**ONLINE FOOD ORDERING SYSTEM**

**A Project Report**

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

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)**

**By**

Rajkishore

&

Aman

ROLL NO : 04 & 05

SEAT NO : 2075082(Rajkishore)

2075083(Aman)

**Under the esteemed guidance of**

**Mrs. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Assistant Professor, Department of Information Technology**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**GURU NANAK VIDYAK SOCIETY’S**

**GURU NANAK COLLEGE**

**OF ARTS, SCIENCE AND COMMERCE**

**G.T.B NAGAR**

**MUMBAI, 400 037**

**MAHARASHTRA**

**20-21 GURU NANAK COLLEGE**

**MUMBAI-MAHARASHTRA-400037**

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that the project entitled, **"online food order system "**, is bonafide work of R**ajkishore & Aman Agrahari** submitted in partial fulfillment of the requirements for the award of the degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from the University of Mumbai.

**Internal Guide Coordinator**

**Internal Examiner External Examiner**

**Date: College Seal Principal**

**ACKNOWLEDGEMENT**

First and foremost, we are grateful to our college “**GURU NANAK VIDYAK SOCIETY’S (GURU NANAK COLLEGE)”** for the timely successful completion of this project.

We are indebted to our respected Principal **Dr. PUSHPINDER G.BHATIA** it was their constant support and guidance that kept us in alignment with our project as a great learning experience.

We owe sincere gratitude to our Head of Project **Prof. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**, who supported us and guided us with a continuous source of information. It led to our dedicated and determined approach, right from the selection of the project, working through it and overcoming several obstacles, and completing it within the schedule.

We convey our deep sense of gratitude to all teaching and non-teaching staff of the INFORMATION TECHNOLOGY Department for their constant encouragement, support, and time-to-time help throughout the project work. It is great pleasure to acknowledge the help and suggestion, which we received from the department of computer engineering. We wish to express our profound thanks to all those who helped us in finding information about the project

**DECLARATION**

I hereby declare that the project entitled, “ONLINE FOOD ORDERING SYSTEM” done at guru nanak college, has not been in any case duplicated to submit to any other universities for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as a final semester project as part of our curriculum.

Name and Signature of the Student

Table of Contents

[Chapter 1 Introduction 12](file:/D:/raj/final@blackbook.docx" \l "_Toc34911884)

[1.1 Background 12](file:/D:/raj/final@blackbook.docx" \l "_Toc34911885)

[1.2 Objectives 12](file:/D:/raj/final@blackbook.docx" \l "_Toc34911886)

[1.3 Purpose, Scope , Applicability (Feasibility Study) 12](file:/D:/raj/final@blackbook.docx" \l "_Toc34911887)

[Chapter 2 Survey of Technologies 13](file:/D:/raj/final@blackbook.docx" \l "_Toc34911888)

[Chapter 3 Requirements and Analysis 14](file:/D:/raj/final@blackbook.docx" \l "_Toc34911889)

[3.1 Problem Definition 14](file:/D:/raj/final@blackbook.docx" \l "_Toc34911890)

[3.2 Requirement Specification 14](file:/D:/raj/final@blackbook.docx" \l "_Toc34911891)

[3.3 Planning and Scheduling 14](file:/D:/raj/final@blackbook.docx" \l "_Toc34911892)

[3.4 Software and Hardware Requirement 14](file:/D:/raj/final@blackbook.docx" \l "_Toc34911893)

[Chapter 4 System Design 15](file:/D:/raj/final@blackbook.docx" \l "_Toc34911894)

[4.1 Basic Modules 15](file:/D:/raj/final@blackbook.docx" \l "_Toc34911895)

[4.2 Data Design (Table Design) 15](file:/D:/raj/final@blackbook.docx" \l "_Toc34911896)

[4.2.1 Schema Design 15](file:/D:/raj/final@blackbook.docx" \l "_Toc34911897)

[4.2.2 Data Integrity and Constraints 15](file:/D:/raj/final@blackbook.docx" \l "_Toc34911898)

[4.3 Diagrams 15](file:/D:/raj/final@blackbook.docx" \l "_Toc34911899)

[4.3.1 E-R Diagram /Block Diagram 15](file:/D:/raj/final@blackbook.docx" \l "_Toc34911900)

[4.3.2 Class Diagram / Data Flow Diagram 15](file:/D:/raj/final@blackbook.docx" \l "_Toc34911901)

[4.3.3 Use Case Diagram 15](file:/D:/raj/final@blackbook.docx" \l "_Toc34911902)

[4.3.4 Sequence Diagram 15](file:/D:/raj/final@blackbook.docx" \l "_Toc34911903)

[4.3.5 Activity Diagram 15](file:/D:/raj/final@blackbook.docx" \l "_Toc34911904)

[4.3.6 Component Diagram 16](file:/D:/raj/final@blackbook.docx" \l "_Toc34911905)

[4.3.7 Menu Tree / Circuit Diagram 16](file:/D:/raj/final@blackbook.docx" \l "_Toc34911906)

[4.3.8 Event Table 16](file:/D:/raj/final@blackbook.docx" \l "_Toc34911907)

[4.3.9 User Interface Design 16](file:/D:/raj/final@blackbook.docx" \l "_Toc34911908)

[4.3.10 Security Issues 16](file:/D:/raj/final@blackbook.docx" \l "_Toc34911909)

[4.3.11 Test Cases Design 16](file:/D:/raj/final@blackbook.docx" \l "_Toc34911910)

[References 23](file:/D:/raj/final@blackbook.docx" \l "_Toc34911925)

[Bibliography 24](file:/D:/raj/final@blackbook.docx" \l "_Toc34911926)

[Website Used 25](file:/D:/raj/final@blackbook.docx" \l "_Toc34911927)

[Glossary 26](file:/D:/raj/final@blackbook.docx" \l "_Toc34911928)

[Appendices 27](file:/D:/raj/final@blackbook.docx" \l "_Toc34911929)

[Summary 28](file:/D:/raj/final@blackbook.docx" \l "_Toc34911930)

[Further Reading 29](file:/D:/raj/final@blackbook.docx" \l "_Toc34911931)

[Plagiarism Report 30](file:/D:/raj/final@blackbook.docx" \l "_Toc34911932)

**Abstract :**

ONLINE FOOD ORDER SYSTEM is a website designed primarily for use in the food delivery industry. This system will allow hotels and restaurants to increase scope of business by reducing the labour cost involved. The system also allows to quickly and easily manage an online menu which customers can browse and use to place orders with just few clicks. Restaurant employees then use these orders through an easy to navigate graphical interface for efficient processing. Introduction It is known globally that, in today’s market, it is extremely difficult to start a new small-scale business and live-through the competition from the well-established and settled owners. In fast paced time of today, when everyone is squeezed for time, the majority of people are finicky when it comes to placing a food order. The customers of today are not only attracted because placing an order online is very convenient but also because they have visibility into the items offered, price and extremely simplified navigation for the order. Online ordering system that I am proposing here, greatly simplifies the ordering process for both the customer and the restaurant. System presents an interactive and up-to-date menu with all available options in an easy to use manner. Customer can choose one or more items to place an order which will land in the Cart. Customer can view all the order details in the cart before checking out. At the end, customer gets order confirmation details. Once the order is placed it is entered in the database and retrieved in pretty much real time. This allows Restaurant Employees to quickly go through the orders as they are received and process all orders efficiently and effectively with minimal delays and confusion.

**Background and Related Work :**

This Case study looks at the problem of setting up a fast food restaurant. In existing system there are few problems:

• For placing any orders customers have to visit hotels or restaurants to know about food items and then place order and pay. In this method time and manual work is required.

• While placing an order over the phone, customer lacks the physical copy of the menu item, lack of visual confirmation that the order was placed correctly.

• Every restaurant needs certain employees to take the order over phone or in-person, to offer a rich dining experience and process the payment. In today’s market, labor rates are increasing day by day making it difficult to find employees when needed. Hence, to solve this issue, what I propose is an “Online Food Order System, originally designed for small scale business like College Cafeterias, Fast Food restaurant or Take-Out, but this system is just as applicable in any food delivery industry. The main advantage of my system is that it greatly simplifies the ordering process for both the customer and the restaurant and also greatly lightens the load on the restaurant’s end, as the entire process of taking orders is automated. Anticipated Benefits are:

1. This will minimize the number of employees at the back of the counter.

2. The system will help to reduce labor cost involved.

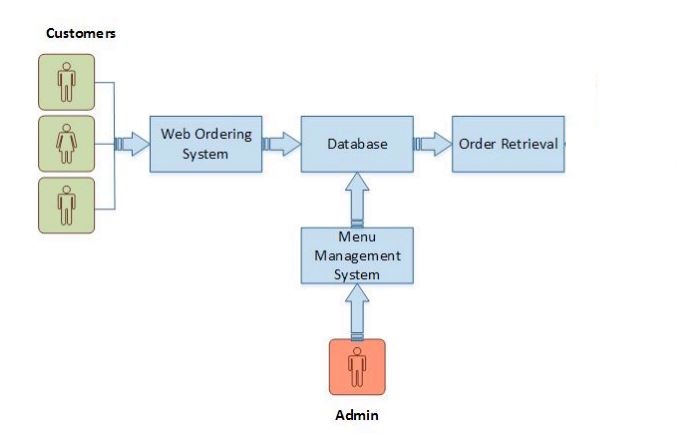
3. The system will be less probable to make mistake, since it’s a machine.

4. This will avoid long queues at the counter due to the speed of execution and number of optimum screens to accommodate the maximum throughput.

**Program Requirements :**

Outline your solution. Describe the "whats" of your project -- what does it do? Product Perspective: The Online Food Order System application is a web-based system. It can be accessed using IE 10.0 and above, Fire Fox 31 and above and Google Chrome.

**System Model:**



The structure of the system can be divided into 3 main logical components:

• Web Ordering System- provides the functionality for customers to place their order and supply necessary details.

• Menu Management-allows the restaurant to control what can be ordered by the customers

• Order Retrieval System-This is a final logical component. Allows restaurant to keep track of all orders placed. This component takes care of order retrieving and displaying order information. Product Function: The Online Food Order System application would have the following basic functions: Web Ordering System Module This module provides the functionality for customers to place their order and supply necessary details. Users of the system, namely restaurant customers, must be provided the following functionality:

• Create an account.

• Manage their account.

• Log in to the system.

• Navigate the restaurant’s menu

• Select an item from the menu.

• Add an item to their current order.

• Review their current order.

• Remove an item/remove all items from their current order.

• Provide payment details.

• Place an order.

• Receive confirmation in the form of an order number.

• View order placed

**Menu Management System Module** :

This module provides functionality for the power user-Administrator only. It will not be available to any other users of the system like Restaurant Employees or Customers. Using a graphical interface, it will allow an Admin to manage the menu that is displayed to users of the web ordering system:

• Add/update/delete food category to/from the menu.

• Add /update/delete food item to/from the menu.

• Update price for a given food item.

• Update additional information (description, photo, etc.) for a given food item. Before customers can actually use this system, functionality provided by this component will have to be configured first. Once the initial configuration is done, this will be the least likely used component as menu updates are mostly seasonal and do not occur frequently.

**Order Retrieval System Module :**

This is the most simplest module out of all 3 modules. It is designed to be used only by restaurant employees, and provides the following functions:

• Retrieve new orders from the database.

• Display the orders in an easily readable, graphical way.

**Implementation :**

Hardware/Software Interface: This section lists the minimum hardware and software requirements needed to run the system efficiently. Hardware Interface:

• Pentium Processor

• 60 MB of free hard-drive space

• 128 MB of RAM Software Interface:

• Operating System: Windows (Vista/7 or above)

• Web Browser: IE 10 or above, Mozilla FF 31 and above or Google Chrome

• Drivers: Java Runtime Environment

• Integrated Development Environment: notepad ++

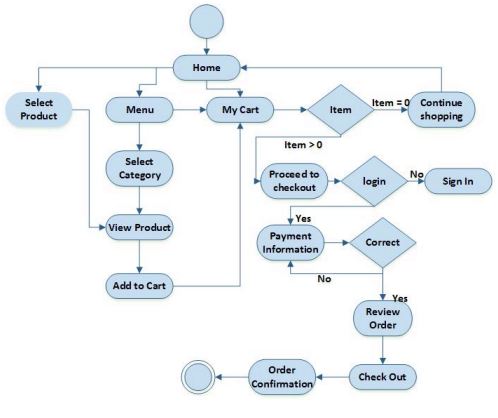
• front end : Html CSS JavaScript photo

• back end : my sql server

**Functional Requirement Specifications:**

**Activity Diagram:**

This section lists the activity diagram and describes the flow of the activities in the system. A detailed description is then given after the figure for each activity. Figure provides the overview of the activity of the Online Food Order System



**Web Ordering System Module :**

Customers of the Web Ordering system will interact with the application through an easy to use top navigation menu.

• “Home” menu option: allows the users to see all food items offered with nice images as well as select an item to place an order

• “Menu”menu option: a ‘Drop-Down’ menu, allows users to see all food items per category. Item can then be added to the cart using a single button click.

• “My Cart (x)”menu option: - Allows users to see details of the items placed in cart. Details include Item #, Product Name, Product Image, Product Description, Quantity, Unit Price, Total per item and final Total of the order. It also allows ‘Update’ and ‘Delete’ an item using single button click. User can then use a ‘Proceed to checkout’ button to proceed further. - Once, Check Out button is selected, user will be prompted for the Sign In/Sign Up process if not logged in else user will be presented with a simple “Payment Information” form. User will be asked to provide all required details in displayed text boxes and make appropriate Dropdown selections. Then, all this information can be saved using a ‘Save’ button. - User will then be presented with a “Review Order” page, which will display Payment Information along with Order details to review. User can then use a ‘Check Out’ button to place an order. - Once order is placed, user will be presented with appropriate Order confirmation success/failure message.

• “MyAccount”: a “Drop Down” menu will display the user orders, Sign In and Sign Out options.

**Menu Management System Module :**

Similar to Web ordering system, this module presents Admin with below additional options under “MyAccount” Drop down menu: • Add Category: Allows to add a food Category name in a simple form. • Add Product: Allows to add Product Name, Description, Price and choose Category in a simple form along with Product Image. • Modify Product: Allows updating or deleting product details.

**Project Aim & Objectives :**

The project aims to build a online food ordering 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 user interface which can be used . 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. sql and Bootstrap 4 will be used for developing the backend and frontend of the system.

4. accessibility will be integrated to the system. So, customer can login and register easliy

5. The system will be tested properly to ensure the quality of the system.

**Order Retrieval System Module :**

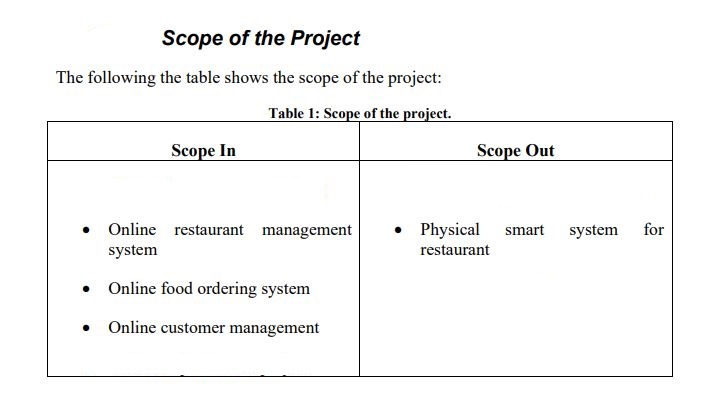
The application will automatically fetch new orders from the database at regular intervals and display the order numbers.

• Under “MyAcoount’ menu a customer will be able to see only his/her order whereas a Restaurant Employee or an Admin can see all users orders.

• To view the details of an order, the user must click on that order number, which will display all order details This structure can intuitively be expanded and collapsed to display only the desired information

**Non-functional Requirements :**

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



**Web Application Development Technologies :**

**PHP .**

PHP is acronym for PHP: Hypertext Preprocessor. PHP is an “open source generalpurpose scripting language that is especially suited for web development and can be embedded into HTML” (The PHP Group, 2018). PHP is mainly used in dynamic web page, including CLI (command line interface) and GUI (graphical user interface) program. It has the feature of good across-platforms and easy transplant (Lei, Ma and Tan, 2014). PHP is not an object-oriented language like Java. But according to Haydar (2007), PHP 5 implements almost complete object-oriented features. 83.1% of all the websites use PHP as server-side programming language and 14.1 of all websites use ASP.NET as server side language by January 2018 (W3Techs, 2018). Along with this, nearly 60% of web servers are running on Apache with PHP (Hayder, 2007). Hence, many researchers tried to compare between these two languages. According to a study of MVC frameworks from PHP and .NET by Jailia et al. (2016), .NET gives better result for page loading time, time to first byte and time to last byte compared to PHP based web applications. While PHP based website gives much more promising result for request and response transfer speed. Bounnady et al., (2016) also made a comparison between PHP and .NET in five different parameters: load static webpage, algorithms processing, managing database, uploading file and reading/writing files. The study concluded that PHP performs better than .NET in managing database operations and uploading files.

**Bootstrap**

As the number of Internet users on mobile devices is growing every moment, websites are no longer built only for the desktop machines. The mobile-first philosophy demands the sites to be fully compatible for all available and future mobile devices. Bootstrap allows and easily enables to design and develop websites congenial to all devices including various screen readers. Bootstrap is a free CSS and JavaScript framework that allows developers to rapidly build responsive web interfaces. The framework was first conceived by Mark Otto and Jacob Thornton while working at Twitter in 2010. Since its release as an open source project in August of 2011, Bootstrap has become a household name among frontend web developers (Jakobus and Marah, 2016). Bootstrap is the most starred project in GitHub since 2012. Bootstrap is largely based on CSS3 that is currently the latest version of Cascading Style Sheet (CSS) and included with powerful capabilities like Media queries, Selectors, Cascading and Inheritance, Template layouts, Namespaces, Flexible and Grid layouts etc. (Bhaumik, 2015). Bootstrap 4 has improved grid system and inclusion of flexbox. For regular grid, another regular grid level is introduced for better targeting smartphones (Lambert, 2016).

**Sql**

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

**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.

**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 analysed 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.

**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 analyse the system and make logical structure for it

**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

**Critical Analysis of the System**

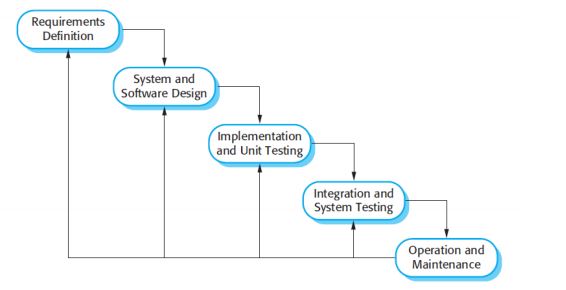
After implementation, the system has been critically evaluated to understand if it meets the requirements. Then the result is analysed. In this step, conclusion and future recommendation has been made

**Software Development Process**

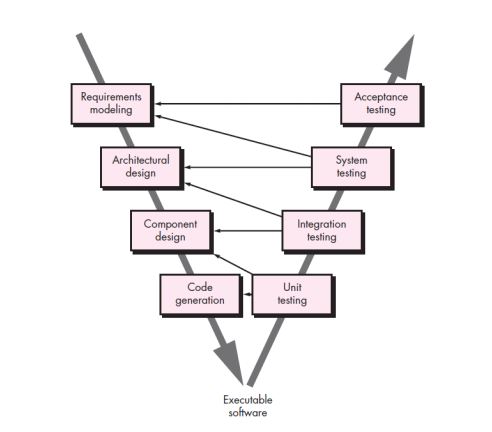
Software Development Process commonly comprises sequence of work activities, actions, and tasks that are undergone to create the final product. In the context of software development, the final product is a software application, a plug-in component, or a software service solution. However, software development processes are complex and unmanaged process could easily lead to catastrophic failure in delivering a usable system. While there are many factors that contribute to its complexity, the two main reasons described by [21] and [22] are: 1) Intellectual and creative processes rely on people’s decisions and judgement; and 2) The environment may vary hence producing rapidly changing software requirements or strictly defined criteria. Consequently, careful planning of development activities is required and this results in the adoption of software development process model. A software process model is an abstract representation of interrelated activities in software development [22, 23]. It describes the general approaches in structuring activities and some techniques to produce deliverables. Selecting a suitable model for WRCS would cut down the development time and increase the quality of the output. Following sections describes several widely applied software process model in the software industry.

**Waterfall Model**

Waterfall model is a traditional software process model introduced by Royce [24]. It is a rigid and linear document driven methodology. This model is known as the waterfall model because it proceeded from one phase to another in a cascading order as shown in Figure 2.1. Before each phase can begin, each of the phases has a definite set of deliverables that must be approved by project sponsor, after the stakeholders have elicited them. However, the process of producing and approving these deliverables will incur significant cost. The waterfall model often receives criticism on its inability to accommodate changes because the project freezes system specification upon deliverables sign-off. In a dynamic business environment, it is often difficult for user to state all requirements explicitly. The waterfall model lacks the ability to accommodate natural uncertainty and the changing need of users.



Another serious disadvantage of the waterfall model is that testing is often left to the end of the project. Errors and feedbacks obtained in later stages will require additional effort to resolve. Eventually, this will lead to a software product that not fit for user need. An enhanced variant of the waterfall model known as the V-model has improved to this issue. Figure 2.2 illustrates the quality assurance actions associated with deliverables of earlier phases in the V-model. Verification and validation approaches applied to earlier engineering work could significantly reduce errors found in later stages. However, the V-model does not explicitly describe actions taken in order to deal with errors found during testing. Nevertheless, waterfall model does show its strength when used in project where requirements are well understood and stable during development [22]. Documents produced during each phase provide traceability to address safety and legal issues when such concerns are critical to the user.



**Evolutionary Mode**

l Evolutionary model encapsulates two fundamental approaches: incremental and interactive; when addressing changing requirements. It organizes processes in a manner that enables the development of increasingly complete versions of software based on customer feedbacks through a series of iteration. The two fundamental types of evolutionary model that will be covered are prototyping and spiral model. The idea of prototyping is to enable users to interact and experiment with early prototypes which encapsulate a set of mutual understanding requirements. In [22], prototypes are described as an initial version of a software system used to demonstrate concepts, explore design options, in-depth problems and their possible solutions. Commonly, there are two types of prototype [22, 25] :

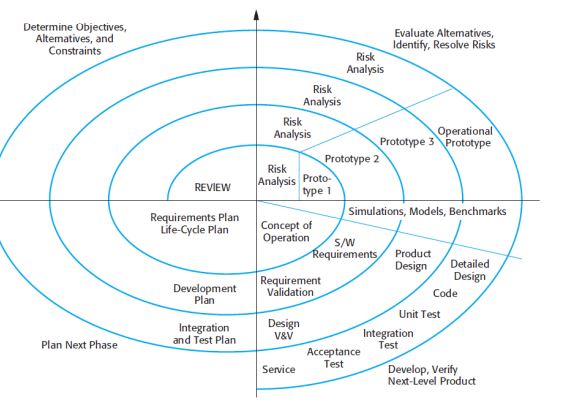
Common prototypes aim to explore customer requirements through building an incrementally usable system. Prototypes with a minimum set of basic requirements are built and presented for the customer’s evaluation. The prototypes evolves by the implementation of customer proposed features and changes until it’s functionalities finally agreed by customer; and

Throwaway prototypes aim to gather information and generate ideas on how system should be built. Commonly during project start up, the user may not fully understand their need and the developer may not share understanding on certain features. To clarify these uncertainties, a design prototype [25] which contain just enough details is built for evaluation. Once issues have been clarified, developers could then move on to an actual design and implementation

Allowing requirements to be implemented rapidly is the key advantage to prototyping. However, this may lead to stakeholder confusion by mistreating what they see as final version of the system. Stakeholders should be well aware that some prototypes only serve as tools to gather requirements and may vary from the final product.

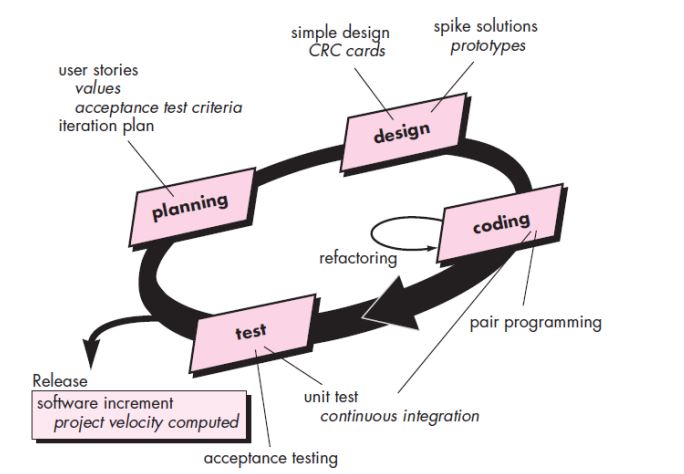
**Spiral model**

The spiral model is an evolutionary process model that combines the iterative nature of prototyping while retaining the systematic approaches of waterfall model [23]. Unlike the waterfall model in which it is hard to backtrack to previous phases once deliverables freeze, spiral model can be adopted throughout system development phases as shown in Figure 2.3. Explicit recognition of risk in the spiral model is the main difference compared with other process model [22]. However, understanding and mitigating the risks potentially reduces things that can go wrong. Iteration over entire phases of software process would be costly; hence, the spiral model is more suitable for large-scale projects, which contains high risk and requires well-

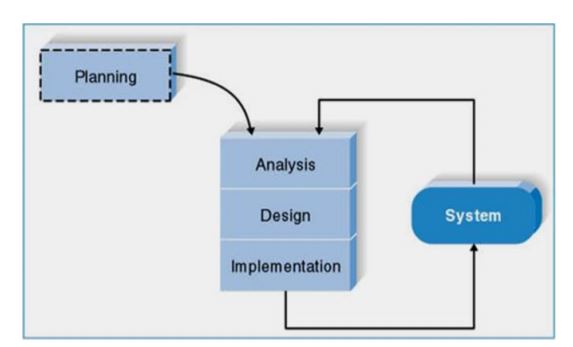
structured approaches. ****

**Agile Development**

Agile development processes have emerged to be the dominant software process model in recent years. Agile processes focuses on people, communication, working software, and responding to change as opposed to plan-driven models that have high process bureaucracy. These are best explained with Agile Manifesto4 . Design and implementation are the central activities in agile development processes [22]. It would also be possible to incorporate requirements elicitation and testing into these activities, for instance, applying test-driven development (TDD). In TDD, the developer first writes test cases before writing actual implementations. This serves as the preliminary steps to clarify requirements and understanding for problem domains. Developers then code the actual implementations and execute tests to verify the implementations.

****

Extreme programming (XP) has been widely known approach since the introduction of agile development concept. Figure 2.4 shows the XP processes and its practise during each phase. XP captures requirements in the form of customer stories or scenarios to determine the features required. In XP, continuous customer engagement in development is important foreedback and acceptance testing. XP favours small and frequent releases of software version like any other agile methods. Thus, design should only meet the current needs and expect refactoring when future improvement is required. XP recommended pair programming among developers because it can enable real time problem solving and quality assurance on solution applied [23]. XP is a lightweight process and fits well for small size projects. However, in a large-scale project where physical interaction among team members is difficult, it could be challenging for XP principles.



• 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 analysed.

• Design: In this phase of the SDLC, the logical model of database and the interface of the system is designed. The normalisation 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.

**Comparison among Software Process Model**

Table 2.1 presents a comparison of the three models discussed above based on several concerns that may affect WRCS development activities. These concerns, together with their explanations, are listed below:

Requirement elicitation, presents approaches to gather requirements for system;

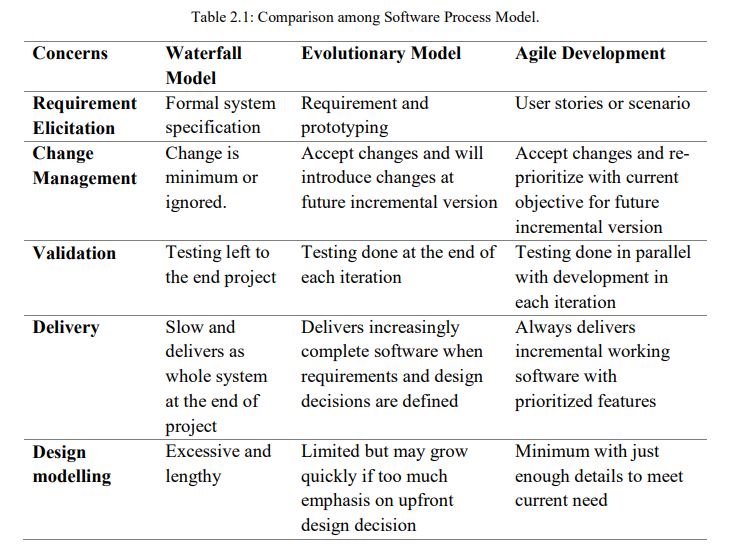
Change management, reflects how changes will be handled throughout project;

Validation, explains when testing will be done during project;

Delivery discuss how quickly and often the software features will be delivered; and

Design modelling covers the depth of design processes during modelling activity.

Based on the comparison, agile development clearly exhibited features that meet WCRS needs. WCRS will require segregation of user tasks and roles to model intuitive UI. Apparently, user stories of agile development fit better with these requirements. In addition, agile development has factored change management in the model. Its ability to cope with changes reduces the risk of delivering products that does not meet the objectives. The earlier the system is tested, the less effort will be spent on the error that may arise in end of the project. TDD of agile practise embraces this idea and encourages testing done before development. Connected to this, frequent delivery also implies that new enhancement have actually been verified in smaller scale. It reduces complexity by testing only parts that have been changed. Upfront design often leads to “design paralysis5” [26] when the developer tries to adopt concerns and considerations that may not be materialized in the future of project. This is why agile development prefers modelling just enough detail to support current need and refactor as required. Finally, lightweight agile process such as XP fits well into small-scale development, as in WRCS, which involves only single developer.

****

**Requirements & Design**

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.

**Requirements Elicitation**

The requirements of a system is characteristics of a system it needs to have. The requirements has been collected in the planning phase of the SDLC. Different kinds of data collection methods have been utilised to obtain the requirements of the system, which are explained above.

**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”

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.

• The Application must have a shopping cart for ordering foods.

• The application must have admin registration and login system.

• The application must have menu add and edit options for admin.

**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”

Following are the non-functional requirements of the project.

• Users should get confirmation and warning message.

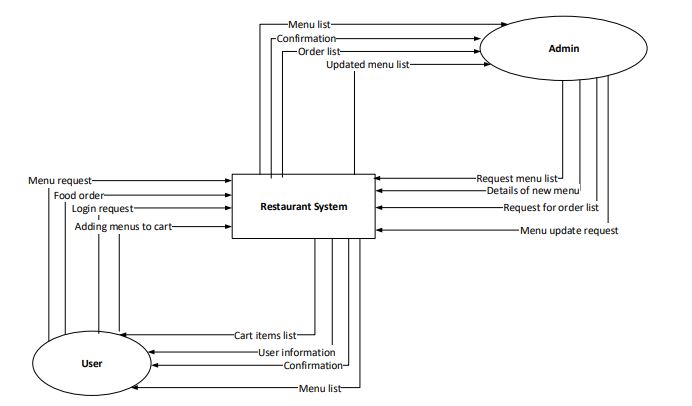
• The user interface must be user-friendly.

**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-ViewController (MVC) design pattern. This 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.

**Context Diagram**

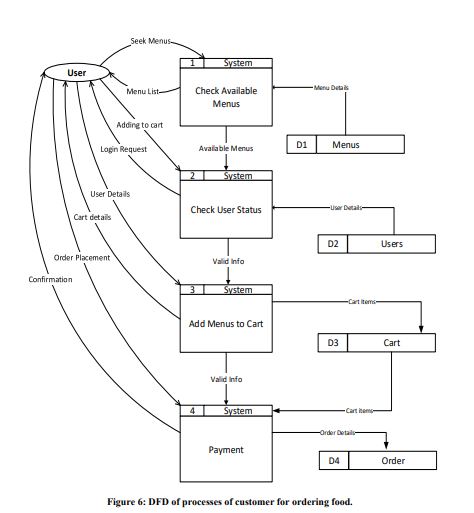
(CD) Context diagrams define “how the business process or computer system interacts with its environment” Context diagrams are used early in a project to describe the entities of the system. It shows the external entities and data flows into and out of the system. The processes and data stores are not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

****

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

**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.



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.

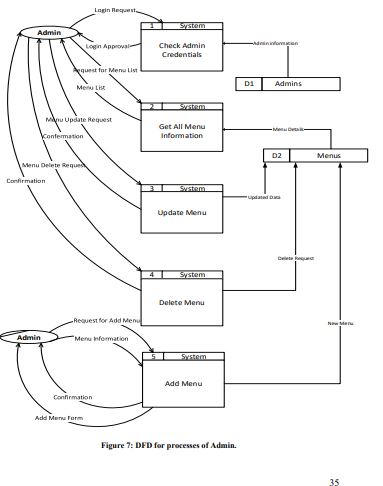


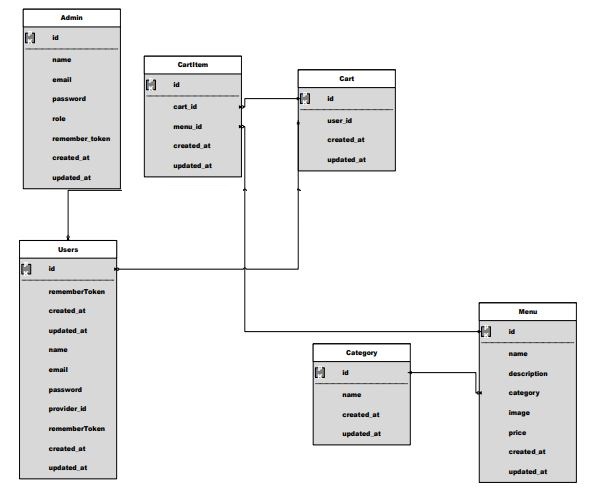
Figure 7 illustrates the processes of admin in for manipulating Menus table of the database. There are five processes for admin in this application. Process 1 depicts the login of admin to the system. When

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.

**Data Modelling :**

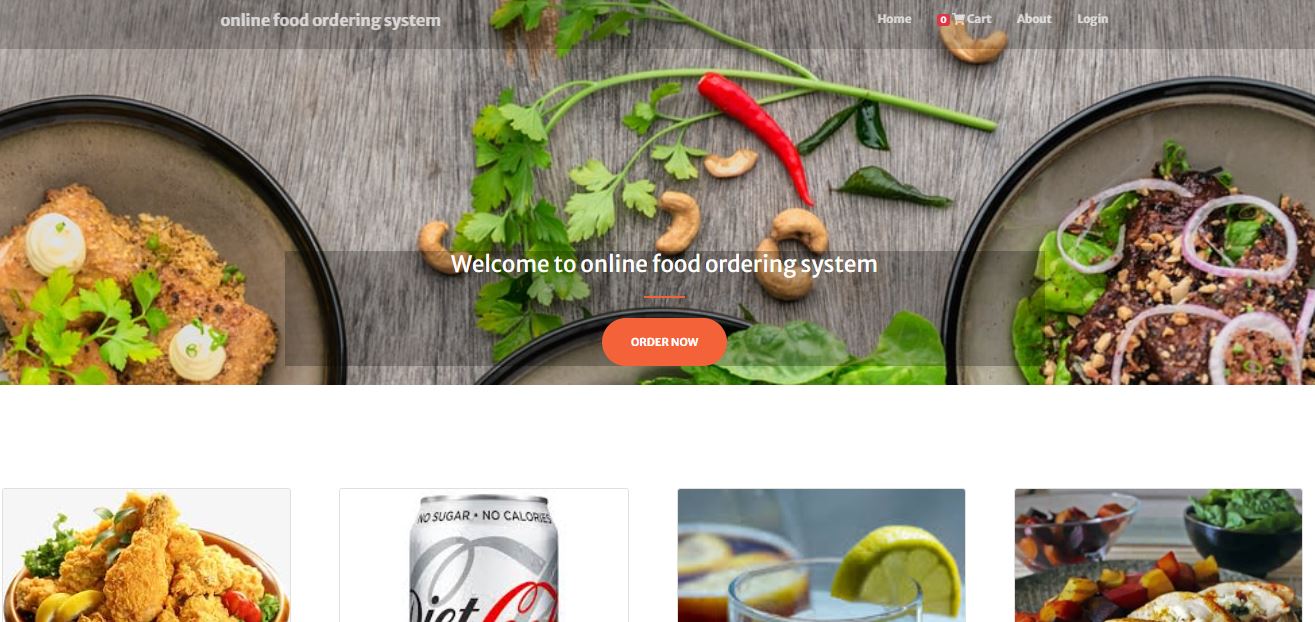
**Entity Relationship Diagram (ERD)**

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” . The following ERD illustrates the database design for restaurant application.

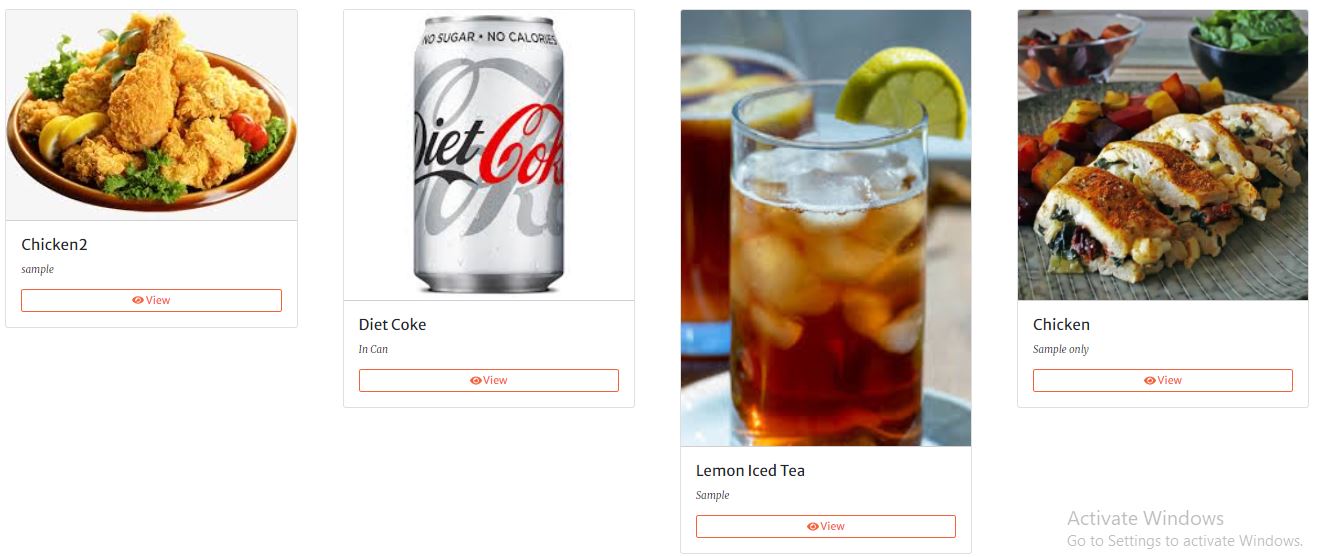
****

**Homepage :**

The homepage is the index page of the application. This is the page that is shown to visitor when s/he accessed the website through URL

****

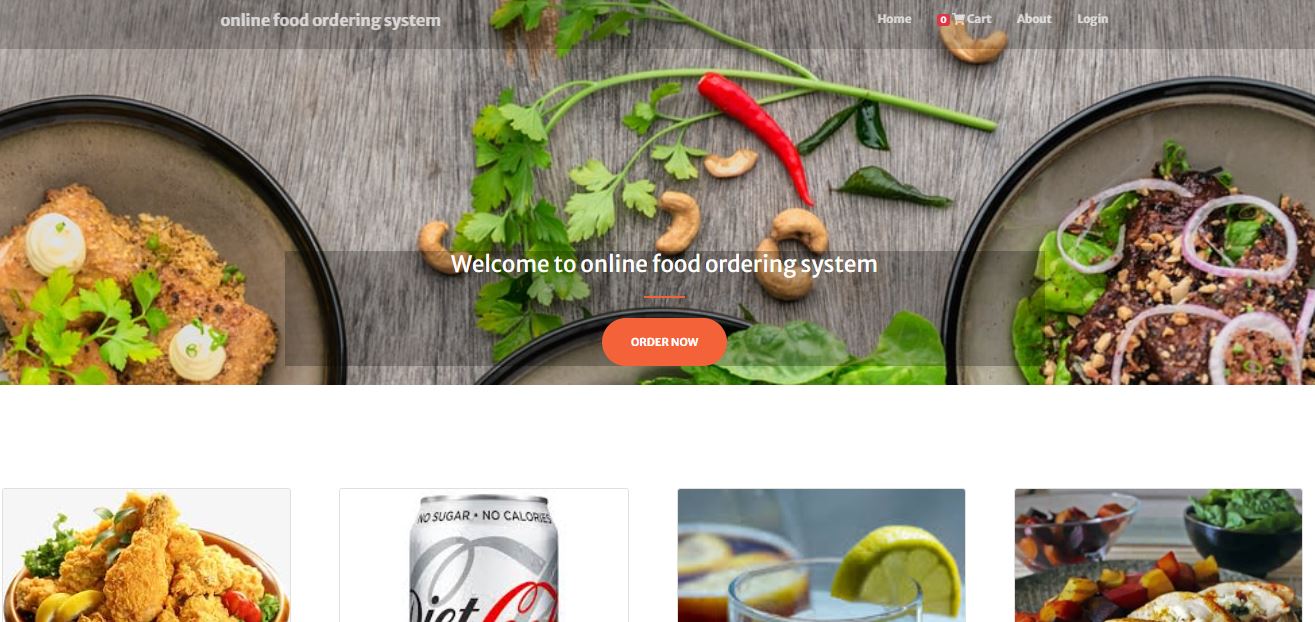
**Menu :**

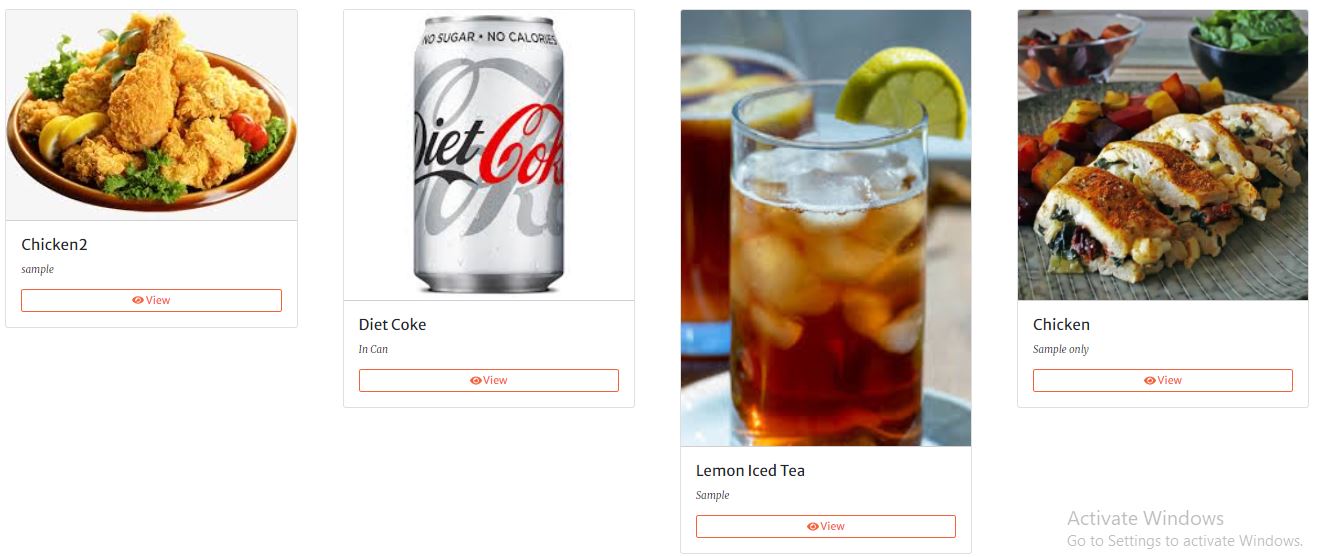
The purpose of menu list page is to display all of the menu items to users. The page shows the list of the menus with image and price. users can add menus to cart from this page or can view the details of the menu. ****

**Implementation :**

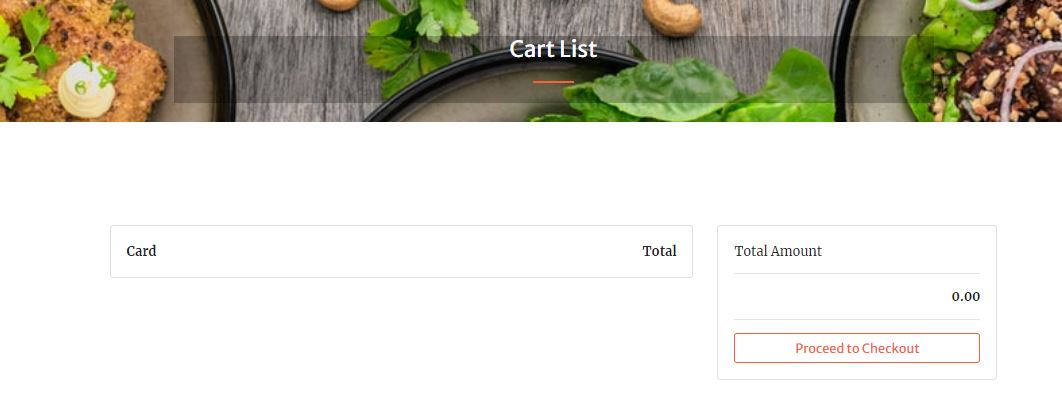
provides an overview of the implementation process. In first, the implementation of the frontend part is described. After that, different backend functionalities have been discussed and evaluated.

**Home page :**

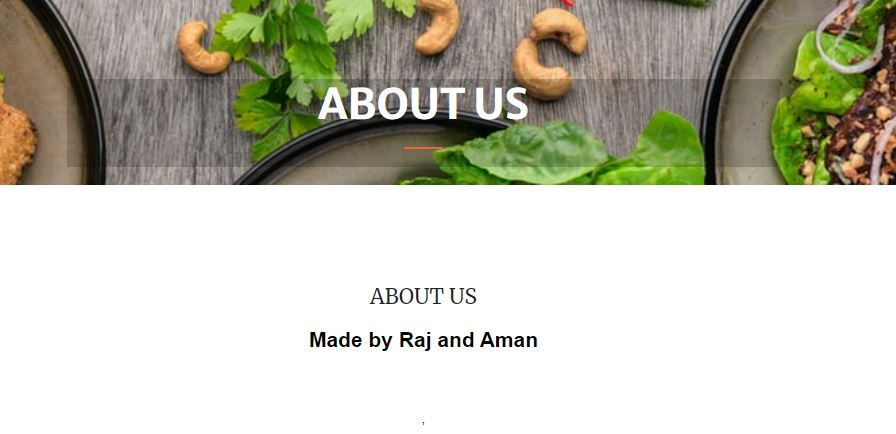
****

**Menu page : **

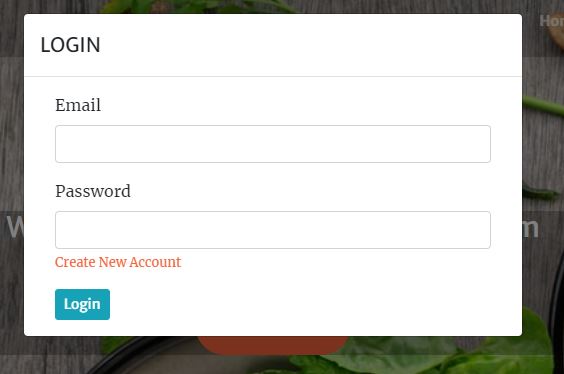
**Cart :**



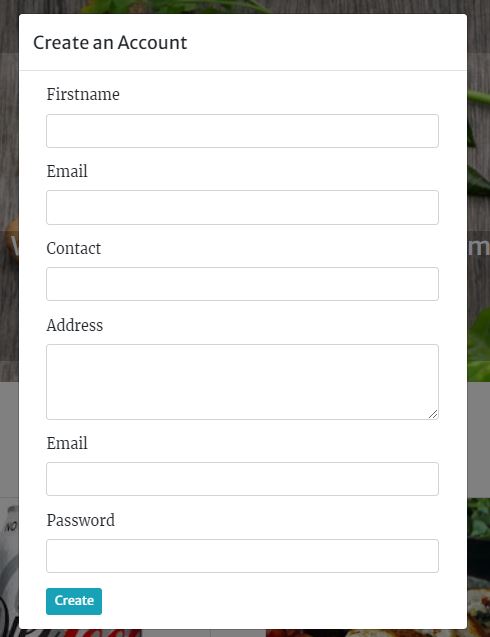
**About :**

****

**Login :**

****

**Sign up :**

****

**Testing :**

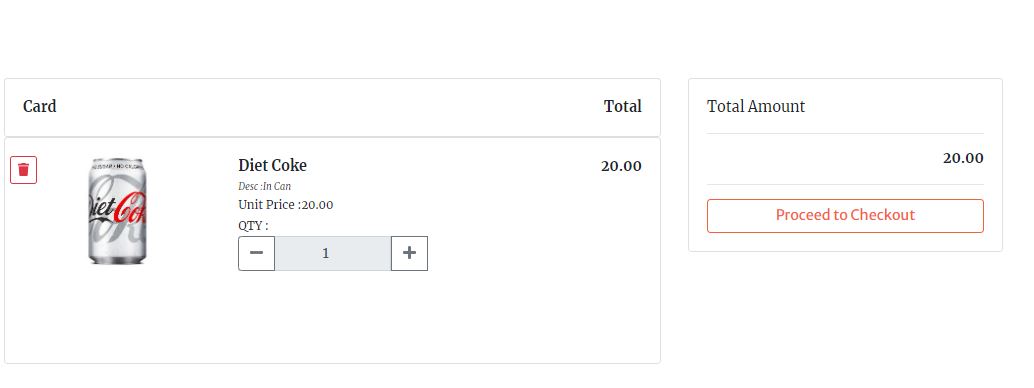
Testing can be divided into two types broadly: functional testing and structural testing. Structural testing, also known as white-box testing, involves examining the internal implementation. It tests the design used by the implementation to verify it correctness. In contrast, Functional testing, sometimes referred to 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.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test code | Test Case | Test Steps | Expected Result | Actual Result | Pass/ Fail |
| Test 1 | Check login functionality with logged in user | 1. Go to registration page  2. Click on “Login with” button | The window is redirected to, then to homepage with showing the timeline name on the top right of the bar. The name and details also saved in database. | As expected | pass |
| Test 2 | Check login functionality with non-logged in user | 1. Insert un logged user email.  2. insert non logged user password | 1. Message showing wrong email or password | As expected | pass |
| Test 3 | Check adding to cart functionality for logged in user | 1. Go to any menu details page 2. Click on add to menu button | The user is redirected to cart page where the subtotal and total price shown. | As expected | pass |
| Test 4 | Check adding a menu twice to cart | 1. Go to any menu details page 2. Click on add to menu button 3. Get back to menu page 4. Add the same menu to cart | The user is redirected to cart page where the updated subtotal and total price shown. | As expected | pass |
| Test 1 |  |  |  | As expected | pass |

**Login :**

****

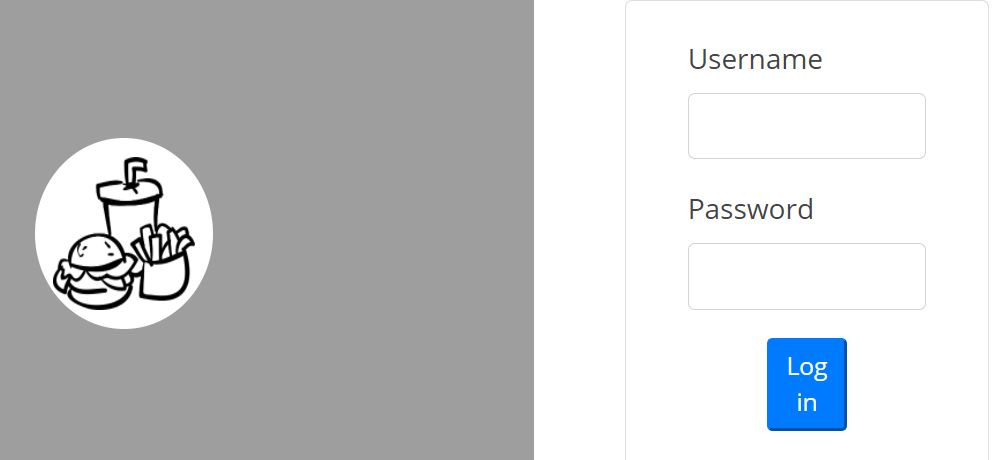
**Cart :**

****

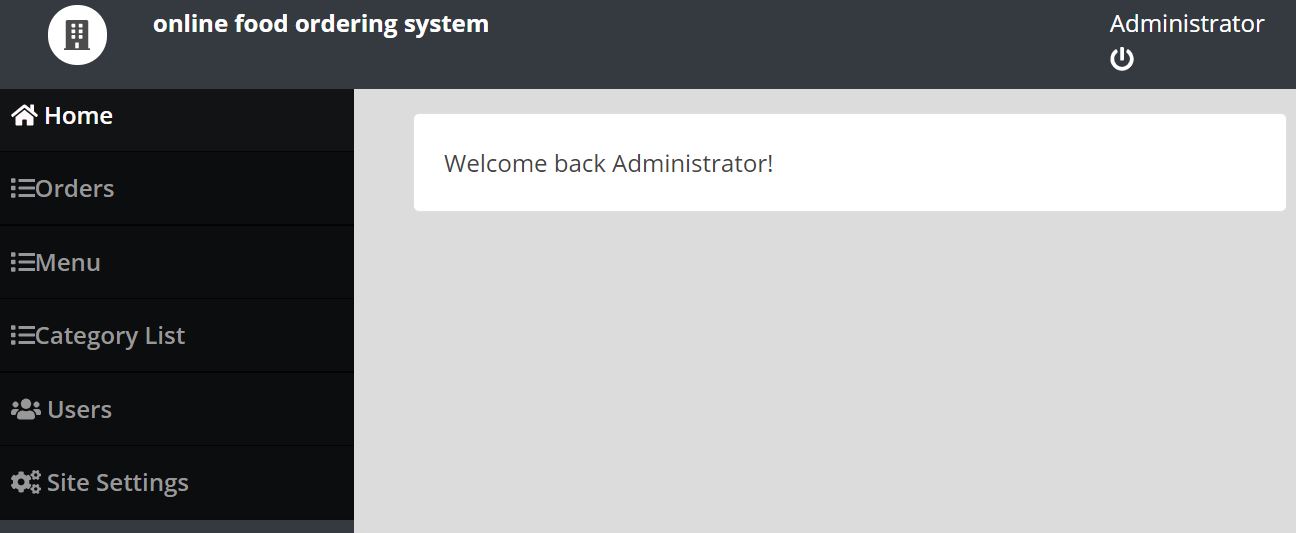
**Confirmation :**

****

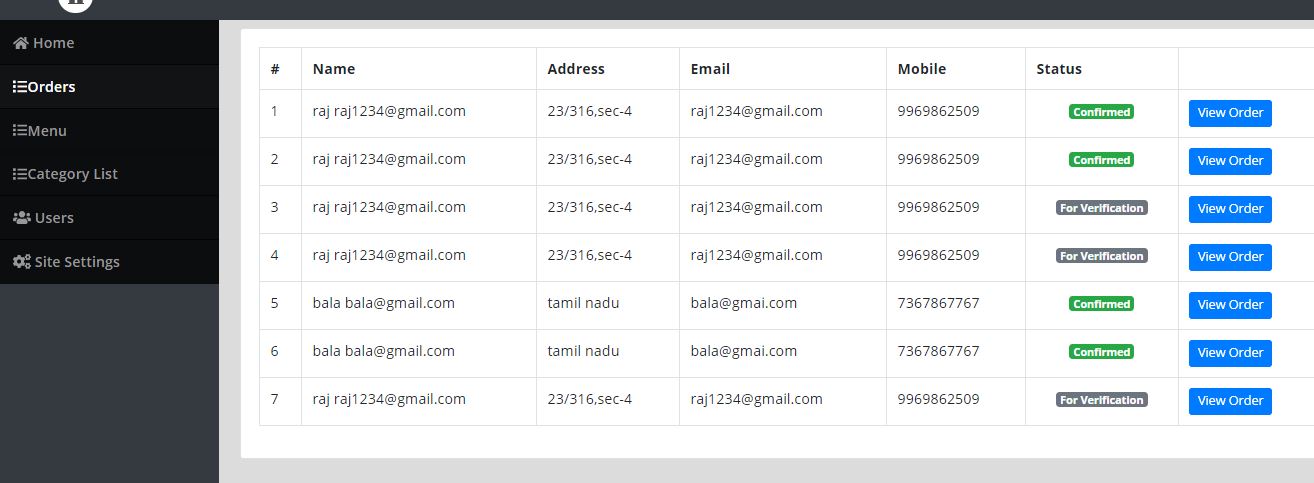
**Admin :**

****

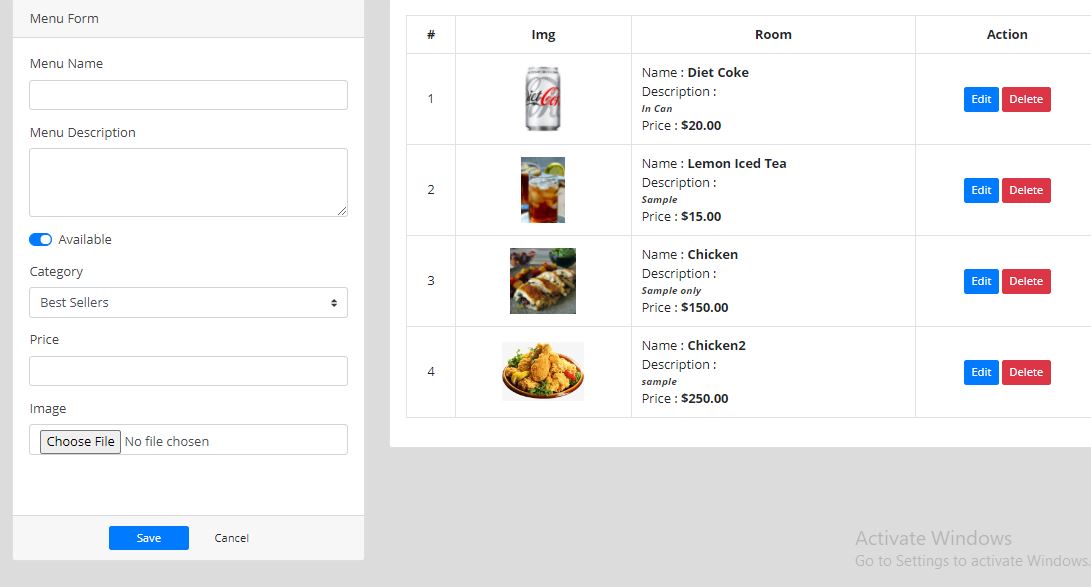
**Admin screen :**

****

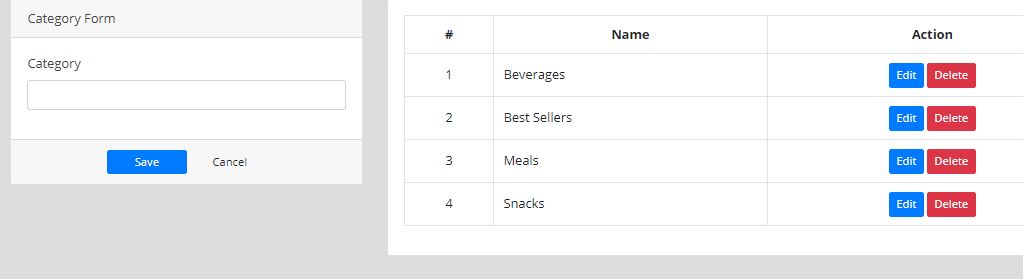
**Order :**

****

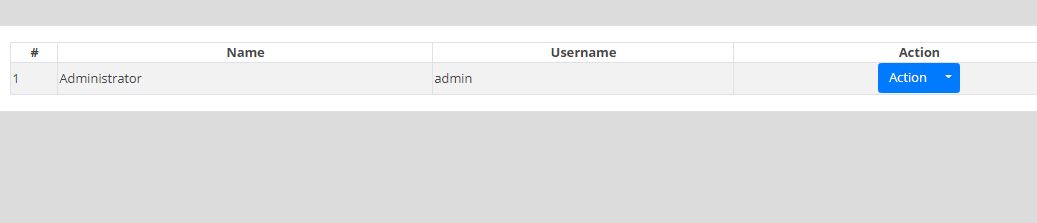
**Menu :**

****

**Category :**

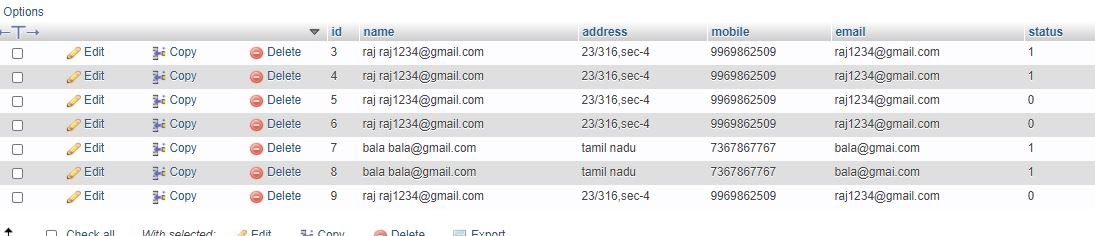
****

**Admins :**

****

**Database**

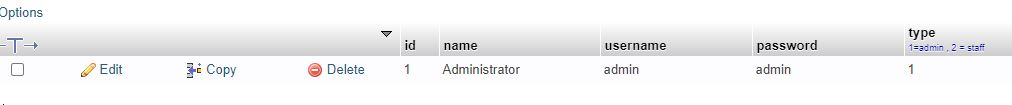
**Order :**

****

**Users :**

****

**Admin :**

****

**Coding :**

**User about**

|  |
| --- |
| **<!DOCTYPE html>**  **<html lang="en">**  **<meta charset="utf-8" />**  **<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />**  **<meta name="description" content="" />**  **<meta name="author" content="" />**  **<title>Online Food Ordering System</title>**  **<!-- Favicon-->**  **<link rel="icon" type="image/x-icon" href="assets/img/favicon.ico" />**  **<!-- Font Awesome icons (free version)-->**  **<script src="https://use.fontawesome.com/releases/v5.13.0/js/all.js" crossorigin="anonymous"></script>**  **<!-- Google fonts-->**  **<link href="https://fonts.googleapis.com/css?family=Merriweather+Sans:400,700" rel="stylesheet" />**  **<link href="https://fonts.googleapis.com/css?family=Merriweather:400,300,300italic,400italic,700,700italic" rel="stylesheet" type="text/css" />**  **<!-- Third party plugin CSS-->**  **<link href="https://cdnjs.cloudflare.com/ajax/libs/magnific-popup.js/1.1.0/magnific-popup.min.css" rel="stylesheet" />**  **<!-- Core theme CSS (includes Bootstrap)-->**  **<link href="admin/assets/vendor/bootstrap-datepicker/css/bootstrap-datepicker.css" rel="stylesheet" />**  **<link href="css/styles.css" rel="stylesheet" />**  **<script src="admin/assets/vendor/jquery/jquery.min.js"></script>**  **<script src="admin/assets/vendor/bootstrap-datepicker/js/bootstrap-datepicker.js"></script>**  **<style>**  **header.masthead {**  **background: url(assets/img/1600654680\_photo-1504674900247-0877df9cc836.jpg);**  **background-repeat: no-repeat;**  **background-size: cover;**  **}**  **</style>**  **<body id="page-top">**  **<!-- Navigation-->**  **<div class="toast" id="alert\_toast" role="alert" aria-live="assertive" aria-atomic="true">**  **<div class="toast-body text-white">**  **</div>**  **</div>**  **<nav class="navbar navbar-expand-lg navbar-light fixed-top py-3" id="mainNav">**  **<div class="container">**  **<a class="navbar-brand js-scroll-trigger" href="./">Online Food Ordering System</a>**  **<button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>**  **<div class="collapse navbar-collapse" id="navbarResponsive">**  **<ul class="navbar-nav ml-auto my-2 my-lg-0">**  **<li class="nav-item"><a class="nav-link js-scroll-trigger" href="index.php?page=home">Home</a></li>**  **<li class="nav-item"><a class="nav-link js-scroll-trigger" href="index.php?page=cart\_list"><span> <span class="badge badge-danger item\_count">0</span> <i class="fa fa-shopping-cart"></i> </span>Cart</a></li>**  **<li class="nav-item"><a class="nav-link js-scroll-trigger" href="index.php?page=about">About</a></li>**  **<li class="nav-item"><a class="nav-link js-scroll-trigger" href="javascript:void(0)" id="login\_now">Login</a></li>**  **</ul>**  **</div>**  **</div>**  **</nav>**    **<!-- Masthead-->**  **<header class="masthead">**  **<div class="container h-100">**  **<div class="row h-100 align-items-center justify-content-center text-center">**  **<div class="col-lg-10 align-self-end mb-4" style="background: #0000002e;">**  **<h1 class="text-uppercase text-white font-weight-bold">About Us</h1>**  **<hr class="divider my-4" />**  **</div>**    **</div>**  **</div>**  **</header>**  **<section class="page-section">**  **<div class="container">**  **<p style="text-align: center; background: transparent; position: relative;"><span style="font-size:28px;background: transparent; position: relative;">ABOUT US</span></b></span></p><p style="text-align: center; background: transparent; position: relative;"><span style="background: transparent; position: relative; font-size: 14px;"><span style="font-size:28px;background: transparent; position: relative;"><b style="margin: 0px; padding: 0px; color: rgb(0, 0, 0); font-family: &quot;Open Sans&quot;, Arial, sans-serif; text-align: justify;"> </b><span style="color: rgb(0, 0, 0); font-family: &quot;Open Sans&quot;, Arial, sans-serif; font-weight: 400; text-align: justify;">&nbsp; </span><br></span></b></span></p><p style="text-align: center; background: transparent; position: relative;"><span style="background: transparent; position: relative; font-size: 14px;"><span style="font-size:28px;background: transparent; position: relative;"><span style="color: rgb(0, 0, 0); font-family: &quot;Open Sans&quot;, Arial, sans-serif; font-weight: 400; text-align: justify;"><br></span></b></span></p><p style="text-align: center; background: transparent; position: relative;"><span style="background: transparent; position: relative; font-size: 14px;"><span style="font-size:28px;background: transparent; position: relative;"><h2 style="font-size:28px;background: transparent; position: relative;"></h2><p style="text-align: center; margin-bottom: 15px; padding: 0px; color: rgb(0, 0, 0); font-family: &quot;Open Sans&quot;, Arial, sans-serif; font-weight: 400;"> </p></span></b></span></p>**    **</div>**  **</section>**  **<!DOCTYPE html>**  **<html lang="en">**  **<meta charset="utf-8" />**  **<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />**  **<meta name="description" content="" />**  **<meta name="author" content="" />**  **<title>Online Food Ordering System</title>**  **<!-- Favicon-->**  **<link rel="icon" type="image/x-icon" href="assets/img/favicon.ico" />**  **<!-- Font Awesome icons (free version)-->**  **<script src="https://use.fontawesome.com/releases/v5.13.0/js/all.js" crossorigin="anonymous"></script>**  **<!-- Google fonts-->**  **<link href="https://fonts.googleapis.com/css?family=Merriweather+Sans:400,700" rel="stylesheet" />**  **<link href="https://fonts.googleapis.com/css?family=Merriweather:400,300,300italic,400italic,700,700italic" rel="stylesheet" type="text/css" />**  **<!-- Third party plugin CSS-->**  **<link href="https://cdnjs.cloudflare.com/ajax/libs/magnific-popup.js/1.1.0/magnific-popup.min.css" rel="stylesheet" />**  **<!-- Core theme CSS (includes Bootstrap)-->**  **<link href="admin/assets/vendor/bootstrap-datepicker/css/bootstrap-datepicker.css" rel="stylesheet" />**  **<link href="css/styles.css" rel="stylesheet" />**  **<script src="admin/assets/vendor/jquery/jquery.min.js"></script>**  **<script src="admin/assets/vendor/bootstrap-datepicker/js/bootstrap-datepicker.js"></script>**  **<style>**  **header.masthead {**  **background: url(assets/img/1600654680\_photo-1504674900247-0877df9cc836.jpg);**  **background-repeat: no-repeat;**  **background-size: cover;**  **}**  **</style>**  **<body id="page-top">**  **<!-- Navigation-->**  **<div class="toast" id="alert\_toast" role="alert" aria-live="assertive" aria-atomic="true">**  **<div class="toast-body text-white">**  **</div>**  **</div>**  **<nav class="navbar navbar-expand-lg navbar-light fixed-top py-3" id="mainNav">**  **<div class="container">**  **<a class="navbar-brand js-scroll-trigger" href="./">Online Food Ordering System</a>**  **<button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>**  **<div class="collapse navbar-collapse" id="navbarResponsive">**  **<ul class="navbar-nav ml-auto my-2 my-lg-0">**  **<li class="nav-item"><a class="nav-link js-scroll-trigger" href="index.php?page=home">Home</a></li>**  **<li class="nav-item"><a class="nav-link js-scroll-trigger" href="index.php?page=cart\_list"><span> <span class="badge badge-danger item\_count">0</span> <i class="fa fa-shopping-cart"></i> </span>Cart</a></li>**  **<li class="nav-item"><a class="nav-link js-scroll-trigger" href="index.php?page=about">About</a></li>**  **<li class="nav-item"><a class="nav-link js-scroll-trigger" href="javascript:void(0)" id="login\_now">Login</a></li>**  **</ul>**  **</div>**  **</div>**  **</nav>**    **<!-- Masthead-->**  **<header class="masthead">**  **<div class="container h-100">**  **<div class="row h-100 align-items-center justify-content-center text-center">**  **<div class="col-lg-10 align-self-end mb-4" style="background: #0000002e;">**  **<h1 class="text-uppercase text-white font-weight-bold">About Us</h1>**  **<hr class="divider my-4" />**  **</div>**    **</div>**  **</div>**  **</header>**  **<section class="page-section">**  **<div class="container">**  **<p style="text-align: center; background: transparent; position: relative;"><span style="font-size:28px;background: transparent; position: relative;">ABOUT US</span></b></span></p><p style="text-align: center; background: transparent; position: relative;"><span style="background: transparent; position: relative; font-size: 14px;"><span style="font-size:28px;background: transparent; position: relative;"><b style="margin: 0px; padding: 0px; color: rgb(0, 0, 0); font-family: &quot;Open Sans&quot;, Arial, sans-serif; text-align: justify;">Made by Raj and Aman</b><span style="color: rgb(0, 0, 0); font-family: &quot;Open Sans&quot;, Arial, sans-serif; font-weight: 400; text-align: justify;"> </span><br></span></b></span></p><p style="text-align: center; background: transparent; position: relative;"><span style="background: transparent; position: relative; font-size: 14px;"><span style="font-size:28px;background: transparent; position: relative;"><span style="color: rgb(0, 0, 0); font-family: &quot;Open Sans&quot;, Arial, sans-serif; font-weight: 400; text-align: justify;"><br></span></b></span></p><p style="text-align: center; background: transparent; position: relative;"><span style="background: transparent; position: relative; font-size: 14px;"><span style="font-size:28px;background: transparent; position: relative;"><h2 style="font-size:28px;background: transparent; position: relative;"> </h2><p style="text-align: center; margin-bottom: 15px; padding: 0px; color: rgb(0, 0, 0); font-family: &quot;Open Sans&quot;, Arial, sans-serif; font-weight: 400;">, </p></span></b></span></p>**    **</div>**  **</section>**  **<div class="modal fade" id="confirm\_modal" role='dialog'>**  **<div class="modal-dialog modal-md" role="document">**  **<div class="modal-content">**  **<div class="modal-header">**  **<h5 class="modal-title">Confirmation</h5>**  **</div>**  **<div class="modal-body">**  **<div id="delete\_content"></div>**  **</div>**  **<div class="modal-footer">**  **<button type="button" class="btn btn-primary" id='confirm' onclick="">Continue</button>**  **<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>**  **</div>**  **</div>**  **</div>**  **</div>**  **<div class="modal fade" id="uni\_modal" role='dialog'>**  **<div class="modal-dialog modal-md" role="document">**  **<div class="modal-content">**  **<div class="modal-header">**  **<h5 class="modal-title"></h5>**  **</div>**  **<div class="modal-body">**  **</div>**  **<div class="modal-footer">**  **<button type="button" class="btn btn-primary" id='submit' onclick="$('#uni\_modal form').submit()">Save</button>**  **<button type="button" class="btn btn-secondary" data-dismiss="modal">Cancel</button>**  **</div>**  **</div>**  **</div>**  **</div>**  **<div class="modal fade" id="uni\_modal\_right" role='dialog'>**  **<div class="modal-dialog modal-full-height modal-md" role="document">**  **<div class="modal-content">**  **<div class="modal-header">**  **<h5 class="modal-title"></h5>**  **<button type="button" class="close" data-dismiss="modal" aria-label="Close">**  **<span class="fa fa-arrow-righ t"></span>**  **</button>**  **</div>**  **<div class="modal-body">**  **</div>**  **</div>**  **</div>**  **</div>**  **<footer class="bg-light py-5">**  **<div class="container"><div class="small text-center text-muted"> <a href="#" target="\_blank"></a></div></div>**  **</footer>**    **<script>**  **$('.datepicker').datepicker({**  **format:"yyyy-mm-dd"**  **})**  **window.start\_load = function(){**  **$('body').prepend('<di id="preloader2"></di>')**  **}**  **window.end\_load = function(){**  **$('#preloader2').fadeOut('fast', function() {**  **$(this).remove();**  **})**  **}**  **window.uni\_modal = function($title = '' , $url=''){**  **start\_load()**  **$.ajax({**  **url:$url,**  **error:err=>{**  **console.log()**  **alert("An error occured")**  **},**  **success:function(resp){**  **if(resp){**  **$('#uni\_modal .modal-title').html($title)**  **$('#uni\_modal .modal-body').html(resp)**  **$('#uni\_modal').modal('show')**  **end\_load()**  **}**  **}**  **})**  **}**  **window.uni\_modal\_right = function($title = '' , $url=''){**  **start\_load()**  **$.ajax({**  **url:$url,**  **error:err=>{**  **console.log()**  **alert("An error occured")**  **},**  **success:function(resp){**  **if(resp){**  **$('#uni\_modal\_right .modal-title').html($title)**  **$('#uni\_modal\_right .modal-body').html(resp)**  **$('#uni\_modal\_right').modal('show')**  **end\_load()**  **}**  **}**  **})**  **}**  **window.alert\_toast= function($msg = 'TEST',$bg = 'success'){**  **$('#alert\_toast').removeClass('bg-success')**  **$('#alert\_toast').removeClass('bg-danger')**  **$('#alert\_toast').removeClass('bg-info')**  **$('#alert\_toast').removeClass('bg-warning')**  **if($bg == 'success')**  **$('#alert\_toast').addClass('bg-success')**  **if($bg == 'danger')**  **$('#alert\_toast').addClass('bg-danger')**  **if($bg == 'info')**  **$('#alert\_toast').addClass('bg-info')**  **if($bg == 'warning')**  **$('#alert\_toast').addClass('bg-warning')**  **$('#alert\_toast .toast-body').html($msg)**  **$('#alert\_toast').toast({delay:3000}).toast('show');**  **}**  **window.load\_cart = function(){**  **$.ajax({**  **url:'admin/ajax.php?action=get\_cart\_count',**  **success:function(resp){**  **if(resp > -1){**  **resp = resp > 0 ? resp : 0;**  **$('.item\_count').html(resp)**  **}**  **}**  **})**  **}**  **$('#login\_now').click(function(){**  **uni\_modal("LOGIN",'login.php')**  **})**  **$(document).ready(function(){**  **load\_cart()**  **})**  **</script>**  **<!-- Bootstrap core JS-->**  **<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js"></script>**  **<!-- Third party plugin JS-->**  **<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing/1.4.1/jquery.easing.min.js"></script>**  **<script src="https://cdnjs.cloudflare.com/ajax/libs/magnific-popup.js/1.1.0/jquery.magnific-popup.min.js"></script>**  **<!-- Core theme JS-->**  **<script src="js/scripts.js"></script> </body>**    **</html>**    **<div class="modal fade" id="confirm\_modal" role='dialog'>**  **<div class="modal-dialog modal-md" role="document">**  **<div class="modal-content">**  **<div class="modal-header">**  **<h5 class="modal-title">Confirmation</h5>**  **</div>**  **<div class="modal-body">**  **<div id="delete\_content"></div>**  **</div>**  **<div class="modal-footer">**  **<button type="button" class="btn btn-primary" id='confirm' onclick="">Continue</button>**  **<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>**  **</div>**  **</div>**  **</div>**  **</div>**  **<div class="modal fade" id="uni\_modal" role='dialog'>**  **<div class="modal-dialog modal-md" role="document">**  **<div class="modal-content">**  **<div class="modal-header">**  **<h5 class="modal-title"></h5>**  **</div>**  **<div class="modal-body">**  **</div>**  **<div class="modal-footer">**  **<button type="button" class="btn btn-primary" id='submit' onclick="$('#uni\_modal form').submit()">Save</button>**  **<button type="button" class="btn btn-secondary" data-dismiss="modal">Cancel</button>**  **</div>**  **</div>**  **</div>**  **</div>**  **<div class="modal fade" id="uni\_modal\_right" role='dialog'>**  **<div class="modal-dialog modal-full-height modal-md" role="document">**  **<div class="modal-content">**  **<div class="modal-header">**  **<h5 class="modal-title"></h5>**  **<button type="button" class="close" data-dismiss="modal" aria-label="Close">**  **<span class="fa fa-arrow-righ t"></span>**  **</button>**  **</div>**  **<div class="modal-body">**  **</div>**  **</div>**  **</div>**  **</div>**  **<footer class="bg-light py-5">**  **<div class="container"><div class="small text-center text-muted"> <a href="" target="\_blank"></a></div></div>**  **</footer>**    **<script>**  **$('.datepicker').datepicker({**  **format:"yyyy-mm-dd"**  **})**  **window.start\_load = function(){**  **$('body').prepend('<di id="preloader2"></di>')**  **}**  **window.end\_load = function(){**  **$('#preloader2').fadeOut('fast', function() {**  **$(this).remove();**  **})**  **}**  **window.uni\_modal = function($title = '' , $url=''){**  **start\_load()**  **$.ajax({**  **url:$url,**  **error:err=>{**  **console.log()**  **alert("An error occured")**  **},**  **success:function(resp){**  **if(resp){**  **$('#uni\_modal .modal-title').html($title)**  **$('#uni\_modal .modal-body').html(resp)**  **$('#uni\_modal').modal('show')**  **end\_load()**  **}**  **}**  **})**  **}**  **window.uni\_modal\_right = function($title = '' , $url=''){**  **start\_load()**  **$.ajax({**  **url:$url,**  **error:err=>{**  **console.log()**  **alert("An error occured")**  **},**  **success:function(resp){**  **if(resp){**  **$('#uni\_modal\_right .modal-title').html($title)**  **$('#uni\_modal\_right .modal-body').html(resp)**  **$('#uni\_modal\_right').modal('show')**  **end\_load()**  **}**  **}**  **})**  **}**  **window.alert\_toast= function($msg = 'TEST',$bg = 'success'){**  **$('#alert\_toast').removeClass('bg-success')**  **$('#alert\_toast').removeClass('bg-danger')**  **$('#alert\_toast').removeClass('bg-info')**  **$('#alert\_toast').removeClass('bg-warning')**  **if($bg == 'success')**  **$('#alert\_toast').addClass('bg-success')**  **if($bg == 'danger')**  **$('#alert\_toast').addClass('bg-danger')**  **if($bg == 'info')**  **$('#alert\_toast').addClass('bg-info')**  **if($bg == 'warning')**  **$('#alert\_toast').addClass('bg-warning')**  **$('#alert\_toast .toast-body').html($msg)**  **$('#alert\_toast').toast({delay:3000}).toast('show');**  **}**  **window.load\_cart = function(){**  **$.ajax({**  **url:'admin/ajax.php?action=get\_cart\_count',**  **success:function(resp){**  **if(resp > -1){**  **resp = resp > 0 ? resp : 0;**  **$('.item\_count').html(resp)**  **}**  **}**  **})**  **}**  **$('#login\_now').click(function(){**  **uni\_modal("LOGIN",'login.php')**  **})**  **$(document).ready(function(){**  **load\_cart()**  **})**  **</script>**  **<!-- Bootstrap core JS-->**  **<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js"></script>**  **<!-- Third party plugin JS-->**  **<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing/1.4.1/jquery.easing.min.js"></script>**  **<script src="https://cdnjs.cloudflare.com/ajax/libs/magnific-popup.js/1.1.0/jquery.magnific-popup.min.js"></script>**  **<!-- Core theme JS-->**  **<script src="js/scripts.js"></script> </body>**    **</html>** |

**Home**

|  |
| --- |
| **<!-- Masthead-->**  **<header class="masthead">**  **<div class="container h-100">**  **<div class="row h-100 align-items-center justify-content-center text-center">**  **<div class="col-lg-10 align-self-end mb-4 page-title">**  **<h3 class="text-white">Welcome to <?php echo $\_SESSION['setting\_name']; ?></h3>**  **<hr class="divider my-4" />**  **<a class="btn btn-primary btn-xl js-scroll-trigger" href="#menu">Order Now</a>**  **</div>**    **</div>**  **</div>**  **</header>**  **<section class="page-section" id="menu">**  **<div id="menu-field" class="card-deck">**  **<?php**  **include'admin/db\_connect.php';**  **$qry = $conn->query("SELECT \* FROM product\_list order by rand() ");**  **while($row = $qry->fetch\_assoc()):**  **?>**  **<div class="col-lg-3">**  **<div class="card menu-item ">**  **<img src="assets/img/<?php echo $row['img\_path'] ?>" class="card-img-top" alt="...">**  **<div class="card-body">**  **<h5 class="card-title"><?php echo $row['name'] ?></h5>**  **<p class="card-text truncate"><?php echo $row['description'] ?></p>**  **<div class="text-center">**  **<button class="btn btn-sm btn-outline-primary view\_prod btn-block" data-id=<?php echo $row['id'] ?>><i class="fa fa-eye"></i> View</button>**    **</div>**  **</div>**    **</div>**  **</div>**  **<?php endwhile; ?>**  **</div>**  **</section>**  **<script>**    **$('.view\_prod').click(function(){**  **uni\_modal\_right('Product','view\_prod.php?id='+$(this).attr('data-id'))**  **})**  **</script>** |

**Login**

|  |
| --- |
| **<?php session\_start() ?>**  **<div class="container-fluid">**  **<form action="" id="login-frm">**  **<div class="form-group">**  **<label for="" class="control-label">Email</label>**  **<input type="email" name="email" required="" class="form-control">**  **</div>**  **<div class="form-group">**  **<label for="" class="control-label">Password</label>**  **<input type="password" name="password" required="" class="form-control">**  **<small><a href="javascript:void(0)" id="new\_account">Create New Account</a></small>**  **</div>**  **<button class="button btn btn-info btn-sm">Login</button>**  **</form>**  **</div>**  **<style>**  **#uni\_modal .modal-footer{**  **display:none;**  **}**  **</style>**  **<script>**  **$('#new\_account').click(function(){**  **uni\_modal("Create an Account",'signup.php?redirect=index.php?page=checkout')**  **})**  **$('#login-frm').submit(function(e){**  **e.preventDefault()**  **$('#login-frm button[type="submit"]').attr('disabled',true).html('Logging in...');**  **if($(this).find('.alert-danger').length > 0 )**  **$(this).find('.alert-danger').remove();**  **$.ajax({**  **url:'admin/ajax.php?action=login2',**  **method:'POST',**  **data:$(this).serialize(),**  **error:err=>{**  **console.log(err)**  **$('#login-frm button[type="submit"]').removeAttr('disabled').html('Login');**  **},**  **success:function(resp){**  **if(resp == 1){**  **location.href ='<?php echo isset($\_GET['redirect']) ? $\_GET['redirect'] : 'index.php?page=home' ?>';**  **}else{**  **$('#login-frm').prepend('<div class="alert alert-danger">Email or password is incorrect.</div>')**  **$('#login-frm button[type="submit"]').removeAttr('disabled').html('Login');**  **}**  **}**  **})**  **})**  **</script>** |

**Sign up**

|  |
| --- |
| **<?php session\_start() ?>**  **<div class="container-fluid">**  **<form action="" id="signup-frm">**  **<div class="form-group">**  **<label for="" class="control-label">Firstname</label>**  **<input type="text" name="first\_name" required="" class="form-control">**  **</div>**  **<div class="form-group">**  **<label for="" class="control-label">Email</label>**  **<input type="text" name="last\_name" required="" class="form-control">**  **</div>**  **<div class="form-group">**  **<label for="" class="control-label">Contact</label>**  **<input type="text" name="mobile" required="" class="form-control">**  **</div>**  **<div class="form-group">**  **<label for="" class="control-label">Address</label>**  **<textarea cols="30" rows="3" name="address" required="" class="form-control"></textarea>**  **</div>**  **<div class="form-group">**  **<label for="" class="control-label">Email</label>**  **<input type="email" name="email" required="" class="form-control">**  **</div>**  **<div class="form-group">**  **<label for="" class="control-label">Password</label>**  **<input type="password" name="password" required="" class="form-control">**  **</div>**  **<button class="button btn btn-info btn-sm">Create</button>**  **</form>**  **</div>**  **<style>**  **#uni\_modal .modal-footer{**  **display:none;**  **}**  **</style>**  **<script>**  **$('#signup-frm').submit(function(e){**  **e.preventDefault()**  **$('#signup-frm button[type="submit"]').attr('disabled',true).html('Saving...');**  **if($(this).find('.alert-danger').length > 0 )**  **$(this).find('.alert-danger').remove();**  **$.ajax({**  **url:'admin/ajax.php?action=signup',**  **method:'POST',**  **data:$(this).serialize(),**  **error:err=>{**  **console.log(err)**  **$('#signup-frm button[type="submit"]').removeAttr('disabled').html('Create');**  **},**  **success:function(resp){**  **if(resp == 1){**  **location.href ='<?php echo isset($\_GET['redirect']) ? $\_GET['redirect'] : 'index.php?page=home' ?>';**  **}else{**  **$('#signup-frm').prepend('<div class="alert alert-danger">Email already exist.</div>')**  **$('#signup-frm button[type="submit"]').removeAttr('disabled').html('Create');**  **}**  **}**  **})**  **})**  **</script>** |

**User cart**

|  |
| --- |
| **<!-- Masthead-->**  **<header class="masthead">**  **<div class="container h-100">**  **<div class="row h-100 align-items-center justify-content-center text-center">**  **<div class="col-lg-10 align-self-end mb-4 page-title">**  **<h3 class="text-white">Cart List</h3>**  **<hr class="divider my-4" />**  **</div>**    **</div>**  **</div>**  **</header>**  **<section class="page-section" id="menu">**  **<div class="container">**  **<div class="row">**  **<div class="col-lg-8">**  **<div class="sticky">**  **<div class="card">**  **<div class="card-body">**  **<div class="row">**  **<div class="col-md-8"><b>Card</b></div>**  **<div class="col-md-4 text-right"><b>Total</b></div>**  **</div>**  **</div>**  **</div>**  **</div>**  **<?php**  **if(isset($\_SESSION['login\_user\_id'])){**  **$data = "where c.user\_id = '".$\_SESSION['login\_user\_id']."' ";**  **}else{**  **$ip = isset($\_SERVER['HTTP\_CLIENT\_IP']) ? $\_SERVER['HTTP\_CLIENT\_IP'] : isset($\_SERVER['HTTP\_X\_FORWARDED\_FOR']) ? $\_SERVER['HTTP\_X\_FORWARDED\_FOR'] : $\_SERVER['REMOTE\_ADDR'];**  **$data = "where c.client\_ip = '".$ip."' ";**  **}**  **$total = 0;**  **$get = $conn->query("SELECT \*,c.id as cid FROM cart c inner join product\_list p on p.id = c.product\_id ".$data);**  **while($row= $get->fetch\_assoc()):**  **$total += ($row['qty'] \* $row['price']);**  **?>**  **<div class="card">**  **<div class="card-body">**  **<div class="row">**  **<div class="col-md-4" style="text-align: -webkit-center">**  **<a href="javascript:void(0)" class="rem\_cart btn btn-sm btn-outline-danger" data-id="<?php echo $row['cid'] ?>"><i class="fa fa-trash"></i></a>**  **<img src="assets/img/<?php echo $row['img\_path'] ?>" alt="">**  **</div>**  **<div class="col-md-4">**  **<p><b><large><?php echo $row['name'] ?></large></b></p>**  **<p class='truncate'> <b><small>Desc :<?php echo $row['description'] ?></small></b></p>**  **<p> <b><small>Unit Price :<?php echo number\_format($row['price'],2) ?></small></b></p>**  **<p><small>QTY :</small></p>**  **<div class="input-group mb-3">**  **<div class="input-group-prepend">**  **<button class="btn btn-outline-secondary qty-minus" type="button" data-id="<?php echo $row['cid'] ?>"><span class="fa fa-minus"></button>**  **</div>**  **<input type="number" readonly value="<?php echo $row['qty'] ?>" min = 1 class="form-control text-center" name="qty" >**  **<div class="input-group-prepend">**  **<button class="btn btn-outline-secondary qty-plus" type="button" id="" data-id="<?php echo $row['cid'] ?>"><span class="fa fa-plus"></span></button>**  **</div>**  **</div>**  **</div>**  **<div class="col-md-4 text-right">**  **<b><large><?php echo number\_format($row['qty'] \* $row['price'],2) ?></large></b>**  **</div>**  **</div>**  **</div>**  **</div>**  **<?php endwhile; ?>**  **</div>**  **<div class="col-md-4">**  **<div class="sticky">**  **<div class="card">**  **<div class="card-body">**  **<p><large>Total Amount</large></p>**  **<hr>**  **<p class="text-right"><b><?php echo number\_format($total,2) ?></b></p>**  **<hr>**  **<div class="text-center">**  **<button class="btn btn-block btn-outline-primary" type="button" id="checkout">Proceed to Checkout</button>**  **</div>**  **</div>**  **</div>**  **</div>**  **</div>**  **</div>**  **</div>**  **</section>**  **<style>**  **.card p {**  **margin: unset**  **}**  **.card img{**  **max-width: calc(100%);**  **max-height: calc(59%);**  **}**  **div.sticky {**  **position: -webkit-sticky; /\* Safari \*/**  **position: sticky;**  **top: 4.7em;**  **z-index: 10;**  **background: white**  **}**  **.rem\_cart{**  **position: absolute;**  **left: 0;**  **}**  **</style>**  **<script>**    **$('.view\_prod').click(function(){**  **uni\_modal\_right('Product','view\_prod.php?id='+$(this).attr('data-id'))**  **})**  **$('.qty-minus').click(function(){**  **var qty = $(this).parent().siblings('input[name="qty"]').val();**  **update\_qty(parseInt(qty) -1,$(this).attr('data-id'))**  **if(qty == 1){**  **return false;**  **}else{**  **$(this).parent().siblings('input[name="qty"]').val(parseInt(qty) -1);**  **}**  **})**  **$('.qty-plus').click(function(){**  **var qty = $(this).parent().siblings('input[name="qty"]').val();**  **$(this).parent().siblings('input[name="qty"]').val(parseInt(qty) +1);**  **update\_qty(parseInt(qty) +1,$(this).attr('data-id'))**  **})**  **function update\_qty(qty,id){**  **start\_load()**  **$.ajax({**  **url:'admin/ajax.php?action=update\_cart\_qty',**  **method:"POST",**  **data:{id:id,qty},**  **success:function(resp){**  **if(resp == 1){**  **load\_cart()**  **end\_load()**  **}**  **}**  **})**  **}**  **$('#checkout').click(function(){**  **if('<?php echo isset($\_SESSION['login\_user\_id']) ?>' == 1){**  **location.replace("index.php?page=checkout")**  **}else{**  **uni\_modal("Checkout","login.php?page=checkout")**  **}**  **})**  **</script>** |

**User checkout**

|  |
| --- |
| **<!-- Masthead-->**  **<header class="masthead">**  **<div class="container h-100">**  **<div class="row h-100 align-items-center justify-content-center text-center">**  **<div class="col-lg-10 align-self-end mb-4 page-title">**  **<h3 class="text-white">Cart List</h3>**  **<hr class="divider my-4" />**  **</div>**    **</div>**  **</div>**  **</header>**  **<section class="page-section" id="menu">**  **<div class="container">**  **<div class="row">**  **<div class="col-lg-8">**  **<div class="sticky">**  **<div class="card">**  **<div class="card-body">**  **<div class="row">**  **<div class="col-md-8"><b>Card</b></div>**  **<div class="col-md-4 text-right"><b>Total</b></div>**  **</div>**  **</div>**  **</div>**  **</div>**  **<?php**  **if(isset($\_SESSION['login\_user\_id'])){**  **$data = "where c.user\_id = '".$\_SESSION['login\_user\_id']."' ";**  **}else{**  **$ip = isset($\_SERVER['HTTP\_CLIENT\_IP']) ? $\_SERVER['HTTP\_CLIENT\_IP'] : isset($\_SERVER['HTTP\_X\_FORWARDED\_FOR']) ? $\_SERVER['HTTP\_X\_FORWARDED\_FOR'] : $\_SERVER['REMOTE\_ADDR'];**  **$data = "where c.client\_ip = '".$ip."' ";**  **}**  **$total = 0;**  **$get = $conn->query("SELECT \*,c.id as cid FROM cart c inner join product\_list p on p.id = c.product\_id ".$data);**  **while($row= $get->fetch\_assoc()):**  **$total += ($row['qty'] \* $row['price']);**  **?>**  **<div class="card">**  **<div class="card-body">**  **<div class="row">**  **<div class="col-md-4" style="text-align: -webkit-center">**  **<a href="javascript:void(0)" class="rem\_cart btn btn-sm btn-outline-danger" data-id="<?php echo $row['cid'] ?>"><i class="fa fa-trash"></i></a>**  **<img src="assets/img/<?php echo $row['img\_path'] ?>" alt="">**  **</div>**  **<div class="col-md-4">**  **<p><b><large><?php echo $row['name'] ?></large></b></p>**  **<p class='truncate'> <b><small>Desc :<?php echo $row['description'] ?></small></b></p>**  **<p> <b><small>Unit Price :<?php echo number\_format($row['price'],2) ?></small></b></p>**  **<p><small>QTY :</small></p>**  **<div class="input-group mb-3">**  **<div class="input-group-prepend">**  **<button class="btn btn-outline-secondary qty-minus" type="button" data-id="<?php echo $row['cid'] ?>"><span class="fa fa-minus"></button>**  **</div>**  **<input type="number" readonly value="<?php echo $row['qty'] ?>" min = 1 class="form-control text-center" name="qty" >**  **<div class="input-group-prepend">**  **<button class="btn btn-outline-secondary qty-plus" type="button" id="" data-id="<?php echo $row['cid'] ?>"><span class="fa fa-plus"></span></button>**  **</div>**  **</div>**  **</div>**  **<div class="col-md-4 text-right">**  **<b><large><?php echo number\_format($row['qty'] \* $row['price'],2) ?></large></b>**  **</div>**  **</div>**  **</div>**  **</div>**  **<?php endwhile; ?>**  **</div>**  **<div class="col-md-4">**  **<div class="sticky">**  **<div class="card">**  **<div class="card-body">**  **<p><large>Total Amount</large></p>**  **<hr>**  **<p class="text-right"><b><?php echo number\_format($total,2) ?></b></p>**  **<hr>**  **<div class="text-center">**  **<button class="btn btn-block btn-outline-primary" type="button" id="checkout">Proceed to Checkout</button>**  **</div>**  **</div>**  **</div>**  **</div>**  **</div>**  **</div>**  **</div>**  **</section>**  **<style>**  **.card p {**  **margin: unset**  **}**  **.card img{**  **max-width: calc(100%);**  **max-height: calc(59%);**  **}**  **div.sticky {**  **position: -webkit-sticky; /\* Safari \*/**  **position: sticky;**  **top: 4.7em;**  **z-index: 10;**  **background: white**  **}**  **.rem\_cart{**  **position: absolute;**  **left: 0;**  **}**  **</style>**  **<script>**    **$('.view\_prod').click(function(){**  **uni\_modal\_right('Product','view\_prod.php?id='+$(this).attr('data-id'))**  **})**  **$('.qty-minus').click(function(){**  **var qty = $(this).parent().siblings('input[name="qty"]').val();**  **update\_qty(parseInt(qty) -1,$(this).attr('data-id'))**  **if(qty == 1){**  **return false;**  **}else{**  **$(this).parent().siblings('input[name="qty"]').val(parseInt(qty) -1);**  **}**  **})**  **$('.qty-plus').click(function(){**  **var qty = $(this).parent().siblings('input[name="qty"]').val();**  **$(this).parent().siblings('input[name="qty"]').val(parseInt(qty) +1);**  **update\_qty(parseInt(qty) +1,$(this).attr('data-id'))**  **})**  **function update\_qty(qty,id){**  **start\_load()**  **$.ajax({**  **url:'admin/ajax.php?action=update\_cart\_qty',**  **method:"POST",**  **data:{id:id,qty},**  **success:function(resp){**  **if(resp == 1){**  **load\_cart()**  **end\_load()**  **}**  **}**  **})**  **}**  **$('#checkout').click(function(){**  **if('<?php echo isset($\_SESSION['login\_user\_id']) ?>' == 1){**  **location.replace("index.php?page=checkout")**  **}else{**  **uni\_modal("Checkout","login.php?page=checkout")**  **}**  **})**  **</script>** |

**Admin login**

|  |
| --- |
| **<!DOCTYPE html>**  **<html lang="en">**  **<head>**  **<meta charset="utf-8">**  **<meta content="width=device-width, initial-scale=1.0" name="viewport">**  **<title>Admin | Online Food Ordering System</title>**    **<?php include('./header.php'); ?>**  **<?php include('./db\_connect.php'); ?>**  **<?php**  **session\_start();**  **if(isset($\_SESSION['login\_id']))**  **header("location:index.php?page=home");**  **$query = $conn->query("SELECT \* FROM system\_settings limit 1")->fetch\_array();**  **foreach ($query as $key => $value) {**  **if(!is\_numeric($key))**  **$\_SESSION['setting\_'.$key] = $value;**  **}**  **?>**  **</head>**  **<style>**  **body{**  **width: 100%;**  **height: calc(100%);**  **/\*background: #007bff;\*/**  **}**  **main#main{**  **width:100%;**  **height: calc(100%);**  **background:white;**  **}**  **#login-right{**  **position: absolute;**  **right:0;**  **width:40%;**  **height: calc(100%);**  **background:white;**  **display: flex;**  **align-items: center;**  **}**  **#login-left{**  **position: absolute;**  **left:0;**  **width:60%;**  **height: calc(100%);**  **background:#00000061;**  **display: flex;**  **align-items: center;**  **}**  **#login-right .card{**  **margin: auto**  **}**  **.logo {**  **margin: auto;**  **font-size: 8rem;**  **background: white;**  **border-radius: 50% 50%;**  **height: 29vh;**  **width: 13vw;**  **display: flex;**  **align-items: center;**  **}**  **.logo img{**  **height: 80%;**  **width: 80%;**  **margin: auto**  **}**  **</style>**  **<body>**  **<main id="main" class=" bg-dark">**  **<div id="login-left">**  **<div class="logo">**  **<img src="../assets/img/sample\_logo.png" alt="">**  **</div>**  **</div>**  **<div id="login-right">**  **<div class="card col-md-8">**  **<div class="card-body">**  **<form id="login-form" >**  **<div class="form-group">**  **<label for="username" class="control-label">Username</label>**  **<input type="text" id="username" name="username" class="form-control">**  **</div>**  **<div class="form-group">**  **<label for="password" class="control-label">Password</label>**  **<input type="password" id="password" name="password" class="form-control">**  **</div>**  **<center><button class="btn-sm btn-block btn-wave col-md-4 btn-primary">Login</button></center>**  **</form>**  **</div>**  **</div>**  **</div>**    **</main>**  **<a href="#" class="back-to-top"><i class="icofont-simple-up"></i></a>**  **</body>**  **<script>**  **$('#login-form').submit(function(e){**  **e.preventDefault()**  **$('#login-form button[type="button"]').attr('disabled',true).html('Logging in...');**  **if($(this).find('.alert-danger').length > 0 )**  **$(this).find('.alert-danger').remove();**  **$.ajax({**  **url:'ajax.php?action=login',**  **method:'POST',**  **data:$(this).serialize(),**  **error:err=>{**  **console.log(err)**  **$('#login-form button[type="button"]').removeAttr('disabled').html('Login');**  **},**  **success:function(resp){**  **if(resp == 1){**  **location.href ='index.php?page=home';**  **}else if(resp == 2){**  **location.href ='voting.php';**  **}else{**  **$('#login-form').prepend('<div class="alert alert-danger">Username or password is incorrect.</div>')**  **$('#login-form button[type="button"]').removeAttr('disabled').html('Login');**  **}**  **}**  **})**  **})**  **</script>**  **</html>** |

**Admin order**

|  |
| --- |
| **<div class="container-fluid">**  **<div class="card">**  **<div class="card-body">**  **<table class="table table-bordered">**  **<thead>**  **<tr>**  **<th>#</th>**  **<th>Name</th>**  **<th>Address</th>**  **<th>Email</th>**  **<th>Mobile</th>**  **<th>Status</th>**  **<th></th>**  **</tr>**  **</thead>**  **<tbody>**  **<?php**  **$i = 1;**  **include 'db\_connect.php';**  **$qry = $conn->query("SELECT \* FROM orders ");**  **while($row=$qry->fetch\_assoc()):**  **?>**  **<tr>**  **<td><?php echo $i++ ?></td>**  **<td><?php echo $row['name'] ?></td>**  **<td><?php echo $row['address'] ?></td>**  **<td><?php echo $row['email'] ?></td>**  **<td><?php echo $row['mobile'] ?></td>**  **<?php if($row['status'] == 1): ?>**  **<td class="text-center"><span class="badge badge-success">Confirmed</span></td>**  **<?php else: ?>**  **<td class="text-center"><span class="badge badge-secondary">For Verification</span></td>**  **<?php endif; ?>**  **<td>**  **<button class="btn btn-sm btn-primary view\_order" data-id="<?php echo $row['id'] ?>" >View Order</button>**  **</td>**  **</tr>**  **<?php endwhile; ?>**  **</tbody>**  **</table>**  **</div>**  **</div>**    **</div>**  **<script>**  **$('.view\_order').click(function(){**  **uni\_modal('Order','view\_order.php?id='+$(this).attr('data-id'))**  **})**  **</script>** |

**Conclusions 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. Finality, the chapter ends by concluding remarks.

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.

• Objective #2 was addressed by utilising Extreme Programming method of Agile Development. Along with this, sdlc tools are used to analyse and design the system.

• Objective #3 was satisfied by developing the system with sql and Bootstrap 4.

• Objective #4 was satisfied by integrating database to the system.

• Objective #5 was addressed with various testing approaches to ensure the prototype system is as robust as possible.

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.

**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 minimised 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.

**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:

• Secured payment system with various payment methods.

• Adding support for food order delivery tracking.

• Managing customer loyalty membership and discount voucher.

• Converting the system to more user friendly

**Bioliography :**

Korth, H. F., Sudarshan, S. and Silberschatz, A. (2010) Database System Concepts.

McGraw-Hill Education. Kothari, C. R. (2004) Research Methodology: Methods and Techniques.

New Age International. Lambert, M. (2016) Learning bootstrap 4. Birmingham, UK: Packt Publishing Ltd.

Korth, H. F., Sudarshan, S. and Silberschatz, A. (2010) *Database System Concepts*. McGraw-Hill Education.

Kothari, C. R. (2004) *Research Methodology: Methods and Techniques*. New Age International.

Lambert, M. (2016) *Learning bootstrap 4*. Birmingham, UK: Packt Publishing Ltd.

Lei, K., Ma, Y. and Tan, Z. (2014) ‘Performance Comparison and Evaluation of Web Development Technologies in PHP, Python, and Node.js’, in. *2014 IEEE 17th International Conference on Computational Science and Engineering*, pp. 661–668. doi: 10.1109/CSE.2014.142.

Leong, W. H. (2016) *Food Ordering System Using Mobile Phone*. University Tunku Abdul Rahman. Available at: <http://eprints.utar.edu.my/1943/1/IA-2016-1203135-> 1.pdf (Accessed: 12 January 2018).

McFarlin, T. (2017) ‘WordPress Templates for Beginners: What’s Templating, Anyway?’, *Tom McFarlin*, 8 November. Available at: https://tommcfarlin.com/wordpress-templates-3/ (Accessed: 26 January 2018).

Ngai, E. W. T., Suk, F. F. C. and Lo, S. Y. Y. (2008) ‘Development of an RFID-based sushi management system: The case of a conveyor-belt sushi restaurant’, *International Journal of Production Economics*. (Special Section on RFID: Technology, Applications, and Impact on Business Operations), 112(2), pp. 630–645. doi: 10.1016/j.ijpe.2007.05.011.

Oddle (2016) *2016 Restaurant Industry Statistics That Made The Industry Tick*, *Oddle*. Available at: https://oddle.me/blog/2016/12/29/2016-restaurant-industry- statistics-that-made-the-industry-tick (Accessed: 7 July 2017).

Olanrewaju, R. F., Islam, T. and Ali, N. (2015) ‘An empirical study of the evolution of php mvc framework’, in *Advanced Computer and Communication Engineering Technology*. Springer, pp. 399–410.

Otwell, T. (2018a) *Blade Templates*, *Laravel Documentation*. Available at: https://laravel.com/docs/5.5/blade (Accessed: 26 January 2018).

Otwell, T. (2018b) *Eloquent*, *Laravel Documentation*. Available at: https://laravel.com/docs/5.4/eloquent (Accessed: 22 January 2018).

Otwell, T. (2018c) *Hashing*, *Laravel Documentation*. Available at: https://laravel.com/docs/5.4/hashing (Accessed: 2 January 2018).

Parr, T. J. (2004) ‘Enforcing strict model-view separation in template engines’, in *Proceedings of the 13th international conference on World Wide Web*. ACM, pp. 224–233.

Patel, K. J., Patel, U. and Obersnel, A. (2007) ‘PDA-based wireless food ordering system for hospitality industry; A case atudy of Box Hill Institute’, in. *2007 Wireless Telecommunications Symposium*, pp. 1–8. doi: 10.1109/WTS.2007.4563314.

Patel, M. (2015) *Online Food Order System for Restaurants*. Master’s thesis. Grand Valley State University. Available at: https://scholarworks.gvsu.edu/cistechlib/219/ (Accessed: 27 January 2018).

Saeed, H. *et al.* (2016) ‘Near-field communication sensors and cloud-based smart restaurant management system’, in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 686–691. doi: 10.1109/WF-IoT.2016.7845440.

Samsudin, N. A. *et al.* (2011) ‘A customizable wireless food ordering system with realtime customer feedback’, in *Wireless Technology and Applications (ISWTA), 2011 IEEE Symposium on*. IEEE, pp. 186–191.

Shankararaman, V. and Lum, E. K. (2013) ‘Integration of Social Media Technologies with ERP:A Prototype Implementation’, *AMCIS 2013 PROCEEDINGS*, p. 11.

Shinde, R. *et al.* (2014) ‘Design and Implementation of Digital dining in Restaurants using Android’, *International Journal of Advance Research in Computer Science and Management Studies*, 2(1).

Smith, A. and Wempen, F. (2011) *CompTIA Strata Study Guide Authorized Courseware: Exams FC0-U41, FC0-U11, and FC0-U21*. John Wiley & Sons.

Statista (2017) *Global social media ranking 2017*, *Statista*. Available at: htt[ps://www.statista.com/statisti](http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-)c[s/272014/global-social-networks-ranked-by-number-](http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-) of-users/ (Accessed: 2 January 2018).

Sullivan, K. (2015) *The History of Restaurant POS Systems*, *Toast*. Available at: https://pos.toasttab.com/blog/the-history-of-restaurant-pos-systems (Accessed: 22

October 2017).

Surguy, M. (2014) *Laravel: My First Framework*. Leanpub. Available at: <http://www.furqanfreed.com/wp-content/uploads/2016/05/laravel-first-framework.pdf> (Accessed: 27 December 2017).

Tan, C. L. (2013) *Implementing a Web-Based Computerized Restaurant System*. Master’s thesis. University of Manchester. Available at: