ABSTRACT

The Virtual medical assistant is an online web platform to integrate various doctors, patients and testing laboratories. We named our Virtual Medical assistant as **VMEDICO.**

Our website Virtual medical assistant (VMEDICO) comes under the health care domain. So, our targeted end users would be like patients, doctors and testing laboratories**.**

VMedico platform will enable the patients to book appointments, preserve their prescriptions and lab-reports and also it allows patients to set a medication reminder. With our website the patients can book their appointments with various doctors across various hospitals same as with the testing laboratories. The doctors and testing laboratories can schedule their daily work by accepting the appointment requests from various patients across the city.

VMedico helps the doctors to track the patient’s medical history which helps to prescribe a good medicine for the patients.

This entire application was developed using the Spring boot. Spring boot is a part of a Spring frame work. The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform. Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

This VMedico will eliminate the idle waiting time in the hospital longue.

**INDEX**

**CONTENTS PAGE.NO**

List of figures

**CHAPTER 1**

**INTRODUCTION**

* 1. **INTRODUCTION & OBJECTIVE**

**CHAPTER 3**

**REQUIREMENTS SPECIFICATION**

**3.1 FUNCTIONAL REQUIREMENTS**

This application consists following modules.

1. Patient

2. Doctor

3. Testing Laboratory

* **Patient:** Each patient should have a facility to login and can book an appointment with various doctors and testing labs across the city. All the prescriptions and lab reports to be persisted. Website should send the email reminder on the day of appointment and medication reminders.
* **Doctor:** Doctor should have a facility to accept or reject the appointment request based on his/her convenience. Doctor can able to update their profile. Doctor can track the entire patient medical history.
* **Testing Laboratory:**  Testing Laboratory should have a facility to accept or reject the appointment request based on their convenience. Testing Laboratory can able to update their profile.

**3.2 NON-FUCTIONAL REQUIREMENTS**

**3.2.1** **USER-FRIENDLY**

The website has to be user-friendly. Easy to understand to perform various operations like uploading a prescription, profile updating, checking the status of current appointment, view the history of appointments etc.

**3.2.2 RELIABILITY**

The website should be reliable to use. The appointment status has to be updated based on the user interactions and it has to be persisted.

**3.2.3 SECURITY**

The web server and database server should be protected from hacking, virus etc.

**3.2.4 AVAILABILITY**

This software will be available always.

**3.3 PERFORMANCE REQUIREMENTS**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, while will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a System, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly sated as given below:

* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system
* The existing system is completely dependent on the user to perform all the duties.

**3.4 HARDWARE REQUIREMENTS**

Processer : Any Update Processer

Ram : Min 1 GB

Hard Disk : Min 100 GB

**3.5 SOFTWARE REQUIREMENTS**

Operating System : Windows family

Technology : Java (1.7/1.8)

Web Technologies : Html, Html-5, JavaScript, CSS, JSP, Spring boot

Web Server : Tomcat 7/8

Database : MySQL

**CHAPTER 4**

**SYSTEM ANALYSIS**

**4.1 STUDY OF THE SYSTEM**

* To conduct studies and analyses of an operational and technological nature and
* To promote the exchange and development of methods and tools for operational analysis as applied to defence problems.

**4.1.1 LOGICAL DESIGN**

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems design are included. Logical design includes ER Diagrams i.e., Entity Relationship Diagrams.

**4.1.2 PHYSICAL DESIGN**

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified / authenticated, how it is processed, and how it is displayed as output. In Physical design, following requirements about the system are decided.

1. Input requirement,
2. Output requirements,
3. Storage requirements,
4. Processing requirements,
5. System control and backup or recovery.

Put another way, the physical portion of systems design can generally be broken down into three sub-tasks:

**1.User Interface Design:** User Interface Design is concerned with how users add information to the system and with how the system presents information back to them.

**2.Data Design:** Data Design is concerned with how the data is represented and stored within the system.

**3.Process Design** data: Process Design is concerned with how over through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the systems design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase.

**4.2 FEASIBILITY STUDY**

Feasibility study is conducted once the problem is clearly understood. The feasibility study which is a high-level capsule version of the entire system analysis and design process. The objective is to determine whether the proposed system is feasible or not and it helps us to the minimum expense of how to solve the problem and to determine, if the Problem is worth solving. The following are the three important tests that have been carried out for feasibility study.

**4.2.1 TECHNICAL FEASIBILITY**

In the technical feasibility study, one has to test whether the proposed system can be developed using existing technology or not. It is planned to implement the proposed system in JSP and Spring boot. The project entitled is technically feasible because of the following reasons.

* All necessary technology exists to develop the system.
* The existing system is so flexible that it can be developed further.

**4.2.2 ECONOMIC FEASIBILITY**

As a part of this, the costs and benefits associated with the proposed systems are to be compared. The project is economically feasible only if tangible and intangible benefits outweigh the cost. We can say the proposed system is feasible based on the following grounds.

* The cost of developing the full system is reasonable.
* The cost of hardware and software for the application is less.

**4.2.3 OPERABILITY FEASIBILITY**

The project is operationally feasible because there is sufficient support from the project management and the users of the proposed system. Proposed system definitely does not harm and will not produce the bad results and no problem will arise after implementation of the system.

**4.3 INPUT & OUTPUT REPRESENTATION**

**4.3.1 INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**4.3.2 OBJECTIVES**

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

**4.3.3 Output Design**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

* Designing computer output should proceed in an organized manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
  1. Select methods for presenting information.
  2. Create document, report, or other formats that contain information produced by the system.

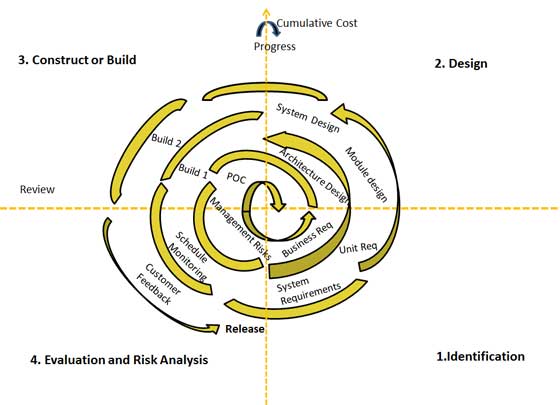
The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**4.4 PROCESS MODEL USED WITH JUSTIFICATION**

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**SDLC (Spiral Model):**

****

Figno:4.4 Software Development Life Cycle

**Stages of SDLC:**

Requirement Gathering and Analysis

* Designing
* Coding
* Testing
* Deployment

**CHAPTER 5**

**SYSTEM DESIGN**

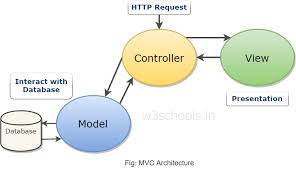
**INTRODUCTION**

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

**SYSTEM ARCHITECTURE**

**MVC Architecture:**

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects.



**Model**: The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data.

**View**: The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

**Controller**: Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

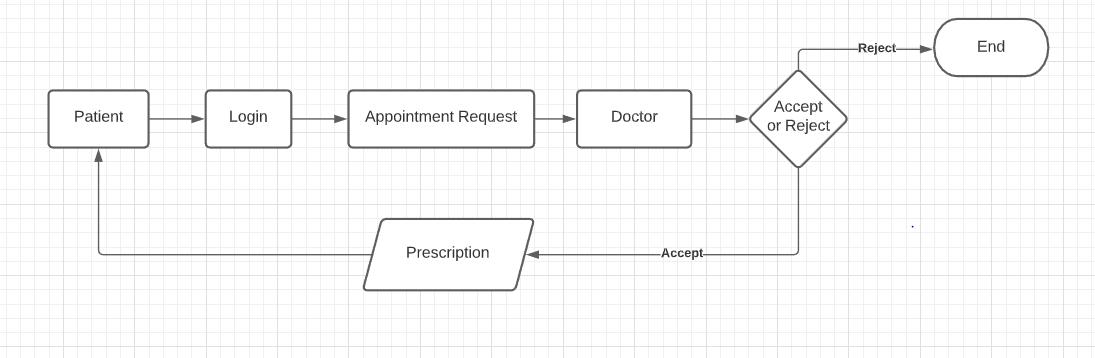
**Advantages** **of MVC architecture are**:

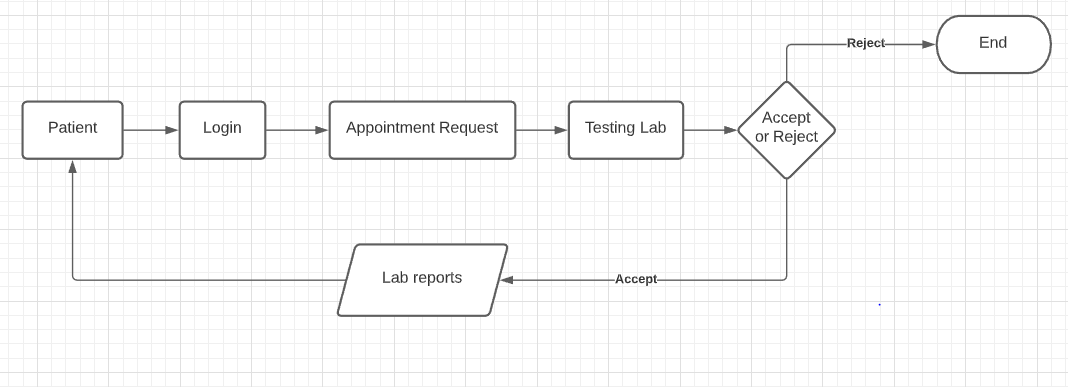
* Faster Web Application Development Process
* MVC Web Application Supports Asynchronous Technique
* Offers the Multiple Views
* Ideal for developing large size web application
* MVC Model Returns the Data Without the Need of Formatting
* The Modification Never Affects the Entire Model

**DATA FLOW DIAGRAM**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system, modelling its process aspects.

A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

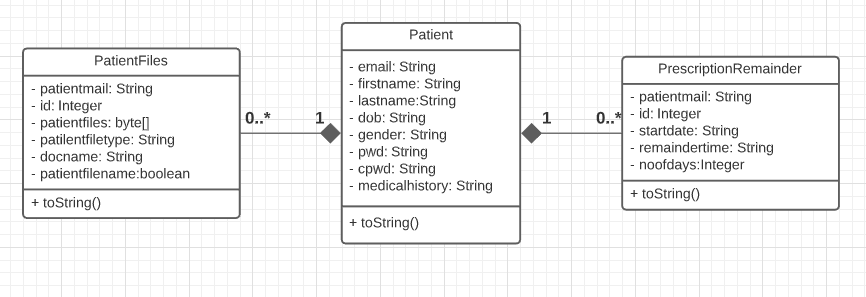


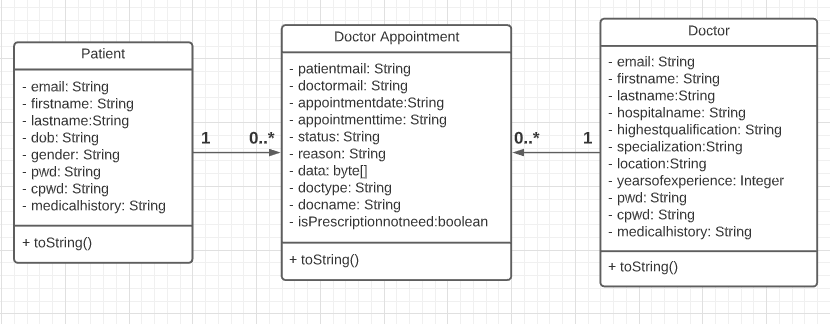


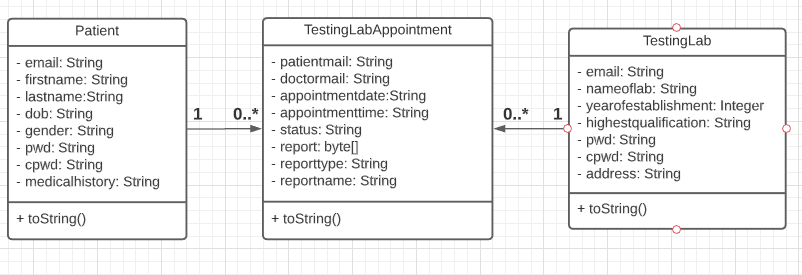
Class diagrams:

A class is a set of objects that share a common structure and common behaviour (the same attributes, operations, relationships and semantics). A class is an abstraction of real-world items.

Our Project Vmedico has 3 main targeted end users. They are Patients, Doctors and Testing laboratories. We created each class for each user and various other classes which performs the required business operations.







USE CASE diagram:

Actor:

Actor represents the role a user plays with respect to the system. An actor interacts with, but has no control over the use cases.

The various actors i.e., users are Patients, Doctors and Testing Laboratories.

Users or actors mainly are of two types. They are primary users and secondary users. In our case Patients are the primary users where as Doctors and Testing Labs are the secondary users.

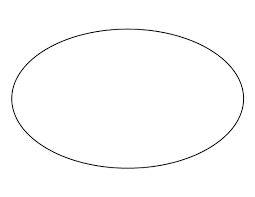
An actor is someone or something that:

* with or uses the system.
* Provides input to and receives information from the system.
* Is external to the system and has no control over the Interacts use cases.

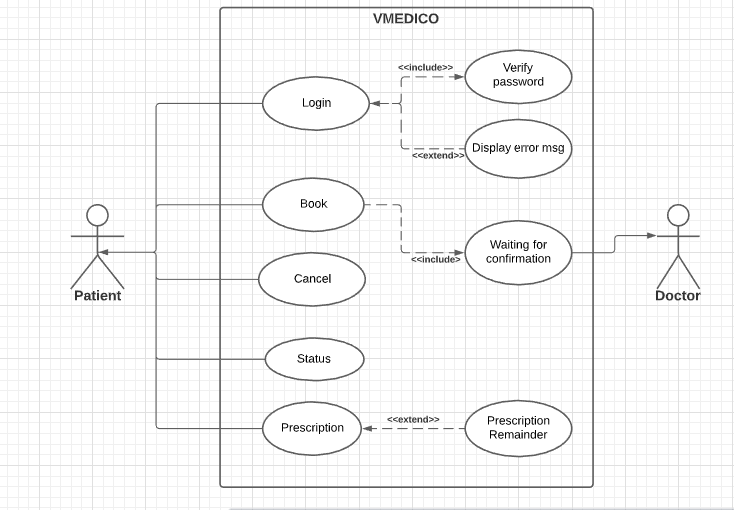
Use case:

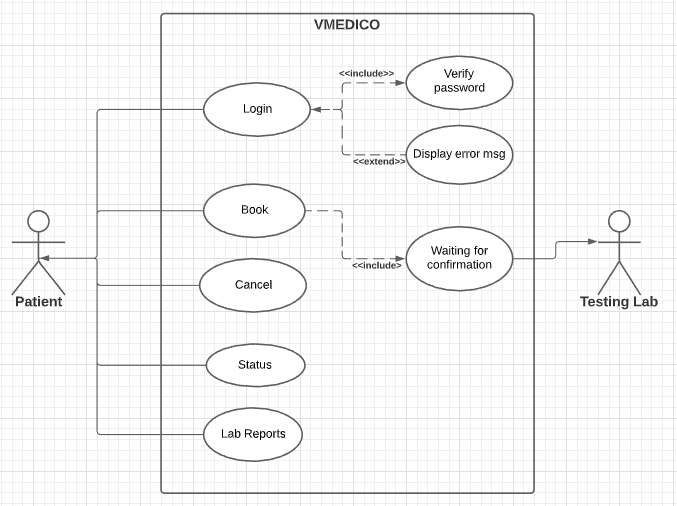
A use case can be described as a specific way of using the system from a user’s (actor’s) perspective.

Graphical Representation:



Use case

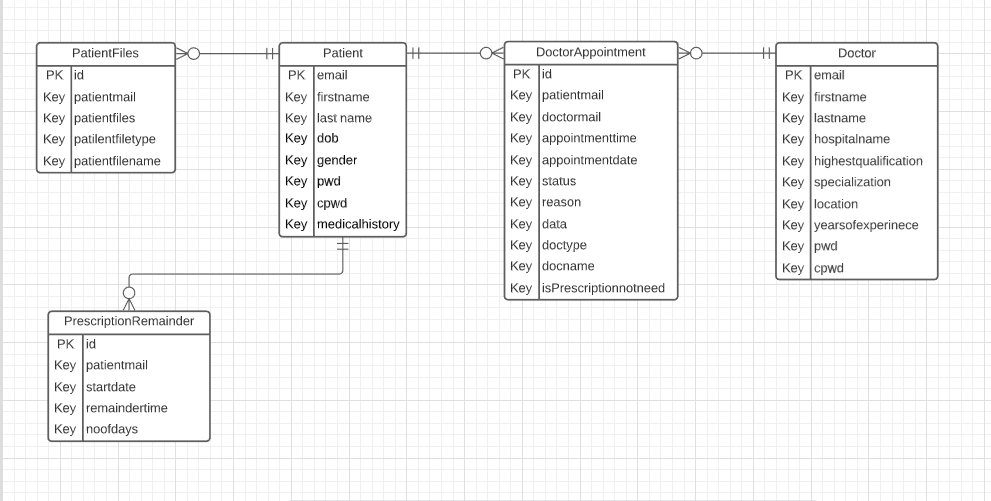


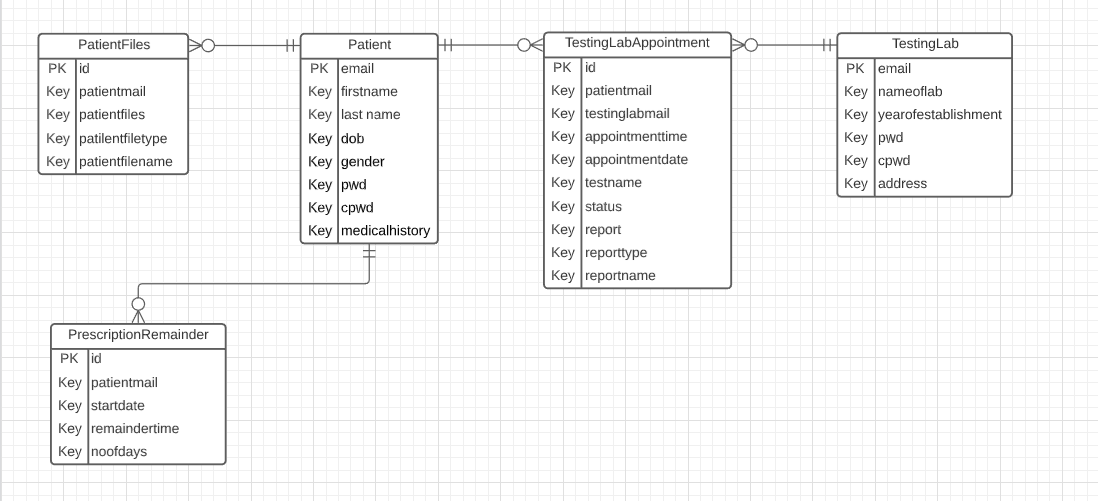


Entity relationship diagram:

The entity relationship diagram provides an overview how the data bases are related to each other. It shows the cardinal relations between the entities i.e., tables.

Here, **PK** represents the **Primary** **key** of an entity.





IMPLEMENTATION DETAILS

MODULES:

**PATIENT MODULE:**

The Patient Module allow the patients to login to the website if they are an existing user otherwise user can create an account. After logging in, the Patients have a facility to book an appointment with the doctors as well as the testing laboratories. They can cancel the appointment in case if they change their mind. The patients have a facility to add a medication reminderand they can view their prescriptions and lab reports on their dashboard.

DOCTOR MODULE:

The Doctor Module allow the doctors to login to the website if they are an existing user otherwise user can create an account. After logging in, the doctors can able to access the appointment request placed by the patients. Based on his/her schedule he can accept or reject the appointment request. The doctors can view the patient’s entire medical history.

TESTING LABORATORY MODULE:

The Testing Laboratory Module allow the test labs to login to the website if they are an existing user otherwise user can create an account. After logging in, the test labs can able to access the appointment request placed by the patients. Based on their schedule they can accept or reject the appointment request.

**JAVA**

The **Java virtual machine** (JVM) is a software implementation of a computer that executes programs like a real machine. The Java virtual machine is written specifically for a specific operating system, e.g., for Linux a special implementation is required as well as for Windows.

For developing the VMedico we have used Java 1.8 version. Java programs are compiled by the Java compiler into so-called byte code. The Java virtual machine interprets this byte code and executes the Java program. Java is case sensitive, e.g., the variables my Value and my value will be treated as different variables.

**SQL: Structured Query Language**

Although SQL is an ANSI (American National Standards Institute) standard, there are different versions of the SQL language.

For developing the VMedico we have used MySQL. However, to be compliant with the ANSI standard, they all support at least the major commands (such as SELECT, UPDATE, DELETE, INSERT, WHERE) in a similar manner.

**JAVA SERVER PAGES**

JSP technology enables you to mix regular static html with dynamically generated content from servlets. Separating the static html from the dynamic content provides a number of benefits over servlets alone.

**SPRING BOOT**

Spring Boot is an open-source Java-based framework used to create a micro-Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications.

Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade spring application that you can **just run**. You can get started with minimum configurations without the need for an entire Spring configuration setup.

The Advantages of Spring boot are:

* Easy to understand and develop spring applications
* Increases productivity
* Reduces the development time

**TESTING**

**7.1 INTRODUCTION**

Testing is the debugging program is one of the most critical aspects of the computer programming triggers, without programming that works, the system would never produce an output of which it was designed. Testing is best performed when user development is asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quantity but quality of the data used the matters of testing.

**7.1.1 TESTING OBJECTIVES:**

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, testing is a process of executing a program with intent of finding an error.

* A successful test is one that uncovers an as yet undiscovered error.
* A good test case is one that has probability of finding an error, if it exists.
* The test is inadequate to detect possibly present errors.
* The software more or less confirms to the quality and reliable standards.

**7.2 STRATEGIC APPROACH TO SOFTWARE TESTING**

**7.2.1 LEVELS OF TESTING:**

In order to uncover present in different phases we have the concept of levels of testing.

**THE BASIC LEVELS OF TESTING:**

Client needs acceptance testing

Requirements system testing

Design integration testing

Code unit testing

Figure 7.2.1: Levels of Testing

**7.2.2 CODE TESTING:**

This examines the logic of the program. For example, the logic for updating various sample data and with the sample files and directories were tested and verified.

**7.2.3 UNIT TESTING:**

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during programming stage itself. In the testing step each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks for fields also. For example, the validation check is done for varying the user input given by the user which validity of the data entered. It is very easy to find error debut the system.

**7.2.4 SYSTEM TESTING:**

Once the individual module testing is completed, modules are assembled and integrated to perform as a system. The top-down testing, which began from upper-level to lower-level module, was carried out to check whether the entire system is performing satisfactorily.

There are three main kinds of System testing:

**7.2.5 ALPHA TESTING:**

This refers to the system testing that is carried out by the test team with the Organization.

**7.2.6 BETA TESTING**:

This refers to the system testing that is performed by a selected group of friendly customers

**7.2.7 ACCEPTANCE TESTING:**

This refers to the system testing that is performed by the customer to determine whether or not to accept the delivery of the system.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Condition ID** | **Description coverage** | **Expected results** | **Covered by script** |
| **1.** | Verification of an email notification | If an appointment is accepted or rejected the patient will get an acknowledgment email. | This is covered by using a dependency called Java Mail Service. |
| 2. | Uploading a file (prescription or lab reports) | Blob data should be updated in the database. | This type of test is covered in JSP. Validations were covered on the size of the file also. |
| 3. | Validity of login | Only the authorized persons must access system. | This is covered in the login procedure for the validity of a user |

Table 7.2.7: Acceptance Testing

**7.2.8 INTEGRATION TESTING:**

Data can be lost across an interface, one module can have an adverse effort on the other sub functions, when combined, may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. The testing was done with sample data. The developed system has run successfully for this sample data. The need for integrated test is to find the overall system performance.

**7.2.9 OUTPUT TESTING:**

After performance of the validation testing, the next step is output testing. The output displayed or generated by the system under consideration is tested by asking the user about the format required by system. The output format on the screen is found to be correct as format was designed in the system phase according to the user needs. Hence the output testing does not result in any correction in the system.

**7.2.10 VALIDATION:**

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed

**7.3 TEST CASES**

**7.3.1 TEST PLAN:**

The test-plan is basically a list of test cases that need to be run on the system. Some of the test cases can be run independently for some components (report generation from the database, for example, can be tested independently) and some of the test cases require the whole system to be ready for their execution. It is better to test each component as and when it is ready before integrating the components. It is important to note that the test cases cover all the aspects of the system (i.e., all the requirements stated in the RS document).

**Test Plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sn** | **Testcase Title** | **Description** | **Expected Outcome** | **The requirement in RS that is being tested** | **Result** |
| 1 | Successful User Verification | The login to the system should be tried with the login assigned by the admin and the correct password | Login should be successful and the user should enter in to the system | RS1 | Passed |
| 2 | Unsuccessful User Verification due to wrong password | Login to the system with a wrong password | Login should fail with an error ‘Invalid Password’ | RS1 | Passed |
| 3 | Unsuccessful User Verification due to invalid login id | Login to the system with a invalid login id | Login should fail with an error ‘Invalid user id’ | RS1 | Passed |

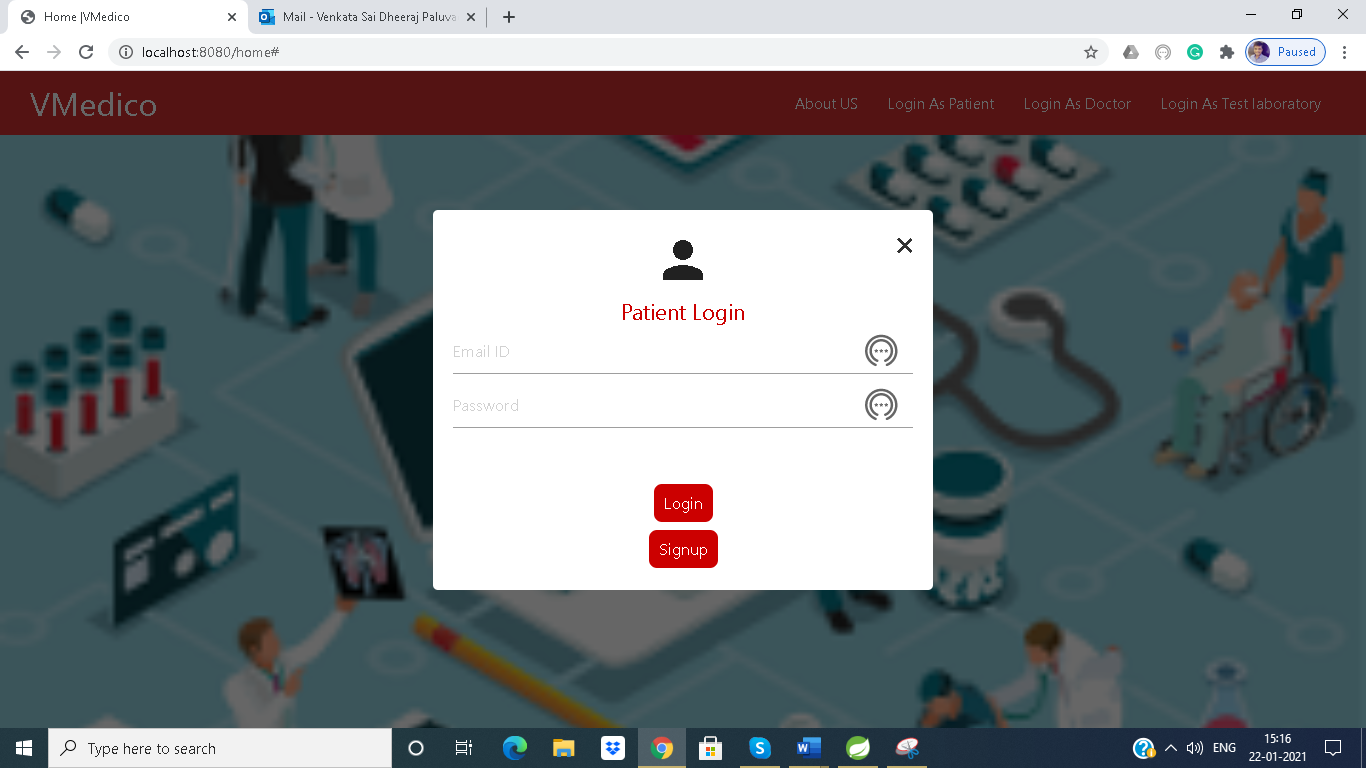
Table:7.3.1 Test Plan

**8.SCREENSHOTS**

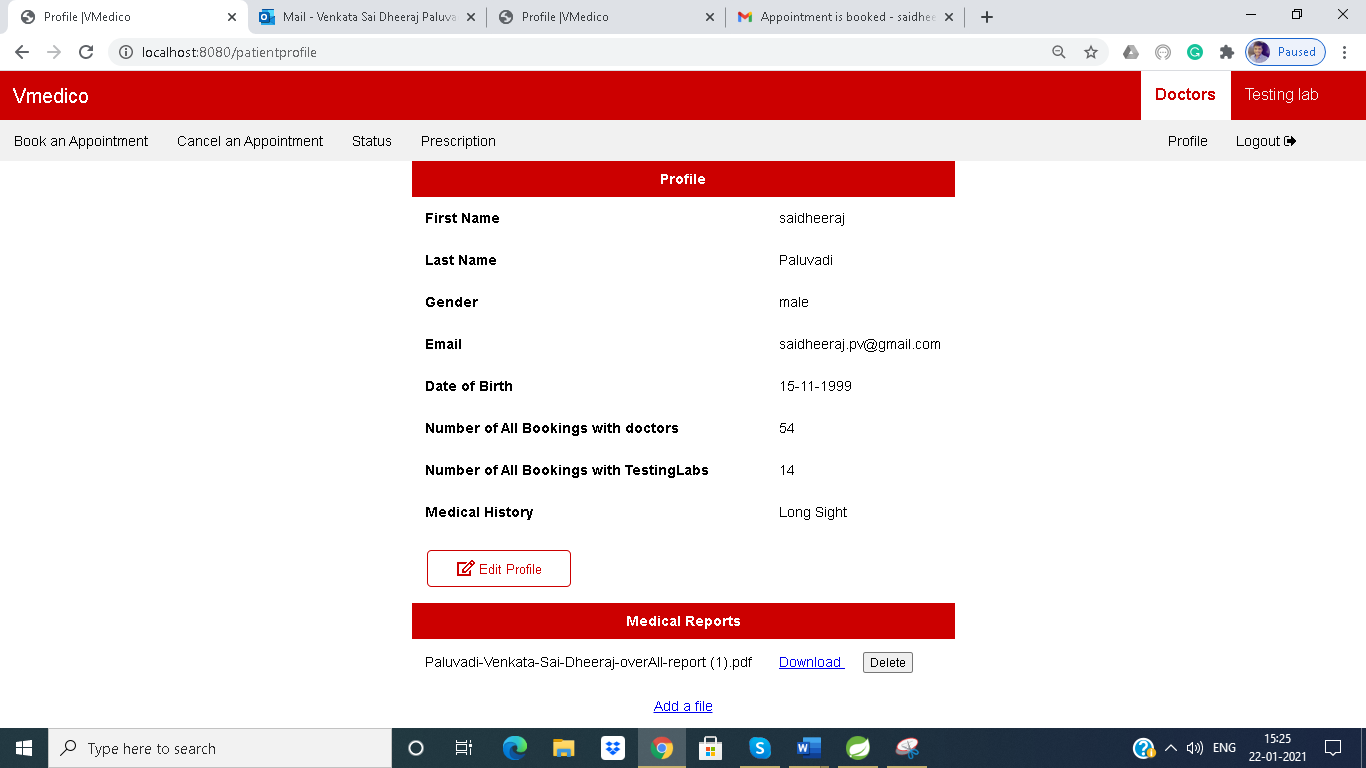


**Landing page of our website**

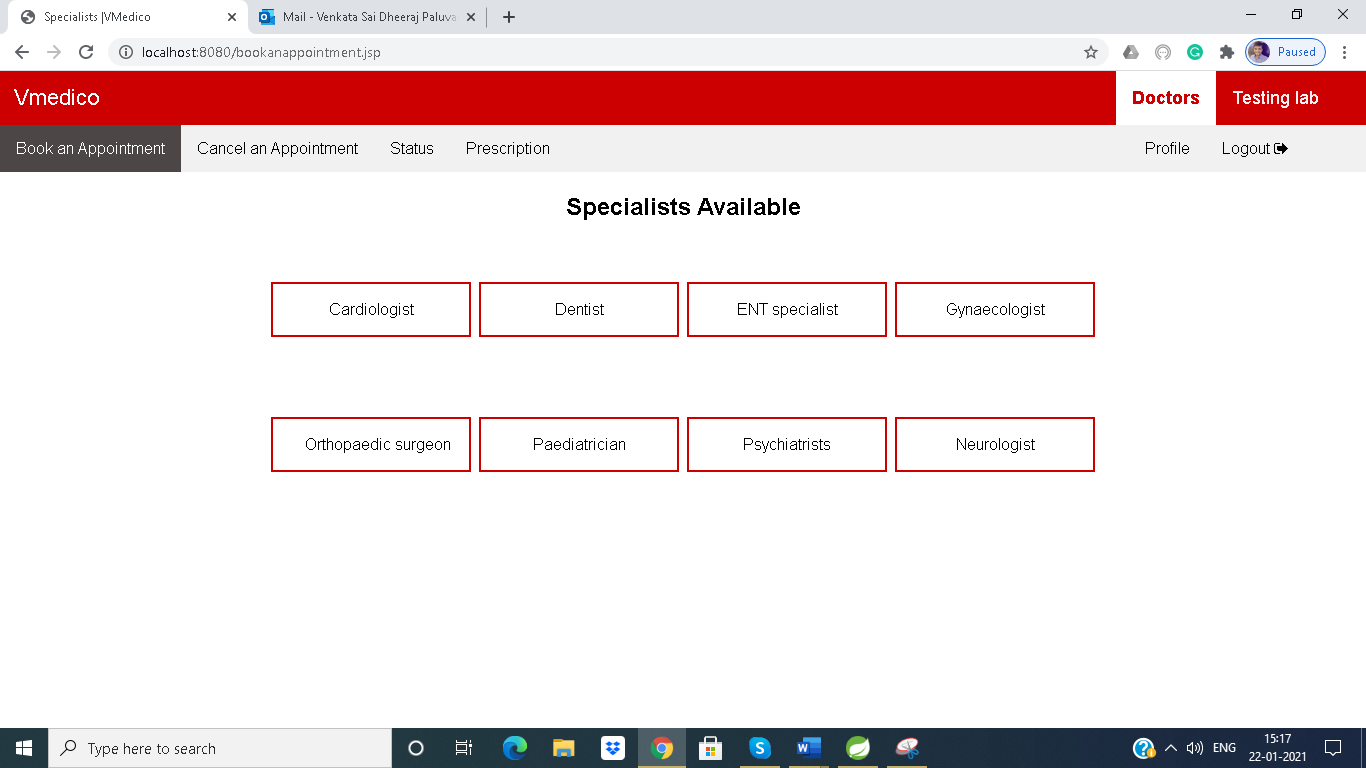
**PATIENT**:



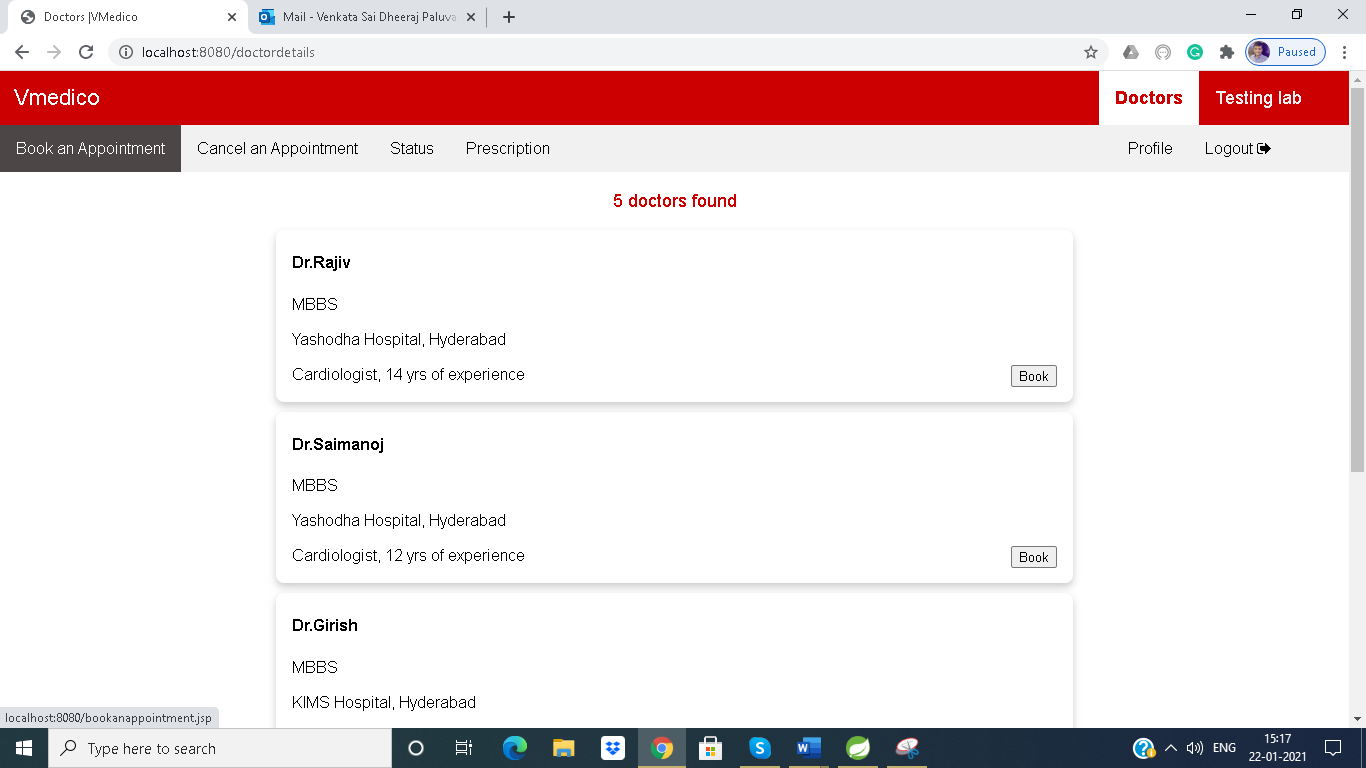
Patient Login form



Patient Profile

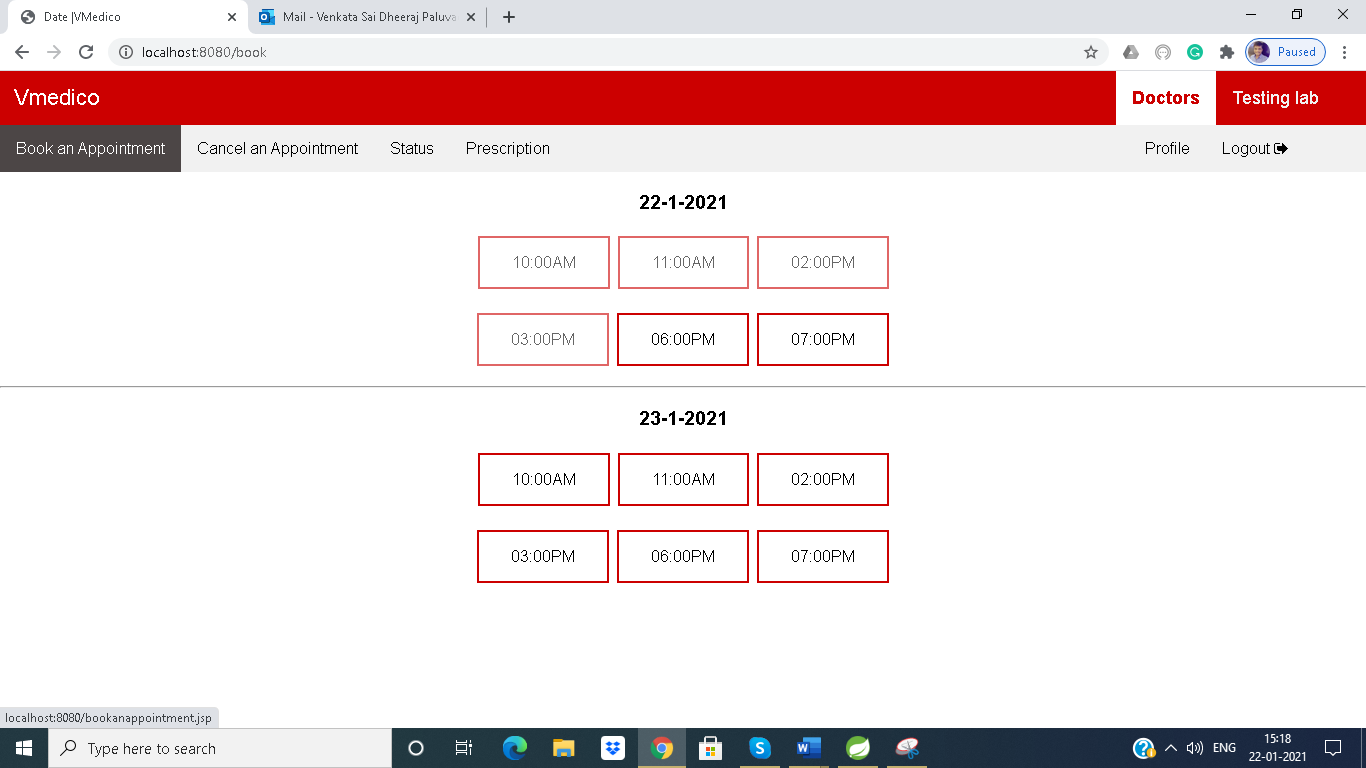


Specialists available for a Patient

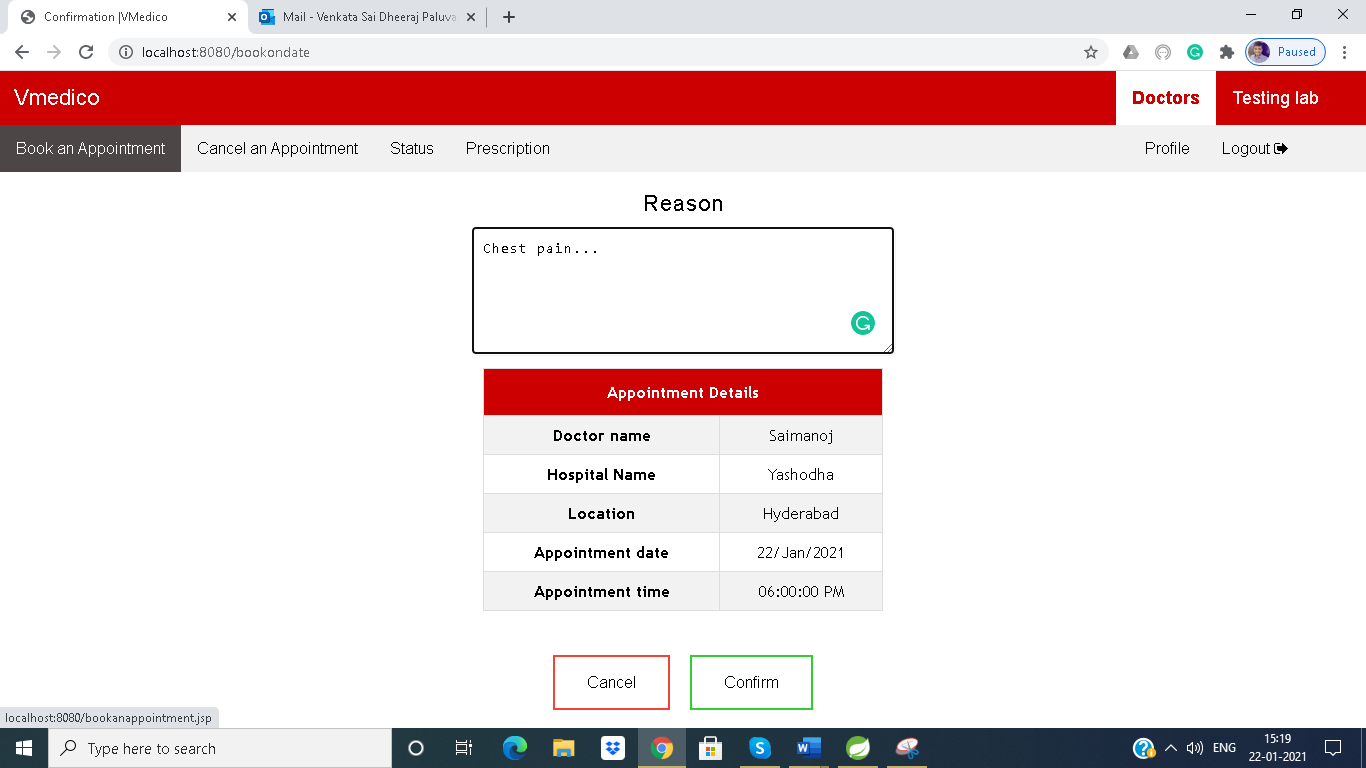


Doctors Available

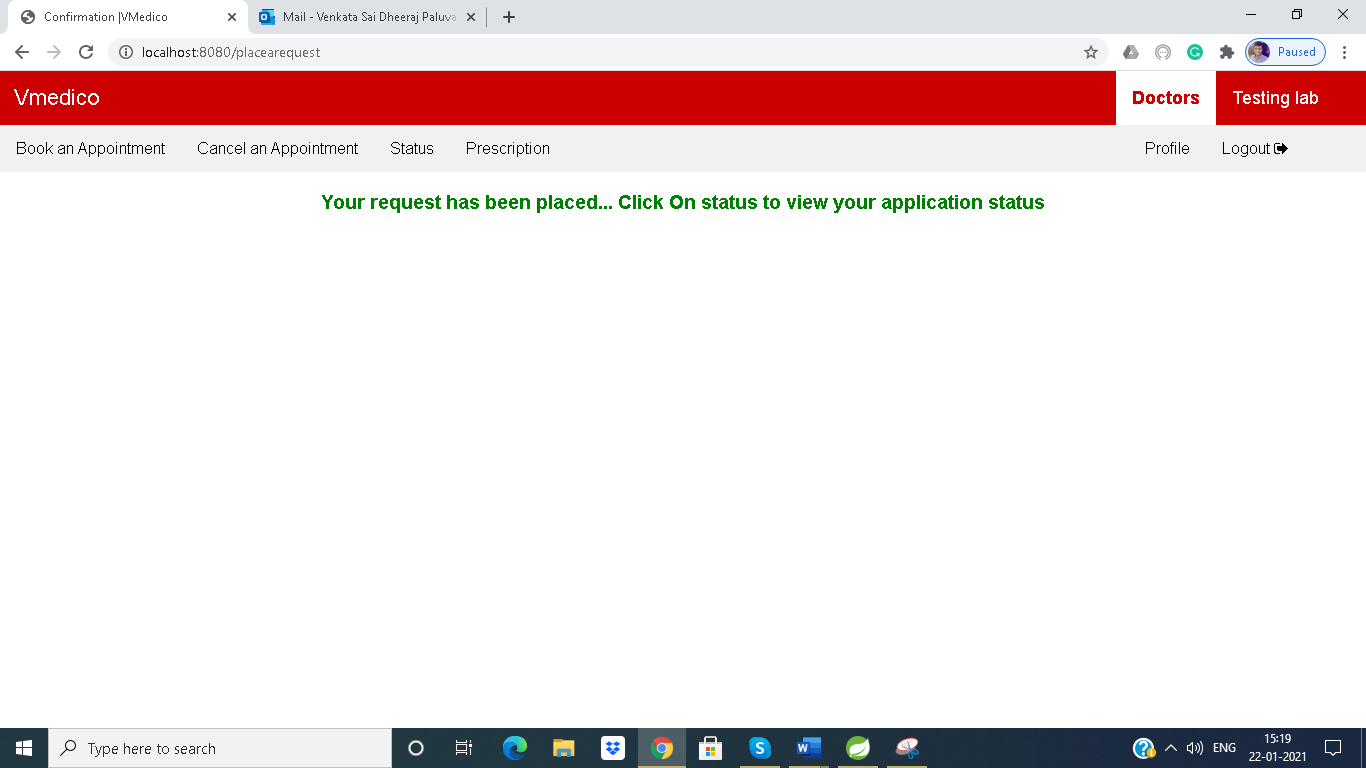
(Doctors are sorted according to their years of experience)



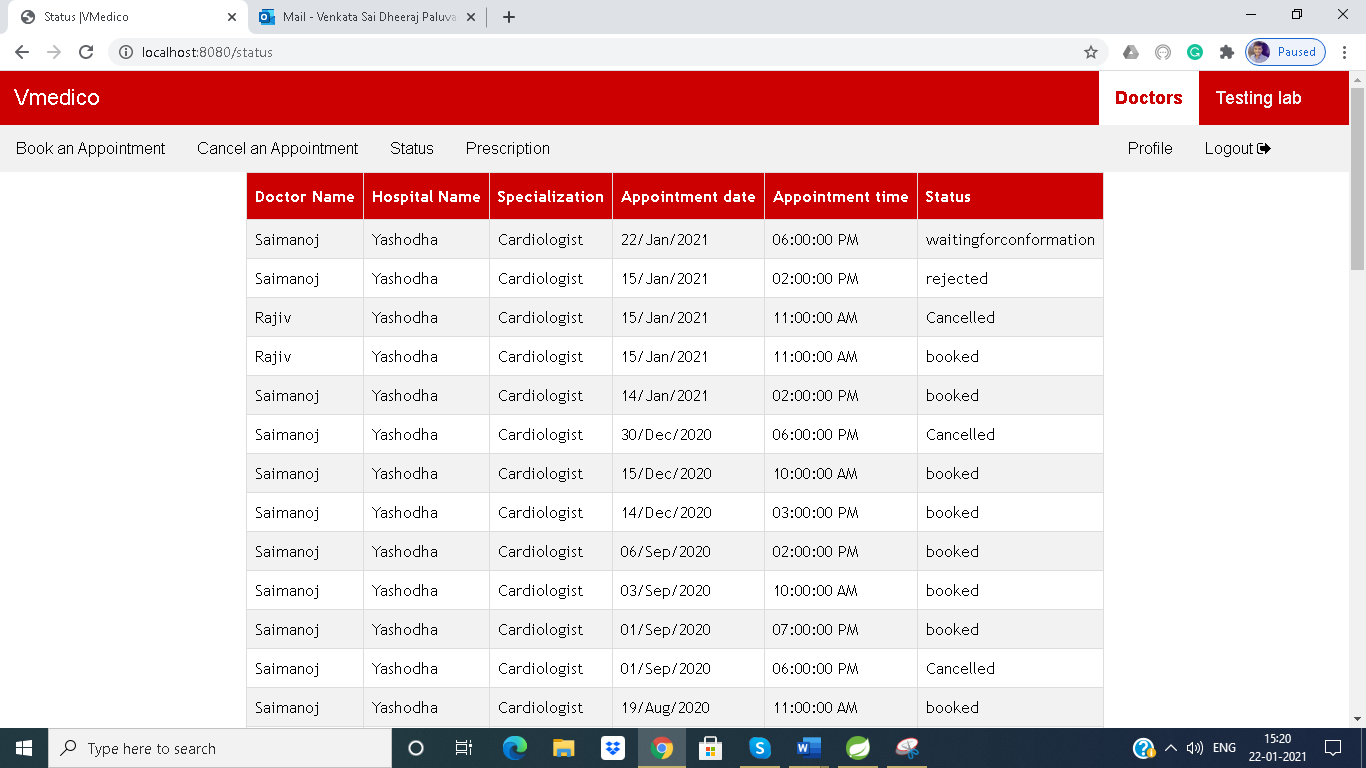
Book on a Particular date



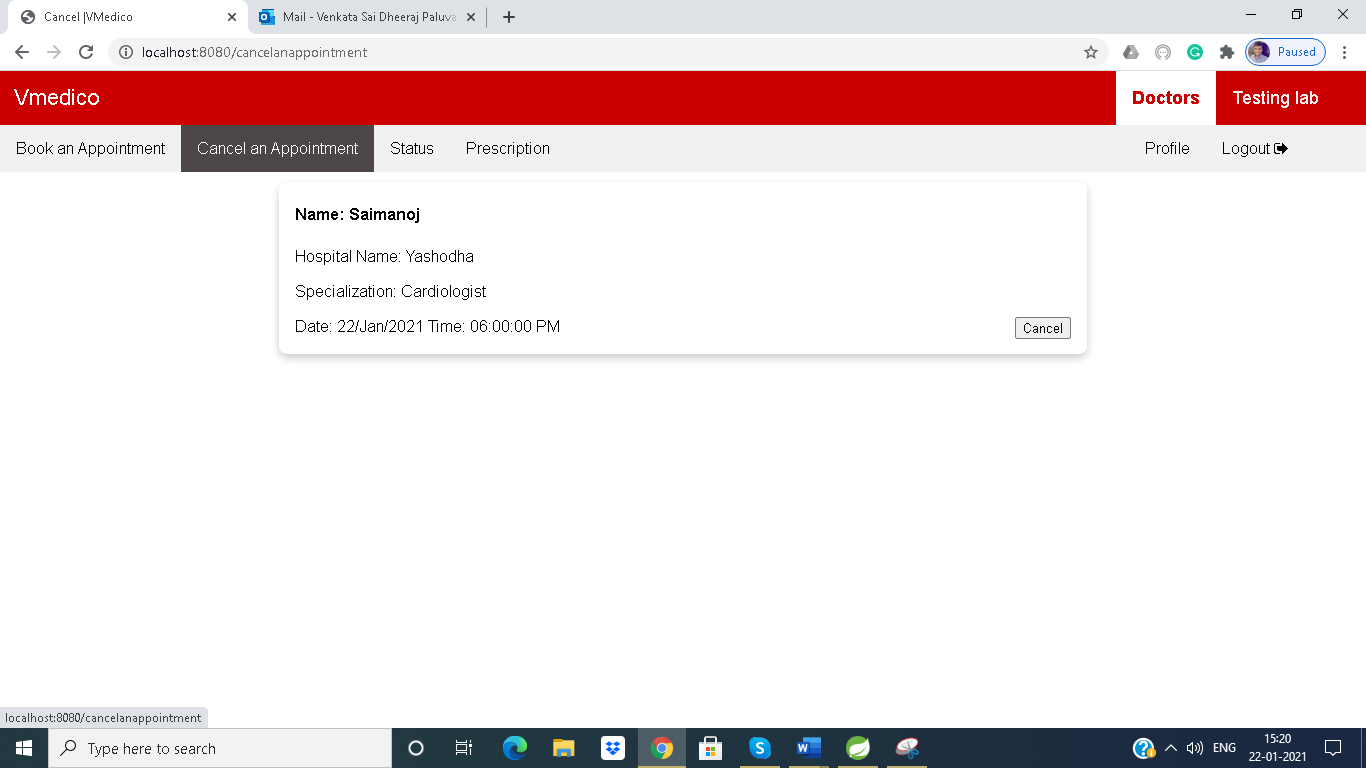
Reason for an appointment



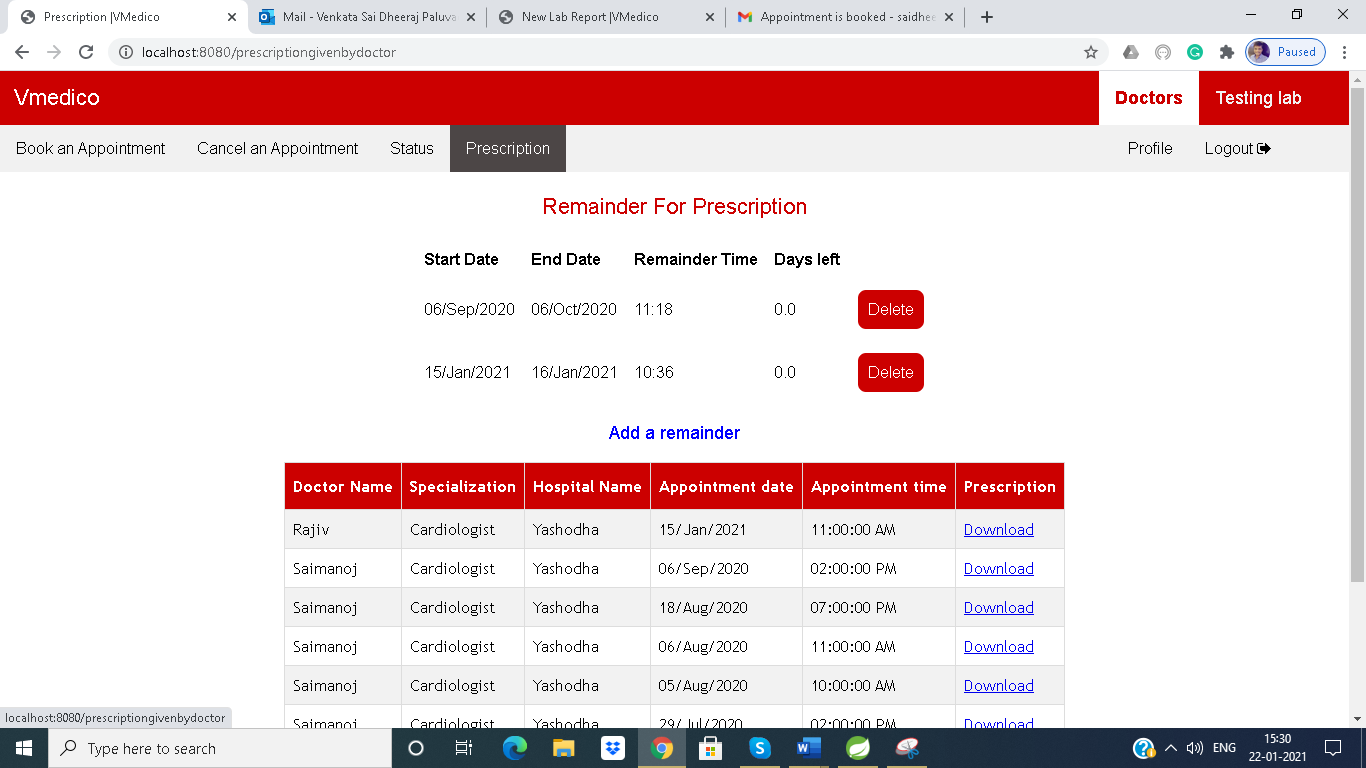
Acknowledgement to the user.



Appointment Status

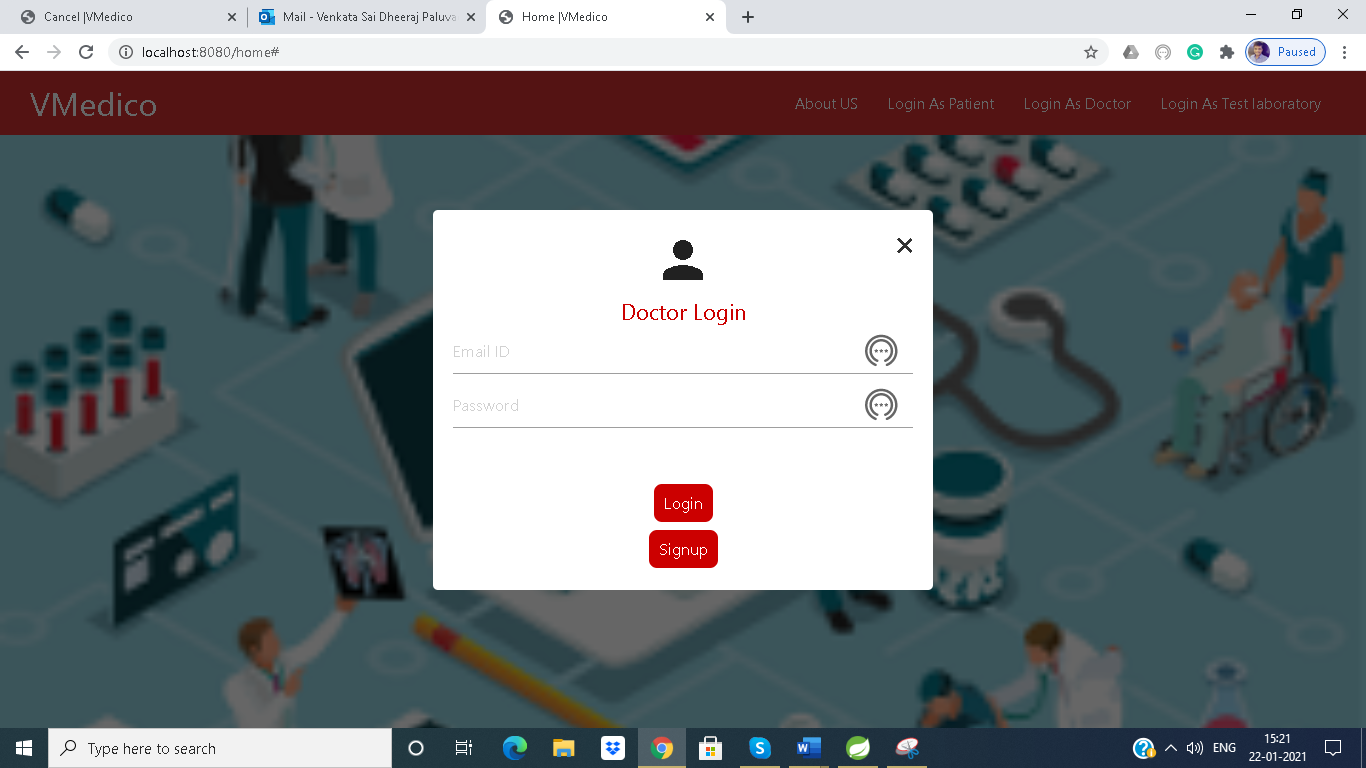


Cancel an appointment

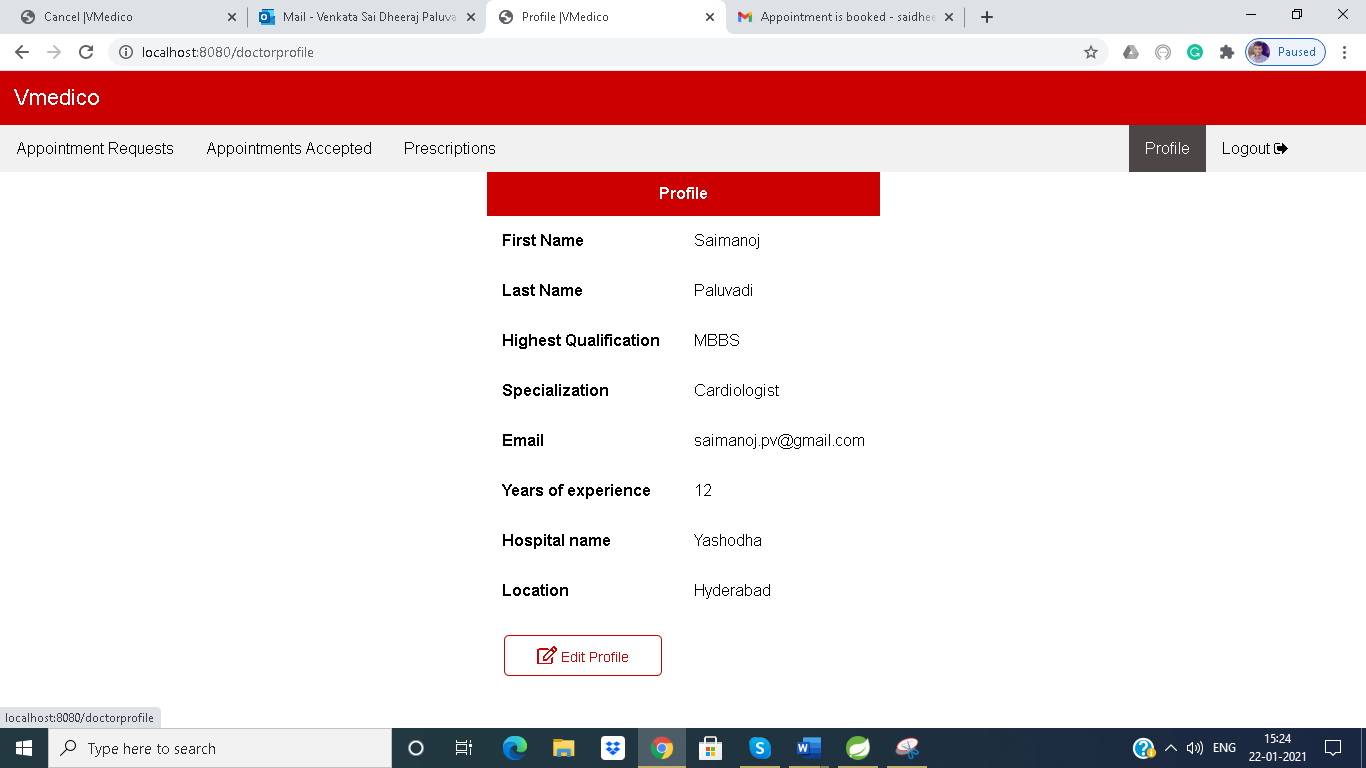


Prescription Tab

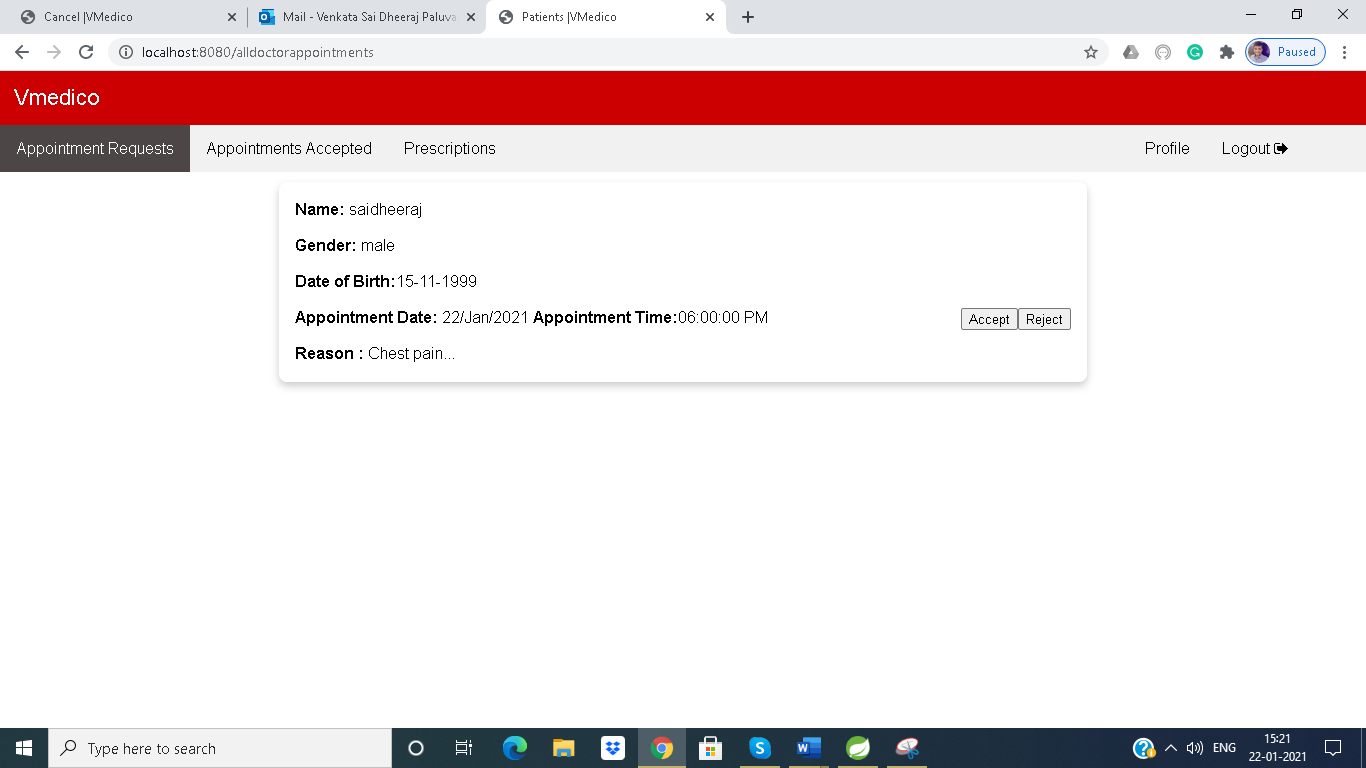
**DOCTORS:**



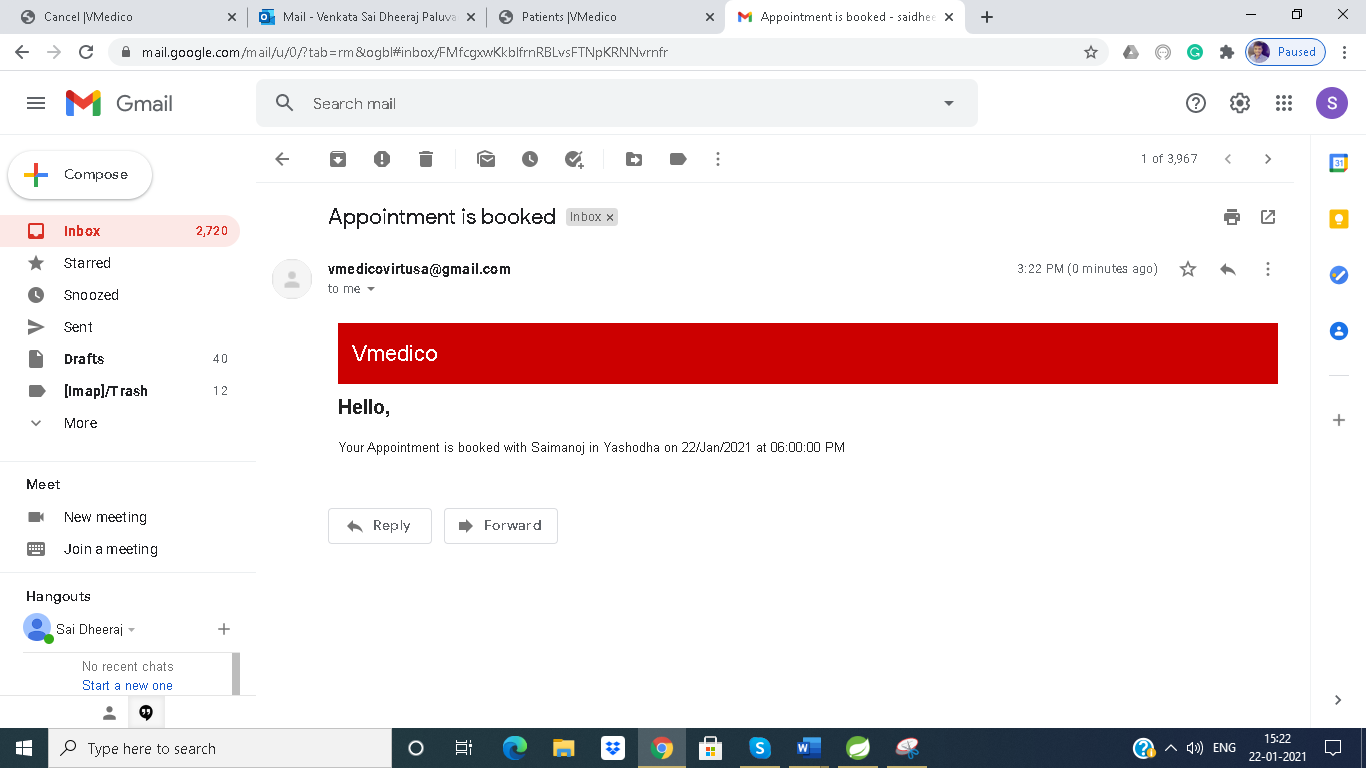
Doctors Login



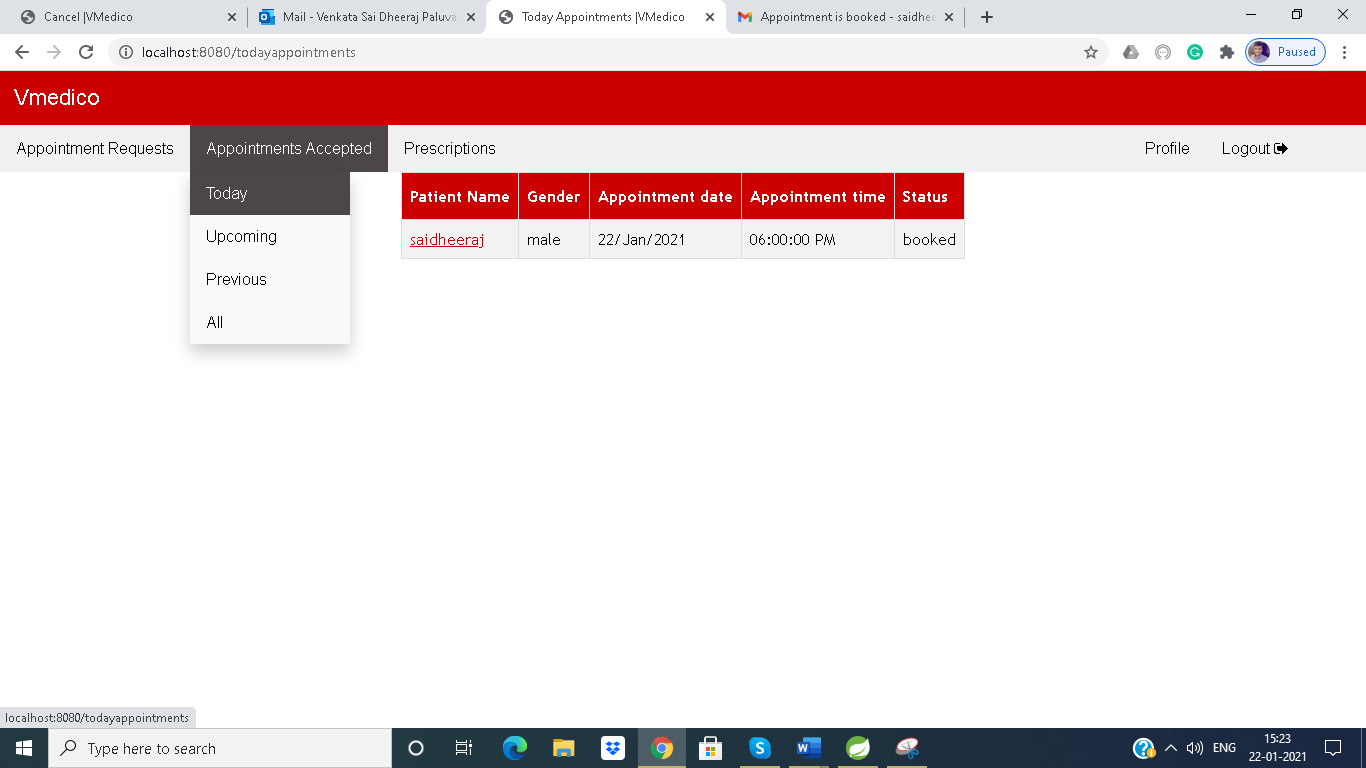
Doctor profile



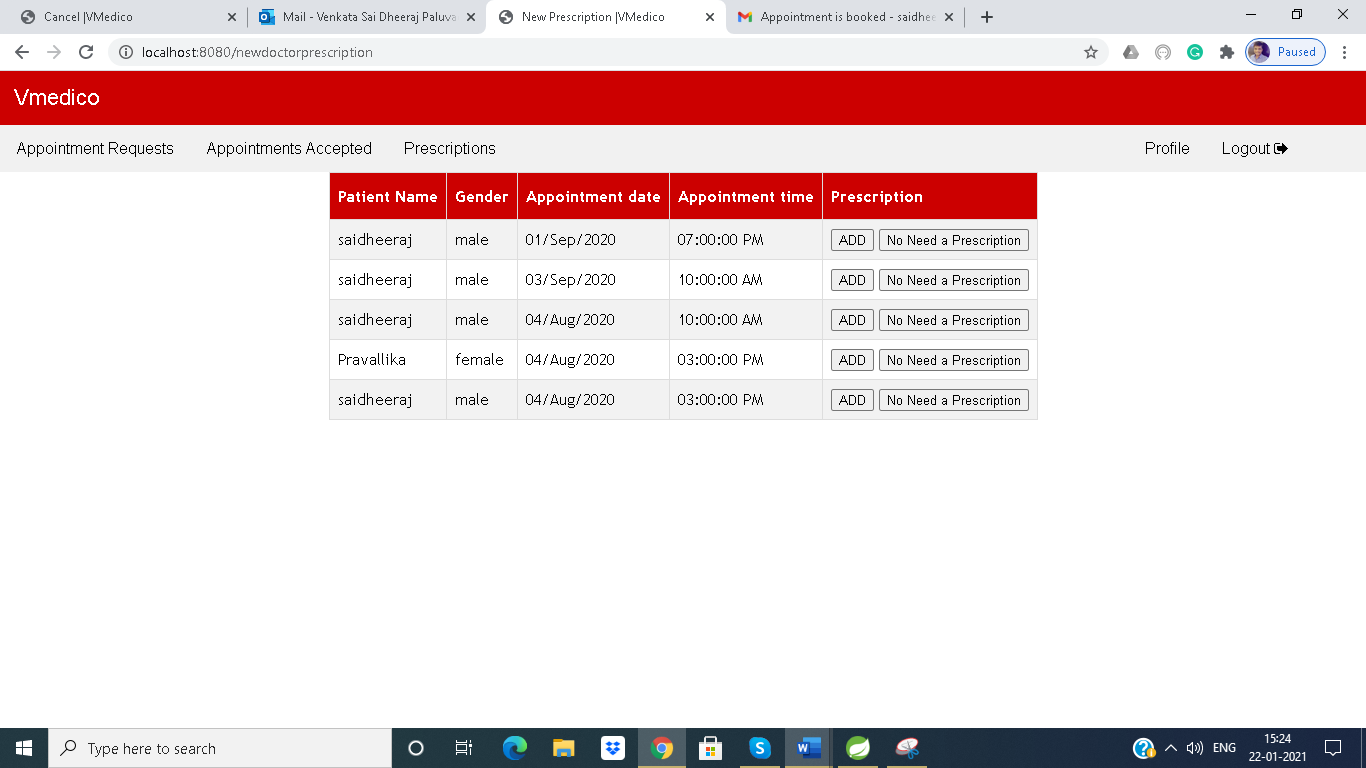
Appointment requests from patients



Email notification for a patient when an appointment is accepted

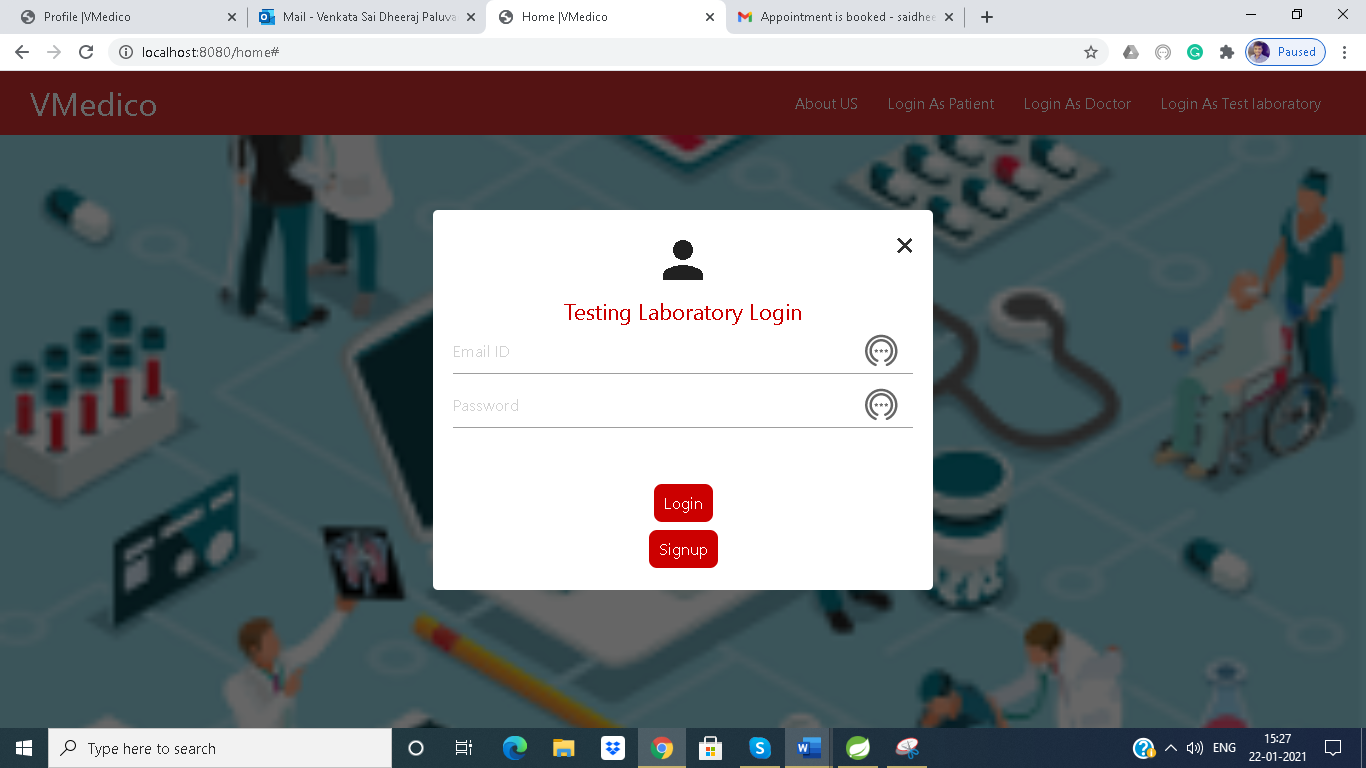


Track of appointments

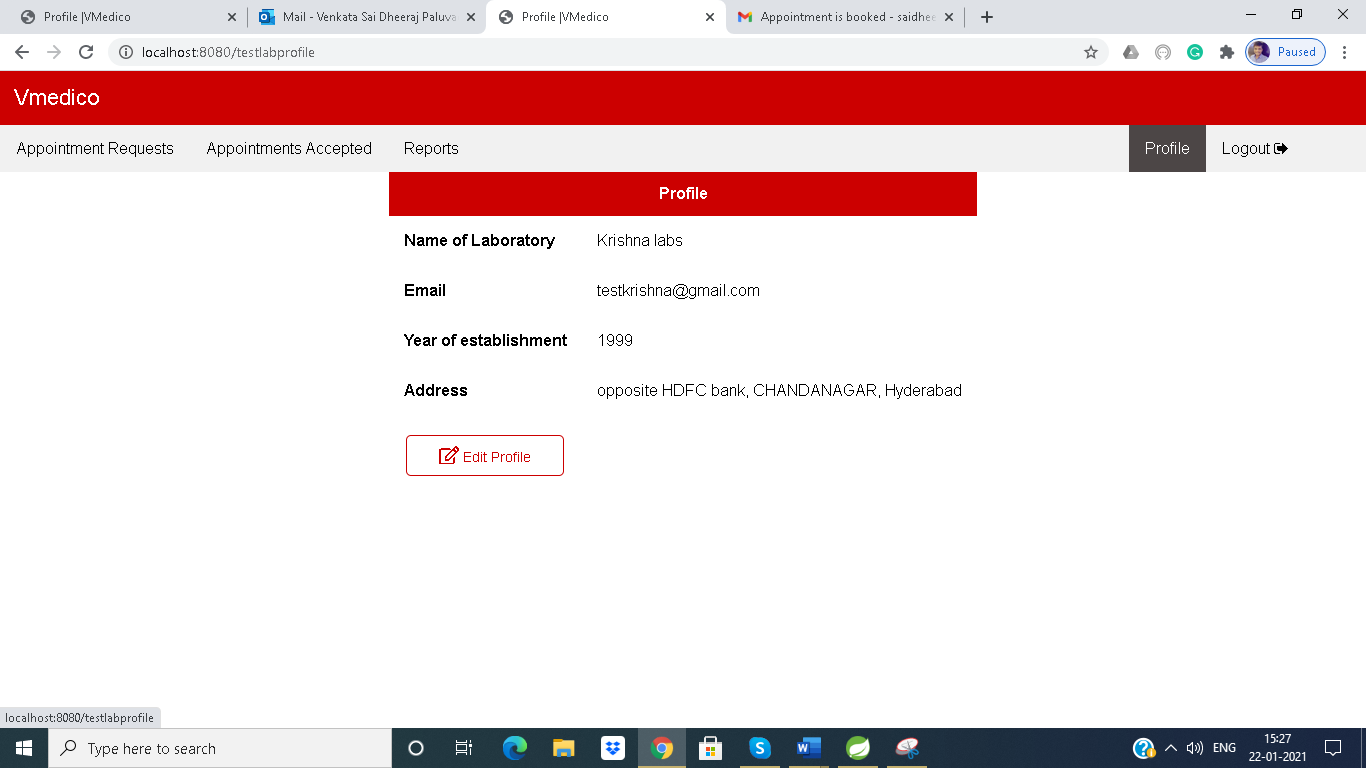


To give any prescription to patient

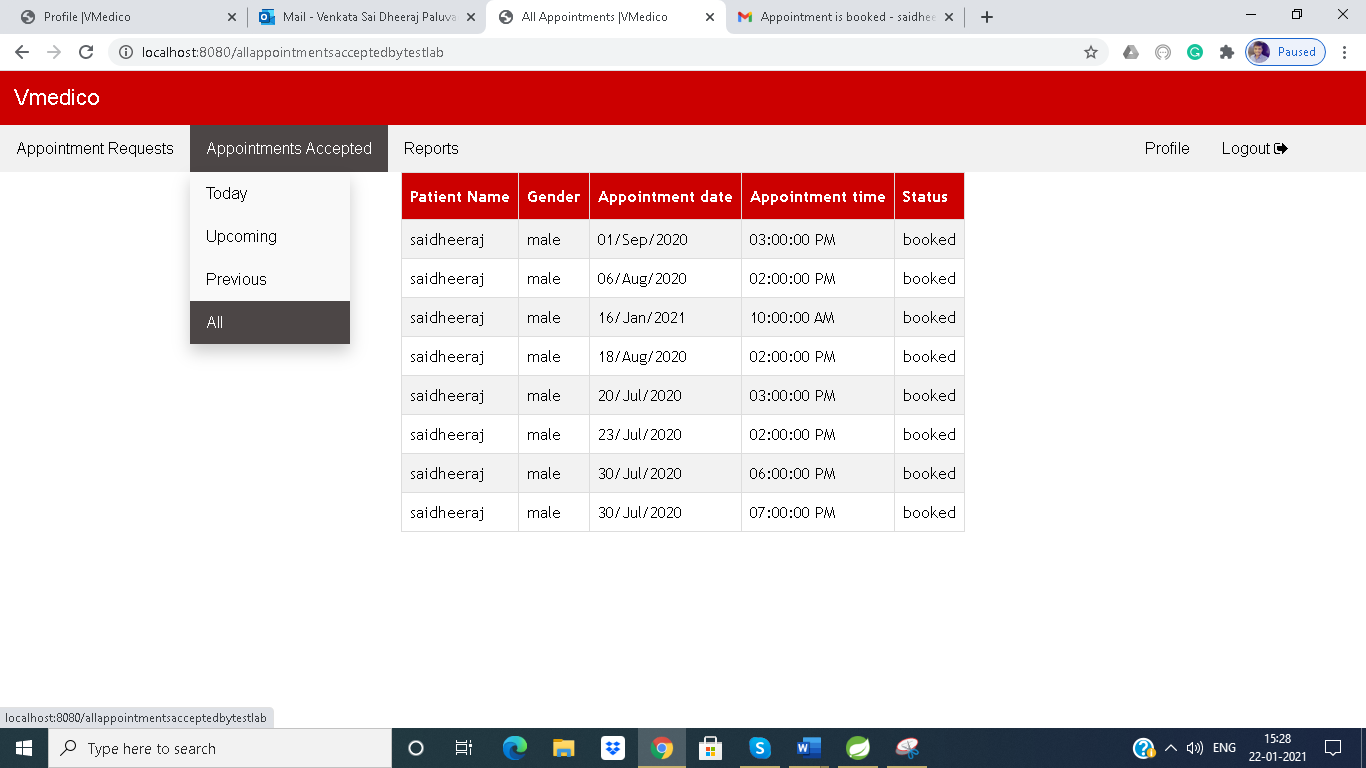
**Testing Laboratory:**



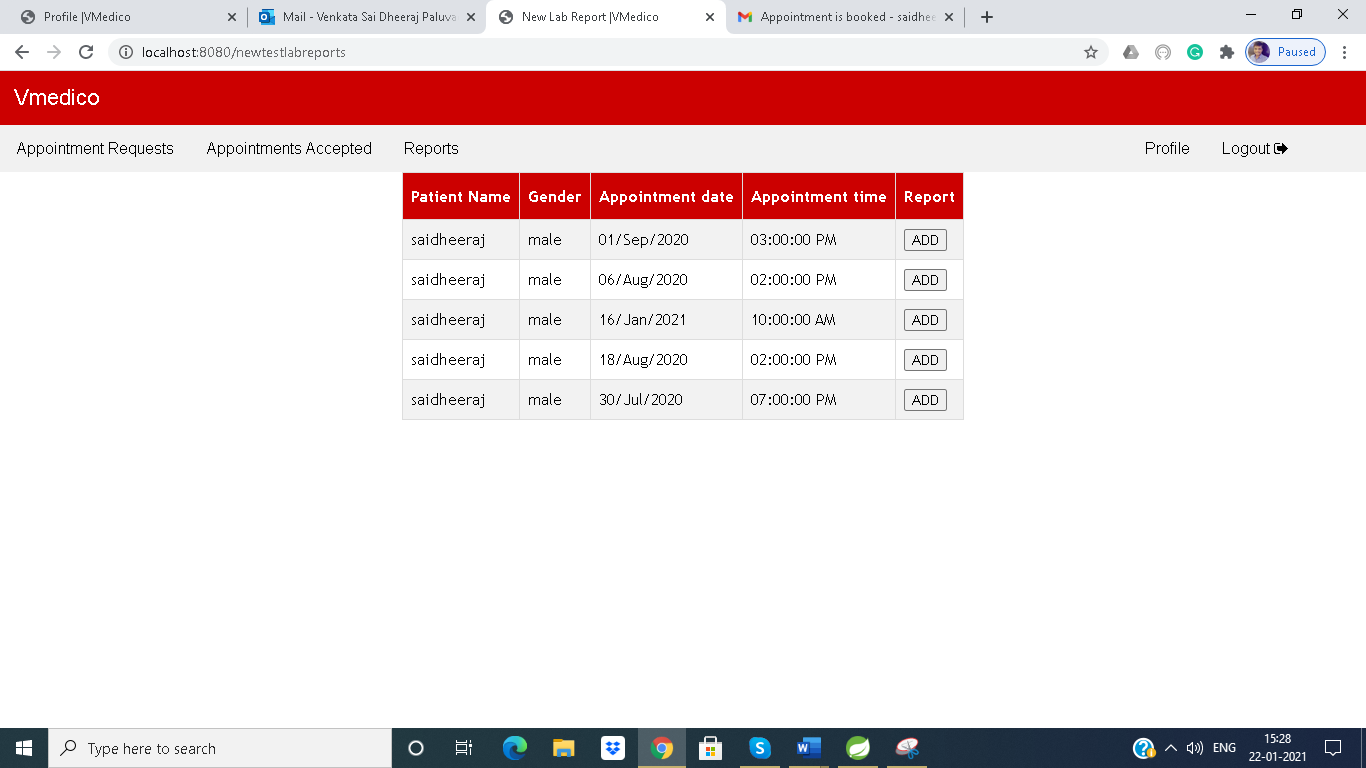
Testing Laboratory Login



Testing Laboratory Profile



Track of Appointments



To give lab reports to patients

**REFERENCES**

1. [https://www.javatpoint.com/spring-tutorial](https://www.baeldung.com/learn-spring-course)
2. [https://www.tutorialspoint.com/spring/index.htm](https://www.baeldung.com/learn-spring-course)
3. [https://youtube.com/playlist?list=PLsyeobzWxl7oA8QOlMtQsRT\_I7Rx2hoX4](https://www.baeldung.com/learn-spring-course)
4. <https://www.baeldung.com/learn-spring-course>