Design and Development of a Reproducible Framework for Interactive Web-based Campus Maps

Aarushi Chakraborty
Student, Dept. of CSBS
Dayananda Sagar College of Engineering
Bangalore, India

ABSTRACT
Large university campuses often lack accessible, building-level digital maps, making navigation difficult for new students, visitors, and staff. The existing solutions include static sign boards which are difficult to read and navigate, Google maps, which does not include all the paths and the irregular building shape, or booklets that are handed out which have similar problems. This paper presents a reproducible framework for developing a web based, interactive campus map using scalable vector graphics (SVG) and web technologies. This proposed pipeline allows institutions to convert their existing campus layout into an interactive map without specialized geographic information system (GIS) tools. This framework consists of five stages, layout digitization, region labelling, data integration, frontend scripting and deployment. A case study on Dayananda Sagar College of Engineering demonstrates the usability, adaptability and low resource requirements of this system.

Keywords
Campus navigation, reproducible framework, SVG, web mapping, usability, smart campus.

1. INTRODUCTION
University campuses are often complex spaces with a large number of buildings, with diverse departments and paths to take. These campuses are frequently visited by large numbers of people, especially during student admission, placement season, symposiums and conferences. Navigating these environments can be a challenge, particularly when existing maps are static, outdated or poorly designed. While online tools such as Google Maps provide broad coverage, they lack building level detail, may not include all the routes within the campus or provide campus specific interactivity, such as departmental information.

With the growing popularity of smart campuses, there is an increasing need for digital, interactive maps with a provision for navigation. Smaller institutions have the need for budget friendly options, a solution that does not require expensive GIS infrastructure, that can be locally developed and maintained. Most existing systems are non-reproducible, and tightly coupled to one campus, or require specialized tools that are difficult to generalize.

To address these limitations, this paper proposes a generalized, reproducible pipeline for developing interactive campus maps using open technologies such as SVG, JavaScript, and HTML5. The framework enables campuses to digitize their campus layouts into scalable vector formats, define transparent clickable regions for irregularly shaped buildings, integrate custom metadata such as departments, facilities, offices, and deploy the system using freely available web hosting platforms. This work presents a reproducible and lightweight framework for developing web-based interactive campus maps. The proposed five-stage open-source pipeline can be adapted to any campus layout and operates entirely on a browser-based technology stack, removing the need for external GIS servers or dependencies. To demonstrate its practicality, the framework is applied to Dayananda Sagar College of Engineering's campus as a case study. The system's modular design further enables scalability and future integration with databases, search features, and indoor navigation systems.

2. RELATED WORK
2.1 Mobile and AR-Based Campus Navigation Systems
Several studies have explored mobile and augmented reality (AR) approaches for campus navigation. Qin et al. proposed a mobile AR-based navigation system that overlays digital directions on real-world camera views [1]. While effective for outdoor orientation, AR-based systems are constrained by camera visibility, sensor calibration, and device performance. Tamhane et al. and Helmi et al. developed AR-assisted navigation systems that combine route computation and user feedback mechanisms for campus wayfinding [2], [3]. These systems demonstrate enhanced user engagement but rely heavily on platform-specific SDKs and continuous camera usage, limiting scalability and reproducibility.

2.2 Platform-Dependent Visualization and Web Mapping Systems
Visualization-centric campus mapping solutions have also been explored. Shelke et al. utilized Tableau to develop an interactive virtual campus map for spatial data exploration [4]. Although visually effective, such systems depend on proprietary platforms, limiting deployment flexibility and increasing maintenance cost. Nordin proposed a web-based AR navigation framework for indoor wayfinding [5]; however, its performance degrades on low-end devices and requires continuous rendering pipelines. Rahman et al. developed a cross-platform campus navigation framework offering broad device compatibility but limited real-time interaction capabilities [10].

2.3 AI-Driven and Algorithmic Navigation Systems
Several works integrate artificial intelligence and algorithmic routing for improved navigation accuracy. Sharma et al. proposed an AI-enhanced navigation system using intelligent routing mechanisms [8], while Liu et al. incorporated collaborative shortest-path algorithms for route optimization [9]. Although these systems improve routing efficiency, they require continuous learning data, centralized computation, and

higher computational resources, making them difficult to reproduce and deploy across institutions with limited infrastructure.

2.4 Research Gap Summary

Existing campus navigation solutions predominantly rely on mobile AR platforms, proprietary visualization tools, or computationally intensive AI models. These approaches often lack reproducibility, depend on closed ecosystems, and are not designed as lightweight web-deployable frameworks. None of the existing works propose a generalized, reproducible, SVG-based web framework evaluated for scalability, usability, and deployment simplicity, which motivates the proposed approach.

3. METHODOLOGY

This section presents the proposed generalized pipeline for developing interactive, web-based campus maps, which would enable any institution to replicate or adapt the framework for their own campus. The proposed workflow consists of five stages, as follows:

- Layout Digitization
- Region Labelling
- Metadata Integration
- Frontend Scripting
- Deployment and Validation

3.1 Layout Digitization

The first stage involves converting the existing campus layout into a vectorized digital format, that includes all the necessary details, accurate geometry and proportions. The process begins by either taking a blueprint, existing static map, or satellite image into Inkscape (or any SVG editor) as a background layer. Each building is traced using the Bezier or polygon tool, and each shape is given a unique identifier such as id= "bld-cse". The output is saved as a scalable vector graphics (SVG) file which preserves the resolution and allows flexible resizing which is crucial for web display. The output is a lightweight base map that can be rendered in any browser.

To ensure spatial accuracy, the digital layout is georeferenced by aligning known reference points (e.g., corner coordinates or landmarks) with the corresponding positions on OpenStreetMap tiles. This step allows the SVG layout to later be overlaid precisely using MapLibre GL JS.

3.2 Region Labelling

Once the layout is digitized and each building or area has a unique identifier, the metadata is stored into a separate JSON or CSV file. The identifiers follow a consistent naming convention and are linked to metadata describing the building name and departments. This structure ensures smooth integration with web-based interactivity. An example is as follows:

```
"bld-cse": {
  "name": "Computer Science Block",
  "department": "CSE Department"
}
```

3.3 Metadata Integration

This stage links the SVG layout elements with the corresponding semantic metadata. A structured JSON schema is used to store building-level information such as block identifiers, department names, functional categories, and descriptive attributes. A dedicated JavaScript module parses the metadata file and dynamically maps the building identifiers to the respective SVG path elements using unique element IDs. Upon interaction, the system retrieves the associated metadata and renders the information within a dynamic side-panel interface. Additional interface functionalities, including attribute-based filtering, color-coding of building categories, and a search-based navigation bar, can be integrated at this stage to enhance usability and navigation efficiency.

3.4 Frontend Scripting

To make the map user friendly, interactivity is added using HTML5, JavaScript, and Tailwind CSS for styling. Features such as hover highlights, information cards, zoom, pan control, search bars, filtering are implemented in this stage. Libraries such as svg-pan-zoom provide smooth navigation for large campus layouts. For example,

```
document.querySelectorAll('path, polygon')
.forEach(region => {
region.addEventListener('click', () => {
const id = region.id; const info = buildingData[id];
displayInfoPanel(
info.name, info.department, info.description
);
});
```

The scripting approach is modular, allowing for future expansion, such as customization of animations, UI elements, or integration with frameworks such as React if required.

3.5 Deployment and Validation

In the final stage, the project is deployed on GitHub pages or any similar web hosting services. The deployment includes the essential files such as index.html, style.css, script.js, campus.svg and metadata.json. Validation can be done by testing across various devices to ensure responsiveness, interactivity and usability through user feedback. Basic functional validation was carried out by verifying interactivity, georeferencing accuracy, and cross-browser compatibility

All digitized layouts are stored as Scalable Vector Graphics (SVG) files, while building metadata are maintained in JSON format. The prototype implementation was developed using standard HTML, CSS, and JavaScript and was tested on Google Chrome and Mozilla Firefox browsers. The framework does not require specialized GIS software and can be reproduced on any standard computing system with internet access and open-source tools.

Table 1. Summary of the five stages

Stages	Input	Tool(s) Used	Output
Layout Digitization	Campus map or blueprint	Inkscape	SVG layout
Region Labelling	SVG paths	Inkscape, text editor	Tagged SVG with IDs
Metadata Integration	Building list	JSON/CSV	Linked metadata
Frontend Scripting	HTML, JS	JavaScript, Tailwind	Interactive map
Deployment	Static site files	GitHub Pages	Online campus map

4. PROTOTYPE IMPLEMENTATION

A prototype of the interactive campus map was developed to validate the feasibility of the proposed pipeline. The current version implements the first three stages- layout digitization, region labelling and basic frontend interactivity, while the metadata integration and deployment are in progress.

This setup validated the core concept of the proposed pipeline demonstrating that even with minimal dependencies, a functional, georeferenced, and interactive mapping system can be achieved. The modularity also ensures that future updates, such as adding new layers or linking building data, can be done without redesigning the existing structure. The prototype further demonstrates that the framework can be progressively expanded into a full-scale campus information system without affecting existing digitized layouts or frontend components.

4.1 Implementation Overview

The prototype was built using Inkscape, HTML5, JavaScript, and Tailwind CSS, with OpenStreetMap (OSM) providing the background layer. The vectorized campus map was georeferenced using QGIS and overlaid on the base map using MapLibre GL JS, allowing accuracy and interactivity.

The main steps included were as follows:

•The campus layout was digitized with the help of a sign board present in the campus, and was made into a SVG file with the help of Inkscape

•The file was then georeferenced with the help of QGIS

•The SVG was overlaid on OpenStreetMap tiles to preserve the geolocation

•Hover and click detection were implemented for each building region using JavaScript

•The buildings were highlighted dynamically to ensure coordinate accuracy and interactivity

Currently, work is in progress on the metadata integration and deployment stages. Metadata for each building, including the name, department, and description, has been structured in a JSON format and is being linked to the corresponding SVG regions using JavaScript. Interactive components such as the information panel and search bar are under development to enable building-specific data display upon user interaction. Deployment is planned on GitHub Pages following an open-source workflow. Upon completion, the system will support category-based filtering, search functionality, and department-level information retrieval for better campus navigation. A simple navigation system to make it easier to find each department and navigate through the large campus will also be implemented, with the help of the paths as shown in Fig. 3.

The overall implementation reflects the feasibility of the proposed low-cost, open-source campus mapping solution. While still in progress, the prototype effectively demonstrates how a modular and transparent workflow can replace traditional closed-source GIS solutions for institutional mapping.

Fig 1: Vectorized campus layout

Fig. 2: Overlay of campus SVG on OpenStreetMap base using MapLibre

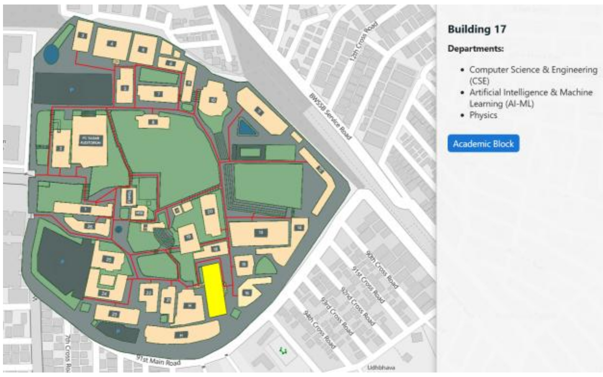

Fig 3: Interactive prototype showing highlighted building information and paths for navigation

5. EVALUATION OF THE PROPOSED PIPELINE

Existing campus navigation systems often depend on proprietary mapping or GIS platforms that require paid licenses, specialized training, and complex setup. The high cost of software subscriptions, API usage, and data management tools makes such solutions difficult to use for many institutions. Moreover, their limited flexibility and dependence prevent customization and independent deployment. Most of the existing systems are also based on augmented reality (AR) or mobile-based navigation, emphasizing user orientation and real-time wayfinding. While effective in immersive visualization, these systems often rely on proprietary platforms, lack reproducibility, and are challenging to adapt for new institutions without redeveloping core components.

In contrast, the proposed framework emphasizes reproducibility, customization and easy maintenance. By using open-source tools such as Inkscape for layout digitization, MapLibre for georeferenced visualization, and GitHub Pages for deployment, the pipeline ensures ease of use and reusability. The use of SVG and JSON integration helps combine visual and semantic data, enabling straightforward maintenance and scalability across different campus environments.

From an evaluation perspective, the proposed framework shows clear potential in terms of cost-efficiency, deployment simplicity, and scalability. Even at its current stage of implementation, the modular design allows for easy integration of additional buildings or metadata without altering the base system. Preliminary testing on the Dayananda Sagar College of Engineering (DSCE) campus layout indicated that the SVG-based approach can accurately represent complex structures and maintain lightweight rendering performance. The use of OpenStreetMap for background visualization demonstrated effective georeferencing; while hosting through GitHub Pages highlighted the ease of deployment using open-source tools. These early results suggest that the framework can offer faster, more flexible, and lower-cost alternatives to conventional GIS or AR-based systems once fully implemented.

In addition to qualitative advantages, quantitative performance evaluation further supports the effectiveness of the proposed framework. As shown in Table 2 and Fig. 4, the proposed framework achieves a significantly lower page load time (281 ms) compared to OpenStreetMap (627 ms) and Google Maps (5670 ms). This demonstrates the lightweight nature of the SVG-based representation and its suitability for deployment on resource-constrained environments. The reduced load time, combined with interactive building-level overlays, indicates that the framework can provide faster access to campus information while maintaining structural completeness and customization flexibility.

Table 2. Performance comparison

Metric	Google Maps	OSM	Proposed Framework
Page load time (ms)	3130	627	281
Click latency (ms)	N/A	9	14
Building-level clarity	Low	Low	High
Reproducibility	Low	Low	High

Table 3. Summary of the existing systems

Feature	Proposed System	Mobile AR Systems	*GIS/* Tableau
Platform	Web (SVG + MapLibre)	Android / AR mobile	Web dashboards
Customization	Full via CSS + JS	SDK-restricted	Template-based
Interactivity	Clickable SVG, pop-ups	Camera overlay AR	Basic map UI
Data Source	OpenStreetMap + SVG	Local / cloud data	Spatial databases
Reproducibility	High (open-source)	Low (SDK-dependent)	Medium
Cost	Free / Open-source	~US $100–1,000 / yr (SDKs)	~US $3,000 + / yr (license)
Offline Use	Yes	No	No

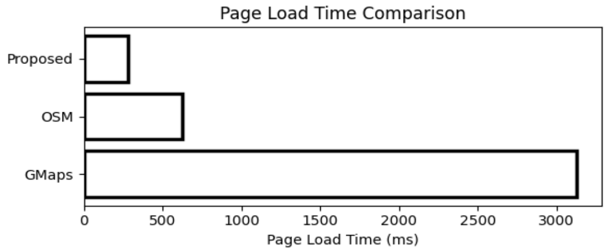

Fig 4: Page load comparison between Google Maps, OpenStreetMap (OSM), and the proposed campus mapping framework.

6. FUTURE WORK

The developed framework establishes a reproducible foundation for interactive campus mapping, integrating open-source tools and modular stages that can be independently improved. While the proposed framework shows efficiency and scalability for campus mapping, there are several enhancements that can be added to enhance its capabilities:

•Automatic Layout Extraction: With the help of computer vision or machine learning, the satellite or blueprint images can be converted to SVG paths, reducing the manual effort

•Database Driven Search and Dynamic Updates: Databases such as firebase, MySQL or MongoDB can be used for real time updates, advanced search and analytics

•Indoor Navigation and Floor Mapping: Indoor positioning via Wi-Fi, SVG for each floor and path guidance using algorithms such as Dijkstra or A* can be added

•AI and Natural Language Interfaces: AI assistants or small LLMs for semantic search, conversational queries and voice interfaces can be implemented

•Accessibility and Localization: Multiple language support, high contrast or dark mode UI options can be added for accessibility

•Smart Campus Integration: IoT systems can be connected for live tracking, occupancy visualization and real time alerts, making into a smart campus dashboard

•Cross-Domain and Industry Applications: The framework can be extended beyond educational campuses to support hospitals, industrial complexes, corporate facilities, enabling interactive navigation, facility management, and emergency response visualization in diverse environments.

7. CONCLUSION

This paper presented a reproducible, open-source framework for creating interactive, web-based campus maps using SVG overlays and lightweight web technologies. The proposed five stage pipeline comprising of layout digitization, region labelling, metadata integration, frontend scripting and deployment helps convert conventional maps into a dynamic browser-based solution. The case study on Dayananda Sagar College of Engineering Campus Map helped validate the feasibility and adaptability of the framework.

Compared to traditional GIS systems, this framework offers significant advantages in accessibility, cost-efficiency, and ease of implementation. It empowers educational institutions to independently design, maintain, and extend digital navigation platforms without specialized software or expertise. Furthermore, it can be expanded to various domains such as hospitals, industrial parks or disaster management systems.

The evaluation against existing systems highlights the framework's practicality and flexibility. Unlike AR or GIS-based approaches, it maintains interactivity and detail without dependence on existing platforms or costly infrastructure, making it particularly suited for academic and institutional deployments.

In conclusion, this work bridges the gap between research and real-world deployment, proving that lightweight web technologies can deliver interactive mapping solutions suited to smart campus environments. Future enhancements can focus on automating layout extraction, integrating AI driven interfaces, and connecting the framework to smart campus infrastructures for real-time, data driven interactivity.

8. REFERENCES

[1] Y.-X. Qin, J.-M. Liu, H. Qiu, B. Yan, and C.-X. Jin, "Campus navigation system based on mobile augmented reality," *Proc. 2013 6th Int. Conf. Intelligent Networks and Intelligent Systems (ICINIS)*, IEEE, 2014.

[2] M. Tamhane, P. Sarjekar, C. Gupta, and V. M. Kharche, "Campus navigation using augmented reality," *Int. J. Scientific & Engineering Research*, 2018.

[3] R. A. A. H. Helmi, H. A.-P. Ravichandran, A. Jamal, and M. N. Mohammed, "Design and development of indoor campus navigation application," *Proc. IEEE 10th Conf. Systems, Process & Control (ICSPC)*, IEEE, 2022.

[4] V. Shelke, J. S. Shrimal, P. Tiwari, and P. Pawar, "Virtual Ally: Campus navigation system using Tableau," *Int. J. Computer Engineering in Research Trends*, 2022.

[5] J. Nordin, "A web-based campus navigation system with mobile augmented reality," *J. Phys.: Conf. Ser.*, 2021.

[6] "Location navigation assistance for educational institutions using augmented reality,"Available online: *ResearchGate*, 2020.

[7] "Indoor navigation using augmented reality,"Available online: *ResearchGate*, 2021.

[8] "Artificial intelligence based navigation system," *IJSART*, 2022.

[9] "Campus mobile navigation system based on shortest-path algorithm and users' collaborations,"Available online: *ResearchGate*, 2015.

[10] "Cross-platform development of a university campus navigation system," *Int. J. Innovative Technology and Exploring Engineering*, 2020.