

ASSIGNMENT-2

Assignment Date	19 September 2022
Student Name	M.Subasri
Student Roll Number	820419104074
Team ID	PNT2022TMID32929
Maximum Marks	2 Marks

Question-1:

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

Solution:

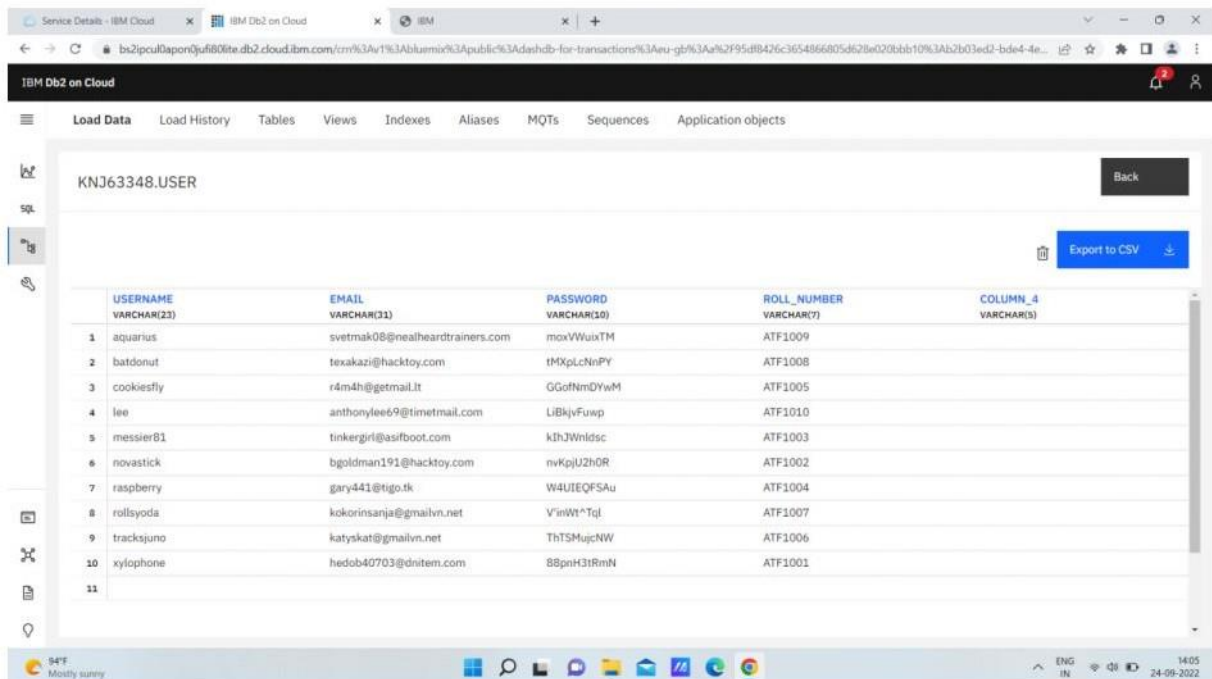
Excel comma file

Username	Email	Password	Roll Number
xylophone	bedob40703@dsitem.com	88puH9FmN	ATF1001
novastick	bpdksm191@hacktoy.com	mKpUz6R	ATF1002
messier81	tsikergst1@autfoot.com	kthWaldse	ATF1003
raspberry	gary441@tigo.tk	W4UTEQFSAu	ATF1004
cookiecity	r4m4h@getmail.it	G0dNnDYwM	ATF1005
truckysimo	katyskat@gmail.naet	ThTSMjgeNW	ATF1006
rollyoda	kokoraseya@gmail.naet	VtaWvTgl	ATF1007
latidout	texakazi@hacktoy.com	tMXpLcNnPY	ATF1008
aquaria	svetmak08@nealhearthrainers.com	moxVWuxTM	ATF1009
lee	anthonylee69@tinetmail.com	LiBkvFunop	ATF1010

User table in Db2

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	Y	23	0
EMAIL	VARCHAR	Y	31	0
PASSWORD	VARCHAR	Y	10	0
ROLL_NUMBER	VARCHAR	Y	7	0
COLUMN_4	VARCHAR	Y	5	0

Data in User table



The screenshot shows the IBM Db2 on Cloud console interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The main content area displays the 'KNJ63348.USER' table. A table with 5 columns (USERNAME, EMAIL, PASSWORD, ROLL_NUMBER, COLUMN_4) and 11 rows of data is shown. An 'Export to CSV' button is visible. The bottom status bar shows '94°F Mostly sunny' and the date '24-09-2022'.

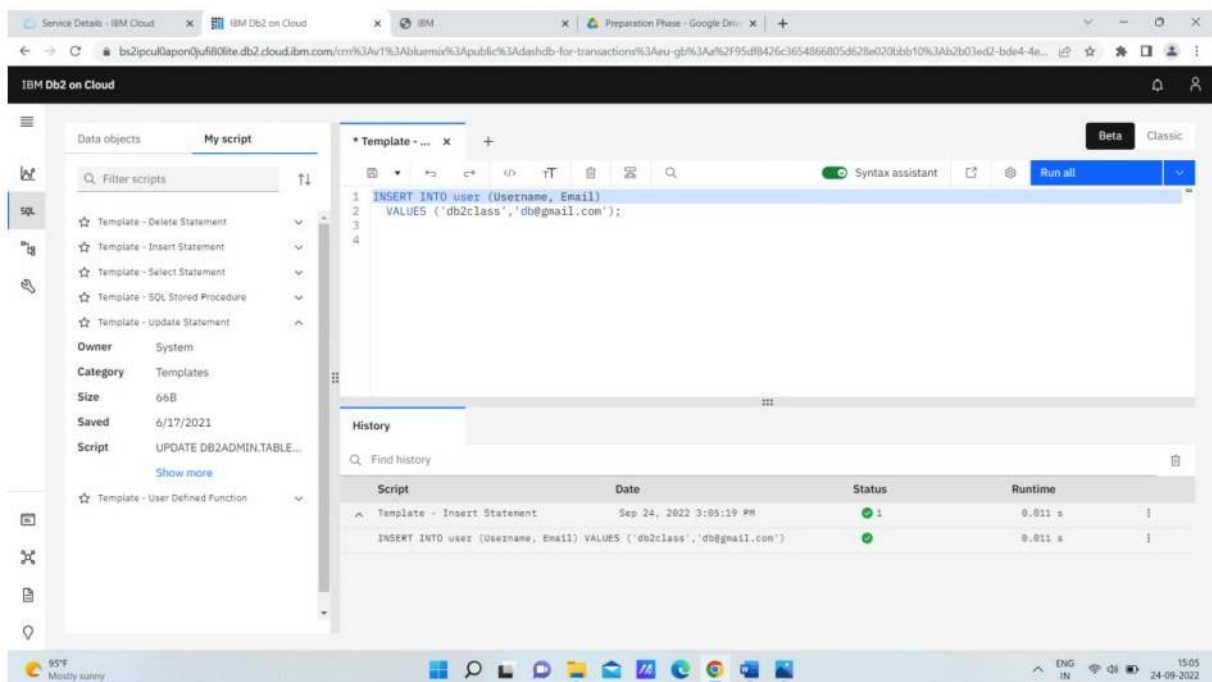
	USERNAME VARCHAR(23)	EMAIL VARCHAR(31)	PASSWORD VARCHAR(10)	ROLL_NUMBER VARCHAR(7)	COLUMN_4 VARCHAR(5)
1	aquarius	svetmak08@nealheardtrainers.com	moxVWuixTM	ATF1009	
2	batdonut	tekakazi@hacktoy.com	tMXpLcNnPY	ATF1008	
3	cookiesfly	r4m4h@getmail.lt	GGoNmDYwM	ATF1005	
4	lee	anthonylee69@timetmail.com	LiBkVfuwp	ATF1010	
5	messier81	tinkergirl@asifboot.com	kIHJWhldsc	ATF1003	
6	novastick	bgoldman191@hacktoy.com	nvKpJU2h0R	ATF1002	
7	raspberry	gary441@tigo.tk	W4UIEQFSau	ATF1004	
8	rollsoda	kokorinsanja@gmailvn.net	V'inWt^TqL	ATF1007	
9	tracksjuno	katyskat@gmailvn.net	ThTSMujcNW	ATF1006	
10	xylophone	hedob40703@dnitem.com	BBgnH3IRmN	ATF1001	
11					

Question-2:

Perform INSERT, SELECT, DELETE Queries in user table

Solution:

Insert Query



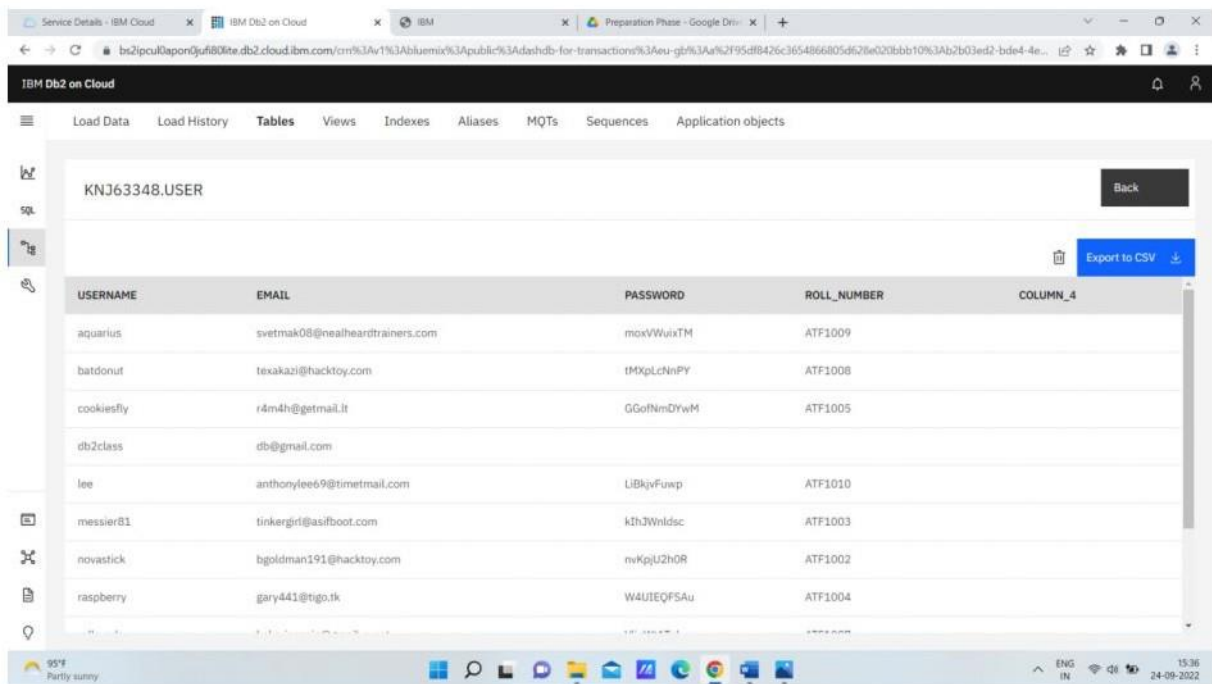
The screenshot shows the IBM Db2 on Cloud console interface. The left sidebar shows 'Data objects' and 'My script'. The main area displays an SQL script editor with the following code:

```
1 INSERT INTO user (Username, Email)
2 VALUES ('db2class', 'db@gmail.com');
3
4
```

The 'Run all' button is highlighted. Below the script editor, a 'History' table shows the execution details:

Script	Date	Status	Runtime
Template - Insert Statement	Sep 24, 2022 3:05:19 PM	✓ 1	0.011 s
INSERT INTO user (Username, Email) VALUES ('db2class', 'db@gmail.com')		✓	0.011 s

Insert Query Output



Service Details - IBM Cloud | IBM Db2 on Cloud | IBM | Preparation Phase - Google Drive

bs2ipcul8apon0juf80ite.db2.cloud.ibm.com/cn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transaction%3Aeu-gb%3A%2F95d8426c3654866805d628e020bbb10%3Ab2b03ed2-bde4-4e...

IBM Db2 on Cloud

Load Data | Load History | **Tables** | Views | Indexes | Aliases | MQTs | Sequences | Application objects

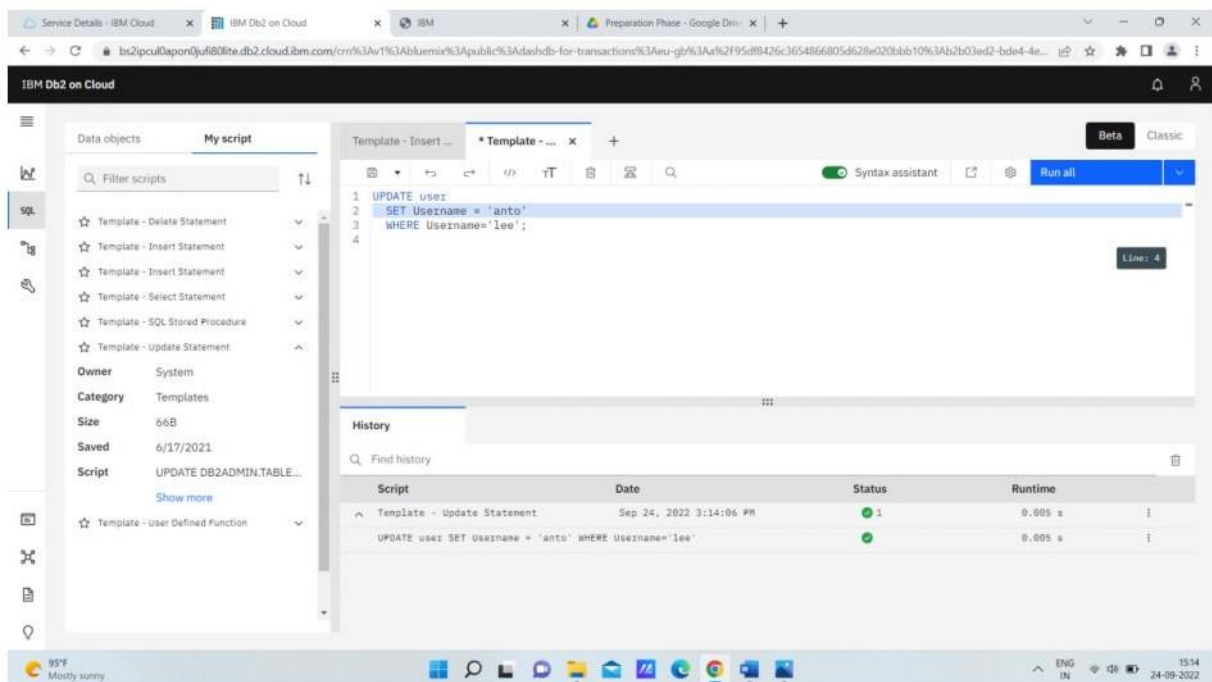
KNJ63348.USER

Export to CSV

USERNAME	EMAIL	PASSWORD	ROLL_NUMBER	COLUMN_4
aquarius	svetmak08@nealheardtrainers.com	moxvWuixTM	ATF1009	
batdonut	texakazi@hacktoy.com	tMXpLcNnPY	ATF1008	
cookiesfly	r4m4h@gmail.it	GGoNmDYwM	ATF1005	
db2class	db@gmail.com			
lee	anthonylee69@tinetmail.com	LiBkqvFuwp	ATF1010	
messier81	tinkergirl@asfboot.com	kIhJWnldsc	ATF1003	
novastick	bgoldman191@hacktoy.com	nvKpU2h0R	ATF1002	
raspberry	gary441@tigo.tk	W4UUEQFSAu	ATF1004	

95°F Partly sunny | 15:36 24-09-2022

Update Query



Service Details - IBM Cloud | IBM Db2 on Cloud | IBM | Preparation Phase - Google Drive

bs2ipcul8apon0juf80ite.db2.cloud.ibm.com/cn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transaction%3Aeu-gb%3A%2F95d8426c3654866805d628e020bbb10%3Ab2b03ed2-bde4-4e...

IBM Db2 on Cloud

Data objects | **My script**

Filter scripts

- Template - Delete Statement
- Template - Insert Statement
- Template - Insert Statement
- Template - Select Statement
- Template - SQL Stored Procedure
- Template - Update Statement
- Template - User Defined Function

Owner: System
Category: Templates
Size: 66B
Saved: 6/17/2021
Script: UPDATE DBADMIN.TABLE...
[Show more](#)

Template - Insert ...

Syntax assistant | Run all

```
1 UPDATE user
2 SET Username = 'anto'
3 WHERE Username='lee';
4
```

History

Find history

Script	Date	Status	Runtime
Template - Update Statement	Sep 24, 2022 3:14:06 PM	1	0.005 s
UPDATE user SET Username = 'anto' WHERE Username='lee'			0.005 s

95°F Mostly sunny | 15:14 24-09-2022

Update Query Output

Service Details - IBM Cloud | IBM Db2 on Cloud | IBM | Preparation Phase - Google Drive

bs2ipculDapon@juf80ite.db2.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transaction%3Aeu-gb%3Aa%2F95df8426c3654866805d628e020bbb10%3Ab2b03ed2-bde4-4e...

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

KNJ63348.USER Back

Export to CSV

USERNAME	EMAIL	PASSWORD	ROLL_NUMBER	COLUMN_4
anto	anthonylee69@timetmail.com	LiBkqvFuwp	ATF1010	
aquarius	svetmak08@nealheardtrainers.com	moxVWux7M	ATF1009	
batdonut	texakazi@hacktoy.com	tMXpLcNnPY	ATF1008	
cookiesfly	r4m4h@gmail.it	GGofNmDYwM	ATF1005	
db2class	db@gmail.com			
messier81	tinkergirl@asifboot.com	kHjJWnldsc	ATF1003	
novastick	bgoldman191@hacktoy.com	nvKpU2h0R	ATF1002	
raspberry	gary441@rigo.tk	W4UIEQFSAu	ATF1004	

95°F Partly sunny

Delete Query

bs2ipculDapon@juf80ite.db2.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transaction%3Aeu-gb%3Aa%2F95df8426c3654866805d628e020bbb10%3Ab2b03ed2-bde4-4e...

IBM Db2 on Cloud

Data objects My script

Filter scripts

- Template - Delete Statement
- Template - Insert Statement
- Template - Insert Statement
- Template - Select Statement
- Template - SQL Stored Procedure
- Template - Update Statement

Owner System

Category Templates

Size 66B

Saved 6/17/2021

Script UPDATE DB2ADMIN.TABLE...

Show more

Template - User Defined Function

Template - Insert ... * Template - Upda... * Template - ... x +

Syntax assistant Run all

```
1 DELETE FROM user
2 WHERE Username = 'db2class';
3
```

History Results

Find history

Script	Date	Status	Runtime
Template - Delete Statement	Sep 24, 2022 3:38:56 PM	1	0.005 s
DELETE FROM user WHERE Username = 'db2class'			0.005 s

Delete Query Output

bs2ipculDapon@juf80ite.db2.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transaction%3Aeu-gb%3Aa%2F95df8426c3654866805d628e020bbb10%3Ab2b03ed2-bde4-4e...

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

KNJ63348.USER Back

Export to CSV

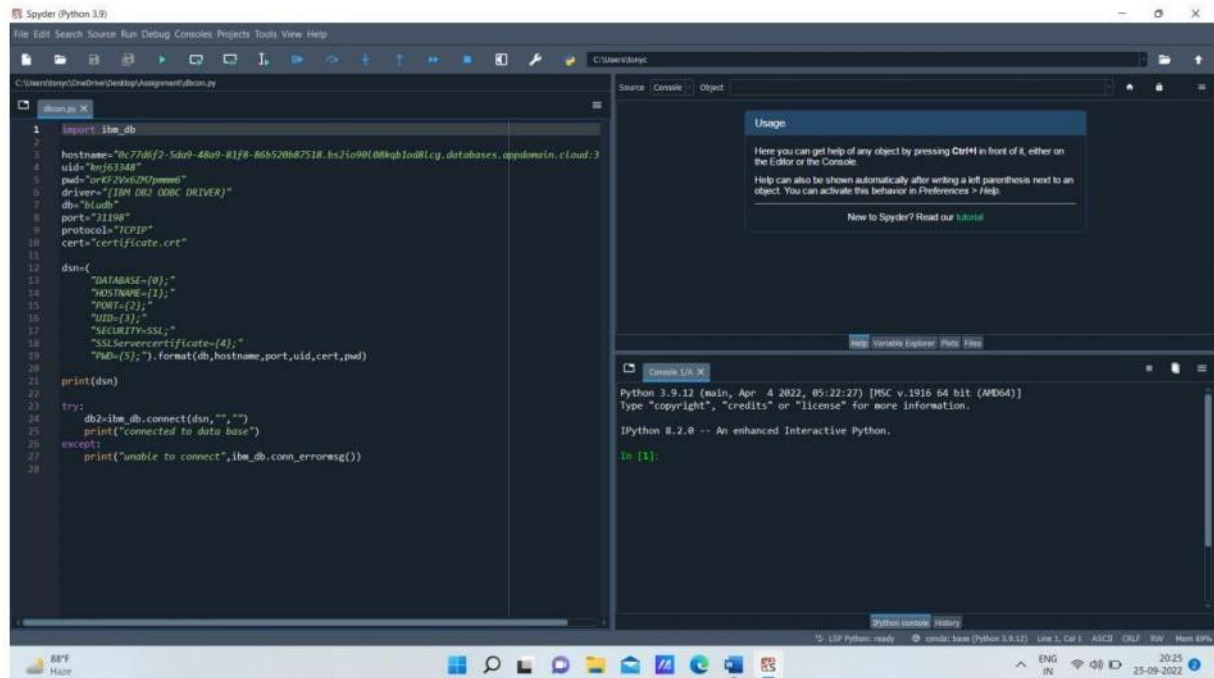
USERNAME	EMAIL	PASSWORD	ROLL_NUMBER	COLUMN_4
anto	anthonylee69@timetmail.com	LiBkqvFuwp	ATF1010	
aquarius	svetmak08@nealheardtrainers.com	moxVWux7M	ATF1009	
batdonut	texakazi@hacktoy.com	tMXpLcNnPY	ATF1008	
cookiesfly	r4m4h@gmail.it	GGofNmDYwM	ATF1005	
messier81	tinkergirl@asifboot.com	kHjJWnldsc	ATF1003	
novastick	bgoldman191@hacktoy.com	nvKpU2h0R	ATF1002	
raspberry	gary441@rigo.tk	W4UIEQFSAu	ATF1004	
roilsyoda	kokorinsanja@gmailvn.net	V'inWt*Qqj	ATF1007	

Question-3:

Connect Python to DB2

Solution:

Source Code for Python to DB2 Connection



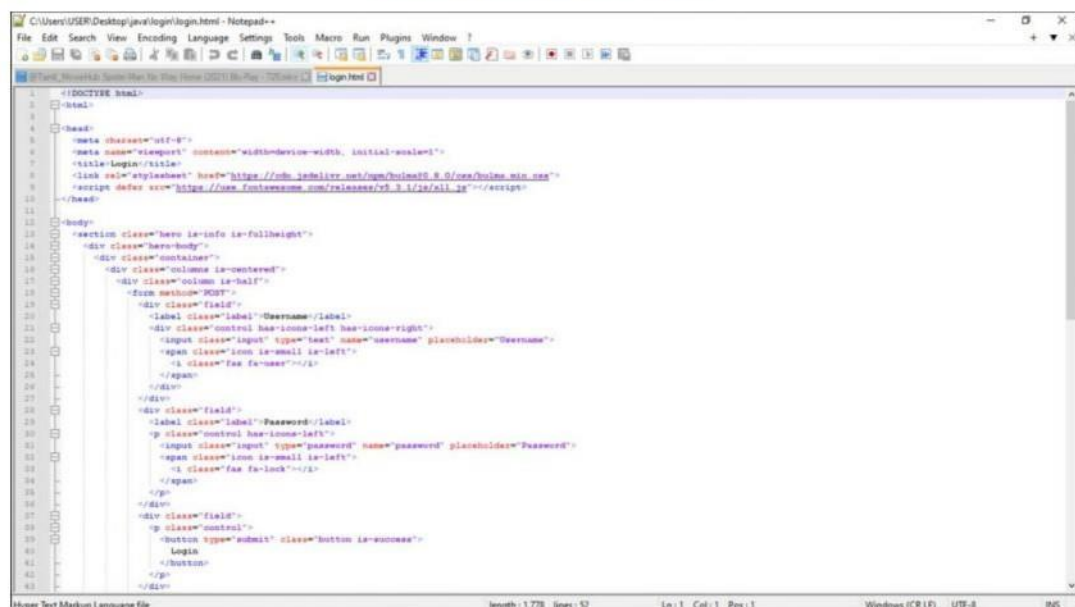
```
1 import ibm_db
2
3 hostname="6c77d6f2-5du9-48a9-81f8-86b5206d7518.b2i090l08qphlodR1cy.databases.appdomain.cloud:3
4 uid="bnj61348"
5 pwd="orE72Vv6DHpmem"
6 driver="IBM ODBC DRIVER"
7 db="blues"
8 port="21198"
9 protocol="TCP/IP"
10 cert="certificate.crt"
11
12 dsn={
13     "DATABASE={0};"
14     "HOSTNAME={1};"
15     "PORT={2};"
16     "UID={3};"
17     "SECURITY=SSL;"
18     "SSLServerCertificate={4};"
19     "PWD={5};".format(db,hostname,port,uid,pwd)
20 }
21
22 print(dsn)
23
24 try:
25     db2=ibm_db.connect(dsn,"")
26     print("connected to data base")
27 except:
28     print("unable to connect",ibm_db.conn_errorsg())
```

Question-4:

Create a flask app with 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

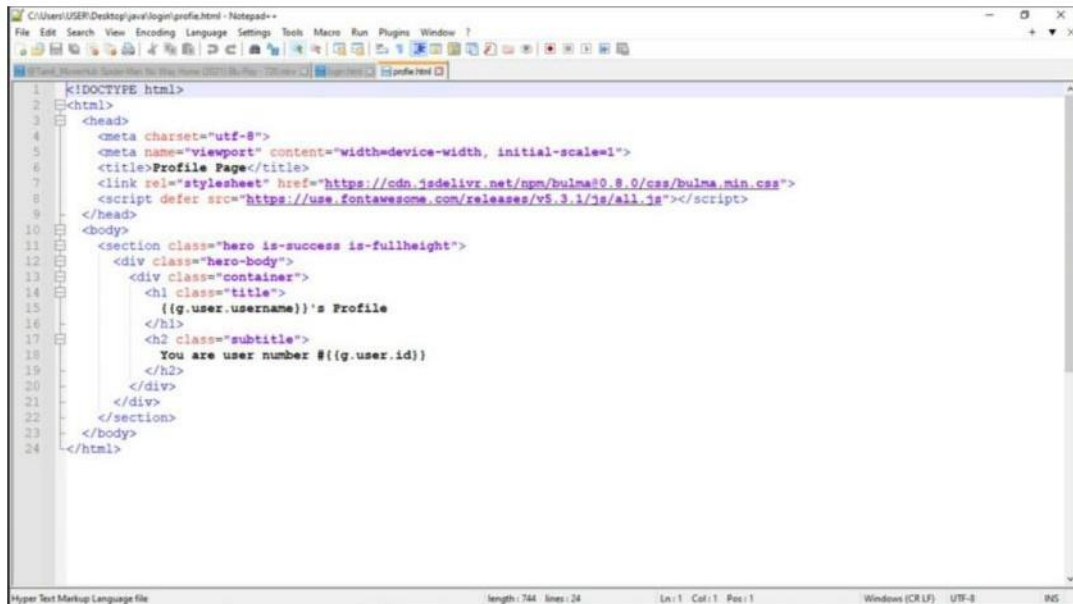
Solution:

Source code for login page



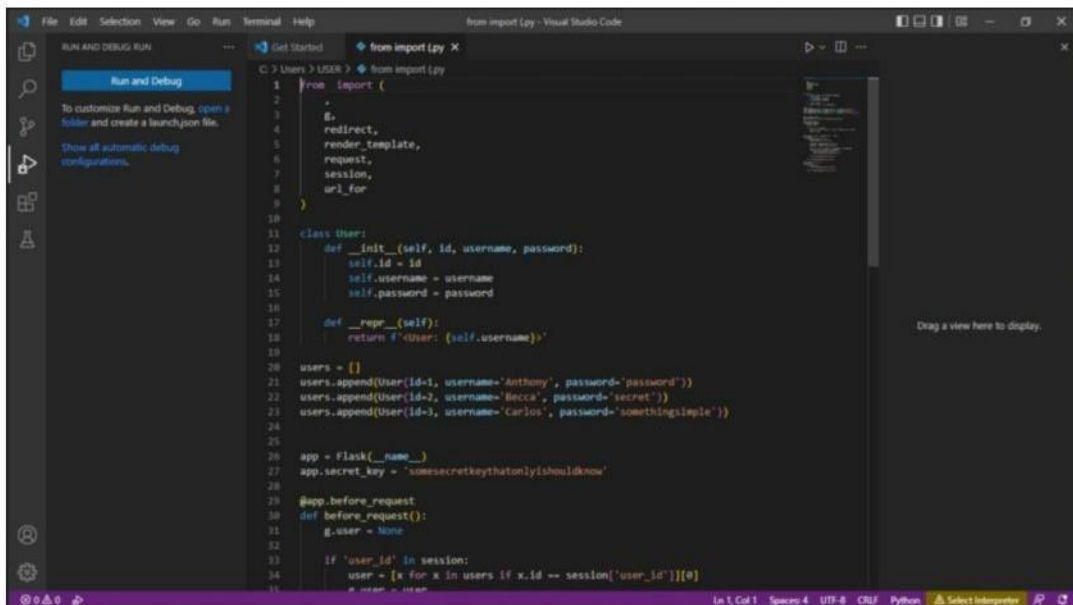
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>login</title>
7 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css">
8 <script defer src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"></script>
9 </head>
10 <body>
11 <div class="container">
12 <div class="row">
13 <div class="col-md-6">
14 <div class="card">
15 <div class="card-body">
16 <div class="text-center">
17 <h3>Login</h3>
18 </div>
19 <div class="form-group">
20 <input type="text" class="form-control" placeholder="Username">
21 </div>
22 <div class="form-group">
23 <input type="password" class="form-control" placeholder="Password">
24 </div>
25 <div class="text-center">
26 <button type="button" class="btn btn-primary">Login</button>
27 </div>
28 </div>
29 </div>
30 </div>
31 </div>
32 </body>
33 </html>
```


Source code for welcome page



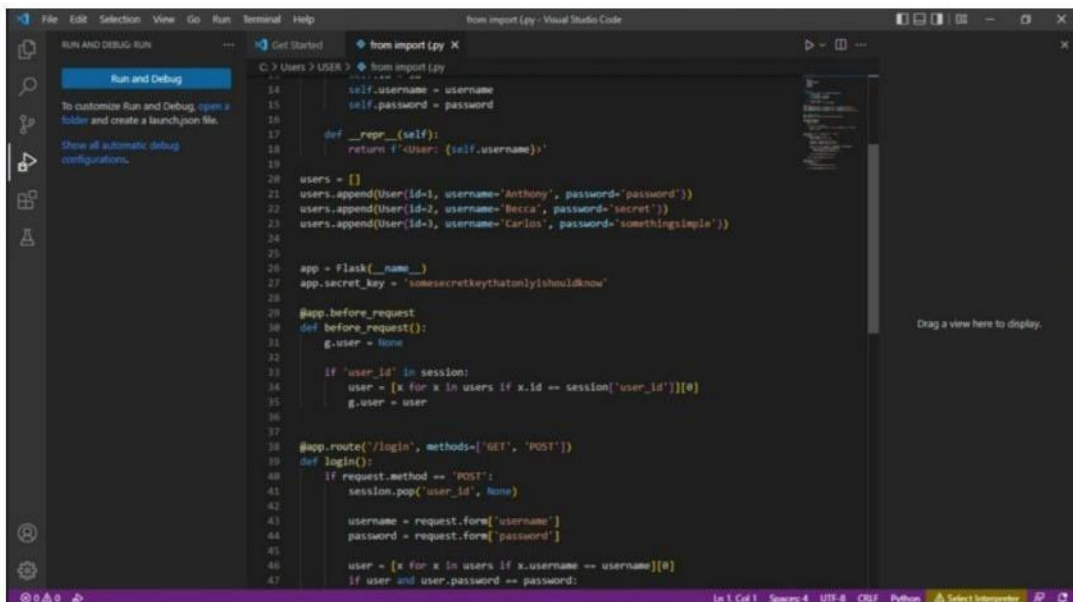
```
1<!DOCTYPE html>
2<html>
3<head>
4<meta charset="utf-8">
5<meta name="viewport" content="width=device-width, initial-scale=1">
6<title>Profile Page</title>
7<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.8.0/css/bulma.min.css">
8<script defer src="https://use.fontawesome.com/releases/v5.3.1/js/all.js"></script>
9</head>
10<body>
11<section class="hero is-success is-fullheight">
12<div class="hero-body">
13<div class="container">
14<h1 class="title">
15  {{g.user.username}}'s Profile
16</h1>
17<h2 class="subtitle">
18  You are user number #({{g.user.id}})
19</h2>
20</div>
21</div>
22</section>
23</body>
24</html>
```

Flask code



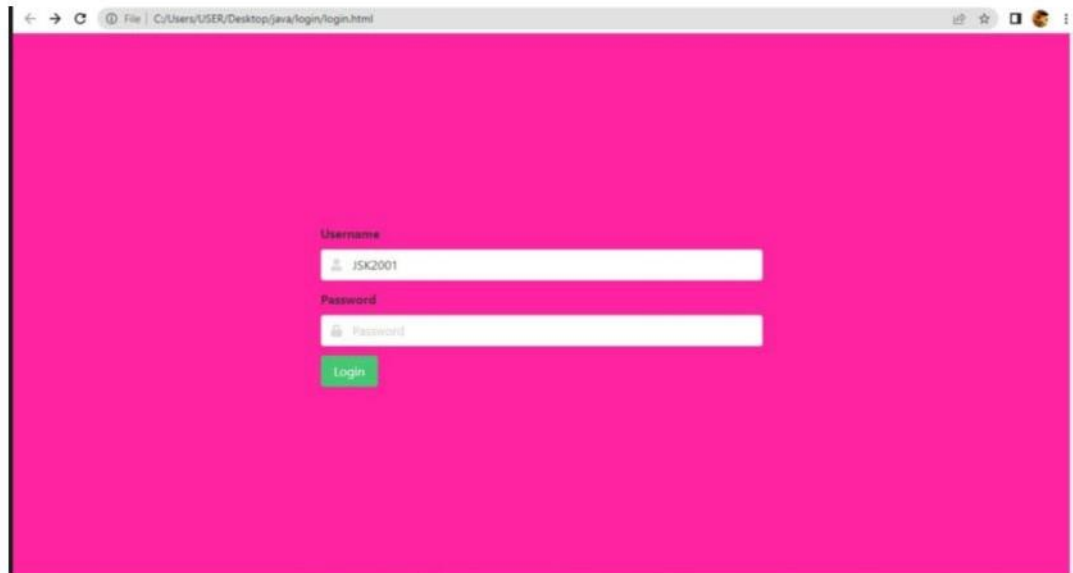
```
1from import (
2
3    redirect,
4    render_template,
5    request,
6    session,
7    url_for
8)
9
10class User:
11    def __init__(self, id, username, password):
12        self.id = id
13        self.username = username
14        self.password = password
15
16    def __repr__(self):
17        return f'<User: {self.username}>'
18
19users = []
20users.append(User(id=1, username='Anthony', password='password'))
21users.append(User(id=2, username='Becca', password='secret'))
22users.append(User(id=3, username='Carlos', password='somethingimple'))
23
24app = Flask(__name__)
25app.secret_key = 'somesecretkeythatonlyishouldknow'
26
27@app.before_request
28def before_request():
29    g.user = None
30
31    if 'user_id' in session:
32        user = [x for x in users if x.id == session['user_id']][0]
33        g.user = user
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Flask code



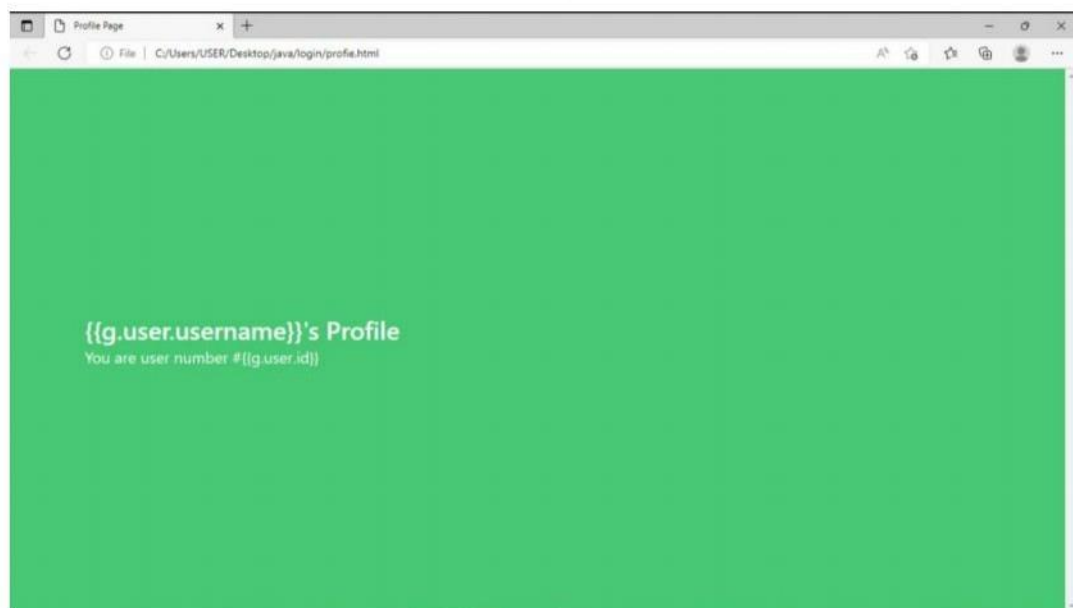
```
14self.username = username
15self.password = password
16
17def __repr__(self):
18    return f'<User: {self.username}>'
19
20users = []
21users.append(User(id=1, username='Anthony', password='password'))
22users.append(User(id=2, username='Becca', password='secret'))
23users.append(User(id=3, username='Carlos', password='somethingimple'))
24
25app = Flask(__name__)
26app.secret_key = 'somesecretkeythatonlyishouldknow'
27
28@app.before_request
29def before_request():
30    g.user = None
31
32    if 'user_id' in session:
33        user = [x for x in users if x.id == session['user_id']][0]
34        g.user = user
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Login Page Output



A screenshot of a web browser displaying a login page. The page has a solid pink background. In the center, there is a login form with two input fields and a button. The first input field is labeled 'Username' and contains the text 'JSK2001'. The second input field is labeled 'Password' and contains the text 'Password'. Below these fields is a green button with the text 'Login' in white. The browser's address bar shows the file path 'C:/Users/USER/Desktop/java/login/login.html'.

Welcome Page Output



A screenshot of a web browser displaying a welcome page. The page has a solid green background. On the left side, there is a heading that reads '{{g.user.username}}'s Profile' and a line of text below it that reads 'You are user number #[[g.user.id]]'. The browser's address bar shows the file path 'C:/Users/USER/Desktop/java/login/profie.html'.