State Finder Oral Report 3

Alyiah Proctor, Emily Crabtree, Anne Nguyen, Darlyn Mendez

Database Subsystem

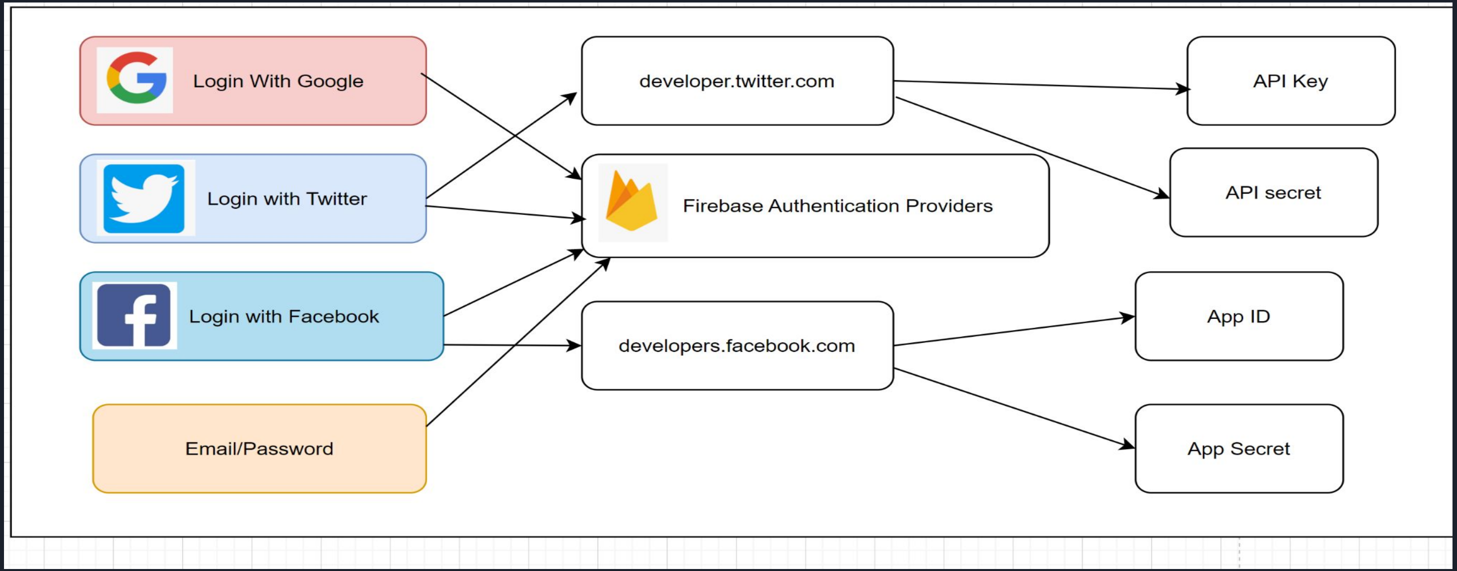
- The Initial Design and Model of the Database includes an ER Diagram and a Schema Diagram that would help us restructure the csv files that we have into tables that we can use later on.
- The Design Choices that we made is to separate two of the data files into several entities to reduce confusion when we reference them during implementation.
- We used diagrams.net and its UML shapes to create the ER diagram and the schema diagram.
- Data dictionary includes attributes from these entities and their data types.
- The changes from the initial model include the addition of the state_descriptions entity and the metro_area_descriptions table.
- The pros of the refined design is that the user is going to have closer view at each state.

Database Subsystem - Coding and User Training

- We used an Azure database.
- Approach - Utilized primary keys, foreign keys, check, numeric, varchar
- To solve an foreign key constraint errors did some vlookups in excel to identify any values that are different from the table that is referencing.
- Language - SQL Server
- The database has been tested by connecting it to the web app.
- Issues with re authentication when running the web app.

Login Subsystem

Diagram showing what we provide to firebase console from the developer sites.



Login Subsystem

- Used firebase docs as a reference for implementation.
- Included service provider icons in the UI.
- We initially were just focusing on google authentication, but we wanted users to have other options as well.
- Coding - Used Javascript to implement the login subsystem.
- Firebase provided some javascript sdks when we created the project.
- Had to make sure to include the the link to our firebase app to connect the project on developers.facebook.com to firebase.
- Tested it on localhost, so it should work once we deploy it and are able to give permission to that domain on the firebase console.

Map Subsystem

Implementation

- Test page is loaded and info is sent from index html to the test function in Routes.py
- Info is used to select correct table and column from the database
- Dataframe is sent to calculate difference function and an array with residual incomes is returned
- Plotly matches residual income array to a geojson shapefile with Metro Areas
    - Done with ID of metro area, state.
- Map is created as a fig object and returned to test.html

Map Subsystem

Changes

- Green, yellow, red to just green and red
    - Color change in choropleth not fast enough for three colors
- Hover feature no longer implemented
    - Decided it was its own subsystem and map
- Separate functions for populating map
    - Initially planned for a function to populate the map and calculate the residual income
    - Now the map population is just the function of the page itself and the residual income is calculated as its own function

State Infographic Subsystem

Coding Process

- Changes: The selected state info page changed from a detailed view of the state consisting of a brief description and any additional graphs to a map of the metro areas of the state.
- Consists of two functions: us_map() and state_map()
- Maps are created using the choropleth map feature in Plotly Graph Objects.
- Ability to view information by hovering is also done in Plotly.
- In state_map(), the outline of the metro areas is generated using a geojson file from the US Census Bureau.
- Language: Python

State Infographic Subsystem

How to Use

- Navigate to state infographic page.
- The user will be able to view a US choropleth map based on states' population size.
- Information about states is displayed when hovering over them.
- The user can select a state from a drop down menu and they will be navigated to another page that will display the state map with similar functionality as the US map.

Front end design

Refinements:

- Merged info collection page and index page
- Merged about page and resources page
- Changed “3+ Children” to “3 Children”
- Fonts, font size, color palette, other design refinements

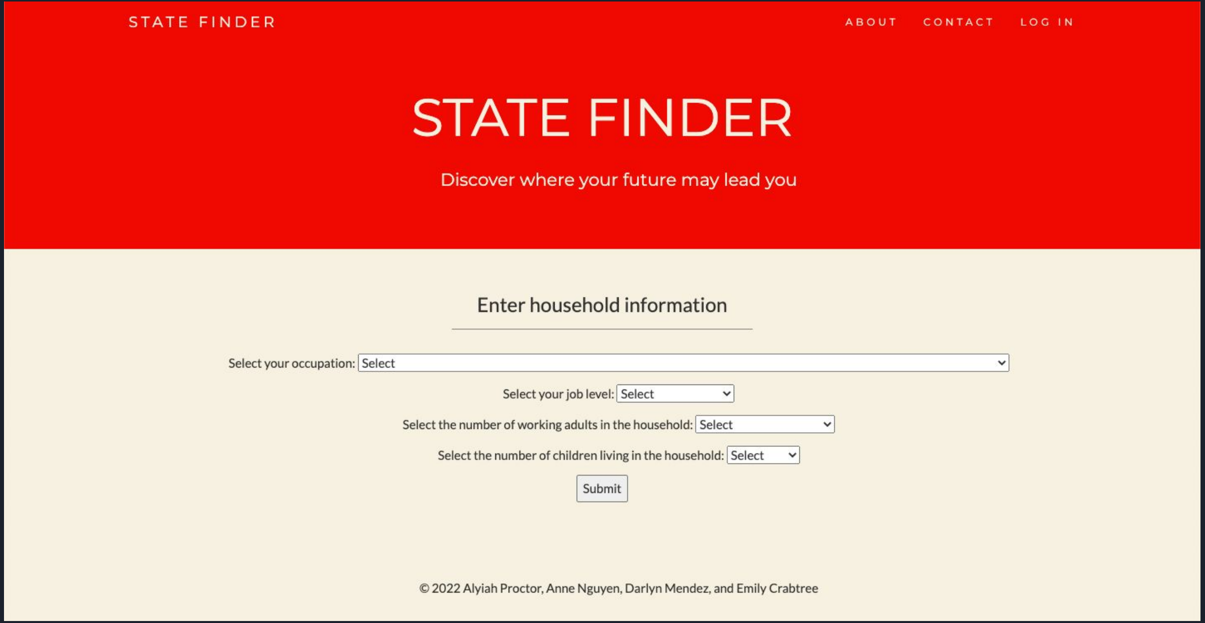
Coding:

- HTML, CSS, Javascript, and Bootstrap
- OOP- Javascript, page scrolling and contact page

Front end design

User Manual:

- Navbar links:
  - About, Contact, Log In
- User is prompted to enter information
  - Occupation
  - Job level
  - # of working adults
  - # of children
- Submit button
  - test.html
  - Choropleth map
  - Metro areas and excess wages
  - Link to other map with state specific information



STATE FINDER

Discover where your future may lead you

Enter household information

Select your occupation: Select

Select your job level: Select

Select the number of working adults in the household: Select

Select the number of children living in the household: Select

© 2022 Alyiah Proctor, Anne Nguyen, Darlyn Mendez, and Emily Crabtree