cid='$cid' and tblmenu.mid=tblcart.mid and tblcart.cstatus='pending'";
// echo $sql_fetch_cm;
$sql_exe_cm = mysqli_query($conn, $sql_fetch_cm);
$i = 0;
while ($cm = mysqli_fetch_array($sql_exe_cm)) {
++$i;

// $mid=$cm['mid'];

?>



<tr>
<td>
<a href="ItemDetails.php?user_id=<?php echo $cm['mid']; ?>" style="text-decoration:
none;">
<figure class=" d-flex ">
<div class="aside"><img style="margin-left: 80px ;" width="100" height="100" class=" img-
fluid rounded" src=" <?php echo  $cm['mimage']; ?>" class="img-sm"></div>
<figcaption class="info" style="width: 250px; margin-left: 20px;">
<h5 class="title "><?php print_r($cm['mname']); ?></h5>
</figcaption>
</figure>
</a>
</td>
<td>
<div class="price-wrap">
<var class="price">Rs. <?php print_r($cm['mrate']); ?></var>
</div>
</td>
<td>
<form style="width: 150px;" class="row needs-validation" method="post"
action="cart_update.php">
<div class="input-group has-validation">
<input name="qty" id="qty<?php echo $i; ?>" type="number" min="1" max="10"
style="border-radius: 0.2rem;width:60px;text-align:center;margin-left:20px;" value="<?php
print_r($cm['qty']); ?>" placeholder="Qty" required>
<input type="button" value="+" onclick="myFunc(<?php echo $i ?>)" style="padding:
10px;">
<input name="item" id="item<?php echo $i; ?>" type="hidden" value="<?php
print_r($cm['mid']); ?>">

<input name="sp" id="rate<?php echo $i; ?>" type="hidden" value="<?php
print_r($cm['mrate']); ?>">
```

```
<input name="tp" id="total<?php echo $i; ?>" type="hidden" value="<?php
print_r($cm['totalamt']); ?>">
<input name="cartid" id="cartid<?php echo $i; ?>" type="hidden" value="<?php
print_r($cm['cartid']); ?>">
<!-- <button type="submit"  class="btn "><i class="bi bi-arrow-right">-></i></button>
-->

</div>
</form>

</td>
<td>

<div class="price-wrap">
<var class="price">Rs. <?php print_r($cm['totalamt']); ?></var>
</div>
</td>
<td>
<a href="cartremove.php?cid=<?php echo $cm['cartid']; ?>" class="btn btn-outline-
danger">Remove</a>
</td>
</tr>
<?php
}

?>

<tr>
<th></th>
<th></th>
<?php
$r = "Select sum(totalamt) from tblcart where cid='$cid' and cstatus='pending'";
$t = mysqli_query($conn, $r);
$tp = mysqli_fetch_array($t);
$ff = $tp[0];
?>
<th scope="col">Total Amount : </th>
<th scope="col">Rs. <?php echo $ff; ?></th>
<?php
if ($c > 0) {
?>
<th><a href="PlaceOrder.php?cid=<?php echo $cm['cartid']; ?>" class="btn btn-primary
disabled"><i class="bi bi-journal-check" style="margin-right: 10px;"></i>Place
Order</a></th>
<?php
} else {
?>
```

```
th><a href="payment.php?amt=<?php echo $ff; ?>" class="btn btn-primary"><i
class="bi bi-journal-check" style="margin-right: 10px;"></i>Place Order</a></th>
<?php
}
}
?>
</tr>

</tbody>
</table>

</div>
</div>
</aside>
<script>
function myFunc(id) {
qty = document.getElementById("qty" + id).value;
rate = document.getElementById("rate" + id).value;
// id = document.getElementById("itemid" + id).value;
cartid = document.getElementById("cartid" + id).value;
window.location = "updatecart.php?cart=" + cartid + "&qty=" + qty + "&rate=" + rate
}
</script>
</div>
</div>
</div>
</div>
<!-- Footer -->
<footer class="page-footer font-small pt-4" style="height: 160px; background-color:
black; color: white;">
<div class="container-fluid d-flex justify-content-between">
<div class="footer-copyright text-center py-1" id="copyright">
</div>
</footer>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVX
M" crossorigin="anonymous"></script>
<script type="text/javascript" src="jquery.js"></script>
<script>
$(".alert").delay(2000).slideUp(200, function() {
$(this).alert('close');
});
</script>
</body>

</html>
```

Bill.php

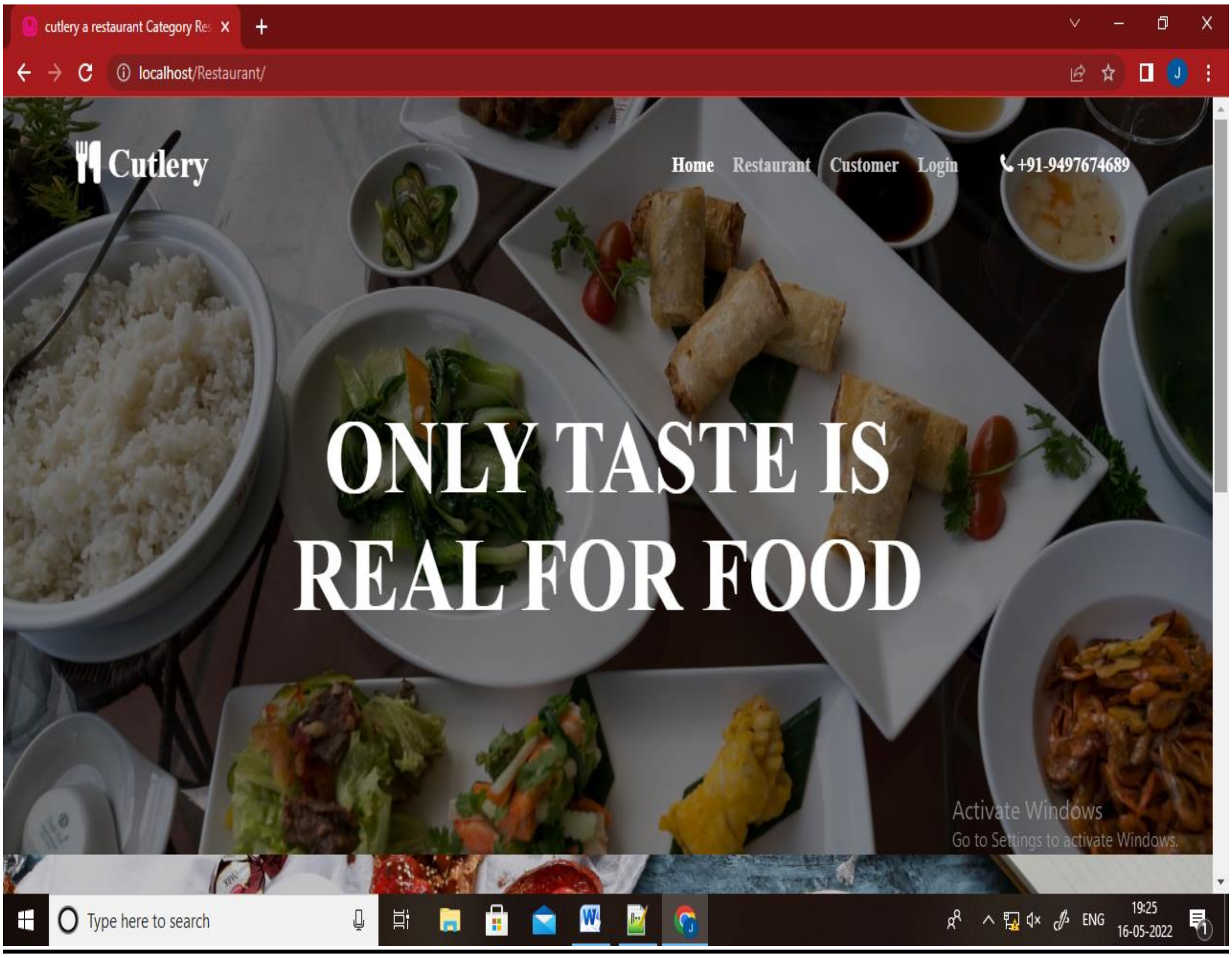
```
<?php
session_start();
if (!isset($_SESSION['email'])) {
header('location:index.php');
}
include 'connection.php';
$cmid = $_GET['cmid'];
$qry = "SELECT * FROM `tblcartmaster`cm, `tblcart`c, `tblmenu`m WHERE
cm.`cmid`='$cmid' AND c.`cmid`='$cmid' AND c.`mid`=m.`mid`";
$res = mysqli_query($conn, $qry);
$cid = $_SESSION['cid'];
$qryCust = "SELECT * FROM `tblcustomer` WHERE `cid`='$cid'";
$resCust = mysqli_query($conn, $qryCust);
$rowCust = mysqli_fetch_array($resCust);
?>

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<link rel="stylesheet" href="BillCss.css">
</head>
<body>
<table class="body-wrap">
<tbody>
<tr>
<td></td>
<td class="container" width="600">
<div class="content">
<table class="main" width="100%" cellpadding="0" cellspacing="0">
<tbody>
<tr>
<td class="content-wrap aligncenter">
<table width="100%" cellpadding="0" cellspacing="0">
<tbody>
<tr>
<td class="content-block">
<h2 style="text-align:center">Invoice</h2>
</td>
</tr>
<tr>
```

```
<tr>
<td colspan="2">
<table class="invoice-items" cellpadding="0" cellspacing="0">
<tbody>
<?php
$total = 0;
while ($row = mysqli_fetch_array($res)) {
?>
<tr>
<td style="border-right: 1px dotted black; padding: 10px"><strong><?php echo
$row['mname'] ?></strong><br><?php echo $row['mdes'] ?></td>
<td class="alignright">₹ <?php echo $row['mrate'] * $row['qty'] ?></td>
</tr>
<?php
$total = $row['totalprice'];
}
?>
</tbody>
<tfoot>
<tr>
<td><strong>Total</strong></td>
<td><?php echo $total ?></td>
</tr>
</tfoot>
</table>
</td>
</tr>
</tbody>
</table>
</td>
</tr>
<tr>
<td class="content-block">
<input type="button" value="Print" onclick="myFun()" id="pt" style="padding:10px">
</td>
</tr>
<tr>
<td class="content-block">
</td>
</tr>
</tbody>
</table>
</td>
</tr>
</tbody>
</table>
```

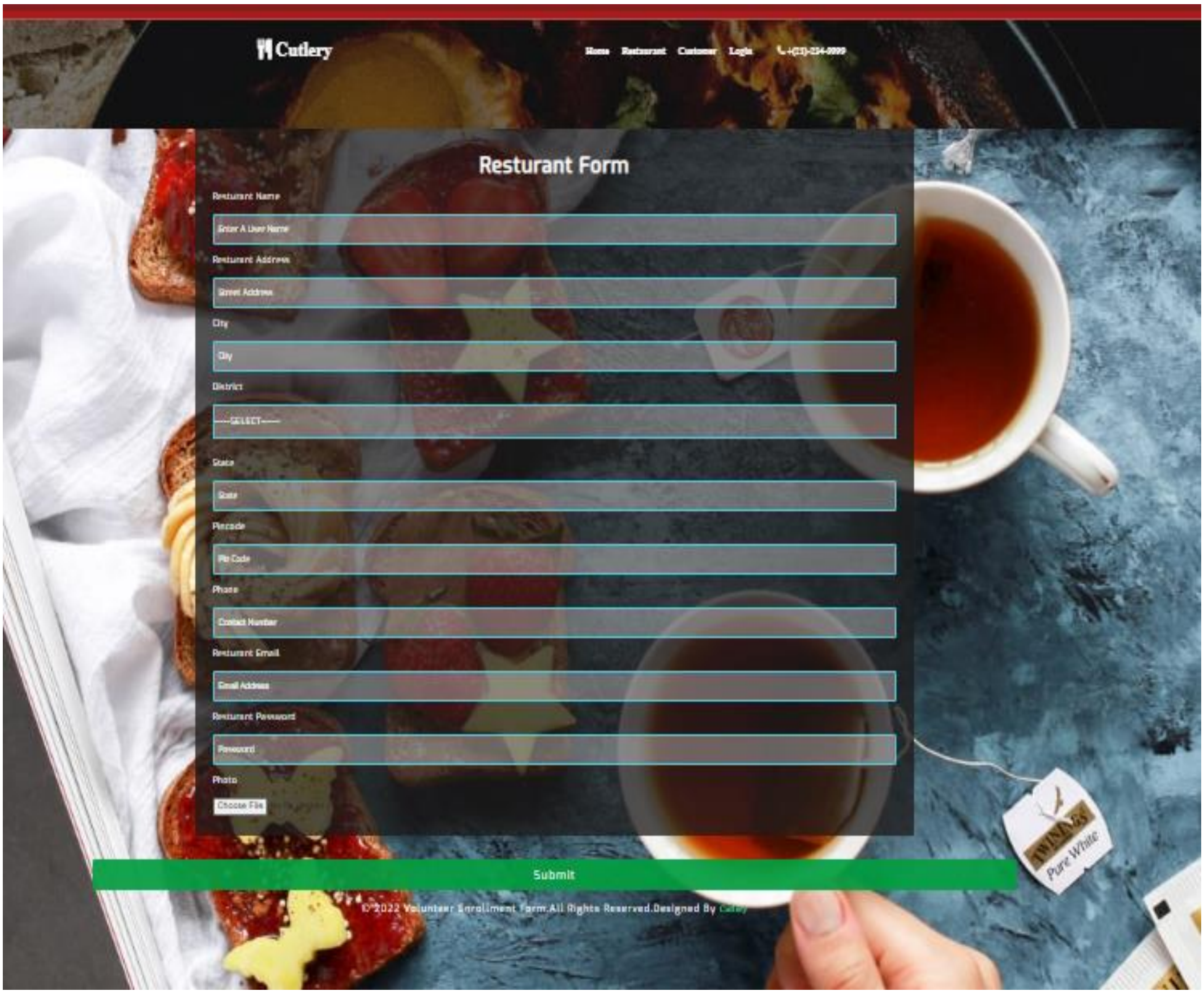
```
<div class="footer">
<table width="100%">
<tbody>
<tr>
</tr>
</tbody>
</table>
</div>
</div>
</td>
<td></td>
</tr>
</tbody>
</table>
<script>
function myFun() {
document.getElementById("pt").style.display = "none"
window.print()
window.location = "custviewo rder.php"
}
</script>
</body>
</html>
```

9.2 Screen Shots

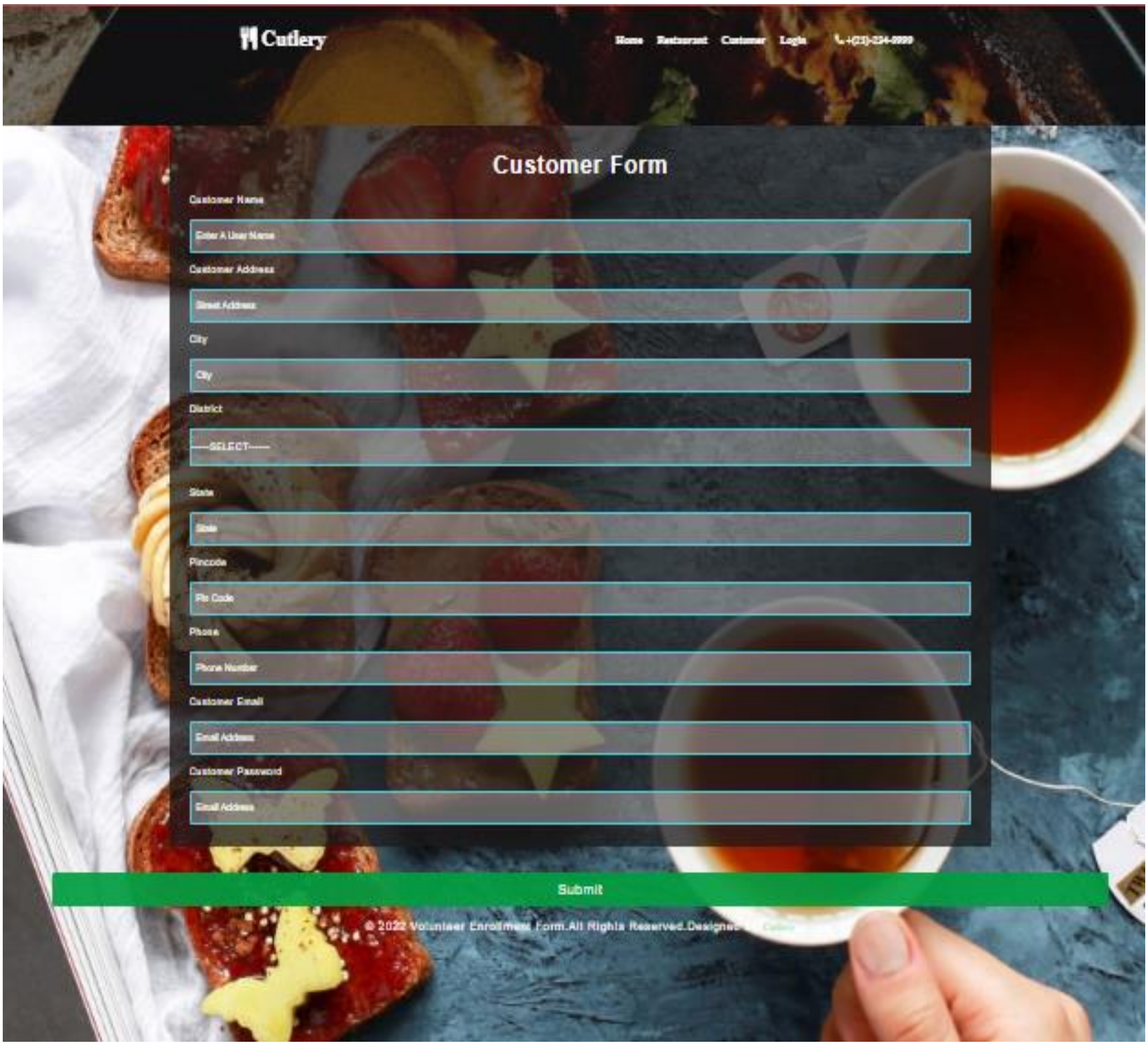
Home Page



Restaurant Registration Form



Customer Registration Form


Cutlery
Home
Restaurant
Customer
Login
+(23)-234-9999
Submit
© 2022 Volunteer Enrollment Form.All Rights Reserved.Designed by Cutlery

Customer Form

Customer Name

Enter A User Name

Customer Address

Street Address

City

City

District

-----SELECT-----

State

State

Pincode

Pin Code

Phone

Phone Number

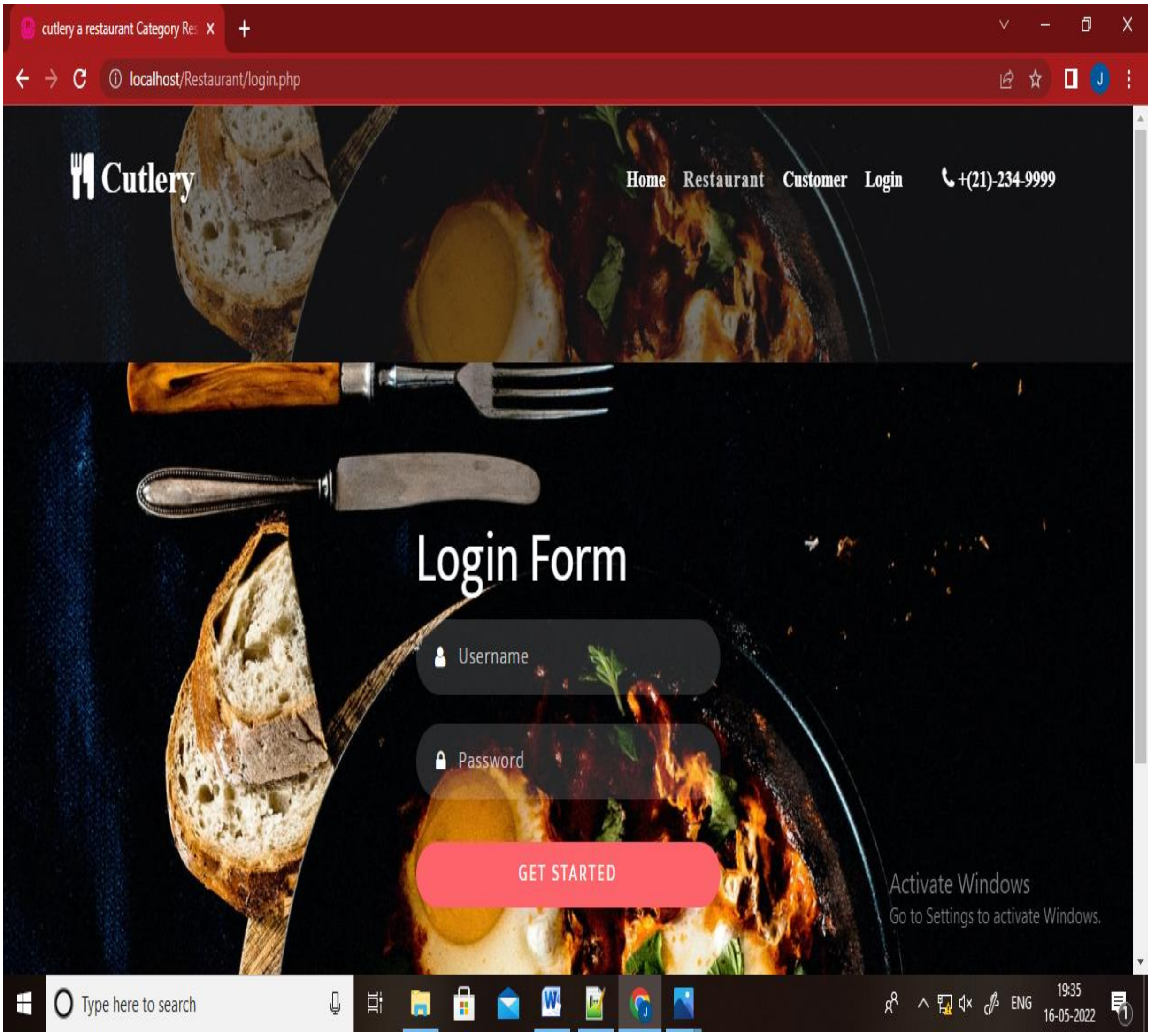
Customer Email

Email Address

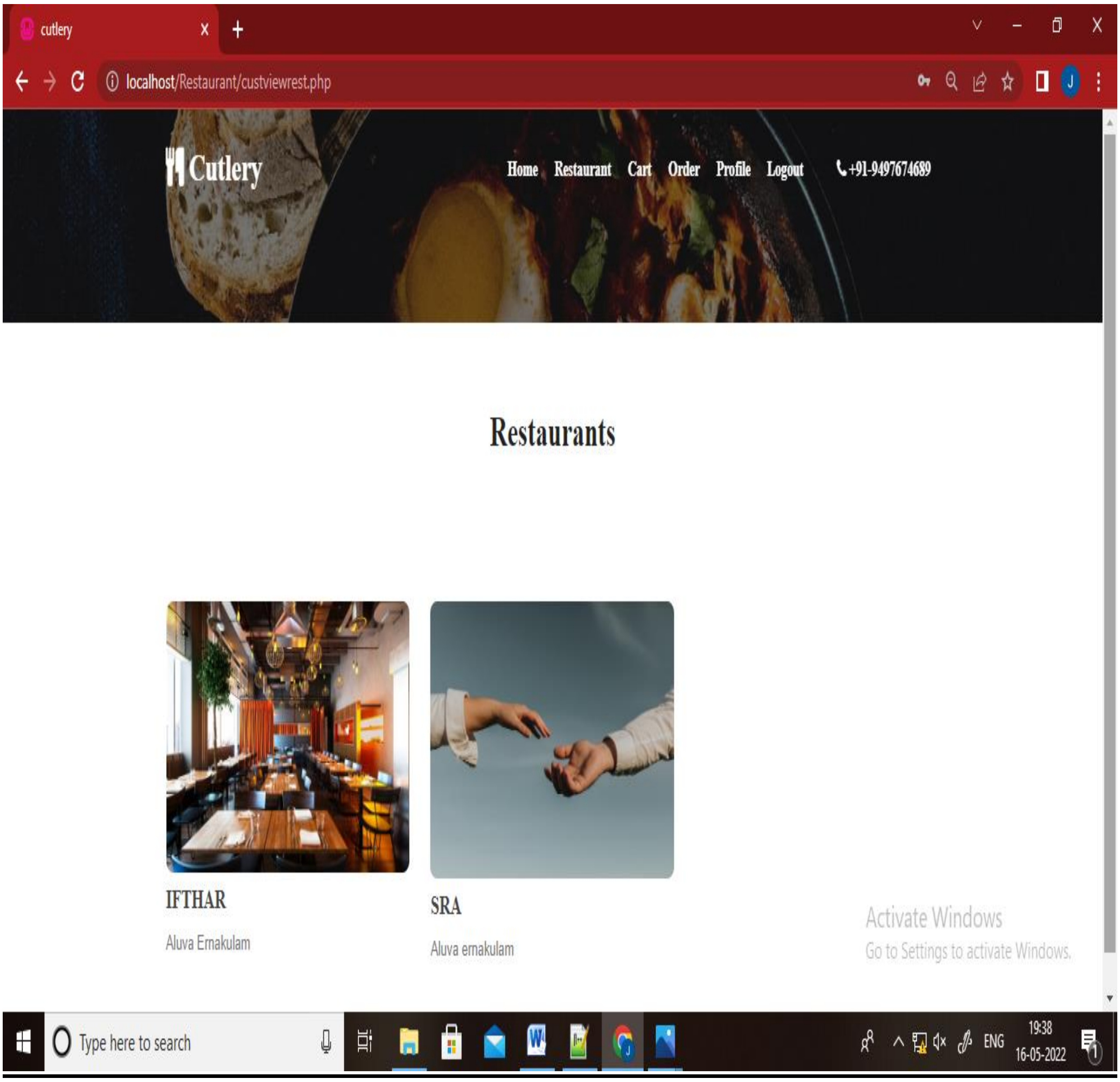
Customer Password

Email Address

Login Form


cutlery a restaurant Category Re
localhost/Restaurant/login.php
Cutlery
Home
Restaurant
Customer
Login
+(21)-234-9999
Login Form
Username
Password
GET STARTED
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
19:35
16-05-2022

Customer Home Page



Restaurants

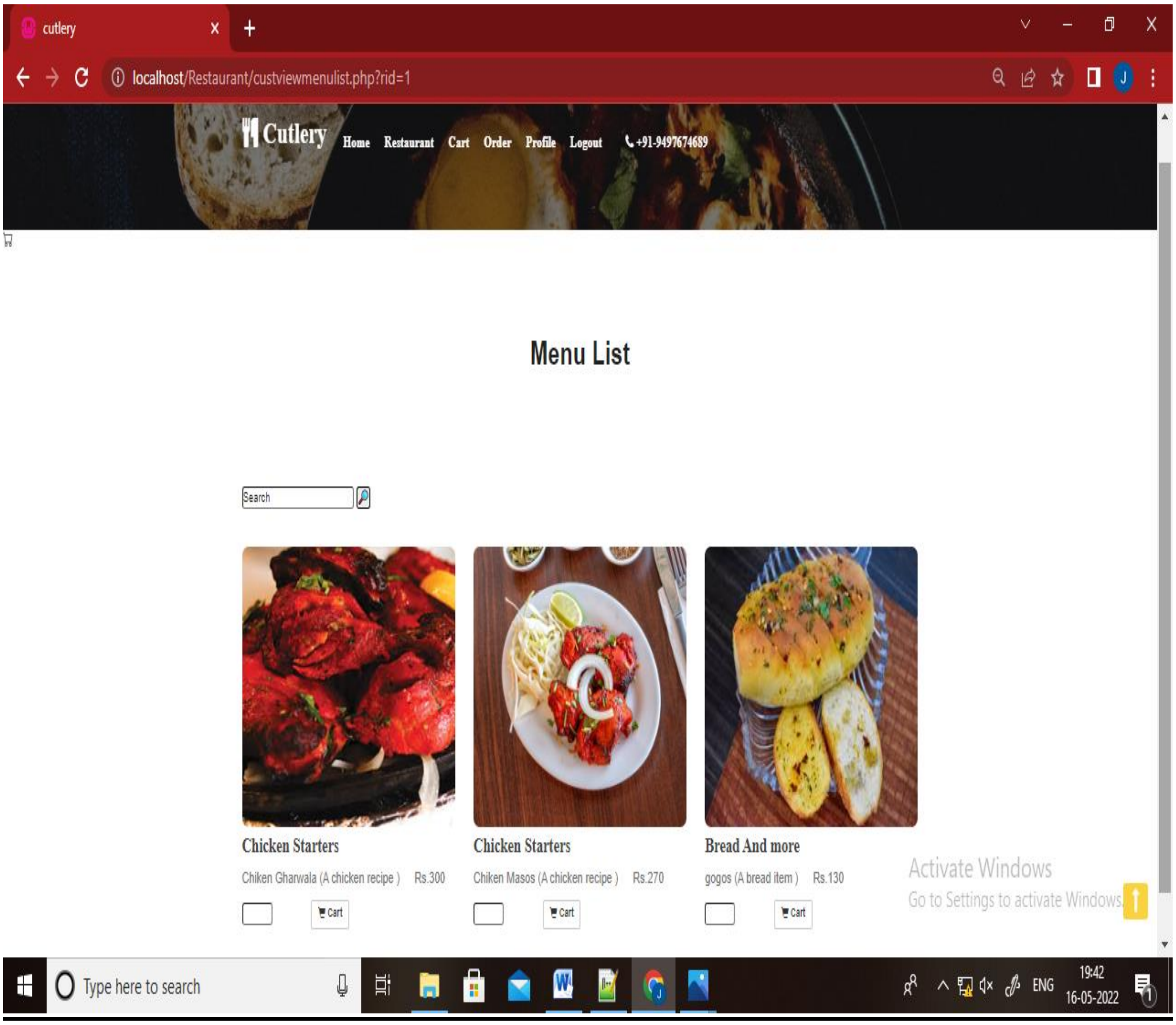
IFTHAR

Aluva Ernakulam

SRA

Aluva ernakulam

Menu List (customers)



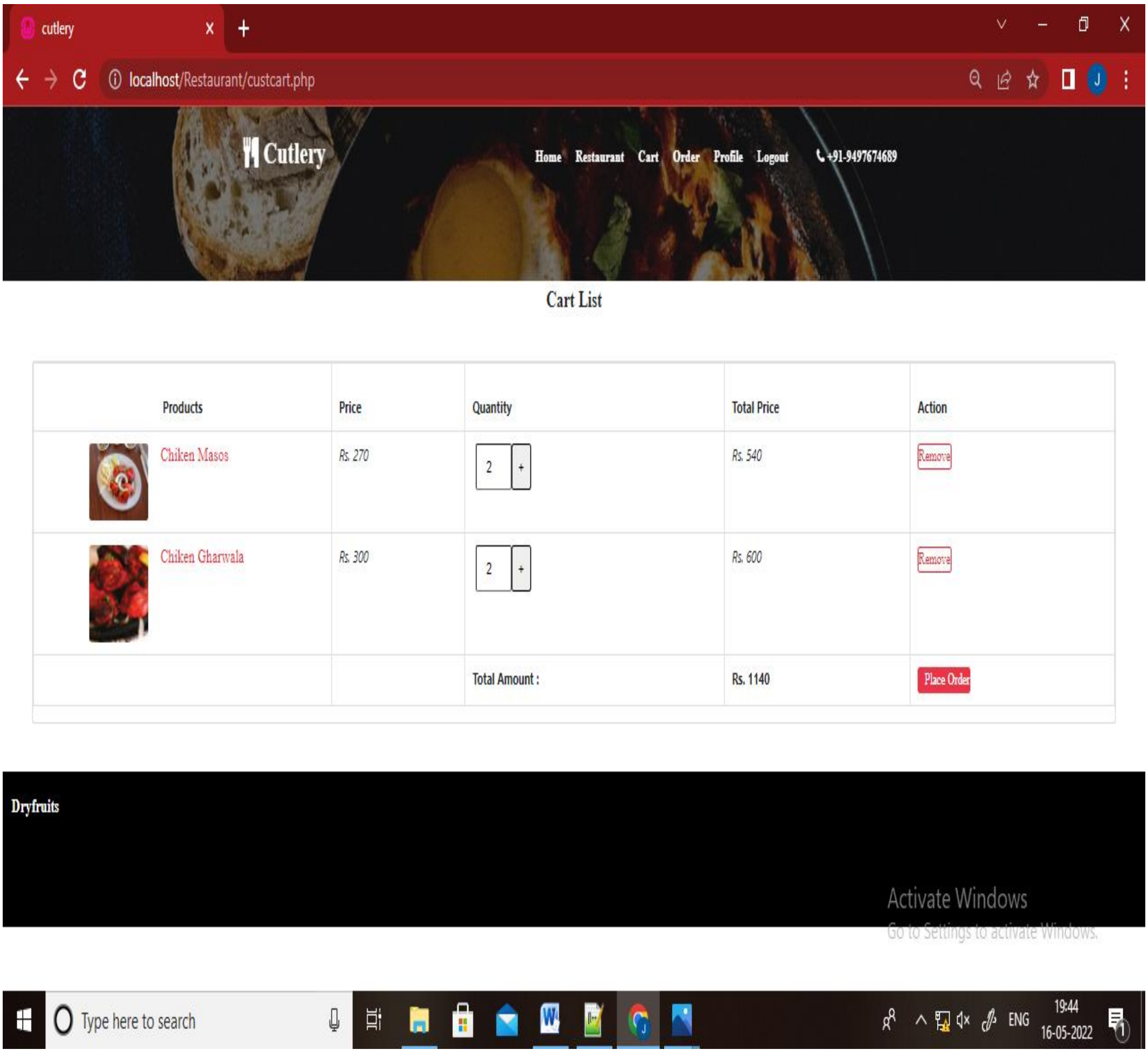
Menu List

Chicken Starters
Chiken Gharwala (A chicken recipe) Rs.300
Cart

Chicken Starters
Chiken Masos (A chicken recipe) Rs.270
Cart

Bread And more
gogos (A bread item) Rs.130
Cart

Cart(customers)



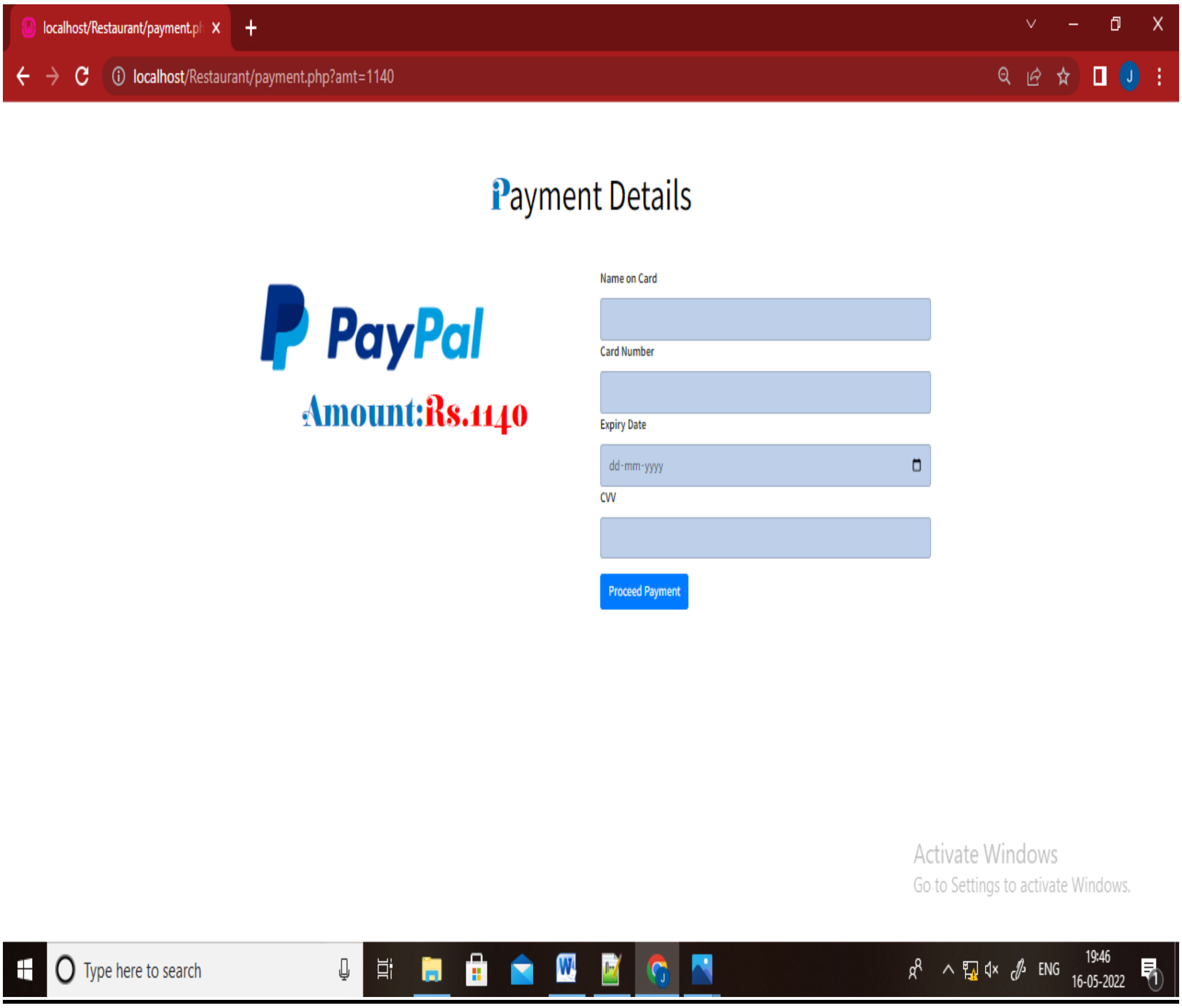
cutlery
localhost/Restaurant/custcart.php
Cutlery
Home Restaurant Cart Order Profile Logout
+91-9497674689

Cart List

Products	Price	Quantity	Total Price	Action
Chiken Masos	Rs. 270	2 +	Rs. 540	Remove
Chiken Gharwala	Rs. 300	2 +	Rs. 600	Remove
		Total Amount :	Rs. 1140	Place Order

Dryfruits
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
19:44
16-05-2022

Payment (customers)



localhost/Restaurant/payment.php?amt=1140

Payment Details

Amount:Rs.1140

Name on Card

Card Number

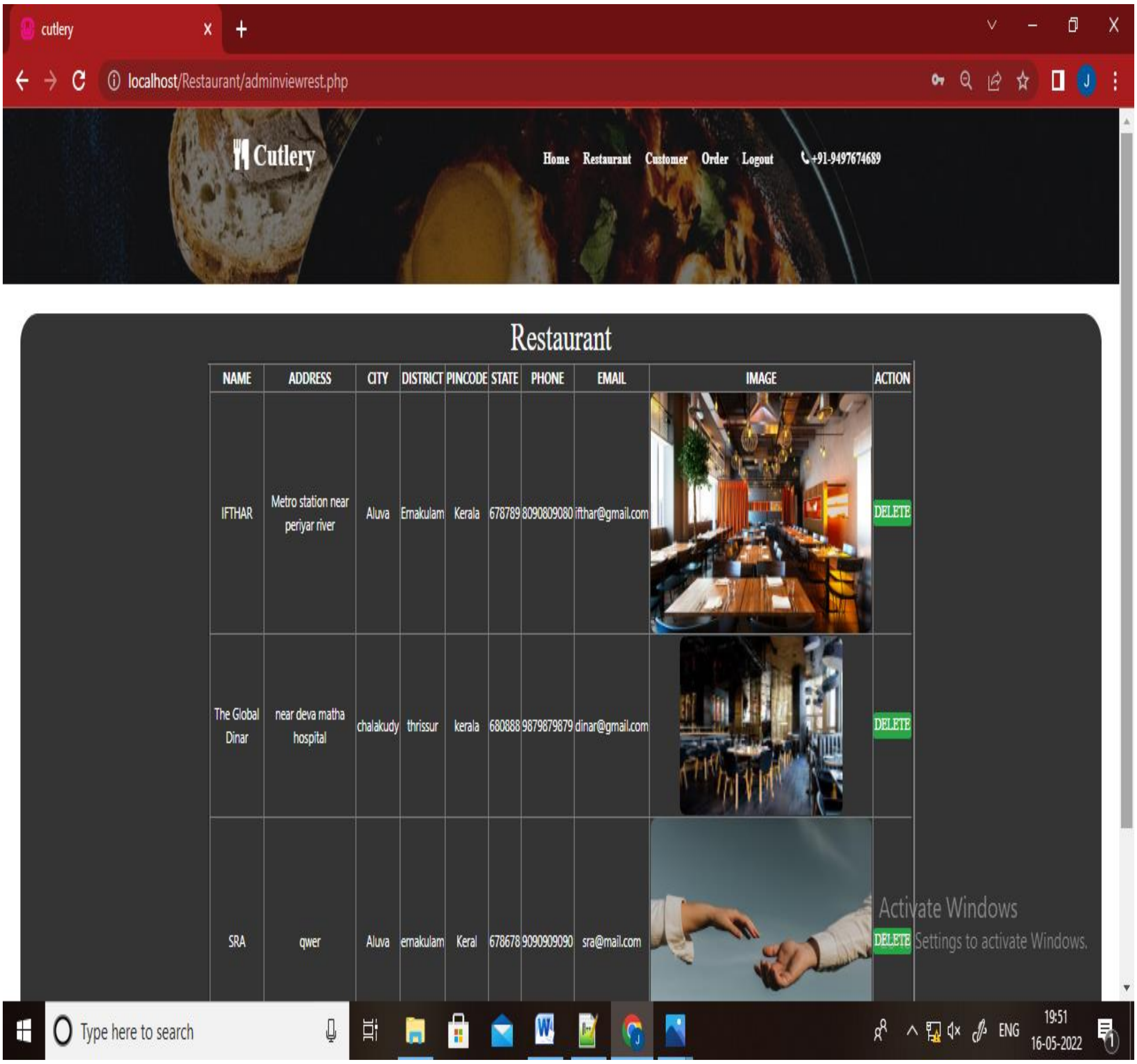
Expiry Date

dd-mm-yyyy

CVV

Admin Pages

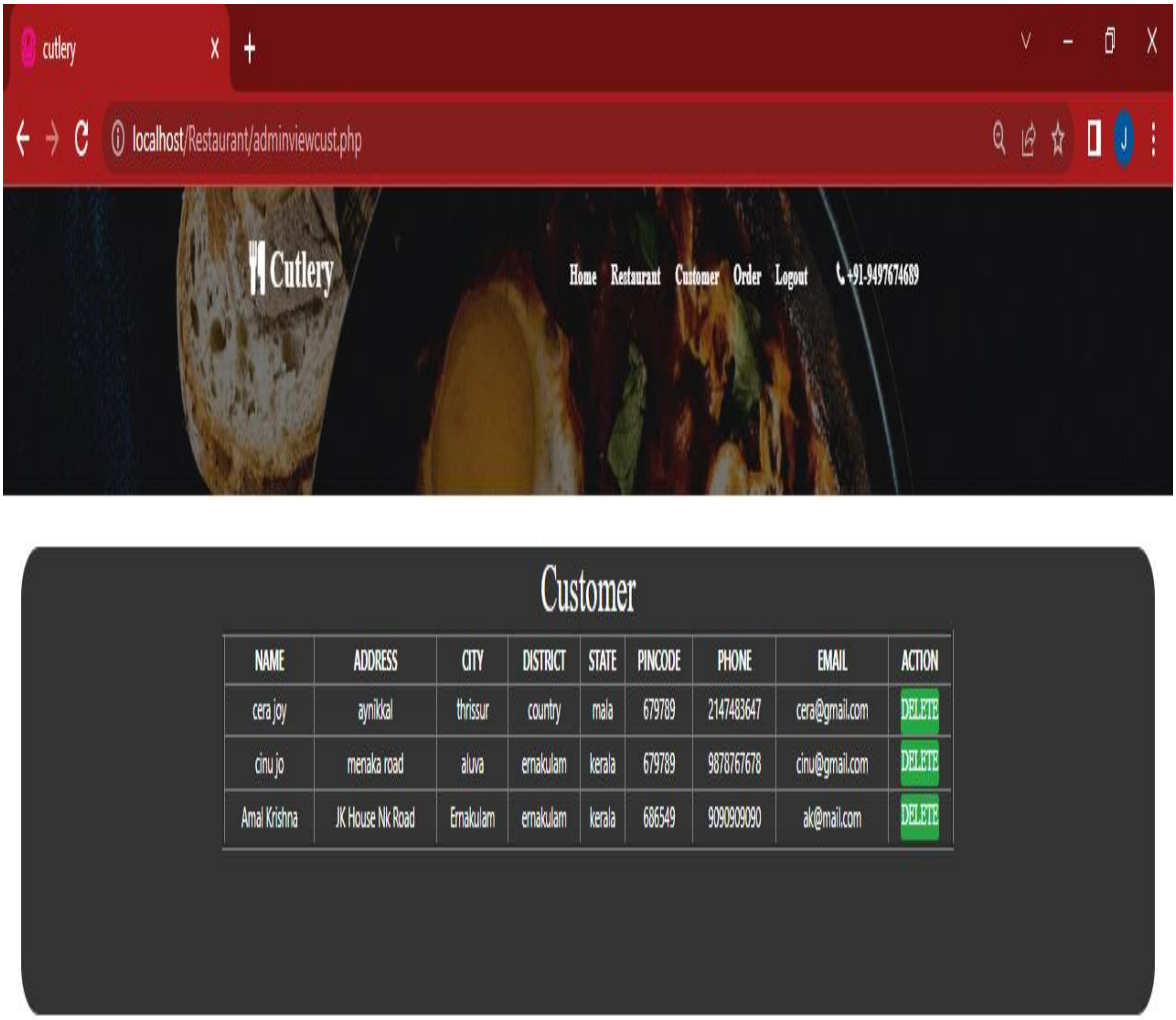
Restaurant Details



Restaurant

NAME	ADDRESS	CITY	DISTRICT	PINCODE	STATE	PHONE	EMAIL	IMAGE	ACTION
IFTHAR	Metro station near periyar river	Aluva	Ernakulam	Kerala	678789	8090809080	ifthar@gmail.com		DELETE
The Global Dinar	near deva matha hospital	chalakudy	thrissur	kerala	680888	9879879879	dinar@gmail.com		DELETE
SRA	qwer	Aluva	ernakulam	Keral	678678	9090909090	sra@mail.com		DELETE

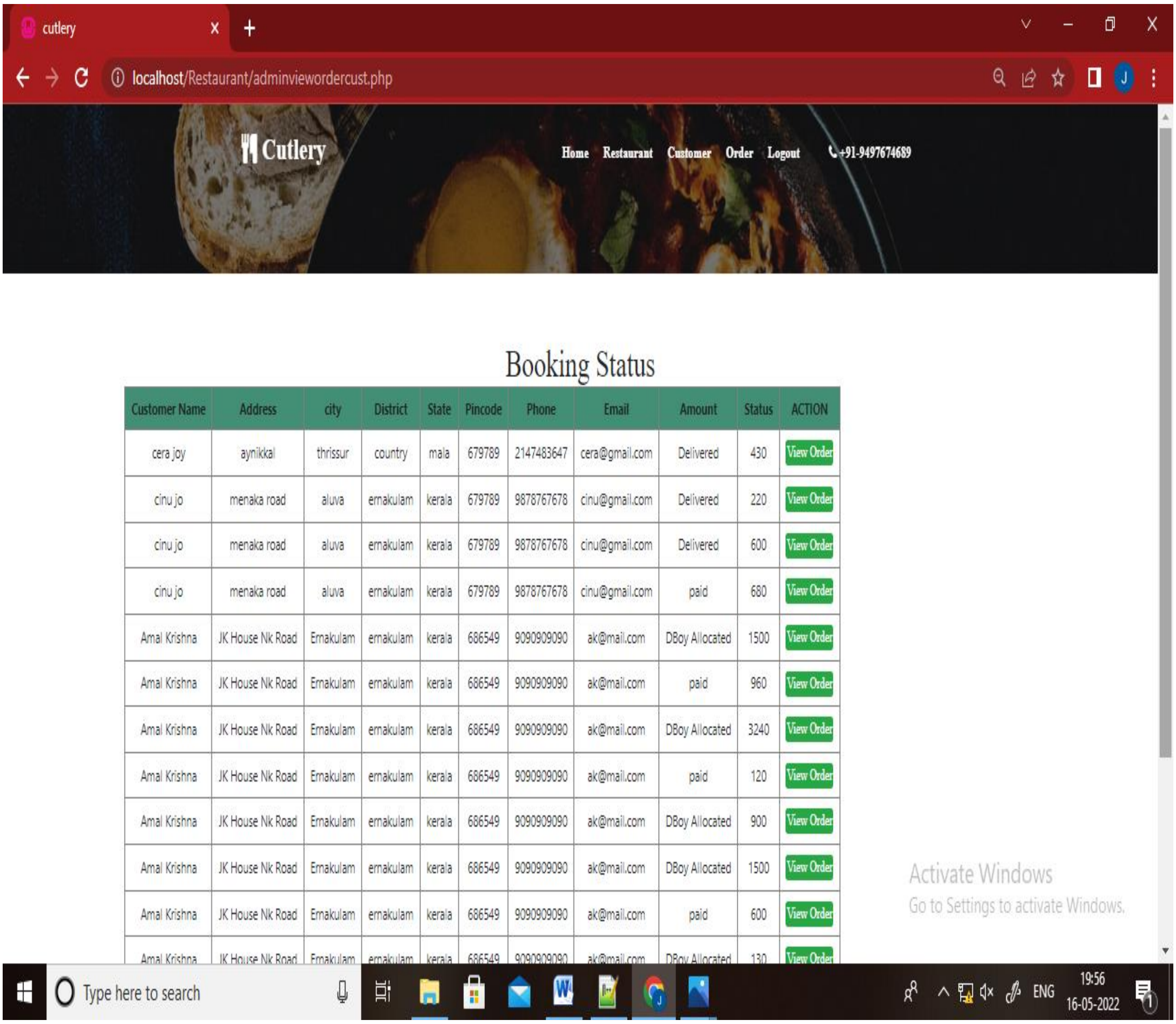
Customer Details


cutlery
localhost/Restaurant/adminviewcust.php
Cutlery
Home Restaurant Customer Order Logout
+91-9497674689

Customer

NAME	ADDRESS	CITY	DISTRICT	STATE	PINCODE	PHONE	EMAIL	ACTION
cera joy	aynikkal	thrissur	country	mala	679789	2147483647	cera@gmail.com	DELETE
cinu jo	menaka road	aluva	ernakulam	kerala	679789	9878767678	cinu@gmail.com	DELETE
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	DELETE

Order Details

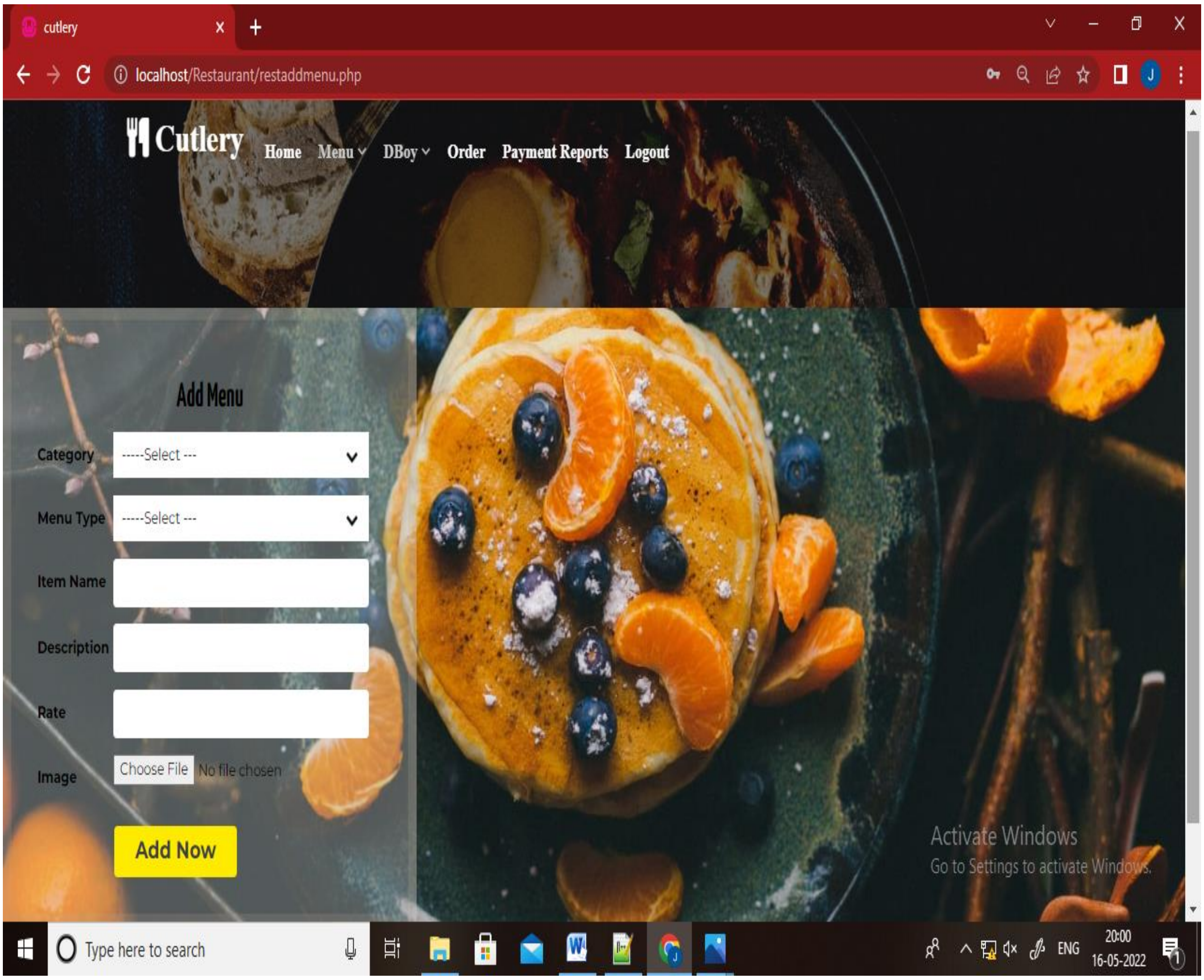


Booking Status

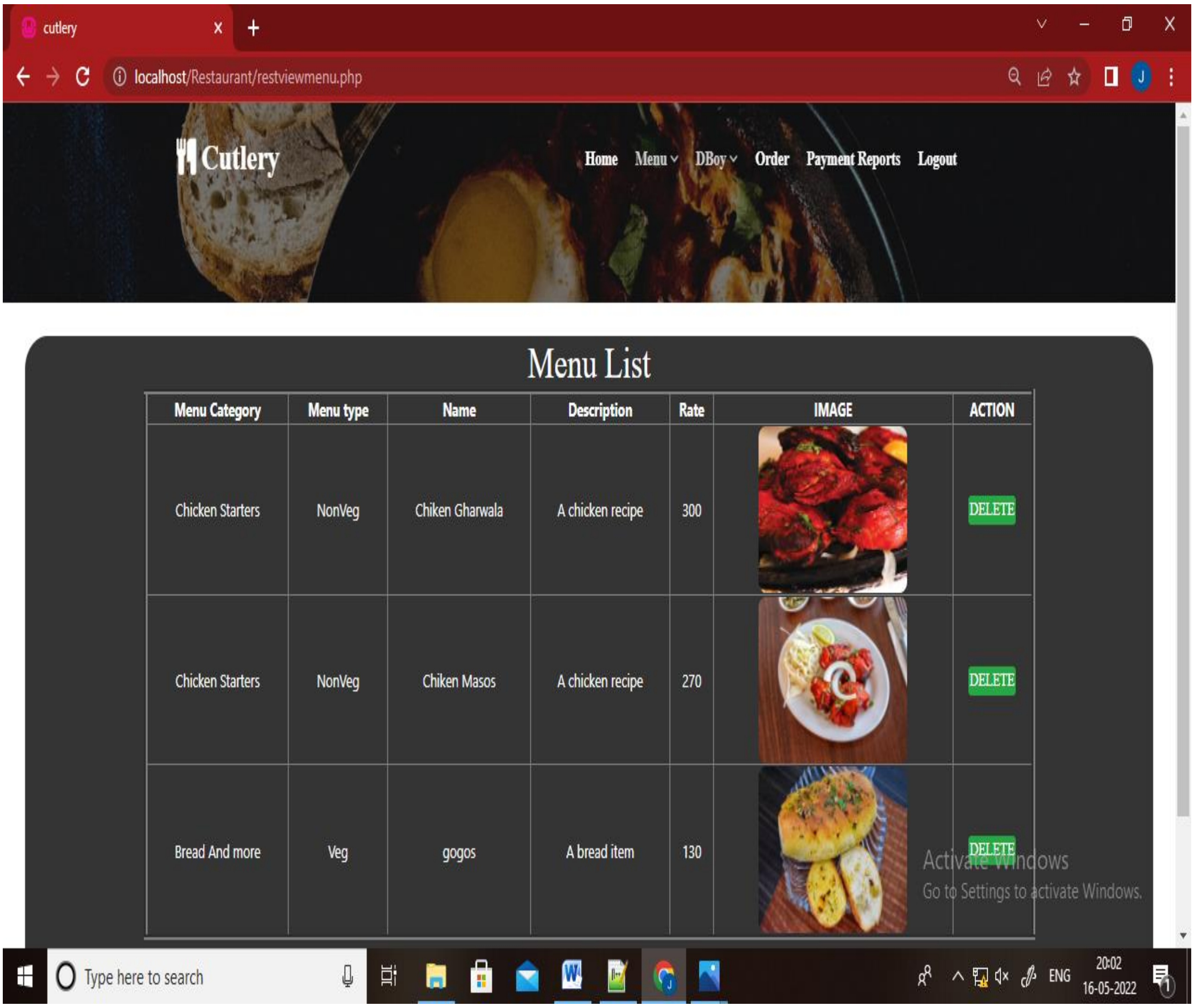
Customer Name	Address	city	District	State	Pincode	Phone	Email	Amount	Status	ACTION
cera joy	aynikkal	thrissur	country	mala	679789	2147483647	cera@gmail.com	Delivered	430	View Order
cinu jo	menaka road	aluva	ernakulam	kerala	679789	9878767678	cinu@gmail.com	Delivered	220	View Order
cinu jo	menaka road	aluva	ernakulam	kerala	679789	9878767678	cinu@gmail.com	Delivered	600	View Order
cinu jo	menaka road	aluva	ernakulam	kerala	679789	9878767678	cinu@gmail.com	paid	680	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	DBoy Allocated	1500	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	paid	960	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	DBoy Allocated	3240	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	paid	120	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	DBoy Allocated	900	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	DBoy Allocated	1500	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	paid	600	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	DBoy Allocated	130	View Order

Restaurant Pages

Add Menu



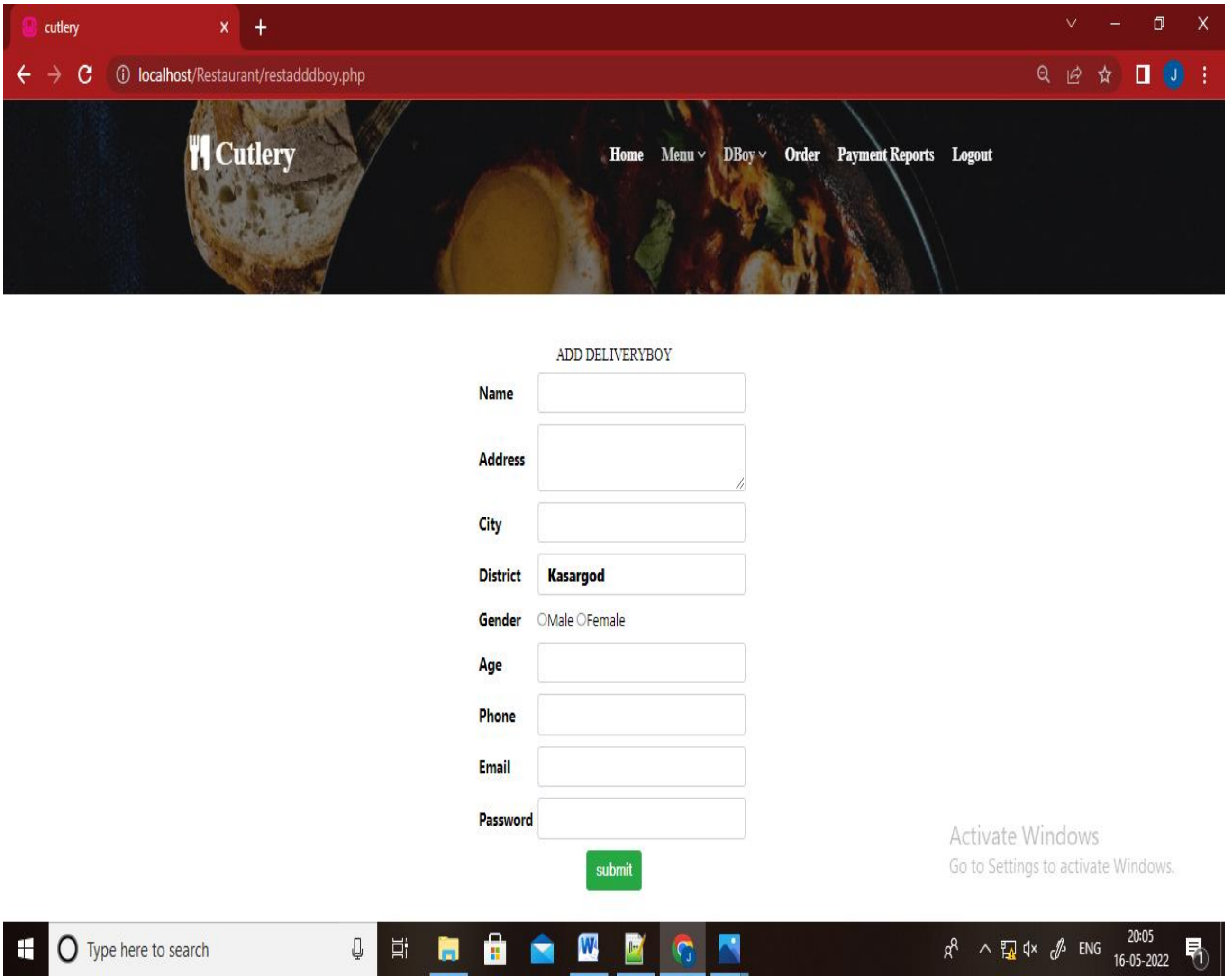
View Menu



Menu List

Menu Category	Menu type	Name	Description	Rate	IMAGE	ACTION
Chicken Starters	NonVeg	Chiken Gharwala	A chicken recipe	300		DELETE
Chicken Starters	NonVeg	Chiken Masos	A chicken recipe	270		DELETE
Bread And more	Veg	gogos	A bread item	130		DELETE

Add Delivery Boy

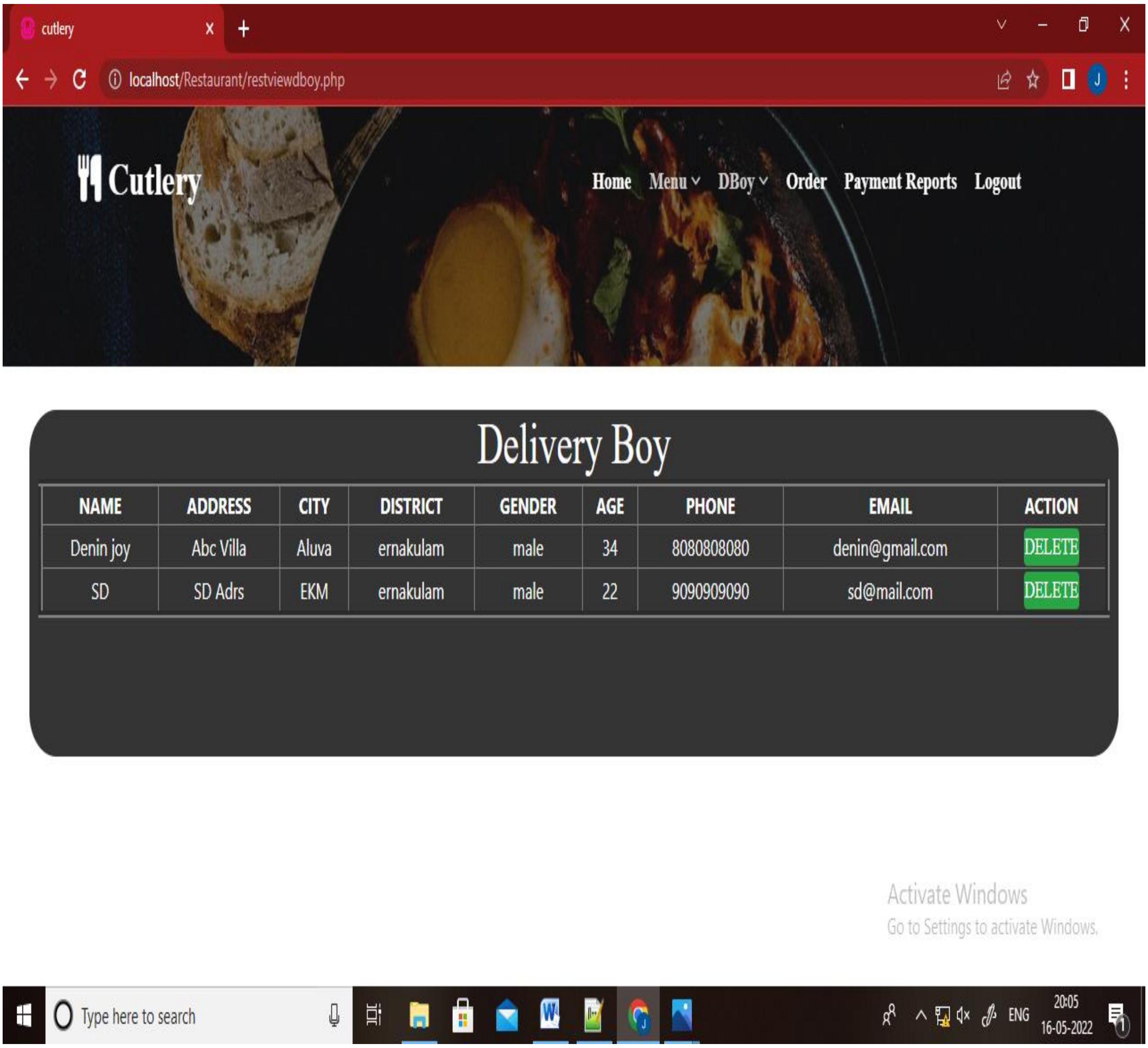


ADD DELIVERYBOY

Name	
Address	
City	
District	Kasargod
Gender	○Male ○Female
Age	
Phone	
Email	
Password	

submit

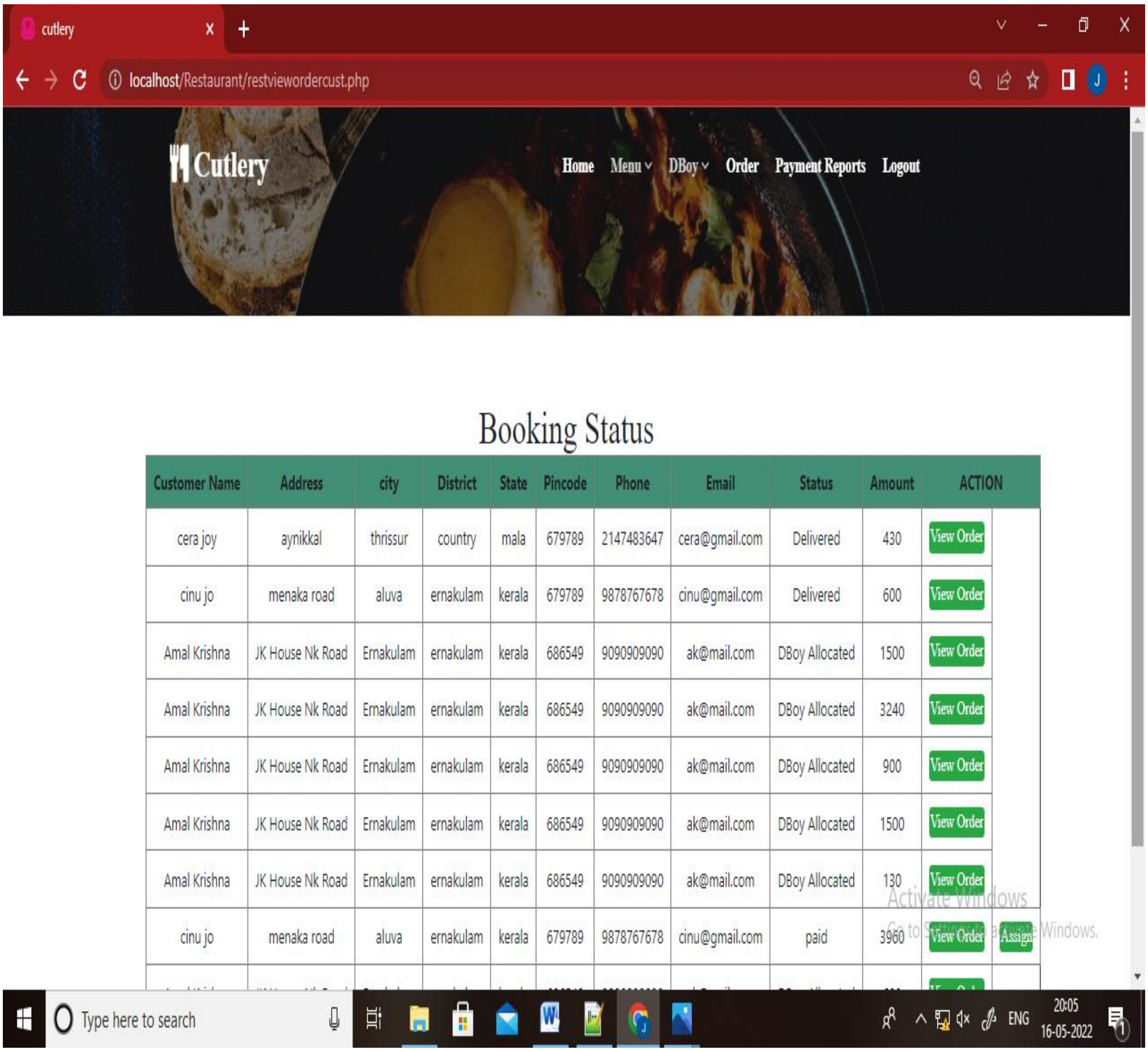
View Delivery Boy



Delivery Boy

NAME	ADDRESS	CITY	DISTRICT	GENDER	AGE	PHONE	EMAIL	ACTION
Denin joy	Abc Villa	Aluva	ernakulam	male	34	8080808080	denin@gmail.com	DELETE
SD	SD Adrs	EKM	ernakulam	male	22	9090909090	sd@mail.com	DELETE

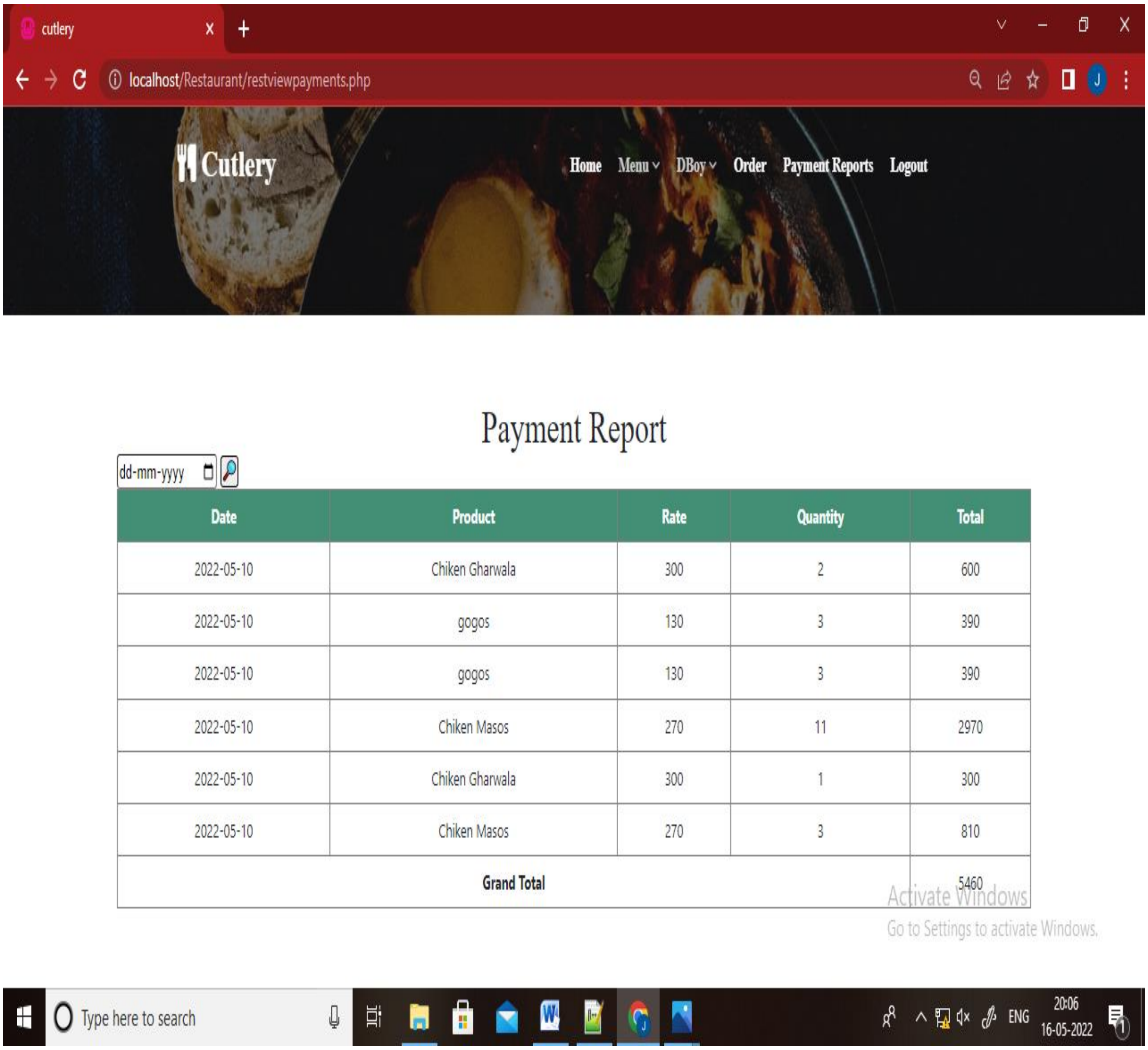
Order Details



Booking Status

Customer Name	Address	city	District	State	Pincode	Phone	Email	Status	Amount	ACTION
cera joy	aynikkal	thrissur	country	mala	679789	2147483647	cera@gmail.com	Delivered	430	View Order
cinu jo	menaka road	aluva	ernakulam	kerala	679789	9878767678	cinu@gmail.com	Delivered	600	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	DBoy Allocated	1500	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	DBoy Allocated	3240	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	DBoy Allocated	900	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	DBoy Allocated	1500	View Order
Amal Krishna	JK House Nk Road	Ernakulam	ernakulam	kerala	686549	9090909090	ak@mail.com	DBoy Allocated	130	View Order
cinu jo	menaka road	aluva	ernakulam	kerala	679789	9878767678	cinu@gmail.com	paid	3960	View Order Assign

Payment Reports



Payment Report

Date	Product	Rate	Quantity	Total
2022-05-10	Chiken Gharwala	300	2	600
2022-05-10	gogos	130	3	390
2022-05-10	gogos	130	3	390
2022-05-10	Chiken Masos	270	11	2970
2022-05-10	Chiken Gharwala	300	1	300
2022-05-10	Chiken Masos	270	3	810
Grand Total				5460