**Online Appointments Booking Application**

**1.ABSTRACT:** A simple Online Appointment booking application built using Spring Boot, Java, Spring MVC, Thymeleaf, Spring Jdbc, and MySQL database. In this project doctor will book the appointment to the patients.  Book, edit, or delete appointments from the H2 console using JavaEE, Spring Boot, and SQL. Styled with Bootstrap.

**2.USE CASE DIAGRAM:**

Diagram

Description automatically generated

**3.IMPLEMENTATION DETAILS:**

I have implemented a wide variety of operations for the Menu, Order entities in our food ordering application web application.

**Tools and Technologies Used**

* Java 16
* Spring Boot
* Spring MVC
* Spring Data JDBC(Hibernate)
* H2 database
* Thyme leaf
* Eclipse STS

**1. Create Spring Boot Project**

After Creating a Spring Boot Project in Spring Tool Suite and add the dependencies while creating spring boot project using spring initializer:

- Spring Web

- Thymeleaf

- Spring Data JDBC

- H2 database Driver

- Spring Boot Devtools

**2. Create Spring Boot Project Structure**

Let's create the below packages in our Spring boot project:

**Controller**- When a user requests from the browser, it first goes to the controller. From the controller, the request is passed to the service layer. So, in this layer A OrderController,RestaurantController class with handler methods is created to handle the get and post requests.

**Service**- This service layer has the business logic and is responsible to send the further request to repository. In this layer, an interface OrderService is created which has all the abstract methods.

**ServiceImpl-** In this folder, the interface of the service layer is implemented by the class OrderImplementation which implements the functionality of all the methods.

**Repository**- In this folder, an interface OrderRepository,MenuRepository that extends the JPARepository is created which acts as a mediator between service and the database.

**Entity**- In this folder, the entity Order is created which has the attributes orderid, ItemName, totalPrice etc. We have also Menu entity which contains dish details.

The annotation @Entity indicates that the class is an entity. This annotation can be used on classes and enums' interfaces.

The JPA annotation @Table provides the database table to which this entity is mapped.

@Id - The @Id JPA annotation gives the entity's primary key.

**Configure Mysql Database:** After configuring the database in the Spring Boot project, Create database restaurant in Mysql Workbench.

**View Layer:** Create Home.html, Orders.html, create\_item.html,checkout.html

I have used *th:each* Thymeleaf attribute in my template to iterate the list of orders and menu items.

**3.Run the SpringBoot application Main class and use the below URL to access the application:**

[**http://localhost:8080/home**](http://localhost:8080/home)

**Output Screenshots:**

**Home Page:**



**List Of Appointments Page:**

Table

Description automatically generated

**Update Appointment:**

Graphical user interface

Description automatically generated with low confidence

**Delete Page:**

Table

Description automatically generated with medium confidence