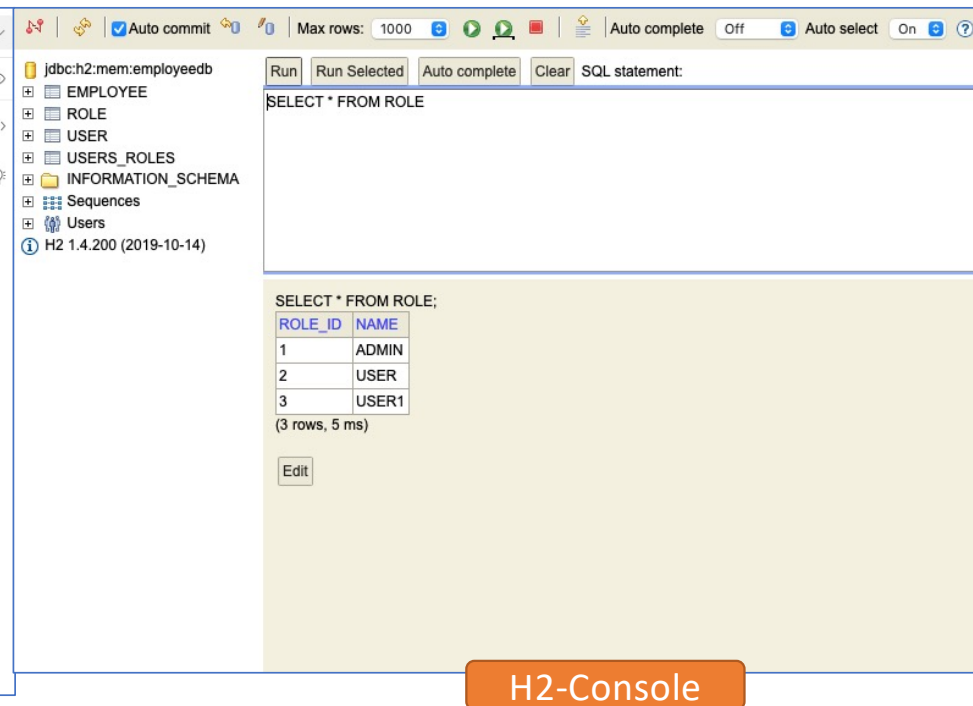
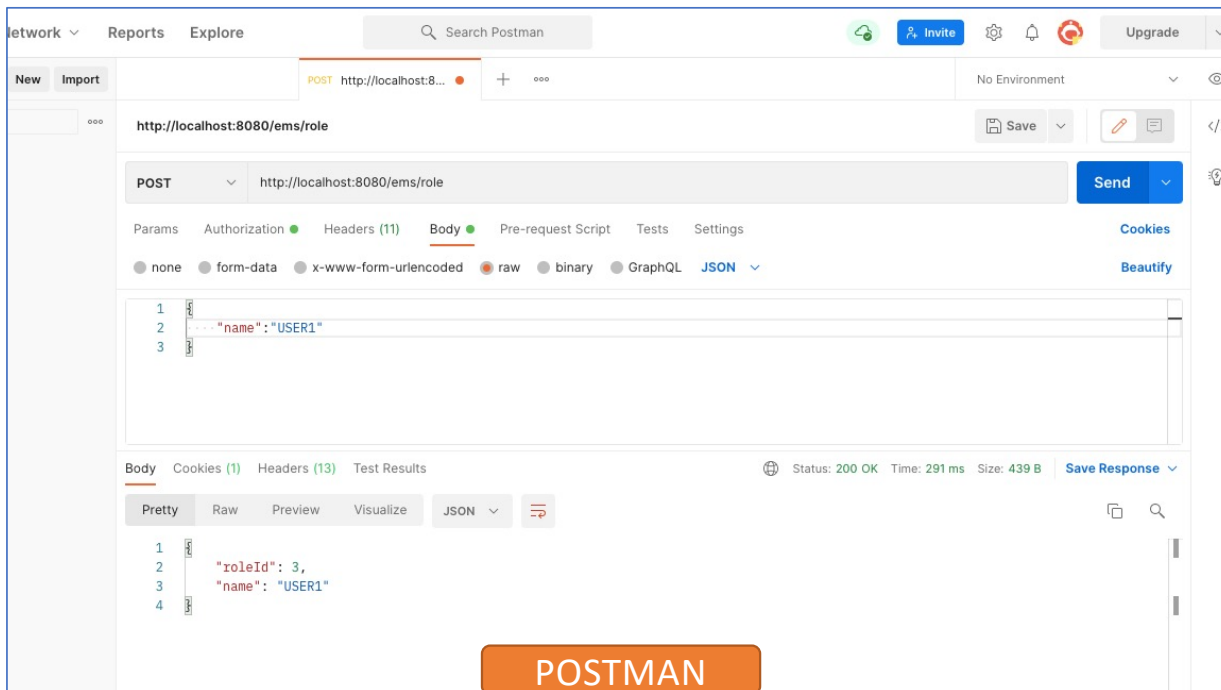


Graded Coding Assignment-6 Riajuddin_RestAPIAssignmentSolution_Screenshots

1. Adding A new Role

URL: <http://localhost:8080/ems/role>



2. Adding User

URL: **http://localhost:8080/ems/user**

Postman interface showing a POST request to `http://localhost:8080/ems/user`. The request body is a JSON object:

```
1 {
2   "username": "riyaz",
3   "password": "riyaz",
4   "roles": [
5     { "name": "USER1" }
6   ]
7 }
```

The response status is 200 OK, Time: 177 ms, Size: 592 B. The response body is a JSON object:

```
1 {
2   "userId": 3,
3   "username": "riyaz",
4   "password": "$2a$10$kFF/B3/VA/rk/SJv.207e0sD2SwuEHjW79K0z0LTrU9EYSROXgtKy",
5   "role": [
6     {
7       "roleId": 4,
8       "name": "USER1"
9     }
10  ],
11  "roles": [
12    {
13      "roleId": 4,
14      "name": "USER1"
15    }
16  ]
17 }
```

POSTMAN

H2-Console interface showing the SQL statement `SELECT * FROM USER`. The results are displayed in a table:

USER_ID	PASSWORD	USERNAME
1	\$2a\$12\$uir4L2hC7skcJ6u/UqIBZejcX2yZZOpYxBUnq3FPnH/AwP4xmSQLu	ADMIN
2	\$2a\$12\$XwWyuI08szaAixB7JNUI3eqL6z6aT8rRexLTmGAERC0fyBaa.Oclu	USER
3	\$2a\$10\$kFF/B3/VA/rk/SJv.207e0sD2SwuEHjW79K0z0LTrU9EYSROXgtKy	riyaz

(3 rows, 4 ms)

H2-Console

3. Adding New Employee
URL: <http://localhost:8080/ems/employee>

Postman interface showing a POST request to `http://localhost:8080/ems/employee`. The request body is a JSON object:

```
{  "firstName": "Charan",  "lastName": "suri",  "emailId": "charan@gmail.com"}
```

The response status is 200 OK, Time: 84 ms, Size: 491 B. The response body is a JSON object:

```
{  "empId": 3,  "firstName": "Charan",  "lastName": "suri",  "emailId": "charan@gmail.com"}
```

POSTMAN

H2-Console before adding charan

SQL statement: `SELECT * FROM EMPLOYEE`

Action	EMP_ID	EMAIL_ID	FIRST_NAME	LAST_NAME
	1	sk.riyaz@gmail.com	Sheik	riyaz
	2	gl@gmail.com	great	learning

(2 rows, 9 ms)

H2-Console after adding charan

SQL statement: `SELECT * FROM EMPLOYEE;`

EMP_ID	EMAIL_ID	FIRST_NAME	LAST_NAME
1	sk.riyaz@gmail.com	Sheik	riyaz
2	gl@gmail.com	great	learning
3	charan@gmail.com	Charan	suri

(3 rows, 5 ms)

4. Adding A New Employee

URL: <http://localhost:8080/ems/employees>

GET http://localhost:80...

No Environment

http://localhost:8080/ems/employees

GET http://localhost:8080/ems/employees

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
-----	-------	-------------	-----------

Body Cookies (1) Headers (13) Test Results Status: 200 OK Time: 66 ms Size: 654 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "empId": 1,
4     "firstName": "Sheik",
5     "lastName": "riyaz",
6     "emailId": "sk.riyaz@gmail.com"
7   },
8   {
9     "empId": 2,
10    "firstName": "great",
11    "lastName": "learning",
12    "emailId": "gl@gmail.com"
13  },
14  {
15    "empId": 3,
16    "firstName": "Charan",
17    "lastName": "suri",
18    "emailId": "charan@gmail.com"
19  }
20 }
```

POSTMAN

Auto commit

Max rows: 1000

Auto complete Off

jdbc:h2:mem:employeeedb

EMPLOYEE

ROLE

USER

USERS_ROLES

INFORMATION_SCHEMA

Sequences

Users

H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM EMPLOYEE

SELECT * FROM EMPLOYEE;

EMP_ID	EMAIL_ID	FIRST_NAME	LAST_NAME
1	sk.riyaz@gmail.com	Sheik	riyaz
2	gl@gmail.com	great	learning
3	charan@gmail.com	Charan	suri

(3 rows, 10 ms)

Edit

H2-Console

5. Finding a employee with ID (1&3)

URL: <http://localhost:8080/ems/employee/1> & <http://localhost:8080/ems/employee/3>

ReportsExplore

Search Postman

GET http://localhost:80...

+

...

http://localhost:8080/ems/employee/1

GET http://localhost:8080/ems/employee/1

ParamsAuthorizationHeaders (11)BodyPre-request ScriptTestsSettings

Query Params

KEYVALUEDESCRIPTION

BodyCookies (1)Headers (13)Test ResultsStatus: 200 OKTime: 55 ms

PrettyRawPreviewVisualizeJSON

```
1 {
2   "empId": 1,
3   "firstName": "Sheik",
4   "lastName": "riyaz",
5   "emailId": "sk.riyaz@gmail.com"
6 }
```

Finding with ID-1

GET http://localhost:80...

+

...

http://localhost:8080/ems/employee/3

GET http://localhost:8080/ems/employee/3

ParamsAuthorizationHeaders (11)BodyPre-request ScriptTestsSettings

Query Params

KEYVALUEDESCRIPTION

BodyCookies (1)Headers (13)Test ResultsStatus: 200 OKTime: 55 ms

PrettyRawPreviewVisualizeJSON

```
1 {
2   "empId": 3,
3   "firstName": "Charan",
4   "lastName": "suri",
5   "emailId": "charan@gmail.com"
6 }
```

Finding with ID-3

6. Updating Employee details-charan(id-3)
URL: <http://localhost:8080/ems/employee>

```
1 {
2   "empId": 3,
3   "firstName": "suresh",
4   "lastName": "ramesh",
5   "emailId": "sr@gl.com"
6 }
```

POSTMAN
Before updating ID-3 from suresh to charan

```
1 {
2   "empId": 3,
3   "firstName": "Charan",
4   "lastName": "suri",
5   "emailId": "charan@gmail.com"
6 }
```

Auto commit | Max rows: 1000 | Auto complete

jdbc:h2:mem:employeeedb

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM EMPLOYEE

SELECT * FROM EMPLOYEE;

EMP_ID	EMAIL_ID	FIRST_NAME	LAST_NAME
1	sk.riyaz@gmail.com	Sheik	riyaz
2	gl@gmail.com	great	learning
3	sr@gl.com	suresh	ramesh

(3 rows, 6 ms)

H2-Console
after updating ID-3 from charan to suresh

PUT http://localhost:8080/ems/employee

Params Authorization Headers (11) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "empId": 3,
3   "firstName": "suresh",
4   "lastName": "ramesh",
5   "emailId": "sr@gl.com"
6 }
```

Body Cookies (1) Headers (13) Test Results

Status: 200 OK Time: 54 ms Size: 486 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "empId": 3,
3   "firstName": "suresh",
4   "lastName": "ramesh",
5   "emailId": "sr@gl.com"
6 }
```

POSTMAN response body
after updating ID-3 from charan to suresh

7. Deleting Employee -suresh(id-3)

URL: <http://localhost:8080/ems/employees/3>

The Postman interface shows a DELETE request to `http://localhost:8080/ems/employees/3`. The request body is a JSON object: `{ "empId": 3, "firstName": "suresh", "lastName": "ramesh", "emailId": "sr@gl.com" }`. The response status is 200 OK, with a time of 77 ms and a size of 435 B. The response body is: `1 Deleted employee id - 3`.

POSTMAN

The H2-Console interface shows the database schema for `jdbc:h2:mem:employeeedb`. The schema includes tables: `EMPLOYEE`, `ROLE`, `USER`, `USERS_ROLES`, `INFORMATION_SCHEMA`, `Sequences`, and `Users`. The version is `H2 1.4.200 (2019-10-14)`. The SQL statement `SELECT * FROM EMPLOYEE` is executed, resulting in 2 rows and 3 ms. The result is displayed in a table:

EMP_ID	EMAIL_ID	FIRST_NAME	LAST_NAME
1	sk.riyaz@gmail.com	Sheik	riyaz
2	gl@gmail.com	great	learning

(2 rows, 3 ms)

An `Edit` button is available below the table.

H2-Console

8. Searching Employee with first name

URL: <http://localhost:8080/ems/employee/search/sheik> & <http://localhost:8080/ems/employee/search/greaT>

GET

http://localhost:8080/ems/employee/search/sheik

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE
Key	Value

Body

Cookies (1)

Headers (13)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   {
3     "empId": 1,
4     "firstName": "Sheik",
5     "lastName": "riyaz",
6     "emailId": "sk.riyaz@gmail.com"
7   }
8 }
```

Output for firstname - Sheik

GET

http://localhost:8080/ems/employee/search/greaT

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCR
Key	Value	Descrip

Body

Cookies (1)

Headers (13)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   {
3     "empId": 2,
4     "firstName": "great",
5     "lastName": "learning",
6     "emailId": "gl@gmail.com"
7   }
8 }
```

Output for firstname - great

7. Sorting(Asc & Desc) Employees with first name

URL: <http://localhost:8080/ems/employee/sort?order=asc> & <http://localhost:8080/ems/employee/sort?order=desc>

GET

http://localhost:8080/ems/employee/sort?order=asc

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	order	asc	
	Key	Value	Description

Body

Cookies (1)

Headers (13)

Test Results

Status: 200 OK

Time:

Pretty

Raw

Preview

Visualize

JSON

```
1  [
2    {
3      "empId": 2,
4      "firstName": "great",
5      "lastName": "learning",
6      "emailId": "gl@gmail.com"
7    },
8    {
9      "empId": 1,
10     "firstName": "sheik",
11     "lastName": "riyaz",
12     "emailId": "sk.riyaz@gmail.com"
13   }
14 ]
```

Response in Ascending

GET

http://localhost:8080/ems/employee/sort?order=desc

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	order	desc	
	Key	Value	Description

Body

Cookies (1)

Headers (13)

Test Results

Status: 200 OK

Time:

Pretty

Raw

Preview

Visualize

JSON

```
1  [
2    {
3      "empId": 1,
4      "firstName": "sheik",
5      "lastName": "riyaz",
6      "emailId": "sk.riyaz@gmail.com"
7    },
8    {
9      "empId": 2,
10     "firstName": "great",
11     "lastName": "learning",
12     "emailId": "gl@gmail.com"
13   }
14 ]
```

Response in descending