COURSE : Software Engineering

COURSE CODE : CSE-1005

SLOT: L25 + L26

**** *LAB REPORT* ****

Software Engineering Project Scope Document

Project Title: EasyMed-Hospital Portal

Team ID: 3

Project Manager: Dr.Subrata Tikadar (Assistant Professor)

♣ Identify your Team Members and Project Title.

Roles and Responsibilities

Name	Registration Number	Designation
G.Sree Harsha	19BCD7114	Team Leader
P.P.K.Satya Raj	19BCD7041	Team Member
K.Chanakya	19BCN7141	Team Member
Venkata sai pavan kalyan	19BCE7414	Team Member

T.D.Srikar	19BCN7240	Team Member
Hruthik Yadav	19BCN7092	Team Member
Rohith	19BCE7303	Team Member
K.Vamsi Krishna	19BCE7438	Team Member
N.Bhanu	19BCE7473	Team Member
B.Adarsh	19BCD7166	Team Member
B.Pavan	19BCE7446	Team Member

CONTENTS

>	one/two-page wri	teup	(4-5
>	Specify and Gath the required deta	er the requirements for you	our chosen project with
	1. PROBLEM AN	NALYSIS	
	1.1.	Overview of the project:	
		Why computerized?	(6-7)
	1.2.	Identification of project sc	ope:
		Task involved:	(8)
	1.3.	Objectives	(8)
	1.4.	Infrastructure	(9)

> Specify and An the required do	•	for your chosen project with
2. SOFTWARE	REQUIREMENT ANALY	YSIS AND PLANNING
2.1 Descript	ion of individual phase/mo	odule:
2.1.1	User characteristics	(9-10)
2.1.2	General constraints	(11)
2.1.3	Assumption:	
	Dependency:	(11)
2.1.4	Functional requirements:	-Input, Output, Description
		(11-12)
2.2 Identify	individual module deliver	rables (13-14)
Perform Design	n and Analysis of require	ements for your chosen
Q	ne required UML diagram	•
3. DATA MODE	ELING	
3.1 System A	Architecture Design	(14-15)
3.2 Use Case	e Diagram	(16-17)
3.3 Activity	Diagram	(18-23)
3.4 DFD Dia	0.040.42	
	agram	(24-26)
3.5 Class Di		(24-26) (27-28)
3.5 Class Di		,
	agram n and Development of Da	,
> Perform Desig	agram n and Development of Da :	(27-28)

5.1 Test Plan	5. SOFTWA	ARE TESTING	
Perform a demonstration of your chosen and developed project:6. PROJECT DEMO	5.1	Test Plan	(32-35)
6. PROJECT DEMO	5.2	Test Results & Debugging	(36)
	Perform a	demonstration of your chose	n and developed project:
6.1 Screen shots	6. PROJECT	ΓDEMO	
	6.1	Screen shots	(37-55)

♣ Perform a Feasibility Study on the Selected Project and prepare a one/two-page writeup.

> Project Objectives:

The main objectives of this project is to implement an end-to-end database focused portal for patient registration organizing information and fixing appointment with clinic doctor in VIT-AP.

> Programming Languages Required:

Java, MySQL, HTML, CSS.

Platforms to be Used:

Android Studio, Eclipse, lucid chart, VS Code.

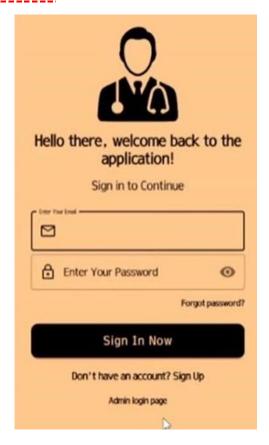
Problem Statement:

Hospital is associated with the lives of common people and their day-to-day routines. The manual handling of the record is time consuming and highly prone to error. 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 a clinic in VITAP, but Students hardly use it. If we automate the process and make a management system that would help our clique.

> Reference-Already Existing Work:

Hospital Management System (HMS) is designed for multispecialty hospitals, to cover a wide range of hospital administration and management processes of patient-centric system. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care (medical records management and billings), and hospital administration, in a seamless flow. In existence, some researchers have contributed positively in the improving of health care institutions management systems. Therefore, we discuss in this section below, some of the related works done in hospital management system by researchers in the field.

Outcome Expectation:



> Improvements:

We want to further improve our project by adding GOOGLE MAPS API'S which adds the nearest medical hospitals details. We want to even add Retail management System in this app as it is similar to execute if time permits.

Conclusion:

There are 2 reasons to choose this project:

- a) We want to gain an insight into the working of the HOSPITAL. This problem statement is a typical real-world situation and executing it hopefully provides rewarding experience with good results for our team.
- b) we want to increase our understanding of database design and innovate user friendly app with genuine effort.

♣ Specify and Gather the requirements for your chosen project with the required details below:

1. PROBLEM ANALYSIS

- 1.1. Overview of the project:
 - Why computerized?
- **1.2.** Identification of project scope:
 - Task involved:
- 1.3. Objectives
- 1.4. Infrastructure

1. Problem Analysis:

***** 1.1 Overview of the project:

Hospital Management System (HMS) is designed for multispecialty hospitals, to cover a wide range of hospital administration and management processes of patient-centric system. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care (medical records management and billings), and hospital administration, in a seamless flow.

We intend to develop an app which maintains activities of our college clinic. The supposed users of a hospital management system may be divided into three categories:

- 1. Hospital administration-who manages billing process & views Appointments
- **2.** Doctors and other authorized employees: Prescribe Medicines for patients & authorized to modify activities of other two users.
- **3.** Patients (Facility and students)- Registered to receive medical treatment and pays bill.

Why Computerized?

Manual Records of patient history is time consuming as well as Error prone, so we intend to develop a computerized automated Management System. Our present modern information system makes use of computers for the execution, each of them connected through an optimized network.

Considering our clinic, many students enter and exit the hospital in a day and maintaining their records safely is tedious. To reduce this type of burdens and to manage the financial, hospital administration and clinical aspects, Hospital management system would automate the process and make it easier. Demerits of manual Record maintenance is:

- Time Consuming
- More Cost
- Error Prone
- No Data Security & No Retrieving Ability
- No Patient Details for further communication

By computerizing we can overcome all these complications. The computerization of the system has speed up the process. In the current system, the front office managing is very slow. Human Errors and Manual mistakes are removed thus providing Reliable software for best use.

We also intend to add API Based Google Map for Knowing nearest medical shops near our college.

1.2 Identification of project scope:

- User friendly interface
- Easy to access patient details

Task involved:

- Feasibility study
- Implementation of Security system for Doctors and Hospital Administration
- Database management system
- API Based Google Map
- Password and Login Management System

1.3 Objectives:

Uncomplicated appointments:

- To provide an easy and error free schedule for doctors and patients (students and faculty) of VITAP.
- To provide retrieving ability and decrease the complexity of manual maintenance of patient records.
- To provide appointment confirmation to all other users (respective patient and doctor) on appointments approved by hospital administration.

Mobile alerts:

- To provide notification alerts on confirmation of appointment and a remainder at time of schedule.
- To provide a clinical report for a patient after the treatment provided by a doctor.
- To provide medical prescriptions suggested by a doctor to a patient and respective bill with consultation fee.

Helpdesk:

- To provide complaints, suggestions to the VITAP administration.
 - For example: Regarding billing process, Treatment, hygiene.

1.4 Infrastructure:

These are the software which would be used border to create our project:

- Android Studio Arctic Fox | 2020.3. 1.
- Visual Studio 2019 version 16.7.12
- Star UML/Argo UML
- My SQL Database
- **♣** Specify and Analyze the requirements for your chosen project with the required details below:
 - 2. SOFTWARE REQUIREMENT ANALYSIS AND PLANNING
 - 2.1 Description of individual phase/module:
 - 2.1.1 User characteristics
 - 2.1.2 General constraints
 - 2.1.3 Assumption:

Dependency:

- **2.1.4 Functional requirements:** -Input, Output, Description
- 2.2 Identify individual module deliverables
- 2. Software Requirement Analysis and Planning:
 - **➤ 2.1 Description of individual phase/module:**
 - **2.1.1 User Characteristics:**
 - **4** Admin:

Admin plays vital role in functioning of Hospital management System. They oversee organizational part of hospital. Either working in a team or independently, they make sure a medical facility is employing effective and efficient practices that deliver the best care possible. What a doctor is to a patient, a hospital admin is to a medical facility.

• Functionalities are:

- Edit Services
- Edit Doctor Details
- Edit patient details

Receptionist:

Confirms appointment of patients on confirmation with doctor.

• Functionality is:

• Confirm Appointment

Patient :

Patient receives services on confirmation by doctor. previous records of their prescription can also be seen by them.

• Functionalities are:

- Consult doctor
- Make Appointment
- Edit Patient Details

Lesson Doctor :

Provides clinical support to patient. The disease history, test results, prescribed treatment can be accessed by doctors without much delay in order to make an accurate diagnosis and monitor the patient's health. They can modify their slots on confirmation with receptionist.

• Functionality is:

• Edit Doctor details

2.1.2 General Constraints:

- ♣ User id of patient should match with OTP sent to patient's registered mobile number, then logging in is allowed.
- ♣ Patient is restricted to view their respective appointments and previous records of their prescription.
- ♣ Doctors are permitted to view their respective patient details and their schedule.

2.1.3 Assumption:

- Livery user knows English
- ♣ All users are equipped with mobile phones

Dependency:

- Patient can only see appointment confirmation after authentication from receptionist on confirmation with doctor.
- Admin of hospital can control, regulate and monitor all executional activities and primary services of hospital

*** 2.1.4 Functional Requirements:**

4 Login

Description:

It displays the welcome notes for four users.

- **Input:** Username and password
- **Output:** Depending upon the input, the homepage displays for the user.

Patient login

Description:

Shows available appointment slots and also disease description to doctor. Previous records of visit are also shown.

- Input: available Services with appointment slots and also disease description
- Output: Confirmation regarding appointment and also billing with disease report after examination.

Doctor login

Description:

Shows timeslots of patient's appointment. patient's medication report and list of medicines are also available for each patient who is visiting hospital.

- Input: Writes Prescription after examination. can send a message to Receptionist to modify any slots.
- **Output:** appointment slots on that day are displayed.

Receptionist login

Description:

Fixes appointment for patient examination by doctor. Also, Complete report after examination is sent to patient.

- **Input:** Report after prescription.
- Output: Appointment slots on that day are displayed.

Admin login

Description:

Shows all other user's (patient, doctor and receptionist) details after opening admin page.

• **Input:** Available to modify services, appointments.

• Output: NILL.

> 2.2 Identify Individual Module Deliverables:

Login Module:

♣ Login id and password is given as input for this page. if the password or login is correct then concerned homepage is shown for respective user.

Patient Module:

- **♣** End user of hospital management system.
- ♣ Available appointment slots are displayed to choose
- Confirmation regarding appointment is received to patient with disease report after examination.
- ♣ In case of any queries then complaint is registered in the portal.

Doctor Module:

- Appointment slots on that day are displayed
- **♣** Writes medication and prescription after examination.
- ♣ In case of any queries then complaint is registered in the portal.
- ♣ Previous are records of patients are also shown (if time permits) and a simple memo to request receptionist for any modification.

Receptionist Module:

Schedules appointment for patient consultation with doctor.

- **♣** Confirmation is sent to patient after appointment confirmation.
- Modifies schedule if requested by doctor.

Admin Module:

- ♣ Access to update and modify Services, timeslots to all other users.
- ♣ Receives complaints and solves all the issues.

♣ Perform Design and Analysis of requirements for your chosen project with the required UML diagrams:

3. DATA MODELING

- 3.1 System Architecture Design
- 3.2 Use Case Diagram
- 3.3 Activity Diagram
- 3.4 DFD Diagram

3. DATA MODELING:

> 3.1 System Architecture Design:

A system architecture or systems architecture is the conceptual model that defines the structure, behavior and more views of a system.

As architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

THREE-TIER ARCHITECTURE

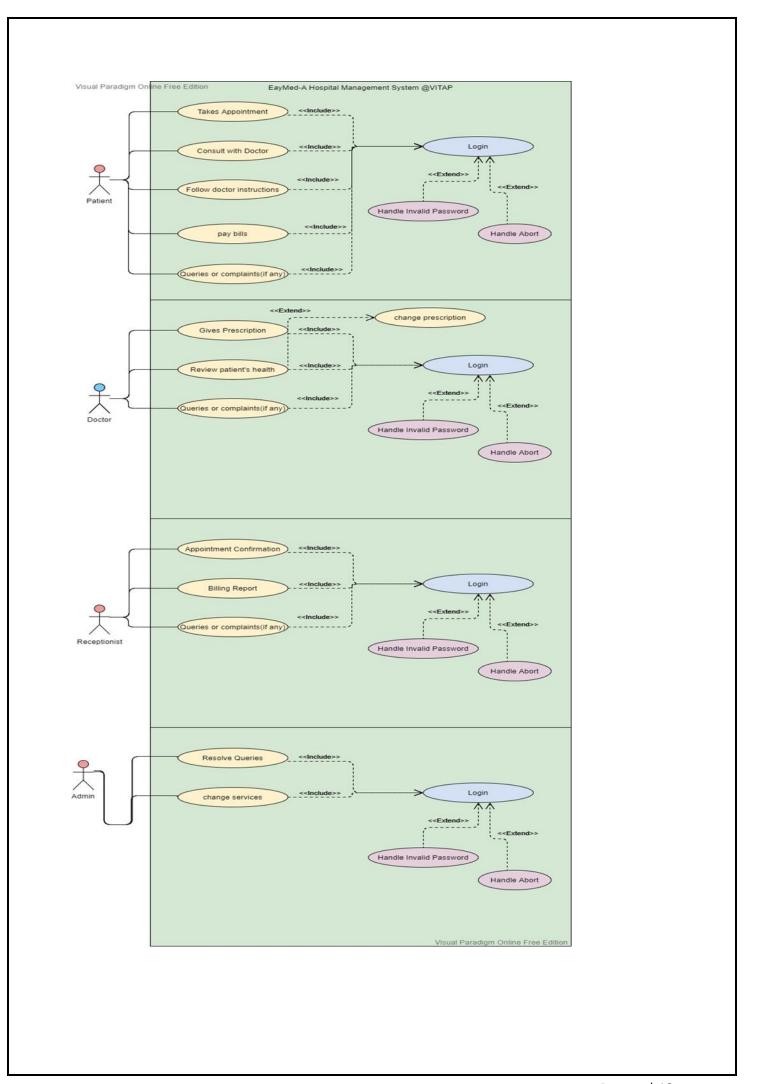
- Three-tier architecture is an architectural deployment style that describe the separation of functionality into layers with each segment being a tier that can be located on a physically separate computer.
- They evolved through the component-oriented approach, generally using platform specific methods for communication instead of a message-based approach.
- The network architecture used in this type is three tier client-server architecture and distributed database architecture.

Doctor Patient Receptionist Admin User Interface Schedules and Request Appointment Diagnose patients Provide Services and confirms and write prescription and pay bills. solves Queries Appointments Acess/Update Acess/Update Acess/Update Bussiness Interface Acess/Update Database Database

> 3.2 Use Case Diagram:

A use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagramdefined by and created from a use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

- Doctor produces medication report and list of medicines are also available for each patient who is visiting hospital.
- The patients' information must be composed of their basic data, sickness complaints or consultations including admission. These data were then recorded and given to the appropriate physician for curing and basis for the kind of services to be done.
- Admin or the main user of the system could have access into the patient information in terms of availing the hospital services.
- Hospital Reception module supports some of the many job duties of hospital receptionist. Receptionist schedules patient's appointments and admission to the hospital, collects information from patient upon patient's arrival and/or by phone. For the patient that will stay in the hospital ("inpatient") she or he should have a bed allottedin a ward. Receptionists might also receive patient's payments, record them in a database and provide receipts, file insurance claims and medical reports.



> 3.3 Activity Diagram:

A UML activity diagram helps to visualize a certain use case at a more detailed level. It is a behavioral diagram that illustrates the flow of activities through a system.

- UML activity diagrams can also be used to depict a flow of events in a business process. They can be used to examine business processes in order to identify its flow and requirements.
- Activity diagrams can be used to model business requirements, create a high-level view of a system's functionalities, analyse use cases and for various other purposes. In each of these cases, here's how to draw an activity diagram from the beginning.
- Activity diagram describes the flow of control in a system. So, it consists of activities and links. The flow can be sequential, concurrent, or branched.
- Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.
- Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

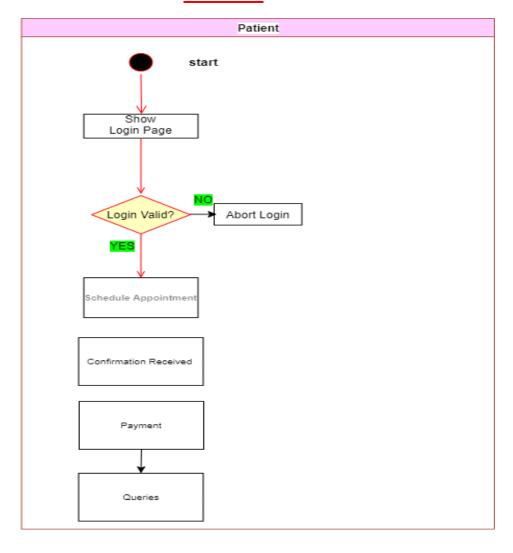
Following flow is shown in below Activity diagram:

- *Patients* Scheduling Appointments for check-up.
- *Receptionist* schedules patient's appointments and admission to the hospital. Receptionists also receive patient's payments.
- **Doctor** produces medication report and list of medicines are also available for each patient who is visiting hospital.
- *Admin* or the *main user* of the system could have access into the patient information in terms of availing the hospital services.

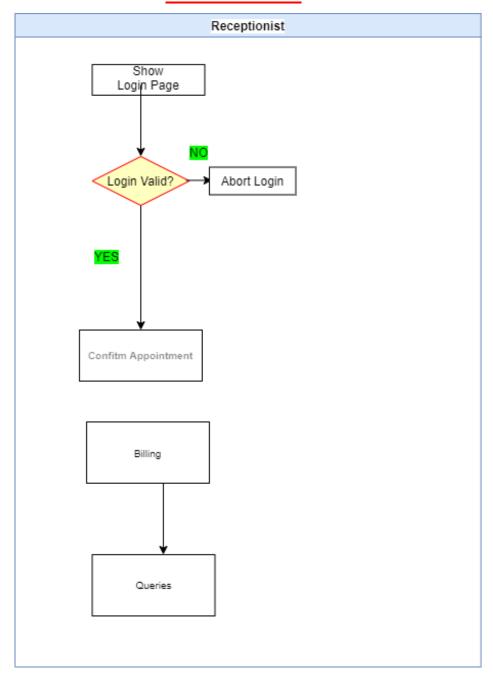
** Activity Diagram of EasyMed

– A hospital management System @ VITAP **

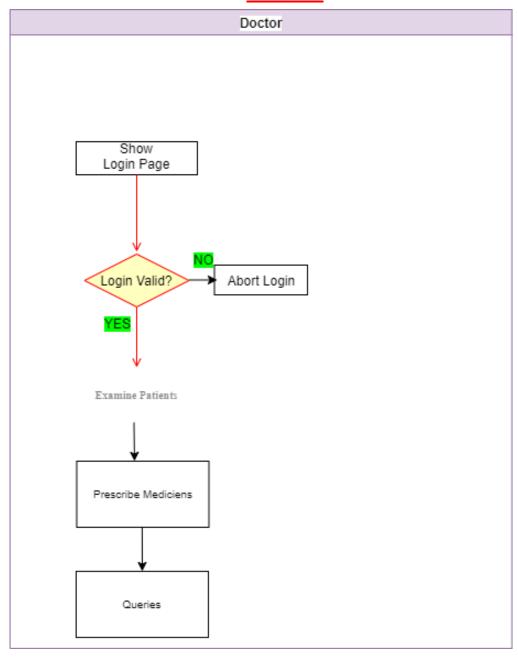
* PATIENT



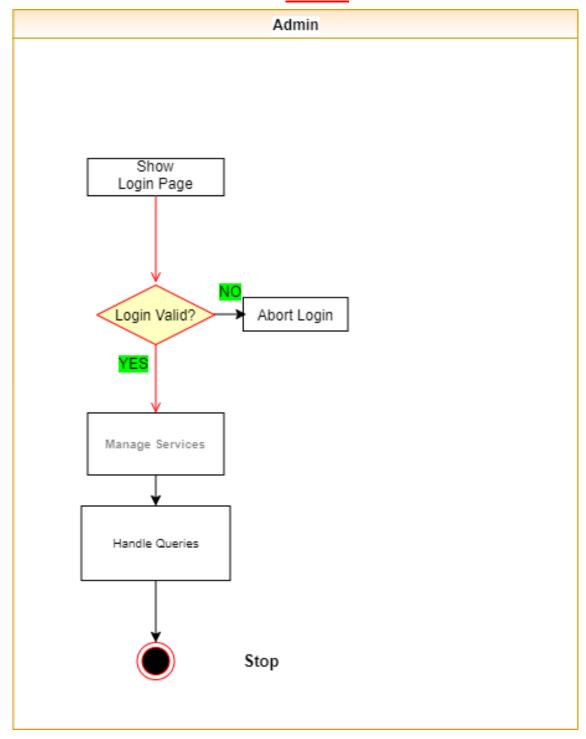
* RECEPTIONIST



❖ DOCTOR

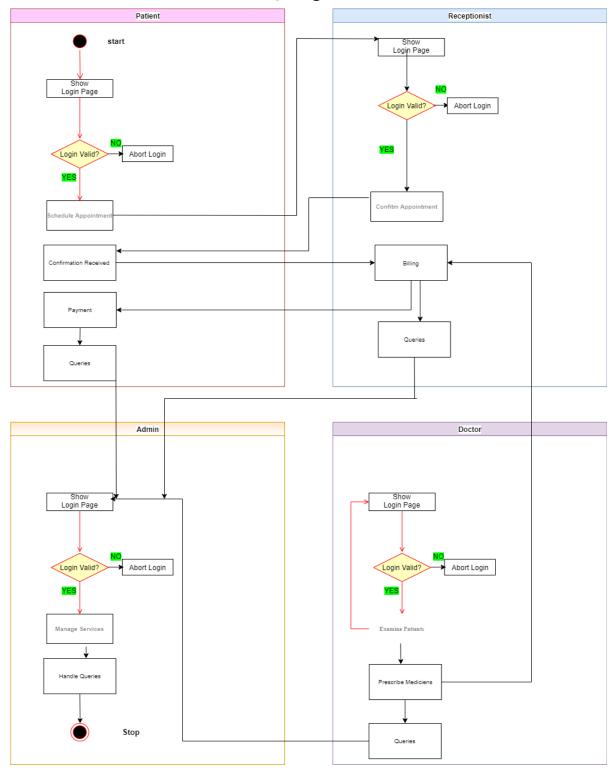


❖ <u>ADMIN</u>



EasyMed - Hospital

EasyMed @ VITAP



> 3.4 DFD (Data Flow Diagram):

- ♣ The *Data Flow Diagram* enables us to develop models of the information domain and functional domain.
- 4 As the DFD is refined into greater levels of detail, we perform an implicit functional decomposition of the system.
- ♣ At the same time, the DFD refinement results in a corresponding refinement of data as it moves through the processes that embody the application.

** The DFD diagram of EasyMed

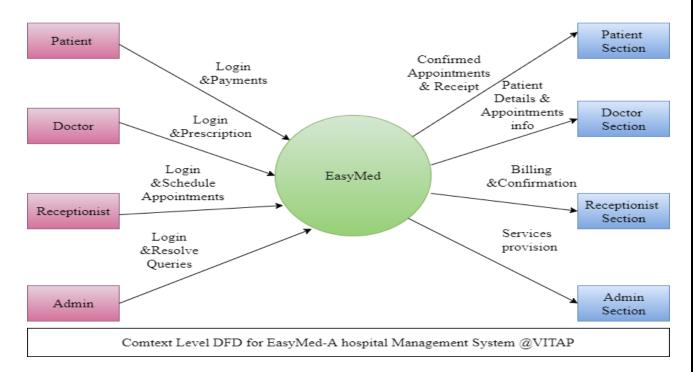
– A hospital management System @ VITAP **

Level-0 DFD (Context diagram) of EasyMed

 Level-0 DFD of Hospital Management System shows how the system is divided into subsystems (processes each of which deals with one or more of the data flows to or from an external agent and which together provide all of the functionality of the Hospital Management System as a whole.



EasyMed-A hospital management System

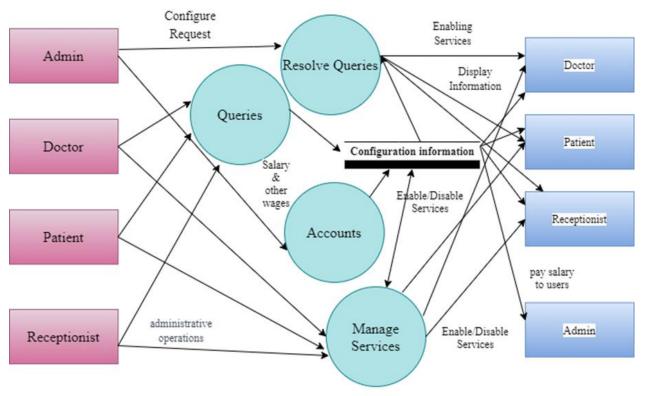


Level-1 Data Flow Diagram of EasyMed

- This level goes deeper into parts of first level.
- Here, we took admin functionalities and went deeper into exploration.



EasyMed-A hospital management System

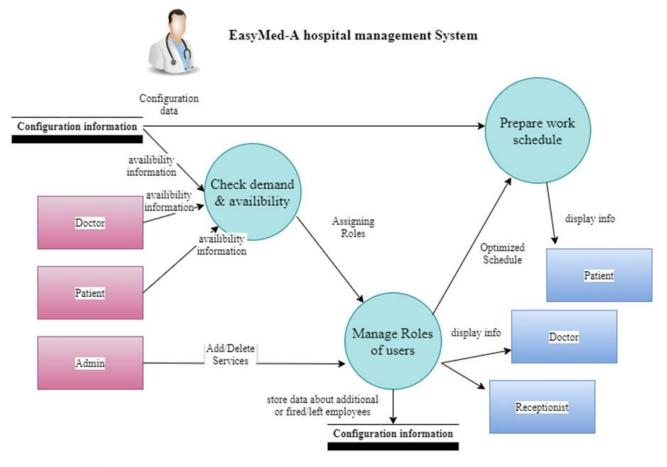




LEVEL 1 DFD for Admin in EasyMed-A hospital management System @ VITAP

Level-2 Data Flow Diagram of EasyMed

 Manage Services from admin functionalities is further drilled in this level.





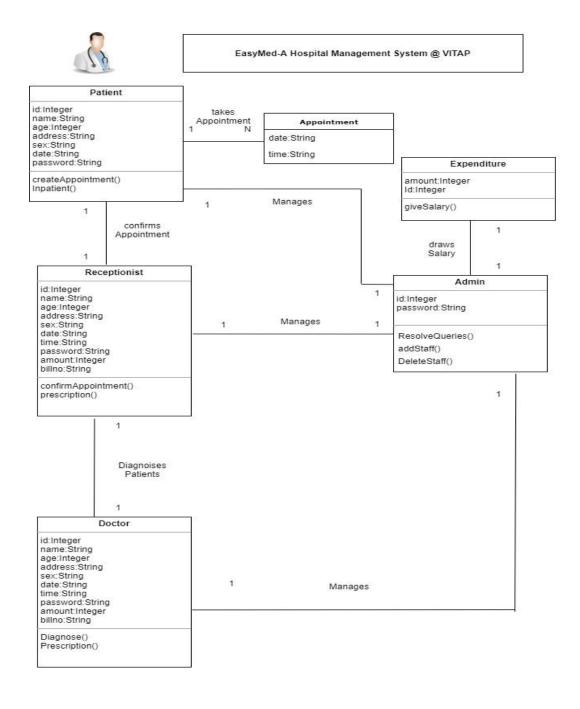
LEVEL 2 DFD for Admin(Manage Services) in EasyMed-A hospital management System @ VITAP

> 3.5 Class Diagram:

- Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.
- ♣ Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.
- ♣ Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

** Class diagram of EasyMed

- A hospital management System @ VITAP **



Perform Design and Development of Database Structure for your chosen project:

4. DEVELOPMENT

4.1 Database Structure

4. DEVELOPMENT:

> 4.1 Database Structure:

- ♣ The Database consists of 5 tables; patient, doctor, login, receptionist and admin
 - The 'patient' table consists of the basic data about the patient.
 - 'doctor' consists of the information about doctor.
 - The table '*admin*' consists of the information regarding the staff.
 - *Receptionist*' consists of data about the scheduling Appointments.
 - Finally, '*login*' table records of user.
 - In this way the entire billing system is integrated within a single database.



Login:

FIELDS	DATATYPE	SIZE	CONSTRAINTS	DESCRIPTION
User_id	Varchar2	20	Primary key	A unique id that
				identifies user
Password	Varchar2	10	Not null	Password to the user which enters into home

Patient:

FIELDS	DATATYPE	SIZE	CONSTRAINTS	DESCRIPTION
Patient_id	Number	2	Primary key	A unique id that identifies patient
Patient_name	Varchar2	20	Not null	Describes names of patient
Age	Integer	3	Not null	Descries age of patient
weight	Integer	3	Not null	Describes Weight of Patient
gender	Varchar2	20	Not null	Mentions gender of patient
Disease	Varchar2	20	Not null	Mention Disease
Disease_Description	Varchar2	100	Not null	Describe disease
Doctor_id	Integer	3	Not null	Describes Assigned doctor
Queries	Varchar2	100	Not null	Describe Queries



FIELDS	DATATYPE	SIZE	CONSTRAINTS	DESCRIPTION
Doctor_id	Number	20	Primary key	A unique id that identifies Doctor
Doctor_name	Varchar2	20	Not null	Describes names of doctors
Patient_id	Integer	2	Not null	Describe associated patient's id
Disease_history	Varchar2	100	Not null	Describe diseases history associated with patient
Queries	Varchar2	100	Not null	Describe Queries



FIELDS	DATATYPE	SIZE	CONSTRAINTS	DESCRIPTION
staff_id	Number	20	Primary key	A staff id that identifies all users
staff_name	Varchar2	20	Not null	Describes names of staff members
Queries_Resolve	Varchar2	100	Not null	Resolve Queries

Receptionist:

FIELDS	DATATYPE	SIZE	CONSTRAINTS	DESCRIPTION
patient_id	Number	2	Composite key	A patient id that identifies all patients
Doctor_id	Varchar2	2	Composite Key	A doctor id that identifies all doctors
Schedule	Varchar2	20	Not null	Fix date & time for doctor and patients.
Queries	Varchar2	100	Not null	Describe Queries

4 Perform Testing on the developed project to improve the quality:

5. SOFTWARE TESTING

- 5.1 **Test Plan**
- 5.2 **Test Results & Debugging**

5. SOFTWARE TESTING:

- ♣ The basic objective of writing test cases is to validate the testing coverage of the web application.
- ♣ So, writing test cases brings some sort of standardization and minimizes the ad-hoc approach in testing.

Fields in test cases:

- Test case id:
- **Unit to test:** What to be verified?
- Assumptions:
- **Test data:** Variables and their values
- Steps to be executed:
- Excepted result:
- **Actual result:** Pass/Fail
- Comments:

> 5.1 Test Plan:

↓ <u>CASE-1</u> :-

Test case name: login

Test case id: 120

Unit to test: login

Assumption: Login process is successfully done

Test data: User id ={valid,invalid,empty,valid email,Invalid email,}

Password: {valid,invalid,empty}

♣ CASE-2 :-

Test case name: patient

Test case id: 121

Unit to test: patient

Assumption: Patient has registered successfully

Test data: User id = {valid,invalid,empty,valid email,Invalid email,}

Password:{valid,invalid,empty}

♣ <u>CASE-3</u> :-

Test case name: doctor

Test case id: 122

Unit to test: doctor

Assumption: Doctor checks the patient successfully

Test data: User id ={valid,invalid,empty,valid email,Invalid email,}

Password: {valid,invalid,empty}

♣ <u>CASE-4</u> :-

Test case name: admin

Test case id: 123

Unit to test: admin

Assumption: Admin resolves queries successfully

Test data: User id = {valid,invalid,empty,valid email,Invalid email,}

Password:{valid,invalid,empty}

♣ CASE-5 :-

Test case name: receptionist

Test case id: 124

Unit to test: receptionist

Assumption: Receptionist successfully fixes date & time for doctor

and patients.

Test data: User id = {valid,invalid,empty,valid email,Invalid email,}

Password: {valid,invalid,empty}

STEPS TO BE EXECUTED :-

- Login page to be displayed
- Category of user in chosen
- User id is given in the test box
- Password is given in the text box
- Login button is viewed
- Click the Login button
- Verify corresponding user page with welcome note

LEAD VALUE :-

- View login page
- Display corresponding welcome page
- View the profile with correct username

ACTUAL VALUE :-

- Login page is viewed
- Displayed welcome page
- Profile is viewed with correct username
- **PASS / FAIL :-** Pass
- **COMMENT:** There is no need for software development team to change

> 5.1 Test Results & Debugging:

Serial no	Test case id	Test case name	Pass/Fail	If fail, Comment on it
1.	120	Login	Pass	
2.	121	Patient	Pass	
3.	122	Doctor	Pass	
4.	123	Admin	Fail	Instead of button label were placed. So, result button not working properly.
5.	124	Receptionist	Fail	Instead of button label were placed. So, result button not working properly.

Perform a demonstration of your chosen and developed project:

6. PROJECT DEMO

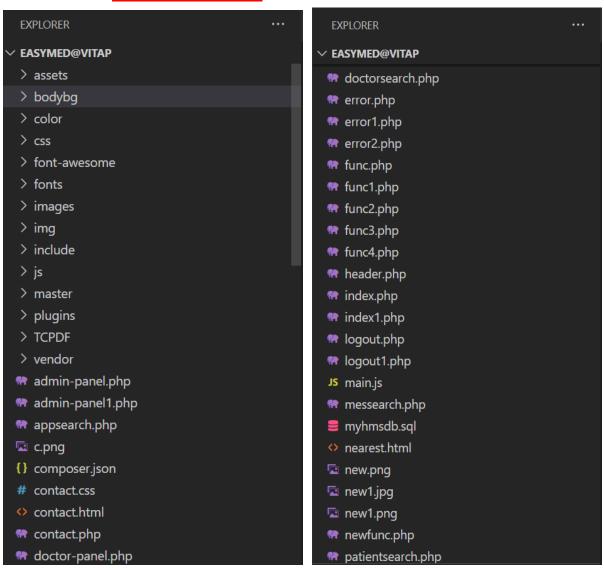
6.1 Screen shots

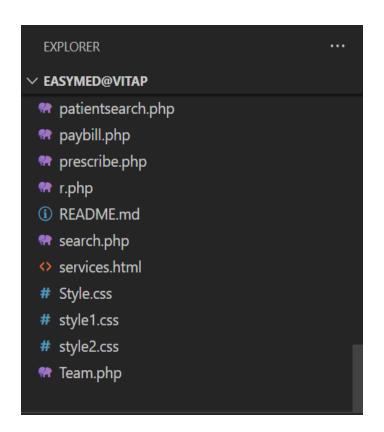
6. PROJECT DEMO:

> 6.1 Screen shots:

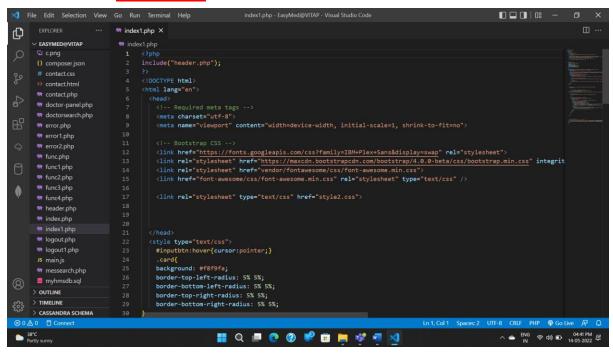
Screen shots of CODE:-

▲ EasyMed@VITAP :-





Login Page :-



4 Patient :-

```
m patientsearch.php X
Ф

∨ EASYMED@VITAP

♥ logout1.php

                                                                                                                   o nearest.html
                                                                                                                   include("newfunc.php");
if(isset($_POST['patient_search_submit']))
                                                                                                                         $contact=$_POST['patient_contact'];
$query = "select * from patreg where
                                                                                                                        $\footnote{\text{Squery}} = "select * from patreg where contact= '$\footnote{\text{Sontact'}};
$\footnote{\text{Squery}} = "select * from patreg where contact= '$\footnote{\text{Scontact'}};
$\footnote{\text{Srow}} = \text{Srow} = \tex
                     m patientsearch.php
                       m paybill.php
  •

 README.md

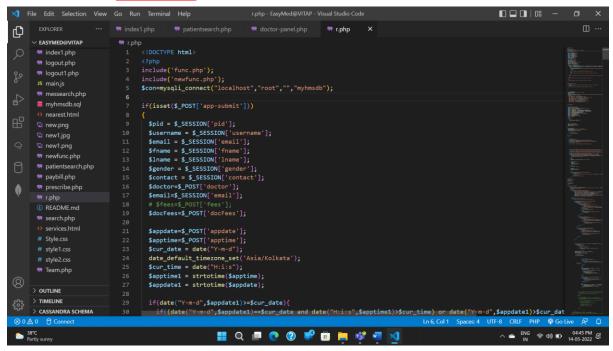
                      search.php
                                                                                                                         else {
| echo "<div class='container-fluid' style='margin-top:50px;'>
                      # Style.css
                                                                                                                   # style2.css
                 > OUTLINE
                                                                                                                                First Name
Last Name
Email

> TIMELINE
                 > CASSANDRA SCHEMA
                                                                                                                                                                                                                                                                                                                                                                                        Ln 1, Col 1 Spaces: 2 UTF-8 LF PHP 📦 Go Live 🔊 🚨
                                                                                                                                                                                       🔡 Q 🔎 🧑 🕐 🙃 ⊨ 💕 🚾 🔀
```

Doctor :-

```
File Edit Selection View Go Run Terminal Help
                                                    doctor-panel.php - EasyMed@VITAP - Visual Studio Code
                                                                                                                  中
                                                       ** doctor-panel.php X
    ∨ EASYME... [‡ 日 ひ 🗗 🐡 doctor-panel.php
                              include('func1.php');
$con=mysqli_connect("localhost","root","","myhmsdb");
$doctor = $_SESSION['dname'];
                              if(isset($_GET['cancel']))
     m doctor-panel.php
                                 $query=mysqli_query($con,"update appointmenttb set doctorStatus='0' where ID = '".$_GET['ID']."'");
if($query)
     m doctorsearch.php
                                   echo "<script>alert('Your appointment successfully cancelled');</script>";
     error2.php
     func.php
     func3.php
0
     m func4.php
     m index1.php
     n logout.php
     myhmsdb.sql
    > OUTLINE
हतु > TIMELINE
                                   28°C
Partly sunn
                                                🔡 Q 🔎 🥙 🕏 🗊 📜 💖 🌉 💌
```

Reception:



♣ Admin :-

```
File Edit Selection View Go Run Terminal Help
                                                                                                     admin-panel.php - EasyMed@VITAP - Visual Studio Code
                                                                                                                                                                                                                          ф
                                                                                                                                                                     * admin-panel.php ×
                                               m admin-panel.php
1 <!DOCTYPE html>

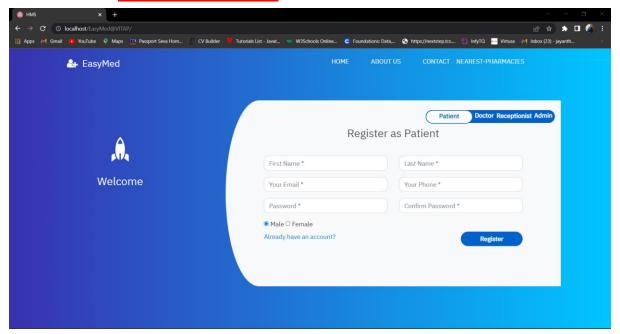
∨ EASYMED@VITAP

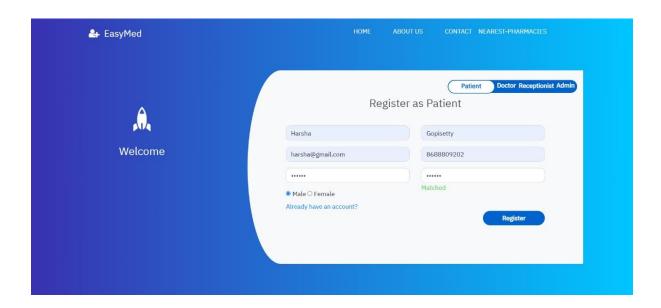
                                                       { Doc The
{
    c?php
include('func.php');
include('newfunc.php');

$con=mysqli_connect("localhost","root","","myhmsdb");
}
           > bodyba
                                                           $pid = $_SESSION['pid'];
$username = $_SESSION['username'];
$email = $_SESSION['email'];
$fname = $_SESSION['fname'];
           > images
                                                           $gender = $_SESSION['gender'];
$lname = $_SESSION['lname'];
$contact = $_SESSION['contact'];
           > master
           > plugins
 •
                                                         if(isset($ POST['app-submit']))
           admin-panel1.php
                                                            $pid = $_SESSION['pid'];
                                                           $pid = $_SESSION['pid'];
Susenname = $_SESSION['username']
$email = $_SESSION['email'];
$fname = $_SESSION['email'];
$lname = $_SESSION['aname'];
$gender = $_SESSION['gender'];
$contact = $_SESSION['contact'];
$doctor=$_POST['doctor'];
$email=$_SESSION['email'];
STATE > TIMELINE
                                                            $docFees=$_POST['docFees'];
        > CASSANDRA SCHEMA
                                                                                                                                                                                      Ln 16, Col 1 Spaces: 2 UTF-8 CRLF PHP @ Go Live 저 요
                                                                                           🔡 Q 🔎 🤨 📝 🕫 ⊨ 💖 🚾 🔀
28°C
Partly sunny
```

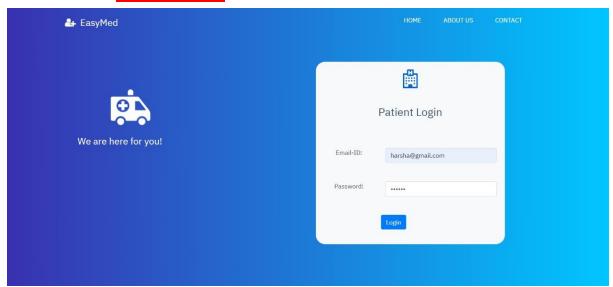
Screen shots of WEBSITE:-

Patient Registration :-





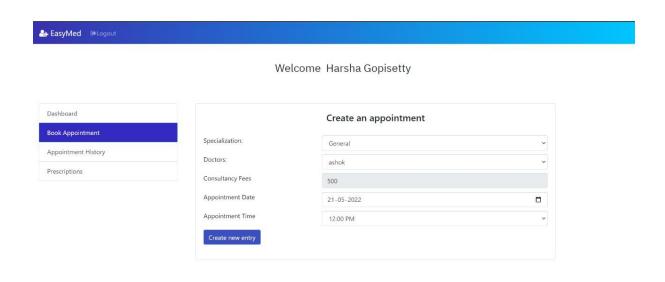
4 Patient Login :-

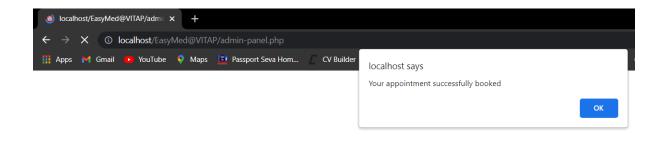




View Prescription List

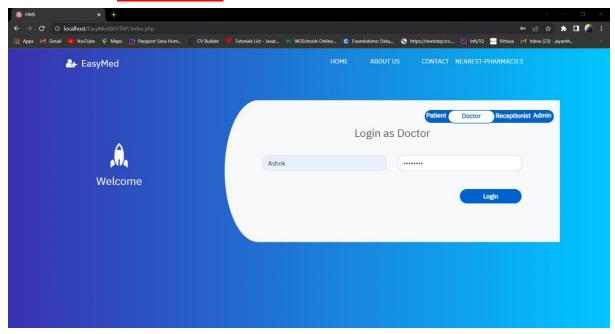


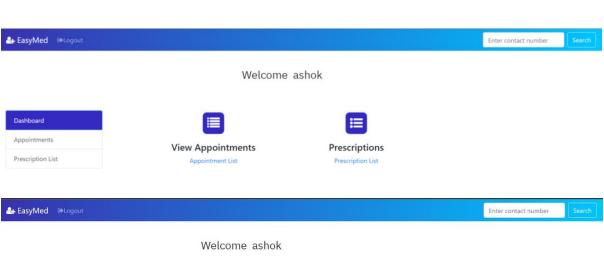




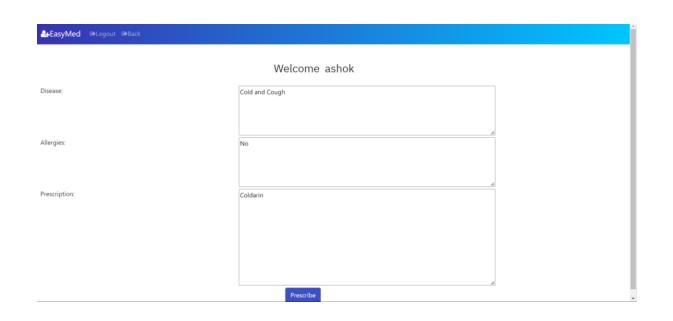


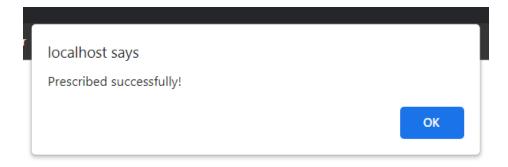
♣ Doctor Login :-









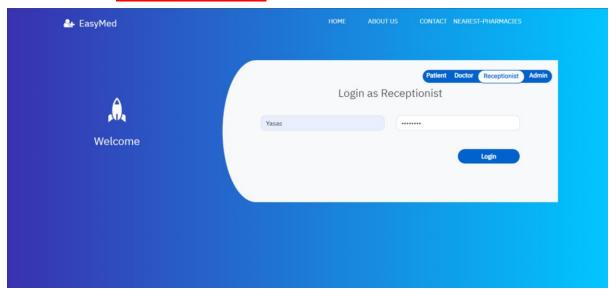




Welcome ashok

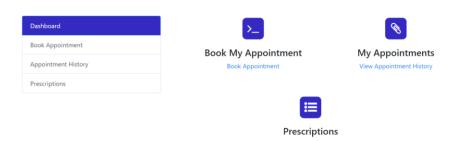


Receptionist Login:



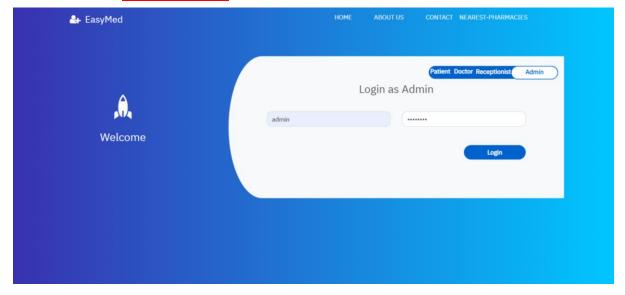


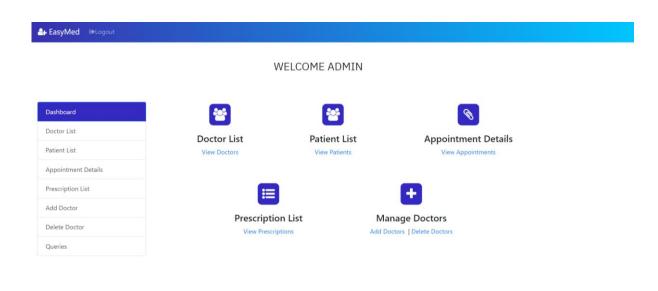
Welcome RECEPTION

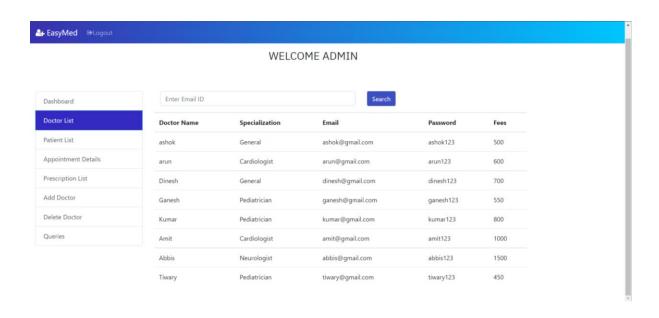


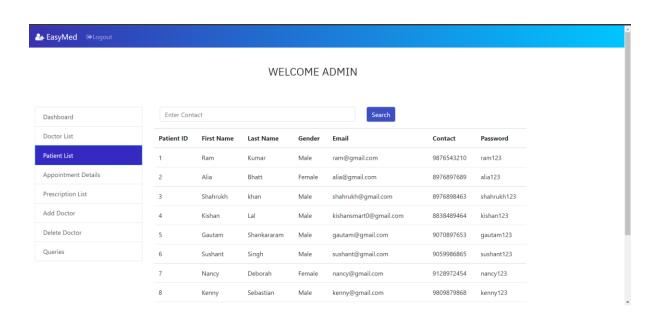


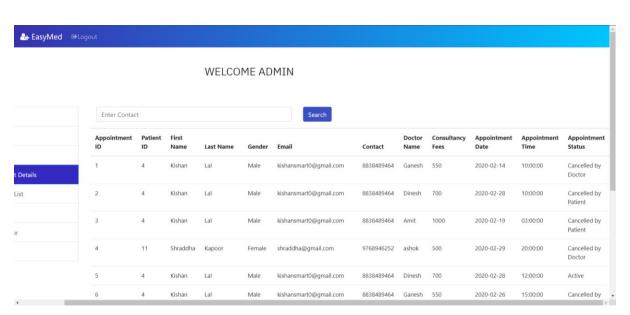
★ Admin Login :-

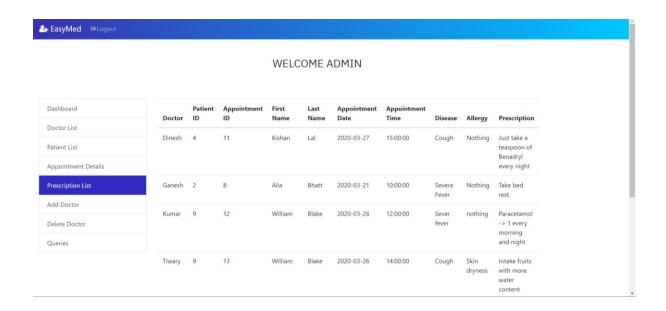






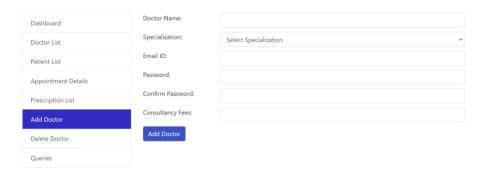


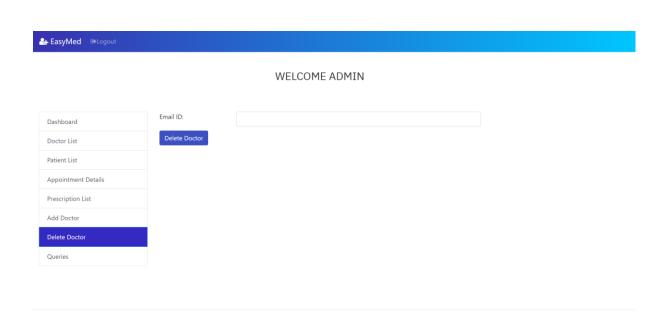


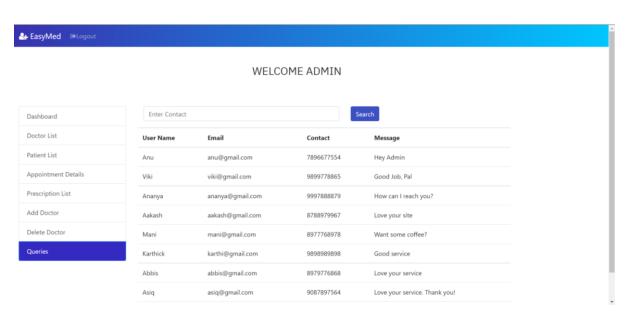




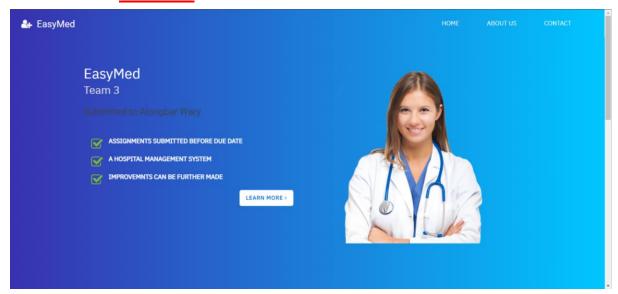
WELCOME ADMIN



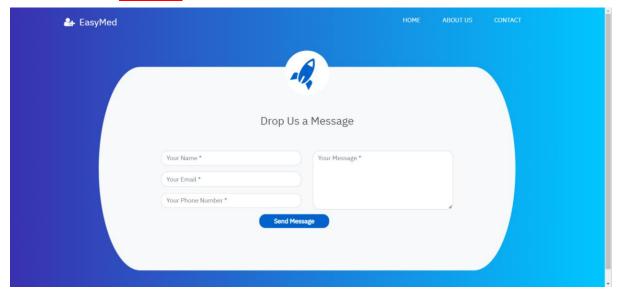




♣ About Us :-

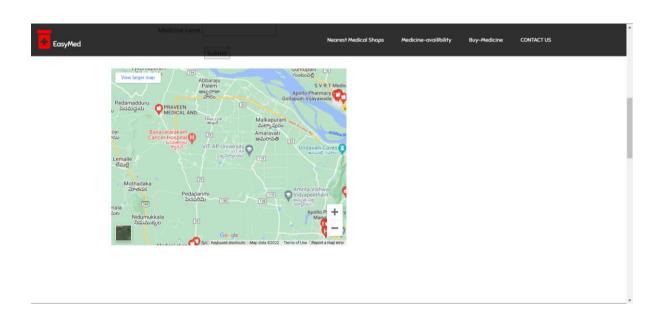


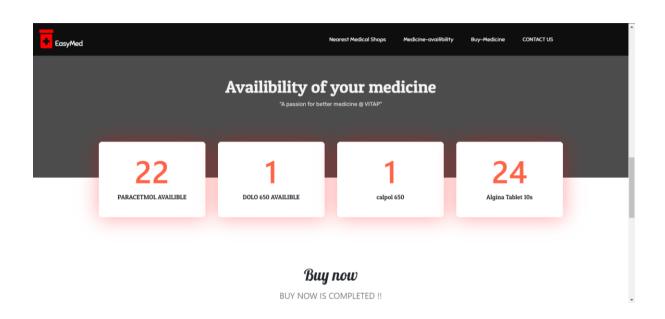
♣ Contact :-

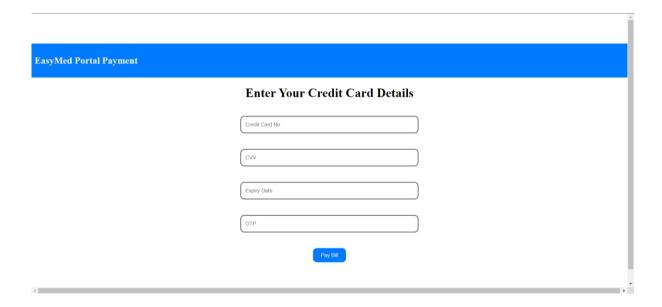


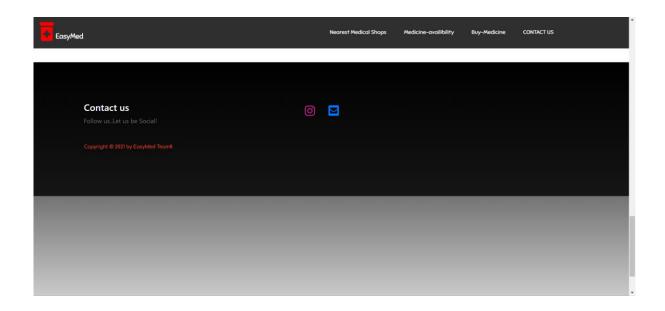
Nearest-Pharmacies:



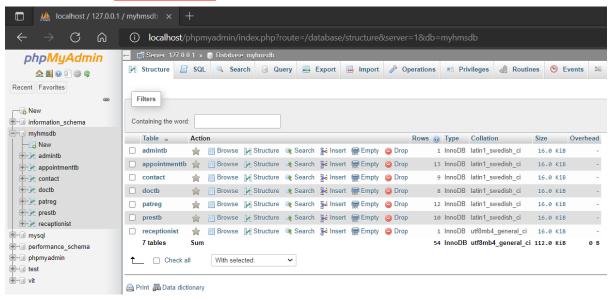








♣ DATABASE :-



**** The End ****