1.INTRODUCTION TO PROJECT

Hospital Management System provides the benefits of streamlined operations, enhanced administration, control, superior patient care, strict cost control and improved profitability. HMS is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. More importantly it is backed by reliable and dependable support. The project 'Hospital Management System' is based on the database, object oriented and networking techniques.

Hospital Management System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required modules and features have been particularly built to just fit in to your requirement. Entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more users friendly and expandable. The package is highly customizable and can be modified as per the needs and requirements of our clients.

The purpose of this project is to automate or make online, the process of day to-day activities like Room activities; Admission of New Patient, Discharge of Patient, Assign a Doctor, and finally compute the bill etc. We have tried our best to make the complicated process Hospital Management System as simple as possible using Structured Modular technique Menu oriented interface.

We have tried to design the software in such a way that user may not have any difficulty in using this package further expansion is possible without much effort. We have used React at our front end for user interface and Spring boot for backend. We have used MySql as our database.

2.REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS

2.1 Admin Account

Admin manages the whole system. He is responsible for employee management.
He is interested into tracking revenue. There are major roles that admin has to handle
Like doctor, receptionist, patient, lab manager. The functionalities are as following.
After login admin can see his/her profile information.

 Appoint employee:

Admin add the detail of employee like name, email, password date of birth, contact number and then select the gender, role and department .

 View, update and remove employee :

When admin click on view employee button he can view the name and role of employee. He can update and also delete the details of employee

 Payment history :

When admin click on Payment history he can view the revenue as per
consultation, ward, test charges in graph format.

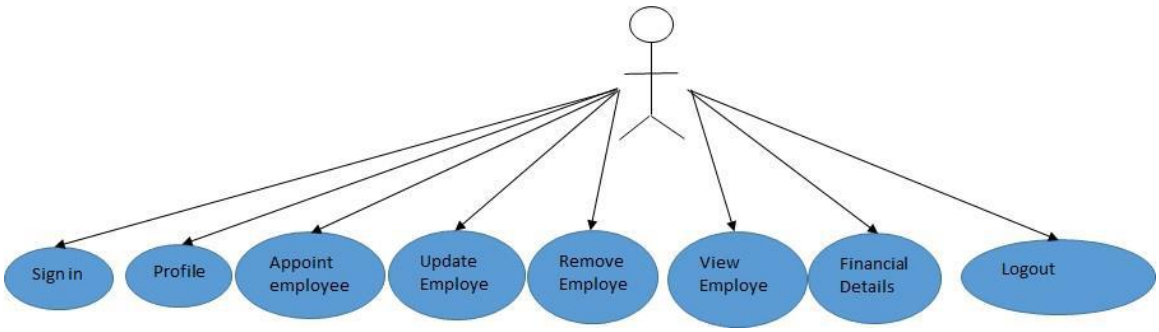


Fig.2.1.Admin Functionalities

2.2 Doctor Account

Doctor can able to see appointments booked by patients. After login he can see his profile as well as following functionalities.

 View appointments :

When doctor click on view appointments he can able to see the appointments for current date only. He can able to see the patient id , appointment date, patient name. Doctor can create prescription for medicine as well as test. While creating medicine he has to select quantity and medicine from list , for creating he can select the test from test list.

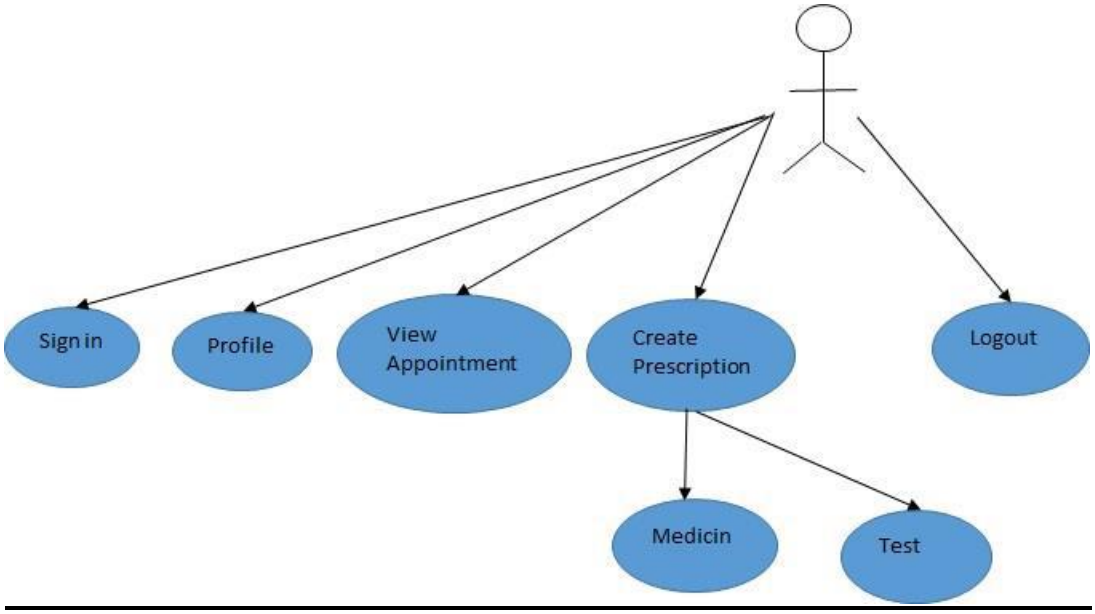


Fig.2.2 Doctor's Functionalities

2.3 Patients Account

 View appointments :

Patient first need to sign up by clicking on book appointment. He/she need to feel the details like name, gender, email and password. After he /she can login and book the appointment.

 Book appointments :

After login patient can select the doctor and particular appointment date, patient need to pay the charges and he will get notification that appointment has booked.

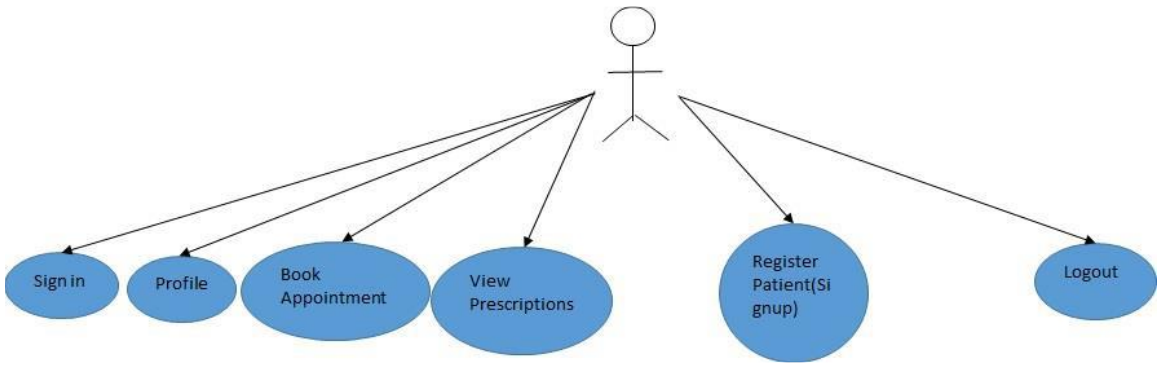


Fig.2.3.Patient's Functionalities

2.4 Receptionist Account

 View appointments :

After login receptionist can see her profile just by clicking on home. she can view the doctor wise and date wise appointments.

 Admit patient :

Receptionist will able to see the details made by patients using book appointments.
According to that list she will be able to admit the patient.

 Discharge Patient :

When the patient is get admitted that list is visible to receptionist and she can discharge that particular patient.

 Accept payment :

As per patient id and appointment she will take ward charges and lab test charges.

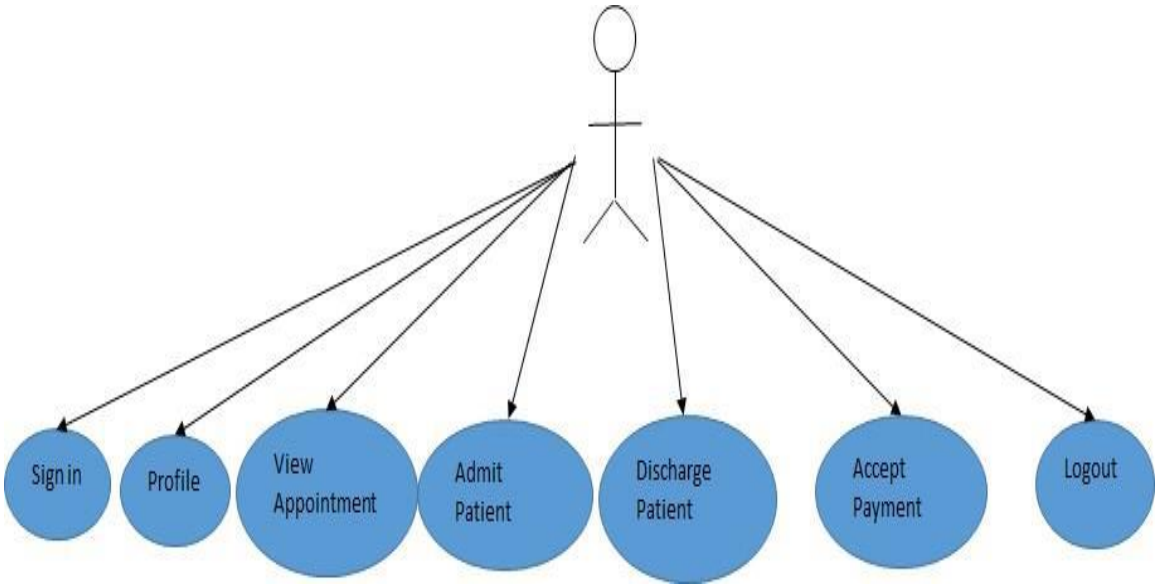


Fig.2.4.Receptionist Functionalities

2.5 Lab manager Account

Lab manager can view the prescription of test by using patient id. He can also able to see list of tests. The test are suggested by doctor to patient are visible to him by view prescription.

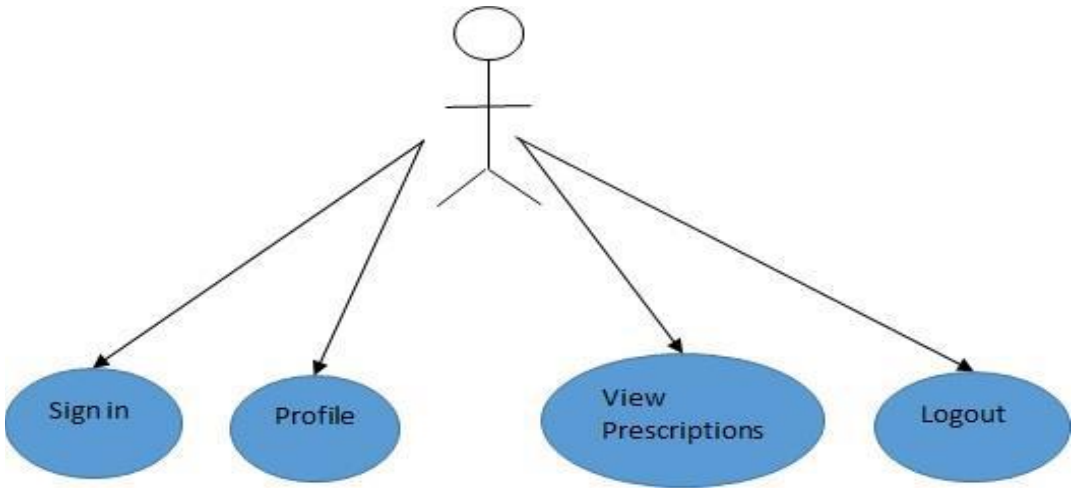


Fig.2.5.LabManager Functionalities

2.2 NON FUNCTIONAL REQUIREMENTS

2.2.1 Interface
Go to Appendix B for user interfaces

2.2.2 Performance
HMS shall be able to handle at least 1000 transactions/inquiries per second.

2.2.3 Constraint
HMS shall be able to handle at least 1000 transactions/inquiries persecond.

2.2.4 Other Requirements:
- **Hardware Interfaces**

The SPMS is expected to function on Intel PIII 900 MHz Processor equivalent or above, 128 MB RAM, 20 GB HDD.

- **Software Interfaces**

The SPMS shall work on MS Windows operating systems family (MS Windows 98, MS Windows NT Workstation, MS Windows 2000, MS Windows XP). It configures to work with Mysql database. This System works on Apache Tomcat server.
We used Spring tool suit(STS) for coding purpose at backend(version: 3.9.18.RELEASE). Visual Studio for front end (version : 1.71.2) and MySql database .

3. DESIGN

3.1 Database Design

The following table structures depict the database design.

Table1: admit_patients

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
ID(primary key)	int	-	0
Admit_date	Date	3	1
discharge_date	Date	3	1
appt_id	int	4	1
dept_id	int	4	1
patient_id	int	4	1

Table2: Appointments

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
ID(primary key)	int	-	0
appt_id	int	4	1
doc_id	int	4	1
patient_id	int	4	1

Table3: Departments

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
ID(primary key)	int	-	0
name	varchar	20	1

Table4: Employees

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
ID(primary key)	int	-	0
Contact	varchar	20	1
email	varchar	20	1
gender	varchar	255	1
name	varchar	20	1
password	varchar	20	1
dob	Date	3	1
role	varchar	255	1
dept_id	int	4	1

Table5: med_presc

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
ID(primary key)	int	-	0
presc_date	date	3	1
doc_id	int	4	1
patient_id	int	4	1

Table6:medical_tests

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
ID(primary key)	int	-	0
charge	double	8	0
name	varchar	20	1
Test_presc_id	int	4	1

Table7:medicine_item_qty

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
qty	int	-	0
med_presc_id(primary)	int	4	0
Med_id(primary)	int	4	0
id	int	4	0
Medicine_id	int	4	1

Table8:medicines

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
ID(primary key)	int	-	0
name	varchar	20	1
Price_per_tab	double	8	0

Table9:patients

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
ID(primary key)	int	-	0
Contact	varchar	20	1
email	varchar	20	1
gender	varchar	255	1
name	varchar	20	1
password	varchar	20	0
age	int	4	0
Blood_group	varchar	10	1
city	varchar	20	1

Table10:payments

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
ID(primary key)	int	-	0
consultation_charges	double	8	0
test_charges	double	8	0
ward_charges	double	8	0
appt_id	int	4	0
patient_id	int	4	0
Last_pay_date	int	3	1

Table11:test_presc

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
ID(primary key)	int	-	0
presc_date	date	3	1
doc_id	int	4	1
patient_id	int	4	1

Table12:test_reports

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
ID(primary key)	int	-	0
content_type	date	3	1
data	longbolb	-	1
name	varchar	255	1
med_test_id	int	4	1
test_presc_id	int	4	1

Table13:test_reports

Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
Presc_id(primary)	int	4	0
test_id(primary)	int	4	0

E-R Diagram:

Go to Appendix A

4. CODING STANDARDS IMPLEMENTED

Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

Identifier	Case	Examples	Additional Notes
Class	Pascal	Employees, Patients	Class names should be based on "objects" or "real things" and should generally be **nouns**. No '_' signs allowed. Do not use type prefixes like 'C' for class.
Method	Camel	getId,getRole	Methods should use **verbs** or verb phrases.
Parameter	Camel	patientId	Use descriptive parameter names. Parameter namesshould be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios.
Interface	Pascal with "I" prefix	IDoctorService	Do not use the '_' sign
Property	Pascal	Role, Gender	Use a noun or noun phrase to name properties.
Exception Class	Pascal with "Exception" suffix	WebException,	

Comments

- Comment each type, each non-public type member, and each region declaration.
- Use end-line comments only on variable declaration lines. End-line comments arecomments that follow code on a single line.
- Separate comments from comment delimiters (apostrophe) or // with one space.
- Begin the comment text with an uppercase letter.
- End the comment with a period.
- Explain the code; do not repeat it.

5. TEST REPORT

GENERAL TESTING:

SR-NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
1	Register Page	Redirected to Next page	OK	Nothing
2	Login Page	Pop-up will come	Ok	Please enter username and password again .
4	Sign in of patient	Need to fill the information and password	Ok	Nothing
5	Booking appointment	All the fields should be filled for submission	Ok	Nothing
6	Checking login or not	User is logged in or not	Ok	Nothing
7	Edit profile for patient	Update the information	Ok	Nothing
8	All profile of employees and patients	Visible after login	Ok	Nothing
9	Payment add	After adding payment result is payment booked	Ok	Nothing
10	Admin add and remove employee	Data gets added into database according tothe role	Ok	Nothing
11	View test reports	Lab manager can view test list according to patient id	Ok	Nothing
12	Create prescription by doctor	List of medicines and test gets added	Ok	Nothing
13	Revenue generation	When payment gets added into account itis visible according totype of charges in graph	Ok	Nothing
14	View appointment of doctor	Only todays appointment is visibleto doctor	Ok	Nothing
15	View appointment of receptionist	Doctor wise and date wise appointments are visible to receptionist	Ok	Nothing

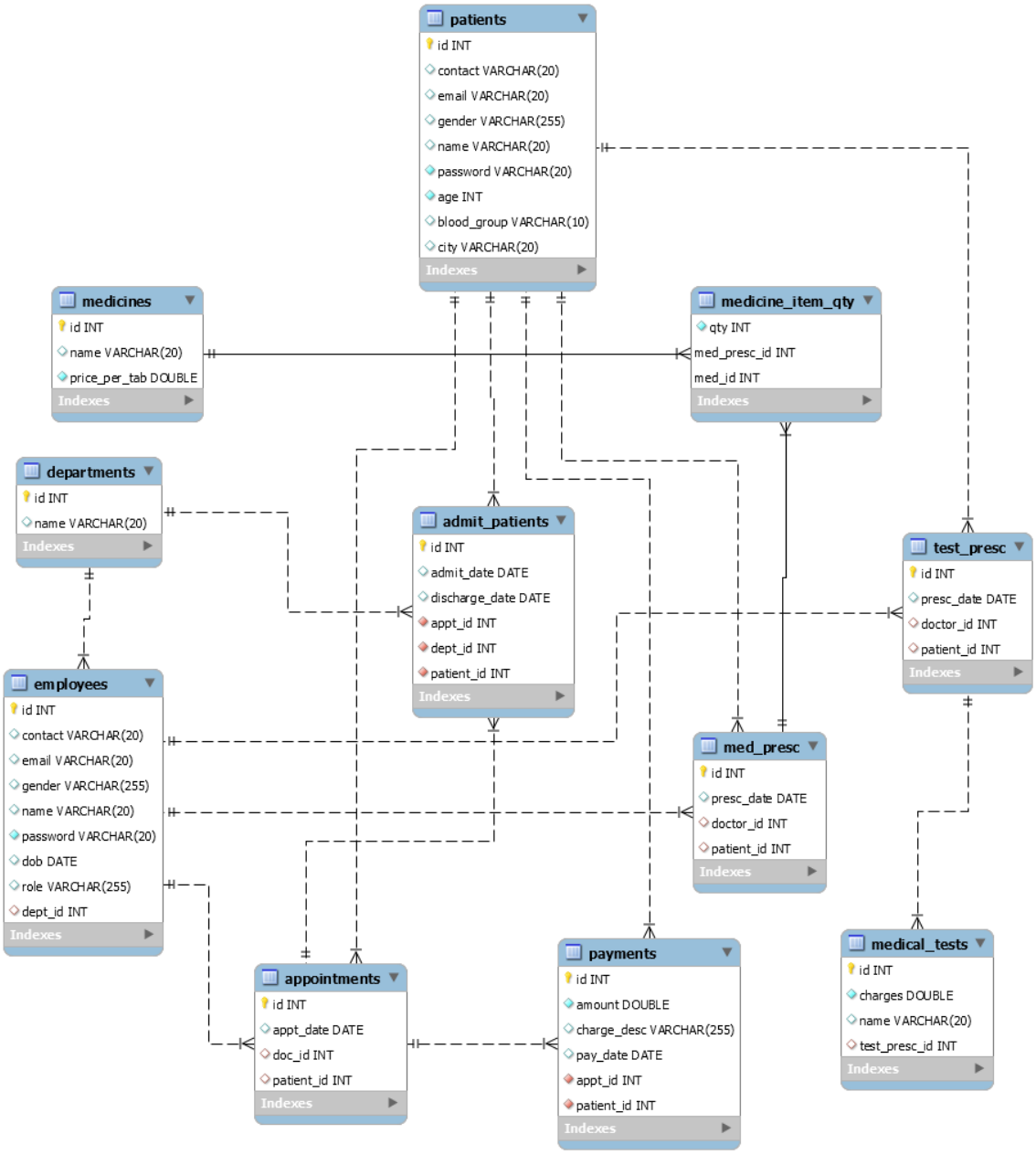
16	Accept payment by receptionist	Ward charges and test charges get accepted	Ok	Nothing
17	Discharge patient	After clicking on discharge patient data get removed from database	Ok	Nothing
18	Admit patient	She can able see the booked appointmentand select the date and department and can admit it	Ok	Nothing
19	Logout	It will logout from user profile.	Ok	Nothing
	STATIC TESTING			
SR-NO	**Deviation**	**Program**		
	Commenting not followed	All Web Application		

6. PROJECT MANAGEMENT RELATED STATISTICS

DATE	WORK PERFORMED	*SLC Phase*	Additional Notes
MAY 6, 2022	Project Allotment and User Requirements Gathering	Feasibility Study	We discussed about project to know the requirements.
MAY10,2022	Initial SRS Document Validation And Team Structure Decided	Requirement Analysis (Elicitation)	The initial SRS was presented to the client to understand his requirements better
MAY 22,2022	Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram and User Interfaces	Requirement Analysis & Design Phase	Database Design completed
JUN 19,2022	Business Logic Component design Started	Design Phase	----------
SEPT 5,2022	Coding Phase Started	Coding Phase	70% of Class Library implemented.
SEPT 8,2022	Implementation of Web Application and Window Application Started	Coding Phase	Class Library Development going on.
SEPT 10, 2022	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	Class Library Modified as per the need.
SEPT 12, 2022	The Project was Tested by the respective Team Leaders and the Project Manager	Testing Phase (Module Testing)	--
SEPT 19, 2022	The Project was Submitted to Other Project Leader of Other Project Group For Testing	Testing Phase (Acceptance Testing)	The Project of Other Team was Taken up by the Team for Testing

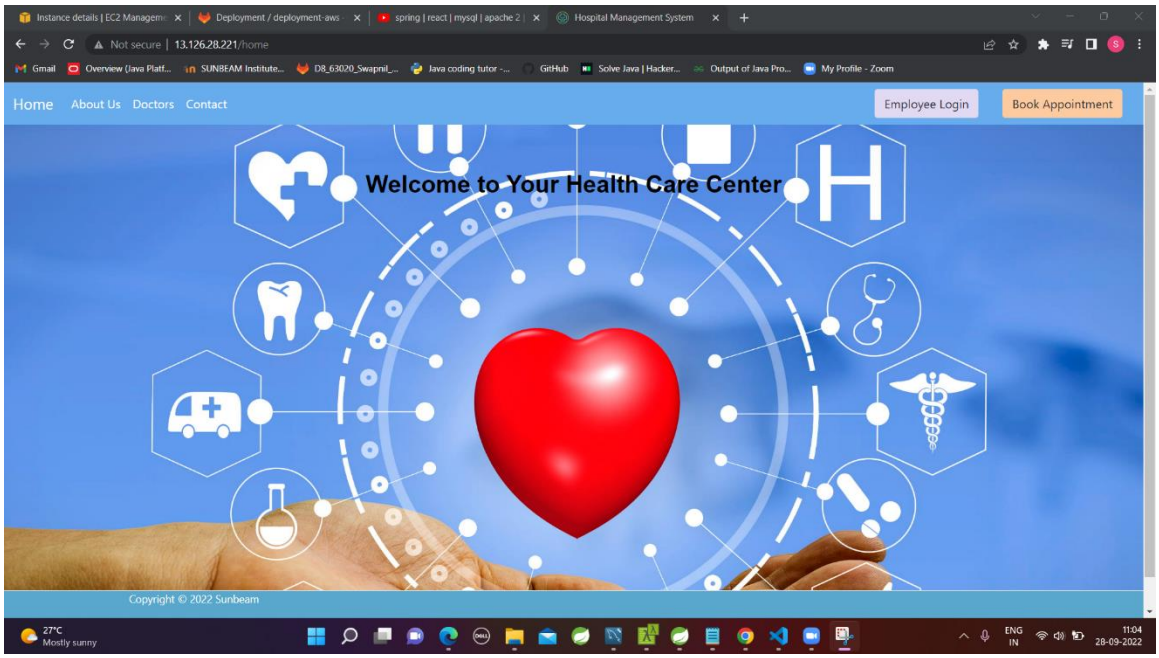
SEPT 28, 2022	The project was deployed on AWS	Deploying	The Project was complete for submission
SEPT 28, 2022	Final Submission of Project		

Appendix A
Entity Relationship Diagram

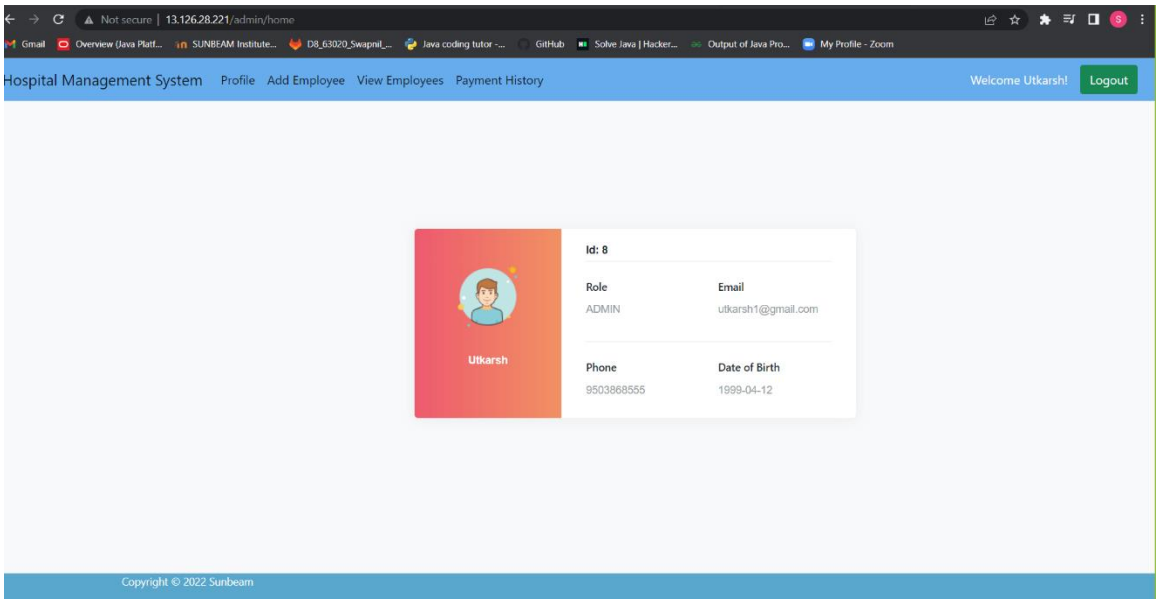


Appendix B

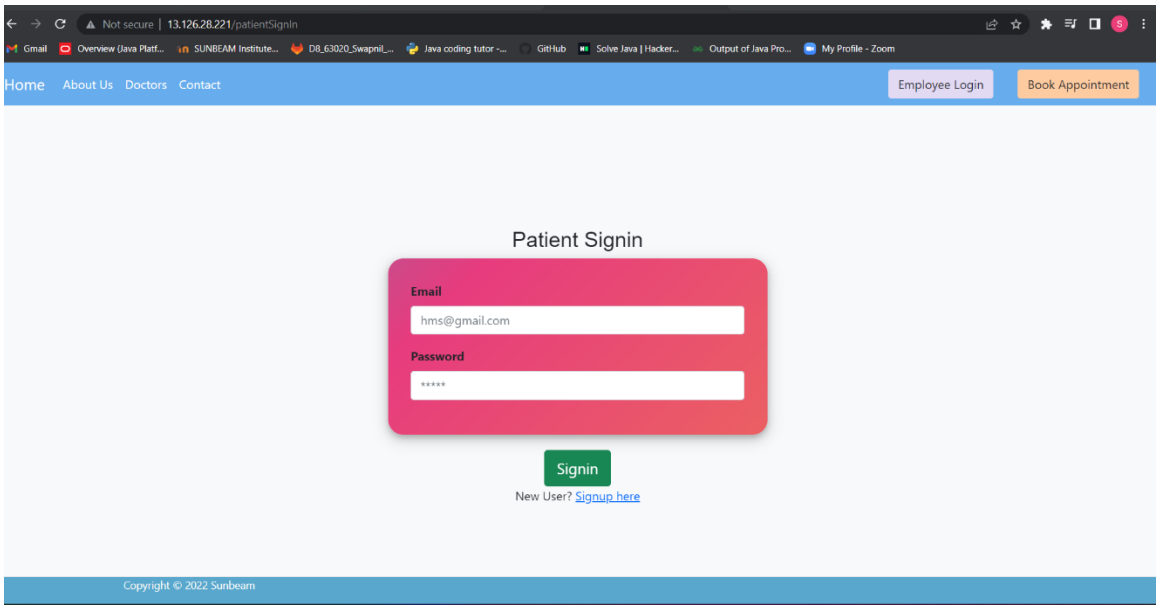
Homepage:



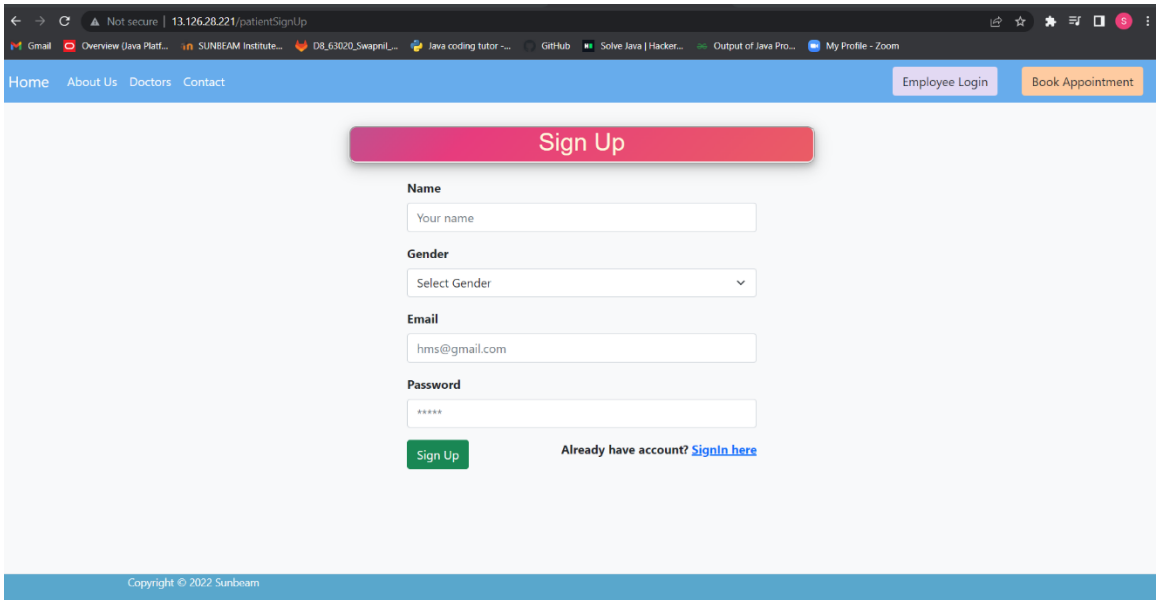
Admin_HomePage:



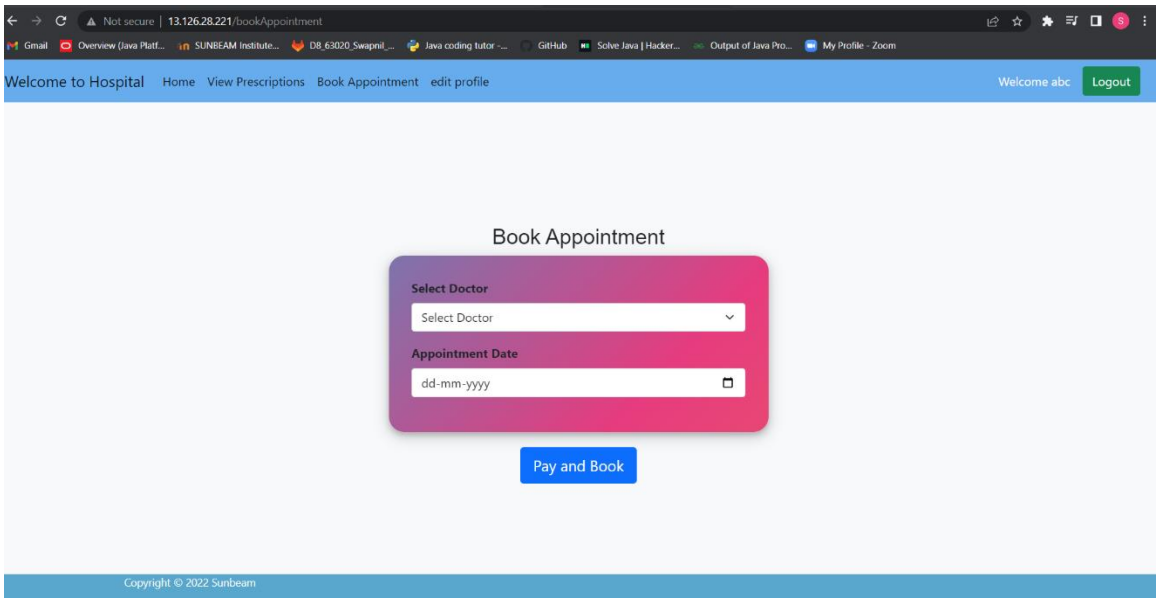
Patients Signin:



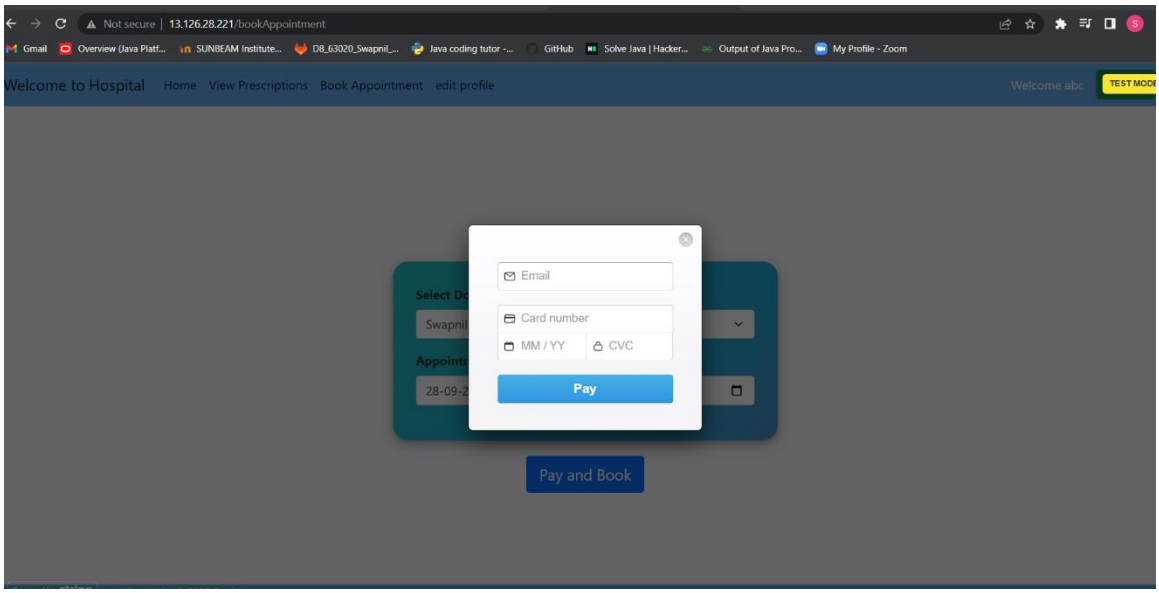
Patients SignUp:



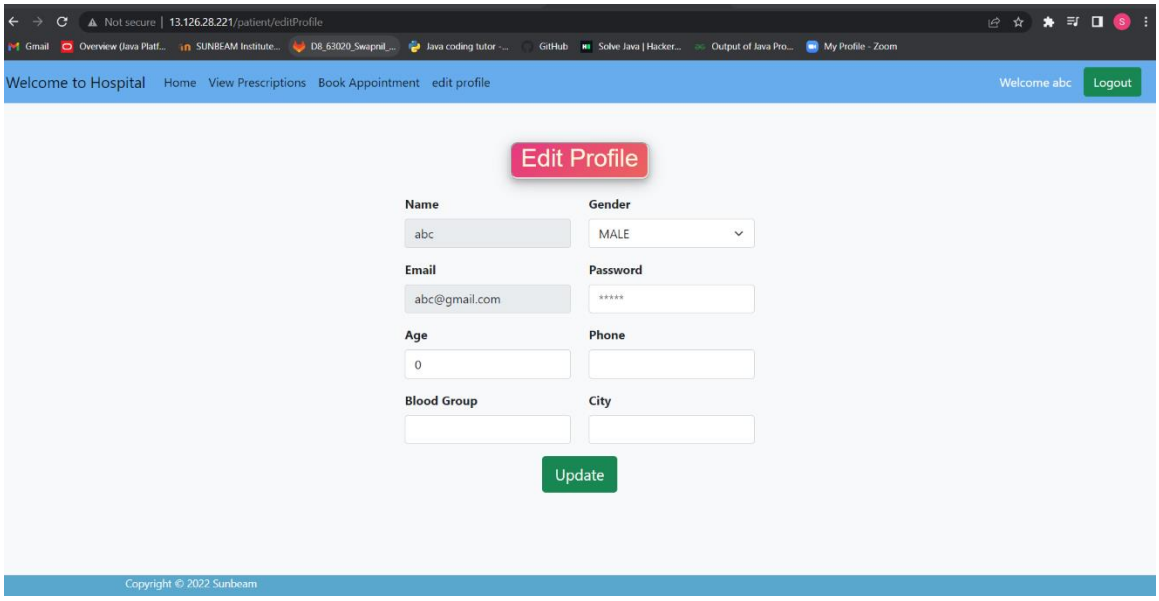
Book appointment:



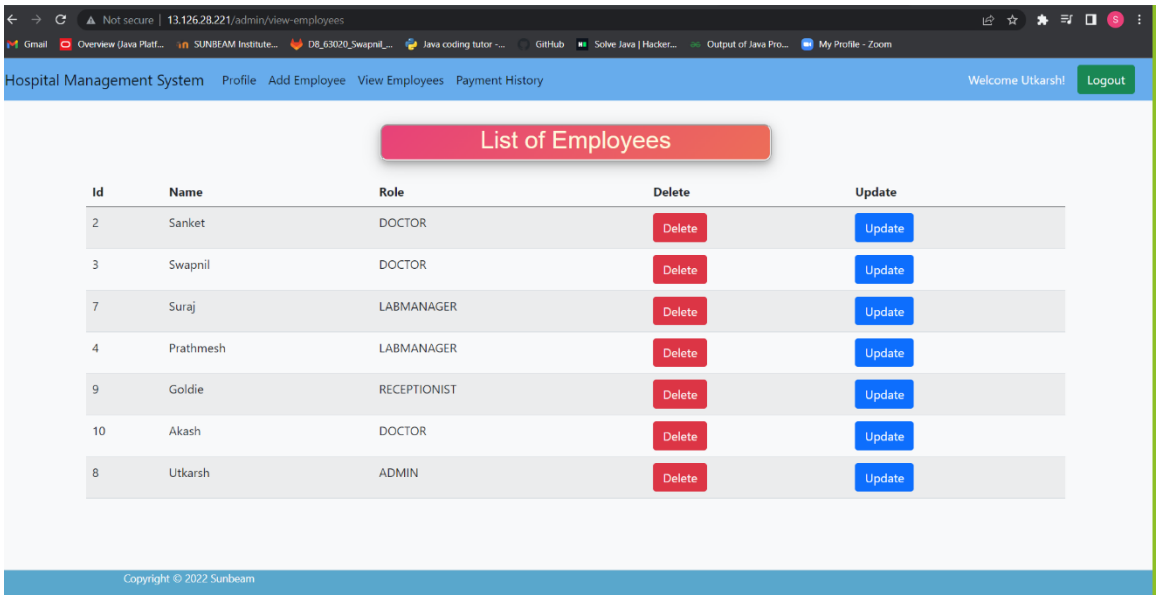
Payment successful:



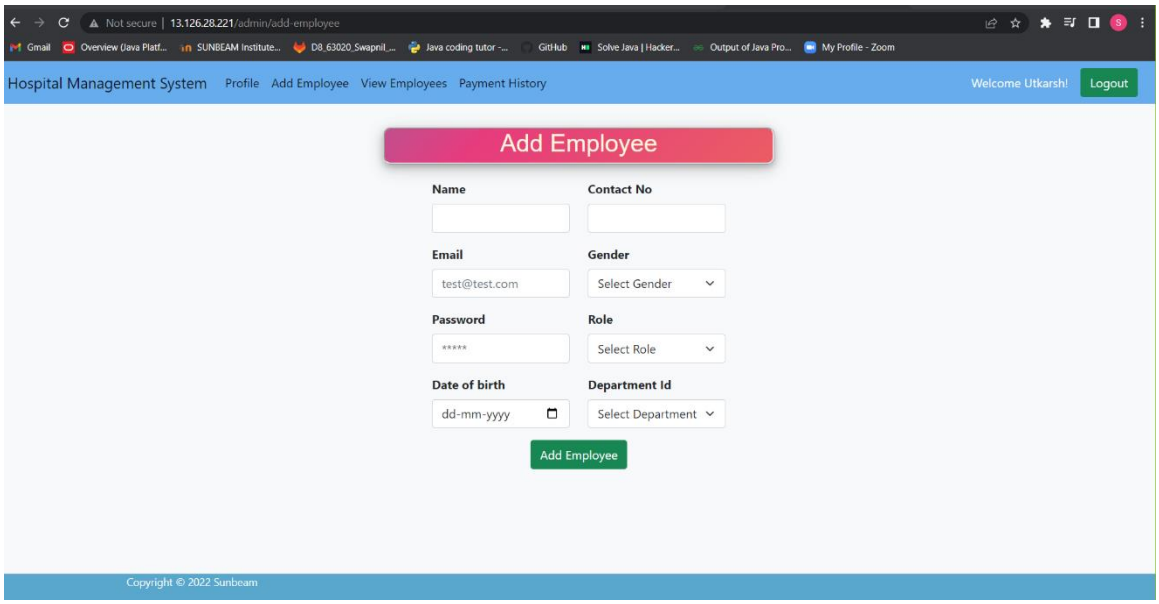
Edit_Patients_Profie:



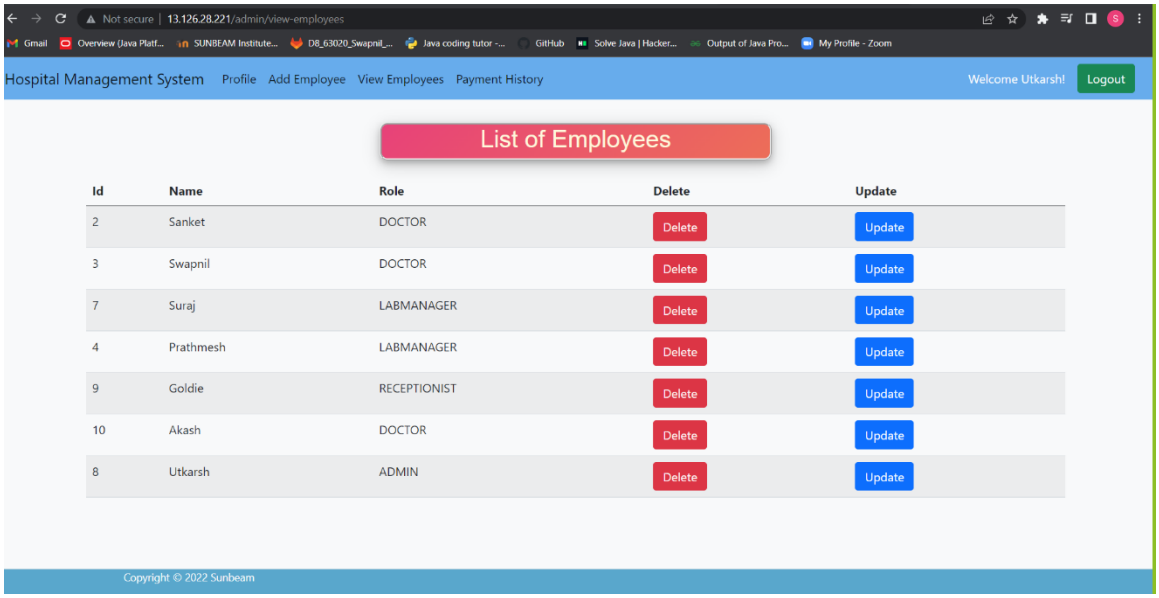
View Employee By Admin:



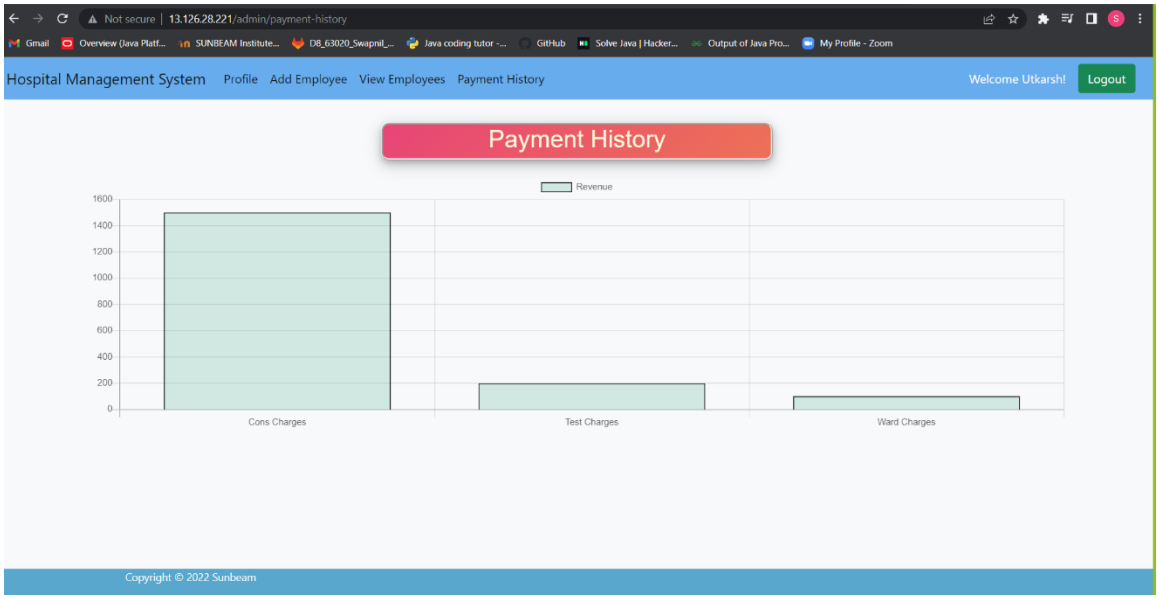
Add employee at admin login:



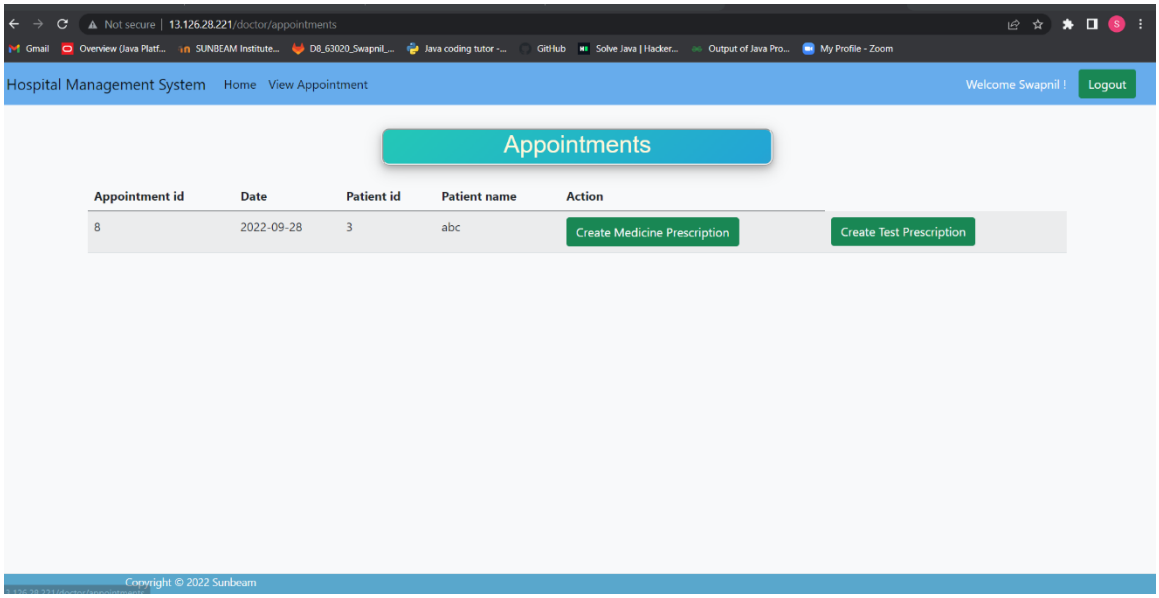
Remove employee by Admin:



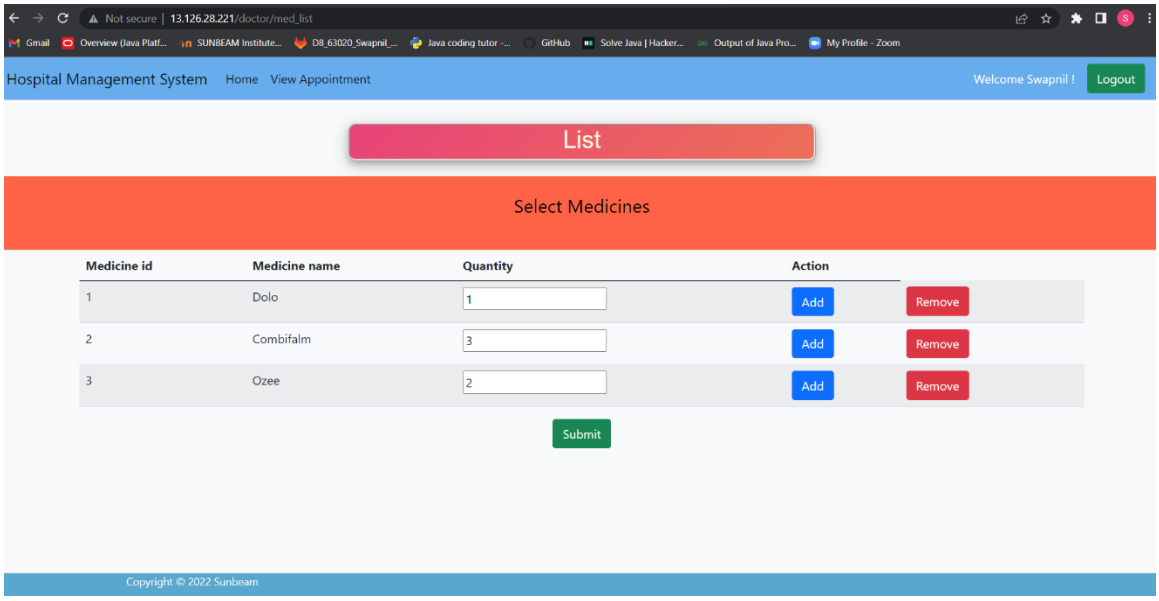
Revenue generation at admin login:



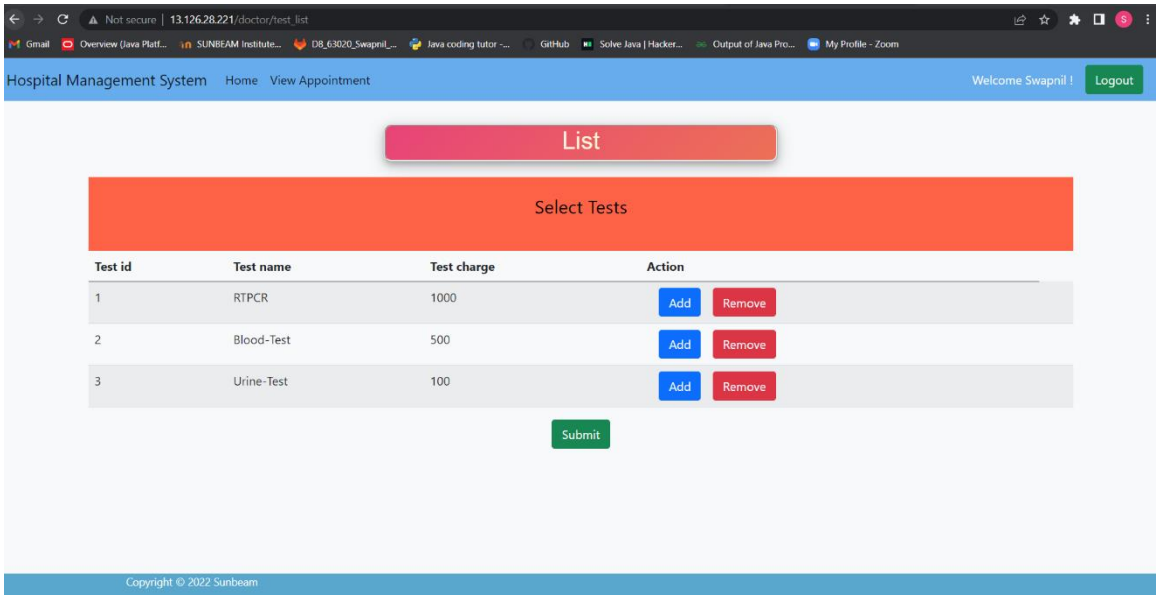
Doctors view appointment:



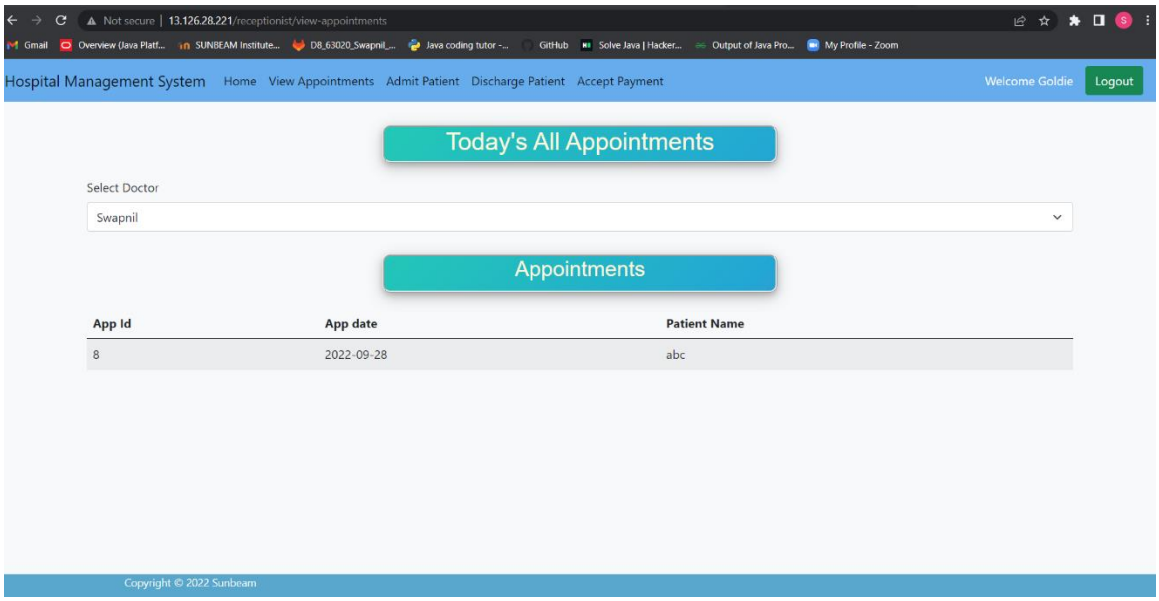
Doctor create medicines prescription:



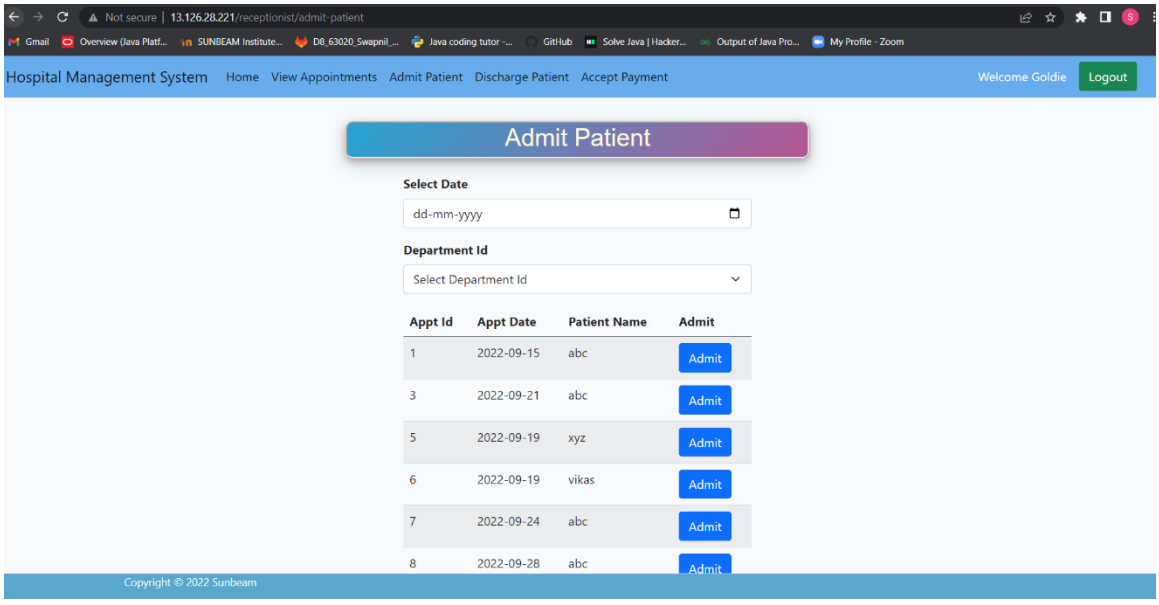
Doctor create tests prescription:



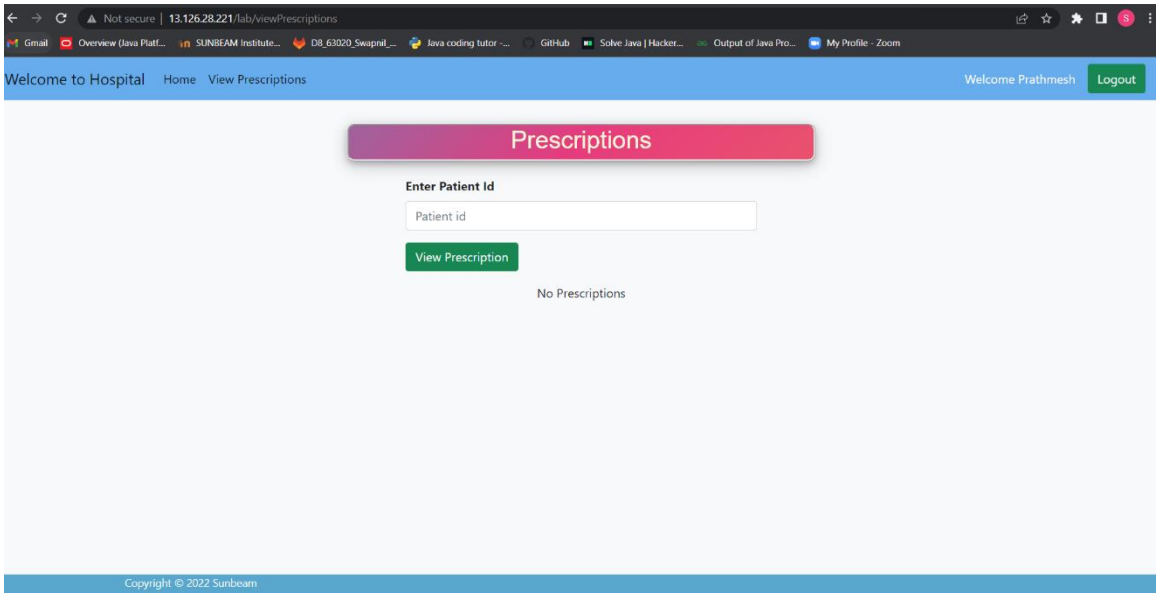
Receptionist's_view_appointment:



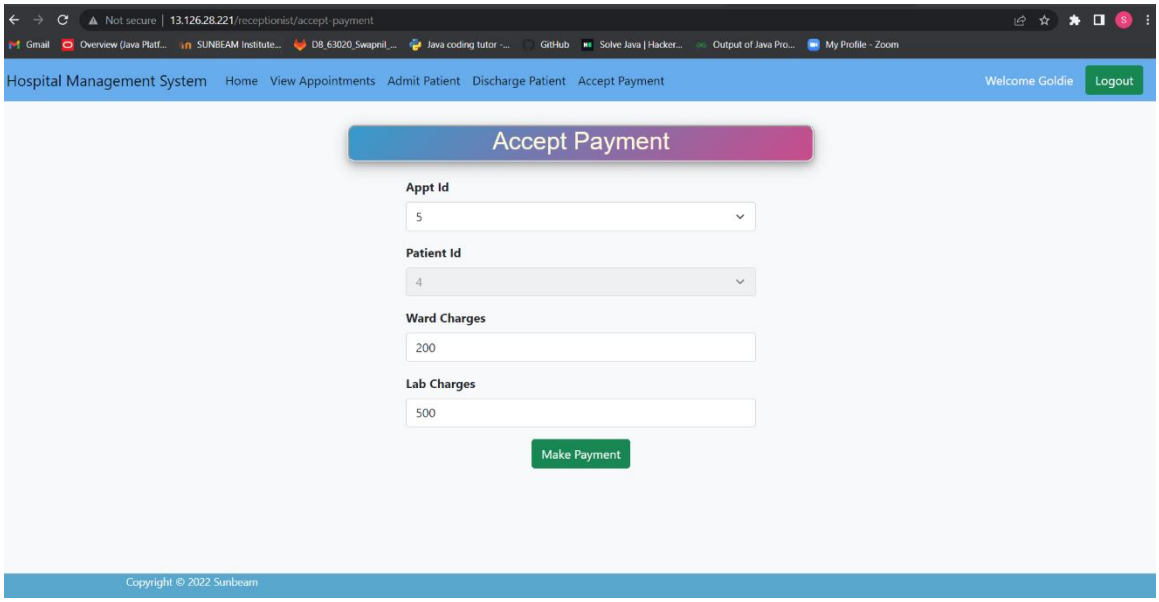
Admit_patient:



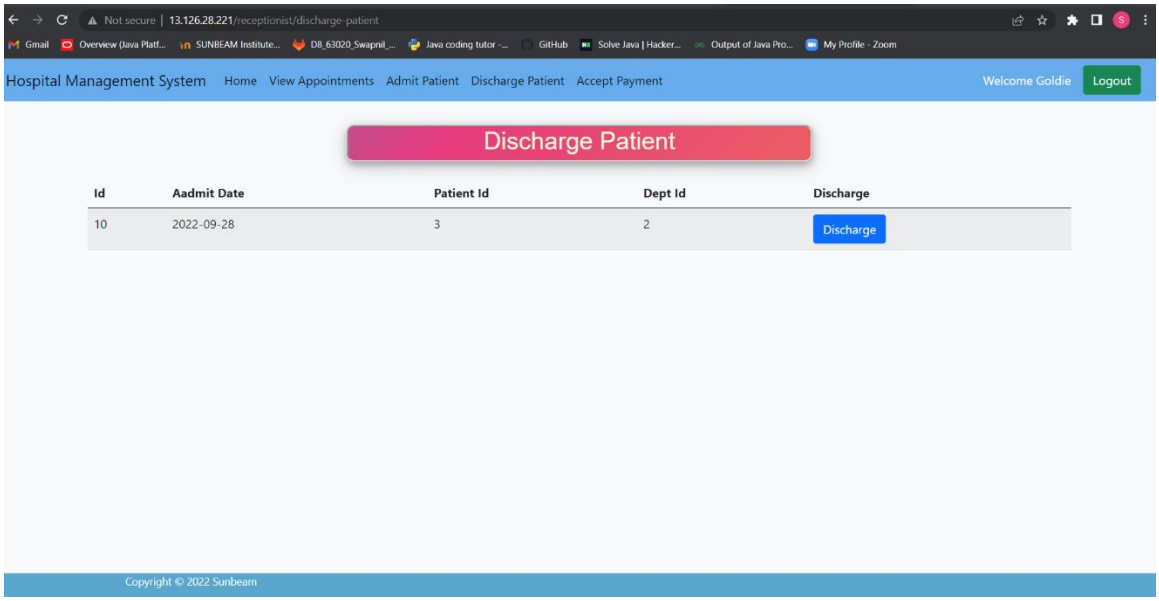
Lab_manager_view_prescription:



Accept_payment_by_receptionist:



Discharge_Patient:



7.REFERENCES:

http://www.google.com
http://www.w3.org
http://www.wikipedia.org
http://www.delta.com