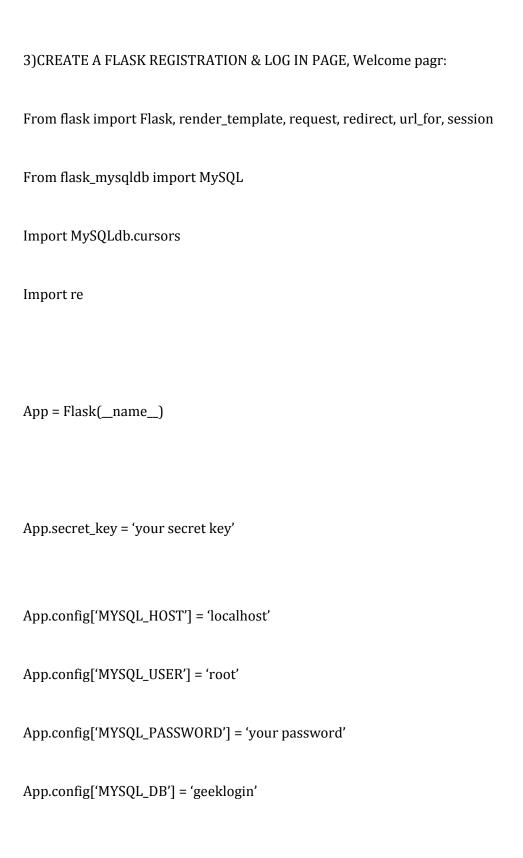
```
From flask import Flask
From flask_mail import Mail, Message
App = Flask(__name__)
Mail= Mail(app)
App.config['MAIL_SERVER']='smtp.gmail.com'
App.config['MAIL PORT'] = 465
App.config['MAIL_USERNAME'] = 'yourld@gmail.com'
App.config['MAIL PASSWORD'] = '*****'
App.config['MAIL_USE_TLS'] = False
App.config['MAIL_USE_SSL'] = True
Mail = Mail(app)
@app.route("/")
Def index():
Msg = Message('Hello', sender = 'yourld@gmail.com', recipients = ['someone1@gmail.com'])
Msg.body = "Hello Flask message sent from Flask-Mail"
Mail.send(msg)
 Return "Sent"
If __name__ == '__main__':
App.run(debug = True)
```

1)Create user table with user with email, user name, roll number, password:

2)PERFORME UPDATE & DELETE QUERIES WITH USER TABLE:

```
From flask import Flask,render_template,request,redirect
From models import db,userModel
App = Flask(__name___)
App.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///data.db'
App.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
Db.init_app(app)
//Update user
App.route('/data/<int:id>/update',methods = ['GET','POST'])
Def update(id):
  Employee = UserModel.query.filter_by(User_id=id).first()
  If request.method == 'POST':
    If user:
Db.session.delete(user)
Db.session.commit()
      Name = request.form['name']
      Age = request.form['age']
      Position = request.form['position']
User = UseeModel(Usee_id=id, name=name, age=age, position = position)
Db.session.add(User)
Db.session.commit()
      Return redirect(f'/data/{id}')
    Return f"Userwith id = {id} Does nit exist"
  Return render_template('update.html', User= User)
```

```
// DELETE USER
@app.route('/data/<int:id>/delete', methods=['GET','POST'])
Def delete(id):
USER= UserModel.query.filter_by(User_id=id).first()
  If request.method == 'POST':
    If User:
Db.session.delete(User)
Db.session.commit()
      Return redirect('/data')
    Abort(404)
  Return render_template('delete.flask')
App.run(host='localhost', port=5000)
3) CONNECT PYTHON CODE TO db2:
db2id = { // service credential dictionary here //}
api = "/dbapi/v3"
host = db2id['https_url']+api
userinfo = {"userid":db2id['username'],"password":db2id['password']}
service = '/auth/tokens'
r = requests.post(host+service,json=userinfo)
access_token = r.json()['token']
auth_header = {"Authorization": "Bearer "+access_token}
```



```
Mysql = MySQL(app)
@app.route('/')
@app.route('/login', methods =['GET', 'POST'])
Def login():
Msg = "
 If request.method == 'POST' and 'username' in request.form and 'password' in request.form:
   Username = request.form['username']
   Password = request.form['password']
   Cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
Cursor.execute('SELECT * FROM accounts WHERE username = % s AND password = % s',
(username, password, ))
   Account = cursor.fetchone()
   If account:
     Session['loggedin'] = True
     Session['id'] = account['id']
```

```
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' and 'username' in request.form and 'password' in request.form and
'email' in request.form:
    Username = request.form['username']
    Password = request.form['password']
    Email = request.form['email']
    Cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
Cursor.execute('SELECT * FROM accounts WHERE username = % s', (username, ))
   Account = cursor.fetchone()
    If account:
Msg = 'Account already exists!'
Elif not re.match(r'[^{\alpha}]+@[^{\alpha}]+.[^{\alpha}]+', email):
Msg = 'Invalid email address!'
Elif not re.match(r'[A-Za-z0-9]+', username):
Msg = 'Username must contain only characters and numbers!'
```

