N.S.S. COLLEGE OF ENGINEERING

PALAKKAD, KERALA - 678008

UNIVERSITY OF CALICUT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MINI PROJECT REPORT

2014-2018

AUTOMATED FOOD ORDERING

SUBMITTED BY

ANUPRIYA J	**NSAOECS012**
KISHAN C	**NSAOECS034**
NAIR POOJA UNNIKRISHNAN	**NSAOECS039**
SRIKANT VIJAY	**NSAOECS057**
SRUTHI K S	**NSAOECS058**

GUIDED BY

Mrs. SRUTHY MANMADHAN

Assistant Professor

Dept. of Computer Science & Engineering

N.S.S.C.E, Palakkad

N.S.S. COLLEGE OF ENGINEERING

PALAKKAD, KERALA - 678008

UNIVERSITY OF CALICUT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that this is the bonafide report of the Mini Project entitled "**AUTOMATED FOOD ORDERING**" done by **ANUPRIYA J, KISHAN C, NAIR POOJA UNNIKRISHNAN, SRIKANT VIJAY** and **SRUTHI K S** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science & Engineering under University of Calicut.

GUIDED BY

Asst. Prof. SRUTHY MANMADHAN

HEAD OF THE DEPARTMENT

Prof. VIJI RAJENDRAN V

STAFF IN CHARGE

Asst. Prof. KIRAN V K

ACKNOWLEDGEMENT

Many noble hearts contributed immense inspiration and technical assistance for the successful completion of our mini-project work. We are unable to express our gratitude in words to such individuals.

Many noble hearts contributed immense inspiration and technical assistance for the successful completion of our mini-project work. We are unable to express our gratitude in words to such individuals.

We acknowledge our sincere thanks to the management of N.S.S College of Engineering for their help in the successful completion of our project. We would also like to express our deep regard to our principal **Prof. Sudha T**, for making our mini-project a grant success.

We take this opportunity to express our profound gratitude to our project guide and Head of the Department, **Prof. Viji Rajendran V**, for all her guidance and valuable support towards making our project a grant success.

We are also thankful to our teachers, **Mrs. Sruthy Manmadhan, Mr. Kiran V K, Mrs. Ramya G** and **Mrs. Manasa N A S** for their valuable guidance throughout our project.

We would also like to thank all the non-teaching staff of our department for their constant encouragement throughout our project. This helped us in proper completion, installation and demonstration of our project. Last, but not the least, we take pleasant privilege in expressing our heart full thanks to our friends who were of precious help in completing this project.

ABSTRACT

The project focus on the creation of a web application *Bay view* for the customers of a restaurant and a web application on the manager-side for improving the customer-manager interactions and thus eliminating the present scenario of having an ineffective communication between them, based on the Automated Food Ordering (AFO) System.

Based on the AFO system, a technology roadmap of the project architecture is planned out: JAVA plus HTML plus PHP plus MySQL database, which simplifies the development process, and thereby develop a better system with scalability and maintainability for improving the nullifying relationships between the customers and restaurant managers by directly providing the needed details and information to a particular customer quickly and helping the managers do their work, also monitoring restaurant activities easily.

CONTENTS

1. INTRODUCTION

Technology has entered almost every field in our life, but still its effect is not yet that evident in the food industry, especially the food serving outlets including restaurants, hotels. Even today, most of the restaurants in India follow the traditional pen and paper method to take orders from customers, wastes a lot of time of both, the customer and the restaurant. This work aims to substitute the traditional pen and paper method by automating the food-ordering process in restaurant and thus improving the dining experience of the customer.

The **Automated Food Ordering (AFM)** System improves efficiency and accuracy for restaurants by saving time, eliminating human errors, getting customers feedback. As the system is automated, it becomes economical even from restaurants point of view, as it reduces manpower and it just requires time investment in installing the devices at tables. The automated food ordering system consists of three main views that is the **customer module**, **manager module** and **kitchen display**.

The customer module is the main module within the system where the customer interactions with the restaurant take place, whose structure is modelled for the project.

The manager module is the module which is designed for the purpose of interactions of the manager of the restaurant with the customers and the chefs.

The kitchen display consisting of a display screen were the chefs' can easily interact with the system.

2. REQUIREMENT ANALYSIS

2.1 SCOPE OF THE PROJECT

Automated food ordering system will be a web based application whose main aim is to simplify and improve the efficiency of the ordering process for both customer and restaurant, minimize manual data entry, security during order placement process and ensure data accuracy. The project is focused on making the restaurant fully automated so that it is easier to co-ordinate various work activities that go on inside a typical restaurant. Customers will be able to view product menus, the cost and also a visual confirmation of the orders placed. The Web application is designed using JS, HTML, PHP and CSS.

2.1.1. EXISTING SYSTEM

Full Service Restaurant:

In this system, every time a customer enters the restaurant and occupies his table, a waiter comes at his table and presents him a menu card, with the record of food-items. Waiter then waits with a notepad and a pen to take down the customer's order. The waiter then notes down the order of the Customer in his notepad and the record is stored in the paper. Finally, the record is given to the chef in the kitchen. Although the food ordering process in this system is very easy, there are many drawbacks here.

Self Service Restaurant:

This process required the guests to place orders at the service counter in the restaurant. The guests have to decide in advance before placing the order at the counter, of which menu items to order. Menu is mostly in the form of posters that are hung behind the order counter.

Limitation

- Inconvenience of customers who wish to have a physical copy of the menu.
- Time consuming
- Lack of visual confirmation that the order was placed correctly.

2.1.2 PROPOSED SYSTEM

Though many steps were taken earlier to automate the food ordering system, they have not yet become popular enough to be adopted by restaurants all over the world. They have many limitations like, they are not customer friendly. Sometimes these systems may take a lot of time for taking the order, don't have proper real time feedback system between customers

and restaurants managers and they are not cost effective. To overcome the limitations of these systems, we propose the design of touch based digital ordering systems for restaurants.

The Objectives of the proposed system are:

- The restaurant-owner's laptop/tablet will keep track of customer records and also customize menu using server application.
- The central database is used for restaurant-owner to store updated menu information and order details.
- Three main areas of restaurant are connected using wireless technology
- Security of data.
- Ensures data accuracy during order placement process.
- Minimized manual data entry.
- Greater efficiency since data processing is very fast.
- User friendly and interactive interface with provision for customer to view menus and have a visual confirmation that the order was placed correctly.
- Minimized time requirement during the order placement process.
- Greatly simplifies the ordering process for both customer and restaurant.

2.2 IDENTIFIED SYSTEM REQUIREMENTS

These consist of the hardware and software components of a computer system that are required to install in order to use the software efficiently.

2.2.1 SOFTWARE REQUIREMENTS

- Operating system: Windows XP / windows 7 and above
- Technologies: HTML, CSS, PHP, JavaScript.
- Database: MySQL
- Tool: Sublime Text 3, WAMP Server

2.2.2 HARDWARE REQUIREMENTS

- Processor: Intel dual core or above
- Processor Speed: 1.0GHZ or above
- RAM: 1 GB RAM or above
- Laptops having Windows 7 OS and above

2.3 PRIORITIZED REQUIREMENTS

2.3.1 Customer Manager Interface:

The Web based application contains provision for customers to view menu details and place orders. It is then send to the manager. The manager can make amendments if they are valid.

2.3.2 Centralised Database:

A database containing all the information regarding the food items like item name, savour, price of each food, images etc. should be maintained. Only the manager has access to the database. Insertion, deletion and editing of data is solely under his/her control

2.4 FUNCTIONAL REQUIREMENTS

This project consists of 3 main modules as follows:

2.4.1 User Tablet(module 1)

- These types of tablets are especially made for the normal users coming to the restaurant.
- Customer can order items. Bill automatically goes in the name of that particular user. These tablets will consist of the whole menu of the restaurant. The items in the menu are non - editable for these types of the tablets.
- They will be enabled with the Wi-Fi connectivity.

2.4.2 Manager's Tablet(module 2)

- These desktops are especially for the use of the restaurant manager.
- The manager should be able to control the function of whole restaurant from a single desktop/tablet.
- He can access any tablet and should be able to make changes to the menu.
- Also he can change price of particular item or disable particular item which is not available at that particular time.

2.4.3 Kitchen Display(module 3)

- These are present at the kitchen near chef so that he should be able to see what a particular has ordered.
- All the ordered items are displayed on the screen giving the table number below.
- They should be sufficiently large to be seen by chef at a reasonable distance.
- Chef should be able to notify when a particular item is ready.

2.5 NON-FUNCTIONAL REQUIREMENTS

All of the application data is stored in a My SQL database, and therefore a My SQL Database must also be installed on the host computer. As with Apache2, this software is freely available and can be installed and run under most operating systems. The server hardware can be any computer capable of running both the web and database servers and handling the expected traffic. For a small scale restaurant that is not expecting to see much web traffic, an average personal computer may be appropriate. Once the site starts generating more hits, though, it will likely be necessary to upgrade to a dedicated host to ensure proper performance. The exact cut offs will need to be determined through a more thorough stress testing of the system

- There should be sufficient network bandwidth
- Backup- provision for data backup
- Maintainability- easy to maintain
- Performance/ response time- fast response
- Usability by target user community- easy to use
- Expandability- needs to be future proof or upgradable
- Safety- should be safe to use.

3. FEASIBILITY ANALYSIS

This is an evaluation and analysis of the potential of the proposed project which is based on extensive investigation and research to support the process of decision making. It assesses the operational, technical, cultural, legal and economic merits of the proposed project. The feasibility study is intended to be a preliminary review of the facts to see if it is worthy of proceeding to the analysis phase. From the systems analyst perspective, the feasibility analysis is the primary tool for recommending whether to proceed to the next phase or to discontinue the project.

3.1 TECHNICAL FEASIBILITY

This assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the needs of the proposed system. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements. To develop the application, a high speed internet connection, a database server, a web server and suitable software development kits are required.

3.2 OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well the project will support the customer and the service provider during the operational phase. It is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. The system will create a web based interface where manager can share and update information, customers can view the menu details, place orders and view bill.

3.3 ECONOMIC FEASIBILITY

This assessment aims to determine the positive economic benefits to the organization that the proposed system will provide. It typically involves a cost/ benefits analysis and it's the most frequently used method for evaluating the effectiveness of a new proposed system. The language, development kit, and editors can be used without any cost. The development environment, database management software, MySQL are free software's which can be readily downloaded from the internet free of cost. All other factors in this project are absolutely free of cost, and hence, so as far as the functions and constraints concerned, this project is economically feasible.

3.4 SCHEDULE FEASIBILITY

It is the measure of how reasonable the project time table is or the deadline is reasonable or not. During the lack of time or the time become mandatory, we must finish the project within a given time period.

The Schedule for the project is as follows:

- Stage 1 (Jan 16 - Jan 23) –Feasibility Study.
- Stage 2 (Jan 24 - Jan 27) –Requirement Analysis.
- Stage 3 (Jan 27 – Feb 6) – Design
- Stage 4 (Feb 7 – Feb 14) – Front End.
- Stage 5 (Feb 1 - March 1) - Back End.
- Stage 6 (March 2 – March 15) –Coding and implementation.
- Stage 7 (March 16 – March 22) –Testing.
- Stage 8 (March 23 – April 1) –Submission
- Stage 9 (March 29 – March 31) –Project close and final report.

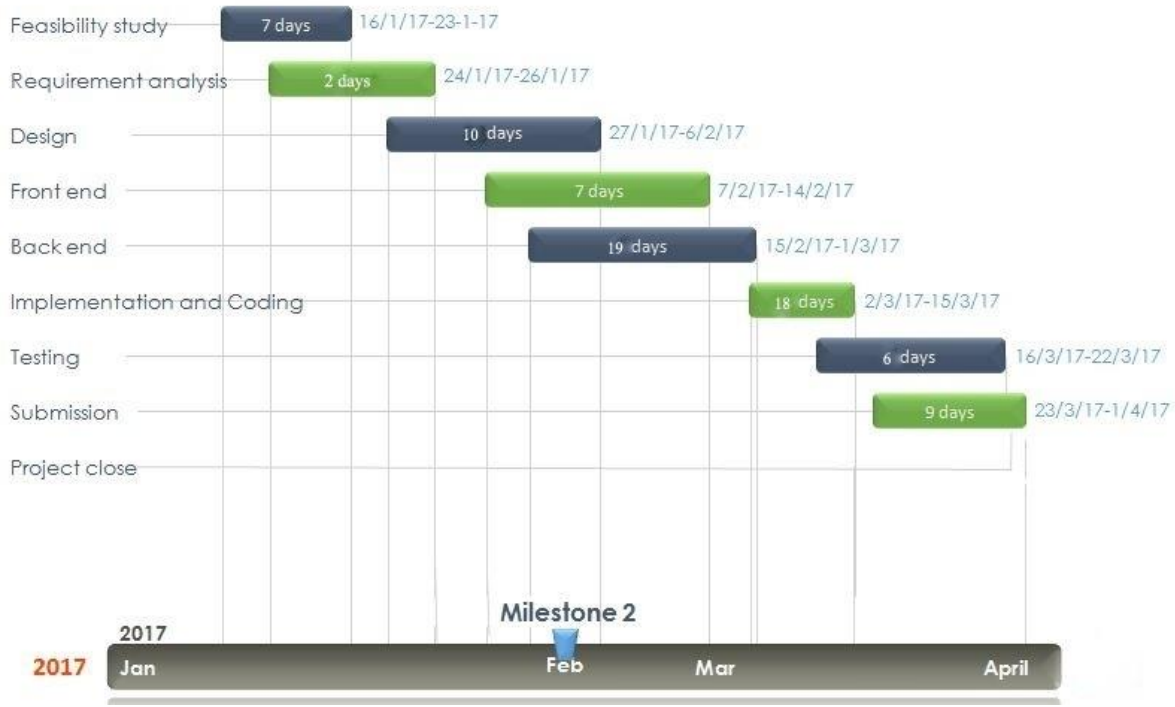


Fig.3.4 Timeline Diagram

3.5 CULTURAL (POLITICAL) FEASIBILITY

The app is developed in an environmental and cultural friendly manner. The automated food ordering app is beneficial to the customer - restaurant society as it improves the interaction between them. The project do not lead to usage of any kind of equipment's or gadgets which are harmful to the society and is done in an ethical way to bring it to reality, hence making it culturally feasible.

3.6 LEGAL FEASIBILITY

While operating the service, it should be monitored that the information and design does not violate any existing copyrights. Since the source code of the system will be licensed using a very permissive free software license (such as GPL or MIT), using free and open source software for development are not expected to pose any legal difficulties. Also, all the development tools and softwares to be used are freewares which can be downloaded from the internet without violating any piracy laws, thus declaring the project to be legally feasible in all means.

4 SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

This is a brief structure(Fig. 4.1) which depicts the environment in which a software system exists and helps in communicating about what lies outside the system boundary.

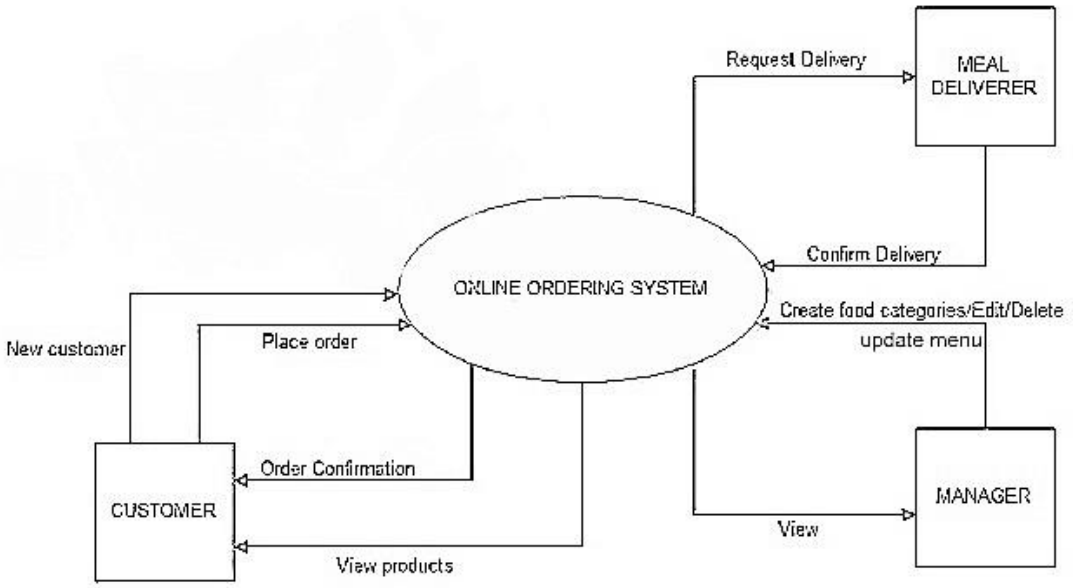


Fig.4.1 Overall System Architecture for Automated Food Ordering System

4.2 DATA FLOW DIAGRAM

It is a two-dimensional diagram that explains how data is processed and transferred in as system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output.

The Dataflow diagram of the project is shown below. There are three levels for the dataflow diagrams

Level 0 DFD :

The customer will have their own user_name and password. Thus the Customer can login and order food items.

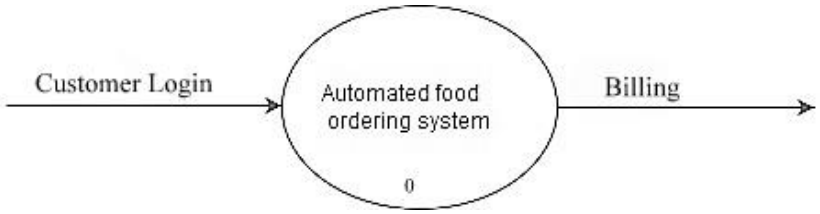


Fig.4.2 Level 0 DFD

Level 1 DFD:

This level contains all the operations that are available on the web-application for the customers and manager.Customer module deals with the provisions for customers to place order. The manager module deals with the current updates of menu and all other related activities.

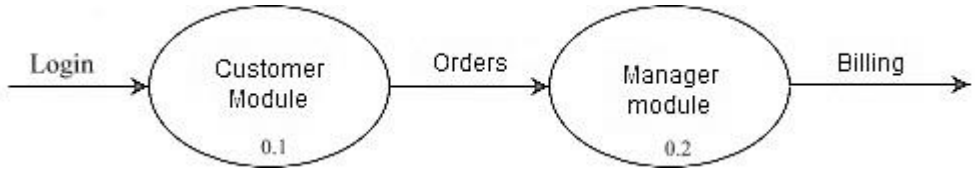


Fig.4.3 Level 1 DFD

Level 2 DFD:

This contains two modules:

Customer module

All customer interactions are done in this module. When the customer login to the application he/she can get the product list and description. Then they could place orders and confirm orders. The bill will be generated and payment procedure is carried out.

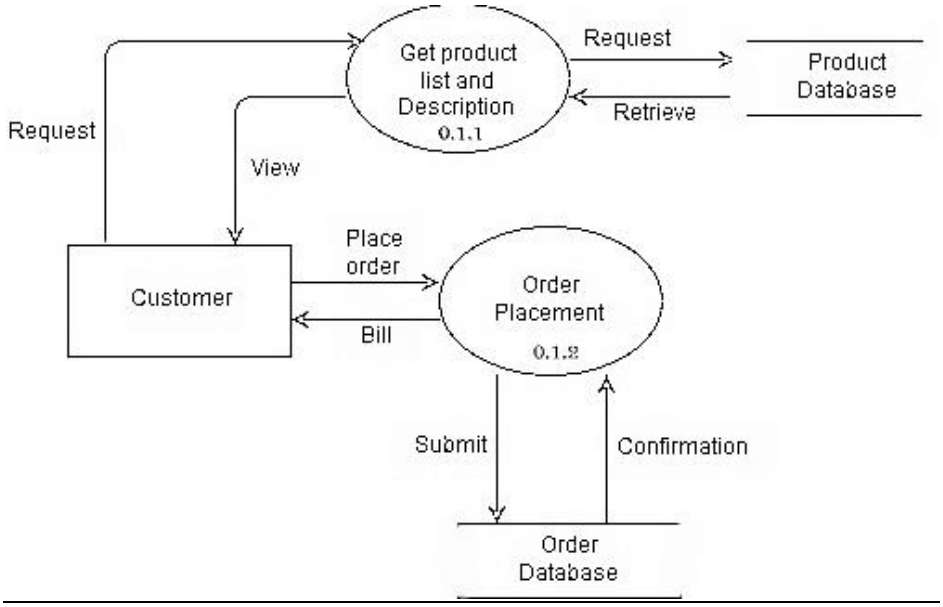


Fig.4.4 Level 2 DFD(Customer Module)

Manager Module

Manager can interact with this module. He can edit/delete/update menu as and when required. It is the manager who informs the chef to deliver the orders.

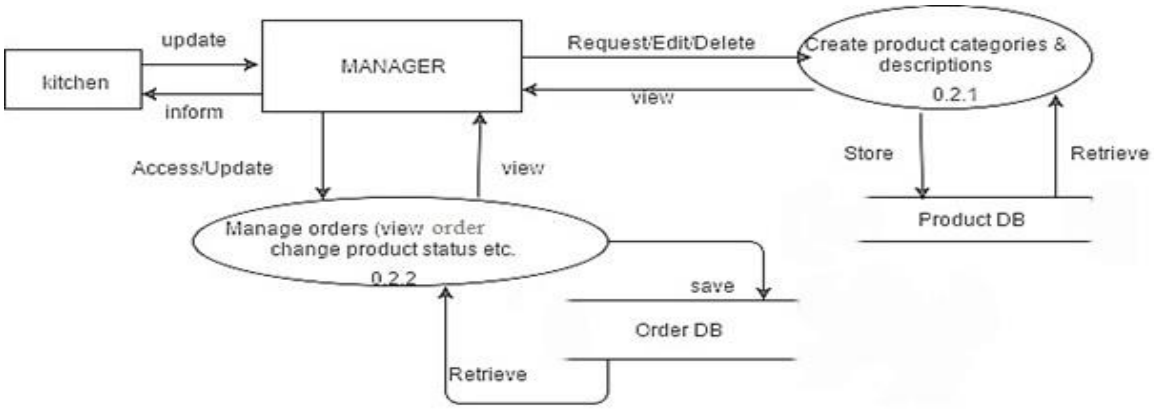


Fig.4.5 Level 2 DFD (Manager Module)

4.3 ENTITY - RELATIONSHIP DIAGRAM

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

The E-R Diagram for an automated food ordering system is shown in Fig. 4.6. There are six main entities – customer, order, item, bill, offer and feedback.

The main attributes in the entity customer are - table_no, user_name, password. Each customer has a unique user_name.
The attributes in the entity order are - table_no, item_no, item_name, quantity, user_name. Each order has unique table_no, item_no and a user_name.
The attributes in the entity item are - item_no, item_name, quantity and price. Each item has a unique item _no.
The attributes in the entity bill are - table_no, user_name and bill_amount.The user_name is unique here.

The attributes in the entity offer are - user_name, total_price among which user_name is the unique one.

Finally the entity feedback has - user_name and item_no and review & rating. Here user_name and item_no are unique.

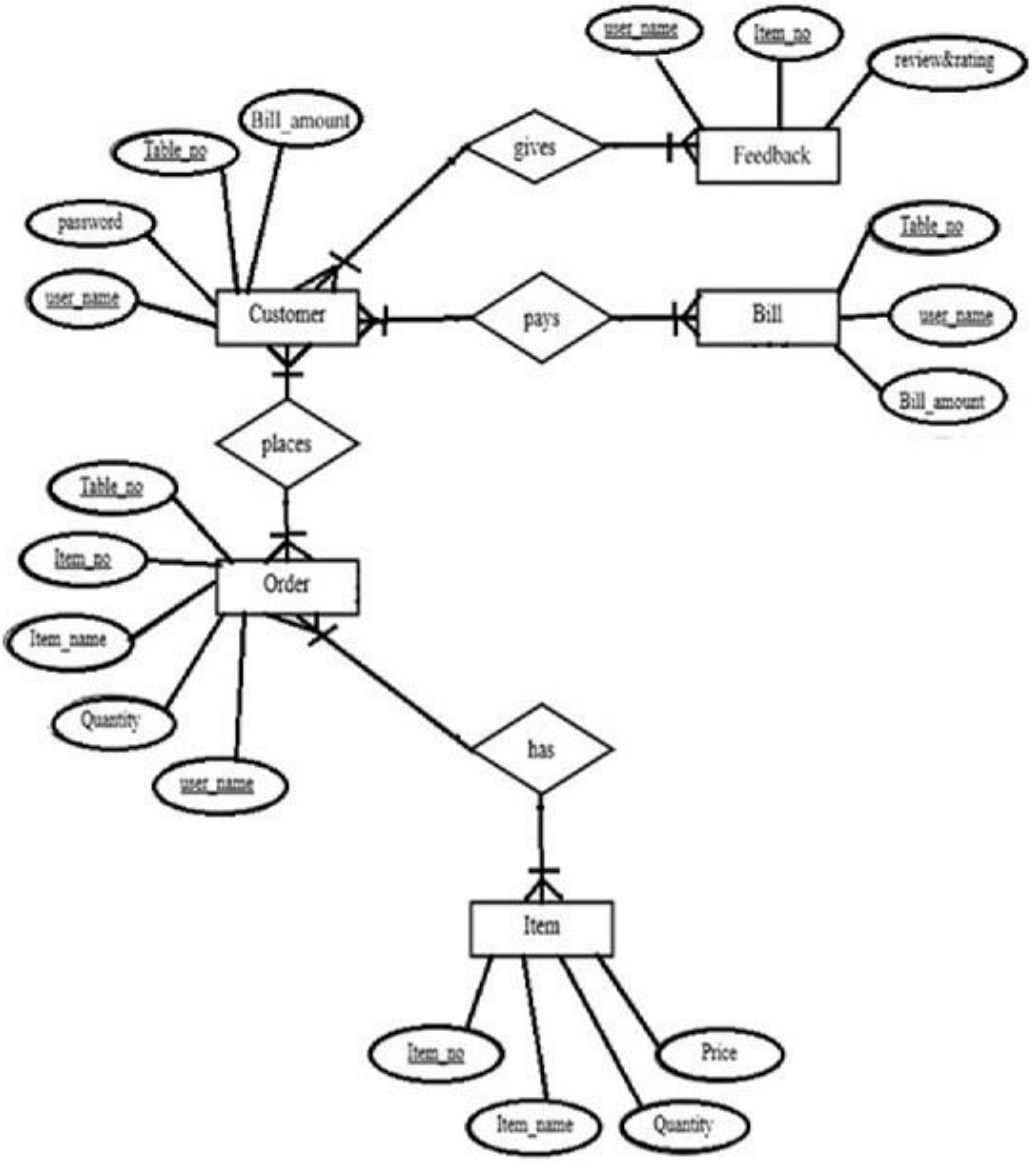


Fig.4.6 E-R Diagram for Automated Food Ordering System

4.4 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The UML class diagram for an automated food ordering system is shown in Fig.4.7.

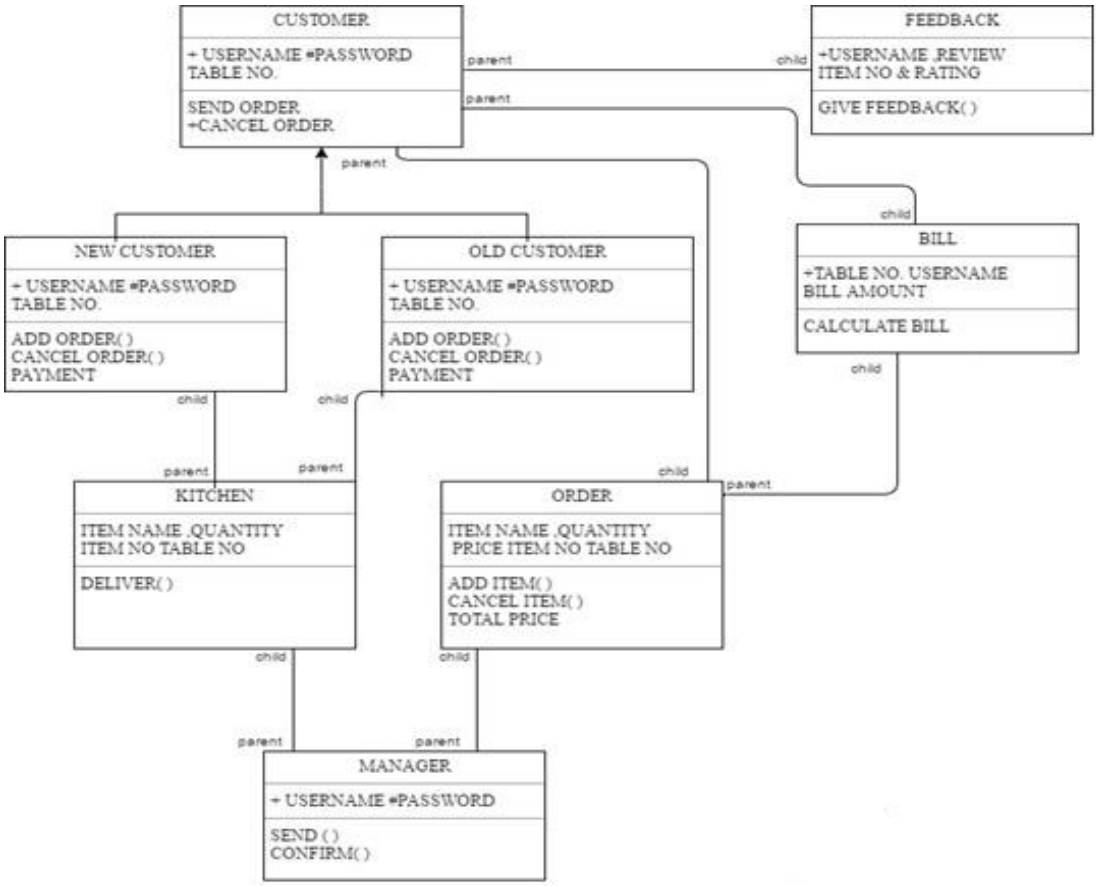


Fig.4.7 UML Class Diagram for Automated Food Ordering System

4.5 USE CASE DIAGRAM

Use Case Diagrams. Use case diagrams are usually referred to as behaviour diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors)

The Use Case Diagram for the project is shown in Fig. 4.8. This shows the various use cases that are involved in the program. A quick description of all the use cases are as follows:

Login: Allows to create new account for customer, edit and delete the existing account.
Order food: With the help of the automated menu the customer orders food.
Billing: The cash payment process is done in this case.
Cook food: This case represents the process of kitchen where chef prepares the ordered food.
Serve: The food is served by the waiter according to table number displayed on kitchen screen.
Relationships in Use Case Diagrams
Customer acts as an actor where he/she logins and order the food. The order is then send to the next actor, manager and he confirms the order and sends it to chef. The chef, who is the next actor prepares the food. Then the waiter serves the food and after having food customer pays the bill which will be confirmed by the manager.

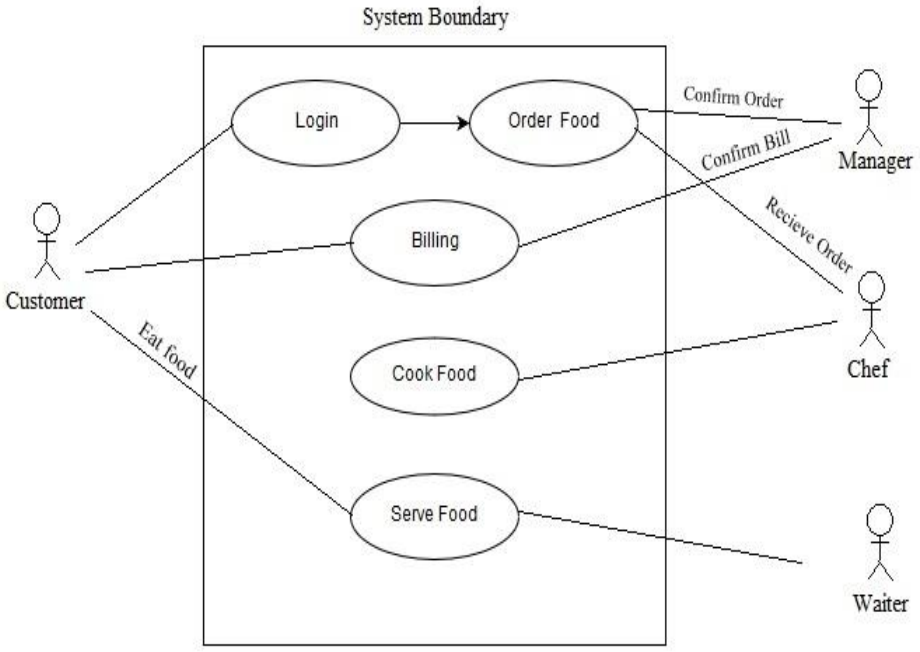


Fig.4.8 Use Case Diagram for Automated Food Ordering System

4.6 SEQUENCE DIAGRAM

It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

The sequence diagram for the project is shown in Fig. 4.9. When the customer orders items and confirms his order the manager gets its details. He then sends this order to the kitchen and the chef sends a response if the order is delivered. Confirmation message from kitchen is send to manager and the food is delivered to the customer. The manager generates the bill according to the customer's click for bill and the payment is done.

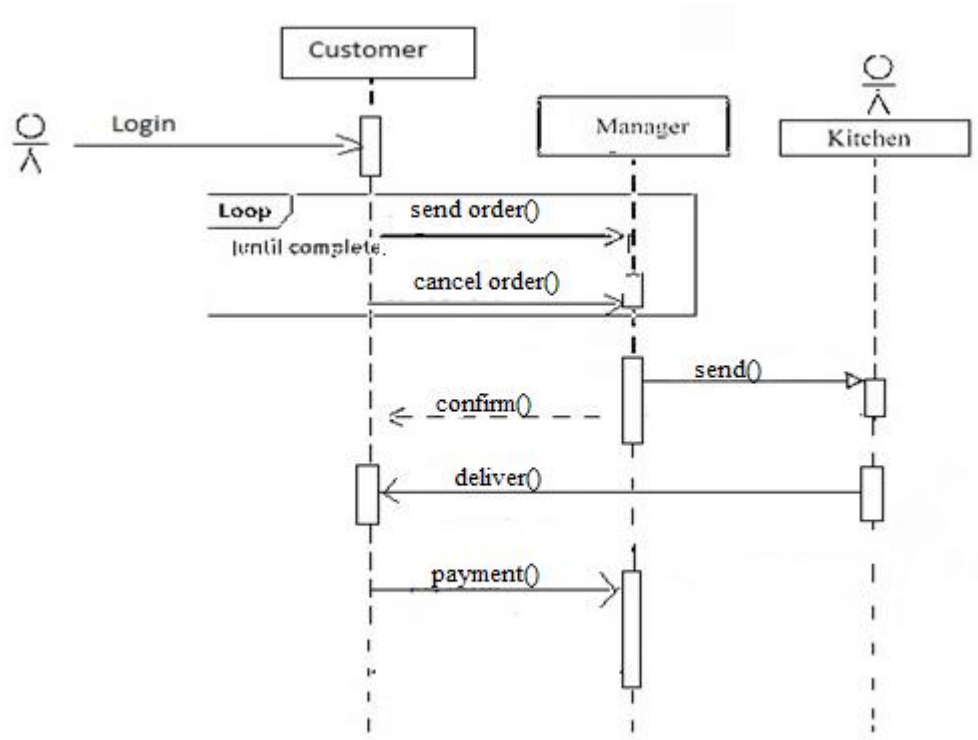


Fig.4.9 Sequence Diagram for Automated Food Ordering System

5. IMPLEMENTATION

5.1 INTRODUCTION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The project implements PHP, MySQL, CSS and standard HTML. The project will be capable of running on standard internet web browsers.

5.1.1 Module Implementation

The project has two web applications - one for the manager and the other one for the customers. The manager web app helps the manager to edit/delete/update menu as and when required. It is the manager who informs the chef to deliver the orders. The customer web app helps the customer to customer login to the application, he/she can get the product list and description. Then they could place orders and confirm orders.

The project is mainly divided into three modules – two Web modules one for the manager and other for the customer and a display module at the kitchen.

5.2 MANAGER WEB MODULE

5.2.1 Login Module

The manager is provided with a username and password and by entering it he can access the centralized database of the restaurant.

The login page is developed in HTML / PHP as shown in the screenshot S.1. After he is entered the username or email id and password he can click login button. It is then checked against the saved username and password in the database table (Table 5.1). If it is found to be mismatching then an error message will be printed. Otherwise he will be directed to the view page.

Main functions used in this module are:

1.mysqli_connect(<servername>,<username>,<password>)

To create a connection to the database, mysql_connect() function is used.

2.stripslashes($_REQUEST['Username'])

3.stripslashes($_REQUEST['Password'])

Both are use to remove the backslashes.

4.mysqli_real_escape_string($con,$username)

5. mysqli_real_escape_string($con,$password)

It escapes special characters in a string.

6. mysqli_query("<sql statement>")

To execute the mysql queries in PHP, mysql_query() function is used.

7. mysqli_query($con,$query)

It performs a query against the database and display the result.

8. mysql_error()

To show the error result we use the above function.

9. mysqli_num_rows($result)

Retrieves the contents of cells from a MySQL result set.

SELECT * FROM `login` WHERE username='$username' and password='$password'

Here we select the username and password from the table 'login' for a particular username and password from the database 'manager'.

Username	Password

Table 5.1 Login Table

5.2.2 View Update Module

When the manager wants to see the ordered items from a particular table he can see it in this page.

The page is developed in HTML as shown in screenshot S.11.

The main functions used in this module are:

1. define('DB_HOST', 'localhost')

It defines the MySQL hostname, which is one parameter of connection

2. define('DB_NAME', 'manager')

To define the name of the database for wordpress, this is one parameter of connection

3.define('DB_USER','root')

To define the MySQL database username, this is one parameter of connection

4. define('DB_PASSWORD','')

It is used to define MySQL database password, this is one parameter of connection

5. mysqli_connect(DB_HOST,DB_USER,DB_PASSWORD)

It is used to connect to MySQL and select the database.

6. mysql_error()

To show the error result we use the above function

7. mysqli_select_db($con,DB_NAME)

The mysql_select_db() function sets the current active database on the server which is associated with the specified connection.

8. isset($_POST['update'])

The **isset** () **function** is used to check whether a variable is set or not

9. mysqli_query($con,$update_query)

It performs a query against the database and display the result.

10. mysqli_fetch_array($myData)

The **mysqli_fetch_array**() function fetches a **result** row as an associative array, a numeric array, or both.

11. mysqli_close($con)

The **mysql_close**() function is used to close an open MySQL connection.

UPDATE menu

SETitemno='$_POST[itemno]',itemname='$_POST[itemname]',quantity='$_POST[quantity]',price='$_POST[price]' WHERE itemno='$_POST[hidden]'

Here we updates the menu table in the database by setting the itemno=$_POST[itemno], itemname=$_POST[itemname], quantity=$_POST[quantity], price=$_POST[price]

SELECT * FROM menu ORDER BY itemno ASC

Here we displays items from the table menu ordered by its item no, which is the updated menu.

5.2.3 View Table Module

Here the manager can see the orders from different tables. The page is developed in HTML / PHP as shown in screenshot S.12 The manager can also send the orders as shown in screenshots S.13

The main functions used in this module are:

1. session_start()

Creates a session or resumes the current one based on a session identifier passed via a GET or POST request, or passed via a cookie. When **session_start**() is called or when a session auto starts, PHP will call the open and read session save handlers.

2.isset()

The **isset** () **function** is used to check whether a variable is set or not.

3.mysqli_connect("localhost", "root", "", "items")

Attempt MySQL server connection .Assuming that you are running MySQL server with default setting (user 'root' with no password)

4.array_column($_SESSION["shopping_cart"], "item_id")

Returns value in the input array.

5.array(

'item_id' => $_GET["id"],

'item_name' => $_POST["hidden_name"],

'item_price' => $_POST["hidden_price"],

'item_quantity' => $_POST["quantity"]

)

Returns values from the specified columns in the input array.

6. mysqli_query($connect, $query)

It performs a query against the database and displays the result.

7. mysqli_num_rows($result)

Retrieves the contents of cells from a MySQL result set.

8. mysqli_fetch_array($result)

The **mysqli_fetch_array**() function fetches a **result** row as an associative array, a numeric array, or both.

9. unset($_SESSION["shopping_cart"][$keys])

The **unset**() **function** destroys a given variable.

10. in_array($_GET["id"], $item_array_id)

The **in_array**() **function** searches an array for a specific value.

11. count($_SESSION["shopping_cart"])

The **count**() **function** is used to **count** the elements of an array or the properties of an object.

12.alert()

An **alert** dialog box is mostly used to give a warning message to the users.

5.3 CUSTOMER WEB MODULE

5.3.1 Login Module

The customer is provided with a welcome page where he is provided with the restaurant's name "BAY VIEW" and the options for knowing details of the restaurant, login, call waiter and contact are provided.

The page is developed in HTML / PHP as shown in screenshot S.5. By clicking about he will be provided with details of BAY VIEW. By clicking the log in menu he will be directed to the login page for customers, which is developed in HTML / PHP as shown in screenshot S.6, where he can see both sign up and log in options. If he is a return customer then he can enter the username and password provided in the same and can order items he need. Otherwise if he is a new customer then he can click sign up and he will be directed to a new page, ie, sign up. He can also view the details about the restaurant as shown in screenshot S.7.

The main functions used in this module are

1.stripslashes($_REQUEST['Username'])

2.stripslashes($_REQUEST['Password'])

Both are used to remove the backslashes these functions are used.

3.mysqli_real_escape_string($con,$username)

4. mysqli_real_escape_string($con,$password)

To escape special characters in a string we use these functions.

5.mysqli_query($con,$query)

It performs a query against the database and displays the result.

6. mysql_error ()

To show the error result we use the above function

7. mysqli_num_rows($result)

Retrieves the contents of cells from a MySQL result set.

SELECT * FROM `login` WHERE username='$username' and password='$password'

Here we select the username and password from the table ‘User’ for a particular username and password from the database authentication

Username	Password

Table 5.2 Users table

5.3.2 Sign Up Module

When a customer is new to the restaurant he will be using this page. The page is developed in HTML / PHP as shown in screenshot S.8.

In the sign up page the customer can see text fields for entering his desired username and password and since he is new to the restaurant he need to confirm the password , provide his Email id and mobile number. By entering all these he can sign up.

The main functions used in this module are:

1.define('DB_HOST', 'local host')

It defines the MySQL hostname, which is one parameter of connection

2. define('DB_PASSWORD','')

It is used to define MySQL database password, which is one parameter of connection

3.define('DB_NAME', 'authentication')

To define the name of the database for wordpress, this is one parameter of connection

4.define('DB_USER','root')

To define the MySQL database username, this is one parameter of connection

5. mysqli_connect(DB_HOST,DB_USER,DB_PASSWORD,DB_NAME)

It is used to connect to MySQL and select the database.

Once he is finished with the sign up procedures he will be again directed to the login page where he has to follow the login procedures as described in the login module.

5.3.3 Menu Module

After finishing login he will be displayed with the menu page of the restaurant. The page is developed in HTML / PHP as shown in the screenshot S.2. When customer clicks 'GET STARTED' the page will scroll down and will show the dishes page as shown in screenshot S.4. When he hovers through INDIAN the screen will be as shown in screenshot S.5 and when he clicks on it he will be directed to INDIAN dishes page as shown in screenshot S.6. This is applicable to all other dishes. He can confirm his order for bill as shown in screenshot S.7.

The main functions used in this module are:

1. session_start()

Creates a session or resumes the current one based on a session identifier passed via a GET or POST request, or passed via a cookie. When **session_start**() is called or when a session auto starts, PHP will call the open and read session save handlers.

2.isset()

The **isset** () **function** is used to check whether a variable is set or not

3.mysqli_connect("localhost", "root", "", "items")

Attempt MySQL server connection .Assuming that you are running MySQL server with default setting (user 'root' with no password)

4.array_column($_SESSION["shopping_cart"], "item_id")

Returns value in the input array.

5.array(

'item_id' => $_GET["id"],

'item_name' => $_POST["hidden_name"],

'item_price' => $_POST["hidden_price"],

'item_quantity' => $_POST["quantity"]

)

Returns values from the specified columns in the input array.

6. mysqli_query($connect, $query)

It performs a query against the database and displays the result.

7. mysqli_num_rows($result)

Retrieves the contents of cells from a MySQL result set.

8. mysqli_fetch_array($result)

The **mysqli_fetch_array**() function fetches a **result** row as an associative array, a numeric array, or both.

9. unset($_SESSION["shopping_cart"][$keys])

The **unset**() **function** destroys a given variable.

10. in_array($_GET["id"], $item_array_id)

The **in_array**() **function** searches an array for a specific value.

11. count($_SESSION["shopping_cart"])

The **count**() **function** is used to **count** the elements of an array or the properties of an object.

12.alert()

An **alert** dialog box is mostly used to give a warning message to the users.

"SELECT * FROM dessert ORDER BY itemno ASC"

Here we select the items from the database table item for the dish Dessert.

Item no.	Item name	Image	Price

Table 5.3 Dessert table

"SELECT * FROM Chinese ORDER BY itemno ASC"

Here we select the items from the database table item for the dish Chinese.

Item no.	Item name	Image	Price

Table 5.4Chinese table

"SELECT * FROM drinks ORDER BY itemno ASC"

Here we select the items from the database table item for the dish Drinks.

Item no.	Item name	Image	Price

Table 5.5 Drinks table

"SELECT * FROM Indian ORDER BY itemno ASC"

Here we select the items from the database table item for the dish Indian.

Item no.	Item name	Image	Price

Table 5.6 Indian table

5.3.4 Review Module

Before the customer is getting out from the restaurant there is a chance for him to give his valuable feedback. It can be given at the Review page. The Review page is developed in HTML / PHP as shown in screenshot S.15.

The main functions used in this module are :

1. define('DB_HOST', 'localhost')

It defines the MySQL hostname, which is one parameter of connection

2. define('DB_NAME', 'reviewsystem')

To define the name of the database for wordpress, this is one parameter of connection

3. define('DB_USER','root')

4. define('DB_PASSWORD','')

It is used to define MySQL database password, which is one parameter of connection

5. mysqli_connect(DB_HOST,DB_USER,DB_PASSWORD,DB_NAME)

It is used to connect to MySQL and select the database.

6. mysql_error()

To show the error result we use the above function

7. isset()

The **isset** () **function** is used to check whether a variable is set or not

8. mysqli_query ($con,$query)

It performs a query against the database and displays the result.

"INSERT INTO review (pname,rating,review) VALUES ('$name', '$rating', '$comment')"

The customer can put his review about the hotel by using this.

5.4 Kitchen Display Module

Here the chef can see the ordered items from different tables. The page is developed in HTML / PHP as shown in screenshot S.14.
The main functions used in this module are:

1.session_start()

Creates a session or resumes the current one based on a session identifier passed via a GET or POST request, or passed via a cookie. When **session_start**() is called or when a session auto starts, PHP will call the open and read session save handlers.

2.isset()

The **isset** () **function** is used to check whether a variable is set or not

3.mysqli_connect("localhost", "root", "", "items")

Attempt MySQL server connection .Assuming that you are running MySQL server with default setting (user 'root' with no password)

4.array_column($_SESSION["shopping_cart"], "item_id")

Returns value in the input array.

5.array(

'item_id' => $_GET["id"],

'item_name' => $_POST["hidden_name"],

'item_price' => $_POST["hidden_price"],

'item_quantity' => $_POST["quantity"]

)

Returns values from the specified columns in the input array.

6. mysqli_query($connect, $query)

It performs a query against the database and displays the result.

7. mysqli_num_rows($result)

Retrieves the contents of cells from a MySQL result set.

8. mysqli_fetch_array($result)

The **mysqli_fetch_array**() function fetches a **result** row as an associative array, a numeric array, or both.

9. unset($_SESSION["shopping_cart"][$keys])

The **unset**() **function** destroys a given variable.

10. in_array($_GET["id"], $item_array_id)

The **in_array**() **function** searches an array for a specific value.

11. count($_SESSION["shopping_cart"])

The **count**() **function** is used to **count** the elements of an array or the properties of an object.

12.alert()

An **alert** dialog box is mostly used to give a warning message to the users.

6. TESTING

6.1 INTRODUCTION

Testing is to confirm that the software product works in conformance with the specified requirements. A test approach tells about the test strategy implementation of a project and defines how testing would be carried out. The approach might be Proactive where test design process is initiated as early as possible in order to find and fix the defects before the build is created or Reactive where the testing is not started until after design and coding are completed. The project uses proactive approach.

6.2 TESTING METHODOLOGY

6.2.1 TEST APPROACH

This project uses a Dynamic testing approach. The dynamic testing includes techniques - Unit testing, Integration testing and System testing. Under unit testing, individual units or modules is tested by the developers which involves testing of source code. In integration testing, individual modules are grouped together and tested by the developers. Here, the purpose is to determine that modules are working as expected once they are integrated. System testing is performed on the whole system by checking whether the system or application meets the requirement specification document.

Further, it uses Web testing for the web modules. Web Testing in checking your web application for potential bugs before its made live or before code is moved into the production environment. During this stage issues such as that of web application security, the functioning of the site, its access to users and its ability to handle traffic is checked. In includes Functionality testing, Usability testing, Interface testing, Database testing, Compatibility testing, Crowd testing, etc.

It also uses Graphical User Interface (GUI) testing for the web app modules. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - tool bar, menu bar, dialog boxes and windows etc. The project uses Manual-based testing for doing this.

6.2.2 UNIT TESTING

Unit testing in the project uses mainly **Functionality testing**, **Database testing** and **GUI testing**.

Functionality testing is used to check all links in the web pages are working correctly and to make sure there are no broken links especially those in the navigation bar and also to check test forms working as expected

Database is one critical component of the project. Database testing here checks whether data integrity is maintained while creating, updating or deleting data in database.

GUI testing check all the GUI elements especially for app, the size, position, width, length and acceptance of characters or numbers and sees whether the error messages are displayed, etc.

Test cases for unit testing includes Table 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 6.8, 6.9, 6.10, 6.11, 6.13,6.14,6.15 and 6.16.

The Test Cases for unit testing are as follows:

Test case ID : AFO_1.1
Test Case : Manager-Login
Module Name : Manager Web Module
Test Title : Verify Manager login with valid username and password
Description : Test the Manager login page
Pre-Conditions : Manager has valid username and password

Test Steps	Test Data	Expected Output	Actual Output
Manager enters a valid username and password	username = anu password = anu	Redirects to view page	Redirected to view page
Manager enters a wrong username or password	username = anu password = anu	Alert as “Invalid username or password”	Alert as “Invalid username or password”

Table 6.1 Test case for Manager login with username and password

Test case ID : AFO_1.2
Test Case : Manager-view table
Module Name : Manager Web Module
Test Title : Verify Manager view table page
Description : Test the Manager view table page
Pre-Conditions : First time successful direction to current page.

Test Steps	Test Data	Expected Output	Actual Output
Manager clicks view Link	Manager Clicks	Redirects to view of order from the clicked table no.	Redirects to view of order from the clicked table no.

Table 6.2 Test case for Manager view

Test case ID : AFO_1.3
Test Case : Manager – navigation bar
Module Name : Manager Web Module
Test Title : Verify Manager navigation bar
Description : Test the Manager confirm order
Pre-Conditions : First time successful direction to Manager view

Test Steps	Test Data	Expected Output	Actual Output
Manager clicks on the navigation bar to go to desired page	Manager click	Redirects to corresponding page	Redirected to corresponding page

Table 6.3 Test case for manager navigation bar

Test case ID : AFO_1.4

Test Case : Manager-view update

Module Name : Manager Web Module

Test Title : Verify Manager view update page

Description : Test the Manager view update page

Pre-Conditions : First time successful direction to Manager view

Test Steps	Test Data	Expected Output	Actual Output
Manager clicks view update	Manager click	Redirects to view update page.	Redirected to view update page.

Table 6.4 Test case for Manager view update

Test case ID : AFO_1.5

Test Case : Manager-send order

Module Name : Manager Web Module

Test Title : Verify Manager send order

Description : Test the Manager send order

Pre-Conditions : First time successful direction to Manager view update

Test Steps	Test Data	Expected Output	Actual Output
Manager clicks send button	Manager click	Redirects to kitchen display	Redirected to kitchen display

Table 6.5 Test case for Manager send order

Test case ID : AFO_1.6

Test Case : Customer – sign up

Module Name : Customer Web Module

Test Title : Verify Customer sign up page

Description : Test the Customer sign up page

Pre-Conditions : Customer has username, password, Email, Mobile Number

Test Steps	Test Data	Expected Output	Actual Output
Customer enters username, password, confirm password, Email and Mobile number	Username = srikant Password = srikant Confirm password = srikant,Email = srikant@gmail.com Mobile no.=1234567890	Redirects to login page	Redirected to login page

Table 6.6 Test case for Customer sign up

Test case ID **:** AFO_1.7

Test Case **:** Customer -login

Module Name **:** Customer Web Module

Test Title **:** Verify Customer login page

Description **:** Test the Customer login page

Pre-Conditions **:** Customer has a valid username and password

Test Steps	Test Data	Expected Output	Actual Output
Customer enters a valid username and password	username = srikant password = srikant	Redirects to menu page	Redirected to menu page
Customer enters a wrong username or password	username = sss password = srt	Alert as “Invalid username or password”	Alert as “Invalid username or password”

Table 6.7Test case for Customer login

Test case ID **:** AFO_1.8

Test Case **:** Customer – menu

Module Name **:** Customer Web Module

Test Title **:** Verify Customer menu page

Description **:** Test the Customer menu page

Pre-Conditions **:** First time successful Customer login

Test Steps	Test Data	Expected Output	Actual Output
Customer hovers through the menu page	Customer click	Redirects to corresponding dish page	Redirected to corresponding dish page

Table 6.8 Test case for Customer menu

Test case ID **:** AFO_1.9

Test Case **:** Customer – order

Module Name **:** Customer Web Module

Test Title **:** Verify Customer order page

Description **:** Test the Customer order page

Pre-Conditions **:** First time successful direction to customer view page

Test Steps	Test Data	Expected Output	Actual Output
Customer clicks on the order button	Customer click	Item gets added to ordered list	Item got added to ordered list

Table 6.9 Test case for Customer order

Test case ID **:** AFO_1.10
Test Case **:** Customer – order list
Module Name **:** Customer Web Module
Test Title **:** Verify Customer order list
Description **:** Test the Customer order list
Pre-Conditions **:** First time successful direction to customer order page

Test Steps	Test Data	Expected Output	Actual Output
Customer clicks on view order	Customer click	Redirects to order list page	Redirected to order list page

Table 6.10 Test case for Customer order list

Test case ID **:** AFO_1.11
Test Case **:** Customer – navigation bar
Module Name **:** Customer Web Module
Test Title **:** Verify Customer navigation bar
Description **:** Test the Customer navigation bar
Pre-Conditions **:** First time successful Customer login

Test Steps	Test Data	Expected Output	Actual Output
Customer clicks on the navigation bar to go to desired page	Customer click	Redirects to corresponding page	Redirected to corresponding page

Table 6.11 Test case for Customer-navigation bar

Test case ID : AFO_1.12
Test Case : Offer menu
Module Name : Customer Web Module
Test Title : Verify Offer menu
Description : Test the Offer menu
Pre-Conditions : First time successful Customer login

Test Steps	Test Data	Expected Output	Actual Output
Customer clicks on the offer option in the navigation bar	Customer click	Redirects to offer page	Redirected to offer page

Table 6.12 Test case for offer page

Test case ID : AFO_1.13
Test Case : About page
Module Name : Customer Web Module
Test Title : Verify About page
Description : Test the About page
Pre-Conditions : First time successful Customer login

Test Steps	Test Data	Expected Output	Actual Output
Customer clicks on the About on navigation bar	Customer click	Redirects to about page	Redirected to About page

Table 6.13 Test case for About page

Test case ID **:** AFO_1.14
Test Case **:** Contact page
Module Name **:** Customer Web Module
Test Title **:** Verify Contact page
Description **:** Test the Contact page
Pre-Conditions **:** First time successful Customer login

Test Steps	Test Data	Expected Output	Actual Output
Customer clicks on the Contact on navigation bar	Customer click	Redirects to Contact page	Redirected to Contact page

Table 6.14 Test case for Contact page

Test case ID **:** AFO_1.15
Test Case **:** Customer - review
Module Name : Customer web module
Test Title **:** Verify Customer - review
Description **:** Test the Customer - review page
Pre-Conditions **:** First time successful Customer login

Test Steps	Test Data	Expected Output	Actual Output
Customer click on review button	Customer click	Redirects to review page	Redirected to review page

Table 6.15 Test case for Review page

Test case ID : AFO_1.16
Test Case : Kitchen – display
Module Name : Kitchen display Module
Test Title : Verify Kitchen display page
Description : Test the Kitchen display page
Pre-Conditions : First time successful send from Manager view

Test Steps	Test Data	Expected Output	Actual Output
Manager click on send button	Manager click	Redirects to kitchen display	Redirected to kitchen display

Table 6.16 Test case for Kitchen display

6.2.3 INTEGRATION TESTING

Integration testing in the project uses mainly Functionality testing, Usability testing, Interface testing, Database testing, Compatibility testing and GUI testing.

Functionality testing here includes checking the forms for whether the default values are being populated and once submitted, the data in the form is submitted to a live database.

Usability testing is a vital part of any web based project. It here includes testing site navigation and content for usability.

Interface Testing has three areas of testing –

 App – Test requests are sent correctly to the Database and output at the client side is displayed correctly.

 Web Server - Test web server is handling all application requests without any service denial.

 Database Server - Make sure queries sent to the database give expected results.

Database testing in integration testing checks whether the data retrieved from the database is shown accurately.

Compatibility testing ensures that the app and website functions correctly across different devices and browsers respectively.

GUI testing is again used here in integration testing to check the user interface of the web application to ensure their proper functioning.

Test case for integration testing includes Table 6.17, Table 6.18, 6.19, 6.20 and Table 6.21.

The Test Cases for integration testing are as follows.

Test Case ID **:** AFO_2.1

Test Case **:** Manager - login

Test Title **:** Login

Description **:** Test Login

Pre-Conditions **:** Manager should know his/her user credentials

Test Steps	Test Data	Expected Output	Actual Output
Manager enters his username and password	Username = anu Password = anu	Verifies the information entered with the data stored in the database and redirects to manager view page	The information is verified using the data stored in the database and the manager is redirected to manager view page

Table 6.17 Test case for Manager login

Test Case ID **:** AFO_2.2

Test Case **:** Manager - view

Test Title **:** Manager view

Description **:** Test Manager view

Pre-Conditions **:** Manager should be successfully logged in

Test Steps	Test Data	Expected Output	Actual Output
Manager views orders from different tables sends it to kitchen and confirms customer orders	Manager hover through table nos. and checks orders from a particular table and sends to kitchen, confirms order	The order gets confirmed	The order got confirmed

Table 6.18 Test case for Manager view

Test Case ID **:** AFO_2.3
Test Case **:**Customer - login
Test Title **:** Login
Description **:** Test Login
Pre-Conditions **:** Customer should know his/her user credentials

Test Steps	Test Data	Expected Output	Actual Output
Customer enters his username and password	Username =srikant Password = srikant	Verifies the information entered with the data stored in the database and redirects to customer order page	The information is verified using the data stored in the database and the customer is redirected to customer page

Table 6.19 Test case for Customer- login

Test Case ID : AFO_2.4
Test Case : Customer - view
Test Title : Customer view
Description : Test Customer view
Pre-Conditions : Preinstalled customer web module

Test Steps	Test Data	Expected Output	Actual Output
Customer goes through the different options provided in the navigation bar and dishes	Customer hover and Customer click	Redirects to the desired pages and does the tasks	Redirected to desired pages and did the tasks

Table 6.20 Test case for Customer- view

Test Case ID : AFO_2.5
Test Case : Kitchen - display
Test Title : Kitchen display
Description : Test Kitchen display
Pre-Conditions : Preinstalled display set up

Test Steps	Test Data	Expected Output	Actual Output
Manager click on send button	Manager click	Redirects to kitchen display	Redirected to kitchen display

Table 6.21 Test case for kitchen display

7 CONCLUSION

We have presented a web application on Automated Food Ordering. Our system offers an web based platform for customers to order and review the food items of the restaurant. The manager's web application helps managers to view the day to day activities of the restaurant and he can ensure the proper working. We strongly believe our system can bring drastic changes and enhance the communication between customers and managers.

The system is found to work efficiently and effectively and delivers the expected outcomes. New features can be added when need arises with slight modification on the app and website, which is relatively easy.

APPENDIX A : USER MANUAL

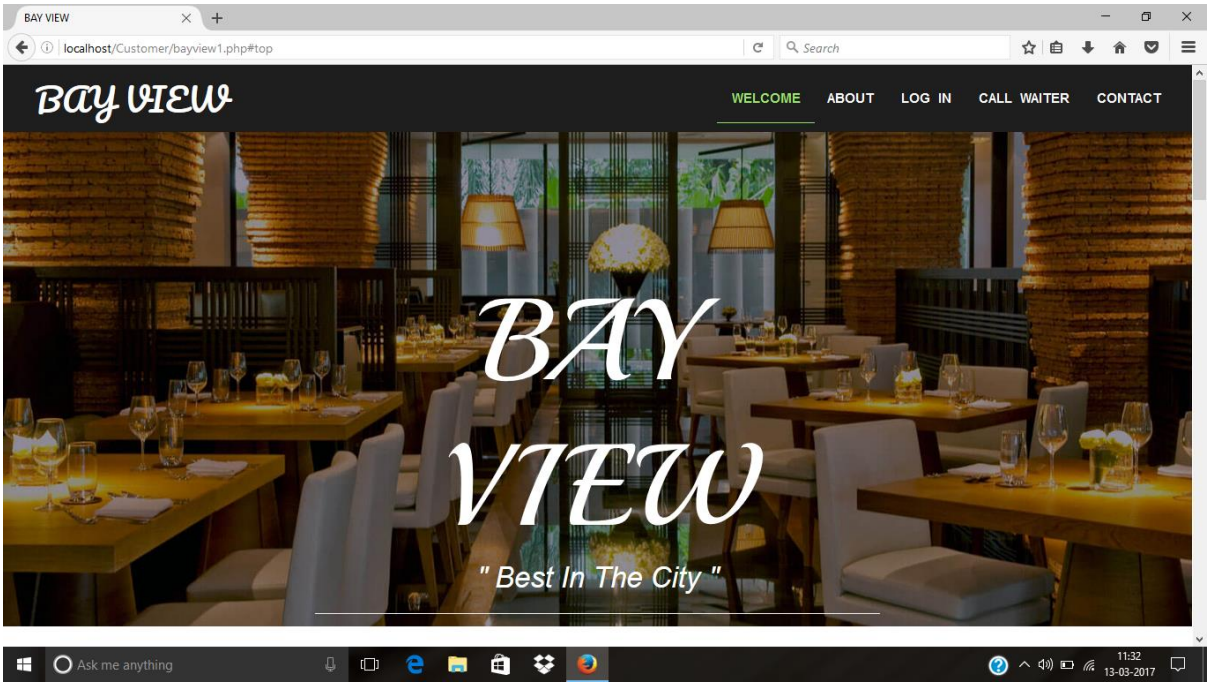
Customer, manager and kitchen have the access to the webapp. Each of them has its own module for customers webapp is installed in each tablets of their respective tables. There is one manager accessing his module and a kitchen display to view the order and to transmit it. They have to use on the browser of a computer having an internet connection.
Initially customers would see a welcome page as shown in (Screenshot S.1) further we see on navigation bar there are options such as about, login and contact as shown in (Screenshot S.1) For the first time customer should sign up with basic details and then login shown in (Screenshot S.3) and (Screenshot S.2).This creates an account for each customers and with that a unique id and password.

After logging in customer with a view page of main menu appears which widely categorize the hotels menu (Screenshots S.4,S.5).Customer need to select any one category and then all items would be displayed for example Indian if selected the view that we see is shown in (Screenshot S.6).The amount, number of quantity to order and an order button is there to place the order this is shown in (Screenshot S.6).

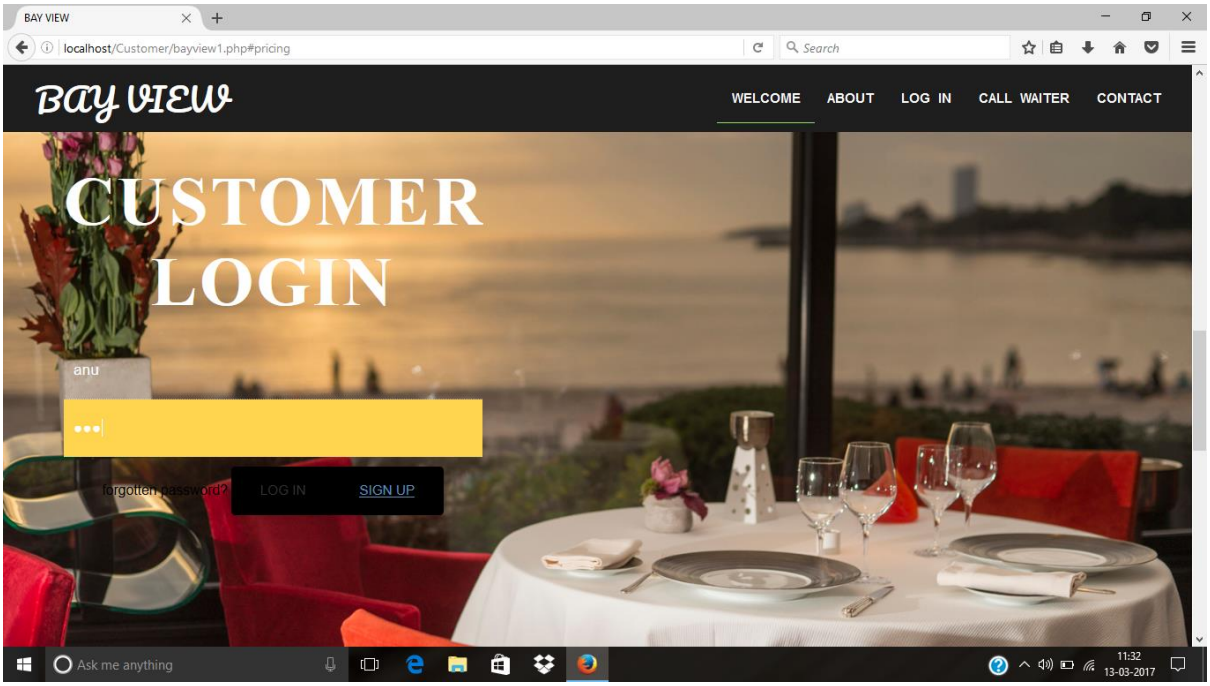
On the navigation bar there is a review option where we can review individual items accordingly.

Manager module include a login page (Screenshot S.10).The hotel manager can access the menu update as shown in (Screenshot S.11).He can view all orders that happens in each specific tables.

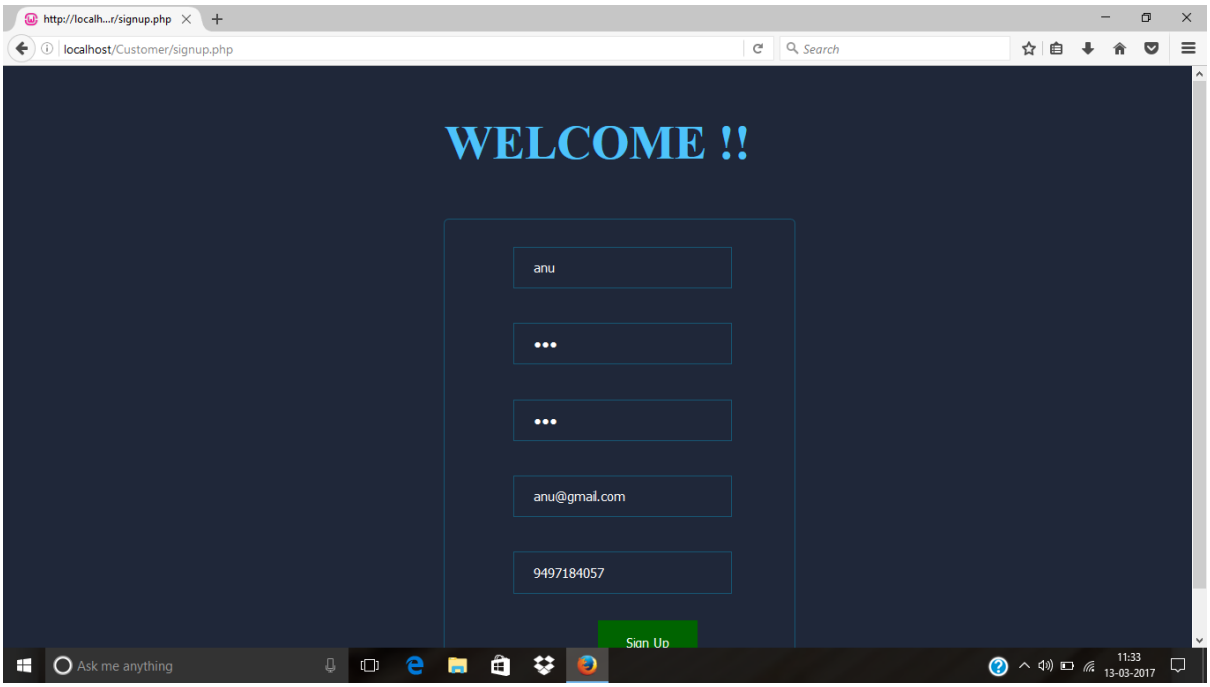
APPENDIX B: SCREENSHOTS



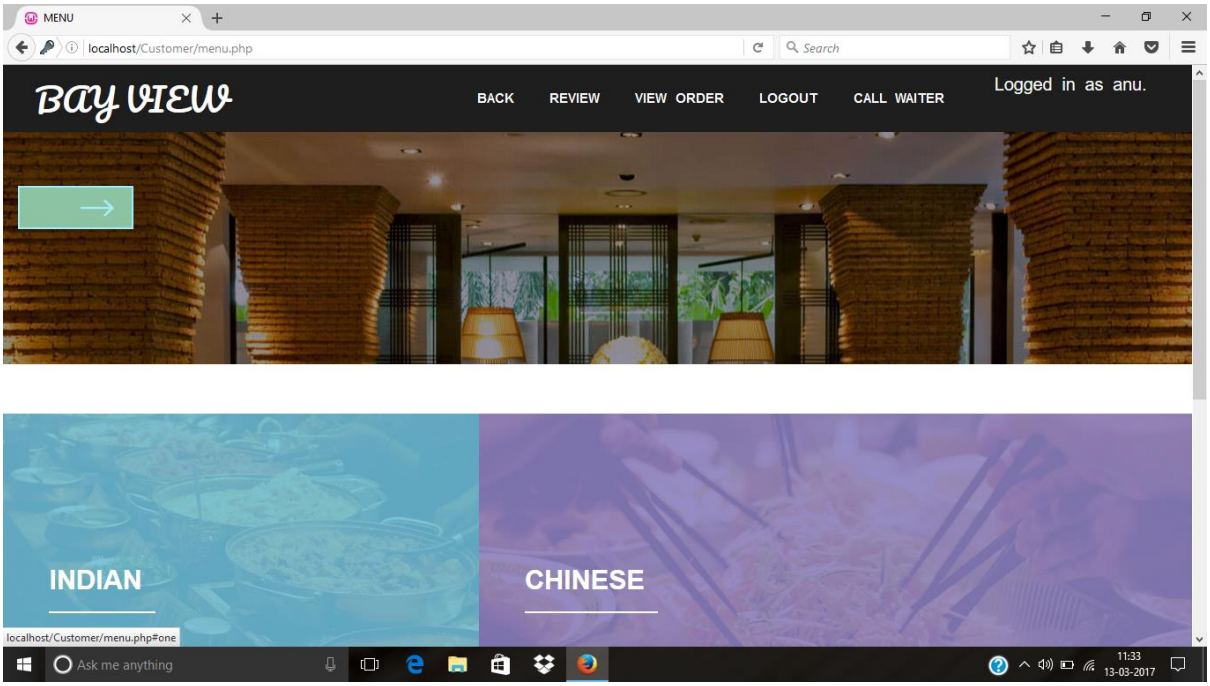
Screenshot S.1 Customer Welcome Page



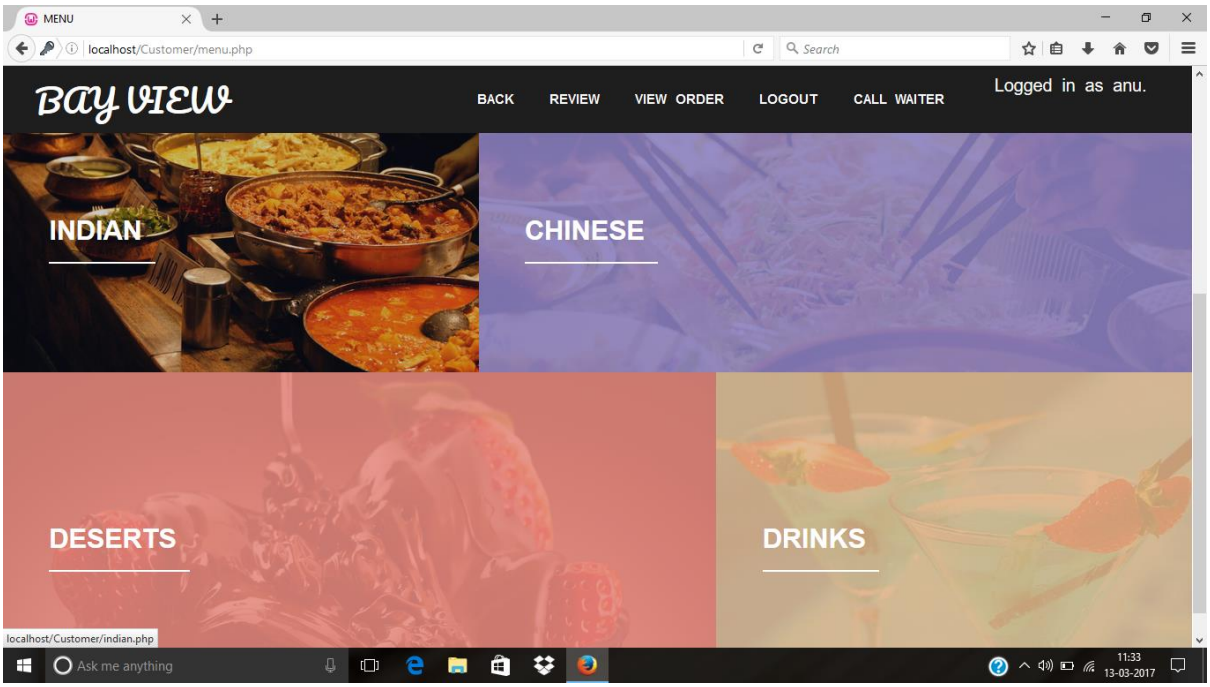
Screenshot S.2 Customer login



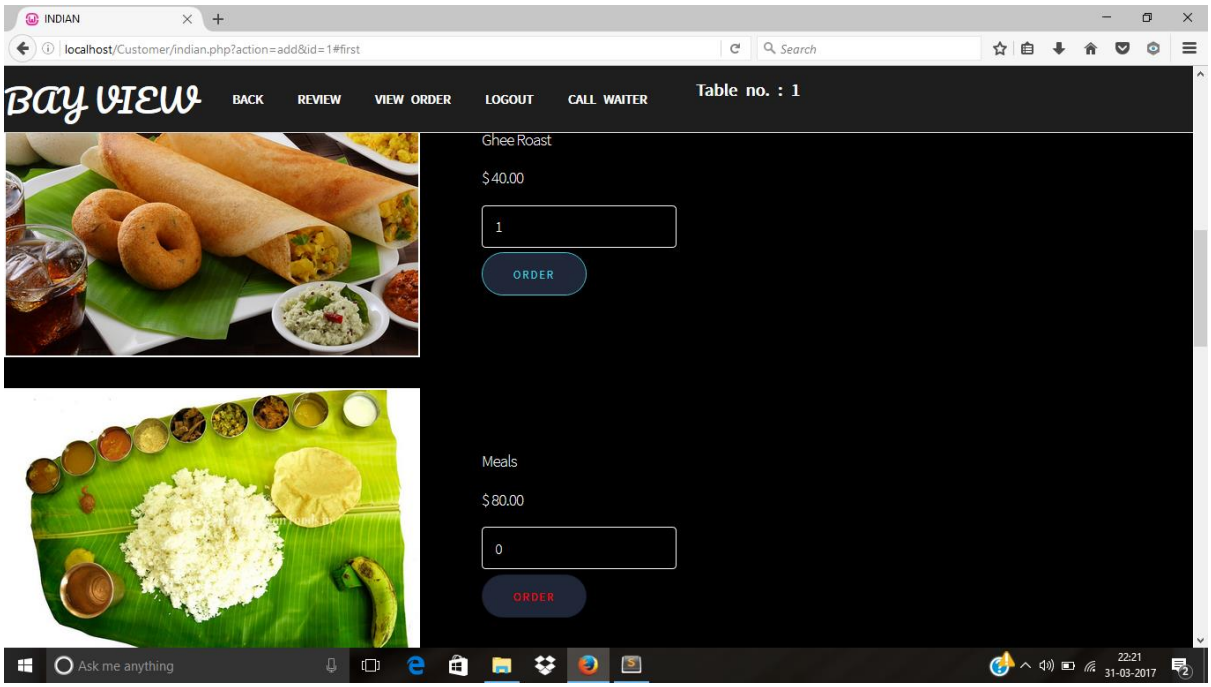
Screenshot S.3 Customer sign up



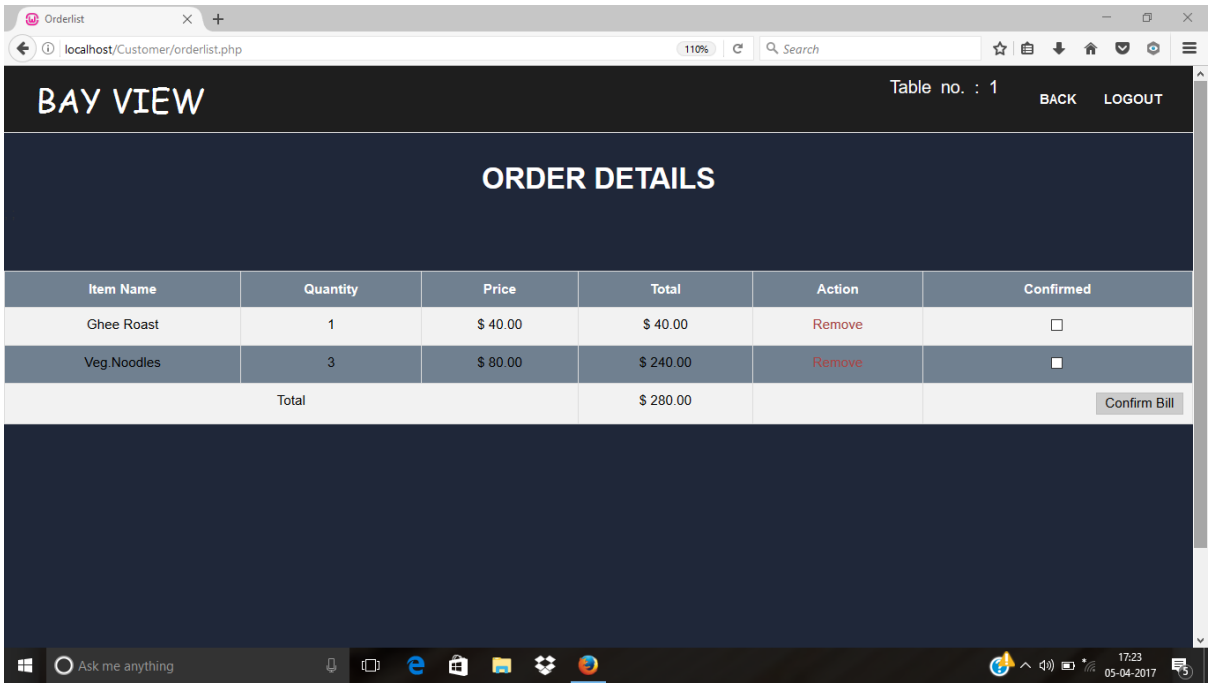
Screenshot S.4 Menu Page



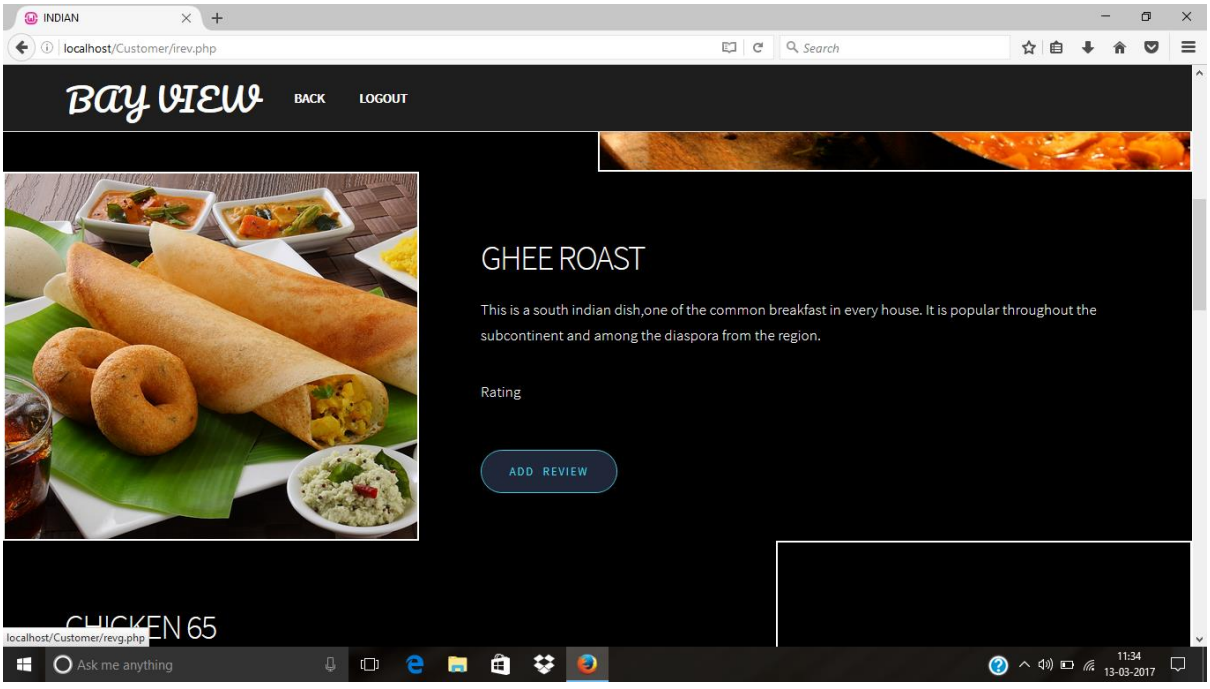
Screenshot S.5 Dishes Page



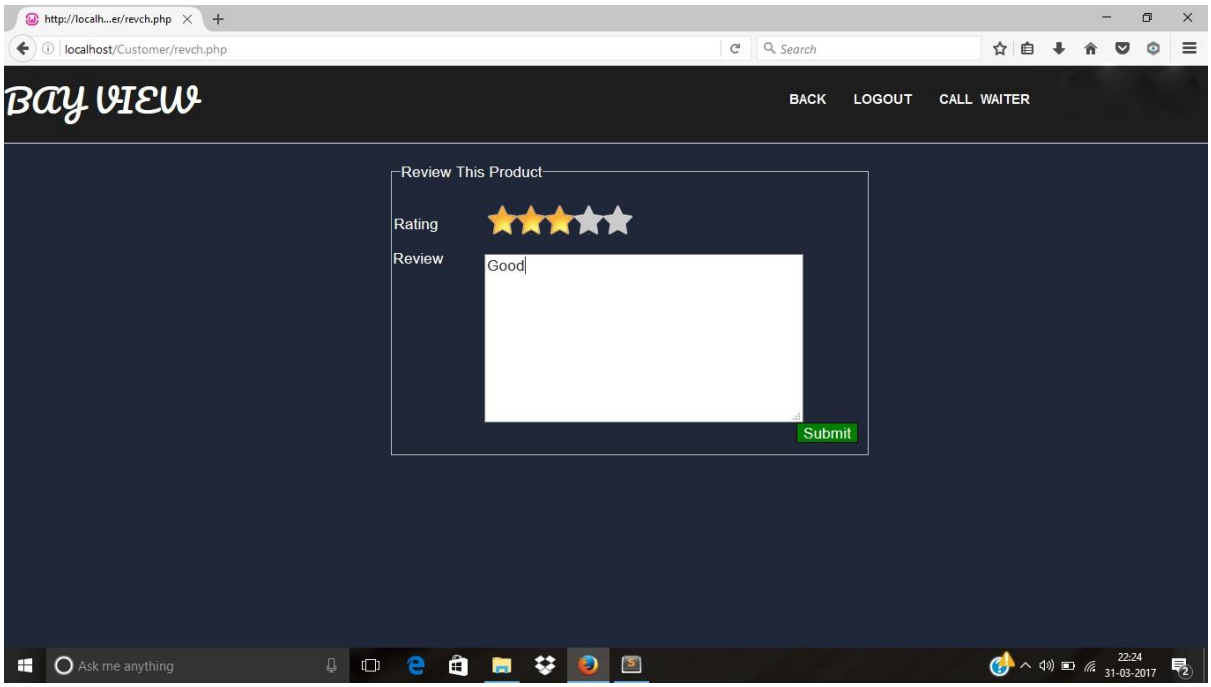
Screenshot S.6 Indian Item Page



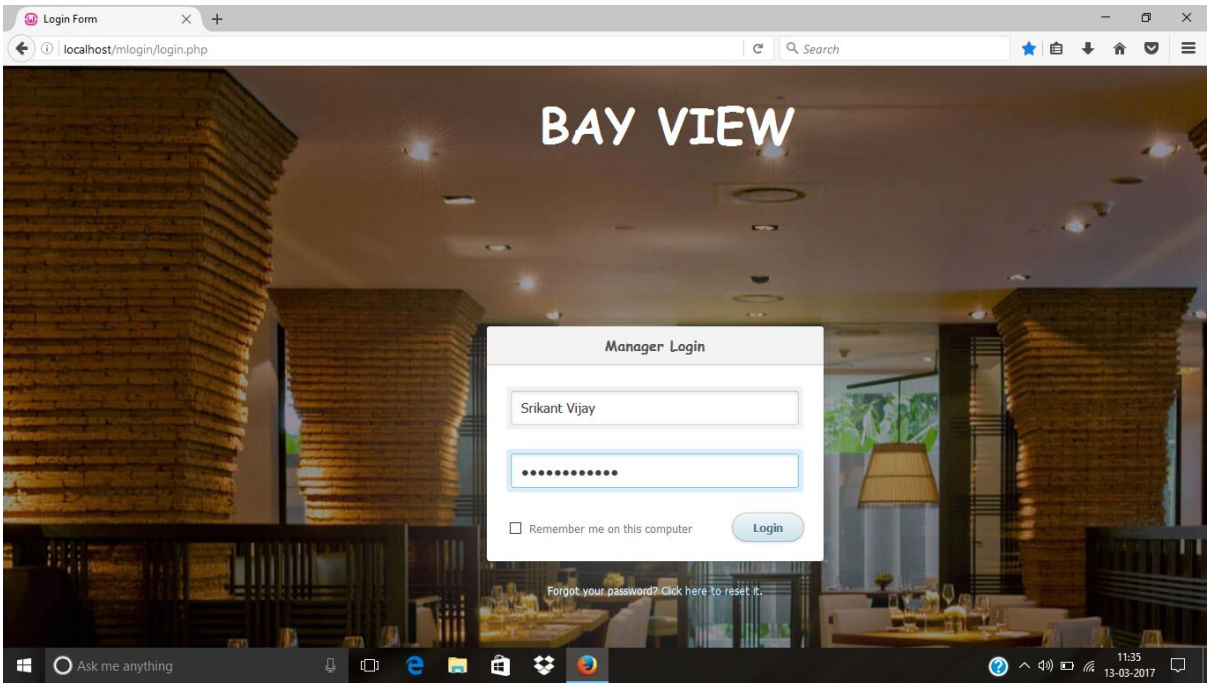
Screenshot S.7 Order page



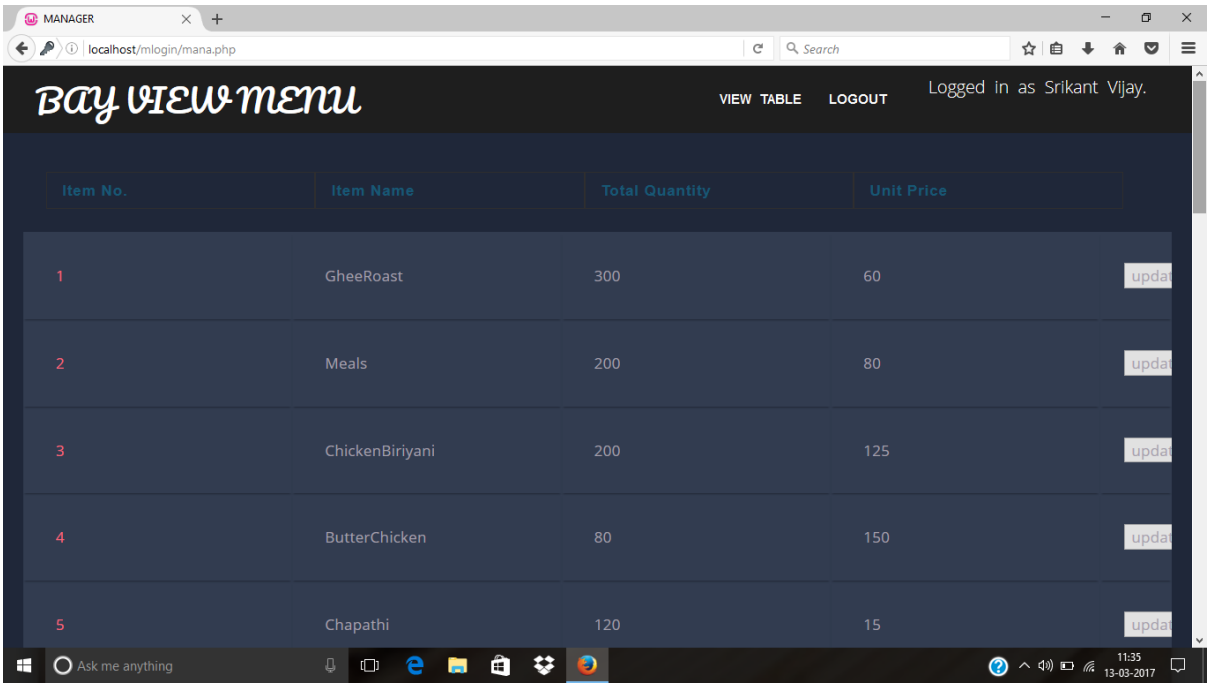
Screenshot S.8 Indian Review Page



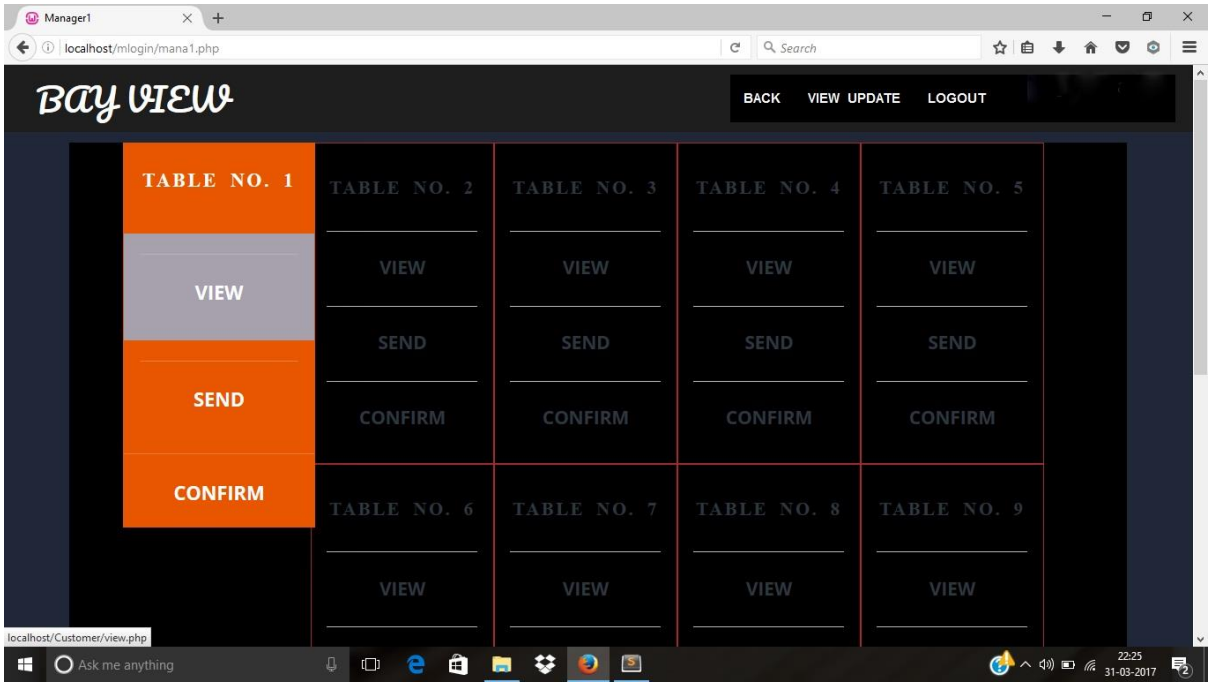
Screenshot S.9 Submit a Review Page



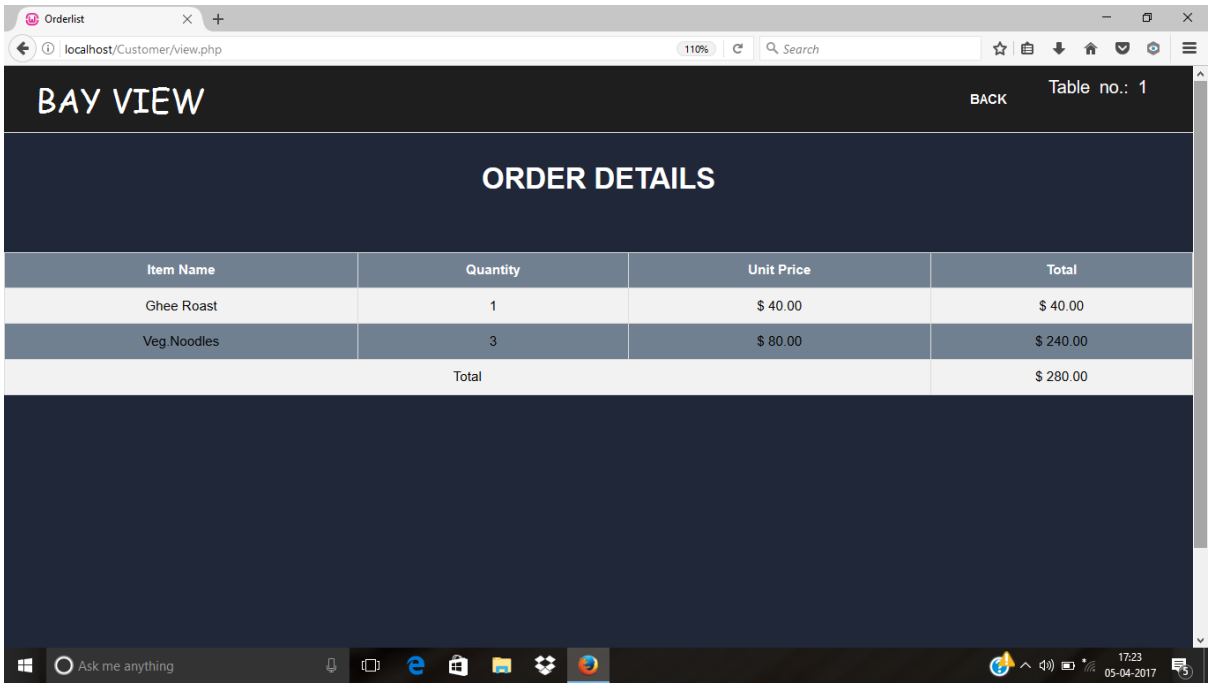
Screenshot S.10 Manager login module



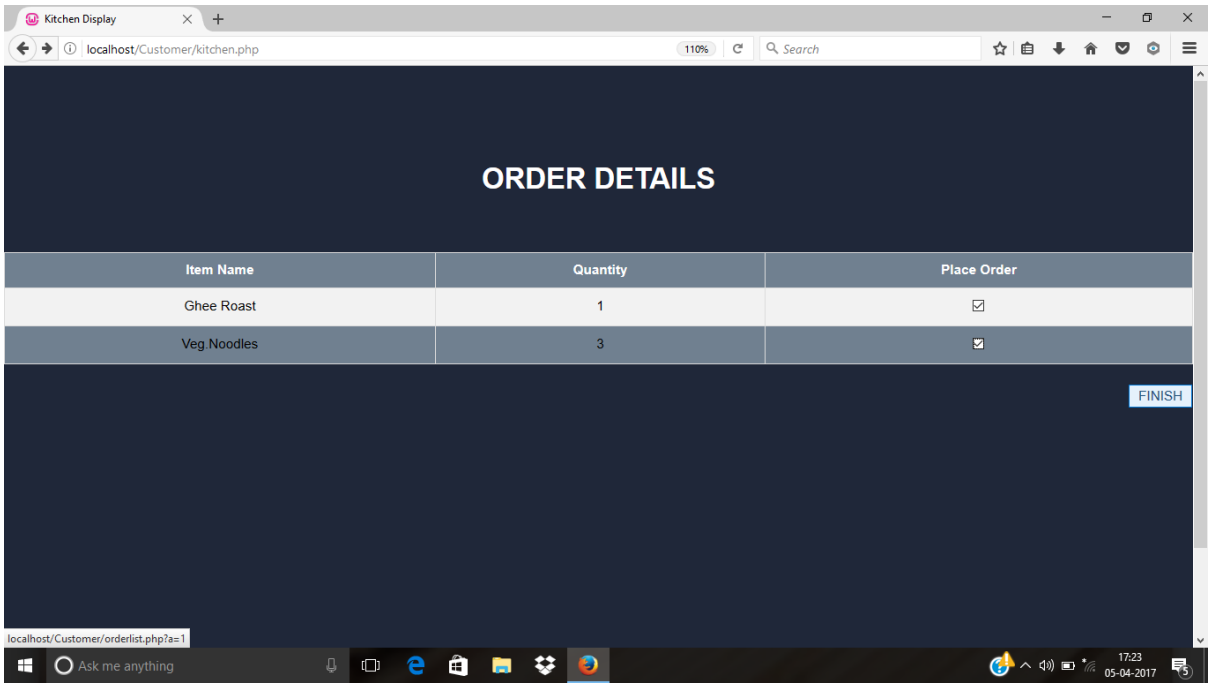
Screenshot S.11 View Update Module



Screenshot S.12 View Table module



Screenshot S.13 View Order page



Screenshot S.14 Kitchen Display

REFERENCES

1. **Bhaskar Kumar Mishra , Bhavani Singh Choudhary and Tanmay Bakshi "**Touch Based Digital Ordering System on Android using Bluetooth for Restaurants**"** *an IEEE paper.*
2. **" http:// www. draw.io.com"** *An online tool for drawing various diagrams.*
3. **"http:// www.stackoverflow.com"** *Project assistance.*
4. **" http://www.w3schools.com"** *Project assistance.*
5. *"**http://www.codecademy.com"** Project assistance.*