

A Project Report
On

CANTEEN FOOD ORDERING MANAGEMENT SYSTEM

Submitted for partial fulfillment of the
requirements for the

BACHELOR OF COMPUTER APPLICATION

BY

Ashok Rawat (1706131794)

Under the guidance of

Prof. Parul Bhatnagar

Department of BCA



**CHAUDHARY CHARAN SINGH UNIVERSITY,
MERRUT**

CERTIFICATE

This is to certify that the project work entitled “**Canteen Food Ordering Management System**” is a bona fide work carried out by Ashok Rawat (1706131794) in partial fulfillment of the requirements for the award of degree of **BACHELOR OF COMPUTER APPLICATION** by **CCS UNIVERSITY, Meerut**, under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Name of Guide

Prof.Parul Bhatnagar

STUDENT'S DECLARATION

I hereby declare that the work being presented in this report entitled **Canteen Food Odering Managent System** is an authentic record of my / our own work carried out under the supervision of **Prof. Parul Bhatnagar**

The matter embodied in this report has not been submitted by me / us for the award of any other degree.

Dated:

Signature of students(s)
(Name- Ashok Rawat)
Department: BCA

This is to certify that the above statement made by the candidate(s) is correct to the best of my knowledge.

Signature of HOD

Signature of Supervisor

Date.....

ACKNOWLEDGEMENT

I would like to express my sincerest gratitude and indebtedness to the person who gave me a moral and technical support & whose kind assistance has been instrumental in completion to this industrial training. It gives me immense pleasure to own my humble gratefulness to my faculty **Prof. Parul Bhatnagar** for this indispensable guidance and providing necessary ideas and facilities to carry out this project.

I would like to place on record my best regards and deepest sense of gratitude to **Mr.JITENDRA KUMAR** (Department Head), **Prof. Parul Bhatnagar** (Project Guide), of college name for their careful and precious guidance which were extremely valuable for my study both theoretically and practically.

Signature of Student

Ashok Rawat

TABLE OF CONTENTS

	PAGE NO.
Certificate	
Declaration	
Acknowledgements	
List of Figures	
List of Tables	
Abstract	8
 CHAPTER 1	
INTRODUCTION	
Objectives	9
Problem specification	10
Scope of study	10
Limitations of the existing System	11
 CHAPTER 2	
REQUIREMENT	
ANALYSIS	
Functional requirement	12
Non-Functional requirement	12
 CHAPTER 3	
METHODOLOGY	
Justification of methodology	14
Description of methodology	15
 CHAPTER 4	
DESIGN AND DEEVELOPMENT	
Introduction	16
System requirement	16
Communication interfaces	16
Operating environment	17
System design	18
Logical model	18
System architecture	18

Entity relationship (E-R) diagram	19
Problem definition	21
CHAPTER 5	
APPLICATION	
Product features	23
Design & Implement constraints	23
User documentation	23
Assumption & dependencies	24
User interface description	24
CHAPTER 6	
TESTING	
Psychology of Testing	29
Testing Objectives	29
System Testing	29
Code Testing	30
Type of Testing	30
Integration Testing	31
Acceptance Testing	31
White Box Testing	31
black Box Testing	31
Criteria Satisfied by Test Cases	32
CHAPTER 7	
CONCLUSIONS &	
FUTURE	
1.1 Future works	33
CHAPTER 8	
CODING	35
REFERENCES	

LIST OF FIGURES

A list of figures with figure number, figure title and page number and a list of tables with table number, table name and page number should be listed after abstract in a separate page for each with roman numbers like ii, iii, etc.

LIST OF FIGURES

Figure 3.1	Waterfall life cycle model
Figure 4.1	Logical model
Figure 4.2	System architecture
Figure 4.3	Entity relationship diagram
Figure 5.1	Main Page
Figure 5.2	Login Page
Figure 5.3	Registration Page
Figure 5.4	Insert Page
Figure 5.5	View Records
Figure 5.6	Edit Update Records Page

LIST OF TABLES

LIST OF TABLES

Table 4.1	Hardware requirement
Table 4.2	Software requirement
Table 4.3	Technologies used

ABSTRACT

The project “Canteen Food Ordering Management System” is a system based on accessing the internet to book for orders in a canteen. The purpose of this study is to develop and implement an online food ordering system for canteens that will replace the manual method of ordering the foods. The previous system for ordering foods were faced with so many problems like, delay in processing the customer orders or paying for rooms that is below or beyond his standard, causes difficulty for emergency orders.

The objects-oriented analysis and design methodology (OOADM) was therefore used to analyse the system in order to discover the various objects involved and how they interact with one another so that a new and improved system can be defined.

The use of online view of food prices and uploading of available foods and facilities was used for the new system so that the customer can view and make his choice before arrival, and also in the case of emergency travelling. This new system assisted the hotel owners in managing their canteens, because they can also regulate the receptionist moves and avoid fraudulent activities. It also increased the efficiency of the canteen managers and also their profit margin, once they have sbetter and good facilities.

CHAPTER 1

Introduction

The manual method of ordering for foods is characterized with numerous problems. Some of these are : customers having little or no information about the items in the canteen; A new guest coming to canteen room that is either too expensive or too unbefitting for his/her personality; Prolonged delay by the receptionist in retrieving certain information about any particular item that's new in the canteen whenever such information being demanded by the manager; The foul play that sometimes occurs when information about the items sold are not officially documented by the canteen manager etc. All these problems and more would definitely make a canteen experience a down turn in business.

The main purpose of this work is therefore to develop a web application program that would circumvent all those problems encountered in the manual food order booking system, so that customers can easily go online with their mobile phones, tablets or laptops in order to browse the relevant information they need about the items in the canteen so that they can book for the appropriate food that is within their budget.

OBJECTIVE

- ❖ Customers can easily go online with their mobile phones or laptops to browse about items in the canteen.
- ❖ Guests can buy food within their budgets after seeing and accessing the items available in the canteen.
- ❖ Canteen managers can easily access information online without delay or with little delay.

➤ Applications:

- ❖ This system can be applied in any canteen.
- ❖ It can also be implemented in resorts, Hotels, Restaurants, Industry , factory etc.

Problem specification

- ❖ Customers having little or no information about the items in the canteen.
- ❖ Prolonged delay by the canteen manager in retrieving certain information about any particular item on demand.

Scope of study

- ❖ This study is aimed at finding out how effective the online food ordering and management system will improve the operations of booking system in canteens. However, out of the several departments that make up the canteen, this research project is restricted to only one section (i.e the food ordering) section.

Advantages:

- Sometimes it happens that the items get finished when one place the order therefore user can make advance booking using this system.
- It saves user time in search of items.
- The system is useful as it calculates an exact cost for requested number of items.
- It saves organization resources and expenses.
- This system is effective and saves time and cost of users.
- Easy registration.

Disadvantages:

- The booking process usually requires a customer identity which the system cannot detect.
- It requires a reliable internet connection.

Limitations of the existing System-

1.MARKET: - To determine whether there is demand for the canteens in the area where the site is located , certain very important factor must beconsidered.

2.LABOUR SITUATION: - Problem of staff is very important in canteen industry .For successfully developing new canteen project, it is the lack of adequate labour supply which creates problems rather than the problem from staff such as theirdemands.

3.SOURCES OF FINANCING: - This is another very important factor and big problems, arranging of finances and control of credit line is very serious problem confronting business today.

Chapter 2

REQUIREMENT ANALYSIS

2.1 Functional Requirements:

- ❖ The system supports customers booking and able to modify them.
- ❖ Customers can search based on foods of different categories.
- ❖ Customers able to cancel their order from their account.
- ❖ Staffs able to edit customers order booking information (updating name, number, items orders etc).
- ❖ Customers can book online and pay with credit or debit card.
- ❖ The system must send booking confirmation email after successful payment.
- ❖ Customers can write reviews about canteen and also rate them.
- ❖ Customers able to check their ordering status from their individual account.
- ❖ Customers can send feedback or call the manager for ordering purposes.
- ❖ Customers can check for latest promotion or sale.

2.2 Non-Functional Requirements

- ❖ The system must ensure that all the transferable data as for examples customers credit or debit card number, CVV Code, e-payment should be done in secured connection.
- ❖ The system must be able to handle multiple transactions a time.
- ❖ The system must provide customers 24*7 hours online booking service.
- ❖ The system should support almost all the browsers (Internet Explorer, Safari, Chrome, and Firefox).
- ❖ System should send the newsletter about ongoing promotions or sales to registered customers.
- ❖ Customers need to cancel the order.

Chapter 3

METHODOLOGY

3. Methodology

➤ Waterfall Model

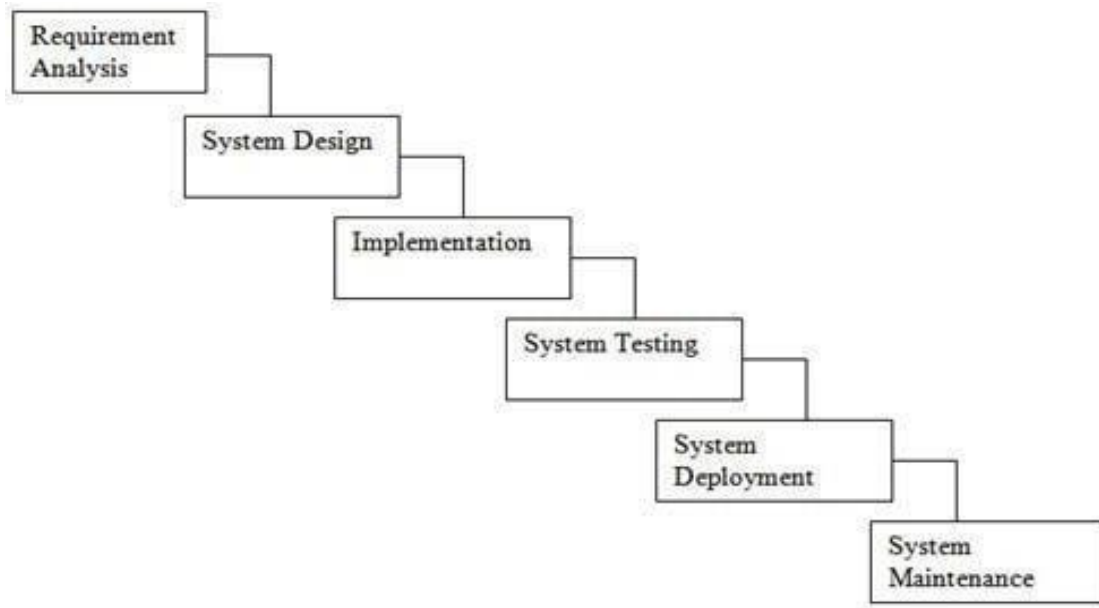


Fig. 3.1 Waterfall Life Cycle Model

Justification of Methodology

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- The project is short.
- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.

Description of Methodology

The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system:** Once the functional and nonfunctional testing is done, the product is deployed in the customer environment or released into the market.
- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

Chapter 4

DESIGN AND DEVELOPMENT

4.1. Introduction

The chapter describes the system study, a design strengths and weaknesses of the current system, Context level diagrams, Entity Relationship Diagram, Architectural design.

System Requirement

This section describes the hardware components and software requirements needed for effective and efficient running of the system

Table: 4.1 Hardware Requirements

SL	Hardware	Minimum System Requirement
01	Processor	Intel Pentium 4 or AMD
02	Memory	2 GB RAM
03	Disk Space	40 GB

Table: 4.2 Software Requirements

SL	Software	Minimum System Requirement
01	Operating System	Windows Server 2008, Windows (XP, 7, 8, 8.1)
02	Database Management System	Microsoft SQL Server 2014
03	Runtime Environment	Visual Studio 2008 Team System

The table above shows software requirements recommended to enable the system to run as required for using Online CANTEEN FOOD ORDERING System (CFOS).

Communications Interfaces

Internet connection and Browser are required in order for several functions to be executed such as downloading. The system uses the following browsers: -

- Mozilla Firefox
- Google Chrome
- Internet Explorer

TABLE :4.3 Technologies Used

Operating System:	Windows 10
Language Requirements:	PHP, Html, CSS, SQL, JavaScript, JSP, Servlet
Web Browser:	Google Chrome, Mozilla Firefox, Internet Explorer
Database Technology:	MySQL
Tools & Development:	Eclipse, MySQL Workbench

4.4 Operating Environment

Programming languages

- **PHP** : PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.
- **HTML**: Hyper Text Markup is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document and to supplement that text with interactive forms, embedded images, and other objects
- **JavaScript**: A client side scripting language used to create dynamic web content and user interface. Tools & Development Environment
- **Apache Tomcat 8.0.1 Server**: Apache Tomcat is a Servlet container developed by the Apache Software Foundation (ASF). Tomcat implements
- the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run.

- **NetBeans IDE** is a modular, standards-based integrated development environment (IDE), written in the Java programming language. The NetBeans project consists of a full-featured open source IDE written in the Java programming language and a rich client application platform, which can be used as a generic framework to build any kind of application

1. System Design

After interpretation of the data, tables were drawn and process of data determined to guide the researcher of the implementation stage of the project. The tools, which were employed during this methodology stage, were mainly tables, Data Flow Diagrams and Entity Relationship Diagrams. The design ensures that only allows authorized users to access the systems information.

Logical Model

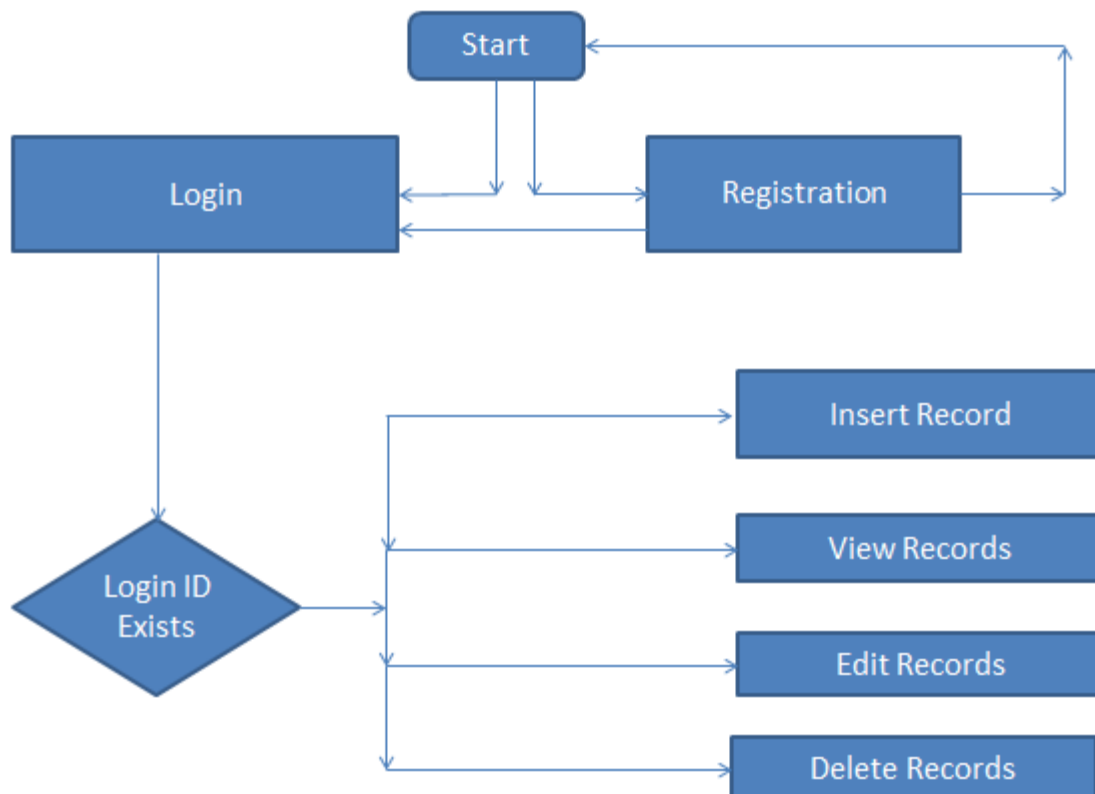


Fig. 4.1 Logical model

System Architecture

This gives a high level view of the new system with the main components of the system and the service they provide and how they communicate. The system is implemented using a three-tier architecture that comprises of our interface, process management and DBMS as illustrated bellow.

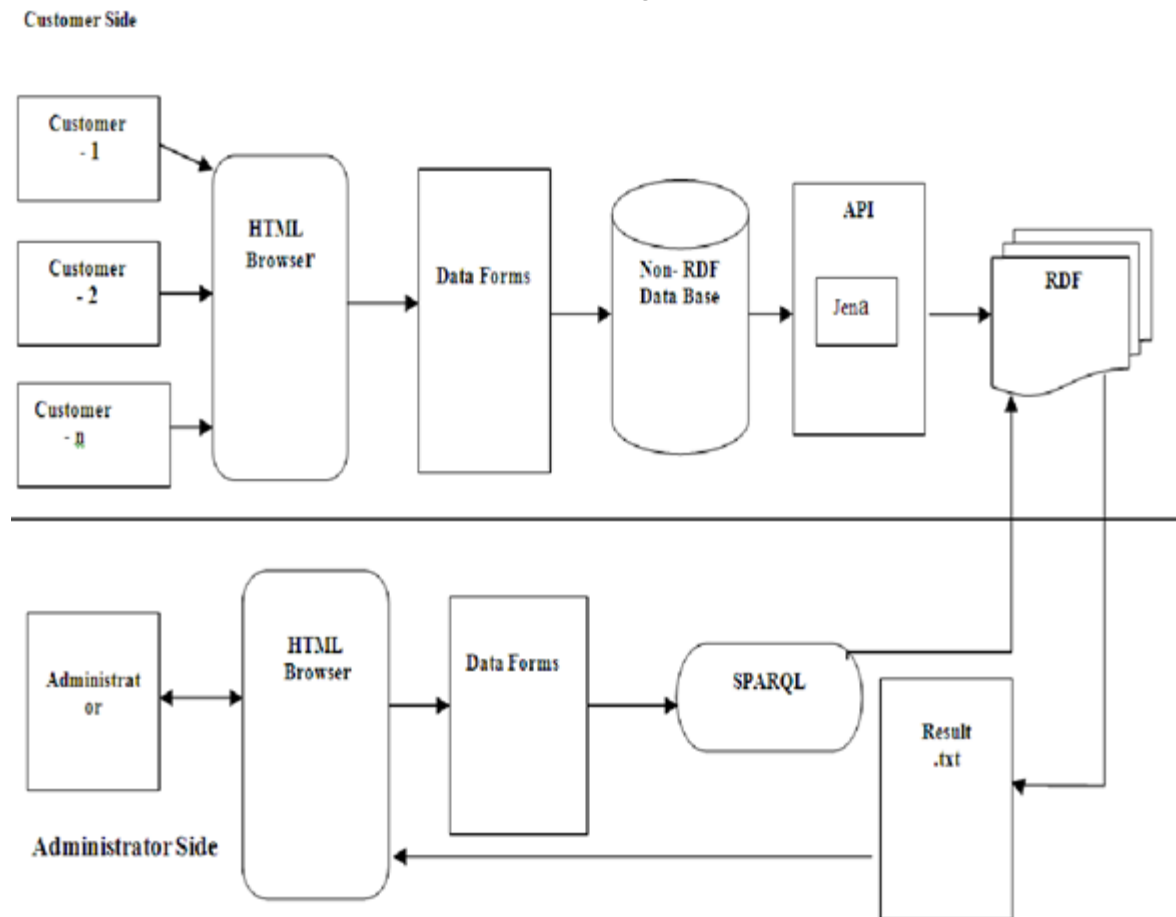


Fig. 4.2 System architecture

Entity Relationship (E-R) 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. An entity relationship diagram is a means of visualizing how the information a system produces is related.

Entity

Which are represented by rectangle. An entity is an object or concept that has its existence in the real world. It includes all those things about which data is collected. A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

Attributes

Which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.

An Entity Set

It is a set of entities of the same type that share the same properties, or attributes.

Process

A process shows a transformation or manipulation of data flows within the system.

Actions

Which are represented by diamond shapes, show how two entities share information in the database.

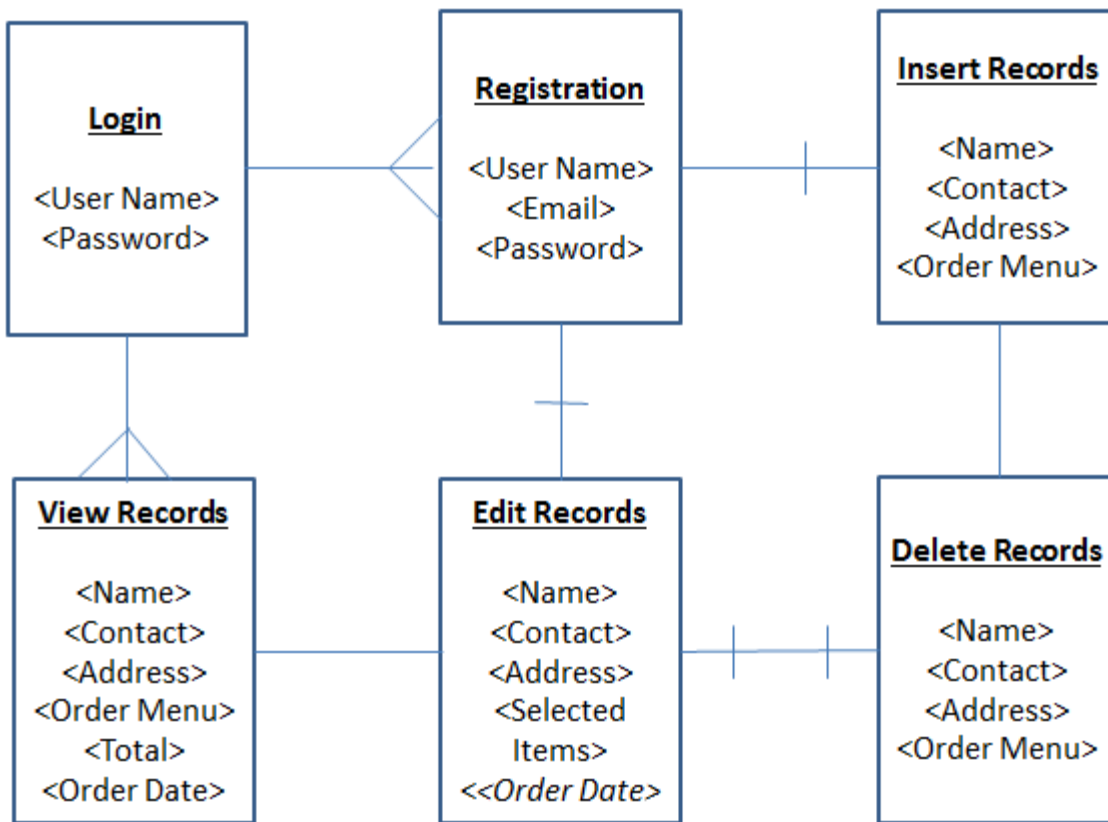


Fig. 4.3 Entity Relationship Diagram

Problem Definition

The sole aim of this project is to find information for the Admin to their canteen staff, provide a one stop solution to the problems of the customer related to their food ordering and also provides the facility.

The main purposes of this project are: -

1. To provide a list of available items to the customer.
2. To provide Admin all information about the items in the canteen.
3. To provide the staff information to the admin.

Benefits of the Project:

The system includes the Staff, Admin, Manager. Benefits to each of these are described below.

Benefit to Admin:

Admin can manage all the information about the canteen and provide the good facility to the customer.

Benefit to Staff:

Staff collects the information of customer order and payment information.

Benefit to Manager:

Manager collects the information about the staff like (staff id, staff phone number, staff email etc...) and also collect the information about customer details.

CHAPTER 5

Application

Here the brief description about how to use the system should be given. It is basically the user manual to understand the interface and operation.

5.1 Product Features

- It is easy to use the online website with all the information provided in the form of links.
- The admin, manager and customers can access the portal with simple registration.
- The product enables the admin to confirm the booking of the food items.
- Customers can access online booking of the room by paying some monetary value.
- The product provides online Canteen management.
- Customer can discuss their problem with the admin.

Design and Implementation Constraints

- There is no maintainability of back up so availability will get affected.
- Limited to HTTP/HTTPS Protocols.
- No multilingual support
- User do not have any rights to edit any data in the system

User Documentation

The user should be familiar with the Online study related terminology like Downloading / Using mail-account/Transaction/Chatting etc.

- The user should be familiar with the Internet.

Assumptions and Dependencies

- The details related to the admin, manager, staff, customers, payment and service transaction provided online.
- Administrator is created in the system already.
- Roles and tasks are predefined.
- Roles and responsibilities are already established.

User Interface Description

Output

Screens 1

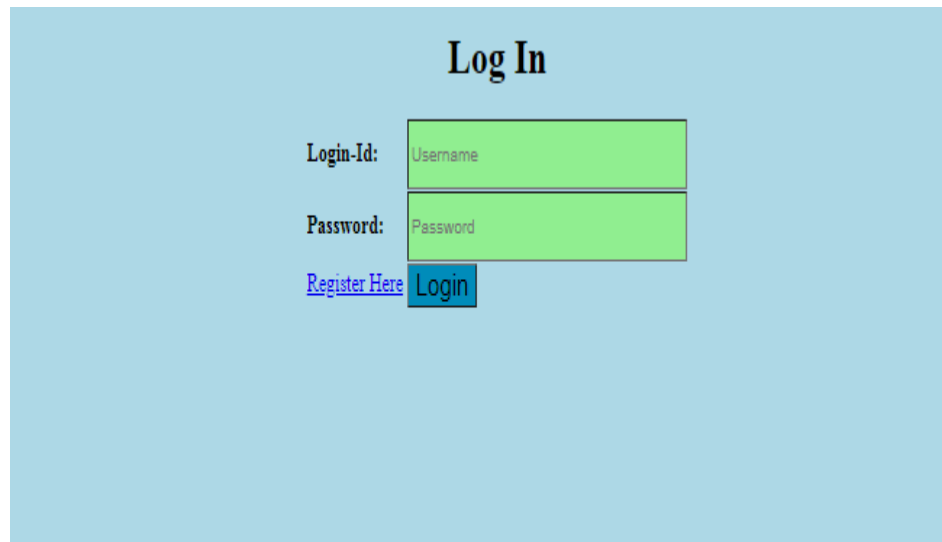
Main Page



Fig. 5.1 Main page

Login

The Login Page



Log In

Login-Id:

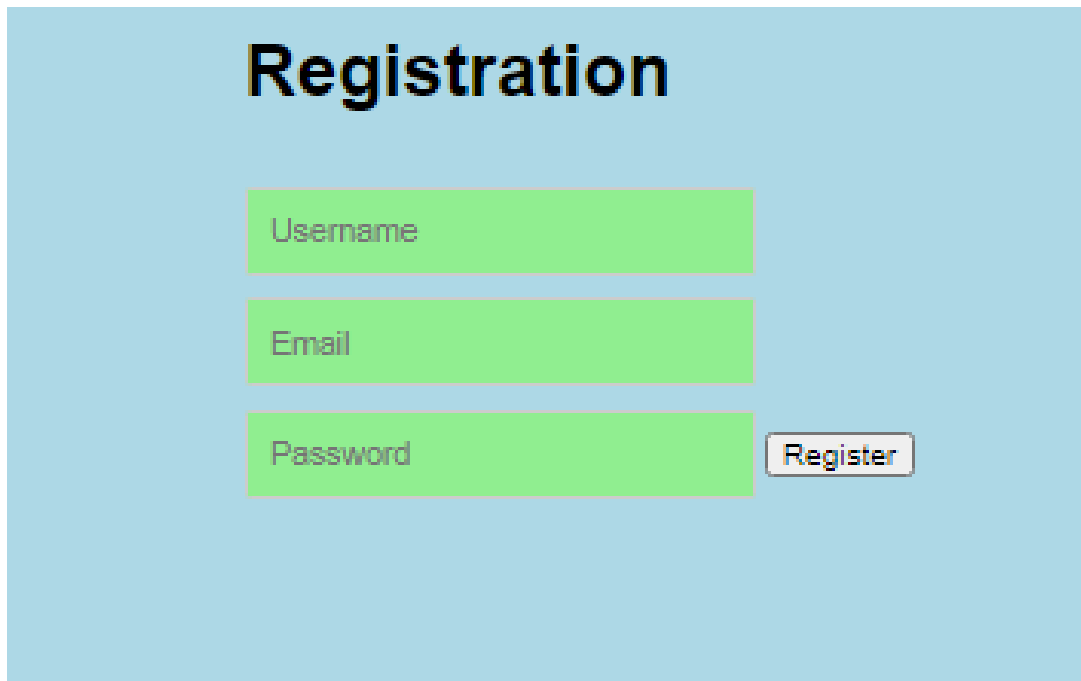
Password:

[Register Here](#)

Fig. 5.2 Login page

Registration

The Registration Page



The image shows a registration form on a light blue background. At the top, the word "Registration" is written in a large, bold, black font. Below this, there are three green rectangular input fields stacked vertically. The first field is labeled "Username", the second "Email", and the third "Password". To the right of the "Password" field is a small, light gray button with the word "Register" in black text.

Registration

Username

Email

Password

Register

Fig. 5.3 Registration page

Insert New Record

[Home](#) | [View Records](#) | [Logout](#)

Insert New Record

Name:

Contact:

Address:

ORDERED MENU

Normal Veg Thali: ☐ Ordered_quantity: Ordered_rate:

Special Veg Thali: ☐ Ordered_quantity: Ordered_rate:

Non Veg Thali: ☐ Ordered_quantity: Ordered_rate:


Ordered_date: 

Fig. 5.4 Insert page

View Records

[Home](#) | [Insert New Record](#) | [Logout](#)

View Records

S.No	Name	Contact	Address	Normal	Rate	Special	Rate	NONveg	Rate	TOTAL	Ordered_date	Edit	Delete
1	Prakash	6985478521	Noida	-	-	3	250	1	350	1100	2020-09-25	Edit	Delete
2	Rajesh	6254202154	Greater Noida	2	150	-	-	-	-	300	2020-09-24	Edit	Delete

Fig. 5.5 View records page

Edit/Update Record

[Home](#) | [Insert New Record](#) | [Logout](#)

Update Record

Name:	Prakash	Ordered Menu
Contact:	0985478521	
Address:	Noida	
Normal_thali: <input type="checkbox"/>		
Special_thali: <input type="checkbox"/>		
NONveg_thali: <input type="checkbox"/>		
ordered_date: 25 - 09 - 2020		

ordered_quantity_normal: Enter quantity

ordered_quantity_special: 3

ordered_quantity_nonveg: 1

ordered_rate_normal: 150

ordered_rate_special: 250

ordered_rate_nonveg: 350

Update

Fig. 5.6 Edit/Update records page

CHAPTER 6

Testing

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

Psychology of Testing

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't work. Testing is the process of executing a program with the intent of finding errors.

Testing Objectives

The main objective of testing is to uncover a host of errors, systematically with minimum effort and time. Stating formally, we can say,

- ❖ Testing is a process of executing a program with the intent of finding an error
- ❖ A successful test is one that uncovers an as yet undiscovered error.
- ❖ The tests are inadequate to detect possibly present errors.
- ❖ The software more or less confirms to the quality and reliable standards

System Testing

The philosophy behind testing is to find errors. Test cases are devised with this in mind. A strategy employed for system testing is code testing.

Code Testing:

This strategy examines the logic of the program. To follow this method we developed some test data that resulted in executing every instruction in the program and module i.e. every path is tested. Systems are not designed as entire nor are they tested as single systems. To ensure that the coding is perfect two types of testing is performed or for that matter is performed or that matter is performed or for that matter is performed on all systems.

Types Of Testing

- Unit Testing
- Link Testing

Unit Testing

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of a module. There are so many modules like Login, HWAdmin, Master Admin, Normal User, and PManager. Giving different sets of inputs has tested each module. When developing the module as well as finishing the development so that each module works without any error. The inputs are validated when accepting from the user.

In this application developer tests the programs up as system. Software units in a system are the modules and routines that are assembled and integrated to form a specific function. Unit testing is first done on modules, independent of one another to locate errors. This enables to detect errors. Through this errors resulting from interaction between modules initially avoided.

Link Testing

Link testing does not test software but rather the integration of each module in system. The primary concern is the compatibility of each module. The Programmer tests where modules are designed with different parameters, length, type etc.

Integration Testing

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions.

In this project integrating all the modules forms the main system. When integrating all the modules I have checked whether the integration effects working of any of the services by giving different combinations of inputs with which the two services run perfectly before Integration

Acceptance Testing

Acceptance Test is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system; the internal logic of program is not emphasized.

In this project 'Network Management Of Database System' I have collected some data and tested whether project is working correctly or not.

Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

White Box Testing

This is a unit testing method where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors. I tested step wise every piece of code, taking care that every statement in the code is executed at least once. The white box testing is also called Glass Box Testing. I have generated a list of test cases, sample data, which is used to check all possible combinations of execution paths through the code at every module level.

Black Box Testing

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details at statement level. Here the module will be treated as a block box that will take some input and generate output. Output for a given set of input combinations are forwarded to other modules.

Criteria Satisfied by Test Cases

Test cases that reduced by a count that is greater than one, the number of additional test cases that much be designed to achieve reasonable testing.

Test cases that tell us something about the presence or absence of classes of errors, rather than an error associated only with the specific test at hand.

CHAPTER 7

CONCLUSION

The system is having some benefits for customers too. In home page customers can put their email address or phone number for downloading the booking apps directly on their phone. The application will work for both OS Platform (IOS & Android). By this customers can book by using this application. Besides, customers can also book online without registering into the system

Again, customers no need to worry for payment. But for reservation they need to key in their credit card details for verification purposes. It will not charge them promptly. So, overall this system will give the customers a better user friendly environment to book online.

At the end it is concluded that we have made afford on following points.....

- ❖ A description of the background and context of the project and its relation to work already done in the area.
- ❖ Made statement of the aims and objectives of the project.
- ❖ The description of Purpose, Scope, and applicability.
- ❖ We define the problem on which we are working in the project
- ❖ We'll describe the requirement Specifications of the system and the actions that can be done on these things.
- ❖ We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- ❖ We included features and operations in detail, including screen layouts.
- ❖ We designed user interface and security issues related to system.
- ❖ Finally the system is implemented and tested according to test cases.

Future Enhancement

Since this system has been generated by using Object Oriented programming, there are many chances of reusability of the codes in other environment even in different platforms. Also its present features can be enhanced by some simple modification in the codes so as to reuse it in the changing scenario.

The website is made in all possible ways to meet the basic functional requirements of searching for food items by customers and online management of the system. However, as needs of the users grow in future, some more functionality like adding a payment gateway so that the user may pay online through credit card. Also, with the introduction of new Operating Systems and new versions of browsers which display the pages more dynamically some more functional modules can be added to the online canteen management system.

CHAPTER 8

CODING

The developed system can be divided into four major modules namely:

1. Login
2. Registration
3. Insert Records
4. View Records
5. Edit/Update Records
6. Delete Records

Each of these modules has certain specific functionality related to them and are described below:-

• Login

Login.php

```
<?php
```

```
session_start();
```

```
    require('db.php');
```

```
    // If form submitted, insert values into the database.
```

```
    if (isset($_POST['username'])) {
```

```
        $username = stripslashes($_REQUEST['username']); //  
removes backslashes
```

```
        $username = mysqli_real_escape_string($con,$username);  
//escapes special characters in a string
```

```
        $password = stripslashes($_REQUEST['password']);
```

```
        $password = mysqli_real_escape_string($con,$password);
```

```
        //Checking is user existing in the database or not
```

```
$query = "SELECT * FROM `users` WHERE username='$username'
and password='".md5($password)."'";
```

```
$result = mysqli_query($con,$query) or die(mysqli_error());
```

```
$rows = mysqli_num_rows($result);
```

```
if($rows==1){
```

```
    $_SESSION['username'] = $username;
```

```
    header("Location: insert.php"); // Redirect user to
index.php
```

```
    }else{
```

```
        echo "<div class='form'><h3>Username/password
is incorrect.</h3><br/>Click here to <a
href='login.php'>Login</a></div>";
```

```
    }
```

```
    }else{
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <meta charset="utf-8">
```

```
        <title>Login</title>
```

```
        <style>
```

```
body{background-color:lightblue;}
```

```
input[type='date'],input[type='password'],input[type='text']{background-
color:lightgreen;height:40px;width:250px;}
```

```
input[type='submit'] {background-color:#008CBA;font-size:20px;}
```

```
p{background-color:yellow;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
    <div align="center" class="form">
```

```
        <h1>Log In</h1>
```

```

        <form action="" method="post" name="login">
            <table>
                <tr>
                    <td><b>Login-Id:</b></td><td><input
type="text" name="username" placeholder="Username" required /></td>
                </tr>
                <tr>
                    <td><b>Password:</b></td><td><input type="password"
name="password" placeholder="Password" required /></td>
                </tr>
                <tr>
                    <td><a
href='registration.php'>Register Here</a></td>
                    <td><input name="submit"
type="submit" value="Login" /></td>
                </tr>
            </table>
        </form>
    </div>
<?php } ?>
</body>
</html>

```

• Registration

Registration.php

```

<?php
?>

```

```

<!DOCTYPE html>

```

```
<html>
<head>
<meta charset="utf-8">

<title>Registration</title>
<link rel="stylesheet" href="css/style.css" />
</head>
<body>
<?php
    require('db.php');
    // If form submitted, insert values into the database.
    if (isset($_REQUEST['username'])) {
        $username = stripslashes($_REQUEST['username']); // removes backslashes
        $username = mysqli_real_escape_string($con,$username); //escapes special
        characters in a string
        $email = stripslashes($_REQUEST['email']);
        $email = mysqli_real_escape_string($con,$email);
        $password = stripslashes($_REQUEST['password']);
        $password = mysqli_real_escape_string($con,$password);

        $trn_date = date("Y-m-d H:i:s");
        $query = "INSERT into `users` (username, password, email, trn_date)
        VALUES ('$username', '".md5($password)."', '$email', '$trn_date')";
        $result = mysqli_query($con,$query);
        if($result){
            echo "<div class='form'><h3>You are registered
            successfully.</h3><br/>Click here to <a href='login.php'>Login</a></div>";
        }
    } else {
?>
```

```
<div class="form">
<h1>Registration</h1>
<form name="registration" action="" method="post">
<input type="text" name="username" placeholder="Username" required />
<input type="email" name="email" placeholder="Email" required />
<input type="password" name="password" placeholder="Password" required />
<input type="submit" name="submit" value="Register" />
</form>
</div>
<?php } ?>
</body>
</html>
```

Insert Records

Insert.php

```
<?php
require('db.php');
include("auth.php");
$status = "";
if(isset($_POST['new']) && $_POST['new']==1)
{
    $trn_date = date("Y-m-d H:i:s");
    $name=$_REQUEST['name'];
    $contact=$_REQUEST['contact'];
    $address=$_REQUEST['address'];
    $ordered_menu=$_REQUEST['ordered_menu'];
    $menu="";
    foreach($ordered_menu as $menu1)
```

```

{
    $menu .= $menu1.",";
}

$ordered_quantity_normal= $_REQUEST['ordered_quantity_normal'];
$ordered_rate_normal = $_REQUEST['ordered_rate_normal'];
$ordered_quantity_special= $_REQUEST['ordered_quantity_special'];
$ordered_rate_special = $_REQUEST['ordered_rate_special'];
$ordered_quantity_nonveg= $_REQUEST['ordered_quantity_nonveg'];
$ordered_rate_nonveg = $_REQUEST['ordered_rate_nonveg'];
$ordered_date = $_REQUEST['ordered_date'];
$submittedby = $_SESSION["username"];

$ins_query="insert into $record_table
(trn_date,name,contact,address,ordered_menu,ordered_quantity_normal,ordered
_rate_normal,ordered_quantity_special,ordered_rate_special,ordered_quantity_
nonveg,ordered_rate_nonveg,ordered_date,submittedby)
values
('$trn_date','$name','$contact','$address','$menu','$ordered_quantity_normal','15
0','$ordered_quantity_special','250','$ordered_quantity_nonveg','350','$ordered_
date','$submittedby)";

mysqli_query($con,$ins_query) or die(mysql_error());

$status = "New Record Inserted Successfully.<br><br><a href='view.php'>View
Inserted Record</a>";
}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Insert New Record</title>

<style>

body{ background-color:lightblue;}

input[type='date'],input[type='text']{ background-

```

```

        color:lightgreen;height:30px;width:250px;}
input[type='submit'] {background-color:#008CBA;font-size:20px;}
p{background-color:yellow;}
</style>
</head>
<body>
<div align="center" class="form">
<p><a href="index.php">Home</a> | <a href="view.php">View Records</a> | <a
href="logout.php">Logout</a></p>
<h2>Insert New Record</h1>
<form name="form" method="post" action="<?php echo
$_SERVER["PHP_SELF"]?>">
<table colspan="10" cellpadding="1" cellspacing="1">
<tr><td><input type="hidden" name="new" value="1"/></td></tr>
<tr><td><b>Name</b>:</td><td><input type="text" name="name"
placeholder="Enter Name" /></td></tr>
<tr><td><b>Contact</b>:</td><td><input type="text" name="contact"
placeholder="Enter contact" /></td></tr>
<tr><td><b>Address</b>:</td><td><input type="text" name="address"
placeholder="Enter address" /></td></tr>
<tr><td><strong>ORDERED MENU</strong></td></tr>
<tr><td>Normal Veg Thali:</td><td><input type="checkbox"
name="ordered_menu[]" value="Normal Veg Thali" /></td>
<td>Ordered_quantity:</td><td><input type="text"
name="ordered_quantity_normal" placeholder="Enter Quantity_normal"
/></td>
<td>Ordered_rate:</td><td><input type="text" name="ordered_rate_normal"
placeholder="150" readonly /></td></tr>
<tr><td>Special Veg Thali:</td><td><input type="checkbox"
name="ordered_menu[]" value="Special Veg Thali" /></td>
<td>Ordered_quantity:</td><td><input type="text"
name="ordered_quantity_special" placeholder="Enter Quantity_special"

```



```

/></td>

<td>Ordered_rate:</td><td><input type="text" name="ordered_rate_special"
placeholder="250" readonly /></td></tr>

<tr><td>Non Veg Thali:</td><td><input type="checkbox"
name="ordered_menu[]" value="Nonveg Thali" /></td>

<td>Ordered_quantity:</td><td><input type="text"
name="ordered_quantity_nonveg" placeholder="Enter Quantity_nonveg"
/></td>

<td>Ordered_rate:</td><td><input type="text" name="ordered_rate_nonveg"
placeholder="350" readonly /></td>

<tr><td>Ordered_date:</td><td><input type="date" name="ordered_date"
placeholder="Enter ordered_date" required /></td></tr>

<tr><td></td><td><input name="submit" type="submit" value="Submit"
/></td></tr>

</table>

</form>

<?php echo $status; ?>

</div>

</body>

</html>

```

- View Records

View.php

```

<?php
require('db.php');

```

```
include("auth.php");

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>View Records</title>

<style>

body{ background-color:lightblue;}

input[type='date'],input[type='text']{ background-color:lightgreen;}

input[type='submit'] { background-color:#008CBA;font-size:20px;}

p{ background-color:yellow;}

table{ background-color:lightgreen;}

</style>

</head>

<body>

<div align="center" class="form">

<p><a href="index.php">Home</a> | <a href="insert.php">Insert New
    Record</a> | <a href="logout.php">Logout</a></p>

<h2>View Records</h2>

<table colspan="10" width="100%" border="1" cellpadding="3"
    cellspacing="0px">

<tr><th align="center"><strong>S.No</strong></th>

<th align="center"><strong>Name</strong></th>

<th align="center"><strong>Contact</strong></th>

<th align="center"><strong>Address</strong></th>

<th align="center"><strong>Normal</strong></th>

<th align="center"><strong>Rate</strong></th>

<th align="center"><strong>Special</strong></th>

<th align="center"><strong>Rate</strong></th>
```

```

<th align="center"><strong>NONveg</strong></th>
<th align="center"><strong>Rate</strong></th>
<th align="center"><strong>TOTAL</strong></th>
<th align="center"><strong>Ordered_date</strong></th>
<th align="center"><strong>Edit</strong></th>
<th align="center"><strong>Delete</strong></th></tr>
<?php
$count=1;
$sel_query="Select * from catering_record ORDER BY id desc;";
$result = mysqli_query($con,$sel_query);
while($row = mysqli_fetch_assoc($result)) {
$total=0;
?>
<tr>
<td align="center"><?php echo $count; ?></td>
<td align="center"><?php echo $row["name"]; ?></td>
<td align="center"><?php echo $row["contact"]; ?></td>
<td align="center"><?php echo $row["address"]; ?></td>
<?php
if ($row["ordered_quantity_normal"] > 0)
{
?>
<td align="center"><?php echo $row["ordered_quantity_normal"]; ?></td>
<td align="center"><?php echo $row["ordered_rate_normal"]; ?></td>
<?php
$total += ($row["ordered_quantity_normal"] * $row["ordered_rate_normal"]);
}
else
{
?>

```

```

        <td align="center"><?php echo "-"; ?></td>
        <td align="center"><?php echo "-"; ?></td>
    <?php
    }
    ?>

    <?php
    if ($row["ordered_quantity_special"] > 0)
    {
    ?>
        <td align="center"><?php echo $row["ordered_quantity_special"]; ?></td>
        <td align="center"><?php echo $row["ordered_rate_special"]; ?></td>
    <?php
        $total += ($row["ordered_quantity_special"] * $row["ordered_rate_special"])
    }
    else
    {
    ?>
        <td align="center"><?php echo "-"; ?></td>
        <td align="center"><?php echo "-"; ?></td>
    <?php
    }
    ?>

    <?php
    if ($row["ordered_quantity_nonveg"] > 0)
    {
    ?>
        <td align="center"><?php echo $row["ordered_quantity_nonveg"]; ?></td>
        <td align="center"><?php echo $row["ordered_rate_nonveg"]; ?></td>

```

```

<?php
    $total += ($row["ordered_quantity_nonveg"] * $row["ordered_rate_nonveg"]);

}
else
{
    ?>
        <td align="center"><?php echo "-"; ?></td>
        <td align="center"><?php echo "-"; ?></td>
    <?php
    }
    ?>

    <td align="center"><?php echo $total; ?></td>

    <td align="center"><?php echo $row["ordered_date"]; ?></td>
    <td align="center"><a href="edit.php?id=<?php echo $row["id"];
        ?>">Edit</a></td><td align="center"><a href="delete.php?id=<?php echo
        $row["id"]; ?>">Delete</a></td></tr>

    <?php $count++; } ?>
</table>
</div>
</body>
</html>

```

- Edit/Update Records

Edit.php

```
<?php
require('db.php');
include("auth.php");
$id=$_REQUEST['id'];
$query = "SELECT * from $record_table where id='".$id.'";
$result = mysqli_query($con, $query) or die ( mysqli_error());
$row = mysqli_fetch_assoc($result);
?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Update Record</title>

<style>
body{ background-color:lightblue;}
input[type='date'],input[type='text']{ background-
    color:lightgreen;height:40px;width:250px;}
input[type='submit'] { background-color:#008CBA;font-size:20px;}
p{ background-color:yellow;}
</style>

</head>

<body>

<div align="center" class="form">

<p><a href="index.php">Home</a> | <a href="insert.php">Insert New
    Record</a> | <a href="logout.php">Logout</a></p>

<?php
$status = "";
```

```

if(isset($_POST['new']) && $_POST['new']==1)
{
$id=$_REQUEST['id'];
$trn_date = date("Y-m-d H:i:s");
$name = $_REQUEST['name'];
$contact = $_REQUEST['contact'];
$address = $_REQUEST['address'];
$order_quantity_normal= $_REQUEST['ordered_quantity_normal'];
$order_rate_normal = $_REQUEST['ordered_rate_normal'];
$order_quantity_special= $_REQUEST['ordered_quantity_special'];
$order_rate_special = $_REQUEST['ordered_rate_special'];
$order_quantity_nonveg= $_REQUEST['ordered_quantity_nonveg'];
$order_rate_nonveg = $_REQUEST['ordered_rate_nonveg'];
$order_date = $_REQUEST['ordered_date'];

$submittedby = $_SESSION["username"];
$update="update $record_table set trn_date='".$_$trn_date."', name='".$_$name."',
contact='".$_$contact."',address='".$_$address."',ordered_menu=' ',
order_quantity_normal='".$_$order_quantity_normal."',ordered_rate_normal='".$_$
$order_rate_normal."',ordered_quantity_special='".$_$order_quantity_special
.'",
order_rate_special='".$_$order_rate_special."',ordered_quantity_nonveg='".$_$ord
$order_quantity_nonveg."',ordered_rate_nonveg='".$_$order_rate_nonveg."',order
ed_date='".$_$order_date."',submittedby='".$_$submittedby.'" where id='".$_$id.'";
mysqli_query($con, $update) or die(mysqli_error($con));
$status = "Record Updated Successfully. </br></br><a href='view.php'>View
Updated Record</a>";
?>
<?php echo $status?></div><?php
}else {
?>
<h1 align="center">Update Record</h1>

```

```

<div align="center">
<form name="form" method="post" action="">
<table colspan="10" cellpadding="1" cellspacing="1">
<tr><input type="hidden" name="new" value="1" />
<tr><td><input name="id" type="hidden" value="<?php echo $row['id'];?>" />
<tr><td>Name: </td><td><input type="text" name="name" placeholder="Enter
Name" value="<?php echo $row['name'];?>" />
<tr><td>Contact:</td><td><input type="text" name="contact" placeholder="Enter
contact" value="<?php echo $row['contact'];?>" />
<tr><td>Address:</td><td><input type="text" name="address"
placeholder="Enter address" value="<?php echo $row['address'];?>" />

```

Ordered Menu

```

<?php
if($row['ordered_quantity_normal'] > 0)
{
?>
    <tr><td><td> Normal_thali:<input type="checkbox" name="ordered_menu[]"
placeholder="Enter Menu" value="" />
    <td>ordered_quantity_normal:</td><td><input type="text"
name="ordered_quantity_normal" placeholder="Enter quantity" value="<?php
echo $row['ordered_quantity_normal'];?>" />
    <td>ordered_rate_normal:</td><td><input type="text"
name="ordered_rate_normal" placeholder="Enter rate" value="<?php echo
$row['ordered_rate_normal'];?>" /></tr>
<?php
}
else
{
?>
    <tr><td><td> Normal_thali:<input type="checkbox" name="ordered_menu[]"
placeholder="Enter Menu" value="" />
    <td>ordered_quantity_normal:</td><td><input type="text"

```



```

        name="ordered_quantity_normal" placeholder="Enter quantity"
        value="<?php?>" />

        <td>ordered_rate_normal:</td><td><input type="text"
        name="ordered_rate_normal" placeholder="Enter rate" value="150" readonly
        /></td>

<?php
}
?>

<?php
if($row['ordered_quantity_special'] > 0)
{
?>

        <tr><td><td> Special_thali:<input type="checkbox" name="ordered_menu[]"
        placeholder="Enter Menu" value="" />

        <td>ordered_quantity_special:</td><td><input type="text"
        name="ordered_quantity_special" placeholder="Enter quantity" value="<?php
        echo $row['ordered_quantity_special'];?>" />

        <td>ordered_rate_special:</td><td><input type="text"
        name="ordered_rate_special" placeholder="Enter rate" value="<?php echo
        $row['ordered_rate_special'];?>" /></td>

<?php
}
else
{
?>

        <tr><td><td> Special_thali:<input type="checkbox" name="ordered_menu[]"
        placeholder="Enter Menu" value="" />

        <td>ordered_quantity_special:</td><td><input type="text"
        name="ordered_quantity_special" placeholder="Enter quantity"
        value="<?php?>" />

```

```

        <td>ordered_rate_special:</td><td><input type="text"
        name="ordered_rate_special" placeholder="Enter rate" value="250" readonly
        /></tr>

<?php
}
?>

<?php
if($row['ordered_quantity_nonveg'] > 0)
{
?>

        <tr><td><td> NONveg_thali:<input type="checkbox" name="ordered_menu["
        placeholder="Enter Menu" value="" />

        <td>ordered_quantity_nonveg:</td><td><input type="text"
        name="ordered_quantity_nonveg" placeholder="Enter quantity" value="<?php
        echo $row['ordered_quantity_nonveg'];?>" />

        <td>ordered_rate_nonveg:</td><td><input type="text"
        name="ordered_rate_nonveg" placeholder="Enter rate" value="<?php echo
        $row['ordered_rate_nonveg'];?>" /></tr>

<?php
}
else
{
?>

        <tr><td><td> NONveg_thali:<input type="checkbox" name="ordered_menu["
        placeholder="Enter Menu" value="" />

        <td>ordered_quantity_nonveg:</td><td><input type="text"
        name="ordered_quantity_nonveg" placeholder="Enter quantity"
        value="<?php?>" />

        <td>ordered_rate_nonveg:</td><td><input type="text"
        name="ordered_rate_nonveg" placeholder="Enter rate" value="350" readonly
        /></tr>

```

```
<?php
}
?>
```

```
<tr><td>ordered_date:</td><td><input type="date" name="ordered_date"
placeholder="Enter ordered_date" value="<?php echo
$row['ordered_date'];?>" />
```

```
<td><input name="submit" type="submit" value="Update" />
```

```
</form>
```

```
<?php } ?>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

• Delete Records

Delete.php

```
<?php
require( 'db.php' );
$id=$_REQUEST['id'];
$query = "DELETE FROM $record_table WHERE id=$id";
$result = mysqli_query($con,$query) or die(mysqli_error($con));
header("Location: view.php");
?>
```

References

This will not be shown as Chapter Heading. Mention Bibliography here.

Bibliography

Books:-

1. Fundamentals of Software Engineering – Rajib Mall
2. SQL – Ivan Bayross
3. Java 2 The Complete Reference 5th Ed-Herbert

Schildt Websites:-

1. www.google.com
2. en.wikipedia.com
3. www.msdn.library
4. roseindia.com