

## Application User Guide

### ENPM 809W - Project Phase 2: Project Implementation

**Name - Aparna Gupta**

**Email id - agupta53@umd.edu**

**UID - 119242025**

#### Part 1: Installation Steps - Python (This is a prerequisite)

- 1) The Restaurant Management System application is built using Python and Django framework. Hence, python version 3.10.x needs to be installed on your base machine or local computer in order to run this application.
- 2) To check whether python is already installed on your computer, go to 'Command Prompt' (or) 'Windows powershell' or 'Terminal' on ubuntu and enter ***python -version***. If the version of the python is 3.10.x, then you can skip the step 3 of installing python and directly go on the step 4.
- 3) Installing python (if python is not pre-installed or the version is older than 3.10.x)
  - a) Go to the link <https://www.python.org/downloads/release/python-3108/>
  - b) Scroll to the bottom and click on **Windows installer (64-bit)** or **Windows installer (32-bit)**, based on the configuration of your PC.
  - c) Wait till the .exe file downloads. Double-click the executable file to start the installation after it has been downloaded. To add the Python path to the environment variables, choose all the radio buttons.
  - d) Once installation is complete, check if python is installed by following the step 2 given above.
- 4) Creating a virtual environment:
  - a) We need to create a python virtual environment for this application.
  - b) Open a terminal and navigate to a folder where you would like the project folder to be located.  
(by using the command cd "folder name")
  - c) Run the command: ***python -m venv <specify folder name here>***  
This step will create a virtual environment with the folder name you have mentioned.
  - d) You need to first activate the virtual environment before moving forward.  
Execute the following command to activate the virtual environment:  
***...\\<the folder name specified above>\Scripts\activate***

The above command must be run in the folder where you have created the virtual environment

- e) In case, if the activation mentioned in step 4 didn't work, run the powershell as admin and execute the following command:  
***set-executionpolicy remotesigned***
- 5) After activating the virtual environment, you can proceed with running the application.

## **Part 2: Installing ‘Restaurant management system’**

- 1) On the terminal, navigate to the project folder, which has ‘manage.py’ file. You can verify whether the folder you chose contains manage.py or not by running the ‘ls’ command. This will list all the files present in that particular folder. Make sure the list contains the ‘manage.py’ file.
- 2) Once you are in the correct folder, run the command mentioned below.  
***pip install -r requirements.txt***
- 3) The above command will install all the dependencies of the project.
- 4) Verify if Django is already installed on your PC by running the following command:  
***django-admin --version***  
It must show version 4.1.2 or similar.
- 5) If Django verification is successful, then install the ‘Restaurant management system’ application by running the following command:  
***python manage.py migrate***
- 6) To create the manager user, run the following command:  
***python manage.py create\_manager\_user***
- 7) Once the above command is executed, the application is set to be run.  
In the terminal, run the following command:  
***python manage.py runserver***
- 8) After completing the above steps, the terminal would look like this -

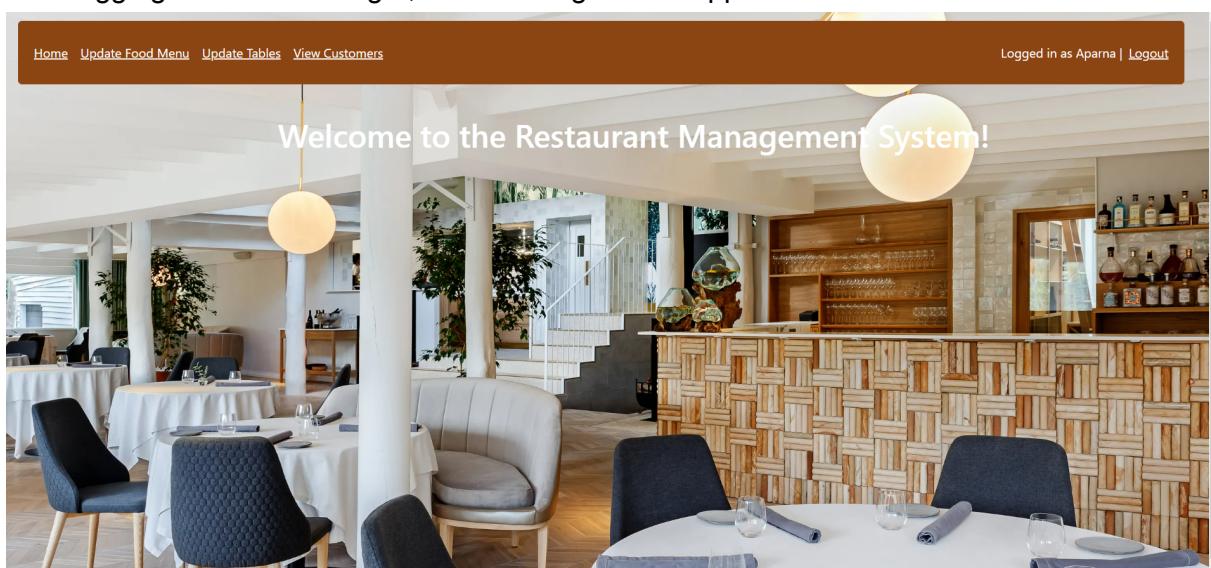
```
(Project_virtual_environment) PS C:\Users\Aparna\OneDrive\Desktop\Aparna\Restaurant Management System\RestaurantManagementSystem> python .\manage.py runserver
Performing system checks...
System check identified no issues (0 silenced).
November 13, 2022 - 18:43:42
Django version 4.1.2, using settings 'RestaurantManagementSystem.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[1]
```

- 9) The URL shown here will take you to the following page in browser :

The screenshot shows a login form for a "Restaurant Management System". The title "Restaurant Management System" is at the top. Below it is a "Login" heading. There are two input fields: "Username\*" and "Password\*". Below the password field is a link "Don't have an account? Create one [Here!](#)". At the bottom is a green "Login" button.

### Part 3: How to use this application -

- 1) Restaurant management system has 2 actors - the users/ customers and a manager (of the restaurant).
- 2) The manager has the following credentials -  
Username - Manager  
Password - Abcd!@#\$
- 3) After logging in as the manager, the following screen appears -



- 4) Now, the manager is authorised to fo 3 tasks -  
Update the food menu  
Update table information  
View the registered customers

*(Following are the use cases mentioned in the SRS)*

5) **2..2.2.1 Use case 6: Updating food menu -**

- (i) On clicking the ‘Update food menu’ option in the navigation bar, following screen will appear -

The screenshot shows a web-based application for managing a food menu. At the top, there is a navigation bar with links: Home, Update Food Menu, Update Tables, and View Customers. On the right side of the bar, it says 'Logged in as Aparna | Logout'. Below the navigation bar, the main title is 'Edit Food Menu'. There are three entries listed:

- Food Item: Pizza  
Food Description: Bread and Cheese  
Food Price: 10  
Buttons: Edit, Delete
- Food Item: Burger  
Food Description: Bun and Patty  
Food Price: 9  
Buttons: Edit, Delete
- Food Item: Pasta  
Food Description: Pasta with tomato sauce  
Food Price: 5  
Buttons: Edit, Delete

On the far right, there is a button labeled 'Add Food Item'.

- (ii) The manager can see the current items in the menu and perform 3 operations -  
Edit an item (it's name, description and price),  
Delete that item from the menu  
Add a new item to the food menu

6) **2.2.2.2 Use case 7: Updating table availability**

- (i) On clicking the ‘Update tables’ option in the navigation bar, following screen will appear -

## Tables

[Add Table](#)

Table type: 4-seater

Status: Available

[Change Status](#)

Table type: 5-seater

Status: Available

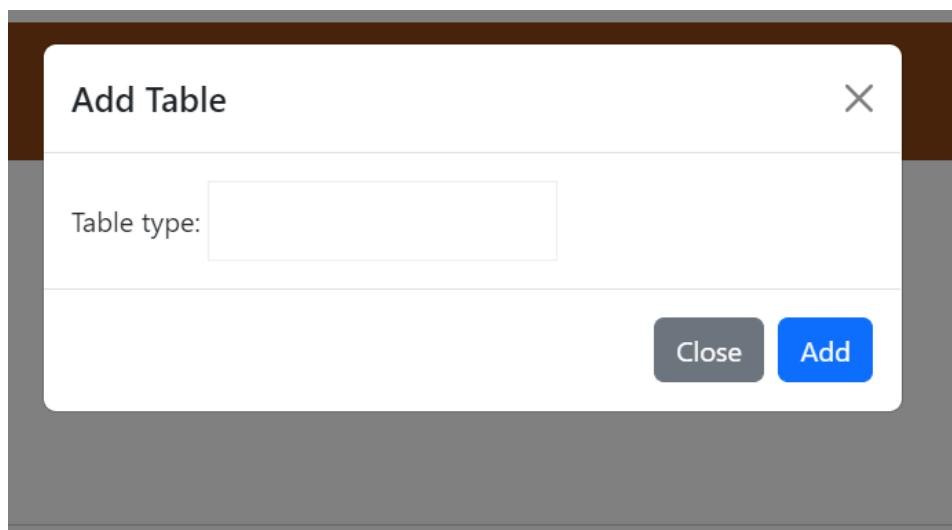
[Change Status](#)

Table type: 5-seater

Status: Available

[Change Status](#)

- (ii) Here, the manager can see all the tables present in the restaurant, along with the table type (3-seater, 4-seater etc.) and its status, that is, whether a table is available or booked.
- (iii) By clicking on the 'Change Status' button below every table, the manager can change the status of the table to either 'booked' or 'available', so that the customers using the application can see the tables available for booking.
- (iv) The Manager can also add new tables (if the capacity of the restaurant increases), by clicking on 'Update Tables' option.



(Here the manager can enter the table type like 3-seater, 4-seater etc.)

### 7) 2.2.2.3 Use case 8: View customer details

- (i) On clicking the 'View Customers' option in the navigation bar, following screen will appear -

## Customers

Full Name: abc

Email: asd@asd.com

Address: gh

Contact: 456

Here, the manager can see the information of all the customers registered in the application.

8) **2.2.1 Customer Use Cases -**  
**2.2.1.1 Use Case 1 : Create account**

(i) The following screen will first appear to the user in the browser-



The image shows a login page for a Restaurant Management System. At the top, there is a dark brown header bar with the text "Restaurant Management System". Below this, the main content area has a white background. It features a large, bold "Login" heading. Underneath the heading are two input fields: one for "Username\*" and one for "Password\*". Both fields have a thin black border. Below the password field, there is a small line of text that says "Don't have an account? Create one [Here!](#)". At the bottom left of the form is a green rectangular button with the word "Login" in white text. The overall design is clean and modern.

(ii) If the user has an account, he/she can login with their credentials. Else, they can click on the 'Here' link and create an account.

(iii) After clicking on 'Here', following screen will appear -

Username\*

Email\*

Password\*

Your password can't be too similar to your other personal information.  
 Your password must contain at least 8 characters.  
 Your password can't be a commonly used password.  
 Your password can't be entirely numeric.

Password confirmation\*

Enter the same password as before, for verification.

Address\*

Contact number\*

Already have an account? Login [Here!](#)

[Register](#)

(iv) After filling the above form and hitting the ‘Register’ button, the user account will be created and the user will be redirected to the login screen again. Here the user needs to enter the username and password that he just used while creating the account.

### 9) 2.2.1.2 Use Case 2: Customer login

(i) In the login screen, the user will have to enter his correct username and password to enter into his/her account.

Restaurant Management System

## Login

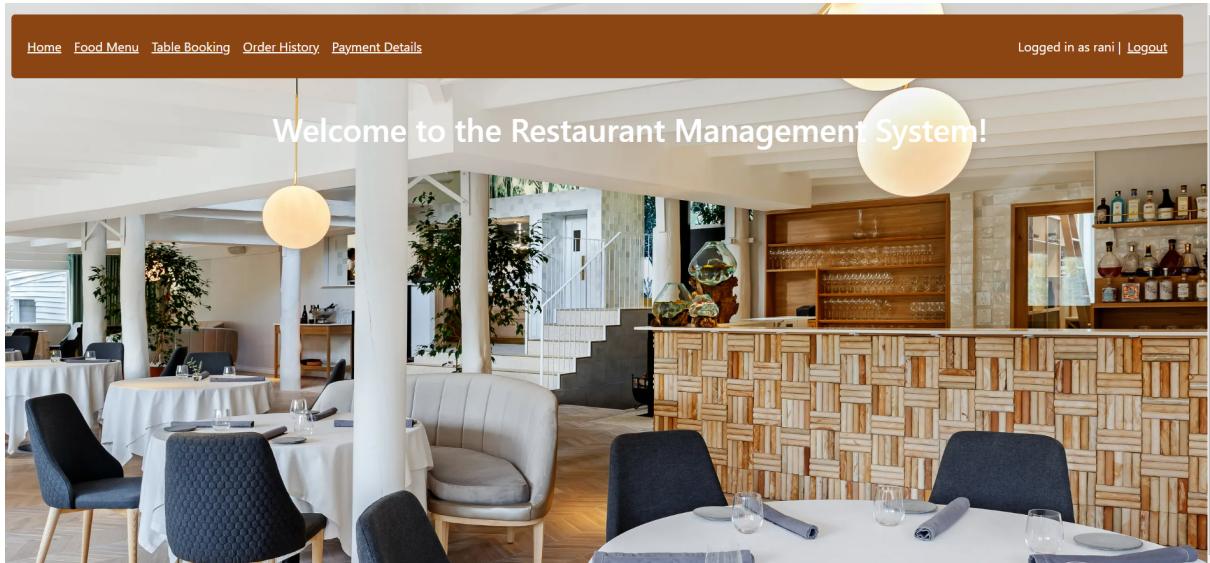
Username\*

Password\*

Don't have an account? Create one [Here!](#)

[Login](#)

(ii) After successfully logging in, the following screen will appear to the users:



#### 10) 2.2.1.3 Use Case 3: View food menu

(i) On logging in, the customer can click on the 'Food Menu' in the navigation bar to view all the items in the food menu, along with the item's corresponding description and price.

The screen looks like this:

Food Item	Description	Price
Pizza	Bread and Cheese	10
Burger	Bun and Patty	9

#### 11) 2.2.1.4 Use Case 4: Ordering food (online)

(i) The user(s) can order food items from the given menu. Under each item, there is a plus(+) and minus(-) sign with which the user can change the quantity of that particular item he wishes to order.

Example -

# Food Menu!

Food Item: Pizza

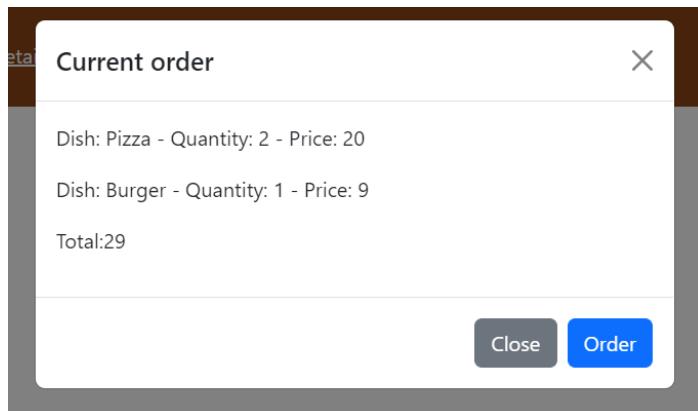
Food Description: Bread and Cheese

Food Price: 10

- 2 +

(ii) After selecting all the items that the user wants to order, he can click on the 'View Cart' button on the top-right corner to view the current items, their quantities and the total price.

Example -



(iii) After clicking on the Order button, the food in the cart can be ordered. A "Order completed" message will appear.

## 12) 2.2.1.5 Use case 5: Reservation of tables

(i) The user can book tables/ reserve tables. After clicking on the 'Table Booking' option, the following screen will appear:

A screenshot of a web page titled "Table Booking!". The page has a navigation bar at the top with links: Home, Food Menu, Table Booking, Order History, and Payment Details. On the right side of the navigation bar, it says "Logged in as rani | Logout". The main content area has three sections, each representing a table type:

- Table type: 5-seater
- Table type: 5-seater
- Table type: 6-seater

- (ii) The above list shows the list of only the tables that are available for booking. After clicking on the 'Book' button under a table, the user can book that table.
- (iii) After booking that particular table, when the user refreshes the page, the table that he just booked will no longer be shown in the list of available tables. This shows that the table has been booked successfully.

### 13) Use Case : Payment Information

- (i) The user can add card details in their account. After clicking on the 'Payment Details' option, the following screen will appear:

The screenshot shows a web page with a brown header bar containing navigation links: Home, Food Menu, Table Booking, Order History, and Payment Details. Below the header is a form with two input fields. The first field is labeled 'Cardnumber\*' and the second is labeled 'Cvv\*'. Both fields have placeholder text and are followed by horizontal line inputs. At the bottom of the form is a green 'Save' button.

- (ii) Here the user needs to enter his/her 16 digit card number and 3 digit cvv and save it.  
After saving, for security purposes, only the last 4 digits of the card number will appear on the screen and the cvv will be hidden.

The screenshot shows the same web page after saving the payment details. The header bar and form structure remain the same. Below the form, the page displays the saved information: 'Card Number: 7642' and 'Cvv: \*\*\*'.

#### 14) Use Case : Order History

- (i) The user can view his/her order history by clicking on the ‘Order History’ button.

The screenshot shows a web application interface titled 'My Order History!'. At the top, there is a navigation bar with links: Home, Food Menu, Table Booking, Order History, and Payment Details. On the right side of the header, it says 'Logged in as rani | Logout'. Below the header, the main content area has the title 'My Order History!'. Underneath the title, there are two sections of order details separated by a horizontal line:

Dish Name	Quantity	Total Price
Burger	1	9
Pizza	2	20

#### Part 4: To run the test cases -

- (i) To run the test cases in the code, use the following command -  
***python manage.py test restaurant\_app***

#### Part 5: To run the application on HTTPS (optional)

To run the application on HTTPS, execute the following commands:

- 1) Open powershell (on administrator) and run ***choco install mkcert***
- 2) Then run ***mkcert -install***
- 3) Now navigate to the project folder where manage.py file is present and run the following command:  
***mkcert -cert-file cert.pem -key-file key.pem localhost 127.0.0.1***
- 4) Now run the following command for running the application on https.

***python manage.py runserver\_plus --cert-file cert.pem --key-file key.pem***