NHANES Data Viewer Project Outline

1. **back\_end** – Python Flask app serving API. Runs on Localhost:5000/, start it up by navigating to back\_end directory and running “python app.py” in an anaconda terminal
   1. **\_\_pycache\_\_** - unsure of what this folder does. Shouldn’t need to edit it
   2. **static** – holds static files such as CSS, or JSON. This is a Flask directory
   3. **Templates** – This is where Flask apps look for html files
      1. **index.html** – This is the html file that renders when you open the base url of the Flask app. Renders text and the time the page was loaded. This is used to confirm the Flask app is running. Python variables can be accessed in html with {{ variable }}
      2. **compYear.html** – This is the html file is just a test file to confirm that I was returning a json object from the /YearComp route.
   4. **app.py** – Boilerplate Flask code. Imports Flask, initiates the app, imports our code from routes.py, and then runs the app in debug mode
   5. **functions.py** – Imports python libraries. This is where our personal python functions go. Functions to scrape, clean, merge data will go here and be called in the routes.py file
   6. **routes.py** – Imports app from app.py, and all of our code in the functions.py file. This is where all the routes are created. A route has an associated python function that is run when an API call is made to the corresponding url. A new route to download the XPT files, and return the user requested data is needed

For Example - When the front\_end makes an API call to “localhost:5000/YearComp/2013/Laboratory-Questionnaire”

The python function YearComp(2013, Laboratory-Questionnaire) is called, which returns a JSON object containing information on Laboratory & Questionnaire surveys from 2013-2014

1. **front\_end** – React app structure. User interaction and API calls handled by React.js. React is written in typescript and you can inject JavaScript code within { }. Launch the webpage by navigating to the front\_end directory and running “npm start” within a command prompt. The webpage is hosted locally at localhost:3000
   1. **node\_modules** – This is where libraries/downloaded packages go
   2. **public** – Haven’t touched this folder yet. I believe this is where we can control the webpage icon/logo and such
   3. **src** – This is where all our custom JavaScript/React.js code goes.
      1. **Components** – This is where React.js code for all our custom components goes. Each component gets its own .js file.
         1. **Header.js** – Header of page, link to documentation that opens in new tab, refresh button, basic description. This component persists throughout the stages of data selection and visualization
         2. **PlotlyGraph.js** – This component loads after the user selects the variables they want to work with. Upon loading is unloads the previous variable select form. Currently this is displaying hardcoded Plotly graphs. I am currently working to pass cleaned NHANES data to this form to display more meaningful graphs
         3. **VariableSelectForm.js** – This is the most complex component so far. The user selects a questionnaire, this populates the second drop down with possible surveys. Upon selecting a survey, a third dropdown is populated with all the variables associated with that survey. If available, any table from the documentation pertaining to that variable is then displayed (this is handled by VariableTable.js)
         4. **VariableTable.js** – This component is nested within the VariableSelectForm component. If the value from dropdown3 has an associated table from its documentation, it is displayed. If not, text saying that the variable has no table is displayed
         5. **YearDataForm.js** – This component handles the first form. This is where the user selects a study year, and all questionnaires they want to view surveys from. I don’t particularly love this setup, but it works. The form checks that a year and at least one questionnaire is selected. If not, a warning message is displayed. This warning goes away once the mistake has been corrected. The component calls the api at the /YearComp route and once it returns data loads VariableSelectForm and closes this component.
      2. **app.js** – Initializes the app by creating a class react component. This is the parent component that all other child components will be built within
      3. **index.html** – This is the base html file. It imports some Bootstrap and Plotly.js dependencies. React.js appends to this file to display in the browser
      4. **index.js** – Similar to app.py, this file has some dependency imports and some boilerplate code telling our code to display in the root div in index.html
      5. **style.css** – This page has external Bootstrap CSS configuration, but any custom CSS styling goes into this file
   4. **.gitignore** – Not sure what this is for yet
   5. **package-lock.json** – Not sure about this
   6. **package.json** – Only used this to add our back\_end base url (localhost:5000/) as a proxy to allow the two to talk without raising a CORS error