VIT

Institute of Technology

(Estd. u/s 3 of UGC Act 1956)

Vellore Institute of Technology, Vellore

Webulance

{Personal Assistance Website}

Done by

19BCE0522 Gokul Raj

19BCE2266 Shreyas K.

19BCE0479 Anmol Sahu

19BCE0661 Shivansh Gupta

19BCE2191 Ashutosh Mahapatra

Under the Guidance of

Prof. A. Vijayarani

For the course

Course Code: CSE3002

Course Name: Internet and Web Programming

Semester: Winter 2021

CONTENT

Sno.	Topic	Page Number
1.	ABSTRACT	3
2.	INTRODUCTION	4-5
3.	ER DIAGRAM	6-8
4.	PROCESS FLOW DIAGRAM	9-11
5.	DATABASE DESIGN	12-15
6.	CODE	16-38
7.	SCREEN SHOTS	39-42
8.	CONCLUSION	43
9.	REFERENCES	44

ABSTRACT

The era of Information technology has become a crucial part of the dynamic life of every human being in the world and thereby indicates the exponential rise of smartphones. Elderly people who are unable to provide accurate information and utilize the emergency phone calls, users who find themselves in an unknown location that can't be described or provide an accurate address in an emergency, casualties caused by the late arrival of the ambulance, and searching for an available nearby ambulance have been some of the hustling factors faced by the current fast pace community.

With more congested roads and insufficient information, the search and rescue operation becomes nearly impossible. This Web-based application project will change the native way of calling an ambulance and improve the efficiency and reliability of emergency medical services (EMS).

With people who require medical assistance, thinking calmly becomes difficult. Visual assistance can be very helpful in such situations. With the use of the 'Webulance' web app, the user not only gets the added advantage of knowing the nearest hospitals and their routes, but can also call for medical assistance in one click. This seems much easier and fast as compared to the traditional method of calling emergency services.

Making full use of the modern technology available, patients in distress can be guided to the nearest health care service based on the availability of ambulances. Hospitals can escort the required equipment directly to the emergency location based on the user's emergency input type.

Making use of technology and reducing the dependency on human services, we believe this platform will save lives.

INTRODUCTION

With the current technology era where mostly everything runs on smartphones and applications, the need for quick and efficient services is important in almost every aspect, especially when it comes to medical services.

Patients are mostly having issues on searching for an ambulance, handling the locations and availability of the limited service in the time of emergency. The lack of such attention and information often has led to casualties. Thus the ambulance drivers must have proper information provided to them so that they won't get lost or find themselves searching for the exact location of the patient.

This made us think about how we could reduce this gap between a common person and the emergency services. This is the problem that our platform solves. When the user has to find ways to check the availability of the ambulance and also when the ambulance has to find the user's precise location in the quickest time possible our website provides the solution.

The website provided reduces response time of emergency services by removing any intermediary and directly approaching the emergency services to request an ambulance, reduce fraud calls and to allow ambulance driver to locate the victim easily by using GPS signal

MODULES

1. REGISTER (addRegistration.php)

- Register page is available for the patient with options of filling in details like full name, age, gender, email address, phone number, username, and password.

- The service providers will not be able to register through the website. They will have to get added after proper verification internally through the product team of the application. This has been done to prevent unwanted spammers from registering.

- After successful registration, the user is redirected to the login page.

2. LOGIN (index.php)

- The login page is available for the patient as well as the service provider.

- Once logged in, they are taken to their respective dashboards.

3. HOSPITAL DASHBOARD (hospitalIndex.php)

- The hospital dashboard allows tracking of the requests made to the specific hospital (implemented) and the ambulance status of all the active ambulances (to be implemented).

- Another tab will show the history of all the ambulances that had been sent out and to which location and for which emergency (to be implemented).

- The service provider will additionally also be able to mark ambulances as active again once they return to the corresponding hospital (to be implemented).

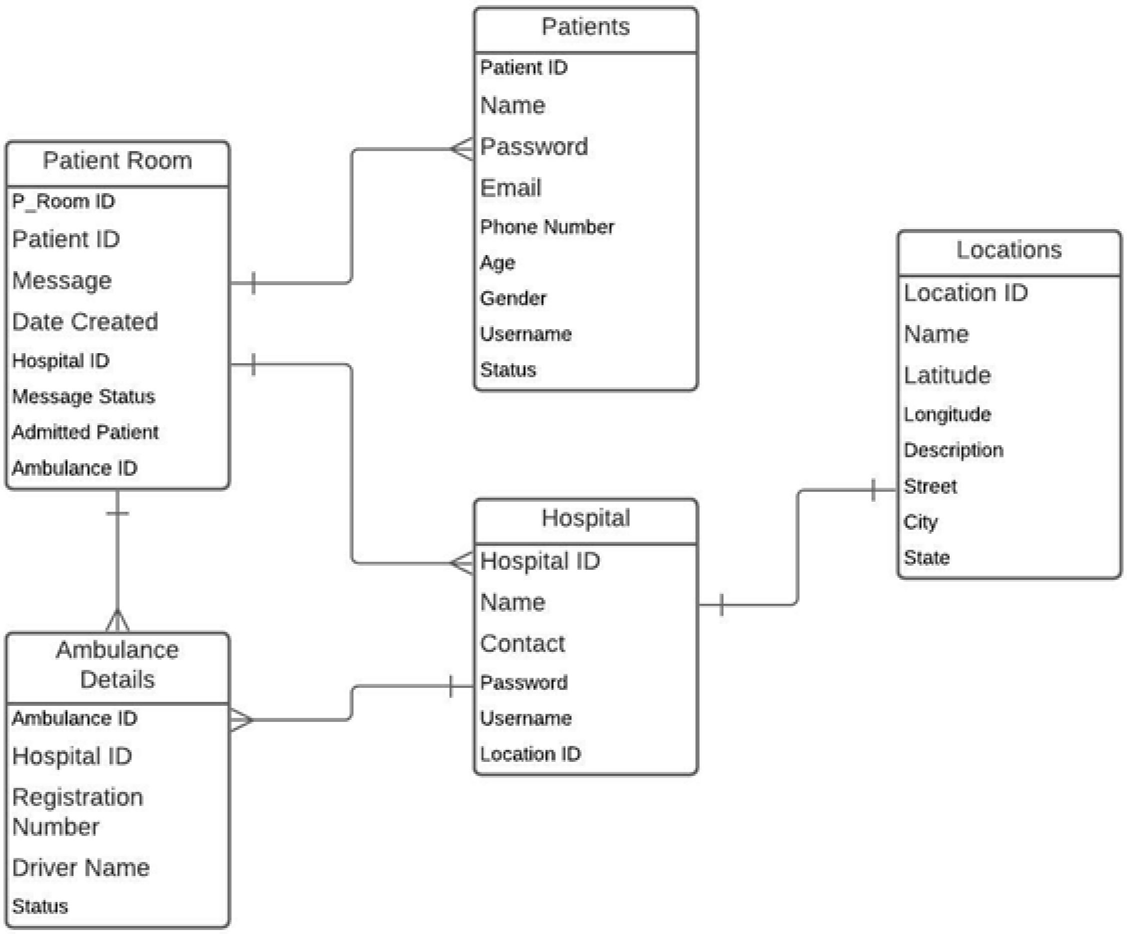
4. PATIENT DASHBOARD (patientIndex.php)

- The patient can select the type of emergency and the hospital.

- A map is presented to the patient showing hospitals in the vicinity (to be implemented) and the patient can select from the choices.

- The patient will then be able to view the request and the approval of their request.

ER DIAGRAM



ER diagrams stand for Entity Relationship Diagrams. They are responsible for showcasing the different elements of the system and what are the relationships between them.

For our web-based system named ‘Webulance’, there are 5 main Entities namely: Patients, Patient Room, Hospitals, Ambulance Details and Locations.

ENTITIES:

Patients Table:

Patients table stores the details of all the users that create an account for the web application. This stores important details such as name, email, age, gender and phone number and asks the user to create a username and password for their account. Every new user is assigned a patient ID. This field is the primary key for the Patient table thus no two values can have the same patient ID. This table is responsible for storing all the user details which will be fetched later to post to the hospital authorities.

Patients Room Table:

Patients Room table is a generic table that is responsible for keeping track of which ambulance is assigned to which patient. The importance of this table lies in the fact that there needs to be an intermediary to keep track of various connections between the hospitals and the patients. This table helps maintain in a systematic way the different connections so that no one ambulance is assigned to two different patients. This table contains important information like Patient ID, Hospital ID and Date created for proper storage of the information in a timely ordered fashion. It also contains the details of the patient admitted and given a patientRoom ID that is the primary key for this table.

Ambulance Details:

As the name suggests, this table is responsible for storing the information of all the ambulances that all the registered hospitals have. It contains important fields like Ambulance Registration Number and driver details. It also specifies which ambulance belongs to which hospital. This field helps sort out easily how many ambulances belong to that hospital. Each ambulance is given a unique Ambulance ID so that they can be differentiated easily.

Hospitals Table:

The Hospitals Table contains important information of all the genuine and verified hospitals entered into the database. This list specifies all the hospitals that provide ambulance services. It contains important information such as Name of the hospital, Contact Information and also the

Location Id that specifies the exact location of the hospital and will be needed to be sent to the patient when an ambulance will be assigned. Every hospital is assigned a unique hospital ID to differentiate between similar named hospitals.

Location Table:

This is a very important table that stores the location details of all the hospitals. This table is responsible for storing the geo location (latitude and longitude) of the hospitals that will later be needed to calculate the distance from the patient. It contains fields such as name of the hospital, its geo location and description.

RELATIONSHIPS:

The above mentioned 5 tables are all inter-related as and work together to connect hospital ambulance services to the desiring patients. The relationships between these tables are as follows:

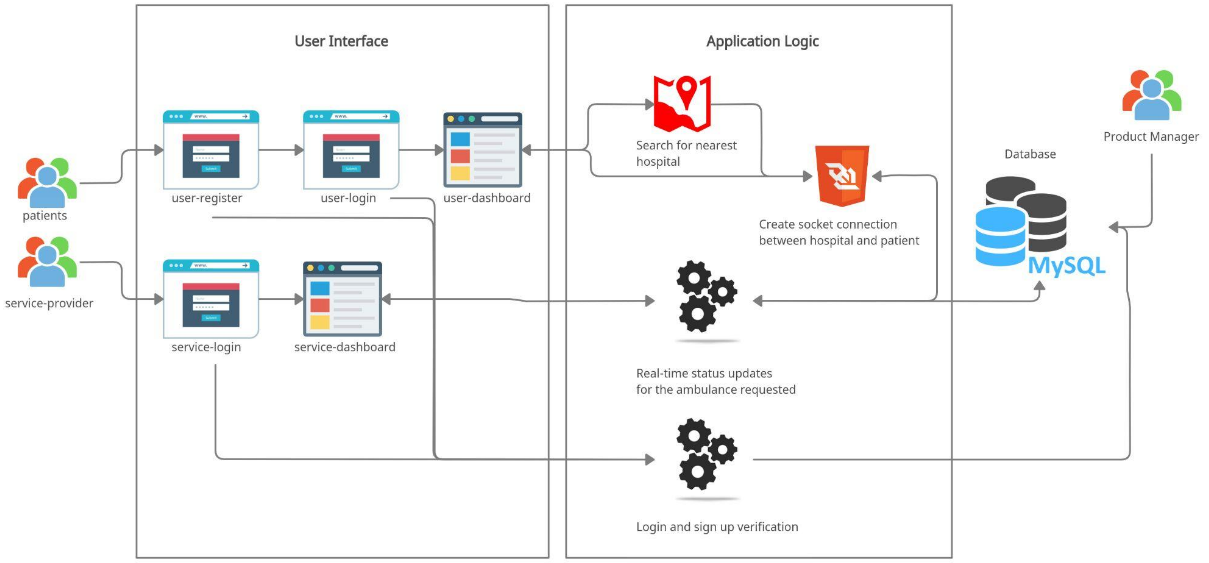
Hospital – Location Relationship: the hospital table has a one-to-one relationship with the location table as only hospitals can be located at one exact location.

Hospital-Ambulance Relationship: the hospital table has a one-to-many relationship with the ambulance table as one hospital can have many different ambulances but an ambulance can belong to only one hospital.

Patient_Room – Patient Relationship: there exists a one-to-many relationship between these two tables as the patient_room table has to maintain the records of all the patients that have been assigned an ambulance service. The patient_room table is a universal table and maintains all records based on a timeline.

Patient_Room – Hospital Relationship: there exists a one-to-many relationship between these two tables as the patient_room table has to maintain the records of all the patients that have been assigned an ambulance service. The patient_room table is a universal table and maintains all records based on a timeline.

PROCESS FLOW DIAGRAM


User Interface
Application Logic
patients
service-provider
user-register
user-login
user-dashboard
service-login
service-dashboard
Search for nearest hospital
Create socket connection between hospital and patient
Real-time status updates for the ambulance requested
Login and sign up verification
Database
MySQL
Product Manager

A system architecture is the conceptual model that defines the structure, behaviour, and more views of a system. It is responsible for describing how the user interacts with the application (through the User Interface) and how the application reacts and works (Application logic)

Description for the User Interface:

For our web-based application "Webulance" there are primarily two types of users: patients and hospital service providers.

1. Patients: these are the group of people who are responsible for creating a prior profile so that their required details such as Name, Mobile Number can be stored in the database. This way they can easily interact with the web application without the need of identity evaluation over and over again.

   a. If a Patients interact with the web application for the very first time, they are presented with the sign-up page. Here the user inputs their required information to secure a free account. Once the user creates an account, they can simply log into the website and wild be provided with the user dashboard.

   b. If a Patient has already secured a free account in the past, they have the added benefit of directly logging into the website. This way they will be presented with the user dashboard directly. Here the user can select the type to emergency and with one touch of a button request for an ambulance service from the hospitals nearby.

2. Service Providers: the second group of people are the service providers. This terminology is for the hospital and not one single group of persons alone. As the hospitals appoints someone to manage the web application, they will be in charge of accessing the websites and managing on application on their part.

   The service providers will not be presented with a signup page. This will be ensured by an authorized person who will be responsible for verifying the authenticity of the hospitals. Then the details of such ‘Genuine and Verified Hospitals’ will be added directly to the database. This is done to ensure that no false service provider registrations take place.

   After logging into the web application, the service providers are provided with the hospital dashboard.

Description of the Application Logic:

The Application Logic describes how the web application responds to the user requests, processes the data and provides the user with desired results. For our web-based application ‘Webulance’ the application logic goes as follows.

When a user calls for an ambulance, the system is responsible for capturing the user’s location. After this, a search for the nearest hospitals that have ambulance facility available with them is run to find the group of best hospitals that the user can rely upon. These hospitals are sorted on the basis of distance to reduce any time gap between the user calling action and the service being provided.

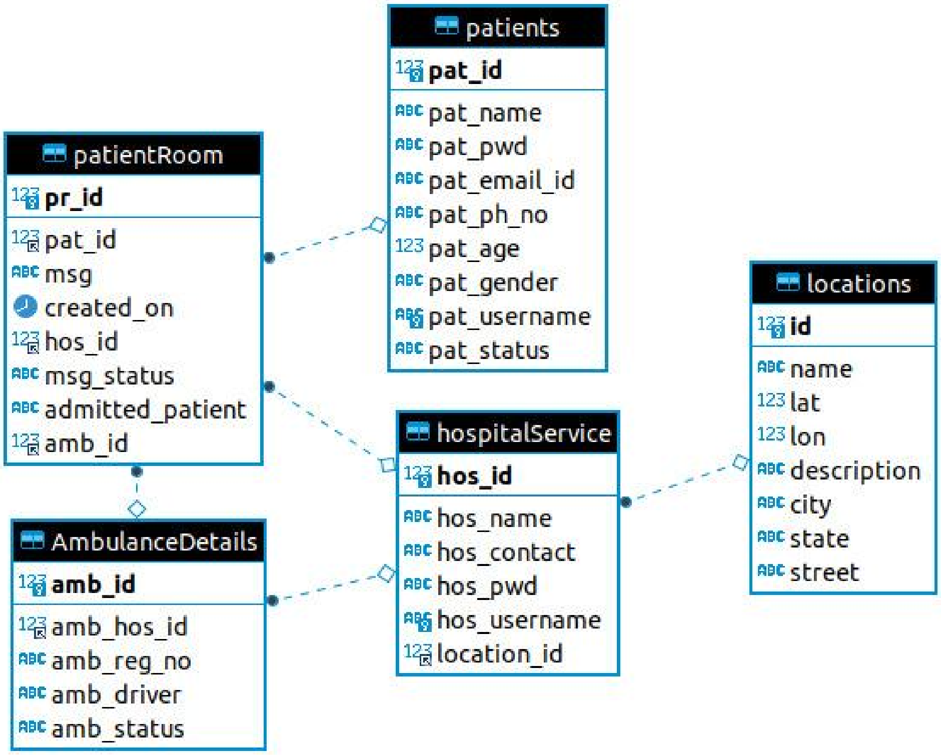
As soon as the nearest hospital is found, the system is responsible for creating a web socket between the user and the hospital. This web socket helps in two-way communication between the user and the hospital management.

Once the communication is established, the user details are sent to the hospital with all his details mentioned along with the current user location for easy track down. On the other hand, the user receives hospital details, details of the ambulance driver and the ambulance vehicle along with the distance and estimated time to the hospital.

‘Real Time Status Updates for the Ambulance Service’ are handled which gives the user better understanding of how the situation is being managed.

Thus, in this way, Webulance connects the hospital management directly to the User with one click of a button.

DATABASE DESIGN



Here, 5 tables are created namely, patents, pateintRoom, AmbulanceDetails, hospitalService and locations.

1. patients

In the patients table all the relevant patient details are stored such as id, name, password, username, email id, phone no., age, gender, username and status. Here id is the primary key of this table and thereby each and every patient has a unique id. Here status is an enum datatype attribute, whose value is either ‘enabled’ or ‘disabled’ depending on whether the user is genuine or spammer.

The patients table consists of 9 fields that are:

pat_id - number

pat_name - string

pat_pwd - string

pat_email_id - string

pat_ph_no - string

pat_age - number

pat_gender - character

pat_username - string

pat_status – string

2. patientsRoom

In the patientsRoom table, each patient is referenced through patient id. ‘pr_id’ is the primary key for the table. The msg attribute denotes the description of request made by the patient. ‘created_on’ attribute stores the time of creation of the request. ‘hos_id’ is the unique id of the hospital to which the request has been sent. ‘msg_status’ denotes the status of the request, that is, whether it is acknowledged, pending or declined. ‘amb_id’ is the referenced id of the ambulance sent for the emergency request.

The Patient Room table consists of 9 fields that are:

Pr_id – number

Pat_id – number

Msg – string

Created_on – date

Hos_id – number

Msg_status – string

Admitted_patient – string

Amb_id - number

3. AmbulanceDetails

The AmbulanceDetails table stores all the relevant details of the ambulance corresponding to a particular hospital referenced through ‘amb_hos_id’. ‘amb_reg_no’ and ‘amb_driver’ store the registration number of the ambulance and details of the driver respectively. ‘amb_status’ denotes whether the ambulance is currently in use or not.

The ambulance table consists of 5 fields that are:

amb_id - string

amb_hos_id – number

amb_reg_no – string

amb_driver – string

amb_status – string

4. hospitalService

In the hospitalService table, all the relevant hospital details are stored such as name, contact, administrator’s password and administrator’s username. Also the location of the hospital is stored in the location_id attribute which is referenced from the locations table.

The hospital table consists of 6 fields that are:

hos_id - number

hos_name – string

hos_contact – string

hos_pwd – string

hos_username – string

location_id - number

5. locations

In the locations table, the name of the hospital, it’s address, latitude, longitude, description of the location and name of the city, state and street is stored.

The location table consists of 8 fields that are:

Location_id - number

Name – string

Lat – number

Lon – number

Description – string

City – string

State – string

Street – string

CODE

Index.php-

```
<!DOCTYPE html>
<html>

<head>
  <title>Login page</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="./assets/css/main.css" />
</head>

<body>
  <section class="left-pane">
    <img src="./assets/vectors/location.svg" alt="Il 1" height="300" />
    <div class="banner">
      <h1 class="main-heading">Webulance</h1>
      <p>
        Get help <span class="highlight">instantly</span>,
        <span class="highlight sec">anywhere</span>
      </p>
    </div>
  </section>
  <section class="right-pane">
    <form class="card" name="loginForm" method="POST">
      <h2>Login</h2>
      <h6 id="msg">Please enter your credentials</h6>
      <input type="text" name="uid" id="uid" class="form-control" placeholder="Username"
required />
      <input type="password" name="pwd" id="pwd" class="form-control"
placeholder="Password" required />
      <div>
        <input type="checkbox" name="isadmin" id="isadmin" />
        <label for="isadmin">Login as service provider </label>
      </div>
      <button type="submit" name="login" onclick="return loginSubmit();">
```

```
        Login
      </button>
      <h6 class="footer">
        Don't have an account?
        <a href="./addRegistration.php">Create account</a>
      </h6>
    </form>
  </section>
  <script language="Javascript">
    const loginSubmit = () => {
      const cb = document.getElementById('isadmin')
      if (cb.checked == true) {
        document.loginForm.action = 'hospitalIndex.php'
      } else {
        document.loginForm.action = 'Displaylocation.php'
      }
      document.loginForm.submit()
      return true
    }
  </script>
</body>

</html>
```

addRegistration.php-

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Register</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet" href="./assets/css/register.css" />
  </head>

<head>
  <title>Add Registration</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="./assets/css/register.css" />
```

```
</head>

<body>
  <form name="registerForm" action="./insertRegistration.php" method="post" class="card">
    <h2 class="header">Sign Up as a Patient</h2>
    <h6 class="info" id="info">Please enter your details</h6>
    <input type="text" name="pname" id="pname" placeholder="Full Name" required />
    <div class="grp">
      <input type="number" name="page" id="page" placeholder="Age" min="0" max="100"
required />
      <select id="pgender" name="pgender">
        <option value="m">Male</option>
        <option value="f">Female</option>
        <option value="t">Others</option>
      </select>
    </div>
    <input type="email" name="pemail" id="pemail" placeholder="Email Address" required />
    <input type="number" name="pphone" id="pphone" placeholder="Mobile Number" required
/>
    <input type="text" name="puser" id="puser" placeholder="Username" required />
    <input type="password" name="ppwd" id="ppwd" class="form-control"
placeholder="Password" min="3" max="15" required />
    <input type="password" name="ppwd1" id="ppwd1" placeholder="Confirm Password"
min="3" max="15" required />
    <button type="button" name="register" onclick="return onSubmit();">
      Register
    </button>
  </form>
</body>
<script>
  const onSubmit = () => {
    let error = ''

    /** Validate password */
    const pass = document.querySelector('#ppwd')
    const pass2 = document.querySelector('#ppwd1')
    const regEx = new RegExp(/^(?=.*[A-Z])(?=.*[a-z])(?=.*\d).*$/)
    if (!pass.checkValidity() || !regEx.test(pass.value)) {
      error =
        'Password should have one lowercase, one uppercase, one digit (3-15 characters)'
```

```
} else if (pass.value !== pass2.value) {
  /** Check for matching passwords */
  error = 'Your passwords do not match'
}

/** Validate username */
const uname = document.querySelector('#puser')
if (!uname.checkValidity()) {
  error = 'Username should be between 3-15 characters'
}

/** Validate phone number */
const pNo = document.querySelector('#pphone')
if (!pNo.checkValidity()) {
  error = 'Mobile number should be 10 digits'
}

/** Validate email */
const email = document.querySelector('#pemail')
if (!email.checkValidity()) {
  error = 'Invalid email entered'
}

/** Validate age */
const age = document.querySelector('#page')
if (!age.checkValidity()) {
  error = 'Invalid age entered'
}

/** Validate Patient Name */
const fullName = document.querySelector('#pname')
if (fullName.value.split(/\W+/).length !== 2) {
  error = 'Please enter your full name'
}

if (error === '') document.registerForm.submit()
else {
  const info = document.querySelector('#info')
  info.textContent = error
  info.style.color = 'red'
```

```
    }
  }
</script>

</html>
```

patientIndex.php-

```
<?php

/*session_start();
if (isset($_SESSION["uid"])) {
    $uid = $_SESSION["uid"];
}

if (isset($_SESSION['errorMessage'])) {
    echo "<script type='text/javascript'>
            alert('" . $_SESSION['errorMessage'] . "');
          </script>";
    //to not make the error message appear again after refresh:
    unset($_SESSION['errorMessage']);
}

if (
    isset($_POST["login"]) && !empty($_POST["uid"])
    && !empty($_POST["pwd"])
) {
    $uid = $_POST['uid'];
    $pwd = $_POST['pwd'];

    include 'config.php';
    //set table name based on local or remote connection
    if ($connection == "local") {
        $t_patients = "patients";
    } else {
        $t_patients = "$database.patients";
    }

    try {
        $db = new PDO("mysql:host=$host", $user, $password, $options);
        $sql_select = "Select * from $t_patients where pat_username = '$uid' and pat_pwd =
'$pwd'";
        //echo "SQL Statement is : $sql_select <BR>";
        //echo "SQL Connection is : $host <BR>";


        $stmt = $db->prepare($sql_select);
        $stmt->execute();

        if ($rows = $stmt->fetch()) {
            //echo "SQL Statement is : $rows <BR>";
            $_SESSION['valid'] = TRUE;
            $_SESSION['uid'] = $_POST["uid"];
            $_SESSION["pwd"] = $_POST["pwd"];
        } else {
            echo '<script> alert("Invalid PatName or Password. Try again");</script>';
            header('refresh:0; url=./index.php');
            exit();
        }
    } catch (PDOException $e) {
        print "Error!: " . $e->getMessage() . "<br/>";
        die();
    }
}*/

?>

<?php
session_start();
if (isset($_SESSION["uid"])) {
$address = $_POST['location'];
$city = $_POST['city'];
$state = $_POST['state'];
$hos = $_POST['hos'];
$hospital = $_POST['destination'];
?>
<html>

<head>
```

```
  <title>Patients Dashboard</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="./assets/css/patients.css" />
</head>

<body>
  <nav>
    <h3>Webulance</h3>
    <a href="./logout.php">
      <img src="./assets/vectors/logout.svg" alt="Logout" height="20" />
      Logout
    </a>
  </nav>
  <main class="main">
    <section class="left-pane">
      <div class="info">
        <div>
          ① Select the <strong>Type of Emergency</strong> and the
          <strong>Hospital</strong>.
        </div>
        <div>② Send your request.</div>
      </div>
    </section>
    <section class="right-pane">
      <div class="message-card">
        <h3>Request Ambulance</h3>
        <select name="EmergencyType" id="EmergencyType">
          <option value="Serious Injuries">Serious Injuries</option>
          <option value="Cardiac Arrests">Cardiac arrests</option>
          <option value="Respiratory">Respiratory</option>
          <option value="Diabetics">Diabetics</option>
          <option value="Unconsciousness">Unconsciousness</option>
          <option value="Animal Bites">Animal bites</option>
          <option value="Infections">Infections</option>
        </select>
        <input type="text" name="patName" id="pat-name" style="padding: 8px"
placeholder="Enter Patient Name"  />
        <button onclick="transmitMessage()">Send</button>
        <script>
          // Create a new WebSocket.
          console.log("about to establish web socket connection");

          var socket = new WebSocket('ws://localhost:8080');

          socket.onopen = function(e) {
            console.log("Connection established!");
          };

          // Define the
          var HospitalName = '<?php echo $hospital; ?>';
          var Username = '<?php echo $_SESSION["uid"]; ?>';
          var Location = '<?php echo $address; ?>';
          var hos = '<?php echo $hos; ?>';

          function makeRequest(url, callback) {
            var request;
            if (window.XMLHttpRequest) {
              request = new XMLHttpRequest(); // IE7+, Firefox, Chrome, Opera, Safari
            } else {
              request = new ActiveXObject("Microsoft.XMLHTTP"); // IE6, IE5
            }
            request.onreadystatechange = function() {
              if (request.readyState == 4 && request.status == 200) {
                callback(request);
              }
            }
            request.open("GET", url, true);
            request.send();
          }

          function transmitMessage() {
            const patName = document.querySelector('#pat-name').value;
            if(patName.split(/\W+/).length !== 2 || patName.length == 0) {
              window.alert("Please enter the full name of the patient to be admitted");
              return;
            }
            makeRequest("get_ambulance.php?q=" + HospitalName + "&r=" + Username,
function(data) {
```

```
var data = JSON.parse(data.responseText);
const emptyHeader = document.querySelector('.info')
if (emptyHeader !== null) emptyHeader.remove()
const docElem = document.querySelector('.left-pane')
docElem.insertAdjacentHTML(
    'beforeend',
    `   <div class="card">
          <div class="bottom-row">
              <div class="field">
              <span class="bold">Driver Assigned:</span>
              <span>${data.driver_name}</span>
              </div>
              <div class="field">
              <span class="bold">Vehicle Registration:</span>
              <span>${data.ambulance_Registration}</span>
              </div>
          </div>
          <div class="bottom-row">
              <div class="field">
              <span class="bold">User Name:</span>
              <span>${data.PatientName}</span>
              </div>
              <div class="field">
              <span class="bold">Mobile Number:</span>
              <span>${data.PatientMob}</span>
              </div>
          </div>
          <div class="bottom-row">
              <div class="field">
              <span class="bold">Admitted Patient:</span>
              <span>${document.getElementById('pat-name').value}</span>
              </div>
              <div class="field">
              <span class="bold">Hospital Name:</span>
              <span>${'<?php echo $hospital; ?>'}</span>
              </div>
          </div>
          </div>`
)
```

```
                            docElem.classList.add('modify')

                            var message = {
                                type: "message",
                                text: document.getElementById('EmergencyType').value,
                                text1: hos,
                                text2: data.driver_name,
                                text3: data.ambulance_Registration,
                                text4: data.PatientName,
                                text5: data.PatientMob,
                                text6: Username,
                                text7: "Patient",
                                text8: Location,
                                text9: document.getElementById('pat-name').value,
                                text10: '<?php echo $hospital; ?>',
                                date: Date.now()
                            };
                            socket.send(JSON.stringify(message));
                        });
                    }

                    socket.onmessage = function(e) {
                        //alert(e.data);
                    }
                </script>
</body>

</html>
<?php } else {
    unset($_SESSION["uid"]);
    unset($_SESSION["pwd"]);
    unset($_SESSION["valid"]);

    //echo 'You have cleaned session';
    //header('Refresh: 2; URL = login.php');
    //header("Refresh: 2; Loocation: ./index2.php");
    header("Refresh: 1; URL = index.php");
    exit();
}  ?>
```

Displaylocation.php-

```
<?php

session_start();
if (isset($_SESSION["uid"])) {
    $uid = $_SESSION["uid"];
}

if (isset($_SESSION['errorMessage'])) {
    echo "<script type='text/javascript'>
            alert('" . $_SESSION['errorMessage'] . "');
          </script>";
    //to not make the error message appear again after refresh:
    unset($_SESSION['errorMessage']);
}

if (
    isset($_POST["login"]) && !empty($_POST["uid"])
    && !empty($_POST["pwd"])
) {
    $uid = $_POST['uid'];
    $pwd = $_POST['pwd'];

    include 'config.php';
    //set table name based on local or remote connection
    if ($connection == "local") {
        $t_patients = "patients";
    } else {
        $t_patients = "$database.patients";
    }

    try {
        $db = new PDO("mysql:host=$host", $user, $password, $options);

        $sql_select = "Select * from $t_patients where pat_username = '$uid' and pat_pwd =
'$pwd'";
        //echo "SQL Statement is : $sql_select <BR>";
        //echo "SQL Connection is : $host <BR>";
        $stmt = $db->prepare($sql_select);
        $stmt->execute();

        if ($rows = $stmt->fetch()) {
            //echo "SQL Statement is : $rows <BR>";
            $_SESSION['valid'] = TRUE;
            $_SESSION['uid'] = $_POST["uid"];
            $_SESSION["pwd"] = $_POST["pwd"];
        } else {
            echo '<script> alert("Invalid PatName or Password. Try again");</script>';
            header('refresh:0; url=./index.php');
            exit();
        }
    } catch (PDOException $e) {
        print "Error!: " . $e->getMessage() . "<br/>";
        die();
    }
}
?>

<?php
if (isset($_SESSION["uid"])) {
?>

    <!DOCTYPE html>

    <html lang="en">

    <head>
        <script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBwfZLkThCCQWYptWELcrp5d9uX
tgvywcc&callback=myMap"></script>
        <meta charset="utf-8" />
        <title>Location Fetch</title>
        <link rel="stylesheet" href="./assets/css/location.css">
        <script type="text/javascript">
            var hospitals;

            function makeRequest(url, callback) {
                var request;
                if (window.XMLHttpRequest) {
                    request = new XMLHttpRequest(); // IE7+, Firefox, Chrome, Opera, Safari
                } else {
                    request = new ActiveXObject("Microsoft.XMLHTTP"); // IE6, IE5
                }
                request.onreadystatechange = function() {
                    if (request.readyState == 4 && request.status == 200) {
                        callback(request);
                    }
                }
                request.open("GET", url, true);
                request.send();
            }

            var map;

            var center = new google.maps.LatLng(12.962180, 77.681427);

            var geocoder = new google.maps.Geocoder();
            var infowindow = new google.maps.InfoWindow();

            var directionsService = new google.maps.DirectionsService();
            var directionsDisplay = new google.maps.DirectionsRenderer();

            function init() {

                var mapOptions = {
                    zoom: 13,
                    center: center,
                    mapTypeId: google.maps.MapTypeId.ROADMAP
                }

                map = new google.maps.Map(document.getElementById("map_canvas"),
mapOptions);

                directionsDisplay.setMap(map);
                directionsDisplay.setPanel(document.getElementById('directions_panel'));

                // Detect user location
```

```
        if (navigator.geolocation) {

          navigator.geolocation.getCurrentPosition(function(position) {

            var userLocation = new google.maps.LatLng(position.coords.latitude,
position.coords.longitude);

            //alert(userLocation);

            geocoder.geocode({
              'latLng': userLocation
            }, function(results, status) {
              //alert(status)
              if (status == google.maps.GeocoderStatus.OK) {
                document.getElementById('start').value = results[0].formatted_address;
                displayLocation(results[0].formatted_address);
                var resaddress = results[0].formatted_address.split(",");
                var count = 0,
                  state, city, street = "";

                for (i = (resaddress.length - 2); i >= 0; i--) {
                  count += 1;
                  if (count == 1) {
                    state = resaddress[i];
                  } else if (count == 2) {
                    city = resaddress[i];
                  } else {
                    street = street + resaddress[i];
                  }
                }
                document.getElementById('city').value = city;
                document.getElementById('state').value = state;
                makeRequest("get_disttime.php?q=" + street.trim() + "&r=" + city.trim() +
"&s=" + state.trim(), function(data) {
                  var data = JSON.parse(data.responseText);
                  let distances = [];
                  let times = [];
                  let names = [];
                  hospitals = data.hospital;
```

```
    for (const [key, value] of Object.entries(data.distance)) {
      distances.push(value);
    }
    for (const [key, value] of Object.entries(data.time)) {
      times.push(value);
    }
    for (const [key, value] of Object.entries(data.hospital)) {
      names.push(value);
    }

    var selectBox = document.getElementById('destination');
    let optionString = '';
    for (let i = 0; i < names.length; i++) {
      optionString = `${names[i]} - ${distances[i]} - ${times[i]}`;
      addOption(selectBox, optionString, names[i]);
      optionString = '';
    }
  });
  // alert(city);
  // alert(state);
  // alert(street);
}

function makeRequest1(url, callback) {
  var request;
  if (window.XMLHttpRequest) {
    request = new XMLHttpRequest(); // IE7+, Firefox, Chrome, Opera, Safari
  } else {
    request = new ActiveXObject("Microsoft.XMLHTTP"); // IE6, IE5
  }
  request.onreadystatechange = function() {
    if (request.readyState == 4 && request.status == 200) {
      callback(request);
    }
  }
  request.open("GET", url, true);
  request.send();
}
```

```
            });


        }, function() {
            alert('Geolocation is supported, but it failed');
        });
    }

    makeRequest('get_locations.php', function(data) {

        var data = JSON.parse(data.responseText);
        var selectBox = document.getElementById('destination');

        for (var i = 0; i < data.length; i++) {
            displayLocationHospital(data[i]);
        }
    });
}

function addOption(selectBox, text, value) {
    var option = document.createElement("OPTION");
    option.text = text;
    option.value = value;
    selectBox.options.add(option);
}

function displayLocation(location) {

    var content = '<div class="infoWindow"><strong>' + "your position" + '</strong>' +
        '<br/>' + location + '</div>';

    console.log(location)
    geocoder.geocode({
        'address': location
    }, function(results, status) {
        if (status == google.maps.GeocoderStatus.OK) {
            var marker = new google.maps.Marker({
                map: map,
                position: results[0].geometry.location,
                title: "My location"
```

```
                        });

                        google.maps.event.addListener(marker, 'click', function() {
                            infowindow.setContent(content);
                            infowindow.open(map, marker);
                        });
                    }
                });
            }

            function displayLocationHospital(location) {

                var content = '<div class="infoWindow"><strong>' + location.name + '</strong>' +
                    '<br/>' + location.address +
                    '<br/>' + location.description + '</div>';

                if (parseInt(location.lat) == 0) {
                    geocoder.geocode({
                        'address': location.address
                    }, function(results, status) {
                        if (status == google.maps.GeocoderStatus.OK) {
                            var marker = new google.maps.Marker({
                                map: map,
                                position: results[0].geometry.location,
                                title: location.name
                            });

                            google.maps.event.addListener(marker, 'click', function() {
                                infowindow.setContent(content);
                                infowindow.open(map, marker);
                            });
                        }
                    });
                } else {
                    var position = new google.maps.LatLng(parseFloat(location.lat),
parseFloat(location.lon));
                    var marker = new google.maps.Marker({
                        map: map,
                        position: position,
                        title: location.name
            });

            google.maps.event.addListener(marker, 'click', function() {
                infowindow.setContent(content);
                infowindow.open(map, marker);
            });
        }
    }

    function gethos() {
        var hospital = document.getElementById("destination").value;
        console.log(hospital);
        for (const [key, value] of Object.entries(hospitals)) {
            if (value == hospital) {
                document.getElementById("hos").value = key;
            }
        }
    }

    // function calculateRoute() {

    //     var start = document.getElementById('start').value;
    //     var destination = document.getElementById('destination').value;

    //     if (start == '') {
    //         start = center;
    //     }

    //     var request = {
    //         origin: start,
    //         destination: destination,
    //         travelMode: google.maps.DirectionsTravelMode.DRIVING
    //     };
    //     directionsService.route(request, function (response, status) {
    //         if (status == google.maps.DirectionsStatus.OK) {
    //             directionsDisplay.setDirections(response);
    //         }
    //     });
    // }
</script>
```

```
  </head>

  <body onload="init();">
    <form id="services" method="POST" action="patientsIndex.php">
      <h3>Updating Your Location</h3>
      <span>Location:</span><input name="location" type="text" id="start" readonly />
      <span>City:</span><input name="city" type="text" id="city" readonly />
      <span>State:</span><input name="state" type="text" id="state" readonly />
      <span>Destination:</span><select name="destination" id="destination"
onchange="return gethos()"></select>
      <input type="text" style="display:none" readonly id="hos" name="hos" />
      <button type="submit">Enter Patient Details</button>
    </form>
    <section id="sidebar">
      <div id="directions_panel"></div>
    </section>
    <section id="main">
      <div id="map_canvas" style="width: 70%; height: 500px;"></div>
    </section>
  </body>

  </html>
<?php } ?>
```

hospitalIndex.php-

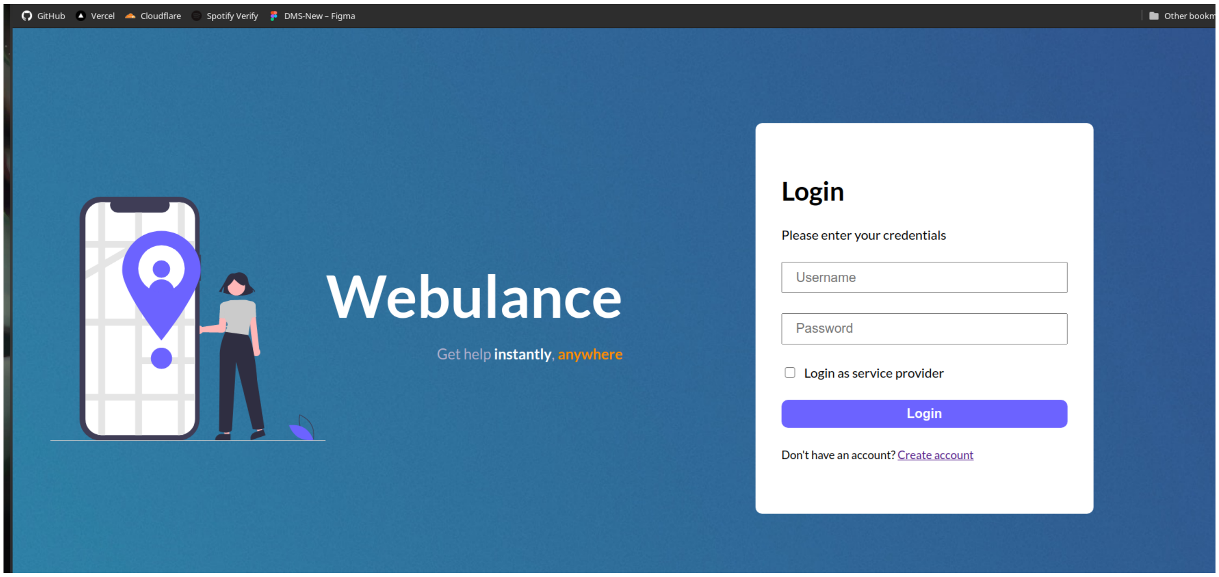
```
<?php

session_start();
if (isset($_SESSION["uid"])) {
  $uid = $_SESSION["uid"];
}

if (
  isset($_POST["login"]) && !empty($_POST["uid"])
  && !empty($_POST["pwd"])
) {
  $uid = $_POST['uid'];
  $pwd = $_POST['pwd'];
  include './config.php';
  //set table name based on local or remote connection
  if ($connection == "local") {
    $t_hospital = "hospitalService";
  } else {
    $t_hospital = "$database.hospitalService";
  }

  try {
    $db = new PDO("mysql:host=$host", $user, $password, $options);
    //echo "Database connected successfully <BR>";

    $sql_select = "Select * from $t_hospital where hos_username = '$uid' and hos_pwd = '$pwd'";

    $stmt = $db->prepare($sql_select);
    $stmt->execute();

    if ($rows = $stmt->fetch()) {
      $_SESSION['valid'] = TRUE;
      $_SESSION['uid'] = $uid;
      $_SESSION["pwd"] = $pwd;
      $_SESSION["isadmin"] = TRUE;
    } else {
      //echo '<script>alert("Invalid PatName or Password. Try again")</script>';
      echo '<script>alert("Invalid PatName or Password. TTTTTry again")</script>';

      header('refresh:0; url=./index.php');
      exit();
    }
  } catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
  }
}

?>

<?php
if (isset($_SESSION["uid"])) {
```

```
?>
 <html>

 <head>
  <title>Register</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="./assets/css/hospital.css" />
 </head>

 <body>
  <nav>
   <h3>Webulance</h3>
   <a href="./logout.php">
    <img src="./assets/vectors/logout.svg" alt="Logout" height="20" />
    Logout
   </a>
  </nav>
  <section class="main">
   <h1 class="empty">No active requests</h1>
  </section>
  <script>
   // Create a new WebSocket.
   console.log('about to establish web socket connection')

   var socket = new WebSocket('ws://localhost:8080')

   var Username = '<?php echo $_SESSION["uid"]; ?>';
   var message = {
    type: "message",
    text6: Username,
    text7: "Hospital"
   };

   socket.onopen = function(e) {
    console.log('Connection established!')
    socket.send(JSON.stringify(message))
   }

   // Define the
```

```
function transmitMessage() {
  //socket.send(JSON.stringify(message))
}

socket.onmessage = function(e) {
  console.log(e.data)
  var object = JSON.parse(e.data)
  const emptyHeader = document.querySelector('.empty')
  if (emptyHeader !== null)
    emptyHeader.remove()
  const docElem = document.querySelector('.main')
  docElem.insertAdjacentHTML('beforeend', `<div class="card">
  <div class="top-row">
    <div class="field">
      <span class="bold">Type of Emergency:</span>
      <span>${object.text}</span>
    </div>
    <div class="field">
      <span class="bold">Hospital Name:</span>
      <span>${object.text10}</span>
    </div>
  </div>
  <div class="bottom-row">
    <div class="field">
      <span class="bold">Driver Assigned:</span>
      <span>${object.text2}</span>
    </div>
    <div class="field">
      <span class="bold">Vehicle Registration:</span>
      <span>${object.text3}</span>
    </div>
  </div>
  <div class="bottom-row">
    <div class="field">
      <span class="bold">User Name:</span>
      <span>${object.text4}</span>
    </div>
    <div class="field">
      <span class="bold">Mobile Number:</span>
      <span>${object.text5}</span>
     </div>
    </div>
    <div class="bottom-row">
     <div class="field">
      <span class="bold">Patient Name:</span>
      <span>${object.text9}</span>
     </div>
     <div class="field">
      <span class="bold">Location:</span>
      <span>${object.text8}</span>
     </div>
    </div>
   </div>`)
   }

   document.getElementById('myButton').onclick = function() {
    location.href = 'http://3007f8e97f51.ngrok.io/MapsBackUp.html'
   }
  </script>
 </body>

 </html>
<?php
}
?>
```

SCREEN SHOTS

Login (Patient and Service Provider)



Login

Please enter your credentials

Username

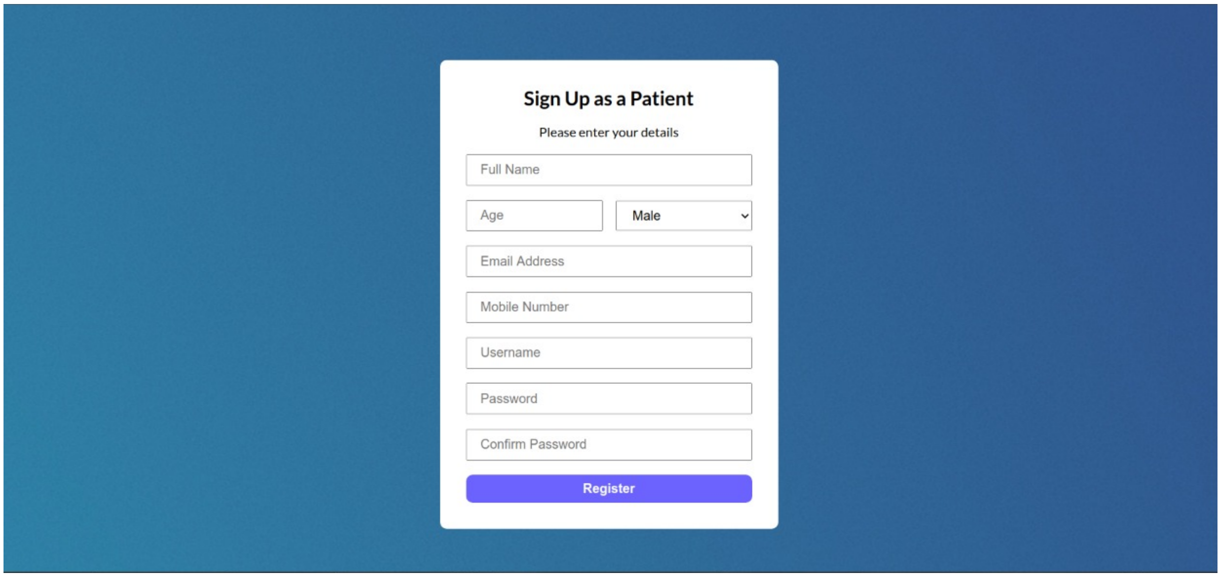
Password

□ Login as service provider

Login

Don't have an account? Create account

Register Patient



Sign Up as a Patient

Please enter your details

Full Name

Age

Male

Email Address

Mobile Number

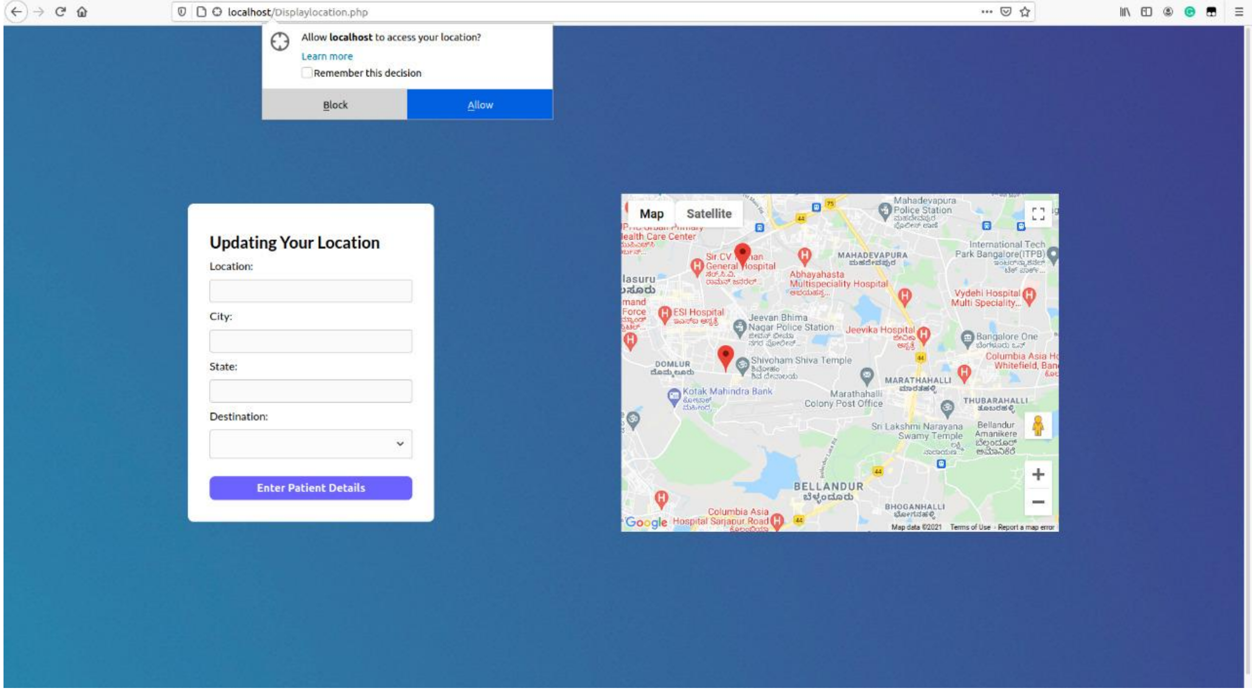
Username

Password

Confirm Password

Register

Patient Dashboard



localhost/Displaylocation.php

Allow **localhost** to access your location?

Learn more

□ Remember this decision

Block | Allow

Updating Your Location

Location:

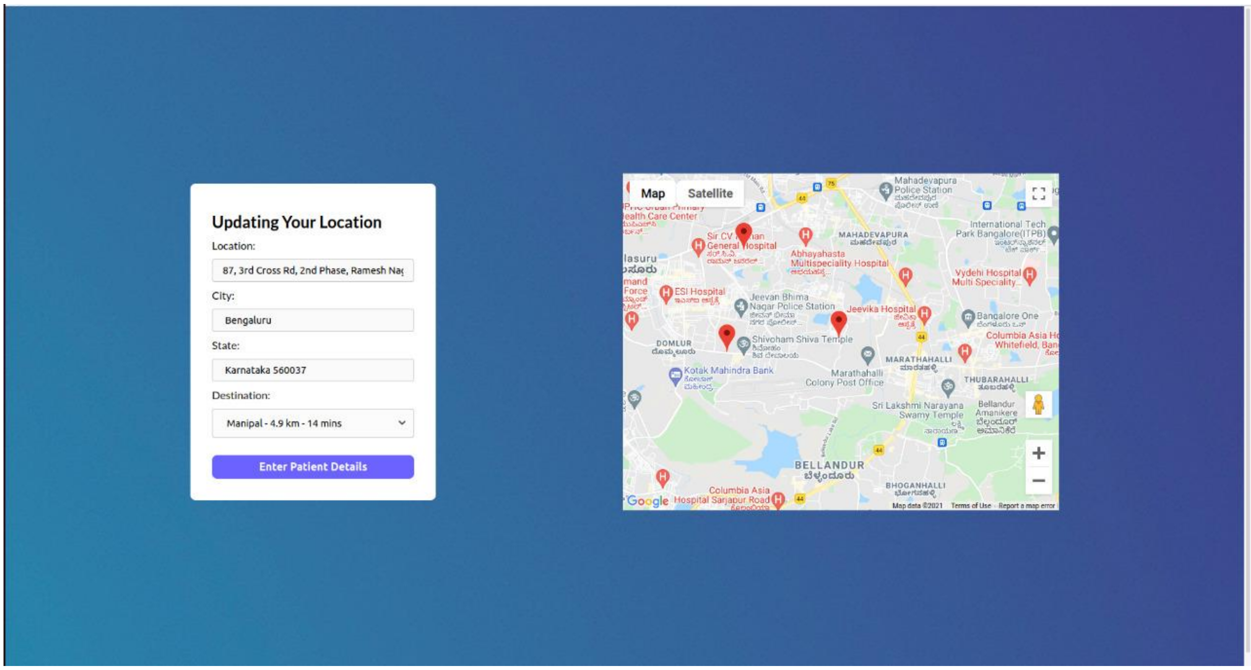
City:

State:

Destination:

Enter Patient Details

Map
Satellite
Mahadevapura Police Station
MAHADEVAPURA
International Tech Park Bangalore(ITPB)
Sir CV Raman General Hospital
Abhayahasta Multispeciality Hospital
Vydehi Hospital Multi Speciality...
ESI Hospital
Jeevan Bhima Nagar Police Station
Jeevika Hospital
Bangalore One
Columbia Asia Hospital Whitefield, Bangalore
DOMLUR
Shivoham Shiva Temple
MARATHAHALLI
Kotak Mahindra Bank
Marathahalli Colony Post Office
THUBARAHALLI
Sri Lakshmi Narayana Swamy Temple
Bellandur Amanikere
BELLANDUR
BHOGANHALLI
Columbia Asia Hospital Sarjapur Road
Google
Map data ©2021
Terms of Use
Report a map error



Updating Your Location

Location: 87, 3rd Cross Rd, 2nd Phase, Ramesh Na

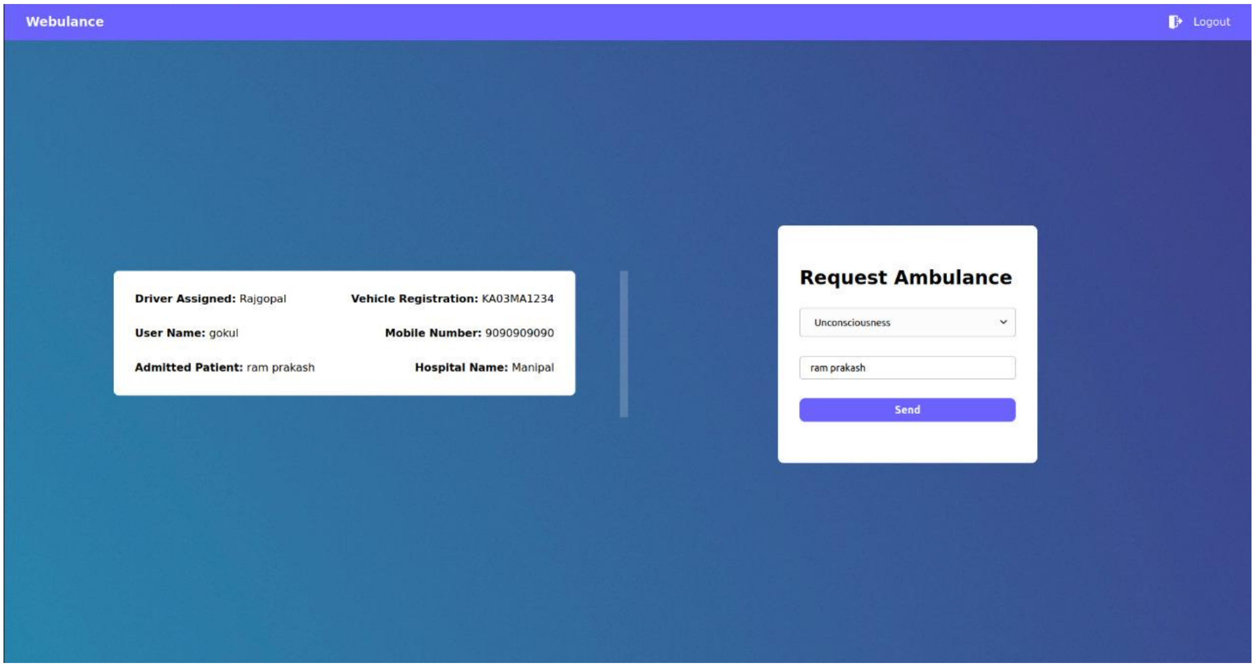
City: Bengaluru

State: Karnataka 560037

Destination: Manipal - 4.9 km - 14 mins

Enter Patient Details

Map
Satellite
Mahadevapura Police Station
International Tech Park Bangalore(ITPB)
Sir CV Raman General Hospital
MAHADEVAPURA
Abhayahasta Multispeciality Hospital
Vydehi Hospital Multi Speciality
ESI Hospital
Jeevan Bhima Nagar Police Station
Jeevika Hospital
Bangalore One
Columbia Asia Hospital Whitefield
DOMLUR
Shivoham Shiva Temple
MARATHAHALLI
Kotak Mahindra Bank
Marathahalli Colony Post Office
THUBARAHALLI
Sri Lakshmi Narayana Swamy Temple
Bellandur Amanikere
BELLANDUR
BHOGANHALLI
Columbia Asia Hospital Sarjapur Road
Google
Map data ©2021
Terms of Use
Report a map error



Webulance

Logout

Driver Assigned: Rajgopal

Vehicle Registration: KA03MA1234

User Name: gokul

Mobile Number: 9090909090

Admitted Patient: ram prakash

Hospital Name: Manipal

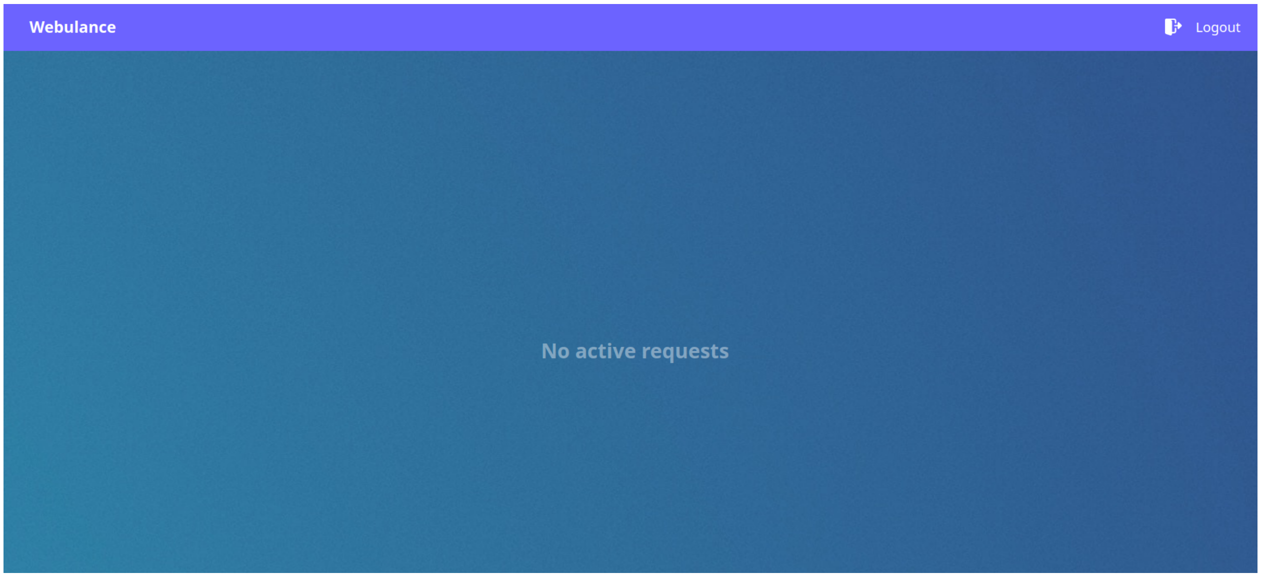
Request Ambulance

Unconsciousness

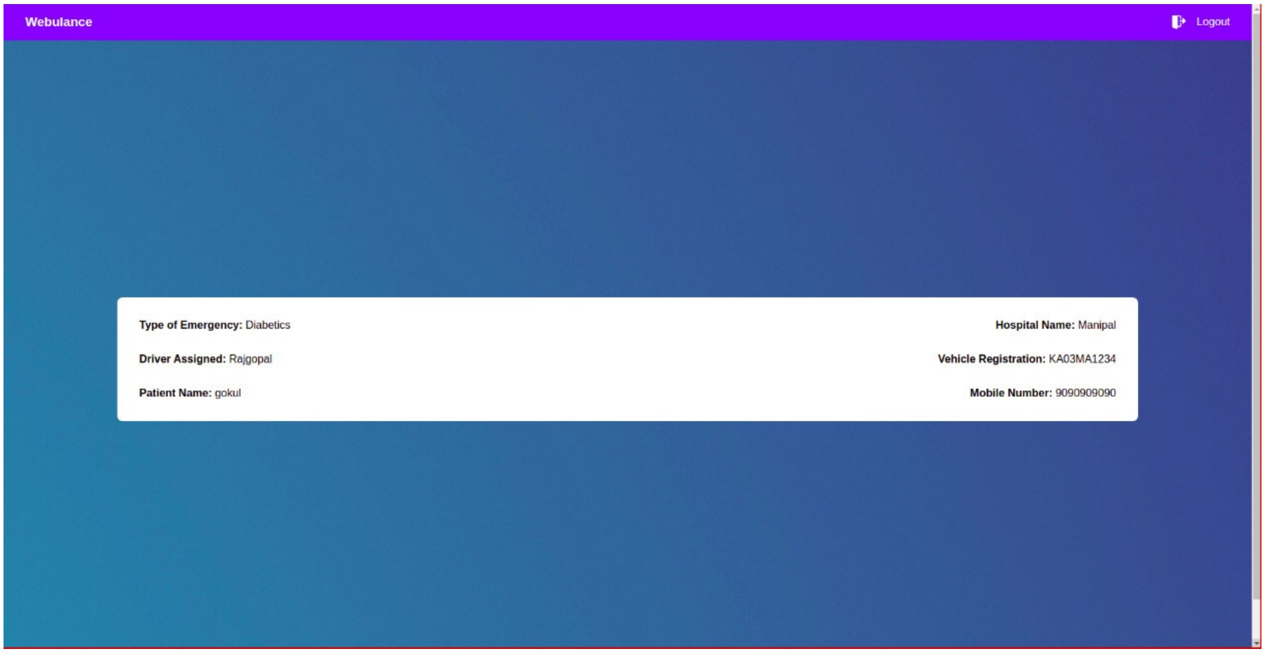
ram prakash

Send

Hospital Dashboard



No active requests



Type of Emergency: Diabetics

Hospital Name: Manipal

Driver Assigned: Rajgopal

Vehicle Registration: KA03MA1234

Patient Name: gokul

Mobile Number: 9090909090

CONCLUSION

Those old days are gone where we could call the ambulance and ask them to come to the user’s location to take the patient to the hospital. As everything is available at a user’s fingertips, implementing it would help save a lot of patients' lives as the ambulance can be booked and availability of different hospitals can be checked at any location from the user’s location.

In the future this application can be upgraded to the next level by making it more interactive in such a way that during the time of the users registration a form will be given for the user’s to fill in all their medical details which can be stored on the cloud, then once the patient is in the ambulance all the medical details that was filled by the patient at the time of their registration along with the present patient condition will be sent to the hospital even before the ambulance reaches the hospital so that the doctors can be ready to treat the patients and same many life.

REFERENCES

1. https://www.php.net/
2. https://www.mysql.com/
3. https://www.w3schools.com/
4. Paul Deitel, Harvey Deitel, Abbey Deitel, Internet & World Wide Web - How to Program, 5th edition, Pearson Education, 2012.
5. Kogent Learning Solutions Inc, Web Technologies Black Book, Dream Tech press, 2013.
6. Brad Dayley, Brendan Dayley, and Caleb Dayley , Node.js, MongoDB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications, 2nd Edition, Pearson Education, 2018
7. Lindsay Bassett, Introduction to JavaScript Object Notation, 1st Edition, O’Reilly Media, 2015

8. Fritz Schneider, Thomas Powell , JavaScript – The Complete Reference, 3rd Edition, McGraw Hill, 2017

9. Steven Holzener , PHP – The Complete Reference, 1st Edition, Mc-Graw Hill, 2017

10. Sandeep Kumar Patel, Developing Responsive Web Applications with AJAX and JQuery, Packt Publications, 2014