CSC 301, Final Project
Spring 2022

In this course you have learned **Advanced Web Development and Web Scraping** techniques. In particular, you learned basic and advanced web development using HTML, CSS, and JavaScript. You saw how web browser extensions can be created by injecting JavaScript into an existing page, and how with a knowledge of web page structure and JSON you can scrape data from web pages or request and process data from an API, using either JavaScript or Python. Finally, we went over how to develop dynamic, data-driven web pages using R/Shiny. For your Final Project, you will select one of the assignments below, and will give a presentation describing your project. You may work with a partner on the final project.

Final Project Options

1. **Develop a Chrome browser web browser extension that adds functionality to one or more web pages.**

2. **Develop a web scraper or use a web page's application programming interface (API) to provide a service or to answer a question of interest.**

3. **Develop an R/Shiny application for visualizing or analyzing data that is read from a local or remote file, or obtained through web scraping (using *rvest*) or an API.**

Useful Resources:

- Links to and examples of many APIs: https://any-api.com/
- Datasets: https://www.kaggle.com/datasets and http://rs.io/100-interesting-data-sets-for-statistics/

Code availability:

- You are encouraged to use *Github* to manage your code. (If you have not used Github before, here is a tutorial: https://guides.github.com/activities/hello-world/)
- If developing a Shiny application, you are encouraged to make your app available on https://www.shinyapps.io/

Submission:

- You must post your project idea to Piazza for feedback and approval (see the Piazza post for more details)
- You will submit the code for your final project and your final presentation separately through Blackboard. Both must be submitted prior to your final presentation.

Piazza Discussion / Presentation Viewing:

Following our presentations, we will use Piazza for questions and continued discussion. More information about this requirement will be given at a later date.

Note: All programs submitted must be your own. If any portions of your program are found to be plagiarized, then your project will not be accepted.

Presentation times:

- All presentations will be given "in-class" beginning on May 2, 2022. A sign-up link will be shared shortly.

Technical Requirements (80 points)			
	Poor (C or below)	Acceptable (B range)	Excellent (A range)
Web browser extension	The extension performs only a trivial function, or the extension does not work as described.	Most, but not all, “Excellent” requirements are met. For example, very little CSS is used, or the extension is functional, but does not have an elegant look.	HTML and CSS are used to create an elegant pop-up or to display meaningful information on a web page. The extension injects JavaScript to perform a non-trivial function, for example by searching and extracting information from multiple elements on a page.
Web scraper / Analysis using a Web API	The web scraping or analysis is trivial (e.g., the web scraper gets only the titles of a set of pages), or the scraper does not work as described.	Most, but not all, “Excellent” requirements are met. For example, the webscraper extracts data from only 2 elements, or 2 pieces of information are extracted from a web API.	Either (1) the webscraper scrapes data from multiple (≥5) pages and extracts data from ≥3 elements per page; or (2) an API request is made, a resulting JSON-formatted response is parsed, and multiple (≥3) pieces of information are extracted. In both cases, results are presented to the user graphically and in a structured text format, and the analysis answers a non-trivial question
R/Shiny Application	The R/Shiny application is trivial or very similar to an available example, or does not work as described.	Most, but not all, “Excellent” requirements are met. For example, the Shiny application allows the user to change only 2 inputs, or a web scraping application does not include a graph.	The Shiny application either (1) allows the user to change at least 2 inputs and updates visualizations and analyses appropriately, (2) accepts user input for the purpose of scraping web page data or querying an API and presents the results in at least 1 graph.

Documentation (20 points)		
Poor (C or below)	Acceptable (B range)	Excellent (A range)
Code documentation is minimal or not provided. Variable and function names are not descriptive and the code is formatted poorly, making the program difficult to read.	The majority of functions and major code segments are documented. Variable and function names are chosen appropriately and proper formatting (such as indentation) is used in the majority of the code. The logic is easy to follow but some aspects of the code are not.	All functions and major code segments are properly documented. Variable and function names are chosen appropriately and proper formatting (such as indentation) is used, making the program logic easy to follow.

Presentation (50 points)

	Poor (C or below)	**Acceptable (B range)**	**Excellent (A range)**
Title Slide (5 points)	A title slide is not included		The title slide contains a descriptive presentation name, the name of the presenters, and the date of the presentation.
Objective (1 slide) (10 points)	The objective is not mentioned.	The objective is mentioned but is not described in a clear way.	The objective of the project and the target audience is clearly described
Demonstration / Results (15 points)	No demonstration or results are included.	A demo is given, or results are provided, but are not clearly explained. For example, graphs are presented but it is unclear what they are showing.	A step-by-step demo of the program is given, or results are presented, and clearly explained.
Methods (2-3 slides) (15 points)	The methodology is not explained.	The methodology is explained but is not entirely clear. For example, screen shots of code are included but are difficult to read.	General methodology (programming languages, modules/packages, etc) is stated. Example code (with screen shots) is included that clearly explains at least one unique and challenging aspect of your project*.
Summary / Conclusions (1 slide) (5 points)	The summary and conclusions are not included or are not informative.	-	The presentation ends with a summary of the project and concluding remarks.
Presentation Time	The presentation is far too short or too long. (-15 points)	The presentation is too short or too long, but only by 1-2 minutes. (-5 points)	The presentation is between 9-11 minutes long

*Because of time constraints, you will not have time to cover too many details of your code. However, you must give a general overview ("Web scraping was carried out using Python; the *requests* module was used to retrieve the contents of each web page, and *Beautiful Soup* was used to extract information from each page…". You must also cover at least *one* aspect of your code in detail. This will involve showing selected lines of code and explaining what the code does. This should not be trivial (such as explaining what *requests.get* does), but should represent something unique and challenging to your project.