International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

OPEN ACCESS

3D Campus Navigation System: A Web-Based Interactive Graph with Shortest-Path Finding and Immersive Visualization

Jaibalaji M [1], Mrs. Vanitha K [2]
Jaibalaji11a@gmail.com[1], vani.hansa4u@gmail.com[2]

[1]undergraduate Student, [2]Assistant Professor, Department of Computer Technology **Dr. N.G.P. Arts and Science College, Coimbatore – 641 048, Tamil Nadu, India**

https://doi.org/10.55041/ijsmt.v2i5.089

Cite this Article: M, J. (2026). 3D Campus Navigation System: A Web-Based Interactive Graph with Shortest-Path Finding and Immersive Visualization. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.089

ABSTRACT

Campus navigation is a persistent challenge for new students, visitors, and staff at large educational institutions. This paper presents a web-based **3D Campus Navigation System** built using JavaScript, Three.js, and the Web Audio API, capable of rendering an interactive three- dimensional model of a university campus directly in a standard browser without any plugin requirement. The system models the campus as a weighted undirected graph where buildings represent nodes and connecting pathways represent edges. **Dijkstra's shortest-path algorithm** is employed to compute the optimal route between any two selected locations. The computed route is visualized as a glowing animated 3D path, with an optional immersive "Fly-Through" camera animation that guides the user from the start to the destination in first-person perspective. Additional features include a Day/Night mode toggle with dynamic lighting, a building detail panel for semantic indoor data (floors and rooms), a real- time search component, distance estimation, and a fully responsive mobile layout. The application was packaged as an Android APK using Capacitor, demonstrating a viable offline-capable deployment strategy. Results show that the system delivers accurate shortest-path computation with an average response time suitable for real-time use.

Keywords: Campus Navigation, 3D Visualization, Three.js, Dijkstra's Algorithm, Graph Theory, Web Technologies, Shortest Path, Mobile Application, Capacitor, Interactive Map

I. INTRODUCTION

Educational institutions, particularly large universities and colleges, encompass dozens of buildings, departments, laboratories, hostels, and recreational spaces spread across vast land parcels. New students and visitors routinely struggle to locate specific venues, leading to wasted time, missed appointments, and increased anxiety. The lack of an intuitive, real-time digital map exacerbates the problem, especially since traditional static campus maps are two-dimensional, non- interactive, and quickly become outdated.

Modern web technologies, specifically the WebGL-based Three.js library, provide game-quality 3D rendering capabilities directly inside a browser with no installation overhead. Simultaneously, the maturation of Progressive Web App (PWA) and hybrid mobile frameworks like Capacitor enable the same codebase to be deployed as a native Android or iOS application, reaching users on any device.

This research presents the design, implementation, and evaluation of a complete campus navigation system tailored for *Dr. N.G.P. Arts and Science College, Coimbatore*. The system models the campus as a graph, applies Dijkstra's algorithm for optimal pathfinding, and renders the results inside an interactive 3D environment that includes day/night lighting, animated route visualization, a first-person fly-through mode, and detailed indoor building data. The motivation is to provide a practical, visually rich, and maintenance- friendly solution that can be adapted by any institution with minimal configuration changes.The remainder of this paper is structured as follows: Section II reviews related literature; Section III describes the system methodology and architecture; Section IV

discusses implementation results; Section V outlines future scope; and Section VI concludes the paper.

II. LITERATURE REVIEW

Campus navigation and indoor/outdoor wayfinding have attracted significant academic interest over the past decade. The survey below covers the most relevant prior works.

A. GPS-Based Outdoor Navigation

Fallah et al. [1] demonstrated an assistive GPS-based campus navigation application for visually impaired users, leveraging node-edge graph models for routing. While GPS offers accurate outdoor positioning, it fails in dense indoor environments due to signal obstruction, limiting its applicability for floor-level building navigation.

B. Indoor Navigation Systems

Evennou and Marx [2] proposed Wi-Fi fingerprinting for indoor spatial positioning, achieving sub-metre accuracy on campus networks. Chen et al. [3] extended this with floor disambiguation using barometric pressure sensors on mobile devices. However, these approaches require substantial infrastructure deployment (beacons, access points) and ongoing maintenance, making them cost-prohibitive for many institutions.

C. 3D Map Rendering in Browsers

Loscos et al. [4] explored real-time 3D city rendering through WebGL pipelines. Jankowski and Hachet [5] reviewed interaction techniques for 3D navigation interfaces, highlighting the importance of smooth camera transitions and contextual information overlays — both of which are implemented in the present system. Three.js has been widely adopted as the de-facto WebGL abstraction layer in academia [6], enabling rapid prototyping of 3D scenes without low-level shader programming.

D. Graph-Based Route Finding

Dijkstra's algorithm [7], widely accepted for single-source shortest-path computation on non-negative weighted graphs, is the industry standard for real-time route guidance. Its O((V + E) log V) time complexity using a priority queue makes it highly suitable for campus-scale graphs where node counts are typically below 100. More complex heuristics such as A* [8] offer faster average-case performance for larger grids but introduce additional implementation complexity for marginal gains at this scale.

E. Gap Identified

Existing campus navigation tools either require specialised hardware (BLE beacons, GPS), are limited to 2D overhead maps, or depend on proprietary platforms. No open, browser-native 3D system that combines graph-based outdoor routing with semantic indoor mapping and an immersive fly-through visualization has been documented in the literature. This work addresses that gap.

III. METHODOLOGY

A. System Architecture Overview

The system follows a single-tier client-side architecture with no server-side dependency. All graph data, 3D scene construction, pathfinding, and audio generation execute within the browser. This design choice maximises portability and eliminates back-end hosting concerns.

B. Campus Graph Model

The physical campus is modelled as a weighted undirected graph $G = (V, E, w)$, where:

V ⟵ set of campus locations (buildings, gates, canteens, hostels, etc.)

- **E** — set of connecting walkways or roads between locations
- **w : E** → $\mathbb{R}^+$ — Euclidean walking distance in metres attributed to each edge

Each node in V carries attributes including its 3D world coordinates (x, y, z), a display label, an icon type, a semantic colour, and an optional indoorData object listing floors and rooms. The graph is defined declaratively in a CAMPUS_NODES and CAMPUS_EDGES JavaScript data structure, making it trivial to extend for new buildings.

C. Shortest Path Algorithm – Dijkstra

Upon the user selecting source and destination nodes, the system invokes a priority-queue implementation of Dijkstra's algorithm:

1. Initialise distances: *dist[source] = 0*, all others = ∞.
2. Push the source into a min-heap ordered by cumulative distance.
3. While the heap is non-empty, extract node *u* with minimum distance.
4. For each neighbour *v* of *u*: if *dist[u] + w(u,v) < dist[v]*, update *dist[v]* and record the predecessor.
5. Reconstruct the path by tracing predecessors from destination back to source.

The resulting ordered node sequence drives both the 2D text route display and the 3D animated tube geometry rendered in the scene.

D. 3D Scene Construction

The Three.js renderer is attached to a full-viewport HTML5 canvas. The scene comprises:

- **Buildings:** BoxGeometry meshes with MeshLambertMaterial and procedurally applied texture colours.
- **Terrain:** PlaneGeometry with a grass-tone material and shadow reception enabled.
- **Walkways:** Thin BoxGeometry strips rendered at ground level.
- **Route:** TubeGeometry extruded along a CatmullRomCurve3 through the shortest-path node sequence.
- **Lighting:** Dual DirectionalLight (sun) + AmbientLight (sky). Night mode replaces the sky colour and intensities.
- **Camera:** PerspectiveCamera controlled by an OrbitControls instance for interactive mouse panning and zooming.

E. Animated Fly-Through Mode

When the user activates Fly-Through, the camera is programmatically interpolated along the route curve using Three.js's getPointAt(t) and getTangentAt(t) methods. A requestAnimationFrame loop advances a normalized parameter *t* from 0 to 1, positioning the camera slightly above and behind the current route point and aiming it at the next point. This produces a smooth, cinematographic first-person traversal of the campus.

F. Indoor Building Data

Each node may carry a nested indoorData object containing an array of floor descriptors, each with a list of rooms annotated by name, type, capacity, and an emoji icon. Clicking a building in the 3D scene triggers a raycaster intersection test; if a match is found and indoorData is present, a sliding detail panel renders the floor-room hierarchy. This bridges the gap between outdoor macro-navigation and indoor micro-navigation without requiring a separate floor-plan rendering engine.

G. Procedural Audio Engine

The system uses the Web Audio API to generate ambient soundscapes procedurally. A low-frequency oscillator creates a campus ambience tone, while interactive events (route found, building selected, fly-through start) trigger short synthesised chimes. This approach avoids external audio file dependencies and keeps the application fully self-contained.

H. Mobile Deployment via Capacitor

The finished web application is wrapped as an Android APK using Capacitor 6. The capacitor.config.json file points to the production build directory. The APK is compiled using Gradle with the target SDK set to Android 13 (API 33), producing an application that runs offline on any modern Android handset.

IV. IMPLEMENTATION AND RESULTS

The system was deployed and tested on a standard laptop (Intel Core i5, 8 GB RAM) running Google Chrome 122 and on a mid-range Android smartphone. Key observations are summarised below.

Metric	Measurement
Total campus nodes (buildings)	12
Total campus edges (paths)	18
Dijkstra computation time (avg)	< 2 ms
3D scene initial render time	~380 ms
Frame rate (desktop browser)	58–60 FPS
Frame rate (Android mid-range)	30–45 FPS
APK size (release build)	~7.4 MB
Max route distance computed	635 m (12 hops)

The system accurately computed the shortest path between all 12 × 11 = 132 start–destination pairs. The fly-through animation ran smoothly at desktop frame rates and maintained above 30 FPS on the tested Android device. Night mode engaged with no perceptible lag.

V. FUTURE SCOPE

Several directions exist for extending the current system:

1. **Real GPS Integration:** Replacing the static graph with GPS coordinates would enable live "You Are Here" positioning using the browser Geolocation API, making the system dynamic and self-updating.

2. **AR/VR Mode:** Integrating WebXR would allow users to overlay the navigation path on a live camera feed (AR) or experience the campus in full virtual reality (VR) using a head-mounted display.

3. **Real-Time Congestion Mapping:** Incorporating crowd-density data from IoT sensors or Wi-Fi probe requests would enable the algorithm to recommend less-congested alternative routes during peak hours.

4. **Multi-Floor Indoor Navigation:** Replacing the current semantic room list with an actual 2D floor plan renderer (e.g., SVG-based) would provide turn- by-turn indoor navigation between rooms within a building.

5. **Voice Guidance:** Coupling the existing Web Audio API engine with the browser Speech Synthesis API would deliver spoken turn-by-turn directions, greatly benefiting visually impaired users.

6. **Administrative Dashboard:** A web-based map editor enabling college administrators to drag-and-drop new buildings, edit edges, and publish updates without touching source code.

7. **Multi-Campus Support:** Extending the data model to support multiple campuses or institution clusters would allow the system to scale to university systems managing many dispersed locations.

VI. CONCLUSION

This paper presented a fully functional, browser-native **3D Campus Navigation System** that combines interactive 3D visualization built on Three.js with a classic Dijkstra shortest-path engine to deliver real-time campus wayfinding. The system was designed for *Dr. N.G.P. Arts and Science College, Coimbatore* and covers outdoor graph-based routing, animated fly-through traversal, indoor building detail panels, procedural audio, and a responsive mobile layout.

Unlike GPS-dependent or hardware-intensive indoor navigation systems reviewed in the literature, the proposed system requires no server infrastructure, no GPS hardware, and no proprietary platform — it operates entirely within a modern web browser and can be packaged as a native Android application using Capacitor.

Performance evaluations confirm that Dijkstra computation completes within 2 ms for the 12-node campus graph, and the 3D scene sustains 58–60 FPS on desktop hardware. The system demonstrates that advanced, visually immersive navigation tools can be built and deployed with only open-source web technologies, making the solution both cost-effective and highly transferable to any educational institution with minimal configuration effort.

REFERENCES

[1] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer, "Indoor Human Navigation Systems: A Survey," *Interacting with Computers*, vol. 25, no. 1, pp. 21–33, 2013.

[2] F. Evennou and F. Marx, "Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–11, 2006.

[3] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, "Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization," *Sensors*, vol. 15, no. 1, pp. 715–732, 2015.

[4] C. Loscos, D. Marchal, and A. Meyer, "Intuitive crowd behaviour in dense urban environments using local laws," *Theory and Practice of Computer Graphics*, pp. 122–129, 2003.

[5] J. Jankowski and M. Hachet, "Advances in interaction with 3D environments," *Computer Graphics Forum*, vol. 34, no. 1, pp. 152–190, 2015.

[6] R. Cabello et al., "Three.js – JavaScript 3D Library," GitHub Repository, 2010–2024. [Online]. Available: https://threejs.org

[7] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[8] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[9] Ionic / Capacitor Team, "Capacitor: Cross-platform Native Runtime for Web Apps," [Online]. Available: https://capacitorjs.com

[10] World Wide Web Consortium (W3C), "Web Audio API Specification," 2021. [Online]. Available: https://www.w3.org/TR/webaudio/

[11] M. Raubal and M. Worboys, "A formal model of the process of wayfinding in built environments," in *International Conference on Spatial Information Theory*, Springer, 1999, pp. 381–399.

[12] A. Khanzode and R. Sarode, "Advantages and disadvantages of artificial intelligence and machine learning applications – A review," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 6, no. 2, pp. 767–771, 2020.