Restaurant Management System Using Chatbot

BY

Zunaitha.B

(19PCS020)

Project Report Submitted

In partial fulfillment of the requirements for the Award of

Master's Degree in Computer Science

DEPARTMENT OF COMPUTER SCIENCE

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND

HIGHER EDUCATION FOR WOMEN (Deemed to be University)

COIMBATORE – 641043

APRIL - 2020

Restaurant Management System Using Chatbot

BY

Zunaitha.B

(19pcs020)

Project Report Submitted

In partial fulfillment of the requirements for the Award of

Master's Degree in Computer Science

DEPARTMENT OF COMPUTER SCIENCE

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND

HIGHER EDUCATION FOR WOMEN (Deemed to be University)

COIMBATORE – 641043

Signature of the
Head of the Department

Signature of Supervisor

Viva-voce Examination Held on _____

Signature of Examiners

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I would like to express my sincere thanks to **God Almighty**, for his constant love and grace that he has showed upon me, which kept me in good health, and sound mind without which my project would not have reached a successful end.

I would like to express my deep sense of reverential gratitude and sincere thanks to **Prof.S.P.Thyagarajan,D.Sc,PhD,M.D,Chancellor,** Avinashilingam Institute of Home Science and Higher Education for Women, Coimbatore, for the opportunity given to me for undertaking this study and for providing all the needed facilities during the course of my study..

I owe my great deal of gratitude to **Dr. Premavathy Vijayan, M.Sc., M.Ed., Dip.Spl.Edn., M.Phil., Ph.D., Vice Chancellor**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for extending all resources that facilitated the conduct of the present study.

I express my gratitude to **Dr. S. Kowsalya, Registrar, M.Sc., M.Phil., Ph.D** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all facilities necessary for the study.

I would express my boundless thanks to **Dr. K. Udaya Chandrika, M.Sc., M.Phil., Ph.D., Dean, School of Physical Sciences & Computational Sciences,** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for granting the facility required.

I wish to place on record my deep sense of gratitude to **Dr. Vasantha Kalyani David, M.Sc., M.Phil(Maths)., M.Phil(CS)., Ph.D, Professor and Head, Department of Computer Science** for support and encouragement to complete the project.

I express my heart full gratitude to my esteemed mentor **Dr. G.Geetha, B.E, M.Tech., M.Phil., Ph.D., Senior Technical Assistant,** Department of Computer Science, for imparting the tremendous assistance and

well-timed support for triumph of our project with guidance and constant supervision as well as for providing necessary resources for the project and also for her support in completing the project.

I am grateful to the project coordinator **Dr. G. Sudhamathy, M.C.A., Ph.D., Assistant Professor,** Department of Computer Science, who was instrumental in granting me the facilities required for doing project.

Finally, yet importantly, I would like to thank my **parents, family members and friends** for their kind inspiration, support, encouragement, blessings and prayers, which were instrumental in the successful completion of the project.

I have great pleasure in expressing my deep sense of gratitude to all other teaching and non-teaching staff members of the Department of Computer Science, who stood behind the screen for the completion of the project.

I would extend my hearty thanks to one and all that helped me directly or indirectly for the successful completion of my project.

ABSTRACT

ABSTRACT

A Chat-bots aims to make a conversation between both human and machine. The machine has embedded knowledge to identify the sentences and making a decision itself as a response to answer a question. Chat-bots will be completely based on a text-based user interface, allowing the user to type commands and receive text as well as text to speech response. Chat-bots are usually stateful services, remembering previous commands in order to provide functionality. It can be utilized securely for even a larger audience, when the chat-bot's technology is integrated with popular web services. The project entitled Restaurant Management System, manages restaurant food ordering and reservation system. This system was developed to facilitate customers approaching restaurant for online food ordering and table reservation. Admin/Customer can login with username and password. After successful login and registration the users can access the booking page. The User can ask the question any hotel-related activities through the chat-bot. The System analyses the question and then answers to the user. With the help of artificial intelligence, the system answers the query asked by the customer. The system replies using an effective Graphical User Interface as if a real person is talking to the user. In the proposed system the user can search for the menu according to his choice i.e. according to price range and category of food. Food Items will be managed by chief user. This system is developed to automate day to day activities of the restaurant. All the reports are updated daily in the restaurant like customer food order report, table booking, feedback and ratings. This is developed using by Python as a front end and back end as MySQL.

TABLE OF CONTENTS

CONTENTS

S.NO	PARTICULARS	PAGE NO
1.	INTRODUCTION	1
	1.1 Problem Definition	1
	1.2 Objective of the Project	2
2.	SYSTEM SPECIFICATION	4
	2.1 Hardware Specification	4
	2.2 Software Specification	4
	2.3 About the Software	5
	.3.1 PYTHON LANGUAGES	5
	2.3.2 FLASK FRAMEWORK	6
	2.3.3 RBDMS TERMINOLGY	7
	2.3.4 MYSQL	8
3.	NEED FOR THE STUDY	9
	3.1 Existing System	10
	3.2 Proposed System	10
4.	SYSTEM DEVELOPMENT	12
	4.1 Module Description	13
	4.2 System Design	14
	4.2.1 DFD flow diagram	14
	4.2.2 Input design	16
	4.2.3 database design	17
	4.2.4 Output design	18
	4.2.5 System testing	19
5.		53
6.	CONCLUSION	22
7.	SCOPE FOR FUTURE ENHANCEMENT	24
8.	BIBLIOGRAPHY	26
9.	APPENDIX	25
10	SCREENSHOTS	61
11.	CODING	48

INTRODUCTION

INTRODUCTION

Restaurant is a kind of business that serves people all over world with readymade food. This system is developed to provide service facility to the customer and also in the ease functionality of restaurant. This restaurant management system can be used by employees in a restaurant to handle the clients, their orders and can help them easily find free tables. The services that are provided is food ordering and reservation table management by the customer through the system online, customer information management and waiter information management, menu information management and report. Main objective of building the system is to provide the customer, the facility of ordering and reservation service by online.

1.1 Problem Definition

Nowadays Amazon, Flipkart, Myntra are managing their business manual especially by taking customer orders. In traditional booking system, a customer has to go to restaurant or make a phone call in order to get his meal reserved. Today, restaurant waiter takes the customer ordering by manual system with using paper. Customer does some formal conversation like hello, hi, etc. Then he demands for today's menu and do some discussion over menu items then he orders. It takes 5 to 10 minutes to book the order and waiter book the order on paper so there is probability of lost and duplication of customer information. Restaurant management system puts the order in a queue 1with specific priority according to time and quantity, and then a cook is assigned for the specific order to complete it.

Besides, the restaurant waiter information also by manual system kept use paper and this is difficult for restaurant administrator to find waiter information, probability missing the paper and difficult to arrange the schedule. Initial problem is that the customer has to get connected over the phone; it would be harder if the restaurant is very popular and busy. Sometimes, waiter information and customer information is important to restaurant administrator for reference in the future. The chances of committing mistakes at the restaurant side in providing a menu list for a specific time would be more.

Many people have experienced going to a restaurant where the service is poor and there is a lack of attention from the waiter staff. The paper menus can be flimsy, hard to navigate, and outdated. To leverage the growing mobile industry, the on –line restaurant proffers solution. This restaurant menu and management system will replace the paper waste, is more maintainable, and allows for greater customer engagement. The problem confronting the research is to determine the Documentation for online restaurant management system.

1.2 Objective of the Project

The main point of developing this system is to help restaurant administrator manage the restaurant business and help customers for online ordering and reserve table. In proposed system, user can search for a menu according to his choice i.e. according to price range and category of food and later he can order a meal.

SYSTEM SPECIFICATION

SYSTEM SPECIFICATION

2.1 HARDWARE REQUIREMENTS

- Processor : i3
- Hard disk : 500 GB
- Mouse : Logitech.
- RAM : 4GB(minimum)
- Keyboard : 110 keys enhanced.

2.2 SOFTWARE REQUIREMENTS

- Front End software : PYTHON
- Back End database : MYSQL
- Framework : FLASK
- Design : HTML
- IDE Tool : VSC

2.3 SOFTWARE DESCRIPTION

2.3.1 PYTHON LANGUAGE INTRODUCTION

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

- There are two major Python versions- Python 2 and Python 3. Both are quite different.
- Windows: There are many interpreters available freely to run Python scripts like IDLE (Integrated Development Environment) which is installed when you install the python software from http://python.org/
- Linux: For Linux, Python comes bundled with the linux.

LANGUAGE FEATURES

Interpreted

- There are no separate compilation and execution steps like C and C++.
- Directly run the program from the source code.
- Internally, Python converts the source code into an intermediate form called bytecodes which is then translated into native language of specific computer to run it.
- No need to worry about linking and loading with libraries, etc.

Platform Independent

- Python programs can be developed and executed on multiple operating system platforms.
- Python can be used on Linux, Windows, Macintosh, Solaris and many more.

Free and Open Source: Redistributable

High-level Language

- In Python, no need to take care about low-level details such as managing the memory used by the program.

Simple

- Nearer to English language hence easy to learn
- More emphasis on the solution to the problem rather than the syntax

Embeddable

- Python can be used within C/C++ program to give scripting capabilities for the program's users.

Robust:

- Exceptional handling features
- Memory management techniques in built

Rich Library Support

- The Python Standard Library is vary vast.
- Known as the "batteries included" philosophy of Python ;It can help do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, email, XML, HTML, WAV files, cryptography, GUI and many more.
- Besides the standard library, there are various other high-quality libraries such as the Python Imaging Library which is an amazingly simple image manipulation library.

2.3.2 FLASK FRAMEWORK

If one has to develop a web app in Python, then taking chances of leveraging a framework is highly adequate. A framework is a code library that makes a developers life easier when building reliable, scalable, and maintainable web applications by providing reusable code or extensions for common operations. There are a number of frameworks for Python, including Flask, Tornado, Pyramid, and Django and the right choice for a novice developer is perplexing.

Flask is a relatively new framework was undertaken for the current project due to few of the

reasons. It has taken the Python web development community by storm: in a short time it became one of the most popular frameworks around. It offers a lot of flexibility and clean code with a lot of extensibility. Option of not becoming dragged down by a huge framework is that it articulates what has to be done by adopting the ease, productive, and creative environment.

2.3.3 RDBMS TERMINOLOGY:

Create a database, the required fields are restaurant_table, user_table, menu_table, booking_table , feedback_table . According to the attributes, the fields are entered and the values are stored in MySQL.

Before we proceed to explain MySQLdatabase system, let's revise few definitions related to database.

- Database: A database is a collection of tables, with related data.
- Table: A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- Column: One column (data element) contains data of one and the same kind, for example the column postcode.
- Row: A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- Redundancy: Storing data twice, redundantly to make the system faster.
- Primary Key: A primary key is unique. A key value can not occur twice in one table. With a key, you can find at most one row.
- Foreign Key: A foreign key is the linking pin between two tables.
- Compound Key: A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- Index: An index in a database resembles an index at the back of a book.
- Referential Integrity: Referential Integrity makes sure that a foreign key value always points to an existing row.

2.3.4 MySQL:

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and **flexibility** of use. We can easily add ,delete or modify the tables as well as our prospective. It provides the food order database design for restaurants. It also covers the tracking of table booking and orders placed by the customers.It can be used for online booking of the tables and pre-order before reaching the restaurant. The security can also be handled by following RBAC Database in MySQL.

- The main purpose for devolping the moduleis to mangage the food item data wise.
- So all the food item will be managed by admin and customer able to see the food item.
- Admin can also see list of all the food items according to the customers.

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

We use Relational Database Management Systems (RDBMS) to store and manage huge volume of data.

- Enables to implement a database with tables, columns and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.
- Interprets an SQL query and combines information from various tables.

- MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:
- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

NEED FOR THE STUDY

SYSTEM STUDY

System study is classified into two types

- Existing system and
- Proposed system

3.1 EXISTING SYSTEM

The current scenario in the Restaurant is waiting for a table to open up and for the waiter to seat them, receiving a non-interactive paper menu displaying limited information about the food available (typically containing only the dishes' names, prices, and brief descriptions) . Waiting for a waiter to arrive to them before customers can order, receiving no information about the progress of their meal while waiting for it to arrive. Waiting once more for the waiter to provide their bill to pay, and finally no automated way for customers to directly provide feedback to the restaurant's management immediately after finishing their meal.

DISADVANTAGES

1. Financial Risk
2. Time Commitment

3.2 PROPOSED SYSTEM

The WEB APP supports a variety of smartphones, and it provides assistance at various stages of the entire dining experience.

The customer enquiries for the current day's menu and does some discussion with the waiter over menu items before he orders.

So there is probability of lost and duplicate customer information

Restauarant Management System RMS capital everywhere

ADVANTAGES

1. No misunderstandings and no frustrations.

2. Online food ordering will be opened 24/7.

3. Online menu is simpler and easy to operate by a user.

4. Number of users or customer’s increases.

5. It is Responsible for faster growth of your business over the internet.

SYSTEM DEVELOPMENT

MODULES

>> Admin Module

>> Restaurant Module

>> User Module

4.1 MODULES DESCRIPTION

Admin Module

- ✓ The admin can login with username and password.
- ✓ They can add and view the restaurant
- ✓ They can also view the feedback

Restaurant Module

- ✓ The restaurant can login with username and password
- ✓ They can view the Table orders
- ✓ They can add and view the tables

User Login

- ✓ Users can login with email id and password.
- ✓ They can select the restaurant.
- ✓ They can select & book the table & chairs.
- ✓ Book the table with the help of chatbot
- ✓ They can also give their feedback.

SYSTEM DESIGN

4.2.1 DATA FLOW DIAGRAM

A data flow diagram is graphical tool used to describe and analyse movement of data through a system. These are the central tool and the basis from which the other components are developed.

The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams.

The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels.

Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

DFD SYMBOLS:

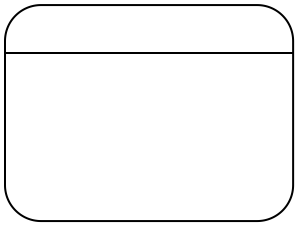
In the DFD, there are four symbols

1. **A square** defines a source(originator) or destination of system data

2. An **arrow** identifies data flow. It is the pipeline through which the information flows

3. A **circle or a bubble** represents a process that transforms incoming data flow into outgoing data flows.

4. An **open rectangle** is a data store, data at rest or a temporary repository of data



Process that transforms data flow.

Source or Destination of data

Data flow

Data Store

Customer

This user will register to be a member to use the online system of this online restaurant management system. This online ordering divided into two type of customer; it is customer dine-in ordering and takeaway ordering. For dine in ordering, customer will view menu, make online ordering and make a reservation table. This system allows customer to place an order or allow him to cancel the book order.

Admin

Administrator or manager is the person who will manage the entire system. he is allowed to reassign the cook according to his priority, he can edit the menu information such as its price, items available currently, etc. This type of user will also do maintenance and control the application of this system. A manager can reassign the cook for a specific order or an item. Administrator takes a responsibility to register new customer, register new

waiter, register new menu into database. A manager can edit/create some or whole part of the menu record on daily basis. That is by changing the menu items, prices, description, etc.

Customer Registration Module

Customer registration module contains customer's information such as customer personal information and other information related to that customer. Then, all of this information recorded into database. Customers are given with a facility to change his existing password.

Customer Online Ordering And Reservation Module

Customer online ordering and reservation module provides a form that needs to be fulfilling in term of ordering food and reservation table via online.

Feedback Module

Based on food or everything about the restaurant, customer can send any suggestion or comment to the restaurant with feedback form. From this form, side of restaurant will know their weaknesses and strengths.

Menu Module

Menu module is food that restaurant prepared for customer. This module, customer can view the menu and make decision for order.

4.2.1 INPUT DESIGN

The Input design in our project is these constituted using the Menu_table, Price_tag and Table_booking.

The input design is the link that ties the Information system into the world of its users. It is a process of converting user-originated inputs to a computer based format. Input data are collected and organized into a group of similar data. Once identified, appropriate input media are selected for processing.

The goal of designing input data is to make entry easy, logical and free form errors. The source document is designed that capture the data and then select the media used to enter them into the computer. The input forms are developed in a user-friendly way so that a layman also can easily understand everything. Menus are provided to users and different icons are designed so the proposed system design looks decorative.

Input design is the part of the overall system design. Source documents initiate a processing cycle as soon as they are entered into the system through the keyboard. A source should be logical and easy to understand.

Objectives of input Design:

- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

4.2.3 DATABASE DESIGN

Databases are structures that hold data. The software that enables the flow of data through these structures is called database management system or DBMS. The most widely used system of DBMS is something called RDBMS or Relational DBMS. This simply means
that the data is stored in tables, moreover. Whatever relationships that exist within the data are stored Within tables.

To put it simply, there are three parts that make a database:

Tables:

We all know what tables are - a matrix of rows and columns. In databases, it's the same. Each row is a record, or a unit of data. A record (row) can have several columns or fields. Each field is like an attribute of that record.

Queries :

Query is a question posed to the database, to retrieve a specific set of records, based on conditions supplied in the query.

Views:

These are virtua1 tables, or (a set of) stored queries. At a physical level, the data is stored in data files specific to the DBMS. Examples of modern-day RDBMSs that are wide1y used include Oracle, MySQL, etc. Oracle is the largest commercially available RDBMS and MySQL (earlier acquired by Sun, and subsequently by oracle) is a free and open source RDBMS that is very Well-known.

Types of SQL statements:

- **SQL:(STRUCTURED QUERY LANGUAGE)**

Statements can be classified into the following four categories:

- **DDL (Data Definition Language):**

  This class of statement is used to create/destroy or define/change database components.

  Examples: CREATE, DROP, RENAME.

- **DML (Data Manipulation Language):**

As the name suggests, these statements manipulate the data itself, and the views related to it.

Examples: SELECT, INSERT.

- **DCL (Data Control Language):**

  These statements control access to the data. There are only two of them: GRANT and REVOKE.

- **TCL (Transaction Control Language):**
  - Transactions are handled using these statements.
  - Examples: COMMIT, ROLLBACK.

4.2.4 OUTPUT DESIGN

The Output design in our project is these constituted using the feedback_table and Rating.

Output forms are designed in a specific manner as per the user requirement. Results are formatted to enhance clarity. Depending on the user the system would generate appropriate output. The output forms are designed in such a way that the entire user required data is presented..

What all things you considered in designing output.

- Determine what information to present.
- Decide whether to display, print or speak information and select the output medium.
- Arrange the presentation of information in an acceptable form.
- Decide how to distribute the output to intended users.

SYSTEM TESTING

It is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems.

4.2.5 System Testing

It is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested.

System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS).

System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartial. It has both functional and non-functional testing.

Need for Testing

1. To click the efficiency of the system
2. To remove the errors of the system
3. To check whether the objectives of the project is accomplished
4. To enable the removal of complexities
5. To check the user-friendliness of the system
6. To check the flexibility of the system.

TYPES OF TESTING

INTEGRATION TESTING

System Integration Testing is defined as a type of software testing carried out in an integrated hardware and software environment to verify the behavior of the complete system. It is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirement.

System Integration Testing (SIT) is performed to verify the interactions between the modules of a software system. It deals with the verification of the high and low-level software requirements specified in the Software Requirements Specification/Data and the Software Design Document.

It also verifies a software system's coexistence with others and tests the interface between modules of the software application. In this type of testing, modules are first tested individually and then combined to make a system.

In our project, software and/or hardware components are combined and tested progressively until the entire system has been integrated. (i.e) Admin and user must be combined for interaction process .

VALIDATION TESTING

Validation testing is the process of testing the output for various inputs. The inputs are classified into string inputs. Integer inputs, decimal inputs. All the inputs were tested for all these three inputs and their behaviours were tested. The behaviours lead to the effecting error handling coding. The error handling coding includes a message box for all types of errors.This testing includes the testing of several of values.

The validation testing is tested for all adding and modification functions. The modifications of primary key values were also tested. While modification the testing was done whether the system accepts duplicate values for primary key and the errors displayed there in.The validation testing leads to the final system testing. This includes testing of various conditions and removing all standard values and providing actual outputs.

SYSTEM TESTING

System testing specifically goes after behaviours and bugs that are properties of the entire system as distinct from properties attributable to components (unless, of course, the component in question is the entire system).

Examples of system testing are Recourse loss bugs, throughput Bugs, performance, security, recovery, transaction bugs, performance, security, recovery, transaction synchronization bugs (Often misnamed “timing bugs”).

SYSTEM IMPLEMENTATION AND MAINTANENCE

SYSTEM IMPLEMENTATION

Implementation is the stage when the theoretical design has been converted into a working system. The implementation phase is used to test the developed package with sample data, correct the error identified, appearing the user of the various special facilities and features of the computerized system. It also involves user training for minimizing resistance to change and giving the system a chance to prove its worth. The successful implementation of people working on the system.

The implementation process included the explanation of the benefits of the system. In this stage the feedback from the users are also taken into considerations and the possible suggestions for solving their Problems are discussed. These feedbacks are used further for the next version of the project in future.

Validation and Verification

In our project the verification and validation are performed, once the user has registered and they login into the website after few moments they have a username and password to use the portal. . Both verification and validation is carried out to check whether the password is correct or not, after which the registered person has the rights to access the portal.

The verification and validation are to assess and improve the quality of the work generated during development and modification of software. Quality attribute of interest include correctness, completeness, consistency, reliability, usefulness, efficiency, conforms to standard and overall cost effectiveness. The verification of the proposed system is one of the final stages of implementation where the computerized output is compared with that of the manual system. The stage compared are made visible to the users and this makes the successful implementation of the project.

The valuable inputs lay the platform for the revision of the new system. The requirement of the customer is compared with the software behavior and their outputs.

CONCLUSION

CONCLUSION

The need for table food ordering is analyzed and its advantages over the traditional food ordering system in restaurants are studied. The proposed online restaurant management system is time saving and error free as compared to the traditional System. This system attracts customers and also adds the efficiency of maintaining the restaurant's ordering and billing. Hence it is the modern way to grow up the business using E-commerce. Here implementation of an advanced e-restaurant Menu ordering system using smart android mobile phone. This system entirely reduces the unnecessary time. Every order is Associated with an individual seat at the table, and orders are built one customer at a time, just like on paper, but with greater Accuracy. Items can also easily be shared by the whole table, moved or modified, and noted and the cost can be calculated in real time. The idea of the advanced e-restaurant is to provide extended facility of using GPRS module. GPRS module can be used to monitor and request the menu ordering from table which will be directly sent to the predefined web link for process of billing the items purchased.

SCOPE FOR FUTURE ENHANCEMENT

SCOPE FOR FUTURE ENHANCEMENT

In addition to the unfinished requirements, there are other possibilities of further improving the project. The improvements may include:

1. Presenting graphical floor plan for table management and reservation.
2. Support food order delivery and driver tracking.
3. Extension of pricing methods for individual or multiple recipes.
4. Advanced inventory control with material storage and expiry information.
5. Managing customer loyalty membership and discount voucher.

BIBLIOGRAPHY

BIBLIOGRAPHY

Introduction to python: https://www.w3schools.com/python/

- HTML tags: https://www.w3schools.com/html/
- Flask: https://en.m.wikipedia.org/wiki/Flask_(web_framework)
- CSS: https://developer.mozilla.org/en-US/docs/Web/CSS
- Matthias, H," Models of software development", Retrieved on 5-OCT-2012, from http:/www.ccs.neu.edu/home/Metthias/670-s05/lectures/2.html.
- Pressman, R 2001, Software Engineering, 5th edition, McGraw-Hill, New York.
- Ross D.T., J.B. Good enough and C.A. Irvine, Software engineering: Process, Principals, and goals. COMPUTER 8(5) (May 1975): 17-27.
- Cartwright, A. (2008, November 1). Beyond the paper chase. Law Enforcement Technology, 35(11), 58, 60-62.
- Geiger, B. (2004, March 4). Citizens reporting crimes online: the Son Francisco experience. The Police Chief, 74(8)

"Programming in Python 3",Developer's library, "Second Edition", Pears Education, 15 April, 2018.

"Core Python Programming", Dr.R.Nagewara Rao, "Second Edition", Dreamtech press, 1 Jan, 2018.

"python bpb", Aditya kanetkas, yashavant kanetkas, Publishes BPB, 1 Jan 2019.

5. "Python Programming For The Absolute Beginners", Michael Dawson, Premier Press, 11 Aug, 2003.

APPENDIX

APPENDIX

A. DATA FLOW DIAGRAM

LEVEL – 0 DFD DIAGRAM

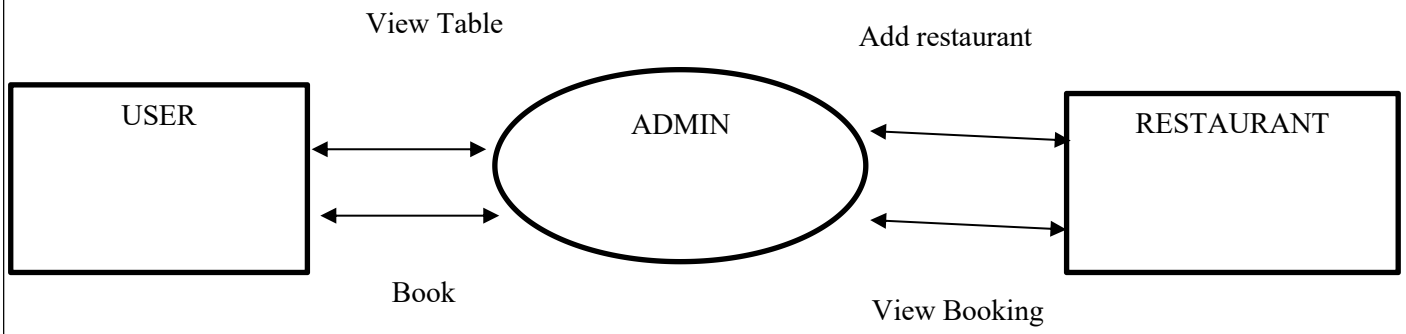
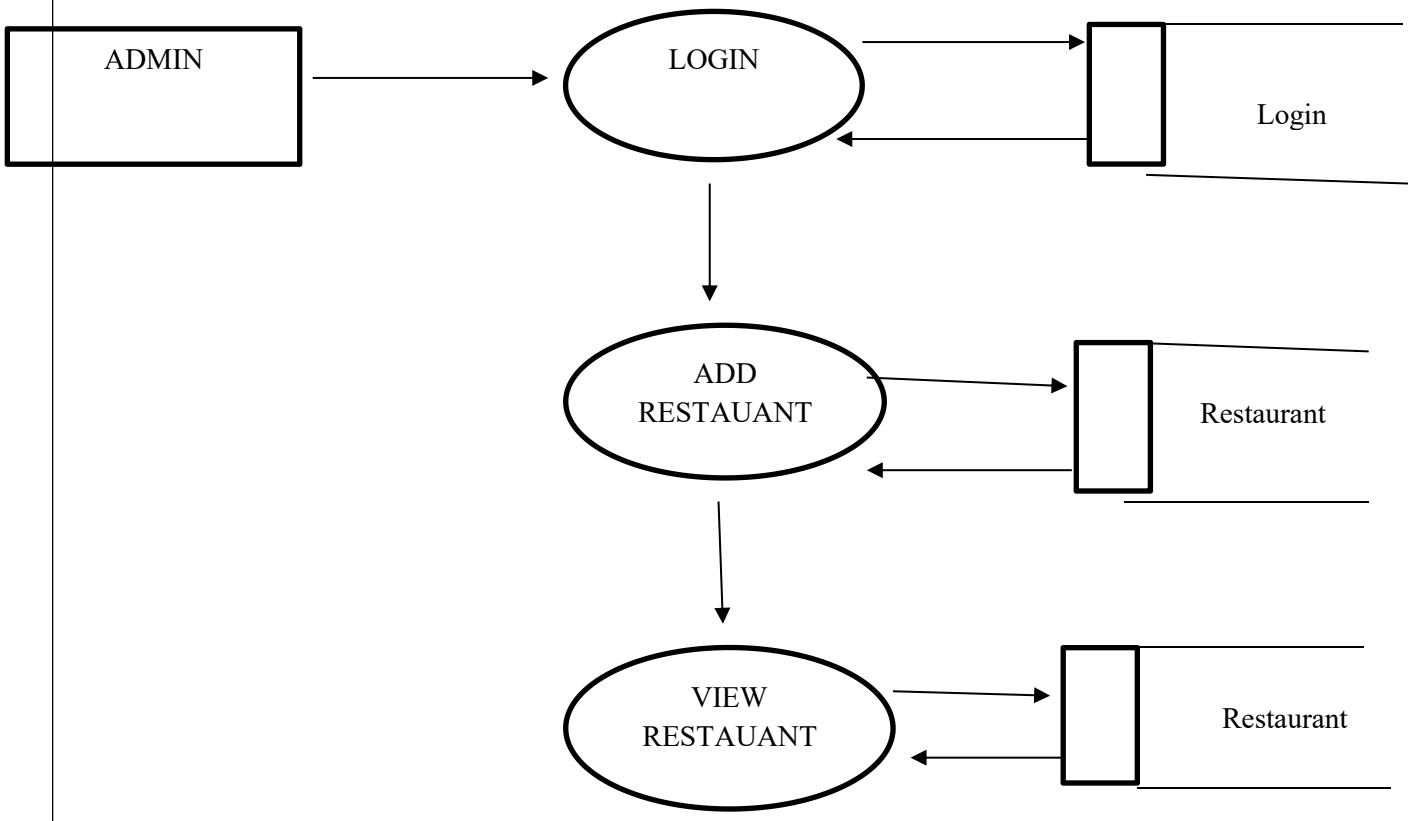


Fig-1.0

LEVEL – 1 DFD DIAGRAM



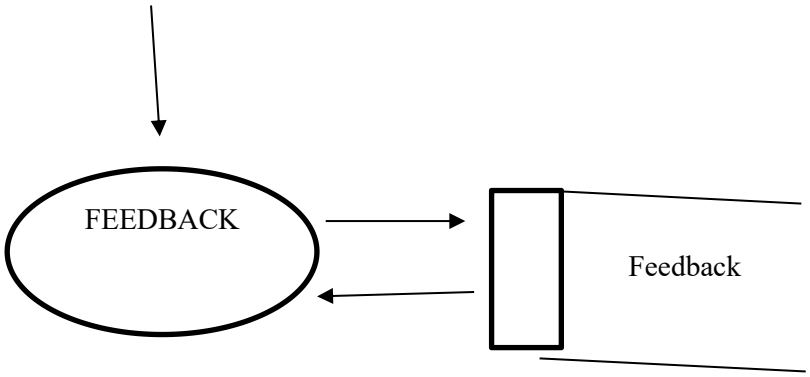


Fig-1.1

LEVEL – 2 DFD DIAGRAM

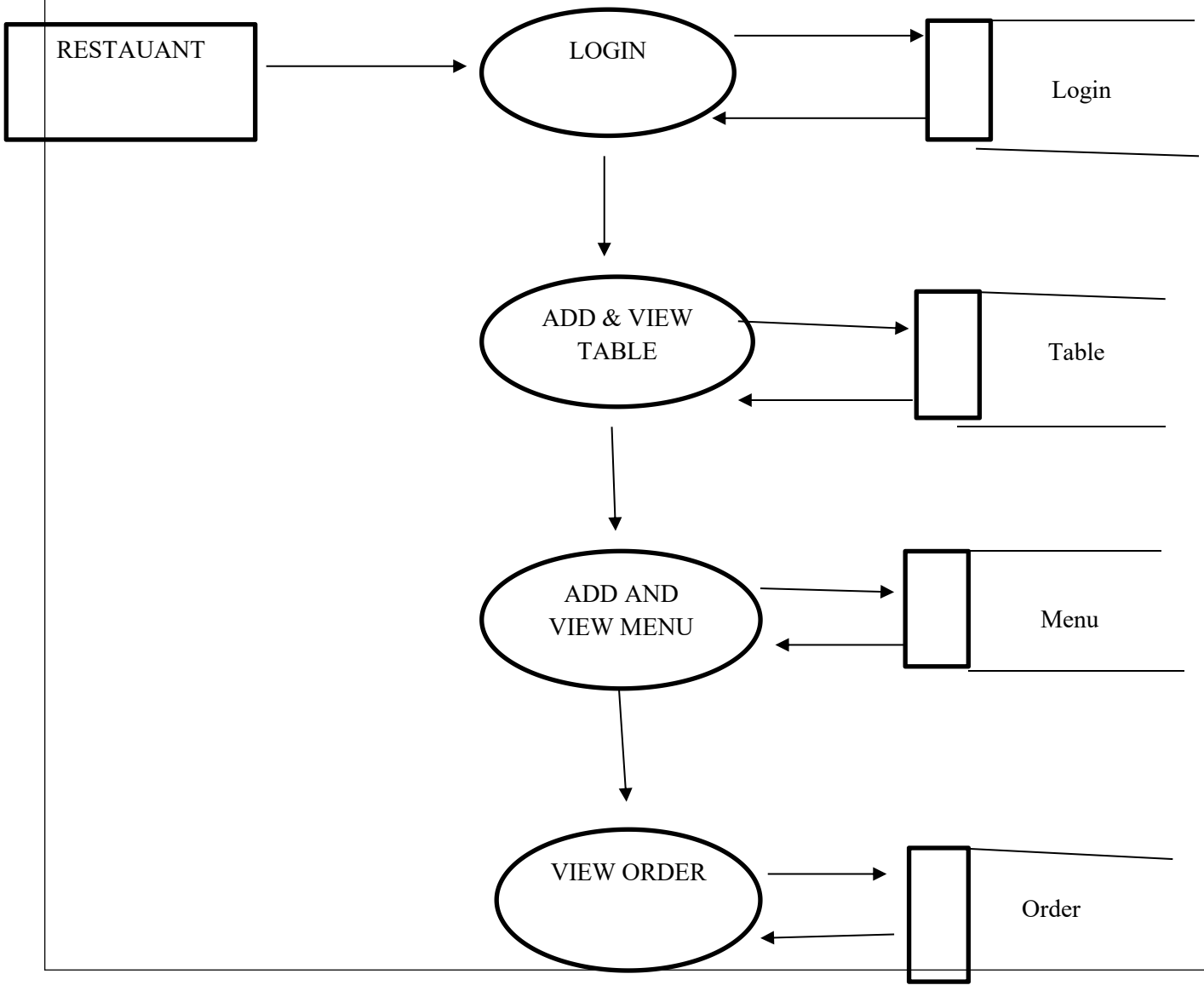


Fig-1.1

LEVEL – 1 DFD DIAGRAM

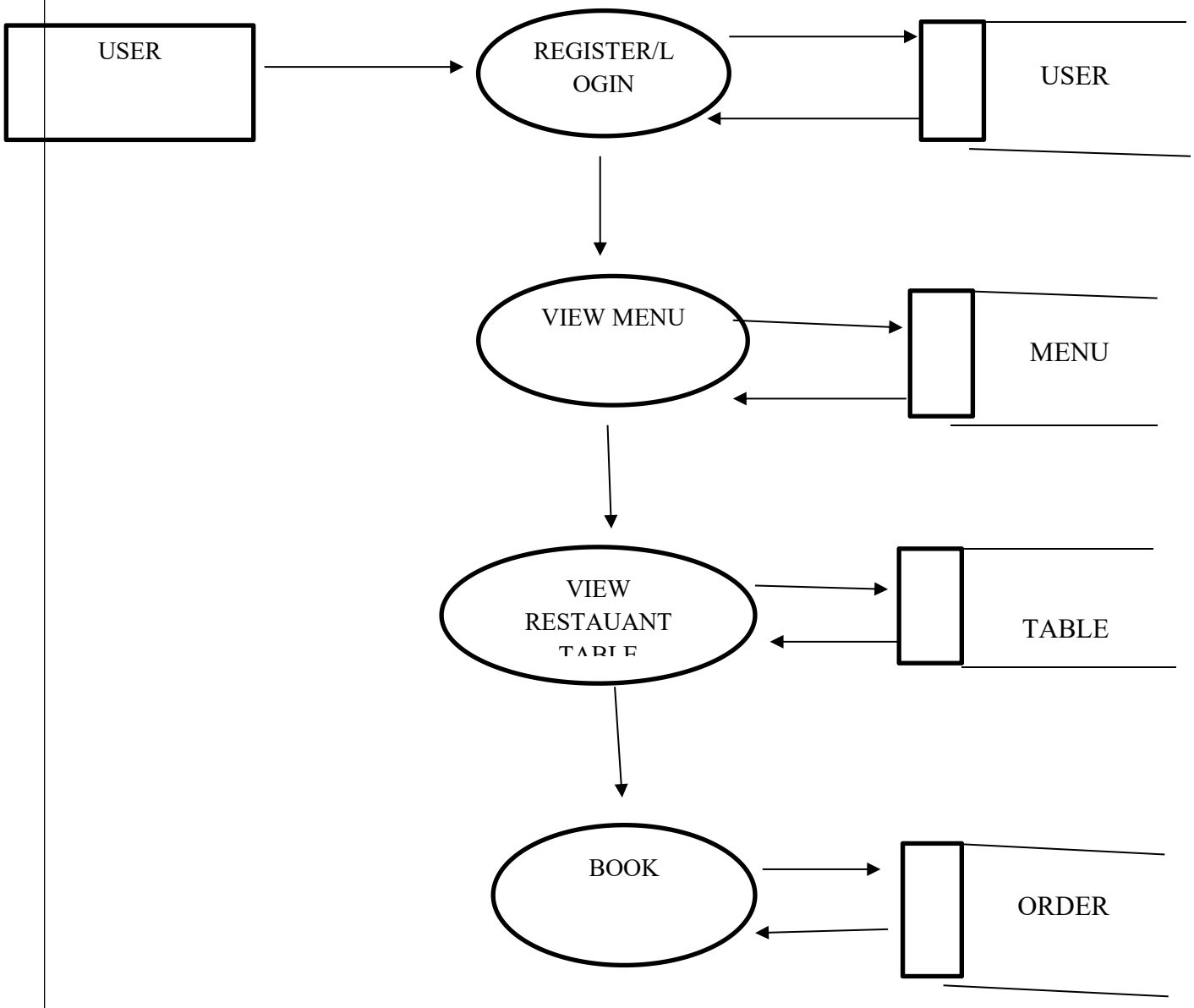


Fig-1.1

B. DATABASE DESIGN

Tb: Restaurant

Attribute	Type
Id	Int(10) Primary Key
Restaurant Name	Varchar (100)
Email	Varchar (100)
Username	Varchar (100)
Password	Varchar (100)
Address	Varchar (100)
Desp	Varchar (100)

Tb: Feedback

Attribute	Type
Id	Int(10) Primary Key
Name	Varchar (100)
Feedback	Varchar (100)
Rating	Varchar (10)

Tb: Menu

Attribute	Type
Id	Int(10) Primary Key
Restid	Int(10) Foreign Key
Foodname	Varchar (100)
Price	Varchar (10)
Category	Varchar (100)
Status	Varchar (10)

Tb: Table

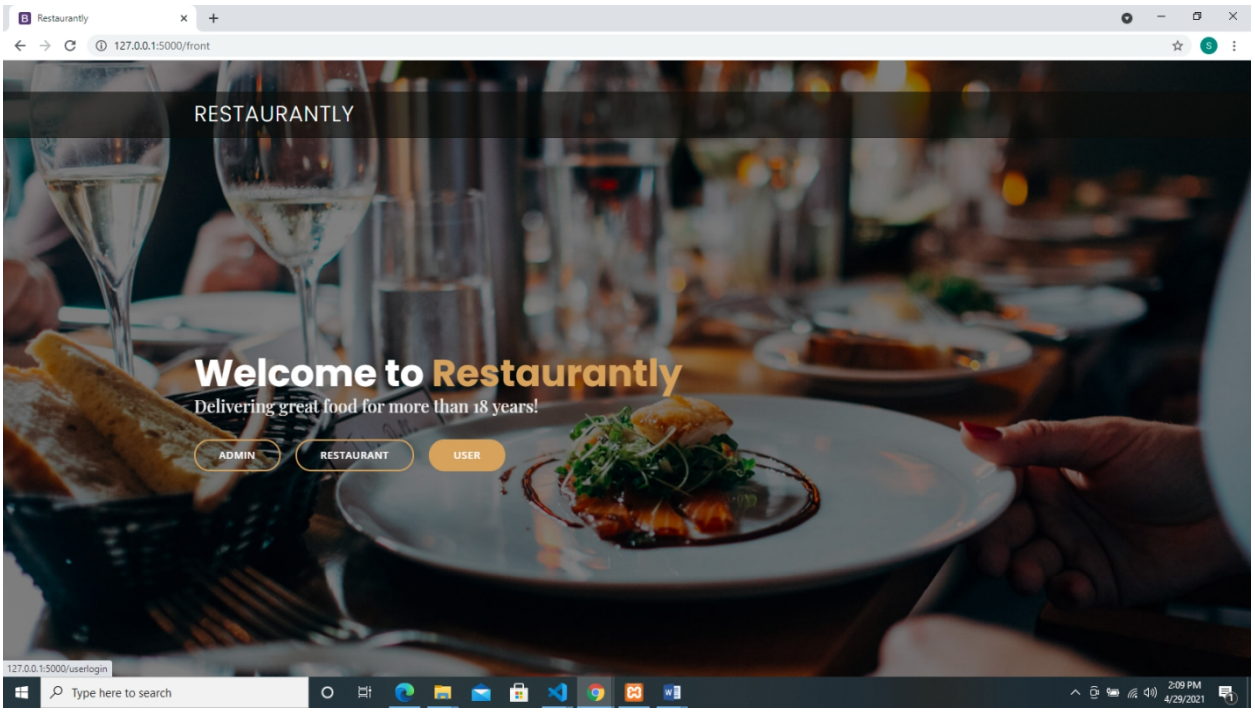
Attribute	Type
Id	Int(10) Primary Key
Restid	Int(10) Foreign Key
Tableno	Varchar (100)
Chair	Varchar (10)
Status	Varchar (100)
Bookid	Varchar (10)

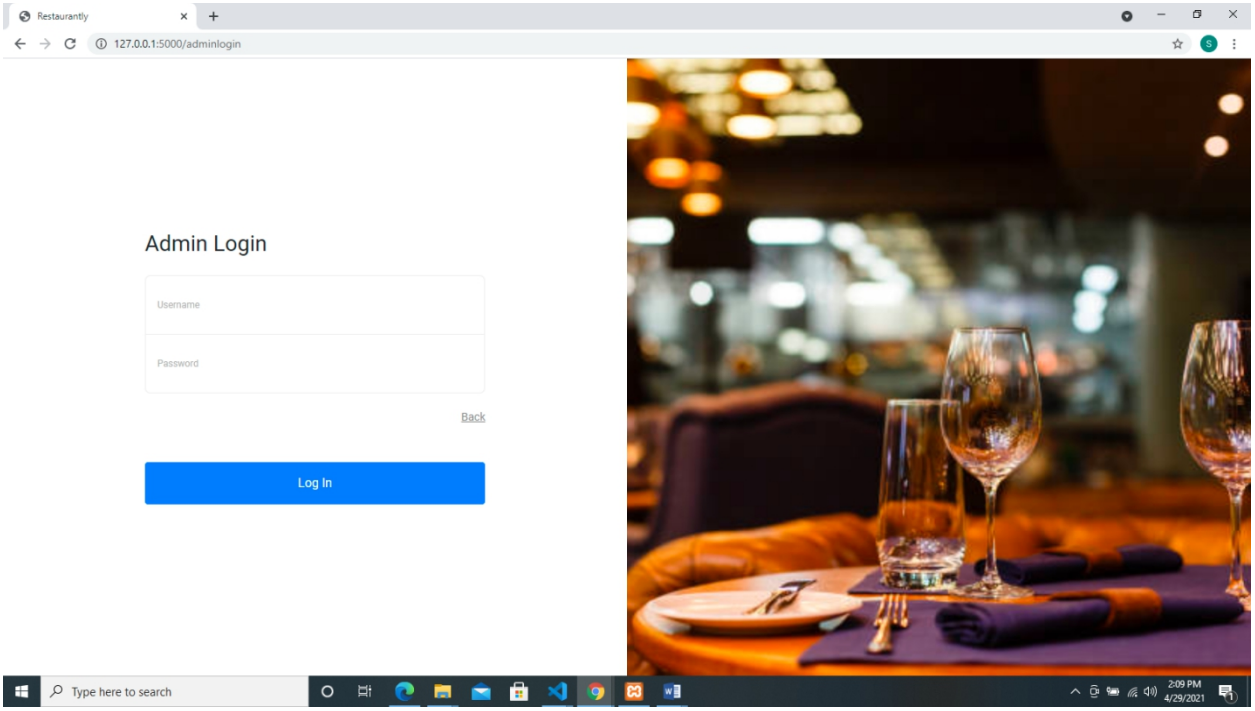
Tb: Userreg

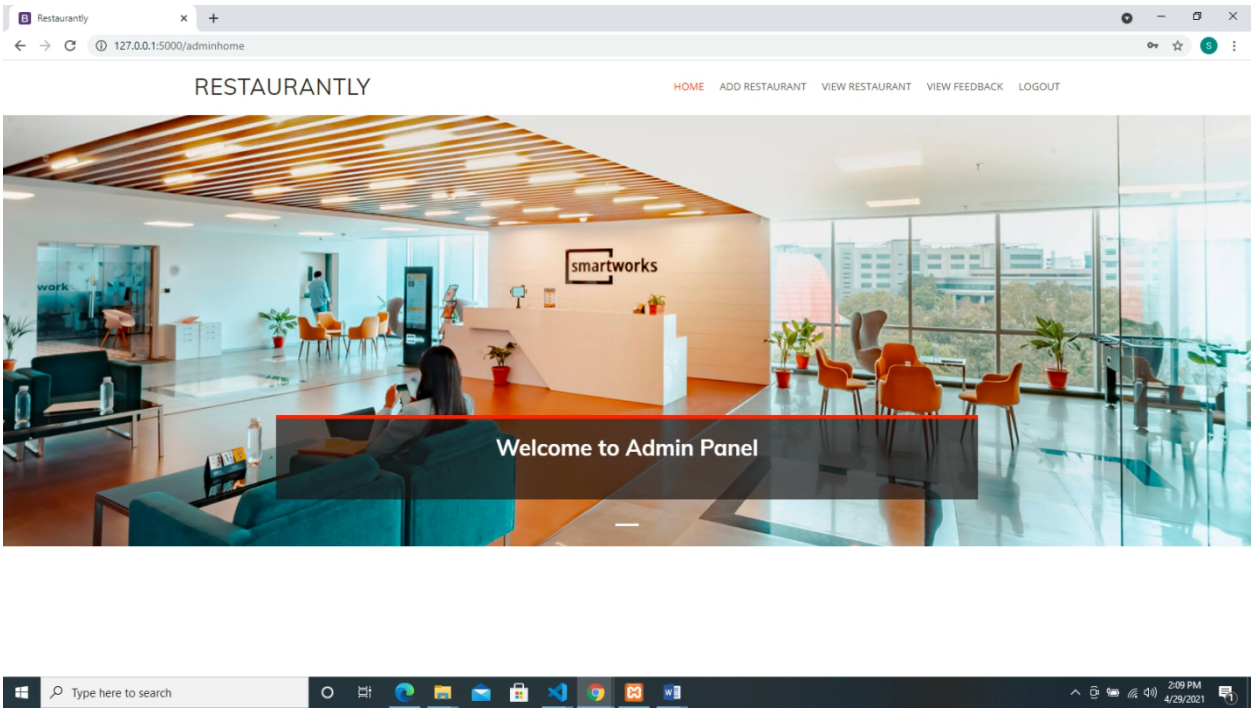
Attribute	Type
Id	Int(10) Primary Key
Name	Varchar (100)
Email	Varchar (100)
Phone No	Varchar (10)
Address	Varchar (100)
Username	Varchar (100)
Password	Varchar (100)

SCREENSHOTS

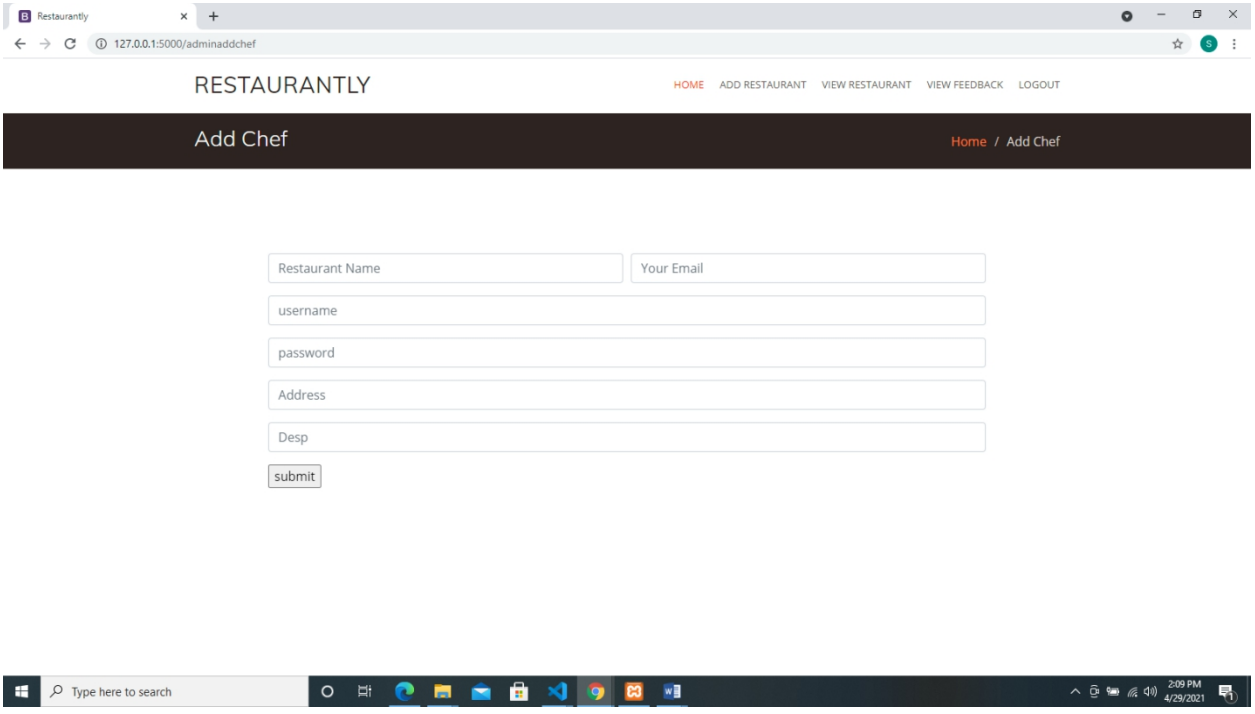
SCREENSHOT


Restaurantly
127.0.0.1:5000/front
RESTAURANTLY
Welcome to Restaurantly
Delivering great food for more than 18 years!
ADMIN
RESTAURANT
USER
127.0.0.1:5000/userlogin
Type here to search
2:09 PM
4/29/2021

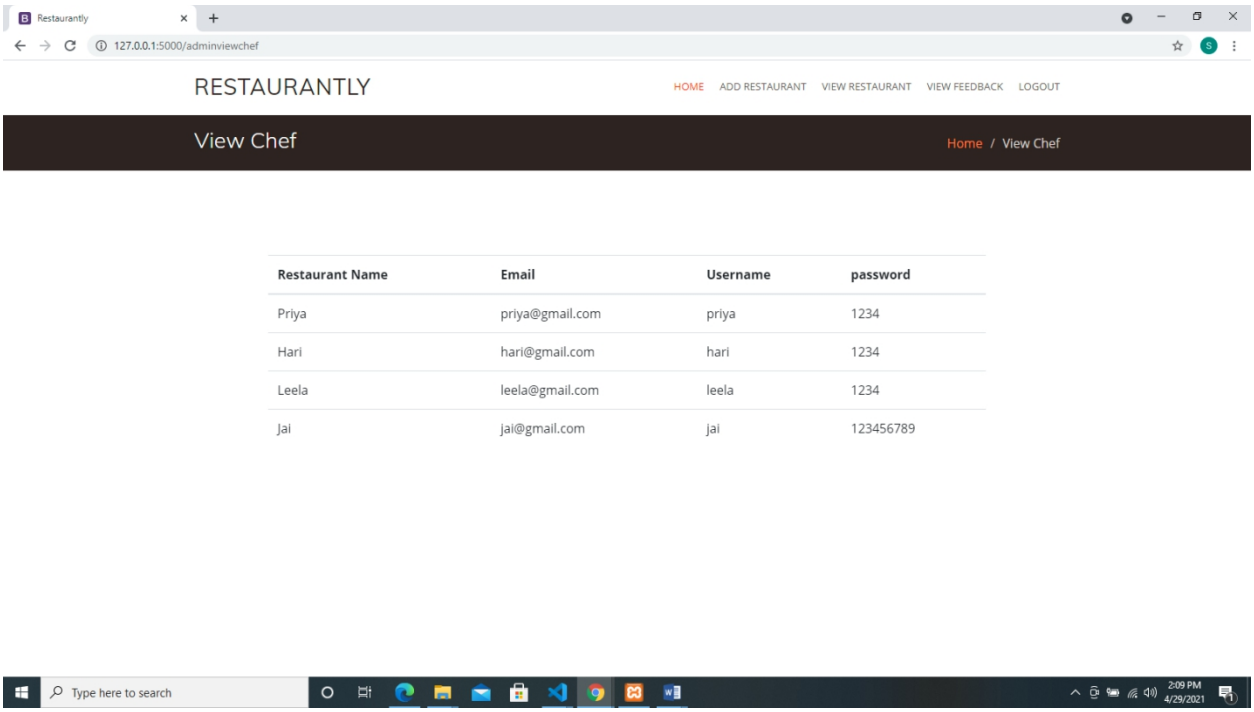

Restaurantly
127.0.0.1:5000/adminlogin
Admin Login
Username
Password
Back
Log In
Type here to search
2:09 PM
4/29/2021

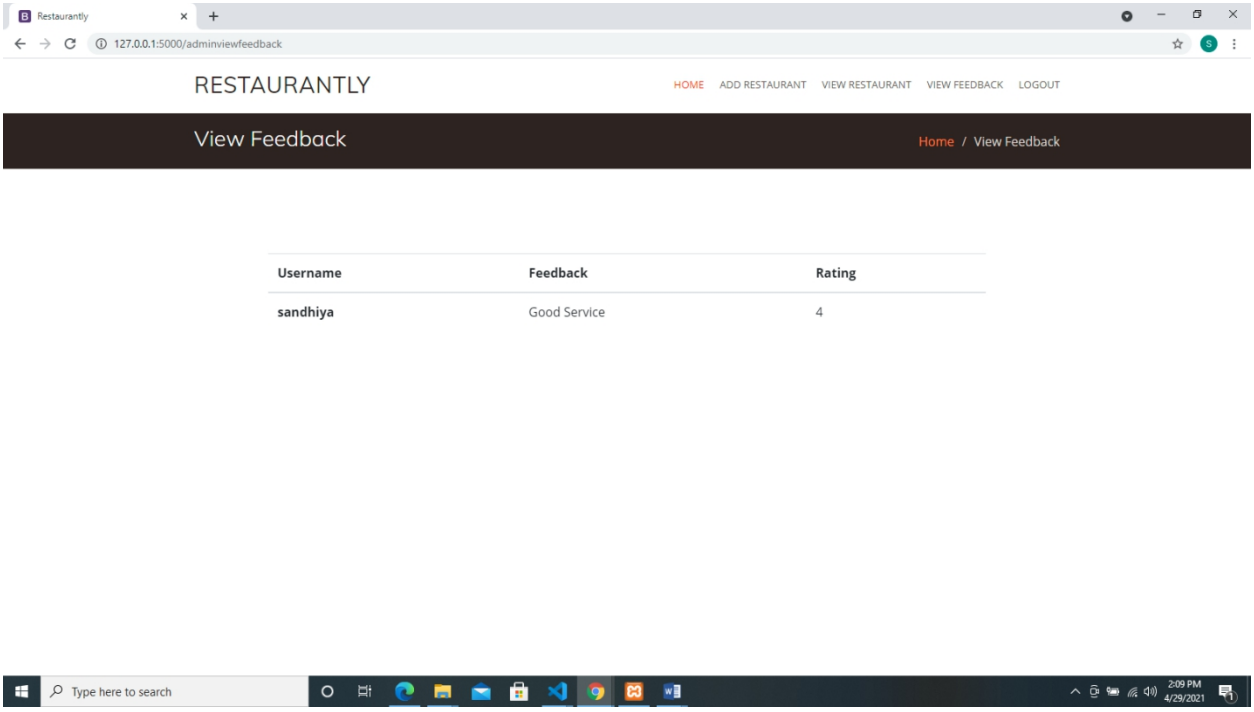

Restaurantly
127.0.0.1:5000/adminhome
RESTAURANTLY
HOME
ADD RESTAURANT
VIEW RESTAURANT
VIEW FEEDBACK
LOGOUT
smartworks
Welcome to Admin Panel

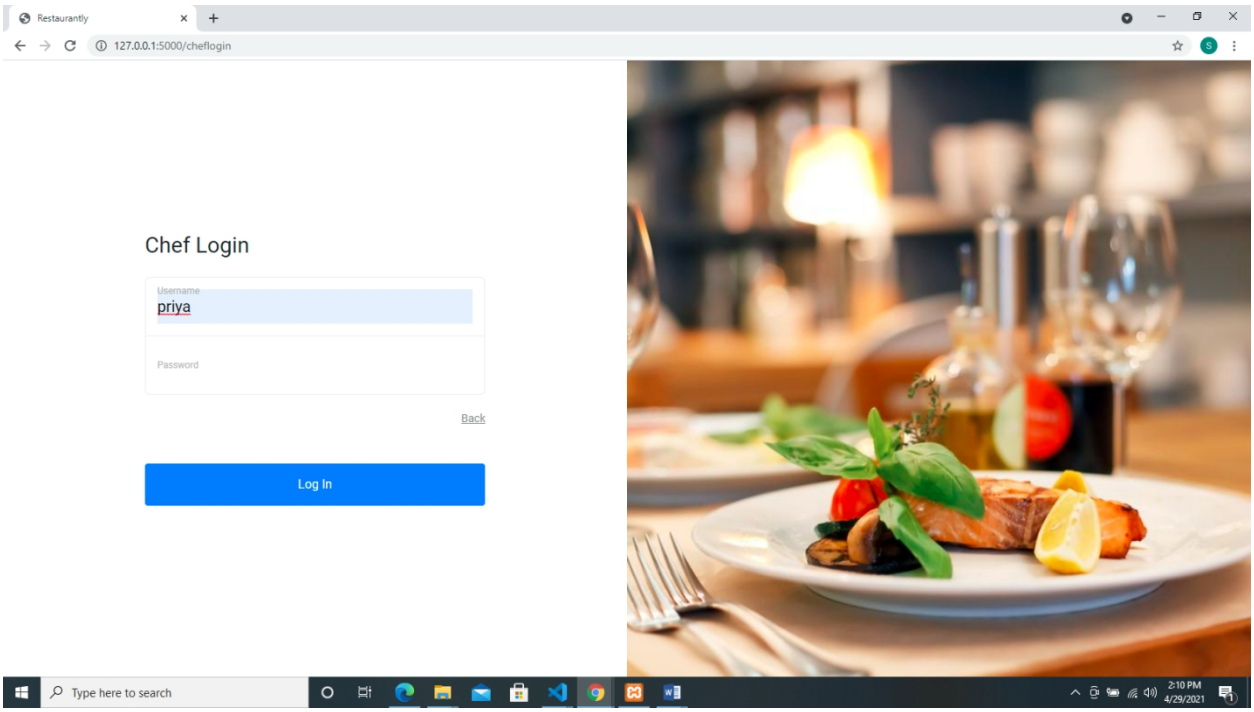
Type here to search
2:09 PM
4/29/2021

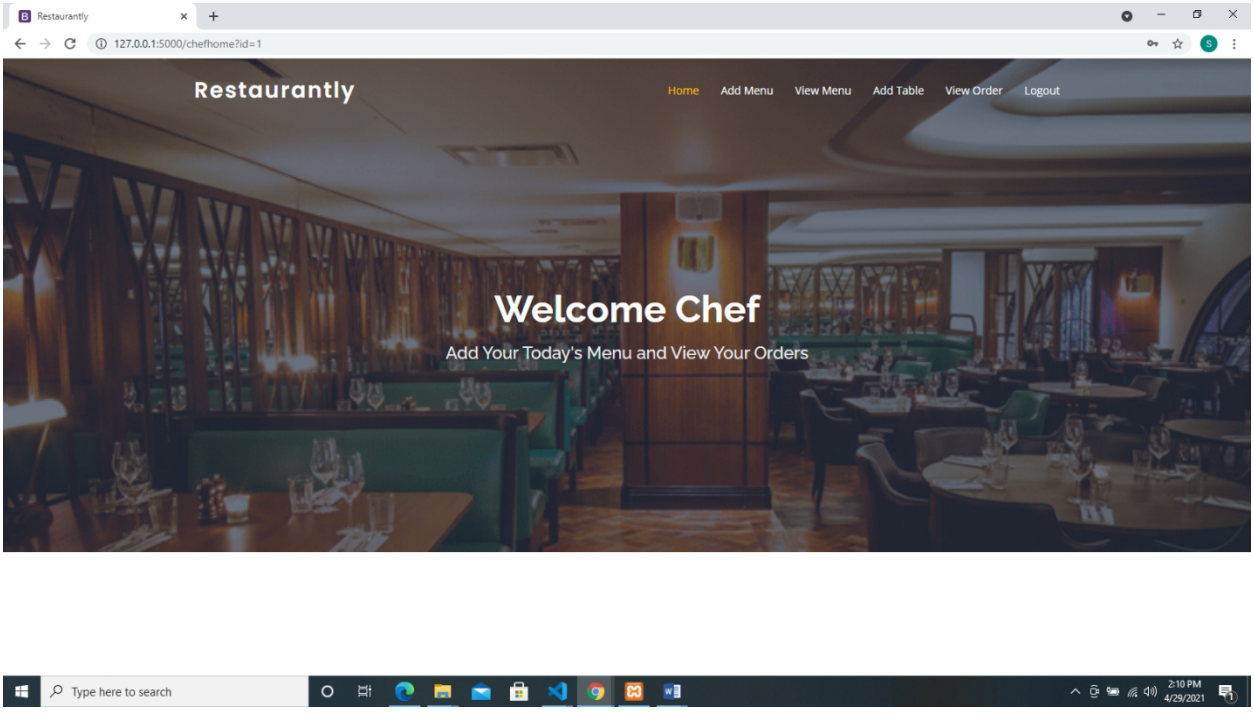

Restaurantly
127.0.0.1:5000/adminaddchef
RESTAURANTLY
HOME
ADD RESTAURANT
VIEW RESTAURANT
VIEW FEEDBACK
LOGOUT
Add Chef
Home / Add Chef
Restaurant Name
Your Email
username
password
Address
Desp
submit

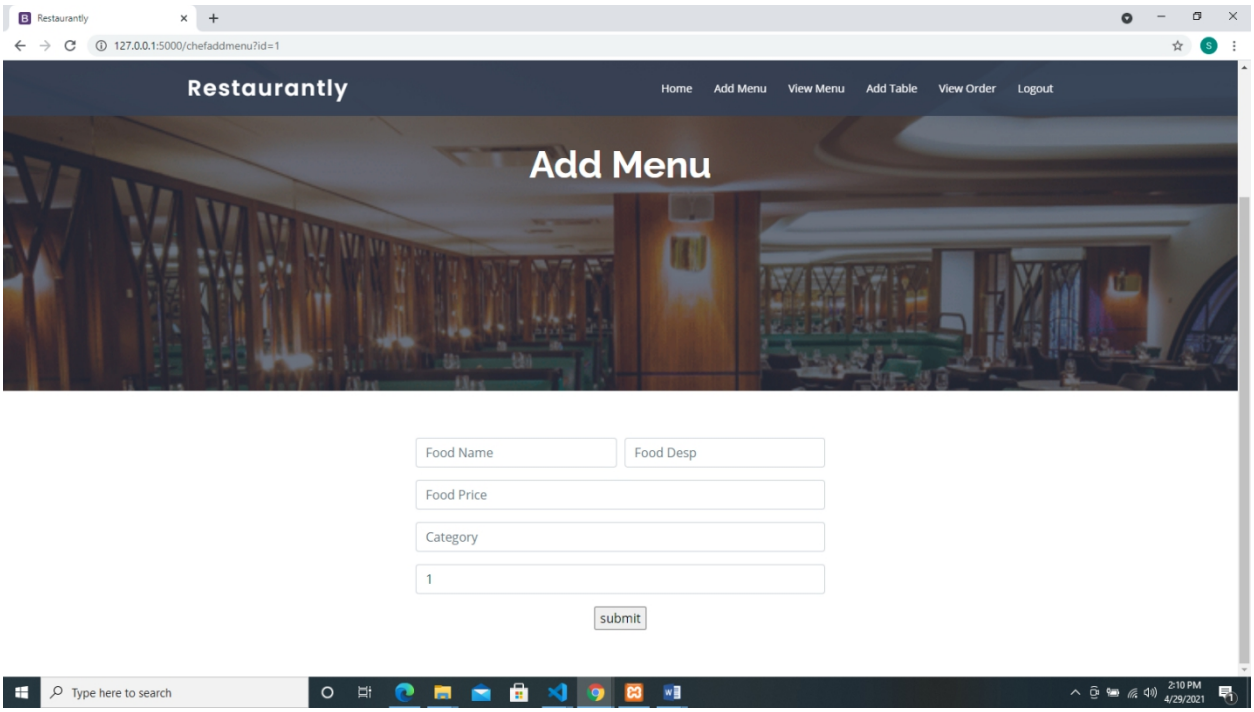
Type here to search
2:09 PM
4/29/2021

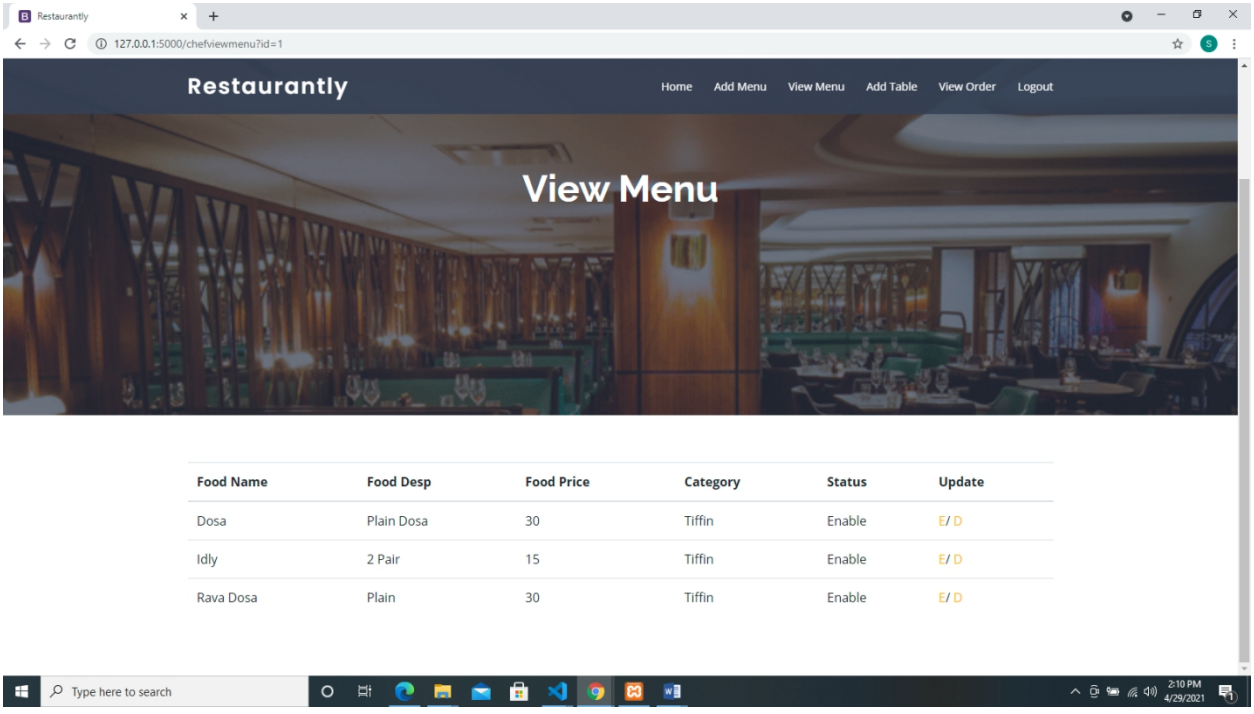

Restaurantly
127.0.0.1:5000/adminviewchef
RESTAURANTLY
HOME ADD RESTAURANT VIEW RESTAURANT VIEW FEEDBACK LOGOUT
View Chef
Home / View Chef
Restaurant Name Email Username password
Priya priya@gmail.com priya 1234
Hari hari@gmail.com hari 1234
Leela leela@gmail.com leela 1234
Jai jai@gmail.com jai 123456789
Type here to search
2:09 PM
4/29/2021

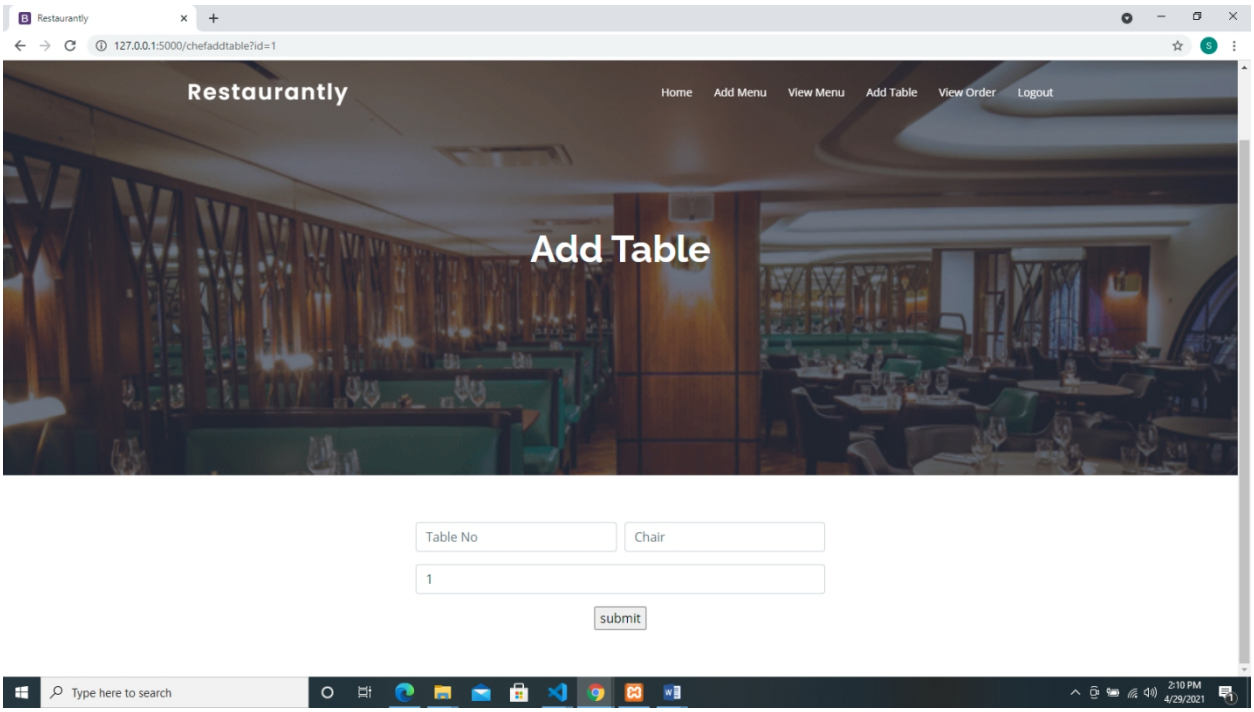

Restaurantly
127.0.0.1:5000/adminviewfeedback
RESTAURANTLY
HOME ADD RESTAURANT VIEW RESTAURANT VIEW FEEDBACK LOGOUT
View Feedback
Home / View Feedback
Username Feedback Rating
sandhiya Good Service 4
Type here to search
2:09 PM
4/29/2021

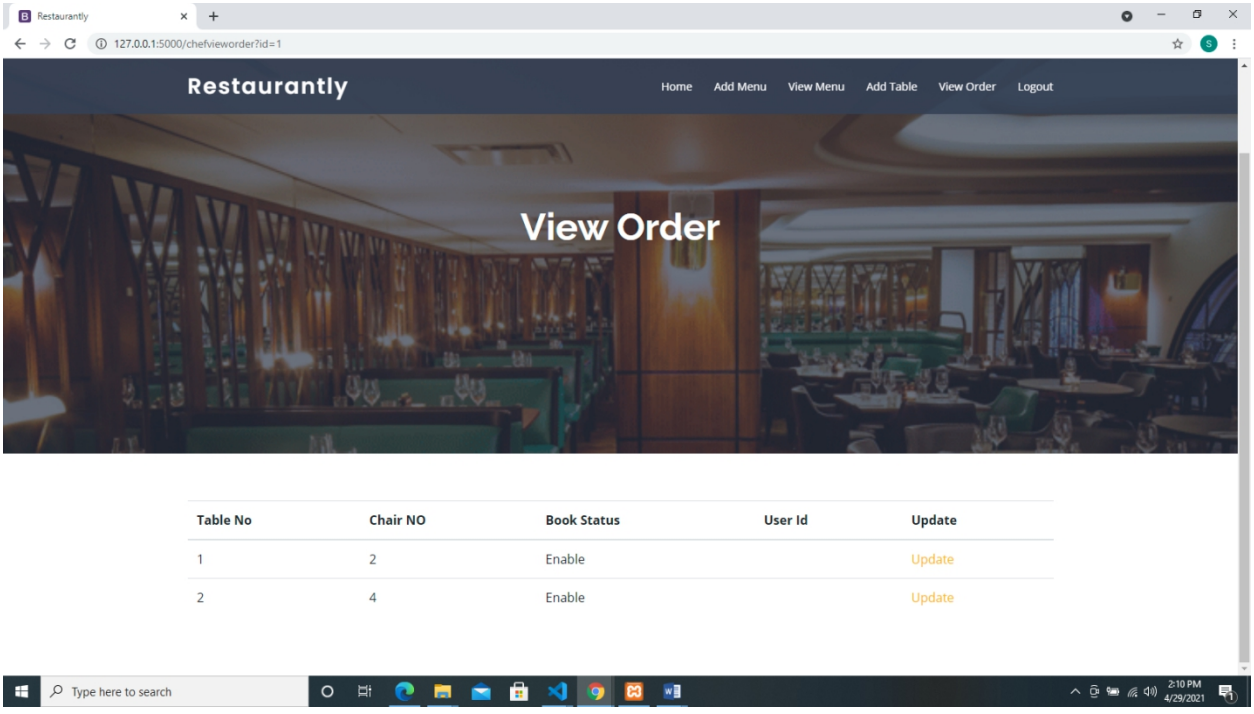

Restaurantly
127.0.0.1:5000/cheflogin
Chef Login
Username
priya
Password
Back
Log In
Type here to search
2:10 PM
4/29/2021

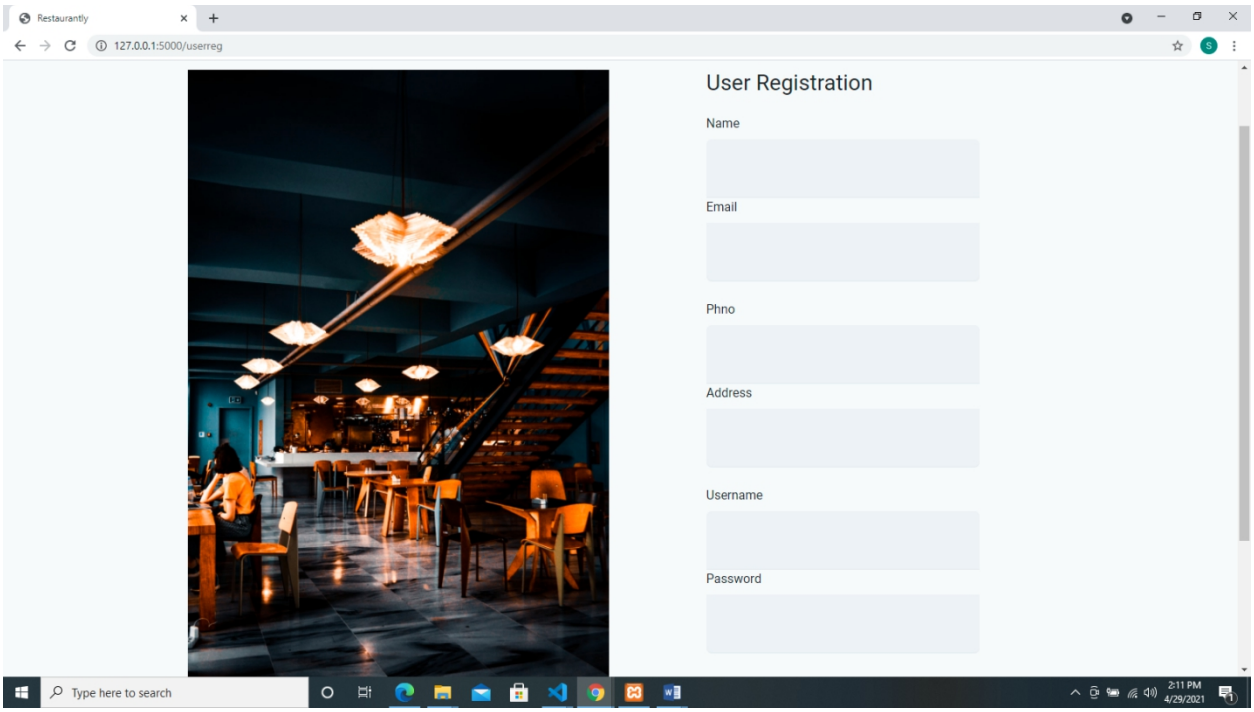

Restaurantly
127.0.0.1:5000/chefhome?id=1
Restaurantly
Home
Add Menu
View Menu
Add Table
View Order
Logout
Welcome Chef
Add Your Today's Menu and View Your Orders
Type here to search
2:10 PM
4/29/2021

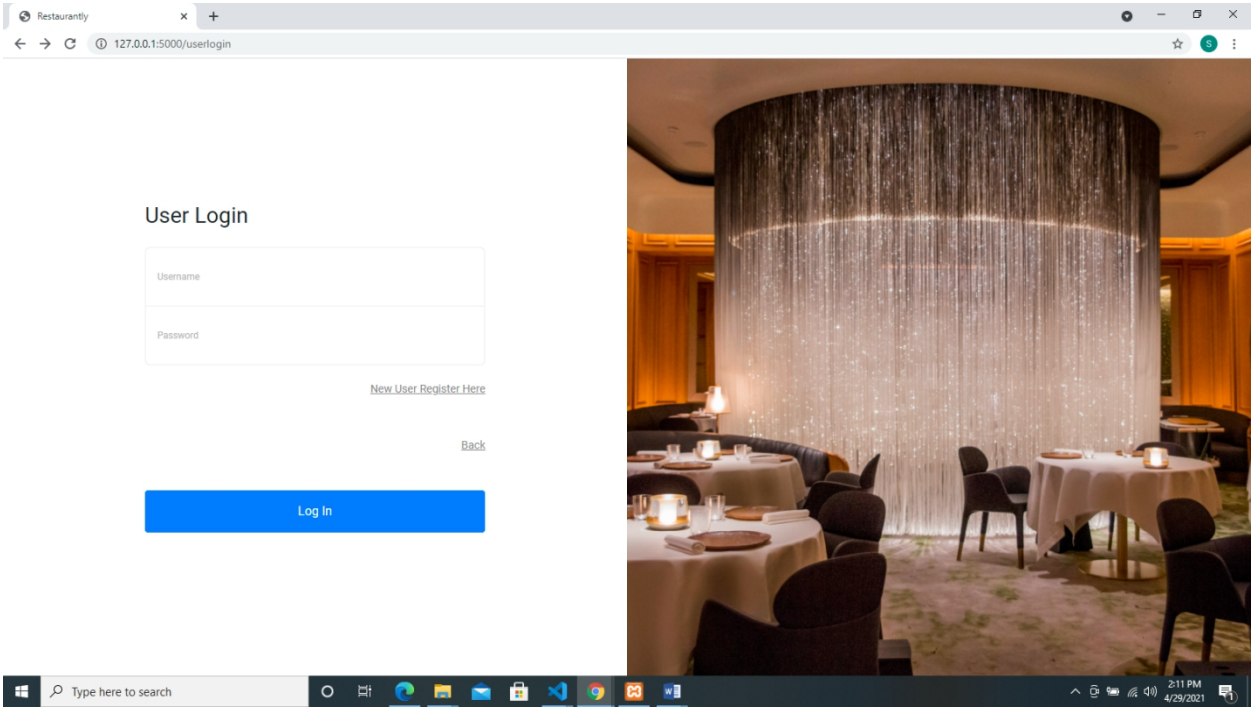

Restaurantly
127.0.0.1:5000/chefaddmenu?id=1
Restaurantly
Home
Add Menu
View Menu
Add Table
View Order
Logout
Add Menu
Food Name
Food Desp
Food Price
Category
1
submit
Type here to search
2:10 PM
4/29/2021

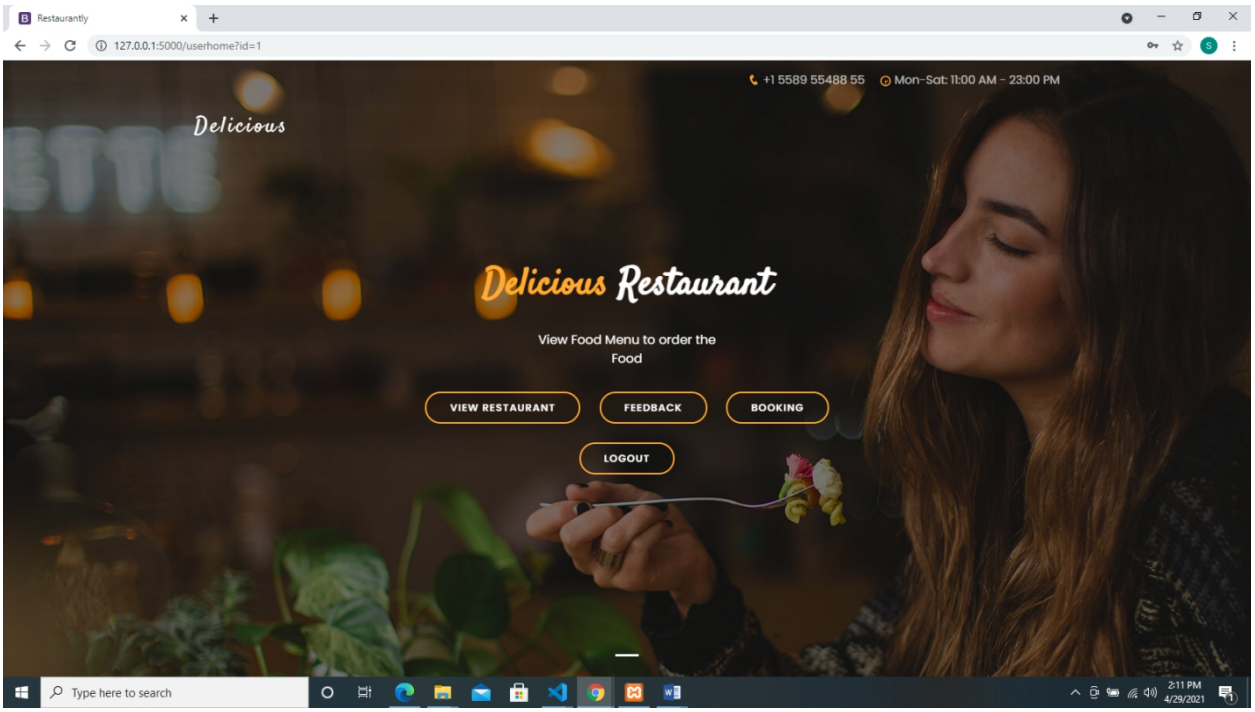

Restaurantly
127.0.0.1:5000/chefviewmenu?id=1
Restaurantly
Home
Add Menu
View Menu
Add Table
View Order
Logout
View Menu
Food Name
Food Desp
Food Price
Category
Status
Update
Dosa
Plain Dosa
30
Tiffin
Enable
E/ D
Idly
2 Pair
15
Tiffin
Enable
E/ D
Rava Dosa
Plain
30
Tiffin
Enable
E/ D
Type here to search
2:10 PM
4/29/2021

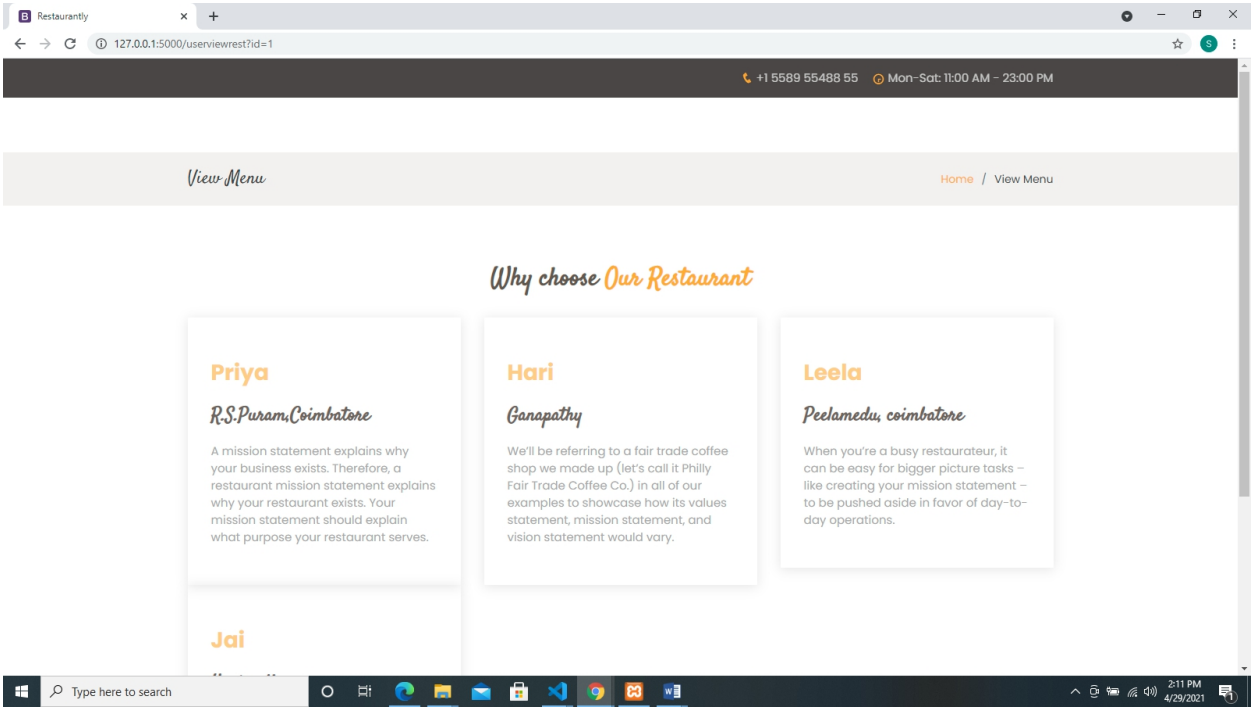

Restaurantly
127.0.0.1:5000/chefaddtable?id=1
Restaurantly
Home
Add Menu
View Menu
Add Table
View Order
Logout
Add Table
Table No
Chair
1
submit
Type here to search
2:10 PM
4/29/2021

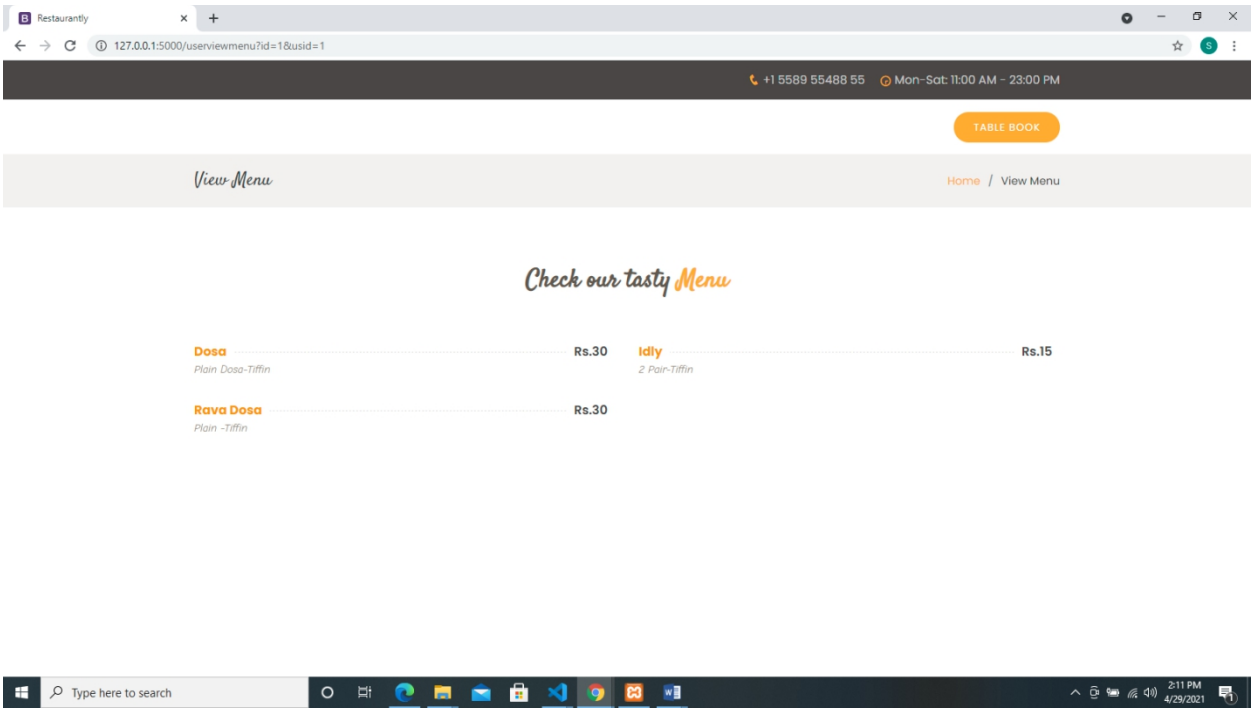

Restaurantly
127.0.0.1:5000/chefvieworder?id=1
Restaurantly
Home
Add Menu
View Menu
Add Table
View Order
Logout
View Order
Table No
Chair NO
Book Status
User Id
Update
1
2
Enable
Update
2
4
Enable
Update
Type here to search
2:10 PM
4/29/2021

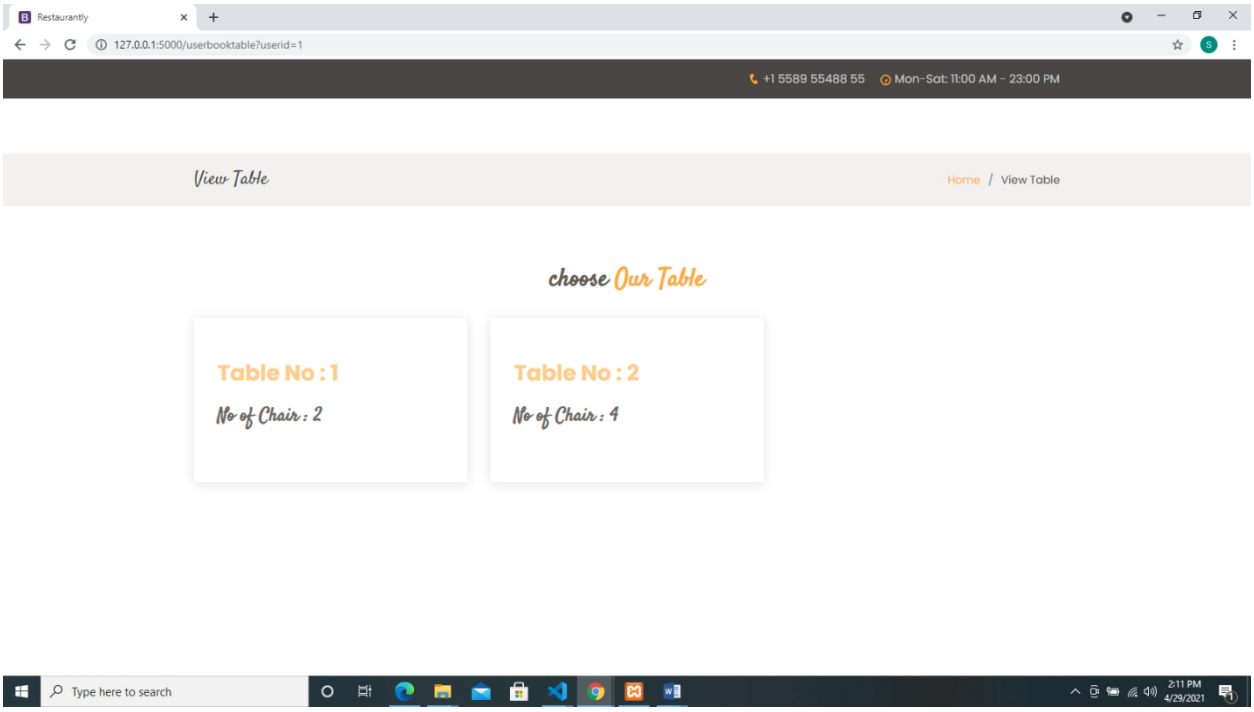

Restaurantly
127.0.0.1:5000/userreg
User Registration
Name
Email
Phno
Address
Username
Password
Type here to search
2:11 PM
4/29/2021

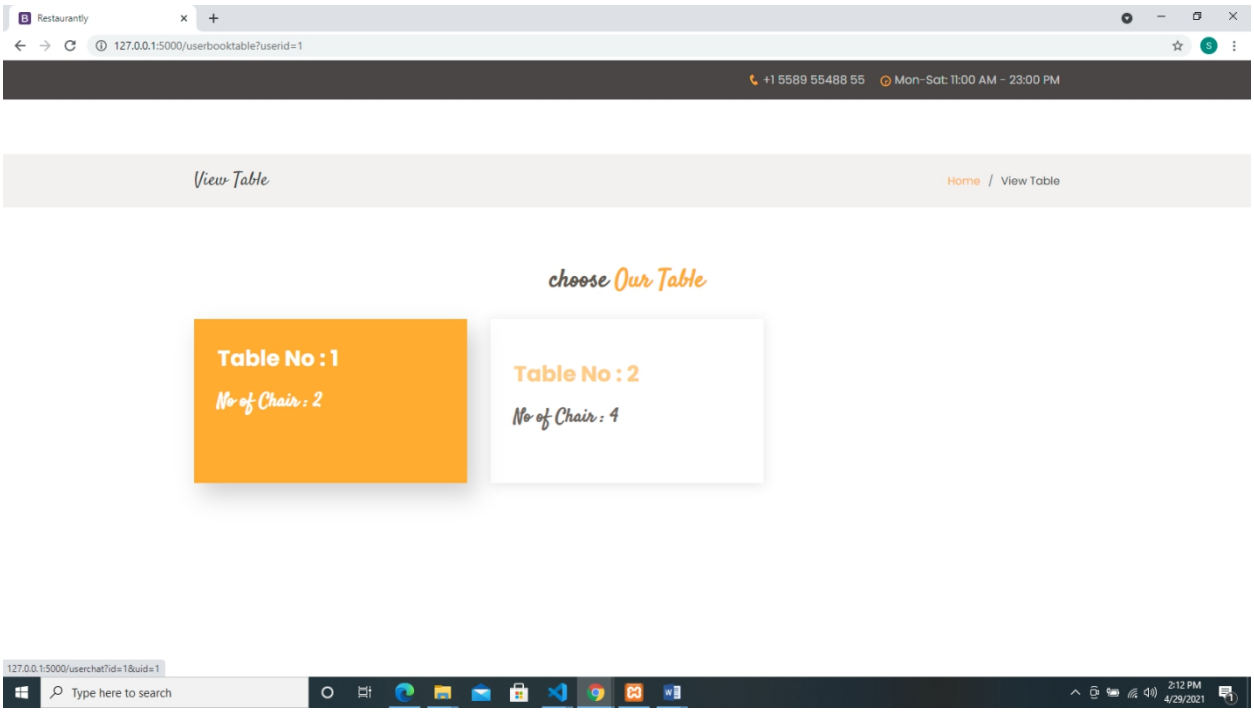

Restaurantly
127.0.0.1:5000/userlogin
User Login
Username
Password
New User Register Here
Back
Log In
Type here to search
2:11 PM
4/29/2021

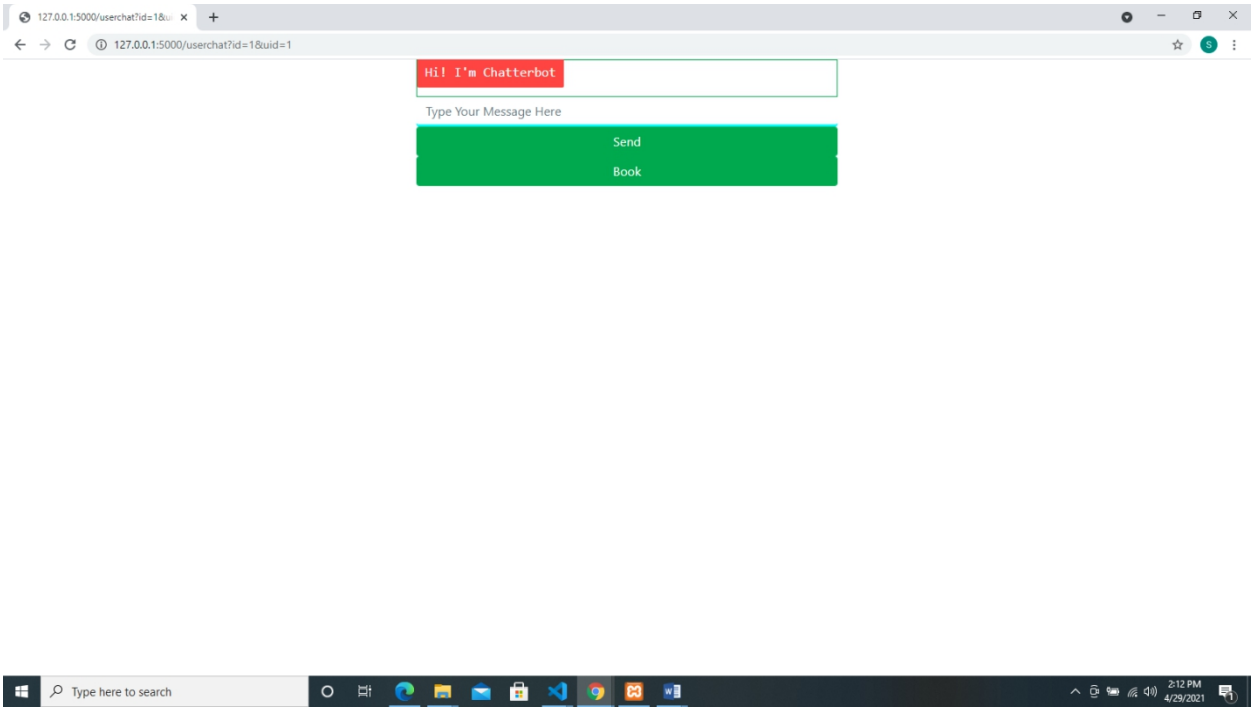

Restaurantly
127.0.0.1:5000/userhome?id=1
+1 5589 55488 55
Mon-Sat: 11:00 AM - 23:00 PM
Delicious
Delicious Restaurant
View Food Menu to order the Food
VIEW RESTAURANT
FEEDBACK
BOOKING
LOGOUT
Type here to search
2:11 PM
4/29/2021

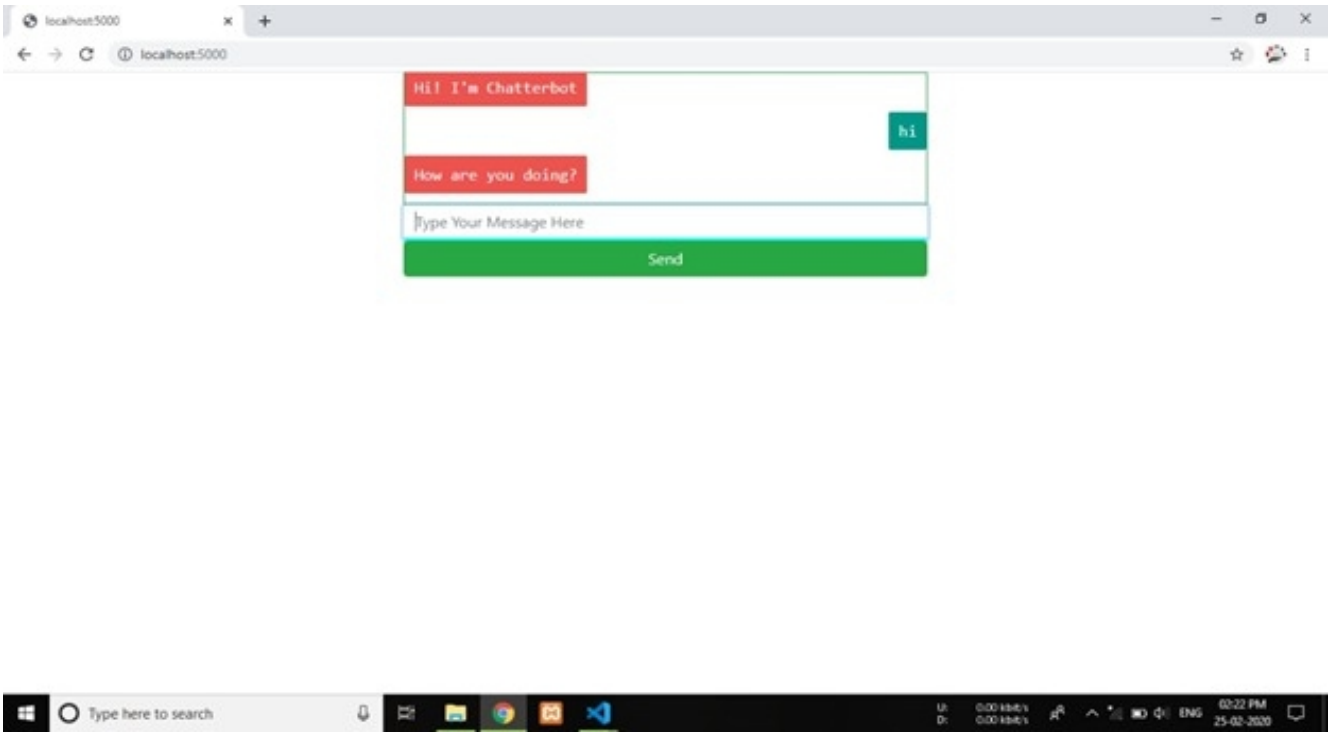

Restaurantly
127.0.0.1:5000/userviewrest?id=1
+1 5589 55488 55
Mon-Sat: 11:00 AM - 23:00 PM
View Menu
Home / View Menu
Why choose Our Restaurant
Priya
R.S.Puram,Coimbatore
A mission statement explains why your business exists. Therefore, a restaurant mission statement explains why your restaurant exists. Your mission statement should explain what purpose your restaurant serves.
Hari
Ganapathy
We'll be referring to a fair trade coffee shop we made up (let's call it Philly Fair Trade Coffee Co.) in all of our examples to showcase how its values statement, mission statement, and vision statement would vary.
Leela
Peelamedu, coimbatore
When you're a busy restaurateur, it can be easy for bigger picture tasks – like creating your mission statement – to be pushed aside in favor of day-to-day operations.
Jai
Type here to search
2:11 PM
4/29/2021

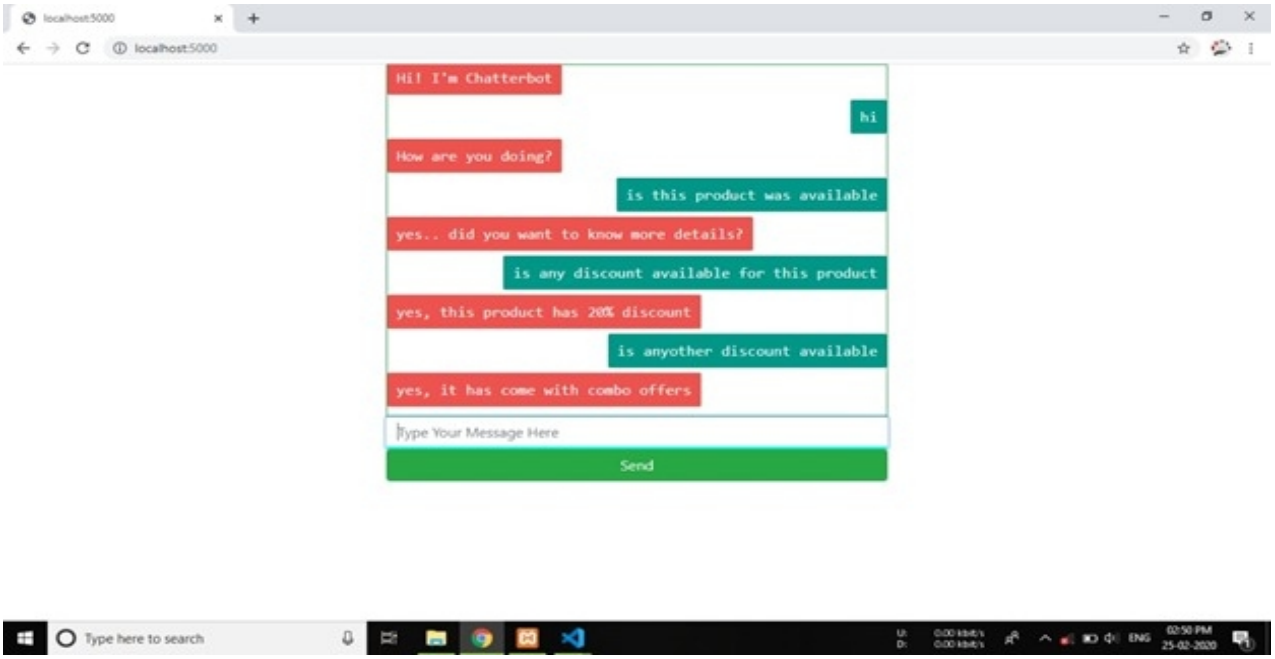

Restaurantly
127.0.0.1:5000/userviewmenu?id=1&usid=1
+1 5589 55488 55
Mon-Sat: 11:00 AM - 23:00 PM
TABLE BOOK
View Menu
Home / View Menu
Check our tasty Menu
Dosa
Plain Dosa-Tiffin
Rs.30
Idly
2 Pair-Tiffin
Rs.15
Rava Dosa
Plain -Tiffin
Rs.30
Type here to search
2:11 PM
4/29/2021

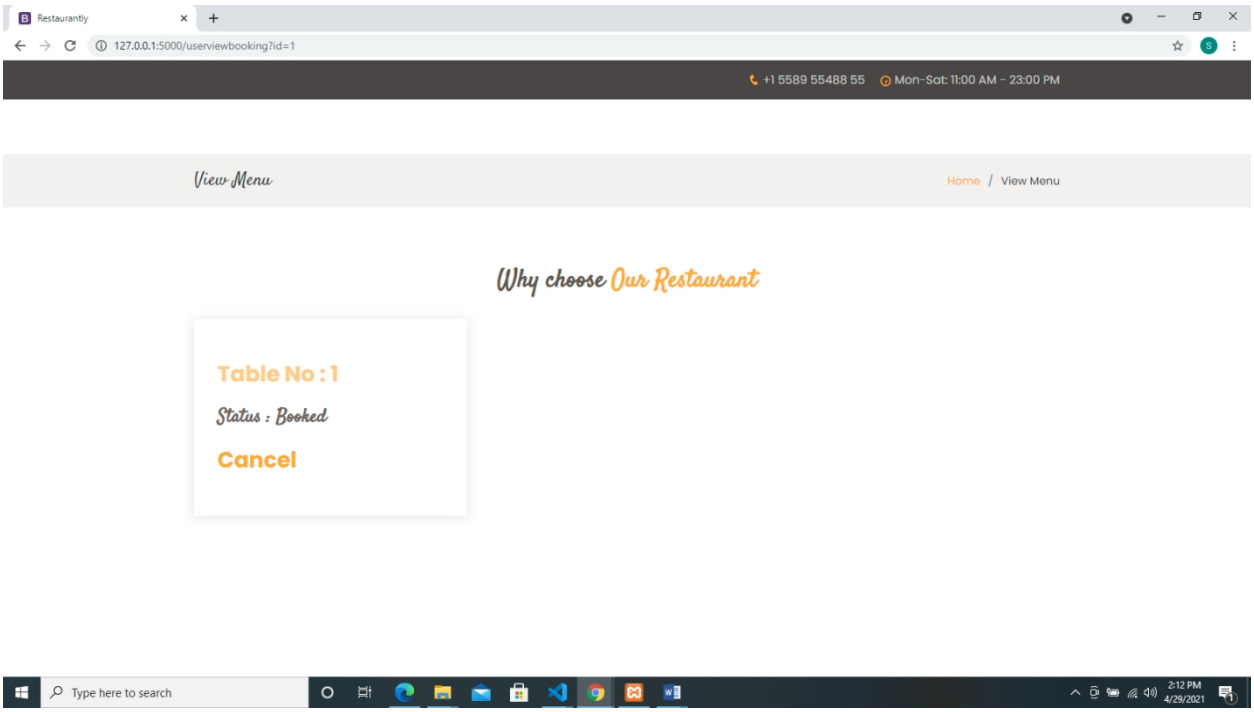

Restaurantly
127.0.0.1:5000/userbooktable?userid=1
+1 5589 55488 55
Mon-Sat: 11:00 AM - 23:00 PM
View Table
Home / View Table
choose Our Table
Table No : 1
No of Chair : 2
Table No : 2
No of Chair : 4
Type here to search
2:11 PM
4/29/2021

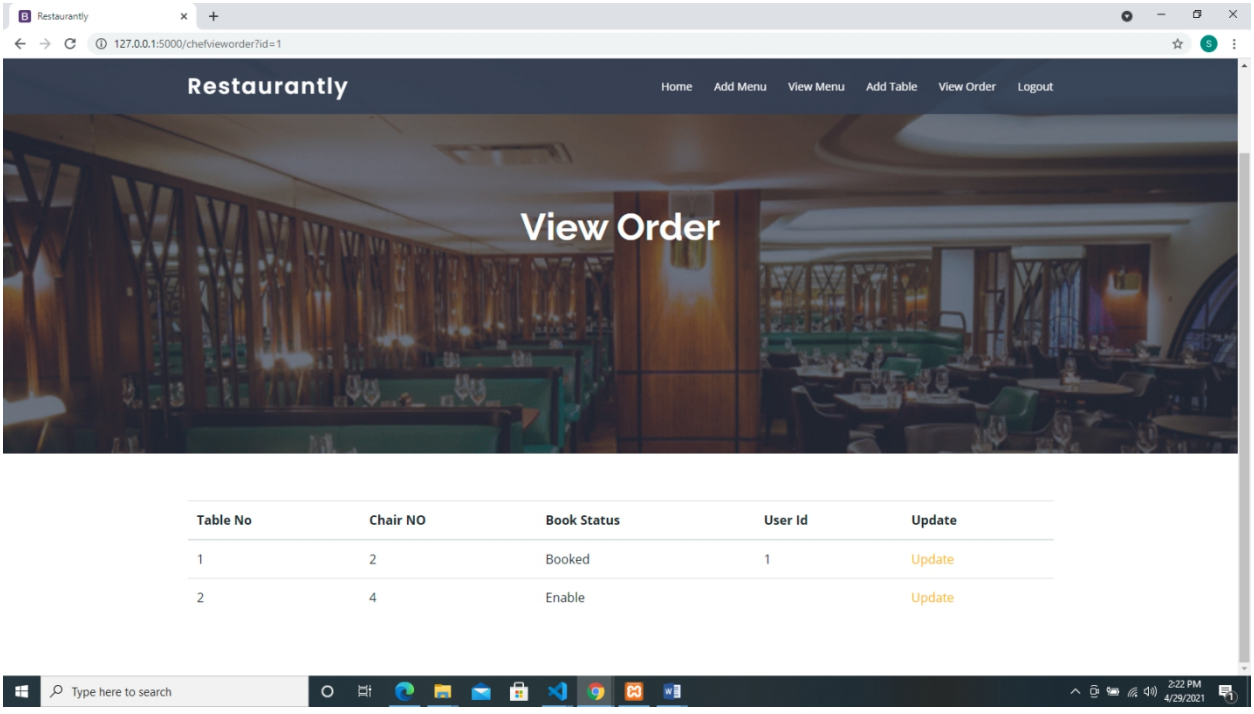

Restaurantly
127.0.0.1:5000/userbooktable?userid=1
+1 5589 55488 55
Mon-Sat: 11:00 AM - 23:00 PM
View Table
Home / View Table
choose Our Table
Table No : 1
No of Chair : 2
Table No : 2
No of Chair : 4
Type here to search
2:12 PM
4/29/2021

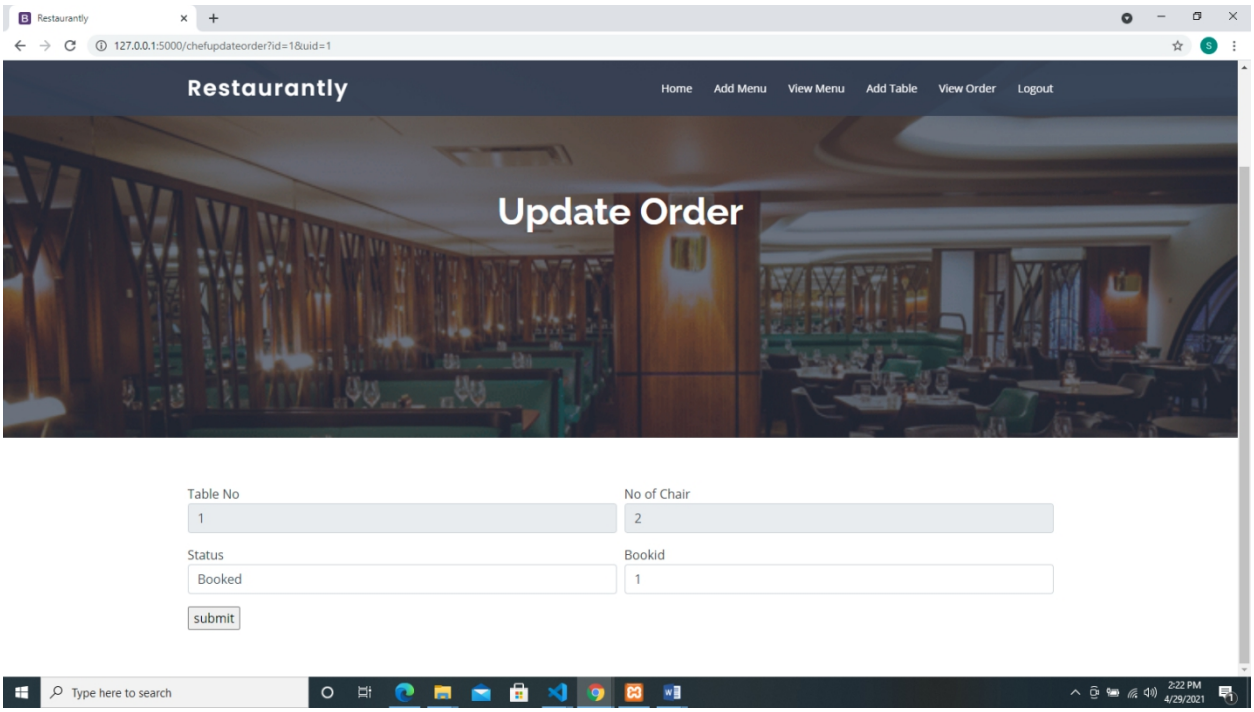

127.0.0.1:5000/userchat?id=1&uid=1
Hi! I'm Chatterbot
Type Your Message Here
Send
Book
Type here to search
2:12 PM
4/29/2021


localhost:5000
Hi! I'm Chatterbot
hi
How are you doing?
Type Your Message Here
Send
Type here to search
ENG
02:22 PM
25-02-2020


localhost:5000
Hi! I'm Chatterbot
hi
How are you doing?
is this product was available
yes.. did you want to know more details?
is any discount available for this product
yes, this product has 20% discount
is anyother discount available
yes, it has come with combo offers
Type Your Message Here
Send
Type here to search
ENG
02:50 PM
25-02-2020


Restaurantly
127.0.0.1:5000/userviewbooking?id=1
+1 5589 55488 55
Mon-Sat: 11:00 AM - 23:00 PM
View Menu
Home / View Menu
Why choose Our Restaurant
Table No : 1
Status : Booked
Cancel
Type here to search
2:12 PM
4/29/2021


Restaurantly
127.0.0.1:5000/chefvieworder?id=1
Restaurantly
Home
Add Menu
View Menu
Add Table
View Order
Logout
View Order
Table No
Chair NO
Book Status
User Id
Update
1
2
Booked
1
Update
2
4
Enable
Update
Type here to search
2:22 PM
4/29/2021


Restaurantly
127.0.0.1:5000/chefupdateorder?id=1&uid=1
Restaurantly
Home
Add Menu
View Menu
Add Table
View Order
Logout
Update Order
Table No
1
No of Chair
2
Status
Booked
Bookid
1
submit
Type here to search
2:22 PM
4/29/2021

CODING

CODING

```
from flask import Flask,render_template,redirect,request,url_for,jsonify,json
from flaskext.mysql import MySQL
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer

app=Flask(__name__)
mysql=MySQL()

english_bot = ChatBot("Chatterbot", storage_adapter="chatterbot.storage.SQLStorageAdapter")
trainer = ChatterBotCorpusTrainer(english_bot)
trainer.train("chatterbot.corpus.english")

app.config['MYSQL_DATABASE_USER']='root'
app.config['MYSQL_DATABASE_PASSWORD']=''
app.config['MYSQL_DATABASE_DB']='restaurant'
app.config['MYSQL_DATABASE_HOST']='localhost'
mysql.init_app(app)

@app.route("/front")
def front():
    return render_template("front.html")

@app.route("/adminlogin",methods=['GET','POST'])
def adminlogin():
    if request.method=='POST':
        username=request.form['username']
```

```
        password=request.form['password']
        if username=="admin" and password=="admin":
            return redirect('adminhome')
    else:
        return render_template("adminlogin.html")


@app.route("/adminhome")
def adminhome():
    return render_template("adminhome.html")


@app.route("/adminaddchef",methods=['GET','POST'])
def adminaddchef():
    if request.method=='POST':
        chefname=request.form['chefname']
        email=request.form['email']
        username=request.form['username']
        password=request.form['password']
        address=request.form['address']
        desp=request.form['desp']
        conn=mysql.connect()
        cur=conn.cursor()
        cur.execute("INSERT INTO `chef`( `chefname`, `email`, `username`, `password`,`address`,`desp`) VALUES(%s,%s,
%s,%s,%s,%s)",(chefname,email,username,password,address,desp))
        conn.commit()
        return redirect(url_for('adminviewchef'))
    else:
        return render_template("adminaddchef.html")


@app.route("/adminviewchef",methods=['GET','POST'])
```

```
def adminviewchef():
    conn=mysql.connect()
    cur=conn.cursor()
    cur.execute("SELECT * FROM `chef` ")
    data=cur.fetchall()
    return render_template("adminviewchef.html",chef=data)


@app.route("/adminviewfeedback",methods=['GET','POST'])
def adminviewfeedback():
    conn=mysql.connect()
    cur=conn.cursor()
    cur.execute("SELECT * FROM `feedback` ")
    data=cur.fetchall()
    return render_template("adminviewfeedback.html",feed=data)


@app.route("/cheflogin",methods=['GET','POST'])
def cheflogin():
    conn=mysql.connect()
    cur=conn.cursor()
    if request.method=='POST':
        username=request.form['username']
        password=request.form['password']
        cur.execute("SELECT * FROM `chef` WHERE `username`='"+username+"' AND `password`='"+password+"' ")
        data=cur.fetchone()
        if data[3]==username and data[4]==password:
            return redirect(url_for('chefhome',id=data[0]))
        else:
            return redirect('cheflogin')
```

```
    else:
        return render_template("cheflogin.html")


@app.route("/chefhome")
def chefhome():
    uid=request.args.get('id')
    return render_template("chefhome.html",userid=uid)


@app.route("/chefaddmenu",methods=['GET','POST'])
def chefaddmenu():
    uid=request.args.get('id')
    if request.method=='POST':
        userid=request.form['userid']
        foodname=request.form['foodname']
        fooddesp=request.form['fooddesp']
        price=request.form['price']
        category=request.form['category']
        status="Enable"
        conn=mysql.connect()
        cur=conn.cursor()
        cur.execute("INSERT INTO `menu`(`userid`,`foodname`, `fooddesp`, `price`, `category`, `status`) VALUES(%s,%s,
%s,%s,%s,%s)",(userid,foodname,fooddesp,price,category,status))
        conn.commit()
        return redirect(url_for('chefviewmenu',id=userid))
    else:
        return render_template("chefaddmenu.html",userid=uid)


@app.route("/chefaddtable",methods=['GET','POST'])
def chefaddtable():
```

```
    uid=request.args.get('id')
    if request.method=='POST':
        userid=request.form['userid']
        tableno=request.form['tableno']
        chair=request.form['chair']
        status="Enable"
        conn=mysql.connect()
        cur=conn.cursor()
        cur.execute("INSERT INTO `tableno`(`userid`,`tableno`, `Chair`, `status`) VALUES(%s,%s,%s,%s)",(userid,tableno,
chair,status))
        conn.commit()
        return redirect(url_for('chefvieworder',id=userid))
    else:
        return render_template("chefaddtable.html",userid=uid)


@app.route("/chefviewmenu",methods=['GET','POST'])
def chefviewmenu():
    uid=request.args.get('id')
    conn=mysql.connect()
    cur=conn.cursor()
    cur.execute("SELECT * FROM `menu` where `userid`='"+uid+"'")
    data=cur.fetchall()
    return render_template("chefviewmenu.html",menu=data,userid=uid)


@app.route("/chefmenuenable",methods=['GET','POST'])
def chefmenuenable():
    status="Enable"
    conn=mysql.connect()
    cur=conn.cursor()
```

```
    menuid=request.args.get('id')
    userid=request.args.get('uid')
    cur.execute("UPDATE `menu` SET `status`='"+status+"' WHERE `id`='"+menuid+"'")
    conn.commit()
    return redirect(url_for('chefviewmenu',id=userid))

@app.route("/chefmenudisable",methods=['GET','POST'])
def chefmenudisable():
    status="Disable"
    conn=mysql.connect()
    cur=conn.cursor()
    menuid=request.args.get('id')
    userid=request.args.get('uid')
    cur.execute("UPDATE `menu` SET `status`='"+status+"' WHERE `id`='"+menuid+"'")
    conn.commit()
    return redirect(url_for('chefviewmenu',id=userid))

@app.route("/chefvieworder",methods=['GET','POST'])
def chefvieworder():
    uid=request.args.get('id')
    conn=mysql.connect()
    cur=conn.cursor()
    cur.execute("SELECT * FROM `tableno` where `userid`='"+uid+"'")
    data=cur.fetchall()
    return render_template("chefvieworder.html",userid=uid,table=data)

@app.route("/chefupdateorder",methods=['GET','POST'])
def chefupdateorder():
```

```
    uid=request.args.get('uid')
    oid=request.args.get('id')
    conn=mysql.connect()
    cur=conn.cursor()
    cur.execute("SELECT * FROM `tableno` where `id`='"+oid+"'")
    data=cur.fetchone()
    if request.method=='POST':
        status=request.form['status']
        bid=request.form['bid']


        conn=mysql.connect()
        cur=conn.cursor()
        cur.execute("UPDATE `tableno` SET `bookuid`='"+bid+"',`status`='"+status+"' WHERE `id`='"+oid+"'")
        conn.commit()
        return redirect(url_for('chefvieworder',id=uid))
    else:
        return render_template("chefupdateorder.html",userid=uid,table=data)


@app.route("/userlogin",methods=['GET','POST'])
def userlogin():
    conn=mysql.connect()
    cur=conn.cursor()
    if request.method=='POST':
        username=request.form['username']
        password=request.form['password']
        cur.execute("SELECT * FROM `userreg` WHERE `username`='"+username+"' AND `password`='"+password+"' ")
        data=cur.fetchone()
        if data[5]==username and data[6]==password:
```

```
            return redirect(url_for('userhome',id=data[0]))
        else:
            return redirect('cheflogin')
    else:
        return render_template("userlogin.html")


@app.route("/userreg",methods=['GET','POST'])
def userreg():
    if request.method=='POST':
        name=request.form['name']
        email=request.form['email']
        phno=request.form['phno']
        address=request.form['address']
        username=request.form['username']
        password=request.form['password']
        conn=mysql.connect()
        cur=conn.cursor()
        cur.execute("INSERT INTO `userreg`( `name`, `email`, `phno`, `address`, `username`, `password`) VALUES (%s,%
s,%s,%s,%s,%s)",(name,email,phno,address,username,password))
        conn.commit()
        return redirect(url_for('userlogin'))
    else:
        return render_template("userreg.html")


@app.route("/userhome")
def userhome():
    userid=request.args.get('id')
    return render_template("userhome.html",uid=userid)
```

```
@app.route("/userviewrest")
def userviewrest():
    userid=request.args.get('id')
    conn=mysql.connect()
    cur=conn.cursor()
    cur.execute("SELECT * FROM `chef`")
    data=cur.fetchall()
    return render_template("userviewrest.html",rest=data,uid=userid)


@app.route("/userviewbooking")
def userviewbooking():
    userid=request.args.get('id')
    conn=mysql.connect()
    cur=conn.cursor()
    cur.execute("SELECT * FROM `tableno` where `bookuid`='"+userid+"'")
    data=cur.fetchall()
    return render_template("userviewbooking.html",rest=data,uid=userid)


@app.route("/userviewmenu")
def userviewmenu():
    restid=request.args.get('id')


    conn=mysql.connect()
    cur=conn.cursor()
    cur.execute("SELECT * FROM `menu` where `userid`='"+restid+"'")
    data=cur.fetchall()
    return render_template("userviewmenu.html",menu=data,uid=restid)
```

```
@app.route("/userfeedback",methods=['GET','POST'])
def userfeedback():
    userid=request.args.get('id')
    conn=mysql.connect()
    cursor=conn.cursor()
    cursor.execute("SELECT * FROM `userreg` WHERE `id`='"+userid+"'")
    student=cursor.fetchone()
    if request.method=='POST':
        name=request.form['name']
        feedback=request.form['feedback']
        rating=request.form['rating']
        conn=mysql.connect()
        cur=conn.cursor()
        cur.execute("INSERT INTO `feedback`( `name`, `feedback`, `rating`) VALUES (%s,%s,%s)",(name,feedback,rating)
)
        conn.commit()
        return redirect(url_for('userhome',id=userid))
    else:
        return render_template("userfeedback.html",stud=student)


@app.route("/userbooktable")
def userbooktable():
    userid=request.args.get('userid')
    status="Enable"
    conn=mysql.connect()
    cur=conn.cursor()
    cur.execute("SELECT * FROM `tableno` where `userid`='"+userid+"' and `status`='"+status+"'")
    data=cur.fetchall()
    return render_template("userbooktable.html",table=data,uid=userid)
```

```
@app.route("/userchat",methods=['GET','POST'])
def userchat():
    userid=request.args.get('uid')
    tid=request.args.get('id')
    if request.method=='POST':
        uid=request.form['uid']
        tableid=request.form['tableid']
        status="Booked"
        conn=mysql.connect()
        cur=conn.cursor()
        cur.execute("UPDATE `tableno` SET `bookuid`='"+uid+"',`status`='"+status+"' WHERE `id`='"+tableid+"'")
        conn.commit()
        return redirect(url_for('userhome',id=userid))
    else:
        return render_template("userchat.html",uid=userid,tab=tid)


@app.route("/usercancelbooking",methods=['GET','POST'])
def usercancelbooking():
    status="Cancel"
    conn=mysql.connect()
    cur=conn.cursor()
    userid=request.args.get('uid')
    tid=request.args.get('tid')
    cur.execute("UPDATE `tableno` SET `status`='"+status+"' WHERE `id`='"+tid+"'")
    conn.commit()
    return redirect(url_for('userhome',id=userid))
```

```
@app.route("/get")


#function for the bot response
def get_bot_response():
    userText = request.args.get('msg')
    return str(bot.get_response(userText))


if(__name__=="__main__"):
    app.run(debug=True)
```