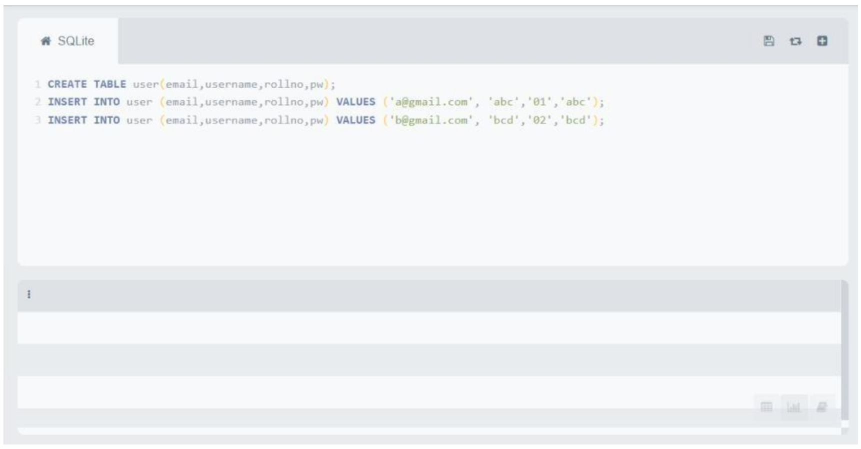
Create Flask project

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

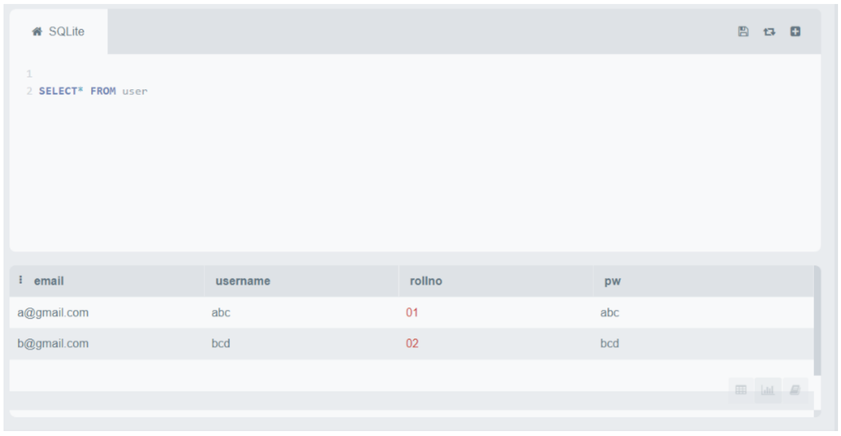
Solution:

CREATE TABLE user(email,username,rollno,pw);
INSERT INTO user (email,username,rollno,pw) VALUES ('a@gmail.com', 'abc','01','abc');
INSERT INTO user (email,username,rollno,pw) VALUES ('b@gmail.com', 'bcd','02','bcd');
INSERT INTO user (email,username,rollno,pw) VALUES ('c@gmail.com', 'cde','03','cde');
INSERT INTO user (email,username,rollno,pw) VALUES ('d@gmail.com', 'def','04','def');

Output:



```
CREATE TABLE user(email,username,rollno,pw);
INSERT INTO user (email,username,rollno,pw) VALUES ('a@gmail.com', 'abc','01','abc');
INSERT INTO user (email,username,rollno,pw) VALUES ('b@gmail.com', 'bcd','02','bcd');
```



```

SELECT* FROM user
```

email	username	rollno	pw
a@gmail.com	abc	01	abc
b@gmail.com	bcd	02	bcd

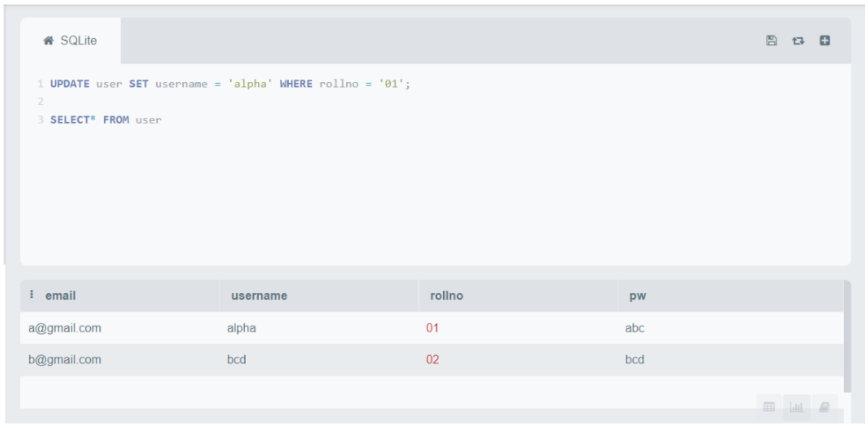
Perform UPDATE,DELETE Queries with user table

Solution:

UPDATE user SET username = 'alpha' WHERE rollno = '01';
SELECT* FROM user

DELETE FROM user WHERE username = 'bcd';
SELECT* FROM user

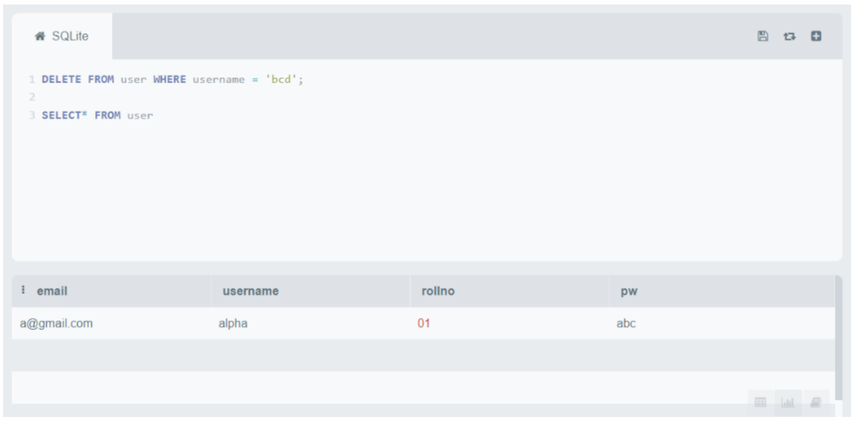
Output:



```
UPDATE user SET username = 'alpha' WHERE rollno = '01';

SELECT* FROM user
```

email	username	rollno	pw
a@gmail.com	alpha	01	abc
b@gmail.com	bcd	02	bcd



```
DELETE FROM user WHERE username = 'bcd';

SELECT* FROM user
```

email	username	rollno	pw
a@gmail.com	alpha	01	abc

Connect python code to db2.

Solution:

```
pip install ibm_db
from ibm_db import connect
connection = connect('DATABASE=<database name>;'
                     'HOSTNAME=<database ip>;' # 127.0.0.1 or localhost works if it's local
                     'PORT=<database port>;'
                     'PROTOCOL=TCPIP;'
                     'UID=<database username>;'
                     'PWD=<username password>;', '', '')
def results(command):

    from ibm_db import fetch_assoc

    ret = []
    result = fetch_assoc(command)
    while result:
        ret.append(result)
        result = fetch_assoc(command)
    return ret # Ditch this line if you choose to use a generator.
from ibm_db import tables

t = results(tables(connection))
```

```
from ibm_db import exec_immediate

sql = 'LIST * FROM ' + t[170]['TABLE_NAME']
rows = results(exec_immediate(connection, sql))
```

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

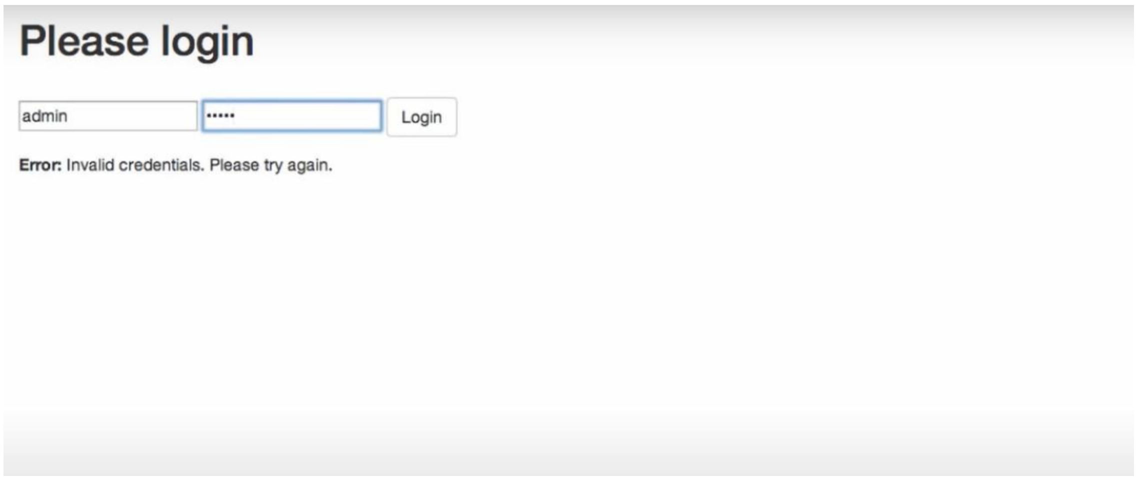
Solution:

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    error = None
    if request.method == 'POST':
        if request.form['username'] != 'admin' or request.form['password'] != 'admin':
            error = 'Invalid Credentials. Please try again.'
        else:
            return redirect(url_for('home'))
    return render_template('login.html', error=error)
```

html code:

```
<html>
  <head>
    <title>Flask Intro - login page</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="static/bootstrap.min.css" rel="stylesheet" media="screen">
  </head>
  <body>
    <div class="container">
      <h1>Please login</h1>
      <br>
      <form action="" method="post">
        <input type="text" placeholder="Username" name="username" value="{{
          request.form.username }}">
         <input type="password" placeholder="Password" name="password" value="{{
          request.form.password }}">
        <input class="btn btn-default" type="submit" value="Login">
      </form>
      {% if error %}
        <p class="error"><strong>Error:</strong> {{ error }}
      {% endif %}
    </div>
  </body>
</html>
```

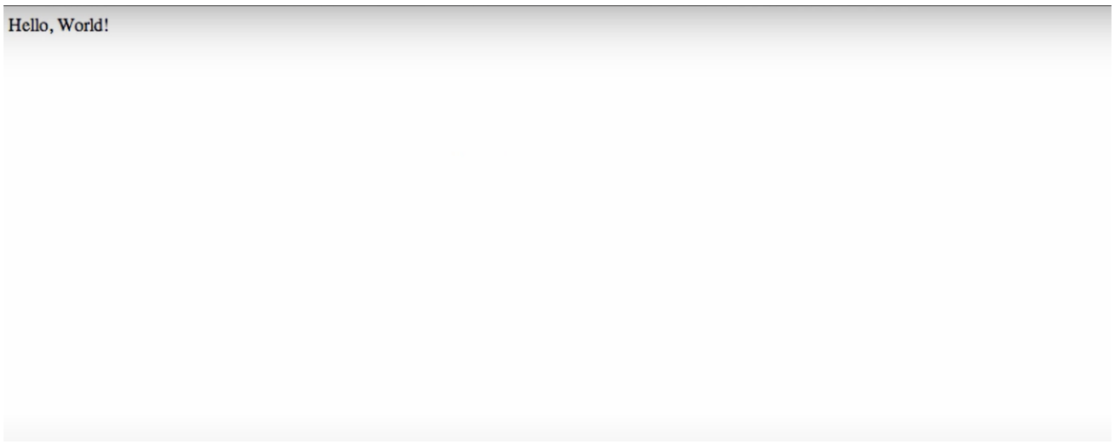
Output:



Please login

admin ••••• Login

Error: Invalid credentials. Please try again.



Hello, World!