Documentation

SpotnSend

Stay Aware. Stay Safe.

Team Cipher Nova

Table of Contents

Introduction

Every day in Egypt, we come across situations that could turn dangerous in seconds. A sudden car accident on the ring road, a fire in a crowded building, a pothole that almost flips a microbus, or a broken streetlight that makes a road unsafe at night. Most of the time, people only hear about these risks after they happen when it's already too late.

This is where SpotnSend comes in. SpotnSend is a mobile application that connects citizens and emergency services on a single map. It gives people the ability to report incidents or complaints in seconds and instantly alert those around them, as well as the authorities.

Why does this matter? Because every minute counts. Whether it's avoiding a blocked road, reaching medical help faster, or preventing an accident before it happens, early alerts save time and save lives.

How does it work? It's simple: a user reports an incident or complaint with photos or videos. They can choose the notification radius to control how far the alert spreads, and decide who gets notified nearby people, the government, or both. Users can also save their important locations, like home or work, and receive automatic alerts if something happens there.

Who is it for? SpotnSend is built for ordinary citizens who want to stay safe, and for officials like ambulance, traffic police, or fire departments who need real-time data to act quickly.

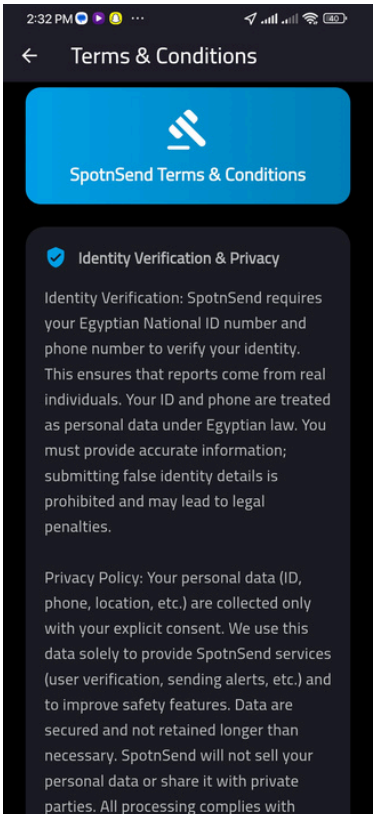
And what's in it for the user? Peace of mind. SpotnSend makes sure you know about dangers around you before you stumble into them. You can protect yourself and your loved ones, help others by reporting, and trust that the right authorities are notified instantly.

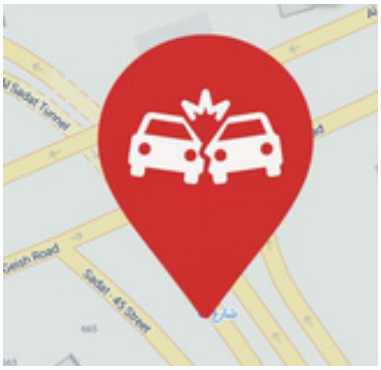
SpotnSend isn't just an app, it's a way to keep communities aware, connected, and safe.

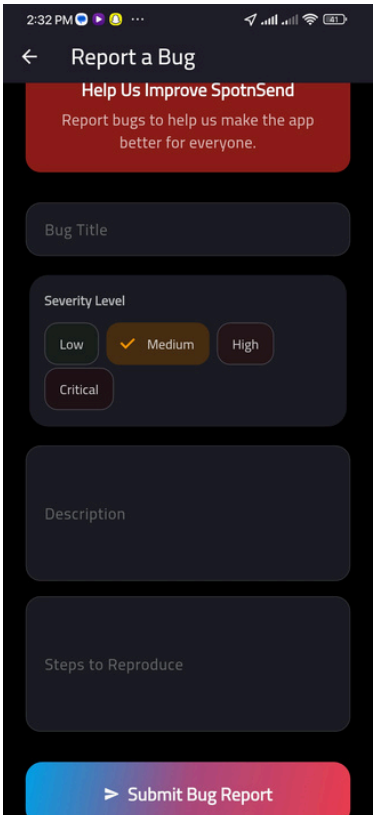
UI / UX

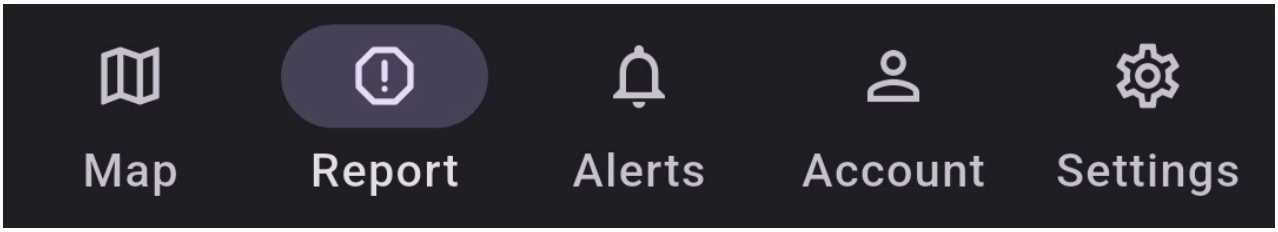
The app & dashboard were designed to be simple and easy to use. Every screen is clean, with enough space so users don't feel lost. Buttons and inputs are large and clear, so they can be used quickly in urgent situations.

- Simple and easy-to-use design for urgent situations.
- Large buttons and clear inputs for quick reporting.
- Blue used for main actions/navigation (calm, readable).
- Red reserved only for alerts & danger zones (instant attention).









Admin Dashboard

The SpotnSend Admin Dashboard is designed to give administrators quick, clear, and efficient access to all platform data. The main pages Login ,Dashboard, Incidents, Complaints, Users, and Reports are always visible as buttons in the top bar, making navigation straightforward without unnecessary clicks. Switching between sections is instant since routing is handled inside React, and there's no page reload.

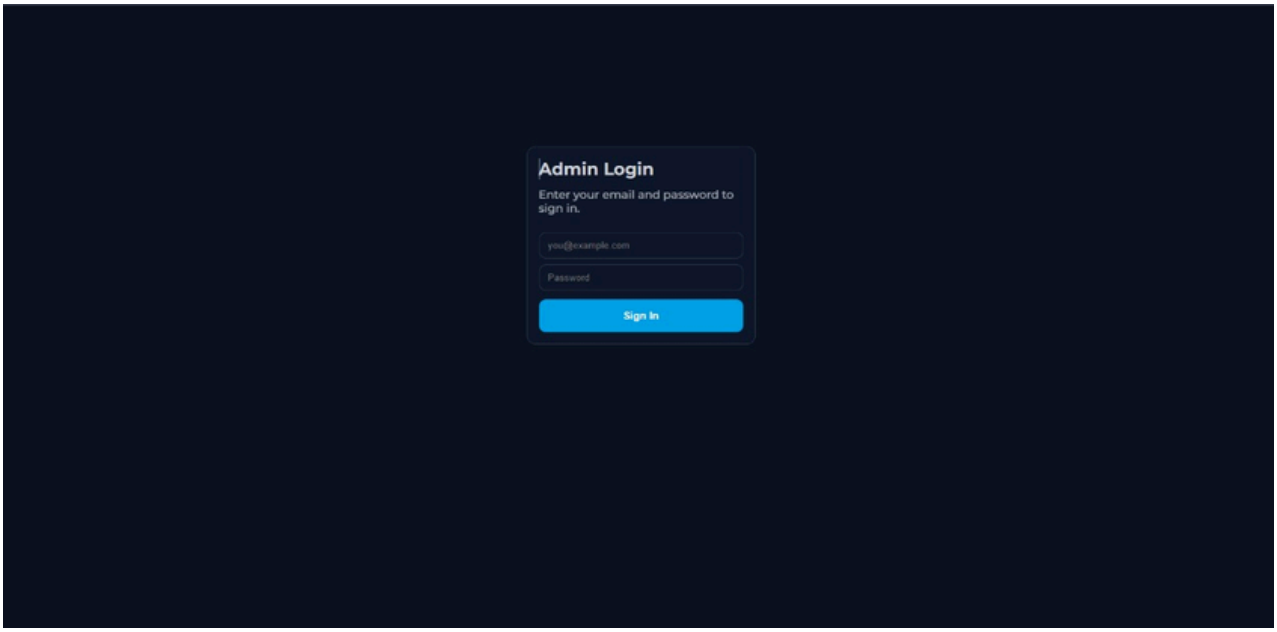
The dashboard supports light and dark modes and can switch between English and Arabic without flipping the entire layout, ensuring consistency and comfort for different users.

Login Page :

Before accessing the dashboard, admins go through a secure login page.

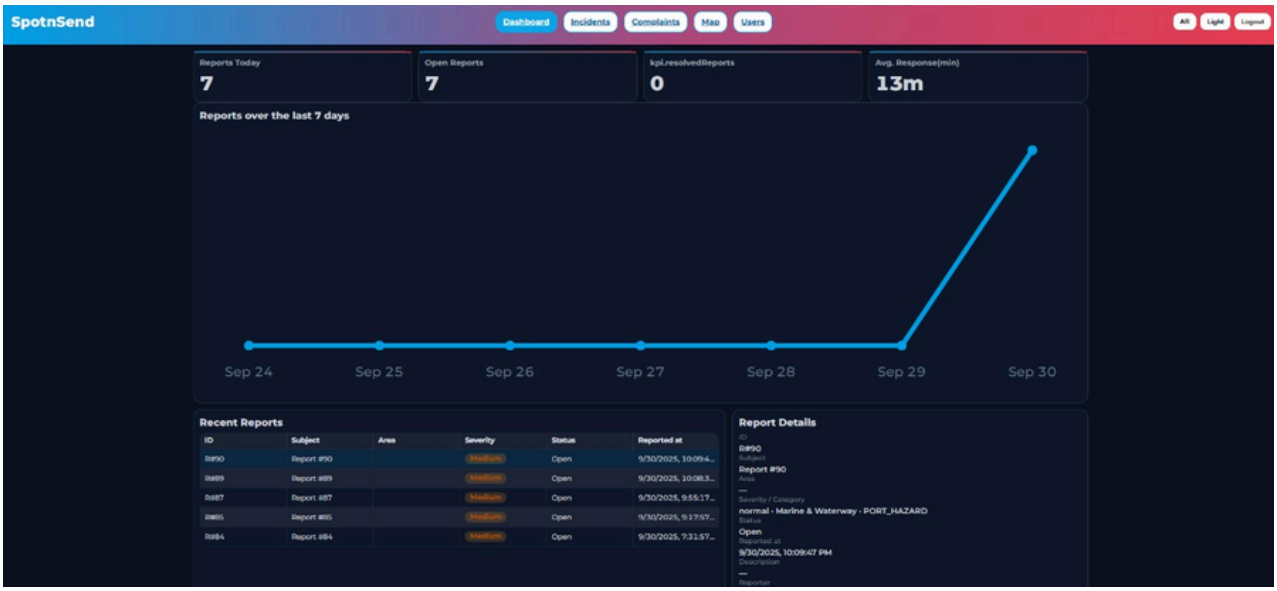
- Simple design with the SpotnSend logo and brand colors (blue/red).
- Fields: Email / Username and Password.
- Validation ensures credentials are correct before login.
- Error messages appear clearly if login fails.
- Once authenticated, the user is redirected instantly to the main Dashboard page (no reloads).

Future extensions could include multi-factor authentication (MFA) for extra security.



Dashboard Page :

Shows a summary of key statistics: total incidents, complaints, average response time, and active users. Interactive charts visualize trends at a glance, helping admins quickly understand the bigger picture.



Incidents & Complaints Pages :

Reports are organized into two clear categories :

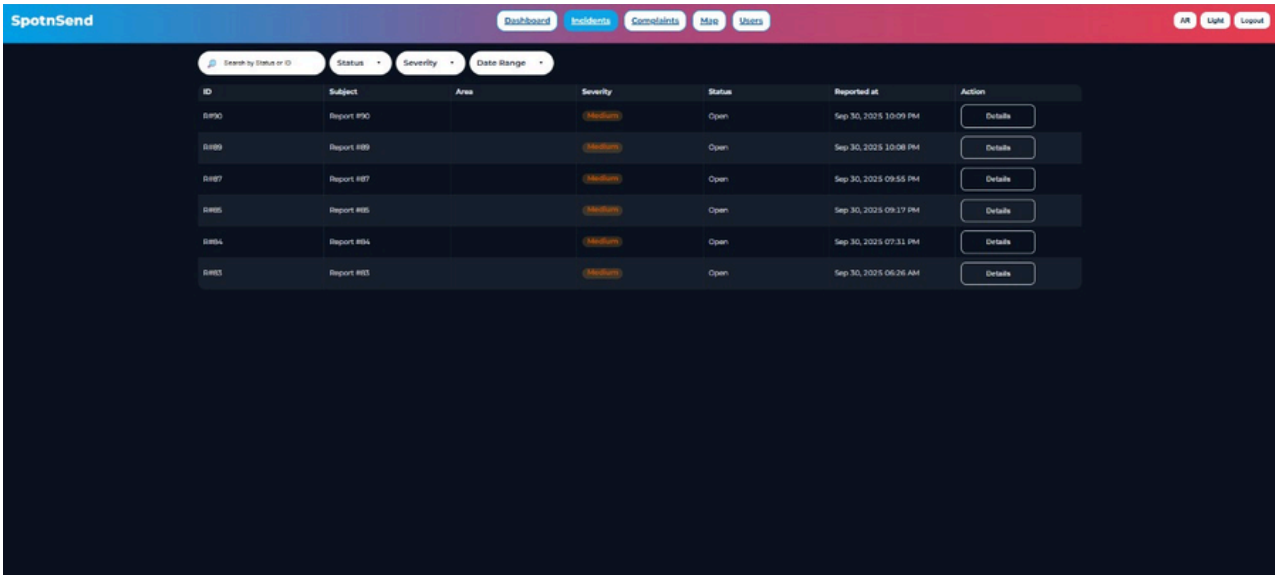
Incidents :

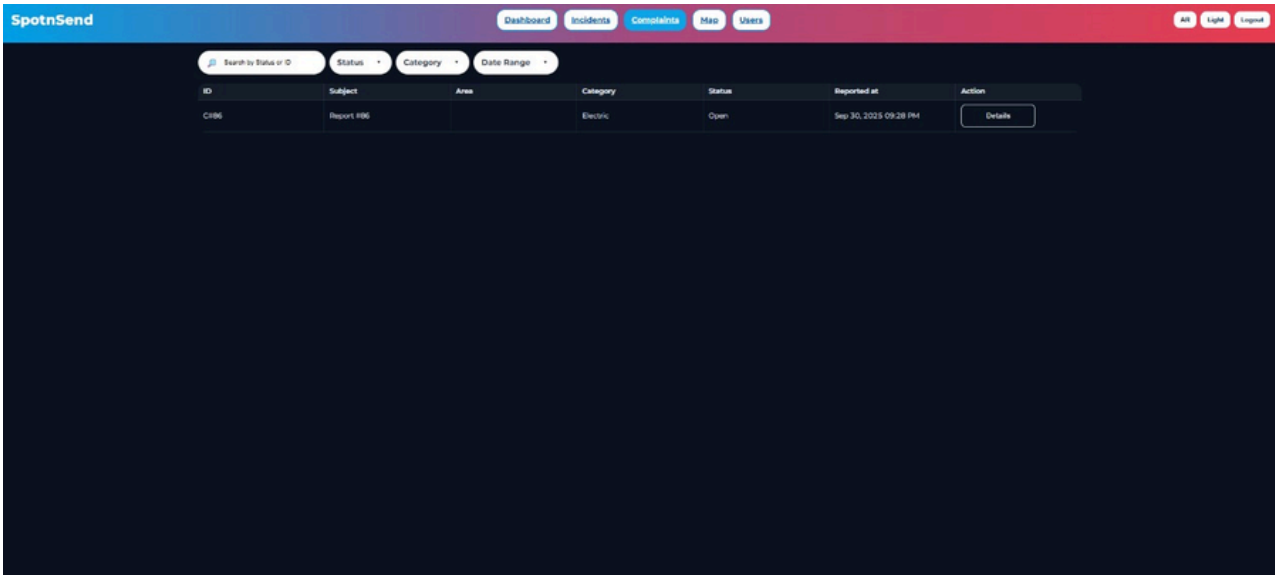
- Urgent events that require immediate response, such as :
  - Road traffic accidents (collisions, pedestrian struck, pileups, hazards like sinkholes or oil spills).
  - Fires and explosions (building fire, vehicle fire, gas leaks, cylinder blasts).
  - Building and infrastructure issues (collapse, falling façade, elevator failures).
  - Rail and public transport accidents (train, metro, bus).
  - Dangerous utility failures (major power outages, gas emergencies).
  - Severe weather (flash floods, coastal surge, sandstorms that block visibility).
  - Medical emergencies (cardiac, respiratory, mass casualty).
  - Industrial accidents (factory, chemical spills).
  - Public safety & crime (violence nearby, suspicious package, robbery in progress).
  - Marine/waterway risks (drowning, boat incidents).

Complaints :

- Issues that municipalities/utilities can handle, but don't need emergency responders:
  - Road issues (potholes, debris, streetlight outages).
  - Utility issues (local power cuts, water supply interruptions, telecom outages).

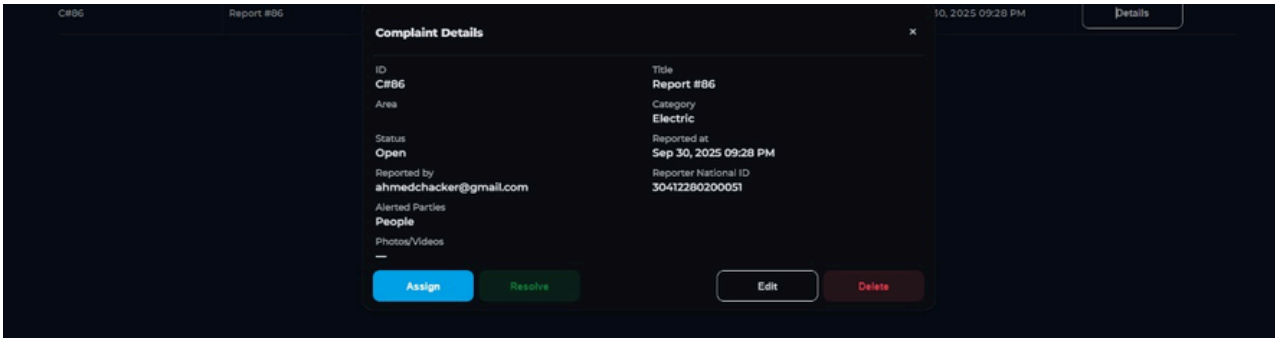
Admin Dashboard





Both Incidents and Complaints are shown in tables where admins can :

- Filter, edit, or delete records.
- See attached media (photos/videos).
- Change report status (open, assigned, resolved).
- Get confirmation prompts before deletion to avoid mistakes.



Users Page :

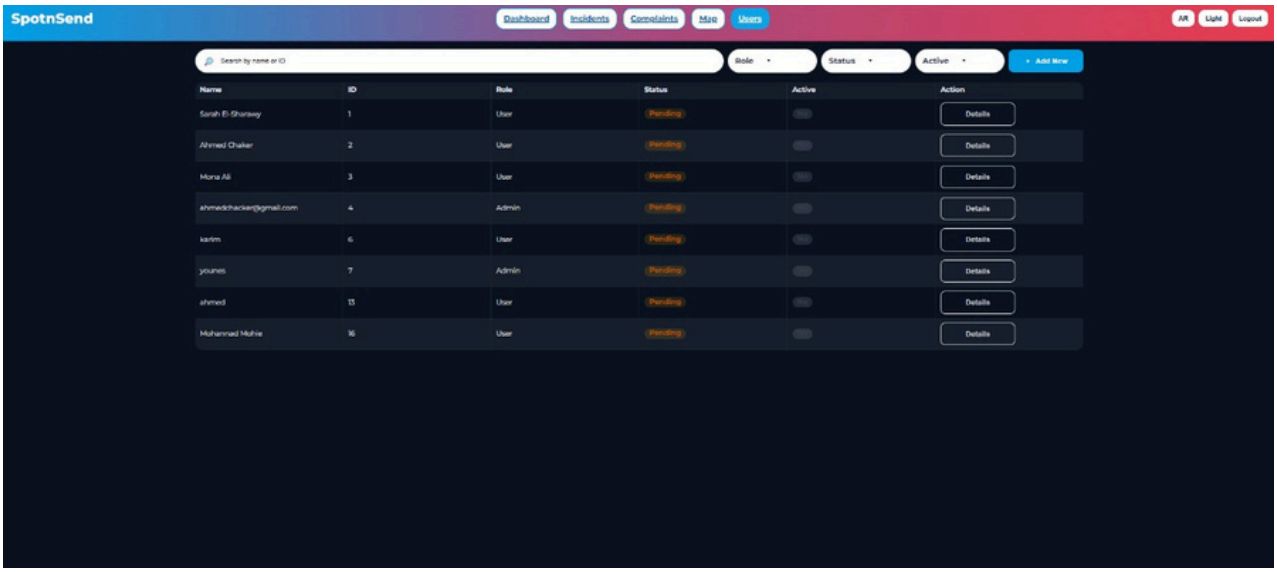
The Users section gives admins full control over registered accounts. Each user is shown in a structured row with details and quick actions :

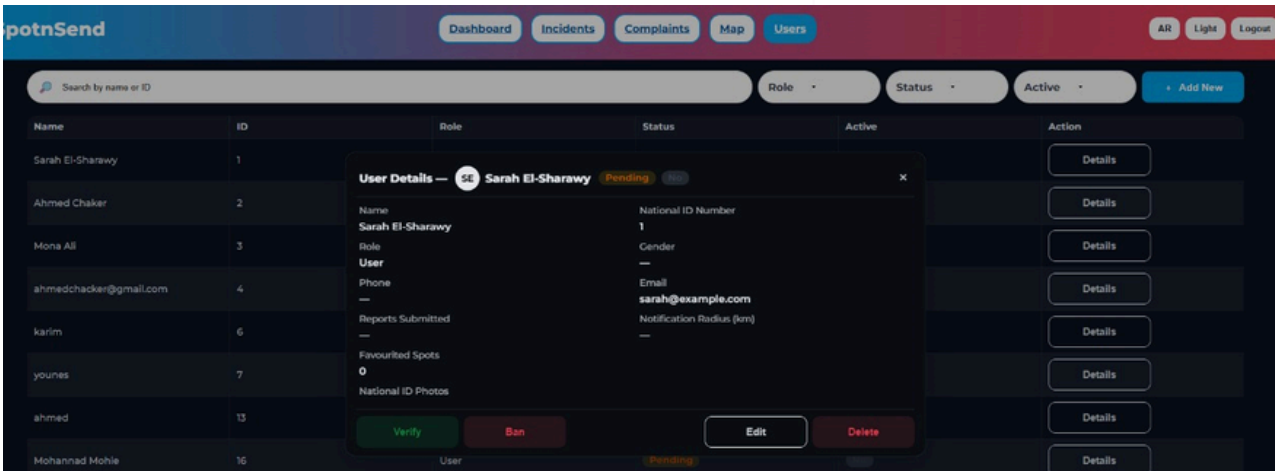
- Add a new user
- Verify, unverify, or leave pending
- Ban a user
- Delete a user (with confirmation popup)

Clicking on a user opens a detailed modal, showing :

- Full name, national ID, phone number, and gender
- Current role (admin/user)
- Number of reports submitted
- Number of saved spots
- Uploaded ID images

This ensures admins have all the information they need in one place.





Consistent Design & Clarity

- Unified styling across tables, popups, and buttons.
- Blue for standard actions, red for urgent or blocked states.
- Focus on clarity, speed, and error-proof workflows.

Technical Implementation

Stack & Backend

- Built with React + TypeScript + Vite for the frontend.
- Backend handled by Node.js + Supabase (Postgres + APIs).

Data Handling

- Incidents and complaints retrieved with utility functions (readIncidents, readComplaints).
- Updates managed with sendIncidentsUpdate, sendComplaintsUpdate.
- All destructive actions require confirmation.

User Management

- Actions (add, verify, ban, delete) are modular, kept separate from UI code for easy maintenance.

Extra Features

- Dark/Light mode toggle.
- English/Arabic support without flipping the whole layout.

Conclusion

The SpotnSend Admin Dashboard combines clean design with a robust technical foundation.

Admins can :

- Monitor incidents and complaints in real time.
- Manage users with full flexibility.
- Switch languages and themes instantly.
- Rely on a consistent, error-proof interface.

This makes the dashboard fast, reliable, and scalable , aligned with SpotnSend's vision as a trusted emergency and safety platform

The Application

Application features include :

01 Overview

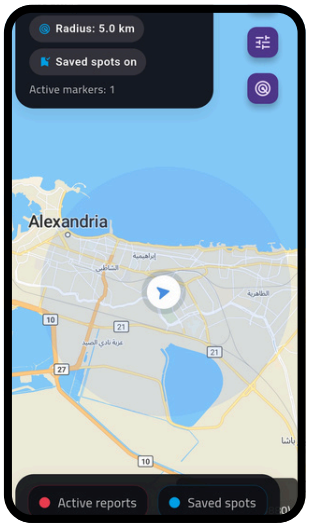
SpotnSend is a safety and reporting application that enables users to:

- View nearby incidents on a live map.
- Submit safety reports with optional media attachments.
- Receive timely alerts and notifications.

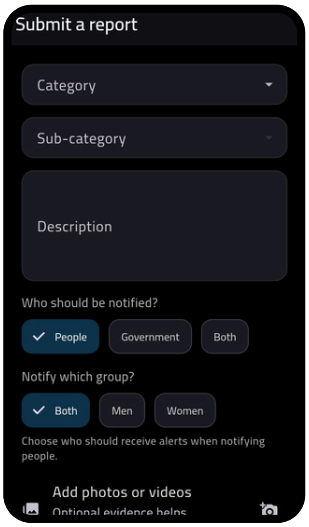
The app is built with Flutter, powered by Supabase for backend services, Riverpod for state management, and MapLibre for map rendering.

The interface is organized into five main tabs :

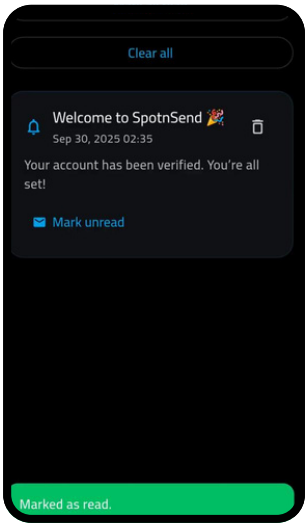
Map



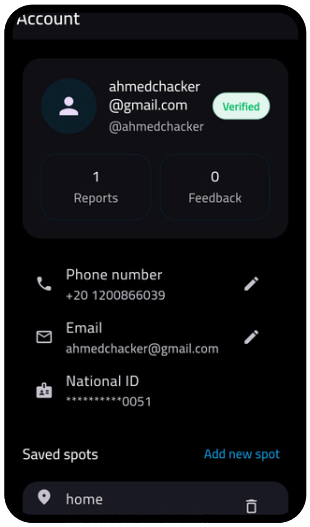
Report



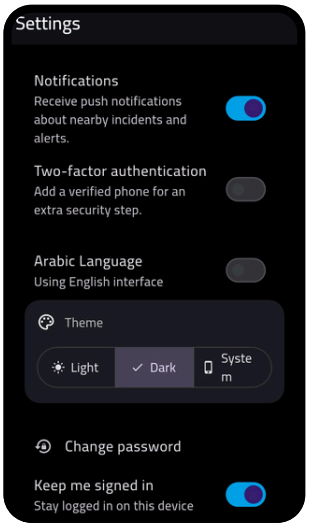
Alerts



Account



Settings



02 Configuration

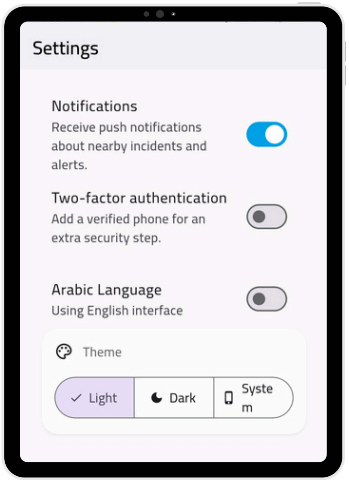
- Secrets are managed via .env.
- Keys include:
  - MAPTILER_KEY (for map rendering).
  - GEMINI_API_KEY (optional, for translations).

03 Navigation

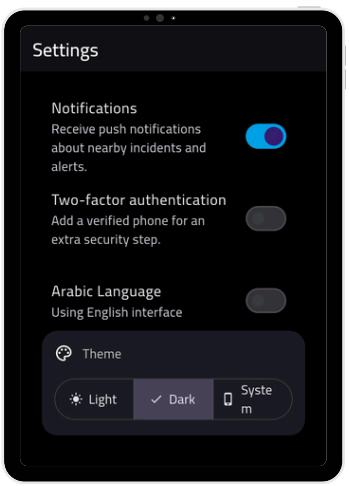
- Five-tab layout with clear separation of features.
- Unauthenticated users are redirected to login.
- Users awaiting verification cannot submit reports.
- Language and authentication state changes update navigation in real time.

04 Design System

Colors : Brand palette plus success/warning/error states.
Typography : Cairo font across the app.
Theme : Light and dark modes with consistent buttons, inputs, and navigation styling.



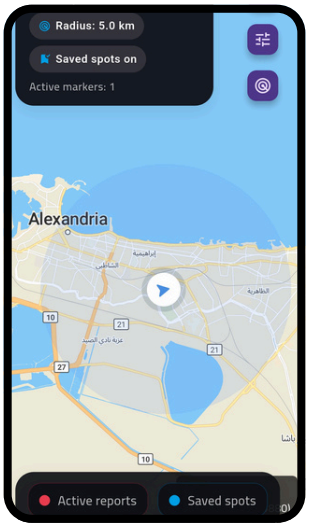
Light Theme



Dark Theme

05 Maps & Location

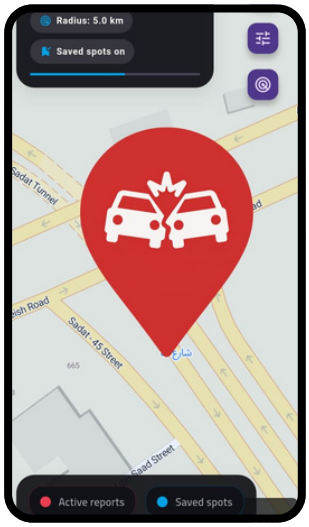
- Maps rendered with MapLibre using MapTiler styles.
- Supports real-time GPS location.
- Filters: category, radius (0.5–30 km), and saved spots.
- Circle overlays show distances, with zoom-adaptive icons.



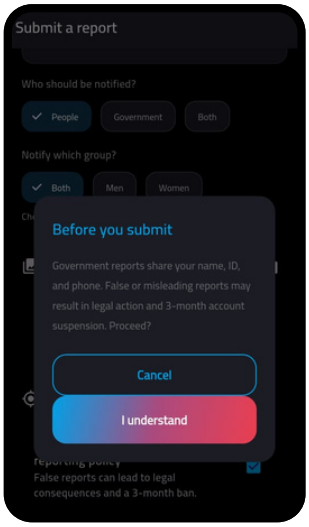
Map

06 Reports

Fetching : Nearby reports displayed on the map.
Submitting : Users can create reports with text and media uploads.
Realtime : Reports update live as they are added or removed.



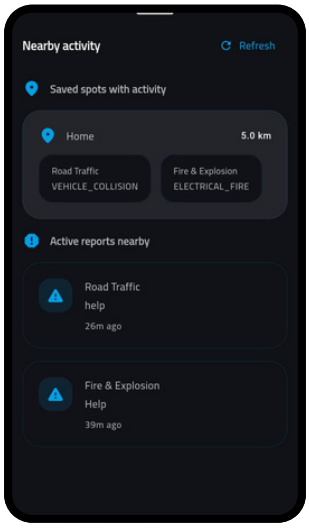
Nearby Reports



Submitting

07 Alerts

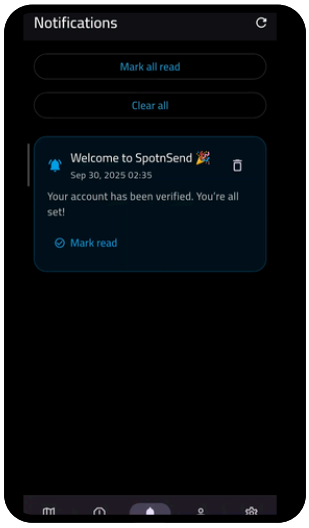
- View nearby alerts or all active alerts.
- Create alerts directly from reports.
- Resolve alerts once addressed.
- Alerts update in real time.



Nearby Activity

08 Notifications

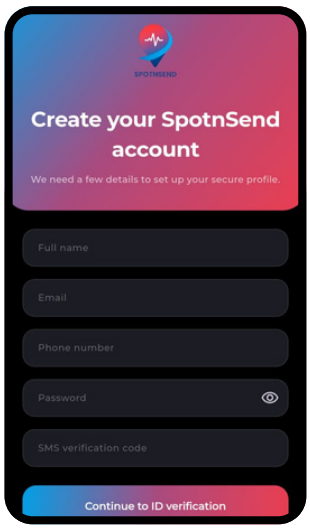
- Users receive notifications for updates and alerts.
- Notifications can be marked as read or cleared.
- Action bar supports “Mark all read” and “Clear all.”



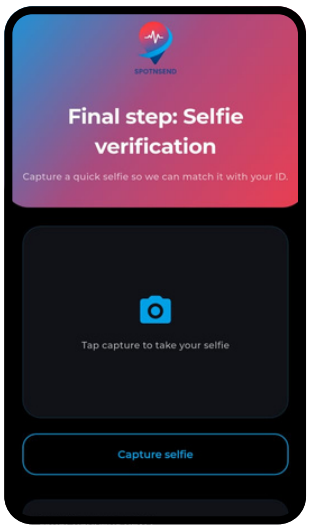
Notifications

09 Authentication & Accounts

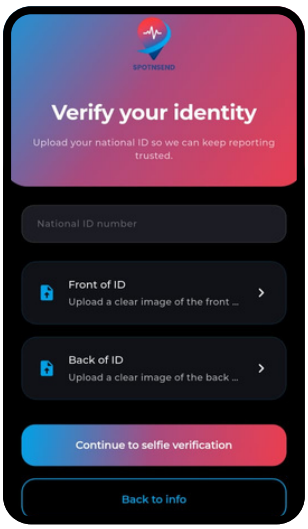
- Supports signup, login, and persistent sessions.
- **Signup flow** :
    1. Basic info + OTP verification.
    2. ID upload (front/back).
    3. Selfie capture for verification.
- Pending verification blocks report submission until approval.
- Password changes supported with re-authentication.



First Phase
(Basic Info)



Second Phase
(ID upload)



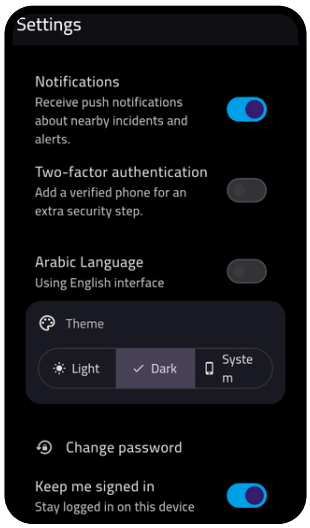
Third Phase
(Selfie Capture)

10 Two-Factor Authentication (TOTP)

- Users can enable 2FA via QR code setup.
- Verification with 6-digit codes.
- 2FA can be disabled with confirmation.

11 Settings

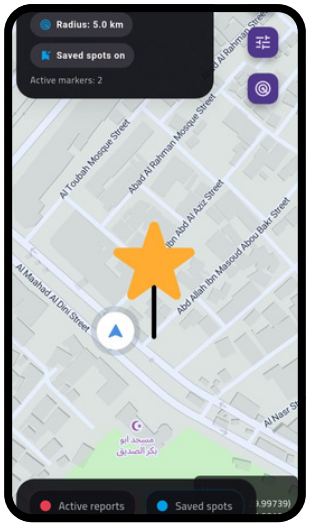
- Manage notifications, 2FA, language (English/Arabic), and theme (light/dark).
- Toggle “Keep me signed in.”



Settings Page

12 Saved Spots

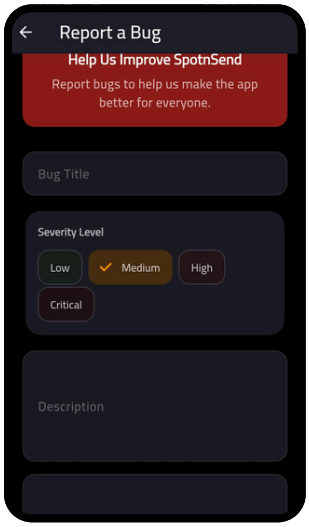
- Saved Spots : Users can add, remove, and display saved locations on the map.



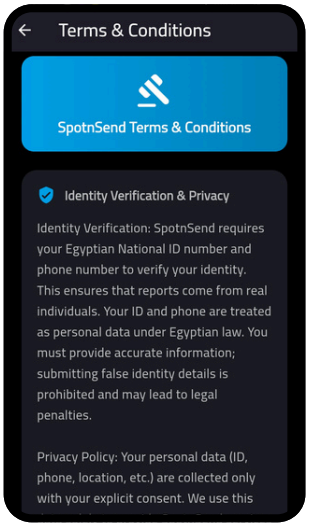
Saved Spots

13 Bugs, Terms, and Guides

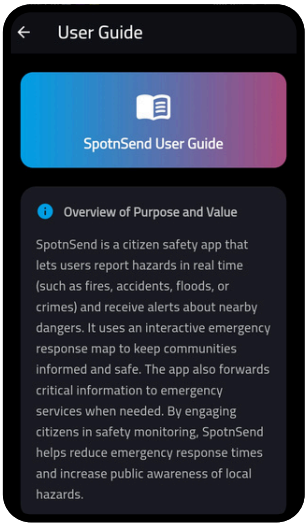
- **Bug Reports**: Users can submit issues with severity levels.
- **Static Pages**: Terms & Conditions and User Guide available in multiple languages.
- Emergency numbers highlighted for quick access.



Bug Reports



Terms & Conditions



SpotnSend User Guide

14 Shared Components

- Buttons, text fields, dialogs, and toasts built with a consistent theme.
- Reusable components provide a unified look and feel.

15 Security & Privacy

- Reports require verified identity to prevent misuse.
- False report warnings include legal disclaimers.
- Government-only features restricted to authorized users.
- Supports 2FA for enhanced security.
- Phone/email updates require confirmation.

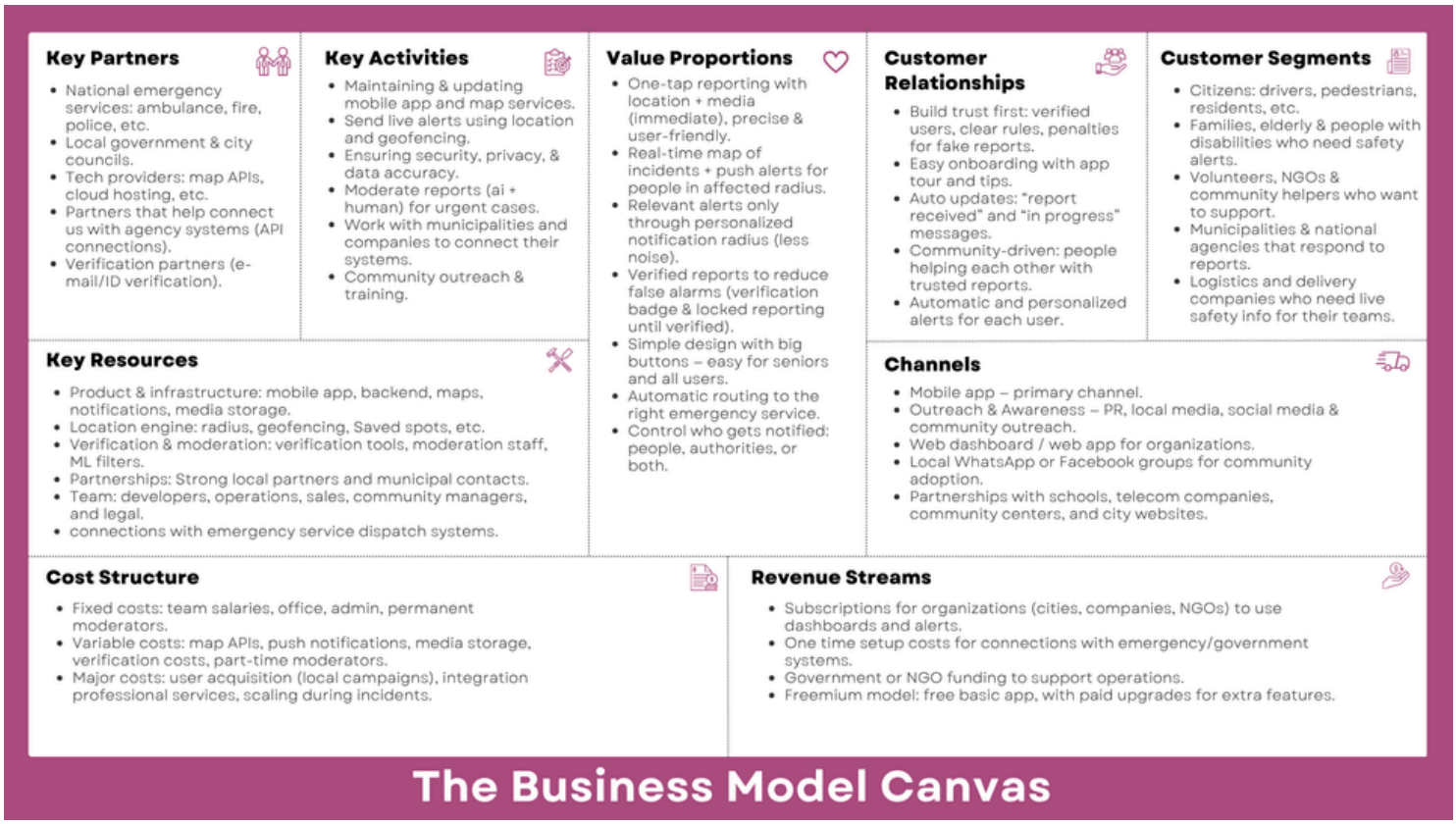
16 Extensibility

- New tabs can be added easily to the navigation structure.
- New report categories can be registered with an icon.
- Features can be extended while keeping widgets declarative and logic in services.

Team Members & Roles

Name	Role
Ahmed Shaker Helmy	Backend - Database Schema - Admin Dashboard Backend
Habiba Mohamed Anter	Business Model - Admin Dashboard frontend - Documentation
Mohannad Mohie Mohamed	Frontend - Testing - Admin Dashboard Backend
Noureen Khaled Saeed	Backend - Documentation - Admin Dashboard Backend
Raneem Mohamed Ibrahim	Pitch Video - Documentation - Admin Dashboard frontend - UI/UX

Business Model



1) Key Partners :

The main organizations and partners that we rely on

- National emergency agencies (ambulance, fire, police) : They are the direct responders to the reports.
- Local governorate & city councils : For local problems (repairing holes in streets, street lighting, etc.).
- Tech providers (map APIs, cloud hosting) : Provide infrastructure for maps, storage and performance.
- Partners that help connect us with agency systems : Help us connect the app with emergency agency systems.
- Verification partners : Ensures that the users are real and reduces false reports.

2) Customer Segments :

People that benefit from the service

- Citizens : Everyday users like drivers, pedestrians, and residents.
- Families, elderly & people with disabilities : Need extra safety alerts.
- Volunteers, NGOs & community helpers : Support user trust and community outreach.
- Municipalities & national agencies : Use the reports for faster response.
- Logistics & delivery companies : Safety teams that need safety information.

3) Value Proposition :

What makes the app distinct and attractive

- One-tap reporting : Quick and easy way for users to report incidents with location and media.
- Real-time map & alerts : See incidents instantly and receive push notifications in the affected radius that is chosen by the user.
- Personalized alerts : Only relevant alerts in the user's radius (to avoid noise).
- Verified reports : To reduce false reports through identity checks and verification badges.
- Simple design : Easy to use for senior and low-literacy users.
- Automatic routing : Reports go directly to the right responder (to reduce duplication).
- Control notifications : Users decide if alerts go to people, authorities, or both.

4) Channels :

How the app reaches users

- Mobile app : The main access point.
- Outreach & Awareness : PR, local media, safety workshops, and social media.
- Web dashboard : For enterprise and government access.
- Social / WhatsApp groups : For viral community adoption.
- Partnerships : through municipal websites, telecom companies, schools, and community websites.

5) Customer Relationships :

How to maintain trust

- Trust-first : Identity verification, clear terms of use, and penalties for users who fake reports.
- Self-service onboarding : In-app guided tour and tips for new users.
- Community-driven : Relies on citizens helping each other to achieve trust.
- Automatic personalized alerts : Smart, specific notifications for each user.

6) Key Activities :

The main steps should make to operate

- Maintain & update the app : Continuous development, bugs and errors fix.
- Notifications & geofencing : Deliver alerts in real time.
- Ensure security & accuracy : Protect user data and validate information.
- Moderate reports : Automated and human review of incidents.
- Municipalities & enterprises : Support connections with agencies and companies.
- Community outreach & training : Educating users and local partners.

7) Key Resources :

The essential resources required

- Product & infrastructure : Mobile apps, backend, maps, servers, storage.
- Location engine : Logic for radius, geofencing, and Saved Spots, etc.
- Verification & moderation : verification tools, staff, and AI filters.
- Partnerships : Municipal contacts and collaborations.
- Team : Developers, operations, sales, community managers, and legal staff.
- Integrations : Connections with emergency service dispatch systems.

8) Cost Structure :

Main costs involved in running the app

- Fixed : Salaries, office/admin costs, permanent moderation team.
- Variable : Map APIs, push notifications, media storage, moderators, etc.
- Major costs : Marketing campaigns, integration with agencies, scaling during major incidents.

9)Revenue Streams :

How the project generates money

- Subscriptions for organizations : Annual subscriptions for municipalities, government agencies, etc. Provides them with dashboards, and priority alerts.
- Setup & connections costs : One-time costs charged when connecting the system with emergency services, telecoms or government platforms.
- Government / NGO Funding : Grants or financial support from public institutions to cover operations or expansion costs
- Freemium : Attracts a wide user base with free basic access, while generating revenue from paid upgrades for advanced features like ai chatbot, extra Saved Spots, and family groups.

Future Plans

Multi-Language :
→ to support users in all languages.

Domain expansion :
→ improve platform's infrastructure to handle more users with higher speed and security.

Technical support :
→ to assets users and resolve problems quickly.

Call Centre :
→ for direct communication and user support.

Phone reporting :
→ to help users report easily, even without internet

Ai verification :
→ implement ai system to automatically verify reports and data to reduce errors.

Quality control after id verification :
→ to ensure high quality service.

Ai chatbot :
→ to assist users with instant answers.

SMS API :
→ connect an SMS API for direct communication and confirmations.

SMS :
→ ability of submitting reports through SMS to expand accessibility.

Challenges & Solutions

Challenge : Balancing design and technical work within the team.
Solution : Instead of sticking to fixed roles, members helped each other across tasks UI ,backend ,testing , so progress was steady even under pressure.

Challenge : Designing a UI/UX that was both simple to use and modern.
Solution : We chose fun, meaningful colors and added both light and dark modes to make the app more accessible and visually appealing.

Challenge : Building the admin dashboard in a way that isn't overwhelming.
Solution : We focused on clarity using tables, filters, and clear navigation so admins can understand and control everything without confusion.

Challenge : Defining a business model and ways to raise funds for the app.
Solution : We brainstormed many directions and shortlisted the most practical ideas that can be applied in the near future.

Challenge : Handling identity verification securely and smoothly.
Solution : We integrated ID images and status management into the dashboard, making it possible to verify, unverify, or ban users quickly.

Challenge : Choosing the right backend and handling deployment.
Solution : After testing different options, we went with Supabase as a BaaS because it gave us authentication, storage, and Postgres without setting up our own server.

Challenge : Making the admin dashboard work well in both Arabic and English.
Solution : We added bilingual support without flipping the layout, so it remains consistent and easy to use in either language.

Conclusion

At the end of this journey, SpotnSend is more than just a student project , it is a vision for safer streets and stronger communities in Egypt. Every day we hear of accidents that could have been avoided, delays that cost lives, or small problems left unattended until they turn into disasters. These are not distant issues; they happen in our neighborhoods, on our way to university, and in the very streets we walk with our families.

Through SpotnSend, we wanted to give citizens a voice and responders a tool. An application that is simple to use, modern in design, and built with features that truly serve people: from quick reporting of incidents and complaints, to a clear admin dashboard that empowers officials to act faster. By making alerts instant and accessible, we believe response times can improve, accidents can be prevented, and lives can be saved.

What makes this project special is that it was shaped with Egypt in mind , the colors, the language support, the challenges we solved, all were done while thinking about how Egyptians interact with technology in their daily lives. Whether it's a dark streetlight, a sudden fire, or a road accident, SpotnSend turns ordinary citizens into active participants in safety.

We know that change never comes from technology alone, but from people who choose to use it for good. SpotnSend is a step toward a more connected, aware, and responsible society , one where help doesn't wait, and where everyone can play a role in protecting each other. If applied at scale, this app could not only improve safety but also strengthen the bond between citizens and authorities, showing how collaboration can transform lives.

SpotnSend began with an idea, grew through teamwork, and now stands as a proof that with creativity and determination, young Egyptians can design solutions for real problems. Our hope is that this project becomes more than a competition entry , that it becomes a spark for a safer, smarter Egypt.

SpotnSend ~ Stay Aware. Stay Safe.

References

[1] “Business Model Canvas Explained with Examples,” YouTube. [Online]. Available: https://youtu.be/CakUeC1sCSs?si=z0q5425FAGYZqXYw

[2] “Master the Business Model Canvas,” YouTube. [Online]. Available: https://youtu.be/Snau1uizuW0?si=oeGSWsFptyRSIP2A

[3]“Node.js Documentation,” Node.js. [Online]. Available: https://nodejs.org/en/docs

[4]“Express – Web Framework for Node.js,” Express. [Online]. Available: https://expressjs.com/

[5]“Supabase Documentation,” Supabase. [Online]. Available: https://supabase.com/docs

[6]“PostgreSQL Documentation,” PostgreSQL. [Online]. Available: https://www.postgresql.org/docs/

[7]“PostGIS Documentation,” PostGIS. [Online]. Available: https://postgis.net/documentation/

[8] “Flutter Documentation,” Flutter. [Online]. Available: https://docs.flutter.dev/

[9]“Dart Guides,” Dart. [Online]. Available: https://dart.dev/guides

[10]“MapLibre Project,” MapLibre. [Online]. Available: https://maplibre.org/