**PROJECT TITLE:** DISASTER RESPONSE FOR FISHING COMMUNITIES

**PROBLEM STATEMENT:**

-Design a mobile app that alerts fishing communities about cyclones or tsunamis with evacuation routes and safety protocols.

**PROJECT DESCRIPTION:**

**SAFE SEA** is an interactive disaster preparedness and response web platform developed to support coastal communities during natural emergencies such as cyclones, tsunamis, and storm surges. The system combines real-time weather tracking, smart alert systems, location-based evacuation guidance, and community collaboration tools to improve public awareness and safety.

**Key Features:**

**1. Real-time Weather Updates**

This feature displays constantly updated weather data specific to coastal regions. Information includes:

* **Temperature:** Helps users assess general climate and risk of heat or cold-related stress.
* **Humidity:** Useful for understanding air moisture levels, which can impact comfort and health.
* **Wind Speed:** Crucial for detecting approaching storms or cyclones.
* **Wave Height:** Important for fishermen and coastal dwellers to evaluate sea conditions.
* **Weather Conditions:** Summarizes overall weather (e.g., stormy, cloudy, clear) for decision-making.

These updates ensure users stay aware of environmental changes and can make timely decisions.

**2. Alerts & Notifications**

The system provides automated alerts and push-style notifications based on real-time data and risk levels. Alerts are categorized into:

* **High Priority (Red):** Immediate danger, e.g., cyclone landfall or tsunami warning.
* **Medium Priority (Yellow):** Potential threat, such as tropical storms or high tides.
* **Low Priority (Green):** General safety information or advisory messages.

The visual alert system ensures quick user understanding and response, even in high-stress situations.

**3. Evacuation Routes**

This feature offers predefinedandlocation**-**sensitiveevacuationroutes:

* Users can either **enter their current location** or allow location access.
* The system calculates the **nearest safest evacuation path**.
* Routes are visually displayed and labelled (e.g., Route A, Route B, etc.).
* In future versions, route safety will be dynamically adjusted based on weather and road conditions.

This ensures faster evacuation and guides users to safety with minimal confusion.

**4. Safety Protocols**

This section provides step-by-step safety guidelines for various disaster scenarios, such as:

* **Before a cyclone or tsunami:** How to prepare an emergency kit, secure property, and stay informed.
* **During the disaster:** Evacuation tips, dos and don’ts, and how to stay safe indoors or outdoors.
* **After the event:** Recovery advice, first aid, and steps to return safely to homes.

Clear protocols help reduce panic and ensure users act wisely during emergencies.

**5. Emergency Contacts**

This feature lists critical contact numbers and simulation-based call buttons for:

* Coast guard
* Disaster management authorities
* Emergency helplines (ambulance, police, etc.)
* Local support centers

Users can simulate calling these services directly from the interface, enabling faster coordination and help-seeking.

**6. Community Forum**

An interactive space for users to:

* Share live updates or warnings
* Ask questions and clarify doubts
* Offer or request assistance
* Build a support network during crisis times

This forum encourages **peer-to-peer communication** and fosters a sense of shared responsibility and support during disasters.

**7. Emergency Resources**

This feature shows nearby critical facilities, including:

* Relief shelters
* Hospitals and clinics
* Rescue operation centers
* Food/water supply distribution points.

**INSTALLATION AND SET UP INSTRUCTIONS:**

**1. Introduction**

This document provides a step-by-step explanation of how to install and run the **SAFE SEA** website on your local system for development, testing, or demonstration purposes. This web application is designed to help coastal communities stay alert and evacuate safely during natural disasters such as cyclones and tsunamis.

**2. Prerequisites**

Before setting up the project, ensure the following software and tools are available on your system:

* **Visual Studio Code (VS Code)** – A code editor for viewing and editing files.
* **Live Server Extension** – A VS Code extension to preview the website in a browser with live reloading.
* **Web Browser** – Such as Google Chrome, Mozilla Firefox, or Microsoft Edge.
* **Internet Connection** – Required to access external APIs (like weather or maps) if integrated.

**3. Downloading and Preparing Project Files**

* Obtain the website project folder, which includes files such as:
* index.html (Main page)
* style.css (Styling)
* script.js (JavaScript functionality)
* Any additional assets like images or libraries
* If the files are in a .zip archive:
* Right-click the .zip file and select **Extract All**.
* Choose a destination folder and extract the contents.

**4. Opening the Project in VS Code**

* Launch **Visual Studio Code**.
* Click on **File → Open Folder**.
* Browse to the extracted project folder and open it.
* You will now see all the website files listed in the Explorer panel.

**5. Installing Live Server Extension (If Not Already Installed)**

* In VS Code, click on the **Extensions** icon on the left sidebar or press Ctrl + Shift + X.
* In the search bar, type **Live Server**.
* Click the **Install** button on the extension published by *Ritwick Dey*.
* This extension will allow you to launch the website in your default browser and view real-time changes.

**6. Running the Project Locally**

* Open the index.html file in VS Code.
* Right-click anywhere inside the file.
* Select **“Open with Live Server.”**
* A new browser window or tab will open with a URL like:  
  http://127.0.0.1:5500/index.html
* Your website is now live locally and ready to use.

**7. Testing Key Features**

Once the website is running, you can explore and test its functionalities:

* **Real-Time Alerts:** View color-coded warning levels for safe, caution, or danger zones.
* **Evacuation Routes:** View suggested safe routes displayed as labels or in a map (if integrated).
* **Location Input:** Enter your location or allow browser access for suggestions.
* **Emergency Contacts & Safety Tips:** Review important phone numbers and protocols.
* **Community Forum:** Simulate messages or discussions for emergency collaboration.
* **Weather Data:** View temperature, humidity, and wind details dynamically.

**TECH STACK USED:**

The SAFE SEA website uses a front-end focused tech stack, along with optional integrations for external APIs and interactive mapping. Below is the complete breakdown of the technologies used:

**Frontend Technologies**

* **HTML5**  
  Used to create the basic structure and layout of the website pages, such as headers, sections, buttons, and forms.
* **CSS3**  
  Responsible for the visual styling of the website, including colors, fonts, layout positioning, animations, and responsiveness across devices.
* **JavaScript (Vanilla JS)**  
  Adds interactivity to the website such as:
  + Handling user input
  + Displaying alerts dynamically
  + Location-based logic
  + Simulating emergency calls or interactions

**Mapping and Geolocation**

* **Leaflet.js**   
  A lightweight open-source JavaScript library used to display interactive maps (using OpenStreetMap tiles). Useful for showing evacuation routes visually.
* **OpenStreetMap** *(used with Leaflet)*  
  Provides free map tiles and geospatial data for rendering evacuation paths and shelter locations on the map.
* **Geolocation API (Browser-based)**  
  Detects the user's current location using the browser, allowing the system to suggest the nearest safe evacuation route.

**External Services / APIs (Optional Integrations)**

* **OpenWeatherMap API** *(Optional)*  
  Used to fetch real-time weather data such as temperature, wind speed, humidity, wave height, and weather conditions. Requires API key.

**Development Tools**

* **Visual Studio Code (VS Code)**  
  A popular source code editor used for writing, editing, and managing the website codebase.
* **Live Server Extension (VS Code Plugin)**  
  Provides a local development server with live reload capability. Automatically reflects changes in the browser as you code.
* **Web Browsers (Chrome, Firefox, Edge)**  
  Used to view, test, and interact with the website during development.

**HOW TO RUN THE PROJECT:**

**Steps to Run the Project**

**✅ Step 1: Locate the HTML File**

* Ensure you have the main file: index.html (or it may be named safe\_sea\_website.html; you can rename it to index.html for convenience).
* This file contains the complete code including design (CSS), content (HTML), and functionality (JavaScript).

**✅ Step 2: Open the File in a Web Browser**

* Right-click on the file and choose **“Open with” → your preferred browser** (e.g., Chrome, Firefox, Edge).
* Alternatively, **double-click** the file, and it will open automatically in the default browser.
* The website will load, and you can start interacting with all the sections like Dashboard, Alerts, Safety Protocols, etc.

✅ This is the easiest and fastest way to run the project without any tools.

**Optional: Run with a Local Server (for Developers)**

If you want to test the website as it would behave in a live environment or make code changes, you can run it using a local development server.

**Method A: Using VS Code + Live Server**

1. Open the project folder in **Visual Studio Code**.
2. Install the **Live Server** extension (from the Extensions tab).
3. Right-click on index.html → Click **“Open with Live Server”**.
4. The site will open in a browser window (e.g., http://127.0.0.1:5500/).

**Method B: Using Python HTTP Server**

1. Open **Command Prompt or Terminal**.
2. Navigate to the folder containing the file:

bash

CopyEdit

cd path/to/project-folder

1. Start a local server with the following command:

nginx

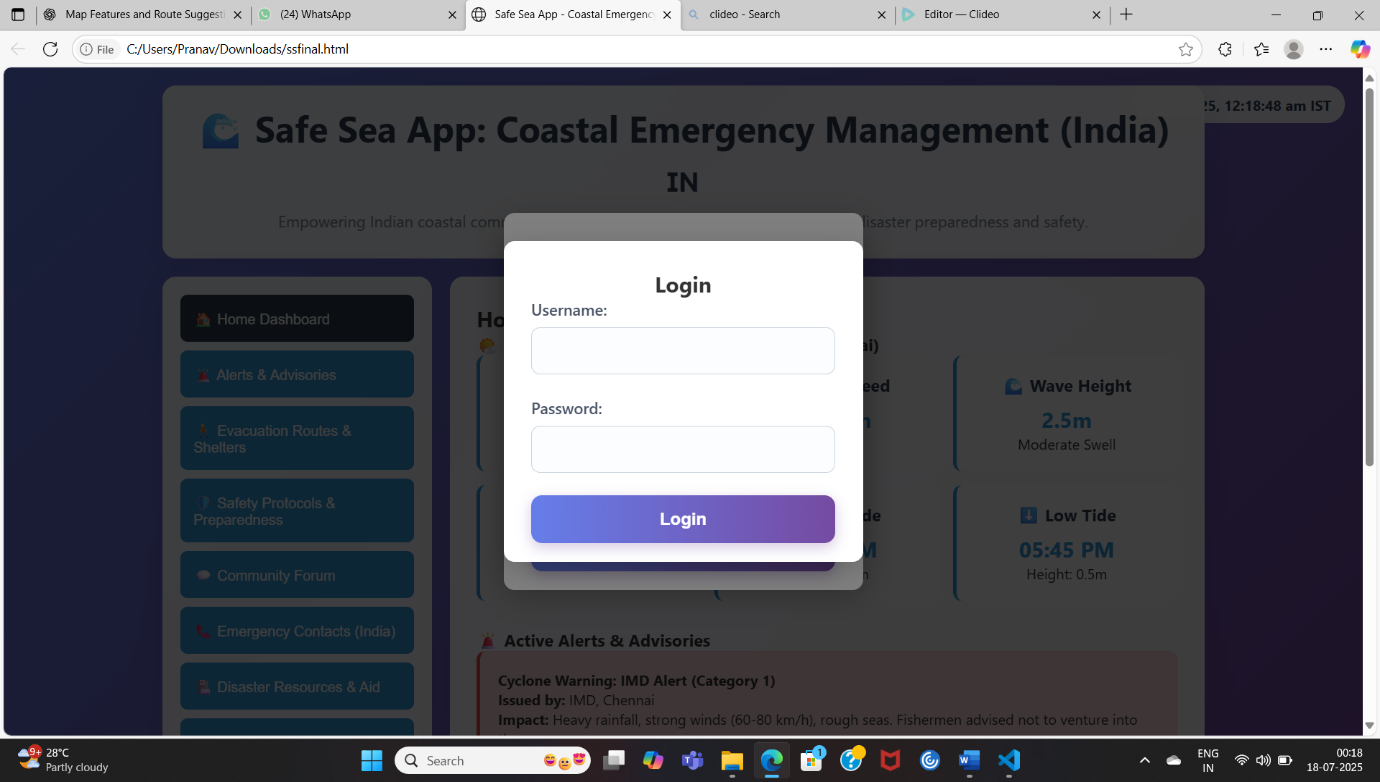
CopyEdit

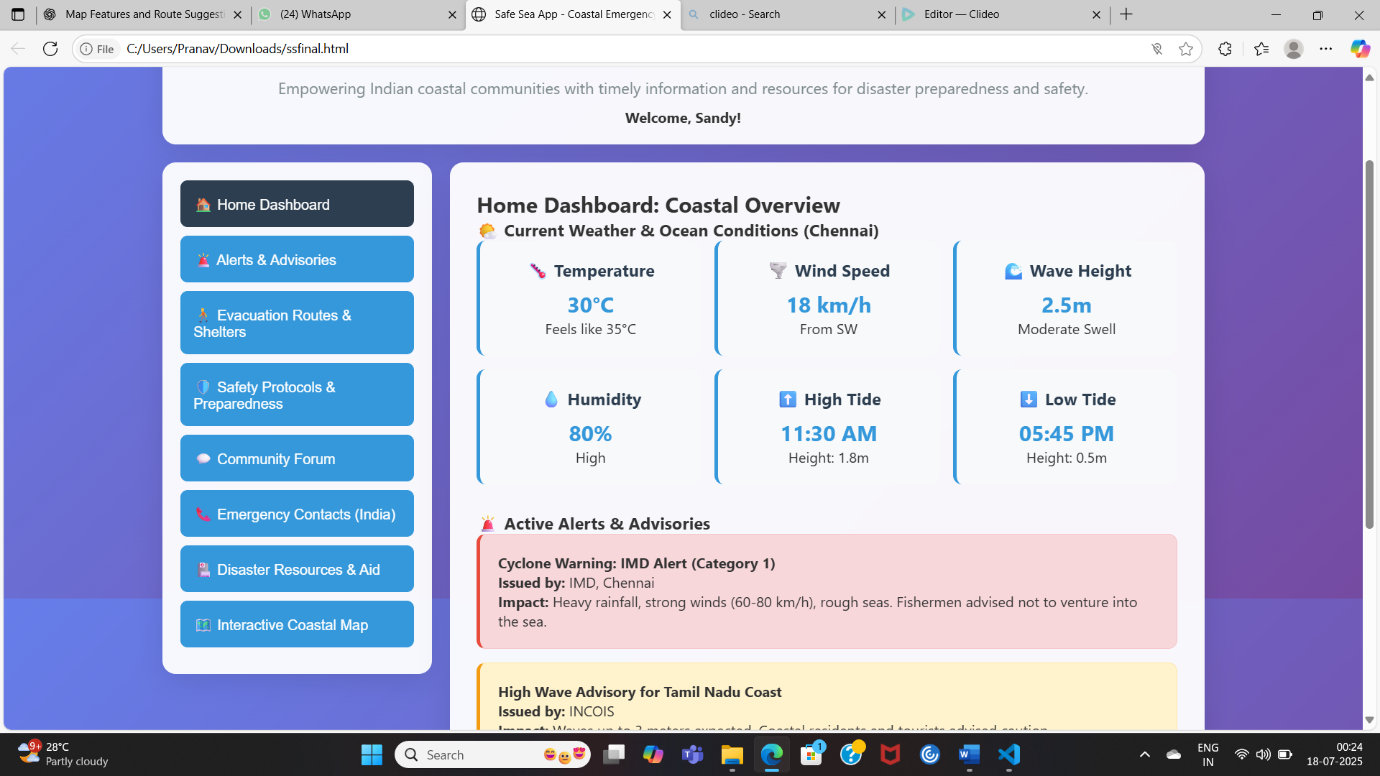
python -m http.server

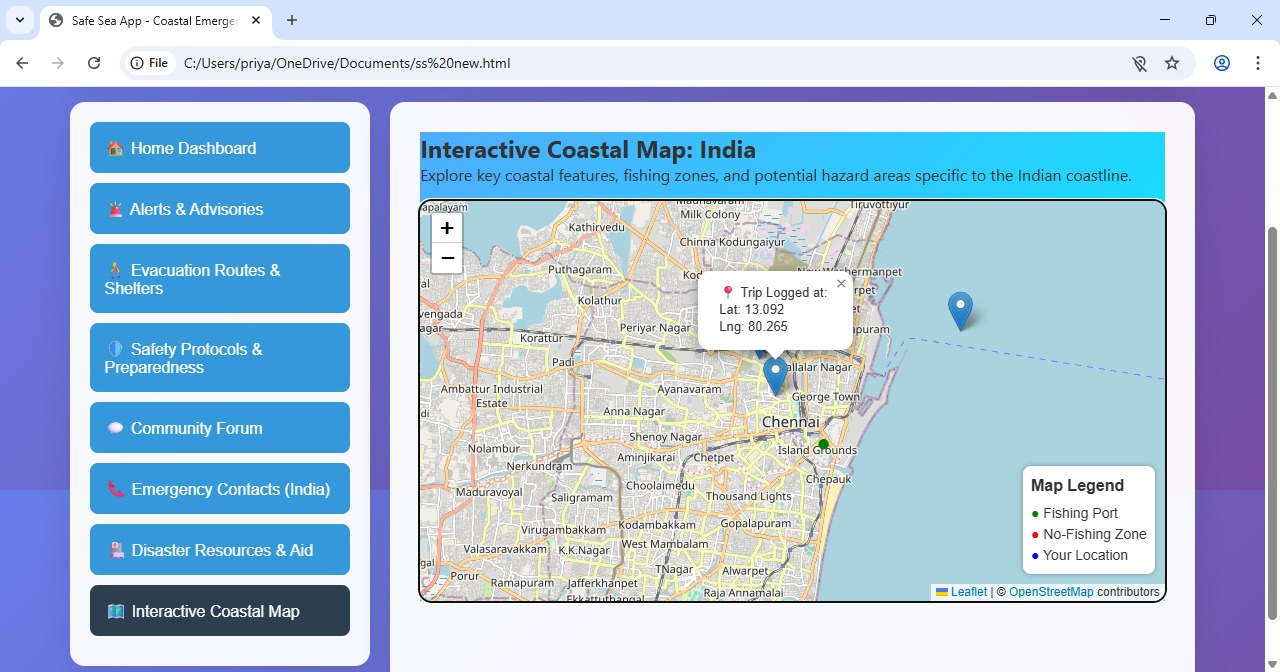
1. Open your browser and go to: http://localhost:8000

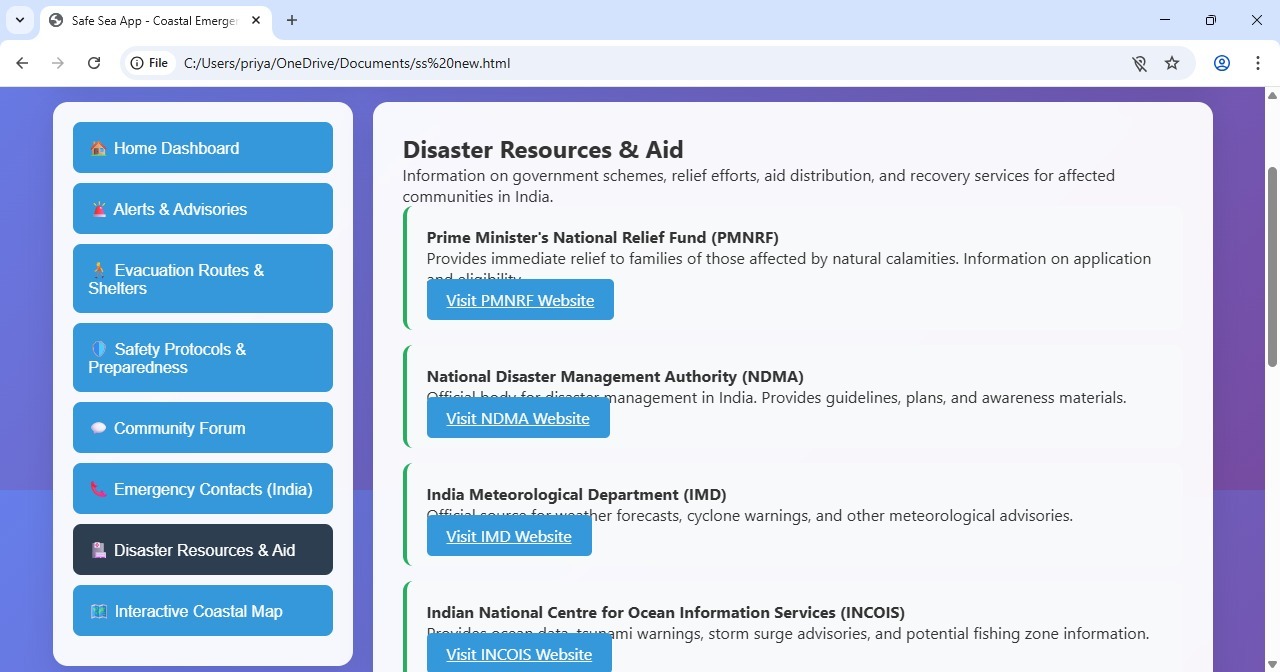
These methods are helpful if you want to simulate a hosted environment or debug changes during development.

**SCREENSHOTS OF OUR WEBSITE:**





****

****

**YOUTUBE LINK:**

<https://youtube.com/shorts/O4HXo6ZIIJU?si=0IluLowJkEoZH-F1>

**TEAM MEMBERS:S**

SANDHYALAKSHMI.S

PRIYANKA.K.

NITHYAVARSHINI.C.N.R.