

Date	20 October 2022
Name	Adarsh S
Roll no	SSNCE195001008
Batch	B7-1A3E

Assignment 2

1. Create User table with user with email, username, roll number, password.
2. Perform UPDATE, DELETE Queries with user table
3. Connect python code to db2.
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

Creation of Table

Table definition					
USER					
No statistics available.					
Name	Data type	Nullable	Length	Scale	
EMAIL	CHAR	N	50	0	🔍
UNAME	CHAR	N	50	0	🔍
ROLLNO	BIGINT	N		0	🔍
PASSWORD	CHAR	N	50	0	🔍

Insertion of Data

* Untitled - 1	
1	INSERT INTO USER VALUES('U1@SAMPLE.COM', 'USER1', '152', 'PASSWORD1');
2	INSERT INTO USER VALUES('U2@SAMPLE.COM', 'USER2', '289', 'PASSWORD2');
3	INSERT INTO USER VALUES('U3@SAMPLE.COM', 'USER3', '213', 'PASSWORD3');
4	INSERT INTO USER VALUES('U4@SAMPLE.COM', 'USER4', '127', 'PASSWORD4');

PZM72183.USER				Back
				Export to CSV
EMAIL	UNAME	ROLLNO	PASSWORD	
U1@SAMPLE.COM	USER1	152	PASSWORD1	
U2@SAMPLE.COM	USER2	289	PASSWORD2	
U3@SAMPLE.COM	USER3	213	PASSWORD3	
U4@SAMPLE.COM	USER4	127	PASSWORD4	

Performing Queries on the table

Update Query

* Untitled - 1

* Untitled - 2

* Template - Dele...

+

1

UPDATE USER SET UNAME='ADARSH' WHERE UNAME='USER1';

Delete Query

1

DELETE FROM USER

2

WHERE UNAME = 'USER3';

3

Table after updates

				Export to CSV
EMAIL	UNAME	ROLLNO	PASSWORD	
U1@SAMPLE.COM	ADARSH	152	PASSWORD1	
U2@SAMPLE.COM	USER2	289	PASSWORD2	
U4@SAMPLE.COM	USER4	127	PASSWORD4	

Connect python code to IBM Db2

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;Security=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pzm72183;PWD=esgc1kHRUqTUfn9I;", "", "")
```

Flask app to create registration form and insert data into Db2

```
from flask import Flask, render_template, request, redirect, url_for, flash
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField
from wtforms.validators import DataRequired, Email, EqualTo
import ibm_db
```

```
app = Flask(__name__)
app.config['SECRET_KEY'] = 'mysecretkey'
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;Security=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pzm72183;PWD=esgc1kHRUqTUfn9I;", "", "")
```

```
class RegistrationForm(FlaskForm):
    first_name = StringField('First Name', validators=[DataRequired()])
    last_name = StringField('Last Name', validators=[DataRequired()])
    email = StringField('Email', validators=[DataRequired(), Email()])
    phone = StringField('Phone', validators=[DataRequired()])
    password = PasswordField('Password', validators=[DataRequired()])
    submit = SubmitField('Submit')
```

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

```
@app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegistrationForm()
    if form.validate_on_submit():
        flash(f'Account created for {form.first_name.data} {form.last_name.data}!', 'success')
        return render_template('success.html', form=form)
    sql = "INSERT INTO USER (EMAIL, UNAME, ROLLNO, PASSWORD) VALUES (?, ?, ?, ?)"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, form.email.data)
    ibm_db.bind_param(stmt, 2, form.first_name.data)
    ibm_db.bind_param(stmt, 3, form.phone.data)
```

```
ibm_db.bind_param(stmt, 4, form.password.data)
ibm_db.execute(stmt)

return render_template('success.html', form=form)

@app.route('/success', methods=['GET', 'POST'])
def success():
    return render_template('success.html',
first_name=request.args.get('first_name'),
last_name=request.args.get('last_name'), email=request.args.get('email'),
phone=request.args.get('phone'))

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