"DATA STRUCTURES AND ALGORITHMS WITH JAVA"

Fitfinity Healthcare

Health for Everyone

Transforming Healthcare, Innovations and Challenges in Modern Medicine

Final Seminar

Date: March 1, 2024

Prepared by: Group 2, Fitfinity Team

FITFINITY TEAM

Aung Kaung Myat

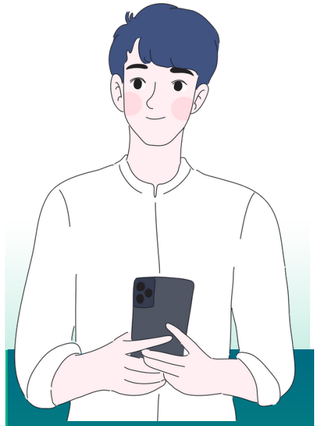
BMI Calculator
Doctor Dashboard
Powerpoint Design

Htet Paing Linn

Frontend
Patient Dashboard
Admin Dashboard
Doctors' Schedule System
Booking System
Flowchart Design

Pyae Linn

Feedback System
Contact Us
Captcha



Htet Myet Zaw

Patient Dashboard
Doctor Dashboard
Salary Management
History Management

Thaw Waiyan Hein

BMI Calculator
Tester

Fitfinity Healthcare

Health for Everyone

Objectives

- Optimizing Appointment Scheduling
- Efficient Patient Data Management
- Effective Doctor Management and Specialties
- Dynamic Department and Specialties Handling
- Appointment Data Management
- Doctor's Timetable Scheduling
- Scalability and Performance Optimization

Patient & Doctor

Doctors access patient records efficiently, while patients manage appointments easily

Administration

The admin team will handle doctors' accounts and salary and patients' data

Fitfinity Healthcare
Health for Everyone

Project Scope

The Java web application for healthcare strives to offer an effective, user-friendly platform for comprehensive healthcare facility management. Key features encompass patient online booking, data management, doctor and department oversight, appointment scheduling and doctor timetables.

Doctor

Medical History Management
View Appointment
View Income

Patient

Online Booking
BMI Calculation
Feedback & Rating

Admin

Human Resource Management and Data management

Doctor module

Data Management

Doctor Log-in, Patient Data Management

Appointment Management

Due to online patient booking, doctors can view all the appointment requests booked by the logged-in patients

Patient Medical History Management

Importing the Medical History and medicine dosage data for patients to the database

Doctor Income

According to number of appointments, our website will calculate the income of the doctor

Patient Module

based on online patient booking system

Signup for new patient

Crete new account and add patient data and medical report and storing database

Login patient

Login to existing account to access the current status of medical reports

View medical reports

Personal Information, Health Metrics, Medical History, Appointments, Emergency Contact Information

Booking system

Find specialty, select available doctor and make a appointment from department

Feedback System

Give Feedback to doctors and write a review of them

Admin Module

Doctor Management

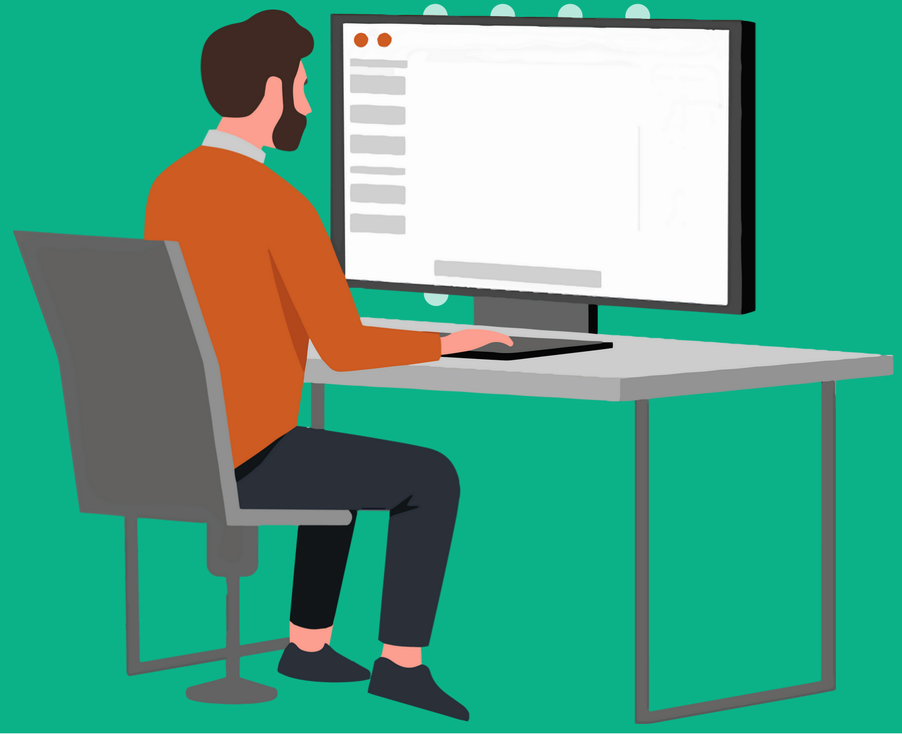
Admins can create and, delete doctor accounts

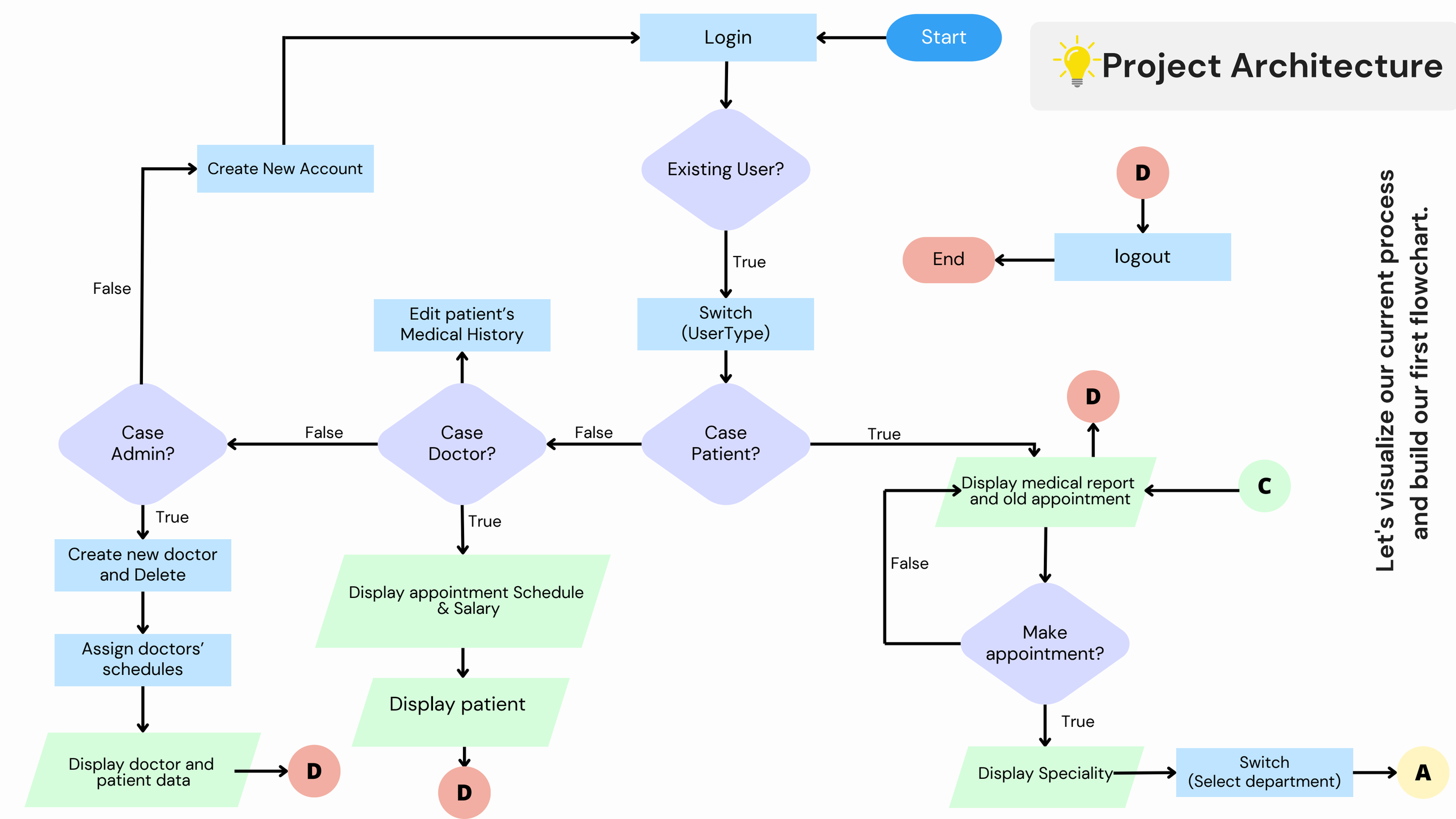
Schedule Management

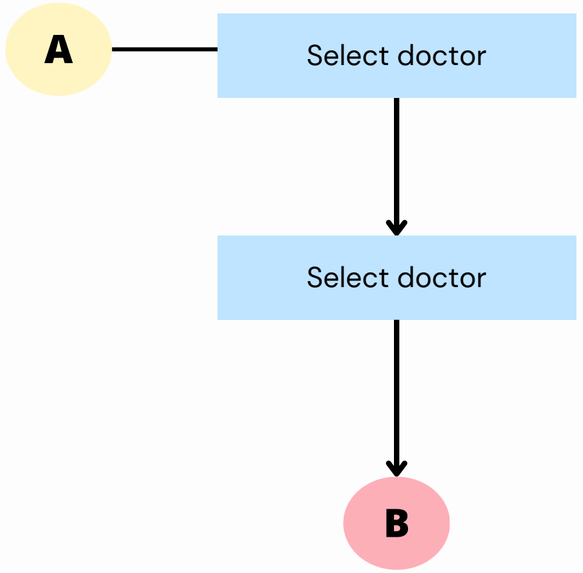
Admins can also arrange schedules for new doctors.

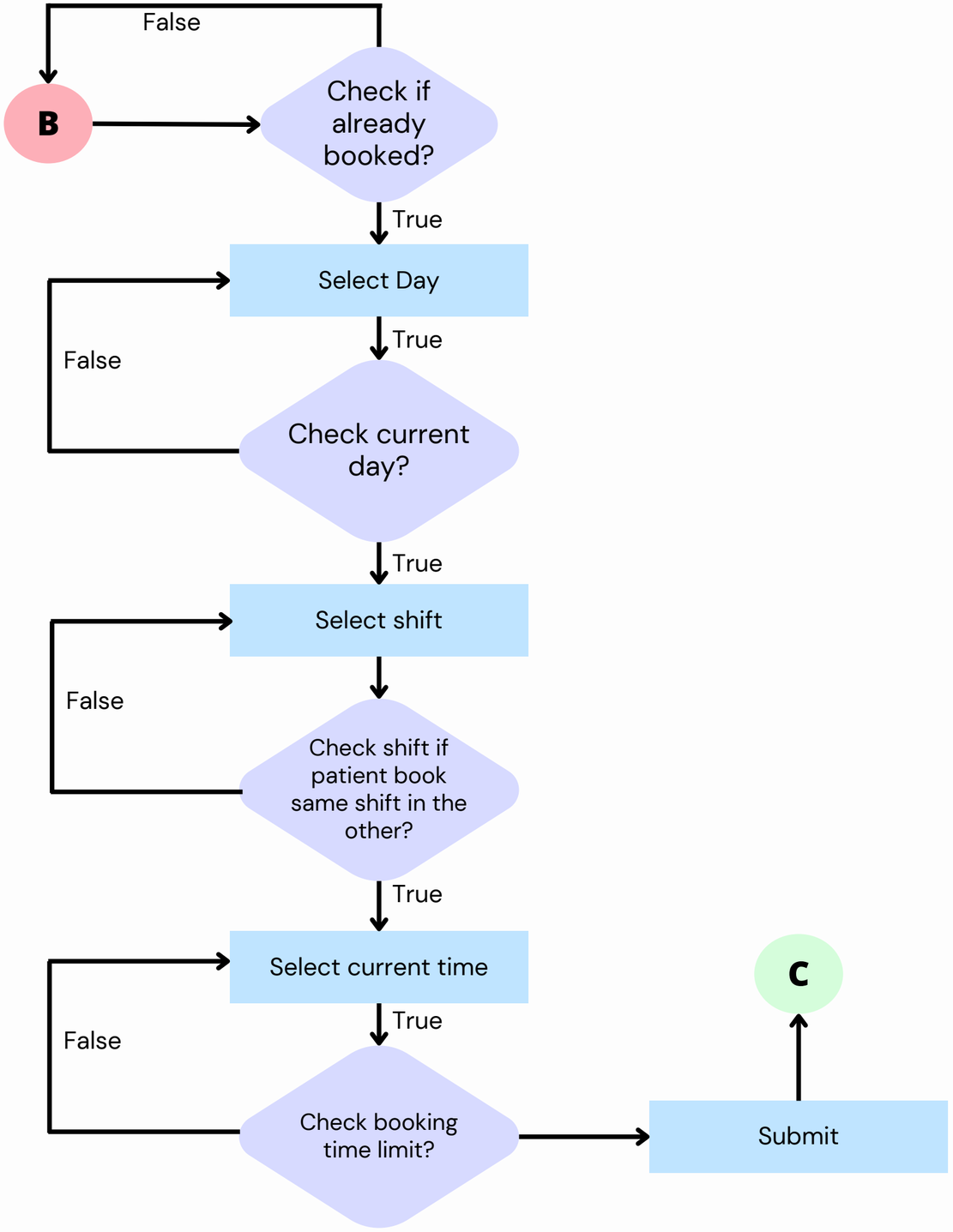
Salary Management

Admin Team will check one’s income and give the right amount to the doctors daily.




Project Architecture
Start
Login
Existing User?
Create New Account
False
True
Switch (UserType)
Case Patient?
False
Case Doctor?
False
Case Admin?
True
Edit patient's Medical History
True
Create new doctor and Delete
Assign doctors' schedules
Display doctor and patient data
D
Display appointment Schedule & Salary
Display patient
D
True
D
Display medical report and old appointment
C
False
Make appointment?
True
Display Speciality
Switch (Select department)
A
D
logout
End
Let's visualize our current process and build our first flowchart.


A
Select doctor
Select doctor
B


False
B
Check if already booked?
True
Select Day
True
False
Check current day?
True
Select shift
False
Check shift if patient book same shift in the other?
True
Select current time
True
False
Check booking time limit?
Submit
C

SYSTEM TECH STACK

Frontend	HTML5, CSS3, Bootstrap 5.3, Javascript, Font Awesome 5.15.4, JSP (Java Server Pages), JSON
Backend	Java J2EE (Servlet 4.0), MySQL connector j.8.3.0
Database	MySQL 5.0.37
Server	Apache Tomcat v9.0

TOOLS

ECLIPSE IDE FOR ENTERPRISE JAVA AND WEB DEVELOPERS 2024-03

REQUIREMENTS

“Any 64-bit operating system”
(Window 7 & above)

“At least 4Gb ram”
of memory storage

“High Speed Internet“
required

works on any broswer!!

Time Complexity

BMI calculator — O(1) Constant Time Complexity

BMI Formula: weight / ((height / 100) * (height / 100))

also based on Gender, Age, User, Selected Activity

Output: The calculated BMI, BMI category, and daily calorie needs are displayed to the user.

Doctor Searching — O(n) + O(m * k)

Department Function	O(n)	where n is the number of departments in the database.
Doctor Function	O(m * k)	where m is the number of doctors in the specified department. k is the length of the resulting HTML string for each doctor

Time Complexity

Booking Process

O(1) Constant Time Complexity

- Retrieving Doctor information — O(1)
- Displaying the doctor's information and schedule — O(m) [where m is the number of rows in the result set]
- *Book Appointment Nature* — O(1)

Running Time

0.01 ms

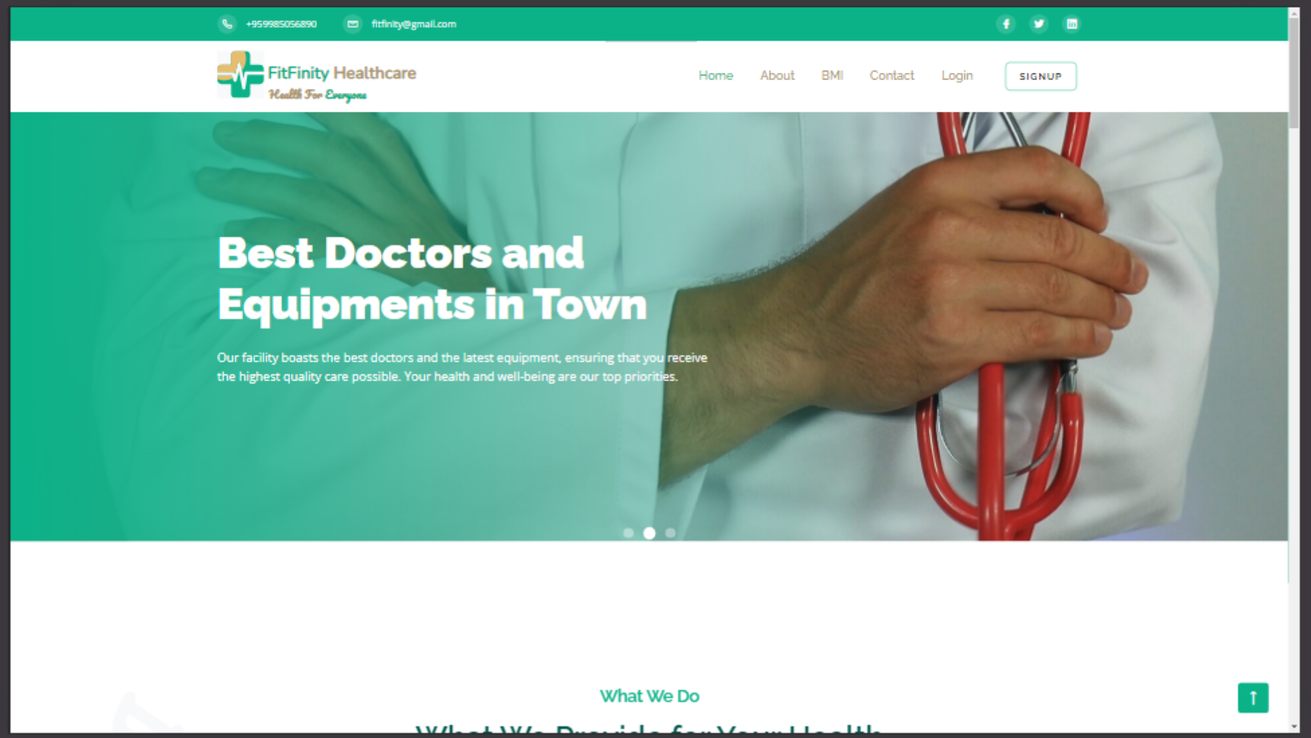
```
364  <%long endTime = System.nanoTime();
365
366  long responseTime = (endTime - startTime) / 1000000;
367  System.out.println("Response time: " + responseTime + " ms"); %>
368 </body>
369 </html>
```

Console × Problems Servers Terminal Data Source Explorer

Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-21\bin\javaw.exe (Feb 29, 2024, 11:02:35 PM) [pid: 3404]

```
        at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1794)
        at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:52)
        at org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1191)
        at org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659)
        at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
        at java.base/java.lang.Thread.run(Thread.java:1583)
Response time: 0 ms
```

Loading Time 569 ms



Dimensions: FullHD Laptop 1920 × 1080 40% No throttling

Elements Console Recorder Network 6 1

Preserve log Disable cache No throttling

upload Invert Hide data URLs Hide extension URLs

All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other

Blocked response cookies Blocked requests 3rd-party requests

100 ms 200 ms 300 ms 400 ms 500 ms 600 ms

Name Status Type Initiator Size Time Waterfall

0 / 44 requests 0 B / 84.1 kB transferred 0 B / 3.4 MB resources Finish: 569 ms DOMConten

PRECAUTIONS OVER LIMITATIONS

Fitfinity Healthcare
Health for Everyone

1

Multiple Appointments

Users cannot book multiple appointments with the same doctor in the same shift, preventing booking overflow.

2

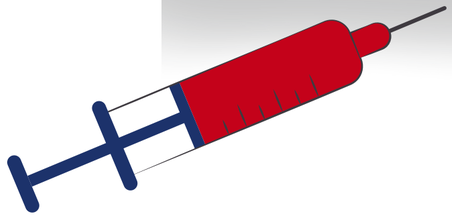
Complex Shifts

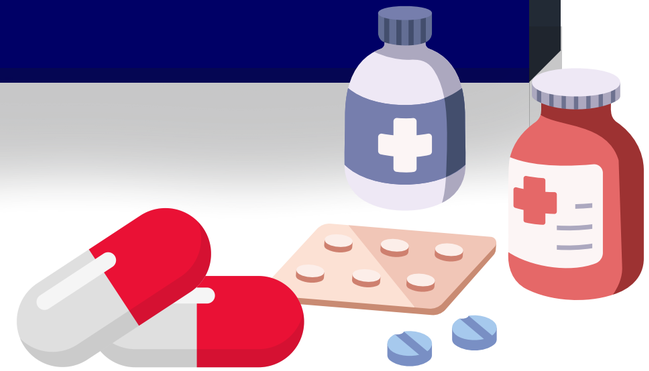
A user cannot make two or more appointments for the same shift, even with the different doctors.

3

Role complexion

Users have to pick a role (admin, patient, or doctor) to avoid conflicts, and validate roles effectively.





Fitfinity Healthcare

Health for Everyone

Facing Challenges

IDE Installation: Eclipse IDE is difficult to start up and configure

Version Incompatibility: Using different versions for the same project can lead to failure at running code

Integration: Combining all the separated codes into one place can be challenging

Expected Achievements & Success

- Improved Patient Care
- Better HR Management
- Long-term Health Monitoring
- Partnerships and Collaborations
- Remote Healthcare Servies
- Patient Satisfaction & Engagement

Fitfinity Healthcare

Health for Everyone

Group II - Section A, Batch 9

THANKS FOR WATCHING

Fitfinity Healthcare System