**INTRODUCTION**

INTRODUCTION

If anybody is ill and wants to visit a doctor for checkup,

he or she needs to visit the hospital and waits until the doctor

is available. The patient also waits in a queue while getting

appointment. If the doctor cancels the appointment for some

emergency reasons then the patient is not able to know about

the cancelation of the appointment unless or until he or she

visits the hospital.

INTRODUCTION

If anybody is ill and wants to visit a doctor for checkup,

he or she needs to visit the hospital and waits until the doctor

is available. The patient also waits in a queue while getting

appointment. If the doctor cancels the appointment for some

emergency reasons then the patient is not able to know about

the cancelation of the appointment unless or until he or she

visits the hospital.

INTRODUCTION

If anybody is ill and wants to visit a doctor for checkup,

he or she needs to visit the hospital and waits until the doctor

is available. The patient also waits in a queue while getting

appointment. If the doctor cancels the appointment for some

emergency reasons then the patient is not able to know about

the cancelation of the appointment unless or until he or she

visits the hospital.

INTRODUCTION

If anybody is ill and wants to visit a doctor for checkup,

he or she needs to visit the hospital and waits until the doctor

is available. The patient also waits in a queue while getting

appointment. If the doctor cancels the appointment for some

emergency reasons then the patient is not able to know about

the cancelation of the appointment unless or until he or she

visits the hospital.

If a person is ill and wants to visit a doctor for a check-up, the person needs to visit the hospital and wait until the doctor is available. The patient also waits in a queue while getting an appointment. If the doctor cancels the appointment for some emergency reason, then the patient is unable to know about the cancellation of the appointment unless or until they visit the hospital. The world is modernizing with leaps and bounds and people have become way too busy to get medical appointments in person and to maintain a proper health care. Hence there is a need for a quicker and easier way for people to access information about doctors near them and also to schedule appointments at their convenience. The main idea of our project “Doctor Appointment Manager” is to provide ease and comfort to patients while taking appointments from doctors and easy cancellation of bookings if the patient is busy at the chosen time and date. The database contains the doctor’s details, patient’s details and appointment details and this entire data is maintained by a website that acts as a server. While beginning the project we divided it into the backend and frontend part and we incorporated CSS for our frontend and Spring Boot for our backend. We have designed our project for the patient that contains a login page and a register page. The patient has to register themselves before logging in to the application. After logging in, the patient has the option of selecting a doctor from the list of doctors and can view the doctor’s details and depending on their requirement select the doctor. The patient can book an appointment based on his or her preferred day or time. The selected day or time slot will be reserved and patient can view their appointment details on the website itself.

Output:

**Files used and the reason:**

* AppointmentApplication – This is the main class for the project.
* ControllerClass – This is the controller class for the project.
* Appointment – This is the Appointment class for the H2 Database.
  + AppointmentDelete – This is a class used for receiving deletion request for an Appointment.
  + AppointmentRepository – This is an interface to implement CRUDRepository as well as a custom repository.
  + AppointmentService – This is a class used for implementing the methods to be used with Appointment.
  + CustomRepository – This is an interface to implement a custom deletion command for the Appointment using the class AppointmentDelete class.
* Doctor – This is the Doctor class for the H2 Database.
  + DoctorRepository – This an interface to implement custom method models.
  + DoctorService – This is a class to implement the custom methods.
* Person – This is the Person class for the H2 Database.
  + PersonRepository – This is an interface to implement CRUDRepository.

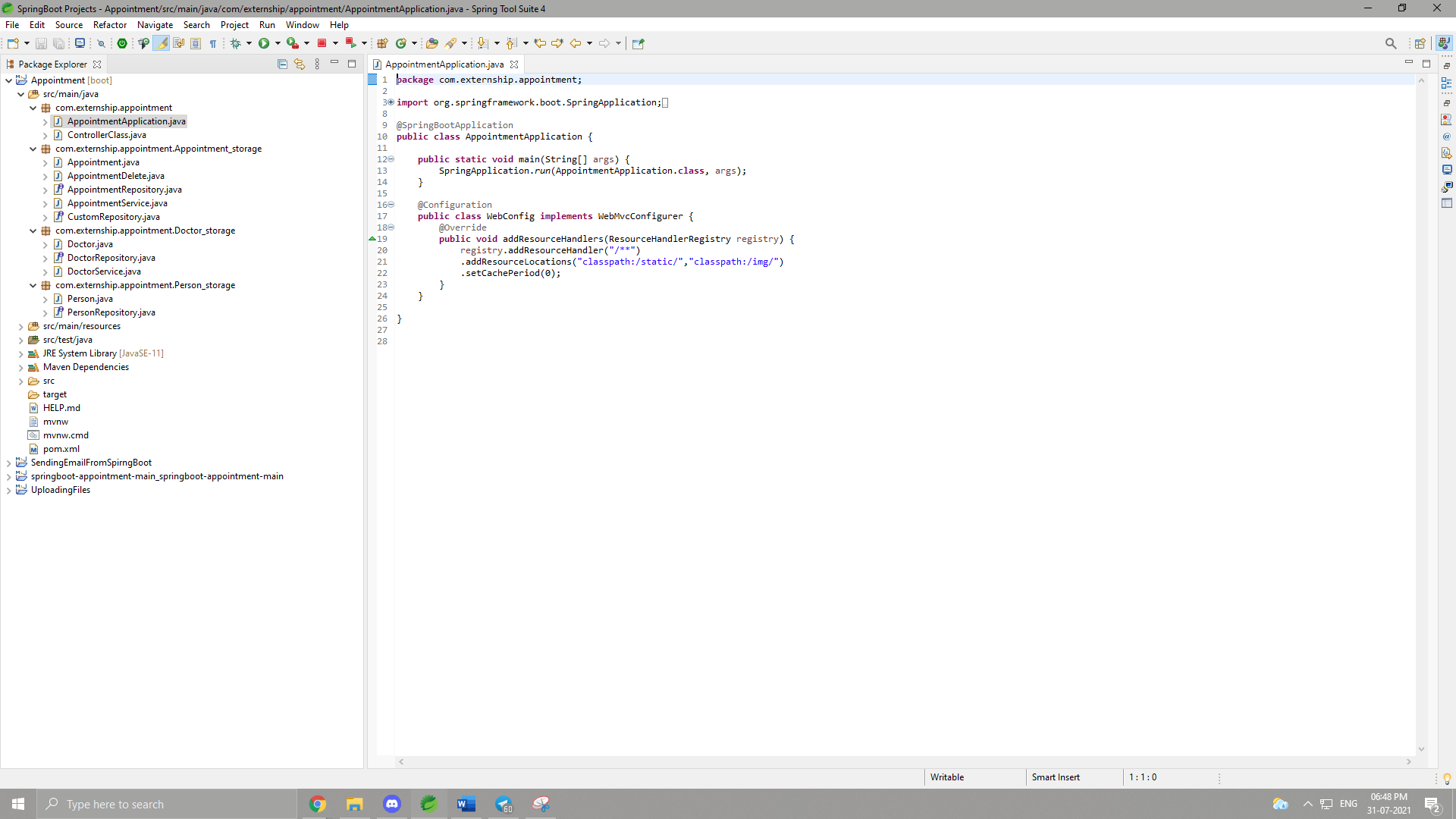
Annotations used in the ControllerClass:

* @Controller
* @Autowired
* @GetMapping
* @PostMapping

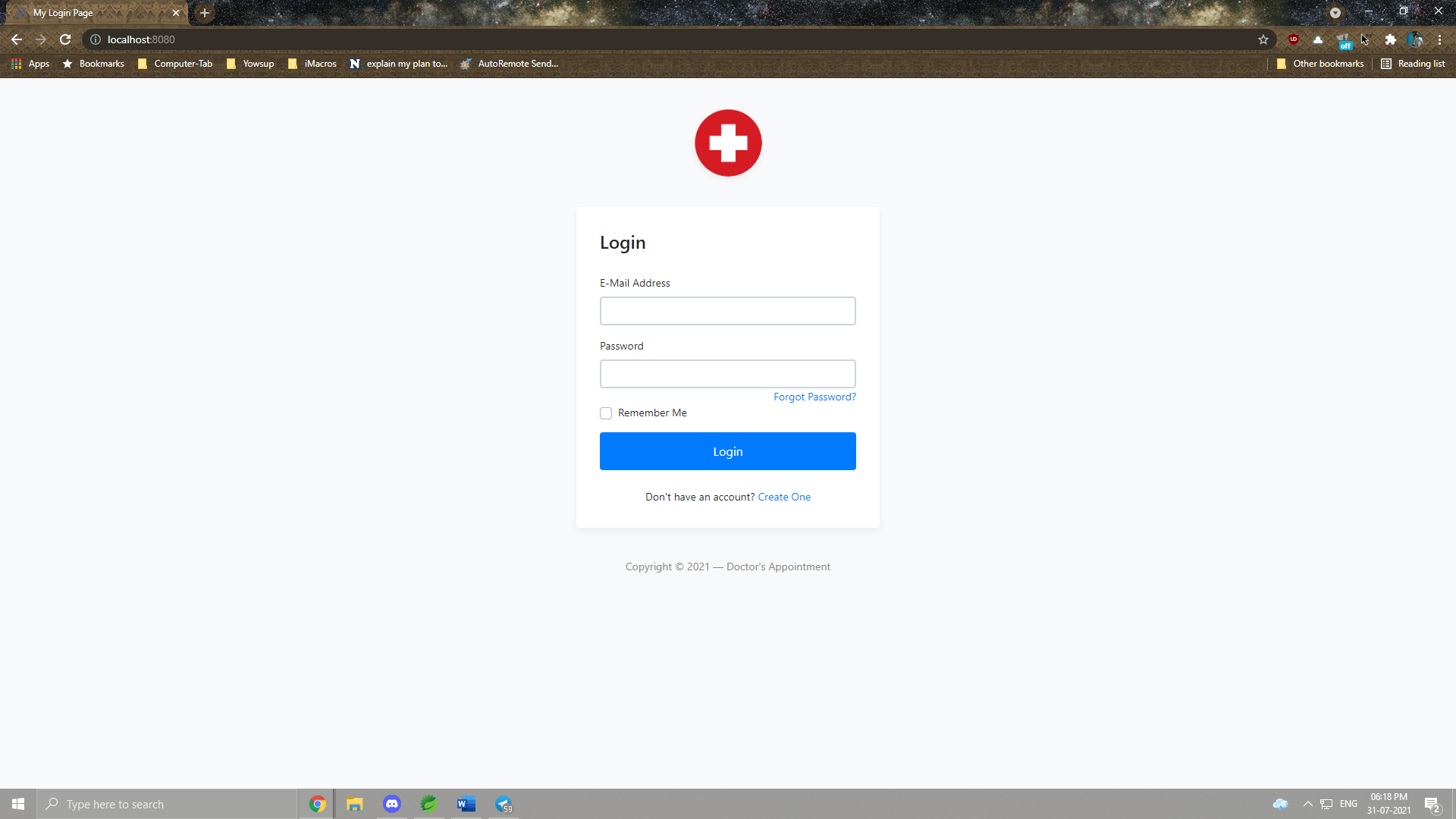
Other Annotations used:

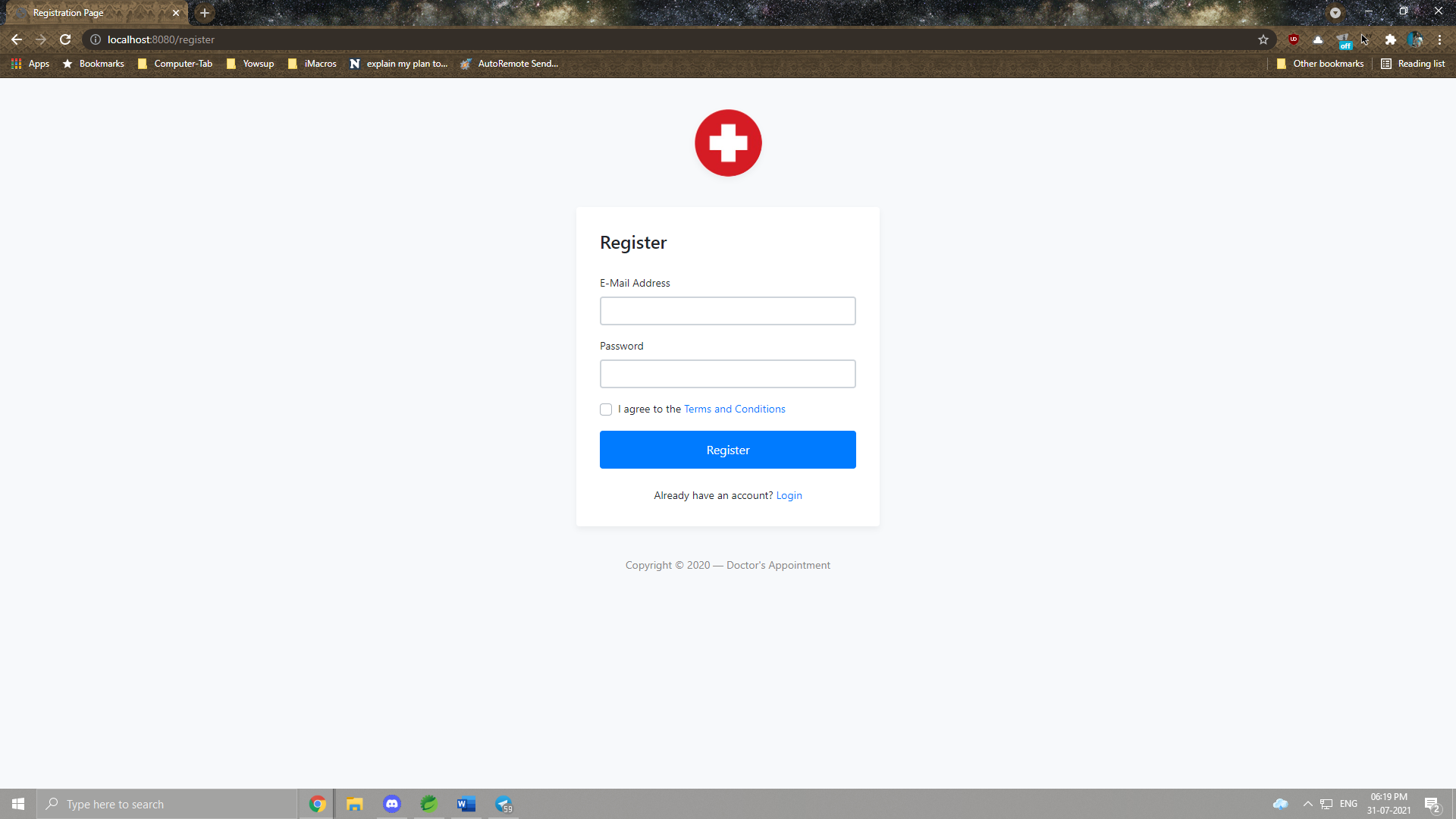
* @Component
* @Entity
* @Service
* @Id

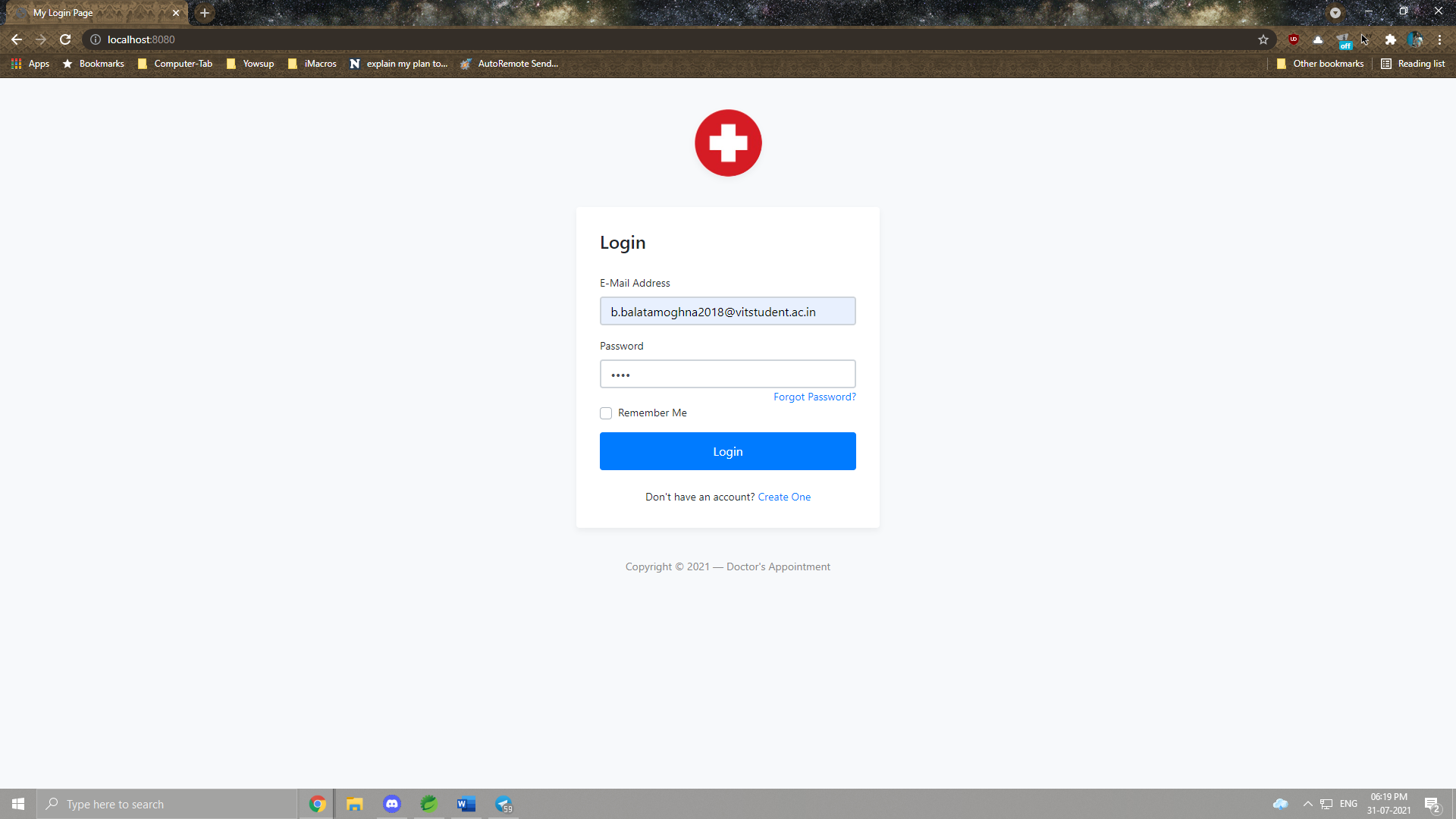
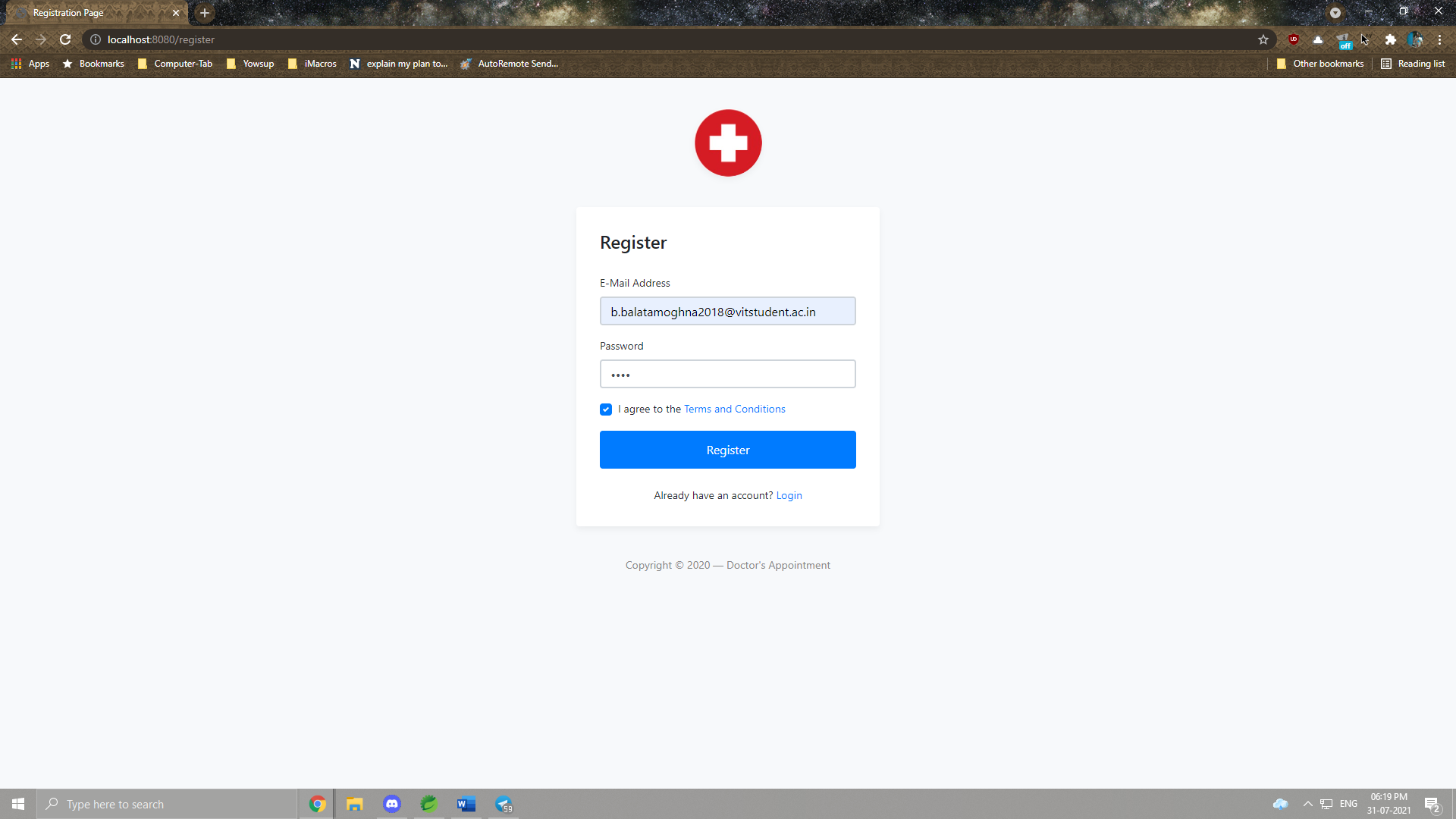
Code:

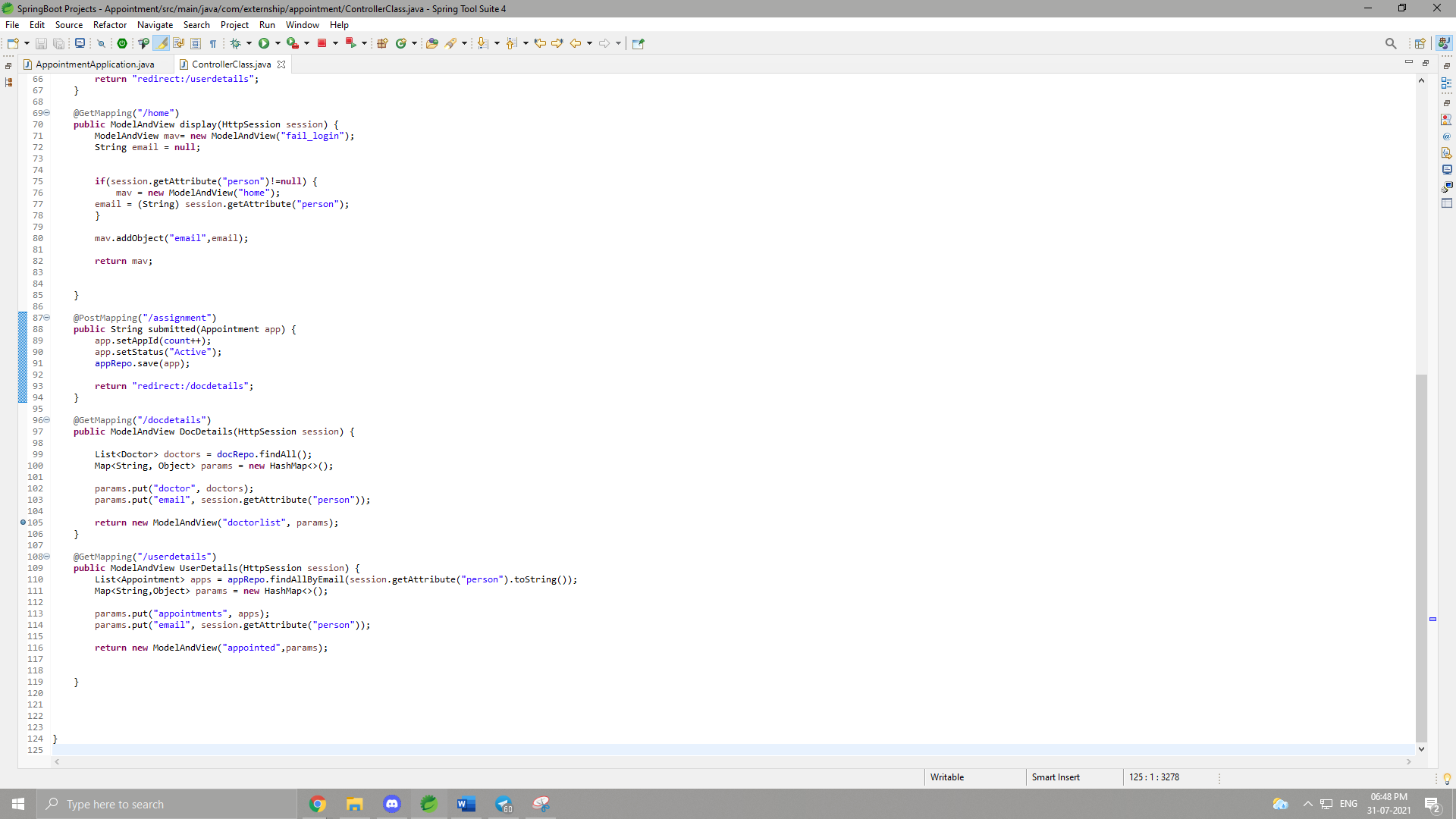
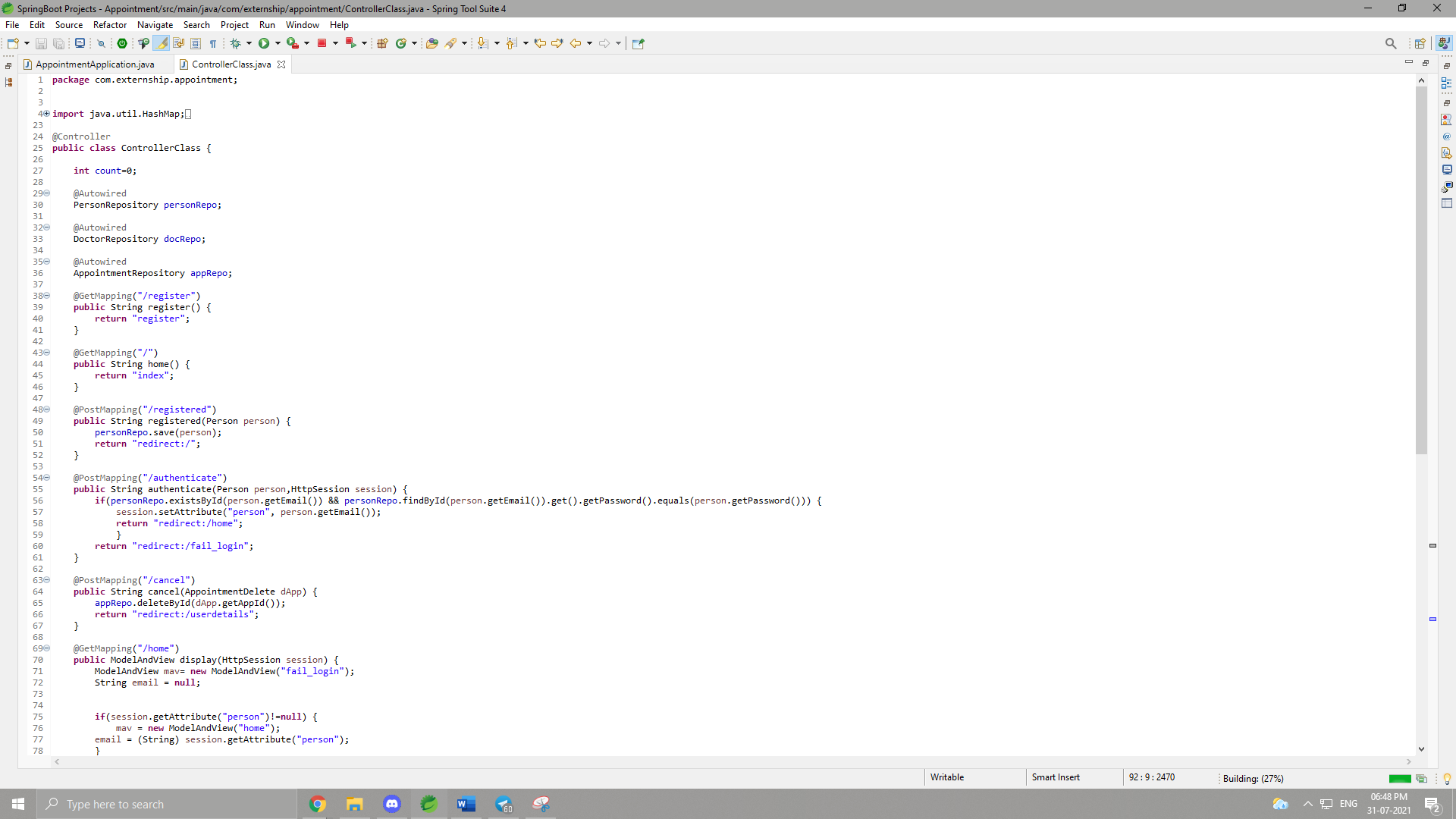


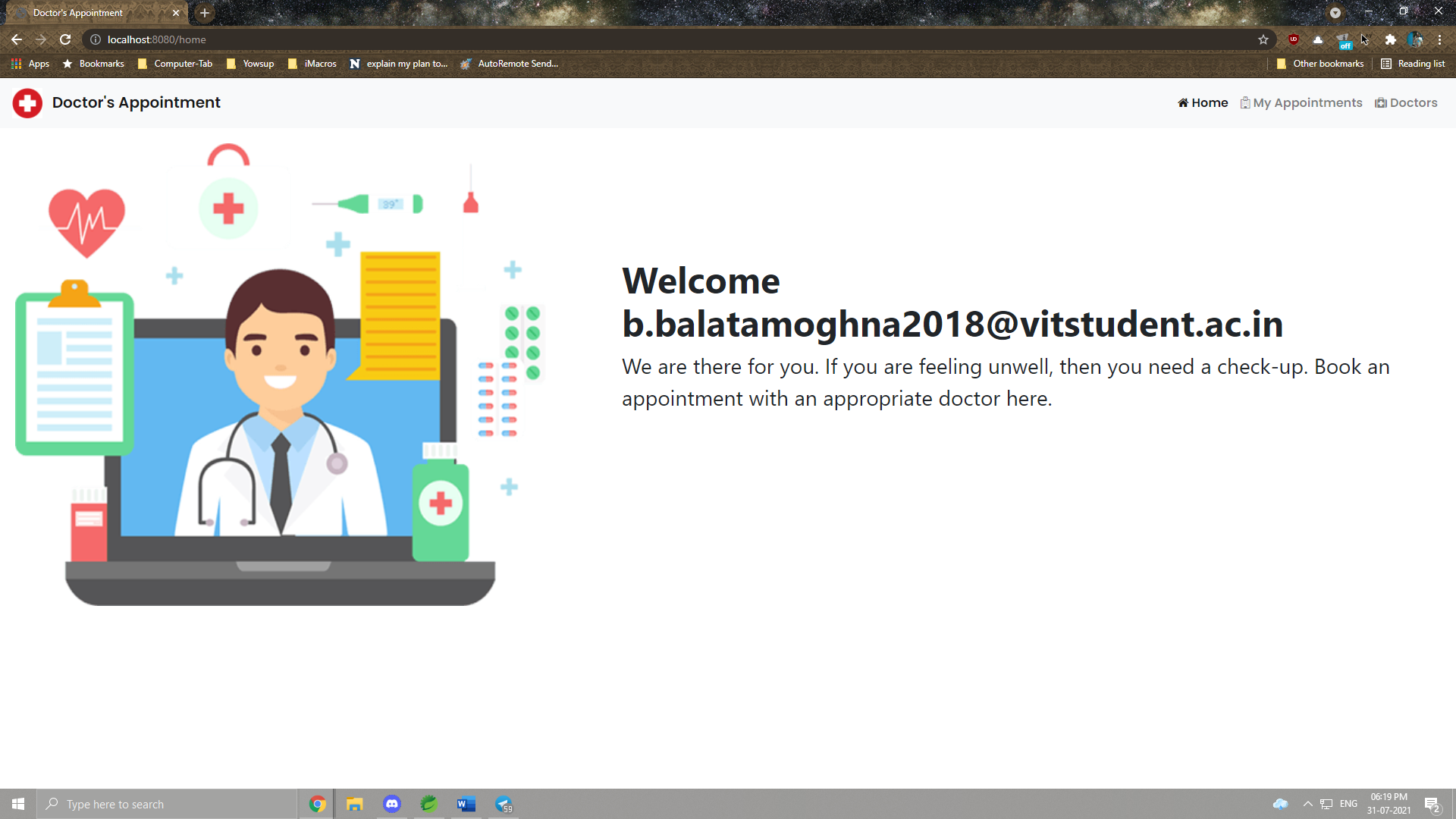
Output:

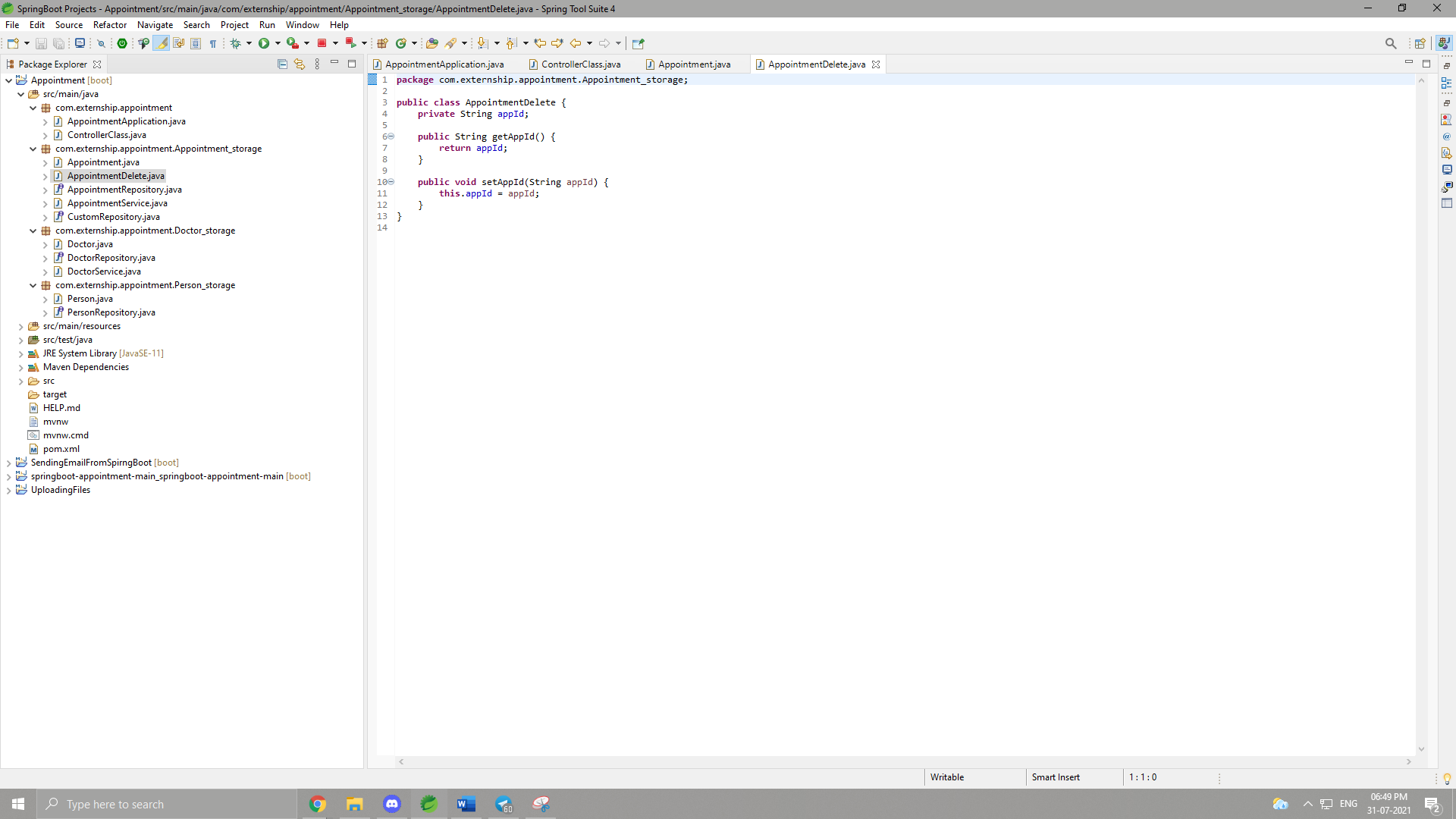
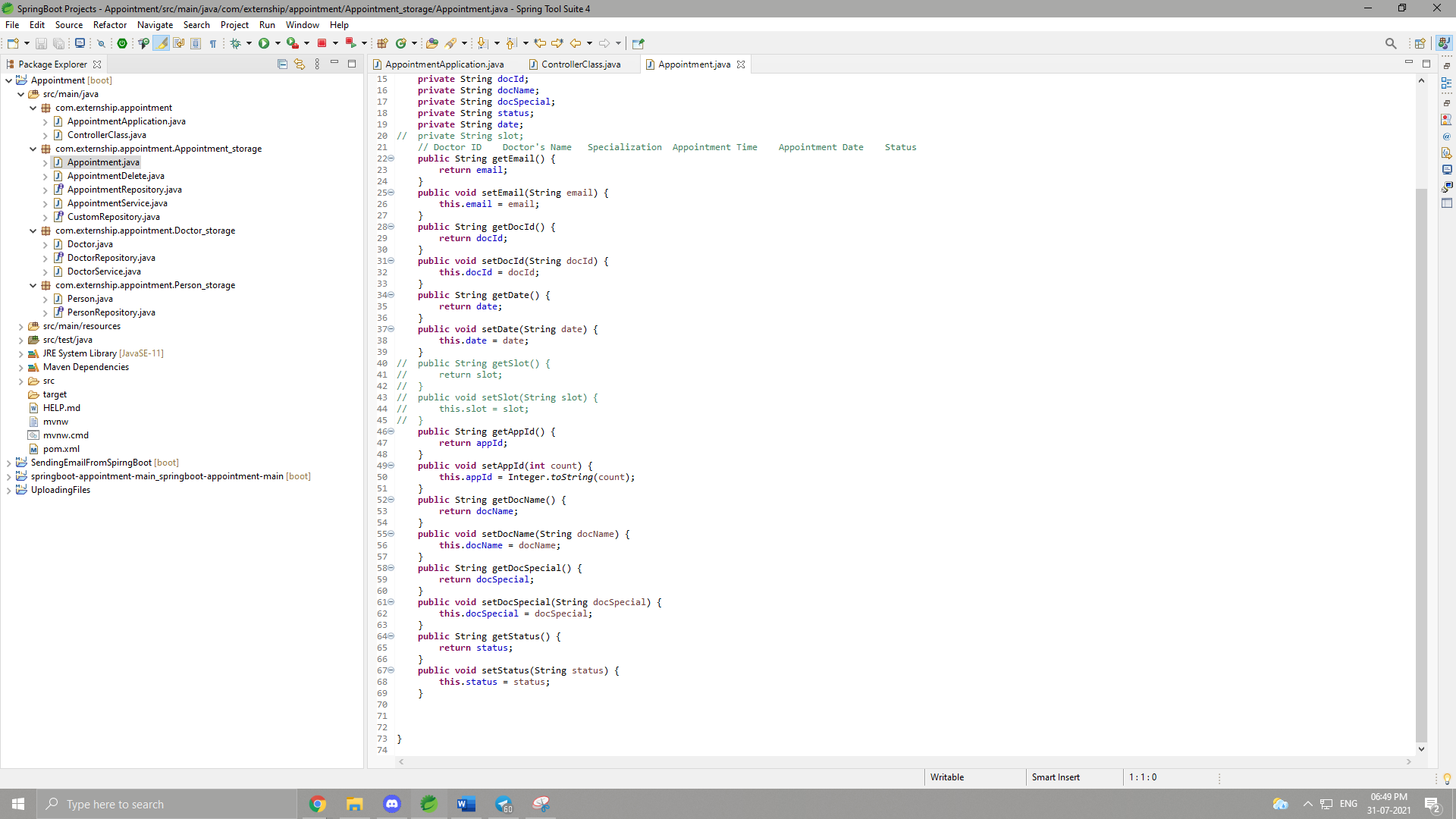
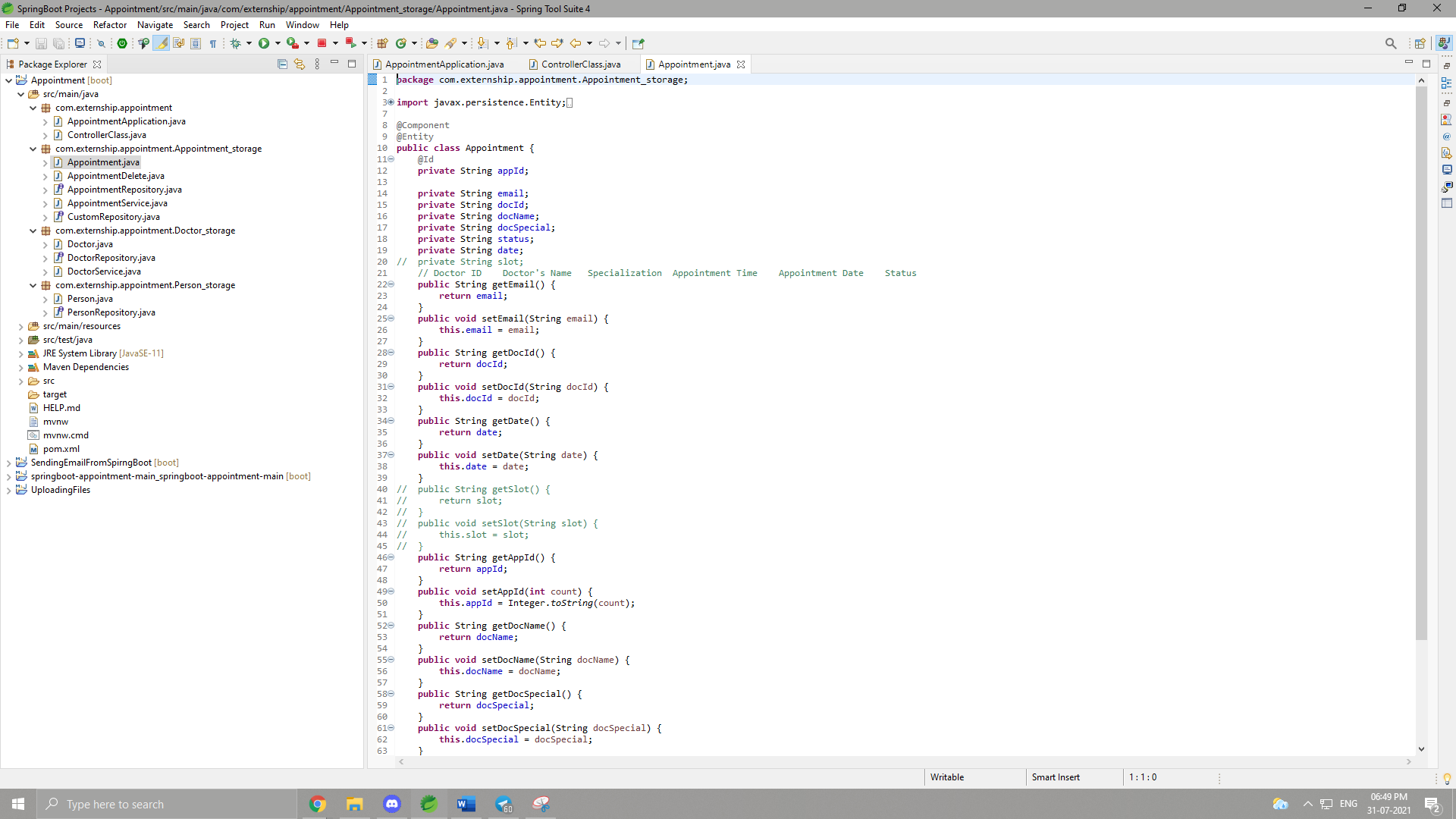


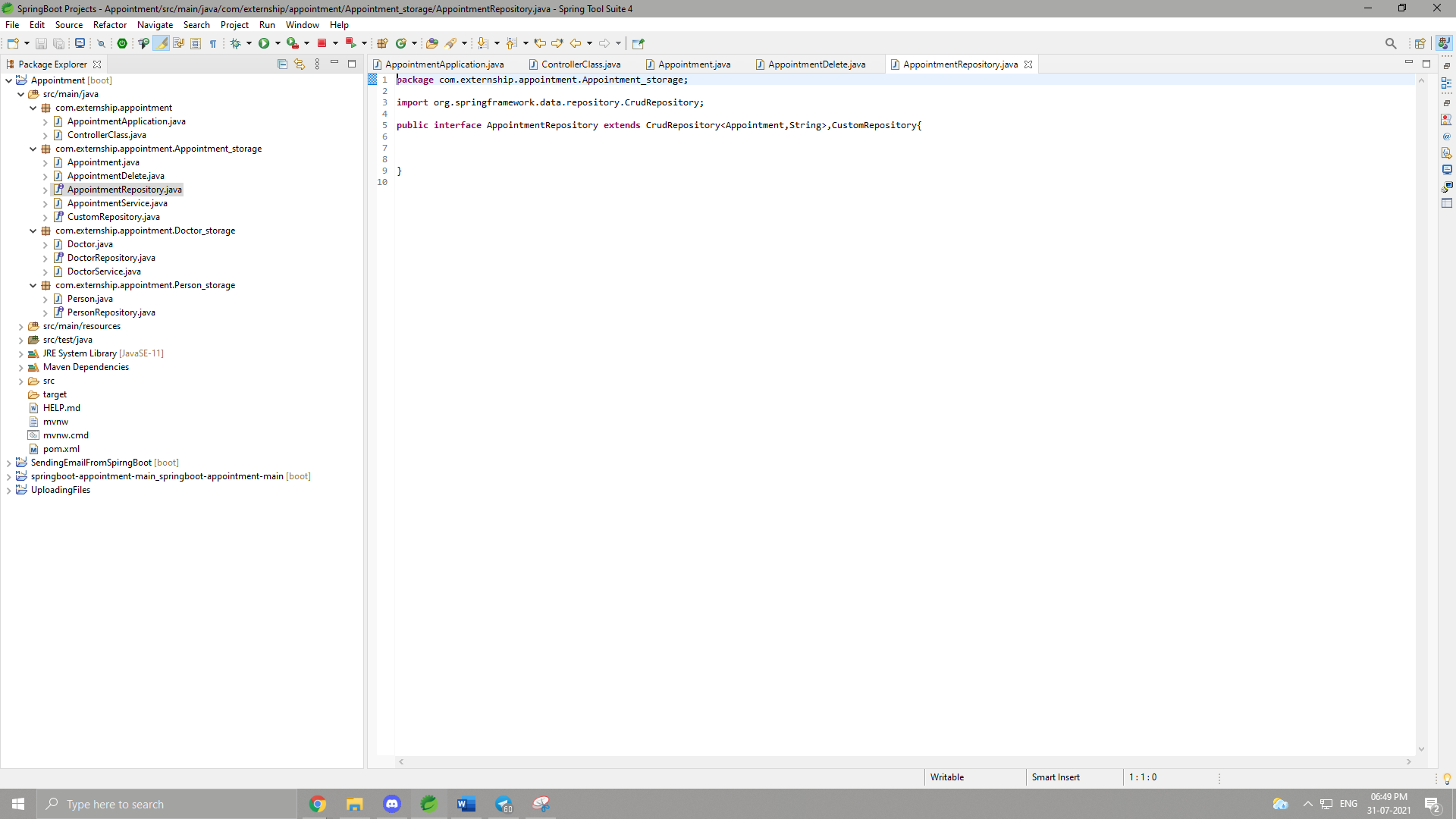
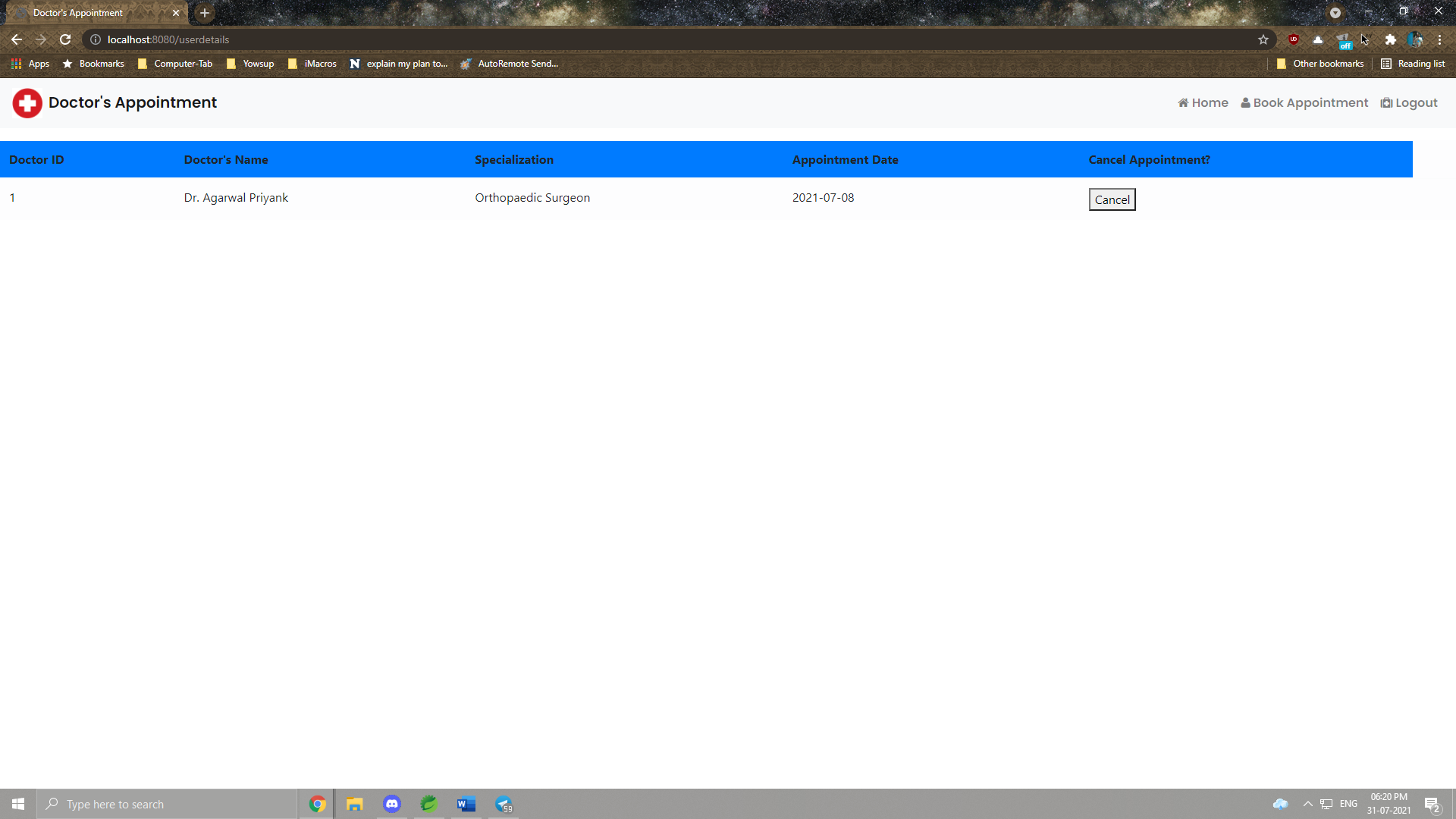


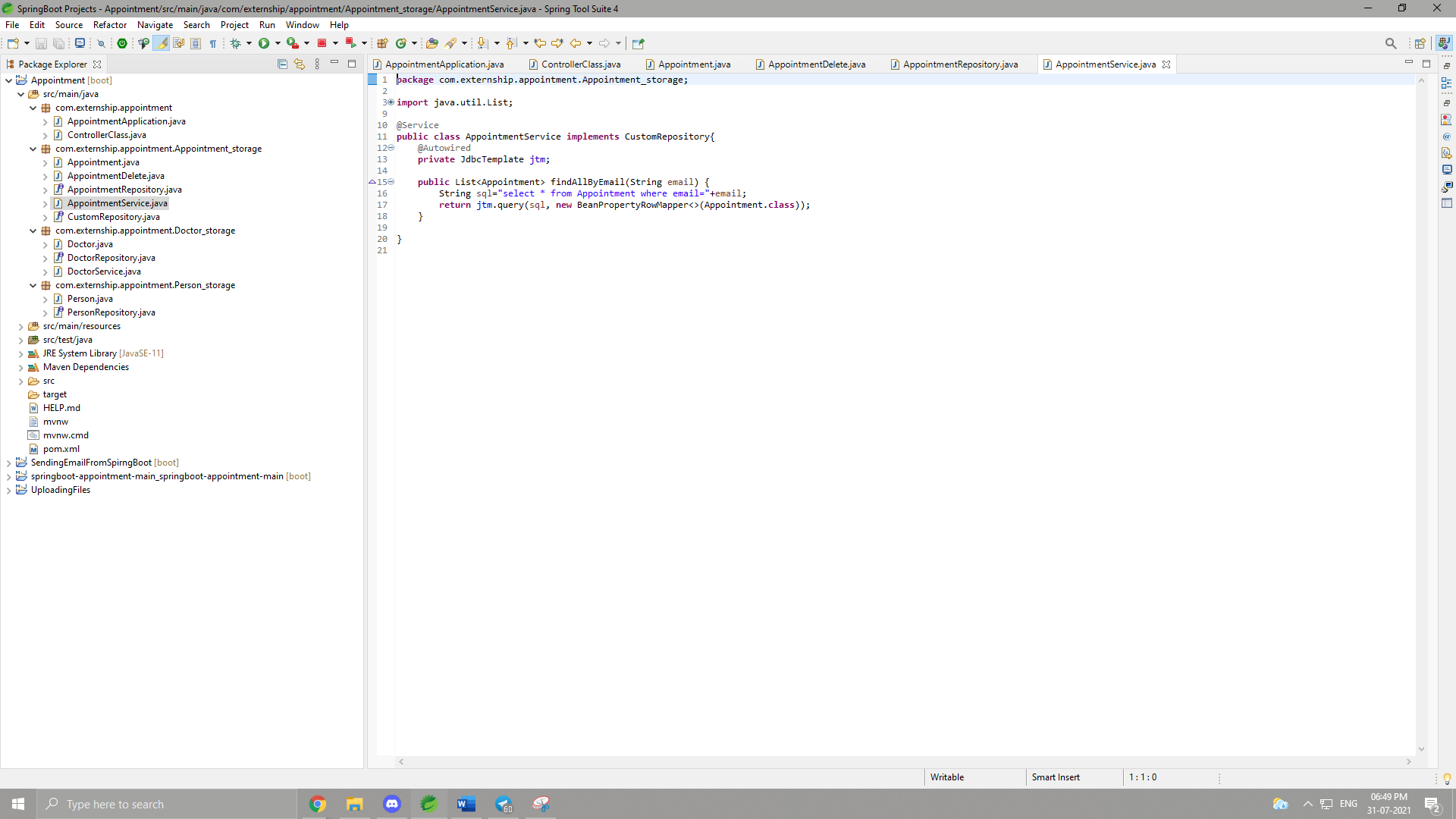
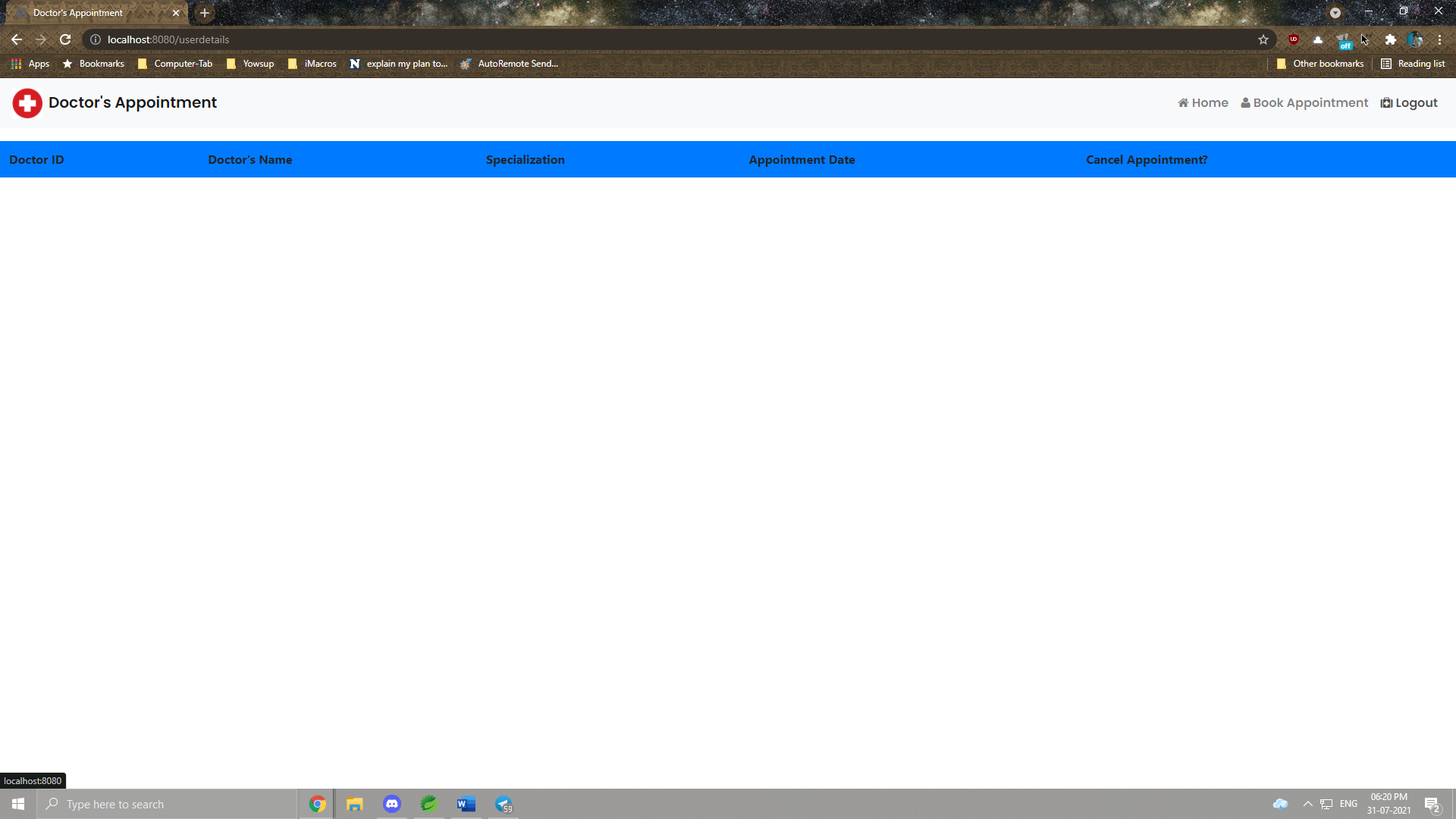


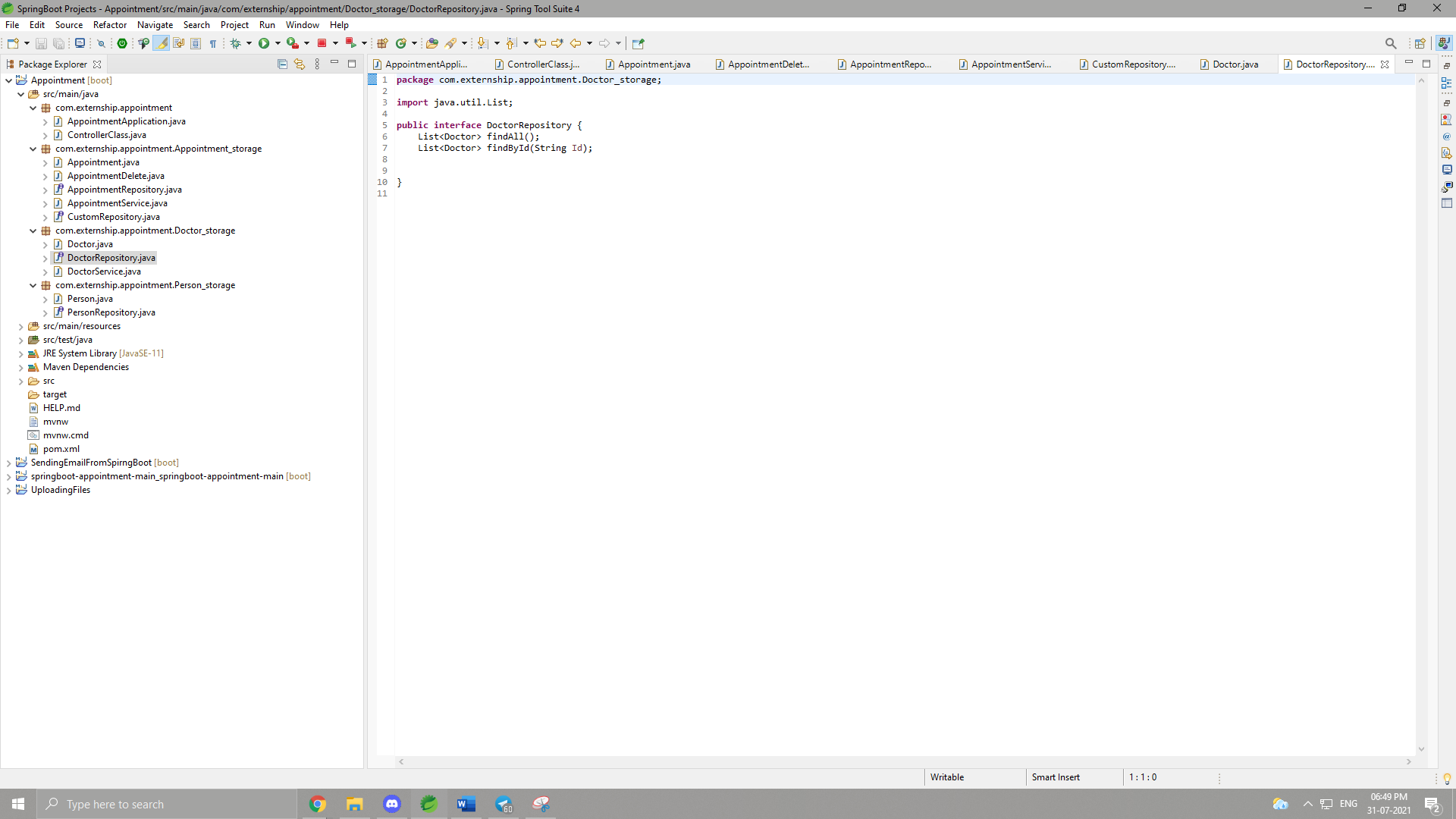
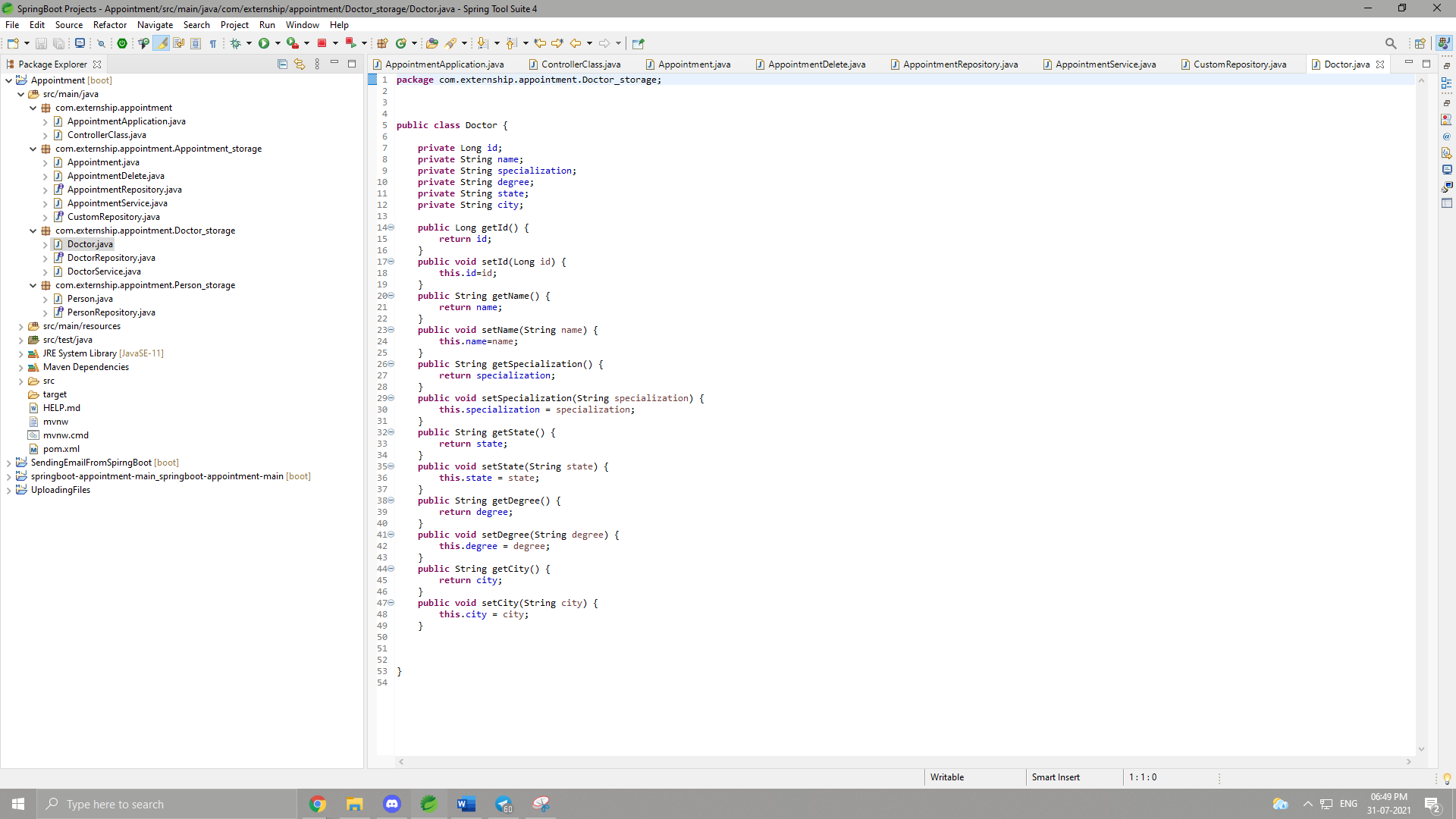
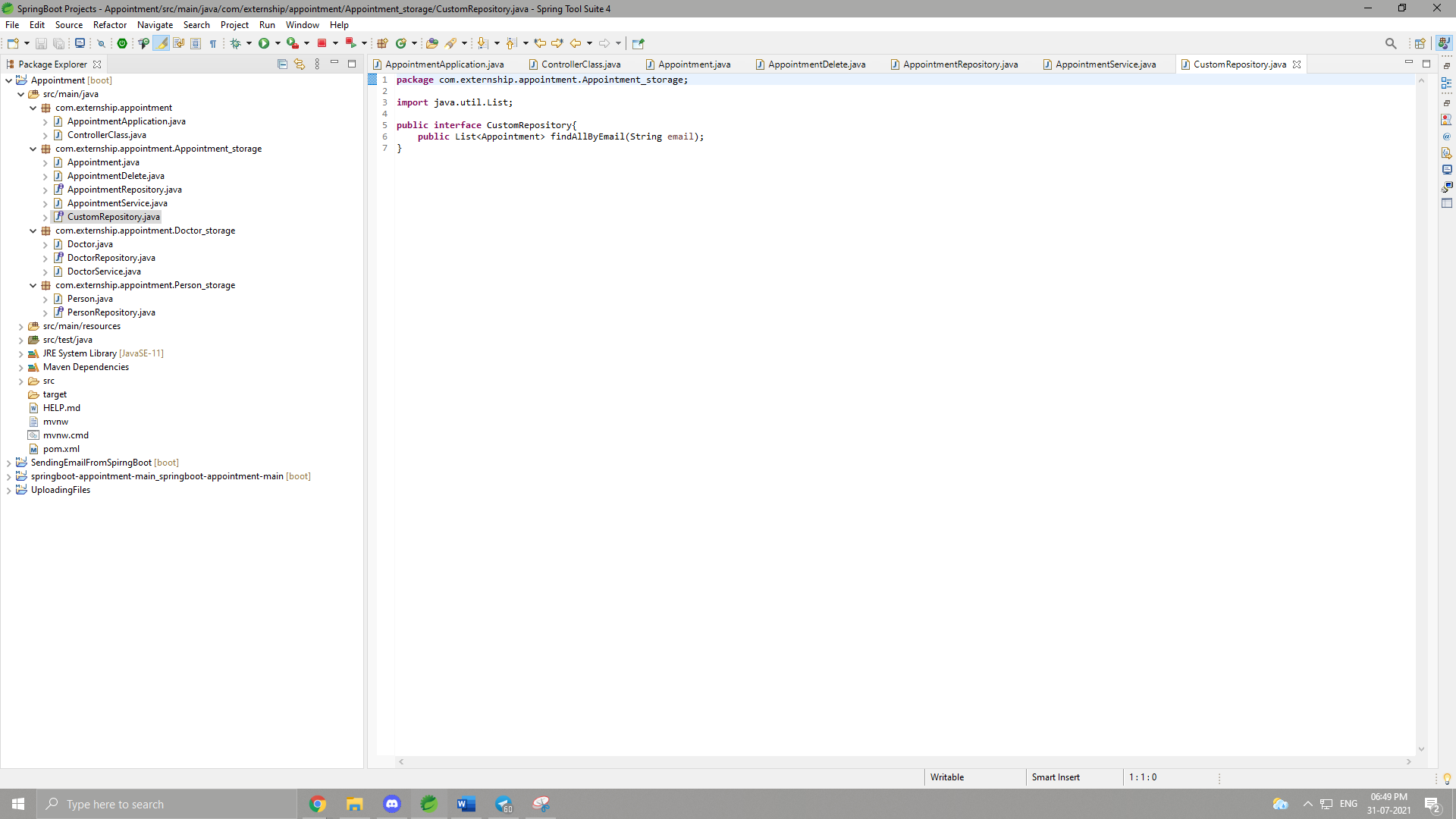
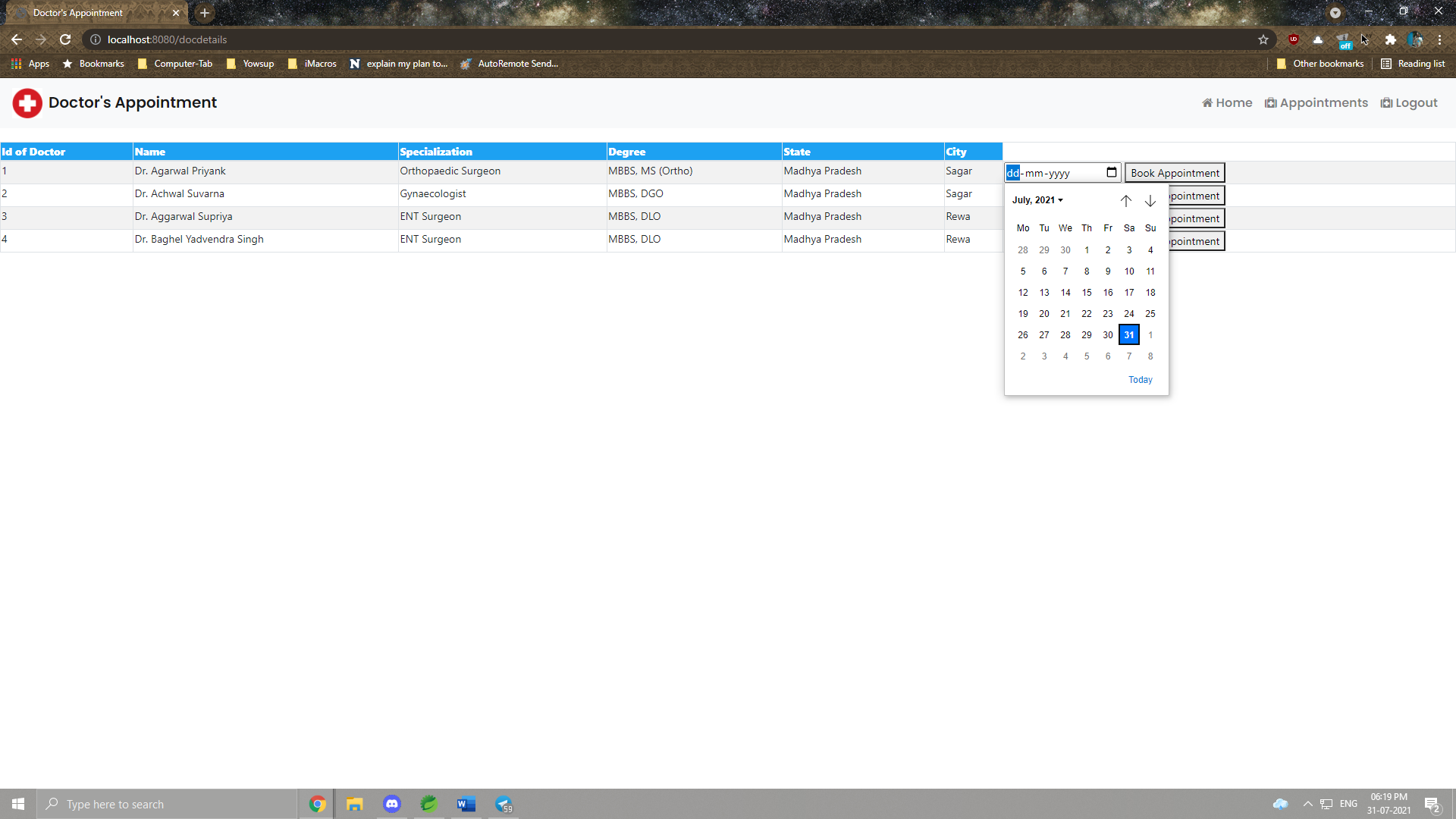


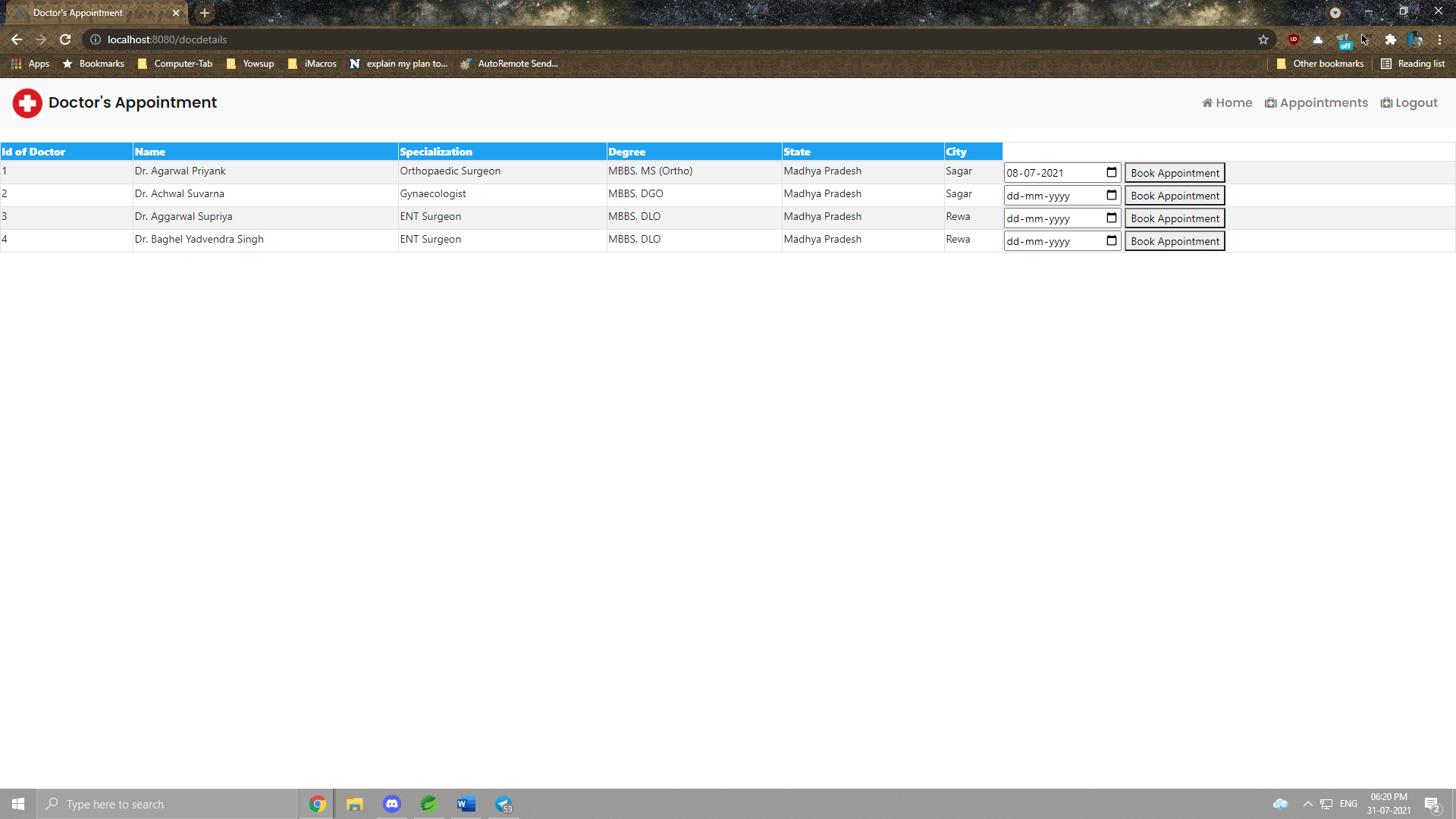


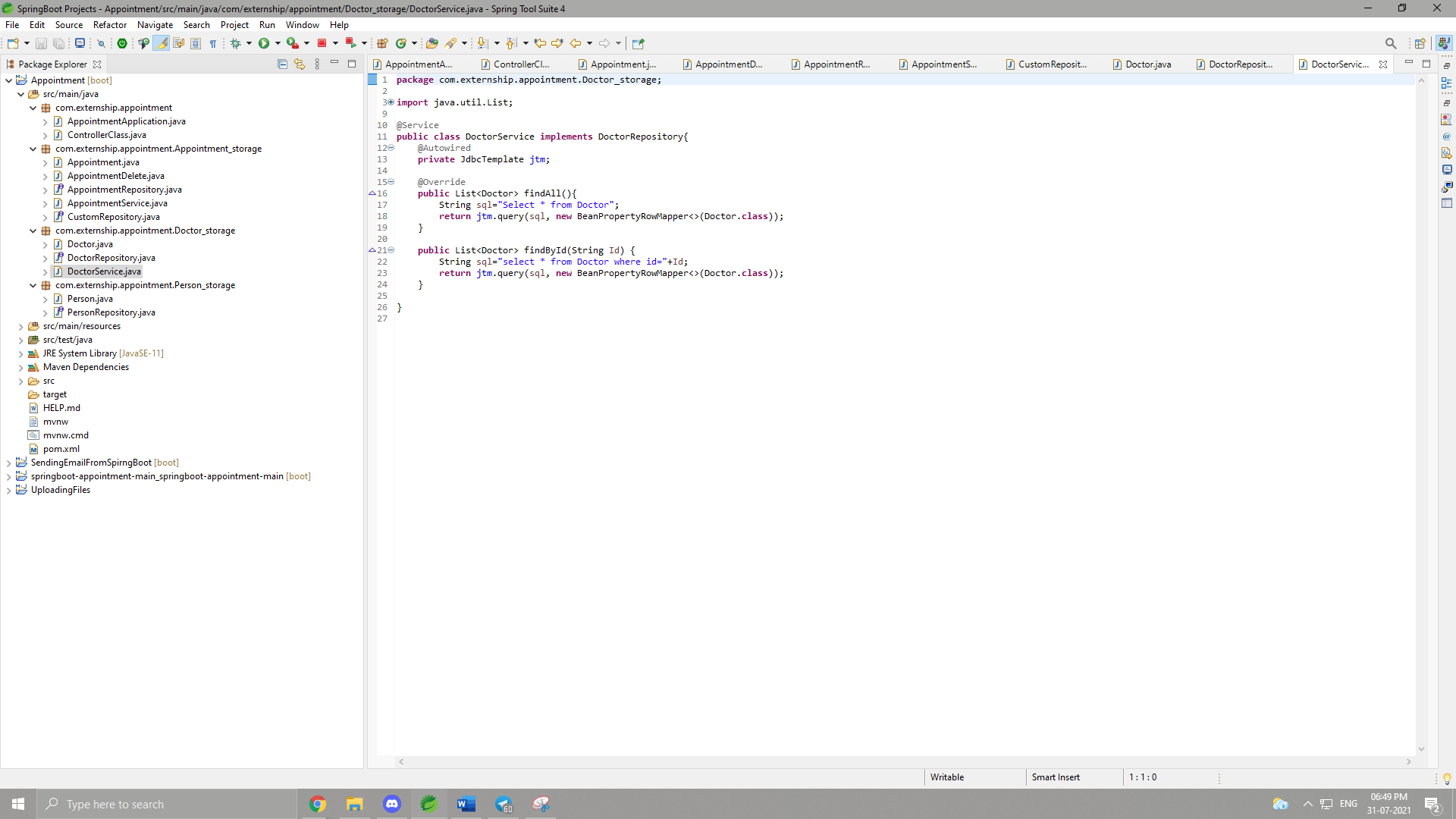
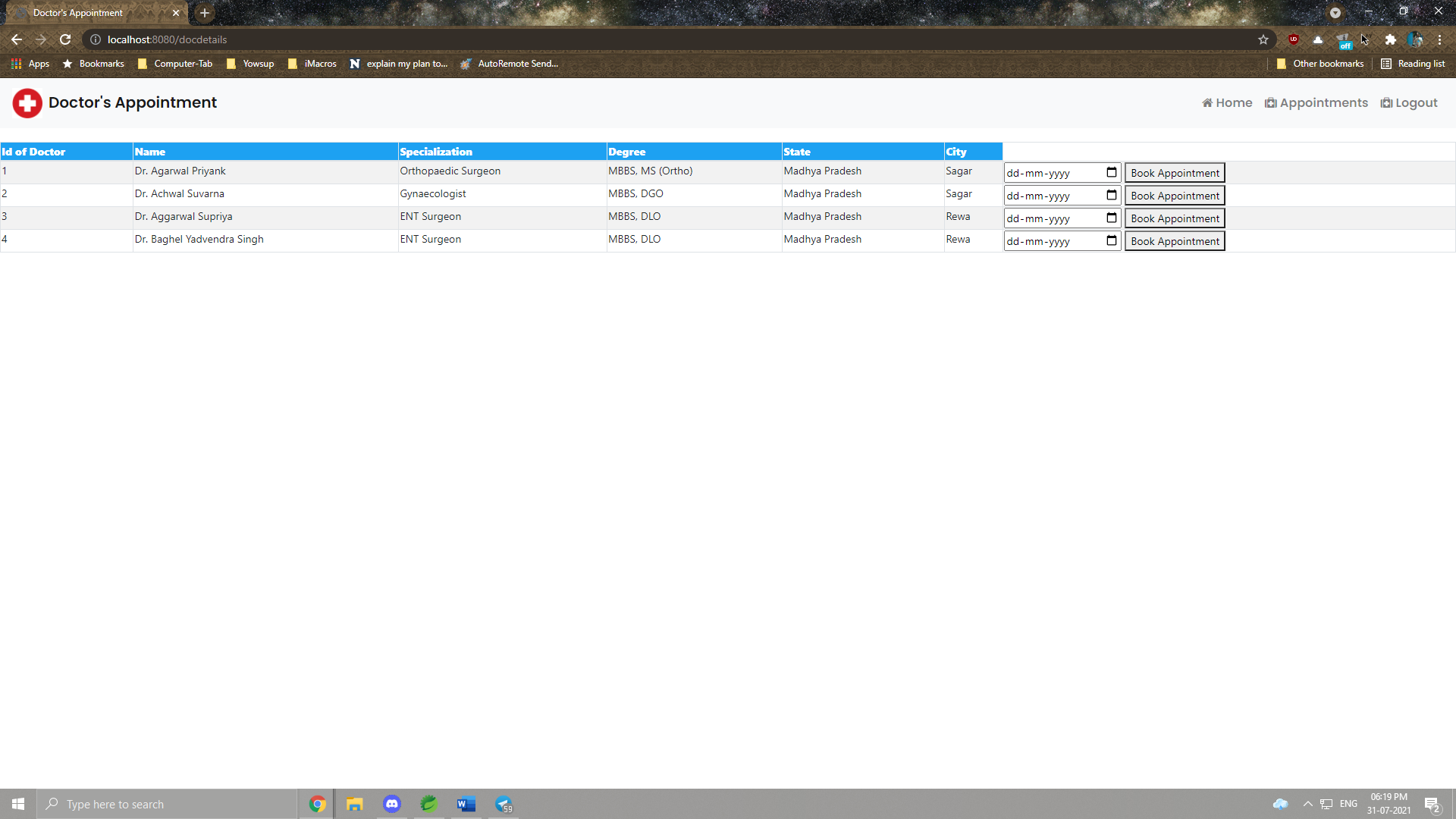


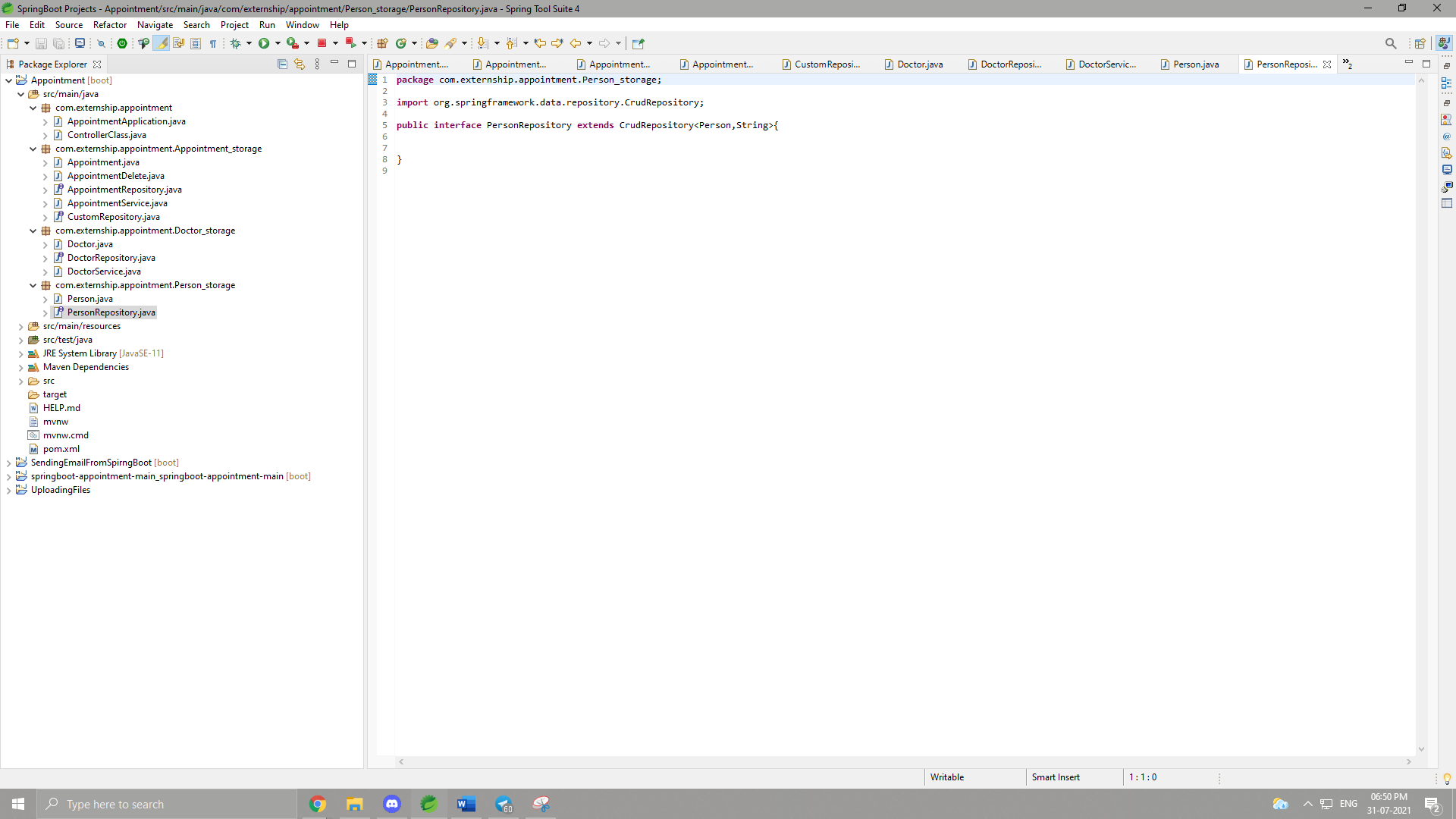
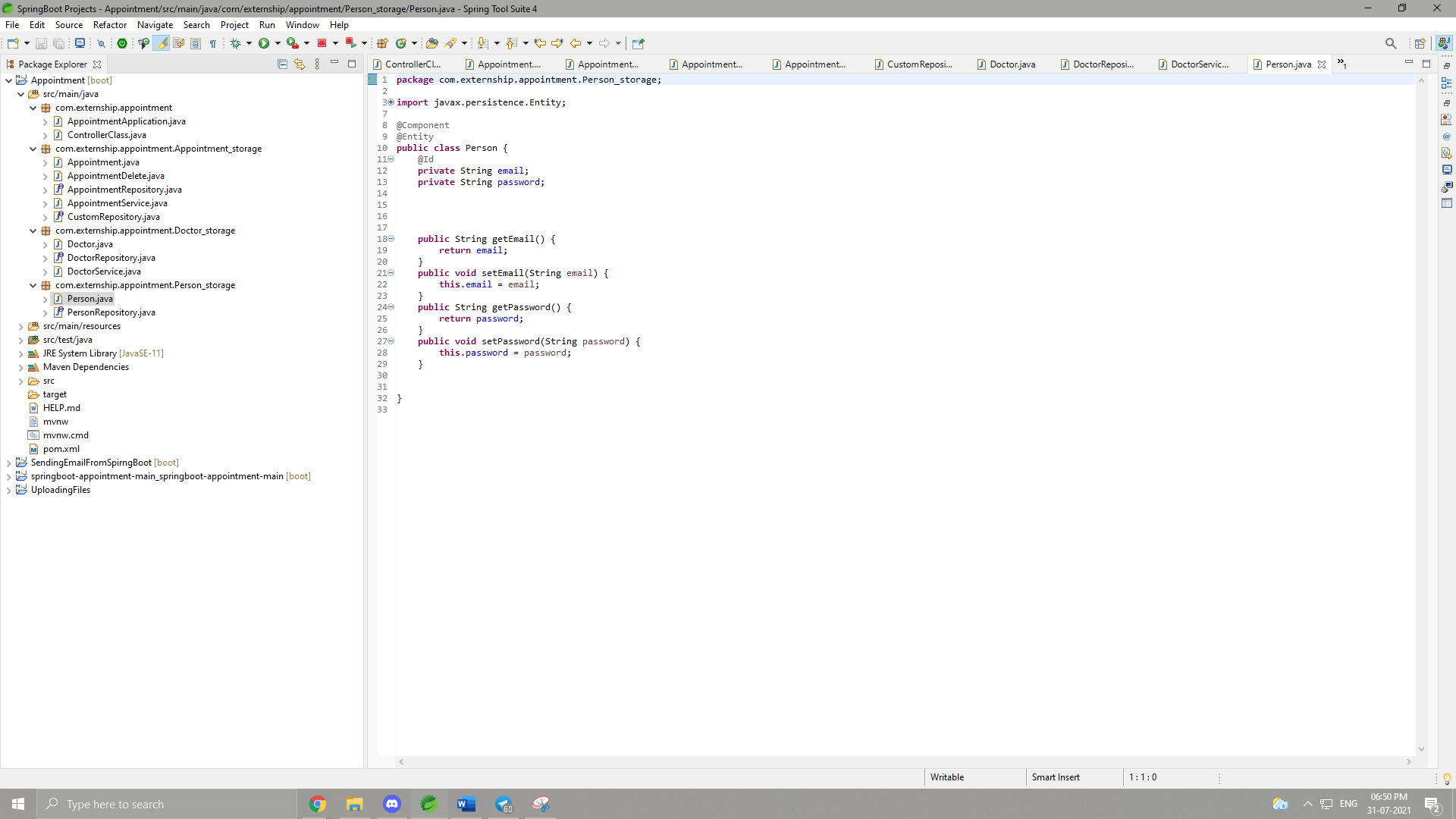
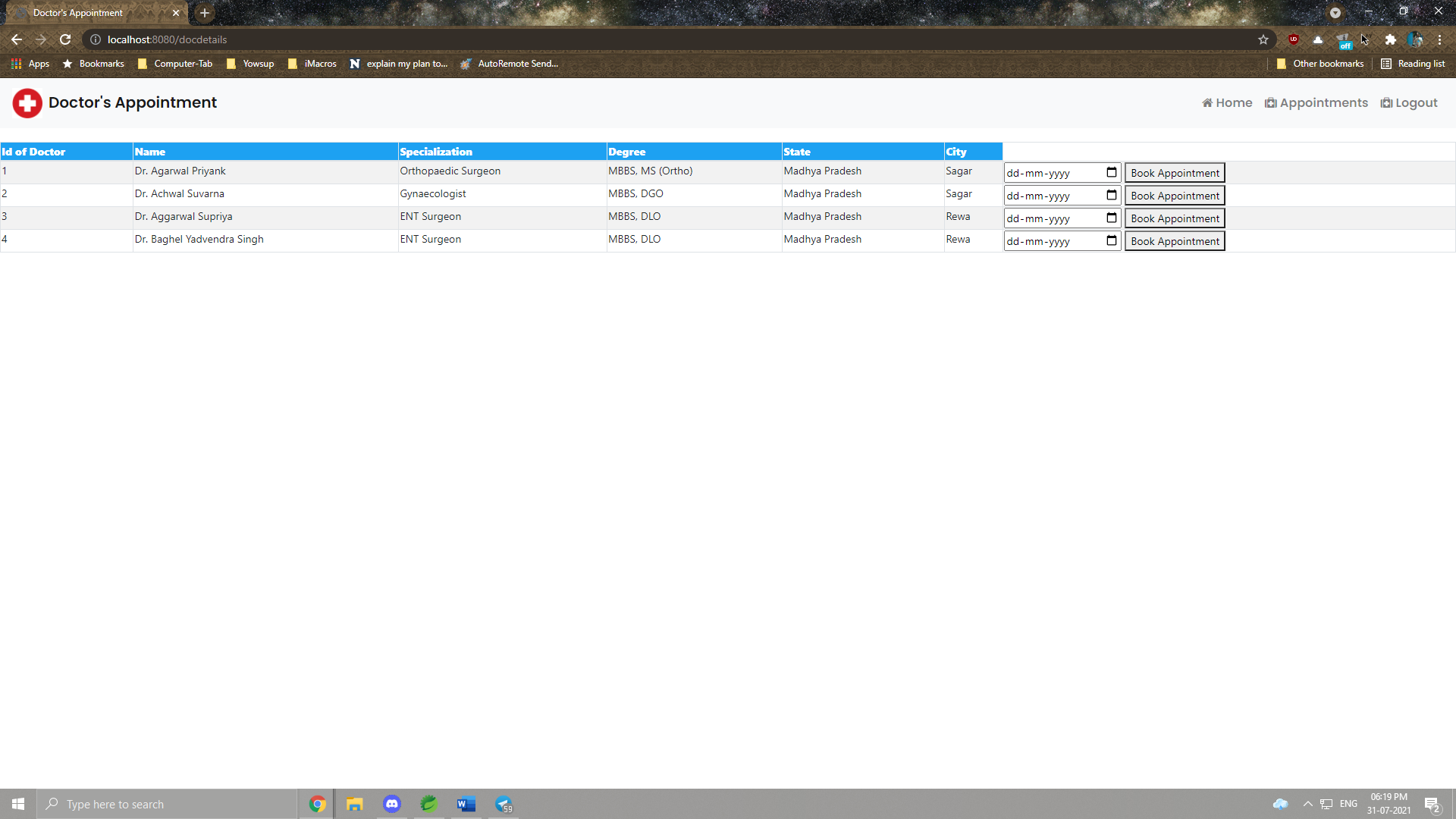


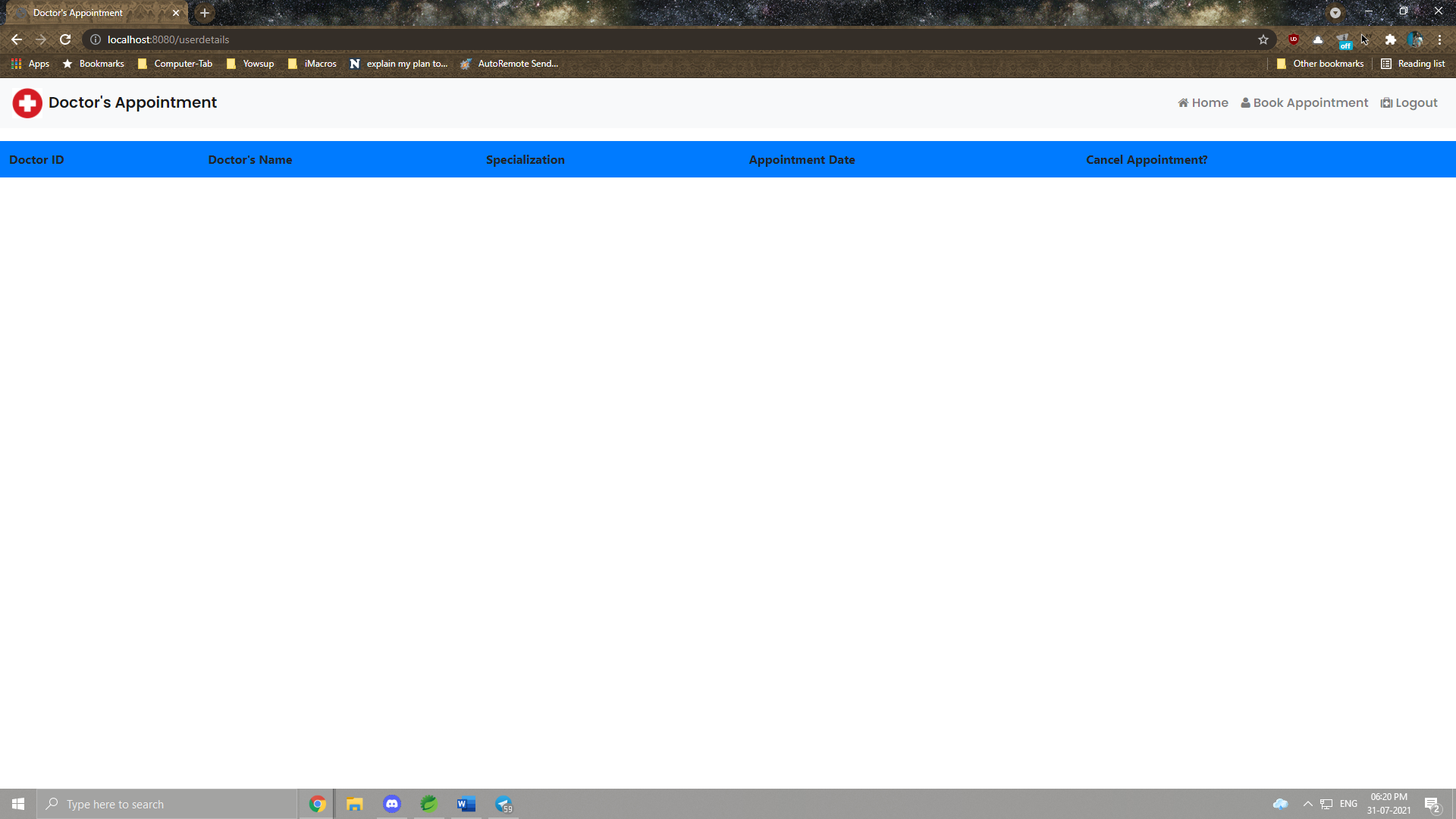












**UPDATES FOR THE EXISTING APP**

1. We can add multiple slots for each day that can show if the doctor is occupied for the day or not.

2. Verification system for the doctors to accept an appointment or not, based on the patient's details.

3. A way for doctors to register on this site through an online process.

4. We can add a map on the website for the patients to see the location of the hospital or clinic where the doctor sits.

5. We can also create an admin module for the website where the admin can add doctors and their details and view appointments as well.

**TECHNOLOGIES USED:**

1. Spring Boot : We have used Spring Boot as our application framework, i.e., we have created both frontend and backend part solely using Spring Boot, along with the testing and debugging of the webapp. Java has been used to develop the backend part and for frontend, HTML, CSS and Bootstrap has been used. We have used H2 database to store the data related to doctors, patients and their associated appointments.
2. HTML: For the frontend, HTML was used to develop the overall structure/skeleton of the webapp. We used various types of html tags such as <h1>, designating the main header or title of a page, <p>, designating a paragraph, <img>,tag representing images linked to the img folder, <div> to create divisions, and many other tags such as <ul>, <li>, <button>, <form>, <href>, <link>, etc.
3. CSS and Bootstrap: To beautify our html pages, we used these two components. Majority of CSS and Bootstrap was used to make navigation bar beautiful. Rest of it were used to get appropriate fonts, margin, padding, etc.
4. Java : Java was used as a backend language. It was used for developing controller, Doctor class, Person Class, Doctor and Person Interfaces. Java acted as a backbone for the whole logic of the webapp.

**GITHUB LINK :**

https://github.com/balatamoghna/springboot-appointment