Cafe Delight

S.E. mini-project report submitted in partial

fulfilment of the requirements of the degree of

Information Technology

by

Ansari Mohammad Rafey (01)

Roshan Ashok Limbani (31)

Sumit Kumar Singh (57)

Harshikesh Rajendra Yadav (67)

Under the guidance of

Dr. Yogita Mane
(HOD)

Department of Information Technology

Universal College of Engineering, Kaman, Vasai

University of Mumbai

2021–2022

CERTIFICATE

This is to certify that the S.E. mini-project entitled **"Cafe Delight"** is a bonafide work of **"Ansari Mohammad Rafey" (01), "Roshan Ashok Limbani" (31),** and **"Sumit Kumar Singh"(57) and "Harshikesh Rajendra Yadav"(67)** submitted to University of Mumbai in partial fulfilment of the requirement for the award of the degree of **"Information Technology Engineering"** during the academic year 2021–2022.

Dr. Yogita Mane

HOD

Dr. Yogita Mane	**Dr. J.B.Patil**
Head of Department	Principal

S.E. Mini-Project Report Approval

This mini-project synopsis entitled **Cafe Delight** by **Ansari Mohammad Rafey, Roshan Ashok Limbani, Sumit Kumar Singh, Harshikesh Rajendra Yadav** is approved for the degree of ***Information Technology Engineering*** from ***University of Mumbai***.

Examiners

1.----------

2.----------

Date:

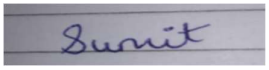
Place:

Declaration

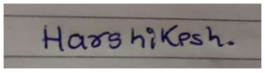
We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Ansari Mohammad Rafey (01)

Roshan Ashok Limbani (31)



Sumit Kumar Singh (57)



Harshikesh Rajendra Yadav (67)

Date:

24/04/2022

Abstract

A coffee shop has always been more than a place to have a beverage. It is a place to socialize, ideate, have fun and meet new people. As Starbucks puts it, it's the third place, a home away from home or office. Also, due to the competitive and busy lifestyles that people lead today, a coffee house near their work or home has become a major point of relaxation.

Almost 80% of the population are coffee lovers. Cafe Delight is guaranteed to become the daily necessity for all the coffee addicts. A place with good ambience where people can escape from their daily stress and cherish with a morning cup of coffee. Our café offers home style delicious breakfast and snacks. We focus on finding the most aromatic and exotic coffee beans.

You've have probably visited or at least seen website or application of many café like Starbucks. We have tried to make such application for a local coffee shop like café delight. At start welcome page of café appears, next there is login page and after that menu page afterwards there is cart and then delivery page.

Table of Contents

Chapter 1

Introduction

Café Delight has created a software based on a concept to maintain orders and management of their coffee shop. There are many sections in this project, they are Login, Coffee Ordering, Cart etc. By selecting Coffee Order the system displays a list of items and the user has to place an order with item quantity. After that, he/she precedes towards Order confirmation and Payment methods. The payment method is kept only in Cash on delivery.

The goal is to provide local café opportunity to represent themselves on the online platform as many people nowdays prefer to make online order rather than going to shop and standing in queue.

1.1 Motivation:

This system can be implemented to any cafe in the locality or to multinational branded cafe having retail outlet chains. The system recommends a facility to accept the orders 24*7 and a home delivery system which can make customers happy.

If cafés are providing an online portal where their customers can enjoy easy delivery from anywhere, the cafés won't be losing any more customers to the trading online cafés such as starbucks and mcafe. Since the application is available in the smartphone it is easily accessible and always available.

1.2 Problem Statement:

The problem is to design a Online platform for café which provides:

- Login page for new and existing customer
- Menu Page
- Items will be added on the cart based on user choice
- Delivery Page

Objectives

The objectives are as follows:

- To design a online café which can be satisfies the need of both seller and user
- To implement a python program which is capable of creating digital café which is convenient and can be modified

1.4 Scope:

The system of café delight is being developed using Python language. This application has been created with Tkinter module. This system generates various knowledge about Graphical User Interface (GUI). User has to just login and order their favorite item. Each item added will give profit to locals.

CHAPTER 2

REVIEW OF LITERATURE

Growth of food tech a comparative study of food delivery services in India, Mr. Mustafa Abbas Bhitvawala, Nishant Bidichandani, Harsh Bailhalimath and Mr. M.P Khond. The study compares growth and operating strategies of four such aggregator food delivery companies in a booming India market (Swiggy, Zomato, FoodPanda and Tinyowl). A combined result of a swot analysis along with a comparative analysis of models found that there are few bottlenecks to early food aggregator services like Scalability, Innovation, Cash-burn.

The information gathered by the researchers focuses on the development of e-commerce or online marketing and its advantages to the economy, consumers and business. Security measurement information in terms of payment and consumer's privacy to be considered in developing an online system are also obtained.

The researchers also sum up the different techniques on how to gain web visitors and eventually convince them to purchase the products based on some surveys gathered by the past studies. Some of the programming background skills needed to create a website is also being discussed further. Survey was conducted with questionnaire adopted from tested item scale. Around100 questionnaires were collected to utilize for empirical study. The prior literature views on the technology adoption, service quality and the increase in usage of online food delivery app are studied.

Chapter 3

Requirement Analysis

We all have buy the coffee from cafe, you never want to be in such a long queue for a cup of coffee. As a kids, teens, Adults or old aged, all love the coffee and they never want to delayed for a zip of coffee. So as techies, we have created a system by which whenever they want coffee can order and can get as possible as they want. This is what Before moving on, lets's have a quick look at all the sub-bits that build Café delight system in python.

1. Hardware Requirements:-

i. Laptop or PC

- i3 processor system or higher
- 4 GB RAM or higher
- 100 GB ROM or higher

2. Software Requirements:-

i. Laptop or PC

- Pycharm
- Tkinter module

Flowchart:

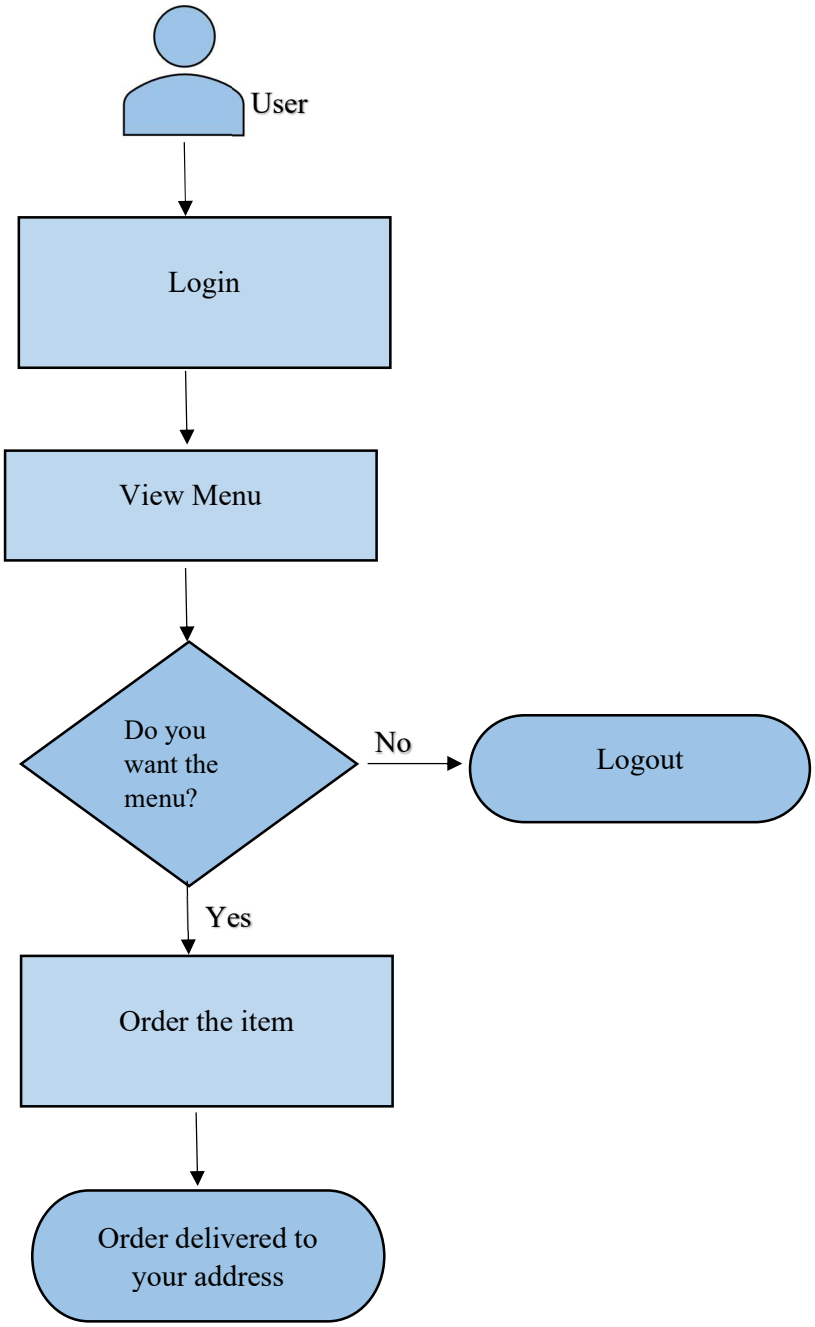


Fig 1.1 System Architecture

Algorithm:

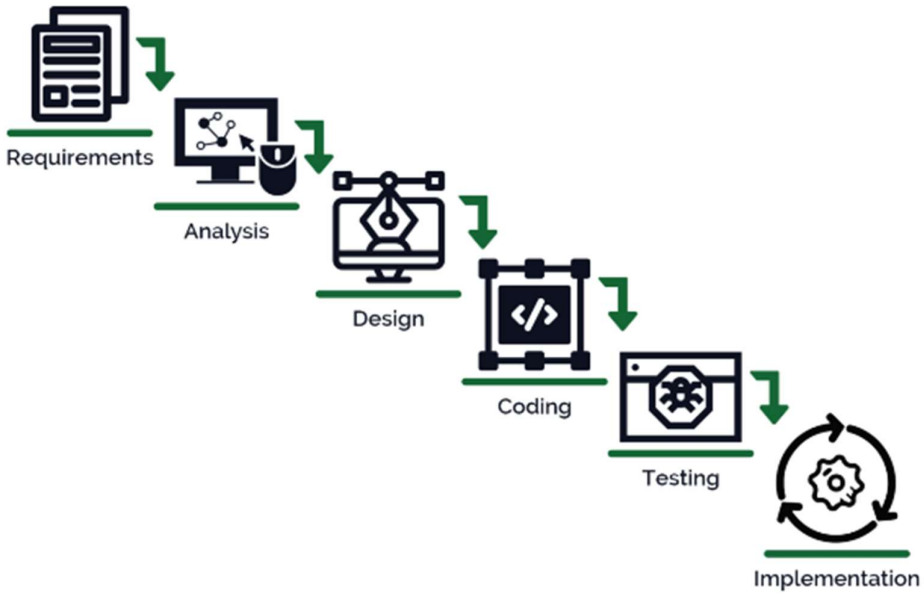


Fig 1.2 System Algorithm

INPUT:-

1.1 Here we design the welcome page GUI of our café using Tkinter module.

```
from tkinter import *


class Welcome:

    def __init__(self):
        self.welcome = Tk()
        self.welcome.title("Welcome to Coffee Maker")
        self.welcome.config(padx=50, pady=50, bg="#2B0F0E")

        # Welcome label
        self.welcome_label = Label(text="Welcome to Yaduvanshi Cafe", font=("Comics", 25, "bold"), fg="#2B0F0E")
        self.welcome_label.place(x=100, y=-50)

        self.canvas = Canvas(width=700, height=575, bg="#ae7d5b", highlightthickness=0)
        self.welcome_logo = PhotoImage(file="Images/6-2-coffee-png-hd.png")
        self.canvas.create_image(400, 250, image=self.welcome_logo)
        self.canvas.grid(column=2, row=3)

        # next button
        self.next_button = Button(text="NEXT→", font=("Times New Roman", 15, "bold"), highlightthickness=0, bg="#ae7d5b",
                                  fg="#2B0F0E", width=15, command=self.next_page)
        self.next_button.grid(column=3, row=3)

        # Emoji
        self.coffee_emoji = Label(text="☕", font=("Times New Roman", 75, "bold"), fg="#ae7d5b", bg="#2B0F0E")
        self.coffee_emoji.place(x=725, y=75)
        self.cookie_emoji = Label(text="🍪", font=("Times New Roman", 75, "bold"), fg="#ae7d5b", bg="#2B0F0E")
        self.cookie_emoji.place(x=725, y=375)
        self.welcome.mainloop()

    def next_page(self):
        self.welcome.destroy()
        from user_details import UserDetails
        users_details = UserDetails()
```

Fig 1.1 Welcome Page Mechanism

1.2 Now user have to login in the app for ordering. The data will be saved in json file.

```
from tkinter import *
from tkinter import messagebox
import json


class UserDetails:

    def __init__(self):
        self.user_details = Tk()
        self.user_details.title("User Details")
        self.user_details.config(bg="#2B0F0E")
        self.user_details.minsize(1000, 600)

        # Logo
        self.users = Canvas(width=300, height=300, highlightthickness=0, bg="#2B0F0E")
        self.user_logo = PhotoImage(file="Images/285645_user_icon (1).png")
        self.users.create_image(150, 150, image=self.user_logo)
        self.users.place(x=350, y=0)

        # Name
        self.name_label = Label(text="Name : ", font=("Times New Roman", 25, "bold"), fg="#ae7d5b", bg="#2B0F0E")
        self.name_label.place(x=34, y=350)
        self.name_input = Entry(width=50, fg="#2B0F0E", bg="#ae7d5b", font=("Times New Roman", 12, "bold"))
        self.name_input.focus()
        self.name_input.place(x=150, y=365)

        # Mobile
        self.mobile_label = Label(text="Mobile No : ", font=("Times New Roman", 25, "bold"), fg="#ae7d5b", bg="#2B0F0E")
        self.mobile_label.place(x=34, y=425)
        self.mobile_input = Entry(width=25, fg="#2B0F0E", bg="#ae7d5b", font=("Times New Roman", 12, "bold"))
        self.mobile_input.place(x=212, y=438)

        # Address
        self.address_label = Label(text="Address : ", font=("Times New Roman", 25, "bold"), fg="#ae7d5b", bg="#2B0F0E")
        self.address_label.place(x=525, y=400)
        self.address_input = Text(height=5, width=35, fg="#2B0F0E", bg="#ae7d5b", font=("Times New Roman", 12, "bold"))
        self.address_input.place(x=670, y=375)

        # Next
        self.next_button = Button(text="Login", width=15, font=("Times New Roman", 15, "bold"), fg="#2B0F0E",
                                  bg="#ae7d5b", command=self.save)
        self.next_button.place(x=775, y=520)

        # Search
        self.search_button = Button(text="Search", width=15, font=("Times New Roman", 15, "bold"), fg="#2B0F0E",
                                    bg="#ae7d5b", command=self.find_user)
        self.search_button.place(x=600, y=520)
        self.user_details.mainloop()

    def save(self):
        name = self.name_input.get().title()
        mobile_no = self.mobile_input.get()
        address = self.address_input.get("1.0", 'end-1c')
        new_data = {
            mobile_no: {
                "name": name,
                "address": address,
            }
        }
```

Fig 1.2 User Data Mechanism

1.3 After login into the app the user will be directed into menu page where he/she will be selecting the item form the menu.

```
from cart import *
from tkinter import *


rupees = "₹"

class Menu:

    def __init__(self):
        self.menu = Tk()
        self.menu.title("Menu")
        self.menu.config(padx=5, pady=5, bg="#ae7d5b")


        # Coffee 1
        self.coffee_1 = Canvas(width=250, height=275, bg="#ae7d5b", highlightthickness=0)
        self.menu1_logo = PhotoImage(file="Images/coffee_1.png")
        self.coffee_1.create_image(150, 100, image=self.menu1_logo)
        self.coffee_1.grid(column=0, row=0)
        self.latte_label = Label(text=f"Latte = {rupees}250", font=("Times New Roman", 15, "bold"),
                                 highlightthickness=0, bg="#2B0F0E", fg="white")
        self.latte_label.place(x=175, y=210)


        self.add1_button = Button(text="+", font=("Times New Roman", 15, "bold"), highlightthickness=0, bg="#2B0F0E",
                                  fg="white", command=latte_plus_price)
        self.add1_button.place(x=125, y=150)
        self.minus1_button = Button(text="−", font=("Times New Roman", 15, "bold"), highlightthickness=0, bg="#2B0F0E",
                                    fg="white", command=latte_minus_price)
        self.minus1_button.place(x=300, y=150)
        # Coffee 2
        self.coffee_2 = Canvas(width=325, height=275, bg="#ae7d5b", highlightthickness=0)
        self.menu2_logo = PhotoImage(file="Images/coffee_2.png")
        self.coffee_2.create_image(150, 100, image=self.menu2_logo)
        self.coffee_2.grid(column=1, row=0)
        self.cold_coffee = Label(text=f"Cold Coffee = {rupees}200", font=("Times New Roman", 15, "bold"),
                                 highlightthickness=0, bg="#2B0F0E", fg="white")
        self.cold_coffee.place(x=570, y=210)
        self.add2_button = Button(text="+", font=("Times New Roman", 15, "bold"), highlightthickness=0, bg="#2B0F0E",
                                  fg="white", command=cold_coffee_plus_price)
        self.add2_button.place(x=535, y=150)
        self.minus2_button = Button(text="−", font=("Times New Roman", 15, "bold"), highlightthickness=0, bg="#2B0F0E",
                                    fg="white", command=cold_coffee_minus_price)
        self.minus2_button.place(x=720, y=150)



        # Coffee 3
        self.coffee_3 = Canvas(width=325, height=275, bg="#ae7d5b", highlightthickness=0)
        self.menu3_logo = PhotoImage(file="Images/coffee_3.png")
        self.coffee_3.create_image(150, 100, image=self.menu3_logo)
        self.coffee_3.grid(column=2, row=0)
        self.cappuccino_label = Label(text=f"Cappuccino = {rupees}300", font=("Times New Roman", 15, "bold"),
                                      highlightthickness=0, bg="#2B0F0E", fg="white")
        self.cappuccino_label.place(x=1000, y=210)
        self.add3_button = Button(text="+", font=("Times New Roman", 15, "bold"), highlightthickness=0, bg="#2B0F0E",
                                  fg="white", command=cappuccino_plus_price)
        self.add3_button.place(x=925, y=150)
        self.minus3_button = Button(text="−", font=("Times New Roman", 15, "bold"), highlightthickness=0, bg="#2B0F0E",
                                    fg="white", command=cappuccino_minus_price)
        self.minus3_button.place(x=1200, y=150)
```

Fig 1.3 Menu Page Mechanism

1.4 After selecting the item from the menu the item will be added to cart. Where bill will be also displayed.

```
class Cart:

    def __init__(self):
        self.latte = 0
        self.cart = Tk()
        self.cart.title("Cart")
        self.cart.minsize(width=1000, height=600)
        self.cart.config(padx=10, pady=10, bg="#ae7d5b")
        self.cart_label = Label(text="Your Food Cart 🛒😊", font=("Times New Roman", 45, "bold"), highlightthickness=0,
                                bg="#ae7d5b", fg="#2B0F0E")
        self.cart_label.place(x=250, y=0)

        def menu():
            self.cart.destroy()
            from menu import Menu
            menu_page = Menu()

        self.menu_button = Button(text="Go back To Menu", font=("Times New Roman", 15, "bold"), highlightthickness=0,
                                  bg="#2B0F0E",
                                  fg="white", command=menu)
        self.menu_button.place(x=50, y=100)

        # Latte
        self.latte = Label(text=f"latte = {l_quantity} x {rupees}250 = {rupees}{latte_price}",
                           font=("Times New Roman", 25, "bold"), highlightthickness=0, bg="#2B0F0E", fg="white")
        if latte_price <= 0 and l_quantity <= 0:
            self.latte.destroy()
        else:
            self.latte.place(x=125, y=150)
```

```
        # Cold Coffee
        self.cold_coffee = Label(text=f"Cold Coffee = {cc_quantity} x {rupees}200 = {rupees}{cold_coffee}",
                                 font=("Times New Roman", 30, "bold"), highlightthickness=0, bg="#2B0F0E", fg="white")
        if cold_coffee <= 0 and cc_quantity <= 0:
            self.cold_coffee.destroy()
        else:
            self.cold_coffee.place(x=125, y=200)

        # Cappuccino
        self.cappuccino = Label(text=f"Cappuccino = {cp_quantity} x {rupees}300 = {rupees}{cappuccino}",
                                font=("Times New Roman", 30, "bold"), highlightthickness=0, bg="#2B0F0E", fg="white")
        if cappuccino <= 0 and cp_quantity <= 0:
            self.cappuccino.destroy()
        else:
            self.cappuccino.place(x=125, y=260)

        # Cookies
        self.cookies = Label(text=f"Choco Cookies = {co_quantity} x {rupees}150 = {rupees}{cookies}",
                             font=("Times New Roman", 30, "bold"), highlightthickness=0, bg="#2B0F0E", fg="white")
        if cookies <= 0 and co_quantity <= 0:
            self.cookies.destroy()
        else:
            self.cookies.place(x=125, y=320)

        # Pizza
        self.pizza = Label(text=f"Pizza = {p_quantity} x {rupees}400 = {rupees}{pizza}",
                           font=("Times New Roman", 30, "bold"), highlightthickness=0, bg="#2B0F0E", fg="white")
        if pizza <= 0 and p_quantity <= 0:
            self.pizza.destroy()
        else:
            self.pizza.place(x=125, y=380)
```

Fig 1.4 Cart Page Mechanism

1.5 After the billing the user will get to know the status of his/her order.

```
from cart import *
from tkinter import *
from datetime import *
import random

delivery_boy = random.choice(["Harshikesh", "Rohan", "Sachin", "Salim"])


delivery = Tk()
delivery.title("Deliver the Food")
delivery.minsize(width=1500, height=750)
delivery.config(padx=10, pady=10, bg="#ae7d5b")

payment = Label(text=f"Pay {rupees}{total} Cash. When you get your food delivered",
                font=("Times New Roman", 40, "bold"), fg="#ae7d5b", bg="#2B0F0E")
payment.place(x=100, y=15)

def order_received():
    order_taken = Label(text=f"At {datetime.now().hour}:{datetime.now().minute} Your order has been received.",
                        font=("Times New Roman", 40, "bold"), fg="#ae7d5b", bg="#2B0F0E")
    order_taken.place(x=100, y=200)


def order_prepare():
    order_prepared = Label(text=f"At {datetime.now().hour}:{datetime.now().minute} Your order is being Prepared✓.",
                           font=("Times New Roman", 40, "bold"), fg="#ae7d5b", bg="#2B0F0E")
    order_prepared.place(x=100, y=300)
```

```
def order_dispatched():
    order_routed = Label(text=f"At {datetime.now().hour}:{datetime.now().minute} Your order is being "
                              f"Dispatched by {delivery_boy}.",
                         font=("Times New Roman", 40, "bold"), fg="#ae7d5b", bg="#2B0F0E")
    order_routed.place(x=100, y=400)

def order_delivered():
    order_deliver = Label(text=f"At {datetime.now().hour}:{datetime.now().minute} Your order is being Delivered☺☺.",
                          font=("Times New Roman", 40, "bold"), fg="#ae7d5b", bg="#2B0F0E")
    order_deliver.place(x=100, y=500)


delivery.after(10000, order_received)
delivery.after(60000, order_prepare)
delivery.after(120000, order_dispatched)
delivery.after(180000, order_delivered)



delivery.mainloop()
```

Fig 1.5 Delivery Page Mechanism

OUTPUT:

2.1 Welcome page has been created

Fig 2.1 Welcome Page

2.2 After greeting the user, he/she need to login in to the app. If the user is existing then he can search his data by his phone no.

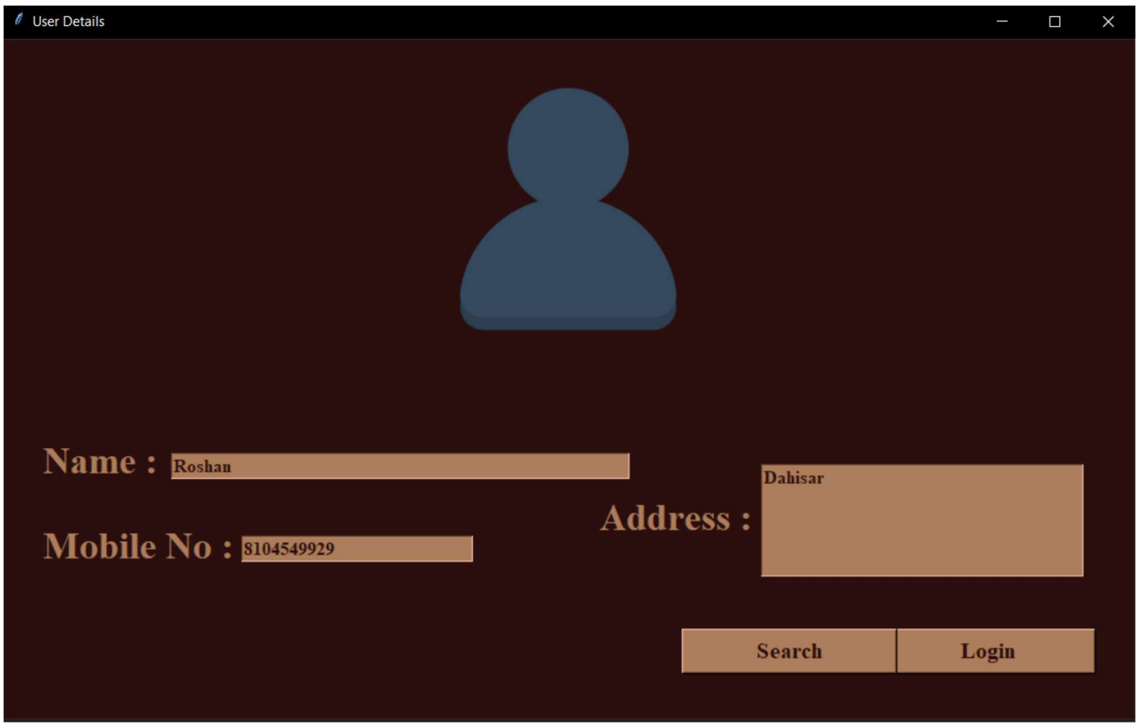


Fig 2.2 User's Data

2.3 If the user is searching for that phone number which is not register then system will throw the error of data not found.

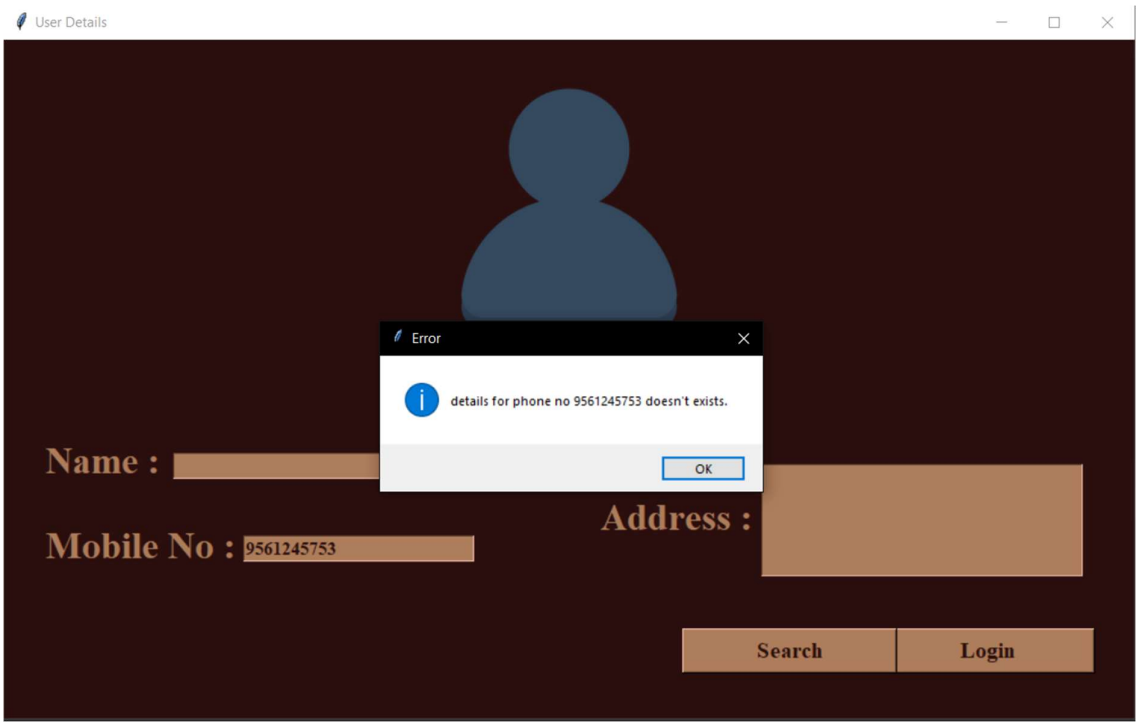


Fig 2.3 User's Data Not Found Error

2.4 If user has entered any numeric value in name or any alphabet in phone number then system will throw the syntax error.

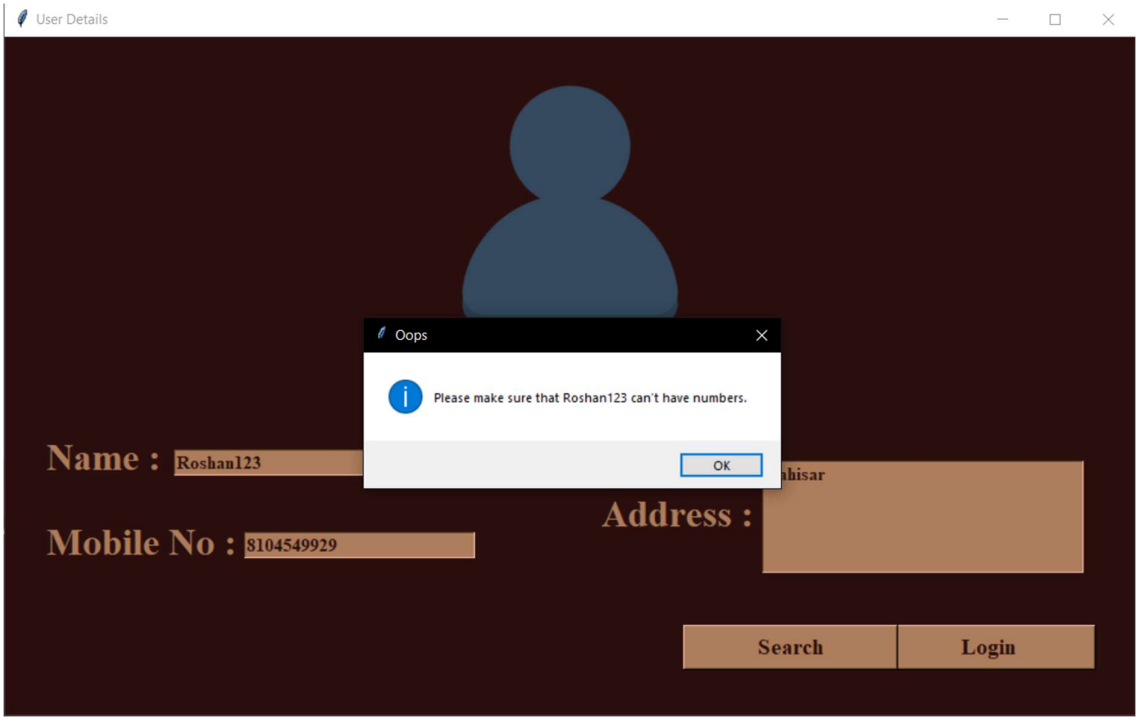


Fig 2.4(a) User's Data Syntax Error in Name

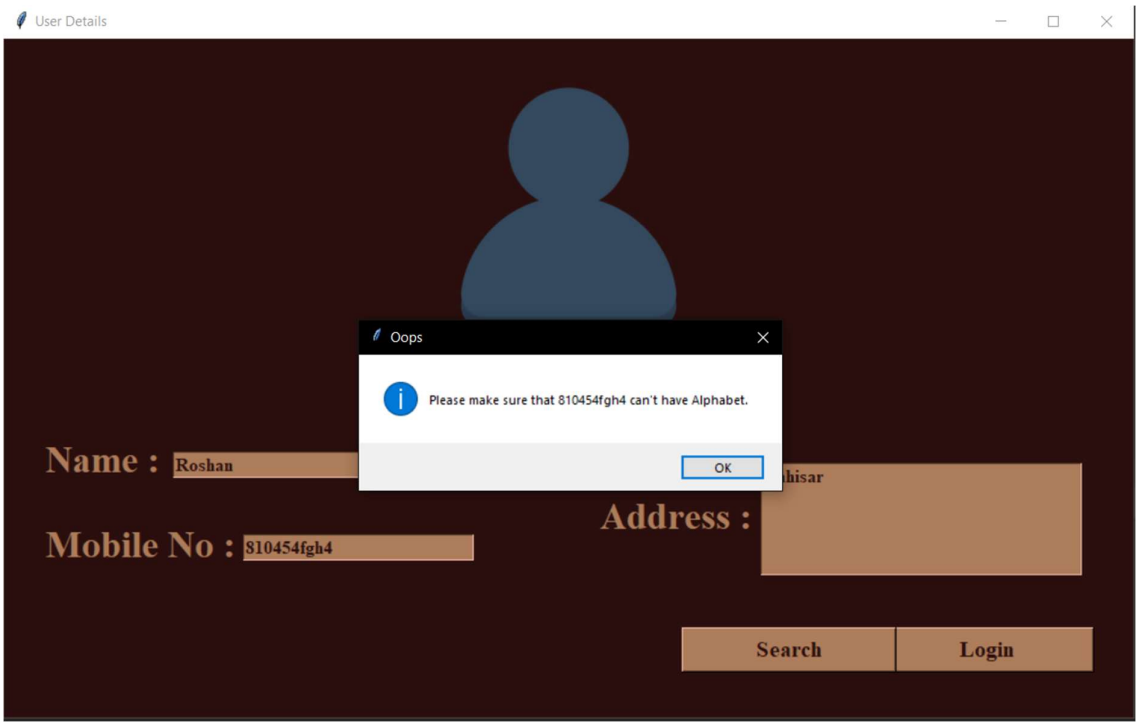


Fig 2.4(b) User's Data Syntax Error in Phone No

2.5 After logging in to the application the user will be address to menu page where he/she will be adding the item to cart

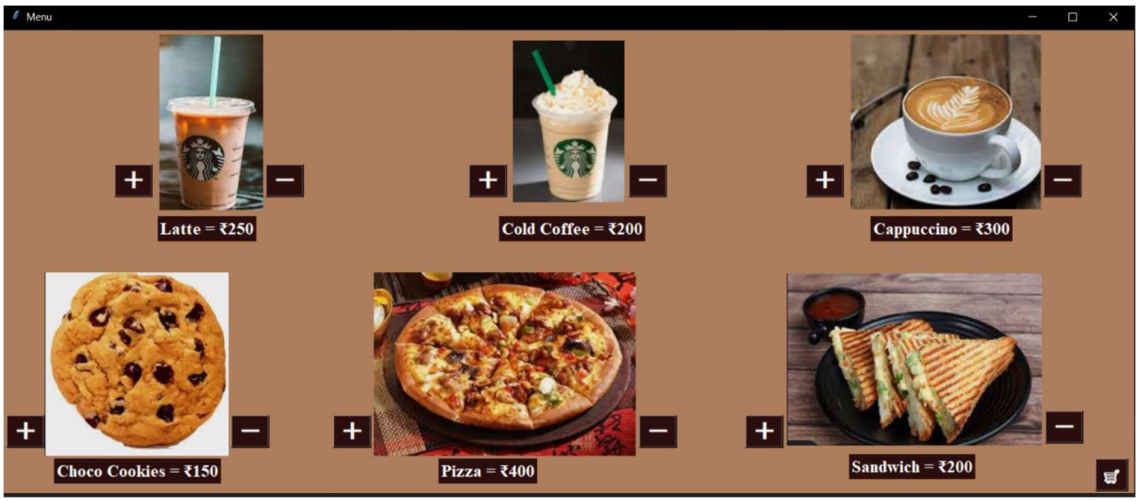


Fig 2.5 Menu Page

2.6 First if you click the cart symbol it will take you to cart where your cart will be empty. If you try to place the order without adding the item, system will give you error of empty cart. So you can go back to menu and add the item in your cart.

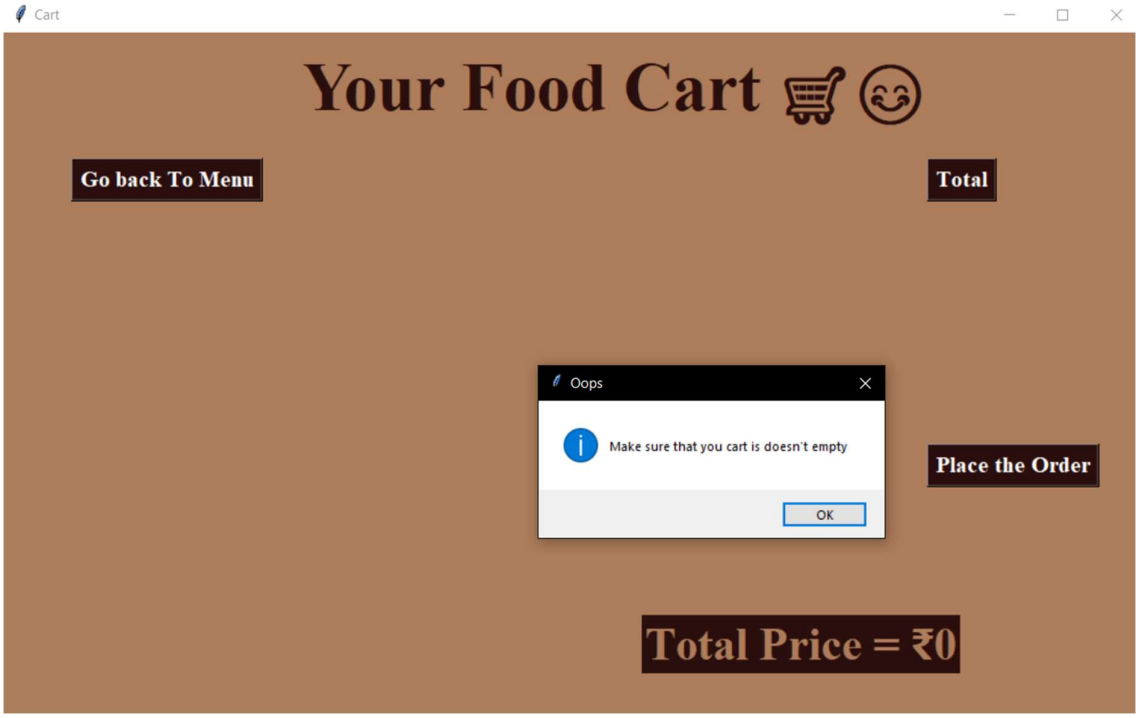


Fig 2.6 Empty Cart Error

2.7 After going back to menu add the items by clicking the plus symbol. And clicking the total button it will give you the total price of your cart. If you have added extra item so you can go back to menu and remove the item from cart.

Fig 2.7 Cart Page

2.8 After clicking the place the order button the order will be placed to your address. The payment method is kept only on cash on delivery. The moment you click it will display like this.

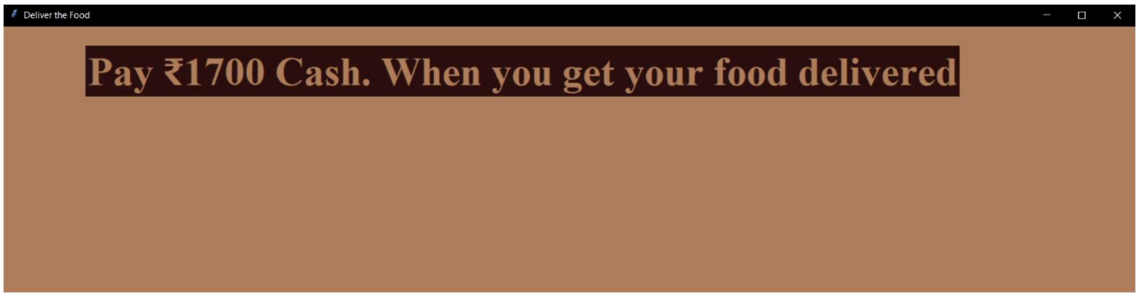


Fig 2.8 Delivery Page

2.9 The system will also give you the track of you order. After getting your food you can close the app.

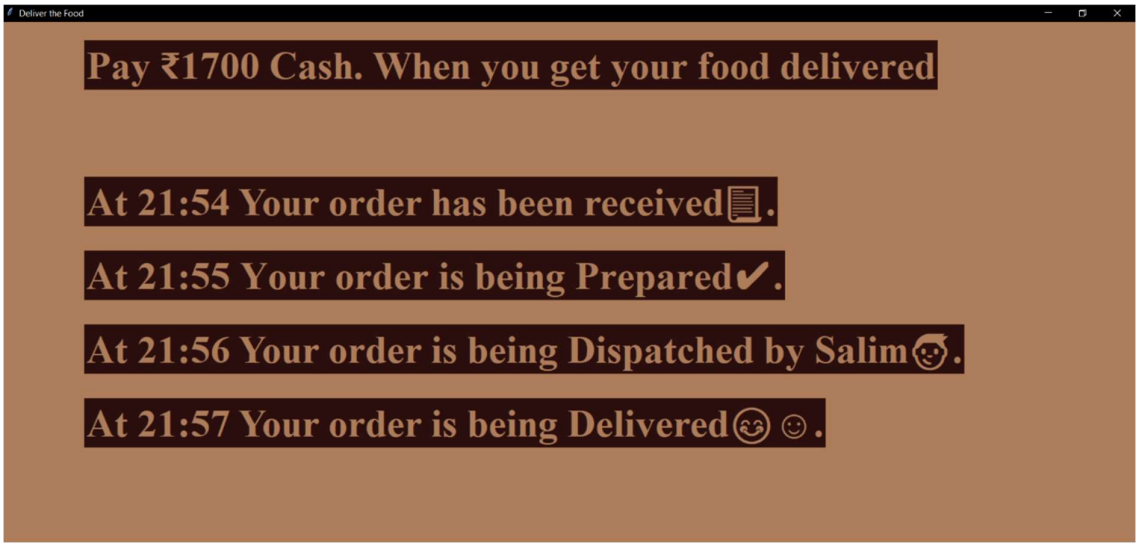


Fig 2.9 Order Status

Survey:

We have also done the surveys of our system through Google Form.

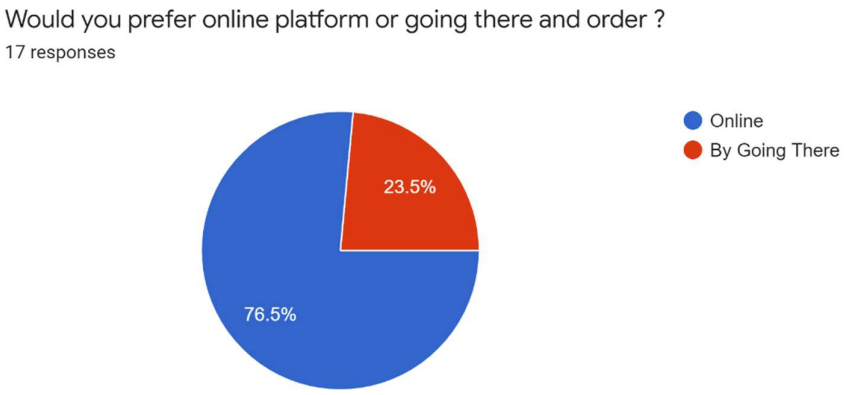


Fig 3.1 User's Response on online or going there

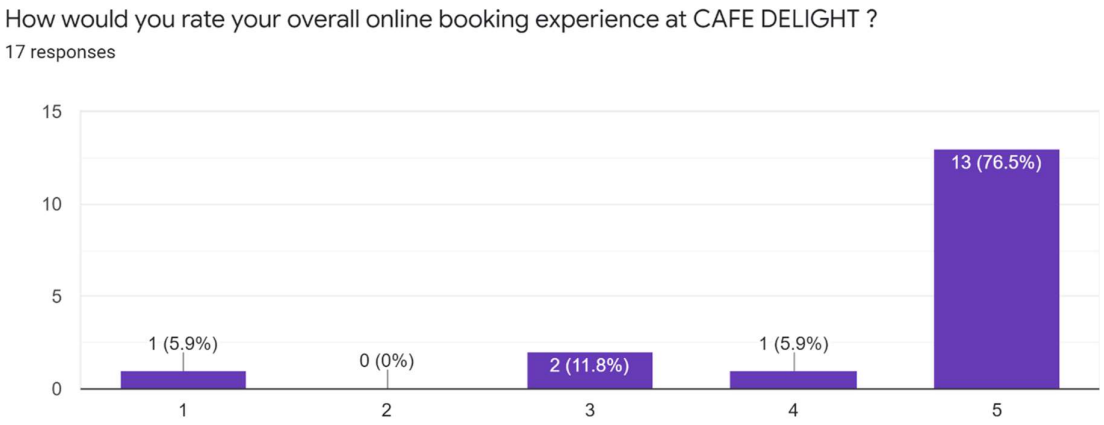


Fig 3.2 User's Online Experience at Café Delight System

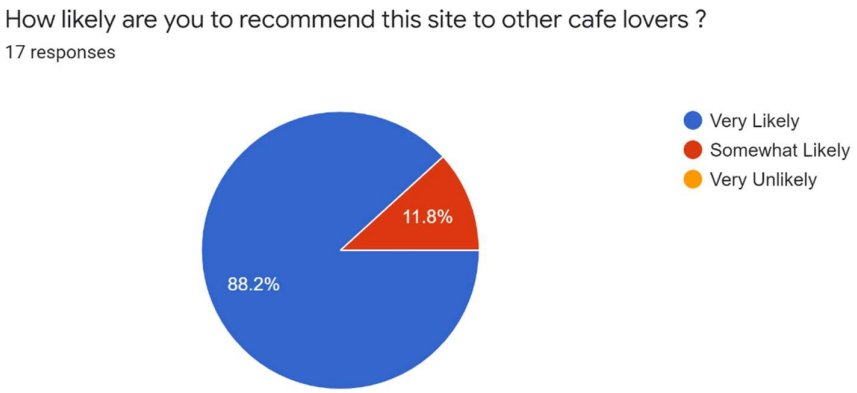


Fig 3.3 User's Recommendation to other's

Conclusion:

This project entitled Café Delight was completed successfully. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop an application for purchasing coffee from a cafe.

This project helped us in gaining valuable information and practical knowledge on several topics like designing GUI, storing data in json format & some others module of Python language. The entire system is secured, Also the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project.

Future Work:

Based on the survey users have given their suggestion & future work what we should can do to make this system more user friendly.

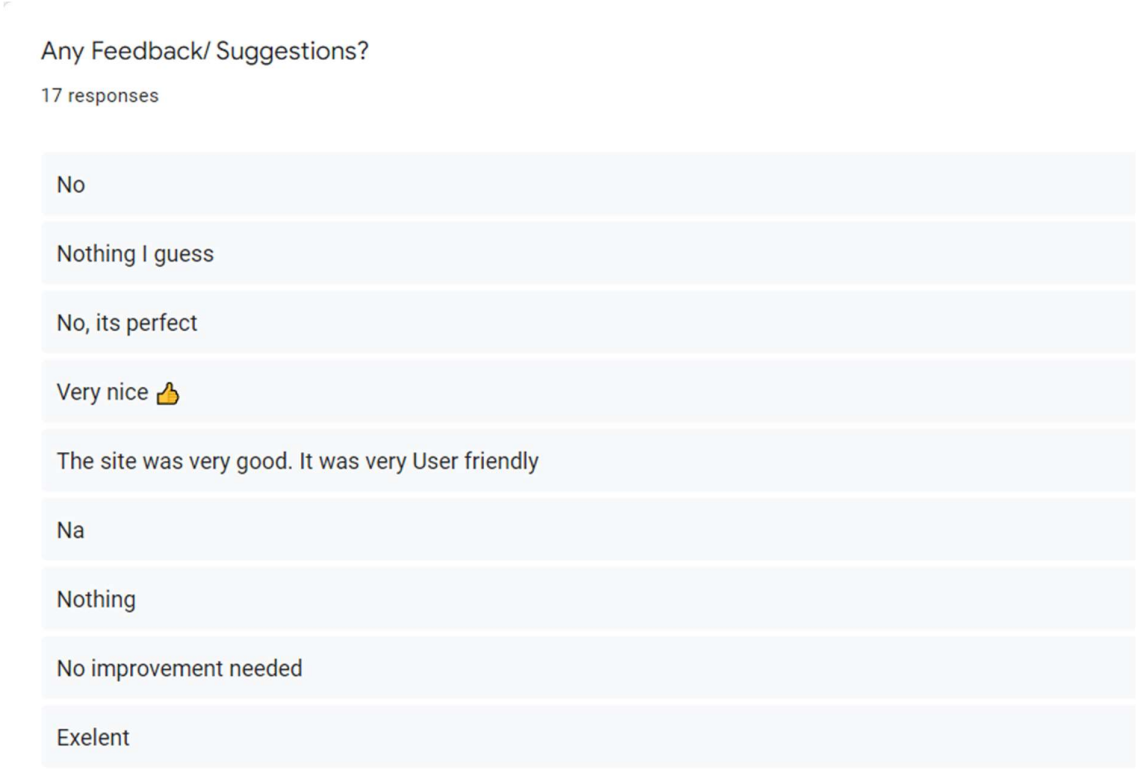


Fig 4.1 User's Suggestions

REFERNCES:

www.google.com

ACKNOWLEDGEMENT

“We would like to express our thanks to the people who have helped us the most throughout our project. We are grateful to our Professor Dr. Yogita Mane for constant support for the project. A special thanks to our group member and our colleague who helped us out in completing the project, where we exchanged our own interesting ideas, thoughts and made this possible to complete our project with all accurate information. We wish to thank our parents for their personal support or attention who inspired us to go our own way.”