

# TOOLING SELECTION

Filip Łapiński

Adam Oporski

Gabriela Jastrzębska

# FRONTEND

- **React:** widely used, well-documented, and flexible; its component-based architecture allows for reusable and efficient UI development
- **React Native:** cross-platform apps (iOS and Android) with the same codebase as your web app; saves development time and costs compared to native solutions
- **React Navigation:** simplifies routing and navigation in React Native apps



# BACKEND

- **Node.js:** event-driven REST APIs; fast and works well with JavaScript, which our team already knows
- **Python:** better suited for data-heavy processing and advanced algorithms; its rich ecosystem (NumPy, Pandas, Scikit-learn) simplifies building analytical features
- **API Frameworks:** Express.js for Node.js, Flask for Python



# DATA BASE

- **PostgreSQL:** best for structured data, complex queries, and transactional operations like user data, booking systems, and financial records
- **MongoDB:** great for unstructured or semi-structured data like user activity logs, preference settings, or flexible JSON-like objects



# HOSTING

## MVP:

- **Render**: simple, cost-effective, and provides hosting for both frontend and backend with minimal configuration; ideal for a small team focusing on product development without diving into complex DevOps early on

## Further development:

- **Google Cloud Platform**: robust tools for data processing and machine learning, making it ideal for apps that incorporate complex algorithms and analytics



# TESTING

## Frontend:

- **Jest + React Testing Library:** these tools focus on testing React components efficiently, simulating user interactions
- **Cypress:** E2E tests for web applications
- **Detox:** E2E tests for mobile applications

## Backend:

- **Jest:** works well with Node.js API
- **Pytest:** most popular python testing tool with large community support
- **Supertest:** works well for integration testing REST APIs



# DEVOPS

## MVP:

- **Docker**: ensures your app works consistently across different environments by packaging it into isolated containers
- **Github actions**: automates testing, building, and deployment workflows; easy to implement

## Further development:

- **Kubernetes**: essential for scaling your app in production as it allows you to manage multiple containers efficiently



# MAPPING API

- **Google Maps Platform** - comprehensive mapping data with global coverage; route planning (driving, cycling, walking, and public transit); real-time traffic updates, predictive traffic data; Extensive Places API for detailed information on landmarks and businesses
- **MapBox** - customizable maps with dynamic styles; offline maps for mobile applications; turn-by-turn navigation and route optimization



THANK YOU  
FOR YOUR  
ATTENTION

Filip Łapiński  
Adam Oporski  
Gabriela Jastrzębska