

**Project Name:** Luvia Medical Center

**With Smart AI Diagnosis Hub**

**Project Subtitle:** Hospital Management System with AI powered.

## **Table of Contents**

<b>1. Introduction</b>	<b>03</b>
• Introduction	
• Motivation	
• Objectives	
<b>2. System Technologies Used</b>	<b>04</b>
• Technologies	
• System Requirements	
• Application Structure	
<b>3. Source Codes</b>	<b>07</b>
• Admin Panel	
• User Panel	
<b>4. Output</b>	<b>17</b>
<b>5. Conclusion</b>	<b>31</b>

## 1. Introduction

---

### **Introduction:**

Luvia Medical Center is an innovative, web-based hospital management system designed to simplify healthcare operations and provide intelligent health predictions. Built on the Flask framework, it integrates machine learning models for Diabetes and Heart Disease prediction, a MySQL database for secure data storage, and Power BI dashboards for advanced data visualization. The system caters to patients, doctors, and administrators, ensuring smooth operations, better decision-making, and improved patient care. With its user-friendly interface, role-based access, and predictive healthcare features, Luvia Medical Center delivers a modern, reliable, and comprehensive solution for hospital management.

### **Motivation:**

In today's fast-paced world, people are constantly seeking smarter, faster, and more reliable solutions for managing their health and daily needs. Traditional approaches often lack real-time insights, proper data-driven decision-making, and personalized services. With the rise of Machine Learning and Data Visualization tools like Power BI, we now have the opportunity to turn raw data into meaningful predictions and insights. By combining these technologies with the simplicity and flexibility of Flask, this project is motivated to:

- Bridge the gap between raw medical data and actionable predictions.
- Provide users with easy-to-use, interactive web applications.
- Enable better healthcare decisions through predictive ML models.
- Present insights in a visually engaging way using dashboards.

This project is inspired by the vision of creating a smart, accessible, and data-driven medical platform that can positively impact people's lives.

### **Objective:**

The main objectives of this project are:

**Donor Management** – Enable easy registration, editing, and management of donor profiles.

**Disease Prediction** – Build ML-powered prediction models (Diabetes, Heart, Pregnancy, etc.) integrated with Flask.

**User-Friendly Interface** – Ensure patients and admins can easily interact through a clean, responsive design.

**Secure Data Handling** – Implement proper authentication, database management, and privacy control.

Interactive Analytics – Use Power BI and dashboards to visualize data dynamically.

**Centralized Management** – Create an all-in-one platform for doctors, donors, patients, and admins to manage healthcare efficiently.

**Scalability** – Develop the system in a way that can expand with more ML models and features in the future.

## 2. System Technologies Used

---

### Technologies:

#### 1. Programming Language

- **Python 3.10+** – Backend development and ML model integration

#### 2. Web Framework

- **Flask** – Lightweight Python web framework for building the web application

#### 3. Frontend Technologies

- **HTML5** – Structuring web pages
- **CSS3 / Bootstrap / Tailwind CSS** – Styling and responsive design
- **JavaScript** – Interactive UI components

#### 4. Database

- **SQLite / MySQL** – Storing user data, hospital records, and form submissions

## 5. Libraries & Packages

- **Pandas, NumPy** – Data processing for ML models
- **Scikit-learn** – Machine learning model implementation
- **Flask-WTF** – Form validation
- **Flask-Login** – User authentication
- **Jinja2** – HTML templating

## 6. Visualization & BI Tools

- **Power BI** – Advanced data visualization and dashboard reporting for insights

## 7. Development Tools

- **VS Code** – IDE for coding
- **Git & GitHub** – Version control and repository management

## 8. Operating System

- Windows 11 pro

### Run the Application

```
flask run
```

Open your browser at: <http://127.0.0.1:5000/>

# System Requirements

### Hardware Requirements

- **Processor:** Intel i3
- **RAM:** Minimum 4 GB (8 GB recommended)
- **Storage:** Minimum 2 GB free (5 GB recommended)
- **Display:** 1366x768 resolution or higher

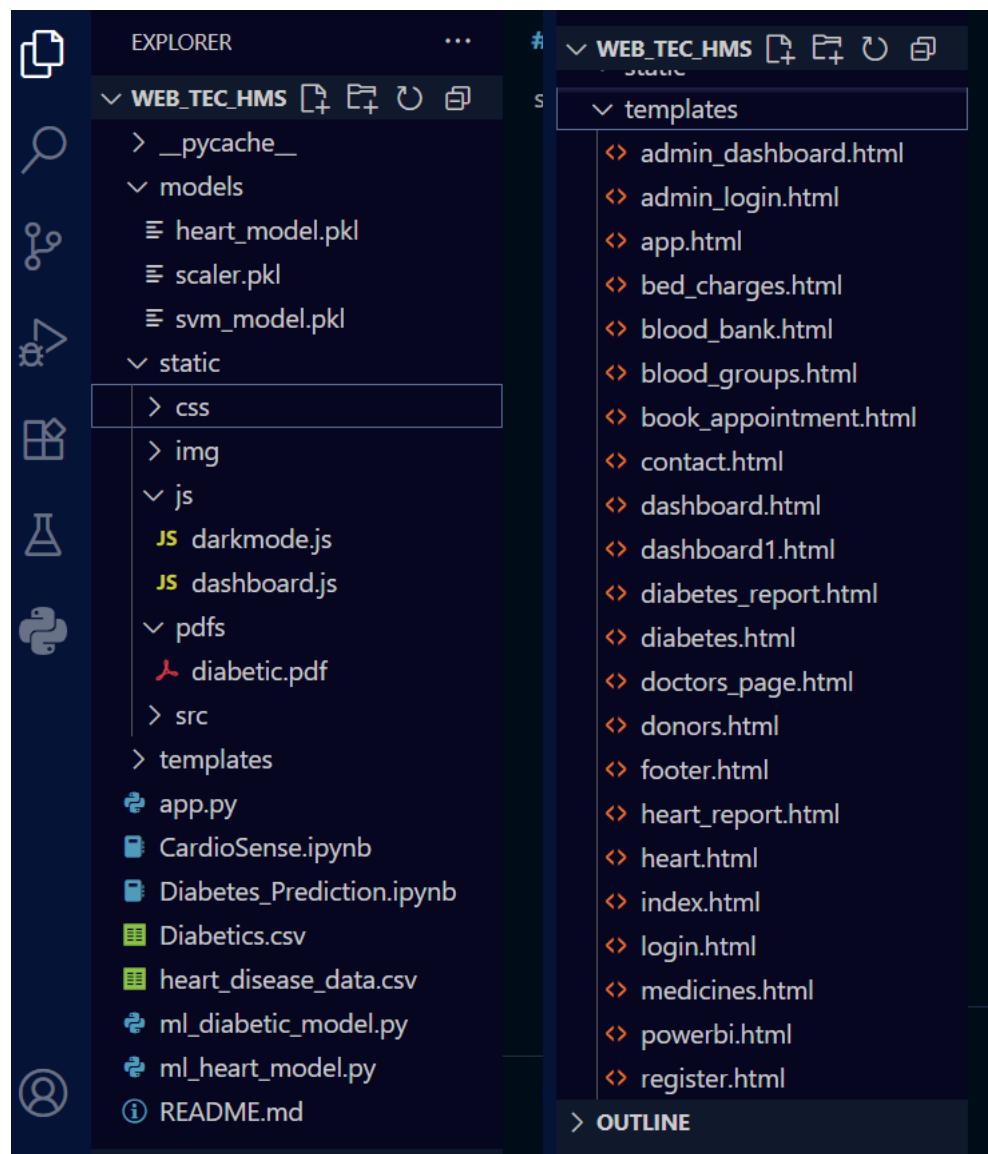
### Software Requirements

- **Operating System:** Windows 10/11

- **Python:** Version 3.8+ (3.10 recommended)
- **Database:** MySQL Server 8.0+
- **Web Browser:** Chrome, Firefox, or Edge (latest versions)
- **Libraries/Frameworks:** Flask, Pandas, NumPy, Scikit-Learn
- **Frontend:** HTML5, CSS3, JavaScript (Tailwind CSS optional)

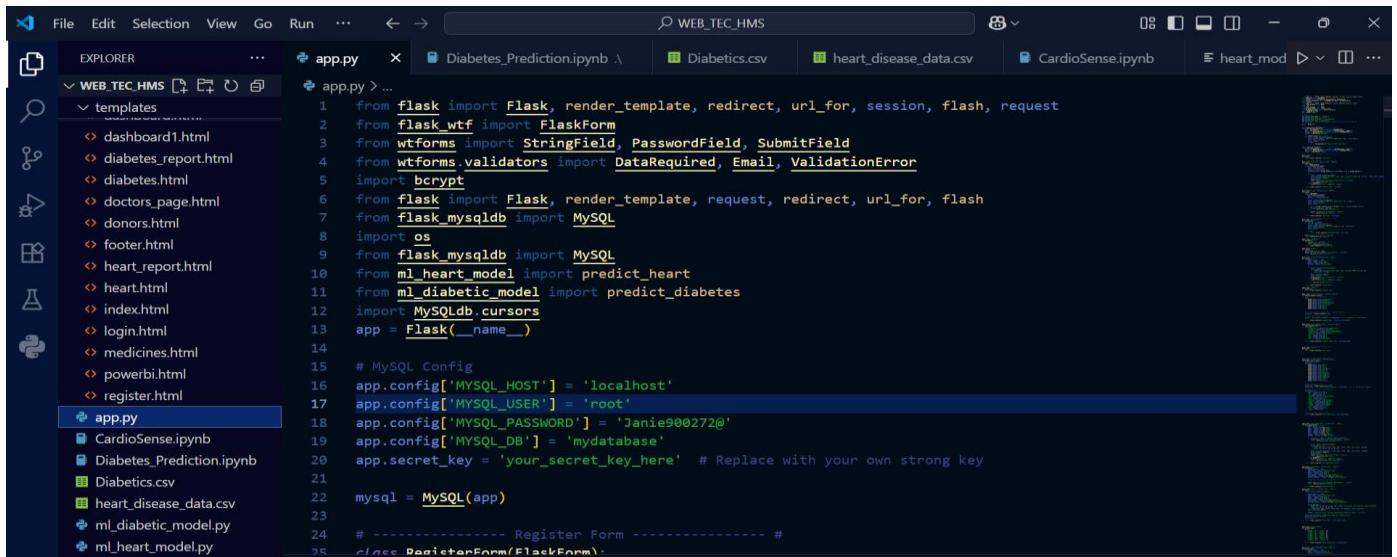
## Application Structure:

Structure of my project files:



### 3. Source Code

- App.py file:



```
File Edit Selection View Go Run ... ⌂ WEB_TEC_HMS app.py Diabetes_Prediction.ipynb Diabetics.csv heart_disease_data.csv CardioSense.ipynb ml_diabetic_model.py ml_heart_model.py EXPLORER dashboard1.html diabetes_report.html doctors.html doctors.page.html footer.html heart_report.html heart.html index.html login.html medicines.html powerbi.html register.html app.py
```

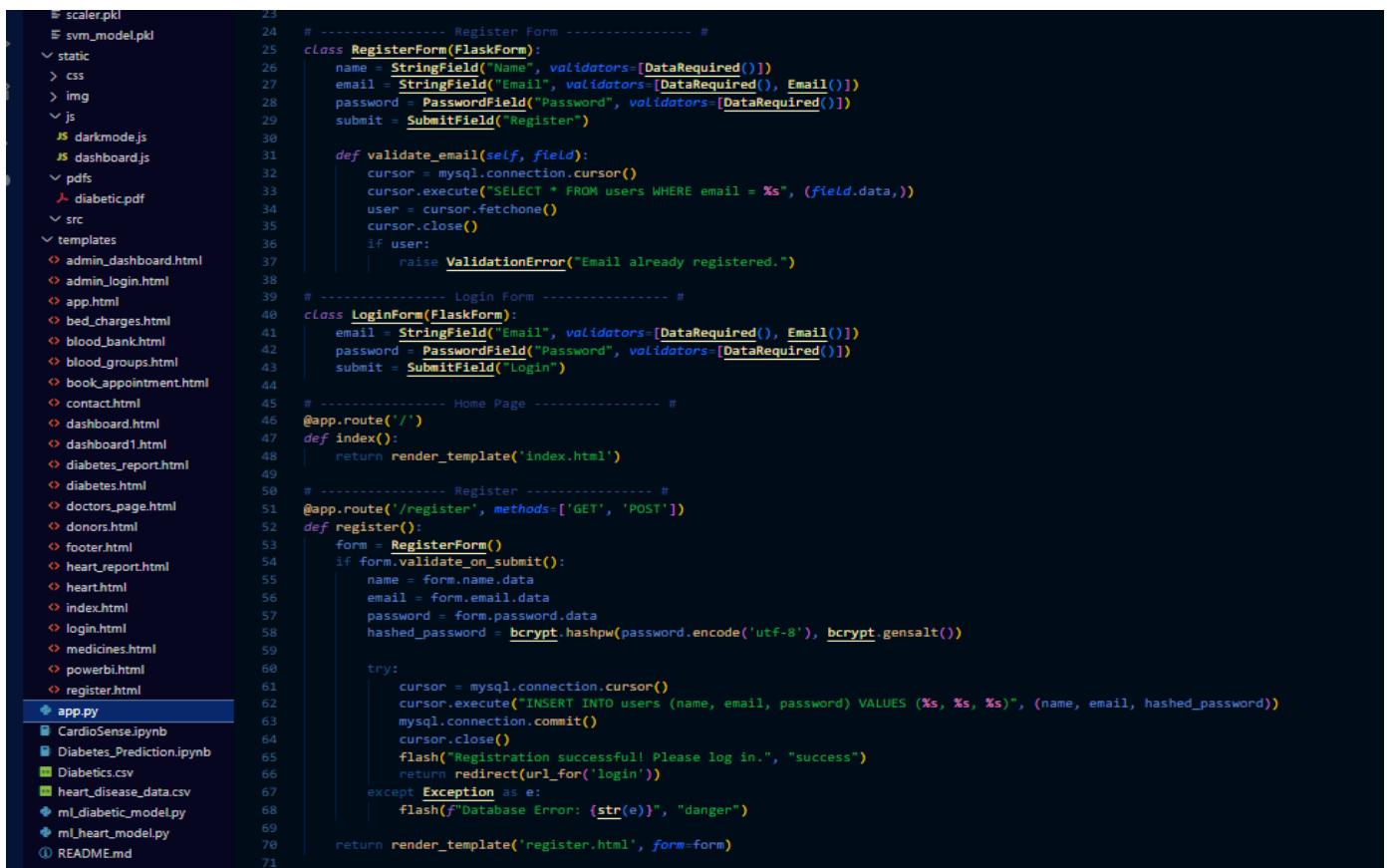
```
from flask import Flask, render_template, redirect, url_for, session, flash, request
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField
from wtforms.validators import DataRequired, Email, ValidationError
import bcrypt
from flask import Flask, render_template, request, redirect, url_for, flash
from flask_mysqldb import MySQL
import os
from flask_mysqldb import MySQL
from ml_heart_model import predict_heart
from ml_diabetic_model import predict_diabetes
import MySQLdb.cursors
app = Flask(__name__)

# MySQL Config
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = 'Janie900272@'
app.config['MYSQL_DB'] = 'mydatabase'
app.secret_key = 'your_secret_key_here' # Replace with your own strong key

mysql = MySQL(app)

# ----- Register Form -----
class RegisterForm(FlaskForm):
```

Install Libraries in Flask and add MySQL



```
scaler.pkl
svm_model.pkl
static
> css
> img
< js
  JS darkmode.js
  JS dashboard.js
pdfs
  diabetic.pdf
src
templates
  admin_dashboard.html
  admin_login.html
  app.html
  bed_charges.html
  blood_bank.html
  blood_groups.html
  book_appointment.html
  contact.html
  dashboard.html
  dashboard1.html
  diabetes_report.html
  diabetes.html
  doctors_page.html
  donors.html
  footer.html
  heart_report.html
  heart.html
  index.html
  login.html
  medicines.html
  powerbi.html
  register.html
app.py
CardioSense.ipynb
Diabetes_Prediction.ipynb
Diabetics.csv
heart_disease_data.csv
ml_diabetic_model.py
ml_heart_model.py
README.md
```

```
# ----- Register Form -----
class RegisterForm(FlaskForm):
    name = StringField("Name", validators=[DataRequired()])
    email = StringField("Email", validators=[DataRequired(), Email()])
    password = PasswordField("Password", validators=[DataRequired()])
    submit = SubmitField("Register")

    def validate_email(self, field):
        cursor = mysql.connection.cursor()
        cursor.execute("SELECT * FROM users WHERE email = %s", (field.data,))
        user = cursor.fetchone()
        cursor.close()
        if user:
            raise ValidationError("Email already registered.")

# ----- Login Form -----
class LoginForm(FlaskForm):
    email = StringField("Email", validators=[DataRequired(), Email()])
    password = PasswordField("Password", validators=[DataRequired()])
    submit = SubmitField("Login")

# ----- Home Page -----
@app.route("/")
def index():
    return render_template('index.html')

# ----- Register -----
@app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegisterForm()
    if form.validate_on_submit():
        name = form.name.data
        email = form.email.data
        password = form.password.data
        hashed_password = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())

        try:
            cursor = mysql.connection.cursor()
            cursor.execute('INSERT INTO users (name, email, password) VALUES (%s, %s, %s)', (name, email, hashed_password))
            mysql.connection.commit()
            cursor.close()
            flash("Registration successful! Please log in.", "success")
            return redirect(url_for('login'))
        except Exception as e:
            flash(f"Database Error: {str(e)}", "danger")
    return render_template('register.html', form=form)
```

Register and Login

```

    < models           146
      └── heart_model.pkl   147 @app.route('/diabetes')
      └── scaler.pkl       148     def diabetes():
      └── svm_model.pkl     149         |     return render_template('diabetes.html')
    < static          150
      > css
      > img
    < js              151
      └── darkmode.js    152     @app.route('/predict_diabetes', methods=['POST'])
      └── dashboard.js   153     def predict_diabetes_route():
    < pdfs            154         |     # Extract data from form inputs
    < diabetic.pdf     155         data = [
      < src             156             float(request.form['pregnancies']),
      < templates        157             float(request.form['glucose']),
      < admin_dashboard.html 158             float(request.form['bloodpressure']),
      < admin_login.html 159             float(request.form['skinthickness']),
      < app.html          160             float(request.form['insulin']),
      < bed_charges.html 161             float(request.form['bmi']),
      < blood_bank.html   162             float(request.form['dpf']),
      < blood_groups.html 163             float(request.form['age'])
      < book_appointment.html 164         ]
      < contact.html      165
      < dashboard.html     166     # Use the prediction function from ml_diabetic_model.py
      < dashboard1.html    167     prediction = predict_diabetes(data)
      < diabetes_report.html 168
    < templates        169     # Prepare user-friendly message
      < admin_dashboard.html 170     result = "The person is Diabetic" if prediction == 1 else "The person is Not Diabetic"
      < admin_login.html    171
      < app.html           172     return render_template('diabetes.html', prediction_text=result)
      < bed_charges.html    173
      < blood_bank.html      174
      < blood_groups.html    175
      < book_appointment.html 176
      < contact.html         177
      < dashboard.html        178
      < dashboard1.html       179
      < diabetes_report.html 180

```

## Diabetes Prediction

```

> __pycache__
< models           191
  < models           192     # 1 Route to show heart disease form
  < heart_model.pkl 193     @app.route('/heart')
  < scaler.pkl       194     def heart():
  < svm_model.pkl     195         |     return render_template('heart.html')
    < static          196     # 2 Route to show heart disease report
    < pdfs            197     @app.route('/predict_heart', methods=['POST'])
    < diabetic.pdf     198     def predict_heart_report():
    < src             199         |     # Extract form data
      < js              200         input_data = [
      < darkmode.js    201             float(request.form['age']),
      < dashboard.js   202             float(request.form['sex']),
      < diabetic.pdf    203             float(request.form['cp']),
      < src             204             float(request.form['trestbps']),
      < templates        205             float(request.form['chol']),
      < app.py          206             float(request.form['fbs']),
      < CardioSense.ipynb 207             float(request.form['restecg']),
      < Diabetes_Prediction.ipynb 208             float(request.form['thalach']),
      < Diabetes.csv     209             float(request.form['exang']),
      < heart_disease.data.csv 210             float(request.form['oldpeak']),
      < ml_diabetic_model.py 211             float(request.form['slope']),
      < ml_diabetic_model.py 212             float(request.form['ca']),
      < ml_diabetic_model.py 213             float(request.form['thal'])
      < README.md        214         ]
      < OUTLINE          215         |     # Make prediction
      < TIMELINE         216         prediction = predict_heart(input_data)
      < README.md        217         result_text = "The person has Heart Disease" if prediction == 1 else "The person is Healthy"
      < OUTLINE          218
      < TIMELINE         219         # Prepare data for the report template
      < README.md        220         report_data = {
      < OUTLINE          221             'age': request.form['age'],
      < TIMELINE         222             'sex': request.form['sex'],
      < README.md        223             'cp': request.form['cp'],
      < OUTLINE          224             'trestbps': request.form['trestbps'],
      < TIMELINE         225             'chol': request.form['chol'],
      < README.md        226             'fbs': request.form['fbs'],
      < OUTLINE          227             'restecg': request.form['restecg'],
      < TIMELINE         228             'thalach': request.form['thalach'],
      < README.md        229             'exang': request.form['exang'],
      < OUTLINE          230             'oldpeak': request.form['oldpeak'],
      < TIMELINE         231             'slope': request.form['slope'],
      < README.md        232             'ca': request.form['ca'],
      < OUTLINE          233             'thal': request.form['thal'],
      < TIMELINE         234             'result': result_text

```

## Heart Disease Prediction and report

```

244
245     @app.route('/book_appointment', methods=['GET', 'POST'])
246     def book_appointment():
247         if request.method == 'POST':
248             name = request.form.get('name')
249             email = request.form.get('email')
250             phone = request.form.get('phone')
251             date = request.form.get('date')
252             time = request.form.get('time')
253             doctor = request.form.get('doctor')
254             message = request.form.get('message')
255
256             if not (name and email and date and time and doctor):
257                 flash("Please fill in all required fields.", "danger")
258                 return redirect(url_for('book_appointment'))
259
260             try:
261                 cursor = mysql.connection.cursor()
262                 insert_query = """
263                     INSERT INTO appointments (name, email, phone, date, time, doctor, message)
264                     VALUES (%s, %s, %s, %s, %s, %s, %s)
265                 """
266
267                 cursor.execute(insert_query, (name, email, phone, date, time, doctor, message))
268                 mysql.connection.commit()
269                 cursor.close()
270                 flash("Appointment successfully booked!", "success")
271                 return redirect(url_for('Login'))
272             except Exception as e:
273                 flash(f"Database error: {str(e)}", "danger")
274                 return redirect(url_for('book_appointment'))
275
276             return render_template('book_appointment.html')
277

```

## Book Appointment to consult with Doctor

```

277
278     @app.route('/blood_bank', methods=['GET', 'POST'])
279     def blood_bank():
280         if request.method == 'POST':
281             donor_name = request.form['donor_name']
282             email = request.form['email']
283             phone = request.form['phone']
284             blood_group = request.form['blood_group']
285             last_donation = request.form.get('last_donation')
286             health_condition = request.form.get('health_condition')
287
288             # Save this info to your database here
289
290             flash('Thank you for your donation registration!', 'success')
291             return redirect(url_for('blood_bank'))
292
293         return render_template('blood_bank.html')
294
295     @app.route('/donors', methods=['GET', 'POST'])
296     def donors():
297         cursor = mysql.connection.cursor()
298
299         if request.method == 'POST':
300             donor_name = request.form['donor_name']
301             email = request.form['email']
302             phone = request.form['phone']
303             blood_group = request.form['blood_group']
304             last_donation = request.form.get('last_donation')
305             health_condition = request.form.get('health_condition')
306
307             cursor.execute("""
308                 INSERT INTO donors (donor_name, email, phone, blood_group, last_donation, health_condition)
309                 VALUES (%s, %s, %s, %s, %s, %s)
310             """, (donor_name, email, phone, blood_group, last_donation, health_condition))
311             mysql.connection.commit()
312
313             flash('Congratulations! You have successfully joined as a donor.', 'success')
314
315             cursor.execute("SELECT * FROM donors ORDER BY created_at DESC")
316             donor_list = cursor.fetchall()
317             cursor.close()
318
319             return render_template('donors.html', donors=donor_list)
320

```

## Blood Section and Add new Donor

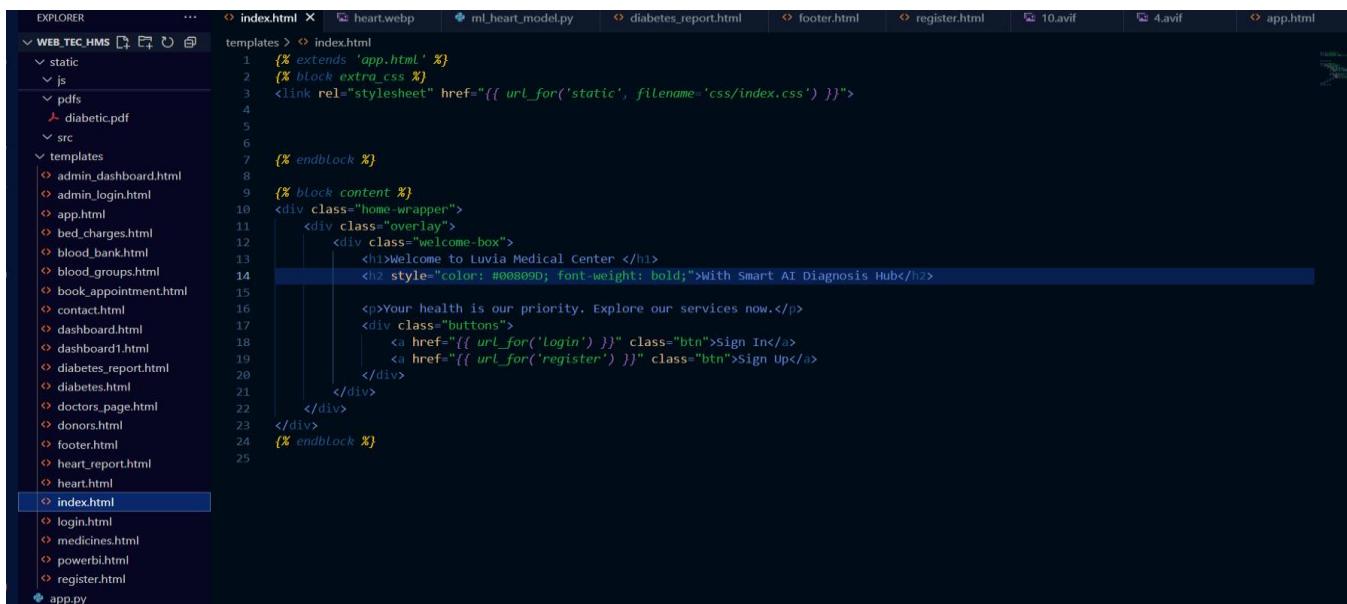
```

> img
  387 @app.route('/admin/dashboard')
  388 def admin_dashboard():
  389     if 'admin_id' not in session:
  390         flash('Please log in to access the admin dashboard.', "warning")
  391         return redirect(url_for('admin_login'))
  392
  393     cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
  394
  395     # Fetch users
  396     cursor.execute("SELECT id, name, email FROM users")
  397     users = cursor.fetchall()
  398
  399     # Fetch appointments
  400     cursor.execute("SELECT id, name, email, phone, date, time, doctor FROM appointments")
  401     appointments = cursor.fetchall()
  402
  403     # Fetch donors
  404     cursor.execute("SELECT id, donor_name, email, phone, blood_group FROM donors")
  405     donors = cursor.fetchall()
  406
  407     # Fetch admin
  408     cursor.execute("SELECT id, username, email, created_at FROM admin")
  409     admins = cursor.fetchall()
  410
  411     # Fetch contact messages
  412     cursor.execute("SELECT id, name, email, message, submitted_at FROM contact_messages")
  413     contact_messages = cursor.fetchall()
  414
  415     # Fetch doctors
  416     cursor.execute("SELECT id, name, email, phone, specialty, image FROM doctors")
  417
  418     doctors = cursor.fetchall()
  419
  420     cursor.close()
  421
  422     return render_template('admin_dashboard.html',
  423                           users=users,
  424                           appointments=appointments,
  425                           donors=donors,
  426                           admins=admins,
  427                           contact_messages=contact_messages,
  428                           doctors=doctors, # pass doctors to template
  429                           admin_email=session['admin_email'])
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451

```

## Admin Dashboard

- Admin & User Panel(html files)



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like index.html, heart.webp, ml\_heart.model.py, diabetes\_report.html, footer.html, register.html, 10.avif, 4.avif, and app.html.
- Code Editor:** Displays the content of index.html. The code includes HTML, CSS, and Jinja2 templating. A specific line of code is highlighted: `<h2 style="color: #008000; font-weight: bold;">With Smart AI Diagnosis Hub</h2>`.

```

EXPLORER      ...
index.html ✘  heart.webp  ml_heart.model.py  diabetes_report.html  footer.html  register.html  10.avif  4.avif  app.html
templates > > index.html
1  {% extends "app.html" %} 
2  {% block extra_css %} 
3    <link rel="stylesheet" href="{{ url_for('static', filename='css/index.css') }}">
4
5
6
7  {% endblock %} 
8
9  {% block content %} 
10 <div class="home-wrapper">
11   <div class="overlay">
12     <div class="welcome-box">
13       <h1>Welcome to Luvia Medical Center </h1>
14       <h2 style="color: #008000; font-weight: bold;">With Smart AI Diagnosis Hub</h2>
15
16       <p>Your health is our priority. Explore our services now.</p>
17       <div class="buttons">
18         <a href="{{ url_for('Login') }}" class="btn">Sign In</a>
19         <a href="{{ url_for('register') }}" class="btn">Sign Up</a>
20       </div>
21     </div>
22   </div>
23 </div>
24 {% endblock %} 
25

```

## Index Page

EXPLORER    ... tes\_report.html    footer.html    register.html    10.avif    4.avif    app.html    admin\_dashboard.html    d ...

WEB.TEC.HMS

- models
  - scaler.pkl
  - svm\_model.pkl
- static
  - css
  - img
- js
  - darkmode.js
  - dashboard.js
- pdfs
  - diabetic.pdf
- src
- templates
  - admin\_dashboard.html
  - admin\_login.html
  - app.html
  - bed\_charges.html
  - blood\_bank.html
  - blood\_groups.html
  - book\_appointment.html
  - contact.html
  - dashboard.html
  - dashboard1.html
  - diabetes\_report.html

OUTLINE    TIMELINE

0 ▲ 0

```

1  {% extends 'app.html' %}

2
3  {% block content %}
4    <link rel="stylesheet" href="{{ url_for('static', filename='css/admin_dashboard.css') }}">
5
6    <div class="container">
7      <!-- Sidebar -->
8      <nav class="sidebar" id="sidebar">
9        <h3>Admin Menu</h3>
10       <a onclick="showSection('admins')" id="link-admins"><span class="icon">👤</span> Admin List</a>
11       <a onclick="showSection('doctors')" id="link-doctors"><span class="icon">👤</span> Doctors</a>
12       <a onclick="showSection('appointments')" id="link-appointments"><span class="icon">📅</span> Appointment List</a>
13       <a onclick="showSection('users')" id="link-users"><span class="icon">👤</span> User List</a>
14       <a onclick="showSection('donors')" id="link-donors"><span class="icon">📌</span> Donor List</a>
15       <a onclick="showSection('contact_messages')" id="link-contact_messages"><span class="icon">💬</span> Contact Messages</a>
16       <a onclick="togglePowerBIMenu()" id="link-powerbi"><span class="icon">📊</span> Power BI</a>
17       <div id="powerbi-submenu" style="display:none; padding-left: 20px;">
18         <a onclick="loadPowerBIPDF('diabetic.pdf')">Diabetes</a>
19         <a onclick="loadPowerBIPDF('cancer.pdf')">Cancer</a>
20     </div>
21   </nav>
22   <!-- Main Content -->
23   <section class="main-content">
24     <h1>Welcome, {{ admin_email }} <span class="icon">👤</span></h1>
25     <hr>
26
27
28
29
30
31
32

```

Ln 23, Col 3   Spaces: 2   UTF-8   CRLF   {} Django HTML   Go Live

## Admin Dashboard

WEB.TEC.HMS

- static
- js
- pdfs
  - diabetic.pdf
- src
- templates
  - admin\_dashboard.html
  - admin\_login.html
  - app.html
  - bed\_charges.html
  - blood\_bank.html
  - blood\_groups.html
  - book\_appointment.html
  - contact.html
  - dashboard.html
  - dashboard1.html
  - diabetes\_report.html
  - doctors.html
  - donors.html
  - footer.html
  - heart\_report.html
  - heart.html
  - index.html
  - login.html
  - medicines.html
  - powerbi.html
  - register.html

doctors.page.html

```

1  {% extends 'app.html' %}

2
3  {% block content %}
4    <link rel="stylesheet" href="{{ url_for('static', filename='css/doctors.css') }}">
5
6    <div class="container mt-5">
7      <h1 class="mb-5 text-center" style="font-weight: 700; color: #2c3e50;">Meet Our Doctors</h1>
8
9      <div class="row">
10        {% for doc in doctors %}
11          <div class="col-md-4 mb-4">
12            <div class="card doctor-card h-100 text-center p-3 border-0 shadow-sm rounded-4">
13              {% if doc.image %}
14                
15              {% else %}
16                
17              {% endif %}
18
19              <div class="card-body">
20                <h5 class="card-title mb-2" style="font-weight:600; color:#34495e;">{{ doc.name }}</h5>
21                <p class="card-text mb-1"><i class="bi bi-envelope-fill me-2"></i>{{ doc.email }}</p>
22                <p class="card-text mb-1"><i class="bi bi-telephone-fill me-2"></i>{{ doc.phone }}</p>
23                <p class="card-text mb-0"><i class="bi bi-heart-pulse-fill me-2"></i>{{ doc.specialty }}</p>
24              </div>
25            </div>
26          {% endfor %}
27        </div>
28
29
30
31
32

```

## Doctor Page

WEB\_TEC\_HMS

```

static
js
pdfs
diabetic.pdf
src
templates
admin_dashboard.html
admin_login.html
app.html
bed_charges.html
blood_bank.html
blood_groups.html
book_appointment.html
contact.html
dashboard.html
dashboard1.html
diabetes_report.html
diabetes.html
doctors_page.html
donors.html
footer.html
heart_report.html
heart.html
index.html
login.html
medicines.html
powerbi.html
register.html
app.py

```

OUTLINE    TIMELINE

templates > blood\_bank.html

```

4   <style>
87    .advice-text h2 {
91     }
92   </style>
93   {% endblock %}
94
95   {% block content %}
96   <div class="blood-section">
97
98     <!-- Sidebar Dashboard -->
99     <div class="dashboard-sidebar">
100       
101       <div style="text-align: center; padding: 15px; color: #fff;">
102         <img alt="Dr. Umme Mezbah Akter" style="width: 170px; height: 170px; border-radius: 50%; object-fit: cover; box-shadow: 0 2px 8px rgba(0,0,0,.3); margin: 0 auto 10px auto;"/>
103         <h5 style="margin: 0; font-size: 1.1em; >Dr. Umme Mezbah Akter</h5>
104         <small style="color: #ffcd00; display: block; margin-bottom: 8px;>Specialist, Blood Bank</small>
105         <p style="margin: 0 auto; font-size: 0.9rem; max-width: 300px; line-height: 1.3em; height: 2.6em; overflow: hidden; text-overflow: ellipsis; >Dr. Umme Mezbah Akter is a highly motivated specialist in Transfusion Medicine, passionate about patient care and education.</p>
106       </div>
107     </div>
108   </div>
109   <a href="{{ url_for('blood_groups') }}>
110     <div class="dashboard-card">
111       <i class="bi bi-droplet-fill"></i>
112       <div class="stat-label" style="color: #FF6100;">Blood Groups</div>
113     </div>
114   </a>
115   <a href="{{ url_for('donors') }}>
116     <div class="dashboard-card">
117       <i class="bi bi-people-fill"></i>
118       <div class="stat-label" style="color: #FF6100;">Donors</div>
119     </div>
120   </a>
121 </div>
122 </div>
123 </div>

```

## Blood\_bank

WEB\_TEC\_HMS

```

static
models
heart_model.pkl
scaler.pkl
svm_model.pkl
src
darkmode.js
dashboard.js
pdfs
diabetic.pdf
src
templates
admin_dashboard.html
admin_login.html
app.html
bed_charges.html
blood_bank.html
blood_group.html
book_appointment.html
contact.html
dashboard.html
dashboard1.html
diabetes_report.html
diabetes.html
doctors_page.html
donors.html
footer.html
heart_report.html
heart.html
index.html
login.html
medicines.html
powerbi.html

```

OUTLINE    TIMELINE

templates > blood\_group.html

```

134
135   {% block content %}
136   <div class="blood-groups-container">
137     <h2>Available Blood Groups</h2>
138     <div class="blood-navbar">
139       {% for group in groups %}
140         <div class="blood-item" data-group="{{ group.name }}>
141           {{ group.name }}
142           <span class="liters">{{ group.liters }} L</span>
143           <button class="contact-btn" onclick="openModal('{{ group.name }}')>Contact for Blood</button>
144         </div>
145       {% endfor %}
146     </div>
147   </div>
148
149   <!-- Modal -->
150   <div id="contactModal" class="modal-overlay">
151     <div class="modal">
152       <h3>Contact for </span> Blood</h3>
153       <p>If you need blood from this group or want to donate, please contact:</p>
154       <strong>Phone:</strong> +1 (555) 123-4567<br/>
155       <strong>Email:</strong> bloodbank@example.com<br/>
156       <button class="close-btn" onclick="closeModal()>Close</button>
157     </div>
158   </div>
159
160   <script>
161     function openModal(bloodGroup) {
162       document.getElementById('modalBloodGroup').textContent = bloodGroup;
163       document.getElementById('contactModal').classList.add('active');
164     }
165
166     function closeModal() {
167       document.getElementById('contactModal').classList.remove('active');
168     }
169
170     // Optional: Close modal if clicked outside content
171     window.addEventListener('click', function(event) {
172       const modal = document.getElementById('contactModal');
173       if(event.target === modal) {
174         closeModal();
175       }
176     });
177   </script>

```

## Blood Groups

```

    v pdfs      96  {% endblock %}
    x diabetic.pdf  97
    v src       98  {% block content %}
    v templates 99
    o admin_dashboard.html 100 <div class="report-wrapper">
    o admin_login.html 101   <h2>Heart Disease Prediction Report</h2>
    o app.html     102
    o bed_charges.html 103   <ul>
    o blood_bank.html 104     <li><strong>Age:</strong> {{ data.age }}</li>
    o blood_groups.html 105     <li><strong>Sex:</strong> {{ data.sex }}</li>
    o book_appointment.html 106     <li><strong>Chest Pain Type:</strong> {{ data.cp }}</li>
    o contact.html 107     <li><strong>Resting BP:</strong> {{ data.trestbps }}</li>
    o dashboard.html 108     <li><strong>Cholesterol:</strong> {{ data.chol }}</li>
    o dashboard1.html 109     <li><strong>Fasting Blood Sugar:</strong> {{ data.fbs }}</li>
    o diabetes_report.html 110     <li><strong>Rest EG:</strong> {{ data.restecg }}</li>
    o diabetes.html 111     <li><strong>Max Heart Rate Achieved:</strong> {{ data.thalach }}</li>
    o doctors_page.html 112     <li><strong>Exercise Induced Angina:</strong> {{ data.exang }}</li>
    o donors.html 113     <li><strong>Oldpeak:</strong> {{ data.oldpeak }}</li>
    o footer.html 114     <li><strong>Slope:</strong> {{ data.slope }}</li>
    o heart_report.html 115     <li><strong>Ca:</strong> {{ data.ca }}</li>
    o heart.html 116     <li><strong>Thal:</strong> {{ data.thal }}</li>
    o indexhtml 117   </ul>
    o login.html 118
    o medicines.html 119   <h3>Result: {{ data.result }}</h3>
    o powerbi.html 120
    o register.html 121   <a href="/heart" class="btn-submit" style="margin-top:20px; display:inline-block;"> Predict Again</a>
    o static 122 </div>
    o stylesheets 123
    o template 124  {% endblock %}
    o static 125

```

## Heart Diseases Report

```

    v WEB_TEC_HMS
    > __pycache__
    v models
    f heart_model.pkl
    f scaler.pkl
    f svm_model.pkl
    v static
    > css
    > img
    v js
    f darkmode.js
    f dashboard.js
    v pdfs
    x diabetic.pdf
    v src
    v templates
    o admin_dashboard.html
    o admin_login.html
    o app.html
    o bed_charges.html 21  {% extends 'app.html' %} 22  {% block extra_css %} 23
    o blood_bank.html 24
    o blood_groups.html 25
    o book_appointment.html 26
    o contact.html 27
    o dashboard.html 28
    o dashboard1.html 29
    o diabetes_report.html 30
    o diabetes.html 31
    o doctors_page.html 32
    o donors.html 33
    o footer.html 34
    o heart_report.html 35
    o heart.html 36
    o indexhtml 37
    o login.html 38
    o medicines.html 39
    o powerbi.html 40
    > OUTLINE 41
    > TIMELINE 42
    0 43
    0 44

```

template > o bed\_charges.html

```

1  {% extends 'app.html' %} 2  {% block extra_css %} 3
4  <link rel="stylesheet" href="{{ url_for('static', filename='css/bed_charges.css') }}">
5
6  {% endblock %}
7  {% block content %}
8
9
10
11 <!-- bed charges.html -->
12 <div class="container my-5">
13   <h2 class="text-center mb-4" style="font-weight:700; color:white;">
14     Bed & Cabin Charges
15   </h2>
16   <p class="text-center mb-4" style="font-size: 1rem; color: white;">
17     Comprehensive list of all hospital room types, availability, daily charges, and amenities.
18 </p>
19
20 <div class="card shadow-lg p-4 border-0 rounded-4" style="background: #F9F9F9;">
21   <table class="table table-hover table-bordered align-middle text-center" style="border-radius:10px; overflow:hidden;">
22     <thead class="table-primary text-white">
23       <tr>
24         <th> Room Type</th>
25         <th> Quantity Available</th>
26         <th> Price per Day</th>
27         <th> Facilities</th>
28         <th> Notes</th>
29       </tr>
30     </thead>
31     <tbody>
32       <tr>
33         <td>General Ward</td>
34         <td>30 Beds</td>
35         <td>$ 500</td>
36         <td>Shared Room, Basic Amenities</td>
37         <td>Suitable for general patients</td>
38       </tr>
39       <tr>
40         <td>Cabin (Non-AC)</td>
41         <td>15 Rooms</td>
42         <td>$ 1,200</td>
43         <td>Private Room, TV, Fan</td>
44       </tr>

```

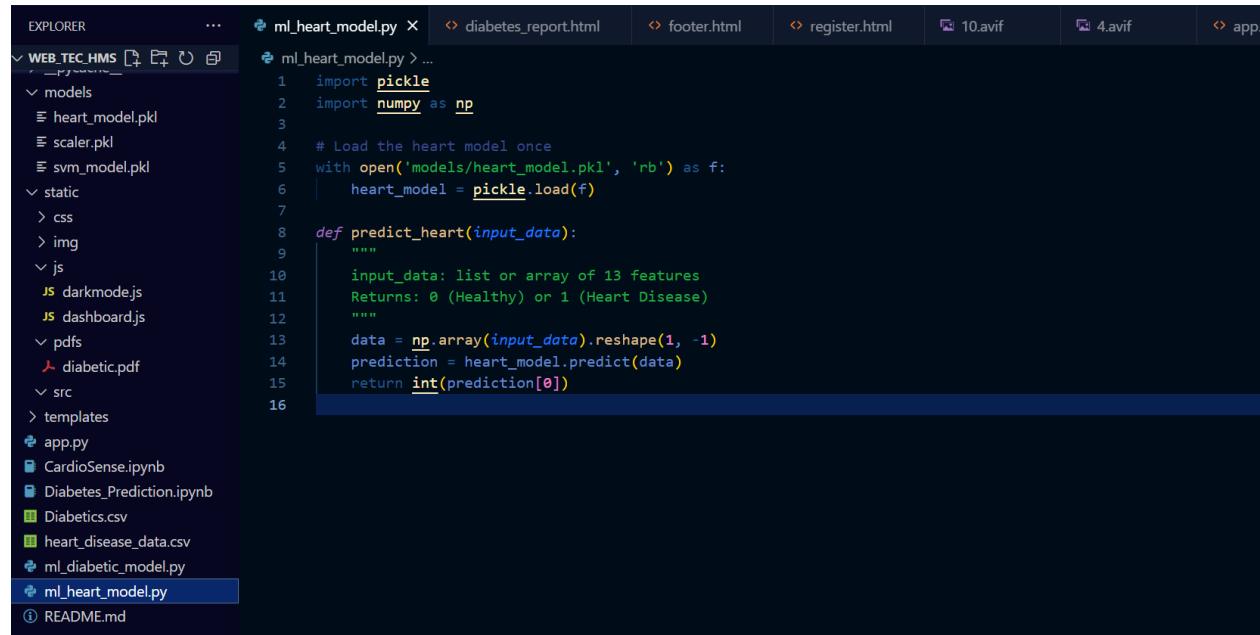
## Bed Info

```
static > JS darkmodejs > ...
1 // static/js/darkmode.js
2 document.addEventListener('DOMContentLoaded', () => {
3   const toggleBtn = document.getElementById('darkModeToggle');
4   if (!toggleBtn) return;
5
6   toggleBtn.addEventListener('click', () => {
7     document.body.classList.toggle('dark-mode');
8
9     // Toggle icon between sun/moon
10    const icon = toggleBtn.querySelector('i');
11    if (document.body.classList.contains('dark-mode')) {
12      icon.classList.replace('bi-moon-fill', 'bi-sun-fill');
13      localStorage.setItem('theme', 'dark');
14    } else {
15      icon.classList.replace('bi-sun-fill', 'bi-moon-fill');
16      localStorage.setItem('theme', 'light');
17    }
18  });
19
20 // Persist theme on reload
21 if (localStorage.getItem('theme') === 'dark') {
22   document.body.classList.add('dark-mode');
23   const icon = toggleBtn.querySelector('i');
24   if (icon) icon.classList.replace('bi-moon-fill', 'bi-sun-fill');
25 }
26 });
27
```

Darkmode.js file

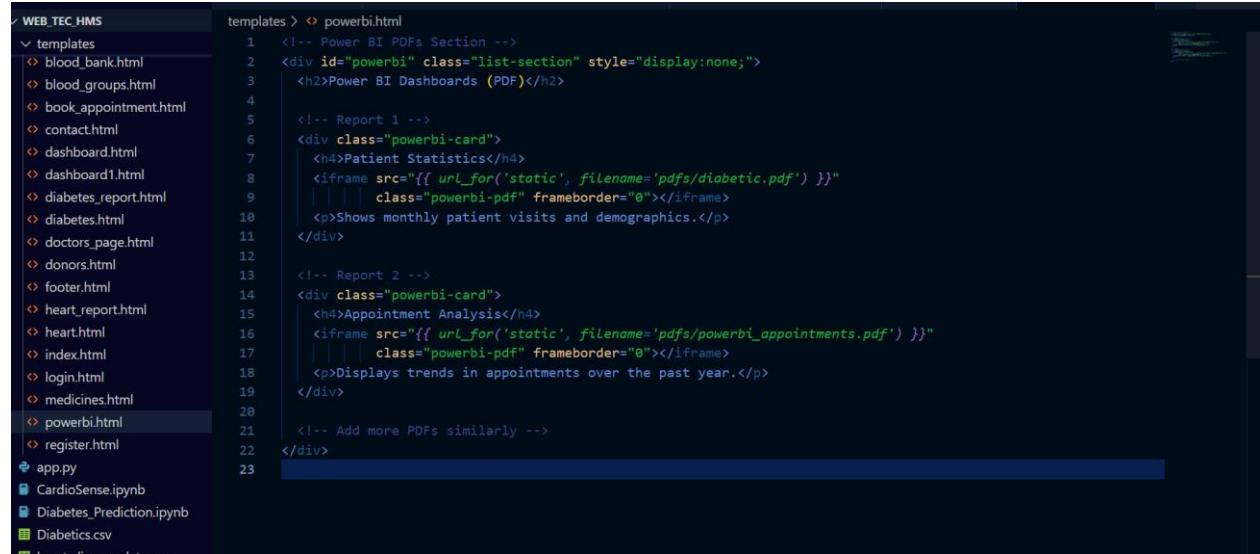
```
ml_diabetic_model.py > ...
1 import pickle
2 import numpy as np
3
4 # Load the diabetes model
5 with open('models/svm_model.pkl', 'rb') as f:
6     model = pickle.load(f)
7
8 with open('models/scaler.pkl', 'rb') as f:
9     scaler = pickle.load(f)
10
11 def predict_diabetes(input_data):
12     data = np.array(input_data).reshape(1, -1)
13     scaled_data = scaler.transform(data)
14     prediction = model.predict(scaled_data)
15     return prediction[0]
```

ML Diabetic Model.py



```
1 import pickle
2 import numpy as np
3
4 # Load the heart model once
5 with open('models/heart_model.pkl', 'rb') as f:
6     heart_model = pickle.load(f)
7
8 def predict_heart(input_data):
9     """
10         input_data: list or array of 13 features
11         Returns: 0 (Healthy) or 1 (Heart Disease)
12     """
13     data = np.array(input_data).reshape(1, -1)
14     prediction = heart_model.predict(data)
15     return int(prediction[0])
16
```

## ML Heart Model.py



```
1 <!-- Power BI PDFs Section -->
2 <div id="powerbi" class="list-section" style="display:none;">
3     <h2>Power BI Dashboards (PDF)</h2>
4
5     <!-- Report 1 -->
6     <div class="powerbi-card">
7         <h4>Patient Statistics</h4>
8         <iframe src="{{ url_for('static', filename='pdfs/diabetic.pdf') }}">
9             <!-- Power BI PDF -->
10        <p>Shows monthly patient visits and demographics.</p>
11    </div>
12
13    <!-- Report 2 -->
14    <div class="powerbi-card">
15        <h4>Appointment Analysis</h4>
16        <iframe src="{{ url_for('static', filename='pdfs/powerbi_appointments.pdf') }}">
17            <!-- Power BI PDF -->
18        <p>Displays trends in appointments over the past year.</p>
19    </div>
20
21    <!-- Add more PDFs similarly -->
22 </div>
23
```

## Power Bi file

- Prediction Model

The screenshot shows a Jupyter Notebook interface with a sidebar containing files like `CardioSense.ipynb`, `Diabetes_Prediction.ipynb`, and `heart_disease_data.csv`. The main code cell contains Python code for loading a model and making predictions:

```

# input random data for testing
input_data = [62,0,0,140,268,0,0,160,0,3,6,0,2,2] # Converted to a list
# Get the feature names from the training data
feature_names = X_train.columns
input_df = pd.DataFrame([input_data], columns=feature_names) # Create a DataFrame with feature names

prediction_model.predict(input_df) # Use the DataFrame for prediction
print(prediction)

if(prediction[0]==0):
    print('The person is Healthy')
else:
    print('The Person is Sick')

...
[0]
The person is Healthy

```

Below the code cell, a message says "heart\_model.pkl saved".

## Heart Diseases

The screenshot shows a Jupyter Notebook interface with a sidebar containing files like `CardioSense.ipynb`, `Diabetes_Prediction.ipynb`, and `heart_disease_data.csv`. The main code cell contains Python code for loading a model and making predictions:

```

input_data = (1 ,85, 66, 29, 0 ,26.6, 0.351, 31 )
# convert in np array
check_data = np.asarray(input_data)
#reshape
check_data.reshape=check_data.reshape(1, 1)
# standardize the input data
std_data=loaded_scaler.transform(check_data.reshape)
print(std_data)
prediction=loaded_model.predict(std_data)
print(prediction)
if(prediction[0]==0):
    print("The person is not diabetic")
else:
    print("The person is diabetic")

```

Below the code cell, a message says "heart\_model.pkl saved".

Another code cell below it contains:

```

from google.colab import files
# 7. Save model to 'svm_model.pkl'
with open('svm_model.pkl', 'wb') as f:
    pickle.dump(classifier, f)

# 8. Save scaler to 'scaler.pkl'
with open('scaler.pkl', 'wb') as f:
    pickle.dump(scaler, f)

# 9. Download both files
files.download('svm_model.pkl')
files.download('scaler.pkl')

```

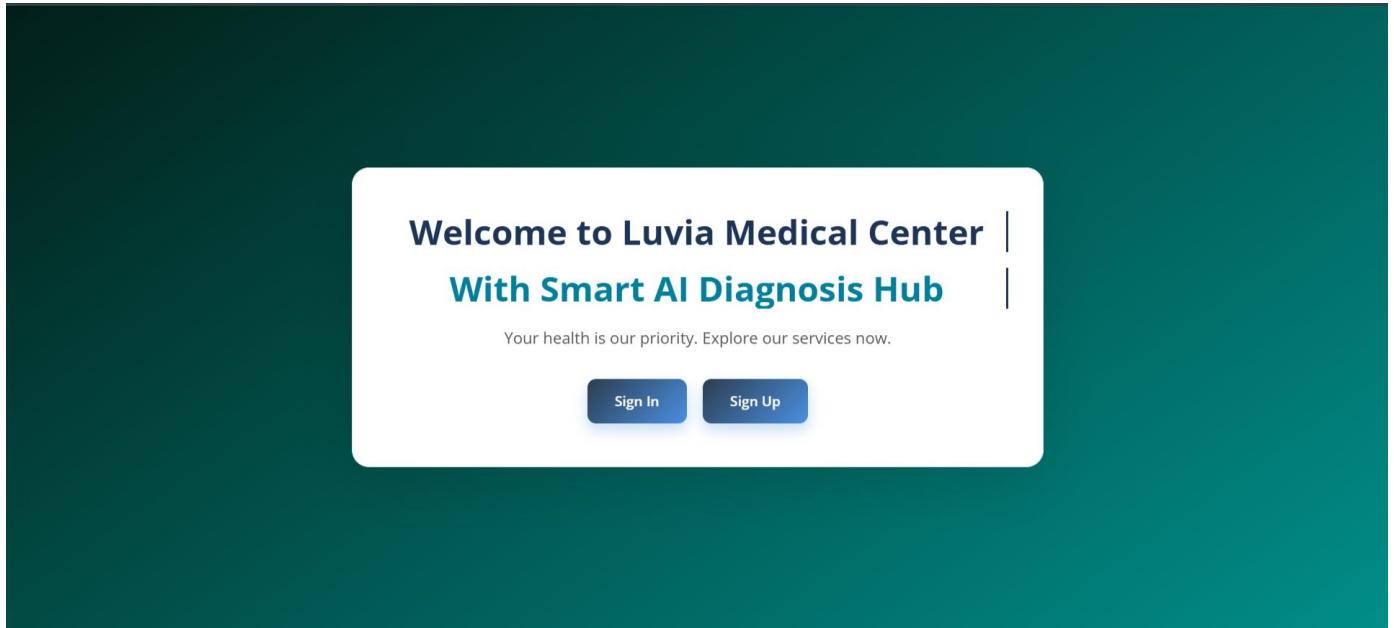
Below this, a message says "heart\_model.pkl saved".

## Diabetic

## 4. Output

---

- User Interface



- User Register Form

The left side shows a "Register form" interface with three input fields for Name, Email, and Password, followed by a "Register" button and a "Login page" link. The right side shows a photograph of several medical professionals in scrubs attending to a patient in an intensive care unit.

- User Login Form



**Login**

Email

Password

⌚ Opening Time: 9:00 AM | ⚓ Closing Time: 6:00 PM

 Luvia Medical Center Appointments Doctors

**Welcome to the Dashboard**

**User Information**

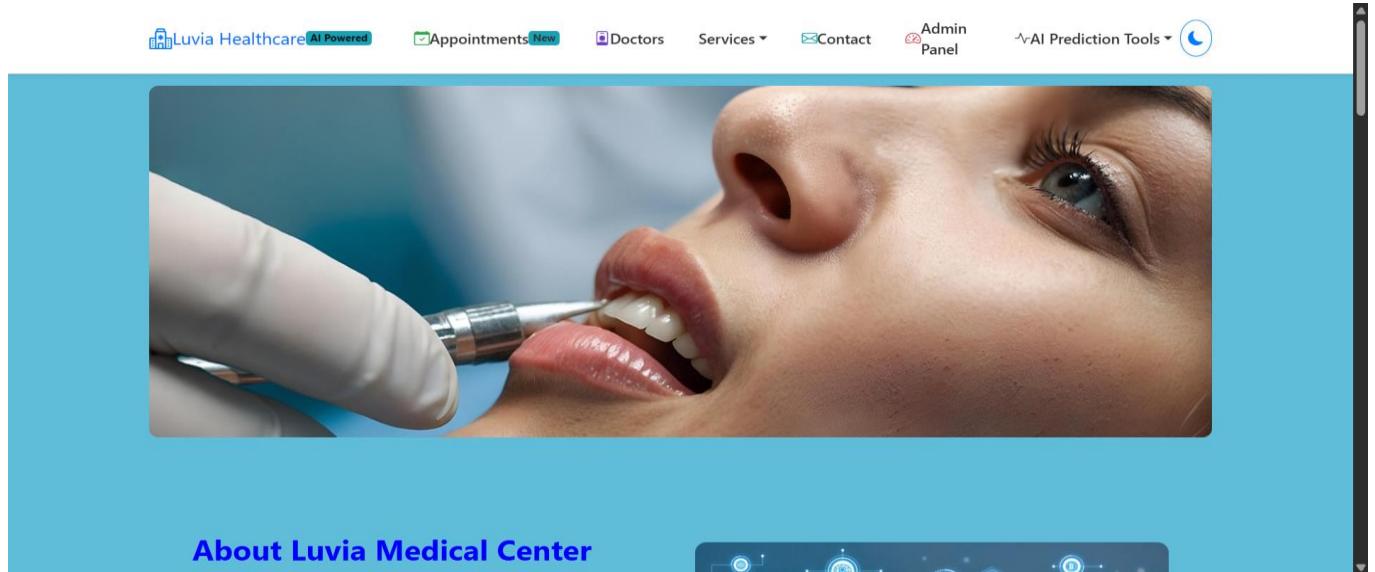
Name: reja

Email: reja@gmail.com

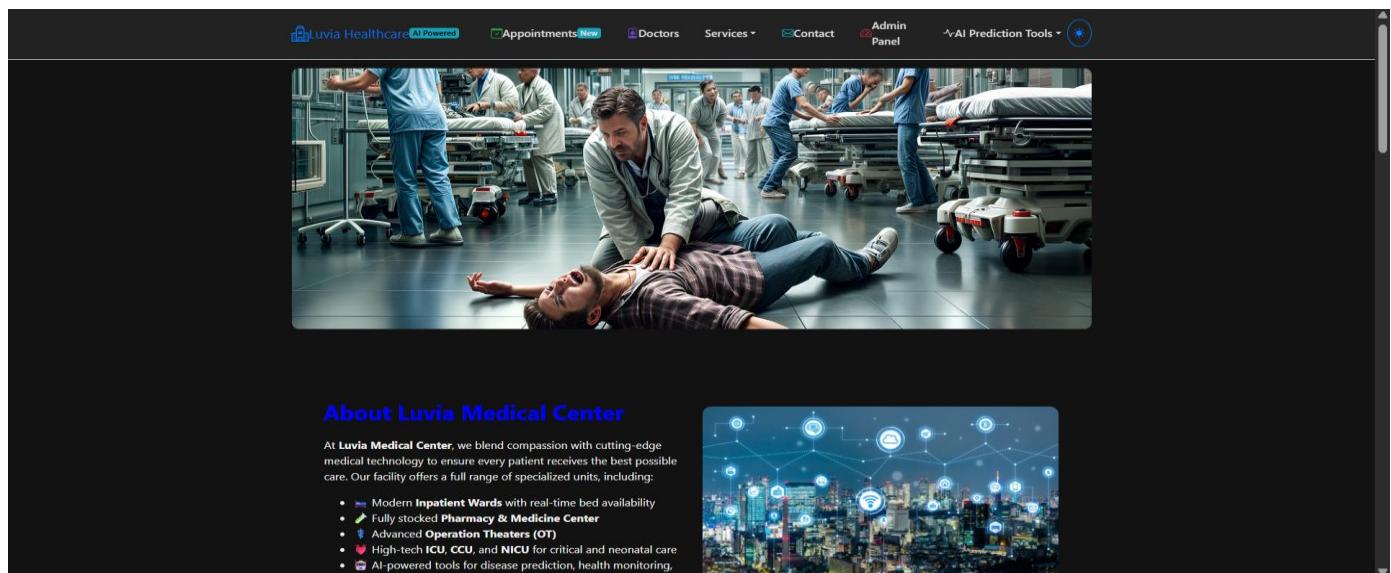
[Logout](#) [Visit Dashboard](#)

127.0.0.1:5000/dashboard

- Dashboard



The screenshot shows a medical dashboard for Luvia Healthcare. At the top, there's a navigation bar with links for Appointments (New), Doctors, Services, Contact, Admin Panel, and AI Prediction Tools. Below the navigation is a large image of a dental procedure where a patient's teeth are being treated. A blue banner at the bottom of the screen reads "About Luvia Medical Center".



The screenshot shows a medical dashboard for Luvia Healthcare. At the top, there's a navigation bar with links for Appointments (New), Doctors, Services, Contact, Admin Panel, and AI Prediction Tools. Below the navigation is a large image of a medical emergency in an operating room where a patient is lying on the floor and a doctor is attending to them. A blue banner at the bottom of the screen reads "About Luvia Medical Center". To the right of the banner is a graphic of a city skyline with various medical icons (like a heart, a brain, a stethoscope) overlaid, representing connectivity and technology.

## About Luvia Medical Center

At Luvia Medical Center, we blend compassion with cutting-edge medical technology to ensure every patient receives the best possible care. Our facility offers a full range of specialized units, including:

- ➡ Modern Inpatient Wards with real-time bed availability
- ➡ Fully stocked Pharmacy & Medicine Center
- ➡ Advanced Operation Theaters (OT)
- ➡ High-tech ICU, CCU, and NICU for critical and neonatal care
- ➡ AI-powered tools for disease prediction, health monitoring, and diagnostics



## Our Facilities

We constantly invest in high-end equipment



- ➡ 24/7 Emergency & ICU
- ➡ Modern Diagnostic Lab
- ➡ Experienced Doctors & Nurses
- ➡ Free Health Checkup Camps
- ➡ Online Appointment & Report

## 💡 AI-Powered Disease Prediction

Use our advanced machine learning models to assess your health risk and get preventive guidance instantly.

### Heart Disease

Check your heart risk with just a few inputs.

[Predict Now](#)

### Diabetes

Get personalized diabetes risk assessment.

[Predict Now](#)

### Cancer

Early detection predictions using ML models.

[Predict Now](#)

### COVID-19

Predict your COVID-19 risk based on symptoms.

[Predict Now](#)

## ⚠️ Emergency Care

Our emergency department is open 24/7 to provide critical care and life-saving support.

### 24/7 ER

Fully staffed emergency room ready for any critical situation.

### Trauma & Critical Care

Advanced trauma and critical care units for immediate treatment.

### Ambulance Service

Rapid response ambulance & mobile ICU for emergencies.

[📞 Call Emergency Now](#)

- Book an Appointment

- Blood Bank

**Dr. Umme Mezbah Akter**  
MBBS, MD (Transfusion Medicine)  
*Specialist, Blood Bank*

Dr. Umme Mezbah Akter is a highly motivated specialist in Transfusion

滴 Blood Groups

人 Donors

"One donation can save up to three lives — your blood is precious, share it."

**Blood Donation Saves Lives**

Donating blood is a simple act of kindness that can make a big difference. Your donation helps patients in emergencies, surgeries, and chronic illnesses. Be a hero — donate blood regularly and help save lives in your community.



## 滴 Donor List

[+ Add Donor](#)

**Farhana Anie (B+)**  
Email: mimi12@gmail.com  
Phone: 01811685234  
Last Donation: 2025-07-28

**Richmond Antor Biswas (B+)**  
Email: rich@gmail.com  
Phone: 01811685119  
Last Donation: 2025-06-02

**Istiak Mamun (A-)**  
Email: mamun@gmail.com  
Phone: 01811685234  
Last Donation: 2024-01-30

**Tasnim Tushi (B+)**  
Email: tushi@gmail.com  
Phone: 01811685143  
Last Donation: 2025-04-27

**Nikolas Khan (AB-)**  
Email: khan@gmail.com  
Phone: 01811685119  
Last Donation: 2025-06-30

## Donor List

[+ Add Donor](#)

**Donor Name**

**Email**

**Phone**

**Blood Group**

**Last Donation Date**  
 Select

**Health Condition**

**Submit**

**Farhana Anie (B+)**  
 Email: mimi12@gmail.com  
 Phone: 01811685234  
 Last Donation: 2025-07-28

**Richmond Antor Biswas (B+)**  
 Email: rich@gmail.com  
 Phone: 01811685119  
 Last Donation: 2025-06-02

**Istiak Mamun (A-)**  
 Email: mamun@gmail.com  
 Phone: 01811685234  
 Last Donation: 2024-01-30

**Tasnim Tushi (B+)**  
 Email: tasnimtushi@gmail.com  
 Phone: 01811685234  
 Last Donation: 2025-07-28

**Nikolas Khan (AB-)**  
 Email: nikolas.khan@gmail.com  
 Phone: 01811685234  
 Last Donation: 2024-01-30

- Bed Information

### 🛏️ Bed & Cabin Charges

Comprehensive list of all hospital room types, availability, daily charges, and amenities.

🛏️ Room Type	🛌 Quantity Available	₹ Price per Day	⭐ Facilities	📝 Notes
General Ward	30 Beds	₹ 500	Shared Room, Basic Amenities	Suitable for general patients
Cabin (Non-AC)	15 Rooms	₹ 1,200	Private Room, TV, Fan	Good for semi-private stay
Cabin (AC)	20 Rooms	₹ 2,000	Private Room, AC, TV, Attached Bathroom	Comfortable for long-term stay
ICU	10 Beds	₹ 5,000	Advanced Monitoring, Oxygen, Ventilator	Critical care patients only
CCU	8 Beds	₹ 6,000	Cardiac Monitoring, Specialist Care	For cardiac patients
VIP Suite	5 Rooms	₹ 10,000	Private Suite, AC, TV, Mini Fridge, Attached Bathroom	Luxury stay for VIP patients

⚠️ All charges are per day. Taxes and additional services may apply. For reservations and inquiries, please contact hospital administration.

- Contact Us

## Contact Us

We'd love to hear from you

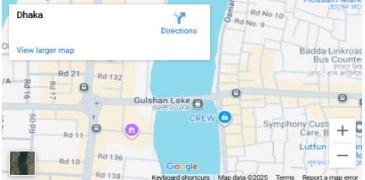


Your Name

Your Email

Your Message

**Send Message**



**Reach Us At**  
Email: support@example.com  
Phone: +880-123-456-7890  
Address: 123 Gulshan Avenue, Dhaka, Bangladesh

- Admin Login Form

## Admin Login

**Email Address**

**Password**

**Login**

- Admin Dashboard

**Welcome, janie@example.com 🙌**

### Admin List

ID	Username	Email	Created At
1	janie	janie@example.com	2025-08-12 01:57:53

**Admin Menu**

- [Admin List](#)
- [Doctors](#)
- [Appointment List](#)
- [User List](#)
- [Donor List](#)
- [Contact List](#)
- [Power BI](#)
- [Logout](#)

**Welcome, janie@example.com 🙌**

### Doctors List

Add Doctor					
ID	Name	Email	Phone	Specialty	Photo
1	Mr.hasan	hasan@gmail.com	01811685143	Medicine	
2	Mridul Datta	datta@gmail.com	01834567234	ENT	

**Admin Menu**

- [Admin List](#)
- [Doctors](#)
- [Appointment List](#)
- [User List](#)
- [Donor List](#)
- [Contact List](#)
- [Power BI](#)
- [Logout](#)

Welcome, janie@example.com 🙌

### Doctors List

Add Doctor

Name:

Email:

Phone:

Specialty:

Photo:  No file chosen

Add Doctor

ID	Name	Email	Phone	Specialty	Photo
1	Mr.hasan	hasan@gmail.com	01811685143	Medicine	

Welcome, janie@example.com 🙌

### Appointment List

ID	Name	Email	Phone	Date	Time	Doctor
1	Faria Jahan Janie	janiefaria315@gmail.com	01811685119	2025-08-07	15:04:00	Dr. Ahmed
2	reja	reja@gmail.com	01811685119	2025-07-29	4:22:00	Dr. Rahman
3	jannat hasan	hasan@gmail.com	01834567234	2025-08-30	19:36:00	Dr. Rahman

**Welcome, janie@example.com 🙌**

### User List

ID	Name	Email
3	Antor	antor@gmail.com
15	janie	janie@gmail.com
16	fdh	janiefaria31hghhhhh5@gmail.com
17	gias	gias4@gmail.com
18	er	a@gmail.com
19	israt	israt@gmail.com
21	reja	reja@gmail.com

**Welcome, janie@example.com 🙌**

### Donor List

ID	Name	Email	Phone	Blood Group
1	Nikolas Khan	khan@gmail.com	01811685119	AB-
2	Tasnim Tushi	tushi@gmail.com	01811685143	B+
3	Istiak Mamun	mamun@gmail.com	01811685234	A-
4	Richmond Antor Biswas	rich@gmail.com	01811685119	B+
5	Farhana Anie	mimi12@gmail.com	01811685234	B+

**Welcome, janie@example.com 🙌**

### Contact List

ID	Name	Email	Message	Submitted At
1	reja	reja@gmail.com	hfqj	2025-08-04 22:33:30
2	mimi	mimi12@gmail.com	I need info about bloodbank	2025-08-13 22:41:45

- Predict Diabetic:

**Diabetes Prediction Form**

Pregnancies e.g. 2	Glucose e.g. 120
Blood Pressure e.g. 70	Skin Thickness e.g. 20
Insulin e.g. 100	BMI e.g. 24.5
Diabetes Pedigree Function e.g. 0.58	Age e.g. 35

**Predict**

**Show Report**

**Diabetes Prediction Form**

Pregnancies e.g. 2	Glucose e.g. 120
Blood Pressure e.g. 70	Skin Thickness e.g. 20
Insulin e.g. 100	BMI e.g. 24.5
Diabetes Pedigree Function e.g. 0.58	Age e.g. 35

**Predict**

The person is Not Diabetic

**Prediction Report**

Pregnancies:	2
Glucose:	140
Blood Pressure:	80
Skin Thickness:	30
Insulin:	100
BMI:	25.9
Diabetes Pedigree Function:	0.55
Age:	40

**Result:** The person is Not Diabetic

- Predict Heart Diseases:



### Heart Disease Prediction Form

Age e.g. 55	Sex (1=Male, 0=Female) e.g. 1	Chest Pain Type (0-3) e.g. 2
Resting BP e.g. 130	Cholesterol e.g. 250	Fasting Blood Sugar >120 mg/dl (1=yes, 0=no) e.g. 0
Rest ECG (0-2) e.g. 1	Max Heart Rate Achieved e.g. 150	Exercise Induced Angina (1=yes, 0=no) e.g. 0
Oldpeak e.g. 1.5	Slope (0-2) e.g. 1	Ca (0-4) e.g. 0
Thal (1=Normal,2=Fixed defect,3=Reversible defect) e.g. 2		
<b>Predict</b>		

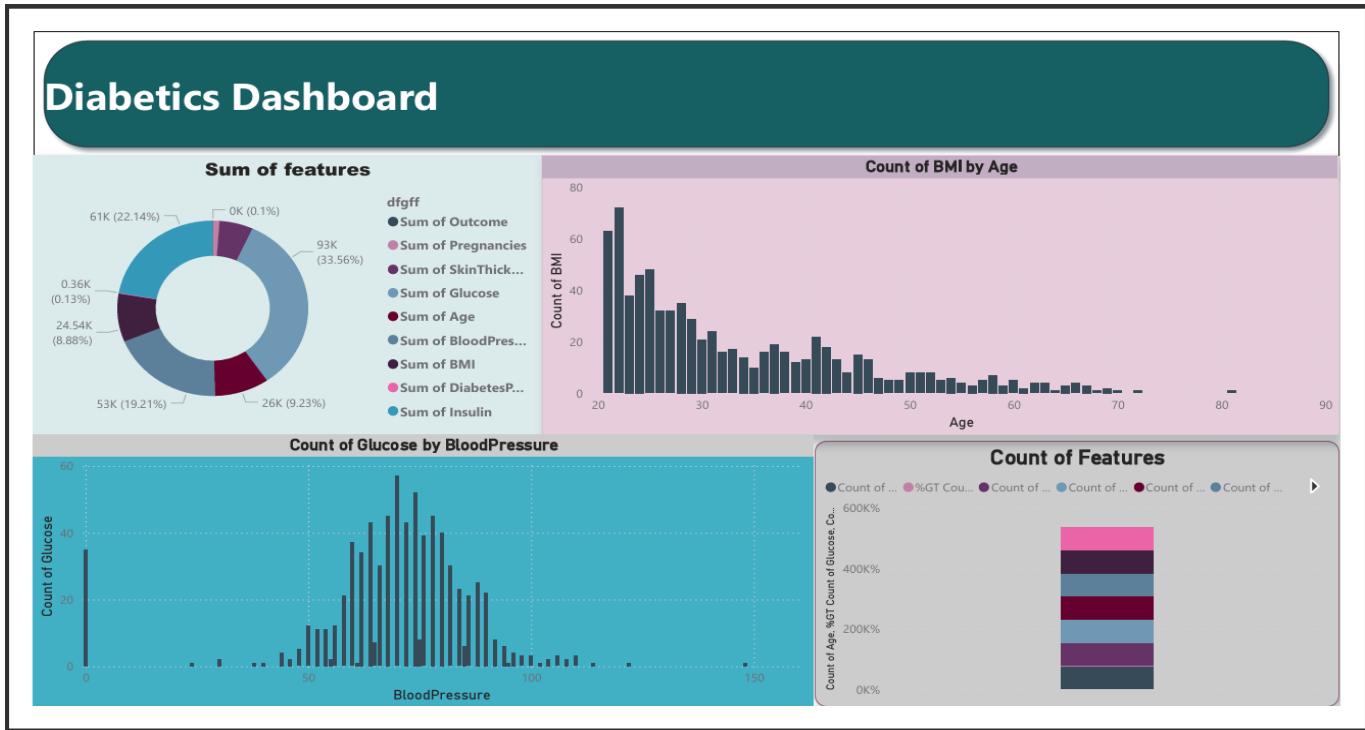
### Heart Disease Prediction Report

Age:	60
Sex:	1
Chest Pain Type:	2
Resting BP:	150
Cholesterol:	300
Fasting Blood Sugar:	1
Rest ECG:	2
Max Heart Rate Achieved:	120
Exercise Induced Angina:	0
Oldpeak:	1.9
Slope:	2
Ca:	4
Thal:	3

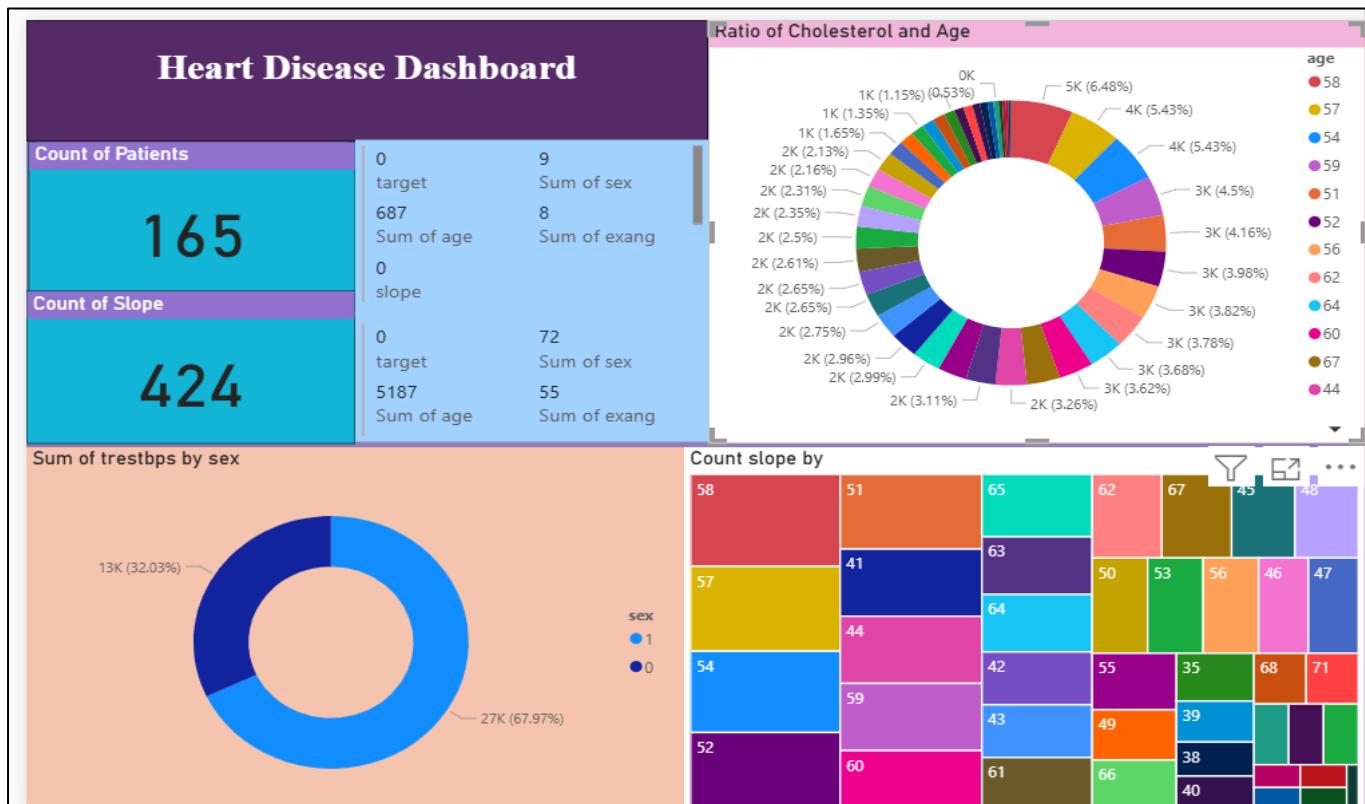
**Result: The person is Healthy**

**Predict Again**

- PowerBI :



Diabetic Dashboard



Heart Disease Dashboard

## **5. Conclusion**

---

This project successfully demonstrates the use of Flask to build a complete hospital management and prediction system. It integrates different technologies such as Python, Flask, MySQL, and machine learning to provide reliable healthcare-related services in one platform. The system includes features like patient management, doctor information, health tips, bed availability, and prediction models for diseases such as diabetes, cancer, and pregnancy. By combining data visualization and machine learning, the project helps in offering quick and effective decision support for both patients and hospital staff. The use of Flask makes the application lightweight, fast, and easy to maintain, while the database ensures secure and organized storage of information. This hospital system not only provides medical predictions but also focuses on improving patient care by offering emergency care options, doctor lists, and contact facilities. Overall, the project highlights how technology can be applied in the healthcare sector to make hospital services smarter, more accessible, and more efficient. It also creates scope for future improvements like adding more advanced prediction models, real-time data updates, and expanding the system into a fully digital healthcare solution.