



**GROUP ASSIGNMENT**  
**BSD2213 DATA SCIENCE PROGRAMMING**  
**SEMESTER I 2023/2024**

NAME	STUDENT ID
ANG MEI YING	SD22061
NURUL ALIS BINTI YUSRI	SD22045
HAWA HUMAIRA BINTI HAMUZAN	SD22043
SITI NUR FARAHANA BINTI SAPUAN	SD21020

## **Table of Content**

No.	Content	Page
1	Introduction	3
2	Why this project?	4
3	How can this project be extended?	5
4	Source Code	6
5	Screenshot for each activity in the GUI	15

## **Introduction**

Waterparks are increasingly popular destinations for adults and families in the entertainment and recreation business. Splash Mania Waterpark is a popular spot for fun, especially for those who love thrills and water adventures. However, as Splash Mania's popularity grows, a closer look at its current state exposes areas that may be improved. Several issues and challenges are prevalent at Splash Mania Waterpark right now, which have an effect on the entire experience for visitors.

First and foremost, a major issue is the inconsistent visitor flow experienced by the waterpark, particularly during peak hours like summer vacations and public holidays. This results in overcrowding and prolonged wait times for attractions. This creates an uneven distribution of visitors throughout the day, affecting the park's operational efficiency and the enjoyment of visitors. Furthermore, the lack of a strong data-driven strategy makes it more difficult for the park to decide on popular attractions, peak hours, and visitor preferences. The waterpark's ability to optimize operations and customize services to the needs of visitors may be limited by this lack of data analytics.

Aside from that, the second challenge that exists with the existing situation is the traditional on-site ticketing system, which frequently causes lengthy lines during peak hours. This reduces the waterpark's operational efficiency in addition to making visitors frustrated. These lines have developed into a bottleneck due to Splash Mania's growing popularity, which is preventing a smooth visitor flow and may eventually reduce the amount of people who may visit the park.

Moreover, the ongoing COVID-19 pandemic has introduced new challenges related to safety and capacity restrictions. The number of guests that can enter the waterpark at any given moment has been limited as a result of the necessity for social distancing measures. It could be difficult for guests to reserve their spots in advance without a simplified reservation process, which could leave them disappointed and dissatisfied.

In summary, the challenges and problems include long queues, inefficient operations, traditional on-site ticketing system and limited capacity due to COVID-19 restrictions. Recognizing these challenges, the introduction of an online booking system emerges as a strategic solution to enhance visitor experience, streamline operations, and address the evolving needs of Splash Mania's diverse audience. Through the development of a strong online booking system, this project seeks to innovate, assess the current situation, and optimize operational factors. The aim is to preserve the thrill and safety that characterize Splash Mania Waterpark while providing every guest with a streamlined and secure booking experience by embracing data science and utilizing technology.

## **Why this project?**

In this project, we are trying to make an online booking system for Splash Mania Waterpark. The advantages of online booking system are that it provides a hassle-free and convenient way for visitors to reserve their spots at Splash Mania. Instead of waiting in long queues, visitors can book their tickets from the comfort of their homes or while on the road. By allowing visitors to book online, the system saves their time and effort. Visitor also don't have to physically visit the waterpark or stand in line, optimizing their overall experience and allowing them to make the most of their time at Splash Mania. Furthermore, the online system offers flexibility in choosing the date or time slot to visit the waterpark. This flexibility enables visitors to plan their trips according to their preferences and available time.

Besides that, due to Covid-19 pandemic, an online booking system can provide a better overall experience for visitor and follows health guidelines. Visitor can easily reserve their spot, plan activities, and enjoy their fun time while being safe. Additionally, with limited spots because of the pandemic, we ensured that the Splash Mania Waterpark Booking System provided real-time updates on ticket availability and time slots. This feature helps visitors make well informed decisions, ensuring they secure their preferred times and lowering the possibility of disappointment from capacity limitation.

The primary scope of project is to streamline the ticketing procedure, cut down on wait times, and make sure that visitors move through the waterpark more smoothly. This not only improves the overall experience but also encourages repeat visits. Secondly, the project aims to develop an engaging platform that make visitors can easily plan their itineraries. Thirdly, record-keeping within the booking system provides insights into peak booking times, allowing the waterpark to efficiently allocate resources and staff during high-demand periods, optimizing overall operational efficiency.

The objective of project is to efficiently manage capacity by providing real-time information on available tickets and time slots. This feature ensures that visitors can make well-informed decisions and reserve their preferred times without having to worry about being disappointed. The project's second objective is to enhance visitors' overall experience by offering a user-friendly interface, tailored suggestions, and easy online booking system navigation. The goal is to make the entire process enjoyable and stress-free. Thirdly, our objective is to provide visitors the freedom to select the day and time that works best for them. Convenience is increased by this flexibility, which lets visitors schedule their itineraries around their personal interests and free time.

## **How can this project be extended?**

To further enhance the Splash Mania online booking system, several features can be added to provide additional convenience and functionality to visitors. First, online booking systems can integrate multiple payment methods. By offering options such as credit or debit cards, e-wallets, online banking and other secure payment options, users will have the flexibility to pay in the way that suits them best. This inclusivity ensures a seamless and adaptable booking experience that can accommodate a wider range of users with different payment preferences. This improvement not only streamlines the booking process, but also helps provide a more user-friendly and inclusive platform for everyone planning to visit Splash Mania.

Besides that, the project also can be extended by integrate an interactive park map within the online booking system. This map can assist users in selecting specific attractions, facilities, or areas within Splash Mania. The visual representation adds a layer of convenience, helping visitors plan their itineraries more effectively and improving their overall experience.

Next, another significant improvement to the online booking system is the introduction of booking cancellation functionality. This service allows visitors to cancel their reservations immediately through the system. This new addition is valuable in providing users with greater flexibility and control over their plans. To ensure that users who may need to reschedule their visit have a worry-free experience, cancellation capability should be made as simple and intuitive as possible. Clear and transparent guidelines will also be included, possibly accompanied by a clear refund policy, to give users a clear understanding of the cancellation process. This not only increases transparency but it also gives users more confidence because they know they can readily change their plans as needed. Ultimately, the introduction of strong cancellation features contributes to the development of an online reservation system that is more user-centric and adapts to changing needs and preferences of Splash Mania visitors.

In addition, we might also extend the booking system to enable visitors to purchase annual passes. This feature can be a cost-effective solution for frequent visitors, providing them with unlimited access to Splash Mania for an entire year. In order to increase customer loyalty and promote return visits, the system should provide a simple and easy way for visitors to purchase, renew, or upgrade their annual passes.

In conclusion, by implementing these enhancements, the Splash Mania online booking system is poised to deliver a more inclusive, user-friendly, and flexible experience for visitors.

## Source Code

```
from tkinter import *
from tkinter import messagebox
from tkcalendar import *
from datetime import datetime
import random
import string

class FirstPage(Frame):
    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        self.configure(bg='white')
        self.img= PhotoImage(file='logo.png')
        self.resized_img = self.img.subsample(2)
        Label(self, image=self.resized_img, bg='white').place(x=50, y=100)

        frame = Frame(self, width=400, height=350, bg='sky blue')
        frame.place(x=450, y=70)

        heading = Label(frame, text='Login', fg='white', bg='sky blue',
font=('Arial', 23, 'bold'))
        heading.place(x=150, y=5)

        usernameEntry = Entry(self, width=30, font=('Arial', 12, 'bold'))
        usernameEntry.place(x=505, y=150)
        usernameEntry.insert(0, 'Username')
        usernameEntry.bind('<FocusIn>', self.username_enter)

        passwordEntry = Entry(self, width=30, font=('Arial', 12, 'bold'))
        passwordEntry.place(x=505, y=220)
        passwordEntry.insert(0, 'Password')
        passwordEntry.bind('<FocusIn>', self.userpassword_enter)

        self.closeEye = PhotoImage(file='closeeyes.png')
        self.resized_closeEye = self.closeEye.subsample(10)
        eyeButton = Button(self, image=self.resized_closeEye,
cursor='hand2')
        eyeButton.place(x=800, y=217)

        forgetButton = Button(self, text='Forgot Password?', font=('Arial',
10, 'bold'), bd=0, bg='sky blue',
                                cursor='hand2', activebackground='steel
blue')
        forgetButton.place(x=715, y=260)

        def login():
            try:
                with open('credential.txt', 'r') as f:
                    info = f.readlines()
                    i = 0
                    for e in info:
                        u, p = e.split(',')
                        if u.strip() == usernameEntry.get() and p.strip()
== passwordEntry.get():
                            controller.show_frame(ThirdPage)
                            i = 1
                            break
                    if i == 0:
                        messagebox.showinfo('Error', 'Please provide
```

```

correct username and password!')
    except:
        messagebox.showinfo('Error', 'Please provide correct
username and password!')

    loginButton = Button(self, text='Login', font=('Arial', 16,
'bold'), command=login)
    loginButton.place(x=500, y=300)

    sign_up_Label = Label(self, text='Dont have an account?',
font=('Arial', 10, 'bold'), bg='sky blue')
    sign_up_Label.place(x=500, y=360)

    def register():
        controller.show_frame(SecondPage)

    newaccButton = Button(self, text='Create New Account',
font=('Arial', 9, 'bold underline'), bd=0,
                        bg='sky blue', cursor='hand2',
activebackground='steel blue', command=register)
    newaccButton.place(x=660, y=360)

    def username_enter(self, event):
        if event.widget.get() == 'Username':
            event.widget.delete(0, END)

    def userpassword_enter(self, event):
        if event.widget.get() == 'Password':
            event.widget.delete(0, END)
            event.widget.config(show='*')

class SecondPage(Frame):
    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        self.configure(bg='white')
        self.head=Label(self,text='Create A New
Account !',font=('Arial',20,'bold'),bg='white')
        self.head.pack(pady=70)
        l1 = Label(self, text='Username', font=('Arial', 10))
        l1.place(x=300, y=150)
        t1 = Entry(self, width=30, bd=3)
        t1.place(x=400, y=150)

        l2 = Label(self, text='Password', font=('Arial', 10))
        l2.place(x=300, y=200)
        t2 = Entry(self, width=30, show='*', bd=3)
        t2.place(x=400, y=200)

        l3 = Label(self, text='Confirm Password', font=('Arial', 10))
        l3.place(x=270, y=250)
        t3 = Entry(self, width=30, show='*', bd=3)
        t3.place(x=400, y=250)

    def check():
        if t1.get() != '' or t2.get() != '' or t3.get() != '':
            if t2.get() == t3.get():
                with open('credential.txt', 'a') as f:
                    f.write(t1.get() + ',' + t2.get() + '\n')

```

```

        messagebox.showinfo('Welcome', 'you are registered
successfully!!')
    else:
        messagebox.showinfo('Error', "Your password didn't get
match!!")
    else:
        messagebox.showinfo('Error', "Please fill the complete
field!!")

    submit = Button(self, text='Submit', font=('Arial', 15), bg='sky
blue', cursor='hand2', command=check)
    submit.place(x=420, y=320)

    backlog= Button(self, text='Back To Login', font=('Arial', 9, 'bold
underline'), bd=0, bg='white', cursor='hand2',
        command=lambda: controller.show_frame(FirstPage))
    backlog.place(x=520, y=330)

class ThirdPage(Frame):
    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        self.configure(bg='white')
        self.homepage = Label(self, text="Welcome to Splash Mania Water
Park !", font=('Arial', 20, 'bold'), bg='white')
        self.homepage.pack(pady=20)
        self.hpimg = PhotoImage(file='waterpark.png')
        self.resized_hpimg = self.hpimg.subsample(2)
        Label(self, image=self.resized_hpimg).place(x=50, y=60)
        self.intro=Label(self, text="""
Where nature meets fun, and where you can
spend the entire day splashing around under
the sun! Experience the thrills and spills of a
one-of-a-kind waterpark n SplashMania.Home
to 39 exhilarating slides and attractions that
cater to families, students, young working
adults and thrill seekers of all kinds, this
waterpark is located in Gamuda Cove,
Selangor.
""", font=('Arial', 15), bg='white', justify='left')
        self.intro.place(x=408, y=60)
        self.intro = Label(self, text='MYR80.00 - MYR125.00', font=('Arial',
20, 'bold'), bg='white', justify='left')
        self.intro.place(x=460, y=330)

        bookingButton=Button(self, text='Book Now',
font=('Arial', 17), cursor='hand2',
            command=lambda:
controller.show_frame(FourthPage))
        bookingButton.place(x=460, y=390)

class FourthPage(Frame):
    visitor = {}
    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        self.configure(bg='white')
        self.selectslot = Label(self, text='Select Slot To Book',
font=('Arial', 15), bg='white')

```



```

        self.selectslot.pack(pady=10)

        self.selectdate = Label(self, text='Select booking date:',
font=('Arial', 12), bg='white')
        self.selectdate.place(x=40, y=70)
        self.selectdateEntry = Entry(self, bg='white', font=('Arial', 11))
        self.selectdateEntry.place(x=200, y=70, width=150)
        self.pick_date_button = Button(self, text='Pick Date',
command=self.pick_date)
        self.pick_date_button.pack(padx=20,pady=17)

        self.child_ticket_label = Label(self, text='Number of Child (90cm-
120cm) ', font=('Arial', 12),bg='white')
        self.child_ticket_label.place(x=40,y=120)
        self.child_pricelabel=Label(self,text='MYR',font=('Arial',
12),bg='white')
        self.child_pricelabel.place(x=450, y=120)
        self.child_ticket_price_var = StringVar()
        self.child_ticket_price_label = Label(self,
textvariable=self.child_ticket_price_var, font=('Arial', 12),
bg='white')
        self.child_ticket_price_label.place(x=490, y=120)
        self.child_spinner = Spinbox(self, from_=0,
to=10,width=5,command=self.calculate_total)
        self.child_spinner.place(x=580, y=120)

        self.adult_ticket_label = Label(self, text='Number of Adult (above
120cm) ', font=('Arial', 12), bg='white')
        self.adult_ticket_label.place(x=40, y=170)
        self.adult_pricelabel = Label(self, text='MYR', font=('Arial', 12),
bg='white')
        self.adult_pricelabel.place(x=450, y=170)
        self.adult_ticket_price_var = StringVar()
        self.adult_ticket_price_label = Label(self,
textvariable=self.adult_ticket_price_var, font=('Arial', 12),
bg='white')
        self.adult_ticket_price_label.place(x=490, y=170)
        self.adult_spinner = Spinbox(self, from_=0,
to=10,width=5,command=self.calculate_total)
        self.adult_spinner.place(x=580, y=170)

        self.seniorcitizen_ticket_label = Label(self, text='Number of
Senior Citizen (Age 60 years and Above) ',
font=('Arial', 12),
bg='white')
        self.seniorcitizen_ticket_label.place(x=40, y=220)
        self.sc_pricelabel = Label(self, text='MYR', font=('Arial', 12),
bg='white')
        self.sc_pricelabel.place(x=450, y=220)
        self.seniorcitizen_ticket_price_var = StringVar()
        self.seniorcitizen_ticket_price_label = Label(self,
textvariable=self.seniorcitizen_ticket_price_var,
font=('Arial',
12),bg='white')
        self.seniorcitizen_ticket_price_label.place(x=490, y=220)
        self.seniorcitizen_spinner = Spinbox(self, from_=0,
to=10,width=5,command=self.calculate_total)
        self.seniorcitizen_spinner.place(x=580, y=220)

        self.total_ticket_label = Label(self, text='Total Tickets:',
font=('Arial', 12), bg='white')

```

```

        self.total_ticket_label.place(x=40, y=270)
        self.total_ticket_count_var = StringVar()
        self.total_ticket_count_label = Label(self,
textvariable=self.total_ticket_count_var, font=('Arial', 12),
                                                    bg='white')
        self.total_ticket_count_label.place(x=150, y=270)

        self.total_price_label = Label(self, text='Total Price:',
font=('Arial', 12), bg='white')
        self.total_price_label.place(x=40, y=320)
        self.total_ticket_price_var = StringVar()
        self.total_ticket_price_label = Label(self,
textvariable=self.total_ticket_price_var, font=('Arial', 12),
                                                    bg='white')
        self.total_ticket_price_label.place(x=150, y=320)

        buynowButton = Button(self, text='Buy Now', font=('Arial', 17),
cursor='hand2',
                                command=lambda: self.buy_now(controller))
        buynowButton.place(x=40, y=400)

    def pick_date(self):
        date_window = Toplevel(self)
        date_window.title('Select Your Booking Date')
        date_window.geometry('250x220+590+370')

        cal = Calendar(date_window, selectmode='day', date_pattern='yyyy-
mm-dd')
        cal.pack(pady=10)

        set_button = Button(date_window, text='Set', command=lambda:
self.set_date(cal))
        set_button.place(x=110, y=195)

    def set_date(self, cal):
        selected_date = cal.get_date()
        if selected_date in FourthPage.visitor and
FourthPage.visitor[selected_date] >= 500:
            messagebox.showinfo('Error', 'Maximum number of visitors reached
for this date. Please choose another date.')
            return
        self.selectdateEntry.config(state=NORMAL)
        self.selectdateEntry.delete(0, END)
        self.selectdateEntry.insert(0, selected_date)
        self.selectdateEntry.config(state='readonly')
        if selected_date in FourthPage.visitor:
            FourthPage.visitor[selected_date] += 1
        else:
            FourthPage.visitor[selected_date] = 1
        date_obj = datetime.strptime(selected_date, '%Y-%m-%d')
        weekday = date_obj.weekday()

        if weekday < 4:
            child_ticket_price = '80.00'
            adult_ticket_price = '110.00'
            seniorcitizen_ticket_price='80.00'
        else:
            child_ticket_price = '95.00'
            adult_ticket_price = '125.00'
            seniorcitizen_ticket_price = '95.00'
        self.child_ticket_price_var.set(child_ticket_price)

```

```

        self.adult_ticket_price_var.set(adult_ticket_price)
        self.seniorcitizen_ticket_price_var.set(seniorcitizen_ticket_price)

    def calculate_total(self):
        child_count = int(self.child_spinner.get())
        adult_count = int(self.adult_spinner.get())
        seniorcitizen_count = int(self.seniorcitizen_spinner.get())

        total_count = child_count + adult_count + seniorcitizen_count
        self.total_ticket_count_var.set(str(total_count))

        child_ticket_price = float(self.child_ticket_price_var.get())
        adult_ticket_price = float(self.adult_ticket_price_var.get())
        seniorcitizen_ticket_price =
float(self.seniorcitizen_ticket_price_var.get())

        total_price = (child_count * child_ticket_price +
                        adult_count * adult_ticket_price +
                        seniorcitizen_count * seniorcitizen_ticket_price)

        self.total_ticket_price_var.set(f'MYR {total_price:.2f}')

    def get_total_count(self):
        child_count = int(self.child_spinner.get())
        adult_count = int(self.adult_spinner.get())
        seniorcitizen_count = int(self.seniorcitizen_spinner.get())
        return child_count + adult_count + seniorcitizen_count

    def buy_now(self, controller):
        selected_date = self.selectdateEntry.get()

        if not selected_date:
            messagebox.showinfo('Error', 'Please select a booking date
before proceeding.')
            return
        total_count = self.get_total_count()
        if total_count == 0:
            messagebox.showinfo('Error', 'Please select at least one ticket
before proceeding.')
            return
        if selected_date in FourthPage.visitor and
FourthPage.visitor[selected_date] >= 500:
            messagebox.showinfo('Error', 'Maximum number of visitors reached
for this date. Please choose another date.')
            return
        controller.show_frame(FifthPage)

class FifthPage(Frame):
    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        self.configure(bg='white')
        self.controller = controller
        booking_detail_label = Label(self, text='Booking Detail',
font=('Arial', 15, 'bold'), bg='white')
        booking_detail_label.place(x=370, y=30)
        self.name_label = Label(self, text='Name:', font=('Arial', 12),
bg='white')
        self.name_label.place(x=350, y=100)
        self.name_entry = Entry(self, width=20, font=('Arial', 12), bd=2)
        self.name_entry.place(x=420, y=100)

```

```

        self.ic_label = Label(self, text="IC No:", font=('Arial', 12),
bg='white')
        self.ic_label.place(x=350, y=150)
        self.ic_entry = Entry(self, width=20, font=('Arial', 12), bd=2)
        self.ic_entry.place(x=420, y=150)

        self.phone_label = Label(self, text="Contact No:", font=('Arial',
12), bg='white')
        self.phone_label.place(x=310, y=200)
        self.phone_entry = Entry(self, width=20, font=('Arial', 12), bd=2)
        self.phone_entry.place(x=420, y=200)

        self.email_label = Label(self, text="Email:", font=('Arial', 12),
bg='white')
        self.email_label.place(x=350, y=250)
        self.email_entry = Entry(self, width=20, font=('Arial', 12), bd=2)
        self.email_entry.place(x=420, y=250)

        self.resident_status_label = Label(self, text="Resident Satus:",
font=('Arial', 12), bg='white')
        self.resident_status_label.place(x=300, y=300)

        options = ["Resident", "Non-Resident"]
        self.resident_status_var = StringVar()
        self.resident_status_var.set(options[0])
        self.resident_status_menu = OptionMenu(self,
self.resident_status_var, *options)
        self.resident_status_menu.place(x=420, y=300)

        nextButton = Button(self, text='Next', font=('Arial', 17),
cursor='hand2',
                                command=self.nextPage)
        nextButton.place(x=400, y=400)

        def nextPage(self):
            if not self.name_entry.get() or not self.ic_entry.get() or not
self.phone_entry.get() or not self.email_entry.get():
                messagebox.showwarning("Missing Information", "Please complete
your booking detail.")

            else:
                booking_info = {
                    'Name': self.name_entry.get(),
                    'IC': self.ic_entry.get(),
                    'Contact No': self.phone_entry.get(),
                    'Email': self.email_entry.get(),
                    'Resident Status': self.resident_status_var.get()}
                self.controller.show_frame(SixthPage, booking_info)

class SixthPage(Frame):
    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        self.configure(bg='white')
        bk_label = Label(self, text='Booking Confirmation', font=('Arial',
18, 'bold underline'), bg='white')
        bk_label.place(x=300, y=10)
        check_label = Label(self, text='Please confirm your booking
details.', font=('Arial', 12, 'bold'), bg='white')
        check_label.place(x=300, y=60)

```

```

        check2_label = Label(self, text='Once the booking is confirmed,
booking information will be send to your email.', font=('Arial', 12,
'bold'), bg='white')
        check2_label.place(x=150, y=90)

        checkoutButton = Button(self, text='Checkout', font=('Arial', 12),
cursor='hand2',
                                command=lambda:
controller.show_frame(SeventhPage))
        checkoutButton.place(x=450, y=400)
        backButton = Button(self, text='Back', font=('Arial', 12),
cursor='hand2',
                                command=lambda:
controller.show_frame(FifthPage))
        backButton.place(x=350, y=400)
        self.name_label = Label(self, text="Name:", bg='white',
font=('Arial', 12))
        self.name_label.place(x=370, y=130)

        self.ic_label = Label(self, text="IC:", bg='white', font=('Arial',
12))
        self.ic_label.place(x=390, y=180)

        self.phone_label = Label(self, text="Contact No:", bg='white',
font=('Arial', 12))
        self.phone_label .place(x=330, y=230)

        self.email_label = Label(self, text="Email:", bg='white',
font=('Arial', 12))
        self.email_label.place(x=370, y=280)

        self.resident_status_label = Label(self, text="Resident Satus:",
font=('Arial', 12), bg='white')
        self.resident_status_label.place(x=340, y=330)

    def display_bk_detail (self, booking_info):
        self.name_label.config(text=f"Name: {booking_info.get('Name',
'')}")
        self.ic_label.config(text=f"IC: {booking_info.get('IC', '')}")
        self.phone_label.config(text=f"Contact No:
{booking_info.get('Contact No', '')}")
        self.email_label.config(text=f"Email: {booking_info.get('Email',
'')}")
        self.resident_status_label.config(text=f"Resident Status:
{booking_info.get('Resident Status', '')}")

class SeventhPage(Frame):
    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        self.configure(bg='white')
        self.ticking = PhotoImage(file='tick.png')
        self.resized_ticking = self.ticking.subsample(5)
        Label(self, image=self.resized_ticking).place(x=370, y=10)

        bkconf_label = Label(self, text="Booking Confirmed !",
font=('Arial', 22, 'bold'), bg='white')
        bkconf_label.place(x=320, y=200)

        booking_reference = self.generate_booking_reference()

        bkref_label = Label(self, text=(f"Your Booking

```

```

Reference:{booking_reference}"), font=('Arial',17, 'bold'), bg='white')
    bkref_label.place(x=260, y=250)

    instr_label = Label(self, text="""
    Please make your payment at entrance counter by showing your
booking reference.
    Please contact 03 9322 1811 (customer service) if you
    wish to cancel your booking.Thanks you! """,
    font=('Arial', 12), bg='white', justify='center')
    instr_label.place(x=150, y=280)

    logoutButton = Button(self, text='Log out', font=('Arial', 12),
command=exit)
    logoutButton.place(x=420, y=380)

    def generate_booking_reference(self):
        random_part = ''.join(random.choices(string.ascii_uppercase +
string.digits, k=6))
        return f"SMBR{random_part}"

class Application(Tk):
    def __init__(self, *args, **kwargs):
        Tk.__init__(self, *args, **kwargs)
        self.title('Splash Mania Water Park')
        window = Frame(self)
        window.pack()

        window.grid_rowconfigure(0, minsize = 500)
        window.grid_columnconfigure(0, minsize=900)

        self.frames = {}
        for F in (FirstPage, SecondPage, ThirdPage, FourthPage,
FifthPage,SixthPage,SeventhPage):
            frame = F(window, self)
            self.frames[F] = frame
            frame.grid(row = 0, column = 0, sticky='nsew')

        self.show_frame(FirstPage)

    def show_frame(self, page, *args, **kwargs):
        frame = self.frames[page]

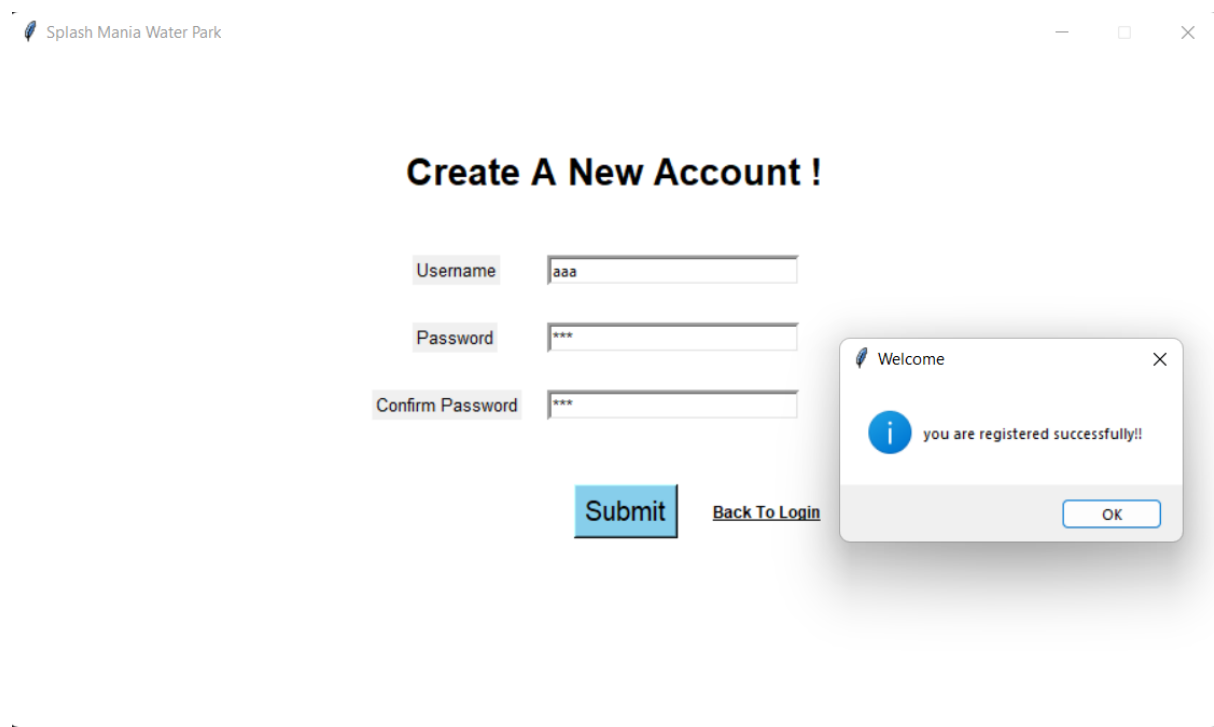
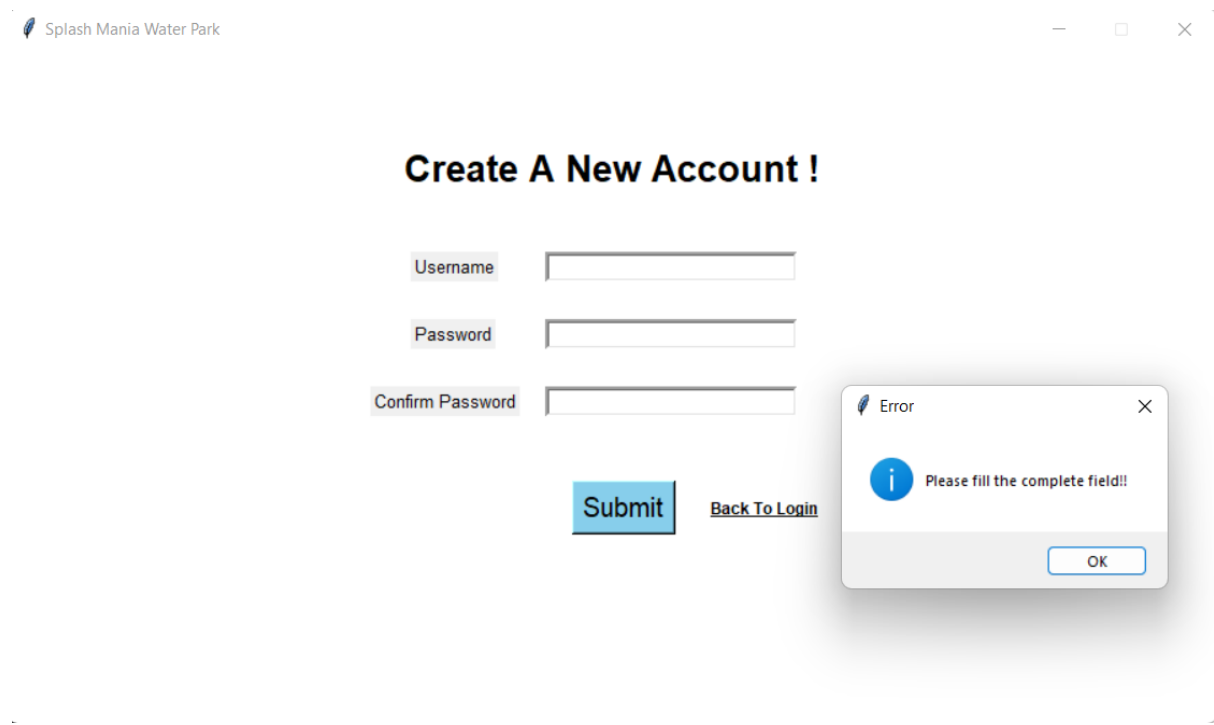
        if hasattr(frame, 'display_bk_detail'):
            frame.display_bk_detail(*args, **kwargs)
            frame.tkraise()

app = Application()
app.resizable(False,False)
app.mainloop()

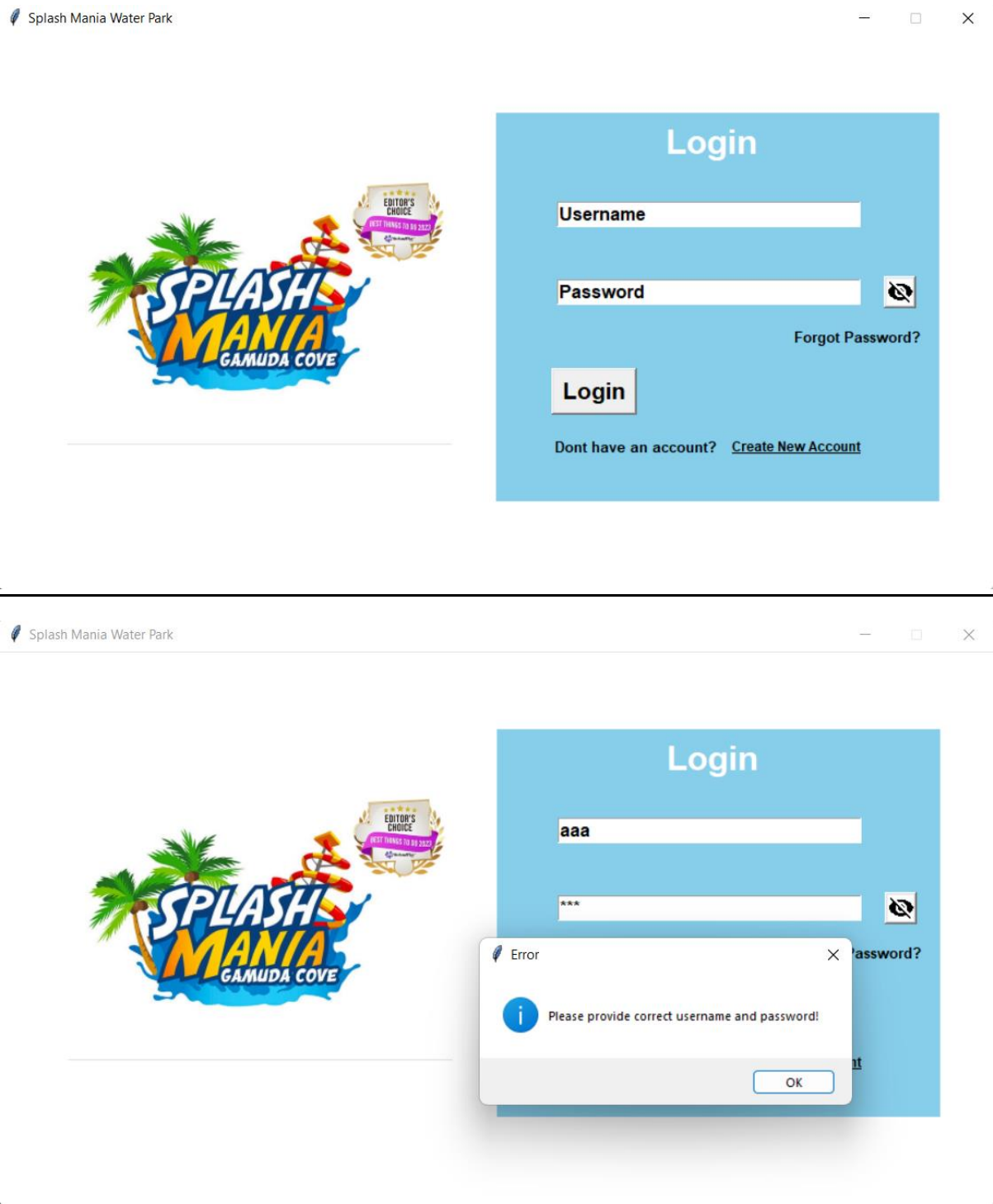
```

## Screenshot for each activity in the GUI

-Users need to create an account before login and user must ensure that they complete fill in for each field before click on 'submit' button.




-After creating an account, users need to make sure that they key in the correct username and password to login.






-An introduction about Splash Mania Waterpark to attract users' attention. User can click the 'Book Now' button to make their reservation.

 Splash Mania Water Park

## Welcome to Splash Mania Waterpark !




Where nature meets fun, and where you can spend the entire day splashing around under the sun! Experience the thrills and spills of a one-of-a-kind waterpark n SplashMania.Home to 39 exhilarating slides and attractions that cater to families, students, young working adults and thril seekers of all kinds, this waterpark is located in Gamuda Cove, Selangor.

**MYR80.00 - MYR125.00**

Book Now

-Users need to click on 'Pick Date' button to choose their booking date. Prices are shown based on the selected date, whether it is a weekday or a weekend. Makes sure that user selects a booking date and at least one ticket before clicking the 'Buy Now' button, otherwise a message box will be displayed if fail to do this. Additionally, the booking system also calculates the total number of tickets and the total price. If the total number of visitors per day reaches a maximum of 500, the day will be blocked.

 Splash Mania Water Park

—□×

### Select Slot To Book

Select booking date:

Pick Date

Number of Child (90cm-120cm)

MYR

0

Number of Adult (above 120cm)

MYR

0

Number of Senior Citizen (Age 60 years and Above)


MYR

0

Total Tickets:

Total Price:

Buy Now

 Splash Mania Water Park

—□×

### Select Slot To Book

Select booking date: 2024-01-09

Pick Date

Number of Child (90cm-120cm)

MYR 80.00

0

Number of Adult (above 120cm)

MYR 110.00

0

Number of Senior Citizen (Age 60 years and Above)

MYR 80.00

0

Total Tickets:

Total Price:

Buy Now

Select Your B...

January

2024

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	1	2	3	4	5	6	7
2	8	9	10	11	12	13	14
3	15	16	17	18	19	20	21
4	22	23	24	25	26	27	28
5	29	30	31	1	2	3	4
6	5	6	7	8	9	10	11

Set

## Select Slot To Book

Select booking date: 2024-01-13

Pick Date

Number of Child (90cm-120cm)

MYR 95.00

0

Number of Adult (above 120cm)

MYR 125.00

0

Number of Senior Citizen (Age 60 years and Above)

MYR 95.00

0

Total Tickets:

Total Price:

Buy Now

Select Your B...

January

2024

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	1	2	3	4	5	6
2	8	9	10	11	12	13
3	15	16	17	18	19	20
4	22	23	24	25	26	27
5	29	30	31	1	2	3
6	5	6	7	8	9	10

Set

## Select Slot To Book

Select booking date:

Pick Date

Number of Child (90cm-120cm)

MYR

0

Number of Adult (above 120cm)

MYR

0

Number of Senior Citizen (Age 60 years and Above)

M

Total Tickets:

Total Price:

Buy Now

Error

Please select a booking date before proceeding.

OK

### Select Slot To Book

Select booking date: 2024-01-09

Pick Date

Number of Child (90cm-120cm)

MYR 80.00

0

Number of Adult (above 120cm)

MYR 110.00

0

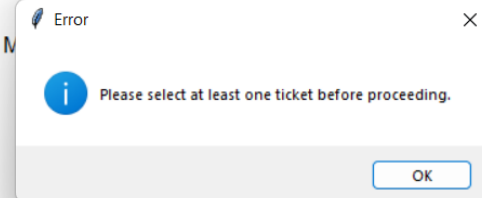
Number of Senior Citizen (Age 60 years and Above)

MYR

Total Tickets:

Total Price:

Buy Now



### Select Slot To Book

Select booking date: 2024-01-09

Pick Date

Number of Child (90cm-120cm)

MYR 80.00

3

Number of Adult (above 120cm)

MYR 110.00

5

Number of Senior Citizen (Age 60 years and Above)

MYR 80.00

4

Total Tickets: 12

Total Price: MYR 1110.00

Buy Now

-Next, users need to fill in their booking details. A message box will be shown if user didn't complete fill in for each field.

Splash Mania Water Park

— □ ×

**Booking Detail**

Name:

IC No:

Contact No:

Email:

Resident Satus:

Resident ▾

Next

Splash Mania Water Park

— □ ×

**Booking Detail**

Name:

IC No:

Contact No:

Email:

Resident Satus:

Resident ▾

Next

Missing Information

×

⚠ Please complete your booking detail.

OK

### Booking Detail

Name:

IC No:

Contact No:

Email:

Resident Satus:

- Since the booking information will be sent via email, users are requested to check and confirm their booking detail. User can click 'Back' button to go to the previous page to change their booking details or click on 'Checkout' button once confirming their booking details.

### Booking Confirmation

Please confirm your booking details.

Once the booking is confirmed, booking information will be send to your email.

Name: Rachel

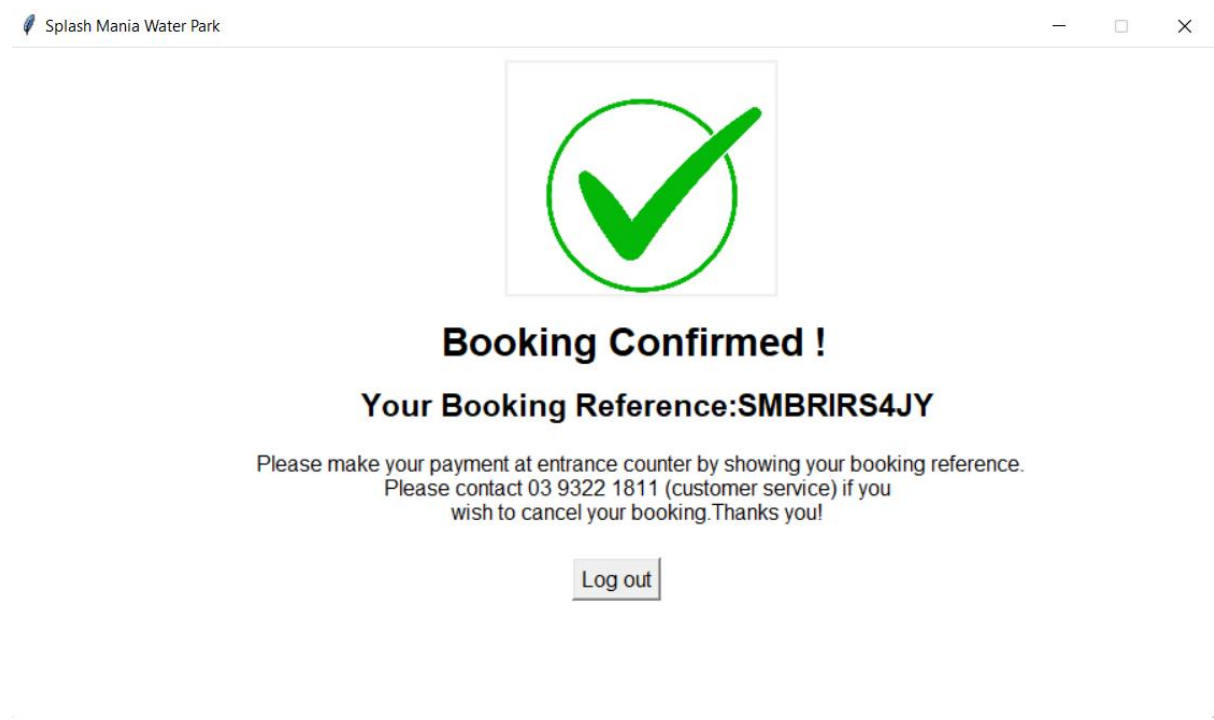
IC: 021005104231

Contact No: 0164523851

Email: rachel0510@gmail.com

Resident Status: Resident

-Once users are confirmed with their booking, a booking references will be generated. Since the booking references is unique and different for each user and transaction, therefore user can show their booking references at entrance counter of waterpark to make their payment and obtain their ticket. Lastly, user can click the 'Log out' button to exit the system.



## MARKING SCHEME

CLO	Description	PLO mapping	Percentage	Marks
CLO2	Use appropriate Python programming technique to solve problem.	PLO2: Cognitive Skills and Functional work skills with focus on Numeracy skills C3: Application	5%	10

LEVEL OF ACHIEVEMENT					
0	1 Inadequate	2 Emerging	3 Developing	4 Good	5 Excellent

ELEMENTS	WEIGHTAGE	SCORE
<b>Combination of appropriate controls and layout manager:</b> <ul style="list-style-type: none"> <li>Input controls such as buttons, toggles, checkboxes etc.</li> <li>Navigation controls such as pull-down menu.</li> <li>Information components eg. message boxes etc.</li> <li>Tkinter geometry manager (place/pack/grid manager).</li> </ul>	1	
<b>Task execution by each controls:</b> <ul style="list-style-type: none"> <li>Each control is labelled using short and precise words representing the task.</li> <li>The task for each controls is specified and written neatly.</li> <li>The task for each control executed correctly and smoothly.</li> </ul>	1	
<b>TOTAL</b>		



CLO	Description	PLO mapping	Percentage	Marks
CLO3	Construct and run program.	PLO3: Functional work skills with focus on Practical, and Digital skills P4: Mechanism	15%	30

CRITERIA	LEVEL OF ACHIEVEMENT						WEIGHTAGE	SCORE
	0	1 Inadequate	2 Emerging	3 Developing	4 Good	5 Excellent		
<b>Theory/ Knowledge</b>	No theoretical knowledge is observed.	Very little knowledge provided or information is incorrect.	Some knowledge or information is provided but missing all major points.	Some knowledge or information is provided but still missing some major points.	Good knowledge is observed, missing some minor points.	Excellent knowledge is observed; provides all necessary background principles.	1	
<b>Assembly</b>	Fail to demonstrate the given task.	Partly demonstrate the given task with errors.	Partly demonstrate the given task with wrong output.	Partly demonstrate the given task correctly.	Fully demonstrate the given task with some wrong output.	Demonstrate the given task correctly and perfectly.	2	
<b>Technique used / Effectiveness</b>	Fail to demonstrate the given task.	Demonstrate inappropriate techniques.	Partly correct techniques demonstrated.	Demonstrated technique is correct but not effective or efficient.	Demonstrated technique is partly effective and efficient.	Demonstrated technique is effective and efficient.	2	
<b>GUI</b>	Not submitting GUI.	The GUI presented was taken from the other sources with no modifications. The GUI presented was not effective in debugging the output with a lot of errors and displayed for an inappropriate time.	The GUI presented was modified from the other sources with minimal modifications. Shows less effective debugging on the output with several errors and displayed for less appropriate time.	The GUI presented was modified well from the other sources. Shows effective debugging on the output with no error and displayed for an appropriate time.	The GUI presented was modified very well from the other sources. Shows effective debugging on the output with no error and displayed for an appropriate time.	The GUI presented was originally developed. Shows effective debugging on the output with no error and displayed for an appropriate time.	1	

CLO	Description	PLO mapping	Percentage	Marks
CLO4	Work collaboratively to solve assigned task.	PLO4: Functional work skills with focus on Interpersonal skills A3: Valuing	5%	10

CRITERIA	LEVEL OF ACHIEVEMENT						WEIGHTAGE	SCORE
	0	1 Inadequate	2 Emerging	3 Developing	4 Good	5 Excellent		
<b>Foster Good Relationship</b>	Show no good relationships and unable to work together effectively with other group members towards goal achievement.	No clear evidence of ability to foster good relationships and work together effectively with other group members towards goal achievement.	Able to foster relationship and work together with other group members towards goal achievement but with limited effect and require improvements.	Able to foster relationship and work together with other group members towards goal achievement with some effect(s) and require minor improvements.	Able to foster good relationship and work together with other group members towards goal achievement.	High ability to foster good relationship and work together effectively with other group members towards goal achievement.	1	
<b>Alternate Roles</b>	Show no ability to assume alternate roles as a group leader and group members.	No clear evidence of ability to assume alternate roles as a group leader and group members demonstrated in practice.	Attempt to demonstrate in practice the ability to alternate roles as a group leader and group members but with limited effect and require improvements.	Able to demonstrate in practice the ability to assume alternate roles as a group leader and group members with some effect(s) and require minor improvements.	Able to demonstrate in practice the ability to assume alternate roles as a group leader and a group member to achieve the same goal.	Show clear evidence to assume alternate roles as a group leader and a group member demonstrated in practice.	1	

CLO	Description	PLO mapping	Percentage	Marks
CLO5	Demonstrate innovative ideas in developing a graphical user interface.	PLO8: Entrepreneurial skills A3: Valuing	5%	10

CRITERIA	LEVEL OF ACHIEVEMENT						WEIGHTAGE	SCORE
	0	1 Inadequate	2 Emerging	3 Developing	4 Good	5 Excellent		
<b>Analyzing an existing situation and identifying areas for improvement</b>	Not providing any analysis of situation and areas for improvement were not identified.	The analysis of the situation was very limited and areas for improvement were not identified.	The analysis of the situation was limited and areas for improvement were not identified.	The analysis of the situation was appropriate but the identification of areas for improvement was limited.	The situation was appropriately analyzed and the identification of areas for improvement was completed.	The analysis of the situation and the identification of areas for improvement was completed and increases over time.	1	
<b>Creativity/ Innovative ideas</b>	Not presenting any GUI.	GUI presented contains lack of significance ideas, no innovative values, lack of creativity and not user friendly.	GUI presented contains lack of significance ideas, no innovative values, creative enough (catchy apps name & attractive) and user friendly.	GUI presented contains lack of significance ideas, but still have innovative values, creative enough (catchy apps name & attractive) and user friendly.	GUI presented contains significance ideas, innovative values, creative enough (catchy apps name & attractive) and user friendly.	GUI presented contains a very significance ideas, high innovative values, creative enough (catchy apps name & attractive) and user friendly.	1	

**END OF GROUP PROJECT'S INSTRUCTIONS**