Faculty of Engineering - Cairo University

Credit Hours System – Spring 2025

SBES240 – Requirements Engineering for Digital Health

Radiology Department

Team 1

Name	ID
Khadiga Ali	4230337
Habiba Mamdouh	4230192
Caroline Ehab	4230163
Mohamed Mostafa	4230197

Table of Contents

Introduction and objective:

We were asked to make a website of any department of our choice. We decided on radiology department. Radiology department has lots of pages as each type of scan has its own page. The type of scan that we included in our website are

- MRI
- CT SCAN
- X-RAY
- ULTRASOUND

Each of them has some features that they share together. Also, we included the main page which is a homepage which has the most features as log in and sign up also search features and a nav bar where the user can decide which scan he wants.

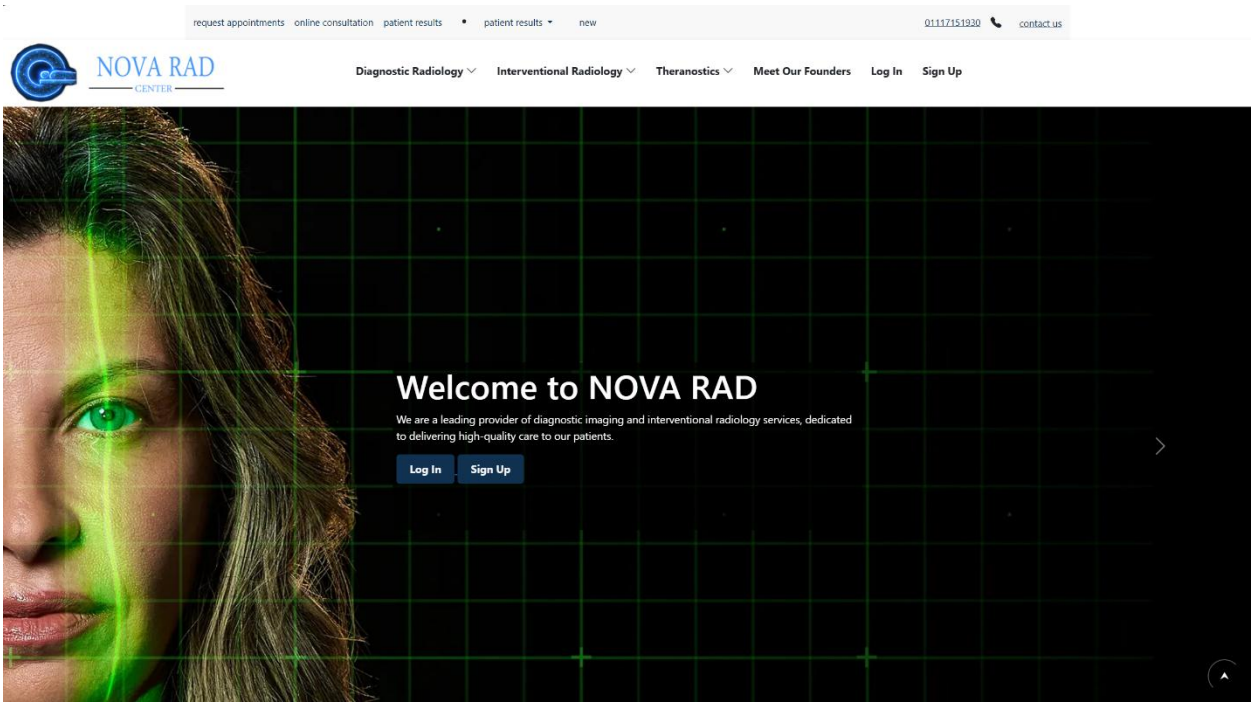
Technologies Used

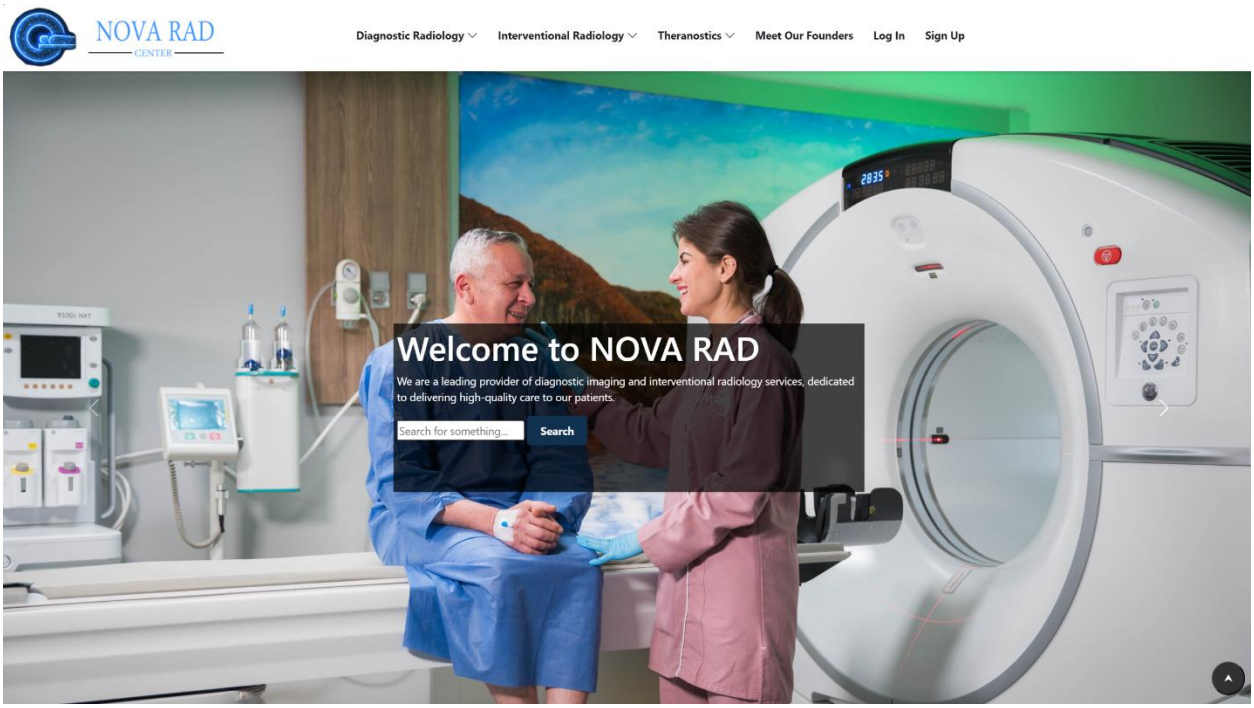
- Backend
  - Python 3.x
  - Flask web framework
  - PostgreSQL database
- Frontend
  - HTML5
  - CSS3
- JavaScript
  - Bootstrap for responsive design
  - Boxicons for icons
- Database
  - PostgreSQL hosted on Neon Tech
  - We used some languages psycopg2 for database connections

Home page UI:

The first page that appears to the user, it is user-friendly helps with the navigation through the whole website and ensures the efficiency of the patient experience.
It includes the following elements:

- **Page header:** it includes all the services provided by the radiology department website (e.g. Appointment booking, online consultation).
- **Department logo:** we designed a personal logo that includes the center's name to catch the users' eyes.
- **Navbar:** it includes all types of imaging provided by the center and it links to separate imaging pages that has all the details the patients need to be aware of, it also links the patient to the Founders introduction page, furthermore it allows the patients to access the Login, Sign up pages.
- **Moving slides:** we provided more information about the center and welcomes the user, the second slide allows the user to search for whatever they need on the website.
- **Our services:** additional way for the patient to access the imaging pages.
- **Working hours:** in provides the patient with further details about the opening hours of the center.
- **Achievements:** we stated all the achievement to gain the patients trust.
- **Footer:** it has a "contact us" button, the locations of the Ceneters' branches, number, the contact email, and copyrights for our design.

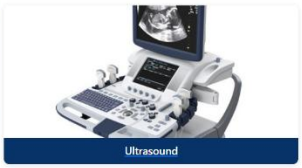

request appointments
online consultation
patient results
patient results
new
01117151930
contact us
NOVA RAD
CENTER
Diagnostic Radiology
Interventional Radiology
Theranostics
Meet Our Founders
Log In
Sign Up
Welcome to NOVA RAD
We are a leading provider of diagnostic imaging and interventional radiology services, dedicated to delivering high-quality care to our patients.
Log In
Sign Up


NOVA RAD
CENTER
Diagnostic Radiology
Interventional Radiology
Theranostics
Meet Our Founders
Log In
Sign Up
Welcome to NOVA RAD
We are a leading provider of diagnostic imaging and interventional radiology services, dedicated to delivering high-quality care to our patients.
Search for something...
Search

Our Services


MRI


CT Scan


Ultrasound


Digital X-ray

Working Hours

Sunday - Thursday	9:00 AM - 11:00 PM
Friday - Saturday	10:00 AM - 5:00 PM

LONG HISTORY OF SUCCESS
Since 1989

EXPERIENCED DOCTORS
and Staff

ADVANCED TECHNOLOGY
and Techniques

BEST PATIENT QUALITY
and Care

RESERVE AN APPOINTMENT OR CONSULT A DOCTOR?

Contact Us

11 Marghani Street, Heliopolis, Cairo

HCC, Off 90 St., by MRC Square, New Cairo

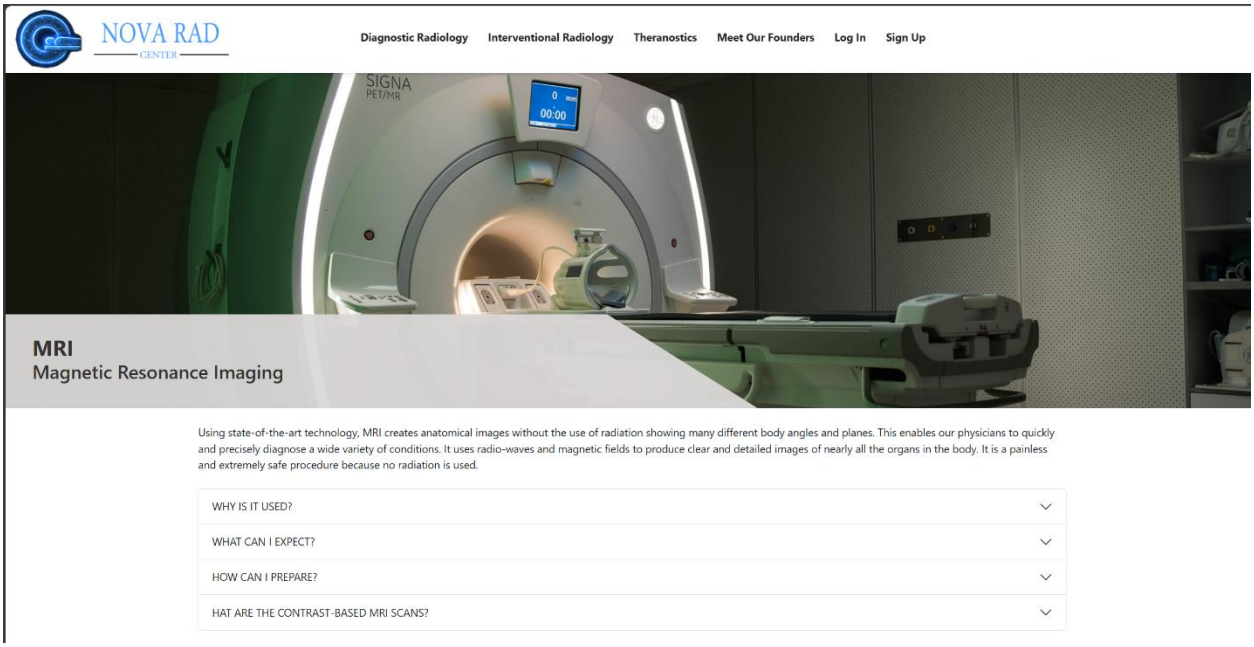
8 Mostafa El-Nahas Street, Nasr City, Cairo

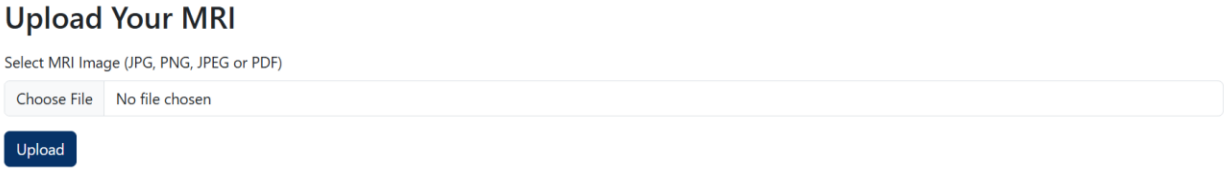
19773

novarad

Scan Pages:

For each of the MRI, Ultrasound, CT, and X-Ray scan pages we added a picture of the room at the top of the page, then an intro about the device and the type of the scan. Drop down menus were also added to make the interface more interactive and user-friendly, also it was added so the patient can be informed about why each scan is used, how to prepare for it.





After the patient performs the scan, they can upload it to their doctor using this form that only accepts images or pdf files. When an image is uploaded it's previewed along with the green text "Successfully uploaded!". If another extension gets uploaded an error message is showed "Invalid format. Please only submit an image or a pdf." The shown images are only of the MRI scan page but the exact same was done for the 4 types of scans we have.

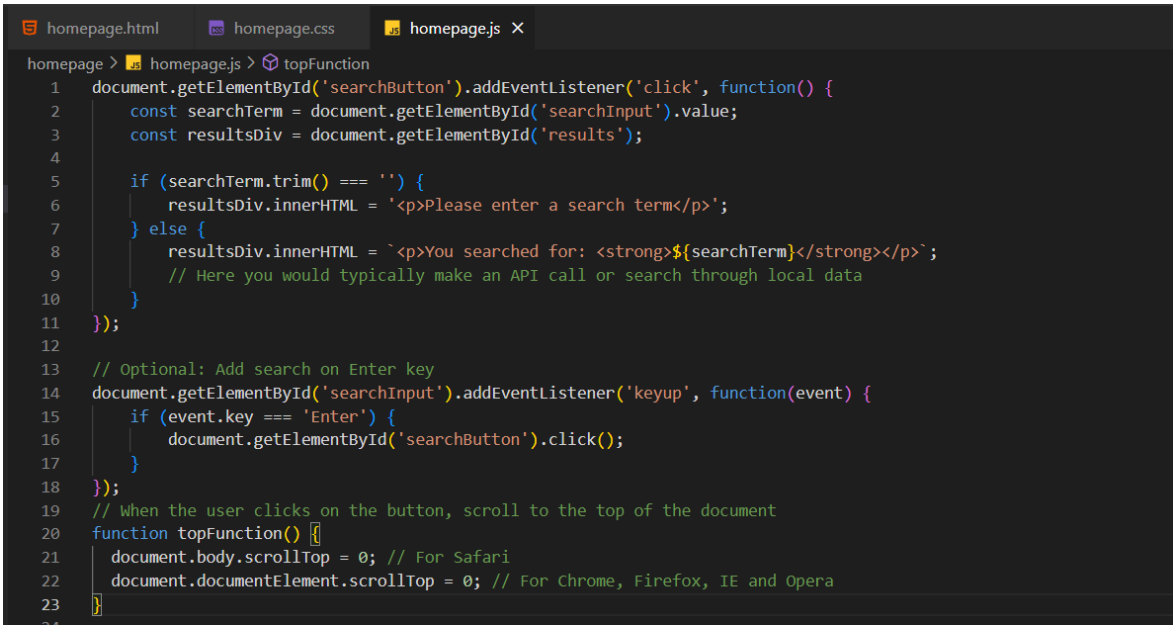
Folder Structure:

```
Final Project/
├── app.py                # Main Flask application
├── static/               # Static files (CSS, JS, images)
│   ├── images/           # Image assets
│   ├── homepage.css      # CSS for homepage
│   ├── x-ray.css         # CSS for X-ray page
│   └── founders.css      # CSS for founders page
├── templates/            # HTML templates
│   ├── homepage.html     # Homepage template
│   ├── x-ray.html        # X-ray service page
│   ├── MRI.html          # MRI service page
│   ├── CT.html           # CT scan service page
│   ├── Founders.html     # Founders information page
│   ├── loginPA.html      # Patient login page
│   └── registerPA.html # Patient registration page
```

Coding:

As mentioned before the languages used now, we'll show you some parts of the code and show an example for each language.

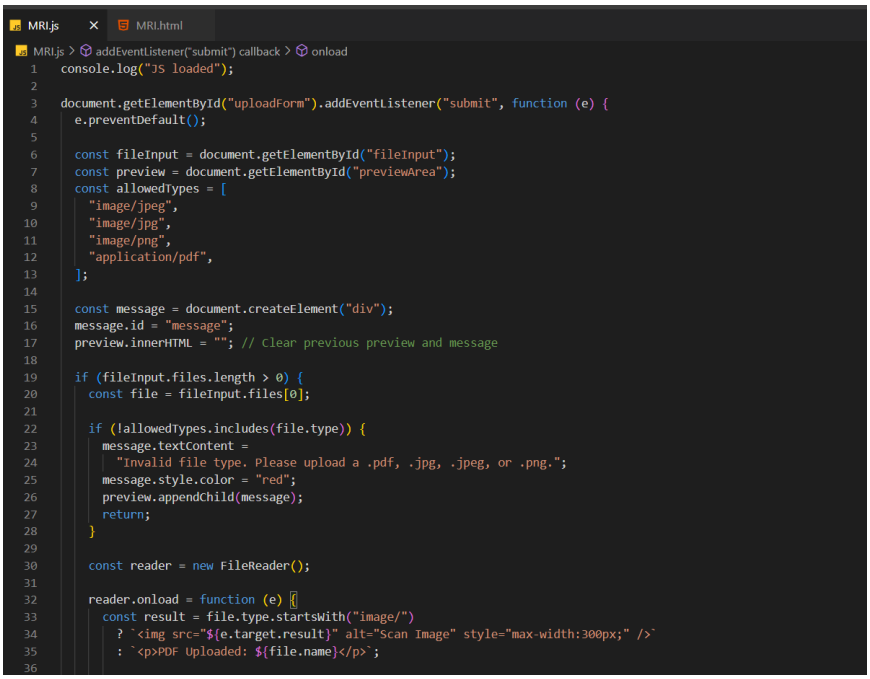
JavaScript:



This code which is in the home page consists of 2 parts. The first part is about the search bar where we write the text, we want to search about then it appears underneath we have an input text that we write into and then after clicking the button or clicking the enter button.

The second part is about a button that reference to the top of the website.

We used 2 lines for it as it differs from one browser to another.



```
MRI.js  MRI.html
MRI.js > addEventListener("submit") callback > onload
console.log("JS loaded");

document.getElementById("uploadForm").addEventListener("submit", function (e) {
  e.preventDefault();

  const fileInput = document.getElementById("fileInput");
  const preview = document.getElementById("previewArea");
  const allowedTypes = [
    "image/jpeg",
    "image/jpg",
    "image/png",
    "application/pdf",
  ];

  const message = document.createElement("div");
  message.id = "message";
  preview.innerHTML = ""; // Clear previous preview and message

  if (fileInput.files.length > 0) {
    const file = fileInput.files[0];

    if (!allowedTypes.includes(file.type)) {
      message.textContent =
        "Invalid file type. Please upload a .pdf, .jpg, .jpeg, or .png.";
      message.style.color = "red";
      preview.appendChild(message);
      return;
    }

    const reader = new FileReader();

    reader.onload = function (e) {
      const result = file.type.startsWith("image/")
        ? `<img src="${e.target.result}" alt="Scan Image" style="max-width:300px;" />`
        : `<p>PDF Uploaded: ${file.name}</p>`;

      preview.innerHTML = `
        <p>Uploaded File:</p>
        ${result}
      `;

      message.textContent = "Successfully uploaded!";
      message.style.color = "green";
      preview.appendChild(message);
    };

    reader.readAsDataURL(file);
  }
});

// Dropdown toggle function
function toggleDrop(dropElement) {
  dropElement.classList.toggle("active");
}
```

This java code is used in all our scan pages which is about uploading a picture of the paper scan so it could be then read by the doctor. It contains a button after clicking on it opens the folder on the laptop so he can choose the picture he wants to upload, and it also show a preview.

There is also a part for toggling between the different bar in each page.

CSS:

For styling of the pages, we used bootstrap to improve the responsiveness and used different elements like container for example, however we personalized the color scheme, and made the navbar from scratch, actions like colour changes when an element is hovered on to enhance our website. So, in conclusion we mixed between bootstrap elements and written stylesheets.

```css
@media (max-width: 768px) {
  header {
    display: none;
  }

  nav div ul, nav div select, .bi-search, nav div button {
    width: 300px;
    height: 100vh;
    position: fixed;
    top: 180px;
    z-index: 2000;
    background-color: var(--bgcolor);
    border-bottom: 1px solid #5C707E;
    right: -100%;
    padding: 0px;
    margin: 0px;
    transition: all 1s ease-in-out;
  }
```

HTML:

```html
<body>
  <header class="container">
    <p>request appointments</p>
    <p>online consultation</p>
    <p>patient results</p>
    <p>
      <li class="nav-item dropdown">
        <button class="btn  dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">
          patient results
        </button>

      </li>
    </p>
    <p>new</p>
    <p></p>
    <a href="contact">01117151930</a>
    <i class="bi bi-telephone-fill"></i>
    <span><a href="contact_us">contact us</a></span>
  </header>

  <nav>
    <div>
      <a href="{{url_for('index')}}"><img src="{{ url_for('static', filename='images/Final_logo.png') }}" alt="Logo"
          class="logo"></a>
    </div>
```

This is a part of the html code and in here we started to place every part of the page as header, nav, section and divs. Also, we added some links for as for the contact us part. We added all the images we wanted and the inputs that are responsible for the data entered that is then worked on with the JavaScript.

Backend:

This is a Flask application forms the backend, it is used to streamline patient registration for a radiology department. It handles the patient authentication and implements a comprehensive registration process.

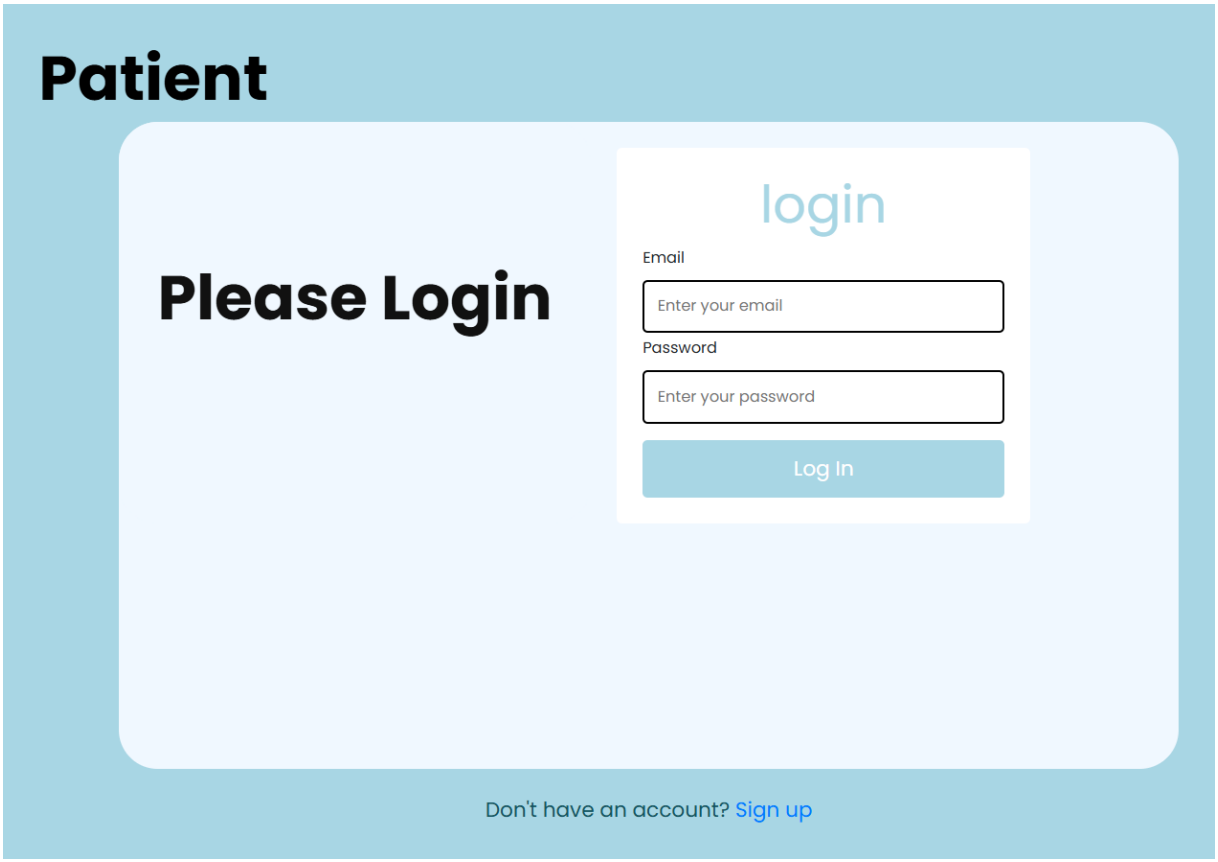
app.py

```python
from flask import Flask, session, render_template, request, redirect, url_for, flash, jsonify
import psycopg2
import psycopg2.extras
import re
import os
from datetime import datetime, timedelta

PGHOST = 'ep-green-lake-a5nsemie.us-east-2.aws.neon.tech'
PGDATABASE = 'neondb'
PGUSER = 'neondb_owner'
PGPASSWORD = '4ZFkXlMWTJA2'
app = Flask(__name__)
db_connection_session = psycopg2.connect(dbname='Surgery_Department', user=PGUSER, password=PGPASSWORD, host=PGHOST,
                                         port=5432)
app.secret_key = '123'

@app.route('/')
def index():···
@app.route('/about')

@app.route('/x-ray')
def x_ray():···

@app.route('/MRI')
def mri():···

@app.route('/CT')
def ct():···

@app.route('/US')
def US():···

@app.route('/founders')
def founders():···


@app.route('/loginPA', methods=['GET', 'POST'])
def login_pa():···


@app.route('/registerPA', methods=['GET', 'POST'])
def register_pa():···
```

Log in:

- Email and password validation
- Database credential verification
- Handles failed login trials

```python
@app.route('/loginPA', methods=['GET', 'POST'])
def login_pa():
    if request.method == 'GET':
        return render_template('loginPA.html')

    elif request.method == 'POST':
        email = request.form.get('Pemail')
        password = request.form.get('Ppassword')

        if not email or not password:
            flash('Please enter both email and password', 'error')
            return redirect(url_for('login_pa'))

        # Query the database for patient email and password
        cur = db_connection_session.cursor(cursor_factory=psycopg2.extras.DictCursor)
        cur.execute('SELECT * FROM patient WHERE P_email = %s AND P_password = %s', (email, password))
        patient_data = cur.fetchone()
        if patient_data is None:
            flash('Incorrect email or password', 'error')
            return render_template('loginPA.html')
        else:
            # Store patient data and image filename in the session
            session['patient'] = patient_data

            return redirect(url_for('index'))
```

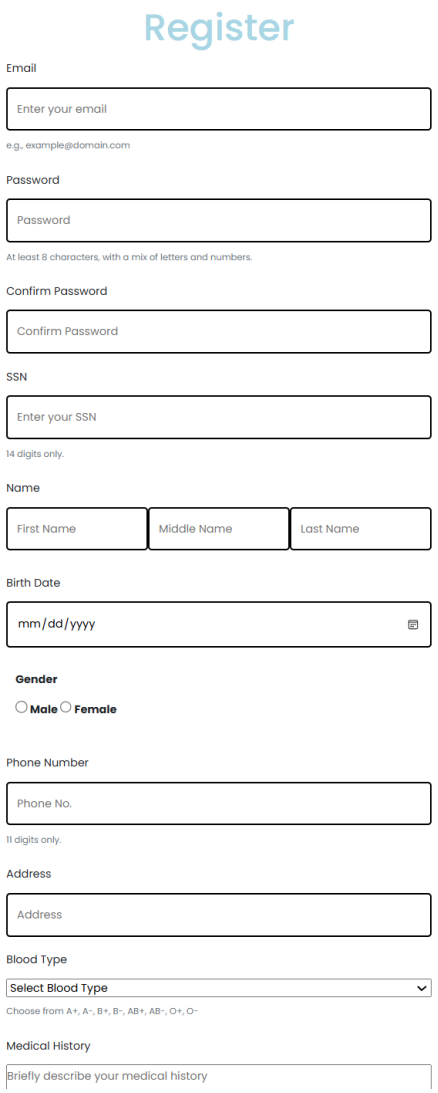


Register:

- Collects patient Information (name, SSN, gender, date of birth, address, phone, email)
- Collects medical data (blood type, medical history)
- Performs some validation:
    - Password strength and matching
    - SSN format validation (14 digits)
    - Phone number validation (11 digits)
    - Email format verification
    - Make sure that email is not already found in database
- After all those validations, data is added to the database

```python
@app.route('/registerPA', methods=['GET', 'POST'])
def register_pa():
    if request.method == 'GET':
        return render_template('registerPA.html')

    elif request.method == 'POST':
        # Get form data
        SSN = request.form.get('SSN')
        first_name = request.form.get('fname')
        middle_name = request.form.get('mname')
        last_name = request.form.get('lname')
        gender = request.form.get('gender')
        date_of_birth = request.form.get('DOB')
        phone_number = request.form.get('phone')
        address = request.form.get('add')
        blood_type = request.form.get('bt')
        medical_history = request.form.get('mh')
        email = request.form.get('email')
        password = request.form.get('password')
        confirm_password = request.form.get('confirm_password')


        # Validate the form data
        if len(password) < 8:
            flash('Password must be at least 8 characters long', 'error')
        elif password != confirm_password:
            flash('Passwords do not match', 'error')
        elif not SSN.isdigit() or len(SSN) != 14:
            flash('SSN is invalid', 'error')
        elif not phone_number.isdigit() or len(phone_number) != 11:
            flash('Phone number is invalid', 'error')
        elif not re.match(r"[^@]+@[^@]+\.[^@]+", email):
            flash('Invalid email address', 'error')
        else:
            # Check if the email is already registered
            cur = db_connection_session.cursor()
            cur.execute('SELECT * FROM patient WHERE P_email = %s', (email,))
            if cur.fetchone():
                flash('Email already registered', 'error')
            else:

                cur.execute('''
                    INSERT INTO patient (
                        P_SSN, P_fname, P_mname, P_lname, P_gender, P_date_of_birth,
                        P_phone_No, P_address, blood_type, medical_history, P_email,
                        P_password
                    ) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
                ''',
                            (SSN, first_name, middle_name, last_name, gender, date_of_birth,
                             phone_number, address, blood_type, medical_history, email,
                             password))
                db_connection_session.commit()
                flash('Registration successful', 'success')

        return redirect('/registerPA')
```

Don't have an account? Register now...



_ssn varchar(14)	p_fname varchar(200)	p_lname varchar(200)	p_mname varchar(200)	p_gender varchar(14)	p_date_of_birth date
2345678912345	karim	ahmed	aly	on	2024-09-02
2345678912346	Mohamed	ahmed	aly	on	2024-08-07
6255262626261	alaa	ahmed	ami	NULL	2024-12-03
6455262626261	alia	karin	amir	F	2024-12-01
6455262626261	Carol	karim	amir	F	2024-08-07
7455262626261	Ehab	karim	amir	M	2024-08-01
7455262696261	Caroline	karim	amir	F	2024-09-04
7455262697261	kkkk	Sobhy	amir	M	2024-07-31
2222242222222	Mohamed	Ahmed	Mostafa	M	2024-10-03
4568486446516	aaaaa	ccccc	bbbbbbbbb	F	1994-02-16
3325852369526	habiba	Ahmed	Mostafa	F	2026-05-13

Reference:

- https://www.misrradiologycenter.com/