VISVESVARAYA TECHNOLOGICAL UNIVERSITY
"JNANA SANGAMA", BELAGAVI-590018

WEB TECHNOLOGY MINI PROJECT

REPORT ON

FOOD ORDERING SYSTEM

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING
Submitted By

SHRAVAN BHAT	**1KS16CS092**
RUTHVIK B.R	**1KS16CS078**
SUHAS KAKADE K	**1KS16CS076**

Under the Guidance of

Mr. KUMAR K	**Mr. ROOPESH KUMAR B N**
Assistant Professor	Assistant Professor

Department of Computer Science & Engineering

K.S. INSTITUTE OF TECHNOLOGY
#14, Raghuvanahalli, Kanakapura Main Road, Bangalore-560109.

2019 - 2020

K.S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bangalore-560062. **Department of Computer Science & Engineering**

CERTIFICATE

This is to certify that the **WEB TECHNOLOGY MINI PROJECT** entitled **"FOOD ORDERING SYSTEM"** presented by **Mr. Ruthvik B.R** bearing USN **1KS16CS078**, **Mr. Suhas Kakade K** bearing USN **1KS16CS103** and **Mr. Shravan Bhat** bearing USN **1KS16CS092** of VII semester in the partial fulfillment for the award of the **Bachelor of Engineering** in **Computer Science & Engineering** of the **Visvesvaraya Technological University, Belgaum**, during the year **2019-2020**. The **WEB TECHNOLOGY MINI PROJECT** has been approved as it satisfies the academic requirements in respect of **Web Technology Laboratory with Mini Project (15CSL77)** prescribed for the Bachelor of Engineering degree.

_____   _____   _____
Mr. Kumar K   **Dr. Rekha B. Venkatapur**   **Dr. T.V. Govindaraju**
Assistant Professor, CSE   Prof. & Head, CSE   Principal / Director
K.S.I.T., Bengaluru   K.S.I.T., Bengaluru   K.S.I.T., Bengaluru

Name of the Student: _____ _____

USN: _____ _____

Signature of the Student: _____ _____

_____   _____
Signature of Internal Examiner   **Signature of External Examiner**

ACKNOWLEDGEMENT

The successful presentation of the **WEB TECHNOLOGY MINI PROJECT** would be incomplete without the mention of the people who made it possible and whose constant guidance crowned our effort with success.

We would like to extend our gratitude to the **MANAGEMENT, KAMMAVARI SANGHAM**, Bengaluru, for providing all the facilities to present the Web Technology Mini Project.

We would like to extend our gratitude to the **Dr. T.V. GOVINDARAJU,** Principal / Director, K.S. Institute of Technology, Bengaluru, for facilitating me to present the Web Technology Mini Project.

We thank **Dr. REKHA B. VENKATAPUR**, Professor and Head, Department of Computer Science and Engineering, K.S. Institute of Technology, Bengaluru, for her encouragement.

We would like to thank our Project Guides, **Mr. KUMAR K**, Assistant Professor and **Mr. ROOPESH KUMAR B N**, Assistant Professor; Department of Computer Science and Engineering, K.S. Institute of Technology, Bengaluru, for their constant guidance and inputs.

We would like to thank all the **Teaching** Staff and **Non-Teaching** Staff of the college for their co-operation.

Finally, we extend our heart-felt gratitude to our **families** for their encouragement and support without which we wouldn't have come so far. Moreover, we thank all our **friends** for their invaluable support and cooperation.

RUTHVIK B.R

SUHAS KAKADE K

SHRAVAN BHAT

ABSTRACT

With high demand on online food ordering, most of the restaurants are falling short of implementing the technology. Our project helps customers have a simple, elegant and easy interface to order food form their favourite restaurants.

The project is developed using PHP for back end coding, MySQL as the server for database, HTML5, CSS3 and Materialize for providing better views and client-side rendering and jQuery for executing queries for extracting data from database.

Features include a login form for customers who want to order as well as the administrator (for restaurant owners). Food menu display with order count updating as soon as food is ordered. Listing of ordered food and bill payment feature. The system also helps the admin to review orders as they are created. Cancelled orders are also recorded with all details.

Customers can also raise complaint tickets if there are any queries.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1   OVERVIEW

The Restaurant Ordering System is a tool that is used to order food with bill payment and menu details. The system makes use of a single centralized database to maintain records of all the orders. Authentication process is implemented to allow only the authorized users to register, view menu and order food. Menu, prices of dishes, order details are all stored in database. All the processes are carried out through an interactive interface on a web platform.

Having a centralized database for all users of the same group helps in maintaining the records and makes administrator job much easier.

1.2   PROBLEM STATEMENT

The aim of the project is to show the real-world use and implementation of a Web Application System developed using technologies such as HTML5, CSS, JavaScript, PHP. The tool is deployed on web for the customers to have easy access to food ordering services and it allows them to track, manage their food orders. It allows staff to access and serve the hotel customers along with customer support. It can be used across different operating systems and is easy to use.

1.3   WEB TECHNOLOGIES

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS). There are many other technologies that are available that helps us to create website best suited for our needs. We will learn more about these technologies in the following sub sections.

1.3.1 HTML5

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard.

It was published in October 2014 by the World Wide Web Consortium (W3C) to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc. HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind.
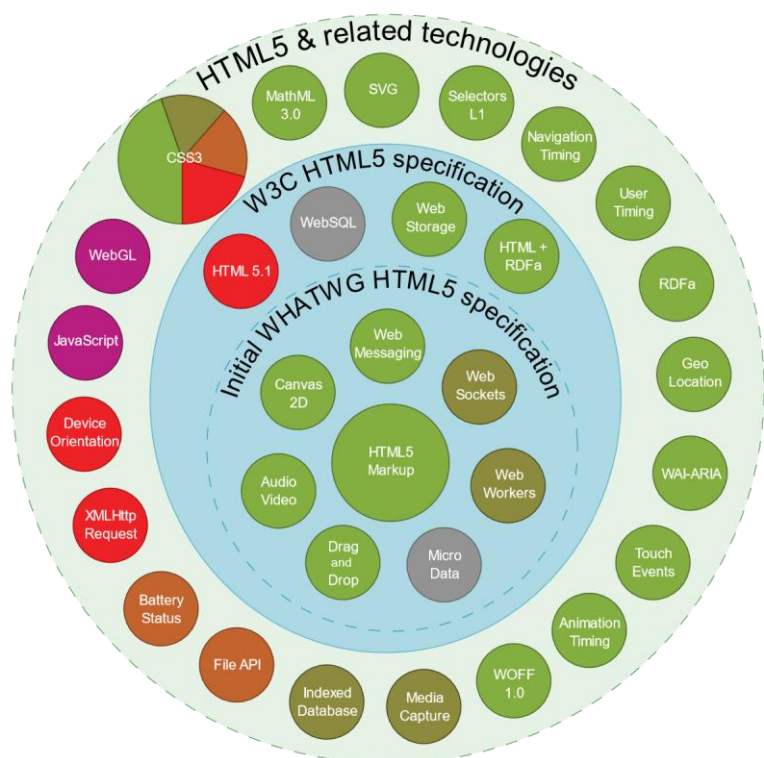

Fig. 1.1 HTML5 APIs and related technologies taxonomy and status

1.3.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS was first proposed by Håkon Wium Lie on October 10, 1994. At the time, Lie was working with Tim Berners-Lee at CERN. Several other style sheet languages for the web were proposed around the same time, and discussions on public mailing lists and inside World Wide Web Consortium resulted in the first W3C CSS Recommendation (CSS1) being released in 1996. In particular, Bert Bos' proposal was influential; he became co-author of CSS1 and is regarded as co-creator of CSS.[1]

Style sheets have existed in one form or another since the beginnings of Standard Generalized Markup Language (SGML) in the 1980s, and CSS was developed to provide style sheets for the web. One requirement for a web style sheet language was for style sheets to come from different sources on the web. Therefore, existing style sheet languages like DSSSL and FOSI were not suitable. CSS, on the other hand, let a document's style be influenced by multiple style sheets by way of "cascading" styles.

CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

Some of the notable advantages are as follows:

- Separation of content from presentation

  CSS facilitates publication of content in multiple presentation formats based on nominal parameters. Nominal parameters include explicit user preferences, different web browsers, the type of device being used to view the content (a desktop computer or mobile Internet device), the geographic location of the user and many other variables.

- Site-wide consistency

  When CSS is used effectively, in terms of inheritance and "cascading", a global style sheet can be used to affect and style elements site-wide. If the situation arises that the styling of the elements should be changed or adjusted, these changes can be made by editing rules in the global style sheet. Before CSS, this sort of maintenance was more difficult, expensive and time-consuming.

- Bandwidth

  A stylesheet, internal or external, specifies the style once for a range of HTML elements selected by 'class' type or relationship to others. This is much more efficient than repeating style information inline for each occurrence of the element. An external stylesheet is usually stored in the browser cache, and can therefore be used on multiple pages without being reloaded, further reducing data transfer over a network.

- Page reformatting

  With a simple change of one line, a different style sheet can be used for the same page. This has advantages for accessibility, as well as providing the ability to tailor a page or site to different target devices. Furthermore, devices not able to understand the styling still display the content.

- Accessibility

  Without CSS, web designers must typically lay out their pages with techniques such as HTML tables that hinder accessibility for vision-impaired users.
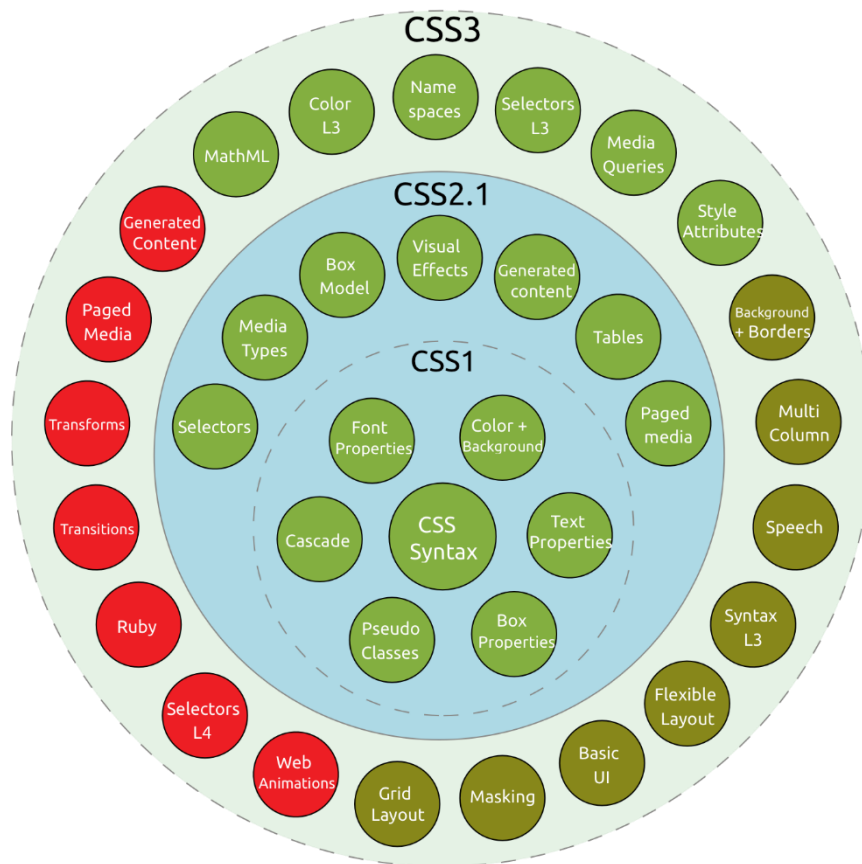
- Recommendation Last Call
- Candidate Recommendation
- Working Draft.

Fig. 1.2 Taxonomy and status of CSS3 modules

1.3.3 JavaScript

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.[2]

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Some of the notable advantages are as follows:

- Speed

  Being client-side, JavaScript is very fast because any code functions can be run immediately instead of having to contact the server and wait for an answer.

- Simplicity

  JavaScript is relatively simple to learn and implement.

- Versatility

  JavaScript plays nicely with other languages and can be used in a huge variety of applications. Unlike PHP or SSI scripts, JavaScript can be inserted into any web page regardless of the file extension. JavaScript can also be used inside scripts written in other languages such as Perl and PHP.

- Server Load

  Being client-side reduces the demand on the website server.

1.3.4 PHP

PHP: Hypertext Preprocessor (or simply PHP) is a server-side scripting language designed for Web development, and also used as a general-purpose programming language. It was originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is

now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.[3]

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The reason behind the popularity of PHP is its several advantages. PHP is most suited for the purpose of web development. The advantages of PHP are discussed briefly below:

- **Cross Platform**

    All the PHP based applications can run on various types of platforms. PHP is supported by majority of Operating Systems, some of which includes Solaris, UNIX, Windows and Linux. The mentioned platforms can be used to write codes in PHP and also view web pages or run the PHP based applications.

    PHP easily interfaces with MySQL and Apache both. An effortless integration of PHP can be done with various other technologies like Java and there is no requirement of re-development. Therefore, saving both time and money, giving it an important advantage.

- **Easy database connection**

    A programming language like PHP is widely used on the internet and needs to connect to the database very often. Therefore, having a feature that could help PHP to connect to database easily is mandatory. Several websites such as the ecommerce websites, require good database management system.

PHP has a built-in module that helps it in connecting with database easily. Therefore, PHP has a great demand in the field of web development where a data driven website needs to be developed. PHP significantly reduces the time needed in developing the web application that needs an efficient database management system.

○ Easy to use

PHP is widely used because it is easy to use. In contrast with other programming languages that are complex, PHP is simple, fluent, clean and organized, hence it is a boon for the new users. PHP has a well-organized syntax which is logical at the same time.

PHP does not require any intensive studying or manual to use it. Command functions of PHP are easily understood as the user can easily figure out from the name of the commands itself what it does. A person who is new to PHP can still code because the syntax is somewhat similar to C.

A person who is new to PHP can still code because the syntax is somewhat similar to C. Hence, if a person who knows C can easily code in PHP. Hence, it is easier to create and optimize the application using PHP.

Speed is the primary need of web development. There are people who face the challenge of slow internet connection and slow data speed. Furthermore, a fast loading website is always preferred by people across the globe. When compared to other programming languages, PHP is found to be the fastest programming language.

In normal circumstances, it takes a lot of time to connect to the database, when you attempt to fetch certain data from the database. It takes a lot of time in connecting to the database, then executing the statement and finally getting the data. PHP performs these set of tasks faster than other scripting languages. PHP is faster in both connecting to the database and in using other important applications.

The high speed of PHP gives it an advantage over other scripting languages and gives it an application in important administrations such as the server administration and mail functionalities.

○ Open source

One of the important advantages of PHP is that it is Open Source. Therefore, PHP is readily available and is entirely free. In contrast to other scripting languages used for web development which requires the user to pay for the support files, PHP is open to everyone, anytime and anywhere.

A beginner in PHP need not worry about the support as PHP is maintained and developed by a large group of PHP developers which helps in creating support community of PHP that helps people in PHP implementation and manipulation.

1.4   DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.[4]

A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified -- and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.4.1 ADVANTAGES OF DBMS

Central storage and management of data within the DBMS provides:

- Data abstraction and independence
- Data security
- A locking mechanism for concurrent access

- An efficient handler to balance the needs of multiple applications using the same data
- The ability to swiftly recover from crashes and errors, including restorability and recoverability
- Robust data integrity capabilities
- Logging and auditing of activity
- Simple access using a standard application programming interface (API)
- Uniform administration procedures for data

1.5   SQL

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS). SQL offers two main advantages: first, it introduced the concept of accessing many records with one single command; and second, it eliminates the need to specify how to reach a record, e.g. with or without an index.

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments.

CHAPTER 2

REQUIREMENTS SPECIFICATION

A computerized way of handling customer orders and ordering details is efficient, organized and time saving, compared to a manual way of doing so and having a single database also makes maintenance much easier. This is done through a database driven web application whose requirements are mentioned in this section.

2.1   OVERALL DESCRIPTION

A reliable and scalable database driven web application with security features, that is easy to use and maintain is the requisite and is developed on Windows 10 OS.

The purpose of developing this application is to have the stats and data of all the hotels of a single brand in one centralized database. This will help keep a record of all the regular customers. All this is implemented in a single web application and used across different hotels.

2.2   SPECIFIC REQUIREMENTS

The specific requirements of the Hotel Chain Desk Tool are stated as follows:

2.2.1 SOFTWARE REQUIREMENTS

- Web Browser – Google Chrome 62.0.3202.89 (stable)

- Editor – Microsoft Visual Studio Code v1.29

- XAMPP v7.2.8

- phpMyAdmin v4.8.2

- PHP v7.2.8

- Apache v2.4.34

- Operating System – Windows 8 or later

- Database Support – MariaDB v10.1.34

- Client API library version - mysqlnd 5.0.12-dev – 20150407

2.2.2 HARDWARE REQUIREMENTS

- ○ Processor - Intel Pentium 4 or higher

- ○ RAM – 2 GB (4 GB Recommended)

- ○ HDD – 4 GB

- ○ Monitor – VGA of 1024x768 screen resolution

- ○ Keyboard and Mouse

2.2.3 TECHNOLOGY STACK

- ○ HTML provides a means to structure text based information in a document. It allows users to produce web pages that include text, graphics and hyperlinks.

- ○ JavaScript is a scripting language which supports the development of both client and server applications. It is preferred at client side to write programs that can be executed by a web browser within the context of a web page.

- ○ CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language.

- ○ SQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data manipulation.

- ○ PHP: Hypertext Preprocessor (or simply PHP) is a server-side scripting language designed for Web development, and also used as a general-purpose programming language.

CHAPTER 3

DETAILED DESIGN

3.1   SYSTEM DESIGN

PHP is written as standard text files with the .php extension. PHP files are often saved within a folder in a web server's public directory (or a web root directory). On most systems this will either be named public or public_html. For example, if a file was saved as index.php in a web root directory, a user could access it by typing http://www.example.org or http://www.example.org/index.php.
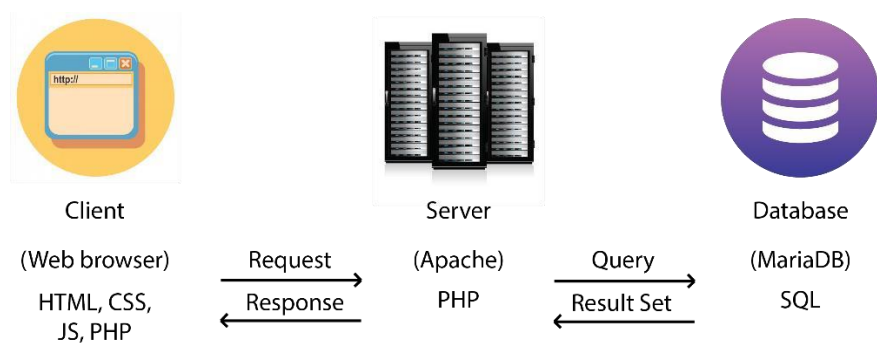

Fig. 3.1 FOOD Chain Desk Tool web architecture

So what exactly is happening when a user types in the URL http://example.org? When a user types in http://example.org in a Web client (a browser, for instance), the client issues a GET request to the server (let's assume that we are both using Apache). When Apache gets this request, it looks for a file named index.php (or index.html, remember the directory indexes from earlier?). If a file named index.php is found, Apache will take this as the root file of any folder and displays it if no further location is specified.

After Apache decides that is a PHP file, it gives it to the PHP interpreter. When PHP receives the file, it reads through it and executes any PHP code it can find. After it is done with the file, the PHP interpreter gives the output of the code, if any, back to Apache. When Apache gets the output back from PHP, it sends that output back to a browser which renders it to the screen.

3.2   IMPLEMENTING DATA

To show the development of our web application and implementation of our database we have taken the simulation of a café with various customers with the name of hotel as "CAFÉ CORNER".

3.3   ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to process in an area of a business.

An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute relationship diagrams, rather than entity-relationship models.
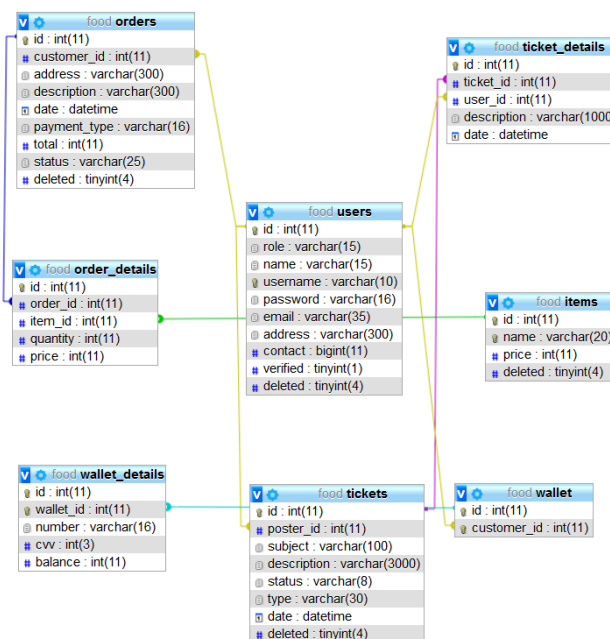

Fig. 3.2 ER Model of the proposed system

3.4   FLOW DIAGRAM

A data flow diagram is a graphical representation of the "flow" of data through an information system, modelling its process aspects.
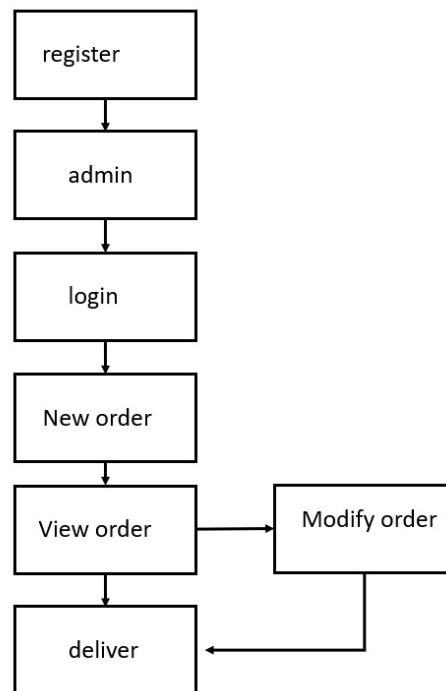

Fig. 3.3 Flow diagram of Food Ordering System

3.5   DESCRIPTION OF TABLES

The database consists of seven tables:

1. users – stores the credentials of authorized users.

    ➢ id – unique id name for accessing the website.

    ➢ password – code word to gain access.

    ➢ name – name of the customer.

    ➢ username – name given by the customer for login

➢ email – email of the user.

➢ address – House address of the customer.

➢ contact – customer's contact details.

2. orders – stores the information of the customer order.

➢ id – the unique order id

➢ customer_id – the id of the customer updating the order

➢ address – address of the customer

➢ description – the specifications given by customer

➢ date – date of order placed

➢ payment_type – mode of payment

➢ total – sum of order

➢ status – the current status of the order

3. items – stores the information of the dishes

➢ id – unique id given to each dish.

➢ name – name of the dish

➢ price – the amount value

➢ deleted – to specify availability

4. order_details – stores the details of orders

➢ id – order id

➢ order_id – unique id for each order

➢ item_id – monthly amount paying to owner

➢ quantity – number of foods ordered

➢ price – payment price

5. tickets – complaint ticket details raised by the customer

  - ➢ id – ticket id
  - ➢ subject – subject of the complaint ticket
  - ➢ description – description of problem faced
  - ➢ type – type of complaint
  - ➢ status – whether it is closed or still open
  - ➢ date – date of creating ticket

6. ticket_details – ticket details

  - ➢ id – ticket id
  - ➢ ticket_id – unique id of ticket
  - ➢ user_id – id of user
  - ➢ description – description of the complaint
  - ➢ date – date of creating ticket

7. wallet_details – wallet details

  - ➢ id – ticket id
  - ➢ wallet_id – unique id of wallet
  - ➢ user_id – id of user
  - ➢ number – account number
  - ➢ cvv – account cvv
  - ➢ balance – Balance details in account

CHAPTER 4

IMPLEMENTATION

4.1   MODULES AND THEIR ROLES

Website is developed using the following modules. Each module has its own functionality and contributes as a feature for the website.

1. Login

   The administrator and customers can login and if they are not registered, the customer can register and then login by filling up the required fields.

```php
<?php
include '../includes/connect.php';
$name = htmlspecialchars($_POST['name']);
$username = htmlspecialchars($_POST['username']);
$password = htmlspecialchars($_POST['password']);
$phone = $_POST['phone'];
function number($length) {
   for($i = 0; $i < $length; $i++)
      $result .= mt_rand(0, 9);
   return $result;
}
$sql = "INSERT INTO users (name, username, password, contact) VALUES ('$name', '$username', '$password', $phone);";
if($con->query($sql)==true){
$user_id =  $con->insert_id;
$sql = "INSERT INTO wallet(customer_id) VALUES ($user_id)";
if($con->query($sql)==true){
$wallet_id =  $con->insert_id;
$cc_number = number(16);
$cvv_number = number(3);
$sql = "INSERT INTO wallet_details(wallet_id, number, cvv) VALUES ($wallet_id, $cc_number, $cvv_number)";
$con->query($sql);
```

Code Snippet –Login and Registration

2. Food menu

The Admin can edit food orders.

```php
<?php
include '../includes/connect.php';
foreach ($_POST as $key => $value)        {
        if(preg_match("/[0-9]+_name/",$key)){
                if($value != ''){
                $key = strtok($key, '_');
                $value = htmlspecialchars($value);
        $sql = "UPDATE items SET name = '$value' WHERE id = $key;";
                $con->query($sql);
                }
        }
        if(preg_match("/[0-9]+_price/",$key)){
                $key = strtok($key, '_');
                $sql = "UPDATE items SET price = $value WHERE id = $key;";
                $con->query($sql);
        }
        if(preg_match("/[0-9]+_hide/",$key)){
                if($_POST[$key] == 1){
                $key = strtok($key, '_');
                $sql = "UPDATE items SET deleted = 0 WHERE id = $key;";
                $con->query($sql);
                } else{
                $key = strtok($key, '_');
                $sql = "UPDATE items SET deleted = 1 WHERE id = $key;";
                $con->query($sql);
```

Code Snippet – view candidate details

3. View order

Customers can view all orders made

```php
$address = htmlspecialchars($_POST['address']);

$description =  htmlspecialchars($_POST['description']);

$payment_type = $_POST['payment_type'];

$total = $_POST['total'];

$sql = "INSERT INTO orders (customer_id, payment_type, address, total, description) VALUES ($user_id, '$payment_type', '$address', $total, '$description')";

if ($con->query($sql) === TRUE){

$order_id =  $con->insert_id;

foreach ($_POST as $key => $value)

{

        if(is_numeric($key)){

        $result = mysqli_query($con, "SELECT * FROM items WHERE id = $key");

        while($row = mysqli_fetch_array($result))

        {

                $price = $row['price'];

        }

                $price = $value*$price;

        $sql = "INSERT INTO order_details (order_id, item_id, quantity, price) VALUES ($order_id, $key, $value, $price)";

        $con->query($sql) === TRUE;

        }

}
```

Code Snippet – To view order detail

4. View tickets

A page to view all customer support tickets.

```php
include '../includes/connect.php';

include '../includes/wallet.php';

$message = htmlspecialchars($_POST['message']);

$ticket_id = $_POST['ticket_id'];
```

```php
$role = $_POST['role'];
if($role == 'Administrator'){
$sql = "UPDATE tickets SET status = 'Answered' WHERE id=$ticket_id;";
$con->query($sql);
}
else{
$sql = "UPDATE tickets SET status = 'Open' WHERE id=$ticket_id;";
$con->query($sql);
}
if($message != ''){
$sql = "INSERT INTO ticket_details (ticket_id, user_id, description) VALUES ($ticket_id, $user_id, '$message')";
$con->query($sql);
}
header("location: ../view-ticket.php?id=".$ticket_id);
```

Code Snippet – View tickets

5. Cancel order.

To cancel any pending orders.

```php
<?php
include '../includes/connect.php';
include '../includes/wallet.php';
$status = $_POST['status'];
$id = $_POST['id'];
$sql = "UPDATE orders SET status='$status', deleted=1 WHERE id=$id;";
$con->query($sql);
$sql = mysqli_query($con, "SELECT * FROM orders where id=$id");
while($row1 = mysqli_fetch_array($sql)){
$total = $row1['total'];
}
if($_POST['payment_type'] == 'Wallet'){
$balance = $balance+$total;
```

```
$sql = "UPDATE wallet_details SET balance = $balance WHERE wallet_id =
$wallet_id;";

$con->query($sql);

}

header("location: ../orders.php");
```

<p align="center">Code Snippet – Cancel orders</p>

6. Edit orders

   To edit any orders

```
<?php

include '../includes/connect.php';

$status = $_POST['status'];

$id = $_POST['id'];


$sql = "UPDATE orders SET status='$status' WHERE id=$id;";

$con->query($sql);


header("location: ../all-orders.php");

?>
```

<p align="center">Code Snippet – Edit and delete the question</p>

4.2   RESULT

The resulting system is able to:

1. Authenticate credentials during login

2. Allow admin to view order details

3. Allow admin to add a new food item and edit it.

4. A customer can add and pay for orders.

5. All user details can be edited.

6. Customers can raise complaints if there are problems with orders.

CHAPTER 5

TESTING

5.1   SOFTWARE TESTING

Testing is the process used to help identify correctness, completeness, security and quality of developed software. This includes executing a program with the intent of finding errors. It is important to distinguish between faults and failures. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. It can be conducted as soon as executable software (even if partially complete) exists. Most testing occurs after system requirements have been defined and then implemented in testable programs.

5.2   MODULE TESTING AND INTEGRATION

Module testing is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program. It is largely white box oriented. The objective of doing Module testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module testing allows implementing of parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

In the Food Ordering System, the user login module accepts email and password and the same is validated with database entries. Suitable error messages are displayed for false information. The customer details can be added and if one field is empty while providing the details of an error message. The details of the customers can be viewed. A food order can be added then can be edited, deleted and viewed. The past orders can be viewed. All the modules are individually tested before integrating for unique test cases.

The final integrated system too has been tested for various test cases such as duplicate entries and type mismatch.

5.3   LIMITATIONS

- ◉ The current system is only for admin and customers and not for delivery side.

- ◉ Requires active internet connection if implemented for a database outside local network.

- ◉ User's session is not kept for continuous login

- ◉ No map of user locations.

CHAPTER 6

SNAPSHOTS

This chapter consists of working screenshots of the project.

6.1 Login

Following snapshot is the login page of our system, using which admin and customers can login. If the user is not registered then he can register and then login.
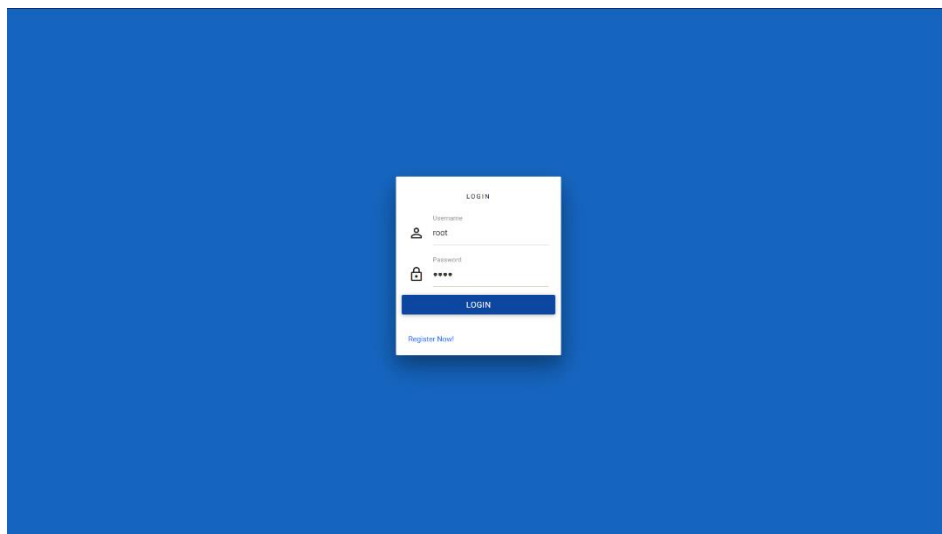
Fig 6.1 Login
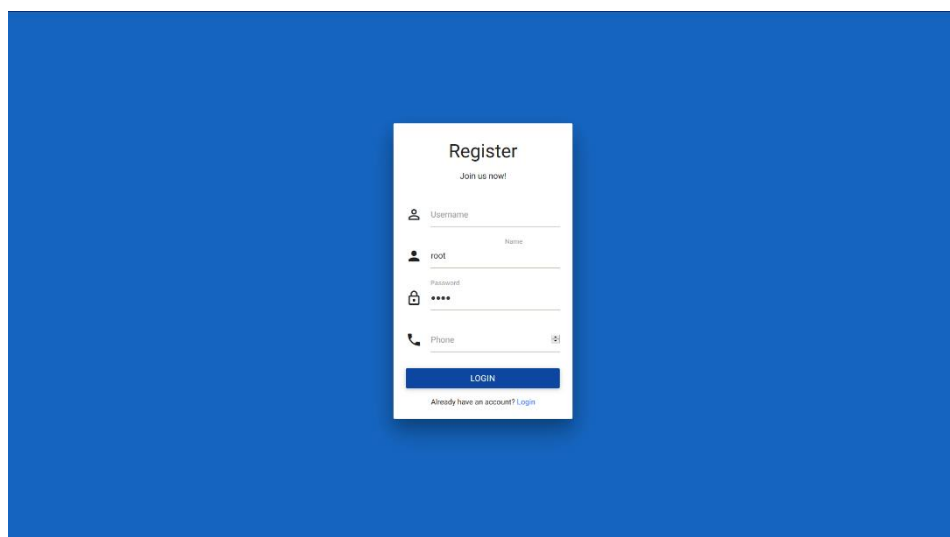
Fig 6.2   Registration page

6.2 Food Menu

Snapshots in the following page contain the home page where the food menu can be viewed and edited
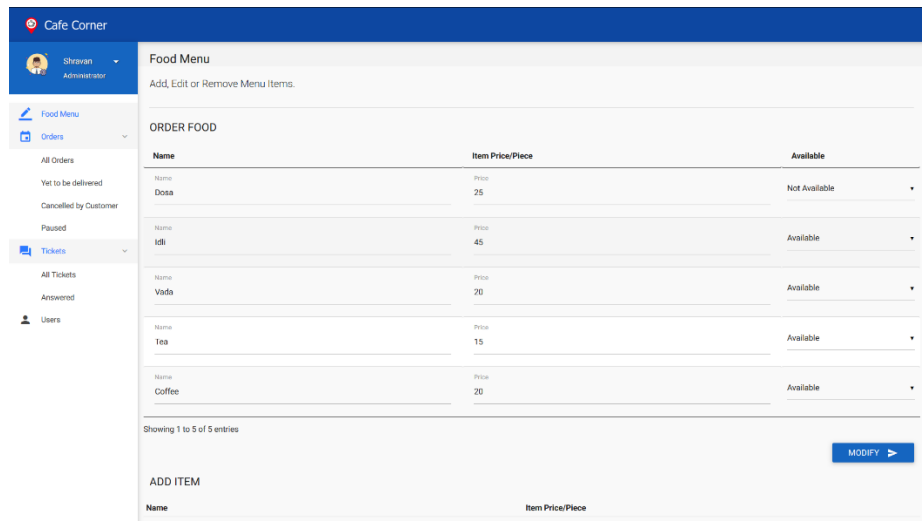
Fig 6.3 Home page where the food orders can be made

6.3 Support List

A support list for to chat with customers and provide a chat
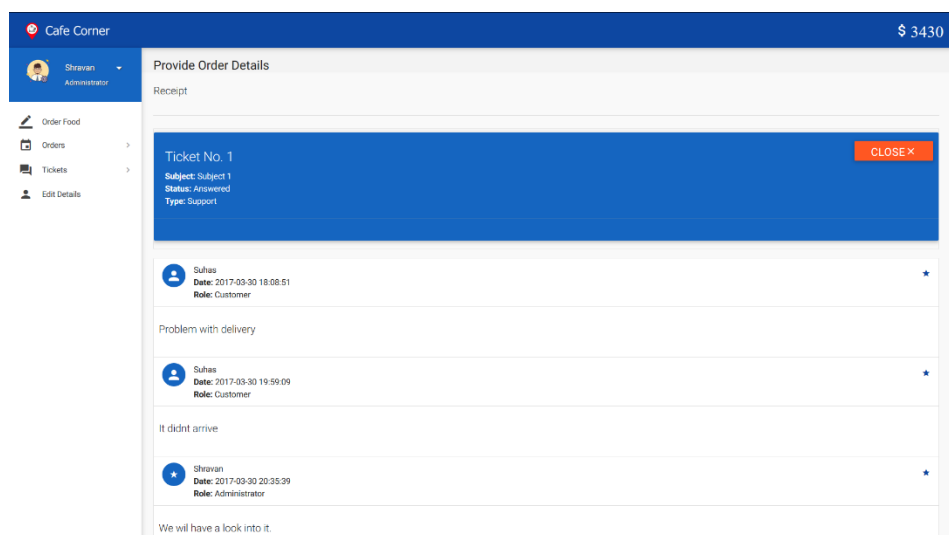
Fig. 6.4 Support ticket

6.4   USERS LIST

A list of all users for admin to manage

Fig 6.5 Users list

6.5   PAST ORDERS

The list of past orders for a customer

Fig 6.6 Past orders

6.6 New Orders

The new orders of a customer can be given here.

Fig 6.7 New Order

CHAPTER 7

CONCLUSION

The Food Ordering System provides easier maintenance of candidate's data and makes the job easier. It allows simplified operation and is a time saving platform. The application has been completed successfully and tested with suitable test cases. This is developed using HTML5, CSS, JavaScript, PHP, Materialize, Bootstrap and SQL. The goals achieved by this project are:

1. Easy food ordering
2. User friendly tool
3. Single centralized database
4. Efficient management of records
5. Support system for users

CHAPTER 8

FUTURE ENHANCEMENTS

In future, we can implement the following features:

- Integration of payment service – Move from a mock payment service to actual integration of services
- Modifying food orders – Add a better system for modification and cancellation of food after ordering.
- Map services – Add a service for seeing in map the location of user to make delivery easier.

REFERENCES

[1] Eric Meyer. CSS: The Definitive Guide, 3rd Edition

[2] Steven Holzner. The Complete Reference: PHP, First Edition

[3] Ramakrishnan, R., & Gehrke, J. (2011). Database management systems. Boston: McGraw-Hill.

[4] Materialize documentation - https://materializecss.com/getting-started.html

[5] PHP Tutorials - https://www.w3schools.com/php/

[6] David F. (2011). JavaScript: The Definitive Guide Sixth edition.