**PROJECT REPORT**

**ON**

**BOOK A TABLE**

**BY**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**UNDER THE GUIDANCE OF**

**MR. PRAKASH KENE SIR**

**DEPARTMENT OF MCA ENGINEERING**

**P.E.S’S MODERN COLLEGE OF ENGINEERING**

**PUNE – 411 005**

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**2020-21**

**Progressive Education Society’s**

**Modern College of Engineering, Pune- 411005**

**CERTIFICATE**

This is to certify that \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Of Master in computer Application have successfully completed the Mini Project work titled ‘**Book A Table’** during the academic year 2020-21. This report is submitted as partial fulfilment of the requirement of degree in MCA Engineering of Savitribai Phule Pune University.

**Dr. Mrs. K. R. Joshi Dr. V.B. Sangvikar Mr. Shripad Bhide**

**Principal Head of Department Project Guide**

**Acknowledgment**

We would like to express our special thanks of gratitude to our teacher **Mr. Prakash Kene** sir as well our HOD Dr**. Mrs. V.B. Sangvikar** andprincipal **Dr. Mrs. K. R. Joshi** gave us the opportunity to do this project on the topic ‘**Book A Table’.** Which also helped us in doing a lot of research and we came to know about too many things. We would like to thank all of you for your expert advice & encouragement throughout this project.

We also thank our family who encouraged me and prayed for me throughout the time of my research.

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Roll No:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Page No.** |
| 1. | Objective | 5 |
| 2. | Proposed System | 6 |
| 3. | H/W & S/W Requirements | 7 |
| 4. | Advantages Of System | 8 |
| 5. | Entity-Relationship Diagram | 9 |
| 6. | Data Flow Diagram | 10 |
| 7. | Data Dictionary | 12 |
| 8. | UML Diagrams | 15 |
| 9. | UI Screens |  |
| 10. | Test Cases |  |
| 11. | Conclusion |  |
| 12. | Bibliography |  |

**Objective**

The labour rates are increasing steadily year on year thus making it difficult to find employees. The food industry is highly labour intensive and the biggest expense in the food industry is the cost of employing the right kind of people to do the work. One of the ways to reduce this expense is to use modern technology to replace some of the jobs done by human beings and make machines do the work. Here we propose an “Table Booking System” that has been designed for Fast Food restaurant, Take-Out or College Cafeterias.This simplifies the process of table booking for both the customer and the restaurant, as the entire process of taking orders is automated.

This project is aimed at developing a web application for the **‘Book A Table’** based on Python andDjango Rest Framework.

The table booking system for hotels and restaurants is developed as an attempt where user can easily book tables online and admin of system can add services in the databases when the hotel/ restaurant owner increases number of services.

The system provides the facility of viewing all the services to user and user can also select the desired service among all services, later can confirm booking of table by selecting date and time according to user’s need.

After booking a table user will receive an message/email including the details of user’s selected hotel, table, menu, date and time and billing amount from the System.

**Proposed System**

The purpose of this system is to help hotel/restaurant owner in order to ease the task where user can easily book their table online and admin of system can add services in the database when owner increases number of services.

This system automates the task of booking tables in the form of receiving the confirmation of booked tables in the form of messages/email.

Our system gives more security to data, ensures data accuracy, reduces workload and saves time, only registered users get chance to book table and receives confirmation message/mail, it makes information flow efficient reduce the space. Admin of system can add services in the database when restaurant owner increases number of services. Proposed system is cost effective.

To smoothen out the booking and to enhance the experience, Hotel/Restaurant must have a Table Booking System with the following features:

* User friendly UI
* Good database system to store data
* Master Maintenance : - This system maintains booking details along with visitor details, payment details and feedback details.
* Booking Process : - This system maintains  booking details of the visitor along with their personal information for security reasons.
* Feedback Process : - This system provides the information about the feedback of the visitor.

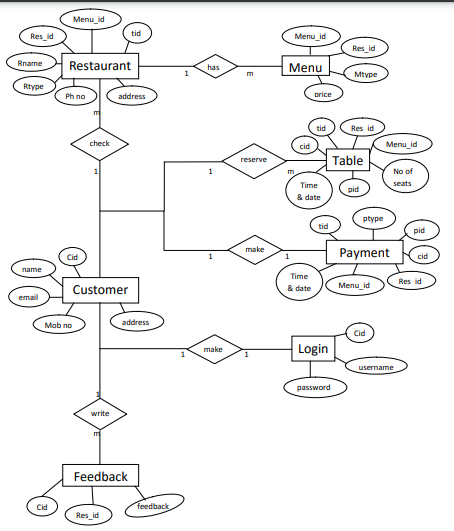
**H/w & S/w Requirement**

1. **Hardware Requirements :**
   * + - Processor (CPU with 2 GHz) frequency.
       - Minimum of 4 GB RAM.
       - Minimum of 500 GB of available space of the Hard Disk.
       - Internet Connection- Broadband (High Speed) .
2. **Software Requirements:**
   * + - Front end: HTML, CSS, BOOTSTRAP.
       - Framework : Django 2.
       - Back end: Python .
       - Database: Postgresql.
       - Operating System : WINDOWS 7/8/10.
       - Browser: Google Chrome, Mozilla Firefox.
       - Softwares : VS Code, Python, Django.

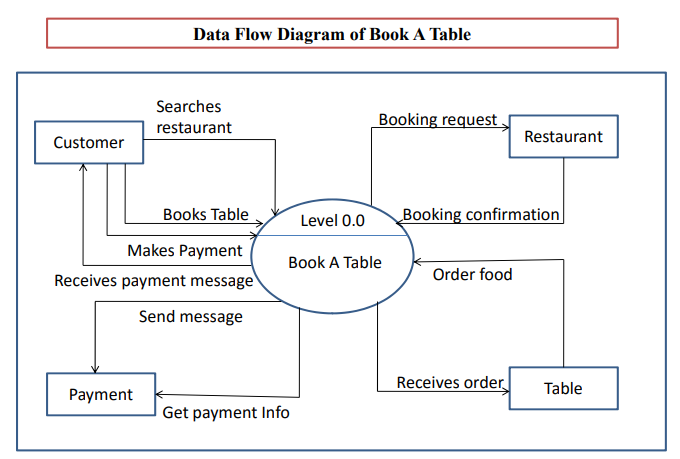
**Advantages Of The System**

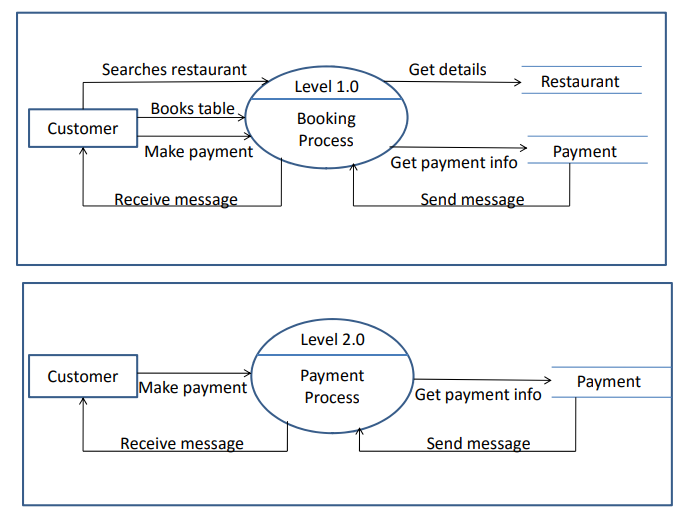
* Saves time of the customer (for example: let's take an example of Good luck cafe we have to wait every time for getting a table. But by this proposed system users will be able to save time as they will get the information of available tables).
* This System will also help to stop the wastage of food because the owner will get the necessary information about demands of customer's and according to the demand they can arrange.
* Customer can book as many tables they wants to book for (Birthday's etc).
* Advance payment will be given to owners after booking the table.

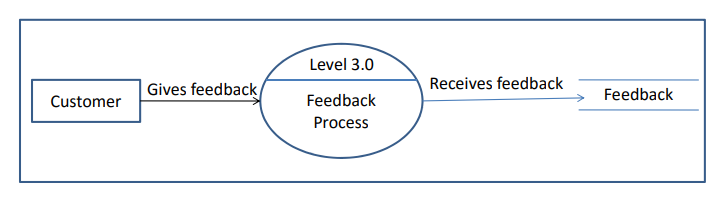
**Entity-Relationship Diagram**

****

**Data Flow Diagram**

****





**Data Dictionary**

**CUSTOMER TABLE:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Size | Description |
| cust\_id(P.K) | Int | - | Customer Id |
| Firstname | Varchar | 80 | First Name |
| Lastname | Varchar | 80 | Last Name |
| mobile\_no | Varchar | 10 | Mobile Number |
| Email | Varchar | 80 | Email Id |
| City | Varchar | 40 | City |
| Password | Varchar | 20 | Password |
| confirm\_password | Varchar | 20 | Confirm Password |

**RESTAURANT TABLE:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Size | Description |
| rest\_id(P.k) | Int | - | Rest Id |
| rest\_name | Varchar | 20 | Rest Name |
| rest\_type | Varchar | 20 | Rest Type |
| ph\_no | Varchar | 10 | Phone Number |
| rest\_address | Varchar | 20 | Rest Address |

**MENU TABLE:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Size | Description |
| menu\_id (P.K) | Int | - | Menu Id |
| Menu\_type | Varchar | 20 | Menu Type |
| menu\_name | Varchar | 30 | Menu Name |
| Price | Int | 10 | Price |
| rest\_id (F.K) | Int | - | Rest Id |

**ORDER TABLE:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Size | Description |
| order\_id(P.K) | Int | - | Order Id |
| rest\_name | Varchar | 20 | Rest Name |
| cust\_id(F.K) | Int | - | Cust Id |
| menu\_id(F.K) | Int | - | Menu Id |
| date\_time | Varchar | 30 | Date & Time |
| no\_of\_persons | Int | - | No Of Persons |
| Qty | Int | - | Qty |

**ORDERMENU TABLE:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Size | Description |
| Id | Int | - | Id |
| Quantity | Int | - | Quantity |
| cust\_id(F.K.) | Int | - | Cust Id |
| menu\_id(F.K.) | Int | - | Menu Id |
| order\_id(F.K.) | Int | - | Order Id |

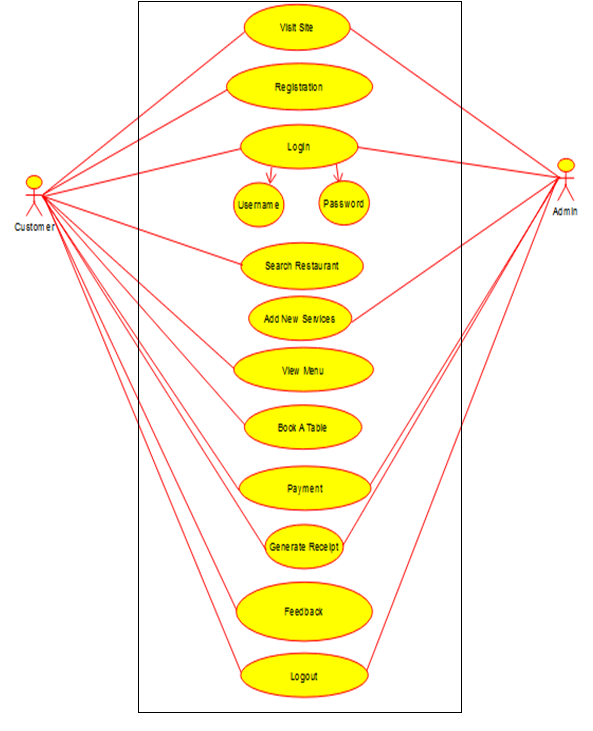
**PAYMENT TABLE:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Size | Description |
| payment\_id(P.K) | Int | - | Payment Id |
| order\_id(F.K) | Int | - | Order Id |
| gross­\_amount | Int | - | Gross Amount |
| tax\_charges | Int | - | Tax Charges |
| net\_amount | Int | - | Net Amount |

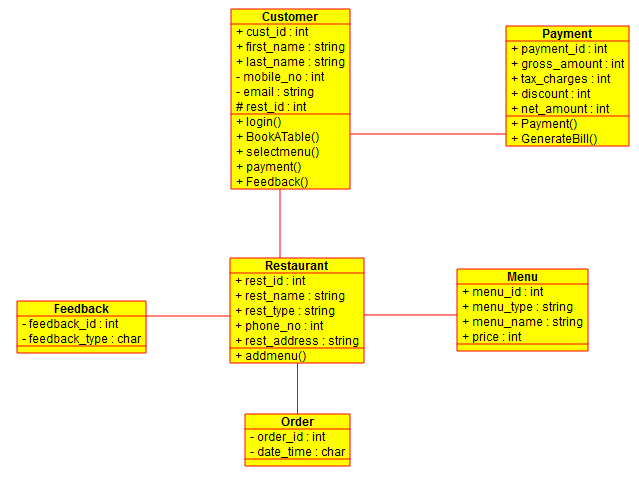
**FEEDBACK TABLE:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Size | Description |
| feedback\_id(P.K) | Int | - | Feedback Id |
| cust\_id(F.K) | Int | - | Cust Id |
| feedback\_type | Varchar | 20 | Feedback Type |
| feedback\_desc | Varchar | 20 | Feedback Desc |
| date\_time | Varchar | 20 | Date & Time |

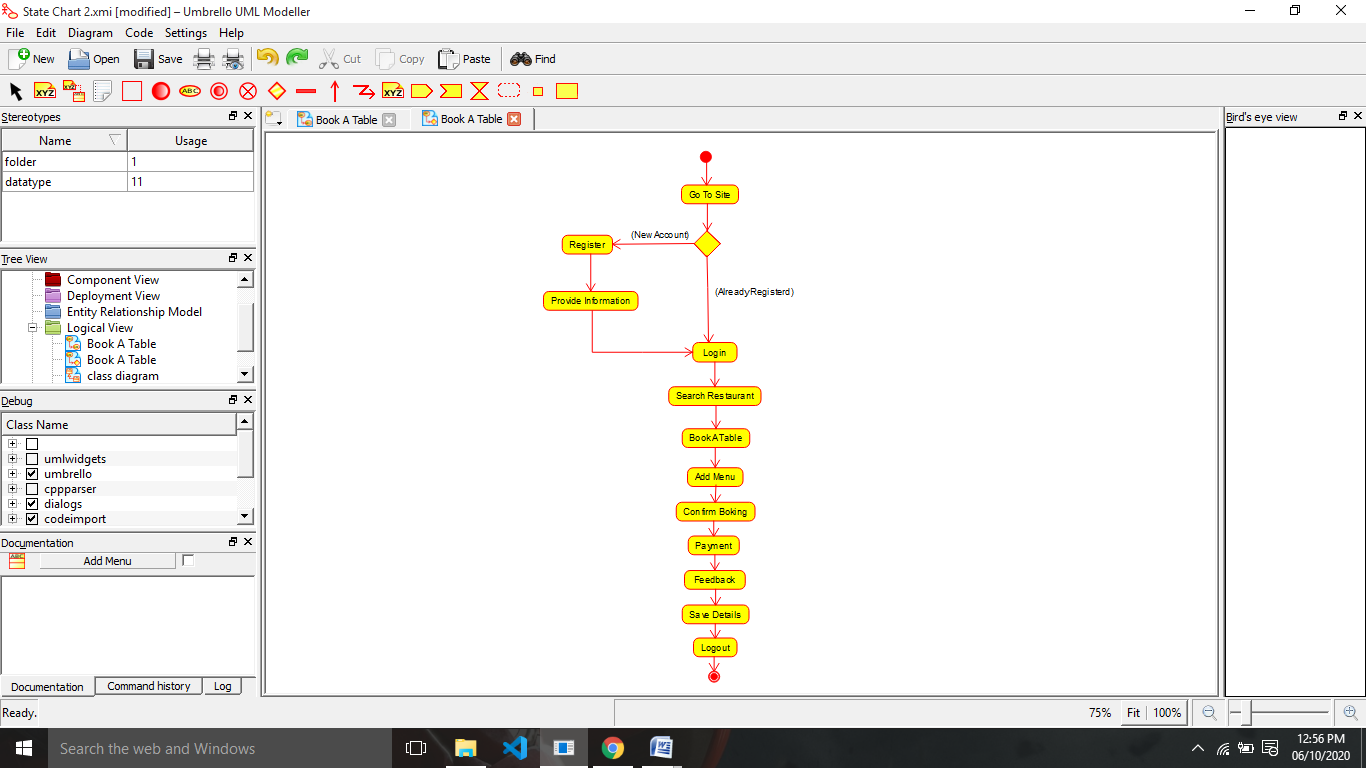
**\*UML Diagrams :-**

**Use Case Diagram**

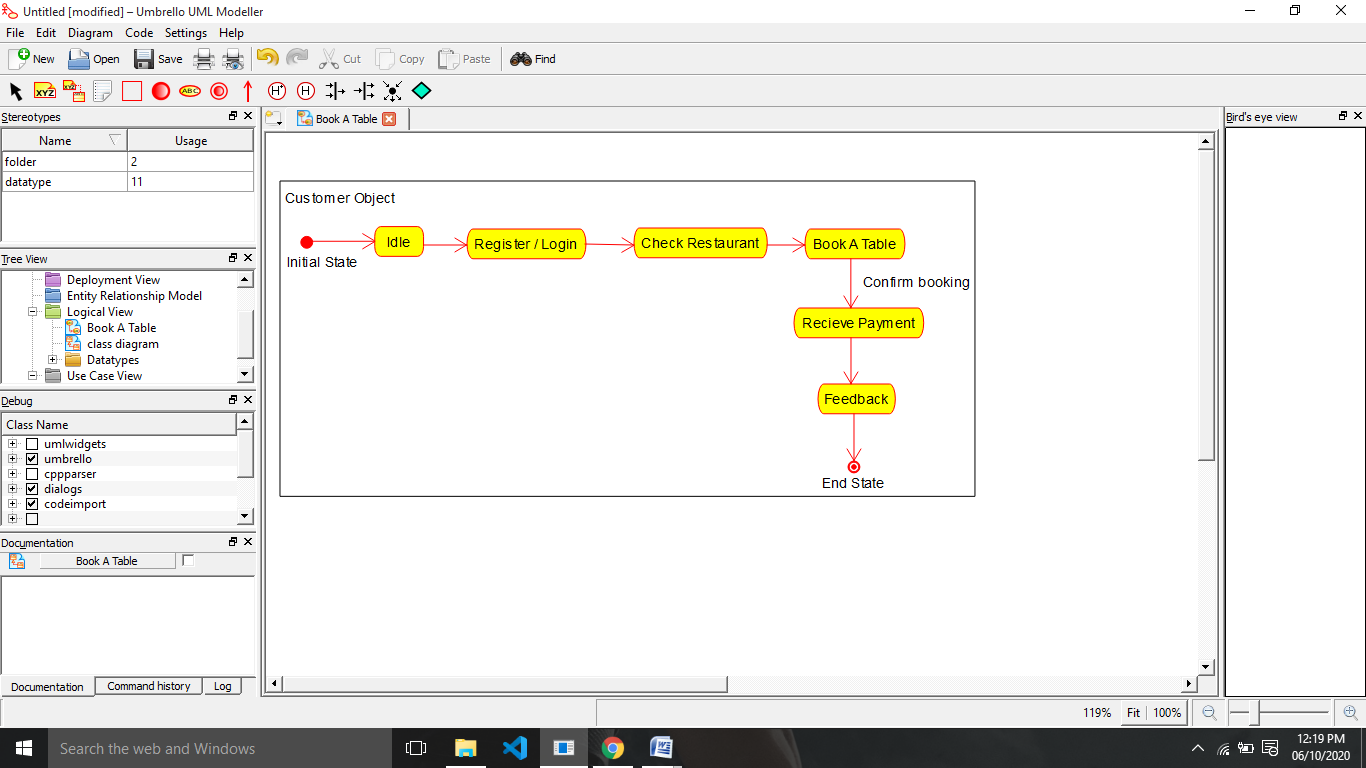
**Class Diagram**

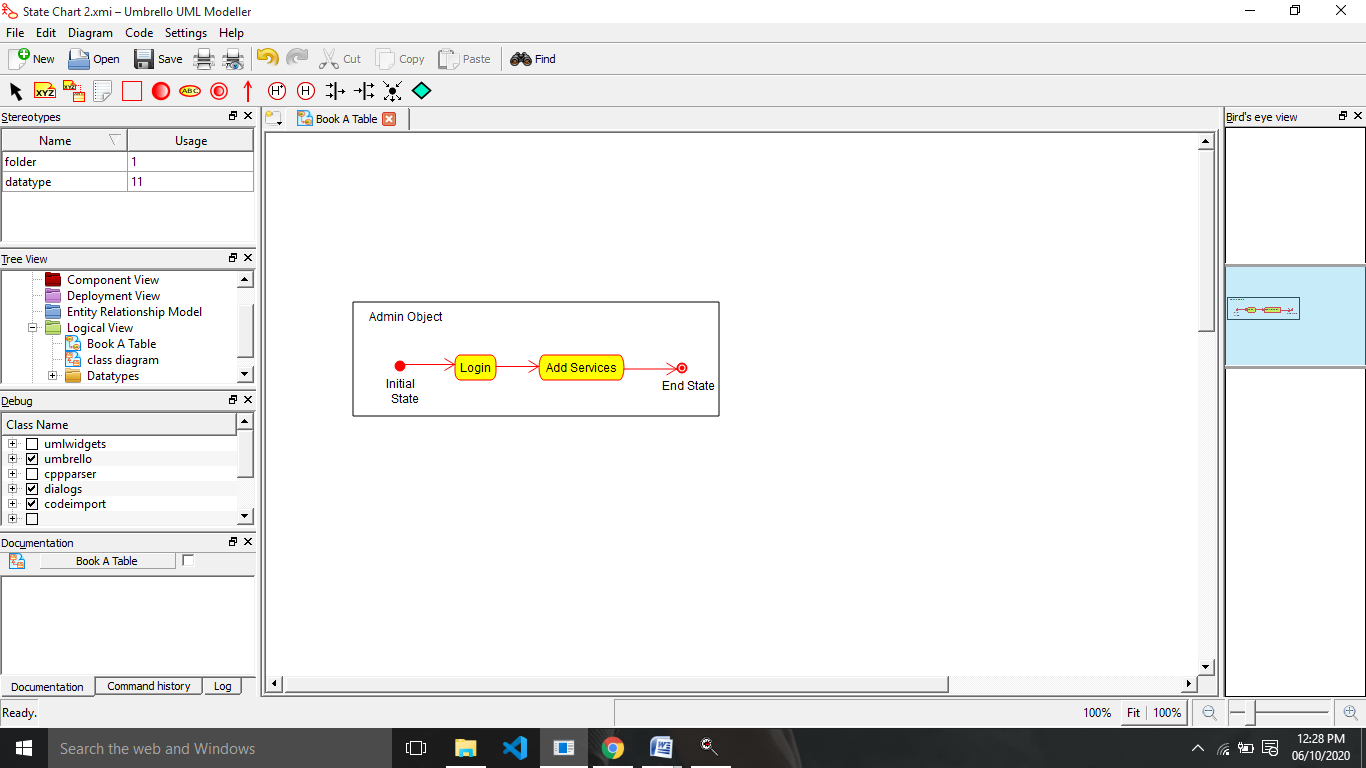
****

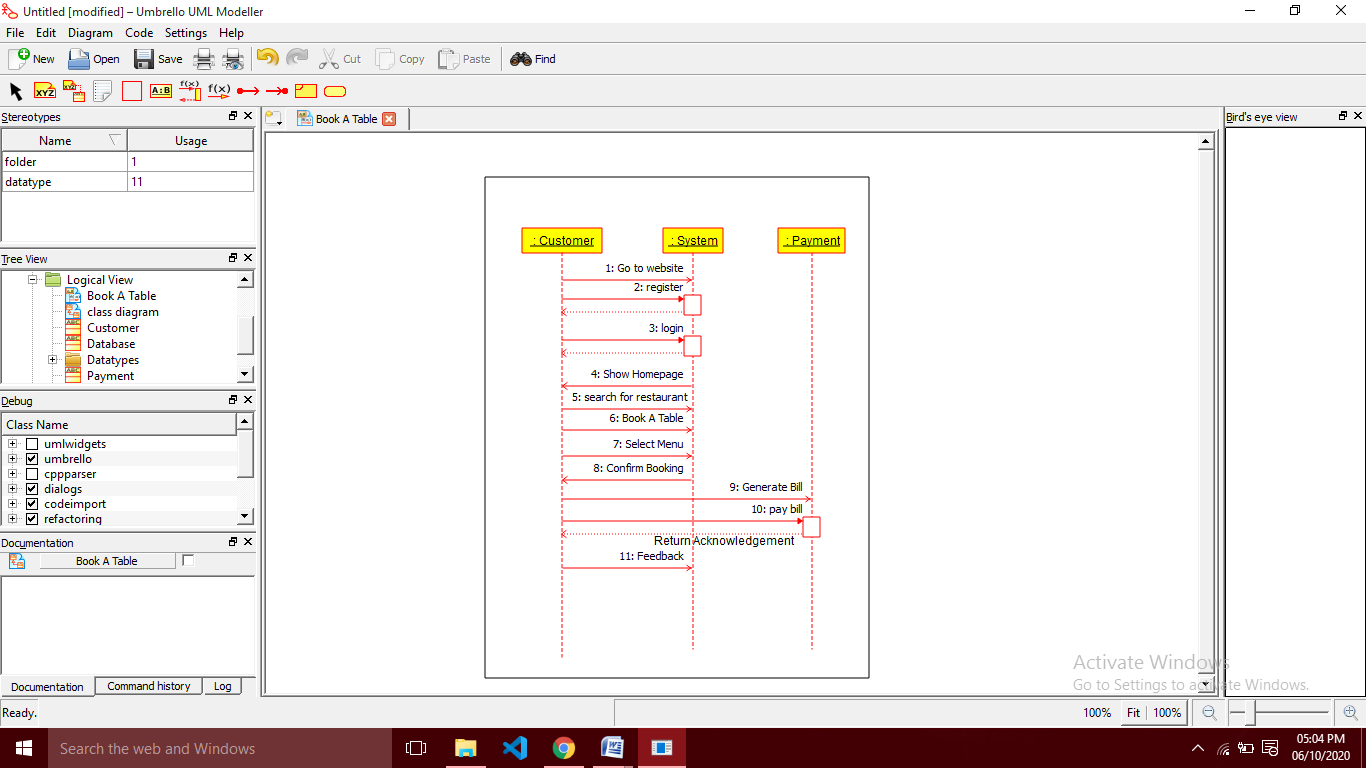
**Activity Diagram**



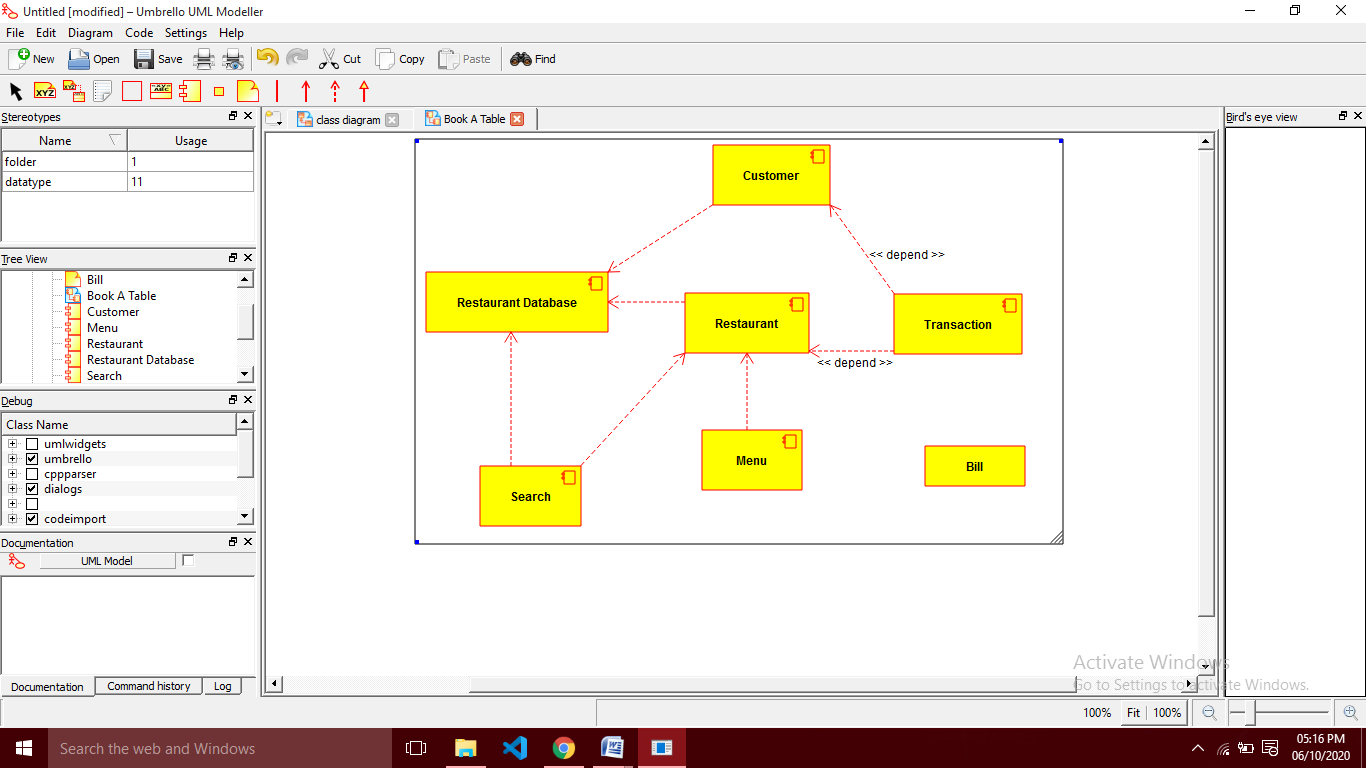
**State Chart Diagram**



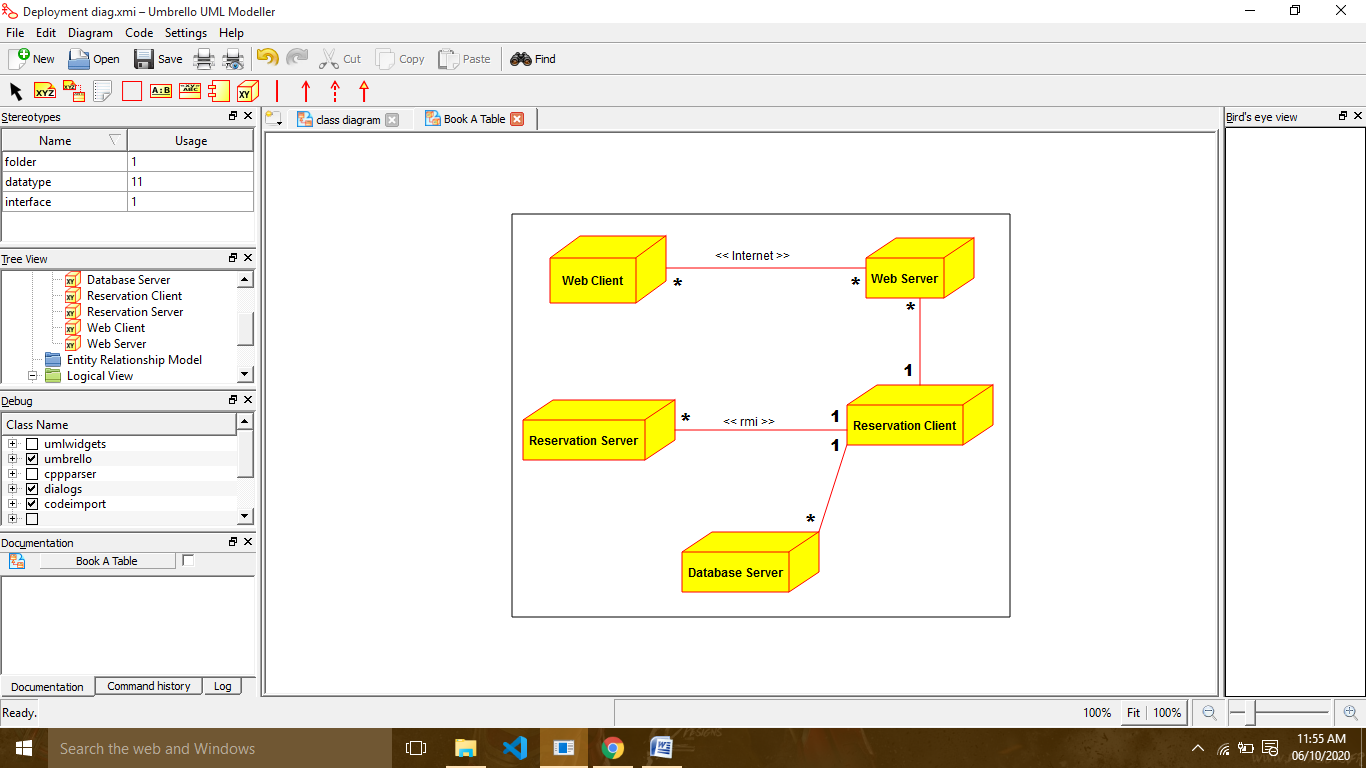


**Sequence Diagram**

**Component Diagram**

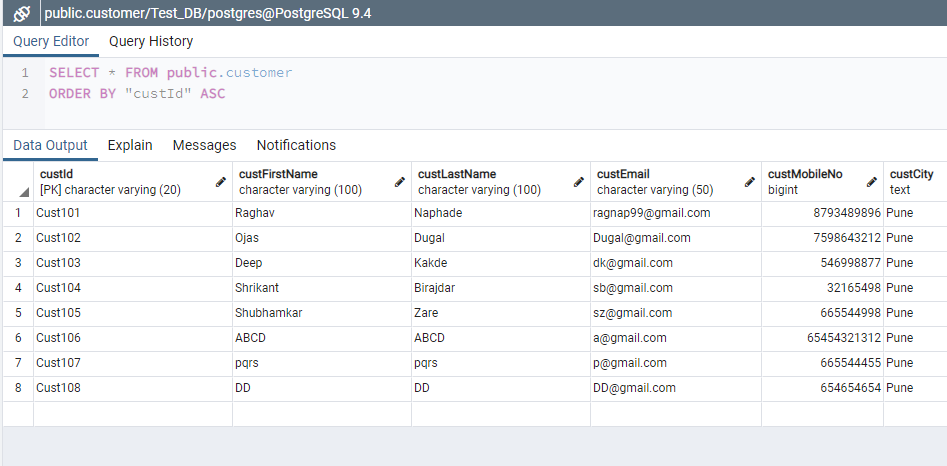
****

**Deployment Diagram**

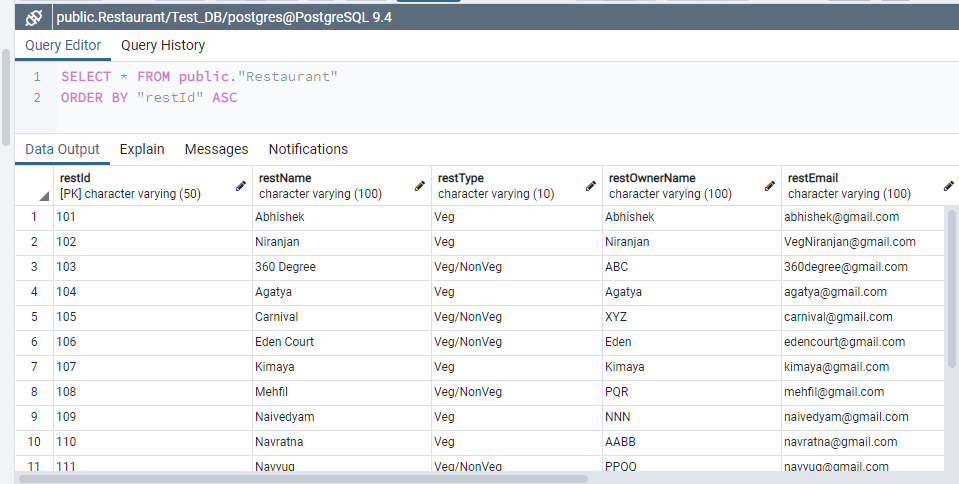


**Database Tables**

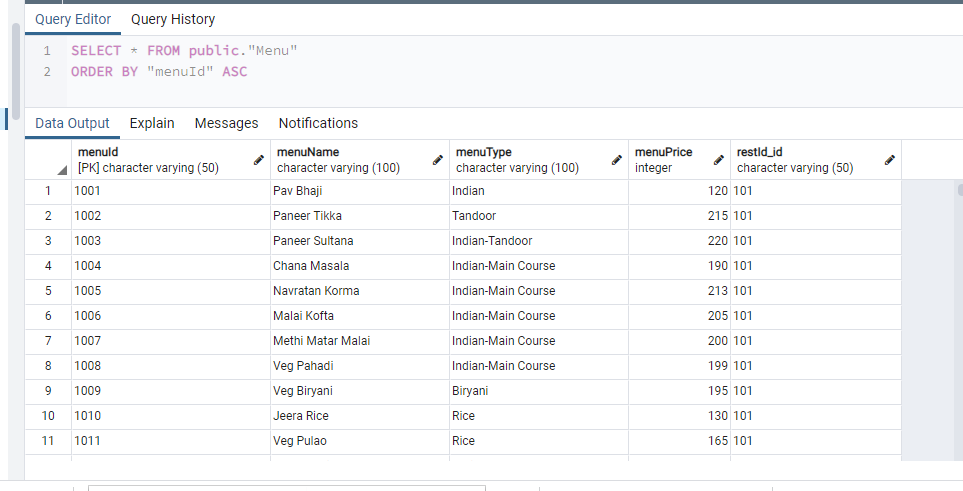
1. **Customer**

****

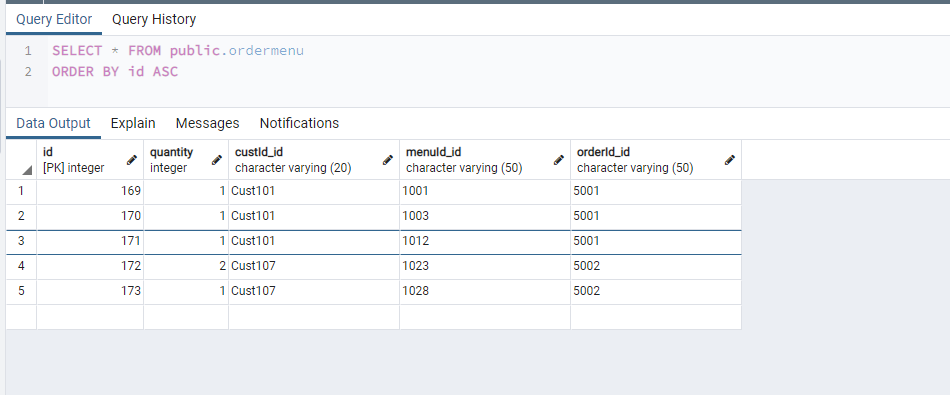
1. **Restaurant**

****

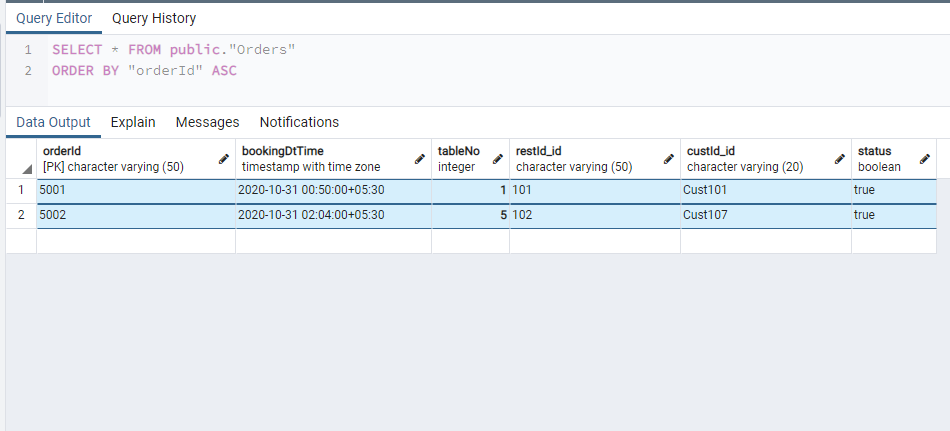
1. **Menu**

****

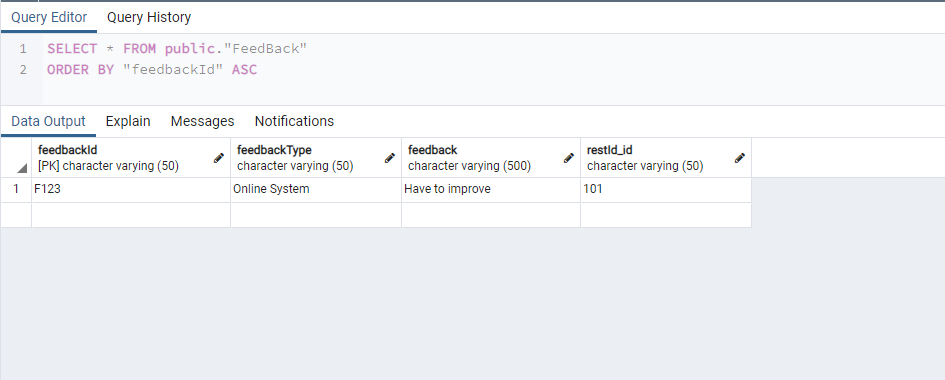
1. **Order Menu**

****

1. **Orders**

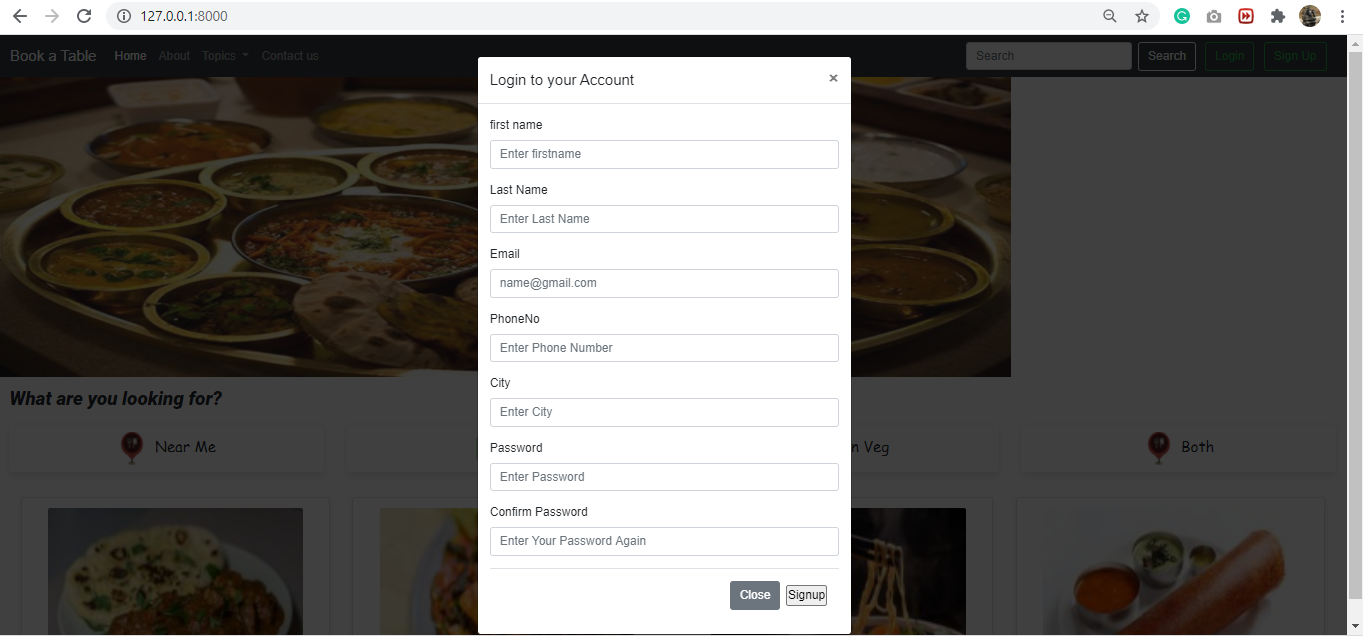
****

1. **FeedBack**

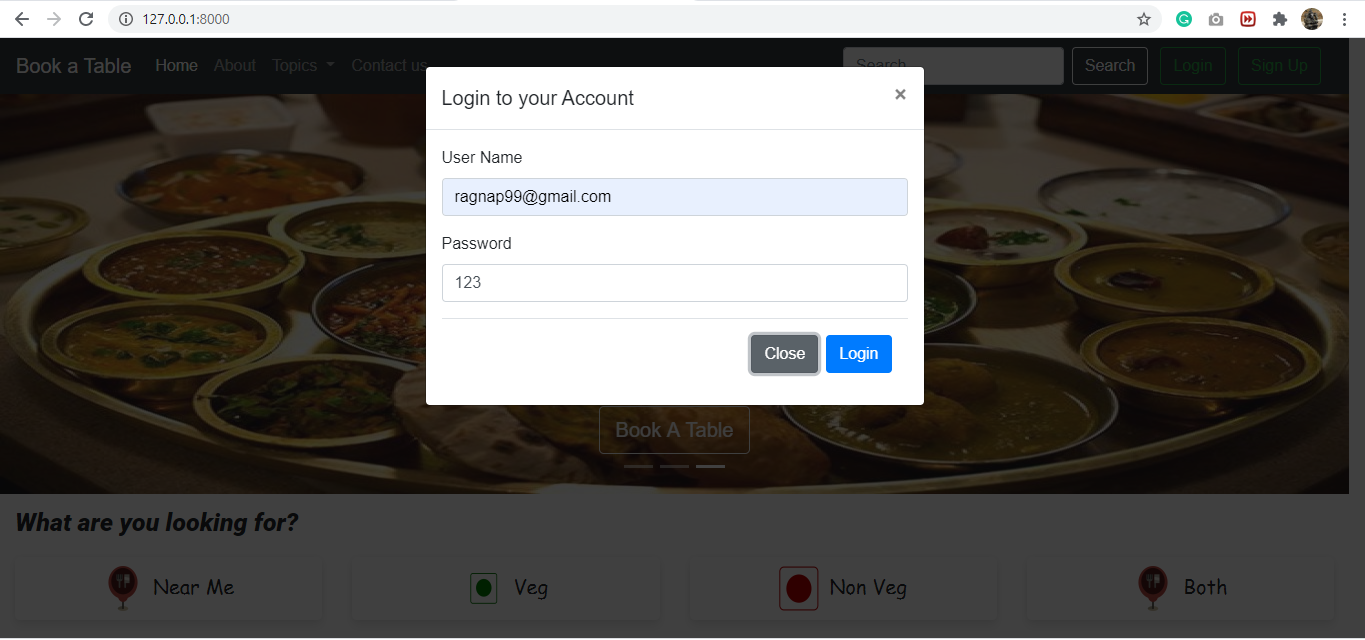
****

**UI Screens**

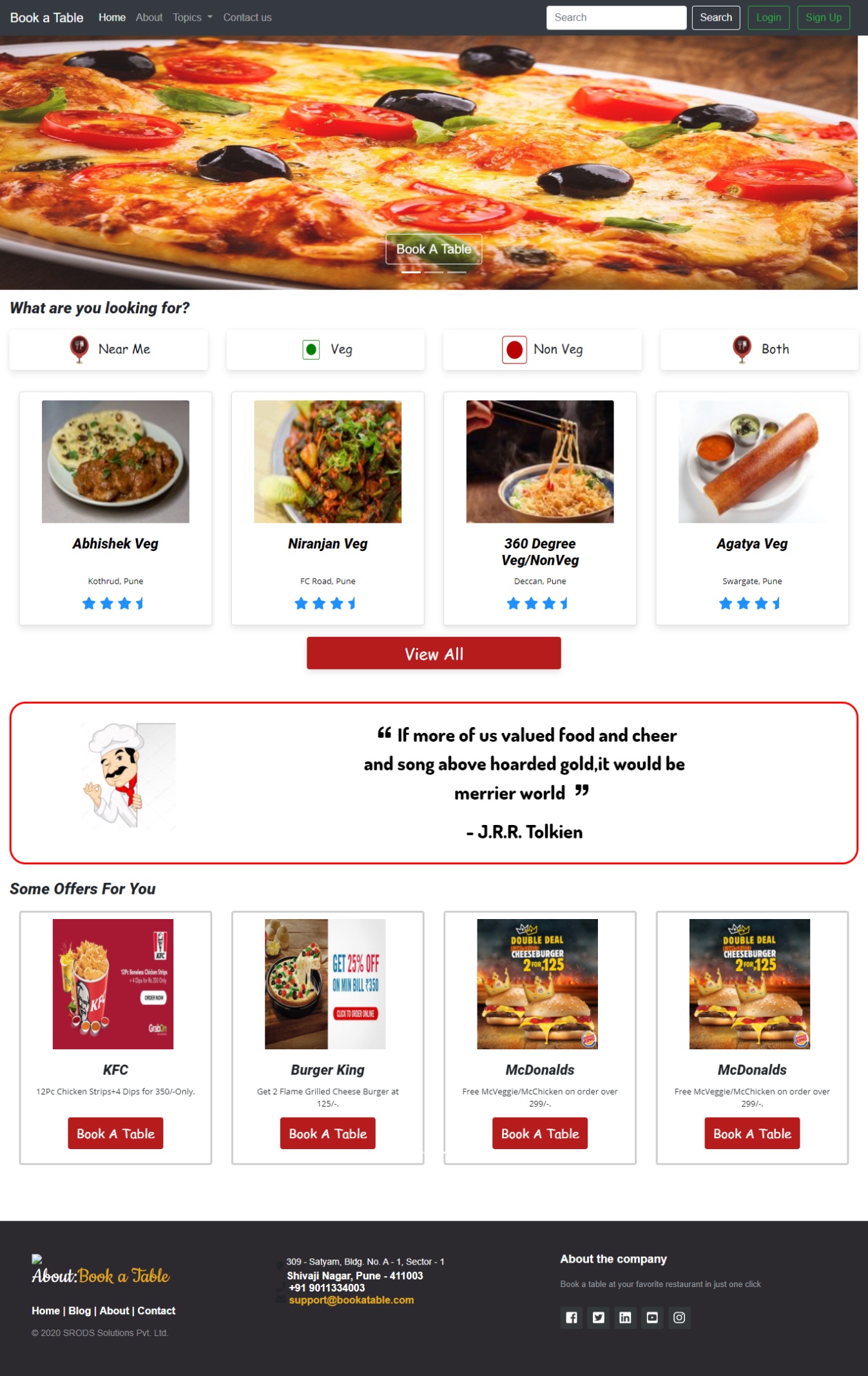
1. **Sign Up**

****

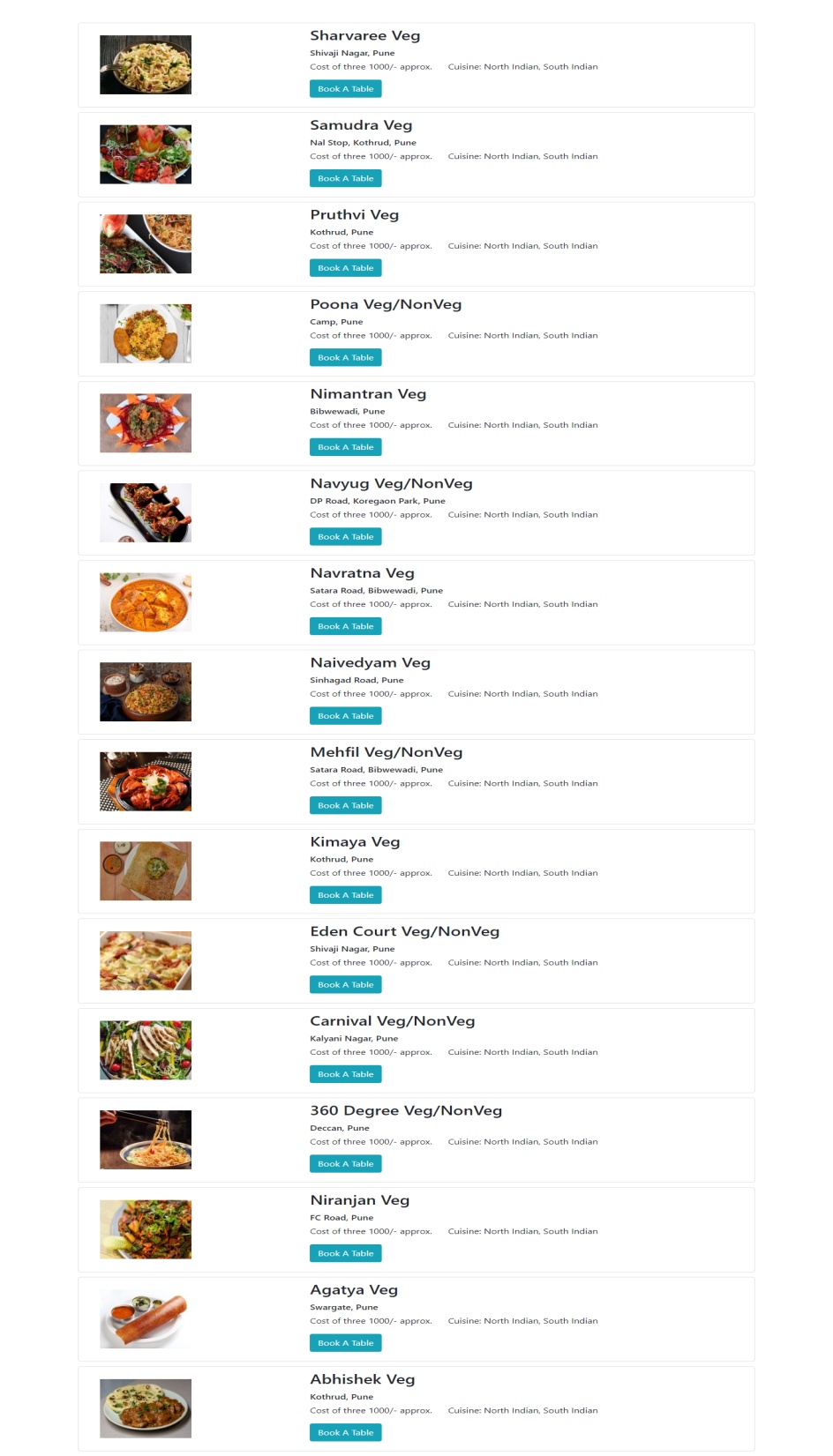
1. **Login**

****

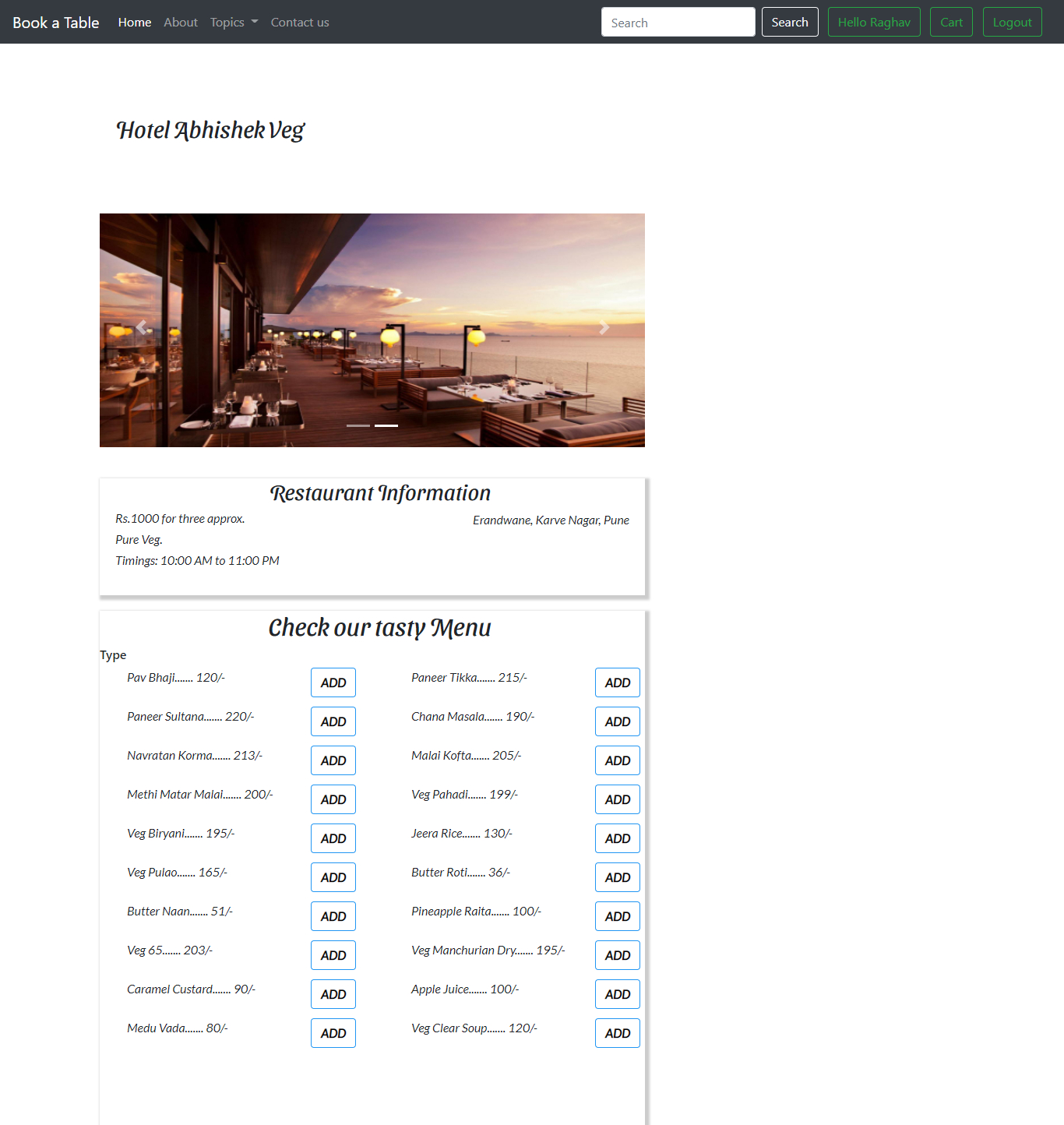
1. **Main Page**

****

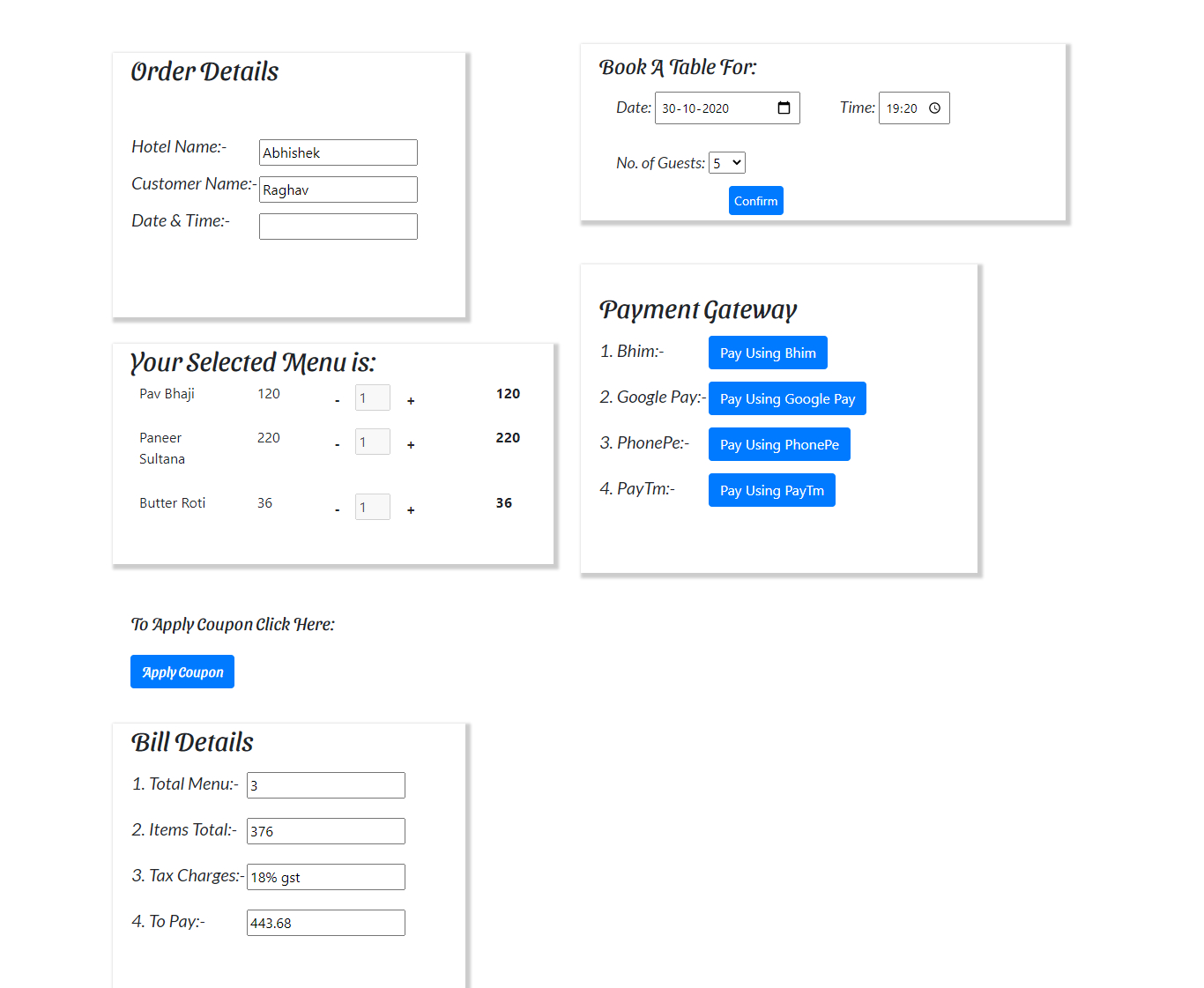
1. **View All Hotels**



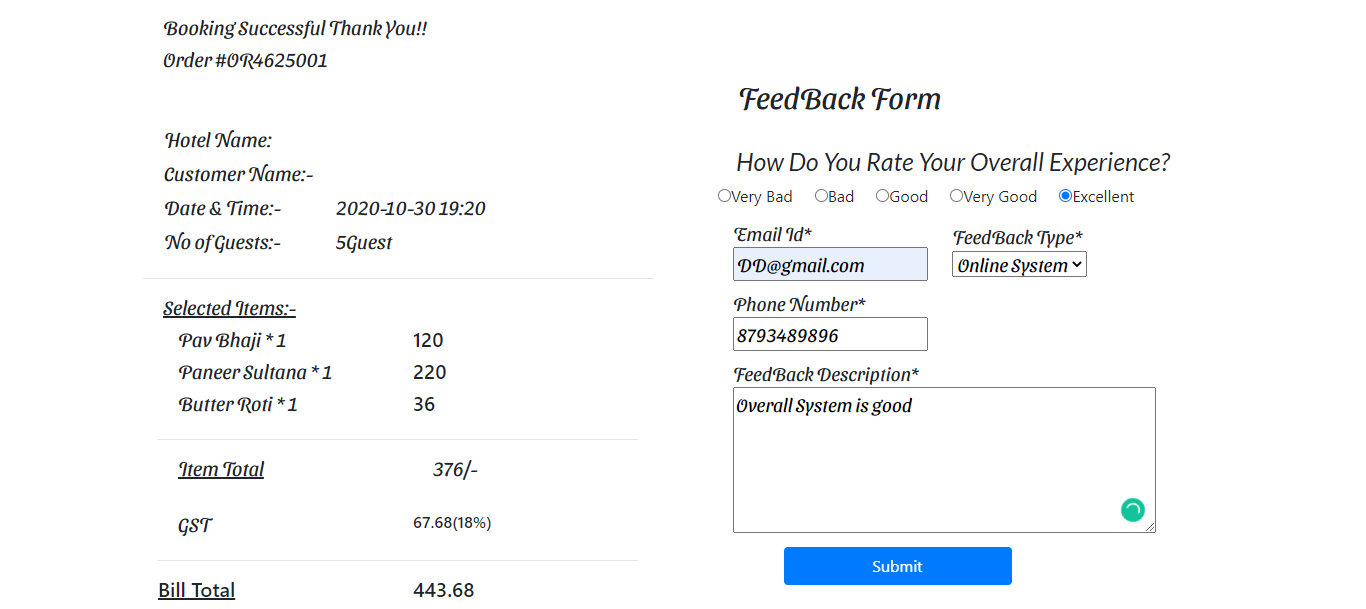
1. **Hotel Page**

****

1. **Book A Table & Bill Details Page**

****

1. **Order Details**

****

**Test Cases**

Test Title: Online Food Ordering and Book A Table

Description: User can order food, book table for a particular hotel.

Test Designed By: Ojas Dugal/Deepen Kakde

Test Designed Date: 17/10/2020

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Scenario** | **Test Case** | **Pre Condition** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (P/F)** |
| TC-001 | Open the browser | Try to open any of the browser | Browser must be open | To check whether browser is getting loaded. | Browser should be present in your OS | Browser should be open | Browser will be opened successfully | Pass |
| TC-002 | Search the website | Try to open the website | Browser must be open | To check whether website is getting loaded | Website should be uploaded on web server | Website should get opened | Website will be opened successfully | Pass |
| TC-003 | Open home page of website | Try to open home page of website | Website must be opened | To check whether home page of website is getting loaded | Home page of website should be opened | Home page should get opened | Home page is opened successfully | Pass |
| TC-004 | Sign Up Functionality | Try to proceed without providing any details | Sign Up Page must be opened | To check whether page is possible to proceed without providing details | All fields should be blank | The page should not proceed without having necessary values | It is not proceeding without necessary details | Pass |
| TC-005 | Sign Up Functionality | Try to proceed with valid Name | Name field must have name of customer | To check whether page is possible to proceed with providing valid Name | Name should be entered | The page should proceed by having valid Name | Page is proceeding | Pass |
| TC-006 | Sign Up Functionality | Try to proceed with invalid Name | Name field must have name of customer | To check whether page is not possible to proceed with providing invalid Name | Name should be entered | The page should not proceed by having invalid Name | Page is not proceeding | Pass |
| TC-007 | Sign Up Functionality | Try to proceed with valid email | Email field must have email of customer | To check whether page is possible to proceed with providing valid Email | Email should be entered | The page should proceed by having valid Email | Page is proceeding | Pass |
| TC-008 | Sign Up Functionality | Try to proceed with invalid Email | Email field must have name of customer | To check whether page is not possible to proceed with providing invalid Email | Email should be entered | The page should not proceed by having invalid Email | Page is not proceeding | Pass |
| TC-009 | Sign Up Functionality | Try to submit details | Submit Button should work | To check whether button is working | Button should be clickable | Button should save details | It is saving the details | Pass |
| TC-010 | Login Functionality | Try to proceed with valid email and password | Email & Password field must have email & password of customer | To check whether page is possible to proceed with providing valid Email & Password | Email & password should be entered | The page should proceed by having valid Email & Password | Page is proceeding | Pass |
| TC-011 | Login Functionality | Try to proceed with invalid email and password | Email & Password field must have email & password of customer | To check whether page is not possible to proceed with providing invalid Email & Password | Email & password should be entered | The page should not proceed by having invalid Email & Password | Page is not proceeding | Pass |
| TC-012 | Login Functionality | Try to login | Login Button should work | To check whether button is working | Button should be clickable | Button should login the user | It is logging in user | Pass |
| TC-013 | Login Functionality | Try to redirect to home page after login successful | Login Button should work | To check whether user is redirected to home page after login successful | Button should be clickable | Login button should redirect user to home page after successful login | It is redirecting user to home page | Pass |
| TC-014 | Home Page Functionality | Try to view all the hotels | List of hotels should be displayed | To check whether user is able you view all hotels | List of hotels viewable | It should display all the hotels to user | It is displaying all the hotels | Pass |
| TC-015 | Home Page Functionality | Try to select the needed hotel | User should select the hotel | To check whether user is able to select the hotel | Hotels should be selectable | Hotel should get selected | It is getting selected | Pass |
| TC-016 | Home Page Functionality | Try to redirect to hotel page after selecting | Select button should work | To check whether user is redirected to hotel page after selecting | Button should be clickable | It should redirect user to hotel page after selecting hotel | It is redirecting user to hotel page | Pass |
| TC-017 | Hotel Page Functionality | Try to add the required menu | Add menu button should work | To check whether user is able to select required menu | Add button should be clickable | It should add the selected menu by the user | It is selecting the required menu | Pass |
| TC-018 | Hotel Page Functionality | Try to select the date for booking a table | Valid date should be selected | To check whether user is able to select date for booking table | Date field should be selectable | User should be able select the date | User is able to select the desired date | Pass |
| TC-019 | Hotel Page Functionality | Try to select the time for booking a table | Valid time should be selected | To check whether user is able to select time for booking table | Time field should be selectable | User should be able select the time | User is able to select the desired time | Pass |
| TC-020 | Hotel Page Functionality | Try to select no. of guests for booking a table | Valid no. of guests should be selected | To check user is able to select guests for booking table | Guests field should be selectable | User should be able to select no. Of guests | User is able to select the no. Of guests | Pass |
| TC-021 | Hotel Page Functionality | Try to redirect to payment page after selecting date, time & no. of guests | Book Button should work | To check whether user is redirected to payment page after selecting date, time & no. of guests | Button should be clickable | Book button should redirect user to payment page after selecting date, time & no. of guests | It is redirecting user to payment page | Pass |
| TC-022 | Payment Page Functionality | Try to select payment gateways for making payment | Payment button should work | To check whether user is able to make payments using various gateways | Buttons should be clickable | Payment button should make payment of various gateways | It is making payment | Pass |

**Scope of Work and Future**

**Enhancements**

The project is basically meant to computerize the existing system for handling  booking information  and so as to gain accuracy, processing speed, data consistency, integrity, better error handling.

Scope of a System where user can easily book a table online, in order to avail table booking related services and admin of system can add services in the database when a restaurant increases number of services. When user books a table, receives the confirmation of booked table in the form of message/email.

Future scope is where staff will get notified when any table get allotted to customer in order to avail service and also admin can delete the service which Book A Table is no longer using.

This project has a large scope as it has the following features which help in making it easy to use, understand and modify it.

1. Automation of table booking Procedure.
2. No Need to do Paper Work.
3. To save the environment by using paper free work.
4. To increase the accuracy and efficiency of system.
5. Management of user's Data and services.

**Conclusion**

Table Booking System enables customers for booking a table with ease. A System where user can easily book a table online in order to avail hotel/restaurant related services and admin of the system add new restaurant’s to the system. It saves a lot of time and paperwork which results in performance and soundness along with time management.

So, overall this system will give the customers a better user friendly environment to book a table online.