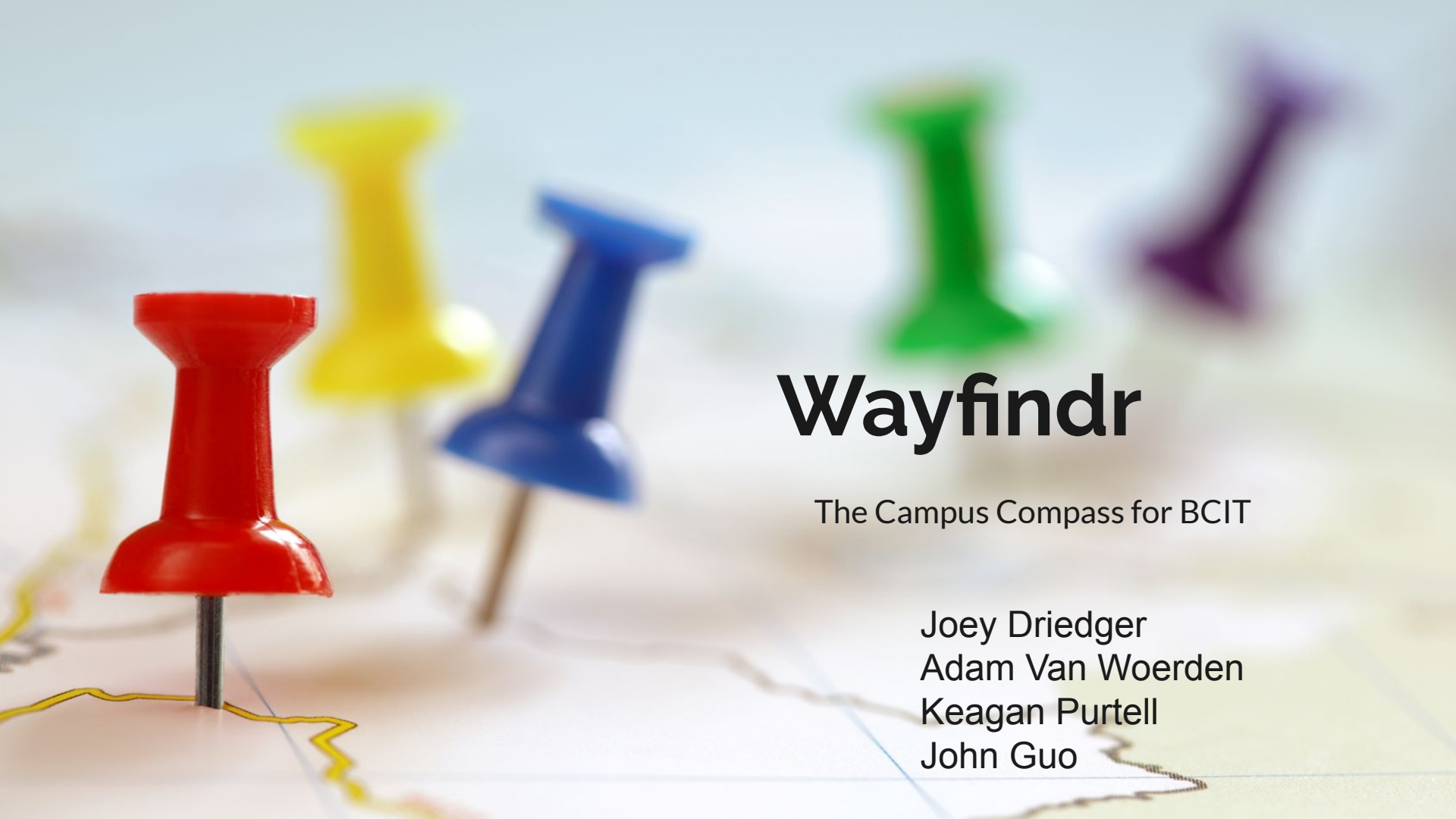


Wayfindr

The Campus Compass for BCIT

Joey Driedger
Adam Van Woerden
Keagan Purtell
John Guo

Introduction

Problem Summary

- Students and visitors struggle to navigate campus efficiently.
- No unified database of campus nodes, paths, or personalized shortcuts.
- Users need a way to browse, store, and manage important navigation points.

What our system solves

- Allows users to search for buildings and room numbers on campus, showing them the fastest way to their destination.
- Allows users save locations on the site for fast repeatable directions to their favorite locations.

Proposed Solution

System Overview

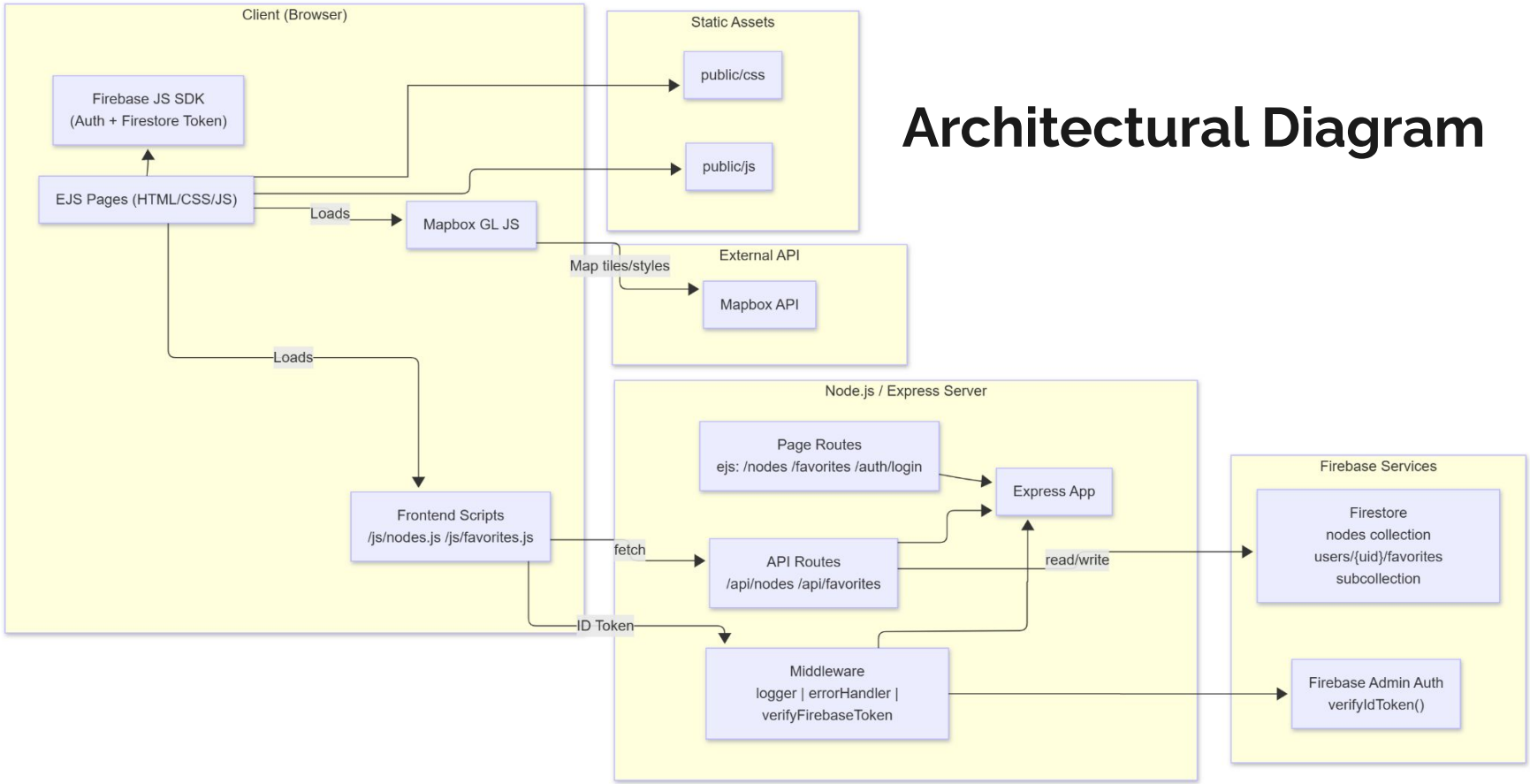
- Web backend and UI for managing:
- Map Nodes with shortest route navigation linking them together
- Inside building navigation.
- Favorites linked to signed-in users
- Uses Firebase Auth (front-end) + Firebase Admin (server) for secure user identification.

Key Features

- Able to expand our node network by adding new nodes
- Load existing nodes
- Add/remove favorite locations for each user with labels and sorting

Core Functionality Demonstration

Architectural Diagram



Search & Locate Buildings

- Instantly finds buildings by name, code, or keyword
- Highlights and centers the map on the selected building
- Shows building details, floors, and services in a quick popup
- Integrates with navigation for “Set Start” or “Navigate Here”
- Makes campus maps faster and easier to use

Turn-by-Turn Navigation

- Step-by-step routes between any two campus locations
- Smart start point selection (user location, map center, or chosen room)
- Clear visual guidance with markers, route line, and “From/To” panel
- Room-to-room navigation directly from floor-plan shapes
- Smooth integration with building search and map interactions

Secure User Authentication

- Firebase Auth handles all login and signup on the frontend
- Backend uses Firebase Admin to verify ID tokens
- Only authenticated users can access protected routes
- Ensures user-specific, secure interactions across the app

Favorite Nodes (Per-User Data)

- Each user has their own favorites subcollection
- Users can add or remove favorites independently
- Favorites persist across sessions and devices
- Guarantees isolation so one user’s favorites never affect another user’s favorites

Technical Challenges

Data Availability Challenges

Challenge:

- Building data for BCIT buildings and rooms was very difficult to find. No known or public information on the coordinates of the BCIT building's rooms themselves.

Reflection:

- Received a GeoJSON file of the BCIT building coordinates (not the most recent data) from BCIT Geomatics.
- Took PDF files of the floorplans from BCIT website and preprocessed them into SVG files which I used the building GeoJSON file with a custom Python script and a custom EJS page to create an approximation of the coordinates for the rooms of the buildings

Technical Challenges

Firebase Auth and Express Sessions

Challenge:

- Getting the frontend Firebase ID token sent to the server and verified using Firebase Admin.

Reflection:

- Added proper token extraction from the Authorization header.
- Implemented checkSession middleware to verify ID tokens reliably.
- Updated frontend to consistently send the token with every request, stabilizing authentication flow.

Technical Challenges

User authentication-authorization

Firestore Structure Decisions

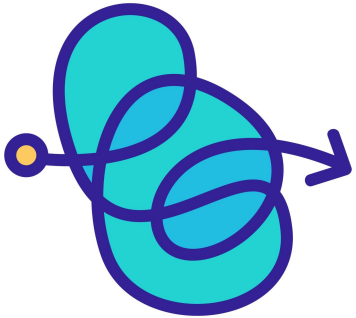
Challenge:

- Designing the database structure, including using a top-level nodes collection and adding favoriteNodes as a subcollection under each user.

Reflection:

- Standardized structure after testing multiple models.
- Used subcollections to reduce read costs and isolate per-user data.
- Confirmed queries remained simple (/users/{uid}/favoriteNodes) and scalable.

Technical Challenges



Node-Based Navigation

Challenge:

- Image Model Navigation does not work at a campus level so we need another solution. Looking at other map solutions like Google Maps we can use nodes!

Reflection:

- Nodes provide a clean solution for navigation
- Nodes are computationally heavy when iterating through all of them

Technical Challenges

Route Structure Cleanup

Challenge:

- Reducing duplicate routes and separating page routes (/nodes) from data routes (/api/nodes).

Reflection:

- Consolidated all REST API logic under /api/... to avoid collisions with EJS-rendered pages.
- Updated frontend fetch calls accordingly.
- Improved maintainability, removed redundant router functions, and clarified server structure.

Technical Challenges

New Technologies

Challenge:

- Required to use unfamiliar tools and technologies
- Steep learning curve and complex features
- Troubleshooting issues took extra time and research

Reflection:

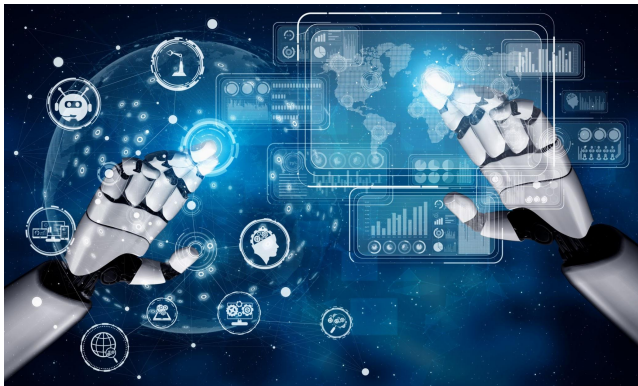
- Difficult at first but we stayed persistent
- Improved our ability to learn new technologies
- Gained confidence through practice and problem-solving

Compromises

- Due to the lack of mapping data available and the complexities of creating the missing data, we were unable to map all the rooms of the whole campus of BCIT, so we focused on a smaller part of the campus.
- Accuracy was underwhelming as the translation of interior navigation to the real coordinates was confusing and time consuming
- With most of our time going into the Navigation part of the app, we had to put aside working on the nice-to-have features such as Accessible Routes, Landmark & Service Highlights, and Event Mapping to ensure that the Navigation part works properly.

Lessons Learned

- How to create new GeoJSON data from existing GeoJSON data
- How to use Mapbox GL API
- Integrating Firebase Auth with a Node.js backend
- Designing scalable Firestore database schemas
- Debugging ES modules & ensuring correct script load order
- The importance of communication
- To properly merge and integrate code earlier
- Not everything can be properly automated



Improvements / Future Work

If I had more time:

- Add schedule page for favorite node that could be setup as different classes of the student throughout the week.
- Add "shared favorites" between group, and BCIT events visible by all users
- Full single-page application (SPA) frontend instead of EJS pages
- Update the maps with BCIT's newer buildings (e.g. SW7 - Tall Timber Student Housing)
- Map the entire interior of the BCIT campus
- Improve accuracy of interior navigation

Q & A on Discord