CMU B.Sc. (HONS) SE /B.Sc. (Hons) SE- ASSIGNMENT FEEDBACK SHEET - ICBT CAMPUS

Student Details (Student should fill the content)				
Name	Thanabalasingam Jageeshan			
Student ID	st20301722			
Scheduled unit details				
Unit code	CIS6003			
Unit title	Advanced Programming			
Unit enrolment details	Year	3		
	Study period			
Lecturer				
Mode of delivery	Full Time			
Assignment Details				
Nature of the Assessment	**Course work 100%**			
Topic of the Case Study	**Lab Appointment System**			
Learning Outcomes covered	**1,2,3**			
Word count	4000			
Due date / Time	Jan 2024			
Extension granted?	Yes	No	Extension Date	
Is this a resubmission?	Yes	No	Resubmission Date	
Declaration				
I certify that the attached material is my original work. No other person's work or ideas have been used without acknowledgement. Except where I have clearly stated that I have used some of this material elsewhere, I have not presented it for examination/assessment in any other course or unit at this or any other institution				
Name/Signature	T.Jageeshan	Date		
Submission				
Return to:				

Result				
Marks by 1st Assessor		Signature of the 1st Assessor		**Agreed Mark**
Marks by2nd Assessor		Signature of the 2nd Assessor		

Comments on the Agreed Mark.

For Office use only (hard copy assignments)

Receipt date		Received by	

STUDENT NAME:	**STUDENT NUMBER:**
Module Number & Title:	**Semester:**
Assignment Type & Title:	
For student use: Critical feedback on the individual progression towards achieving the assignment outcomes	
<u>**For the Assessors' feedback**</u> **Indicate the Task number strengths and Weaknesses and the marks for each task**	
Task No/Question No	**Strengths**
Task No / Question No	**Weaknesses**
Areas for future improvement	

Marks			
Task /Question No	**Allocated Marks**	**Awarded Marks**	**Remarks**
Total Marks			
Name and the Signature of the Assessor			
Date			

Acknowledgement

I would like to express my sincere gratitude to God for blessing me with the completion of this project. I extend my thanks to my lecturer for their guidance. I am thankful to Cardiff Metropolitan University, Western Av, Cardiff, Wales, UK, and International College of Business and Technology Campus, Jaffna, for providing me with the opportunity to work on this assignment. Lastly, I am grateful to my family and friends for their unwavering support throughout this project.

Thanabalasingam Jageeshan

st20301722

Table of Contents

List of Figures

1.Introduction

ABC Medical Laboratories is providing medical tests for patients. Currently, they are maintaining patient details and records manually. Due to finding issues in the organization team of the firm decided to move toward a web-based application as an Online Lab Appointment System. Such as issues, patients waiting queue more time, staff missing information, suddenly doctor cancelation, mistakenly granting another person’s reports and Token Numbers through the use of this system. Patients with a better and more efficient way to book appointments across lab tests avoid bottlenecks caused by long wait times and complicated booking procedures. With its Instinctive interface, patients effortlessly navigate through available time slots, handpick appointments tailored to their preferences, and confirm bookings with unparalleled easily. The System's advantages sickout to both patients and healthcare providers. For patients, this outstandingly reduces the need to physically visit facilities through the system. The patient does not wait long while trying to get appointments over the phone. Instead, they can conveniently book appointments from their home or on the go, saving our patients time and reducing stress. Additionally, it is a catalyst to extend access to health services, which advances those who suffer from mobility issues or busy schedules. however, by granting rapid access to critical diagnostic tests, the computer plays a movement role in the early, easy diagnosis and management of medical disorders, eventually leading to better laboratory final outgrowth.

Medical-based experts discussed that this web-based solution is innovative in terms of improved operational efficiency and resource management. Medical side work persons can concentrate on delivering quality care rather than gimmicking administrative responsibilities by automating the appointment web-based booking system. Moreover, the system's capacity to reduce the probability of overscheduling or scheduling conflicts guarantees improved clinic operations and patient pleasure. To put it explain that, through the move from manual to web-based, which benefits a user-friendly, effective, and easily accessible method of scheduling medical test appointments, heralds a new day of healthcare technology. The system can completely change how patients take necessary diagnostic services by using the revolutionary power of digital innovation. This would unlimited lead to better health outcomes and an unmatched level of patient intimacy.

2. Requirement and Specification (Task A)

2.1 Requirements

A requirement is defined as a high-level abstract statement or a detailed mathematical functional specification of a system's services, functions, and controls. They are depictions of the characteristics and functions of the target system. Requirements represent the users' expectations from the software product.

2.2 Functional Requirement

Functional requirements are software requirements expressly requested by end users as basic features of a system. So, these requirements for operations must needs be incorporated into the system as part of the contract. They describe computer behavior under specific conditions. In other words, they are functions that can be seen directly in the final product and it is also the needs of the users.

2.3 Requires an External Interface

Some examples are user interfaces (the communication logic between the program and the user), screen layouts, buttons, features on each screen, hardware interfaces (the list of devices the program wants to run on), and other related information. These requirements.

Software interfaces such as database management systems and front-end and back-end stacks should also be included.

2.4 Non-Functional Requirement

Nonfunctional requirements specify how the system should operate rather than anything to do with its functionality. They frequently have an impact on the whole user experience and are essential for guaranteeing the system's usability, dependability, and effectiveness. The primary types of nonfunctional needs will be covered in more detail later on.

Performance: Since users spend more time viewing content than uploading content, loading content is prioritized.

Usability: Users can intuitively navigate between profiles and subscriptions.

Reliability and scalability: The system should work well at low latency levels, displaying media content with as little latency as possible.

Security: Instagram has extensive authentication protocols, photo uploading, API integration, photo embedding, and encryption of direct messages.

2.5 Different of functional requirement and non-functional requirement

- Functional requirements are easy to define as they are driven by the business idea. They cover all the features of the program and the ways users engage with those features.
- On the other hand, nonfunctional needs are experientially driven. The best way to identify unfunctional requirements is to analyze the performance of your product to make it useful and convenient for users.
- To draw a line between functional and non-functional requirements, it is easy to look at an example. Let's take Instagram as an example and consider how a development

2.6 Needs of Minimum Hardware Specification

- Hard Disk: Software performance can be dependent on storage capacity and type (SSD or HDD), certainly about loading times and data access rates. Make sure the program and any of its files fit on your storage device. Must be leased Space(1GB)
- Memory: The computer uses RAM to store data that is being used or processed right now. More is required for software with higher memory requirements to function properly. confirm the software's recommended RAM size(1GB).
- Processor: The central processing unit (CPU) is the brain of the computer, and software operation depends heavily on its processing capacity. More powerful CPUs are frequently needed for more complicated software. Look for details like the number of cores and clock speed (in GHz). This system run Must of needs minimum core 2 Duo processors.
- Internet connection: needs of Updates and online features may require an internet connection, depending on the software.

2.7 Needs of Minimum Software Specification

Make sure the program you want to use is compatible with the operating system on your computer. Certain software may only function with particular operating systems, such as Windows 7,8,10,10

3. UML diagram in Use Case diagram, Class diagram, Sequence diagram According to this system (Task B)

The Unified Modeling Language (UML) can be used to imagine software and systems through UML diagrams. To comprehend the designs, code architecture, and suggested implementation of critical software systems, software engineers generate UML diagrams. Moreover, business processes and workflows are modeled using UML diagrams.

3.1.1 Use Case diagram Components According to this system

Actors: -An actor is any external entity that interacts with your application or system, be it a person, group, or organization. They must be external data-producing or data-consuming objects. According to this system actor is Admin, patient.

System: - An organized set of behaviors and exchanges between actors and the system. Another name for a system is a view. Refer to the system (Booking, check out, managing details)

Relation: - The patient can book for the appointment, booked, checkout. Admin can manage laboratory and patient

3.1.2Use case diagram benefits

Use cases highlight the demands of the actual system early on because they center around the users, not the system itself. All system-based stakeholders, not only developers and testers, may easily understand a use case because it mostly consists of narrative prose, including customers, users, and executives. All stakeholders benefit from early planning since it involves the people who are most familiar with the issues at hand, encourages end user buy-in, and removes surprises when the system is implemented.

Each use case outlines a method for using the system However, one of the main advantages of use case modeling is that it covers every possible scenario. Identifying nuanced requirements early in the project helps save a lot of time by identifying exceptions in a successful scenario.

Ultimately, the generated use case model will serve as the basis for many software development processes, such as object models, test case definitions, user documentation, and project planning (cost, complexity, and schedule estimates).

Use case alternative pathways record extra behavior that can strengthen the resilience of the system. Use cases come very handy while scoping. Use cases facilitate the implementation of a phased delivery strategy for software projects, as they may be added or deleted with relative ease in response to shifting priorities. Use cases have shown to be a great way to bridge the gap between software developers and end users since they are simple enough for business people to understand.

3.1.3 Use Case diagram

A use case diagram is a visual representation of the connections and interactions inside a system. These drawings offer a comprehensive view of a system. They could show customer interactions, computer programs, or business procedures. A use case diagram shows an imaginary scenario in which users utilize a set of connectors and particular symbols to interact with a system.

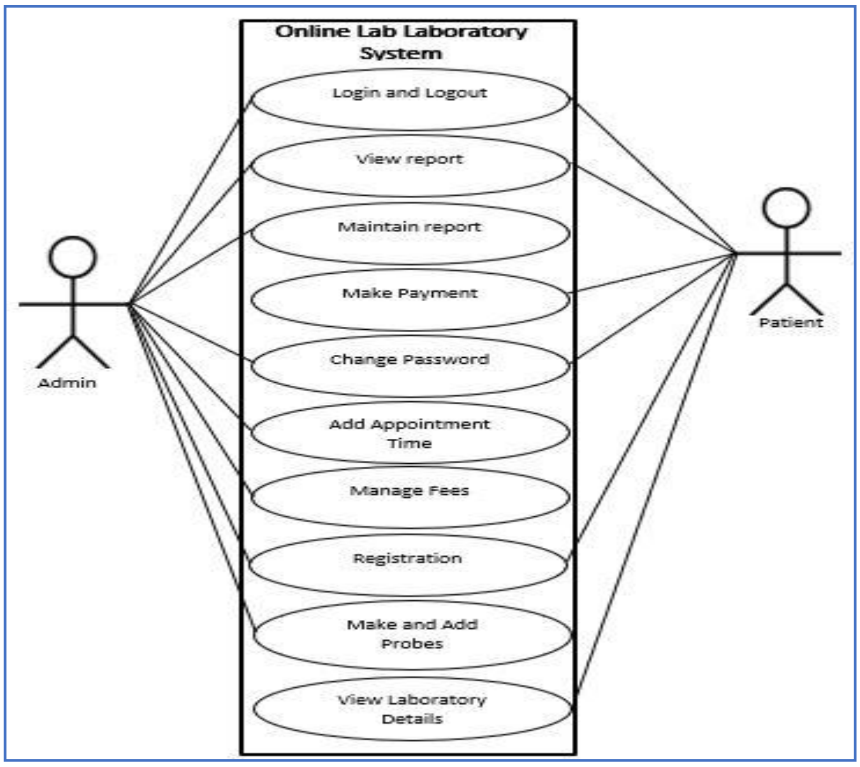


Figure 1:Use Case Diagram

3.2 Class Diagram

A class diagram is representing a static view of the application. used not only to visualize, describe and document various factor of a system but also to create the execute code of a software application.

This diagram attributes and functions of a class and the constraints imposed on the system. This diagram is used in modeling object-oriented systems because they are the only UML diagrams that map directly to object-oriented languages. Class Name: Generally, the class name is bold, centered and written in the top box of the class box.

Attributes: Also referred to as fields or properties, attributes serve as a representation of the class's data members. They are located in the class box's second compartment and frequently include each attribute's data type as well as visibility (public, private, etc.).

Methods: The activity or functionality of the class is represented by methods, which are often referred to as functions or operations. The parameters, return type, and visibility (public, private, etc.) of every method are listed in the third compartment of the class box.

Visibility Notation: Visibility notations show which properties and methods are accessible at what level. Common visible notations include (+ for public: visible to all class, - for private: visible only with in class, # for protected, ~ for package or default visibility.

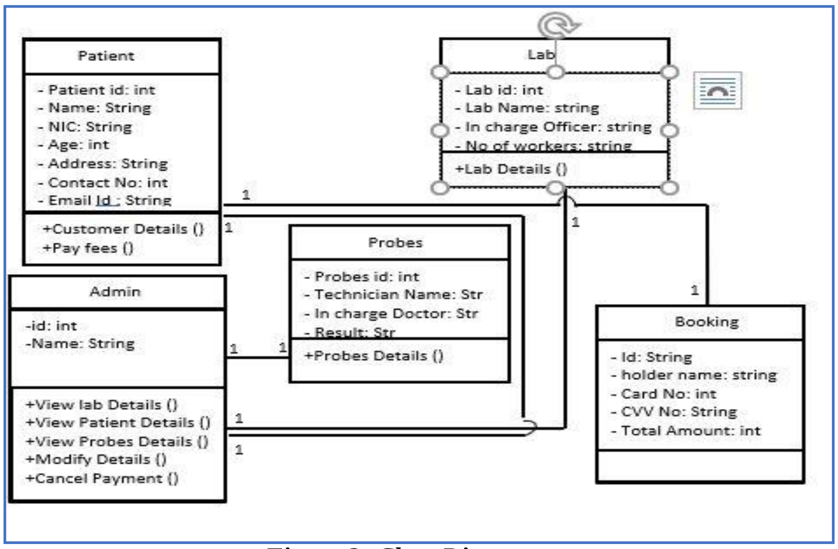


Figure 2: Class Diagram

3.3 Sequence diagram

A queue diagram is a communication diagram that emphasizes the temporal ordering of messages. It depicts the objects and classes involved in the scene and the sequence of messages exchanged between the objects necessary to implement the scene's functionality.

Time-based sequence diagrams help users visualize the sequence of exchanges by using a vertical axis to show which messages are sent when.

Sequence diagrams show interactions at various granularities:

Subsystems (also called system sequence diagrams), high-level interactions between a system and its user or between the system and other systems

Whether it's a use case or activity (example diagrams or generic diagrams), communication is a collaborative effort.

In the MVC paradigm, representation objects interact with the program architecture (model, view, and controller).

About this diagram: -The website is where patients initially view this system. then called to ask for advice on an issue. after his search for his medical examination. regularly obtained data and made reservations using the system.

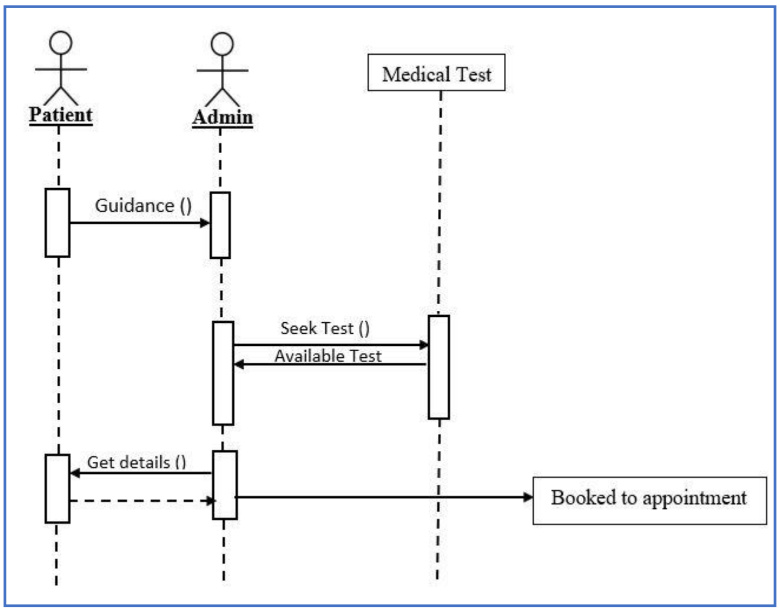


Figure 3: Sequence Diagram

4. Pattern of the system (Task C)

4.1 Singleton

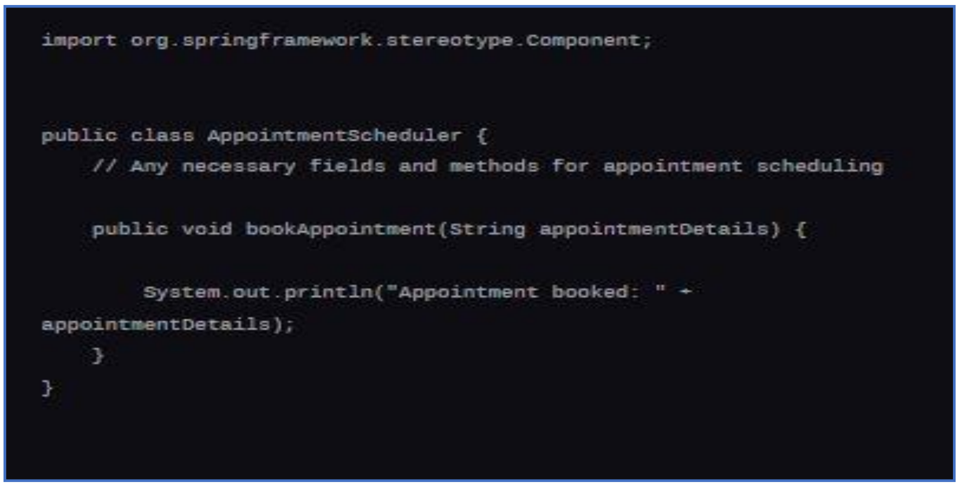
The singleton pattern is among the simplest design patterns in Java. As it provides one of the most efficient ways to produce an object, this type of design pattern is categorized as a creational pattern.

This pattern lets one class produce objects and makes sure that only one is created at a time. This class provides a way to access its single object, which doesn't need to be created in order to be acquired right away. This method is used in the Online Lab Appointment System.

Methods of implementation Single Tone

- In a singleton instance as access modifier private. static field is store class instance and attribute.
- Declare a public static create method to receive a singleton instance.
- Enable "lazy boot" by default. It should create a new object on its first call and place it in the static field. The method must always return that event on all subsequent calls.
- Make the constructor of the class private. A static method of a class can still calls the constructor, but not other objects.

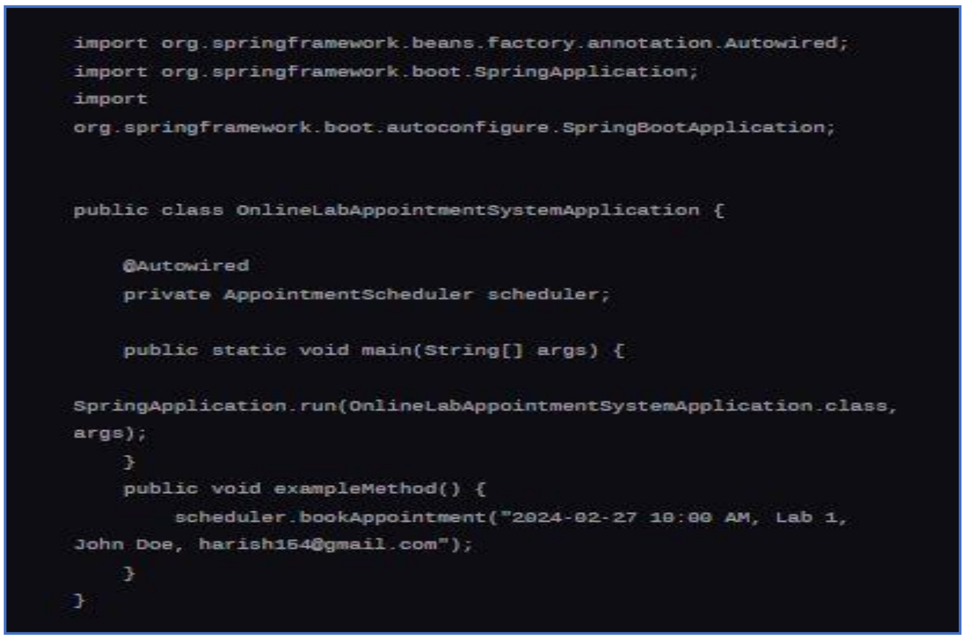
Create Components



```
import org.springframework.stereotype.Component;


public class AppointmentScheduler {
    // Any necessary fields and methods for appointment scheduling

    public void bookAppointment(String appointmentDetails) {

        System.out.println("Appointment booked: " +
appointmentDetails);
    }
}
```

Figure 4: Create Components

Using Components



```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.SpringApplication;
import
org.springframework.boot.autoconfigure.SpringBootApplication;


public class OnlineLabAppointmentSystemApplication {

    @Autowired
    private AppointmentScheduler scheduler;

    public static void main(String[] args) {

SpringApplication.run(OnlineLabAppointmentSystemApplication.class,
args);
    }
    public void exampleMethod() {
        scheduler.bookAppointment("2024-02-27 10:00 AM, Lab 1,
John Doe, harish164@gmail.com");
    }
}
```

Figure 5:Using Components

• Advantage of singleton pattern

Some of the many advantages that make the Java Singleton design pattern an invaluable resource for Java developers include:

Resource Management: By limiting the number of instances of a class, a singleton guarantees effective resource management.

Global access: By providing a global access point, it makes it easy for other classes to interact with a singleton instance.

Lazy loading: By allocating resources only when needed, lazy loading increases instant speed.

4.2 Factory

This type of design method is classified as a creative form because it provides one of the most efficient ways to produce a product. By using the factory design, we can create an object and keep the build code secret from the client by referencing it through a common interface.

The factory system gives the sub-sectors the freedom to choose the products they want to make.

Code only interacts with the resulting interface or abstract class to interact with any classes that implement the interface or extend the abstract class as needed.

Use of factory pattern design

1. A class is not sure about the subclasses it should create

2. When a class requires that the articles of manufacture be specified by its subclasses.

3. When parent classes decide which of their subclasses should create objects for themselves.

Factory Pattern advantages

The end user does not need to worry about the technical details of material creation.

A factory method gives subclasses full control or access to choose the type of object to create.

The main goal of the factory design pattern is to hide the technical details behind object type selection and client code generation. It encourages loose-coupling, so client code only interacts with interface or abstract methods instead of implemented classes.

4.3 Abstract Factory

An interface for creating families of related or dependent objects without specifying specific classes is provided by the abstract factory creation design pattern. It also goes by the name Kit Pattern.

Software engineering systems often require the creation of multiple families of related objects or dependencies. The Abstract Factory pattern provides an abstraction layer that captures the process of creating these groups of objects. It encourages loose coupling between client and concrete classes by allowing client code to create objects without knowing the individual classes or implementation details.

4.3.1 Type of abstract

1. Abstract factory

Factory methods for creating families of related objects are declared in this interface, also known as an abstract class. Each factory method requires a specific type of material to be produced.

2. Concrete factory

These are how the abstract factory interface is implemented. every concrete plant to produce a family of products that fall under a particular variation or category.

3. Abstract product

Abstract The common methods to be implemented by all products (objects) produced by the factory are declared in this interface, also known as the abstract class.

4. Concrete product

These are how the abstract product interface comes into play. Each concrete product produced by a concrete factory is a unique product.

5.Distributed Application with webservice and database (Task D)

5.1 Distributed Application with webservice

A web service is a communicate with web-based application pair to over a network. A web services implementation allows two web applications developed in different languages to communicate with each other using standardized media such as XML, SOAP, HTTP, etc.

Features of Web Services

Web services are based on open standards like XML, HTTP. So, these are operating system independent.

Web services typically use standard web protocols that are accessible to a wide range of clients.

Once created, a web service can be reused by different clients, promoting code reuse and reducing development time. So, the web services can update or change their underlying logic without affecting the consumer.

5.1.1 Login page

A login page is a web page that asks a user to enter their login information to gain access to a service or website. Usually, one can access the website by clicking the login button or link on the home page.

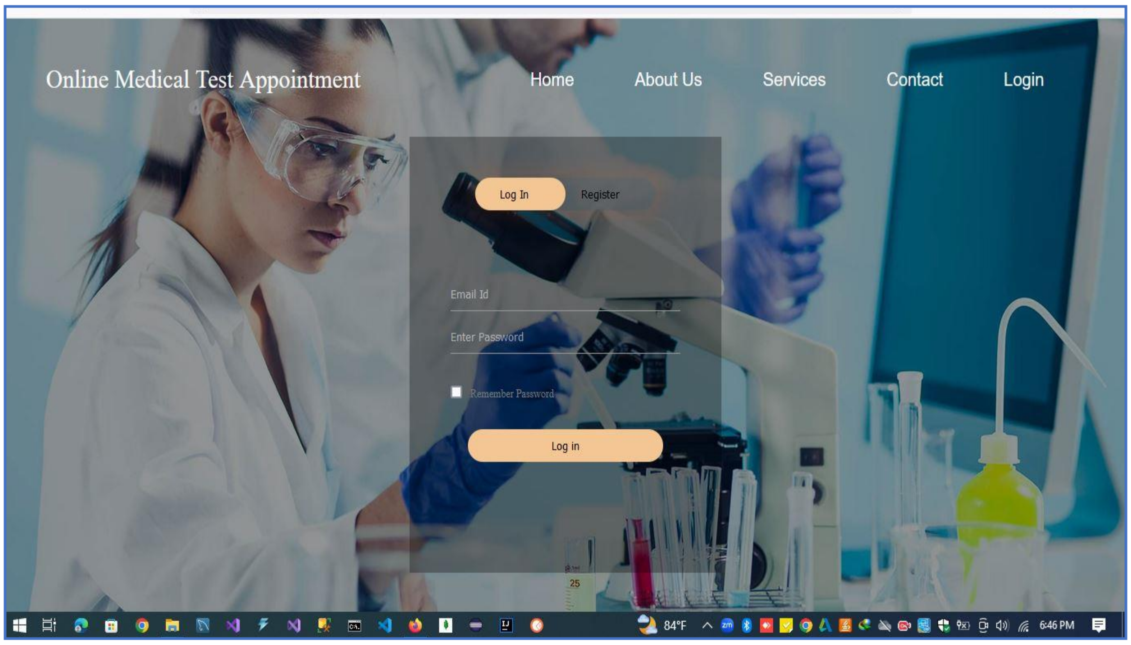


Figure 6: Login Page

5.1.2 User Registration

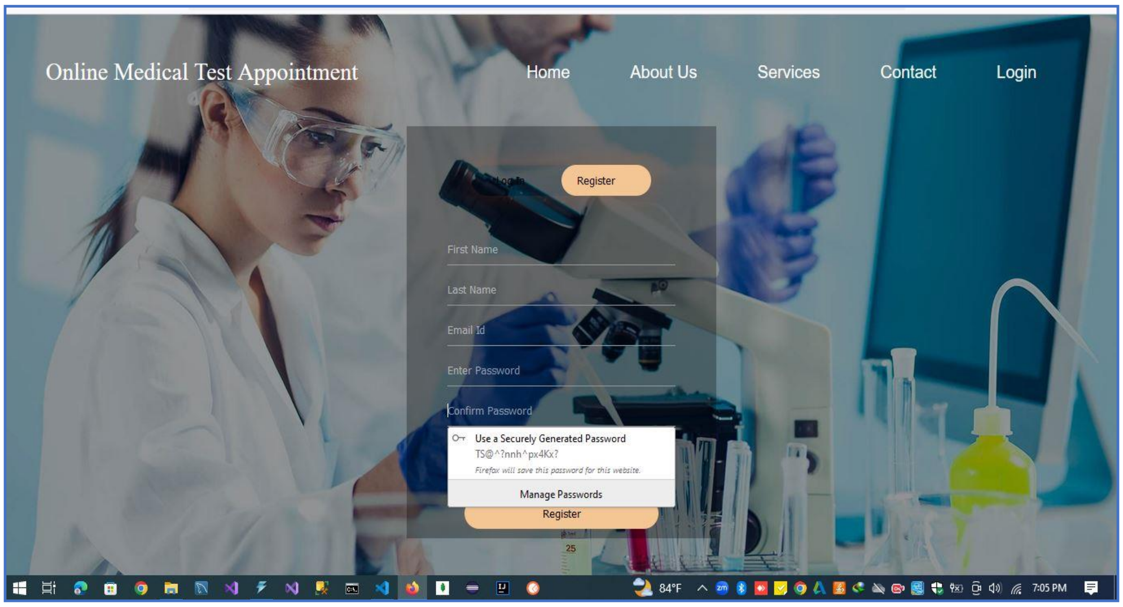


Figure 7: User Registration

5.1.3 User login with validation

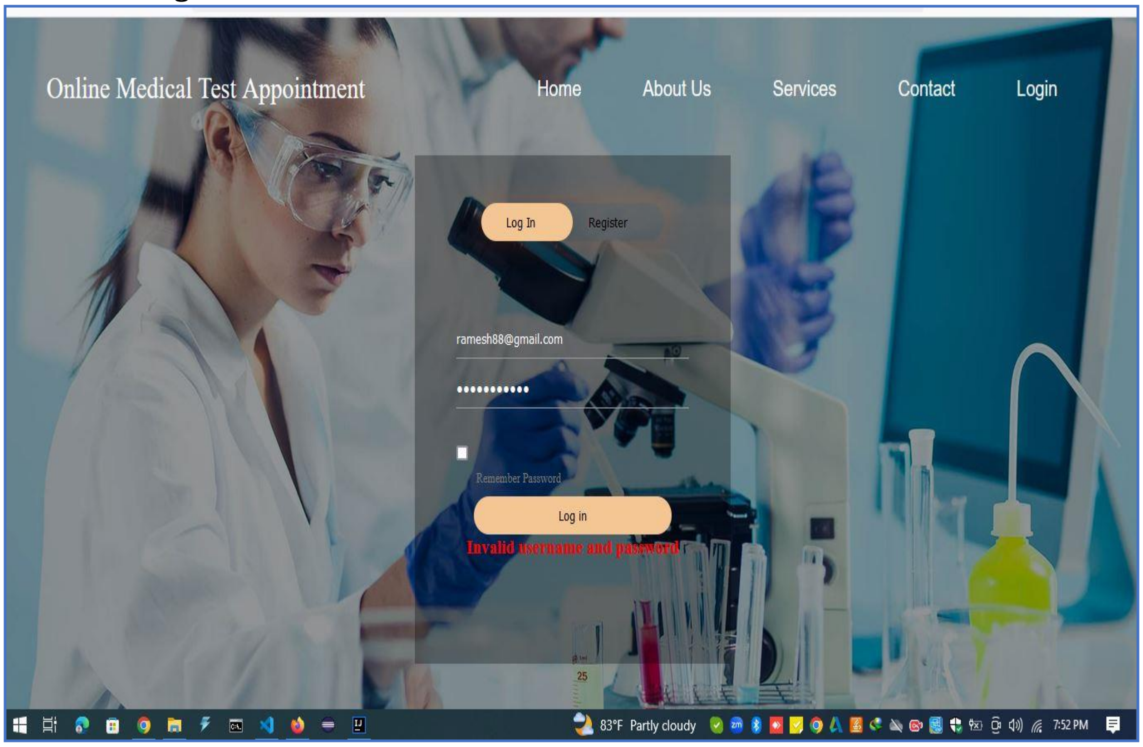


Figure 8: User login with validation

5.1.4 After User Login Home Page

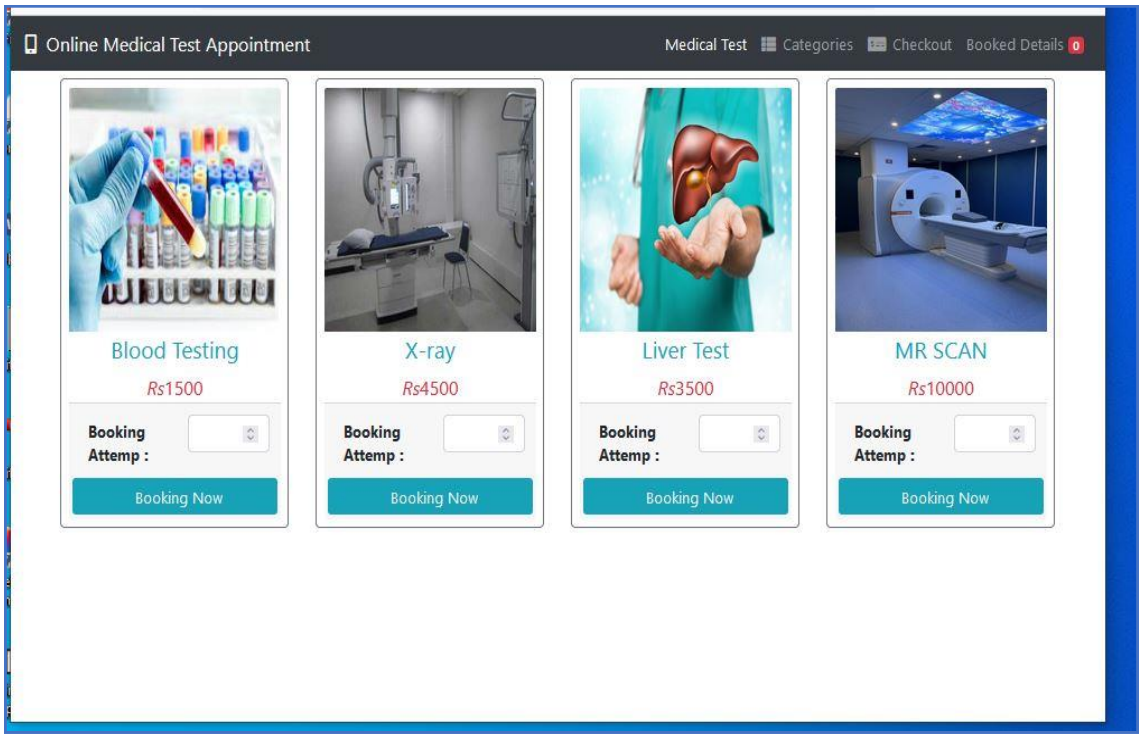


Figure 9: After User Login Home Page

5.1.5 Patient Booking

After login the patient viewing homepage. Then select suitable medical test.

(ex: - Booking X-ray)

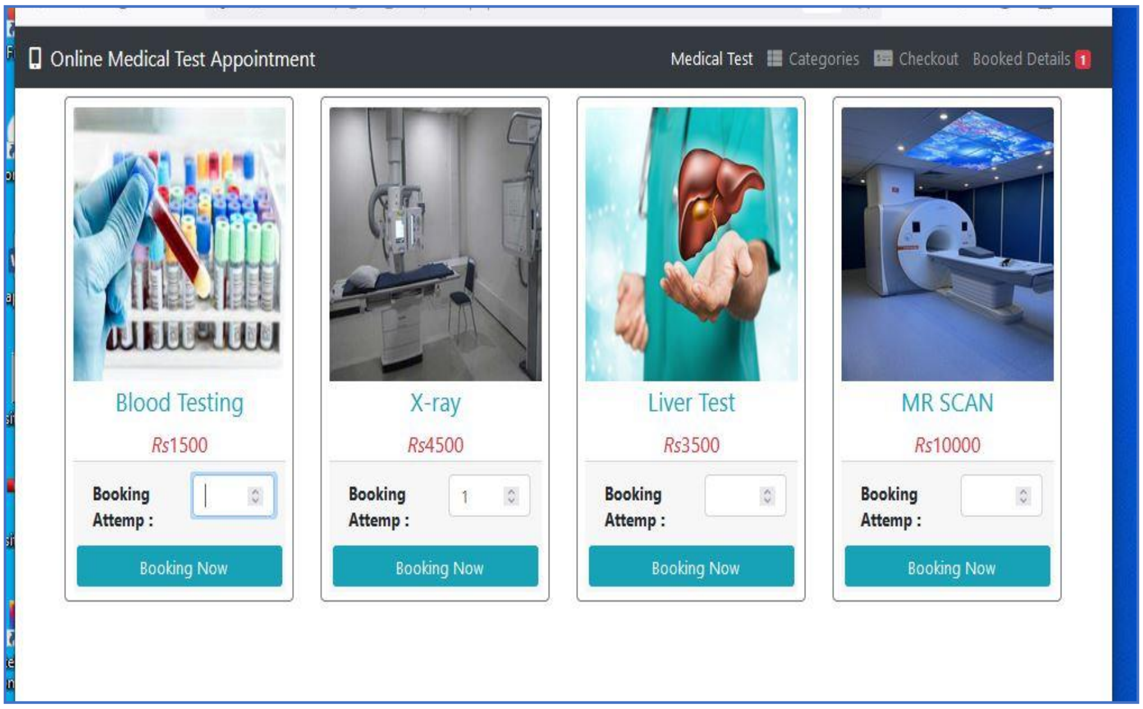
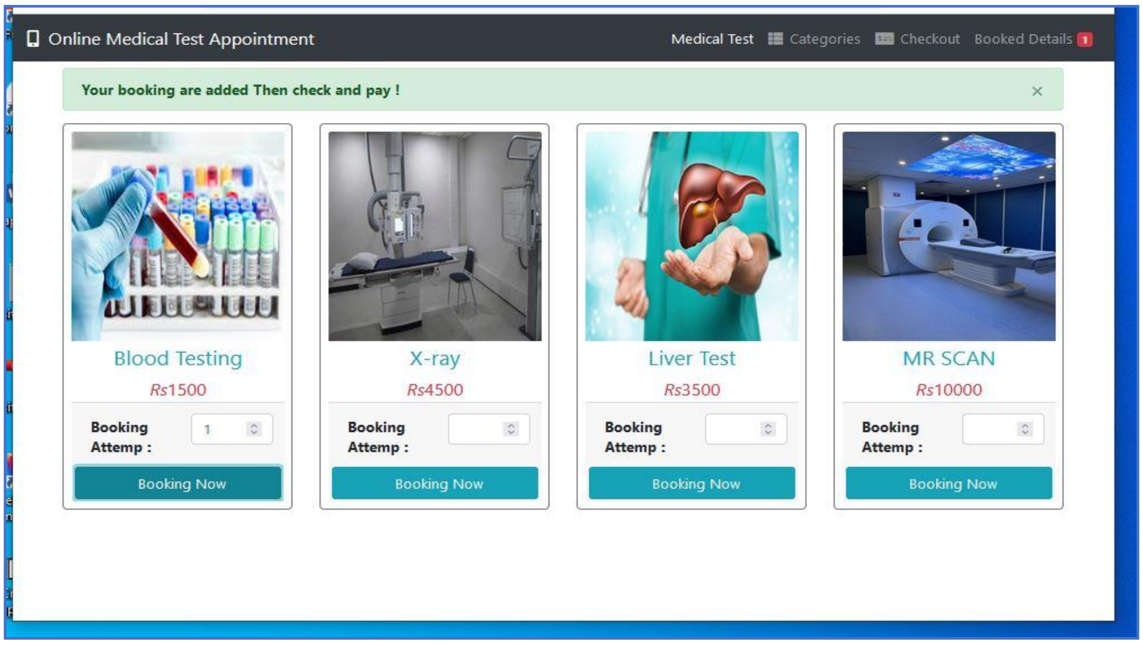


Figure 10:Patient Booking

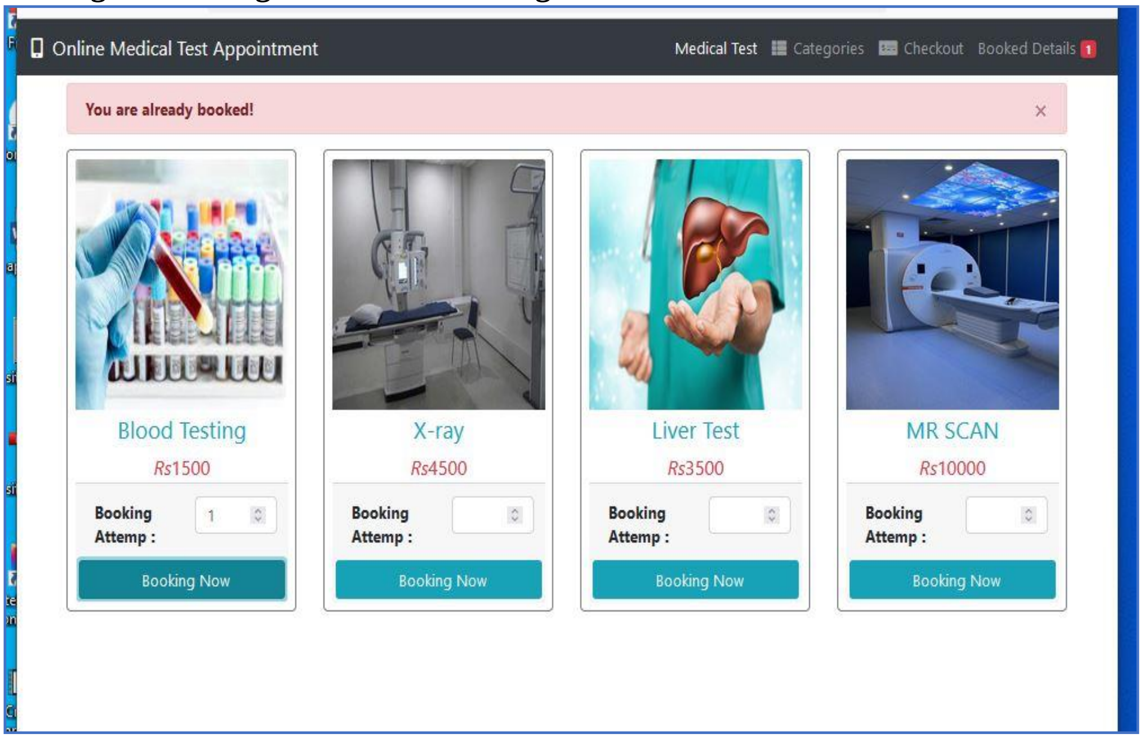
5.1.6 Add Booking



Your booking are added Then check and pay !

Figure 11:Add Booking

5.1.7 Again Booking Show Error Message

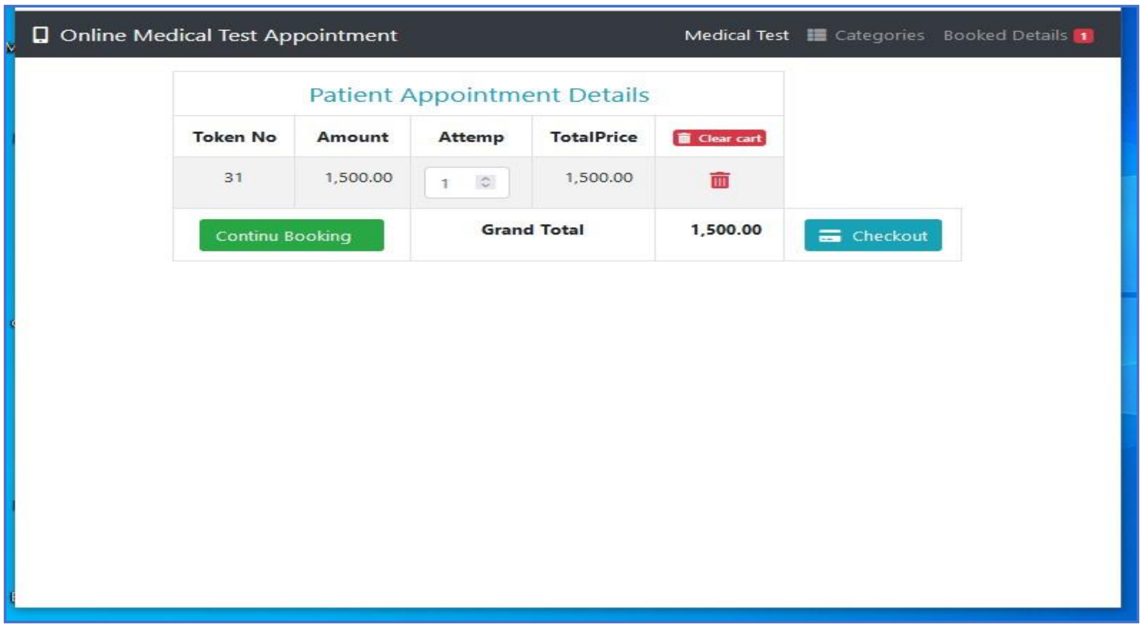


You are already booked!

Blood Testing	X-ray	Liver Test	MR SCAN
Rs1500	Rs4500	Rs3500	Rs10000
Booking Attemp : 1	Booking Attemp :	Booking Attemp :	Booking Attemp :

Figure 12: Again, Booking Show Error Message

5.1.8 Check Out

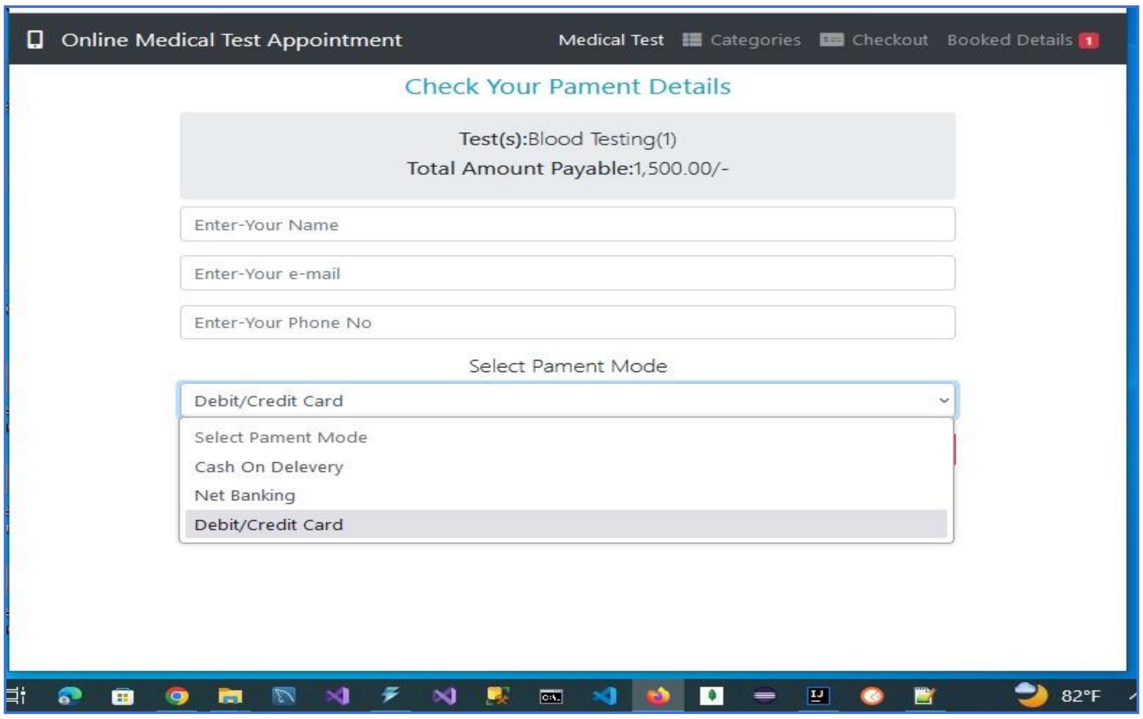


Patient Appointment Details

Token No	Amount	Attemp	TotalPrice	
31	1,500.00	1	1,500.00	
		Grand Total	1,500.00	

Figure 13: Check Out

5.1.9 Payment Details



Check Your Pament Details

Test(s):Blood Testing(1)
Total Amount Payable:1,500.00/-

Enter-Your Name
Enter-Your e-mail
Enter-Your Phone No

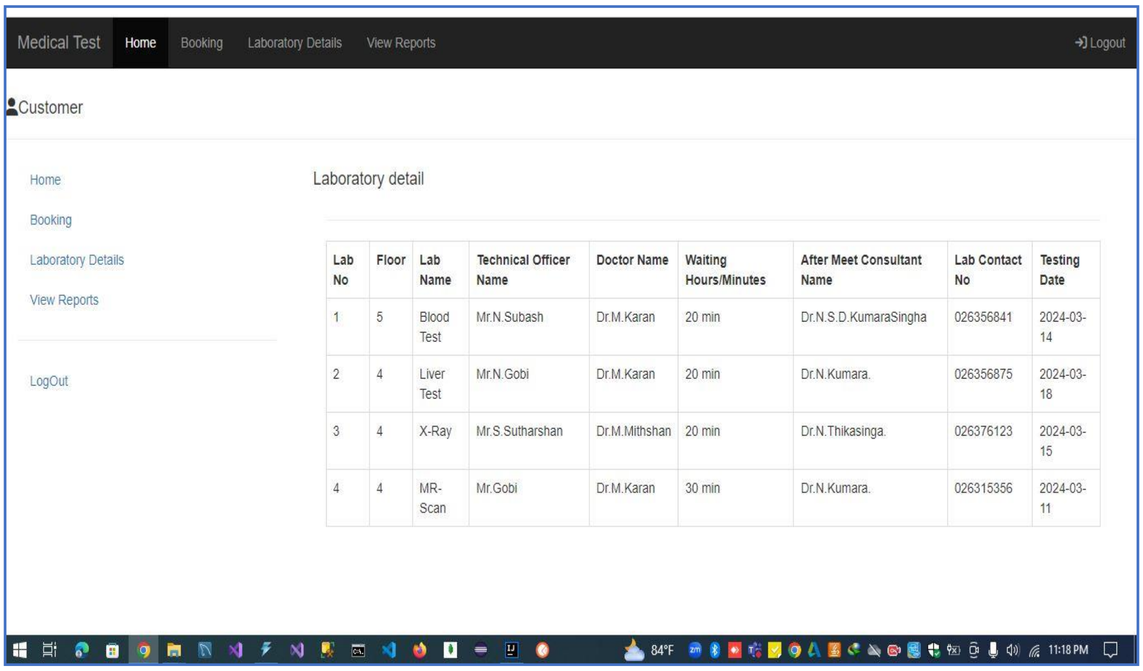
Select Pament Mode

Debit/Credit Card

- Select Pament Mode
- Cash On Delevery
- Net Banking
- Debit/Credit Card

Figure 14:Payment Details

5.1.10 Customer View Laboratory Details

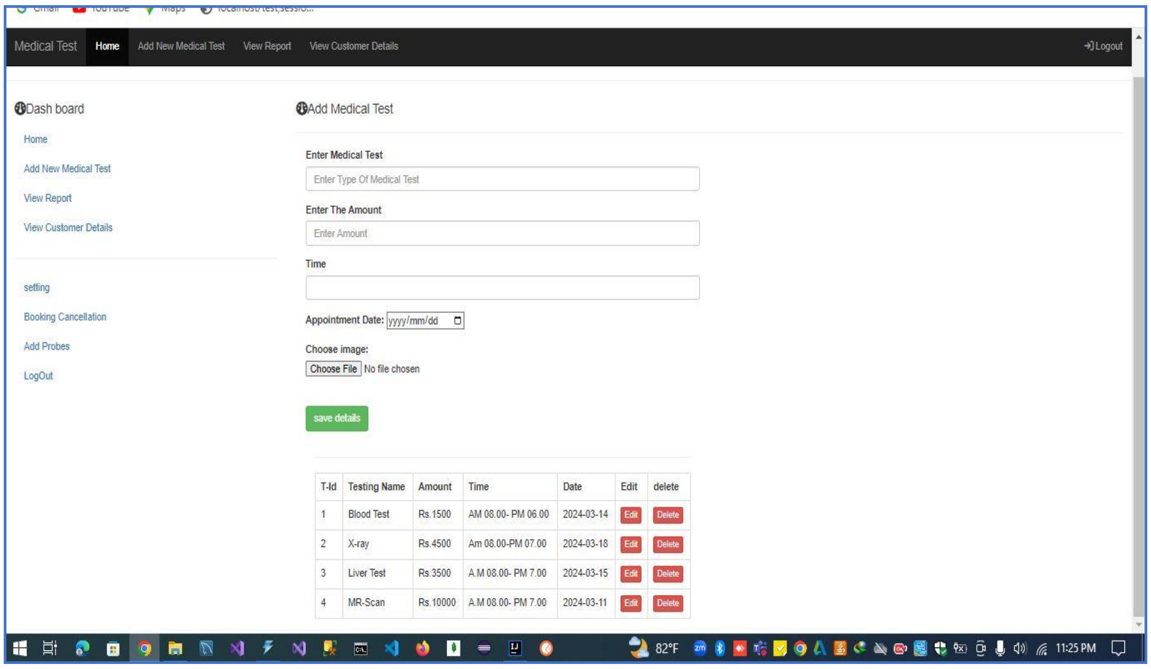


Laboratory detail

Lab No	Floor	Lab Name	Technical Officer Name	Doctor Name	Waiting Hours/Minutes	After Meet Consultant Name	Lab Contact No	Testing Date
1	5	Blood Test	Mr.N.Subash	Dr.M.Karan	20 min	Dr.N.S.D.KumaraSingha	026356841	2024-03-14
2	4	Liver Test	Mr.N.Gobi	Dr.M.Karan	20 min	Dr.N.Kumara.	026356875	2024-03-18
3	4	X-Ray	Mr.S.Sutharshan	Dr.M.Mithshan	20 min	Dr.N.Thikasinga.	026376123	2024-03-15
4	4	MR-Scan	Mr.Gobi	Dr.M.Karan	30 min	Dr.N.Kumara.	026315356	2024-03-11

Figure 15: Customer View Laboratory Details

5.1.11 Add Medical Test



Enter Medical Test

Enter Type Of Medical Test

Enter The Amount

Enter Amount

Time

Appointment Date: yyyy/mm/dd

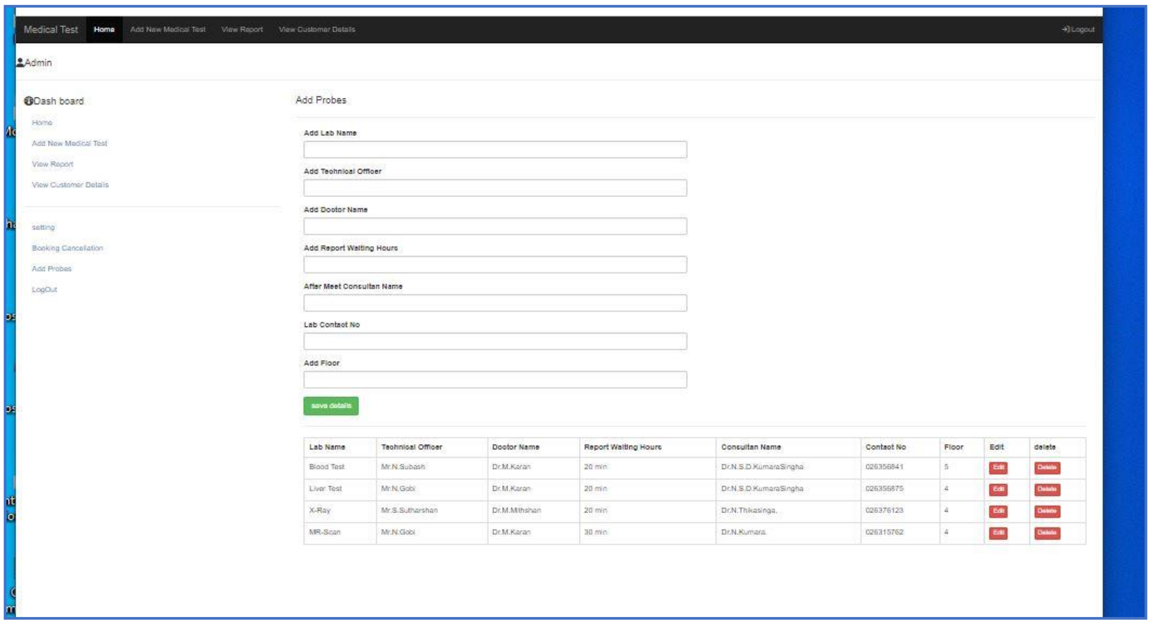
Choose image:

No file chosen

T-Id	Testing Name	Amount	Time	Date	Edit	delete
1	Blood Test	Rs.1500	AM 08.00- PM 06.00	2024-03-14	Edit	Delete
2	X-ray	Rs.4500	Am 08.00-PM 07.00	2024-03-18	Edit	Delete
3	Liver Test	Rs.3500	A.M 08.00- PM 7.00	2024-03-15	Edit	Delete
4	MR-Scan	Rs.10000	A.M 08.00- PM 7.00	2024-03-11	Edit	Delete

Figure 16:Add medical Test

5.1.12 Add Probs



Add Lab Name

Add Technical Officer

Add Doctor Name

Add Report Waiting Hours

After Meet Consultan Name

Lab Contact No

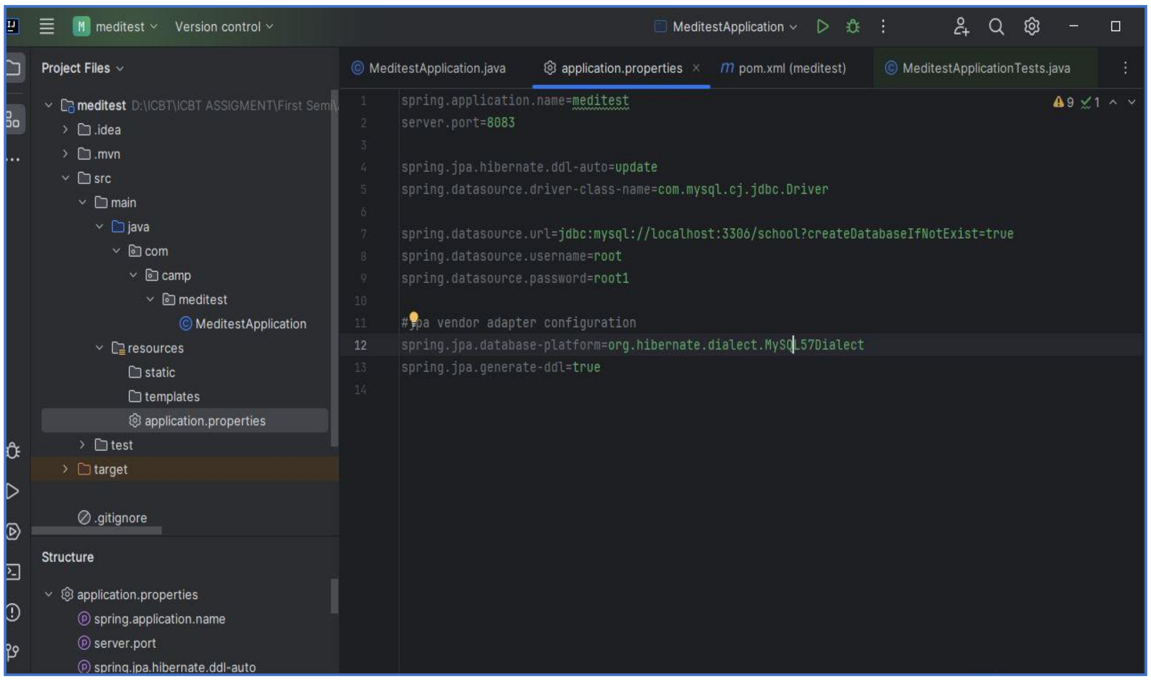
Add Floor

Lab Name	Technical Officer	Doctor Name	Report Waiting Hours	Consultan Name	Contact No	Floor	Edit	delete
Blood Test	Mr.N.Subash	Dr.M.Karan	20 min	Dr.N.S.D.KumaraSingha	026356841	5	Edit	Delete
Liver Test	Mr.N.Gobi	Dr.M.Karan	20 min	Dr.N.S.D.KumaraSingha	026356875	4	Edit	Delete
X-Ray	Mr.S.Sutharshan	Dr.M.Mithshan	20 min	Dr.N.Thikasinga.	026376123	4	Edit	Delete
MR-Scan	Mr.N.Gobi	Dr.M.Karan	30 min	Dr.N.Kumara.	026315762	4	Edit	Delete

Figure 17:Add Probs

5.2 Database Connectivity

5.2.1 Database Config



```
spring.application.name=meditest
server.port=8083

spring.jpa.hibernate.ddl-auto=update
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/school?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=root1

#jpa vendor adapter configuration
spring.jpa.database-platform=org.hibernate.dialect.MySQL57Dialect
spring.jpa.generate-ddl=true
```

Figure 18: database Config

5.2.2 Create Table add Medical Test

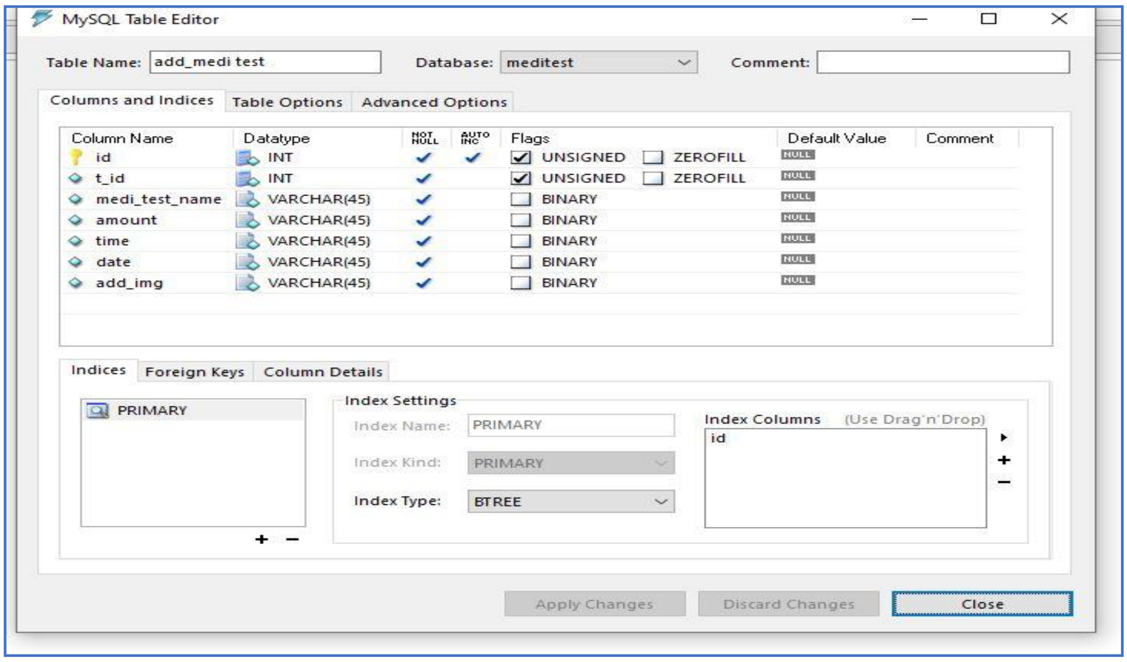


Table Name: add_medi test   Database: meditest   Comment:

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT	✓	✓	✓ UNSIGNED ☐ ZEROFILL	NULL	
t_id	INT	✓		✓ UNSIGNED ☐ ZEROFILL	NULL	
medi_test_name	VARCHAR(45)	✓		☐ BINARY	NULL	
amount	VARCHAR(45)	✓		☐ BINARY	NULL	
time	VARCHAR(45)	✓		☐ BINARY	NULL	
date	VARCHAR(45)	✓		☐ BINARY	NULL	
add_img	VARCHAR(45)	✓		☐ BINARY	NULL	

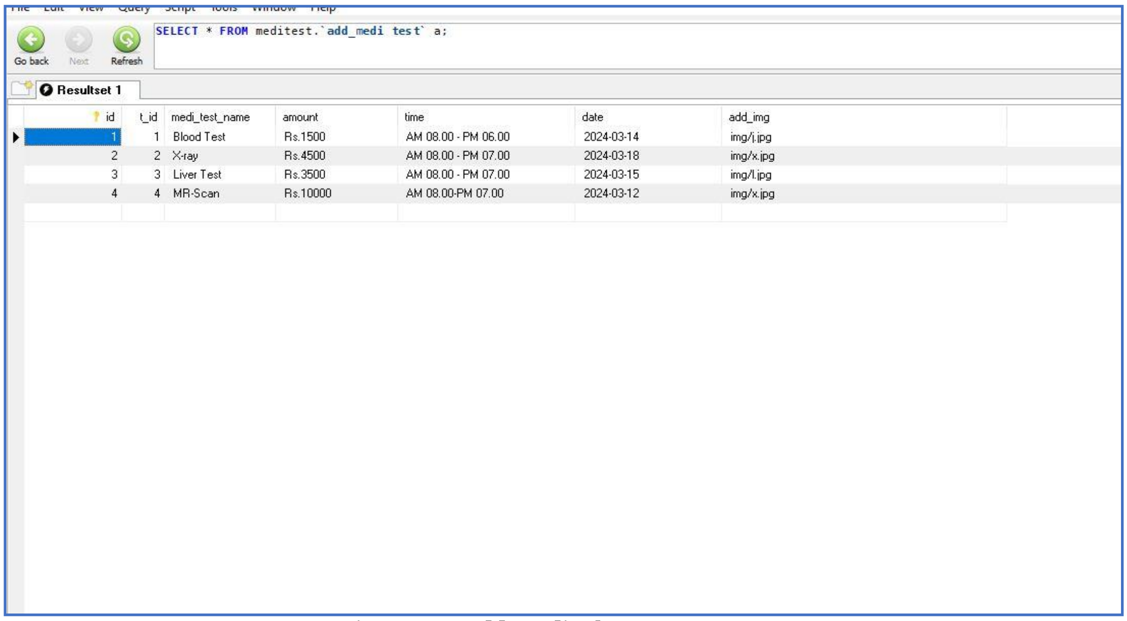
Indices: PRIMARY

Index Settings
Index Name: PRIMARY
Index Kind: PRIMARY
Index Type: BTREE

Index Columns (Use Drag'n'Drop): id

Figure 19: Create Table add Medical Test

5.2.3 Add Medical Test Data

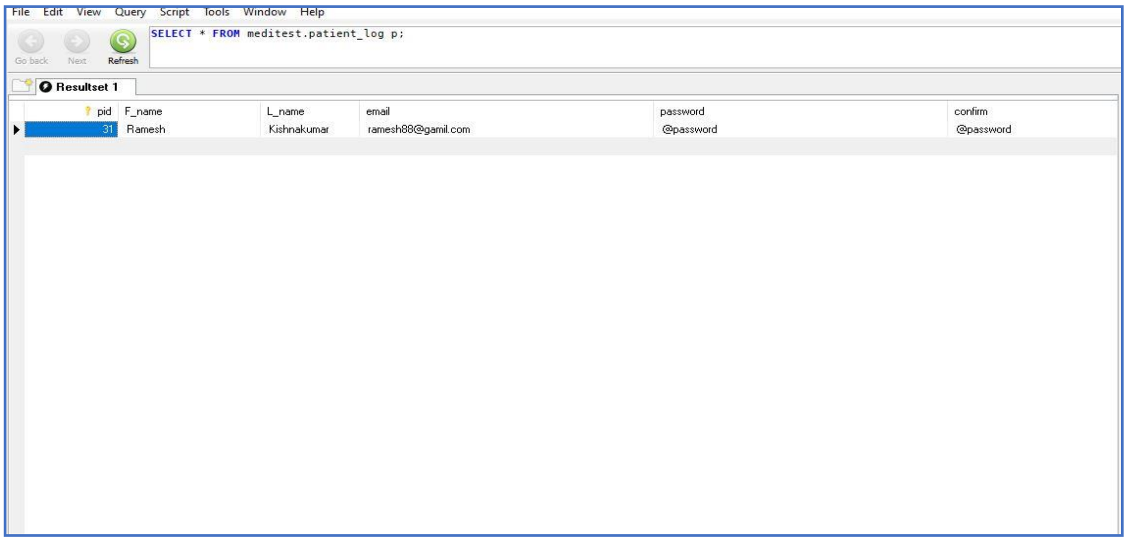


```
SELECT * FROM meditest.`add_medi test` a;
```

id	t_id	medi_test_name	amount	time	date	add_img
1	1	Blood Test	Rs.1500	AM 08.00 - PM 06.00	2024-03-14	img/j.jpg
2	2	X-ray	Rs.4500	AM 08.00 - PM 07.00	2024-03-18	img/x.jpg
3	3	Liver Test	Rs.3500	AM 08.00 - PM 07.00	2024-03-15	img/l.jpg
4	4	MR-Scan	Rs.10000	AM 08.00-PM 07.00	2024-03-12	img/x.jpg

Figure 20: Add Medical Test Data

5.2.4 Table User Login



```
SELECT * FROM meditest.patient_log p;
```

pid	F_name	L_name	email	password	confirm
31	Ramesh	Kishnakumar	ramesh88@gamil.com	@password	@password

Figure 21: Table User Login

6. Test Rationale, Test Plan, Test (Task E)

6.1 Test Rational

A software application that develops partial effective verification, validation and testing strategies is always a is challenging. Testing strategy strikes an important balance between cost and quality should be considered and appropriate transactions should be made depending on the particular scheme in this chapter, we discuss how and what to test for the existence of a Software Engineering Rationale (SER).

Rational components

1.Issue – A problem, concern, or question that requires discussion for the problem solving to proceed

2.Position – A statement or assertion that responds to an issue

3.Argument – Statement that support or object to a position.

4.Assumption – The basis for an argument

5.Decision – Resolving issues by selecting a position

6.2 Test Plane

A software test plan is a document that outlines the strategy, goals, and scope for a software release. This is an important document that provides clarity on the necessary tests that must be verified to ensure correct functionality of the software.

Type of test plan

1. Master test plan

A master test plan is a type of test plan that consists of multiple levels of tests. It includes a complete testing strategy.

2. Phase test plan

A phased test plan is a type of test plan that represents any phase of the test strategy. For example, list of tools, list of test cases, etc.

3. Testing type specific test pan

This type of test plan is designed for specific types of tests, especially for programs to conduct performance tests or safety tests, especially non-functional tests.

6.3 Test

Software testing is the process of determining the quality, functionality and performance of a software product before it is released. To perform software testing, testers manually interact with the software or run test scripts to find bugs and errors, ensuring that the software performs as expected. Software audits are conducted to immediately report whether business logic is met or if any gaps in requirements are missing.

Software testing is a very important part development life cycle. Without it, bugs that damage the system can negatively impact the bottom line and go undetected. Over time, the applications have sometimes appeared more problems and the testing programs have advanced with many new techniques and methods.

3.Important of software testing

Throughout the history of software development, we can recall many cases of software failure that resulted in serious losses and security issues affecting hundreds of customers. To the organization, For the enterprise, software testing is essential to deliver functional applications and avoid damage.

6.3.1 Below Laboratory Appointment System Testing

- **Login**

Test case 1: without user id and password then click the login button

shows an error message enter user name and password.

Test case 2: Enter user id without password then click the login button and throw the message enter your password.

Test case 3: Enter password without user id then click the login button shows the error message Enter the user id.

Test case 4: Enter the wrong user id and password then click the login button throw the message enter the correct user id and password.

Test case 5: Enter the correct user name and password then click the login button goes to the next page.

- **Add Medical Test**

Test case 1: Press the login button Without the medical test name, amount date and time throw a message full fill this.
Test case 2: Only enter the medical test name without, the amount, date and time then click the save button and throw and message please enter the amount, date and time.
Test case 3: Only enter the medical test amount without, name, date and time then click the save button and throw a message please enter an amount
Test case 4: Only enter the medical test date and time without the name, and the amount then click the save button and throw a message please enter the date and time.
Test case 5: Only enter the medical test date and time, and medical test name without, the amount then click the save button throw the message Please enter the medical test name.
Test case 6: Enter into the text box the medical test name, amount, date and time then click the login to button show message Data saved successfully.

7.Task F:

The Lab Appointment system is mostly customer base create the system. The system platform web-based. By using this method, the customer can get details such as the Tasting time, consultant name and type of test. After finishing the patient test can get the report on our website. Patient allows booking then online cart payment. This document provides a user guide and includes technical details for the development.

First Documentation

1. Getting and start

First, the patient goes to the Online Lab Appointment System after that patient registers. Customers can access the page after creating the customer platform.

2. Patient view type of medical test

The patient after login can view the medical test amount, and meet the consultant's date and time.

3. Booking for the clinic

Customer will make his confirmation after checking the appointment details.

4. Cancellation

In some cases, the order can be canceled after notifying the company if it is not required.

Sight of Architecture

1.The online Lab Appointment system is client server architecture

The client is a web browser that communicate with the server using HTTP/HTTPS Protocols.

2.The Technology stacks

Following the technology used in this system

Java, HTML, CSS, Spring Boot Frame Work, MySQL Database

3.API Documentation

The system provides postman APIs for communication between the front-end and back-end.

8.GitHub (Task G)

It is a web-based Git repository. This hosting service has clouds-based storage. GitHub provides distributing version control and share source code with management functionality while adding own benefits. This makes it easy to modifying using Git.

Additionally, GitHub repositories are open to the public. Developers from around the world can interact and contribute to one another's code, change or improve it, making GitHub a networking platform for web professionals. The process of interaction and contribution is also known as social coding.

8.1 Git hub benefits

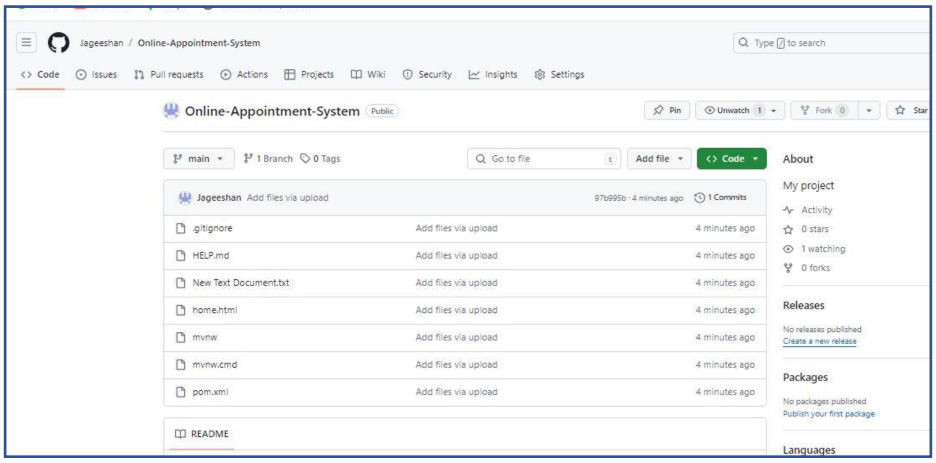
Version Control: This allows developers to easily replace previous versions of their code, collaborate on projects with others, and manage multiple branches of their code.

Collaboration: Developers can fork each other's repositories, create pull requests, and review code changes. GitHub also offers a number of features that make it easy to track and manage issues.

Code Review: GitHub makes it easy to review code changes before committing them to the main branch of a repository. It helps that the code is of high quality and meets all requirements.

Continuous Integration and Continuous Delivery (CI/CD): GitHub integrates with CI/CD tools to automate building, testing, and deployment of code. It helps developers produce their code quickly and reliably.

8.2 Git Hub page



Jageeshan Add files via upload 97b995b · 4 minutes ago 1 Commits

.gitignore	Add files via upload	4 minutes ago
HELP.md	Add files via upload	4 minutes ago
New Text Document.txt	Add files via upload	4 minutes ago
home.html	Add files via upload	4 minutes ago
mvnw	Add files via upload	4 minutes ago
mvnw.cmd	Add files via upload	4 minutes ago
pom.xml	Add files via upload	4 minutes ago

Figure 22: Git Hub page

Link of My GitHub: https://github.com/Jageeshan/Online-Appointment-System

9. Conclusion

The Lab Appointment System is created by a web-based platform that especially benefits healthcare providers and patients, including improved easy access, reduced waiting times, quick communication during the journey and simplified data analysis. The system can customers pre-book and same time get other tests and reports through this system. The user guide and technical based, documentation will support end-users and system developers to easily understand. how it accesses and it is improving in the next adaptation.

10.Referance

Creately (2023) *UML-Use Case Diagram*. Available at: https://www.softwaretestinghelp.com/use-case-diagram-tutorial/ (Accessed: 20 February 2024).

Bailey (2021) *Springboot-api*. Available at: https://hevodata.com/learn/spring-boot-rest-api/ (Accessed: 20 February 2024).

Elen (2023) *Design Patterns in OOAD*. Available at: https://stackify.com/introduction-to-design-patterns-in-software-development/ (Accessed: 29 February 2024).

Kumari, V. (2024) *Java Programming Examples*. Available at: https://www.c-sharpcorner.com/article/how-can-i-get-last-characters-of-a-string-in-java/ (Accessed: 29 February 2024).

R, S. (2024) *Online drawing software*. Available at: https://www.ilovephd.com/10-simple-online-drawing-tools-for-effective-thesis-diagrams/ (Accessed: 29 February 2024).

Software requirement specification (2023). Available at: https://www.scaler.com/topics/software-engineering/srs/ (Accessed: 20 February 2024).