

Name: David Arazi

# URSP 688Y: Smart Cities and Urban Data Science

## Smart Zoning Assistant for College Park



---

## Table of Content

Introduction and Problem Statement	3
Data Sources and Analytical Methods	3
Results and Functionality	4
Planning Impact and Equity Considerations	6
Limitations and Future Enhancements	7
Conclusion	8

---

## Introduction and Problem Statement

I wanted to explore one of the most persistent and foundational questions in planning: *where can you build?*

In Prince George's County, zoning regulations are complex and frequently updated. Although zoning affects almost every facet of land development, access to zoning information remains difficult for the general public. Residents often have to go through long, complicated PDFs or use advanced GIS platforms that are not user-friendly. This makes it hard for many people to get the information they need and limits how much they can take part in decisions about land use.

The core research question of this project is:

*Can zoning classification and land use suitability be automated and delivered in real time through an accessible web-based tool?*

The Smart Zoning Assistant solves this by combining geographic data and browser-based spatial analysis to provide immediate zoning information to users based on their location. The tool democratizes zoning data which in turn reduces the dependency on manual lookups and increases transparency and awareness around local land regulations.

## Data Sources and Analytical Methods

To answer the research question, I designed and implemented a functional web tool that combines multiple open-source technologies and authoritative datasets. The primary zoning data source is the Prince George's County GIS Open Data Portal, which includes polygon shapefiles and GeoJSON files representing the zoning classifications of every parcel in the county. Each polygon contains a zoning code attribute (e.g., R-R, R-55, M-X-T) that determines permissible land use on that parcel.

I used the Leaflet.js library to build the interactive map interface. Users can view the zoning landscape of the county rendered with OpenStreetMap tiles. The frontend also includes geolocation functionality, powered by the Browser Geolocation API, which

---

allows users to retrieve their exact coordinates upon granting permission. When a user clicks the “*locate me*” button or taps on any point on the map, the location data is passed into Turf.js, an open-source JavaScript geospatial library. Turf performs a point-in-polygon test to see if the selected location falls inside any zoning polygon in the dataset.

The result is presented directly on the web page where it displays the zoning classification associated with the position of the user. All analysis runs in the browser, so there is no need for a backend server. Therefore, the tool is extremely fast, lightweight, and privacy-respecting. The code is modular and documented, and I plan to release it as open-source in the future.



**Source:** Wordpress backend

---

## Results and Functionality

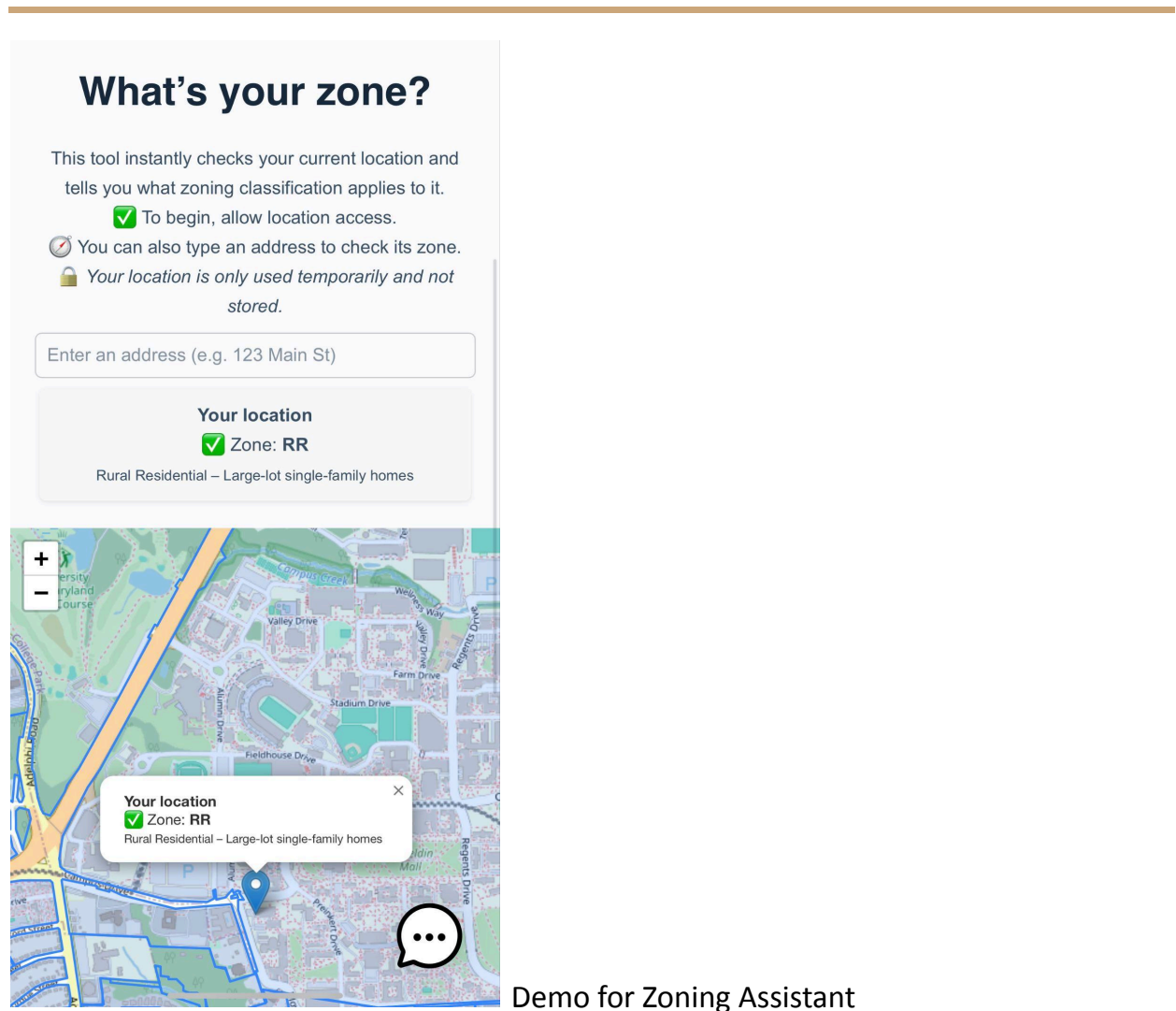
The Smart Zoning Assistant is now fully deployed and accessible to the public at [urban610.com/smartai](https://urban610.com/smartai). It opens with an interactive base map, and once permission is granted, the application immediately zooms to the current location of the user. The zoning class for that area appears as a simple text overlay. Users can also manually explore the map and click on different areas to retrieve zoning designations.

The tool supports:

- Automatic detection of zoning class based on GPS coordinates
- Manual location lookup by clicking on the map
- Fast response time due to in-browser processing
- Mobile-optimized interface for use during field visits

The above features make Zi AI ideal for planners checking zoning during site analysis, residents trying to determine what they can build, and developers evaluating project feasibility. It further reduces the burden on government agencies and improves the user experience significantly.

Screenshots and user flows will be included to demonstrate how the tool behaves during typical use cases, such as identifying a zoning type in College Park, MD or exploring development restrictions in a rural area.



Demo for Zoning Assistant

## Planning Impact and Equity Considerations

One of the most important goals of this project is to increase transparency and support more equitable access to planning data. Zoning often determines who gets to build, what they can build, and where resources are allocated. Yet for many low-income or historically marginalized communities, zoning is a technical barrier that separates them from meaningful engagement in the development process.

By offering real-time, browser-based zoning information, the Smart Zoning Assistant breaks down these barriers. It can help a homeowner understand whether they can build an accessory dwelling unit (ADU), a tenant see what is allowed on their block, or a

---

small developer assess a site without hiring a consultant. Another advantage is that the tool is free, requires no GIS knowledge, and works on any internet-connected device.

In future iterations, I plan to integrate permitted use tables and a zoning code explanation to help users understand the implications of their zoning category. These additions will support more informed decision-making at every level, from residents and business owners to county officials and advocacy groups.

## **Limitations and Future Enhancements**

The current version of the Smart Zoning Assistant is powerful but limited in scope. It only reports the zoning code and does not include full definitions or dynamic updates from the planning department of the county. While the interface is user-friendly, it cannot yet search by parcel ID or address, and it does not incorporate other relevant overlays such as flood zones, protected areas, or slope gradients that would affect development suitability.

The theoretical limitation lies in the fact that zoning, as a legal and regulatory framework, often requires context-specific interpretation. While the tool can simplify spatial analysis, it cannot account for site-specific exceptions, legal overlays, or subjective discretionary reviews that influence actual land use outcomes.

My planned next steps include:

- Integrating a zoning code glossary with permitted uses
- Adding a search function by address or parcel number
- Visualizing other constraints like flood risk or slope
- Creating a simple dashboard showing zoning distribution across regions
- Launching a mobile app version for offline access
- Exploring the use of AI to identify underutilized parcels and suggest potential rezoning areas

These features will allow the Smart Zoning Assistant to transition from a zoning lookup tool to a planning analysis platform.

---

## Conclusion

My passion for practical innovation in urban planning led me to develop the Smart Zoning Assistant to solve a real-world problem. I connected spatial datasets with web technology to create a tool that helps everyday people understand zoning and interact with land use information in a simple, meaningful way.

The project demonstrates how data science can support transparency, efficiency, and equity in planning. With further development, I hope the tool will inspire similar initiatives in other regions and become part of a new wave of accessible, smart planning tools.



---

## Reference

Arazi, D. (2025). *Smart Zoning Assistant* [Web application]. Urban610.

<https://urban610.com/smartai>

Prince George's County Government. (n.d.). *GIS Open Data Portal*.

<https://data.princegeorgescountymd.gov/>

Leaflet. (2024). *Leaflet.js: An open-source JavaScript library for interactive maps*.

<https://leafletjs.com/>

OpenStreetMap contributors. (n.d.). *OpenStreetMap*. <https://www.openstreetmap.org/>

MDN Web Docs contributors. (n.d.). *Geolocation API*. Mozilla Developer Network.

[https://developer.mozilla.org/en-US/docs/Web/API/Geolocation\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API)

Turf. (2024). *Turf.js: Advanced geospatial analysis for browsers and Node.js*. <https://turfjs.org/>