Project

COM339 ADVANCE PROGRAMMING

Prepared by

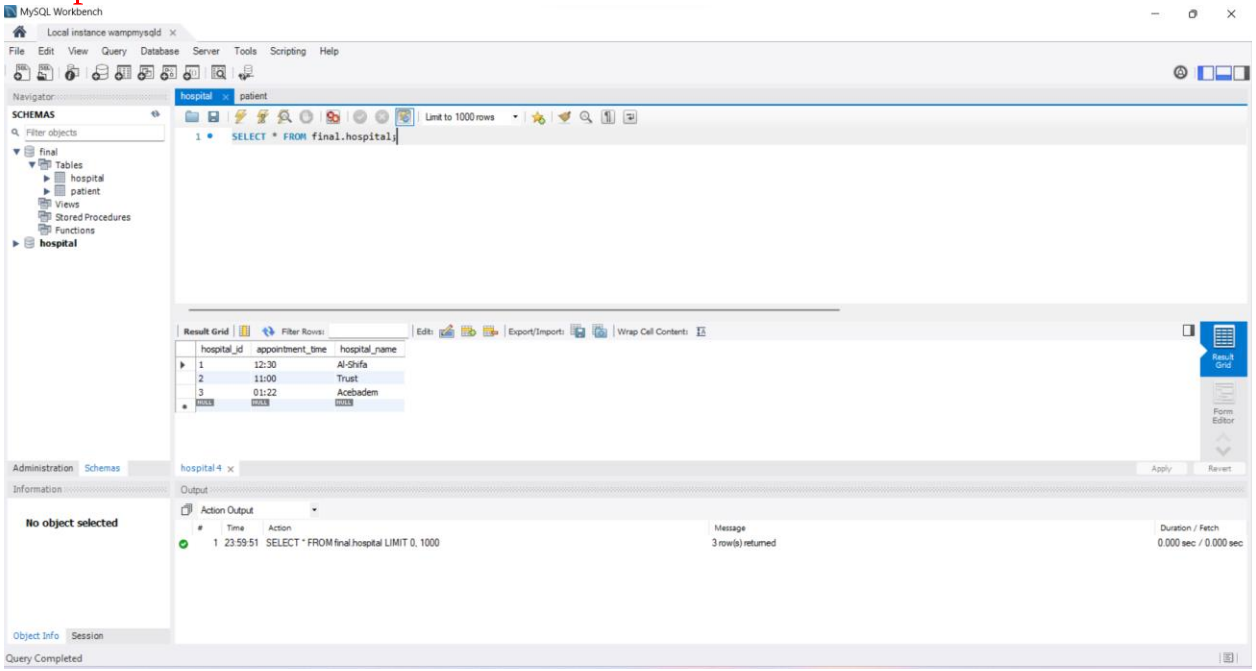
Reema Salem – B1705.010067

Supervisor

Dr. Öğr. Üyesi Selcuk Sener

This project involves a single database called "final" that contains two tables: "Hospital" and "Patient." The primary objective of the project is to retrieve the hospital name, appointment details, and corresponding ID based on patient information. The "Hospital" table has three columns, and the "Patient" table has four columns, with a foreign key relationship established using the "HospitalId" column. Furthermore, the project encompasses two entity classes (Hospital and Patient), two repository interfaces (HospitalRepository and PatientRepository), and two controllers (HospitalController and PatientController).

Hospital Table:

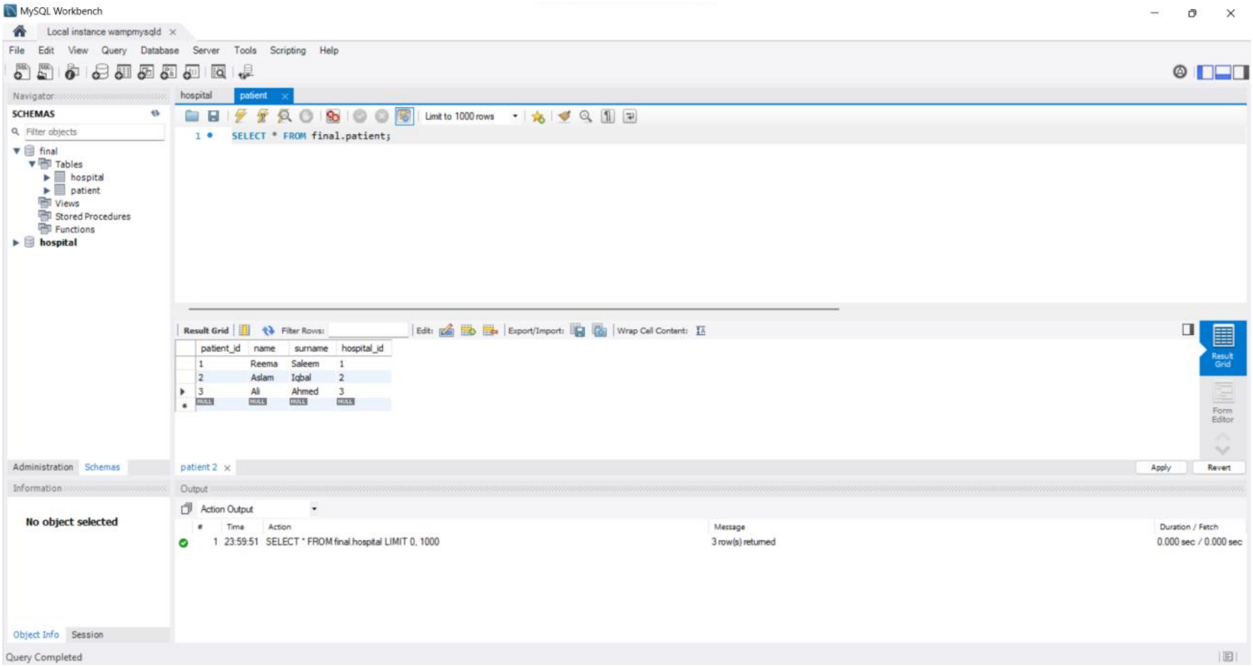


```
1 • SELECT * FROM final.hospital;
```

hospital_id	appointment_time	hospital_name
1	12:30	Al-Shifa
2	11:00	Trust
3	01:22	Acebadem
NULL	NULL	NULL

```
#  Time      Action                                   Message            Duration / Fetch
1  23:59:51  SELECT * FROM final.hospital LIMIT 0, 1000   3 row(s) returned  0.000 sec / 0.000 sec
```

Patient Table:

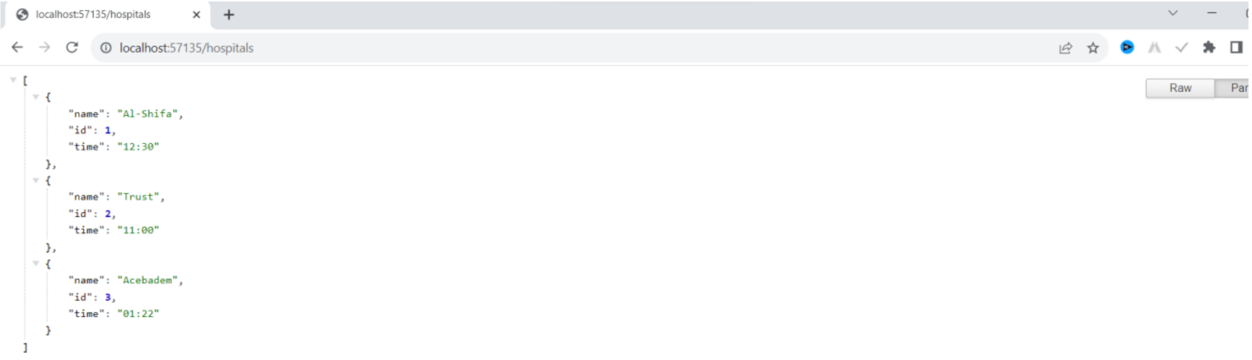


```
SELECT * FROM final.patient;
```

patient_id	name	surname	hospital_id
1	Reema	Saleem	1
2	Aslam	Iqbal	2
3	Ali	Ahmed	3
NULL	NULL	NULL	NULL

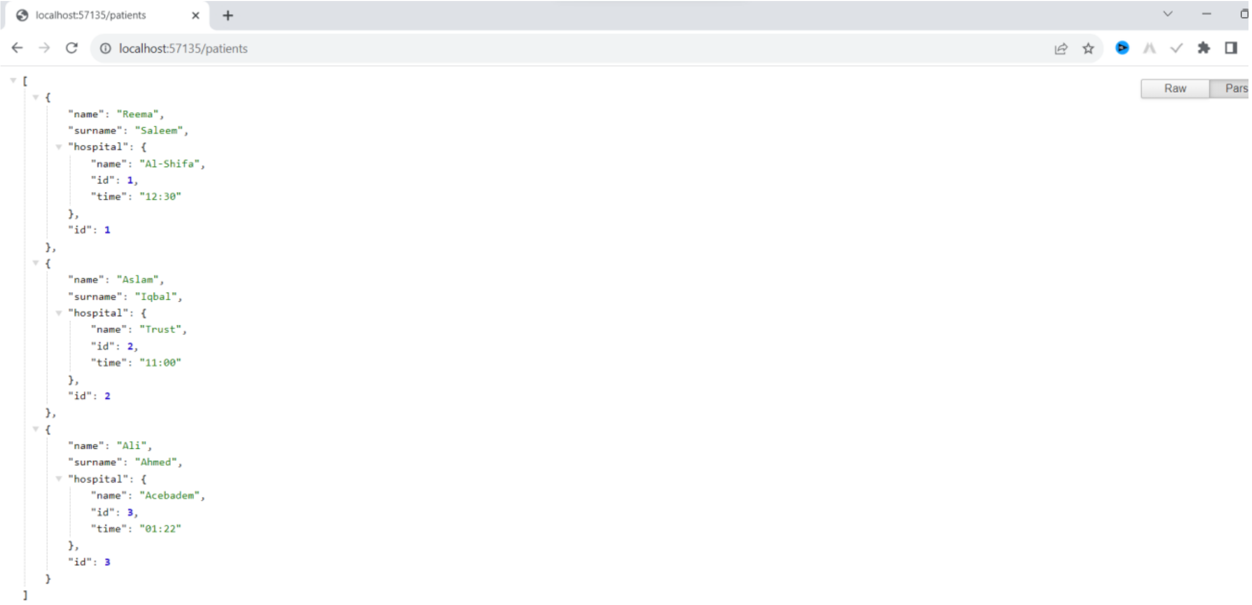
#	Time	Action	Message	Duration / Fetch
1	23:59:51	SELECT * FROM final.hospital LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Fetching all Hospitals data from mysql:



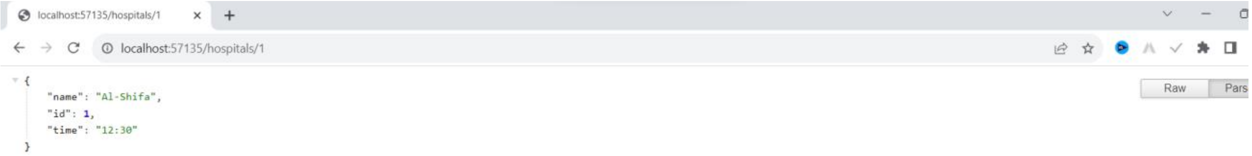
```
[
  {
    "name": "Al-Shifa",
    "id": 1,
    "time": "12:30"
  },
  {
    "name": "Trust",
    "id": 2,
    "time": "11:00"
  },
  {
    "name": "Acebadem",
    "id": 3,
    "time": "01:22"
  }
]
```

Fetching all patients data from mysql:



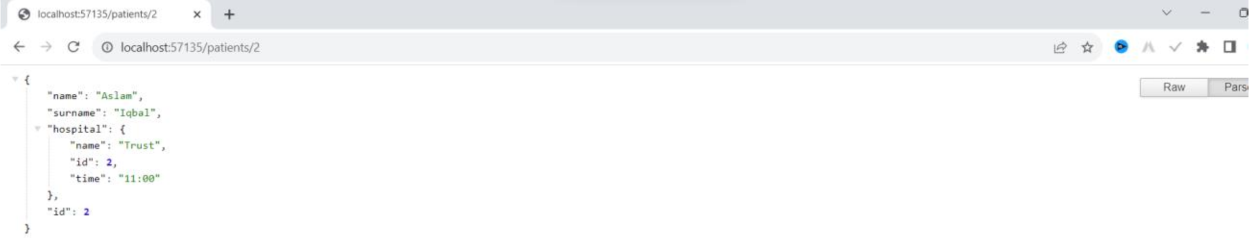
```
[
    {
        "name": "Reema",
        "surname": "Saleem",
        "hospital": {
            "name": "Al-Shifa",
            "id": 1,
            "time": "12:30"
        },
        "id": 1
    },
    {
        "name": "Aslam",
        "surname": "Iqbal",
        "hospital": {
            "name": "Trust",
            "id": 2,
            "time": "11:00"
        },
        "id": 2
    },
    {
        "name": "Ali",
        "surname": "Ahmed",
        "hospital": {
            "name": "Acebadem",
            "id": 3,
            "time": "01:22"
        },
        "id": 3
    }
]
```

Fetching hospital data according to ID:



```
{
    "name": "Al-Shifa",
    "id": 1,
    "time": "12:30"
}
```

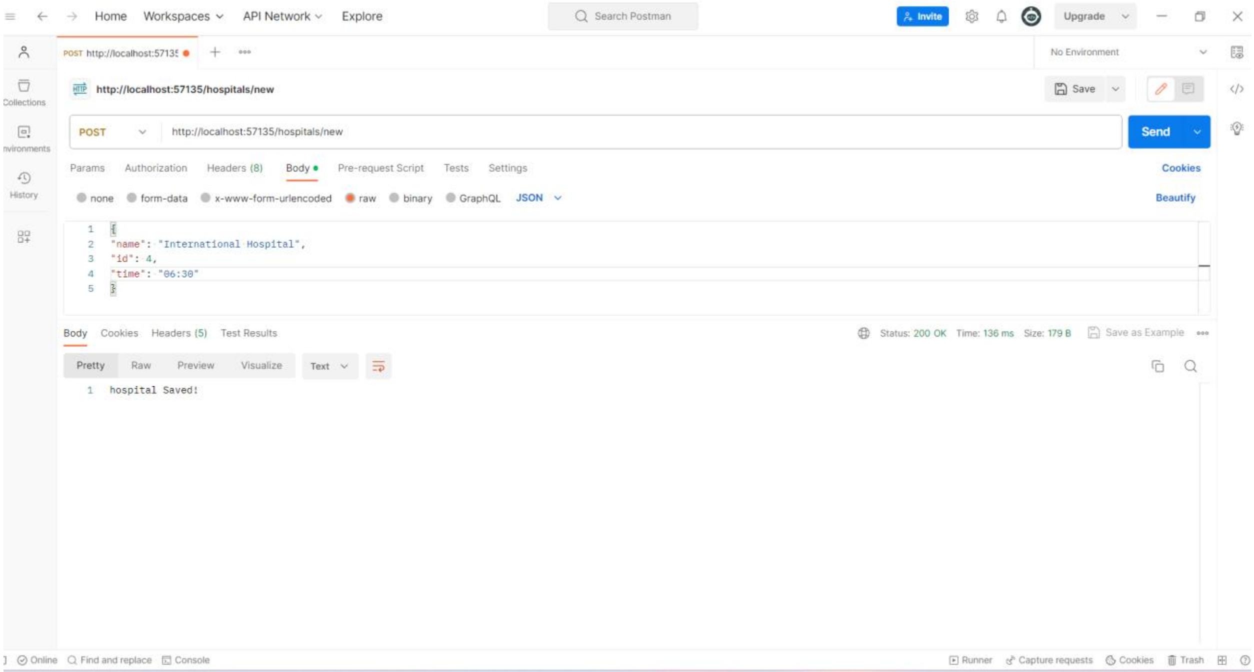
Fetching patient data according to ID:



localhost:57135/patients/2

```
{
    "name": "Aslam",
    "surname": "Iqbal",
    "hospital": {
        "name": "Trust",
        "id": 2,
        "time": "11:00"
    },
    "id": 2
}
```

Saving Hospital into Hospital Table:



POST http://localhost:57135/hospitals/new

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
"name": "International Hospital",
"id": 4,
"time": "06:30"
}
```

Status: 200 OK Time: 136 ms Size: 179 B

```
hospital Saved!
```

Hospital Entity Class:

```
package com.example.demo3.Entity;


import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;


@Entity
@Table(name="hospital")
public class Hospital {



   @Id
   @GeneratedValue(strategy=GenerationType.IDENTITY)
        Long HospitalId;


   @Column(name="hospital_name")
        String name;


   String appointmentTime;




        public Long getId() {
                return HospitalId;
        }
```

```
    public void setId(Long id) {
        this.HospitalId = id;
    }


    public String getName() {
        return name;
    }


    public void setName(String name) {
        this.name = name;
    }


    public String getTime() {
        return appointmentTime;
    }


    public void setTime(String time) {
        this.appointmentTime = time;
    }


}
```

Patient Entity Class:

```
package com.example.demo3.Entity;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;

@Entity
@Table(name = "patient")
public class Patient {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    Long patientId;

    String name;
    String surname;

    @ManyToOne
    @JoinColumn(name = "HospitalId")
    Hospital hospital;

    public Long getId() {
        return patientId;
    }
}

public void setId(Long id) {
        this.patientId = id;
}

public String getName() {
        return name;
}

public void setName(String name) {
        this.name = name;
}

public String getSurname() {
        return surname;
}

public void setSurname(String surname) {
        this.surname = surname;
}

public Hospital getHospital() {
        return hospital;
}

public void setHospital(Hospital hos) {
        this.hospital = hos;
}
```

```
}
```

Hospital Repository Interface:

```
package com.example.demo3.Repository;

import org.springframework.data.jpa.repository.JpaRepository;

import com.example.demo3.Entity.Hospital;

public interface HospitalRepository extends JpaRepository<Hospital , Long>{

}
```

Patient Repository Interface:

```
package com.example.demo3.Repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import com.example.demo3.Entity.Patient;

public interface PatientRepository extends JpaRepository<Patient , Long> {

        List<Patient> findByPatientId(Long PatientId);

}
```

Hospital Controller Class:

```
package com.example.demo3.Controller;


import java.util.List;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.example.demo3.Entity.Hospital;
import com.example.demo3.Repository.HospitalRepository;


@RestController
@RequestMapping("/hospitals")
public class HospitalController {


    @Autowired
    HospitalRepository hospitalRep;


    @GetMapping
    List<Hospital> getAllHospitals()  {
    return   hospitalRep.findAll();
    }


    @GetMapping("/{id}")
    Optional<Hospital> getHospital(@PathVariable Long id) {
        return hospitalRep.findById(id);
    }

    @PostMapping("new")
    String newHospital(@RequestBody Hospital hos) {
      hospitalRep.save(hos);
        return "hospital Saved!";
    }

}
```

Patient Controller Class:

```
package com.example.demo3.Controller;

import java.util.List;
import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.example.demo3.Entity.Patient;
import com.example.demo3.Entity.Hospital;
import com.example.demo3.Repository.PatientRepository;
import com.example.demo3.Repository.HospitalRepository;
```

```
@RestController
@RequestMapping("/patients")
public class PatientController {

    @Autowired
    PatientRepository patRepo;
    @Autowired
    HospitalRepository hosRepo;

    @GetMapping
    List<Patient> getPatients() {
        return patRepo.findAll();
    }

    @GetMapping("/{id}")
    Optional<Patient> getPatients(@PathVariable Long id) {
        return patRepo.findById(id);
    }

    String newPatient(@PathVariable long id , @RequestBody Patient pa) {
        for(Hospital t : hosRepo.findAll()) {
            if(t.getId()==id) {
                pa.setHospital(t);
                break;
            }
        }
        patRepo.save(pa);
```

```
                return "Patient saved!";
        }

}
```

Application Properties:

```
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost/final
spring.datasource.username=root
spring.datasource.password=
server.port=0
```

Demo Application:

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DemoApplication {

        public static void main(String[] args) {
                SpringApplication.run(DemoApplication.class, args);
        }

}
```

THANK YOU!