Wine Data Lake Project
Executive Summary

For more information about this product, visit: https://winedatalake.com/info

Build Status

Project is live at https://winedatalake.com.

Known Bugs

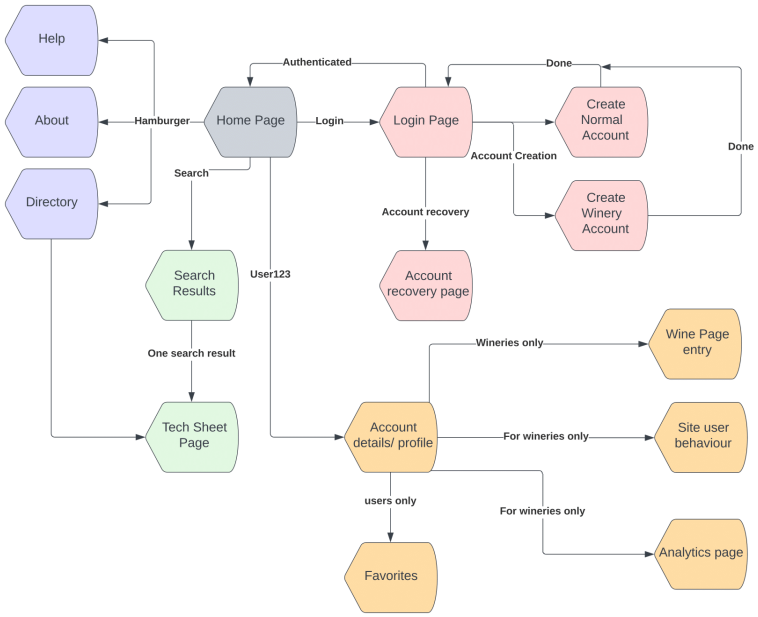
- Search results page loads continuously
- User gets logged out moving between pages
- Home page scales unevenly
- Web Server crashes when SQL errors occur
- Directory does load correctly

Code Style

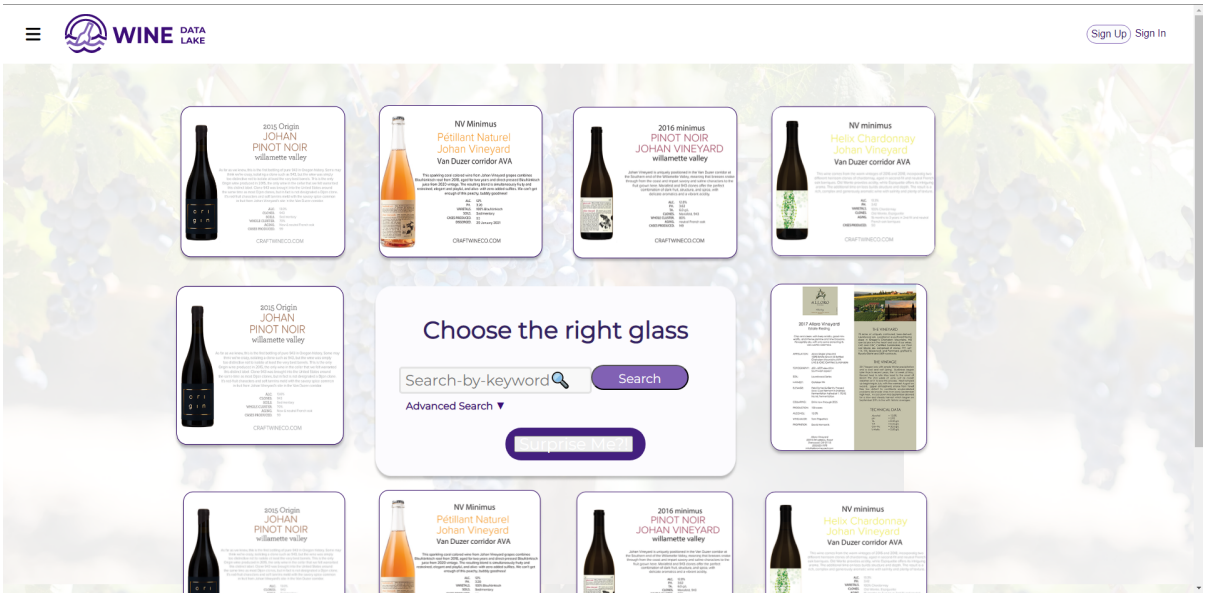
Web server is written in Javascript, with routing handled via express callbacks. Routes for both the webpages and api were written into their own packages. Web pages were developed using handlebars as a templating engine, written in HTML with CSS for styling and Javascript for any functionality. The relational database was written in PostgreSQL, with all database calls from the server to the database handled via SQL api calls that were then parsed in Javascript via a callback system.

Screenshots

Website Road Map



Homepage / Search



Tech / Framework Used

- NPM
- Express
- Handlebars
- Bcrypt
- GCP
- PostgreSQL

Features

- Relational database hosted remotely on GCP

- Web-based interface for viewing data hosted on the database, that allows user to search and filter from the database

- User account management system, allowing users to register with the website and “favorite” wines and wine qualities, additionally allowing users to view wines/attributes that they have favorited

- Meta-data collection of user interactions, including recording user view, favorite dates, and physical location

- Mathematical analysis of user metadata to determine which wines/attributes are most popular among users

- Publicly accessible API, allowing third-party services to query wine data from the database

- Publicly available cloud hosting of wine tech sheets, that users are able to read and download

- Winery account management system, where winery owners can view metrics about their wine releases

- The ability for wineries to upload their own data to the database, via webpage as well as API

- A system to recommend new wines to users based on their history and meta-data on the website

Installations

PgAdmin (for PostgreSQL): https://www.pgadmin.org/download/
Visual Studio Code: https://code.visualstudio.com/download

API References

widedatalake.com/api

/wines?limit=[?]&offset=[?]

- Returns list of wines and stored wine information, specify limit of returned wines, and offset starting point of returned list

/winePopularity?id=[?]

- Returns popularity information for a wine with specified id

/favoriteScores

- Gets favorite scores for wines in the database, order by score descending

How To Use

- To view official page: https://winedatalake.com/
- WDL 2023 Repo: https://github.com/bowersch/Wine_Data_Lake
- Local:
    - After cloning the repo:
        - For database: in pgAdmin (with full permissions)
            - Create an empty database
            - Right click on the database name, select Query Tool
                - Add code from wdl_tables.sql and run the query to add the tables
                - Add code from wdl_data.sql and run the query to add some data

- In a terminal (at the repo location), run:
  npm install
  This will download any dependencies from package.json.
- In main.js, comment out the secrets.js lines and update the database authorization information with the new database information (host, database, user, and password)
- Two ways to run the code:
  - npm start
  - npm test
    - This is for using the nodemon package

Credits

- Jim Cupples - Project Partner
- Austin Chayka - Lead Backend Developer
- Chelsea Bowers - Full Stack Developer
- Sam - Back end Developer
- John Perez Almejo - Lead Front End Developer
- Chris Styles - Additional Outside Help
- Aaron Thompson - Additional Outside Help