B.TECH. (CSE)

V SEMESTER

UE20CS303 –SOFTWARE ENGINEERING

PROJECT REPORT

ON

ONLINE FOOD DELIVERY System

SUBMITTED BY

NAME:	SRN:
Shafiudeen Kameel	PES2UG20CS320
Ram Selevaraj	PES2UG20CS265
Sagrikha Muralidharan	PES2UG20CS292
Riya Hurtis	PES2UG20CS277

August – Nov 2022

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BENGALURU – 560100, KARNATAKA, INDIA

TABLE OF CONTENTS

PROPOSAL OF THE PROJECT

Project Description:

The online food delivery platform enables users to choose from a wide variety of cuisines. It has an Admin Dash which provides information about the pending orders, cancelled orders, on-delivery orders, update orders etc. It also has a User Dash which shows details about present and past orders, re-order etc. It also shows recommended food items

This project is done by Shafiudeen Kameel, Ram Selevaraj, Sagarikha Muralidharan and Riya Hurtis for the Software Engineering Project for the year 2022 (August- December). This project has features like – Privacy protection of user credentials using cryptography, wallet features, weather module for dynamic pricing, and has a Website for admin user interface. In the present decade, there are many competitive transport platforms available that do not provide customer budget friendly prices along with current scenario dynamic fee structure.

Plan of work and product ownership:

We are currently planning to explore the different software available in the marketrelated to our topic to figure out necessary features. Then we are planning to do a literature review. Then we'll look ahead modify the source code to our requirements. While doing that we will divide the work equally among the team of 4 members.

Kameel: Project lead and doing the encryption and decryption modules for the privacy of user details.
Ram: Team Lead and Wallet feature.
Sagarikha: Programmer Analyst, Food aggregator list feature.
Riya: Testing Engineer, Algorithms, and reuse components code manipulation.

Functionality:

- Food aggregators list: build component
- Dynamic Fare: reuse component
- Encryption/Decryption: build component
- Wallet: build component
- Weather forecast: build component
- User details: reuse component
- Algorithm: build component.

User interface, graphic design, database interaction, unit testing, etc. are not functional features.

- Website for administrative/user use only, no graphics design, databases flat file – encryptedtext file

Qualitative property, if any, that you will contribute, such as fine tune the system performance to achieve response time under 5 seconds, or develop and evaluate an easy-to-use user interface for this-and-this specific functional feature, or ensure confidentiality of a specific set of data, etc.

We tweaked our algorithm for the dynamic fare calculator to improve its efficiency. The parameter of weather was included and based on that condition; the technique evolved in its performance.

SRS: SOFTWARE REQUIREMENTS SPECIFICATION

Table of Contents

Introduction

Purpose

The software is supposed to enable the user to order food online and have
it delivered to their house.

Intended Audience

Users above the age of 13, those who do not have time to cook food at home or visit

Product Scope

The online food delivery platform enables users to choose from a wide variety of cuisines. It has an Admin Dash which provides information about the pending orders, cancelled orders, on-delivery orders, update orders etc. It also has a User Dash which shows details about present and past orders, re-order etc. It also shows recommended food items

References

Swiggy

Zomato

Uber Eats

Amazon Food

Overall Description

Product Functions

1) The software provides checkboxes and textboxes to collect data from the user
2) The interface is easy to use
3) Keeps track of current and past orders
4) Generates the total cost including tax and delivery fee based on distance and weather conditions

User Classes and Characteristics

The product is going to be used by a person who needs to know how to read the English language only. And is not of unsound mind. He/ She needs to get used to the simple UI for the administrator account and should be willing to help the customers who are his clients.
Please note that this software isn't for the common users to install or run in their devices. But limited to only the cab Unions.

Operating Environment

The software will operate on an online browser based platform. It is cross-platform available on PC and Mobile. The phone app has the UPI API hook which enables easy payments

.

Design and Implementation

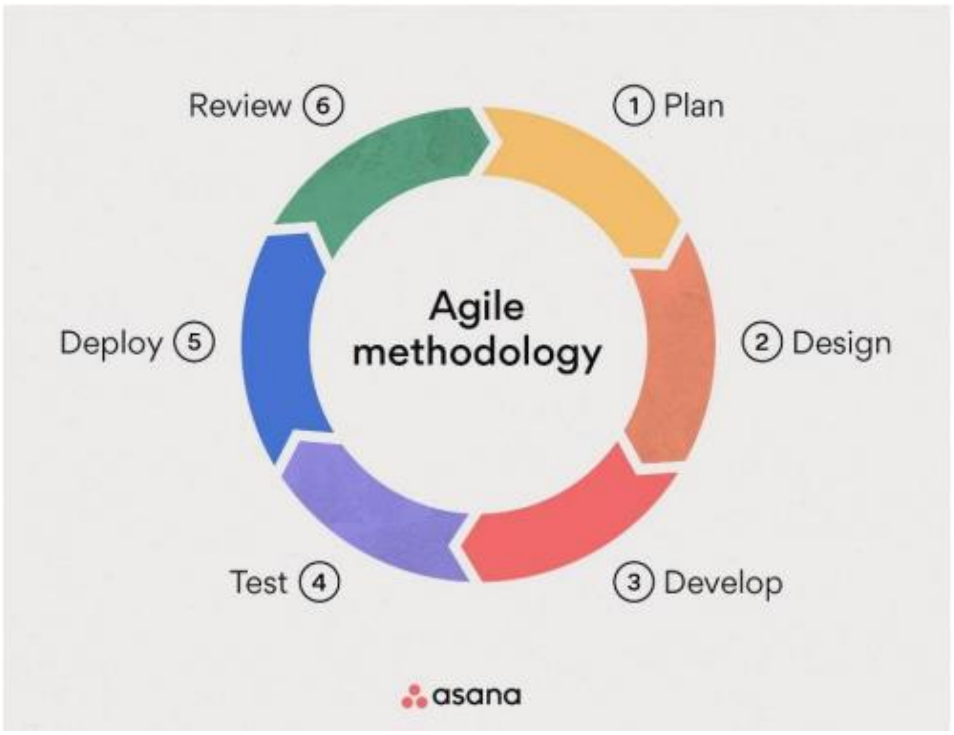
Implementation of payment systems relies on integration with the UPI payment service, which is a major dependency. The software is only in English as of now although a large portion of userbase might only speak local languages. We will be using a database that consists of all the Food items, user ids etc.

Constraints Design

When planning a project, it is important to use a systems development life cycle (SDLC) to successfully take a project through the various stages of development there are twomain SDLC methodologies to plane and design the project based on the available of resources and other constrains namely; waterfall and agile

Agile Methodology:

The agile methodology is used when the requirements may change during the development of a project. Projects using agile development are broken up into sprints, each with their own analysis, design, implementation and testing phases. These sprints begin with a meeting, detailing the work to be done in the sprint with daily meetings to keep the project on target. Essentially, this breaks the project into smaller segments witheach segment having its own development phase, then at the end, the segments are combined to form the finished product. Figure 2.2 shows the agile method as a series of sprints with their own development life cycles.

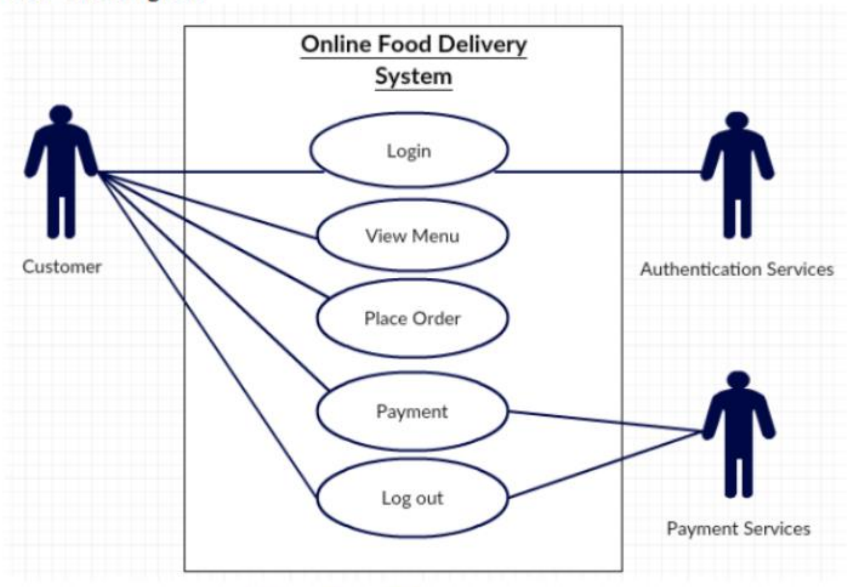


For this project, the agile methodology will be utilised. As there is project team, meetings on each sprint is being held, and the method of breaking down the project into smaller, more manageable segments will be used. This will allow the project to be able to adapt toany changes in the strategies of work. As this project consists of a web crawler and a webapplication, both the agile methodology and waterfall method are better suited to handlingmultiple deliverables than the single method.

System Architecture

There are 3 layers in this program, online web ordering layer, menu management layer and order retrieval system layer. These 3 work in tandem and simultaneously to make the process of ordering food easier.

User-case Diagram



Use Case Diagrams

Implementation Constraints

Moral regulations along with sensitive data to be protected are the only constraints which we will encounter during this time

2.6 Assumptions and Dependencies

Assumptions include that the User is literate in English and has basic technological acumen. We also assume that the UPI payments service is up and running at all times so the User never experiences failed transactions. All assets and components will be well documented with troubleshooting steps

External Interface Requirements

User Interfaces

We are using HTML,CSS and PHP to design the website. It is a fully interactive, intuitive and multi platform app. We also add some PHP animations to make the UI more appealing to the user.

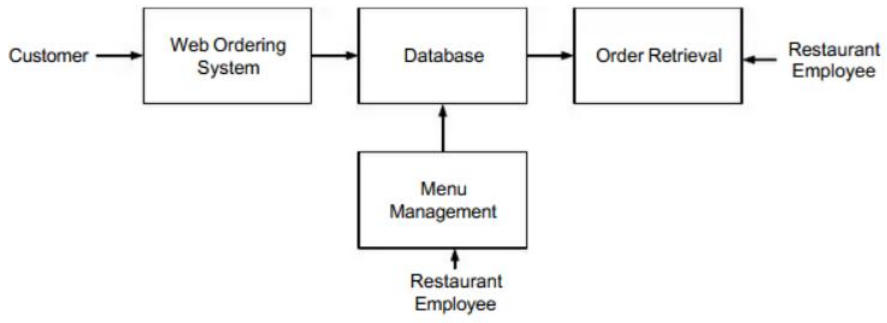
Software Interfaces

• Windows 10/Linux/ubuntu
• Visual studio code/Notepad++
• Browsers i.e., chrome, Bing
• Python 3.8 and above • HTML, CSS, JS

Communications Interfaces

When the user adds items, we temporarily store the items in a cart, then we calculate the total bill after retrieving the prices from our Food database, and once the user confirms the purchase, add the bill number and details to our Orders database. The user can then view his current and past orders from the dashboard

Analysis Models:



System Feature

The online food delivery platform enables users to choose from a wide variety of international cuisines. It has an Admin Dash which provides information about pending orders, cancelled orders, on-delivery orders etc. It also has a User Dash which shows details about present and past orders

A Easy to use UI.

7.1.1 Description and Priority

We provide High level of priority because we are the main connection point between the restaurant and end user. We handle the payments on the user’s end and payment to the restaurants This is very convenient for people who don’t have quick access to food.

7.1.2 Stimulus/Response Sequences When the user opens the website, it will list the food items available for purchase, along with pictures and price details, the user can then select and add the desired food items to his/her cart and confirm the purchase. He/She can then view the current order and past orders from the dashboard

7.1.3 Functional Requirements

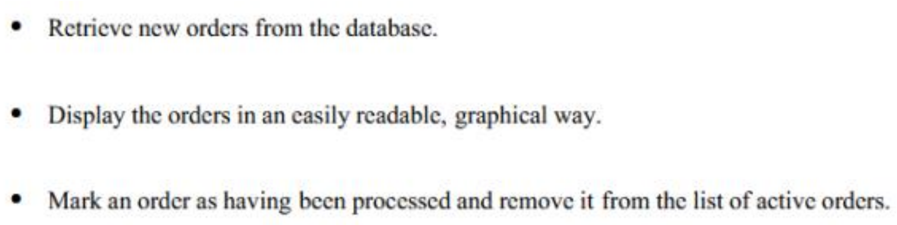
Web Ordering System –

- 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.

Menu Management System-

- 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.

Order Retrieval System -



- 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.

Other Non-functional Requirements

Performance Requirements

The server shall be capable of supporting an arbitrary number of surface computers, tablets and displays, that is, it shall provide no limit on how many devices are in the system. The server shall be capable of supporting an arbitrary number of active customer payments, that is, no payments shall be lost under any circumstance

Safety Requirements

• The system shall log every state and state change of every surface computer, tablet and display to provision recovery from system failure.
• The system shall be capable of restoring itself to its previous state in the event of failure (e.g. a system crash or power loss).
• The system shall be able to display a menu at all times to facilitate manual order taking should the need arise.
• The system shall utilise periodic 30-second keep-alive messages between tablets and the server to monitor tablet operational status.

Security Requirements

• A user password used for Account login must have a bit-strength of at least 64 bits.
• A user password used for Account login must be changed once a year.
• A user shall only be able to place orders on one system at a time.
• OTP will be generated each time the user wants to log in.

Software Quality Attributes

The software shall be capable of supporting an arbitrary number of surface computers, tablets and displays, that is, it shall provide no limit on how many devices are in the system. It shall be capable of supporting an arbitrary number of active meals/orders, that is, no meals/orders shall be lost under any circumstances

Business Rules

• The customer must supply a local telephone number number for web orders.
• A valid credit/debit card must be available for web order confirmation.
• The web order must be confirmed by the head waitress via return phone call.
• Preparation of the web order is not started until after the confirmation call.
• The credit/debit card number is held for web orders if the customer chooses to pay with cash or personal check.
• Payment for a web order is not processed until (failure of) customer pick-up.

Easy to run on most system-Portable

The app is designed to run on the various devices that have an android version of 5 and above.The app consists of the best and basic UI design which can be easily used by everyone without any struggles and comes with one time verification password(OTP) upon booking every ride which needs to be shared with the driver to confirm the ride effortlessly.

Other Requirements

The database us done with the help of phpadmin to store the user details, amount remaining in their wallet, order details.The software is open-source software.

Appendix A: Glossary

We have not used any short forms in this document; hence we are skipping this glossary.

Appendix B: Field Layouts

Appendix B: Field Layouts

An Excel sheet containing field layouts and properties/attributes and report requirements.

Sample sheet with information required to register the customer

Field	**Length**	**Data Type**	**Description**	**Is Mandatory**
Account Number	16	Numeric		Y
ISFC code	11	Alphanumeric		Y
Card Amount	20	Numeric		Y
Mandate Start Date	8	Date	Date of Mandate Registration	N
Mandate End Date	8	Date	Date of Mandate Expiry	N
Status	25	Alphanumeric	Status of Registration	Y
Customer Name	60	String		Y
Reject Reason Code	4	String	Reject Reason code in casemandate is rejected	N

Sample Report Requirements: Include the fields to be included in the report

Registration Report	**Transaction Report**
Bank Account Number	Transaction Reference Number
ISFC Code	Bank Account Number
Bank Name	IFSC Code
Account Status	Bank Name
Account Type	Customer Name
Customer Name	Card Number
Card Number	Debit Transaction Amount
SI Start Date	Transaction Date
Status	Status
Remarks	Debit Attempt Number
	Remarks

Appendix C: Requirement Traceability Matrix

SL No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID
1	WOS101	Interface for users to place their orders and view and track orders	agile	Swiggy & Zomato	-	TC-1	STC-10
2	MMS145	Interface for Admins to view and manage the menu visible to the users.	agile		-	TC-2	STC-20
3	ORS200	Show the orders in a graphical and easily readable manner	agile	Swiggy & Zomato	-	TC-3	STC-30

PROJECT PLAN DOCUMENT:

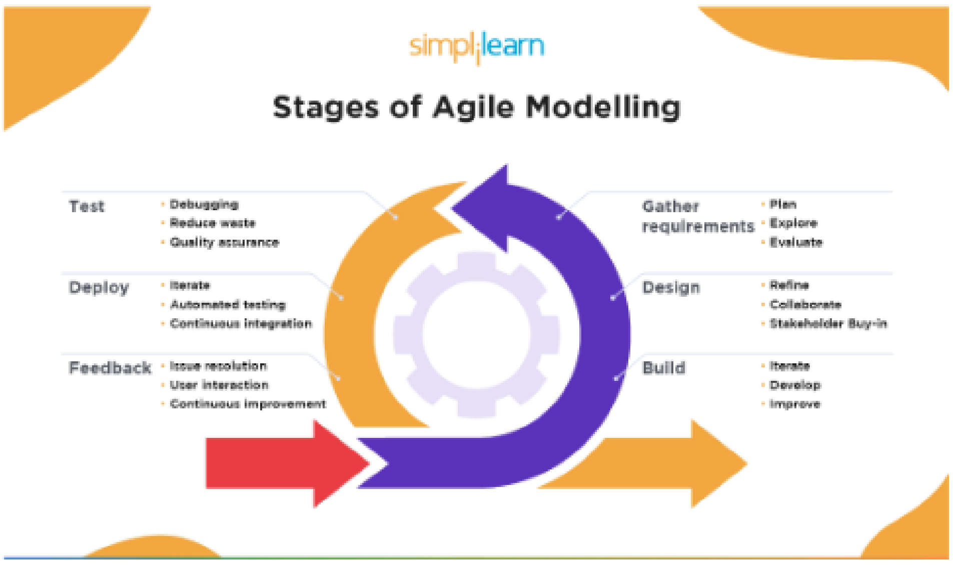
INTRODUCTON:

Our PHP-based Online Food Delivery System enables both external and internal clients (admins or employees) to interact with each other. Admin users can receive orders from users of our website, and users can choose what they would like to order from a large variety of food businesses. They may add multiple items from the same restraint and have it delivered to their doorstep. The option of adding more than one address (in the event that the user has a home and work address) is also available. Multiple payment options such as UPI, cash on Delivery have been implemented to make our website efficient and user friendly. Software-Hardware Integration: While there are many online food delivery systems on the market, it should be well integrated with hardware as well since not everyone may have access to smartphones. Due to this our website is laptop or pc compatible. Contactless booking and access to food is the main objective of this project. After the recent pandemic, many wish to order food from their favourite restraunts and eat it in the comfort of their homes. Some people may want to cancel their order, which we have made possible 3 minutes within placing the order. We also provide a feedback page, and a list of reasons for cancelling of the order. This helps us to improve backend glitches or problems if any. The aim is to automate its existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same.

1. PROCESS MODEL:

Our model is Agile Process model. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance. Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.
Following are the phases in the Agile model are as follows:

✓ Requirements gathering
✓ Design the requirements
✓ Construction/ iteration
✓ Testing/ Quality assurance
✓ Deployment
✓ Feedback



2. TOOLS USED:

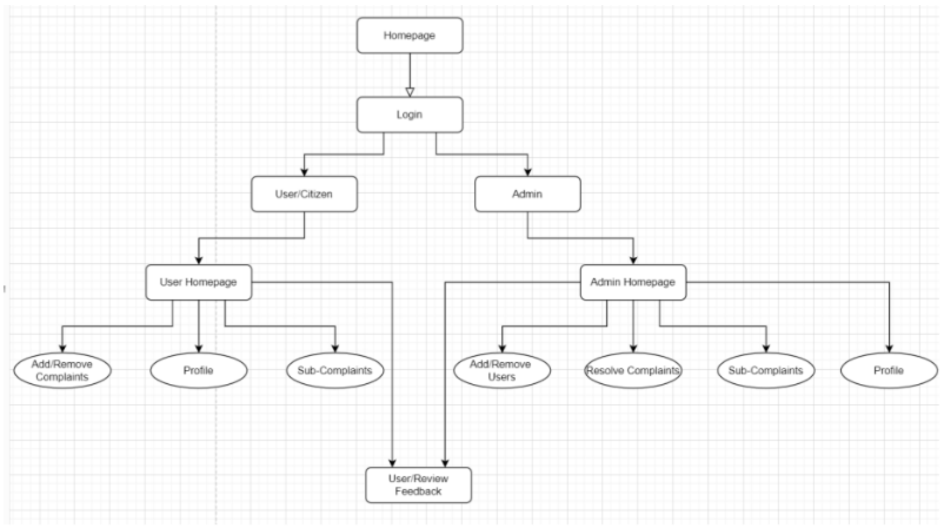
Planning tool: Jira
Design tool: Powerpoint, Adobe
Version Control: GIT
Development tool: VS code, Git hub tools
Bug tracking: Jira Testing tool: Jira

3. DELIVERABLES:

A website encompassing:

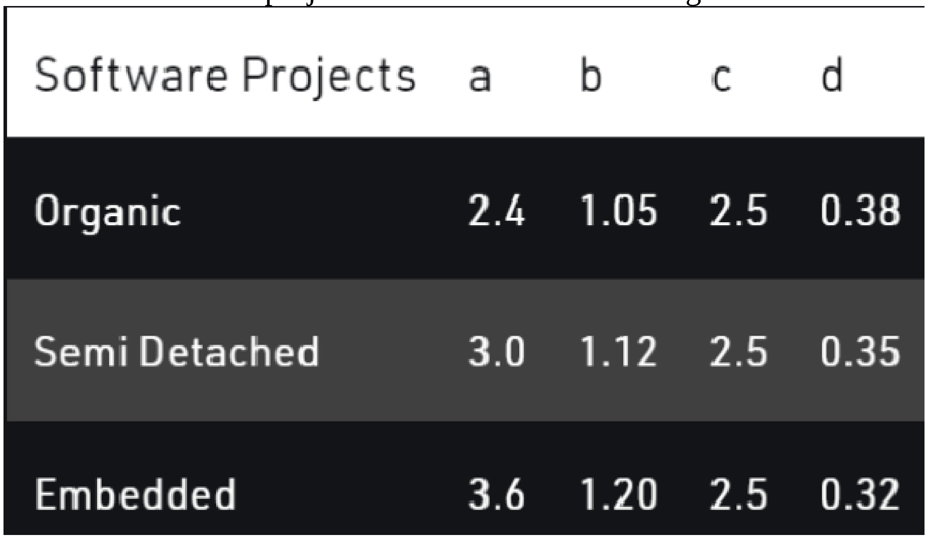
✓ A login page for user with otp 2- factor authentication
✓ A login page for admin with admin credentials
✓ A checkout page where user can review and place their order
✓ Payment page where mode of payment is chosen and payment is made
Code for the same
✓ User analytical tools integrated with the database

4. WORK BREAK DOWN STRUCTURE:



5. ESTIMATE OF EFFORT:

The problem statement of our project is well understood by us and it has been solved many times in the industry. The team members possess a decent knowledge in the domain. Our project would be classified as Organic.



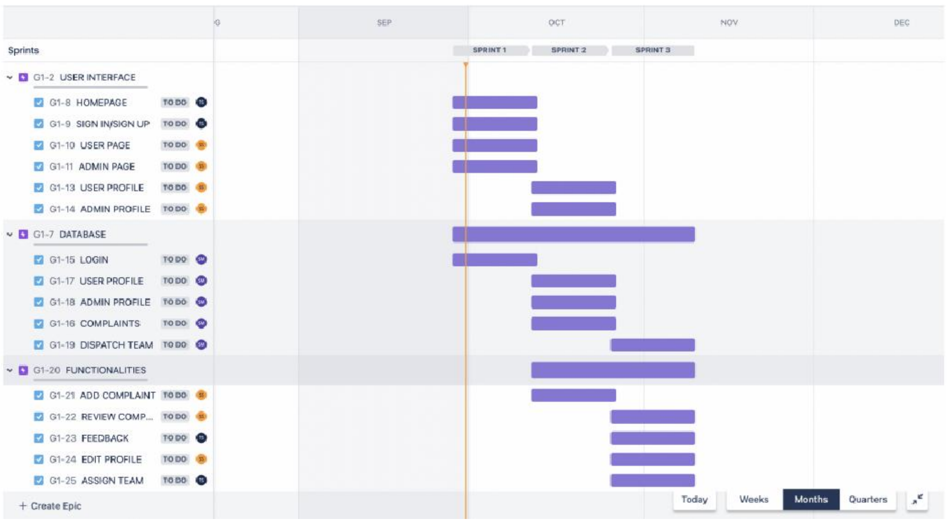
Software Projects	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

KLOC = 2

A = 2.4, B = 1.05

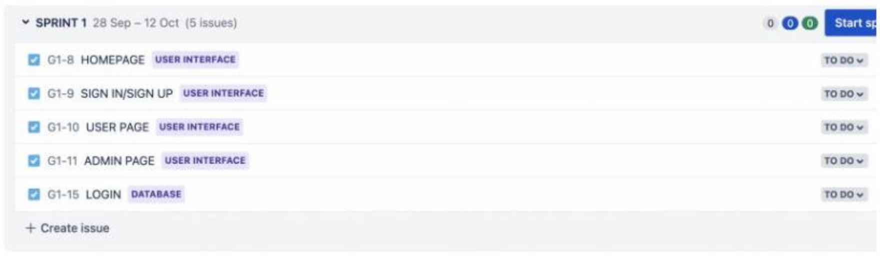
Effort = a(KLOC)^b = 2.4*(2)^1.05 = 4.96

6. GANT CHART:



Sprints	
G1-2 USER INTERFACE	
G1-8 HOMEPAGE	TO DO
G1-9 SIGN IN/SIGN UP	TO DO
G1-10 USER PAGE	TO DO
G1-11 ADMIN PAGE	TO DO
G1-13 USER PROFILE	TO DO
G1-14 ADMIN PROFILE	TO DO
G1-7 DATABASE	
G1-15 LOGIN	TO DO
G1-17 USER PROFILE	TO DO
G1-18 ADMIN PROFILE	TO DO
G1-16 COMPLAINTS	TO DO
G1-19 DISPATCH TEAM	TO DO
G1-20 FUNCTIONALITIES	
G1-21 ADD COMPLAINT	TO DO
G1-22 REVIEW COMP...	TO DO
G1-23 FEEDBACK	TO DO
G1-24 EDIT PROFILE	TO DO
G1-25 ASSIGN TEAM	TO DO

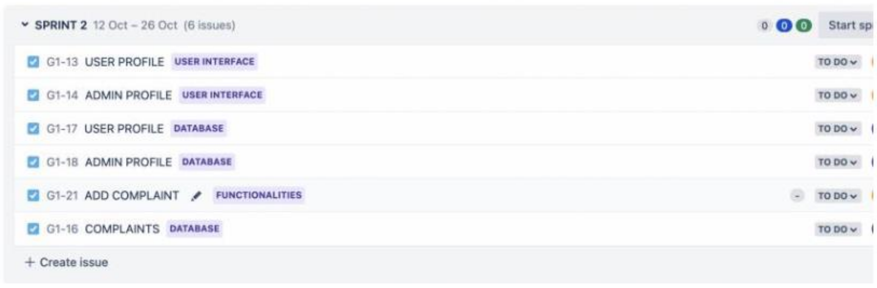
7. PRODUCT BACKLOG:



SPRINT 1 28 Sep – 12 Oct (5 issues) 0 0 0 Start sp

G1-8 HOMEPAGE	USER INTERFACE	TO DO
G1-9 SIGN IN/SIGN UP	USER INTERFACE	TO DO
G1-10 USER PAGE	USER INTERFACE	TO DO
G1-11 ADMIN PAGE	USER INTERFACE	TO DO
G1-15 LOGIN	DATABASE	TO DO

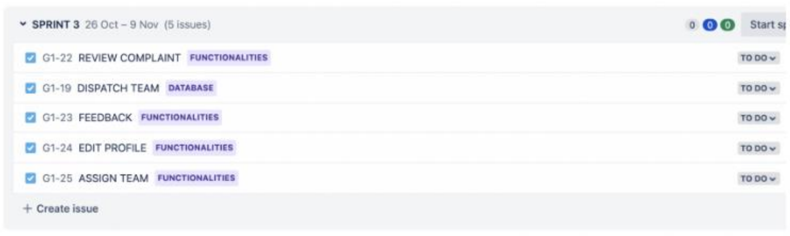
+ Create issue



SPRINT 2 12 Oct – 26 Oct (6 issues) 0 0 0 Start sp

G1-13 USER PROFILE	USER INTERFACE	TO DO
G1-14 ADMIN PROFILE	USER INTERFACE	TO DO
G1-17 USER PROFILE	DATABASE	TO DO
G1-18 ADMIN PROFILE	DATABASE	TO DO
G1-21 ADD COMPLAINT	FUNCTIONALITIES	TO DO
G1-16 COMPLAINTS	DATABASE	TO DO

+ Create issue

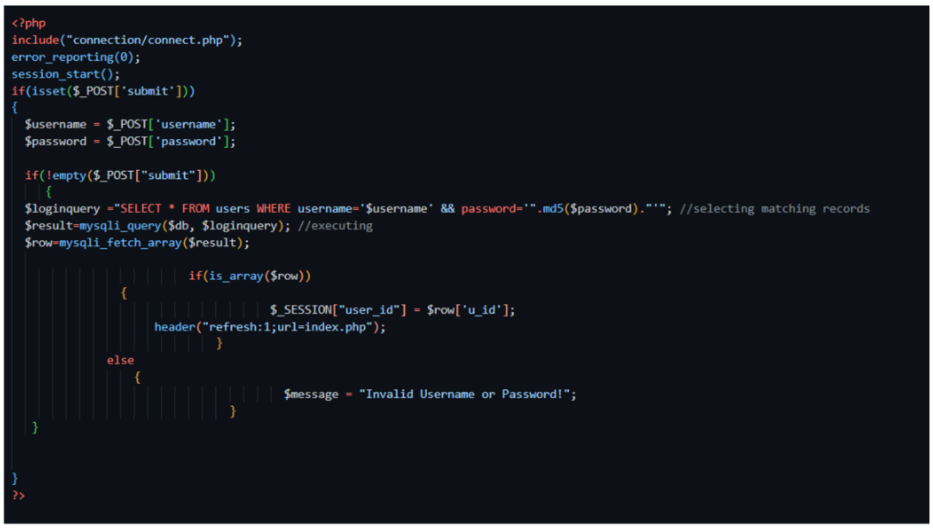


SPRINT 3 26 Oct – 9 Nov (5 issues) 0 0 0 Start sp

G1-22 REVIEW COMPLAINT	FUNCTIONALITIES	TO DO
G1-19 DISPATCH TEAM	DATABASE	TO DO
G1-23 FEEDBACK	FUNCTIONALITIES	TO DO
G1-24 EDIT PROFILE	FUNCTIONALITIES	TO DO
G1-25 ASSIGN TEAM	FUNCTIONALITIES	TO DO

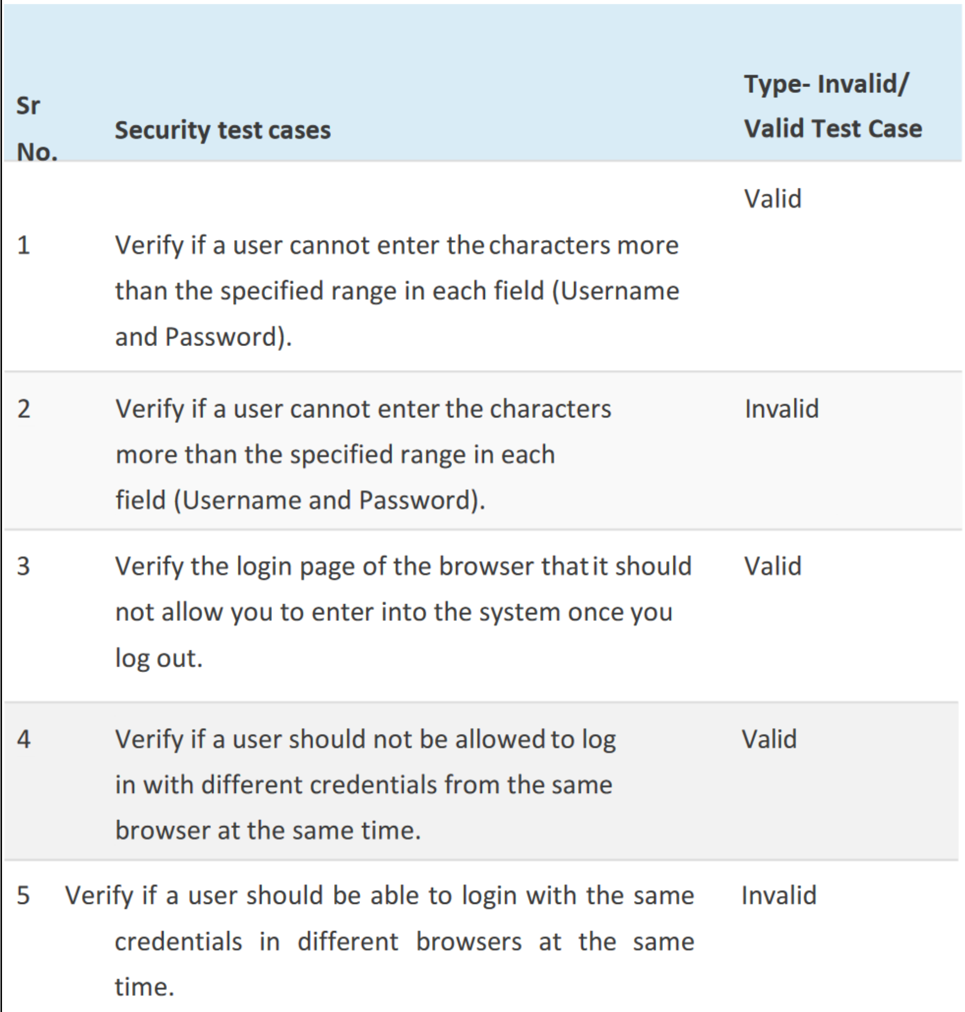
+ Create issue

TESTING:



```php
<?php
include("connection/connect.php");
error_reporting(0);
session_start();
if(isset($_POST['submit']))
{
  $username = $_POST['username'];
  $password = $_POST['password'];

  if(!empty($_POST["submit"]))
     {
  $loginquery ="SELECT * FROM users WHERE username='$username' && password='".md5($password)."'"; //selecting matching records
  $result=mysqli_query($db, $loginquery); //executing
  $row=mysqli_fetch_array($result);

                        if(is_array($row))
              {
                                    $_SESSION["user_id"] = $row['u_id'];
                   header("refresh:1;url=index.php");
                             }
              else
                {
                                      $message = "Invalid Username or Password!";
                              }
   }


}
?>
```

Sr No.	Functional Test Cases	Type- Invalid/ Valid Test Case
1	Verify if a user will be able to login with a valid username and valid password.	Valid
2	Verify if a user cannot login with a valid username and an invalid password.	Invalid
3	Verify the login page for both, when the field is blank and the Submit button is clicked.	Invalid
4	Verify the messages for invalid login.	Valid
5	Verify if the data in the password field is either visible as asterisk or bullet signs.	Valid
6	Verify if a user is able to login with a new password only after he/she has changed the password.	Valid
7	Verify if the login page allows you to log in simultaneously with different credentials in a different browser.	Valid
8	Verify if the 'Enter' key of the keyboard is working correctly on the login page.	Valid



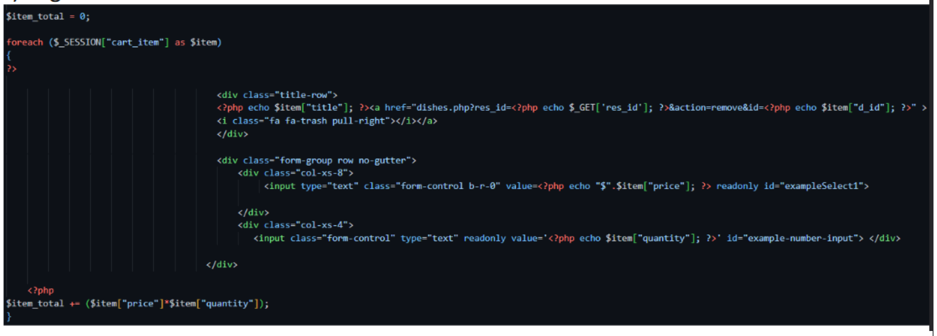
Sr No.	Security test cases	Type- Invalid/ Valid Test Case
1	Verify if a user cannot enter the characters more than the specified range in each field (Username and Password).	Valid
2	Verify if a user cannot enter the characters more than the specified range in each field (Username and Password).	Invalid
3	Verify the login page of the browser that it should not allow you to enter into the system once you log out.	Valid
4	Verify if a user should not be allowed to log in with different credentials from the same browser at the same time.	Valid
5	Verify if a user should be able to login with the same credentials in different browsers at the same time.	Invalid

Boundary testing is a black-box testing technique that software developers often use to check the errors at the boundaries or extreme ends of a given input domain. An input domain comprises all the possible inputs available in a software program. Software developers utilize black-box testing to analyze the behavior of a software program and examine its functionality. The extreme ends in boundary testing might include start to end, lower to upper or minimum to maximum. Instead of focusing only on the center of the data, boundary testing helps detect errors occurring at the boundary values of valid or invalid partitions.

The boundary testing will be applied to create an Account
a) The user’s password count should not exceed a value of 222
b) The minimum value of password count should be 6
c) The maximum value of password count should be 222

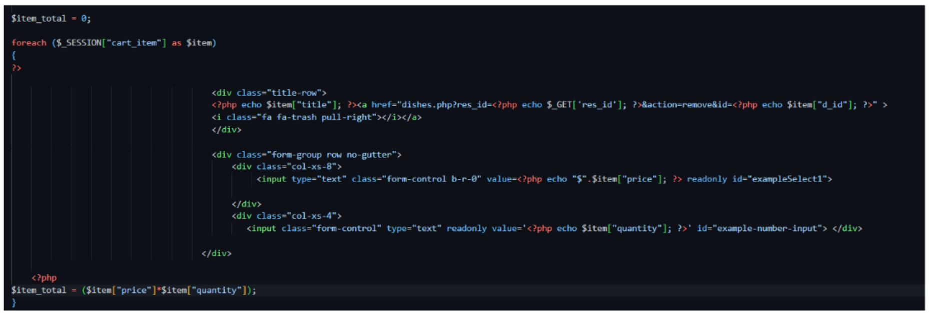
Mutation testing is a fault-based testing technique where variations of a software program are subjected to the test dataset. This is done to determine the effectiveness of the test set in isolating the deviations.

1)Original:



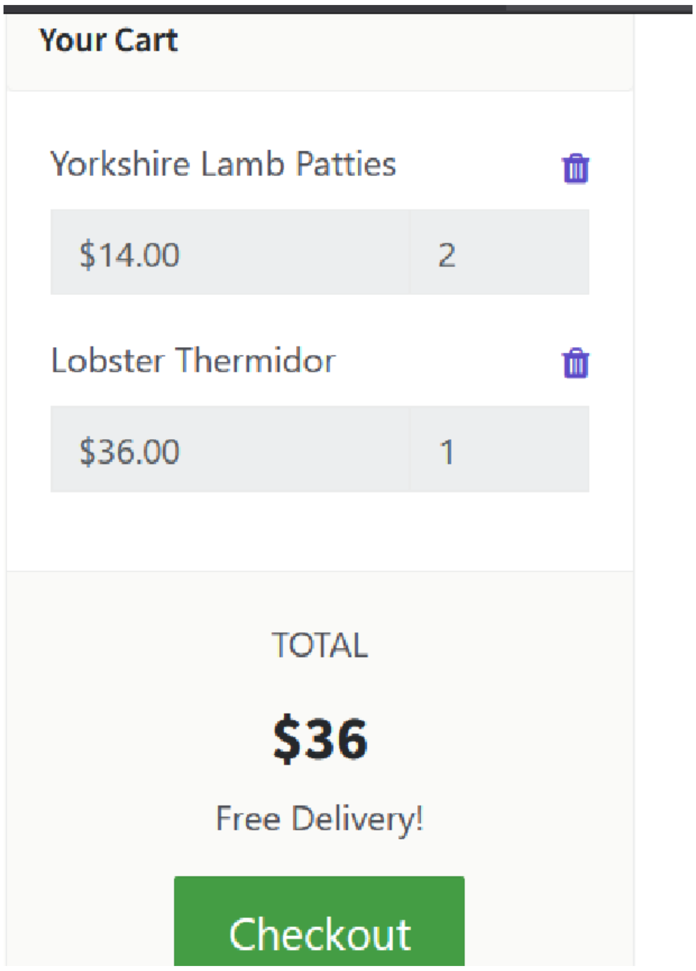
```
$item_total = 0;

foreach ($_SESSION["cart_item"] as $item)
{
?>

                    <div class="title-row">
                    <?php echo $item["title"]; ?><a href="dishes.php?res_id=<?php echo $_GET['res_id']; ?>&action=remove&id=<?php echo $item["d_id"]; ?>" >
                    <i class="fa fa-trash pull-right"></i></a>
                    </div>

                    <div class="form-group row no-gutter">
                        <div class="col-xs-8">
                            <input type="text" class="form-control b-r-0" value=<?php echo "$".$item["price"]; ?> readonly id="exampleSelect1">

                        </div>
                        <div class="col-xs-4">
                          <input class="form-control" type="text" readonly value='<?php echo $item["quantity"]; ?>' id="example-number-input"> </div>

                  </div>

    <?php
$item_total += ($item["price"]*$item["quantity"]);
}
```

Mutant 1:



```
$item_total = 0;

foreach ($_SESSION["cart_item"] as $item)
{
?>

                    <div class="title-row">
                    <?php echo $item["title"]; ?><a href="dishes.php?res_id=<?php echo $_GET['res_id']; ?>&action=remove&id=<?php echo $item["d_id"]; ?>" >
                    <i class="fa fa-trash pull-right"></i></a>
                    </div>

                    <div class="form-group row no-gutter">
                        <div class="col-xs-8">
                            <input type="text" class="form-control b-r-0" value=<?php echo "$".$item["price"]; ?> readonly id="exampleSelect1">

                        </div>
                        <div class="col-xs-4">
                          <input class="form-control" type="text" readonly value='<?php echo $item["quantity"]; ?>' id="example-number-input"> </div>

                  </div>

    <?php
$item_total = ($item["price"]*$item["quantity"]);
}
```

Adding item to the Cart and checking the price


Your Cart

Yorkshire Lamb Patties

$14.00	2

Lobster Thermidor

$36.00	1

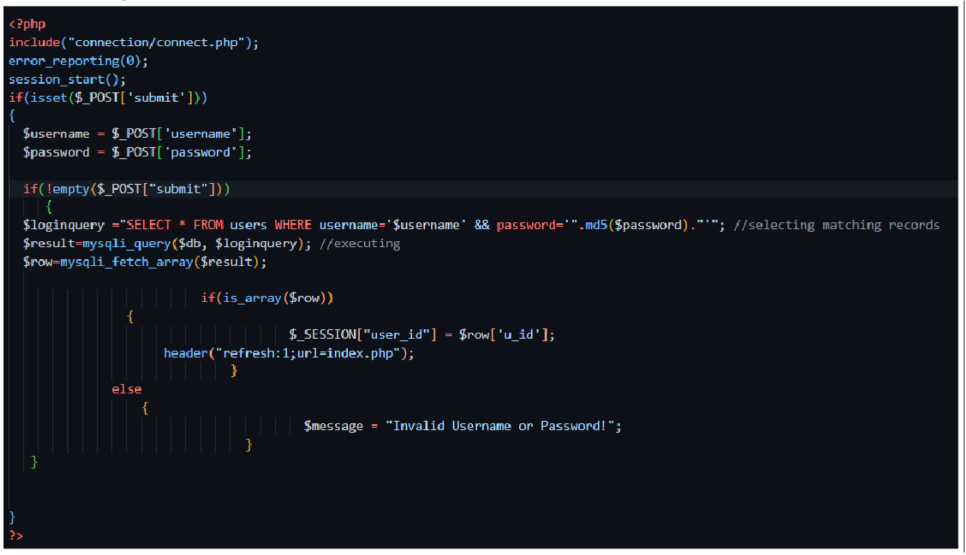
TOTAL

$36

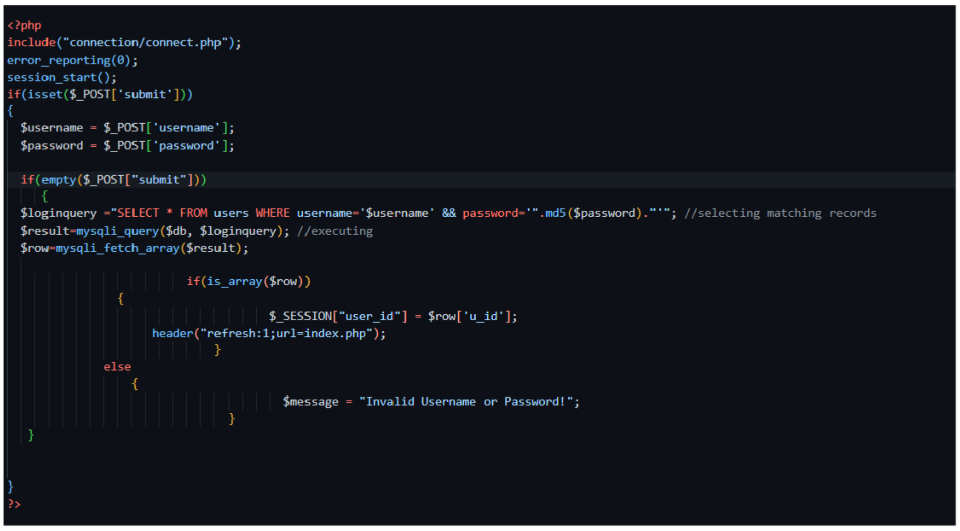
Free Delivery!

As a new item is added the total value does not get added to the original value but
gets replaced

2) Original:

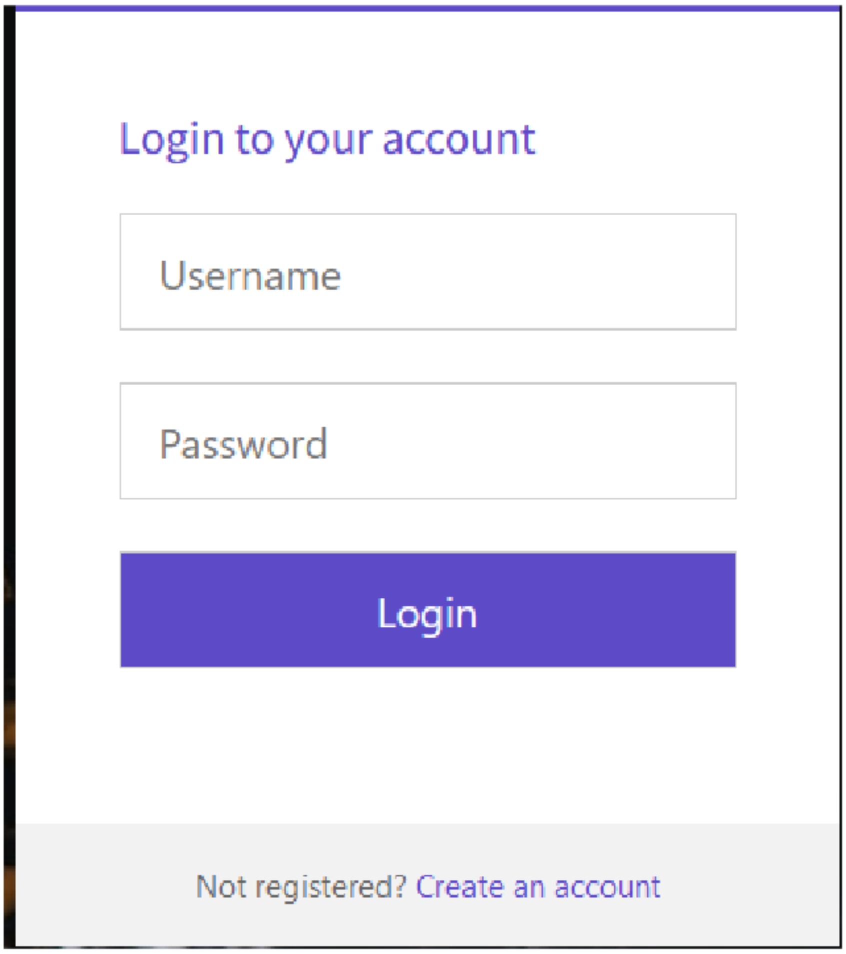

```php
<?php
include("connection/connect.php");
error_reporting(0);
session_start();
if(isset($_POST['submit']))
{
  $username = $_POST['username'];
  $password = $_POST['password'];

  if(!empty($_POST["submit"]))
     {
  $loginquery ="SELECT * FROM users WHERE username='$username' && password='".md5($password)."'"; //selecting matching records
  $result=mysqli_query($db, $loginquery); //executing
  $row=mysqli_fetch_array($result);

                         if(is_array($row))
                {
                                    $_SESSION["user_id"] = $row['u_id'];
                     header("refresh:1;url=index.php");
                              }
              else
                  {
                                      $message = "Invalid Username or Password!";
                                }
   }


}
?>
```

Mutant 2:

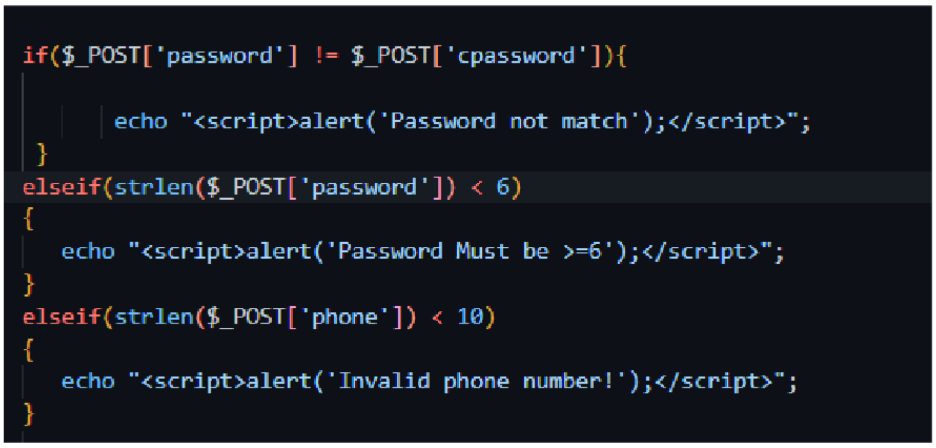


```
<?php
include("connection/connect.php");
error_reporting(0);
session_start();
if(isset($_POST['submit']))
{
  $username = $_POST['username'];
  $password = $_POST['password'];

  if(empty($_POST["submit"]))
     {
  $loginquery ="SELECT * FROM users WHERE username='$username' && password='".md5($password)."'"; //selecting matching records
  $result=mysqli_query($db, $loginquery); //executing
  $row=mysqli_fetch_array($result);

                          if(is_array($row))
                 {
                                 $_SESSION["user_id"] = $row['u_id'];
                     header("refresh:1;url=index.php");
                             }
               else
                   {
                                   $message = "Invalid Username or Password!";
                               }
   }


}
?>
```

As we removed the ! in the if statement. The page does not fetch any user with given username and password

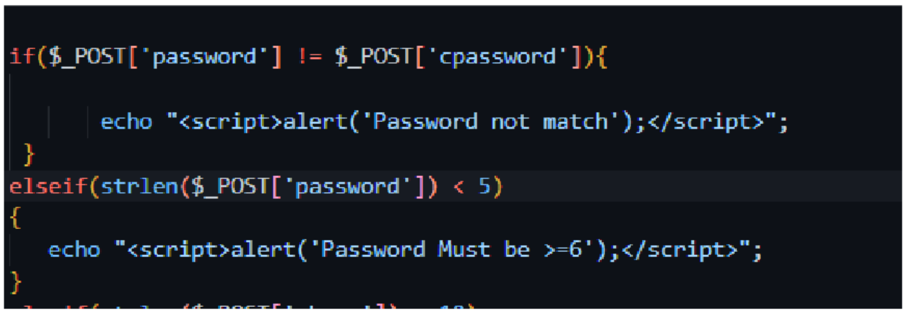


Original:



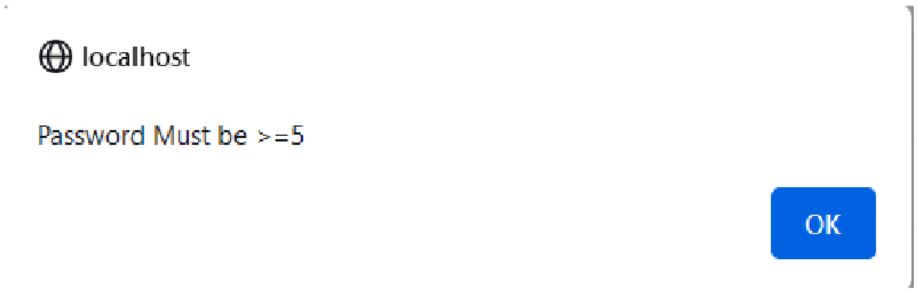
```
if($_POST['password'] != $_POST['cpassword']){

        echo "<script>alert('Password not match');</script>";
 }
elseif(strlen($_POST['password']) < 6)
{
   echo "<script>alert('Password Must be >=6');</script>";
}
elseif(strlen($_POST['phone']) < 10)
{
   echo "<script>alert('Invalid phone number!');</script>";
}
```

Mutant 3:



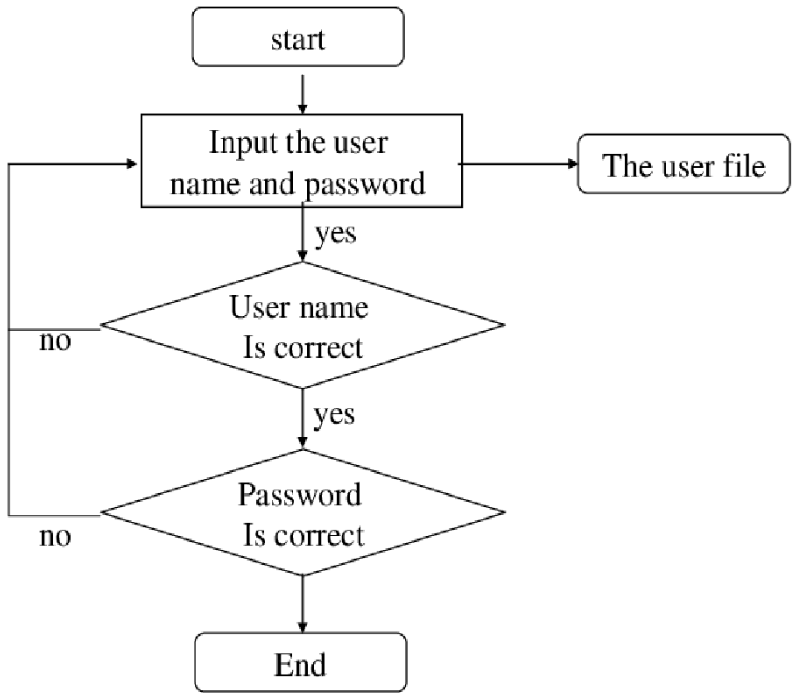
```
if($_POST['password'] != $_POST['cpassword']){

        echo "<script>alert('Password not match');</script>";
 }
elseif(strlen($_POST['password']) < 5)
{
   echo "<script>alert('Password Must be >=6');</script>";
}
```

The password should exceed a count of 5



localhost

Password Must be >=5



M = E – N + 2P

where,

M = cyclomatic complexity

E = the number of edges in the control flow graph N = the number of nodes in the control flow graph P = the number of connected components

E = 7

N = 6

P = 1

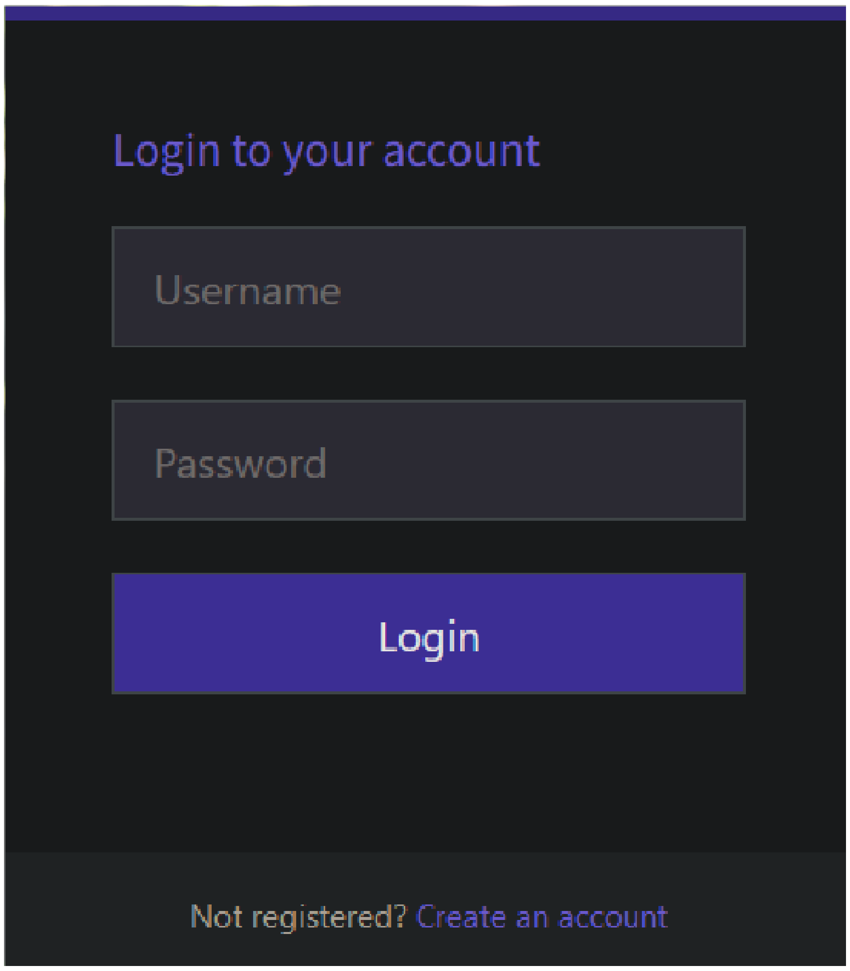
M = 7 - 6 + 2(1)

M = 3

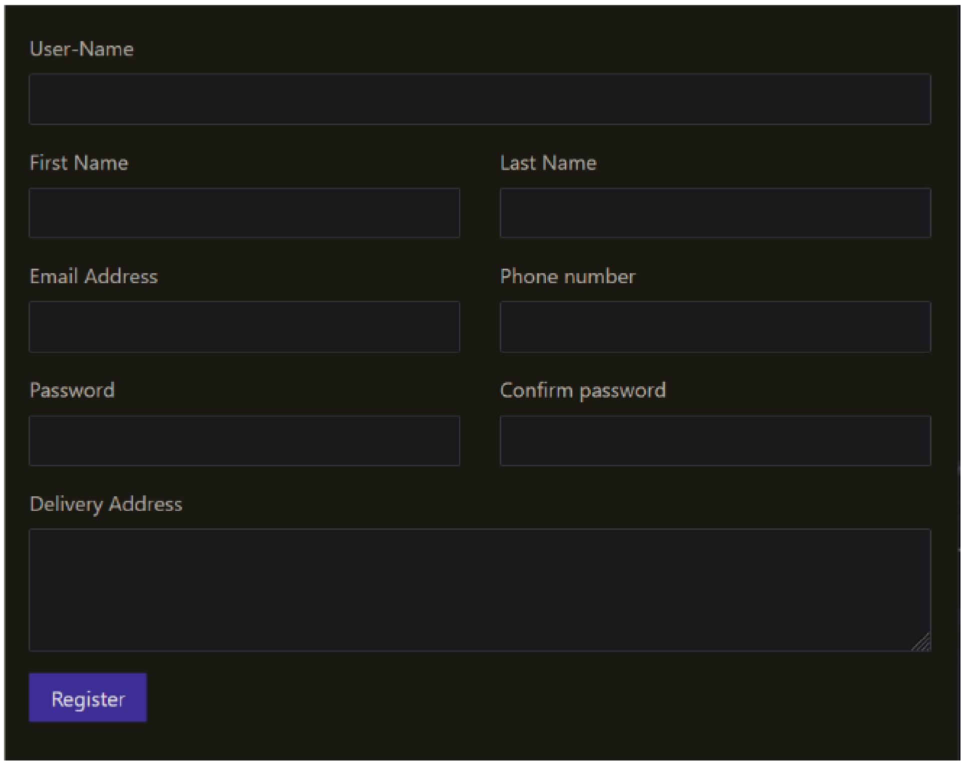
Therefore, the cyclomatic complexity of our above code is 3.

Therefore, the cyclomatic complexity of our above code is 3.

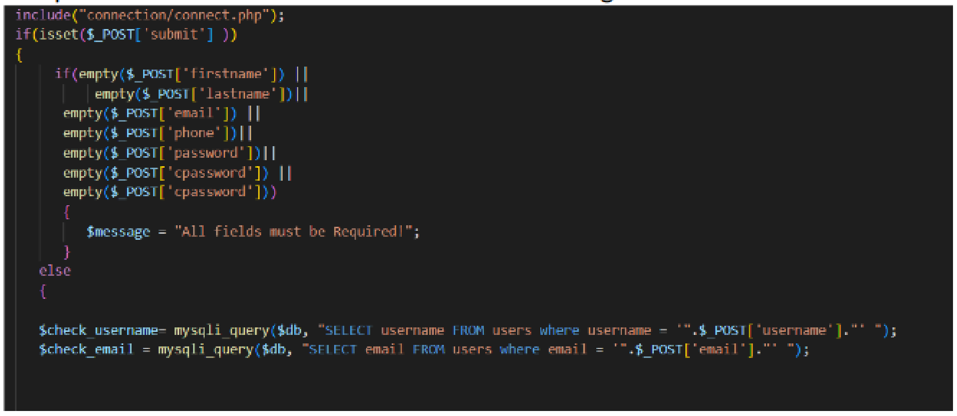
• After understanding the code block, Re-Engineer the code. Ideate how to refactor the code and the portion of the code base you would have to change. Discuss how the new changes would impact the time and space complexity of the project during execution
• After Reverse Engineering and Re-Engineering, the code, perform acceptance testing between the teams.
Software Re-Engineering is the examination and alteration of a system to reconstitute it in a new form. The principle of Re-Engineering when applied to the software development process is called software re-engineering. It affects positively software cost, quality, service to the customer, and speed of delivery. In Software Re-engineering, we are improving the software to make it more efficient and effective



A user can be allowed to enter his username and password to login And also create an account if the account doesn't
exist

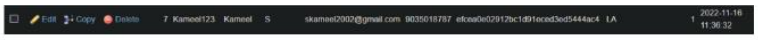


In Create account the user is supposed to enter all the details required to create an account.
The portion of the code that we would like to change:



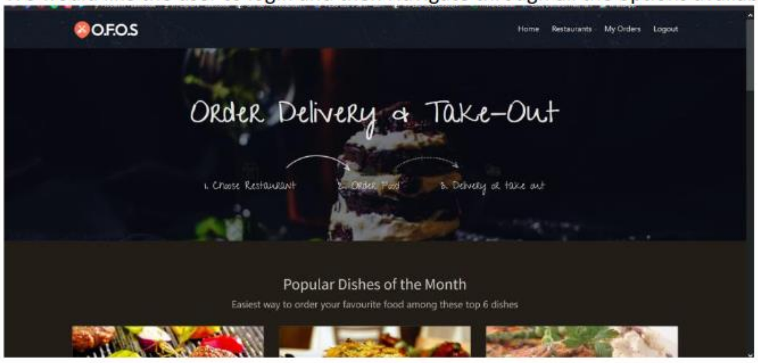
```
include("connection/connect.php");
if(isset($_POST['submit'] ))
{
    if(empty($_POST['firstname']) ||
        empty($_POST['lastname'])||
     empty($_POST['email']) ||
     empty($_POST['phone'])||
     empty($_POST['password'])||
     empty($_POST['cpassword']) ||
     empty($_POST['cpassword']))
     {
        $message = "All fields must be Required!";
     }
   else
   {

   $check_username= mysqli_query($db, "SELECT username FROM users where username = '".$_POST['username']."' ");
   $check_email = mysqli_query($db, "SELECT email FROM users where email = '".$_POST['email']."' ");
```

In Create account the user is supposed to enter all the details required to create an account.
The portion of the code that we would like to change:
All the options can be provided to the user after a Creating Account. The time complexity remains the same but the
space complexity increases due to the introduction of a new feature to improve user experience.



```
☐ Edit  Copy  Delete   7  Kameel123  Kameel  S   skamael2002@gmail.com  9035018787  efcea0e02912bc1d91eced3ed5444ac4  LA   1  2022-11-16 11:36:32
```

We would like the user to login and then Navigate through all the options available on the website


O.F.O.S
Home
Restaurants
My Orders
Logout
Order Delivery & Take-Out
1. Choose Restaurant
2. Order Food
3. Delivery or take out
Popular Dishes of the Month
Easiest way to order your favourite food among these top 6 dishes

As we can see there is no login page. Now the user can only see the options after logging in
As a client, the product meets all the user requirements. We have done user acceptance testing. User acceptance testing is used to determine whether the product is working for the user correctly. Specific requirements which are
quite often used by the customers are primarily picked for the testing purpose. This is also termed as End-User Testing.
The product is tested against the SRS provided by the product manager. No bug is found in the new product. The new
product will have the following features discussed during the acceptance testing:
• Encrypted password
• Input validation

TEST PLAN:

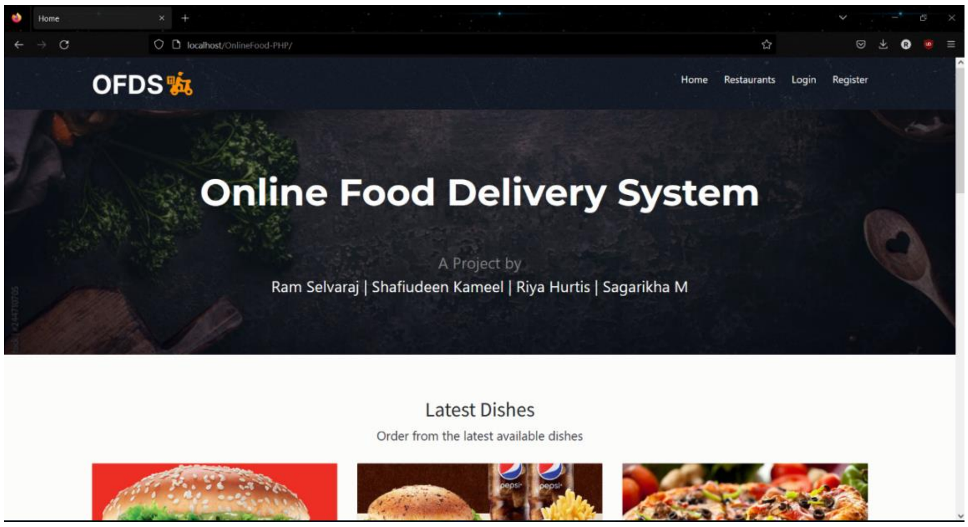
Test Number	Description	Input	Actual Output	Expected Output	Outcome

TC01	Checking Login functionality	admin(user) admin1234 (password)	valid	valid	pass
TC02	Checking Login functionality	admin(user) 1234 (password)	invalid	invalid	pass
TC03	Checking Login functionality	admin1(user) admin1234 (password)	invalid	invalid	pass
TC04	Changing mysql collection name and adding a new admin	kameel(user) kameel123 (password)	Not able to login	Should not be able to login	pass
TC05	Entering password less than 6 characters, invalidates current passwords too	kameel(user) hello (password)	Not able to login, throws error	Throws error	pass
TC06	Changing name of mysql connection from conf to room	kameel(user) kameel123 (password)	Page doesn’t load	Page shouldn’t load as data cant be retrieved	pass
TC07	Updating name of import location of logout function	kameel(user) kameel123 (password)	Logout button doesn’t work	Logout functionality removed	pass
TC08	Getting all necessary details for	Ask user to enter details if not present	Gives warning when details not present	Give user warning when details not present	pass

	booking				

		prompt			
TC10	Send booking info to database	Booking info of user must be visible in database	Show user booking dates on backend	Show user booking dates on backend as table entry	pass
TC11	Testing UI	Clicking all buttons, field inputs, links	All of the fields and buttons work according to functionalit y	No error when a button is pressed or text is input or link is clicked	pass
TC12	Test installation of dependencies	Dependenc ie s using PHPUnit	No installati on error	No installati on error	Pass
TC13	Login Test	Login	Login to system	Login to system	Pass

SCREENSHOTS OF IMPLEMENTATION



Home

localhost/OnlineFood-PHP/

OFDS

Home Restaurants Login Register

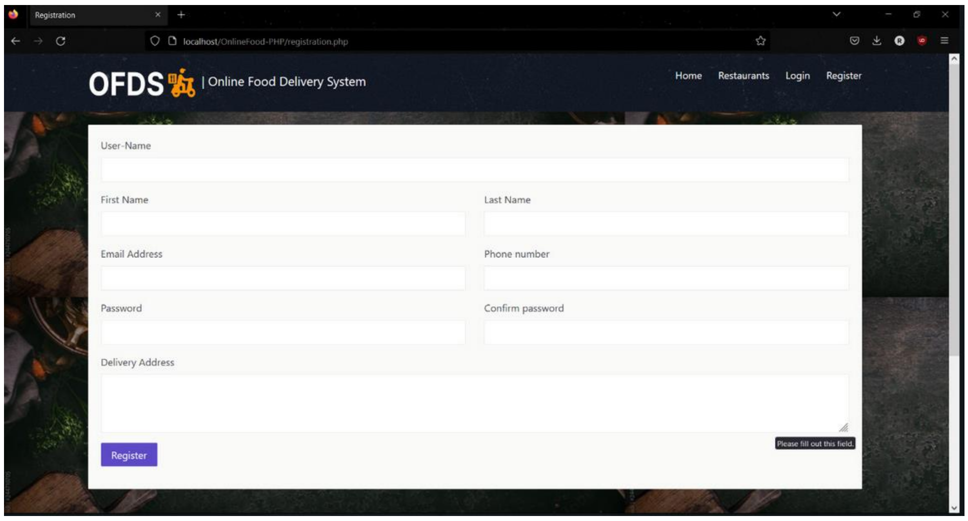
Online Food Delivery System

A Project by

Ram Selvaraj | Shafiudeen Kameel | Riya Hurtis | Sagarikha M

Latest Dishes

Order from the latest available dishes



Registration

localhost/OnlineFood-PHP/registration.php

OFDS | Online Food Delivery System

Home Restaurants Login Register

User-Name

First Name

Last Name

Email Address

Phone number

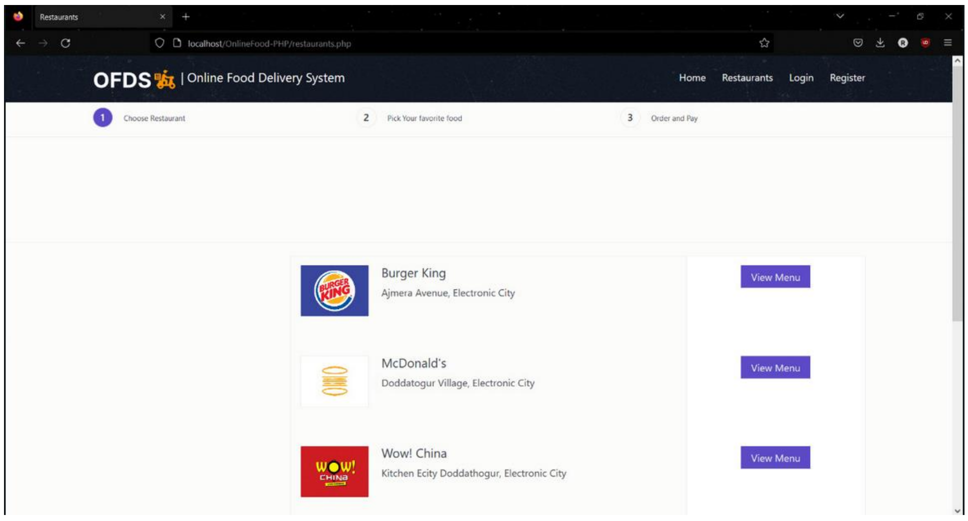
Password

Confirm password

Delivery Address

Please fill out this field.

Register



Restaurants

localhost/OnlineFood-PHP/restaurants.php

OFDS | Online Food Delivery System

Home Restaurants Login Register

1 Choose Restaurant

2 Pick Your favorite food

3 Order and Pay

Burger King

Ajmera Avenue, Electronic City

View Menu

McDonald's

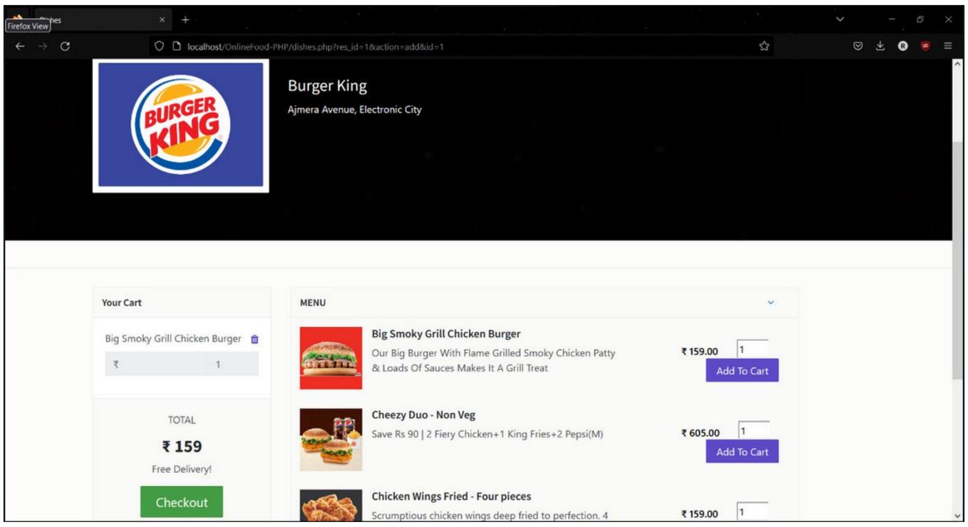
Doddatogur Village, Electronic City

View Menu

Wow! China

Kitchen Ecity Doddathogur, Electronic City

View Menu


Firefox View
Dishes
localhost/OnlineFood-PHP/dishes.php?res_id=1&action=add&id=1

Burger King

Ajmera Avenue, Electronic City

Your Cart

Big Smoky Grill Chicken Burger

₹ 1

TOTAL

₹ 159

Free Delivery!

Checkout

MENU

Big Smoky Grill Chicken Burger

Our Big Burger With Flame Grilled Smoky Chicken Patty & Loads Of Sauces Makes It A Grill Treat

₹ 159.00 1 Add To Cart

Cheezy Duo - Non Veg

Save Rs 90 | 2 Fiery Chicken+1 King Fries+2 Pepsi(M)

₹ 605.00 1 Add To Cart

Chicken Wings Fried - Four pieces

Scrumptious chicken wings deep fried to perfection. 4

₹ 159.00 1