# RESTAURANT MANAGEMENT (SERVEASY MANAGER)

**A MINI-PROJECT REPORT** ***Submitted by***

**DEVADHARSHINI K 240701626**

***in partial fulfillment of the award of the degree***

***of***

**BACHELOR OF ENGINEERING**
**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI**

**An Autonomous Institute**

**CHENNAI**

**NOVEMBER 2025**

# BONAFIDE CERTIFICATE

Certified that this project **“RESTAURANTMANAGEMENT SYSTEM”** bonafide work of “ **DEVADHARSHINI K** ” who carried out the project work under my supervision.

**SIGNATURE**

**V. SATHIYAVATHI**

**ASSISTANT PROFESSOR**

Dept. of Computer Science and Engg,
Chennai
Rajalakshmi Engineering College

This mini project report is submitted for the viva voce examination to be held on

__________

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

## ABSTRACT

In today's fast-paced dining culture, digital solutions are essential for efficient restaurant operations. While large chains benefit from advanced systems, many local restaurants lack streamlined tools for managing customer interactions and reservations. To address this gap, our team developed a restaurant management system that simplifies seat selection, cart handling, and user authentication. The platform allows customers to view available seats, reserve them after payment, and securely manage their accounts. It also enables restaurant staff to track reservations and user activity through a unified database. By integrating real-time seat availability with a user-friendly interface, this system enhances service quality and operational control. Ultimately, it empowers local restaurants to compete effectively and deliver a modern, hassle-free dining experience.

## ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN**and the chairperson **DR. M.THANGAM MEGANATHAN**for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. E.M. MALATHY** and our Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide **V. SATHIYAVATHI** for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

**1. DEVADHARSHINI K**

## TABLE OF CONTENTS

## LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 INTRODUCTION

The project helps people to know the necessary information and list of bookings and assess the feasibility in the locality. The necessary information about availability of cars and list of bookings will be mentioned according to user convenience.

## 1.2 SCOPE OF THE WORK

The cab management system will help people access the local taxi service amidst the heavy demand of taxi and transportation services. It helps the people to have quick access and easy accessibility for a wide range of people.

## 1.3 PROBLEM STATEMENT

The need for the project is that many corporate and international companies have taken over various important places and companies in many places and still the services aren't accessible to many people in many local areas and high demand for these services and causing inconvenience to people.

## 1.4 AIM AND OBJECTIVES OF THE PROJECT

The main objective of this project is to allocate the available taxis according to the customer requirement. This system helps to maintain the availability and details of taxis in a local transportation agency. This will allow the agency to beat the competition from rival agencies by providing efficient service.

# CHAPTER 2

## SYSTEM SPECIFICATIONS

### 2.1 HARDWARE SPECIFICATIONS

| | | |
|---|---|---|
| Processor | : | Intel i5 |
| HDD | : | 1 TB (Minimum) |
| Memory Size | : | 8GB (Minimum) |

### 2.2 SOFTWARE SPECIFICATIONS

| | | |
|---|---|---|
| Operating System | : | WINDOWS 10 |
| Front – End | : | HTML, CSS, JAVA SCRIPT |
| Back - End | : | SQL LITE |
| Language | : | python,SQL |

# CHAPTER 3

## MODULE DESCRIPTION

This application consists of two modules. When the program runs, it will ask for a confirmation to the login window. The person who interacts can login as an Administrator or as a User. The description of the modules are as follows:

1. **Admin login**

   When the person who interacts tries to login as Admin then he needs to login with his username and password. The administrator only has the power to change and manipulate the data in the database.

2. **User login**

   When the person tries to login as a user then he/she will be prompted to enter the number of symptoms and the final result will be printed in the form of table.

# CHAPTER 4

## CODING

```
from tkinter import *

import sqlite3

import pyttsx3

# connection to database

conn = sqlite3.connect('database.db')

c = conn.cursor()

# empty lists to append later

number = []

patients = []

sql = "SELECT * FROM appointments"

res = c.execute(sql)

for r in res:

   ids = r[0]

   name = r[1]

   number.append(ids)

   patients.append(name)
```

```
# window
class Application:
    def __init__(self, master):
        self.master = master
        self.x = 0
       # heading
        self.heading = Label(master, text="Bookings", font=('arial 60 bold'), fg='green')
        self.heading.place(x=450, y=50)
        # button to change bookings
        self.change = Button(master, text="Next Booking", width=25, height=2,
bg='steelblue', command=self.func)
        self.change.place(x=500, y=550)
        # empty text labels to later config
        self.n = Label(master, text="", font=('arial 80 bold'))
        self.n.place(x=500, y=210)
        self.pname = Label(master, text="", font=('arial 80 bold'))
        self.pname.place(x=500, y=320)
    # function to speak the text and update the text
    def func(self):
```

```
    y = str(number[self.x] - 25)

    self.n.config(text=str(y))

    self.pname.config(text=str(patients[self.x]))

    engine = pyttsx3.init()

    voices = engine.getProperty('voices')

    rate = engine.getProperty('rate')

    engine.setProperty('rate', rate-30)

    engine.say('Booking number ' + str(y) + str(patients[self.x]))

    engine.runAndWait()

    self.x += 1

root = Tk()

b = Application(root)

root.geometry("1530x780+0+0")

root.resizable(False, False)

root.mainloop()
```

**CODE 1**

depicts the display code, that gets the data from the database i.e. being stored there and represented on users demand with the layout and measurements i.e. being already specified.

```
from tkinter import *

import tkinter.messagebox

import sqlite3

import sys

conn = sqlite3.connect('database.db')

c = conn.cursor()

class Application:

  def __init__(self, master):

      self.master = master

      # heading label

      self.heading = Label(master, text="Bookings ",fg='grey', font=('arial 40 bold'))

      self.heading.place(x=550, y=20)

      # search criteria -->name

      self.name = Label(master, text="Enter Owner's Name", font=('arial 20 bold'))

      self.name.place(x=400, y=142)

      # entry for  the name

      self.namenet = Entry(master, width=35)

      self.namenet.place(x=800, y=150)
```

```
     # search button

      self.search = Button(master, text="Search", width=14, height=2, bg='steelblue', command=self.search_db)

      self.search.place(x=700, y=210)

def search_db(self):

  self.input = self.namenet.get()

  # execute sql

  sql = "SELECT * FROM appointments WHERE name LIKE ?"

  self.res = c.execute(sql, (self.input,))

  for self.row in self.res:

      self.name1 = self.row[1]

      self.age = self.row[2]

      self.gender = self.row[3]

      self.location = self.row[4]

      self.phone = self.row[5]

      self.phone = self.row[6]
```

**CODE 2**

2 depicts the booking part of the code, where it displays booking details and enter user data and store it in database

# CHAPTER 5

## SCREEN SHOTS

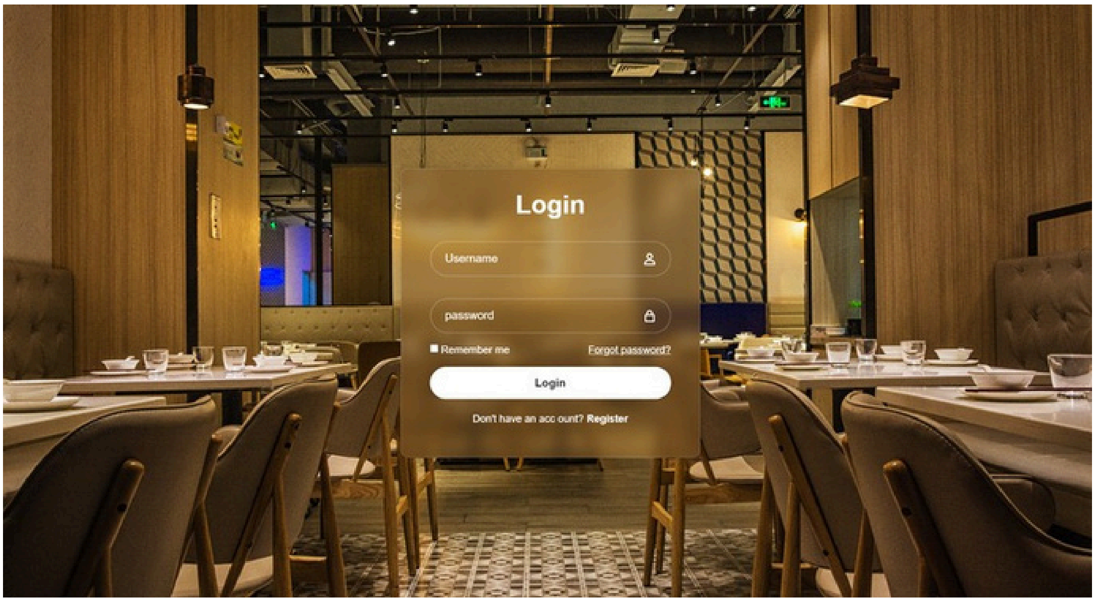
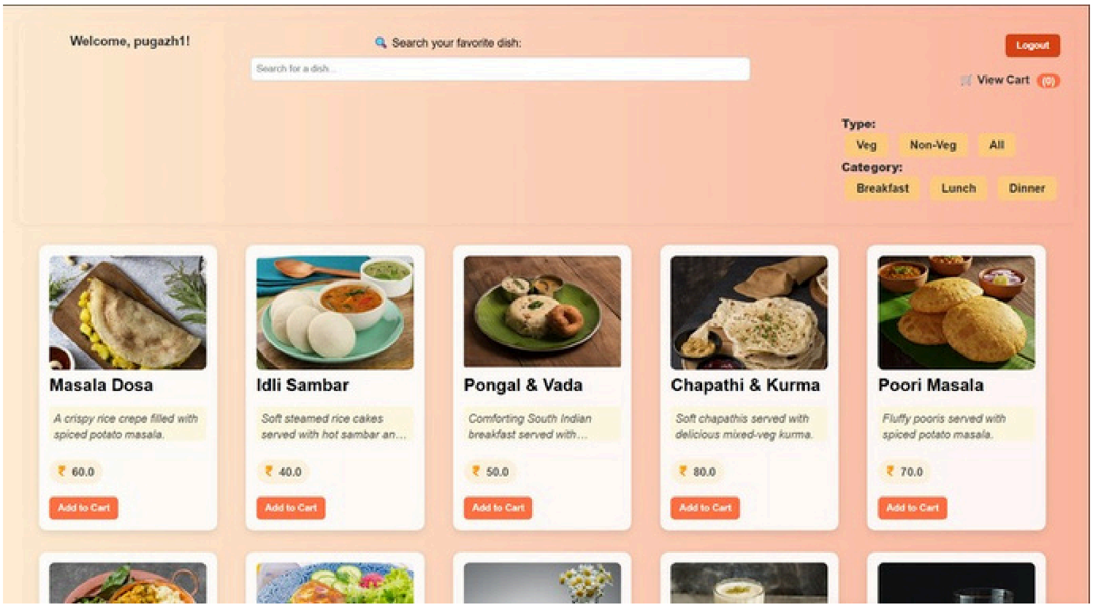


Fig 5.1 Login page



Masala Dosa

A crispy rice crepe filled with spiced potato masala.

₹ 60.0

Idli Sambar

Soft steamed rice cakes served with hot sambar an...

₹ 40.0

Pongal & Vada

Comforting South Indian breakfast served with...

₹ 50.0

Chapathi & Kurma

Soft chapathis served with delicious mixed-veg kurma.

₹ 80.0

Poori Masala

Fluffy pooris served with spiced potato masala.

₹ 70.0

Fig 5.2 Customer details

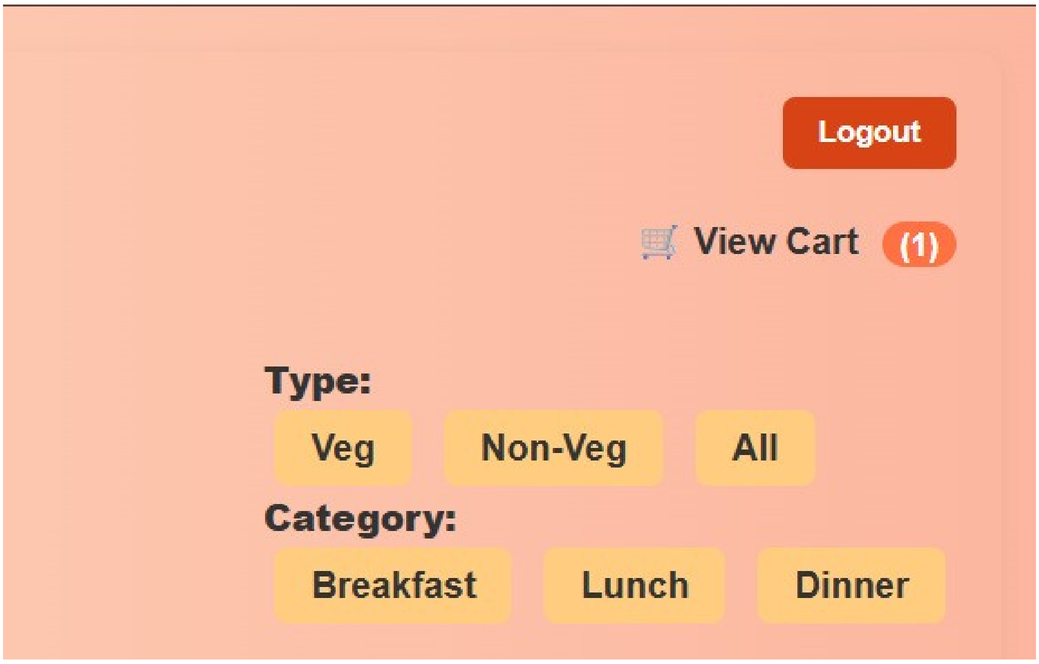
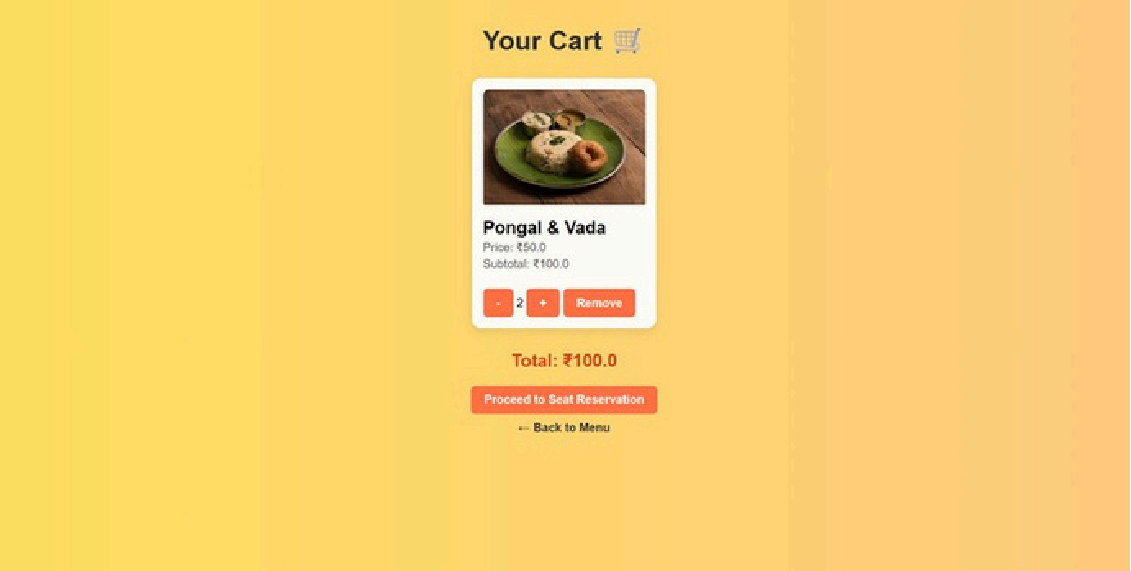


**Fig 5.3 FILTER SECTION**


Your Cart

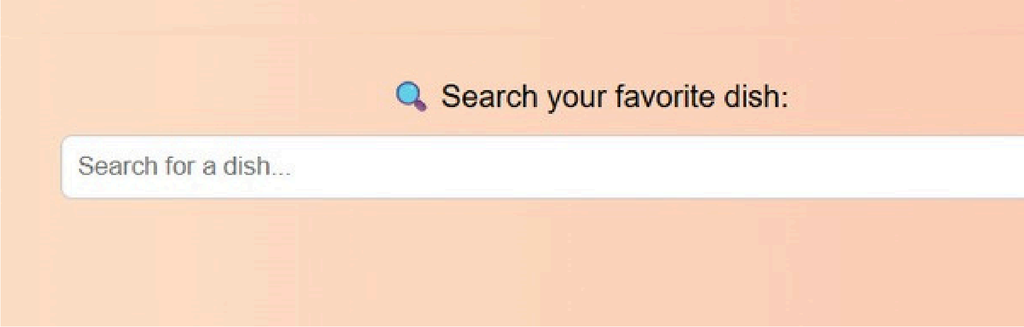
Pongal & Vada

Price: ₹50.0

Subtotal: ₹100.0

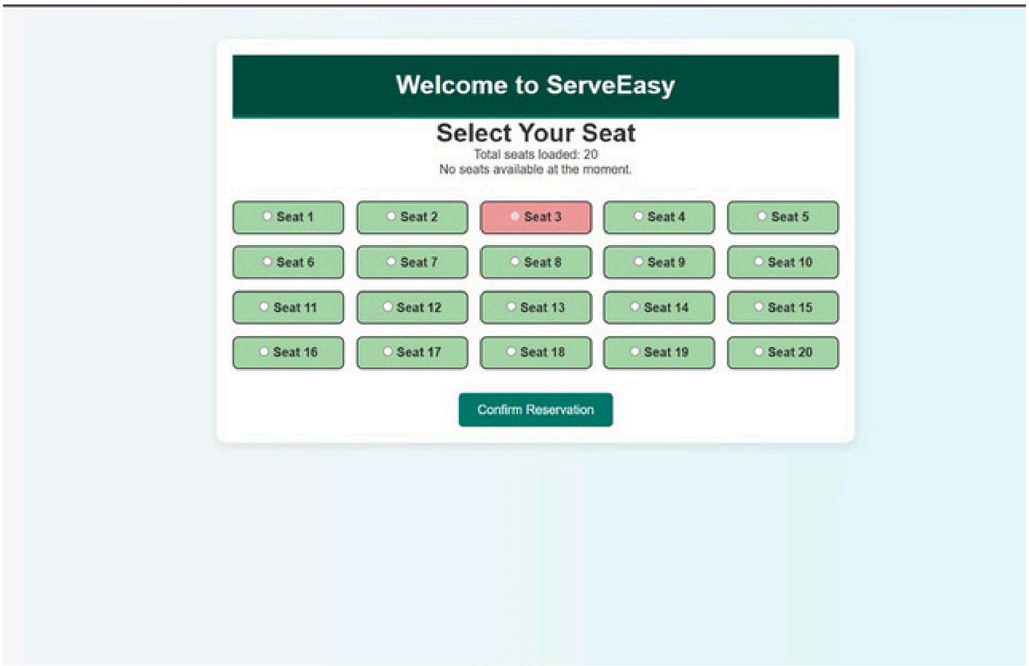
Total: ₹100.0

**Fig 5.4 Cart Page**



Search your favorite dish:

**Fig 5.5 Search Food**



Welcome to ServeEasy

Select Your Seat

Total seats loaded: 20

No seats available at the moment.

**Fig 5.6 Seat Selection**

# CHAPTER 6

## CONCLUSION AND FUTURE ENHANCEMENT

In such a way, with the help of our project, customers will be able to view available seats and reserve them based on their preferences. The reservation system clearly displays the current seat status and user details, making the booking process smooth and transparent. Through a centralized database, the restaurant can efficiently manage customer interactions and track reservations using a booking log. In future, users will be able to select seats and complete reservations based on real-time availability and system updates. Hence, this project benefits both customers and restaurant staff by simplifying operations and enhancing service quality, allowing local restaurants to deliver a modern and hassle-free dining experience.

# REFERENCES

1. https://www.w3schools.com/sql/
2. https://www.tutorialspoint.com/sqlite/index.htm
3. https://www.wikipedia.org/
4. https://www.learnpython.org/
5. https://www.codecademy.com/learn/learn-python