1. INTRODUCTION

1.1 Brief Information about the Project

Food Order System is an Application which will help restaurant to optimized and control over their restaurants. For the waiters, it is making life easier because they don't have to go kitchen and give the orders to chef easily. For the management point of view, the manager will able to control the restaurant by having all reports to hand and able to see the records of each employees and orders.

This application helps the restaurants to do all functionalities more accurately and faster way. Food Ordering System reduces manual works and improve efficiency of restaurant. This application is helping Food Ordering to maintain the stock and cash flows and there are many more functionalities like.

- To store records.
- Control orders and services.
- Billings
- Control multiple branches
- Helps Manager to control each part of the restaurant.

The main goal is to maintain the restaurant's functions in an effective and accurate manner and also it is reducing the use of manual entries. This software helps food orders to maintain day to day records in system. It is keeping a proper record of database.

1.2 Motivation and contribution of the project

- ✓ Many restaurants is storing all their data in manual way. They have huge number of customers daily. So because large number of customers, they need the help of some features so they can maintain and store the records accurately. For managers it is difficult to view the tables, orders, and kitchen simultaneously.
- ✓ They need full-fielded software to maintain their day to day transactions, orders and also regular update on records.

1.3 Objective of the Project

This application will attempt to replace the traditional manual ordering process and is a new self-contained software system that consisting of two parts: one web application and the other is Firebase database. The web application will be used for ordering and interacting with the inventory while the Firebase database will be for

used for storing the inventory and ordering related information about the food items like pending and complete orders queries.

1.3 Organization of the Project

- **Chapter 2: Literature Survey:** This chapter consists of background of the project, possible approaches, introduction and comparison of technologies.
- **Chapter 3: System Analysis:** This chapter consists of description of current system, proposed system, algorithms and requirement specifications.
- **Chapter 4: System Design:** This chapter mainly consists of modules description and unified modelling language diagrams: use case diagrams, class diagrams, sequence diagrams, collaboration diagrams, and activity diagrams.
- **Chapter 5: Technology Description:** This chapter mainly consists of technology used in this project.
- **Chapter 6: Sample code:** This chapter mainly consists of sample code for few modules.
- **Chapter 7: Testing:** This chapter mainly consists of testing techniques and test cases for few modules.
- **Chapter 8: Screenshots:** This chapter mainly consists of output screens of this project.

Conclusion: Main conclusion of this project.

2. LITERATURE SURVEY

Ajinkyakumar jadhave proposed the "Development of wireless ordering system for hotel". This work presented in-depth analysis on the technical operation of microcontroller and zigbee module based Wireless Ordering System (WOS) including systems architecture, function and limitations. Implementation of Smart Restaurant with e-menu Card This paper highlighted the limitations of the existing technologies and proposed the advanced system, which focuses on low cost touch-screen development to enhance the dining experience.

The system is compared to advanced food ordering traditional methods such as, traditional pen and paper methods, KIOSK technology which is used for self-service and to enable transaction automation, and PDA's(Personal Digital Assistant) based system brings advancement in the field of food industry by automating the system through mobile and wireless technology. It has the possible to attract customers and changing their view about dining experience in a better and efficient way. [1] The system consists a user phone at the customer table contains the android application with all the menu details. The user phone, kitchen display connects directly with each other through Wi-Fi. Orders made by the customers will quickly reach the kitchen module. Integrating advance features of all the hardware components is used to develop in systems. Presence of every module has reasoned out and placed carefully, contributing to the best working of the unit. [2] System is developed to reduce the number of manpower and at the same time reduce the monthly cost for the restaurant. Customers give their order through the system and directly stored to database. [3] Some of the restriction of the PDA i.e. (Personal Digital Assistant) based food ordering system and proposed the Multi-touchable E restaurant Management System as a solution for the customers. The actual system is consists of the multi- touchable mutual dining menu that allows customers to give their order conveniently on the developed multi touchable dining table during the busy hours. Orders gave by the customers will be updated instantly to a database and subsequently reach to the cashier and the kitchen module respectively. [4] Some efforts have already been taken to carry the process of ordering in hotels by using hardware components like, LCD display module. The Existing system is fully dependent on hardware and it is very difficult to club all the components to make a system. In addition to that understanding and operating the system is very difficult for

some users and this system is not going to manage the business model properly. [5] The integration of touch based technology in restaurants using android system. The tablet at user table contains the android application with the detail information about restaurant and menus. The user tablet, kitchen display and cashier counter connects directly with each other by using Wi-Fi network. [6] Application of integrated restaurant management systems by web services technology is presented through Touch based system. Digital Hotel Management integrates lots of techniques in hotel industry such as Ordering System Kitchen, Order Ticket, Billing System, Customer Relationship Management system (CRM) together. [7] Intelligent Restaurant a Graphical User Interface programmed by embedded system is used for food ordering system. It is required customers to order through Touch Screen device that placed on each table in the restaurant. Customers view the menus, price of food and making order directly using this touch screen device available on table in the restaurant. Then customer order will sent to the database and cash counter computer and also viewed on the computer screen at the kitchen for food preparation. [8] Touch based system interfaces can be effectively increases the operator accuracy, reduce training time, and improve operational efficiencies. Transmission of data which is necessary and performed through Zig-bee which is a wireless technology developed as an open global standard to address the unique needs of systems [9].

3. SYSTEM ANALYSIS

3.1 Existing System

Many restaurants is storing all their data in manual way. The earlier food ordering system was entirely a manual process which involved waiters, pen and paper. The waiter had to note down orders from customers, take these orders to kitchen, update them in records and again make bill. Even though this system is simple it may involve human errors in noting down the orders. To overcome these limitations in manual system some systems were developed later like PDA(personal digital assistants) based systems and multi-touchable restaurant management systems to automate food ordering process.

The usual procedure used for food ordering in restaurants is a manual process. It involves the waiters noting down the menu from customers, transferring the orders to the kitchen, serving the menu, and finally preparing bills. This process even though looks simple, is prone to human errors while note making & delays involved. So the customers end-up with an unsatisfactory experience.

Disadvantages

- This will creates lot of mistakes like misspellings, calculation problems, and duplicate orders.
- It may involve human errors in noting down the orders on papers.
- It is difficult for Manage to supervise all the sides of restaurants Kitchen, floor, counter simultaneously.
- There is functionality to update items at all times.
- If any customer cancel their order it causes wastage of food.

3.2 Proposed System

The proposed system helps in many ways. It helps to do billing very easily. Account maintenance also becomes easier. The software is provided with the facilities to find out favourite food of the customers, and customers to give a suggestion.

Once you entered in Restaurant, management provide a small QR code or android tab at every table. You can just scan that code it will show the table numbers and also current available items are presently at restaurant. All the tables are numbered.You can select various items from a drop down menu and input the quantity ordered. Which will return at Kitchen Deportment.After ordering it shown the order status like order is placed or not. We can cancel order for some particular time period, once

given time lapse after we can't modify your order items. We can track your order prepared time on the screen. This system calculate the total bill for ordered items. Customer can view the rating of food items. Customer want to give any special instructions to chef by using the grievances box. We can pay the bill amount at the table by using the online payment systems.

Advantages

- Managers can view the kitchen side, floor side and the counter side details simultaneously. Dynamic tuning strategies to tailor to individual's memorization strength and recall habits based on relevance feedback are developed for performance improvement.
- All the orders per day will updated in the database daily. It is used in surveys and some businesses how to increase their business.
- Also keeps the records of food items prepared and the sales of food and the record of balance food.
- Less use of manual works.
- Eliminate the paper and pen based orders via waiters.
- Reduces the human powers.
- No need to carry paper based cash.
- Saves the time for food ordering.

3.3 Feasibility study

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Economical Feasibility
- Operational Feasibility
- Technical Feasibility

Economic Feasibility

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the

development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?

- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

3.4 Functional Requirements

- Food Ordering is having many modules, which make the software more efficient and user friendly. The modules make the maintenance of database easy.
- Every module is divided on the basis of the senarios. Bellow are the main scenarios
    - Cheff/Kitchen
    - Admin/Manger
    - Waiter/ Floor
- The different modules in this project are:
        - Food Items
        - Orders
        - Cancellations.

3.5 Non Functional Requirements

The major non-functional Requirements of the system are as follows

- **Adaptability**: There can be a change in the menu and information stored in the database about employee and inventory.
- **Availability:** The system is up and running for most of the time and server is not down for more than few minutes to avoid inconvenience of the customers.
- **Correctness**: The bill generated by the application must be accurate and the orders placed should exactly be the same which the user has selected.
- **Flexibility**: If need arises in the future, software can be modified to change the requirements.
- **Inter-operability**: The data is transferred from the customer's end to the kitchen and then head of chef assigns orders to each chef.
- **Maintainability**: software can be easily repaired if a fault occurs.
- **Portability**: Software can be easily installed on devices and would run smoothly according to the requirement.
- **Reliabily**: No matter how many orders are placed, system must give the correct results.
- **Reusability**: Current version can be used in the future versions with more functionality added.
- **Robustness**: Software must have checks to ensure that the items that are not available in the menu cannot be selected.
- **Usability**: Interface of the software must be easy to use. It would not be complex since managers, chefs have a view, so interface should be simple.

3.6 Requirements Specification

3.6.1 Minimum Hardware Requirements

- Processor : Pentium IV
- Hard Disk : 40GB
- RAM : 512MB

3.6.2 Software Requirements

- Operating System : Windows XP/2003

- User Interface : HTML, CSS
- Programming Language : Raw PHP, Jquery
- IDE : Sublime Text 3
- Web Applications : XAMPP, Apache
- Database : MySQL
- Server Deployment : Apache

4. SYSTEM DESIGN

4.1 Introduction

After analyzing the requirement of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is more difficult, improper understanding of present system can lead diversion from solution.

4.2 System Architecture

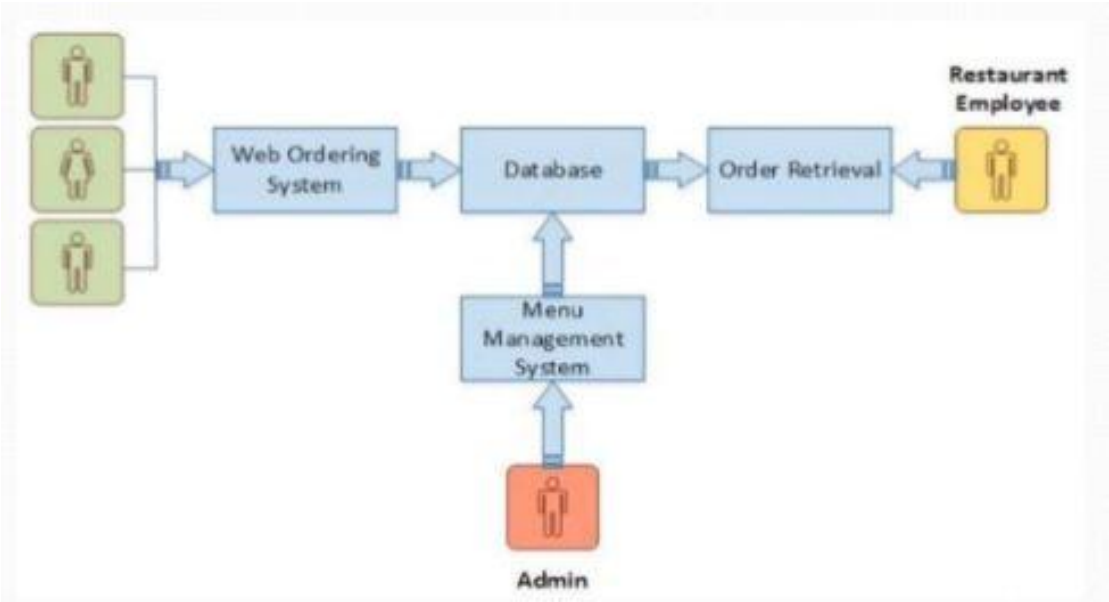


Fig 4.2.1: System Architecture

Architecture Description:

- Admin can manage all activities in Restaurant. Some of them are Add items, Modify Items, Manage the Orders Status, Adding Tables, and Monitor the user activates.
- Restaurant Employee are Retrieval order details From database.
- Customers / User who are interacting restaurant through "web Ordering System interface". Its shown the all available items from the database and it update the Restaurant administrator.

4.3 Modules Description

Food Ordering is having many modules, which make the software more efficient and user friendly. The modules make the maintenance of database easy. Every module is divided on the basis of the scenarios. Below are the main scenarios

1. Chef /Kitchen
2. Admin/Manger
3. Waiter/ Floor

The different modules in this project are:

➢ Food Items:

- **Manager Side/Admin Side:**

It include checking menu items, Modify Items in Menu, Reserve the Tables, Checking and Process of payments. In this module Admin can Insert, Update, Delete Food and their prices like stores the details of items. In this mainly focus into bellow page:

1. Insert food items.
2. Insert Prices.
3. Delete Food Items.
4. Delete Prices.
5. Modify Price.

➢ Order:

- **Kitchen Side / Cheff Side:**

It will keep the records of orders. So chef's will able to see and give the order waiters. In this module, Stores all types of orders.

1. Insert Order.
2. Insert Type of Order.
3. Insert Price.
4. Delete Order
5. Delete Price
6. Modify Price.

- **Cancellation :**

  - **Floor Side / Waiter Side:**

There will be details about orders, Details about waiter and billing facility will be there in this module, the customer will cancel ordered items its will stored into cancel items table.

4.4 Data Dictionary

Column name	Data type	Constraint name	Description
Id	Integer	primary key	specifies user id
Name	varchar2(20)	not null	specifies username
Email	varchar2(20)	not null	specifies email
Password	varchar2(20)	not null	specifies password
Status	varchar2(20)	not null	specifies Status

Table 4.4.1: Data Table for Users

Column name	Data type	Constraint name	Description
Id	Integer	primary key	specifies ItemId
Item_name	varchar2(20)	not null	specifies itemname
Item_category	varchar2(20)	not null	specifies cat_name
Item_cost	Integer	not null	specifies Itemcost
Item_type	varchar2(20)	not null	specifies Item type
photo	text	not null	Specifies photo
avaliable	Integer	Notnull	Specifies itemavaliable
Quantity	Integer	not null	specifies Item Quality
Status	Integer	not null	specifies status

Table 4.4.2: Data Table for Items Details.

Column name	Datatype	Constraint	Description
Id	Integer	Primary Key	Specifies ID
Table_number	Varchar2(20)	Not null	Specifies table Name
T_number	Integer	Not null	Specifies table Number
T_capacity	Varchar2(20)	Not null	Specifies table cost
T_cost	Varchar2(20)	Not null	Specifies table cost
Timestamp	Timestamp	Current time	Specifies timestamp
status	Integer	Not null	Specifies table status

Table 4.4.3: Data Table for Table List Details.

Column name	Data type	Constraint	Description
Id	Integer	Primary Key	Specifies ID
C_name	Varchar2(20)	Not null	Specifies Category Name
Photo	Text	Not null	Specifies Category Photo
status	Varchar2(20)	Not null	Specifies Category status

Table 4.4.4: Data Table for Category List Details.

Column name	Data type	Constraint	Description
Id	Integer	Primary Key	Specifies ID
User_id	Integer		Specifies User Id
Item_id	Integer		Specifies Item Id
quantity	Integer		Specifies item quantity
status	Varchar2(20)	Not null	Specifies Cart status

Table 4.4.5: Data Table for Cart Details.

Column name	Data type	Constraint	Description
Id	Integer	Primary Key	Specifies ID
Order_id	Integer		Specifies Order number
User_id	Integer		Specifies User Id
Item_id	Integer		Specifies Item Id
Table_no	Integer		Specifies Table number
quantity	Integer		Specifies item quantity

Giverrence	Varchar2(20)	Not null	Specifies User's Requests
Timestamp	Timestamp	Current time	Specifies time stamp
status	Varchar2(20)	Not null	Specifies orders status

Table 4.4.6: Data Table for Order Table Details.

Column name	**Data type**	**Constraint**	**Description**
Id	Integer	Primary Key	Specifies ID
tableno	Varchar2(20)	Not null	Specifies table number
Num_of_guest	Varchar2(20)	Not null	Specifies number of guests
Email	Varchar2(20)	Not null	Specifies User email id
Phone	Varchar2(20)	Not null	Specifies user phone number
Date_res	Date	Not null	Specifies date of reserve
time	Datetime	Not null	Specifies time of revered
suggestions	Varchar2(20)	Not null	Specifies User's Requests
cost	Integer	Not null	Specifies table Cost
status	Varchar 2(20)	Not null	Specifies orders status

Table 4.4.7: Data Table for reservation TableDetails.

4.5 UML Diagrams

- The unified modelling language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.
- A UML system is represented using five different views that describe the system from distinctly different perspective. UML is specifically constructed through two different domains.
- UML Analysis modelling, this focuses on the user model and structural model views of the system.
- UML design modelling, which focuses on the behavioural modelling, implementation modelling and environmental model views.

These are divided into the following types.

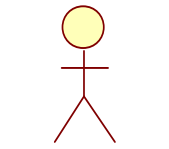
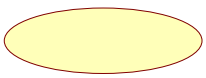
- Use case diagram
- Class diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram

4.5.1 Use case Diagram

Use Case diagrams identify the functionality provided by the system (use cases), the users who interact with the system (actors), and the association between the users and the functionality. Use Cases are used in the Analysis phase of software development to articulate the high-level requirements of the system. The primary goals of Use Case diagrams include:

- Providing a high-level view of what the system does.
- Identifying the users ("actors") of the system.
- Determining areas needing human-computer interfaces.

Graphical Notation: The basic components of Use Case diagrams are the Actor, the Use Case, and the Association.

Actor	An Actor, as mentioned, is a user of the system, and is depicted using a stick figure. The role of the user is written beneath the icon. Actors are not limited to humans. If a system communicates with another application, and expects input or delivers output, then that application can also be considered an actor.	 **Actor Role Name**
Use Case	A Use Case is functionality provided by the system, Use Cases are depicted with an ellipse. The name of the use case is written within the ellipse.	 **Use Case Name**
Directed Association	These Associations are used to link Actors with Use Cases, and indicate that an Actor	

participates in the Use Case in some form. 

Behind each Use Case is a series of actions to achieve the proper functionality, as well as alternate paths for instances where validation fails, or errors occur. These actions can be further defined in a Use Case description. Because this is not addressed in UML, there are no standards for Use Case descriptions. However, there are some common templates can follow, and whole books on the subject writing of Use Case description.

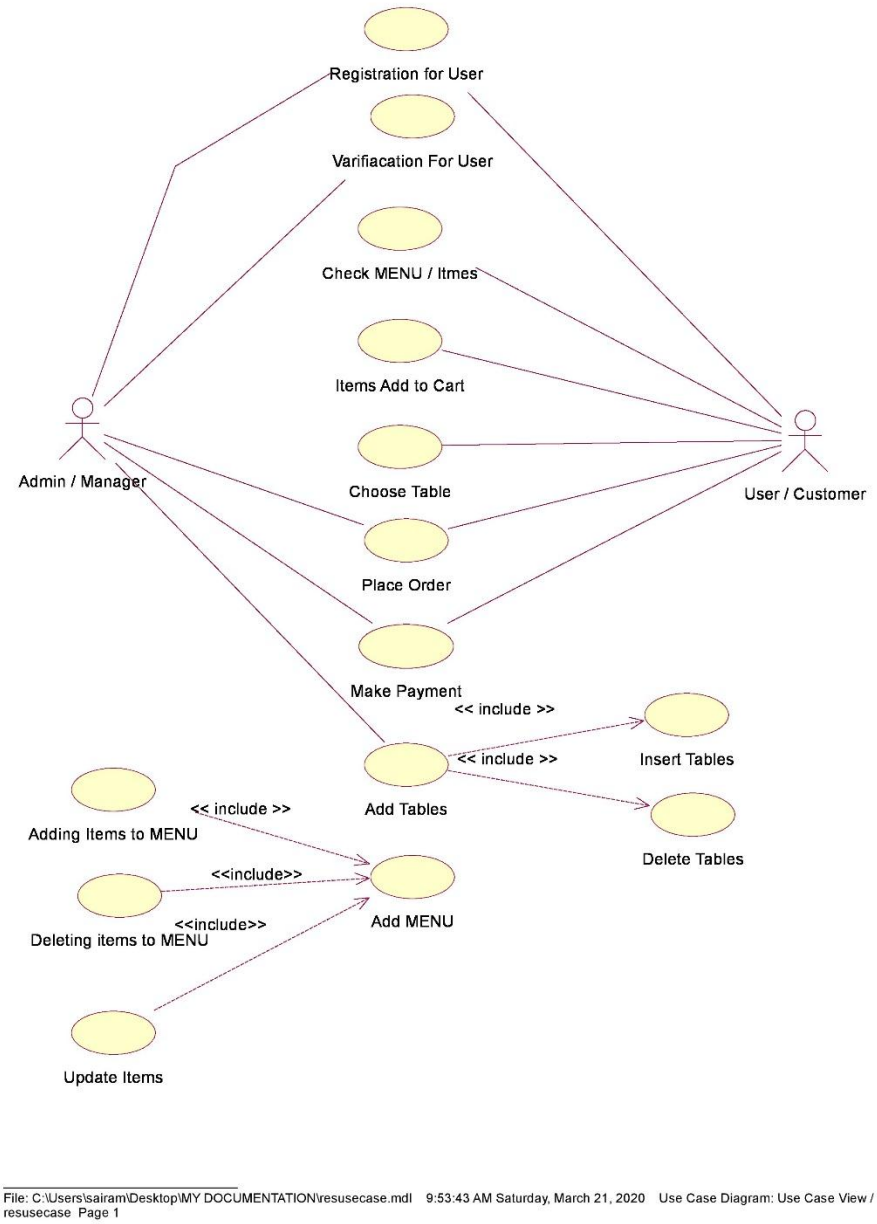


Fig 4.5.1 Use case diagram for system

Description: The above diagram describes the use case diagram of the project "Automation of Waiter's Role at Restaurant". In the above use case diagram, Admin, user are participated as actors. Admin can verify the users; add items to the menu list, add tables, modify the list of items based on availability. User can view the all updated items on the tab Screen and User can place order from this list. Finally paid the total bill.

Use case Templates

Usecasename	**Login**
Participating actors	Admin/Manager, User / Customer.
Flow of Events	1.User can Register into System. 2.User Login in the System By using entering the Details. 3.User details stores in database.
Entry condition	Source should login into the system
Exit condition	Success to registered.

Table 4.5.1.1: Use case template for login

Usecasename	**Check Menu / Items**
Participating actors	User
Flow of Events	1.User can view already displayed Items by using Login Credentials.
Entry condition	Enter into Menu Page
Exit condition	Items added to Cart.

Table 4.5.1.2: Use case template for Menu/Items

Usecasename	**Items Add to Cart**
Participating actors	User
Flow of Events	1.User can views items from server 2.Choose Items Form Menu. 3.By click add to cart button items are added..
Entry condition	Through Login.

Exit condition	After add to cart.

Table 4.5.1.3: Use case template for add to cart

Usecasename	**Place Order**
Participating actors	User, admin/manager
Flow of Events	1.User can checks cart details 2.Choose table form table list. 3.By click add to cart button items are added..
Entry condition	Through Login. And checking cart page.
Exit condition	After place Order.

Table 4.5.1.4: Use case template for Place Order.

4.5.2 Class Diagram

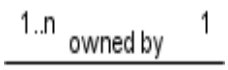
Class diagrams identify the class structure of a system, including the properties and methods of each class. Also depicted are the various relationships that can exist between classes, such as an inheritance relationship. Part of the popularity of Class diagrams stems from the fact that many CASE tools, such as Rational XDE, will auto-generate code in a variety of languages, these tools can synchronize models and code, reducing the workload, and can also generate Class diagrams from object-oriented code.

Graphical Notation: The elements on a Class diagram are classes and the relationships between them.

Class

Classes are the building blocks in object-oriented programming. A Class is depicted using a rectangle divided into three sections. The top section is the name of the Class. The middle section defines the properties of the Class. The bottom section lists the methods of the class.

Association



An Association is a generic relationship between two classes, and is modeled by a line connecting the two classes. This line can be qualified with the type of relationship, and can also feature multiplicity rule (e.g. one-to-one, one-to-many, many-to-many) for the relationship.

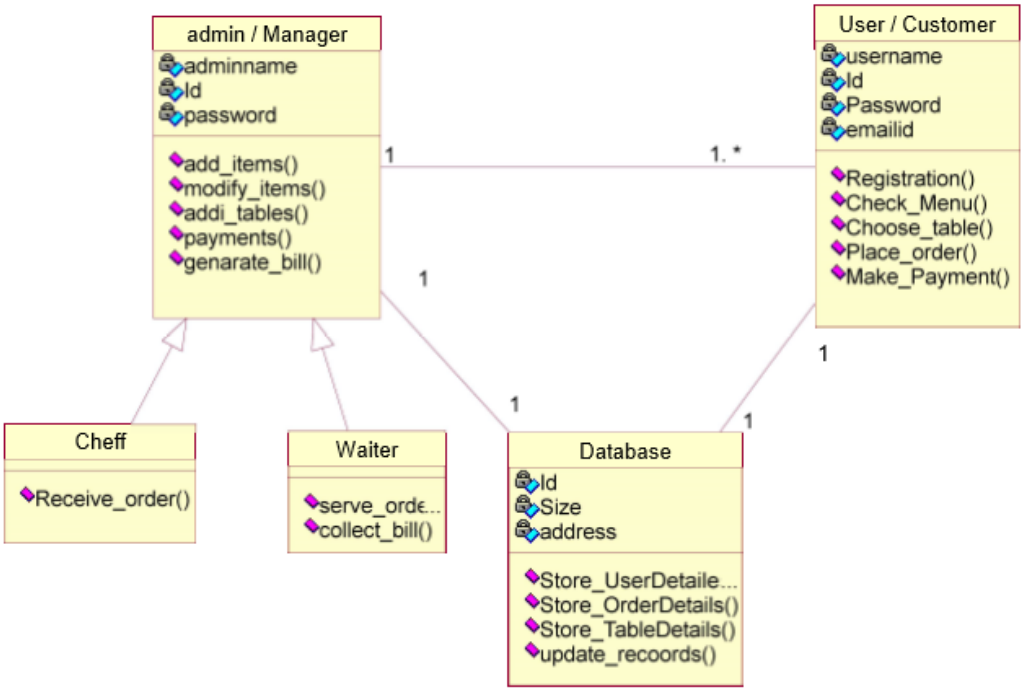


Fig 4.5.2 Class diagram for system

Description: The above diagram describes the class diagram of the project "Automation of Waiter's Role at Restaurant". In the above class diagram describes the relationship between admin, user, Database. User should browse the data and it sends to the admin. User can send the data through the Database. Admin can generalized into chef and waiter.

4.5.3 Sequence diagram

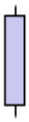
Sequence diagrams document the interactions between classes to achieve a result, such as a use case. Because UML is designed for object-oriented programming, these communications between classes are known as messages. The Sequence diagram lists objects horizontally, and time vertically, and models these messages over time.

Graphical Notation: In a Sequence diagram, classes and actors are listed as columns, with vertical lifelines indicating the lifetime of the object over time.

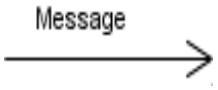
Object Objects are instances of classes, and are arranged horizontally. The pictorial representation for an Object is a class (a rectangle) with the name prefixed by the object name (optional) and a semi-colon.

Lifeline The Lifeline identifies the existence of the object over time. The notation 2 for a Lifeline is a vertical dotted line extending from an object.

Activation Activations, modeled as rectangular boxes on the lifeline, indicate when the object is performing an action.



Message Messages, modeled as horizontal arrows between Activations, indicate the communications between objects.



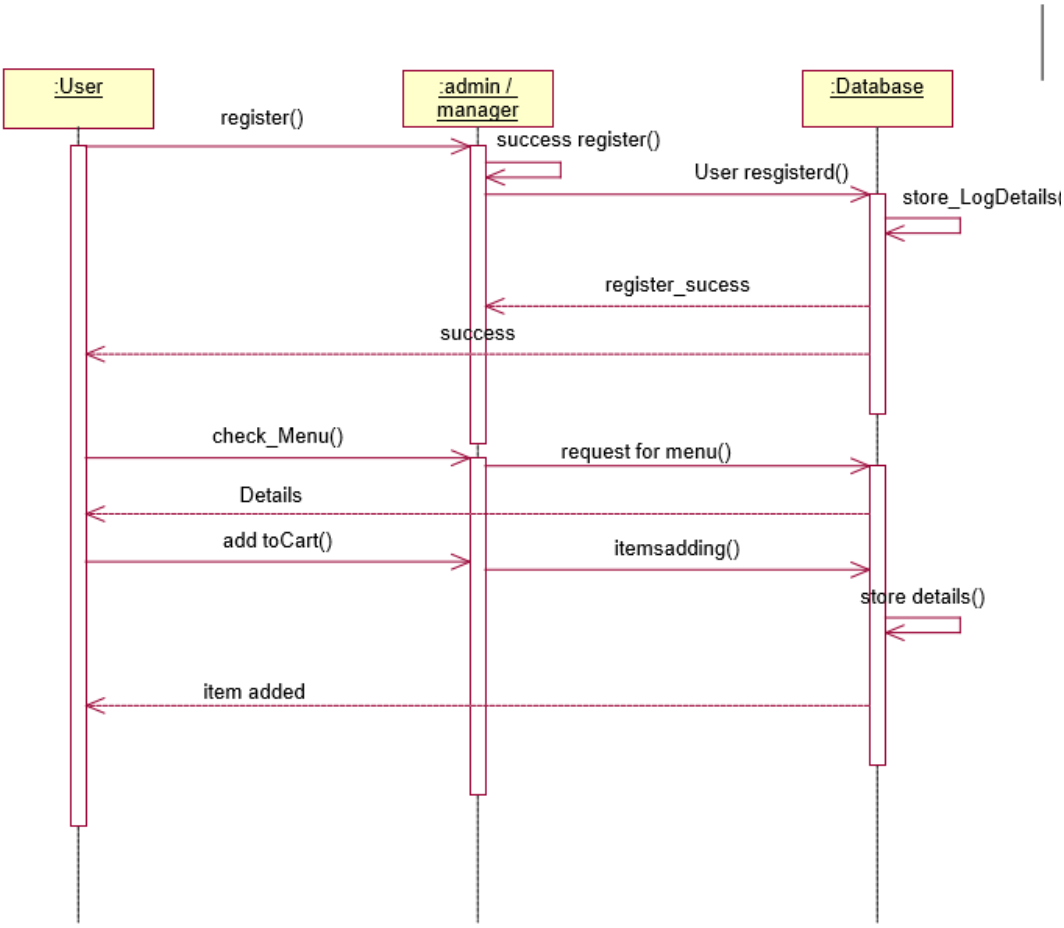


Fig 4.5.3 Sequence diagram for addtocart

Description: The above diagram describes the sequence diagram of the project "Automation of Waiter's Role At Restaunrant". In the above sequence diagram, user, admin, database are objects. These three objects are communicated by sending the messages. In this user can check menu and send the data to the admin. In that process User can add items to this cart page. Database can store all data into their log table.

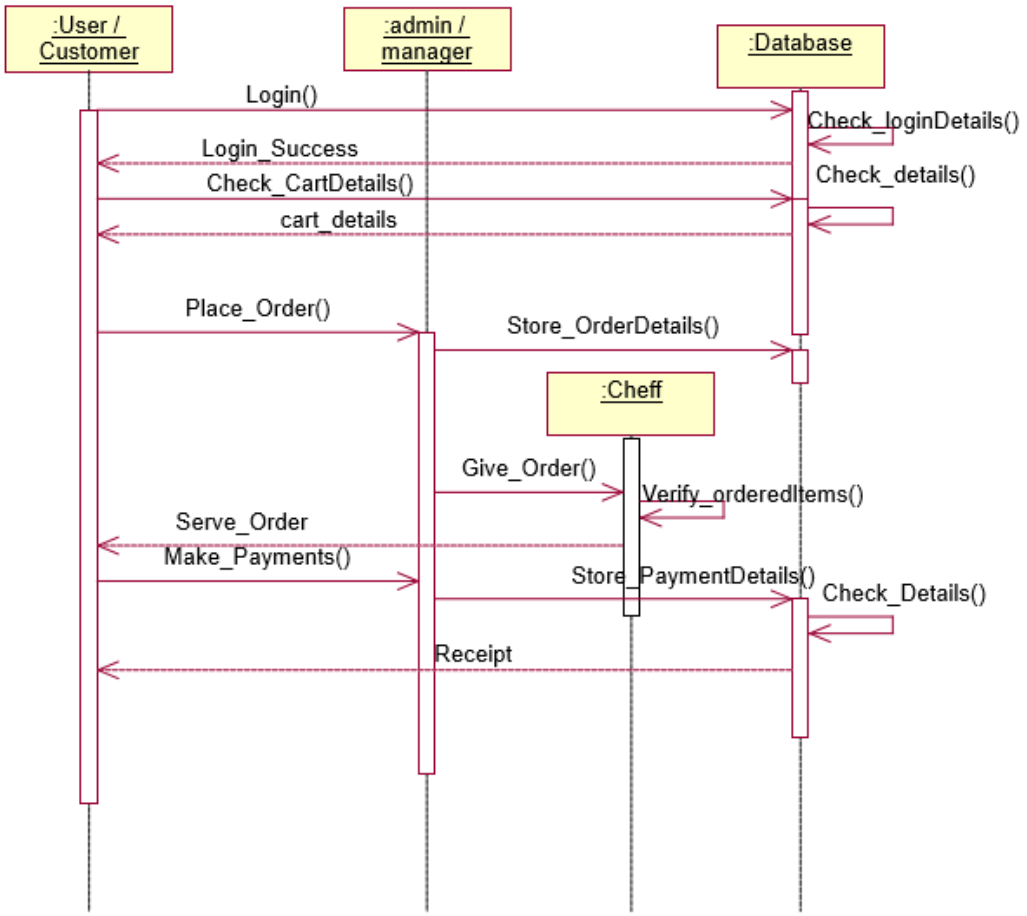


Fig4.5.3.2: Sequence diagram for placeOrder.

Description: The above diagram describes the sequence diagram of Placing Order to the project "Automation of Waiter's Role At Restaurant". In the above sequence diagram, user, chef, admin, database are objects. These four objects are communicated by sending the messages. In this user can check cart table and send the data to the admin. In that process User can place orders to this orders page. Database can store all data into their orders table.

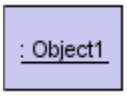
4.5.4 Collaboration Diagram

Like the other Behavioral diagrams, Collaboration diagrams model the interactions between objects. This type of diagram is a cross between an object diagram and a sequence diagram. Unlike the Sequence diagram, which models the

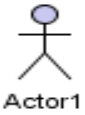
interaction in a column and row type format, the Collaboration diagram uses the free-form arrangement of objects as found in an Object diagram. This makes it easier to see all interactions involving a particular object.

Graphical Notation

Object — Objects are instances of classes, and are one of the entity types that can be involved in communications. An Object is drawn as a rectangular box, with the class name inside prefixed with the object name (optional) and a semi-colon.



Actor — Actors can also communicate with Objects, so they too can be listed on Collaboration diagrams. An Actor is depicted by a stick figure.



Message — Messages, modeled as arrows between objects, and labeled with an ordering number. Indicate the communications between objects.

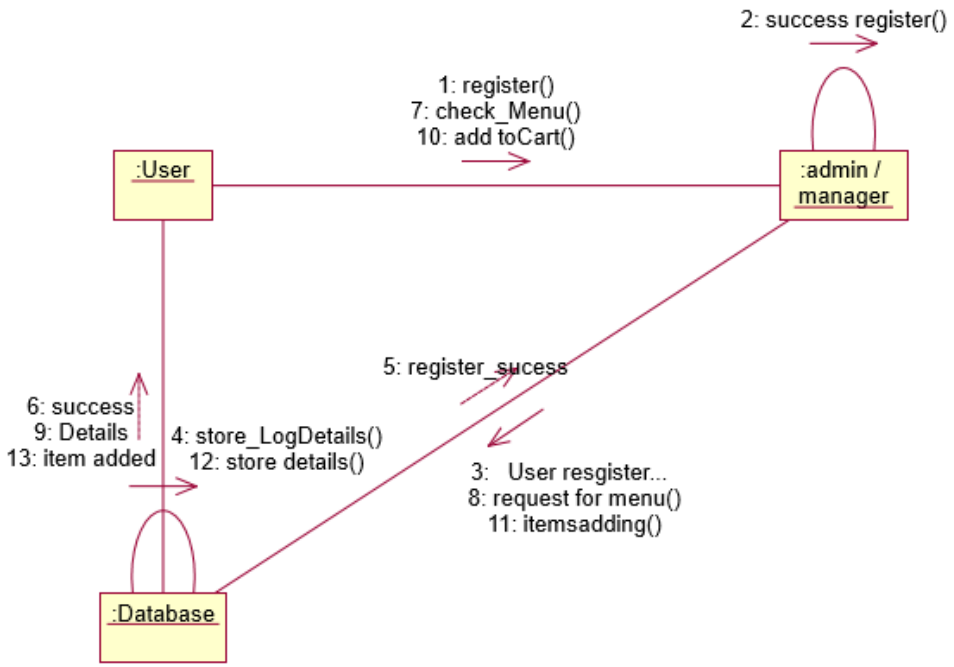


Fig 4.5.4 Collaboration diagram for addtocart

Description: The above diagram describes the use case diagram of the project "Automation

of Waiter's Role at Restaurant". In the above collaboration diagram, there are three objects that are user, admin, Database. Using the sequencing number the objects are communicated. In this user can send the data to the admin.

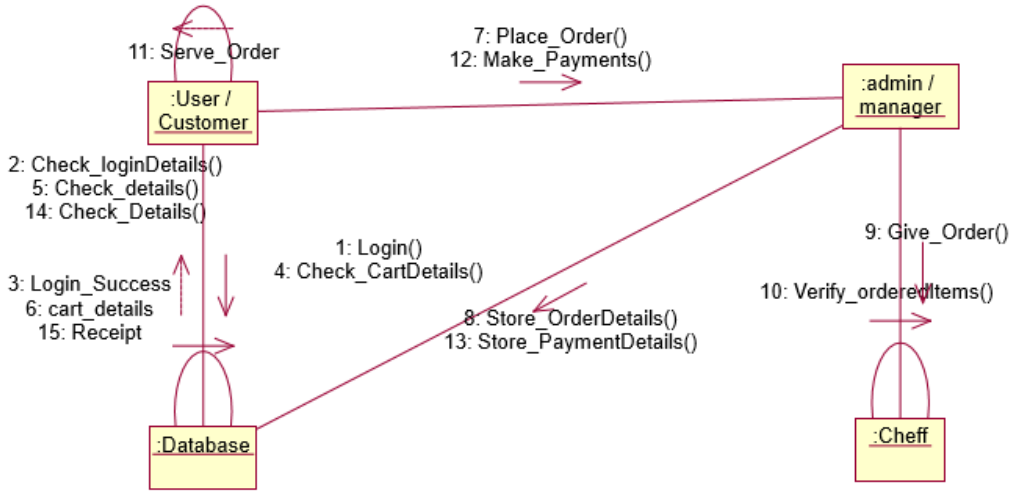


Fig4.5.4.2: Collaboration diagram for place Order.

Description: The above diagram describes the use case diagram of the project "Automation of Waiter's Role at Restaurant". In the above collaboration diagram, four are three objects that are user, admin, chef, Database. Using the sequencing number the objects are communicated. In this user can send the data to the admin.

4.5.5 Activity Diagram

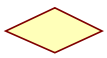
This shows the flow of events within the system. The activities that occur within a use case or within an objects behavior typically occur in a sequence. An activity diagram is designed to be simplified look at what happens during an operations or a process.

Each activity is represented by a rounded rectangle the processing within an activity goes to compilation and then an automatic transmission to the next activity occurs. An arrow represents the transition from one activity to the next. An activity

diagram describes a system in terms of activities. Activities are the state that represents the execution of a set of operations. These are similar to flow chart diagram and dataflow.

Initial state	Which state is starting the process?	
Action State	An action state represents the execution of an atomic action, typically the invocation of an operation. An action state is a simple state with an entry action whose only exit transition is triggered by the implicit event of completing the execution of the entry action.	
Transition	A transition is a directed relationship between a source state vertex and a target state vertex. It may be part of a compound transition, which takes the static machine from one static configuration to another, representing the complete response of the static machine to a particular event instance	
Final state	An end point represents the last or "final" activity of the enclosing composite activity. There may be more than one final point at any level signifying that the composite activity can end in different ways or conditions. When a final activity is reached and there are no other enclosing activity it means that the entire activity has completed its transitions and no more transitions can occur.	

Decision A state diagram (and by derivation an activity diagram) expresses decision when guard conditions are used to indicate different possible transitions that depend on Boolean conditions of the owning object.



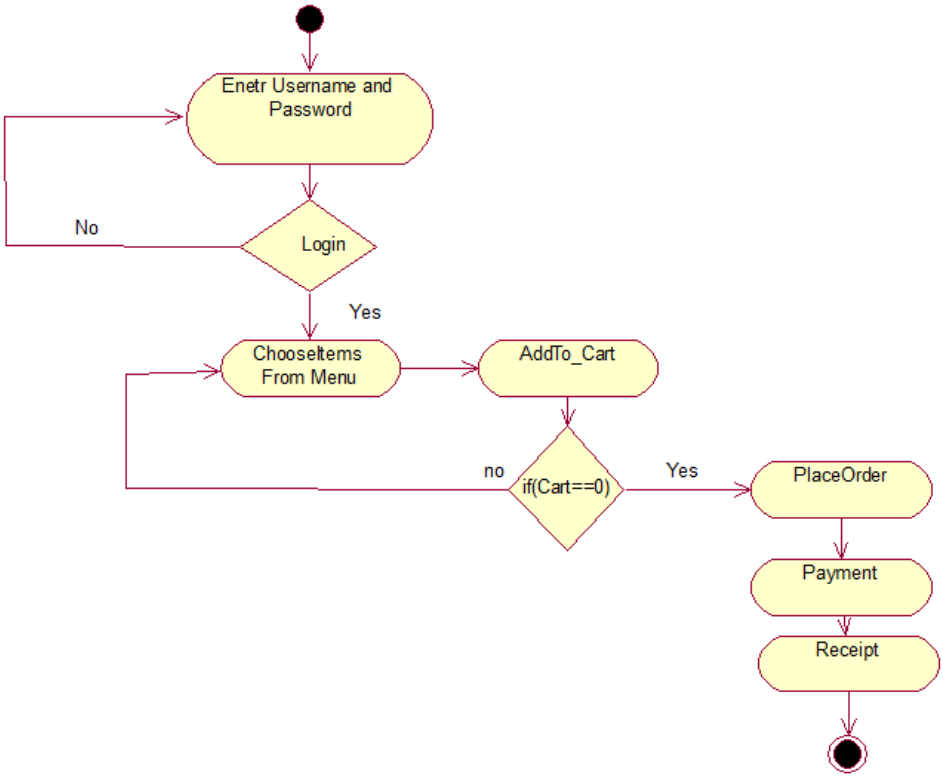


Fig 4.5.5 Activity diagram for system

Description: The above diagram describes the sequence diagram of the project "Automation of Waiter's Role at Restaurant". In the above activity diagram, it first register the restaurant customer then login to their account. After that check cart table, if items are in their cart then place order or else again to back to home page and choose items and finally place orders.

5. TECHNOLOGY DESCRIPTION

5.1 Introduction to PHP

Java technology is both a programming language and a platform.

PHP Technology

PHP is an acronym for "PHP: Hypertext Preprocessor"

PHP is a widely-used, open source scripting language.

PHP scripts are executed on the server.

PHP is free to download and use.

The PHP Scripting Language

PHP is an alternative to Microsoft's Active Server Page (ASP) **Technology**. As with ASP, the **PHP** script is embedded within a Web page along with its HTML. Before the page is sent to a user that has requested it, the Web server calls **PHP** to interpret and perform the operations called for in the **PHP** script.

PHP is an **HTML**-embedded Web scripting language. This means **PHP** code can be inserted into the HTML of a Web page. When a PHP page is accessed, the **PHP** code is read or "parsed" by the server the page resides on. The output from the **PHP** functions on the page are typically returned as **HTML** code, which can be read by the browser. Because the **PHP** code is transformed into HTML before the page is loaded, users cannot view the PHP code on a page. This make **PHP** pages secure enough to access databases and other secure information.

A lot of the syntax of PHP is borrowed from other languages such as C, Java and Perl. However, PHP has a number of unique features and specific functions as well. The goal of the language is to allow Web developers to write dynamically generated pages quickly and easily. PHP is also great for creating database-driven Web sites. If you would like to learn more about PHP, the official site is PHP.net.

"PHP is a server-side, cross-platform, HTML embedded scripting language."

That's a mouthful, but if we break the definition down into smaller pieces, it is easier to understand.

- Server-Side
- Cross-platform

- Html Embedded Scripting Language
- PHP Development
- PHP includes & require Statements
- PHP Variables
- PHP echo & print Statements
- PHP Functions
- PHP MySQL Database
- PHP Connect to MySQL
- Secure

Server Side:

This means that PHP scripts execute on the Web server, not within the browser on your local machine.

Cross-Platform:

Cross-platform means that PHP scripts can run on many different operating systems and Web servers. PHP is available for the two most popular Web server configurations (IIS running on Windows NT and Apache running on UNIX).

HTML Embedded Scripting Language:

This means that PHP statements and commands are actually embedded in your HTML documents. When the Web server sees the PHP statements in the Web page, the server executes the statements and sends the resulting output along with the rest of the HTML. PHP commands are parsed by the server much like Active Server Pages or Cold Fusion tags.

PHP Development:

PHP is strong tool for create dynamic and interactive Web pages. PHP is the widely-used, free, and efficient for rich applications/website development. This is open source technology, runs on Apache web server which in turn runs seamlessly on Windows, Linux, Solaris, and various other UNIX platforms. Sun core Microsystems's PHP development services offers unique, dynamic and highly functional web applications for across the world. We have complete experience in providing solutions to companies ranging from small websites to more complex/large

websites. Our team of experts always follows well-defined development methodology and applies quality standards with each website.

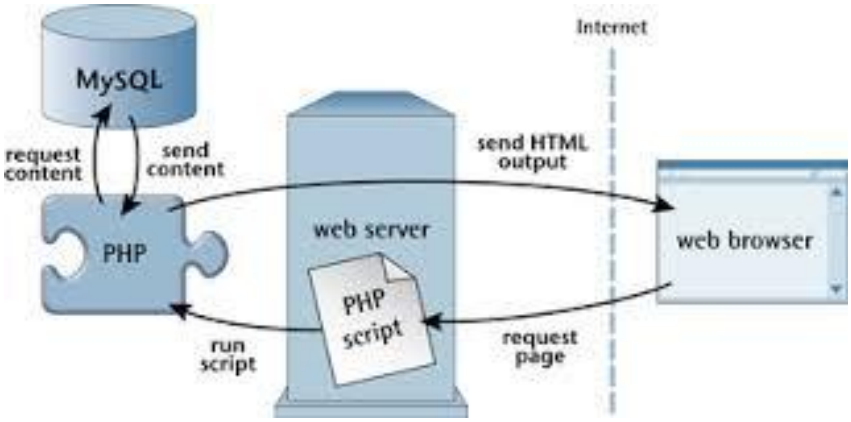


Fig: Program execution process

PHP includes & require statements:

It is possible to insert the content of one PHP file into another PHP file (before the server executes it), with the include or require statement.

- require will produce a fatal error (E_COMPILE_ERROR) and stop the script
- include will only produce a warning (E_WARNING) and the script will continue

So, if you want the execution to go on and show users the output, even if the include file is missing, use the include statement. Otherwise, in case of Framework, CMS, or a complex PHP application coding, always use the require statement to include a key file to the flow of execution. This will help avoid compromising your application's security and integrity, just in-case one key file is accidentally missing.

Including files saves a lot of work. This means that you can create a standard header, footer, or menu file for all your web pages. Then, when the header needs to be updated, you can only update the header include file.

PHP Variables: In PHP, a variable starts with the $ sign, followed by the name of the variable:

Rules for PHP variables:

- A variable starts with the $ sign, followed by the name of the variable

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive ($age and $AGE are two different variables.

PHP echo and print Statements:

echo and print are more or less the same. They are both used to output data to the screen.

The differences are small: echo has no return value while print has a return value of 1 so it can be used in expressions. echo can take multiple parameters (although such usage is rare) while print can take one argument. echo is marginally faster than print.

PHP Functions:

The real power of PHP comes from its functions.PHP has more than 1000 built-in functions, and in addition you can create your own custom functions. Besides the built-in PHP functions, it is possible to create your own functions.

- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute automatically when a page loads.
- A function will be executed by a call to the function.

PHP MySQL Database:

With PHP, you can connect to and manipulate databases. MySQL is the most popular database system used with PHP.

What is MySQL?

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications

- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use
- MySQL is developed, distributed, and supported by Oracle Corporation
- MySQL is named after co-founder Monty Widenius's daughter: My

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows.

PHP + MySQL Database System

A query is a question or a request. We can query a database for specific information and have a record setreturned. Look at the following query (using standard SQL):

SELECT LastName FROM Employees

The query above selects all the data in the "LastName" column from the "Employees" table.

PHP Connect to MySQL:

- **MySQLi extension** (the "i" stands for improved)
- **PDO (PHP Data Objects)**

Earlier versions of PHP used the MySQL extension. However, this extension was deprecated in 2012.

PHP - AJAX Introduction:

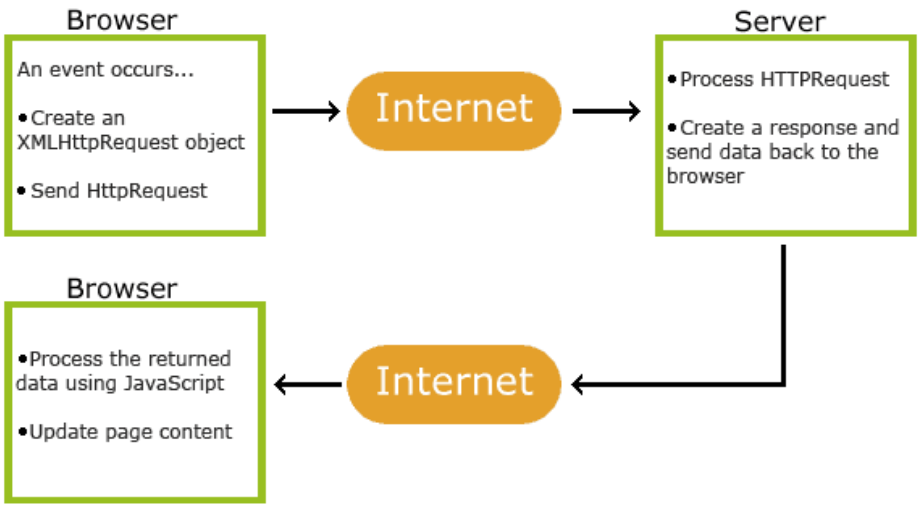
AJAX = Asynchronous JavaScript and XML.

AJAX is a technique for creating fast and dynamic web pages.

AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page. Classic web pages, (which do not

use AJAX) must reload the entire page if the content should change. Examples of applications using AJAX: Google Maps, Gmail, YouTube, and Facebook tabs.

How AJAX Works



XAMMP: XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl. XAMPP is really very easy to intall and to use –just download, extract and start.

XAMPP for Windows: The distribution for Windows 2000, 2003 , XP, Vista, 7, 8 and 10. This version contains: Apache, MySQL, PHP + PEAR, Perl, mod_Php, mod_ssl, OpenSSL PhpMyAdmin.

Webalizer, Mercury Mail Transport System for Win32 and NetWare System v3.32, Ming, FileZilla FTP Server, mcrypt, eAccelerator, SQLite, and WEB-DAV + mod_auth_mysql.

1. Apache v3.2.2
2. MySQL 5.5.32
3. PHP 5.4.16
4. phpMyAdmin 4.0.4
5. FileZilla FTP Server 0.9.41
6. Tomcat 7.0.41
7. Strawberry Perl 5.16.3.1 Portable

MySQL Workbench:MySQL Workbench is a unified visual tool for database architects, developers, and DBAs, MySQL development, and comprehensive administration tools for server configuration, user administration, and much more. MySQL Workbench is available on Windows, Linux and Mac OS. MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. It includes everything a data modeler needs ER models, forward and reverse engineering, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort. MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries. The SQL Editor provides color syntax highlighting reuse of SQL snippets, and execution history of SQL. The database Connections Panel enables developers to easily manage database connections. The Object Browser provides instant access to database schema and objects. MySQL Workbench provides a visual console to easily administer MySQL environments and gain better visibility into databases. Developers and DBAs can use the visual tools for configuring servers, administering users, and viewing database health.

5.3 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, they 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. They 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, this 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 the programming environment, they 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, "SQL-92" 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. They 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. They may study the source code and change it to suit for needs. The MySQL software uses the GPL (GNU General Public License), http://www.fsf.org/licenses/, to define what they may and may not do with the software in different situations. If they feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, they can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information (http://www.mysql.com/company/legal/licensing/).

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use**

If that is what they are looking for, they should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside the other applications, web servers, and so on, requiring little or no attention. If they dedicate an entire machine to MySQL, they can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together. They can find a performance comparison of MySQL Server with other database managers on benchmark page. MySQL Server was originally developed to handle large databases much faster than existing solutions and has been

successfully used in highly demanding production environments for several years. Although under constant development.

- **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).

They also provide MySQL Server as an embedded multi-threaded library that they can link into the application to get a smaller, faster, easier-to-manage standalone product.

5.4Sublime Text 3

Sublime Text is an integrated development environment (IDE) used in computer programming, and is the most widely used PHP IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Sublime is written mostly in PHP and HTML and its primary use is for developing PHP applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, COBOL, D, Fortran, Haskell, JavaScript, Julia, JAVA, Lasso, Lua, NATURAL, Perl, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, Clojure, Groovy, Scheme, and Erlang.

SublimeText is a shareware cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

Sublime Text Editor Some features:

CROSS PLATFORM

- Sublime Text is available for Mac, Windows and Linux. One license is all you need to use Sublime Text on every computer you own, no matter what operating system it uses.

- Sublime Text uses a custom UI toolkit, optimized for speed and beauty, while taking advantage of native functionality on each platform.

GOTO ANYTHING

Use *Goto Anything* to open files with only a few keystrokes, and instantly jump to symbols, lines or words.

Triggered with Ctrl+P, it is possible to:

- Type part of a file name to open it.
- Type @ to jump to symbols, # to search within the file, and : to go to a line number.

These shortcuts can be combined, so tp@rf may take you to a function read_file within a file text_parser.py. Similarly, tp:100 would take you to line 100 of the same file.

MULTIPLE SELECTIONS

Make ten changes at the same time, not one change ten times. Multiple selections allow you to interactively change many lines at once, rename variables with ease, and manipulate files faster than ever.

Try pressing Ctrl+Shift+L to split the selection into lines and Ctrl+D to select the next occurrence of the selected word. To make multiple selections with the mouse, take a look at the Column Selection documentation.

POWERFUL API AND PACKAGE ECOSYSTEM

Sublime Text has a powerful, Python API that allows plugins to augment built-in functionality.

Package Control can be installed via the command palette, providing simple access to thousands of packages built by the community.

SPLIT EDITING

Get the most out of your wide screen monitor with split editing support. Edit files side by side, or edit two locations in the one file. You can edit with as many rows and columns as you wish. Take advantage of multiple monitors by editing with multiple windows, and using multiple splits in each window.

INSTANT PROJECT SWITCH

Projects in Sublime Text capture the full contents of the workspace, including modified and unsaved files. You can switch between projects in a manner similar to *Goto Anything*, and the switch is instant, with no save prompts - all your modifications will be restored next time the project is opened.

PERFORMANCE

Sublime Text is built from custom components, providing for unmatched responsiveness. From a powerful, custom cross-platform UI toolkit, to an unmatched syntax highlighting engine, Sublime Text sets the bar for performance.

CUSTOMIZE ANYTHING

Key bindings, menus, snippets, macros, completions and more - just about everything in Sublime Text is customizable with simple JSON files. This system gives you flexibility as settings can be specified on a per-file type and per-project basis.

6. SAMPLE CODE

Source Code For User Registration User Login :

```
<!DOCTYPE html>

<html>

<head>

<title>login page nerw</title>

<linkrel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.12.1/css/all.min.css">

<style>

.container{

padding:8% 14%;

margin: auto;

}

body{

background:#f8f8f8;
```

```
}
.box{
width: 100px;
height: 100px;
background: red;
position :relative;
left: 0px;
border-radius:50%;
color:white;
text-align:center;
top:-40px;
z-index: 1;
}
.box2{
width: 350px;
height: 450px;
background: red;
position :relative;
left: 50px;
color:white;
text-align:center;
```

```
top:-120px;
background: white;
box-shadow: 2px 3px 4px #CCC;
z-index: 0;
}
.icon{
font-size:54px;
line-height:80px;
font-weight:bold;
transition:ease all 0.1s;
}
.move{
animation: mymove 0.8s forwards;
}
.iconclose{
transform:rotate(45deg);
position :relative;
background:green;
padding:20px;
border-radius:50%;
width:100px;
```

```
height:100px;

line-height:50px;

left:-30px;

top:-20px;

text-align:center;

cursor:pointer;

z-index:1500;color:white;

}

/* Standard syntax */

@keyframes mymove {

0% {left: 0px;border-radius:50%;width:100px;height:100px;}

20% {border-radius:50%;}

60% {left: 50px;border-radius:0%;width:450px;height:550px;}

100% {left: 50px;border-radius:0%;width:350px;height:450px;}

}

#login{

padding:12% 0%;

color:red;

}

h3{

border-right:10px solid red;
```

```
}

input{

font-size: 16px;

padding: 12px;

width: 65%;

margin:6% 0%;

outline: none;

border:none;

/* border-bottom: 2px solid red; */

transition: ease all 0.8s;

background: #d6d6d6;

margin-left: -5px;

}

input:focus{

border: black;

color:white;

background: #313131;

transition: ease all 0.8s;

width: 80%;

box-shadow: inset 0px 1px 4px 4px #040404;

}
```

```
.usericon{
position: relative;
left:5px;
font-size: 16px;
background: red;
color:white;
padding: 12px;
margin: 0px;
}
a{
color:red;
}
.btnedesigns{
outline: none;
background: red;
border-radius: 20px;
border:none;
width: 80%;
padding: 10px;
color: white;
font-weight: bold;
```

```
margin: 10% 0%;transition: ease all 0.8s;
}
.btnedesigns:hover{
background: green;
transition: ease all 0.8s;
}
#signup{
display: none;
margin-top: -50px;
}
#signup .usericon{
background: black;
color:white;
}
#signup input{
background: #ccc;
margin:2% 0%;
}
#signup .btnedesigns{
background: black;
}
```

```
#signup h3{
border-right:10px solid white;
}
#signup input:focus{
box-shadow: inset 0px 1px 4px 4px #c3c3c3;
}
</style>
</head>
<script>
$(document).ready(function(){
$(".icon").click(function(){
$(".box").toggleClass("move");
$(".icon").toggleClass("iconclose");
$(".box2").toggle();
$("#signup").toggle();
});
});
</script>
<body>
<div class="container">
<div class="box">
```

```
<div class="icon">+</div>

<form id="signup" method="post">

<h3>SIGNUP</h3>

<hr/>

<div>

<span class="fa fa-user usericon" ></span>

<input type="text" name="user" placeholder="Enter Your User Name"/>

</div>

<div>

<span class="fa fa-envelope usericon"></span>

<input type="email" name="email" placeholder="Enter Your Email Id"/>

</div>

<div>

<span class="fa fa-key usericon"></span>

<input type="password"name="pwd" placeholder="Enter Your Password"/>

</div>

<div>

<span class="fa fa-key usericon"></span>

<input type="password" placeholder="Confirm Password"/>

</div>

<input name="signup" type="submit">signup <span class="fa fa-key"></span></input>
```

```
</form>

</div>

<div class="box2">

<form id="login" method="post" >

<h3>LOGIN</h3>

<hr/>

<div>

<span class="fa fa-user usericon"></span>

<input type="text" name="username" id="1" placeholder="Enter Your User Name"/>

</div>

<div>

<span class="fa fa-key usericon"></span>

<input type="password" name="password" id="2" placeholder="Enter Your Password"/>

</div>

<a href="#" title="forgot Password">Forgot Your Password</a>

<button class="btnedesigns" name="login" type="submit">LOGIN <span class="fa fa-
key"></span></button>

</form>

</div>

</div>

</body>

</html>
```

```
<script>
$(".btnedesigns").click(function(e) {
username = $("#1").val();
username = username.trim();
password = $("#2").val();
password = password.trim();
if (username=="") {
$("#1").css("border","1px solid #f00");
$("#1").focus();
e.preventDefault();
}
else if (password=="") {
$("#2").focus();
$("#2").css("border","1px solid #f00");
e.preventDefault();
}
})
</script>
```

Source Code For Order Items:

```
<style>
.dropdown-toggle{
```

```
margin-top:100px;

}

</style>

<header id="header">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<a href="cart.php" class="navbar-brand">

<h3 class="px-2">

Restaurant Name

</h3>

</a>

<button class="navbar-toggler"

type="button"

data-toggle="collapse"

data-target = "#navbarNavAltMarkup"

aria-controls="navbarNavAltMarkup"

aria-expanded="false"

aria-label="Toggle navigation"

>

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavAltMarkup">
```

```
<div class="mr-auto"></div>

<div class="navbar-nav">

<a href="billing.php" class="nav-item nav-link active">

<h5 class="px-2 cart">

<i class="fa fa-shopping-cart" aria-hidden="true"></i> Cart

<span class="CartCount">0</span>

</h5>

</a>

<div class="navbar-nav">

<ul class="nav navbar-nav">

<!-- <li class=""><a href="#" class="nav-item nav-link active">Home</a></li> -->

<li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown"
href="#">Categories <span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="cart.php?Category=U291dGggSW5kaWFu">South Indian</a></li>

<li><a href="cart.php?Category=Tm9ydGggSW5kYWlu">North Indain</a></li>

<li><a href="cart.php?Category=Q2hpbmVzZQ==">Chinese</a></li>

<li><a href="cart.php?Category=U25hY2tz">Snacks</a></li>

<li><a href="cart.php?Category=SGltYWxheWFuIEZvb2Q=">Himalayan Food</a></li>

<li><a href="cart.php?Category=bWFuY2h1cmlh">manchuria</a></li>

</ul>
```

```
</li>

?>

</ul>

</div>

<div class="navbar-nav">

<ul class="nav navbar-nav">

<!-- <li class=""><a href="#" class="nav-item nav-link active">Home</a></li> -->

<li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" href="#">Food
Type <span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="cart1.php?type=veg">veg</li></a>

<li><a href="cart1.php?type=non_veg">non_veg</li></a>

</ul>

</div>

<div class="navbar-nav">

<a href="logout.php" class="nav-item nav-link active">

<h5 class="px-2 user_login">

<i class=" fas fa-sign-out" style="color: red"></i>logout <span></span>

</h5>

</a>

</div>
```

```
</div>

</div>

</nav>

</header>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width,user-scalable =no,initial-scale=1.0,maximum-scale=1.0,minimum-scale=1.0">

<meta http-equiv="x-ua-compatible" content="ie=edge">

<title>addto cart</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.12.1/css/all.min.css">

<style type="text/css">

img{

max-width: 100%;

height: auto;
```

```
background: radial-gradient(white 30%,lightblue 70%);
}
.fa-star,
.fa-star-half{
color: yellowgreen;
padding: 3% 0;
}
#cart_count{
text-align: center;
padding: 0 0.9rem 0.1rem 0.9rem;
border-radius: 3rem;
}
.shopping-cart{
padding: 3% 0;
}
.cart-items + .cart-items{
padding: 2% 0;
}
.price-details h6{
padding: 3% 2%;
}
```

```
.mb-60 {
margin-bottom:5px;
}
.section-title p {
font-size: 24px;
font-family: Oleo Script;
margin-bottom: 0px;
margin-top:50px;
}
.section-title h4 {
font-size: 40px;
text-transform: capitalize;
color: #FF5E18;
position: relative;
display: inline-block;
padding-bottom: 25px;
}
.section-title h4::before {
width: 80px;
height: 1.5px;
bottom: 0;
```

```
left: 50%;

margin-left: -40px;

}

.section-title h4::before, .section-title h4::after {

position: absolute;

content: "";

background-color: #FF5E18;

}

</style>

</head>

<body>

<div class="container" style="background-color:#">

<div class="col-xl-12 mb-60">

<div class="section-title text-center">

<p>Deliciousness jumping into the mouth</p>

<h4>our menu</h4>

</div>

</div>

<div class="row text-center py-5">

<div class="col-md-3 col-sm-6 my-3 my-md-0">

<div class="card shadow" style="margin-bottom:25px;border: 1px solid black;">
```

```
<div>

<img class="img img-responsive" style="width:258px; height:120px; border-radius: 1px;" src="uploads/images/7.jpg">

</div>

<div class="card-body" style="height: 250px;">

<h5 class="card-title">VEG BRIYANI</h5>

<h6><i class="fas fa-star"></i><i class="fas fa-star"></i><i class="fas fa-star"></i><i class="fas fa-star"></i><i class="far fa-star"></i></h6>

<p class="card-text">

example one product..!

</p>

<h5>

<small><s class="text-secondary">$100</s></small>

<span class="price">$200</span>

</h5>

<button type="submit" name="add" class="btn AddToCart btn-warning my-3" value="1">add to cart <i class="fas fa-shopping-cart"></i></button>

</div>

</div>

<!-- </form> -->

</div>

</div>
```

```
</div>

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>

<scriptsrc="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<script>

$(document).ready(function() {

$(".AddToCart").click(function(){

item_id = $(this).val();

$.ajax({

url:"ajax.php",

method:"POST",

data:{"item":item_id,"action":"AddToCart" },

success: function(respose){

alert(respose)

if (respose=="Item Added") {
```

```
CartCount = $(".CartCount").text();

$(".CartCount").text(parseInt(CartCount)+1)

}

}

})

})

})

</script>

</body>

</html>
```

7. TESTING

7.1 Introduction

In general, software engineers distinguish software faults from software failures. In case of a failure, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failure. A fault can also be described as an error in the correctness of the semantic of a computer program. A fault will become a failure if the exact computation conditions are met, one of them being that the faulty portion of computer software executes on the CPU. A fault can also turn into a failure when the software is ported to a different hardware platform or a different compiler, or when the software gets extended. Software testing is the technical investigation of the product under test to provide stakeholders with quality related information.

System Testing and Implementation

The purpose is to exercise the different parts of the module code to detect coding errors. After this the modules are gradually integrated into subsystems, which are then integrated themselves too eventually forming the entire system. During integration of module integration testing is performed. The goal of this is to detect designing errors, while focusing the interconnection between modules. After the system was put together, system testing is performed. Here the system is tested against the system requirements to see if all requirements were met and the system performs as specified by the requirements. Finally accepting testing is performed to demonstrate to the client for the operation of the system.

For the testing to be successful, proper selection of the test case is essential. There are two different approaches for selecting test case. The software or the module to be tested is treated as a black box, and the test cases are decided based on the specifications of the system or module. For this reason, this form of testing is also called "black box testing".

The focus here is on testing the external behavior of the system. In structural testing the test cases are decided based on the logic of the module to be tested. A common approach here is to achieve some type of coverage of the statements in the code. The two forms of testing are complementary: one tests the external behavior, the other tests the internal structure.

Testing is an extremely critical and time-consuming activity. It requires proper planning of the overall testing process. Frequently the testing process starts with the test plan. This plan identifies all testing related activities that must be performed and specifies the schedule, allocates the resources, and specifies guidelines for testing. The test plan specifies conditions that should be tested; different units to be tested, and the manner in which the module will be integrated together. Then for different test unit, a test case specification document is produced, which lists all the different test cases, together with the expected outputs, that will be used for testing. During the testing of the unit the specified test cases are executed and the actual results are compared with the expected outputs. The final output of the testing phase is the testing report and the error report, or a set of such reports. Each test report contains a set of test cases and the result of executing the code with the test cases. The error report describes the errors encountered and the action taken to remove the error.

Testing Techniques

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and the output is evaluated to determine whether the program is performing as expected. In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

- **Black Box Testing**

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure or external database access
- Performance errors
- Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

- **White Box Testing**

In this, the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational.
- Execute internal data structures to ensure their validity.

Testing Strategies

- **Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

This System consists of 4 modules. Those are Reputation module, route discovery module, audit module. Each module is taken as unit and tested. Identified errors are corrected and executable unit are obtained.

- **Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

- **System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An

example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

- **Functional Testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes.

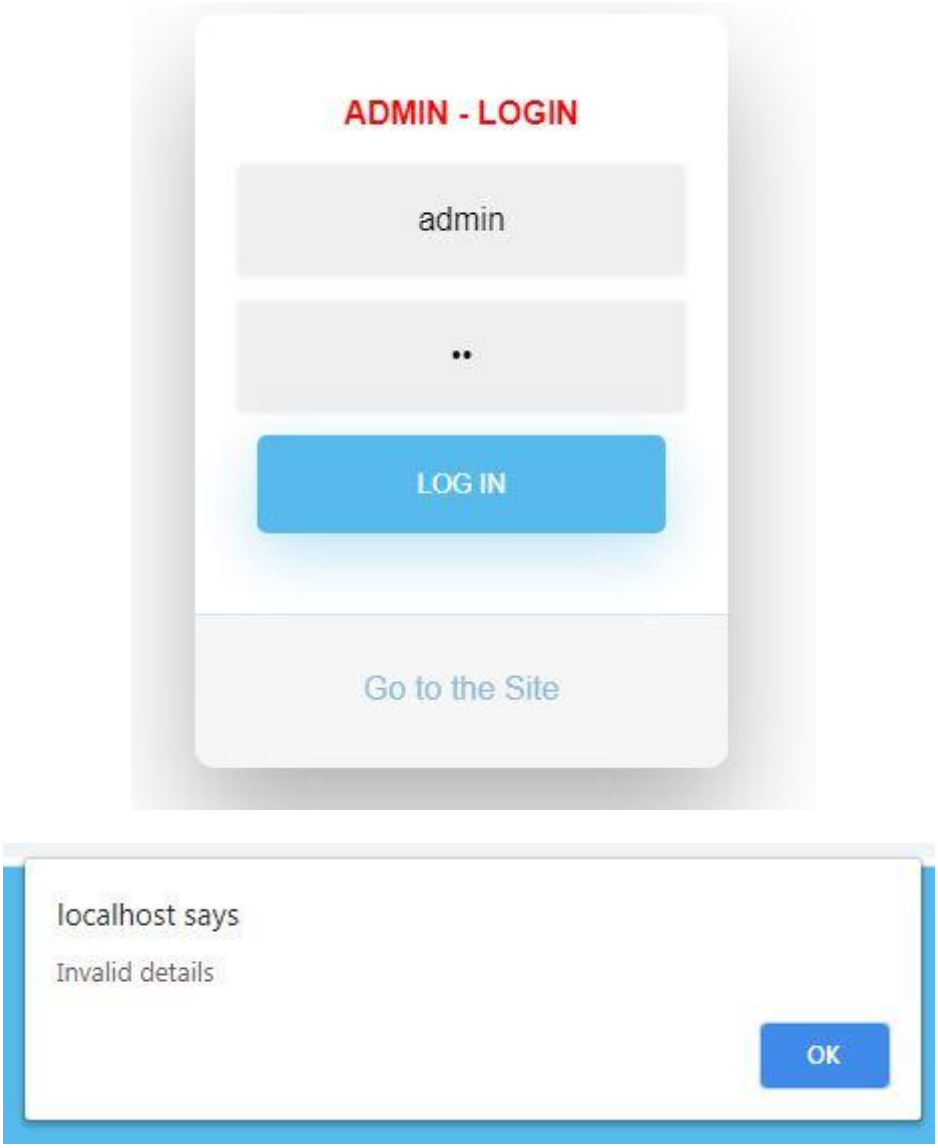
7.2 Sample Test Cases Specification

Test case id	**Test case name**	**Input**	**Expected Output**	**Observed Output**	**Result**
T1	admin login	Enter invalid Login id or password	Error message should be displayed	Error message is displayed.	Pass
T2	admin login	Enter valid login id and password	Admin login successful	Admin login successful	Pass
T3	user login	Enter invalid Login id or password	Error message should be displayed	Error message is displayed.	Pass
T4	user login	Enter valid login id and password	login success	user login success	Pass
T5	Add to cart	When click add to cart button	Item added	Item added	Pass
T6	Place Order	If items in cart page	Your order is placed	Your order is placed	Pass

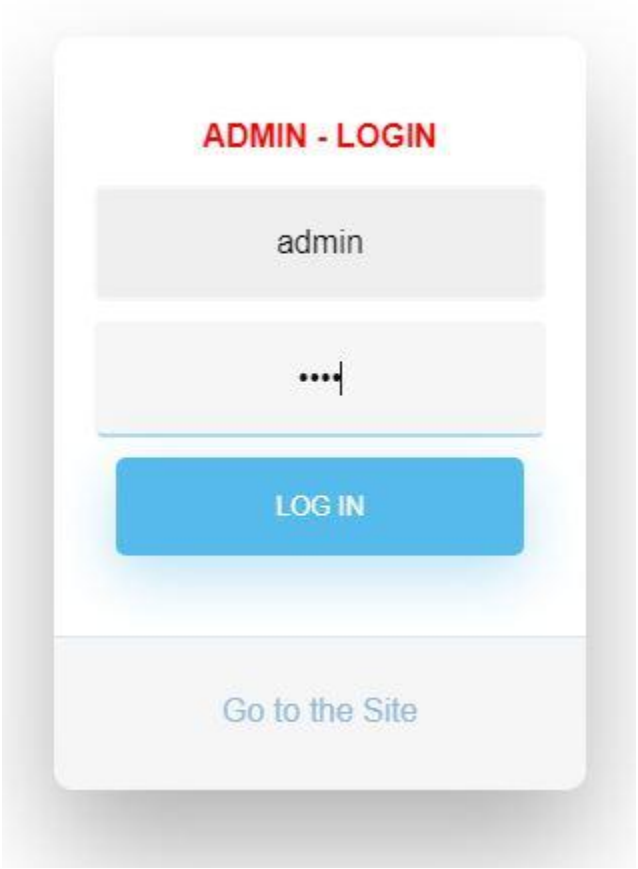
Table 7.2.1: Sample Test Cases Specification

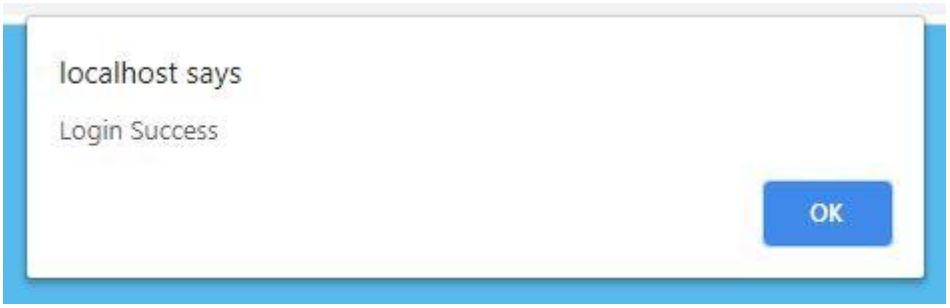
7.3 Test Cases screens

Test case T1



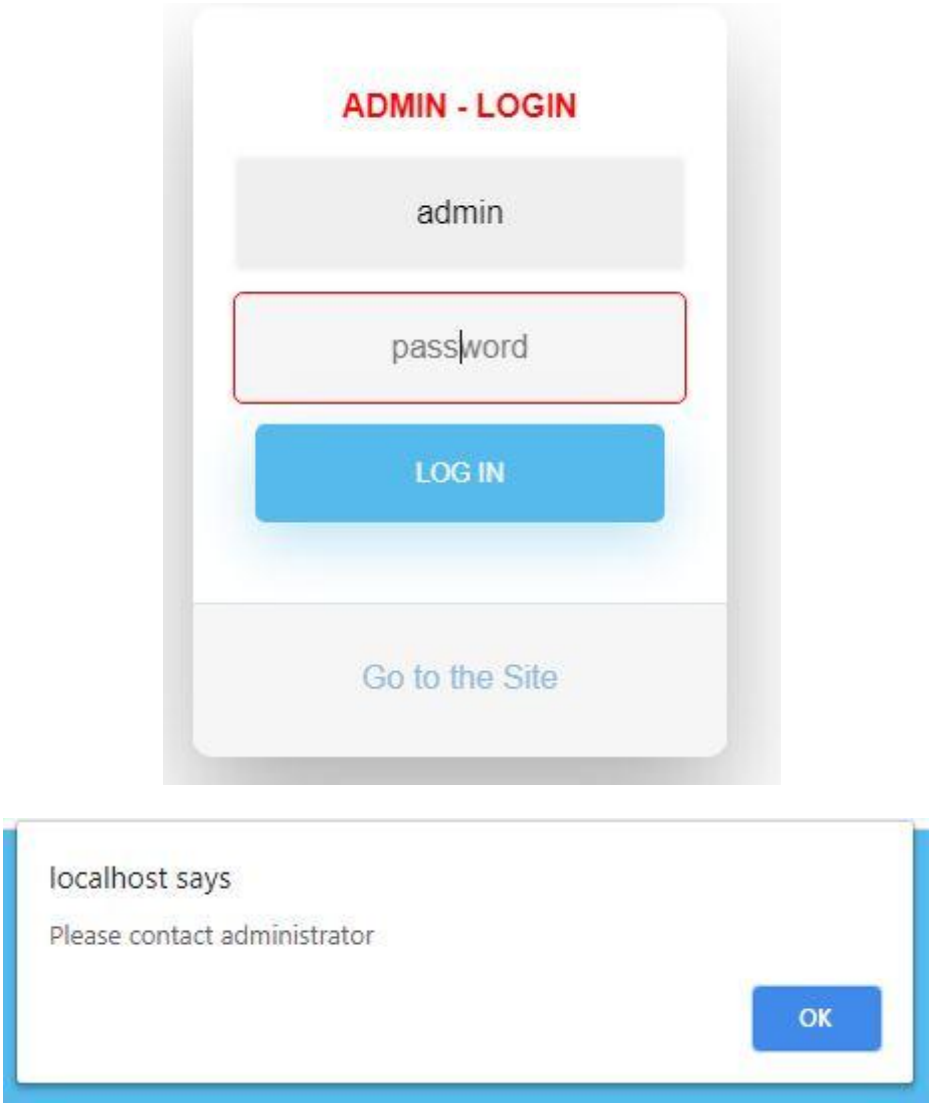
Test screen 7.3.1: Invalid admin login





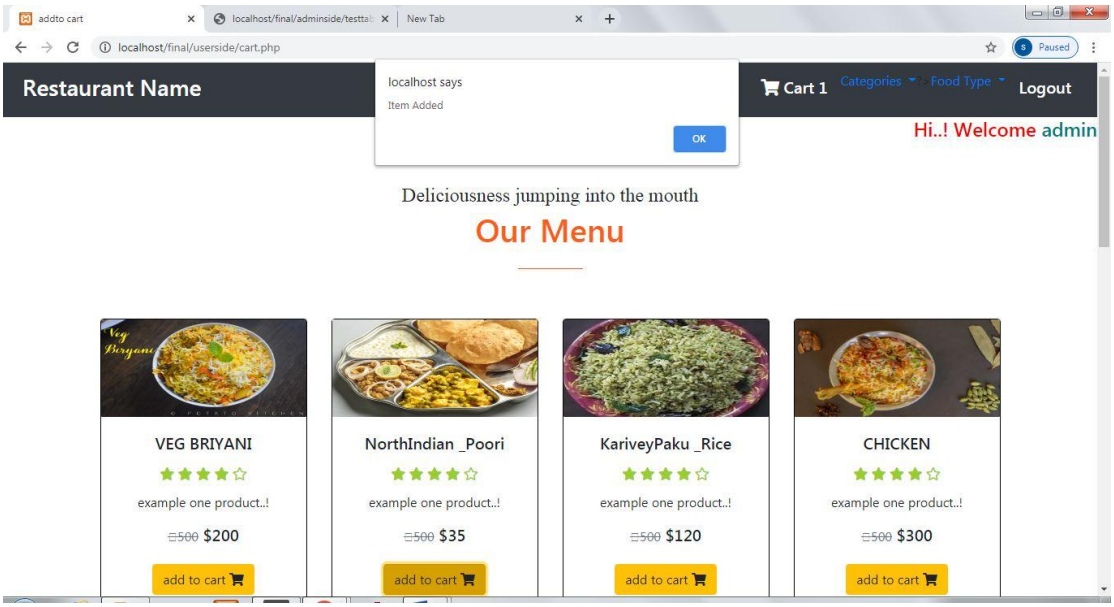
Test screen 7.3.2: Valid admin login

Test case T2



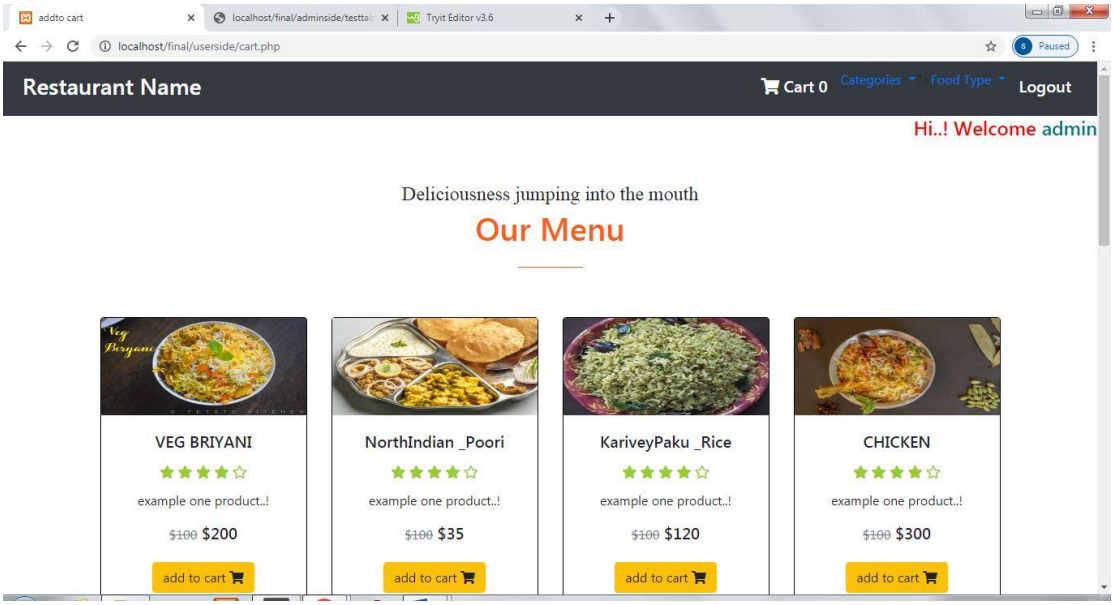
Test screen 7.3.3: Invalid user login

Test case T4



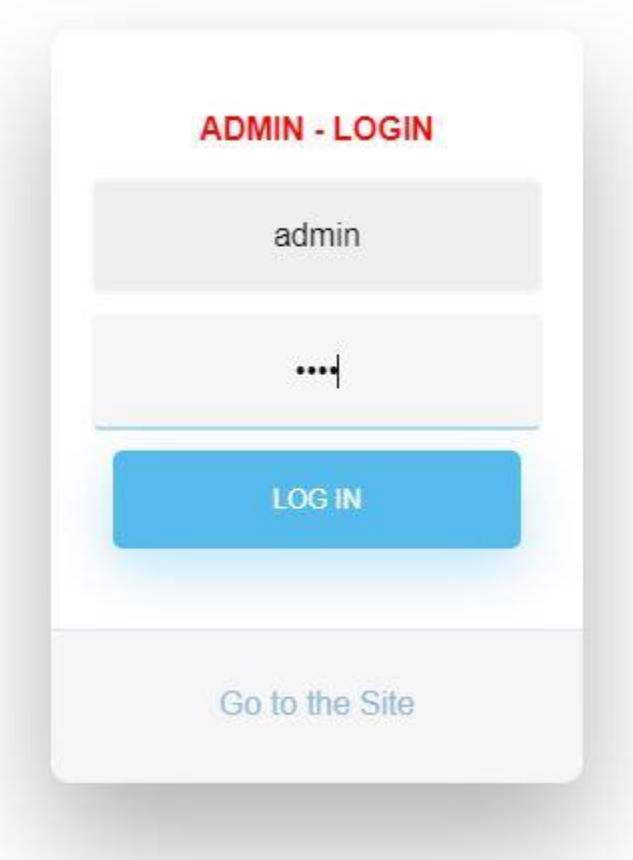
Test screen 7.3.4:Items Add to Cart page

8. SCREENSHOTS



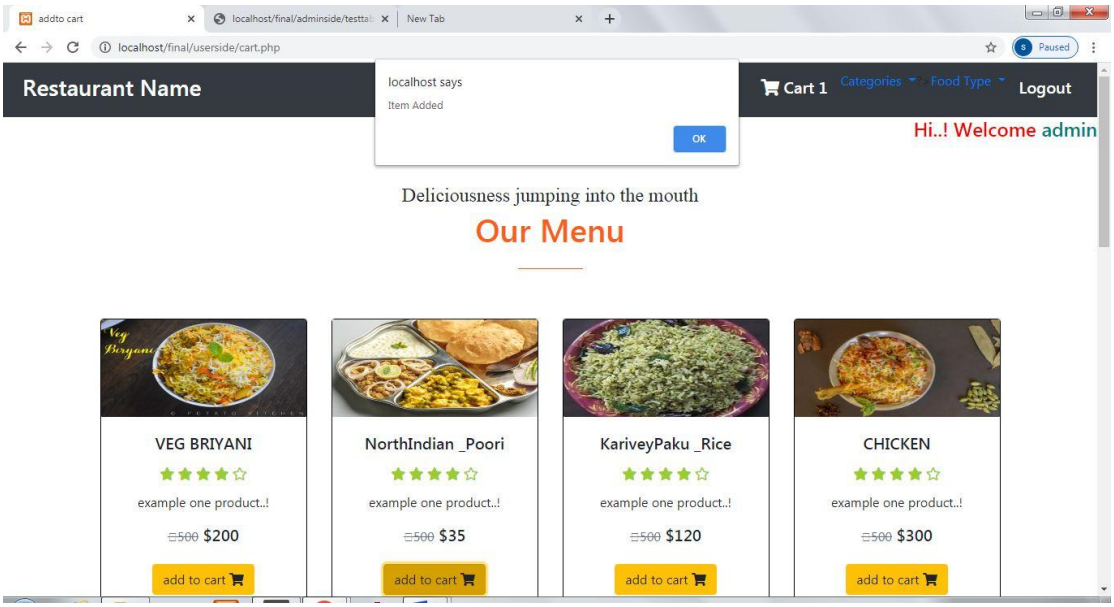
8.1 Screenshot forHome page

Description: The above screenshot displays the home page of this project. It shows the login details or entering process of the project. First user or admin registered in to the project then login with valid user id and password.



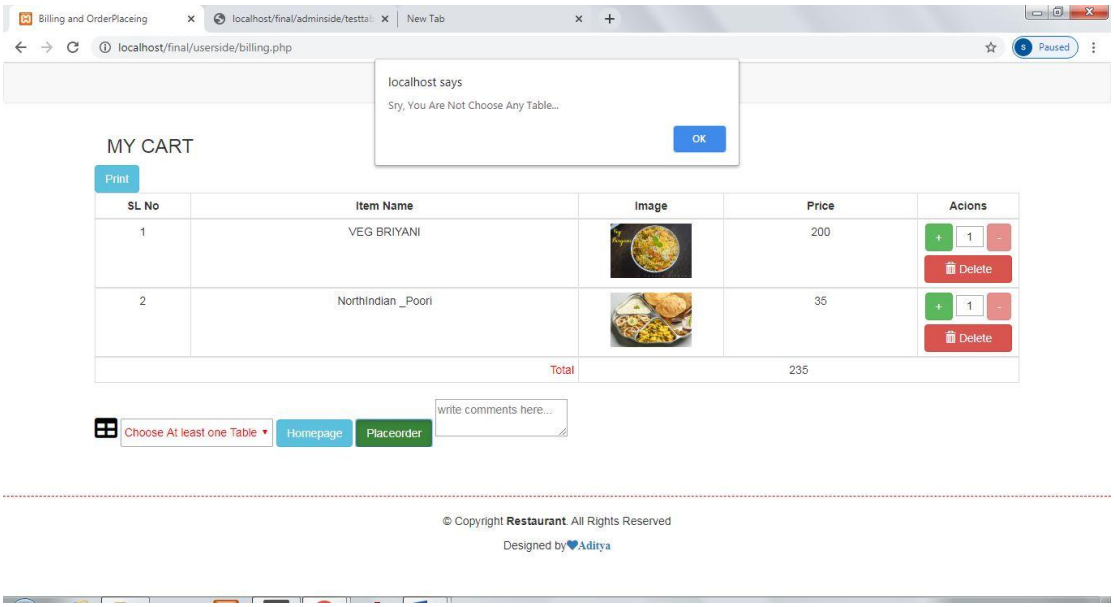
8.2 Screenshot for Admin login page

Description: The above screenshot displays the log in page of the admin. Admin can login with valid admin id with password.



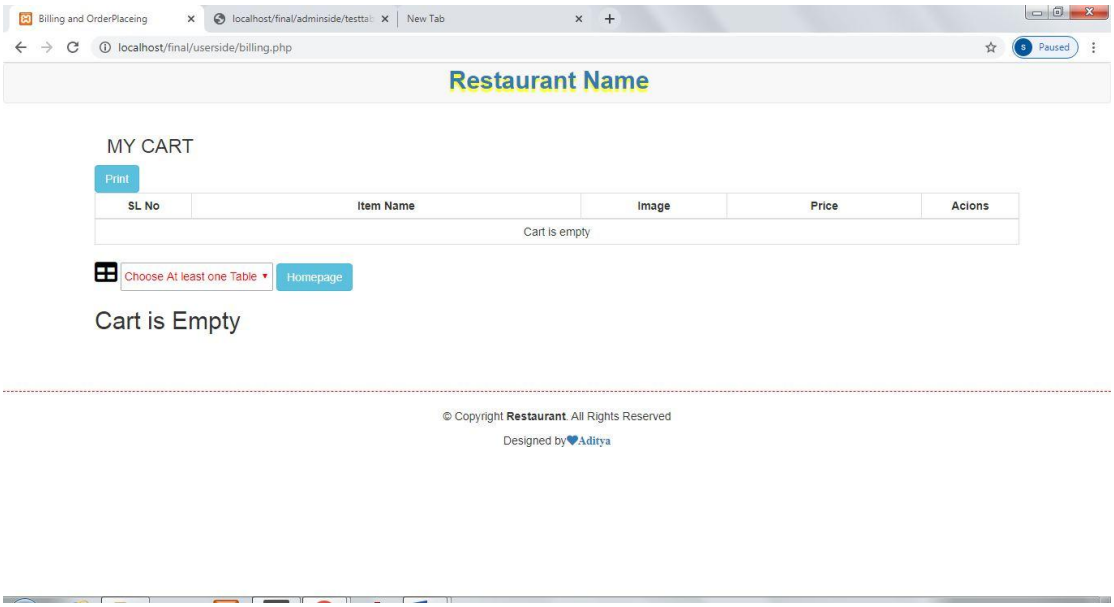
8.3 Screenshot for Admin Items Add to Cart page

Description: The above screenshot displays the Admin Items Add to cart page of this project. It shows the user added items details.



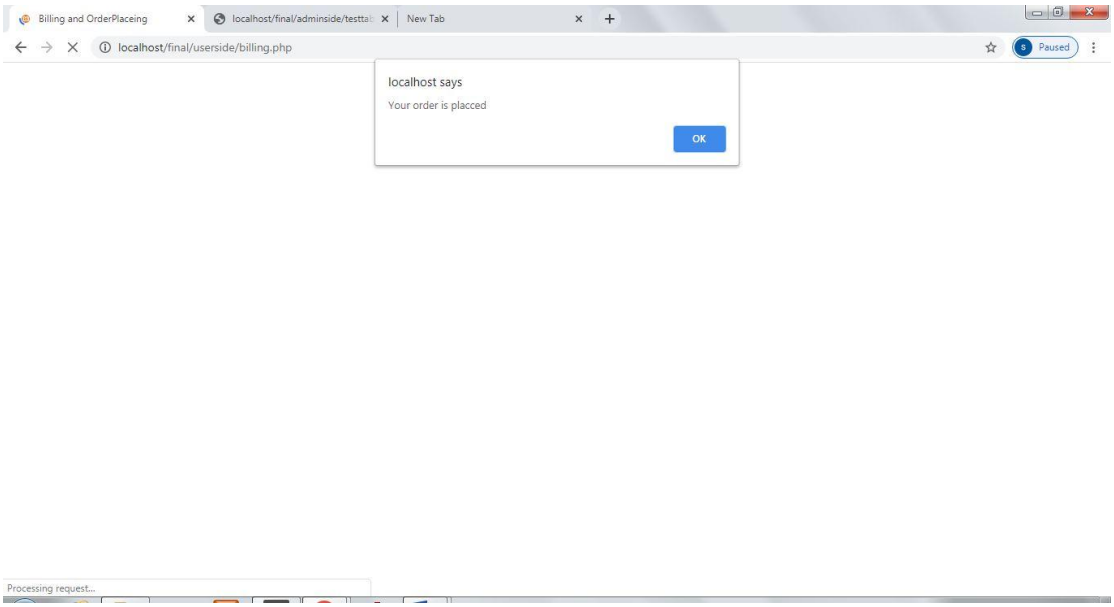
8.4 Screenshot for Cart List details.

Description: The above screenshot displays the Cart List. The user list contain Item Name ,Item Image, Item Price, Actions Like add quantity , remove Items From to cart.



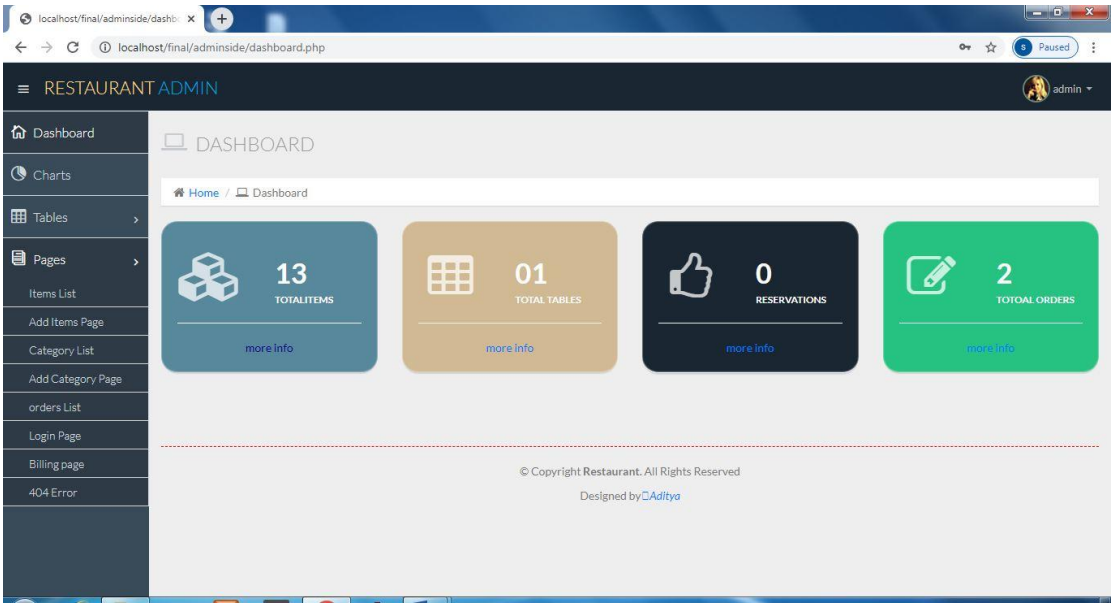
8.5 Screenshot for Cart details After placed Order.

Description: The above screenshot displays the user cart details after place order. It shows now Cart empty.



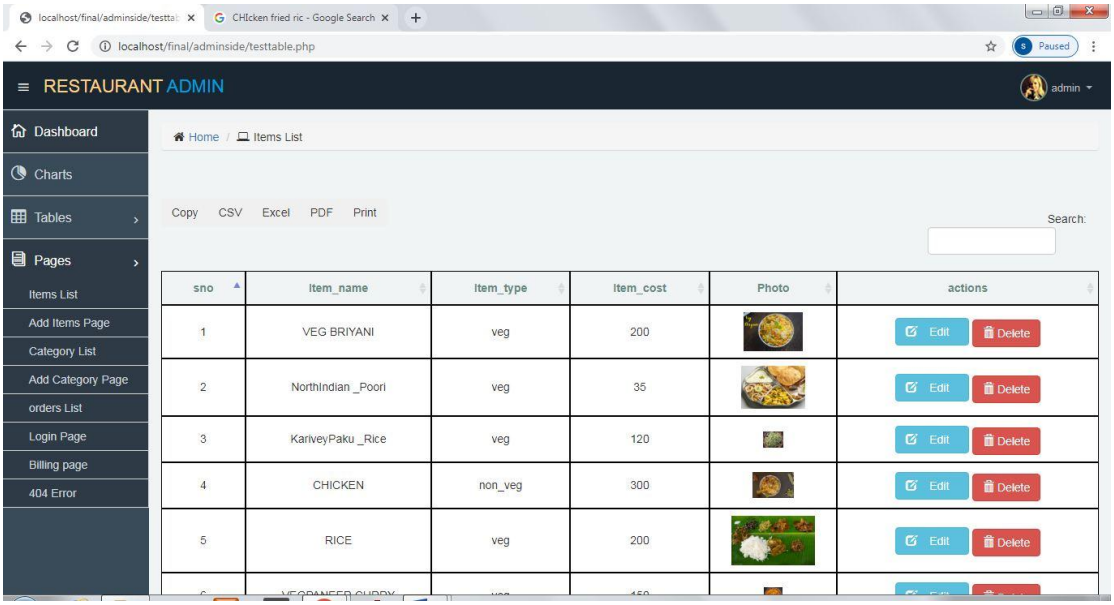
8.6 Screenshot for Order Status page

Description: The above screenshot displays the order status of the user. User can view the order is placed or not.



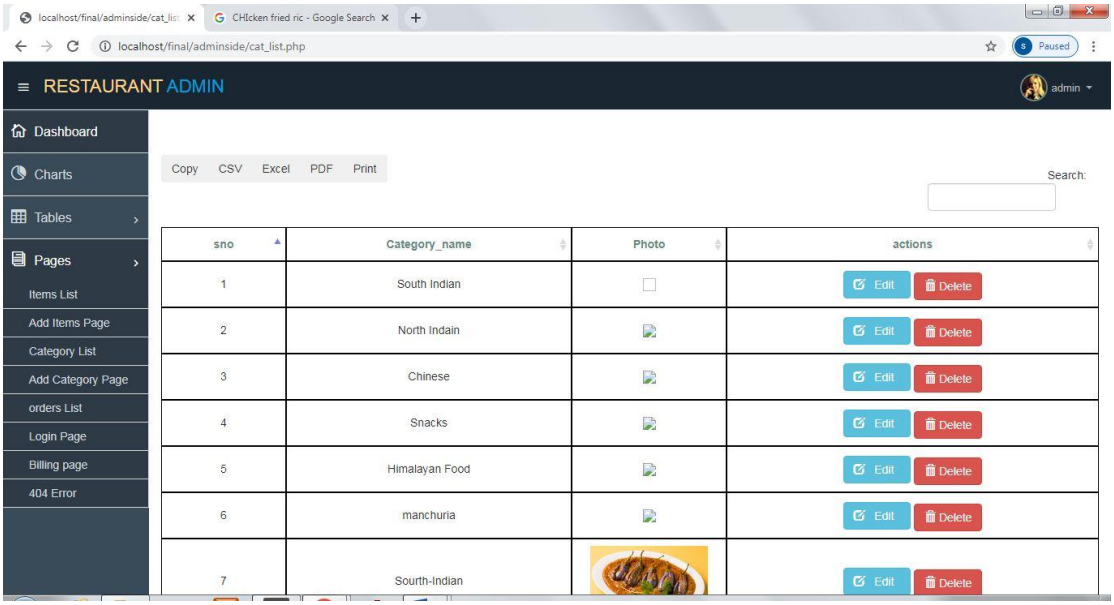
8.7 Screenshot for admin Dashboard page

Description: The above screenshot displays the Admin Dashboard page of this project. It shows how many total items, total reservations, total Orders, Total Table count. By click more you can full details of above each and every details as separately.



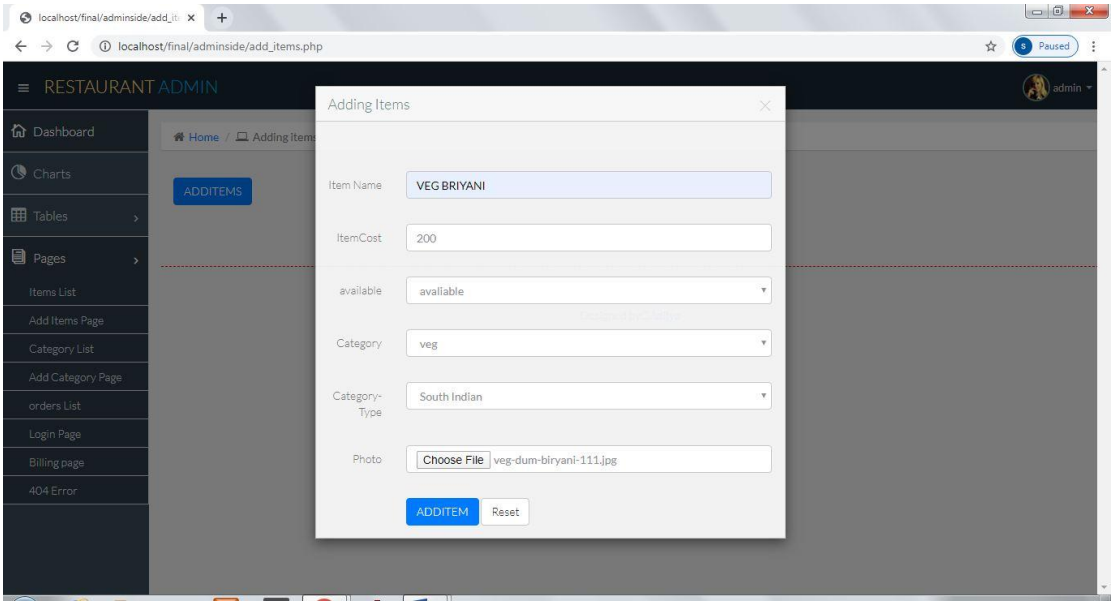
8.8 Screenshot for Total Items List

Description: The above screenshot displays Items List. In this admin can search required details by using context words and content words and also delete and Edit particular items.



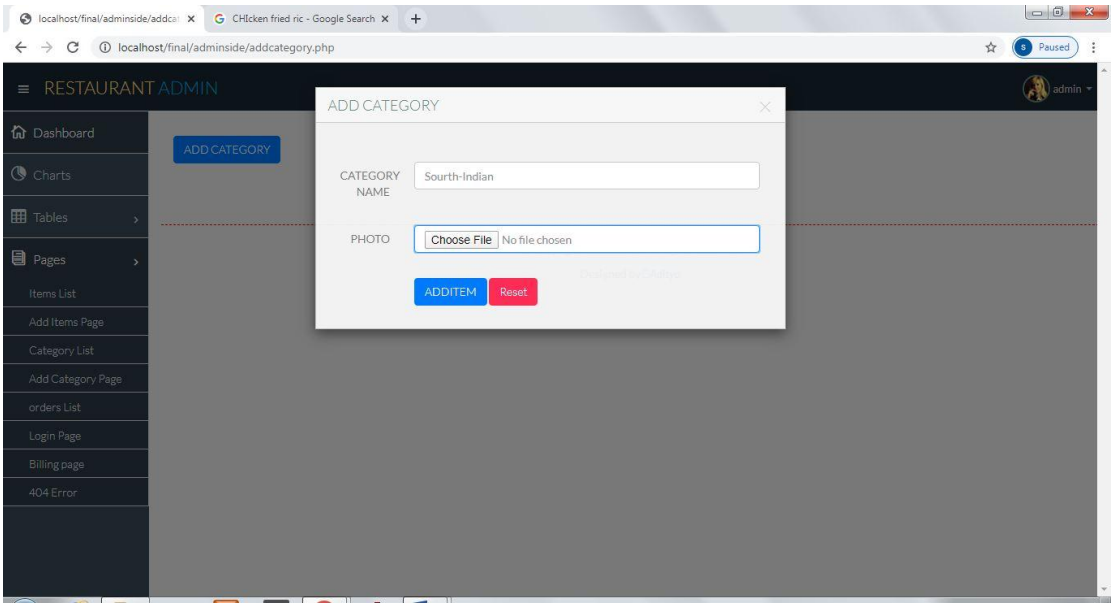
8.9 Screenshot for Category list.

Description: The above screenshot displays Item Category List. It shows available categories at Restaurant.



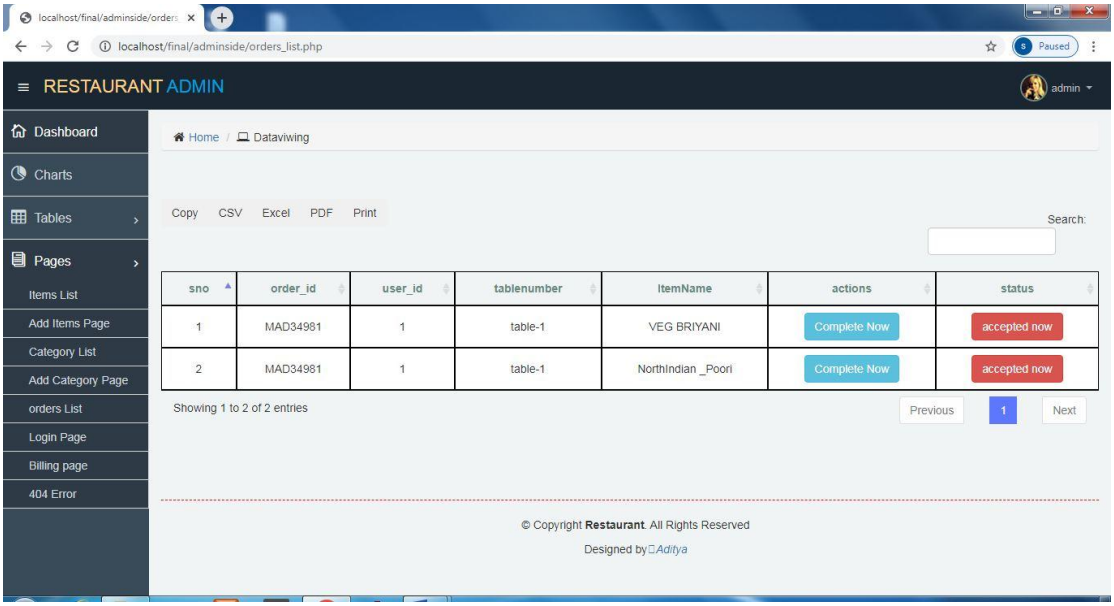
8.10 Screenshot for adding Items.

Description: The above screenshot displays add item to the list. In this user can fill required Item information.



8.11 Screenshot for Add Category.

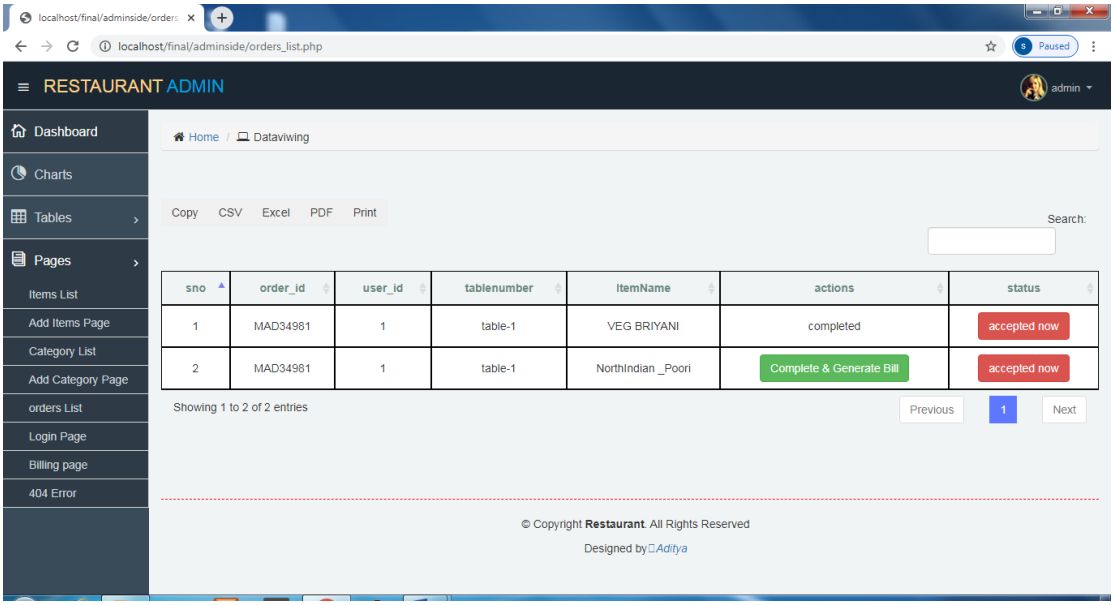
Description: The above screenshot displays Add Category. In this user can see the fill the require details.



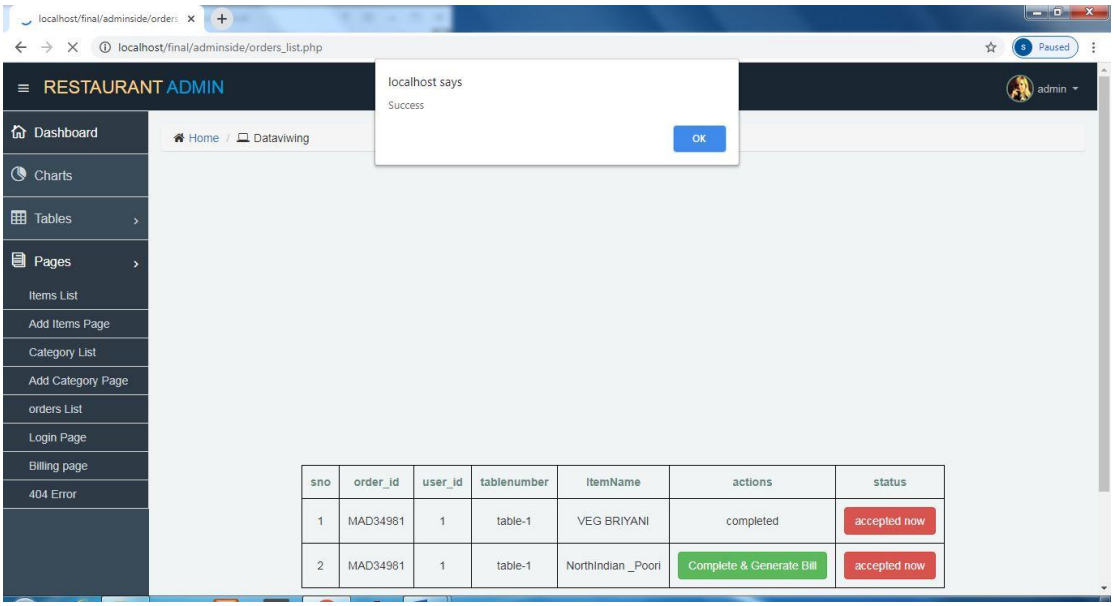
8.12 Screenshot for Orders List.

Description: The above screenshot displays Orders List. In this admin can see the all orders. It shows the action button like Complete now and status accepted Now . Those two actions are change the orders status.

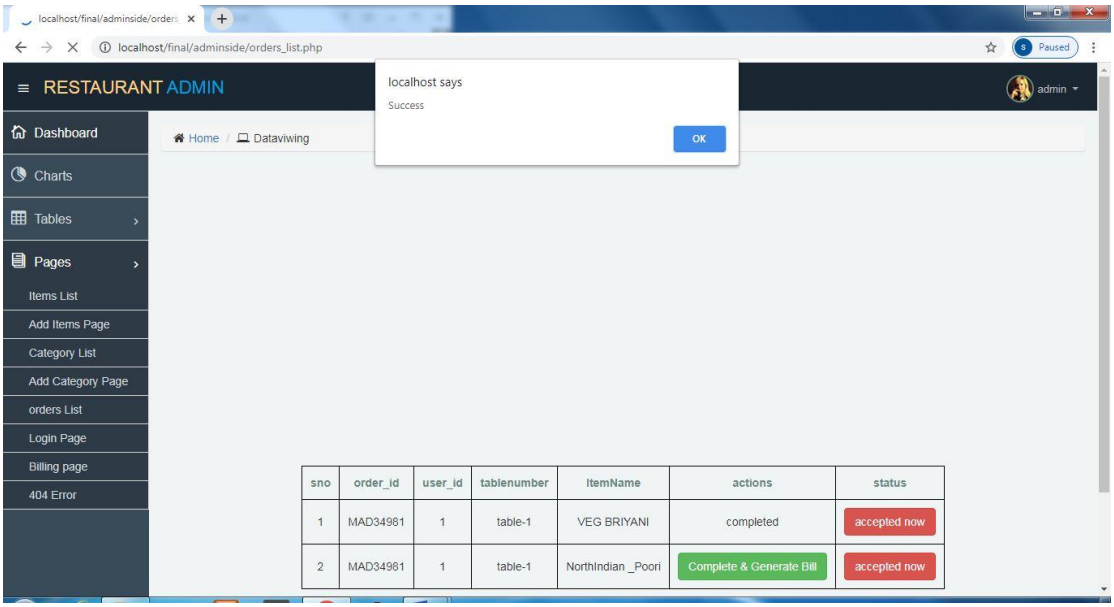
Suppose click the 'complete now' then its shows the completed and reaming same order id items fields status will turn to complete and Generate bill. This Functionality shown on bellow diagrams with clearly.

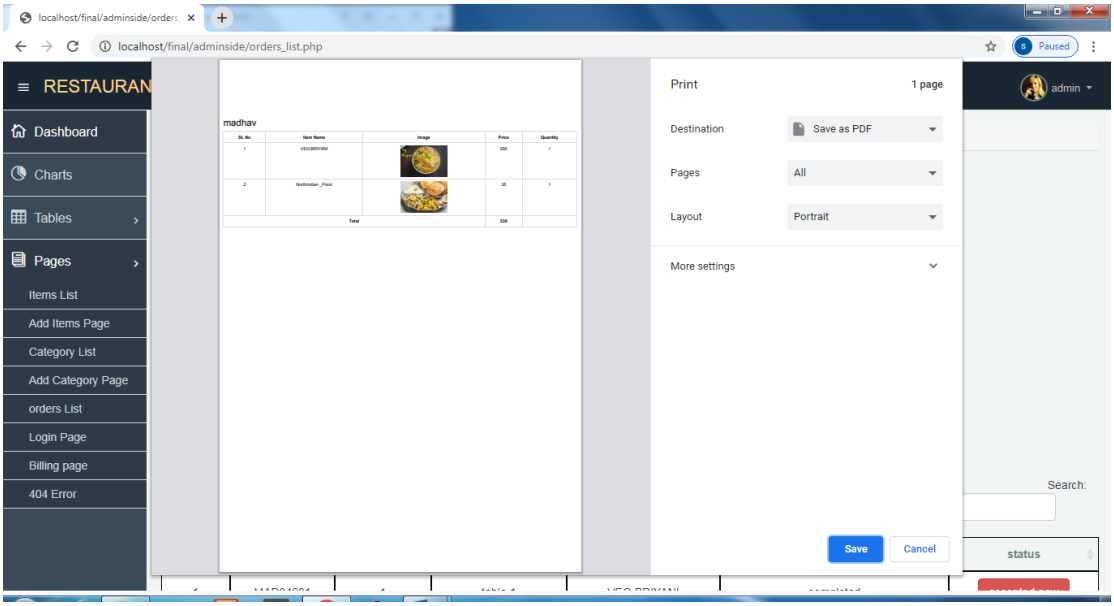


8 .12.1 Screenshot for Complete Now Action.



8.12.2 Screenshot for Complete Now & Generate Bill Action.





8.12.4 Screenshot for Generate Bill Action And Print Bill.

CONCLUSION

The proposed System provides a low cost, efficient, convenient and easy to use system for placing orders for food in hotels and restaurants. Now a day's people are very familiar with touch screen interface due to greater advancements in the field of technology. It will be easier for the users to navigate through the web pages by simply touching the display screen. The chances of errors are reduced and updating of menu and its prices can be done easily. It will be much comfortable and easier for the customers to place orders of their wish. This system is user-friendly and ensures good quality of services and customer satisfactions.

BIBLIOGRAPHY/REFERENCES

[1] Bhaskar Kumar Mishra, "Touch based digital ordering system on android using GSM and Bluetooth for restaurants", in international journal of emerging technology and advanced engineering. 2015.

[2] M. Z. H. NOOR, IEEE control and System Graduate Research Colloquium (2012) "The Development of Self Service restaurant ordering system."

[3] Wen Jiun Yap, "Design and Development of Multi-Touchable E-Restaurant Management System", 2010 International Conference on Science and Social Research (CSSR 2010), December 5 - 7, 2010 SOWNDARYA H K, ABHINAYA R, PRATHIBA B S," INTELLIGENT FOOD MENU ORDERING SYSTEM" International Research Journal of Engineering and Technology (IRJET) (2017)

[4] Sushmita Sarkar, Resham Shinde, Priyanka Thakare, "Integration of Touch Technology in Restaurants using Android", International Journal of Computer Science and Mobile Computing (2014).

[5] Ashutosh Bhargave, Niranja Jadhav, Apurva Joshi,"Digital Ordering System for Restaurant Using Android" International Journal of Scientific and Research Publications, Volume 3, Issue 4, April 2013. [6] Deeplakshmi Zingade, Mrunal Jadhav, Neha Rane, "Touch Based Digital Ordering System on Android", International Journal of Advance Engineering and Research Development (2017). [15] Ashwini Bankar, Mamta Mahajan ," Design of Intelligent Restaurant with a Touch Screen Based Menu Ordering System. ", IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) (2015). [7] Krushna A. Patil, Aakanksha P. Gawande, " ZIGBEE BASED HOTEL MENUCARD ORDERING SYSTEM", IJARIIE-ISSN(O)-2395-43962(2018)