Ride Sharing and CO2 Emissions: A Project Assessment and Proposal for Sustainability

1. Introduction

1.1 Overview

Carbon dioxide (CO2) emissions are a major contributor to climate change and global warming. They are produced by the burning of fossil fuels, such as coal, oil, and natural gas, which are used to power transportation, electricity generation, and other activities. Transport is a key contributor to carbon dioxide (CO2) emissions in India. It is 14% of energy-related direct CO2 emissions and one of the fastest growing emissions sectors in the country, along with industry. Within the transport sector, road transport is responsible for 90% of total energy consumption.

India's motor vehicle fleet has grown rapidly in recent years, and before the COVID-19 pandemic, the on-road vehicle stock was expected to almost double to over 200 million by 2030. Nationally Determined Contribution (NDC) under the Paris Agreement, India has committed to reduce the emissions intensity of its gross domestic product (GDP) by 33% to 35% below 2005 levels by 2030 (Climate Action Tracker, 2020).

1.2 Purpose

Ride pooling, also known as carpooling, is a form of transportation in which multiple passengers share a ride in a single vehicle. By sharing a ride, passengers can reduce the number of trips taken, which can lead to lower fuel consumption and fewer emissions per person.

Ride pooling can be arranged informally, such as between friends or colleagues, or through organized programs, such as those offered by employers or government agencies. It can also be facilitated through technology, such as smartphone apps, which allow passengers to connect and share rides with others who are traveling along similar routes. In addition to reducing CO2 emissions, ride pooling also has other benefits, such as reducing traffic congestion, improving access to transportation, and reducing the costs of transportation. As a result, it is considered an important strategy for addressing the challenge of reducing emissions from transportation and mitigating the impacts of climate change.

Diya Ride Share Support the government policy by sharing the vehicle from one point to another within the state.

2.Theoritical Analysis

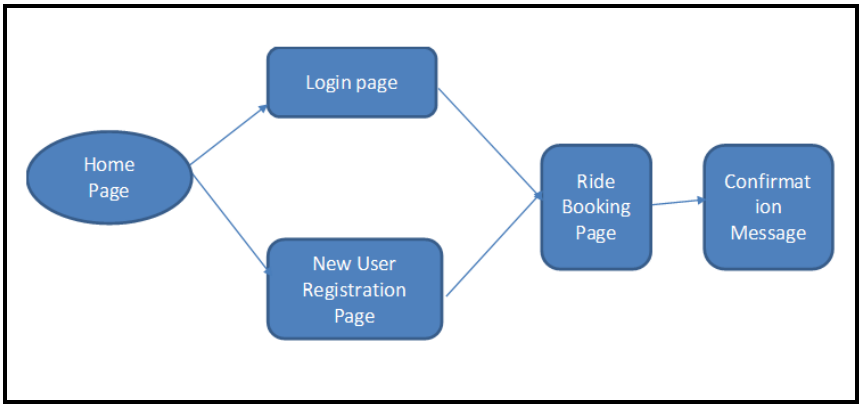
The Diya ride app is developed using python flask framework. Flask is a web framework that allows developers to build lightweight web applications quickly and easily with Flask Libraries. It is basically based on the WSGI toolkit and Jinja2 templating engine. Flask is installed using pip command. Flask server file "app.py" is connected with ibm cloud db2. The ibm-db python library is used for the connectivity. The register table is created. The table structure is shown below.

Fname	Varchar
Lname	Varchar
Designation	Varchar
Organization	Varchar
Email	Varchar
Phone	Number

Password	Varchar

Diya ride application has home,login,newuser and display html files. The application files were uploaded in the git- hub repository. The DigiCertGlobalRootCA certificate is SSL certificate required to connect to ibm-db2 and requirement .txt file contains the details micro services used in the application development. These files also uploaded in the github repository.

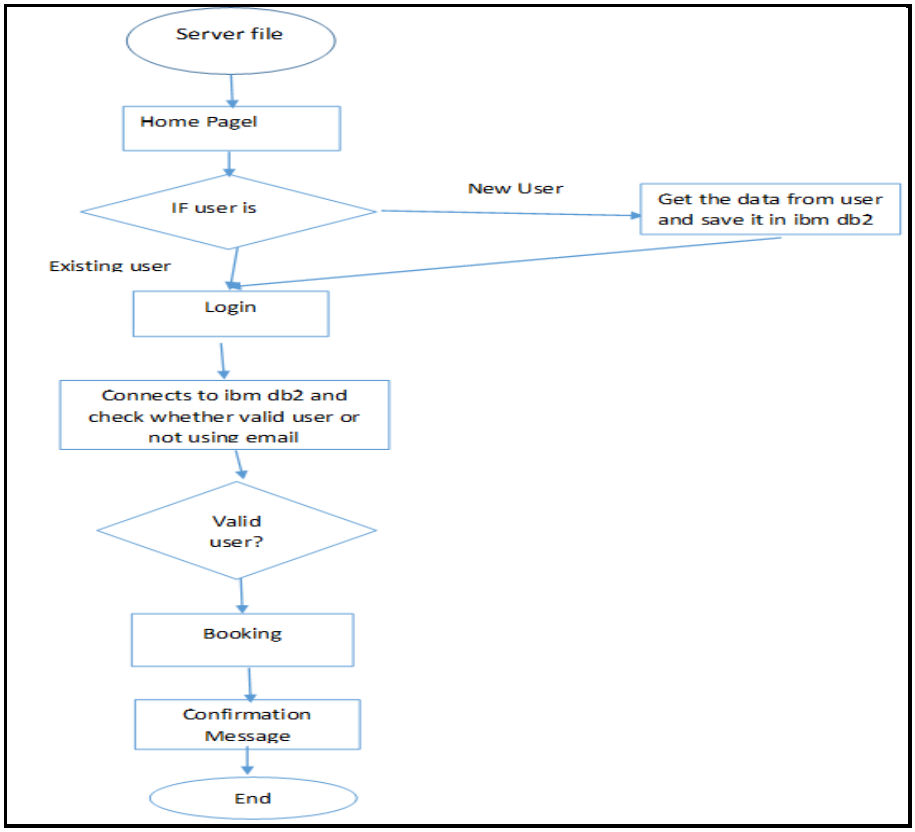
2.1 Block Diagram



2.2 Hardware /Software Designing

Hardware : Intelcore i5, Windows 64 bit Operating System	Software :Anaconda Distribution – Spyder, HTML ,IBM Cloud

3 Flow Chart



4. Result
This application is developed using python flask framework. The Diya ride share is used to group the passenger travelling from one point to another. Depending upon the number of passenger the transport can be chosen by user. By sharing the transport, the emission of Co2 can be reduced.

5. Advantages: Building the app using flask and deploying it in the cloud is a very simple procedure.

collaborating with github with open redhat shift is very user friendly and any one can do that.

Disadvantages: Proper libraries with versions are required to connect with IBM db2.

6.Conclusion: The project gave an idea of how to develop an application and deploy it in cloud platform. The practical approach of using the Saas, Paas and Iaas was informative and gained knowledge about it. Handling of flask framework and integrating python with it was a user friendly steps. Creating Wastson assistant Chabot and interacting with user gave an exposure to functionality of Chabot and how to integrate it with flask application. Also the practical approach of using github repository and deploying github files with red hat open shift gave knowledge on how to deploy application in cloud. The ibm cloud object storage supports in storing files that can be shared to any one by making it public url. Finally, developing Diya ride share app has developed skills in various technologies.

7.Future Scope: The Diya ride app is designed using flask python. It is designed to book rides within the state. In the future, apps can be extended to book the ride between states.

8. Bibliography:

1. https://community.ibm.com/community/user/datamanagement/blogs/youssef-sbai-idrissi1/2023/07/24/ibm-db2-with-python

2. https://www.ibm.com/docs/en/cloud-private/3.1.0?topic=services-watson-assistant

9. Appendix

A. Source Code

App.py

```
from flask import Flask, render_template,request,url_for,redirect
from flask import *
from datetime import datetime
import ibm_db
app = Flask(__name__,template_folder='template')
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY=SSL;SS
LServerCertificate=DigiCertGlobalRootCA.crt;UID=fqr84171;PWD=b9VLdFMZ1xDRiRKr;", ", ")

#@app.route("/", methods=['GET'])
@app.route("/")
def home():
    return render_template("home.html")
#@app.route("/login", methods=['GET'])
@app.route("/login")
def login():
    return render_template('login.html')
@app.route("/newuser")
def newuser():
    return render_template("newuser.html")
   @app.route("/newsubmit",methods = ['POST'])
def newsubmit():
```

```
    x= [x for x in request.form.values()]
    Fname=x[0]
    Lname = x[1]
    Designation=x[2]
    Organisation = x[3]
    email=x[4]
    phone=x[5]
    password = x[6]
    sql = "SELECT * FROM FQR84171.REGISTER WHERE EMAIL = ?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, email)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    #print(account)
    if account:
        return render_template('login.html', pred="You are already a member, please login using your
details")
    else:
        insert_sql = "INSERT INTO  REGISTER VALUES (?, ?, ?, ?, ?, ?, ?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, Fname)
        ibm_db.bind_param(prep_stmt, 2, Lname)
        ibm_db.bind_param(prep_stmt, 3, Designation)
        ibm_db.bind_param(prep_stmt, 4, Organisation)
        ibm_db.bind_param(prep_stmt, 5, email)
        ibm_db.bind_param(prep_stmt, 6, phone)
        ibm_db.bind_param(prep_stmt, 7, password)
        ibm_db.execute(prep_stmt)
        return render_template('login.html', pred="Registration Successful, please login using your
details")

@app.route('/submit',methods=['POST'])
def login1():
    email = request.form['email']
    password = request.form['password']
    sql = "SELECT * FROM REGISTER WHERE email =? AND password=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.bind_param(stmt,2,password)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    if account:
        return render_template('Booking.html',result=email)
    else:
        return render_template('login.html', pred="Login unsuccessful. Incorrect username/password !")

@app.route('/book', methods=['POST'])
def book():
    p=request.form['pickup']
    d=request.form['drop']
    return render_template('Display.html',result=p,d=d,pred="Your Ride is Confirmed.. Our Person
will call you shortly",datetime = str(datetime.now()))

@app.route('/cancel', methods=['POST'])
def cancel():
```

```
    p=""
    d=""
    return render_template('Display.html',result=p,d=d,predic="Your Ride is Cancelled",datetime =
str(datetime.now()))


@app.route('/co2test', methods=['GET','POST'])
def co2test():
    return render_template('co2test.html')

if __name__ == "__main__":
    app.run(debug=True,port=5000)
```

Home.html

```
<!DOCTYPE html>
<html>
<head>
<link
      rel="stylesheet"
      href= "{{url_for('static', filename='css/stylesheet.css')}}"/>

            <title>  DIYA RIDE SHARING </title>
</head>
<style>
.button {
  background-color:#008CBA;
font-size: 12px;
  padding: 15px 32px;
border-radius: 12px;
 display: inline-block;
}
.p{text-align: Justify;
      font-size: 14;
      font-style:Times New Roman ;
}
a {
   color: Black;
   text-align: center;
   font-size: 16px;
}</style>
<body style = "background-color: pink;">
<form >
<div>
<img src="/static/taxiapp.jpg"  style="float:left;" width="100" height="100" >
<h1 align = "center" style="color: Green;" > DIYA RIDE SHARING</h1>
</div>
            <h1 align = "center" style ="color:green;"><p>Right Choice for the Travellers.</p></h1>
<p><h2 align="center" style="color:Green;">Travellers can use Diya Ride Sharing to book your cab
and enjoy your travel with in the State.</h2></p>
<br>
<p style="color: blue;">Transport is a key contributor to carbon dioxide (CO2) emissions in India. It
is 14% of energy-related direct CO2 emissions and one of the fastest growing emissions sectors in the
country, along with industry.Within the transport sector, road transport is responsible for 90% of total
energy consumption.</p>
```

```
<p style="color: blue;">India’s motor vehicle fleet has grown rapidly in recent years, and before the COVID-19 pandemic, the on-road vehicle stock was expected to almost double to over 200 million by 2030. Nationally Determined Contribution (NDC) under the Paris Agreement, India has committed to reduce the emissions intensity of its gross domestic product (GDP) by 33% to 35% below 2005 levels by 2030 (Climate Action Tracker, 2020). While the government has accordingly defined economy-wide emissions reduction targets, thus far there are not any sector-specific targets for high-emitting sectors like industry and transport.</p>

<p style="color: blue;">Diya Ride Sharing Support the government policy by sharing the vehicle from one point to another within the state</p>
<br><br>
<div align = "center" >
<a href ="/login" ><button class = "button">Login</a></button>
<a href ="/newuser" ><button class = "button">Newuser</a></button>
<a href = "/co2test"><button class = "button">Co2 Emission</a></button>
</div>
</form>
</body>
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "eeda699e-872d-4dc1-8ff8-63a69f7ccb02", // The ID of this integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "6c186676-4c7d-4cf8-be28-9925f47bc511", // The ID of your service instance.
    onLoad: async (instance) => { await instance.render(); }
  };
  setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
  });
  </script>
</html>
```

Login.Html

```
<!DOCTYPE html>
<html>
<head>
        <title>Login</title>
</head>
<style>
.button {
                background-color: #555555;
                font-size: 12px;
                padding: 10px 24px;
                border-radius: 12px;
        }

        body {
  background-color: pink;
 }
.header {
  background-color: rgba(255, 0, 0, 0.3);
  padding: 20px;
  text-align: center;
```

```
  text-decoration-color: ;
  font-size: 16px;
}
p{text-align: center;
 font-size: 14;
font-style: italic;
}
</style>
<body>
<form action = "submit", method ="post">
<header class="header"><h1> DIYA RIDE SHARING</h1></header>
  <h1 style="background-color:DodgerBlue;"><p>Right Choice for the Travellers </p></h1>
    <h3>{{pred}}</h3>
    <div align="center">
            <h3 >Login Form</h3>
            <LABEL>Email ID</LABEL>
            <input align="center" type="email" name="email" required="" placeholder= "enter email">
            <br>     <br>
<LABEL>Password</LABEL>
            <input align="center" type="password" name="password" required="" placeholder= "enter
Password">
            <br>
            <br>
<a href ="/submit"><button class="button">Submit</button>
<br><br><br>
<a href = "/newuser"> New User </a>
</div>
</form>
</body>
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "eeda699e-872d-4dc1-8ff8-63a69f7ccb02", // The ID of this integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "6c186676-4c7d-4cf8-be28-9925f47bc511", // The ID of your service instance.
    onLoad: async (instance) => { await instance.render(); }
  };
  setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
  });
</script>
</html>
```

Newuser.html

```
<!DOCTYPE html>
<html>
<head>
            <title>Registration Page</title>
</head>
<style >
            body {
  background-color: pink;
  background-image: url("paper.gif");
```

```
}
.header {
  background-color: rgba(255, 0, 0, 0.3);
  padding: 10px;
  text-align: center;
  text-decoration-color: ;
  font-size: 16px;
}
p{text-align: center;
 font-size: 14;
font-style: italic;
}
</style>
<body>
<form action = "newsubmit", method = "post" >
<header class="header"><h1> DIYA RIDE SHARING</h1></header>
<h1  align = "center" style= "background-color:DodgerBlue;"><p>Right Choice for the Travellers
</p></h1>
<div align="center">
<h3>New Registration </h3>
        <LABEL>First Name : <input type="text" name="Fname" required="" placeholder= "Enter
Your First Name"></LABEL>
        <br><br>

        <LABEL>Last Name : <input type="text" name="Lname" required="" placeholder= "Enter
Your Last Name"></LABEL>

        <br><br>

<LABEL>Designation : <input type="text" name="Designation" required="" placeholder= "Enter
Your Designation"></LABEL>
<br><br>
        <LABEL>Organisation : <input type="text" name="Organisation" required="" placeholder=
"Enter Your Organisation"></LABEL>
<br><br>
        <LABEL>    Email     : <input type="Email" name="email" required="" placeholder= "Enter
Your valid email"></LABEL>
        <br><br>
        <LABEL>Phone           : <input type="text" name="phone" required="" placeholder= "Enter
Your Phone Number"></LABEL>
<br><br>
<LABEL>Password        : </LABEL>
        <input type="password" name="password" required="" placeholder= "Enter password">
        <br><br>
        <a href  = "/newsubmit"><button class = "button">Submit</button>
        </div>
</form>
</body>
<html>
```

Booking.html

```
<!DOCTYPE html>
<html>
<head>
        <title>Book Your Ride</title>
```

```
</head>
<style >
        body {
  background-color: pink;
}
.header {
  background-color: rgba(255, 0, 0, 0.3);
  padding: 10px;
  text-align: center;
  text-decoration-color: ;
  font-size: 16px;
}
.button {
  background-color:#008CBA;
font-size: 12px;
  padding: 15px 32px;
border-radius: 12px;
 display: inline-block;
}
p{text-align: center;
 font-size: 14;
font-style: italic;
}
</style>

<body>
<form action= "book", method="post">
<!--<form action = "book" method = "post" >-->
<header class="header"><h1> DIYA RIDE SHARING</h1></header>
<h1  align = "center" style= "background-color:DodgerBlue;"><p>Right Choice for the Travellers
</p></h1>
<div align="center">
<h3>Book Your Ride {{result}} </h3>
<br>
<br>
        Enter Pick up Location :<input type="text" name= "pickup"  placeholde = "Enter pickup
location">
        <br><br>

        <LABEL>Enter Drop Location : <input type="text" name="drop"  placeholder= "Enter drop
location"></LABEL>
        <br><br>
        <label for="cars">Choose a car:</label>
<select id="cars">
  <option value="Auto" >Auto</option>
  <option value="Indica">Indica</option>
  <option value="Sedan" >Sedan</option>
  <option value="Tavera">Tavera</option>
</select>
<br>
<br>
<br>
   Confirm Your Ride : <a href = "/book"> <button class = "button"/>Yes</button></a>
   <a href = "/cancel"> <button class = "button">No</button></a>
</div>
```

```
</form>
</body>
</html>
```

Display.html

```
<!DOCTYPE html>
<html>
<head>
        <title>Book Your Ride</title>
</head>
<style >
        body {
  background-color: pink;
 }
.header {
  background-color: rgba(255, 0, 0, 0.3);
  padding: 10px;
  text-align: center;
  text-decoration-color: ;
  font-size: 16px;
}
.button {
  background-color:#008CBA;
font-size: 12px;
  padding: 15px 32px;
border-radius: 12px;
 display: inline-block;
}
p{text-align: center;
 font-size: 14;
font-style: italic;
}
</style>
<body>
<form method="post">
<header class="header"><h1> DIYA RIDE SHARING</h1></header>
<h1  align = "center" style= "background-color:DodgerBlue;"><p>Right Choice for the Travellers
</p></h1>
<div align="center">
 <h3>Confirmation Message </h3>
<h3>Pick Up Location : {{result}}</h3>
<h3>Drop Location :  {{d}}</h3>
<h3>Message : {{pred}}</h3>
<h3>Message : {{predic}}</h3>
<Label> Date and Time : {{datetime}}<br><br>
</div>
</form>
</body>
</html>
```

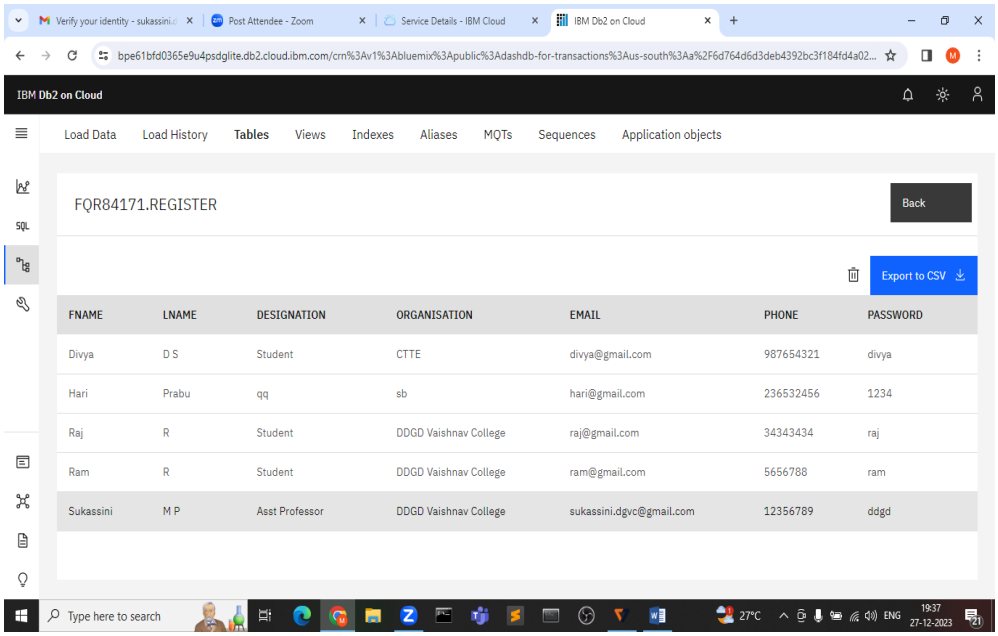


Figure 1 : Structure of register table

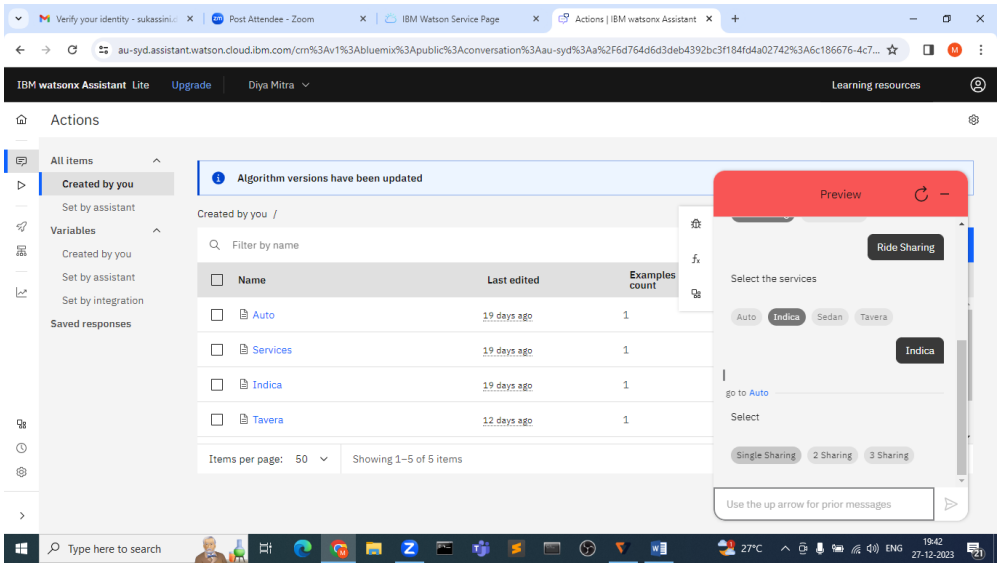


Figure 2: Diya Mitra Virtual Assistant

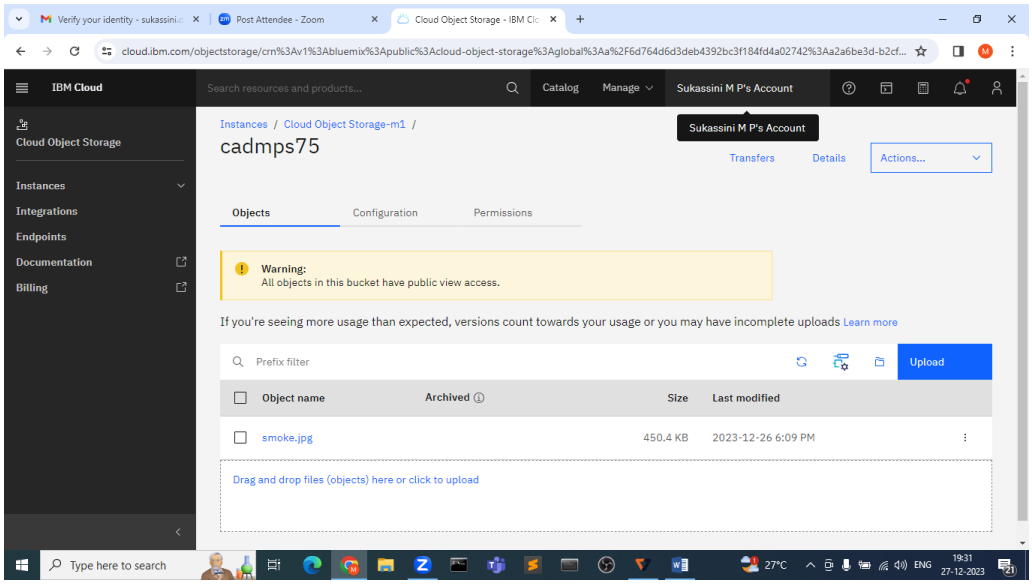


Figure 3: Ibm Cloud Object storage

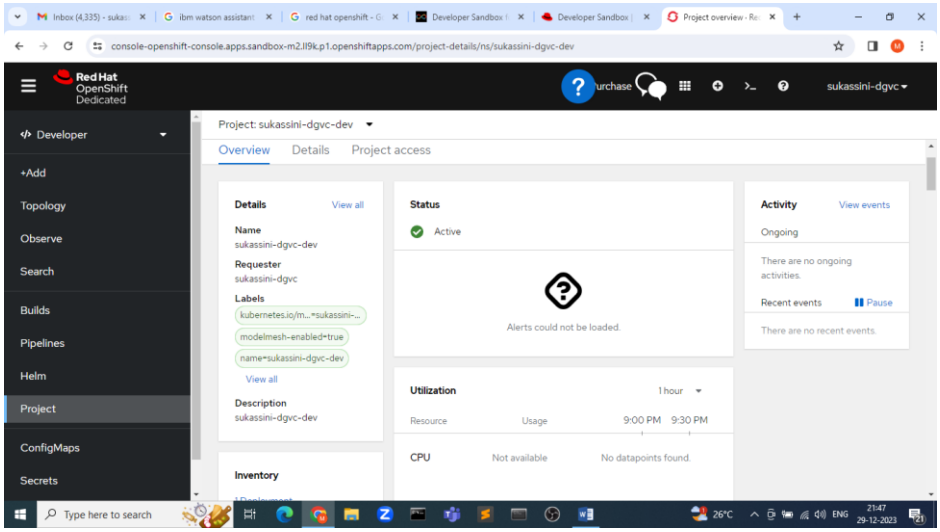


Figure 4: Application Deployment in red hat openshift

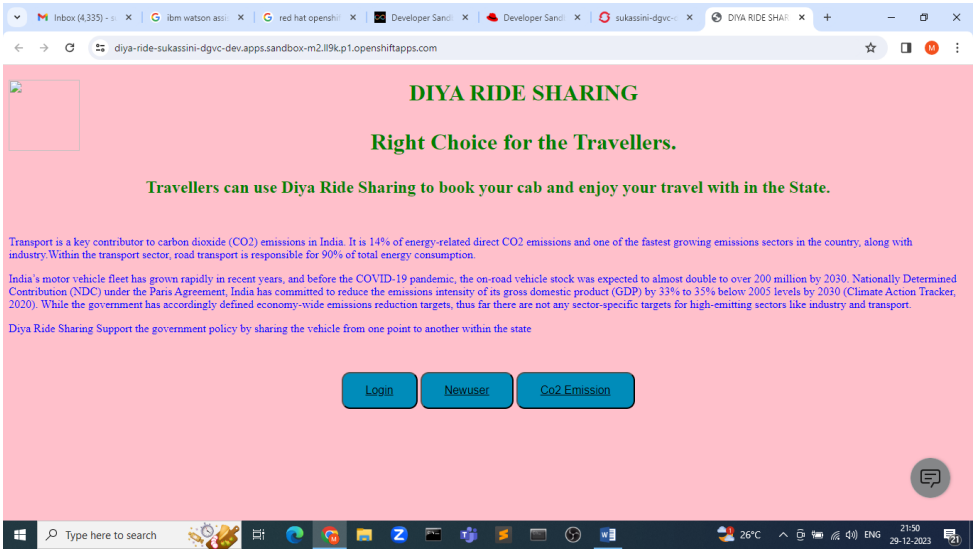


Figure 5: Home page of Diya ride share

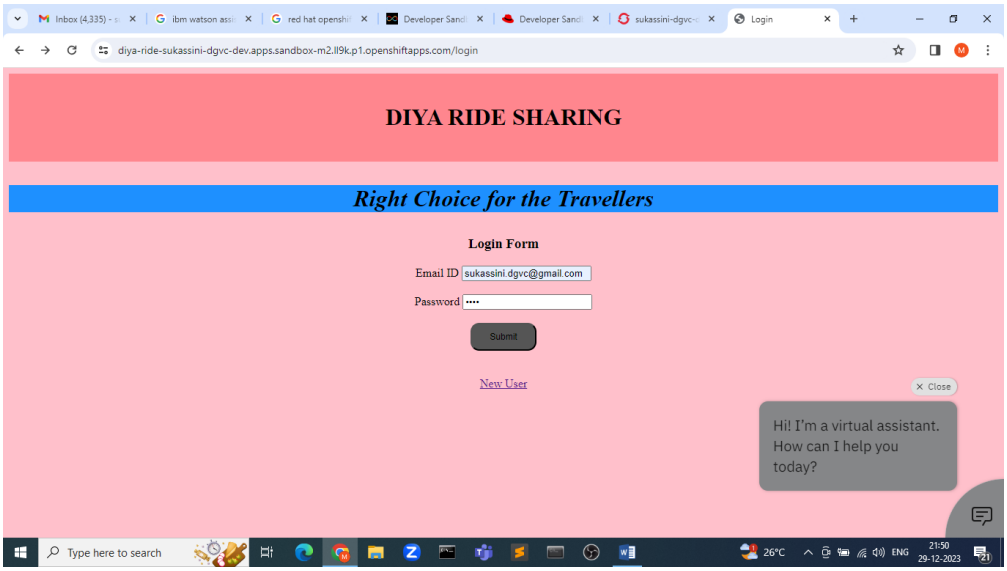


Figure 6: Login page of Diya ride share

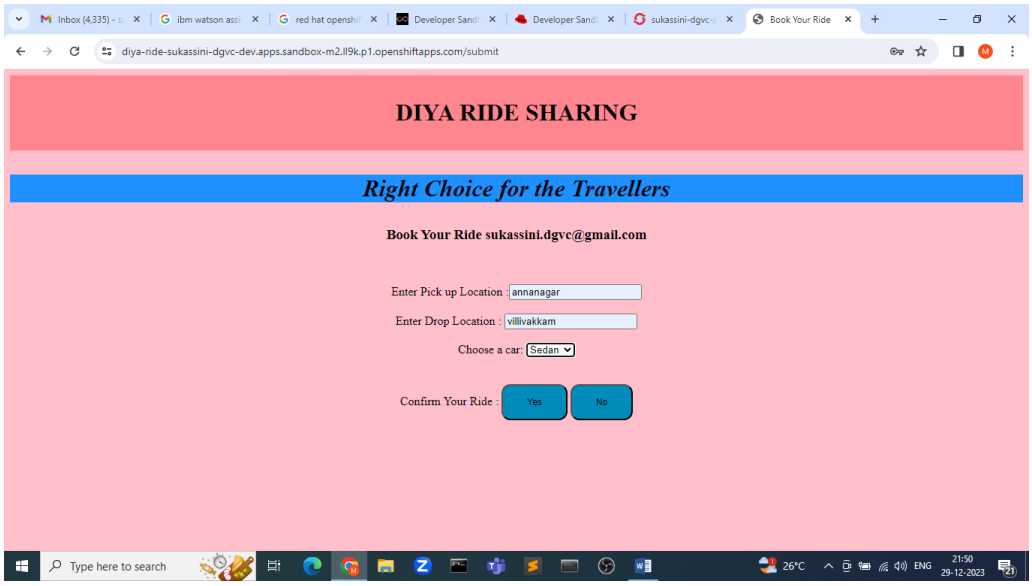


Figure 7: Ride booking page of Diya ride share

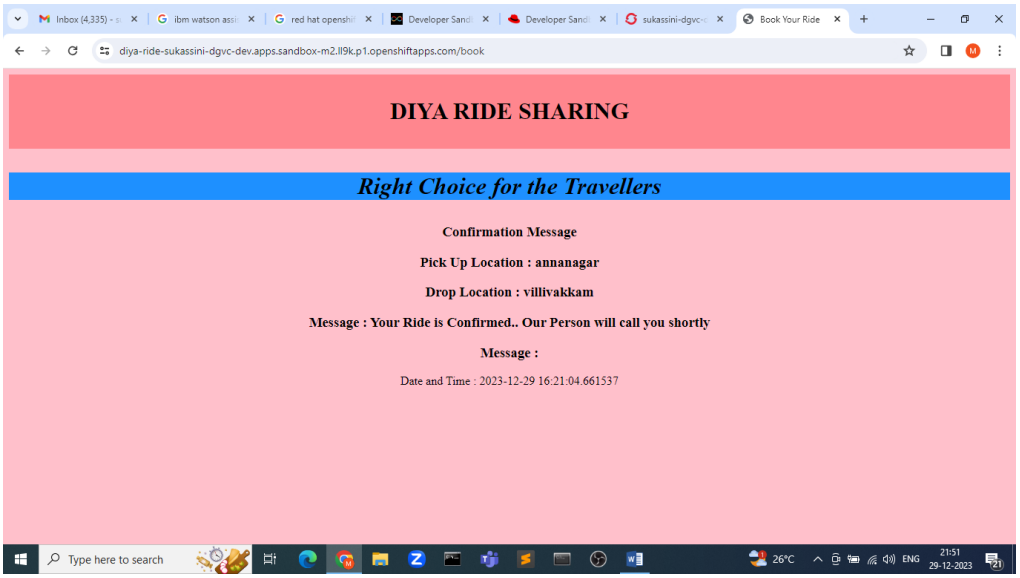


Figure 8: Booking Confirmation page of Diya ride share

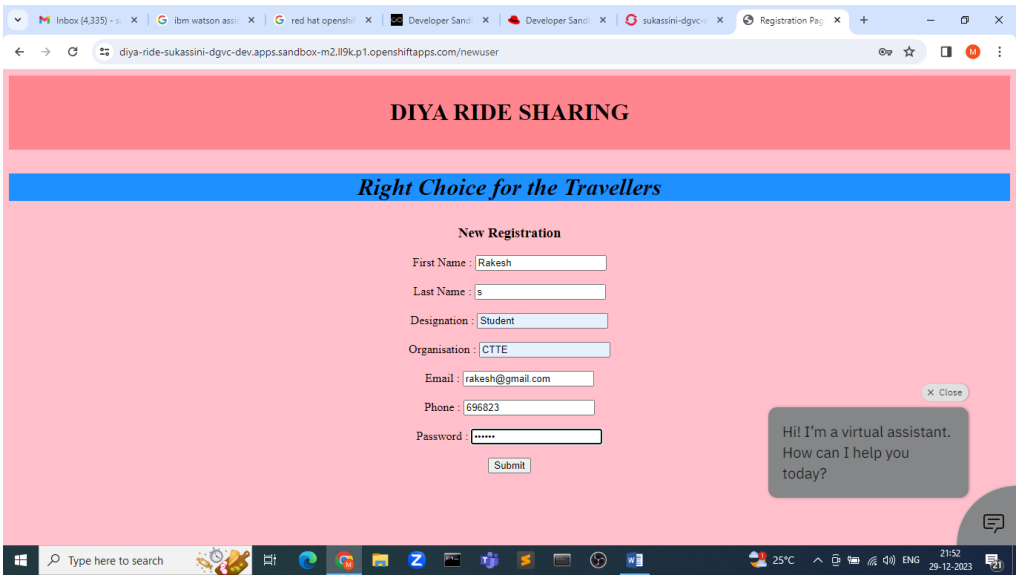


Figure 9: New user registration page of Diya ride share