**Proposer Details**

| Group Number | G-06 |
| --- | --- |
| Registration Number of Group Members | Member 1:2020-CS-111  Name: Aqsa Riaz  Member 2:2020-CS-129  Name: Noor Atif |

**Proposal Details**

|  |  |
| --- | --- |
| ***Project*** |  |
| Proposed Project Title | Laptop Scraper |
| Executive Summary | The Title of our project is Laptop-Scraper. No matter how you use computers, the overarching reason to own a laptop is portability. Unlike smartphones and most tablets, laptop computers run the same programs as their desktop counterparts, so you can take your work and entertainment with you wherever you go without relying on mobile apps. Depending on your needs, you can either own a laptop alongside a desktop computer or replace your desktop entirely. Now a days laptops are essential for us in our day to day life so that is why we are scrapping laptops, all the latest and old models in a single file so that is becomes easier for a person to compare while buying a laptop. We will basically scrap different sort of laptops including gaming ones and other heavy machines. The data will be scrapped from multiple websites including Daraz.pk, Flipkart, Amazon.com, Ali Baba etc and the scrapped data will be then appended in one single Csv/Excel File. It will have all the basic entities including product name, Brand, Generation, General Description, Price, Rating, colour etc. The UI will be designed by using PY-QT. Scrapping will be done by using Beautiful Soup and various other features. It will have the option to pause, start, resume and stop, with the progress bar showing the progress of tasks or the number of entities scrapped. Now talking about sorting of data. We will apply all the sorting algorithms studies in class including Insertion sort, Merge Sort, Selection Sort, Bubble sort etc The user will have the option to sort the data column wise or row wise in many ways in ascending order or descending order based upon various features. |
| ***Business Case*** |  |
| Outline the business need for the project | We are Scraping 1 million laptops, after loading the data in the csv file our python code will present the whole scraped data and then we can sort the data further through different sorting techniques*.*  Different E-commerce companies and Retailers choose the Best laptops to put on their websites, those ones which are good in performance and are meeting all the definite standards. So for this purpose we are scraping the complete description of a laptop so that it can be easier for them to compare and sale them according to their requirements and customer demands. |
| End user of the product | End user could be any user, any organization Particularly retailers or even buyers willing to buy a laptop. |
| Motivation for Project | **Problem:** Suppose the retailer or any person is buying a new laptop. It becomes very difficult to like compare all the specifications of a laptop. One has to check a lot of things including brand, color, generation, Processor and most importantly price range. This whole procedure takes a lot of time. Like opening a particular website then checking different brands etc will require a lot of time and efforts.  **Solution:** So Our project will help the end user to compare all the information of the laptops at a single click. |
| State the level of impact expected should the project proceed and implications of not proceeding | Whenever the end user wants to compare laptops, the complete data will be available at a single place with just a single click  **Benefits:**   * Much Easier Process * Convenient * Very Less Time consumption * Almost no efforts required * Everything will be Just A click away |
| ***Technical Details*** |  |
| Name of Entity | Laptop |
| Attributes of Entity  (Minimum seven attributes/rows can be increased) | |  |  |  | | --- | --- | --- | | *Name* | *Data Type* | *Description* | | Brand | Array | This attribute will be used to store the brand of the product. Every product has a different brand. So thorough this attribute the user can access any brand easily. | | Processor | Array | This attribute will describe the processors used in different laptops | | Generation | Array | Generation means the age of the CPU. So in this attribute will store the age of the CPU means through this attribute user can access the generation of the laptop. According to the generation the price of the laptop will be decided | | Ratings | Array | Rating means quality use that how much a user is satisfied with this product. A brand having good rating is considered batter in the industry. So through this attribute the users will access the ratings of the product | | Price | Array | This attribute will be used to store the price of the product. | | Color | Array | Different colors of the product can be decided or chosen by the user through this attribute | | Hard-Disk |  | This attribute we will describe the Hard-Disk used in different laptops. | | RAM |  | This attribute we will describe the RAM used in different laptops.  Like 8 Gb,12 Gb etc. | | Size |  | This attribute we will describe the Size in Inches of each laptop.  Like 13 Inch, 15 Inch etc. | | Capacity |  | This attribute we will describe the capacities of different laptops.  Like 256 Gb, 512 Gb etc. | |
| Sample of Scrapping Source |  |
| Git-hub Repository Link | https://github.com/NoorAtif/CS261F21PID54 |
| Sorting Algorithms | Sorting means arrange the data and store it into a list so by using the sorting algorithms we can arrange the data in different orders. Like if we want to arrange the data in descending or in ascending order this will be done with the help of the sorting algorithms.  Following are the sorting algorithms that are used in this project |
| |  |  | | --- | --- | | **Algorithm Name** | **Description(Each algorithm in 2-3 lines)** | | Insertion Sort | In computer science the insertion sort is categorized in the comparison sorting algorithm. Time Complexity of the insertion sort is O (n). Insertion sort is the method that return a completed sorted array by selecting one element per loop | | Selection Sort | In computer science the selection sort is categorized in the comparison sorting algorithm. Time complexity of the election sort is O (n^2).Selection sort is not better than the Insertion sort. It will always return the sorted elements store in the list | | Merge Sort | In computer science the merge sort is categorized in the comparison sorting algorithm. Time complexity of the merge sort is n-log-n. In the merge sort we will get the sorted lit by dividing the unsorted array in to halves and then merging it again to get the sorted array | | Bubble Sort | In computer science the bubble sort is categorized in the comparison sorting algorithm. Time complexity taken by the bubble sort is O (n^2). In the bubble sort algorithm we will get a complete sorted list at the end of the loop by compare each element of the list | | Quick Sort | In computer science the quick sort is categorized in the comparison sorting algorithm. Time complexity by the quick sort is O (n) and it is not better than the merge sort. Merge sort is the best sorting algorithm.Quick sort is a popular sorting algorithm that is often faster in practice compared to other sorting algorithms. It utilizes a divide-and-conquer strategy to quickly sort data items by dividing a large array into two smaller arrays. | | |  |  | | --- | --- | | Hybrid Sort | A Hybrid Algorithm is an algorithm that combines two or more other algorithms that solve the same problem, either choosing one (depending on the data) or switching between them throughout the algorithm. | | Time Sort | Time sort is a hybrid stable sorting algorithm, derived from merge sort and insertion sort, designed to perform well on many kinds of real-world data. The algorithm finds subsequences of the data that are already ordered (runs) and uses them to sort the remainder more efficiently. | | Heap Sort | Heap sort is a comparison-based sorting technique based on Binary Heap data structure. It is similar to selection sort where we first find the minimum element and place the minimum element at the beginning. We repeat the same process for the remaining elements. | | Shell Sort | Shell sort is an optimization of insertion sort that allows the exchange of items that are far apart. The idea is to arrange the list of elements so that, starting anywhere, taking every element produces a sorted list. Such a list is said to be h-sorted. | | Tree Sort | A tree sort is a sort algorithm that builds a binary search tree from the elements to be sorted, and then traverses the tree (in-order) so that the elements come out in sorted order. Its typical use is sorting elements online: after each insertion, the set of elements seen so far is available in sorted order. | | Radix Sort | Radix sort is an integer sorting algorithm that sorts data with integer keys by grouping the keys by individual digits that share the same significant position and value (place value). Radix sort uses counting sort as a subroutine to sort an array of numbers. | | Cube Sort | Cube sort is a parallel sorting algorithm that builds a self-balancing multi-dimensional array from the keys to be sorted. As the axes are of similar length the structure resembles a cube. After each key is inserted, the cube can be rapidly converted to an array. | | | | | |
| Searching Algorithms | **Linear Search:**  A linear search is the simplest method of searching a data set. Starting at the beginning of the data set, each item of data is examined until a match is made. Once the item is found, the search ends. If there is no match, the algorithm must deal with this.  **Binary Search:**  Binary search is an efficient algorithm for finding an item from a sorted list of items. It works by repeatedly dividing in half the portion of the list that could contain the item until you have narrowed down the possible locations to just one.  **Jump Search:**  Like Binary Search, Jump Search is a searching algorithm for sorted arrays. The basic idea is to check fewer elements (than linear search) by jumping ahead by fixed steps or skipping some elements in place of searching all elements. |
| Searching Filters for each data type | **For data type of Integer:**  Numbers are compared and a smaller number would be decided.  **For data type of String:**  Strings are compared with each other and a character of smaller ASCII code would be decided. |
| Multi-Level Sorting | Multi-level sorting would be inserted in this project to sort data for particular regions or for particularly filtering out the data. |
| Any other features | **Sorting Label:**  During sorting time of any column, a label that display the progress of the data being sorted.  **URL Option:**  We Will try to take a URL as input and then the data will be scrapped from that particular URL given by the user. |
| ***Interfaces for your project*** |  |
| *[Draw layouts in the pencil tool. For each picture of the UI, provide the following table.]*   |  |  |  | | --- | --- | --- | | UI Component Name | Type of UI component | Purpose of UI Component/Other details | | Pause-Button | Button | It will Pause the process of scrapping | | Stop-Button | Button | It will Stop the process of scrapping | | Start-Button | Button | It will again start the process of scrapping once paused or stopped. | | Annotation | Text Area | It Just Gives the Basic Introduction Of Our Software. It’s Basically a Welcoming text on the dashboard. | | Progress-Bar | Progress-Bar | To show the visual representation of the data being scraped with time in Milliseconds. | | Combo-Selector | Combo Box | To Ask The User various sorting options. I-e ascending, descending etc. And to select the sorting algorithm to sort data by selected column. | | Data-Table(It’s not included right now in the GUI as the entities may change) | Table | To show all the scrapped data | | Text area | Input | To take the input from the user I-e the username and password | | Buttons | Button | To manage various tasks and move from one form to another. | | Dash-Board Window | Area | In the dashboard we will have the facility to choose what we want I-e sorting, searching etc. It’s basically a main menu. | | Search and sorting Screens | To-do list | In this part of GUI we will search the scrapped data on the basis of the different attributes like color and brands of the product. On searching the list will provide the selected elements from the scrapped data. So searching the data is the main process of the project. And we can also sort our data | | |
| **UI Display:**  C:\Users\Hp\Pictures\Screenshots\Screenshot (540).png  C:\Users\Hp\Pictures\Screenshots\Screenshot (551).png  C:\Users\Hp\Pictures\Screenshots\Screenshot (542).pngC:\Users\Hp\Pictures\Screenshots\Screenshot (544).pngC:\Users\Hp\Pictures\Screenshots\Screenshot (547).pngC:\Users\Hp\Pictures\Screenshots\Screenshot (549).png | |