

---

# Design Plan Stream 2

---

## PROJECT BRIEF

In this project you'll be building a data-driven frontend and backend website using the technologies that you have learned throughout Streams 1 and 2.

You can either choose to use the example brief below or you can use your own idea for the website.

---

## PROJECT GUIDELINES

---

- The data that you choose to work with, should be stored in either an SQL database (preferably MySQL), or a noSQL database (preferably MongoDB)
  - The project must use Flask to retrieve the data from the database and return it to the browser.
  - The dashboard should include, at the minimum 3 separate graphs. Line/bar graphs and pie charts are particularly useful, but you can use any graph types that you feel suit your dataset.
  - The front end should use the D3.js library (or similar) to create data-driven charts and the user must have some way of interacting and filtering the data in the charts using DC.js and crossfilter.js, or another equivalent.
  - Use as much functionality as you deem necessary from the lessons
  - Make sure your site is as responsive as possible. Use appropriate testing sites to test your web sites in several different environments
  - Write a README.md file for your project (in Markdown format) that explains what the project does and the need that it fulfils. It should also describe the functionality of the project, as well as the technologies used. Detail how the project was deployed and tested and if some of the work was based off other code, explain what was kept and/or how it was changed to fit your need. **A project submitted without a README.md file will FAIL.**
  - In addition to the README.md file, you may include in your repository supplementary documentation and/or other relevant supporting material for the assessor in any format that is automatically handled by web browsers, such as html, pdf, jpg, etc. Files in proprietary formats such as Microsoft doc/docx will be ignored; but this is generally not a hindrance, since the vast majority of formats can be easily exported to PDF.
  - Use Git & GitHub for version control. Each new piece of functionality should be in a separate commit.
- 
- You should deploy the final working version of your code to Heroku (or an alternative hosting platform that you are familiar with)
-

---

## IDEA

---

### *World Countries Info Dashboard*

- *Includes information on World Countries:*
  - *Continent*
  - *Population*
  - *Area*
  - *Currency*
  - *Languages*
  - *Drives on the left/right*

---

## BUILD

---

### *Dashboard of graphs*

- *Database built on MongoDB*
- *Information fetched by Flask*
- *Information filtered by Crossfilter*
- *Data manipulated into graphs using D3.js & DC.js*

---

## LAYOUT AND COLOR SCHEME

---

