Vellore Institute of Technology
School of Computer Science and Engineering

Software Engineering

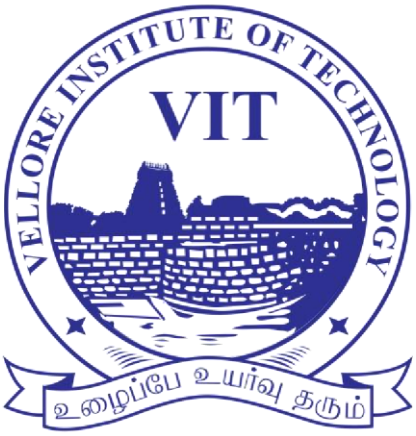
Course Code: BCSE301L

Class Number: VL2024250503163

Slot: L57 – L58

PROJECT ON

HOSPITAL MANAGEMENT SYSTEM



-BY

SHREYA GUPTA

ACKNOWLEDGEMENT

I would like to express my deep appreciation and heartfelt gratitude to my supervisor, Prof.Yoganand S., from VIT University. Without his unwavering motivation and continuous encouragement, this research would not have been completed successfully.

I am sincerely grateful to the Chancellor of VIT University, Dr. G Viswanathan, the Vice Presidents, and the Vice Chancellor for their inspiring guidance and support. Their encouragement and provision of essential infrastructure and resources were instrumental in the progress of my research.

I would also like to extend my sincere thanks to Dr. N. Jaisankar, the Dean of the School of Computer Science and Engineering (SCOPE) at VIT University, for his kind words of support and encouragement throughout my research journey. Additionally, I am grateful to my 'HOD's' and classmates for their support in various ways throughout my research work.

Furthermore, I wish to express my profound gratitude to my parents for their enduring sacrifices during my research. Their unwavering support and encouragement have been invaluable to me.

I am truly grateful to all those mentioned above for their contributions to my research, and I feel incredibly fortunate to have had their guidance and support throughout this endeavour.

Place – Vellore

Date- 27.03.2025

Team Members

1. Shreya Gupta

Signature of Supervisor

Table of Contents:

S.NO	CONTENTS	PAGE NUMBER
1.	INTRODUCTION	3
2.	ABSTRACT	4
3.	SRS	5
4.	DESIGN	12
5.	IMPLEMENTATION (CODE + SCREEN SHOTS)	16
6.	TESTING	23
7.	CONCLUSION	29
8.	REFERENCES	30

INTRODUCTION:

A **Hospital Management System (HMS)** is designed to simplify and digitize various hospital operations, ensuring better organization and efficiency. Many hospitals still rely on manual processes for handling patient records, scheduling appointments, and managing staff, which can lead to misplaced data, scheduling conflicts, and administrative delays. A digital HMS helps address these issues by providing a centralized platform to manage hospital activities more effectively.

This system includes features such as **appointment scheduling, patient record management, staff and doctor information handling, and a contact portal** for seamless communication. Additionally, a **Skin Disease Predictor** is integrated into the system, allowing users to get a preliminary assessment of skin conditions before visiting a specialist.

By transitioning from manual to digital processes, the HMS improves hospital workflow, enhances patient experience, and reduces paperwork. It ensures that information is easily accessible, minimizes errors, and helps medical professionals focus more on patient care rather than administrative tasks.

ABSTRACT:

Hospital management involves multiple tasks, including patient record maintenance, appointment scheduling, staff coordination, and communication between doctors and patients. Traditionally, these processes are handled manually, which can lead to inefficiencies such as misplaced records, appointment delays, and administrative burdens. To address these challenges, this **Hospital Management System (HMS**) provides a **centralized digital platform** that automates and streamlines hospital operations.

The system includes essential features such as **patient registration, appointment booking, medical record storage, staff and doctor management, and a contact portal** for seamless communication. Additionally, a **Skin Disease Predictor** is integrated to assist patients in obtaining a preliminary assessment of skin conditions before consulting a doctor. This predictive model utilizes an image-based dataset to analyze symptoms and provide an initial diagnosis, offering users a convenient way to check for potential concerns.

By digitizing hospital processes, the HMS enhances accessibility, reduces errors, and improves overall efficiency. The system ensures that patient data is securely stored and easily retrievable, minimizing the risk of information loss. Furthermore, automated appointment scheduling reduces wait times and enhances the patient experience**. The HMS serves as a comprehensive solution for modern healthcare facilities**, aiming to improve operational workflow and deliver better healthcare services.

SRS (as per the earlier format circulated)

1. INTRODUCTION:

1.1 Purpose-

The purpose of this project is to develop a web-based Hospital Management System (HMS) to streamline hospital operations, enhance patient management, and improve appointment scheduling. The system enables hospitals to digitally store patient records, manage doctor and staff information, schedule appointments, and facilitate communication between patients and healthcare providers. Additionally, it includes a Skin Disease Predictor, allowing users to get an initial assessment of skin conditions before consulting a specialist.

1.2 Intended audience-

End Users: Patients, doctors, hospital staff, and administrators who require an efficient system for managing appointments, records, and communication.

Stakeholders: Instructors evaluating the project, healthcare professionals providing feedback, and users interacting with the system for medical services.

1.3 Project Scope-

This project aims to provide essential hospital management features, focusing on:

- **Patient registration and medical record management** to securely store and retrieve patient data.
- **Appointment scheduling system** for easy booking and management of doctor visits.
- **Doctor and staff management** to organize hospital workforce information.
- **Contact and communication portal** to enable seamless interaction between patients and hospital staff.
- **Skin Disease Predictor** for preliminary diagnosis based on image analysis.
- **User authentication and security features** to ensure data privacy.
- **A web-based application** accessible via any modern browser for ease of use and accessibility.

2. OVERALL DESCRIPTION:

2.1 Product perspective-

This web-based **Hospital Management System (HMS)** will be developed using the following technologies:

- **Frontend**: HTML, CSS, and JavaScript for creating a responsive and user-friendly interface.
- **Backend**: MySQL for data storage, user authentication, and real-time database management.

2.2 User interface-

Login: Allows users to authenticate themselves by entering their credentials.

- **Fields**: Email, Password
- **Features**:
  - "Forgot Password" link for resetting credentials
  - Basic validation for required fields and valid email format
  - Error message display for incorrect login attempts

Register: Enables new users (patients) to create an account.

- **Fields**: Name, Email, Password, Confirm Password
- **Features**:
  - Validation for required fields and password strength
  - Email verification for secure account creation

Main Page: Serves as the homepage of the system.

- **Features**:
  - Overview of hospital services
  - Navigation to other sections

Appointment Page: Allows patients to book an appointment.

- **Fields**: Name, Email, Doctor Selection (dropdown), Appointment Date & Time
- **Features**:
  - Displays available slots
  - Confirmation message upon successful booking

Contact Us Page: Allows users to reach out to the hospital for inquiries.

- **Fields**: Name, Email, Message
- **Features**:
  - Sends inquiries to hospital administration via email
  - Displays hospital contact details

About Us Page: Provides information about the hospital.

- **Features**:
  - Details about hospital history, mission, and services
  - Team or staff information

2.3 System interface

The Hospital Management System (HMS) will use MySQL as the database for storing and managing user data, appointments, and other essential records.

2.4 Constraints, assumptions and dependencies-

Constraints

- Limited to desktop browsers in this version; mobile responsiveness may be limited.
- Relies on MySQL database for storage, requiring a server environment to operate.
- Appointment availability depends on predefined doctor schedules.

Assumptions

- Users will enter accurate details when booking appointments or submitting queries.
- A stable internet connection is available for users to access the web-based system.
- Admins will regularly update doctor availability and appointment statuses.

Dependencies

- MySQL for database storage and management.
- PHP (or any backend framework you're using) for server-side operations and database interaction.
- HTML, CSS, JavaScript for frontend development.

2.5 User characteristics-

Primary Users: Patients, hospital staff, and administrators.
Patients: Individuals of all ages booking appointments, making inquiries, or accessing basic hospital services.
Administrators: Hospital management overseeing system operations and maintaining records.
Technical Skills: Users have basic computer literacy, including navigating websites and filling out forms.
Access Mode: The system is primarily accessed via **desktop or laptop browsers**, with potential for future mobile compatibility.

3. FUNCTIONAL REQUIREMENTS:

3.1 Actors-

Primary Actor:

Patient:

- Books appointments with doctors.
- Views their appointment history.
- Contacts the hospital for inquiries.

Supporting Actor:

System

- Handles authentication and security.
- Stores and retrieves patient and appointment data.
- Sends notifications for appointment confirmations or reminders.

3.2 Actions-

Book Appointment:

- Patients select a doctor and available time slot.
- System verifies availability and confirms the booking.

View Appointments:

- Patients view their upcoming and past appointments.
- Doctors check their scheduled appointments for the day.

Manage Patient Records:

- Doctors update patient medical history and prescriptions.
- Admins maintain records for hospital management.

Cancel or Reschedule Appointment:

- Patients can request cancellations or changes.
- System notifies doctors and updates availability.

Send Notifications:

- System sends appointment reminders to patients via email/SMS.
- Alerts for appointment confirmations, cancellations, or follow-ups.

Login & Authentication:

- Patients, doctors, and admins log in using secure credentials.
- System ensures role-based access to relevant data.

3.3 Object-

Patient Data:

- Name: Full name of the patient.
- Age: Patient's age (numeric).
- Medical History: Past and current medical conditions.
- Contact Information: Phone number and email for communication.

Appointment Data:

- Patient Name: Name of the person booking the appointment.
- Doctor Assigned: Name of the doctor assigned.
- Date & Time: When the appointment is scheduled.

3.4 Qualifier-

Validation Rules:

Patient name must only contain alphabetic characters.
Age must be a numeric value and greater than 0.
Appointment date must be in **YYYY-MM-DD** format.
Contact information must be a valid phone number or email.
Doctors can only be assigned appointments during their availability.

Error Handling:

If incorrect login credentials are entered, display **"Invalid Username or Password."**
If an appointment slot is unavailable, show **"Selected time is already booked. Please choose another slot."**
Ensure that all required fields are filled; otherwise, display **"Please complete all fields before submitting."**

4. USE CASES:

Use Case 1: Login

Actor: User (Patient, Doctor, or Admin)
Description: Allows users to securely access their accounts.
Steps:

- User enters email and password.
- System validates credentials.
- On success, user is redirected to the appropriate dashboard (Patient, Doctor, or Admin).
- On failure, an error message is displayed.

Use Case 2: Book Appointment

Actor: Patient
Description: Enables patients to schedule an appointment with a doctor.
Steps:

- User selects the "Book Appointment" option.
- User selects a doctor, date, and preferred time slot.
- System validates availability.
- Appointment is saved in the database, and confirmation is shown.

Use Case 3: View appointments

Actor: Patient, Doctor, or Admin
Description: Allows users to view scheduled appointments.
Steps:

- User selects the "View Appointments" option.
- System retrieves scheduled appointments.
- Data is displayed in a tabular format (sortable by date, doctor, or patient).

Use Case 4: Manage Patients

Actor: Admin / Doctor
Description: Allows doctors and admins to manage patient records.
Steps:

- User selects "Manage Patients."
- System displays a list of registered patients.
- Admin/Doctor can add, edit, or remove patient records.

Use Case 5: Contact Hospital

Actor: User (Patient or Visitor)
Description: Enables users to send inquiries via the contact form.
Steps:

- User enters name, email, subject, and message in the "Contact Us" form.
- System validates input and stores the inquiry in the database.
- Admin receives the inquiry and responds.

Use Case 6: Register New User

Actor: Patient
Description: Allows new users to create an account.
Steps:

- User selects "Register."
- User enters required details (name, email, password, phone number).
- System validates input and stores the data.
- Registration confirmation is displayed.

5. NON-FUNCTIONAL REQUIREMENTS:

5.1 Safety-

HTTPS is used for secure data transmission.
Encryption of sensitive data, such as passwords, to prevent unauthorized access.

5.2 Portability-

Compatible with **Chrome, Firefox, and Edge** for smooth user experience.
No support for mobile devices in the current version.

5.3 Security-

User authentication via email and password.
I**nput validation** to prevent SQL injection and other security threats.
Access control to ensure only authorized users can manage hospital records.

5.4 Reliability-

The system is designed to handle up to **500 daily active users** without performance issues.
Regular database backups to prevent data loss.

5.5 Compatibility-

Developed for **desktop browsers**, with no dedicated mobile support in this version.
Works across **Windows, macOS, and Linux** with a compatible browser.

5.6 Scalability-

The system architecture allows for future enhancements, such as:

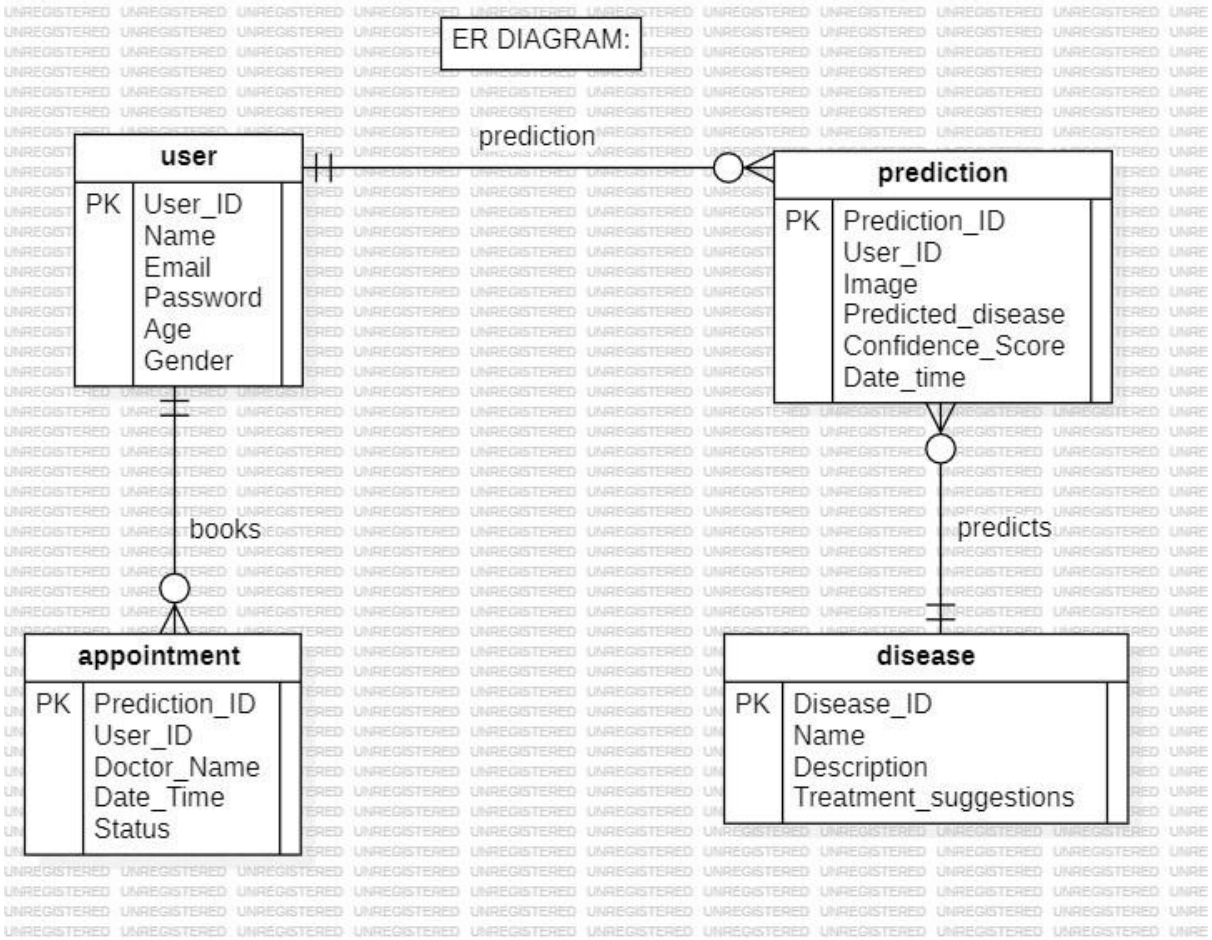
- **Adding more departments** (e.g., pharmacy, billing).
- **Mobile version development** for broader accessibility.
- **Advanced analytics** for hospital performance tracking.

Design

a. **Low level**
b. **High Level**
c. **ER and UI/UX**

1. ER diagram

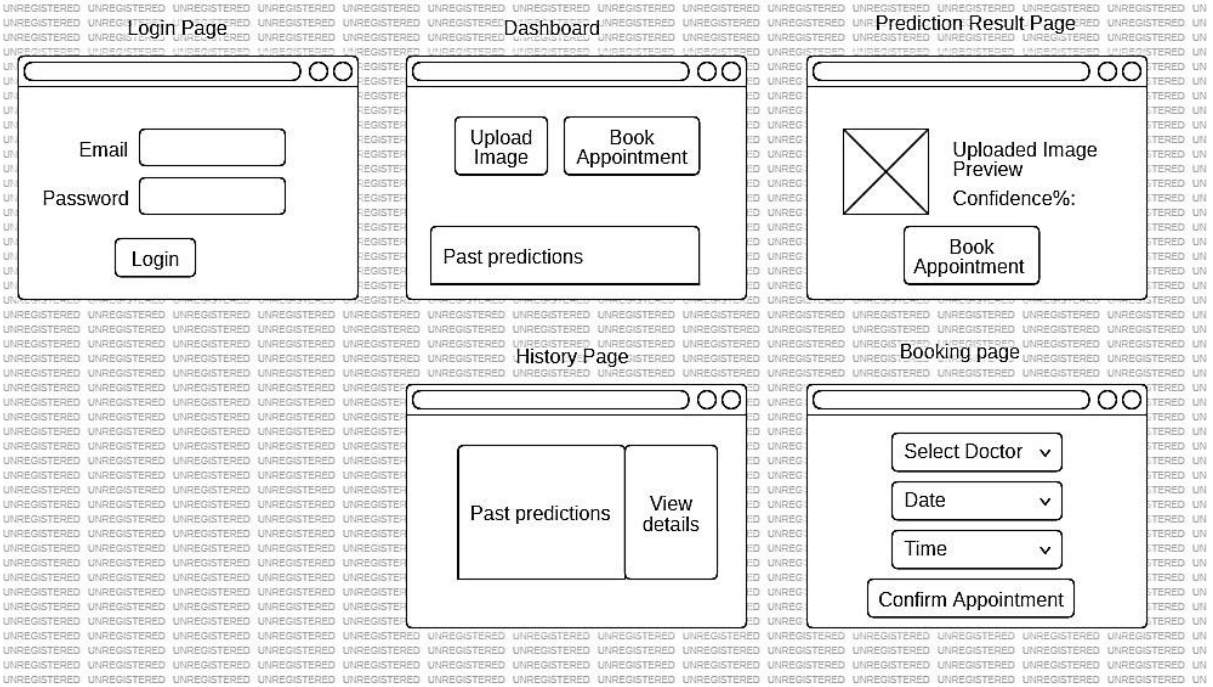
The ER diagram for the Skin Disease Predictor project represents the relationship between users, predictions, diseases, and appointments. The User entity stores details such as User_ID, Name, and Email. Users upload skin images for analysis, and the Prediction entity stores the results, linking each prediction to a specific disease from the Disease entity. If necessary, users can book an appointment, which is stored in the Appointment entity. The diagram illustrates the relationships: one user can have multiple predictions and appointments, while multiple predictions can be linked to the same disease.



2. UI diagram:

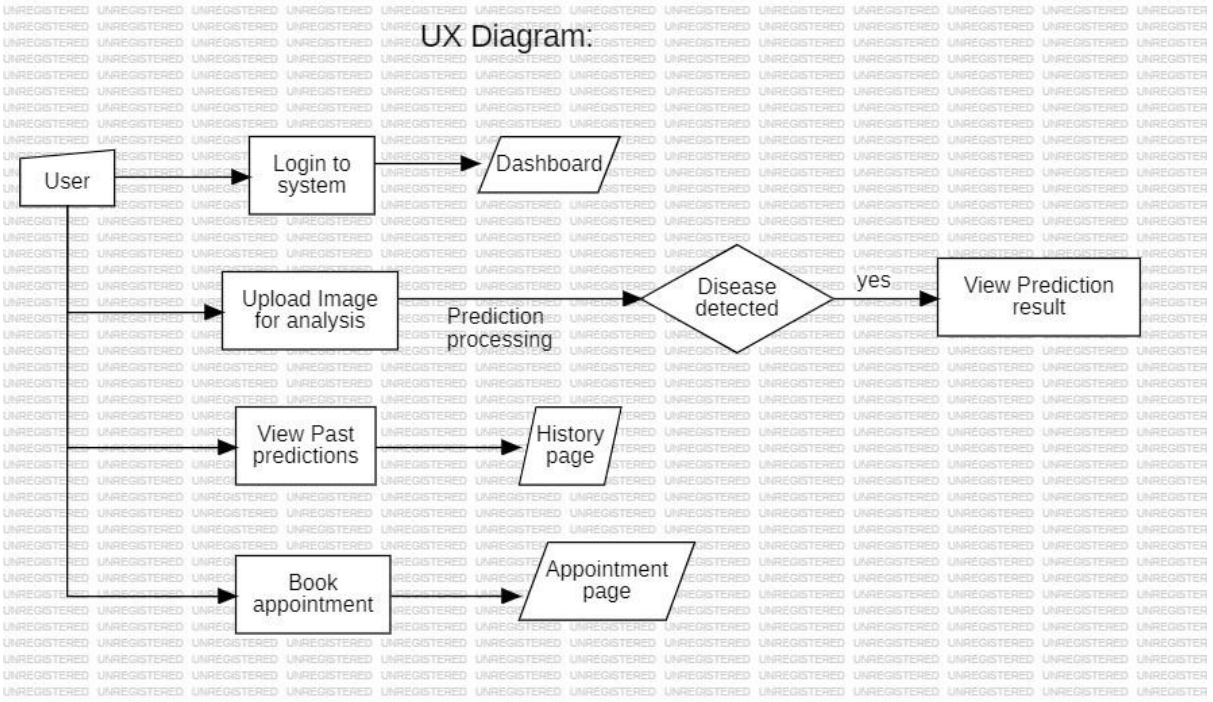
The UI Diagram (Wireframe) for the Skin Disease Predictor represents the layout and structure of different screens within the system. It visually organizes essential pages like the Login Page, Dashboard, Prediction Page, Appointment Booking Page, and History Page. Each frame contains user interface elements such as buttons, input fields, labels, and images to indicate functionality. Navigation between screens is implied through buttons and links, ensuring a seamless user experience. This wireframe serves as a blueprint for designing the system's user interface, ensuring clarity and usability.

UI Diagram:

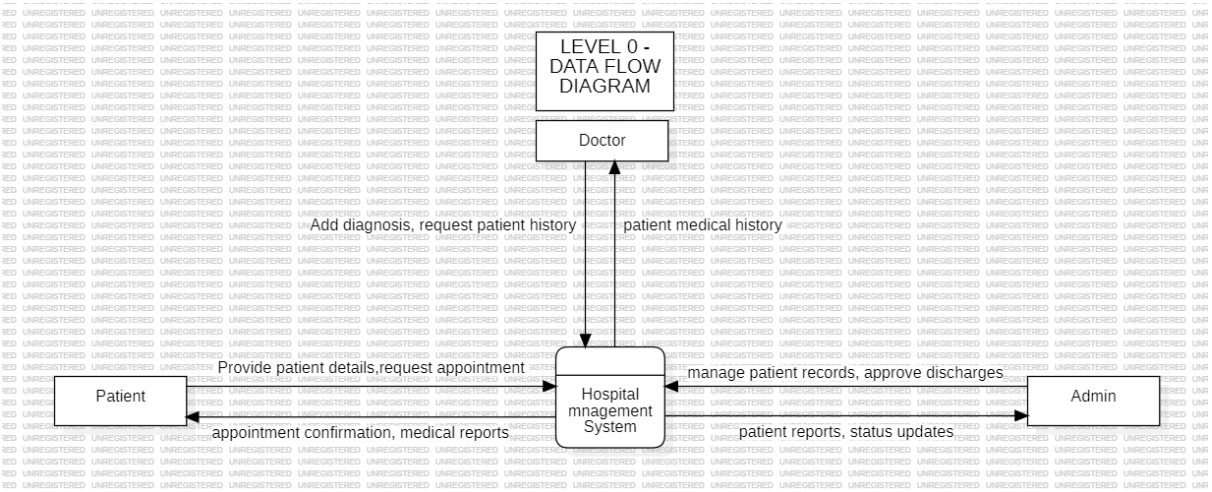


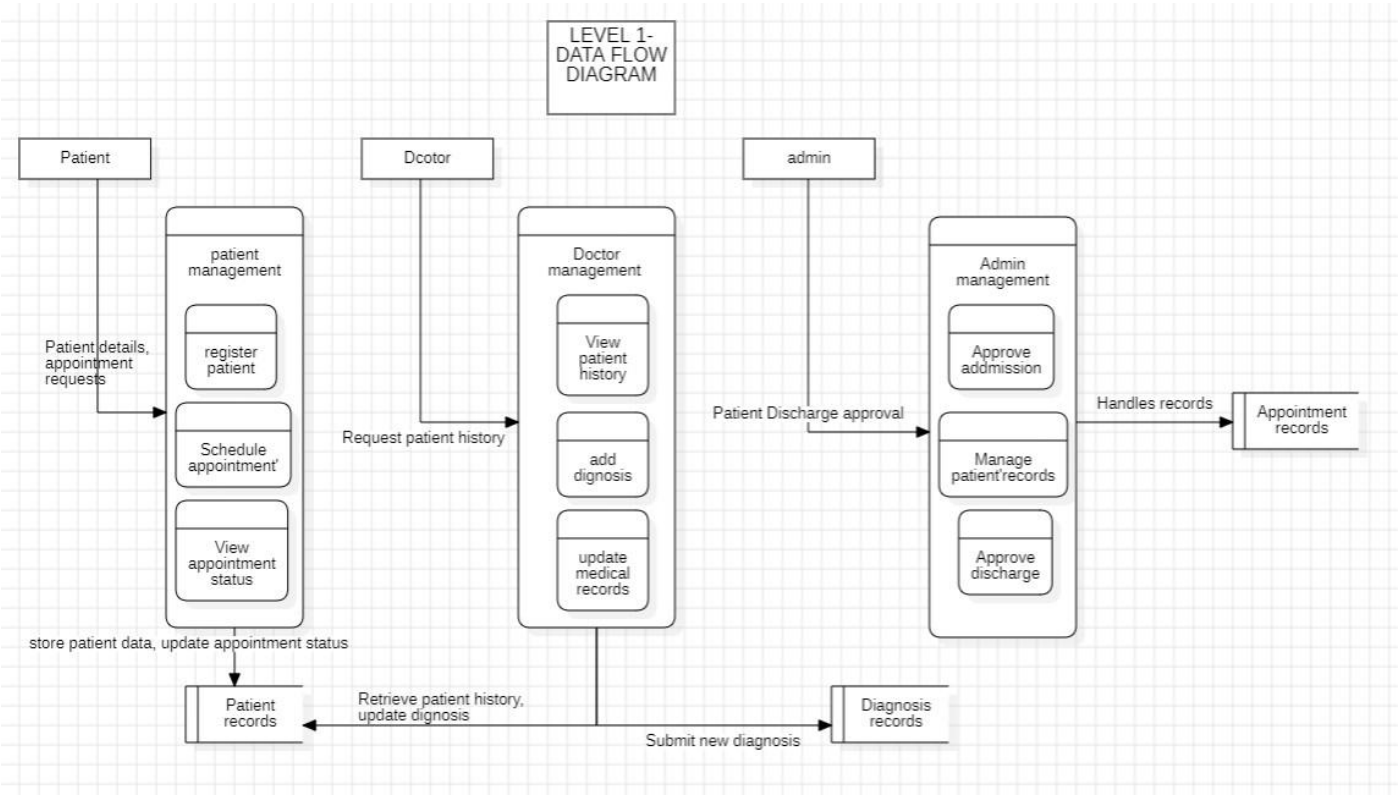
3. UX diagram:

The UX Diagram (User Flow) for the Skin Disease Predictor outlines the step-by-step interactions a user has with the system. It begins with the user logging in and navigating through key actions like uploading an image for analysis, receiving a prediction, booking an appointment, or viewing past results. The diagram includes decision points, such as whether the image analysis provides a clear result or requires re-uploading. This flowchart ensures a smooth and intuitive user experience by mapping out every possible interaction within the system.



4. Context based- DFD (level 0(HIGH LEVEL) and level 1(LOW LEVEL))





ALGORITHM – CODE:

Code: https://github.com/shreya320/HMS

Algorithm for Hospital Management System (HMS) with Skin Disease Predictor:

1. **Start**

2. **User Authentication:**
   - Prompt the user to **log in**.
   - Verify login credentials from the **MySQL database**.
   - If valid, proceed to the respective dashboard; otherwise, display an error message.

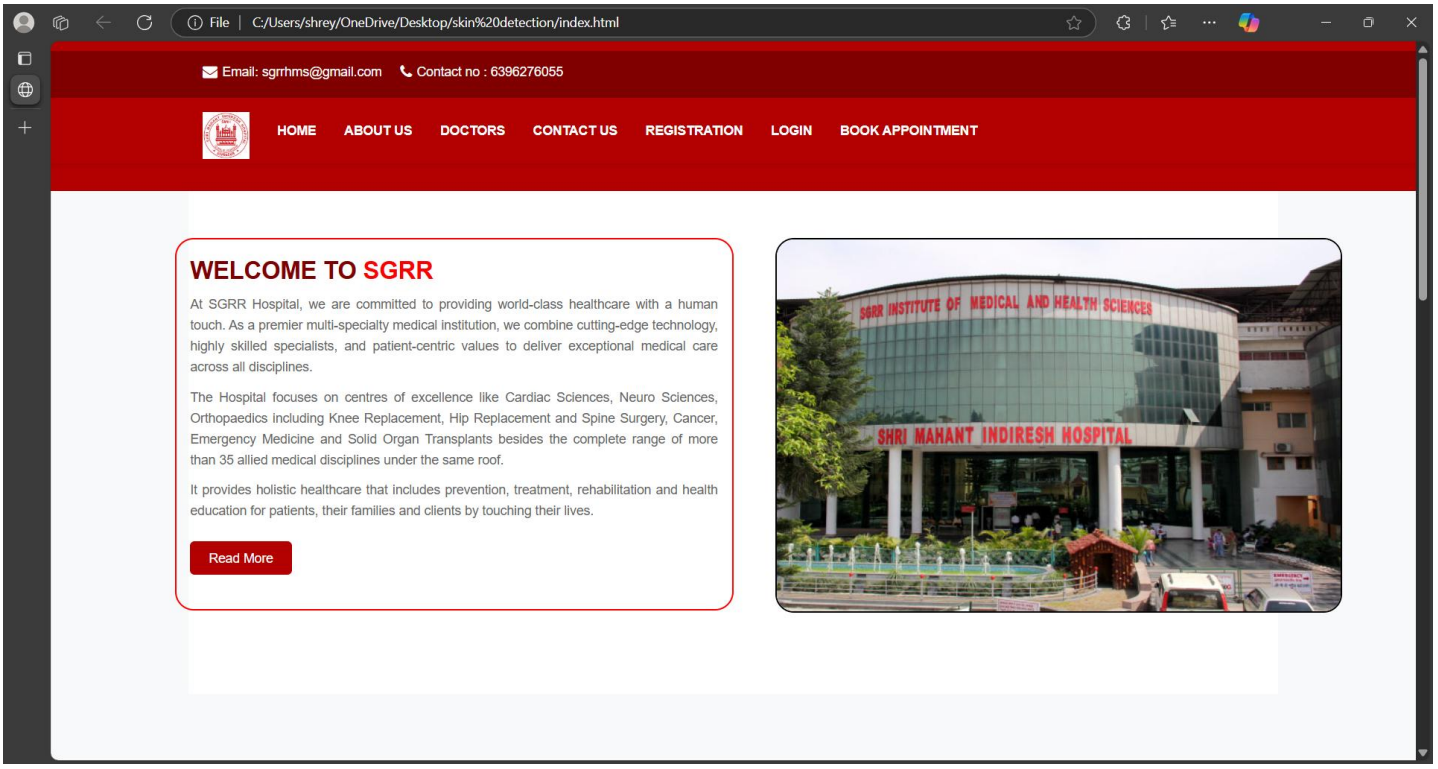
3. **Patient Actions:**
   - Register a new patient (if not registered).
   - Book an appointment by selecting a doctor, date, and time.
   - Upload an image for **Skin Disease Prediction**.
   - Receive diagnosis results and recommended actions.
   - View appointment details.
   - Check medical history (previous visits, prescriptions).
   - Logout.
   - .

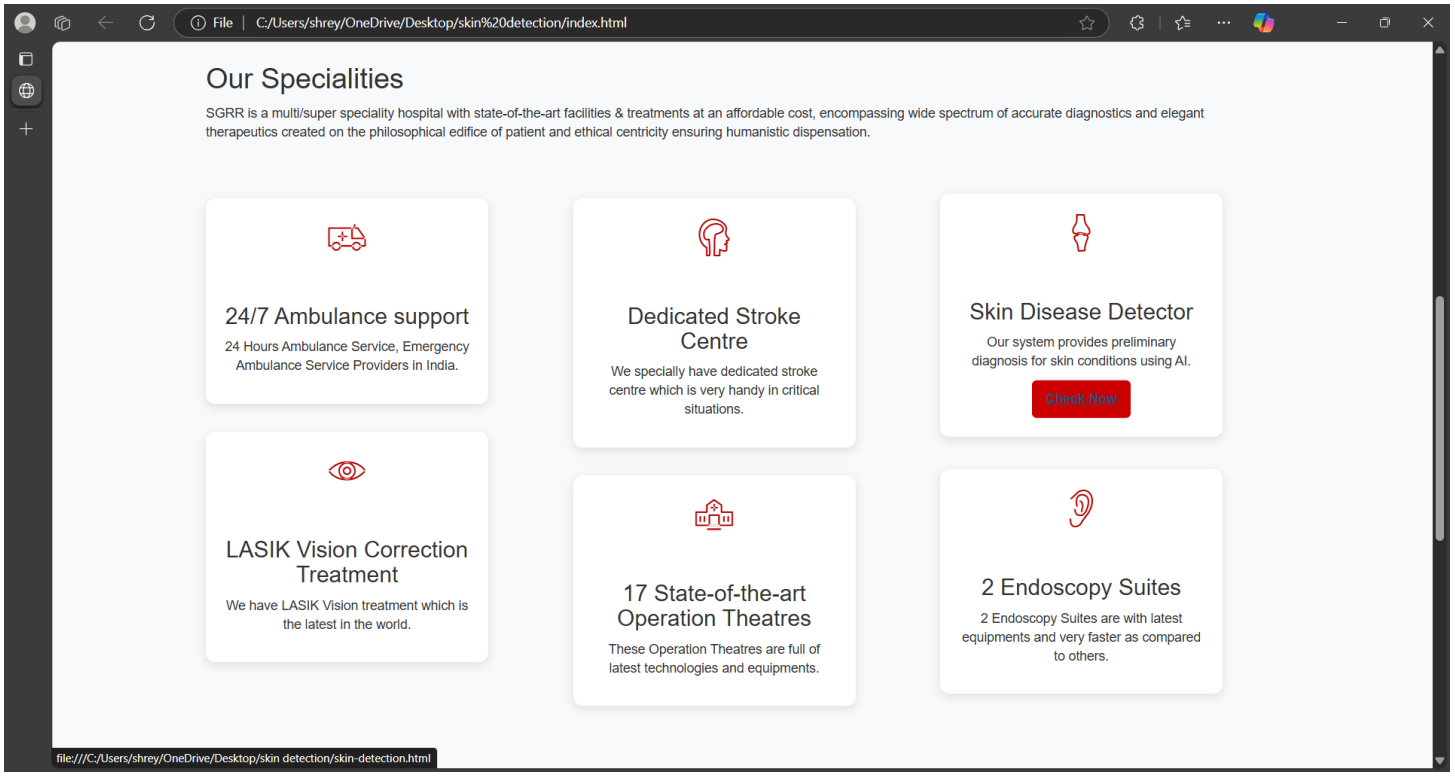
4. **Skin Disease Prediction:**
   - Patient uploads an image.
   - Image is processed and analyzed using a **machine learning model**.
   - Prediction result is displayed with confidence level.
   - Recommendation is provided based on diagnosis.

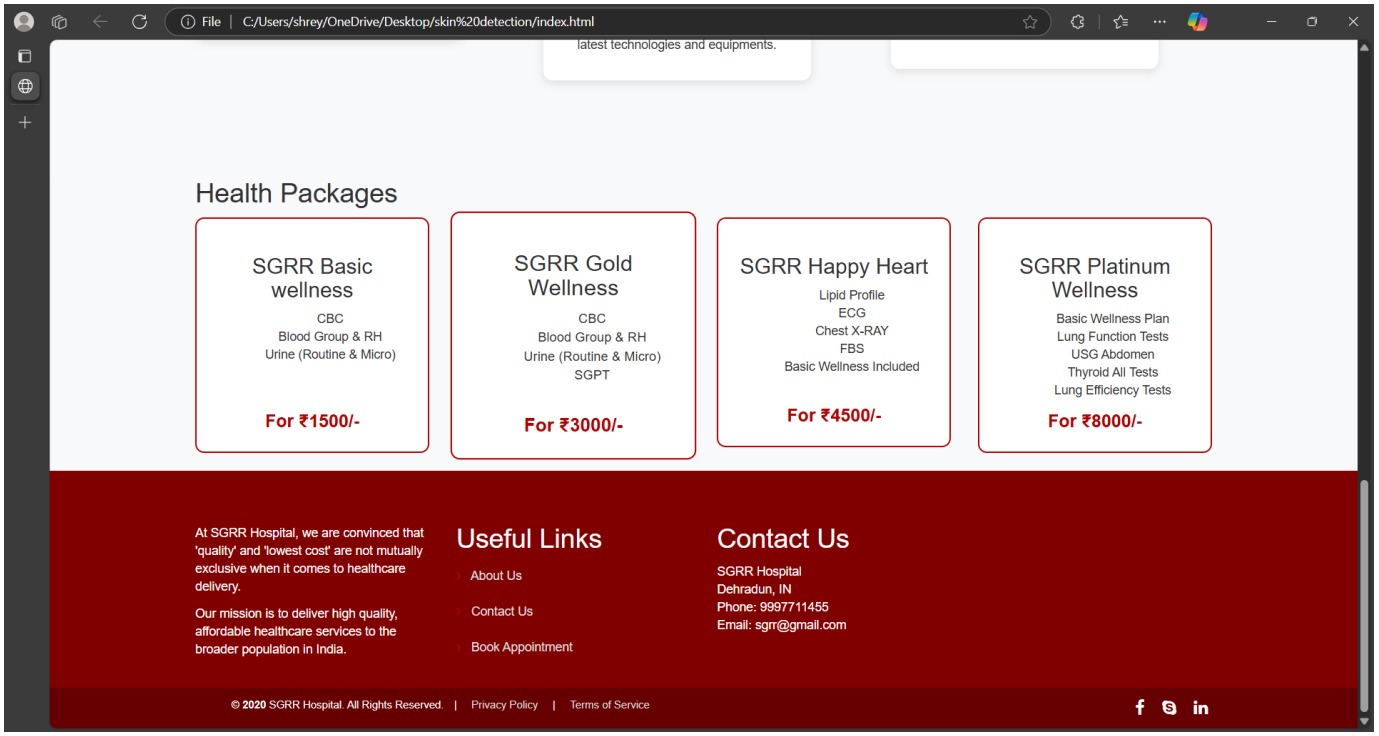
5. **Data Storage & Retrieval:**
   - Store all patient, doctor, and appointment details in **MySQL database**.
   - Store **Skin Disease Prediction** results for future reference.
   - Fetch and update records as needed.

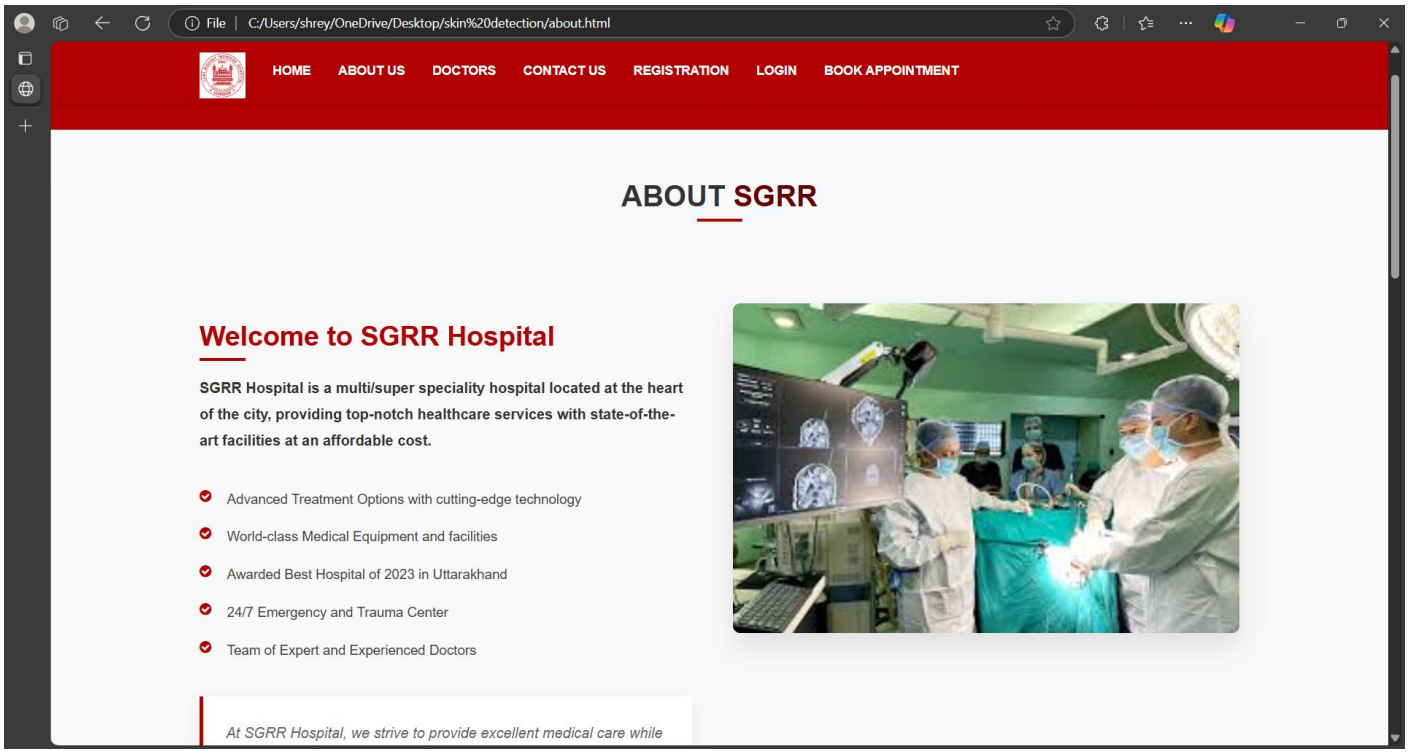
6. **End**

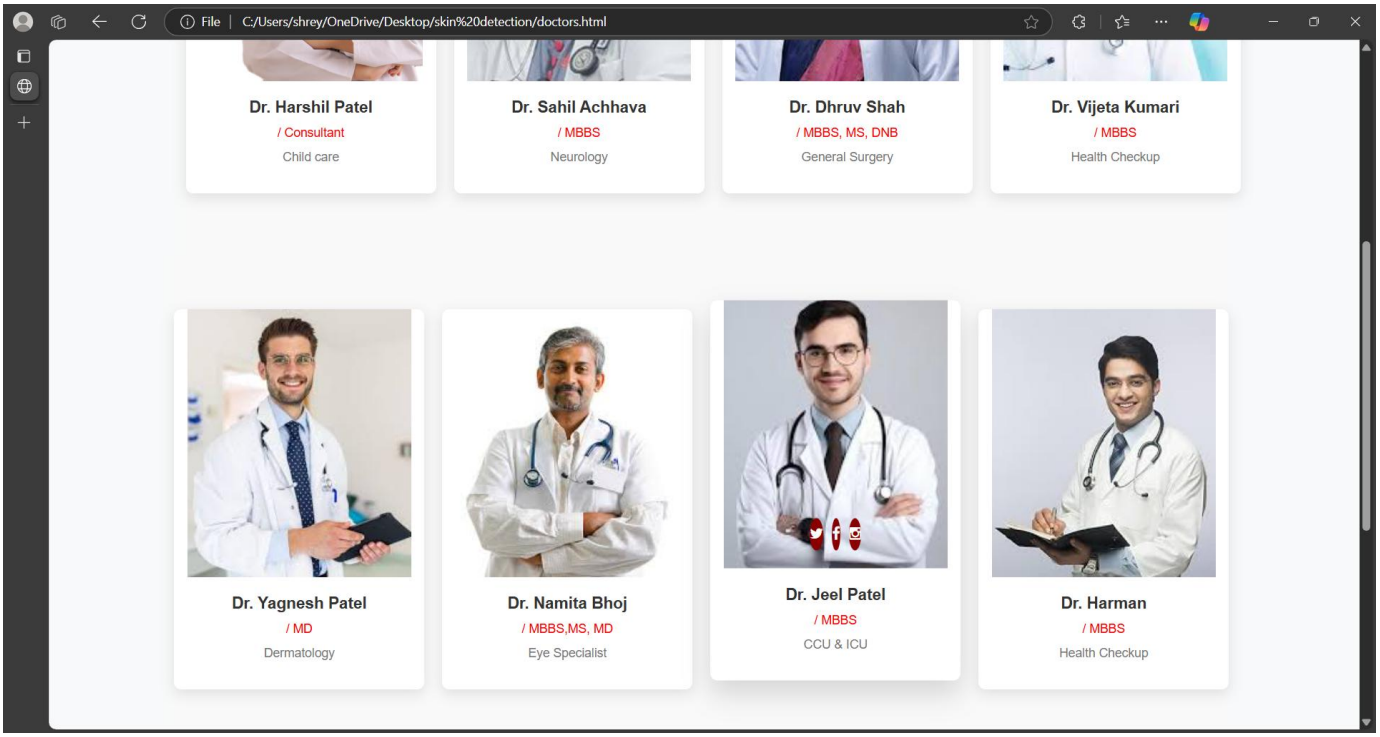
IMPLEMENTATION: (it should include all functionalities/modules)

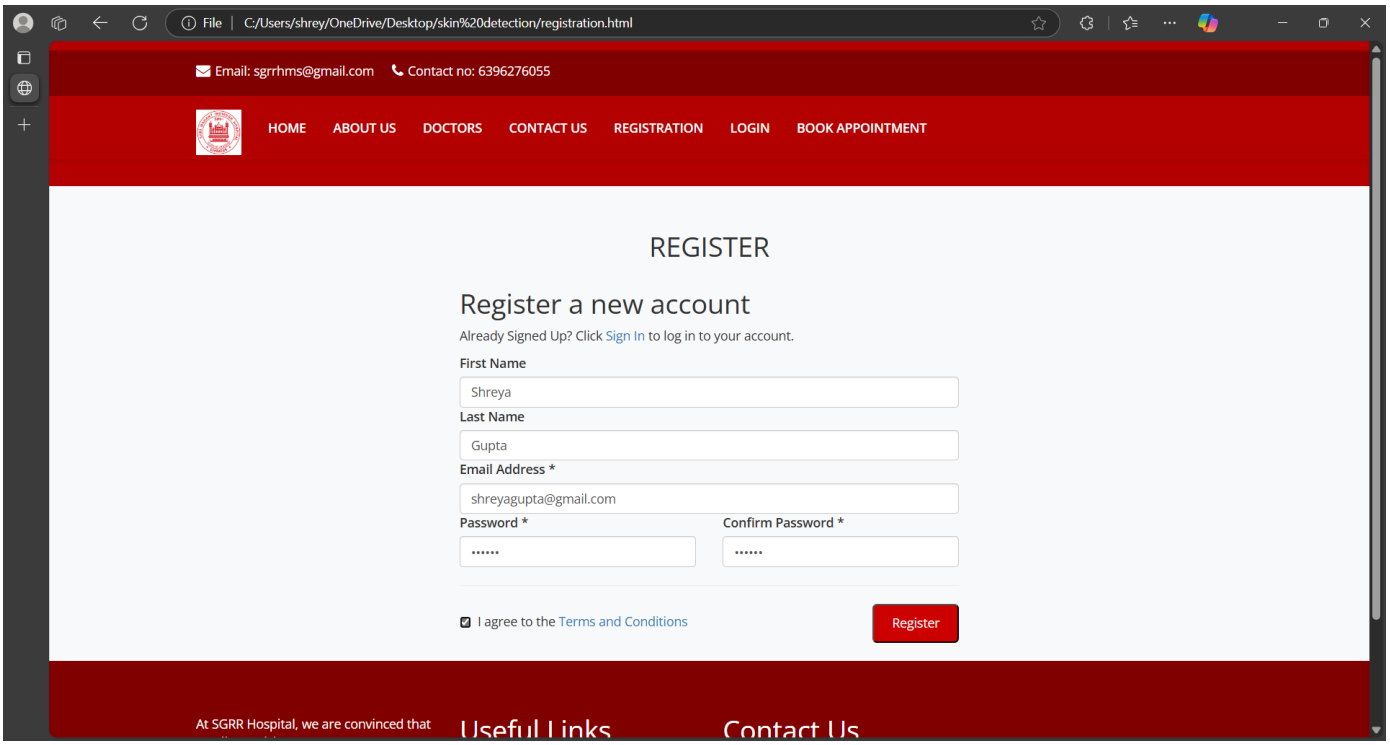


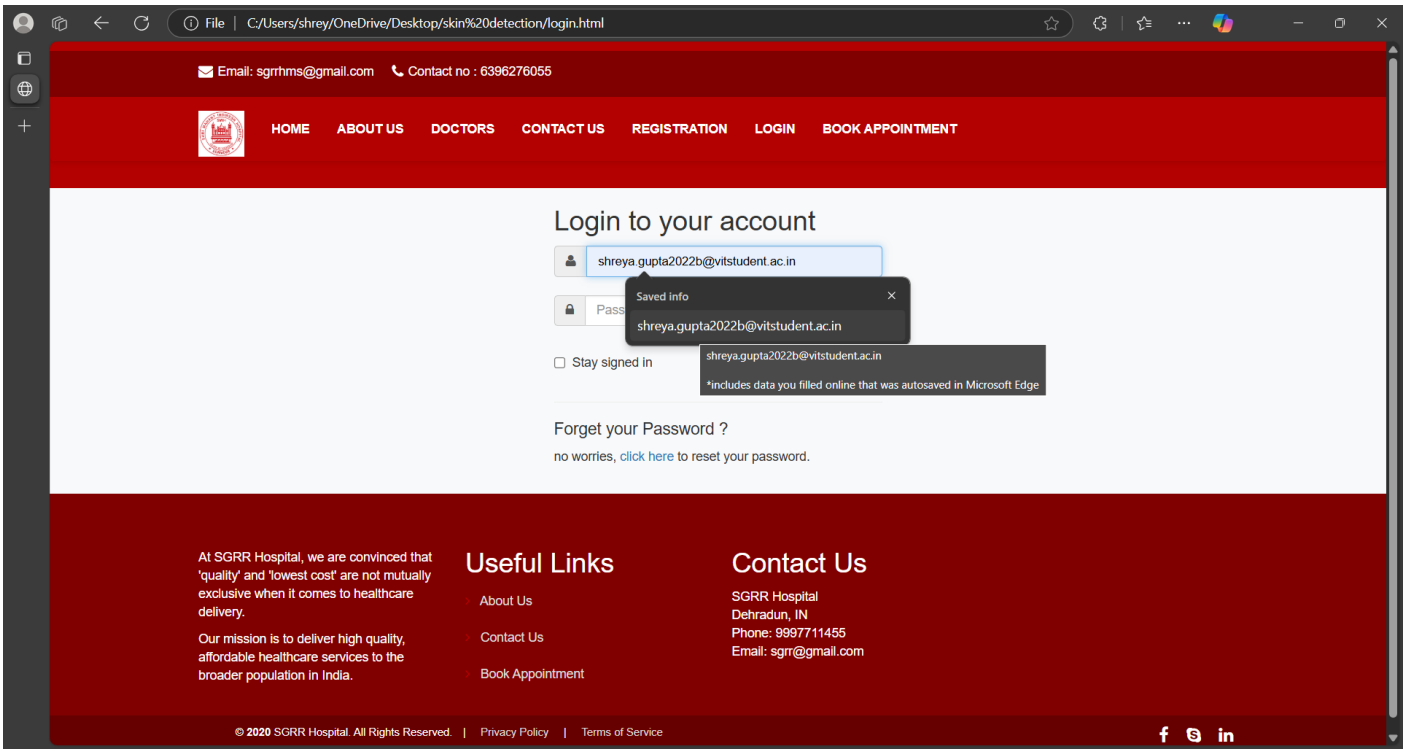


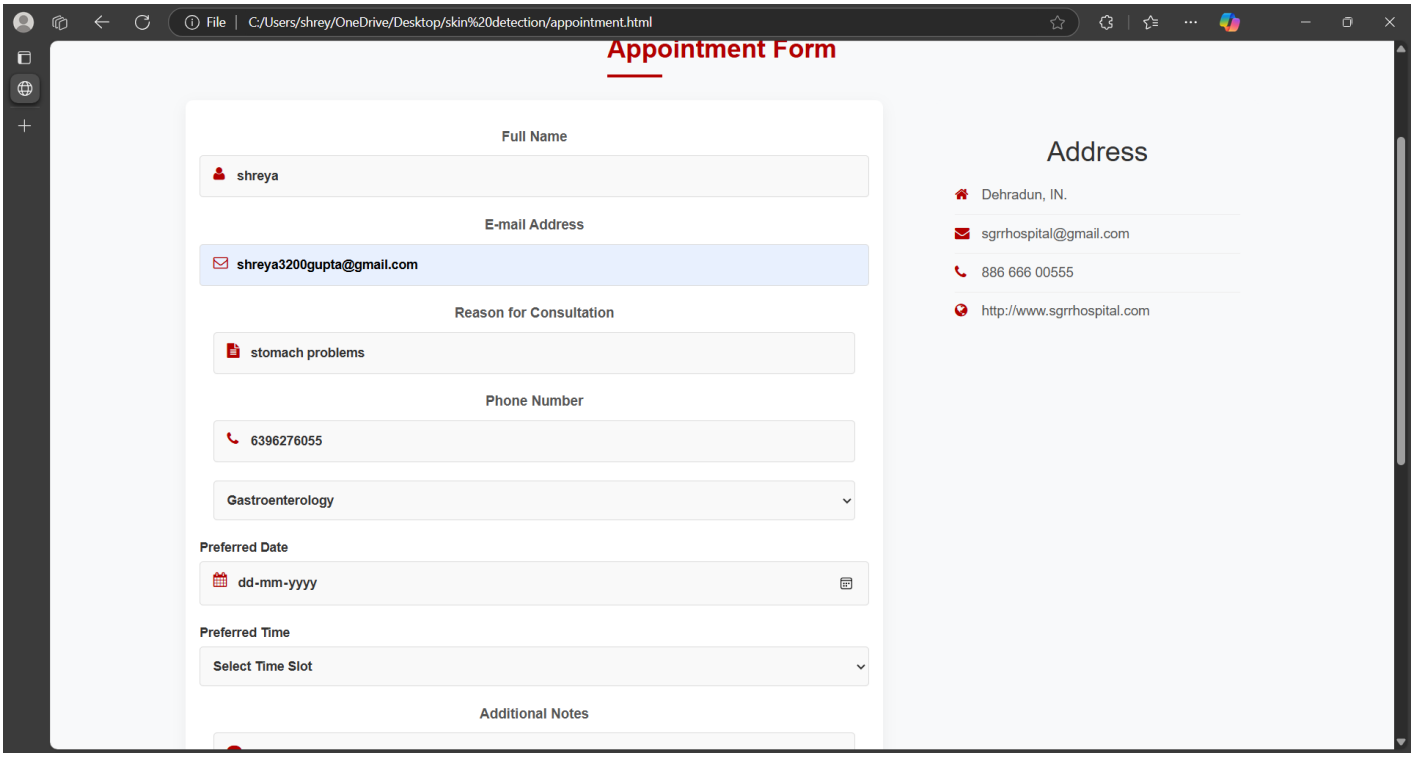

File | C:/Users/shrey/OneDrive/Desktop/skin%20detection/index.html
latest technologies and equipments.
Health Packages
SGRR Basic wellness
CBC
Blood Group & RH
Urine (Routine & Micro)
For ₹1500/-
SGRR Gold Wellness
CBC
Blood Group & RH
Urine (Routine & Micro)
SGPT
For ₹3000/-
SGRR Happy Heart
Lipid Profile
ECG
Chest X-RAY
FBS
Basic Wellness Included
For ₹4500/-
SGRR Platinum Wellness
Basic Wellness Plan
Lung Function Tests
USG Abdomen
Thyroid All Tests
Lung Efficiency Tests
For ₹8000/-
At SGRR Hospital, we are convinced that 'quality' and 'lowest cost' are not mutually exclusive when it comes to healthcare delivery.
Our mission is to deliver high quality, affordable healthcare services to the broader population in India.
Useful Links
About Us
Contact Us
Book Appointment
Contact Us
SGRR Hospital
Dehradun, IN
Phone: 9997711455
Email: sgrr@gmail.com
© 2020 SGRR Hospital. All Rights Reserved. | Privacy Policy | Terms of Service

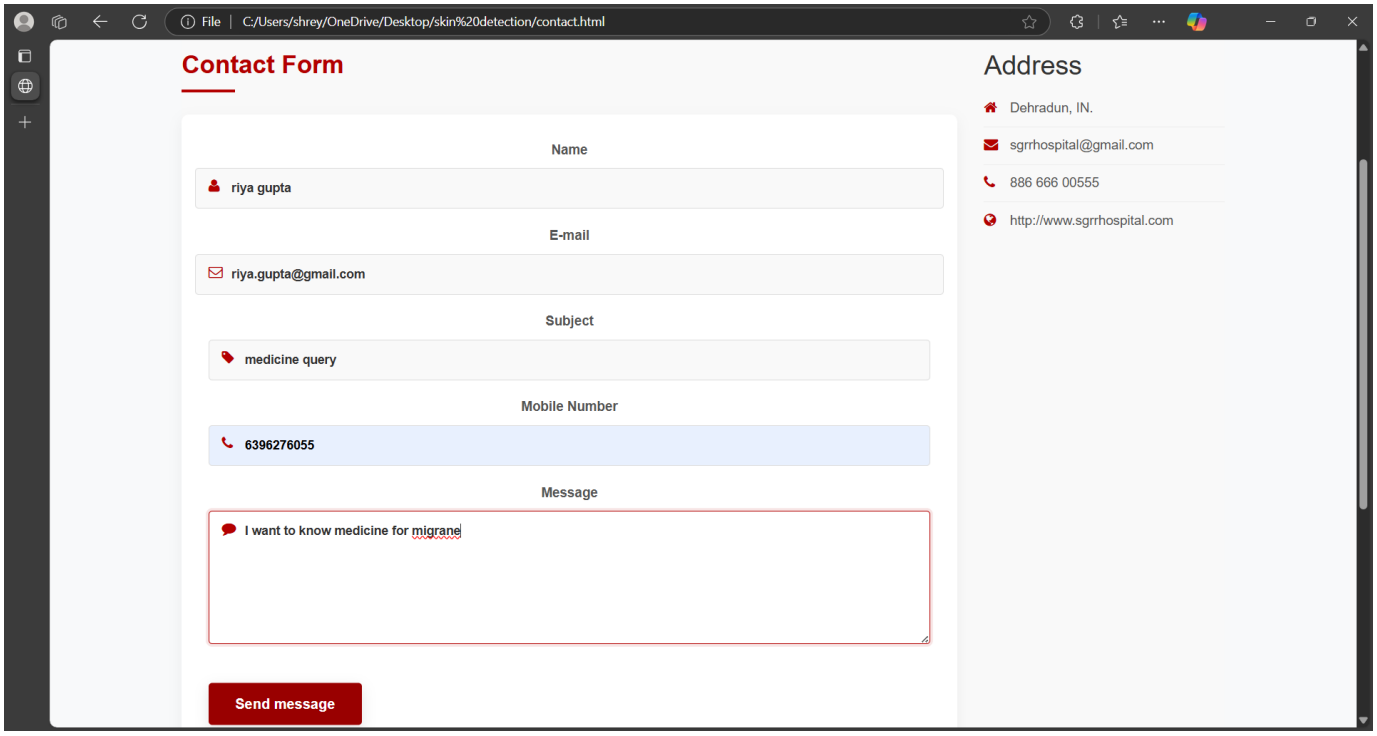

File | C:/Users/shrey/OneDrive/Desktop/skin%20detection/about.html
HOME
ABOUT US
DOCTORS
CONTACT US
REGISTRATION
LOGIN
BOOK APPOINTMENT
ABOUT SGRR
Welcome to SGRR Hospital
SGRR Hospital is a multi/super speciality hospital located at the heart of the city, providing top-notch healthcare services with state-of-the-art facilities at an affordable cost.
Advanced Treatment Options with cutting-edge technology
World-class Medical Equipment and facilities
Awarded Best Hospital of 2023 in Uttarakhand
24/7 Emergency and Trauma Center
Team of Expert and Experienced Doctors
At SGRR Hospital, we strive to provide excellent medical care while

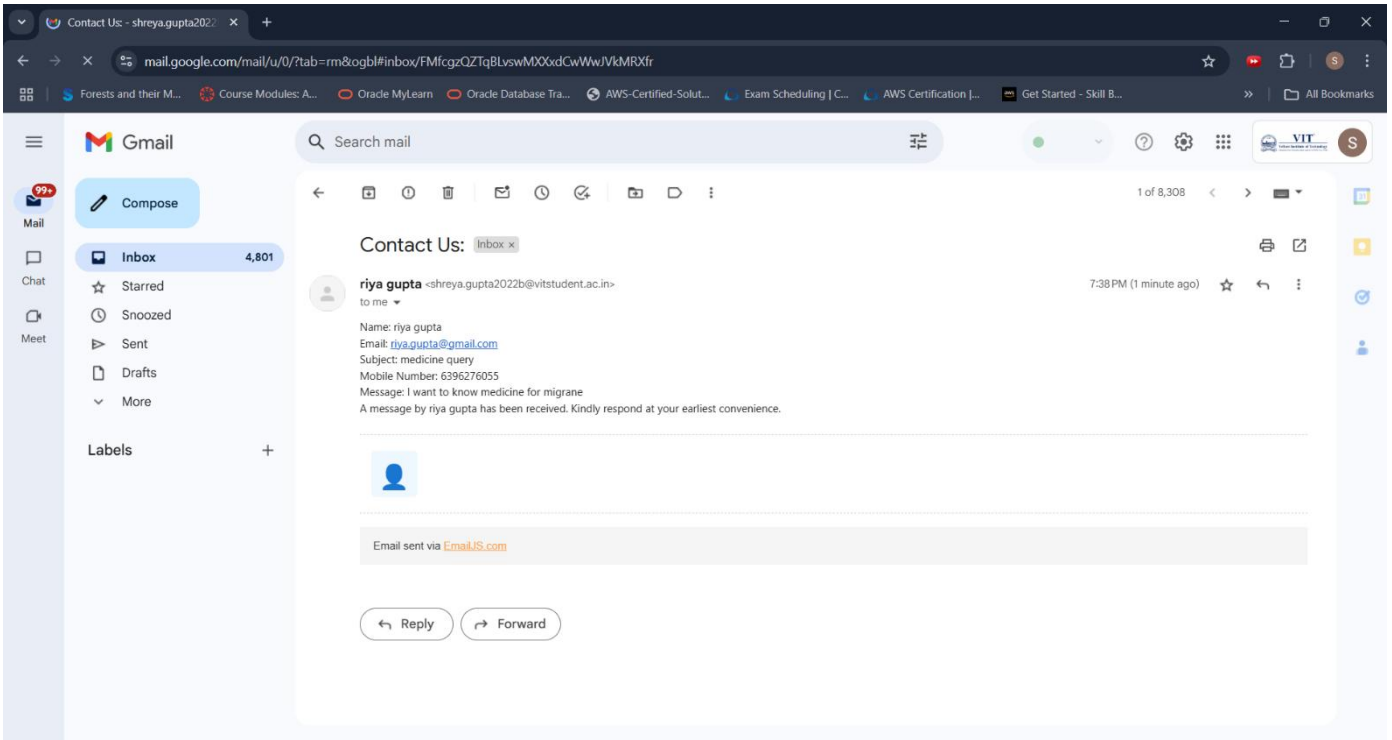

File | C:/Users/shrey/OneDrive/Desktop/skin%20detection/doctors.html
Dr. Harshil Patel
/ Consultant
Child care
Dr. Sahil Achhava
/ MBBS
Neurology
Dr. Dhruv Shah
/ MBBS, MS, DNB
General Surgery
Dr. Vijeta Kumari
/ MBBS
Health Checkup
Dr. Yagnesh Patel
/ MD
Dermatology
Dr. Namita Bhoj
/ MBBS,MS, MD
Eye Specialist
Dr. Jeel Patel
/ MBBS
CCU & ICU
Dr. Harman
/ MBBS
Health Checkup

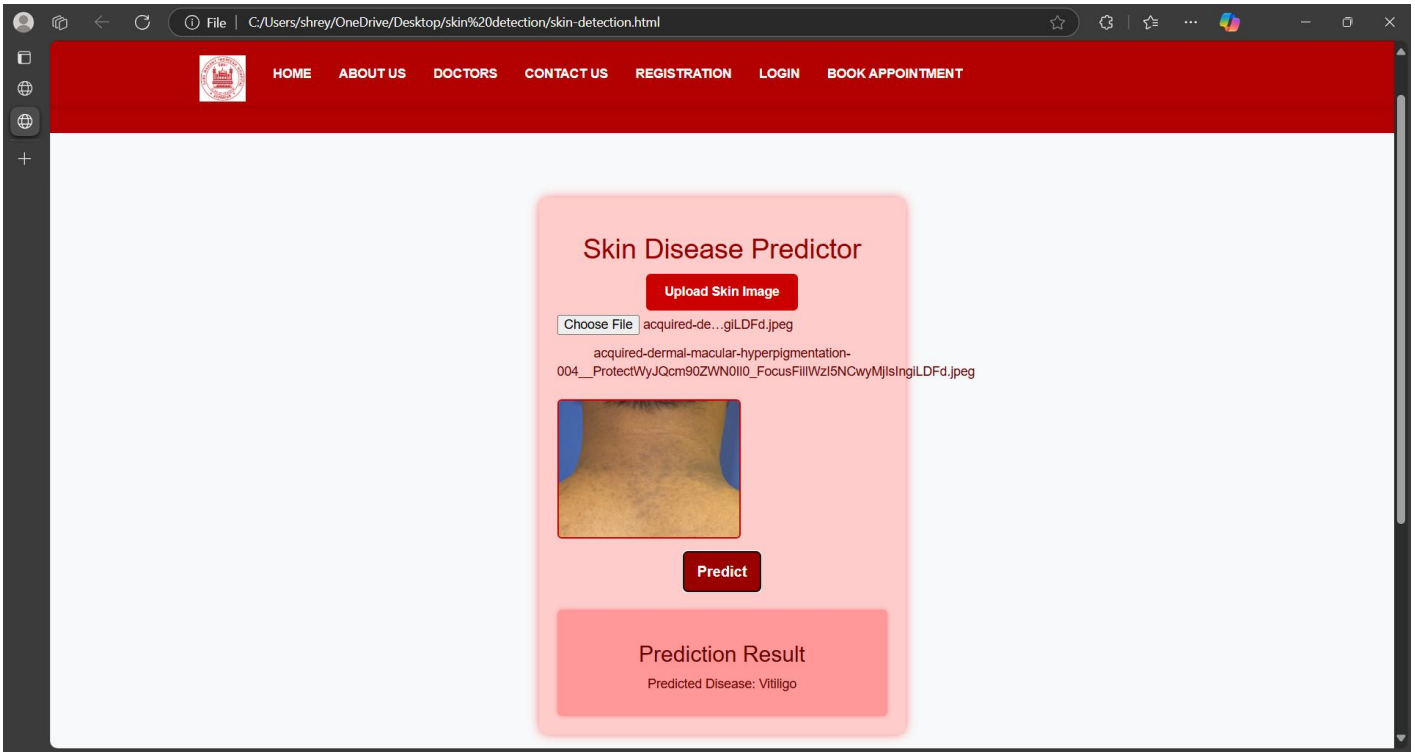

File | C:/Users/shrey/OneDrive/Desktop/skin%20detection/registration.html
Email: sgrrhms@gmail.com
Contact no: 6396276055
HOME
ABOUT US
DOCTORS
CONTACT US
REGISTRATION
LOGIN
BOOK APPOINTMENT
REGISTER
Register a new account
Already Signed Up? Click Sign In to log in to your account.
First Name
Shreya
Last Name
Gupta
Email Address *
shreyagupta@gmail.com
Password *
Confirm Password *
I agree to the Terms and Conditions
Register
At SGRR Hospital, we are convinced that
Useful Links
Contact Us


File | C:/Users/shrey/OneDrive/Desktop/skin%20detection/login.html
Email: sgrrhms@gmail.com
Contact no : 6396276055
HOME
ABOUT US
DOCTORS
CONTACT US
REGISTRATION
LOGIN
BOOK APPOINTMENT
Login to your account
shreya.gupta2022b@vitstudent.ac.in
Saved info
shreya.gupta2022b@vitstudent.ac.in
Stay signed in
shreya.gupta2022b@vitstudent.ac.in
*includes data you filled online that was autosaved in Microsoft Edge
Forget your Password ?
no worries, click here to reset your password.
At SGRR Hospital, we are convinced that 'quality' and 'lowest cost' are not mutually exclusive when it comes to healthcare delivery.
Our mission is to deliver high quality, affordable healthcare services to the broader population in India.
Useful Links
About Us
Contact Us
Book Appointment
Contact Us
SGRR Hospital
Dehradun, IN
Phone: 9997711455
Email: sgrr@gmail.com
© 2020 SGRR Hospital. All Rights Reserved. | Privacy Policy | Terms of Service


File | C:/Users/shrey/OneDrive/Desktop/skin%20detection/appointment.html
Appointment Form
Full Name
shreya
E-mail Address
shreya3200gupta@gmail.com
Reason for Consultation
stomach problems
Phone Number
6396276055
Gastroenterology
Preferred Date
dd-mm-yyyy
Preferred Time
Select Time Slot
Additional Notes
Address
Dehradun, IN.
sgrrhospital@gmail.com
886 666 00555
http://www.sgrrhospital.com


File | C:/Users/shrey/OneDrive/Desktop/skin%20detection/contact.html
Contact Form
Name
riya gupta
E-mail
riya.gupta@gmail.com
Subject
medicine query
Mobile Number
6396276055
Message
I want to know medicine for migrane
Send message
Address
Dehradun, IN.
sgrrhospital@gmail.com
886 666 00555
http://www.sgrrhospital.com


Contact Us: - shreya.gupta202
mail.google.com/mail/u/0/?tab=rm&ogbl#inbox/FMfcgzQZTqBLvswMXXxdCwWwJVkMRXfr
Gmail
Search mail
Compose
Inbox 4,801
Starred
Snoozed
Sent
Drafts
More
Labels
Mail
Chat
Meet
1 of 8,308
Contact Us: Inbox
riya gupta <shreya.gupta2022b@vitstudent.ac.in>
to me
7:38 PM (1 minute ago)
Name: riya gupta
Email: riya.gupta@gmail.com
Subject: medicine query
Mobile Number: 6396276055
Message: I want to know medicine for migrane
A message by riya gupta has been received. Kindly respond at your earliest convenience.
Email sent via EmailJS.com
Reply
Forward


File | C:/Users/shrey/OneDrive/Desktop/skin%20detection/skin-detection.html
HOME
ABOUT US
DOCTORS
CONTACT US
REGISTRATION
LOGIN
BOOK APPOINTMENT
Skin Disease Predictor
Upload Skin Image
Choose File
acquired-de…giLDFd.jpeg
acquired-dermal-macular-hyperpigmentation-004__ProtectWyJQcm90ZWN0Il0_FocusFillWzI5NCwyMjIsIngiLDFd.jpeg
Predict
Prediction Result
Predicted Disease: Vitiligo

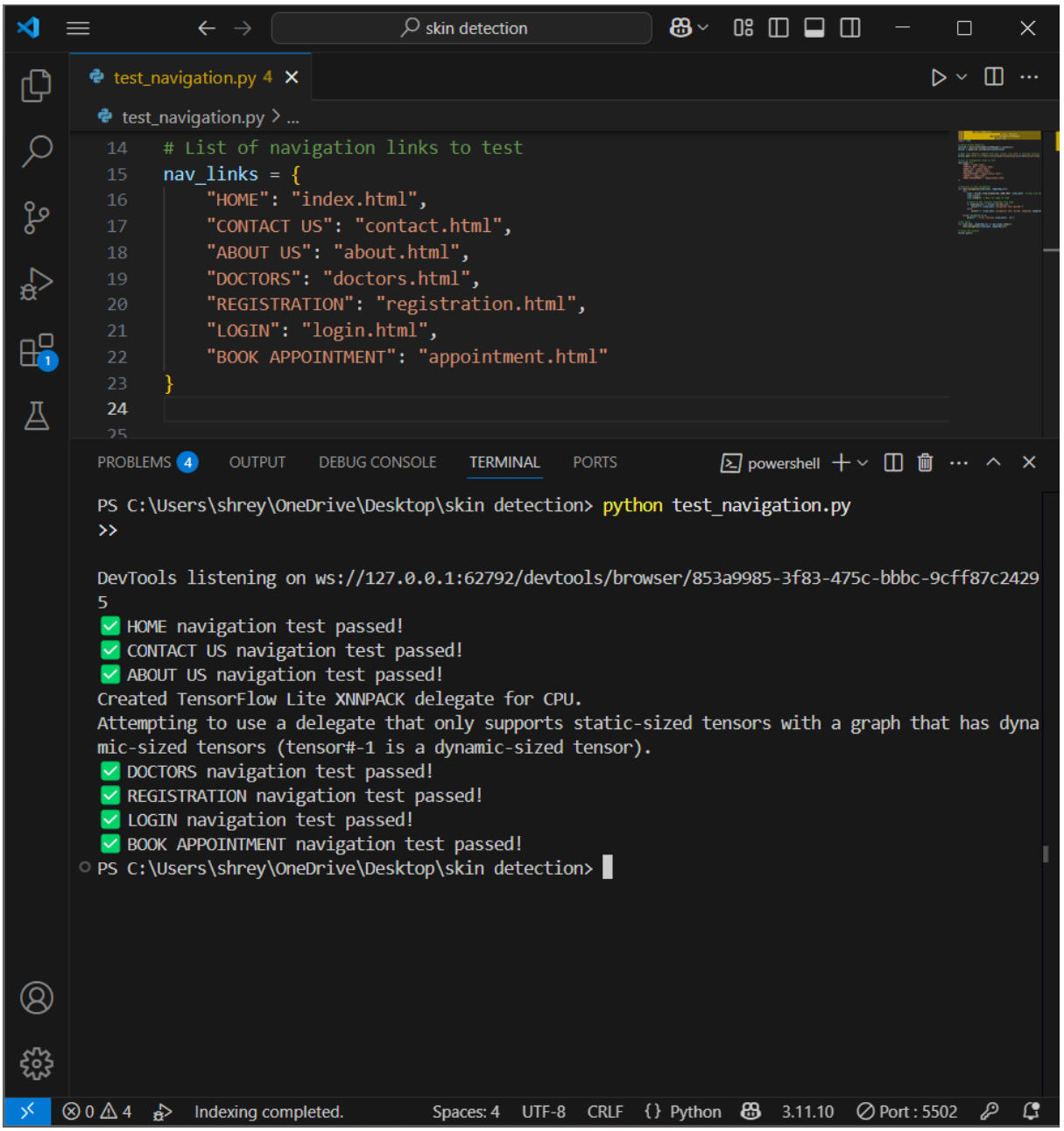
TESTING (Test document as prepared in DA)

Test 1: This test ensures that the navigation links, such as "HOME" and "CONTACT US," redirect users to the correct pages when clicked, confirming that the site's navigation functions as expected.

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By
import time

# Setup Chrome WebDriver
service = Service(ChromeDriverManager().install())
driver = webdriver.Chrome(service=service)

# Open your website (update with your actual file path if testing locally)
driver.get("file:///C:/Users/shrey/OneDrive/Desktop/skin%20detection/index.html")

# List of navigation links to test
nav_links = {
    "HOME": "index.html",
    "CONTACT US": "contact.html",
    "ABOUT US": "about.html",
    "DOCTORS": "doctors.html",
    "REGISTRATION": "registration.html",
    "LOGIN": "login.html",
    "BOOK APPOINTMENT": "appointment.html"
}

# Function to test navigation
def test_navigation(link_text, expected_url):
    try:
        link = driver.find_element(By.LINK_TEXT, link_text)  # Find link by text
        link.click()
        time.sleep(2)  # Wait for page to load

        # Check if URL contains expected file name
        if expected_url in driver.current_url:
            print(f"☑ {link_text} navigation test passed!")
        else:
            print(f"❌ {link_text} navigation test failed. Expected {expected_url}, but got
{driver.current_url}")

    except Exception as e:
        print(f"⚠ Error testing {link_text}: {e}")

# Run tests
for link_text, expected_url in nav_links.items():
```

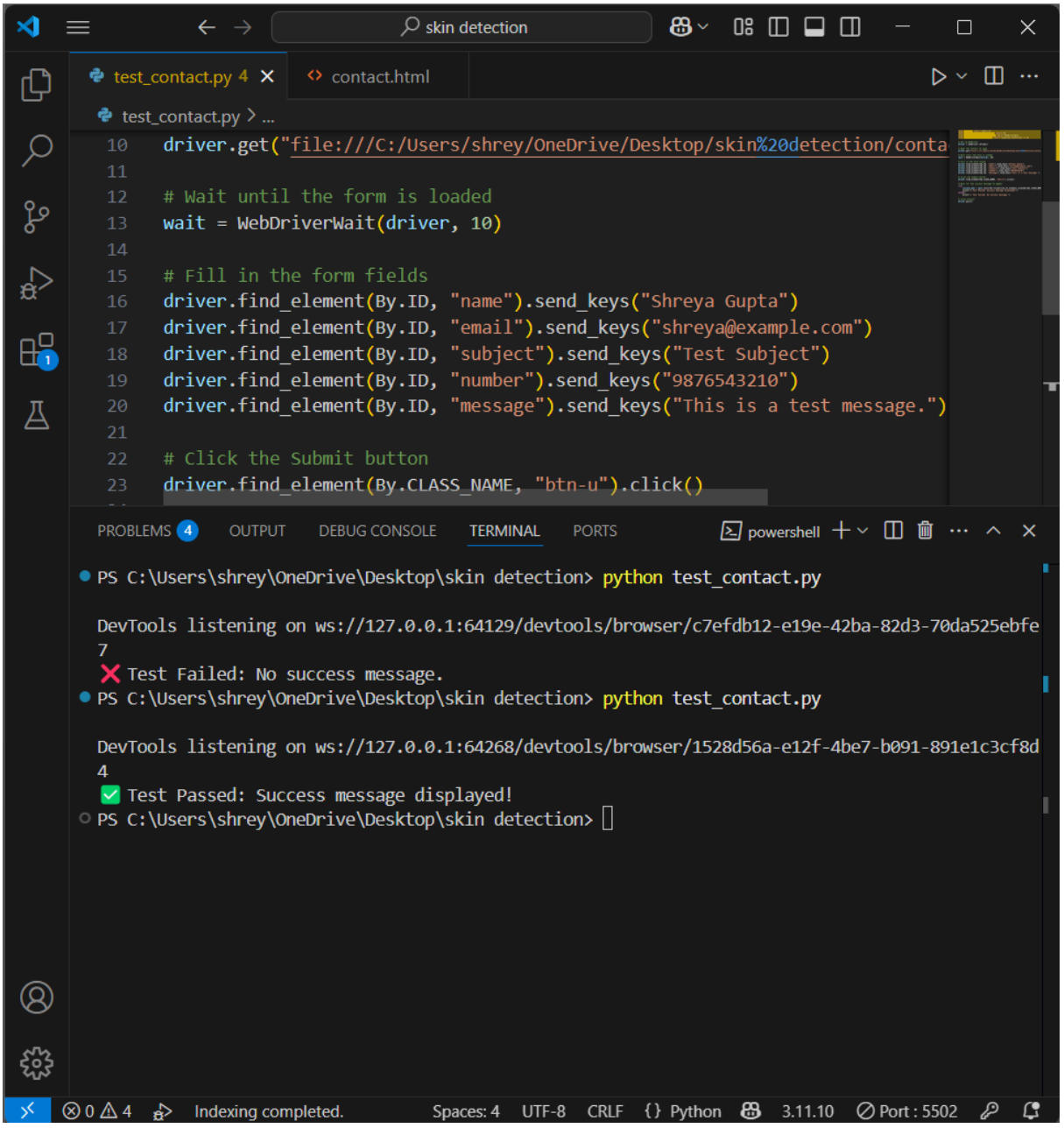
test_navigation(link_text, expected_url)

# Close the browser
driver.quit()



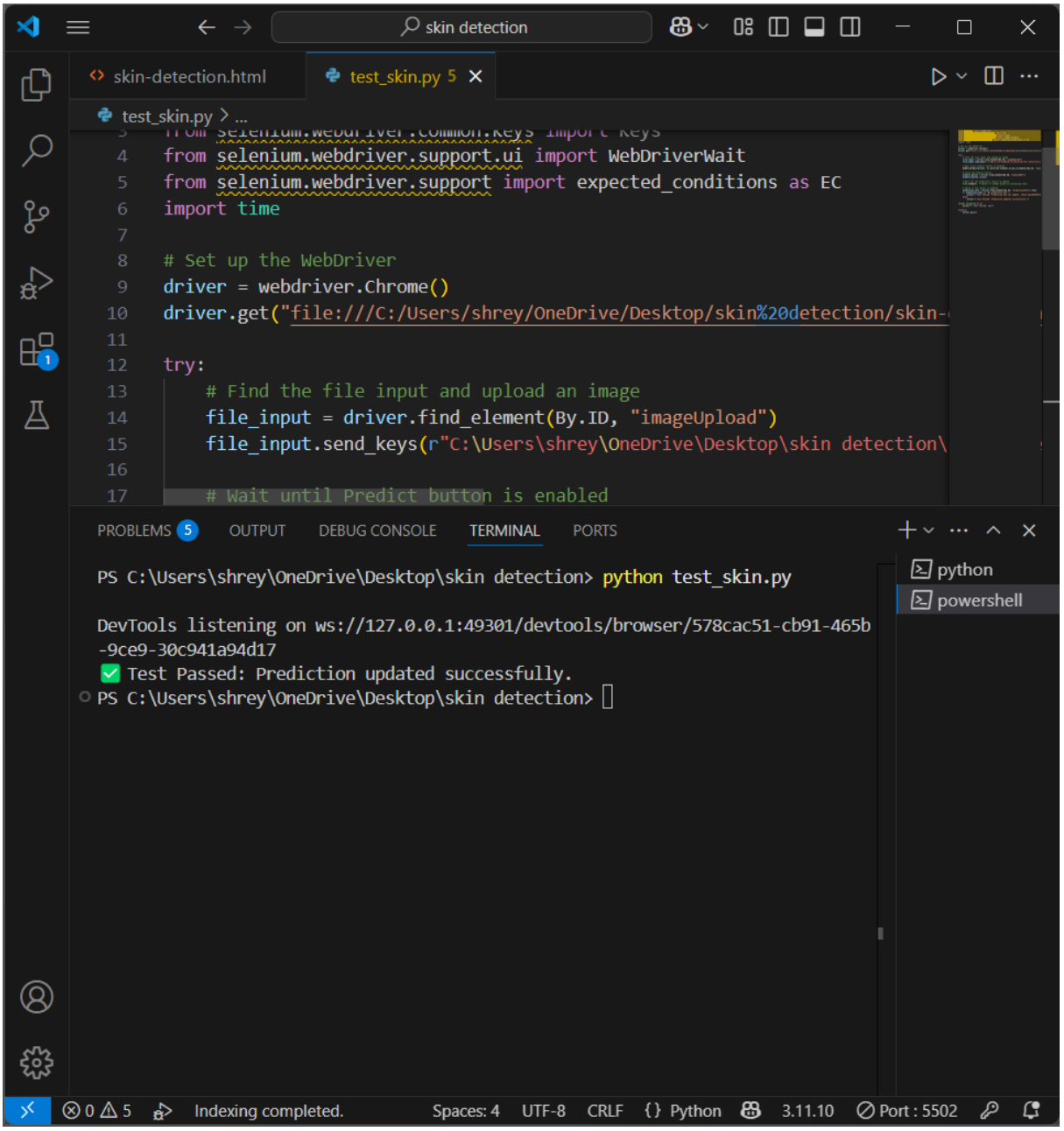
Test 2: This test case fills out the contact form, submits it, and waits to verify if the success message appears, indicating the email was sent successfully via EmailJS.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

# Set up WebDriver
driver = webdriver.Chrome()

# Open the Contact Us page
driver.get("file:///C:/Users/shrey/OneDrive/Desktop/skin%20detection/contact.html")

# Wait until the form is loaded
wait = WebDriverWait(driver, 10)

# Fill in the form fields
driver.find_element(By.ID, "name").send_keys("Shreya Gupta")
driver.find_element(By.ID, "email").send_keys("shreya@example.com")
driver.find_element(By.ID, "subject").send_keys("Test Subject")
driver.find_element(By.ID, "number").send_keys("9876543210")
driver.find_element(By.ID, "message").send_keys("This is a test message.")

# Click the Submit button
driver.find_element(By.CLASS_NAME, "btn-u").click()

# Wait for the success message to appear
try:
    success_box = wait.until(EC.visibility_of_element_located((By.CLASS_NAME,
"successBox")))
    print("☑ Test Passed: Success message displayed!")
except:
    print("❌ Test Failed: No success message.")

# Close browser
driver.quit()
```


skin detection
test_contact.py 4
contact.html
test_contact.py > ...
10 driver.get("file:///C:/Users/shrey/OneDrive/Desktop/skin%20detection/conta
11
12 # Wait until the form is loaded
13 wait = WebDriverWait(driver, 10)
14
15 # Fill in the form fields
16 driver.find_element(By.ID, "name").send_keys("Shreya Gupta")
17 driver.find_element(By.ID, "email").send_keys("shreya@example.com")
18 driver.find_element(By.ID, "subject").send_keys("Test Subject")
19 driver.find_element(By.ID, "number").send_keys("9876543210")
20 driver.find_element(By.ID, "message").send_keys("This is a test message.")
21
22 # Click the Submit button
23 driver.find_element(By.CLASS_NAME, "btn-u").click()
PROBLEMS 4
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
PS C:\Users\shrey\OneDrive\Desktop\skin detection> python test_contact.py
DevTools listening on ws://127.0.0.1:64129/devtools/browser/c7efdb12-e19e-42ba-82d3-70da525ebfe7
Test Failed: No success message.
PS C:\Users\shrey\OneDrive\Desktop\skin detection> python test_contact.py
DevTools listening on ws://127.0.0.1:64268/devtools/browser/1528d56a-e12f-4be7-b091-891e1c3cf8d4
Test Passed: Success message displayed!
PS C:\Users\shrey\OneDrive\Desktop\skin detection>
0 4
Indexing completed.
Spaces: 4
UTF-8
CRLF
Python
3.11.10
Port : 5502

Test case 3: This Selenium test script verifies that uploading an image enables the "Predict" button and updates the prediction result correctly. It includes proper waits and error handling for reliability.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time

# Set up the WebDriver
driver = webdriver.Chrome()
driver.get("file:///C:/Users/shrey/OneDrive/Desktop/skin%20detection/skin-detection.html")

try:
    # Find the file input and upload an image
    file_input = driver.find_element(By.ID, "imageUpload")
    file_input.send_keys(r"C:\Users\shrey\OneDrive\Desktop\skin
detection\skin_diseases\Vitiligo\Crop-
0350_0300vitiligo1__ProtectWyJQcm90ZWN0Il0_FocusFillWzI5NCwyMjIsIngiLDFd.jpg")

    # Wait until Predict button is enabled
    WebDriverWait(driver, 5).until(EC.element_to_be_clickable((By.ID, "analyzeBtn")))

    # Click the Predict button
    predict_button = driver.find_element(By.ID, "analyzeBtn")
    predict_button.click()

    # Wait for the prediction result to update
    time.sleep(3)  # Adjust if needed based on processing time

    # Check if the result is updated
    prediction_text = driver.find_element(By.ID, "predictionText").text
    if prediction_text == "No prediction yet.":
        print("❌ Test Failed: Prediction did not update. Check uploadAndPredict() function.")
    else:
        print("☑ Test Passed: Prediction updated successfully.")

except Exception as e:
    print(f"❌ Test Failed: {e}")

finally:
    driver.quit()
```


skin detection
skin-detection.html
test_skin.py 5
test_skin.py > ...
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time
# Set up the WebDriver
driver = webdriver.Chrome()
driver.get("file:///C:/Users/shrey/OneDrive/Desktop/skin%20detection/skin-
try:
# Find the file input and upload an image
file_input = driver.find_element(By.ID, "imageUpload")
file_input.send_keys(r"C:\Users\shrey\OneDrive\Desktop\skin detection\
Wait until Predict button is enabled
PROBLEMS 5
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\shrey\OneDrive\Desktop\skin detection> python test_skin.py
DevTools listening on ws://127.0.0.1:49301/devtools/browser/578cac51-cb91-465b-9ce9-30c941a94d17
Test Passed: Prediction updated successfully.
PS C:\Users\shrey\OneDrive\Desktop\skin detection>
python
powershell
Indexing completed.
Spaces: 4
UTF-8
CRLF
Python
3.11.10
Port : 5502

CONCLUSION:

The **Hospital Management System (HMS)** is designed to streamline hospital operations by managing patient records, doctor appointments, and essential administrative tasks efficiently. By integrating **MySQL for data management**, the system ensures secure and structured storage of patient and hospital-related information. The intuitive user interface, developed using **HTML, CSS, and JavaScript**, provides ease of access and navigation for both patients and hospital staff.

This system not only enhances **appointment scheduling and patient record management** but also improves **overall hospital efficiency** by reducing manual workload and minimizing errors. Future enhancements could include **mobile compatibility, automated reminders, and integration with medical devices** to further improve patient care and hospital services.

With its user-friendly approach and essential healthcare functionalities, the **HMS serves as a reliable solution** for modern healthcare institutions, ensuring a **seamless and organized workflow**.

REFERENCES:

1. https://data.mendeley.com/datasets/3hckgznc67/1
2. https://www.sgrru.ac.in/
3. https://www.lucidchart.com
4. https://staruml.io/download/
5. https://www.python.org/downloads/release/python-3110/
6. https://www.emailjs.com/
7. https://getbootstrap.com/
8 https://fontawesome.com/
9. https://fonts.google.com/
10. https://www.adroitinfosystems.com/products/ehospital-systems/