

ONLINE FOOD ORDER SYSTEM



Estd. 1919

Dissertation submitted to

National college (Autonomous), Tiruchirappalli

(Affiliated to **Bharathidasan university**)

In partial fulfilment of the requirements for the award of

Bachelor of Computer Applications

Submitted By

A.VISHWANATH

Reg No: BCUS18068

Under the guidance of

Dr. M.ANUSHA M.Sc., M.Phil., M.Tech., Ph.D.,

Assistant Professor & Head, BCA ,

PG & RESEARCH Department of Computer Science,

National COLLEGE (Autonomous),

(Nationally Accredited at 'A' Level by NAAC)

(College with potential for Excellent by UGC)

Tiruchirappalli-620001.

MARCH-2021

Dr. M.ANUSHA M.Sc., M.Phil., M.Tech., Ph.D.,

Assistant Professor & Head, BCA,

PG & Research Department of Computer Science,

NATIONAL COLLEGE (Autonomous),

(Nationally Accredited at “A” Level by NAAC)

(College with potential for Excellence by UGC)

Tiruchirappalli-620001.

CERTIFICATE

This is to certify that the dissertation entitled “**ONLINE FOOD ORDER SYSTEM**”
submitted in partial fulfilment of the requirements for the Degree of Bachelor of Computer
Applications for the academic year 2021 to the National college (Autonomous),
Tiruchirappalli (Affiliated to Bharathidasan university, Tiruchirappalli) in a Bonafide project
of research work Carried out by

NAME
A.VISHWANATH

REGISTER NUMBER
BCUS18068

Under the guidance and this being the original work of this candidate.

Head of the Department

Project Guide

Examiners :

1.

Date: 31.03.2021

2.

A.VISHWANATH

DEPARTMENT OF COMPUTER APPLICATIONS

NATIONAL COLLEGE (Autonomous)

Tiruchirappalli-620 001

DECLARATION

I declare that the dissertation entitled “**ONLINE FOOD ORDER SYSTEM**” submitted by me for the Degree of Bachelor of Computer Applications is the record of work carried out by us under the guidance **Dr. M.ANUSHA M.Sc., M.Phil., M.Tech., Ph.D., Assistant Professor & Head, BCA, PG & Research Department of Computer Science, National College (Autonomous), Tiruchirappalli** and this has not formed the basis for the award of any other project titles.

Place: Tiruchirappalli

Date: 31.03.2021

Signature of the candidate

ACKNOWLEDGEMENT

First and foremost I thank the lord and my parents for valuable support to complete this project work successfully.

I express my highest and sincere gratitude to our respected **Principal Dr. R.SUNDARARAMAN M.Com., B.Ed., Ph.D.,**

I express my sincere thanks to our respected **Vice – Principal & Head of the Computer Science Dr. P.S.S.AKILASHRI M.Sc., M.C.A., M.Phil., B.Ed., Ph.D.,** her guidance throughout the course for the benefit to us.

I express my sincere thanks to our respected guide & **Head of Computer Applications Dr. M.ANUSHA M.Sc., M.Phil., M.Tech., Ph.D.,** for his valuable guidance, encouragement, criticisms and suggestions at each stage of this project work.

Finally I extend my gratitude of the staff members of our department for their support.

A.VISHWANATH

Table of Contents

Table of Contents	5
List of Code Snippets	8
List of Figures	8
Abstract.....	9
1. Introduction	10
1.1. Project Overview	10
1.2. Project Aim & Objectives	11
1.3. Project Rationale	11
1.4. Feasibility of the Project	13
2. Literature Review	14
2.1. Past Works on This Domain	14
2.1.1. Web Application Based System	14
2.1.2. Evaluation of the Past Works	15
2.2. Web Application Development Technologies	16
2.3. PHP	16
2.4. JavaScript	27
2.5. HTML	28
3. Methodology	22
3.1. Research Methodology	22
3.1.1. Initial Study	23
3.1.2. Data Collection	24
3.1.3. Implementation	24
3.1.4. Analysis	24
3.1.5. Critical Analysis of the System	24
3.2. System Development Methodology	25
3.2.1. Waterfall	25
3.2.2. System Prototyping	26
3.2.4. Rational for Selected SDLC	26
4. Requirements & Design	29
4.1. Requirements Elicitation	29

4.1.1. Functional Requirements	29
4.1.2. Non-Functional Requirements	30
4.2. Process Modelling	30
4.2.1. Context Diagram (CD)	30
4.2.2. Data Flow Diagram (DFD)	31
4.3. Data Modelling	34
4.3.1. Entity Relationship Diagram (ERD)	34
4.4 User Interface Design	35
4.4.1. Homepage	36
4.4.2. User Details	37
4.4.3. Admin Details	43
5. Implementation	48
5.1. Views	48
6. Testing	55
6.1. Black Box Testing	55
6.1.1. Test 1	55
6.1.2. Test 2	56
6.1.3. Test 3	57
6.1.4. Test 4	58
6.1.5. Test 5	59
6.2. User Acceptance Testing	60
6.2.1. Ease of use	60
6.2.2. User Friendly	61
6.2.3 Responsiveness of Interface	62
6.2.4. Speed & Performance	63
7. Results & Evaluations	64
7.1. Security of the System	64
7.2. Responsive User Interface	65
8. Conclusions and Recommendations	66
8.1. Achievement of the Project	66
8.2. Limitations of the System	66
8.3. Future Recommendations	67
8.4. Personal Reflection	67
8.4.1. Acquiring Research Skills	67

8.4.2. Acquiring Programming Skills	67
8.5. Concluding Remarks	68
9. Bibliography	69
10. Appendices	71
10.1. Appendix A: Code Snippets	71

List of Code Snippets

Snippet 1: Main view	49
Snippet 2: Food Cart	51
Snippet 3: Add food form	53
Snippet 4: Main Home Page.	71
Snippet 5: User Home Page	91
Snippet 6: Database Connection	99

List of Figures

Figure 1: MVC pattern ecosystem	20
Figure 2: Steps of the research methods in Waterfall model	23
Figure 3: Phases of Extreme Programming	27
Figure 4: Context diagram of the system	31
Figure 5: DFD of processes of customer for ordering food	32
Figure 6: DFD for processes of Admin	33
Figure 7: Entity relationship diagram of the system	34
Figure 8: Difference of screen layouts for different size of screens	35
Figure 9: Homepage	36
Figure 10: User Login	37
Figure 11: User Register	38
Figure 12: User Panel	39
Figure 13: Edit Profile	40
Figure 14: User change password	41
Figure 15: User Order Detail	42

Figure 16: Admin home page	43
Figure 17: User register display	44
Figure 18: User comment display	45
Figure 19: User order detail display	46
Figure 20: Chef details	47
Figure 21: Now menu shown in database	57
Figure 22: The image of the new menu upload to the folder	58
Figure 23: Menu list page	59
Figure 24: Add food form page	53
Figure 25: Showing total for one menu	55
Figure 26: Total multiple men	56
Figure 27: Ease of use of the application	60
Figure 28: Percentage of view on user friendliness	61
Figure 29: Percentage of view on responsiveness	62
Figure 30: Satisfaction about speed and performance	63



ABSTRACT

The Online Food Ordering System described in this document has been designed to fill a specific niche in the market by providing small restaurants with the ability to offer their customers an online ordering option without having to invest large amounts of time and money in having custom software designed specifically for them. The system, which is highly customizable, allows the restaurant employees to easily manage the site content, most importantly the menu, themselves through a very intuitive graphical interface. The website, which is the only component seen by the restaurant customers, is then built dynamically based on the current state of the system, so any changes made are reflected in real time. Visitors to the site, once registered, are then able to easily navigate this menu, add food items to their order, and specify delivery options with only a few clicks, greatly simplifying the ordering process. Back in the restaurant, placed orders are promptly retrieved and displayed in an easily readable format for efficient processing. The purpose of this document is to provide in-depth descriptions of design and implementation details of the system, as well as descriptions of all available functionality and plans for evolution. In addition, user manuals and trouble-shooting tips have been included for all three components to give the reader a clear idea of intended typical use cases for the system.

CHAPTER 1

INTRODUCTION

This chapter starts with a high-level overview of the project. It then describes the specific aims and objectives of the project. Finally, it analyses the feasibility of the project and provides with a feasibility report of the system.

1.1. Project Overview

The project sets to develop an online ordering system for restaurant. Many industries now quickly adopting technologies. Restaurant industry also embraces different types of technologies which make daily processes easier and faster. But the adoption of technology in restaurant industry is slower compared to other industries (Saeed *et al.*, 2016). Restaurants can use technology in different ways. One of them is to build an online presence by a web application which may also help in online ordering. Because the use of internet-based ordering system is in rise in today's world. Since 2014, about 30% of customers aged between 18 and 54 have ordered food from a restaurant's website via their phone or tablet in USA. On the other hand, QSR Web found that digital restaurant ordering is growing 300% faster than dine-in traffic (Bultis, 2016). So, the market for online food delivery now remains as large as ever. Nevertheless, a huge section of this market is still being overwhelmingly offline. When it comes to buying goods or foods online, customers want to be ensured about the quality of the foods that s/he is going to buy. People tend to eat foods that is suggested by their friends or nearest one. This thing can be achieved in an online food ordering system by integrating social media properly. A study performed by OTX Research found that two out of every three customers are influenced in their buying decision through information they find in social media. Nearly, 67% of customers are likely to pass this information on to other customers and over 60% of customers trust information they find through social media such as Facebook, Twitter and other social media sites (Shankararaman and Lum, 2013). Suraworachet et al. (2012) studied the effect of Facebook's features such as "like" and observed a positive impact on consumer's attitude to buying products. In summary, social media has served as a new form of word-of-mouth and influence customer decision-making while buy something online. So, social media will be implemented with the application. Facebook API will be used with the system, so that customers can login with their Facebook credential and like any menu. Along with this, Twitter sharing system also integrated to the system. The ubiquitous connectivity and

internet make different kinds of devices interact to web applications. A web application now can be accessed from different types of devices including desktop personal computer, laptop or notebook PCs, portable tablets and smartphones. So, a modern web application should be optimized for every kind of devices. But the existing restaurant websites are not in a good level. For instance, 51 percent of all mobile searches on Google are for restaurants, yet as few as 5 percent have mobile-optimized websites (Oddly, 2016).

1.2. Project Aim & Objectives

The project aims to build a web-based system for restaurant, which automates food ordering system. It will also help the management to manage the online orders and view the status. The management can add menus and take orders with the system. The system also has a simple mobile-friendly user interface which can be used through different types of devices and screens. Facebook API will be integrated with the application so customers can login with their Facebook account and like or share menus which can work a new customer as word of mouth.

In order to achieve the mentioned aim, following objectives should be achieved:

1. An extensive literature review will be conducted to find out the past work done to automate the food ordering process of restaurant.
2. Appropriate methods and methodologies will be used for designing and developing the whole system in systematic approach.
3. Laravel 5.4 and Bootstrap 4 will be used for developing the backend and frontend of the system.
4. Social Media will be integrated to the system. So, customer can login and register through their social media account and share their views about menu.
5. The system will be tested properly to ensure the quality of the system.

1.3. Project Rationale

Online ordering from restaurants is getting easier as mobile and internet technology progresses. Food lovers now tends to order foods through online ordering system as it is easier and fast. QSR Web found that digital restaurant ordering is growing 300 percent faster than dine-in traffic. But most of the restaurant do not give concentration to make solid appearance

on the internet. As a result, 51 percent of all mobile searches on Google are for restaurants, yet as few as 5 percent have mobile-optimized websites (bultis, 2016).

There are also some problems of traditional food ordering system, which results wasting time and making conflicts. In existing system there are few problems:

- ❖ For placing orders, customers have to visit restaurants to know about food items and then place order and pay. In this method time and manual work is required.
- ❖ Some restaurants take order via phone. While placing an order over the phone, customer lacks the copy of available menu items in restaurant.
- ❖ Every restaurant needs certain employees to take the order over phone or in-person. In today's market, labor rates are increasing day by day making it difficult to find employees when needed (Patel, 2015).

To solve these issues, an Online Food Order System has been developed which is originally designed for small scale business. But this system is applicable in any restaurant. The main advantage of this online system is that it greatly simplifies the food ordering process for both of the customer and the restaurant. The anticipated benefits of the project are:

1. This will speed up the ordering process.
2. The system will help to reduce labor cost involved.
3. This will avoid long queues at the counter due to the speed of execution and number of optimum screens to accommodate the maximum throughput.
4. The system will be less probable to make mistake, since it's a machine.
5. The top benefit of online ordering was a savings in labor, since employees are not tied up on the phone or at the counter.
6. Order accuracy was another benefit for restaurant (Kites, 2011).

On the other hand, social websites and media play a vital role for business. Every business can use social media to get information about potential customers, analyses sentiments and better relationship with customers. A study by Algos and Hekimoglu (2012) shows that trust is one of the main drivers to place order online in restaurant. Today, a recommendation from friends via social media plays as a word of mouth in building trust while ordering online

(Shankararaman and Lum, 2013). Hence, Social media can be integrated to restaurant web app to ensure customers about the food quality. New customers can choice their desired products by studying reviews made by other users.

1.4. Feasibility of the Project

The project requires a complete web application for restaurant that can be used to manage online orders. This will be mobile-optimized web app, so that it can be viewed in a mobile browser as well as in a PC. All the technologies that will be used to develop the system are open-sourced, such as PHP, Bootstrap 4, and Apache Server. Hence, the technologies can be used without any cost.

CHAPTER 2

LITERATURE REVIEW

This chapter discusses the different approaches for automating the ordering processes of restaurant. After that, it explains the web application development tools and technologies used in this project.

2.1. Past Works on This Domain

A computerized restaurant system is “an integrated IT system that supervises, manages and facilitates the planning operations in restaurant” (Tan, 2013). Before the automated system has been introduced, the orders and payments were managed manually using register books etc. The Point-of-Sales (POS) was introduced in the era of 1990. After the system got popularity, it started being used in restaurant industry. After 1990, the internet and wireless technology moved on so much fast. Restaurants also started using different technology-based system to increase the efficiency of the system (Sullivan, 2015). Researchers also started introducing different types of solutions for restaurants. Lots of work have been done for automating the operations of the restaurants. Every researcher chose an aspect or problem and tried to make good solution for that. The solutions provided by researchers can be categorized in different categories. The following sections will discuss about those systems.

2.1.1. Web Application Based System

To overcome the device and operating system centric approach for automating restaurant operation, K *et al.* (2016) proposed an online web application which can be accessed through internet. For using this application, computer screen will be placed on each table of restaurant for customers to order. Customers can order food for take away and make payment through the application. However, the idea is innovative, but a little bit weird as computers need to be installed in every table. Sometimes it may create complex problem rather than making a solution. Patel (2015) developed an online food ordering system with Java in backend system and HTML with CSS in frontend. It was a robust and nice application. But the responsive development method was not used in the application. Hence, the application cannot viewed properly with a smart phone. Tan (2013) developed a web based application with ASP .Net and Bootstrap to automate the ordering process of the restaurant and make it paperless.

But the system is only focused on the management of the restaurant. There is no option for customers to order food online. The system was responsive.

These web applications, however, overcame the problem of being dependent on specific device (e.g., PDA) or operating system. These systems also have few limitations, such as:

- ❖ None of them tried to make the online ordering system more trustable to customers.
- ❖ They did not integrate social media to the system which can make customer registration and login process easier.

2.1.2. Evaluation of the Past Works

The previous lines illustrate that many researchers worked for implementing IT based systems in restaurants to gain more revenue and reduce order processing time. But most of them did not focus on the online based restaurant system where customers can order food over internet from their home. With the expansion of internet technology in this era, it is not feasible to stick with only technologies demonstrated above as most of the proposed system only tried to automate the in-house ordering system.

On the other hand, the systems which is developed to order food remotely are device or operating systems oriented. For obtaining the benefits of the system, customers need to use a smartphone with specific operating system like Windows or Android. No one tried to find a solution which does not depend on any specific device, while a web application can be accessed by through kind of devices with internet connection from anywhere. Moreover, there is no need for installing extra software for browsing a web application. If the web application is developed using modern mobile friendly manner properly, it will give a good performance on any kind and size of device. The proposed solutions also lack the use of the power of social community, which can be a big marketing medium for the restaurant and help influencing new customers to have a food experience in the restaurant. The earlier part of this paper stated that the digital restaurant ordering is growing 300 percent faster than dine-in traffic (Beltis, 2016). So, the restaurants should embrace internet technology to make a solid appearance on the internet and provide the service in a wider range of customers within minimum cost and time.

2.2. Web Application Development Technologies

The backend of the system is developed with PHP and Laravel and the frontend of the system is developed with Bootstrap 4. Hence, this section is providing a brief description of these technologies used for this project.

2.3. PHP

PHP Introduction:

PHP code is executed on the server.

What is PHP?

- 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

PHP is an amazing and popular language!

- It is powerful enough to be at the core of the biggest blogging system on the web (WordPress)!
- It is deep enough to run the largest social network (Facebook)!
- It is also easy enough to be a beginner's first server-side language!

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server

- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

What's new in PHP 7

- PHP 7 is much faster than the previous popular stable release (PHP 5.6)
- PHP 7 has improved Error Handling
- PHP 7 supports stricter Type Declarations for function arguments
- PHP 7 supports new operators (like the spaceship operator: `<=>`)

2.4 JavaScript

JavaScript (JS) is a lightweight, interpreted, or [just-in-time](#) compiled programming language with [first-class functions](#). While it is most well-known as the scripting language for Web pages, [many non-browser environments](#) also use it, such as [Node.js](#), [Apache CouchDB](#) and [Adobe Acrobat](#). JavaScript is a [prototype-based](#), multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. Read more [about JavaScript](#).

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about [API](#) specifics to Web pages, please see [Web APIs](#) and [DOM](#).

The standard for JavaScript is [ECMAScript](#). As of 2012, all [modern browsers](#) fully support ECMAScript 5.1. Older browsers support at least ECMAScript 3. On June 17, 2015, [ECMA International](#) published the sixth major version of ECMAScript, which is officially called ECMAScript 2015, and was initially referred to as ECMAScript 6 or ES6. Since then, ECMAScript standards are on yearly release cycles. This documentation refers to the latest draft version, which is currently [ECMAScript 2020](#).

Do not confuse JavaScript with the [Java programming language](#). Both "Java" and "JavaScript" are trademarks or registered trademarks of Oracle in the U.S. and other countries. However, the two programming languages have very different syntax, *semantic, and use*.

2.5 HTML

HTML (Hyper Text Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation ([CSS](#)) or functionality/behavior ([JavaScript](#)).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

2.6 Laravel

Today, web applications are mostly developed using a framework. Surguy (2014) defines PHP framework as follows: "A PHP Web application framework is a set of classes, libraries or components written in PHP server-side scripting language that aim to solve common web development problems and promote code reuse." Frameworks are so popular precisely because they dramatically decrease the amount of work (Gilmore, 2015). Laravel is a PHP open source framework that prefers the rapidness of development. The framework is

developed based on Model-View-Controller (MVC) design pattern. Along with this, the framework also implements Object Oriented Programming (OOP), Dependency Injection (DI) and Object Relational Mapping (ORM) (Bean, 2015). The following lines briefly describe the features of the framework.

- **Interaction with Database:** There are three techniques to interact with databases in Laravel: (1) Eloquent ORM, (2) Query builder (3) Raw SQL. Laravel supports running raw SQL commands which is the traditional way for writing SQL within PHP code. The Eloquent ORM of Laravel provides Active Record implementation for working with database. Each database table has a corresponding "Model" which is used to interact with that table. Models allow to run query for data in database tables, as well as insert new records into the table. On the other hand, Laravel's database query builder provides an interface to creating and running database queries. It can be used to perform most database operations on all supported database systems (Bean, 2015). The Laravel query builder uses PDO parameter binding to protect the application against SQL injection attacks. There is no need to clean strings being passed as bindings (Otwell, 2018b). As we have seen that there are three methods for interaction with database, it is important to select the right database technique when developing a web application because there are pros and cons with each approach. Jound and Halimi (2016) made a performance comparison between Eloquent ORM and raw SQL and concluded that Eloquent ORM doesn't give better performance and response time compared to Raw SQL. So, Eloquent ORM is not suitable for the large applications with a huge amount of data. Instead, Eloquent is good for building small to medium sized applications.
- **Model-View-Controller Pattern:** In MVC pattern, the view represents a page's template or templates, the controller represents both the server's dispatch infrastructure that maps a URL to a code snippet and the code snippet itself, and the model represents an application's data (the "state"), most of the business logic, and any model-related computations (Parr, 2004). MVC was first introduced in Smalltalk'80 by Krasner and Pope (Jailia *et al.*, 2016). Today, MVC is one of the most popular architectural design pattern for web application. Laravel is a fully MVC compliant (Dockins, 2016).

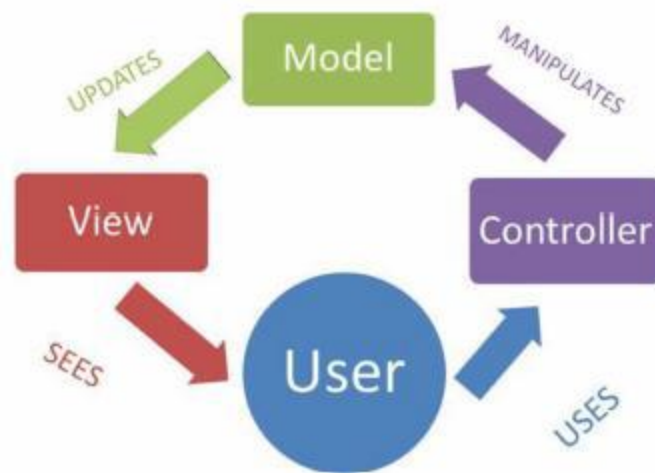


Figure 1: MVC pattern ecosystem (Jailia et al., 2016).

- **Blade Templating Engine:** MVC based Web applications often implemented by using HTML templates, often seeking to separate the webpage presentation from the business logic and objects. It has become a de facto standard programming model for Web application development. Template Engines are used in web applications to separate the front-end webpage presentation from its underlying business logic and objects (Tatsubori and Suzumura, 2009; McFarlin, 2017). Blade is a templating engine provided with Laravel. Unlike other popular PHP templating engines, Blade does not restrict developers from using plain PHP code in views. All Blade views are compiled into plain PHP code and cached until they are modified (Otwell, 2018).
- **Built-In Security:** Since authentication is a common thing for every application Laravel comes with built-in authentication system for users which can be used for user registration, login, password reset etc. The authentication process is easily customizable to the need of the application and business logic. There are many built-in features in Laravel which makes application development faster without compromising the security and quality of the application (Otwell, 2018c).
- **Performance:** Olanrewaju, Islam and Ali (2015) evaluated the performance of the four popular PHP frameworks; CodeIgniter (CI), Symfony, CakePHP and Laravel. The frameworks are benchmarked on several criteria such as request per second, system load average, memory usage, number of function calls and number of files required.

Laravel gets the best result over other three frameworks in request per second, system load average, memory usage and number of function calls. Das and Saskia (2016) made a comparison between raw PHP, CodeIgniter (CI) and Laravel frameworks where Laravel performed better than raw PHP and CodeIgniter (CI).

CHAPTER 3

METHODOLOGY

This chapter of the report aims to provide a clear vision on methods and methodologies used for this project. In first, it describes the methodology and methods selected for the research and gives rational why these methods and methodology have been used. Then it discusses about software development methodology and give rational for it.

3.1. Research Methodology

Research is “a scientific and systematic search for pertinent information on a specific topic” (Kothari, 2004). The research methodology is a process used to collect information and data for the purpose of making decisions. There are broadly two approaches for conducting a research. Those are: 1) Quantitative analysis 2) Qualitative Analysis. For developing the system, combined method has been applied. A systematic approach should be used for conducting any research to find the solution of the problem in systematic way. For this purpose, the research methods have been applied based on waterfall model.

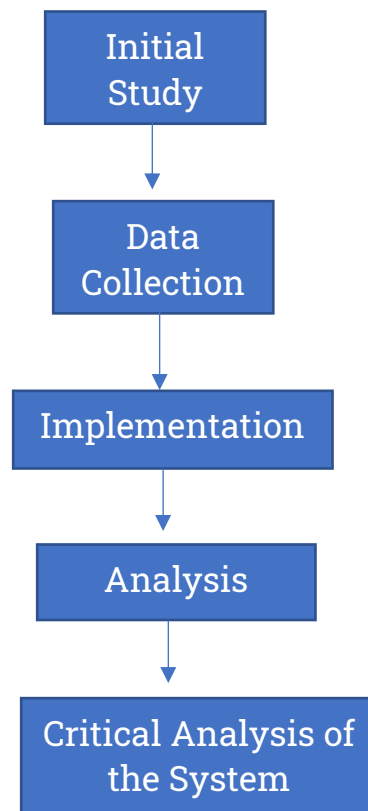


Figure 2: Steps of the research methods in Waterfall model.

There are different models for research. Among them, waterfall model is chosen as it is easy to understand and implement. Hence, many beginners use this methodology for research (Dawson, 2009). Hence, this methodology is used for research in this project. Following steps have been used to formulate the hypothesis and determining project requirements: 25

3.1.1 Initial Study

This is the first step of the research. In this step, different topics and subject area has been reviewed to find problem. After setting the goal to developing restaurant system, the study has been narrowed to online food ordering system.

3.1.2. Data Collection

Once the goal was set, data collection process began to get information about the online food ordering system history and past works on this domain. Following methods have been used for data collection:

- **Literary Analysis:** Literatures have been reviewed and critically analyzed to find what kind of works have been suggested in past. These works helped in finding the shortcomings of past solutions and defining the aim and objectives of the project. Mostly, primary data has been collected to serve the purpose. But in some cases, secondary data is also used.
- **Observation:** Number of restaurant websites have been reviewed to find the current trend in developing web application for restaurants. Along with this, typical restaurant ordering system has been reviewed to get an insight of online order processing system.

3.1.3. Implementation

After getting the logical structure of the system, the implementation step began. In this step, the logical structure converted to physical architecture through coding and development of the system. The frontend and backend of the system has been developed and tested.

3.1.4. Analysis

After data collection, an overview on the system has been gained. Then different methods (Data Flow Diagram, Entity Relationship Diagram etc.) of Structured System Analysis & Design Methodology (SSADM) is used to analyses the system and make logical structure for it.

3.1.5. Critical Analysis of the System

After implementation, the system has been critically evaluated to understand if it meets the requirements. Then the result is analyzed. In this step, conclusion and future recommendation has been made.

3.2. System Development Methodology

For developing any information system, a System Development Methodology should be used which will provide a structured way for development of an IT based systems. SDLC refers to System or Software Design Life Cycle. It is phases of processes taken down to build a system properly. The main aim of SDLC process is to help provide a system that is effective, cost-efficient, and of high quality. SDLC methodologies typically has the following stages: Analysis (requirements and design), construction, testing, release, and maintenance (response). But the phases can be changed in deferent SDLC mythologies. There are many software developments models for different types of projects. In following lines, only popular three of them will be discussed.

3.2.1. Waterfall

The waterfall model is a well-known structured methodology for software development. The whole process of system development is divided into distinct phases. The model has been introduced in 1970s. Every phase has a unique output. It was the first SDLC model to be used widely. So that, sometime it is referred to Waterfall by SDLC. The waterfall model is used when the system requirements are well known, technology is understood and the system is a new version of an existing product (Dennis, Wixom and Roth, 2012). Mainly there are six phases in Waterfall model. If there is a problem faced in any phase of the cycle, the system goes to the previous phase. The phases of Waterfall method are:

- ❖ **Requirements Gathering & Analysis:** In this phase, all possible requirements of the system are captured and documented in a requirement specification doc.
- ❖ **System Design:** The requirements documented in previous phase are studied in this phase and the system design is prepared.
- ❖ **Implementation:** With inputs from system design, the system is developed in several unites. Then the units are tested.
- ❖ **Integration & Testing:** The units of the program developed in previous phase are integrated into a system. Then the whole system is tested.

- ❖ **Deployment of the System:** When the all kind of testing is done, the product is deployed in the customer environment.
- ❖ **Maintenance:** There are some issues which are found in the client environment. Patches are released to fix those issues.

3.2.2. System Prototyping

Prototyping is a Rapid Action Development (RAD) method. In this method, the analysis, design and implementation phases performed concurrently and repeatedly in a cycle until the system is completed. With this methodology, the basics of analysis and design are completed. Then the work on the system prototype begins immediately. So that, many bugs and problems remained on the system. After that, the users or project sponsors provide comment on the system. Then, the system is reanalysed, redesigned and implemented based on feedbacks. This process continues until the users or project sponsors satisfied with the system (Gould, 2016).

3.2.4. Rational for Selected SDLC

On the other hand, Waterfall methodology is good for developing complex and reliable system. But its performance is poor when the schedule is short and deadline is visible. The System Prototyping method has an excellent performance with unclear requirements and within short time schedule. So, it could be a good choice for building this restaurant system. But the reliability of product developed with System Prototyping is poor. No one wants to build a system that he or she cannot rely on. In these criteria, Agile Development provides good result. So, the Agile Methodology has been chosen as the software development methodology for the system.

There are many models of implementing Agile Methodology. Extreme Programming (XP) has been chosen. Extreme Programming (XP) model has four stages for developing a project, as the figure illustrates below.

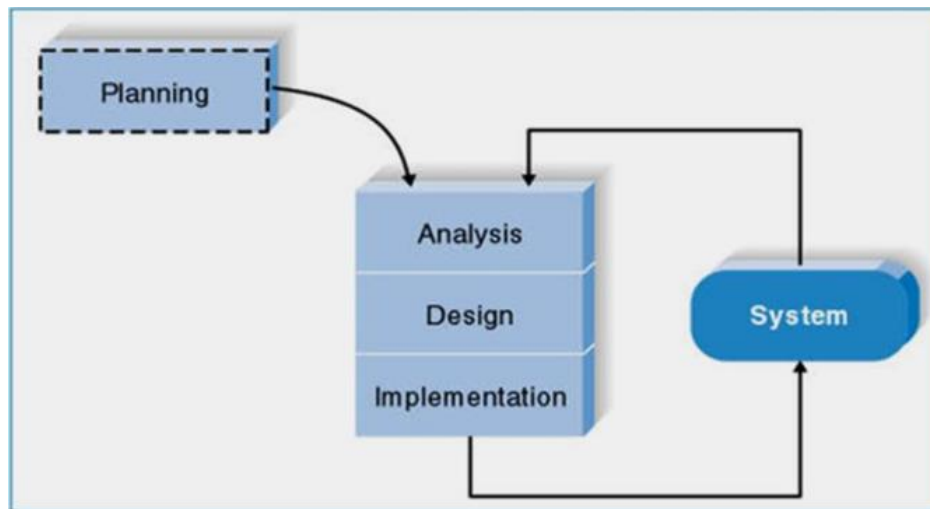


Figure 3: Phases of Extreme Programming (XP) (Dennis, Wixom and Roth, 2012).

As the figure above shows, there are mainly four phases in Extreme Programming method. These phases are discussed below:

- ❖ **Planning:** Extreme Programming starts with the planning stage. In this phase, the requirements for the system have been collected and documented. In this step, the plan, time, and costs of carrying out the iterations is prepared.
- ❖ **Analysis:** In this phase, the logical model of the system has been developed. SSADM is used to make the logical structure of the system. A bottom-up approach is used for analysis of the system, as there is not any previous system. The system will be developed from scratch. Along with these, the user interface requirements are also analyzed.
- ❖ **Design:** In this phase of the SDLC, the logical model of database and the interface of the system is designed. The normalization of the database schema is done in this phase.
- ❖ **Implementation:** In this phase, the system has been implemented through coding. As extreme programming is an iterative method, it is possible to use test driven development method using unit testing. After the system has been developed, the end-to-end testing (black box testing, user acceptance testing) will be used to evaluate the system. It should be kept in mind that analysis, design and implementation phases are

iterative phases. After completing one iteration, feedback has been taken. Then all the phases began for the next iteration.

CHAPTER 4

REQUIREMENTS & DESIGN

This chapter illustrates the approaches taken to design the system for restaurant. The chapter first addresses different types of requirements of the system. Then it discusses about the system design and gives an overview of the systems processes. Then, the database schema of the system is illustrated. Lastly, the user interface design has been developed.

4.1. Requirements Elicitation

The requirements of a system are characteristics of a system it needs to have. The requirements have been collected in the planning phase of the SDLC. Different kinds of data collection methods have been utilized to obtain the requirements of the system, which are explained in *Chapter 3.1.2*.

4.1.1. Functional Requirements

According to International Institute of Business Analysis (IIBA), functional requirements are “the product capabilities, or things that the product must do for its users” (Dennis, Wixom and Roth, 2012). Following are the functional requirements of the project:

- The application must have user registration and login option.
- The application must have registration and sign in option with Facebook API.
- The Application must have a shopping cart for ordering foods.
- The application must have admin registration and login system.
- The application must have password recovery system with email address for users and admins.
- The application must have menu add and edit options for admin.

4.1.2. Non-Functional Requirements

International Institute of Business Analysis (IIBA) defines non-functional requirements as “the quality attributes, design, and implementation constraints, and external interfaces which a product must have” (Dennis, Wixom and Roth, 2012). Following are the non-functional requirements of the project.

- ✓ The application must have a user interface.
- ✓ The user interface must be mobile-friendly.
- ✓ Exception handling methods must be used.
- ✓ Users should get confirmation and warning message.

4.2. Process Modelling

Process modelling is used in a project to depict the processes of data in an application. The restaurant application will be developed and implemented using Model-View- Controller (MVC) design pattern. These processes are mostly implemented as business logic in application controllers. There are different tools for process modelling in SSADM. Context diagram and data flow diagram will be used to model the processes of the system.

4.2.1. Context Diagram (CD)

Context diagrams define “how the business process or computer system interacts with its environment” (Dennis, Wixom and Roth, 2012). Context diagrams are used early in a project to describe the entities of the system. It shows the external entities and data flows into and out of the system. The processes and data stores are not shown in context diagram. It will be shown in data flow diagram.

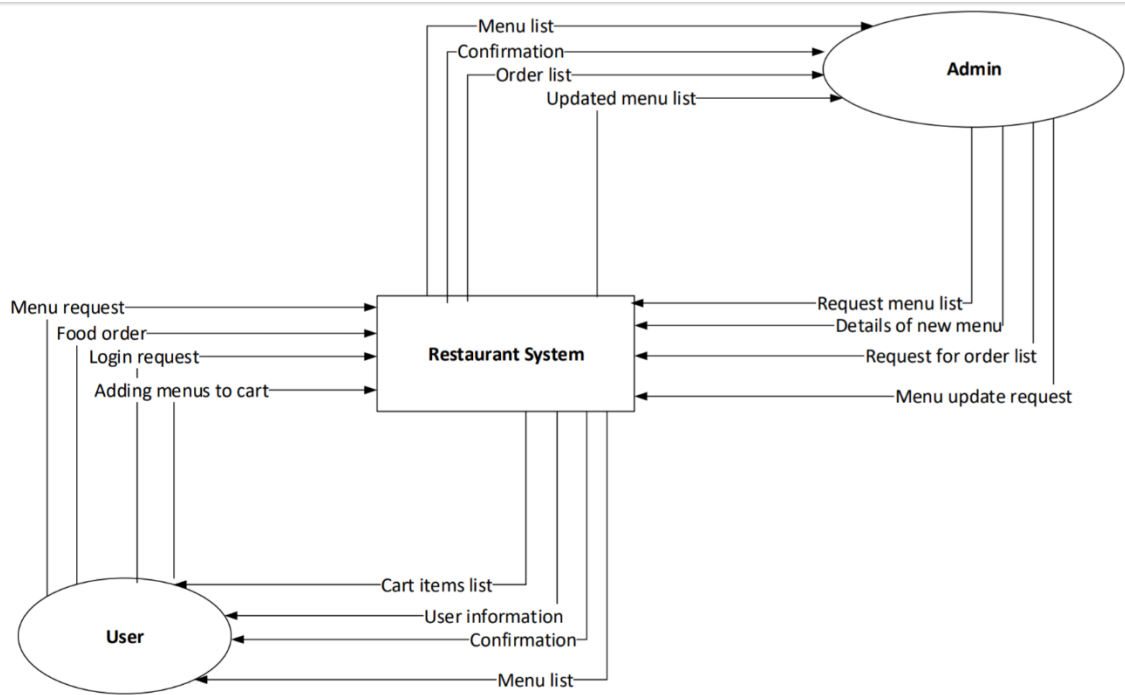


Figure 4: Context diagram of the system.

The above context diagram shows the processes of two entities, User and Admin, with the restaurant system. Both of the entities have four processes with the system.

4.2.2. Data Flow Diagram (DFD)

This is a data flow diagram (DFD), which shows the process of ordering food through online restaurant system. The oval shape means an entity or user. The boxes in the middle of the diagram are process box. The process box shows the number of process and its description. The boxes in the right side of diagram are data store. The box represents when a piece of data is stored. The between the boxes and oval shape shows the flow of data. The main purpose of a DFD is to show the flow of data while doing a process. *Figure 6* shows the DFD of ordering foods process for users.

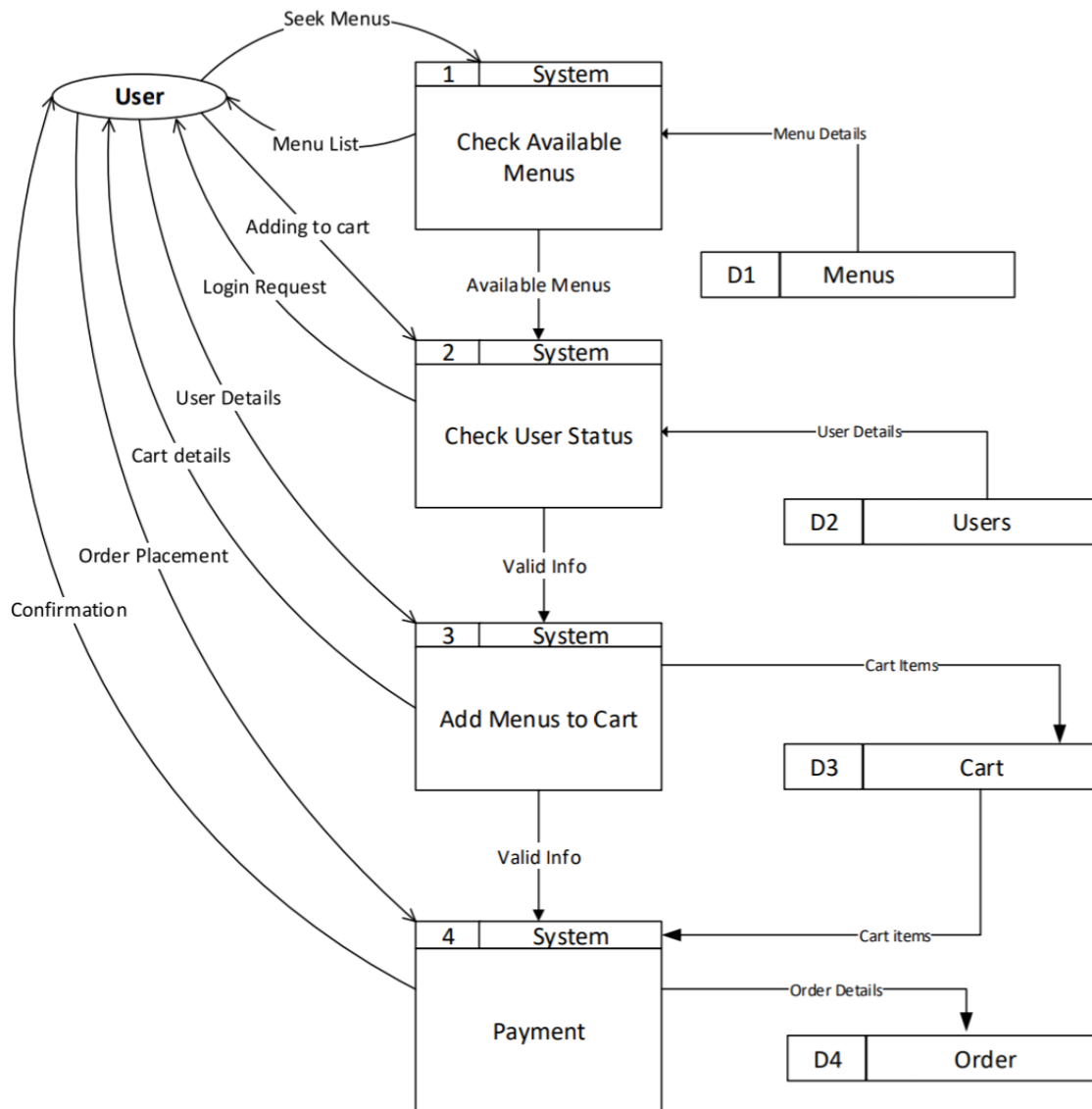


Figure 5: DFD of processes of customer for ordering food.

Process 1 shows that whenever a visitor request the menus page, the system makes a query in the menu table of the database and acquires available menus and returns the menu list to visitor. If visitor likes to add any menu the menu list (shopping cart), s/he hits “Add to Menu” button. Then in Process 2, the systems checks if the user is logged in. If the user is not logged in, the system requests user to log in by redirecting him/her to the log in page. After getting user’s credential, the system compares the credential against the user database. In Process 3, if the user credential matched with the credentials saved in User table in database, the system adds the menu to Cart table and redirects to the cart page. In Process 4, if user orders the menu, the order details saved in the Order table.

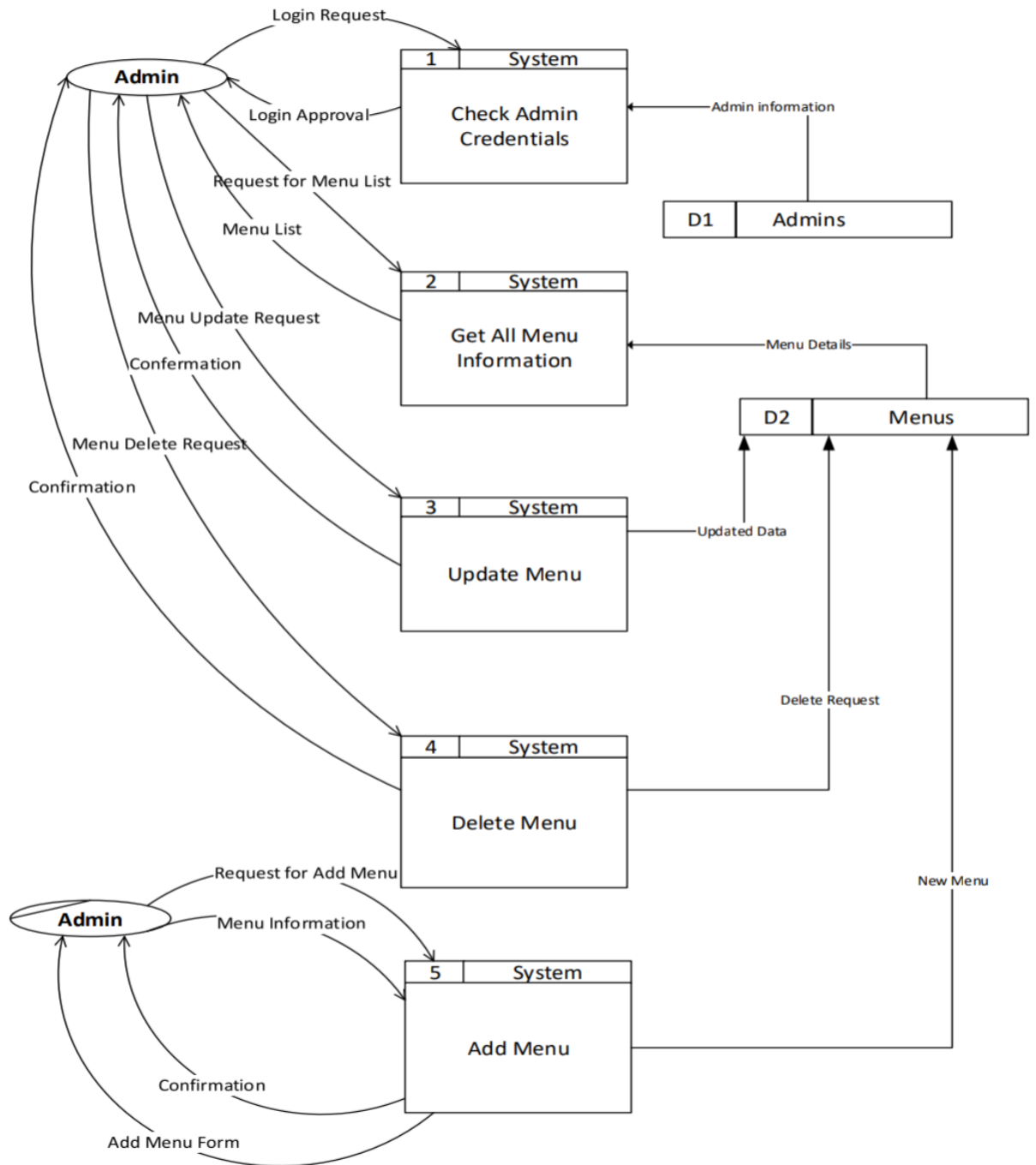


Figure 6: DFD for processes of Admin.

Figure 6 illustrates the processes of admin in for manipulating *Menus* table of the database. There are five processes for admin in this application. Process 1 depicts the login of admin to the system. When admin gives credentials and send login request to system, the system checks the credentials against the *Admins* table of the database and gives access. Process 2 illustrates the acquisition of available menu list for admin. In Process 3, the update process of menu has

been depicted. In Process 4, deletion of menu has been illustrated. In Process 5, adding process of new menu is shown.

4.3. Data Modelling

4.3.1. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) is a visual presentation which “facilitates database design by allowing specification of an enterprise schema that represents the overall logical structure of a database” (Korth, Sudarshan and Silberschatz, 2010). The following ERD illustrates the database design for restaurant application.

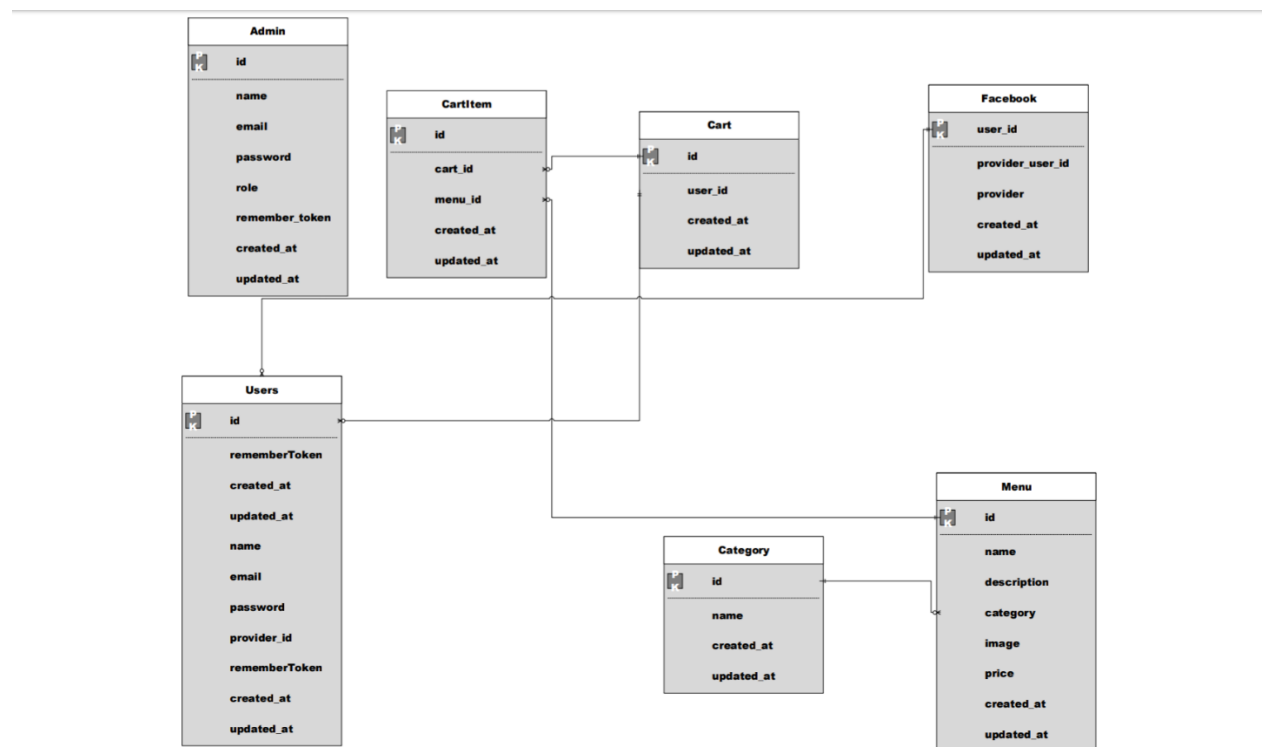


Figure 7: Entity relationship diagram of the system.

The crow foots shows the links between entities. A line with crowfoot in one side and normal line in another side means many-to one or one-to- many relationship. The ERD shows that, there is not any relationship between admin table and other tables in the database. *User* table

has relationship with *Cart* tables. The *Menu* table has a relationship with *Cart Item* and *Category* tables. The *Cart* table has a relationship with *Cart Item* and *User* tables.

4.4. User Interface Design

As specified earlier, the user interface of the system should be responsive. So, the interface can adopt any screen size. As the system can be accessed by devices with large screen (e.g., desktop computer, smart television browser) as well as devices with small screen (e.g., smart phone, tablet), every page of the application is designed for two types of screen differently.

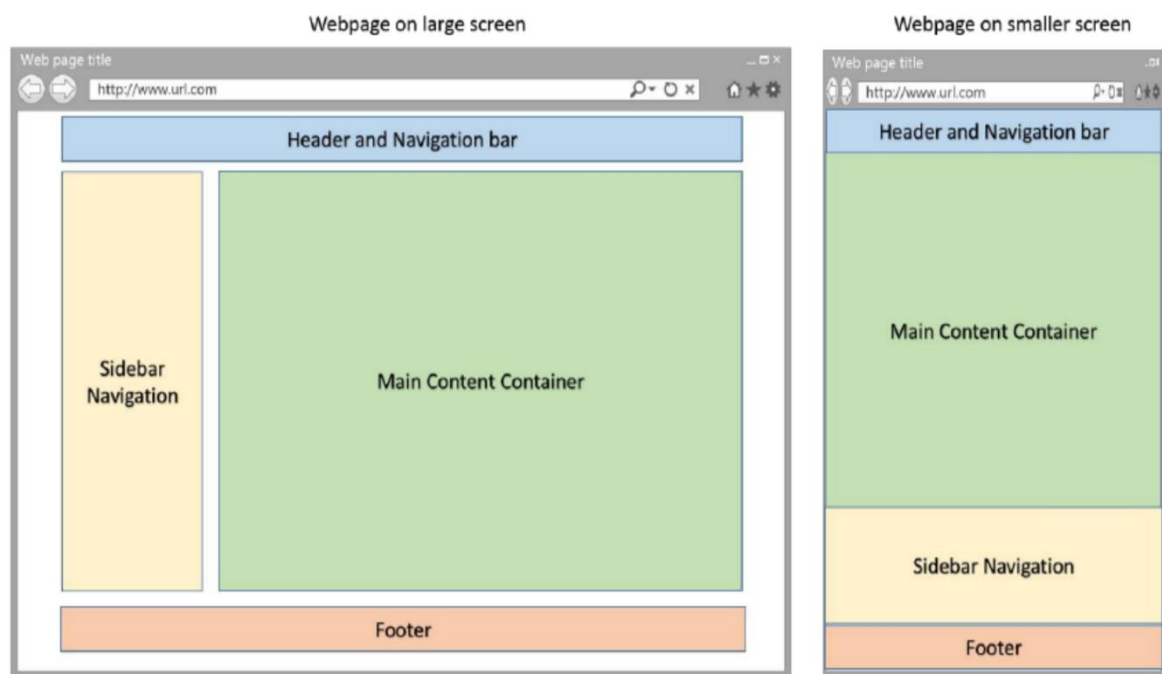


Figure 8: Difference of screen layouts for different size of screens.

4.4.1. Homepage

The homepage is the index page of the application. This is the page that is shown to visitor when she/he accessed the website through URL. *Figure 9* illustrates the layout of the homepage in large screen.

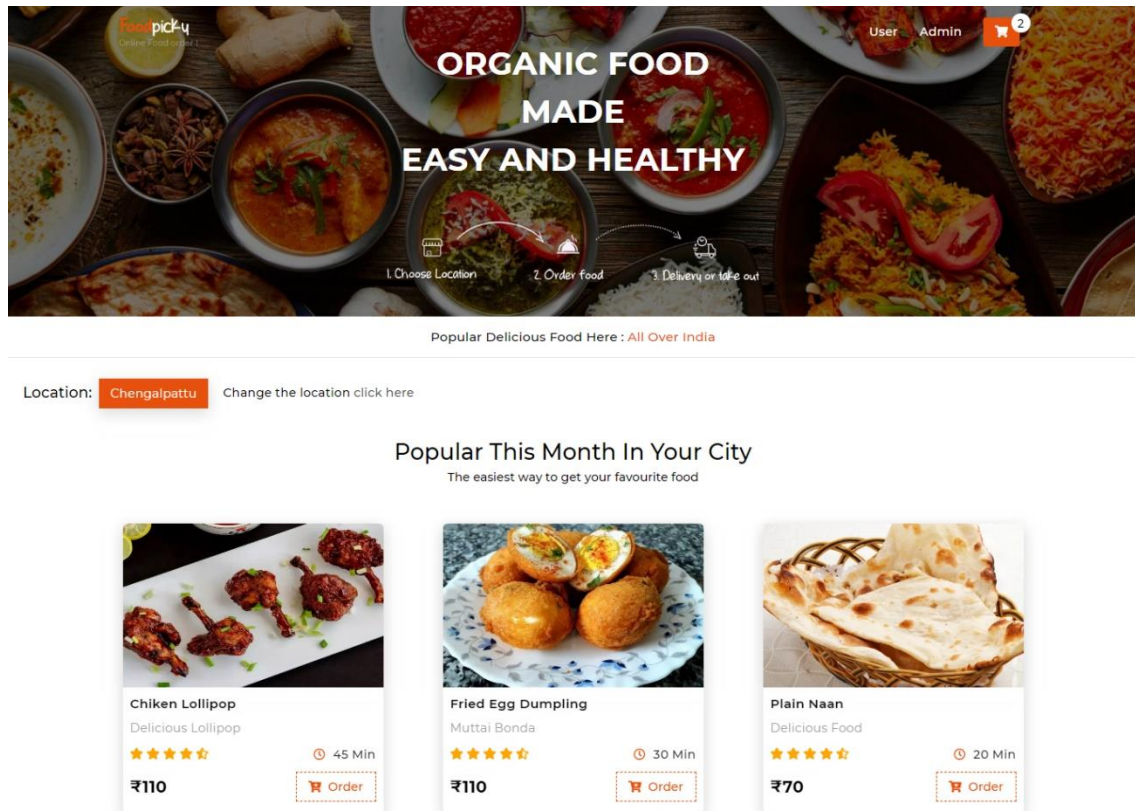


Figure 9: Homepage.

4.4.2. User Details

❖ User Login

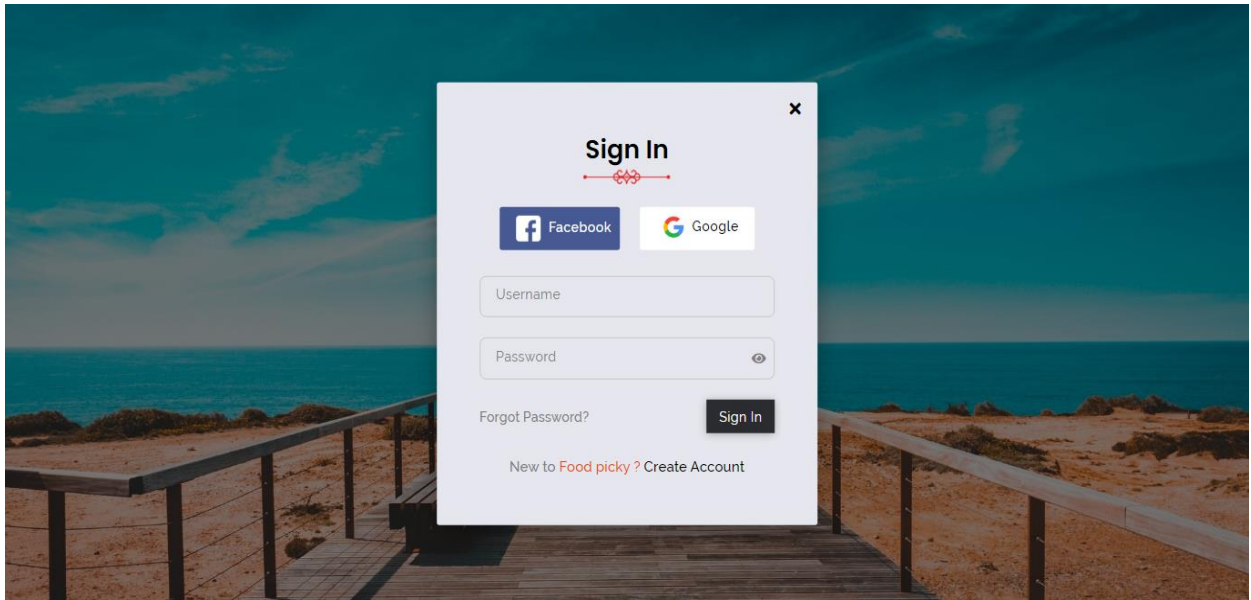
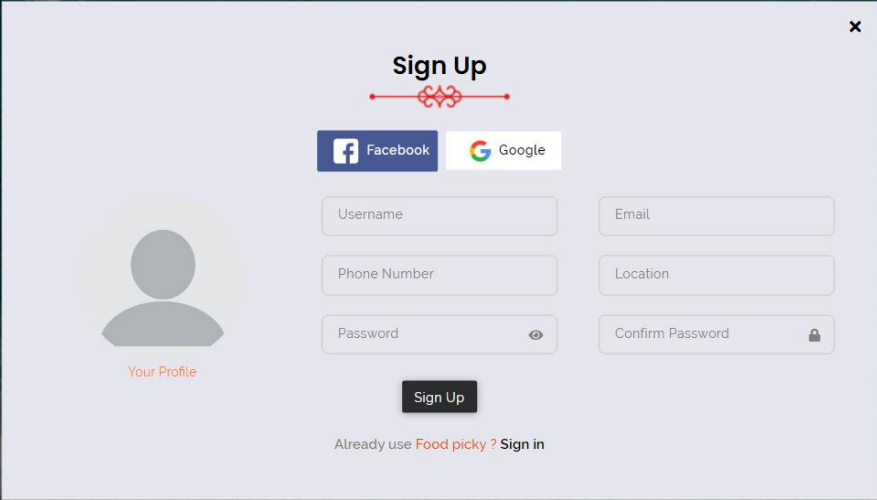


Figure 10: User Login.

❖ User Register



The image shows a user registration form titled "Sign Up" with a close button (X) in the top right corner. The form is set against a background of a dark, abstract image. On the left side of the form, there is a placeholder for a profile picture, a gray circle, with the text "Your Profile" below it. To the right of the profile picture, there are two buttons for social media login: "Facebook" and "Google". Below these buttons, there are six input fields arranged in two columns. The left column contains fields for "Username", "Phone Number", and "Password" (with an eye icon for toggling visibility). The right column contains fields for "Email", "Location", and "Confirm Password" (with a lock icon). Below the input fields, there is a "Sign Up" button. At the bottom of the form, there is a link that says "Already use Food picky ? Sign in".

Figure 11: User Register.

❖ User Panel

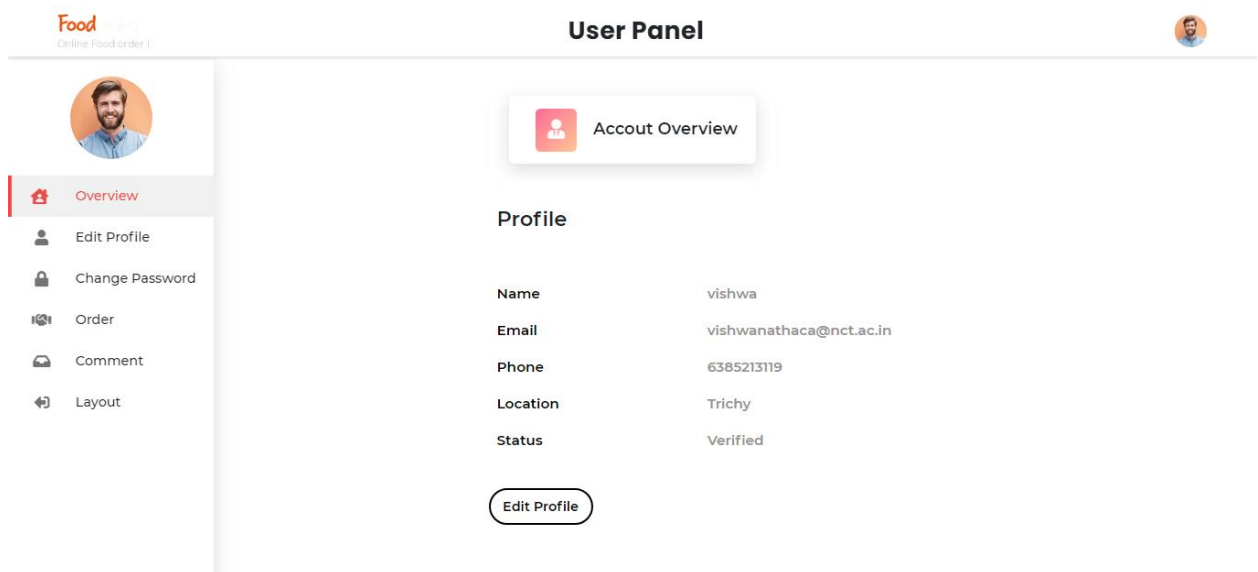



Figure 12: User Panel.


❖ Edit Profile

Food


Online Food order

User Panel







Overview




Edit Profile




Change Password




Order




Comment



Layout



Update Profile



Your Profile

Username

vishwa

Email

vishwanathaca@nct.ac.in

Phone Number

6385213119

Location

Trichy


Update

Figure 13: Edit Profile.

❖ User Change Password

Food ORDER

Online food order (1)



Overview

Edit Profile


Change Password

Order

Comment

Layout

User Panel

 Change Password

Old Password

New Password

Confirm Password



Change

Figure 14: User Change Password.

❖ User Order Details

Thank You! for coming
Your Order Placed Successfully!

Your Details	
Order No	133003
Name	vishwanath
Email	vishwanathaca@nct.ac.in
Phone No	6385213119
City	Chengalpattu
Area	Karumandapam
Landmark	Ambika Bakery
Payment Method	Cash On Delivery
Order Date	2021-03-20 23:46:20
Delivery	Free

Order Details			
#	Food	Name	Price
1		chiken lollipop	₹110
2		fried egg dumpling	₹110
Grand Total			₹220

LagoutsaveOrder Cancel

Figure 15: User Order Details.

4.4.3. Admin Details

❖ Admin Home Page

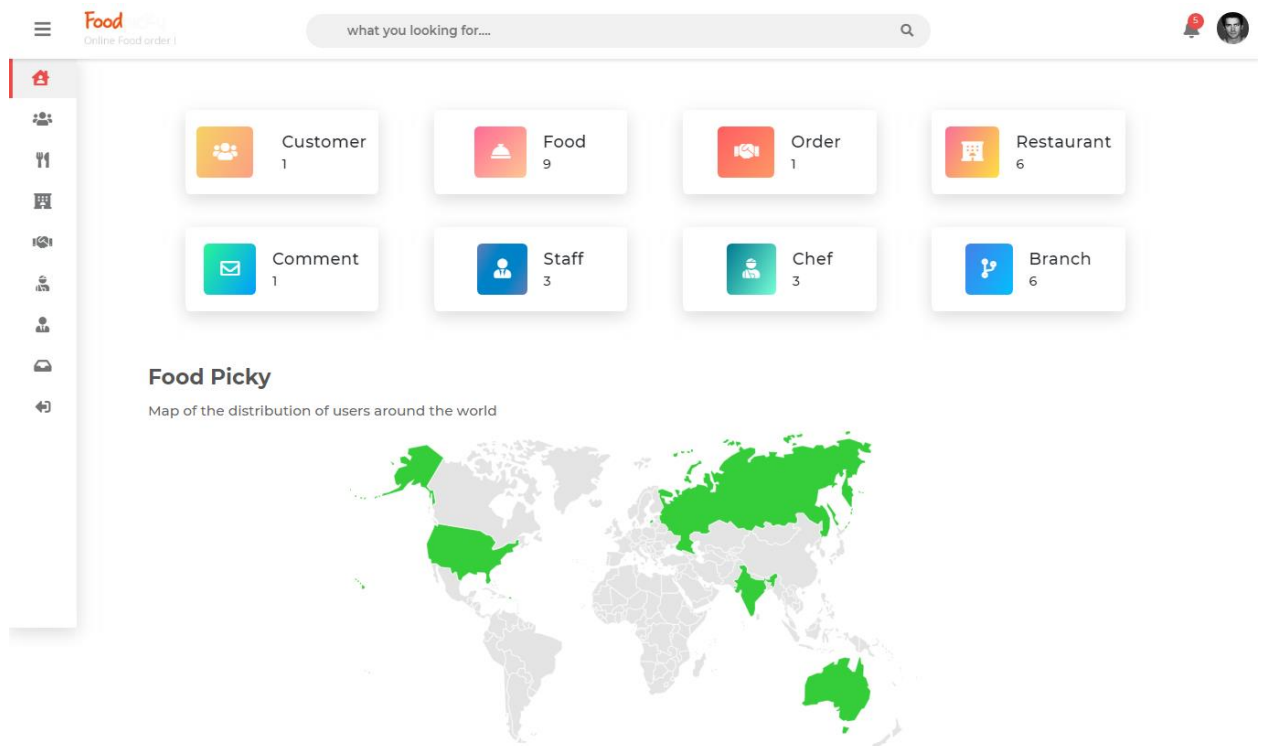
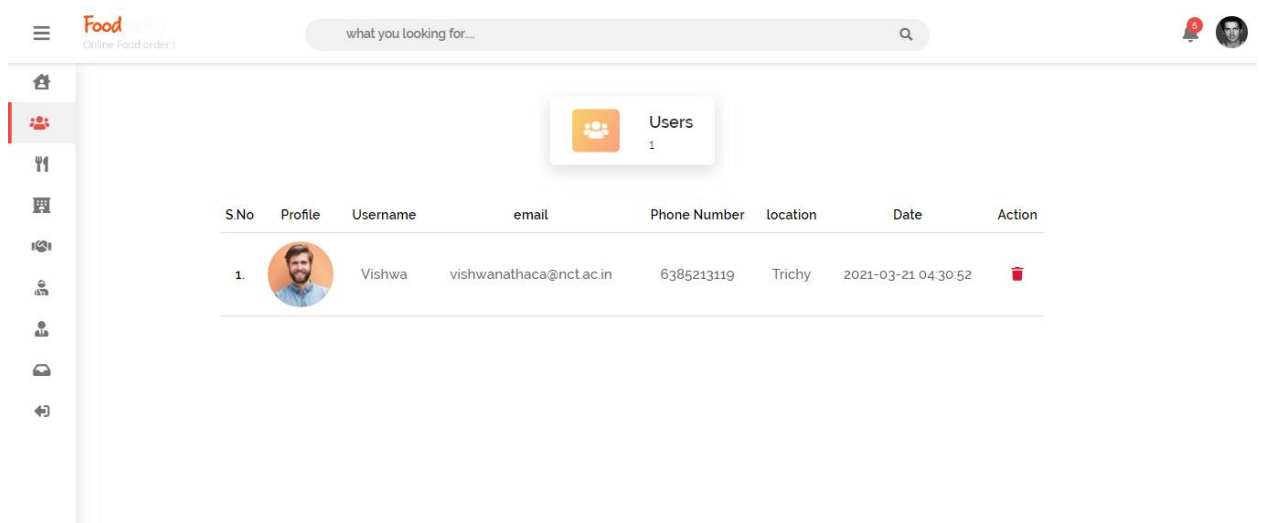


Figure 16: Admin Home Page.

❖ User Register Display





S.No	Profile	Username	email	Phone Number	location	Date	Action
1.		Vishwa	vishwanathaca@nct.ac.in	6385213119	Trichy	2021-03-21 04:30:52	

Figure 17: User Register Display.

❖ User Comment Display

The screenshot displays a web application interface for 'Food City Online Food order'. The top navigation bar features a search bar with the placeholder text 'what you looking for...'. A sidebar on the left contains various icons for navigation. The main content area shows a 'Comments' card with a speech bubble icon and the text 'Comments 1'. Below this, a table lists the comments:

S.No	Username	email	Phone Number	Message	Date	Action
1.	Vishwanath	vishwanathaca@nct.ac.in	6385213119	Nice	2021-03-21 08:10:43	

Below the table, there is a red button labeled 'All clear'.

Figure 17: User Comment Display.

❖ User Order Detail Display

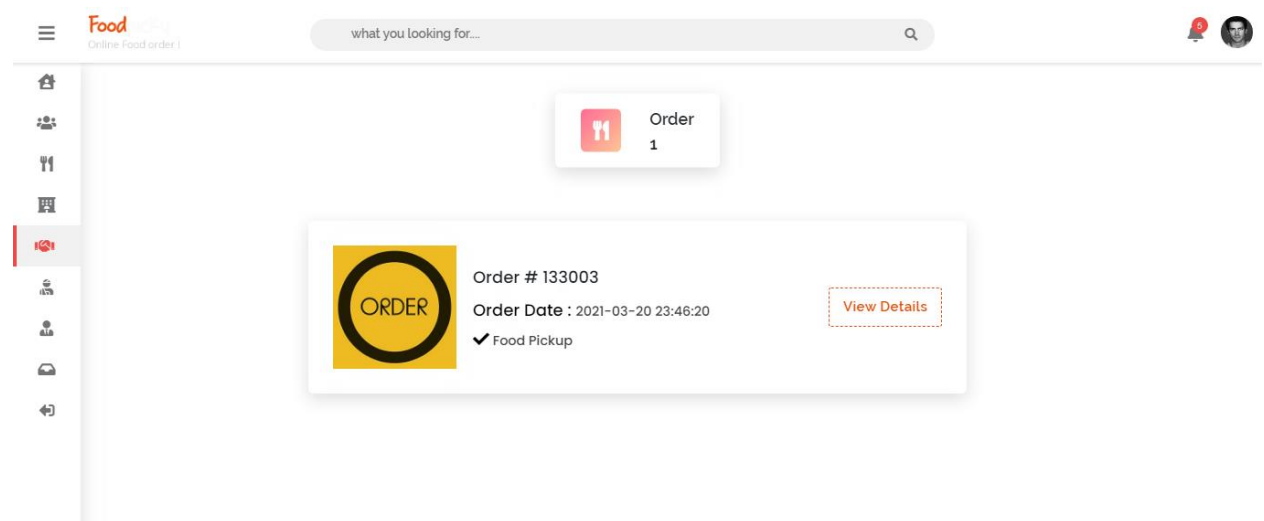








Figure 18: User Order Detail Display.

❖ Chef Detail Display

The screenshot shows a web application for 'Food Order' with the tagline 'Online Food order'. The interface includes a top navigation bar with a search bar containing the text 'what you looking for...' and a user profile icon. A left sidebar contains icons for home, users, menu, orders, and a red-highlighted 'chefs' icon. The main content area features a 'Chef 3' summary card with a chef icon. Below this is a table listing three chefs with columns for S.No, Profile, Name, Email, Phone Number, Location, Date, and Action. Each row includes a profile picture, name, contact details, location, date, and a delete icon. An 'All clear' button is positioned at the bottom of the table.

S.No	Profile	Name	Email	Phone Number	Location	Date	Action
1.		joe	joe@gmail.com	6385213119	Madurai	2021-03-17 14:20:15	
2.		Iris	iris@gmail.com	6385213154	Chennai	2021-03-17 14:22:35	
3.		Barry	barry@gmail.com	6836723145	Trichy	2021-03-17 14:24:42	

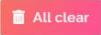
 All clear

Figure 19: Chef Detail Display.

CHAPTER 5

IMPLEMENTATION

This chapter provides an overview of the implementation process. In first, the implementation of the frontend part is described. After that, different backend functionalities have been discussed and evaluated.

5.1. Views

The user interface of the application is responsive. So, the interface can occupy the full screen in any kind of device. For making the user interface of the application mobile friendly, Bootstrap 4 is used. Laravel uses blade templating engine for making views and presentation layer of applications. Blade templating engine is used to develop and present the views. There is a main frontend view is developed, which has been extended for different views of different pages. The following code snippet shows the main view of the application.

```
<section class="food">
  <div class="max-width">
    <div id="message"></div>
    <div class="food-grid">
      <?php foreach ($result as $result) : ?>
        <div class="food-box-1">
          
          <p class="f-para_1"><?php echo $result['food Name']; ?></p>
          <p class="f-para_2"><?php echo $result['food Title']; ?></p>
          <div class="food-flex">
            <div class="star">
              <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="fas fa-star-half-alt">
            <p><i class="far fa-clock"></i><?php echo $result['foodTime']; ?></p>
          </div>
        </div>
      </?>
    </div>
  </div>
</section>
```



```

        </div>
        <div class="food-link">
            <p><?php echo $result['foodPrice']; ?></p>
            <form action="foodAction.php" method="POST">
                <!-- <a href="orderDisplay.php?order=<?php echo $result['id'];
?>">Order</a> -->
                <input type="hidden" class="fid" value="<?=$result['id'] ?>">
                <input type="hidden" class="fimage" name="foodImage" value="<?=$result['foodImage'] ?>">
                <input type="hidden" class="fname" name="foodName" value="<?=$result['foodName'] ?>">
                <input type="hidden" class="ftitle" name="foodTitle" value="<?=$result['foodTitle'] ?>">
                <input type="hidden" class="ftime" name="foodTime" value="<?=$result['foodTime'] ?>">
                <input type="hidden" class="fprice" name="foodPrice" value="<?=$result['foodPrice'] ?>">
                <button type="submit" class="f-btn" name='submit' style="cursor: pointer;" onclick="return confirm ('Are you sure want to Add your food in Card Item');"><i class="fas fa-cart-plus"></i>&nbsp; &nbsp; Order</button>
            </form>
        </div>
    </div>
<?php endforeach; ?>
</div>
</div>
</section>

```

Snippet 1: Main View

We can see that the main view extends and includes three other views which are header, navbar and footer. The header file is the HTML header files, which contains metatags of page. The navbar blade contains the navigation bar of the application. The footer blade contains the footer of the application. The footer blade also contains some JavaScript files needed for the application. For example, it contains the JavaScript file for connecting to Facebook API.

The following code snippet shows the header blade. The header blade contains the metatags and CSS links.

```
<?php session_start(); ?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Food</title>
    <link rel="icon" href="assets/img/icon.png">
    <link rel="stylesheet" href="assets/css/mainPage.css?<?php echo time(); ?>">
    <link rel="stylesheet" href="assets/font/fontawesome/css/all.css">
    <script src="assets/js/plugin/jquery.min.js"></script>
    <script src="assets/plugin/chosen.jquery.js"></script>
    <link rel="stylesheet" href="assets/plugin/chosen.min.css">
    <script src="assets/js/plugin/sweetalert.min.js"></script>

</head>

<body>

    <!-- ERROR MSG -->
    <?php
    if (isset($_SESSION['message'])) {
    ?>
        <script>
            swal({
                title: "<?php echo $_SESSION['message']; ?>",
                icon: "<?php echo $_SESSION['status_code']; ?>",
                button: "Ok",
            });
        </script>
    }
```

```

    </script>
<?php
    unset($_SESSION['message']);
}
?>

```

Snippet 2: Food

Food Cart

```

<?php foreach ($result as $result) : ?>
    <tr>
        <td>
            <?php echo $orderNo += 1; ?>
        </td>
        <td>
            
        </td>
        <td>
            <?php echo $result['foodName']; ?>
        </td>
        <td>
            <?php echo $result['foodTitle']; ?>
        </td>
        <td>
            <?php echo $result['foodTime']; ?>
        </td>
        <td>
            <?php echo $result['foodPrice']; ?>

```

```
</td>
<td>
    <a href="foodAction.php?id=<?php echo $result['id']; ?>" onclick="return
confirm('Are you sure want to clear your food?');"><i class="fas fa-trash"></i></a>
</td>
</tr>
<?php $grandTotal += 110; ?>
```

Snippet 3: Food Cart

Add Food Form

The screenshot shows a web application interface for adding food. At the top, there is a navigation bar with a search bar labeled "what you looking for..." and a user profile icon. On the left, a sidebar contains icons for home, users, food (highlighted), orders, and reviews. The main content area is titled "Food Add" and features a "Food image" placeholder with a fork and knife icon. To the right of the image placeholder are four input fields: "Food Name", "Food Description", "Food Time", and "Food Price". Below these fields is a blue "Add" button.

Figure 23: Add food form page.

Add food form code

```
<form action="foodInsert.php" method="POST" enctype="multipart/form-data">
  <div class="grid">
    <div class="box-1">
      <div class="form-groupp">
        <br>
        <label for="">Food image</label>
        <input type="file" name="profileImage" onchange="displayImg(this)"
id="profileImage" class="form-control" style="display: none;">
      </div>
    </div>
  </div>
```

```

        <input type="text" class="form_input" name="foodName" placeholder=" "
required>

        <label for="" class="form_label">Food Name</label>

    </div>

    <div class="form_div">

        <input type="text" class="form_input" name="foodTitle" placeholder=" "
required>

        <label for="" class="form_label">Food Description</label>

    </div>

    <div class="form_div">

        <input type="text" class="form_input" name="foodTime" placeholder=" "
required>

        <label for="" class="form_label">Food Time</label>

    </div>

    <div class="form_div">

        <input type="text" class="form_input" name="foodPrice" placeholder=" "
required>

        <label for="" class="form_label">Food Price</label>

    </div>

</div>

<div class="form_link">

    <button type="submit" class="form_button" style="background-image: linear-
gradient(120deg, #4facfe 0%, #00f2fe 100%);">Add</button>

</div>

</form>

```

Snippet 4: Add Food Form

CHAPTER 6

TESTING

This chapter elaborates the test process of the application and shows the result of testing.

6.1. Black Box Testing

Testing can be divided into two types broadly: functional testing and structural testing. Structural testing, also known as white-box testing, involves examining the internal implementation. It tests the design used by the implementation to verify its correctness. In contrast, Functional testing, sometimes referred to as black box testing, is testing on the functionality of the system based on the specified requirement. The test itself has little knowledge about the testing target's internal structure. The following table shows the test cases for black box testing.

6.1.1. Test 1

When a logged in user adds a menu-to-menu list, the page should be redirected to the cart page with correct subtotal and total.

Your Shopping Cart


S.No	Food	Name	Title	Time	Price	Action
1		chiken lollipop	Delicious lollipop	45 min	₹110	
Total				₹110		







Figure 24: Showing total for one menu.

6.1.2. Test 2

Whenever multiple menus added, the changes should be shown in the total price.

Your Shopping Cart

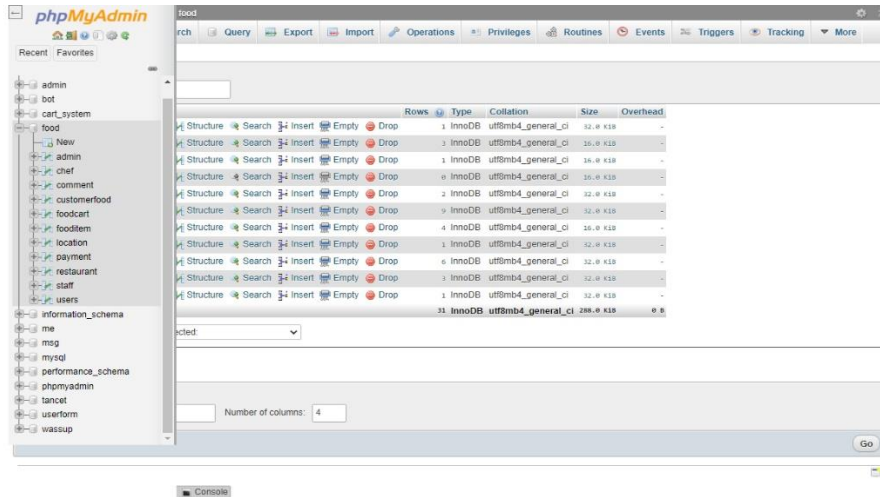
S.No	Food	Name	Title	Time	Price	Action
1		chiken lolliipop	Delicious lolliipop	45 min	₹110	
2		fried egg dumpling	Muttai Bonda	30 min	₹110	
Total				₹220		

 Continue Shopping  All clear  Checkout

Figure 25: Total of multiple menus.

6.1.3. Test 3

The new menu is shown in the menu table in database.



The screenshot shows the phpMyAdmin interface for the 'food' database. The 'menu' table is selected, and its structure is displayed. The table has 11 columns: 'id', 'name', 'description', 'price', 'category', 'status', 'created_at', 'updated_at', 'deleted_at', 'created_by', and 'updated_by'. The table is using the InnoDB engine and the utf8mb4_general_ci collation. The table size is 32.0 Kib, and the overhead is 0.8 Kib. The table contains 11 rows of data.

id	name	description	price	category	status	created_at	updated_at	deleted_at	created_by	updated_by
1	admin									
2	bot									
3	cart_system									
4	New									
5	admin									
6	chef									
7	comment									
8	customerfood									
9	foodcart									
10	fooditem									
11	location									
12	payment									
13	restaurant									
14	staff									
15	users									
16	information_schema									
17	me									
18	msg									
19	mysql									
20	performance_schema									
21	phpmyadmin									
22	tancet									
23	userform									
24	wassup									

Figure 20: New menu shown in database.

6.1.4. Test 4

The image of the new menu should be uploaded to the images folder directory.

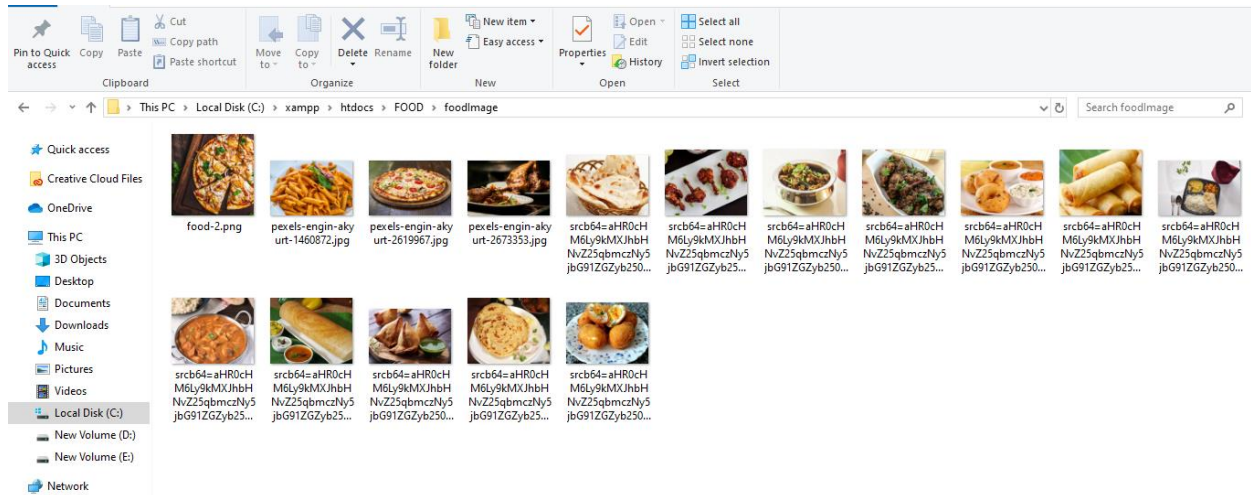


Figure 21: The image of the new menu uploaded to the folder.

6.1.5. Test 5

The admin navigated to the index of menu page and pressed edit button.

The screenshot displays the 'Menu list page' of a web application. The interface includes a sidebar with navigation icons, a top header with a search bar and user profile, and a main content area. A modal titled 'Restaurant 6' is open at the top right. Below it, a table lists 6 restaurants with their details and actions.

S.No	Restaurant	Name	Food	Location	Offer	Action
1.		kannapa	veg and Non-veg	Trichy	10% offer	
2.		Rayal	Dosa, chicken	Karumandapam	10% offer	
3.		New mass	veg and Non-veg	Trichy	10% offer	
4.		Bannana	veg and Non-veg	Trichy	8% offer	
5.		Meenachi	veg and Non-veg	veralimalai	15% offer	
6.		Rayal	Dosa, chicken	Karumandapam	15% offer	

At the bottom of the table, there is an 'All clear' button.

Figure 22: Menu list page.

6.2. User Acceptance Testing

User acceptance testing of the application is used to determine if the application meets the requirements of focused people. For user acceptance testing, random selection process is used. A questionnaire is made to obtain user evaluation of the system. The questionnaire consists of five question, which are set to gain user perspective on the ease of use, user friendliness, and responsiveness of UI and performance of the application. The answers have been taken anonymously from 20 random persons. The questionnaire can be found in *Appendix B*.

6.2.1. Ease of use

The first question of the questionnaire is for addressing the ease of use of the system. *Figure 35* illustrates the result of this question.

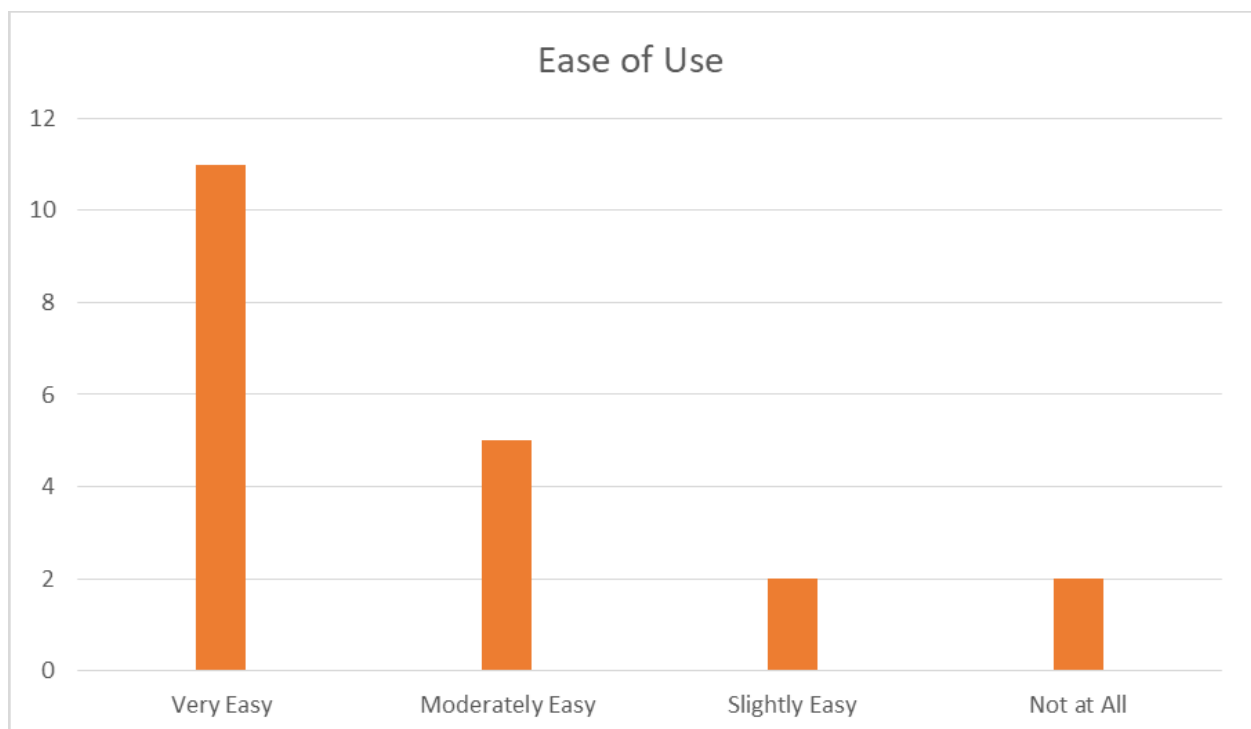


Figure 26: Ease of use of the application.

6.2.2. User Friendly

The pie chart in *Figure 36* illustrates the percentage of views of respondents on user friendliness of the application. Half of all respondents agreed that the system is moderately user friendly and 35% of them found the system very user friendly. So, it can be said that the website is user friendly. But there is a room for improvement as 15% of respondents do not find it so much user friendly.

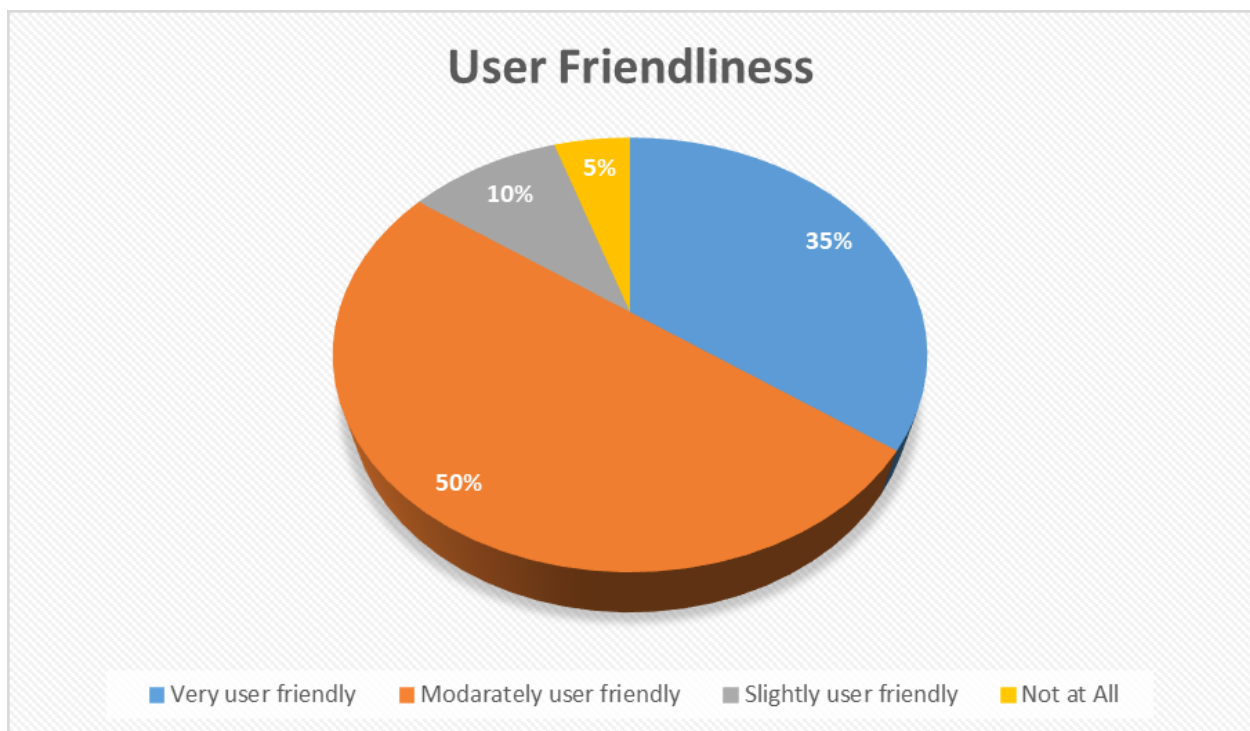


Figure 27: Percentage of views on user friendliness.

6.2.3. Responsiveness of Interface

The responsiveness of UI means that the website UI is optimized properly for different size of screens. The following donut pie chart shows that all respondent found the website responsive. But there are many issues to improve the responsiveness of the UI as 45% of respondent faced problem with their device.

RESPONSIVNESS OF UI

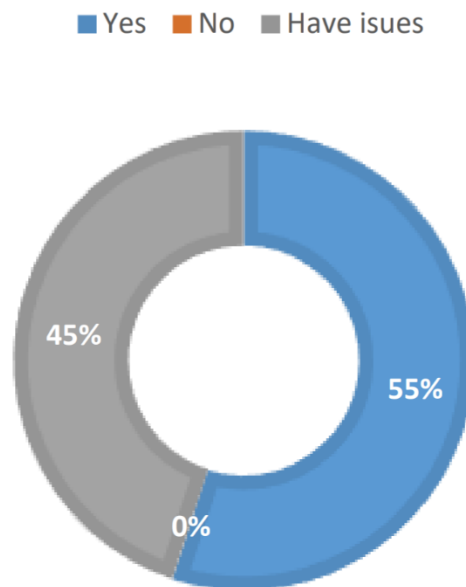


Figure 27: Percentage of views on responsiveness.

6.2.4. Speed & Performance

The pie chart in *Figure 39* illustrates the percentage of satisfaction with speed and performance of the system.

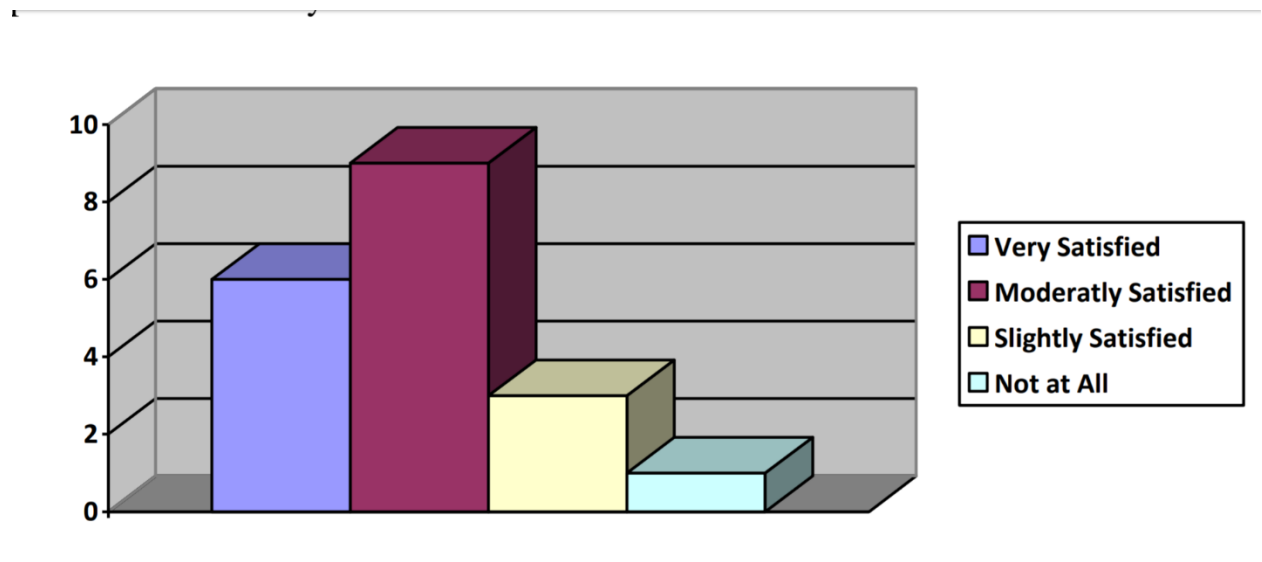


Figure 28: Satisfaction about speed and performance.

According to pie chart, 47% of respondent is very satisfied with the performance and speed and 32% of respondent is moderately satisfied with the system.

CHAPTER 7

RESULTS & EVALUATIONS

This chapter provides an overview of the project and its outcomes. The outcomes of the project also are analyzed and evaluated.

7.1. Security of the System

Security is one of the most important things for a web application, especially if the system contains transaction and user personal data. As it is an online food ordering system, transactions will be made through the app and the application also saves the different kind of personal information. In regards of ensuring the security of the application different techniques have been used. For ensuring the security of the password, default hashing function of Laravel has been used. Laravel handles all the passwords by encrypting and decrypting automatically, which has been hashed based on the secret key of the application. One the biggest problem for web applications is Cross-site request forgery (CSRF) attack. CSRF is a type of malicious exploit whereby unauthorized commands are performed on behalf of an authenticated user when s/he clicked on malicious link. Laravel automatically generates a CSRF "token" for each active user session managed by the application. This token is used to verify that the authenticated user is the one actually making the requests to the application. Anytime a HTML form in the application is defined, a CSRF hidden token field is included in the form so that the default CSRF protection middleware of Laravel can validate the request.

7.2. Responsive User Interface

The responsiveness of the application in any kind of screen was a big requirement for the system. In regards of achieving this objective, Bootstrap 4 front end framework has been used to develop the user interface of the system. This enables the UI of the application to glue any kind of screen whether it be smart phone, tablet PC, laptop or desktop PC. It also responsive for the proximity of the device. The application interface matches the size of the device automatically whether it is kept horizontally or vertically. Bootstrap 4 also reorganize the elements of the page based on the screen size.

CHAPTER 8

CONCLUSION AND RECOMMENDATIONS

This chapter concludes the report of this project. This chapter starts with discussing the achievements of this project. Following that, it describes the limitations in the system. It then proposes and recommends some features to be added to the system. Finally, the chapter ends by concluding remarks.

8.1. Achievement of the Project

The project has gone through a series of activities to develop a complex solution for the online food ordering system. After analysis of the project's goal and research direction, a set of objectives were established, as specified in Chapter 1.2. All the activities done during the project were attempts to realise these objectives. At the end of the project, the developed prototype software has fulfilled these objectives by the following means:

- ❖ Objective #1 was satisfied by reviewing the past works for automating the restaurant food ordering process. Along with this, the web application development technologies are briefly discussed.

The project was time-consuming. It has been tried to implement as many features as possible within the very limited timeframe. It has successfully satisfied the Functional Requirements. Some Non-functional Requirements of the system is not implemented. These requirements have top priority and reflect the most needed features. Some requirements are not implemented due to time constraints. However, their absence would not result in major operational issues as they are the lower priority features. These features could be implemented in the future. 71

8.2. Limitations of the System

There are also some limitations of the system. The shopping cart of the system has basic functionalities and does not support advanced cart modification features. Along with this, validation functionalities and almost all functionalities of the application are handled with server-side programming. It makes extra load on the server, especially when the application

gets lots of viewers. This limitation can be minimized by validating data using client-side language like JavaScript or HTML 5. Along with this, the order model has been developed. But the controllers and functions for pushing data into order table is not written. So, the placed orders cannot be viewed.

8.3. Future Recommendations

In addition to the unfinished requirements, there are other possibilities of further improving the project. The respondent of user acceptance testing also suggest some improvement ideas. The improvements may include:

- ❖ Overcoming the limitations mentioned in the *Section 7.3*.
- ❖ Secured payment system with various payment methods.
- ❖ Presenting graphical floor plan for table management and reservation.
- ❖ Converting the system to progressive web application.

8.4. Personal Reflection

I have gained some experiences and developed skills through the project. I also became aware of some aspects which needs improvements.

8.4.1. Acquiring Research Skills

The literature review chapter required reviewing lots of journal articles and research papers. By reading and reviewing journal articles, I have gained some knowledge about how to write a journal paper in a structured way and how a research is conducted. It also improved my presentation skills. I also become aware of my lacking on researching which should be improved to be a better researcher. 72

8.4.2. Acquiring Programming Skills

To develop the application, I have to learn Laravel and Bootstrap. Learning Laravel made my concept about object oriented programming clearer. I have also gained some experiences on problem solving skills, as sometimes I had to brainstorm, search and think for

two or three days to solve a piece of code that was not working. Sometimes it comes out a silly problem. Along with this, I was not good in developing user interface.

8.5. Concluding Remarks

This project is for developing a web application for restaurant. For developing the application, a systematic approach has been taken into account. Extreme Programming method of Agile development has been applied to develop the system. Along with this, SSADM has been used for modelling processes and data. The application has been developed using Laravel 5.4 PHP framework.

The project successfully implemented a working complex prototype of an online food ordering system with Facebook integrations. The implemented prototype software has been fully tested to demonstrate the quality and performance of the system. This report also documented all the relevant research details and decision-makings processes. In summary, the project has satisfied its objectives and fulfilled its purpose. I hope, the application can meet most of the requirements of restaurant online ordering.

BIBLIOGRAPHY

Alagoz, S. M. and Hekimoglu, H. (2012) ‘A Study on Tam: Analysis of Customer Attitudes in Online Food Ordering System’, *Procedia - Social and Behavioral Sciences*, 62, pp. 1138–1143. doi: 10.1016/j.sbspro.2012.09.195.

Bean, M. (2015) *Laravel 5 Essentials*. Packt Publishing Ltd.

Beltis, A. J. (2016) *9 Advantages of an Online Food Ordering System*. Available at: <https://pos.toasttab.com/blog/online-food-ordering-system> (Accessed: 7 July 2017).

Ben-Zahia, M. A. and Jaluta, I. (2014) ‘Criteria for selecting software development models’, in. *2014 Global Summit on Computer Information Technology (GSCIT)*, pp. 1–6. doi: 10.1109/GSCIT.2014.6970099.

Bhaumik, S. (2015) *Bootstrap Essentials*. Birmingham, UK: Packt Publishing.

Bounnady, K. *et al.* (2016) ‘Comparison the processing speed between PHP and ASP.NET’, in. *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 1–5. doi: 10.1109/ECTICon.2016.7561484.

Cheong, S. N., Chiew, W. W. and Yap, W. J. (2010) ‘Design and development of Multi-touchable E-restaurant Management System’, in. *2010 International Conference on Science and Social Research (CSSR 2010)*, IEEE, pp. 680–685. doi: 10.1109/CSSR.2010.5773867.

Das, R. and Saikia, L. P. (2016) ‘Comparison of Procedural PHP with Codeigniter and Laravel Framework’, *International Journal of Current Trends in Engineering & Research*, 2(6).

Dawson, C. W. (2009) *Projects in Computing and Information Systems: A Student’s Guide*. Addison-Wesley.

Dennis, A., Wixom, B. H. and Roth, R. M. (2012) *Systems Analysis and Design*. 5th Edition. John Wiley & Sons.

Dhore, V. B. *et al.* (2014) 'Digital Table Booking and Food Ordering System Using Android Application', *International Journal of Emerging Engineering Research and Technology*, 2(7), pp. 76–81.

Dockins, K. (2016) *Design Patterns in PHP and Laravel*. Apress.

APPENDICES

10.1. Appendix A: Code Snippets

```
<?php session_start(); ?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

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

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Food</title>
```

```
    <link rel="icon" href="assets/img/icon.png">
```

```
    <link rel="stylesheet" href="assets/css/mainPage.css?<?php echo time(); ?>">
```

```
    <link rel="stylesheet" href="assets/font/fontawesome/css/all.css">
```

```
    <script src="assets/js/plugin/jquery.min.js"></script>
```

```
    <script src="assets/plugin/chosen.jquery.js"></script>
```

```
    <link rel="stylesheet" href="assets/plugin/chosen.min.css">
```

```
    <script src="assets/js/plugin/sweetalert.min.js"></script>
```

```
</head>
```

```
<body>
```

```
    <!-- ERROR MSG -->
```

```

<?php
if (isset($_SESSION['message'])) {
?>

<script>

    swal({

        title: "<?php echo $_SESSION['message']; ?>",

        icon: "<?php echo $_SESSION['status_code']; ?>",

        button: "Ok",

    });

</script>

<?php
    unset($_SESSION['message']);
}
?>

<!-- END OF ERROR MSG -->

<!-------

    NAVBAR

----->

<div class="scroll-up-btn">

    <i class="fas fa-caret-up"></i>

</div>

<nav class="navbar">

    <div class="max-width">

        <div class="logo">

            <a href="#"></a>

```



```

</div>

<ul class="menu">

    <li><a href="#" class="u-btn">User</a></li>

    <li><a href="#" class="a-btn">Admin</a></li>

    <li> <a class="nav-link active" href="foodCart.php"><i class="fas fa-shopping-cart"></i>
<span id="cart-item" class="badge badge-danger"></span></a></li>

</ul>

</div>

</nav>

<?php
include "database.php";

$sql2 = "SELECT * FROM users";

$query2 = $con->query($sql2);

$result2 = mysqli_fetch_all($query2, MYSQLI_ASSOC)

?>

<!-- user -->

<div class="user-dropdown">

    <div class="user-img">

        <?php foreach ($result2 as $result2) { ?>

        <?php } ?>

    </div>

    <div class="user-item">

```

```
<p style="color: #E8500E; font-size:16px; font-weight:500;"><?php echo  
@$_SESSION['username']; ?></p>
```

```
</div><br>
```

```
<div class="user-item">
```

```
<a href="userLogin.php">Login</a>
```

```
</div><br>
```

```
<div class="user-item">
```

```
<a href="userHomeLogin.php">User Panel</a>
```

```
</div><br>
```

```
<!-- <div class="user-item">
```

```
<a href="index.php" onclick="return confirm('Are you sure want to logout');">Logout</a>
```

```
</div> -->
```

```
</div>
```

```
<!-- admin -->
```

```
<?php
```

```
include "database.php";
```

```
$sql = "SELECT * FROM admin";
```

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

```
$result = mysqli_fetch_all($query, MYSQLI_ASSOC)
```

```
?>
```

```
<div class="a-dropdown">
```

```
<div class="a-img">
```

```
<?php
```

```
foreach ($result as $result) { ?>
```

```
        
```

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

```
    </div>
```

```
    <div class="a-item">
```

```
        <a href="adminLogin.php">Login</a>
```

```
    </div>
```

```
</div>
```

```
<section class="home">
```

```
    <div class="home-content">
```

```
        <h2>organic food <br> made <br> easy and healthy </h2>
```

```
        
```

```
    </div>
```

```
</section>
```

```
<div class="home-bottom">
```

```
    <p>Popular Delicious Food Here : <span>All Over India</span></p>
```

```
</div>
```

```
<div class="location">
```

```
    <h2>Location:<span class="l-color"><?php echo $_SESSION['location']; ?></span> </h2>
```

```
    <p>Change the location <a href="index.php">click here</a></p>
```

```
</div>
```

```
<!-- <nav class="f-nav">
```

```
<div class="max-width">

  <ul class="f-menu">

    <li><a href="#">Tiffins</a></li>

    <li><a href="#">Veg Appetizer</a></li>

    <li><a href="#">Non Veg Appetizer</a></li>

    <li><a href="#">Classic Meal Box</a></li>

    <li><a href="#">Rice and Breads</a></li>

    <li><a href="#">Thalapakatu Briyani</a></li>

  </ul>

</div>

</nav> -->
```

```
<div class="food-head">

  <h2>popular this month in your city</h2>

  <p>The easiest way to get your favourite food</p>

</div>
```

```
<?php
```

```
include "database.php";
```

```
$sql = "SELECT * FROM fooditem";
```

```
$query = mysqli_query($con, $sql);
```

```
$result = mysqli_fetch_all($query, MYSQLI_ASSOC);
```

?>

```
<section class="food">
```

```
  <div class="max-width">
```

```
    <div id="message"></div>
```

```
    <div class="food-grid">
```

```
      <?php foreach ($result as $result) : ?>
```

```
        <div class="food-box-1">
```

```
          
```

```
          <p class="f-para_1"><?php echo $result['foodName']; ?></p>
```

```
          <p class="f-para_2"><?php echo $result['foodTitle']; ?></p>
```

```
          <div class="food-flex">
```

```
            <div class="star">
```

```
              <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="fas fa-star-half-alt"></i>
```

```
            </div>
```

```
            <div>
```

```
              <p><i class="far fa-clock"></i><?php echo $result['foodTime']; ?></p>
```

```
            </div>
```

```
          </div>
```

```
        <div class="food-link">
```

```
          <p><?php echo $result['foodPrice']; ?></p>
```

```
          <form action="foodAction.php" method="POST">
```

```

        <!-- <a href="orderDisplay.php?order=?php echo $result['id']; ?>">Order</a> --
    >

        <input type="hidden" class="fid" value="<?= $result['id'] ?>">

        <input type="hidden" class="fimage" name="foodImage" value="<?=
$result['foodImage'] ?>">

        <input type="hidden" class="fname" name="foodName" value="<?=
$result['foodName'] ?>">

        <input type="hidden" class="ftitle" name="foodTitle" value="<?=
$result['foodTitle'] ?>">

        <input type="hidden" class="ftime" name="foodTime" value="<?=
$result['foodTime'] ?>">

        <input type="hidden" class="fprice" name="foodPrice" value="<?=
$result['foodPrice'] ?>">

        <button type="submit" class="f-btn" name='submit' style="cursor: pointer;"
onclick="return confirm('Are you sure want to Add your food in Card Item');"><i class="fas fa-cart-
plus"></i>&nbsp;&nbsp; Order</button>

    </form>

</div>

</div>

<?php endforeach; ?>

</div>

</div>

</section>

<section class="restaurant">

    <div class="max-width">

        <h2>Easy 3 Step Order</h2>

```

```

<div class="r-grid">

  <div class="r-box_1">

    <h3>Choose a restaurant</h3>

    <p>We've got your covered menus from over 345 delivery resturants online.</p>

  </div>

  <div class="r-box_2">

    <h3>Choose a tasty dish</h3>

    <p>We've got your covered menus from over 345 delivery resturants online.</p>

  </div>

  <div class="r-box_1">

    <h3>Pick up or Delivery</h3>

    <p>Get your delivered! And enjoy your meal! pay online on public or delivery.</p>

  </div>

</div>

</div>

</section>

```

```

<section class="res-head">

  <div class="max-width">

    <h2>Featured restaurants</h2>

    <div class="line_1"></div>

    <div class="line_2"></div>

```

```
</div>
```

```
</section>
```

```
<?php
```

```
include "database.php";
```

```
$sql = "SELECT * FROM restaurant";
```

```
$query = mysqli_query($con, $sql);
```

```
$result = mysqli_fetch_all($query, MYSQLI_ASSOC);
```

```
?>
```

```
<section class="res">
```

```
<div class="max-width">
```

```
<div class="res-grid">
```

```
<?php foreach ($result as $result) : ?>
```

```
<div class="res-box-1">
```

```
<div class="res-flexx">
```

```
<div>
```

```

```

```
</div>
```

```
<div>
```

```
<p class="r-para_1"><?php echo $result['restaurantName']; ?></p>
```

```
<p class="r-para_2"><?php echo $result['foodItem']; ?></p>
```



```

<p class="r-para_2"><?php echo $result['restaurantLocation']; ?></p>

<div class="res-flex">

    <div class="star">

        <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="fas fa-star-half-alt"></i>

    </div>

    <div>

        <p><?php echo $result['offer']; ?></p>

    </div>

</div>

</div>

</div>

</div>

<?php endforeach; ?>

</div>

</div>

</section>

```

```

<section class="customerFood">

    <div class="max-width">

        <div class="c-head">

            <h2>Package</h2>

```

```

<div class="line_1"></div>

<div class="line_2"></div>

</div>

<div class="c-header">

    <h2>Lots food item available in <br> <span>f</span>ood <span>p</span>icky</h2>

</div>

<form action="customerFoodInsert.php" method="POST" class="c-form">

    <br><br><br><br>

    <select class="chosen" name="food" required style="width: 400px;">

        <option value=""> Choose your food...</option>

        <option value="" style="color:#E8500E; background:#fff; font-size:19px; font-weight:500; padding:10px 0px;">Breakfast & Tiffin Recipes</option><br>

        <option value="Dosai தோசை">Dosai தோசை</option>

        <option value="Ven Pongal வென் பொங்கல்">Ven Pongal வென் பொங்கல்</option>

        <option value="Idli இட்லி">Idli இட்லி</option>

        <option value="Idiyappam இடியப்பம்">Idiyappam இடியப்பம்</option>

        <option value="Aapam ஆபம்">Aapam ஆபம்</option>

        <option value="Rava Dosai ரவ தோசை">Rava Dosai ரவ தோசை</option>

        <option value="Rava Puttu ரவ புட்டு">Rava Puttu ரவ புட்டு</option>

        <option value="Idli Upma இட்லி உப்ம்">Idli Upma இட்லி உப்ம்</option>

        <option value="Kushboo Idli குஷ்பூ இட்லி">Kushboo Idli குஷ்பூ இட்லி</option>

        <option value="Idli Upma இட்லி உப்ம்">Idli Upma இட்லி உப்ம்</option>

```

<option value="" style="color:#E8500E; background:#fff; font-size:19px; font-weight:500; padding:10px 0px;">Rice Varieties</option>

<option value="White Rice வெள்ளை அரிசி (சடம்)">White Rice வெள்ளை அரிசி (சதாம்)</option>

<option value="Thayir Sadam தாயீர் சதாம்">Thayir Sadam தாயீர் சதாம்</option>

<option value="Takkali Sadam தக்காளி சதாம்">Takkali Sadam தக்காளி சதாம்</option>

<option value="Thengai Sadam தெங்கை சதாம்">Thengai Sadam தெங்கை சதாம்</option>

<option value="Sambar Sadam சம்பர் சதாம்">Sambar Sadam சம்பர் சதாம்</option>

<option value="Carrot Sadam கேரட் அரிசி">Carrot Sadam கேரட் அரிசி</option>

<option value="Elumicahi Sadam எலுமிச்சாய் சதாம்">Elumicahi Sadam எலுமிச்சாய் சதாம்

<option>

<option value="Karuvepillai Sadam கருவேபில்லை சதாம்">Karuvepillai Sadam கருவேபில்லை சதாம்</option>

<option value="Khuska Biryani குஸ்கா பிரியாணி">Khuska Biryani குஸ்கா பிரியாணி</option>

<option value="Vegetable Biryani காய்கறி பிரியாணி">Vegetable Biryani காய்கறி பிரியாணி</option>

<option value="" style="color:#E8500E; background:#fff; font-size:19px; font-weight:500; padding:10px 0px;">Curries and Side Dishes</option>

<option value="Sambar சாம்பார்">Sambar சாம்பார்</option>

<option value="Vada Curry வட கறி">Vada Curry வட கறி</option>

<option value="Vendakkai Pulikulambu வெண்டக்காய் புலிக்குலம்பு">Vendakkai Pulikulambu வெண்டக்காய் புலிக்குலம்பு</option>

<option value="Rasam ரசம்">Rasam ரசம்</option>

<option value="Thakkali Kurma தக்காளி குர்ம">Thakkali Kurma தக்காளி குர்ம</option>

<option value="Pattani Kurma பட்டனி குர்ம">Pattani Kurma பட்டனி குர்ம</option>

<option value="Meen Kuzhambu மீன் குழம்பு">Meen Kuzhambu மீன் குழம்பு</option>

<option value="Mutton Sukka மட்டன் சுக்க">Mutton Sukka மட்டன் சுக்க</option>

<option value="Potato Kurma பொட்டொ குர்ம">Potato Kurma பொட்டொ குர்ம</option>

<option value="Meen Varuval மீன் வருவல்">Meen Varuval மீன் வருவல்</option>

<option value="" style="color:#E8500E; background:#fff; font-size:19px; font-weight:500; padding:10px 0px;">Snack</option>

<option value="Sundal சுண்டல்">Sundal சுண்டல்</option>

<option value="Medhu Vadai மெது வடை">Medhu Vadai மெது வடை</option>

<option value="Bajji பஜ்ஜி">Bajji பஜ்ஜி</option>

<option value="Masala Vadai மசல வடை">Masala Vadai மசல வடை</option>

<option value="Thayir Vadai தயிர் வடை">Thayir Vadai தயிர் வடை</option>

<option value="Kaara Boondi கார பூண்டி">Kaara Boondi கார பூண்டி</option>

<option value="Murukku முருக்கு">Murukku முருக்கு</option>

<option value="Sambar Vada சம்பர் வட">Sambar Vada சம்பர் வட</option>

<option value="Thaen Mittai தேன் மிட்டை">Thaen Mittai தேன் மிட்டை</option>

<option value="Thattai தட்டை">Thattai தட்டை</option>

<option value="" style="color:#E8500E; background:#fff; font-size:19px; font-weight:500; padding:10px 0px;">Desserts and Sweets </option>

<option value="Banana Bonda பன்ன பொண்ட">Banana Bonda பன்ன பொண்ட</option>

<option value="Parppu Payasam பர்ப்பு பயசம்">Parppu Payasam பர்ப்பு பயசம்</option>

<option value="Vella Cheedai வெல்ல சீடை">Vella Cheedai வெல்ல சீடை</option>

<option value="Rava Ladoo ரவ லடூ">Rava Ladoo ரவ லடூ</option>

<option value="Paal Kozhukattai பால் கொழுகட்டை">Paal Kozhukattai பால் கொழுகட்டை</option>

<option value="Semiya Payasam செமிய பயசம்">Semiya Payasam செமிய பயசம்</option>

<option value="Kaju Katli கஜு கட்லி">Kaju Katli கஜு கட்லி</option>

<option value="Asoka Halwa அசொக ஹல்வ">Asoka Halwa அசொக ஹல்வ</option>

</select>

<select class="chosen" name="value" required style="width: 400px;">

<option value=""> Choose Your Food KG</option>

<option value="" style="color:#E8500E; background:#fff; font-size:19px; font-weight:500; padding:10px 0px;">Grams</option>

<option value="250g ₹50">250g ₹50</option>

<option value="500g ₹100">500g ₹100</option>

<option value="750g ₹150">750g ₹150</option>

<option value="" style="color:#E8500E; background:#fff; font-size:19px; font-weight:500; padding:10px 0px;">Kilograms</option>

<option value="1kg ₹150">1kg ₹150</option>

<option value="2kg ₹300">2kg ₹300</option>

<option value="3kg ₹450">3kg ₹450</option>

<option value="4kg ₹600">4kg ₹600</option>

<option value="5kg ₹1050">5kg ₹1050</option>

<option value="6kg ₹1200">6kg ₹1200</option>

<option value="7kg ₹1350">7kg ₹1350</option>

<option value="8kg ₹1500">8kg ₹1500</option>

<option value="9kg ₹1650">9kg ₹1650</option>

<option value="10kg ₹1800">10kg ₹1800</option>

<option value="11kg ₹1950">11kg ₹1950</option>

<option value="12kg ₹2100">12kg ₹2100</option>

<option value="13kg ₹2250">13kg ₹2250</option>

<option value="14kg ₹2300">14kg ₹2300</option>

<option value="15kg ₹2450">15kg ₹2450</option>

<option value="" style="color:#E8500E; background:#fff; font-size:19px; font-weight:500; padding:10px 0px;">Litre</option>

<option value="1L ₹60">1L ₹60</option>

<option value="2L ₹120">2L ₹120</option>

```

        <option value="3L ₹180">3L ₹180</option>
        <option value="4L ₹240">4L ₹240</option>
        <option value="5L ₹300">5L ₹300</option>
        <option value="6L ₹360">6L ₹360</option>
        <option value="7L ₹420">7L ₹420</option>
        <option value="8L ₹480">8L ₹480</option>
        <option value="9L ₹540">9L ₹540</option>
        <option value="10L ₹600">10L ₹600</option>
        <option value="11L ₹660">11L ₹660</option>
        <option value="12L ₹720">12L ₹720</option>
        <option value="13L ₹780">13L ₹780</option>
        <option value="14L ₹840">14L ₹840</option>
        <option value="15L ₹900">15L ₹900</option>
    </select>

    <button type="submit" name="submit" class="f-btn">Order</button>

</form>

</div>

<section class="c-food">
    <div class="max-width">

        </div>
    </section>

<?php include "footer.php"; ?>

<script type="text/javascript">

```

```

$(document).ready(function() {

    // Load total no.of items added in the cart and display in the navbar

    load_cart_item_number();

    function load_cart_item_number() {

        $.ajax({

            url: 'foodAction.php',

            method: 'get',

            data: {

                cartItem: "cart_item"

            },

            success: function(response) {

                $("#cart-item").html(response);

            }

        });

    }

    $(".u-btn").click(function() {

        $(".user-dropdown").toggleClass("open")

    })

    $(".a-btn").click(function() {

        $(".a-dropdown").toggleClass("open")

    })

});

</script>

```



```
<script>

    $(".chosen").chosen();

</script>

</body>

</html>
```

Snippet 5: Main Home Page

```
<?php
session_start();

include "database.php";

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>User</title>

    <link rel="shortcut icon" href="assets/img/icon.png" type="image/x-icon">

    <link rel="stylesheet" href="assets/css/userHome.css?<?php echo time(); ?>">

    <link rel="stylesheet" href="assets/font/fontawesome/css/all.css">

</head>

<body class="overlay-scrollbar">

    <div class="navbar">

        <!-- nav left -->

        <ul class="navbar-nav">

            <li class="nav-item">

                <a class="nav-link" style="margin:0px 21px;">

                    <i class="fas fa-bars" onclick="collapseSidebar()"></i>

                </a>

            </li>

            <li class="nav-item">
```

```

    </li>

</ul>

<!-- end nav left -->

<div class="navbar-search">

    <p>User Panel</p>

</div>

<!-- nav right -->

<ul class="navbar-nav nav-right">

    <?php

include "database.php";

$sql = "SELECT * FROM users";

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

$result = mysqli_fetch_all($query, MYSQLI_ASSOC)

?>

<li class="nav-item avt-wrapper">

    <div class="avt dropdown" style="max-width: 300px;">

        <?php

        foreach ($result as $result) { ?>

            <?php } ?>

            <ul id="user-menu" class="dropdown-menu">

```

```

<li class="dropdown-menu-item">

    <a class="dropdown-menu-link" href="adminProfile.php">

        <div>

            <i class="fas fa-user-tie" style="color: #202940;"></i>

        </div>

        <span>Profile</span>

    </a>

</li>

<li class="dropdown-menu-item">

    <a class="dropdown-menu-link" href="adminHome.php">

        <div>

            <i class="fas fa-sign-out-alt" style="color: #202940;"></i>

        </div>

        <span>Logout</span>

    </a>

</li>

</ul>

</div>

</li>

</ul>

<!-- end nav right -->

</div>

<!-- end navbar -->

<?php

```

```
include "database.php";
```

```
$sql1 = "SELECT * FROM users";
```

```
$query1 = $con->query($sql1);
```

```
$result1 = mysqli_fetch_all($query1, MYSQLI_ASSOC)
```

```
?>
```

```
<!-- sidebar -->
```

```
<div class="sidebar">
```

```
    <ul class="sidebar-nav">
```

```
        <li class="u-profile">
```

```
            <?php
```

```
                foreach ($result1 as $result1) { ?>
```

```
                    
```

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

```
        </li>
```

```
        <li class="sidebar-nav-item">
```

```
            <a href="#" class="sidebar-nav-link active">
```

```
                <div><i class="fas fa-house-user"></i></div>
```

```
                <span>Overview</span>
```

```
            </a>
```

```
        </li>
```

```
        <li class="sidebar-nav-item">
```

```
            <a href="user-edit.php" class="sidebar-nav-link ">
```

```
                <div><i class="fas fa-user"></i></div>
```

```
                <span>Edit Profile</span>
```

```

        </a>

    </li>

    <li class="sidebar-nav-item">

        <a href="userChangePass.php" class="sidebar-nav-link">

            <div><i class="fas fa-lock"></i></div>

            <span>Change Password</span>

        </a>

    </li>

    <li class="sidebar-nav-item">

        <a href="user-order.php" class="sidebar-nav-link">

            <div><i class="fas fa-handshake"></i></div>

            <span>Order</span>

        </a>

    </li>

    <li class="sidebar-nav-item">

        <a href="userComment.php" class="sidebar-nav-link">

            <div><i class="fas fa-inbox"></i></div>

            <span>Comment</span>

        </a>

    </li>

    <li class="sidebar-nav-item">

        <a href="mainPage.php" class="sidebar-nav-link" onclick="return confirm('Are you sure
want to logout');">

            <div style="transform:rotate(180deg);"><i class="fas fa-sign-out-alt"></i></div>

            <span>Layout</span>

        </a>

```


</div>

<?php

include "database.php";

\$sql2 = "SELECT * FROM users";

\$query2 = \$con->query(\$sql2);

\$result2 = mysqli_fetch_all(\$query2, MYSQLI_ASSOC)

?>

<section class="card">

<div class="max-width">

<div class="card-box-1">

<i class="fas fa-user-tie" style="background-image: linear-gradient(-45deg, #FFC796 0%, #FF6B95 100%);"></i>

<div>

<h3>Accout Overview</h3>

</div>

</div>

</div>

</section>

<section class="user-d">

```

<div class="max-width-1">

    <h3>Profile</h3>

    <div class="user-flex">

        <?php foreach ($result2 as $result2) { ?>

            <div class="user-box-1">

                <p>Name</p>

                <p>Email</p>

                <p>Phone</p>

                <p>Location</p>

                <p>Status</p>

            </div>

            <div class="user-box-2">

                <p><?php echo $result2['username']; ?></p>

                <p><?php echo $result2['email']; ?></p>

                <p><?php echo $result2['phoneNumber']; ?></p>

                <p><?php echo $result2['location']; ?></p>

                <p><?php echo $result2['status']; ?></p>

            </div>

            <?php } ?>

        </div><br><br>

        <a href="user-edit.php">Edit Profile</a>

    </div>

</section>

<script src="assets/js/script.js"></script>

```



```
<script src="assets/js/plugin/jquery-1.10.2.js"></script>  
</body>  
</html>
```

Snippet 6: User Home Page

```
<?php

$con = mysqli_connect("localhost","root","","food");

?>
```

Snippet 7: Database Connection

```
<div class="max-width">

    <div class="card-grid">

        <div class="card-box-1">

            <i class="fas fa-users" style="margin-right: 35px; background-image: linear-
gradient(120deg, #f6d365 0%, #fda085 100%);"></i>

            <div>

                <h3>Customer</h3>

                <?php

                    $query = "SELECT id FROM users ORDER BY id";

                    $query_run = mysqli_query($con, $query);

                    $row = mysqli_num_rows($query_run);

                    echo '<p>' . $row . '</p>';

                ?>

            </div>

        </div>

        <div class="card-box-1">

            <i class="fas fa-concierge-bell" style="background-image: linear-gradient(-45deg,
#FFC796 0%, #FF6B95 100%);"></i>

            <div>

                <h3>Food</h3>
```

```

        <?php
        $query = "SELECT id FROM fooditem ORDER BY id";

        $query_run = mysqli_query($con, $query);

        $row = mysqli_num_rows($query_run);

        echo '<p>' . $row . '</p>';

        ?>

    </div>

</div>

<div class="card-box-1">

    <i class="fas fa-handshake" style="background-image: linear-gradient(-45deg, #ff9966,
#ff5e62);"></i>

    <div>

        <h3>Order</h3>

        <?php
        $query = "SELECT id FROM payment ORDER BY id";

        $query_run = mysqli_query($con, $query);

        $row = mysqli_num_rows($query_run);

        echo '<p>' . $row . '</p>';

        ?>

    </div>

</div>

<div class="card-box-1">

    <i class="fas fa-hotel" style="background-image: linear-gradient(120deg, #fa709a 0%,
#fee140 100%);"></i>

    <div>

        <h3>Restaurant</h3>

```

```

        <?php
        $query = "SELECT id FROM restaurant ORDER BY id";

        $query_run = mysqli_query($con, $query);

        $row = mysqli_num_rows($query_run);

        echo '<p>' . $row . '</p>';

        ?>

    </div>

</div>

<div class="card-box-1">

    <i class="far fa-envelope " style="background-image: linear-gradient(120deg, #2af598
0%, #009efd 100%);"></i>

    <div>

        <h3>Comment</h3>

        <?php

        include "database.php";

        $query = "SELECT id FROM comment ORDER BY id";

        $query_run = mysqli_query($con, $query);

        $row = mysqli_num_rows($query_run);

        echo '<p>' . $row . '</p>';

        ?>

    </div>

</div>

<div class="card-box-1">

    <i class="fas fa-user-tie" style="background-image: linear-gradient(120deg, #667db6,
#0082c8, #0082c8, #667db6);"></i>

```

THANK YOU

THANK YOU

