Employee Management System

Introduction

The Employee Management System (EMS) is a software application designed to streamline the process of managing employee data within an organization. It provides functionalities for creating, reading, updating, and deleting employee records efficiently.

Technologies Used

Frontend: HTML, CSS, JavaScript

Backend: Spring Boot3

Database: MySQL

API: Axios

CRUD Operations

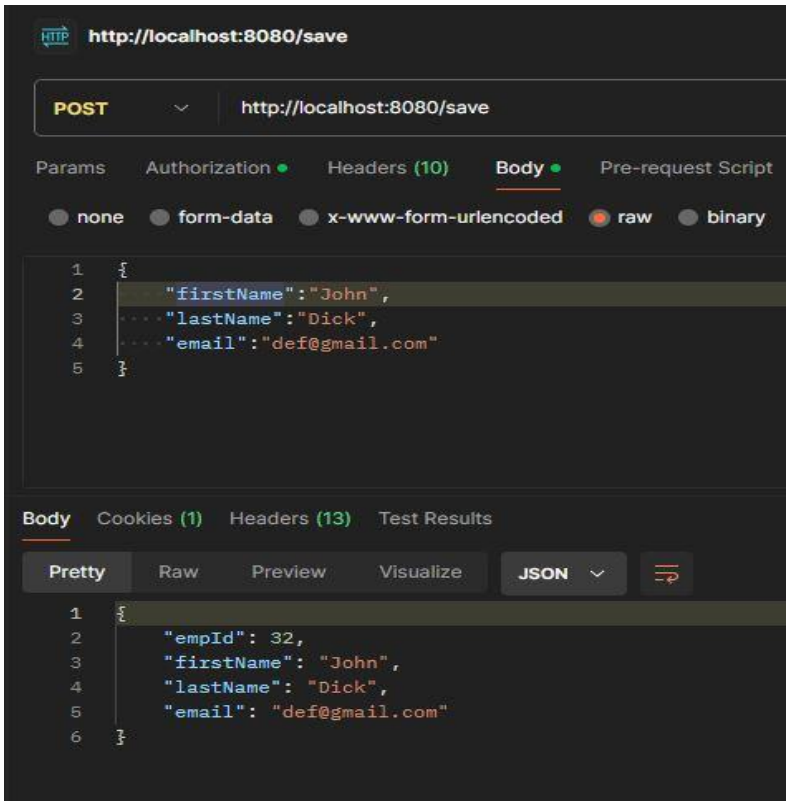
1. Create

Endpoint: http://localhost:8080/save

Method: POST

Request Payload: Employee details (firstName, lastName, email)

Response: empId, firstName, lastName, email.



2. Read

Endpoint: http://localhost:8080/employees

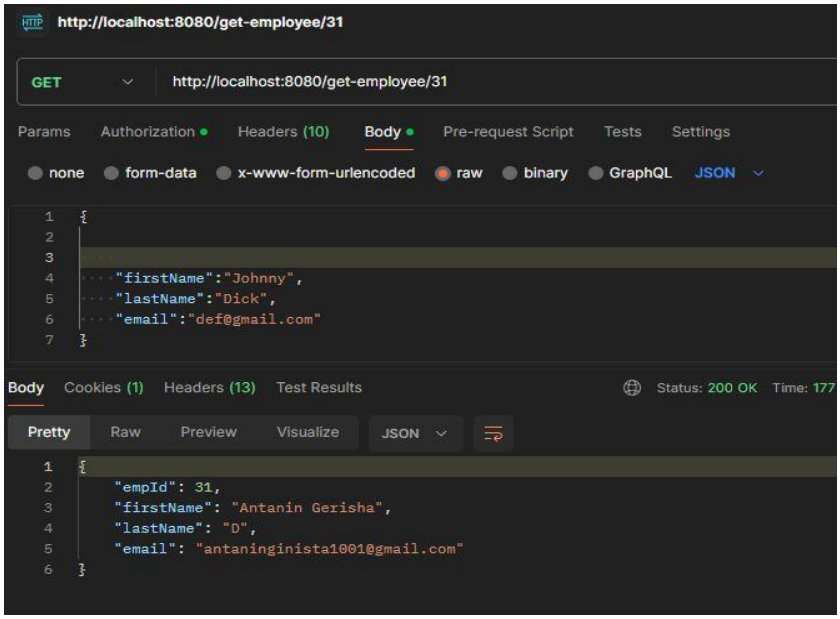
Method: GET

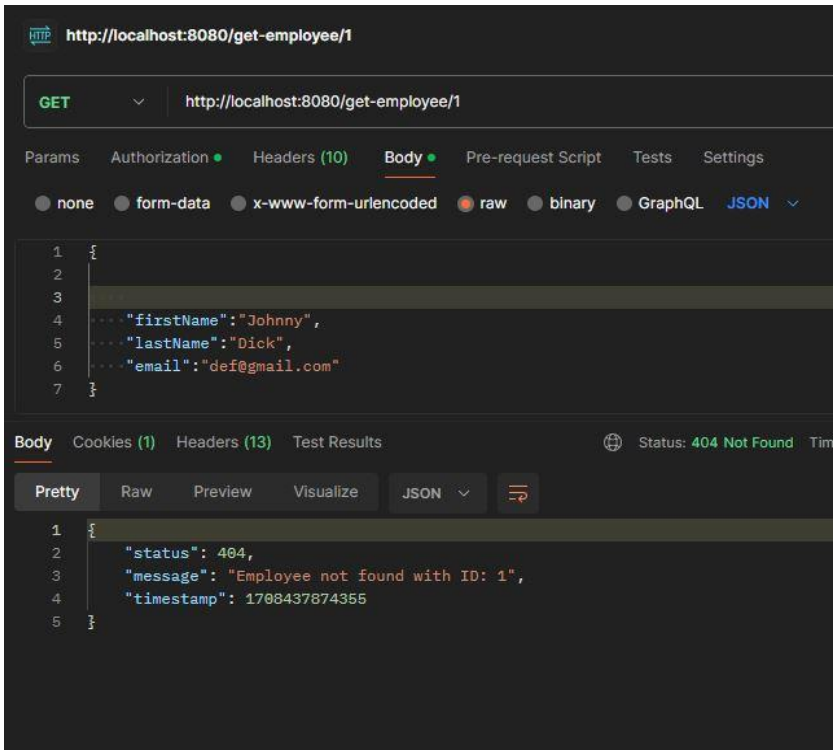
Response: List of employee records.

Endpoint: http://localhost:8080/get-employee/{id}

Response: Employee details corresponding to the provided ID (if employee is present)

Status, message, timestamp (if employee is not present)





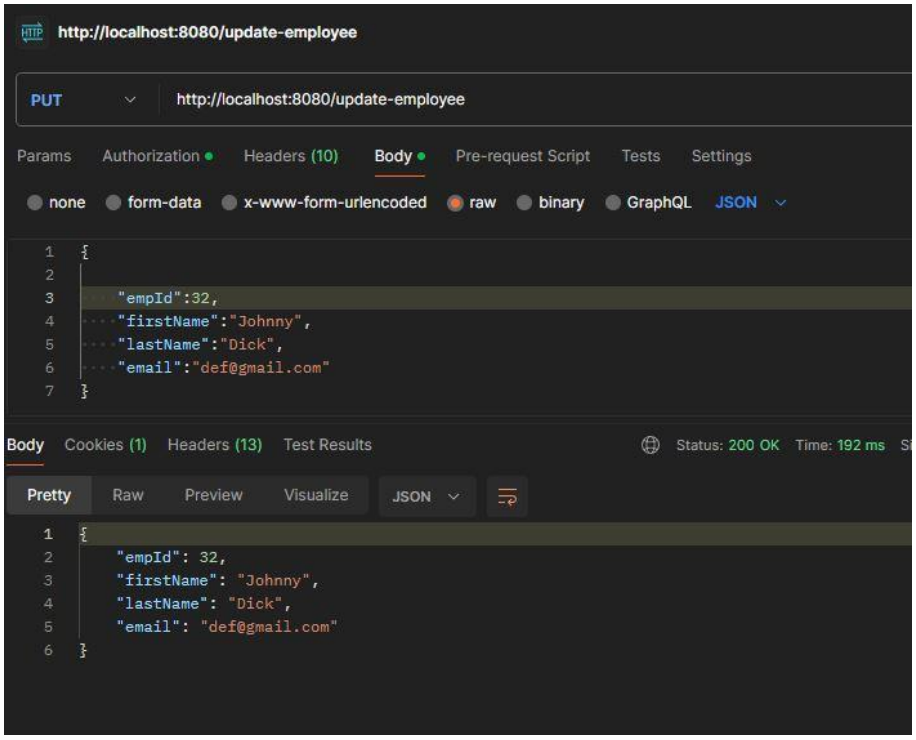
3. Update

Endpoint: http://localhost:8080/update-employee

Method: PUT

Request Payload: Updated employee details.

Response: Updated employee details.

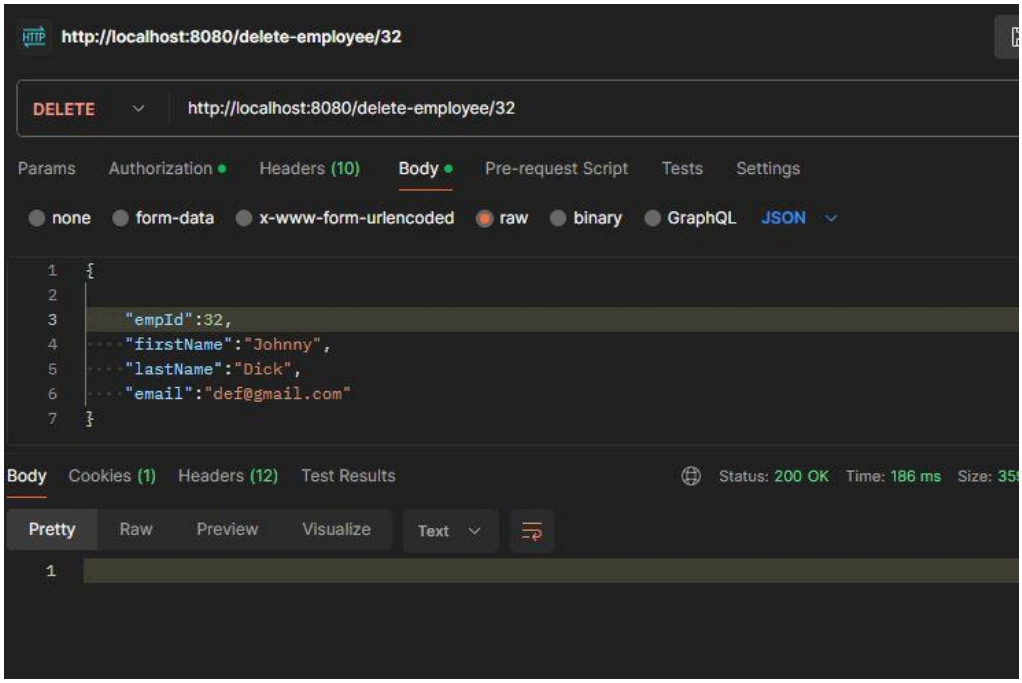


4. Delete

Endpoint: http://localhost:8080/employees /delete-employee/ { id }

Method: DELETE

Response: Success message.



Authentication Details

User Details:

Username: user

Password: password

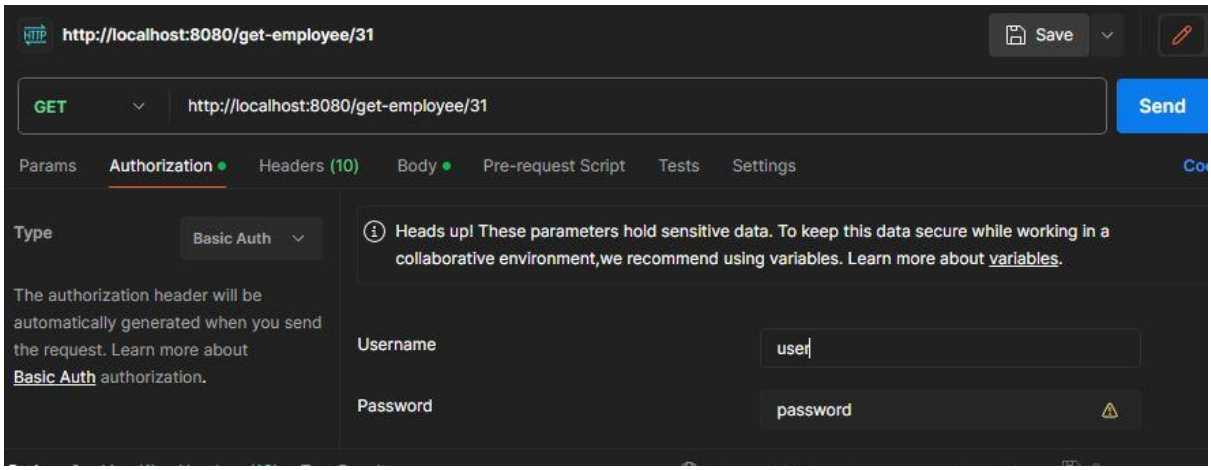
Roles: group1

Admin Details:

Username: admin

Password: password

Roles: group1

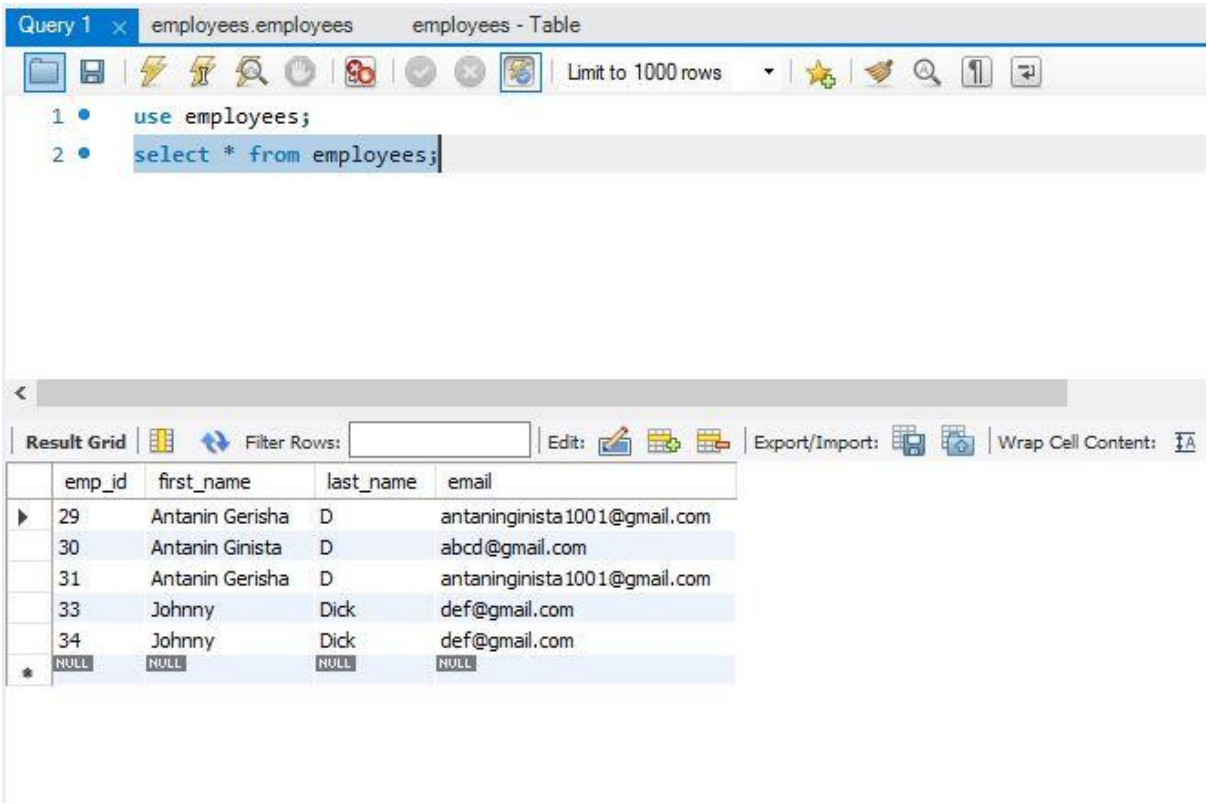


```java
package com.guvi.employeeManagement.config;

import org.springframework.context.annotation.Bean;

@Configuration
@EnableWebSecurity
public class ApplicationSecurityConfig {

    @Bean
    public UserDetailsService users() {
        // The builder will ensure the passwords are encoded before saving in memory
        UserBuilder users = User.withDefaultPasswordEncoder();
        UserDetails user = users.username("user").password("password").roles("group1").build();
        UserDetails admin = users.username("admin").password("password").roles("group1").build();
        return new InMemoryUserDetailsManager(user, admin);
    }

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http.authorizeHttpRequests((authz) -> authz.anyRequest().authenticated())
                                .httpBasic()
                                .and()
                                .csrf()
                                .disable()
                                .headers().frameOptions().disable();
        return http.build();
    }

}
```

ORM Mapping

```java
package com.guvi.employeeManagement.entity;

import jakarta.persistence.Column;

@Entity
@Table(name = "employees")
@Getter
@Setter
@NoArgsConstructor
public class Employee {

    @Id
    @Column(name = "empId")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int empId;

    @NotNull(message = "First Name is required")
    @Column(name = "firstName")
    private String firstName;

    @NotNull(message = "Last Name is required")
    @Column(name = "lastName")
    private String lastName;

    @Email(message = "Invalid Email Address")
    @Column(name = "email")
    private String email;
}
```

Database



Conclusion

The Employee Management System provides a comprehensive solution for managing employee data efficiently

Demo Url: https://drive.google.com/file/d/1BaU9X-xoLg2pPWHEPbrt4vzZjPzLxmY_/view