

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

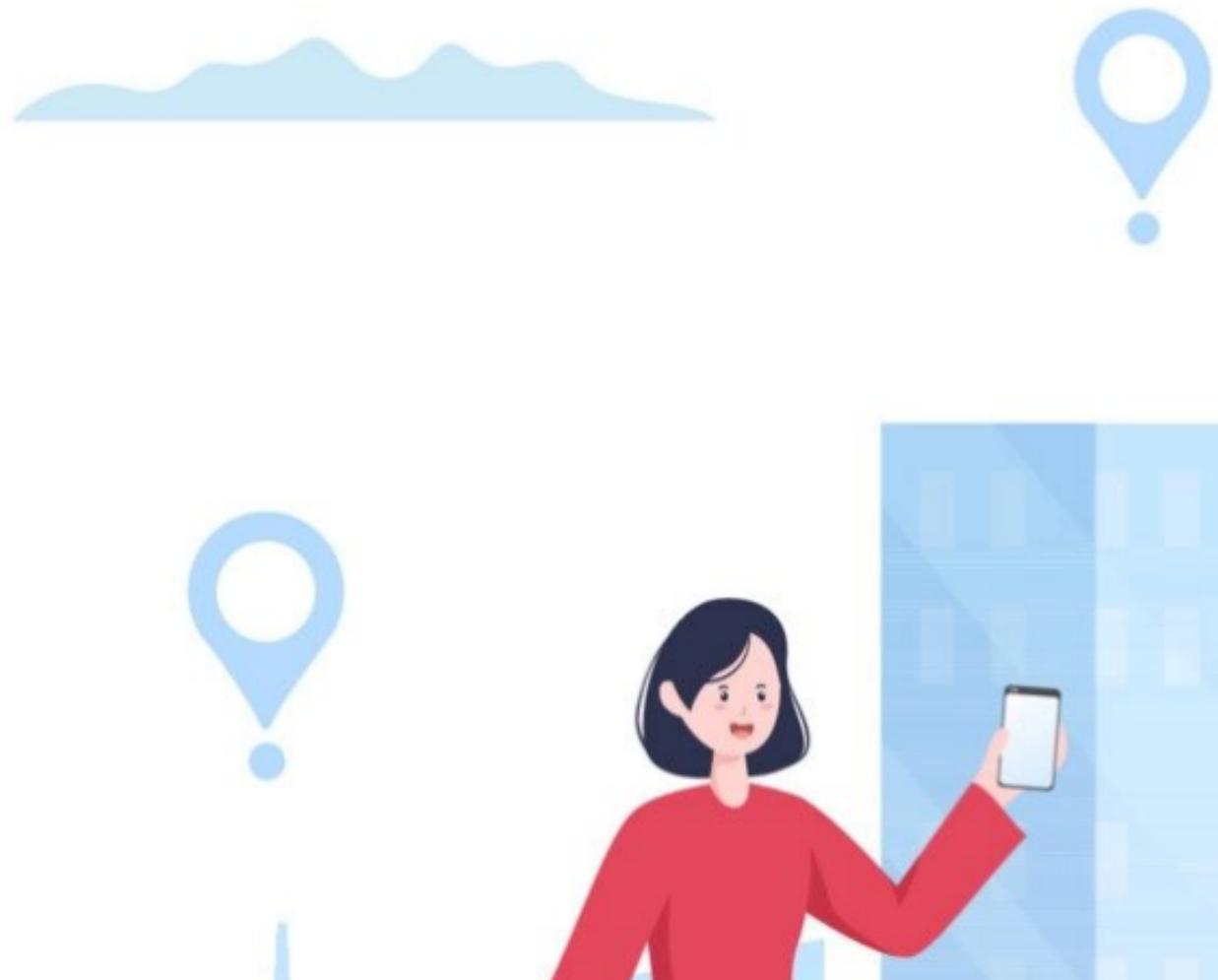
Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

**CIS020-1 – Introduction to Software
Development**
**CIS093-1 – Mathematics and Concepts for
Computational Thinking**
**Assignment 2 – Group / Individual Project –
CaseStudy**
(Taxi Booking System)

Student Name: Baibhav Paudel - 2146504



CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

Table of Contents

Introduction/Overview.....	3
Task Description.....	3
Assumptions about the system :.....	4
Project Plan/Schedule.....	4
Requirements Analysis.....	5
Functional Requirements.....	5
Non-functional Requirements.....	6
Usability Requirements.....	6
Design.....	6
UML Diagrams.....	6
Uses Case Diagram(s).....	7
.....	10
Class Diagram(s).....	12
Database Design.....	15
.....	15
Logical Database Design.....	15
Physical Database Design.....	16
Skeleton Tables (with Primary Keys and Foreign Keys).....	16
Data Dictionary.....	16
User Interface Design.....	20
Algorithm Design.....	25
Implementation.....	28
Testing.....	44
.....	66
Discussion / Reflection / Critical Analysis.....	71
Conclusion.....	72
References.....	72
Appendix.....	73
PACKAGES AND MODULES.....	73
SOURCE CODE OF TURBO TAXI BOOKING SYSTEM.....	74

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

CIS020-1 – Introduction to Software Development - 2021-2022

CIS093-1 – Mathematics and Concepts for Computational Thinking – 2021-2022

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

Introduction/Overview

The Taxi Booking System project, together with the project brief and requirements to execute this assignment, was the final task handed to us. Our primary objective in order to finish Assignment 2 was to create a system for booking taxis. In order to book a cab, a user must first register and sign in using login information (username and password), hence a user registration system and a user login system were created. There must be a way for the system user to reserve a taxi for their journey, thus a booking system was developed. A mechanism for admin control was created to allocate cabs to user booking requests. A driver dashboard was established where he or she could monitor assigned bookings, mark them as completed when the journey was finished, and view forthcoming bookings. A database was made and the system was connected to it using a database connector in order to keep all of the user information and details in the system.

Task Description

The main aim was to create software for a taxi company so that customers could schedule cabs and view their upcoming journeys online using a desktop application's graphical user interface. All the important data regarding user, booking, taxi, and the driver had to be kept in an external database system.

The user must first register as a user with accurate information in order to make a cab reservation. After registering, the user must be able to input the location, date, and time of their pickup as well as the destination of their journey throughout the booking process.

The system must have a capability that allows users to update or cancel their reservations and view forthcoming trips through a graphical user interface. Admin from the business side must

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

be allowed to approve the user's booking once they submit it and choose a taxi driver to complete the booking. Each driver must have a single forthcoming trip.

Assumptions about the system :

- The cost of a journey is determined during the booking process based on the distance between the pickup place and the destination.
- Credit card payments will be added in the driver's account balance.
- The driver is paid immediately once the journey is finished.
- Even after cabs have been assigned, reservations can still be canceled, but only with the admin's permission if an issue occurs.
- Taxi assignments cannot be changed.
- The driver keeps his or her earnings throughout the month, but at the end of the month, he or she is required to pay the corporate service fee which has already been computed and is shown to him in his driver dashboard in order to continue using the system the following month.

Project Plan/Schedule

Week No.	Tasks	Priority
1	→ Will familiarize myself with Python and Tkinter. → Additionally, I'll establish a development environment and list the characteristics that the taxi booking system must have (text editor, version control).	Must
2	→ will utilize Balsamiq to create the user interface for the taxi booking system. → I'll use Tkinter to create a user interface that is comparable to the design.	Must
3	→ I'll also implement the user registration and login features.	Must

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

4	→ I'll put the taxi booking feature into action. → Users will be able to search for and choose a taxi, and I'll also integrate the payment option.	Must
5	→ I'll add the driver management functionality during the fourth week. → The allocated reservations will be visible to drivers.	Must
6	→ I'll check the taxi booking system's operation, troubleshoot any problems, and solve them. → Create the project's documentation.	Should
7	→ Once the project is finished, I will present my institution with the taxi reservation system. → I'll also turn in the project for evaluation.	Must

Requirements Analysis

Functional Requirements

TTBS = Turbo Taxi Booking System

Req. No	Requirement	Priority*
1	A signup system allows customers to register.	MUST
2	To access the system, users must log in.	MUST
3	Request booking from booking form.	MUST
4	User and reservation information is saved to a separate file.	MUST
5	An administrator must be able to add a cab to a user's reservation.	MUST
6	All booking requests will be visible to the administrator.	MUST
7	Bookings must be able to be canceled by admin.	MUST
8	The user must be able to see information about forthcoming bookings.	MUST
9	A booking must be able to be cancelled by the user.	MUST
10	Booking specifics must be editable by the user.	SHOULD

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

11	The ability to log into the system is required for cab drivers.	MUST
12	An forthcoming journey must be visible to taxi drivers.	MUST
13	It should be possible for taxi drivers to see their earnings.	SHOULD

Non-functional Requirements

Req. No	Requirement	Priority*
1	The system's processing speed should be high.	MUST
2	Must be screen-adaptive for devices.	SHOULD
3	System upgrades must be possible.	MUST
4	must be free of crashes or run-time problems.	MUST

Usability Requirements

Req. No	Requirement	Priority*
1	System GUI must be user friendly.	MUST
2	All forms for entering data should be brief, simple to complete, and include entry validation.	MUST

Design

o UML Diagrams

Uses Case Diagrams(s)

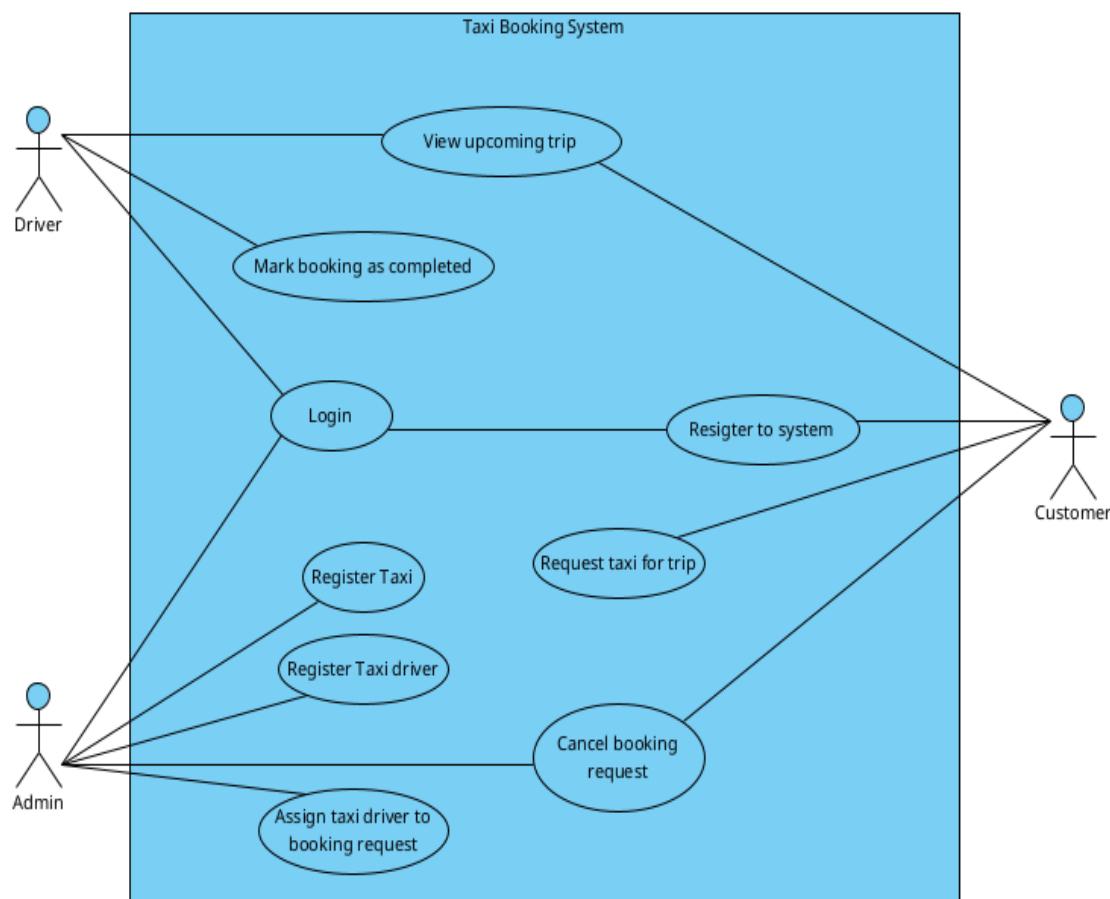


Figure 1: Taxi_Booking_System_Usecase_diagram

(Visual Paradigm,2021)

“A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped”.

Use Case Name: Taxi Booking System

- Description: This use case allows a user to book a taxi from their pickup location to a specified destination.
- Actors: Customer, Admin, Driver

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Conditions:

- The user must register and login into the system.
- The user must confirm with their trip details.
- There is at least one taxi available for booking in the system.

Flow of Events:

1. The user selects the "Book Taxi" option from the user dashboard.
2. The system allows the user to enter their trip detail through a booking form.
3. The user enters their booking details.
4. In order to allocate a cab to a booking, the admin checks the booking request.
5. The system displays a list of available taxis, along with their driver's name, car model, and estimated fare.
6. The admin assigns a taxi from the list.
7. The system confirms the booking and displays the pickup location and estimated arrival time for the taxi.
8. If the user cancels the booking before taxi is assigned, the system will cancel the booking.

Postconditions:

1. The user's booking is recorded in the system.
2. The driver of the selected taxi is notified of the booking and given the pickup location and destination along with all booking details.
3. The user receives updated status that the reservation has been approved and has access to the allocated driver's details.

Activity Diagram(s)

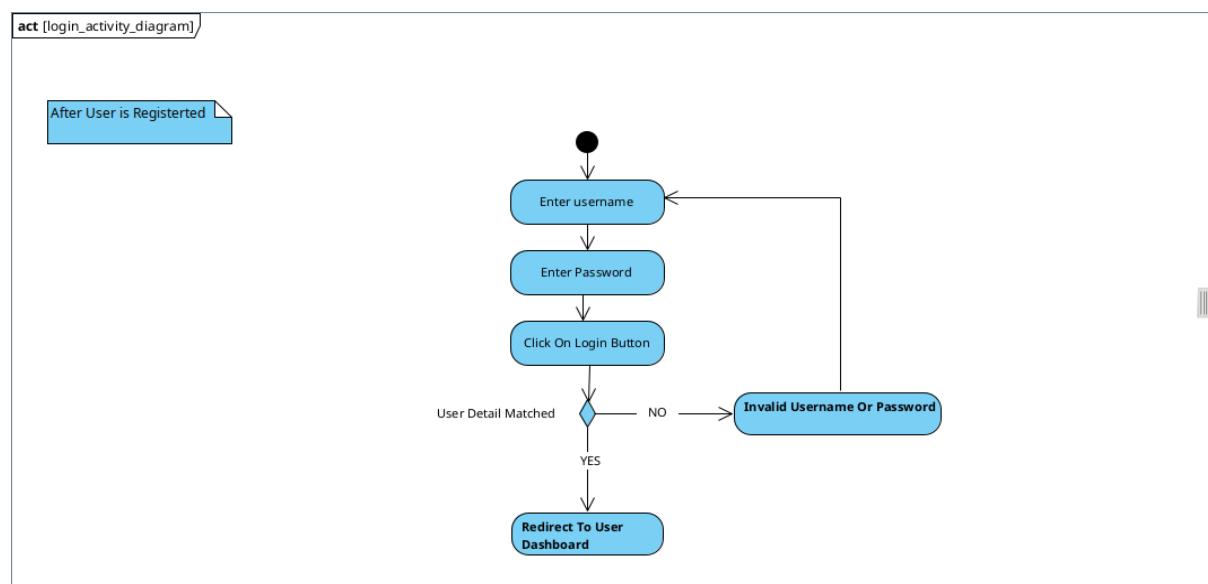


Figure 2: Login_System_Activity_Diagram

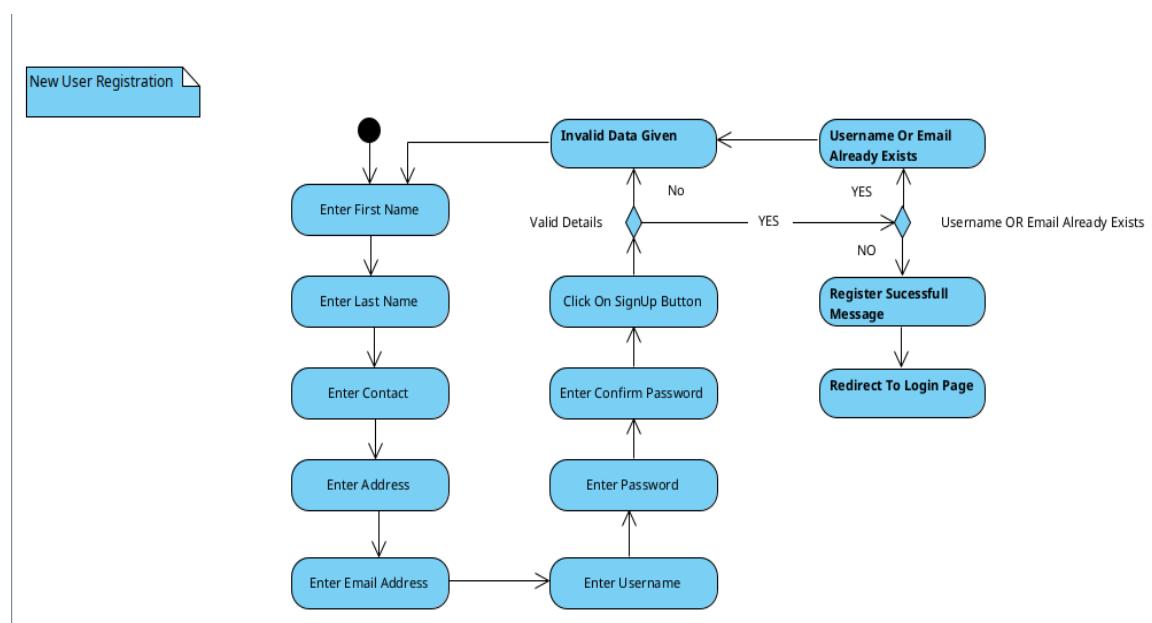


Figure 3: Register_System_Activity_Diagram

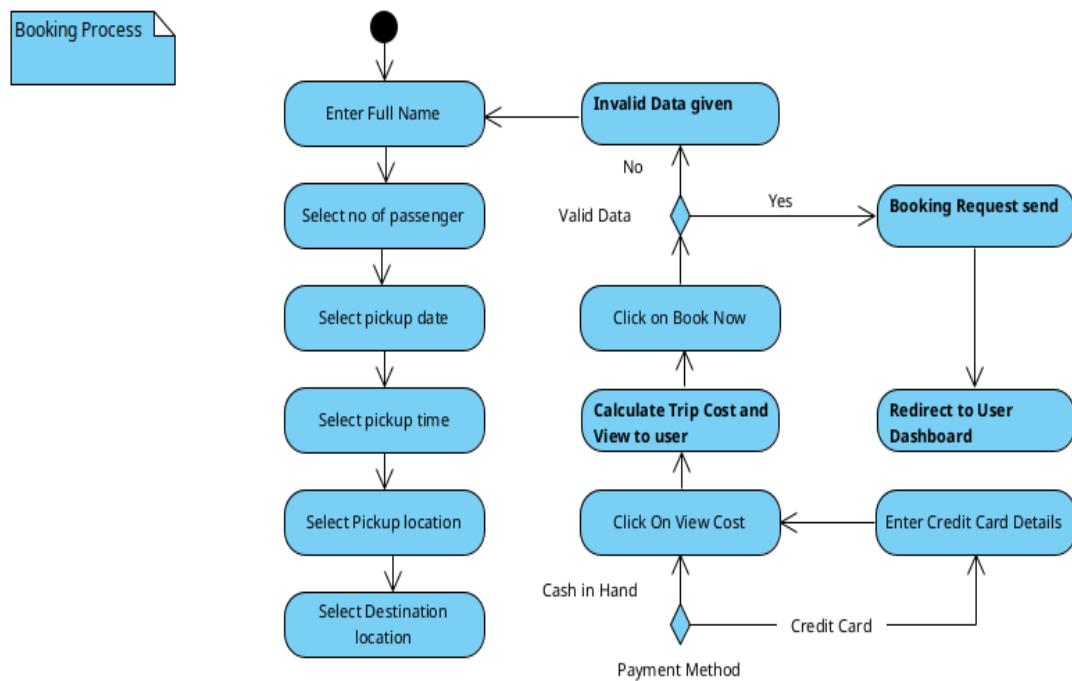


Figure 4: Booking_System_Activity_Diagram

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

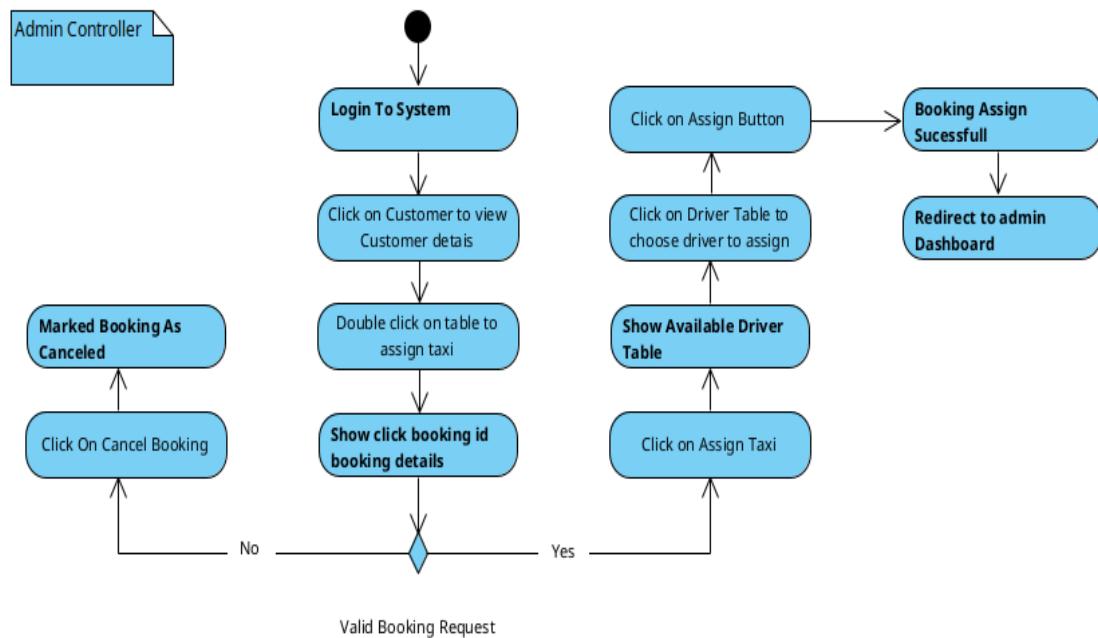


Figure 5: Admin_Control_System_Activity_Diagram

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Class Diagram(s)

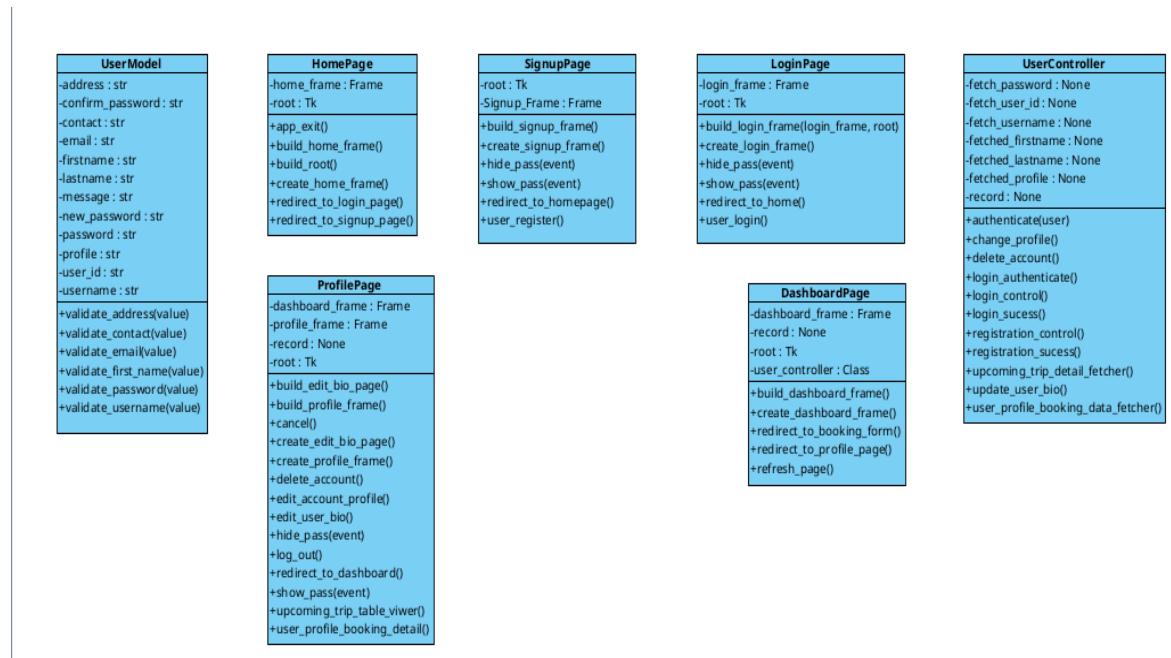


Figure 6: user_class_diagram

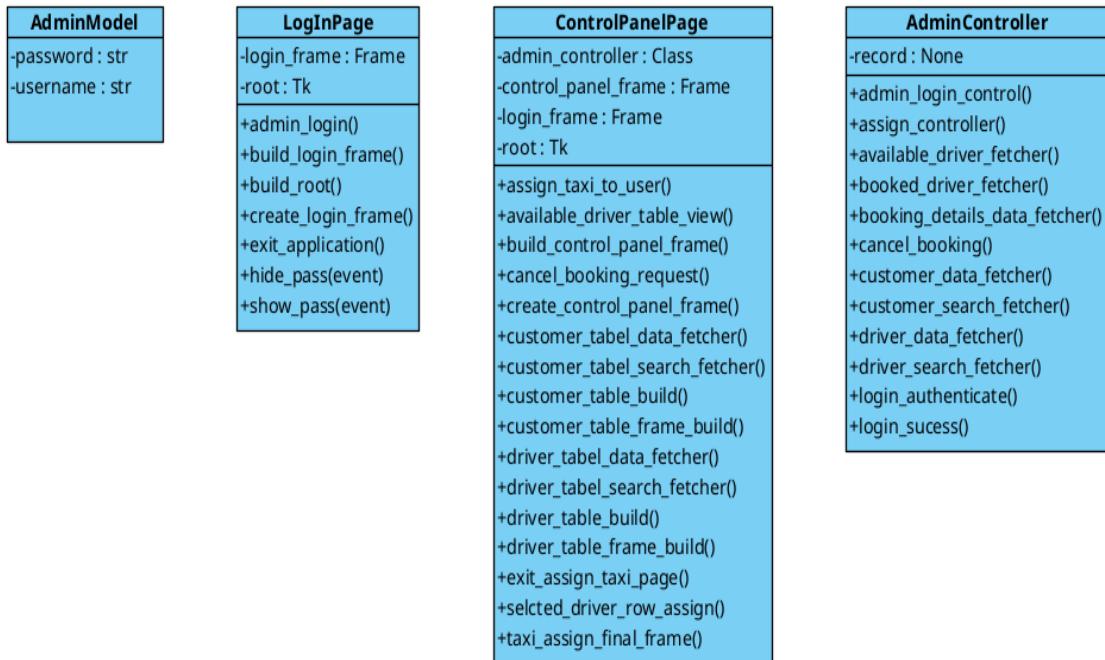


Figure 7: admin_class_diagram

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

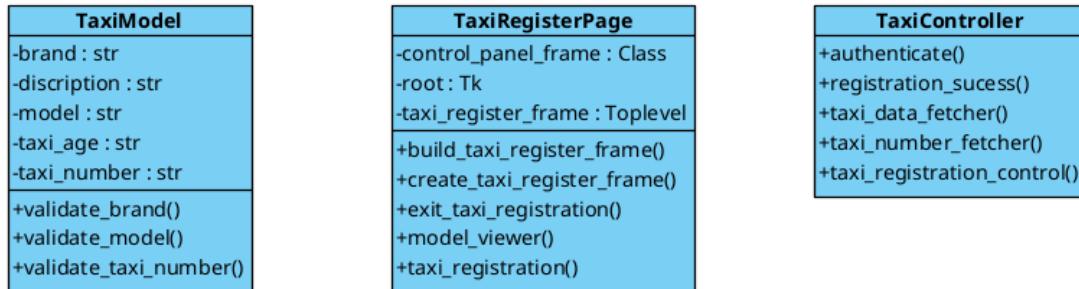


Figure 8: taxi_class_diagram

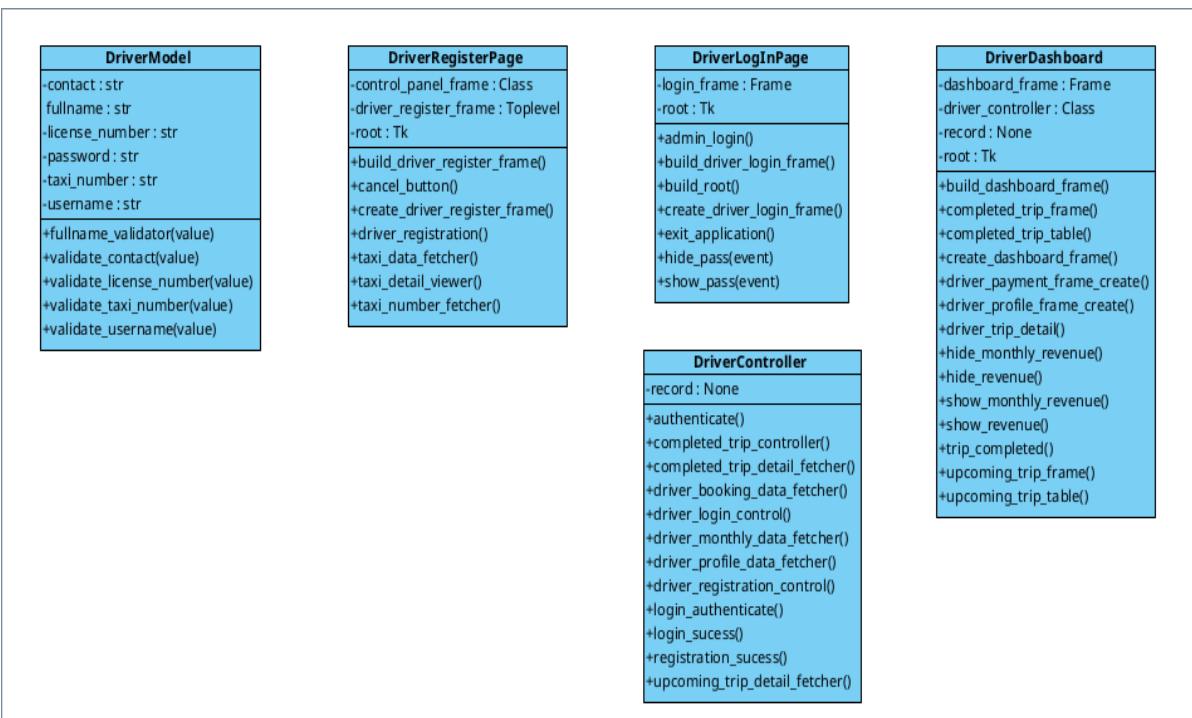


Figure 9: driver_class_diagram

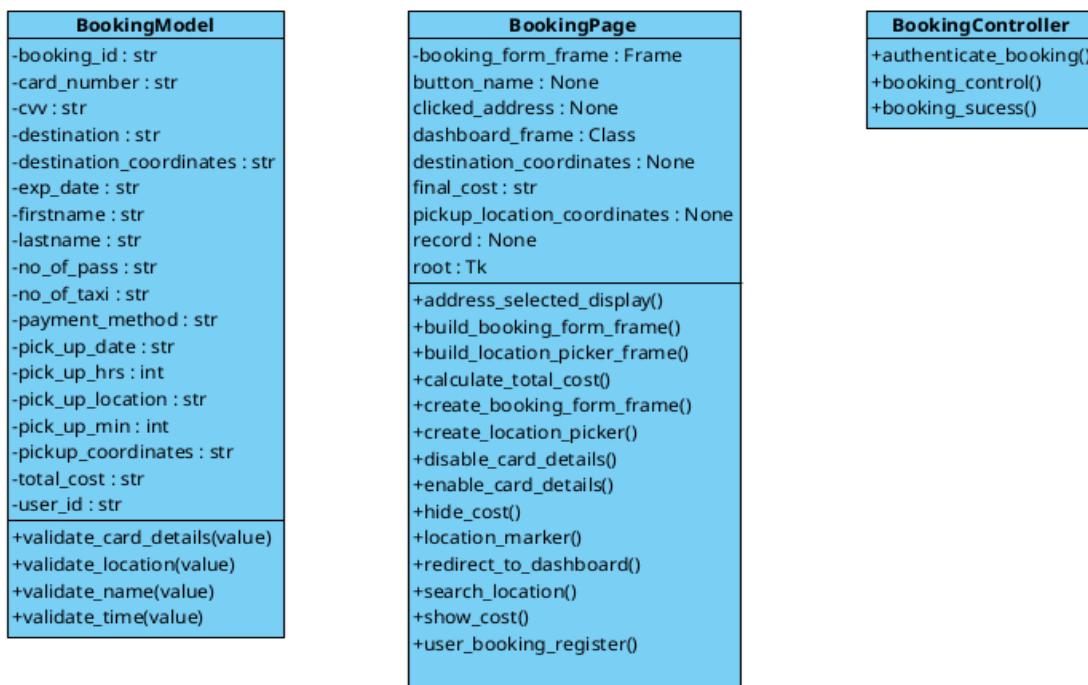


Figure 10: booking_class_diagram

(IBM, 2021)

“Class diagrams are fundamental to the object modeling process and model the static structure of a system”.

Class diagrams are the blueprints of a system or subsystem. They can be used to model the objects that make up the system, display the relationships between the objects, and describe the objects' behaviors and the services they provide.

o Database Design

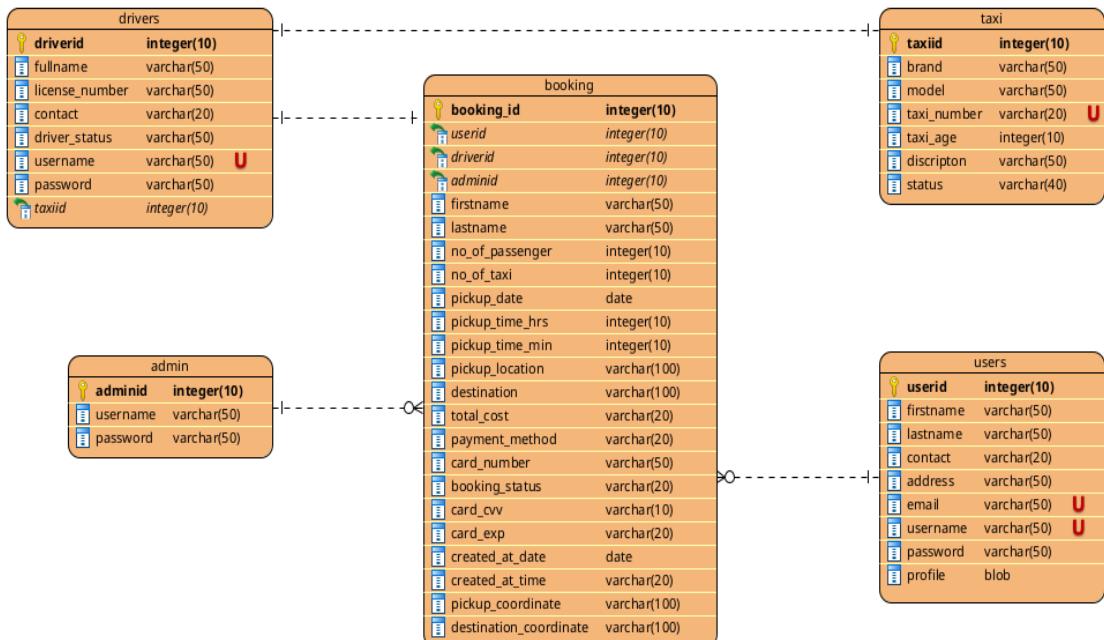


Figure 11: turbo_tb_database_ERD

Logical Database Design

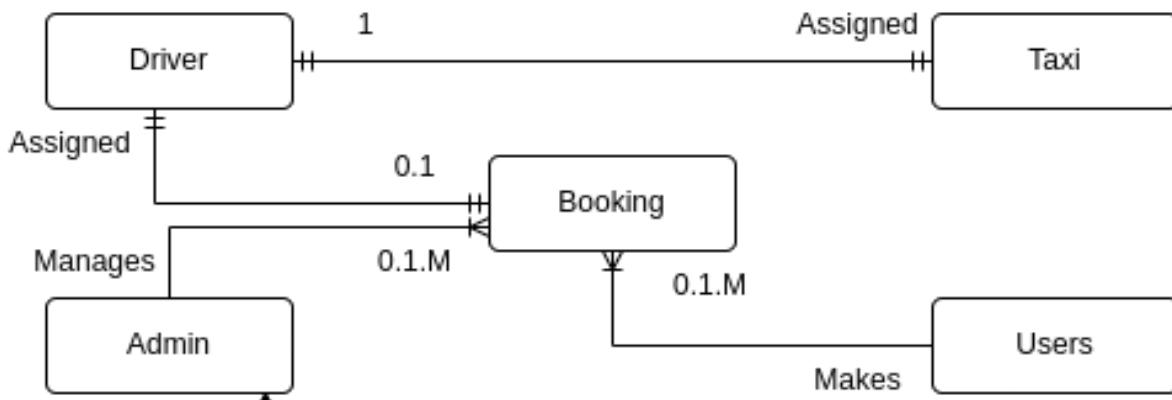


Figure 12: database_ERM

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Physical Database Design

o Skeleton Tables (with Primary Keys and Foreign Keys)

Users(**userid**, firstname, lastname, contact, address, email, username, password, profile)

Booking (**booking_id**, userid*, driverid*, adminid*, firstname, lastname, no_of_passenger, no_of_taxi, pickup_date, pickup_time_hrs, pickup_time_min, pickup_location, destination, total_cost, payment_method, card_number, booking_status, card_cvv, card_exp, created_at_date, created_at_time, pickup_coordinates, destination_coordinates)

Taxi (**taxiid**, brand, model, taxi_number, taxi_age, discription, status)

driver (**driverid**, fullname, license_number, contact, driver_status, username, password, taxiid*)

Admin (**adminid**, username, password)

o Data Dictionary

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Table 1: user_data_dicrionary_table

Users					
Description : Users Details					
Field Name	Data Type	Length	Index	Null	Description
Userid (Primary)	int	10	PK	No	Auto Increment
firstname	varchar	50		No	Firstname of user
lastname	varchar	50		N0	Lastname of user
contact	varchar	20		No	Contact detail of user
address	varchar	50		No	User's location detail
email	varchar	50	U	No	Valid email address of user
username	varchar	50	U	No	Username for login
password	varchar	50		No	Password for personal security
profile	blob			No	User's profile picture

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Table 2: booking_data_dictionary_table

Booking					
Description : Booking details made by user					
Field Name	Data Type	Length	Index	Null	Description
booking_id (Primary)	int	10	PK	No	Auto Increment
userid	int	10	Fk	No	User table foreign key
driverid	int	10	Fk		drivers table foreign key
adminid	int	10	Fk		admin table foreign key
firstname	varchar	50		No	User's first name
lastname	varchar	50		No	User's last name
no_of_passenger	int	50		No	No of passenger in a trip
pickup_date	date	50		No	Date for the pickup
pickup_time_hrs	int	10		No	Time for the pickup HRS
pickup_time_min	int	10		No	Time for the pickup MIN
pickup_location	varchar	100		No	Location to pickup
destination	varchar	100		No	Final destination of trip
total_cost	varchar	20		No	Total cost to be paid
payment_method	varchar	20		No	Method of payment
card_number	varchar	50			Credit card number
card_cvv	varchar	20			Credit card cvv
card_exp	varchar	10			Credit card exp date
created_at_date	date			No	Booking created date
created_at_time	varchar	20		No	Booking created time
pickup_coordinates	varchar	100		No	Pickup location coordinates
destination_coordinates	varchar	100		No	Destination coordinates
booking_status	varchar	20		No	Status of booking request

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Table 3: admin_data_dictionary_table

Admin					
Description : Admin Details					
Field Name	Data Type	Length	Index	Null	Description
adminid (Primary)	int	10	PK	No	Auto Increment
username	varchar	50		No	Username to login to system
password	varchar	50		N0	Account password

Table 4: taxi_data_dictionary_table

Taxi					
Description : Taxi Details					
Field Name	Data Type	Length	Index	Null	Description
taxiid (Primary)	int	10	PK	No	Auto Increment
brand	varchar	50		No	Brand of a taxi
model	varchar	50		N0	Model of a taxi brand
taxi_number	varchar	20		No	Taxi number
taxi_age	int	10		No	Used time of taxi
discription	varchar	50		No	Details about taxi
status	varchar	40		No	Taxi assign status

Table 5: drivers_data_dictionary_table

Drivers					
Description : Drivers Details					
Field Name	Data Type	Length	Index	Null	Description
driverid (Primary)	int	10	PK	No	Auto Increment
fullname	varchar	50		No	Firstname of driver
license_number	varchar	50		N0	Lastname of driver
contact	varchar	20		No	Contact detail of driver
driver_status	varchar	50		No	Driver's assign status
username	varchar	50	U	No	Username for login
password	varchar	50	U	No	Password for personal security
taxiid	int	10	FK	No	Foreign key from taxi table

o User Interface Design

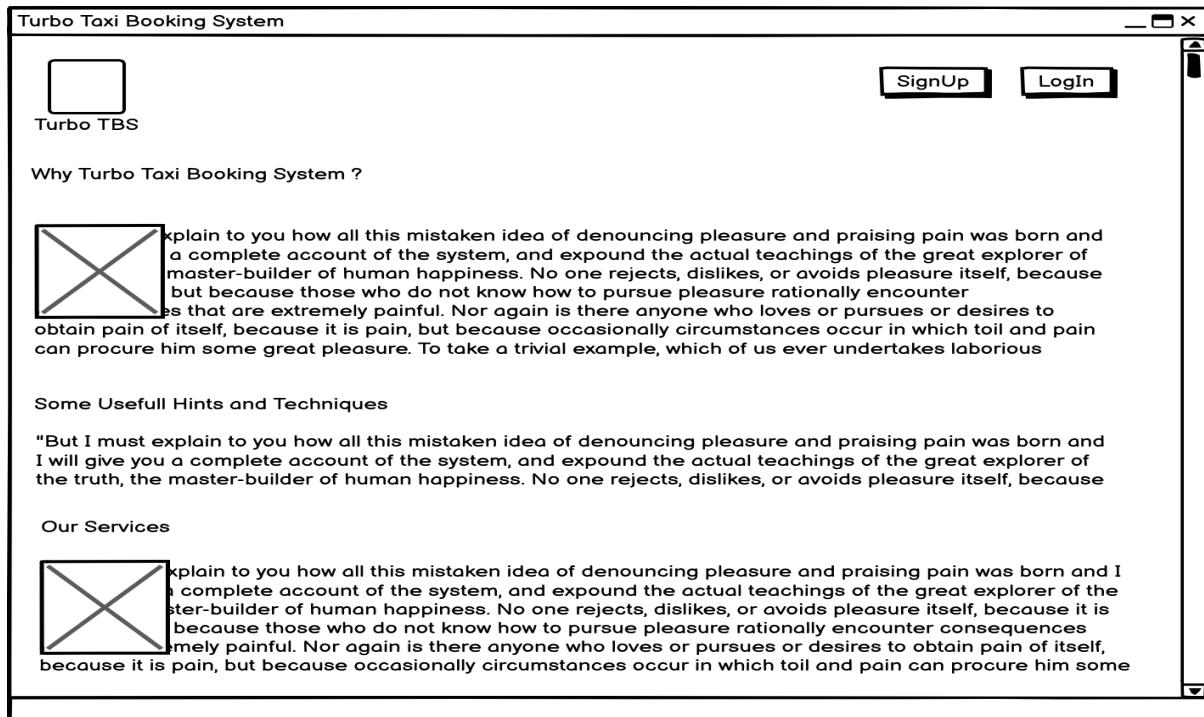


Figure 13: homepage_interface_design

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

Turbo Taxi Booking System



Turbo TBS

Sign Up
The quickest way to book a trip

First Name Last Name
Contact Address
Email Address
UserName Password
Confirm Password
Date Of Birth
Jan q 2002
Gender
 Male Female Other
Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam
rem aperiam, eaque ipsa quae ab illo inventore veritatis

Figure 14: user_signup_interface_design

Turbo Taxi Booking System



Turbo TBS

Log In
Good to see you again

Username or Email
Password
[forgot password ?](#)

[Don't have a account ? Sign Up](#)

[LogIn as Turbo TBS Driver](#)

Figure 15: user_login_interface_design

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

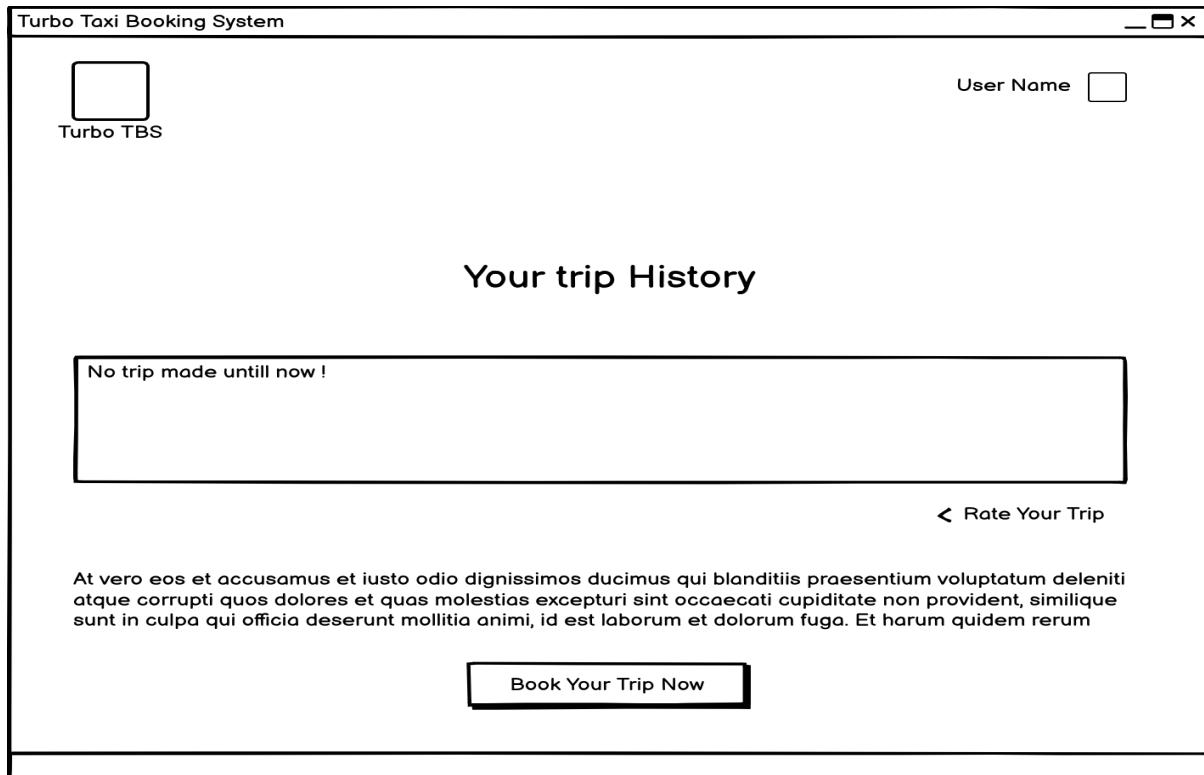


Figure 16: user_dashboard_interface_design

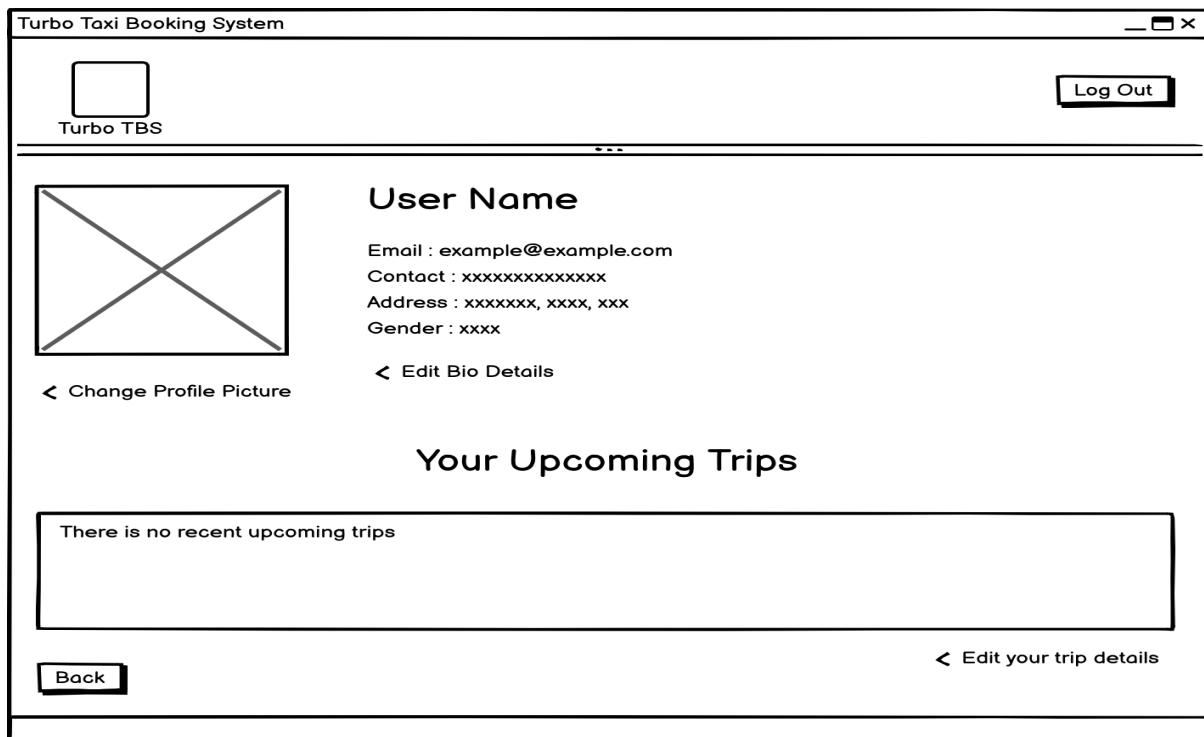


Figure 17: user_profile_interface_design

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

Turbo Taxi Booking System

Book a Taxi Now

On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain

Full Name
 First Name Last Name

No. of passengers 

No. of Taxi required  A Taxi can carry 5 passengers at max

Pickup Date & Time  Hour  : Min  AM 

Pickup Address
 Street Address State Country

Return with us Returning with us will cost price per day

Destination
 Street Address State Country

Payment Method Credit Card  xxxx xxxx xxxx xxxx
 Card Number  xx/xx Exp Date  xxx CVV

On the other hand, we denounce with righteous indignation and dislike men who are so beguile

Total Cost : xxxx

Book Now

Figure 18: booking_form_interface_design

Figure 19: admin_interface_design

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Turbo Taxi Booking System

Assigned Taxi :

User Name
Email : example@example.com
Contact :xxxxxxxxxxxxxx
Address : xxxxxxxx, xxxx, xxx
Gender : xxxx

Booking Details

Full Name : xxxxxxxxxxxxxxxxx
No. of passengers : xx
No. of Taxi required : xx
Pickup Date & Time : xxxx/ xx/ xx
Pickup Address : xxxxxx xxxx xxxx
Return with us : yes/no
Destination : xxxxxxxx xxxx xxxx
Card Number : xxxxxxxxxxxxxxxxxxxx
Exp Date : xx xx CVV : xxx
TOTAL COST : xxxxx

Figure 20: booking_detail_admin_view_interface_design

Turbo Taxi Booking System

Turbo TBS

[Login as Turbo TBS Customer](#)

Log In
Good to see you again

Username or Email
 Password
[forgot password ?](#)

[Don't have a account ? Sign Up.](#)

Figure 21: driver_login_interface_design

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

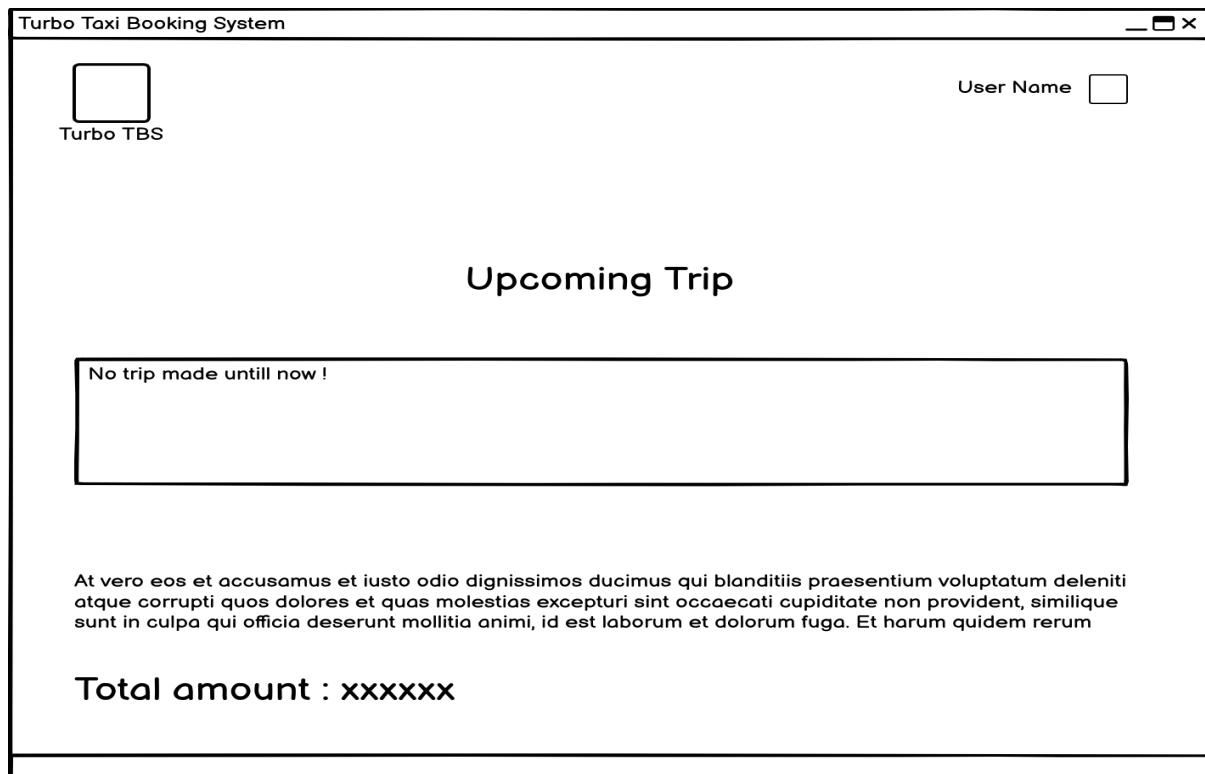


Figure 22: driver_daashboard_interface_design

o Algorithm Design

Flow of the development process

- Flow of the system hierarchy

Taxi booking system

- The system :

Main dashboard : SignUp and SignIn on the main dashboard, with a visible working gallery

1. After clicking SignUp, "SignUp as Customer" will open up and also provide a link for "SignUp as Worker."

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

2. Both forms will require different information and will be saved on different tables of the database.
3. After clicking Sign In, the Sign In as Customer page will appear, along with a link to Sign In as Worker.
4. The same information will be required for both forms. username, email, and password

type of user

i. Customer (The one who does the booking action) :

Working of a customer :

1. Register as a customer
2. Sign up with all the required details.
3. If all the validation is completed, the user will be redirected to the login page.
4. The user will login with the given login email or username and password.
5. The user will be redirected to the user dashboard, which contains a "book now" button, a profile picture, a profile name, and some other details about the system (a hint).
6. Clicking on "Book Now" will redirect the user to the booking form.
7. After entering all of the necessary information, the user can click the Book button, which sends a booking request to the taxi company for approval.
8. This will redirect the user to their dashboard.
9. By clicking on the profile picture or the username on the dashboard, users will be redirected to the profile page.
10. Profile page will include all the previous booking history and ongoing booking information
11. The profile page will allow you to edit the booking details as well as your personal information.
12. The GUI must be good, as each function should change the pages for each different function.
13. Bookings can be canceled from the profile page until the trip date is more than one day away.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

14. The completed trip can be seen, and the driver can be rated afterward.

ii . Admin (The one who assign the taxi and driver to the request trip)

Working of a Admin :

1. Log in as an administrator with the provided username and password.
2. Admin should be redirected to the admin dashboard.
3. The portal should contain two options (a taxi driver control panel and a customer booking request control panel).
4. Clicking on "Taxi Driver" (this will bring up a list of all registered taxi drivers as well as all taxi drivers who wanted to register with valid information).
5. It will also provide all the information about the taxi driver after clicking the taxi driver's name on an ID-sized card.
6. It will give the administrator full authority to accept or decline a registration request for a taxi driver after documentation review.
7. Admin will be able to go back to the previous page where they can select between the driver and the customer control panel with a back button.
8. Clicking on "customer" (it will provide a list of all the customers, and you can view their details by clicking their name, which will pop up an ID-type UI) 9. You can view the booking request made by the customer.
10. You can see the drivers that are already booked for a trip and the available taxi drivers that are suitable for the trip.
11. To accept a trip, click on a requested customer name, an Id type UI pops up, and then click on assign taxi after checking the requested booking detail (number of taxi will allow you to input number of taxi vehicle number with add button).
12. The "booking for Customer XXXXX has been completed" message will be shown, and then the pop-up will close on its own.
13. Admin have full authority to accept or decline an offer if it seems reasonable.
14. Admin can ban a registered user or a driver if they are found breaking the rules of an organization.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

iii . Driver (The One who are assigned for customer to complete the trip)

Working of Driver :

1. Sign up as a driver
2. Sign up with all the details (full name, address, contact information, gender, date of birth, username, email address, experience, upload of a license photo up to 25 MB, vehicle number, vehicle registration number).
3. If all the validation is completed, the user will be redirected to the login page.
4. The user will login with the given login email or username and password.
5. The user will be redirected to the user dashboard.
6. The driver dashboard contains total income, upcoming trips, some information for easy use, Profile Page
7. By clicking on the profile picture or the username on the dashboard, the user will be redirected to the profile page.
8. The profile page will include all the previous booking history and personal details.
9. After the trip is assigned to the driver, the driver must take the trip as the company requests, but if the driver is not available, the driver can change the status from "available" to "not available."
10. Changing the status will affect the admin page, so the filter won't show the driver as an available driver, and assigning the tax trip to such a driver is not allowed.

Implementation

Using a Tkinter-based GUI, users of the Turbo Cab Booking System can request a taxi and calculating fares for the journey. Users may transmit or request data via a graphical user interface and a client server infrastructure, which allows for easy data sharing. With only one click, the user may use an external database system rather than using user storage to save data in the booking system. In addition, drivers have access to a list of their allocated travels, which are marked as finished when finished.

The planning and prototype phases of the development process were the first. All the criteria were initially put down after creating a weekly timetable. Visual Paradigm was used to create ER diagrams for the database design, and Balsamiq was utilized as a prototyping tool for the GUI concept design.

In order to learn and test PostgreSQL as we all know Postgres is a powerful and reliable database that is well-suited to a wide range of applications, Data was stored in a PostgreSQL database, and psycopg2 was utilized as a connector between the system and the database.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
import psycopg2

class Turbo_db:
    message = ""

    @staticmethod
    def turbo_connection():
        try:
            conn = psycopg2.connect(
                host="localhost",
                dbname="turbo_db",
                user="increi",
                password="fastrack",
                port=5432,
            )
            return conn
        except Exception as error:
            Turbo_db.message = error
```

Figure 23: database_connection

For frontend development, Tkinter, a built-in Python module, was used. The non-visual, or backend, portion of the code was likewise created in Python utilizing the MVC design pattern, where the model manages all of a module's important getter and setter and carries out validation. The frontend, or visual representation, of a program and function that provides data to the model layer's getter for validation is managed by the view layer of a project and The function that establishes a connection to the database system and puts data in the appropriate table according to the SQL command is called by the view layer from controller layer when the validation is finished.

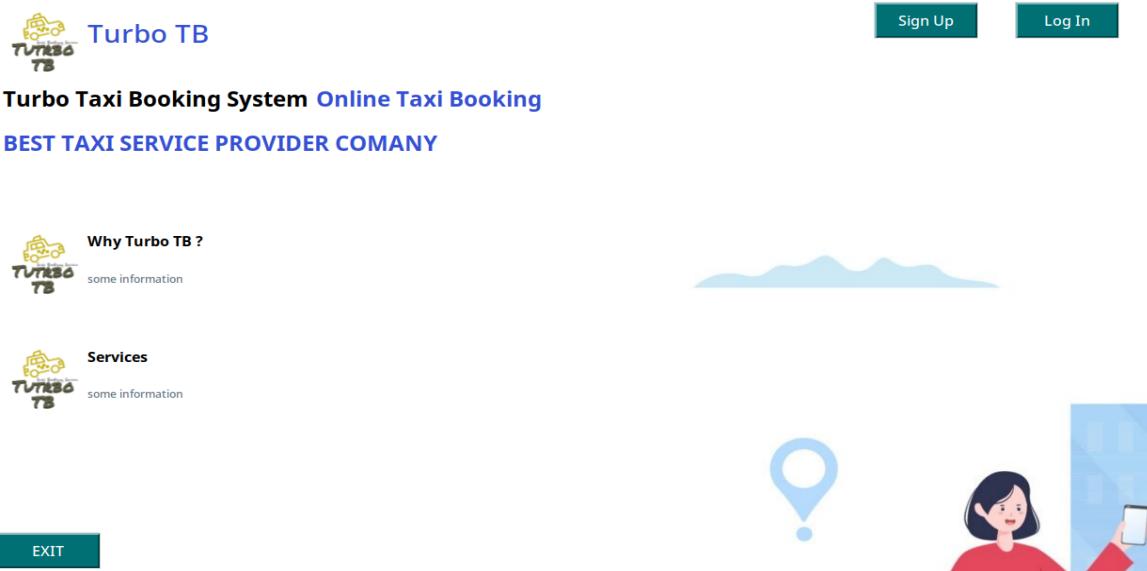


Figure 24: home_page

Home page of taxi booking system .

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

The image shows the signup page for the Turbo TB travel application. The page has a header "Welcome To Turbo TB" and a tagline "Travel For Your Heart". It features a central illustration of a woman holding a smartphone. On either side of the illustration are decorative elements: clouds on the left and location pins on the right. Below the illustration are two buttons: a green "Back" button on the left and a teal "Sign Up" button on the right. The form itself consists of several input fields: First Name, Last Name, Contact, Address, Email Address, User Name, Password, and Confirm Password. Each field is represented by a grey rectangular input box.

First Name	Last Name
Contact	Address
Email Address	
User Name	Password
Confirm Password	oo

Sign Up

Figure 25: signup_page

Signup page for the user where unregistered user register for the first time.

This image shows the same signup page as Figure 25, but with actual data entered into the fields. The "First Name" field contains "Baibhav", "Last Name" contains "Paudel", "Contact" contains "9845145785", "Address" contains "Dhangadhi, Kailali", "Email Address" contains "pbaibhav@gmail.com", "User Name" contains "inci", "Password" contains "*****", and "Confirm Password" also contains "*****". The "Sign Up" button remains at the bottom right.

First Name	Last Name
Baibhav	Paudel
Contact	Address
9845145785	Dhangadhi, Kailali
Email Address	
pbaibhav@gmail.com	
User Name	Password
inci	*****
*****	oo

Sign Up

Figure 26: signup_page_with_data

Registering with valid data.

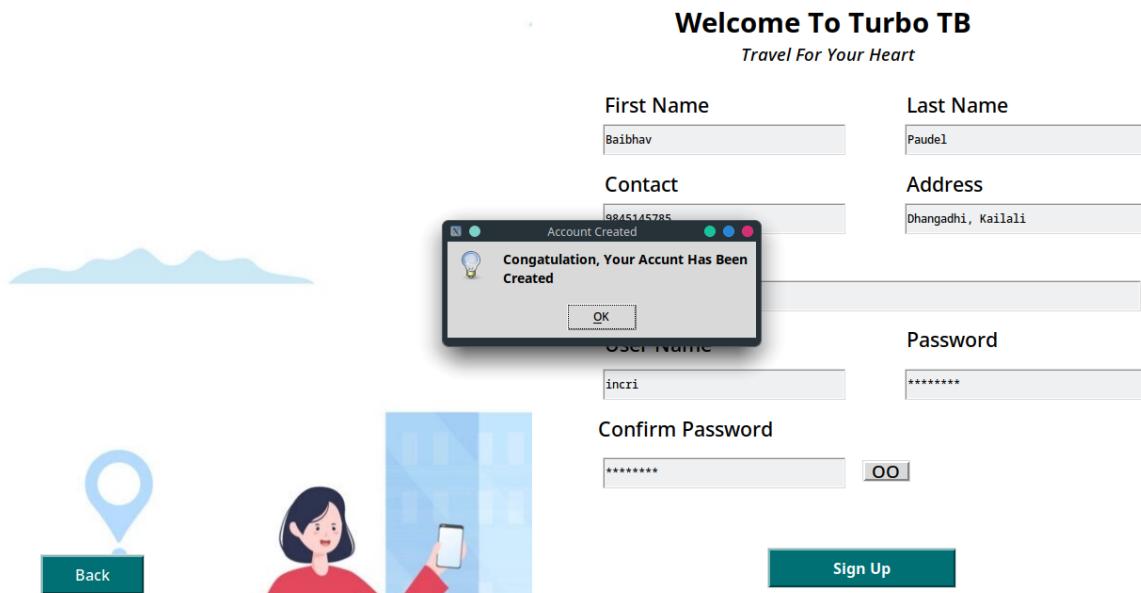


Figure 27: after_clicking_signup_button

After entering proper information and pressing the signup button, a messagebox with a successful account creation message appears.

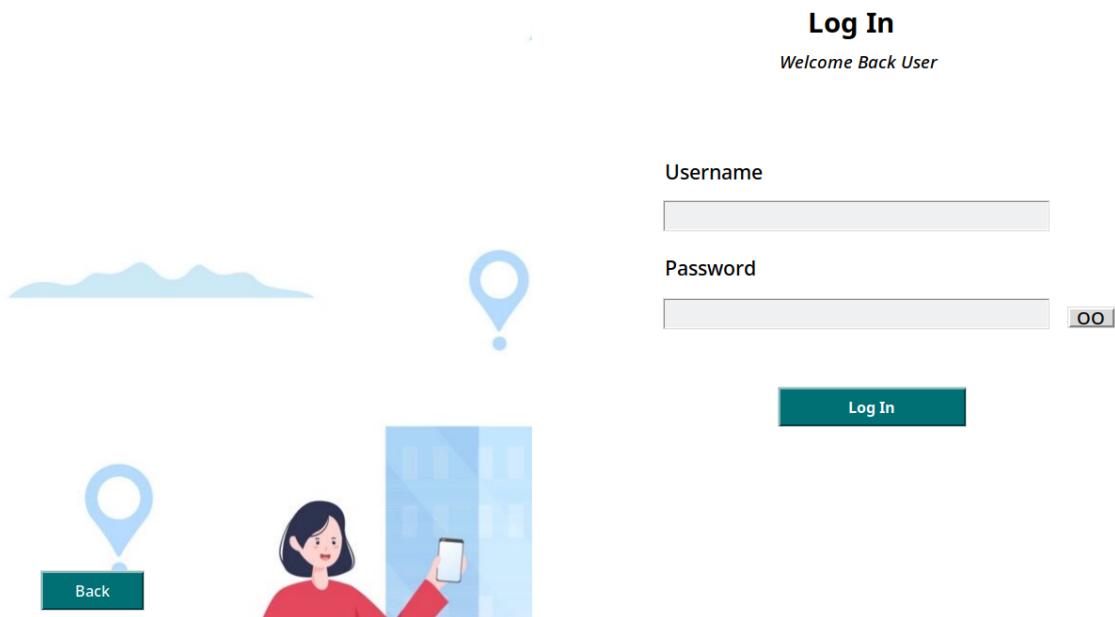


Figure 28: login_page

The login page is used to log into the system if the user is already registered.

CIS020-1 – Introduction to Software Development

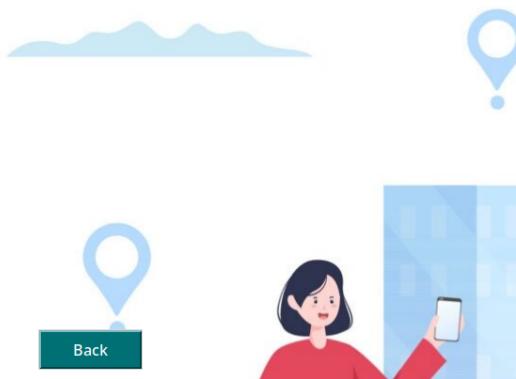
CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Log In
Welcome Back User



Username

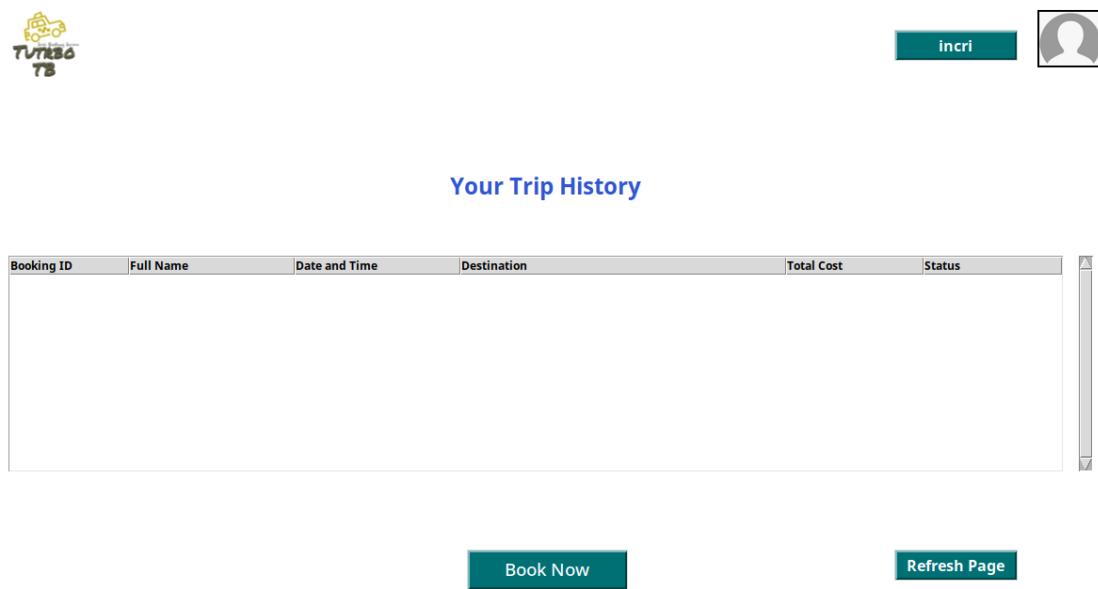
Password

Log In

Back

Figure 29: login_page_with_data

Logging in with valid registered data.



Your Trip History

Booking ID	Full Name	Date and Time	Destination	Total Cost	Status
No trips found					

Book Now **Refresh Page**

Figure 30: user_dashboard

Valid login information enables user dashboard redirection.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

The screenshot shows a user profile page. At the top right is a 'Log Out' button. On the left is a placeholder profile picture icon. The user's name is listed as 'incr1'. Below the name are three lines of personal information: 'Email : pbaibhav@gmail.com', 'Phone no : 9845145785', and 'Address : Dhangadhi, Kailali'. Below this information are two buttons: 'Edit Profile >' and 'Edit Bio >'. At the bottom of the page is a section titled 'Your Upcoming Trips' with a table header showing columns for 'Booking ID', 'Full Name', 'Date and Time', 'Destination', 'Total Cost', and 'Status'. A large empty table body follows. At the very bottom left is a 'Back' button.

Figure 31: profile_page

Users may access their profile pages, which include personal information and information about planned trips.

The screenshot shows an 'Admin Log In' page. It features a decorative background with blue clouds and a location pin icon. On the left, there is a small illustration of a woman holding a smartphone. On the right, there are input fields for 'Username' (containing 'admin') and 'Password' (containing '*****'). To the right of the password field is a 'Log In' button. At the bottom left is an 'Exit' button. The overall design is clean and modern.

Figure 32: admin_login

Admin can redirect to the Admin dashboard using valid login information.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Control Panel

The screenshot shows a web-based control panel titled "Control Panel". At the top right are three buttons: "Driver Register", "Customer Register", and "Taxi Register". Below the buttons is a section titled "Customer Table" with a search bar and a dropdown menu set to "Pending". A table displays a single row of data:

User ID	Booking ID	Created At	Full Name	Contact	Address	Email	Booking Status
1	1	2023-01-08	Baibhav Paudel	9845145785	Dhangadhi, Kailali	pbaibhav@gmail.com	Pending

At the bottom left are two buttons: "Customer" and "Driver".

Figure 33: admin_dashboard

Admin dashboard with search and data filter functionality for seeing booking request information.

The screenshot shows a modal dialog box titled "Taxi Register" overlaid on the main "Control Panel". The dialog contains fields for "Brand" (dropdown), "Model" (dropdown), "Taxi Number" (text input), "Age" (dropdown), and "Description" (text area). At the bottom are "Cancel" and "Register" buttons. In the background, the main panel shows the "Customer Table" with one entry and the "Taxi Register" button at the top right.

Figure 34: register_taxi

Additionally, the admin dashboard enables admin to register new taxis with relevant information.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

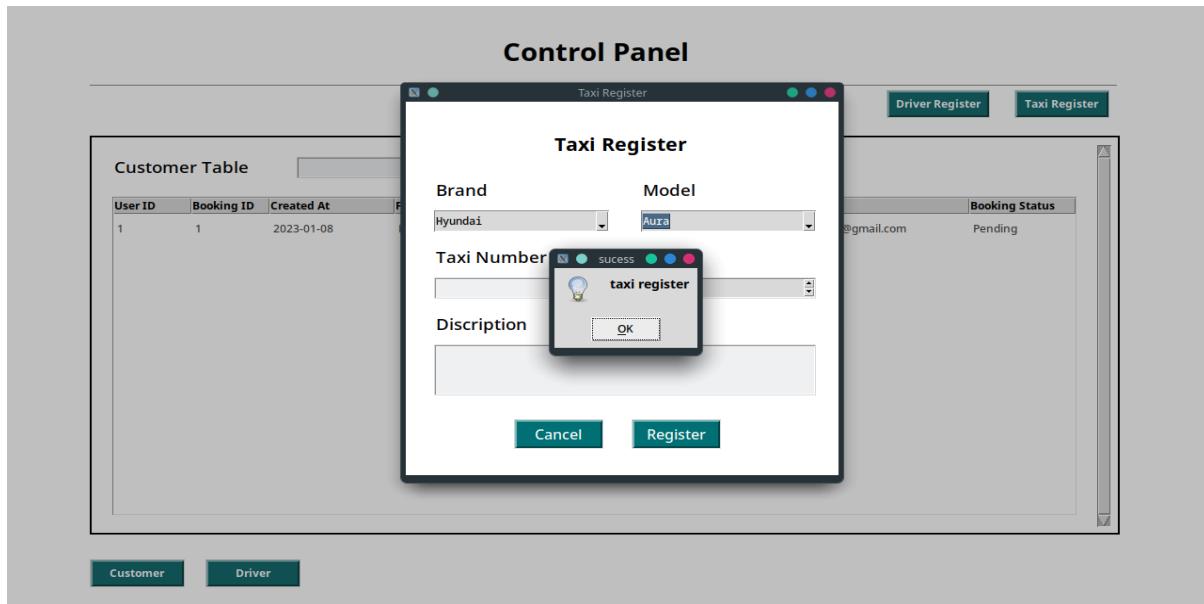


Figure 35: after_clicking_on_taxi_register

After taxi is registered message box pop up and inserted data get deleted but the toplevel remain there allowing admin to add more taxi detail without extra clicking work.

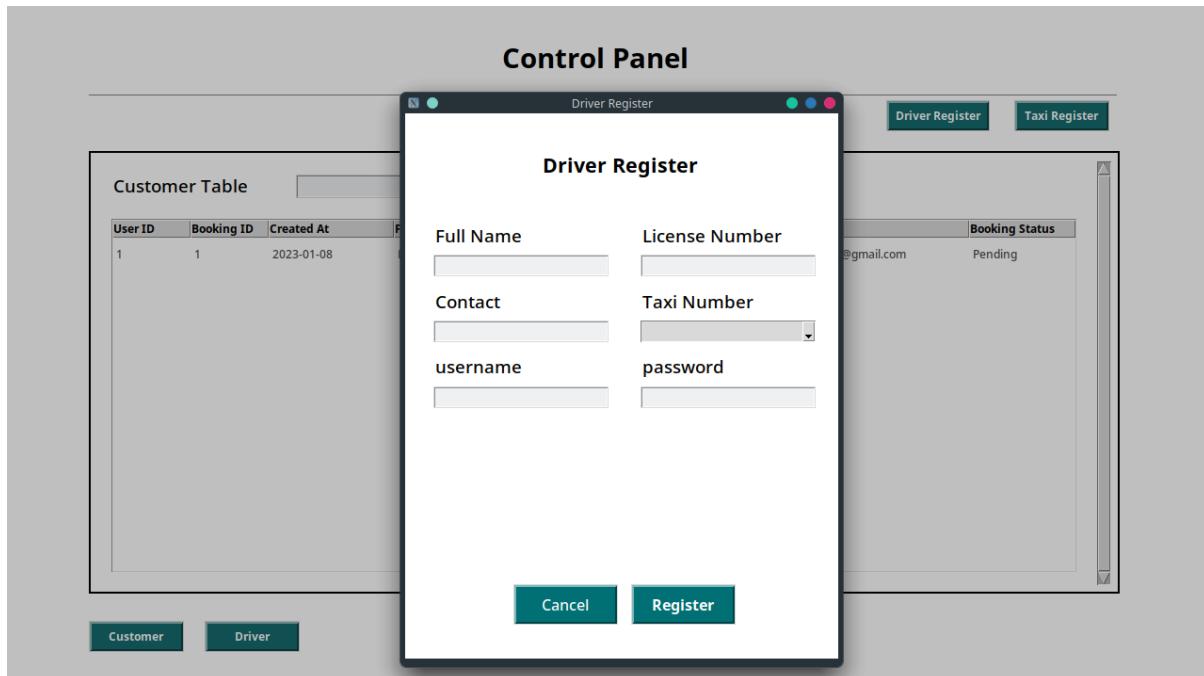


Figure 36: driver_register

A new driver's personal information can be registered on the admin dashboard as well.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

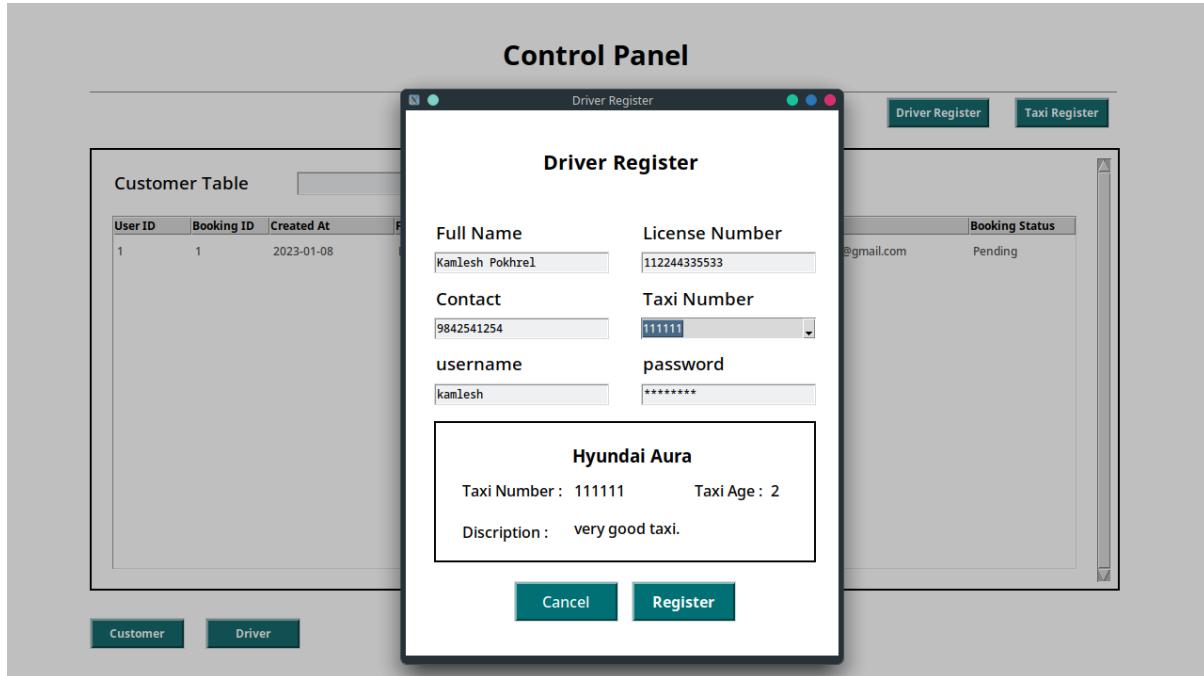


Figure 37: driver_register_with_data

Admin must select a taxi number to provide to the driver of the taxi by using a tkinter combobox, which displays all the available taxi numbers to assign to the driver. After selecting on a taxi number, the details of the taxi are displayed in the frame below.

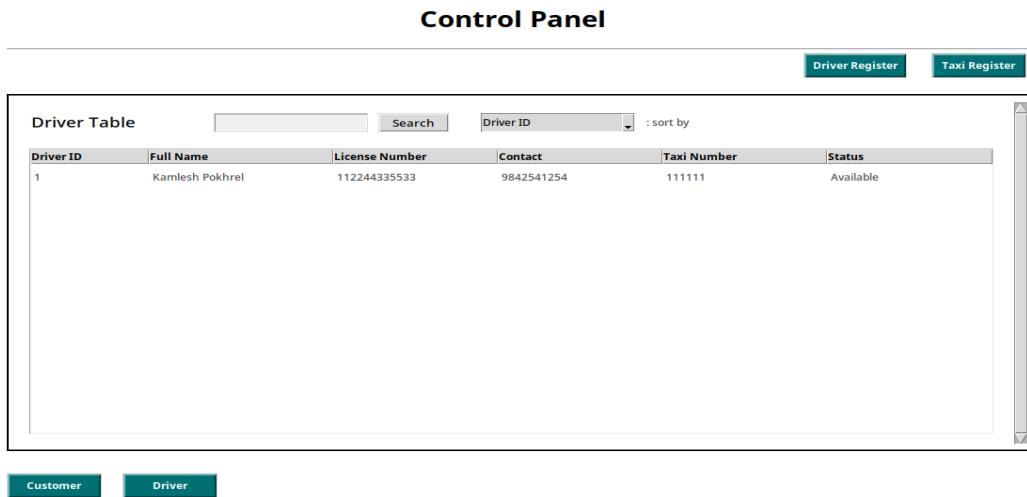


Figure 38: after_driver_is_registered

Administrators may view registered drivers and their status on the driver table after registering them.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

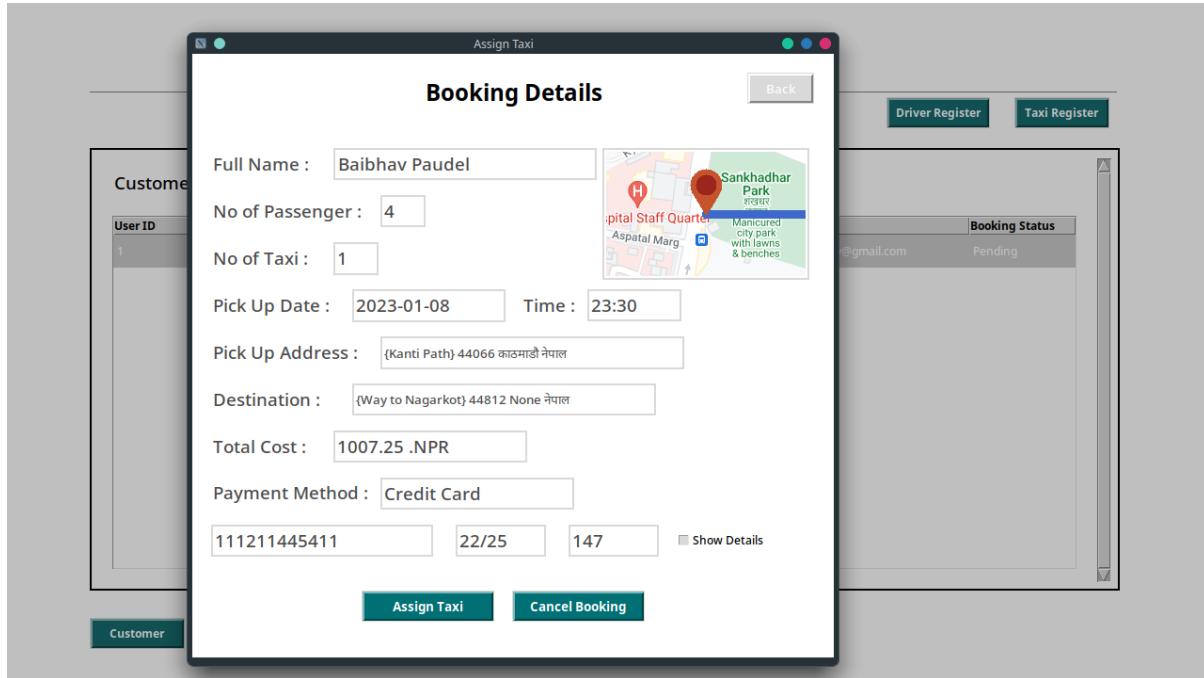


Figure 39: double_clicking_on_booking_request

Double clicking on the top level of the booking request will bring up a pop-up with all the booking details, a minimap showing the pickup and drop-off locations, and a button to assign or cancel the service.

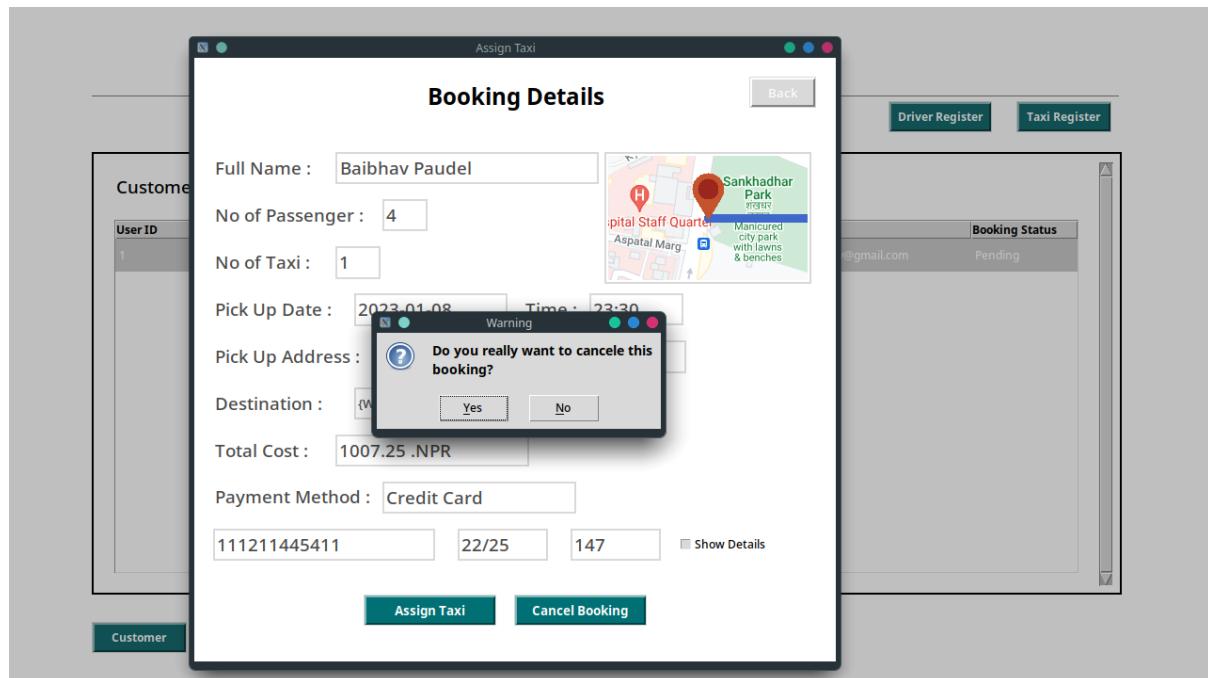


Figure 40: after_clicking_on_cancel_booking

A message box with a security warning appears when a booking request is canceled.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Control Panel

Driver Register Taxi Register

Customer Table Search : View

User ID	Booking ID	Created At	Full Name	Contact	Address	Email	Booking Status
1	1	2023-01-08	Baibhav Paudel	9845145785	Dhangadhi, Kailali	pbaibhav@gmail.com	Canceled

Customer **Driver**

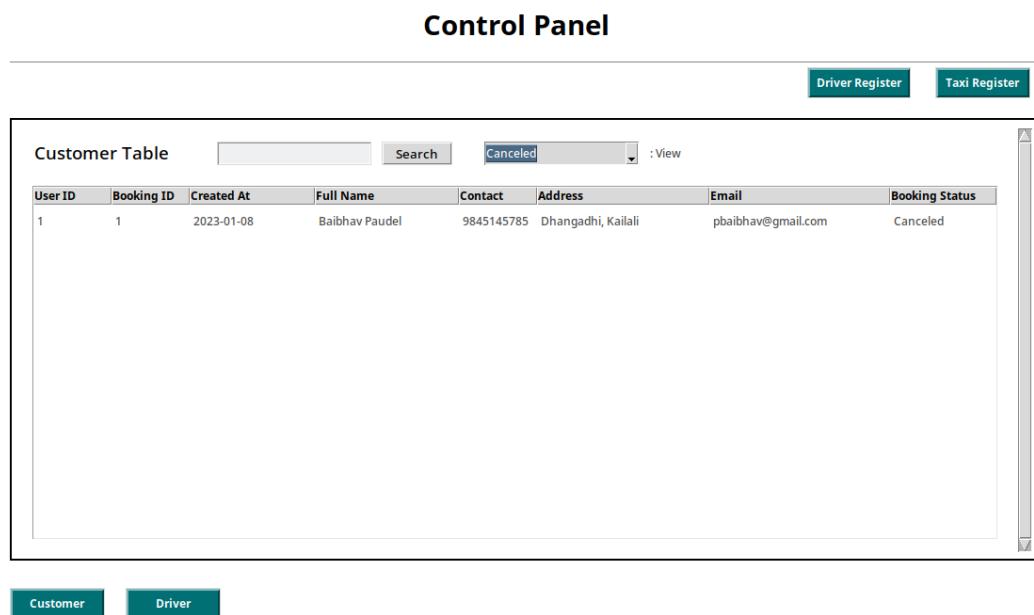


Figure 41: after_booking_is_canceled

When a booking request is canceled, it can be kept track of by using the canceled filter, which displays status as canceled.

Main Frame <-2>

Search : View

Driver ID	Full Name	Contact	Taxi Number
1	Kamlesh Pokhrel	9842541254	111111

Assign

Total Cost : 651.75 .NPR

Payment Method : Cash In Hand

Show Details

Driver Register Taxi Register

Booking Status

mail.com	Pending
----------	---------

Customer

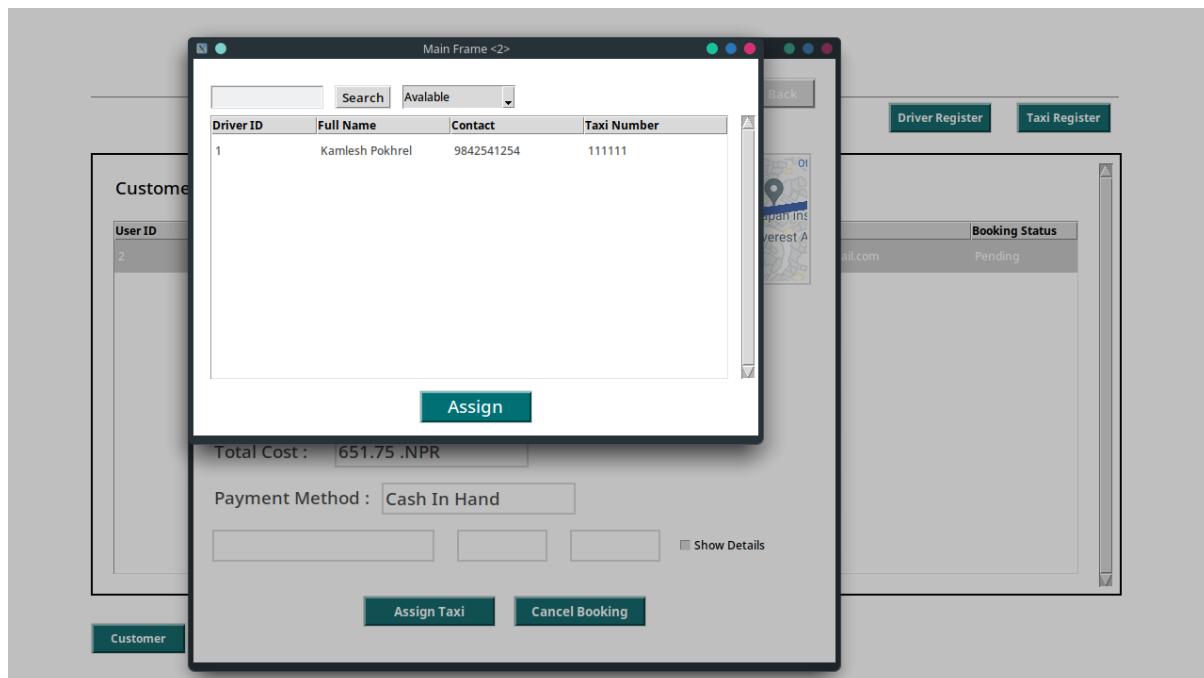


Figure 42: after_clicking_on_assign_button

A new top level with a list of all available taxi drivers that may be assigned to the booking request appears when the assign taxi button is pressed.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Control Panel

Driver Register Taxi Register

Customer Table Search : View

User ID	Booking ID	Created At	Full Name	Contact	Address	Email	Booking Status
2	2	2023-01-08	Krishal Pudasaini	9821451245	Hetauda, Nepal	krsl@gmail.com	Pending

 Success
Driver ID 1 Sucessfully assigned to booking id 2

Customer Driver

Figure 43: after_clicking_on_register

A success message box appears with the necessary information after selecting the driver to assign to the booking.

Control Panel

Driver Register Taxi Register

Customer Table Search : View

User ID	Booking ID	Created At	Full Name	Contact	Address	Email	Booking Status
2	2	2023-01-08	Krishal Pudasaini	9821451245	Hetauda, Nepal	krsl@gmail.com	Accepted

Customer Driver

Figure 44: booking_detail_after_booking_is_accepted

The booking request status changes to accepted when the taxi is assigned, and the Accepted filter may be used to view it.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Control Panel

Driver Register Taxi Register

Driver Table Search : sort by

Driver ID	Full Name	License Number	Contact	Taxi Number	Status
1	Kamlesh Pokhrel	112244335533	9842541254	111111	Booked

[Customer](#) [Driver](#)

Figure 45: driver_detail_after_booking_is_accepted

Driver status changes to booked once a driver is allocated to a booking in order to prevent the driver from being assigned to any other booking requests until the trip is over.

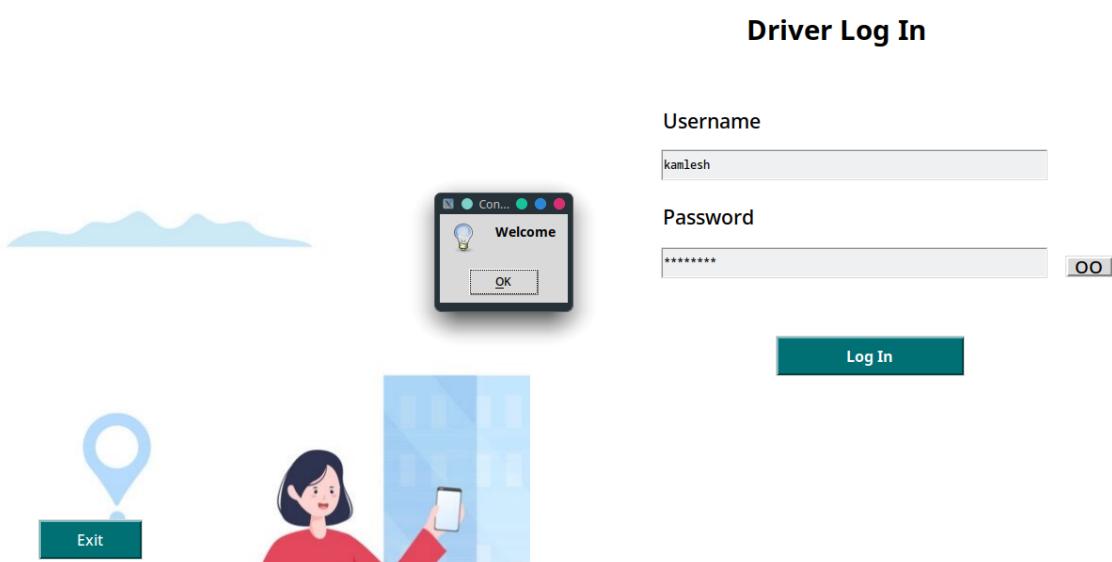


Figure 46: driver_login_page

Driver can access the driver dashboard by using the admin's provided valid login information.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

The screenshot shows a driver dashboard interface. At the top left is a logo with a yellow truck icon and the text 'TURBO TB'. On the right, there's a section for 'kamlesh' with details: Name: Kamlesh Pokhrel, Contact: 9842541254, Taxi: Hyundai Aura, and Taxi number: 111111. Below this is a 'Monthly Revenue' summary with fields for Gross Income, Service Cost, and Net Income, all currently set to 'XXXXX'. A 'Total Revenue' field also shows 'XXXXX'. An 'Update' button is at the bottom right of this section. To the left, a table titled 'Upcoming Trip' lists a single trip: Booking ID 2, Full Name Krishal Pudasaini, Date and Time 2023-01-08 --- 23:15, Destination {Bajra Yogini Temple Road} 44603 None नेपाल, Total Cost 651.75, and Status Accepted. At the bottom are two buttons: 'Upcoming' (highlighted in green) and 'Completed'.

Figure 47: driver_dashboard

Driver dashboards enable drivers to see the trips that have been allocated to them in the near future.

This screenshot shows a modal window titled 'Assign Taxi' with a sub-section 'Booking Details'. It contains fields for Full Name (Krishal Pudasaini), No of Passenger (1), No of Taxi (1), Pick Up Date (2023-01-08), Time (23:15), Pick Up Address (None 13958 None नेपाल), Destination ({Bajra Yogini Temple Road} 44603 None नेपाल), Total Cost (651.75 .NPR), and Payment Method (Cash In Hand). To the right of this window is a sidebar for 'kamlesh' with the same contact information as Figure 47. Below the sidebar is a table with one row: Total Cost 651.75 and Status Accepted. At the bottom of the modal window is a green 'Trip Completed' button.

Figure 48: driver_assigned_booking_detail

When you double-click a booking request, the full details of the booking are displayed on the top level along with a trip finished button that enables you to declare a trip complete after the trip date has begun.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

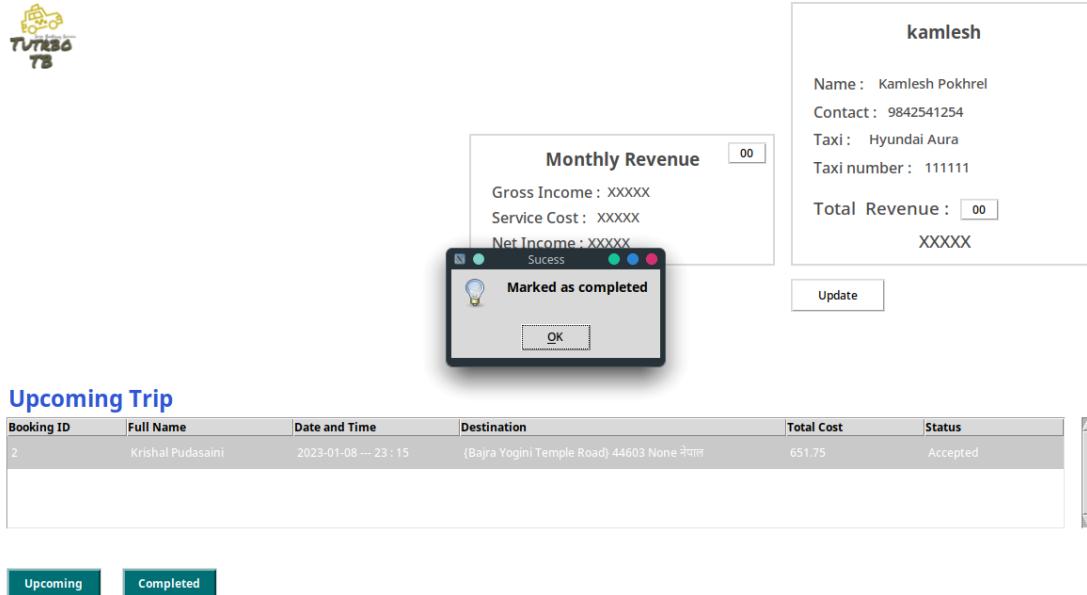


Figure 49: Trip_marked_as_completed

The system notifies the driver via a message box that the journey has been designated as completed after clicking on Trip completed.

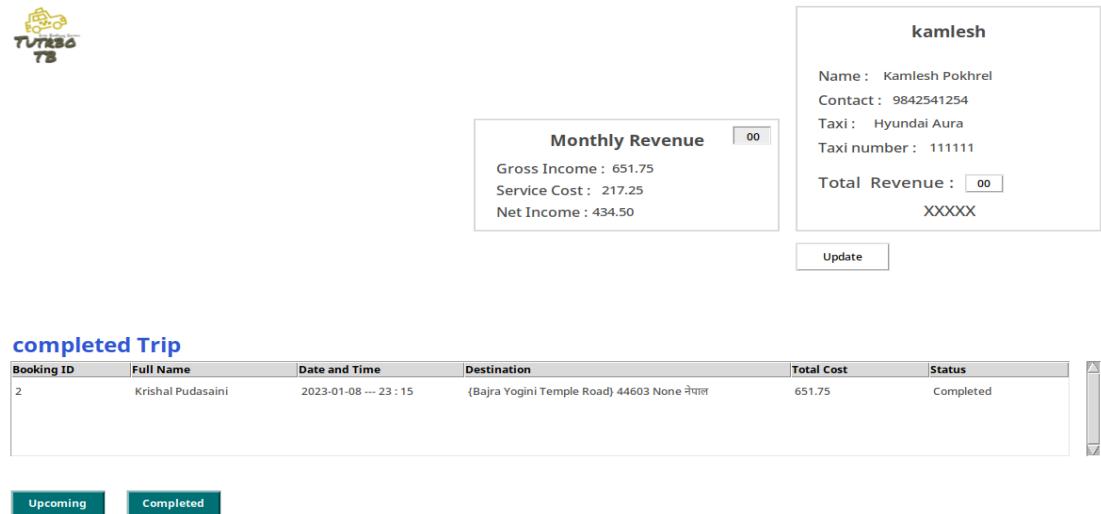


Figure 50: completed_trip_details

Completed trip money is contributed to total revenue as well as monthly revenue calculations that calculate gross income, service cost, and net income and are reset every month for fresh month data. Completed trip data is presented in the completed trip table.

(DZone, 2021)

Pydantic, a Python package that offers a set of tools for developing data validation and data parsing code, was utilized to tackle a problem that arose during the development process regarding verifying my user-given data before putting it in a database.

```
import re
from pydantic import BaseModel, root_validator, validator
from helper.exceptions import CustomException


class UserModel(BaseModel):

    firstname: str = ""
    lastname: str = ""
    contact: str = ""
    address: str = ""
    email: str = ""
    username: str = ""
    password: str = ""
    confirm_password: str = ""
    user_id: str = ""
    message: str = ""
    profile: str = ""
    new_password: str = ""

    @validator("firstname", "lastname")
    def validate_first_name(cls, value):
        if not value:
            raise CustomException("name cannot be empty")
        if not name_reg.match(value):
            raise CustomException("invalid format for name")
        return value
```

Figure 51: pydantic_code_snippets

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

(Schimansky, T.)

A tile-based interactive map renderer widget for Python, TkinterMapView The Tkinter library was also utilized to resolve my other encounter issue, which included giving the system exact pickup and departure locations so that it could determine the trip's overall cost.

Testing

```
@staticmethod
def build_location_picker_frame(
    location_picker_frame,
    location_entry,
):
    google_map_view = TkinterMapView(
        location_picker_frame,
        width=580,
        height=480,
    )
    google_map_view.set_tile_server(
        "https://mt0.google.com/vt/lyrs=m&hl=en&x={x}&y={y}&z={z}&s=Ga",
        max_zoom=22,
    )
    google_map_view.place(relx=0, rely=0.07)
    google_map_view.set_address("kathmandu, nepal")
    google_map_view.set_zoom(13)

    selected_address = tk.Label(location_picker_frame)
    selected_address.place(relx=0.0, rely=0.010, height=33, relwidth=0.999)
    selected_address.config(background="#FFFFFF")

    google_map_view.add_left_click_map_command(
        lambda coordinates: BookingPage.location_marker(
            coordinates,
            selected_address,
        ),
    )
)
```

Figure 52: TkinterMapView_code_snippets

In the context of software development, testing is the process of assessing a system or its component in order to ascertain whether it fulfills the requirements or not.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

SIGNUP PAGE TESTING

Test No.	Test Purpose	Input Data/Action	Expected Result	Actual Result	Reference
1	To check validation for email address input.	Wrong format of email.	Error Message “ Invalid Email Format”	Expected Result	Figure 53
2	To check password and confirm password match validation	Unmatching data password : fastrack confirm password : google	Error Message “ passwords not matched”	Expected Result	Figure 54 Figure 55
3	To check Username and Email unique for each user.	Username and Email that already exist in database	Error Message “ Username or Email already used “	Expected Result	Figure 56 Figure 57 Figure 58
4	Result After all valid data input	Valid data	Sucess Message and Data added to database	Expected Result	Figure 59 Figure 60

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

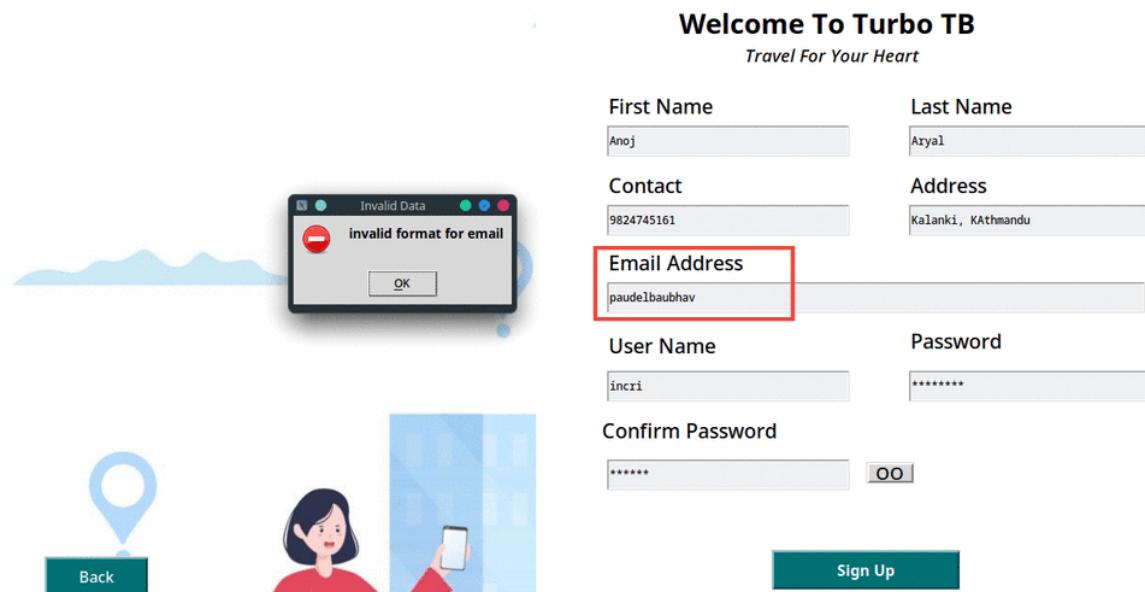


Figure 53: test_with_invalid_email_format

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

The screenshot shows the 'Welcome To Turbo TB' sign-up page. The page has fields for First Name (Anoj), Last Name (Aryal), Contact (9824745161), Address (Kalanki, Kathmandu), Email Address (franzy@gmail.com), User Name (incri), Password (fastrack), and Confirm Password (google). The 'Password' and 'Confirm Password' fields are highlighted with red boxes. A 'Sign Up' button is at the bottom.

Figure 54: test_with_unmatch_confirm_password

The screenshot shows the 'Welcome To Turbo TB' sign-up page. The page has fields for First Name (Anoj), Last Name (Aryal), Contact (9824745161), Address (Kalanki, Kathmandu), Email Address (franzy@gmail.com), User Name (incri), Password (*****), and Confirm Password (*****). A modal dialog box titled 'Invalid Data' with a red minus sign icon displays the message 'The two passwords did not match.' A 'Sign Up' button is at the bottom.

Figure 55: result_of_test Figure 54

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

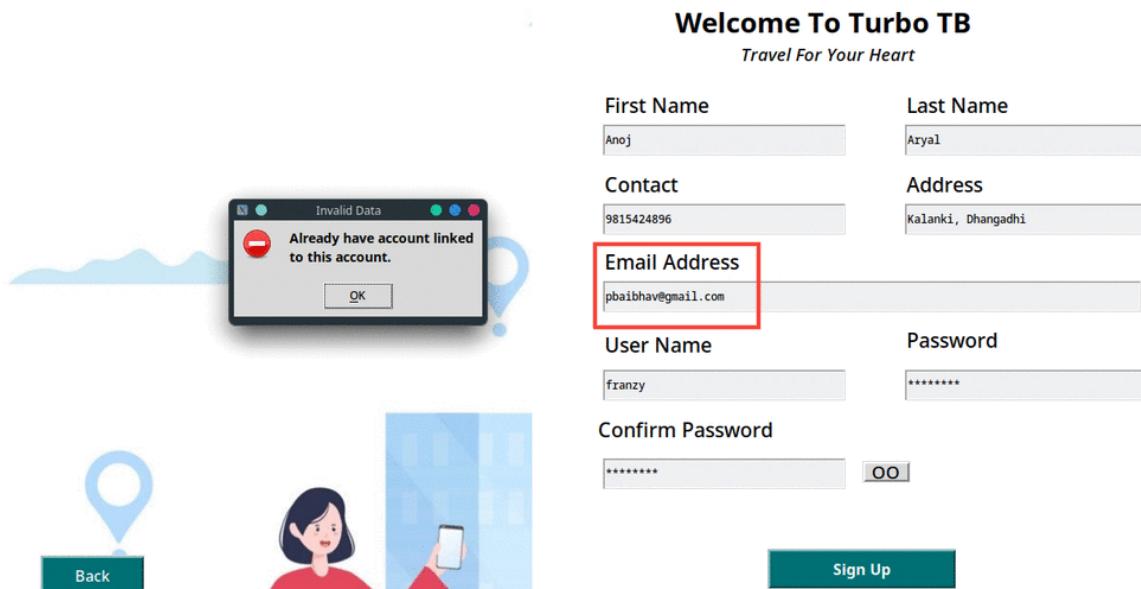


Figure 57: test_signup_with_existing_email

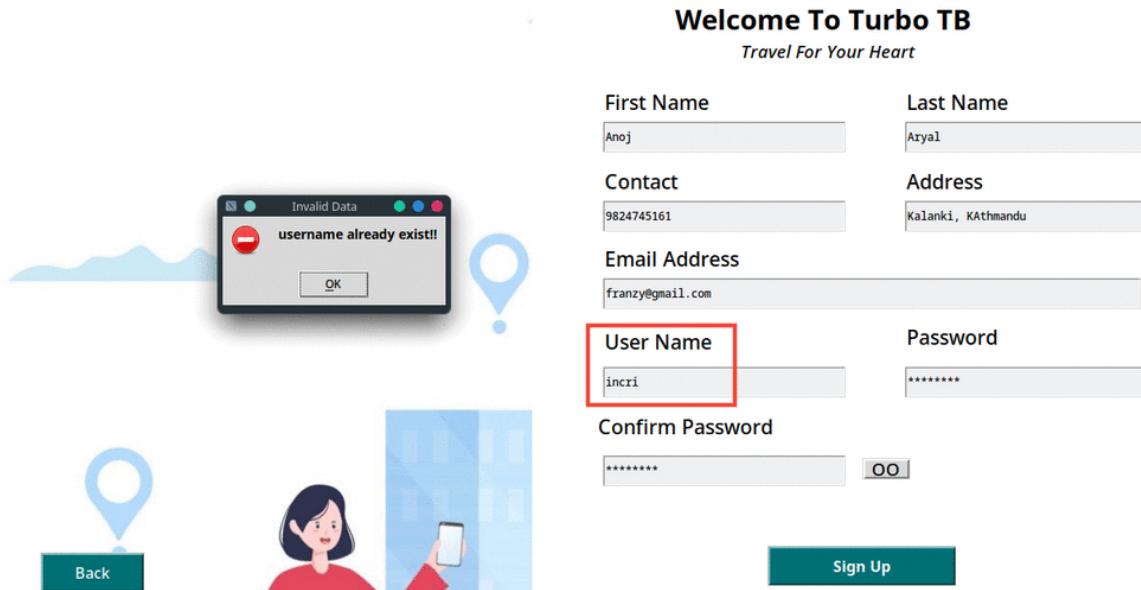


Figure 58: test_signup_with_existing_username

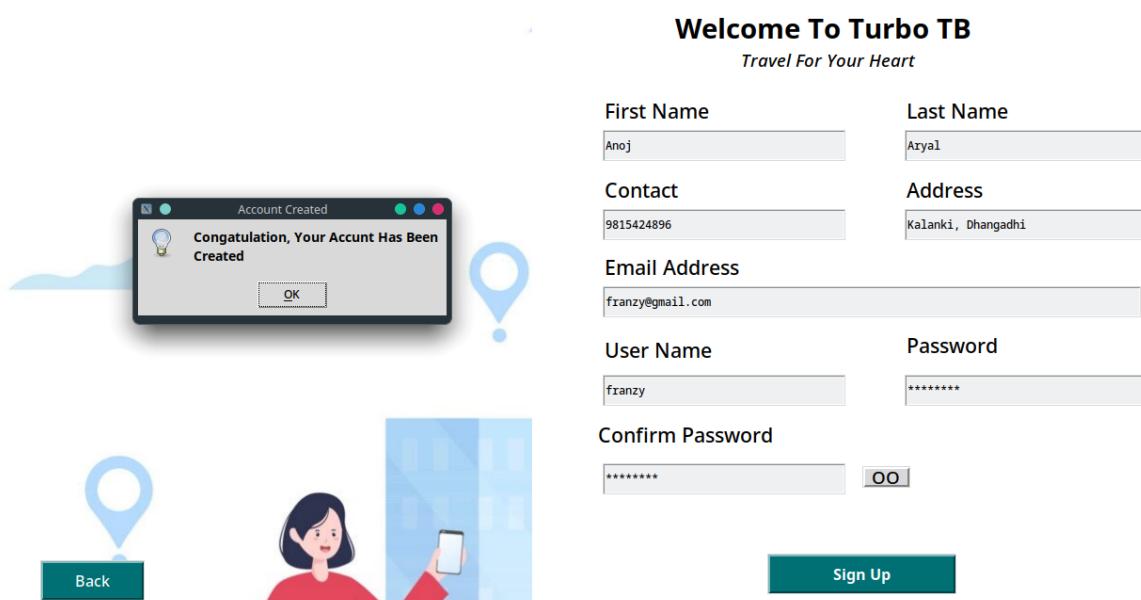


Figure 59: test_with_all_valid_data

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

userid	firstname	lastname	contact	address	email
1	Baibhav	Paudel	9845145785	Dhangadhi, Kailali	pbaibhav@gmail.com
2	Krishal	Pudasaini	9821451245	Hetauda, Nepal	lkrsl@qmail.com
7	Anoj	Aryal	9815424896	Kalanki, Dhangadhi	franzy@gmail.com

Figure 60: signup_sucess_result

LOGIN PAGE TESTING

Test No.	Test Purpose	Input Data/Action	Expected Result	Actual Result	Reference
1	To check user must input valid data to login	No data	Error Message “Entry box cannot be empty”	Expected Result	Figure 61
2	To check user must input valid username	Wrong username : superman	Error Message “Invalid username or password”	Expected Result	Figure 62
3	To check user must input valid username as well as password	Right username : franzy Wrong password : google	Error Message “Invalid username or password”	Expected Result	Figure 63
4	User must	Right username :	User redirect to	Expected Result	Figure 65

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

redirect to	franzy	there dashboard.
there account		
dashboard	Right password :	
after valid	fastrack	
username and		
password		

Figure 66

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

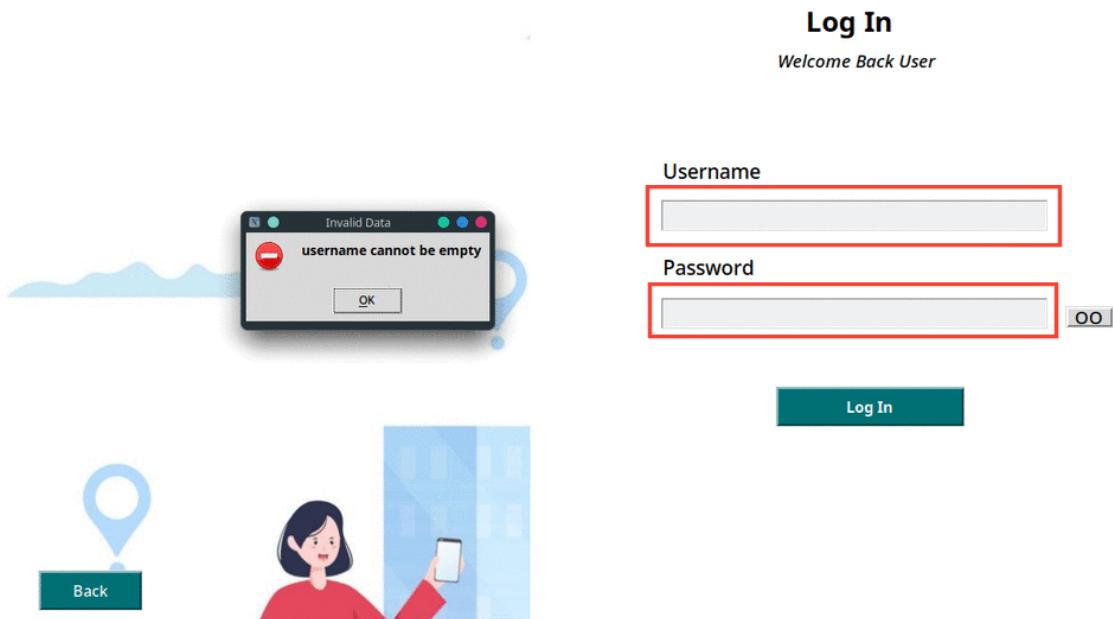


Figure 61: login_with_no_data

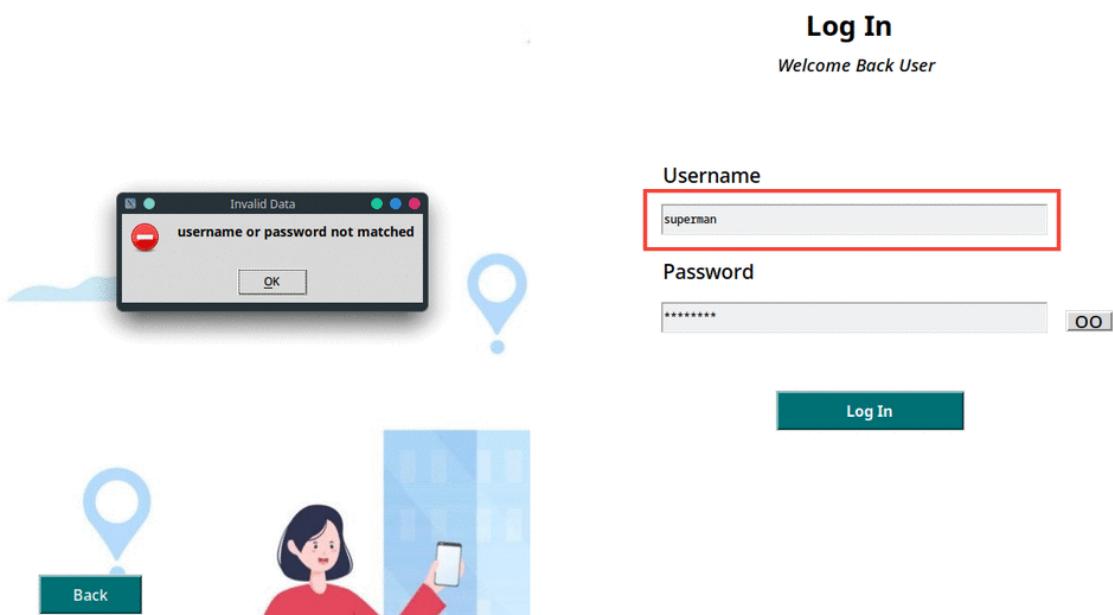


Figure 62: login_with_non_registered_username

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

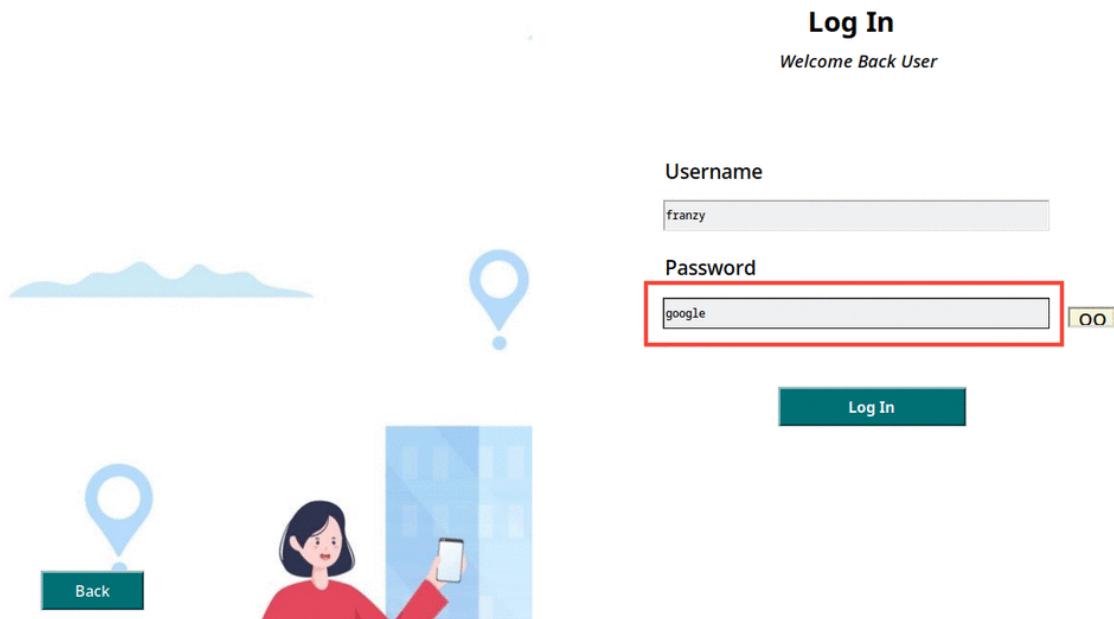


Figure 63: registered_username_but_wrong_password

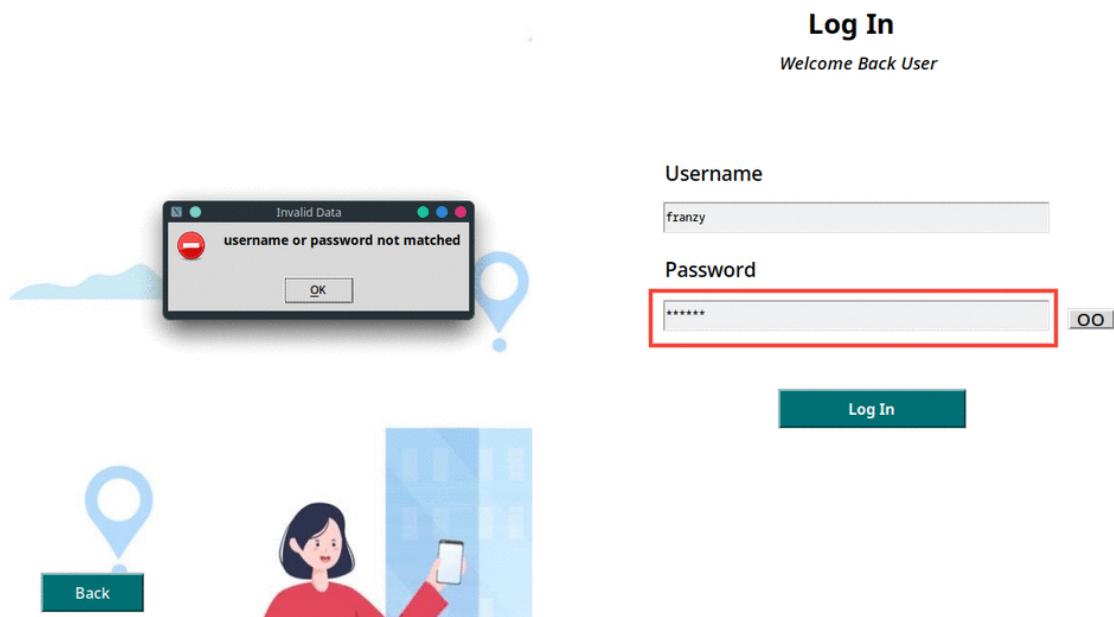


Figure 64: result_of Figure 63

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

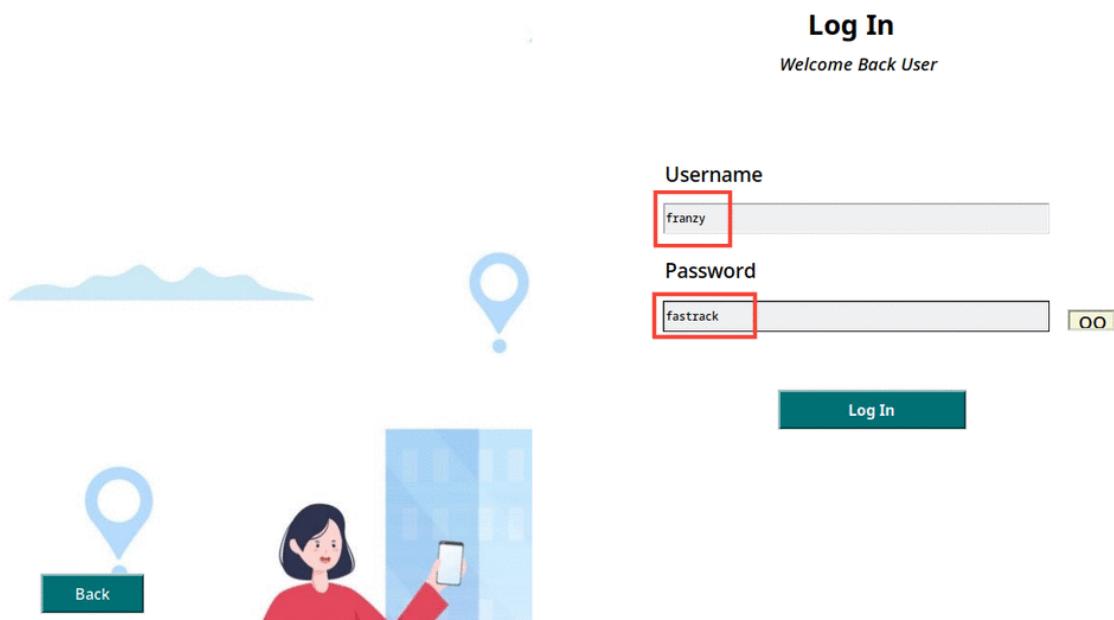


Figure 65: valid_data_for_login

This image shows the "Your Trip History" section of the application. At the top, there's a logo for "TURBO TB" featuring a yellow car icon. Below the header, a table lists trip details with columns: Booking ID, Full Name, Date and Time, Destination, Total Cost, and Status. The table is currently empty. At the bottom, there are two buttons: "Book Now" and "Refresh Page".

Figure 66: result_for Figure 65

CIS020-1 – Introduction to Software Development**CIS093-1 – Mathematics and Concepts for Computational Thinking****Assignment 2 – Individual Project – Case Study (Taxi Booking System)**

University ID : 2146504

Student Name : **BAIBHAV PAUDEL****BOOKING FORM PAGE TESTING**

Test No.	Test Purpose	Input Data/Action	Expected Result	Actual Result	Reference
1	Booking page must contain booking form with today date and full name of user.	Clicked on Book now button in dashboard	Booking form with today date and full name of user.	Expected Result	Figure 67
2	Pickup time must not be empty and before today date and time.	No time given and then no minute given for secound test.	Error Message “Time cannot be empty”	Expected Result	Figure 68 Figure 69
3	User must provide pickup location.	No pickup location given	Error Message “Pickup Location cannot be empty”	Expected Result	Figure 70
4	Select location must provide user with map with view location when clicked , search for location and working select location button.	Clicked on select location on dash board, clicked on place of map, search location and clicked on select location button.	Map, selected location, location that has been searched, selected location to view on pickup location entry.	Expected Result	Figure 71 Figure 72 Figure 73
5	User must provide destination location.	No destination location given	Error Message “Destination Location cannot be empty”	Expected Result	Figure 74
6	User must be able to view cost for there trip when pickup location and destination location is selected and cost price must change with change in location	Both pickup location and destination location given and clicked to view cost.	Must view cost for the trip and cost must change with change in location.	Expected Result	Figure 75 Figure 76

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

7	Selecting on payment method as credit card, user must provide there credit detail before requesting booking.	No credit detail given	Error Message “ Credit detail cannot be empty”	Expected Result	Figure 77
8	Selecting on payment method as cash in hand don't need any credit detail.	Selected payment method as cash in hand and no credit detail given.	“Booking requested sucessfully”	Message box	Expected Result Figure 78
9	Sucess booking must save all booking details into database.	Checked database.	New Booking request with recent booking details.	Expected Result	Figure 79

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Book Now

Back

some information

Full Name :

First Name Last Name

Number Of Passenger :

Number Of Taxi : a taxi can carry only upto 4 people

Pick Up Date :

MM/DD/YY HH(24hrs) : MM

Pick Up Address :

Destination :

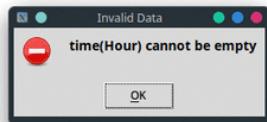
Total Charge : XXXXXX

Figure 67: booking_form_with_data

Book Now

Back

some information



Full Name :

First Name Last Name

Number Of Passenger :

Number Of Taxi : a taxi can carry only upto 4 people

Pick Up Date :

MM/DD/YY HH(24hrs) : MM

Pick Up Address :

Destination :

Total Charge : XXXXXX

Figure 68: without_time_given

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

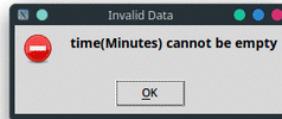
University ID : 2146504

Student Name : BAIBHAV PAUDEL

Book Now

Back

some information



Full Name : Anoj Aryal

First Name Last Name

Number Of Passenger : 1

Number Of Taxi : 1 a taxi can carry only upto 4 people

Pick Up Date : 2023/01/11 15 **20**

MM/DD/YY HH(24hrs) : MM

Pick Up Address : Select Address

Destination : Select Address

Total Charge : XXXXX

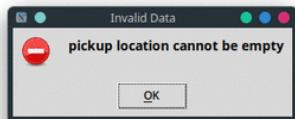
Payment Method: Credit Card (radio button selected), Cash In Hand (radio button unselected). Fields for Card Number, Exp Date, and CVV are present.

Figure 69: without_minute_given

Book Now

Back

some information



Full Name : Anoj Aryal

First Name Last Name

Number Of Passenger : 1

Number Of Taxi : 1 a taxi can carry only upto 4 people

Pick Up Date : 2023/01/11 15 20

MM/DD/YY HH(24hrs) : MM

Pick Up Address : **Empty input field** Select Address

Destination : Select Address

Total Charge : XXXXX

Payment Method: Credit Card (radio button selected), Cash In Hand (radio button unselected). Fields for Card Number, Exp Date, and CVV are present.

Figure 70: pickup_location_not_given

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

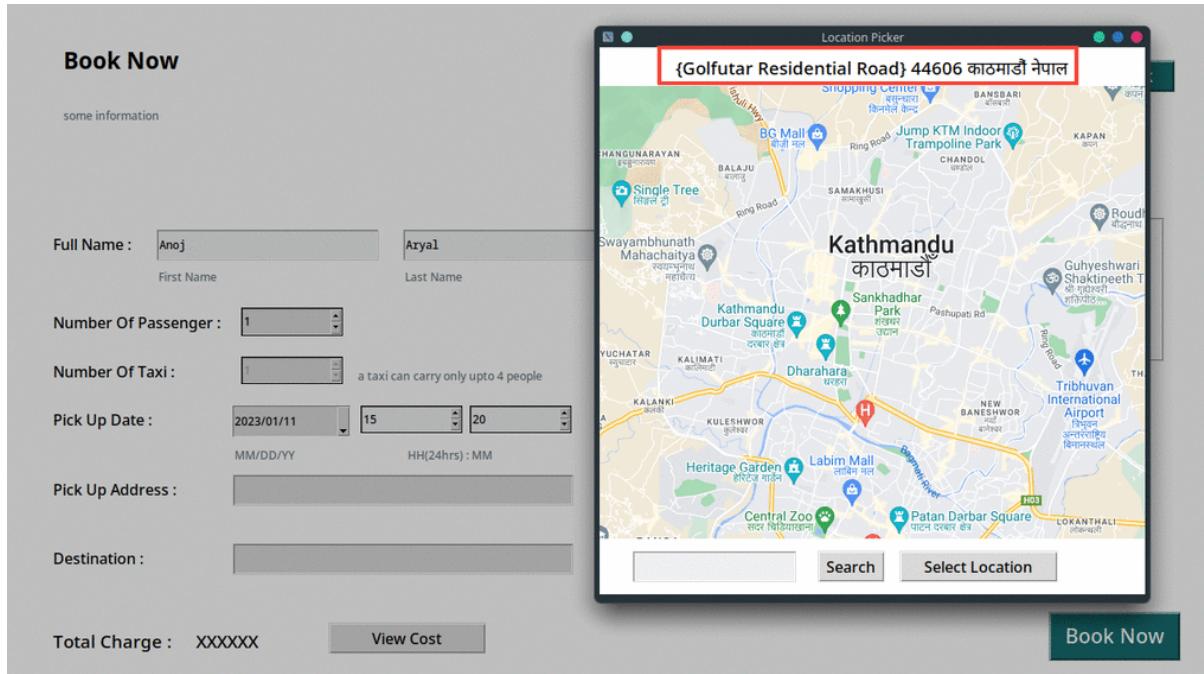


Figure 71: click_on_map_to_view_location

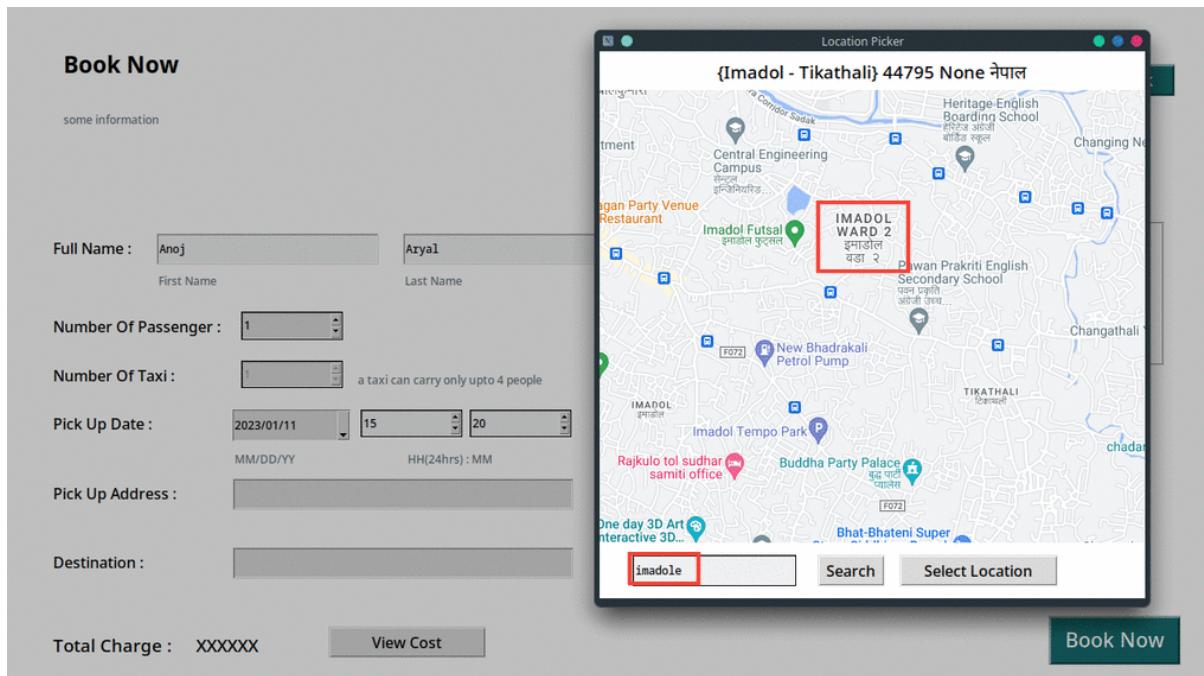


Figure 72: search_feature_on_map

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Book Now

Back

some information

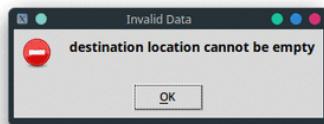
Full Name :	Anoj	Aryal	Payment Method
First Name			<input type="radio"/> Credit Card
Last Name			<input type="radio"/> Cash In Hand
Number Of Passenger :	1		Card Number
Number Of Taxi :	1	a taxi can carry only upto 4 people	
Pick Up Date :	2023/01/11	15	20
MM/DD/YY	HH(24hrs) : MM		
Pick Up Address :	(Imadol - Tikathali) 44795 None नेपाल		
Destination :	<input type="button" value="Select Address"/>		
Total Charge :	XXXXXX		
<input type="button" value="View Cost"/>			<input type="button" value="Book Now"/>

Figure 73: when_location_is_selected

Book Now

Back

some information



Full Name :	Anoj	Aryal	Payment Method
First Name			<input type="radio"/> Credit Card
Last Name			<input type="radio"/> Cash In Hand
Number Of Passenger :	1		Card Number
Number Of Taxi :	1	a taxi can carry only upto 4 people	
Pick Up Date :	2023/01/11	15	20
MM/DD/YY	HH(24hrs) : MM		
Pick Up Address :	(Imadol - Tikathali) 44795 None नेपाल		
Destination :	<input type="button" value="Select Address"/>		
Total Charge :	XXXXXX		
<input type="button" value="View Cost"/>			<input type="button" value="Book Now"/>

Figure 74: without_destination_location

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Book Now

Back

some information

Full Name :	Anoj	Aryal	
First Name	Last Name		
Number Of Passenger :	1		
Number Of Taxi :	1	a taxi can carry only upto 4 people	
Pick Up Date :	2023/01/11	15	20
MM/DD/YY	HH(24hrs) : MM		
Pick Up Address :	(Imadol - Tikathali) 44795 None नेपाल		
Destination :	(महेन्द्र राजपाल) 44619 None नेपाल		
Total Charge :	NPR. 8591.25		View Cost
Book Now			

Figure 75: view_cost

Book Now

Back

some information

Full Name :	Anoj	Aryal	
First Name	Last Name		
Number Of Passenger :	1		
Number Of Taxi :	1	a taxi can carry only upto 4 people	
Pick Up Date :	2023/01/11	15	20
MM/DD/YY	HH(24hrs) : MM		
Pick Up Address :	(Imadol - Tikathali) 44795 None नेपाल		
Destination :	(तरिशुवन राजपाल) 44619 None नेपाल		
Total Charge :	NPR. 829.5		View Cost
Book Now			

Figure 76: change_cost_with_change_in_location

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Book Now

Back

some information



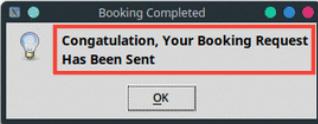
Full Name :	Anoj	Aryal	
First Name	Last Name		
Number Of Passenger :	1		
Number Of Taxi :	1	a taxi can carry only upto 4 people	
Pick Up Date :	2023/01/11	15	20
MM/DD/YY	HH(24hrs) : MM		
Pick Up Address :	(Imadol - Tikathali) 44795 None नेपाल		
Destination :	(तरिश्वन राजपथ) 44619 None नेपाल		
Total Charge :	NPR. 829.5		
View Cost			
Book Now			

Figure 77: credit_card_method_without_details

Book Now

Back

some information



Full Name :	Anoj	Aryal	
First Name	Last Name		
Number Of Passenger :	1		
Number Of Taxi :	1	a taxi can carry only upto 4 people	
Pick Up Date :	2023/01/11	15	20
MM/DD/YY	HH(24hrs) : MM		
Pick Up Address :	(Imadol - Tikathali) 44795 None नेपाल		
Destination :	(तरिश्वन राजपथ) 44619 None नेपाल		
Total Charge :	NPR. 829.5		
View Cost			
Book Now			

Figure 78: cash_in_hand_payment_method

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Grid	booking_id	user_id	firstname	lastname	no_of_passenger	no_of_taxis	pickup_date	pickup_time_hrs	pickup_time_min
Text	1	1	Baibhav	Paudel	4	1	2023-01-08	23	
Record	2	2	Krishal	Pudasaini	1	1	2023-01-08	23	
	3	7	Anoj	Aryal	1	1	2023-01-11	15	

Figure 79: database_with_new_booking_details

CONTROL PANEL PAGE TESTING

Test No.	Test Purpose	Input Data/Action	Expected Result	Actual Result	Reference
1	To check control panel filter feature.	Logged in as admin.	Customer table data change with filter.	Expected Result	Figure 80 Figure 81
2	To check taxi register feature.	Clicked on taxi register.	Toplevel with taxi register feature.	Expected Result	Figure 82
3	To check taxi register validation for without taxi number.	Taxi number not given.	Error Message “taxi number cannot be empty”	Expected Result	Figure 83
4	To check taxi register with valid data stores the taxi details into database	Valid data given, clicked to register taxi.	Taxi details sucessfully stored.	Expected Result	Figure 84 Figure 85

CIS020-1 – Introduction to Software Development**CIS093-1 – Mathematics and Concepts for Computational Thinking****Assignment 2 – Individual Project – Case Study (Taxi Booking System)**

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

5	To check driver register feature.	Clicked on driver register.	Toplevel with driver register feature.	Expected Result	Figure 86
6	To check driver register license number feature	Tried registering without driver license number	Error Message “License cannot be empty”	Expected Result	Figure 87
7	To check selecting available taxi number show details to the taxi number taxi.	Selected available taxi number to assign to driver.	Detail of selected taxi number is available.	Expected Result	Figure 88
8	To check driver register with valid data stores the driver details into database	Valid data given, clicked to register driver.	Taxi details sucessfully stored.	Expected Result	Figure 89 Figure 90
9	To check taxi assign feature to booking.	Tried assigning taxi by double clicking on customer table request.	Toplevel with full detail of booking request along with assign booking and cancel booking feature.	Expected Result	Figure 91
10	To check working of taxi feature.	Clicked on assign taxi, selected driver among available one, assigned taxi.	Booking status should change to Accepted and assigned driver status change to booked.	Expected Result	Figure 92 Figure 93 Figure 94

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Control Panel

[Driver Register](#) [Taxi Register](#)

Customer Table Search :View

User ID	Booking ID	Created At	Full Name	Contact	Address	Email	Booking Status
7	3	2023-01-11	Anoj Aryal	9815424896	Kalanki, Dhangadhi	franzy@gmail.com	Pending

[Customer](#) [Driver](#)

Figure 80: filter_at_pending

Control Panel

[Driver Register](#) [Taxi Register](#)

Customer Table Search :View

User ID	Booking ID	Created At	Full Name	Contact	Address	Email	Booking Status
1	1	2023-01-08	Baibhav Paudel	9845145785	Dhangadhi, Kailali	pbaibhav@gmail.com	Canceled
2	2	2023-01-08	Krishal Pudasaini	9821451245	Hetauda, Nepal	krsli@gmail.com	Completed
7	3	2023-01-11	Anoj Aryal	9815424896	Kalanki, Dhangadhi	franzy@gmail.com	Pending

[Customer](#) [Driver](#)

Figure 81: filter_at_all

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

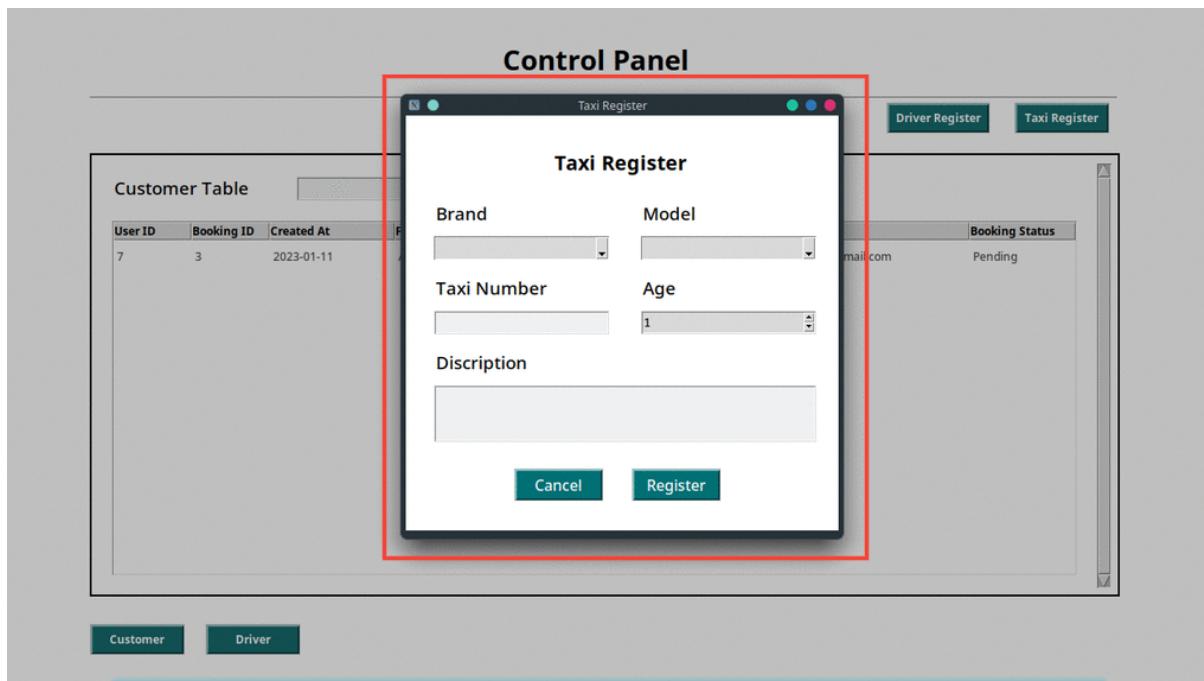


Figure 82: taxi_register_toplevel

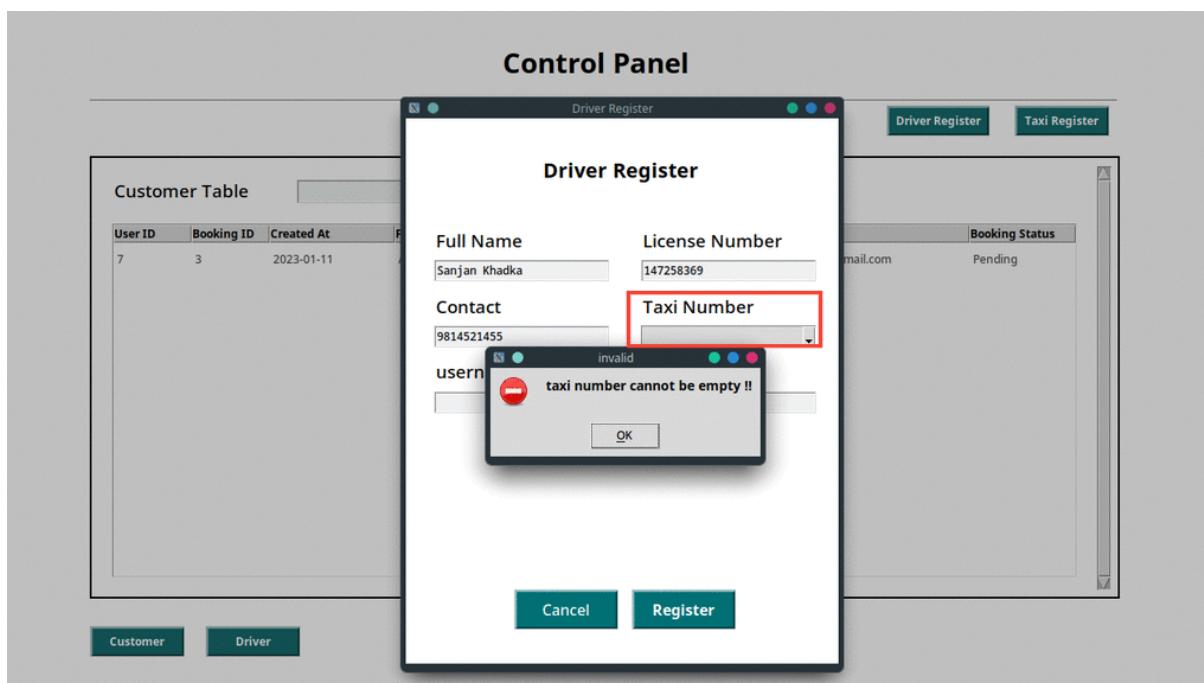


Figure 83: taxi_register_without_taxi_number

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

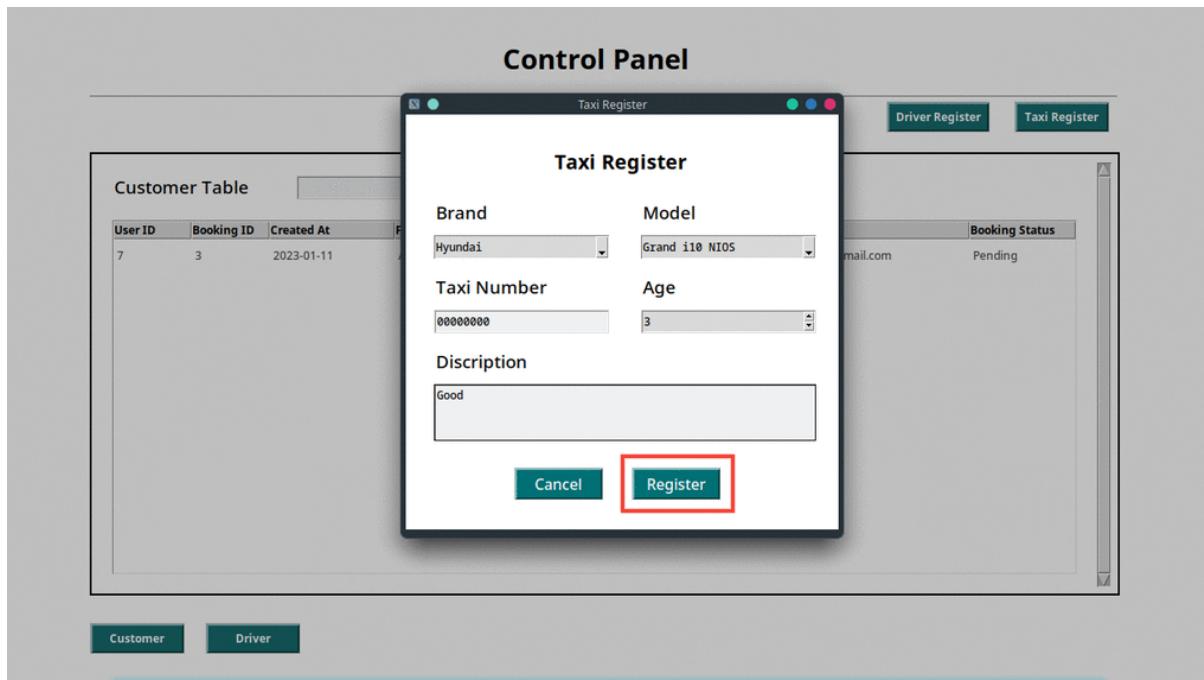


Figure 84: taxi_register_with_valid_data

The screenshot shows a terminal window with the command "taxi" entered. The output is a table with the following data:

id	brand	model	taxi_number	taxi_age	discription	status
1	Hyundai	Aura	111111	1	2 very good taxi.	Assigned
2	Hyundai	Grand i10 NIOS	00000000	3	Good	Available

Figure 85: taxi_register_result

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

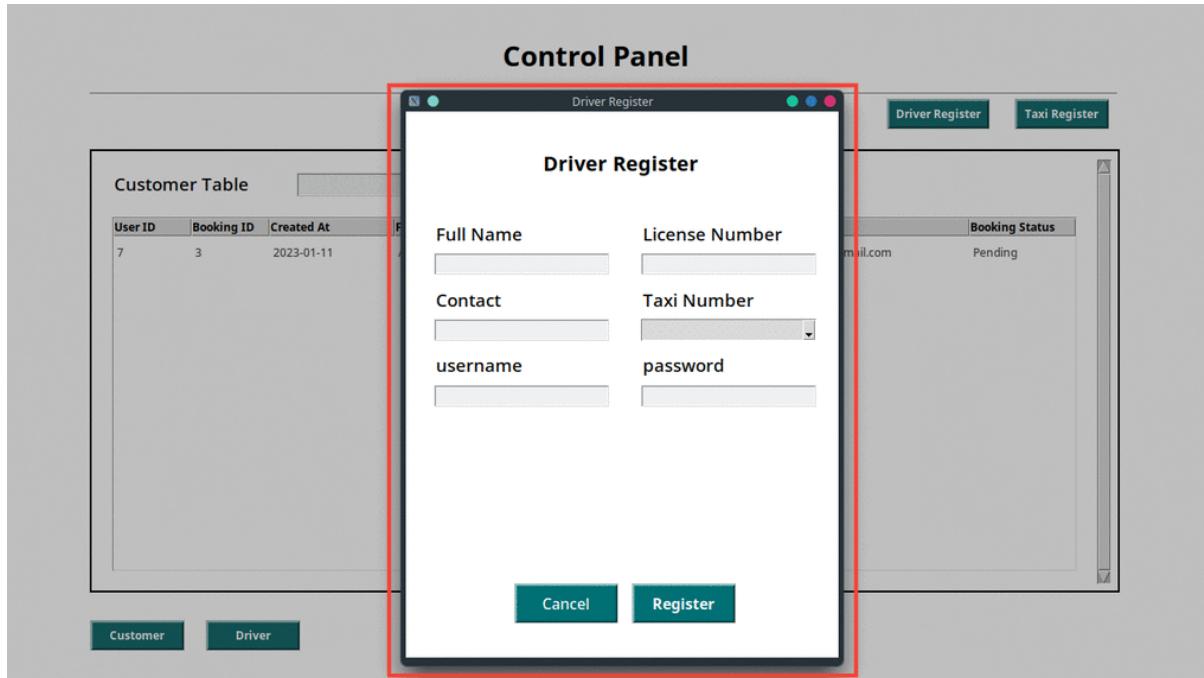


Figure 86: driver_register_page

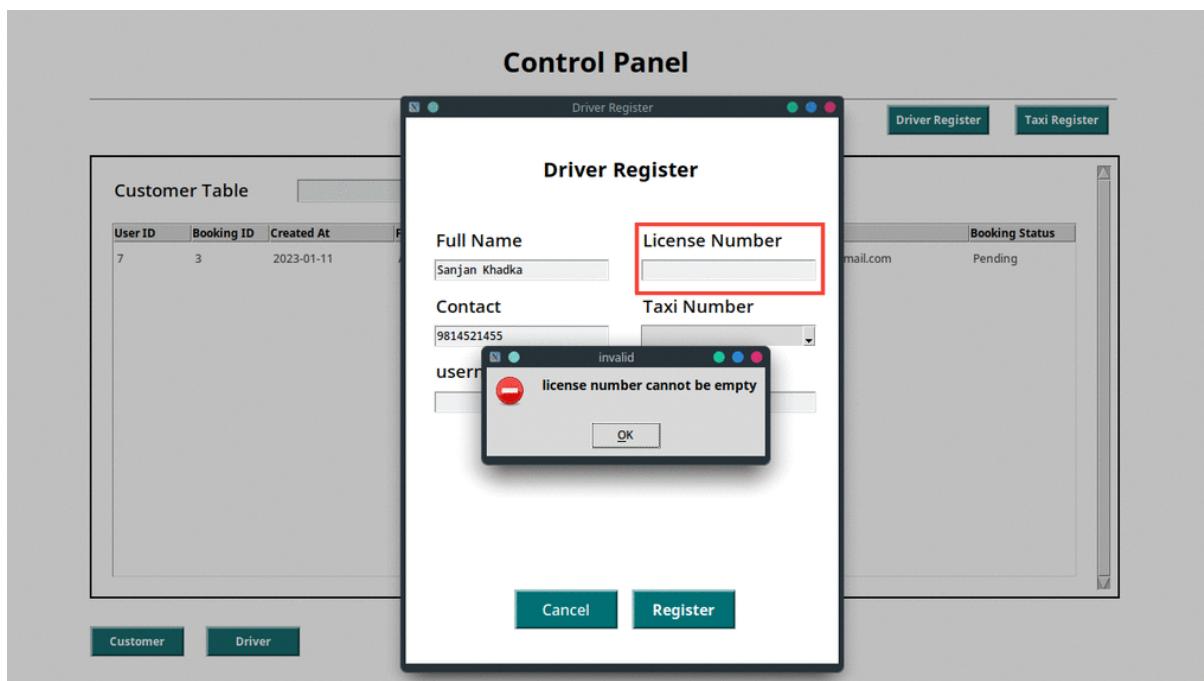


Figure 87: driver_register_without_license_number

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

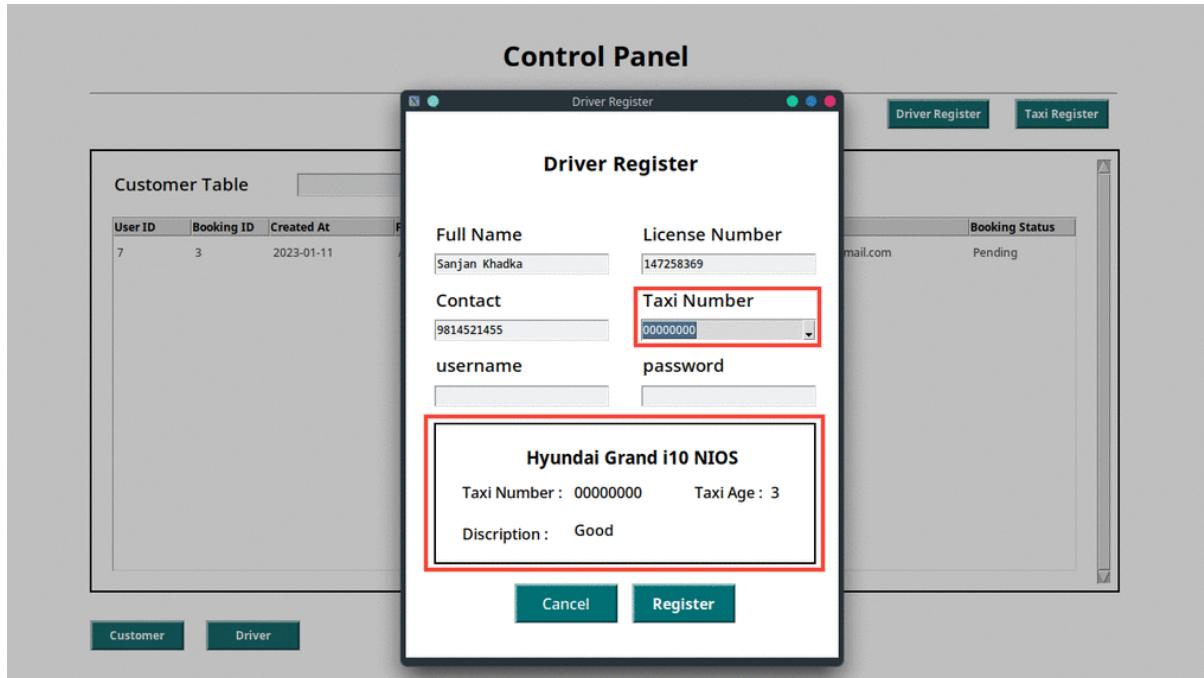


Figure 88: driver_register_after_taxi_number_is_selected

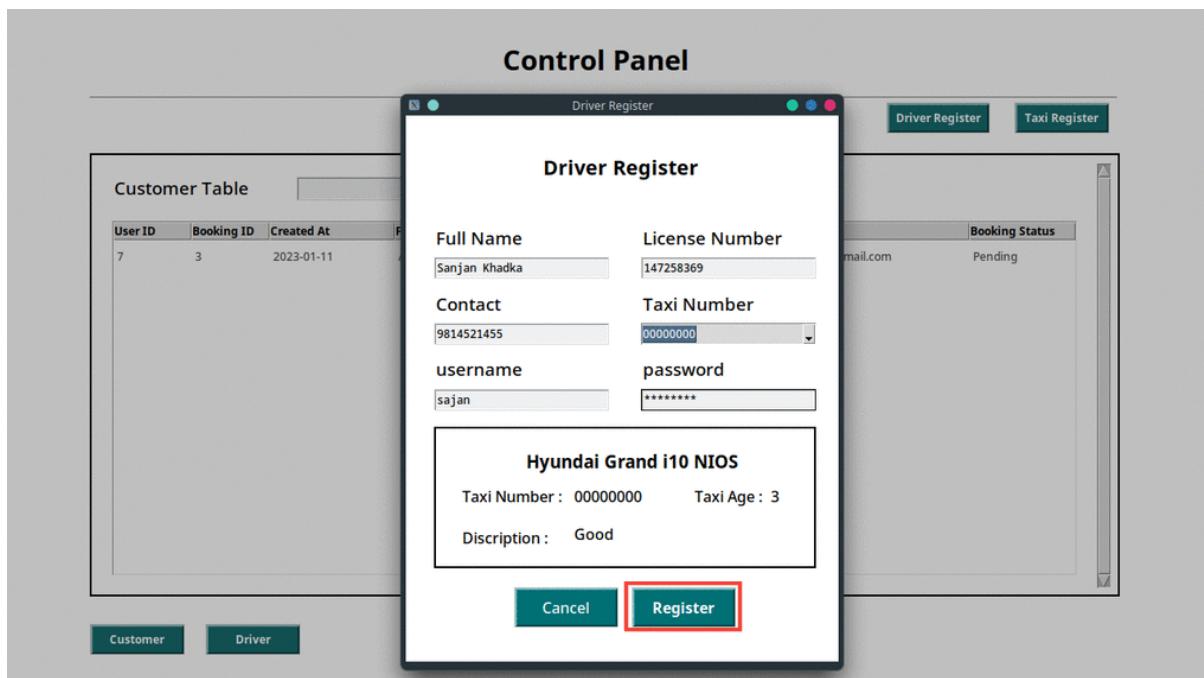


Figure 89: taxi_driver_register_with_valid_data

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

drivers						
Grid	driverid	fullname	license_number	contact	taxi_number	driver_status
	1	Kamlesh Pokhrel	112244335533	9842541254	111111	Available
	2	Sanjan Khadka	147258369	9814521455	00000000	Available

Figure 90: driver_register_result

The screenshot shows a software interface for a taxi booking system. In the center, a modal window titled "Booking Details" is displayed. It contains fields for "Full Name" (Anoj Aryal), "No of Passenger" (1), "No of Taxi" (1), "Pick Up Date" (2023-01-11), "Time" (15:20), "Pick Up Address" (Imadol - Tikathali) 44795 None नेपाल), "Destination" (लक्ष्मीन राजपथ) 44619 None नेपाल), "Total Cost" (829.5 .NPR), and "Payment Method" (Cash In Hand). Below these fields are three empty input boxes and a "Show Details" link. At the bottom of the modal are two buttons: "Assign Taxi" and "Cancel Booking", with "Assign Taxi" highlighted by a red box. In the background, there's a sidebar labeled "Customer" with a "User ID" list, and tabs for "Driver Register" and "Taxi Register". To the right, a panel shows a map with a red pin and a table with a single row: "Booking Status" (Pending).

Figure 91: booking_request_detail_page

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

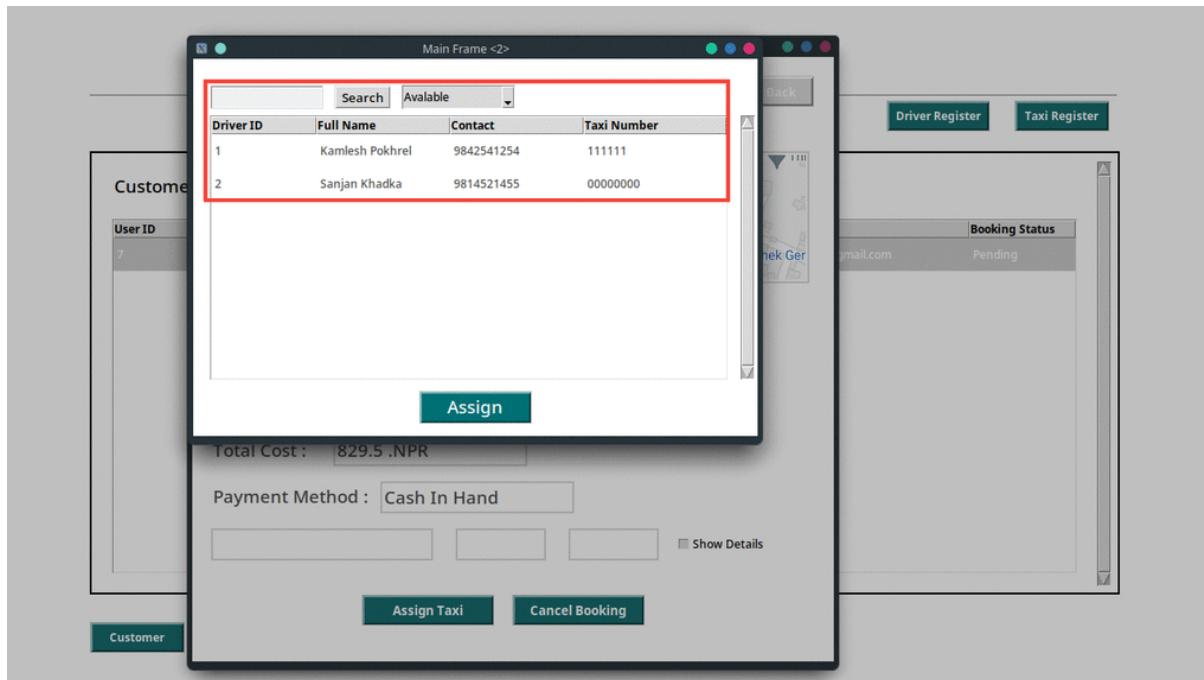


Figure 92: assign_taxi_from_available_driver

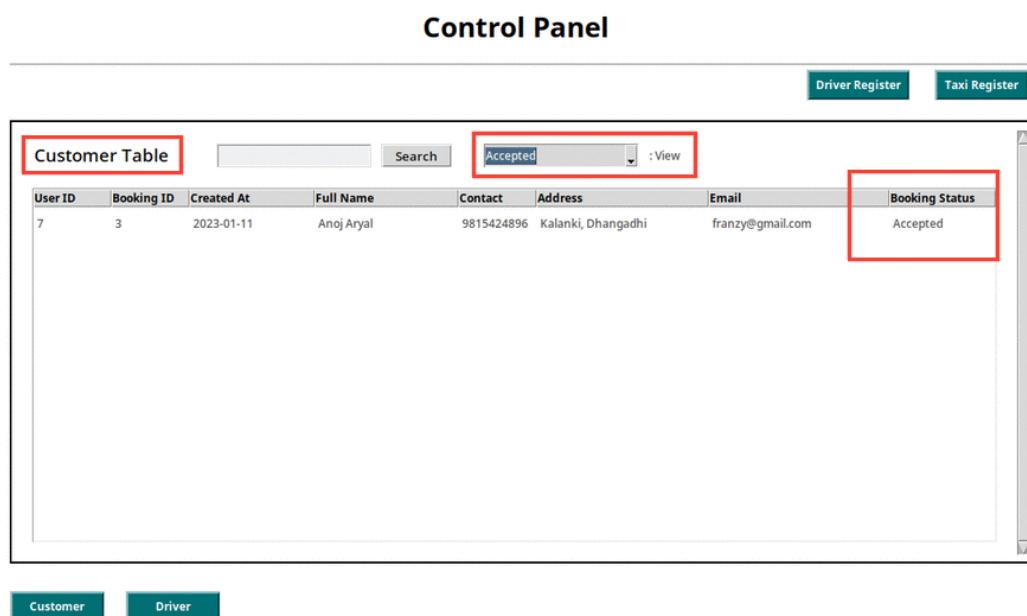


Figure 93: booking_status_changed_after_assign

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Discussion / Reflection / Critical Analysis

As an individual working on a taxi booking software project, here are some things that have gone well or wrong during the development process:

What went well:

- In order to achieve the project's objectives, I was able to describe them precisely and lay out a plan.
- I was able to effectively research and learn any new technologies or tools necessary for the project.
- I was successful in keeping progress on schedule and meeting project deadlines.

What went wrong:

- I struggled with feeling overwhelmed or burnt out while working on the project alone.
- I had difficulty finding the necessary resources to complete the project.
- I have encountered technical challenges that were difficult to resolve on your own.

As an individual, I had to cultivate strong time management skills throughout the development process. I had to be mindful of deadlines and ensure that I was making steady progress towards completing my tasks on schedule. I used tools such as project management software to help me track my progress and identify any potential bottlenecks or delays.

Yes, I was able to achieve all the required specifications of the assignment brief. I carefully reviewed the requirements and made sure to incorporate all necessary features into my project. I did encounter a few problems along the way, but I was able to overcome them through careful debugging and troubleshooting. For example, I initially had difficulty implementing the user login system, but I was able to resolve the issue by reviewing the code and consulting online resources for guidance.

I gained a lot of knowledge about creating user interfaces and using databases in Python by working on this assignment. In addition, I learned a lot about utilizing Tkinter to create a useful and attractive taxi booking system. I valued the chance to use my programming knowledge on a practical project, and I believe this experience has helped me become a better programmer. Overall, I believe that working on this assignment has helped me understand software development more thoroughly.

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

Conclusion

In conclusion, it was difficult yet satisfying to create a desktop application for a taxi booking system using Python and Tkinter. In order to design a useful and user-friendly application, I had to leverage my understanding of programming languages and software development concepts. I ran across a couple obstacles, but I was able to get beyond them by carefully debugging and troubleshooting. I also had the chance to improve my understanding of Python and tkinter and acquire new abilities.

Overall, I am pleased with the outcome and think this experience has helped me develop as a programmer. I am certain that the Taxi booking system application will be able to satisfy the demands of the intended audience and that it has the potential to be a helpful and handy tool for users.

References

Visual Paradigm (2021). What is Use Case Diagram. Retrieved from <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>. (Accessed on January 10, 2023)

IBM (2021). Structure Class Diagrams. Retrieved from

<https://www.ibm.com/docs/en/rsm/7.5.0?topic=structure-class-diagrams>. (Accessed on January 10, 2023)

Kumar, R. (2021). What is Pydantic? [online] DZone. Available at:

<https://dzone.com/articles/what-is-pydantic-1>. (Accessed 10 January 2023)

Schimansky, T. (n.d.). TkinterMapView. Retrieved from

<https://github.com/TomSchimansky/TkinterMapView>. (Accessed on January 10, 2023)

Appendix

PACKAGES AND MODULES

In a Python project, packages and modules are used to organize the code and make it more reusable. All the packages and modules created for this project are listed below:

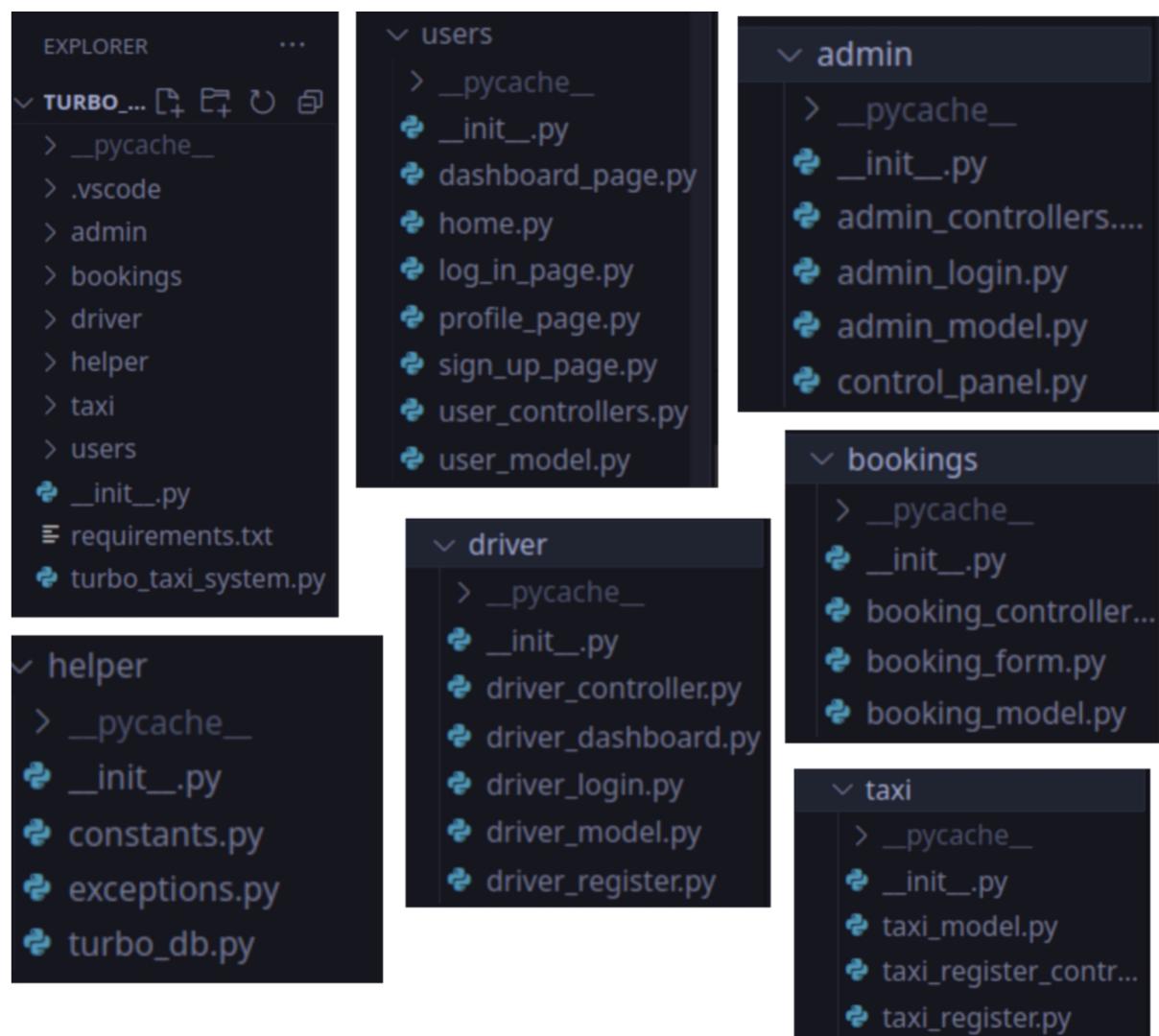


Figure 94: packages_and_modules

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

SOURCE CODE OF TURBO TAXI BOOKING SYSTEM

Package : turbo_taxi_booking_system.py

code :

```
from users.home import HomePage  
HomePage()
```

- **Module : helper**

Package : constants.py

code :

```
LOGO_LOCATION = "/home/vivu/Class Stuff V2/ISD_Files/TurboTB_System/img/logo.png"  
BG_LOCATION = "/home/vivu/Class Stuff V2/ISD_Files/TurboTB_System/img/background.png"  
ADMIN_LOGO_LOCATION = (  
    "/home/vivu/Class Stuff V2/ISD_Files/TurboTB_System/img/admin_logo.png"  
)  
DRIVER_LOGO_LOCATION = (  
    "/home/vivu/Class Stuff V2/ISD_Files/TurboTB_System/img/driver_logo.png"  
)
```

Package : exceptions.py

code :

```
class CustomException(Exception):  
    # raised when empty data  
    pass
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Package : turbo_db.py

code :

```
import psycopg2
```

```
class Turbo_db:  
    message = ""
```

```
@staticmethod  
def turbo_connection():  
    try:  
        conn = psycopg2.connect(  
            host="localhost",  
            dbname="turbo_db",  
            user="incri",  
            password="fastrack",  
            port=5432,  
        )  
        return conn
```

```
    except Exception as error:  
        Turbo_db.message = error
```

- **Module : users**

Package : home.py

code :

```
import tkinter as tk  
from tkinter import messagebox  
from helper.constants import (  
    LOGO_LOCATION,  
    BG_LOCATION,  
    ADMIN_LOGO_LOCATION,  
    DRIVER_LOGO_LOCATION,  
)  
from users import sign_up_page  
from users import log_in_page  
from admin.admin_login import LogInPage  
from driver.driver_login import DriverLogInPage
```

```
class HomePage:  
    def __init__(self):
```

```
        self.root = tk.Tk()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
self.build_root()  
self.create_home_frame()
```

```
def build_root(self):  
    self.root.title("Main Frame")  
    self.root.resizable(0, 0)  
    self.root.configure(background="#FFFFFF")  
    self.root.attributes("-fullscreen", True)
```

```
def create_home_frame(self):  
    self.home_frame = tk.Frame(self.root)  
    self.home_frame.configure(background="#FFFFFF")  
    self.home_frame.place(relx=-0.07, rely=0, height=768, width=1366)  
    self.build_home_frame(home_frame=self.home_frame, root=self.root)
```

```
@staticmethod  
def build_home_frame(home_frame, root):  
  
    top_title = tk.Label(home_frame)  
    top_title.place(relx=0.17, rely=0.068, height=51, width=153)  
    top_title.configure(  
        anchor="w",  
        background="#FFFFFF",  
        compound="left",  
        font="-family {Noto Sans} -size 22",  
        foreground="#3650D3",  
        text="""Turbo TB""",  
    )
```

```
hp_sub_title_1 = tk.Label(home_frame)  
hp_sub_title_1.place(relx=0.104, rely=0.163, height=61, width=353)  
hp_sub_title_1.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 18 -weight bold",  
    text="""Turbo Taxi Booking System""",  
)
```

```
hp_sub_title_2 = tk.Label(home_frame)  
hp_sub_title_2.place(relx=0.348, rely=0.176, height=41, width=396)  
hp_sub_title_2.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 18 -weight bold",  
    foreground="#3650D3",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    text=""""Online Taxi Booking""",  
    )
```

```
hp_sub_title_3 = tk.Label(home_frame)  
hp_sub_title_3.place(relx=0.104, rely=0.244, height=40, width=597)  
hp_sub_title_3.configure(activebackground="#f9f9f9")  
hp_sub_title_3.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 18 -weight bold",  
    foreground="#3650D3",  
    text=""""BEST TAXI SERVICE PROVIDER COMANY""",  
)
```

```
top_logo_lable = tk.Label(home_frame)  
top_logo_lable.place(relx=0.110, rely=0.068, height=71, width=80)  
top_logo_lable.configure(  
    anchor="w",  
    compound="left",  
    background="#FFFFFF",  
)  
logo_image = tk.PhotoImage(file=LOGO_LOCATION)  
top_logo_lable.configure(image=logo_image)
```

```
admin_logo_image = tk.PhotoImage(file=ADMIN_LOGO_LOCATION)  
sub_logo_label_1 = tk.Button(home_frame)  
sub_logo_label_1.place(relx=0.110, rely=0.407, height=71, width=80)  
sub_logo_label_1.configure(background="#FFFFFF", relief="flat")  
sub_logo_label_1.configure(  
    image=admin_logo_image,  
    command=lambda: HomePage.redirect_to_admin_page(root),  
)
```

```
info_level_1 = tk.Label(home_frame)  
info_level_1.place(relx=0.17, rely=0.393, height=39, width=153)  
info_level_1.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 12 -weight bold",  
    text=""""Login As Admin ?""",  
)
```

```
sub_info_level_1 = tk.Label(home_frame)  
sub_info_level_1.place(relx=0.17, rely=0.448, height=45, width=510)  
sub_info_level_1.configure(
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        anchor="w",
        background="#FFFFFF",
        compound="left",
        foreground="#637381",
    )
driver_logo_image = tk.PhotoImage(file=DRIVER_LOGO_LOCATION)
sub_info_level_1.configure(text="click on the admin icon to login as admin.")
```

```
sub_logo_label_2 = tk.Button(home_frame)
sub_logo_label_2.place(relx=0.110, rely=0.583, height=71, width=80)
sub_logo_label_2.configure(background="#FFFFFF", relief="flat")
sub_logo_label_2.configure(
    image=driver_logo_image,
    command=lambda: HomePage.redirect_to_driver_page(root),
)
```

```
info_level_2 = tk.Label(home_frame)
info_level_2.place(relx=0.17, rely=0.57, height=39, width=153)
info_level_2.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12 -weight bold",
    text="""Login As Driver ?""",
)
```

```
sub_info_level_2 = tk.Label(home_frame)
sub_info_level_2.place(relx=0.17, rely=0.624, height=45, width=510)
sub_info_level_2.configure(
    anchor="w",
    foreground="#637381",
    background="#FFFFFF",
    compound="left",
    text="click on the admin icon to login as driver.",
)
```

```
homebg_lable = tk.Label(home_frame)
homebg_lable.place(relx=0.538, rely=0.0, height=770, width=664)
homebg_lable.configure(anchor="w", compound="left")
background_image = tk.PhotoImage(file=BG_LOCATION)
homebg_lable.configure(image=background_image)
```

```
signup_redirect_button = tk.Button(home_frame)
signup_redirect_button.place(relx=0.783, rely=0.054, height=43, width=111)
signup_redirect_button.configure(
    activebackground="beige",
    background="#007074",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
borderwidth="2",
compound="left",
font="-family {Noto Sans} -size 12",
foreground="#FFFFFF",
text="""Sign Up""",
command=lambda: HomePage.redirect_to_signup_page(root),
)
```

```
login_redirect_button = tk.Button(home_frame)
login_redirect_button.place(relx=0.893, rely=0.054, height=43, width=111)
login_redirect_button.configure(
    activebackground="beige",
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#FFFFFF",
    text="""Log In""",
    command=lambda: HomePage.redirect_to_login_page(root),
)
```

```
app_exit_button = tk.Button(home_frame)
app_exit_button.place(relx=0.100, rely=0.865, height=43, width=111)
app_exit_button.configure(
    activebackground="beige",
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#FFFFFF",
    text="""EXIT""",
    command=lambda: HomePage.app_exit(root),
)
```

```
home_frame.mainloop()
```

```
@staticmethod
def app_exit(root):
    response = messagebox.askquestion(
        title="Exit !", message="Do you really want to exit ?", icon="warning"
    )
    if response == "yes":
        root.destroy()
```

```
@staticmethod
def redirect_to_signup_page(root):
    sign_up_page.SignUpPage(root)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
@staticmethod  
def redirect_to_admin_page(root):  
    LogInPage(root)
```

```
@staticmethod  
def redirect_to_driver_page(root):  
    DriverLogInPage(root)
```

```
@staticmethod  
def redirect_to_login_page(root):  
    log_in_page.LogInPage(root)
```

Package : sign_up_page.py

code :

```
import tkinter as tk  
from tkinter import messagebox  
  
from pydantic import ValidationError  
from helper.constants import BG_LOCATION  
from .user_model import UserModel  
from .user_controllers import UserController  
from helper.exceptions import CustomException
```

```
class SignUpPage:  
    def __init__(self, root):  
        self.root = root  
        self.create_signup_frame()
```

```
    def create_signup_frame(self):  
        self.signup_frame = tk.Frame(self.root)  
        self.signup_frame.configure(background="#FFFFFF")  
        self.signup_frame.place(relx=-0.07, rely=0, height=768, width=1366)  
        self.build_signup_frame(signup_frame=self.signup_frame)
```

```
@staticmethod  
def build_signup_frame(signup_frame):  
  
    welcome = tk.Label(signup_frame)  
    welcome.place(relx=0.593, rely=0.090, height=60, width=323)  
    welcome.configure(  
        activebackground="#f9f9f9",  
        anchor="w",  
        background="#FFFFFF",  
        compound="left",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    text="""Welcome To Turbo TB""",
    font="-family {Noto Sans} -size 22 -weight bold",
)
```

```
quoest = tk.Label(signup_frame)
quoest.place(relx=0.644, rely=0.160, height=20, width=205)
quoest.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14 -slant italic",
    text="""Travel For Your Heart""",
)
```

```
firstname_label = tk.Label(signup_frame)
firstname_label.place(relx=0.538, rely=0.231, height=20, width=135)
firstname_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    font="-family {Noto Sans} -size 16",
    compound="left",
    background="#FFFFFF",
    text="""First Name""",
)
```

```
firstname_entry = tk.Entry(signup_frame)
firstname_entry.place(relx=0.538, rely=0.27, height=33, relwidth=0.189)
firstname_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
)
```

```
lastname_label = tk.Label(signup_frame)
lastname_label.place(relx=0.773, rely=0.231, height=20, width=135)
lastname_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 16",
    text="""Last Name""",
)
```

```
lastname_entry = tk.Entry(signup_frame)
lastname_entry.place(relx=0.773, rely=0.27, height=33, relwidth=0.189)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
lastname_entry.configure(  
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"  
)
```

```
contact_label = tk.Label(signup_frame)  
contact_label.place(relx=0.538, rely=0.338, height=21, width=135)  
contact_label.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    compound="left",  
    background="#FFFFFF",  
    font="-family {Noto Sans} -size 16",  
    text="\"Contact\"",  
)
```

```
contact_entry = tk.Entry(signup_frame)  
contact_entry.place(relx=0.538, rely=0.379, height=33, relwidth=0.189)  
contact_entry.configure(  
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"  
)
```

```
address_label = tk.Label(signup_frame)  
address_label.place(relx=0.773, rely=0.338, height=21, width=135)  
address_label.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 16",  
    text="\"Address\"",  
)
```

```
address_entry = tk.Entry(signup_frame)  
address_entry.place(relx=0.773, rely=0.379, height=33, relwidth=0.189)  
address_entry.configure(  
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"  
)
```

```
email_label = tk.Label(signup_frame)  
email_label.place(relx=0.538, rely=0.446, height=20, width=185)  
email_label.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 16",  
    text="\"Email Address\"",  
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

[REDACTED]

```
email_entry = tk.Entry(signup_frame)
email_entry.place(relx=0.538, rely=0.486, height=33, relwidth=0.419)
email_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
)
```

```
username_label = tk.Label(signup_frame)
username_label.place(relx=0.538, rely=0.554, height=31, width=135)
username_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 16",
    text="User Name",
)
```

```
username_entry = tk.Entry(signup_frame)
username_entry.place(relx=0.538, rely=0.608, height=33, relwidth=0.189)
username_entry.configure(
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"
)
```

```
password_label = tk.Label(signup_frame)
password_label.place(relx=0.773, rely=0.554, height=21, width=135)
password_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 16",
    text="Password",
)
```

```
password_entry = tk.Entry(signup_frame)
password_entry.place(relx=0.773, rely=0.608, height=33, relwidth=0.189)
password_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
    show="*",
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
confrim_password_label = tk.Label(signup_frame)
confrim_password_label.place(relx=0.533, rely=0.678, height=20, width=205)
confrim_password_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 16",
    text="Confirm Password",
)
```

```
confrim_password_entry = tk.Entry(signup_frame)
confrim_password_entry.place(relx=0.538, rely=0.73, height=33, relwidth=0.189)
confrim_password_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
    show="*",
)
```

```
show_pass_button = tk.Button(signup_frame)
show_pass_button.place(relx=0.74, rely=0.735, height=23, width=51)
show_pass_button.configure(
    activebackground="beige",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 14",
    text="OO",
)
show_pass_button.bind(
    "<Button-1>",
    lambda event: SignUpPage.show_pass(
        event,
        password_entry,
        confrim_password_entry,
    ),
)
```

```
show_pass_button.bind(
    "<ButtonRelease-1>",
    lambda event: SignUpPage.hide_pass(
        event,
        password_entry,
        confrim_password_entry,
    ),
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
signup_button = tk.Button(signup_frame)
signup_button.place(relx=0.666, rely=0.855, height=43, width=201)
signup_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12 -weight bold",
    foreground="#FFFFFF",
    text="Sign Up",
    command=lambda: SignUpPage.user_register(
        firstname_entry,
        lastname_entry,
        contact_entry,
        address_entry,
        email_entry,
        username_entry,
        password_entry,
        confrim_password_entry,
        signup_frame,
    ),
)
```

```
background_label = tk.Label(signup_frame)
background_label.place(relx=-0.029, rely=0.0, height=770, width=700)
background_label.configure(anchor="w", compound="left")
background_image = tk.PhotoImage(file=BG_LOCATION)
background_label.configure(image=background_image)
```

```
back_button = tk.Button(signup_frame)
back_button.place(relx=0.100, rely=0.865, height=43, width=111)
back_button.configure(
    activebackground="beige",
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#FFFFFF",
    text="Back",
    command=lambda: SignUpPage.redirect_to_homepage(signup_frame),
)
```

```
signup_frame.mainloop()
```

```
@staticmethod
def show_pass(
    event,
    password_entry,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    confirm_password_entry,  
):  
    password = password_entry.get()  
    confirm_password = confirm_password_entry.get()  
    password_entry.config(show="", text=password)  
    confirm_password_entry.config(show="", text=confirm_password)
```

```
@staticmethod  
def hide_pass(  
    event,  
    password_entry,  
    confirm_password_entry,  
):  
  
    password_entry.config(show="*")  
    confirm_password_entry.config(show="*")
```

```
@staticmethod  
def redirect_to_homepage(signup_frame):  
    signup_frame.destroy()
```

```
@staticmethod  
def user_register(  
    firstname_entry,  
    lastname_entry,  
    contact_entry,  
    address_entry,  
    email_entry,  
    username_entry,  
    password_entry,  
    confirm_password_entry,  
    signup_frame,  
):  
    try:  
        user = UserModel(  
            firstname=firstname_entry.get(),  
            lastname=lastname_entry.get(),  
            contact=contact_entry.get(),  
            address=address_entry.get(),  
            email=email_entry.get(),  
            username=username_entry.get(),  
            password=password_entry.get(),  
            confirm_password=confirm_password_entry.get(),  
        )  
        user_control = UserController()  
        user_control.registration_control(user, signup_frame)  
    except CustomException as e:
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
messagebox.showerror("Invalid Data", e)
```

Package : log_in_page.py

code :

```
import imp
import tkinter as tk
from tkinter import messagebox
from helper.constants import BG_LOCATION
from .user_model import UserModel
from .user_controllers import UserController
from helper.exceptions import CustomException
```

```
class LogInPage:
    def __init__(self, root):
        self.root = root
        self.create_login_frame()
```

```
def create_login_frame(self):
    self.login_frame = tk.Frame(self.root)
    self.login_frame.configure(background="#FFFFFF")
    self.login_frame.place(relx=-0.07, rely=0, height=768, width=1366)
    self.build_login_frame(login_frame=self.login_frame, root=self.root)
```

```
@staticmethod
def build_login_frame(login_frame, root):
```

```
welcome = tk.Label(login_frame)
welcome.place(relx=0.696, rely=0.068, height=59, width=114)
welcome.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 22 -weight bold",
    text="Log In",
)
```

```
quoest = tk.Label(login_frame)
quoest.place(relx=0.674, rely=0.149, height=20, width=185)
quoest.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14 -slant italic",
    text="Welcome Back User",
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
username_label = tk.Label(login_frame)
username_label.place(relx=0.585, rely=0.299, height=21, width=206)
username_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 16",
    text="Username",
)
```

```
username_entry = tk.Entry(login_frame)
username_entry.place(relx=0.585, rely=0.353, height=33, relwidth=0.301)
username_entry.configure(
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"
)
```

```
password_label = tk.Label(login_frame)
password_label.place(relx=0.585, rely=0.434, height=20, width=185)
password_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 16",
    text="Password",
)
```

```
password_entry = tk.Entry(login_frame)
password_entry.place(relx=0.585, rely=0.488, height=33, relwidth=0.301)
password_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
    show="*",
)
```

```
login_button = tk.Button(login_frame)
login_button.place(relx=0.674, rely=0.611, height=43, width=201)
login_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12 -weight bold",
    foreground="#FFFFFF",
    text="Log In",
    command=lambda: LogInPage.user_login(
        username_entry,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
password_entry,  
login_frame,  
root,  
)  
)
```

```
background_label = tk.Label(login_frame)  
background_label.place(relx=-0.029, rely=0.0, height=739, width=700)  
background_label.configure(anchor="w", compound="left")  
background_image = tk.PhotoImage(file=BG_LOCATION)  
background_label.configure(image=background_image)
```

```
back_button = tk.Button(login_frame)  
back_button.place(relx=0.100, rely=0.865, height=43, width=111)  
back_button.configure(  
    activebackground="beige",  
    background="#007074",  
    borderwidth="2",  
    compound="left",  
    font="-family {Noto Sans} -size 12",  
    foreground="#FFFFFF",  
    text="Back",  
    command=lambda: LogInPage.redirect_to_home(login_frame),  
)
```

```
show_pass_button = tk.Button(login_frame)  
show_pass_button.place(relx=0.900, rely=0.500, height=23, width=51)  
show_pass_button.configure(  
    activebackground="beige",  
    borderwidth="2",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    text="OO",  
)
```

```
show_pass_button.bind(  
    "<Button-1>",  
    lambda event: LogInPage.show_pass(  
        event,  
        password_entry,  
)  
)
```

```
show_pass_button.bind(  
    "<ButtonRelease-1>",  
    lambda event: LogInPage.hide_pass(  
        event,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        password_entry,
    ),
)
login_frame.mainloop()
```

```
@staticmethod
def show_pass(
    event,
    password_entry,
):
    password = password_entry.get()
    password_entry.config(show="", text=password)
```

```
@staticmethod
def hide_pass(
    event,
    password_entry,
):
    password_entry.config(show="*")
```

```
def redirect_to_home(login_frame):
    login_frame.destroy()
```

```
@staticmethod
def user_login(
    username_entry,
    password_entry,
    login_frame,
    root,
):
```

```
try:
    user = UserModel(
        username=username_entry.get(),
        password=password_entry.get(),
        confirm_password=password_entry.get(),
    )
    login_frame.destroy()
    user_control = UserController()
    user_control.login_control(user, root)
except CustomException as e:
    messagebox.showerror("Invalid Data", e)
```

Package : dashboard_page.py

code :

```
import imp
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
import tkinter as tk
from tkinter import NO, W, Scrollbar, ttk
from helper.constants import LOGO_LOCATION
from PIL import Image, ImageTk
from io import BytesIO
from .profile_page import ProfilePage
from bookings.booking_form import BookingPage
from .user_model import UserModel
```

```
class DashboardPage:
    def __init__(self, root, record, user_controller):
        self.root = root
        self.record = record
        self.user_controller = user_controller
        self.create_dashboard_frame()
```

```
def create_dashboard_frame(self):
    self.dashboard_frame = tk.Frame(self.root)
    self.dashboard_frame.configure(background="#FFFFFF")
    self.dashboard_frame.place(relx=-0.07, rely=0, height=768, width=1366)
    self.build_dashboard_frame(
        dashboard_frame=self.dashboard_frame,
        root=self.root,
        record=self.record,
        user_controller=self.user_controller,
    )
```

```
@staticmethod
def build_dashboard_frame(
    dashboard_frame,
    root,
    record,
    user_controller,
):
```

```
    title = tk.Label(dashboard_frame)
    title.place(relx=0.452, rely=0.285, height=40, width=212)
    title.configure(
        anchor="w",
        background="#FFFFFF",
        compound="left",
        font="-family {Noto Sans} -size 18 -weight bold",
        foreground="#3056D3",
        text="Your Trip History",
    )
```

```
    profile_label = tk.Label(dashboard_frame)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
profile_label.place(relx=0.91, rely=0.068, height=61, width=70)
profile_label.configure(
    anchor="w",
    compound="left",
    borderwidth=2,
    relief="solid",
)
```

```
for data in record:
    fetched_username = data[6].rstrip()
    fetched_profile_picture = data[8]
```

```
profile_img = Image.open(BytesIO(fetched_profile_picture))
small_img = profile_img.resize((70, 61))
view_profile = ImageTk.PhotoImage(small_img)
profile_label.configure(image=view_profile)
```

```
logo_label = tk.Label(dashboard_frame)
logo_label.place(relx=0.110, rely=0.068, height=71, width=80)
logo_label.configure(
    anchor="w",
    compound="left",
    text="""Logo""",
    background="#FFFFFF",
)
photo_location = tk.PhotoImage(file=LOGO_LOCATION)
logo_label.configure(image=photo_location)
```

```
username_button = tk.Button(dashboard_frame)
username_button.place(relx=0.799, rely=0.095, height=33, width=131)
username_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    text=fetched_username,
    font="-family {Noto Sans} -size 12 -weight bold",
    foreground="#FFFFFF",
    command=lambda: DashboardPage.redirect_to_profile_page(
        root,
        dashboard_frame,
        record,
        user_controller,
    ),
)
```

```
tabel_label = tk.Label(dashboard_frame)
tabel_label.place(relx=0.110, rely=0.407, height=229, width=1160)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    tabel_label.configure(anchor="w", compound="left", text="Table")
```

```
book_now_button = tk.Button(dashboard_frame)
book_now_button.place(relx=0.466, rely=0.814, height=43, width=171)
book_now_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#FFFFFF",
    text="Book Now",
    command=lambda: DashboardPage.redirect_to_booking_form(
        root,
        dashboard_frame,
        record,
    ),
)
```

```
refresh_button = tk.Button(dashboard_frame)
refresh_button.place(relx=0.799, rely=0.814, height=33, width=131)
refresh_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    text="Refresh Page",
    font="-family {Noto Sans} -size 12 -weight bold",
    foreground="#FFFFFF",
    command=lambda: DashboardPage.refresh_page(
        user_controller,
        record,
        root,
        dashboard_frame,
    ),
)
```

```
DashboardPage.booking_history_frame(dashboard_frame, record, user_controller)
```

```
    dashboard_frame.mainloop()
```

```
@staticmethod
def redirect_to_profile_page(
    root,
    dashboard_frame,
    record,
    user_controller,
):
    ProfilePage(root, dashboard_frame, record, user_controller)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
@staticmethod
def redirect_to_booking_form(
    root,
    dashboard_frame,
    record,
):
    BookingPage(
        root,
        dashboard_frame,
        record,
    )
```

```
@staticmethod
def refresh_page(user_controller, record, root, dashboard_frame):
```

```
    dashboard_frame.destroy()
    for data in record:
        fetched_username = data[6]
        fetched_password = data[7]
```

```
    user = UserModel(
        username=fetched_username,
        password=fetched_password,
        confirm_password=fetched_password,
    )
    user_control = user_controller()
    user_control.login_control(user, root)
```

```
@staticmethod
def booking_history_frame(dashboard_frame, record, user_controller):
```

```
    for data in record:
        userid = data[0]
```

```
    history_table_frame = tk.Frame(dashboard_frame, bg="#FFFFFF")
    history_table_frame.place(
        relx=0.110,
        rely=0.407,
        height=229,
        width=1160,
    )
    table_style = ttk.Style()
    table_style.theme_use("default")
    table_style.configure(
        "Treeview",
        background="#FFFFFF",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
foreground="#4A4A4A",
rowheight="35",
fieldbackground="#FFFFFF",
)
table_style.map("Treeview", background=[("selected", "#C9C9C9")])
```

```
table_scroll_bar = Scrollbar(history_tabel_frame)
table_scroll_bar.place(relx=0.980, rely=0, height=229, width=15)
```

```
history_booking_table = ttk.Treeview(
    history_tabel_frame,
    yscrollcommand=table_scroll_bar.set,
    selectmode="extended",
)
history_booking_table.place(
    relx=0,
    rely=0,
    height=229,
    width=1120,
)
```

```
table_scroll_bar.config(
    command=history_booking_table.yview,
)
```

```
history_booking_table["columns"] = (
    "Booking ID",
    "Full Name",
    "Date and Time",
    "Destination",
    "Total Cost",
    "Status",
)
```

```
history_booking_table.column("#0", width=0, stretch=NO)
history_booking_table.column("Booking ID", width=20, anchor=W)
history_booking_table.column("Full Name", width=70, anchor=W)
history_booking_table.column("Date and Time", width=70, anchor=W)
history_booking_table.column("Destination", width=240, anchor=W)
history_booking_table.column("Total Cost", width=40, anchor=W)
history_booking_table.column("Status", width=40, anchor=W)
```

```
history_booking_table.heading("#0", text="", anchor=W)
history_booking_table.heading("Booking ID", text="Booking ID", anchor=W)
history_booking_table.heading("Full Name", text="Full Name", anchor=W)
history_booking_table.heading("Date and Time", text="Date and Time", anchor=W)
history_booking_table.heading("Destination", text="Destination", anchor=W)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
history_booking_table.heading("Total Cost", text="Total Cost", anchor=W)
history_booking_table.heading("Status", text="Status", anchor=W)
```

```
history_trip_control = user_controller()
trip_fetched_data = history_trip_control.history_trip_detail_fetcher(userid)
```

```
for data in trip_fetched_data:
```

```
    history_booking_table.insert(
        "", index="end",
        values=(
            data[0],
            (data[2], data[3]),
            (data[6], "---", data[7], ":", data[8]),
            data[10],
            data[11],
            data[16],
        ),
    )
```

Package : profile_page.py

code :

```
import tkinter as tk
from tkinter import NO, W, Scrollbar, messagebox, filedialog
from tkinter import ttk
from helper.constants import LOGO_LOCATION
from PIL import Image, ImageTk
from io import BytesIO
from .user_model import UserModel
from helper.exceptions import CustomException
from tkintermapview import TkinterMapView
```

```
class ProfilePage:
    def __init__(self, root, dashboard_frame, record, user_controller):
        self.root = root
        self.dashboard_frame = dashboard_frame
        self.record = record
        self.user_controller = user_controller
        self.create_profile_frame()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
def create_profile_frame(self):
    self.profile_frame = tk.Frame(self.root)
    self.profile_frame.configure(background="#FFFFFF")
    self.profile_frame.place(
        relx=-0.010,
        rely=0,
        height=768,
        width=1366,
    )
    self.build_profile_frame(
        profile_frame=self.profile_frame,
        root=self.root,
        dashboard_frame=self.dashboard_frame,
        record=self.record,
        user_controller=self.user_controller,
    )
```

```
@staticmethod
def build_profile_frame(
    profile_frame,
    root,
    dashboard_frame,
    record,
    user_controller,
):
```

```
    logo = tk.Label(profile_frame)
    logo.place(x=91, y=60, height=81, width=82)
    logo.configure(
        anchor="w",
        compound="left",
        text="logo",
        background="#FFFFFF",
    )
```

```
    photo_location = tk.PhotoImage(file=LOGO_LOCATION)
    logo.configure(image=photo_location)
```

```
    logout_button = tk.Button(profile_frame)
    logout_button.place(x=1159, y=60, height=43, width=111)
    logout_button.configure(
        background="#007074",
        borderwidth="2",
        compound="left",
        font="-family {Noto Sans} -size 12",
        foreground="#FFFFFF",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
text=""""Log Out""",  
command=lambda: ProfilePage.log_out(  
    profile_frame,  
    dashboard_frame,  
)  
)
```

```
for data in record:  
    featched_username = data[6].rstrip()  
    fetched_email = data[5].rstrip()  
    fetched_address = data[4].rstrip()  
    fetched_contact = data[3].rstrip()  
    fetched_profile = data[8]
```

```
profile_img = Image.open(BytesIO(fetched_profile))  
big_img = profile_img.resize((153, 133))  
view_profile = ImageTk.PhotoImage(big_img)
```

```
profile_photo = tk.Label(profile_frame)  
profile_photo.place(x=59, y=210, height=133, width=153)  
profile_photo.configure(  
    anchor="w",  
    compound="left",  
    text=""""profile""",  
    borderwidth=2,  
    relief="solid",  
)  
profile_photo.configure(image=view_profile)
```

```
username = tk.Label(profile_frame)  
username.place(x=259, y=210, height=31, width=115)  
username.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 16 -weight bold",  
    text=featched_username,  
)
```

```
email = tk.Label(profile_frame)  
email.place(x=259, y=260, height=21, width=67)  
email.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 12",  
    text=""""Email :""",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

[REDACTED]

```
contact = tk.Label(profile_frame)
contact.place(x=259, y=290, height=20, width=95)
contact.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    text="Phone no :",
)
```

```
address = tk.Label(profile_frame)
address.place(x=259, y=320, height=20, width=87)
address.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    text="Address :",
)
```

```
email_data = tk.Label(profile_frame)
email_data.place(x=320, y=255, height=30, width=300)
email_data.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    text=fetched_email,
)
```

```
contact_data = tk.Label(profile_frame)
contact_data.place(x=359, y=290, height=20, width=95)
contact_data.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    text=fetched_contact,
)
```

```
address_data = tk.Label(profile_frame)
address_data.place(x=350, y=317, height=30, width=300)
address_data.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    text=fetched_address,
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        compound="left",
        font="-family {Noto Sans} -size 12",
        text=fetched_address,
    )
```

```
edit_profile_picture = tk.Button(profile_frame)
edit_profile_picture.place(x=59, y=380, height=23, width=91)
edit_profile_picture.configure(
    background="#FFFFFF",
    borderwidth="2",
    compound="left",
    text="Edit Profile >",
    command=lambda: ProfilePage.edit_account_profile(
        user_controller,
        record,
        profile_frame,
        dashboard_frame,
    ),
)
```

```
edit_bio_button = tk.Button(profile_frame)
edit_bio_button.place(x=259, y=380, height=23, width=101)
edit_bio_button.configure(
    background="#FFFFFF",
    borderwidth="2",
    compound="left",
    text="Edit Bio >",
    command=lambda: ProfilePage.create_edit_bio_page(
        record,
        user_controller,
        profile_frame,
        dashboard_frame,
    ),
)
```

```
upcoming_trips = tk.Label(profile_frame)
upcoming_trips.place(x=550, y=440, height=41, width=245)
upcoming_trips.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 16",
    foreground="#3056D3",
    text="Your Upcoming Trips",
)
```

```
back_button = tk.Button(profile_frame)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
back_button.place(x=50, y=630, height=43, width=111)
back_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#FFFFFF",
    text="Back",
    command=lambda: ProfilePage.redirect_to_dashboard(profile_frame),
)
```

```
ProfilePage.upcoming_trip_table_viewer(
    profile_frame,
    user_controller,
    record,
)
```

```
profile_frame.mainloop()
```

```
@staticmethod
def redirect_to_dashboard(profile_frame):
    profile_frame.destroy()
```

```
@staticmethod
def log_out(profile_frame, dashboard_frame):
    response = messagebox.askquestion(
        title="Log Out !",
        message="Do you really want to Log Out ?",
        icon="warning",
    )
```

```
if response == "yes":
    profile_frame.destroy()
    dashboard_frame.destroy()
```

```
@staticmethod
def create_edit_bio_page(
    record,
    user_controller,
    profile_frame,
    dashboard_frame,
):
    edit_bio_frame = tk.Toplevel(profile_frame)
    edit_bio_frame.title("Edit Bio")
    edit_bio_frame.resizable(0, 0)
    edit_bio_frame.configure(background="#FFFFFF")
    edit_bio_frame.geometry("460x579+420+91")
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
ProfilePage.build_edit_bio_page(  
    edit_bio_frame,  
    record,  
    user_controller,  
    profile_frame,  
    dashboard_frame,  
)
```

```
@staticmethod  
def build_edit_bio_page(  
    edit_bio_frame,  
    record,  
    user_controller,  
    profile_frame,  
    dashboard_frame,  
):
```

```
welcome = tk.Label(edit_bio_frame)  
welcome.place(relx=0.413, rely=0.069, height=41, width=104)  
welcome.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    text="Edit Bio",  
    font="-family {Noto Sans} -size 16 -weight bold",  
)
```

```
for data in record:  
    fetched_firstname = data[1].rstrip()  
    fetched_lastname = data[2].rstrip()  
    fetched_email = data[5].rstrip()
```

```
firstname = tk.Label(edit_bio_frame)  
firstname.place(relx=0.065, rely=0.19, height=41, width=104)  
firstname.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    font="-family {Noto Sans} -size 14",  
    compound="left",  
    background="#FFFFFF",  
    text="First Name",  
)
```

```
firstname_entry = tk.Entry(edit_bio_frame)  
firstname_entry.insert(0, fetched_firstname)  
firstname_entry.place(
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
    relx=0.065,  
    rely=0.259,  
    height=23,  
    relwidth=0.404,  
)  
firstname_entry.configure(  
    background="#EFF0F2",  
    font="TkFixedFont",  
    selectbackground="#c4c4c4",  
    state="disabled",  
)  
  
lastname = tk.Label(edit_bio_frame)  
lastname.place(relx=0.543, rely=0.19, height=41, width=104)  
lastname.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    text="Last Name",  
)
```

```
lastname_entry = tk.Entry(edit_bio_frame)  
lastname_entry.insert(0, fetched_lastname)  
lastname_entry.place(  
    relx=0.543,  
    rely=0.259,  
    height=23,  
    relwidth=0.404,  
)  
lastname_entry.configure(  
    background="#EFF0F2",  
    font="TkFixedFont",  
    selectbackground="#c4c4c4",  
    state="disabled",  
)
```

```
contact = tk.Label(edit_bio_frame)  
contact.place(relx=0.065, rely=0.311, height=41, width=104)  
contact.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    compound="left",  
    background="#FFFFFF",  
    font="-family {Noto Sans} -size 14",  
    text="Contact",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

[REDACTED]

```
contact_entry = tk.Entry(edit_bio_frame)
contact_entry.place(
    relx=0.065,
    rely=0.38,
    height=23,
    relwidth=0.404,
)
contact_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
)
```

```
address = tk.Label(edit_bio_frame)
address.place(relx=0.543, rely=0.311, height=41, width=104)
address.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    text="Address",
)
```

```
address_entry = tk.Entry(edit_bio_frame)
address_entry.place(
    relx=0.543,
    rely=0.38,
    height=23,
    relwidth=0.404,
)
address_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
)
```

```
email = tk.Label(edit_bio_frame)
email.place(relx=0.065, rely=0.432, height=41, width=300)
email.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
    text=""""Email Address""",  
    )
```

```
email_entry = tk.Entry(edit_bio_frame)  
email_entry.insert(0, fetched_email)  
email_entry.place(  
    relx=0.065,  
    rely=0.501,  
    height=23,  
    relwidth=0.883,  
)  
email_entry.configure(  
    background="#EFF0F2",  
    font="TkFixedFont",  
    selectbackground="#c4c4c4",  
    state="disabled",  
)
```

```
username = tk.Label(edit_bio_frame)  
username.place(relx=0.065, rely=0.553, height=41, width=104)  
username.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    text=""""User Name""",  
)
```

```
username_entry = tk.Entry(edit_bio_frame)  
username_entry.place(  
    relx=0.065,  
    rely=0.622,  
    height=23,  
    relwidth=0.404,  
)  
username_entry.configure(  
    background="#EFF0F2",  
    font="TkFixedFont",  
    selectbackground="#c4c4c4",  
)
```

```
password = tk.Label(edit_bio_frame)  
password.place(relx=0.543, rely=0.553, height=41, width=300)  
password.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    )
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
        background="#FFFFFF",
        compound="left",
        font="-family {Noto Sans} -size 14",
        text="""Old Password""",
    )
```

```
password_entry = tk.Entry(edit_bio_frame)
password_entry.place(
    relx=0.543,
    rely=0.622,
    height=23,
    relwidth=0.404,
)
password_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
    show="*",
)
```

```
confirm_password = tk.Label(edit_bio_frame)
confirm_password.place(
    relx=0.065,
    rely=0.674,
    height=41,
    width=300,
)
confirm_password.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    text="""New Password""",
)
```

```
confrim_password_entry = tk.Entry(edit_bio_frame)
confrim_password_entry.place(
    relx=0.065,
    rely=0.76,
    height=23,
    relwidth=0.404,
)
confrim_password_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
    show="*",
)
)
```

```
show_password = tk.Button(edit_bio_frame)
show_password.place(
    relx=0.5,
    rely=0.757,
    height=23,
    width=51,
)
show_password.configure(
    activebackground="beige",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 14",
    text="OO",
)
show_password.bind(
    "<Button-1>",
    lambda event: ProfilePage.show_pass(
        event,
        password_entry,
        confirm_password_entry,
    ),
)
```

```
show_password.bind(
    "<ButtonRelease-1>",
    lambda event: ProfilePage.hide_pass(
        event,
        password_entry,
        confirm_password_entry,
    ),
)
```

```
update_button = tk.Button(edit_bio_frame)
update_button.place(relx=0.522, rely=0.864, height=43, width=111)
update_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12 -weight bold",
    foreground="#FFFFFF",
    text="Update",
    command=lambda: ProfilePage.edit_user_bio(
        record,
        edit_bio_frame,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
profile_frame,  
dashboard_frame,  
user_controller,  
contact_entry,  
address_entry,  
username_entry,  
password_entry,  
confrim_password_entry,  
)  
)
```

```
delete_button = tk.Button(edit_bio_frame)  
delete_button.place(relx=0.690, rely=0.757, height=23, width=120)  
delete_button.configure(  
    activebackground="beige",  
    borderwidth="2",  
    compound="left",  
    font="-family {Noto Sans} -size 10",  
    text="Delete Account",  
    command=lambda: ProfilePage.delete_account(  
        user_controller,  
        profile_frame,  
        dashboard_frame,  
        edit_bio_frame,  
        record,  
)  
)
```

```
cancel_button = tk.Button(edit_bio_frame)  
cancel_button.place(relx=0.250, rely=0.864, height=43, width=111)  
cancel_button.configure(  
    activebackground="beige",  
    background="#007074",  
    borderwidth="2",  
    compound="left",  
    font="-family {Noto Sans} -size 12",  
    foreground="#FFFFFF",  
    text="Cancel",  
    command=lambda: ProfilePage.cancel(edit_bio_frame),  
)
```

```
edit_bio_frame.grab_set()  
edit_bio_frame.mainloop()
```

```
@staticmethod  
def show_pass(  
    event,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
password_entry,  
confrim_password_entry,  
):  
    password = password_entry.get()  
    confrim_password = confrim_password_entry.get()  
    password_entry.config(show="", text=password)  
    confrim_password_entry.config(show="", text=confrim_password)
```

```
@staticmethod  
def hide_pass(  
    event,  
    password_entry,  
    confrim_password_entry,  
):  
    password_entry.config(show="*")  
    confrim_password_entry.config(show="*")
```

```
@staticmethod  
def cancel(edit_bio_frame):  
    edit_bio_frame.destroy()
```

```
@staticmethod  
def delete_account(  
    user_controller,  
    profile_frame,  
    dashboard_frame,  
    edit_bio_frame,  
    record,  
):  
    account_control = user_controller()  
    response = messagebox.askquestion(  
        "Delete Account",  
        "Do you really want to delete your account? \n \\  
        you wont be able to recover your account again",  
    )  
    if response == "yes":  
        account_control.delete_account(  
            profile_frame,  
            dashboard_frame,  
            edit_bio_frame,  
            record,  
        )
```

```
@staticmethod  
def edit_account_profile(  
    user_controller,  
    record,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
profile_frame,  
dashbaard_frame,  
):  
for data in record:  
    fetched_password = data[7]
```

```
file = filedialog.askopenfilename()  
if file:  
    user = UserModel(  
        profile=file,  
        password=fetched_password,  
        confirm_password=fetched_password,  
    )  
    profile_picture_control = user_controller()  
    profile_picture_control.change_profile(  
        user,  
        record,  
        profile_frame,  
        dashboaard_frame,  
    )
```

```
@staticmethod  
def edit_user_bio(  
    record,  
    edit_bio_frame,  
    profile_frame,  
    dashboard_frame,  
    user_controller,  
    contact_entry,  
    address_entry,  
    username_entry,  
    password_entry,  
    confrim_password_entry,  
):
```

```
for data in record:  
    fetched_contact = data[3]  
    fetched_address = data[4]  
    fetched_username = data[6]  
    fetched_password = data[7]
```

```
# checking if data is empty or not  
  
if contact_entry.get() == "":  
    new_contact = fetched_contact  
else:  
    new_contact = contact_entry.get()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
if address_entry.get() == "":
    new_address = fetched_address
else:
    new_address = address_entry.get()
if username_entry.get() == "":
    new_username = fetched_username
else:
    new_username = username_entry.get()
if confirm_password_entry.get() == "":
    new_password_update = fetched_password
else:
    new_password_update = confirm_password_entry.get()
if password_entry.get() == fetched_password:
    old_password = password_entry.get()
else:
    old_password = password_entry.get()
```

```
try:
    user = UserModel(
        contact=new_contact,
        address=new_address,
        username=new_username,
        password=old_password,
        confirm_password=old_password,
        new_password=new_password_update,
    )
```

```
user_bio_control = user_controller()
user_bio_control.update_user_bio(
    user,
    record,
    edit_bio_frame,
    profile_frame,
    dashboard_frame,
```

```
except CustomException as e:
    messagebox.showerror("Invalid Data", e, parent=edit_bio_frame)
```

```
@staticmethod
def upcoming_trip_table_viewer(profile_frame, user_controller, record):
```

```
for data in record:
    userid = data[0]
```

```
upcoming_trip_table = tk.Frame(profile_frame, bg="#FFFFFF")
upcoming_trip_table.place(
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
x=49,  
y=490,  
height=130,  
width=1212,  
)
```

```
# Upcoming booking details table
```

```
table_style = ttk.Style()  
table_style.theme_use("default")  
table_style.configure(  
    "Treeview",  
    background="#FFFFFF",  
    foreground="#4A4A4A",  
    rowheight="35",  
    fieldbackground="#FFFFFF",  
)  
table_style.map("Treeview", background=[("selected", "#C9C9C9")])
```

```
# driver Table Scroll Bar
```

```
table_scroll_bar = Scrollbar(upcoming_trip_table)  
table_scroll_bar.place(relx=0.985, rely=0, height=130, width=15)
```

```
# driver Table Build
```

```
upcoming_booking_table = ttk.Treeview(  
    upcoming_trip_table,  
    yscrollcommand=table_scroll_bar.set,  
    selectmode="extended",  
)  
upcoming_booking_table.place(  
    relx=0,  
    rely=0,  
    height=280,  
    width=1180,  
)  
# upcoming a scroll bar
```

```
table_scroll_bar.config(  
    command=upcoming_booking_table.yview,  
)
```

```
upcoming_booking_table["columns"] = (  
    "Booking ID",  
    "Full Name",  
    "Date and Time",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
"Destination",
"Total Cost",
>Status",
)
```

Format Our Columns

```
upcoming_booking_table.column("#0", width=0, stretch=NO)
upcoming_booking_table.column("Booking ID", width=20, anchor=W)
upcoming_booking_table.column("Full Name", width=70, anchor=W)
upcoming_booking_table.column("Date and Time", width=70, anchor=W)
upcoming_booking_table.column("Destination", width=240, anchor=W)
upcoming_booking_table.column("Total Cost", width=40, anchor=W)
upcoming_booking_table.column("Status", width=40, anchor=W)
```

Create Heading

```
upcoming_booking_table.heading("#0", text="", anchor=W)
upcoming_booking_table.heading("Booking ID", text="Booking ID", anchor=W)
upcoming_booking_table.heading("Full Name", text="Full Name", anchor=W)
upcoming_booking_table.heading("Date and Time", text="Date and Time", anchor=W)
upcoming_booking_table.heading("Destination", text="Destination", anchor=W)
upcoming_booking_table.heading("Total Cost", text="Total Cost", anchor=W)
upcoming_booking_table.heading("Status", text="Status", anchor=W)
```

```
upcoming_trip_control = user_controller()
```

```
trip_fetched_data = upcoming_trip_control.upcoming_trip_detail_fetcher(userid)
```

```
for data in trip_fetched_data:
```

```
upcoming_booking_table.insert(
    "",
    index="end",
    values=(
        data[0],
        (data[2], data[3]),
        (data[6], "---", data[7], ":", data[8]),
        data[10],
        data[11],
        data[16],
    ),
)
```

```
upcoming_booking_table.bind(
    "<Double-1>",
    lambda event: ProfilePage.user_profile_booking_detail(
        event,
        upcoming_booking_table,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        user_controller,  
    ),  
)
```

```
@staticmethod  
def user_profile_booking_detail(  
    event,  
    upcoming_booking_table,  
    user_controller,  
):  
  
    selected = upcoming_booking_table.focus()  
    selected_row = upcoming_booking_table.item(  
        selected,  
        "values",  
    )  
    selected_booking_id = selected_row[0]
```

Booking Details Data Fetcher

```
booking_data_control = user_controller()  
fetched_booking_details = (  
    booking_data_control.user_profile_booking_data_fetcher(  
        selected_booking_id,  
    ),  
)
```

```
data = fetched_booking_details[0]
```

```
upcoming_booking_detail_frame = tk.Toplevel(upcoming_booking_table)  
upcoming_booking_detail_frame.title("Assign Taxi")  
upcoming_booking_detail_frame.geometry("682x641+193+31")  
upcoming_booking_detail_frame.resizable(0, 0)  
upcoming_booking_detail_frame.configure(background="#FFFFFF")
```

BUILDING BOOKING DETAIL PAGE

```
title = tk.Label(upcoming_booking_detail_frame)  
title.place(relx=0.360, rely=0.031, height=43, width=250)  
title.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 18 -weight bold",  
    text="Booking Details",  
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
fullname_label = tk.Label(upcoming_booking_detail_frame)
fullname_label.place(relx=0.029, rely=0.156, height=34, width=122)
fullname_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="Full Name :",
)

```

```
fullname_data = tk.Label(upcoming_booking_detail_frame)
fullname_data.place(relx=0.22, rely=0.156, height=34, width=280)
fullname_data.configure(
    anchor="w",
    background="#FFFFFF",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[2] + " " + data[3],
)

```

```
map_view_frame = tk.Frame(upcoming_booking_detail_frame)
map_view_frame.place(relx=0.640, rely=0.156, height=140, width=220)
map_view_frame.configure(
    background="#FFFFFF",
    borderwidth="2",
    highlightthickness="2",
)

```

```
location_detail_map_view = TkinterMapView(map_view_frame)
location_detail_map_view.set_tile_server(
    "https://mt0.google.com/vt/lyrs=m&hl=en&x={x}&y={y}&z={z}&s=Ga",
    max_zoom=60,
)

```

```
pickup_cordinates = eval(data[19])
destination_coordinates = eval(data[20])
```

```
location_detail_map_view.set_marker(
    pickup_cordinates[0],
    pickup_cordinates[1],
)

```

```
location_detail_map_view.set_marker(
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
    destination_coordinates[0],  
    destination_coordinates[1],  
)
```

```
location_detail_map_view.set_position(  
    pickup_cordinates[0],  
    pickup_cordinates[1],  
)
```

```
location_detail_map_view.set_path(  
    position_list=[pickup_cordinates, destination_coordinates]  
)
```

```
location_detail_map_view.pack()
```

```
passenger_no_label = tk.Label(upcoming_booking_detail_frame)  
passenger_no_label.place(relx=0.029, rely=0.234, height=34, width=175)  
passenger_no_label.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    text="No of Passenger :")
```

```
passenger_no_data = tk.Label(upcoming_booking_detail_frame)  
passenger_no_data.place(relx=0.293, rely=0.234, height=34, width=48)  
passenger_no_data.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    highlightthickness="2",  
    text=data[4],  
)
```

```
booking_status = tk.Label(upcoming_booking_detail_frame)  
booking_status.place(relx=0.029, rely=0.312, height=35, width=180)  
booking_status.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
    foreground="#4A4A4A",
    text="Booking Status :",
)
```

```
booking_status_data = tk.Label(upcoming_booking_detail_frame)
booking_status_data.place(relx=0.250, rely=0.312, height=35, width=150)
booking_status_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[16],
)
```

```
pick_up_date_lable = tk.Label(upcoming_booking_detail_frame)
pick_up_date_lable.place(relx=0.029, rely=0.39, height=34, width=133)
pick_up_date_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="Pick Up Date :",
)
```

```
pick_up_date_data = tk.Label(upcoming_booking_detail_frame)
pick_up_date_data.place(relx=0.249, rely=0.39, height=35, width=163)
pick_up_date_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[6],
)
```

```
time_label = tk.Label(upcoming_booking_detail_frame)
time_label.place(relx=0.513, rely=0.39, height=34, width=69)
time_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
compound="left",
font="-family {Noto Sans} -size 14",
foreground="#4A4A4A",
text="Time :",
)
```

```
data7 = str(data[7])
data8 = str(data[8])
time_data = tk.Label(upcoming_booking_detail_frame)
time_data.place(relx=0.616, rely=0.39, height=34, width=100)
time_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=(data7 + ":" + data8),
)
```

```
pick_up_address_label = tk.Label(upcoming_booking_detail_frame)
pick_up_address_label.place(relx=0.029, rely=0.468, height=35, width=175)
pick_up_address_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="Pick Up Address :",
)
```

```
pickup_address_data = tk.Label(upcoming_booking_detail_frame)
pickup_address_data.place(relx=0.293, rely=0.468, height=35, width=323)
pickup_address_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 10",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[9],
)
```

```
destination_label = tk.Label(upcoming_booking_detail_frame)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
destination_label.place(relx=0.029, rely=0.546, height=34, width=132)
```

```
destination_label.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    text="Destination :")
```

```
destination_address_lable = tk.Label(upcoming_booking_detail_frame)  
destination_address_lable.place(relx=0.249, rely=0.546, height=34, width=323)  
destination_address_lable.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 10",  
    foreground="#4A4A4A",  
    highlightthickness="2",  
    text=data[10],  
)
```

```
total_cost_label = tk.Label(upcoming_booking_detail_frame)  
total_cost_label.place(relx=0.029, rely=0.624, height=34, width=111)  
total_cost_label.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    text="Total Cost :")
```

```
total_cost_data = tk.Label(upcoming_booking_detail_frame)  
total_cost_data.place(relx=0.22, rely=0.624, height=34, width=206)  
total_cost_data.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    highlightthickness="2",  
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
        text=data[11] + ".NPR",
    )

payment_method_lable = tk.Label(upcoming_booking_detail_frame)
payment_method_lable.place(relx=0.029, rely=0.702, height=33, width=185)
payment_method_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""Payment Method :"""",
)
payment_method_data = tk.Label(upcoming_booking_detail_frame)
payment_method_data.place(relx=0.293, rely=0.702, height=34, width=206)
payment_method_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[12],
)

driver_details_lable = tk.Label(upcoming_booking_detail_frame)
driver_details_lable.place(relx=0.029, rely=0.780, height=33, width=185)
driver_details_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""Driver Details :"""",
)
driver_details_name = tk.Label(upcoming_booking_detail_frame)
driver_details_name.place(relx=0.250, rely=0.780, height=34, width=206)
driver_details_name.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    highlightthickness="2",
    text=data[23],
)
```

```
driver_details_number = tk.Label(upcoming_booking_detail_frame)
driver_details_number.place(relx=0.600, rely=0.780, height=34, width=206)
driver_details_number.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[25],
)
```

```
taxi_details_lable = tk.Label(upcoming_booking_detail_frame)
taxi_details_lable.place(relx=0.025, rely=0.858, height=33, width=185)
taxi_details_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="Taxi Details :",
)
```

```
data29 = str(data[31])
data30 = str(data[32])
```

```
if (data29) == "None":
    data29 = ""
if (data30) == "None":
    data30 = ""
```

```
taxi_details_name = tk.Label(upcoming_booking_detail_frame)
taxi_details_name.place(relx=0.220, rely=0.858, height=34, width=465)
taxi_details_name.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    text=data29 + " " + data30,  
    )
```

```
taxi_details_number = tk.Label(upcoming_booking_detail_frame)  
taxi_details_number.place(relx=0.220, rely=0.936, height=34, width=206)  
taxi_details_number.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    highlightthickness="2",  
    text=data[33],  
)
```

```
back_button = tk.Button(upcoming_booking_detail_frame)  
back_button.place(relx=0.865, rely=0.031, height=33, width=71)  
back_button.configure(  
    activebackground="beige",  
    borderwidth="2",  
    compound="left",  
    font="-family {Noto Sans} -size 10",  
    foreground="#FFFFFF",  
    text="Back",  
)
```

```
cancel_booking_button = tk.Button(upcoming_booking_detail_frame)  
cancel_booking_button.place(relx=0.580, rely=0.936, height=34, width=130)  
cancel_booking_button.configure(  
    activebackground="beige",  
    borderwidth="2",  
    compound="left",  
    font="-family {Noto Sans} -size 10 -weight bold",  
    text="Cancel Booking",  
    command=lambda: ProfilePage.user_cancel_booking(  
        user_controller, upcoming_booking_detail_frame, data  
)  
)
```

```
if data[16] == "Accepted":  
    cancel_booking_button.config(state="disabled")
```

```
upcoming_booking_detail_frame.wait_visibility()  
upcoming_booking_detail_frame.grab_set()  
upcoming_booking_detail_frame.mainloop()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
@staticmethod
def user_cancel_booking(
    user_controller,
    upcoming_booking_detail_frame,
    data,
):
    booking_id = data[0]
    response = messagebox.askquestion(
        "Cancel Booking", "DO you really want to cancel this booking ?"
    )
    if response == "yes":
        user_booking_cancel_control = user_controller()
        user_booking_cancel_control.cancel_booking(booking_id)
        upcoming_booking_detail_frame.destroy()
```

Package : user_model.py

code :

```
import re
from pydantic import BaseModel, root_validator, validator
from helper.exceptions import CustomException

name_reg = re.compile("^[A-Z][a-z]+$") # firstname, lastname regex
contact_reg = re.compile("^[0-9]{10}$") # contact regex
address_reg = re.compile(".+") # address not null regex
email_reg = re.compile("[a-z0-9]+@[a-z]+\.[a-z]{2,3}") # email address regex
username_reg = re.compile("[a-z][a-z0-9]*([._-][a-z0-9]+){0,3}$") # username regex
```

```
class UserModel(BaseModel):
```

```
    firstname: str = ""
    lastname: str = ""
    contact: str = ""
    address: str = ""
    email: str = ""
    username: str = ""
    password: str = ""
    confirm_password: str = ""
    user_id: str = ""
    message: str = ""
    profile: str = ""
    new_password: str = ""
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
@validator("firstname", "lastname")
def validate_first_name(cls, value):
    if not value:
        raise CustomException("name cannot be empty")
    if not name_reg.match(value):
        raise CustomException("invalid format for name")
    return value
```

```
@validator("contact")
def validate_contact(cls, value):
    if not value:
        raise CustomException("contact cannot be empty")
    if not contact_reg.match(value):
        raise CustomException("invalid format for contact")
    return value
```

```
@validator("address")
def validate_address(cls, value):
    if not value:
        raise CustomException("address cannot be empty")
    if not address_reg.match(value):
        raise CustomException("invalid format for address")
    return value
```

```
@validator("email")
def validate_email(cls, value):
    if not value:
        raise CustomException("email cannot be empty")
    if not email_reg.match(value):
        raise CustomException("invalid format for email")
    return value
```

```
@validator("username")
def validate_username(cls, value):
    if not value:
        raise CustomException("username cannot be empty")
    if not username_reg.match(value):
        raise CustomException("invalid format for username")
    return value
```

```
@root_validator
def validate_password(cls, value):
    password = value.get("password")
    confirm_password = value.get("confirm_password")
    if not password:
        raise CustomException("password cannot be empty")
    if password != confirm_password:
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    raise CustomException("The two passwords did not match.")  
    return value
```

Package : user_controllers.py

code :

```
from curses import use_default_colors  
from tkinter import messagebox  
from helper.turbo_db import Turbo_db  
import psycopg2  
from .dashboard_page import DashboardPage
```

```
class UserController:  
    def __init__(self) -> None:  
        self._connection = Turbo_db.turbo_connection()  
  
    def registration_control(self, user, signup_frame):  
        if self.authenticate(user):  
            self.registration_sucess(signup_frame)  
  
    def authenticate(self, user):  
        try:  
            cursor = self._connection.cursor()  
            statement = """CREATE TABLE IF NOT EXISTS users(  
                userID SERIAL PRIMARY KEY,  
                firstname VARCHAR(50) NOT NULL,  
                lastname VARCHAR(50) NOT NULL,  
                contact VARCHAR(20) NOT NULL,  
                address VARCHAR(50) NOT NULL,  
                email VARCHAR(50) NOT NULL UNIQUE,  
                username VARCHAR(20) NOT NULL UNIQUE,  
                password VARCHAR(20) NOT NULL,  
                profile BYTEA  
            );"""  
        default_image = open(  
            "/home/vivu/Class Stuff V2/ISD_Files/TurboTB_System/img/profile.png",  
            "rb",  
        ).read()
```

```
    data_insert = """INSERT INTO users(firstname, lastname, contact,  
        address, email, username, password, profile) VALUES (%s, %s, %s, %s, %s, %s, %s);  
    """  
    data_values = (
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        user.firstname,
        user.lastname,
        user.contact,
        user.address,
        user.email,
        user.username,
        user.password,
        psycopg2.Binary(defult_image),
    )
    cursor.execute(statement)
    cursor.execute(data_insert, data_values)
    self._connection.commit()
    return True
except Exception as error:
    if "username" in str(error):
        messagebox.showerror(
            "Invalid Data",
            "username already exist!!",
        )
    elif "email" in str(error):
        messagebox.showerror(
            "Invalid Data",
            "Already have account linked \
            to this account.",
        )
    else:
        print(error)
finally:
    if cursor is not None:
        cursor.close()
    if self._connection is not None:
        self._connection.close()
```

```
def registration_sucess(self, signup_frame):
    messagebox.showinfo(
        "Account Created",
        "Congatulation, Your Account Has Been Created",
    )
    signup_frame.destroy()
```

```
def login_control(self, user, root):
    if self.login_authenticate(user):
        self.login_sucess(root)
    else:
        messagebox.showerror("Invalid Data", "username or password not matched")
```

```
def login_authenticate(self, user):
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
cursor = self._connection.cursor()
statement = "SELECT * FROM users WHERE username=%s AND password = %s;"
data = (user.username, user.password)
cursor.execute(statement, data)
self.record = cursor.fetchall()
if self.record:
    for data in self.record:
        self.fetch_user_id = data[0]
        self.fetched_firstname = data[1].rstrip()
        self.fetched_lastname = data[2].rstrip()
        self.fetch_username = data[6]
        self.fetch_password = data[7].rstrip()
        self.fetched_profile = data[8]
    return True
return False
```

```
def login_sucess(self, root):
    DashboardPage(
        root,
        self.record,
        UserController,
    ).
```

```
def delete_account(
    self,
    profile_frame,
    dashboard_frame,
    edit_bio_frame,
    record,
):
    for data in record:
        fetched_username = data[6]
        try:
            cursor = self._connection.cursor()
            statement = """DELETE FROM users WHERE username = %s;"""
            data_values = (fetched_username,)
            cursor.execute(statement, data_values)
            self._connection.commit()
```

```
messagebox.showinfo(
    "Sucess!!",
    "Account Sucessfully deleted !",
)
```

```
edit_bio_frame.destroy()
profile_frame.destroy()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    dashboard_frame.destroy()
```

```
except Exception as e:  
    print(e)  
finally:  
    if cursor is not None:  
        cursor.close()  
    if self._connection is not None:  
        self._connection.close()
```

```
def change_profile(  
    self,  
    user,  
    record,  
    profile_frame,  
    dashboard_frame,  
):
```

```
for data in record:  
    fetched_firstname = data[6]  
try:  
    image_path = user.profile  
    new_profile = open(image_path, "rb").read()  
    cursor = self._connection.cursor()  
    statement = """UPDATE users  
SET profile = %s WHERE username = %s;"""  
    dataValues = (  
        psycopg2.Binary(new_profile),  
        fetched_firstname,  
    )
```

```
    cursor.execute(statement, dataValues)  
    self._connection.commit()
```

```
    messagebox.showinfo(  
        "Sucess!",  
        "Profile Picture Changed !",  
    )
```

```
# profile_frame.destroy()  
# dashboard_frame.destroy()
```

```
except Exception as e:  
    print(e)  
finally:  
    if cursor is not None:  
        cursor.close()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
if self._connection is not None:  
    self._connection.close()
```

```
def update_user_bio(  
    self,  
    user,  
    record,  
    edit_bio_frame,  
    profile_frame,  
    dashboard_frame,  
):  
    for data in record:  
        fetched_username = data[6]  
    try:  
        cursor = self._connection.cursor()  
        statement = """UPDATE users  
        SET contact = COALESCE(NULLIF(%s,""),contact),  
        address = COALESCE(NULLIF(%s,""),address),  
        username = COALESCE(NULLIF(%s,""),username),  
        password = COALESCE(NULLIF(%s,""),password)  
        WHERE username = %s"""
```

```
    dataValues = (  
        user.contact,  
        user.address,  
        user.username,  
        user.new_password,  
        fetched_username,  
    )
```

```
    cursor.execute(statement, dataValues)  
    self._connection.commit()
```

```
    messagebox.showinfo(  
        "Sucess!!",  
        "Updated !",  
    )
```

```
    edit_bio_frame.destroy()
```

```
except Exception as error:  
    if "username" in str(error):  
        messagebox.showerror(  
            "Invalid Data",  
            "username already exist!!",  
            parent=edit_bio_frame,  
        )
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        else:  
            print(error)  
    finally:  
        if cursor is not None:  
            cursor.close()  
        if self._connection is not None:  
            self._connection.close()
```

```
def upcoming_trip_detail_fetcher(self, user_id):
```

```
    try:  
        cursor = self._connection.cursor()  
        statement = "SELECT * FROM booking as b LEFT JOIN drivers as d on b.driver_id = d.driverid  
LEFT JOIN taxi as t on d.taxi_number = t.taxi_number WHERE b.booking_status = 'Pending' AND  
user_id = %s OR b.booking_status = 'Accepted' AND user_id = %s ;"  
        uid = str(user_id)  
        data = (uid, uid)  
        cursor.execute(statement, data)  
        self.record = cursor.fetchall()  
        return self.record  
    except Exception as error:  
        print(error)
```

```
def user_profile_booking_data_fetcher(self, selected_booking_id):
```

```
    try:  
        cursor = self._connection.cursor()  
        statement = "SELECT * FROM booking as b LEFT JOIN drivers as d on b.driver_id = d.driverid  
LEFT JOIN taxi as t on d.taxi_number = t.taxi_number WHERE b.booking_id = %s;"  
        bid = str(selected_booking_id)  
        data = bid  
        cursor.execute(statement, data)  
        self.record = cursor.fetchall()  
        return self.record  
    except Exception as error:  
        print(error)
```

```
def cancel_booking(self, booking_id):
```

```
    try:  
        statement = "UPDATE booking SET booking_status = 'Canceled' WHERE booking_id  
= %s AND booking_status = 'Pending';"  
  
        b_id = str(booking_id)  
        cursor = self._connection.cursor()  
        data = b_id  
        cursor.execute(statement, data)  
        self._connection.commit()  
  
    except Exception as error:
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    print(error)
```

```
def history_trip_detail_fetcher(self, user_id):  
  
    try:  
        cursor = self._connection.cursor()  
        statement = "SELECT * FROM booking as b LEFT JOIN drivers as d on b.driver_id = d.driverid  
LEFT JOIN taxi as t on d.taxi_number = t.taxi_number WHERE b.booking_status = 'Completed' AND  
user_id = %s OR b.booking_status = 'Canceled' AND user_id = %s ;"  
        uid = str(user_id)  
        data = (uid, uid)  
        cursor.execute(statement, data)  
        self.record = cursor.fetchall()  
        return self.record  
    except Exception as error:  
        print(error)
```

- **Module : bookings**

Package : booking_form.py

code :

```
from datetime import date, timedelta  
from helper.exceptions import CustomException  
from .booking_model import BookingModel  
from .booking_controllers import BookingController  
import math  
from tkcalendar import DateEntry  
from tkinter import StringVar, messagebox  
from tkintermapview import TkinterMapView, convert_coordinates_to_address  
import haversine as hs  
  
import tkinter as tk
```

```
class BookingPage:  
    def __init__(self, root, dashboard_frame, record) -> None:  
        self.root = root  
        self.dashboard_frame = dashboard_frame  
        self.record = record  
        self.create_booking_form_frame()
```

```
    def create_booking_form_frame(self):  
        self.booking_form_frame = tk.Frame(self.root)  
        self.booking_form_frame.configure(background="#FFFFFF")  
        self.booking_form_frame.place(  
            relx=-0.010,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    rely=0,
    height=768,
    width=1366,
)
self.build_booking_form_frame(
    dashboard_frame=self.dashboard_frame,
    record=self.record,
    booking_form_frame=self.booking_form_frame,
)
```

```
pickup_location_coordinates = None
destination_coordinates = None
button_name = None
```

```
@staticmethod
def build_booking_form_frame(
    dashboard_frame,
    record,
    booking_form_frame,
):
    title_label = tk.Label(booking_form_frame)
    title_label.place(relx=0.052, rely=0.042, height=59, width=143)
    title_label.configure(
        activebackground="#f9f9f9",
        anchor="w",
        background="#FFFFFF",
        compound="left",
        font="-family {Noto Sans} -size 18 -weight bold",
        text="Book Now",
    )
```

```
message_label = tk.Message(booking_form_frame)
message_label.place(relx=0.052, rely=0.122, relheight=0.067, relwidth=0.671)
message_label.configure(
    anchor="w",
    background="#FFFFFF",
    foreground="#637381",
    padx="1",
    pady="1",
    text="some information",
    width=907,
)
```

```
for data in record:
    fetched_firstname = data[1].rstrip()
    fetched_lastname = data[2].rstrip()
    fetched_user_id = data[0]
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
fullname = tk.Label(booking_form_frame)
fullname.place(relx=0.044, rely=0.312, height=31, width=93)
fullname.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    text="Full Name :",
)
```

```
firstname_entry = tk.Entry(booking_form_frame)
firstname_entry.place(relx=0.126, rely=0.312, height=33, relwidth=0.173)
firstname_entry.configure(
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"
)
firstname_entry.insert(0, fetched_firstname)
```

```
lastname_entry = tk.Entry(booking_form_frame)
lastname_entry.place(relx=0.318, rely=0.312, height=33, relwidth=0.173)
lastname_entry.configure(
    background="#EFF0F2", selectbackground="#c4c4c4", font="TkFixedFont"
)
lastname_entry.insert(0, fetched_lastname)
```

```
firstname = tk.Label(booking_form_frame)
firstname.place(relx=0.126, rely=0.366, height=20, width=93)
firstname.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    foreground="#637381",
    text="First Name",
)
```

```
lastname = tk.Label(booking_form_frame)
lastname.place(relx=0.318, rely=0.366, height=20, width=93)
lastname.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    foreground="#637381",
    text="Last Name",
)
```

```
no_of_passenger = tk.Label(booking_form_frame)
no_of_passenger.place(relx=0.044, rely=0.421, height=31, width=205)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
no_of_passenger.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 12",  
    text="Number Of Passenger :",  
)
```

```
no_of_passenger_entry = tk.Spinbox(  
    booking_form_frame, from_=1.0, to=4.0, state="readonly")  
no_of_passenger_entry.place(  
    relx=0.192, rely=0.421, relheight=0.038, relwidth=0.079)  
no_of_passenger_entry.configure(  
    background="#EFF0F2",  
    font="TkDefaultFont",  
    highlightbackground="black",  
    selectbackground="#c4c4c4",  
)
```

```
no_of_taxi_label = tk.Label(booking_form_frame)  
no_of_taxi_label.place(relx=0.044, rely=0.488, height=31, width=175)  
no_of_taxi_label.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 12",  
    text="Number Of Taxi :",  
)
```

```
no_of_taxi_entry = tk.Spinbox(  
    booking_form_frame, from_=1.0, to=2.0, state="disabled")  
no_of_taxi_entry.place(relx=0.192, rely=0.488, relheight=0.038, relwidth=0.079)  
no_of_taxi_entry.configure(  
    activebackground="#f9f9f9",  
    background="#EFF0F2",  
    font="TkDefaultFont",  
    highlightbackground="black",  
    selectbackground="#c4c4c4",  
)
```

```
taxi_message = tk.Message(booking_form_frame)  
taxi_message.place(relx=0.281, rely=0.501, relheight=0.028, relwidth=0.152)  
taxi_message.configure(  
    anchor="w",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
        background="#FFFFFF",
        foreground="#637381",
        padx="1",
        pady="1",
        text="""a taxi can carry only upto 4 people""",
        width=205,
    )
```

```
pickup_date = tk.Label(booking_form_frame)
pickup_date.place(relx=0.044, rely=0.556, height=30, width=134)
pickup_date.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    text="""Pick Up Date :""",
)
```

```
today_date = date.today()
last_date = date.today() + timedelta(days=5)
```

```
date_entry = DateEntry(
    booking_form_frame,
    selectmode="day",
    mindate=today_date,
    maxdate=last_date,
    state="readonly",
    date_pattern="yyyy/mm/dd",
)
date_entry.place(relx=0.185, rely=0.556, height=33, relwidth=0.092)
```

```
date_label = tk.Label(booking_form_frame)
date_label.place(relx=0.185, rely=0.612, height=20, width=93)
date_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    foreground="#637381",
    text="""MM/DD/YY""",
)
```

```
hour_entry = tk.Spinbox(booking_form_frame, from_=0.0, to=23.0)
hour_entry.place(relx=0.285, rely=0.556, relheight=0.038, relwidth=0.079)
hour_entry.configure(
    activebackground="#f9f9f9",
    background="#EFF0F2",
    font="TkDefaultFont",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    highlightbackground="black",
    selectbackground="#c4c4c4",
)
```

```
minute_entry = tk.Spinbox(booking_form_frame, from_=0.0, to=55.0, increment=5.0)
minute_entry.place(relx=0.370, rely=0.556, relheight=0.038, relwidth=0.079)
minute_entry.configure(
    activebackground="#f9f9f9",
    background="#EFF0F2",
    font="TkDefaultFont",
    highlightbackground="black",
    selectbackground="#c4c4c4",
)
```

```
time_label = tk.Label(booking_form_frame)
time_label.place(relx=0.320, rely=0.612, height=20, width=93)
time_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    foreground="#637381",
    text="HH(24hrs) : MM",
)
```

```
pickup_address = tk.Label(booking_form_frame)
pickup_address.place(relx=0.044, rely=0.651, height=31, width=134)
pickup_address.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    text="Pick Up Address :",
)
```

```
pickup_address_entry = tk.Entry(booking_form_frame, state="readonly")
pickup_address_entry.place(relx=0.185, rely=0.651, height=33, relwidth=0.265)
pickup_address_entry.configure(
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"
)
```

```
pickup_address_button = tk.Button(booking_form_frame)
pickup_address_button.place(
    relx=0.47,
    rely=0.651,
    height=33,
    relwidth=0.122,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
pickup_address_button.configure(  
    activebackground="beige",  
    borderwidth="1",  
    compound="left",  
    font="-family {Noto Sans} -size 12",  
    text="Select Address",  
)
```

```
pickup_address_button.bind(  
    "<ButtonRelease-1>",  
    lambda event: BookingPage.create_location_picker(  
        event,  
        booking_form_frame,  
        pickup_address_entry,  
    ),  
)
```

```
pickup_address_button.bind("<Button-1>", BookingPage.button_name_finder)
```

```
destination = tk.Label(booking_form_frame)  
destination.place(relx=0.044, rely=0.746, height=31, width=134)  
destination.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 12",  
    text="Destination :")
```

```
destination_entry = tk.Entry(booking_form_frame, state="readonly")  
destination_entry.place(relx=0.185, rely=0.746, height=33, relwidth=0.265)  
destination_entry.configure(  
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4")
```

```
destination_button = tk.Button(booking_form_frame)  
destination_button.place(relx=0.47, rely=0.746, height=33, relwidth=0.122)  
destination_button.configure(  
    activebackground="beige",  
    borderwidth="1",  
    compound="left",  
    font="-family {Noto Sans} -size 12",  
    text="Select Address",  
)  
destination_button.bind(  
    "<ButtonRelease-1>",  
    lambda event: BookingPage.create_location_picker()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
        event,
        booking_form_frame,
        destination_entry,
    ),
)
destination_button.bind("<Button-1>", BookingPage.button_name_finder_2)
```

```
total_cost = tk.Label(booking_form_frame)
total_cost.place(relx=0.044, rely=0.855, height=40, width=144)
total_cost.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    text="""Total Charge :""",
)
```

```
cost = tk.Label(booking_form_frame)
cost.place(relx=0.155, rely=0.855, height=40, width=143)
cost.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    text="""XXXXXX""",
)
```

```
view_total_cost_button = tk.Button(booking_form_frame)
view_total_cost_button.place(
    relx=0.260,
    rely=0.855,
    height=33,
    relwidth=0.122,
)
view_total_cost_button.configure(
    activebackground="beige",
    borderwidth="1",
    compound="left",
    font="-family {Noto Sans} -size 12",
    text="""View Cost""",
)
```

```
view_total_cost_button.bind(
    "<Button-1>",
    lambda event: BookingPage.show_cost(
        event,
        no_of_taxi_entry,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
    cost,  
    ),  
)  
)
```

```
view_total_cost_button.bind(  
    "<ButtonRelease-1>",  
    lambda event: BookingPage.hide_cost(  
        event,  
        cost,  
    ),  
)
```

```
book_now_button = tk.Button(booking_form_frame)  
book_now_button.place(relx=0.820, rely=0.840, height=53, width=141)  
book_now_button.configure(  
    background="#007074",  
    borderwidth="2",  
    compound="left",  
    font="-family {Noto Sans} -size 16",  
    foreground="#FFFFFF",  
    text="Book Now",  
    command=lambda: BookingPage.user_booking_register(  
        booking_form_frame,  
        record,  
        firstname_entry,  
        lastname_entry,  
        no_of_passenger_entry,  
        no_of_taxi_entry,  
        date_entry,  
        hour_entry,  
        minute_entry,  
        pickup_address_entry,  
        destination_entry,  
        rb_valv,  
        card_number_entry,  
        exp_entry,  
        cvv_entry,  
        cost,  
    ),  
)
```

```
payment_method = tk.LabelFrame(booking_form_frame)  
payment_method.place(relx=0.570, rely=0.285, relheight=0.21, relwidth=0.34)  
payment_method.configure(  
    relief="groove", background="#FFFFFF", text="Payment Method")  
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
card_number_entry = tk.Entry(payment_method)
card_number_entry.place(
    relx=0.047, rely=0.583, height=33, relwidth=0.508, bordermode="ignore"
)
card_number_entry.configure(
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"
)
```

```
exp_entry = tk.Entry(payment_method)
exp_entry.place(
    relx=0.611, rely=0.583, height=33, relwidth=0.142, bordermode="ignore"
)
exp_entry.configure(
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"
)
```

```
cvv_entry = tk.Entry(payment_method)
cvv_entry.place(
    relx=0.806, rely=0.583, height=33, relwidth=0.142, bordermode="ignore"
)
cvv_entry.configure(
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"
)
```

```
card_number = tk.Label(payment_method)
card_number.place(
    relx=0.043, rely=0.84, height=20, width=93, bordermode="ignore"
)
card_number.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    foreground="#637381",
    text="Card Number",
)
```

```
exp_date = tk.Label(payment_method)
exp_date.place(relx=0.611, rely=0.84, height=20, width=62, bordermode="ignore")
exp_date.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    foreground="#637381",
    text="Exp Date",
)
```

```
cvv = tk.Label(payment_method)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
cvv.place(relx=0.806, rely=0.84, height=20, width=63, bordermode="ignore")
cvv.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    foreground="#637381",
    text="""CVV""",  
)
```

```
rb_valv = StringVar()
```

```
credit_card = tk.Radiobutton(
    payment_method,
    value="Credit Card",
    variable=rb_valv,
    command=lambda: BookingPage.enable_card_details(
        card_number_entry,
        exp_entry,
        cvv_entry,  
),
)
credit_card.place(relx=0.043, rely=0.256, relheight=0.212, relwidth=0.267)
credit_card.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 10",
    justify="left",
    selectcolor="#d9d9d9",
    text="""Credit Card""",  
)
```

```
cashInHand = tk.Radiobutton(
    payment_method,
    value="Cash In Hand",
    variable=rb_valv,  

)
cashInHand.place(relx=0.350, rely=0.256, relheight=0.212, relwidth=0.267)
cashInHand.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 10",
    justify="left",
    selectcolor="#d9d9d9",
    text="""Cash In Hand""",  
command=lambda: BookingPage.disable_card_details(  
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
    card_number_entry,
    exp_entry,
    cvv_entry,
),
)
rb_valv.set("Credit Card")
```

```
back = tk.Button(booking_form_frame)
back.place(x=1165, y=60, height=35, width=90)
back.config(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#FFFFFF",
    text="Back",
    command=lambda: BookingPage.redirect_to_dashboard(
        booking_form_frame,
    ),
)
```

```
booking_form_frame.mainloop()
```

```
@staticmethod
def redirect_to_dashboard(booking_form_frame):
    booking_form_frame.destroy()
```

```
@staticmethod
def disable_card_details(
    card_number_entry,
    exp_entry,
    cvv_entry,
):
    card_number_entry.delete(0, "end")
    card_number_entry.config(state="disabled")
    exp_entry.delete(0, "end")
    exp_entry.config(state="disabled")
    cvv_entry.delete(0, "end")
    cvv_entry.config(state="disabled")
```

```
    card_number_entry.update()
    exp_entry.update()
    cvv_entry.update()
```

```
@staticmethod
def enable_card_details(card_number_entry, exp_entry, cvv_entry):
    card_number_entry.config(state="normal")
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
exp_entry.config(state="normal")
cvv_entry.config(state="normal")
card_number_entry.update()
exp_entry.update()
cvv_entry.update()
```

```
@staticmethod
def create_location_picker(
    event,
    booking_form_frame,
    location_entry,
):
    location_picker_frame = tk.Toplevel(booking_form_frame)
    location_picker_frame.title("Location Picker")
    location_picker_frame.resizable(0, 0)
    location_picker_frame.configure(background="#FFFFFF")
    location_picker_frame.geometry("580x580")
    BookingPage.build_location_picker_frame(
        location_picker_frame,
        location_entry,
    )
```

```
@staticmethod
def build_location_picker_frame(
    location_picker_frame,
    location_entry,
):
    google_map_view = TkinterMapView(
        location_picker_frame,
        width=580,
        height=480,
    )
    google_map_view.set_tile_server(
        "https://mt0.google.com/vt/lyrs=m&hl=en&x={x}&y={y}&z={z}&s=Ga",
        max_zoom=22,
    )
    google_map_view.place(relx=0, rely=0.07)
    google_map_view.set_address("kathmandu, nepal")
    google_map_view.set_zoom(13)
```

```
selected_address = tk.Label(location_picker_frame)
selected_address.place(relx=0.0, rely=0.010, height=33, relwidth=0.999)
selected_address.config(background="#FFFFFF")
```

```
google_map_view.add_left_click_map_command(
    lambda coordinates: BookingPage.location_marker(
        coordinates,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    selected_address,  
    ),  
)
```

```
search_entry = tk.Entry(location_picker_frame)  
search_entry.place(relx=0.060, rely=0.920, height=33, relwidth=0.300)  
search_entry.configure(  
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"  
)
```

```
search_button = tk.Button(location_picker_frame)  
search_button.place(relx=0.400, rely=0.920, height=33, relwidth=0.122)  
search_button.configure(  
    activebackground="beige",  
    borderwidth="1",  
    compound="left",  
    font="-family {Noto Sans} -size 12",  
    text="Search",  
    command=lambda: BookingPage.search_location(  
        google_map_view,  
        search_entry,  
    ),  
)
```

```
select_location_button = tk.Button(location_picker_frame)  
select_location_button.place(  
    relx=0.550,  
    rely=0.920,  
    height=33,  
    relwidth=0.290,  
)  
select_location_button.configure(  
    activebackground="beige",  
    borderwidth="1",  
    compound="left",  
    font="-family {Noto Sans} -size 12",  
    text="Select Location",  
    command=lambda: BookingPage.address_selected_display(  
        location_entry,  
        location_picker_frame,  
    ),  
)  
location_picker_frame.grab_set()  
location_picker_frame.mainloop()
```

```
@staticmethod  
def search_location(google_map_view, search_entry):
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
google_map_view.set_address(search_entry.get())
google_map_view.set_zoom(15)
```

```
clicked_address = None
```

```
@staticmethod
def location_marker(
    coordinates,
    selected_address,
):
    if BookingPage.button_name == "pickup_button":
        BookingPage.pickup_location_coordinates = coordinates
    else:
        BookingPage.destination_coordinates = coordinates
```

```
clicked_coordinates_1 = coordinates[0]
clicked_coordinates_2 = coordinates[1]
BookingPage.clicked_address = convert_coordinates_to_address(
    clicked_coordinates_1,
    clicked_coordinates_2,
)
selected_address.config(
    text=(
        BookingPage.clicked_address.street,
        BookingPage.clicked_address.postal,
        BookingPage.clicked_address.city,
        BookingPage.clicked_address.country,
    ),
    font="-family {Noto Sans} -size 14",
)
```

```
@staticmethod
def address_selected_display(
    location_entry,
    location_picker_frame,
):
    location_entry.config(state="normal")
    location_entry.delete(0, "end")
    location_entry.insert(
        0,
        (
            BookingPage.clicked_address.street,
            BookingPage.clicked_address.postal,
            BookingPage.clicked_address.city,
            BookingPage.clicked_address.country,
        ),
    )
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
location_entry.config(state="readonly")
location_picker_frame.destroy()
```

```
final_cost = ""
```

```
@staticmethod
def calculate_total_cost(no_of_taxi_entry, cost):
```

```
try:
    total_cost = (
        math.ceil(
            hs.haversine(
                BookingPage.pickup_location_coordinates,
                BookingPage.destination_coordinates,
            )
        ) * 59.25
    )
no_of_taxi = int(no_of_taxi_entry.get())
```

```
showCost = total_cost * no_of_taxi
```

```
BookingPage.final_cost = str(round(showCost, 2))
cost.config(text="NPR. " + BookingPage.final_cost)
except Exception as error:
    cost.config(text="XXXXXX")
```

```
@staticmethod
def button_name_finder(event):
    BookingPage.button_name = "pickup_button"
```

```
@staticmethod
def button_name_finder_2(event):
    BookingPage.button_name = "destination_button"
```

```
@staticmethod
def show_cost(
    event,
    no_of_taxi_entry,
    cost,
):
    BookingPage.calculate_total_cost(no_of_taxi_entry, cost)
```

```
@staticmethod
def hide_cost(
    event,
    cost,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
):  
    cost.config(text="XXXXXX")
```

```
@staticmethod  
def user_booking_register(  
    booking_form_frame,  
    record,  
    firstname_entry,  
    lastname_entry,  
    no_of_passenger_entry,  
    no_of_taxi_entry,  
    date_entry,  
    hour_entry,  
    minute_entry,  
    pickup_address_entry,  
    destination_entry,  
    rb_valv,  
    card_number_entry,  
    exp_entry,  
    cvv_entry,  
    cost,  
):  
    for data in record:  
        fetched_user_id = data[0]
```

```
BookingPage.calculate_total_cost(no_of_taxi_entry, cost)  
try:  
    user_booking = BookingModel(  
        user_id=fetched_user_id,  
        firstname=firstname_entry.get(),  
        lastname=lastname_entry.get(),  
        no_of_pass=no_of_passenger_entry.get(),  
        no_of_taxi=no_of_taxi_entry.get(),  
        pick_up_date=date_entry.get(),  
        pick_up_hrs=hour_entry.get(),  
        pick_up_min=minute_entry.get(),  
        pick_up_location=pickup_address_entry.get(),  
        destination=destination_entry.get(),  
        payment_method=rb_valv.get(),  
        card_number=card_number_entry.get(),  
        exp_date=exp_entry.get(),  
        cvv=cvv_entry.get(),  
        total_cost=BookingPage.final_cost,  
        pickup_coordinates=str(BookingPage.pickup_location_coordinates),  
        destination_coordinates=str(BookingPage.destination_coordinates),  
)  
    booking_control = BookingController()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    booking_control.booking_control(user_booking, booking_form_frame)
```

```
except CustomException as e:  
    messagebox.showerror("Invalid Data", e)
```

Package : booking_model.py

code :

```
from ast import Return  
from datetime import datetime  
import re  
from pydantic import BaseModel, ValidationError, root_validator, validator  
from helper.exceptions import CustomException
```

```
name_reg = re.compile("^[A-Z][a-z]+$") # firstname, lastname regex
```

```
class BookingModel(BaseModel):
```

```
    booking_id: str = ""  
    user_id: str = ""  
    firstname: str = ""  
    lastname: str = ""  
    no_of_pass: str = ""  
    no_of_taxi: str = ""  
    pick_up_date: str = ""  
    pick_up_hrs: int = 0  
    pick_up_min: int = 0  
    pick_up_location: str = ""  
    destination: str = ""  
    total_cost: str = ""  
    payment_method: str = ""  
    card_number: str = ""  
    exp_date: str = ""  
    cvv: str = ""  
    pickup_coordinates: str = ""  
    destination_coordinates: str = ""
```

```
@validator("firstname", "lastname")  
def validate_name(cls, value):  
    if not value:  
        raise CustomException("name cannot be empty")  
    if not name_reg.match(value):  
        raise CustomException("invalid format for name")  
    return value
```

```
@root_validator  
def validate_time(cls, value):  
    pick_up_hrs = value.get("pick_up_hrs")
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
pick_up_min = value.get("pick_up_min")
pick_up_date = value.get("pick_up_date")
```

```
if not pick_up_hrs:
    raise CustomException("time(Hour) cannot be empty")
if not pick_up_min:
    raise CustomException("time(Minutes) cannot be empty")
```

```
if pick_up_hrs == str:
    raise CustomException("time(Hour) invalid")
if pick_up_min == str:
    raise CustomException("time(Minutes) invalid")
```

```
if pick_up_hrs >= 24:
    raise CustomException("time(Hour) invalid")
if pick_up_min >= 60:
    raise CustomException("time(Minutes) invalid")
else:
    trip_date = datetime.strptime(pick_up_date, "%Y/%m/%d").date()
    trip_time = str(pick_up_hrs) + ":" + str(pick_up_min) + ":00"
    time_final = datetime.strptime(trip_time, "%H:%M:%S").time()
```

```
trip_date_and_time = datetime.combine(trip_date, time_final)
today = datetime.now().isoformat(" ", "seconds")
today_date = datetime.strptime(today, "%Y-%m-%d %H:%M:%S")
```

```
if today_date > trip_date_and_time:
    raise CustomException("invalid time given")
else:
    return value
```

```
@root_validator
def validate_location(cls, value):
    pick_up_location = value.get("pick_up_location")
    destination = value.get("destination")
    if not pick_up_location:
        raise CustomException("pickup location cannot be empty")
    if not destination:
        raise CustomException("destination location cannot be empty")
    return value
```

```
@root_validator
def validate_card_details(cls, value):
    payment_method = value.get("payment_method")
    credit_number = value.get("card_number")
    exp_date = value.get("exp_date")
    cvv = value.get("cvv")
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
if payment_method == "Credit Card":  
    if credit_number == "":  
        raise CustomException("credit number cannot be empty!!")  
    if exp_date == "":  
        raise CustomException("exp date cannot be empty!!")  
    if cvv == "":  
        raise CustomException("cvv cannot be empty!!")  
    return value  
else:  
    return value  
from ast import Return  
from datetime import datetime  
import re  
from pydantic import BaseModel, ValidationError, root_validator, validator  
from helper.exceptions import CustomException
```

```
name_reg = re.compile("^[A-Z][a-z]+$") # firstname, lastname regex
```

```
class BookingModel(BaseModel):
```

```
    booking_id: str = ""  
    user_id: str = ""  
    firstname: str = ""  
    lastname: str = ""  
    no_of_pass: str = ""  
    no_of_taxi: str = ""  
    pick_up_date: str = ""  
    pick_up_hrs: int = 0  
    pick_up_min: int = 0  
    pick_up_location: str = ""  
    destination: str = ""  
    total_cost: str = ""  
    payment_method: str = ""  
    card_number: str = ""  
    exp_date: str = ""  
    cvv: str = ""  
    pickup_coordinates: str = ""  
    destination_coordinates: str = ""
```

```
@validator("firstname", "lastname")  
def validate_name(cls, value):  
    if not value:  
        raise CustomException("name cannot be empty")  
    if not name_reg.match(value):  
        raise CustomException("invalid format for name")  
    return value
```

```
@root_validator
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
def validate_time(cls, value):
    pick_up_hrs = value.get("pick_up_hrs")
    pick_up_min = value.get("pick_up_min")
    pick_up_date = value.get("pick_up_date")

    if not pick_up_hrs:
        raise CustomException("time(Hour) cannot be empty")
    if not pick_up_min:
        raise CustomException("time(Minutes) cannot be empty")

    if pick_up_hrs == str:
        raise CustomException("time(Hour) invalid")
    if pick_up_min == str:
        raise CustomException("time(Minutes) invalid")

    if pick_up_hrs >= 24:
        raise CustomException("time(Hour) invalid")
    if pick_up_min >= 60:
        raise CustomException("time(Minutes) invalid")
    else:
        trip_date = datetime.strptime(pick_up_date, "%Y/%m/%d").date()
        trip_time = str(pick_up_hrs) + ":" + str(pick_up_min) + ":00"
        time_final = datetime.strptime(trip_time, "%H:%M:%S").time()

        trip_date_and_time = datetime.combine(trip_date, time_final)
        today = datetime.now().isoformat(" ", "seconds")
        today_date = datetime.strptime(today, "%Y-%m-%d %H:%M:%S")

    if today_date > trip_date_and_time:
        raise CustomException("invalid time given")
    else:
        return value

@root_validator
def validate_location(cls, value):
    pick_up_location = value.get("pick_up_location")
    destination = value.get("destination")
    if not pick_up_location:
        raise CustomException("pickup location cannot be empty")
    if not destination:
        raise CustomException("destination location cannot be empty")
    return value

@root_validator
def validate_card_details(cls, value):
    payment_method = value.get("payment_method")
    credit_number = value.get("card_number")
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
exp_date = value.get("exp_date")
cvv = value.get("cvv")
if payment_method == "Credit Card":
    if credit_number == "":
        raise CustomException("credit number cannot be empty!!")
    if exp_date == "":
        raise CustomException("exp date cannot be empty!!")
    if cvv == "":
        raise CustomException("cvv cannot be empty!!")
    return value
else:
    return value
```

Package : booking_control.py

code :

```
from tkinter import messagebox
from helper.turbo_db import Turbo_db
from datetime import date, datetime
```

```
class BookingController:
    def __init__(self) -> None:
        self._connection = Turbo_db.turbo_connection()

    def booking_control(self, user_booking, booking_form_frame):
        if self.authenticate_booking(user_booking):
            self.booking_sucess(booking_form_frame)

    def authenticate_booking(self, user_booking):
        booking_created_at_date = date.today()
        now = datetime.now()
        booking_created_at_time = now.strftime("%H:%M:%S")

    try:
        cursor = self._connection.cursor()
        statement = """CREATE TABLE IF NOT EXISTS booking(
            booking_id      SERIAL PRIMARY KEY,
            user_id        INT NOT NULL,
            firstname       VARCHAR(50) NOT NULL,
            lastname        VARCHAR(50) NOT NULL,
            no_of_passenger  INT NOT NULL,
            no_of_taxis      INT NOT NULL,
            pickup_date     DATE NOT NULL,
            pickup_time_hrs  INT NOT NULL,
            pickup_time_min   INT NOT NULL,
            pickup_location   VARCHAR(100) NOT NULL,)"""
    
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
destination      VARCHAR(100) NOT NULL,  
total_cost       VARCHAR(20) NOT NULL,  
payment_method   VARCHAR(20) NOT NULL,  
card_number      VARCHAR(50),  
card_exp         VARCHAR(20),  
card_cvv         VARCHAR(10),  
booking_status   VARCHAR(20),  
created_at_date  DATE NOT NULL,  
created_at_time  VARCHAR(30),  
pickup_coordinate VARCHAR(100),  
destination_coordinate VARCHAR(100),  
driver_id        INT
```

```
);""";
```

```
dataInsert = """INSERT INTO booking(user_id, firstname, lastname, \  
no_of_passenger, no_of_taxi, pickup_date, pickup_time_hrs, \  
pickup_time_min, pickup_location, destination, total_cost, \  
payment_method, card_number, card_exp, card_cvv, booking_status, created_at_date, \  
created_at_time, \  
pickup_coordinate, destination_coordinate) \  
VALUES  
(%s, %s, %s);"""
```

```
dataValues = (  
    user_booking.user_id,  
    user_booking.firstname,  
    user_booking.lastname,  
    user_booking.no_of_pass,  
    user_booking.no_of_taxi,  
    user_booking.pick_up_date,  
    user_booking.pick_up_hrs,  
    user_booking.pick_up_min,  
    user_booking.pick_up_location,  
    user_booking.destination,  
    user_booking.total_cost,  
    user_booking.payment_method,  
    user_booking.card_number,  
    user_booking.exp_date,  
    user_booking.cvv,  
    "Pending",  
    booking_created_at_date,  
    booking_created_at_time,  
    user_booking.pickup_coordinates,  
    user_booking.destination_coordinates,  
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    cursor.execute(statement)
    cursor.execute(dataInsert, dataValues)
    self._connection.commit()
except Exception as error:
    print(error)
finally:
    if cursor is not None:
        cursor.close()
    if self._connection is not None:
        self._connection.close()
return True
```

```
def booking_sucess(self, booking_form_frame):
    messagebox.showinfo(
        "Booking Completed",
        "Congatulation, Your Booking Request Has Been Sent",
    )
    booking_form_frame.destroy()
```

- **Module : admin**

Package : admin_model.py

code :

```
from pydantic import BaseModel

class AdminModel(BaseModel):
```

```
    username: str = ""
    password: str = ""
```

Package : admin_login.py

code :

```
import tkinter as tk
from admin.admin_controllers import AdminController
from helper.constants import BG_LOCATION
from .admin_model import AdminModel
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
class LogInPage:  
    def __init__(self, root):  
        self.root = root  
        self.create_login_frame()  
  
    def create_login_frame(self):  
        self.login_frame = tk.Frame(self.root)  
        self.login_frame.configure(background="#FFFFFF")  
        self.login_frame.place(relx=-0.07, rely=0, height=768, width=1366)  
        self.build_login_frame(login_frame=self.login_frame, root=self.root)
```

```
@staticmethod  
def build_login_frame(login_frame, root):
```

```
    welcome = tk.Label(login_frame)  
    welcome.place(relx=0.650, rely=0.149, height=59, width=200)  
    welcome.configure(  
        anchor="w",  
        background="#FFFFFF",  
        compound="left",  
        font="-family {Noto Sans} -size 22 -weight bold",  
        text="Admin Log In",  
    )
```

```
    username_label = tk.Label(login_frame)  
    username_label.place(relx=0.585, rely=0.299, height=21, width=206)  
    username_label.configure(  
        anchor="w",  
        background="#FFFFFF",  
        compound="left",  
        font="-family {Noto Sans} -size 16",  
        text="Username",  
    )
```

```
    username_entry = tk.Entry(login_frame)  
    username_entry.place(relx=0.585, rely=0.353, height=33, relwidth=0.301)  
    username_entry.configure(  
        background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"  
    )
```

```
    password_label = tk.Label(login_frame)  
    password_label.place(relx=0.585, rely=0.434, height=20, width=185)  
    password_label.configure(  
        anchor="w",  
        background="#FFFFFF",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        compound="left",
        font="-family {Noto Sans} -size 16",
        text="Password",
    )
```

```
password_entry = tk.Entry(login_frame)
password_entry.place(relx=0.585, rely=0.488, height=33, relwidth=0.301)
password_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
    show="*",
)
```

```
login_button = tk.Button(login_frame)
login_button.place(relx=0.674, rely=0.611, height=43, width=201)
login_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12 -weight bold",
    foreground="#FFFFFF",
    text="Log In",
    command=lambda: LogInPage.admin_login(
        username_entry,
        password_entry,
        login_frame,
        root,
    ),
)
```

```
background_label = tk.Label(login_frame)
background_label.place(relx=-0.029, rely=0.0, height=739, width=700)
background_label.configure(anchor="w", compound="left")
background_image = tk.PhotoImage(file=BG_LOCATION)
background_label.configure(image=background_image)
```

```
exit_button = tk.Button(login_frame)
exit_button.place(relx=0.100, rely=0.865, height=43, width=111)
exit_button.configure(
    activebackground="beige",
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#FFFFFF",
    text="Back",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        command=lambda: LogInPage.exit_application(login_frame),  
    )
```

```
show_pass_button = tk.Button(login_frame)  
show_pass_button.place(relx=0.900, rely=0.500, height=23, width=51)  
show_pass_button.configure(  
    activebackground="beige",  
    borderwidth="2",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    text="OO",  
)
```

```
show_pass_button.bind(  
    "<Button-1>",  
    lambda event: LogInPage.show_pass(  
        event,  
        password_entry,  
    ),  
)
```

```
show_pass_button.bind(  
    "<ButtonRelease-1>",  
    lambda event: LogInPage.hide_pass(  
        event,  
        password_entry,  
    ),  
)
```

```
login_frame.mainloop()
```

```
@staticmethod  
def show_pass(  
    event,  
    password_entry,  
):  
    password = password_entry.get()  
    password_entry.config(show="", text=password)
```

```
@staticmethod  
def hide_pass(  
    event,  
    password_entry,  
):  
    password_entry.config(show="*")
```

```
def exit_application(login_frame):
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
login_frame.destroy()
```

```
@staticmethod
def admin_login(
    username_entry,
    password_entry,
    login_frame,
    root,
):
    admin = AdminModel(
        username=username_entry.get(),
        password=password_entry.get(),
    )
    user_control = AdminController()
    user_control.admin_login_control(admin, login_frame, root)
```

Package : admin_controller.py

code :

```
from tkinter import messagebox
from helper.turbo_db import Turbo_db
from .control_panel import ControlPanelPage
```

```
class AdminController:
    def __init__(self) -> None:
        self._connection = Turbo_db.turbo_connection()

    def admin_login_control(self, admin, login_frame, root):
        if self.login_authenticate(admin):
            self.login_sucess(root, login_frame)
        else:
            messagebox.showerror(
                "Invalid Data",
                "username or password not matched",
            )

    def login_authenticate(self, admin):
        try:
            cursor = self._connection.cursor()
            statement = "SELECT * FROM admin WHERE username=%s AND password = %s;"
            data = (admin.username, admin.password)
            cursor.execute(statement, data)
            self.record = cursor.fetchall()
            if self.record:
                return True
            return False
        except Exception as error:
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    print(error)
```

```
def login_sucess(self, root, login_page):
    messagebox.showinfo(title="Congratulation", message="Welcome ")
    login_page.destroy()
    ControlPanelPage(root, login_page, AdminController)
```

```
def customer_data_fetcher(self):
```

```
    try:
        cursor = self._connection.cursor()
        statement = "SELECT u.userid, b.booking_id, b.created_at_date, CONCAT(u.firstname,'
,u.lastname) as fullname, u.contact, u.address, u.email, b.booking_status from users as u JOIN booking as
b on u.userid = b.user_id where b.booking_status = 'Pending' order by b.created_at_date,
b.created_at_time;"
        cursor.execute(statement)
        self.record = cursor.fetchall()
        return self.record
    except Exception as error:
        print(error)
```

```
def customer_search_fetcher(self, search_entry, selected_sort_by):
```

```
    if selected_sort_by.get() == "Pending":
        statement = "SELECT u.userid, b.booking_id, b.created_at_date, CONCAT(u.firstname,'
,u.lastname) as fullname, u.contact, u.address, u.email, b.booking_status from users as u JOIN booking as
b on u.userid = b.user_id where concat(u.firstname , ' ', u.lastname) like concat('%%','%s','%%') and
b.booking_status = 'Pending' order by b.created_at_date, b.created_at_time;"
    else:
        if selected_sort_by.get() == "Accepted":
            statement = "SELECT u.userid, b.booking_id, b.created_at_date, CONCAT(u.firstname,'
,u.lastname) as fullname, u.contact, u.address, u.email, b.booking_status from users as u JOIN booking as
b on u.userid = b.user_id where concat(u.firstname , ' ', u.lastname) like concat('%%','%s','%%') and
b.booking_status = 'Accepted' order by b.created_at_date, b.created_at_time;"
```

```
    else:
        if selected_sort_by.get() == "Canceled":
            statement = "SELECT u.userid, b.booking_id, b.created_at_date, CONCAT(u.firstname,'
,u.lastname) as fullname, u.contact, u.address, u.email, b.booking_status from users as u JOIN booking as
b on u.userid = b.user_id where concat(u.firstname , ' ', u.lastname) like concat('%%','%s','%%') and
b.booking_status = 'Canceled' order by b.created_at_date, b.created_at_time;"
```

```
    else:
        if selected_sort_by.get() == "All":
            statement = "SELECT u.userid, b.booking_id, b.created_at_date, CONCAT(u.firstname,'
,u.lastname) as fullname, u.contact, u.address, u.email, b.booking_status from users as u JOIN booking as
b on u.userid = b.user_id where concat(u.firstname , ' ', u.lastname) like concat('%%','%s','%%') order by
b.created_at_date, b.created_at_time;"
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
try:  
    cursor = self._connection.cursor()  
    data = (search_entry.get(),)  
    cursor.execute(statement, data)  
    self.record = cursor.fetchall()  
    return self.record  
except Exception as error:  
    print(error)
```

```
def driver_data_fetcher(self):
```

```
try:  
    cursor = self._connection.cursor()  
    statement = "SELECT * FROM drivers;"  
    cursor.execute(statement)  
    self.record = cursor.fetchall()  
    return self.record  
except Exception as error:  
    print(error)
```

```
def driver_search_fetcher(self, search_entry, selected_sort_by):
```

```
if selected_sort_by.get() == "Driver ID":  
    statement = "SELECT * FROM drivers WHERE fullname like concat('%%',%s,'%%') ORDER  
BY driverid;"  
else:  
    if selected_sort_by.get() == "Full Name":  
        statement = "SELECT * FROM drivers WHERE fullname like concat('%%',%s,'%%') ORDER  
BY fullname;"  
    else:  
        if selected_sort_by.get() == "Available":  
            statement = "SELECT * FROM drivers WHERE fullname like concat('%%',%s,'%%') and  
driver_status = 'Available' ORDER BY driverid;"  
        else:  
            if selected_sort_by.get() == "Booked":  
                statement = "SELECT * FROM drivers WHERE fullname like concat('%%',%s,'%%')  
and driver_status = 'Booked' ORDER BY driverid;"  
            else:  
                return  
try:  
    cursor = self._connection.cursor()  
    data = (search_entry.get(),)  
    cursor.execute(statement, data)  
    self.record = cursor.fetchall()  
    return self.record  
except Exception as error:  
    print(error)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
def booking_details_data_fetcher(self, selected_booking_id):
```

```
    try:
        statement = "SELECT * FROM booking WHERE booking_id = %s"
        cursor = self._connection.cursor()
        data = selected_booking_id
        cursor.execute(statement, data)
        self.record = cursor.fetchall()
        return self.record
    except Exception as error:
        print(error)
```

```
def cancel_booking(self, booking_id):
```

```
    try:
        statement = "UPDATE booking SET booking_status = 'Canceled' WHERE booking_id
= %s AND booking_status = 'Pending';"

        b_id = str(booking_id)
        cursor = self._connection.cursor()
        data = b_id
        cursor.execute(statement, data)
        self._connection.commit()

    except Exception as error:
        print(error)
```

```
def available_driver_fetcher(self):
```

```
    try:
        statement = "SELECT * FROM drivers WHERE driver_status = 'Available'"
        cursor = self._connection.cursor()
        cursor.execute(statement)
        self.record = cursor.fetchall()
        return self.record
    except Exception as error:
        print(error)
```

```
def booked_driver_fetcher(self):
```

```
    try:
        statement = "SELECT * FROM drivers WHERE driver_status = 'Booked'"
        cursor = self._connection.cursor()
        cursor.execute(statement)
        self.record = cursor.fetchall()
        return self.record
    except Exception as error:
        print(error)
```

```
def assign_controller(self, driver_id, booking_id):
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
try:  
    booking_statement = "Update booking set booking_status = 'Accepted', driver_id = %s WHERE  
booking_id = %s ;"  
    did = str(driver_id)  
    bid = str(booking_id)  
    booking_data = (did, bid)  
  
    driver_statement = (  
        "Update drivers set driver_status = 'Booked' where driverid = %s;"  
    )  
  
    driver_data = did  
  
    cursor = self._connection.cursor()  
    cursor.execute(booking_statement, booking_data)  
    cursor.execute(driver_statement, driver_data)  
    self._connection.commit()  
except Exception as error:  
    print(error)
```

Package : control_panel.py

code :

```
from select import select  
from sre_parse import State  
import tkinter as tk  
from tkinter import W, Scrollbar, ttk  
from tkinter import messagebox  
from tkinter.messagebox import NO  
  
from click import command
```

```
from taxi.taxi_register import TaxiRegisterPage  
from driver.driver_register import DriverRegisterPage  
from tkintermapview import TkinterMapView
```

```
class ControlPanelPage:  
    def __init__(self, root, login_frame, admin_controller):  
        self.root = root  
        self.admin_controller = admin_controller  
        self.login_frame = login_frame  
        self.create_control_panel_frame()
```

```
    def create_control_panel(self):  
        self.control_panel_frame = tk.Frame(self.root)  
        self.control_panel_frame.configure(background="#FFFFFF")  
        self.control_panel_frame.place(relx=-0.07, rely=0, height=768, width=1366)  
        self.build_control_panel_frame()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    self.root, self.control_panel_frame, self.admin_controller  
)  
)
```

```
@staticmethod  
def build_control_panel_frame(root, control_panel_frame, admin_controller):  
    title = tk.Label(control_panel_frame)  
    title.place(relx=0.450, rely=0.025, height=71, width=224)  
    title.configure(  
        anchor="w",  
        background="#FFFFFF",  
        compound="left",  
        font="-family {Noto Sans} -size 22 -weight bold",  
        text="\"Control Panel\"",  
    )
```

```
    line_seperator = ttk.Separator(control_panel_frame)  
    line_seperator.place(relx=0.130, rely=0.123, relwidth=0.800)
```

```
    taxi_register_button = tk.Button(control_panel_frame)  
    taxi_register_button.place(relx=0.850, rely=0.130, height=33, width=101)  
    taxi_register_button.configure(  
        activebackground="beige",  
        background="#007074",  
        borderwidth="2",  
        compound="left",  
        font="-family {Noto Sans} -size 9 -weight bold",  
        foreground="#FFFFFF",  
        text="\"Taxi Register\"",  
        command=lambda: TaxiRegisterPage(  
            root,  
            control_panel_frame,  
        ),  
    )
```

```
    driver_register_button = tk.Button(control_panel_frame)  
    driver_register_button.place(relx=0.750, rely=0.130, height=33, width=110)  
    driver_register_button.configure(  
        activebackground="beige",  
        background="#007074",  
        borderwidth="2",  
        compound="left",  
        font="-family {Noto Sans} -size 9 -weight bold",  
        foreground="#FFFFFF",  
        text="\"Driver Register\"",  
        command=lambda: DriverRegisterPage(  
            root,  
            control_panel_frame,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
customer_table_button = tk.Button(control_panel_frame)
customer_table_button.place(relx=0.130, rely=0.850, height=33, width=101)
customer_table_button.configure(
    activebackground="beige",
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 9 -weight bold",
    foreground="#FFFFFF",
    text="Customer",
    command=lambda: ControlPanelPage.customer_tabel_data_fetcher(
        admin_controller, control_panel_frame
    ),
)
```

```
driver_table_button = tk.Button(control_panel_frame)
driver_table_button.place(relx=0.220, rely=0.850, height=33, width=101)
driver_table_button.configure(
    activebackground="beige",
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 9 -weight bold",
    foreground="#FFFFFF",
    text="Driver",
    command=lambda: ControlPanelPage.driver_tabel_data_fetcher(
        admin_controller,
        control_panel_frame,
    ),
)
```

```
ControlPanelPage.customer_tabel_data_fetcher(
    admin_controller, control_panel_frame
)
control_panel_frame.mainloop()
```

```
@staticmethod
def customer_table_frame_build(
    record,
    control_panel_frame,
    admin_controller,
):
```

```
    customer_table_frame = tk.Frame(control_panel_frame)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
customer_table_frame.place(  
    relx=0.130,  
    rely=0.200,  
    height=470,  
    width=1095,  
)
```

```
customer_table_frame.config(  
    background="#FFFFFF",  
    borderwidth=2,  
    relief="solid",  
    highlightbackground="#000000",  
)
```

```
table_title = tk.Label(customer_table_frame)  
table_title.place(relx=0.020, rely=0.010, height=60, width=200)  
table_title.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    text="""Customer Table""",  
)
```

```
search_entry = tk.Entry(customer_table_frame)  
search_entry.place(relx=0.200, rely=0.050, height=25, relwidth=0.150)  
search_entry.configure(  
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"  
)
```

```
search_button = tk.Button(customer_table_frame)  
search_button.place(relx=0.360, rely=0.050, height=25, relwidth=0.069)  
search_button.configure(  
    activebackground="beige",  
    borderwidth="1",  
    compound="left",  
    font="-family {Noto Sans} -size 10",  
    text="""Search""",  
    command=lambda: ControlPanelPage.customer_tabel_search_fetcher(  
        admin_controller,  
        search_entry,  
        selected_sort_by,  
        customer_table_frame,  
    ),  
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
selected_sort_by = tk.StringVar()
```

```
sort_by_combo = ttk.Combobox(  
    customer_table_frame,  
    textvariable=selected_sort_by,  
    state="readonly",  
    values=[  
        "Pending",  
        "Accepted",  
        "Canceled",  
        "All",  
    ],  
)
```

```
sort_by_combo.place(relx=0.460, rely=0.050, height=25, relwidth=0.150)  
sort_by_combo.current(0)
```

```
sort_by_combo.bind(  
    "<<ComboboxSelected>>",  
    lambda event: ControlPanelPage.customer_tabel_search_fetcher(  
        admin_controller,  
        search_entry,  
        selected_sort_by,  
        customer_table_frame,  
    ),  
)
```

```
sort_by_label = tk.Label(  
    customer_table_frame,  
    text=": View",  
    anchor=W,  
    background="#FFFFFF",  
    foreground="#4A4A4A",  
)
```

```
sort_by_label.place(  
    relx=0.620,  
    rely=0.050,  
    height=25,  
    relwidth=0.060,  
)
```

```
ControlPanelPage.customer_table_build(  
    customer_table_frame,  
    record,  
    admin_controller,  
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
@staticmethod
def driver_table_frame_build(record, control_panel_frame, admin_controller):
    driver_table_frame = tk.Frame(control_panel_frame)

    driver_table_frame.place(
        relx=0.130,
        rely=0.200,
        height=470,
        width=1095,
    )

    driver_table_frame.config(
        background="#FFFFFF",
        borderwidth=2,
        relief="solid",
        highlightbackground="#000000",
    )

    table_title = tk.Label(driver_table_frame)
    table_title.place(relx=0.020, rely=0.010, height=60, width=200)
    table_title.configure(
        anchor="w",
        background="#FFFFFF",
        compound="left",
        font="-family {Noto Sans} -size 14",
        text="Driver Table",
    )

    search_entry = tk.Entry(driver_table_frame)

    search_entry.place(relx=0.200, rely=0.050, height=25, relwidth=0.150)
    search_entry.configure(
        background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"
    )

    search_button = tk.Button(driver_table_frame)

    search_button.place(relx=0.360, rely=0.050, height=25, relwidth=0.069)
    search_button.configure(
        activebackground="beige",
        borderwidth="1",
        compound="left",
        font="-family {Noto Sans} -size 10",
        text="Search",
        command=lambda: ControlPanelPage.driver_table_search_fetcher(
            admin_controller,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    search_entry,  
    selected_sort_by,  
    driver_table_frame,  
)  
)
```

```
selected_sort_by = tk.StringVar()
```

```
sort_by_combo = ttk.Combobox(  
    driver_table_frame,  
    textvariable=selected_sort_by,  
    state="readonly",  
    values=("Driver ID", "Full Name", "Available", "Booked"),  
)
```

```
sort_by_combo.place(relx=0.460, rely=0.050, height=25, relwidth=0.150)  
sort_by_combo.current(0)
```

```
sort_by_combo.bind(  
    "<    lambda event: ControlPanelPage.driver_tabel_search_fetcher(  
        admin_controller,  
        search_entry,  
        selected_sort_by,  
        driver_table_frame,  
)  
)
```

```
sort_by_label = tk.Label(  
    driver_table_frame,  
    text=": sort by",  
    anchor=W,  
    background="#FFFFFF",  
    foreground="#4A4A4A",  
)
```

```
sort_by_label.place(  
    relx=0.620,  
    rely=0.050,  
    height=25,  
    relwidth=0.060,  
)
```

```
ControlPanelPage.driver_table_build(driver_table_frame, record)
```

```
@staticmethod  
def driver_table_build(dirver_table_frame, record):
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
# Driver TABLE LOOK
table_style = ttk.Style()
table_style.theme_use("default")
table_style.configure(
    "Treeview",
    background="#FFFFFF",
    foreground="#4A4A4A",
    rowheight="35",
    fieldbackground="#FFFFFF",
)
table_style.map("Treeview", background=[("selected", "#C9C9C9")])
```

```
# driver Table Scroll Bar
```

```
table_scroll_bar = Scrollbar(dirver_table_frame)
table_scroll_bar.place(relx=0.980, rely=0.020, height=450, width=15)
```

```
# driver Table Build
```

```
driver_booking_table = ttk.Treeview(
    dirver_table_frame,
    yscrollcommand=table_scroll_bar.set,
    selectmode="extended",
)
driver_booking_table.place(
    relx=0.020,
    rely=0.150,
    height=375,
    width=1025,
)
```

```
# driver a scroll bar
```

```
table_scroll_bar.config(
    command=driver_booking_table.yview,
)
# Define Column
```

```
driver_booking_table["columns"] = (
    "Driver ID",
    "Full Name",
    "License Number",
    "Contact",
    "Taxi Number",
    "Status",
)
# Format Our Columns
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
driver_booking_table.column("#0", width=0, stretch=NO)
driver_booking_table.column("Driver ID", width=70, anchor=W)
driver_booking_table.column("Full Name", width=140, anchor=W)
driver_booking_table.column("License Number", width=120, anchor=W)
driver_booking_table.column("Contact", width=120, anchor=W)
driver_booking_table.column("Taxi Number", width=120, anchor=W)
driver_booking_table.column("Status", width=120, anchor=W)
```

Create Heading

```
driver_booking_table.heading("#0", text="", anchor=W)
driver_booking_table.heading("Driver ID", text="Driver ID", anchor=W)
driver_booking_table.heading("Full Name", text="Full Name", anchor=W)
driver_booking_table.heading("License Number", text="License Number", anchor=W)
driver_booking_table.heading("Contact", text="Contact", anchor=W)
driver_booking_table.heading("Taxi Number", text="Taxi Number", anchor=W)
driver_booking_table.heading("Status", text="Status", anchor=W)
```

```
for data in record:
    driver_booking_table.insert(
        "",
        index="end",
        values=[
            data[0],
            data[1],
            data[2],
            data[3],
            data[4],
            data[5],
        ],
    )
dirver_table_frame.mainloop()
```

@staticmethod

```
def customer_table_build(customer_table_frame, record, admin_controller):
```

```
# CUSTOMISING TABLE LOOK
table_style = ttk.Style()
table_style.theme_use("default")
table_style.configure(
    "Treeview",
    background="#FFFFFF",
    foreground="#4A4A4A",
    rowheight="35",
    fieldbackground="#FFFFFF",
)
table_style.map("Treeview", background=[("selected", "#C9C9C9")])
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
# Customer Table Scroll Bar
```

```
table_scroll_bar = Scrollbar(customer_table_frame)
table_scroll_bar.place(relx=0.980, rely=0.020, height=450, width=15)
```

```
# Customer Table Build
```

```
customer_booking_table = ttk.Treeview(
    customer_table_frame,
    yscrollcommand=table_scroll_bar.set,
    selectmode="extended",
)
customer_booking_table.place(
    relx=0.020,
    rely=0.150,
    height=375,
    width=1025,
)
```

```
customer_booking_table.bind(
    "<Double-1>",
    lambda event: ControlPanelPage.assign_taxi_to_user(
        event,
        customer_booking_table,
        admin_controller,
    ),
)
```

```
# Configure a scroll bar
```

```
table_scroll_bar.config(
    command=customer_booking_table.yview,
)
```

```
# Define Column
```

```
customer_booking_table["columns"] = (
    "User ID",
    "Booking ID",
    "Created At",
    "Full Name",
    "Contact",
    "Address",
    "Email",
    "Booking Status",
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

Format Our Columns

```
customer_booking_table.column("#0", width=0, stretch=NO)
customer_booking_table.column("User ID", width=70, anchor=W)
customer_booking_table.column("Booking ID", width=70, anchor=W)
customer_booking_table.column("Created At", width=120, anchor=W)
customer_booking_table.column("Full Name", width=140, anchor=W)
customer_booking_table.column("Contact", width=70, anchor=W)
customer_booking_table.column("Address", width=170, anchor=W)
customer_booking_table.column("Email", width=180, anchor=W)
customer_booking_table.column("Booking Status", width=100, anchor=W)
```

Create Heading

```
customer_booking_table.heading("#0", text="", anchor=W)
customer_booking_table.heading("User ID", text="User ID", anchor=W)
customer_booking_table.heading("Booking ID", text="Booking ID", anchor=W)
customer_booking_table.heading("Created At", text="Created At", anchor=W)
customer_booking_table.heading("Full Name", text="Full Name", anchor=W)
customer_booking_table.heading("Contact", text="Contact", anchor=W)
customer_booking_table.heading("Address", text="Address", anchor=W)
customer_booking_table.heading("Email", text="Email", anchor=W)
customer_booking_table.heading(
    "Booking Status", text="Booking Status", anchor=W
)
```

```
for data in record:
    customer_booking_table.insert(
        parent="",
        index="end",
        text="",
        values=(
            data[0],
            data[1],
            data[2],
            data[3],
            data[4],
            data[5],
            data[6],
            data[7],
        ),
    )
```

customer_table_frame.mainloop()

@staticmethod

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
def customer_table_data_fetcher(admin_controller, control_panel_frame):  
    user_booking = admin_controller()  
    required_customer_data = user_booking.customer_data_fetcher()
```

```
    ControlPanelPage.customer_table_frame_build(  
        required_customer_data,  
        control_panel_frame,  
        admin_controller,  
    )
```

```
@staticmethod  
def driver_table_data_fetcher(admin_controller, control_panel_frame):  
    user_booking = admin_controller()  
    required_customer_data = user_booking.driver_data_fetcher()
```

```
    ControlPanelPage.driver_table_frame_build(  
        required_customer_data,  
        control_panel_frame,  
        admin_controller,  
    )
```

```
@staticmethod  
def customer_table_search_fetcher(  
    admin_controller,  
    search_entry,  
    selected_sort_by,  
    customer_table_frame,  
):  
    user_booking = admin_controller()  
    required_customer_search = user_booking.customer_search_fetcher(  
        search_entry,  
        selected_sort_by,  
    )
```

```
    ControlPanelPage.customer_table_build(  
        customer_table_frame,  
        required_customer_search,  
        admin_controller,  
    )
```

```
@staticmethod  
def driver_table_search_fetcher(  
    admin_controller,  
    search_entry,  
    selected_sort_by,  
    driver_table_frame,  
):
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
user_booking = admin_controller()
required_customer_search = user_booking.driver_search_fetcher(
    search_entry,
    selected_sort_by,
)
```

```
ControlPanelPage.driver_table_build(
    driver_table_frame,
    required_customer_search,
)
```

```
@staticmethod
def assign_taxi_to_user(event, customer_booking_table, admin_controller):
```

```
selected = customer_booking_table.focus()
selected_row = customer_booking_table.item(
    selected,
    "values",
)
selected_booking_id = selected_row[1]
```

```
# Booking Details Data Fetcher
```

```
booking_data_control = admin_controller()
fetched_booking_details = booking_data_control.booking_details_data_fetcher(
    selected_booking_id,
)
```

```
data = fetched_booking_details[0]
```

```
assign_taxi_frame = tk.Toplevel(customer_booking_table)
assign_taxi_frame.title("Assign Taxi")
assign_taxi_frame.geometry("682x641+193+31")
assign_taxi_frame.resizable(0, 0)
assign_taxi_frame.configure(background="#FFFFFF")
```

```
# BUILDING BOOKING DETAIL PAGE
```

```
title = tk.Label(assign_taxi_frame)
title.place(relx=0.360, rely=0.031, height=43, width=250)
title.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 18 -weight bold",
    text="Booking Details",
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
fullname_label = tk.Label(assign_taxi_frame)
fullname_label.place(relx=0.029, rely=0.156, height=34, width=122)
fullname_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="Full Name :",
)
```

```
fullname_data = tk.Label(assign_taxi_frame)
fullname_data.place(relx=0.22, rely=0.156, height=34, width=280)
fullname_data.configure(
    anchor="w",
    background="#FFFFFF",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[2] + " " + data[3],
)
```

```
map_view_frame = tk.Frame(assign_taxi_frame)
map_view_frame.place(relx=0.640, rely=0.156, height=140, width=220)
map_view_frame.configure(
    background="#FFFFFF",
    borderwidth="2",
    highlightthickness="2",
)
```

```
location_detail_map_view = TkinterMapView(map_view_frame)
location_detail_map_view.set_tile_server(
    "https://mt0.google.com/vt/lyrs=m&hl=en&x={x}&y={y}&z={z}&s=Ga",
    max_zoom=60,
)
```

```
pickup_cordinates = eval(data[19])
destination_coordinates = eval(data[20])
```

```
location_detail_map_view.set_marker(
    pickup_cordinates[0],
    pickup_cordinates[1],
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
location_detail_map_view.set_marker(  
    destination_coordinates[0],  
    destination_coordinates[1],  
)  
  
location_detail_map_view.set_position(  
    pickup_cordinates[0],  
    pickup_cordinates[1],  
)  
  
location_detail_map_view.set_path(  
    position_list=[pickup_cordinates, destination_coordinates]  
)  
  
location_detail_map_view.pack()  
  
passenger_no_label = tk.Label(assign_taxi_frame)  
passenger_no_label.place(relx=0.029, rely=0.234, height=34, width=175)  
passenger_no_label.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    text="No of Passenger :")  
  
passenger_no_data = tk.Label(assign_taxi_frame)  
passenger_no_data.place(relx=0.293, rely=0.234, height=34, width=48)  
passenger_no_data.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    highlightthickness="2",  
    text=data[4],  
)  
  
taxi_number = tk.Label(assign_taxi_frame)  
taxi_number.place(relx=0.029, rely=0.312, height=35, width=111)  
taxi_number.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""No of Taxi :"",
)
```

```
taxi_number_data = tk.Label(assign_taxi_frame)
taxi_number_data.place(relx=0.22, rely=0.312, height=35, width=48)
taxi_number_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[5],
)
```

```
pick_up_date_lable = tk.Label(assign_taxi_frame)
pick_up_date_lable.place(relx=0.029, rely=0.39, height=34, width=133)
pick_up_date_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""Pick Up Date :"",
)
```

```
pick_up_date_data = tk.Label(assign_taxi_frame)
pick_up_date_data.place(relx=0.249, rely=0.39, height=35, width=163)
pick_up_date_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[6],
)
```

```
pick_up_address_label = tk.Label(assign_taxi_frame)
pick_up_address_label.place(relx=0.029, rely=0.468, height=35, width=175)
pick_up_address_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
background="#FFFFFF",
compound="left",
font="-family {Noto Sans} -size 14",
foreground="#4A4A4A",
text="""Pick Up Address :"",
}

time_label = tk.Label(assign_taxi_frame)
time_label.place(relx=0.513, rely=0.39, height=34, width=69)
time_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""Time :"",
)
data7 = str(data[7])
data8 = str(data[8])
time_data = tk.Label(assign_taxi_frame)
time_data.place(relx=0.616, rely=0.39, height=34, width=100)
time_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=(data7 + ":" + data8),
)
pickup_address_data = tk.Label(assign_taxi_frame)
pickup_address_data.place(relx=0.293, rely=0.468, height=35, width=323)
pickup_address_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 10",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[9],
)
destination_label = tk.Label(assign_taxi_frame)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
destination_label.place(relx=0.029, rely=0.546, height=34, width=132)
```

```
destination_label.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    text="Destination :")
```

```
destination_address_lable = tk.Label(assign_taxi_frame)  
destination_address_lable.place(relx=0.249, rely=0.546, height=34, width=323)  
destination_address_lable.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 10",  
    foreground="#4A4A4A",  
    highlightthickness="2",  
    text=data[10],  
)
```

```
total_cost_label = tk.Label(assign_taxi_frame)  
total_cost_label.place(relx=0.029, rely=0.624, height=34, width=111)  
total_cost_label.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    text="Total Cost :")
```

```
total_cost_data = tk.Label(assign_taxi_frame)  
total_cost_data.place(relx=0.22, rely=0.624, height=34, width=206)  
total_cost_data.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    highlightthickness="2",  
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
    text=data[11] + ".NPR",
)
```

```
payment_method_lable = tk.Label(assign_taxi_frame)
payment_method_lable.place(relx=0.029, rely=0.702, height=33, width=185)
payment_method_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="\"Payment Method :\"",
```

```
)
```

```
payment_method_data = tk.Label(assign_taxi_frame)
payment_method_data.place(relx=0.293, rely=0.702, height=34, width=206)
payment_method_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[12],
```

```
)
```

```
credit_number_lable = tk.Label(assign_taxi_frame)
credit_number_lable.place(relx=0.029, rely=0.78, height=34, width=236)
credit_number_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    text=data[13],
    highlightthickness="2",
    foreground="#4A4A4A",
    font="-family {Noto Sans} -size 14",
    compound="left",
```

```
)
```

```
exp_date_lable = tk.Label(assign_taxi_frame)
exp_date_lable.place(relx=0.411, rely=0.78, height=34, width=96)
exp_date_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        compound="left",
        font="-family {Noto Sans} -size 14",
        foreground="#4A4A4A",
        highlightthickness="2",
        text=data[14],
    )
```

```
cvv_lable = tk.Label(assign_taxi_frame)
cvv_lable.place(relx=0.587, rely=0.78, height=34, width=96)
cvv_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[15],
)
```

```
show_credit_detail_lable = tk.Checkbutton(assign_taxi_frame)
show_credit_detail_lable.place(
    relx=0.748, rely=0.78, relheight=0.051, relwidth=0.164
)
show_credit_detail_lable.configure(
    activebackground="beige",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    highlightthickness="0",
    justify="left",
    selectcolor="#d9d9d9",
    text="Show Details",
)
```

```
assign_taxi_button = tk.Button(assign_taxi_frame)
assign_taxi_button.place(relx=0.264, rely=0.889, height=33, width=141)
assign_taxi_button.configure(
    activebackground="beige",
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 10 -weight bold",
    foreground="#FFFFFF",
    text="Assign Taxi",
    command=lambda: ControlPanelPage.taxi_assign_final_frame(
        assign_taxi_frame, admin_controller, data
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
cancel_booking_button = tk.Button(assign_taxi_frame)
cancel_booking_button.place(relx=0.499, rely=0.889, height=33, width=141)
cancel_booking_button.configure(
    activebackground="beige",
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 10 -weight bold",
    text="Cancel Booking",
    foreground="#FFFFFF",
    command=lambda: ControlPanelPage.cancel_booking_request(
        admin_controller, data, assign_taxi_frame
    ),
)
if data[16] == "Canceled":
    assign_taxi_button.config(state="disabled")
    cancel_booking_button.config(state="disabled")
```

```
if data[16] == "Accepted":
    assign_taxi_button.config(state="disabled")
```

```
back_button = tk.Button(assign_taxi_frame)
back_button.place(relx=0.865, rely=0.031, height=33, width=71)
back_button.configure(
    activebackground="beige",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 10",
    foreground="#FFFFFF",
    text="Back",
    command=lambda: ControlPanelPage.exit_assign_taxi_page(
        assign_taxi_frame,
    ),
)
```

```
assign_taxi_frame.wait_visibility()
assign_taxi_frame.grab_set()
assign_taxi_frame.mainloop()
```

```
@staticmethod
def exit_assign_taxi_page(assign_taxi_frame):
    assign_taxi_frame.destroy()
```

```
@staticmethod
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
def cancel_booking_request(admin_controller, data, assign_taxi_frame):
```

```
    booking_id = data[0]
```

```
    response = messagebox.askquestion(  
        "Warning",  
        "Do you really want to cancel this booking?",  
        parent=assign_taxi_frame,  
    )
```

```
    if response == "yes":  
        cancel_booking_control = admin_controller()  
        cancel_booking_control.cancel_booking(booking_id)  
        assign_taxi_frame.destroy()
```

```
@staticmethod
```

```
def taxi_assign_final_frame(assign_taxi_frame, admin_controller, booking_id):
```

```
    assign_table_frame = tk.Toplevel(assign_taxi_frame)  
    assign_table_frame.geometry("600x400+193+31")  
    assign_table_frame.resizable(0, 0)  
    assign_table_frame.configure(background="#FFFFFF")
```

```
    search_entry = tk.Entry(assign_table_frame)
```

```
    search_entry.place(relx=0.030, rely=0.070, height=25, relwidth=0.200)  
    search_entry.configure(  
        background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"  
    )
```

```
    search_button = tk.Button(assign_table_frame)
```

```
    search_button.place(relx=0.250, rely=0.070, height=25, relwidth=0.100)  
    search_button.configure(  
        activebackground="beige",  
        borderwidth="1",  
        compound="left",  
        font="-family {Noto Sans} -size 10",  
        text="Search",  
    )
```

```
    selected_driver_type = tk.StringVar()
```

```
    sort_by_combo = ttk.Combobox(  
        assign_table_frame,  
        textvariable=selected_driver_type,  
        state="readonly",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
values=(  
    "Available",  
    "Booked",  
)  
)
```

```
sort_by_combo.place(relx=0.370, rely=0.070, height=25, relwidth=0.200)  
sort_by_combo.current(0)
```

```
sort_by_combo.bind(  
    "<<ComboboxSelected>>",  
    lambda event: ControlPanelPage.available_driver_table_view(  
        assign_table_frame,  
        admin_controller,  
        selected_driver_type,  
        booking_id,  
        assign_taxi_frame,  
)  
)
```

```
# Define Column
```

```
if selected_driver_type.get() == "Available":  
    ControlPanelPage.available_driver_table_view(  
        assign_table_frame,  
        admin_controller,  
        selected_driver_type,  
        booking_id,  
        assign_taxi_frame,  
)
```

```
assign_table_frame.wait_visibility()  
assign_table_frame.grab_set()  
assign_table_frame.mainloop()
```

```
@staticmethod  
def available_driver_table_view(  
    assign_table_frame,  
    admin_controller,  
    selected_driver_type,  
    booking_id,  
    assign_taxi_frame,  
):
```

```
if selected_driver_type.get() == "Available":
```

```
    table_style = ttk.Style()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
table_style.theme_use("default")
table_style.configure(
    "Treeview",
    background="#FFFFFF",
    foreground="#4A4A4A",
    rowheight="35",
    fieldbackground="#FFFFFF",
)
table_style.map("Treeview", background=[("selected", "#C9C9C9")])
```

driver Table Scroll Bar

```
table_scroll_bar = Scrollbar(assign_table_frame)
table_scroll_bar.place(relx=0.970, rely=0.150, height=280, width=15)
```

driver Table Build

```
driver_booking_table = ttk.Treeview(
    assign_table_frame,
    yscrollcommand=table_scroll_bar.set,
    selectmode="extended",
)
driver_booking_table.place(
    relx=0.030,
    rely=0.150,
    height=280,
    width=550,
)
```

driver a scroll bar

```
table_scroll_bar.config(
    command=driver_booking_table.yview,
)
```

```
driver_booking_table["columns"] = (
    "Driver ID",
    "Full Name",
    "Contact",
    "Taxi Number",
)
```

Format Our Columns

```
driver_booking_table.column("#0", width=0, stretch=NO)
driver_booking_table.column("Driver ID", width=40, anchor=W)
driver_booking_table.column("Full Name", width=70, anchor=W)
driver_booking_table.column("Contact", width=70, anchor=W)
driver_booking_table.column("Taxi Number", width=80, anchor=W)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
# Create Heading
```

```
driver_booking_table.heading("#0", text="", anchor=W)
driver_booking_table.heading("Driver ID", text="Driver ID", anchor=W)
driver_booking_table.heading("Full Name", text="Full Name", anchor=W)
driver_booking_table.heading("Contact", text="Contact", anchor=W)
driver_booking_table.heading("Taxi Number", text="Taxi Number", anchor=W)
```

```
availavble_driver_control = admin_controller()
record = availavble_driver_control.available_driver_fetcher()
```

```
for data in record:
    driver_booking_table.insert(
        "",
        index="end",
        values=[
            data[0],
            data[1],
            data[3],
            data[4],
        ],
    )
```

```
assign_button = tk.Button(assign_table_frame)
assign_button.place(relx=0.400, rely=0.880, height=35, width=120)
assign_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#FFFFFF",
    text="""Assign""",
    command=lambda: ControlPanelPage.selcted_driver_row_assign(
        driver_booking_table,
        admin_controller,
        booking_id,
        assign_taxi_frame,
    ),
)
```

```
else:
```

```
table_style = ttk.Style()
table_style.theme_use("default")
table_style.configure(
    "Treeview",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        background="#FFFFFF",
        foreground="#4A4A4A",
        rowheight="35",
        fieldbackground="#FFFFFF",
    )
table_style.map("Treeview", background=[("selected", "#C9C9C9")])
```

driver Table Scroll Bar

```
table_scroll_bar = Scrollbar(assign_table_frame)
table_scroll_bar.place(relx=0.970, rely=0.150, height=280, width=15)
```

driver Table Build

```
driver_booking_table = ttk.Treeview(
    assign_table_frame,
    yscrollcommand=table_scroll_bar.set,
    selectmode="extended",
)
driver_booking_table.place(
    relx=0.030,
    rely=0.150,
    height=280,
    width=550,
)
# driver a scroll bar
```

```
table_scroll_bar.config(
    command=driver_booking_table.yview,
)
```

```
driver_booking_table["columns"] = (
    "Driver ID",
    "Full Name",
    "Contact",
    "Taxi Number",
)
# Format Our Columns
```

```
driver_booking_table.column("#0", width=0, stretch=NO)
driver_booking_table.column("Driver ID", width=40, anchor=W)
driver_booking_table.column("Full Name", width=70, anchor=W)
driver_booking_table.column("Contact", width=70, anchor=W)
driver_booking_table.column("Taxi Number", width=80, anchor=W)
```

Create Heading

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
driver_booking_table.heading("#0", text="", anchor=W)
driver_booking_table.heading("Driver ID", text="Driver ID", anchor=W)
driver_booking_table.heading("Full Name", text="Full Name", anchor=W)
driver_booking_table.heading("Contact", text="Contact", anchor=W)
driver_booking_table.heading("Taxi Number", text="Taxi Number", anchor=W)
```

```
availavble_driver_control = admin_controller()
record = availavble_driver_control.booked_driver_fetcher()
```

```
for data in record:
    driver_booking_table.insert(
        "",
        index="end",
        values=[
            data[0],
            data[1],
            data[3],
            data[4],
        ],
    )
```

```
assign_button = tk.Button(assign_table_frame)
assign_button.place(relx=0.400, rely=0.880, height=35, width=120)
assign_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#FFFFFF",
    text="""Not Now""",
    state="disabled",
)
```

```
@staticmethod
def selcted_driver_row_assign(
    driver_booking_table,
    admin_controller,
    booking_details,
    assign_taxi_frame,
):
    selected_row = driver_booking_table.focus()
    driver_id = driver_booking_table.item(selected_row, "values")
    selcted_driver_id = driver_id[0]
    booking_id = booking_details[0]
```

```
assign_taxi_controll = admin_controller()
assign_taxi_controll.assign_controller(selcted_driver_id, booking_id)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
messagebox.showinfo(  
    "Sucess",  
    "Driver ID "  
    + str(seleted_driver_id)  
    + " Sucessfully assigned to booking id "  
    + str(booking_id),  
)
```

```
assign_taxi_frame.destroy()
```

- **Module : driver**

Package : driver_login.py

code :

```
import tkinter as tk  
from helper.constants import BG_LOCATION  
from .driver_model import DriverModel  
from .driver_controller import DriverController
```

```
class DriverLogInPage:  
    def __init__(self, root):  
        self.root = root  
        self.create_driver_login_frame()  
  
    def create_driver_login_frame(self):  
        self.login_frame = tk.Frame(self.root)  
        self.login_frame.configure(background="#FFFFFF")  
        self.login_frame.place(relx=-0.07, rely=0, height=768, width=1366)  
        self.build_driver_login_frame(login_frame=self.login_frame, root=self.root)
```

```
@staticmethod  
def build_driver_login_frame(login_frame, root):  
  
    welcome = tk.Label(login_frame)  
    welcome.place(relx=0.650, rely=0.149, height=59, width=200)  
    welcome.configure(  
        anchor="w",  
        background="#FFFFFF",  
        compound="left",  
        font="-family {Noto Sans} -size 22 -weight bold",  
        text="Driver Log In",  
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
username_label = tk.Label(login_frame)
username_label.place(relx=0.585, rely=0.299, height=21, width=206)
username_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 16",
    text="Username",
)
```

```
username_entry = tk.Entry(login_frame)
username_entry.place(relx=0.585, rely=0.353, height=33, relwidth=0.301)
username_entry.configure(
    background="#EFF0F2", font="TkFixedFont", selectbackground="#c4c4c4"
)
```

```
password_label = tk.Label(login_frame)
password_label.place(relx=0.585, rely=0.434, height=20, width=185)
password_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 16",
    text="Password",
)
```

```
password_entry = tk.Entry(login_frame)
password_entry.place(relx=0.585, rely=0.488, height=33, relwidth=0.301)
password_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
    show="*",
)
```

```
login_button = tk.Button(login_frame)
login_button.place(relx=0.674, rely=0.611, height=43, width=201)
login_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12 -weight bold",
    foreground="#FFFFFF",
    text="Log In",
    command=lambda: DriverLogInPage.admin_login(
        username_entry,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
    password_entry,  
    login_frame,  
    root,  
)  
)
```

```
background_label = tk.Label(login_frame)  
background_label.place(relx=-0.029, rely=0.0, height=739, width=700)  
background_label.configure(anchor="w", compound="left")  
background_image = tk.PhotoImage(file=BG_LOCATION)  
background_label.configure(image=background_image)
```

```
exit_button = tk.Button(login_frame)  
exit_button.place(relx=0.100, rely=0.865, height=43, width=111)  
exit_button.configure(  
    activebackground="beige",  
    background="#007074",  
    borderwidth="2",  
    compound="left",  
    font="-family {Noto Sans} -size 12",  
    foreground="#FFFFFF",  
    text="Back",  
    command=lambda: DriverLogInPage.exit_application(login_frame),  
)
```

```
show_pass_button = tk.Button(login_frame)  
show_pass_button.place(relx=0.900, rely=0.500, height=23, width=51)  
show_pass_button.configure(  
    activebackground="beige",  
    borderwidth="2",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    text="OO",  
)
```

```
show_pass_button.bind(  
    "<Button-1>",  
    lambda event: DriverLogInPage.show_pass(  
        event,  
        password_entry,  
)  
)
```

```
show_pass_button.bind(  
    "<ButtonRelease-1>",  
    lambda event: DriverLogInPage.hide_pass(  
        event,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        password_entry,  
    ),  
)  
)
```

```
login_frame.mainloop()
```

```
@staticmethod  
def show_pass(  
    event,  
    password_entry,  
):  
    password = password_entry.get()  
    password_entry.config(show="", text=password)
```

```
@staticmethod  
def hide_pass(  
    event,  
    password_entry,  
):  
    password_entry.config(show="*")
```

```
def exit_application(login_frame):  
    login_frame.destroy()
```

```
@staticmethod  
def admin_login(  
    username_entry,  
    password_entry,  
    login_frame,  
    root,  
):  
    driver = DriverModel(  
        username=username_entry.get(),  
        password=password_entry.get(),  
    )  
    user_control = DriverController()  
    user_control.driver_login_control(driver, login_frame, root)
```

Package : driver_dashboard.py

code :

```
from datetime import datetime, date  
import tkinter as tk  
from tkinter import NO, W, Scrollbar, ttk  
from tkinter import messagebox
```

```
from helper.constants import LOGO_LOCATION
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
from tkintermapview import TkinterMapView
```

```
class DriverDashboard:  
    def __init__(self, root, driver_controller, record):  
        self.root = root  
        self.driver_controller = driver_controller  
        self.record = record  
        self.create_dashboard_frame()  
  
    def create_dashboard_frame(self):  
        self.dashboard_frame = tk.Frame(self.root)  
        self.dashboard_frame.configure(background="#FFFFFF")  
        self.dashboard_frame.place(relx=-0.07, rely=0, height=768, width=1366)  
        self.build_dashboard_frame()  
            dashboard_frame=self.dashboard_frame,  
            root=self.root,  
            driver_controller=self.driver_controller,  
            record=self.record,  
        )  
  
    @staticmethod  
    def build_dashboard_frame(  
        dashboard_frame,  
        root,  
        driver_controller,  
        record,  
    ):  
        logo_label = tk.Label(dashboard_frame)  
        logo_label.place(relx=0.110, rely=0.068, height=71, width=80)  
        logo_label.configure(  
            anchor="w",  
            compound="left",  
            background="#FFFFFF",  
        )  
        photo_location = tk.PhotoImage(file=LOGO_LOCATION)  
        logo_label.configure(image=photo_location)  
  
        DriverDashboard.upcoming_trip_frame(  
            dashboard_frame,  
            driver_controller,  
            record,  
        )  
  
        customer_table_button = tk.Button(dashboard_frame)  
        customer_table_button.place(relx=0.110, rely=0.850, height=33, width=101)  
        customer_table_button.configure(
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
activebackground="beige",
background="#007074",
borderwidth="2",
compound="left",
font="-family {Noto Sans} -size 9 -weight bold",
foreground="#FFFFFF",
text="Upcoming",
command=lambda: DriverDashboard.upcoming_trip_frame(
    dashboard_frame,
    driver_controller,
    record,
),
)
```

```
driver_table_button = tk.Button(dashboard_frame)
driver_table_button.place(relx=0.200, rely=0.850, height=33, width=101)
driver_table_button.configure(
    activebackground="beige",
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 9 -weight bold",
    foreground="#FFFFFF",
    text="Completed",
    command=lambda: DriverDashboard.completed_trip_frame(
        dashboard_frame,
        driver_controller,
        record,
    ),
)
```

```
profile_refresh_button = tk.Button(dashboard_frame)
profile_refresh_button.place(relx=0.720, rely=0.450, height=35, width=100)
profile_refresh_button.configure(
    background="#FFFFFF",
    compound="left",
    text="Update",
    command=lambda: DriverDashboard.driver_profile_frame_create(
        dashboard_frame,
        driver_controller,
        record,
    ),
)
```

```
DriverDashboard.driver_profile_frame_create(
    dashboard_frame,
    driver_controller,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    record,
```

```
)
```

```
    dashboard_frame.mainloop()
```

```
@staticmethod
```

```
def upcoming_trip_frame(dashboard_frame, driver_controller, record):
```

```
    table_frame = tk.Frame(dashboard_frame, bg="#FFFFFF")
```

```
    table_frame.place(relx=0.110, rely=0.590, height=160, width=1160)
```

```
    DriverDashboard.upcoming_trip_table(
```

```
        table_frame,
```

```
        driver_controller,
```

```
        record,
```

```
)
```

```
@staticmethod
```

```
def upcoming_trip_table(table_frame, driver_controller, record):
```

```
    title = tk.Label(table_frame)
```

```
    title.place(relx=0, rely=0, height=40, width=212)
```

```
    title.configure(
```

```
        anchor="w",
```

```
        background="#FFFFFF",
```

```
        compound="left",
```

```
        font="-family {Noto Sans} -size 18 -weight bold",
```

```
        foreground="#3056D3",
```

```
        text="Upcoming Trip",
```

```
)
```

```
    table_style = ttk.Style()
```

```
    table_style.theme_use("default")
```

```
    table_style.configure(
```

```
        "Treeview",
```

```
        background="#FFFFFF",
```

```
        foreground="#4A4A4A",
```

```
        rowheight="35",
```

```
        fieldbackground="#FFFFFF",
```

```
)
```

```
    table_style.map("Treeview", background=[("selected", "#C9C9C9")])
```

```
    table_scroll_bar = Scrollbar(table_frame)
```

```
    table_scroll_bar.place(relx=0.985, rely=0.250, height=118, width=15)
```

```
    upcoming_booking_table = ttk.Treeview(
```

```
        table_frame,
```

```
        yscrollcommand=table_scroll_bar.set,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    selectmode="extended",
)
)
```

```
upcoming_booking_table.bind(
    "<Double-1>",
    lambda event: DriverDashboard.driver_trip_detail(
        event, table_frame, trip_fetched_data, driver_controller
    ),
)
)
```

```
upcoming_booking_table.place(
    relx=0,
    rely=0.250,
    height=118,
    width=1125,
)
)
```

```
table_scroll_bar.config(
    command=upcoming_booking_table.yview,
)
)
```

```
upcoming_booking_table["columns"] = (
    "Booking ID",
    "Full Name",
    "Date and Time",
    "Destination",
    "Total Cost",
    "Status",
)
)
```

Format Our Columns

```
upcoming_booking_table.column("#0", width=0, stretch=NO)
upcoming_booking_table.column("Booking ID", width=20, anchor=W)
upcoming_booking_table.column("Full Name", width=70, anchor=W)
upcoming_booking_table.column("Date and Time", width=70, anchor=W)
upcoming_booking_table.column("Destination", width=240, anchor=W)
upcoming_booking_table.column("Total Cost", width=40, anchor=W)
upcoming_booking_table.column("Status", width=40, anchor=W)
```

Create Heading

```
upcoming_booking_table.heading("#0", text="", anchor=W)
upcoming_booking_table.heading("Booking ID", text="Booking ID", anchor=W)
upcoming_booking_table.heading("Full Name", text="Full Name", anchor=W)
upcoming_booking_table.heading("Date and Time", text="Date and Time", anchor=W)
upcoming_booking_table.heading("Destination", text="Destination", anchor=W)
upcoming_booking_table.heading("Total Cost", text="Total Cost", anchor=W)
upcoming_booking_table.heading("Status", text="Status", anchor=W)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
for data in record:  
    driver_id = data[0]
```

```
    upcoming_trip_control = driver_controller()  
    trip_fetched_data = upcoming_trip_control.upcoming_trip_detail_fetcher(  
        driver_id  
    )
```

```
    for data in trip_fetched_data:
```

```
        upcoming_booking_table.insert(  
            "",  
            index="end",  
            values=  
                [data[0],  
                 (data[2], data[3]),  
                 (data[6], "---", data[7], ":", data[8]),  
                 data[10],  
                 data[11],  
                 data[16],  
                ],  
            )
```

```
@staticmethod  
def completed_trip_frame(dashboard_frame, driver_controller, record):  
    table_frame = tk.Frame(dashboard_frame, bg="#FFFFFF")  
    table_frame.place(relx=0.110, rely=0.590, height=160, width=1160)
```

```
DriverDashboard.completed_trip_table(  
    table_frame,  
    driver_controller,  
    record,  
)
```

```
@staticmethod  
def driver_trip_detail(  
    event,  
    table_frame,  
    trip_fetched_data,  
    driver_controller,  
):
```

```
    for data in trip_fetched_data:
```

```
        driver_trip_frame = tk.Toplevel(table_frame)  
        driver_trip_frame.title("Assign Taxi")
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
driver_trip_frame.geometry("682x641+193+31")
driver_trip_frame.resizable(0, 0)
driver_trip_frame.configure(background="#FFFFFF")
```

```
title = tk.Label(driver_trip_frame)
title.place(relx=0.360, rely=0.031, height=43, width=250)
title.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 18 -weight bold",
    text="""Booking Details""",
)
```

```
fullname_label = tk.Label(driver_trip_frame)
fullname_label.place(relx=0.029, rely=0.156, height=34, width=122)
fullname_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""Full Name : """,
)
```

```
fullname_data = tk.Label(driver_trip_frame)
fullname_data.place(relx=0.22, rely=0.156, height=34, width=280)
fullname_data.configure(
    anchor="w",
    background="#FFFFFF",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[2] + " " + data[3],
)
```

```
map_view_frame = tk.Frame(driver_trip_frame)
map_view_frame.place(relx=0.640, rely=0.156, height=140, width=220)
map_view_frame.configure(
    background="#FFFFFF",
    borderwidth="2",
    highlightthickness="2",
)
```

```
location_detail_map_view = TkinterMapView(map_view_frame)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
location_detail_map_view.set_tile_server(  
    "https://mt0.google.com/vt/lyrs=m&hl=en&x={x}&y={y}&z={z}&s=Ga",  
    max_zoom=60,  
)
```

```
pickup_cordinates = eval(data[19])  
destination_coordinates = eval(data[20])
```

```
location_detail_map_view.set_marker(  
    pickup_cordinates[0],  
    pickup_cordinates[1],  
)
```

```
location_detail_map_view.set_marker(  
    destination_coordinates[0],  
    destination_coordinates[1],  
)
```

```
location_detail_map_view.set_position(  
    pickup_cordinates[0],  
    pickup_cordinates[1],  
)
```

```
location_detail_map_view.set_path(  
    position_list=[pickup_cordinates, destination_coordinates]  
)
```

```
location_detail_map_view.pack()
```

```
passenger_no_label = tk.Label(driver_trip_frame)  
passenger_no_label.place(relx=0.029, rely=0.234, height=34, width=175)  
passenger_no_label.configure(  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    text="No of Passenger :")
```

```
passenger_no_data = tk.Label(driver_trip_frame)  
passenger_no_data.place(relx=0.293, rely=0.234, height=34, width=48)  
passenger_no_data.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    foreground="#4A4A4A",  
    text="")
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[4],
)
```

```
taxi_number = tk.Label(driver_trip_frame)
taxi_number.place(relx=0.029, rely=0.312, height=35, width=111)
taxi_number.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""No of Taxi :"",
)
```

```
taxi_number_data = tk.Label(driver_trip_frame)
taxi_number_data.place(relx=0.22, rely=0.312, height=35, width=48)
taxi_number_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[5],
)
```

```
pick_up_date_lable = tk.Label(driver_trip_frame)
pick_up_date_lable.place(relx=0.029, rely=0.39, height=34, width=133)
pick_up_date_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""Pick Up Date :"",
)
```

```
pick_up_date_data = tk.Label(driver_trip_frame)
pick_up_date_data.place(relx=0.249, rely=0.39, height=35, width=163)
pick_up_date_data.configure(
    activebackground="#f9f9f9",
    anchor="w",

```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
background="#FFFFFF",
compound="left",
font="-family {Noto Sans} -size 14",
foreground="#4A4A4A",
highlightthickness="2",
text=data[6],
)
```

```
pick_up_address_label = tk.Label(driver_trip_frame)
pick_up_address_label.place(relx=0.029, rely=0.468, height=35, width=175)
pick_up_address_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""Pick Up Address :""",
)
time_label = tk.Label(driver_trip_frame)
time_label.place(relx=0.513, rely=0.39, height=34, width=69)
time_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""Time :""",
)
```

```
data7 = str(data[7])
data8 = str(data[8])
time_data = tk.Label(driver_trip_frame)
time_data.place(relx=0.616, rely=0.39, height=34, width=100)
time_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=(data7 + ":" + data8),
)
```

```
pickup_address_data = tk.Label(driver_trip_frame)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
pickup_address.place(relx=0.293, rely=0.468, height=35, width=323)
pickup_address.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 10",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[9],
)
```

```
destination_label = tk.Label(driver_trip_frame)
destination_label.place(relx=0.029, rely=0.546, height=34, width=132)
```

```
destination_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""Destination :"""",
)
```

```
destination_address_lable = tk.Label(driver_trip_frame)
destination_address_lable.place(
    relx=0.249, rely=0.546, height=34, width=323
)
destination_address_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 10",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[10],
)
```

```
total_cost_label = tk.Label(driver_trip_frame)
total_cost_label.place(relx=0.029, rely=0.624, height=34, width=111)
total_cost_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""Total Cost :""",
)
```

```
total_cost_data = tk.Label(driver_trip_frame)
total_cost_data.place(relx=0.22, rely=0.624, height=34, width=206)
total_cost_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[11] + " .NPR",
)
```

```
payment_method_lable = tk.Label(driver_trip_frame)
payment_method_lable.place(relx=0.029, rely=0.702, height=33, width=185)
payment_method_lable.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="""Payment Method :""",
)
```

```
payment_method_data = tk.Label(driver_trip_frame)
payment_method_data.place(relx=0.293, rely=0.702, height=34, width=206)
payment_method_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    highlightthickness="2",
    text=data[12],
)
```

```
trip_completed_button = tk.Button(driver_trip_frame)
trip_completed_button.place(relx=0.365, rely=0.880, height=33, width=160)
trip_completed_button.configure(
    activebackground="beige",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
background="#007074",
borderwidth="2",
compound="left",
font="-family {Noto Sans} -size 10 -weight bold",
foreground="#FFFFFF",
text="""Trip Completed""",
command=lambda: DriverDashboard.trip_completed(
    trip_fetched_data,
    driver_controller,
    driver_trip_frame,
),
)
```

Comparing today date and trip date

```
trip_date = data[6]
trip_time = str(data[7]) + ":" + str(data[8]) + ":00"
time_final = datetime.strptime(trip_time, "%H:%M:%S").time()
```

```
trip_date_and_time = datetime.combine(trip_date, time_final)
today = datetime.now().isoformat(" ", "seconds")
today_date = datetime.strptime(today, "%Y-%m-%d %H:%M:%S")
```

```
if today_date < trip_date_and_time:
    trip_completed_button.config(state="disabled")
    trip_completed_button.update()
else:
    trip_completed_button.config(state="normal")
    trip_completed_button.update()
```

```
if data[16] == "Completed":
    trip_completed_button.config(state="disabled")
    trip_completed_button.update()
```

```
back_button = tk.Button(driver_trip_frame)
back_button.place(relx=0.865, rely=0.031, height=33, width=71)
back_button.configure(
    activebackground="beige",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 10",
    foreground="#FFFFFF",
    text="""Back""",
```

```
driver_trip_frame.wait_visibility()
driver_trip_frame.grab_set()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
driver_trip_frame.mainloop()
```

```
@staticmethod
def trip_completed(trip_fetched_data, driver_controller, driver_trip_frame):
    for data in trip_fetched_data:
        booking_id = data[0]
        driver_id = data[22]
```

```
trip_complete_control = driver_controller()
trip_complete_control.completed_trip_controller(booking_id, driver_id)
messagebox.showinfo("Sucess", "Marked as completed ")
driver_trip_frame.destroy()
```

```
@staticmethod
def completed_trip_table(table_frame, driver_controller, record):
    title = tk.Label(table_frame)
    title.place(relx=0, rely=0, height=40, width=212)
    title.configure(
        anchor="w",
        background="#FFFFFF",
        compound="left",
        font="-family {Noto Sans} -size 18 -weight bold",
        foreground="#3056D3",
        text="Completed Trip",
    )
)
```

```
table_style = ttk.Style()
table_style.theme_use("default")
table_style.configure(
    "Treeview",
    background="#FFFFFF",
    foreground="#4A4A4A",
    rowheight="35",
    fieldbackground="#FFFFFF",
)
table_style.map("Treeview", background=[("selected", "#C9C9C9")])
```

```
table_scroll_bar = Scrollbar(table_frame)
table_scroll_bar.place(relx=0.985, rely=0.250, height=118, width=15)
```

```
completed_booking_table = ttk.Treeview(
    table_frame,
    yscrollcommand=table_scroll_bar.set,
    selectmode="extended",
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
completed_booking_table.bind(  
    "<Double-1>",  
    lambda event: DriverDashboard.driver_trip_detail(  
        event, table_frame, trip_fetched_data, driver_controller  
    ),  
)  
  
completed_booking_table.place(  
    relx=0,  
    rely=0.250,  
    height=118,  
    width=1125,  
)
```

```
table_scroll_bar.config(  
    command=completed_booking_table.yview,  
)
```

```
completed_booking_table["columns"] = (  
    "Booking ID",  
    "Full Name",  
    "Date and Time",  
    "Destination",  
    "Total Cost",  
    "Status",  
)
```

Format Our Columns

```
completed_booking_table.column("#0", width=0, stretch=NO)  
completed_booking_table.column("Booking ID", width=20, anchor=W)  
completed_booking_table.column("Full Name", width=70, anchor=W)  
completed_booking_table.column("Date and Time", width=70, anchor=W)  
completed_booking_table.column("Destination", width=240, anchor=W)  
completed_booking_table.column("Total Cost", width=40, anchor=W)  
completed_booking_table.column("Status", width=40, anchor=W)
```

Create Heading

```
completed_booking_table.heading("#0", text="", anchor=W)  
completed_booking_table.heading("Booking ID", text="Booking ID", anchor=W)  
completed_booking_table.heading("Full Name", text="Full Name", anchor=W)  
completed_booking_table.heading("Date and Time", text="Date and Time", anchor=W)  
completed_booking_table.heading("Destination", text="Destination", anchor=W)  
completed_booking_table.heading("Total Cost", text="Total Cost", anchor=W)  
completed_booking_table.heading("Status", text="Status", anchor=W)
```

```
for data in record:  
    driver_id = data[0]
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
completed_trip_control = driver_controller()
trip_fetched_data = completed_trip_control.completed_trip_detail_fetcher(
    driver_id
)
```

```
for data in trip_fetched_data:
```

```
    completed_booking_table.insert(
        "",
        index="end",
        values=(
            data[0],
            (data[2], data[3]),
            (data[6], "---", data[7], ":", data[8]),
            data[10],
            data[11],
            data[16],
        ),
    )
```

```
@staticmethod
def driver_profile_frame_create(
    dashboard_frame,
    driver_controller,
    record,
):
    profile_details_label = tk.Frame(dashboard_frame)
    profile_details_label.place(relx=0.720, rely=0.068, height=280, width=325)
    profile_details_label.configure(
        background="#FFFFFF",
        highlightthickness="2",
        relief="flat",
    )
```

```
for data in record:
    driver_id = data[0]
```

```
driver_profile_data_control = driver_controller()
profile_details = driver_profile_data_control.driver_profile_data_fetcher(
    driver_id
)
```

```
driver_profile_data_control = driver_controller()
booking_details = driver_profile_data_control.driver_booking_data_fetcher(
    driver_id
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
profile_data = profile_details[0]
booking_data = booking_details[0]
```

```
username_data = tk.Label(profile_details_label)
username_data.place(relx=0.215, rely=0.036, height=41, width=184)
username_data.configure(
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14 -weight bold",
    foreground="#4A4A4A",
    text=profile_data[6],
)
```

```
name_label = tk.Label(profile_details_label)
name_label.place(relx=0.062, rely=0.25, height=31, width=64)
name_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#4A4A4A",
    text="Name :",
)
```

```
name_data = tk.Label(profile_details_label)
name_data.place(relx=0.277, rely=0.25, height=31, width=184)
name_data.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 11",
    foreground="#4A4A4A",
    text=profile_data[1],
)
```

```
contact_label = tk.Label(profile_details_label)
contact_label.place(relx=0.062, rely=0.357, height=31, width=74)
contact_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#4A4A4A",
    text="Contact :",
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
contact_data = tk.Label(profile_details_label)
contact_data.place(relx=0.308, rely=0.357, height=31, width=104)
contact_data.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 11",
    foreground="#4A4A4A",
    text=profile_data[3],
)
```

```
taxi_label = tk.Label(profile_details_label)
taxi_label.place(relx=0.062, rely=0.464, height=31, width=54)
taxi_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#4A4A4A",
    text="Taxi :",
)
taxi_data = tk.Label(profile_details_label)
taxi_data.place(relx=0.246, rely=0.464, height=31, width=184)
taxi_data.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 11",
    foreground="#4A4A4A",
    text=(profile_data[9] + " " + profile_data[10]),
)
```

```
taxi_number_label = tk.Label(profile_details_label)
taxi_number_label.place(relx=0.062, rely=0.571, height=31, width=114)
taxi_number_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#4A4A4A",
    text="Taxi number :",
)
```

```
taxi_number_data = tk.Label(profile_details_label)
taxi_number_data.place(relx=0.431, rely=0.571, height=31, width=134)
taxi_number_data.configure(
    anchor="w",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
        background="#FFFFFF",
        compound="left",
        font="-family {Noto Sans} -size 11",
        foreground="#4A4A4A",
        text=profile_data[11],
    )
```

```
total_revenue_label = tk.Label(profile_details_label)
total_revenue_label.place(relx=0.062, rely=0.714, height=41, width=154)
total_revenue_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="Total Revenue :",
)
```

```
total_revenue_data = tk.Label(profile_details_label)
total_revenue_data.place(relx=0.031, rely=0.857, height=31, width=304)
total_revenue_data.configure(
    anchor="center",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    foreground="#4A4A4A",
    text="XXXXX",
)
```

```
data_view_button = tk.Button(profile_details_label)
data_view_button.place(relx=0.554, rely=0.75, height=23, width=41)
data_view_button.configure(
    background="#FFFFFF",
    compound="left",
    text="00",
)
```

```
data_view_button.bind(
    "<Button-1>",
    lambda event: DriverDashboard.show_revenue(
        event,
        total_revenue_data,
        booking_data,
    ),
)
```

```
data_view_button.bind(
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
"<ButtonRelease-1>",
lambda event: DriverDashboard.hide_revenue(
    event,
    total_revenue_data,
),
)
```

```
DriverDashboard.driver_payment_frame_create(
    dashboard_frame,
    record,
    driver_controller,
)
```

```
@staticmethod
def show_revenue(event, total_revenue_data, booking_data):
    total_revenue_data.config(text=booking_data)
    total_revenue_data.update()
```

```
@staticmethod
def hide_revenue(
    event,
    total_revenue_data,
):
    total_revenue_data.config(text="XXXXX")
```

```
@staticmethod
def driver_payment_frame_create(
    dashboard_frame,
    record,
    driver_controller,
):
```

```
    today = date.today().strftime("%Y-%m")
```

```
    for data in record:
        driver_id = data[0]
```

```
        driver_profile_data_control = driver_controller()
        monthly_revenue_details = (
            driver_profile_data_control.driver_monthly_data_fetcher(
                driver_id,
                today,
            )
        )
```

```
        monthly_income = monthly_revenue_details[0]
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
profile_details_label = tk.Frame(dashboard_frame)
profile_details_label.place(relx=0.470, rely=0.250, height=140, width=325)
profile_details_label.configure(
    background="#FFFFFF",
    highlightthickness="2",
    relief="flat",
)
```

```
payment_label = tk.Label(profile_details_label)
payment_label.place(relx=0.215, rely=0.036, height=41, width=184)
payment_label.configure(
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14 -weight bold",
    foreground="#4A4A4A",
    text="Monthly Revenue",
)
```

```
gross_income_label = tk.Label(profile_details_label)
gross_income_label.place(relx=0.062, rely=0.320, height=31, width=120)
gross_income_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#4A4A4A",
    text="Gross Income :",
)
```

```
gross_income_data = tk.Label(profile_details_label)
gross_income_data.place(relx=0.445, rely=0.320, height=31, width=160)
gross_income_data.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 11",
    foreground="#4A4A4A",
    text="XXXXXX",
)
```

```
service_cost_label = tk.Label(profile_details_label)
service_cost_label.place(relx=0.062, rely=0.520, height=31, width=110)
service_cost_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
foreground="#4A4A4A",
text="Service Cost :",
)
service_cost_data = tk.Label(profile_details_label)
service_cost_data.place(relx=0.410, rely=0.520, height=31, width=160)
service_cost_data.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 11",
    foreground="#4A4A4A",
    text="XXXXX",
)

net_income_label = tk.Label(profile_details_label)
net_income_label.place(relx=0.062, rely=0.720, height=31, width=100)
net_income_label.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#4A4A4A",
    text="Net Income :",
)
net_income_data = tk.Label(profile_details_label)
net_income_data.place(relx=0.380, rely=0.720, height=31, width=160)
net_income_data.configure(
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 11",
    foreground="#4A4A4A",
    text="XXXXX",
)

data_view_button = tk.Button(profile_details_label)
data_view_button.place(relx=0.850, rely=0.055, height=23, width=41)
data_view_button.configure(
    background="#FFFFFF",
    compound="left",
    text="00",
)
data_view_button.bind(
    "<Button-1>",
    lambda event: DriverDashboard.show_monthly_revenue(
        event,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
gross_income_data,  
service_cost_data,  
net_income_data,  
monthly_income,  
),  
)
```

```
data_view_button.bind(  
    "<ButtonRelease-1>",  
    lambda event: DriverDashboard.hide_monthly_revenue(  
        event,  
        gross_income_data,  
        service_cost_data,  
        net_income_data,  
    ),  
)
```

```
@staticmethod  
def show_monthly_revenue(  
    event,  
    gross_income_data,  
    service_cost_data,  
    net_income_data,  
    monthly_income,  
):
```

```
gross_income = str(round(monthly_income[0], 2))  
service_cost = str(round(monthly_income[1], 2))  
net_income = str(round(monthly_income[2], 2))
```

```
gross_income_data.config(text=gross_income)  
gross_income_data.update()  
service_cost_data.config(text=service_cost)  
service_cost_data.update()  
net_income_data.config(text=net_income)  
net_income_data.update()
```

```
@staticmethod  
def hide_monthly_revenue(  
    event,  
    gross_income_data,  
    service_cost_data,  
    net_income_data,  
):  
    gross_income_data.config(text="XXXXXX")  
    gross_income_data.update()  
    service_cost_data.config(text="XXXXXX")
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
service_cost_data.update()  
net_income_data.config(text="XXXXXX")  
net_income_data.update()
```

Package : driver_register.py

code :

```
import tkinter as tk  
from tkinter import ttk  
from tkinter import messagebox  
from helper.exceptions import CustomException  
from taxi.taxi_register_controller import TaxiController  
from .driver_model import DriverModel  
from .driver_controller import DriverController
```

```
class DriverRegisterPage:  
    def __init__(self, root, control_panel_frame):  
        self.root = root  
        self.control_panel_frame = control_panel_frame  
        self.create_driver_register_frame()
```

```
def create_driver_register_frame(self):  
    self.driver_register_frame = tk.Toplevel(self.root)  
    self.driver_register_frame.title("Driver Register")  
    self.driver_register_frame.configure(background="#FFFFFF")  
    self.driver_register_frame.resizable(0, 0)  
    self.driver_register_frame.geometry("460x579+420+91")  
    self.build_driver_register_frame(  
        driver_register_frame=self.driver_register_frame,  
    )
```

```
@staticmethod  
def build_driver_register_frame(driver_register_frame):
```

```
    welcome = tk.Label(driver_register_frame)  
    welcome.place(relx=0.313, rely=0.060, height=41, width=250)  
    welcome.configure(  
        activebackground="#f9f9f9",  
        anchor="w",  
        background="#FFFFFF",  
        compound="left",  
        text="Driver Register",  
        font="-family {Noto Sans} -size 16 -weight bold",  
    )
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

```
fullname = tk.Label(driver_register_frame)
fullname.place(relx=0.065, rely=0.19, height=41, width=104)
fullname.configure(
    activebackground="#f9f9f9",
    anchor="w",
    font="-family {Noto Sans} -size 14",
    compound="left",
    background="#FFFFFF",
    text="Full Name",
)
```

```
fullname_entry = tk.Entry(driver_register_frame)
fullname_entry.place(
    relx=0.065,
    rely=0.259,
    height=23,
    relwidth=0.404,
)
fullname_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
)
```

```
license_label = tk.Label(driver_register_frame)
license_label.place(relx=0.543, rely=0.19, height=41, width=200)
license_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    text="License Number",
)
```

```
license_entry = tk.Entry(driver_register_frame)
license_entry.place(
    relx=0.543,
    rely=0.259,
    height=23,
    relwidth=0.404,
)
license_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
contact = tk.Label(driver_register_frame)
contact.place(relx=0.065, rely=0.311, height=41, width=104)
contact.configure(
    activebackground="#f9f9f9",
    anchor="w",
    compound="left",
    background="#FFFFFF",
    font="-family {Noto Sans} -size 14",
    text="\"Contact\"",
)
```

```
contact_entry = tk.Entry(driver_register_frame)
contact_entry.place(
    relx=0.065,
    rely=0.38,
    height=23,
    relwidth=0.404,
)
contact_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
)
```

```
taxi_number_label = tk.Label(driver_register_frame)
taxi_number_label.place(
    relx=0.543,
    rely=0.311,
    height=41,
    width=200,
)
taxi_number_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    text="\"Taxi Number\"",
)
```

```
selected_taxi_number = tk.StringVar()
```

```
taxi_number_entry = ttk.Combobox(
    driver_register_frame,
    textvariable=selected_taxi_number,
    state="readonly",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    values=("taxi_number"),
)
taxi_number_entry.place(
    relx=0.543,
    rely=0.38,
    height=23,
    relwidth=0.404,
)
taxi_number_entry.bind(
    "<<ComboboxSelected>>",
    lambda event: DriverRegisterPage.taxi_number_fetcher(
        event,
        driver_register_frame,
        taxi_number_entry,
        selected_taxi_number,
    ),
)
```

```
username = tk.Label(driver_register_frame)
username.place(relx=0.065, rely=0.430, height=41, width=104)
username.configure(
    activebackground="#f9f9f9",
    anchor="w",
    compound="left",
    background="#FFFFFF",
    font="-family {Noto Sans} -size 14",
    text="username",
```

```
username_entry = tk.Entry(driver_register_frame)
username_entry.place(
    relx=0.065,
    rely=0.500,
    height=23,
    relwidth=0.404,
)
username_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
```

```
password = tk.Label(driver_register_frame)
password.place(relx=0.543, rely=0.430, height=41, width=104)
password.configure(
    activebackground="#f9f9f9",
    anchor="w",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        compound="left",
        background="#FFFFFF",
        font="-family {Noto Sans} -size 14",
        text=""password""",
    )
```

```
password_entry = tk.Entry(driver_register_frame)
password_entry.place(
    relx=0.543,
    rely=0.500,
    height=23,
    relwidth=0.404,
)
password_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
    show="*",
)
```

```
update_button = tk.Button(driver_register_frame)
update_button.place(relx=0.522, rely=0.864, height=43, width=111)
update_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12 -weight bold",
    foreground="#FFFFFF",
    text=""Register""",
    command=lambda: DriverRegisterPage.driver_registration(
        driver_register_frame,
        fullname_entry,
        license_entry,
        contact_entry,
        selected_taxi_number,
        username_entry,
        password_entry,
    ),
)
```

```
cancel_button = tk.Button(driver_register_frame)
cancel_button.place(relx=0.250, rely=0.864, height=43, width=111)
cancel_button.configure(
    activebackground="beige",
    background="#007074",
    borderwidth="2",
    compound="left",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
font="-family {Noto Sans} -size 12",
foreground="#FFFFFF",
text="Cancel",
command=lambda: DriverRegisterPage.cancel_button(
    driver_register_frame,
),
)
driver_register_frame.grab_set()
driver_register_frame.mainloop()
```

```
@staticmethod
def taxi_number_fetcher(
    event,
    driver_register_frame,
    taxi_number_entry,
    selected_taxi_number,
):
    TaxiController.taxi_number_fetcher(taxi_number_entry)
```

```
DriverRegisterPage.taxi_detail_viewer(
    driver_register_frame,
    selected_taxi_number,
)
```

```
@staticmethod
def taxi_detail_viewer(
    driver_register_frame,
    selected_taxi_number,
):
    driver_register_frame = tk.Frame(driver_register_frame)
    driver_register_frame.place(
        relx=0.065,
        rely=0.570,
        height=150,
        relwidth=0.883,
    )
    driver_register_frame.config(
        background="#FFFFFF",
        borderwidth=2,
        relief="solid",
        highlightbackground="#000000",
    )
```

```
welcome = tk.Label(driver_register_frame)
welcome.place(relx=0.065, rely=0.10, height=41, width=365)
welcome.configure(
    background="#FFFFFF",
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
compound="left",
text="Driver Register",
font="-family {Noto Sans} -size 14 -weight bold",
anchor="center",
)
```

```
taxi_number = tk.Label(driver_register_frame)
taxi_number.place(relx=0.065, rely=0.35, height=41, width=110)
taxi_number.configure(
    activebackground="#f9f9f9",
    anchor="w",
    font="-family {Noto Sans} -size 12",
    compound="left",
    background="#FFFFFF",
    text="Taxi Number :",
)
```

```
taxi_number_data = tk.Label(driver_register_frame)
taxi_number_data.place(relx=0.360, rely=0.35, height=41, width=110)
taxi_number_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    font="-family {Noto Sans} -size 12",
    compound="left",
    background="#FFFFFF",
)
```

```
taxi_age = tk.Label(driver_register_frame)
taxi_age.place(relx=0.680, rely=0.35, height=41, width=75)
taxi_age.configure(
    activebackground="#f9f9f9",
    anchor="w",
    font="-family {Noto Sans} -size 12",
    compound="left",
    background="#FFFFFF",
    text="Taxi Age :",
)
```

```
taxi_age_data = tk.Label(driver_register_frame)
taxi_age_data.place(relx=0.880, rely=0.35, height=41, width=40)
taxi_age_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    font="-family {Noto Sans} -size 12",
    compound="left",
    background="#FFFFFF",
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
taxi_descripion = tk.Label(driver_register_frame)
taxi_descripion.place(relx=0.065, rely=0.65, height=41, width=100)
taxi_descripion.configure(
    activebackground="#f9f9f9",
    anchor="w",
    font="-family {Noto Sans} -size 12",
    compound="left",
    background="#FFFFFF",
    text="\"Description :\"",

)
taxi_descripion_data = tk.Label(driver_register_frame)
taxi_descripion_data.place(relx=0.360, rely=0.70, height=41, width=250)
taxi_descripion_data.configure(
    activebackground="#f9f9f9",
    anchor="w",
    font="-family {Noto Sans} -size 12",
    compound="left",
    background="#FFFFFF",
)

```

```
DriverRegisterPage.taxi_data_fetcher(
    welcome,
    taxi_number_data,
    taxi_age_data,
    taxi_descripion_data,
    selected_taxi_number,
)
```

```
@staticmethod
def taxi_data_fetcher(
    welcome,
    taxi_number_data,
    taxi_age_data,
    taxi_descripion_data,
    selected_taxi_number,
):
    TaxiController.taxi_data_fetcher(
        welcome,
        taxi_number_data,
        taxi_age_data,
        taxi_descripion_data,
        selected_taxi_number,
    )
```

```
@staticmethod
def cancel_button(driver_register_frame):
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
driver_register_frame.destroy()
```

```
@staticmethod
def driver_registration(
    driver_register_frame,
    fullname_entry,
    license_entry,
    contact_entry,
    selected_taxi_number,
    username_entry,
    password_entry,
):
    try:
        driver = DriverModel(
            fullname=fullname_entry.get(),
            license_number=license_entry.get(),
            contact=contact_entry.get(),
            taxi_number=selected_taxi_number.get(),
            username=username_entry.get(),
            password=password_entry.get(),
        )
        driver_control = DriverController()
        driver_control.driver_registration_control(
            driver,
            driver_register_frame,
        )
    except CustomException as error:
        messagebox.showerror("invalid", error, parent=driver_register_frame)
```

Package : driver_model.py

code :

```
from pydantic import BaseModel, validator
from helper.exceptions import CustomException
import re
```

```
name_reg = re.compile(
    "^[A-Z][a-zA-Z]{3,}(?: [A-Z][a-zA-Z]*){0,2}$",
) # fullname regex
contact_reg = re.compile("^\\d{10}$") # contact regex

username_reg = re.compile("[a-zA-Z][a-zA-Z]*([._-][a-zA-Z0-9]+){0,3}$") # username regex
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

class DriverModel(BaseModel):

```
fullname: str = ""  
license_number: str = ""  
contact: str = ""  
taxi_number: str = ""  
username: str = ""  
password: str = ""
```

```
@validator("fullname")  
def fullname_validator(cls, value):  
    if not value:  
        raise CustomException("please enter your fullname !!")  
    if not name_reg.match(value):  
        raise CustomException("invalid firname format !!")  
    return value
```

```
@validator("license_number")  
def validate_license_number(cls, value):  
    if not value:  
        raise CustomException("license number cannot be empty")  
    return value
```

```
@validator("contact")  
def validate_contact(cls, value):  
    if not value:  
        raise CustomException("contact cannot be empty")  
    if not contact_reg.match(value):  
        raise CustomException("invalid format for contact")  
    return value
```

```
@validator("taxi_number")  
def validate_taxi_number(cls, value):  
    if not value:  
        raise CustomException("taxi number cannot be empty !!")  
    if value == "taxi_number":  
        raise CustomException("please select taxi number!!")  
    return value
```

```
@validator("username")  
def validate_username(cls, value):  
    if not value:  
        raise CustomException("username cannot be empty")  
    if not username_reg.match(value):  
        raise CustomException("invalid format for username")  
    return value
```

Package : driver_controller.py

code :

```
from tkinter import messagebox
from helper.turbo_db import Turbo_db
from .driver_dashboard import DriverDashboard

class DriverController:
    def __init__(self) -> None:
        self._connection = Turbo_db.turbo_connection()

    def driver_registration_control(
        self,
        driver,
        driver_register_frame,
    ):
        if self.authenticate(driver):
            self.registration_sucess(driver_register_frame)

    def authenticate(self, driver):
        try:
            cursor = self._connection.cursor()
            statement = """CREATE TABLE IF NOT EXISTS drivers(
                driverID SERIAL PRIMARY KEY,
                fullname VARCHAR(50) NOT NULL,
                license_number VARCHAR(50) NOT NULL UNIQUE,
                contact VARCHAR(20) NOT NULL,
                taxi_number VARCHAR(50) NOT NULL,
                driver_status VARCHAR(50) NOT NULL,
                username VARCHAR(50) NOT NULL UNIQUE,
                password VARCHAR(50) NOT NULL
            );
        """;"""

            data_insert = """INSERT INTO drivers(fullname, license_number,
            contact, taxi_number, driver_status, username, password) VALUES
            (%s, %s, %s, %s, %s, %s, %s);"""
            data_values = (
                driver.fullname,
                driver.license_number,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
    driver.contact,  
    driver.taxi_number,  
    "Available",  
    driver.username,  
    driver.password,  
)
```

```
    cursor.execute(statement)  
    cursor.execute(data_insert, data_values)
```

```
taxi_statement = (  
    """UPDATE taxi SET status = 'Assigned' WHERE taxi_number = %s;"""  
)  
data_taxi = (driver.taxi_number,)
```

```
    cursor.execute(taxi_statement, data_taxi)  
    self._connection.commit()
```

```
return True
```

```
except Exception as e:  
    print(e)  
finally:  
    if cursor is not None:  
        cursor.close()  
    if self._connection is not None:  
        self._connection.close()
```

```
def registration_sucess(self, driver_register_frame):  
    driver_register_frame.destroy()
```

```
def driver_login_control(self, driver, login_frame, root):  
    if self.login_authenticate(driver):  
        self.login_sucess(root, login_frame)  
    else:  
        messagebox.showerror(  
            "Invalid Data",  
            "username or password not matched",  
        )
```

```
def login_authenticate(self, admin):  
    try:  
        cursor = self._connection.cursor()  
        statement = "SELECT * FROM drivers WHERE username=%s AND password = %s;"  
        data = (admin.username, admin.password)  
        cursor.execute(statement, data)  
        self.record = cursor.fetchall()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
if self.record:  
    return True  
return False  
except Exception as error:  
    print(error)
```

```
def login_sucess(self, root, login_page):  
    messagebox.showinfo(title="Congratulation", message="Welcome ")  
    login_page.destroy()  
    DriverDashboard(root, DriverController, self.record)
```

```
def upcoming_trip_detail_fetcher(self, driver_id):  
    try:  
        cursor = self._connection.cursor()  
        statement = "SELECT * FROM booking as b LEFT JOIN drivers as d on b.driver_id = d.driverid  
LEFT JOIN taxi as t on d.taxi_number = t.taxi_number WHERE d.driverid = %s AND b.booking_status =  
'Accepted';"  
        did = str(driver_id)  
        data = did  
        cursor.execute(statement, data)  
        self.record = cursor.fetchall()  
        return self.record  
    except Exception as error:  
        print(error)
```

```
def completed_trip_detail_fetcher(self, driver_id):  
    try:  
        cursor = self._connection.cursor()  
        statement = "SELECT * FROM booking as b LEFT JOIN drivers as d on b.driver_id = d.driverid  
LEFT JOIN taxi as t on d.taxi_number = t.taxi_number WHERE d.driverid = %s AND b.booking_status =  
'Completed';"  
        did = str(driver_id)  
        data = did  
        cursor.execute(statement, data)  
        self.record = cursor.fetchall()  
        return self.record  
    except Exception as error:  
        print(error)
```

```
def completed_trip_controller(self, booking_id, driver_id):  
    try:  
        cursor = self._connection.cursor()  
        booking_statement = (  
            "Update booking set booking_status = 'Completed' Where booking_id = %s;"  
        )  
        bid = str(booking_id)  
        did = str(driver_id)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
booking_data = bid
cursor.execute(booking_statement, booking_data)
```

```
driver_statement = (
    "Update drivers set driver_status = 'Available' Where driverid = %s;"
)
driver_data = did
cursor.execute(driver_statement, driver_data)
```

```
self._connection.commit()
except Exception as error:
    print(error)
```

```
def driver_booking_data_fetcher(self, driver_id):
    try:
        cursor = self._connection.cursor()
        statement = "select sum(CAST(total_cost AS decimal)) from booking as b join drivers as d on
b.driver_id = d.driverid where b.booking_status = 'Completed' AND d.driverid = %s; "
        did = str(driver_id)
        data = did
        cursor.execute(statement, data)
        self.record = cursor.fetchall()
        return self.record
    except Exception as error:
        print(error)
```

```
def driver_profile_data_fetcher(self, driver_id):
    try:
        cursor = self._connection.cursor()
        statement = "select * from drivers as d join taxi as t on d.taxi_number = t.taxi_number where
d.driverid = %s; "
        did = str(driver_id)
        data = did
        cursor.execute(statement, data)
        self.record = cursor.fetchall()
        return self.record
    except Exception as error:
        print(error)
```

```
def driver_monthly_data_fetcher(self, driver_id, today):
    try:
        cursor = self._connection.cursor()
        statement = "SELECT sum(CAST(total_cost AS decimal)) as gross_income,
sum(CAST(total_cost AS decimal)/3) as service_cost, sum(CAST(total_cost AS decimal)-
CAST(total_cost AS decimal)/3) as net_income from booking as b join drivers as d on b.driver_id =
d.driverid where b.booking_status = 'Completed' AND d.driverid = %s AND CAST(b.pickup_date AS
VARCHAR(50)) like concat('%%',%s,'%%');"
        did = str(driver_id)
```

```
td = str(today)
data = (did, td)
```

```
cursor.execute(statement, data)
self.record = cursor.fetchall()
return self.record
except Exception as error:
    print(error)
```

- **Module : taxi**

Package : taxi_model.py

code :

```
from pydantic import BaseModel, validator
from helper.exceptions import CustomException
```

```
class TaxiModel(BaseModel):
```

```
brand: str = ""
model: str = ""
taxi_number: str = ""
taxi_age: str = ""
discription: str = ""
```

```
@validator("brand")
def validate_brand(cls, value):
    if not value:
        raise CustomException("brand cannot be empty")
    return value
```

```
@validator("model")
def validate_model(cls, value):
    if not value:
        raise CustomException("model cannot be empty")
    return value
```

```
@validator("taxi_number")
def validate_taxi_number(cls, value):
    if not value:
        raise CustomException("taxi number cannot be empty")
    return value
```

Package : taxi_register.py

code :

```
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox

from helper.exceptions import CustomException
from .taxi_model import TaxiModel
from .taxi_register_controller import TaxiController

class TaxiRegisterPage:
    def __init__(self, root, control_panel_frame):
        self.root = root
        self.control_panel_frame = control_panel_frame
        self.create_taxi_register_frame()

    def create_taxi_register_frame(self):
        self.taxi_register_frame = tk.Toplevel(self.root)
        self.taxi_register_frame.title("Taxi Register")
        self.taxi_register_frame.configure(background="#FFFFFF")
        self.taxi_register_frame.resizable(0, 0)
        self.taxi_register_frame.geometry("460x440+420+91")
        self.build_taxi_register_frame(
            taxi_register_frame=self.taxi_register_frame,
        )

    @staticmethod
    def build_taxi_register_frame(taxi_register_frame):

        welcome = tk.Label(taxi_register_frame)
        welcome.place(relx=0.340, rely=0.069, height=41, width=200)
        welcome.configure(
            activebackground="#f9f9f9",
            anchor="w",
            background="#FFFFFF",
            compound="left",
            text="Taxi Register",
            font="-family {Noto Sans} -size 16 -weight bold",
        )

        brand_label = tk.Label(taxi_register_frame)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
brand_label.place(relx=0.065, rely=0.19, height=41, width=104)
brand_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    font="-family {Noto Sans} -size 14",
    compound="left",
    background="#FFFFFF",
    text="Brand")
```

```
list_of_brand = [
    "Hyundai",
    "Maruti",
    "Honda"]
```

```
selected_brand = tk.StringVar()
selected_model = tk.StringVar()
selected_age = tk.StringVar()
```

```
brand_entry = ttk.Combobox(
    taxi_register_frame,
    textvariable=selected_brand,
    values=list_of_brand,
)
brand_entry.place(
    relx=0.065,
    rely=0.29,
    height=25,
    relwidth=0.404,
)
brand_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    state="readonly",
)
brand_entry.bind(
    "<>",
    lambda event: TaxiRegisterPage.model_viewer(
        event,
        model_entry,
        selected_brand,
    ),
)
```

```
model_user = tk.Label(taxi_register_frame)
model_user.place(relx=0.543, rely=0.19, height=41, width=104)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
model_user.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    background="#FFFFFF",  
    compound="left",  
    font="-family {Noto Sans} -size 14",  
    text="Model",  
)
```

```
model_entry = ttk.Combobox(  
    taxi_register_frame,  
    textvariable=selected_model,  
    state="readonly",  
)  
model_entry.place(  
    relx=0.543,  
    rely=0.29,  
    height=25,  
    relwidth=0.404,  
)  
model_entry.configure(  
    background="#EFF0F2",  
    font="TkFixedFont",  
)
```

```
taxi_number_label = tk.Label(taxi_register_frame)  
taxi_number_label.place(relx=0.065, rely=0.370, height=41, width=170)  
taxi_number_label.configure(  
    activebackground="#f9f9f9",  
    anchor="w",  
    compound="left",  
    background="#FFFFFF",  
    font="-family {Noto Sans} -size 14",  
    text="Taxi Number",  
)
```

```
taxi_number_entry = tk.Entry(taxi_register_frame)  
taxi_number_entry.place(  
    relx=0.065,  
    rely=0.470,  
    height=25,  
    relwidth=0.404,  
)  
taxi_number_entry.configure(  
    background="#EFF0F2",  
    font="TkFixedFont",  
    selectbackground="#c4c4c4",  
)
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : **BAIBHAV PAUDEL**

[REDACTED]

```
taxi_age_label = tk.Label(taxi_register_frame)
taxi_age_label.place(relx=0.543, rely=0.370, height=41, width=104)
taxi_age_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    text="Age",
)
```

```
taxi_age_entry = tk.Spinbox(
    taxi_register_frame,
    from_=1.0,
    to=9.0,
    state="readonly",
    textvariable=selected_age,
)
taxi_age_entry.place(
    relx=0.543,
    rely=0.470,
    height=25,
    relwidth=0.404,
)
taxi_age_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
)
```

```
discription_label = tk.Label(taxi_register_frame)
discription_label.place(relx=0.065, rely=0.550, height=41, width=300)
discription_label.configure(
    activebackground="#f9f9f9",
    anchor="w",
    background="#FFFFFF",
    compound="left",
    font="-family {Noto Sans} -size 14",
    text="Discription",
)
```

```
discription_entry = tk.Text(taxi_register_frame)
discription_entry.place(
    relx=0.065,
    rely=0.650,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        height=60,
        relwidth=0.883,
    )
description_entry.configure(
    background="#EFF0F2",
    font="TkFixedFont",
    selectbackground="#c4c4c4",
)
```

```
register_taxi_button = tk.Button(taxi_register_frame)
register_taxi_button.place(relx=0.522, rely=0.850, height=35, width=95)
register_taxi_button.configure(
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#FFFFFF",
    text="""Register""",
    command=lambda: TaxiRegisterPage.taxi_registration(
        taxi_register_frame,
        selected_brand,
        selected_model,
        taxi_number_entry,
        selected_age,
        discription_entry,
    ),
)
```

```
cancel_button = tk.Button(taxi_register_frame)
cancel_button.place(relx=0.250, rely=0.850, height=35, width=95)
cancel_button.configure(
    activebackground="beige",
    background="#007074",
    borderwidth="2",
    compound="left",
    font="-family {Noto Sans} -size 12",
    foreground="#FFFFFF",
    text="""Cancel""",
    command=lambda: TaxiRegisterPage.exit_taxi_registration(
        taxi_register_frame,
    ),
)
```

```
taxi_register_frame.grab_set()
taxi_register_frame.mainloop()
```

`@staticmethod`

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
def model_viewer(event, model_entry, selected_brand):
```

```
    hyundai_model = ["Grand i10 NIOS", "Aura", "Venue"]
    maruti_model = ["Swift dzire", "Eeco", "Ritz"]
    Honda_model = ["Amaze V", "city E", "mobilio s"]
```

```
if selected_brand.get() == "Hyundai":
    model_entry["values"] = hyundai_model
    model_entry.current(0)
    model_entry.update()
if selected_brand.get() == "Maruti":
    model_entry["values"] = maruti_model
    model_entry.current(0)
    model_entry.update()
if selected_brand.get() == "Honda":
    model_entry["values"] = Honda_model
    model_entry.current(0)
    model_entry.update()
```

```
@staticmethod
def exit_taxi_registration(taxi_register_frame):
    taxi_register_frame.destroy()
```

```
@staticmethod
def taxi_registration(
    taxi_register_frame,
    selected_brand,
    selected_model,
    taxi_number_entry,
    selected_age,
    discription_entry,
):
    try:
        taxi = TaxiModel(
            brand=selected_brand.get(),
            model=selected_model.get(),
            taxi_number=taxi_number_entry.get(),
            taxi_age=(selected_age.get()),
            discription=discription_entry.get("1.0", "end"),
        )
```

```
    taxi_number_entry.delete("0", "end")
    discription_entry.delete("1.0", "end")
```

```
    taxi_controller = TaxiController()
    taxi_controller.taxi_registration_control(
        taxi,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PADEL

```
    taxi_register_frame,
)
except CustomException as error:
    messagebox.showerror(
        "Invalid Data",
        error,
        parent=taxi_register_frame,
)
```

Package : taxi_register_controller.py

code :

```
from tkinter import messagebox
from helper.turbo_db import Turbo_db
```

```
class TaxiController:
    def __init__(self) -> None:
        self._connection = Turbo_db.turbo_connection()

    def taxi_registration_control(self, taxi, taxi_register_frame):
        if self.authenticate(taxi, taxi_register_frame):
            self.registration_sucess(taxi_register_frame)

    def authenticate(self, taxi, taxi_register_frame):
        try:
            cursor = self._connection.cursor()
            statement = """CREATE TABLE IF NOT EXISTS taxi(
                taxiID SERIAL PRIMARY KEY,
                brand VARCHAR(50) NOT NULL,
                model VARCHAR(50) NOT NULL,
                taxi_number VARCHAR(20) NOT NULL UNIQUE,
                taxi_age INT NOT NULL,
                discription VARCHAR(50) NOT NULL ,
                status VARCHAR(40)
            );"""
            cursor.execute(statement)
            self.registration_sucess(taxi_register_frame)
        except Exception as e:
            messagebox.showerror("Error", str(e))

    def registration_sucess(self, taxi_register_frame):
        messagebox.showinfo("Success", "Registration successful")

    def insert_data(self, taxi):
        data_insert = """INSERT INTO taxi(brand, model, taxi_number,
            taxi_age, discription,status) VALUES (%s, %s, %s, %s, %s, %s);"""
        data_values = (
            taxi.brand,
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
        taxi.model,
        taxi.taxi_number,
        taxi.taxi_age,
        taxi.description,
        "Available",
    )
cursor.execute(statement)
cursor.execute(data_insert, data_values)
self._connection.commit()
return True
```

```
except Exception as error:
    messagebox.showerror(
        "Invalid",
        "taxi number already exists",
        parent=taxi_register_frame,
    )
    print(error)
finally:
    if cursor is not None:
        cursor.close()
    if self._connection is not None:
        self._connection.close()
```

```
def registration_sucess(self, taxi_register_frame):
    messagebox.showinfo(
        "sucess",
        "taxi register",
        parent=taxi_register_frame,
    )
```

```
@staticmethod
def taxi_number_fetcher(taxi_number_entry):
    _connection = Turbo_db.turbo_connection()
    try:
        cursor = _connection.cursor()
        statement = "SELECT taxi_number FROM taxi WHERE status='Available';"
        cursor.execute(statement)
        taxi_number = cursor.fetchall()
        taxi_number_entry["values"] = taxi_number
        taxi_number_entry.update()
```

```
    except Exception as error:
        print(error)
    finally:
        if cursor is not None:
            cursor.close()
```

CIS020-1 – Introduction to Software Development

CIS093-1 – Mathematics and Concepts for Computational Thinking

Assignment 2 – Individual Project – Case Study (Taxi Booking System)

University ID : 2146504

Student Name : BAIBHAV PAUDEL

```
if _connection is not None:  
    _connection.close()
```

```
@staticmethod  
def taxi_data_fetcher(  
    welcome,  
    taxi_number_data,  
    taxi_age_data,  
    taxi_descripion_data,  
    selected_taxi_number,  
):  
    _connection = Turbo_db.turbo_connection()  
    try:  
        cursor = _connection.cursor()  
        statement = "SELECT * FROM taxi WHERE taxi_number=%s;"  
        data = (selected_taxi_number.get(),)
```

```
        cursor.execute(statement, data)  
        taxi_data = cursor.fetchall()  
        if taxi_data:  
            for data in taxi_data:  
                fetched_car_brand = data[1]  
                fetched_car_model = data[2]  
                fetched_car_number = data[3]  
                fetched_car_age = data[4]  
                fetched_car_descripion = data[5]  
        else:  
            return False
```

```
welcome.config(text=fetched_car_brand + " " + fetched_car_model)  
taxi_number_data.config(text=fetched_car_number)  
taxi_age_data.config(text=fetched_car_age)  
taxi_descripion_data.config(text=fetched_car_descripion)  
welcome.update()  
taxi_number_data.update()  
taxi_number_data.update()  
taxi_descripion_data.update()
```

```
except Exception as error:  
    print(error)  
finally:  
    if cursor is not None:  
        cursor.close()  
    if _connection is not None:  
        _connection.close()
```