

Project Name : Student Management Rest API

Completed by: Sanket Chandrakant Bondre

1) Pre given data:

localhost:8080/h2-console/login.do?sessionId=078e8bf0ea48ae9190926db883aca479

Auto commitMax rows: 1000Auto completeOffAuto selectOn

jdbc:h2:mem:testdb

SCHOOLTEACHERINFORMATION_SCHEMAUsersH2 2.3.232 (2024-08-11)

RunRun SelectedAuto completeClearSQL statement

SELECT * FROM TEACHER

SELECT * FROM TEACHER

ID	AGE	DOB	EXPERIENCE	QUALIFICATION	TEACHER_ADDRESS	TEACHER_NAME
1	45	1980-05-15	20	PHD	Pune	Sanket Bondre
2	50	1975-03-10	25	MSC	Mumbai	Mohan Mishra
3	38	1985-08-20	15	BTECH	Chennai	Ram Babu
4	42	1981-11-30	18	MTECH	Bangalore	Maresh Goswal
5	48	1977-04-05	22	PHD	Delhi	Arun Rampal
6	34	1986-12-14	12	MSC	Pune	Maresh Shinde
7	40	1982-01-25	17	BSC	Kolkata	Gopi Kumar
8	37	1983-09-12	14	MTECH	Chennai	Venkaresh

(8 rows, 1 ms)

Edit

localhost:8080/h2-console/login.do?sessionId=078e8bf0ea48ae9190926db883aca479

Auto commitMax rows: 1000Auto completeOffAuto selectOn

jdbc:h2:mem:testdb

SCHOOLTEACHERINFORMATION_SCHEMAUsersH2 2.3.232 (2024-08-11)

RunRun SelectedAuto completeClearSQL statement

SELECT * FROM SCHOOL

SELECT * FROM SCHOOL

ID	SCHOOL_ADDRESS	SCHOOL_NAME	TEACHER_ID
1	Hyderabad	SNDT	1
2	Mumbai	JWTS Highschool	2
3	Chennai	Springfield Academy	3
4	Bangalore	Maple Leaf School	4
5	Delhi	Crescent School	5
6	Pune	Fergusson College	6
7	Kolkata	MIT School	7
8	Chennai	VIT Academy	8

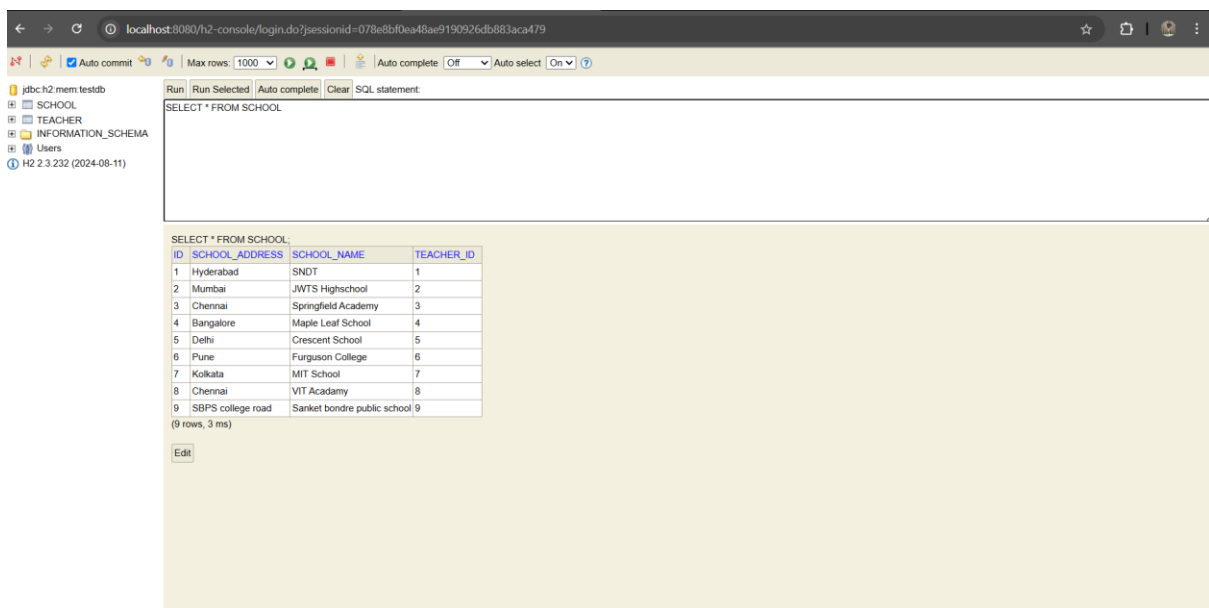
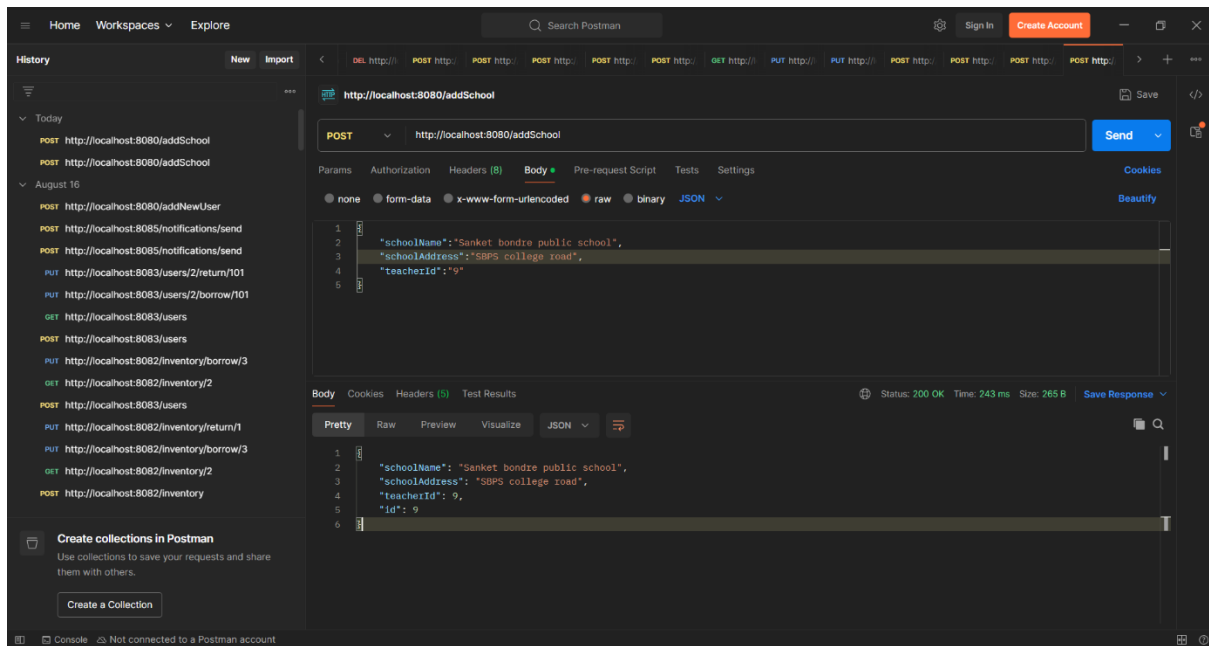
(8 rows, 0 ms)

Edit

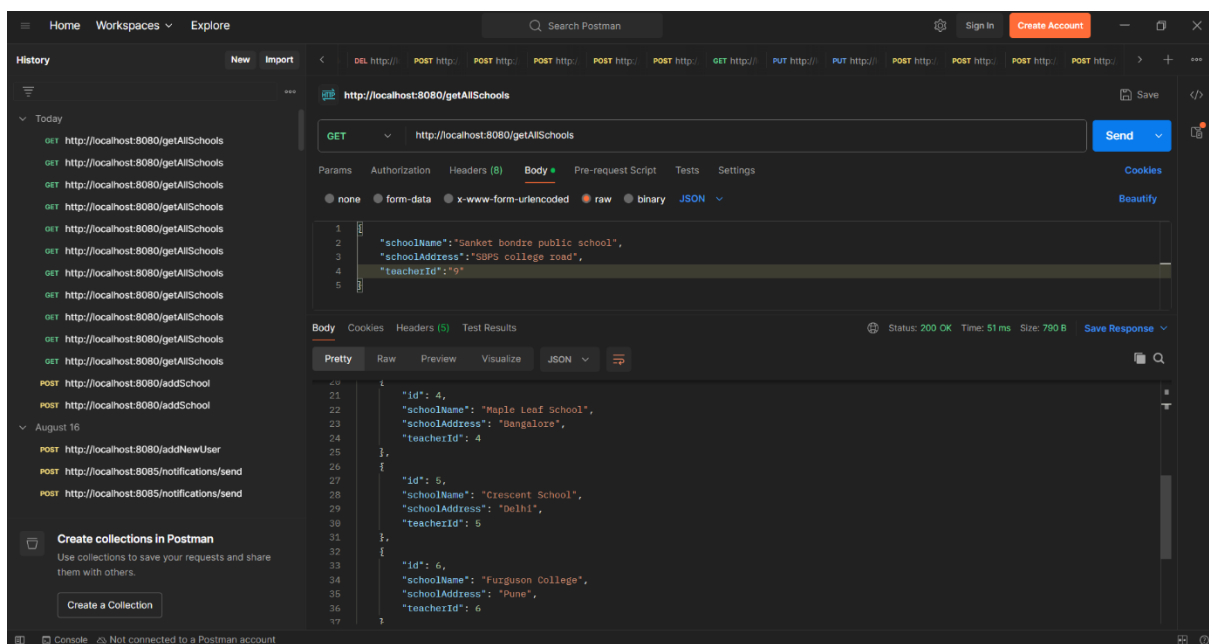
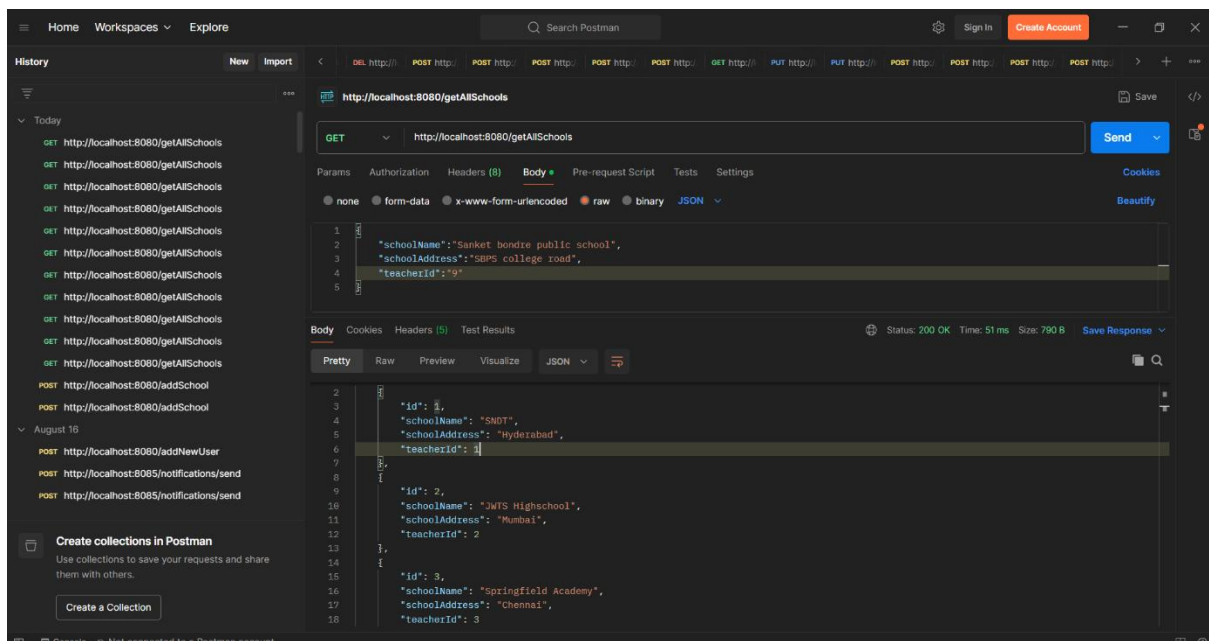
➔ I have Implemented following Endpoints(Postman) in my Project:

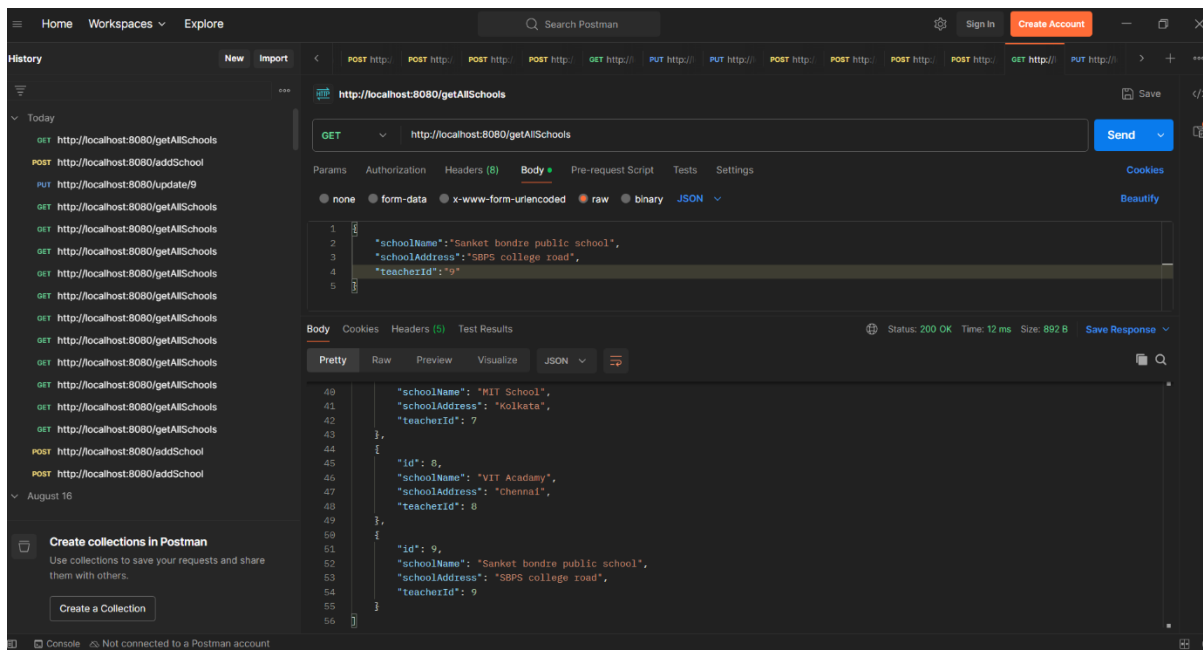
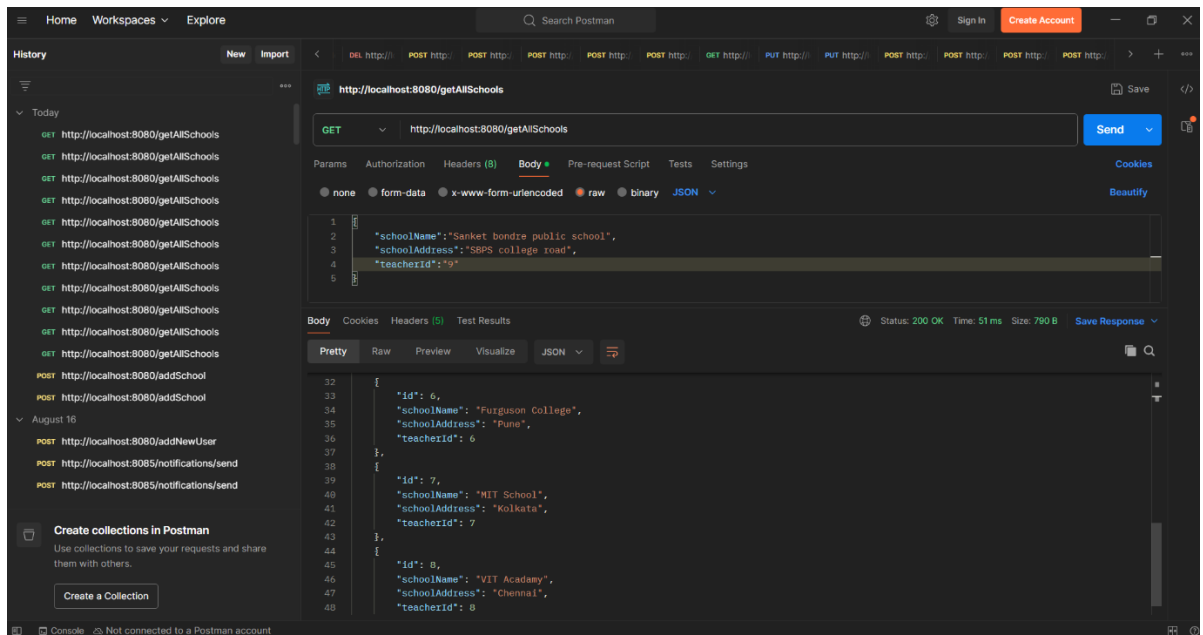
```
@PostMapping("/addSchool")
@GetMapping("getAllSchools")
@PutMapping("/update/{id}")
@PatchMapping("/patch/{id}")
@DeleteMapping("/deleteSchool/{id}")
@GetMapping("/school/{schoolAddress}")
@DeleteMapping("/school/{schoolAddress}")
@GetMapping("/search")
@PostMapping("/addTeacher")
@GetMapping("getAllTeachers")
@PutMapping("/updateTeacher/{id}")
@PatchMapping("/patchTeacher/{id}")
@Delete Mapping("/deleteTeacher/{id}")
@GetMapping("/teachers/{qualification}")
@Delete Mapping("delete/{teacherAddress}")
```

2)@PostMapping After("/addSchool")



3)@GetMapping After ("/getAISchools")





44) @PutMapping After ("/update/{id}")

The image shows the Postman application interface. On the left, the 'History' panel lists several requests, including PUT requests to `http://localhost:8080/update/9` and GET requests to `http://localhost:8080/getAllSchools`. The main panel displays a PUT request to `http://localhost:8080/update/9`. The 'Body' tab is selected, showing a JSON payload:

```
{  "id": 9,  "schoolName": "Sanket bondre public school",  "schoolAddress": "SBPS college road",  "teacherId": 9}
```

The response status is 200 OK, with a time of 21 ms and a size of 265 B. The 'Body' tab also shows the response in 'Pretty' format:

```
{  "id": 9,  "schoolName": "Sanket bondre public school",  "schoolAddress": "SBPS college road",  "teacherId": 9}
```

The image shows a database console interface. The top bar indicates the connection is to `localhost:8080/h2-console/login.do?sessionId=acb89a3fe70ea6e35620924af133f8b7`. The left sidebar shows the database schema with tables `SCHOOL`, `TEACHER`, `INFORMATION_SCHEMA`, and `Users`. The main area displays a SQL query:

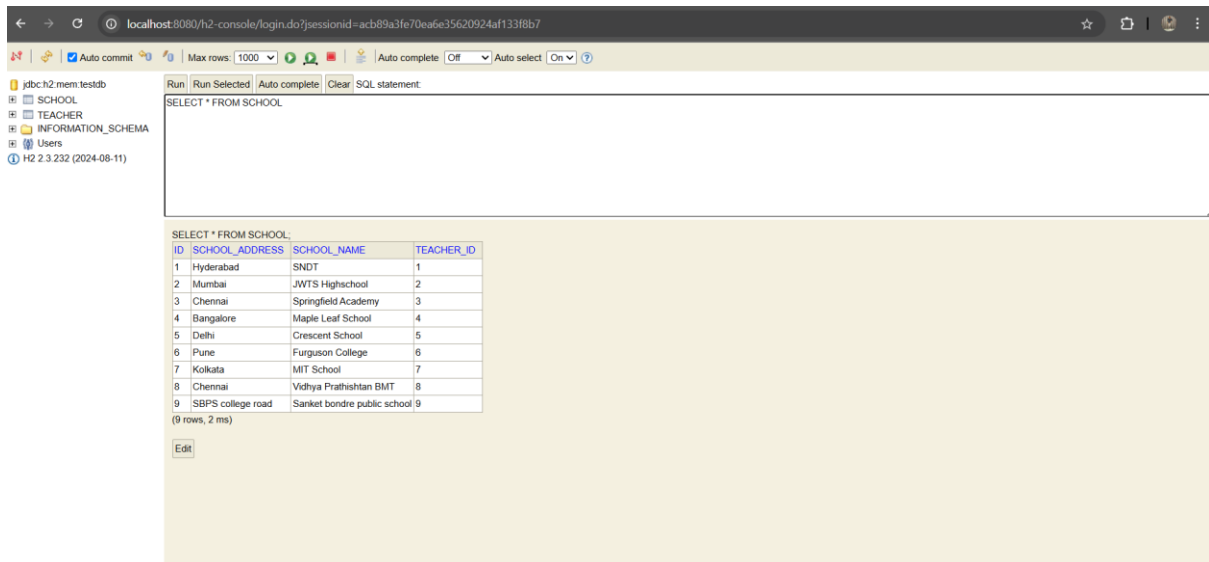
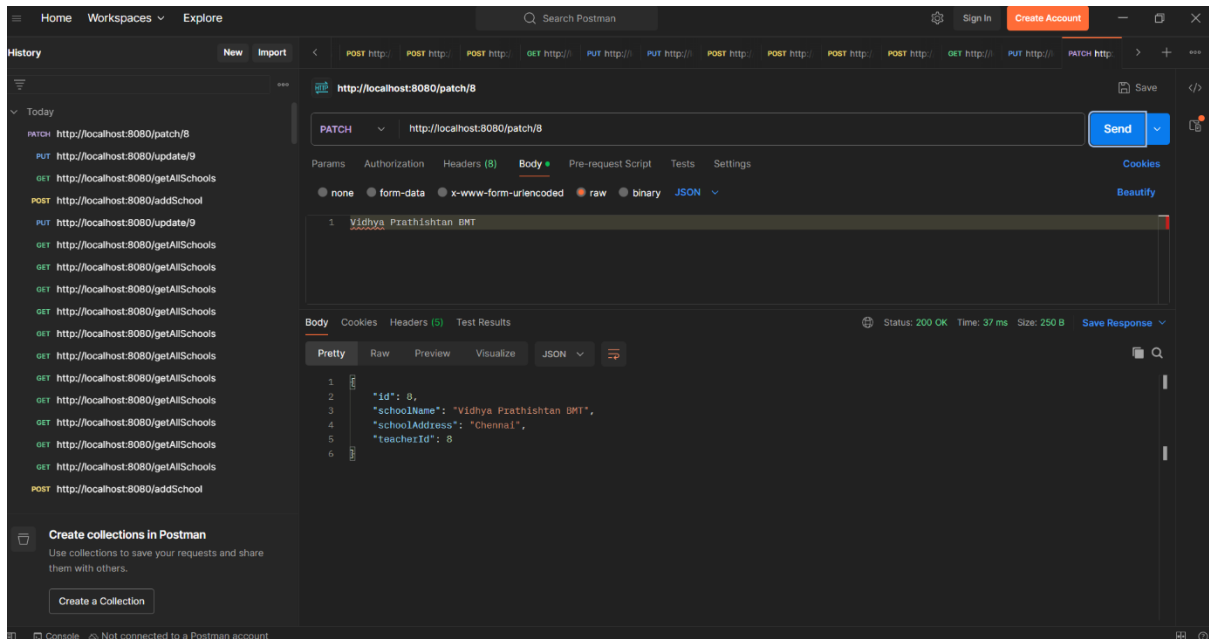
```
SELECT * FROM SCHOOL
```

The query results are shown in a table with 9 rows:

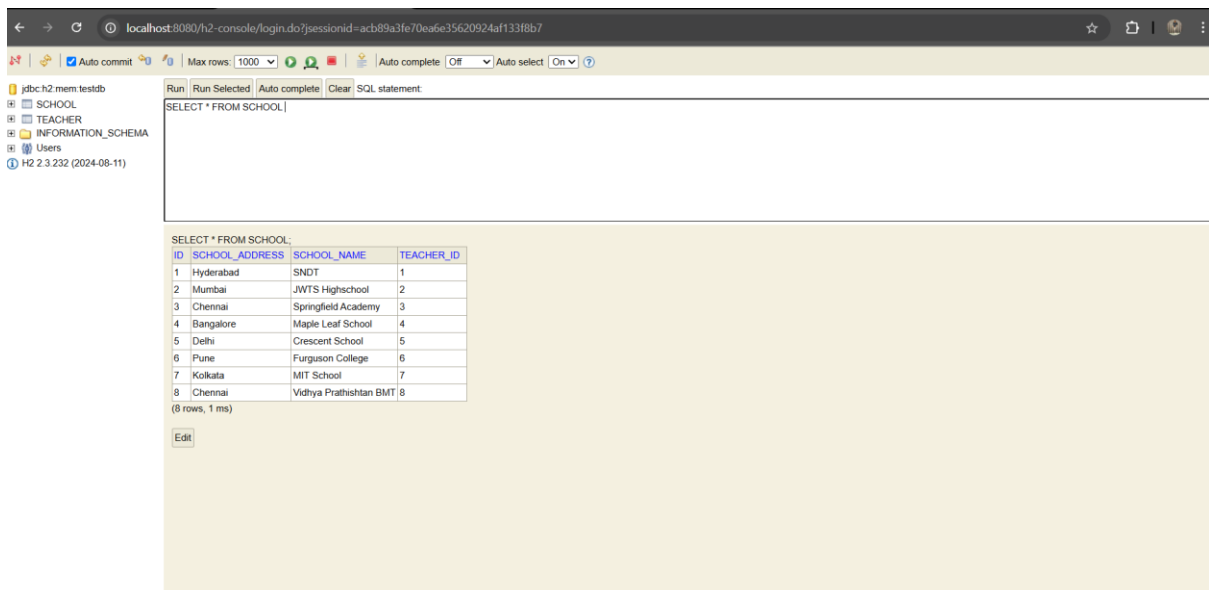
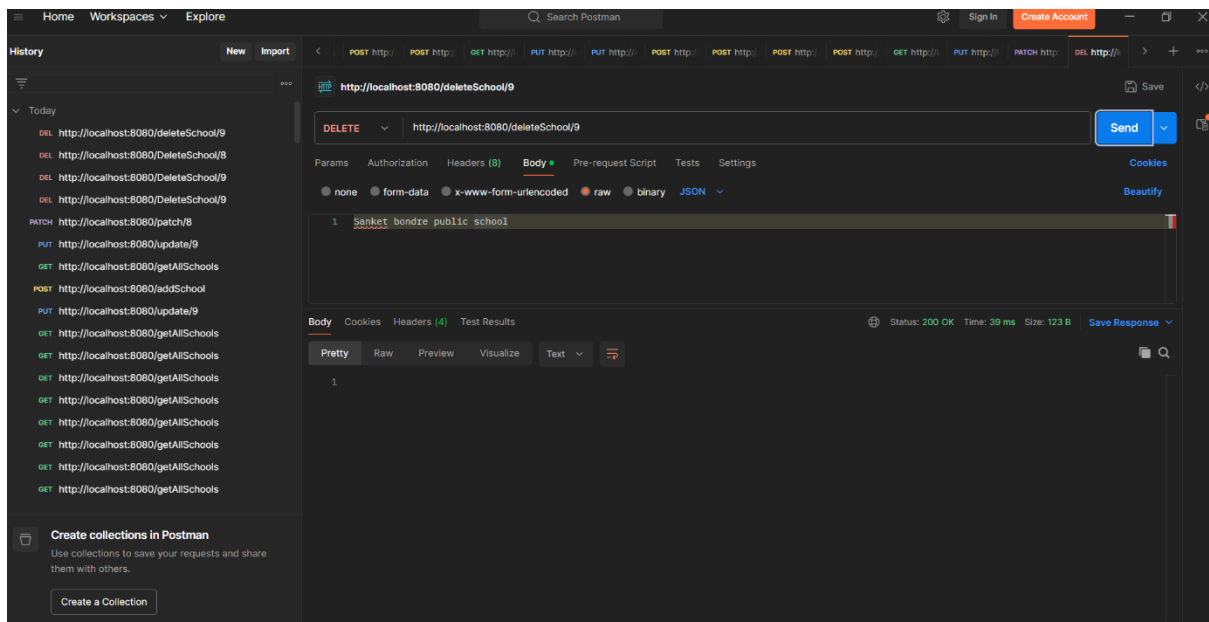
ID	SCHOOL_ADDRESS	SCHOOL_NAME	TEACHER_ID
1	Hyderabad	SNDDT	1
2	Mumbai	JWTS Highschool	2
3	Chennai	Springfield Academy	3
4	Bangalore	Maple Leaf School	4
5	Delhi	Crescent School	5
6	Pune	Ferguson College	6
7	Kolkata	MIT School	7
8	Chennai	VIT Academy	8
9	SBPS college road	Sanket bondre public school	9

The results are displayed as 9 rows, 0 ms. An 'Edit' button is visible at the bottom left of the results area.

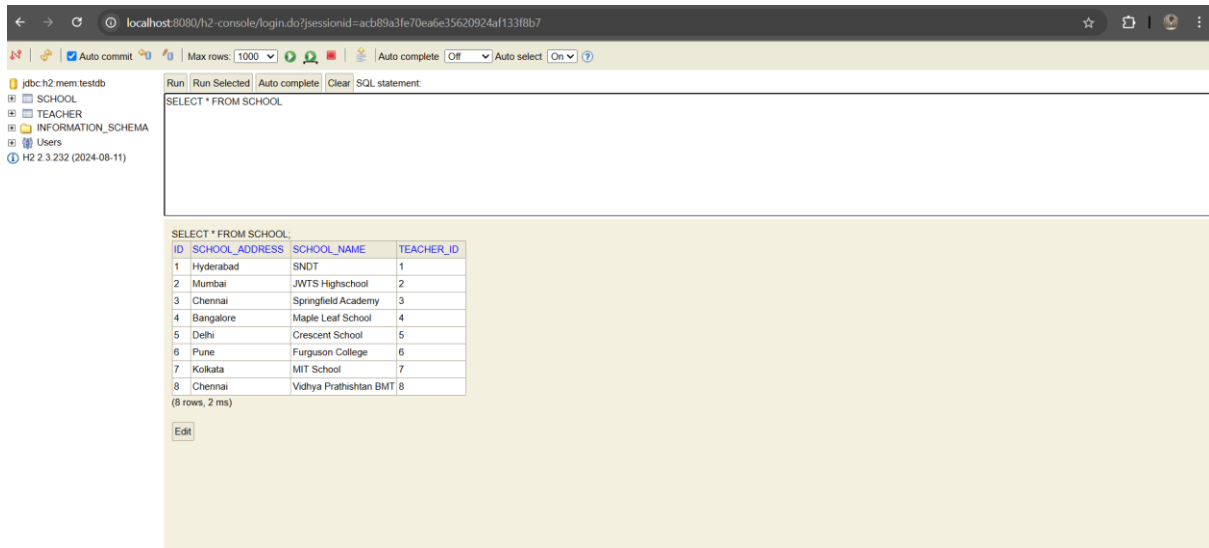
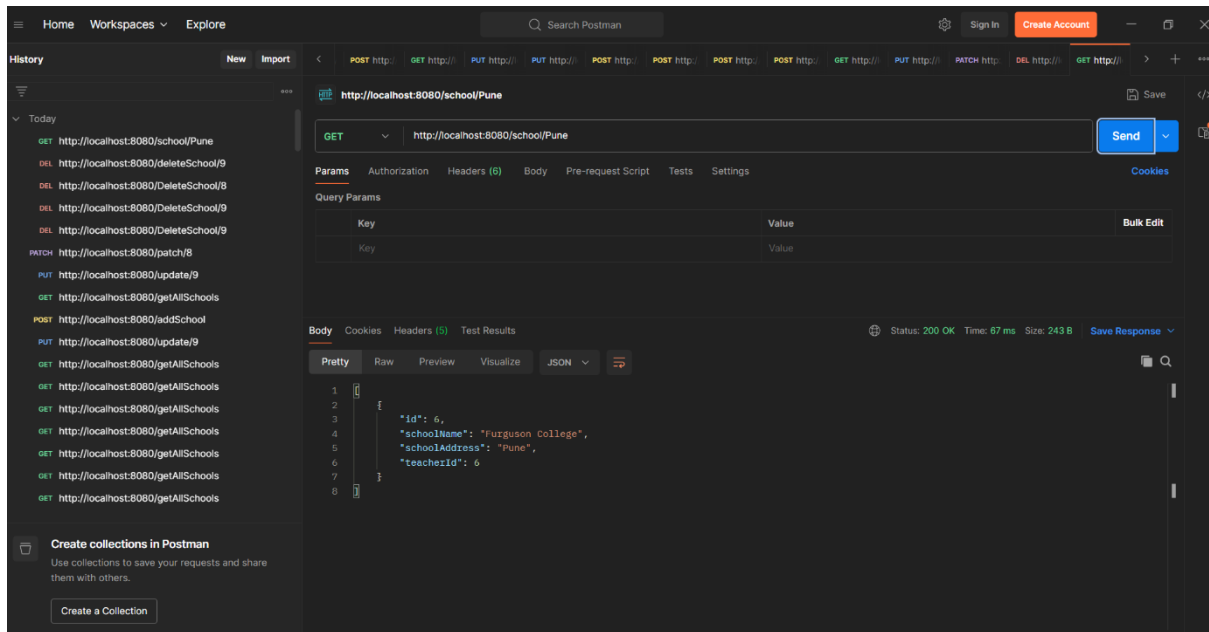
5) @PatchMapping("/patch/{id}")



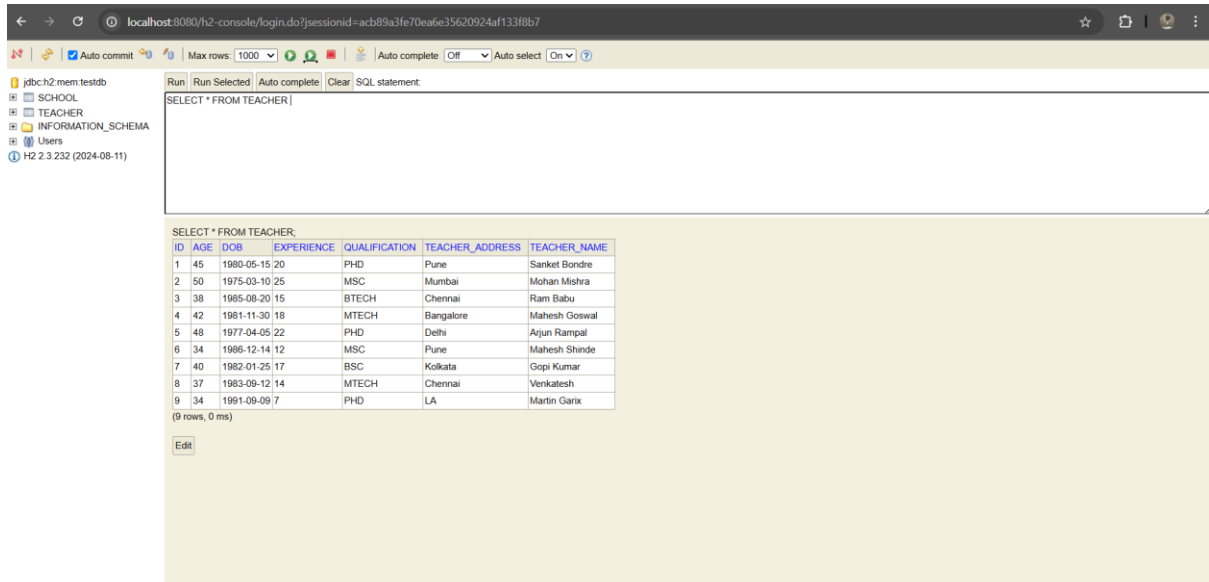
6) @DeleteMapping("/deleteSchool/{id}")



7) @GetMapping After("/school/{schoolAddress}")



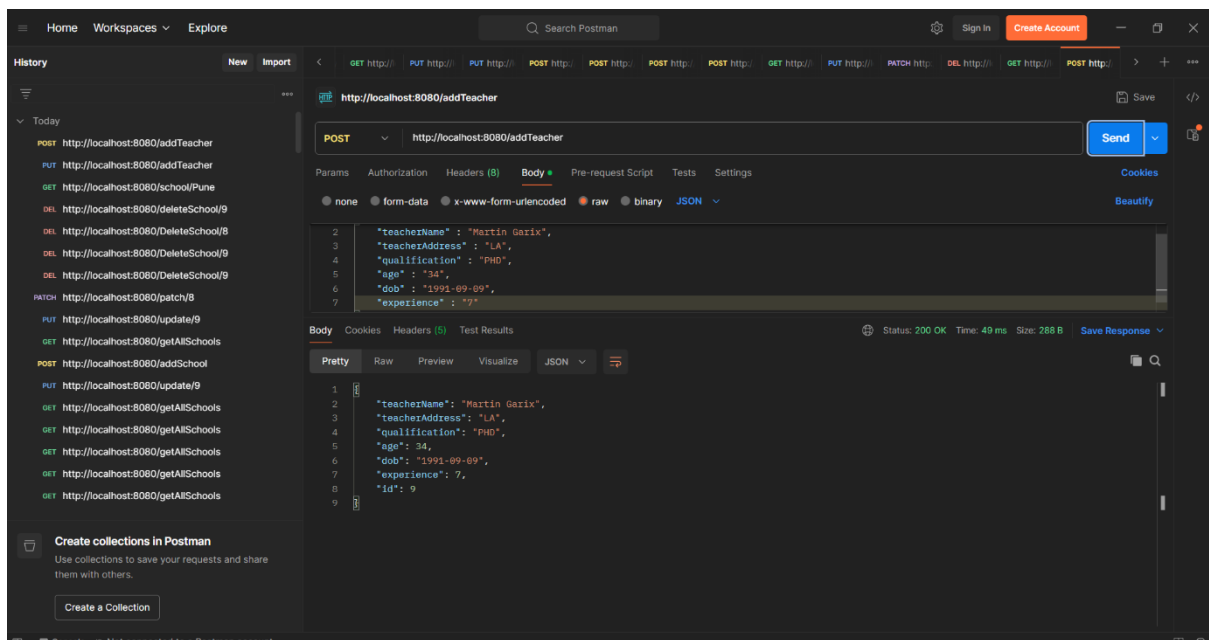
8) After("/addTeacher")



The screenshot shows a web browser window with the address bar displaying `localhost:8080/h2-console/login.do?sessionId=acb89a3fe70ea6e35620924af1338b67`. The browser's developer tools are open, showing the SQL statement `SELECT * FROM TEACHER;` and its results. The results are displayed in a table with 9 rows and 7 columns: ID, AGE, DOB, EXPERIENCE, QUALIFICATION, TEACHER_ADDRESS, and TEACHER_NAME.

ID	AGE	DOB	EXPERIENCE	QUALIFICATION	TEACHER_ADDRESS	TEACHER_NAME
1	45	1980-05-15	20	PHD	Pune	Sanket Bondre
2	50	1975-03-10	25	MSC	Mumbai	Mohan Mishra
3	38	1985-08-20	15	BTECH	Chennai	Ram Babu
4	42	1981-11-30	18	MTECH	Bangalore	Maresh Goswal
5	48	1977-04-05	22	PHD	Delhi	Arun Rampal
6	34	1986-12-14	12	MSC	Pune	Maresh Shinde
7	40	1982-01-25	17	BSC	Kolkata	Gopi Kumar
8	37	1983-09-12	14	MTECH	Chennai	Venkatesh
9	34	1991-09-09	7	PHD	LA	Martin Garix

Below the table, it indicates "(9 rows, 0 ms)" and an "Edit" button.



The screenshot shows the Postman application interface. The left sidebar displays a list of requests under the "Today" tab. The main panel shows a POST request to `http://localhost:8080/addTeacher` with a status of 200 OK. The response body is displayed in JSON format.

```
{  "teacherName": "Martin Garix",  "teacherAddress": "LA",  "qualification": "PHD",  "age": 34,  "dob": "1991-09-09",  "experience": 7,  "id": 9}
```

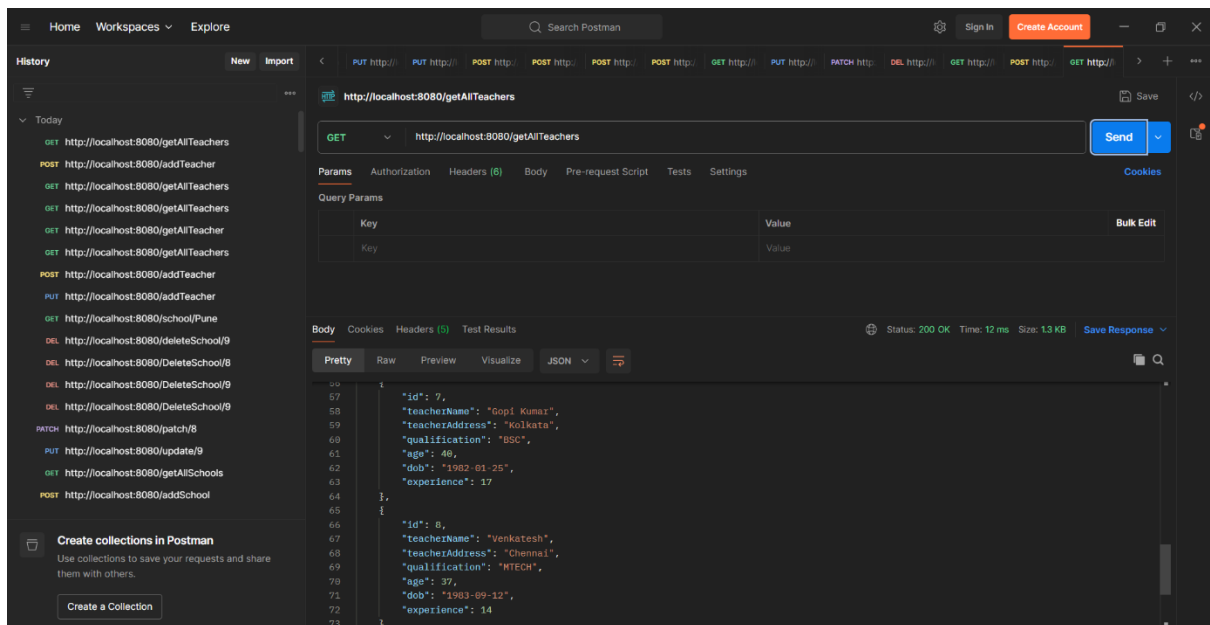
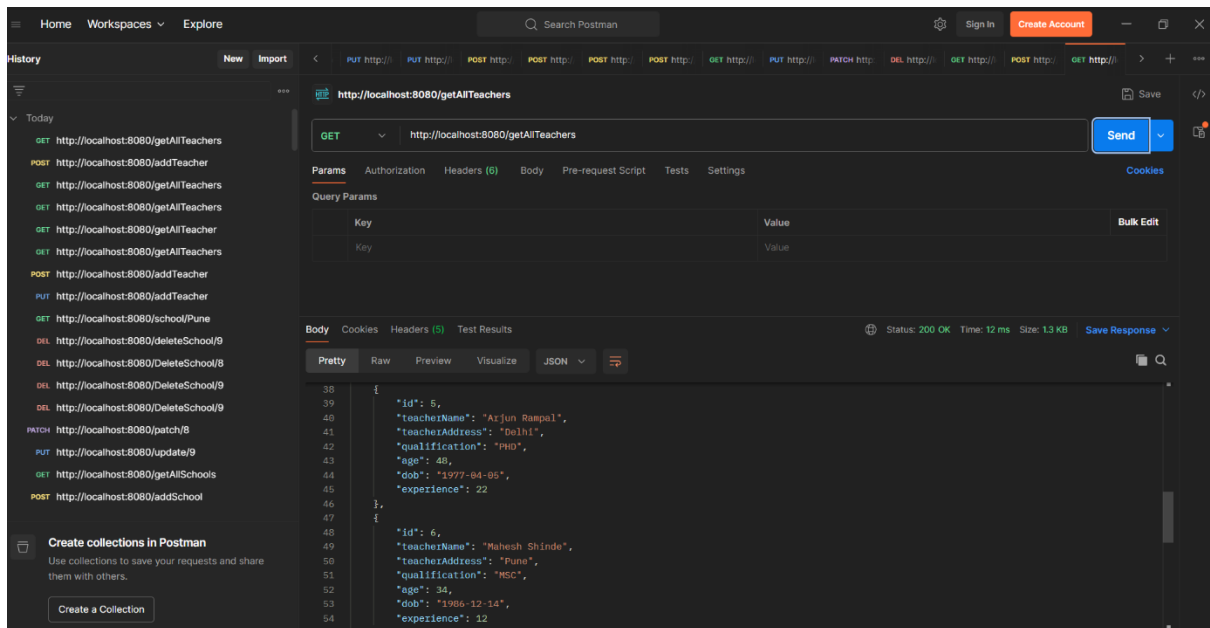
9) After (“getAllTeachers”)

The screenshot shows the Postman interface with the 'http://localhost:8080/getAllTeachers' endpoint selected. The response is a JSON array containing two teacher objects. The status is 200 OK, and the response size is 1.3 KB.

```
2 {
3   {
4     "id": 1,
5     "teacherName": "Sanket Bondre",
6     "teacherAddress": "Pune",
7     "qualification": "PND",
8     "age": 45,
9     "dob": "1968-05-15",
10    "experience": 28
11  },
12  {
13    "id": 2,
14    "teacherName": "Mohan Mishra",
15    "teacherAddress": "Mumbai",
16    "qualification": "MSc",
17    "age": 58,
18    "dob": "1975-03-10",
19    "experience": 25
20  }
21 }
```

The screenshot shows the Postman interface with the 'http://localhost:8080/getAllTeachers' endpoint selected. The response is a JSON array containing four teacher objects. The status is 200 OK, and the response size is 1.3 KB.

```
28 {
29   {
30     "id": 3,
31     "teacherName": "Ravi Babu",
32     "teacherAddress": "Chennai",
33     "qualification": "BTECH",
34     "age": 38,
35     "dob": "1985-08-28",
36     "experience": 15
37   },
38   {
39     "id": 4,
40     "teacherName": "Mahesh Goswal",
41     "teacherAddress": "Bangalore",
42     "qualification": "MTECH",
43     "age": 42,
44     "dob": "1981-11-30",
45     "experience": 18
46   }
47 }
```



HomeWorkspacesExplore

Search Postman

Sign InCreate Account

History

NewImport

PUT http://PUT http://POST http://POST http://POST http://POST http://GET http://PUT http://PATCH http://DEL http://GET http://POST http://GET http://

Send

Today

GET http://localhost:8080/getAllTeachers

POST http://localhost:8080/addTeacher

GET http://localhost:8080/getAllTeachers

GET http://localhost:8080/getAllTeacher

GET http://localhost:8080/getAllTeachers

POST http://localhost:8080/addTeacher

PUT http://localhost:8080/addTeacher

GET http://localhost:8080/school/Pune

DEL http://localhost:8080/deleteSchool/9

DEL http://localhost:8080/DeleteSchool/9

DEL http://localhost:8080/DeleteSchool/9

PATCH http://localhost:8080/patch/8

PUT http://localhost:8080/update/9

GET http://localhost:8080/getAllSchools

POST http://localhost:8080/addSchool

Create collections in Postman

Use collections to save your requests and share them with others.

Create a Collection

http://localhost:8080/getAllTeachers

Save

GET

http://localhost:8080/getAllTeachers

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

Key	Value
Key	Value

Bulk Edit

Body

Cookies

Headers (5)

Test Results

Status: 200 OKTime: 12 msSize: 1.3 KBSave Response

Pretty

Raw

Preview

Visualize

JSON

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

```
{
  "teacherName": "Venkatesh",
  "teacherAddress": "Chennai",
  "qualification": "MTECH",
  "age": 37,
  "dob": "1983-09-12",
  "experience": 14
},
{
  "id": 9,
  "teacherName": "Mallin Garix",
  "teacherAddress": "LA",
  "qualification": "PHD",
  "age": 34,
  "dob": "1991-09-09",
  "experience": 7
}
```

10) After (""/updateTeacher/{id}"")

Postman interface showing a PUT request to `http://localhost:8080/updateTeacher/9`. The request body is a JSON object:

```
{  "teacherAddress": "France",  "qualification": "MTECH",  "age": 44,  "dob": "1987-09-09",  "experience": 18}
```

The response status is 200 OK. The response body is a JSON object:

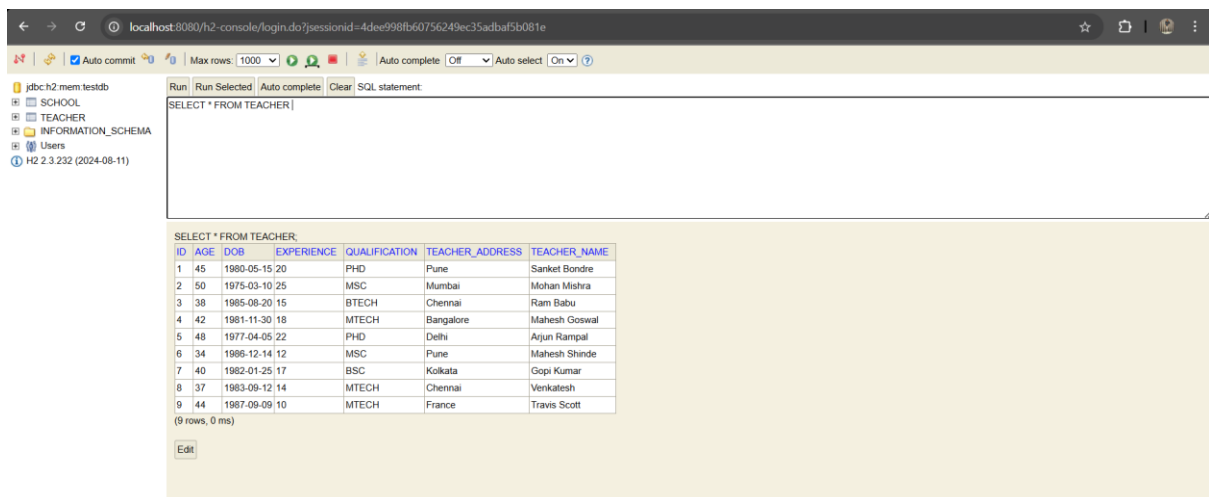
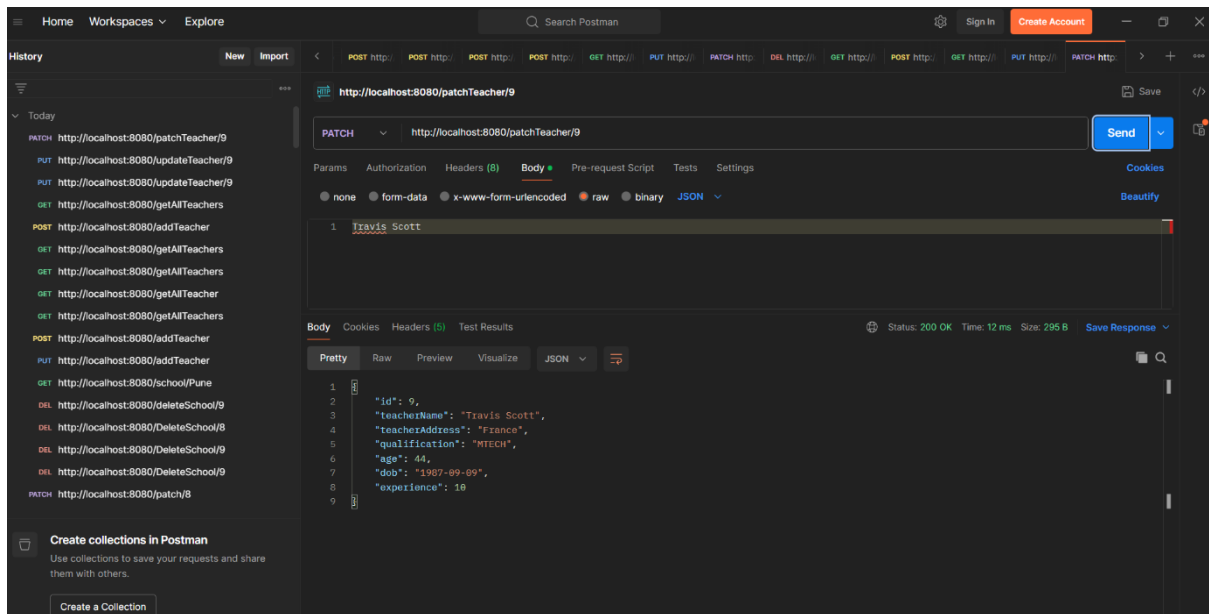
```
{  "id": 9,  "teacherName": "Travis Head",  "teacherAddress": "France",  "qualification": "MTECH",  "age": 44,  "dob": "1987-09-09",  "experience": 18}
```

Web browser interface showing a SQL query result. The query is `SELECT * FROM TEACHER;`. The result is a table with 9 rows:

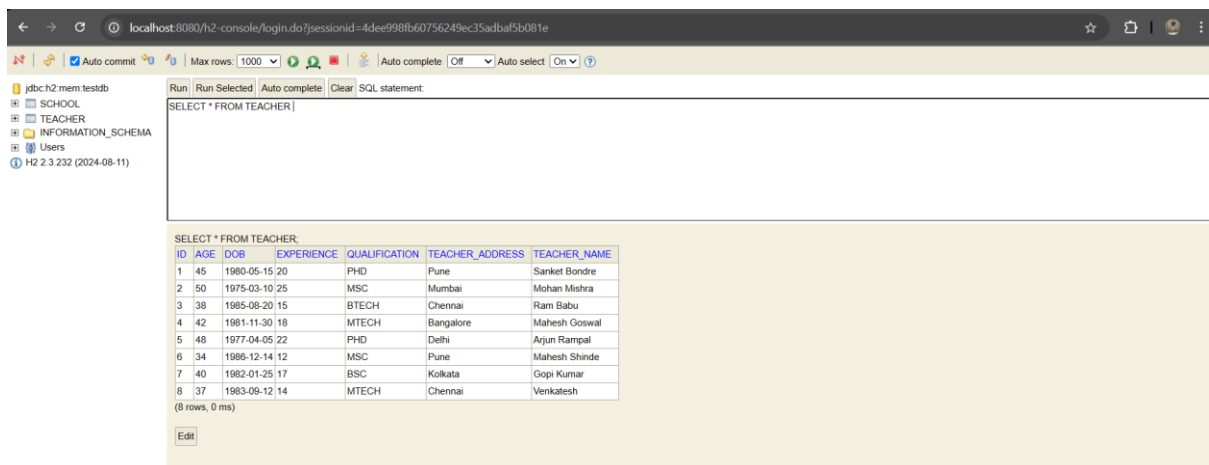
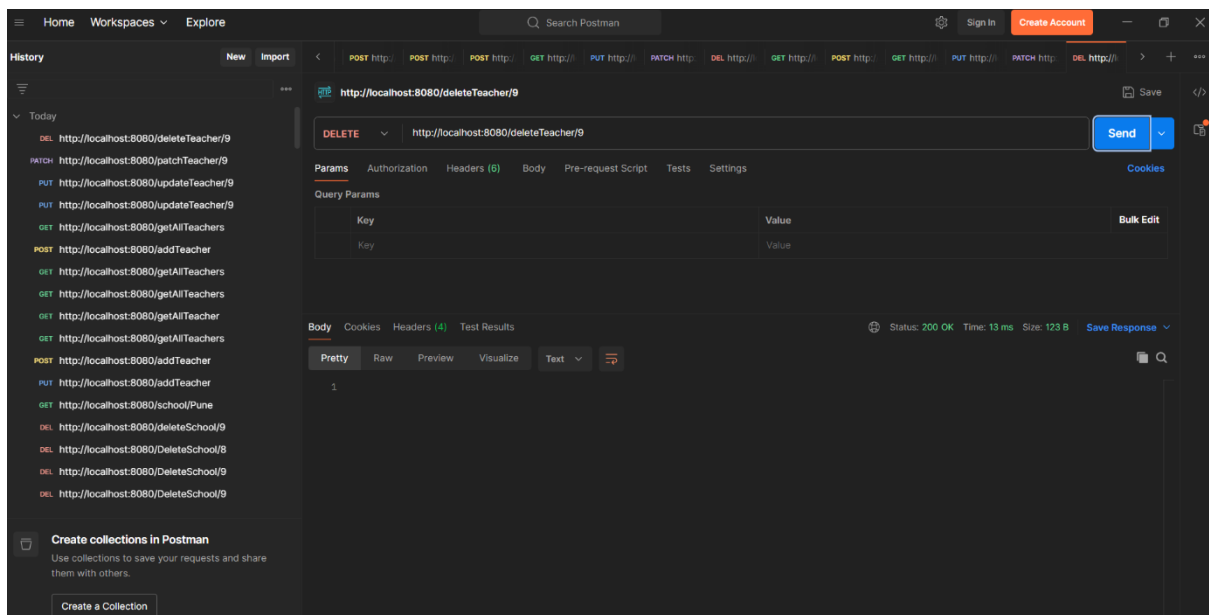
ID	AGE	DOB	EXPERIENCE	QUALIFICATION	TEACHER_ADDRESS	TEACHER_NAME
1	45	1980-05-15	20	PHD	Pune	Sanket Bondre
2	50	1975-03-10	25	MSC	Mumbai	Mohan Mishra
3	38	1985-08-20	15	BTECH	Chennai	Ram Babu
4	42	1981-11-30	18	MTECH	Bangalore	Maresh Goswal
5	48	1977-04-05	22	PHD	Delhi	Arun Rampal
6	34	1988-12-14	12	MSC	Pune	Maresh Shinde
7	40	1982-01-25	17	BSC	Kolkata	Gopi Kumar
8	37	1983-09-12	14	MTECH	Chennai	Venkatesh
9	44	1987-09-09	10	MTECH	France	Travis Head

(9 rows, 0 ms)

11) After (“/patchTeacher/{id}”)



12) After(“/deleteTeacher/{id}”)



13) After (“/teachers/{qualification}”)

