

# DBMS MILESTONE : 3

Member 1 : Chinmayee Behera (2020CS10337)

Member 2 : Rajni Poonia (2020CS10371)

Member 3 : Sheetal Kanel (2020CS10381)

## Overall architecture

Connector used - psycopg2

Interface - flask

Database - postgresql

Pages designed in html


Styling done in css/js

## Description :

- **RideOn** is a web application used for booking on-demand transportation services such as cabs and auto-rickshaws. Customers can book rides according to their needs.
- We used **psycopg2** which is PostgreSQL database adapter for the Python. It also allows Python programs to connect to PostgreSQL databases and perform various operations such as executing SQL statements, and managing and retrieving results.
- **psycopg2.connect()** is used to connect to database, which takes in the necessary connection parameters such as the host, database name, username, and password. We then create a cursor object using inbuilt function.
- THE **os** module is used to access environment variables that are set on the operating system. Flask applications often use environment variables to store configuration information that can be changed without modifying the application code.
- The **os.environ** dictionary is used to access the value of the DATABASE\_URL environment variable, which contain the connection string to the application's database.
- The **Flask** module is used to create the Flask application instance. The Flask constructor takes the name of the application module or package as its

argument, and is typically used to define the routes and configuration for the application.

- The **redirect** module is used to redirect the user to a different URL.
- The **render\_template** module is used to render an HTML template.
- The **request** module is used to access data submitted in an HTTP request.
- The **url\_for** module is used to generate URLs for Flask routes.




---

Passenger login

Driver login

Admin login



---

Email address

Enter your email address

Password

Enter your password

Login

SOURCE	Enter your pick up location (number from 1 to 10)
DESTINATION	Enter your drop location (number from 1 to 10)
START TIME	Enter 'now' to book a cab now, or enter a later time today (HH:MM:SS)
WAITING TIME	Enter your waiting time (HH:MM:SS)
WAITING PLACE	Enter your waiting location (number from 1 to 10)
CAPACITY	Enter Number of Passenger (number from 1 to 10)

Search Cabs

Street

Enter the street

Locality

Enter the locality

City

Enter the city

State

Enter the state

Pincode

Enter the pincode

Submit

## **Pages (short description + input parameters) :**

1. Homepage: Main page of our app.It contain 3 options :
  - Passenger login
  - Driver login
  - Admin login
2. Login:
  - Enter email id and password.
3. Sign Up:
  - Enter details:
  - First name
  - Last name
  - Contact number
  - Email id
  - Gender
  - Date of birth(that will be the password)
4. Cab detail
  - Your unique cab id
  - Condition of cab
  - Cab type
5. User home page
  - Source
  - Destination
  - Start time
  - Waiting time
  - Waiting place
  - capacity
6. Add location
  - Street
  - Locality
  - City
  - State
7. Driver profile page will contain these details
  - Driver ID, vehicle id, Name, contact, Rating ,travel completed
8. Admin home page

10 top drivers with highest rating

View top 10 saved locations

View 10 lowest earning drivers

View top 10 start and end locations with the maximum rides

9. Most visited

View top 10 start and end locations with the maximum rides

10. pastrides:

Complete rides of passenger in past

## **Flow of website for driver/passenger/admin :**

When the website is started the home page is the landing page regardless of the identity of the user i.e. passenger or driver or admin user

Depending upon the options clicked by the user, he is redirected to one of login/signup pages.

If signup is clicked, the user details are inserted into the respective tables in the database. If a driver signs up then an additional page appears in which he has to input the details of the vehicle he is using.

If login is clicked, the user details are verified from the existing database. If the user exists then he is redirected to the home page of passenger/driver home page.

### **1. For passenger :**

A passenger then has to enter source, destination, waiting place(if any), start time, waiting time(if any). Then upon clicking on the search ride button, a list of available drivers appear on the screen. This screening is done from the database by tallying the driver's location with the source input by the passenger. The list viewed is in a decreasing order of the driver's rating. Another criteria is used for screening which is the capacity input by the passenger. For each passenger only those drivers are viewed whose cabs have a maximum capacity more than the number input by the passenger

Once the passenger clicks on any one of the drivers, the availability of the driver is updated to unavailable and a review is shown where the passenger can rate the driver and also update/add any feedback.

### **2. For Driver :**

The driver home page has options of viewing:

- a. Your earnings where his earnings are added up from all the successful rides and shown on the screen
- b. Cab details where the vehicle details are viewable

- c. When a current booking option is clicked, the ride selected by the passenger is viewed with the passenger details, source, destination, etc.
- d. Past booking button redirects the driver to a page where he can view all his past rides
- e. Your profile button views his present rating and all other personal details.

### **3. For Admin :**

The admin has access to the entire database. He can edit/add/drop any users in the tables. He will basically use the home page to view details of the website

Details he can view on his home page:

- a. View top 10 saved locations
- b. View top 10 drivers according to rating
- c. View 10 lowest earning drivers
- d. View top 10 start and end locations with the maximum rides