

Helpify

**A Project Report Submitted by
Hittanshu Upadhyay, Yuvika Singh
*Under the Guidance of***

Pranav More

***In partial fulfillment for the award of the
Degree of***

**BACHELORS OF TECHNOLOGY
Computer Engineering**

At



**SCHOOL OF TECHNOLOGY MANAGEMENT AND
ENGINEERING, NAVI MUMBAI**

APRIL 2023

DECLARATION

I, Hittanshu Upadhyay, Roll No. A045 B.Tech (CE), IV semester understand that plagiarism is defined as anyone or combination of the following:

1. Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.
2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did write what. (Source: IEEE, The institute, Dec. 2004)
4. I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.
5. I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidents of plagiarism in this body of work.

Signature of the Student:

Name: Hittanshu Upadhyay

Roll No. A045

Place:

Date: 07/04/2023

CERTIFICATE

This is to certify that the project entitled “Helpify” is the bonafide work carried out by Hittanshu Upadhyay of B.Tech, School of Technology, Management & Engineering, NMIMS, Navi Mumbai, during the IVth semester of the academic year 2022-23, in partial fulfillment of the requirements for the award of the Degree of Bachelors of Engineering as per the norms prescribed by NMIMS. The project work has been assessed and found to be satisfactory.

Pranav More

Examiner 1

Examiner 2

Table of Contents

CHAPTER NO.	TITLE	PAGE NO.
	List of Figures	i
	Abbreviations	ii
1.	INTRODUCTION	iii
	1.1 Project Overview	
	1.2 Hardware Specification	
	1.3 Software Specification	
2.	TECHNOLOGY USED	vi
3.	ANALYSIS & DESIGN	vii
	3.1 Analysis	
	3.2 Design	
4.	OUTPUT SCREENSHOTS	xi
5.	DATABASE SCHEMA	xiii
6.	CONCLUSION & FUTURE SCOPE	xiv
6.1	- CONCLUSION	
6.2	- FUTURE SCOPE	
7.	REFERENCES & APPENDIX	

List of Figures

CHAPTER NO.	TITLE	PAGE NO.
3.	Analysis and Design	vii
	Fig 3.1 - NGO registration, validation and php code	
	Fig 3.2 - Find NGO angular table, HTML & PHP code	
	Fig 3.3 - NGO-NGO angular table, HTML & PHP code	
	Fig 3.4 Contact us validation, HTML & PHP code	
	Fig 3.5 User registration validation, HTML & PHP code	
4.	Output Screenshots	xi
	Fig 4.1 - Home page	
	Fig 4.2 - Find your NGO page	
	Fig 4.3 -NGO-NGO page	
	Fig 4.4 - Contact us page	
	Fig 4.5 - Login/registration page	
5.	DATABASE SCHEMA	xiii
	Fig 5.1 - Database Schema	

List of Tables

CHAPTER NO.	TITLE	PAGE NO.
5.	Table 1.1 NGO Details	
	Table 1.2 Query Details	
	Table 1.2 User Details	

Abbreviations

Abbreviation	Description
HTML	Hyper Text Markup Language
SQL	Structured Query Language
CSS	Cascading Style Sheet
JS	Java Script

Chapter 1

Introduction

1.1 Project Overview

Vision

To create a user-friendly website that helps people stay connected to donation drives and donating surplus commodities that they have to the needy. We aim at giving a bigger, better platform to the NGOs not known to the masses in detail. It helps them to receive donations but also donate themselves to other organizations as well. It's a mediating party website for donors to NGOs.

Mission

Our mission is to provide a helpful platform for any non-profit organization to register and display the resources they're lacking. Which enables them to connect to donors reaching out to them digitally or even make payments. Searching with respect to their locality/facilities. Not only this, it exercises the practice of organizations helping each other out by donating goods that any one of them has in excess. We aim at resources being used efficiently in the right hands, for the right place.

Feature List

Current functionality of the app includes -

1. Find your NGO!

This page will be available to the donors/users after signing up or logging in. It displays a table connected to the database featuring all of the organizations along with their types about all the resources they're lacking and where they're situated. Users can search according to their requirements and visit the NGO's site. Or even make a payment through an online link.

2. NGO-NGO

This is a similar design to Find your NGO!, but for organizations themselves. It becomes available after registration.

3. User Sign-up/login

Located on the same page, and validated completely. This page allows interested donors to sign up to our website, by filling in some details and getting added to our database. Once you've signed up, the page will allow to log you in by matching the existing details, if present.

4. Contact us

This page allows any person using the website to get in touch with our officials personally by submitting a query.

1.2 Hardware Specifications

XAMPP Software package: Requires a machine Processor: 1.8 GHz dual-core or higher processor, RAM: 2 GB or higher, Hard disk space: 1 GB of free disk space or more, Display: 1024 x 768 screen resolution or higher, Operating system: Windows, macOS, or Linux Also includes a php script connected within for database connectivity.

Network: Reliable internet connectivity to ensure smooth communication between the server and clients accessing the website.

1.3 Software Specification

Operating System: The server should be running a compatible operating system, such as Linux, Windows, or macOS, that supports the installation and operation of XAMPP.

XAMPP: The latest stable version of XAMPP installed on the server, along with necessary dependencies and libraries for building server-side applications.

Front-end Technologies: HTML, CSS, and Bootstrap for building the user interface and responsive design of the website. JavaScript for client-side interactivity and dynamic Content. Angular JS for validation and table searching filter.

Back-end Technologies: PHP runs on the server-side, which means that it is executed on the server before the web page is sent to the client. This allows PHP to process and manipulate data on the server before sending it to the client, making it easier to build dynamic web applications. PHP has built-in functions and extensions for connecting to databases such as MySQL, PostgreSQL, and Oracle.

Chapter 2

Technologies used

HTML, CSS, Bootstrap: HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and Bootstrap (a popular front-end framework) are used for designing and building the user interface of the website.

JavaScript: A programming language used for client-side interactivity, form validation, and dynamic content generation on the website.

AngularJS: AngularJS is a popular open-source JavaScript-based front-end web application framework used to build dynamic web applications. It was developed by Google and is now maintained by a community of developers.

XAMPP: XAMPP is a free and open-source software package that includes several components needed to run web applications on a local computer. The name XAMPP stands for Cross-platform, Apache, MySQL, PHP, and Perl.

PHP: PHP (Hypertext Preprocessor) is a server-side scripting language used to create dynamic web pages and web applications. It is an open-source, cross-platform scripting language that is widely used for web development.

Chapter 3

Analysis and design

3.1 Analysis

The analysis phase involved gathering and analyzing the requirements from the Organizations and other donors who were interested. This included understanding the functionalities and features that the users would find necessary to be included in a website and make use of their resources, such as searching with locality, required facilities, and payment links. Additionally, technical requirements such as the use of XAMPP for server-side development, PHP for database management, and HTML, CSS with Bootstrap for front-end development were identified. The team also conducted a thorough review of a pre-existing website that was used as a reference for design inspiration, analyzing its layout, features, and user experience to gather insights for the new non-profit website.

3.2 Design

The design phase involved creating a comprehensive design plan for the website, incorporating the requirements identified during the analysis phase and drawing inspiration from the pre-existing website reference. The following design aspects were considered:

Data Modeling: A data model was designed to define the structure of the database, including the entities and their relationships, such as name of organization, type, and other relevant data. This ensured efficient storage and retrieval of data from the XAMPP database.

System Architecture: The system architecture was designed using PHP for server-side development. The front-end was designed using HTML, CSS with Bootstrap, and JavaScript for interactivity and dynamic content generation.

User Interface (UI) Design: The UI design incorporated a clean and visually appealing layout with easy-to-navigate menus, forms for data input, and visual elements that enhanced the user experience. The use of Bootstrap ensured responsiveness.

Interaction Design: Interaction design was focused on providing an intuitive and seamless user experience. This included designing forms with input validation, error handling, and feedback mechanisms to guide users through the process of calculating BMI and managing their measurements.

3.3 Codes

The screenshot shows a modal window titled "NGO Registration Form". The form contains the following fields:

- Organization Name:** An input field.
- Type of NGO:** A dropdown menu showing "Please select an option".
- Address:** A text area.
- Contact Details:** An input field.
- Missions and Objectives:** A text area.
- Lack of Resources:** A text area.
- Payment Link:** A text area.
- Link to Website:** An input field containing "https://www.example.com".
- Governing Document:** A section with a "Choose File" button (which shows "No file chosen") and a "Submit" button.

Figure 3.1
Screenshot of NGO Registration form with ANGULAR JS validation

```
<div id="overlay-content" ng-app="myApp" ng-controller="formCtrl">
    <h1 style="display:inline-block; margin-right:10px;">NGO Registration
Form</h1>
    <button class="close-button" onclick="off()" style="display:inline-block; background:transparent; border:none;
float:right;">&times;</button>
```

```
<form name="registrationForm" action='ngoregistration.php'
method="post" ng-submit="submitForm(registrationForm.$valid)" novalidate>
    <div class="form-row">
        <label for="orgName">Organization Name</label>
        <input type="text" id="orgName" name="orgName"
ng-model="ngo.orgName" required>
        <label for="type">Type of NGO</label>
        <select id="type" name="type" ng-model="ngo.type" required>
            <option value="" disabled selected>Please select an
option</option>
            <option value="Charity">Charity</option>
            <option value="Non-profit">Non-profit</option>
            <option value="Advocacy">Advocacy</option>
            <option value="International">International</option>
            <option value="Community-based">Community-based</option>
            <option value="Research and Education">Research and
Education</option>
            <option value="Religious">Religious</option>
            <option value="Other">Other</option>
        </select>
    </div>
    <div class="form-row">
        <label for="address">Address</label>
        <textarea id="address" name="address" ng-model="ngo.address"
required></textarea>
    </div>
    <div class="form-row">
        <label for="contact">Contact Details</label>
        <input type="text" id="contact" name="contact"
ng-model="ngo.contact" required>
    </div>
    <div class="form-row">
        <label for="mission">Missions and Objectives</label>
        <textarea id="mission" name="mission" ng-model="ngo.mission"
required></textarea>
    </div>
    <div class="form-row">
        <label for="resources">Lack of Resources</label>
        <textarea id="resources" name="resources"
ng-model="ngo.resources"></textarea>
    </div>

```

```

        </div>
        <div class="form-row">
            <label for="payment-link">Payment Link</label>
            <input type="text" id="payment_link" name="payment_link"
ng-model="ngo.paymentLink">
            <label for="website">Link to Website</label>
            <input type="text" id="website" name="website"
ng-model="ngo.website" placeholder="https://www.example.com">
            <label for="document">Governing Document</label>
            <input type="file" id="document" name="document"
ng-model="ngo.document">
        <div class="form-actions">
            <input type="submit" name="save" value="Submit"
ng-disabled="registrationForm.$invalid">
        </div>
    </form>

<script>
    function on() {
        document.getElementById("overlay-content").style.display =
"block";
    }
    function off(){
        document.getElementById("overlay-content").style.display =
"none";
    }
    var app = angular.module("myApp", []);
    app.controller("formCtrl", function($scope, $window) {
        $scope.submitForm = function(isValid) {
            if (isValid) {
                $window.alert('Form submitted successfully!');
                off();
            } else {
                console.log('Form is invalid');
            }
        };
    });
</script>

```

```
</div>
```

Figure 3.1

Code snippet for NGO Registration form using HTML and Angular JS validation

```
<?php  
$server_name="localhost";  
$username="root";  
$password="";  
$database_name="helpify";  
  
$conn = mysqli_connect($server_name,$username,$password,$database_name);  
//check connection  
if(!$conn)  
{  
    die("Connection failed:" . mysqli_connect_error());  
}  
  
if(isset($_POST['save']))  
{  
    $organisation_name = $_POST['orgName'];  
    $ngo_type = $_POST['type'];  
    $ngo_address = $_POST['address'];  
    $contact_number = $_POST['contact'];  
    $mission = $_POST['mission'];  
    $lack_of_resources = $_POST['resources'];  
    $payment_link = $_POST['payment_link'];  
    $governing_doc = $_POST['document'];  
    $website = $_POST['website'];  
  
    $sql_query = "INSERT INTO  
ngo_details(organisation_name,ngo_type,ngo_address,contact_number,mission,  
lack_of_resources,payment_link,governing_doc,website)  
VALUES('$organisation_name','$ngo_type','$ngo_address','$contact_number','  
$mission','$lack_of_resources','$payment_link','$governing_doc','$website'  
)";
```

```

if(mysqli_query($conn,$sql_query)) {
    echo "NGO registered successfully!";
    header("Location:ngo-ngo.html");
} else{
    echo "Error" . $sql . mysqli_error($conn);
}
mysqli_close($conn);
?

```

Figure 3.1

Code snippet for connecting NGO Registration form to backend using PHP

Search with keywords				
Organisation Name	Type of organisation	Address	Payment Link	Website
Swastikam	Charity	Navi Mumbai	https://www.swastikam.org/support	https://www.swastikam.org/
Goonj	Non-profit	J-93, Sarita Vihar, New Delhi, Delhi 110076	https://rzp.io/l/goonj	https://goonj.org/
Akshaya Patra	Non-profit	#72, 3rd Floor, 3rd Main Road, 1st & 2nd Stage, Ye	https://rzp.io/l/akshayapatra	https://www.akshayapatra.org/
Foster home	Non-profit	Navi Mumbai	http://localhost/Helpify/homepage.html	http://localhost/Helpify/homepage.html

Organisation Name	Type of organisation	Address	Payment Link	Website
Goonj	Non-profit	J-93, Sarita Vihar, New Delhi, Delhi 110076	https://rzp.io/l/goonj	https://goonj.org/

Figure 3.2

Screenshot of database-connected table with Angular JS search functionality

```

<div ng-app="myApp" ng-controller="myCtrl">
    <input type="text" class="search-box" placeholder="Search with
keywords" ng-model="lookup" />
    <div id="ngo-table" class="table-container">
        <table>
            <thead>

```

```

<tr>
    <th>Organisation Name</th>
    <th>Type of organisation</th>
    <th>Address</th>
    <th>Payment Link</th>
    <th>Website</th>
</tr>
</thead>
<tbody>
    <tr ng-repeat="ngo in ngos | filter: lookup">
        <td>{{ ngo.organisation_name }}</td>
        <td>{{ ngo.ngo_type }}</td>
        <td>{{ ngo.ngo_address }}</td>
        <td><a href="{{ ngo.payment_link }}">{{ ngo.payment_link }}</a></td>
        <td><a href="{{ ngo.website }}">{{ ngo.website }}</a></td>
    </tr>
</tbody>
</table>
</div>
</div>

<script>
// Define the Angular module
var app = angular.module("myApp", []);

// Define the Angular controller
app.controller("myCtrl", function($scope, $http) {
$http.get("displayfindngo.php")
.then(function(response) {
$scope.ngos = response.data;
});
});

</script>

```

*Figure 3.2
Code snippet of database-connected table with Angular JS search functionality*

```
<?php  
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "helpify";  
  
// Create connection  
$conn = new mysqli($servername, $username, $password, $dbname);  
  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
  
// Retrieve data from the ngo_details table  
$sql = "SELECT organisation_name, ngo_type, ngo_address, payment_link,  
website FROM ngo_details";  
$result = $conn->query($sql);  
  
// Generate JSON-formatted response from the retrieved data  
$data = array();  
if ($result->num_rows > 0) {  
    while($row = $result->fetch_assoc()) {  
        $data[] = $row;  
    }  
}  
  
// Output the JSON-formatted response  
header('Content-Type: application/json');  
echo json_encode($data);  
  
// Close the database connection  
$conn->close();  
?>
```

Figure 3.2

Code snippet of database connectivity to fetch data from backend

The screenshot displays a web application interface for managing NGOs. At the top, there is a search bar with the placeholder "Search with keywords". Below the search bar is a table with four columns: "Organisation Name", "Lack of Resources", "Contact Number", and "Website". The table contains five rows of data. A second search bar below the table has the word "hygiene" entered. A third search bar at the bottom of the page also has "hygiene" entered.

Organisation Name	Lack of Resources	Contact Number	Website
Swastikam	Hygiene products	9876543202	https://www.swastikam.org/
Goonj	Lack of funding, lack of awareness about their wor	+91 11 269	https://goonj.org/
Akshaya Patra	Lack of funding, limited access to resources and s	+91 80 301	https://www.akshayapatra.org/
Foster home	Medications	9876543201	http://localhost/Helpify/homepage.html

hygiene

hygiene

Figure 3.3
Screenshot of database-connected table with Angular JS search functionality

```

<div ng-app="myApp" ng-controller="myCtrl">
  <input type="text" class="search-box" placeholder="Search with
  keywords" ng-model="lookup" />
  <div id="ngo-table" class="table-container">
    <table>
      <thead>
        <tr>
          <th>Organisation Name</th>
          <th>Lack of Resources</th>
          <th>Contact Number</th>
          <th>Website</th>
        </tr>
      </thead>
      <tbody>
        <tr ng-repeat="ngo in ngos | filter: lookup">
          <td>{{ ngo.organisation_name }}</td>
          <td>{{ ngo.lack_of_resources }}</td>
          <td>{{ ngo.contact_number }}</td>
          <td><a href="{{ ngo.website }}">{{ ngo.website }}</a></td>
        </tr>
      </tbody>
    </table>
  </div>

```

```

</div>

<script>
// Define the Angular module
var app = angular.module("myApp", []);

// Define the Angular controller
app.controller("myCtrl", function($scope, $http) {
$http.get("displayngo-ngo.php")
.then(function(response) {
$scope.ngos = response.data;
}) ;
}) ;

</script>

```

*Figure 3.3
Code snippet of database-connected table with Angular JS search functionality*

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "helpify";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Retrieve data from the ngo_details table
$sql = "SELECT organisation_name, lack_of_resources, contact_number,
website FROM ngo_details";
$result = $conn->query($sql);

```

```
// Generate JSON-formatted response from the retrieved data
$data = array();
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $data[] = $row;
    }
}

// Output the JSON-formatted response
header('Content-Type: application/json');
echo json_encode($data);

// Close the database connection
$conn->close();
?>
```

Figure 3.3
Code snippet of database connectivity to fetch data from backend

Contact Us

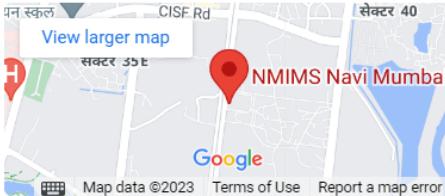
Get in Touch with Us

Hi there! Thanks for visiting our Contact Us page. We're here to lend a helping hand and answer any questions you may have. Don't hesitate to reach out to us, we're always happy to hear from you. Use any of the methods listed below to get in touch.

Phone Number
+91 9876543201

Email Address
text@gmail.com

Our location
NMIMS, near Pethpada Metro Station, Sector-33, Kharghar, Navi Mumbai, Maharashtra, 410210



Hittanshu

hittanshu

Invalid email address.

987

Invalid phone number.

Your Query

Submit your message

Figure 3.4
Screenshot of Contact form with ANGULAR JS validation

```
<form action="query.php" method="post" name="myForm"
ng-submit="submitForm(myForm.$valid)"><br><br><br><br><br>
    <input type="text" name="name" ng-model="name" placeholder="Your Name"
required pattern="[A-Za-z\s]+" /><br>
    <div ng-show="myForm.name.$dirty && myForm.name.$invalid">
        <span ng-show="myForm.name.$error.required" style="color: red;">Name
is required.</span>
        <span ng-show="myForm.name.$error.pattern" style="color:
red;">Invalid Name.</span>
    </div>
    <input type="email" name="email" ng-model="email" placeholder="Your
Email" required ng-pattern="/^[\s@]+@[^\s@]+\.[^\s@]{2,}$/ " /><br>
    <div ng-show="myForm.email.$dirty && myForm.email.$invalid">
```

```

        <span ng-show="myForm.email.$error.required" style="color:red;">>Email is required.</span>
        <span ng-show="myForm.email.$error.pattern" style="color:red;">>Invalid email address.</span>
    </div>
    <input type="tel" name="phone" ng-model="phone" placeholder="Your Phone" required pattern="^[0-9]{10}$" /><br>
    <div ng-show="myForm.phone.$dirty && myForm.phone.$invalid">
        <span ng-show="myForm.phone.$error.required" style="color:red;">>Phone number is required.</span>
        <span ng-show="myForm.phone.$error.pattern" style="color:red;">>Invalid phone number.</span>
    </div>
    <textarea name="message" ng-model="message" placeholder="Your Query" rows="5" cols="35" required></textarea><br><br>
    <input type="submit" name="save" value="Submit your message" ng-disabled="myForm.$invalid">
</form>

```

*Figure 3.4
Code Snippet of Contact form with ANGULAR JS validation*

```

<?php
$server_name="localhost";
$username="root";
$password="";
$database_name="helpify";

$conn = mysqli_connect($server_name,$username,$password,$database_name);
//check connection
if(!$conn)
{
    die("Connection failed:" . mysqli_connect_error());
}

if(isset($_POST['save']))
{

```

```
$name = $_POST['name'];
$email = $_POST['email'];
$phone = $_POST['phone'];
$message = $_POST['message'];

$sql_query = "INSERT INTO query_details(name,email,phone,message)
VALUES('$name','$email','$phone','$message')";

if(mysqli_query($conn,$sql_query)) {
    echo "Query submitted successfully!";
    header("Location:homepage.html");
} else{
    echo "Error" . $sql . mysqli_error($conn);
}

mysqli_close($conn);
}
?>
```

Figure 3.4
Code snippet of database connection for contact form

Welcome back!

Please sign in to continue your journey with us!

Username: hittanshu

Please enter a valid email or contact number.

Password: ••

Password should be at least 6 characters long.

Create Account

First name

Last name

Select User Type

Select Region

Contact

Email

Password

Confirm-Password

Sign In

Sign Up

Figure 3.5
Screenshot of User Registration form with ANGULAR JS validation

```
<div class="login">
    <h1>Welcome back!</h1><br><br>
    <p style="text-align: left; font-family: Karla; margin-left: 50px;">Please sign in to continue your<br> journey with us!</p><br><br>
    <form name="loginForm" ng-submit="submitLoginForm()" method="post">
        <label for="username">Username:</label>
        <input type="text" id="username" placeholder="Email or contact number" style="width: 180px" name="username" ng-model="username" ng-pattern="/^(?:[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}|\d{10,12})$/" required/>
        <div ng-show="loginForm.username.$error.required && loginForm.username.$dirty">
```

```

        <small class="error" style="color:red;">Username is
required.</small>
    </div>
    <div ng-show="loginForm.username.$error.pattern &&
loginForm.username.$dirty">
        <small class="error" style="color:red;">Please enter a
valid email or contact number.</small>
    </div><br>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password"
placeholder="Password" style="width: 180px" ng-model="password"
ng-minlength="6" required/>
        <div ng-show="loginForm.password.$error.required &&
loginForm.password.$dirty">
            <small class="error" style="color:red;">Password is
required.</small>
        </div>
        <div ng-show="loginForm.password.$error.minlength &&
loginForm.password.$dirty">
            <small class="error" style="color:red;">Password should be
at least 6 characters long.</small>
        </div>

        <input type="submit" name='save' value="Sign In">
    </form>
</div>
<div class="vr"></div>
<div class="reg">
    <h1 style="text-align: center;">Create Account</h1>
    <form action = 'userregistration.php' method='post'
name="SignForm" ng-submit="submitSignForm()" novalidate>

        <input type="text" id="first-name" name="first-name"
placeholder="First name" ng-pattern="/[a-zA-Z\s]+/" required>

        <input type="text" id="last-name" name="last-name"
placeholder="Last name" required><br><br>

```

```

<select id="user-type" name="user-type" required>
    <option value="" disabled selected>Select User
Type</option>
    <option value="donor">Donor</option>
    <option value="ngo">NGO</option>
    <option value="foster-home">Foster Home</option>
</select>&nbsp;&nbsp;&nbsp;

<select id="region" name="region"
onchange="showDropInfo()" required>
    <option value="" disabled selected>Select
Region</option>
    <option value="mumbai">Mumbai City</option>
    <option value="suburban">Mumbai Suburban</option>
    <option value="thane">Thane</option>
    <option value="palghar">Palghar</option>
    <option value="raigad">Raigad</option>
</select><p id="pT" style="margin-left:
10px;"></p><br><br>

<input type="text" name="number" placeholder="Contact">
<input type="text" name="email" placeholder="Email">
<input type="password" id="password" name="password"
placeholder="Password" ng-model="pass" ng-minlength="6" required>
    <input type="password" id="confirm-password"
name="confirm-password" placeholder="Confirm-Password" required><br>

    <input type="submit" value="Sign Up" name = 'save'
style="margin-left: 17%;">
</form>
</div>

```

*Figure 3.5
Code Snippet of User Registration form with ANGULAR JS validation*

```

<?php
$server_name="localhost";
$username="root";
$password="";
$database_name="helpify";

$conn = mysqli_connect($server_name,$username,$password,$database_name);
//check connection
if(!$conn)
{
    die("Connection failed:" . mysqli_connect_error());
}

if(isset($_POST['save']))
{
    $first_name = $_POST['first-name'];
    $last_name = $_POST['last-name'];
    $user_type = $_POST['user-type'];
    $region = $_POST['region'];
    $phone = $_POST['number'];
    $email = $_POST['email'];
    $pass = $_POST['password'];

    $sql_query = "INSERT INTO
user_details(first_name,last_name,type,region,phone,email,pass)
VALUES('$first_name','$last_name','$user_type','$region','$phone','$email',
'$pass')";

    if(mysqli_query($conn,$sql_query)){
        echo "User registered successfully!";
        header("Location:findngo.html");
    }else{
        echo "Error" . $sql . mysqli_error($conn);
    }
    mysqli_close($conn);
}
?>

```

*Figure 3.5
Code Snippet of database connection for User Registration form*

Chapter 4

Output Screenshots

The screenshot shows the homepage of Helpify. At the top, there is a navigation bar with links for "About Us", "Find your NGO!", "NGO-NGO", "Contact us", and "Login". The main header "Helpify" is on the left. Below the header is a large banner with a background image of several people's silhouettes flexing their biceps. Overlaid on the banner is the text "Partner with us in creating a positive change!". Below this, a paragraph of text reads: "Helpify aims at bringing underprivileged people voice out their needs on a bigger, better platform. There is a plethora of population residing in our country, deprived of fundamental amenities. We plan on moving toward helping people reach out to foster homes, donation drives. Connecting you to the needy. Don't let your opportunity go to waste. Start today." A "Register Now" button is located at the bottom of the banner. Below the banner, a quote by Steve Goodier is displayed: "Money is not the only commodity that is fun to give. We can give time, we can give our expertise, we can give our love, or simply give a smile. What does that cost? The point is, none of us can ever run out of something worthwhile to give." The quote is attributed to "Steve Goodier".

*Figure 4.1
Screenshot of Homepage*

The screenshot shows the homepage of Helpify. At the top, there is a navigation bar with links for "About Us", "Find your NGO!" (which is highlighted in blue), "NGO-NGO", "Contact us", and "Login". The main header "Helpify" is on the left. Below the header is a large banner with a background image showing a group of people, including children, in what appears to be a slum or a low-income residential area. Overlaid on the banner is the text: "We aim at Helping you find NGOs who need you. Or even near you for offline donations! Donate to make a difference!"

Figure 4.2
Screenshot of Find Your NGO page

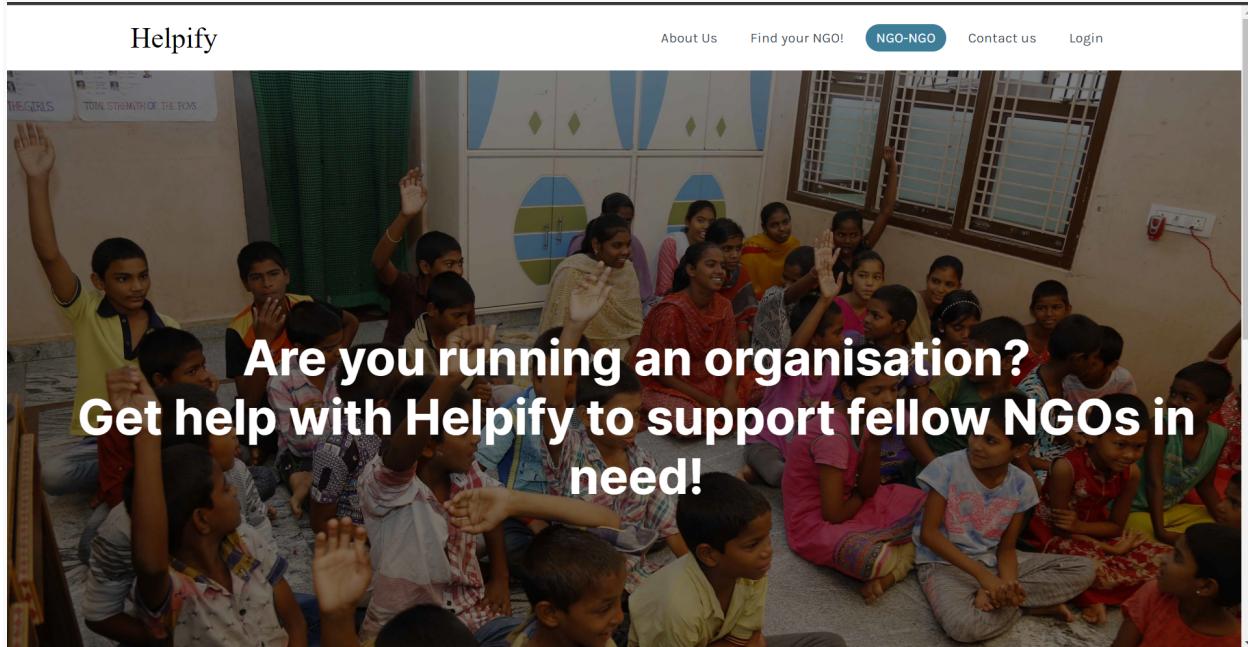


Figure 4.3
Screenshot of NGO-NGO page

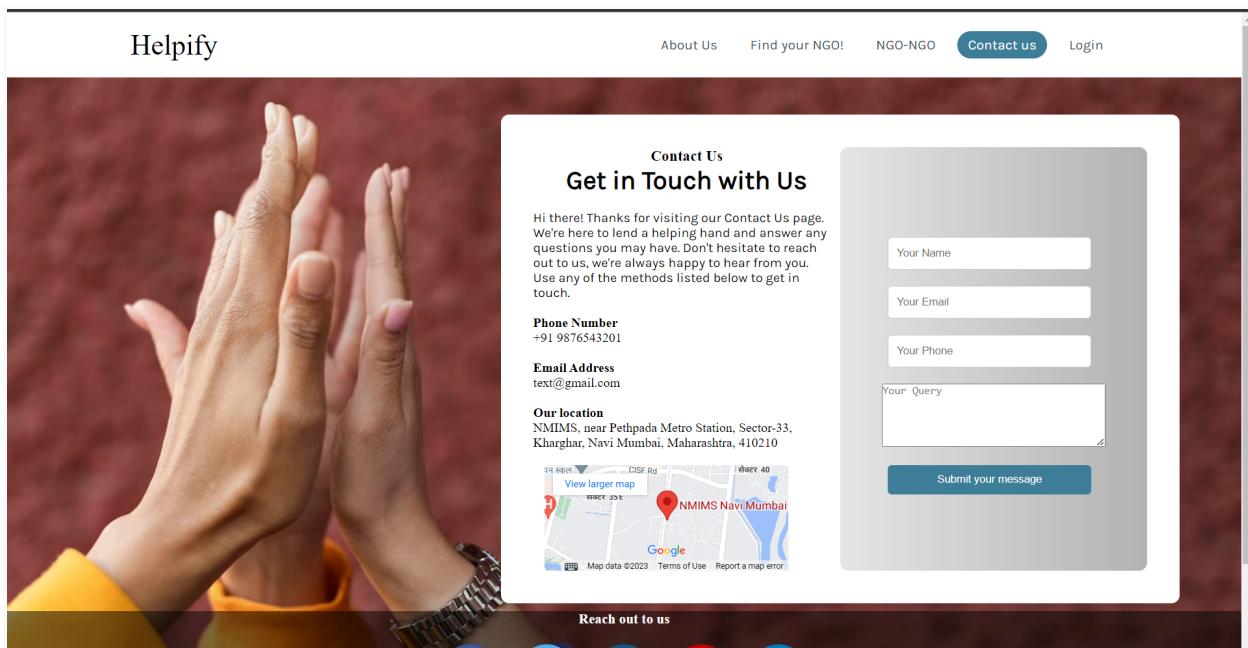


Figure 4.4
Screenshot of Contact Us page

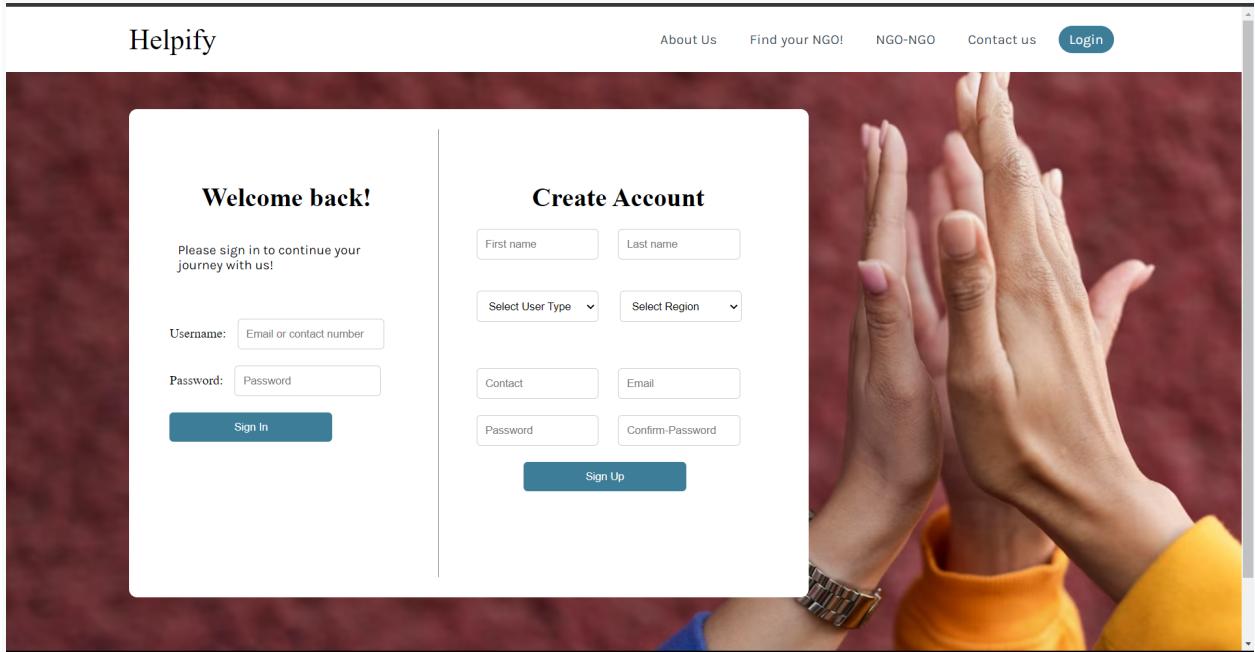


Figure 4.5
Screenshot of Login/Registration page

Chapter 5

Database Schema

Table	Action	Rows	Type	Collation	Size	Overhead
ngo_details		5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
query_details		4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
user_details		2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
3 table(s)	Sum					48 KiB

*Figure 5.1
Screenshot of Tables of Database*

organisation_name	ngo_type	ngo_address	contact_number	mission	lack_of_resources	payment_link	governing_doc	website
Swastikam	Charity	Navi Mumbai	9876543202	To make world a better place	Hygiene products	https://www.swastikam.org/support	[BLOB - 13 B]	https://www.swastikam.org/
Goonj	Non-profit	J-93, Sarita Vihar, New Delhi, Delhi 110076	+91 11 269	To work on various issues related to poverty, disa...	Lack of funding, lack of awareness about their wor	https://rzp.io/l/goonj	[BLOB - 13 B]	https://goonj.org/
Akshaya Patra	Non-profit	#72, 3rd Floor, 3rd Main Road, 1st & 2nd Stage, Ye	+91 80 301	To provide mid-day meals to school children across...	Lack of funding, limited access to resources and s	https://rzp.io/l/akshayapatra		https://www.akshayapatra.org/
Foster home	Non-profit	Navi Mumbai	9876543201	Lack of awareness	Medications	http://localhost/Helpify/homepage.html	[BLOB - 13 B]	http://localhost/Helpify/homepage.html

*Table 1.1
Screenshot of table NGO Details*

```
CREATE TABLE `ngo_details` (
  `organisation_name` varchar(30) NOT NULL,
  `ngo_type` varchar(30) NOT NULL,
  `ngo_address` varchar(255) NOT NULL,
  `contact_number` varchar(10) NOT NULL,
  `mission` varchar(255) NOT NULL,
  `lack_of_resources` varchar(255) NOT NULL,
  `payment_link` varchar(50) NOT NULL,
  `governing_doc` blob NOT NULL,
  `website` varchar(50) NOT NULL
)
```

*Figure
Code snippet for creating table NGO Details*

name	email	phone	message
Hittanshu Upadhyay	hittupadhyay@gmail.com	9372856186	Not able to find you.
Reyansh Gupta	reygupta@gmail.com	9322223603	Not able to find you.
Hittanshu	hittupadhyay@gmail.com	9874563201	yes
Yuvika	yuvikaslp@gmail.com	9874562310	Unable to find NGO

*Table 1.2
Screenshot of table Query Details*

```
CREATE TABLE `query_details` (
  `name` varchar(30) NOT NULL,
  `email` varchar(40) NOT NULL,
  `phone` varchar(10) NOT NULL,
  `message` varchar(255) NOT NULL
)
```

*Figure
Code snippet for creating table query Details*

first_name	last_name	type	region	phone	email	pass
Hittanshu	Upadhyay	donor	mumbai	9372856186	hittupadhyay@gmail.com	hhelpify
Yuvika	Singh	donor	mumbai	6897452130	hittupadhyay@gmail.com	helpify

*Table 1.3
Screenshot of table User Details*

```
CREATE TABLE `user_details` (
  `first_name` varchar(20) NOT NULL,
  `last_name` varchar(20) NOT NULL,
  `type` varchar(20) NOT NULL,
  `region` varchar(20) NOT NULL,
  `phone` varchar(10) NOT NULL,
  `email` varchar(30) NOT NULL,
  `pass` varchar(30) NOT NULL
)
```

*Figure
Code snippet for creating table User Details*

Chapter 6

Conclusion and Future scope

6.1 Conclusion -

In conclusion, Helpify was a successful journey to make a non-profit mediating website for helping donors and organizations. And also giving them a platform and opportunity to help each other. The project utilized technologies such as PHP, HTML, CSS with Bootstrap, and followed best practices in data modeling, system architecture, UI design, interaction design, security design, and test design. The analysis and design phase played a crucial role in setting the direction for the project, ensuring that the website met the requirements of today's audience. The completion of the project marked a significant achievement in creating a valuable tool and step together for a better India.

6.2 Future Scope -

1. **GPS:** Further GPS tracking can be incorporated in the search filter to list out all the organizations reaching out in need. Even without registering. Like a google search with keyword tracking.

2. **Statistics:** The page can also display a statistical graph of how many Organizations have been helped through live data.

Chapter 7

Links

Github Link -

<https://github.com/hittanshu/Helpify>