AccessMap

TOOLING SELECTION AND JUSTIFICATION

Arek Pająk, Julia Pozorska

Programming Languages

Python
Purpose: Backend development, data processing, and integrating mapping algorithms.
Justification: Python's rich libraries (e.g., NumPy, Pandas, and GeoPandas) and frameworks like Flask or Django make it ideal for handling geospatial data and APIs. It is widely used in data science, which aligns with the need for processing accessibility data efficiently.

JavaScript (TypeScript)
Purpose: Frontend development for a responsive, interactive user interface.
Justification: JavaScript, enhanced by TypeScript, is essential for creating interactive web applications. Frameworks like React or Vue.js provide flexibility and scalability for building intuitive navigation and map-based features.

SQL
Purpose: Database management for storing detailed accessibility data.
Justification: SQL is the standard for relational databases, which are necessary for efficiently querying accessibility data, user settings, and map features.

Frameworks

React (Frontend)
Purpose: Building user-friendly interfaces for web and mobile platforms.
Justification: React's component-based architecture facilitates the development of a dynamic interface and ensures reusability across web and mobile platforms (via React Native).

Django/Flask (Backend)
Purpose: Backend web framework for handling APIs and server-side operations.
Justification: Django's ORM simplifies database interactions, while Flask offers lightweight flexibility for API development.

Leaflet or Mapbox
Purpose: Interactive mapping and visualization of accessibility routes.
Justification: Leaflet is lightweight and customizable, while Mapbox provides robust tools for geospatial data visualization and overlays, perfect for accessibility-focused maps.

Integrated Development Environments (IDEs)

Visual Studio Code

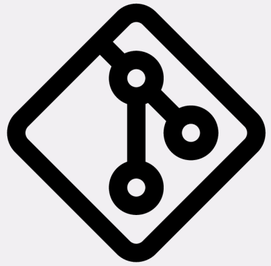
- Purpose: Development environment for coding.
- Justification: VS Code supports multiple programming languages (Python, JavaScript) and offers extensions like ESLint and Prettier, making it ideal for collaborative and efficient coding.

Project Management Software

Jira

- Purpose: Agile project management.
- Justification: Jira supports Scrum/Kanban boards and issue tracking, making it ideal for managing tasks and sprints for a multi-phase project like AccessMap.

Version Control Systems



Git (with GitHub)

- Purpose: Version control and collaboration.
- Justification: Git is an industry-standard for version control, and GitHub offers robust collaboration features, such as pull requests and code reviews.

CI/CD Systems

GitHub Actions
Purpose: Continuous integration and deployment.
Justification: GitHub Actions integrates seamlessly with GitHub repositories, automating testing, builds, and deployments to ensure code reliability.

Design Tools

Figma
Purpose: Prototyping and UI/UX design.
Justification: Figma's collaborative design features allow teams to create and iterate on interface designs efficiently, essential for accessibility-focused UI.

Adobe Illustrator
Purpose: Creating detailed map overlays and visual assets.
Justification: Illustrator provides advanced tools for designing precise and visually appealing elements for the map interface.

Collaboration Platforms

Slack
Purpose: Team communication and collaboration.
Justification: Slack’s integrations with tools like Jira, GitHub, and Figma streamline project updates and team communication.

Notion
Purpose: Documentation and knowledge sharing.
Justification: Notion helps organize project documentation, user stories, and accessibility guidelines in a centralized space.

Database Management Systems

PostgreSQL with PostGIS
Purpose: Storing and managing geospatial data.
Justification: PostgreSQL is scalable and reliable, while PostGIS extends it to handle geospatial data, essential for mapping and route calculations.

Testing Tools

Pytest
Purpose: Unit and integration testing.
Justification: Pytest is a Python testing framework suitable for validating backend functionality and API responses.