HUNGRY HOSTEL

PRESENTED BY:-
SHRADDHA AKANKSHYA-1405428
SHUBHJEET SHEKHAR-1405432
SIDDHARTH MAHAPATRA-1405433

SUBMITTED TO:-
ARUP ABHINNA ACHARYA

SCHOOL OF COMPUTER ENGINEERING
KIIT UNIVERSITY
BHUBANESWAR - 24

SCHOOL OF COMPUTER ENGINEERING

KIIT UNIVERSITY

BHUBANESWAR – 24

CERTIFICATE

This is to certify that the project entitled “HUNGRY HOSTEL” is being carried out by Shraddha Akankshya(1405428) , Shubhjeet Shekhar (1405432) and Siddharth Mahaptra (1405433) , as a curriculum project for 6th semester in Computer Science and Engineering at School of Computer Engineering, KIIT University , Bhubaneswar during the academic year 2016-2017 under your supervision. The matter embodied in this project is original and has not been submitted for the award of any other degree.

Signature of dean	**Signature of project mentor**
(Dr. S. Mishra)	**(Dr. Arup A. Acharya)**

ACKNOWLEDGEMENT

Apart from our efforts, the success of this project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take immense pleasure in thanking and warmly acknowledging the continuous encouragement, invaluable supervision, timely suggestions and inspired guidanc e offered by our project mentor. Dr. Arup A. Acharya, Assistant Professor , School of Computer Engineering, KIIT University, in bringing this report to a successful completion.

We also express our sincere thanks to all our friends who have patiently extended all sorts of help for accomplishing this undertaking. Also, we would like to express our heartfelt thanks to each of our beloved parents for their blessings, for their help and wishes for the successful completion of this project.

Finally we extend our gratefulness to one and all that are directly or indirectly involved in the successful completion of this project work.

Shraddha Akankshya – 1405428

Shubhjeet Shekhar – 1405432

Siddharth Mahapatra - 1405433

ABSTRACT

Hungry Hostel aims to provide an online interface, to reduce the food problem faced by the college students who stay in hostel, by providing them food cooked by the home makers who stay close to their location. The cook and the student are the key characters for the entire application.

The job of the cook, is to post the name of the food items he/she has prepared for that day and the cook can also view the food items requested by the students in his/her news feed. If he/she can prepare those items then they can accept the request of the student and a notification will be sent to the student with the cook's location as the pickup address.

The job of the student, is to request for the food items he/she wants to have and the student can also view the food items that has already been prepared by the cook nearby his/her location in his/her newsfeed. The student can either request for his own food item or can select the food items present in his/her newsfeed. If the cook accepts his/her request , then he/she will be notified the pickup location where he/she can go and fetch the food and pay the cook.

By this, both the homemaker and the student will be benefitted. The cook will get a source of extra income without putting much of effort and the student will get to eat home - made food with paying minimal amount.

Table of contents

1. INTRODUCTION

1.1 METHODOLOGY

WATERFALL MODEL

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

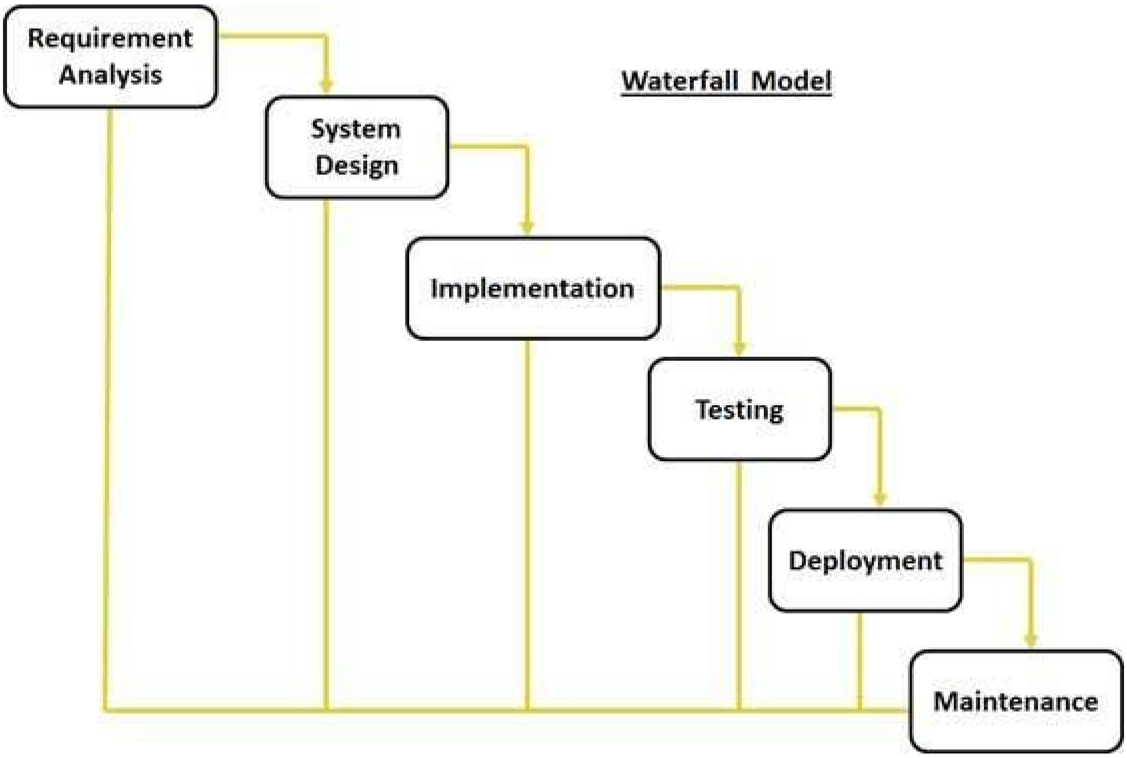
Waterfall model is the earliest SDLC approach that was used for software development .

The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap.

Waterfall Model design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

Following is a diagrammatic representation of different phases of waterfall model.



The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

- **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.

- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

Waterfall Model Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.

- Product definition is stable.

- Technology is understood and is not dynamic.

- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model Pros & Cons

Advantage

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Disadvantage

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

1.2 PURPOSE

As college students, in a city/state away from home, one of the major adversities we face on a day to day basis is a deep craving for homemade food.Food that tastes just like home. Food that has been cooked and served with care.Something unlike the expensive dishes and delicacies that are served at restaurants and posh kitchens around us along with a hard pocket pinch.

Here, the student can get home-made food by spending relatively less amount of money, which not only be affordable but also healthy. And the cook on the other hand, will get a source of income by investing negligible amount of money and effort.

1.3 SCOPE

This webpage would mainly comprise two sections:

1) COOK SECTION:

This section would be made highly secure by passwords and could be accessed by cook only. This section would comprise information required by cook for cooking the food items.

2) CUSTOMER SECTION:

This section would comprise all the basic facilitate of the site and would not be password protected. It will be accessed by the customer i.e student.

LINKING: All the food items requested and accepted and related data to it would be stored in a common database.

- The customer and cook can communicate with each other through notifications.

After the cook accepts the request of the customer, the notification will provide the details of the cook to the customer and vice versa.

- The cook can see the food items requested by the customer and he/she can also submit the food items he/she is going to prepare or has already prepared.
- The customer can see the food items that are already available in the market and also can submit the food items, he/she wants with the number of headcounts.
- Both the cook and the customer will be registered users provided with unique user ID and password.
- The cook and the user and access their respective webpage by providing their user ID and password.

1.4 TOOLS USED

STARUML:

A sophisticated software modeler. It can create Entity-Relationship Diagrams (ERD). ERD is one of the most frequently used diagram for database modeling. Quick Edit to create elements and relationships at once such as sub-classes, supporting interfaces, etc.

2. OBJECTIVE

The major objectives of this web application are:

a) Solve this craving of our generation by engaging anyone who is staying home and is the master chef of their respective kitchens with little or no appreciation.

b) Provide quality assurance and availability of food from a kitchen just like your mother's, at lower costs and at hours when the restaurants around you won't.

c) Provide a platform for them to work indoors and provide a regular and channelized source of income using their ingrained skillset with opportunities at their doorstep.

d) Our initiative wants to help every such person (especially women) who have been held back by social taboos which bar them from earning because leaving the house is highly disgraced by their societies.

e) Provide a source of income through the smartphone in one's hands thereby increasing the technological usage and know how among homemakers.

3. PROJECT PLANNING

During the planning of this project, different housewives, students, and the working class were interviewed and then the result was analysed to determine the functional and non-functional requirements and the constraints.

3.1 QUESTIONNAIRE FOR REQUIREMENT ANALYSIS

Q: How often do you buy outside food

(a) Almost everyday

(b) Half the month

(c) 5-10 days a month

(d) Rarely

Q: Rate your hostel mess food on a scale of 7

Q: Is it affordable to regularly purchase outside food? How budget friendly on a scale of 10?

Q: Would you purchase home-made food, if available around you?

(a) Everyday

(b) Most of the days

(c) Sometimes, for change of taste

(d) No, that's boring.

Q: What do you think about an Application that can connect you to Home Chef(s) enabling you to buy home cooked food on a regular basis?

3.2 FUNCTIONAL REQUIREMENT

1. Registration

Input – User details for a/c creation

Output – Account created

Constraints – Details entered into all fields in correct format

2. Log In

Input – ID & Password

Output – Directed to Main Page after Verification

Constraints – Filling both fields mandatory

3. Placing Order

Input – Customer selected/customised order details

Output – Notification to Customer & Addition of "New Order" to Home chef's feed

Constraints – Minimum >=1 order has to be placed

4. Accept Order

Input – Order Selection by Home chef

Output – Notification to Customer & Home chef

Constraint – Has to select >=1 order

5. What’s Cooking

Input – Dish details entered my Home chef

Output – Dish added to Feed of Customer

Constraint – Entering dish details is mandatory

6. Payment

Input – Customer pays using the Payment Portal

Output – Pay added to Home chef’s wallet

Constraint – Exact amount to be paid

7. Rating

Input – Customer’s rating on 5 stars

Output – Rating submission

Constraint – Rating field filled with non-negative number b/w 0 & 5

3.3 NON-FUNCTIONAL REQUIREMENTS

-24x7 availability :

The application page of Hungry Hostel would be available 24x7 so that customer could contact the home chef and order their food any time as per their ease.

- Performance Requirements:

Performance of the system depends on the response time and the speed of the data submission. The response time of the system is direct and the application is real-time. System should have a fast response time which depends on the efficiency of the implemented algorithm.

The first version of the system will have limited file submission speed; hence there will be no need for a large network. However, it may grow depending on the increase in usage.

- Safety Requirements:

  System has to check If HTML content is syntactically well formed. If another server is not running on the same port which might not allow Apache-MySQL server to run. If web forms with the services processing form input are consistent. Referential integrity of hyperlink in both static and dynamically generated content. Statically safe binding of code of session operations to variables defined with session scope.

- Security Requirements:

For security of the system the technique known as database replication should be used so that all important data is kept safe. In case of crash, the system should be able to backup and recover the data.

- Software Quality Attribute:

The system will have a simple and user friendly graphical interface. Users will be able to understand and use all the features of the website easily.

- Flexible service based architecture will be highly desirable for future extension.

3.4 CONSTRAINTS

- The application will be available only in English.

- The user must sign up to request or accept any orders.

- Only registered users can utilize some services of the application.

- Identity proof submission is mandatory to create a profile.

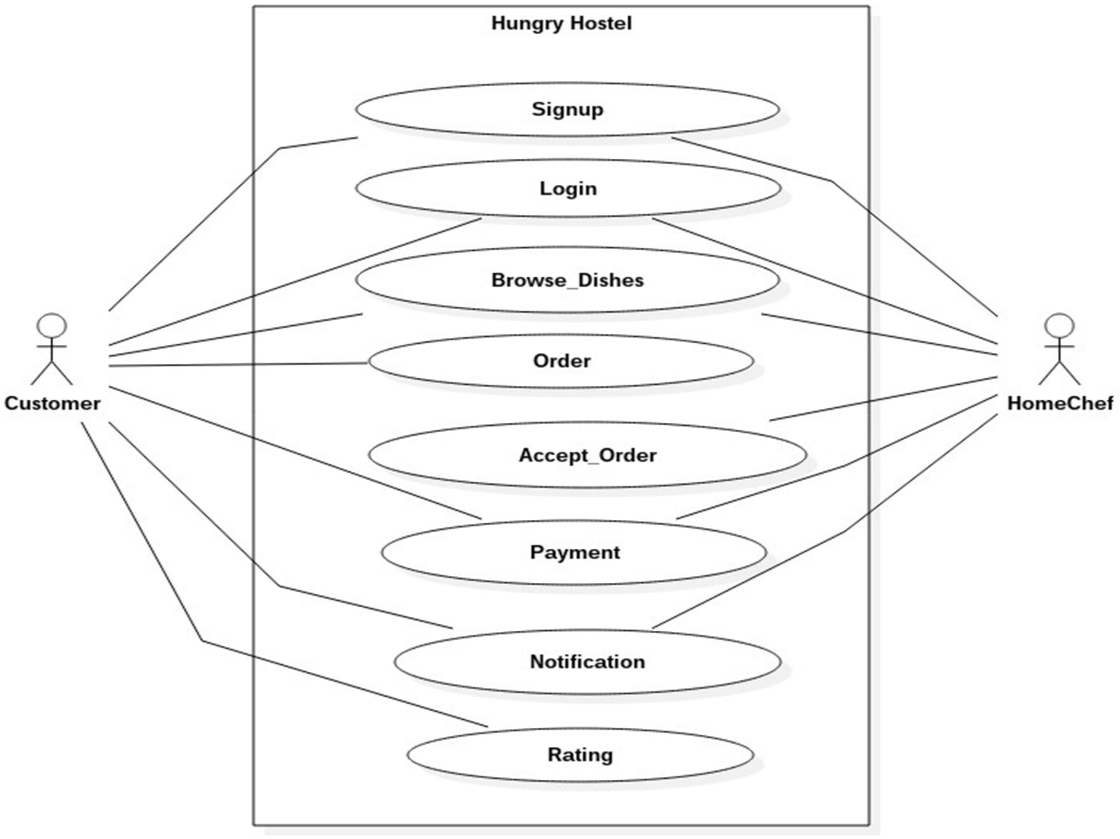
- Users who have the knowledge of accessing net facilities can only utilize the offered services.

4. DESIGN

4.1 UML DESIGN

4.1.1 USE-CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

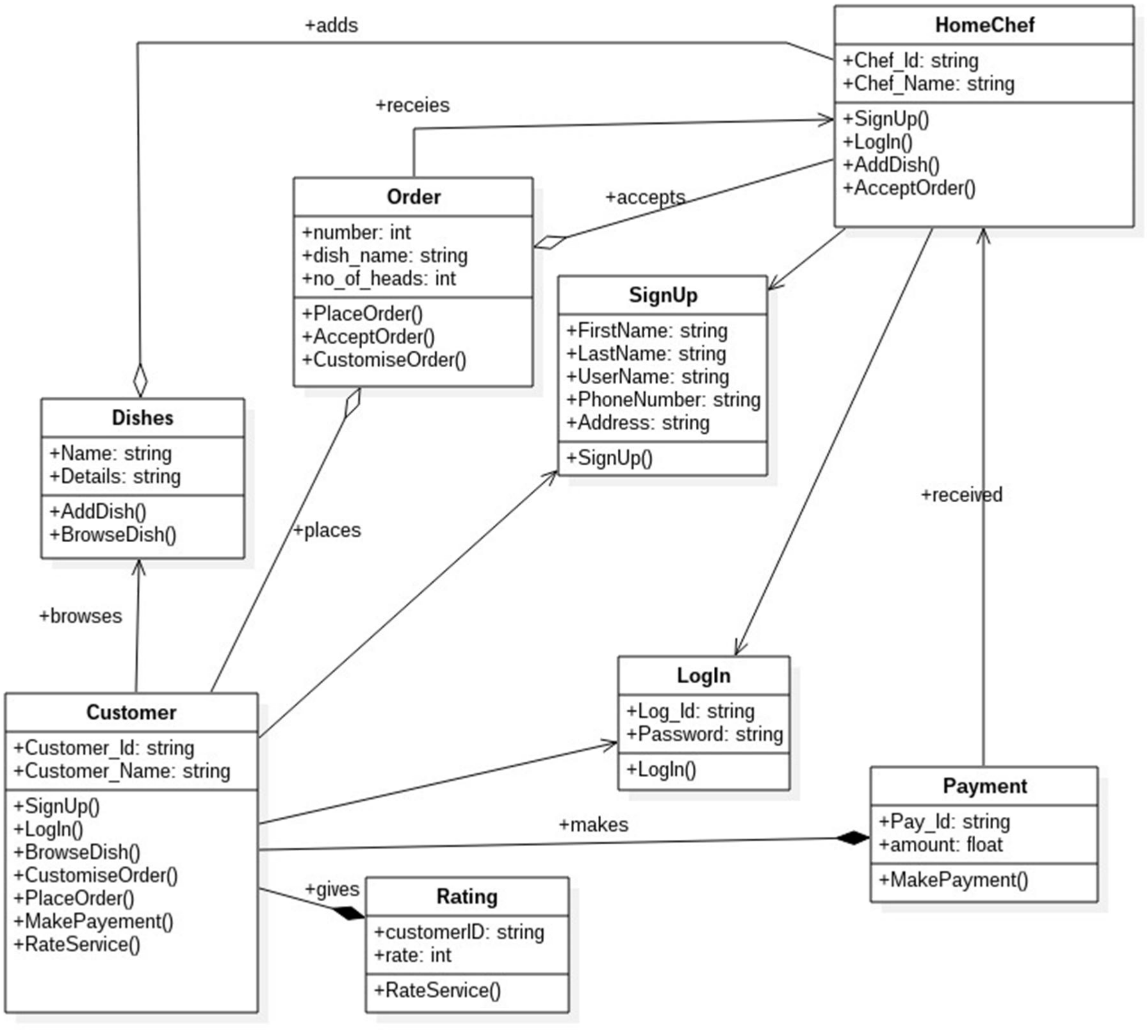


USE-CASE REPORT:-

USE CASE	DESCRIPTION
SIGN UP	Every user can sign up to create their profile.
LOGIN	Allows registered user to login to their respective accounts.
BROWSE DISHES	Home chef browses through the dishes requested by the customer and the customer browses through the dishes already cooked by the chef.
ORDER	Order is placed by the customer.
ACCEPT ORDER	Order is accepted by the home chef.
PAYMENT	Payment is done by the customer to the home chef.
NOTIFICATION	It is sent to the home chef when the customer requests for any dish and it is sent to the customer when the home chef accepts his/her request.
RATING	It is given by the customer to the home chef based on his experience.

4.1.2 CLASS DIAGRAM

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

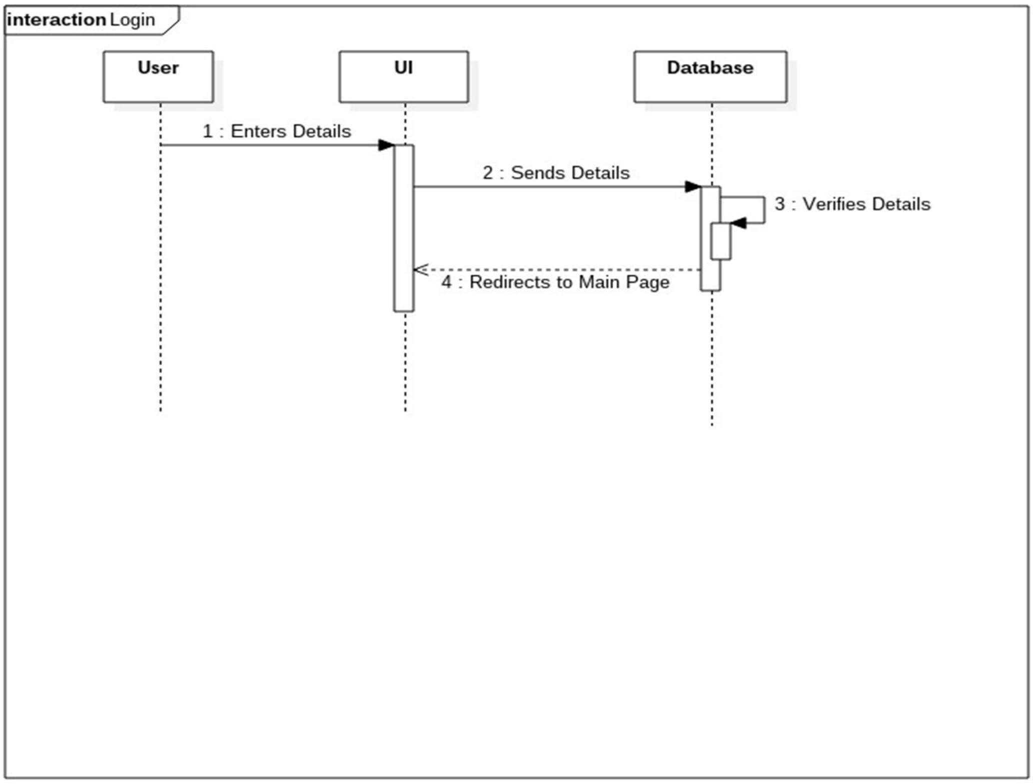


4.1.3 SEQUENCE DIAGRAM

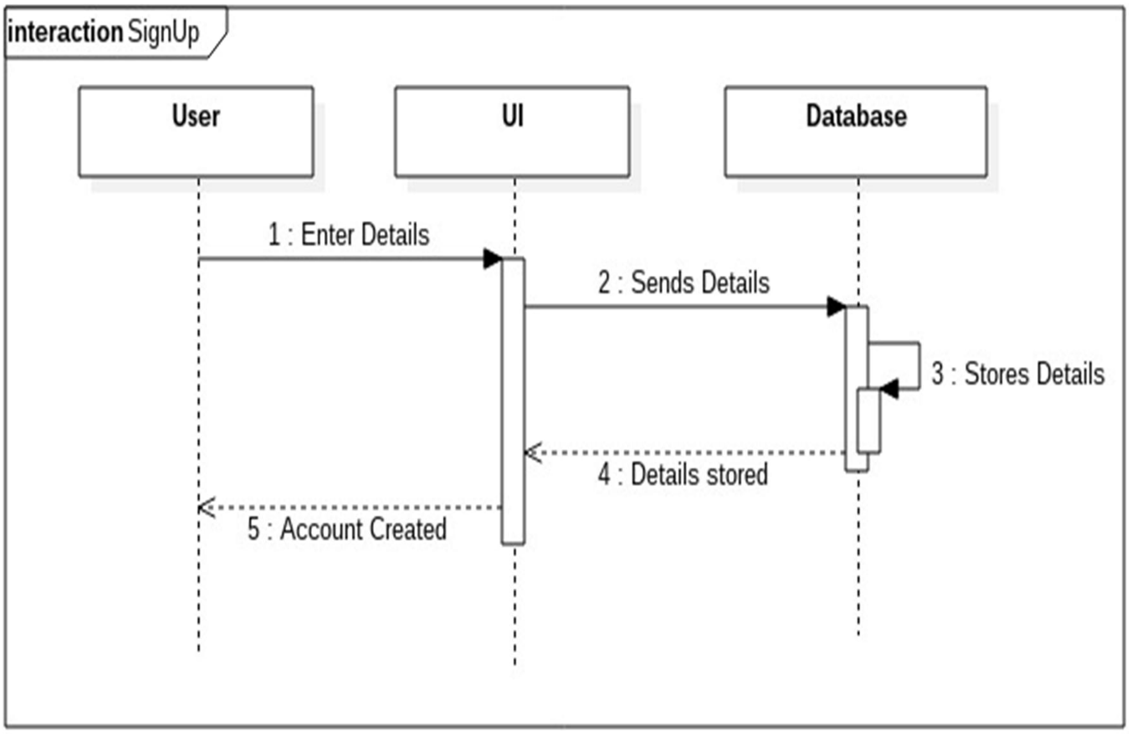
A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

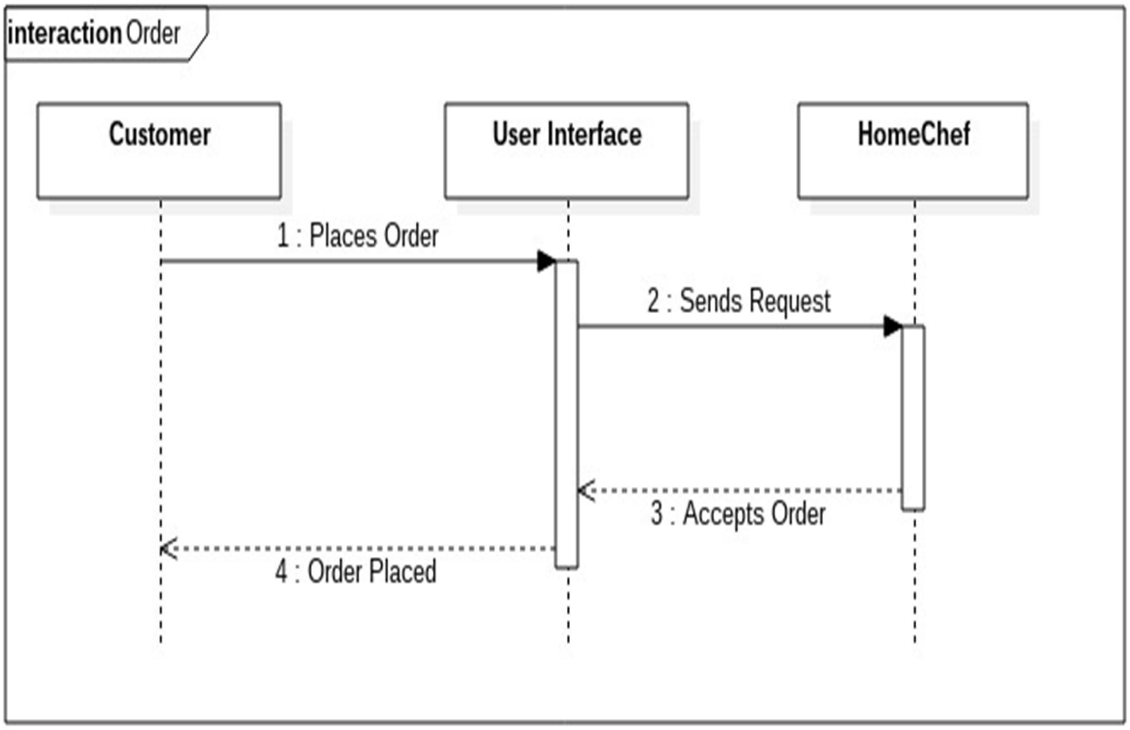
1) Sequence diagram for “login” use-case:



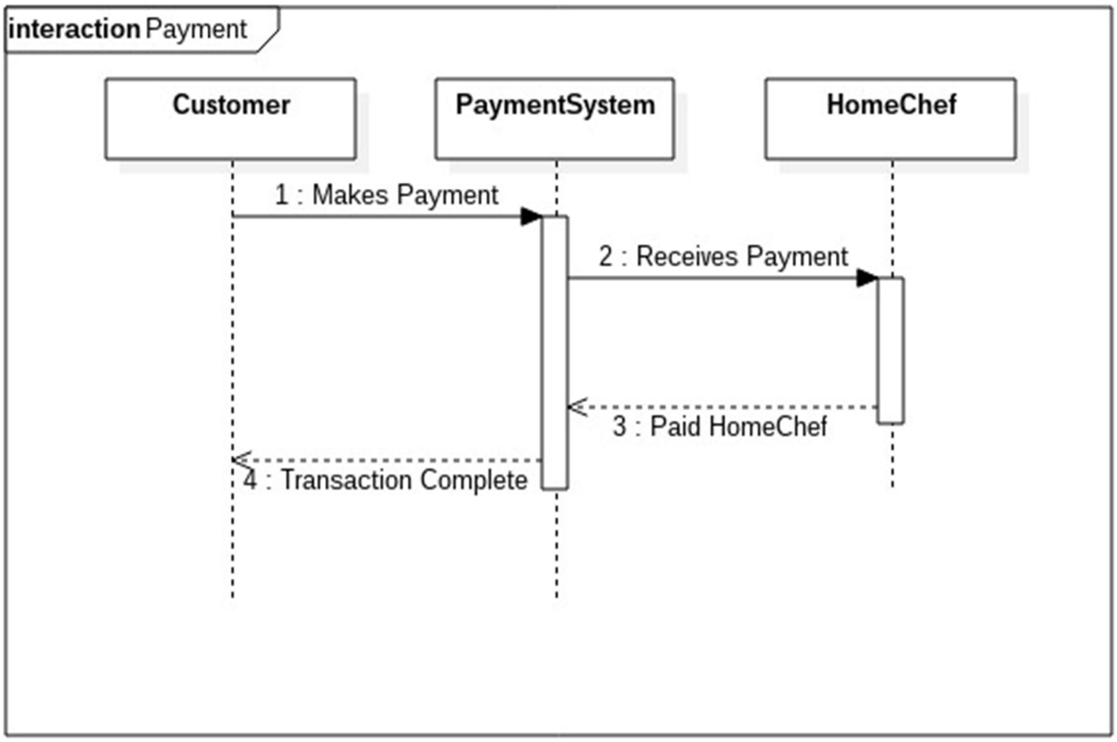
2)Sequence diagram for “sign up” use-case:



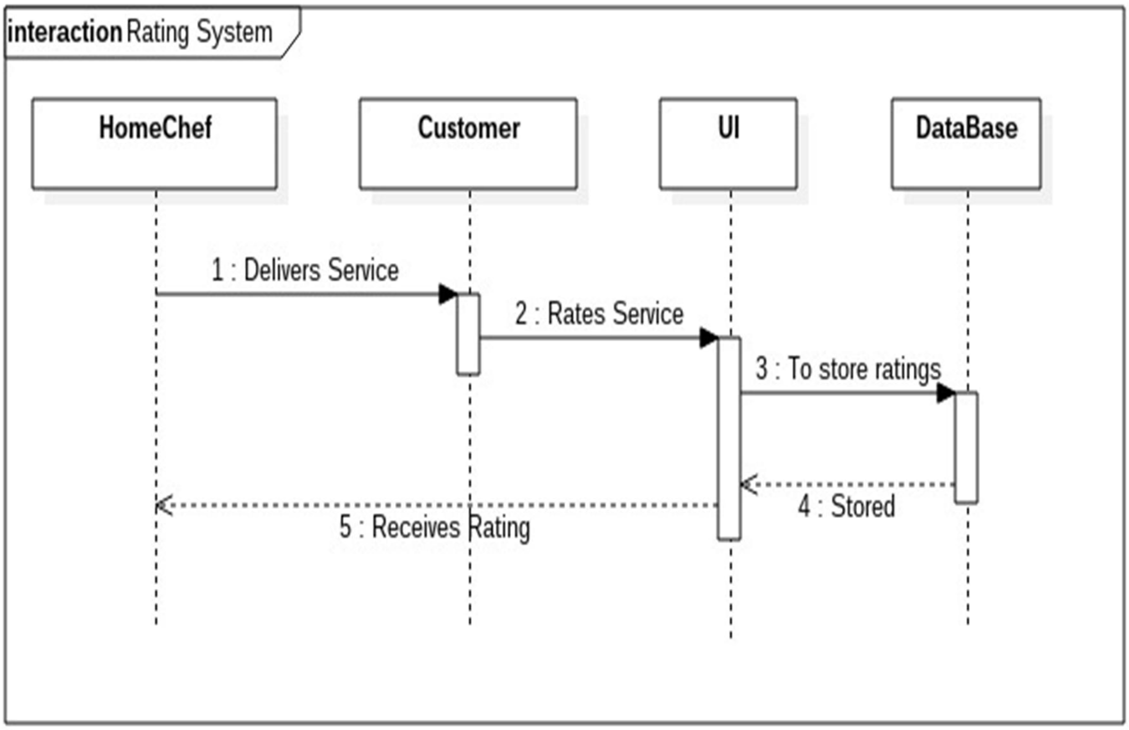
3) Sequence diagram for "order" use-case:



4) Sequence diagram for “payment” use-case:



5) Sequence diagram for "rating" use-case:



4.1.4) ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Activity diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. They may be used to detail situations where parallel processing may occur in the execution of some activities.

Activity diagrams are useful for business modelling where they are used for detailing the processes involved in business activities.

The following sections describe the elements that constitute an activity diagram.

Actions:

An action represents a single step within an activity. Actions are denoted by round cornered rectangles.

Control Flow:

A control flow shows the flow of control from one action to the next. Its notation is a line with an arrowhead.

Initial Node:

An initial or start node is depicted by a large black spot, as shown below.

Final Node:

There are two types of final node: activity and flow final nodes. The activity final node is depicted as a circle with a dot inside.

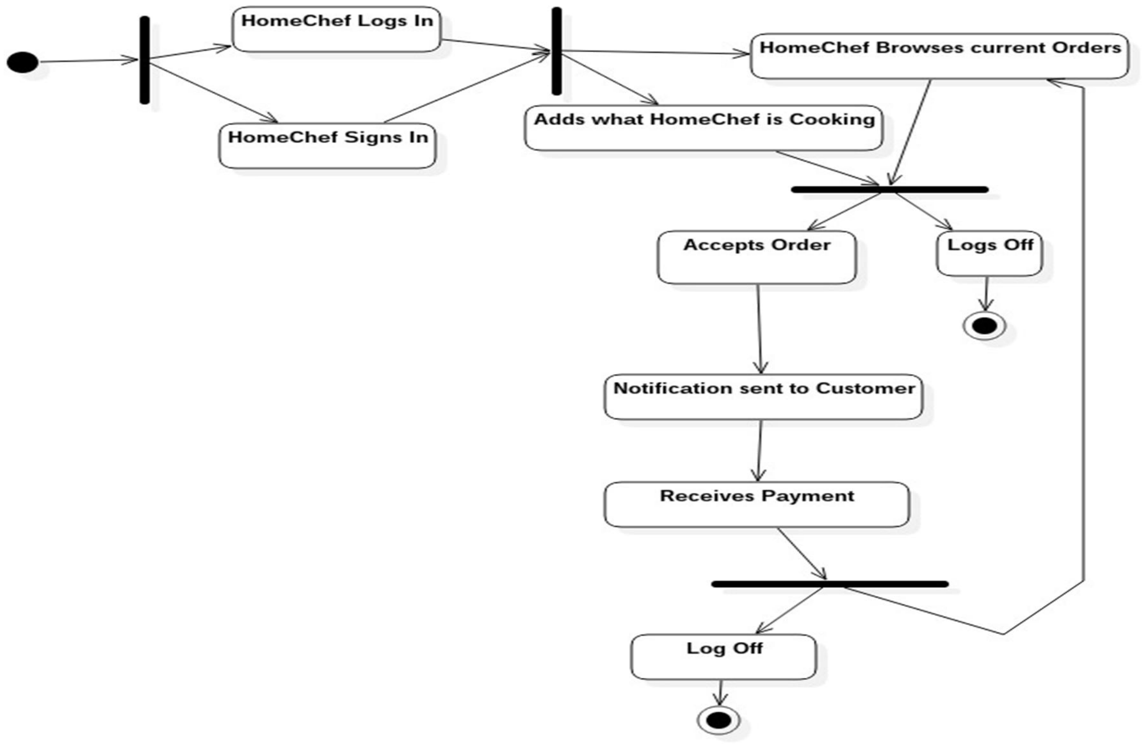
Decision and Merge Nodes:

Decision nodes and merge nodes have the same notation: a diamond shape. They can both be named. The control flows coming away from a decision node will have guard conditions which will allow control to flow if the guard condition is met.

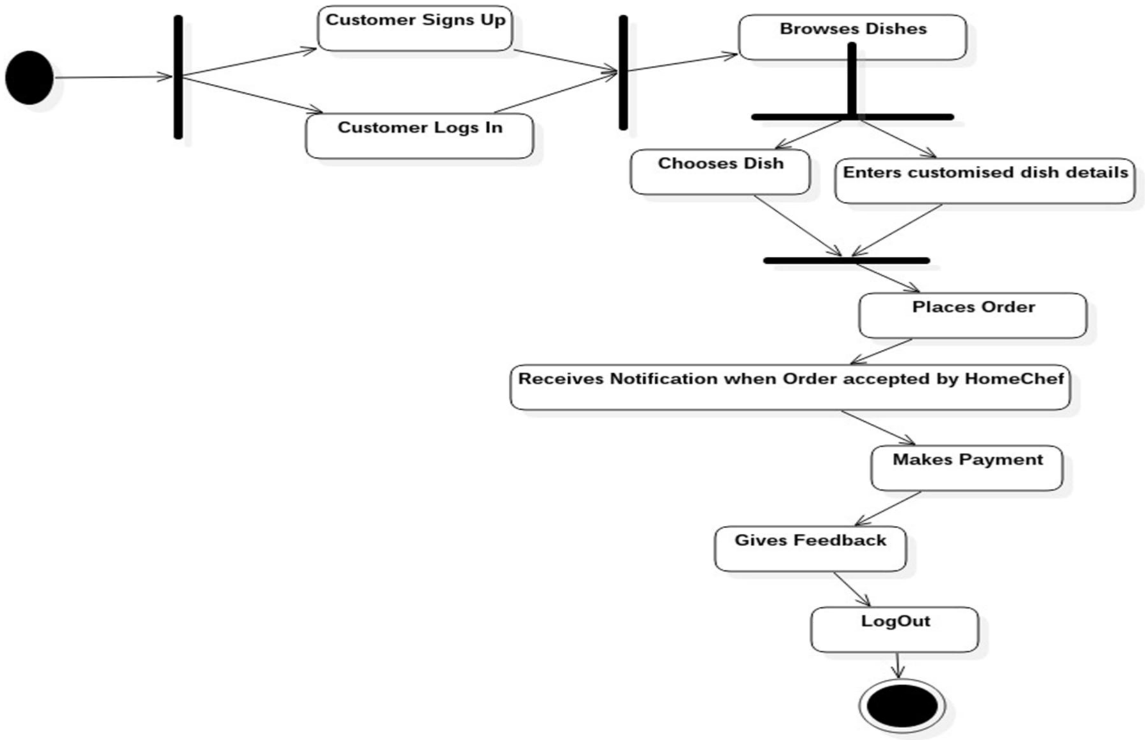
Fork and Join Nodes:

Forks and joins have the same notation: either a horizontal or vertical bar (the orientation is dependent on whether the control flow is running left to right or top to bottom). They indicate the start and end of concurrent threads of control.

1) HOME CHEF:


HomeChef Logs In
HomeChef Signs In
HomeChef Browses current Orders
Adds what HomeChef is Cooking
Accepts Order
Logs Off
Notification sent to Customer
Receives Payment
Log Off

2) CUSTOMER:

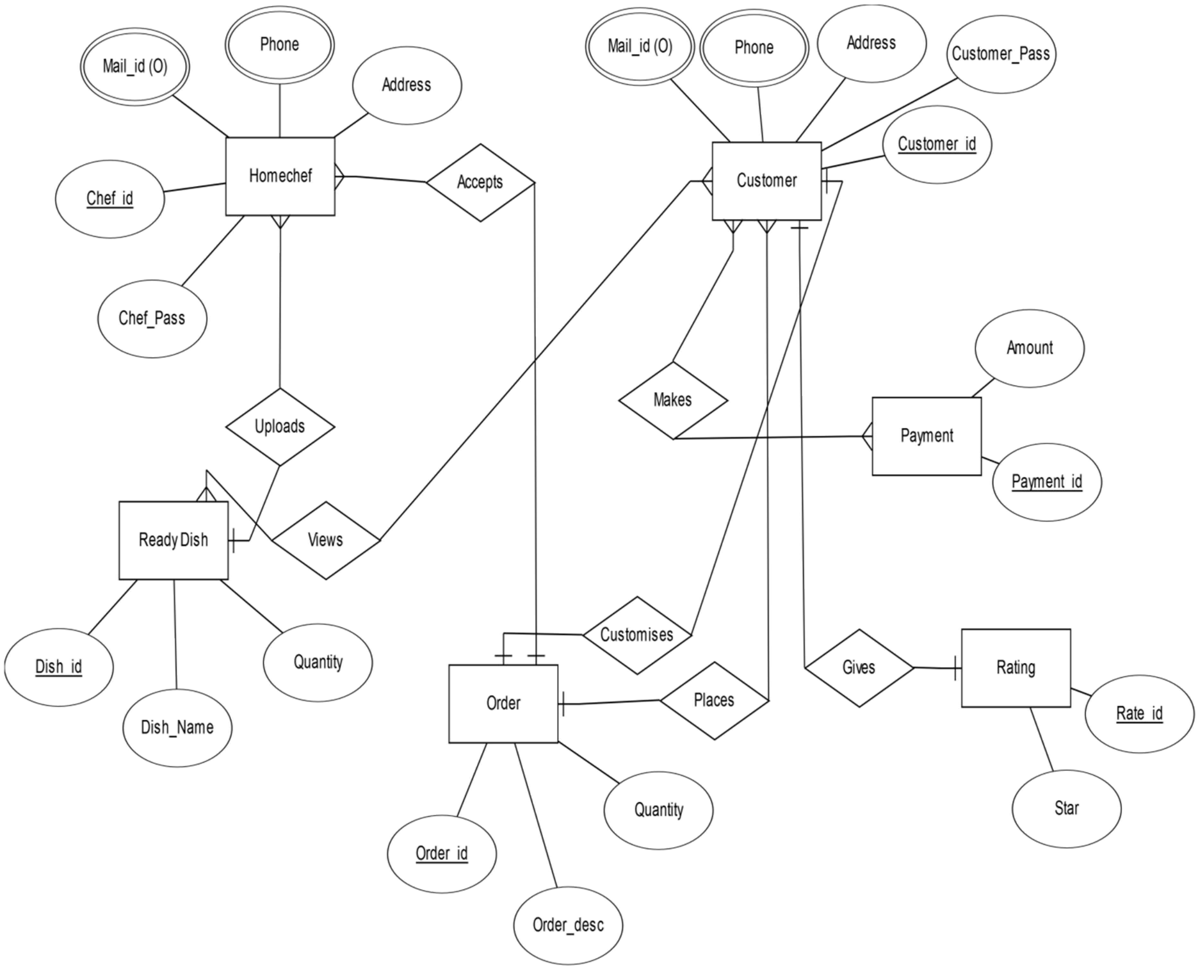

Customer Signs Up
Customer Logs In
Browses Dishes
Chooses Dish
Enters customised dish details
Places Order
Receives Notification when Order accepted by HomeChef
Makes Payment
Gives Feedback
LogOut

4.2) DATABASE DESIGN

ER DIAGRAM

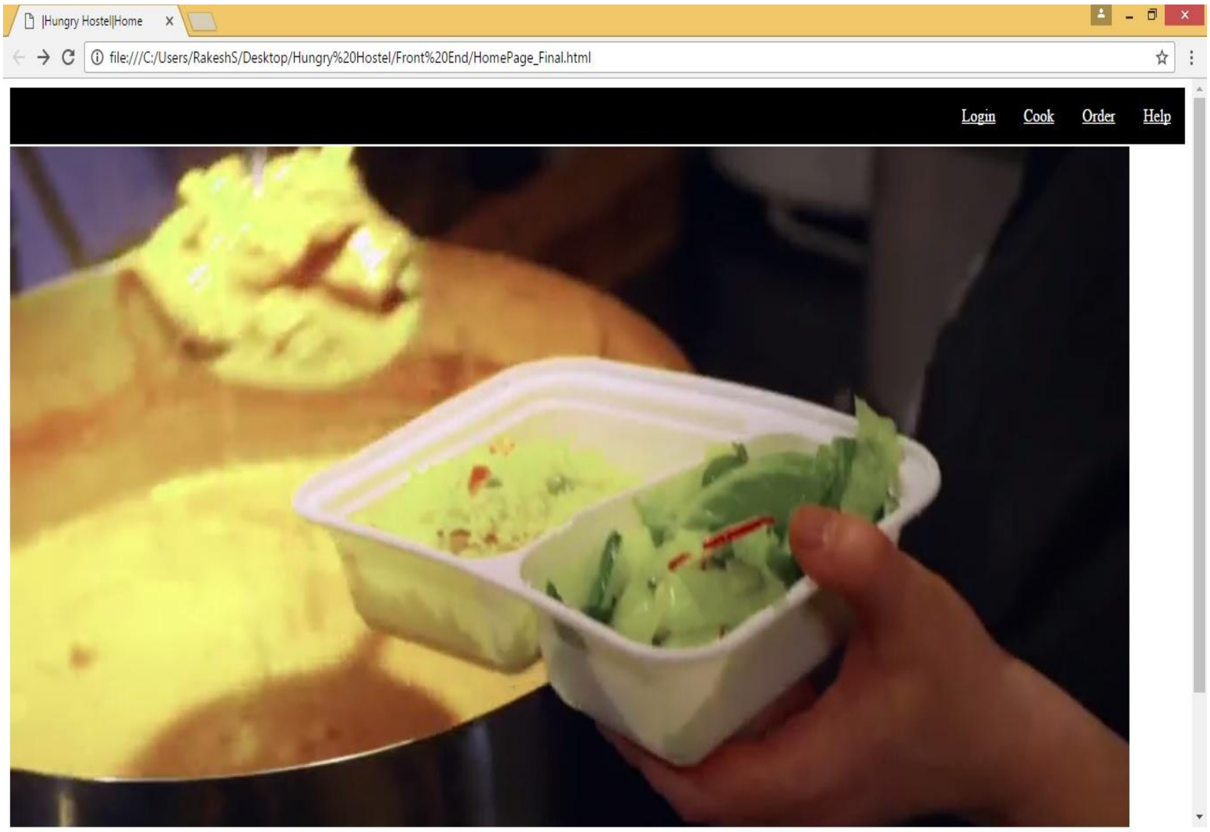
An entity–relationship model is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities and the relationships that can exist among them.

An entity may be defined as a thing capable of an independent existence that can be uniquely identified. Entities are represented by means of their properties, called attributes. The association among entities is called relationship.


Mail_id (O)
Phone
Address
Chef_id
Homechef
Chef_Pass
Accepts
Uploads
Ready Dish
Views
Dish_id
Dish_Name
Quantity
Mail_id (O)
Phone
Address
Customer_Pass
Customer_id
Customer
Makes
Payment
Amount
Payment_id
Customises
Order
Places
Quantity
Order_id
Order_desc
Gives
Rating
Rate_id
Star

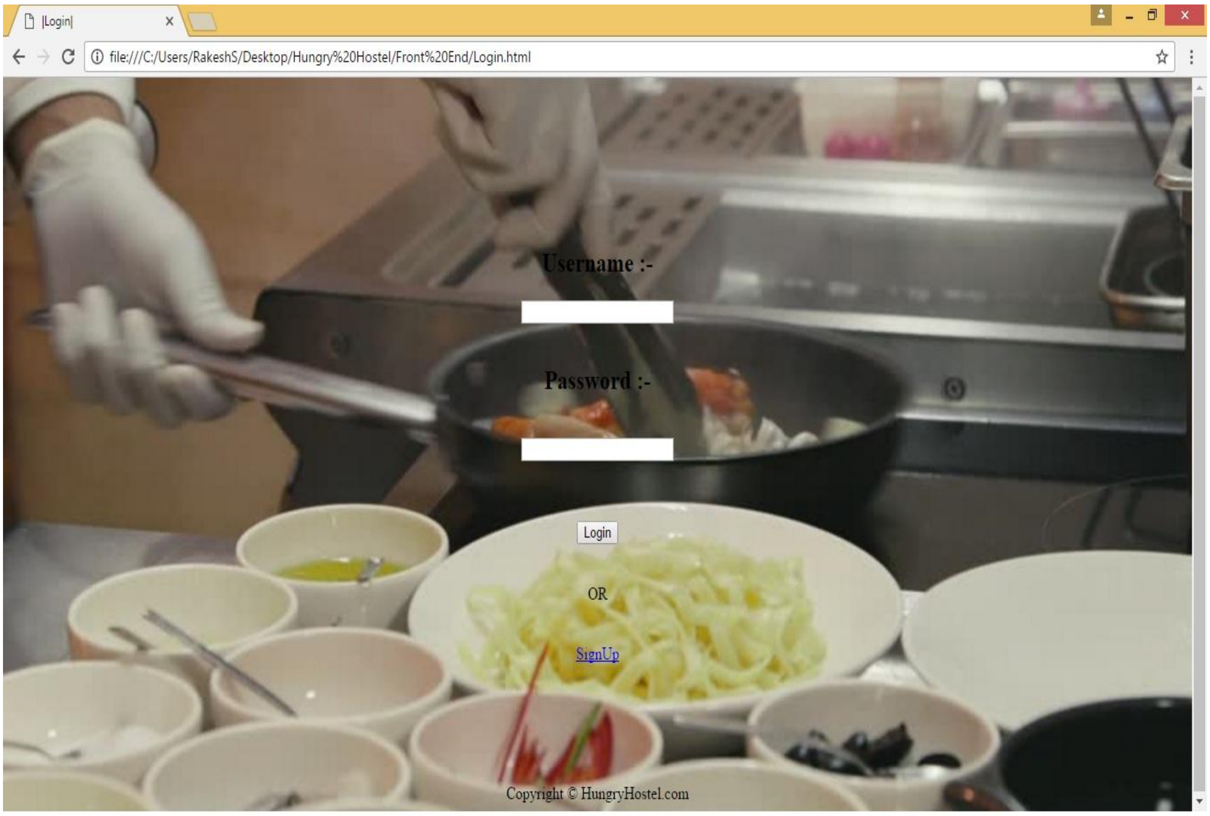
4.3 GUI DESIGN

HOMEPAGE:


|Hungry Hostel|Home
file:///C:/Users/RakeshS/Desktop/Hungry%20Hostel/Front%20End/HomePage_Final.html
Login
Cook
Order
Help

This is the homepage of the website Hungry Hostel 24X7.From homepage links are given for accessing different facilities on this web page like login,order, cook, and help.

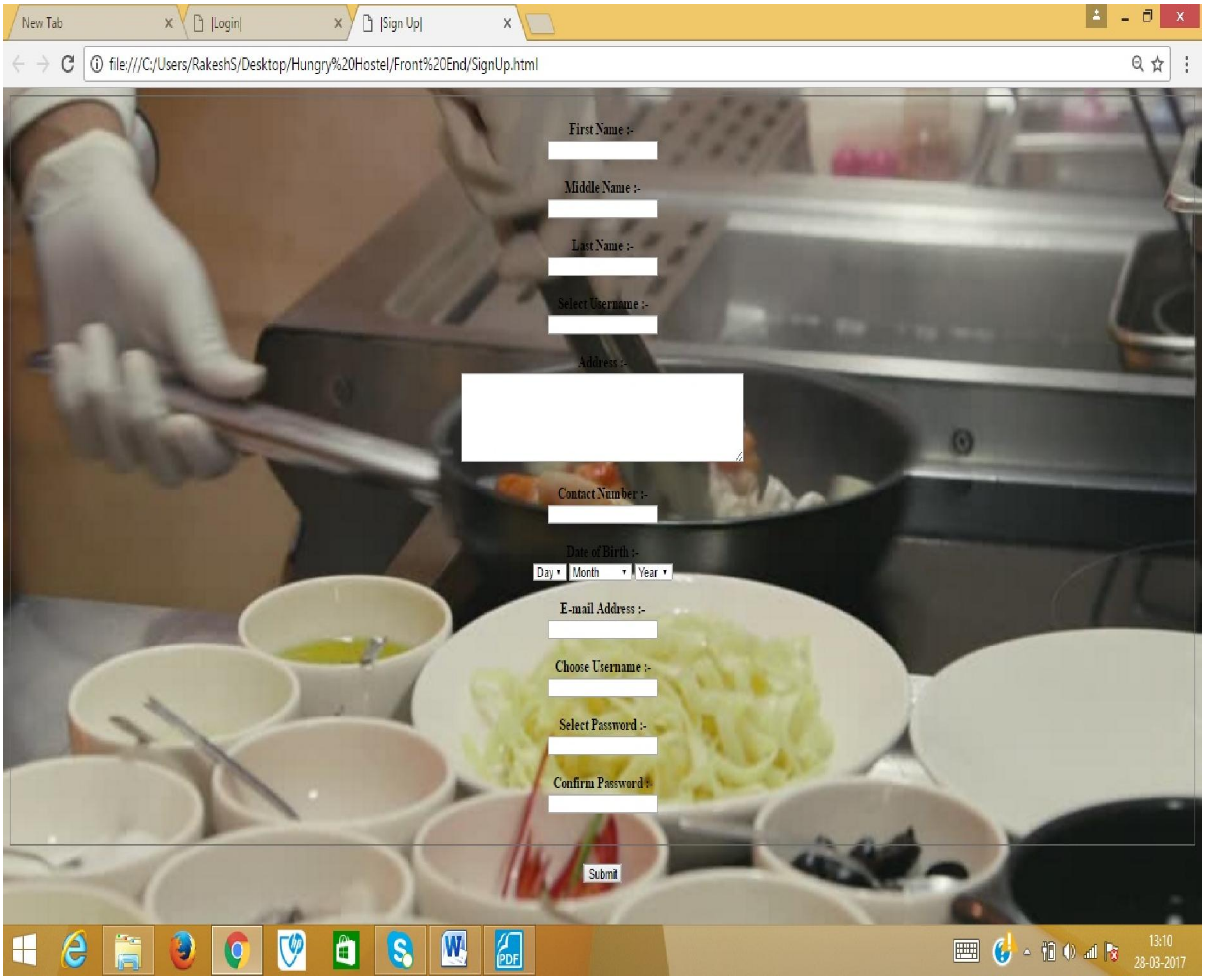
LOGIN:



|Login|
file:///C:/Users/RakeshS/Desktop/Hungry%20Hostel/Front%20End/Login.html
Username :-
Password :-
Login
OR
SignUp
Copyright © HungryHostel.com

The login page allows the user to access his/her webpage by providing his/her user-name and password.

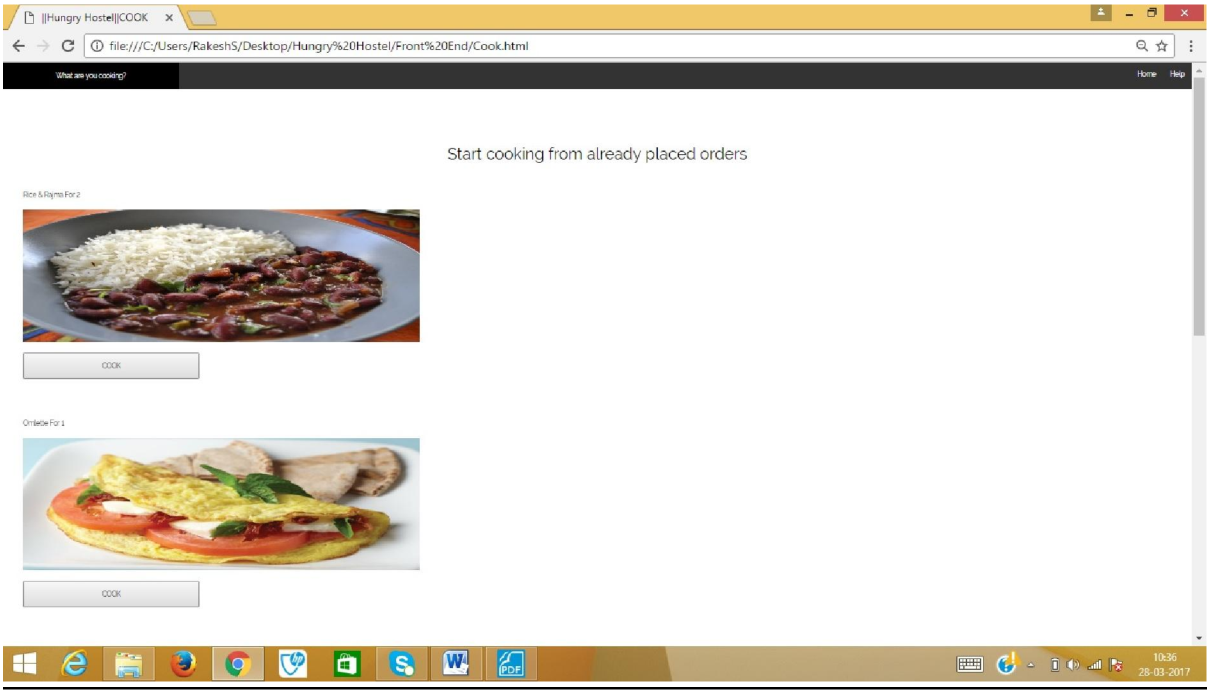
SIGN UP:



If the user doesn’t have an account, then he/she can register himself/herself. The user can register for the cook or for the customer by providing authentic information.

COOK:

ALREADY PLACED ORDERS



||Hungry Hostel||COOK
file:///C:/Users/RakeshS/Desktop/Hungry%20Hostel/Front%20End/Cook.html
What are you cooking?
Home
Help

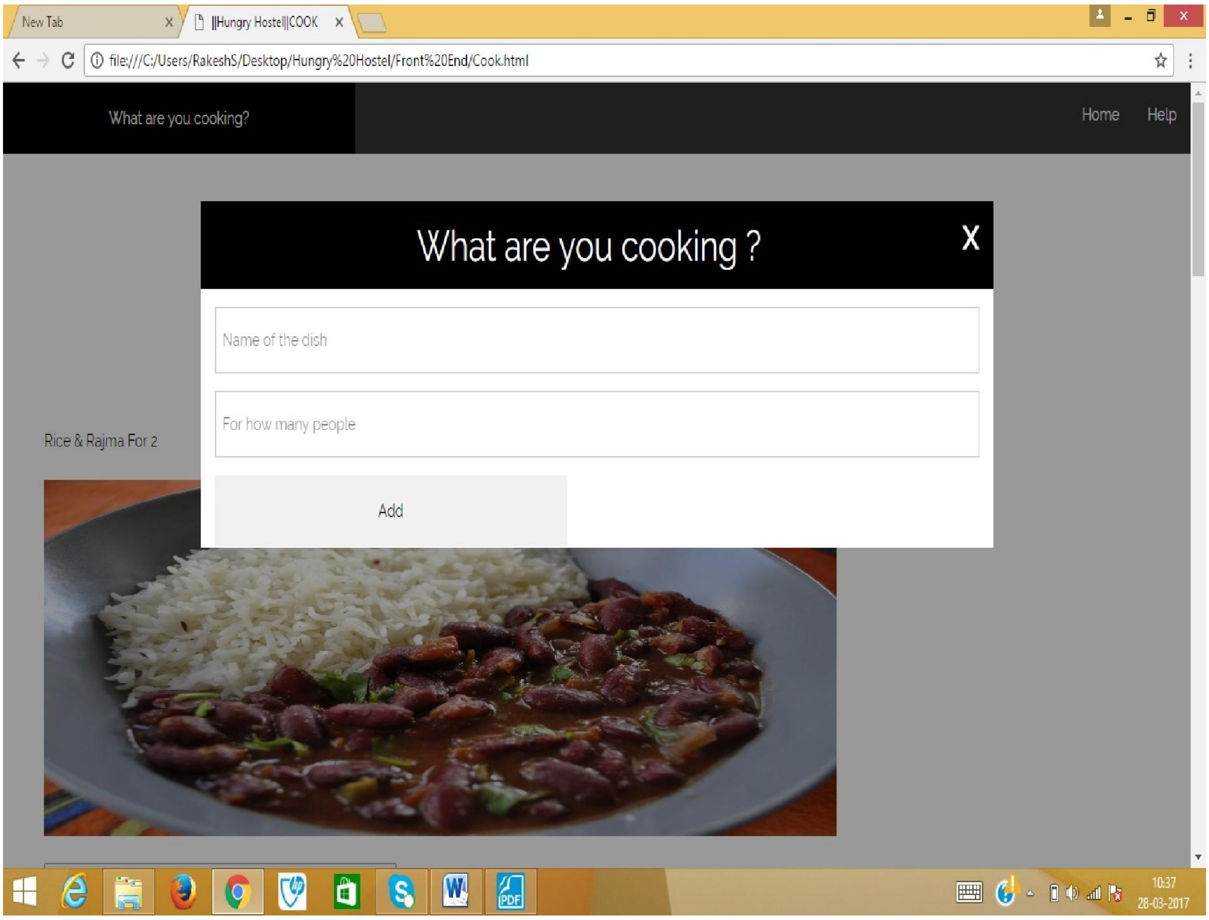
Start cooking from already placed orders

Rice & Rajma For 2

Omlette For 1

In this section, the cook can see the orders requested by the customer. The cook will accept the order that he/she wants to prepare.

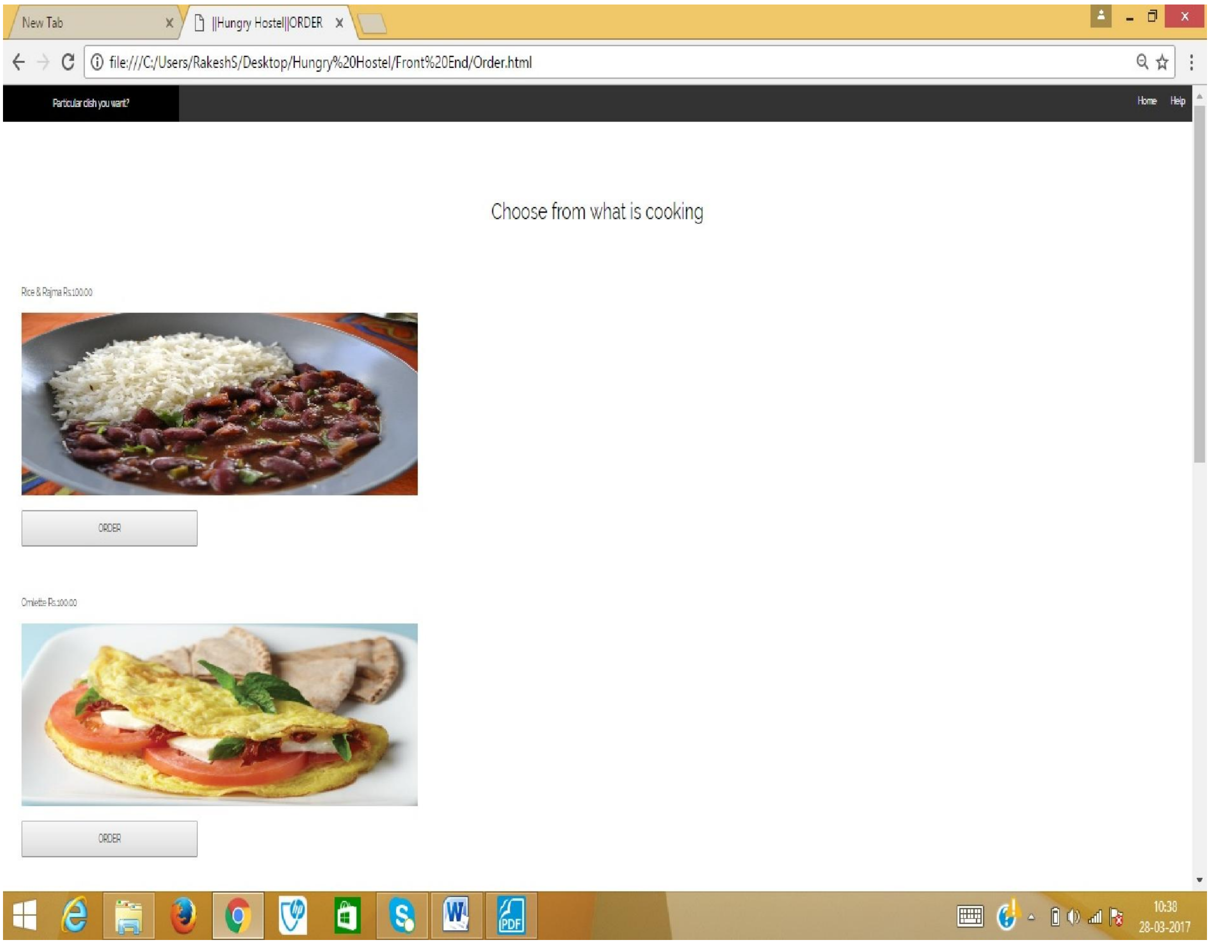
ALREADY COOKED ITEMS BY THE CHEF



In this section, the cook will submit the food items that has already been prepared by him/her. Interested customer can order those dishes.

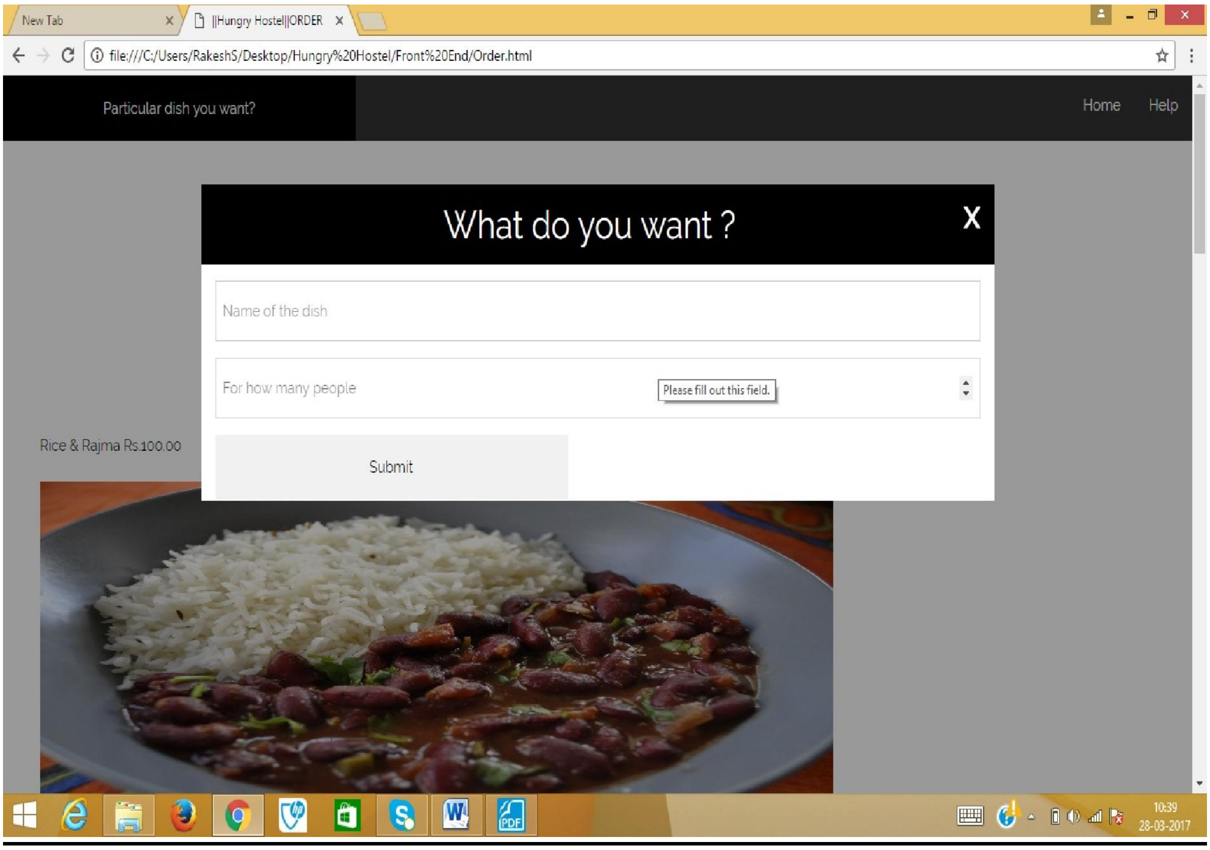
ORDER:

DISHES WHICH ARE ALREADY COOKED


New Tab
||Hungry Hostel||ORDER
file:///C:/Users/RakeshS/Desktop/Hungry%20Hostel/Front%20End/Order.html
Particular dish you want?
Home
Help
Choose from what is cooking
Rice & Rajma Rs.100.00
ORDER
Omlette Rs.100.00
ORDER
10:38
28-03-2017

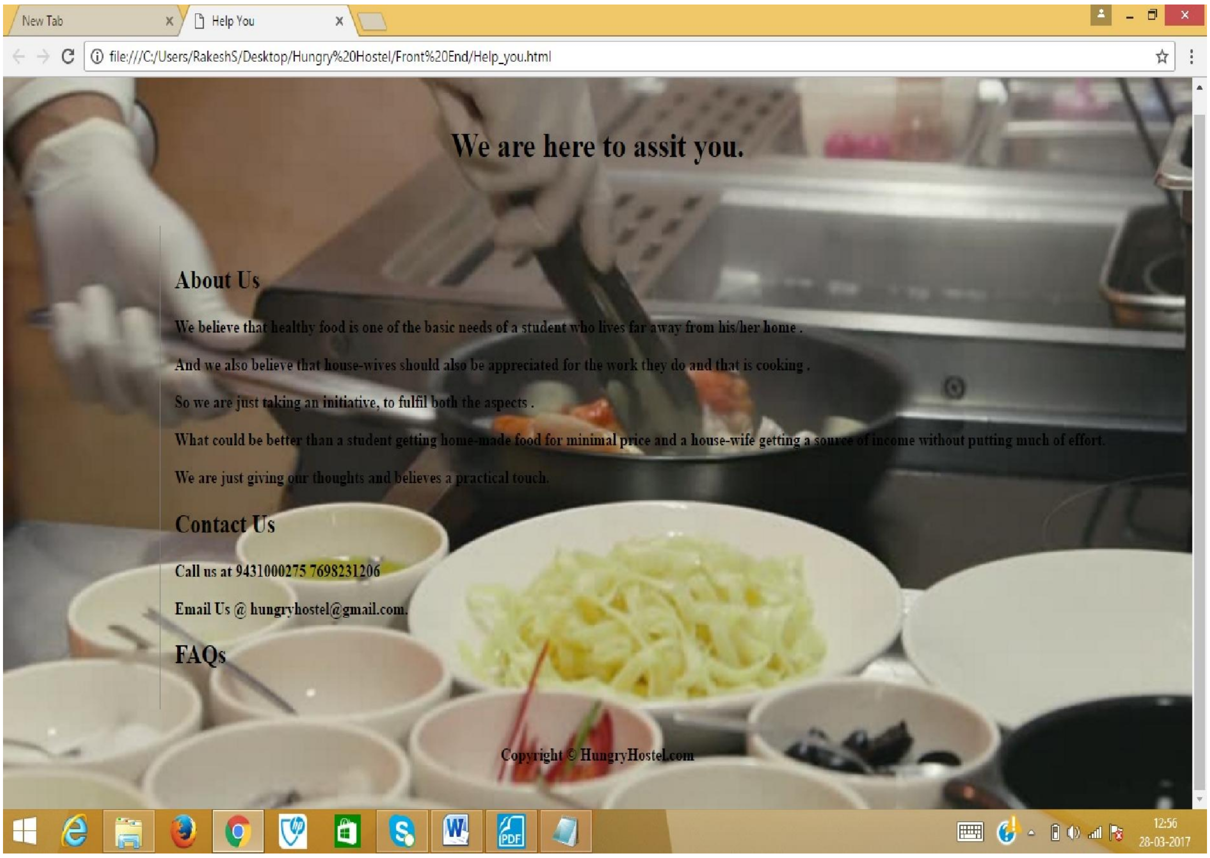
In the section, the customer will see the food-items that has already been prepared by the home-chef. And the customer if interested can select any of the items.

DISHES REQUESTED BY THE CUSTOMER



In this section, the customer can request for the food items he/she wants by providing the name of the dish and the number of head counts.

HELP PAGE:



New Tab
Help You
file:///C:/Users/RakeshS/Desktop/Hungry%20Hostel/Front%20End/Help_you.html
12:56
28-03-2017

We are here to assit you.

About Us

We believe that healthy food is one of the basic needs of a student who lives far away from his/her home .

And we also believe that house-wives should also be appreciated for the work they do and that is cooking .

So we are just taking an initiative, to fulfil both the aspects .

What could be better than a student getting home-made food for minimal price and a house-wife getting a source of income without putting much of effort.

We are just giving our thoughts and believes a practical touch.

Contact Us

Call us at 9431000275 7698231206

Email Us @ hungryhostel@gmail.com.

FAQs

Copyright © HungryHostel.com

This page contains details about the developer, provides link for the users to contact them and also the FAQs.

4.4 TEST CASES DESIGN

Test Cases for registration:

ID	PRECONDITION	PREOCEDURE	EXPECTED OUTPUT	REMARKS
REG1.	Page with blank entries and sign up button is active	Blank entries and click the sign up button	Should display an error message and prompt the user to fill all details.	Enter details into fields and click sign up.
REG2.	Page with blank entries and sign up button is active	Wrong format details is entered and sign up button is clicked.	Should display an error message and prompt the user to fill all details	Enter details in correct format and click sign up.
REG3.	Page with blank entries and sign up button is active	The details is entered in correct format and the sign up button is clicked	Homepage is displayed.	Process completed.

Test Cases for Login:

ID	PRECONDITION	PROCEDURE	EXPECTED OUTPUT	REMARKS
LOG1.	Login page with blank entries and login button is active	Blank entries and the login button is clicked	Should display an error message and prompt the user to fill all details.	Enter ID and password and click login
LOG2.	Login page with blank entries and login button is active	Wrong format/ details is entered and the login button is clicked.	Display error message and prompt the user wrong ID or password	Enter correct ID and password and click login.
LOG3.	Login page with blank entries and login button is active	Correct format / details is entered and the login button is clicked	Directs the user to the homepage.	Process Completed.

Test Cases for Placing_Order:

ID	PRECONDITION	PROCEDURE	EXPECTED OUTPUT	REMARKS
PO1.	Order page with currently available dishes and blank custom order field with order button active.	Customised order with wrong dish name.	Error message prompting to enter correct dish name.	Enter correct dish name and place order
PO2.	Order page with currently available dishes and blank custom order field with order button active.	Customised order with wrong head counts.	Error message prompting to enter head counts within the range(1-10)	Enter correct head counts and place order.
PO3.	Order page with currently available dishes and blank custom order field with order button active.	Select dish from live dish field.	Order will be placed.	Process completed.
PO4.	Order page with currently available dishes and blank custom order field with order button active.	Customised order with correct dish name and head counts.	Order will be placed.	Process completed.

Test Cases for Cooking:

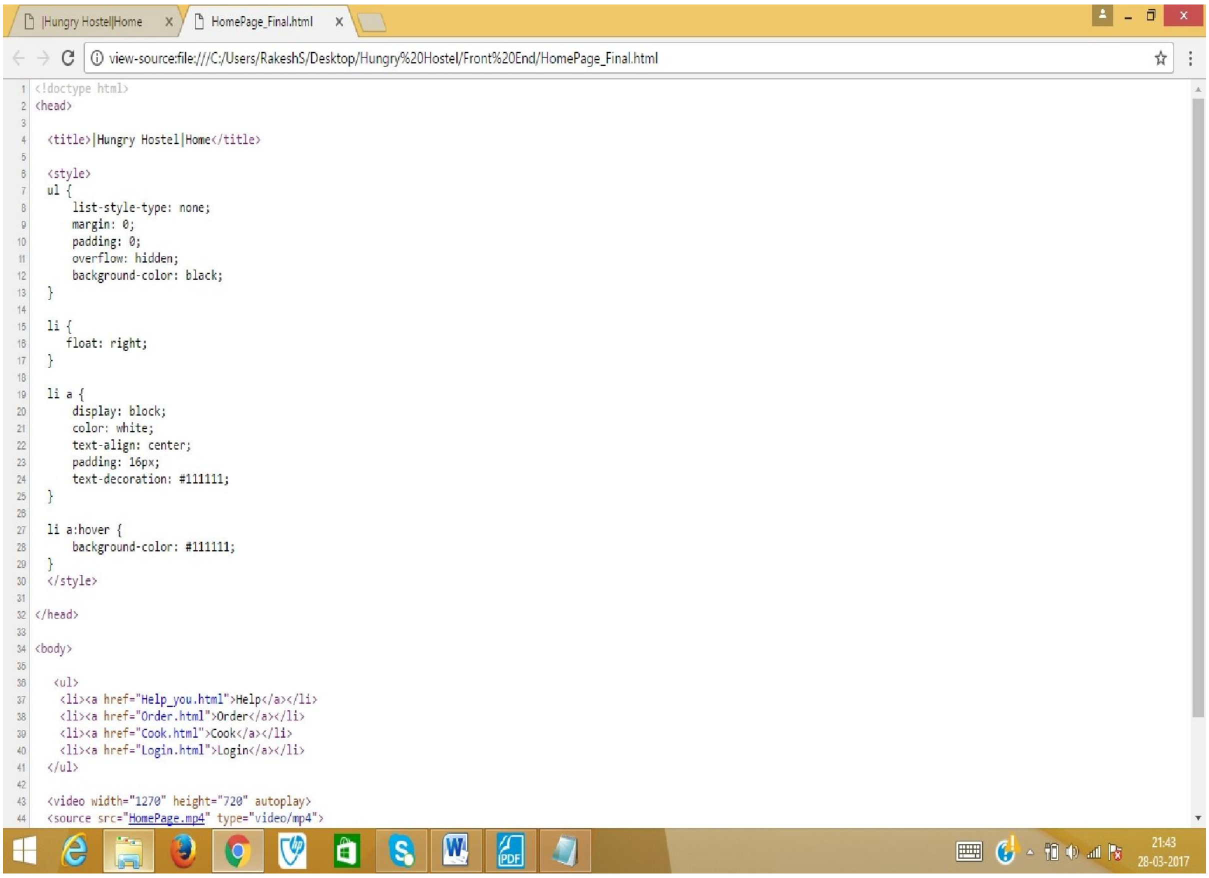
ID	PRECONDITION	PROCEDURE	EXPECTED OUTPUT	REMARKS
CO1.	Cooking page with received/current orders in the market and blank custom/cook field with cook button active.	Customised order with wrong dish name.	Error message prompting to enter correct dish name.	Enter correct dish name and place order
CO2.	Cooking page with received/current orders in the market and blank custom/cook field with cook button active.	Customised order with wrong head counts.	Error message prompting to enter head counts within the range(1-10)	Enter correct head counts and place order.
CO3.	Cooking page with received/current orders in the market and blank custom/cook field with cook button active.	Select dish from live dish field.	Order will be placed.	Process completed.
CO4.	Cooking page with received/current orders in the market and blank custom/cook field with cook button active.	Customised order with correct dish name and head counts.	Order will be placed.	Process completed.

Test Cases for Rating:

ID	PRECONDITION	PROCEDURE	EXPECTED OUTPUT	REMARKS
R1.	Rating page with blank rating field and rate button is active.	Blank entries is entered and rate button is active.	Error message prompting user to enter valid rating(0-5).	Enter a valid rating and click the rate button.
R2.	Rating page with blank rating field and rate button is active.	Invalid rating is entered and the rate button is clicked.	Error message prompting user to enter valid rating(0-5).	Enter a valid rating and click the rate button.
R3.	Rating page with blank rating field and rate button is active.	Valid rating is entered within range(0-5) and the rate button is clicked.	Thank you message and redirected to homepage.	Process completed.

5.CODE SNIPPETS

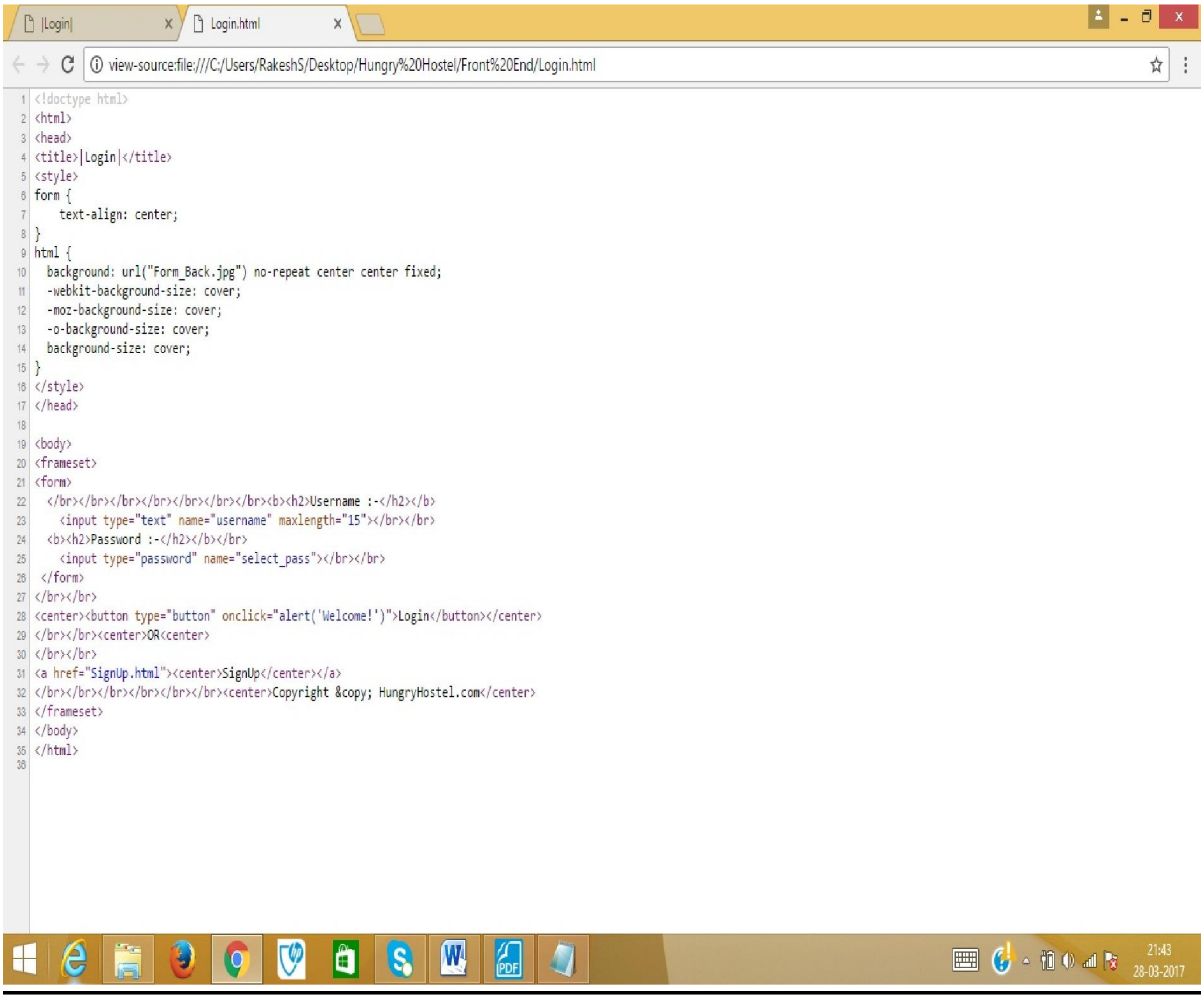
CODE FOR HOMEPAGE:



|Hungry Hostel|Home
HomePage_Final.html
view-source:file:///C:/Users/RakeshS/Desktop/Hungry%20Hostel/Front%20End/HomePage_Final.html
21:43
28-03-2017

```
<!doctype html>
<head>

  <title>|Hungry Hostel|Home</title>

  <style>
  ul {
      list-style-type: none;
      margin: 0;
      padding: 0;
      overflow: hidden;
      background-color: black;
  }

  li {
     float: right;
  }

  li a {
      display: block;
      color: white;
      text-align: center;
      padding: 16px;
      text-decoration: #111111;
  }

  li a:hover {
      background-color: #111111;
  }
  </style>

</head>

<body>

   <ul>
    <li><a href="Help_you.html">Help</a></li>
    <li><a href="Order.html">Order</a></li>
    <li><a href="Cook.html">Cook</a></li>
    <li><a href="Login.html">Login</a></li>
  </ul>

  <video width="1270" height="720" autoplay>
  <source src="HomePage.mp4" type="video/mp4">
```

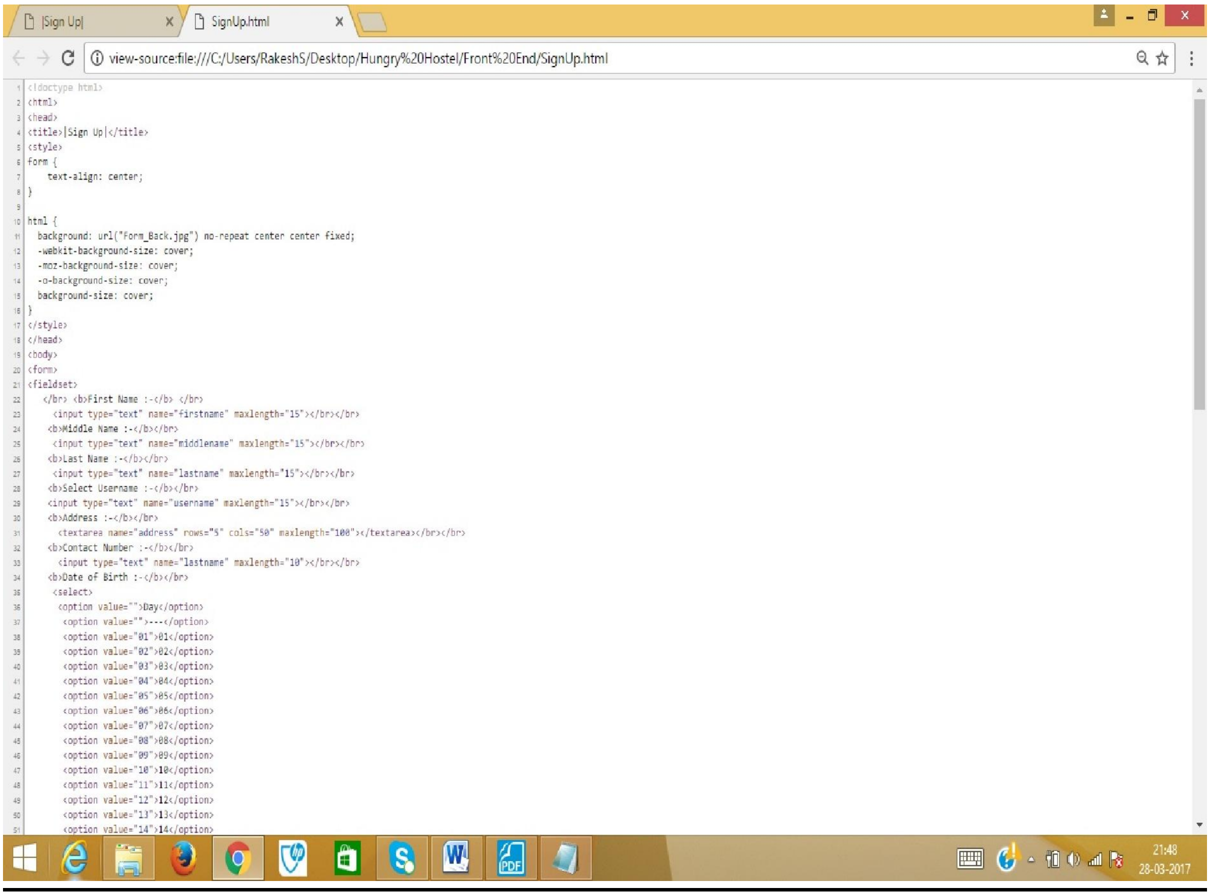
CODE FOR LOGIN PAGE:



|Login|
Login.html
view-source:file:///C:/Users/RakeshS/Desktop/Hungry%20Hostel/Front%20End/Login.html
21:43
28-03-2017

```
<!doctype html>
<html>
<head>
<title>|Login|</title>
<style>
form {
    text-align: center;
}
html {
  background: url("Form_Back.jpg") no-repeat center center fixed;
  -webkit-background-size: cover;
  -moz-background-size: cover;
  -o-background-size: cover;
  background-size: cover;
}
</style>
</head>

<body>
<frameset>
<form>
  </br></br></br></br></br></br></br><b><h2>Username :-</h2></b>
    <input type="text" name="username" maxlength="15"></br></br>
  <b><h2>Password :-</h2></b></br>
    <input type="password" name="select_pass"></br></br>
 </form>
</br></br>
<center><button type="button" onclick="alert('Welcome!')">Login</button></center>
</br></br><center>OR<center>
</br></br>
<a href="SignUp.html"><center>SignUp</center></a>
</br></br></br></br></br></br><center>Copyright &copy; HungryHostel.com</center>
</frameset>
</body>
</html>
```

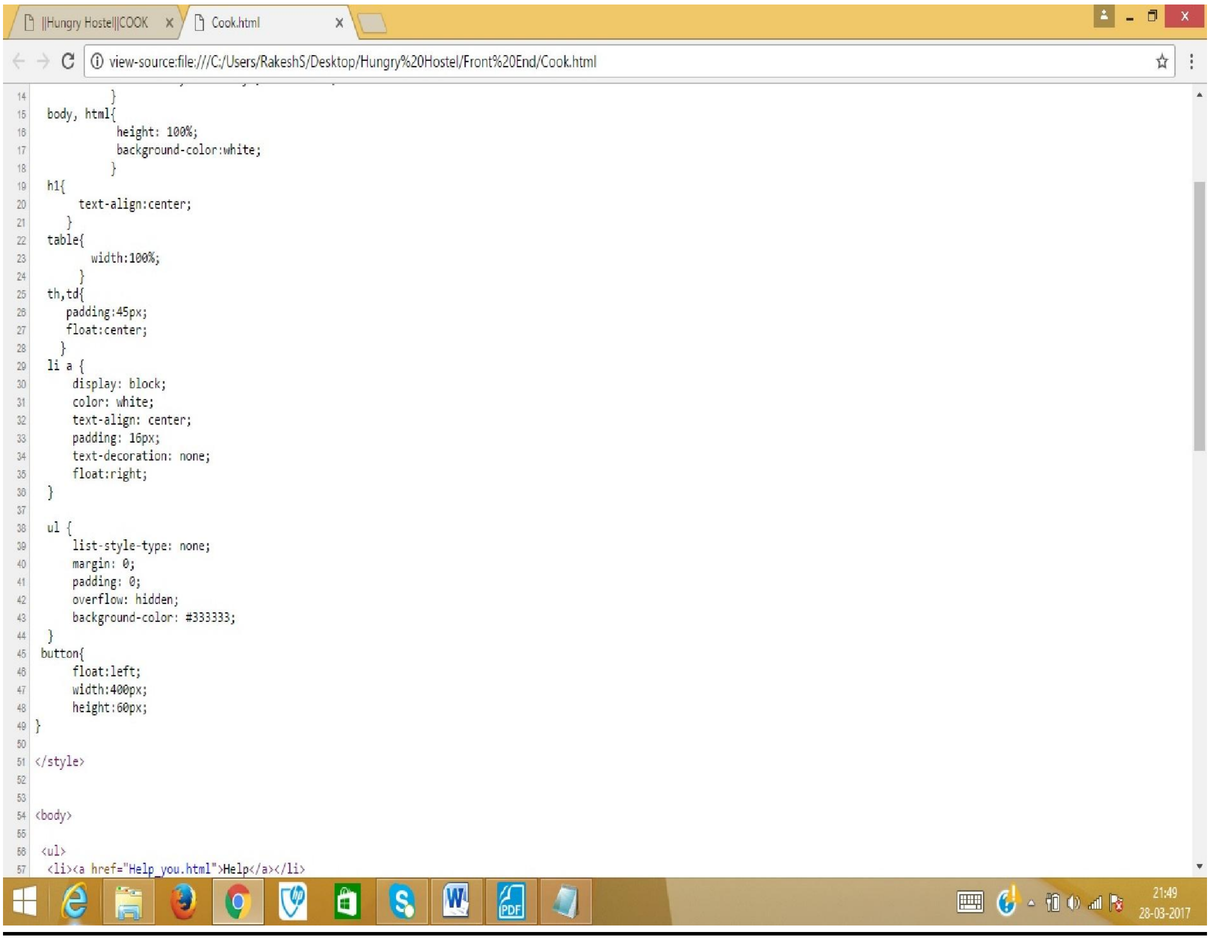
CODE FOR SIGN UP PAGE:



|Sign Up|
SignUp.html
view-source:file:///C:/Users/RakeshS/Desktop/Hungry%20Hostel/Front%20End/SignUp.html
21:48
28-03-2017

```
<!doctype html>
<html>
<head>
<title>|Sign Up|</title>
<style>
form {
    text-align: center;
}

html {
  background: url("Form_Back.jpg") no-repeat center center fixed;
  -webkit-background-size: cover;
  -moz-background-size: cover;
  -o-background-size: cover;
  background-size: cover;
}
</style>
</head>
<body>
<form>
<fieldset>
   </br> <b>First Name :-</b> </br>
     <input type="text" name="firstname" maxlength="15"></br></br>
    <b>Middle Name :-</b></br>
     <input type="text" name="middlename" maxlength="15"></br></br>
    <b>Last Name :-</b></br>
     <input type="text" name="lastname" maxlength="15"></br></br>
    <b>Select Username :-</b></br>
    <input type="text" name="username" maxlength="15"></br></br>
    <b>Address :-</b></br>
      <textarea name="address" rows="5" cols="50" maxlength="100"></textarea></br></br>
    <b>Contact Number :-</b></br>
      <input type="text" name="lastname" maxlength="10"></br></br>
    <b>Date of Birth :-</b></br>
     <select>
      <option value="">Day</option>
       <option value="">---</option>
       <option value="01">01</option>
       <option value="02">02</option>
       <option value="03">03</option>
       <option value="04">04</option>
       <option value="05">05</option>
       <option value="06">06</option>
       <option value="07">07</option>
       <option value="08">08</option>
       <option value="09">09</option>
       <option value="10">10</option>
       <option value="11">11</option>
       <option value="12">12</option>
       <option value="13">13</option>
       <option value="14">14</option>
```

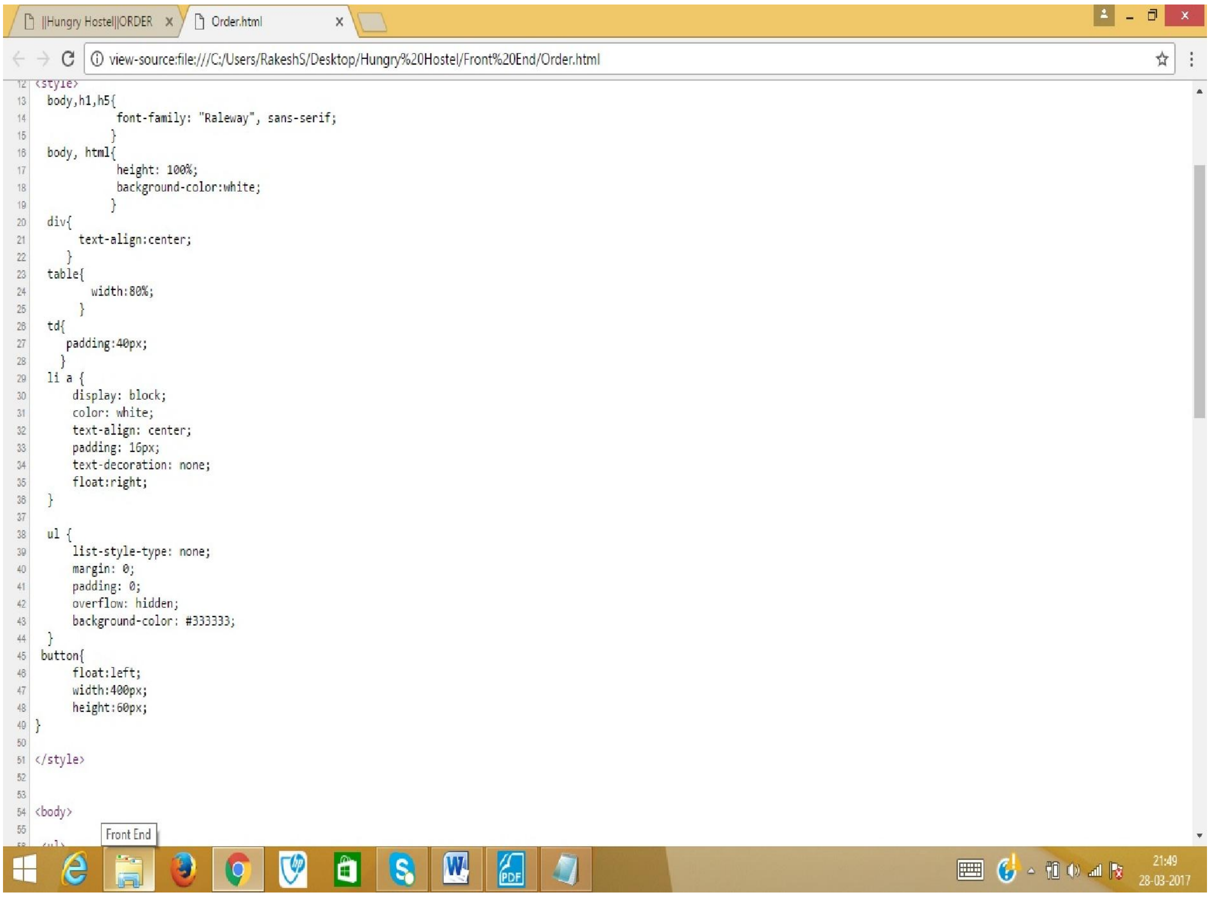
CODE FOR COOK PAGE:



||Hungry Hostel||COOK
Cook.html
view-source:file:///C:/Users/RakeshS/Desktop/Hungry%20Hostel/Front%20End/Cook.html
21:49
28-03-2017

```
          }
  body, html{
           height: 100%;
           background-color:white;
          }
  h1{
      text-align:center;
    }
  table{
        width:100%;
      }
  th,td{
     padding:45px;
     float:center;
    }
  li a {
      display: block;
      color: white;
      text-align: center;
      padding: 16px;
      text-decoration: none;
      float:right;
  }

  ul {
      list-style-type: none;
      margin: 0;
      padding: 0;
      overflow: hidden;
      background-color: #333333;
  }
 button{
      float:left;
      width:400px;
      height:60px;
}

</style>


<body>

 <ul>
  <li><a href="Help_you.html">Help</a></li>
```

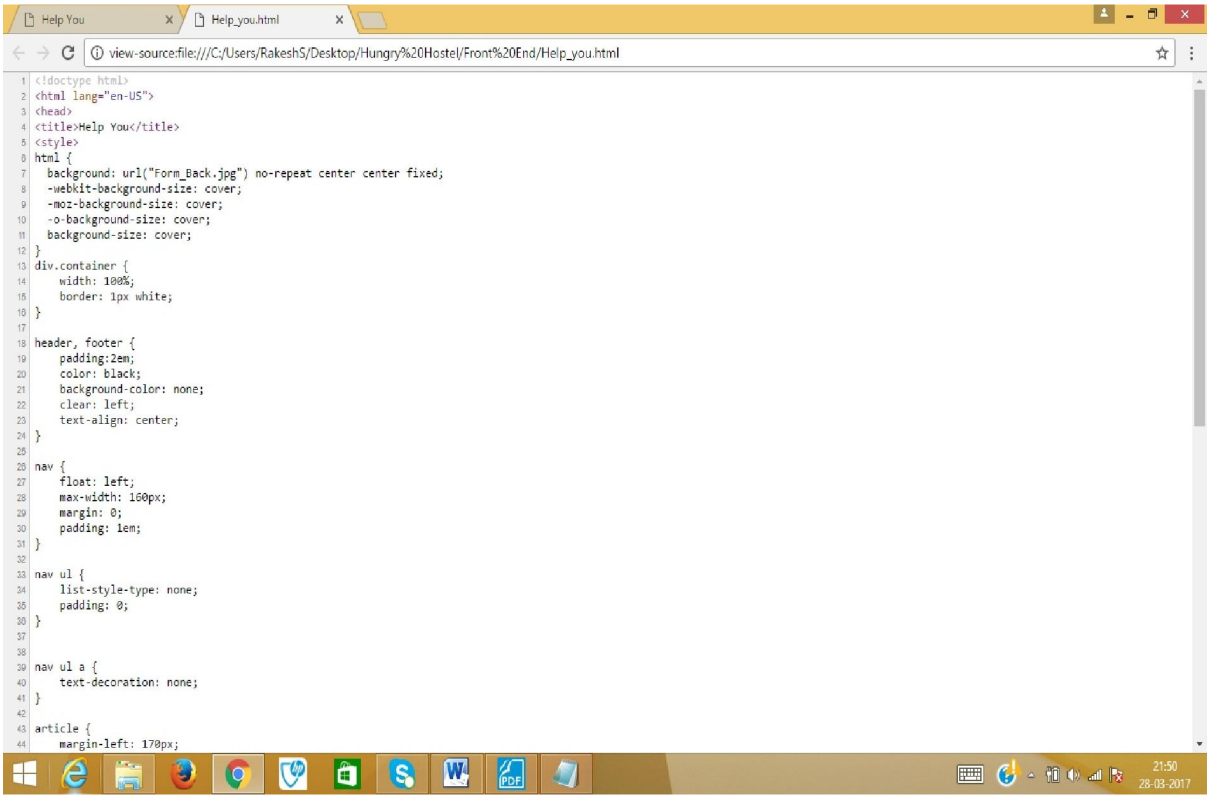
CODE FOR ORDER PAGE:



||Hungry Hostel||ORDER
Order.html
view-source:file:///C:/Users/RakeshS/Desktop/Hungry%20Hostel/Front%20End/Order.html
Front End
21:49
28-03-2017

```
<style>
  body,h1,h5{
             font-family: "Raleway", sans-serif;
            }
  body, html{
             height: 100%;
             background-color:white;
            }
  div{
       text-align:center;
     }
  table{
         width:80%;
       }
  td{
     padding:40px;
    }
  li a {
      display: block;
      color: white;
      text-align: center;
      padding: 16px;
      text-decoration: none;
      float:right;
  }

  ul {
      list-style-type: none;
      margin: 0;
      padding: 0;
      overflow: hidden;
      background-color: #333333;
  }
 button{
      float:left;
      width:400px;
      height:60px;
}

</style>


<body>

  <ul>
```

CODE FOR HELP PAGE:


Help You
Help_you.html
view-source:file:///C:/Users/RakeshS/Desktop/Hungry%20Hostel/Front%20End/Help_you.html
21:50
28-03-2017

```
<!doctype html>
<html lang="en-US">
<head>
<title>Help You</title>
<style>
html {
  background: url("Form_Back.jpg") no-repeat center center fixed;
  -webkit-background-size: cover;
  -moz-background-size: cover;
  -o-background-size: cover;
  background-size: cover;
}
div.container {
    width: 100%;
    border: 1px white;
}

header, footer {
    padding:2em;
    color: black;
    background-color: none;
    clear: left;
    text-align: center;
}

nav {
    float: left;
    max-width: 160px;
    margin: 0;
    padding: 1em;
}

nav ul {
    list-style-type: none;
    padding: 0;
}


nav ul a {
    text-decoration: none;
}

article {
    margin-left: 170px;
```

6. FUTURE WORK

The project “HUNGRY HOSTEL” aims at providing home-made food to the students at an affordable price and it also aims at providing a source of income for the housewives without putting much of effort.

As an extension of this project, we plan the following:-

- Google map API(s).
- The payment system will also include e-wallet(s), net banking along with cash on delivery.
- Developing an android application for the same project to make it more available.

7. CONCLUSION

Students now a days, are getting more used to roadside foods, junk foods etc. For having a meal they prefer going out to restaurants as hostel mess food in way cannot suffice. Regular consumption of outside food always has an adverse effect on the health. And surveys demonstrate that home-made food is always preferred over outside food or roadside food.

Hence, to minimise this problem faced by the students, we developed this application, which helps the student to get home-made food by paying half the price he/she was paying for outside food.

Moreover, the housewives can get a source of income for the job she loves and that is cooking, without putting much effort.

This application is just an initiative to make life of a student more comfortable when it comes to food.

8. REFERENCE

1) Quora – topic “Uber for X”.
2) Websites or online application we referred to :

   a) Foodpanda.com
   b) Zomato.com
   c) Uber.com
   d) Wikipedia.com
3) Web technology we referred:

   a) W3schools.com
   b) Bento.io
   c) Freecodecamp.com