Squidward Crusaders: Linda Zheng, Claire Song, Benjamin Rudinski, Tanzeem Hasan
SoftDev
P04: Makers Makin' It, Act II -- The Seequel
2025-03-27
Time Spent: 6 hrs
TARGET SHIP DATE: {2025-04-16}

Project overview:
We will be finding the relationship between traffic congestion and the concentration of 4 air pollutants (O3, CO, SO2, and NO2) by comparing data from 2 datasets. Users will be able to pick an air pollutant upon logging in. Then, they will see maps of its concentration for each year from 2000 to 2011 alongside maps of traffic congestion over that same time period to see how changes in congestion affect the concentration of the air pollutant that they chose.

- Air quality data: https://www.kaggle.com/datasets/guslovesmath/us-pollution-data-200-to-2022
- Congestion data: https://www.bts.gov/content/annual-roadway-congestion-index
- Geojson (geospatial data for US counties): https://gist.github.com/sdwfrost/d1c73f91dd9d175998ed166eb216994a?short_path=bd97547

Program Components and Connections:
Frontend Components:

1. Jinja Templates: Updated as new data is requested by Python
    a. map.html: used for the pages that show maps of the air quality/congestion in different states/regions in the US
    b. login.html: lets users log in to existing accounts
    c. register.html: lets users create new accounts
    d. home.html: the page that users are directed to upon logging in; shows options of maps to click on (maps of O3, CO, SO2, and NO2)
2. D3.js: Allows users to visualize our data by generating choropleth maps
    a. Generate one map for each pollutant
    b. Each state is a different color depending on the concentration of the pollutant
    c. Allow users to see how the map colors change throughout the years
3. Tailwind CSS: For structuring the layout of our site (explained in further detail below)

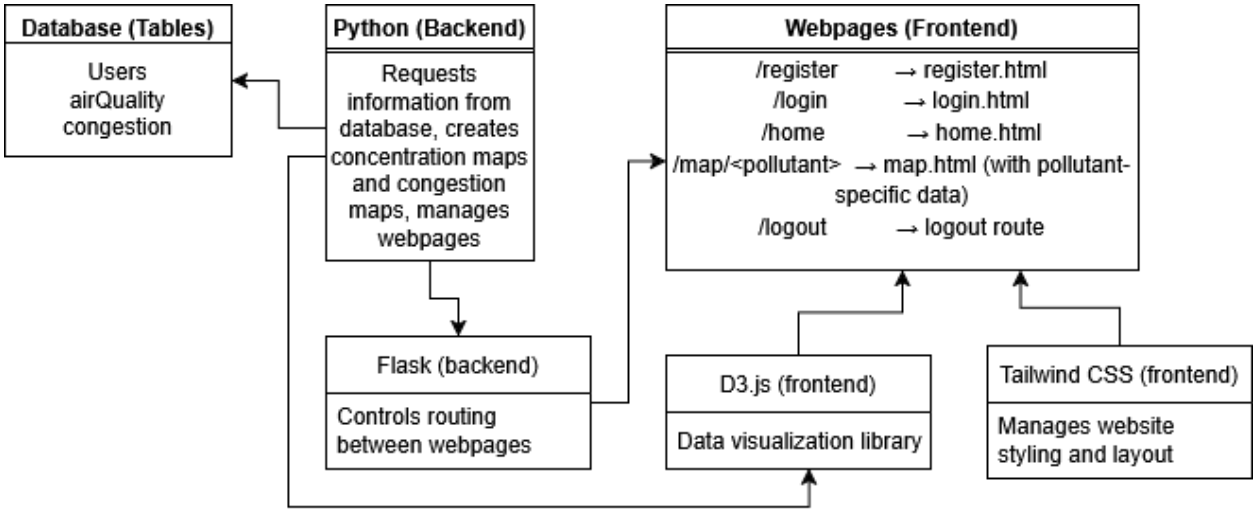
Backend Components:

4. SQLite3: Stores user info and data from our datasets
    a. users: stores all the data related to the users' accounts
    b. airQuality: stores the levels of the air pollutants, the year the data was collected, and the region it was collected from

c. congestion: stores the Roadway Congestion Index, the year the data was collected, and the region it was collected from

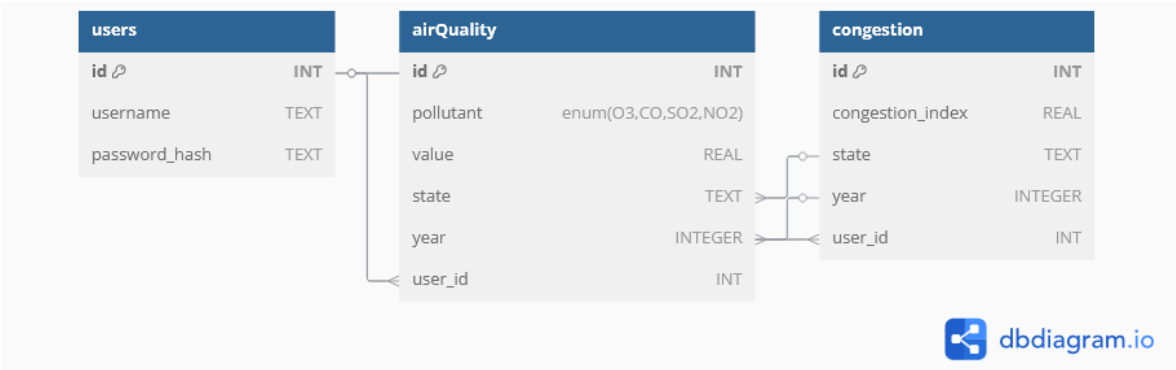
5. Flask app: Retrieves data from databases, allows it to be displayed on the front end

Component Map:



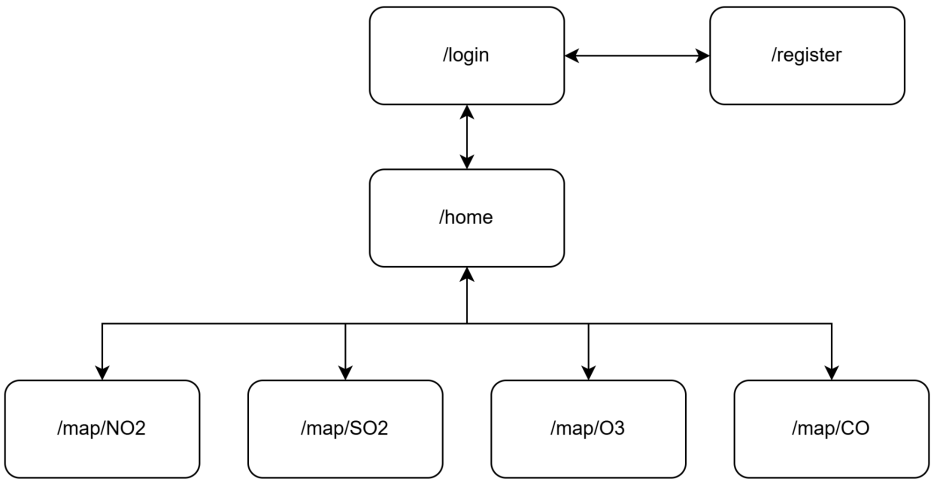
Database Organization: (SQLITE3)

- **Users**
  - id (INT, Primary Key)
  - username (TEXT)
  - password_hash (TEXT)
- **airQuality**
  - id (INT, Primary Key)
  - pollutant (TEXT: 'O3', 'CO', 'SO2', 'NO2')
  - value (REAL)
  - state (TEXT)
  - year (INT)
  - userID (INT)
- **congestion**
  - id (INT, Primary Key)
  - congestion_index (REAL)
  - state (TEXT)
  - userID (INT)
  - year (INT)



Site Map:

/register → register.html
/login → login.html
/home → home.html
/map/<pollutant> → map.html (with pollutant-specific data)
/logout → logout route



Front End Framework (FEF):

Tailwind will be used to:

- Create a responsive, mobile-friendly layout.
- Ensure consistency in fonts, colors, spacing, and layout.
- Use utility classes for grid and flex-based layouts.
- Design intuitive buttons and dropdowns for pollutant selection and map navigation.
- Animate transitions between years on map view with smooth effects.

APIs:

No APIs will be needed. Our data will come from the 2 datasets listed above.

Data Visualization Library:

We will use D3.js because it will let us create more complex graphs such as choropleths to show the different levels of air pollution in different states. It will give us more flexibility when making our graphs and let us control a lot more aspects of our chart compared to the other libraries, allowing the graphs to be catered to our preferences.

Tasks:

Team Member	Task
Claire	HTML & CSS
Linda	D3.js
Tanzeem	Database
Ben	Flask app