package com.example.springprofile.entity;

import jakarta.persistence.*; import lombok.Data;

@Entity @Data public class User {

```
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;

private String name;

private String email;
```

}

By using LOMBOK framework i learn to build a sample project without manually write code like getter setter argsconstructor etc bcoz it have annotation called @data here i used and creted classes like service ,repository ,controller to manage rest apis with endpoints ..

---

package com.example.springprofile.repository; import com.example.springprofile.entity.User; import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> { }

package com.example.springprofile.service;

import com.example.springprofile.entity.User; import com.example.springprofile.repository.UserRepository; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service;

import java.util.List;

@Service public class UserService {

```
@Autowired
private  UserRepository userRepository;

public List<User> getAllUsers() {
    return userRepository.findAll();
}
```

```
    public User getUserById(Long id) {
        return userRepository.findById(id).orElse(null);
    }


    public User createUser(User user) {
        return userRepository.save(user);
    }


    public void deleteUser(Long id) {
        userRepository.deleteById(id);
    }
```

}

@RequestMapping("/api/users") public class UserController { @Autowired private UserService userService;

```
    @GetMapping
    public List<User> getAllUsers() {
        return userService.getAllUsers();
    }


    @GetMapping("/{id}")
    public User getUserById(@PathVariable Long id) {
        return userService.getUserById(id);
    }


    @PostMapping
    public User createUser(@RequestBody User user) {
        return userService.createUser(user);
    }


    @DeleteMapping("/{id}")
    public void deleteUser(@PathVariable Long id) {
        userService.deleteUser(id);
    }
```

}

Ans i used springboot profiles for configure application based on environment like develope

,testing ,production..etc here i used to profile in h2 in memory database to test those profile will works but there is some issue in my configuration its works but the excepted output is different i will sure find out the mistake what i had done ..

spring.application.name=springprofile spring.profiles.active=dev server.port=8080

spring.datasource.url=jdbc:h2:mem:devdb spring.datasource.driverClassName=org.h2.Driver spring.datasource.username=sa spring.datasource.password=password spring.jpa.hibernate.ddl-auto=create server.port=8081

Test Database Configuration

spring.datasource.url=jdbc:h2:mem:testdb spring.datasource.driverClassName=org.h2.Driver spring.datasource.username=sa spring.datasource.password=password spring.jpa.hibernate.ddl-auto=create server.port=8082