CSIS 4490 - CityScope (Sahan Nonis 300389470)

Progress Report Number: 4

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.
Oct 26, 2025	2.5	Began transition from monorepo architecture to a simpler Python-based analytics setup. Removed NestJS and Next.js dependencies to focus on data processing and visualization.
Oct 28, 2025	3	Set up new project folder structure (`data/`, `scripts/`, `frontend/`). Installed Python dependencies (osmnx, geopandas, folium, streamlit).
Oct 30, 2025	3	Imported CMHC 2024 BC rental dataset and explored structure. Cleaned and prepared columns for analysis (city, rent, property type, bedroom count).
Nov 1, 2025	2.5	Integrated OpenStreetMap (OSM) API using osmnx to extract amenities (transit, schools, malls). Defined scoring logic based on amenity density per km².
Nov 3, 2025	3	Implemented Python scripts for automated data processing (`01_build_osm_data.py`, `02_process_cmhc_data.py`). Debugged geometry NaN errors in OSM data.

Nov 5, 2025	3	Configured GitHub repository for the new version of CityScope. Committed refactored structure and initial working code. Verified environment setup.
Nov 6, 2025	2.5	Built Streamlit dashboard layout with interactive map and filtering controls for amenities. Connected CMHC and OSM data for preliminary visualizations.
Nov 8, 2025	2	Finalized scoring algorithm combining CMHC rent metrics and amenity density. Tested comparison feature between BC neighbourhoods (e.g., Downtown vs Kitsilano).

Description of Work Done

During this reporting period, the project transitioned from a complex monorepo structure to a simplified, modular analytics-oriented setup. The focus shifted fully to **data visualization, analytics, and neighbourhood scoring**, minimizing the backend complexity while improving data integration and usability.

Key development efforts included:

- **Repository Restructuring:**
  The original monorepo, which combined backend and frontend modules, was split into a lean structure optimized for a Python-based analytics dashboard.
  - Introduced clear folders for `scripts`, `data`, and `frontend`.
  - Simplified setup and environment management.
  - Successfully pushed to Git for version control and collaboration.

- **Data Integration:**
  - Integrated **CMHC rental data (Excel dataset)** for the Province of British Columbia.
  - Established connection to **OpenStreetMap (OSM)** APIs to extract points of interest (transit stops, schools, malls, etc.) for neighbourhood scoring.
  - Defined scoring criteria using **area (km²)** to weight amenities per neighbourhood density.

- **Analytics Focus:**
  Shifted the focus from raw data engineering to user-facing insights:
  - Comparative analytics between **neighbourhoods (e.g., Downtown vs Kitsilano)**.
  - Developed scoring logic for amenities, accessibility, and livability.
  - Outlined visual components for filtering and comparison.

- **Error Handling & Debugging:**
  Initial OSM data extraction encountered invalid geometry errors, which were documented and are being resolved by refining bounding polygons and excluding

incomplete geometry entries.

Technical Stack

- Language: Python 3.12
- Core Libraries: `osmnx`, `geopandas`, `pandas`, `folium`, `plotly`, `streamlit`
- Data Sources:
    - CMHC 2024 Rental Market Report (BC dataset)
    - OpenStreetMap (OSM) via Overpass API

- Version Control: GitHub
- Visualization Framework: Streamlit (for dashboard UI with filters and map layers)

Features Implemented

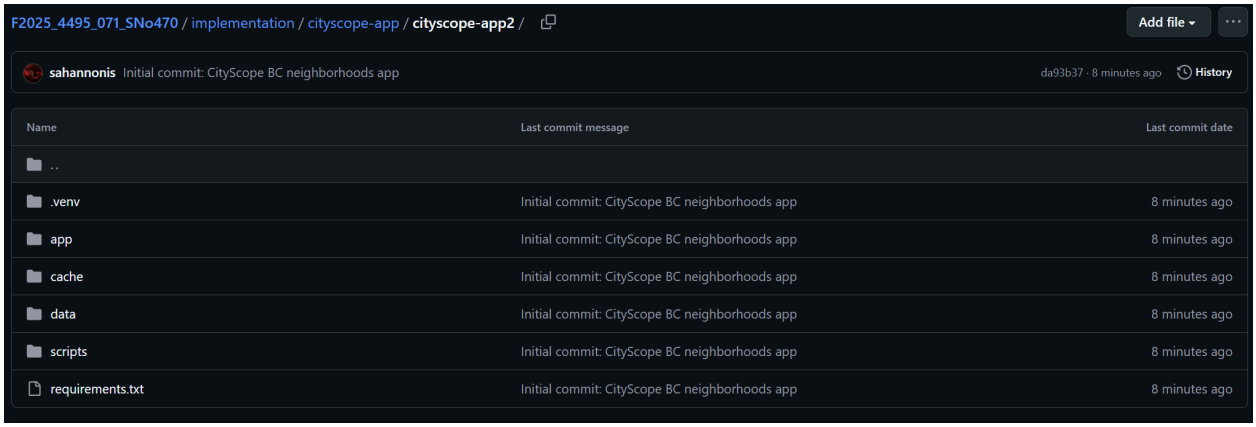
- **Neighbourhood Map Layer:**
    - Interactive map showing amenities (transit, schools, malls, etc.) with filter controls.
- **Amenity Scoring System:**
    - Weighted score = `(amenities count / area_km2) × category weight`
    - Overall livability score calculated by aggregating weighted categories.
- **Comparison Dashboard:**
    - Side-by-side neighbourhood comparison panel (e.g., Downtown vs Kitsilano).
    - Visuals: radar charts, bar graphs, and density maps.
- **Data Pipeline Scripts:**
    - `01_build_osm_data.py`: Downloads and structures OSM data for selected regions.
    - `02_process_cmhc_data.py`: Cleans and aggregates rental price data.
    - `03_generate_scores.py`: Calculates livability indices.

Challenges Encountered

- **Geometry Validation Errors:**
  Certain polygons in OSM queries returned invalid area values (NaN). This issue is being addressed by adding geometry cleaning functions (`buffer(0)` and `dropna` on area fields).
- **Monorepo Complexity:**
  Initial design involved multiple interlinked modules that became difficult to maintain. Migrating to a simplified repo improved focus and debugging efficiency.
- **Data Alignment:**
  Normalizing OSM data with CMHC neighbourhood boundaries required manual geospatial reconciliation to ensure consistent coordinate reference systems (CRS).

Repo Check-In of Implementation Completed

New files/folders since Report 3:



Created the new python program inside cityscope-app/cityscope-app2/