American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science &Technology (FST)

Summer 22 23

Section: E

Software Quality Assurance and Testing

HOSPITAL MANAGEMENT SYSTEM

A Report submitted
By

SN	Student Name	Student ID
1	MD. Sydur Rahman Razi	20-42155-1

Checked By Industry Personnel

Name: Sheikh Aysha Khatun

Designation: SQA Engineer

Company: TechnoNext Limited

Sign:

Date:

Software Test Plan

for

HOSPITAL MANAGEMENT SYSTEM

Version 1.0 approved.

Prepared by MD. Sydur Rahman

American International University – Bangladesh

12 August 2023

Checked By Industry Personnel

Name: Sheikh Aysha Khatun

Designation: SQA Engineer

Company: TechnoNext Limited

Sign:

Date:

Table of Contents

Revision History

Revision	Date	Updated by	Update Comments
0.1	2023.07.07	MD. Sydur Rahman Razi	First Draft
0.2	2023.07.14	MD. Sydur Rahman Razi	Second Draft
0.3	2023.07.21	MD. Sydur Rahman Razi	Third Draft
0.4	2023.08.05	MD. Sydur Rahman Razi	Fourth Draft

1. TEST PLAN IDENTIFIER:RS-MTP01.3

2. REFERENCES

[1]. Mahmoud, Maha. "A Case Study on Hospital Management System."
[2]. Siddique MA. Hospital Management System (Doctoral dissertation, East West University).
[3]. Ali, Syed Waqas, Qazi Arbab Ahmed, and Imran Shafi. "Process to enhance the quality of software requirement specification document." 2018 International Conference on Engineering and Emerging Technologies (ICEET). IEEE, 2018.

3. INTRODUCTION

Background to the Problem

The Hospital Management System Project addresses the critical need for an integrated and efficient approach to managing healthcare institutions. In the complex landscape of modern medical facilities, there exists a pressing challenge of optimizing patient care, resource utilization, and administrative tasks. This project aims to revolutionize healthcare administration by providing a comprehensive digital platform that streamlines every facet of hospital operations. Within the bustling environment of a hospital, various departments, including patient registration, appointment scheduling, medical records, billing, pharmacy, and inventory management, often operate in isolation, leading to inefficiencies, errors, and communication gaps. The Hospital Management System Project seeks to bridge these gaps by offering a unified system where patient information, treatment histories, diagnostic results, and billing details are seamlessly integrated. This empowers healthcare professionals to make well-informed decisions promptly, enhances patient experiences through swift and accurate services, and facilitates robust resource planning. Furthermore, the project's digital architecture aims to enhance data security and compliance with privacy regulations, assuring patients that their sensitive information is safeguarded. By aligning the efforts of medical staff, administrators, and IT specialists, the Hospital Management System Project is poised to elevate healthcare standards, elevate operational efficiency, and ultimately contribute to better patient outcomes.

The root cause of the problem addressed by the Hospital Management System Project is the lack of integrated and streamlined processes within healthcare institutions. Traditional manual and disjointed methods of managing patient information, appointments, records, and billing result in inefficiencies, errors, and communication breakdowns. These fragmented approaches hinder effective decision-making, compromise patient care, and impede resource optimization. The project aims to tackle this issue by providing a cohesive digital solution that centralizes data, promotes seamless collaboration, and enhances overall hospital management, thereby mitigating the root cause of the existing challenges.
The problem addressed by the Hospital Management System Project holds immense importance due to its far-reaching impact on healthcare delivery and patient well-being. Inefficient management practices in hospitals lead to delayed patient care, increased medical errors, and

heightened operational costs. These consequences not only jeopardize patient safety but also strain healthcare resources and staff morale. By centralizing and automating processes, the project ensures accurate and timely access to critical patient information, optimizing treatment plans and reducing medical errors. It also empowers healthcare professionals to allocate resources effectively, minimizing wastage and enhancing cost-efficiency. Moreover, improved coordination among departments through the system enhances patient satisfaction by streamlining appointment scheduling, reducing waiting times, and facilitating smoother billing processes.

Ultimately, the successful implementation of the Hospital Management System Project has the potential to revolutionize healthcare service delivery, elevate patient outcomes, and contribute to a more sustainable and responsive healthcare system. The urgency of this problem stems from its direct impact on both patient care quality and the overall efficiency of healthcare institutions, making its resolution a paramount concern for healthcare providers, administrators, and patients alike.

Solution to the Problem

The proposed Hospital Management System software will be a web-based application developed using modern software development technologies and methodologies. The solution for the Hospital Management System Project entails the development and implementation of a comprehensive, integrated digital platform. This platform will centralize patient information, appointment scheduling, medical records, billing, pharmacy, and inventory management, facilitating seamless communication and coordination among hospital departments.
This solution is particularly appropriate for several reasons:

1. Streamlined Operations: The integrated platform will eliminate redundancy and improve workflow efficiency. It allows healthcare professionals to access real-time patient data, enabling accurate diagnoses and timely treatment decisions.

2. Enhanced Patient Care: With instant access to patient histories, allergies, and treatment plans, medical staff can provide personalized care and reduce medical errors.

3. Resource Optimization: The system will facilitate efficient allocation of resources, minimizing waste and reducing operational costs.

4. Improved Patient Experience: Patients will benefit from quicker appointment scheduling, reduced waiting times, and smoother billing processes, leading to higher satisfaction levels.

5. Data Security and Compliance: The digital platform can be designed with robust security measures to protect patient confidentiality and adhere to data protection regulations.

The specified software is a Hospital Management System (HMS), designed to streamline and integrate various aspects of healthcare administration. Its purpose is to centralize patient information, appointments, medical records, billing, pharmacy, and inventory management, fostering efficient communication between departments. The HMS aims to enhance patient care by providing real-time data access, reducing medical errors, optimizing resource allocation, and improving the overall patient experience. Its objectives include seamless workflow automation, data security, and comprehensive reporting, ultimately leading to elevated healthcare standards, cost-effectiveness, and increased patient satisfaction.

Numerous studies have highlighted the significance of integrated Hospital Management Systems (HMS) in improving healthcare efficiency and patient outcomes. Existing software solutions like Epic Systems' "EpicCare," Cerner Corporation's "Power Chart," and Allscripts' "Sunrise Clinical Manager" offer comprehensive HMS functionalities. These systems centralize patient data, streamline workflows, and enhance communication among hospital departments. They enable precise diagnoses, reduce medical errors, optimize resource allocation, and improve patient experiences. While effective, customization, user training, and cost considerations are vital factors for the successful implementation and utilization of such solutions.

4. REQUEIREMNT SPECIFICATION

4.1 System Features

1. Sign Up

Functional Requirements

I. Verify the user’s identity through a secure and reliable process.
II. Gather essential information for user account creation and platform access authorization.

Priority Level: High

Preconditions: User has valid first and last name, ID, Email, Address and Phone

2. Login

Functional Requirements

I. Ensure user rules and permissions to restrict access to authorized resources and features within the platform.
II. Login requires Email and Password.

Priority Level: High

Preconditions: User has valid Email and Password.

3. Update Info

Functional Requirements

I. Verify user input to maintain data accuracy and integrity.
II. Store the updated data in the database or storage system to maintain data consistency and ensure its availability.

Priority Level: High

Preconditions: User has valid data.

4. Doctors Data

Functional Requirements

I. Limit access to doctors' data to authorized personnel exclusively.
II. Monitor and log all activities conducted on doctors' data to ensure accountability and facilitate traceability.

Priority Level: Medium

Preconditions: Database should consist of all doctors' information.

5. Patients Data

Functional Requirements

I. Validate user input to ensure accuracy and integrity.
II. Handle tourist data securely and protect user privacy to build trust and prevent data breaches.

Priority Level: Medium.

Preconditions: Database should consist of all patients' information

6. Add Doctor

Functional Requirements

I. Verify the doctor's identity through a secure and reliable process.
II. Gather essential information for doctor's account creation and platform access authorization.

Priority Level: High

Preconditions: Doctor has valid first and last name, ID, Email, Designation, phone, and salary.

7. Search Doctor

Functional Requirements

I. Enable users to search for doctors based on various criteria such as specialization, availability, and location.

Priority Level: Medium

Preconditions: Doctor has valid ID to search.

8. Delete Doctor

Functional Requirements

I. Delete a doctor profile.

Priority Level: Medium

Preconditions: Doctor has valid ID to delete.

9. Search Patient

Functional Requirements

I. Enable users to search for Patient based on various criteria such as specialization, availability, and location.

Priority Level: Medium

Preconditions: Patient has valid ID to search.

4.2 System Quality Attributes

Usability: Like the previous context, usability relates to how intuitively users can navigate and interact with the Hospital Management System. It includes user-friendly interfaces, clear instructions, and logical workflows for tasks such as patient registration, appointment scheduling, and billing.

Maintainability: This attribute pertains to the ease with which the Hospital Management System can be maintained, updated, and enhanced over time. It includes well-structured code, documentation, and support resources for troubleshooting and system improvements.

Scalability: This attribute refers to the capability of the Hospital Management System to handle a growing number of users, patient records, and transactions without significant performance degradation. It includes factors such as database optimization, load balancing, and the ability to scale resources as needed to meet increasing demands.

Reliability: This attribute emphasizes the consistent and error-free operation of the system. It encompasses factors like minimizing system downtime, accurate data storage and retrieval, and the ability to recover from failures.

Availability: This attribute relates to the system's ability to remain operational and accessible to users consistently. It involves factors like fault tolerance, disaster recovery mechanisms, and redundancy to ensure uninterrupted access to critical patient data and administrative functions.

Interoperability: This attribute concerns the Hospital Management System's capacity to seamlessly integrate and exchange data with other healthcare systems, such as electronic health record systems or medical equipment. It involves adherence to standardized communication protocols and data formats.

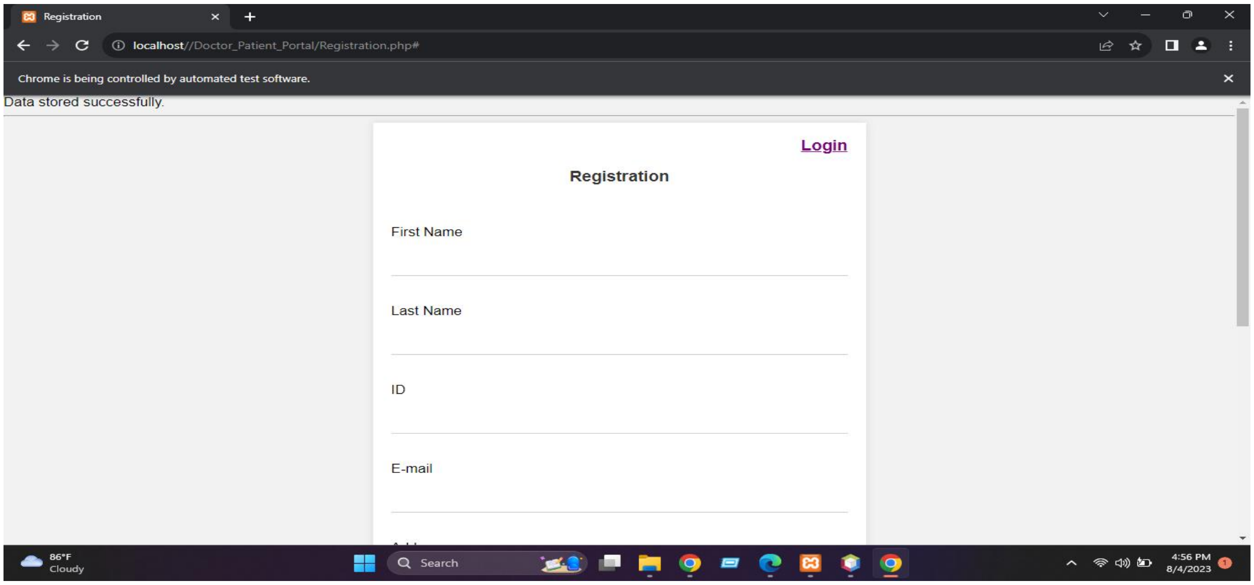
Adaptability: This attribute reflects the system's ability to accommodate changes and updates, including regulatory changes in the healthcare industry or modifications to administrative processes. It includes flexible configuration options, modular design, and ease of implementing new features.

Performance: This attribute pertains to the system's responsiveness and efficiency in executing tasks. It encompasses factors like quick retrieval of patient records, minimal processing delays, and efficient handling of concurrent user requests.

Auditability: This attribute refers to the system's capability to log and track user activities and changes made to patient records, ensuring accountability and compliance with regulations. It involves comprehensive audit trials and user activity monitoring.

Security: This attribute focuses on safeguarding patient data and sensitive information from unauthorized access or breaches. It involves robust user authentication, data encryption, access controls, and compliance with healthcare data protection standards.

4.3 System Interface

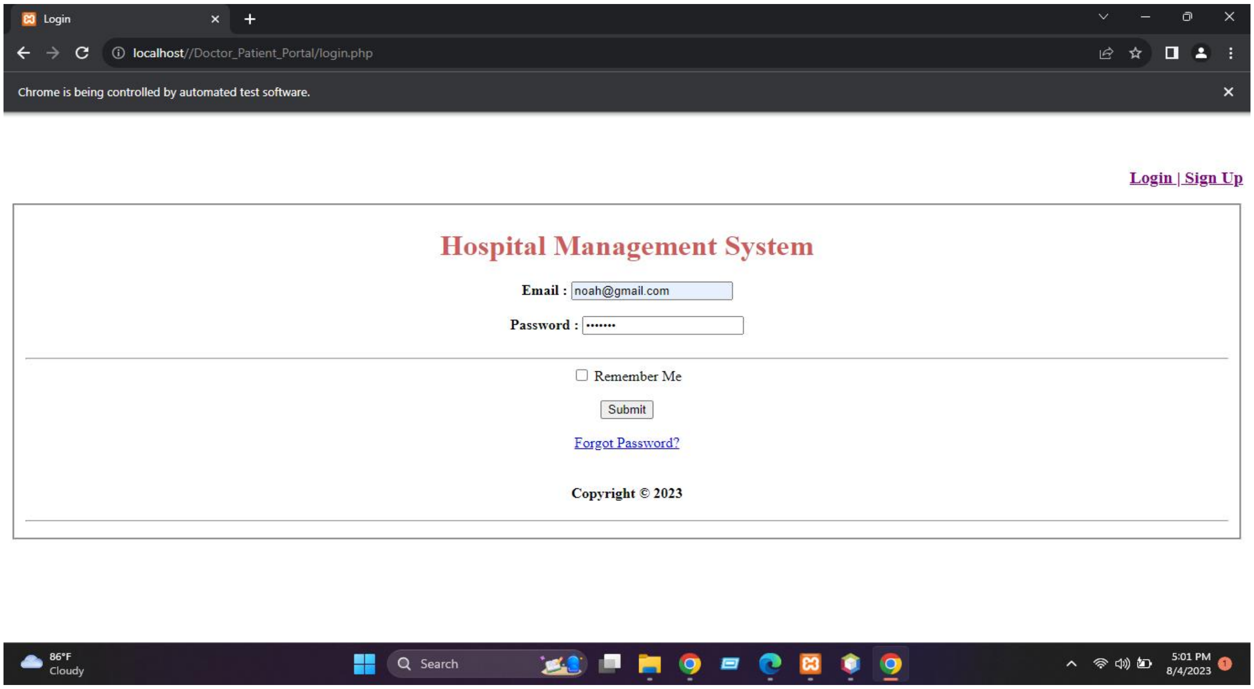


localhost//Doctor_Patient_Portal/Registration.php#

Chrome is being controlled by automated test software.

Data stored successfully.

This is the interface of Registration Module



localhost//Doctor_Patient_Portal/login.php

Chrome is being controlled by automated test software.

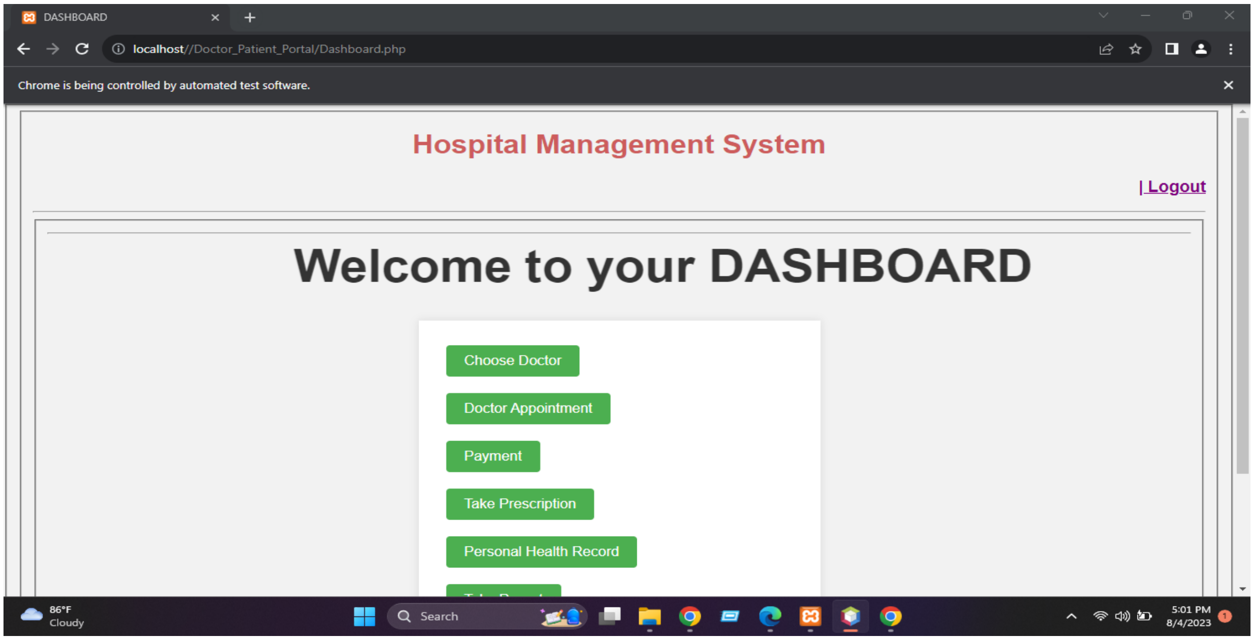
Hospital Management System

Email : noah@gmail.com

Password : •••••••

Copyright © 2023

This is the interface of Login Module



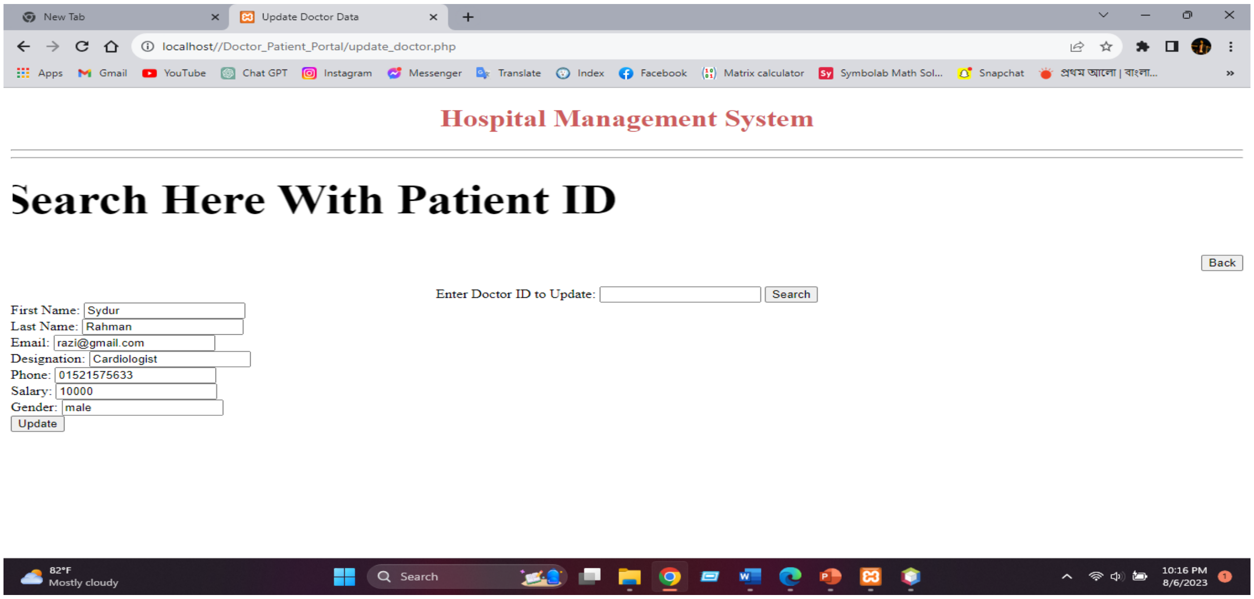
Chrome is being controlled by automated test software.

Hospital Management System

| Logout

Welcome to your DASHBOARD

This is the interface of Patient Dashboard Module



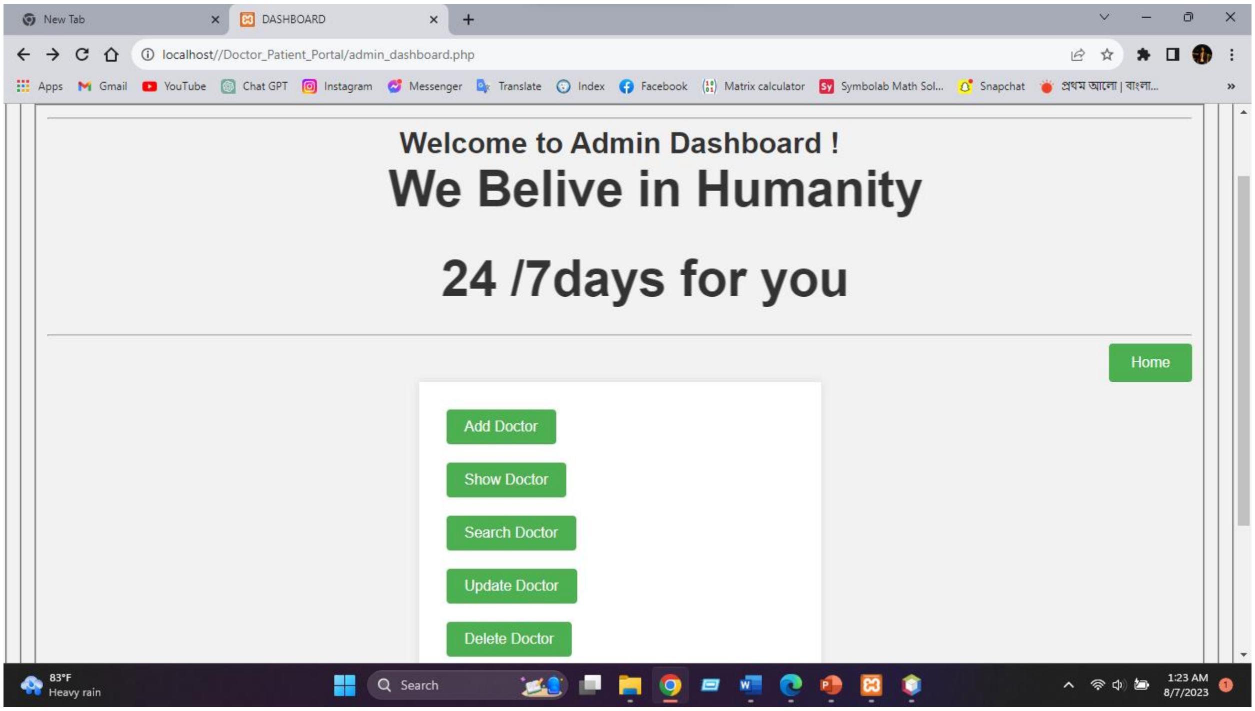
Hospital Management System

Search Here With Patient ID

Enter Doctor ID to Update:

First Name: Sydur
Last Name: Rahman
Email: razi@gmail.com
Designation: Cardiologist
Phone: 01521575633
Salary: 10000
Gender: male

This is the interface of Update Doctor Module

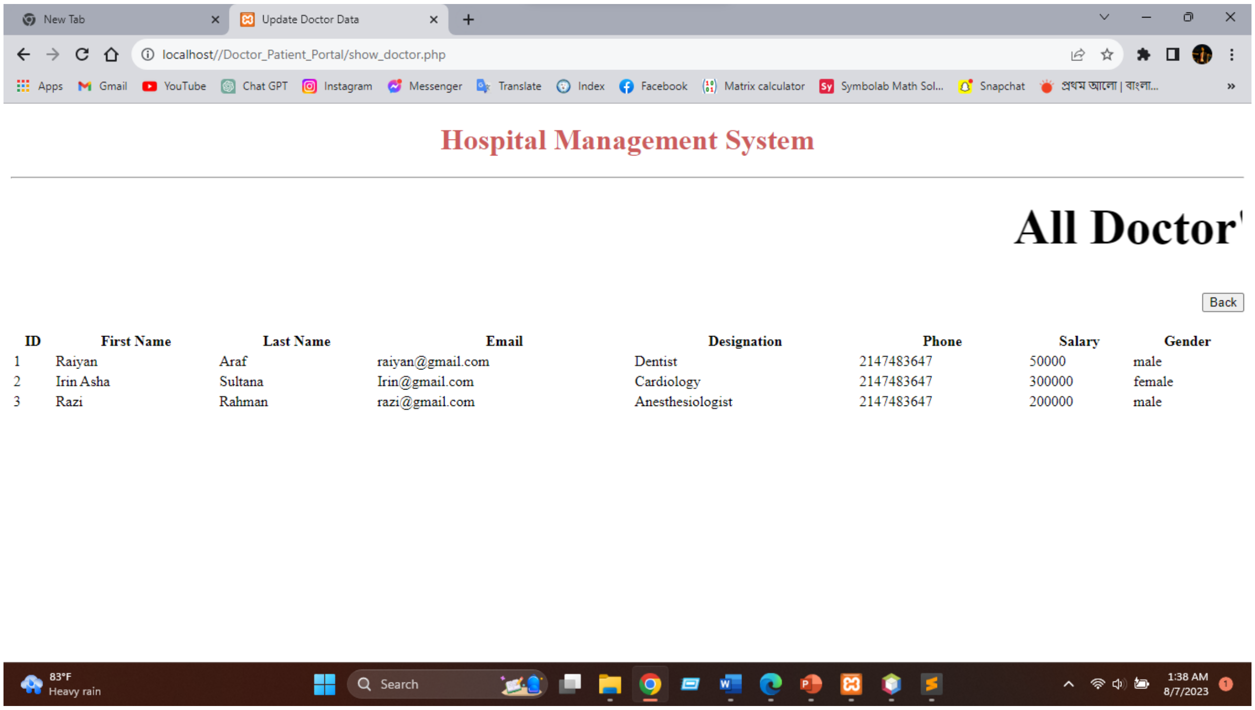


Welcome to Admin Dashboard !

We Belive in Humanity

24 /7days for you

This is the interface of Admin Dashboard Module

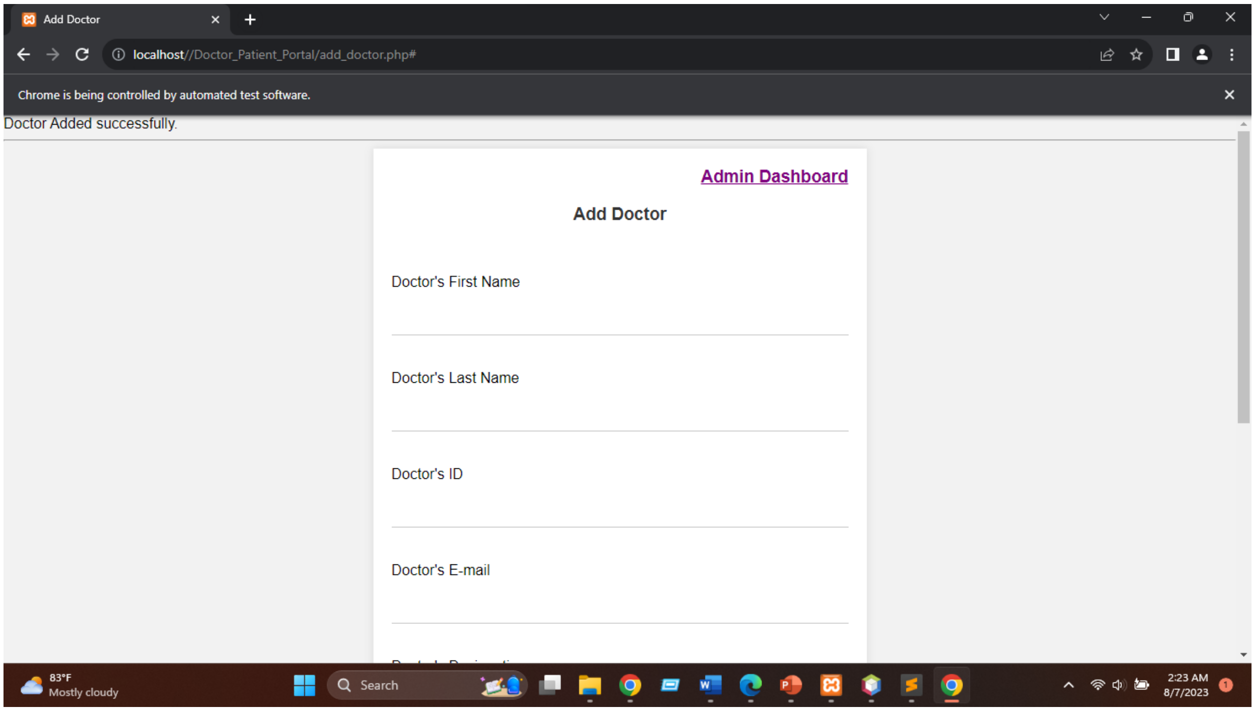


Hospital Management System

All Doctor'

ID	First Name	Last Name	Email	Designation	Phone	Salary	Gender
1	Raiyan	Araf	raiyan@gmail.com	Dentist	2147483647	50000	male
2	Irin Asha	Sultana	Irin@gmail.com	Cardiology	2147483647	300000	female
3	Razi	Rahman	razi@gmail.com	Anesthesiologist	2147483647	200000	male

This is the interface of Show All Doctor Module


Add Doctor
localhost//Doctor_Patient_Portal/add_doctor.php#
Chrome is being controlled by automated test software.
83°F
Mostly cloudy
Search
2:23 AM
8/7/2023

Doctor Added successfully.

Admin Dashboard

Add Doctor

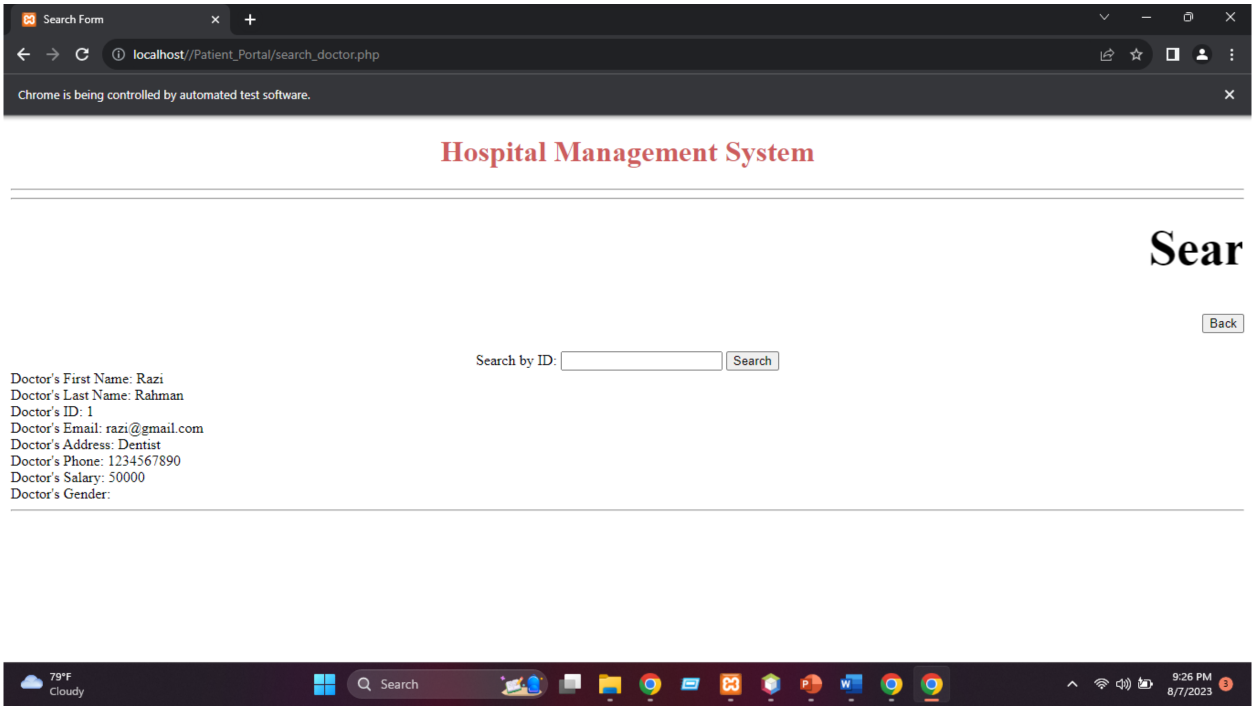
Doctor's First Name

Doctor's Last Name

Doctor's ID

Doctor's E-mail

This is the interface of Add Doctor Module


Search Form
localhost//Patient_Portal/search_doctor.php
Chrome is being controlled by automated test software.
79°F
Cloudy
Search
9:26 PM
8/7/2023

Hospital Management System

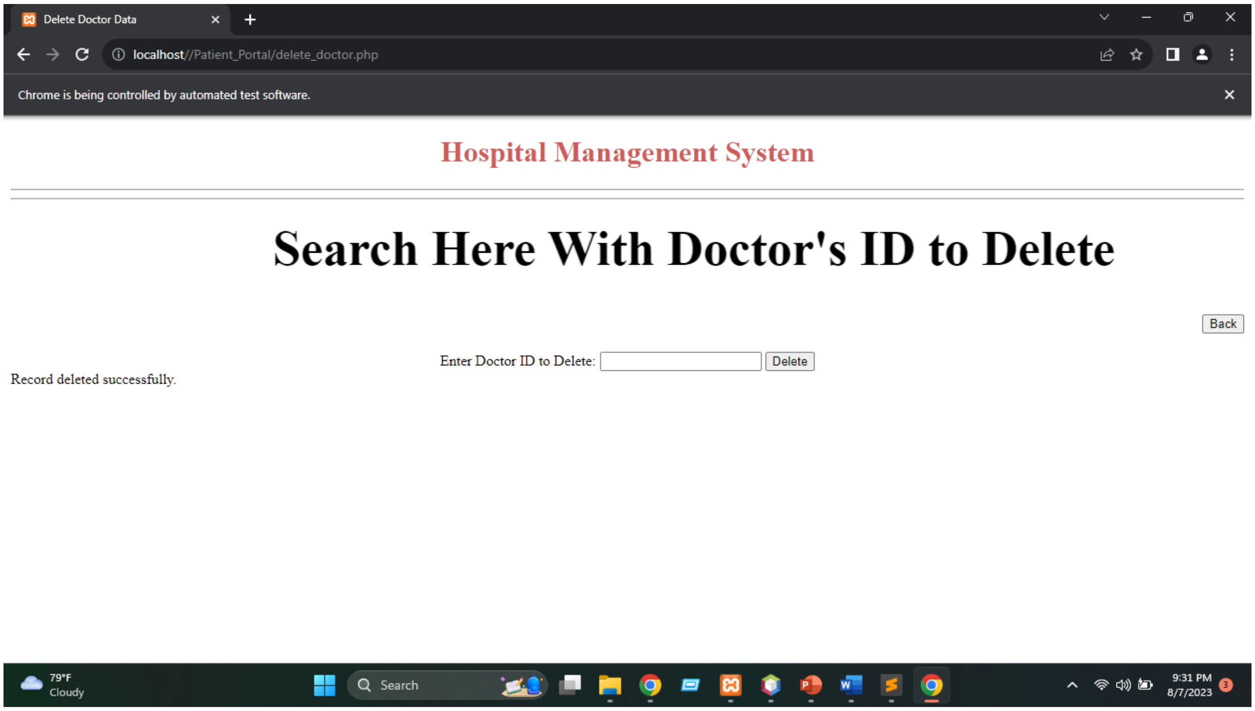
Sear

Back

Search by ID: Search

Doctor's First Name: Razi
Doctor's Last Name: Rahman
Doctor's ID: 1
Doctor's Email: razi@gmail.com
Doctor's Address: Dentist
Doctor's Phone: 1234567890
Doctor's Salary: 50000
Doctor's Gender:

This is the interface of Search Doctor Module



Chrome is being controlled by automated test software.

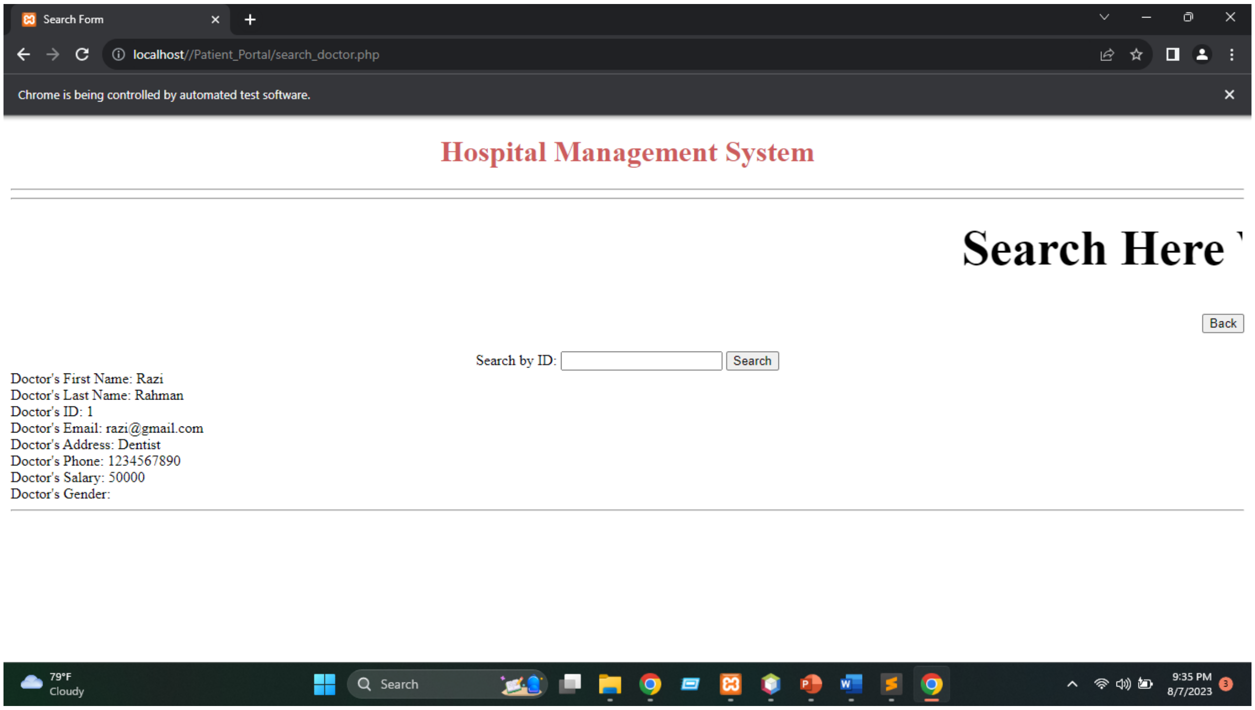
Hospital Management System

Search Here With Doctor's ID to Delete

Enter Doctor ID to Delete:

Record deleted successfully.

This is the interface of Delete Doctor Module



Chrome is being controlled by automated test software.

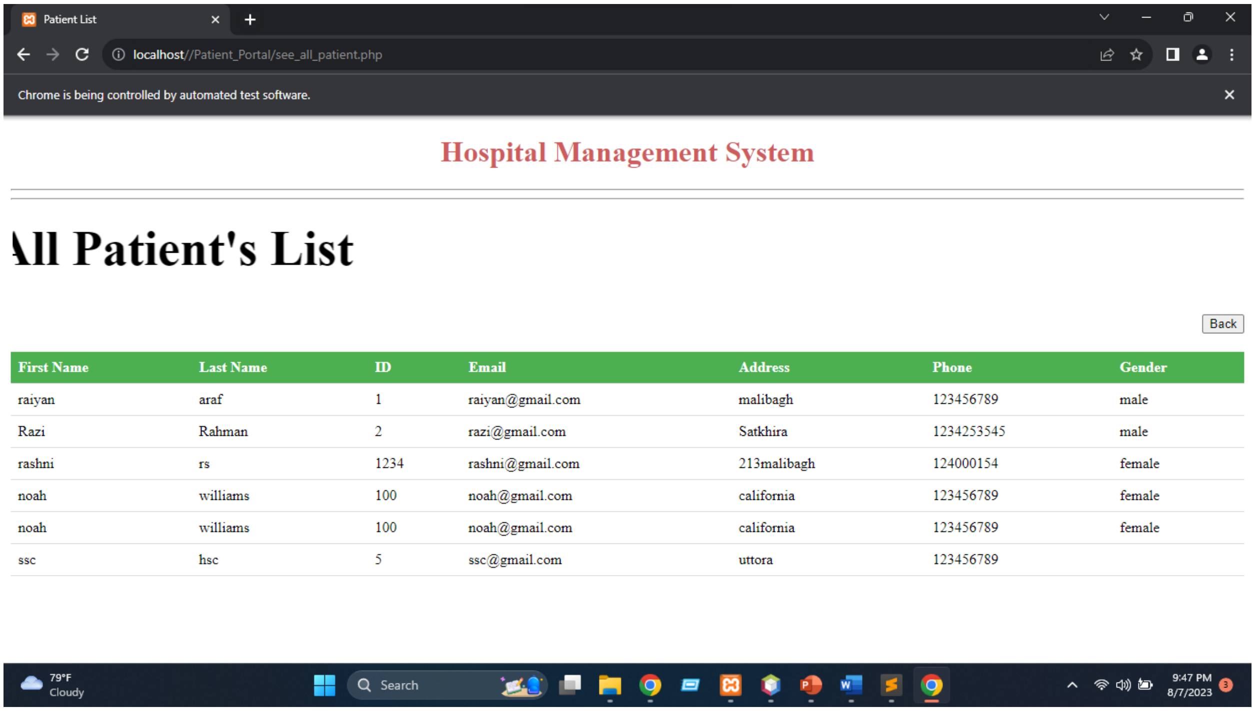
Hospital Management System

Search Here

Search by ID:

Doctor's First Name: Razi
Doctor's Last Name: Rahman
Doctor's ID: 1
Doctor's Email: razi@gmail.com
Doctor's Address: Dentist
Doctor's Phone: 1234567890
Doctor's Salary: 50000
Doctor's Gender:

This is the interface of Search Doctor Module


Patient List
localhost//Patient_Portal/see_all_patient.php
Chrome is being controlled by automated test software.

Hospital Management System

\ll Patient's List

First Name	Last Name	ID	Email	Address	Phone	Gender
raiyan	araf	1	raiyan@gmail.com	malibagh	123456789	male
Razi	Rahman	2	razi@gmail.com	Satkhira	1234253545	male
rashni	rs	1234	rashni@gmail.com	213malibagh	124000154	female
noah	williams	100	noah@gmail.com	california	123456789	female
noah	williams	100	noah@gmail.com	california	123456789	female
ssc	hsc	5	ssc@gmail.com	uttora	123456789	

This is the interface of See All Patient Module

4.4 Project Requirements

The central objective of our hospital management system project is to achieve the seamless implementation of a robust healthcare solution while adhering to the defined budget and quality benchmarks. This endeavor is accompanied by a multitude of intricacies encompassing time, financial considerations, scope definition, resource allocation, and the operational context that necessitate adept management. The project's success hinges on meticulous adherence to timelines, cost limitations, and functional requisites, ensuring the healthcare system meets the needs of medical professionals and patients alike. Precise resource utilization is of paramount importance, encompassing both human and technological assets. Skillfully navigating and resolving each of these challenges will culminate in a highly favorable and impactful outcome for the hospital and its stakeholders.

Time Management: The project's execution must adhere to the stipulated timeframe, with efforts concentrated on minimizing any potential delays. The creation of a software prototype demands 100 hours, while development necessitates 500 hours.

Subsequent software revisions will account for 80 hours while testing and debugging activities will encompass around 220 hours. This aggregates to a total workload of 900 hours. Considering a daily operational span of 12 hours, the project is poised for conclusion in roughly 45 days, translating to a span of 2.5 months or 10 weeks.

Budget Control: The project's execution must align with the assigned budget, and efforts must be made to curtail any supplementary expenditures. The designated budget for the project stands at 3,00,000 BDT.
Security: The project necessitates strict adherence to pertinent security protocols and regulations, safeguarding the system against unauthorized entry.

Customization: The system's adaptability is pivotal, catering to the distinct requisites of diverse tourism entities.

User-Friendliness: User experience is paramount, demanding an intuitive interface facilitating seamless task execution.

Maintenance and Support: The system's construct prioritizes straightforward upkeep and responsive support, promptly addressing any arising concerns.

5. FEATURES NOT TO BE TESTED

Thoroughly testing all integrated software features is essential.

6. TESTING APPROACH

6.1 Testing Levels

The system testing phase will be organized into several stages, beginning with Unit Testing, and progressing to Acceptance Testing. A dedicated independent testing professional will be engaged on a full-time basis for system and integration testing. The bulk of the testing responsibilities will be overseen by the test manager, in collaboration with the development teams.

Unit Testing: In software development, the preliminary and pivotal phase of unit testing involves scrutinizing each discrete and self-contained module of a program. The core aim of unit testing is to ascertain the accurate functioning of the software's tiniest components, preempting complications upon their assimilation into the broader framework. During unit testing, the code is isolated, and individual segments are methodically examined, usually by the coder during module creation. Oversight of unit tests may be assumed by the development team lead. Official repositories offer a suite of testing tools and packages that facilitate the unit testing process. Conducting unit tests at an early stage facilitates the timely resolution of glitches within project fragments before amalgamation, mitigating potential setbacks. This premature error identification streamlines subsequent debugging efforts and augments the ultimate product's global quality.

System/Integration Testing: Integration testing stands as a pivotal phase within the software testing continuum, succeeding unit testing, and orchestrated by a dedicated testing team. Its primary objective resides in ensuring the self-reliant efficacy of each software module while validating the operational prowess, functionality, and steadfastness of each assimilated component. The systematic validation of individual module performance curtails defects and streamlines debugging, irrespective of the top-down or bottom-up approach. Integration testing plays a pivotal role in validating developer-implemented functionalities against user requisites. A spectrum of integration testing methodologies exists, encompassing Big Bang, Incremental, Top-down, Bottom-up, and Sandwich approaches. The chosen method should harmonize with the developmental trajectory and practicality. Prior to commencement, a comprehensive test strategy must be devised, prioritizing testing for critical modules. Automated testing tools are vital for executing test cases and identifying defects, which then require reporting for retesting.

Following integration testing, system testing assumes significance to verify the unified functionality of all software modules when amalgamated into a cohesive system. Executed by a specialized testing team, this testing is categorized as black-box, devoid of internal module comprehension. System testing mandates a holistic evaluation of the software system, grounded in an understanding of real-world usage and stipulated requirements. Antecedent to system testing, a well-defined test strategy must be charted, and critical modules should be identified for priority testing. End-to-end testing is integral, ensuring comprehensive component interplay and interactions with external applications. Effective system testing vastly facilitates future software management and mitigation endeavors. Low-priority bugs can be reserved for acceptance testing, while system testing underscores meticulous assessment of software quality attributes, thereby ensuring the highest echelons of product quality.

Acceptance Testing: A pivotal juncture in the software development trajectory, Acceptance Testing ensues post-system testing and is orchestrated by the end-user. Herein, the software's readiness for market release is appraised by the customer. This phase necessitates the preparation of an alpha or beta product version, with the user well-versed in product specifics, domain intricacies, and feature sets. Any discerned issues during Acceptance Testing demand immediate prioritization and rectification. The test's outcomes substantiate both testing and development endeavors, encapsulating the software's overarching quality. Precise control of the testing audience is imperative, safeguarding ample data presence to forestall Acceptance Test outcome distortions.

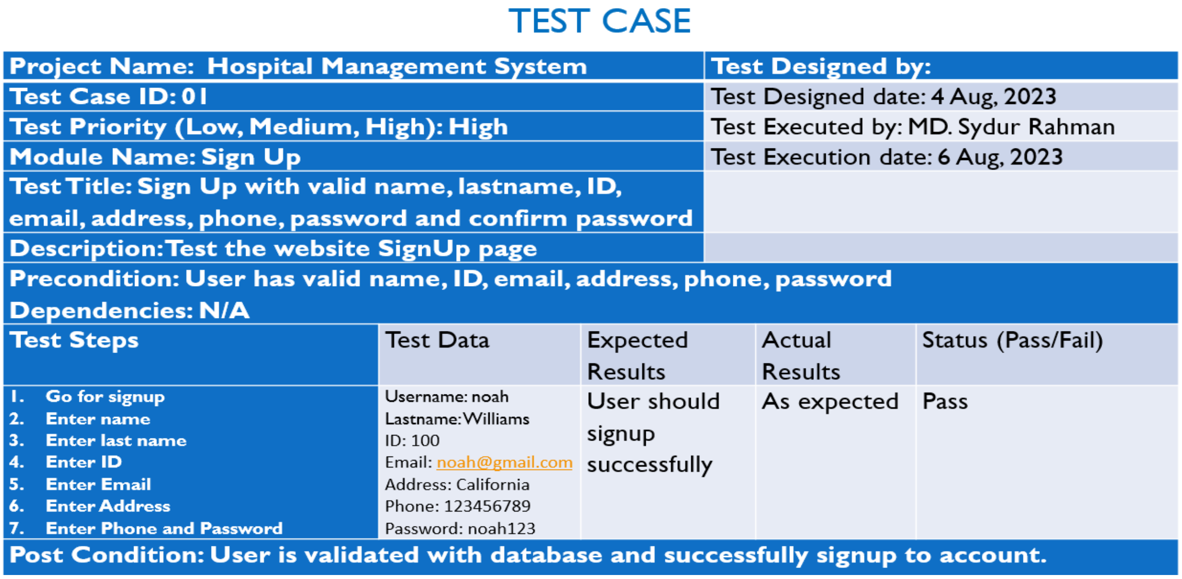
6.2 Test Tools

Selenium: A widely embraced open-source browser automation tool, Selenium proficiently automates web applications for exhaustive testing purposes, running scripts seamlessly across multiple browsers. With an extensive feature set, it offers cross-platform, cross-browser capabilities, seamlessly integrating with diverse tools. Language-neutral, Selenium accommodates popular languages, such as Java, C#, Python, Ruby, PHP, and JavaScript, among others. Complementary to several testing tools, it facilitates parallel testing and interfaces well with Agile, DevOps, and similar frameworks. Selenium is an assemblage of varied tools, including Selenium IDE, Selenium WebDriver, and Selenium Grid, enabling comprehensive report handling. Its versatility extends to mobile testing, covering hybrid, native, and mobile web apps. Noteworthy mobile testing tools like Appium, Splendored, Robotism, and IOS-driver, among others, support diverse operating systems. While codeless trends emerge, Selenium remains universally relevant, encouraging tester engagement. Third-party solutions like TestNG, JUnit, Extend Library, and Allure provide diverse reporting formats comprising timelines, graphs, pictorial charts, screenshots, error logs, and more.

6.3 Meetings

Our testing team assembles on a weekly basis to deliberate their advancements, obstacles, task statuses, and innovations aimed at refining the testing protocol. These gatherings stand as linchpins in elevating functionality, error pinpointing, and feasibility assessments. Biweekly, the testing team lead or supervisor undertakes a comprehensive review, guaranteeing alignment with user requisites and upholding lofty quality benchmarks. Continuous vigilance and oversight are imperative in realizing peak quality standards. In exigent circumstances, supplementary meetings may be convened as warranted. Testers must foster intercommunication, sharing challenges and strides amongst themselves. Virtual participation is enabled via home-based live chat sessions, facilitating discourse on enhancements and fresh concepts before scheduled assemblies. The team lead orchestrates synchronization with diverse units, tracking their progress and maintaining consistent contact through recurring meetings.

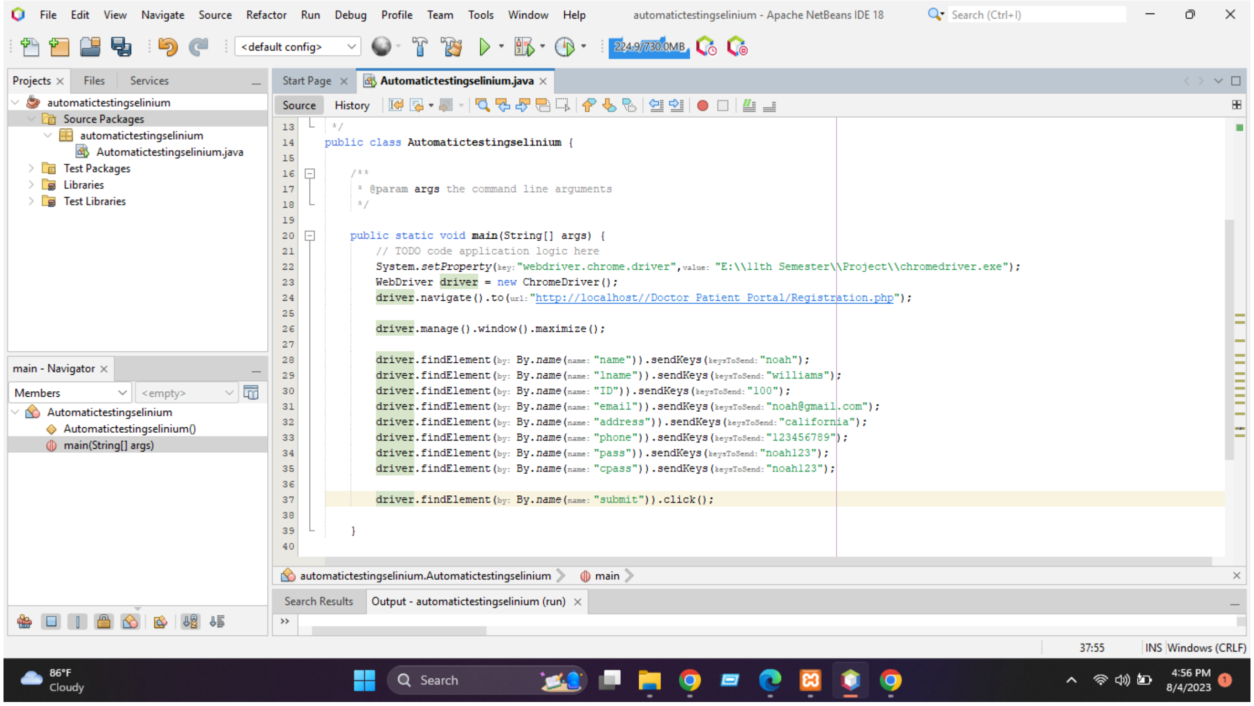
7. TEST CASES/TEST ITEMS



TEST CASE

Project Name: Hospital Management System			**Test Designed by:**	
Test Case ID: 01			Test Designed date: 4 Aug, 2023	
Test Priority (Low, Medium, High): High			Test Executed by: MD. Sydur Rahman	
Module Name: Sign Up			Test Execution date: 6 Aug, 2023	
Test Title: Sign Up with valid name, lastname, ID, email, address, phone, password and confirm password				
Description: Test the website SignUp page				
Precondition: User has valid name, ID, email, address, phone, password **Dependencies: N/A**				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go for signup **2. Enter name** **3. Enter last name** **4. Enter ID** **5. Enter Email** **6. Enter Address** **7. Enter Phone and Password**	Username: noah Lastname: Williams ID: 100 Email: noah@gmail.com Address: California Phone: 123456789 Password: noah123	User should signup successfully	As expected	Pass
Post Condition: User is validated with database and successfully signup to account.				

This is test case for testing Sign Up Module



```
 */
public class Automatictestingselinium {

    /**
     * @param args the command line arguments
     */

    public static void main(String[] args) {
        // TODO code application logic here
        System.setProperty(key:"webdriver.chrome.driver",value: "E:\\11th Semester\\Project\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to(url:"http://localhost//Doctor Patient Portal/Registration.php");

        driver.manage().window().maximize();

        driver.findElement(by: By.name(name: "name")).sendKeys(keysToSend:"noah");
        driver.findElement(by: By.name(name: "lname")).sendKeys(keysToSend:"williams");
        driver.findElement(by: By.name(name: "ID")).sendKeys(keysToSend:"100");
        driver.findElement(by: By.name(name: "email")).sendKeys(keysToSend:"noah@gmail.com");
        driver.findElement(by: By.name(name: "address")).sendKeys(keysToSend:"california");
        driver.findElement(by: By.name(name: "phone")).sendKeys(keysToSend:"123456789");
        driver.findElement(by: By.name(name: "pass")).sendKeys(keysToSend:"noah123");
        driver.findElement(by: By.name(name: "cpass")).sendKeys(keysToSend:"noah123");

        driver.findElement(by: By.name(name: "submit")).click();

    }

```

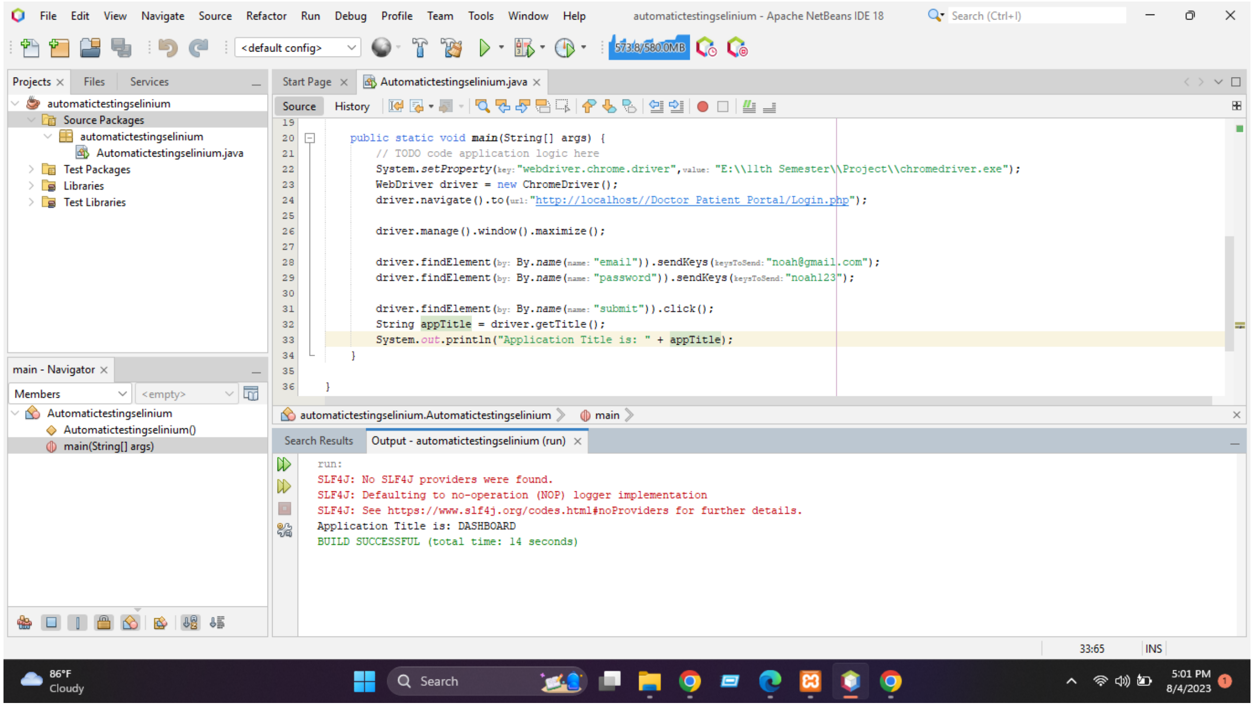
Here we can see successfully tested Sign Up Module

TEST CASE

Project Name: Hospital Management System	**Test Designed by:**
Test Case ID: 02	Test Designed date: 4 Aug, 2023
Test Priority (Low, Medium, High): High	Test Executed by: MD. Sydur Rahman
Module Name: Login	Test Execution date: 6 Aug, 2023
Test Title: Verify login with valid email and password	
Description: Test the website Login page	
Precondition: User has valid email and password **Dependencies: N/A**	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go for login **2. Enter email** **3. Enter password** **4. Click Login**	Email: noah@gmail.com Password: noah123	User should login successfully	As expected	Pass
Post Condition: User is validated with database and successfully login to account.				

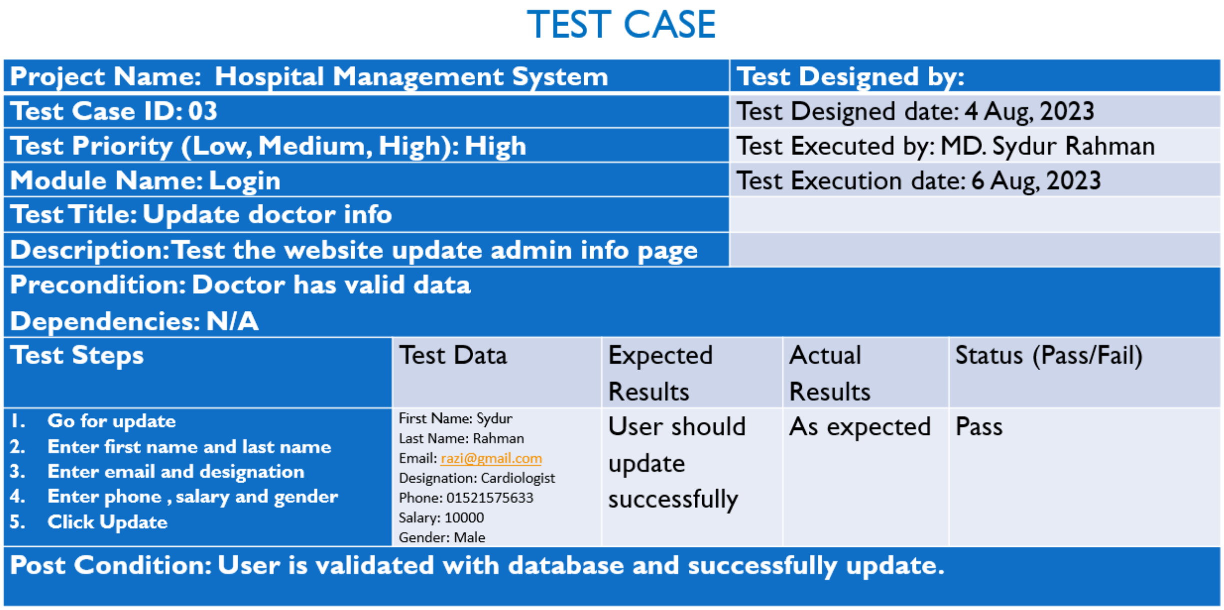
This is the test case for Login Module



```
    public static void main(String[] args) {
        // TODO code application logic here
        System.setProperty(key:"webdriver.chrome.driver",value: "E:\\11th Semester\\Project\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to(url:"http://localhost//Doctor Patient Portal/Login.php");

        driver.manage().window().maximize();

        driver.findElement(by: By.name(name: "email")).sendKeys(keysToSend:"noah@gmail.com");
        driver.findElement(by: By.name(name: "password")).sendKeys(keysToSend:"noah123");

        driver.findElement(by: By.name(name: "submit")).click();
        String appTitle = driver.getTitle();
        System.out.println("Application Title is: " + appTitle);
    }

}
```

```
run:
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Application Title is: DASHBOARD
BUILD SUCCESSFUL (total time: 14 seconds)
```

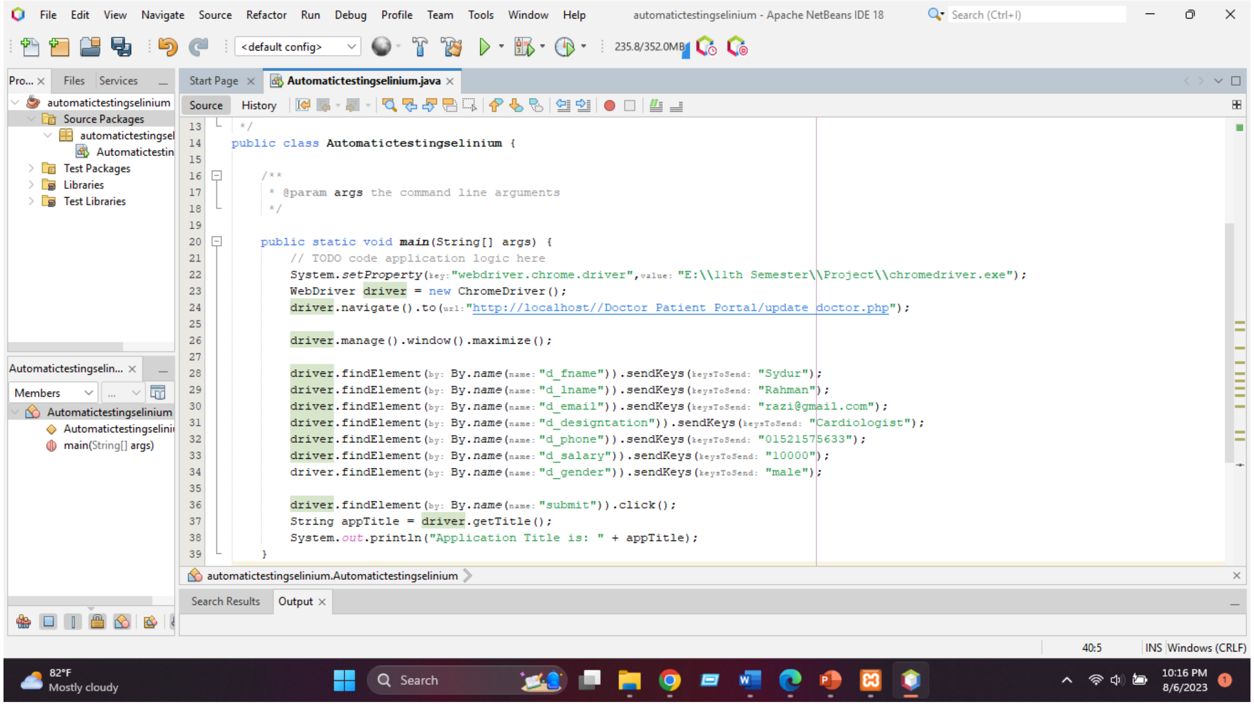
Here we can see successfully tested login module



TEST CASE

Project Name: Hospital Management System			**Test Designed by:**	
Test Case ID: 03			Test Designed date: 4 Aug, 2023	
Test Priority (Low, Medium, High): High			Test Executed by: MD. Sydur Rahman	
Module Name: Login			Test Execution date: 6 Aug, 2023	
Test Title: Update doctor info				
Description:Test the website update admin info page				
Precondition: Doctor has valid data				
Dependencies: N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go for update **2. Enter first name and last name** **3. Enter email and designation** **4. Enter phone , salary and gender** **5. Click Update**	First Name: Sydur Last Name: Rahman Email: razi@gmail.com Designation: Cardiologist Phone: 01521575633 Salary: 10000 Gender: Male	User should update successfully	As expected	Pass
Post Condition: User is validated with database and successfully update.				

This is the test case of Update Doctor Information



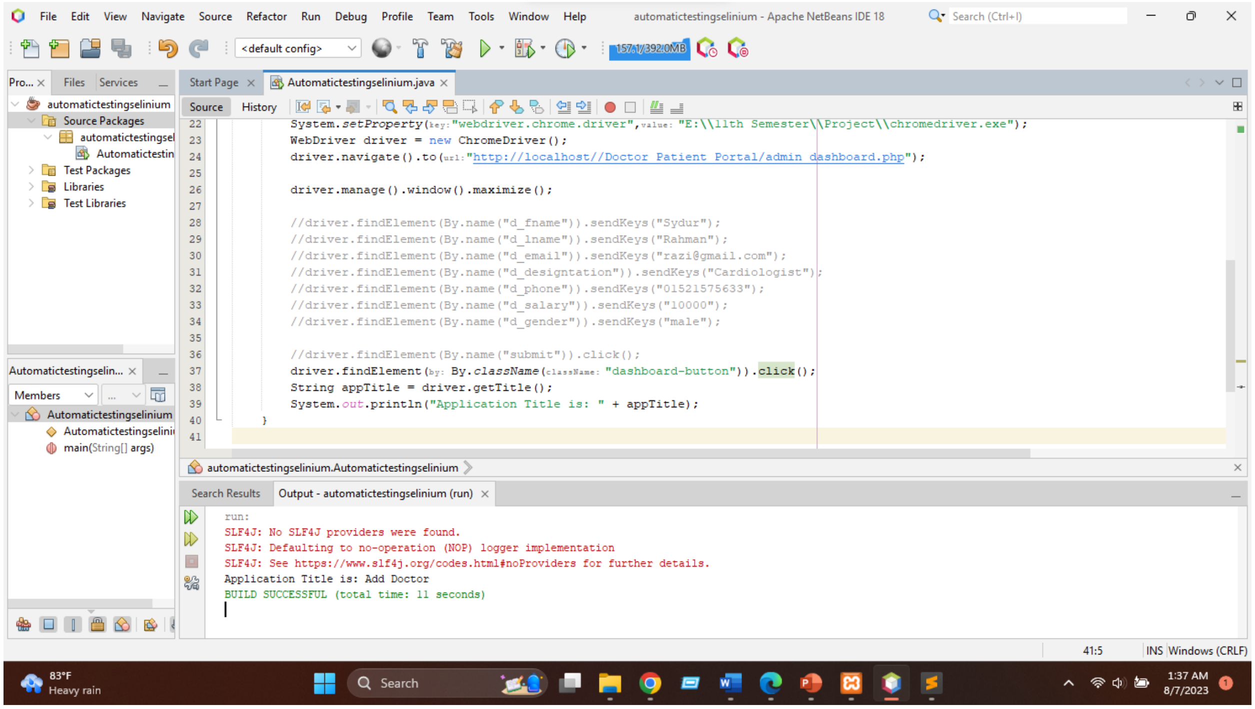
```java
 */
public class Automatictestingselinium {

    /**
     * @param args the command line arguments
     */

    public static void main(String[] args) {
        // TODO code application logic here
        System.setProperty(key:"webdriver.chrome.driver",value: "E:\\11th Semester\\Project\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to(url:"http://localhost//Doctor Patient Portal/update doctor.php");

        driver.manage().window().maximize();

        driver.findElement(by: By.name(name: "d_fname")).sendKeys(keysToSend: "Sydur");
        driver.findElement(by: By.name(name: "d_lname")).sendKeys(keysToSend: "Rahman");
        driver.findElement(by: By.name(name: "d_email")).sendKeys(keysToSend: "razi@gmail.com");
        driver.findElement(by: By.name(name: "d_designtation")).sendKeys(keysToSend: "Cardiologist");
        driver.findElement(by: By.name(name: "d_phone")).sendKeys(keysToSend: "01521575633");
        driver.findElement(by: By.name(name: "d_salary")).sendKeys(keysToSend: "10000");
        driver.findElement(by: By.name(name: "d_gender")).sendKeys(keysToSend: "male");

        driver.findElement(by: By.name(name: "submit")).click();
        String appTitle = driver.getTitle();
        System.out.println("Application Title is: " + appTitle);
    }
```

This is the code for testing update doctor module

TEST CASE

Project Name: Hospital Management System			**Test Designed by:**	
Test Case ID: 04			Test Designed date: 4 Aug, 2023	
Test Priority (Low, Medium, High): Medium			Test Executed by: MD. Sydur Rahman	
Module Name: Doctor Data			Test Execution date: 6 Aug, 2023	
Test Title: See doctor data				
Description: Test the existing of all doctor data				
Precondition: Database should consist all doctor info **Dependencies: N/A**				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go for check SHOW DOCTOR info **2. Click to check**	N/A	Fetch all data and make it visible	As expected	Pass
Post Condition: N/A				

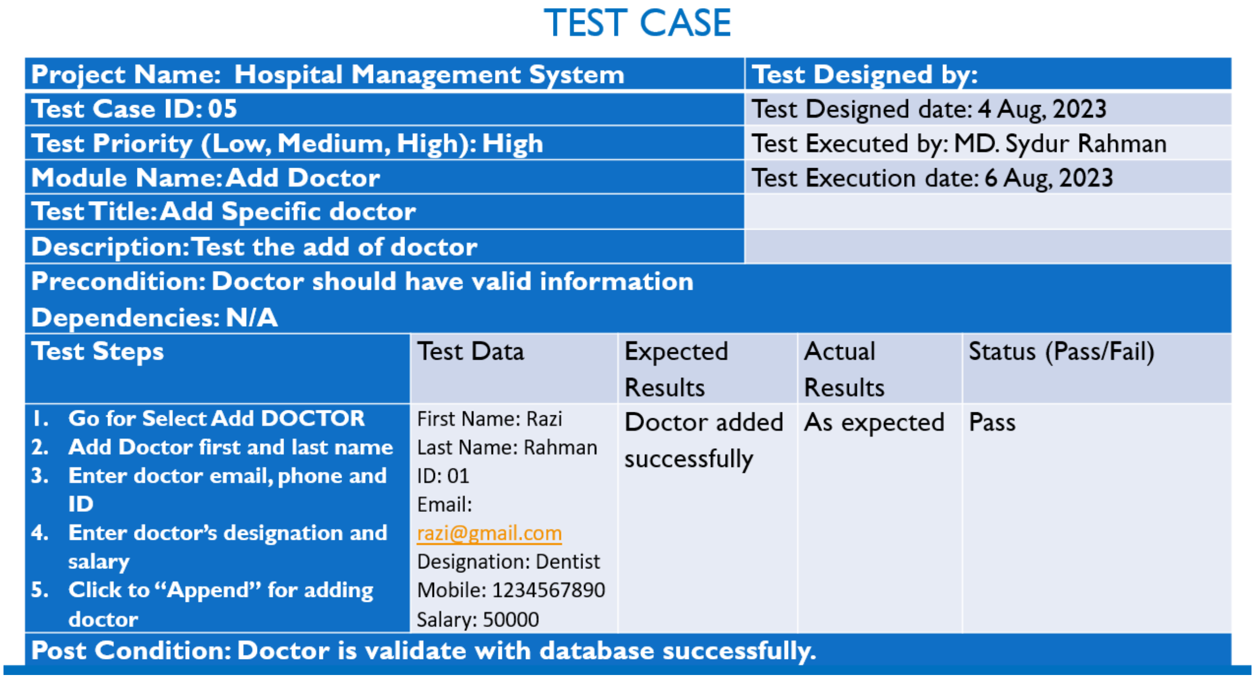
This is the test case of See Doctor Module



```java
        System.setProperty(key:"webdriver.chrome.driver",value: "E:\\11th Semester\\Project\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to(url:"http://localhost//Doctor Patient Portal/admin dashboard.php");

        driver.manage().window().maximize();

        //driver.findElement(By.name("d_fname")).sendKeys("Sydur");
        //driver.findElement(By.name("d_lname")).sendKeys("Rahman");
        //driver.findElement(By.name("d_email")).sendKeys("razi@gmail.com");
        //driver.findElement(By.name("d_designtation")).sendKeys("Cardiologist");
        //driver.findElement(By.name("d_phone")).sendKeys("01521575633");
        //driver.findElement(By.name("d_salary")).sendKeys("10000");
        //driver.findElement(By.name("d_gender")).sendKeys("male");

        //driver.findElement(By.name("submit")).click();
        driver.findElement(by: By.className(className: "dashboard-button")).click();
        String appTitle = driver.getTitle();
        System.out.println("Application Title is: " + appTitle);
    }

```

```
run:
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Application Title is: Add Doctor
BUILD SUCCESSFUL (total time: 11 seconds)
```

This is the code for testing see doctor module

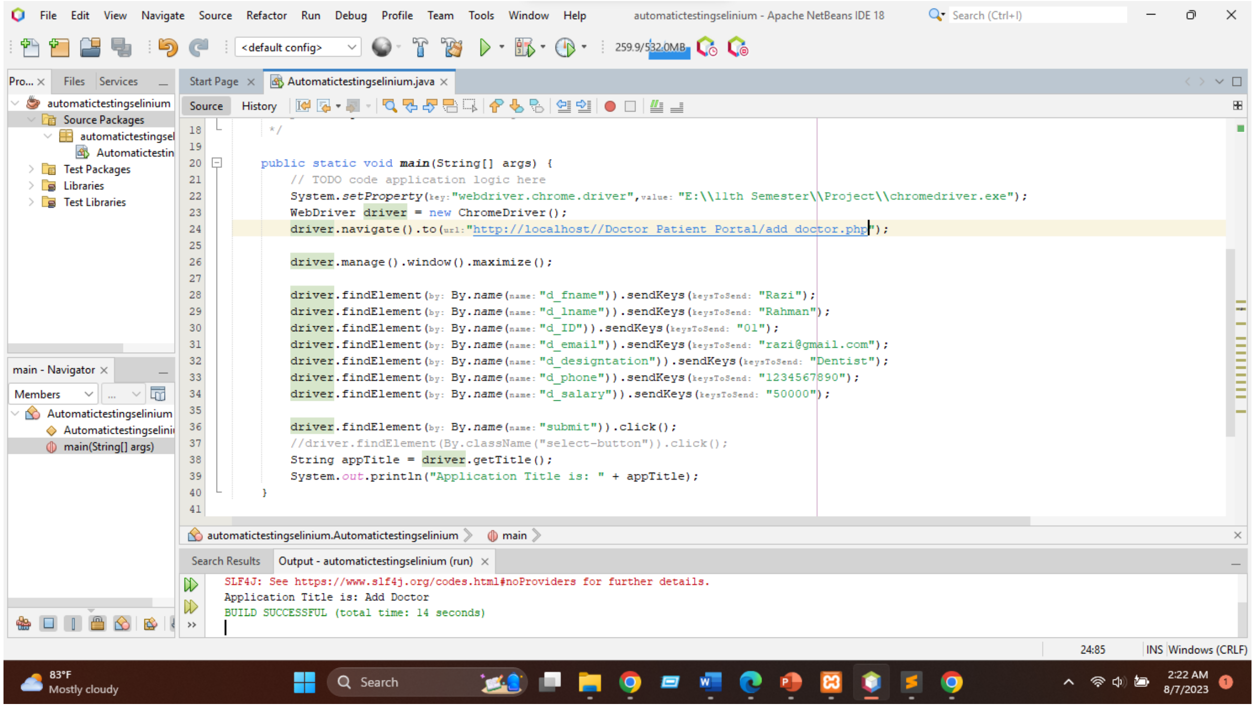


TEST CASE

Project Name: Hospital Management System	**Test Designed by:**
Test Case ID: 05	Test Designed date: 4 Aug, 2023
Test Priority (Low, Medium, High): High	Test Executed by: MD. Sydur Rahman
Module Name: Add Doctor	Test Execution date: 6 Aug, 2023
Test Title: Add Specific doctor	
Description: Test the add of doctor	
Precondition: Doctor should have valid information	
Dependencies: N/A	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go for Select Add DOCTOR **2. Add Doctor first and last name** **3. Enter doctor email, phone and ID** **4. Enter doctor's designation and salary** **5. Click to "Append" for adding doctor**	First Name: Razi Last Name: Rahman ID: 01 Email: razi@gmail.com Designation: Dentist Mobile: 1234567890 Salary: 50000	Doctor added successfully	As expected	Pass
Post Condition: Doctor is validate with database successfully.				

This is the test case of Add doctor module



```java
    */

    public static void main(String[] args) {
        // TODO code application logic here
        System.setProperty(key:"webdriver.chrome.driver",value: "E:\\11th Semester\\Project\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to(url:"http://localhost//Doctor Patient Portal/add doctor.php");

        driver.manage().window().maximize();

        driver.findElement(by: By.name(name: "d_fname")).sendKeys(keysToSend: "Razi");
        driver.findElement(by: By.name(name: "d_lname")).sendKeys(keysToSend: "Rahman");
        driver.findElement(by: By.name(name: "d_ID")).sendKeys(keysToSend: "01");
        driver.findElement(by: By.name(name: "d_email")).sendKeys(keysToSend: "razi@gmail.com");
        driver.findElement(by: By.name(name: "d_designtation")).sendKeys(keysToSend: "Dentist");
        driver.findElement(by: By.name(name: "d_phone")).sendKeys(keysToSend: "1234567890");
        driver.findElement(by: By.name(name: "d_salary")).sendKeys(keysToSend: "50000");

        driver.findElement(by: By.name(name: "submit")).click();
        //driver.findElement(By.className("select-button")).click();
        String appTitle = driver.getTitle();
        System.out.println("Application Title is: " + appTitle);
    }
```

```
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Application Title is: Add Doctor
BUILD SUCCESSFUL (total time: 14 seconds)
```

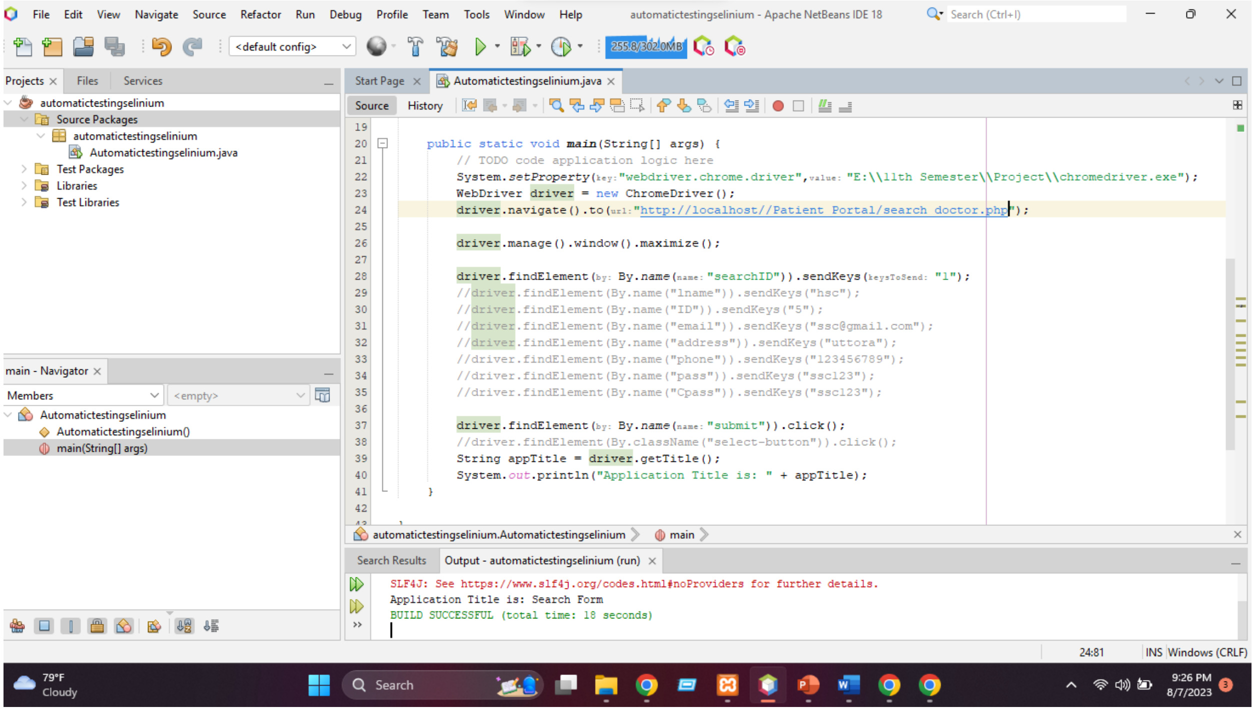
This is the code for testing add doctor module

TEST CASE

Project Name: Hospital Management System	**Test Designed by:**
Test Case ID: 06	Test Designed date: 4 Aug 2023
Test Priority (Low, Medium, High): Medium	Test Executed by: MD. Sydur Rahman
Module Name: Search Doctor	Test Execution date: 7 Aug 2023
Test Title: Search doctor with valid Doctor's ID	
Description:Test the doctor search page	
Precondition: Doctor has valid ID to search **Dependencies: N/A**	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the doctor search option **2. Enter doctor's ID** **3. Click submit**	Doctor's ID: 1	Doctor's Details will be show	As expected	Pass
Post Condition: Doctor is validated with database and successfully show the details.				

This is the test case of Search Doctor Module

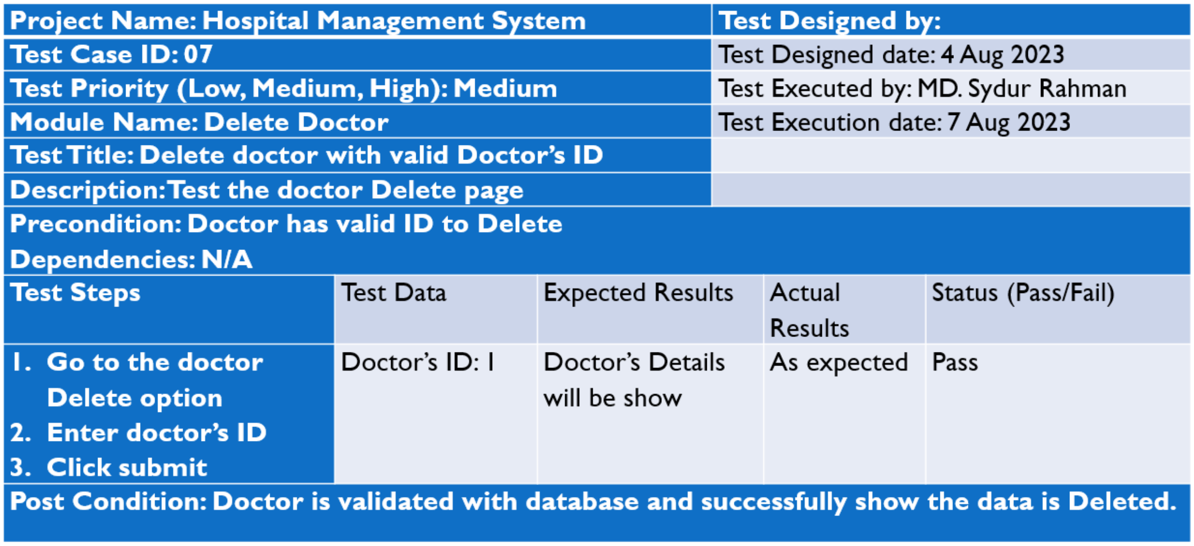


```java
    public static void main(String[] args) {
        // TODO code application logic here
        System.setProperty(key:"webdriver.chrome.driver",value: "E:\\11th Semester\\Project\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to(url:"http://localhost//Patient Portal/search doctor.php");

        driver.manage().window().maximize();

        driver.findElement(by: By.name(name: "searchID")).sendKeys(keysToSend: "1");
        //driver.findElement(By.name("lname")).sendKeys("hsc");
        //driver.findElement(By.name("ID")).sendKeys("5");
        //driver.findElement(By.name("email")).sendKeys("ssc@gmail.com");
        //driver.findElement(By.name("address")).sendKeys("uttora");
        //driver.findElement(By.name("phone")).sendKeys("123456789");
        //driver.findElement(By.name("pass")).sendKeys("ssc123");
        //driver.findElement(By.name("Cpass")).sendKeys("ssc123");

        driver.findElement(by: By.name(name: "submit")).click();
        //driver.findElement(By.className("select-button")).click();
        String appTitle = driver.getTitle();
        System.out.println("Application Title is: " + appTitle);
    }
```

```
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Application Title is: Search Form
BUILD SUCCESSFUL (total time: 18 seconds)
```

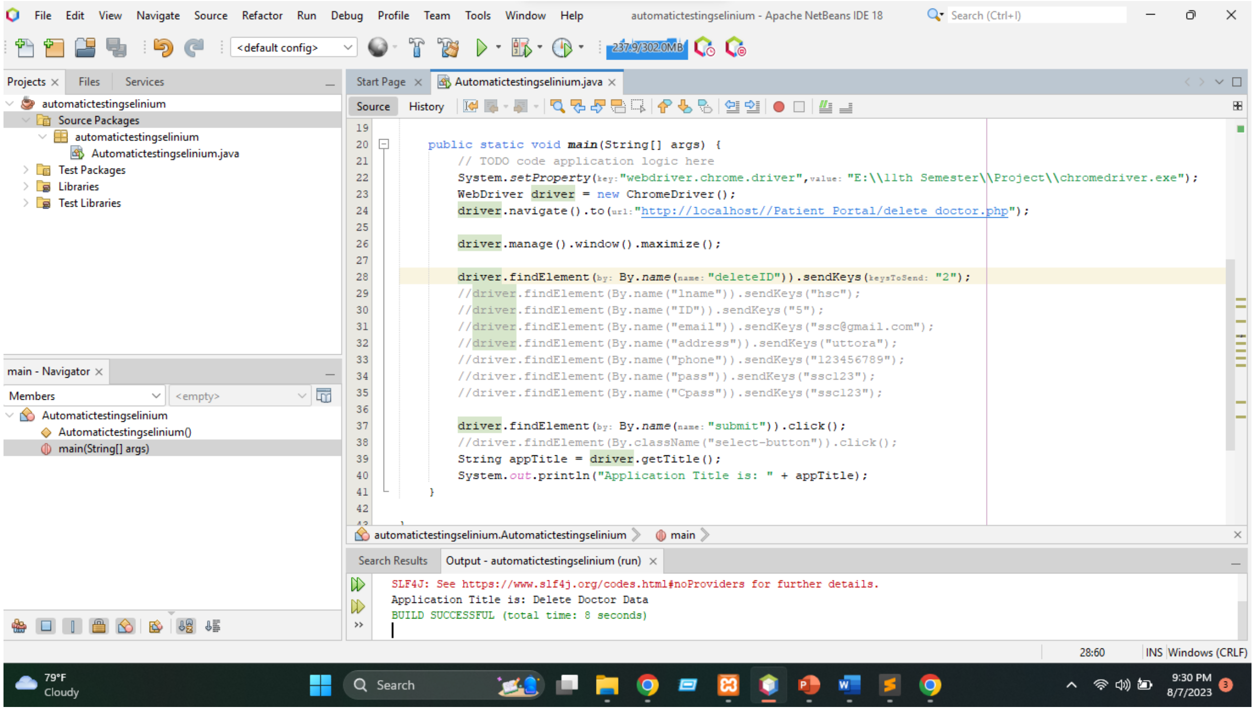
This is the code for testing Search Doctor module

TEST CASE



Project Name: Hospital Management System			**Test Designed by:**	
Test Case ID: 07			Test Designed date: 4 Aug 2023	
Test Priority (Low, Medium, High): Medium			Test Executed by: MD. Sydur Rahman	
Module Name: Delete Doctor			Test Execution date: 7 Aug 2023	
Test Title: Delete doctor with valid Doctor's ID				
Description: Test the doctor Delete page				
Precondition: Doctor has valid ID to Delete **Dependencies: N/A**				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the doctor Delete option **2. Enter doctor's ID** **3. Click submit**	Doctor's ID: 1	Doctor's Details will be show	As expected	Pass
Post Condition: Doctor is validated with database and successfully show the data is Deleted.				

This is the test case of Delete Doctor Module



```
    public static void main(String[] args) {
        // TODO code application logic here
        System.setProperty(key:"webdriver.chrome.driver",value: "E:\\11th Semester\\Project\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to(url:"http://localhost//Patient Portal/delete doctor.php");

        driver.manage().window().maximize();

        driver.findElement(by: By.name(name: "deleteID")).sendKeys(keysToSend: "2");
        //driver.findElement(By.name("lname")).sendKeys("hsc");
        //driver.findElement(By.name("ID")).sendKeys("5");
        //driver.findElement(By.name("email")).sendKeys("ssc@gmail.com");
        //driver.findElement(By.name("address")).sendKeys("uttora");
        //driver.findElement(By.name("phone")).sendKeys("123456789");
        //driver.findElement(By.name("pass")).sendKeys("ssc123");
        //driver.findElement(By.name("Cpass")).sendKeys("ssc123");

        driver.findElement(by: By.name(name: "submit")).click();
        //driver.findElement(By.className("select-button")).click();
        String appTitle = driver.getTitle();
        System.out.println("Application Title is: " + appTitle);
    }

```

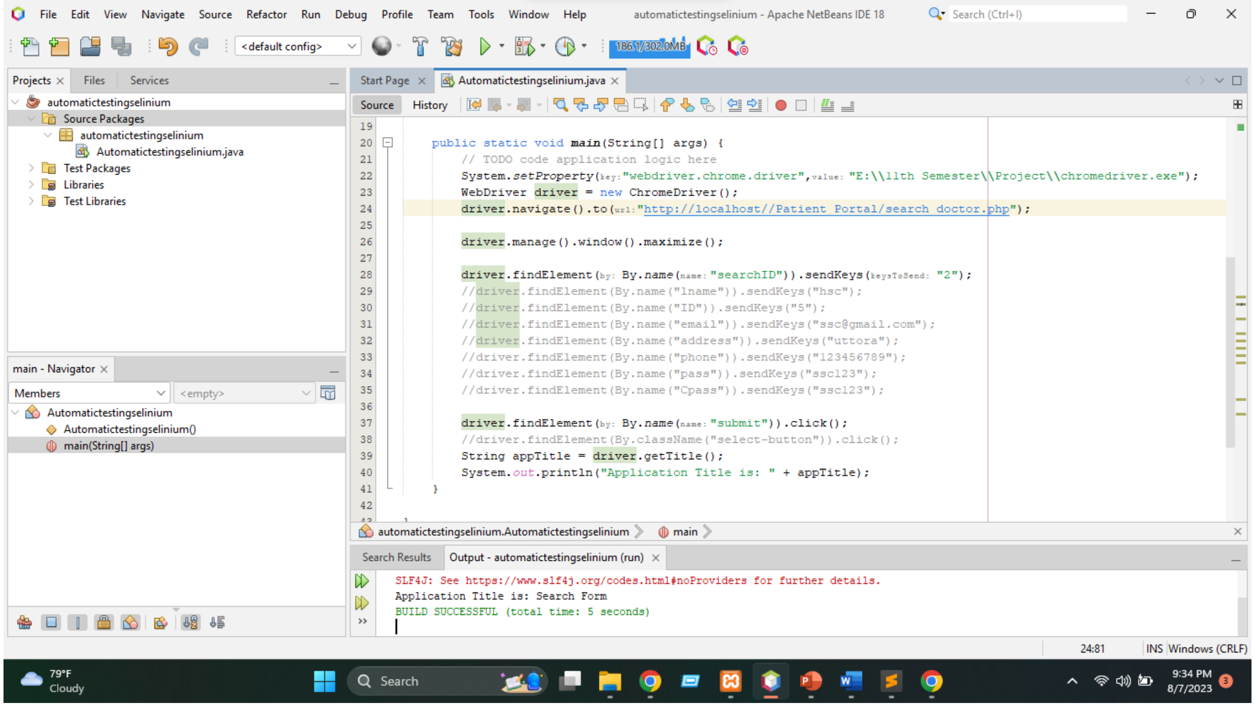
```
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Application Title is: Delete Doctor Data
BUILD SUCCESSFUL (total time: 8 seconds)
```

This is the code for testing Delete Doctor module

TEST CASE

<table>
<tr><td colspan="3"><b>Project Name: Hospital Management System</b></td><td colspan="2"><b>Test Designed by:</b></td></tr>
<tr><td colspan="3"><b>Test Case ID: 08</b></td><td colspan="2">Test Designed date: 4 Aug 2023</td></tr>
<tr><td colspan="3"><b>Test Priority (Low, Medium, High): Medium</b></td><td colspan="2">Test Executed by: MD. Sydur Rahman</td></tr>
<tr><td colspan="3"><b>Module Name: Search Patient</b></td><td colspan="2">Test Execution date: 7 Aug 2023</td></tr>
<tr><td colspan="3"><b>Test Title: Search Patient with valid Patient’s ID</b></td><td colspan="2"></td></tr>
<tr><td colspan="3"><b>Description: Test the Search Patient page</b></td><td colspan="2"></td></tr>
<tr><td colspan="5"><b>Precondition: Patient has valid ID to Search</b>
<b>Dependencies: N/A</b></td></tr>
<tr><td><b>Test Steps</b></td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td><b>1. Go to the Search Patient option</b>
<b>2. Enter patient’s ID</b>
<b>3. Click search</b></td><td>Doctor’s ID: 2</td><td>Patient’s Details will be show</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5"><b>Post Condition: Patient is validated with database and successfully show the data.</b></td></tr>
</table>

This is the test case of Search Patient Module



```java
    public static void main(String[] args) {
        // TODO code application logic here
        System.setProperty(key:"webdriver.chrome.driver",value: "E:\\11th Semester\\Project\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to(url:"http://localhost//Patient Portal/search doctor.php");

        driver.manage().window().maximize();

        driver.findElement(by: By.name(name: "searchID")).sendKeys(keysToSend: "2");
        //driver.findElement(By.name("lname")).sendKeys("hsc");
        //driver.findElement(By.name("ID")).sendKeys("5");
        //driver.findElement(By.name("email")).sendKeys("ssc@gmail.com");
        //driver.findElement(By.name("address")).sendKeys("uttora");
        //driver.findElement(By.name("phone")).sendKeys("123456789");
        //driver.findElement(By.name("pass")).sendKeys("ssc123");
        //driver.findElement(By.name("Cpass")).sendKeys("ssc123");

        driver.findElement(by: By.name(name: "submit")).click();
        //driver.findElement(By.className("select-button")).click();
        String appTitle = driver.getTitle();
        System.out.println("Application Title is: " + appTitle);
    }
```

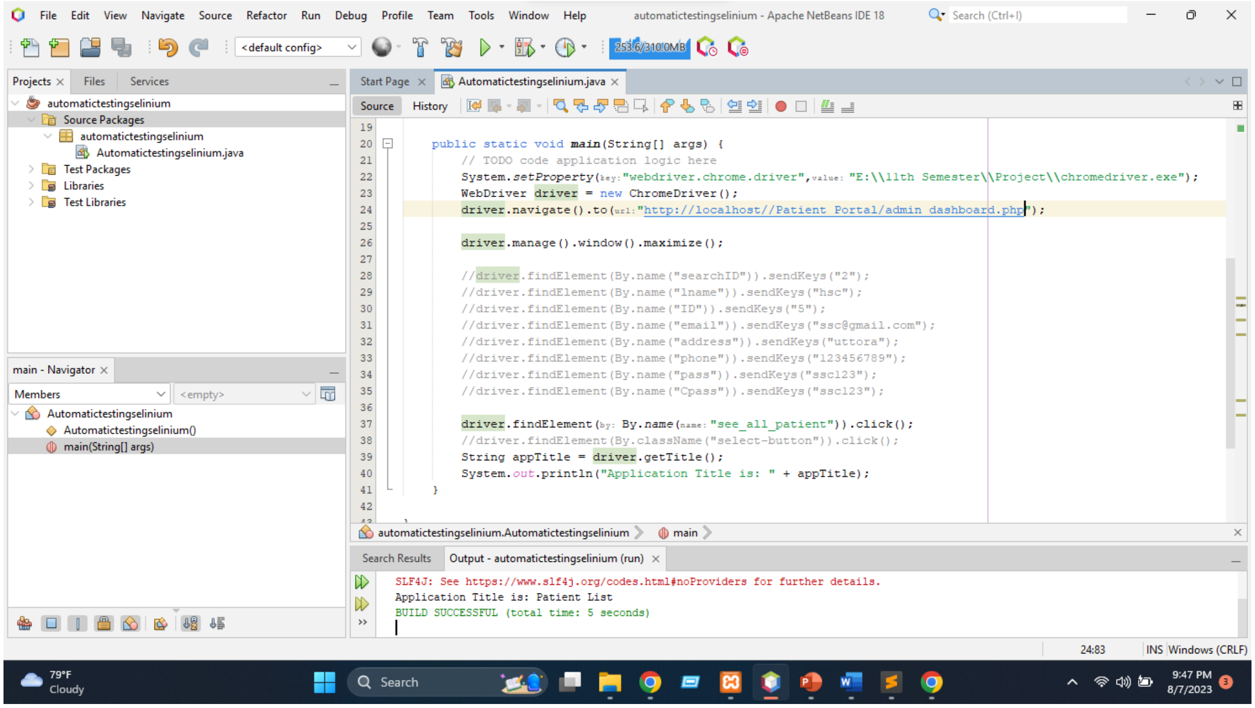
```
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Application Title is: Search Form
BUILD SUCCESSFUL (total time: 5 seconds)
```

This is the code for testing Search Patient module

TEST CASE

<table>
<tr><td colspan="3"><b>Project Name: Hospital Management System</b></td><td colspan="2"><b>Test Designed by:</b></td></tr>
<tr><td colspan="3"><b>Test Case ID: 09</b></td><td colspan="2">Test Designed date: 4 Aug 2023</td></tr>
<tr><td colspan="3"><b>Test Priority (Low, Medium, High): Medium</b></td><td colspan="2">Test Executed by: MD. Sydur Rahman</td></tr>
<tr><td colspan="3"><b>Module Name: See All Patient</b></td><td colspan="2">Test Execution date: 7 Aug 2023</td></tr>
<tr><td colspan="3"><b>Test Title: See All Patient</b></td><td colspan="2"></td></tr>
<tr><td colspan="3"><b>Description:Test the See All Patient module</b></td><td colspan="2"></td></tr>
<tr><td colspan="5"><b>Precondition: Patient has valid to database</b>
<b>Dependencies: N/A</b></td></tr>
<tr><td><b>Test Steps</b></td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td><b>1. Go to the Admin Dashboard</b>
<b>2. Click See All Patient Button</b></td><td>All Patient list will be showed</td><td>All Patient's Details will be showed</td><td>As expected</td><td>Pass</td></tr>
<tr><td colspan="5"><b>Post Condition: Patient is validated with database and successfully show the data.</b></td></tr>
</table>

This is the test case of See All Patient Module



```java
    public static void main(String[] args) {
        // TODO code application logic here
        System.setProperty(key:"webdriver.chrome.driver",value: "E:\\11th Semester\\Project\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to(url:"http://localhost//Patient Portal/admin dashboard.php");

        driver.manage().window().maximize();

        //driver.findElement(By.name("searchID")).sendKeys("2");
        //driver.findElement(By.name("lname")).sendKeys("hsc");
        //driver.findElement(By.name("ID")).sendKeys("5");
        //driver.findElement(By.name("email")).sendKeys("ssc@gmail.com");
        //driver.findElement(By.name("address")).sendKeys("uttora");
        //driver.findElement(By.name("phone")).sendKeys("123456789");
        //driver.findElement(By.name("pass")).sendKeys("ssc123");
        //driver.findElement(By.name("Cpass")).sendKeys("ssc123");

        driver.findElement(by: By.name(name: "see_all_patient")).click();
        //driver.findElement(By.className("select-button")).click();
        String appTitle = driver.getTitle();
        System.out.println("Application Title is: " + appTitle);
    }
```

```
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Application Title is: Patient List
BUILD SUCCESSFUL (total time: 5 seconds)
```

This is the code for testing See All Patient module.

8. ITEM PASS/FAIL CRITERIA

The primary aim of this section is to establish clear PASS/FAIL benchmarks for the tests integrated into this project. A system or unit achieving a score ranging from 92% to 96% will qualify under the pass criterion, while a score below 80% will signify failure. We have adopted this metric to evaluate the reliability and user satisfaction quotient of our project.

9. TEST DELIVERABLES

- Careful selection of participants for an acceptance test is vital as unreliable testers can generate erroneous results and feedback. It operates as a contractual agreement between the development team's release and software delivery.
- Thorough documentation of system integration modules, test strategies, and outcomes is imperative. Special emphasis on Commercial Off-The-Shelf (COTS) and third-party integrations is essential. Clear articulation of COTS tolerance levels and component functionality is warranted. Calculations involving DRE, Phage, Spoilage, and other system testing metrics should be computed and documented before software handover, facilitating comprehensible QA enhancement and effort.

- Proper documentation of unit test findings and results is crucial. Rigorous product analysis post-testing is vital. A prepared turnover document is a prerequisite.
- Incorporating screen-based prototyping concepts and software system toolkits elucidates software operations and potential challenges. It serves to communicate requirement fulfillment to end-users.
- Completion of mockup reports pre-project delivery provides a visual blueprint, offering a tangible representation of the envisaged design. A variety of chart, graph, and illustration formats allow seamless comprehension without unnecessary scrolling.
- Incident reports are generated to enhance employee safety and foster workplace best practices, elevating industry/organizational standards and ensuring project success. Our project boasts a comprehensive summary and report.
- The test log chronicles events transpiring throughout a test regimen and execution, alongside phase statuses. Rigorous revision and data recording elucidate actions and distinct approaches adopted within the project.

10. STAFFING AND TRAINING NEEDS

Due to the intricate project distribution and stages, it is advisable to appoint at least one dedicated inspector. For evaluation purposes, the designated individual will require an initial allocation of time at the project's commencement, followed by full-time engagement approximately six months later. In the absence of an alternative tester, the project/test manager will assume the role. To ensure a comprehensive and relevant analysis, the ensuing topics related to preparation warrant consideration. Personnel allocation for this project has been strategically outlined. A significant portion of the team will partake in specific research tasks, expounded upon extensively in the responsibilities section.

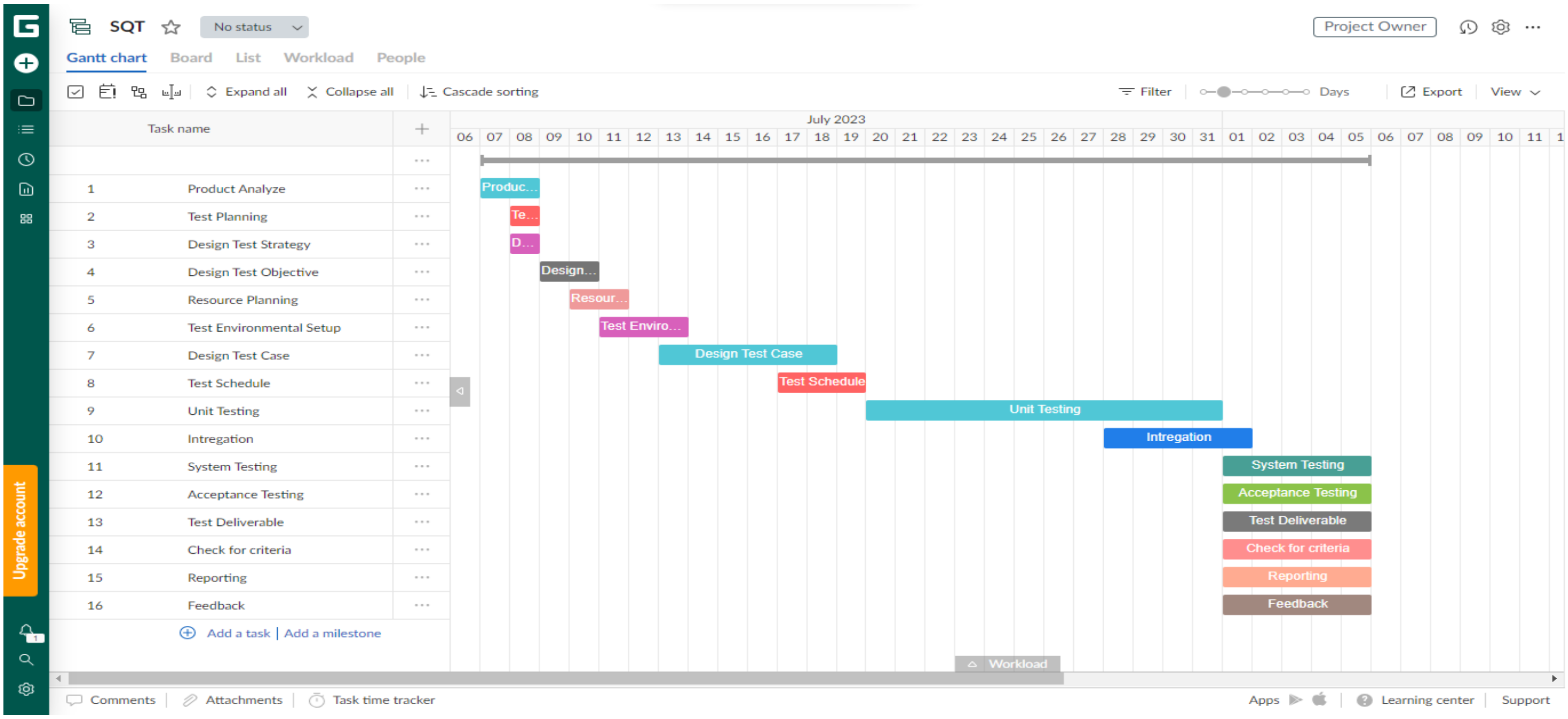
- Proficiency in Java, C++, Dart, Flutter, and MySQL is essential for developers and testers.
- Automation testers must possess apt knowledge and hands-on expertise in utilizing the tools.
- Adequate training in the fundamental operations of the EDI interface is imperative for developers and tester(s). Comprehensive training on the EDI communications process is essential for operations staff prior to project finalization.
- The sales administration team should undergo training for the new screens and reports.

11. RESPONSIBILITIES

	TM	PM	Dev Team	Test Team	Client
Acceptance test Documentation & Execution	X	X		X	X
System/Integration test Documentation & Exec.	X		X	X	
Unit test documentation & execution	X		X	X	
System Design Reviews	X	X	X	X	X
Detail Design Reviews	X	X	X	X	
Test procedures and rules	X	X	X	X	
Screen & Report prototype reviews			X	X	X
Change Control and regression testing	X	X	X	X	X

12. TESTING SCHEDULE

The project plan has earmarked dedicated time slots for the subsequent testing undertakings. Precise dates and timings for each activity are meticulously outlined within the project plan's timeline. Correspondingly, the individuals necessary for executing each task are distinctly elucidated in the project timeline and plan. Streamlining the orchestration of personnel encompassing the test team, development team, management, and the customer shall be effectively managed by the project manager, collaboratively with the development and test team leaders.



	Task name
1	Product Analyze
2	Test Planning
3	Design Test Strategy
4	Design Test Objective
5	Resource Planning
6	Test Environmental Setup
7	Design Test Case
8	Test Schedule
9	Unit Testing
10	Intregation
11	System Testing
12	Acceptance Testing
13	Test Deliverable
14	Check for criteria
15	Reporting
16	Feedback

13. PLANNING RISKS AND CONTINGENCIES

S/N	Risk Description	Probability	Impact	Mitigation Plan
1	Inadequate Data Backup	15%	Moderate	Implement an automated backup system to regularly safeguard critical data.
2	Network Connectivity Issues	25%	Significant	Establish redundant network connections to ensure seamless operations during disruptions.
3	Integration Challenges	40%	High	Conduct thorough compatibility tests and collaborates closely with third-party providers.
4	Incomplete User Training	60%	Moderate	Offer comprehensive user training sessions with readily accessible documentation.

13. APROVALS

Project Sponsor	Approved
Development Management	Approved
EDI Project Manager	Approved
RS Test Manager	Approved
RS Development Team Manager	Approved
Reassigned Sales	Approved
Order Entry EDI Team Manager	Approved