**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Contents** | **Page No.** |
| **1** | **ABSTRACT** | **3** |
| **2** | **OBJECTIVES** | **4-5** |
| **3** | **INTRODUCTION/THEORY** | **6** |
| **4** | **ALOGORITHM & FLOWCHART** | **7-9** |
| 5 | CODE | **9-10** |
| **6** | **OUTPUT** | **10-11** |
| **7** | **CONCLUSIONS** | **14-15** |
| **8** | **REFERENCES** | **17** |

**ABSTRACT**

This is a Python project using Tkinter library for GUI, designed to create a simple cab/taxi booking system with user registration and login. Users can fill out details such as pick-up and destination locations, cab selection, pooling, etc. The system generates a total cost, tax, and sub-total and provides a receipt with reference number, date, cab number, total cost, and other details. The GUI is simple and designed for a good user experience.

**OBJECTIVES**

**Develop a user-friendly GUI for managing cab services using the Tkinter library.**

To create a user-friendly GUI for managing cab services, you need to design the layout of the GUI, create the necessary widgets such as buttons, labels, text fields, etc., and add functionality to the GUI such as the ability to add, delete, and modify cab details, view cab availability, book cabs, and generate reports.

Testing the GUI is also an important step to ensure that all the buttons and text fields work as expected, and the GUI is easy to use and visually appealing.

To make the GUI user-friendly, it is important to keep the layout simple and uncluttered, use clear and concise labels and instructions, use consistent and intuitive button and menu designs, use colors and graphics sparingly, and provide feedback to the user at all stages of the process.

**Integrate the GUI with the underlying logic of the cab management system to provide a seamless user experience.**

Integrating the GUI with the underlying logic of the cab management system involves connecting the user interface with the back-end system that manages cab bookings and other operations. This integration is important to provide a seamless user experience where the user can interact with the GUI and see real-time updates on their bookings and other activities.

The integration involves implementing functions and methods in the back-end system that can receive user inputs from the GUI and perform the necessary operations such as booking a cab, displaying cab availability, calculating fares, and generating reports. These functions and methods should be designed to work in a way that is compatible with the GUI and can provide real-time updates to the user interface.

**INTRODUCTION / THEORY**

A simple project based on Cab/Taxi Booking System which uses Python Language with Tkinter Library for GUI. Following Python with Tkinter Library project contains the least, but important features which can be in use for the first-year IT students for their college projects. It has features that will allow all the users to interact in a way that the cab drivers interact with their customers regarding their cab/car bookings. This system as well as the python application’s concept is all clear, it’s the same as real-life scenarios and well-implemented on it. Moving on, this cab booking system project in Python focuses mainly on dealing with customer’s booking details with their respective bill amounts. Also, the system allows registration of customers. The project only contains User Panel. In an overview of this app, the customer has to register and login in order to use the system. Talking more about the project, the user simply has to fill up details such as name, contact details, address information, select cabs, pick up location, destination location, pooling, etc. Besides, the system generates total cost, tax and sub total with all the details given by the user. The user can also view overall receipt of their details with receipt reference number, date,

cab number, total cost and other information.Last but not least, a clean and simple GUI is presented with simple color combinations for a greater user experience while using this simple cab booking system project in Python. For its UI elements, a standard GUI library; Tkinter is on board. Presenting a new taxi, car booking system project in Python which includes a user panel that contains all the essential features to follow up, and a knowledgeable resource for learning purposes**.**



**ALGORITHM OF THE CAB BOOKING SYSTEM**

**Step1 - User Authentication:**

1. User logs into the system using their credentials (username and password).
2. The system verifies the user's credentials.
3. If the user is not registered, they can create an account.

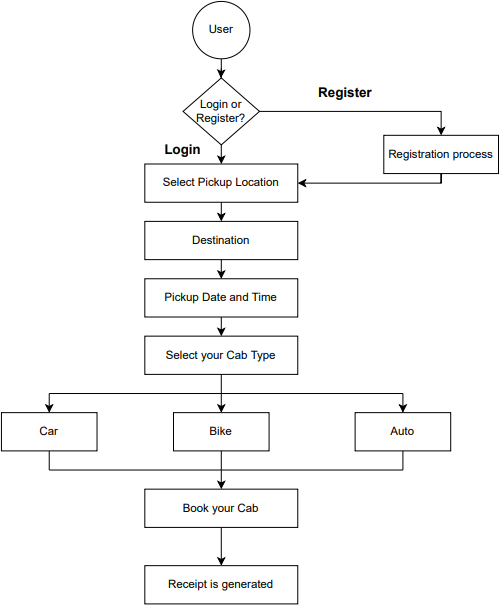
**Step2 - Selecting Cab:**

1. User selects the type of cab they want to book (auto, bike, car).
2. The system shows the available cabs of the selected type.
3. User selects the desired cab.

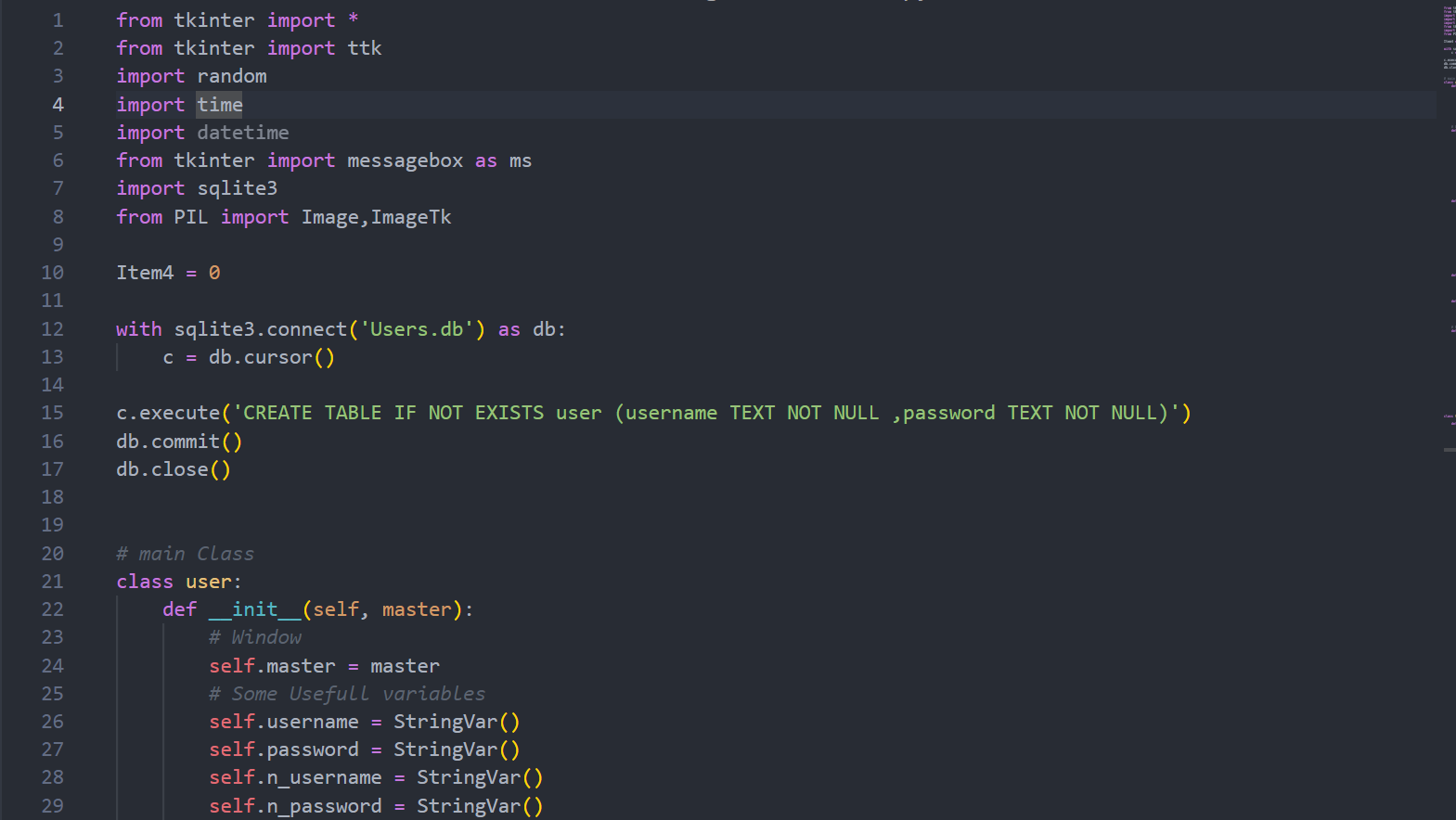
**Step 3 - Booking Cab:**

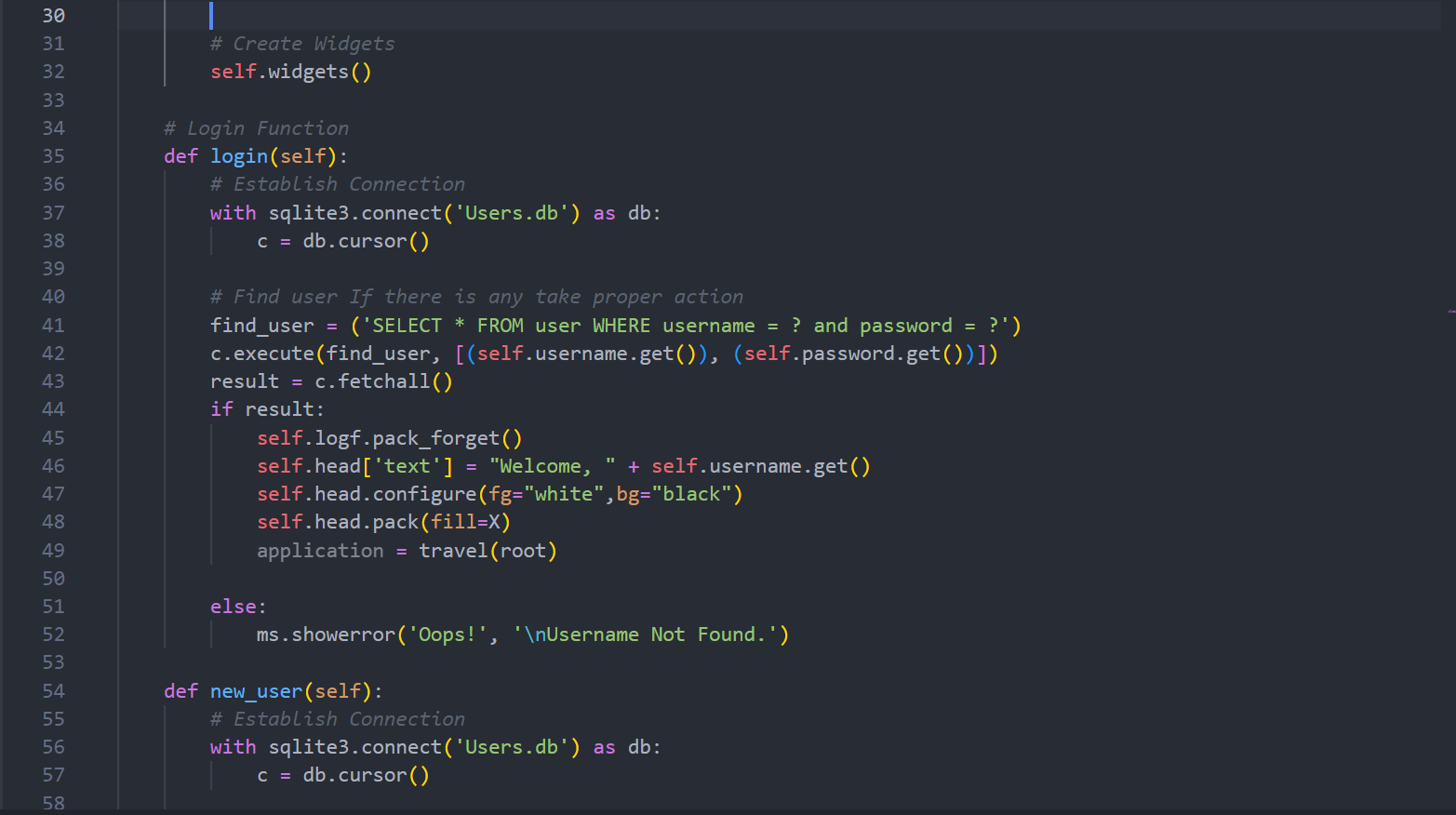
1. User enters their pickup location and destination.
2. The system calculates the fare and taxes.
3. User confirms the booking.
4. The system reserves the cab for the user.

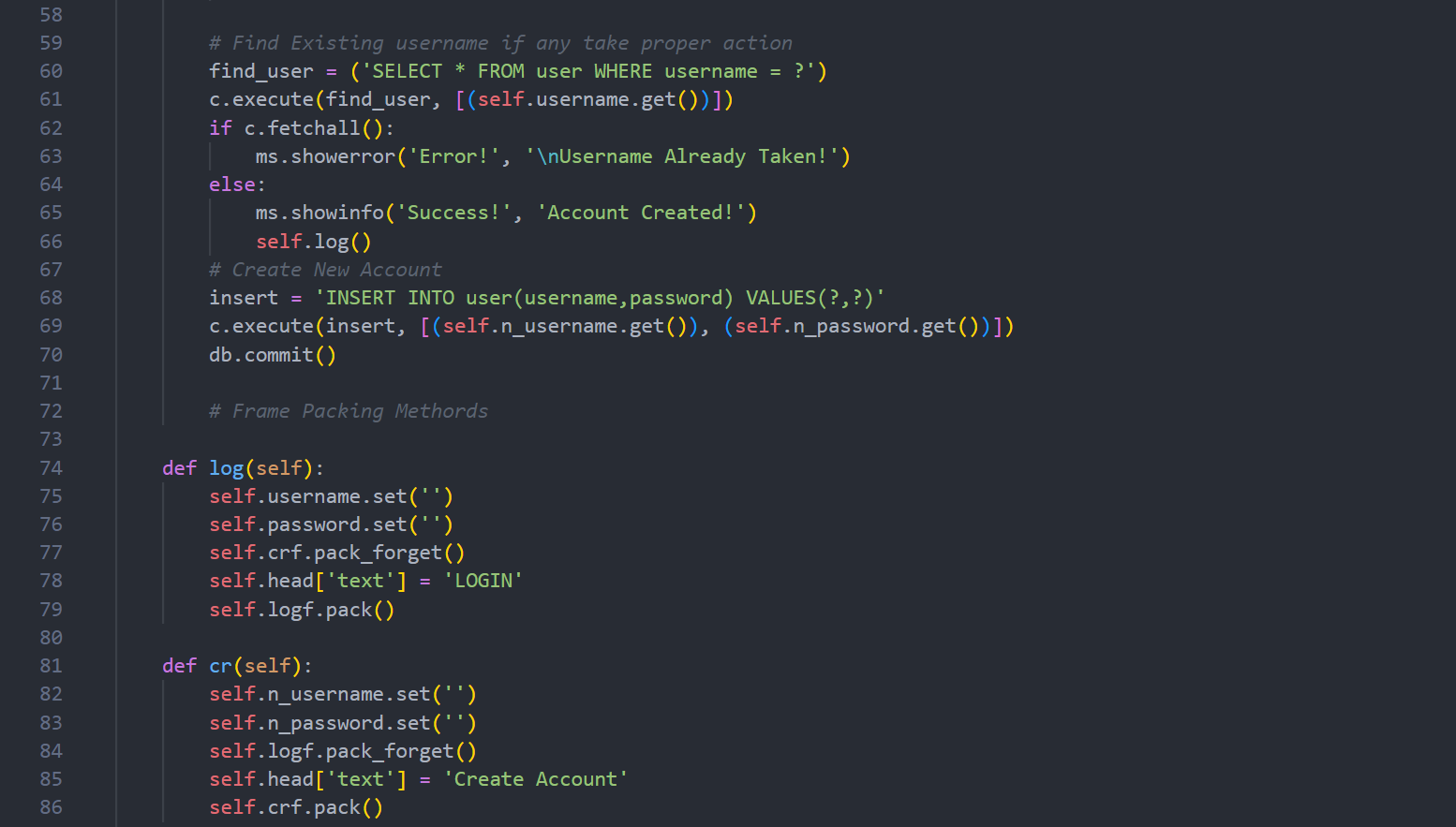
**Step 4 - Storing User Information:**  
a. The system stores the user's information (name, address, contact details, email.).  
b. The system stores the details of the booked cab (Receipt reference number, date ,cab number, total cost).  
 **Step 5 - Cost of Cab Booking and Tax Generation:**  
a. The system calculates the fare based on distance and time.  
b. The system adds the taxes to the fare.  
c. The system displays the total cost to the user.  
  
**Step 6 - Booking Receipt:**  
a. The system generates a booking receipt that includes the user's details, receipt reference number, cab details, pickup and drop-off locations, fare, taxes, and booking date and time.  
b. The system sends the booking receipt to the user's email or mobile number.  
 **Step 7 - Log Out:**  
Once the user has completed their booking, they can log out of the system.

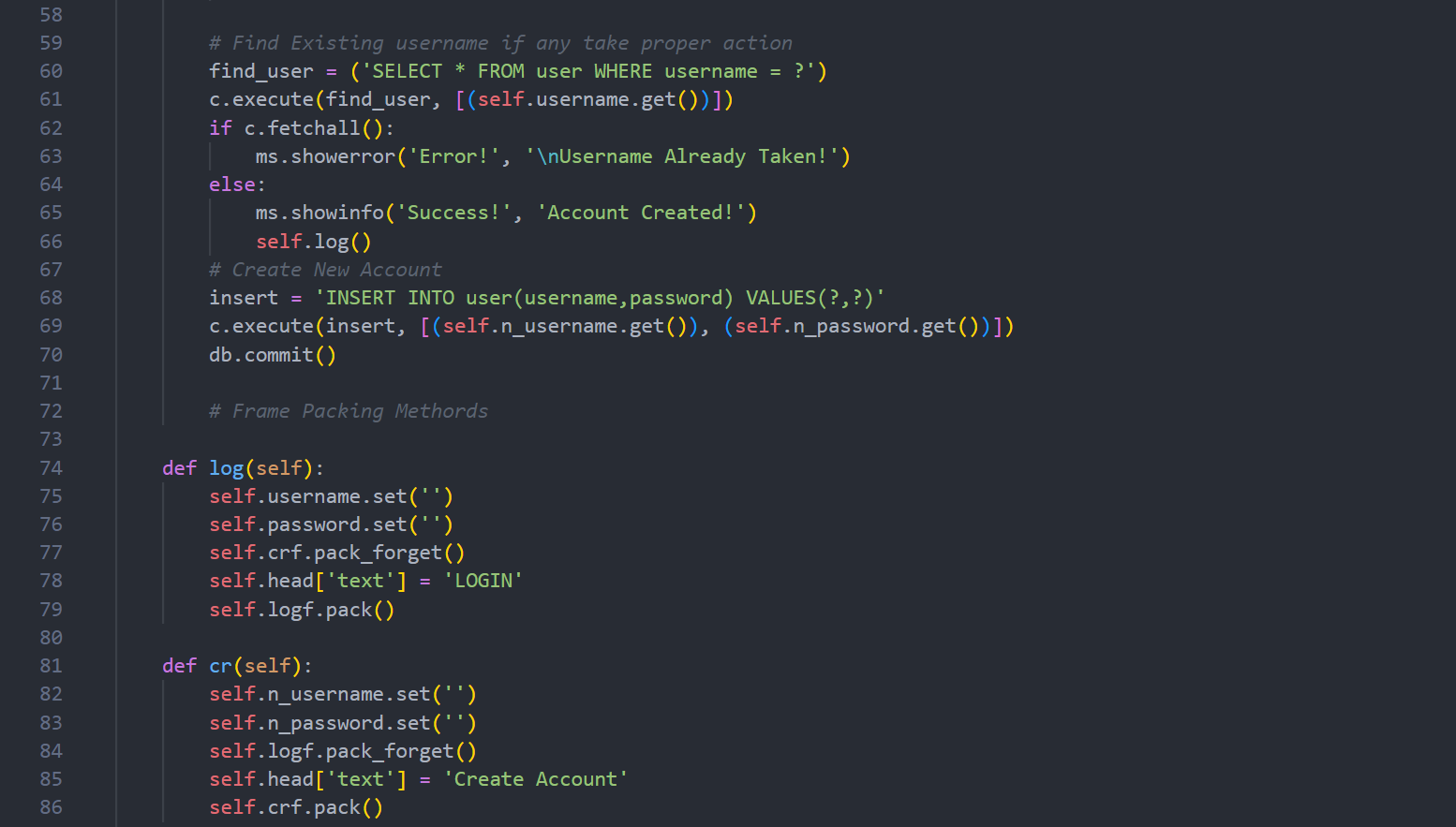
**FLOWCHART**

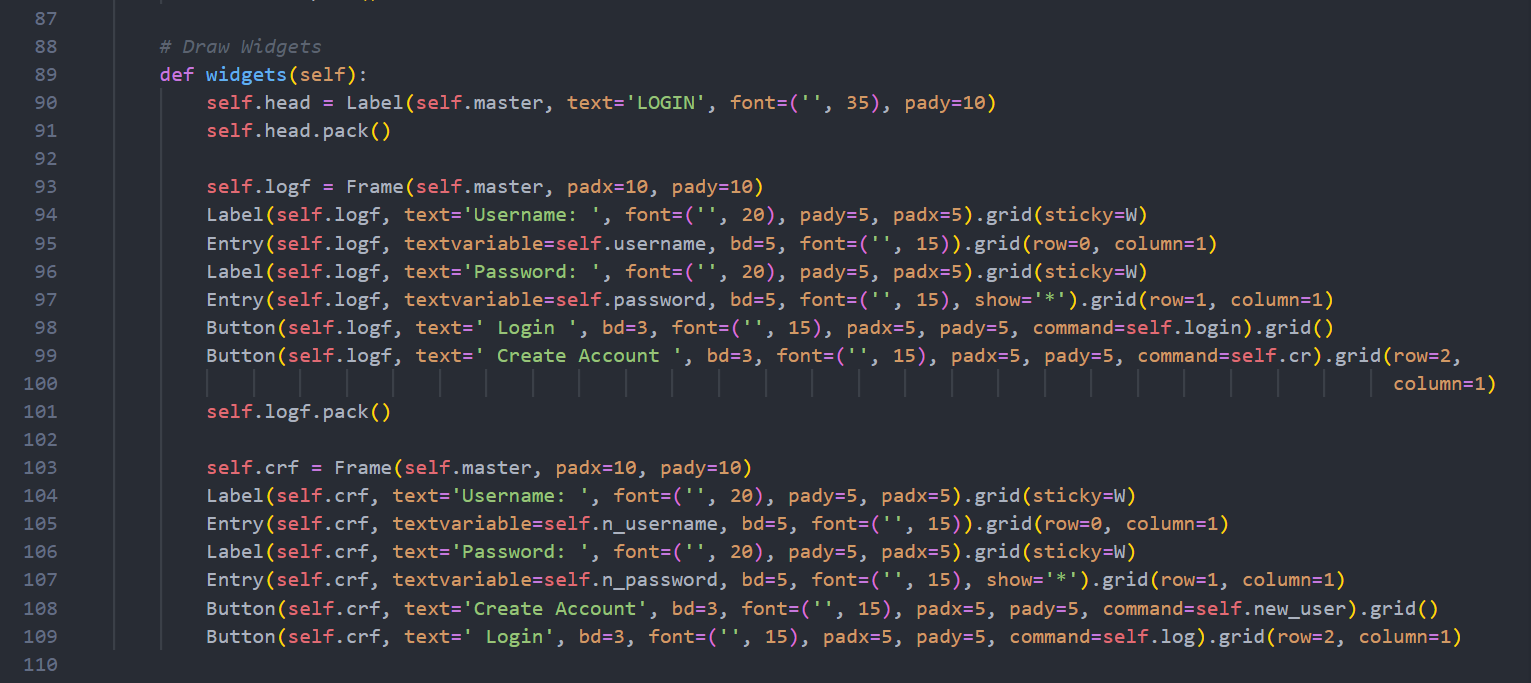
**CODE**

****

****

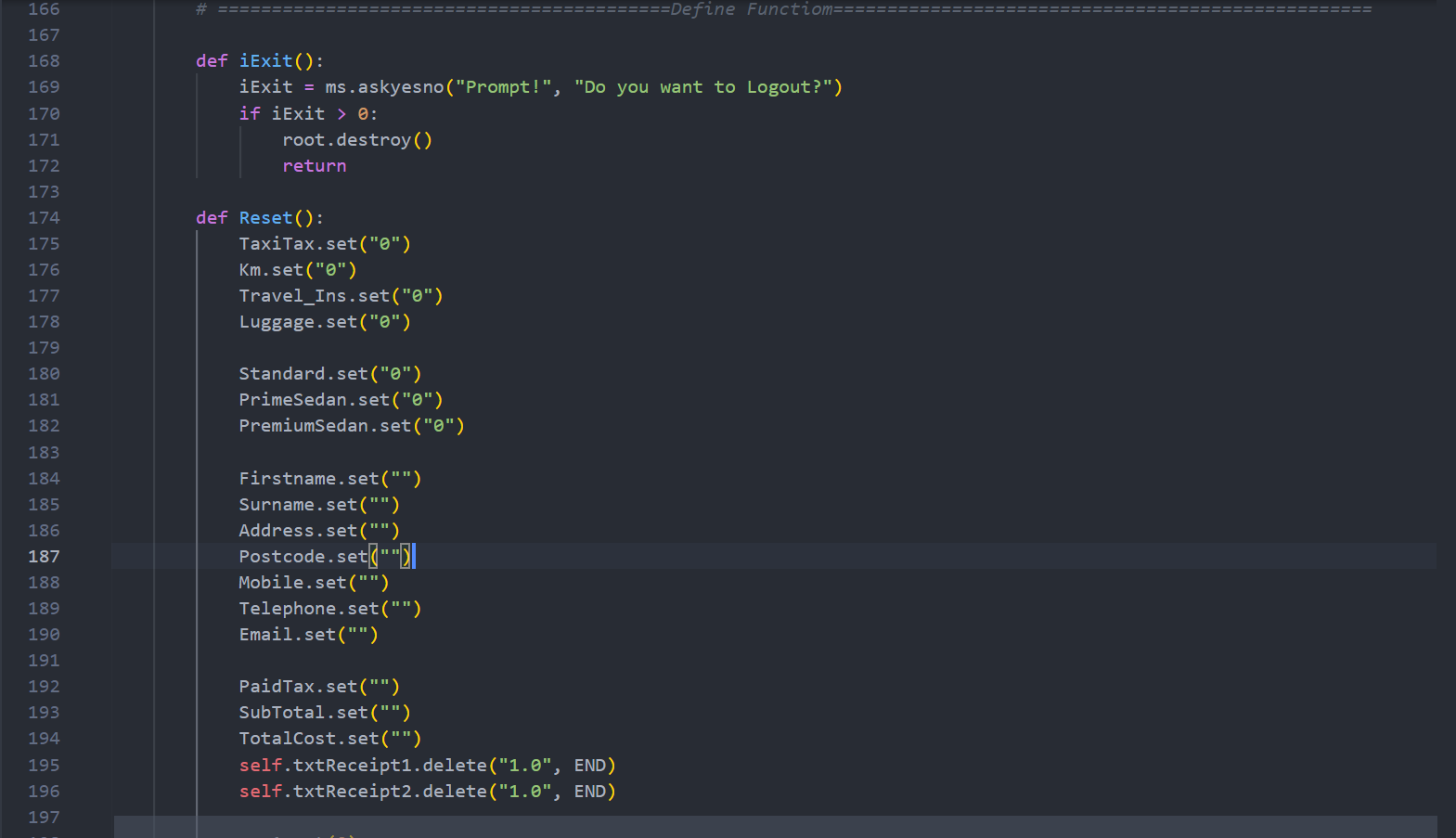
****

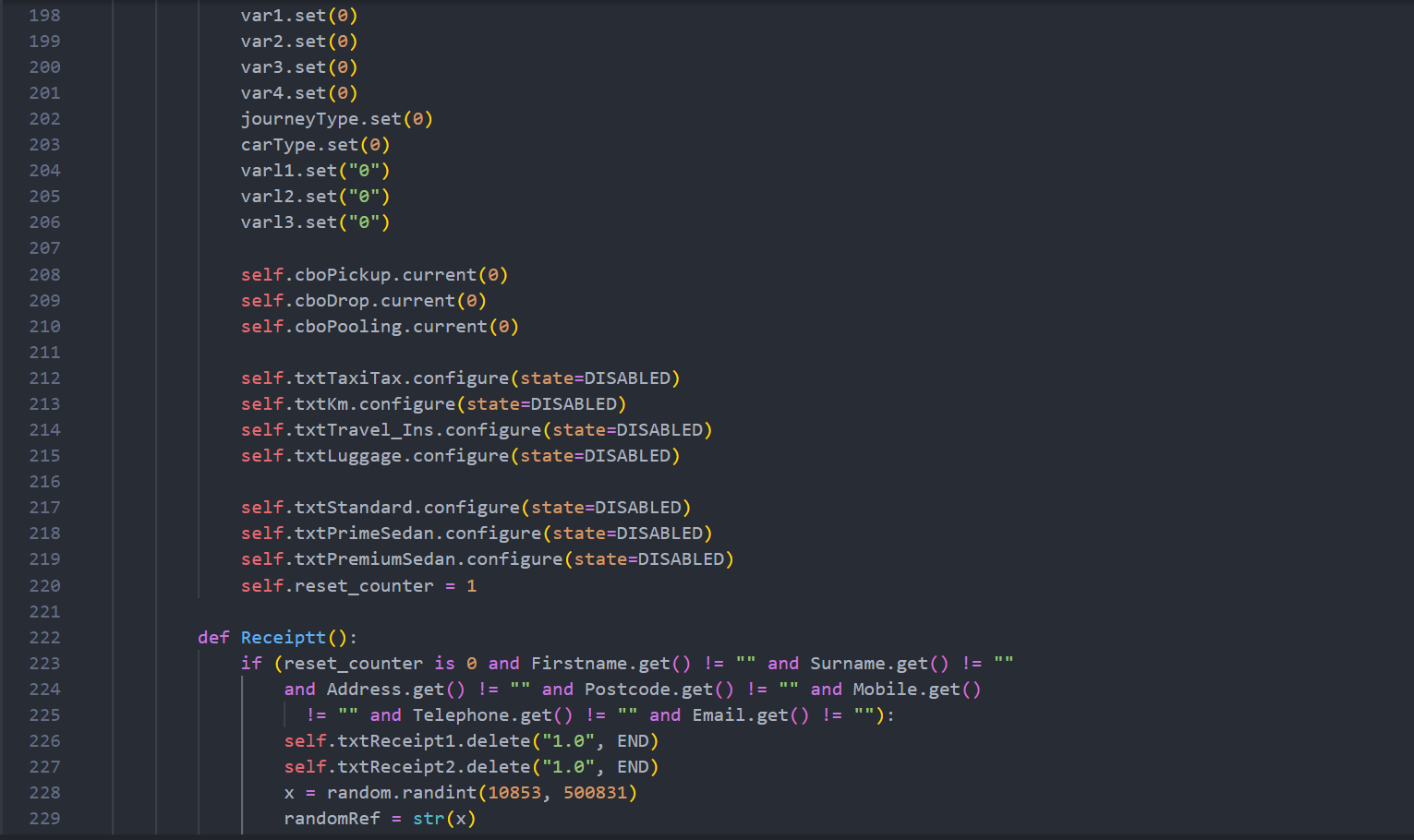
****

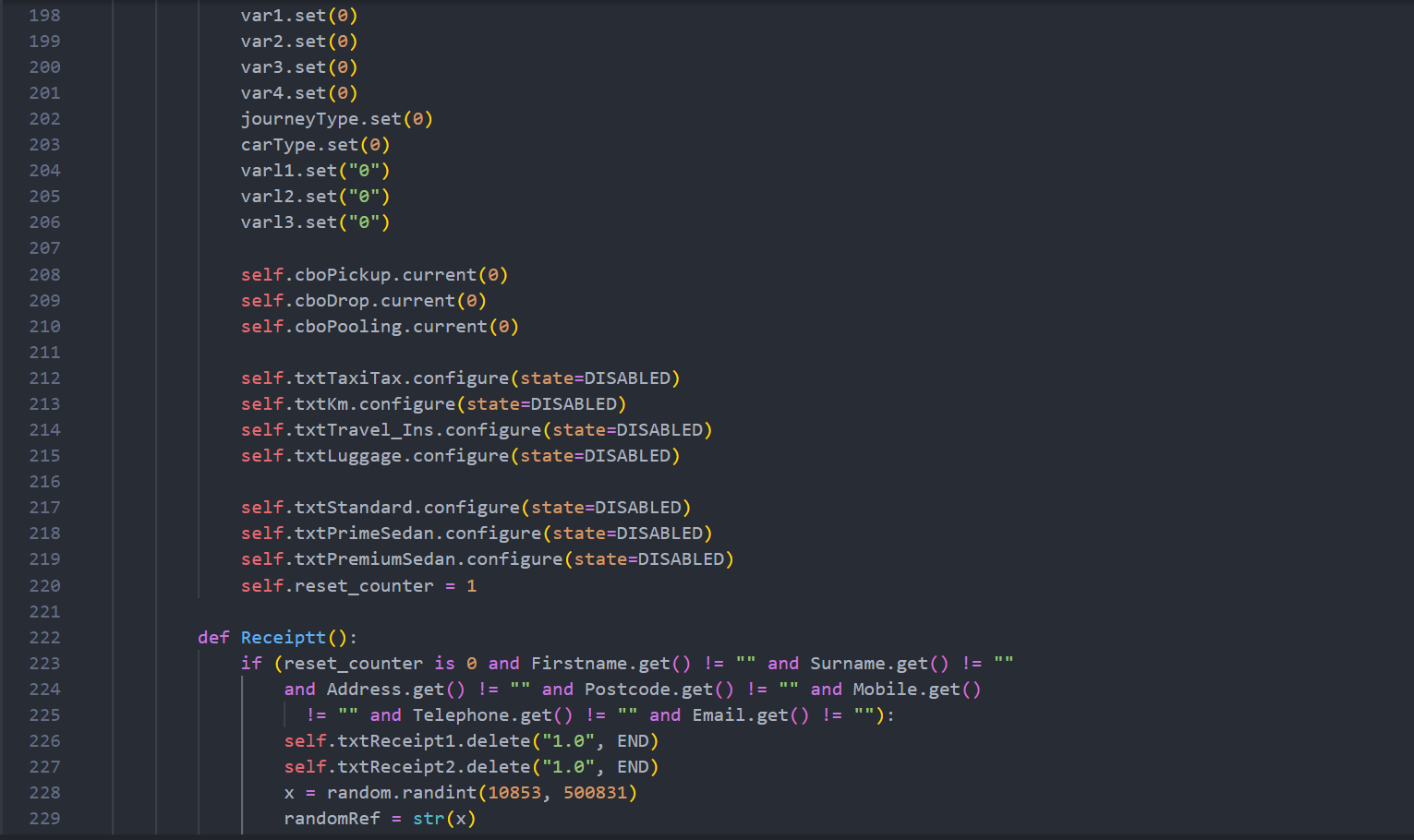
****

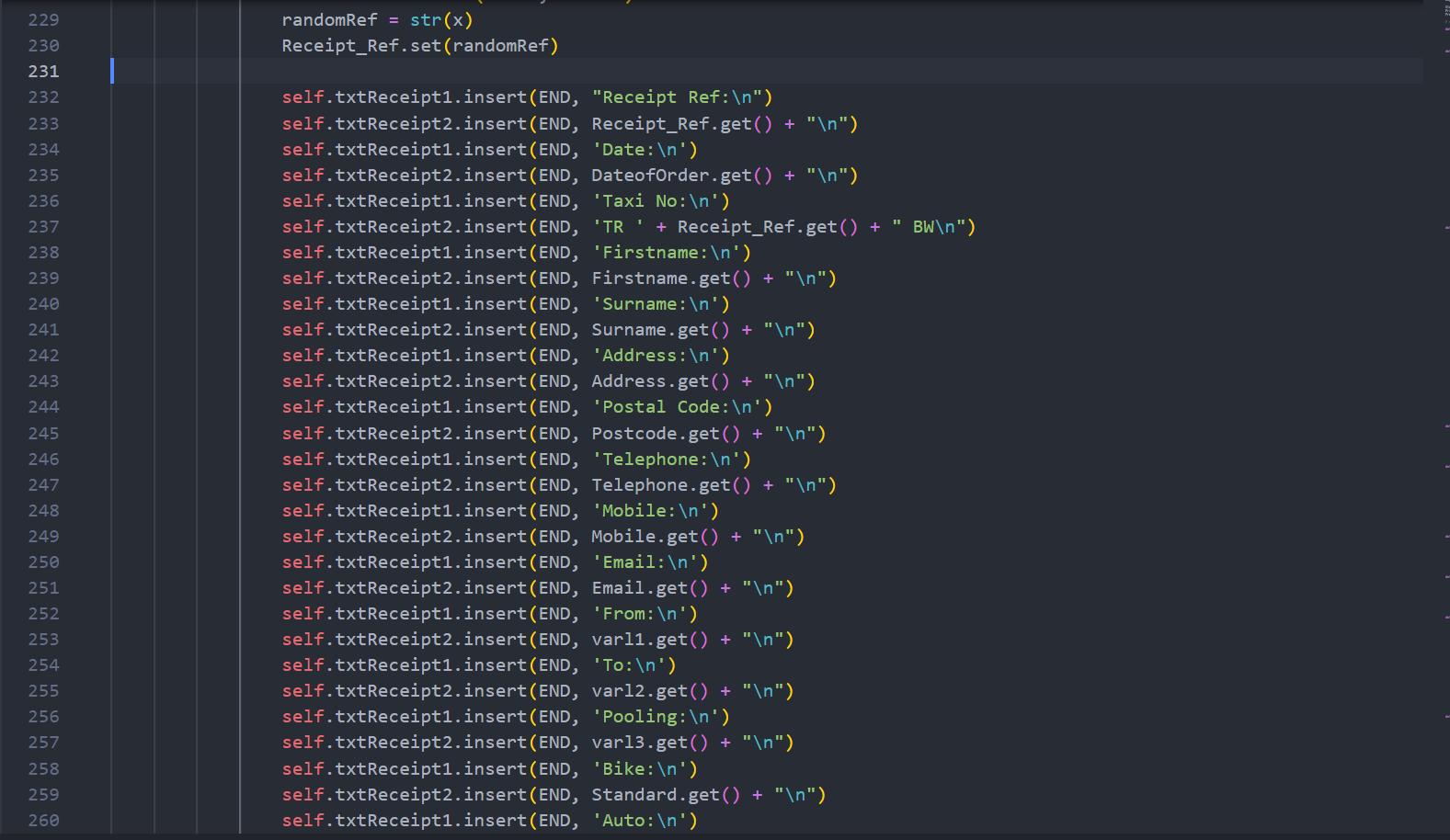
****

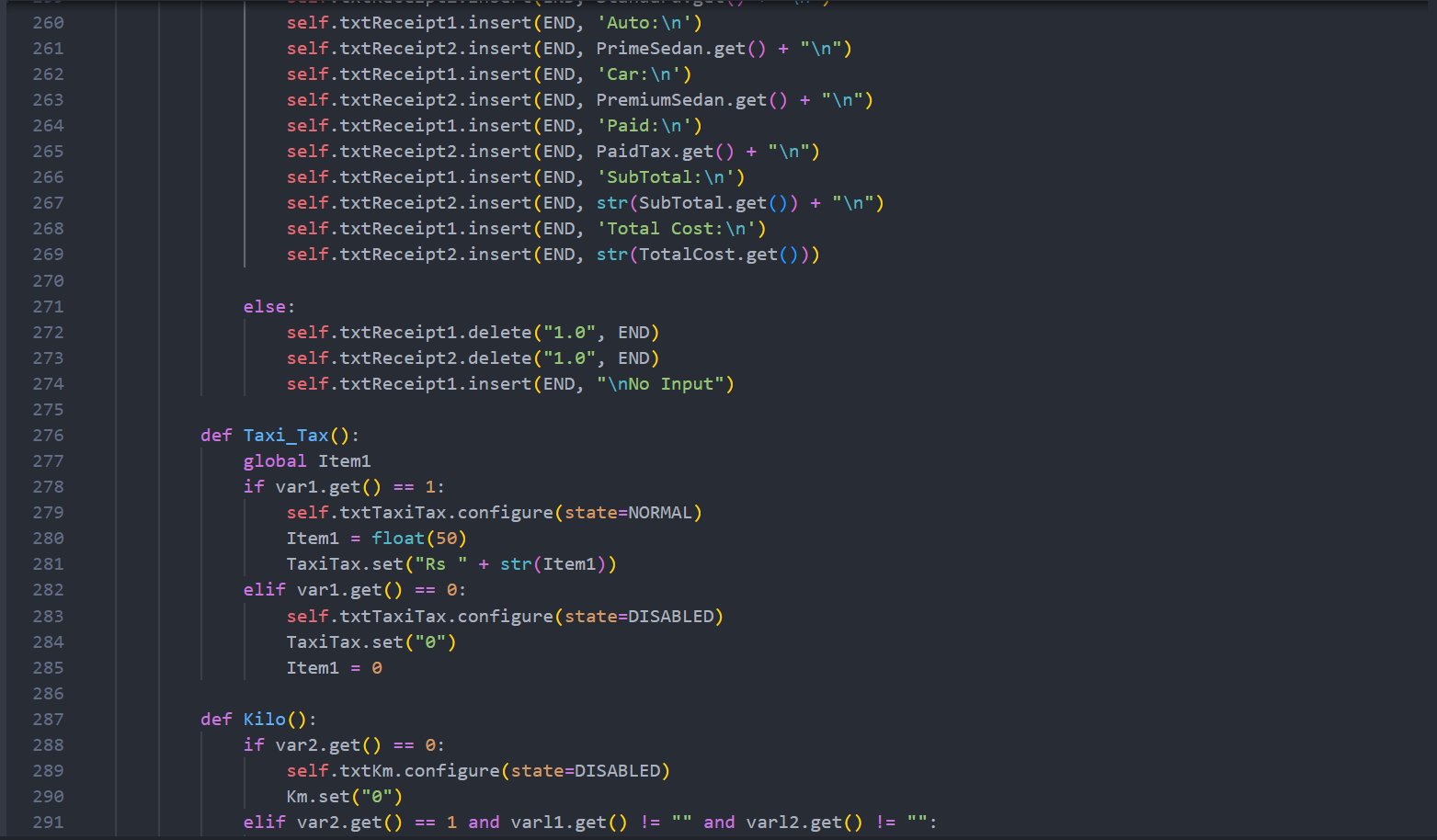
****

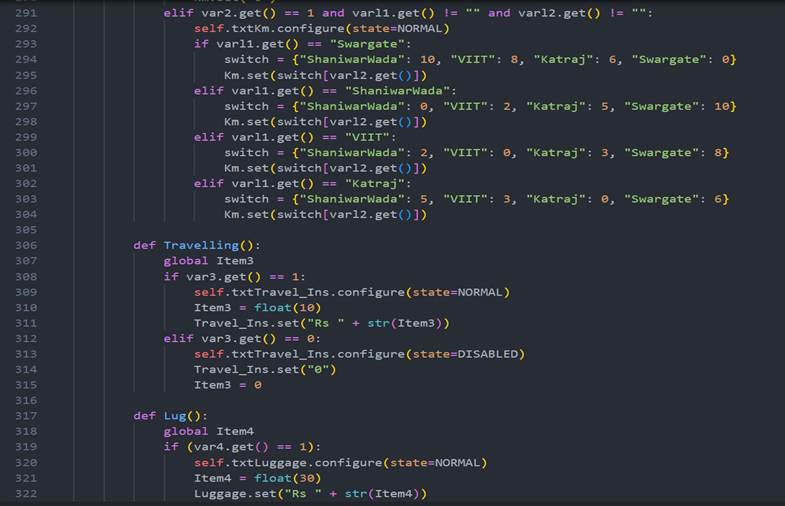
****

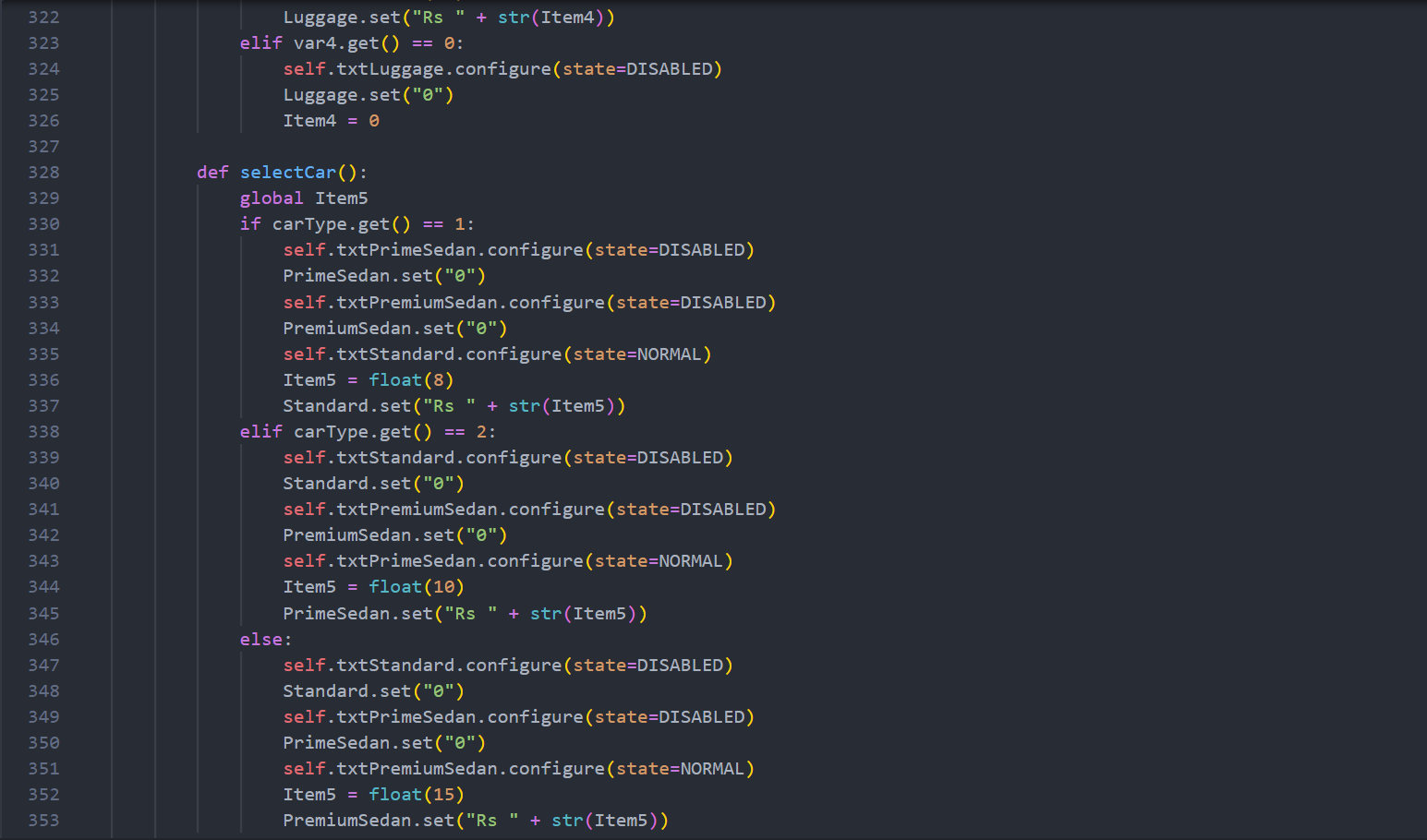
****

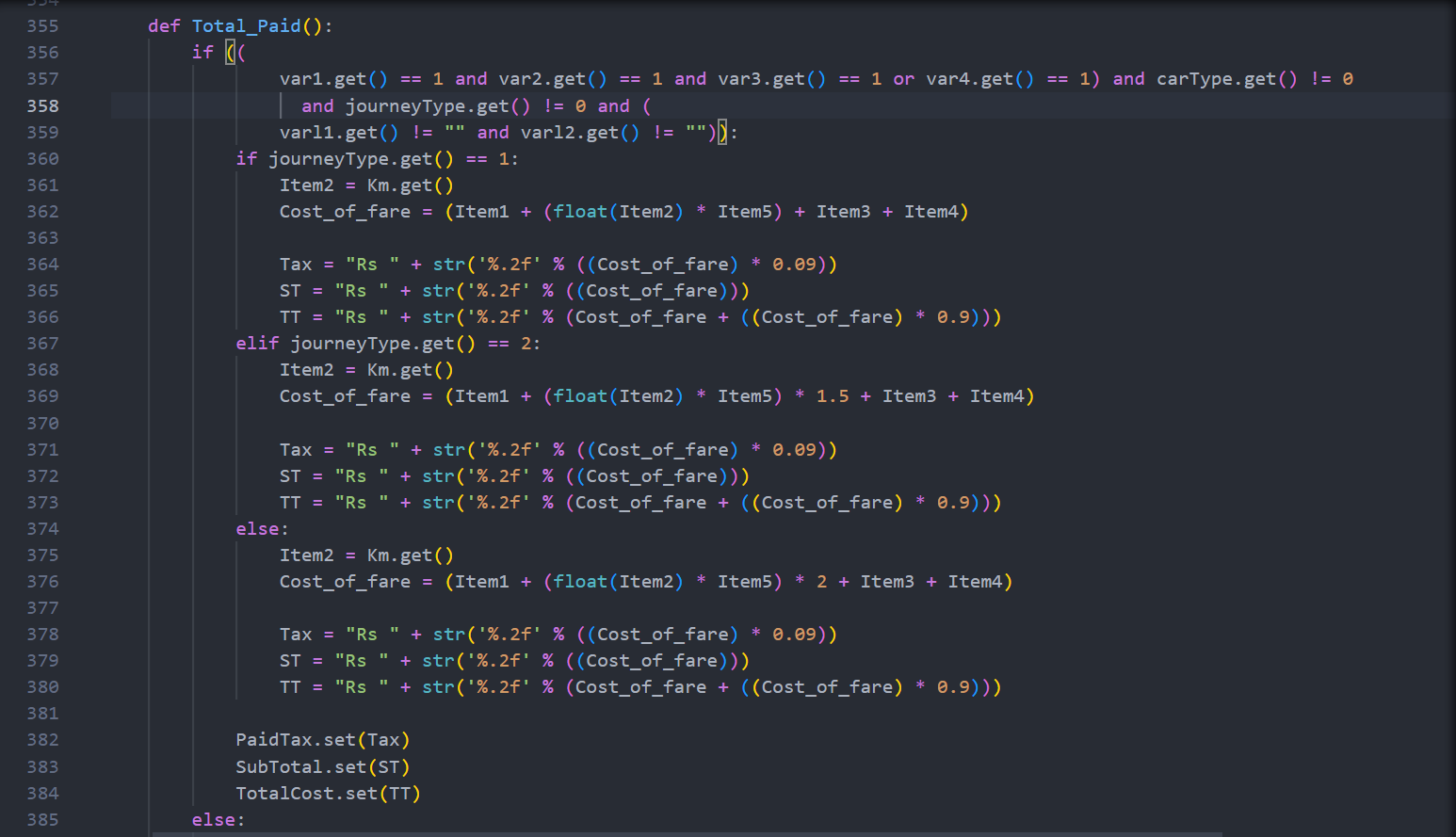
****

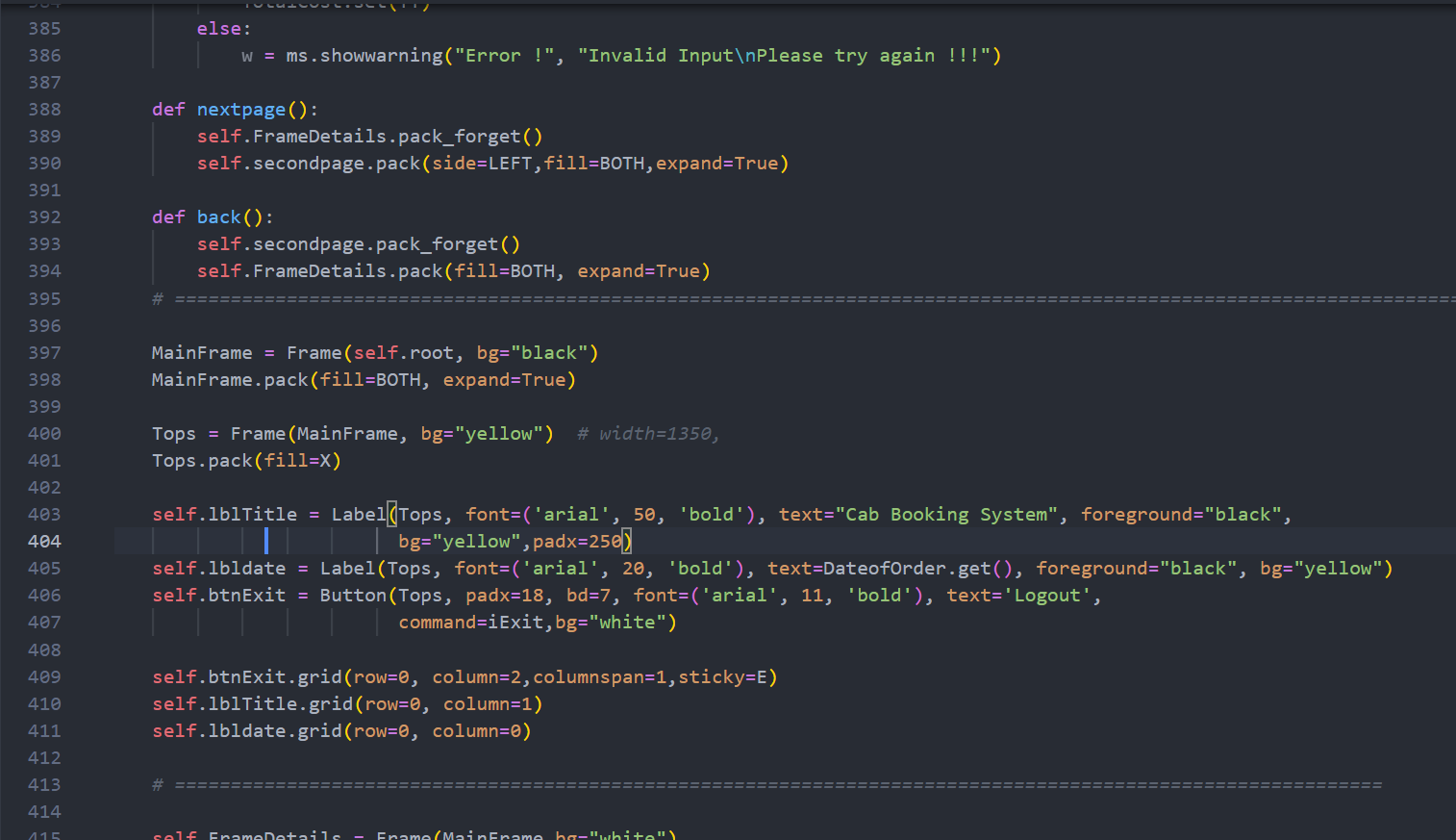
****

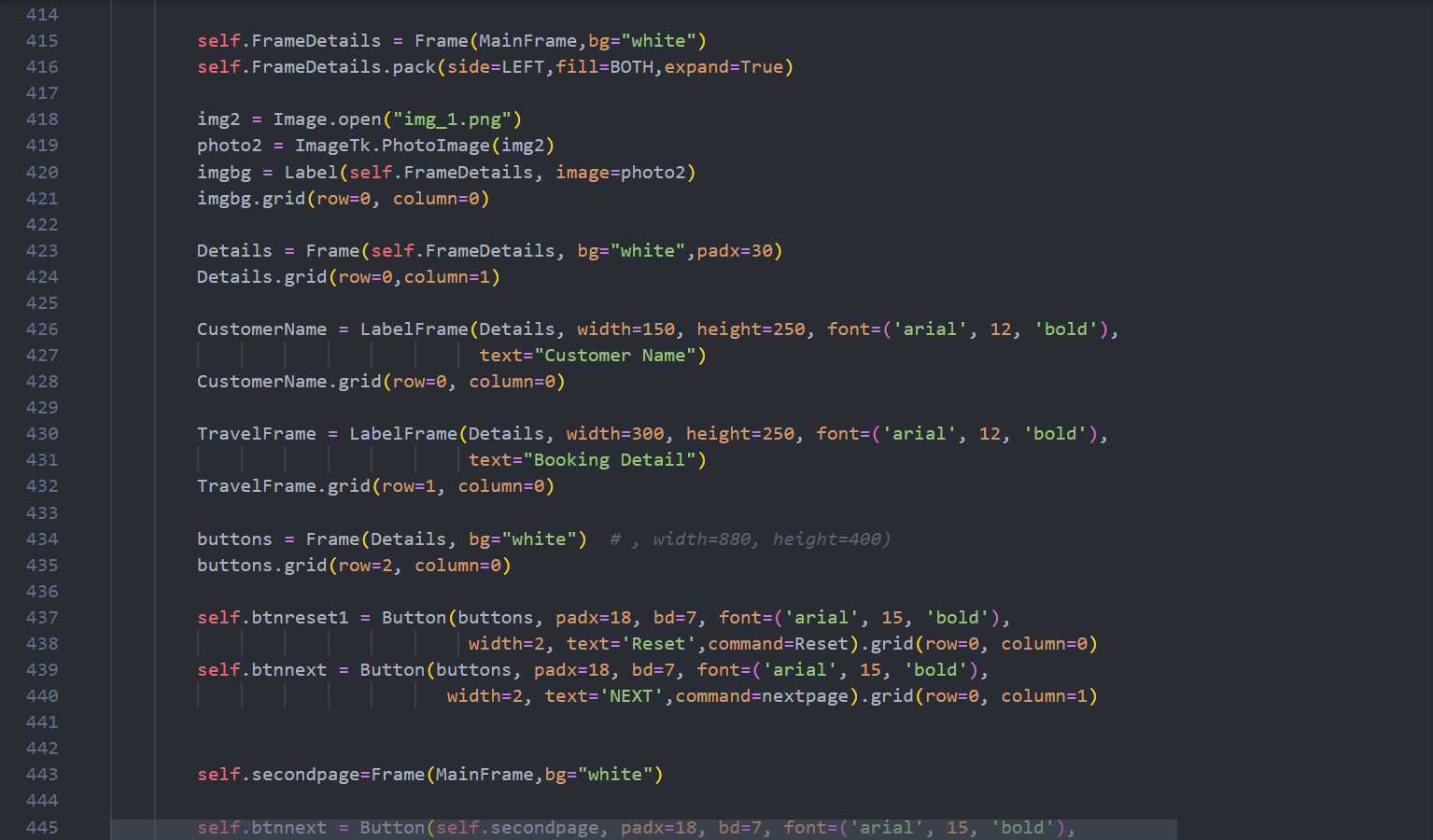
****

****

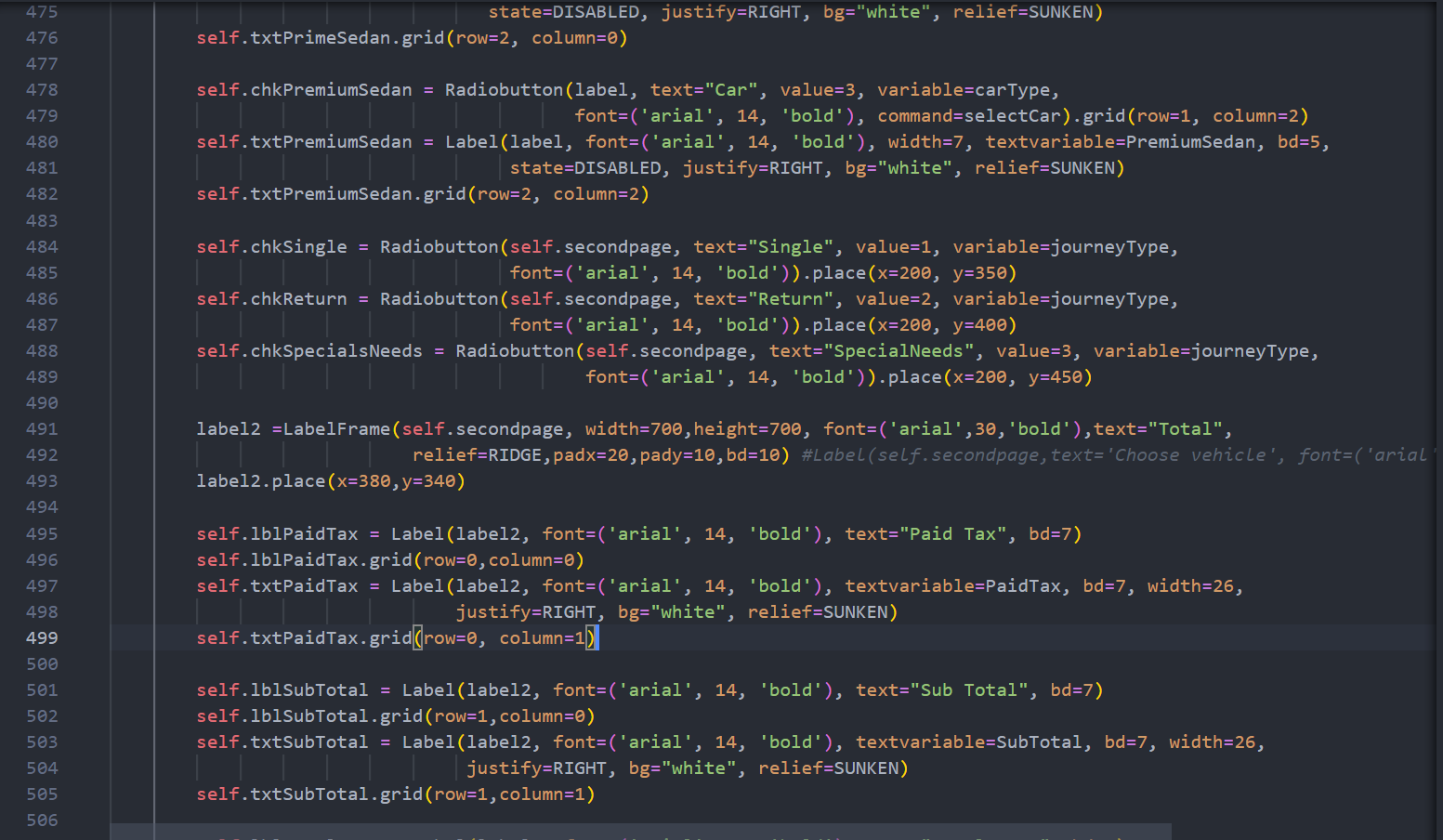
****

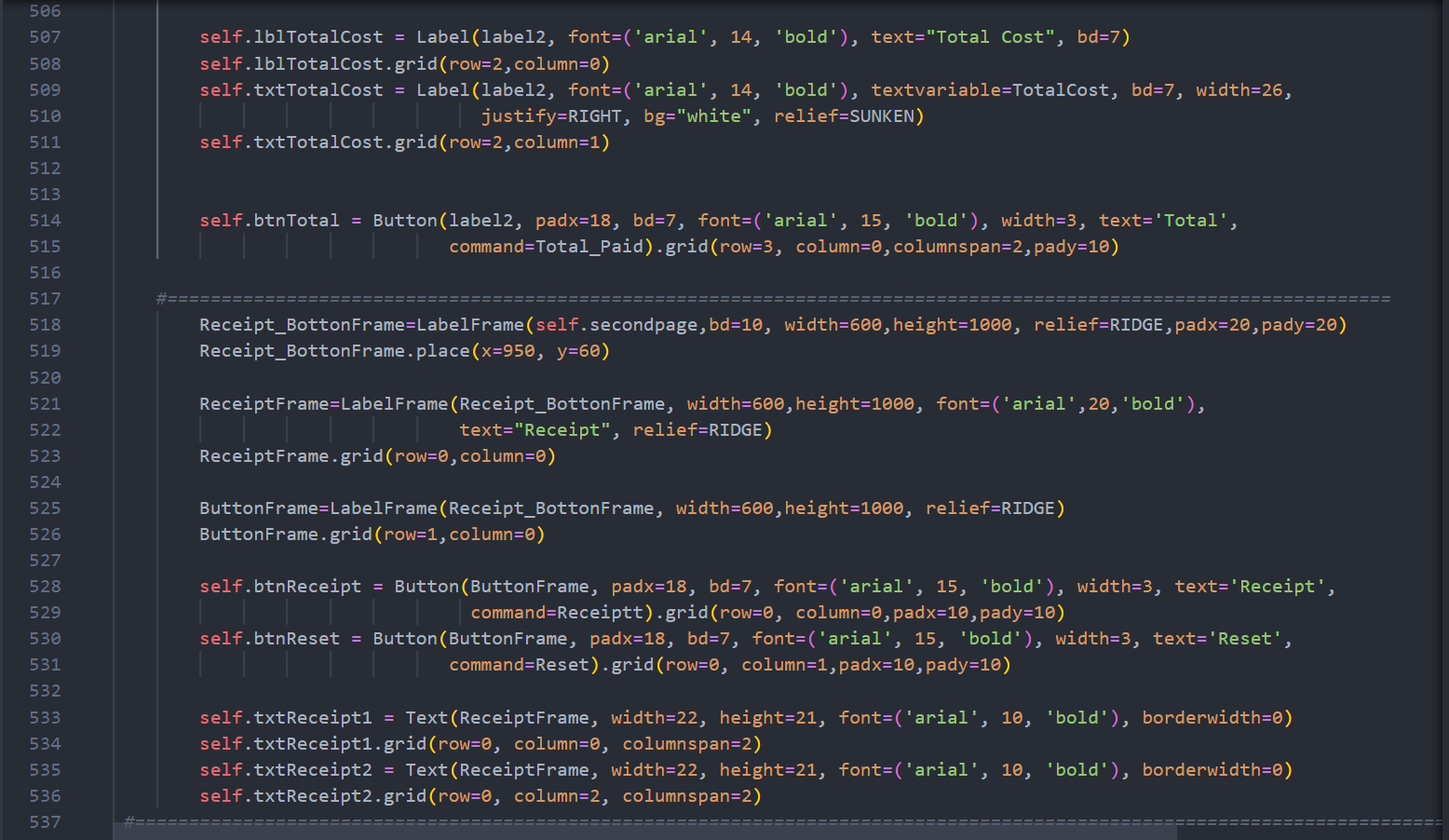
****

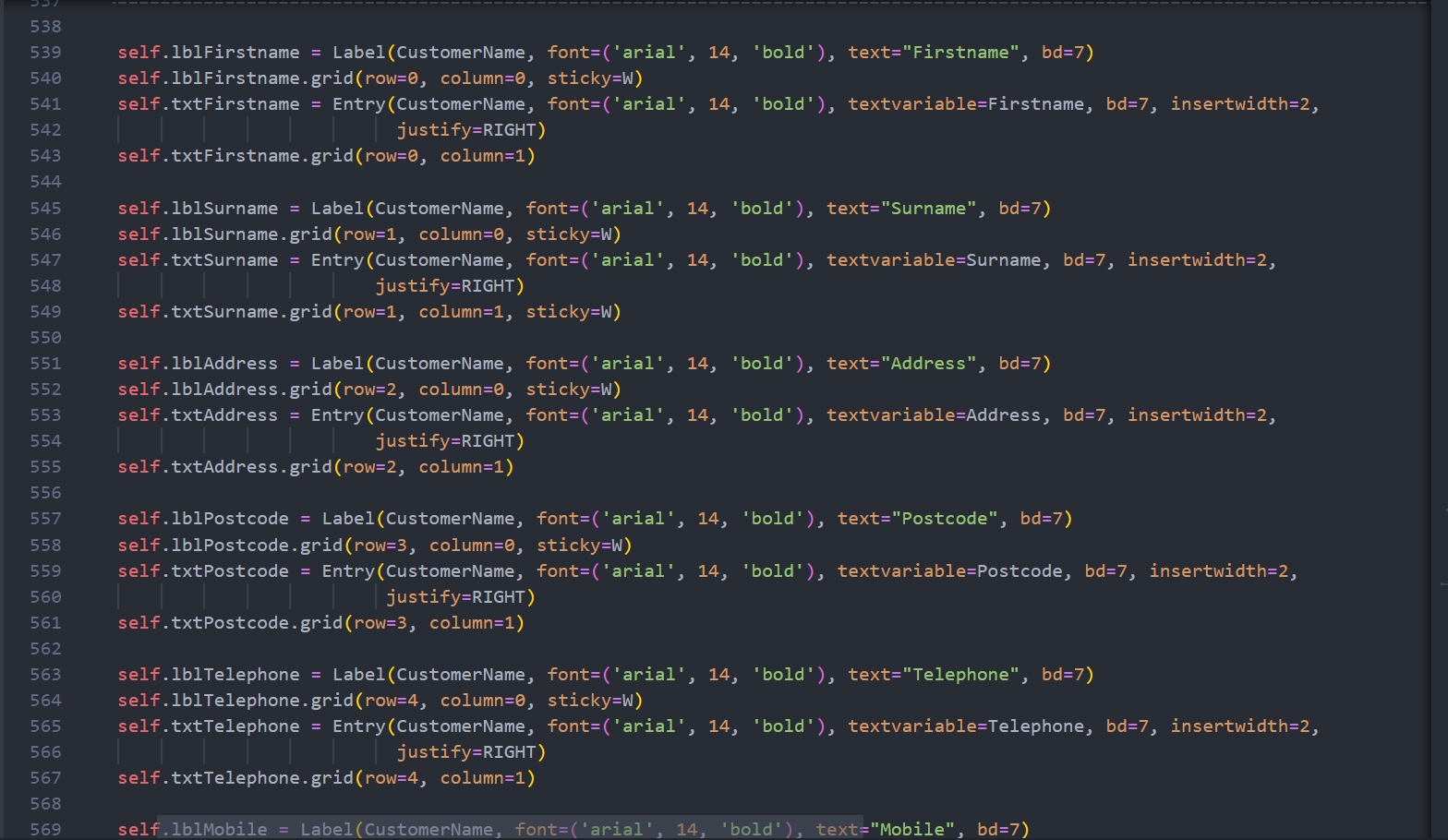
****

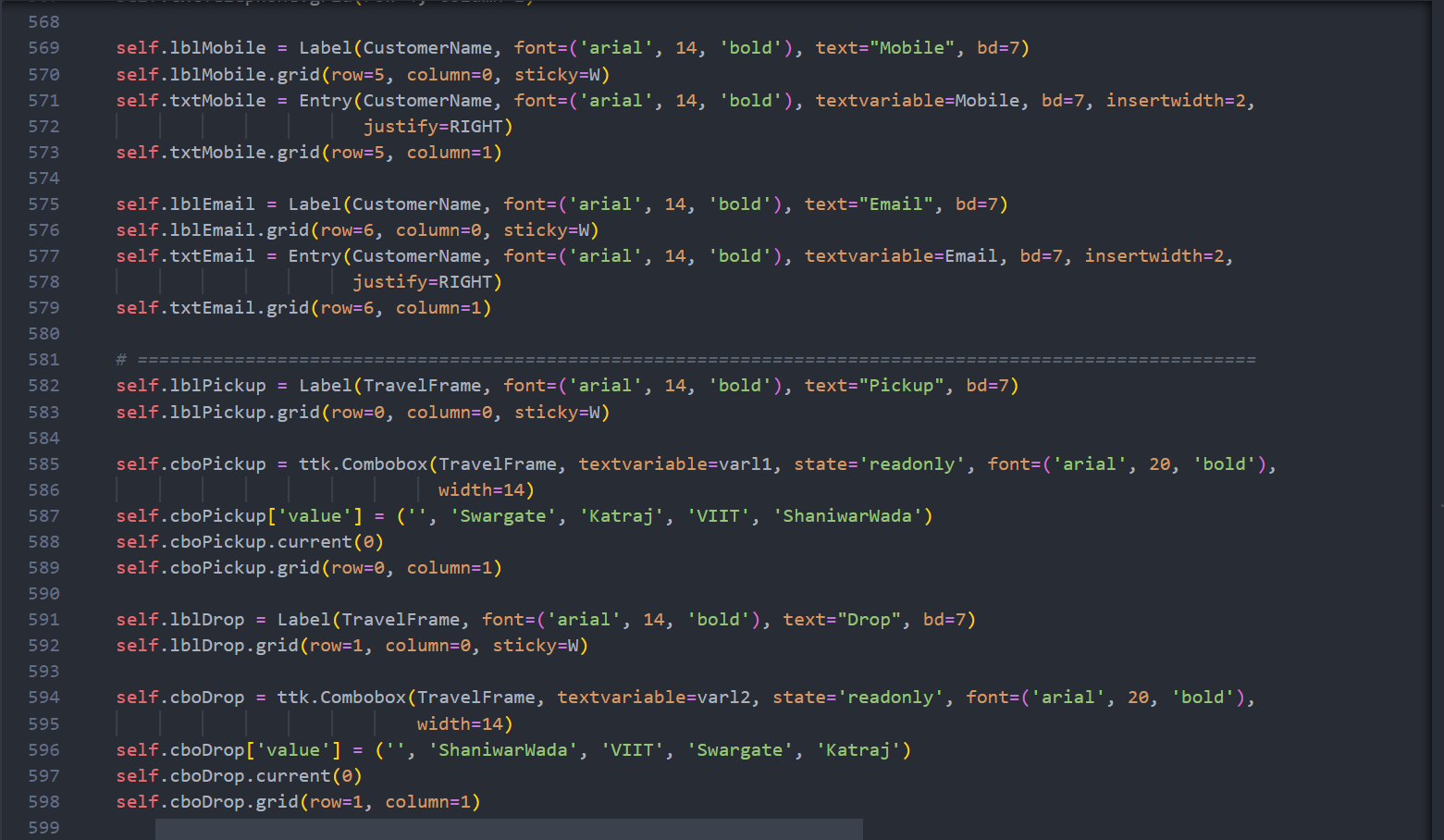
****

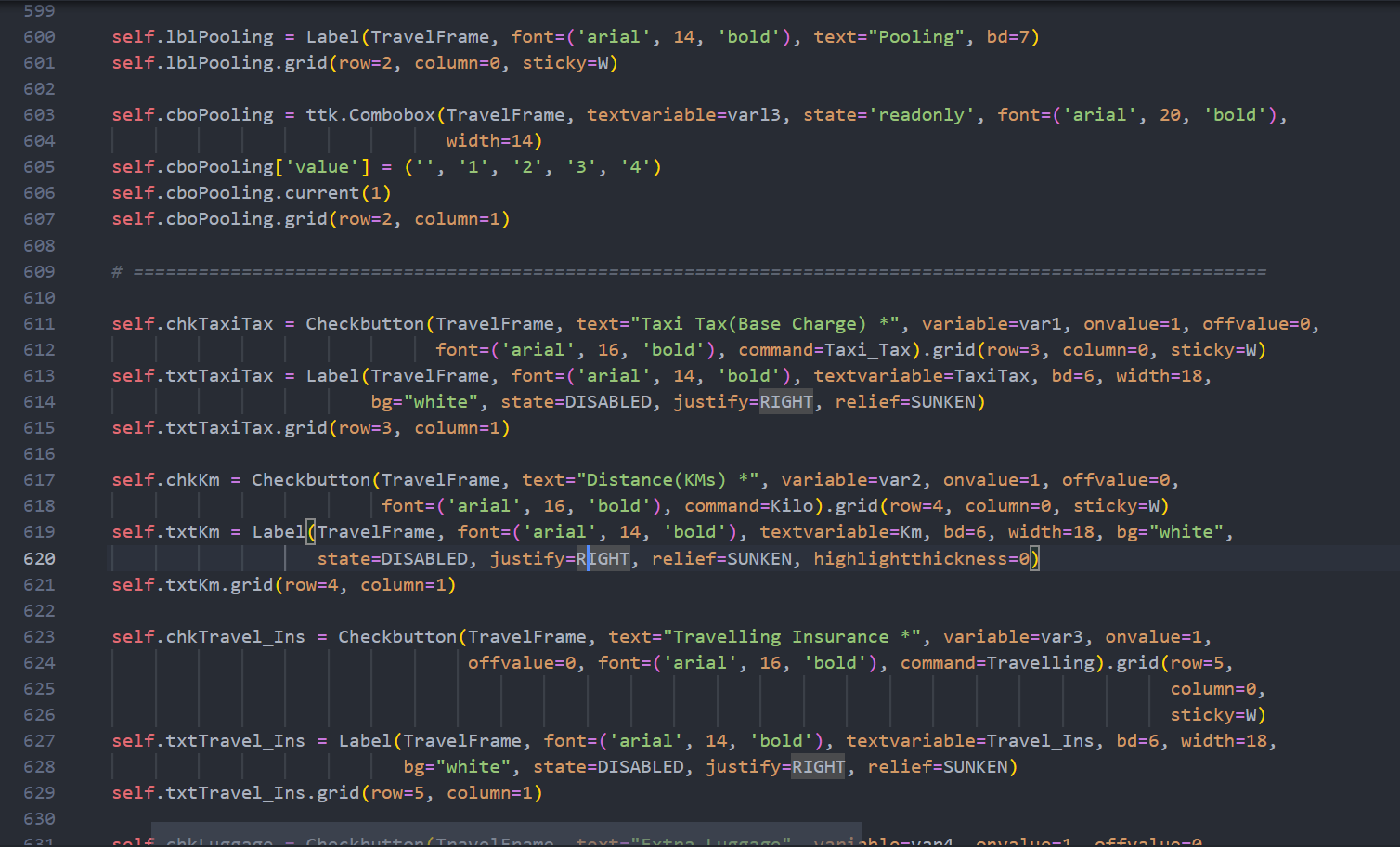
****

****

****

****

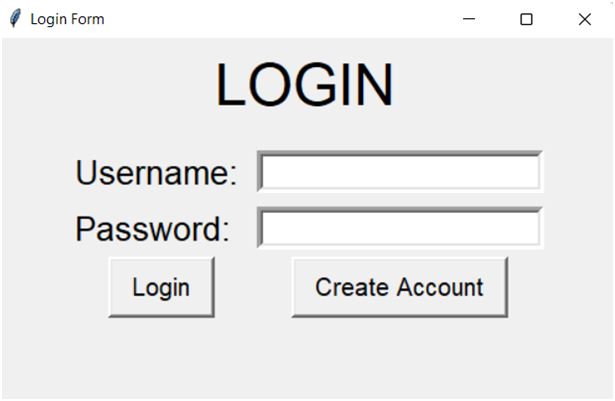
****

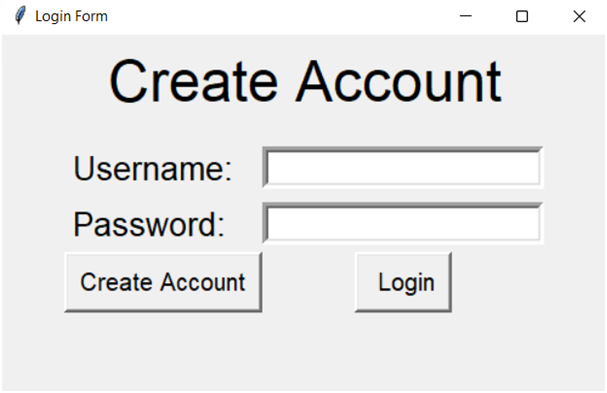
****

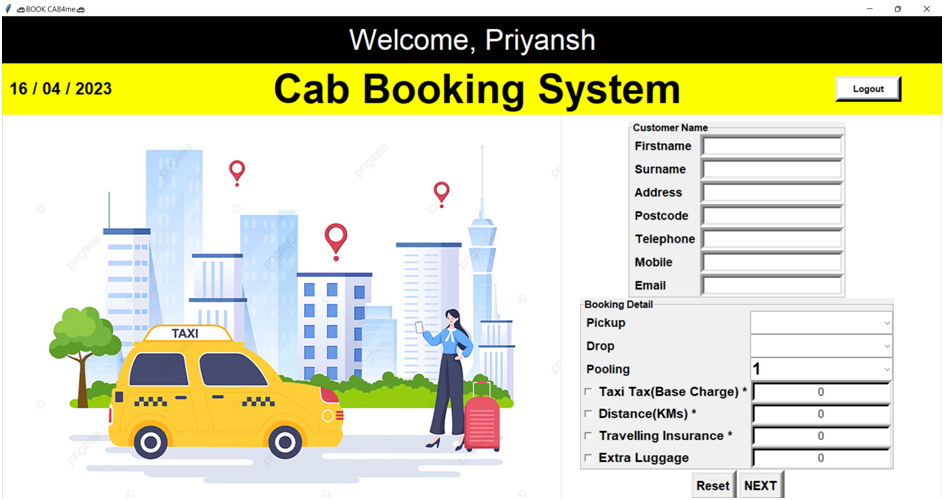
****

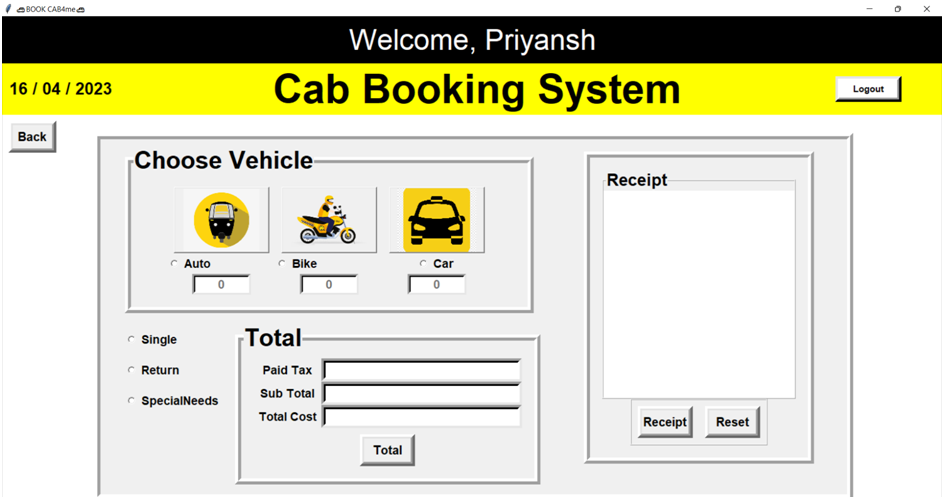
****

**OUTPUT**

****

****

****

****

**Conclusion**

The project that we have developed using the Tkinter library has taught us a lot about the various capabilities of this library and how it can be used to create user-friendly graphical user interfaces (GUIs) for managing complex systems. We have gained significant experience in designing and implementing GUIs using Tkinter, which is a standard GUI library in Python.

The project involved creating a cab management system with features such as booking, payment processing, and generating receipts. Through this project, we have learned how to use various Tkinter widgets such as buttons, labels, entry boxes, and dropdown menus to create a well-designed and functional user interface.

In addition to designing the GUI, we also learned how to integrate it with the underlying logic of the cab management system. This involved implementing functions and methods in the back-end system that can receive user inputs from the GUI and perform the necessary operations such as booking a cab, displaying cab availability, calculating fares, and generating reports.

Overall, this project has provided us with valuable knowledge and skills in using Tkinter to create effective GUIs for managing complex systems. We have learned how to create user-friendly interfaces that are easy to navigate and provide real-time updates to users. The project has also taught us the importance of testing the integration thoroughly to ensure a seamless user experience.

**References**

* [**https://youtu.be/D0xVD8eUk4o**](https://youtu.be/D0xVD8eUk4o)
* [**https://inprogrammer.com/bank-management-system-in-python-using-tkinter/**](https://inprogrammer.com/bank-management-system-in-python-using-tkinter/)
* [**Python Tkinter Tutorial - GeeksforGeeks**](https://www.geeksforgeeks.org/python-tkinter-tutorial/)
* [**Python SQLite – GeeksforGeeks**](https://www.geeksforgeeks.org/python-sqlite/)