SEGNALAZIONE GEOLOCALIZZATA DI EMERGENZE: SAFECITY

Mattia Piazza & Giammaria Mascandola

SafeCity nasce per creare uno strumento digitale che migliori la sicurezza urbana, permettendo agli utenti di segnalare in tempo reale eventi emergenziali e disastri naturali sul territorio.

Il progetto sfrutta diverse tecnologie:

* React.js : framework JavaScript per la creazione dell’interfaccia utente dinamica e interattiva;
* Leaflet.js : libreria specializzata per la visualizzazione e l’interazione con mappe geografiche;
* Node.js : backend RESTful per la gestione della logica, delle API e dell’autenticazione;
* PostgreSQL + PostGIS : database relazionale con estensione spaziale per la gestione efficiente di dati geografici (coordinate, query spaziali, triangolazione eventi);
* Axios : comunicazione client-server;
* OpenWeatherMap API : integrazione dati meteo in tempo reale ;

All’apertura della wep app, l’utente visualizza una pagina di autenticazione: qui sceglie se effettuare il login (se già registrato) o effettuare la registrazione come nuovo utente, specificando il suo ruolo (utente normale/moderatore).

Dopo l’accesso l’utente viene reindirizzato alla dashboard principale, la quale è divisa in due aree:

* Una sidebar che consente di accedere velocemente a funzioni come la segnalazione di emergenze nella posizione corrente, la verifica delle condizioni meteo, il filtro delle segnalazioni per tipologia e il logout.
* Un’area centrale, dominata dalla mappa (Leaflet): qui vengono visualizzate tutte le emergenze segnalate dagli utenti, con icone e popup differenziati per tipologia di evento.

Ogni marker sulla mappa corrisponde a una segnalazione, inserita tramite apposito form dopo aver selezionato la posizione geografica (tramite click sulla mappa o tramite posizione attuale).  
Per ogni emergenza vengono raccolti:

* Tipo d’evento
* Descrizione
* Data, ora
* Coordinate GPS
* Stato segnalazione

Gli utenti possono filtrare le segnalazioni per tipologia, verificare la presenza di eventi nella loro zona e contribuire a validare gli eventi lasciando commenti (anche con foto). Ogni emergenza può raccogliere testimonianze, conferme o smentite da parte di più utenti, contribuendo a rendere l’applicazione web una rete collaborativa di segnalazioni. L’interfaccia per i moderatori permette anche di eliminare segnalazioni inappropriate, garantendo così una maggiore qualità e attendibilità delle informazioni condivise.

In aggiunta alle funzionalità di base ci sono anche funzionalità aggiuntive:

* l’utente può verificare il meteo attuale di una determinata area tramite un modulo dedicato, integrato con la mappa.
* i commenti possono essere corredati da foto multiple, caricate e servite tramite backend.

Inoltre, il sistema è pensato per gestire in futuro notifiche automatiche per utenti nella zona di rischio e per la triangolazione automatica di eventi simili vicini.

SafeCity si divide in due aree strutturali:

Il Frontend (React) organizza componenti e pagine come AuthPage, Dashboard, Map, ReportForm, FilterForm, CommentsForm, WeatherForm. Inoltre, l’integrazione con Leaflet.js consente la gestione di mappe vettoriali, marker dinamici, filtri geografici e selezione puntuale di coordinate..

Il Backend (Node.js/Express) gestisce API RESTful, autenticazione, salvataggio e validazione delle segnalazioni e dei commenti, upload e gestione sicura delle immagini. Sfrutta Node.js con Express.js, adottando un’architettura modulare suddivisa in route, controller e servizi. Tutte le operazioni critiche (registrazione, autenticazione, creazione/emergenza, gestione immagini) sono protette da middleware per la validazione dei dati e la gestione degli errori. L’upload sicuro dei file è gestito con Multer, che salva le immagini in percorsi statici accessibili solo tramite le rotte apposite.

Il Database (PostgreSQL+PostGIS) è strutturato con tabelle relazionate (1 a m) tra loro quali:

1. users: informazioni sugli utenti registrati e i loro ruoli
2. emergency\_types: tipologie di emergenza, con label e icona dedicata
3. emergencies: segnalazioni geolocalizzate, collegate all’utente e alla tipologia di evento
4. comments: commenti e conferme sulle emergenze, foto allegate

il progetto sfrutta PostgreSQL come database relazionale, con estensione PostGIS per la gestione di dati geografici (campi GEOGRAPHY e supporto a query spaziali).  
L’uso dell’ORM Sequelize offre vantaggi in termini di portabilità, sicurezza (evita SQL injection) e migrazioni del database.  
L’uso di PostGIS permette di realizzare query geospaziali efficienti, fondamentali per filtri, calcolo delle distanze e futura implementazione della triangolazione automatica degli eventi.

Le password degli utenti sono hashate e il sistema di autenticazione si basa su token JWT memorizzati nel localStorage del browser, garantendo così un accesso sicuro alle risorse protette e la gestione della sessione.

La flessibilità e modularità della piattaforma permettono di estendere facilmente le funzionalità in futuro con statistiche avanzate come previsioni sugli eventi ricorrenti ed esportazione dati per enti pubblici (anche con implementazione di sistemi d’IA).