Proposer Details

Group Number	50
Registration Number of Group Members	2020-CS-02 2020-CS-30

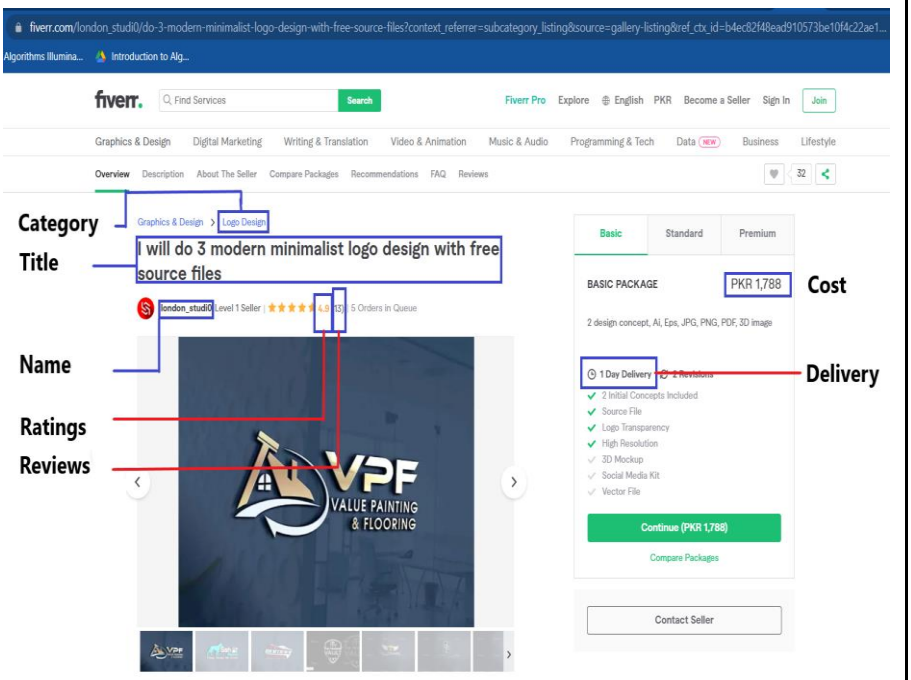
Proposal Details

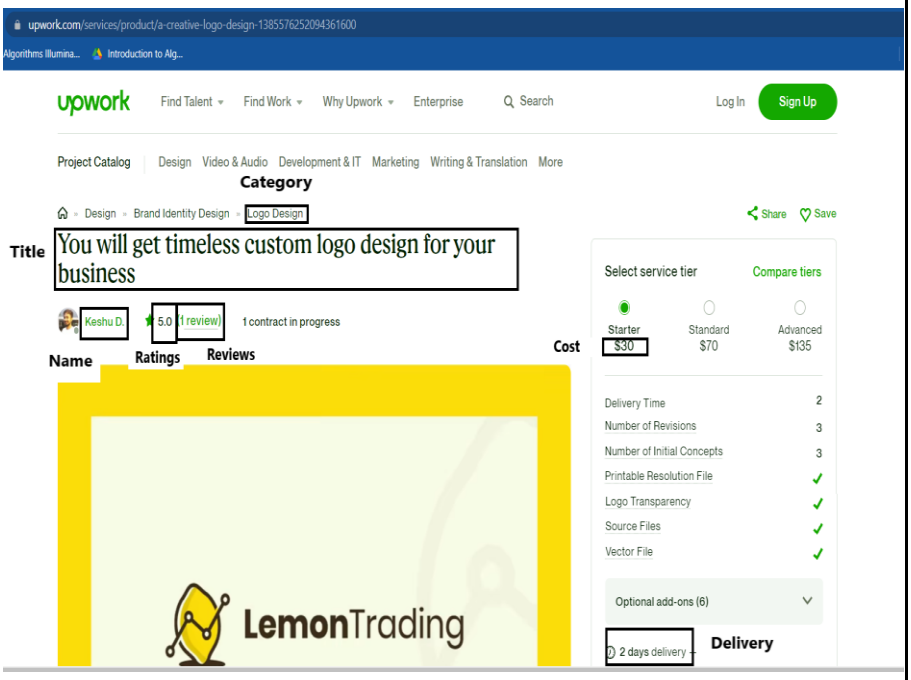
Project	
Proposed Project Title	Virtual World Projects
Executive Summary	The project is a retrieval system where data related to projects available on freelancing websites will be extracted. The data will be useful for online jobs analysts where they would be able to see project's details in a specific manner. For example, they would be able to see projects with the highest payments, highest reviews, ratings, etc. 1 million data is to be fetched and different sorting and searching techniques will be applied converting the data into some type of information. The project's flow will be in a manner where 1 million data will be extracted as a result from different freelancing websites using their URLs. The data scraping will be controlled by the users where they would be able to start, pause, resume, and stop the scrapping. The data extracted then will undergo sorting techniques where the users would be able to sort the data according to the types provided to them. Searching on different attributes of the entity will be according to the data type of the respective attribute for example, for integer the searching will be according to the ranges of integers and in case of string it will be according to the letters in the word. A progress bar will be displayed on the UI showing the scrapping progress. The main purpose of the project will be to show the time taken for each sorting technique. We would be able to examine the time that takes for each sorting technique and tell which technique will be the best for a specific amount of data. Sorting on a particular column and among columns (multi-level sorting) will be a feature providing the user with different types of sorting.
Business Case	
Outline the business need for the project	Analysis of online projects available on freelancing websites requires them to be in an area where they can be sorted according to their ratings, prices, categories. Analysts can use this to determine the projects available in a certain category, the prices of each project, and the ratings of the project dealer telling if it is suitable and safe to take the project of the person. Moreover, the time analysis of sorting

<table>
<tr><td></td><td>techniques can help Algorithm Analyzers to conclude which algorithm works best for a certain amount of information.</td></tr>
<tr><td>End user of the product</td><td>Online Project’s Analysts, People searching for projects, and Algorithm Experts analyzing time for each algorithm.</td></tr>
<tr><td>Motivation for Project</td><td>The project will be helping in visualizing how sorting is applied on a real life problem. So far, we have been implementing different types of sorting algorithms on integers and now to extend them on real life examples will give a more broader and vivid understanding of these algorithms. Seeing the time complexity of algorithms on large data will help to explain their limitations and as a result tell which algorithm will be the best when we have large data or small data.</td></tr>
<tr><td>State the level of impact expected should the project proceed and implications of not proceeding</td><td>If the project is featured, Online Project’s analysts can have a platform from which they can analyze the projects available and make conclusions regarding the availability of projects of a certain category. Moreover, sorting the data and providing a graph for the analysis of algorithms will help Algorithm Experts to view how the number of data can change the feasibility of an algorithm .It won’t make a big impact because websites like Upwork and Fiverr are providing these sorting and searching techniques from where analysis can be done, but the graph feature will help in studying optimization of algorithms.</td></tr>
<tr><td>Technical Details</td><td></td></tr>
<tr><td>Name of Entity</td><td>Freelancing Projects</td></tr>
<tr><td>Attributes of Entity
(Minimum seven attributes/rows can be increased)</td><td>
<table>
<tr><th>Name</th><th>Data Type</th><th>Description</th></tr>
<tr><td>Title</td><td>String</td><td>The title will tell what the project is about.</td></tr>
<tr><td>Category</td><td>String</td><td>The category in which the project falls for example, making logos, symbols, trademarks , etc. will fall in the category of logo design.</td></tr>
<tr><td>Name</td><td>String</td><td>The person who is advertising the project.</td></tr>
<tr><td>Cost ($)</td><td>Integer</td><td>The cost for completing the project.</td></tr>
<tr><td>Delivery (Days)</td><td>Integer</td><td>Number of days in which the project will</td></tr>
</table>
</td></tr>
</table>

		be delivered.
Reviews	Integer	Number of reviews of the person's project.
Ratings	Float	Rating of the person providing the project.

Sample of Scrapping Source





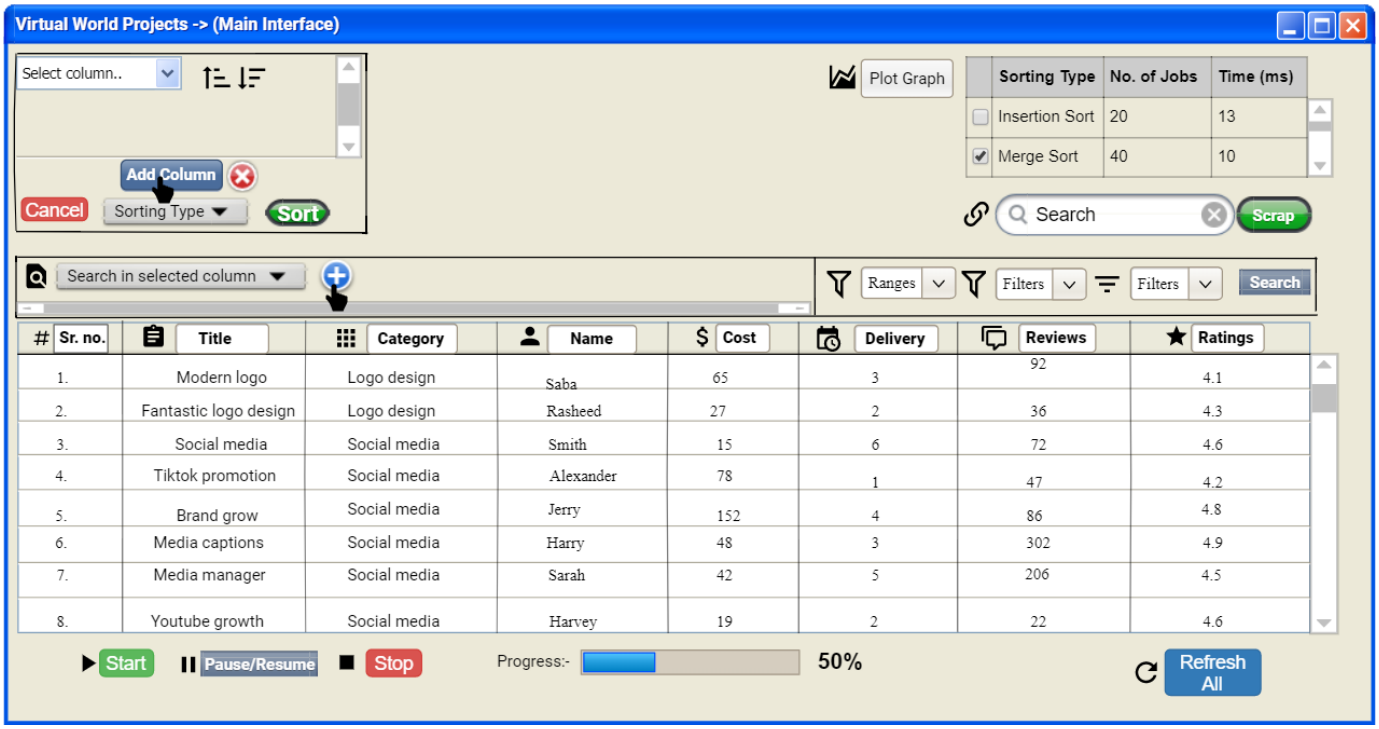
Github Repository Link	https://github.com/Huzaifa-crypton/CS261F21PID50.git

Sorting Algorithms	Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, Heap Sort, Radix Sort, Bucket Sort, Counting Sort, Comb Sort & Cycle Sort.

Algorithm Name	**Description(Each algorithm in 2-3 lines)**
Insertion Sort	We have three parts of the array; the sorted part, current element, and the unsorted part. Current element is picked starting from index 1 of the array and is inserted in the sorted array at its correct position.
Selection Sort	First the minimum element throughout the array is selected and inserted as a first element of the array. Then the second minimum element is selected and this goes on until all elements don't get sorted.
Merge Sort	A recursive algorithm which divides the array into two halves until single elements are left. Then each half is sorted and combined with its other half making the array sorted.
Bubble Sort	Iterative algorithm which swaps adjacent elements w.r.t the condition applied e.g. if the left element is larger than the right adjacent element, then swap. This causes a sorted array to appear from the end of the array.
Quick Sort	We select an element from the array and arrange the elements smaller than it on the left and greater to the right. Then the Quick Sort function is called again and sorts the left and right parts around the pivot.
Radix Sort	Sorts the numbers from least significant integers to most significant. Numbers are inserted into buckets from 0 to 9 for each integer and at the end the first element inserted is removed first from the bucket.
Counting Sort	Make an array containing 0's of length equal to the largest number contained in the input array. Count each number in the input array and increment at the index equal to the element in the new array. Create a final Element and insert the elements according to a rule.
Bucket Sort	Make Buckets equal to the largest element of the array. Insert each element in its bucket and sort each bucket individually. At the end, combine all of the buckets in a final array.
Heap Sort	We create heaps of the input elements and then make a max. heap by making the root nodes value maximum. After that we swap the root node with the smallest element at the end of the heap and remove the last largest element.
Gnome Sort	Type of bubble sort in which iterate through the array and check if elements at current index and previous index are at the correct position or not. If not then swap and decrement the current index, otherwise, keep on iterating.
Comb Sort	Calculate a gap dividing the total number of elements with 1.3 and compare elements starting from index 0 with index = [gap+index] and swap if the left element is greater. In the next pass, divide the gap by 1.3 again and keep doing the swapping until a sorted array appears.
Pigeonhole Sort	Make an array of size equal to the range. Iterate through the array and subtract the number with the minimum number which will give the index of the new array where the current element is to be placed. At the end, place the elements of the new array in the original array giving a sorted array.

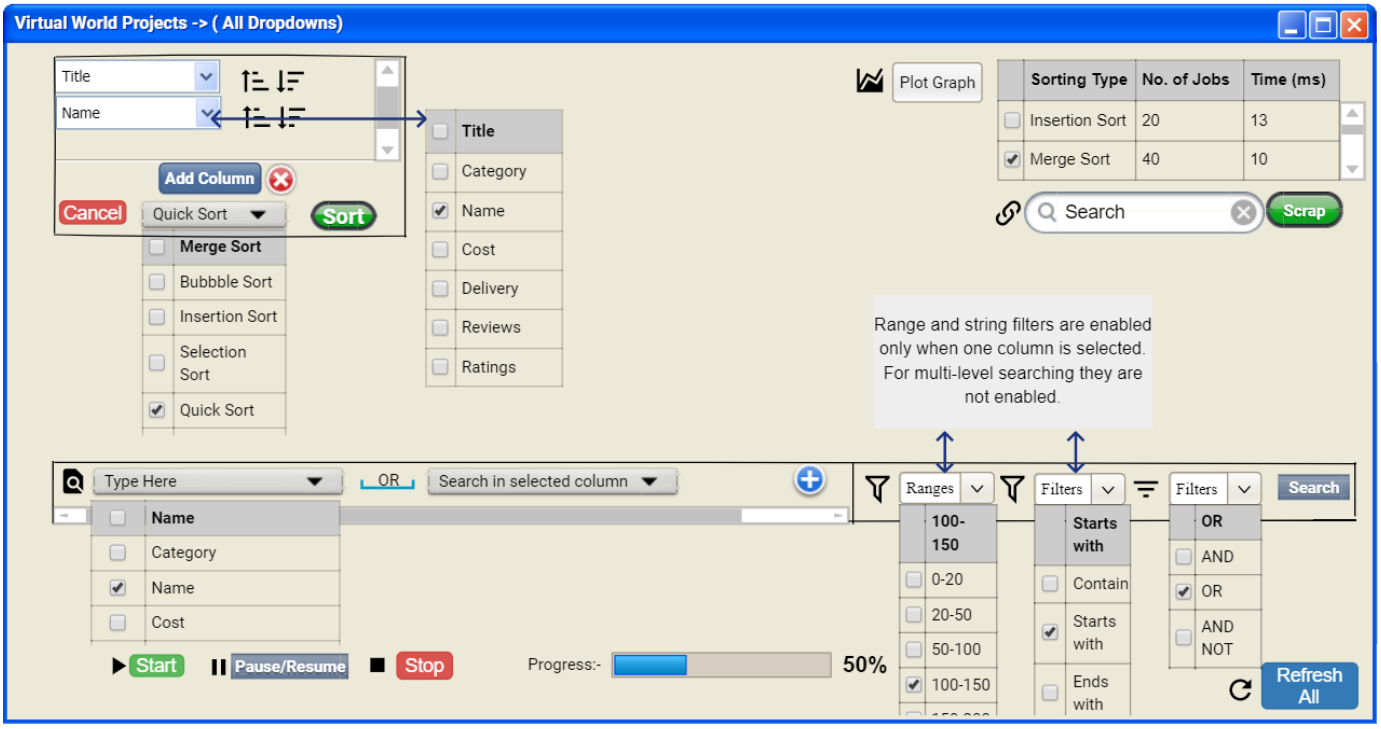
Shell Sort	Calculate the gap starting from n/2 and compare elements starting from index 0 with elements at index (n/2+current Index). Swap the elements if the left element is larger. Decrease the gap again by n/2 and again check the elements until a sorted array appears.
Searching Algorithms	1. Linear Search: Where we will iterate through the complete array one by one and find the required information. 2. Binary Search: A divide and conquer technique where we sort the array and divide it into two parts and check in which part our answer would be and discard the other half. We continue to do this until we find our required number.
Searching Filters for each data type	**Strings:** Contains, ends with , starts with **Integers:** Ranges Cost, Reviews>>> (0 -20), (20-50), (50-100), (100-150), (150-200), (>200) Ratings>> (0-1) (1-2) (2-3) (3,3.5) (3.5,4) (4,4.5) (4.5,5)
Multi-Level Sorting	We will first sort a column e.g., we sort our project titles using their first letters and afterward we can sort the next column of cost in such a way that Cost gets sorted in the range of alphabet A, then for B it sorts in the range of B alphabet and it goes on for other letters.
Any other features	We will be putting in the option to make graphs of the number of items and the time taken to sort them using each Algorithm giving us a clear view that from which value an algorithm's performance is worse than other algorithms.

Interfaces for your project	



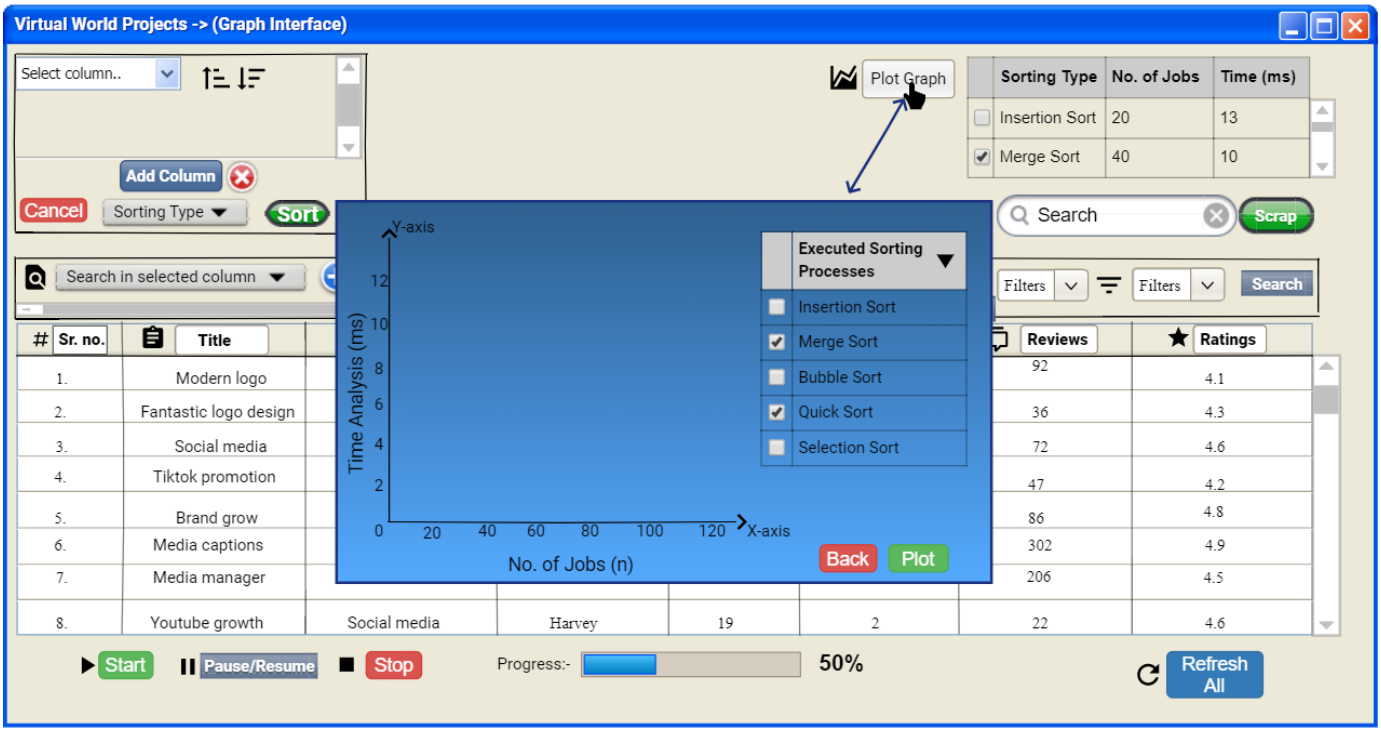
UI Component Name	Type of UI component	Purpose of UI Component/Other details
Select column	Dropdown	It will be used to avail the option of selecting column or multiple columns.
Sort ascending	Icon	By clicking this icon, we will be able to sort the selected column in an ascending manner.
Sort descending	Icon	By clicking this icon, we will be able to sort the selected column in a descending order.
Select sorting type	Dropdown	This dropdown will provide an option to select different types of sorting algorithms in order to sort a specific column.
Sort	Button	This button will enable the process of sorting.
Cancel	Button	By clicking this button, the user can unselect the column or algorithm type.
Plot graph	Button	This button will allow us to plot a graph between two types of sorting
Search	Search bar	This bar will take the URL as input.
Scrap	Button	The button will scrap the material

		according to the given URL.
Search in selected column	Dropdown	It will allow us to select a column & in result, we can search in that column using filters.
Add	Button	This will allow you to search through multiple columns using filters.
Ranges	Dropdown	In order to select the range for columns in terms of integers.
Filters	Dropdown	In order to select the filters for columns in terms of strings like contains, starts with & ends with, etc.
Filters	Dropdown	These composite filters will serve for the process of multi-column searching using AND, OR & NOT.
Start	Button	To start scraping.
Pause/Resume	Button	To pause and resume the scraping process.
Stop	Button	To stop scraping.
Progress	Bar	To show the progress of scraping.
Refresh All	Button	To reload the whole page and to discard all the changes.



UI Component Name	Type of UI component	Purpose of UI Component/Other details

Title / Name	Drop Down	To select the column name.
Add Column	Button	To add columns for multi-level sorting.
Quick Sort	Drop Down	To select the sorting type.
Sort	Button	To start sorting.
Plot Graph	Button	To plot the graph and view graphs.
Type Here	Text Field	After selecting the column, write text to search in the selected column.
Or	Text Box	In multi-column searching when AND, OR and NOT are selected, the text box fills up.
Ranges/Filters	Drop Downs	To select filters.
Search	Button	To search for the written text in type here.
Search	URL Text Box	For entering the URL of the website from where scraping has to be done.
Scrap	Button	To start scrapping.



UI Component Name	Type of UI component	Purpose of UI Component/Other details
Plot Graph	Button	To open the Blue graph Interface.
Back	Button	To go back to the main page.
Plot	Button	To plot a graph.
Executed Sorting Processes	Drop Down	To select the algorithms whose graph is to be plotted.