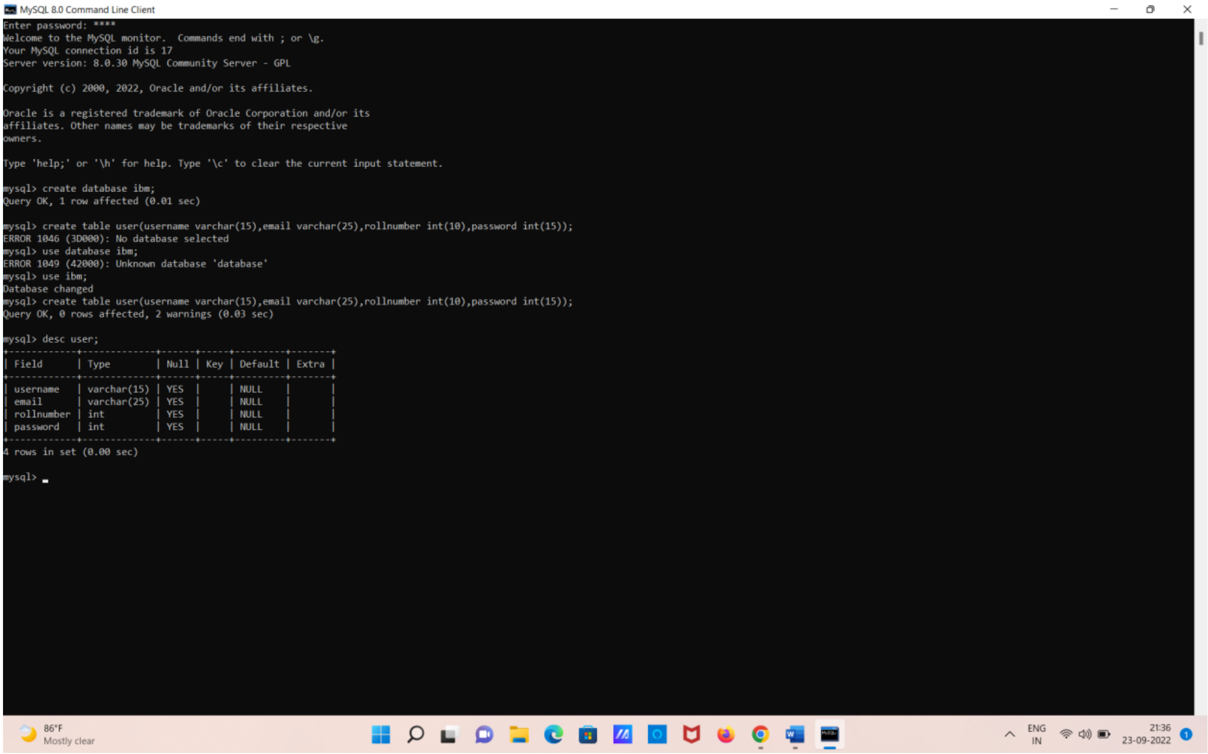
IBM ASSIGNMENT 3

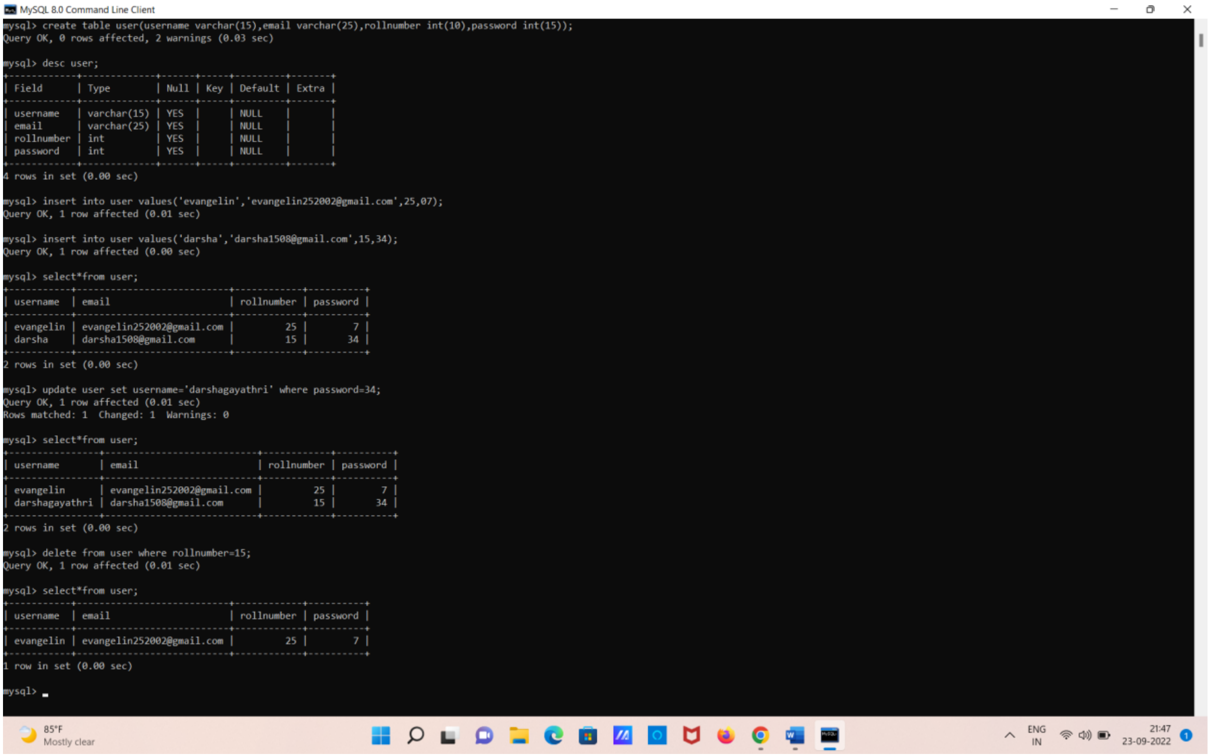
Team ID	**Team ID: PNT2022PMID49593**
Project Name	**Project – PLASMA DONOR APPLICATION**
Name	**C.Evangelin Arockiya Shirley**
Roll No	**950019104701**

1. Create user table with user with email, username, roll number, password.



```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database ibm;
Query OK, 1 row affected (0.01 sec)

mysql> create table user(username varchar(15),email varchar(25),rollnumber int(10),password int(15));
ERROR 1046 (3D000): No database selected
mysql> use database ibm;
ERROR 1049 (42000): Unknown database 'database'
mysql> use ibm;
Database changed
mysql> create table user(username varchar(15),email varchar(25),rollnumber int(10),password int(15));
Query OK, 0 rows affected, 2 warnings (0.03 sec)

mysql> desc user;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| username   | varchar(15) | YES  |     | NULL    |       |
| email      | varchar(25) | YES  |     | NULL    |       |
| rollnumber | int         | YES  |     | NULL    |       |
| password   | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> _
```

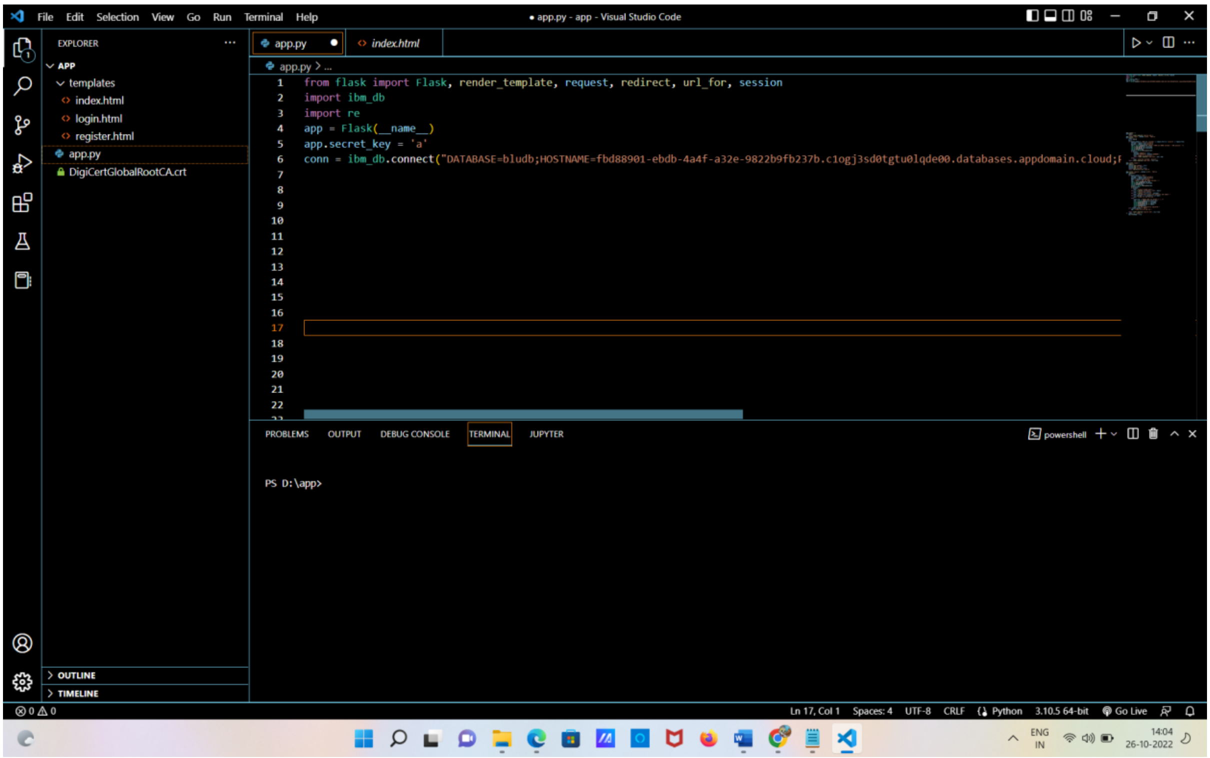
2.Perform UPDATE, DELETE Queries with user table.



```
mysql> create table user(username varchar(15),email varchar(25),rollnumber int(10),password int(15));
Query OK, 0 rows affected, 2 warnings (0.03 sec)

mysql> desc user;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| username   | varchar(15) | YES  |     | NULL    |       |
| email      | varchar(25) | YES  |     | NULL    |       |
| rollnumber | int         | YES  |     | NULL    |       |
| password   | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> insert into user values('evangelin','evangelin252002@gmail.com',25,07);
Query OK, 1 row affected (0.01 sec)

mysql> insert into user values('darsha','darsha1508@gmail.com',15,34);
Query OK, 1 row affected (0.00 sec)

mysql> select*from user;
+-----------+---------------------------+------------+----------+
| username  | email                     | rollnumber | password |
+-----------+---------------------------+------------+----------+
| evangelin | evangelin252002@gmail.com |         25 |        7 |
| darsha    | darsha1508@gmail.com      |         15 |       34 |
+-----------+---------------------------+------------+----------+
2 rows in set (0.00 sec)

mysql> update user set username='darshagayathri' where password=34;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select*from user;
+----------------+---------------------------+------------+----------+
| username       | email                     | rollnumber | password |
+----------------+---------------------------+------------+----------+
| evangelin      | evangelin252002@gmail.com |         25 |        7 |
| darshagayathri | darsha1508@gmail.com      |         15 |       34 |
+----------------+---------------------------+------------+----------+
2 rows in set (0.00 sec)

mysql> delete from user where rollnumber=15;
Query OK, 1 row affected (0.01 sec)

mysql> select*from user;
+-----------+---------------------------+------------+----------+
| username  | email                     | rollnumber | password |
+-----------+---------------------------+------------+----------+
| evangelin | evangelin252002@gmail.com |         25 |        7 |
+-----------+---------------------------+------------+----------+
1 row in set (0.00 sec)

mysql> _
```

3.Connect python code to db2.

App.py:



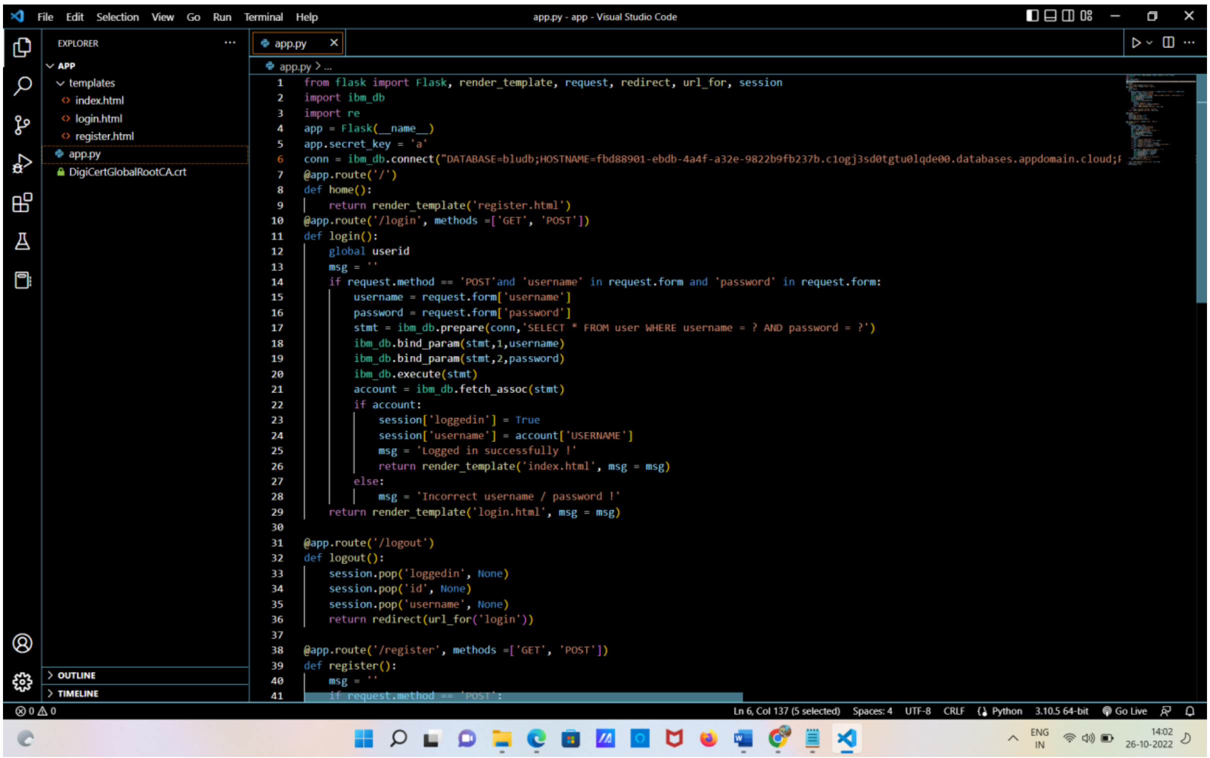
File Edit Selection View Go Run Terminal Help
• app.py - app - Visual Studio Code
EXPLORER
APP
templates
index.html
login.html
register.html
app.py
DigiCertGlobalRootCA.crt
app.py
index.html
app.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
powershell
OUTLINE
TIMELINE
Ln 17, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.5 64-bit Go Live
ENG IN 14:04 26-10-2022

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;
```

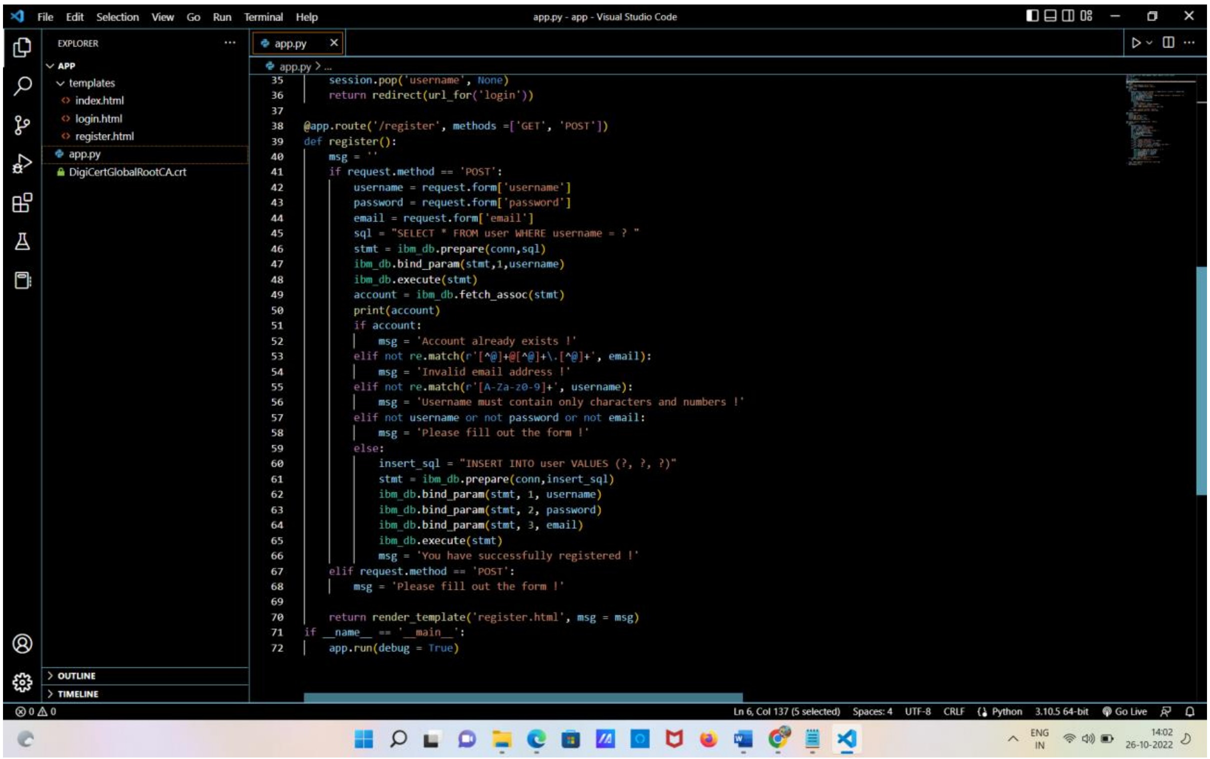
```
PS D:\app>
```

4. Create a flask app with registration page, login page and welcome page. By default load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user username and password. If the user is valid show the welcome page.

App.py:

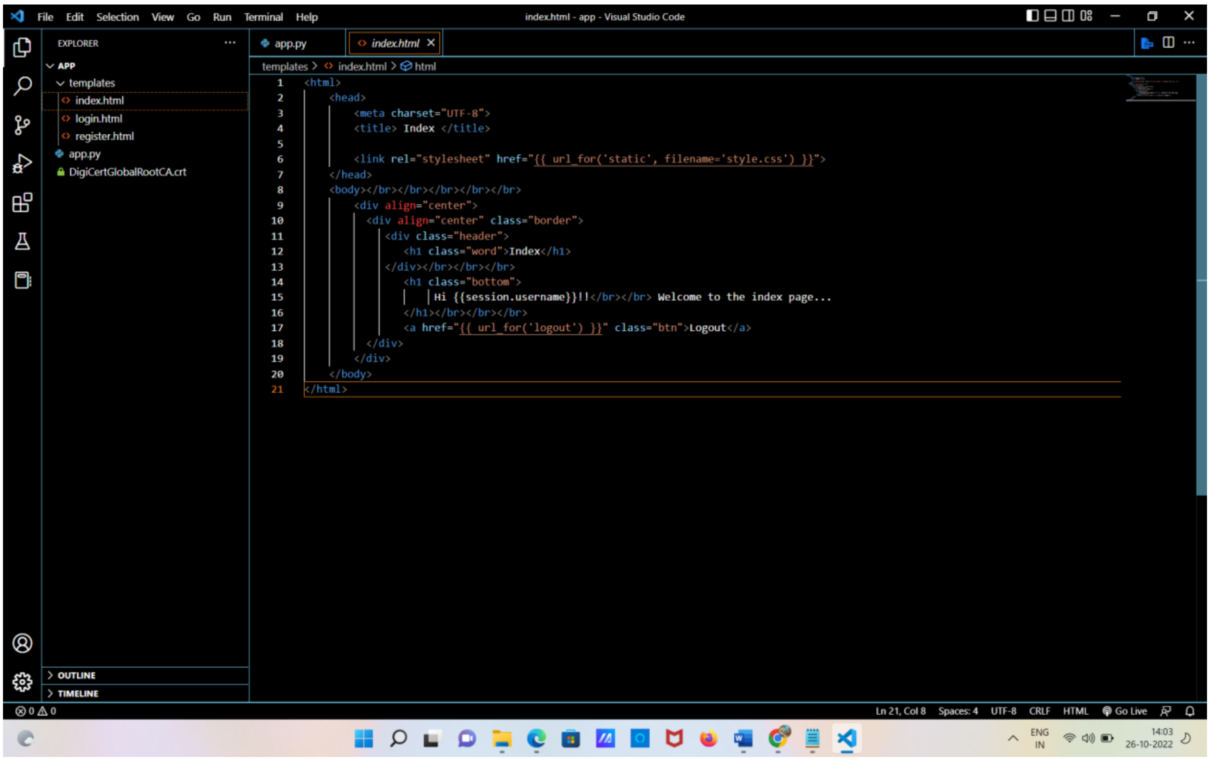

File Edit Selection View Go Run Terminal Help
app.py - app - Visual Studio Code
EXPLORER
APP
templates
index.html
login.html
register.html
app.py
DigiCertGlobalRootCA.crt
OUTLINE
TIMELINE
app.py
app.py > ...
Ln 6, Col 137 (5 selected) Spaces: 4 UTF-8 CRLF Python 3.10.5 64-bit Go Live
ENG IN
14:02
26-10-2022

```python
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;
@app.route('/')
def home():
    return render_template('register.html')
@app.route('/login', methods =['GET', 'POST'])
def login():
    global userid
    msg = ''
    if request.method == 'POST'and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        stmt = ibm_db.prepare(conn,'SELECT * FROM user WHERE username = ? AND password = ?')
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            session['loggedin'] = True
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'
            return render_template('index.html', msg = msg)
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect(url_for('login'))

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
```


File Edit Selection View Go Run Terminal Help
app.py - app - Visual Studio Code
EXPLORER
APP
templates
index.html
login.html
register.html
app.py
DigiCertGlobalRootCA.crt
OUTLINE
TIMELINE
app.py
app.py > ...
Ln 6, Col 137 (5 selected) Spaces: 4 UTF-8 CRLF Python 3.10.5 64-bit Go Live
ENG IN
14:02
26-10-2022

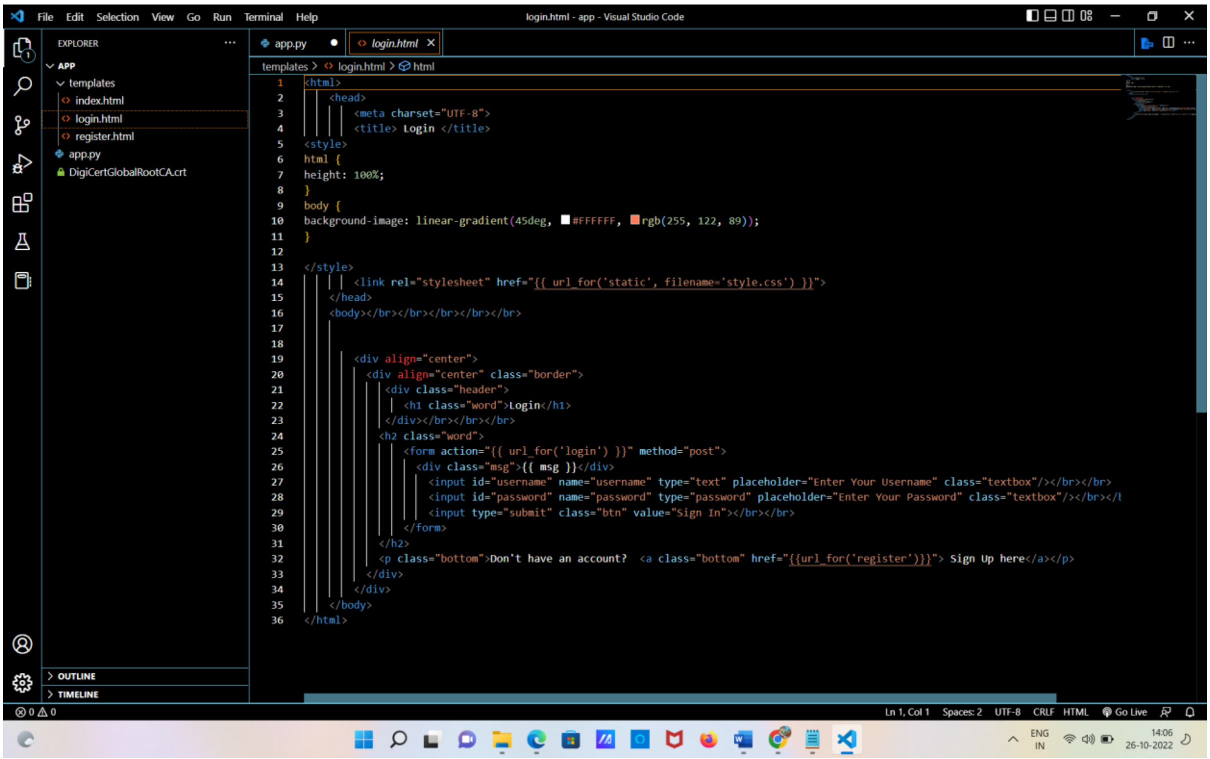
```python
    session.pop('username', None)
    return redirect(url_for('login'))

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        email = request.form['email']
        sql = "SELECT * FROM user WHERE username = ? "
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'Username must contain only characters and numbers !'
        elif not username or not password or not email:
            msg = 'Please fill out the form !'
        else:
            insert_sql = "INSERT INTO user VALUES (?, ?, ?)"
            stmt = ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, password)
            ibm_db.bind_param(stmt, 3, email)
            ibm_db.execute(stmt)
            msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'

    return render_template('register.html', msg = msg)
if __name__ == '__main__':
    app.run(debug = True)
```

Index.html:


index.html - app - Visual Studio Code
app.py
index.html
templates > index.html > html

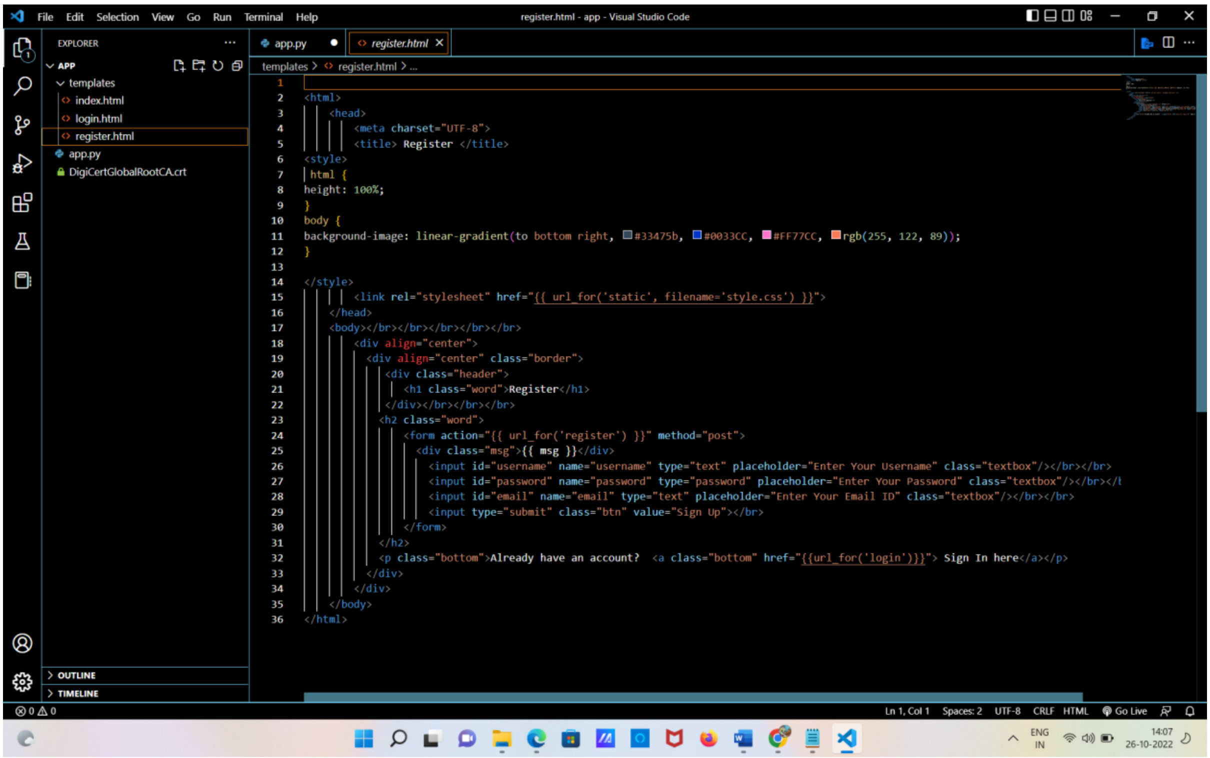
```html
<html>
    <head>
        <meta charset="UTF-8">
        <title> Index </title>

        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
    </head>
    <body></br></br></br></br></br>
        <div align="center">
          <div align="center" class="border">
             <div class="header">
                <h1 class="word">Index</h1>
             </div></br></br></br>
                <h1 class="bottom">
                    Hi {{session.username}}!!</br></br> Welcome to the index page...
                </h1></br></br></br>
                <a href="{{ url_for('logout') }}" class="btn">Logout</a>
          </div>
        </div>
    </body>
</html>
```

Login.html:


login.html - app - Visual Studio Code
app.py
login.html
templates > login.html > html

```html
<html>
    <head>
        <meta charset="UTF-8">
        <title> Login </title>
<style>
html {
height: 100%;
}
body {
background-image: linear-gradient(45deg, #FFFFFF, rgb(255, 122, 89));
}

</style>
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
    </head>
    <body></br></br></br></br></br>


        <div align="center">
          <div align="center" class="border">
             <div class="header">
                <h1 class="word">Login</h1>
             </div></br></br></br>
            <h2 class="word">
                <form action="{{ url_for('login') }}" method="post">
                  <div class="msg">{{ msg }}</div>
                    <input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox"/></br></br>
                    <input id="password" name="password" type="password" placeholder="Enter Your Password" class="textbox"/></br></
                    <input type="submit" class="btn" value="Sign In"></br></br>
                </form>
            </h2>
            <p class="bottom">Don't have an account?  <a class="bottom" href="{{url_for('register')}}"> Sign Up here</a></p>
          </div>
        </div>
    </body>
</html>
```

Register.html:

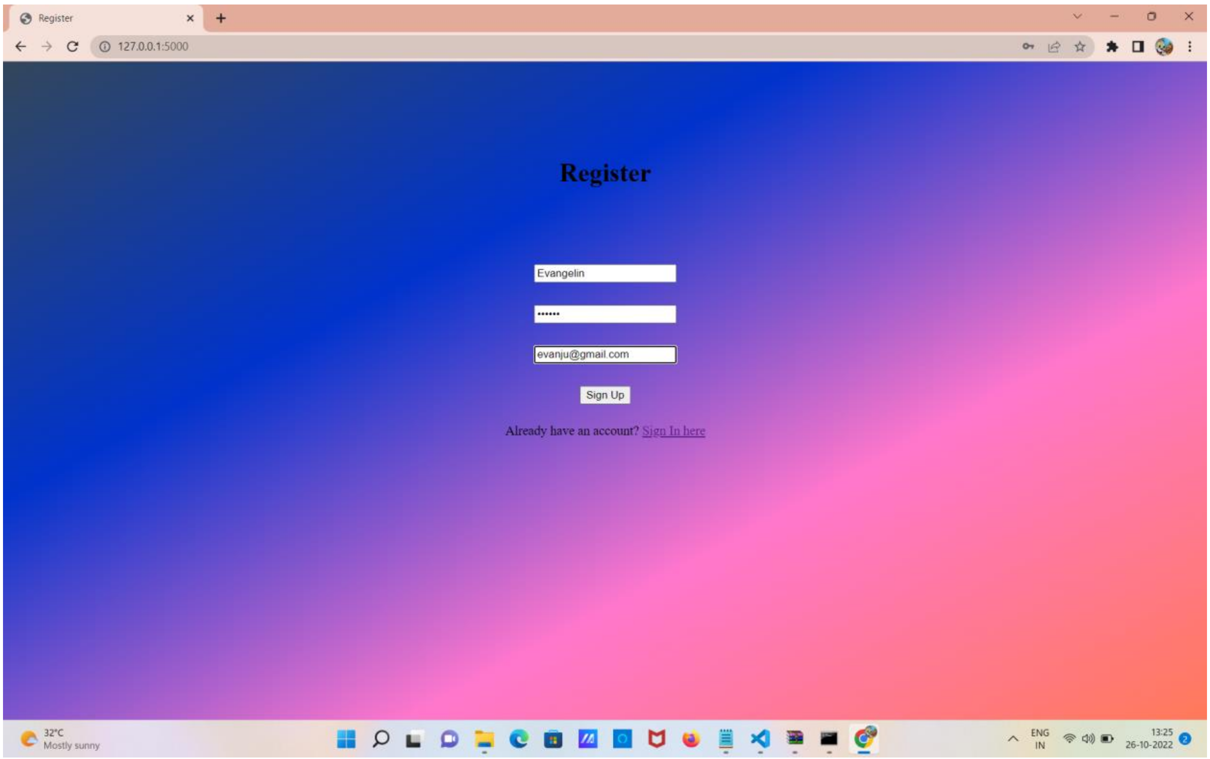


```html

<html>
    <head>
        <meta charset="UTF-8">
        <title> Register </title>
<style>
 html {
height: 100%;
}
body {
background-image: linear-gradient(to bottom right, #33475b, #0033CC, #FF77CC, rgb(255, 122, 89));
}

</style>
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
    </head>
    <body></br></br></br></br></br>
        <div align="center">
          <div align="center" class="border">
            <div class="header">
              <h1 class="word">Register</h1>
            </div></br></br></br>
            <h2 class="word">
              <form action="{{ url_for('register') }}" method="post">
                <div class="msg">{{ msg }}</div>
                  <input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox"/></br></br>
                  <input id="password" name="password" type="password" placeholder="Enter Your Password" class="textbox"/></br></
                  <input id="email" name="email" type="text" placeholder="Enter Your Email ID" class="textbox"/></br></br>
                  <input type="submit" class="btn" value="Sign Up"></br>
              </form>
            </h2>
            <p class="bottom">Already have an account?  <a class="bottom" href="{{url_for('login')}}"> Sign In here</a></p>
          </div>
        </div>
    </body>
</html>
```

File Edit Selection View Go Run Terminal Help
register.html - app - Visual Studio Code
EXPLORER
APP
templates
index.html
login.html
register.html
app.py
DigiCertGlobalRootCA.crt
app.py
register.html
templates > register.html > ...
OUTLINE
TIMELINE
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF HTML Go Live
ENG IN
14:07
26-10-2022

OUTPUT:

Registering page:



Register

127.0.0.1:5000

Register

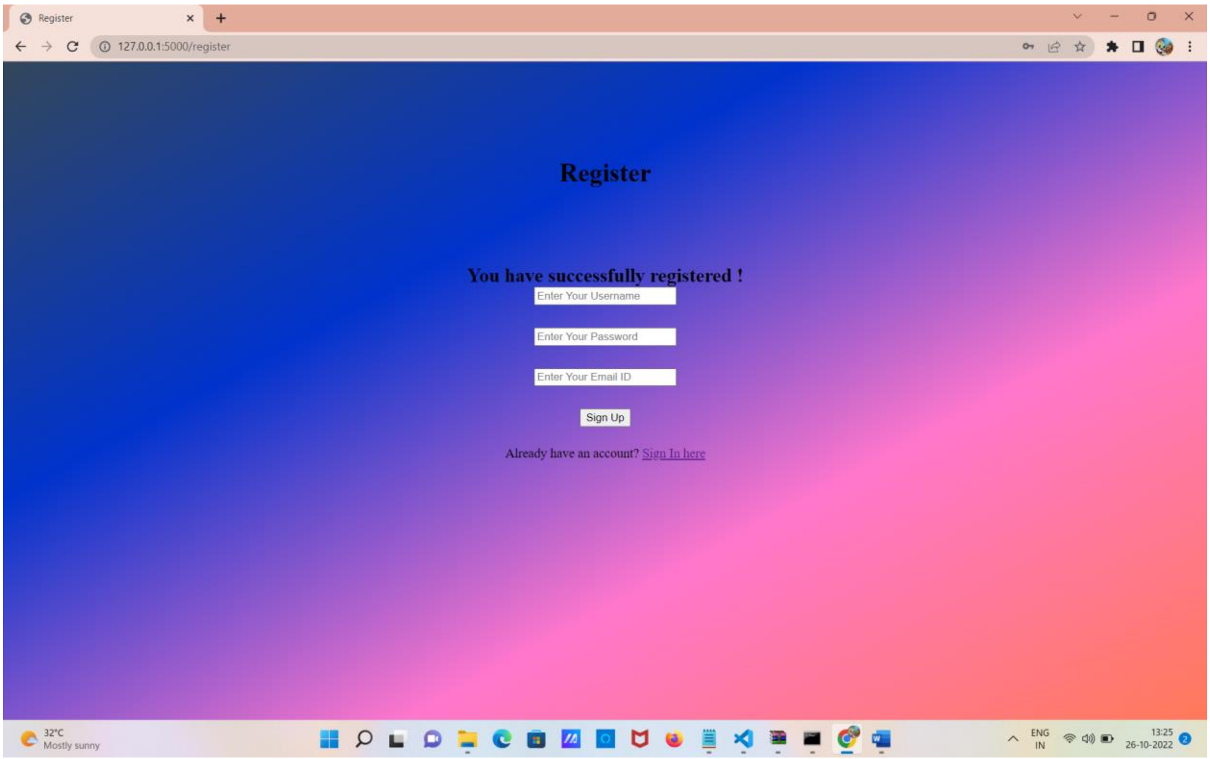
Evangelin

••••••

evanju@gmail.com

Sign Up

Already have an account? Sign In here



Register

127.0.0.1:5000/register

Register

You have successfully registered !

Enter Your Username

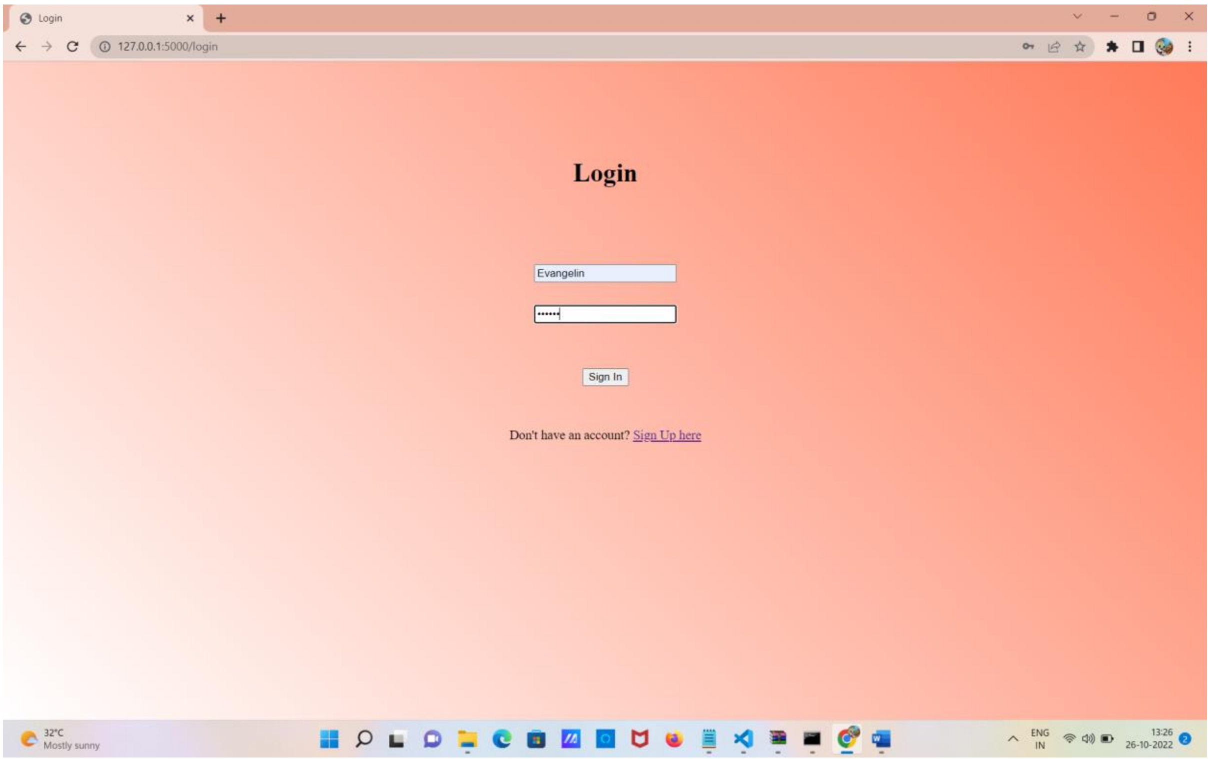
Enter Your Password

Enter Your Email ID

Sign Up

Already have an account? Sign In here

Login page:



Login

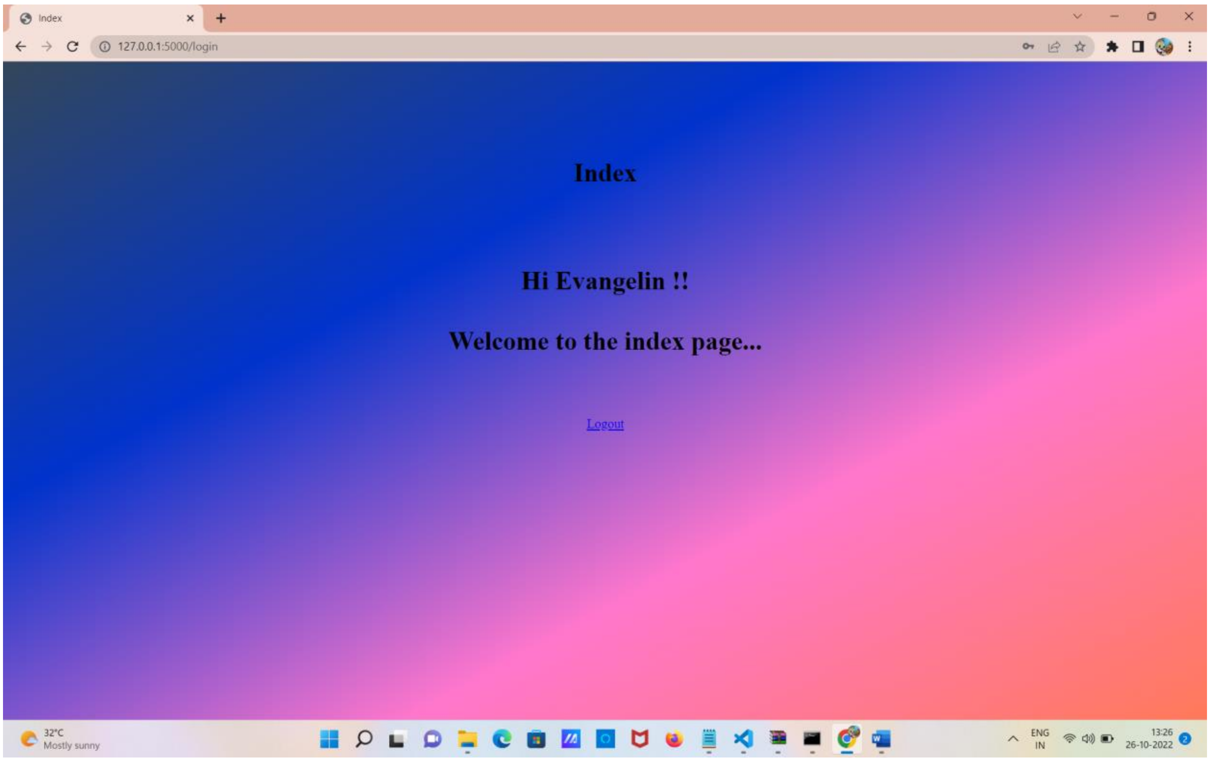
Evangelin

••••••

Sign In

Don't have an account? Sign Up here

Index page:



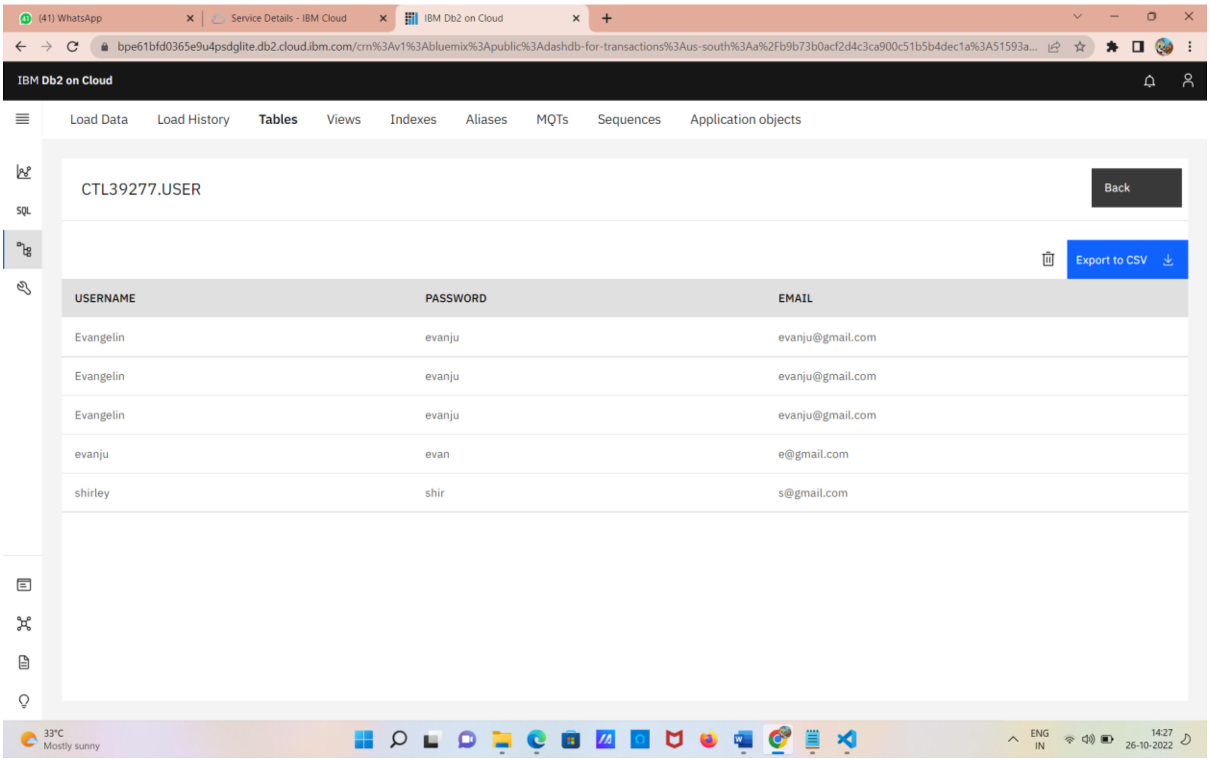
Index

Hi Evangelin !!

Welcome to the index page...

Logout

Database:


(41) WhatsApp
Service Details - IBM Cloud
IBM Db2 on Cloud
bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fb9b73b0acf2d4c3ca900c51b5b4dec1a%3A51593a...
IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
SQL
Back
Export to CSV
33°C
Mostly sunny
ENG
IN
14:27
26-10-2022

CTL39277.USER

USERNAME	PASSWORD	EMAIL
Evangelin	evanju	evanju@gmail.com
Evangelin	evanju	evanju@gmail.com
Evangelin	evanju	evanju@gmail.com
evanju	evan	e@gmail.com
shirley	shir	s@gmail.com