Software Requirements Document (SRD) for EcoTrace: Community-Driven Environmental Monitoring Platform

1. Introduction

1.1 Purpose

This document provides a detailed overview of the software requirements for EcoTrace, a web-based platform designed to enable communities to monitor and address environmental issues effectively. EcoTrace empowers users to report pollution, deforestation, and other environmental concerns, fostering community action towards sustainability.

1.2 Scope

The EcoTrace platform will feature capabilities for issue reporting, data visualization, community engagement, educational resources, and gamification to encourage ongoing participation. It aims to enhance community awareness and drive collaborative action for environmental conservation.

1.3 Definitions, Acronyms, and Abbreviations

GIS: Geographic Information System

API: Application Programming Interface

SRD: Software Requirements Document

2. Overall Description

2.1 Product Perspective

EcoTrace will integrate with geographic information systems and leverage existing APIs for mapping and data processing. It serves as a community portal for environmental activism, data sharing, and educational outreach.

2.2 Product Features

Issue Reporting: Enables users to report environmental issues with multimedia support and geolocation tagging.

Data Visualization: Utilizes interactive maps and dashboards to display environmental data and trends.

Community Engagement: Supports forums, updates on environmental projects, and organization of local events.

Educational Resources: Provides learning materials on environmental topics relevant to the community.

Gamification: Rewards active participants with badges and recognition on leaderboards.

2.3 User Classes and Characteristics

General Public: Individuals interested in reporting and learning about local environmental issues.

Environmental Activists: Users engaged in organizing and participating in conservation efforts.

Educators and Students: Users accessing the platform for educational purposes.

2.4 Operating Environment

EcoTrace will be hosted on cloud services such as AWS or Azure to ensure scalability and reliability. It will be accessible via web browsers on both desktop and mobile devices.

2.5 Design and Implementation Constraints

Compliance with data privacy laws and regulations regarding user data.

Integration with third-party services must ensure reliability and timely data updates.

3. System Features

3.1 Issue Reporting

Description: Allows users to report environmental issues with photos, videos, and descriptions.

Functional Requirements:

FR1: The system shall enable users to upload multimedia files.

FR2: The system shall automatically capture and tag the geographical location of the reported issue.

3.2 Data Visualization

Description: Provides interactive maps and dashboards to visualize and monitor environmental data.

Functional Requirements:

FR3: The system shall integrate GIS functionalities to display reported issues on a map.

FR4: The system shall offer filters to view data based on different criteria (e.g., date, type of issue).

3.3 Community Engagement

Description: Facilitates community discussions, updates, and coordination of local environmental actions.

Functional Requirements:

FR5: The system shall provide a forum for users to discuss environmental issues.

FR6: The system shall allow users to organize and promote local cleanup events.

4. External Interface Requirements

4.1 User Interfaces

Responsive web design compatible with multiple browsers and devices.

4.2 Hardware Interfaces

No specific hardware interfaces required.

4.3 Software Interfaces

Backend developed using Node.js with Express.

Database management through MongoDB for storing environmental reports and user data.

4.4 Communications Interfaces

RESTful API interfaces for backend communication.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The platform should handle up to 10,000 active users simultaneously.

5.2 Security Requirements

Implement standard security measures including data encryption and secure API interactions.

5.3 Quality Attributes

Usability: Intuitive user interface designed for ease of use by all age groups.

Reliability: High availability and error handling capabilities to ensure continuous service.

6. Development Timeline

Days 1-3: Infrastructure setup, including server and database configuration. Begin frontend development for registration and issue reporting.

Days 4-6: Development of GIS integration and community engagement features.

Days 7-9: Implementation of gamification elements and educational resources.

Days 10-12: System integration and initial testing with community feedback.

Days 13-14: Final testing, UI polishing, and preparation of demonstration materials.