



UNIVERSITI TEKNOLOGI MARA
KEDAH BRANCH
SCHOOL OF INFORMATION SCIENCE
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATICS (CDIM144)
IML 208 : PROGRAMMING FOR LIBRARIES

INDIVIDUAL PROJECT : CAB BOOKING SYSTEM

PREPARED BY :
NURUL ALIEYA NATASA BINTI ALIAS SANI
(2022487572)

GROUP KCDIM1443D

PREPARED FOR :
SIR MOHD FIRDAUS MOHD HELMI

SUBMISSION DATE :
December 2023

INDIVIDUAL PROJECT : CAB BOOKING SYSTEM

PREPARED BY
NURUL ALIEYA NATASA BINTI ALIS SANI

GROUP KCDIM1443D

IM144 – DIPLOMA IN LIBRARY INFORMATICS
SCHOOL OF INFORMATION SCIENCE
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS
UNIVERSITI TEKNOLOGI MARA (UITM)
KEDAH BRANCH

PROJECT NAME : Cab Booking System

FILE NAME : bookingsystem.py

PROMPT DATA :

1. **Name** : name of customer
2. **Surname** : surname of customer
3. **Address** : address of customer
4. **Postcode** : postcode customer
5. **Telephone** : no. telephone customer for contact
6. **Email** : email for sent receipt
7. **Location** : location to pick up and drop

FUCTION :

1. **CREATE**
2. **READ**
3. **RESERT**
4. **EXIT**

CONDITIONAL STATEMENT :YES

If-else statement

```
def Kilo():
    if var2.get() == 0:
        self.txtKm.configure(state=DISABLED)
        Km.set("0")
    elif var2.get() == 1 and var11.get() != "" and var12.get() != "":
        self.txtKm.configure(state=NORMAL)
        if var11.get() == "UiTM":
            switch = {"Amanjaya Mall": 12, "Central Square": 13, "KTM SP": 15, "UiTM": 0}
            Km.set(switch[var12.get()])
        elif var11.get() == "Amanjaya Mall":
            switch = {"Amanjaya Mall": 0, "Central Square": 6, "KTM SP": 5, "UiTM": 12}
            Km.set(switch[var12.get()])
        elif var11.get() == "Central Square":
            switch = {"Amanjaya Mall": 6, "Central Square": 0, "KTM SP": 1, "UiTM": 13}
            Km.set(switch[var12.get()])
        elif var11.get() == "KTM SP":
            switch = {"Amanjaya Mall": 5, "Central Square": 1, "KTM SP": 0, "UiTM": 15}
            Km.set(switch[var12.get()])
```

```

def new_user(self):
    #Establish Connection
    with sqlite3.connect('Users.db') as db:
        c = db.cursor()

    #Find Existing username if any take proper action
    find_user = ('SELECT * FROM user WHERE username = ?')
    c.execute(find_user,[self.username.get()])
    if c.fetchall():
        ms.showerror('Error!', 'Username Already Taken!')
    else:
        ms.showinfo('Success!', 'Account Created!')
        self.log()

```

```

bookingsystem.py X
> User > OneDrive > Documents > SEM 3 UITM > IML208 > bookingsystem.py > travel > _init_ > Exit
class Var11:
    def __init__(self):
        switch = {"Amanjaya Mall": 5, "Central Square": 1, "KTM SP": 0, "UITM": 15}
        Km.set(switch[var12.get()])

def Travelling():
    global Item3
    if var3.get() == 1:
        self.txtTravel_Ins.configure(state = NORMAL)
        Item3=float(2)
        Travel_Ins.set("RM " + str(Item3))
    elif var3.get() == 0:
        self.txtTravel_Ins.configure(state = DISABLED)
        Travel_Ins.set("0")
        Item3=0

def Lug():
    global Item4
    if (var4.get()==1):
        self.txtLuggage.configure(state = NORMAL)
        Item4=float(10)
        Luggage.set("RM " + str(Item4))
    elif var4.get()==0:
        self.txtLuggage.configure(state = DISABLED)
        Luggage.set("0")
        Item4=0

```

```

bookingsystem.py X
> User > OneDrive > Documents > SEM 3 UITM > IML208 > bookingsystem.py > travel > _init_ > Exit
class Var11:
    def __init__(self):
        Item4=0

def selectCar():
    global Item5
    if carType.get() == 1:
        self.txtFordGalaxy.configure(state = DISABLED)
        FordGalaxy.set("0")
        self.txtFordMondeo.configure(state = DISABLED)
        FordMondeo.set("0")
        self.txtStandard.configure(state = NORMAL)
        Item5 = float(8)
        Standard.set("RM " + str(Item5))
    elif carType.get() == 2:
        self.txtStandard.configure(state =DISABLED)
        Standard.set("0")
        self.txtFordMondeo.configure(state = DISABLED)
        FordMondeo.set("0")
        self.txtFordGalaxy.configure(state = NORMAL)
        Item5 = float(13)
        FordGalaxy.set("RM " + str(Item5))
    else:
        self.txtStandard.configure(state =DISABLED)
        Standard.set("0")
        self.txtFordGalaxy.configure(state = DISABLED)
        FordGalaxy.set("0")
        self.txtFordMondeo.configure(state = NORMAL)
        Item5 = float(15)
        FordMondeo.set("RM " + str(Item5))

```

```

def Total_Paid():
    if ((var1.get() == 1 and var2.get() == 1 and var3.get() == 1 or var4.get() == 1) and carType.get() != 0 and journeyType.get() != 0 and
        journeyType.get() != 1):
        Item2=km.get()
        Cost_of_fare = (Item1+(float(Item2)*Item5)+Item3+Item4)

        Tax = "RM " + str("%.1f"%(Cost_of_fare) *0.03))
        ST = "RM " + str("%.1f"%(Cost_of_fare)))
        TT = "RM " + str("%.1f"%(Cost_of_fare+(Cost_of_fare)*0.2)))
    elif journeyType.get() == 2:
        Item2=km.get()
        Cost_of_fare = (Item1+(float(Item2)*Item5)*1.1+Item3+Item4)

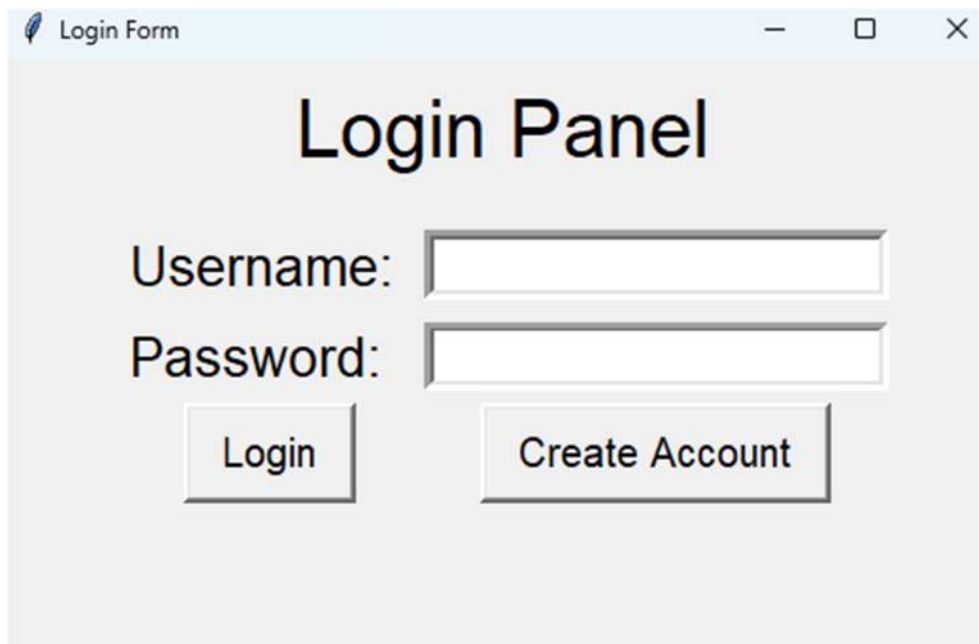
        Tax = "RM " + str("%.1f"%(Cost_of_fare) *0.06))
        ST = "RM " + str("%.1f"%(Cost_of_fare)))
        TT = "RM " + str("%.1f"%(Cost_of_fare+(Cost_of_fare)*0.2)))
    else:
        Item2=km.get()
        Cost_of_fare = (Item1+(float(Item2)*Item5)*2+Item3+Item4)

        Tax = "RM " + str("%.1f"%(Cost_of_fare) *0.09))
        ST = "RM " + str("%.1f"%(Cost_of_fare)))
        TT = "RM " + str("%.1f"%(Cost_of_fare+(Cost_of_fare)*0.2)))

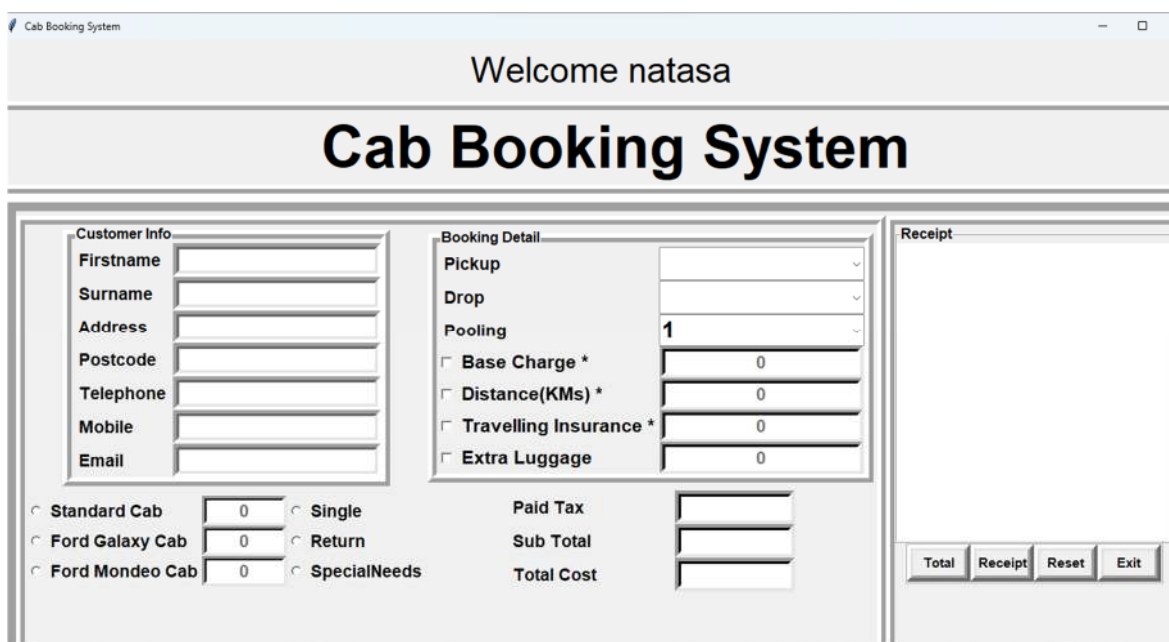
    PaidTax.set(Tax)
    SubTotal.set(ST)
    TotalCost.set(TT)
else:
    w = ms.showwarning("Error !","Invalid Input\nPlease try again !!!")

```

GUI : YES



A screenshot of a 'Login Form' window. The window has a title bar with a feather icon, the text 'Login Form', and standard window controls (minimize, maximize, close). The main content area has a light gray background. At the top, the text 'Login Panel' is displayed in a large, bold, black font. Below this, there are two labels: 'Username:' and 'Password:'. Each label is followed by a white rectangular text input field with a thin gray border. Below the input fields, there are two buttons: 'Login' and 'Create Account'. Both buttons are white with a thin gray border and black text.



A screenshot of a 'Cab Booking System' window. The window has a title bar with a feather icon, the text 'Cab Booking System', and standard window controls. The main content area has a light gray background. At the top, the text 'Welcome natasa' is displayed in a black font. Below this, the text 'Cab Booking System' is displayed in a large, bold, black font. The main content area is divided into three sections: 'Customer Info', 'Booking Detail', and 'Receipt'. The 'Customer Info' section contains labels for 'Firstname', 'Surname', 'Address', 'Postcode', 'Telephone', 'Mobile', and 'Email', each followed by a white rectangular text input field. The 'Booking Detail' section contains labels for 'Pickup', 'Drop', and 'Pooling', each followed by a white rectangular dropdown menu. Below these, there are four checkboxes: 'Base Charge *', 'Distance(KMs) *', 'Travelling Insurance *', and 'Extra Luggage', each followed by a white rectangular text input field. The 'Receipt' section is a large white rectangular area. At the bottom of the window, there are four buttons: 'Total', 'Receipt', 'Reset', and 'Exit'. On the left side of the bottom section, there are three radio buttons: 'Standard Cab', 'Ford Galaxy Cab', and 'Ford Mondeo Cab', each followed by a white rectangular text input field. To the right of these, there are three radio buttons: 'Single', 'Return', and 'SpecialNeeds', each followed by a white rectangular text input field. In the center of the bottom section, there are three labels: 'Paid Tax', 'Sub Total', and 'Total Cost', each followed by a white rectangular text input field.

RESULTS :

Welcome natasa

Cab Booking System

Customer Info	Booking Detail	Receipt
Firstname: <input type="text" value="alleya"/>	Pickup: <input type="text" value="Amanjaya Mall"/>	Receipt Ref: 223545
Surname: <input type="text" value="alias"/>	Drop: <input type="text" value="UiTM"/>	Date: 13 / 12 / 2023
Address: <input type="text" value="sungai petani"/>	Pooling: <input type="text" value="4"/>	Cab No: TR 223545 BW
Postcode: <input type="text" value="084000"/>	<input checked="" type="checkbox"/> Base Charge * <input type="text" value="RM 3.0"/>	Firstname: alleya
Telephone: <input type="text" value="0165067022"/>	<input checked="" type="checkbox"/> Distance(KMs) * <input type="text" value="12"/>	Surname: alias
Mobile: <input type="text" value="0124676421"/>	<input checked="" type="checkbox"/> Travelling Insurance * <input type="text" value="RM 2.0"/>	Address: sungai petani
Email: <input type="text" value="natasa@gmailcom"/>	<input type="checkbox"/> Extra Luggage <input type="text" value="0"/>	Postal Code: 084000
<input type="radio"/> Standard Cab <input type="text" value="0"/>	Paid Tax <input type="text" value="RM 10.6"/>	Telephone: 0165067022
<input checked="" type="radio"/> Ford Galaxy Cab <input type="text" value="RM 13.0"/>	Sub Total <input type="text" value="RM 176.6"/>	Mobile: 0124676421
<input type="radio"/> Ford Mondeo Cab <input type="text" value="0"/>	Total Cost <input type="text" value="RM 211.9"/>	Email: natasa@gmailcom
<input type="radio"/> Single		From: Amanjaya Mall
<input checked="" type="radio"/> Return		To: UiTM
<input type="radio"/> SpecialNeeds		Pooling: 1
		Standard: 0
		Prime Sedan: RM 13.0
		Premium Sedan: 0
		Paid: RM 10.6
		SubTotal: RM 176.6
		Total Cost: RM 211.9
		<input type="button" value="Total"/> <input type="button" value="Receipt"/> <input type="button" value="Reset"/> <input type="button" value="Exit"/>

STRENGTH :

1. Code Organization:

- Consider organizing your code into multiple modules or classes to improve readability and maintainability.
- Use functions or methods to encapsulate logical blocks of code, making it easier to understand.

2. User Interface:

- Improve the user interface by providing clear instructions and labels for the input fields and buttons.
- Add comments in your code to explain the purpose and functionality of each section.

3. Database Handling:

- Consider adding error handling and validation when interacting with the database. For example, check if the connection to the database is successful and handle potential errors gracefully.

4. Security:

- Implement secure password handling. Storing passwords in plain text is not recommended. Consider using techniques like hashing for storing and validating passwords.

5. Error Handling:

- Implement comprehensive error handling to handle unexpected situations and provide meaningful error messages to the users.

6. Code Reusability:

- Try to write reusable code by creating functions or classes for common tasks.

7. Consistent Naming:

- Maintain a consistent naming convention for variables and functions. This enhances code readability.

8. User Feedback:

- Provide informative messages to the users, especially when errors occur. This helps users understand what went wrong and how to fix it.

9. Testing:

- Test your code thoroughly with different scenarios to ensure its robustness. Consider using unit tests to automate the testing process.

10. Documentation:

- Add comments and docstrings to explain the purpose of your functions, classes, and modules.

KAIZEN (ROOM FOR IMPROVEMENT) :

1. Need to improve code for background color
2. Need to specify the place
3. Need more place for student