# EcoTracker

**Sahil Kumar Valecha**

ID: 2312172

**Saleena Ahuja**

ID: 2312174

# ACKNOWLEDGMENT

We are really grateful because we managed to complete our project within the time given by our teacher Abid Ali. This assignment cannot be completed without the effort and co-operation from our group members, Group members Sahil Kumar Valecha And Saleena Ahuja

We also sincerely thank our teacher Abid Ali for the guidance and encouragement in finishing this project and also for teaching us in this course.

Last but not the least, we would like to express our gratitude to our friends and resondents for the support and willingness directly or indirectly to spend some times with us to fill in the questionnaires.

# CONTENT

# Introduction

This EcoTracker is a community-driven environmental incident reporting platform designed to promote ecological awareness and civic responsibility. Built using a Node.js and Express.js backend connected to a PostgreSQL (Neon DB) database, the system allows users to report issues such as pollution, deforestation, and illegal dumping in their neighborhoods. The reports can include optional images, categories, severity levels, and anonymity preferences. A dynamic frontend interface enables users to browse EcoAlerts, vote on reports, and comment, while admins get role-based access and control. The real-time system ensures data freshness, secure session handling, and responsive performance, enhancing user engagement and accountability.

# WORK ANALYSIS

| Task | Sahill Kumar | Saleena Ahuja |
|---|---|---|
| Analysis | ✓ | ✓ |
| Design | ✓ | ✓ |
| Coding | ✓ | ✓ |
| Testing | ✓ | ✓ |
| Documentation | ✓ | ✓ |

# Technologies Used

- **Frontend:** HTML5, CSS, Tailwind CSS, JavaScript
- **Backend:** Node.js with Express.js
- **Database:** PostgreSql
- **Deployment:** Localhost

# EcoTracker Incident Reporting Module

This module forms the core of the application, enabling users to report and view environmental incidents with contextual details.

- **Reports & Users:**
  - Tables: reports, users
  - Stores report data (title, description, severity, anonymity) and user details (name, email, role).
- **Categories & Locations:**
  - Tables: categories, locations
  - Classifies incidents (e.g., air pollution, water waste) and logs exact areas (latitude, longitude, neighborhood).
  - Enables area-based filtering and category tagging.
- **Image Uploads:**
  - Column: image_url in reports
  - Stores optional image evidence of incidents via Multer, hosted for frontend display.

---

## Voting & Feedback Module

Facilitates community interaction through engagement and user feedback.

- **Voting System:**
  - Table: report_votes
  - Users can upvote or downvote reports once.
- **Comments Section:**
  - Table: comments
  - Enables threaded discussions under each report with user ID mapping.
- **Vote Aggregation & Status Tracking:**
  - SQL joins calculate upvotes, downvotes, and retrieve current report status for analytics and display.

---

## Authentication & Access Control Module

Handles secure session management and user role differentiation.

- **Session Handling:**
  - Libraries: express-session, connect-pg-simple
  - Implements secure login sessions with session regeneration to prevent fixation attacks.
- **Role-Based Access:**
  - Admin detection via email domain (@ecotracker.pk)
  - Custom session payloads manage user roles, accessible via middleware (requireLogin, checkAuth).

## Map & Filtering Module

Visualizes reported incidents on a dynamic map with filters.

- **Map Integration:**
  - Data source: locations and reports tables
  - Frontend fetches location-coordinates to plot report pins.
- **Filter API:**
  - Route: /api/incidents
  - Query params for latitude, longitude, and category; returns scoped results for area-based views.

## Admin Panel Module

Allows privileged users to moderate reports and manage data.

- **Admin Verification:**
  - Checks logged-in user's email domain before granting access.
- **Moderation Features (Planned):**
  - Includes toggling report status, reviewing flagged reports, and potentially adding verified badges.
  -

# Salient Features of the EcoTracker Platform

- **Location-Based Incident Mapping:**
  - Allows dynamic plotting of reports on a map using latitude and longitude coordinates.
  - Filters reports by category and proximity, enhancing local relevance and visibility.
- **Anonymous Reporting Option:**
  - Users can choose to submit incidents without revealing their identity.
- **Vote-Driven Feedback System:**
  - Reports can be upvoted or downvoted to reflect public consensus on urgency or validity.
  - Conflict-handled vote submissions ensure each user can only vote once per report.
- **Role-Based Access Control (RBAC):**
  - Admins (identified via @ecotracker.pk domain) have elevated privileges.
  - Ensures protected access to moderation tools and sensitive data views.
- **RESTful API Design:**
  - Backend architecture built using Express.js and PostgreSQL with clean, modular endpoints.

  - Supports JSON responses for frontend fetch requests and third-party integrations.
- **Secure Session Management:**
  - Sessions regenerated on login to prevent fixation; securely stored via connect-pg-simple.
- **Scalable & Normalized Database Schema:**
  - Tables follow normalization principles with primary and foreign keys (e.g., category_id, location_id).
- **Frontend Integration with Dashboard:**
  - Admin or user dashboards display real-time stats such as:
    - Total Reports
    - Critical Incidents
    - Anonymous vs. Verified Submissions
- **Media Support for Reports:**
  - Frontend displays these images in a responsive card layout.
- **Modular & Extensible Architecture:**
  - Designed to scale with future features like real-time notifications, commenting, analytics, or mobile app support.
  - Minimal schema and route changes needed for upgrades

# Dashboard & Reporting

Built to provide a clear overview of environmental incident trends and platform usage.
Interactive features empower users and admins to:

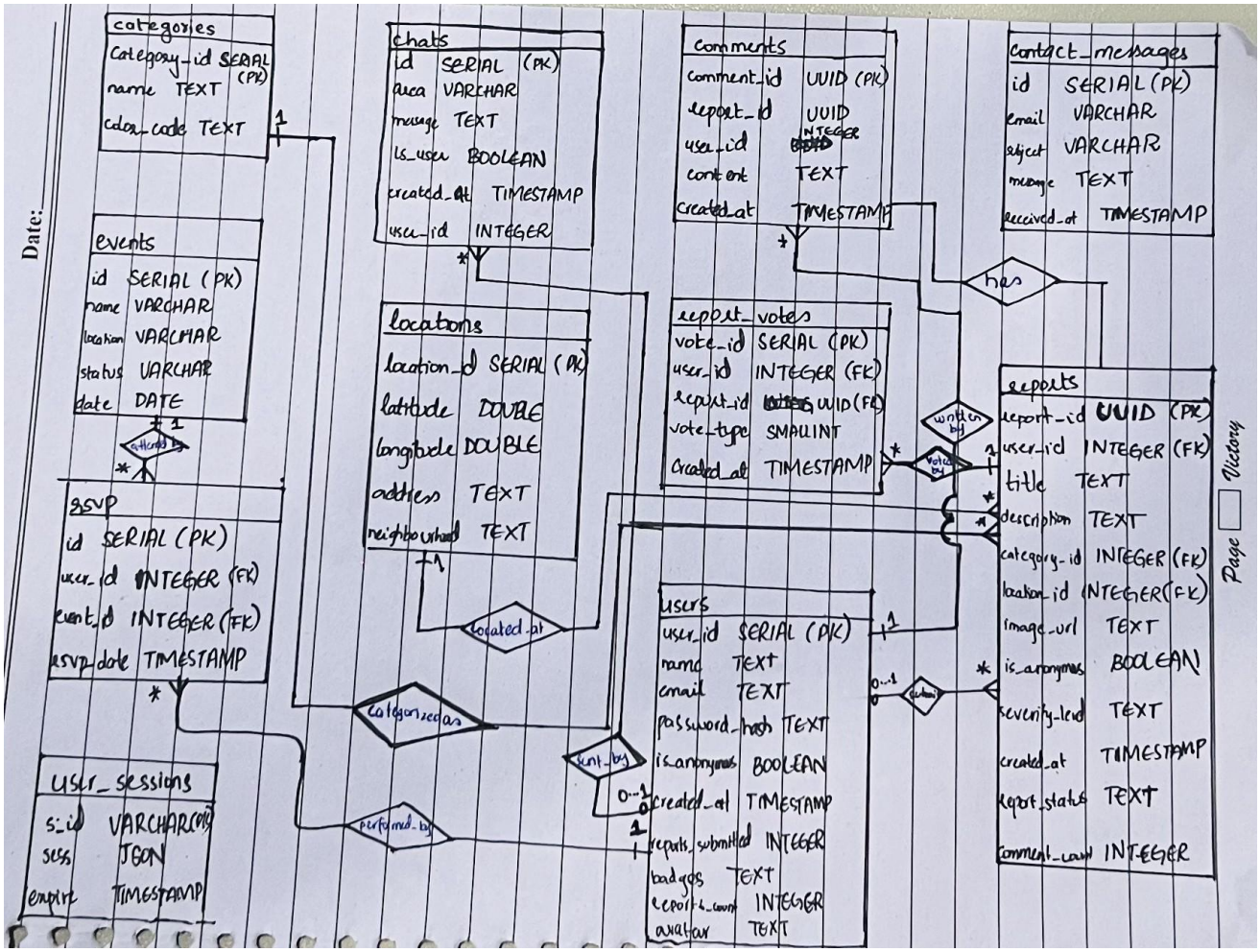
- **View live counts** of total reports, categories, anonymous submissions, and upvotes/downvotes..
- **Filter incidents by location, category, or severity level** in real time.
- Designed to integrate with **Chart.js** for future inclusion of:
  - Area-specific incident trends
  - Category distribution via bar charts

# Security & Compliance

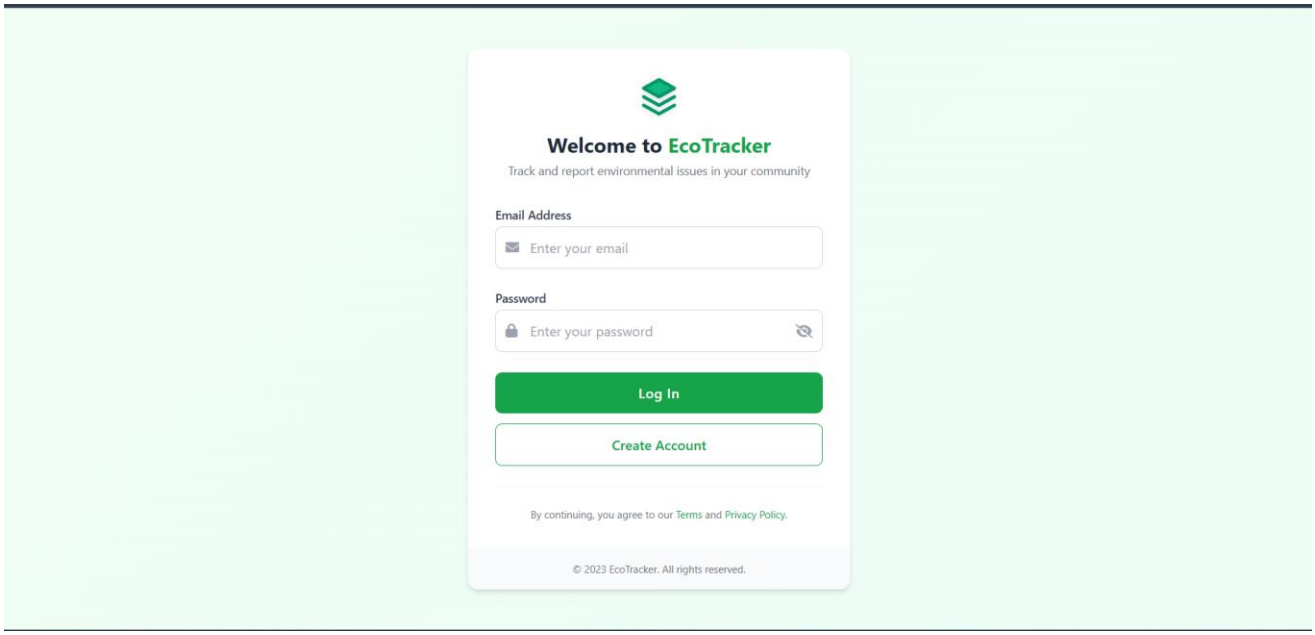
EcoTracker is designed with data security and integrity as core principles.

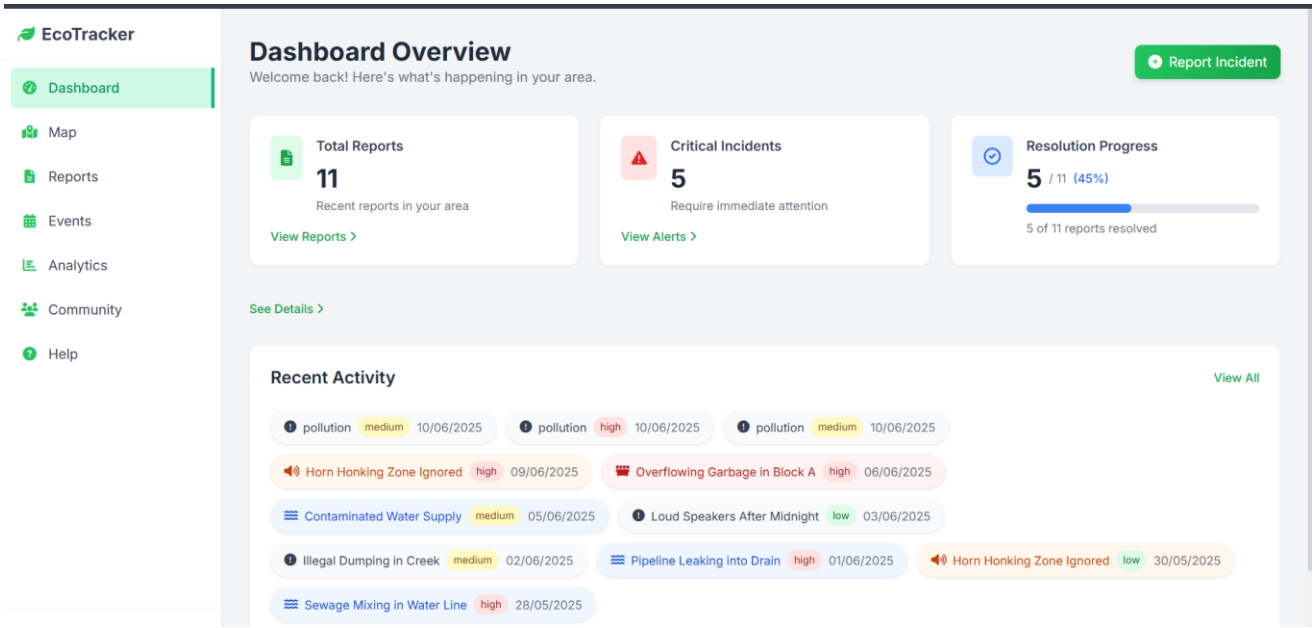
- **Parameterized SQL queries** across all endpoints prevent SQL injection vulnerabilities.
- **Unique constraints** on user sessions, report IDs, and email addresses maintain database consistency.
- **Session-based access control** ensures protected resources are only accessible by logged-in users.
- **Anonymous reporting option** hides user identity while still linking actions to valid sessions.
- **Audit-friendly structure** allows administrators to trace reports, votes, and user interactions for compliance and moderation purposes.

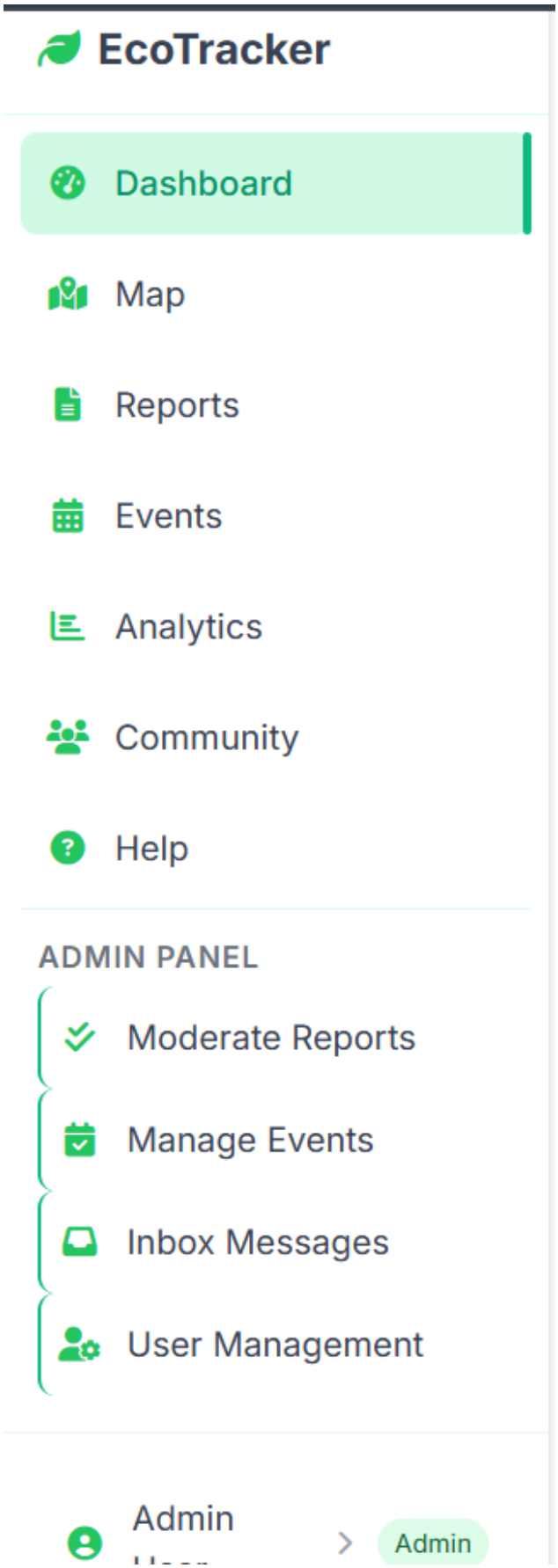
# ERD:


categories
Category_id SERIAL (PK)
name TEXT
color_code TEXT
chats
id SERIAL (PK)
area VARCHAR
message TEXT
is_user BOOLEAN
created_at TIMESTAMP
user_id INTEGER
comments
comment_id UUID (PK)
report_id UUID
user_id INTEGER
content TEXT
created_at TIMESTAMP
contact_messages
id SERIAL (PK)
email VARCHAR
subject VARCHAR
message TEXT
received_at TIMESTAMP
events
id SERIAL (PK)
name VARCHAR
location VARCHAR
status VARCHAR
date DATE
locations
location_id SERIAL (PK)
latitude DOUBLE
longitude DOUBLE
address TEXT
neighbourhood TEXT
report_votes
vote_id SERIAL (PK)
user_id INTEGER (FK)
report_id UUID (FK)
vote_type SMALLINT
created_at TIMESTAMP
has
reports
report_id UUID (PK)
user_id INTEGER (FK)
title TEXT
description TEXT
category_id INTEGER (FK)
location_id INTEGER (FK)
image_url TEXT
is_anonymous BOOLEAN
severity_level TEXT
created_at TIMESTAMP
report_status TEXT
comment_count INTEGER
rsvp
id SERIAL (PK)
user_id INTEGER (FK)
event_id INTEGER (FK)
rsvp_date TIMESTAMP
users
user_id SERIAL (PK)
name TEXT
email TEXT
password_hash TEXT
is_anonymous BOOLEAN
created_at TIMESTAMP
reports_submitted INTEGER
badges TEXT
report_count INTEGER
avatar TEXT
user_sessions
s_id VARCHAR(PK)
sess JSON
expire TIMESTAMP
located_at
categorized_as
sent_by
performed_by
written by
voted by
submits
Date:
Victory Page

# User Interface




EcoTracker
Dashboard
Map
Reports
Events
Analytics
Community
Help
ADMIN PANEL
Moderate Reports
Manage Events
Inbox Messages
User Management
Admin
Admin

# Conclusion

EcoTracker presents a practical and socially impactful solution for reporting and visualizing environmental incidents in real time. By combining a responsive frontend, a secure backend, and a normalized PostgreSQL database, the platform ensures seamless interaction between users, data, and insights. Features like anonymous reporting, interactive maps, and vote-based prioritization encourage community involvement and transparency.

The project not only addresses the growing need for environmental accountability but also lays a strong foundation for future enhancements such as analytics dashboards, multilingual support, and mobile accessibility. With its modular design and RESTful architecture, EcoTracker stands ready to scale and adapt to the evolving challenges of civic and environmental engagement.

# THANK YOU