

**RESLOCATE**

**A Group Project Submitted for Undergraduate DBMS Lab (BCA481)**

**2023-2024**

By

**HARDIK ABHINEET SHAH(2241131)**

**NAYANA K BENNY (2241145)**

**SANDEEP MATHEW (2241155)**

**Under the supervision of**:

**Dr Smitha Vinod**

**Project report submitted in partial fulfilment**

**of the requirements of IV semester BCA, CHRIST (Deemed To Be University)**

**April- 2024**



CERTIFICATE

This is to certify that the report titled **ResLocate** is a bona fide record of work done by **Hardik Abhineet Shah(2241131),Nayana K Benny(2241145), Sandeep Mathew(2241155)** of CHRIST (Deemed to be University), Bangalore, in partial fulfilment of the requirements of

IV Semester **BCA** during the year 2024.

**Head of the Department Project Guide**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Valued-by:**

1 Name: Sandeep Mathew

Register Number: 2241155

2 Examination Centre: CHRIST (Deemed to be University)

Date of Exam:

**ACKNOWLEDGEMENTS**

First of all, we thank God almighty for his immense grace and blessings showered on us at every stage of this work. We are grateful to our respectable Head, Department of Computer Science, CHRIST (Deemed to be University), **Dr Ashok Immanuel V**, for providing the opportunity to take up this project as part of my curriculum.

We also pay our gratitude to the Coordinator, Department of Computer Science, CHRIST (Deemed to be University) **Dr Beaulah Soundarabai P** for their support throughout.

We are grateful to our guide, Associate Professor, Department of Computer Science, CHRIST (Deemed to be University), **Dr Smitha Vinod**, whose insightful leadership and knowledge benefited us to complete this project successfully. Thank you so much for your continuous support and presence whenever needed.

We express our sincere thanks to all faculty members and staff of the Department of Computer Science, CHRIST (Deemed to be University), for their valuable suggestions during the course of this project. Their critical suggestions helped us to improve the project work.

Last but not the least, we would like to thank everyone who is involved in the project directly or indirectly.

**ABSTRACT**

A website that assists users to locate safe and clean sources of water and adequate sanitization .Users will be able to find the nearest source of clean water, sanitization, free food (food banks).The software gathers information through geolocation and crowdsourcing. The software additionally educates users about the need for water conservation and good hygiene habits. It also intends to solve water scarcity and health issues in developing rural communities, where access to safe drinking water and basic health services is frequently limited. The app is designed to be user friendly, affordable and inclusive, and to promote behaviour change and social impact.

**TABLE OF CONTENTS**

**Title Page**

**Certificate Page**

**Acknowledgements iii**

**Abstract iv**

**Table of Contents v**

**List of Figures viii**

|  |  |
| --- | --- |
| **1 Introduction** | **1** |
| **1.1. The Project Description** | **1** |
| **1.2. Existing System** | **2** |
| **1.3. Objectives** | **2** |
| **1.4. Purpose, Scope and Applicability** | **3** |
| **1.4.1 Purpose** | **3** |
| **1.4.2 Scope** | **3** |
| **1.4.3 Applicability** | **3** |
| **1.5. Overview of the Project** | **3** |
| **2. System Analysis and Requirements** | **4** |
| **2.1. Problem Definition** | **3** |
| **2.2. Requirement Specification** | **5** |
| **2.2.1. Functional Requirements** | **5** |
| **2.2.2. Technical Requirements** | **6** |
| **2.3. System Requirements** | **7** |
| **2.3.1. Software Requirements** | **7** |
| **2.3.2 Hardware Requirements** | **7** |
| **2.3.3. Network Requirements** | **7** |
| **2.4. Conceptual Models** | **8** |
| **2.4.1. Data Flow Diagram** | **8** |
| **2.4.2. ER Diagram**  **2.4.3. Use Case Diagram** | **9**  **10** |
| **2.5. Planning and Scheduling** | **11** |
| **2.5.1. Gantt Chart** | **11** |
| **2.6. Proposed Tools** | **11** |

1. **Module Description 14**

**3.1 Find Resources 14**

* 1. **Give Back To Community 14**

**3.3 Post Volunteering Ads 14**

**3.4 Add resource Location 14**

1. **Proposed Logic of Modules 16**
2. **Testing 17**
   1. **Test Cases and the Reports 18**
3. **Tables 19**
4. **UI and Implementation 21**
5. **Future Enhancements 31**
6. **Conclusion 32**
7. **References 33**
8. **Appendix 34**

## List of Figures

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
| 1 | DFD Level 2 | 8 |
| 2 | ER Diagram | 9 |
| 3 | Use Case Diagram | 10 |
| 4 | Home Page | 21 |
| 5 | Learn page | 22 |
| 6 | About Us Page | 23 |
| 7 | Find Resources Page | 21 |
| 8 | Give Back to Community Page | 22 |
| 9 | Login Page | 24 |
| 10 | Contributors Page | 24 |
| 11 | Post Volunteering Ads Page | 25 |
| 12 | Add Resource Location | 25 |
| 13 | Resource Data | 19 |
| 14 | Users Table | 20 |
| 15 | Work Data Table | 20 |

**1. INTRODUCTION**

**1.1PROJECT DESCRIPTION**

Access to clean water, sanitation, and other resources in rural areas can be a significant

challenge due to various factors. Geographical isolation often leaves rural

regions distant from infrastructure development, making it difficult and expensive to

establish water and sanitation systems. Economic constraints in rural areas, where

incomes are typically lower than in urban areas, limit the ability to invest in such

infrastructure. According to the World Bank, 80% of those lacking access to safe

drinking water and improved sanitation live in rural areas.

1. The World Health Organization (WHO) reports that over 844 million people in

rural areas lack access to even a basic drinking water service.

2. The United Nations reported in 2020 that around 2.2 billion people worldwide

lacked access to safe drinking water services, and about 4.2 billion people lacked

access to safe sanitation services.

3. Inadequate water, sanitation, and hygiene, according to the WHO, contribute to

842,000 deaths annually.

4. In 2022, 35-40% of the 122 million smartphones sold in the country were

bought in rural India, according to research firm IDC India. There is at least

one smartphone per family in rural areas in India.

**1.3 OBJECTIVES**

A web/mobile application that assists users to locate safe and clean sources of water and adequate sanitization. Users will be able to find the nearest source of clean water, sanitization, free food. The software gathers information through geolocation and crowdsourcing. The software additionally educates users about the need for water conservation and good hygiene habits. It also intends to solve water scarcity and health issues in developing rural communities, where access to safe drinking water and basic health services is frequently limited. The app is designed to be user-friendly, affordable and inclusive, and to promote behaviour change and social impact.

**1.4 PURPOSE, SCOPE AND APPLICABILITY**

**1.4.1. Purpose**

The purpose of this project is to address critical needs surrounding access to clean water, sanitation, and basic health services, particularly in developing rural communities where such resources are often scarce.

**1.4.2. Scope**

We live in a fast era of technology where no one has spare time to find resources and search for them hence the website helps to identify the resources.

**1.4.3. Applicability**

* **Rural Communities**: The project is particularly applicable in rural areas of developing countries where access to clean water and basic health services is limited. It addresses the immediate needs of these communities by providing information on nearby sources of safe water, sanitation facilities, and healthcare resources.
* **Emergency Situations:** During natural disasters, conflicts, or other emergencies, access to clean water and sanitation becomes even more critical. This project's ability to quickly identify and disseminate information about available resources can be invaluable in such situations, aiding relief efforts and saving lives.

**1.5. OVERVIEW OF THE PROJECT**

ResLocate is a comprehensive web platform designed to address critical challenges related to access to clean water, sanitation facilities, and basic health services, particularly in underserved communities. This project aims to provide users with easy access to vital information, promote behaviour change, and foster long-term social impact through a user-friendly and inclusive digital interface

**2.SYSTEM ANALYSIS AND REQUIREMENTS**

**2.1 PROBLEM DEFINITION**

Access to clean water, sanitation, and other resources in rural areas can be a significant challenge due to various factors. Geographical isolation often leaves rural regions distant from infrastructure development, making it difficult and expensive to establish water and sanitation systems. Economic constraints in rural areas, where incomes are typically lower than in urban areas, limit the ability to invest in such infrastructure. According to the World Bank, 80% of those lacking access to safe drinking water and improved sanitation live in rural areas.

**2.2 EXISTING SYSTEMS**

* <https://foodforfree.org/>
* <https://cleanwater.org/>

Our initiative offers a unique approach compared to platforms like Food for Free and Clean Water. While those websites collect cash donations from users and directly provide resources, our platform serves as a comprehensive database, offering users access to information about the locations where they can procure these resources themselves.

Additionally, we aim to streamline accessibility by consolidating both free food and clean water services on a single website. This design simplifies navigation and comprehension, particularly for residents in rural areas.

**2.3. REQUIREMENT SPECIFICATION**



**2.2.1 Functional Requirement**

* + - This Software facilitates reducing the risk of waterborne diseases, such as diarrhea, cholera, typhoid, and dysentery, by helping users find safe and clean sources of drinking water and sanitation facilities.
    - This Software also facilitates reducing hunger and malnutrition by helping users find food banks that offer free or low-cost food to those who need it.
    - There is an option which provides for new user account creation.
    - Users can further view details and update restricted information.
    - There is a volunteer section for the users who are willing to contribute to the website.

For example, users can post free food available location’s, volunteer ads, post clean water locations.

* + - Users can also request to check the water purity.

**2.2.2 Technical Requirement**

* + - **Accessibility and Usability**: Ensure the website is accessible across different devices and internet speeds prevalent in rural areas. Optimize for mobile use, as it might be the primary mode of internet access.
    - **Interactive Maps:** Incorporate maps showcasing areas lacking water and sanitation facilities. This can help visualize the scope of the issue and target interventions effectively.
    - **Donation/Support Mechanisms:** Include options for online donations or ways for visitors to support ongoing projects financially or through volunteering.
    - **Data Storage:** The user information is stored in a secure manner such as account passwords. The data is securely stored in a database which can only be accessed by authenticated admins of the software.
    - **Reliability:** The website is well designed with attractive color palettes for good user interface, it is optimized for easy access and working.

**2.3 SYSTEM REQUIREMENT**

**2.3.1 Hardware Requirement**

**Processor:** Minimum Intel i3 or equivalent

**Memory:** 4 GB DDR3 RAM

**Storage:** 1 GB available space

* + 1. **Software Requirements**

**Operating System:** Windows 7 or later

**DBMS:** MongoDB

**Backend:** NodeJS.

**Frontend:** HTML, CSS, JavaScript.

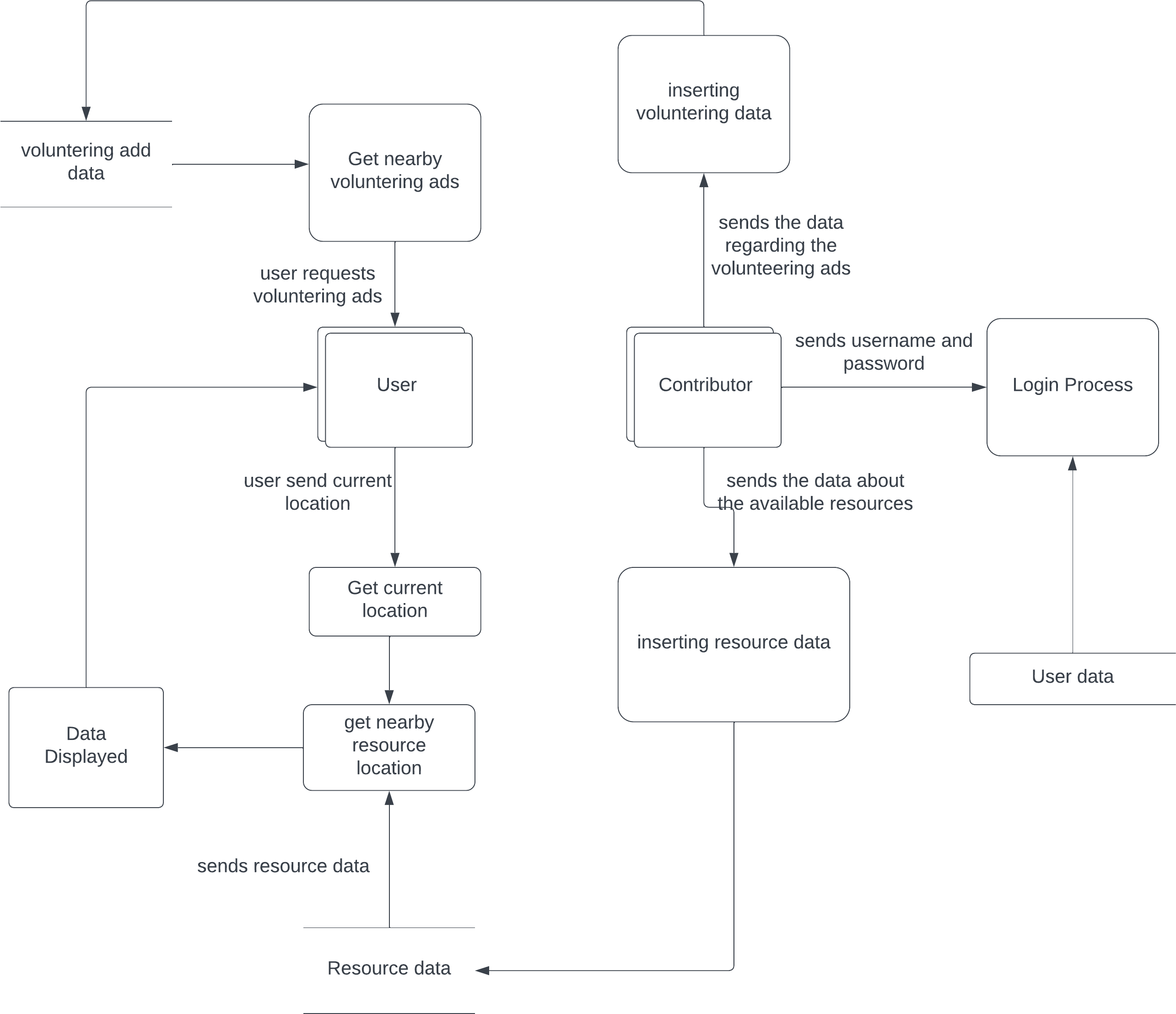
**API:** Google Maps API, W3C Geolocation API.

**2.3.3 Network Requirements**

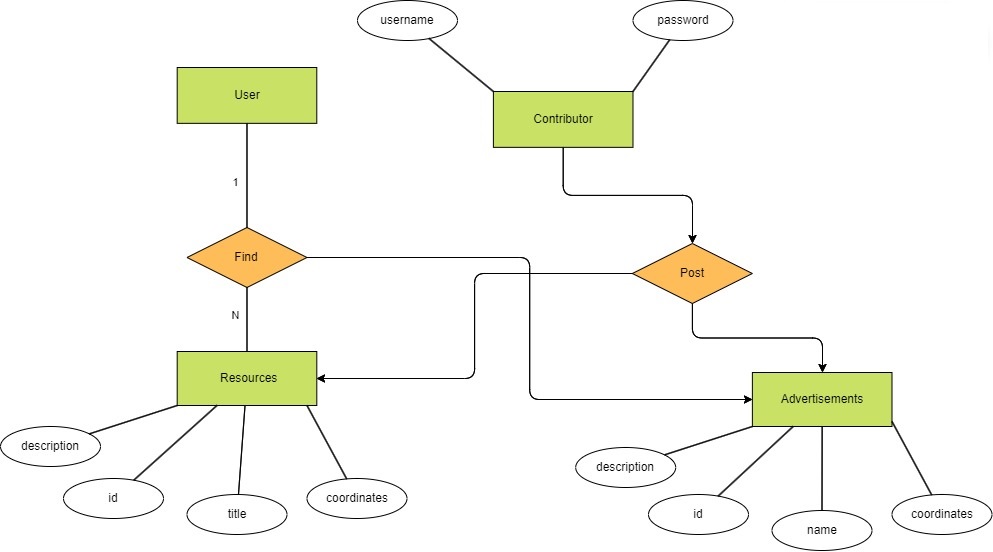
**Network:** Google Chrome / Mozilla or equivalent

**2.4 CONCEPTUAL MODEL**

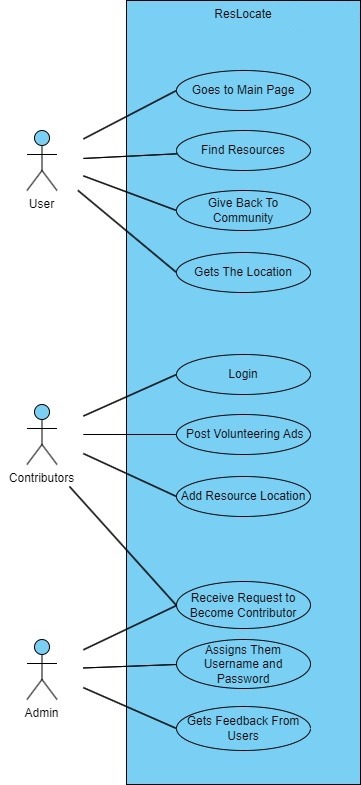
**2.4.1 DATA FLOW DIAGRAM (DFD)**



**2.4.2 ER DIAGRAM**

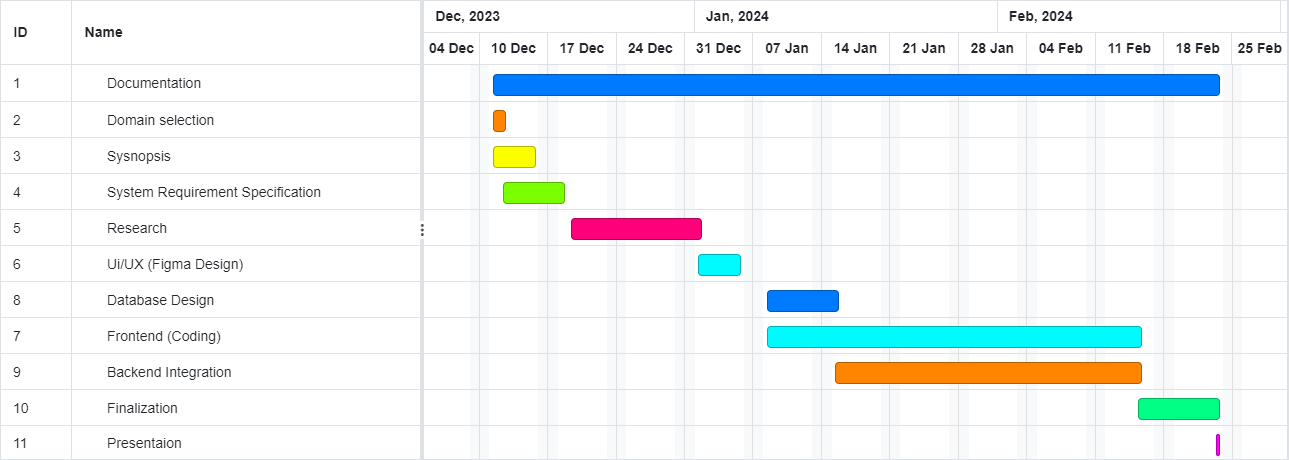


**2.4.3 USE CASE DIAGRAM**



**2.5 PROJECT PLANNING**

**2.5.1 GANTT CHART**



**2.6. PROPOSED TOOLS AND PLATFORMS**

**2.6.1Windows 10**

Microsoft Windows 10 is a personal computer operating system developed and released by Microsoft. All the files containing code and database will be created in the Windows 10 operating system.

**2.6.2 Google Chrome**

Google Chrome is a cross platform web browser developed and maintained by Google. All the html and php files will be opening using Google Chrome for viewing their functionality and working.

**2.6.3 HTML**

HTML stands for Hypertext Mark-up Language which is the standard mark-up language for documents designed to be displayed in a web browser. The basic web pages design will be implemented using HTML.

* + 1. **CSS**

CSS or Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a mark-up language such as HTML. It will be used for adding more design and improving the

look and feel of the website.

* + 1. **JavaScript**

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. It is high-level, often just-in-time compiled, and multi-paradigm. It will be used to make the website responsive and dynamic.

* + 1. **MongoDB**

MongoDB is an open-source platform that provides a set of tools for building and deploying applications with a focus on backend infrastructure. It aims to simplify the process of building scalable and secure applications by offering a variety of features, including a real-time database, authentication, and storage.

**2.6.7 VISUAL STUDIO CODE**

Visual Studio Code is a source-code editor that can be used with a variety of

programming languages, including C#, Java, JavaScript, Go, Node.js, Python, C++, C, Rust and Fortran. It is based on the Electron framework, which is used

to develop Node.js web applications that run on the Blink layout engine. Visual

Studio Code employs the same editor component (codenamed "Monaco") used in

Azure DevOps (formerly called Visual Studio Online and Visual Studio Team

Services).

**3.MODULE DESCRIPTION**

**3.1 FIND RESOURCES**

Users can search for specific resources based on their location or desired service. The search function allows for filtering by resource type, distance, and availability, enabling users to quickly find relevant information tailored to their needs.

**3.2 GIVE BACK TO COMMUNITY**

The module provides a centralized platform for users to explore volunteer opportunities within their community. Users can browse through listings of local organizations, nonprofits, and community initiatives seeking volunteers to support various causes, including water sanitation projects, health outreach programs, and humanitarian efforts.

**3.3 POST VOLUNTERING ADS**

Users have access to a dedicated forum where they can create and post advertisements for volunteer opportunities related to cleaning initiatives. The forum provides a platform for users to communicate details about upcoming clean-up events, including the date, time, location, and objectives of each initiative.

**3.4 ADD RESOURCE LOCATION**

Users can categorize the resource location based on its type, such as water source, sanitation facility, food bank Categorization ensures that users can easily filter and search for specific types of resources based on their needs and preferences.

**3.5 LOGIN**

The module ensures secure authentication for contributors, requiring them to enter their username and password to access their accounts. User authentication protocols protect sensitive information and prevent unauthorized access to contributor profiles and data.

**4. PROPOSED PROCESS LOGIC OF MODULES**

The clean water and free food resource locator website, "Resource Relief," serves as a vital platform for individuals to access essential resources while also promoting awareness and connectivity between communities and available aid providers. Here's how the website facilitates access to clean water and free food resources

First customer opens the website:

* + - 1. Clicks on desired options and Find resources or Give Back to Community
      2. It goes to a page where they select from which category they are and process
      3. Contributors will go to the login page and enter via user id & password
      4. Then move on to their portal
      5. Login page where they can overview the work going on with customer accounts
      6. New Contributors can Register via filling in their details and the admin will give their user id and password log in and they can create the account
      7. The Contributors can then post ads, or give resource location
      8. These points make up the process logic of the project

**5. TESTING**

A strategy for software testing integrates software test cases into a series of  well-planned steps that result in a successful construction of software.  Software testing is a broader topic for what is referred to as verification and  validation. Verification refers to the set of activities that ensure that the  software that has been built is traceable to customer’s requirements.

The steps involved in testing are

**5.1.1 Unit testing**

Unit testing means testing each unit of design separately. Here in this  project, we tested each unit of design separately and verify that there  were no errors. For this testing each design is run individually After  executing each page if there any error occurs correction mechanism is  done instantly.

**5.1.2 Integration testing**

In our project we combine many units of module to form a sub system.  These sub systems are then tested. This is done to see whether the  modules can be integrated properly. Based on integration testing some  necessary changes were made to the design.

**5.1.3 System testing**

System testing is done to ensure the entire software performs  its function as intended. In our project all the tested subsystems  were integrated and tested for all the possible ranges of  coupling variables, based on the testing errors were rectified  for pleasant working experience.

**5.1.4** **Acceptance testing**

The goal of acceptance testing is to see if the software meets all the  requirements as needed. The testing was performed by data of all the  users of the system.

**5.2 TEST CASES AND REPORTS**

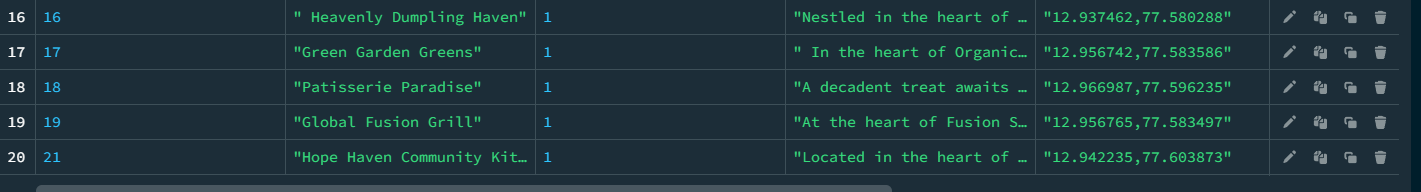
**Login Module:**

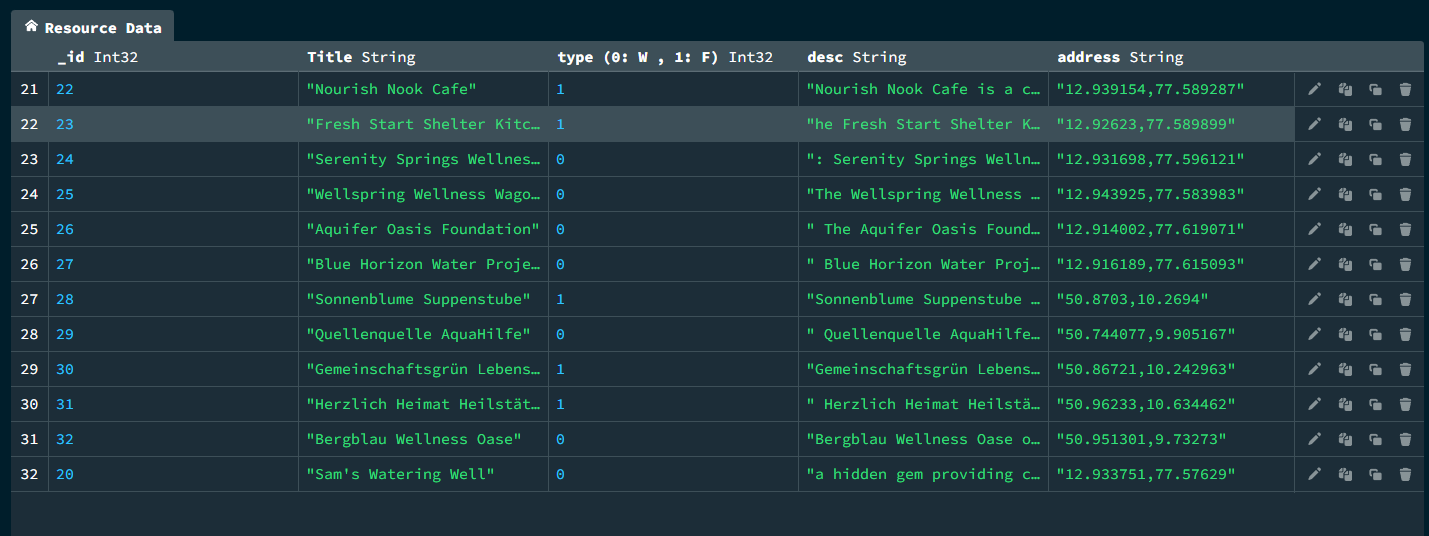
|  |  |  |  |
| --- | --- | --- | --- |
| **1.1** | **Input:** Valid  credentials and the right username, password | **Output:** Successfully  logged in | The user is  successfully logged  into his account and  redirected to their page |
| **1.2** | **Input:** Invalid  credentials | **Output:** Enter the right  credentials, Incorrect  Username and password | The user is sent a  prompt to recheck  their details and  enter the right credentials. |

**6. TABLES**

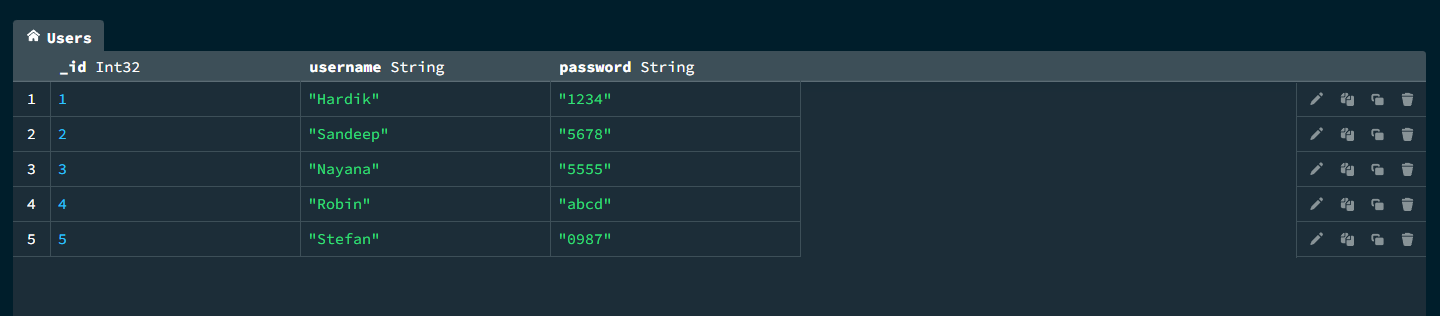
**Find Resources**

****

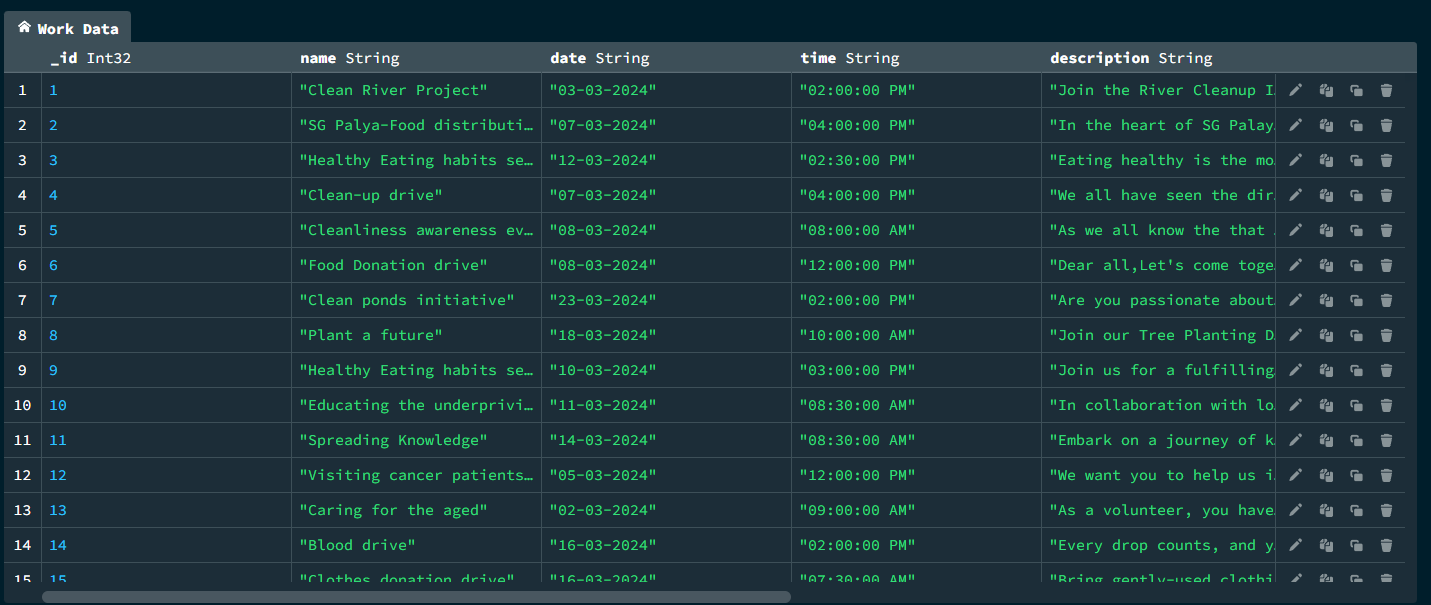
****

****

**Users**

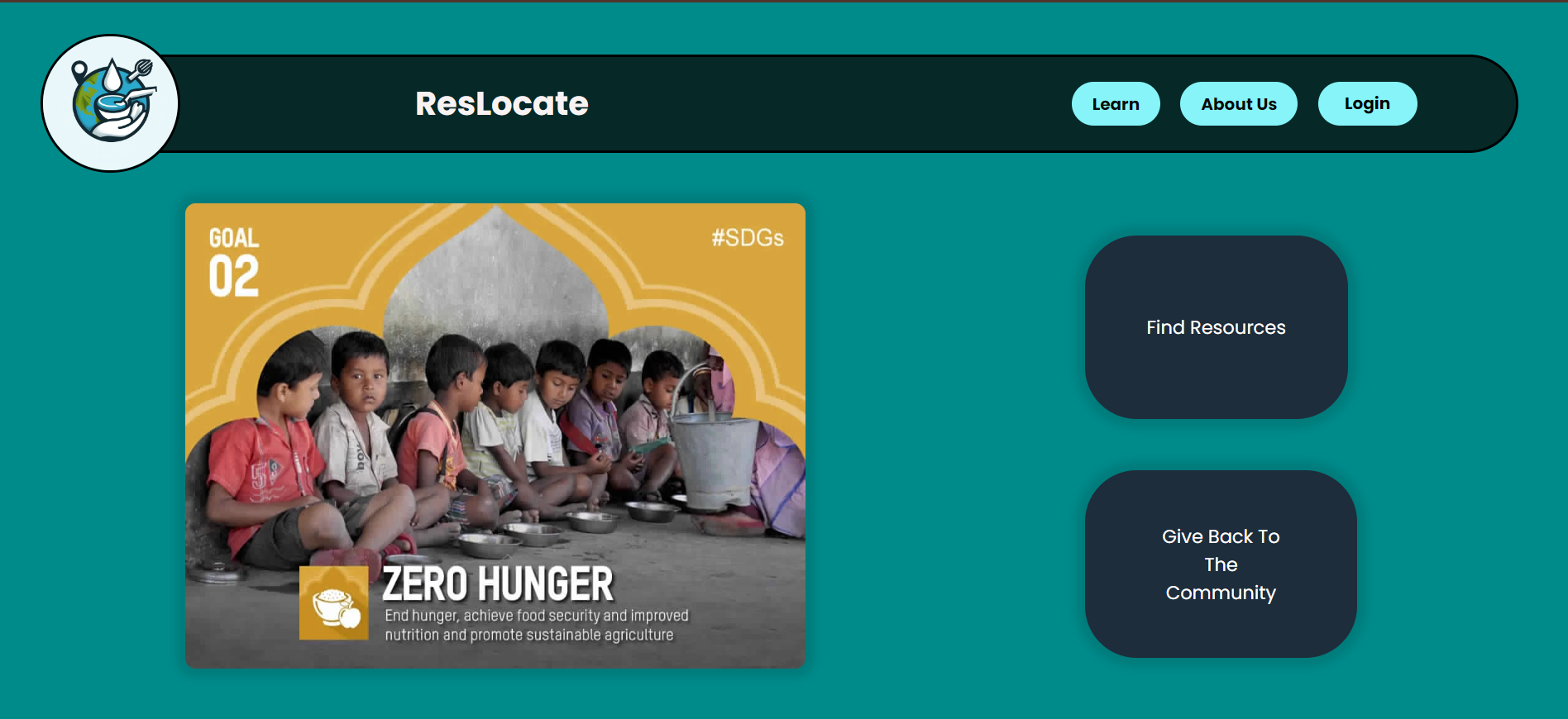
****

**Work Data**

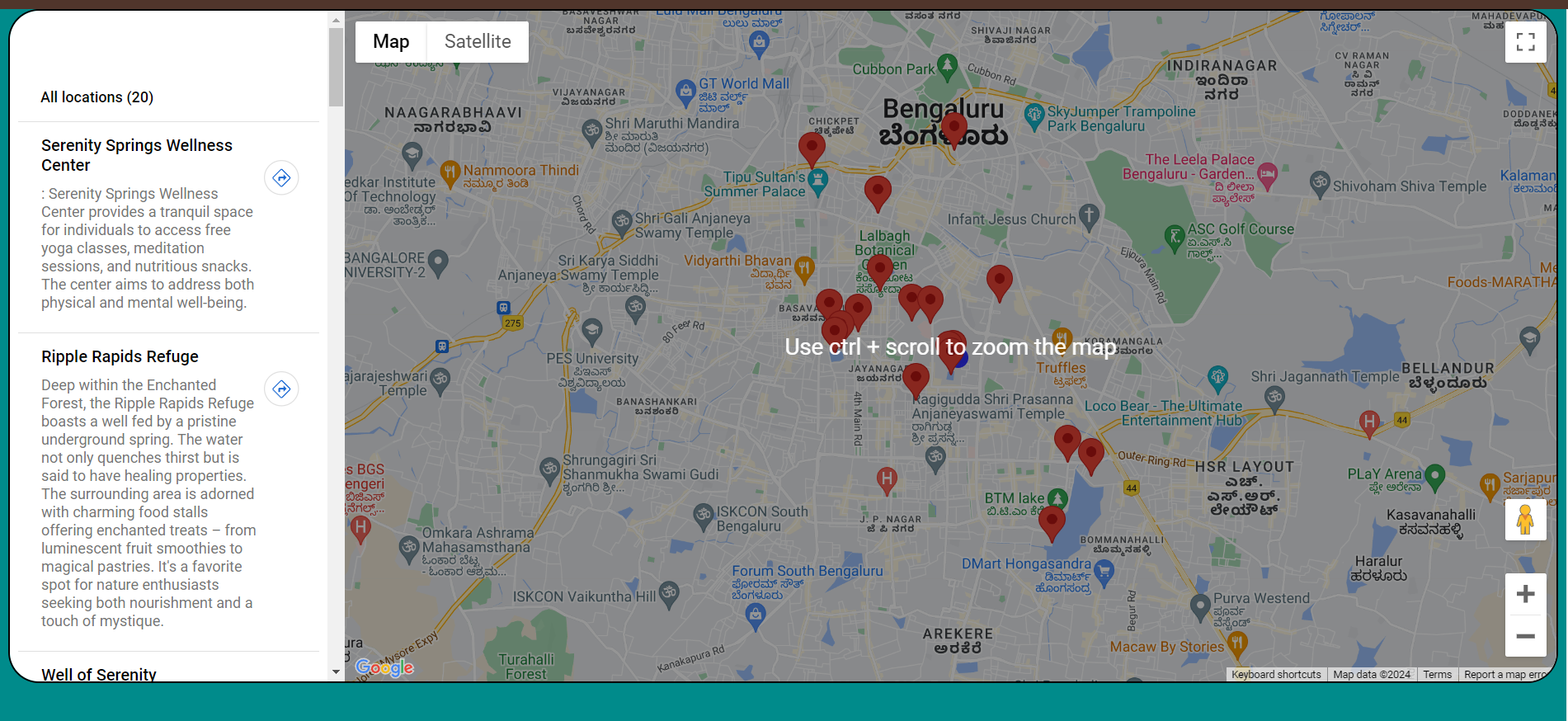
****

**7.IMPLEMENTATION &UI**

**MAIN PAGE**

****

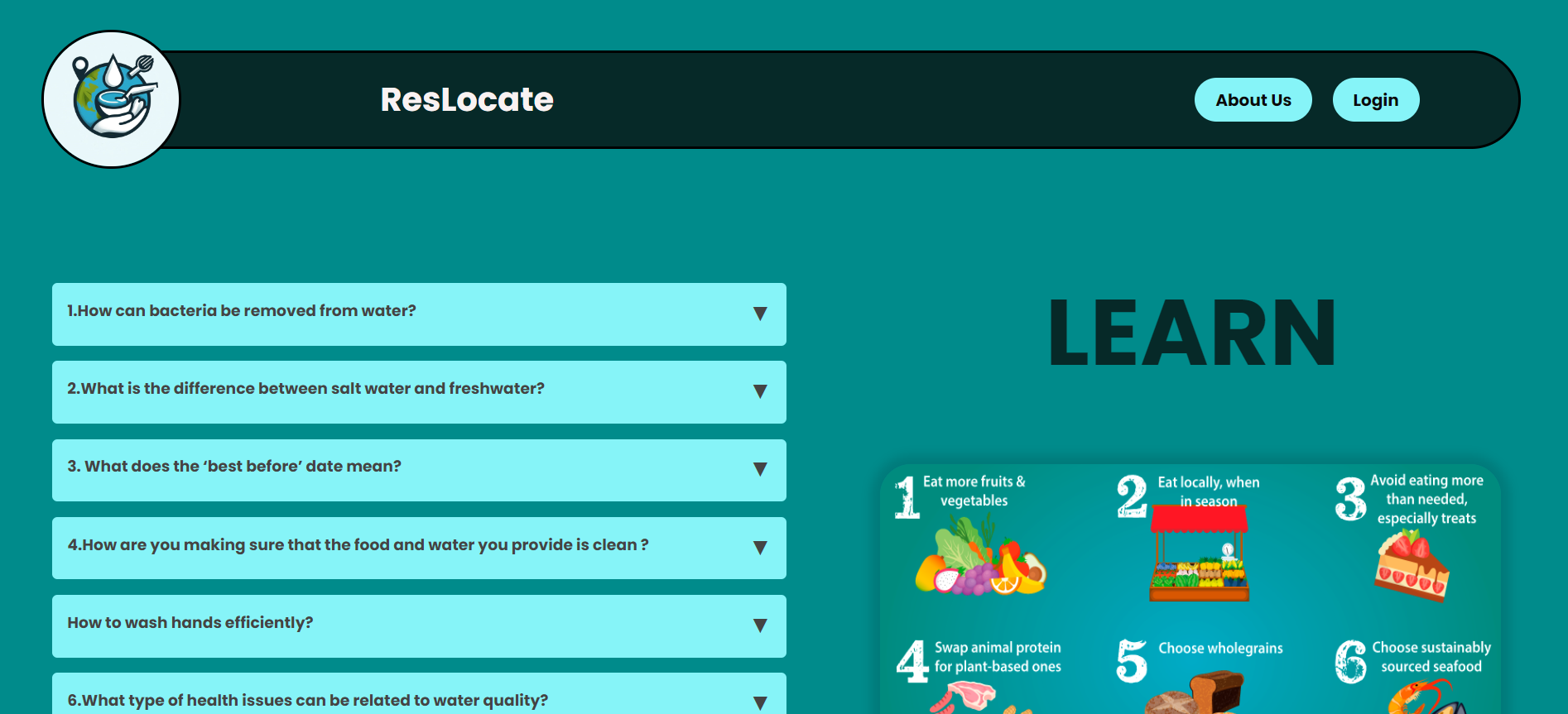
**FIND RESOURCES**

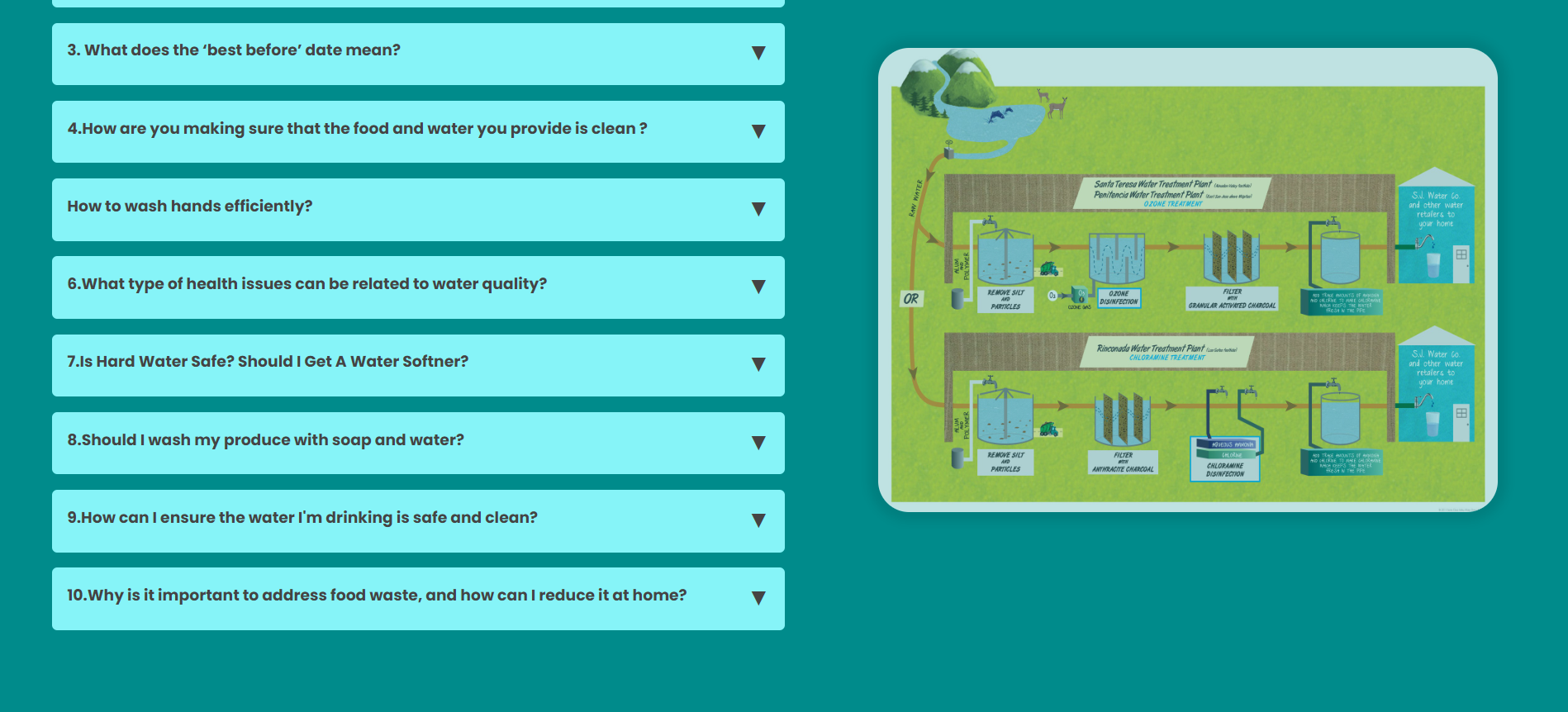
****

**GIVE BACK TO COMMUNITY**

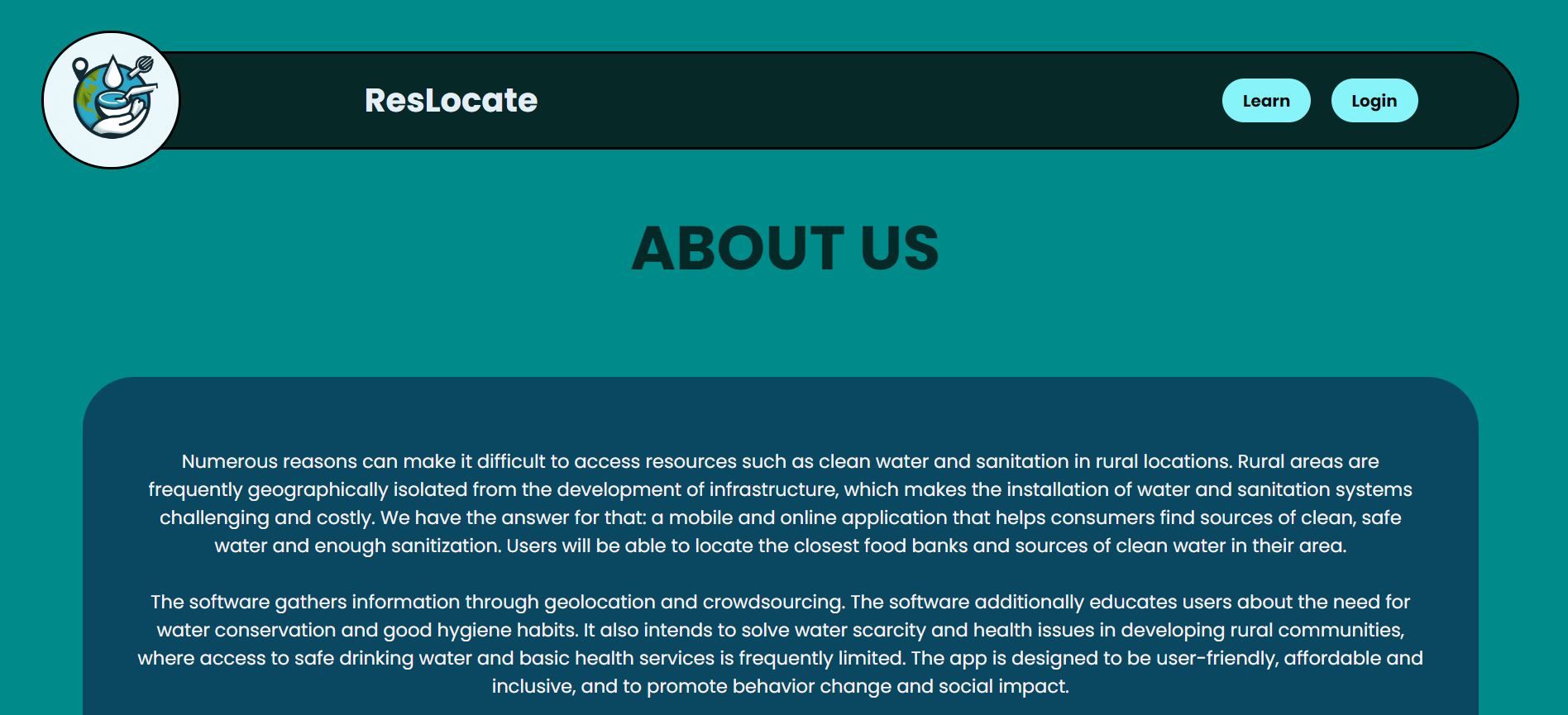
****

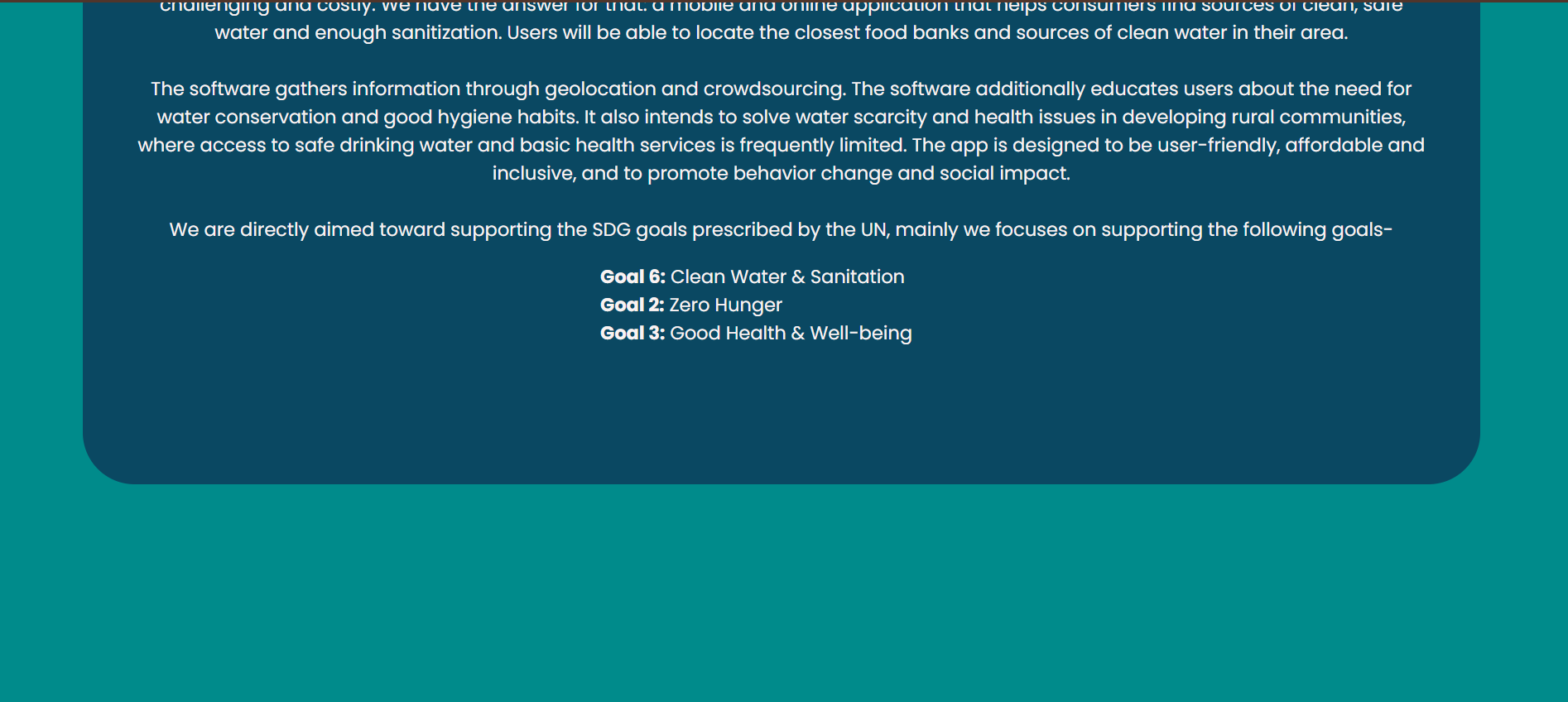
**LEARN PAGE**

****

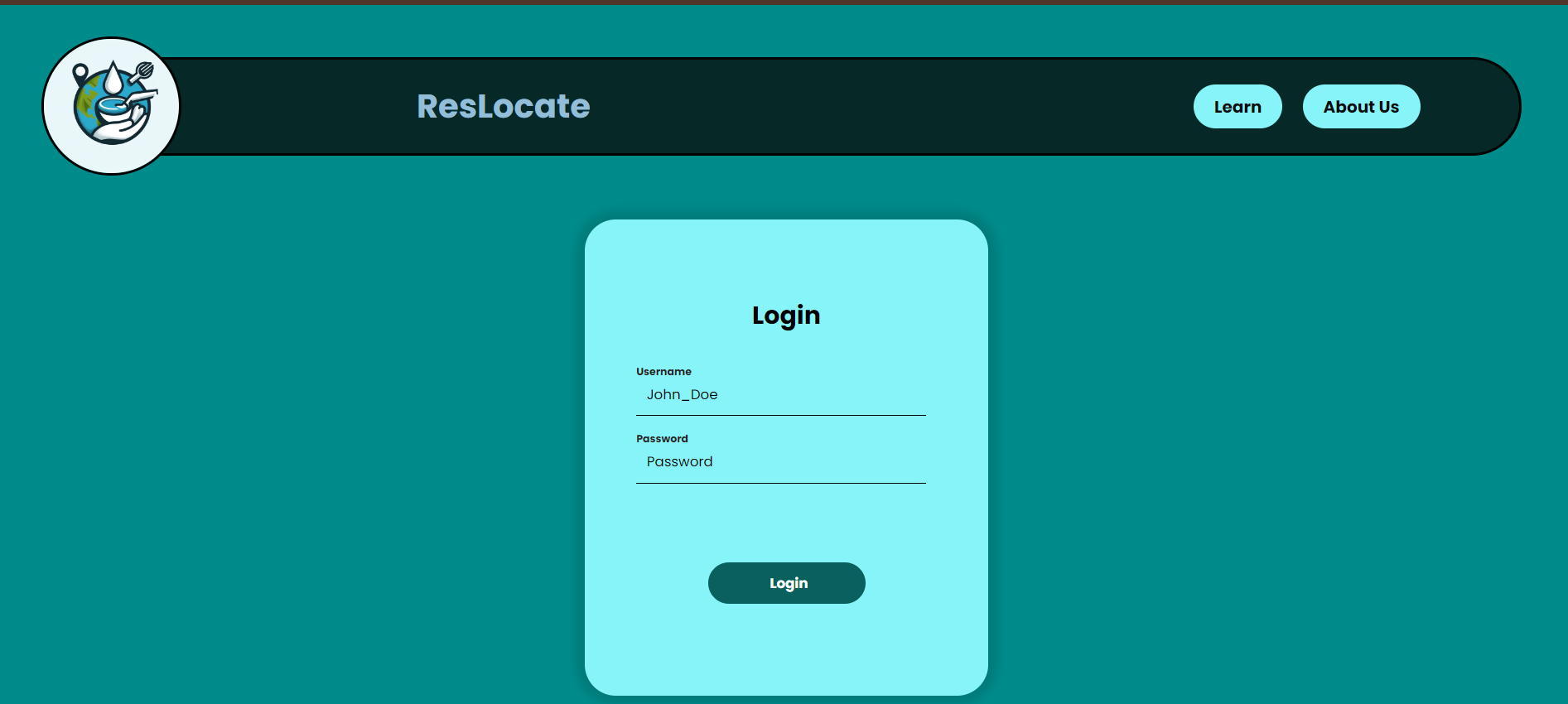
****

**ABOUT US**

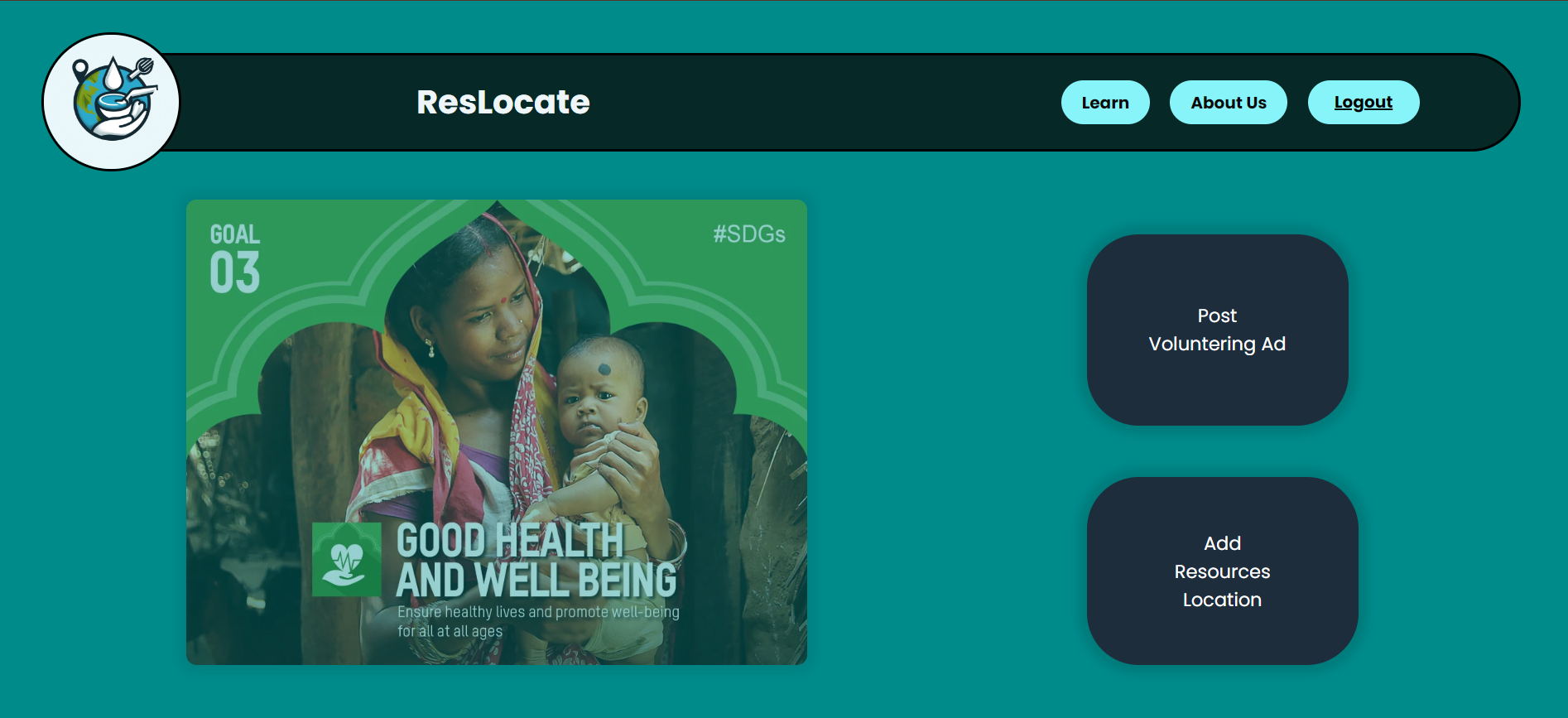
****

****

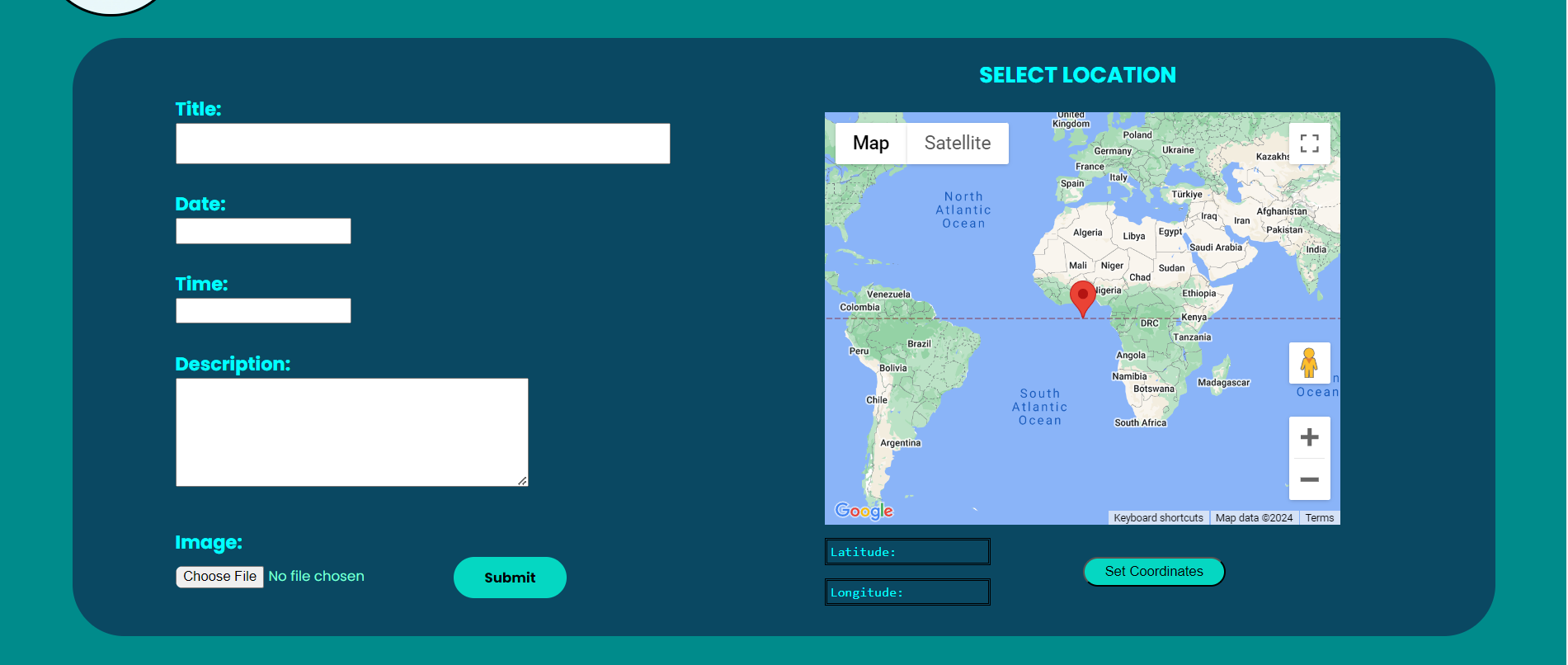
**LOGIN**

****

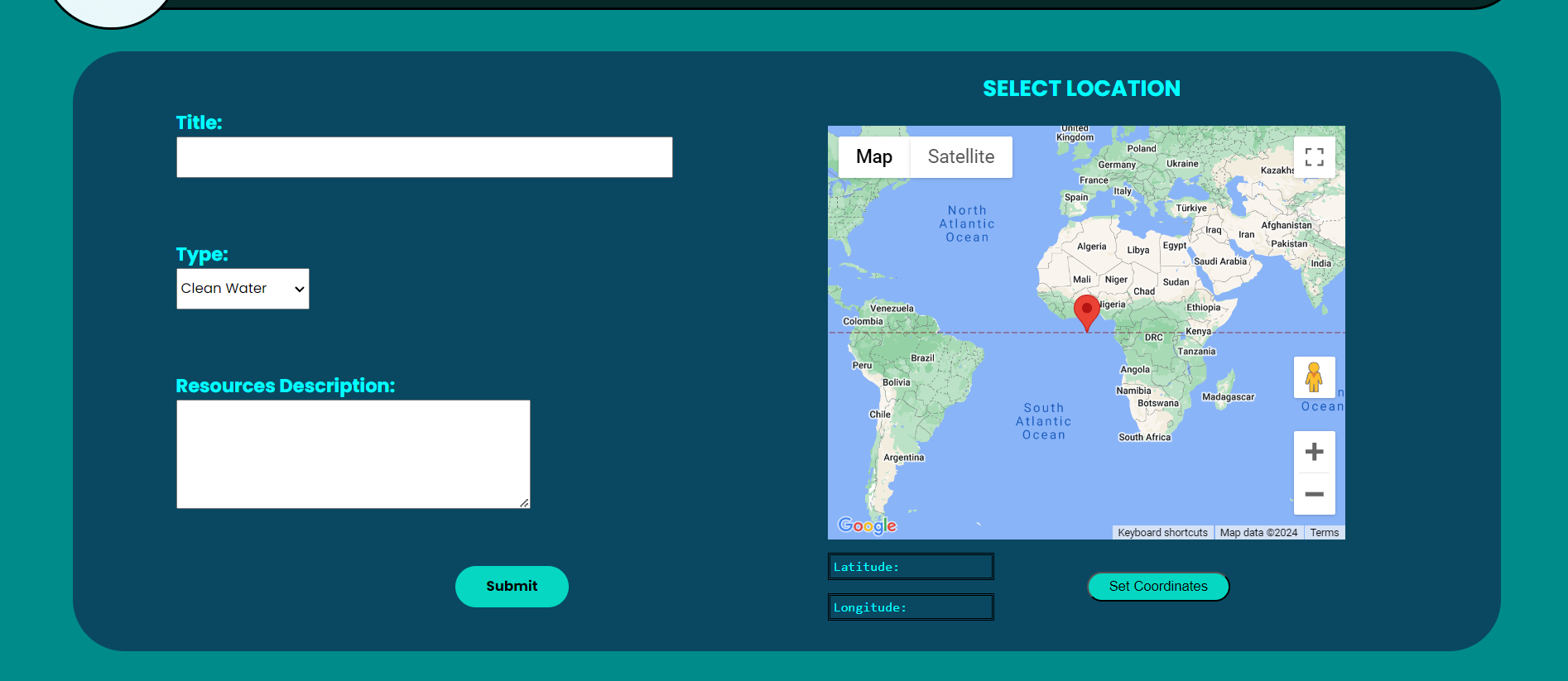
**CONTRIBUTORS PAGE**

****

**POST VOLUNTERING ADS**

****

**ADD RESOURCE LOCATION**

****

**IMPLEMENTATION**

**Main Page**

<html>

<head>

    <link rel="stylesheet" href="Main\_style.css">

    <title>ResLocate</title>

    <style>

    </style>

</head>

<body style="background-color:darkcyan;">

    <header>

        <div class="container">

            <h1 class="name">ResLocate</h1>

            <nav class="list">

                <ul >

                    <li><a href="http://localhost:5000/learn">Learn</a></li>

                    <li><a href="http://localhost:5000/aboutUs">About Us</a></li>

                    <li><a><button class="login\_but">Login</button></a></li>

                </ul>

            </nav>

        </div>

        <div class="logo" >

            <img id="main-logo" src="./logo.png" usemap="#workmap" >

            <map name="workmap">

                <area shape="default" href="http://localhost:5000/home">

            </map>

        </div>

    </header>

    <div>

        <div>

            <button class="button1">Find Resources</button>

        </div>

        <div >

            <button class="button2">Give Back To<div>The<div>Community</div> </button>

        </div>

    </div>

    <div class="slider">

        <img id="slider-image" src="" alt="Slider Image">

    </div>

    <footer>

        <div class="container1">

            <div class="footer-left">

                <img id="footer-logo" src="./logo.png" alt="Logo">

            </div>

            <div class="footer-links">

                <ul>

                    <li id="contact"><b>Contact</b>

                        <ul>

                            <li>Phone: +91 7412589632</li>

                            <li>Email: mail@ResLocate.in</li>

                            <li><a id="contri-link" href="https://docs.google.com/forms/d/e/1FAIpQLSevGwMErB9sGiPtax7SKSB7\_2nT2jDhaxjnitWlwZ3hJJbZcg/viewform">Become a Contributor</a></li>

                        </ul>

                    </li>

                    <li id="about us"><b>About Us</b>

                        <ul>

                            <li>About Us</li>

                            <li>Privacy Policy</li>

                            <li>Terms and Conditions</li>

                        </ul>

                    </li>

                    <li id="Account"><b>Account</b>

                        <ul>

                            <li>Login</li>

                            <li>Help</li>

                        </ul>

                    </li>

                </ul>

            </div>

        </div>

    </footer>

<script src="Main\_script.js"></script>

</body>

</html>

**Login Page**

<html>

<head>

    <link rel="stylesheet" href="Login\_style.css">

    <title>ResLocate</title>

</head>

<body style="background-color:darkcyan;">

    <header>

        <div class="container">

            <h1 class="name">ResLocate</h1>

            <nav>

                <ul>

                    <li><a href="http://localhost:5000/learn">Learn</a></li>

                    <li><a href="http://localhost:5000/aboutUs">About Us</a></li>

                </ul>

            </nav>

        </div>

        <div class="logo" >

            <img id="l" src="./logo.png" usemap="#workmap" >

            <map name="workmap">

                <area shape="default" href="http://localhost:5000/home">

            </map>

        </div>

    </header>

    <div class="hero">

        <div class="form-box">

            <div class="button-box">

                <div id="btn"></div>

                <h2 type="button" class="toggle-btn" ><b>Login</b></h2>

            </div>

            <form  id="login" class="input">

                <label for="text" class="username" >Username</label>

                <input type="text" class="input-field" id="name\_inp" placeholder="John\_Doe" required>

                <label for="password" class="username" >Password</label>

                <input type="password" class="input-field" id= "pass\_inp" placeholder="Password" required>

            </form>

            <button  type ="submit" class="sumbit-btn" id="submit"><b>Login</b></button>

            <center><h3 hidden class="auth\_fail">Invalid Username Or Password !</h3></center>

        </div>

    </div>

    <footer>

        <div class="container1">

            <div class="footer-left">

                <img id="footer-logo" src="./logo.png" alt="Logo">

            </div>

            <div class="footer-links">

                <ul>

                    <li id="contact"><b>Contact</b>

                        <ul>

                            <li>Phone: +91 7412589632</li>

                            <li>Email: mail@ResLocate.in</li>

                            <li><a id="contri-link" href="https://docs.google.com/forms/d/e/1FAIpQLSevGwMErB9sGiPtax7SKSB7\_2nT2jDhaxjnitWlwZ3hJJbZcg/viewform">Become a Contributor</a></li>

                        </ul>

                    </li>

                    <li id="about us"><b>About Us</b>

                        <ul>

                            <li>About Us</li>

                            <li>Privacy Policy</li>

                            <li>Terms and Conditions</li>

                        </ul>

                    </li>

                    <li id="Account"><b>Account</b>

                        <ul>

                            <li>Login</li>

                            <li>Help</li>

                        </ul>

                    </li>

                </ul>

            </div>

        </div>

    </footer>

    <script src="Login\_script.js"></script>

</body>

</html>

**8 .FUTURE ENHANCEMENTS**

While ResLocate is certainly heading in the right direction, it can never be considered complete, new requirements can always be accommodated and obsolete ones removed. If given a chance of updating the website in the near future following modules could be considered to enhance the website further:

• Multi Language Support

• More Resources to Locate

• Refinement Of UI

• Development of a mobile application

**9. CONCLUSION**

In conclusion, the Water Locator and Free Food initiative has successfully developed an intuitive platform that empowers individuals to locate nearby water sources and access free food resources effortlessly. This innovative solution eliminates the challenges associated with finding clean water and nutritious meals, providing a lifeline for those in need. With features such as real-time location tracking and resource availability updates, the Water Locator and Free Food platform ensures that essential resources are accessible to everyone, regardless of their circumstances. Overall, this initiative represents a significant stride in addressing basic human needs and fostering community support in the modern era.

**10. REFERENCES**

[1] "WaterAid," WaterAid, [Online]. Available: <<https://www.wateraid.org/in/blog/water-scarcity#:~:text=There%20are%20several%20causes%20contributing,have%20depleted%20crucial%20water%20sources>>

[2] "Food security in India," Wikipedia, the free encyclopedia, [Online]. Available: <[https://en.wikipedia.org/wiki/Food\_security\_in\_India#:~:text=Food%20security%20has%20been%20a,111th%20out%20of%20125%20countries](https://en.wikipedia.org/wiki/Food_security_in_India%23:~:text=Food%20security%20has%20been%20a,111th%20out%20of%20125%20countries)>

[3] "who are we clean water action," clean water action, [Online]. Available: <<https://cleanwater.org/who-we-are>.>

[4] "52% of Indian population had internet access in 2022, says report," The Economic Times, [Online]. Available: <<https://economictimes.indiatimes.com/tech/technology/52-of-indian-population-had-internet-access-in-2022-says-report/articleshow/99964704.cms?from=mdr>>

[5] "global food security index 2022," economist impact, [Online]. Available: <<https://impact.economist.com/sustainability/project/food-security-index/>>

**11. APPENDIX**

**Login.js**

let Uname=document.getElementById('name\_inp');

let pass=document.getElementById('pass\_inp');

let button=document.querySelector('.sumbit-btn');

button.addEventListener("click",()=>{

    let obj={

        username: Uname.value,

        password: pass.value

    }

    fetch("/auth",{

        method:"POST",

        headers:{

            "Content-Type":"application/json"

        },

        body: JSON.stringify(obj)

    }).then(response => response.json()).then(auth => {

        if(auth.auth==1){

            window.location.href = "http://localhost:5000/Contributors\_page";

        }

        else{

            document.querySelector(".auth\_fail").removeAttribute("hidden");

        }

    })

})

**INDEX.JS**

// Requirements-

const express= require('express');

const { MongoClient, ServerApiVersion } = require('mongodb');

// Create Express App-

const app=express();

app.use(express.json());

app.use(express.static(\_\_dirname+"/client"));

// Connect Mongo Client-

const uri = "mongodb+srv://Hardik:1234@mdb.eo3b4jv.mongodb.net/?retryWrites=true&w=majority";

// Create a MongoClient with a MongoClientOptions object to set the Stable API version

const client = new MongoClient(uri, {

  serverApi: {

    version: ServerApiVersion.v1,

    strict: true,

    deprecationErrors: true,

  }

});

// CRUD Functions Mongo-

// List Databases

async function listDatabases(client){

    const dblist= await client.db().admin().listDatabases();

    console.log("Databases:");

    dblist.databases.forEach(db => {

        console.log(`- ${db.name}`);

    });

}

// Create

async function create(client,data,table){

    const result= await client.db("Main").collection(table).insertOne(data);

    console.log(`New Data Inserted: ${result.insertedID}`);

}

// Read

async function read(client,data){

    const res=await client.db("Main").collection("Users").findOne({ username: data });

    if (res){

        return res;

    }

}

// Read Mult.

async function readMult(client,data){

    const res=await client.db("Main").collection("Resource Data").find({ "coords.lat": {$gte: 0} });

    const result=await res.toArray();

    if(res){

        return result;

    }

    else{

        console.log("No DATA!");

    }

}

// Delete

async function del(client,data){

    const result=await client.db("Main").collection("Users").deleteOne({\_id:data});

    console.log("Deleted!");

}

//-------------------------------------------

// Calculate diatance-

function rad(degrees){

  var pi = Math.PI;

  return degrees \* (pi/180);

}

function distance(Lat1,Lat2,Lon1,Lon2){

  let dist=Math.acos((Math.sin(rad(Lat1))\*Math.sin(rad(Lat2)))+(Math.cos(rad(Lat1))\*Math.cos(rad(Lat2)))\*(Math.cos(rad(Lon2)-rad(Lon1))))\*6371

  return dist;

}

//-------------------------------------------

// Main Runner-

async function run() {

    try {

      // Vardiables-

      let is\_logged\_in=0; // 0- Not Logged In | 1- Logged In.

      let lat=0;

      let lon=0;

      //-------------------------

      // Connect the client to the server (optional starting in v4.7)

      await client.connect();

      // Send a ping to confirm a successful connection

      await client.db("admin").command({ ping: 1 });

      console.log("Pinged your deployment. You successfully connected to MongoDB!");

      app.get("/",(req,res)=>{

        res.sendStatus(200).status("Good");

      })

// Init Pages--------------------------------------------------

      app.get("/login",(req,res)=>{

        res.sendFile(\_\_dirname + "/Client/login.html");

      })

      app.get("/Contributors\_page",(req,res)=>{

        res.sendFile(\_\_dirname + "/Client/Contributors.html");

      })

      app.get("/home",(req,res)=>{

        res.sendFile(\_\_dirname + "/Client/main.html");

      })

      app.get("/aboutUs",(req,res)=>{

        res.sendFile(\_\_dirname + "/Client/aboutUs.html");

      })

      app.get("/learn",(req,res)=>{

        res.sendFile(\_\_dirname + "/Client/learn.html");

      })

      app.get("/maps",(req,res)=>{

        res.sendFile(\_\_dirname + "/Client/GoogleMaps.html");

      })

      app.get("/AddRec",(req,res)=>{

        res.sendFile(\_\_dirname + "/Client/AddRec.html");

      })

      app.get("/findRes",(req,res)=>{

        res.sendFile(\_\_dirname + "/Client/find\_res.html");

      })

      app.get("/findWork",(req,res)=>{

        res.sendFile(\_\_dirname + "/Client/Find\_Work.html");

      })

      app.get("/AddWork",(req,res)=>{

        res.sendFile(\_\_dirname + "/Client/AddWork.html");

      })

//-------------------------------------------------------------------

// User Auth-

      app.post("/auth",async function(req,res){

          const result= await read(client,req.body.username);

          //console.log("res-",result);

          if(result){

            if(result.password == req.body.password){

              is\_logged\_in=1;

              res.json({

                auth: 1

              })

            }

            else{

              is\_logged\_in=0;

              res.json({

                auth: 0

              })

            }

          }

          else{

            res.json({

              auth: 0

            })

          }

      })

//-------------------------------------------------------------------

// Logout-

      app.get("/logout",(req,res)=>{

        //console.log("logging Out: ",is\_logged\_in);

        is\_logged\_in=0;

        res.json({

          log: 0

        })

      })

//-------------------------------------------------------------------

// check loging-

      app.get("/check",(req,res)=>{

        //console.log("Ckecking login");

        if(is\_logged\_in==0){

          res.json({

            log: 0

          })

        }

        else{

          res.json({

            log: 1

          })

        }

      })

//-------------------------------------------------------------------

// Get Locations-

      app.post("/getLoc",async function(req,res){

        const lat=req.body.lat;

        const lon=req.body.lon;

        const agg = [

          {

            '$geoNear': {

              'near': {

                'type': 'Point',

                'coordinates': [

                  lat,lon

                ]

              },

              'distanceField': 'distance',

              "distanceMultiplier" : 0.001,

              'maxDistance': 3000,

              'spherical': false

            }

          }

        ];

        // console.log(agg);

        // console.log(lat,lon);

        const coll = client.db('Main').collection('Resource Data');

        const cursor = coll.aggregate(agg);

        const result = await cursor.toArray();

        //console.log(result);

        res.json(result);

      })

//-------------------------------------------------------------------

// Get Locations-

app.post("/getLocWork", async (req, res) => {

  const lat = req.body.lat;

  const lon = req.body.lon;

  const agg = [

    {

      '$geoNear': {

        'near': {

          'type': 'Point',

          'coordinates': [lat, lon]

        },

        'distanceField': 'distance',

        'maxDistance': 5000,

        'spherical': false

      }

    }

  ];

  const coll = client.db('Main').collection('Work Data');

  const cursor = coll.aggregate(agg);

  try {

    const result = await cursor.toArray();

    //console.log(result);

    res.json(result);

  } catch (error) {

    console.error(error);

    res.status(500).json({ error: 'Internal Server Error' });

  }

});

//-------------------------------------------------------------------

// Add Resources---------------------

      app.post("/sendCords",(req,res)=>{

        lat=req.body.lat;

        lon=req.body.lon;

      })

      app.get("/getCord",(req,res)=>{

        res.json({

          latC: lat,

          lonC: lon

        })

      })

      app.post("/sendData",async (req,res) => {

        let obj=req.body;

        const count=(await client.db("Main").collection("Resource Data").countDocuments())+1;

        obj = Object.assign({\_id:count},obj)

        create(client,obj,"Resource Data"); // Create Doc.

        //console.log("Data Created");

        res.json({

          log:1

        })

      })

// Add Wok data-

      app.post("/sendWorkData",async function(req,res){

        let obj=req.body;

        const count=(await client.db("Main").collection("Work Data").countDocuments())+1; // get next id and push it to object.

        obj = Object.assign({\_id:count},obj)

        //console.log(obj);

        create(client,obj,"Work Data"); // Create Doc.

        //console.log("Data Created");

        res.json({

          log:1

        })

      })

//-------------------------------------------------------------------

  } //Dont touch.

//-------------------------------------------------------------------

  finally {

    //Ensures that the client will close when you finish/error

    //await client.close();

  }

}

run().catch(console.dir);

app.listen(5000,()=> {console.log("Server Started On Port 5000")});