THE COMMERCIAL VEHICLE BOOKING

MUHSINA

MES24MCA-2035

Department of Computer Applications

MES College of Engineering, Kuttippuram

17-10-25

PRODUCT OWNER

Ms.FEBIN AZIZ

Assistant Professor

DEPARTMENT OF COMPUTER APPLICATIONS

MES COLLEGE OF ENGINEERING, KUTTIPPURAM

TABLE OF CONTENTS

THE COMMERCIAL VEHICLE BOOKING

- Booking commercial vehicles is inefficient and lacks price transparency.
- An online platform to simplify commercial vehicle booking.
- The service will be accessible via a website and a mobile application for user convenience.

Key Features:

- Browse and select vehicles.
- Transparent pricing for easy comparison.
- Real-time vehicle availability and scheduling.

OBJECTIVES

- **Easy Booking**: Create a simple platform for booking commercial vehicles.
- **User-Friendly**: Ensure easy navigation and access to vehicle info.
- **Wide Selection**: Maintain a large database of commercial vehicles.
- **Admin Tools**: Develop a dashboard for managing users, drivers, and vehicles.
- **Self-Service for Users/Drivers**: Allow users and drivers to manage their own profiles and bookings.

EXISTING SYSTEM

- Vehicle booking is mostly done manually through phone calls or local agents
- Limited visibility of available vehicles leads to booking delays
- No unified platform to compare vehicle types, prices, or schedules
- Manual record-keeping causes errors and inefficiencies
- Lack of transparency in pricing and availability

PROPOSED SYSTEM

- **Online Platform:** Connects businesses and individuals with commercial vehicle operators.
- **Access:** Available via a website or mobile application.
- **Key Features:**Transparent pricing, Real-time availability and scheduling, Vehicle tracking.
- **Benefits:** Offers convenience, time savings, and cost efficiency.

MOTIVATIONS

- This project aims to create a centralized online platform to fix the inefficient and non-transparent traditional process of booking commercial vehicles.
- Enhance the user experience with a user-friendly interface for easy navigation and quick access to information.
- Provide transparent pricing , real-time availability , and vehicle tracking to offer convenience, save time, and promote cost efficiency.
- A key motivation was to address difficulties in project planning and creating an attractive user interface.

FUNCTIONALITIES

- **User & Driver Management:**
- **Login & Registration:** Users and drivers can create accounts and log in.
- **Profile Management:** Users and drivers can add and update their personal details and change passwords.
- **User/Driver Access:** Users can view information about available drivers and vehicles.

FUNCTIONALITIES

- **Vehicle & Booking Management:**
- **Vehicle Booking:** Users can search, view, select, and book commercial vehicles based on various requirements (type, capacity, location, dates).
- **Vehicle Management (Drivers):** Drivers can add and delete their registered vehicles.
- **Vehicle Filtering (Users):** Users can filter vehicles by type for easier access.
- **Admin Dashboard:** An administrative interface for managing:
- Vehicle inventory (add, edit, delete).
- User accounts and permissions.
- Booking records.

MODULE DESCRIPTION

- **User Module:** Manages user accounts and allows users to browse, book, and view commercial vehicles.
- **Driver Module:** For vehicle operators to register, log in, add or delete vehicles, and manage their bookings and profile.
- **Admin Module:** Provides a dashboard for administrators to manage users, drivers, vehicle inventory, and bookings.

DEVELOPING ENVIRONMENT

- **Framework:** Django.
- **Front End:** Templates from Colorlib and Bootstrapdash.
- **Back End:** Python with the Django framework.
- **Database:** A relational database is used to manage user, driver, vehicle, booking, and payment data.
- **Libraries:**
- **Django** Jazzmin:Improves the Django admin's look and navigation.
- **Pillow:** Processes images, enabling resizing and cropping.
- **IDE:**VS Code

SPRINT BACKLOG

Backlog tem	Status And Completion Date	Original Estimation in Hours	Day 1 hrs	Day 2 hrs	Day 3 hrs	Day 4 hrs	Day 5 hrs	Day 6 hrs	Day 7 hrs	Day 8 hrs	Day 9 hrs	Day 10 hrs
SPRINT1												
Design user interface/registration	10-07-25	8	2	2	0	1	2	1	1	0	0	0
Database setup	16-05-25	10	1	1	1	1	2	2	1	1	0	0
Basic backend API	24-07-25	8	1	1	1	1	1	1	1	0	1	0
SPRINT2												
Vehicle data management	30-07-25	10	2	1	1	2	1	2	1	0	0	0
Vehicle Search functinality	07-08-25	8	1	1	1	1	1	1	1	1	0	0

SPRINT BACKLOG

Backlog tem	Status And Completion Date	Original Estimation in Hours	Day 1 hrs	Day 2 hrs	Day 3 hrs	Day 4 hrs	Day 5 hrs	Day 6 hrs	Day 7 hrs	Day 8 hrs	Day 9 hrs	Day 10 hrs
Vehicle details page	13-08-25	10	1	2	1	2	2	1	1	0	0	0
Vehicle API development	27-08-25	12	2	2	2	2	2	2	0	0	0	0
SPRINT3												
Booking form implementation	05-09-25	7	1	1	1	1	1	1	1	0	0	0
Booking api development	10-09-25	7	1	1	1	1	1	1	1	1	0	0
Booking confirmation	20-09-25	10	1	2	2	2	1	1	0	0	0	1

SPRINT BACKLOG

Backlog tem	Status And Completion Date	Original Estimation in Hours	Day 1 hrs	Day 2 hrs	Day 3 hrs	Day 4 hrs	Day 5 hrs	Day 6 hrs	Day 7 hrs	Day 8 hrs	Day 9 hrs	Day 10 hrs
SPRINT4												
Admin dashboorad	25-09-25	7	1	1	1	1	1	1	1	0	0	0
User &Booking Management(admin)	05-10-25	7	1	1	1	1	1	1	1	0	0	0
Responsive Design & UX Review	08-10-25	5	1	1	1	1	0	0	0	0	1	0
TOTAL		102	13	17	14	17	15	15	15	9	1	2

PRODUCT BACKLOG

ID	NAME	PRIORITY <high/medium/low>	ESTIMATE (Hours)	STATUS <Planned/In progress/Completed>
1	User Authentication & Profiles	High	23	compeleted
2	Core Database Infrastructure	High	13	compeleted
3	Basic Backend Services	High	20	compeleted
4	Initial Frontend Setup	MEDIUM	10	compeleted
5	Admin Vehicle Management (CRUD)	HIGH	10	compeleted

PRODUCT BACKLOG

ID	NAME	PRIORITY <high/medium/low>	ESTIMATE (Hours)	STATUS <Planned/In progress/Completed>
6	Vehicle Search Functionality	High	23	compeleted
7	Detailed vehicle information	High	13	compeleted
8	Vehicle data APIs	High	20	compeleted
9	Search & Listing Testing	MEDIUM	10	compeleted
10	Vehicle Booking Process	HIGH	20	compeleted

PRODUCT BACKLOG

ID	NAME	PRIORITY <high/medium/low>	ESTIMATE (Hours)	STATUS <Planned/In progress/Completed>
11	Booking Management APIs	High	23	compeleted
12	Booking Confirmation &history	High	13	compeleted
13	Admin Dashboard	MEDIUM	20	compeleted
14	Admin User & Booking Control	HIGH	10	compeleted
15	Responsive Design & UX Review	MEDIUM	10	compeleted

USER STORY

User Story ID	As a type of User	I want to <Perform some task>	So that i can <Achieve Some Goal>
1	USER	Register	Acess the booking system and personalize my experience
2	USER	Login	securely access my profile and booking history
3	USER	View Profile	View users profile in application
4	ADMIN	manage user accounts (CRUD)	control system access and assist users with their profiles
5	ADMIN	manage vehicle inventory (CRUD)	(CRUD)add, update, and remove vehicles available for booking
6	ADMIN	Manage booking records	Oversees all booking and assist with modifications
7	USER	View Profile	View users profile in application

USER STORY

User Story ID	As a type of User	I want to <Perform some task>	So that i can <Achieve Some Goal>
8	USER	search for available commercial vehicles	find a suitable vehicle for my transport needs
9	USER	view detailed information about a vehicle	make an informed decision before booking
10	USER	book a commercial vehicle	reserve a specific vehicle for a defined period
11	ADMIN	view an administrative dashboard	get an overview of system activity, bookings, and users
12	ADMIN	send system-wide notifications	inform users about important updates or announcements

PROJECT PLAN

User StoryID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	16/08/2025	17/08/2025	13	completed
2		18/08/2025	19/08/2025		completed
4		20/08/2025	22/08/2025		completed
3		23/08/2025	24/08/2025		completed
1,2		25/08/2025	27/08/2025		completed
5	Sprint 2	13/09/2025	23/09/2025	14	completed
7		23/09/2025	24/09/2025		completed
8,9		24/09/2025	29/09/2025		completed

PROJECT PLAN

User StoryID	Task Name	Start Date	End Date	Days	Status
10	SPRINT 3	03/10/2025	06/10/2025	15	completed
11		07/10/2025	10/10/2025		completed
12,13		13/10/2025	20/10/2025		completed
15	SPRINT 4	28/10/2025	5/11/2025	15	completed
4,6		7/11/2025	11/11/2025		completed
14,16		08/10/2025	09/11/2025		completed

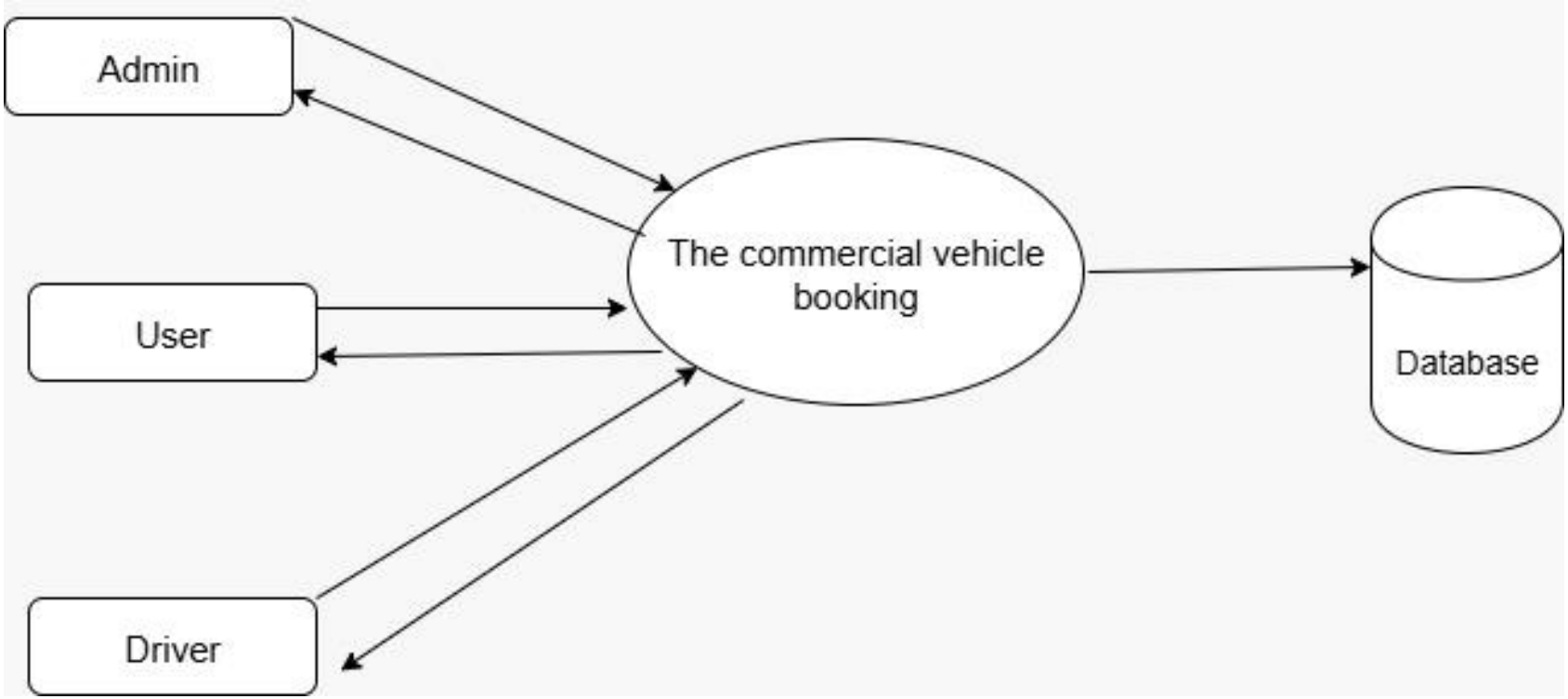
DATA FLOW DIAGRAM

- LEVEL 0

The figure given above is for reference only. Create a figure with your data.

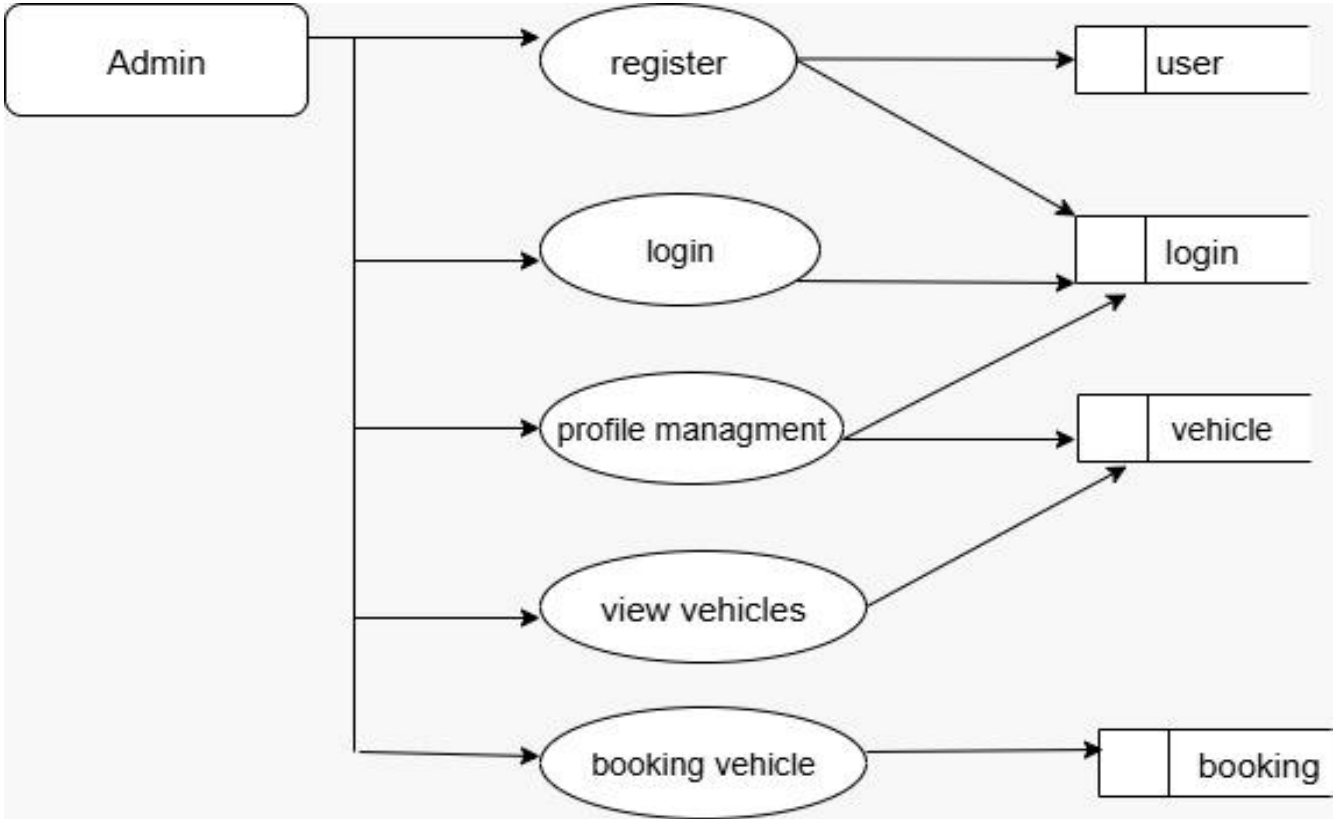
DATA FLOW DIAGRAM

- Level 1



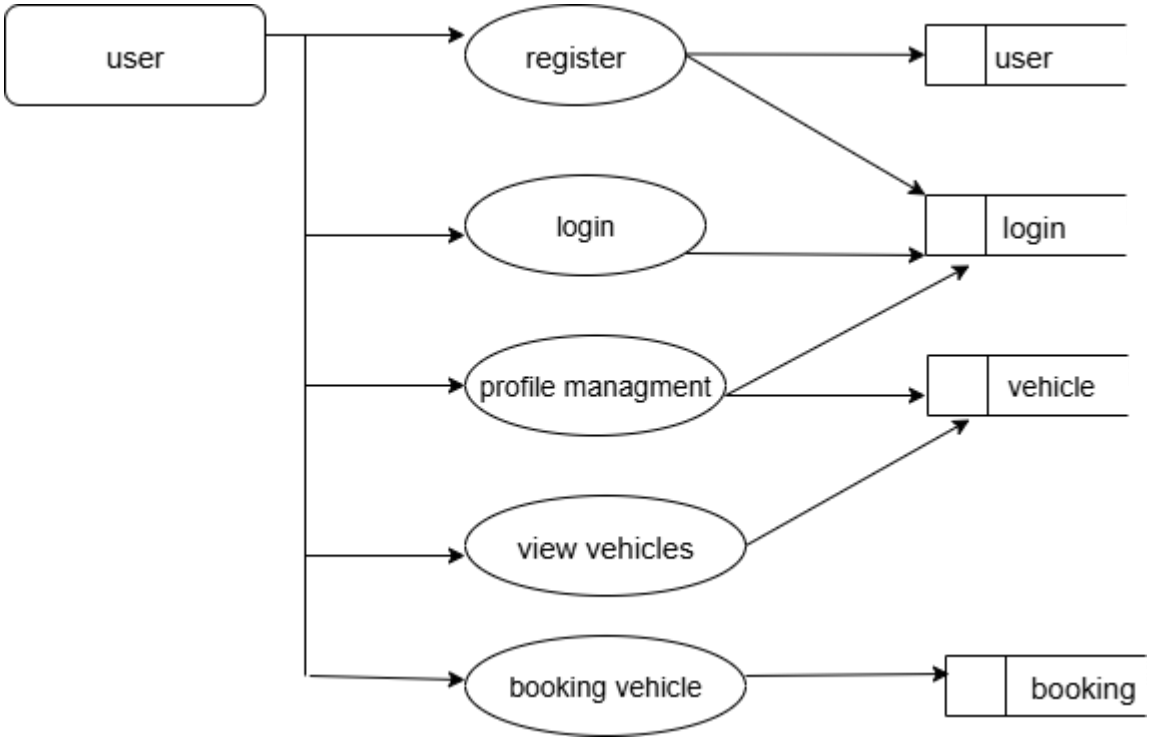
DATA FLOW DIAGRAM

- Level 1.1



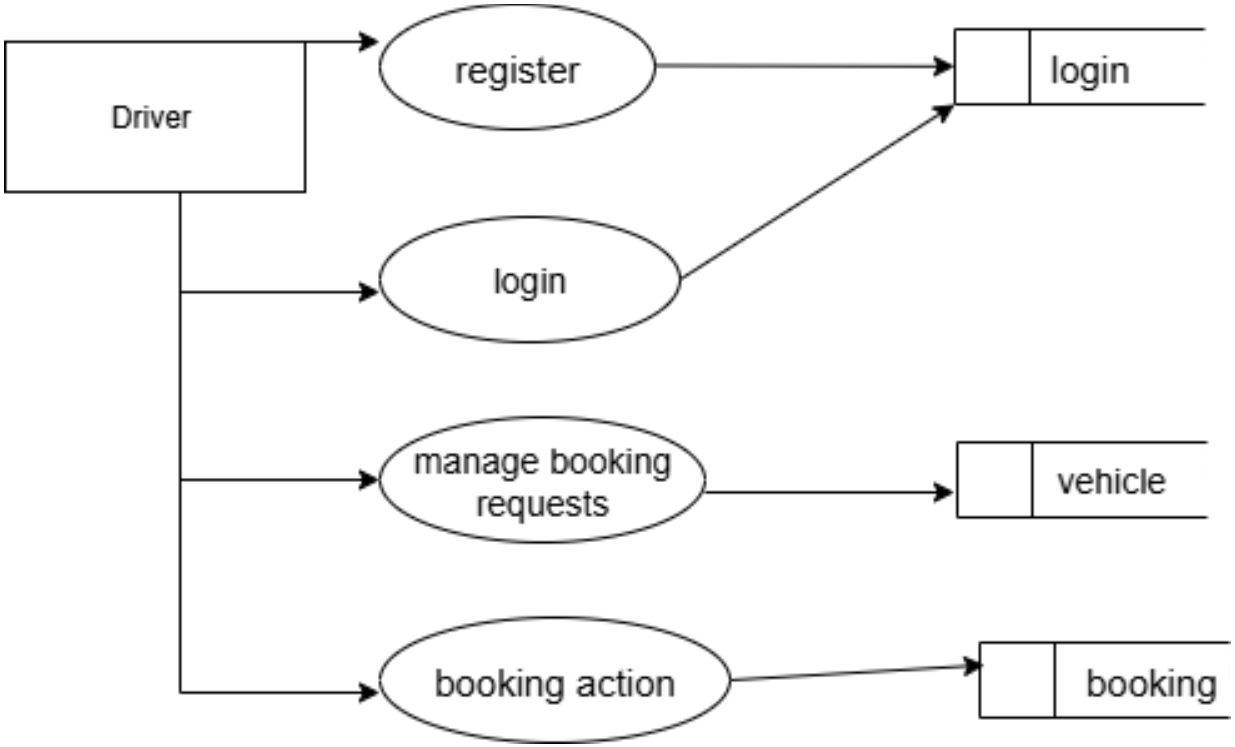
DATA FLOW DIAGRAM

- Level 1.2

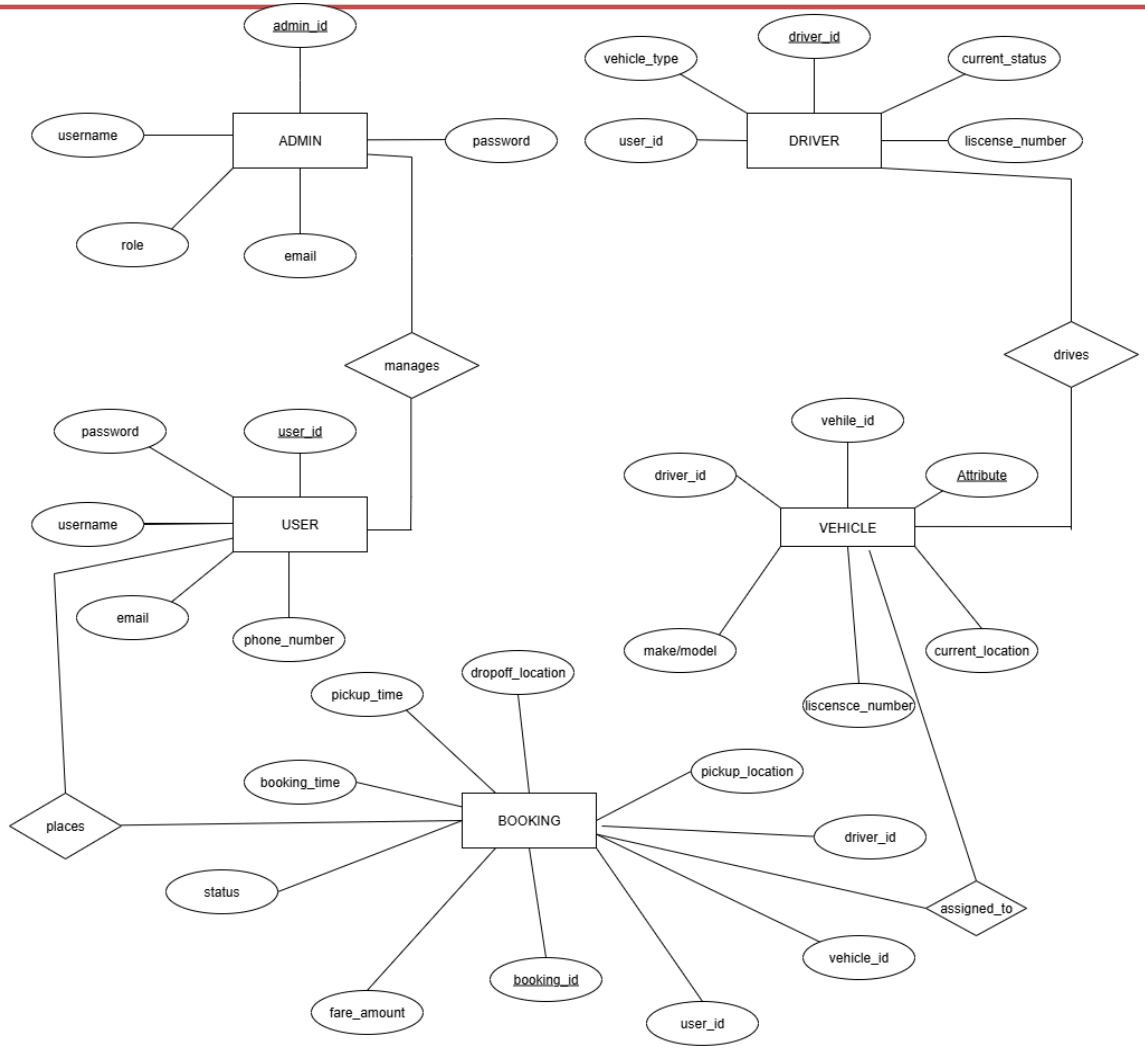


DATA FLOW DIAGRAM

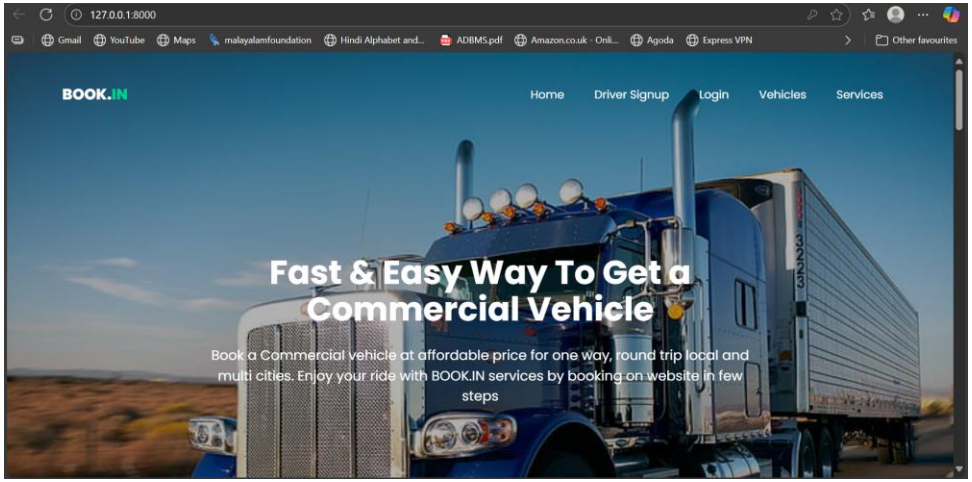
- Level 1.3



ER DIAGRAM

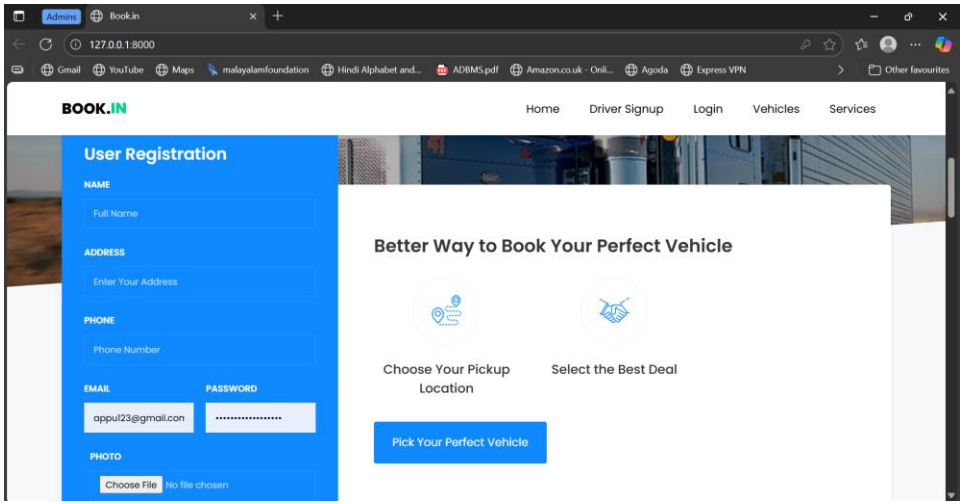

admin_id
username
ADMIN
password
role
email
manages
vehicle_type
driver_id
current_status
user_id
DRIVER
liscense_number
drives
password
user_id
username
USER
email
phone_number
vehile_id
driver_id
Attribute
VEHICLE
make/model
current_location
liscensce_number
dropoff_location
pickup_time
booking_time
pickup_location
places
BOOKING
driver_id
status
assigned_to
vehicle_id
booking_id
fare_amount
user_id

Screenshots

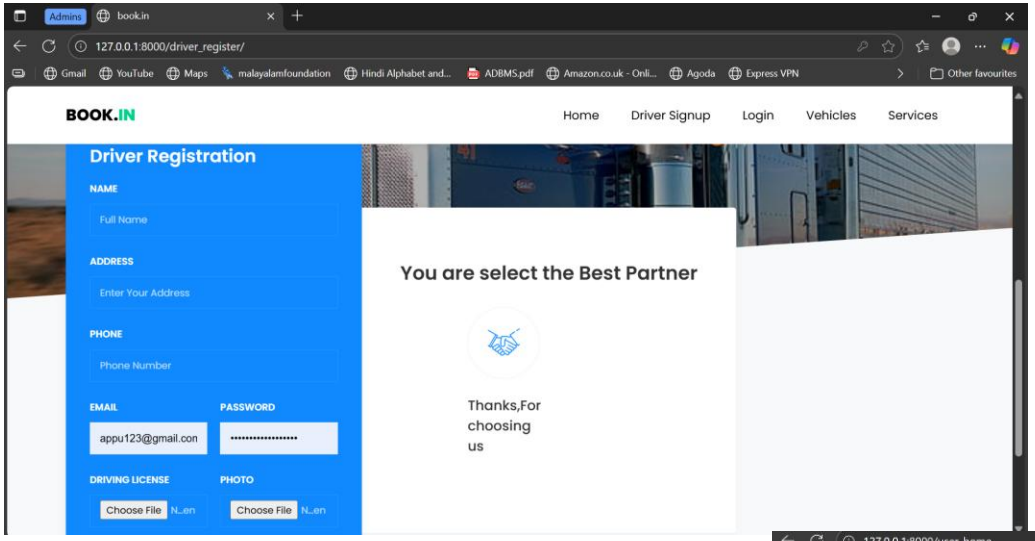


Home Page

user registration

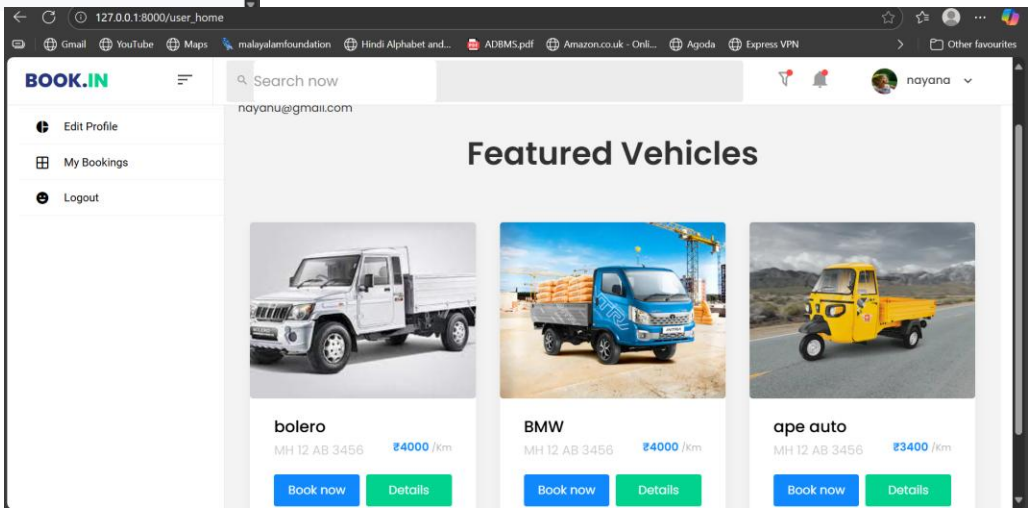


Screenshots



Driver Registration

Bookings



Source Code

❖ Urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name='index'),
    path('detail/<int:pk>/',views.details_vehicle),
    path('vehicles',views.vehicles),
    path('driver_register/',views.driver_registration, name='driver_registration'),
    path('login',views.login,name='login'),
    path('view_license/<int:id>/', views.view_license, name='view_license'),
    path('services',views.services),
    path('user_home',views.user_home),
    path('logout/', views.logout_view, name='logout'),
    path('driver_home',views.driver_home),
    path('add_vehicle',views.add_vehicle),
    path('delete_vehicle/<int:id>/',views.delete_vehicle),
    path('filter/<int:fid>/',views.filter),
    path('drivers',views.view_drivers),
    path('search_vehicles',views.search_vehicles,name='search_vehicles'),
    path('book_vehicle/<int:vehicle_id>/',views.book_vehicle),
    path('viewbookings',views.view_booking),
    path('mybookings',views.my_booking),
    path('stats',views.view_stats),
    path('make_payment/<int:booking_id>/', views.make_payment, name='make_payment'),
    path('edit-user',views.edituser),
    path('changep-user',views.changepassword_user),
    path('edit-driver',views.editdriver),
    path('changep-driver',views.changepassword_driver),
    path('uview_drivervehicle/<int:did>/',views.view_driver_vehicles)
]
```

Source Code

❖ **models.py**

```
from django.db import models
from django.urls import path

class User(models.Model):
    name = models.CharField(max_length=100)
    email = models.EmailField(unique=True)
    password = models.CharField(max_length=100)
    image=models.ImageField(upload_to='drimg/',default='nothing')
    address=models.CharField(max_length=100,default='nothing')
    phone=models.IntegerField(default='1')
    def __str__(self):
        return self.name


class Driver(models.Model):
    name = models.CharField(max_length=100)
    email=models.CharField(max_length=50,default='nothing')
    license = models.ImageField(upload_to='license/',default='nothing')
    password =models.CharField(max_length=18,default='nothing')
    image=models.ImageField(upload_to='drimg/',default='nothing')
    address=models.CharField(max_length=100,default='nothing')
    phone=models.IntegerField(default='1')
    status=models.CharField(max_length=20,default='Not Verified')
    is_verified = models.BooleanField('Is Verified', default=False)
    def __str__(self):
        return self.name


class vehicle(models.Model):
    userid=models.ForeignKey(Driver,on_delete=models.CASCADE)
    vehicle_name = models.CharField(max_length=100)
    vehicle_reg = models.CharField(max_length=20)
    vehicle_type = models.CharField(max_length=500)
    vehicle_image = models.ImageField(upload_to="vehicles/", blank=True)
    rate=models.IntegerField(null=True)
    status=models.CharField(max_length=20,default="not booked")
```

THANK YOU