

Question-1:

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

Solution:

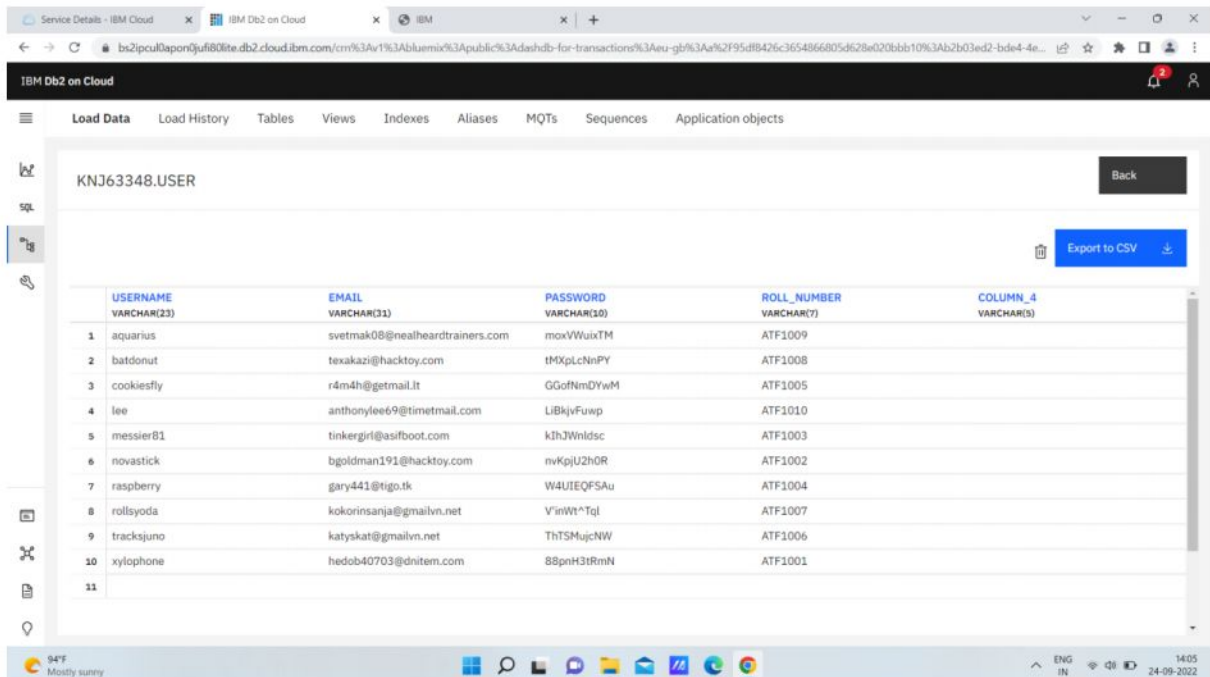
Excel comma file

Username	Email	Password	Roll Number
xylophone	hedob40703@dnitem.com	88pnH3RmN	ATF1001
novastick	bgoldman191@blacktoy.com	svKpJ2b0R	ATF1002
messierf1	tinkergirl@swifboot.com	kihJWaldse	ATF1003
raspberry	gary441@tigo.dk	W4UIEQFSau	ATF1004
cookiecify	r4m4b@getmail.it	OGofNmDYwM	ATF1005
trackyjano	kutykat@gmail.vn.net	TlTSMajeNW	ATF1006
rolloyoda	kokorinasenja@gmail.vn.net	VisWvTqj	ATF1007
baidomat	texokazi@blacktoy.com	dMXpLcNuPY	ATF1008
aquarius	vvetmsk08@neelheardtrainers.com	mexVWuaTM	ATF1009
lee	anthonylee69@tinetmail.com	LiBkjvFuvsp	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 table 'KNJ63348.USER'. 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 in the top right corner of the table view.

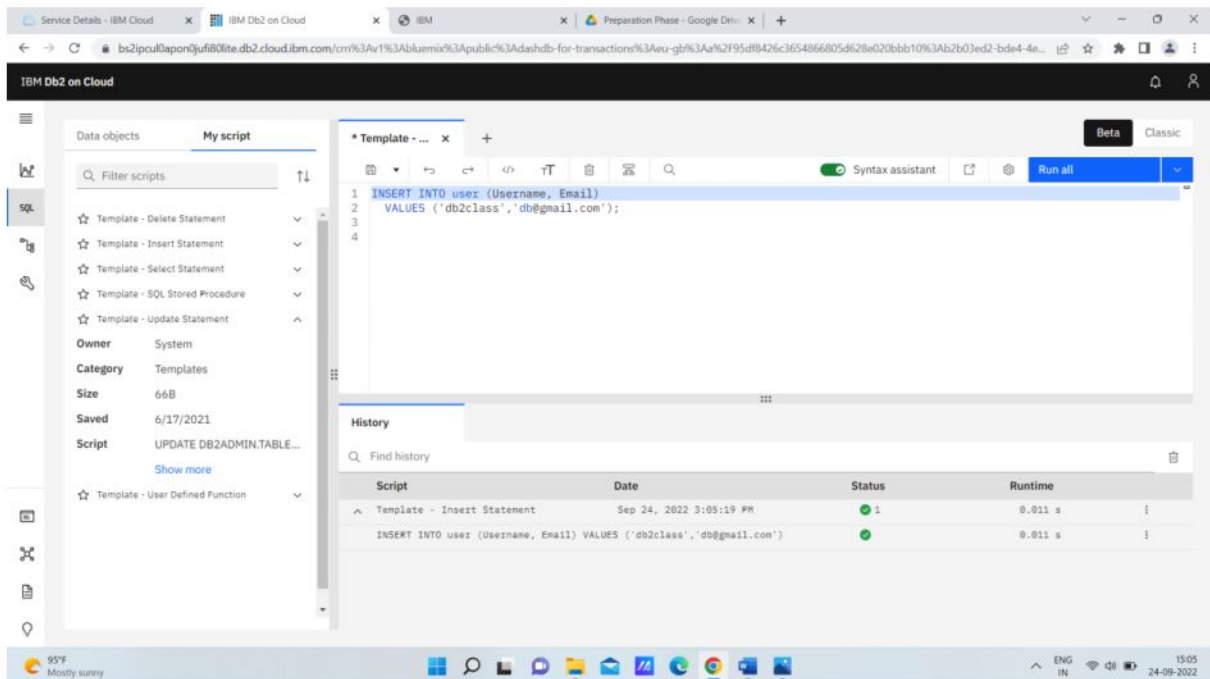
	USERNAME VARCHAR(23)	EMAIL VARCHAR(31)	PASSWORD VARCHAR(10)	ROLL_NUMBER VARCHAR(7)	COLUMN_4 VARCHAR(5)
1	aquarius	svetmak08@nealheardtrainers.com	moxVWuixTM	ATF1009	
2	batdonut	texakazi@hacktoy.com	tMXpLcNnPY	ATF1008	
3	cookiesfly	r4m4h@getmail.it	GGofNmDYwM	ATF1005	
4	lee	anthonylee69@timetmail.com	LiBkijvFuwp	ATF1010	
5	messier81	tinkergirl@asifboot.com	kIHjWnldsc	ATF1003	
6	novastick	bgoldman191@hacktoy.com	nvKpiU2h0R	ATF1002	
7	raspberry	gary441@tigo.tk	W4UTEQFSAu	ATF1004	
8	rollslyoda	kokorinsanja@gmailvn.net	V'inWt^TqI	ATF1007	
9	tracksjuno	katyskat@gmailvn.net	ThTSMujcNW	ATF1006	
10	xylophone	hedob40703@dnitem.com	8SpnH3trmH	ATF1001	
11					

Question-2:

Perform INSERT, SELECT, DELETE Queries in user table

Solution:

Insert Query

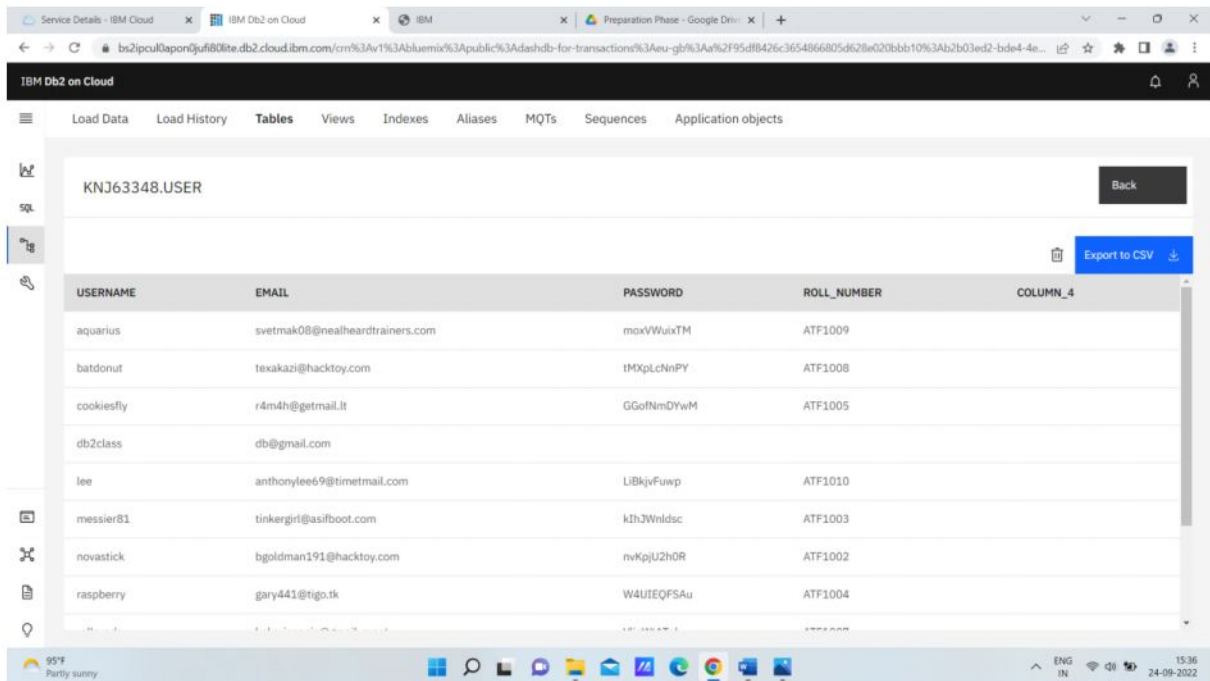


The screenshot shows the IBM Db2 on Cloud console interface. The 'My script' tab is active, displaying an SQL script. The script contains an INSERT statement. The 'Run all' button is visible in the top right corner of the script editor. Below the script editor, the 'History' tab shows the execution history of the script.

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

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



IBM Db2 on Cloud

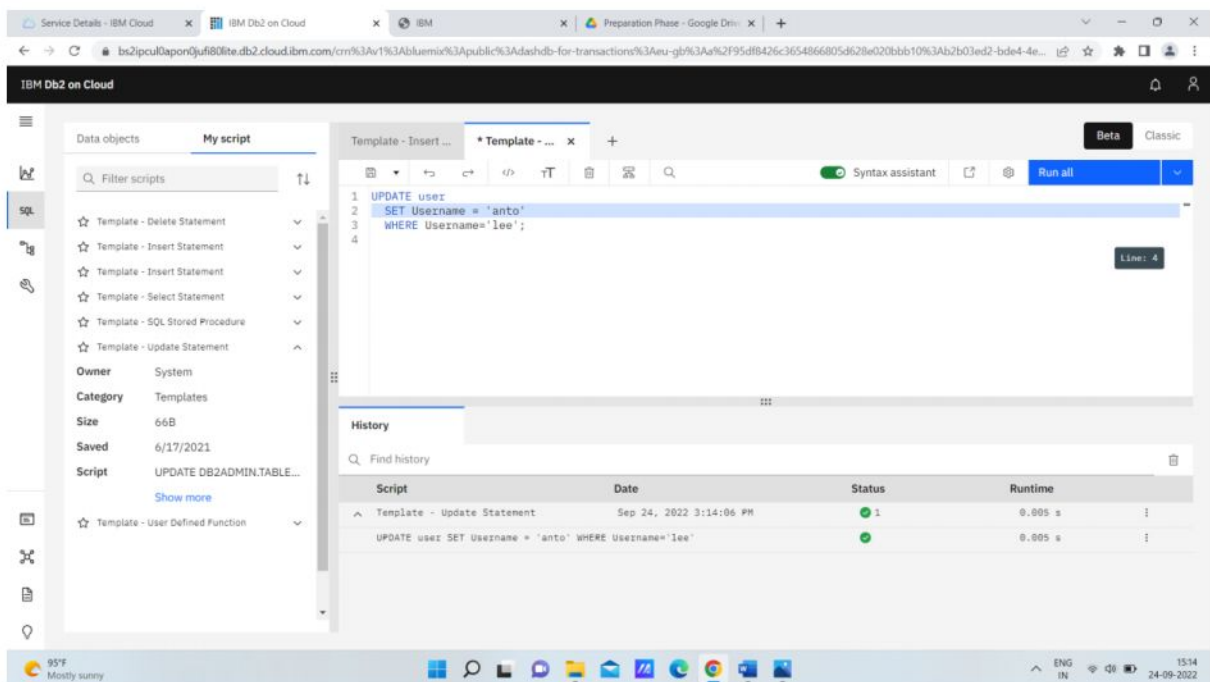
Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

KNJ63348.USER Back

Export to CSV

USERNAME	EMAIL	PASSWORD	ROLL_NUMBER	COLUMN_4
aquarius	svetmak08@nealheardtrainers.com	moxVWuixTM	ATF1009	
batdonut	texakazi@hacktoy.com	IMXpLcNnPY	ATF1008	
cookiesfly	r4m4h@getmail.it	GGofNmDYwM	ATF1005	
db2class	db@gmail.com			
lee	anthonylee69@timetmail.com	LiBkxvFuwp	ATF1010	
messier81	tinkergirl@asifboot.com	kIhJWnldsc	ATF1003	
novastick	bgoldman191@hacktoy.com	nvKpiU2h0R	ATF1002	
raspberry	gary441@tigo.tk	W4UIEQFSAu	ATF1004	

Update Query



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 - ... x +

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

Update Query Output

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

bs2ipcul0apon0ufi80lite.db2.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transaction%3Aeu-gb%3A%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@tinetmail.com	LiBkivFuwp	ATF1010	
aquarius	svetmak08@nealheardtrainers.com	moxVWuixTM	ATF1009	
batdonut	texakazi@hacktoy.com	tMXpLcNnPY	ATF1008	
cookiesfly	r4m4h@getmail.it	GGofNmDYwM	ATF1005	
db2class	db@gmail.com			
messier81	tinkergirl@asifboot.com	kIhJWnlsc	ATF1003	
novastick	bgoldman191@hacktoy.com	nvKpiU2h0R	ATF1002	
raspberry	gary441@tigo.tk	W4UIEQFSau	ATF1004	

95°F Partly sunny

Delete Query

bs2ipcul0apon0ufi80lite.db2.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transaction%3Aeu-gb%3A%2F95df8426c3654866805d628e020bbb10%3Ab2b03ed2-bde4-4e...

IBM Db2 on Cloud

Data objects My script

Filter scripts

- Template - Delete 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 - ...

Beta Classic

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

bs2ipcul0apon0ufi80lite.db2.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transaction%3Aeu-gb%3A%2F95df8426c3654866805d628e020bbb10%3Ab2b03ed2-bde4-4e...

IBM Db2 on Cloud

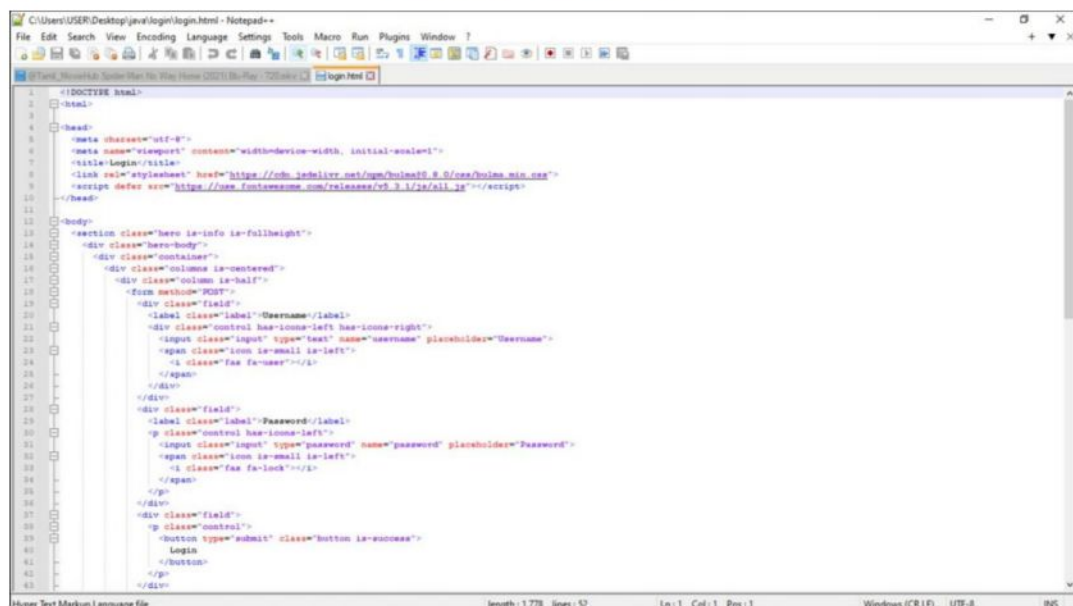
Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

KNJ63348.USER

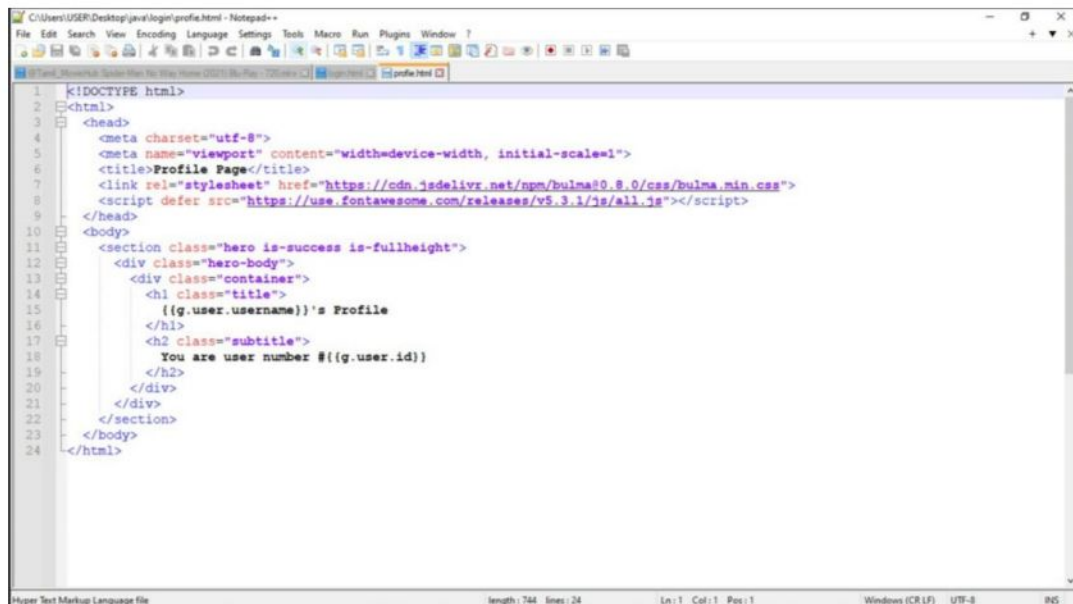
Back

Export to CSV

USERNAME	EMAIL	PASSWORD	ROLL_NUMBER	COLUMN_4
anto	anthonylee69@tinetmail.com	LiBkivFuwp	ATF1010	
aquarius	svetmak08@nealheardtrainers.com	moxVWuixTM	ATF1009	
batdonut	texakazi@hacktoy.com	tMXpLcNnPY	ATF1008	
cookiesfly	r4m4h@getmail.it	GGofNmDYwM	ATF1005	
messier81	tinkergirl@asifboot.com	kIhJWnlsc	ATF1003	
novastick	bgoldman191@hacktoy.com	nvKpiU2h0R	ATF1002	
raspberry	gary441@tigo.tk	W4UIEQFSau	ATF1004	
rollsodya	kokorinsanja@gmailvn.net	V'inWt~Tql	ATF1007	

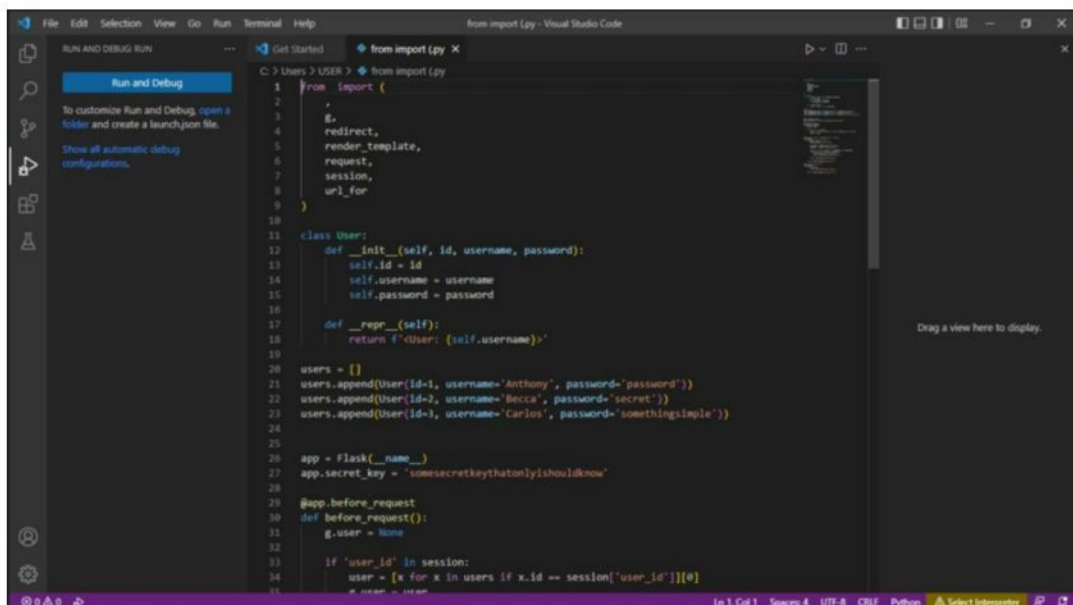


Source code for welcome page



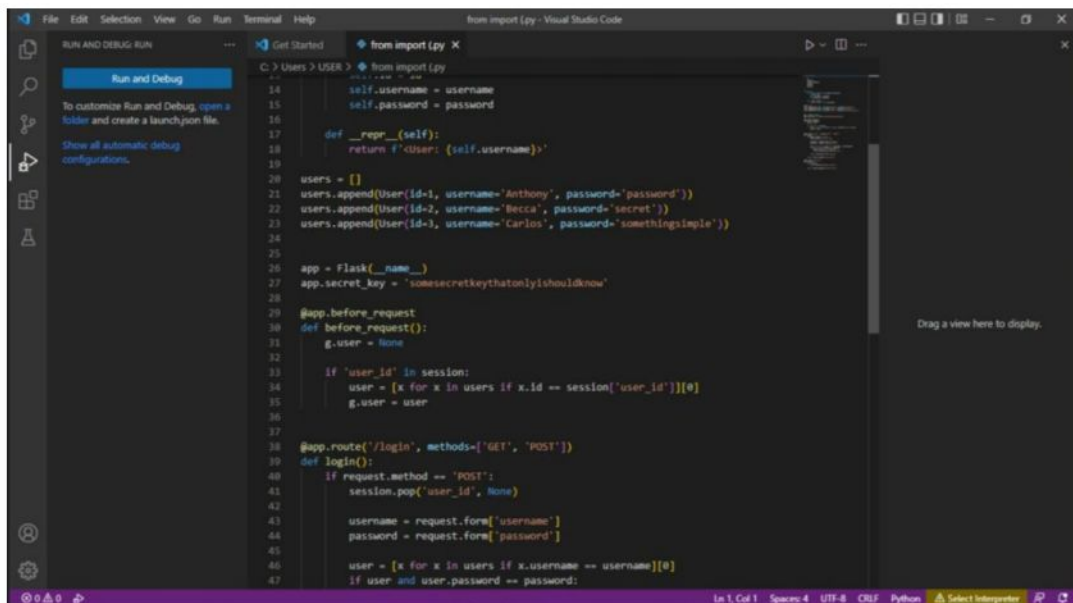
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Profile Page</title>
7   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@0.8.0/css/bootstrap.min.css">
8   <script defer src="https://use.fontawesome.com/releases/v5.3.1/js/all.js"></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



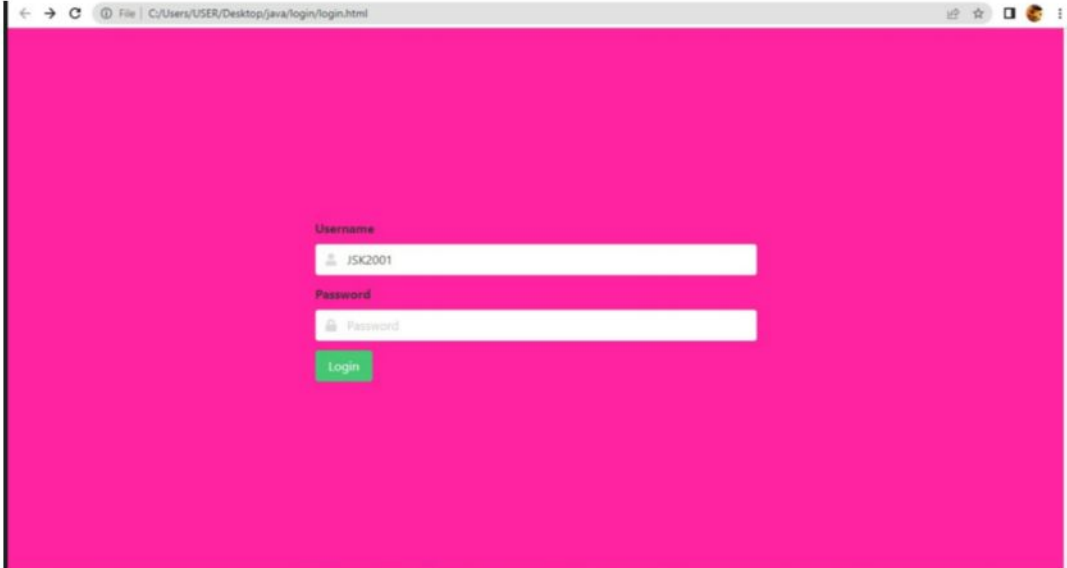
```
1 from import (
2   ,
3   ,
4   redirect,
5   render_template,
6   request,
7   session,
8   url_for
9 )
10
11 class User:
12     def __init__(self, id, username, password):
13         self.id = id
14         self.username = username
15         self.password = password
16
17     def __repr__(self):
18         return f'User: {self.username}'
19
20 users = []
21 users.append(User(id=1, username='Anthony', password='password'))
22 users.append(User(id=2, username='Becca', password='secret'))
23 users.append(User(id=3, username='Carlos', password='somethingimple'))
24
25 app = Flask(__name__)
26 app.secret_key = 'somesecretkeythatonlyishouldknow'
27
28 @app.before_request
29 def 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 @app.route('/login', methods=['GET', 'POST'])
38 def login():
39     if request.method == 'POST':
40         session.pop('user_id', None)
41
42         username = request.form['username']
43         password = request.form['password']
44
45         user = [x for x in users if x.username == username][0]
46         if user and user.password == password:
```

Flask code



```
14 self.username = username
15 self.password = password
16
17 def __repr__(self):
18     return f'User: {self.username}'
19
20 users = []
21 users.append(User(id=1, username='Anthony', password='password'))
22 users.append(User(id=2, username='Becca', password='secret'))
23 users.append(User(id=3, username='Carlos', password='somethingimple'))
24
25 app = Flask(__name__)
26 app.secret_key = 'somesecretkeythatonlyishouldknow'
27
28 @app.before_request
29 def 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 @app.route('/login', methods=['GET', 'POST'])
38 def login():
39     if request.method == 'POST':
40         session.pop('user_id', None)
41
42         username = request.form['username']
43         password = request.form['password']
44
45         user = [x for x in users if x.username == username][0]
46         if user and user.password == password:
```

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 white rectangular area containing the login form. The form has two input fields: the first is labeled "Username" and contains the text "JSK2001"; the second is labeled "Password" and contains the text "Password". Below these fields is a green button with the text "Login". The browser's address bar shows the file path "C:/Users/USER/Desktop/java/login/login.html".

Username

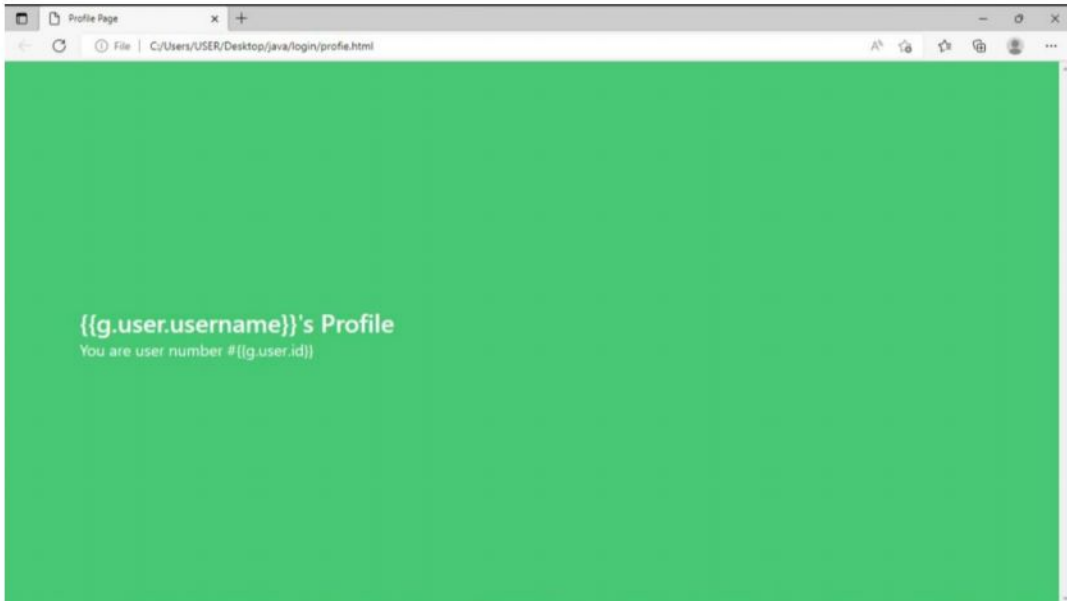
JSK2001

Password

Password

Login

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 white text that reads "{{g.user.username}}'s Profile" followed by "You are user number #{{g.user.id}}". The browser's address bar shows the file path "C:/Users/USER/Desktop/java/login/profile.html".

{{g.user.username}}'s Profile

You are user number #{{g.user.id}}