

WholeException



Final Presentation

Team Members:

Ashutosh Mishra(45393819)

Harsh Raj(45393544)

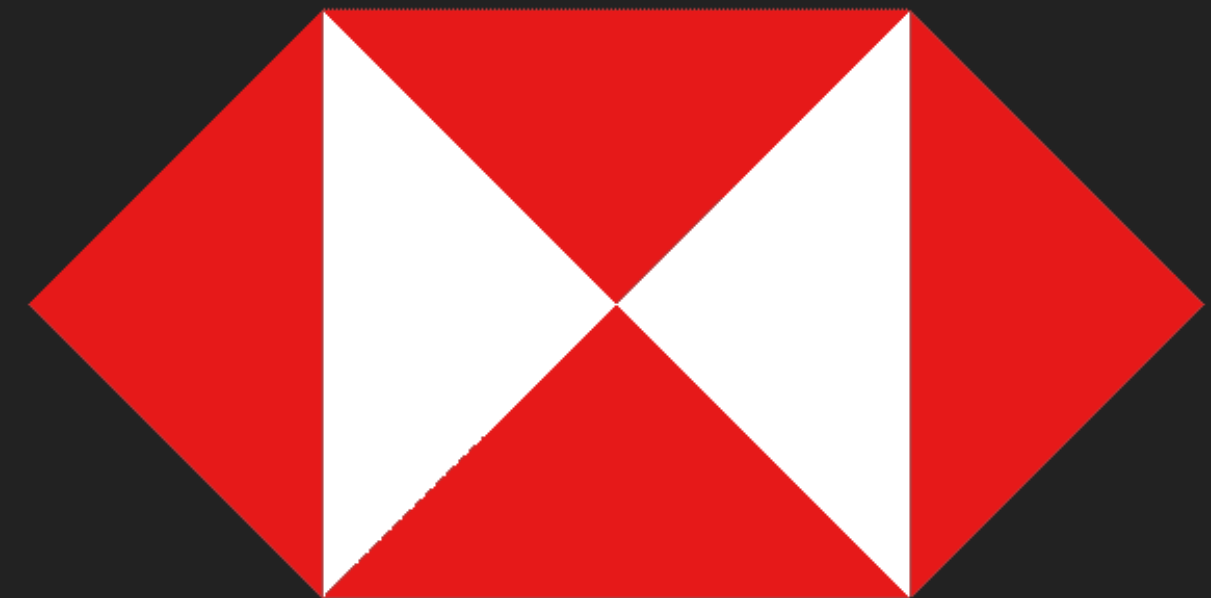
Nishtha Jain(45393708)

Pallav Aggarwal(45393887)

Rishabh Kumar(45393911)

Shaily Sinha(45393544)

Swathi Vailappilly(45393939)



Project Topic:

Hospital Managment System





Team Formation

1

Frontend

- Ashutosh
- Pallav
- Rishabh
- Swati

2

Backend

- Harsh
- Nishtha
- Shaili





Topic Selection

Selected Topic: Hospital Management System

Reason for Selection:

- Boosts efficiency by automating data, appointments, and prescriptions.
- Enhances care with faster processing and improved satisfaction.
- Reduces errors through automated data management.
- Integrates functions for better staff coordination.
- Scales and adapts for future growth and relevance.





Topics to **be Covered**

1

Project Synopsis

2

Implementation

3

Conclusion





Project Synopsis

Objective: Develop a comprehensive hospital management system to streamline administrative and medical processes.

Scope: Patient management, appointment scheduling, doctor records, and reporting.

Expected Outcomes: Improved operational efficiency, reduced manual errors, better patient and doctor experience.





Implementation

Features Developed:

- **Patient Management:** Registration, record updating, book and reschedule appointments
- **Admin Features:** Add/manage doctors, view all doctors, cancel appointments, show patients.
- **Doctor's Dashboard:** Add schedule, view appointments and suggest medicine and tests.





Implementation

Frontend: Admin home page

HSBC Hospital

- Dashboard
- Doctors
- Users
- Appointments
- Patients

Admin Dashboard

6 Doctors

7 Users

12 Patients

22 Appointments

Stats

No. of appointments by each doctor

doc01 (Example Doctor01)	--> 4
doc02 (Example Doctor02)	--> 4
doc03 (Example Doctor03)	--> 2
doc04 (Example Doctor04)	--> 5
doc03 (Example Doctor03)	--> 2
doc04 (Example Doctor04)	--> 5

No. of visits done by each patient

pat01 --> 1	pat02 --> 1	pat03 --> 1	pat04 --> 2
pat06 --> 3	pat07 --> 3	pat05 --> 1	pat08 --> 2
pat09 --> 2	pat10 --> 2	pat11 --> 2	pat12 --> 2

No. of appointments per day

No. of appointments on 2024-08-17	--> 11
No. of appointments on 2024-08-18	--> 8
No. of appointments on 2024-08-19	--> 3

No. of patients by each gender

No. of Male patients	--> 6
No. of Female patients	--> 6



Implementation

Frontend: View/Edit/Remove Doctors page

Admin Page

127.0.0.1:5500/application/frontend/Admin/AdminDoctorPage.html

HSBC Hospital

Dashboard

Doctors

Users

Appointments

Patients

Add New Doctor

+ Add New

All Doctors(6)

Doctor ID	Doctor Name	Actions
doc01	Example Doctor01	<a>Edit <a>View <a>Remove
doc02	Example Doctor02	<a>Edit <a>View <a>Remove
doc03	Example Doctor03	<a>Edit <a>View <a>Remove
doc04	Example Doctor04	<a>Edit <a>View <a>Remove
doc03	Example Doctor03	<a>Edit <a>View <a>Remove
doc04	Example Doctor04	<a>Edit <a>View <a>Remove

127.0.0.1:5500/application/frontend/Admin/AdminDoctorPage.html#





Implementation

Frontend: View/Edit/Remove Users page

Admin Page

127.0.0.1:5500/application/frontend/Admin/AdminUserPage.html

HSBC Hospital

Dashboard

Doctors

Users

Appointments

Patients

Add New User

+ Add New

All Users(7)

User ID	User Name	Actions
user01	Example User01	<div>EditViewRemove</div>
user02	Example User02	<div>EditViewRemove</div>
user03	Example User03	<div>EditViewRemove</div>
user04	Example User04	<div>EditViewRemove</div>
user05	Example User05	<div>EditViewRemove</div>
user06	Example User06	<div>EditViewRemove</div>
user07	Example User07	<div>EditViewRemove</div>

127.0.0.1:5500/application/frontend/Admin/AdminUserPage.html#





Implementation

Frontend: View/Edit/Remove Appointments page

Admin Page

127.0.0.1:5500/application/frontend/Admin/AdminAppointmentPage.html

HSBC Hospital

Dashboard

Doctors

Users

Appointments

Patients

Schedule an Appointment

Date: dd-mm-yyyy

Doctor: Choose Doctor from the list

Filter

Clear Filter

+ Schedule New

Upcoming Appointments(22)

Date	Time Slot	Doctor ID	Doctor Name	Patient ID	Patient Name	Actions
2024-08-17	11:00-13:00	doc01	Example Doctor01	pat01	Example Patient01	<div>Edit Remove</div>
2024-08-17	14:00-16:00	doc01	Example Doctor01	pat02	Example Patient02	<div>Edit Remove</div>
2024-08-18	09:00-11:00	doc01	Example Doctor01	pat03	Example Patient03	<div>Edit Remove</div>
2024-08-18	14:00-16:00	doc01	Example Doctor01	pat04	Example Patient04	<div>Edit Remove</div>
2024-08-17	11:00-13:00	doc02	Example Doctor02	pat04	Example Patient04	<div>Edit Remove</div>
2024-08-18	11:00-13:00	doc02	Example Doctor02	pat06	Example Patient06	<div>Edit Remove</div>
2024-08-18	14:00-16:00	doc02	Example Doctor02	pat07	Example Patient07	<div>Edit Remove</div>
2024-08-19	14:00-16:00	doc02	Example Doctor02	pat05	Example Patient05	<div>Edit Remove</div>
2024-08-17	09:00-11:00	doc03	Example Doctor03	pat08	Example Patient08	<div>Edit Remove</div>
2024-08-19	14:00-16:00	doc03	Example Doctor03	pat09	Example Patient09	<div>Edit Remove</div>
2024-08-17	09:00-11:00	doc04	Example Doctor04	pat10	Example Patient10	<div>Edit Remove</div>
2024-08-17	11:00-13:00	doc04	Example Doctor04	pat11	Example Patient11	<div>Edit Remove</div>



Implementation

Frontend: View/Edit/Remove Patients page

Admin Page

127.0.0.1:5500/application/frontend/Admin/AdminPatientPage.html

HSBC Hospital

Dashboard

Doctors

Users

Appointments

Patients

Add New Patient

Age(in years):

Gender:

Choose Gender from the list

Filter

Clear Filter

+ Add New

All Patients(12)

Patient ID	Patient Name	Age	Gender	Actions
pat01	Example Patient01	23	male	<div>Edit View Remove</div>
pat02	Example Patient02	35	female	<div>Edit View Remove</div>
pat03	Example Patient03	60	male	<div>Edit View Remove</div>
pat04	Example Patient04	45	female	<div>Edit View Remove</div>
pat05	Example Patient05	18	male	<div>Edit View Remove</div>
pat06	Example Patient06	30	female	<div>Edit View Remove</div>
pat07	Example Patient07	52	male	<div>Edit View Remove</div>
pat08	Example Patient08	40	female	<div>Edit View Remove</div>
pat09	Example Patient09	67	male	<div>Edit View Remove</div>
pat10	Example Patient10	55	female	<div>Edit View Remove</div>
pat11	Example Patient11	29	male	<div>Edit View Remove</div>
pat12	Example Patient12	75	female	<div>Edit View Remove</div>





Implementation

Frontend: Doctor Appointments page

HSBC Hospital

View Doctor Appointments

Add 3 days Schedule

Suggest Test

Suggest Medicine

Add 3 day Schedule Appointment

+ Schedule New

Upcoming Appointments(15)

Date	Time Slot	Doctor ID	Doctor Name	Patient ID	Patient Name
2024-08-17	11:00-13:00			pat01	Example Patient01
2024-08-17	14:00-16:00			pat02	Example Patient02
2024-08-18	09:00-11:00			pat03	Example Patient03
2024-08-18	14:00-16:00			pat04	Example Patient04
2024-08-17	11:00-13:00			pat04	Example Patient04
2024-08-18	11:00-13:00			pat06	Example Patient06
2024-08-18	14:00-16:00			pat07	Example Patient07
2024-08-17	09:00-11:00			pat08	Example Patient08
2024-08-18	14:00-16:00			pat08	Example Patient08
2024-08-19	14:00-16:00			pat09	Example Patient09
2024-08-17	09:00-11:00	doc04	Example Doctor04	pat10	Example Patient10
2024-08-17	11:00-13:00	doc04	Example Doctor04	pat11	Example Patient11
2024-08-17	14:00-16:00	doc04	Example Doctor04	pat12	Example Patient12
2024-08-18	11:00-13:00	doc04	Example Doctor04	pat06	Example Patient06
2024-08-18	14:00-16:00	doc04	Example Doctor04	pat07	Example Patient07

Schedule an Appointment

Select Doctor's ID

Select Patient's ID

Select Date

dd-mm-yyyy

Select Time Slot

Save

Close



Implementation

Frontend: Suggested test according to disease

All Patients(13)

Patient ID	Patient Name	Actions
pat01	Example Patient01	View
pat02		View
pat03		View
pat04		View
pat05		View
pat06		View
pat07		View
pat08		View
pat09		View
pat10	Example Patient10	View
pat11	Example Patient11	View
pat12	Example Patient12	View

Details of Patient: pat01

Patient ID
pat01

Name
Example Patient01

Disease Name
FLU

Suggested Tests
Blood Test,Influenza Test





Implementation

Frontend: Suggested medicine according to disease

Patient ID	Patient Name	Actions
pat01	Example Patient01	View
pat02		View
pat03		View
pat04		View
pat05		View
pat06		View
pat07		View
pat08		View
pat09		View
pat10	Example Patient10	View
pat11	Example Patient11	View

Details of Patient: pat09

Patient ID

pat09

Name

Example Patient09

Disease Name

ARTHRITIS

Suggested Medicine

Ibuprofen,Methotrexate,Hydroxychloroquine



Implementation

Frontend: User: all patient appointments

Book Patient

127.0.0.1:5500/application/frontend/User/BookPatient.html

HSBC Hospital

Book Patient

View Patient Appointment

Add Patient detail

All Patient Appointment

+ Schedule New

Date	Time Slot	Doctor ID	Doctor Name	Patient ID	Patient Name
2024-08-17	11:00-13:00	doc01	Example Doctor01	pat01	Example Patient01
2024-08-17	14:00-16:00	doc01	Example Doctor01	pat02	Example Patient02
2024-08-18	09:00-11:00	doc01	Example Doctor01	pat03	Example Patient03
2024-08-18	14:00-16:00	doc01	Example Doctor01	pat04	Example Patient04
2024-08-17	11:00-13:00	doc02	Example Doctor02	pat04	Example Patient04
2024-08-18	11:00-13:00	doc02	Example Doctor02	pat06	Example Patient06
2024-08-18	14:00-16:00	doc02	Example Doctor02	pat07	Example Patient07
2024-08-19	14:00-16:00	doc02	Example Doctor02	pat05	Example Patient05
2024-08-17	09:00-11:00	doc03	Example Doctor03	pat08	Example Patient08
2024-08-19	14:00-16:00	doc03	Example Doctor03	pat09	Example Patient09
2024-08-17	09:00-11:00	doc04	Example Doctor04	pat10	Example Patient10
2024-08-17	11:00-13:00	doc04	Example Doctor04	pat11	Example Patient11
2024-08-17	14:00-16:00	doc04	Example Doctor04	pat12	Example Patient12
2024-08-18	11:00-13:00	doc04	Example Doctor04	pat06	Example Patient06



Implementation

Frontend: User: add new patient

HSBC Hospital

Book Patient

View Patient Appointment

Add Patient detail

Add New Patient

+ Add New

All Patients(12)

Patient ID	Patient Name	Actions
pat01	Example Patient01	View
pat02	Example Patient02	View
pat03	Example Patient03	View
pat04	Example Patient04	View
pat05	Example Patient05	View
pat06	Example Patient06	View
pat07	Example Patient07	View
pat08	Example Patient08	View
pat09	Example Patient09	View
pat10	Example Patient10	View
pat11	Example Patient11	View
pat12	Example Patient12	View



Implementation

Backend: Admin functionalities

The screenshot displays the IntelliJ IDEA IDE with the following components:

- Project Explorer:** Shows the project structure with the following hierarchy:
 - src
 - main
 - java
 - com.hsbc
 - dao
 - AdminDaoImpl
 - GsonConfig
 - LocalDateAdapter
 - LoginDao
 - LoginDaoImpl
 - UserDaoImpl
 - DatabaseConnect
- Code Editor:** Shows the implementation of `AdminDaoImpl` in `AdminDaoImpl.java`. The code is as follows:

```
16  
17 public class AdminDaoImpl implements AdminDao { 2 usages  
18  
19  
20     public boolean writeToJson(String fileName, Object data) throws IOException { 5 usages  
21         Gson gson = GsonConfig.createGson();  
22         try (FileWriter writer = new FileWriter(fileName)) {  
23             gson.toJson(data, writer);  
24             return true;  
25         }  
26     }  
27  
28 }
```
- Run Window:** Shows the output of the application. The output is as follows:

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.4\lib\idea_rt.jar=62795:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.4\lib" -jar HospitalSystem.jar  
Welcome to HSBC Hospital!!  
Enter your login Id:  
admin01  
Enter your password:  
Admin12345  
----->  
What operation do you want to perform?  
1. Generate Reports  
2. View details of all Doctors  
3. Check schedule of a doctor  
4. Cancel appointments for a specific doctor  
5. List All Patients  
6. Filter patients by gender
```
- Bottom Bar:** Shows the current file path: `HospitalSystem > src > main > java > com > hsbc > dao > AdminDaoImpl > writeToJson`. The status bar also shows the time (21:45), encoding (CRLF), character set (UTF-8), and indentation (4 spaces).



Implementation

Backend: Importing doctor from json file by admin

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows the project structure with folders 'src', 'main', and 'java'. Under 'java', there is a package 'com.hsbc' containing a sub-package 'dao'. The 'dao' package contains several files: 'AdminDaoImpl', 'GsonConfig', 'LocalDateAdapter', 'LoginDao', 'LoginDaoImpl', 'UserDaoImpl', and 'DatabaseConnect'.
- Editor:** Displays the code for 'AdminDaoImpl.java'. The code is as follows:

```
16  
17 public class AdminDaoImpl implements AdminDao { 2 usages  
18  
19  
20     public boolean writeToJson(String fileName, Object data) throws IOException { 5 usages  
21         Gson gson = GsonConfig.createGson();  
22         try (FileWriter writer = new FileWriter(fileName)) {  
23             gson.toJson(data, writer);  
24             return true;  
25         }  
26     }  
27  
28 }
```
- Run Console:** Shows the output of the application. The user has selected option 8 from the menu. The console output is:

```
6. Filter patients by gender  
7. Filter patients according to the age group  
8. Add a doctor  
9. Exit  
Enter your choice:  
8  
-----  
Enter the filepath:  
doctor.json  
Added to the system successfully  
-----  
What operation do you want to perform?  
1. Generate Reports  
2. View details of all Doctors  
3. Check schedule of a doctor
```
- Status Bar:** Shows the current file path as 'HospitalSystem > src > main > java > com > hsbc > dao > AdminDaoImpl' and the current method as 'writeToJson'. It also shows the time '21:45', encoding 'CRLF', and file format 'UTF-8'.



Implementation

Backend: Doctor functionalities medicines, tests

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows the project structure with folders like `src`, `main`, `java`, `com.hsbc`, and `dao`. The `dao` folder contains `LoginDaoImpl`, `AdminDaoImpl`, `GsonConfig`, `LocalDateAdapter`, `LoginDao`, `UserDaoImpl`, and `DatabaseConnec`.
- Code Editor:** Displays the `Main.java` file. The code defines a `Main` class with a `main` method. It initializes `AdminService` and `UserService` and uses a `while` loop to handle user and doctor login requests. The code is as follows:

```
12 public class Main {
13     public static void main(String[] args) {
14         AdminService adminService = new AdminServiceImpl();
15         flag = 1;
16         while (flag == 1) {...}
17         break;
18         case "User":
19             User user = loginService.getUserByLoginId(loginId);
20             UserService userService = new UserServiceImpl();
21             flag = 1;
22             while (flag == 1) {...}
23             break;
24         case "Doctor":
25             Doctor doctor = loginService.getDoctorByLoginId(loginId);
```
- Run Console:** Shows the output of the program. The output is as follows:

```
3. List All appointments
4. Suggest medical tests
5. Suggest medicines
6. Exit
5
Enter disease name:
Asthma
Spirometry Test
Enter the operation choice:
1. Add schedule
2. Cancel Appointment
3. List All appointments
4. Suggest medical tests
5. Suggest medicines
```




Implementation

Backend: Testcases for login & doctor service

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows the project structure with folders like `resources`, `test`, and `java`. The `com.hsbc` package is selected.
- Editor:** Displays the code for `DoctorServiceImplTest.java`. The code includes package declarations, imports, and test methods using Mockito annotations like `@Mock`, `@InjectMocks`, and `@Before`.
- Run Console:** Shows the execution of the tests. It indicates that all 5 tests passed successfully within 1 second and 19 milliseconds. The output also shows a Java HotSpot(TM) 64-Bit Server VM warning about the bootstrap classpath.

```
package com.hsbc;

import ...

public class DoctorServiceImplTest {

    @Mock 5 usages
    private DoctorDao doctorDao;

    @InjectMocks 5 usages
    private DoctorServiceImpl doctorService;

    @Before
    public void setUp() { MockitoAnnotations.initMocks( testClass: this); }
```

Run DoctorServiceImplTest x

✓ DoctorServiceImplTest (c 1 sec 19 ms) ✓ Tests passed: 5 of 5 tests – 1 sec 19 ms

- ✓ testSuggestMedicines 1 sec 2 ms
- ✓ testGetAllAppointment 5 ms
- ✓ testSuggestMedicalTests 4 ms
- ✓ testAddSchedule 4 ms
- ✓ testCancelAppointment 4 ms

"C:\Program Files\Java\jdk-17\bin\java.exe" ...
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Process finished with exit code 0

HospitalSystem > src > test > java > com > hsbc > DoctorServiceImplTest 1:1 CRLF UTF-8 4 spaces



Implementation

Backend: Testcases for user & admin service

The screenshot shows an IDE window for a project named "HospitalSystem". The file explorer on the left shows the project structure, including a "test" directory with a "java" subdirectory containing the "com.hsbc" package. The main editor displays the code for "AdminServiceImplTest.java". The code includes two test methods: "testPatientsFilteredByGender()" and "testPatientsInAnAgeGroup()". The "Run" button is visible on the left side of the editor. Below the editor, the "Run" console shows the test results for "AdminServiceImplTest", indicating that all 12 tests passed successfully. The console also displays the Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended. The status bar at the bottom shows the current file path: "HospitalSystem > src > test > java > com > hsbc > AdminServiceImplTest > testGeneratePatientVisitsReport".

```
public class AdminServiceImplTest {  
    public void testPatientsFilteredByGender() throws SQLException, ClassNotFoundException {  
        List<Patient> patients = Arrays.asList(new Patient(), new Patient());  
        when(adminDao.patientsFilteredByGender(gender)).thenReturn(patients);  
  
        List<Patient> result = adminService.patientsFilteredByGender(gender);  
        assertTrue("condition: result.size() == 2");  
    }  
  
    @Test  
    public void testPatientsInAnAgeGroup() throws SQLException, ClassNotFoundException {  
        int age1 = 20;  
        int age2 = 30;  
        List<Patient> patients = Arrays.asList(new Patient(), new Patient());  
        when(adminDao.patientsInAnAgeGroup(age1, age2)).thenReturn(patients);  
  
        List<Patient> result = adminService.patientsInAnAgeGroup(age1, age2);  
    }  
}
```

Run AdminServiceImplTest

✓ AdminServiceImplTest (col: 974 ms)

- ✓ testAllDoctors 939 ms
- ✓ testImportDoctor 5 ms
- ✓ testGeneratePatientsByGe 4 ms
- ✓ testGenerateAppointment 3 ms
- ✓ testGenerateAppointment 3 ms
- ✓ testListAllPatients 3 ms
- ✓ testScheduleOfDoctor 4 ms
- ✓ testCancelAppointmentsF 3 ms
- ✓ testGeneratePatientsInAgi 3 ms

✓ Tests passed: 12 of 12 tests - 974 ms

"C:\Program Files\Java\jdk-17\bin\java.exe" ...
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Process finished with exit code 0

HospitalSystem > src > test > java > com > hsbc > AdminServiceImplTest > testGeneratePatientVisitsReport



Implementation

Backend: Testcases for user & admin service

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project named 'HospitalSystem' with a package structure including 'com.hsbc' and 'test'.
- Code Editor:** Displays the 'AdminServiceImpTest.java' file. The code includes a test method 'testPatientsFilteredByGender()' and a test method 'testPatientsInAnAgeGroup()'.
- Run Console:** Shows the execution of 'UserServiceImpTest' with the following output:

```
Tests passed: 3 of 3 tests - 913 ms
testAllAppointments 904 ms
testAddNewPatient 6 ms
testBookAnAppointment 3 ms
Process finished with exit code 0
```



Conclusion

Summary of Achievements:

- Successfully integrated patient management, scheduling, doctor management, admin controls, and reporting.
- Achieved goals of enhancing efficiency, reducing errors, and improving patient care.

Future Enhancements:

- Advanced analytics, mobile access, system integration, and personalized care features.





THANK
YOU