WOMAP – Women’s Safety Route Companion

SUNEETH KOKALA

4252195

Introduction

- Women’s safety remains a major concern in modern urban environments.
- Womap is a web based solution designed to enhance women’s safety through intelligent route planning and real time alerts.
- The system analyzes available routes and recommends the **safest possible path** based on area crime data and user reports.
- Users can **report incidents**, **receive safety alerts** and **share live locations** with trusted contacts.
- Combines **AI powered risk assessment**, **mapping APIs** and **real time communication tools** to ensure proactive protection.
- Designed for accessibility, reliability and scalability with a **FastAPI, React, PostgreSQL** architecture.

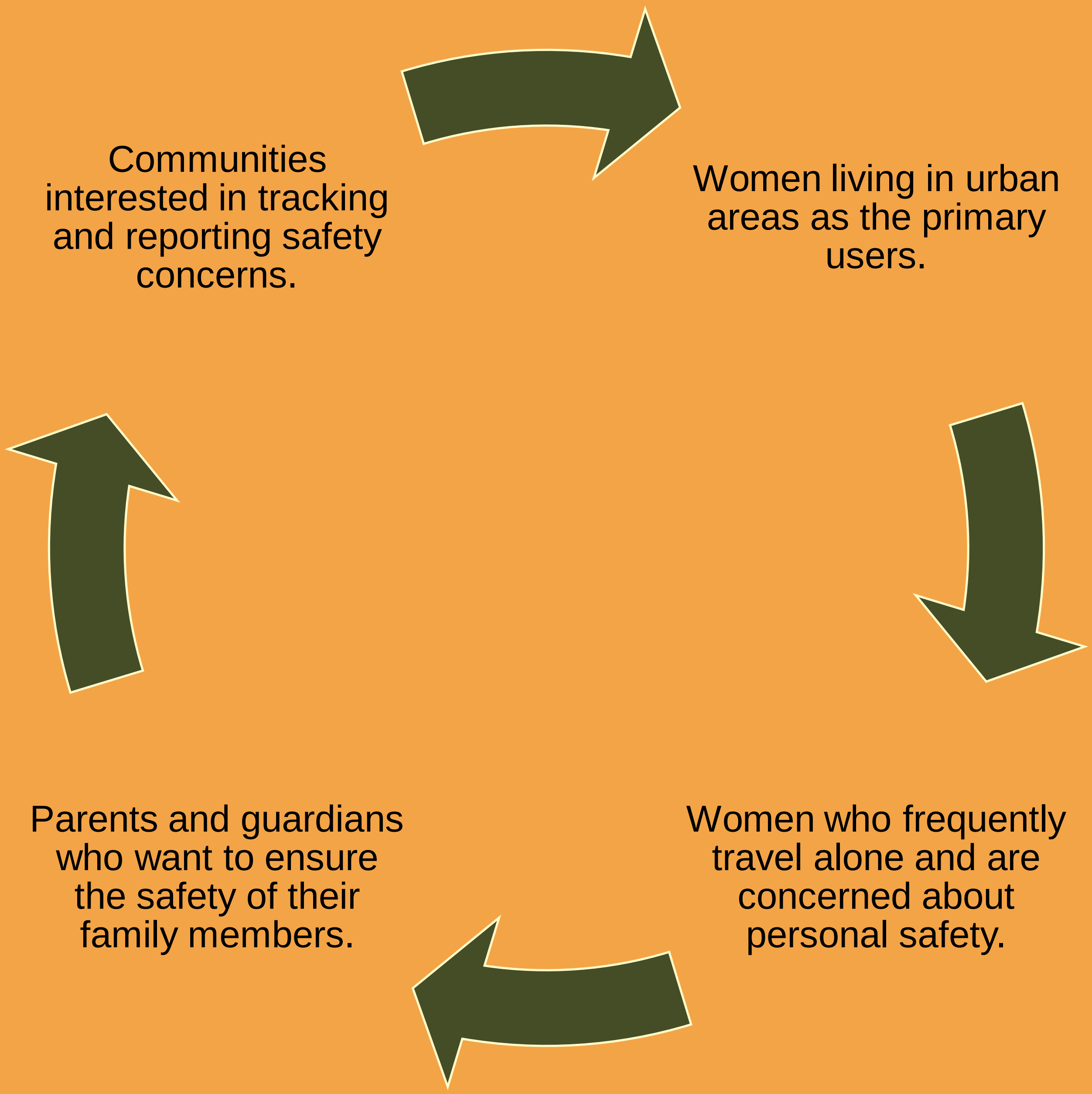
Problem Statement

- Industrial sites such as factories and construction areas face ongoing safety risks due to **noncompliance with PPE regulations**.
- Manual monitoring is **inefficient, unreliable and lacks real time alerts** leading to increased **accidents and injuries**.
- There is a need for an **automated system** that can **detect PPE compliance, alert supervisors instantly and generate safety reports** to enhance workplace safety and accountability.

Objectives and Scope

- Provide women with a reliable web based companion tool for safer travel in urban environments.
- Integrate mapping APIs, real time crime data and user feedback to recommend safer routes instead of just the shortest ones.
- Develop a web based frontend for user interaction.
- Implement a backend service with route optimization logic.
- Maintain a database to store crime incidents and safety reports with real time notification support.
- Include MVP features such as user registration, safe route suggestions and incident reporting while reserving advanced AI based predictive safety scoring for later phases.
- Users can also share their live location with their trustworthy persons like parents or guardians.

Target Group



Communities interested in tracking and reporting safety concerns.

Women living in urban areas as the primary users.

Women who frequently travel alone and are concerned about personal safety.

Parents and guardians who want to ensure the safety of their family members.

Risks

- **Technical risks**: Integration with real time mapping APIs may be complex.
- **Data risks**: Limited or incomplete safety/crime datasets could affect recommendation accuracy.
- **Adoption risks**: Users may hesitate to rely on the app if trust and accuracy are not established.

Project Plan:

- **Conception Phase (current):** Define objectives, scope, requirements and architecture.
- **Development Phase:** Implement core features (frontend, backend, route planning, incident reporting).
- **Finalization Phase:** Cloud deployment, testing, documentation and installation instructions.

Project Organization:

- As this is a course project, All the roles will be performed by myself:
  - **Developing**: Building frontend and backend features.
  - **Testing**: Validating functionality and requirements.
  - **Documentation**: Preparing project documentation, diagrams and reports.

Software Development Methodology

- The project will follow an **Agile methodology** to handle evolving requirements especially with user experience and data integration.
- Development will proceed through **iterative sprints**, delivering incremental features such as route planning, incident reporting and real time alerts.
- **Product Owner (me):** Defines requirements and prioritizes features.
- **Scrum Master (tutor):** Ensures Agile principles are applied throughout project phases.
- **Development (self managed role):** Handles coding, testing and integration with continuous feedback loops to reduce risks, show progress and allow scope adjustments.

Functional Requirements (FRs)

- **User Management:** Account creation with verification, secure login and profile update/delete.
- **Route Planning & Safety Guidance:** Enter start/destination get multiple route options, highlight safest route and view on interactive map.
- **Safety Data Integration:** Use crime/safety database, allow incident reporting and show past reports for routes/locations.
- **Real time Alerts & Notifications:** Notify users in unsafe areas and update routes if safer paths are found.
- **Admin Features:** Approve/edit/delete incident reports and manage safety datasets.

Non-Functional Requirements (NFRs)

- **Performance:** Fast route suggestions (≤3 sec) and support 100+ simultaneous users.
- **Security:** Encrypt user data, enforce HTTPS and restrict access to authorized users.
- **Usability:** Intuitive, mobile friendly interface, main functions accessible within 3 clicks.
- **Availability & Reliability:** 99% uptime and quick recovery from failures.
- **Maintainability & Scalability:** Modular code for easy feature additions and scalable for more users/data.

System Design

Actors: User, Admin, Mapping API, Database

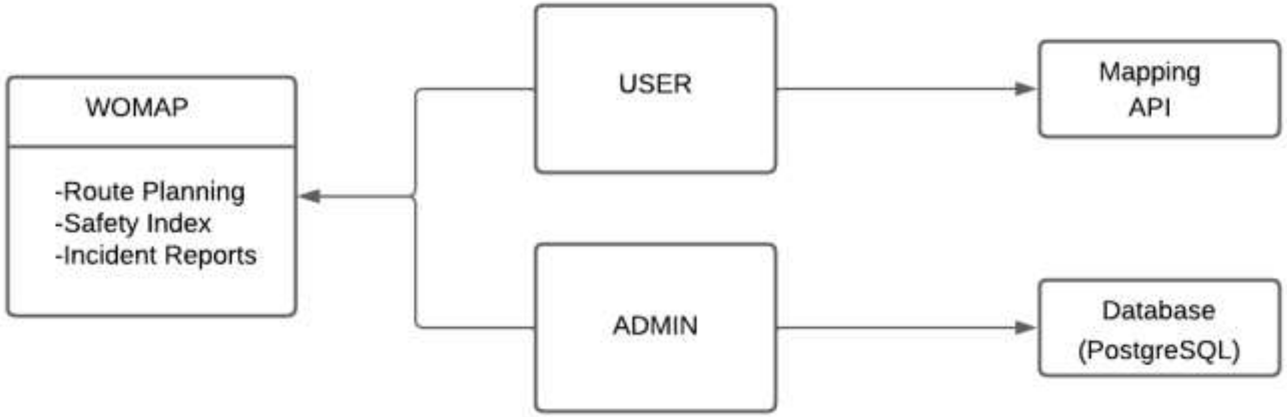


Figure1: System Design

System Decomposition into Building Blocks

A. **Presentation Layer (Frontend – React.js)**
   - Map and Route Visualization
   - User Authentication
   - Incident Reporting Form

B. **Application Layer (Backend – FastAPI)**
   - Route Calculation Service (mapping APIs, safety index)
   - Incident Management Service (reporting, storage, retrieval)
   - User Management Service (registration, login, authentication)
   - Notification Service (Whatsapp alerts)

C. **Data Layer (Database – PostgreSQL)**
   - User Table (profiles, credentials)
   - Incident Table (reports, timestamps, severity)
   - Safety Data Table (crime datasets, indexes)

Architecture Decomposition Diagram

Components & Dependencies:

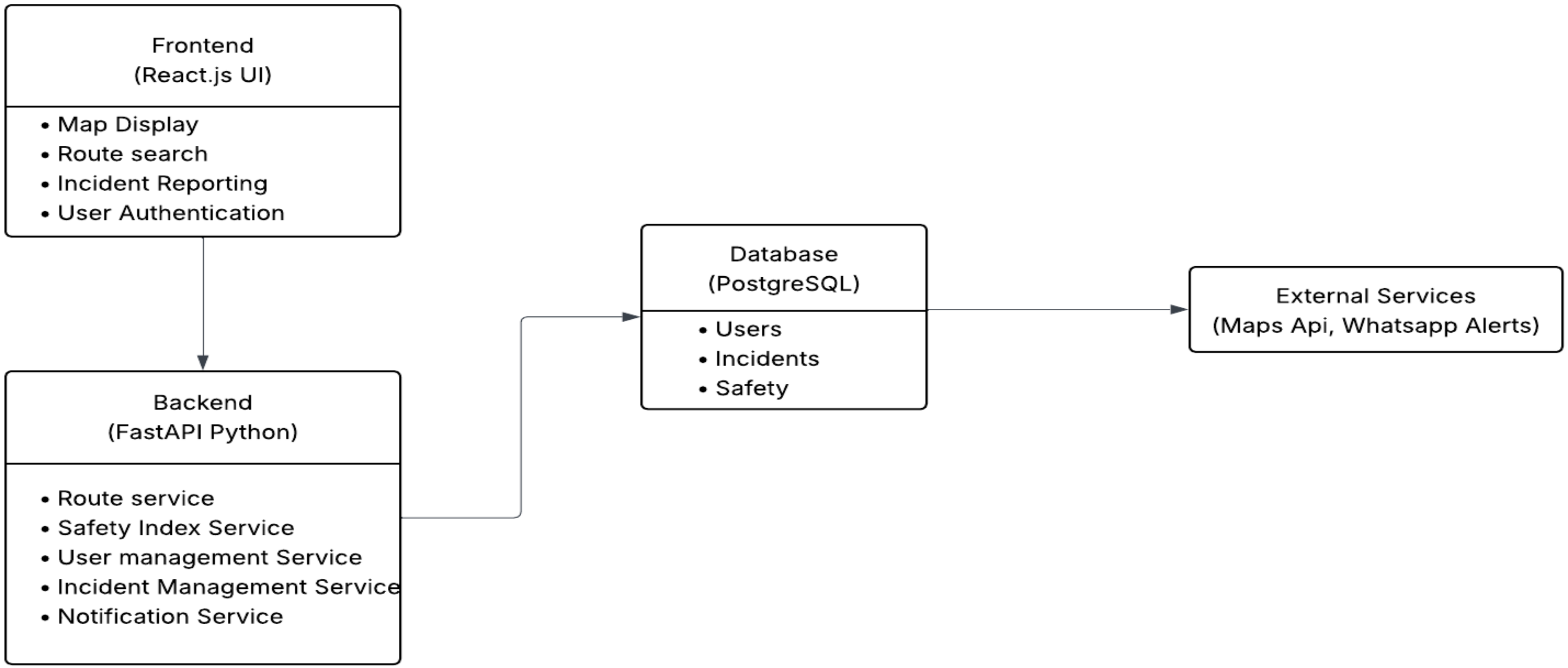


Figure 2: Architecture Decomposition Diagram

Summary of Architecture

- The Womap system follows a **three layer architecture**:
    - **Frontend (Presentation Layer – React.js)**
        - Provides the web based user interface.
        - Key features: Map visualization, safe route search, incident reporting and user authentication.
        - Delivers an intuitive and responsive experience to end users.
    - **Backend (Application Layer – FastAPI)**
        - Acts as the core of the system, processing requests from the frontend.
        - Main services:
            - **Route Service:** Fetches and calculates safest routes using external mapping APIs.
            - **Incident Service:** Manages reporting and retrieval of safety incidents.
            - **User Service:** Handles registration, login and authentication.
            - **Notification Service:** Sends safety alerts (via Whatsapp APIs).
        - Ensures data flow between frontend and database.

[Cont…] Summary of Architecture

- **Database (Data Layer - PostgreSQL)**
  - Stores user data, incident reports and safety datasets.
  - Provides persistent and structured storage for fast retrieval.
  - Supports geospatial extensions for route safety calculations.
- **External Services**
  - **Mapping API (Google Maps):** Supplies route and geolocation data.
  - **WhatsApp (Twilio):** Sends real time alerts to users or admins.

Flow of Data:

- User requests a safe route.
- Frontend sends request to FastAPI endpoint /route.
- Backend calls MapService to fetch route data.
- MapService queries crime data from the database.
- External API (Google Maps) returns route options.
- Backend computes safety score and sends result to frontend.
- Frontend displays the route with color coded safety markers

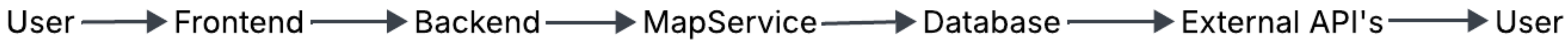


Figure 3: Flow of Data

Glossary of Key Terms

- **Womap** - Web app providing safe travel routes for women.
- **MVP** - Minimum functional version with core features.
- **User** - Individual using Womap to plan routes/report incidents.
- **Admin** - User managing reports and datasets.
- **Route Planner** - Calculates routes considering distance and safety.
- **Safety Index** - Numerical measure of route safety.
- **Incident Report** – User submitted report of unsafe event/location.
- **Crowdsourced Data** – User submitted safety information.

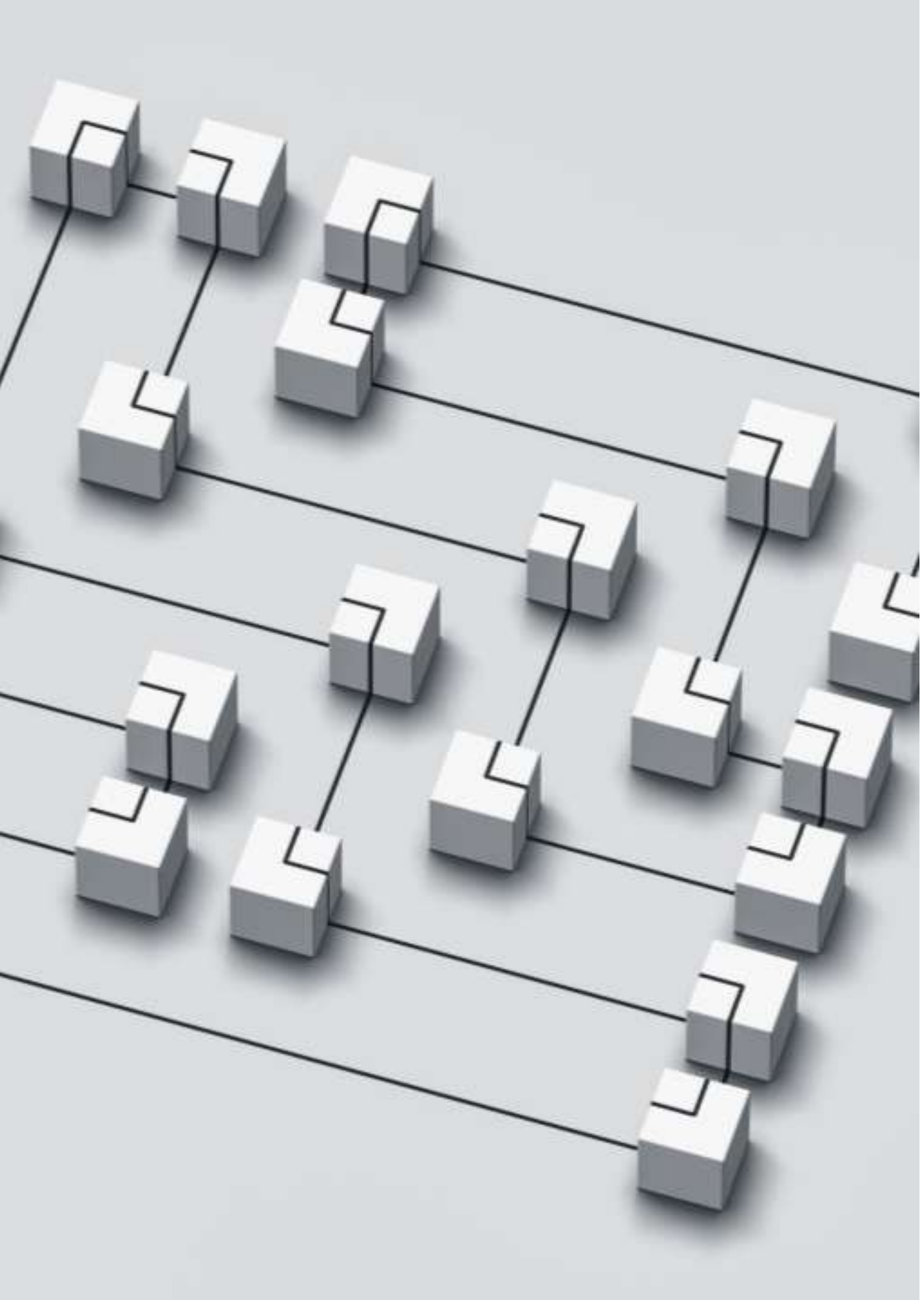
[Cont…] Glossary of Key Terms

- **Crowdsourced Data** – User submitted safety information.
- **Mapping API** - External service for maps and routes (e.g., Google Maps).
- **Frontend** - Web UI (React.js).
- **Backend** - Server logic (FastAPI/Python).
- **Database** - Stores users, incidents, safety data (PostgreSQL).
- **Real time Alerts** - Notifications for unsafe areas/routes.
- **Technical Debt** - Features postponed for future work.

Design and Implementation:

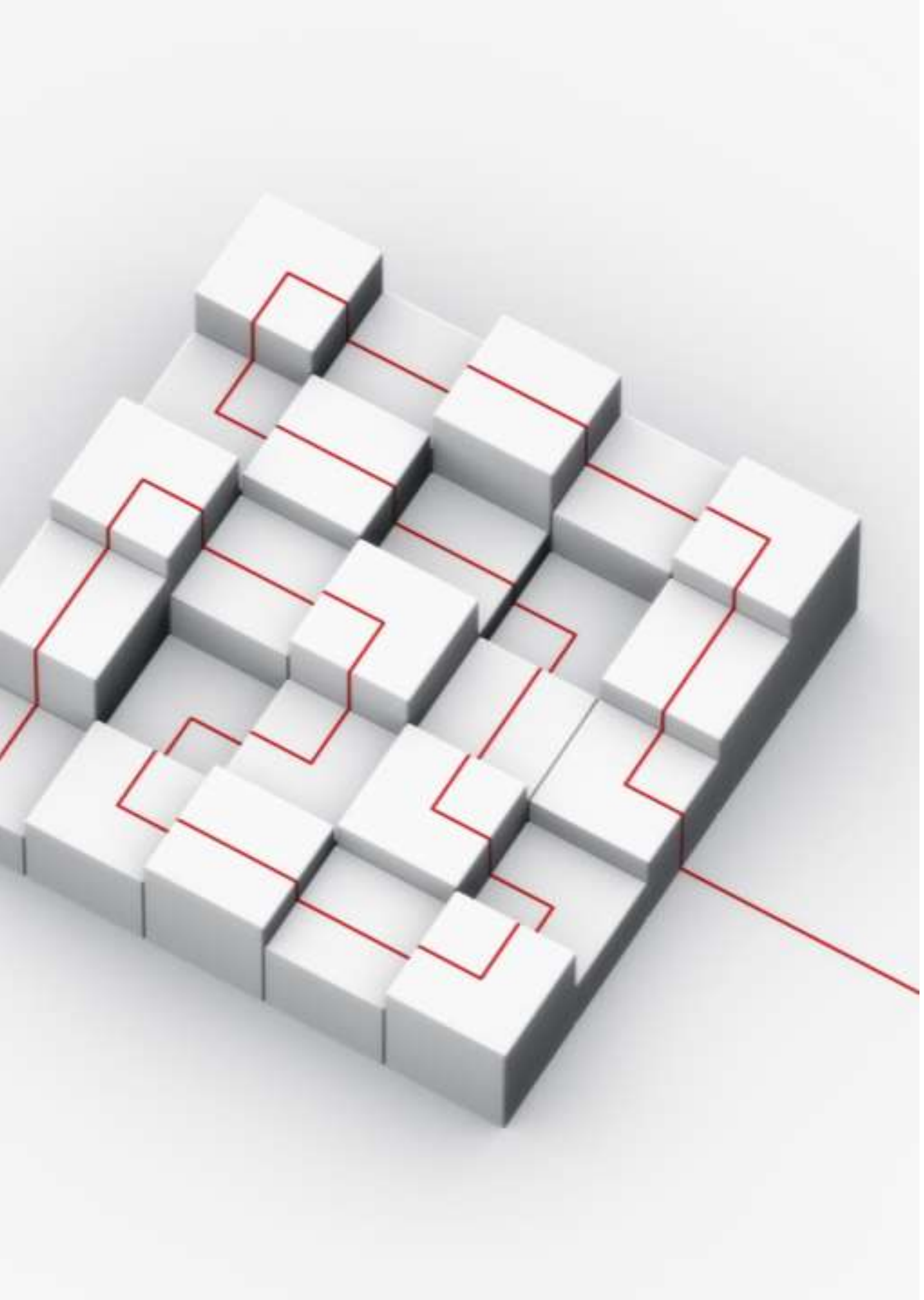
Design Overview:

- Designed a **modular** and **scalable architecture** based on the FastAPI React PostgreSQLstack.
- Adopted a **client server model** ensuring separation of concerns and maintainability.
- Defined **key modules**: User Authentication, Route Safety Analysis, Incident Reporting, Alert Service and Admin Dashboard.
- Integrated **external APIs (Google Maps, Twilio)** for real time route data and safety notifications.
- Ensured **responsive design** and a user friendly interface for seamless experience across devices.
- Prioritized **security and privacy** using encrypted credentials and safe API communication (HTTPS).



Implementation Details:

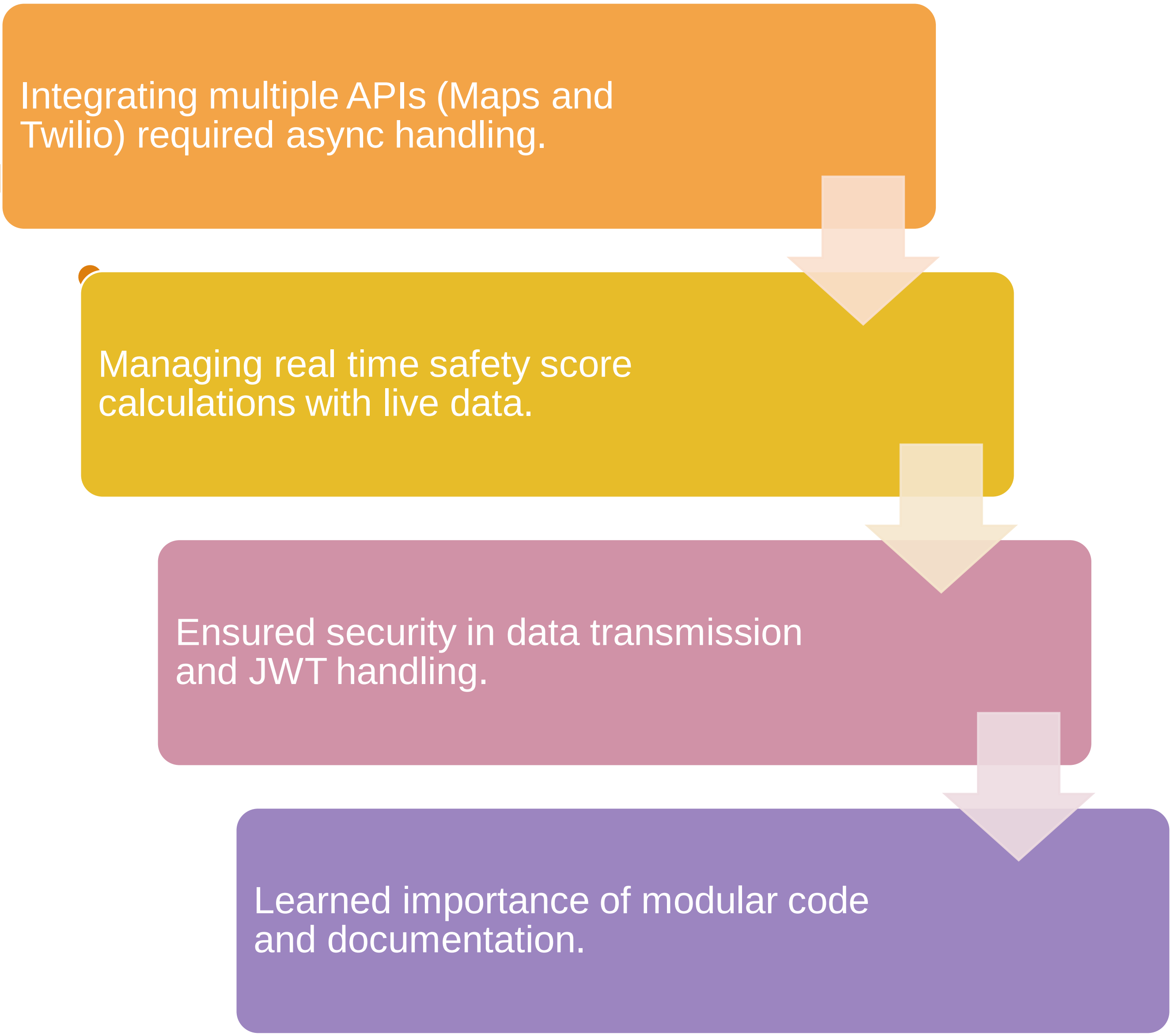
- **Frontend** built with React.js for dynamic UI and modular component structure.
- **Backend** developed using FastAPI providing RESTful APIs for data processing and communication.
- **Database** implemented with *PostgreSQL* using relational schema to store users, incidents and routes.
- **Map services** integrated with Google Maps API to compute and visualize safe routes.
- **Alert system** built with Twilio API to send real time WhatsApp alerts in emergencies.
- **Dockerized environment** for both backend and frontend, ensuring easy deployment and version control.
- Added **logging and exception handling** to enhance reliability and simplify debugging.
- Used **GitHub for version control** and maintained clean, commented and documented code.



Design Decisions and Rationale:

- **FastAPI** chosen for async performance and type safety.
- **PostgreSQL** chosen for relational and spatial queries.
- **React.js** for fast and interactive front end.
- **Docker** for consistent development and deployment.
- **GitHub** for version control and CI/CD pipeline management.

Implementation Challenges and Learnings



Integrating multiple APIs (Maps and Twilio) required async handling.

Managing real time safety score calculations with live data.

Ensured security in data transmission and JWT handling.

Learned importance of modular code and documentation.

References and hyperlinks to the frameworks

- Google Maps Platform. (2025). Routes and Directions APIs Documentation. Retrieved from

  https://developers.google.com/maps/documentation

- Twilio. (2025). Twilio Messaging API for WhatsApp and SMS. Retrieved from

  https://www.twilio.com/docs/whatsapp

- FastAPI. (2025). FastAPI Framework High performance web framework for building APIs. Retrieved from

  https://fastapi.tiangolo.com/

- React.js. (2025). React A JavaScript library for building user interfaces. Meta Platforms, Inc.

  https://react.dev/

- PostgreSQL Global Development Group. (2025). PostgreSQL 16 Documentation.

  https://www.postgresql.org/docs/

- GitHub Docs. (2025). Version Control and Collaboration using Git and GitHub.

  https://docs.github.com/

Conclusion

- Womap provides a reliable web based route companion to enhance women's personal safety in urban areas.
- It uses mapping APIs, real time crime data and user feedback to prioritize safer routes.
- Features include user registration, incident reporting, interactive maps and real time alerts.
- Agile methodology ensured iterative development, continuous feedback and secure, scalable delivery.
- The system empowers informed travel, promotes community awareness and supports future enhancements like AI based safety scoring.

References:

- **Gupta, P., & Rani, P.** (2022). Women safety application using location tracking and emergency alert system. International Journal of Innovative Research in Computer and Communication Engineering, 10(4), 2456–2462. https://doi.org/10.15680/IJIRCCE.2022.1004005
- **Patel, A., & Sharma, R.** (2021). A smart route recommendation system for women safety using GIS and IoT. International Journal of Advanced Computer Science and Applications (IJACSA), 12(7), 133–140. https://doi.org/10.14569/IJACSA.2021.0120717
- **Chakraborty, S., & Saha, S.** (2021). SafePath: A location aware mobile application for safe route prediction. IEEE International Conference on Smart Cities and Green ICT Systems (SMARTGREENS), 112–120. https://doi.org/10.1109/SMARTGREENS52869.2021.9462374
- **Kumar, N., & Mishra, R.** (2023). AI based predictive route optimization for personal safety in urban areas. IEEE Access, 11, 45029–45041. https://doi.org/10.1109/ACCESS.2023.3267001
- **Tripathi, R., & Banerjee, A.** (2020). Geo spatial data analytics for safe city navigation and women safety enhancement. Journal of Urban Technology, 27(5), 95–112. https://doi.org/10.1080/10630732.2020.1798682