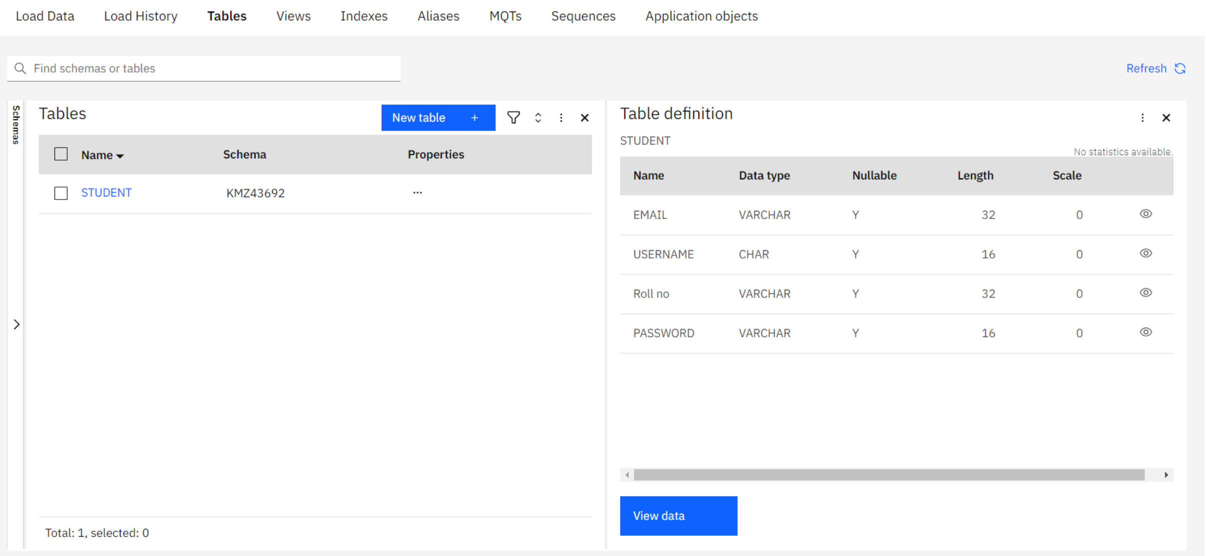
Assignment -2

Flask Program - connected with IBM DB

Assignment Date	19 September 2022
Student Name	Mr. Nithinrhaj S
Student Roll Number	2019115062
Maximum Marks	2 Marks

Q1 .Create User table with user with email, username, roll number, password.



Tables

☐	Name	Schema	Properties
☐	STUDENT	KMZ43692	···

Total: 1, selected: 0

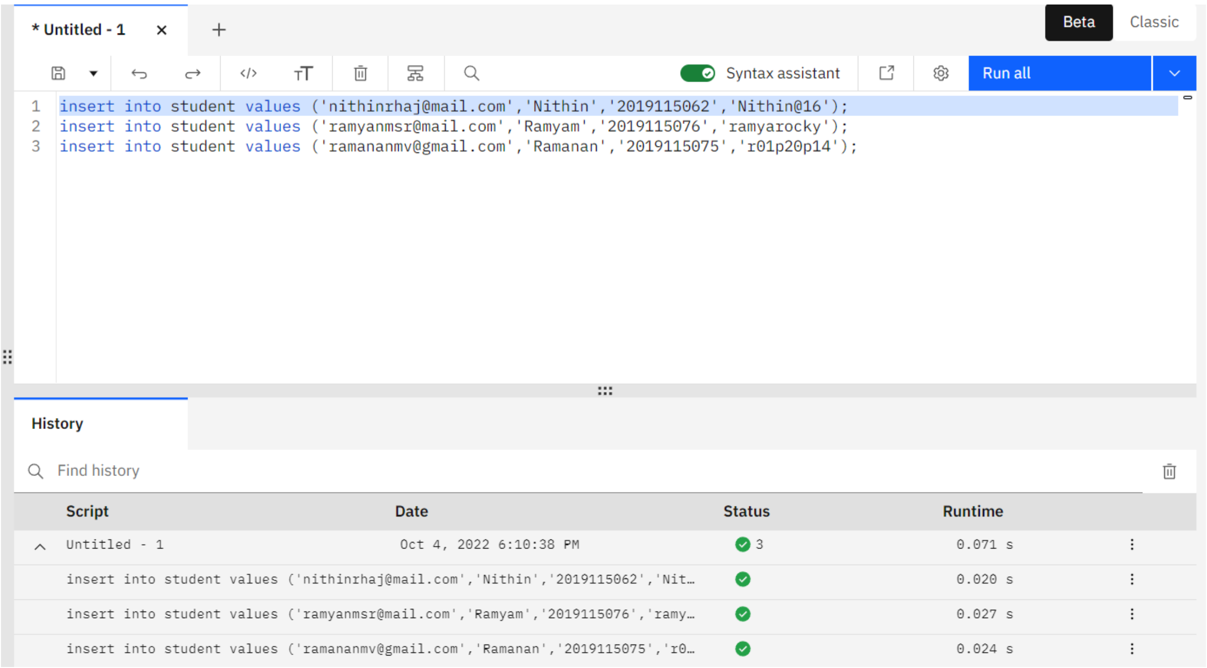
Table definition

STUDENT

No statistics available

Name	Data type	Nullable	Length	Scale
EMAIL	VARCHAR	Y	32	0
USERNAME	CHAR	Y	16	0
Roll no	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	16	0

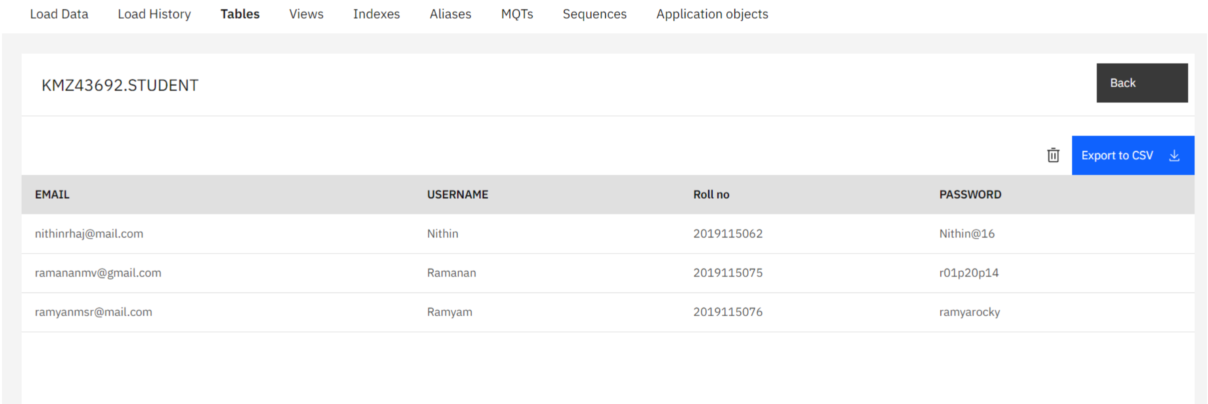
Inserting values into the table:



```
insert into student values ('nithinrhaj@mail.com','Nithin','2019115062','Nithin@16');
insert into student values ('ramyanmsr@mail.com','Ramyam','2019115076','ramyarocky');
insert into student values ('ramananmv@gmail.com','Ramanan','2019115075','r01p20p14');
```

Script	Date	Status	Runtime
Untitled - 1	Oct 4, 2022 6:10:38 PM	✓ 3	0.071 s
insert into student values ('nithinrhaj@mail.com','Nithin','2019115062','Nit…		✓	0.020 s
insert into student values ('ramyanmsr@mail.com','Ramyam','2019115076','ramy…		✓	0.027 s
insert into student values ('ramananmv@gmail.com','Ramanan','2019115075','r0…		✓	0.024 s

Data stored in the table Student:

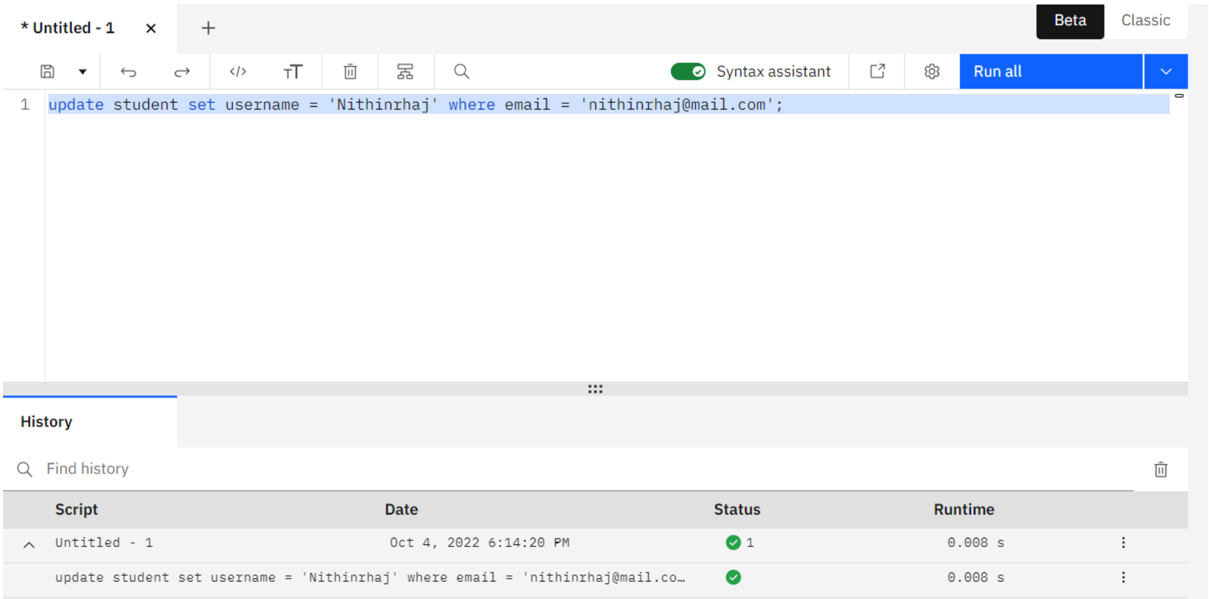


KMZ43692.STUDENT

EMAIL	USERNAME	Roll no	PASSWORD
nithinrhaj@mail.com	Nithin	2019115062	Nithin@16
ramananmv@gmail.com	Ramanan	2019115075	r01p20p14
ramyanmsr@mail.com	Ramyam	2019115076	ramyarocky

Q2 . Perform UPDATE,DELETE Queries with user table

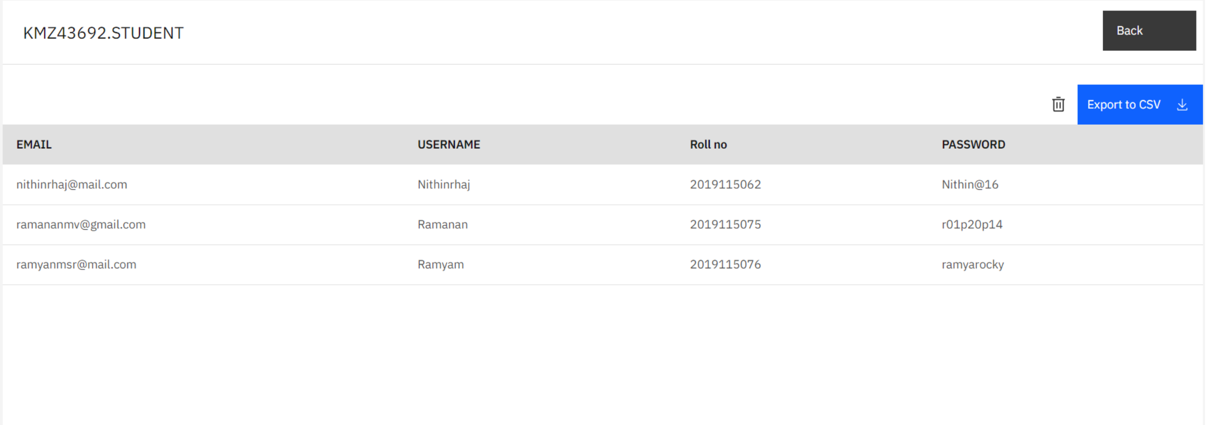
1. Updating username of entry where email is ‘nithinrhaj@gmail.com’



```
update student set username = 'Nithinrhaj' where email = 'nithinrhaj@mail.com';
```

	Script	Date	Status	Runtime
^	Untitled - 1	Oct 4, 2022 6:14:20 PM	✓ 1	0.008 s
	update student set username = 'Nithinrhaj' where email = 'nithinrhaj@mail.co…		✓	0.008 s

Updated result in database:

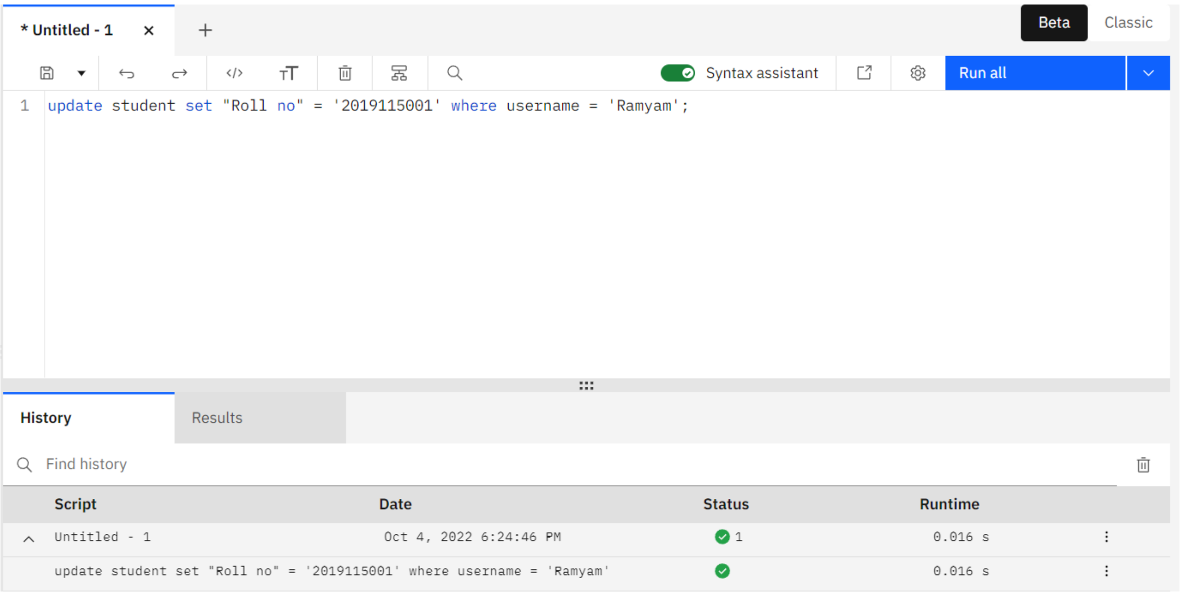


KMZ43692.STUDENT

EMAIL	USERNAME	Roll no	PASSWORD
nithinrhaj@mail.com	Nithinrhaj	2019115062	Nithin@16
ramananmv@gmail.com	Ramanan	2019115075	r01p20p14
ramyanmsr@mail.com	Ramyam	2019115076	ramyarocky

[The username has been changed to Nithinrhaj.]

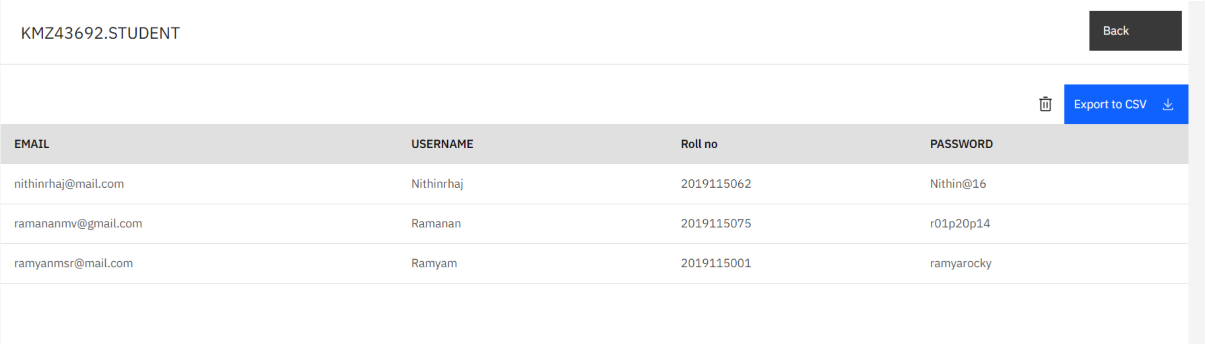
2. Updating the roll number of entries where the username is ‘Ramyam’.



```
update student set "Roll no" = '2019115001' where username = 'Ramyam';
```

Script	Date	Status	Runtime
Untitled - 1	Oct 4, 2022 6:24:46 PM	✓ 1	0.016 s
update student set "Roll no" = '2019115001' where username = 'Ramyam'		✓	0.016 s

Updated result in database:

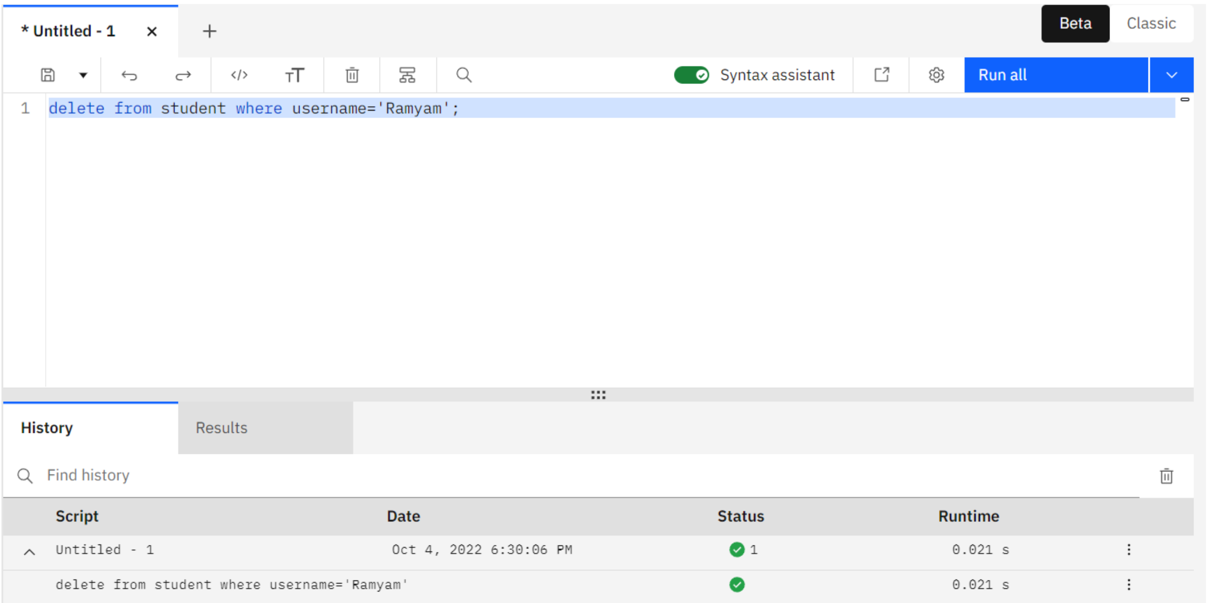


KMZ43692.STUDENT

EMAIL	USERNAME	Roll no	PASSWORD
nithinrhaj@mail.com	Nithinrhaj	2019115062	Nithin@16
ramananmv@gmail.com	Ramanan	2019115075	r01p20p14
ramyanmsr@mail.com	Ramyam	2019115001	ramyarocky

[The Roll no of entry with username ‘Ramyam’ is changed.]

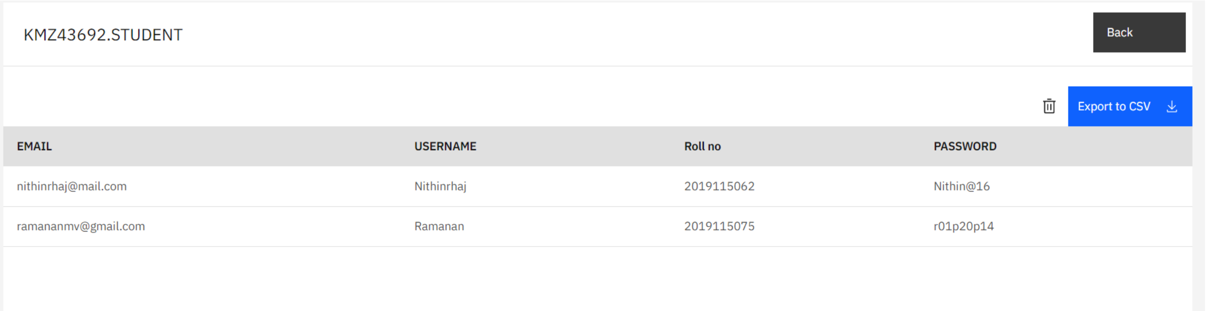
3. Delete operation



```
delete from student where username='Ramyam';
```

Script	Date	Status	Runtime
Untitled - 1	Oct 4, 2022 6:30:06 PM	✓ 1	0.021 s
delete from student where username='Ramyam'		✓	0.021 s

Updated result in database:



KMZ43692.STUDENT

EMAIL	USERNAME	Roll no	PASSWORD
nithinrhaj@mail.com	Nithinrhaj	2019115062	Nithin@16
ramananmv@gmail.com	Ramanan	2019115075	r01p20p14

[The row is deleted from the database.]

3. Connecting python to database:

Solution:

```
from flask import
Flask,render_template,request,redirect,url_for,session
import ibm_db
import re

app=Flask(__name__)
app.secret_key='a'
```

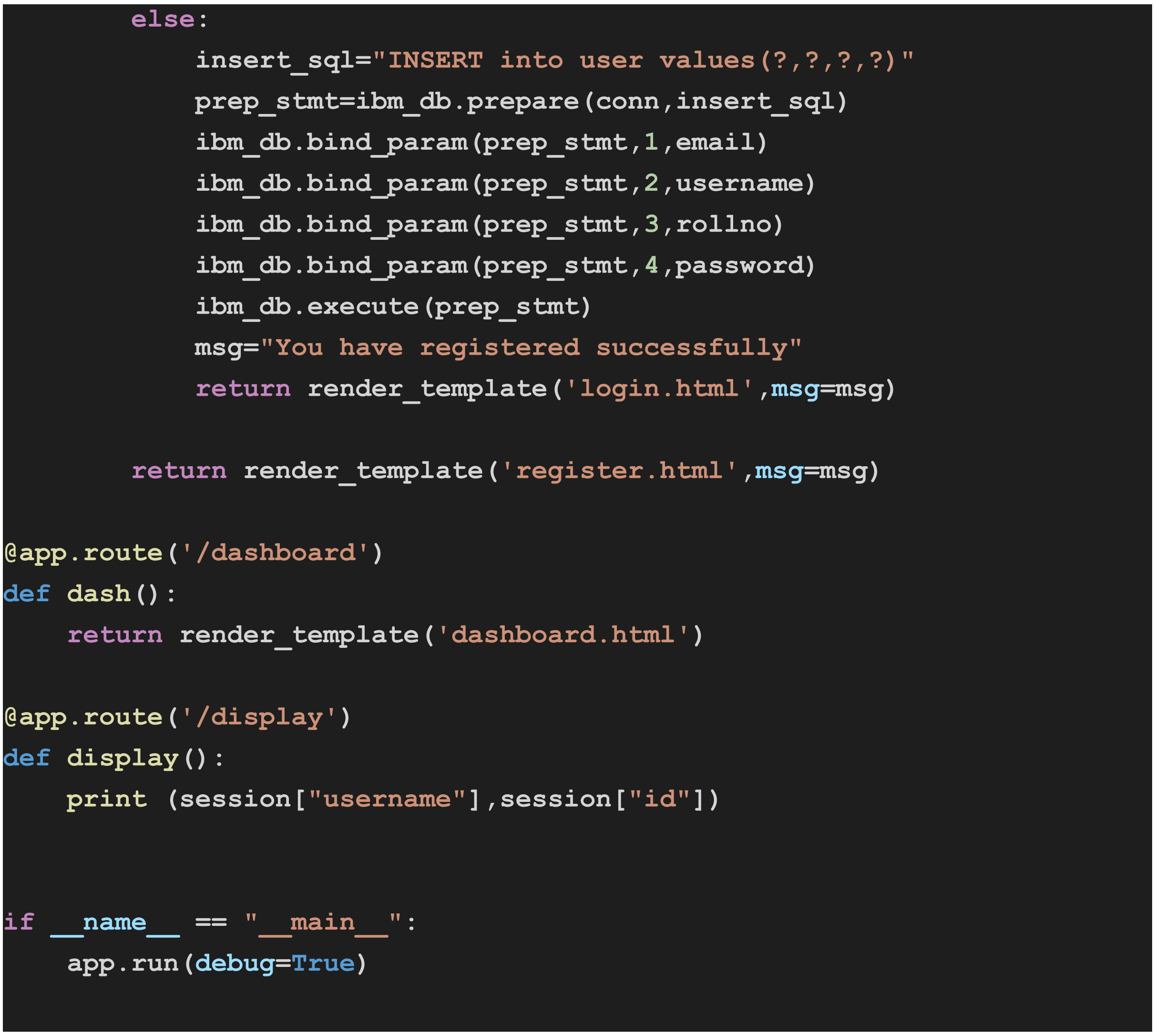
```
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=b70af05b-76e4-4bca-a1f5-23
dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;Se
curity=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=dgp98488;P
WD=T4ZhPsFfVgztgO7H;","","")
```

Q4. 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

Solution:

CODE:

```
#!/usr/bin/env 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=b70af05b-76e4-4bca-a1f5-23
dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;Se
curity=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=dgp98488;P
WD=T4ZhPsFfVgztgO7H;","","")

@app.route('/')
def home():
    return render_template('register.html')

@app.route("/login",methods=['GET','POST'])
def login():
    global userid
    msg=" "

    if request.method=="POST":
        username=request.form['username']
        password=request.form['password']
        sql="SELECT * from user where username=? AND password=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin']=True
            session['id']=account['USERNAME']
            userid=account['USERNAME']
            session['username']=account['USERNAME']
            msg='Logged in successfully!!'
            return render_template("dashboard.html",username=request.form['username'],msg=msg)
        else:
            msg="Incorrect Username/Password"
    return render_template('login.html',msg=msg)

@app.route("/register",methods=["GET","POST"])
def register():
    msg=" "
    if request.method=="POST":
        username=request.form['username']
        email=request.form['email']
        rollno=request.form['rollno']
        password=request.form['password']
        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-Z]|[0-9])+',username):
            msg="Name must contain only characters and numbers!"
```



```
        else:
            insert_sql="INSERT into user values(?,?,?,?)"
            prep_stmt=ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(prep_stmt,1,email)
            ibm_db.bind_param(prep_stmt,2,username)
            ibm_db.bind_param(prep_stmt,3,rollno)
            ibm_db.bind_param(prep_stmt,4,password)
            ibm_db.execute(prep_stmt)
            msg="You have registered successfully"
            return render_template('login.html',msg=msg)

        return render_template('register.html',msg=msg)

@app.route('/dashboard')
def dash():
    return render_template('dashboard.html')

@app.route('/display')
def display():
    print (session["username"],session["id"])


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

Dashbord.html:

```
<!doctype html>
<html>
  <head>
    <title>Dashboard</title>
  </head>
  <body>
    <script>
      if("{{msg}}"!="")
      {
    window.alert("{{msg}}")}
      </script>
    Welcome to the dashboard {{username}} !!
  </body>
```

```
</html>
```

Login.html:

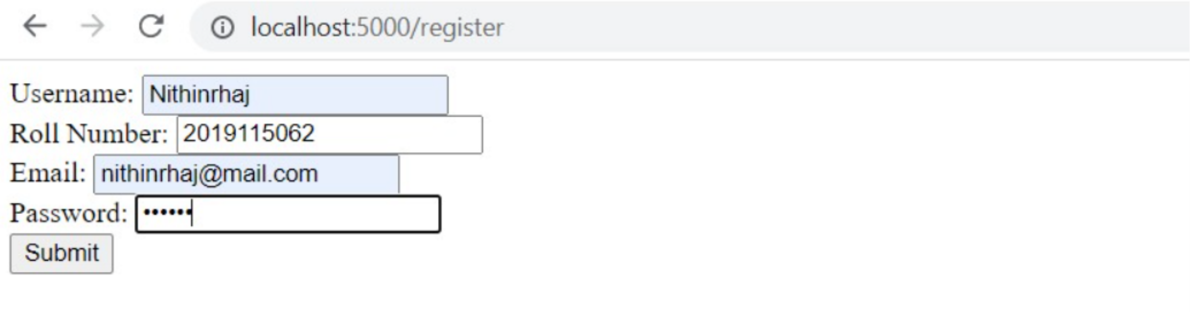
```
<!doctype html>
<html>
  <head>
    <title>Login Page</title>
  </head>
  <body>
    <script>
      if("{{msg}}"!="")
      {
    window.alert("{{msg}}")}
    </script>
    <form action="/login" method="POST">
      <div>
        <div>
          <label for="username">Username: </label>
          <input type="text" name="username" id="username" />

        </div>
        <div>
          <label for="password">Password: </label>
          <input type="password" name="password" id="password" />
        </div>
        <div>
          <input type="submit" value="Submit" />
        </div>
      </div>
    </form>
  </body>
</html>
```

Registration.html:

```
<!doctype html>
<html>
  <head>
    <title>Registeration Page</title>
  </head>
  <body>
    <script>
     if("{{msg}}"!="")
      {
    window.alert("{{msg}}")}
      </script>
    <form action="/register" method="POST">
      <div>
        <div>
          <label for="username">Username: </label>
          <input type="text" name="username" id="username" />

        </div>
        <div>
          <label for="rollno">Roll Number: </label>
          <input type="number" name="rollno" id="rollno" />
        </div>
        <div>
          <label for="email">Email: </label>
          <input type="email" name="email" id="email" />
        </div>
        <div>
          <label for="password">Password: </label>
          <input type="password" name="password" id="password" />
        </div>
        <div>
          <input type="submit" value="Submit" />
        </div>
      </div>
    </form>
  </body>
</html>
```

Outputs:

- **An pre-existing user trying to sign in:**



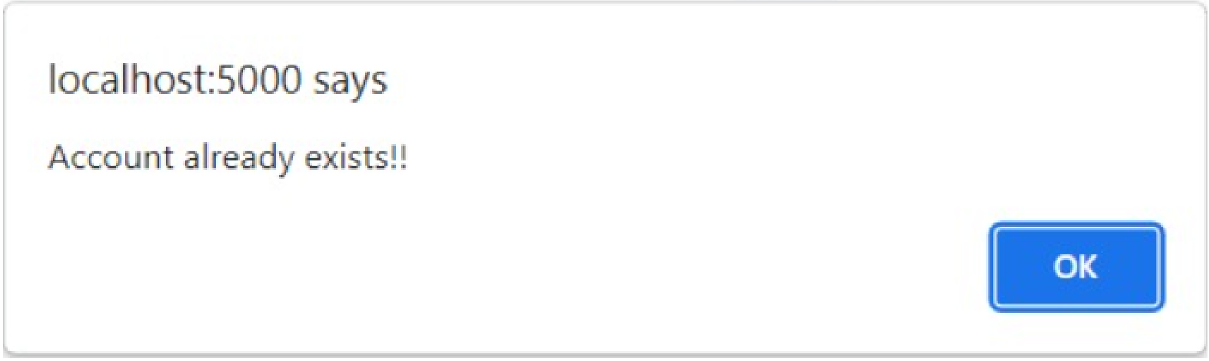
localhost:5000/register

Username: Nithinrhaj

Roll Number: 2019115062

Email: nithinrhaj@mail.com

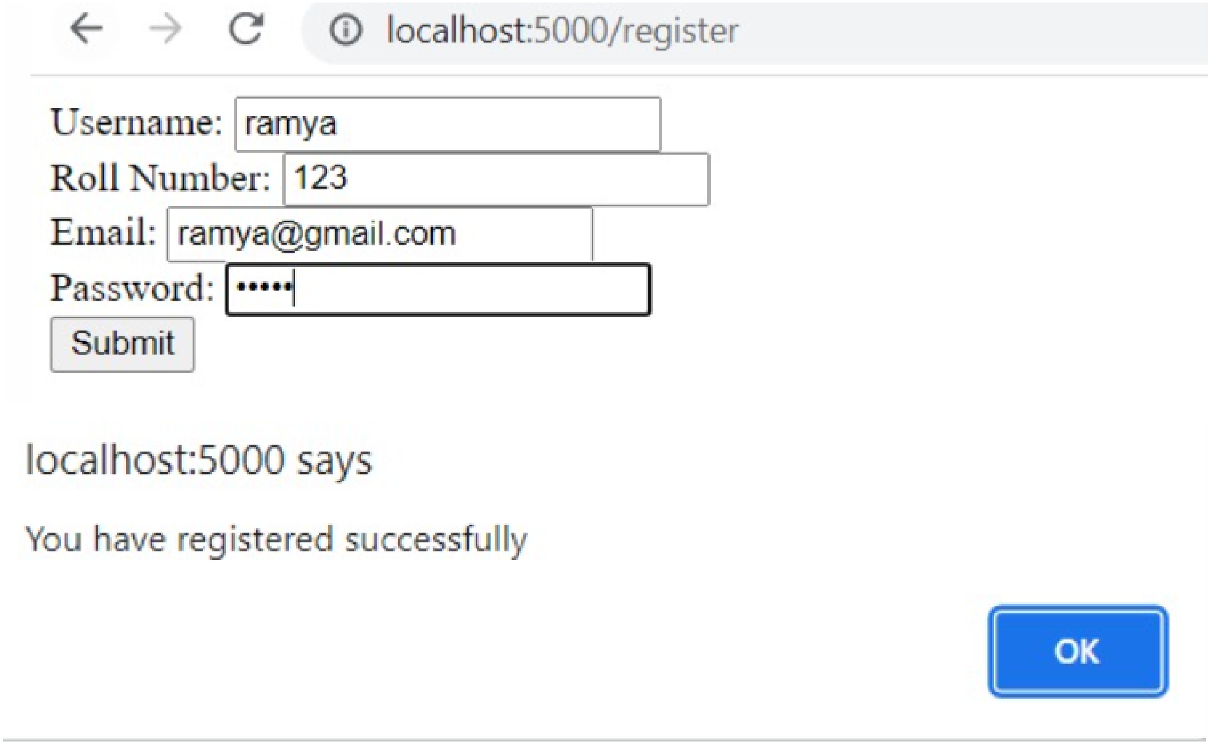
Password: ••••••



localhost:5000 says

Account already exists!!

- **A new user signing in:**



localhost:5000/register

Username: ramya

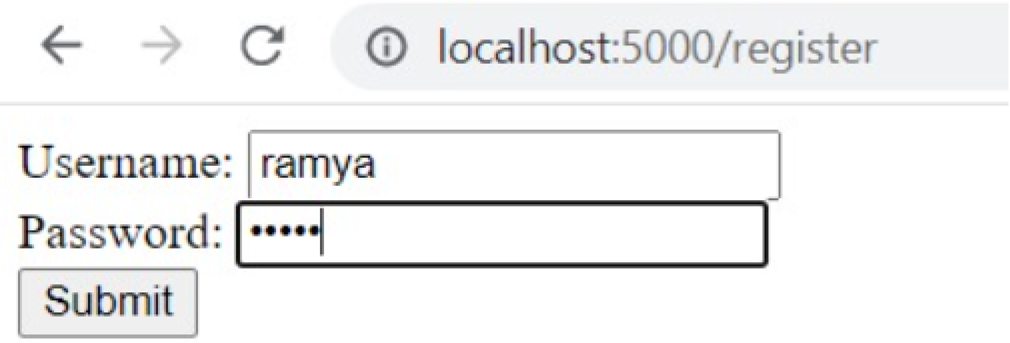
Roll Number: 123

Email: ramya@gmail.com

Password: •••••

localhost:5000 says

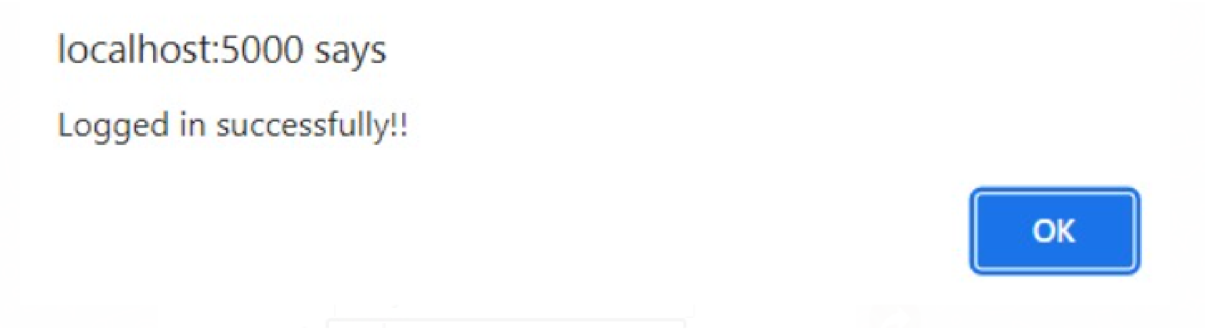
You have registered successfully



localhost:5000/register

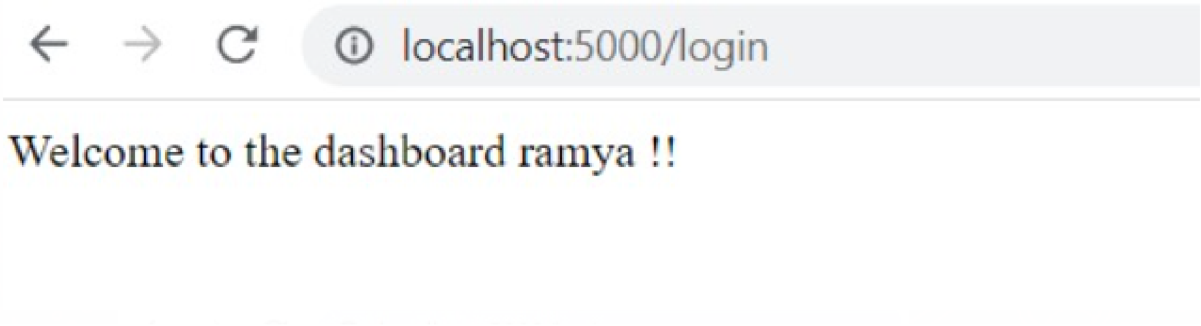
Username: ramya

Password: •••••



localhost:5000 says

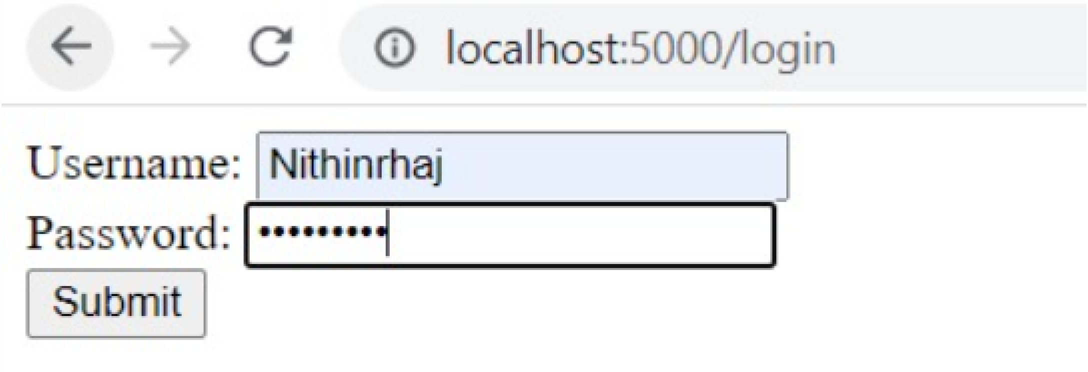
Logged in successfully!!



localhost:5000/login

Welcome to the dashboard ramya !!

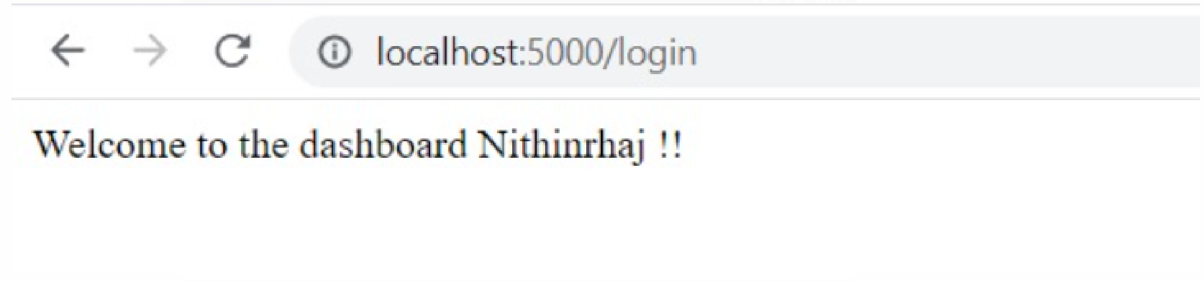
- **Signing in with a pre-existing user:**



localhost:5000/login

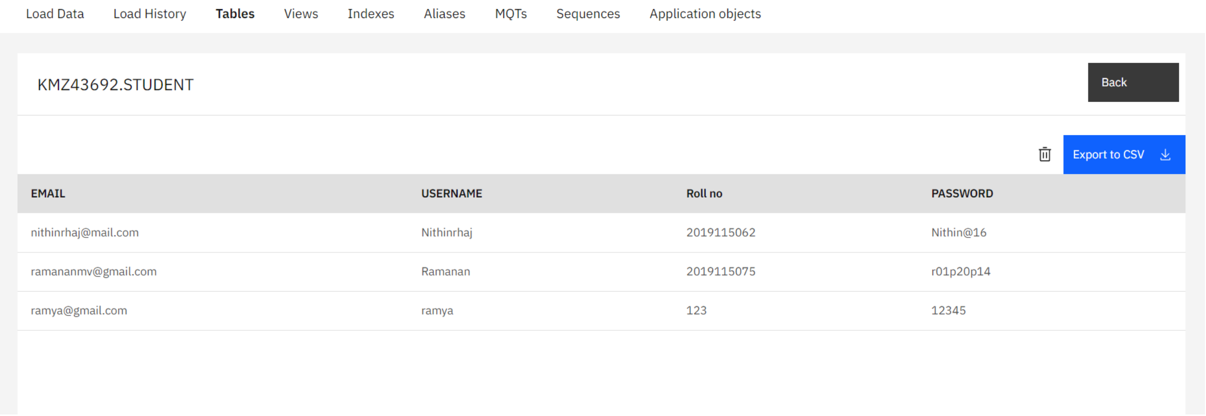
Username: Nithinrhaj

Password: •••••••••



Welcome to the dashboard Nithinrhaj !!

- **Updated database table after new user registers:**



KMZ43692.STUDENT

EMAIL	USERNAME	Roll no	PASSWORD
nithinrhaj@mail.com	Nithinrhaj	2019115062	Nithin@16
ramananmv@gmail.com	Ramanan	2019115075	r01p20p14
ramya@gmail.com	ramya	123	12345