

CSIS 4490 - CityScope (Sahan Nonis 300389470)

Progress Report Number: 2

Date	Hours	Description of Work Done
Sep 20, 2025	2	Initial brainstorming of project ideas, reviewed proposal requirements, and finalized CityScope as the main project concept.
Sep 23, 2025	3	Drafted project proposal document with sections on introduction, methodology, and timeline. Added references.
Sep 27, 2025	2	Created project GitHub repository and organized folder structure (Implementation, Documents, etc.).
Sep 29, 2025	3	Set up NestJS API service inside the monorepo (apps/api). Installed Prisma, TypeScript, and supporting dependencies. Initialized Prisma schema with SQLite for simplicity.
Oct 1, 2025	2.5	Designed database schema (Neighborhood, MetricSnapshot) in Prisma. Created .env for database configuration and tested connection.
Oct 3, 2025	3	Created CSV datasets for neighborhoods, rental listings, transit points, and shopping malls. Prepared initial ETL (Extract-Transform-Load) pipeline.
Oct 5, 2025	3	Wrote prisma/seed.ts to parse CSV data, compute aggregated metrics (average rent, transit count, mall count), normalize values, and generate a composite livability score.
Oct 7, 2025	2.5	Debugged Prisma authentication and environment setup issues. Reconfigured .env for SQLite and re-generated Prisma client. Successfully seeded database with computed metrics.
Oct 9, 2025	3	Implemented API endpoints (/neighborhoods, /neighborhoods/:id/summary, /compare) in NestJS controller. Verified JSON responses in browser and ensured data flows correctly.
Oct 10, 2025	3	Final validation of MVP backend: tested full ETL → database → API cycle. Prepared Git commit and documented Phase 1 completion.

Oct 12, 2025	3	Initialized Next.js frontend app under apps/web inside monorepo. Configured TailwindCSS, TypeScript, and React libraries. Verified connection to backend API endpoints.
Oct 14, 2025	3	Built initial UI with neighborhood table and comparison chart. Integrated /neighborhoods API to dynamically fetch and display metrics (average rent, transit count, mall count).
Oct 16, 2025	3	Implemented Mapbox GL map visualization to show neighborhood boundaries and markers. Displayed data from backend on the map.
Oct 18, 2025	2.5	Added support for transit stops and shopping mall markers (loaded from CSV → database → API). Debugged display alignment between geographic coordinates and Mapbox rendering.
Oct 20, 2025	3	Integrated Overpass (OSM) JSON export for malls and TransLink GTFS stops.txt for transit data. Converted formats into CSV for ingestion and linked to Prisma ETL seed script.
Oct 22, 2025	3	Enhanced charts to reflect live metrics from database. Created counter widgets for “Transit Stops” and “Malls” per neighborhood, ensuring values match CSV imports.
Oct 24, 2025	2.5	Completed Phase 2 deliverables: API + frontend integration, working map with neighborhood overlays, dynamic analytics.

Description of Work Done

Since the last report, the focus has been on **Phase 2 of CityScope**: integrating the frontend interface with the backend API and visualizing neighborhood data in a way that aligns with the project proposal.

1. Frontend Setup (Next.js + TailwindCSS)

- Created apps/web application inside the monorepo.
- Configured TailwindCSS for styling, TypeScript for type safety, and linked the frontend to the backend API (/neighborhoods, /compare).
- Established reusable components (Map view, Chart view, Counter widgets).

2. Data Visualization

- Implemented a **map view** using Mapbox GL to show neighborhoods and related amenities.
- Added markers for **transit stops** and **malls** sourced from CSV datasets.
- Integrated charts to visualize average rent, transit, and amenities scores across

neighborhoods.

3. External Data Integration

- Used **TransLink GTFS (stops.txt)** to represent bus/skytrain stops.
- Queried **OpenStreetMap Overpass API** for shopping mall POIs in Vancouver; converted JSON exports into CSV for ingestion.
- Updated Prisma seed script to process these datasets and recalculate livability scores.

4. Analytics & User Interface

- Added **neighborhood comparison charts** displaying side-by-side metrics (rent, transit, malls).
- Developed **summary counters** that dynamically reflect total transit stops and malls imported from datasets.
- Designed UI to align with project proposal: allowing users to evaluate livability of neighborhoods (Burnaby, Downtown Vancouver, Kitsilano).

5. Challenges & Solutions

- **OSM JSON format** required conversion before use; solved by writing transformation into CSV.
- Transit stops dataset was large; optimized to a filtered subset for MVP presentation.
- Ensured consistent values across database, API responses, and frontend counters.

6. Outcome of Phase 2

- Working MVP frontend connected to backend API.
- Map with neighborhoods, transit, and mall data displayed.
- Dynamic analytics for livability scoring successfully presented.
- System now demonstrates **proof-of-concept** for real estate & community exploration as described in the proposal.

Repo Check-In of Implementation Completed

New files/folders since Report 2:

- **apps/web/**
 - **pages/**
 - `index.tsx` – landing page with map and analytics.
 - `compare.tsx` – neighborhood comparison view.
 - **components/**
 - `MapView.tsx` – Mapbox integration for neighborhoods + markers.
 - `ChartView.tsx` – chart.js/Recharts visualization for metrics.
 - `Counters.tsx` – displays transit and mall counts.

- `package.json` – frontend dependencies.
- `tailwind.config.js` – Tailwind configuration.
- **data/**
 - `malls.csv` – converted from OSM JSON.
 - `stops.txt` – GTFS transit dataset.
- **apps/api/prisma/seed.ts** – updated ETL pipeline to ingest new datasets.
- **apps/api/src/**
 - Updated controller/services to expose mall + transit data alongside neighborhoods.