1. Project Title
UrbanBird

2. Project Summary
UrbanBird is a community-driven web platform that captures, aggregates, and visualises bird-sighting data in densely populated cities. By merging crowdsourced observations with authoritative ornithology datasets, the site reveals how avian species adapt to—and sometimes struggle in—urban habitats.

The application helps casual birders discover local "hot spots," assists researchers tracking biodiversity trends, and gives city planners evidence for bird-friendly green-space design.

3. Description
UrbanBird is a user driven web application with the aim of monitoring and viewing bird activity in urban environments. It combines confirmed bird sightings from users and publicly available databases like eBird. This application allows users to store, visualize, and explore bird information through an interactive interface. The application further allows users to view taxonomic information of each bird by cross referencing information from the AvesData dataset with eBird and user submitted sightings.

This application addresses the lack of centralized bird sighting information of birds in urban environments. While birdwatchers often collect observations and publish them in public datasets, the lack of centralized systems limits insight into how bird populations interact with urban landscapes. UrbanBird solves this by enabling users to log sightings with details like species, time, location, and, potentially, photos, while also offering tools to visualize bird sighting patterns and read taxonomic information for each sighting.

4. Creative Component
UrbanBird will implement a sighting-map overlay with an information on demand feature. All logged sighting locations will be visible on the map. If the user wishes to see more detailed information on a specific datapoint - they will be able to press on it and the user will be able to see all the logged data from the database (e.g., species code, timestamp, latitude/longitude, taxonomy information, etc.).

Another feature that would satisfy the creative component requirement we intend to implement is as follows, users would be able to filter the data visualized based on species, location, or taxonomic information. This gives users instant feedback on the filters they specified. Implementing this feature would, essentially, connect the UI state with SQL query commands and visualize them in real time.

5. Usefulness

UrbanBird is a useful web application that centralizes bird sighting data in urban environments, helping birdwatchers, researchers, and city planners monitor bird activity more effectively. Users can log sightings with details like species, time, and location, explore an interactive map with clustering of sightings, and filter results in real time by species or taxonomy. While similar platforms like eBird and iNaturalist exist, they either lack a specific focus on urban environments or do not offer advanced visualization features. UrbanBird sets itself apart by combining user-submitted and public data, applying clustering algorithms to identify bird hot zones, and offering real-time filtering tied directly to the map view.

6. Realness

UrbanBird relies on four open, production-grade datasets that together deliver expert sightings, crowd-sourced photos, a taxonomic backbone, and urban geospatial context. Each paragraph below states where the data comes from, its file format, cardinality (rows) and degree (columns), the main information captured, and typical size, exactly what the Stage 1 rubric requests.

1. **eBird Basic Dataset (2024) – Cornell Lab of Ornithology**

A compressed, tab-separated archive containing the world's largest bird-observation collection: ≈ 1.5 billion rows × 48 columns. Key fields are species code, observation date-time, latitude/longitude, checklist ID, and count. We will ingest a city-specific slice (e.g., Chicago 2015–2025) for storage efficiency.

2. **AvesData Clements Cross-walk (2023) – McTavish Lab**

A lightweight CSV mapping > 11 000 species codes to scientific and common names, family, order, and Open Tree taxonomy IDs. Cardinality ≈ 11 k rows; degree ≈ 18 columns. This file anchors our taxonomy look-ups and species pages.

3. **iNaturalist Research-grade Export (2025) – via GBIF/API**

Delivered as a Darwin-Core text file inside a ZIP. Global cardinality exceeds 250 million rows with ≈ 30 columns (taxon ID, lat/lon, media URL, observation time, licence flag). A region-filtered extract keeps the working set below 200 MB for development.

4. **OpenStreetMap City Extracts – Geofabrik**

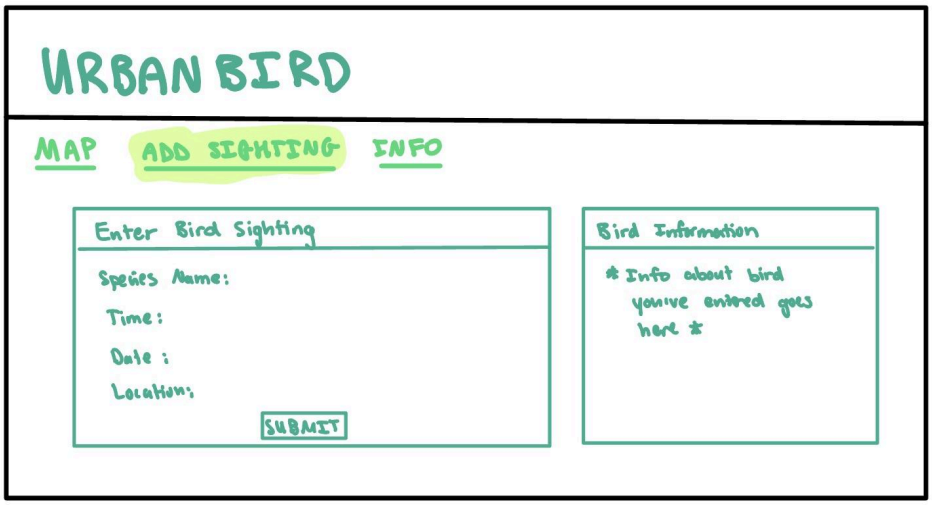
Available either as raw .osm.pbf files or pre-split shapefiles. A typical metro pack is 30–50 MB and contains thousands of polygons/lines (parks, land-use, roads; ≈ 10–30 columns of tags). These layers add park boundaries and other urban features for spatial queries and map styling.

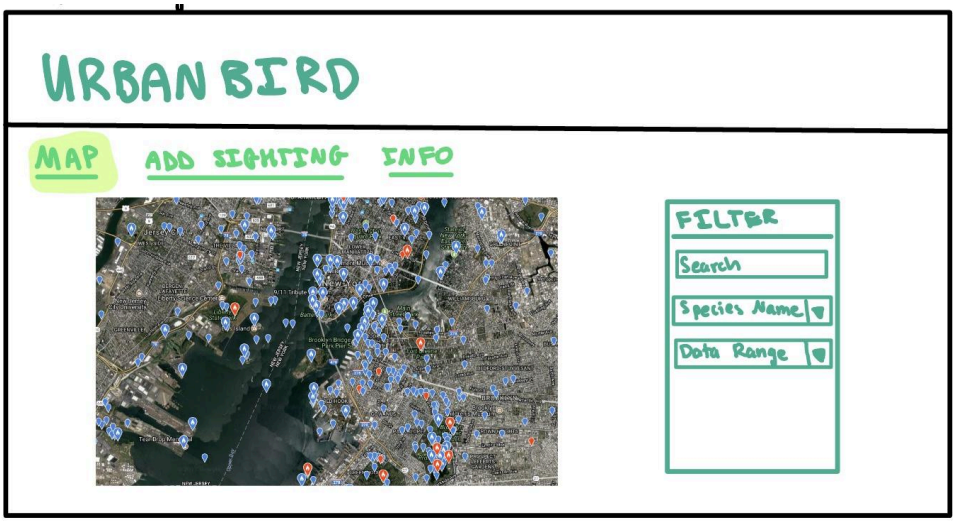
Collectively, these sources meet the realness criterion by supplying multiple independent datasets in open formats (TSV, CSV, Darwin-Core TXT, Shapefile/PBF), each with ample rows (high cardinality) and rich attribute sets (high degree). They enable UrbanBird's planned clustering, filtering, and species-insight features while keeping us fully compliant with open-data licences.

7. Description of Functionality

UrbanBird is a website that allows users to log sightings, explore bird data, and analyze clustering patterns in interactive maps and visualizations. Going into specifics, when a user logs bird sightings, they would be able to put in species name, time and date, and location which would be stored in a database. Users would be able to read more about that bird with the data pulled from the database. On another page, a user is also able to filter bird sightings in terms of species name, date range, and location as well as search for birds when looking at the map of bird sightings and click on the point to see what kind of bird was there. Lastly, users would be able to edit and delete their own sightings.

7.1 Low Fidelity Mockup





7.2 Project Work Distribution

Frontend:

Roberts: Implement map UI

Viki: Implement Add sighting form

Furkan: Implement filters for map and bird information display on Add Sighting page

Sahil: Implement information page and search bar

Backend:

Roberts: Design database schema and help with writing complex queries

Viki: Write scripts to implement datasets into database and help with writing complex queries

Furkan: Implement stored procedures and help with writing queries

Sahil: Implement triggers and help with writing queries