Final Project Document for Data Analytics Team Project

Using earlier project web design as a guide

[MaternityExample](https://maternal-mortality-project.herokuapp.com/united-states-mmr-affordable-care-act) - example

Project Repository: <https://github.com/embrexemb/Project-03>

* Web assets and libraries
* Bootswatch – theme – Spacelab <https://bootswatch.com/spacelab/>
  + - Bootstrap current release is v5.0.2 <https://getbootstrap.com/>
    - D3.js – release version – <https://cdnjs.cloudflare.com/ajax/libs/d3/4.11.0/d3.js>
      * Used with Bootstrap version 3.3.7 [assignment 14 – UFO Search]
* D3 assignment with plotly.js [assignment 15 – Belly button]
  + Used Bootstrap version 3.3.7
  + <https://d3js.org/d3.v5.min.js>
  + <https://d3js.org/d3-scale-chromatic.v0.3.min.js> - for colors
  + Plotly.js – release version – <https://cdn.plot.ly/plotly-lastest.min.js>
* D3 assignment for dynamic scatter plot [assignment 16 – Scatterplots]
  + Used bootstrap 4.1.3
  + <https://code.jquery.com/jquery-3.3.1.slim.min.js>
  + <https://cdnjs.cloudflare.com/ajax/libs/popper.js/4.1.3/js/bootstrap.min.js>
  + <https://d3js.org/d3.v5.min.js>
  + <https://cdnjs.cloudflare.com/ajax/libs/d3-tip/0.9.1/d3-tip.js>
* Each chart has its own javascript code
  + Candlestick\_chart.js – Rupesh -Dashboard
  + Dynamic\_Scatter\_Plot.js – Eve/John -Dashboard – shows stocks by industry sector
  + DailyActivityExchanges\_Chart.js Eve -shows on Homepage and Dashboard

[assumptions: Using HTML5, CSS and Javascript with Bootstrap components that are tested and compatible]

* Webpage template has navbar with menu, Header, page content section, Footer

[give links to the templates folder on git]

* Master Page
* Navbar menu will be horizontal with
* Home, Home page is a splash page with introduction
* Dashboard, [Layout will get done last - each chart on the dashboard can be clicked through to display on its own page] – 4 chart pages will be set up
* Methodology, [Describes data sources, data collection, data cleaning; APP architecture using Flask API program flow, how the webpage listeners are triggered to access data and display it on a chart or page]
  + - * Methodology page should have an entry from each person for the work they did
* Flask API
  + Requirements.txt for the Heroku build
  + Runtime.txt
  + Procfile to set up app instruction
  + app.py – wrapper for Flask application – set up and deploy initial flask api to confirm a webpage can be loaded on Heroku and data can be sent to Mongo db and retrieved.
* Data Repositories
  + MongoDB – for saving historic data by stock and close date
  + S3 bucket to save daily processing files that are used by Heroku
* Machine Learning work in the Notebooks section – the charts can be plotted from results saved to a file