**Week:08**

**Project back-end code**

Go to the command prompt and give command

**mysqld**- to login

Enter the user name and password as:

**mysql –u root –p**

**Enter Password: \*\*\*\*\*\*\*\*\*\*\***

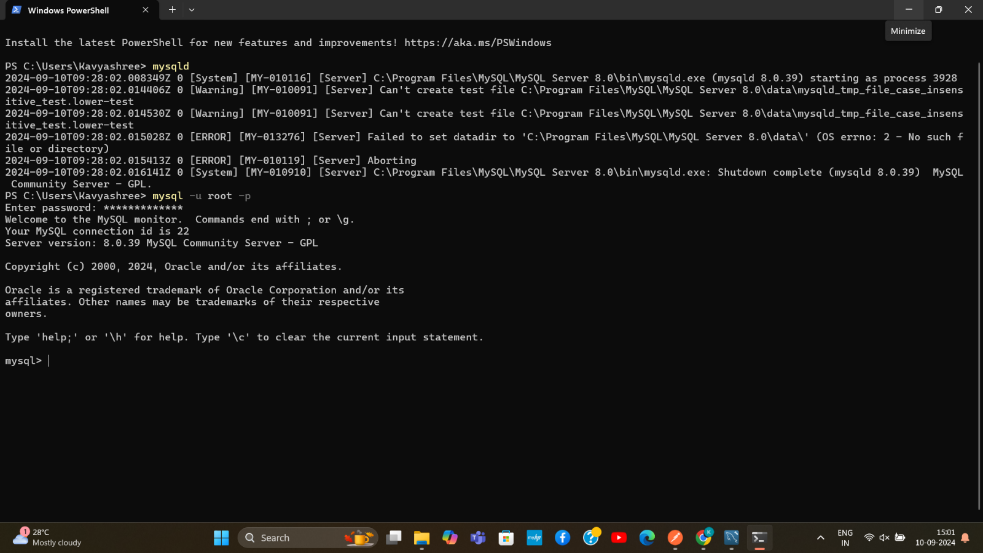
Create a database:

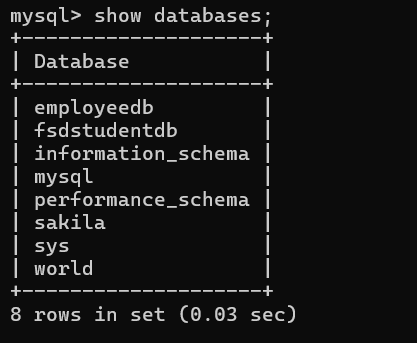
**create database employeedb**

**use employeedb**

to check in which port mysql is running:

**show variables like “port”**



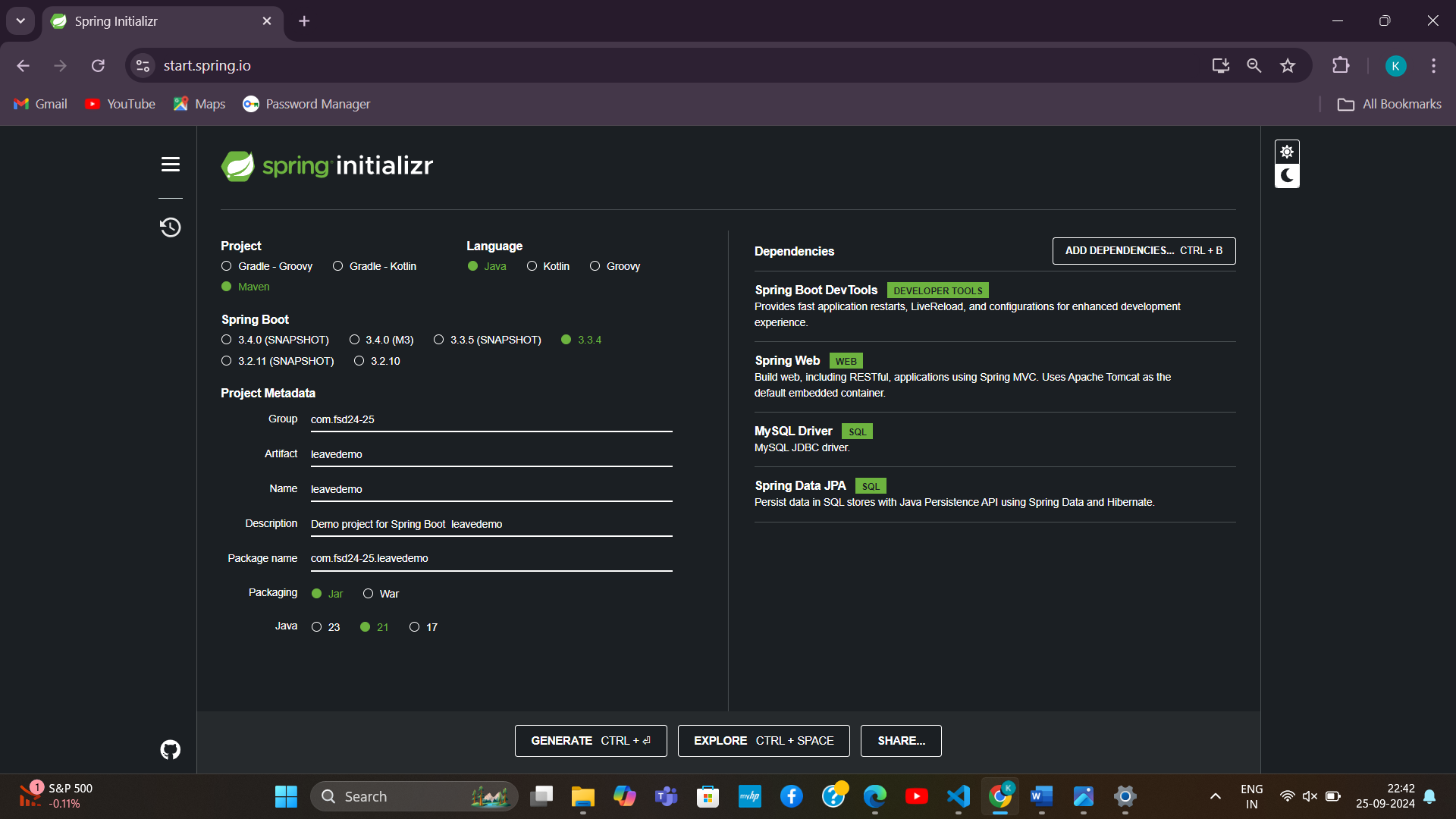




To set project configuration🡪Go to browser🡪Search **start.spring.io**

In spring initializr🡪Select Project-Maven🡪Language-Java🡪Spring Boot-3.3.3🡪 Group- com.fsd24\_25🡪Artifact-leavedemo🡪Description-Demo project for spring boot leavedemo 🡪Packaging-Jar🡪Java-21.

Dependencies🡪ADD DEPENDENCIES🡪Select-Spring Boot Dev Tools ,Spring Web,MySQL Driver and Spring Data JPA🡪 Click on Generate.

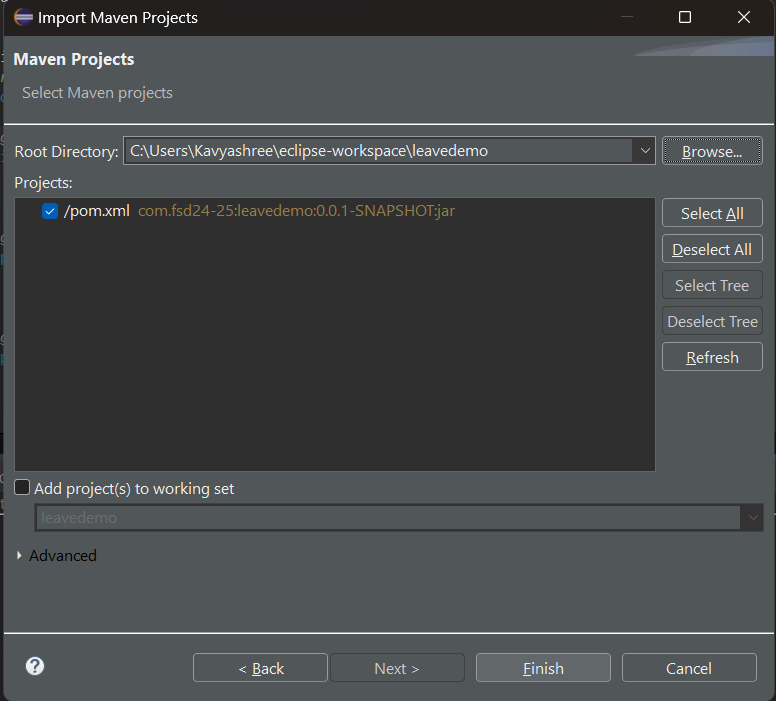


Next extract your zip file to workspace.

Select a destination.

Go to Eclipse🡪File🡪Import🡪Maven🡪Existing maven🡪Click on Browse and select the folder from where you extracted🡪Tick mark on pom.xml🡪Click on Finish.

Save the pom.xml



Create four packages:

com.fsd24\_25. leavedemo.Entity

com.fsd24\_25. leavedemo.Controller

com.fsd24\_25. leavedemo.Repository

com.fsd24\_25. leavedemo.Service

Create a Student class in com.fsd24\_25. leavedemo.Entity

**Employee.java:**

package com.fsd24\_25.leavedemo.Entity;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name = "employee")

public class employee {

@Id

private String empId;

private String name;

@Column(name = "email", nullable = false)

private String email;

@Column(name = "password", nullable = false)

private String password;

private String contactNo;

private String status;

public employee() {}

public employee(String empId, String name, String email, String password, String contactNo, String status) {

this.empId = empId;

this.name = name;

this.email = email;

this.password = password;

this.contactNo = contactNo;

this.status = status;

}

public String getEmpId() {

return empId;

}

public void setEmpId(String empId) {

this.empId = empId;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getContactNo() {

return contactNo;

}

public void setContactNo(String contactNo) {

this.contactNo = contactNo;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

@Override

public String toString() {

return "{ \"empId\":\"" + empId + "\", \"name\":\"" + name + "\", \"email\":\"" + email + "\", " +

"\"password\":\"" + password + "\", \"contactNo\":\"" + contactNo + "\", \"status\":\"" + status + "\" }";

}

}

**Week:09**

Create class on Repository .

**RepoEmp.java:**

package com.fsd24\_25.leavedemo.Repository;

import com.fsd24\_25.leavedemo.Entity.employee;

import java.util.Optional;

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

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

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

import org.springframework.data.repository.query.Param;

import org.springframework.stereotype.Repository;

import jakarta.transaction.Transactional;

@Repository

public interface RepoEmp extends JpaRepository<employee, String> {

Optional<employee> findByEmail(String email);

@Modifying

@Transactional

@Query("UPDATE employee SET status = :status WHERE empId = :empId")

int updateemployeeStatus(@Param("empId") String empId, @Param("status") String status);

}

**leavdetails.java**

package com.fsd24\_25.leavedemo.Entity;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

import java.time.LocalDate;

@Entity

@Table(name = "leavedetails")

public class leavedetails {

@Id

private String empId;

private LocalDate startDate;

private LocalDate endDate;

private String type;

private String reason;

private String status; // Added status field

public leavedetails() {}

public leavedetails(String empId, LocalDate startDate, LocalDate endDate, String type, String reason, String status) {

this.empId = empId;

this.startDate = startDate;

this.endDate = endDate;

this.type = type;

this.reason = reason;

this.status = status; // Initialize status

}

public String getEmpId() {

return empId;

}

public void setEmpId(String empId) {

this.empId = empId;

}

public LocalDate getStartDate() {

return startDate;

}

public void setStartDate(LocalDate startDate) {

this.startDate = startDate;

}

public LocalDate getEndDate() {

return endDate;

}

public void setEndDate(LocalDate endDate) {

this.endDate = endDate;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public String getReason() {

return reason;

}

public void setReason(String reason) {

this.reason = reason;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

@Override

public String toString() {

return "{ \"empId\":\"" + empId + "\", \"startDate\":\"" + startDate + "\", \"endDate\":\"" + endDate + "\", " +

"\"type\":\"" + type + "\", \"reason\":\"" + reason + "\", \"status\":\"" + status + "\" }";

}

}

**Repoleavedetails.java**

package com.fsd24\_25.leavedemo.Repository;

import com.fsd24\_25.leavedemo.Entity.leavedetails;

import java.time.LocalDate;

import java.util.Optional;

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

import org.springframework.stereotype.Repository;

@Repository

public interface Repoleavedetails extends JpaRepository<leavedetails, String> {

Optional<leavedetails> findByEmpId(String empId);

void deleteByEmpId(String empId);

Optional<leavedetails> findByEmpIdAndStartDate(String empId, LocalDate startDate);}

**admin.java**

package com.fsd24\_25.leavedemo.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 = "admin")

public class admin {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

@Column(name = "email", nullable = false)

private String email;

@Column(name = "password", nullable = false)

private String password;

private String contactNo;

public admin() {}

public admin(Long id, String name, String email, String password, String contactNo) {

super();

this.id = id;

this.name = name;

this.email = email;

this.password = password;

this.contactNo = contactNo;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getContactNo() {

return contactNo;

}

public void setContactNo(String contactNo) {

this.contactNo = contactNo;

}

@Override

public String toString() {

return "{ \"Id\":\"" + id + "\", \"name\":\"" + name + "\", \"email\":\"" + email + "\", " +

"\"password\":\"" + password + "\", \"contactNo\":\"" + contactNo + "\" }";

}

}

**Adminrepo.java**

package com.fsd24\_25.leavedemo.Repository;

import java.util.Optional;

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

import com.fsd24\_25.leavedemo.Entity.admin;

public interface Adminrepo extends JpaRepository<admin, Long> {

Optional<admin> findByEmail(String email);

}

**Week:10**

Create servEmp class inside com.fsd24\_25. leavedemo.Service

Type the code below

**servEmp.java:**

package com.fsd24\_25.leavedemo.Service;

import com.fsd24\_25.leavedemo.Entity.admin;

import com.fsd24\_25.leavedemo.Entity.employee;

import com.fsd24\_25.leavedemo.Entity.leavedetails;

import com.fsd24\_25.leavedemo.Repository.Adminrepo;

import com.fsd24\_25.leavedemo.Repository.NotificationRepository;

import com.fsd24\_25.leavedemo.Repository.RepoEmp;

import com.fsd24\_25.leavedemo.Repository.Repoleavedetails;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.time.LocalDate;

import java.util.List;

import java.util.Optional;

@Service

public class servEmp {

@Autowired

private RepoEmp repoEmp;

@Autowired

private Repoleavedetails repoleavedetails;

@Autowired

private Adminrepo adminrepo;

public Optional<employee> getEmployeeById(String empId) {

return repoEmp.findById(empId);

}

public List<employee> getAllEmployees() {

return repoEmp.findAll();

}

public employee addEmployee(employee emp) {

return repoEmp.save(emp);

}

public void updateEmployeeStatus(String empId, String status) {

repoEmp.updateemployeeStatus(empId, status);

}

public boolean loginEmployee(String empId, String password) {

return repoEmp.findById(empId)

.map(emp -> emp.getPassword().equals(password)) // Simplified with map

.orElse(false); // Return false if employee not found or password mismatch

}

public void deleteEmployee(String empId) throws EmployeeNotFoundException {

Optional<employee> emp = repoEmp.findById(empId);

if (emp.isPresent()) {

repoEmp.deleteById(empId); // Deletes the employee from the database

} else {

throw new EmployeeNotFoundException("Employee with ID " + empId + " not found");

}

}

public List<leavedetails> getAllLeaveDetails() {

return repoleavedetails.findAll();

}

public Optional<leavedetails> getLeaveDetailsByEmpId(String empId) {

return repoleavedetails.findById(empId); // Consider whether this is the correct method

}

public leavedetails addLeaveDetails(leavedetails leaveDetail) {

return repoleavedetails.save(leaveDetail);

}

public void deleteLeaveDetails(String empId) {

repoleavedetails.deleteById(empId);

}

public Optional<admin> loginAdmin(String email, String password) {

return adminrepo.findByEmail(email)

.filter(admin -> admin.getPassword().equals(password)); // Using filter for cleaner code

}

public void approveLeave(String empId, LocalDate startDate) {

leavedetails leave = repoleavedetails.findByEmpIdAndStartDate(empId, startDate)

.orElseThrow(() -> new RuntimeException("Leave not found")); // Provide a meaningful exception

leave.setStatus("Approved");

repoleavedetails.save(leave);

Notification notification = new Notification(empId, "Your leave has been approved.", "Unread");

notificationRepo.save(notification);

}

public void rejectLeave(String empId, LocalDate startDate) {

leavedetails leave = repoleavedetails.findByEmpIdAndStartDate(empId, startDate)

.orElseThrow(() -> new RuntimeException("Leave not found")); // Provide a meaningful exception

leave.setStatus("Rejected");

repoleavedetails.save(leave);

}

**Week:11**

* Create ctrlEmpl class inside com.fsd24\_25. leavedemo.Controller
* Type the code below.

**contrlEmpl.java:**

package com.fsd24\_25.leavedemo.Controller;

import com.fsd24\_25.leavedemo.Entity.admin;

import com.fsd24\_25.leavedemo.Entity.employee;

import com.fsd24\_25.leavedemo.Entity.leavedetails;

import com.fsd24\_25.leavedemo.Service.servEmp;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.time.LocalDate;

import java.util.List;

import java.util.Optional;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@RestController

public class cntrlEmpl {

private static final Logger logger = LoggerFactory.getLogger(cntrlEmpl.class);

@Autowired

private servEmp service;

@GetMapping("/employee")

public ResponseEntity<List<employee>> getEmployeeList() {

logger.info("Fetching all employees");

List<employee> employees = service.getAllEmployees();

return ResponseEntity.ok(employees);

}

@PostMapping("/emp")

public ResponseEntity<employee> insertEmployee(@RequestBody employee emp) {

logger.info("Inserting new employee: {}", emp.getEmpId());

try {

employee savedEmployee = service.addEmployee(emp);

return ResponseEntity.status(HttpStatus.CREATED).body(savedEmployee);

} catch (Exception e) {

logger.error("Error inserting employee: {}", e.getMessage());

return ResponseEntity.status(HttpStatus.INTERNAL\_SERVER\_ERROR).body(null);

}

}

@PostMapping("/login")

public ResponseEntity<?> loginEmployee(@RequestBody employee emp) {

logger.info("Attempting to log in employee with ID: {}", emp.getEmpId());

boolean isLoggedIn = service.loginEmployee(emp.getEmpId(), emp.getPassword());

if (isLoggedIn) {

return ResponseEntity.ok("Login successful");

} else {

logger.warn("Invalid login attempt for employee ID: {}", emp.getEmpId());

return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Invalid credentials for employee ID: " + emp.getEmpId());

}

}

@PutMapping("/{empId}/status")

public ResponseEntity<?> updateEmployeeStatus(@PathVariable String empId, @RequestParam String status) {

logger.info("Updating status for employee ID: {} to {}", empId, status);

try {

service.updateEmployeeStatus(empId, status);

return ResponseEntity.ok("Employee status updated successfully to " + status);

} catch (Exception e) {

logger.error("Error updating employee status: {}", e.getMessage());

return ResponseEntity.status(HttpStatus.INTERNAL\_SERVER\_ERROR).body("Error updating status");

}

}

@PostMapping("/login/admin")

public ResponseEntity<?> loginAdmin(@RequestBody admin admin) {

logger.info("Attempting to log in admin with email: {}", admin.getEmail());

Optional<admin> adminLogin = service.loginAdmin(admin.getEmail(), admin.getPassword());

if (adminLogin.isPresent()) {

return ResponseEntity.ok("Admin login successful");

} else {

logger.warn("Invalid login attempt for admin email: {}", admin.getEmail());

return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Invalid admin credentials");

}

}

@GetMapping("/employee/profile/{empId}")

public ResponseEntity<employee> getEmployeeProfile(@PathVariable String empId) {

logger.info("Fetching profile for employee ID: {}", empId);

Optional<employee> employeeProfile = service.getEmployeeById(empId);

if (employeeProfile.isPresent()) {

return ResponseEntity.ok(employeeProfile.get());

} else {

logger.warn("Employee not found for ID: {}", empId);

return ResponseEntity.status(HttpStatus.NOT\_FOUND).body(null); // or return an error message

}

}

@GetMapping("/leaves")

public List<leavedetails> getAllLeaveDetails() {

return service.getAllLeaveDetails();

}

@GetMapping("/leave/{empId}")

public Optional<leavedetails> getLeaveDetailsByEmpId(@PathVariable String empId) {

return service.getLeaveDetailsByEmpId(empId);

}

@PostMapping("/leave")

public leavedetails addLeaveDetails(@RequestBody leavedetails leaveDetail) {

return service.addLeaveDetails(leaveDetail);

}

@DeleteMapping("/leave/{empId}")

public void deleteLeaveDetails(@PathVariable String empId) {

service.deleteLeaveDetails(empId);

}

@PutMapping("/leave/{empId}/approve")

public ResponseEntity<?> approveLeave(@PathVariable String empId, @RequestParam("startDate") LocalDate startDate) {

try {

service.approveLeave(empId, startDate);

return ResponseEntity.ok("Leave approved successfully.");

} catch (Exception e) {

logger.error("Error approving leave for employee ID: {} on startDate: {}", empId, startDate, e);

return ResponseEntity.status(HttpStatus.INTERNAL\_SERVER\_ERROR).body("Failed to approve leave.");

}

}

}

**Application.properties:**

spring.application.name=leavedemo

spring.datasource.url=jdbc:MySql://localhost:3306/employeedb

spring.datasource.username=root

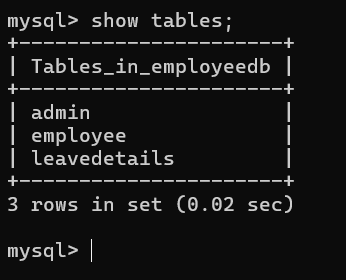
spring.datasource.password=@Kavyagowda23

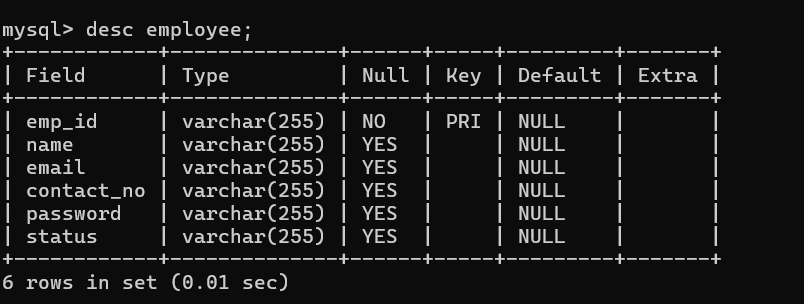
spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

**In mysql check the tables:**





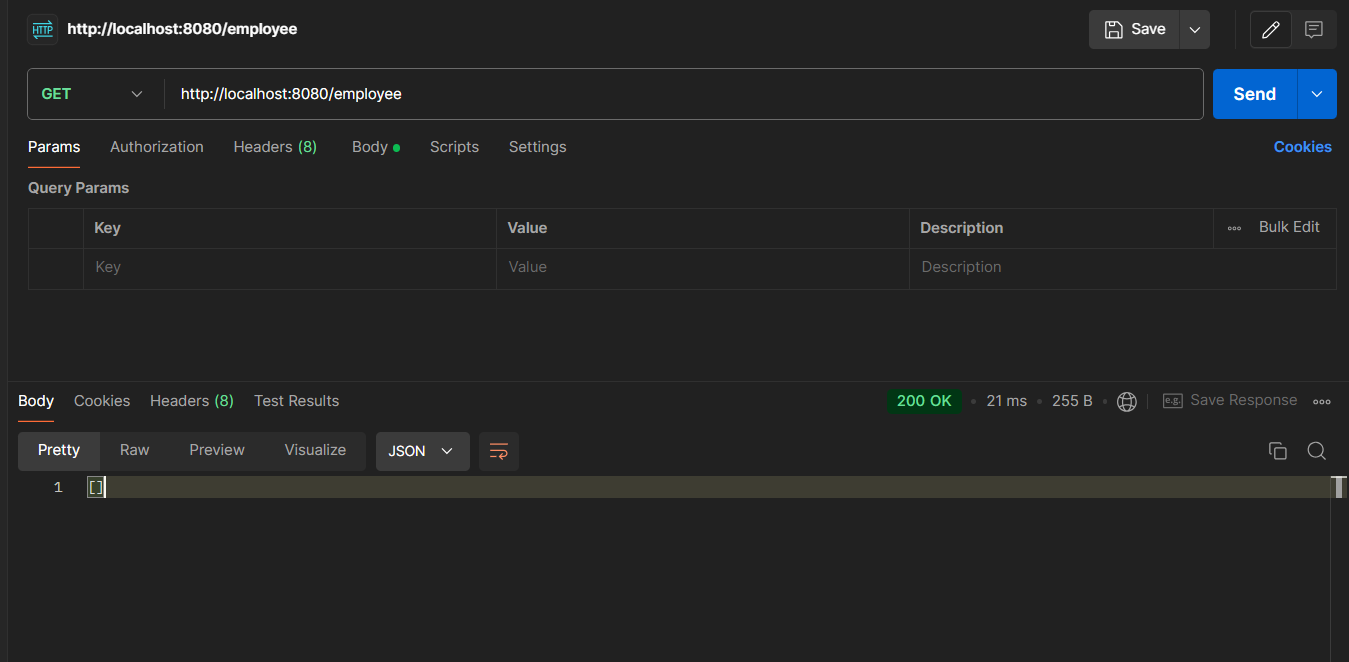
**Now run the code:**

Right click to your project🡪Run as🡪Java Application🡪Search your project🡪Click on OK.

Go to the POSTMAN🡪Set the request GET🡪<http://localhost:8080/employee>

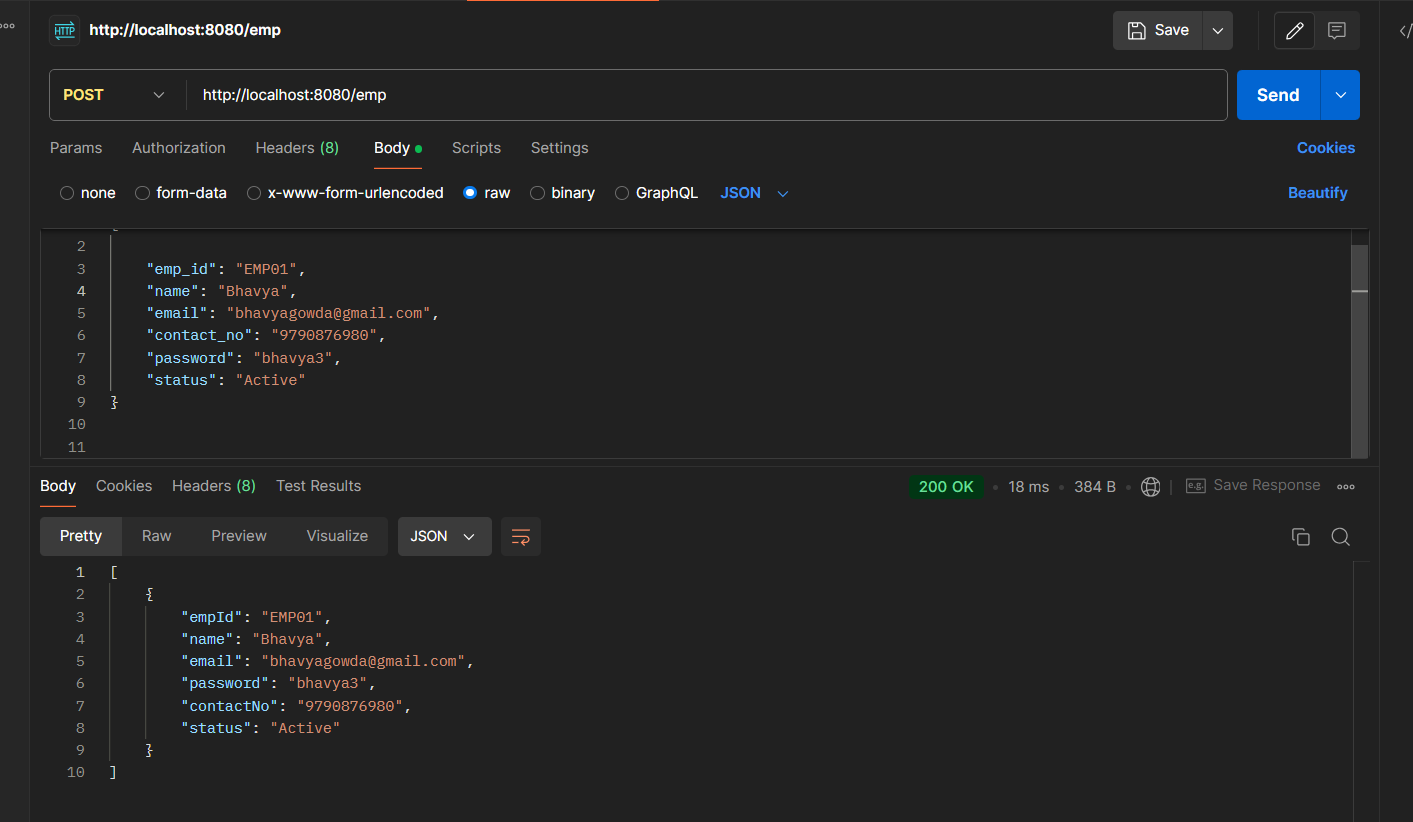
Click on send.

You will receive an empty table

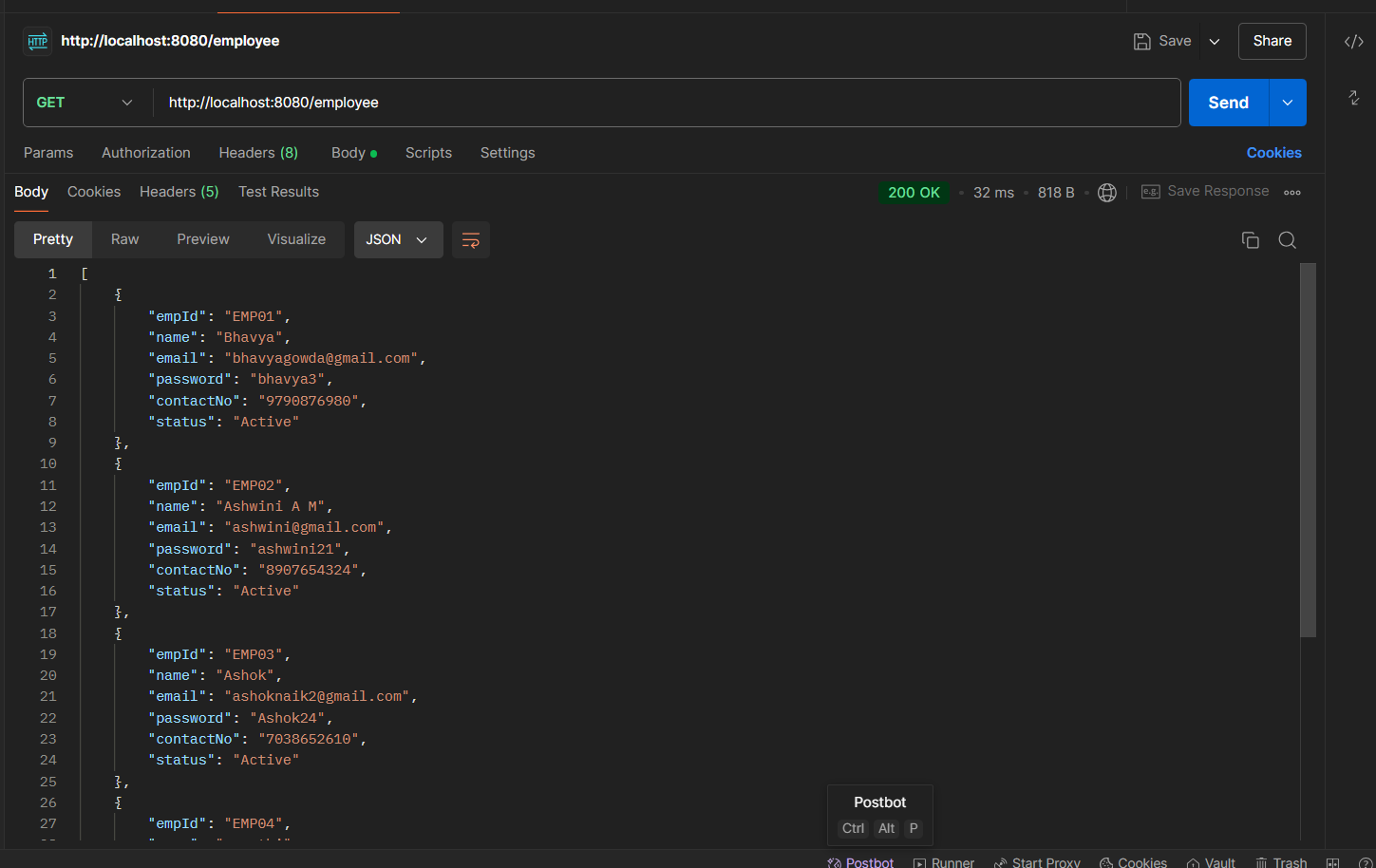


Set the request in POST🡪Body-select row🡪Write employee details to insert into employee table🡪Click on send

<http://localhost:8080/emp>



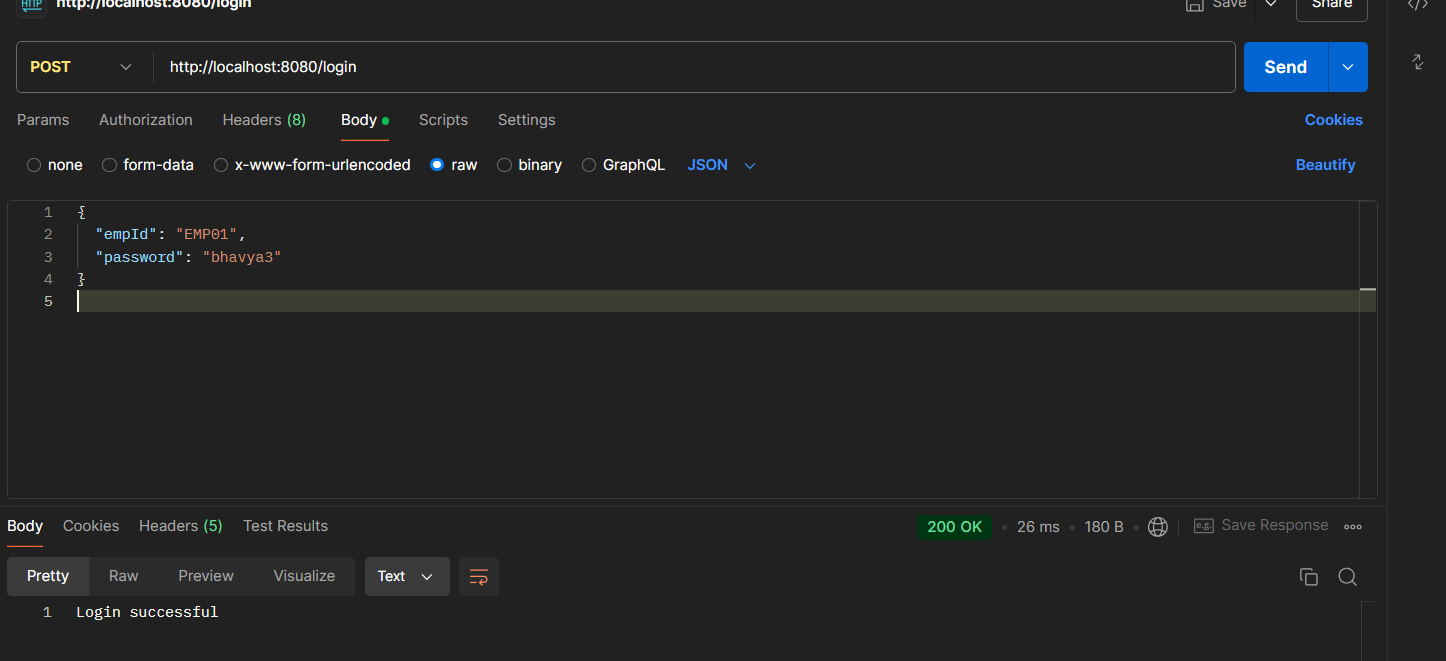
Then insert many Employee: Now we can see the employee details after insertion in GET request🡪 <http://localhost:8080/employe>e





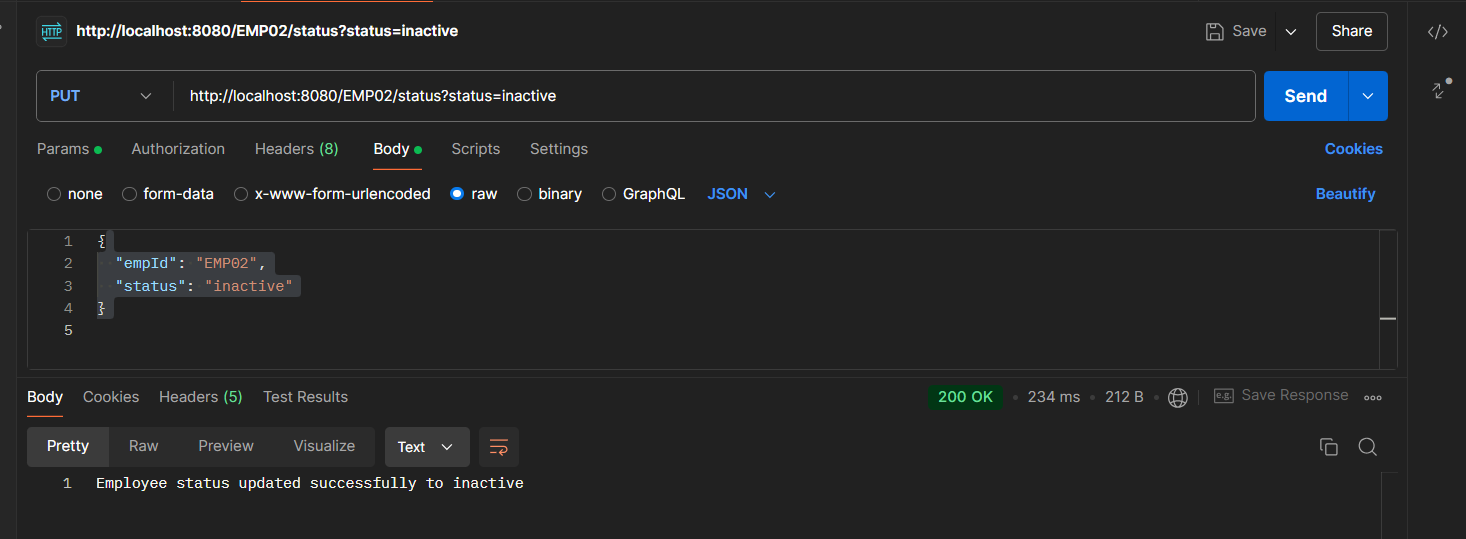
Set the request in POST🡪to login as employee🡪click on send

[http://localhost:8080/ login](http://localhost:8080/%20login%20)



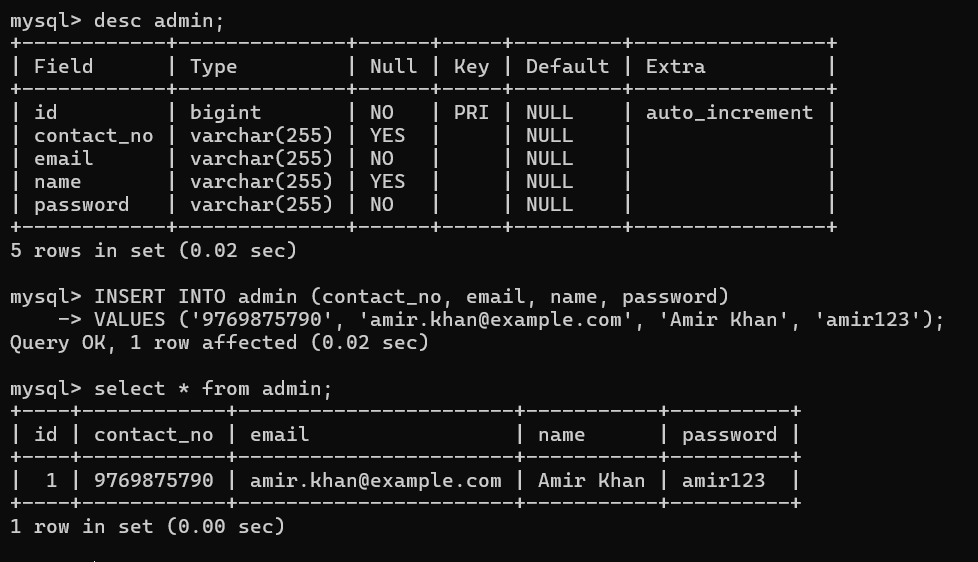
Set the request in PUT

<http://localhost:8080/EMP02/status?status=inactive>



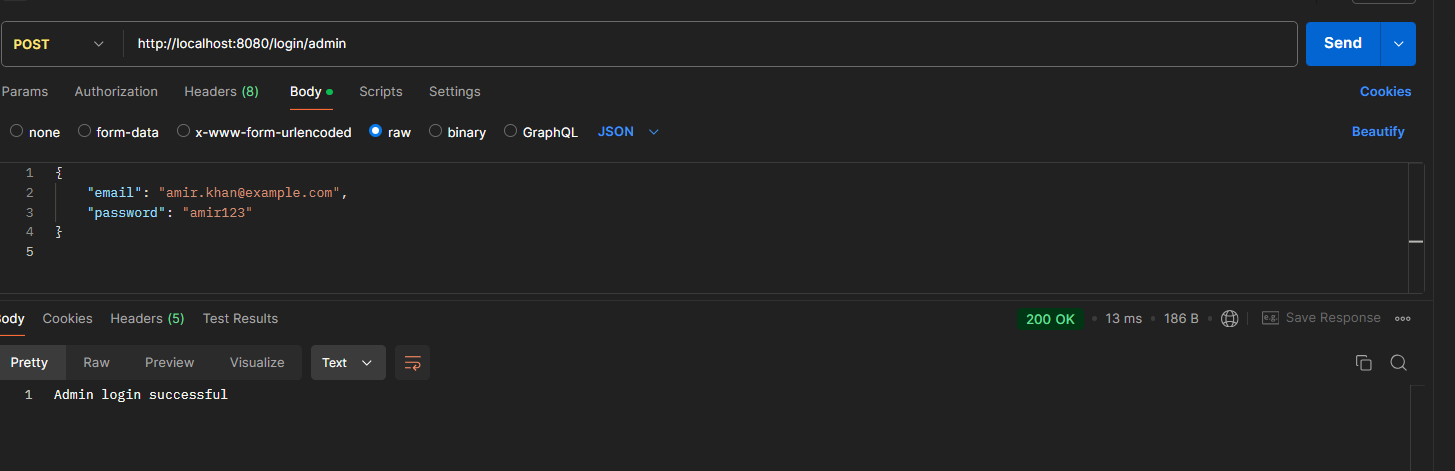
**Admin**

Now to insert the admin🡪Go to the command prompt and give the command as follows:

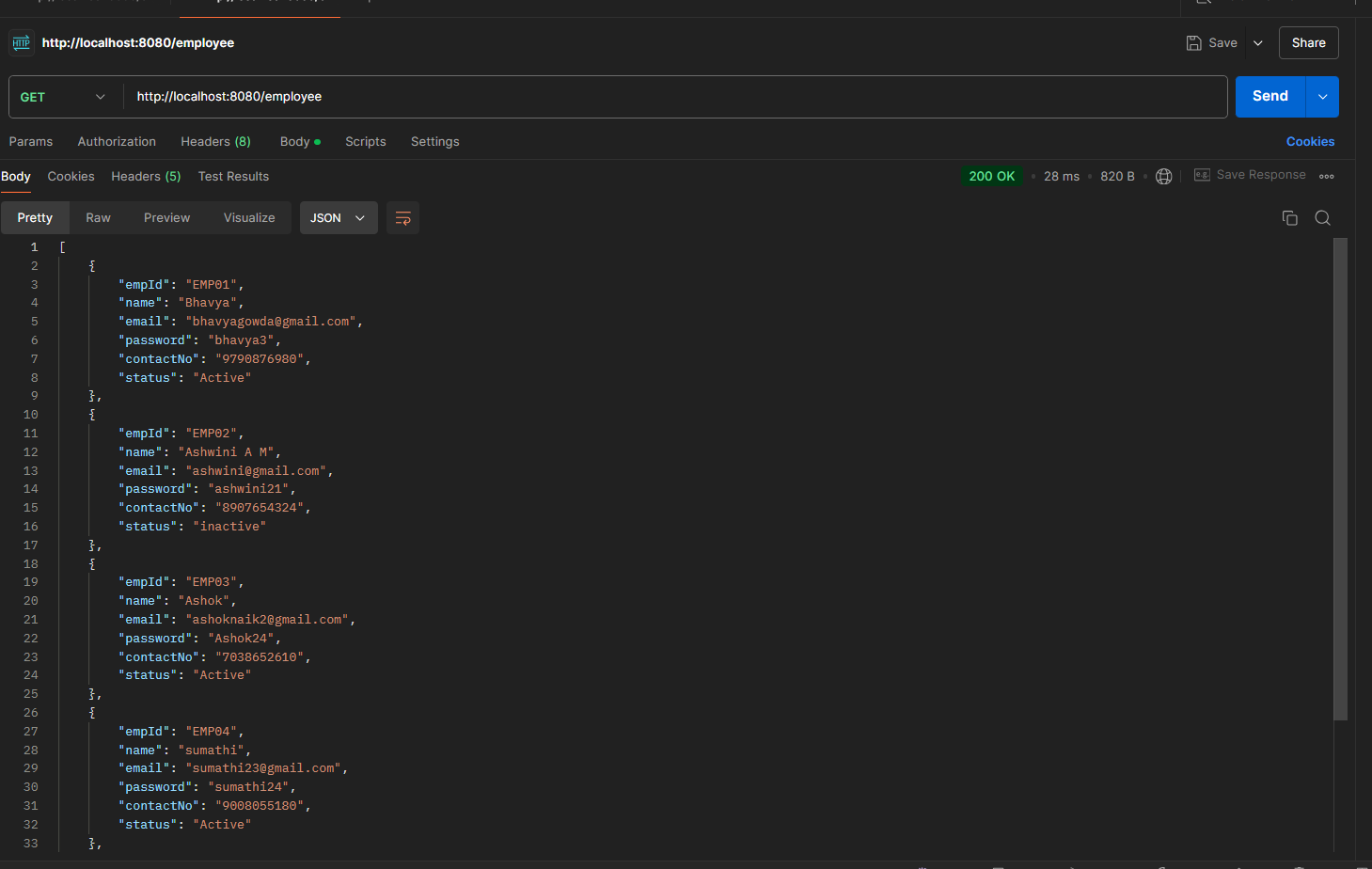


Set the request in POST🡪to login as admin🡪click on send

[http://localhost:8080/ login/admin](http://localhost:8080/%20login/admin%20)

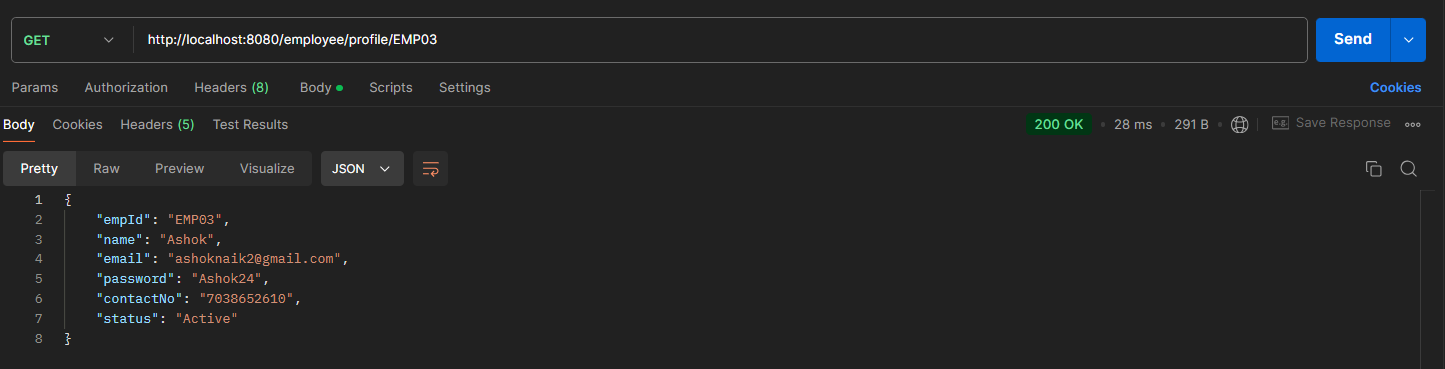


Now we can see the employee details after insertion in GET request🡪 <http://localhost:8080/employee>

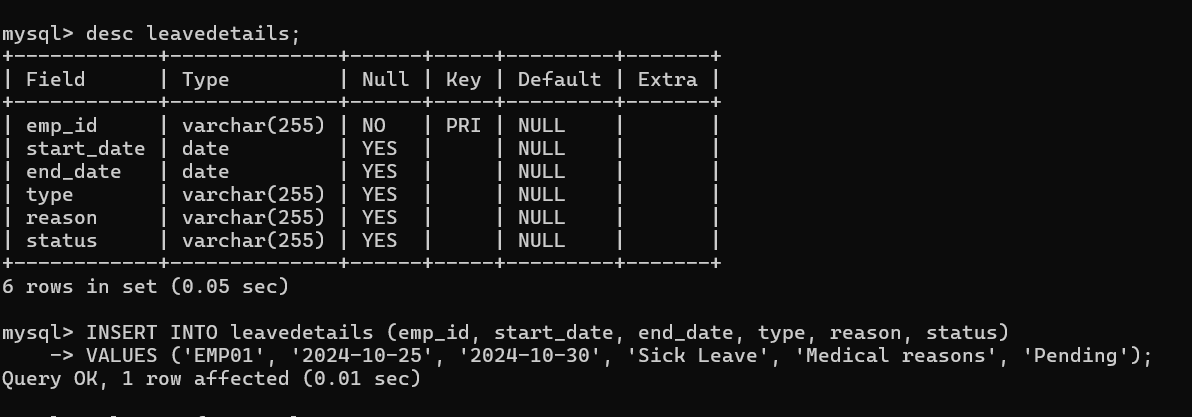


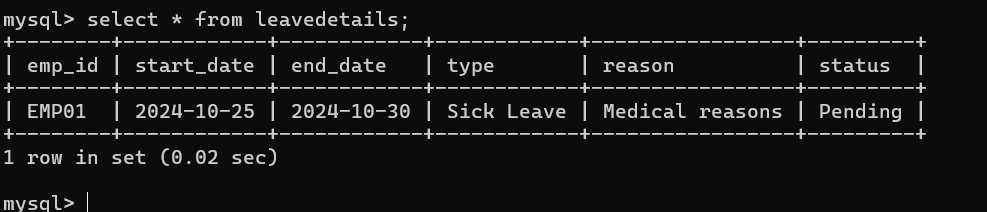
Now we can see the employee profile after insertion in GET request🡪

<http://localhost:8080/employee/profile/EMP03>



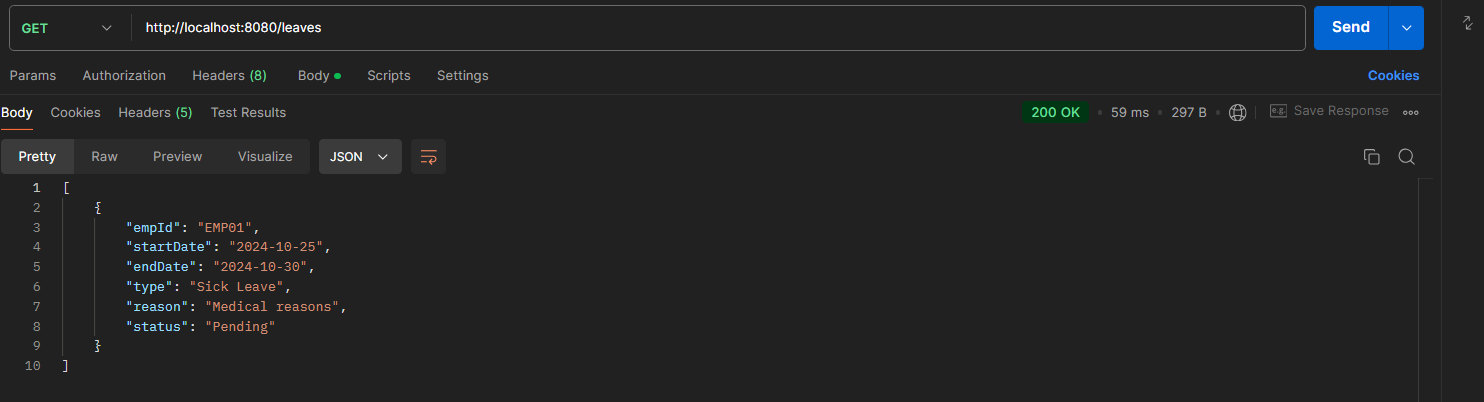
Now to insert the leavedetails🡪Go to the command prompt and give the command as follows:





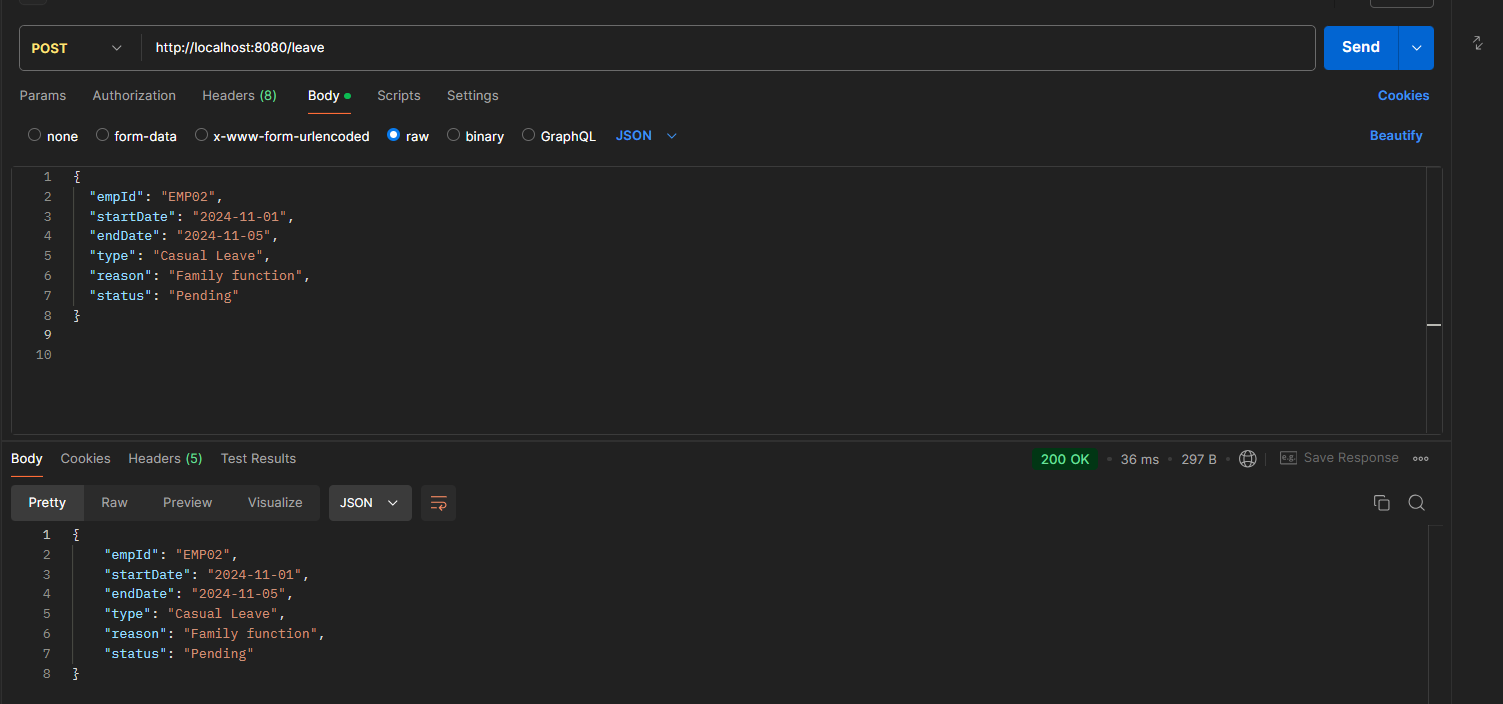
Now go to the POSTMAN🡪set the request to GET and see the output.

<http://localhost:8080/leaves>



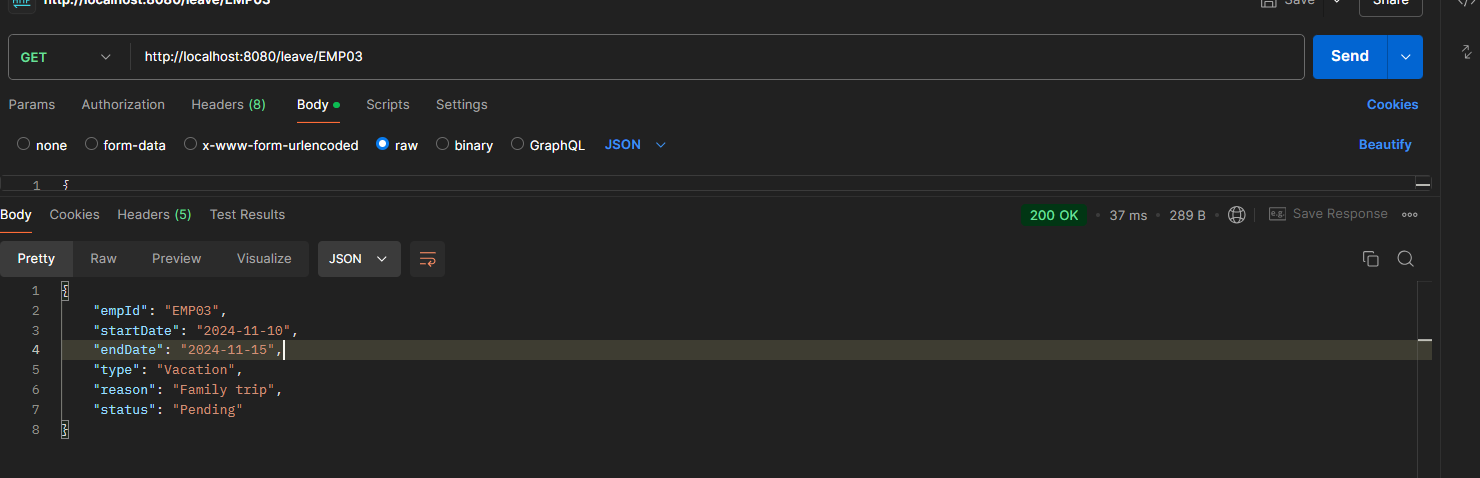
Set the request in POST🡪to leavedetails of employee🡪click on send

<http://localhost:8080/leave>



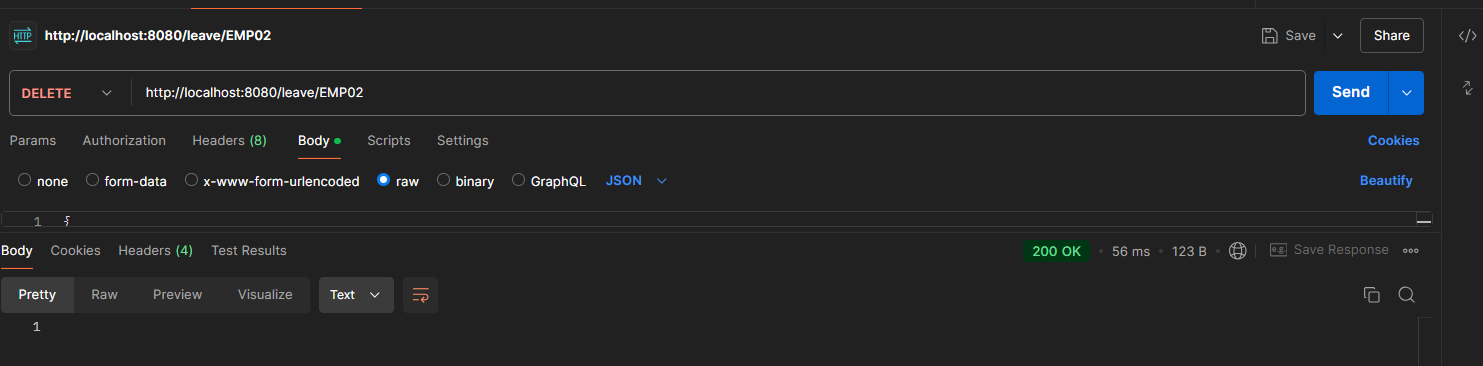
Get any one leavedetails the request in GET.

[http://localhost:8080/leave/EMP03](http://localhost:8080/leave/EMP03e)



Set the request in Delete🡪To delete specify the empId.

<http://localhost:8080/leave/EMP02>



Then check the leavedetails from Command Prompt

