Digital Research Lab

Stage 2 – Ability Test

Following test is divided into 4 sections. Order of completion is up to you. Your objective is to complete as much as possible of given tasks in a span of 1 week (7 days). At least 1 fully completed section is required. Detailed requirements will be provided within tasks. Start and end dates will be provided below. By the end of testing phase, you are expected to prepare presentation / all source codes for a review. Contents of presentation should include objective of the task, how it was completed, and end result (demo in case of web-apps or standalone apps). You are expected to provide codes via Github or in any other way that's comfortable for you.

Best of Luck.

Datasets: https://1drv.ms/u/s!AqfcrlkI5ANOgkYoSXp69UR2paLa?e=jI62O2
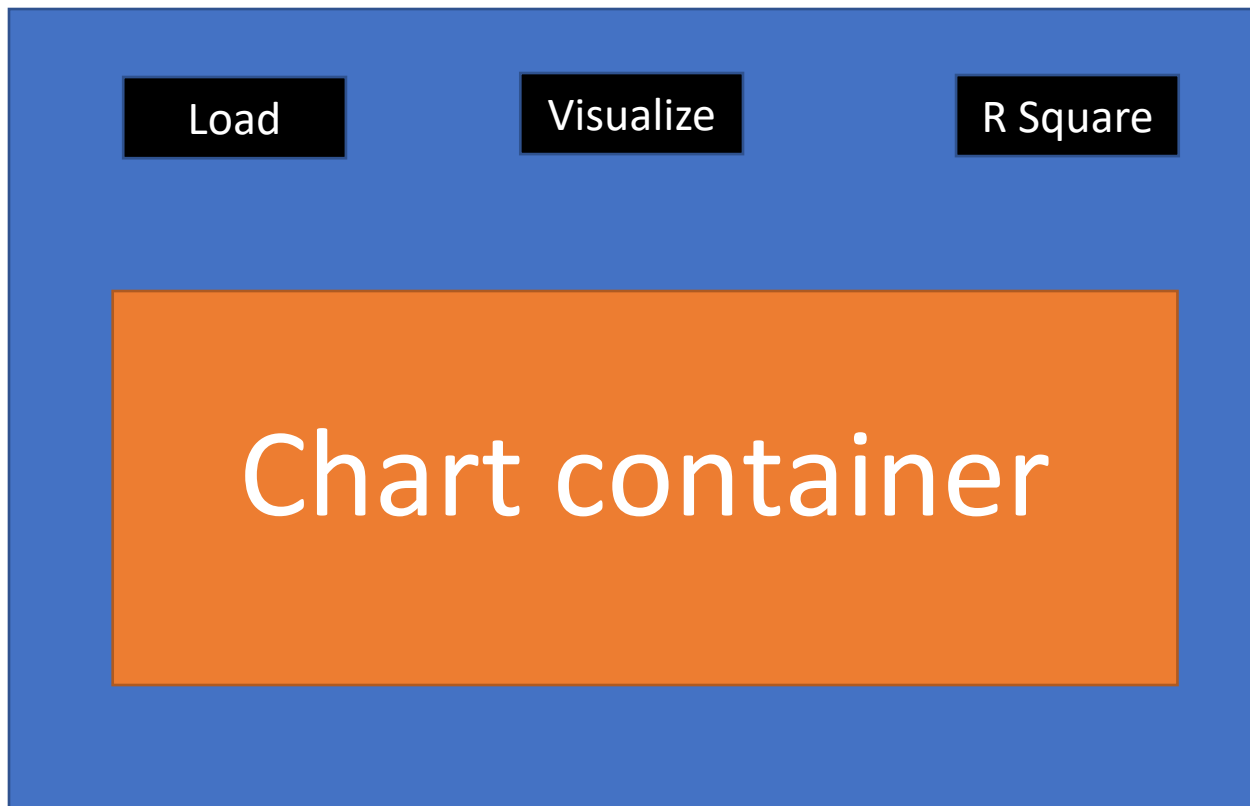
Start Date:

End Date:

Section 1:

This section will test your UI designing skills, as well as, basic programming knowledge.

Task list:

Create simple User interface with three options (data loading, visualization (**a vs b**) and calculation of R square for **a vs b.** Draft provided below:

Load Data – button which will load provided dataset (Section_1.xlsx) into the system.

Visualize – button which create simple scatter plot of provided data in Chart Container.

R Square – button that should calculate **coefficient of determination (R^2)** and provide output to the user.

Whether user interface is WUI or GUI is up to you. You should provide source code by the end of the task.

Section 2:

This section will test you programming skills, critical thinking skills, and dataset optimizations

Task list:

In the next section you will see part of the sequence for a **specific type** of numbers:

$$\dots, 5, 7, 11, 13, 17, \dots$$

1) Identify what kind of sequence it is. Write algorithm that will find numbers from this sequence till Nth number. Sum 1st, 2nd, 101st, 1001st, and 10001st numbers from aforementioned sequence. Provide list of every number in the sequence till 10001st in csv file. You should provide source code by the end of the task.

Quadratic equation of type

$$A * n^2 + B * n + C \,,$$

could be used to find numbers from sequence in first part of this exercise.

With following conditions:

$$A = 1, \qquad B \leq 1000, \qquad C \leq 1000,$$

2) Find combinations of **B** & **C** that will give top 3 amounts of consecutive numbers from sequence in the first part of this exercise for $0 \leq n < +\infty$. Example: $n^2 + n + 41$ could be used to find numbers from sequence for 40 consecutive n numbers ($0 \leq n \leq 39$). After calculating top three sets of **B** & **C** combinations, multiply them to each other (**B1*C1, B2*C2,** etc.). Provide list of those 3 numbers, as well as, sum of all 3. You can use task 1 in order to verify task 2. You should provide source code by the end of the task.

Consider following matrix of 5 by 5:

21 22 23 24 25
20 7 8 9 10
19 6 1 2 11
18 5 4 3 12
17 16 15 14 13

If we consider numbers on diagonals (red numbers) they add up to 101.

3) Create an algorithm that will generate same matrix for **N** x **N**. Use this matrix to find following:

    a. Sum of all numbers on diagonals for N = 7, 13, 19

    b. Multiply numbers on the corners for N = 7, 13, 19 (In the example: 13, 17, 21, 25)

You should provide source code by the end of the task.

Section 3:

This section will test your data analysis skills

Task List:

Yelp Reviews. Analyze provided dataset (Section_3.csv) and find answers to following questions:

1) How many unique restaurants could be found in this data set?
   Hint: Use [Business_ID] column for this evaluation.
2) Which restaurant received highest number of reviews? What about percentage-wise?
3) Which cities have got at least one 5 star review in Nevada (NV) state?
4) Which city has got the highest number of reviews in the Business Category of "Hotels & Travel"? What about percentage-wise?
5) At what day of the week people are more likely to post their reviews?
6) Showcase if there are any trends regarding restaurant performance as time goes by.
7) Based on analyzed data showcase if there any steps that restaurant can take in order to improve their public appeal.

   Bonus Question:

8) Based on this data-set which user had the highest cumulative travel distance? What distance has been covered by him/her?

Dataset Columns Description:

'Review_Date' – date when the review was posted by user

'Review_Text' – text of the review

'User_ID' – Unique Identification Number of users, who made this post

'Business_ID' – Unique ID of business. Please note, this column distinguishes business with the same names (for instance chain of restaurants)

'Business_Name' – Official business name

'Business_Category' – Category in which this business operates

'City' – City location

'State' – State Location

'Latitude' – X coordinates of business

'Longitude' – Y coordinates of business

'Avg_Business_Star_Rating' – Rating review left by user.

Section 4:

This section will test your full-stack development capabilities, as well as, database manipulation

Task List:

IP Dataset.

Please develop a web application which searches given IP address within stored IP address ranges.

Input value: IP address; (it is necessary to provide client-side validation, as well as, server-side validation to ensure validity of IP address).

IP Address ranges could be stored in a text file, xml file, json, or RDBMS. (Section_4.xlsx)

This application should be a web application. You can use any combination of programming language and web framework. (Ex. Python – Django, C# - ASP, etc.)

User enters IP Address to the input and clicks on a search button. If the IP address is valid, Web application looks through all IP ranges and if found, shows the range to the user.

If IP Address not found then application offers to the user to add new IP Range to the list. If User Clicks OK then web application asks a new IP range (2 textboxes) and after clicking on the save button, web application adds the new range to the list of IP address ranges. But there should not be any intersection between stored IP ranges.

You should provide source code by the end of the task.