**Online Food Ordering System**

**A PROJECT REPORT**

**Submitted By**

**Atik Ahmad – 21002901400**

**Harshit Kaushik – 2100290140068**

**Mohd Wasim - 2100290140085**

**Submitted in partial fulfilment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Ms. Vidhushi**

**Associate Professor**



**Submitted to**

**CERTIFICATE**

Certified that **Atik Ahmed -2100290140041. Mohd Wasim-2100290140085. Harshit Kaushik-2100290140068** have carried out the project work having “**Online Food Ordering System**.” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**Atik Ahmed 2100290140041**

**Harshit Kaushik 2100290140068**

**Mohd Wasim 2100290140085**

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

Date:

**Dr. Vidushi**

**Associate Professor**

**Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

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

**Dr. Arun Tripathi**

**Head, Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

# ABSTRACT

The Online Food Ordering System described in this document has been designed to fill a specific niche in the market by providing small restaurants with the ability to offer their customers an online ordering option without having to invest large amounts of time and money in having custom software designed specifically for them. The system, which is highly customizable, allows the restaurant employees to easily manage the site content, most importantly the menu, themselves through a very intuitive graphical interface.

The website, which is the only component seen by the restaurant customers, is then built dynamically based on the current state of the system, so any changes made are reflected in real time. Visitors to the site, once registered, are then able to easily navigate this menu, add food items to their order, and specify delivery options with only a few clicks, greatly simplifying the ordering process. Back in the restaurant, placed orders are promptly retrieved and displayed in an easily readable format for efficient processing.

The purpose of this document is to provide in-depth descriptions of design and implementation details of the system, as well as descriptions of all available functionality and plans for evolution. In addition, user manuals and trouble-shooting tips have been included for all three components to give the reader a clear idea of intended typical use cases for the system.

# ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Vidushi** for his guidance, help, and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Atik Ahmed 2100290140041**

**Harshit Kaushik 2100290140068**

**Mohd Wasim 2100290140085**

|  |
| --- |
| Contents  [CERTIFICATE 2](#_Toc118291471)  [ABSTRACT 3](#_Toc118291475)  [ACKNOWLEDGEMENT 4](#_Toc118291476)  [Chapter 1: 5](#_Toc118291477)  [INTRODUCTION 7](#_Toc118291478)  [[Chapter 2: Requirements Specification 8](#_Toc118291480)](#_Toc279002576)  [[System Model 8](#_Toc118291480)](#_Toc279002577)  [[Functional Requirements 8](#_Toc118291480)](#_Toc279002578)  [[The Web Ordering System 9](#_Toc118291480)](#_Toc279002579)  [[Menu Management System 10](#_Toc118291480)](#_Toc279002580)  [[Order Retrieval System 11](#_Toc118291480)](#_Toc279002581)  [[User Interface Specifications 11](#_Toc118291480)](#_Toc279002582)  [[Web Ordering System 11](#_Toc118291480)](#_Toc279002583)  [[Menu Management System 12](#_Toc118291480)](#_Toc279002584)  [[Order Retrieval System 12](#_Toc118291480)](#_Toc279002585)  [[Non-functional Requirements 12](#_Toc118291480)](#_Toc279002586)  [[System Evolution 3](#_Toc118291480)](#_Toc279002587) [8](#_Toc118291480)  [CHAPTER 3 9](#_Toc118291481)  [3.1 GENERAL DESCRIPTION 9](#_Toc118291482)  [3.2 PROBLEM STATEMENT 9](#_Toc118291483)  [3.3 SYSTEM OBJECTIVES 10](#_Toc118291484)  [3.4 Requirement Specification 10](#_Toc118291485)  [3.5 Functional Requirements 10](#_Toc118291486)  [Register and Login 11](#_Toc118291487)  [3.6 Non-functional Requirements Performance Requirements 11](#_Toc118291488)  [3.5 SOFTWARE AND HARDWARE REQUIREMENTS 11](#_Toc118291489)  [SOFTWARE REQUIREMENTS 11](#_Toc118291490)  [HARDWARE REQUIREMENTS 12](#_Toc118291491)  [CHAPTER 4 ANALYSIS 16](#_Toc118291499)  [4.1 Feasibility study 16](#_Toc118291500)  [o Economical Feasibility 16](#_Toc118291501)  [o Technical Feasibility 16](#_Toc118291502)  [o Operational Feasibility 17](#_Toc118291503)  [CHAPTER 5 18](#_Toc118291504)  [Gantt chart 18](#_Toc118291505)  [5.2 DATA FLOW DIAGRAM 19](#_Toc118291506)  [5.3 CLASS DIAGRAM 22](#_Toc118291507)  [5.5 ENTITY RELATIONSHIP DIAGRAM 23](#_Toc118291508)  [5.4 USE CASE DIAGRAM 25](#_Toc118291509)  [5.5 ACTIVITY DIAGRAM 27](#_Toc118291510)  [5.6 SEQUENCE DIAGRAM 28](#_Toc118291511)  [5.7 INPUT OUTPUT DIAGRAM 29](#_Toc118291512)  CHAPTER 6…………………………………………………………………………………………...30  CODING AND IMPLEMENTATION……………………………………………………………………………….…30  CONCLUSION………………………………………………………………………………………...49 |
|  |

**CHAPTER 1**

## INTRODUCTION

**1.1 PROJECT DESCRIPTION**

In today’s age of fast food and take-out, many restaurants have chosen to focus on quick preparation and speedy delivery of orders rather than offering a rich dining experience. Until very recently, all of these delivery orders were placed over the phone, but there are many disadvantages to this system. First, the customer must have a physical copy of the restaurant’s menu to look at while placing their order and this menu must be up to date. While this expectation is not unreasonable, it is certainly inconvenient.

Second, the orders are placed using strictly oral communication, which makes it far more difficult for the customer to receive immediate feedback on the order they have placed. This often leads to confusion and incorrect orders. The current system is also inconvenient for the restaurant itself, as they must either have a dedicated staff member to answer the phone and take orders, or some employees must perform double-duty, distracting them from their regular tasks.

What I propose is an online ordering system, originally designed for use in college cafeterias, but 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. When the customer visits the ordering webpage, they are presented with an interactive and up-to-date menu, complete with all available options and dynamically adjusting prices based on the selected options. After making a selection, the item is then added to their order, which the customer can review the details of at any time before checking out. This provides instant visual confirmation of what was selected and ensures that items in the order are, in fact, what was intended.

# CHAPTER 2

## PROJECT REQUIREMENTS

**2.1 TECHNOLOGIES USED**

**Operating Systems**: Microsoft Windows Front End: React, HTML/CSS, Javascript UI/UX: Figma

**Software Requirements**:

* Windows 10/11 or equivalent
* React
* Spring Boot
* RDBMS (Back end): MySQL

**Hardware Requirements***:*

* Processor – Intel i3 5th generation or higher
* RAM – Minimum 4 GB, recommended 8 GB
* Disk space - Minimum 10 GB of free disk space
* Network Connectivity

## CHAPTER 3

**SOFTWARE REQUIREMENT SPECIFICATION**

## System Model

The structure of the system can be divided into three main logical components. The first component must provide some form of menu management, allowing the restaurant to control what can be ordered by customers. The second component is the web ordering system and provides the functionality for customers to place their order and supply all necessary details. The third and final logical component is the order retrieval system. Used by the restaurant to keep track of all orders which have been placed, this component takes care of retrieving and displaying order information, as well as updating orders which have already been processed.



## Functional Requirements:

As can be seen in the system model diagramed above, each of the three system components essentially provides a layer of isolation between the end user and the database. The motivation behind this isolation is twofold. Firstly, allowing the end user to interact with the system through a rich interface provide a much more enjoyable user experience, particularly for the non-technical users which will account for the majority of the system’s users. In addition, this isolation layer also protects the integrity of the database by preventing users from taking any action outside those which the system is designed to handle. Because of this design pattern, it is essential to enumerate exactly which functions a user will be presented and these functions are outlined below, grouped by component.

### The Web Ordering System:

Users of the web ordering 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.
* Customize options for a selected item.
* Add an item to their current order.
* Review their current order.
* Remove an item/remove all items from their current order.
* Provide delivery and payment details.
* Place an order.
* Receive confirmation in the form of an order number.

As the goal of the system is to make the process of placing an order as simple as possible for the customer, the functionality provided through the web ordering system is restricted to that which most pertinent to accomplish the desired task. All of the functions outlined above, with the exceptions of account creation and management, will be used every time a customer places an order. By not including extraneous functions, I am moving towards my goal of simplifying the ordering process.

### Menu Management System:

The menu management system will be available only to restaurant employees and will, as the name suggests, allow them to manage the menu that is displayed to users of the web ordering system. The functions afforded by the menu management system provide user with the ability to, using a graphical interface:

* Add a new/update/delete vendor to/from the menu.
* Add a new/update/delete food category to/from the menu.
* Add a new/update/delete food item to/from the menu.
* Add a new/update/delete option for a given food item.
* Update price for a given food item.
* Update default options for a given food item.
* Update additional information (description, photo, etc.) for a given food item.

It is anticipated that the functionality provided by this component will be one of the first things noted by the restaurant user, as they will have to go through it to configure their menu, etc. before beginning to actually take orders. Once everything is initially configured, however, this component will likely be the least used, as menu updates generally do not occur with great frequency.

### Order Retrieval System:

Of the three components, the order retrieval system is functionally the simplest. Like the menu management system, 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.
* Mark an order as having been processed and remove it from the list of active orders.

## User Interface Specifications:

Each of the system components will have their own unique interface. These are described below.

### Web Ordering System:

Users of the web ordering system will interact with the application through a series of simple forms. Each category of food has its own form associated with it which presents a drop down menu for choosing which specific item from the category should be added to the order, and a series of check boxes and radio buttons for selecting which options are to be included. Adding an item to the order is accomplished by a single button click. Users select which category of food they would like to order, and therefore which form should be displayed, by navigating a menu bar, an approach which should be familiar to most users.

Entering delivery and payment deals is done in a similar manner. The user is presented with a form and must complete the required fields, which include both drop down and text boxes, before checking out and receiving a confirmation number. One thing worth noting here is that whenever possible drop down boxes and buttons were used over freeform input in order to both simplify the ordering process and reduce the possibility of and SQL injection attempt.

### Menu Management System:

User interaction with the menu management system is similar to that with the web ordering system. Users navigate a tree structure to find the vendor, category, or specific food item that they would like to modify and after making their selection they are presented with a form which displays all of the current fields and values associated with that item, all of which can be modified or removed. The form also presents buttons which allow the addition of new fields and values. Unlike the web ordering system, however, most of the input here will be freeform, specifically in the form of text boxes, since there is no finite set of fields which could be added. This does not raise a major concern though, as input sanitation will be performed, and the user, who is assumed to be a restaurant employee, is less likely to be malicious than a web user.

### Order Retrieval System:

User interaction with the order retrieval will be very simple. The application will automatically fetch new orders from the database at regular intervals and display the order numbers, along with delivery time, in a panel on the left hand side of the application. To view the details of an order, the user must simply click on that order number, which will populate the right-hand panel with the details, displayed in an easy to read and navigate tree structure. This structure can intuitively be expanded and collapsed to display only the desired information. Finally, once and order is processed, the user clicks a single button, labeled “Processed”, to remove it from the list of active orders.

## Non-functional Requirements:

Because the design patterns of the Online Ordering System are pretty much the standard for a web application, the non-functional requirements of the system are very straightforward. Although written using Google Web Toolkit, the application is cross-compiled to HTML and JavaScript, along with a PHP backend, all of which are supported by any reasonably well maintained web server, although I would recommend Apache2, and particularly the free XAMPP distribution.

All of the application data is stored in a SQL database, and therefore a SQL server must also be installed on the host computer. As with Apache2, 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 restaurant that is not expecting to see much web traffic, or possibly doing only a limited test run, 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.

## System Evolution:

As mentioned in the system model, at the heart of the entire ordering system is the database. In fact, the system could be completely operational using nothing but the database and an appropriate shell utility, assuming that all users are well-versed in SQL and enjoy using it to order food. While this would be a bit extreme, it does illustrate the point that the one part of the system which will stay relatively constant is the database. On the other hand, it is very probable that the other components will continue to evolve with time. For example, with the booming popularity of mobile applications, I would really like to make the web interface available as a phone application as well. Also it may make sense to at some point migrate the menu management and order retrieval systems to web, or even mobile, applications as well, as some users may prefer to use them as such.

I am also certain that if this system goes into actual use, many requests will arise for additional features which I had not previously considered, but would be useful to have. For this reason, I feel as though the application can be constantly evolving, which I consider a very good thing.

# CHAPTER 4 ANALYSIS

## Feasibility study

After studying and analyzing all the existing and requires functionalities of the system, the next task is to do the feasibility study for the project. Feasibility study includes consideration of all the possible ways to provide a solution to a given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

## Economical Feasibility

For the economic feasibility, Economic analysis or cost/benefits analysis is most frequently used technique the effectiveness of a proposed system. it is a procedure to determine the benefits and saving those are expected from the proposes system and compare them with cost .if the benefits outweigh the costs, a decision is taken to design and implement the system. otherwise, further justification or alternative in proposed system will have to be made if it is to have a chance of being approved this is ongoing effort that improves in accuracy at each phase of a system life cycle

## Technical Feasibility

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionalities to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of front end and back end platform.

## Operational Feasibility

No doubt the technically growing world needs more enhancement in technology, this apps is very user friendly and all inputs to be taken all self-explanatory even to a layman. As far our study is concerned, the clients will be comfortable and happy as the system has cut down their loads and bring the young generation to the same virtual world they are growing drastically.

Operational feasibility cover two aspects.one technical performance aspects and the other is acceptance within the organization.

Operation feasibility determine how the proposed the system will fit in with the current operation and what needs to implement the system.

# CHAPTER 5

* 1. **Planning and scheduling**

## Gantt chart

A Gantt chart can be developed for the entire project or a separate chart can be developed for each function. A tabular form is maintained where rows indicate the task with milestones and columns indicate duration (Weeks).



Fig. 2: Gantt Chart

**Software Requirements with specifications:**

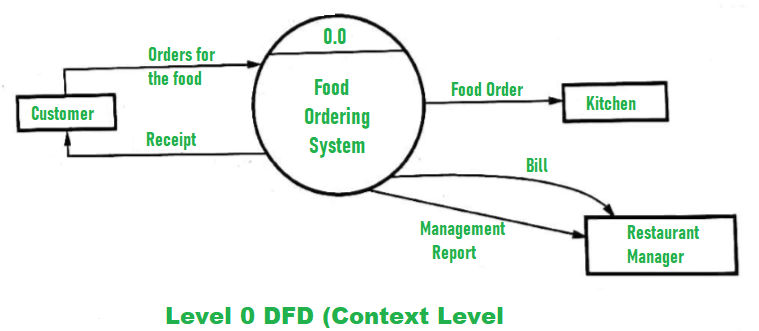
|  |  |
| --- | --- |
| Name of Components | Specifications |
| Operating system | Windows |
| Language | HTML/CSS,Javascript,PHP,SQL |
| Software Development kit | Google Chrome |
| Markup Language Enable | HTML |

**Hardware Requirements with specifications:**

|  |  |
| --- | --- |
| Name of Components | Specifications |
| Desktop/Laptop | Any Configuration |
|  |  |
| Memory Used | 6.31 MB |
|  |  |

## DATA FLOW DIAGRAM

t shows the flow of data among the sub module in it Admin data flow on the sub screen.



User Performance System this system shows the flow of data in User Modules on many Action. It shows the flow of data among the sub module in it User data flow on the sub screen. It is with who is someone’s expense manager.

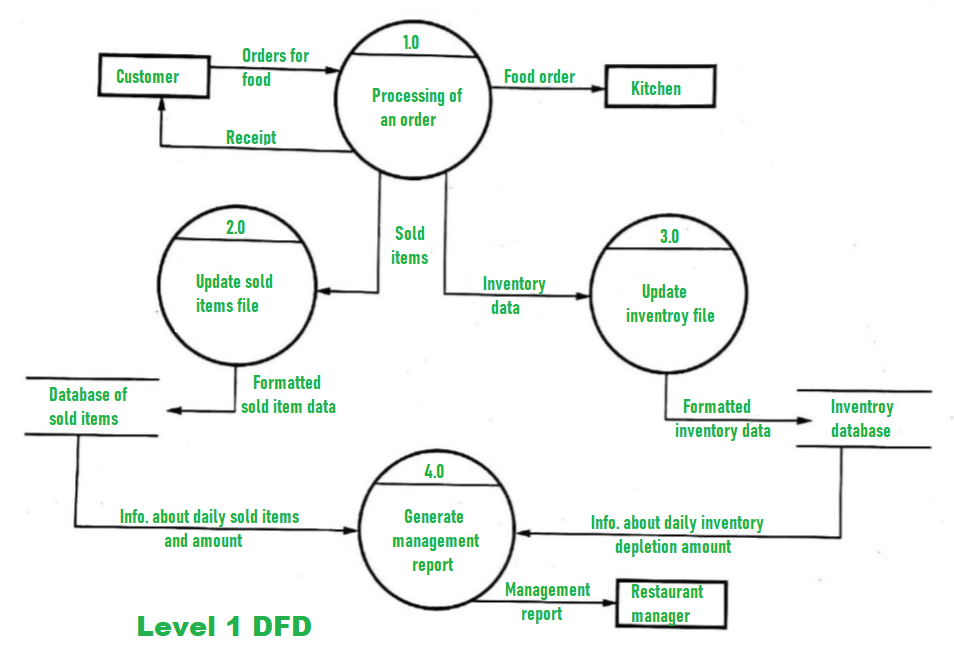


Fig. 4: User DFD Level 1

User Performance System this system shows the flow of data in User Modules on many Action. It shows the flow of data among the sub module in it User data flow on the sub screen. It is with who is not someone’s expense manager.

## CLASS DIAGRAM

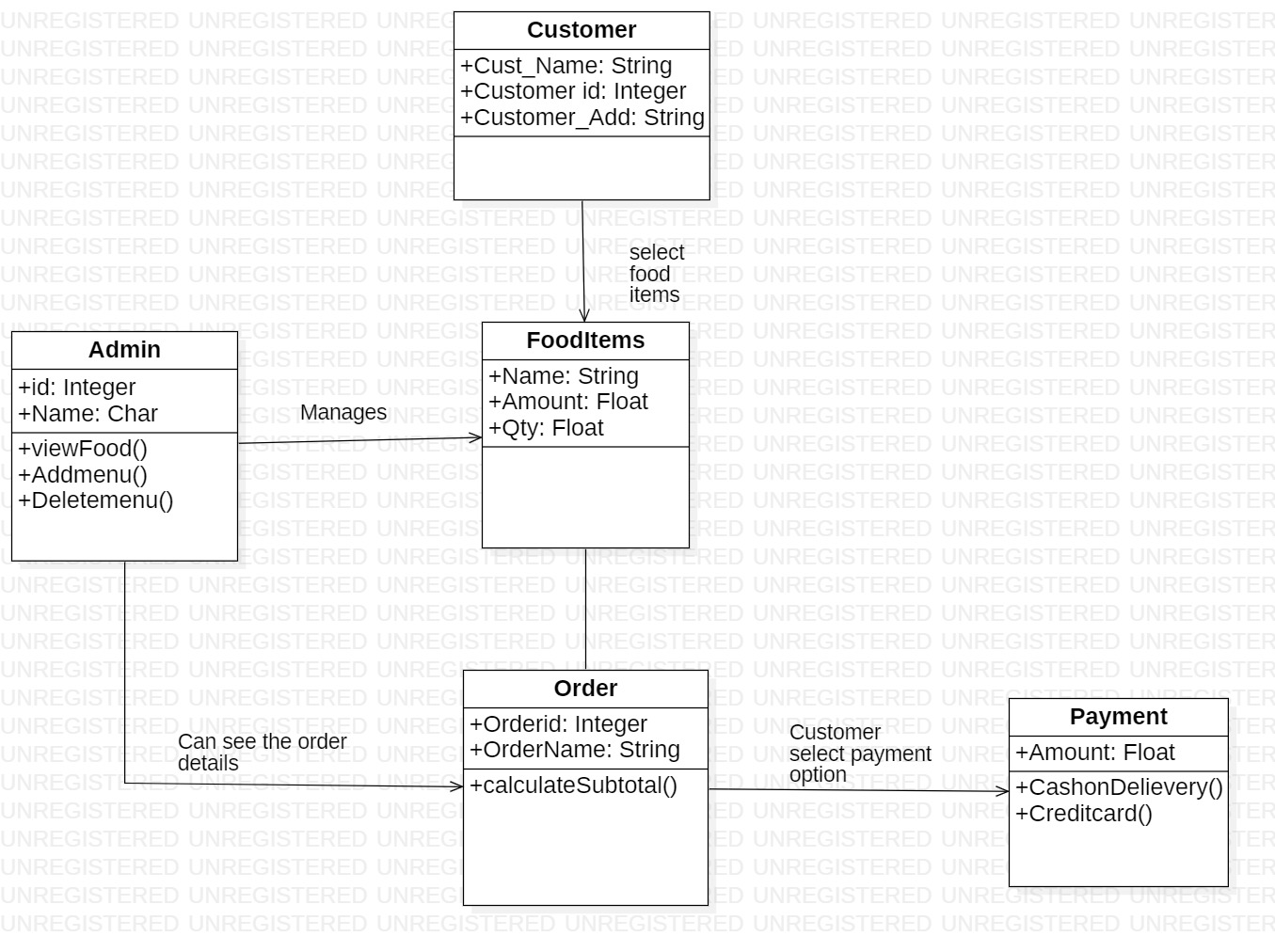
****

Fig. 6: Class Diagram

# ENTITY RELATIONSHIP DIAGRAM

This ER Diagram represents the model of WorkPlace CoAction System Entity. The Entity Relationship Diagram show all visual instrument of Database table and relation between HomePage, Admin Page, User Page. All of it have Structured data and every entity may have some attributes.

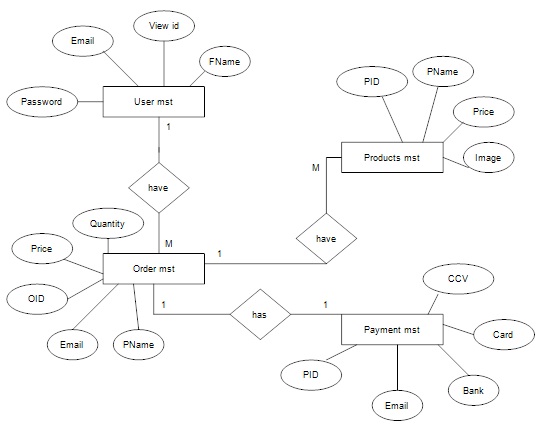


Fig. 5: Entity relationship Diagram

# USE CASE DIAGRAM

Use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors.

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high- level functionality of a system and also tells how the user handles a system.

Purposes of a use case diagram given below:

* + 1. It gathers the system's needs.
    2. It depicts the external view of the system.
    3. It recognizes the internal as well as external factors that influence the system.
    4. It represents the interaction between the actors.

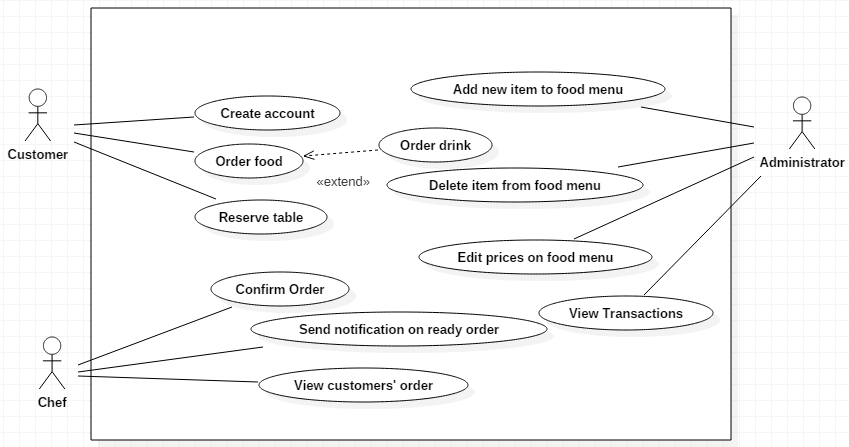


Fig. 6: Use Case Diagram

# ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

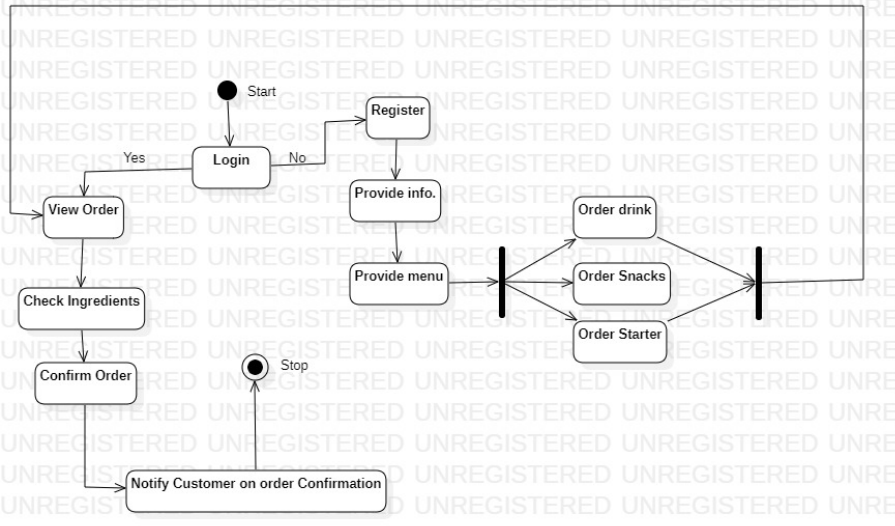


Fig. 7: Activity Diagram

# SEQUENCE DIAGRAM

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching

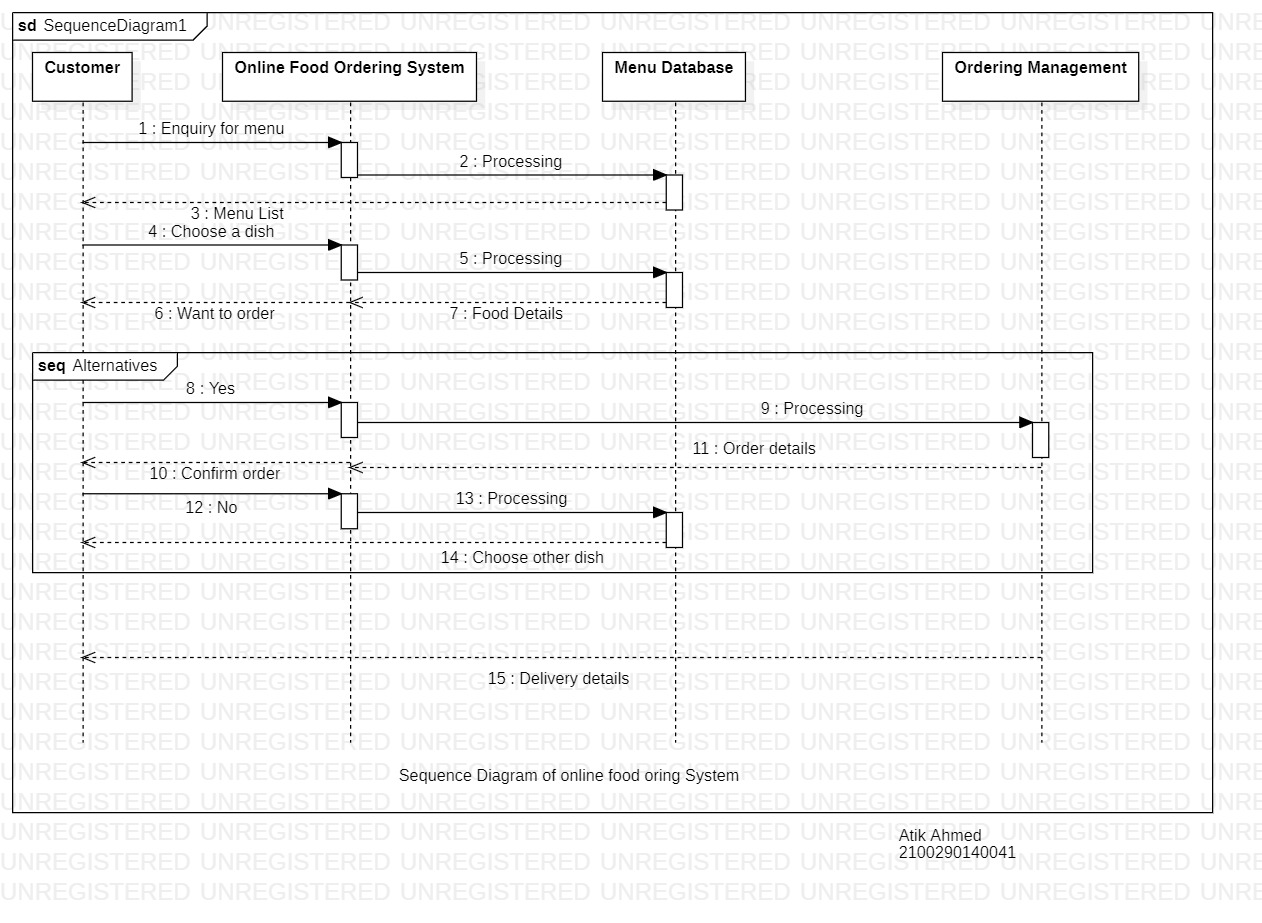
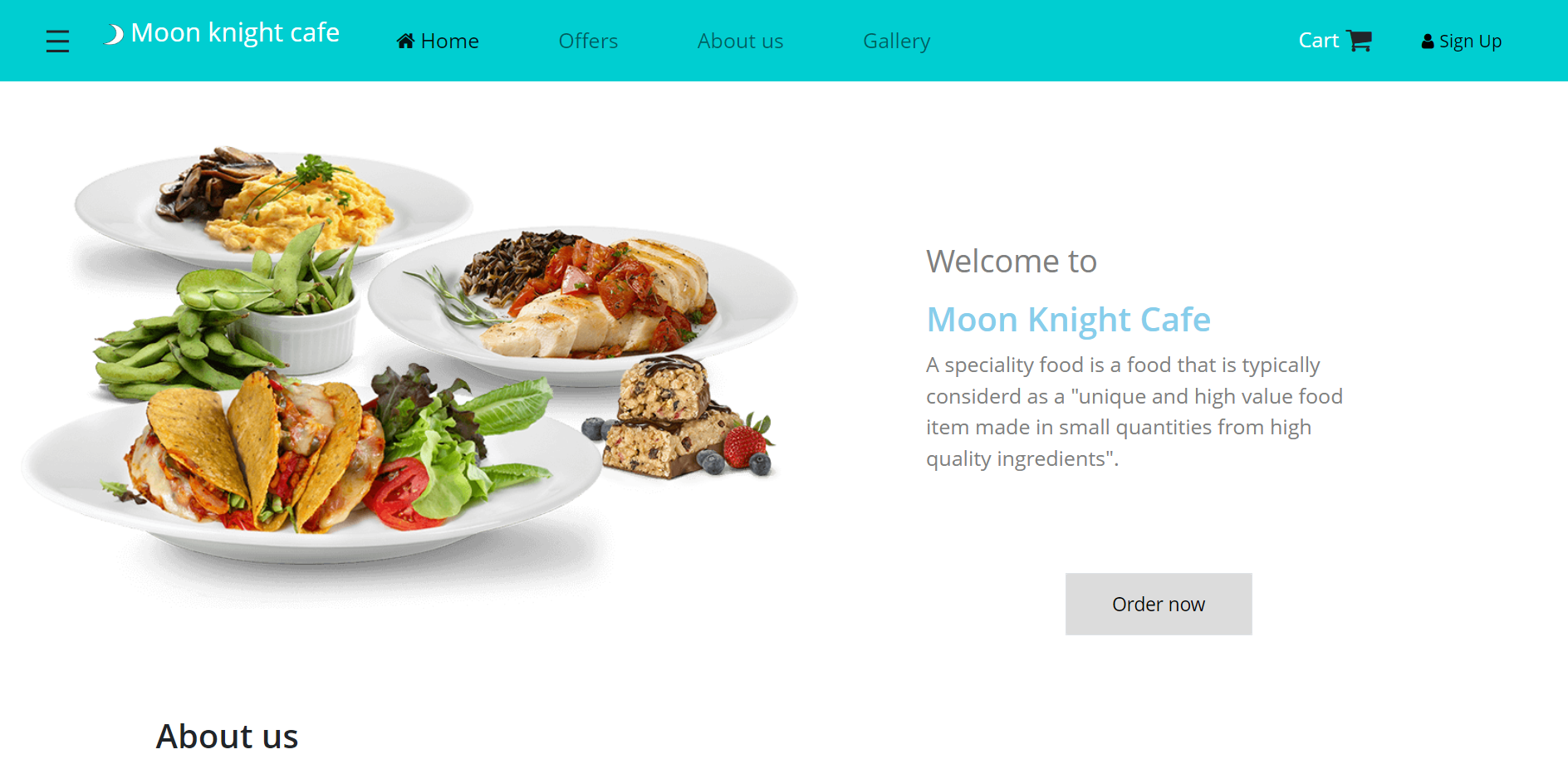


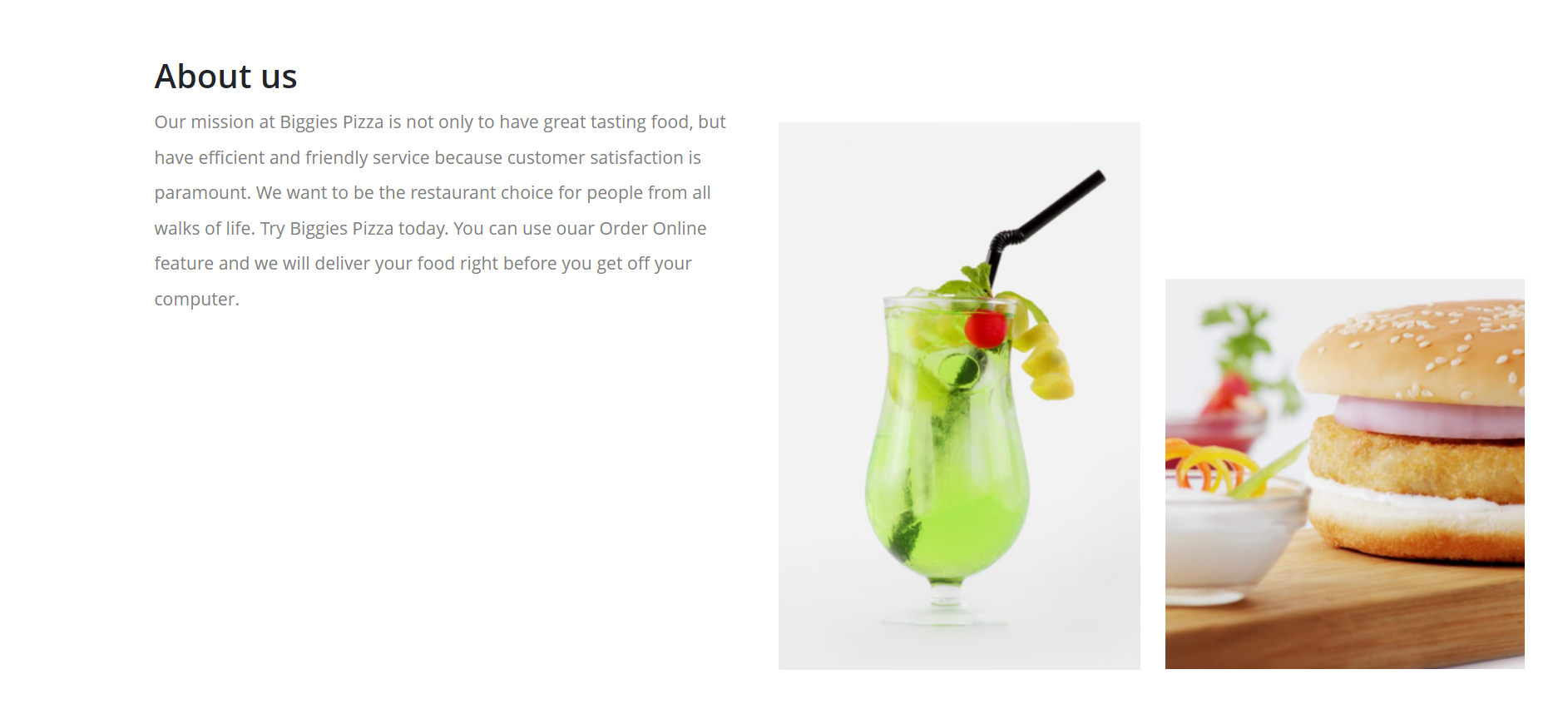
Fig. 8: Sequence Diagram

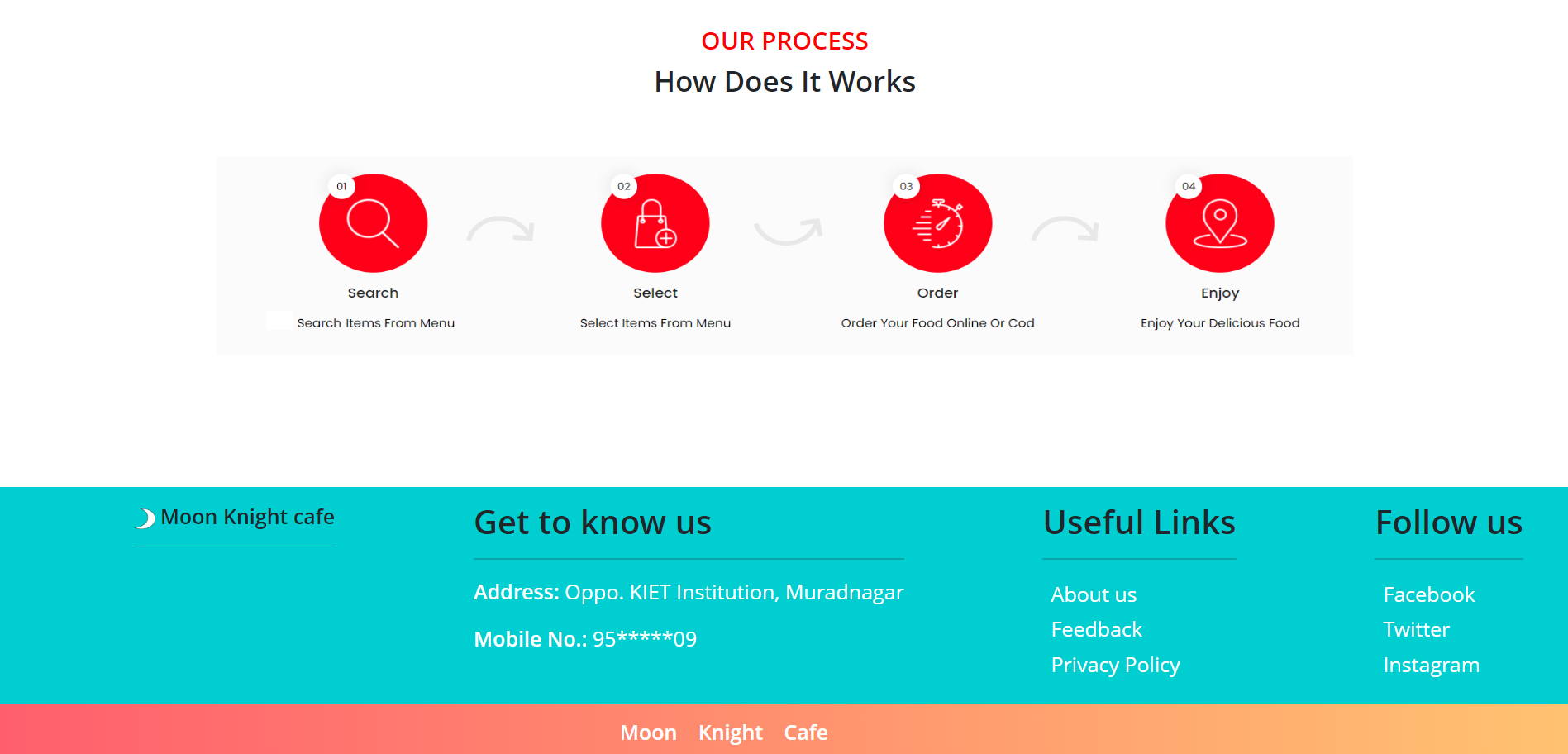
**CHAPTER 6**

**CODING AND IMPLEMENTATION**

**6.1 Homepage**

****

****

****

<!DOCTYPE html>

<html>

<head>

<meta name='viewport' content='width=device-width, initial-scale=1'>

<link rel="stylesheet" type="text/css" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="E:\Project 2/Index.css">

<title>Home</title>

</head>

<body>

<script type="text/javascript" src="index.js"></script>

<nav class="navbar navbar-expand-lg navbar-light " style="background-color: darkturquoise;">

<div class="container-fluid">

<div class="top" style="margin-left: 30px;">

<div id="mySidenav" class="sidenav">

<a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>

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

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

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

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

</div>

<span class="open" style="font-size:30px;cursor:pointer" onclick="openNav()">&#9776;</span>

</div>

<p class="navbar-brand" href="#" style="color: white; font-size: 25px; padding-left: 30px;">

<img src="moon 2.png" height="20px" width="20px"\> Moon knight cafe</p>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

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

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav" style="margin-left: -30px;">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="#"><i class="fa fa-home" style="padding-right: 5px" aria-hidden="true"></i>Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Offers</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">About us</a>

</li>

<li class="nav-item">

<a class="nav-link " href="#" tabindex="-1" aria-disabled="true">Gallery</a>

</li>

</ul>

<div id="id01" class="cont\_principal">

<div class="cont\_centrar">

<span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close Modal">&times;</span>

<div class="cont\_login">

<form action="server.php" method="post">

<div class="cont\_tabs\_login">

<ul class='ul\_tabs'>

<li class="active"><a href="#" onclick="sign\_in()">SIGN IN</a>

<span class="linea\_bajo\_nom"></span>

</li>

<li><a href="#up" onclick="sign\_up()">SIGN UP</a><span class="linea\_bajo\_nom">

</span>

</li>

</ul>

</div>

<div class="cont\_text\_inputs">

<input type="text" class="input\_form\_sign " placeholder="NAME" name="name" />

<input type="text" class="input\_form\_sign d\_block active\_inp" placeholder="EMAIL" name="email" />

<input type="password" class="input\_form\_sign d\_block active\_inp" placeholder="PASSWORD" name="password" />

<input type="password" class="input\_form\_sign" placeholder="CONFIRM PASSWORD" name="conf\_pass" />

<a href="#" class="link\_forgot\_pass d\_block" >Forgot Password ?</a>

<div class="terms\_and\_cons d\_none">

<p><input type="checkbox" name="terms\_and\_cons" /> <label for="terms\_and\_cons">Accept Terms and Conditions.</label></p>

</div>

</div>

<div class="cont\_btn">

<button class="btn\_sign" >SIGN IN</button>

</div>

</form>

</div>

</div>

</div>

<p style="color: white;position: relative;left: 170px;top: 7px;cursor: pointer;">Cart</p><i style="font-size: 27px;" class="fa fa-shopping-cart" aria-hidden="true"></i><button class="signup" onclick="document.getElementById('id01').style.display='block'" style="width:auto;"><i class="fa fa-user" style="padding-right: 5px;" aria-hidden="true"></i>Sign Up</button>

</div>

</div>

</nav>

<div class="container">

<img src="pngegg.png" width="100%" alt="..." >

<div class="content">

<p>Welcome to</p><br><br>

<h2>Moon Knight Cafe</h2>

<p style="font-size: 20px; color: grey">A speciality food is a food that is typically considerd as a "unique and high value food item made in small quantities from high quality ingredients".</p>

<div class="containers">

<div class="center">

<button class="btn">

<svg width="180px" height="60px" viewBox="0 0 180 60" class="border">

<polyline points="179,1 179,59 1,59 1,1 179,1" class="bg-line" />

<polyline points="179,1 179,59 1,59 1,1 179,1" class="hl-line" />

</svg>

<span><a href="E:\Project 2\dishes.html" style="position: relative; text-decoration: none;color: black"> Order now</a></span>

</div>

</div>

</div>

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

<div style="padding-left: 150px;padding-top: 50px;">

<h2>About us</h2>

<p class="about">Our mission at Biggies Pizza is not only to have great tasting food, but have efficient and friendly service because customer satisfaction is paramount. We want to be the restaurant choice for people from all walks of life.

Try Biggies Pizza today. You can use ouar Order Online feature and we will deliver your food right before you get off your computer.</p></div>

<div class="about\_img" style="padding-left: 750px;margin-top: -300px;word-spacing: 20px; ">

<img src="drink.png" height: 200px >

<img src="burger.png" style="margin-top: 150px;">

</div><br>

<hr><br>

<div>

<div class="process">

<h4 style="color: red">OUR PROCESS</h4>

<h3>How Does It Works</h3><br><br>

</div> <header style="position: relative;height: 320px; width: 1100px; left: 0px;">

<img src="process.png" width="100%" height="60%"></div>

</header>

<div class="allheader" style="width: 100%">

<div class="overview" id="dt" style="position: relative;top: -10px;">

<h5><img src="moon 2.png" height="20px" width="20px"> Moon Knight cafe</h5><hr>

</div>

<div class="Download" id="dt">

<h2> Get to know us</h2><hr><div style="color: white;">

<p><STRONG>Address:</strong> Oppo. KIET Institution, Muradnagar</p>

<p><strong>Mobile No.:</strong> 95\*\*\*\*\*09</p>

</div></div>

<div class="Links" id="dt">

<h2>Useful Links</h2><hr>

<a class="imgs" href="#">About us</a><br>

<a class="imgs" href="#">Feedback</a><br>

<a class="imgs" href="#">Privacy Policy</a><br>

</div>

<div class="Media" id="dt">

<h2>Follow us</h2><hr>

<a class="imgs" href="#">Facebook</a><br>

<a class="imgs" href="#">Twitter</i></a><br>

<a class="imgs" href="#">Instagram</a><br>

</div>

</div>

<div class="copyright">

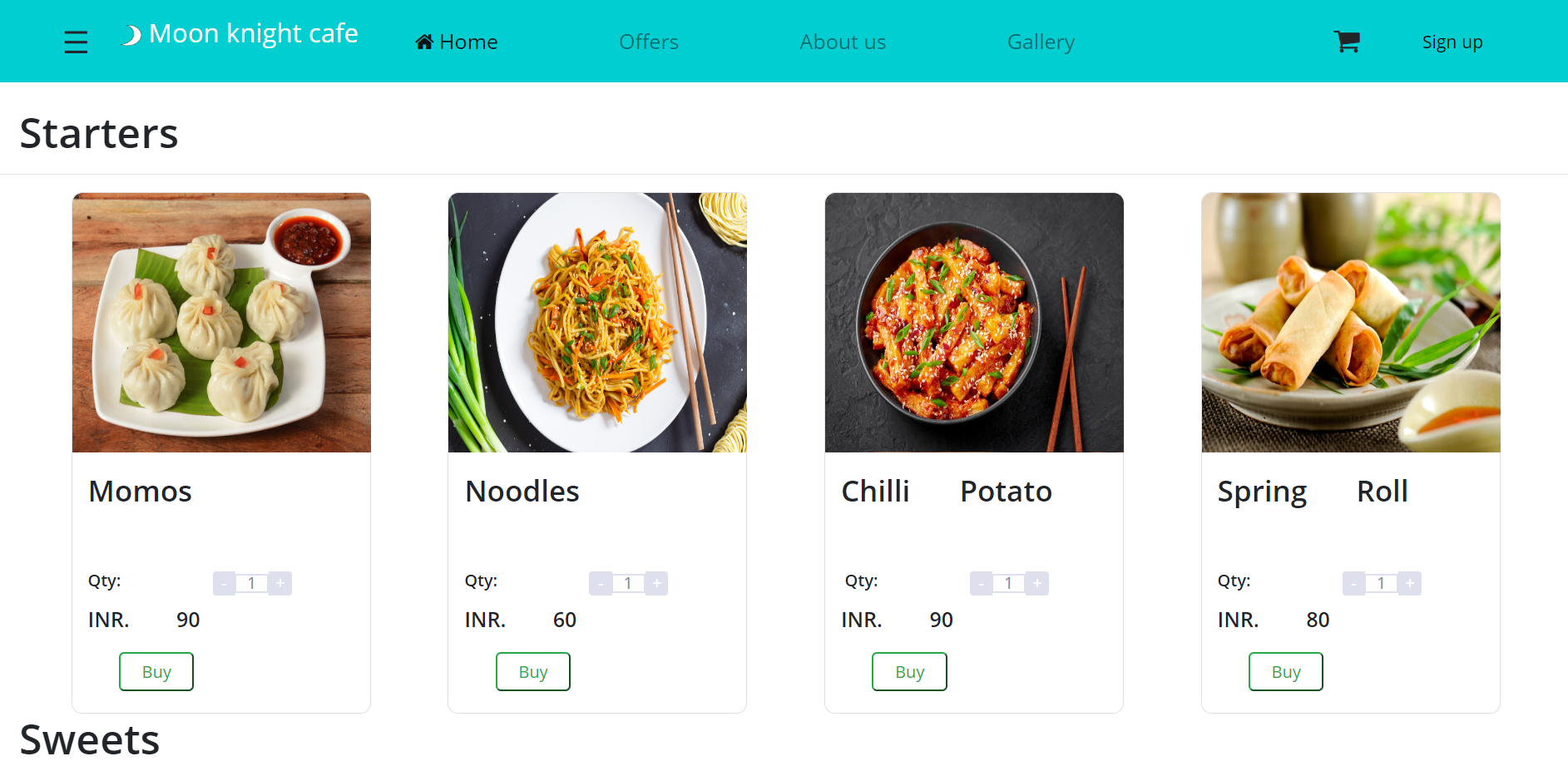
<p style="font-family: serif;"><strong>Moon Knight Cafe</p>

</div>

</body>

</html>

**Dishes:**





<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" type="text/css" href="E:\Project 2/Index.css">

<script type="text/javascript" src="index.js"></script>

<title>Dishes</title>

<style type="text/css">

.nav-item{

padding-right: 40px;

}

li a:hover{

border: 1px solid black;

background: linear-gradient(to right, #abbaab, #ffffff);

transition-delay: 1s;

}

.card{

position: static;

display: inline-grid;

text-decoration-style: none;

word-spacing: 40px ;

margin-left:70px;

border-radius: 10px;

}

.card-body{

margin-left: -5px;

}

.card-img-top{

width: 287px;

height: 250px;

border-top-left-radius: 10px;

border-top-right-radius: 10px;

}

.card:hover{

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

transition-duration: 0.5s;

cursor: pointer;

}

.btn-outline-success{

padding: 5px 20px 5px 20px;

margin-left: 30px;

border-radius: 5px;

transition-duration: 0.5s;

}

.close{

position: relative;

top: 5px;

left: -10px;

font-size: 25px;

cursor: pointer;

}

/\* a:link, a:visited {

transition: color 0.15s ease 0s, border-color 0.15s ease 0s, background-color 0.15s ease 0s;

} \*/

.quantity {

display: flex;

align-items: center;

justify-content: center;

padding: 0;

}

.quantity\_\_minus,

.quantity\_\_plus {

display: block;

width: 22px;

height: 23px;

margin: 0;

background: #dee0ee;

text-decoration: none;

text-align: center;

line-height: 23px;

}

.quantity\_\_minus:hover,

.quantity\_\_plus:hover {

background: #575b71;

color: #fff;

}

.quantity\_\_minus {

border-radius: 3px 0 0 3px;

}

.quantity\_\_plus {

border-radius: 0 3px 3px 0;

}

.quantity\_\_input {

width: 32px;

height: 19px;

margin: 0;

padding: 0;

text-align: center;

border-top: 2px solid #dee0ee;

border-bottom: 2px solid #dee0ee;

border-left: 1px solid #dee0ee;

border-right: 2px solid #dee0ee;

background: #fff;

color: #8184a1;

}

.quantity\_\_minus:link,

.quantity\_\_plus:link {

color: #8184a1;

}

.quantity\_\_minus:visited,

.quantity\_\_plus:visited {

color: #fff;

}

.qty{

font-size: 16px;

position: relative;

top: 5px;

}

.btn-outline-success{

cursor: pointer;

}

</style>

</script>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-light " style="background-color: darkturquoise;">

<div class="container-fluid">

<div class="top" style="margin-left: 30px;">

<div id="mySidenav" class="sidenav">

<a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>

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

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

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

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

</div>

<span class="open" style="font-size:30px;cursor:pointer" onclick="openNav()">&#9776;</span>

</div>

<p class="navbar-brand" href="#" style="color: white; font-size: 25px; padding-left: 30px;">

<img src="moon 2.png" height="20px" width="20px"\> Moon knight cafe</p>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

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

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav" style="margin-left: -30px;">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="E:\Project 2/index.html"><i class="fa fa-home" style="padding-right: 5px" aria-hidden="true"></i>Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Offers</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">About us</a>

</li>

<li class="nav-item">

<a class="nav-link " href="#" tabindex="-1" aria-disabled="true">Gallery</a>

</li>

</ul>

<div id="id01" class="cont\_principal">

<div class="cont\_centrar">

<span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close Modal">&times;</span>

<div class="cont\_login">

<form action="">

<div class="cont\_tabs\_login">

<ul class='ul\_tabs'>

<li class="active"><a href="#" onclick="sign\_in()">SIGN IN</a>

<span class="linea\_bajo\_nom"></span>

</li>

<li><a href="#up" onclick="sign\_up()">SIGN UP</a><span class="linea\_bajo\_nom">

</span>

</li>

</ul>

</div>

<div class="cont\_text\_inputs">

<input type="text" class="input\_form\_sign " placeholder="NAME" name="name\_us" />

<input type="text" class="input\_form\_sign d\_block active\_inp" placeholder="EMAIL" name="emauil\_us" />

<input type="password" class="input\_form\_sign d\_block active\_inp" placeholder="PASSWORD" name="pass\_us" />

<input type="password" class="input\_form\_sign" placeholder="CONFIRM PASSWORD" name="conf\_pass\_us" />

<a href="#" class="link\_forgot\_pass d\_block" >Forgot Password ?</a>

<div class="terms\_and\_cons d\_none">

<p><input type="checkbox" name="terms\_and\_cons" /> <label for="terms\_and\_cons">Accept Terms and Conditions.</label></p>

</div>

</div>

<div class="cont\_btn">

<button class="btn\_sign">SIGN IN</button>

</div>

</form>

</div>

</div>

</div>

<i style="font-size: 27px;" class="fa fa-shopping-cart" aria-hidden="true"></i> <button class="signup" onclick="document.getElementById('id01').style.display='block'" style="width:auto;">Sign up</button>

</div>

</div>

</nav>

<br>

<header class="border-0" > <h1 style="margin-left: 20px;">Starters</h1><hr>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="momos.jpg" alt="Card image cap">

<div class="card-body">

<h3>Momos</h3><br>

<div class="container">

<div class="quantity"> <strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 90</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="noodles.jpg" alt="Card image cap">

<div class="card-body">

<h3>Noodles</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 60</p>

<a href="E:\Project 2\checkout.html"><button type="button" class="btn-outline-success">Buy</button></a>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="chiili potato.png" alt="Card image cap">

<div class="card-body">

<h3>Chilli Potato</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 90</p>

<a href="E:\Project 2\checkout.html"><button type="button" class="btn-outline-success">Buy</button></a>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="spring roll.jpg" alt="Card image cap">

<div class="card-body">

<h3>Spring Roll</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 80</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

</header>

<header class="border-0" class="card"><h1 style="margin-left: 20px;">Sweets</h1><hr>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="rasgulla.jpg" alt="Card image cap">

<div class="card-body">

<h3>Rosgulla</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 80</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="Rasmalai.webp" alt="Card image cap">

<div class="card-body">

<h3>Rasmalai</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 90</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="kaju\_barfi.gif" alt="Card image cap">

<div class="card-body">

<h3>Kaju Barfi</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 860</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="ghewar.jpg" alt="Card image cap">

<div class="card-body">

<h3>Ghewar</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 360</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

</header><br><br>

<header class="border-0" class="card"><h1 style="margin-left: 20px;">Special Chaap</h1><hr>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="Achari Chaap.jpg" alt="Card image cap">

<div class="card-body">

<h3>Achari Chaap</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 180</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="Afghani chaap.jpg" alt="Card image cap">

<div class="card-body">

<h3>Afghani Chaap</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 190</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="Tandoori chaap.jpg" alt="Card image cap">

<div class="card-body">

<h3>Tandoori Chaap</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 190</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="Malai Chaap.jpg" alt="Card image cap">

<div class="card-body">

<h3>Malai Chaap</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 190</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

</header>

<header class="border-0" class="card"><h1 style="margin-left: 20px;">Non-Veg</h1><hr>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="Chicken lolipop.jpg" alt="Card image cap">

<div class="card-body">

<h3>Chicken lolipop</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 230</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="Chiken Tikka.jpg" alt="Card image cap">

<div class="card-body">

<h3>Chicken Tikka</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 190</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="Chicken Kabab.jpg" alt="Card image cap">

<div class="card-body">

<h3>Chicken Kabab</h3><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 160</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="Chicken Tangadi.jpg" alt="Card image cap">

<div class="card-body">

<h4>Chicken tangadi</h4><br>

<div class="container">

<div class="quantity"><strong><p class="qty"> Qty: &nbsp </strong></p>

<a href="#" class="quantity\_\_minus"><span>-</span></a>

<input name="quantity" type="text" class="quantity\_\_input" value="1">

<a href="#" class="quantity\_\_plus"><span>+</span></a>

</div>

</div>

<p class="card-text"><STRONG>INR. 290</p>

<button type="button" class="btn-outline-success">Buy</button>

</div>

</div>

</header>

<br>

<div>

</div>

<div class="allheader" style="width: 100%">

<div class="overview" id="dt">

<h5><img src="moon 2.png" height="20px" width="20px"> Moon Knight cafe</h5><br><hr>

</div>

<div class="Download" id="dt">

<h2> Get to know us</h2><hr>

<div class="know\_us">

<p><STRONG>Address:</strong> Oppo. KIET Institution, Muradnagar</p>

<p><strong>Mobile No.:</strong> 95\*\*\*\*\*09</p>

</div></div>

<div class="Links" id="dt">

<h2>Useful Links</h2><hr>

<a class="imgs" href="#">About us</a><br>

<a class="imgs" href="#">Feedback</a><br>

<a class="imgs" href="#">Privacy Policy</a><br>

</div>

<div class="Media" id="dt">

<h2>Follow us</h2><hr>

<a class="imgs" href="#">Facebook</a><br>

<a class="imgs" href="#">Twitter</i></a><br>

<a class="imgs" href="#">Instagram</a><br>

</div>

</div>

<div class="copyright">

<p style="font-family: serif;"><strong>Moon Knight Cafe</p>

</div>

</body>

</html>

**CONCLUSION**

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses. At the end it is concluded that we have made effort on following points...

• A description of the background and context of the project and its relation to work already done in the area.

• Made statement of the aims and objectives of the project.

• The description of Purpose. Scope, and applicability

• We define the problem on which we are working in the project.

• We describe the requirement Specifications of the system and the actions that can be done on these things.

• We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.

• We included features and operations in detail, including screen layouts.

• We designed user interface and security issues related to system . Finally the system is implemented and tested according to test cases.

**Limitation of Project on Online Food Ordering System:-**

Although I have put my best efforts to make the software flexible, easy to operate butlimitations cannot be ruled out even by me. Though the software presents a broad range ofoptions to its users some intricate options could not be covered into it; partly because oflogistic and partly due to lack of sophistication. Paucity of time was also major constraint,thus it was not possible to make the software foolproof and dynamic Lack of time alsocompelled me to ignore some part such as storing old result of the candidate etc.Considerable efforts have made the software easy to operate even for the people not relatedto the field of computers but it is acknowledged that a layman may find it a bit problematicat the first instance. The user is provided help at each step for his convenience in working with the software.

**List of limitations which is available in the Online Food Ordering System:-**

**•** Excel export has not been developed for Food Item Category due to some criticality

• The transactions are executed in off-line mode, hence on-line data for Customer Order capture and modification is not possible.

• Off-line reports of Food Item. Confirm Order. Customer cannot be generated due to batch mode execution.