Title	Sprint 2
Team ID	PNT2022TMID29827
Project Name	Plasma Donor Application

Base.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.2/css/all.min.css"/>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

    <title>Document</title>

    {% block head %}

    {% endblock %}
</head>
<body>
    {% block body %}

    {% endblock %}
</body>
</html>
```

Index.html

```
{% extends 'base.html' %}

{% block head %}
```

```
  <link rel="stylesheet" href="{{ url_for('static', filename= 'css/style.css')
}}">
  <link rel="stylesheet" href="{{ url_for('static', filename= 'css/FAQ.css') }}">
{% endblock %}

{% block body %}
<!-- Move to up button -->
  <div style="background-color: red;color: aliceblue;">{{msg}}</div>
  <div class="scroll-button">
    <a href="#home"><i class="fas fa-arrow-up"></i></a>
  </div>
  <!-- navgaition menu -->
  <nav>
    <div class="navbar">
      <div class="logo"><a href="#">Plasma Bank</a></div>
      <ul class="menu">
          <li><a href="#home">Home</a></li>
          <li><a href="#about">About</a></li>
          <li><a href="#product">Categories</a></li>
          <li><a href="#faq">FAQ</a></li>
          <li><a href="#contact">Contact</a></li>
          <li><a href="{{ url_for('signup') }}">Signup</a>
          </li>
          <li><a href="{{ url_for('login') }}">Login</a>
          </li>

          <div class="cancel-btn">
            <i class="fas fa-times"></i>
          </div>
      </ul>
      <!-- <div class="media-icons">
        <a href="#"><i class="fab fa-facebook-f"></i></a>
        <a href="#"><i class="fab fa-twitter"></i></a>
        <a href="#"><i class="fab fa-instagram"></i></a>
      </div> -->
    </div>
    <div class="menu-btn">
      <i class="fas fa-bars"></i>
    </div>
  </nav>

<!-- Home Section Start -->
 <section class="home" id="home">
   <div class="home-content">
     <div class="text">
```

```
        <div class="text-one">Welcome,</div>
        <div class="text-two">Plasma Bank</div>
        <div class="text-three">Donate Blood, Save Life</div>
        <div class="text-four">From India</div>
      </div>
      <div class="button">
        <a href="/login"><button>Donate Now</button></a>    <a
href="/login"><button>Need Plasma</button></a>
      </div>
    </div>
  </section>

<!-- About Section Start -->
<section class="about" id="about">
  <div class="content">
    <div class="title"><span>About Us</span></div>
  <div class="about-details">
    <div class="left">
      <img src="static/images/home_pic1.jpg" style="height: 100%; width: 100%;"
alt="">
    </div>
    <div class="right" style="padding-left: 70px;">
      <div class="topic">Plasma Bank Info</div>
      <p>We aim to provide prompt, economical and reliable services of the safest
blood and other blood products like RBCs, Platelets, et. Offering the industry-
leading, advanced technology and well-equipped inventory. We make all types of
blood available for the patients and many hospitals. We contribute in saving many
lives in the time of need or in an emergency.</p>
      <div class="button">

      </div>
    </div>
  </div>
  </div>
</section>



<!-- My Skill Section Start -->
<!-- Section Tag and Other Div will same where we need to put same CSS -->


<!-- My Services Section Start -->
 <section class="services" id="product">
   <div class="content">
```

```
     <div class="title"><span>Categories</span></div>
     <div class="boxes">
       <div class="box">
         <div class="icon">
          <i class="fa fa-heartbeat"></i>
       </div>
       <div class="topic">Whole Blood Donation</div>
       <p>Whole blood is the most flexible type of donation. It can be transfused
in its original form, or used to help multiple people when separated into its
specific components of red cells, plasma and platelets.</p>
     </div>
       <div class="box">
         <div class="icon">
           <i class="fa fa-car"></i>
       </div>
       <div class="topic">Power Red Donation</div>
       <p>During a Power Red donation, you give a concentrated dose of red cells,
the part of your blood used every day for those needing transfusions as part of
their care. This type of donation uses an automated process that separates your
red blood cells from the other blood components,
         </p>
     </div>
       <div class="box">
         <div class="icon">
           <i class="fa fa-home"></i>
       </div>
       <div class="topic">Platelet Donation</div>
       <p>In a platelet donation, an apheresis machine collects your platelets
along with some plasma, returning your red cells and most of the plasma back to
you. A single donation of platelets can yield several transfusable units, whereas
it takes about five whole blood donations to make up a single transfusable unit
of platelets.      </p>
     </div>
       <div class="box">
         <div class="icon">
           <i class="fa fa-plug"></i>
       </div>
       <div class="topic">Plasma Donation</div>
       <p>During an AB Elite donation, you give plasma, a part of your blood used
to treat patients in emergency situations. AB plasma can be given to anyone
regardless of their blood type. Plasma is collected through an automated process
that separates plasma from other blood components,       </p>
     </div>
       <div class="box">
         <div class="icon">
```

```
            <i class="fab fa-android"></i>
        </div>
        <div class="topic">About Blood Types</div>
        <p>There are actually more than 8 different blood types, some of which are
not compatible with each other. Find out how your blood type can help hospital
patients in need of a transfusion.      </p>
      </div>
        <div class="box">
          <div class="icon">
            <i class="fa fa-cart-plus"></i>
        </div>
        <div class="topic">About Blood Components</div>
        <p>During medical treatment, patients may receive whole blood or just the
specific blood components they need. Learn more about how blood components impact
patient transfusions.      </p>
      </div>
    </div>
    </div>
</section>



<div class="section" id="faq">
        <div class="content">
          <h1 class="title">FAQ Section</h1>
          <div class="container-1">

              <div class="faq">
                  <div class="question">
                      <h2>Who can donate?</h2>
                      <i class="fa fa-arrow-circle-o-right"></i>
                  </div>
                  <div class="answer">
                      <p>Generally, plasma donors must be 18 years of age and
weigh at least 110 pounds (50kg). All individuals must pass two separate medical
examinations, a medical history screening and testing for transmissible viruses,
before their donated plasma can be used to manufacture plasma protein
therapies.</p>
                  </div>
              </div>

              <div class="faq">
                  <div class="question">
                      <h2>Does it hurt?</h2>
                      <i class="fa fa-arrow-circle-o-right"></i>
```

```
                </div>
                <div class="answer">
                    <p>Most people compare the feeling of the needle to a mild
bee sting. You will also be required to submit to a finger stick test each time
you donate so the collection center medical staff can evaluate your protein and
hemoglobin levels.                    </p>
                </div>
            </div>

            <div class="faq">
                <div class="question">
                    <h2>Is donating plasma safe?                    </h2>
                    <i class="fa fa-arrow-circle-o-right"></i>
                </div>
                <div class="answer">
                    <p>Yes. Plasma donation in IQPP certified collection
centers is performed in a highly controlled, sterile environment by
professionally trained medical staff. All plasma collection equipment is
sterilized and any equipment that comes into contact with you is used only once
to eliminate the possibility of transmitting viral infections.
                    </p>
                </div>
            </div>

        </div></div>

      <div class="container-2">

          <div class="faq">
              <div class="question">
                  <h2>How long does it take?
                  </h2>
                  <i class="fa fa-arrow-circle-o-right"></i>
              </div>
              <div class="answer">
                  <p>Your first donation will take approximately 2 hours.
Return visits on average take about 90 minutes.
                  </p>
              </div>
          </div>

          <div class="faq">
              <div class="question">
                  <h2>What do you do with my plasma?
                  </h2>
```

```
                    <i class="fa fa-arrow-circle-o-right"></i>
                </div>
                <div class="answer">
                    <p>Nearly 500 different types of proteins have been found in
human blood plasma. Approximately 150 of these may be used for diagnosing disease
or manufacturing therapies.
                    </p>
                </div>
            </div>

            <div class="faq">
                <div class="question">
                    <h2>What type of medical screening and testing is done?
                    </h2>
                    <i class="fa fa-arrow-circle-o-right"></i>
                </div>
                <div class="answer">
                    <p>You must have a pre-donation physical which includes
answering medical history questions, tests for viruses such as HIV and Hepatitis
and evaluating your protein and hemoglobin levels.
                    </p>
                </div>
            </div>

        </div>

    </div>


<!--<section class="contact" id="contact">
  <div class="content">
    <div class="title"><span>click button </span></div>
    <div class="text">

      <div class="button">
        <p><a href="../register.html"><button>register now</button></a></p>

      </div>
    </div>
  </div>
</section>
<br>
<br>-->
```

```
<!--<section class="contact" id="contact">
  <div class="content">
    <div class="title"><span>click button </span></div>
    <div class="text">

      <div class="button">
        <p><a href="../Customer Care Registry/login.html"><button>Sign
up</button></a></p>

      </div>
    </div>
  </div>
</section>
<br>
<br>-->


<!-- Contact Me section Start -->
<section class="contact" id="contact">
  <div class="content">
    <div class="title"><span>Contact Us</span></div>
    <div class="text">
      <div class="button">
        <button>Let's Chat</button>
        <br><br>
        <div class="media-icons">
          <a href="#"><i class="fab fa-facebook-f" style="font-size:
30px;"></i></a>      
          <a href="#"><i class="fab fa-twitter" style="font-size:
30px;"></i></a>      
          <a href="#"><i class="fab fa-instagram" style="font-size:
30px;"></i></a>
        </div>
      </div>
    </div>
  </div>
</section>


<!-- Footer Section Start -->
<footer>
  <div class="text">
    <span>Created By   </span>
    <br>
    <span>Praveenkumar.S    |</span>
```

```
    <span>Sangeeth Kumar P  |</span>
    <span>Tharani N P  |</span>
    <span>Vaishnavi.S</span>
  </div>
</footer>
  <script>
    window.watsonAssistantChatOptions = {
      integrationID: "0032bdd2-5b59-4312-a00c-cece7ea268f0", // The ID of this
integration.
      region: "au-syd", // The region your integration is hosted in.
      serviceInstanceID: "30b01793-f193-4ada-a147-4610ae753688", // The ID of
your service instance.
      onLoad: function(instance) { instance.render(); }
    };
    setTimeout(function(){
      const t=document.createElement('script');
      t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/"
+ (window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
      document.head.appendChild(t);
    });

    // Sticky Navigation Menu JS Code
let nav = document.querySelector("nav");
let scrollBtn = document.querySelector(".scroll-button a");
console.log(scrollBtn);
let val;
window.onscroll = function()
  { if(document.documentElement.scrollTop > 20){
    nav.classList.add("sticky");
    scrollBtn.style.display = "block";
  }else{
    nav.classList.remove("sticky");
    scrollBtn.style.display = "none";
  }

}

// Side NavIgation Menu JS Code
let body = document.querySelector("body");
let navBar = document.querySelector(".navbar");
let menuBtn = document.querySelector(".menu-btn");
let cancelBtn = document.querySelector(".cancel-btn");
menuBtn.onclick = function(){
  navBar.classList.add("active");
```

```
  menuBtn.style.opacity = "0";
  menuBtn.style.pointerEvents = "none";
  body.style.overflow = "hidden";
  scrollBtn.style.pointerEvents = "none";
}
cancelBtn.onclick =
  function(){ navBar.classList.remove("active");
  menuBtn.style.opacity = "1";
  menuBtn.style.pointerEvents = "auto";
  body.style.overflow = "auto";
  scrollBtn.style.pointerEvents = "auto";
}

// Side Navigation Bar Close While We Click On Navigation Links
let navLinks = document.querySelectorAll(".menu li a");
for (var i = 0; i < navLinks.length; i++)
  { navLinks[i].addEventListener("click" , function() {
    navBar.classList.remove("active");
    menuBtn.style.opacity = "1";
    menuBtn.style.pointerEvents = "auto";
  });
}


//faq



const question = document.querySelectorAll('.faq');

question.forEach(faq => {
    faq.addEventListener("click", () =>
        { faq.classList.toggle("active");
    })
})
</script>
{% endblock %}
```

Reqplasma.html

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename = 'css/reg.css') }}" />
</head>
<body>
```

```
<div class="container" id="container">
    <div class="form-container sign-up-container">

    </div>
    <div class="form-container sign-in-container">
        <form action="{{ url_for('needplasma') }}" method="post">
            <h1>Request now</h1><br>


            <input type="text" placeholder="name"  name="uname" required>
            <input type="text" placeholder="phone no" name="phone" required>
            <input type="text" name="bloodgroup" placeholder="Your Blood Group" required>
            <input type="text" name="place" placeholder="Your Location" required>
            <input type="text" name="district" placeholder="Your District" value="" required>
            <!-- <p>Already have an account <a href="{{ url_for('login') }}">login</a></p> -->
            <br>
            <button type="submit">request</button>
        </form>
    </div>
    <div class="overlay-container">
        <div class="overlay">
            <div class="overlay-panel overlay-left">


            </div>
            <div class="overlay-panel overlay-right">
                <h1>I'm Admin!</h1>
                <p>Enter your personal details and start journey </p>
                <img src="static/images/admin.png" width="200px">
            </div>
        </div>
    </div>
</div>
```

```
</body>
```

Thanks.html

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename= 'css/login.css') }}" />
</head>
<body>


<div class="container" id="container">
    <div class="form-container sign-up-container">

    </div>
```

```
    <div class="form-container sign-in-container" style="margin-top: 100px; margin-left: 200px;">
        <h1 style="text-align: center;">Thanks for your contribution.</h1><br><br>
        <img src="../static/images/thanks.jpg" alt="" style="height: 200px; width: 200px; margin-left:
100px;"><br>
```

```
        <a href="/dashboard" style="color: blue; font-size: 20px;">Back to home</a>
    </div>
    <div class="overlay-container">
        <!-- <div class="overlay">
            <div class="overlay-panel overlay-left">
                <h1>Welcome Back!</h1>
                <p>To keep connected with us please login with your personal info</p>
                <button class="ghost" id="signIn">Sign In</button>
            </div>
            <div class="overlay-panel overlay-right">
                <h1>Hello, Friend!</h1>
                <p>You Have got a result.</p>
                 <div style="height: 100px; width: 100px; border: 2px solid black; background-image:
url('../static/images/happy.jpg');"></div>
                <img src="../static/images/happy.jpg" width="200px">
            </div>
        </div> -->
    </div>
    </div>
</html>
```

Admin.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Dashboard</title>
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='css/dashboard.css')}}">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
</head>
<body>
    <div class="container-fluid" style="width: 100%;">

        <div class="navigation">
            <ul>
                <li>
                    <a href="#">
                        <!-- <span class="icon"><ion-icon name="cart-sharp"></ion-icon></span> -->
                          <span class="title"><b><h3 style="margin-top: 10px;">PLASMA
DONOR</h3></b></span>
                    </a>
                </li>
                <li>
                    <a href="#">
                        <span class="icon"><ion-icon name="home-sharp"></ion-icon></span>
                        <span class="title"><h3 style="margin-top: 14px;">Dashboard</h3></span>
                    </a>
                </li>
                <!-- <li>
                    <a href="#">
                        <span class="icon"><ion-icon name="logo-dropbox"></ion-icon></span>
                        <span class="title"><h3>My Donations</h3> </span>
                    </a>
                </li>
                <li>
                    <a href="#">
                        <span class="icon"><ion-icon name="create-sharp"></ion-icon></span>
```

```
                    <span class="title"><h3>Rewards</h3></span>
                </a>
            </li>  -->
            <li>
                <a href="/">
                    <span class="icon"><ion-icon name="exit-outline"></ion-icon></span>
                    <span class="title"><h3 style="margin-top: 14px;">Sign out</h3></span>
                </a>
            </li>
        </ul>
    </div>
    <div class="main">
        <div class="topbar">
            <div class="toggle">
                <ion-icon name="menu-outline"></ion-icon>
            </div>
            <!--search-->
            <div class="search">
                <label>
                     <input type="text" placeholder="Search here">
                     <ion-icon name="search-outline"></ion-icon>
                </label>
            </div>
            <!--userImg-->
            <div class="user">
                <img src="static/images/user-dash.jpeg" alt="">
            </div>
          </div>
          <!--cards-->
          <div style="padding: 30px;">
            <table style="padding: 20px;">

<!-- ####################################   COUNT     ################################### -->


                    <table class="table" style="padding: 20px;">
                        <thead class="thead-dark">
                            <tr>
                              <th scope="col">PLACE</th>
                              <th scope="col">COUNT</th>
```

```
                            </tr>
                        </thead>
                        <tbody>
                            <h3>A+VE BLOOD</h3>
                            {% for row in apcount %}
                                        <tr></tr>
                                        <tr>
                                            <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                            <td style="color: black;">{{row[1]}}</td>
```

```
                                            <!-- <td style="color: black;">{{row["PASSWORD"]}}</td> -
->
                                            <!-- <td style="color: black;">{{row["BLOODGROUP"]}}</td>
-->
                                        </tr>
                            {% endfor %}
                        </tbody>
                    </table>
```

```
                        <table class="table" style="padding: 20px;">
                            <thead class="thead-dark">
                                <tr>
                                  <th scope="col">PLACE</th>
                                  <th scope="col">COUNT</th>
```

```
                                </tr>
                            </thead>
                            <tbody>
                                <h3>A-VE BLOOD</h3>
                                {% for row in ancount %}
                                                <tr></tr>
                                                <tr>
                                                    <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                    <td style="color: black;">{{row[1]}}</td>
```

```
                                                    <!-- <td style="color: black;">{{row["PASSWORD"]}}</td> -
->
                                                    <!-- <td style="color: black;">{{row["BLOODGROUP"]}}</td>
-->
                                                </tr>
                                {% endfor %}
                            </tbody>
                        </table>
```

```
                        <table class="table" style="padding: 20px;">
                            <thead class="thead-dark">
                                <tr>
                                  <th scope="col">PLACE</th>
                                  <th scope="col">COUNT</th>
```

```
                                </tr>
                            </thead>
                            <tbody>
                                <h3>B+VE BLOOD</h3>
                                {% for row in bpcount %}
                                                <tr></tr>
                                                <tr>
                                                    <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                    <td style="color: black;">{{row[1]}}</td>
```

```
                                                    <!-- <td style="color: black;">{{row["PASSWORD"]}}</td> -
->
                                                    <!-- <td style="color: black;">{{row["BLOODGROUP"]}}</td>
-->
                                                </tr>
                                {% endfor %}
                            </tbody>
                        </table>
```

```
                        <table class="table" style="padding: 20px;">
                            <thead class="thead-dark">
                                <tr>
                                  <th scope="col">PLACE</th>
                                  <th scope="col">COUNT</th>
```

```
                                </tr>
                            </thead>
                            <tbody>
                                <h3>B-VE BLOOD</h3>
                                {% for row in bncount %}
                                                <tr></tr>
                                                <tr>
                                                    <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                    <td style="color: black;">{{row[1]}}</td>
```

```
                                                    <!-- <td style="color: black;">{{row["PASSWORD"]}}</td> -
->
                                                    <!-- <td style="color: black;">{{row["BLOODGROUP"]}}</td>
-->
                                                </tr>
                                {% endfor %}
                            </tbody>
                        </table>
```

```
                        <table class="table" style="padding: 20px;">
                            <thead class="thead-dark">
                                <tr>
                                  <th scope="col">PLACE</th>
                                  <th scope="col">COUNT</th>
```

```
                                </tr>
                            </thead>
                            <tbody>
                                <h3>AB+VE BLOOD</h3>
                                {% for row in abpcount %}
                                                <tr></tr>
                                                <tr>
                                                    <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                    <td style="color: black; ">{{row[1]}}</td>
```

```
                                                    <!-- <td style="color: black;">{{row["PASSWORD"]}}</td> -
->
                                                    <!-- <td style="color: black;">{{row["BLOODGROUP"]}}</td>
-->
                                                </tr>
                                {% endfor %}
                            </tbody>
                        </table>
```

```
                        <table class="table" style="padding: 20px;">
                            <thead class="thead-dark">
                                <tr>
                                  <th scope="col">PLACE</th>
                                  <th scope="col">COUNT</th>
```

```
                                </tr>
                            </thead>
                            <tbody>
                                <h3>AB-VE BLOOD</h3>
                                {% for row in abncount %}
                                                <tr></tr>
                                                <tr>
```

```
                                                <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                <td style="color: black;">{{row[1]}}</td>
```

```
                                                <!-- <td style="color: black;">{{row["PASSWORD"]}}</td> -
->
                                                <!-- <td style="color: black;">{{row["BLOODGROUP"]}}</td>
-->
                                            </tr>
                                {% endfor %}
                            </tbody>
                        </table>
```

```
                        <table class="table" style="padding: 20px;">
                            <thead class="thead-dark">
                                <tr>
                                  <th scope="col">PLACE</th>
                                  <th scope="col">COUNT</th>
```

```
                                </tr>
                            </thead>
                            <tbody>
                                <h3>O+VE BLOOD</h3>
                                {% for row in opcount %}
                                            <tr></tr>
                                            <tr>
                                                <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                <td style="color: black;">{{row[1]}}</td>
```

```
                                                <!-- <td style="color: black;">{{row["PASSWORD"]}}</td> -
->
                                                <!-- <td style="color: black;">{{row["BLOODGROUP"]}}</td>
-->
                                            </tr>
                                {% endfor %}
                            </tbody>
                        </table>
```

```
                        <table class="table" style="padding: 20px;">
                            <thead class="thead-dark">
                                <tr>
                                  <th scope="col">PLACE</th>
                                  <th scope="col">COUNT</th>
```

```
                                </tr>
                            </thead>
                            <tbody>
                                <h3>O-VE BLOOD</h3>
                                {% for row in oncount %}
                                            <tr></tr>
                                            <tr>
                                                <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                <td style="color: black;">{{row[1]}}</td>
```

```
                                                <!-- <td style="color: black;">{{row["PASSWORD"]}}</td> -
->
```

```
                                                <!-- <td style="color: black;">{{row["BLOODGROUP"]}}</td>
-->
                                            </tr>
                                {% endfor %}
                            </tbody>
                        </table>
<!-- ********************************************** Requested List ********************************** -->
                        <table class="table" style="padding: 20px;">
                            <thead class="thead-dark">
                                <tr>
                                  <th scope="col">NAME</th>
                                  <th scope="col">PHONENUMBER</th>
                                  <th scope="col">BLOODGROUP</th>
                                  <th scope="col">PLACE</th>
                                  <th scope="col">DISTRICT</th>


                                </tr>
                            </thead>
                            <tbody>
                                <h3>REQUESTED LIST</h3>
                                {% for row in selled %}
                                            <tr></tr>
                                            <tr>

                                                <td style="color: black; padding-right:
10px;">{{row["NAME"]}}</td>
                                                <td style="color: black; padding-right:
10px;">{{row["PHONE"]}}</td>
                                                <td style="color: black; padding-right:
10px;">{{row["BLOODGROUP"]}}</td>
                                                <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                <td style="color: black;">{{row["DISTRICT"]}}</td>
```

```
                                                <!-- <td style="color: black;">{{row["PASSWORD"]}}</td> -
->
                                                <!-- <td style="color: black;">{{row["BLOODGROUP"]}}</td>
-->
                                            </tr>
                                {% endfor %}
                            </tbody>
                        </table>
```

```
<!-- ****************************************** Donors List ************************************** -->
```

```
                        <table class="table" style="padding: 20px;">
                        <thead class="thead-dark">
                            <tr>
                              <th scope="col">NAME</th>
                              <th scope="col">PLACE</th>
                              <th scope="col">PHONENUMBER</th>

                            </tr>
                        </thead>
                        <tbody>
                            <h3>A+VE BLOOD DONORS</h3>
                            {% for row in apcustomer %}
                                        <tr></tr>
                                        <tr>
```

```
                                                <td style="color: black; padding-right:
10px;">{{row["NAME"]}}</td>
                                                <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                <td style="color: black;">{{row["PHONENUMBER"]}}</td>

                                            </tr>
                            {% endfor %}
                        </tbody>
                    </table>
```

```
                                    <table class="table" style="padding: 20px;">
                                        <thead class="thead-dark">
                                            <tr>
                                            <th scope="col">NAME</th>
                                            <th scope="col">PLACE</th>
                                            <th scope="col">PHONENUMBER</th>

                                            </tr>
                                        </thead>
                                        <tbody>
                                            <h3>A-VE BLOOD DONORS</h3>
                                            {% for row in ancustomer %}
                                                    <tr></tr>

                                                    <tr>
                                                        <td style="color: black; padding-right:
10px;">{{row["NAME"]}}</td>
                                                        <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                        <td style="color: black;">{{row["PHONENUMBER"]}}</td>

                                                    </tr>
                                            {% endfor %}
                                        </tbody>
                                    </table>
```

```
                        <table class="table" style="padding: 20px;">
                            <thead class="thead-dark">
                                <tr>
                                  <th scope="col">NAME</th>
                                  <th scope="col">PLACE</th>
                                  <th scope="col">PHONENUMBER</th>

                                </tr>
                            </thead>
                            <tbody>
                                <h3>B+VE BLOOD DONORS</h3>

                                    {% for row in bpcustomer %}
                                            <tr>
                                                <td style="color: black; padding-right:
10px;">{{row["NAME"]}}</td>
                                                <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                <td style="color: black;">{{row["PHONENUMBER"]}}</td>

                                            </tr>
                                    {% endfor %}
```

```
                        </tbody>
                    </table>
```

```
                    <table class="table" style="padding: 20px;">
                        <thead class="thead-dark">
                            <tr>
                              <th scope="col">NAME</th>
                              <th scope="col">PLACE</th>
                              <th scope="col">PHONENUMBER</th>

                            </tr>
                        </thead>
                        <tbody>
                            <h3>B-VE BLOOD DONORS</h3>
                                {% for row in bncustomer %}
                                            <tr>
                                                <td style="color: black; padding-right:
10px;">{{row["NAME"]}}</td>
                                                <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                <td style="color: black;">{{row["PHONENUMBER"]}}</td>

                                            </tr>
                                {% endfor %}
                        </tbody>
                    </table>
```

```
                    <table class="table" style="padding: 20px;">
                        <thead class="thead-dark">
                            <tr>
                              <th scope="col">NAME</th>
                              <th scope="col">PLACE</th>
                              <th scope="col">PHONENUMBER</th>

                            </tr>
                        </thead>
                        <tbody>
                            <h3>AB+VE BLOOD DONORS</h3>
                                {% for row in abpcustomer %}
                                            <tr>
                                                <td style="color: black; padding-right:
10px;">{{row["NAME"]}}</td>
                                                <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                <td style="color: black;">{{row["PHONENUMBER"]}}</td>

                                            </tr>
                                {% endfor %}
                            </tbody>
                    </table>
```

```
                    <table class="table" style="padding: 20px;">
                        <thead class="thead-dark">
                            <tr>
                              <th scope="col">NAME</th>
                              <th scope="col">PLACE</th>
                              <th scope="col">PHONENUMBER</th>

                            </tr>
                        </thead>
```

```
                        <tbody>
                            <h3>AB-VE BLOOD DONORS</h3>

                                {% for row in abncustomer %}
                                            <tr>
                                                <td style="color: black; padding-right:
10px;">{{row["NAME"]}}</td>
                                                <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                <td style="color: black;">{{row["PHONENUMBER"]}}</td>

                                            </tr>
                                {% endfor %}
                        </tbody>
                    </table>
```

```
                    <table class="table" style="padding: 20px;">
                        <thead class="thead-dark">
                            <tr>
                              <th scope="col">NAME</th>
                              <th scope="col">PLACE</th>
                              <th scope="col">PHONENUMBER</th>

                            </tr>
                        </thead>
                        <tbody>
                            <h3>O+VE BLOOD DONORS</h3>

                            {% for row in opcustomer %}
                                            <tr>
                                                <td style="color: black; padding-right:
10px;">{{row["NAME"]}}</td>
                                                <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                <td style="color: black;">{{row["PHONENUMBER"]}}</td>

                                            </tr>
                            {% endfor %}
                        </tbody>
                    </table>
```

```
                    <table class="table" style="padding: 20px;">
                        <thead class="thead-dark">
                            <tr>
                              <th scope="col">NAME</th>
                              <th scope="col">PLACE</th>
                              <th scope="col">PHONENUMBER</th>

                            </tr>
                        </thead>
                        <tbody>
                            <h3>O-VE BLOOD DONORS</h3>

                            {% for row in oncustomer %}
                                            <tr>
                                                <td style="color: black; padding-right:
10px;">{{row["NAME"]}}</td>
                                                <td style="color: black; padding-right:
10px;">{{row["PLACE"]}}</td>
                                                <td style="color: black;">{{row["PHONENUMBER"]}}</td>
```

```
                                        </tr>
                            {% endfor %}
                        </tbody>
                    </table>
```

```
                    <table class="table" style="padding: 20px;">
                        <thead class="thead-dark">
                            <tr>
                              <th scope="col">NAME</th>
                              <th scope="col">PHONENUMBER</th>
                              <th scope="col">BLOODGROUP</th>

                            </tr>
                        </thead>
                        <tbody>
                            <h3>OUR CUSTOMERS</h3>

                            {% for row in customer %}
                                        <tr>
                                            <td style="color: black; padding-right:
10px;">{{row["NAME"]}}</td>
                                            <td style="color: black; padding-right:
10px;">{{row["PHONENUMBER"]}}</td>
                                            <td style="color: black;">{{row["BLOODGROUP"]}}</td>

                                        </tr>
                            {% endfor %}
                        </tbody>
                    </table>
```

```
                    </table>
            </div>

        </div>
```

```
    </div>
</body>
```

Signup.html

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename=
'css/reg.css') }}" />
</head>
<body>


<div class="container" id="container">
    <div class="form-container sign-up-container">

    </div>
    <div class="form-container sign-in-container">
        <form action="{{ url_for('register') }}" method="post">
            <h1>Register now</h1><br>


            <input type="text" placeholder="name"  name="uname"/>
            <input type="email" placeholder="Email" name="email"/>
            <input type="text" placeholder="phone no" name="phone"/>
            <input type="password" placeholder="creat new Password"
name="password"/>
            <input type="password" placeholder="Confirm new Password" />
            <input type="text" name="bloodgroup" placeholder="Your Blood Group">
            <p>Already have an account <a href="{{ url_for('login')
}}">login</a></p>
            <br>
            <button type="submit">register</button>
        </form>
    </div>
    <div class="overlay-container">
        <div class="overlay">
            <div class="overlay-panel overlay-left">


            </div>
            <div class="overlay-panel overlay-right">
                <h1>I'm Admin!</h1>
                <p>Enter your personal details and start journey </p>
                <img src="static/images/admin.png" width="200px">
```

```
            </div>
        </div>
    </div>
</div>




</body>
```

Login.html

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename=
'css/login.css') }}" />
</head>
<body>


<div class="container" id="container">
    <div class="form-container sign-up-container">

    </div>
    <div class="form-container sign-in-container">
        <form action="{{url_for('signin')}}" method="POST">
            <h1>Sign in</h1>
            <span>or use your account</span>
            <br>
            <input type="email" placeholder="Email" name="email" />
            <input type="password" placeholder="Password" name="password" />
            <p>Don't have an account <a href="{{ url_for('signup') }}">Sign
up</a></p>
            <button>Sign In</button>
        </form>
    </div>
    <div class="overlay-container">
        <div class="overlay">
            <div class="overlay-panel overlay-left">
                <h1>Welcome Back!</h1>
```

```
                <p>To keep connected with us please login with your personal
info</p>
                <button class="ghost" id="signIn">Sign In</button>
            </div>
            <div class="overlay-panel overlay-right">
                <h1>Hello, Friend!</h1>
                <p>Enter your personal details and start journey with us</p>
                <img src="static/images/lady0.png" width="200px">
            </div>
        </div>
    </div>
</div>



</body>
```

App.py

```
import email
from email import message
from importlib.resources import contents
from tkinter import S
from turtle import title
from flask import Flask,redirect, render_template, request, session, url_for, flash
# from flask_restful import Resource, Api, reqparse
# import sendgrid
import sys
import os
import json
import requests
from pyexpat import model
from sqlalchemy import PrimaryKeyConstraint
from werkzeug.utils import secure_filename
import ibm_db
from flask_mail import Mail, Message
from markupsafe import escape

# import required module
import requests
import json
# mention url
url = "https://www.fast2sms.com/dev/bulkV2"
```

```
# create a dictionary
```

```
# create a dictionary
headers = {
    'authorization': 'QqbHW076SFDTledzUu4yhiYNIK2tf3LEnkc9Br5ZasOjp1VwxMLsyMZXA8IUPcEbdB6GJgvnDhwFfV2a',
    'Content-Type': "application/x-www-form-urlencoded",
```

```
    'Cache-Control': "no-cache"
}
# make a post request
```

```
app = Flask(__name__)
```

```
app.secret_key = b'_5#y2L"F4Q8z\n\xec]/'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=8e359033-a1c9-4643-82ef-
8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30120;SECURITY=SSL;SSLServerCertificate=Digi
CertGlobalRootCA.crt;UID=vnp47940;PWD=owxeRayzLL2DAzER;", "", "")
print(conn)
print("connection successful...")
```

```
@app.route('/')
def home():
    message = "TEAM ID : PNT2022TMID37544" +" "+ "BATCH ID : B1-1M3E "
    return render_template('index.html',mes=message)
```

```
@app.route('/anegative/<andis>')
def anegative(andis):
    ancustomer = []
    sql = f"SELECT * FROM ANEGATIVE where district = '{escape(andis)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        ancustomer.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
    sql = "SELECT PHONENUMBER FROM ANEGATIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
    nums = ''
```

```
    length = len(ancustomer)
```

```
    for i in range(0,length):
        nums = nums + ancustomer[i][1] + ','
    print(nums)
```

```
    my_data = {
        # Your default Sender ID
        'sender_id': 'FTWSMS',

        # Put your message here!
        'message': 'Urgent.....There is a demand for your blood group. We request you to donate your blood in
your nearby BloodBank connect with our Organization.',

        'language': 'english',
        'route': 'p',

        # You can send sms to multiple numbers
        # separated by comma.

         'numbers': nums
```

```
        }
    response = requests.request("POST",
                                url,
                                data = my_data,
                                headers = headers)
        #load json data from sourc
    returned_msg = json.loads(response.text)
```

```
        # print the send message
    print(returned_msg['message'])
```

```
    return render_template('comments.html', ancustomer = ancustomer)
```

```
@app.route('/apositive/<apdis>')
def apositive(apdis):
    apcustomer = []
    sql = f"SELECT * FROM APOSITIVE where district = '{escape(apdis)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        apcustomer.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
    if apcustomer:
        sql = "SELECT * FROM APOSITIVE"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)

    nums = ''
```

```
    length = len(apcustomer)
```

```
    for i in range(0,length):
        nums = nums + apcustomer[i][1] + ','
    print(nums)
```

```
    my_data = {
        # Your default Sender ID
        'sender_id': 'FTWSMS',

        # Put your message here!
        'message': 'Argent.....There is a demand for your blood group. We request you to donate your blood in
your nearby BloodBank connect with our Organization.',

        'language': 'english',
        'route': 'p',

        # You can send sms to multiple numbers
        # separated by comma.

         'numbers': nums
        }
    response = requests.request("POST",
                                url,
                                data = my_data,
```

```
                            headers = headers)
        #load json data from sourc
    returned_msg = json.loads(response.text)
```

```
        # print the send message
    print(returned_msg['message'])
```

```
    return render_template('comments.html', apcustomer = apcustomer)
```

```
@app.route('/bnegative/<bndis>')
def bnegative(bndis):
    bncustomer = []
    sql = f"SELECT * FROM BNEGATIVE where district = '{escape(bndis)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        bncustomer.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
    if bncustomer:
        sql = "SELECT * FROM BNEGATIVE"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)
```

```
    nums = ''
```

```
    length = len(bncustomer)
```

```
    for i in range(0,length):
        nums = nums + bncustomer[i][1] + ','
    print(nums)
```

```
    my_data = {
        # Your default Sender ID
        'sender_id': 'FTWSMS',

        # Put your message here!
        'message': 'Argent.....There is a demand for your blood group. We request you to donate your blood in
your nearby BloodBank connect with our Organization.',

        'language': 'english',
        'route': 'p',

        # You can send sms to multiple numbers
        # separated by comma.

         'numbers': nums
        }
    response = requests.request("POST",
                            url,
                            data = my_data,
                            headers = headers)
        #load json data from sourc
    returned_msg = json.loads(response.text)
```

```
        # print the send message
    print(returned_msg['message'])
```

```
    return render_template('comments.html', bncustomer = bncustomer)
```

```
@app.route('/bpositive/<bpdis>')
```

```
def bpositive(bpdis):
    bpcustomer = []
    sql = f"SELECT * FROM BPOSITIVE where district = '{escape(bpdis)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        bpcustomer.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
    if bpcustomer:
        sql = "SELECT * FROM BPOSITIVE"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)
```

```
    nums = ''
```

```
    length = len(bpcustomer)
```

```
    for i in range(0,length):
        nums = nums + bpcustomer[i][1] + ','
    print(nums)
```

```
    my_data = {
        # Your default Sender ID
        'sender_id': 'FTWSMS',

        # Put your message here!
        'message': 'Argent.....There is a demand for your blood group. We request you to donate your blood in
your nearby BloodBank connect with our Organization.',

        'language': 'english',
        'route': 'p',

        # You can send sms to multiple numbers
        # separated by comma.

         'numbers': nums
        }
    response = requests.request("POST",
                            url,
                            data = my_data,
                            headers = headers)
        #load json data from sourc
    returned_msg = json.loads(response.text)
```

```
        # print the send message
    print(returned_msg['message'])
    return render_template('comments.html', bpcustomer = bpcustomer)
```

```
@app.route('/abnegative/<abndis>')
def abnegative(abndis):
    abncustomer = []
    sql = f"SELECT * FROM ABNEGATIVE where district = '{escape(abndis)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        abncustomer.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
    if abncustomer:
```

```
        sql = "SELECT * FROM ABNEGATIVE"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)
```

```
    nums = ''
```

```
    length = len(abncustomer)
```

```
    for i in range(0,length):
        nums = nums + abncustomer[i][1] + ','
    print(nums)
```

```
    my_data = {
        # Your default Sender ID
        'sender_id': 'FTWSMS',

        # Put your message here!
        'message': 'Argent.....There is a demand for your blood group. We request you to donate your blood in
your nearby BloodBank connect with our Organization.',

        'language': 'english',
        'route': 'p',

        # You can send sms to multiple numbers
        # separated by comma.

         'numbers': nums
        }
    response = requests.request("POST",
                                url,
                                data = my_data,
                                headers = headers)
        #load json data from sourc
    returned_msg = json.loads(response.text)
```

```
        # print the send message
    print(returned_msg['message'])
```

```
    return render_template('comments.html', abncustomer = abncustomer)
```

```
@app.route('/abpositive/<abpdis>')
def abpositive(abpdis):
    abpcustomer = []
    sql = f"SELECT * FROM ABPOSITIVE where district = '{escape(abpdis)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        abpcustomer.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
    if abpcustomer:
        sql = "SELECT * FROM ABPOSITIVE"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)
```

```
    nums = ''
```

```
    length = len(abpcustomer)
```

```
    for i in range(0,length):
        nums = nums + abpcustomer[i][1] + ','
```

```
    print(nums)
```

```
    my_data = {
        # Your default Sender ID
        'sender_id': 'FTWSMS',

        # Put your message here!
        'message': 'Argent.....There is a demand for your blood group. We request you to donate your blood in
your nearby BloodBank connect with our Organization.',

        'language': 'english',
        'route': 'p',

        # You can send sms to multiple numbers
        # separated by comma.

         'numbers': nums
        }
    response = requests.request("POST",
                                url,
                                data = my_data,
                                headers = headers)
        #load json data from sourc
    returned_msg = json.loads(response.text)
```

```
        # print the send message
    print(returned_msg['message'])
```

```
    return render_template('comments.html', abpcustomer = abpcustomer)
```

```
@app.route('/onegative/<ondis>')
def onegative(ondis):
    oncustomer = []
    sql = f"SELECT * FROM ONEGATIVE where district = '{escape(ondis)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        oncustomer.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
    if oncustomer:
        sql = "SELECT * FROM ONEGATIVE"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)
```

```
    nums = ''
```

```
    length = len(oncustomer)
```

```
    for i in range(0,length):
        nums = nums + oncustomer[i][1] + ','
    print(nums)
```

```
    my_data = {
        # Your default Sender ID
        'sender_id': 'FTWSMS',

        # Put your message here!
        'message': 'Argent.....There is a demand for your blood group. We request you to donate your blood in
your nearby BloodBank connect with our Organization.',
```

```
        'language': 'english',
        'route': 'p',

        # You can send sms to multiple numbers
        # separated by comma.

         'numbers': nums
        }
    response = requests.request("POST",
                            url,
                            data = my_data,
                            headers = headers)
        #load json data from sourc
    returned_msg = json.loads(response.text)
```

```
        # print the send message
    print(returned_msg['message'])
```

```
    return render_template('comments.html', oncustomer = oncustomer)
```

```
@app.route('/opositive/<opdis>')
def opositive(opdis):
    opcustomer = []
    sql = f"SELECT * FROM OPOSITIVE where district = '{escape(opdis)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        opcustomer.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
    if opcustomer:
        sql = "SELECT * FROM OPOSITIVE"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)
```

```
    nums = ''
```

```
    length = len(opcustomer)
```

```
    for i in range(0,length):
        nums = nums + opcustomer[i][1] + ','
    print(nums)
```

```
    my_data = {
        # Your default Sender ID
        'sender_id': 'FTWSMS',

        # Put your message here!
        'message': 'Argent.....There is a demand for your blood group. We request you to donate your blood in
your nearby BloodBank connect with our Organization.',

        'language': 'english',
        'route': 'p',

        # You can send sms to multiple numbers
        # separated by comma.

         'numbers': nums
        }
    response = requests.request("POST",
```

```
                            url,
                            data = my_data,
                            headers = headers)
        #load json data from sourc
    returned_msg = json.loads(response.text)
```

```
        # print the send message
    print(returned_msg['message'])
```

```
    return render_template('comments.html', opcustomer = opcustomer)
```

```
@app.route('/login', methods=['GET','POST'])
def login():
    return render_template('login.html')
```

```
@app.route('/signup', methods = ['GET','POST'])
def signup():
    return render_template('signup.html')
```

```
@app.route('/reqplasma', methods = ['GET','POST'])
def reqplasma():
    return render_template('plasmareq.html')
```

```
@app.route('/complaint')
def complaint():
    return render_template('complaint.html')
```

```
@app.route('/agentreg')
def agentreg():
    return render_template('agentreg.html')
```

```
@app.route('/agentlogin')
def agentlogin():
    return render_template('agentlogin.html')
```

```
@app.route('/agenthome')
def agenthome():
    return render_template('agenthome.html')
```

```
@app.route('/dashboard')
def dashboard():
    return render_template('dashboard.html')
```

```
@app.route('/admin')
def admin():
```

```
    customer = []
    sql = f"SELECT * FROM CUSTOMER;"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        customer.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
if customer:
    sql = "SELECT * FROM CUSTOMER"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
apcustomer = []
sql = f"SELECT * FROM APOSITIVE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    apcustomer.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if apcustomer:
    sql = "SELECT * FROM APOSITIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
ancustomer = []
sql = f"SELECT * FROM ANEGATIVE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    ancustomer.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if ancustomer:
    sql = "SELECT * FROM ANEGATIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
bpcustomer = []
sql = f"SELECT * FROM BPOSITIVE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    bpcustomer.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if bpcustomer:
    sql = "SELECT * FROM BPOSITIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
bncustomer = []
sql = f"SELECT * FROM BNEGATIVE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    bncustomer.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if bncustomer:
    sql = "SELECT * FROM BNEGATIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
abpcustomer = []
sql = f"SELECT * FROM ABPOSITIVE;"
stmt = ibm_db.exec_immediate(conn, sql)
```

```
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    abpcustomer.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if abpcustomer:
    sql = "SELECT * FROM ABPOSITIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
abncustomer = []
sql = f"SELECT * FROM ABNEGATIVE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    abncustomer.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if abncustomer:
    sql = "SELECT * FROM ABNEGATIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
opcustomer = []
sql = f"SELECT * FROM OPOSITIVE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    opcustomer.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if opcustomer:
    sql = "SELECT * FROM OPOSITIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
oncustomer = []
sql = f"SELECT * FROM ONEGATIVE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    oncustomer.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if oncustomer:
    sql = "SELECT * FROM ONEGATIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
apcount = []
sql = f"SELECT PLACE, count(*) as num FROM APOSITIVE GROUP BY PLACE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    apcount.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if apcount:
    sql = "SELECT * FROM APOSITIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
ancount = []
sql = f"SELECT PLACE, count(*) as num FROM ANEGATIVE GROUP BY PLACE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    ancount.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if ancount:
    sql = "SELECT * FROM ANEGATIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
bpcount = []
sql = f"SELECT PLACE, count(*) as num FROM BPOSITIVE GROUP BY PLACE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    bpcount.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if bpcount:
    sql = "SELECT * FROM BPOSITIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
bncount = []
sql = f"SELECT PLACE, count(*) as num FROM BNEGATIVE GROUP BY PLACE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    bncount.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if bncount:
    sql = "SELECT * FROM BNEGATIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
abpcount = []
sql = f"SELECT PLACE, count(*) as num FROM ABPOSITIVE GROUP BY PLACE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    abpcount.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
if abpcount:
    sql = "SELECT * FROM ABPOSITIVE"
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_both(stmt)
```

```
abncount = []
sql = f"SELECT PLACE, count(*) as num FROM ABNEGATIVE GROUP BY PLACE;"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_both(stmt)
while dictionary != False:
    abncount.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
```

```
    if abncount:
        sql = "SELECT * FROM ABNEGATIVE"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)
```

```
    opcount = []
    sql = f"SELECT PLACE, count(*) as num FROM OPOSITIVE GROUP BY PLACE;"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        opcount.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
    if opcount:
        sql = "SELECT * FROM OPOSITIVE"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)
```

```
    oncount = []
    sql = f"SELECT PLACE, count(*) as num FROM ONEGATIVE GROUP BY PLACE;"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        oncount.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
    if oncount:
        sql = "SELECT * FROM ONEGATIVE"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)
```

```
    selled = []
    sql = f"SELECT * FROM REQPLASMA;"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        selled.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
```

```
    if selled:
        sql = "SELECT * FROM REQPLASMA"
        stmt = ibm_db.exec_immediate(conn, sql)
        user = ibm_db.fetch_both(stmt)
```

```
    return render_template('admin.html', customer = customer, apcustomer = apcustomer, ancustomer =
ancustomer,bpcustomer = bpcustomer, bncustomer = bncustomer, abncustomer = abncustomer, abpcustomer =
abpcustomer, opcustomer = opcustomer, oncustomer = oncustomer, apcount = apcount, ancount = ancount, bpcount
= bpcount, bncount = bncount, abpcount = abpcount, abncount = abncount, opcount = opcount, oncount = oncount,
selled = selled )
```

```
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        uname = request.form['uname']
        mail = request.form['email']
        phone = request.form['phone']
        password = request.form['password']
        bloodgrp = request.form['bloodgroup']
```

```
        sql = "SELECT * FROM customer WHERE name=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,uname)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
```

```
    if account:
        return render_template('index.html', msg="You are already a member, please login using your
details....")

    else:
      insert_sql = "INSERT INTO customer VALUES (?,?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, uname)
      ibm_db.bind_param(prep_stmt, 2, mail)
      ibm_db.bind_param(prep_stmt, 3, phone)
      ibm_db.bind_param(prep_stmt, 4, password)
      ibm_db.bind_param(prep_stmt, 5, bloodgrp)
      ibm_db.execute(prep_stmt)

    return render_template('login.html', msg="Student Data saved successfuly..")
```

```
@app.route('/signin', methods=['GET', 'POST'])
def signin():
    sec = ''
    if request.method == 'POST':

        mail = request.form['email']
        password = request.form['password']
        print(mail, password)
```

```
        if mail == 'admin@gmail.com' and password == 'admin':
            return redirect(url_for('admin'))
```

```
        else:
            sql = f"select * from customer where email='{escape(mail)}' and password= '{escape(password)}'"
            stmt = ibm_db.exec_immediate(conn, sql)
            data = ibm_db.fetch_both(stmt)

        if data:
            session["mail"] = escape(mail)
            session["password"] = escape(password)
            return redirect(url_for('dashboard'))
```

```
        else:
            flash("Mismatch in credetials", "danger")
```

```
@app.route('/needplasma', methods=['GET', 'POST'])
def needplasma():
    if request.method == 'POST':
        uname = request.form['uname']
        phone = request.form['phone']
        bloodgrp = request.form['bloodgroup']
```

```
        place = request.form['place']
        district = request.form['district']
```

```
        # sql = "SELECT * FROM reqplasma WHERE name=?"
        # stmt = ibm_db.prepare(conn, sql)
        # ibm_db.bind_param(stmt,1,uname)
        # ibm_db.execute(stmt)
        # account = ibm_db.fetch_assoc(stmt)
```

```
    # if account:
    #     return render_template('index.html', msg="You are already a member, please login using your
details....")

    # else:
        if bloodgrp == 'A-VE':
```

```
            insert_sql = "INSERT INTO reqplasma VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
```

```
            return redirect(url_for("anegative", andis = district))
```

```
        elif bloodgrp == 'A+VE':
            insert_sql = "INSERT INTO reqplasma VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
            return redirect(url_for("apositive", apdis = district))
```

```
        elif bloodgrp == 'B+VE':
            insert_sql = "INSERT INTO reqplasma VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
            return redirect(url_for("bpositive", bpdis = district))
```

```
        elif bloodgrp == 'B-VE':
            insert_sql = "INSERT INTO reqplasma VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
            return redirect(url_for("bnegative", bndis = district))
```

```
        elif bloodgrp == 'AB-VE':
            insert_sql = "INSERT INTO reqplasma VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
            return redirect(url_for("abnegative", abndis = district))
```

```
        elif bloodgrp == 'AB+VE':
            insert_sql = "INSERT INTO reqplasma VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
            return redirect(url_for("abpositive", abpdis = district))
```

```
        elif bloodgrp == 'O-VE':
            insert_sql = "INSERT INTO reqplasma VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
            return redirect(url_for("onegative", ondis = district))
```

```
        elif bloodgrp == 'O+VE':
            insert_sql = "INSERT INTO reqplasma VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
            return redirect(url_for("opositive", opdis = district))
```

```
        else:
            return "Please INSERT a valid Blood Group and Enter the Blood group in CAPITAL LETTERS like (A+VE,
B-VE, AB+VE)..."

    # return render_template('comments.html', msg="Student Data saved successfuly..")
```

```
@app.route('/donateplasma', methods=['GET', 'POST'])
def donateplasma():
    if request.method == 'POST':
        uname = request.form['uname']
        phone = request.form['phone']
        bloodgrp = request.form['bloodgroup']
        place = request.form['place']
        district = request.form['district']
```

```
        # sql = "SELECT * FROM donateplasma WHERE name=?"
```

```
        # stmt = ibm_db.prepare(conn, sql)
        # ibm_db.bind_param(stmt,1,uname)
        # ibm_db.execute(stmt)
        # account = ibm_db.fetch_assoc(stmt)
```

```
    # if account:
    #     return render_template('index.html', msg="You are already a member, please login using your
details....")

    # else:
        if bloodgrp == 'A+VE':
            insert_sql = "INSERT INTO APOSITIVE VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
```

```
        elif (bloodgrp == 'A-VE'):
            insert_sql = "INSERT INTO ANEGATIVE VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
```

```
        elif (bloodgrp == 'B+VE'):
            insert_sql = "INSERT INTO BPOSITIVE VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
```

```
        elif (bloodgrp == 'B-VE'):
            insert_sql = "INSERT INTO BNEGATIVE VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
```

```
        elif (bloodgrp == 'AB+VE'):
            insert_sql = "INSERT INTO ABPOSITIVE VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)

        elif (bloodgrp == 'AB-VE'):
```

```
            insert_sql = "INSERT INTO ABNEGATIVE VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)

        elif (bloodgrp == 'O+VE'):
            insert_sql = "INSERT INTO OPOSITIVE VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
```

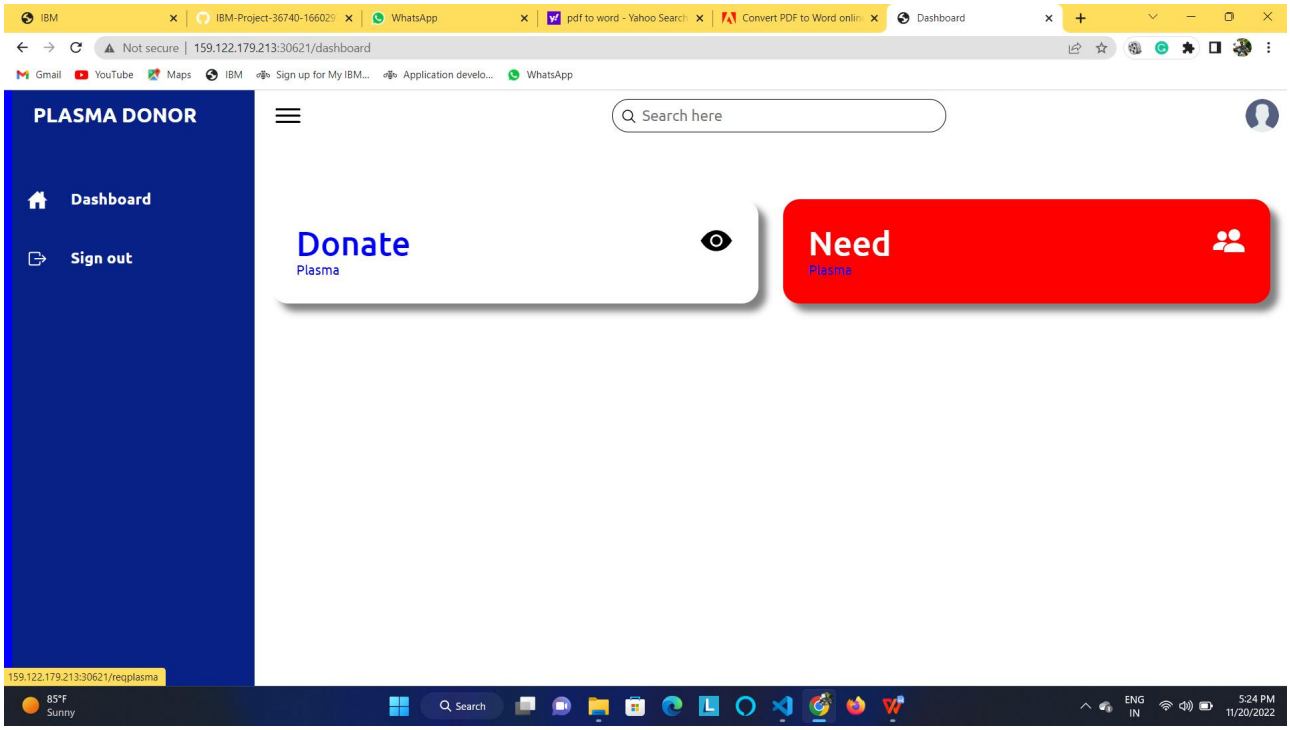
```
        elif (bloodgrp == 'O-VE'):
            insert_sql = "INSERT INTO ONEGATIVE VALUES (?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, uname)
            ibm_db.bind_param(prep_stmt, 2, phone)
            ibm_db.bind_param(prep_stmt, 3, bloodgrp)
            ibm_db.bind_param(prep_stmt, 4, place)
            ibm_db.bind_param(prep_stmt, 5, district)
            ibm_db.execute(prep_stmt)
```

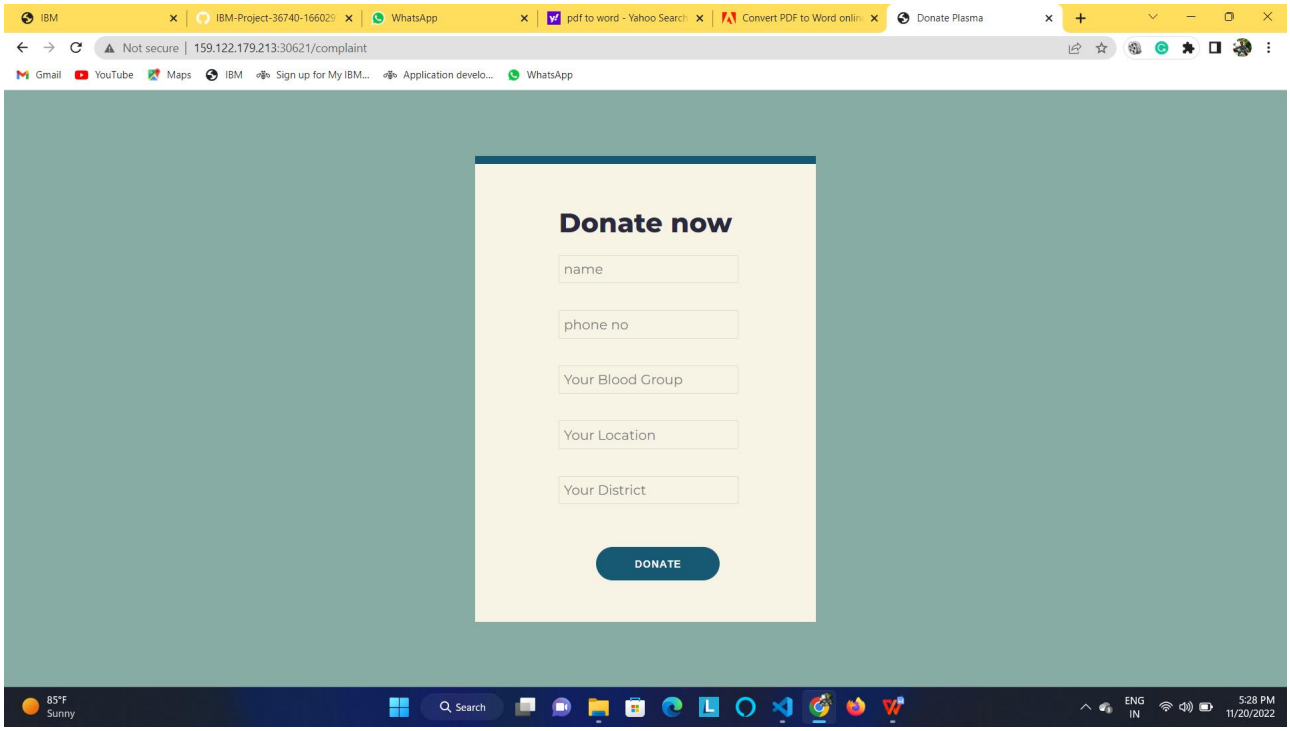
```
        else:
            return "Please INSERT a valid Blood Group and Enter the Blood group in CAPITAL LETTERS like (A+VE,
B-VE, AB+VE)..."

    return render_template('thanks.html', msg="Student Data saved successfuly..")
    # return redirect(url_for("place",plc = place))
```

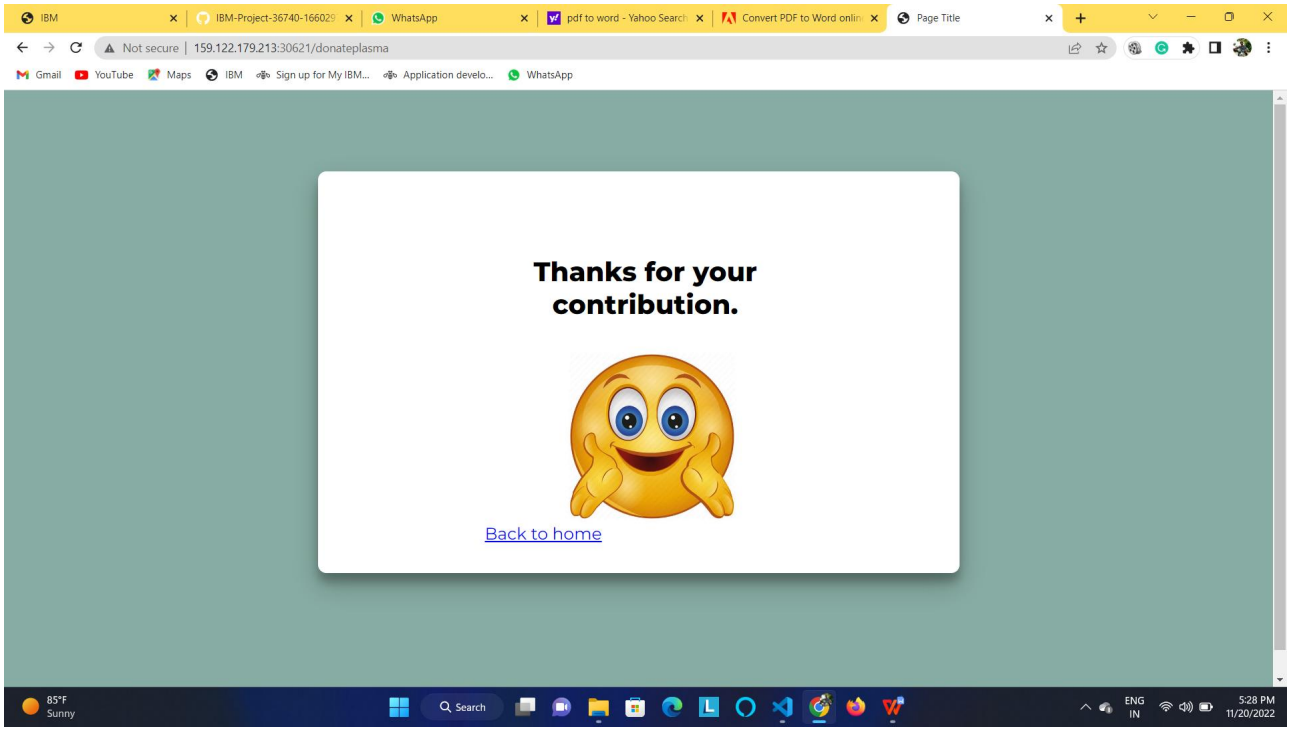
```
#-------------------------------------- Count -----------------------------------
```

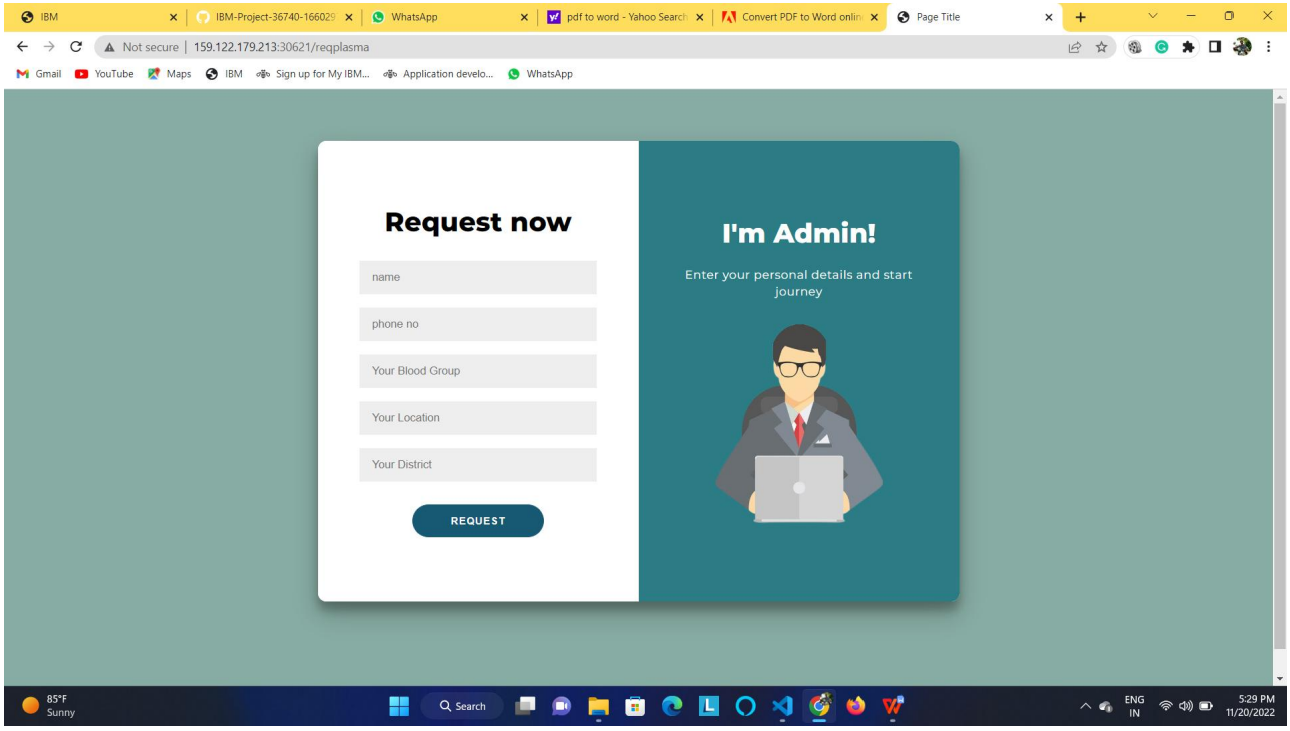
```
if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000, debug=True)
```

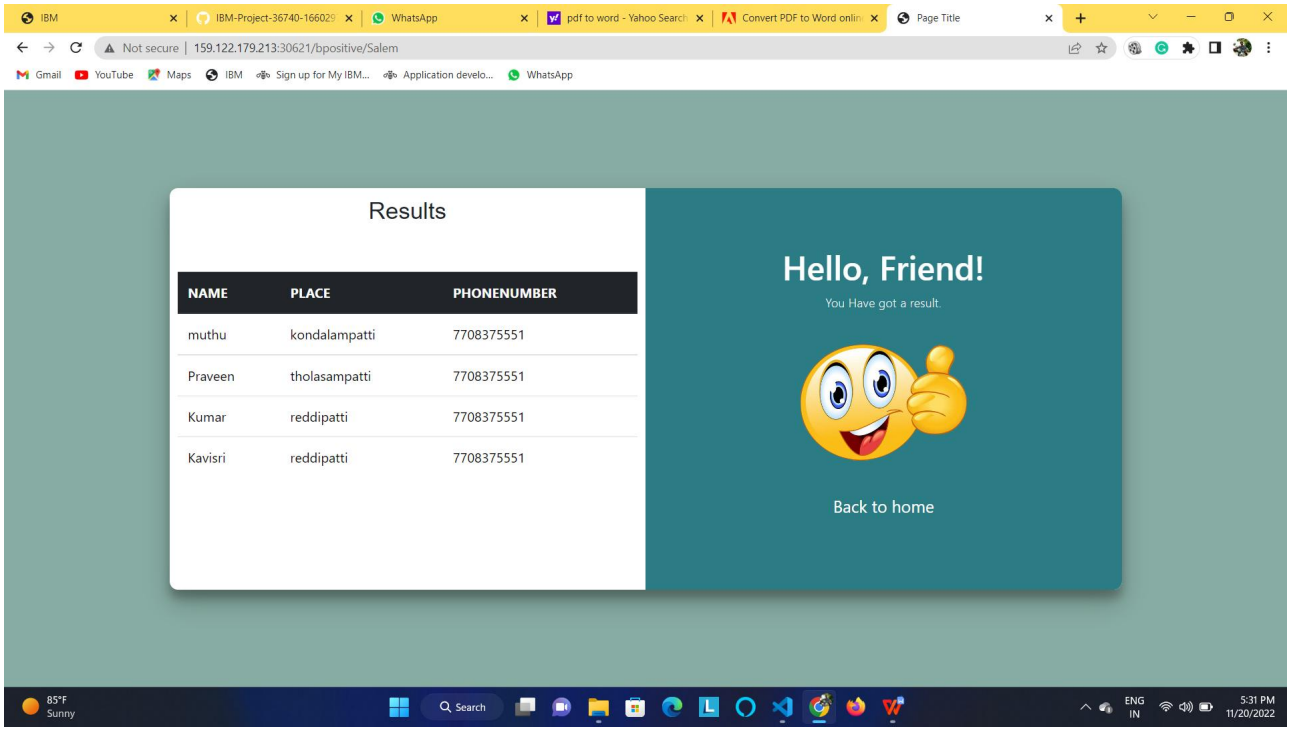
SCREEENSHOTS (OUTPUT) :

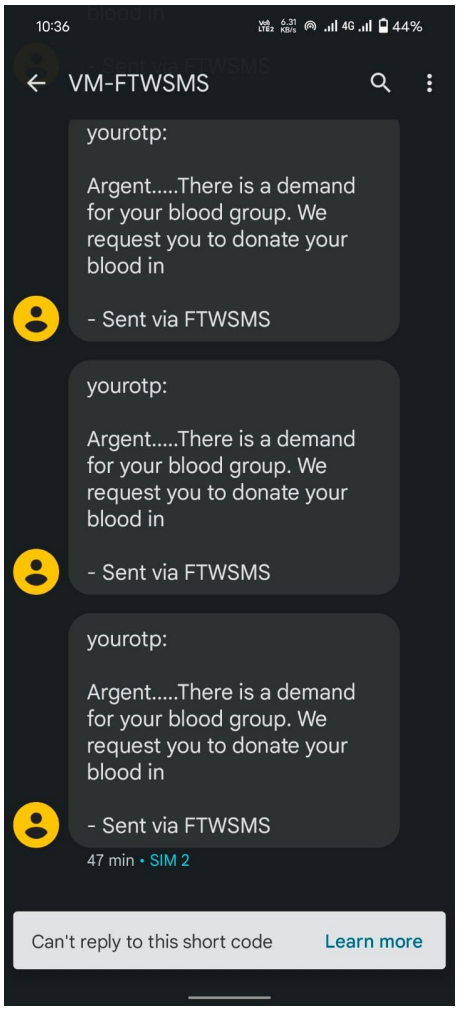




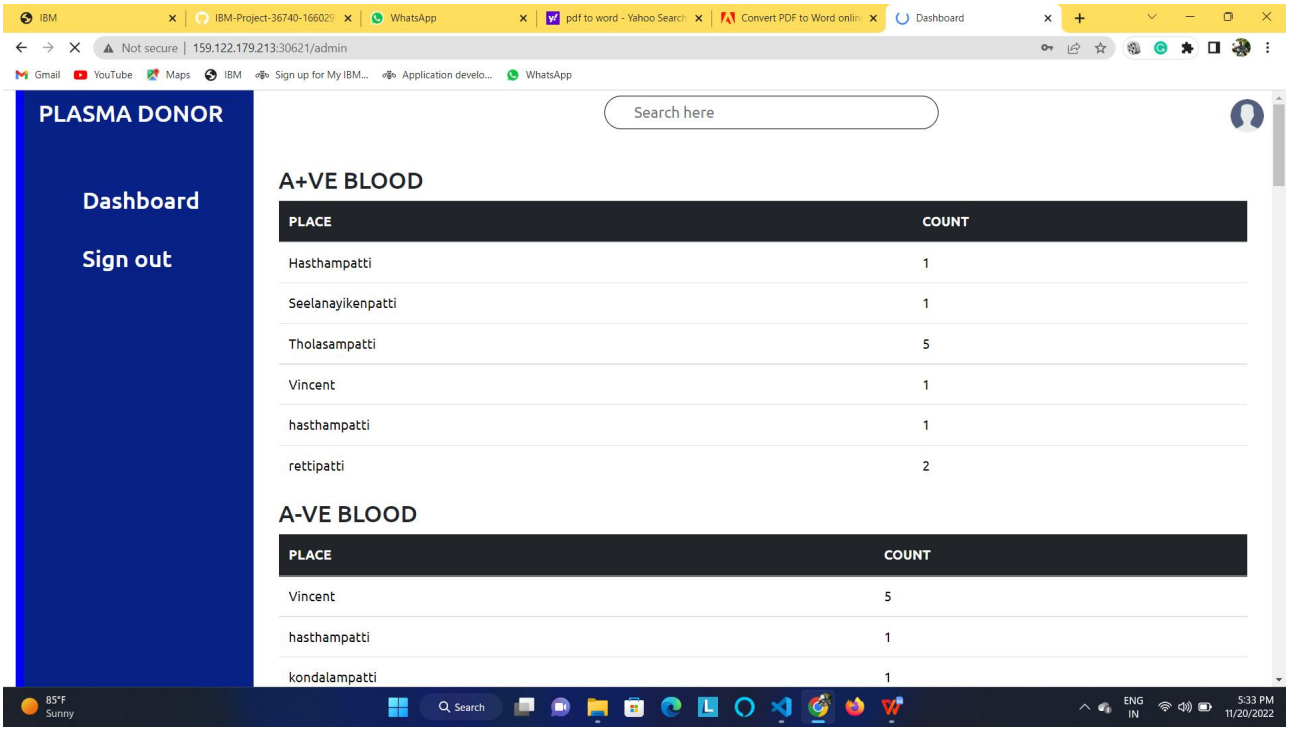

IBM
IBM-Project-36740-166029
WhatsApp
pdf to word - Yahoo Search
Convert PDF to Word onlin
Page Title
Not secure | 159.122.179.213:30621/donateplasma
Gmail
YouTube
Maps
IBM
Sign up for My IBM...
Application develo...
WhatsApp
Thanks for your contribution.
Back to home
85°F
Sunny
Search
ENG
IN
5:28 PM
11/20/2022

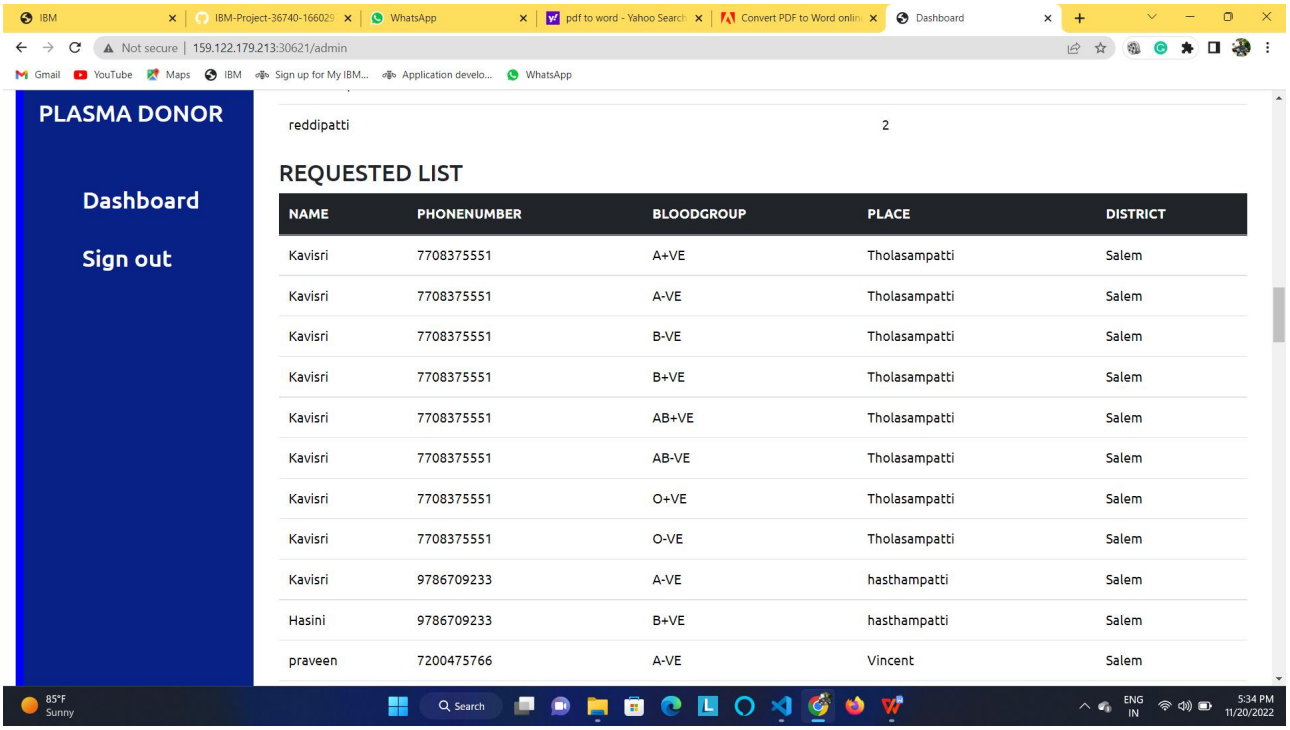

IBM
IBM-Project-36740-166029
WhatsApp
pdf to word - Yahoo Search
Convert PDF to Word onlin
Page Title
Not secure | 159.122.179.213:30621/reqplasma
Gmail
YouTube
Maps
IBM
Sign up for My IBM...
Application develo...
WhatsApp
Request now
name
phone no
Your Blood Group
Your Location
Your District
REQUEST
I'm Admin!
Enter your personal details and start journey
85°F
Sunny
Search
ENG
IN
5:29 PM
11/20/2022

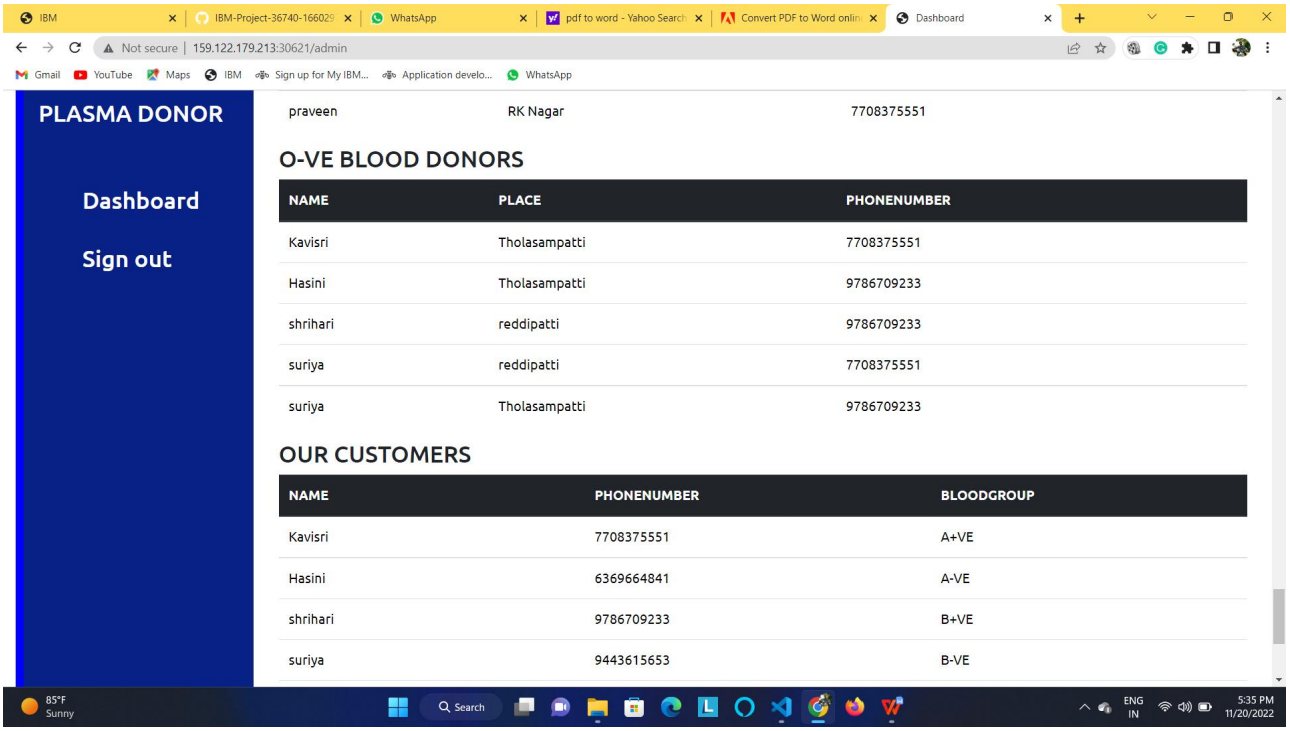

Not secure | 159.122.179.213:30621/bpositive/Salem
Results
NAME PLACE PHONENUMBER
muthu kondalampatti 7708375551
Praveen tholasampatti 7708375551
Kumar reddipatti 7708375551
Kavisri reddipatti 7708375551
Hello, Friend!
You Have got a result.
Back to home


10:36
44%
VM-FTWSMS
yourotp:
Argent.....There is a demand for your blood group. We request you to donate your blood in
- Sent via FTWSMS
yourotp:
Argent.....There is a demand for your blood group. We request you to donate your blood in
- Sent via FTWSMS
yourotp:
Argent.....There is a demand for your blood group. We request you to donate your blood in
- Sent via FTWSMS
47 min • SIM 2
Can't reply to this short code
Learn more

Admin DashBoard :






IBM
IBM-Project-36740-166029
WhatsApp
pdf to word - Yahoo Search
Convert PDF to Word online
Dashboard
Not secure | 159.122.179.213:30621/admin
Gmail
YouTube
Maps
IBM
Sign up for My IBM...
Application develo...
WhatsApp
PLASMA DONOR
Dashboard
Sign out
praveen RK Nagar 7708375551
O-VE BLOOD DONORS
NAME PLACE PHONENUMBER
Kavisri Tholasampatti 7708375551
Hasini Tholasampatti 9786709233
shrihari reddipatti 9786709233
suriya reddipatti 7708375551
suriya Tholasampatti 9786709233
OUR CUSTOMERS
NAME PHONENUMBER BLOODGROUP
Kavisri 7708375551 A+VE
Hasini 6369664841 A-VE
shrihari 9786709233 B+VE
suriya 9443615653 B-VE
85°F
Sunny
Search
ENG
IN
5:35 PM
11/20/2022