Alexander Brown, Ruijie Rao, Ahmed Alsalim

DSCI551 Spring 2022

Project Progress Report

**Labor Statistics Dashboard for Data-Informed Career Path Decisions (Midterm Progress Report)**

**Progress in Meeting Goals**

Below is our initial plan used for carrying out the tasks in our project. We will divide each task into categories: completed, in progress/changed, not completed. We will also add a few additional notes to tasks that were significantly changed.

**Legend:**

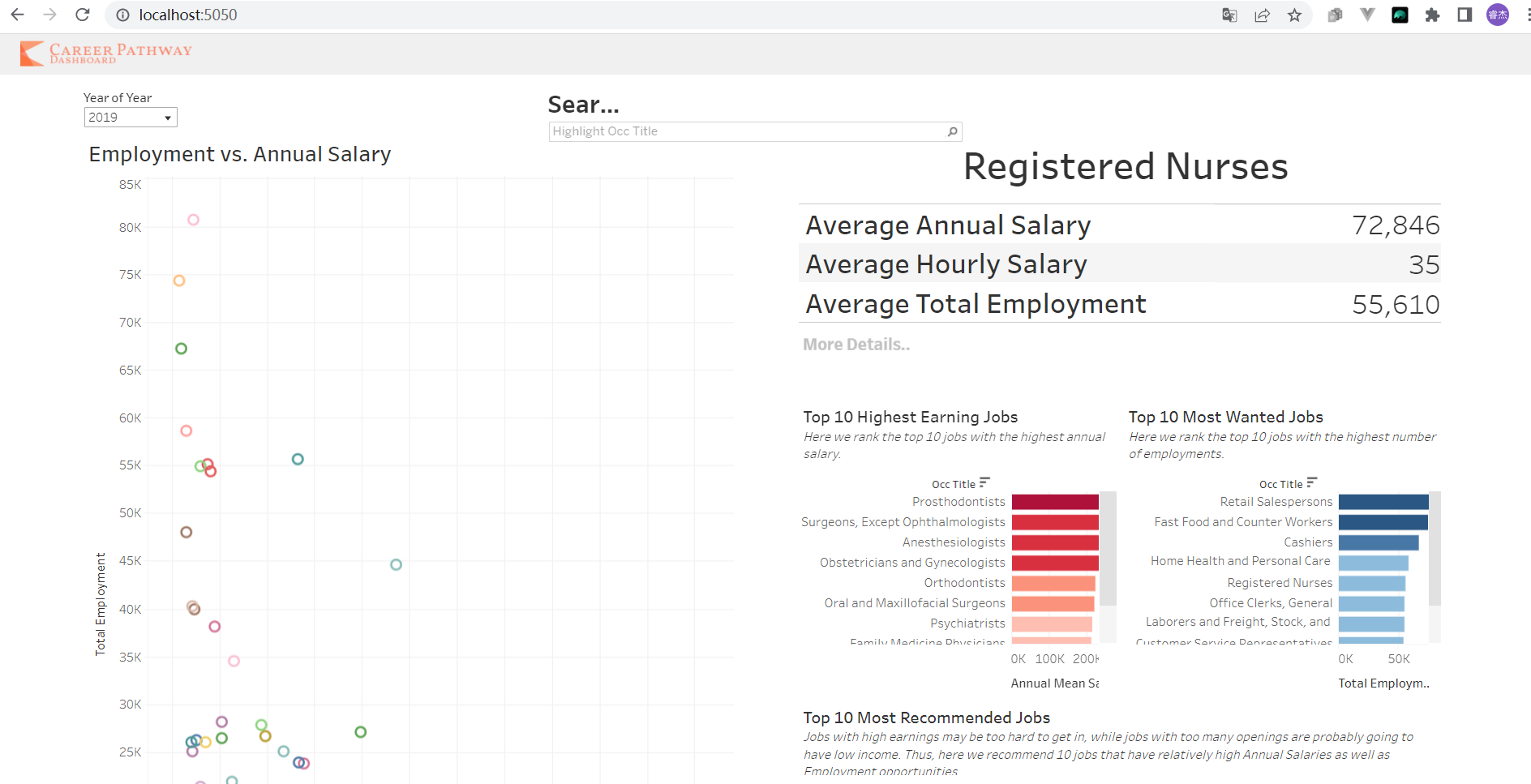
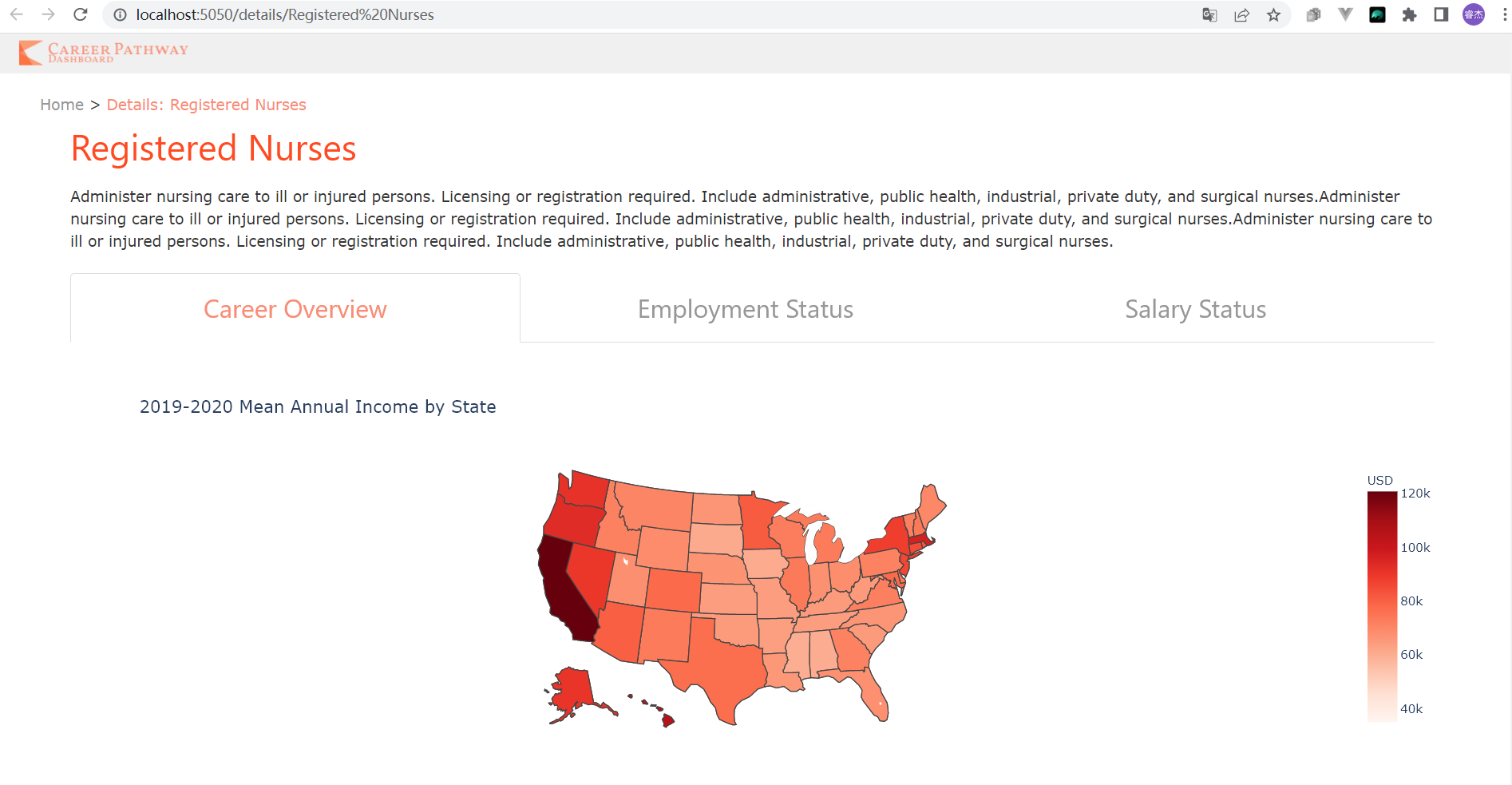
* Completed
* In Progress/Changed
* Not Completed

**Initial plan:**

***Phase 1*:** Data Acquisition/Preparation and Database Configuration

* Acquire static BLS data for 2019 and 2020
  + Monthly
  + State-level
  + Static datasets (xls)
  + Download from the website or use REST API
  + Put it into the Google Drive
  + Configure Google Colab Notebook for data prep
* Data preparation
  + Cleaning (Google Colab/Pandas/Excel)
    - Missing values
    - Remove duplicates
    - Make feature data is in the right format(data type)
      * Saved raw data into CSVs, cleaned with pandas, uploaded to Firebase
    - Remove unnecessary features
    - Text reduction (NOTE: turned out to be unnecessary)
  + Joining time-series data for 2019-2020
* Database management
  + Load cleaned data into database
    - Cloud relational database or NoSQL database such as Firebase
  + Build queries scripts for retrieving data

***Phase 2:***Data visualization and Web app development

* Data visualization
  + Find tool for this (Tableu/Power BI/statsmodel/Python)
  + Develop visualizations
    - Which data to show?
    - (Tableau for the first page, Plotly for the hyperlinked pages)
    - Interactive or not
    - Both the Tableu and Plotly Plots are onteractive
    - How does a user select different visualizations?
      * Clicking on Tableau and hovering over value on Plotly
    - Linking data to visualizations
      * Done with Tableau and Plotly
* Webapp Development
  + Select tools
    - Python Backend, Flask, and Django framework
  + Backend routing
  + Frontend designing

***Phase 3****:* Web app testing and deployment

* Webapp Development
  + Backend routing
  + Frontend designing
  + Frontend developing
  + Deployment with AWS
  + User testing
* Demonstration and report

**Challenges Faced**

* Missing data/how we cleaned data -Alexander/Ahmed
  + There were quite a few missing values to be dealt with, one example was the dataset contained ‘\*’, ‘\*\*’for missing values and ‘#’ for values that were above the set range. To deal with the ‘\*’ missing values we set them to NaNs and ignored them in aggregation functions. Large gaps between salaries across states (aggregation errors) - Alexander
* The upper limit on salaries (caps 200K) - Alexander/Ahmed
  + Values marked with ‘#’ indicated that the value was outside the range of the dataset, and therefore should be treated as outliers. One example of this would be the occupation (CEO), in which the values vary wildly. Instead of dripping the rows, the #s were imputed to the max/and min values for a given feature. This was done so that aggregations wouldn’t be massively skewed by outliers and we wouldn’t lose information.
* Limited time/space in Firebase
  + Adding additional data to firebase was one of the major challenges as it has limited free space. The same issue will be encountered when the data is expanded to include more years than just 2019-2020. -Ahmed
  + The time it takes to add data to Firebase was significant. Loading more than 72,000 for a single year of entries to the firebase takes a long time. Likewise, updating and adding entries took time. -Ahmed
* Plotting (Tableu/Plottly) - Ruijie/Alexander
  + Tableau Public is the tool we found that integrates almost perfectly with web apps structures. As a result of its huge potential, learning to use it has been a great challenge. This includes designing the dashboard, considering the best visualizations to use for different purposes, and inserting user interactions. The most challenging aspect was figuring out how to link the user interaction on the tableau interface with the web app backend. The solution was to add a hyperlink button on the dashboard which directs to the routing url, which includes the string used for querying the Firebase. The backend receives the parameter from the URL and then filters the data accordingly.
  + Another difficulty was on linking the second page to the first page. As you can see on the screenshots, the first page is a dashboard for all the careers, while the second page is about details of a certain career that the user chose. Tableau is not convenient on doing filtering and computing data, so we need another plotting tool. Thus, we chose Plotly which integrates well with Python Flask and also provides a range of plots as well as interactivities.
  + We also ran into a few issues with plotting using Plotly. The main issues plotting the changes in values (income, employment) between 2019 and 2020 for certain states due to missing values in one or the other year. To correct for this we had to filter the plots based on the states/years that don’t have missing values.
* Deploying Webapp - Ruijie
  + Backend routing using flask has been simple, but frontend development is a new challenge. I have never fully designed AND developed a website frontend before, so I have been learning design thinking and CSS/HTML knowledge during the first phase of our project. I used Figma with my mock/flow designs as well as static image design including website icon and navigation bar.
  + Although I have experience with webapp deployment on Azure, Amazon AWS is another thing. I have spent some time on researching guides and tutorials, and also with the help from lecture notes, I was able to upload my scripts on to my EC2 instance and build up the server really fast.

**The expectation on the on-time completion of the project.**

Below is a basic list of things that we know we will be able to complete before the end of the semester and will be present in our presentation. We will also note a few extensions to the project we could possibly complete if time permits.

**Things we will complete before the final presentation:**

* Finish plotting
  + Career overview - Alexander
    - Write a function that returns keys (‘mean salary (hourly/annual)’, ‘total employment’, ‘education level’) and aggregated corresponding values
  + Maps for each occupation - Alexander
    - Annual salary (annual/hourly)
    - Total employment
  + Distribution Plot (?) - Alexander
    - Show an aggregated (by state) distribution of salaries for a given profession.
  + Change plot - (bar/line/gradient arrow)
    - Show the change
* Test website functionality
  + Front end
    - User testing by using a subset of data
  + Back end
    - test plots/queries using all data