NY Civic Sphere

Empowering New Yorkers with AI-Powered Civic Intelligence

Microsoft AI Innovation Challenge
November 2025

Hackers:
- **Damanpreet Kaur**
- **Olabimpe Sanni**

Problem Statement

The Microsoft Hackathon has challenged us to build an AI-driven civic engagement platform designed to solve the following problems:

Information Accessibility & Reliability

- **Lack of easy access to accurate, timely local information**
  Residents struggle to find trustworthy updates about local policies, events, or services.
- **Fragmented or non-interactive traditional channels**
  Existing sources (websites, flyers, hotlines) are outdated, hard to navigate, and do not support two-way communication.
- **Need for aggregated, trusted public/government data**
  Information is scattered across multiple agencies, making it difficult for residents to gather needed facts.

User Experience & Comprehension

- **Desire for conversational, AI-powered access**
  People want simple, natural-language interaction to understand complex civic information.
- **Requirement for personalized updates**
  Users benefit from tailored notifications that reflect their location, interests, and civic priorities.
- **Need for simplified explanations of civic content**
  Residents require clear, unbiased breakdowns of ballot measures, political issues, and public statements.

Community Engagement & Trust

- **Encouraging respectful community dialogue**
  People need a safe, moderated space to discuss topics without toxicity or intimidation.
- **Combatting disengagement and misinformation**
  A trustworthy civic platform should reduce confusion, increase participation, and prevent the spread of false information.

Project Goals

Unified Information Access

- Centralize civic information in one dashboard.
- Surface relevant local information with context.

Timely & Trusted Updates

- Provide real-time updates on policies, events, and elections.

Intelligent Civic Assistance

- Enable AI-powered Q&A about NYC civic matters.

Stronger Community Participation

- Improve civic engagement through accessible information.

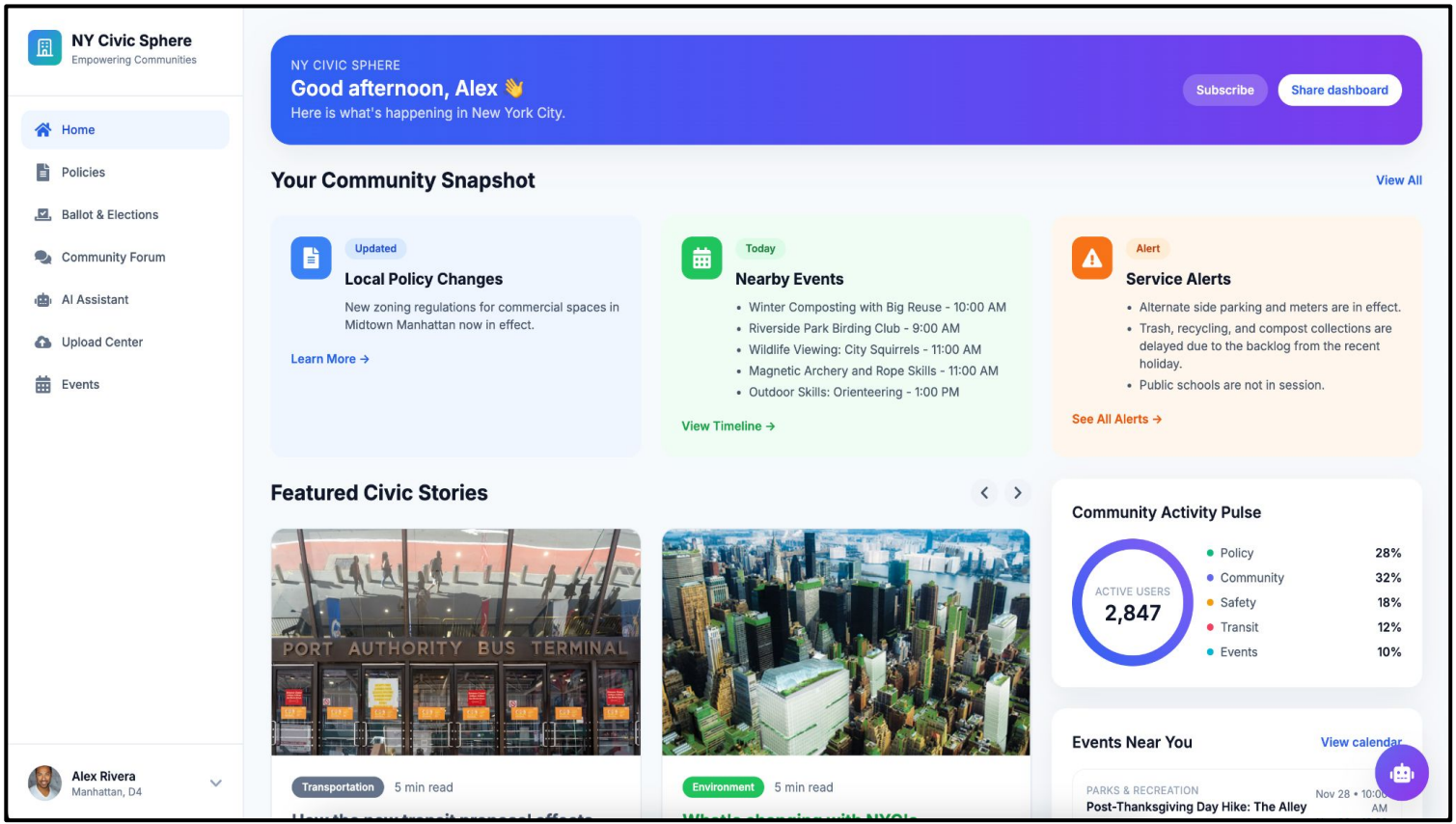
Our Solution - NY Civic Sphere

An AI-powered civic engagement platform that centralizes NYC's public data into one intuitive dashboard and delivers real-time updates on policies, events, and elections, allowing residents to ask natural-language questions, receive contextual local insights, and easily access the information they need to stay informed and engaged.

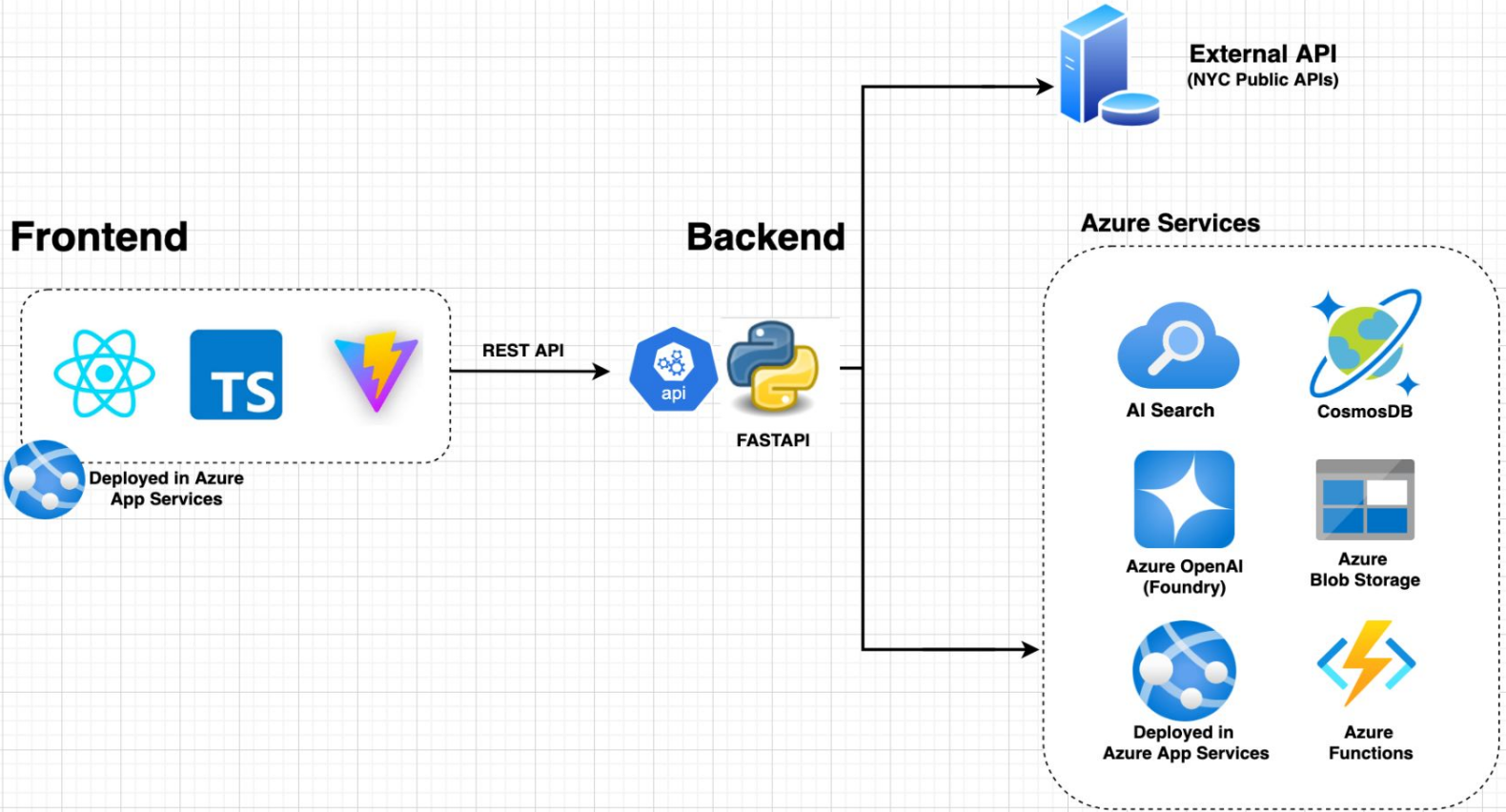
Core Features

- **Community Snapshot**: Real-time metrics and trends
- **Featured Stories**: Civic news and updates
- **Policies & Rules**: Legislative updates with highlights
- **Discussions**: Community sentiment and trending topics
- **Events & Elections**: NYC Calendar API integration + ballot information
- **Service Alerts**: Real-time NYC service notifications (Powered by NY Public APIs)
- **AI Assistant**: RAG-powered Q&A with source citations (Powered by Azure AI Search + Azure OpenAI)

What We Built - NY Civic Sphere



Architecture & Technical Approach



How We Built It

- **Iterative development:** Began with mock data and progressively integrated Azure services and external APIs.
- **Backend using FastAPI:** Defined FastAPI schemas and routes before building UI components, ensuring a consistent contract between frontend and backend.
- **Component-driven frontend:** Created reusable React components being used across multiple forum pages.
- **Seamless local development:** Used FastAPI with a Vite proxy for smooth frontend–backend integration.
- **Use of public NYC APIs:** To minimize setup effort and eliminate subscription costs, we used public NYC.gov APIs for live civic information.
- **Azure AI Search (RAG):** To deliver accurate and trustworthy AI responses, we built a RAG pipeline using public PDFs on communities and shelters, stored them in **Azure Blob Storage**, and indexed them with AI Search for citation-backed answers
- **Azure Functions:** Generated AI-powered summaries on the Community Page using lightweight serverless functions.
- **Cosmos DB:** Stored user chat interactions and community form conversations for flexible NoSQL data storage.
- **Azure App Services:** Hosted both the React frontend and FastAPI backend for easy deployment and scaling.

Key Learnings

- **Start with Mock Data:** Use mock data first, then add real APIs and Azure services gradually.
- **API-first approach:** Define FastAPI routes and schemas early so frontend and backend stay consistent.
- **Modular frontend:** Reusable React components helped build the solution quickly and customize them later if needed.
- **Local setup:** Hot reload from npm run dev and Uvicorn enabled quick iterations during local development.
- **Use free public data:** NYC.gov public APIs help avoid unnecessary signups and subscription fees while experimenting.
- **RAG for accuracy:** Upload public PDFs (community, shelters, etc.) to Blob Storage and index them with AI Search so the AI gives sourced, reliable answers.
- **Keep RAG flexible:** Azure fields vary, so your extraction logic must adapt; fallback logic is helpful when services aren't fully configured.
- **Learn core Azure tools:** App Services for deployment, Functions for summaries, AI Search for RAG, and Blob Storage for documents.