Week-11

Roll Number: 238W1A1296

1. Using h2 database

Project name: h2db

Author.java

```
package com.example.mysqldemo;
public class Author {
private String firstName;
private String lastName;
public String getFirstName() {return firstName;}
public void setFirstName(String firstName) {this.firstName = firstName;}
public String getlastName() {return lastName;}
public void setLastName(String lastName) {this.lastName = lastName;}
public Author(String firstName, String lastName) {
super();
this.firstName = firstName;
this.lastName = lastName;}}
```

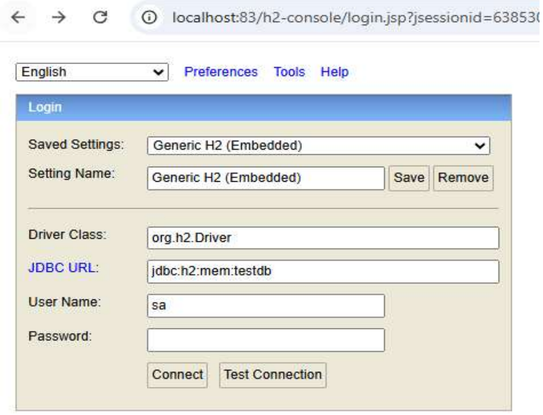
AuthorService

```
package com.example.mysqldemo;
import java.util.ArrayList;
import java.util.*;
import org.slf4j.*;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Service;
import jakarta.annotation.PostConstruct;
@Service
public class AuthorService {
private static final Logger log =LoggerFactory.getLogger(AuthorService.class);
@Autowired
JdbcTemplate jdbcTemplate;
@PostConstruct
public void postConstruct() {
        Author author1=new Author("Manogna","IT");
        Author author2=new Author("Varshini","CSE");
        List <Author>authors = new ArrayList<>();
```

```
          authors.add(author1);
          authors.add(author2);
log.info("Creating tables");
          jdbcTemplate.execute("DROP TABLE IF EXISTS author");
          jdbcTemplate.execute("CREATE TABLE author(" + " first_name varchar(255),
last_name varchar(255))");
authors.forEach(i->jdbcTemplate.update("INSERT INTO author(first_name, last_name)
VALUES (?,?)",i.getFirstName(),i.getLastName()));
log.info("Records Saved");
log.info("Retrieving records");
authors= jdbcTemplate.query("select * from author", (rs, rowNum)-> new
Author(rs.getString("first_name"),rs.getString("last_name")));
autgors.forEach(i -> log.info(i.getFirstName() + " " +i.getLastName()));}}
```

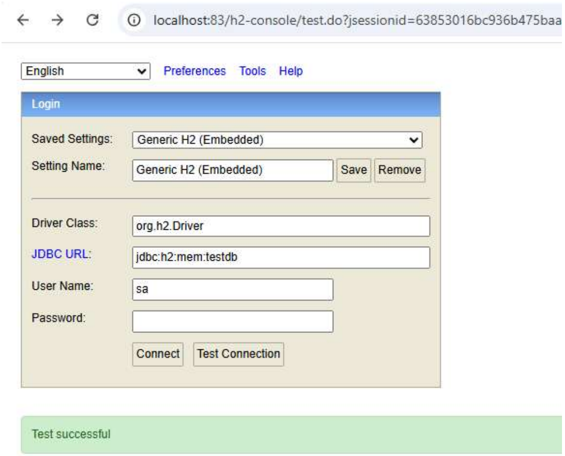
Application.properties:

```
server.port=83
spring.application.name=h2db
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.hibernate.ddl-auto=none
```



localhost:83/h2-console/login.jsp?jsessionid=63853

Saved Settings: Generic H2 (Embedded)
Setting Name: Generic H2 (Embedded)
Driver Class: org.h2.Driver
JDBC URL: jdbc:h2:mem:testdb
User Name: sa
Password:


localhost:83/h2-console/test.do?jsessionid=63853016bc936b475baa

English

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded)

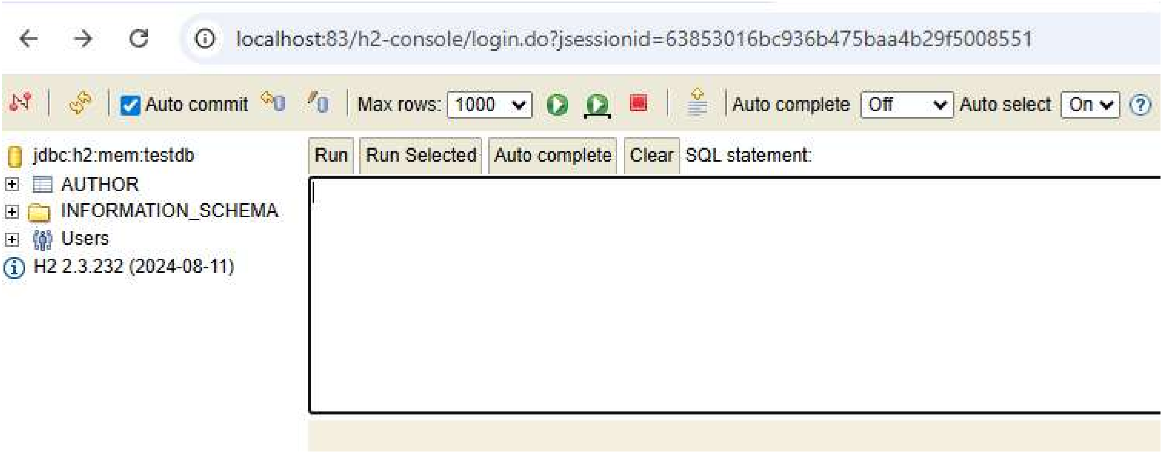
Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

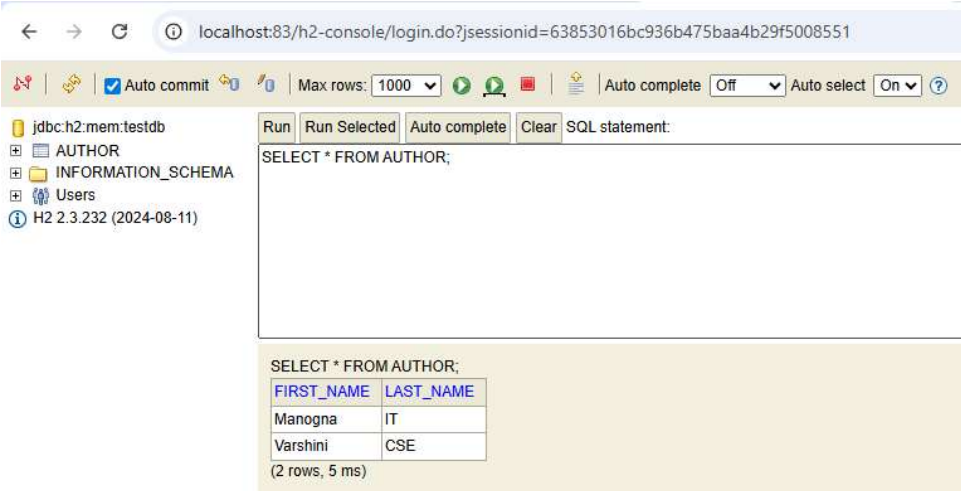
User Name: sa

Password:

Test successful


localhost:83/h2-console/login.do?jsessionid=63853016bc936b475baa4b29f5008551

☑ Auto commit   Max rows: 1000   Auto complete Off   Auto select On


localhost:83/h2-console/login.do?jsessionid=63853016bc936b475baa4b29f5008551

☑ Auto commit   Max rows: 1000   Auto complete Off   Auto select On

```
SELECT * FROM AUTHOR;
```

SELECT * FROM AUTHOR;

FIRST_NAME	LAST_NAME
Manogna	IT
Varshini	CSE

(2 rows, 5 ms)

JDBC with MySQL:

Employee.java

```
package com.example.demo;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
@Entity
public class Employee {
    @Id
    private int id;
    private String name;
    private String department;
    public Employee() {}
    public Employee(int id, String name, String department) {
        this.id = id;
        this.name = name;
        this.department = department;
    }
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getDepartment() { return department; }
    public void setDepartment(String department) { this.department = department; }}
```

EmployeeRepository.java

```
package com.example.demo;
import org.springframework.data.jpa.repository.JpaRepository;
public interface EmployeeRepository extends JpaRepository<Employee, Integer> { }
```

MysqlJdbcDemoApplication.java

```java
package com.example.demo;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class MysqlJdbcDemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(MysqlJdbcDemoApplication.class, args);
        System.out.println("welcome");
    }
}
```

Applications.properties

```
spring.application.name=MysqlJdbcDemo
server.port=85
spring.datasource.url=jdbc:mysql://localhost:3306/employee_db?useSSL=false&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.hibernate.ddl-auto=none
spring.jpa.show-sql=true
spring.sql.init.mode=always
spring.jpa.defer-datasource-initialization=true
```

DATABASE

```sql
Create database employee_db;
Use employee_db;
CREATE TABLE IF NOT EXISTS employee (
    id INT PRIMARY KEY,
```

    name VARCHAR(255),

    department VARCHAR(255)

);

INSERT INTO employee (id, name, department) VALUES

(1, 'Manogna', 'IT'),

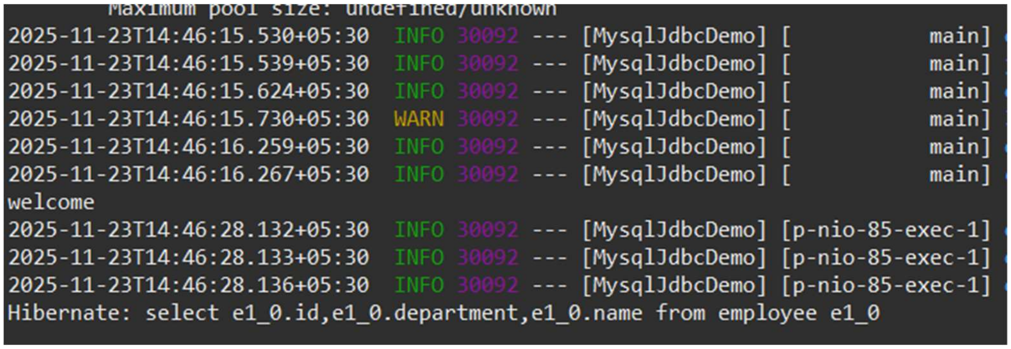
(2, 'Keerthi’, 'CSE'),

(3, 'Sai', 'ECE');

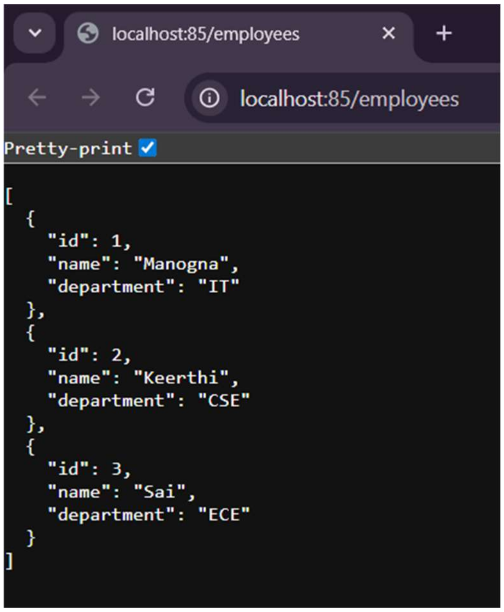
Open:

http://localhost:85/employees

Output:



```
        Maximum pool size: undefined/unknown
2025-11-23T14:46:15.530+05:30  INFO 30092 --- [MysqlJdbcDemo] [           main]
2025-11-23T14:46:15.539+05:30  INFO 30092 --- [MysqlJdbcDemo] [           main]
2025-11-23T14:46:15.624+05:30  INFO 30092 --- [MysqlJdbcDemo] [           main]
2025-11-23T14:46:15.730+05:30  WARN 30092 --- [MysqlJdbcDemo] [           main]
2025-11-23T14:46:16.259+05:30  INFO 30092 --- [MysqlJdbcDemo] [           main]
2025-11-23T14:46:16.267+05:30  INFO 30092 --- [MysqlJdbcDemo] [           main]
welcome
2025-11-23T14:46:28.132+05:30  INFO 30092 --- [MysqlJdbcDemo] [p-nio-85-exec-1]
2025-11-23T14:46:28.133+05:30  INFO 30092 --- [MysqlJdbcDemo] [p-nio-85-exec-1]
2025-11-23T14:46:28.136+05:30  INFO 30092 --- [MysqlJdbcDemo] [p-nio-85-exec-1]
Hibernate: select e1_0.id,e1_0.department,e1_0.name from employee e1_0
```



```
[
  {
    "id": 1,
    "name": "Manogna",
    "department": "IT"
  },
  {
    "id": 2,
    "name": "Keerthi",
    "department": "CSE"
  },
  {
    "id": 3,
    "name": "Sai",
    "department": "ECE"
  }
]
```