

USER MANUAL

Date: 19 December 2024



**MONTCLAIR STATE
UNIVERSITY**

Course: CSIT 515 Software Engineering & Reliability - Fall 2024

Instructor: Dr. Hubert A. Johnson

**Environmental Monitoring and Reporting System
(EMRS)**

USER MANUAL**UNITY CODERS****Team members:**

UBBANI SAMSON RAJ,

PASAM PRATHUSHA,

RAHEEMA BEGUM SHAIK,

KAITHAPALLI ABHISHEK,

KAMALA SREE VALLI TATIPATIPARTHI

Table of Contents

LIST OF FIGURES	3
1. Introduction	5
1.1 Overview of the Environmental Monitoring and Reporting System (EMRS).....	5
1.2 Purpose and Benefits	5
1.3 Key Features	5
2. System Requirements	6
2.1. Hardware Requirements.....	6
2.1.1 For Users (Client Machines):	6
2.1.2 For Servers (Hosting the EMRS):.....	6
2.2. Software Requirements	7
2.3. Network Requirements	7
2.4. AI Component Requirements	8
3. Project Folder Structure	9
4. Installation Instructions.....	10
4.1 Installation Prerequisites	10
4.2 Setting Up the Project.....	10
5. Features of EMRS.....	14
5.1 User Capabilities	14
6. How AI Works in the Project	18
6.1 AI Features	18
6.2 Steps to Run the AI Module.....	19
7. Troubleshooting.....	20
7.1 Common Issues and Solutions	20
7.2 System Logs and Diagnostics	20

LIST OF FIGURES

Figure number	Figure name	Page number
1	Flask Server	13
2	XAMPP server	13
3	Welcome Page	14
4	Report Submit Page	15
5	All reports	16
6	Climate Impact Data Visualization	16
7	Predicted Impact Data	17
8	Community Discussions	17
9	Real-time Alerts Map	18
10	Environmental Tips	18
11	Admin Dashboard	19
12	Flask server in AI folder	20
13	AI server for real-time predictions	20
14	Impact Visualization	20

15	Admin Page Console Log	21
----	------------------------	----

1. Introduction

1.1 Overview of the Environmental Monitoring and Reporting System (EMRS)

The **Environmental Monitoring and Reporting System (EMRS)** is a comprehensive, user-friendly platform designed to empower individuals, communities, and organizations to actively participate in combating environmental crimes and addressing climate change. The system integrates real-time data reporting, AI-powered climate visualizations, and interactive community features to foster environmental awareness, collaboration, and accountability.

This web-based system allows users to report environmental crimes such as pollution, deforestation, illegal fishing, and more. It features advanced tools for visualizing the impact of environmental activities, analyzing climate data trends, and engaging in community discussions. The system is equipped with multilingual support and real-time alerts to ensure inclusivity and rapid response.

1.2 Purpose and Benefits

The purpose of EMRS is to bridge the gap between citizens, environmental organizations, and authorities, facilitating a collaborative approach to addressing environmental challenges. The system provides:

- **Real-Time Reporting:** Users can report environmental violations instantly, including details like location, images, and descriptions.
- **AI-Powered Insights:** The system utilizes artificial intelligence to analyze reports and predict environmental impacts.
- **Community Engagement:** A platform for discussions, sharing solutions, and raising awareness about pressing environmental issues.
- **Data Visualization:** Interactive tools to visualize environmental trends and impacts for better understanding and action planning.
- **Alerts and Tips:** Real-time alerts for critical environmental issues and tips to promote eco-friendly behavior.

1.3 Key Features

- **User-Friendly Interface:** A simple, intuitive design that ensures accessibility for users with varying levels of technical expertise.
- **Multilingual Support:** Available in multiple languages to cater to diverse user demographics.

- **AI Integration:** Advanced machine learning models predict the scale and severity of environmental impacts.
 - **Real-Time Alerts:** A live map to visualize ongoing environmental activities and reported crimes.
 - **Community Platform:** Engage with other users to discuss, share, and propose solutions for environmental issues.
 - **Comprehensive Reporting:** Submit detailed reports with multimedia attachments for better documentation of incidents.
-

2. System Requirements

This section outlines the technical and system requirements needed to install, run, and use the **Environmental Monitoring and Reporting System (EMRS)**. It explains what hardware, software, and network configurations are necessary for both users and administrators.

2.1. Hardware Requirements

This part explains the physical devices (like computers and servers) needed to run the system smoothly.

2.1.1 For Users (Client Machines):

- **Processor:** At least a Dual-Core Processor to handle website operations like browsing, submitting reports, or viewing maps.
- **Memory (RAM):** Minimum 2 GB, but 4 GB is recommended for better performance.
- **Storage:** 500 MB free space is required to store browser cache and files.
- **Display:** The user interface works best with a screen resolution of 1024x768 or higher.

2.1.2 For Servers (Hosting the EMRS):

- **Processor:** A more powerful Quad-Core Processor for handling multiple requests and running background services.
- **Memory (RAM):** At least 8 GB, as the server hosts the database, the AI, and handles user requests.

- **Storage:** At least 10 GB of free disk space for storing reports, uploaded files, and the database.
- **Network:** A stable internet connection is required for users to access the website.

Why This Matters: Good hardware ensures the system runs without delays or crashes, providing users with a smooth experience.

2.2. Software Requirements

This section lists the software tools required for the system to function properly.

1. For Users:

- A modern web browser like **Google Chrome, Firefox, or Edge** to access the system. Ensure JavaScript and Cookies are enabled for interactive features.

2. For Server Hosting the System:

- **Operating System:** A modern OS like Windows 10/11 or Ubuntu 20.04 for compatibility with web servers and databases.
- **Database Software:** **XAMPP**, which includes MySQL and Apache to host the system and store reports.
- **Programming Environment:** The system is built using **PHP 7.4+**, so this version or higher must be installed.

3. For the AI Component:

- The AI module uses Python, Flask, and other libraries to process data and generate predictions. Python dependencies include Flask for the web framework, pandas for handling data, and scikit-learn for machine learning tasks.

Why This Matters: Having the right software ensures the system runs as intended, with all features working correctly.

2.3. Network Requirements

This explains the internet and networking setup needed to run the system.

1. For General Use:

- Users need a stable internet connection with a speed of at least 5 Mbps to access real-time features, maps, and reports.

2. For Hosting the System:

- If the system is hosted online, a **public IP address** or a registered domain is needed for users to access it.
- **Port Forwarding:** Ensure ports 80 (HTTP) and 443 (HTTPS) are open for web traffic.

Why This Matters: The system relies on the internet for real-time features like alerts and updates, so a reliable connection is essential.

2.4. AI Component Requirements

The AI module processes environmental data and generates insights. This section lists the specific needs for running the AI component.

1. Hardware for AI:

- A powerful processor (Intel i5 or higher) ensures the AI can handle computations efficiently.
- **RAM:** At least 8 GB (or 16 GB for faster processing).
- **GPU:** A graphics processor is optional but useful for running heavy machine learning models.

2. Software for AI:

- Python libraries:
 - **Flask:** For running the AI as a web service.
 - **NumPy & pandas:** For managing and analyzing data.
 - **scikit-learn:** For running machine learning models.
 - **matplotlib:** For visualizing data.
 - **requests:** For API communication between the AI and the main system.

Why This Matters: The AI module enhances the system by providing intelligent insights (e.g., analyzing climate impact), so its dependencies must be set up correctly.

By following the hardware, software, and network requirements listed here, the **Environmental Monitoring and Reporting System** will run smoothly, providing users and administrators with a seamless experience.

3. Project Folder Structure

EMRS/

```
|— public/
|   |— assets/
|   |   |— css/
|   |   |— js/
|   |   |— images/
|   |— dashboard.php
|   |— login.php
|   |— logout.php
|   |— ...
|— config/
|   |— Database.php
|   |— ...
|— ai/
|   |— model/
|   |   |— model.pkl
|   |— app.py
|   |— requirements.txt
|— uploads/
|— logs/
|— emrs.sql
|— README.md
```

4. Installation Instructions

This section provides a step-by-step guide to install, configure, and run the **Environmental Monitoring and Reporting System (EMRS)**. Follow the instructions carefully to ensure the system is set up correctly.

4.1 Installation Prerequisites

Before setting up the system, ensure the following software is installed:

1. XAMPP (for PHP and MySQL):

- Download and install XAMPP from apachefriends.org.
- XAMPP provides an easy way to host PHP applications and manage MySQL databases.

2. Python 3.8 or Later:

- Download Python from python.org.
- Ensure that Python is added to the system PATH during installation.

3. Code Editor:

- Install a code editor like [Visual Studio Code](https://code.visualstudio.com/) for editing files and running commands.

Why These Are Needed:

- XAMPP is required to host the website and database.
- Python is essential for running the AI component.
- A code editor simplifies project configuration and execution.

4.2 Setting Up the Project

1. Extract the Project Files:

- Download and extract the project folder (named emrs).
- Place the extracted folder into the htdocs directory of your XAMPP installation (e.g., C:\xampp\htdocs\).

2. Create the Database:

- Open your browser and go to <http://localhost/phpmyadmin>.
- Create a new database named **emrs**.
- Import the provided emrs.sql file into the database using phpMyAdmin.

3. Update Database Configuration:

- Open the config/Database.php file in your code editor.
- Update the database credentials to match your local MySQL setup (e.g., username root and an empty password).

Why This Step is Important: The website won't function without the database, and correct configuration ensures the application can communicate with the database.

4.3 Setting Up the AI System

The AI component processes environmental data and provides insights. Follow these steps to configure it:

1. Navigate to the AI Folder:

- Open a terminal or command prompt.
- Navigate to the ai/ folder in the project directory.

2. Install Dependencies:

- Install the required Python libraries by running:
pip install -r requirements.txt
- The requirements.txt file contains libraries like Flask, pandas, and scikit-learn.

3. Start the Flask Server:

- Run the following command to start the AI server:
python ai_service.py
- Ensure the Flask server is running on <http://127.0.0.1:8000>. As shown in the figure below.

```

C:\xampp\htdocs\Softwareproject\emrs>python ai_service.py
Starting AI Service...
* Serving Flask app 'ai_service'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:8000
Press CTRL+C to quit
* Restarting with stat
Starting AI Service...
* Debugger is active!
* Debugger PIN: 135-114-912

```

Fig 1 Flask.

What the AI Does: The AI system analyzes environmental reports, predicts the impact of reported incidents, and provides climate-related insights.

Fig 1 stated that the flask is successfully running in the server with the port number <http://127.0.0.1:8000>.

4.4 Running the Project

1. Start Apache and MySQL Servers:

- Open the XAMPP control panel.
- Start the **Apache** and **MySQL** modules.

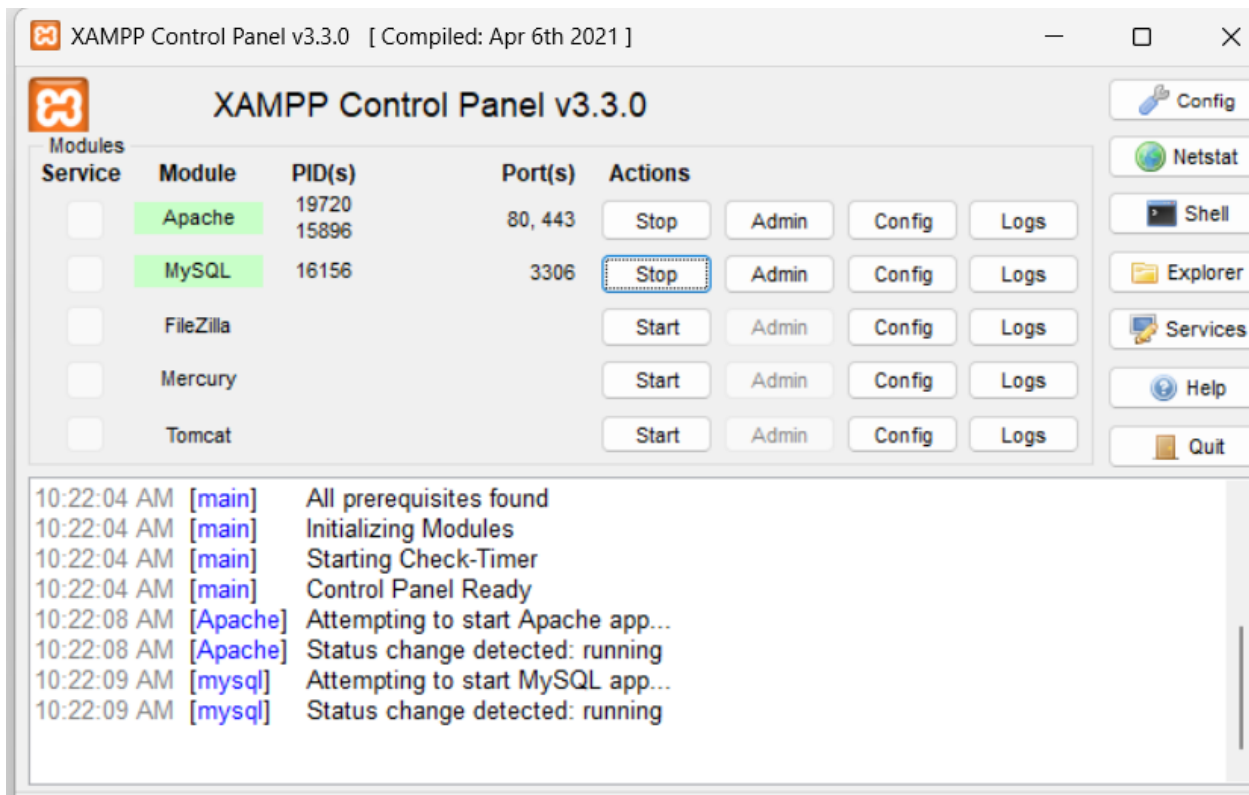


Fig 2 XAMPP server.

You will notice that apache and MySQL are successfully running as shown in the fig 2.

2. Access the Website:

- Open your browser and go to:
- <http://localhost/Softwareproject/emrs/public/index.php>
- You can open the index.php in the using <http://localhost/>

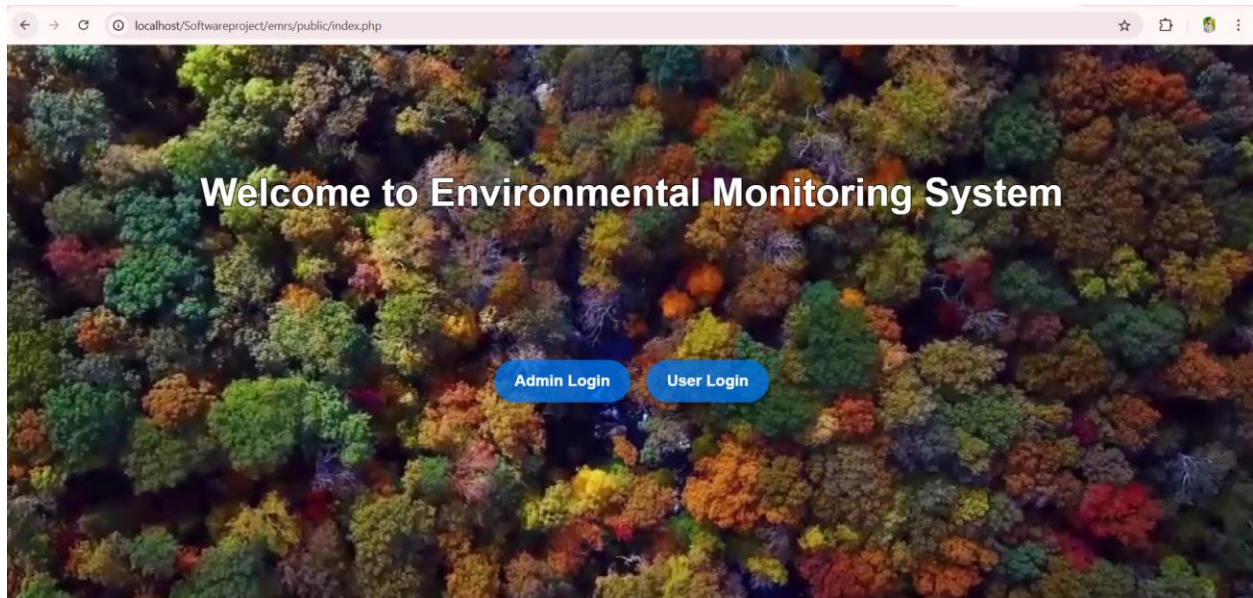


Fig 3 Welcome Page

On successful opening of the index.php you can see the welcome page for the project as shown in the fig 3.

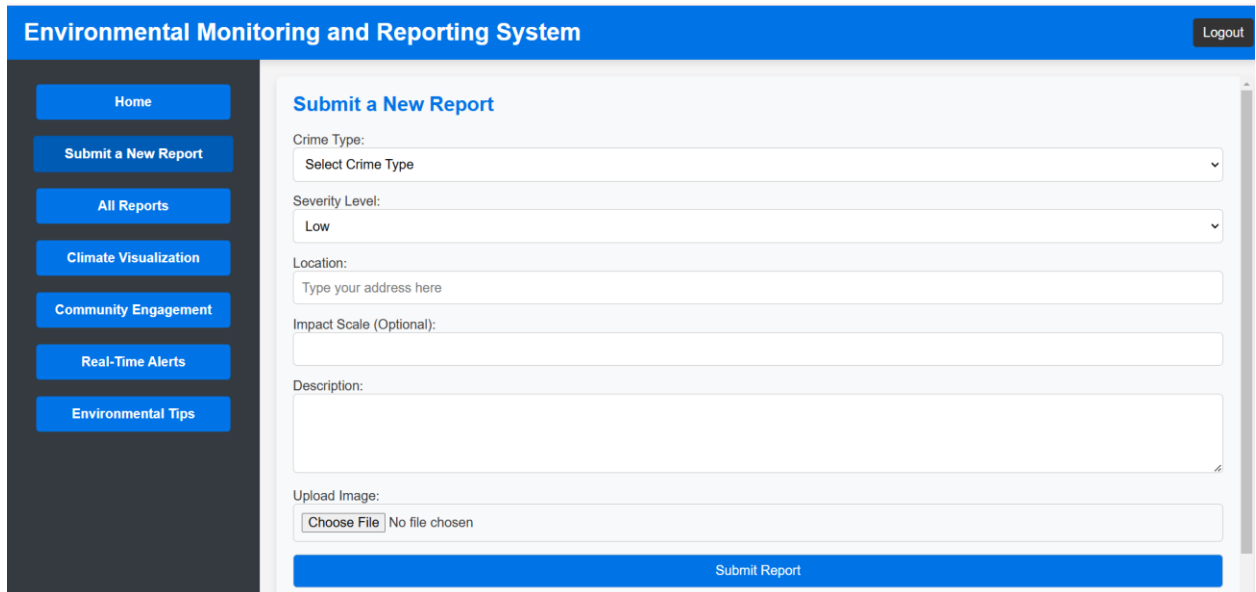
Why This Step is Important: Starting the Apache server allows the system to serve web pages, while MySQL handles the database operations.

By following these instructions, you will have the **Environmental Monitoring and Reporting System** fully operational on your local machine, with both the website and AI component working seamlessly.

5. Features of EMRS

5.1 User Capabilities

- **Submit Reports:** Users can report environmental crimes with details like crime type, severity, and images.



The screenshot displays the 'Environmental Monitoring and Reporting System' interface. On the left is a dark sidebar with blue buttons for 'Home', 'Submit a New Report', 'All Reports', 'Climate Visualization', 'Community Engagement', 'Real-Time Alerts', and 'Environmental Tips'. The main content area has a blue header with the system name and a 'Logout' button. Below the header, the 'Submit a New Report' form is shown with the following fields: 'Crime Type' (a dropdown menu with 'Select Crime Type'), 'Severity Level' (a dropdown menu with 'Low'), 'Location' (a text input with placeholder 'Type your address here'), 'Impact Scale (Optional)' (a text input), 'Description' (a large text area), and 'Upload Image' (a file upload button labeled 'Choose File' with 'No file chosen' text). A blue 'Submit Report' button is at the bottom of the form.

Fig 4. Submit Report.

Fig 4 shows that user can submit the form in environment crime that report will be directly save in the database and admin can see that report.

- **View Reports:** Browse all reports submitted by the community.

In fig 5 you can see that the user can browse all reports and see the count of the reports in the database.

- [illegible]

15

#	Location	Impact (sq.m)	Predicted Impact
1	20 Graham Street, Jersey City, NJ, USA	3000 sq.m	Low Impact
2	48 Manhattan Avenue, Jersey City, NJ, USA	0 sq.m	Low Impact
3	55 Central Avenue, Jersey City, NJ, USA	3000 sq.m	Moderate Impact
4	New York, NY	500 sq.m	Moderate Impact
5	Amazon Rainforest, Brazil	1000 sq.m	Low Impact
6	Pacific Ocean	200 sq.m	Low Impact
7	Serengeti National Park, Tanzania	50 sq.m	Low Impact
8	Los Angeles, CA	0 sq.m	Low Impact
9	South China Sea	500 sq.m	Moderate Impact
10	Kaziranga National Park, India	20 sq.m	Moderate Impact
11	Beijing, China	0 sq.m	Moderate Impact
12	Madagascar Rainforest	800 sq.m	Low Impact
13	Delhi, India	600 sq.m	Moderate Impact
14	Kruger National Park, South Africa	10 sq.m	Low Impact
15	Bay of Bengal	100 sq.m	Low Impact
16	Congo Rainforest	1200 sq.m	Low Impact
17	Paris, France	0 sq.m	Moderate Impact
18	Atlantic Ocean	300 sq.m	Low Impact
19	Yellowstone National Park, USA	15 sq.m	Moderate Impact
20	Ganges River, India	0 sq.m	Moderate Impact
21	Sumatra, Indonesia	750 sq.m	Low Impact
22	Caribbean Sea	600 sq.m	Low Impact
23	Botswana National Park	30 sq.m	Low Impact
24	Gulf of Mexico	0 sq.m	Low Impact
25	Costa Rica Rainforest	0 sq.m	Low Impact
26	Arctic Ocean	250 sq.m	Moderate Impact
27	Nairobi National Park, Kenya	0 sq.m	Low Impact
28	Tokyo, Japan	0 sq.m	Moderate Impact
29	Papua New Guinea Forest	1000 sq.m	Low Impact
30	North Sea	150 sq.m	Moderate Impact

Fig 7. Predicted Impact Data

User can see all the AI predictions in this section, AI will generate the bar chart and the prediction impact table as shown in the fig 6 and fig 7.

- **Participate in Discussions:** Engage with the community through discussion threads.

Environmental Monitoring and Reporting System Logout

Community Discussions

pollution
there are more pollutions happening in the jersey measures need to be taken by government
Posted on: 12/19/2024, 3:33:57 AM

Start a New Discussion
Title:

Content:

Post Thread

Fig 8. Community Discussions

In this section the user can post the threads on our website directly and the posted threads can be seen by remaining users as shown in fig 8.

- **Receive Alerts:** Access real-time environmental alerts on a map.

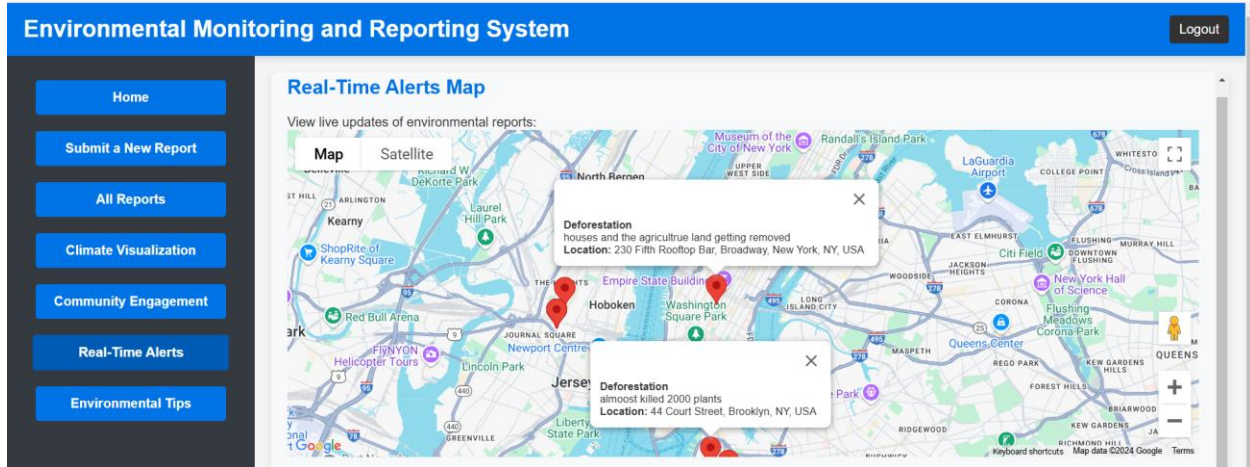


Fig 9. Real-Time Alerts Map

In this section use can be able to see the alerts in the map directly when the report is posted. when the report is submitted, alert will be shown pointing the location of the incident in the Real-Time Alerts section as shown in the fig 9.

- **Access Environmental Tips:** Learn eco-friendly practices.

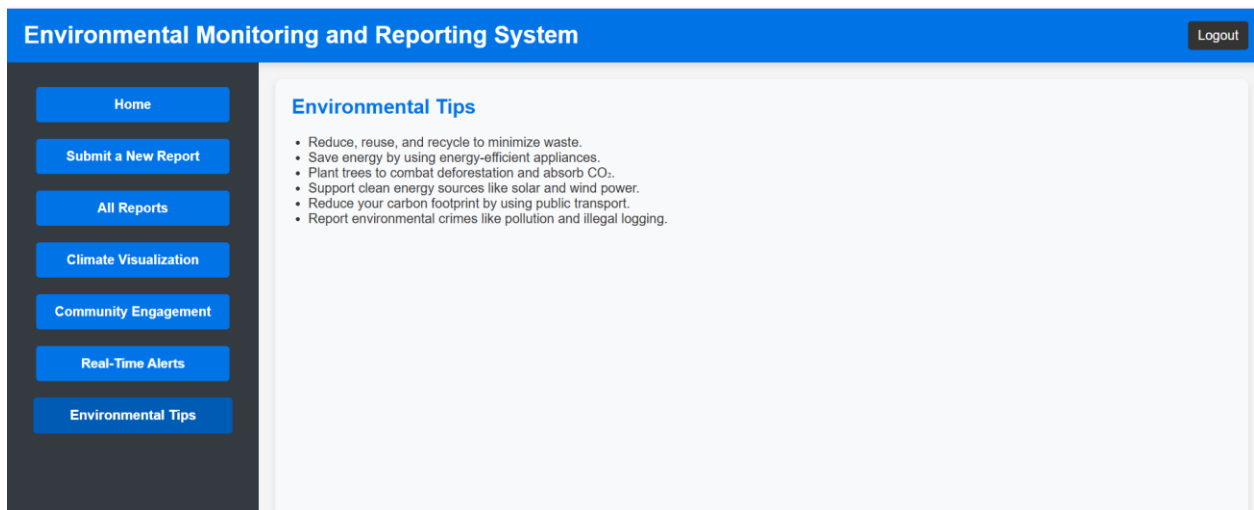


Fig 10. Environmental Tips

In this section the user can see the environmental tips posted by the administrator as shown in the fig 10.

5.2 Admin Capabilities

- **Manage Reports:** View, reject, or delete reports.
- **Monitor Discussions:** Moderate community discussions.
- **Analyze Data:** Access advanced analytics for reports and user activity.

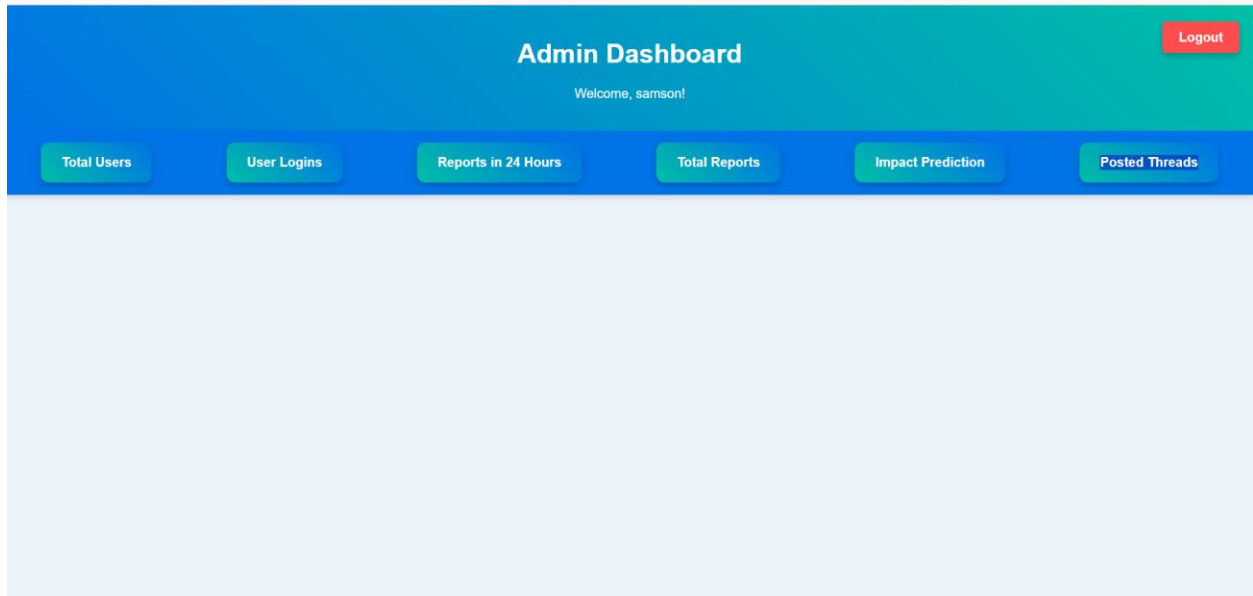


Fig 11. Admin Dashboard

Fig 11 shows the admin dashboard interface where administrator will be have access to view the users count and see the count of the user logins.admin dashboard contains more options for viewing the reports, AI predictions with the submitted reports, and can see the posted threads. As shown in the above fig 11.

6. How AI Works in the Project

6.1 AI Features

The AI module predicts environmental impact based on user-submitted data. Key functionalities include:

- **Predictive Analysis:** AI predicts the severity of climate impact based on historical data as shown in Fig 6 & Fig 7.
- **Visualization:** Displays AI-generated predictions in charts and tables as shown in Fig 14.

6.2 Steps to Run the AI Module

1. Start the Flask server in the ai/ folder using *python ai_service.py*.

```
Microsoft Windows [Version 10.0.26100.2454]
(c) Microsoft Corporation. All rights reserved.

C:\Users\prath>cd "C:\xampp\htdocs\Softwareproject\emrs"

C:\xampp\htdocs\Softwareproject\emrs>python ai_service.py
```

Fig 12. Flask server

2. The system connects with the AI server for real-time predictions.

```
C:\xampp\htdocs\Softwareproject\emrs>python ai_service.py
Starting AI Service...
* Serving Flask app 'ai_service'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:8000
Press CTRL+C to quit
* Restarting with stat
Starting AI Service...
* Debugger is active!
* Debugger PIN: 135-114-912
```

Fig 13. AI server for real-time predictions

3. Visualizations will automatically update in the "Climate Visualization" section.

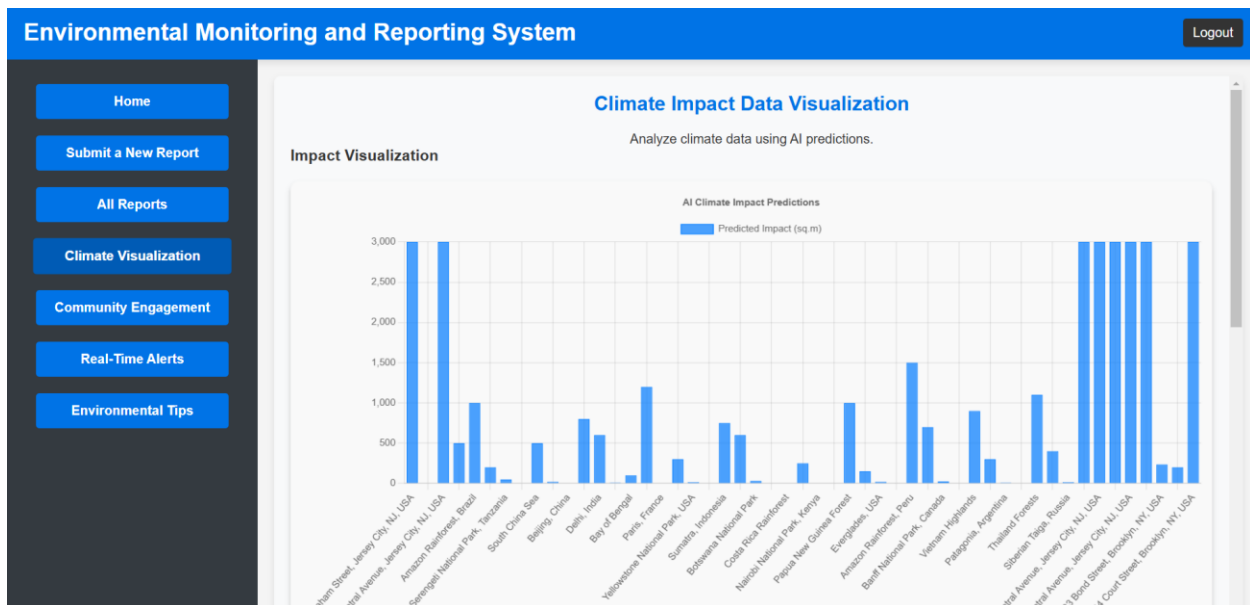


Fig 14. Climate Impact Visualization

7. Troubleshooting

7.1 Common Issues and Solutions

- **Issue:** Flask server not starting.
Solution: Ensure Python and dependencies are installed.

Run `pip install -r requirements.txt` again.

- **Issue:** Database connection error.
Solution: Verify database credentials in `Database.php`.
- **Issue:** Reports not displaying.
Solution: Check the database import and ensure data is inserted.

7.2 System Logs and Diagnostics

If there are issues, click the document in google chrome and inspect the console log by pressing the `fun + fl2` key you will as showed in the bellow fig 15.

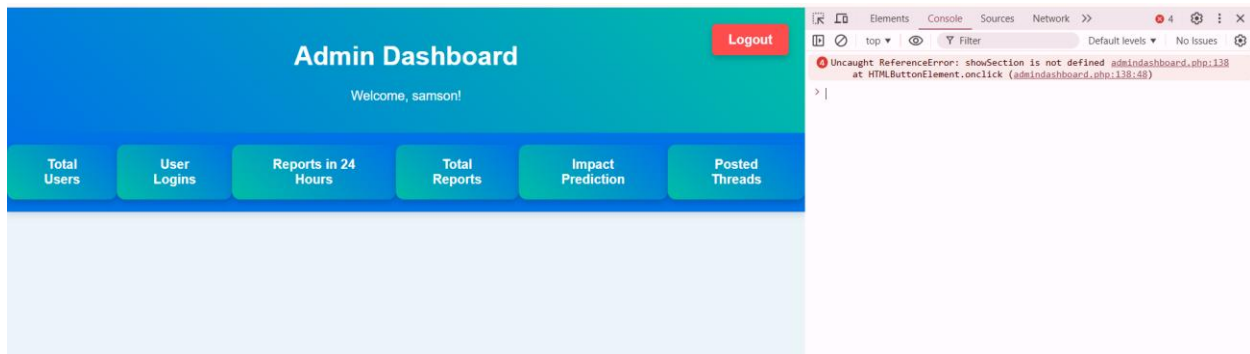


Fig 15. Console Log for Admin Dashboard