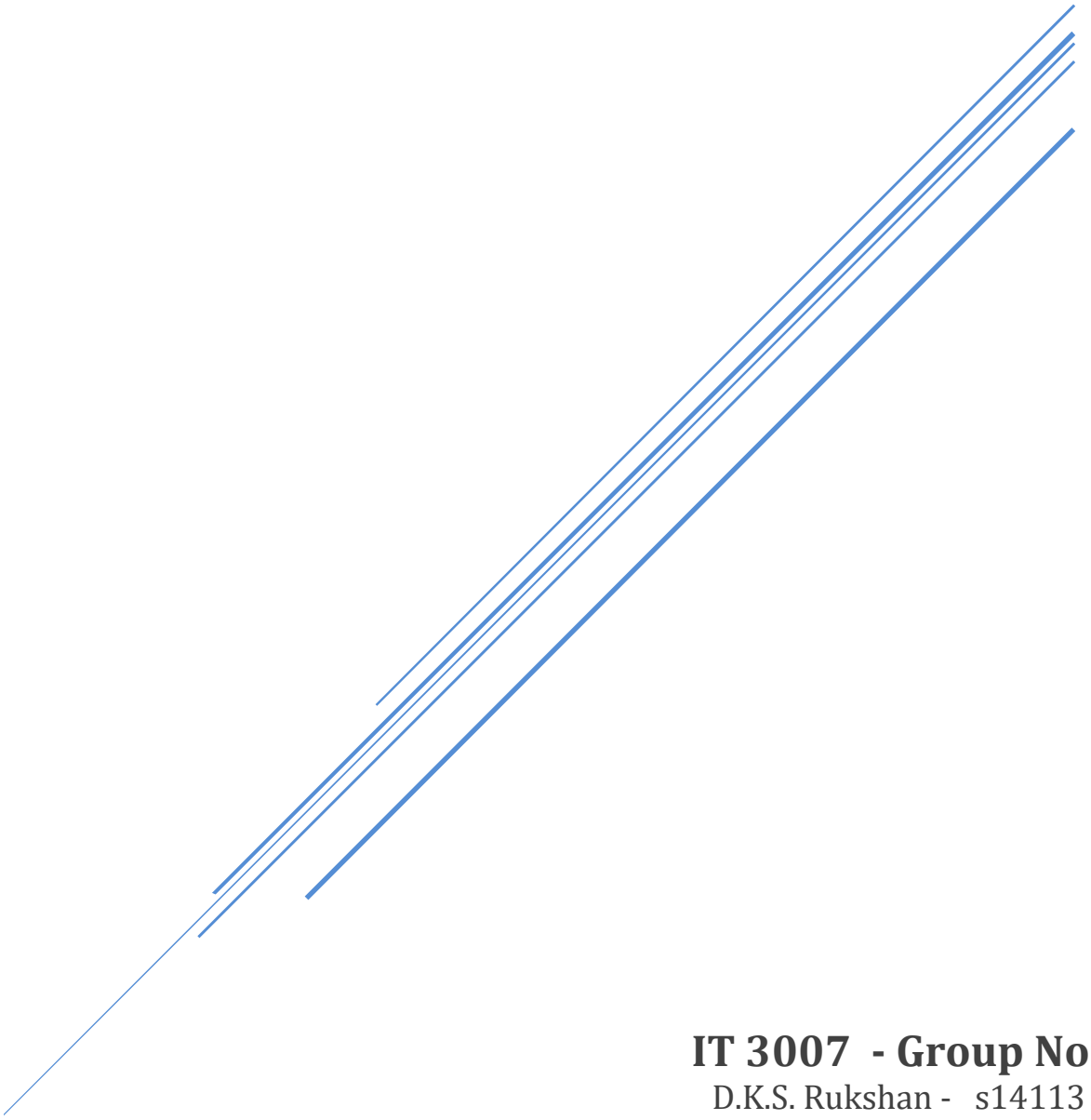
APPLICATION USING B-TREE, MERGE SORT & BINARY SEARCH

Documentation



IT 3007 - Group No: 7

D.K.S. Rukshan - s14113 (L)
A.G.K.M. Ariyarathne - s14043
H.G.S. Sarika - s14119
M.H.D. Maduraarachchi – s14089
L.L.S. Prematilaka - s14105

Application using B- tree, Merge sort and Binary Search

Here are brief explanations about the algorithms used in the application.

B - Tree :

In computer science, a B-tree is a self-balancing tree data structure that maintains sorted data and allows searches, sequential access, insertions, and deletions in logarithmic time. The B-tree generalizes the binary search tree, allowing for nodes with more than two children. Unlike other self-balancing binary search trees, the B-tree is well suited for storage systems that read and write relatively large blocks of data, such as disks. It is commonly used in databases and file systems. Height of the B -tree is h = O(logn) where n is the number of the keys stored in the tree

Algorithm	Average	Worst case
Search	O(log *n*)	O(log *n*)
Insert	O(log *n*)	O(log *n*)
Delete	O(log *n*)	O(log *n*)

Merge Sort :

Merge Sort is a sorting algorithm, which is commonly used in computer science. Merge Sort is a divide and conquer algorithm. It works by recursively breaking down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem. So, Merge Sort first divides the array into equal halves and then combines them in a sorted manner.

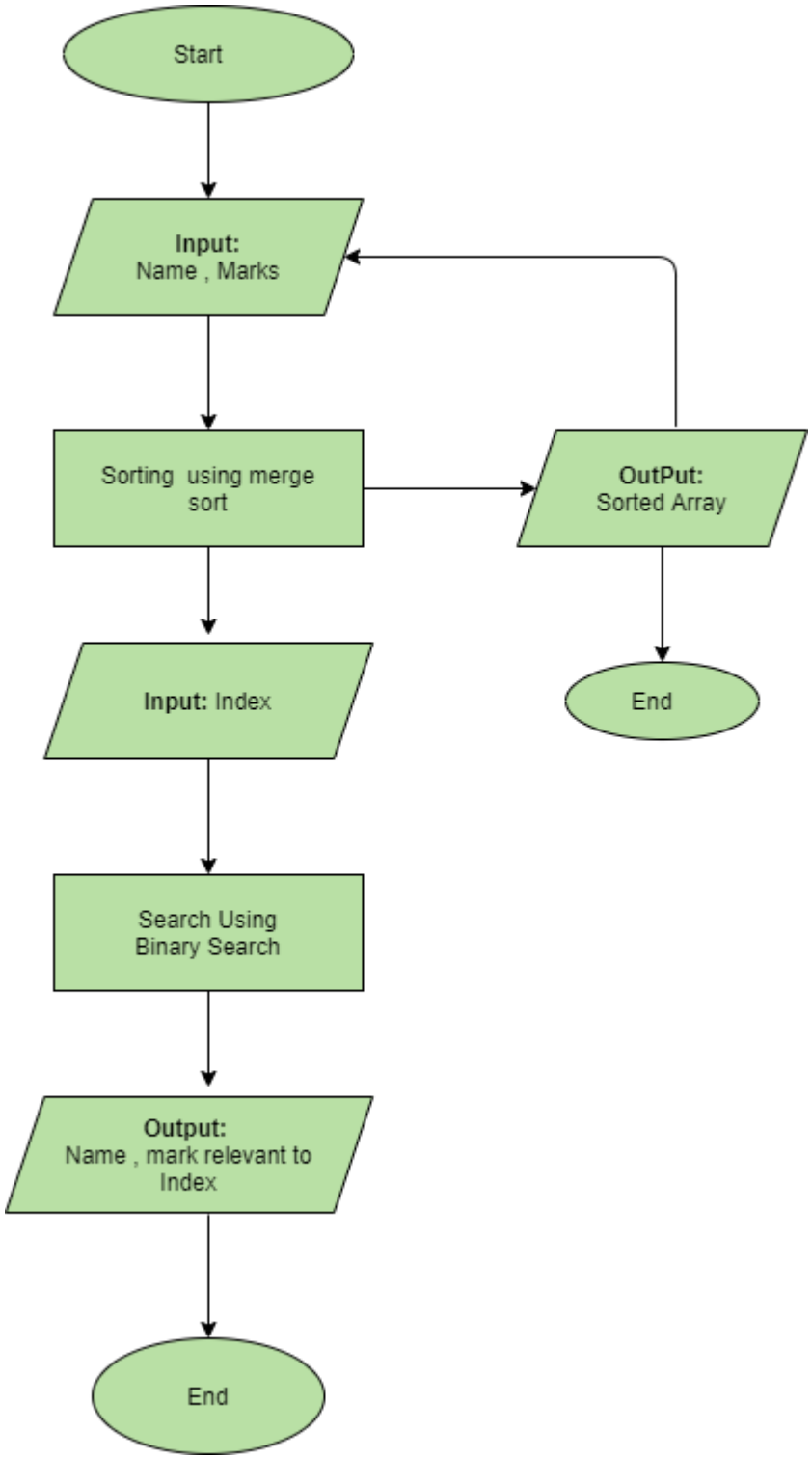
Overall time complexity	O(nlog *n*)
Insert	O(log *n*)
Average time complexity	O(n^2)

Binary Search :

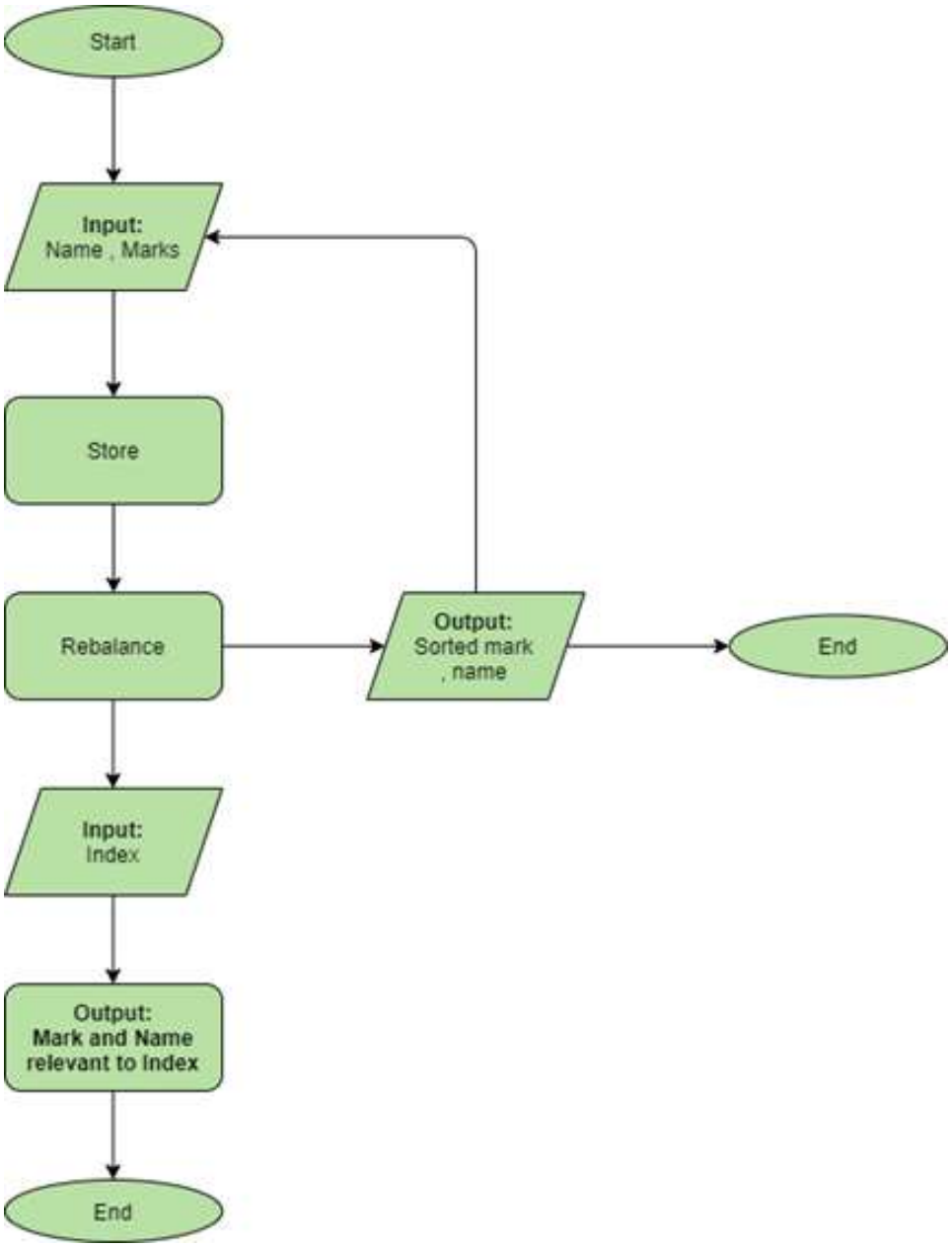
Binary search is a search algorithm that finds the position of a target value within a sorted array. This search algorithm works on the principle of divide and conquer. Binary search compares the target value to the middle element of the array. If they are not equal, the half in which the target cannot lie is eliminated and the search continues the remaining half, again taking the middle element to compare to the target value and repeating this until the target value is found. If the search ends with the remaining half being empty, the target is not in the array.

Algorithm	Average	Worst case
Search	**O(log *n*)**	**O(*n*)**
Insert	**O(log *n*)**	**O(*n*)**
Delete	**O(log *n*)**	**O(*n*)**

Process of Merge sort & Binary search in the application

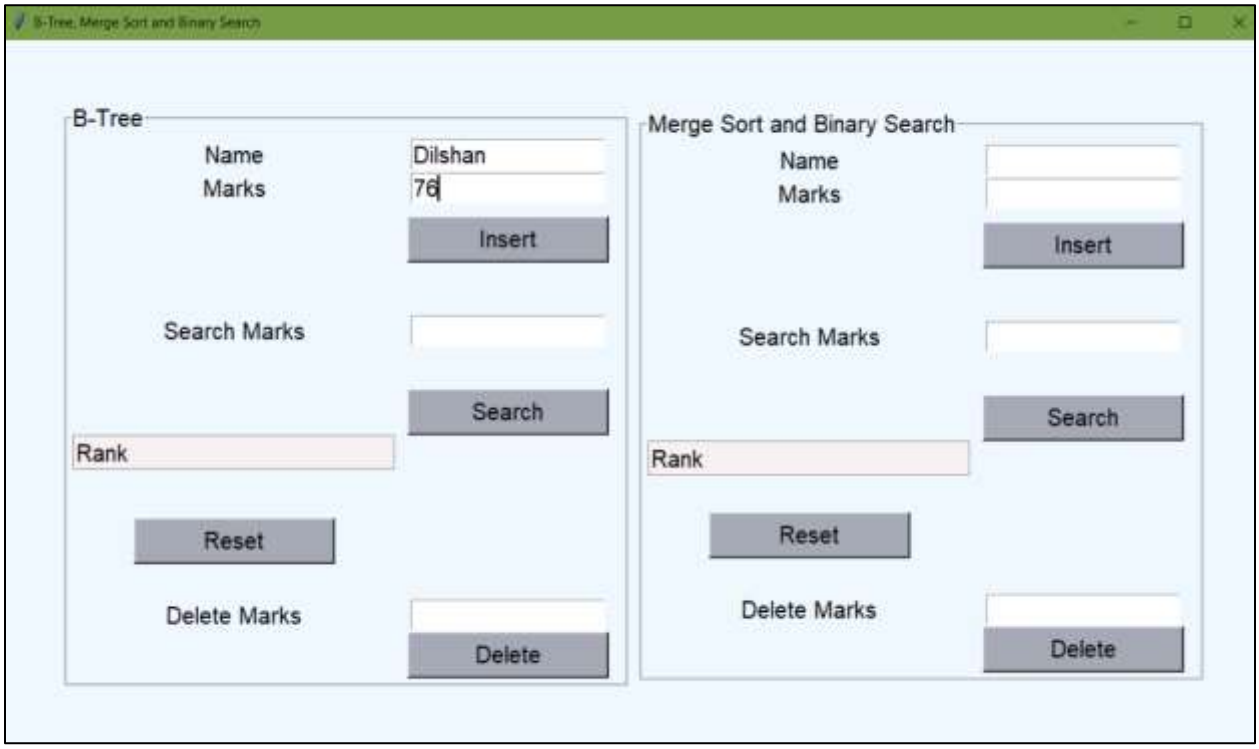


Process of B - Tree in the application



Application :

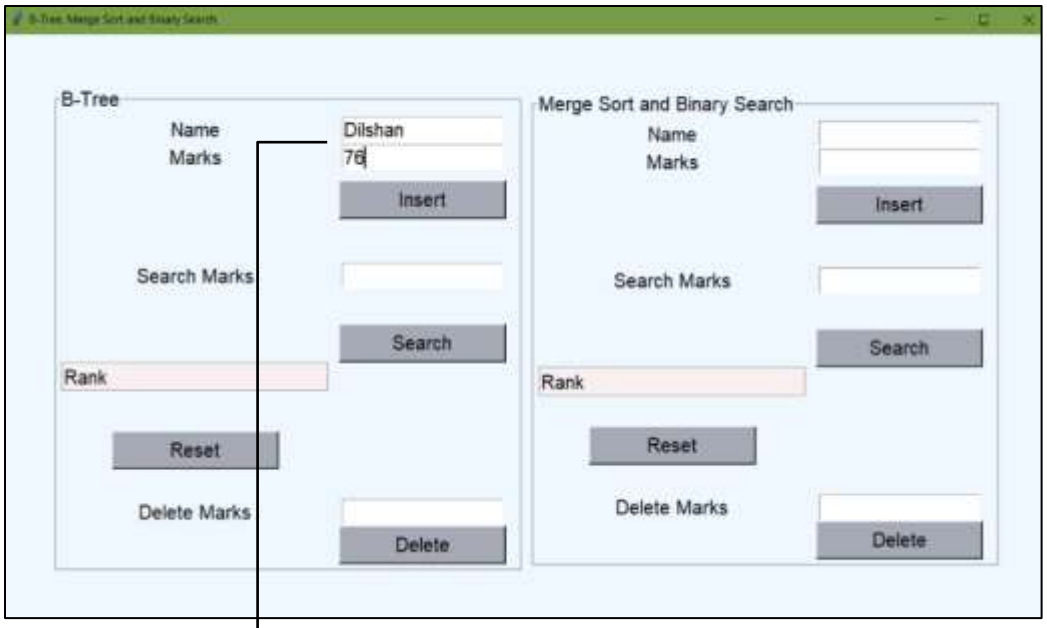
The program implemented contains a simple application that maintains a table of names and marks for a particular subject that is ranked according to the marks they have taken.



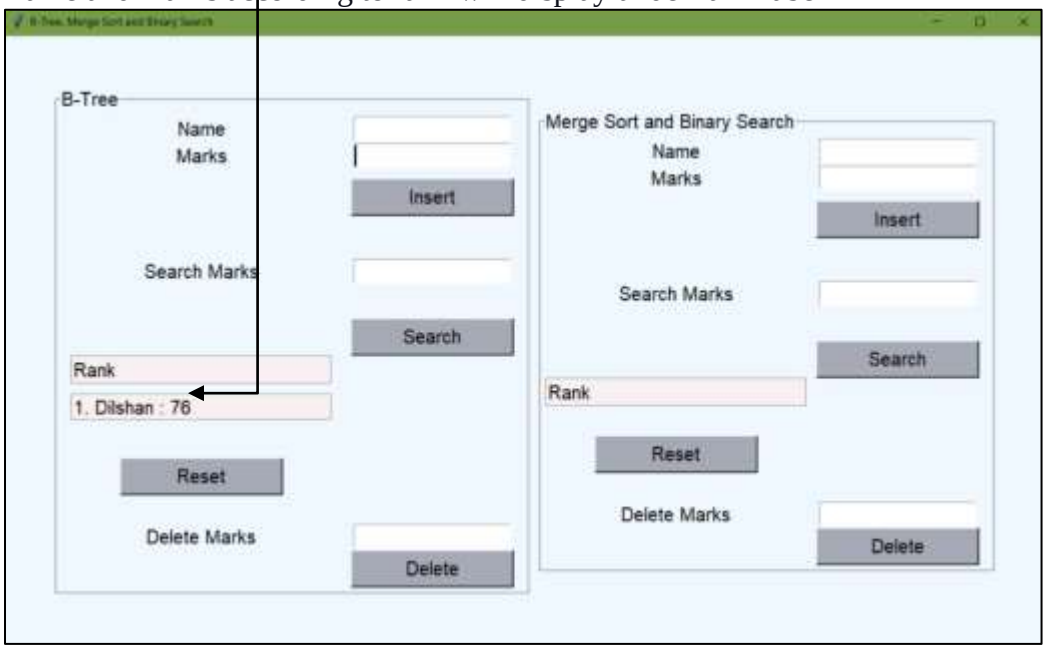
The functionality of the B Tree application as follows.

- Insert the **name** of a person and their **marks** for a particular subject.

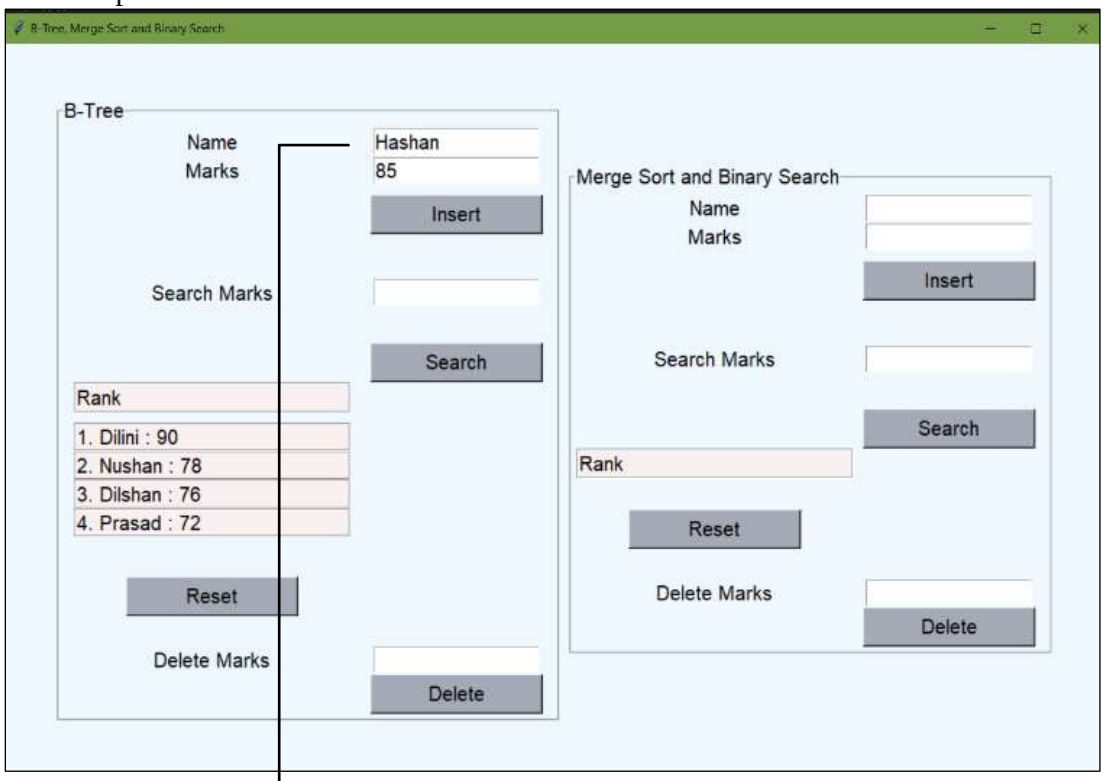
Enter Name and Marks then click insert



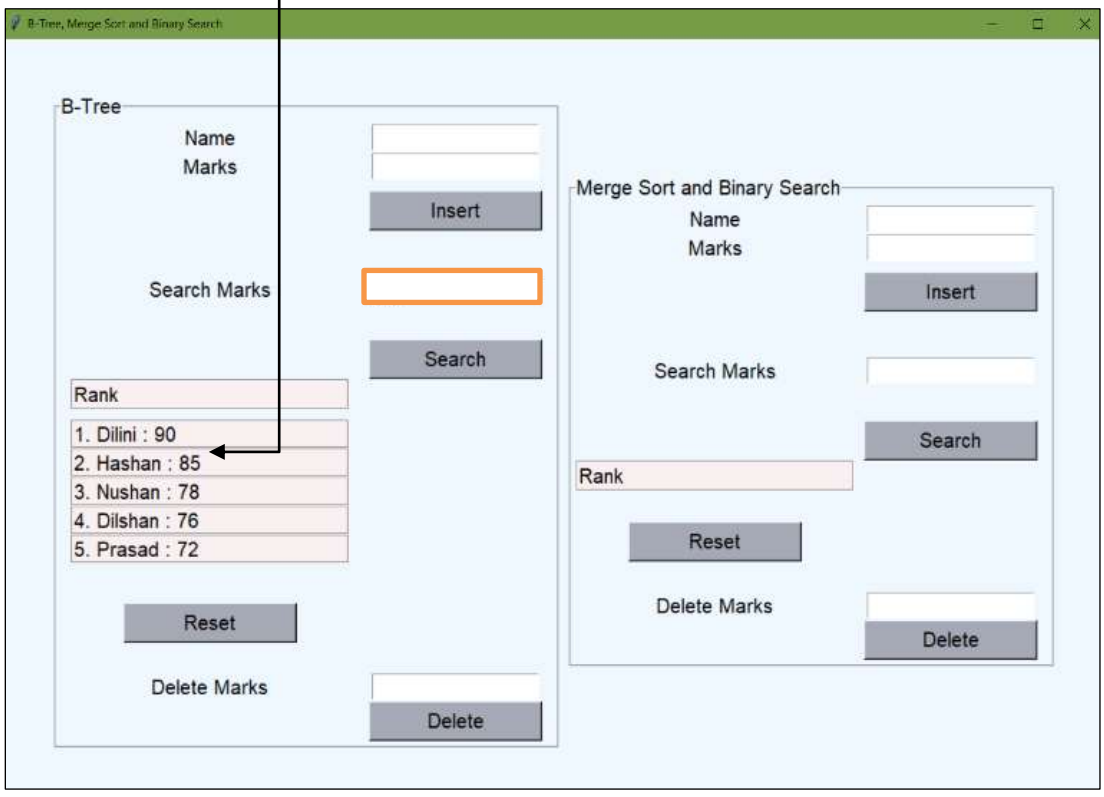
Name and marks according to rank will display under rank label



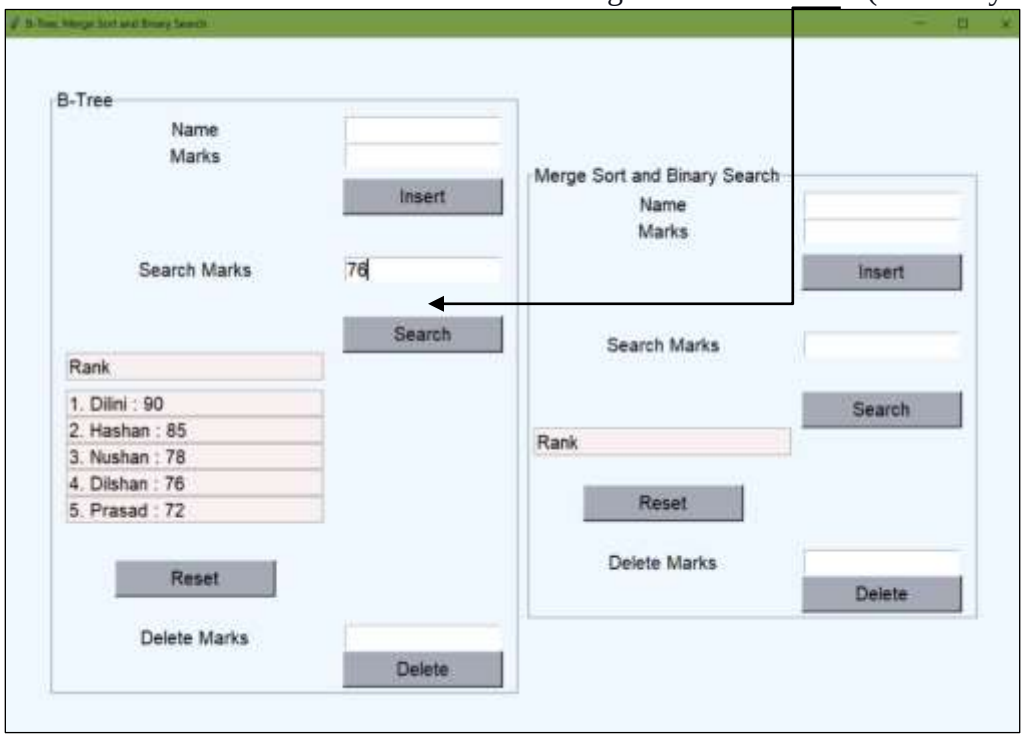
- Once the value is inserted the table is updated and the name and marks field are added to the correct place of the rank table.



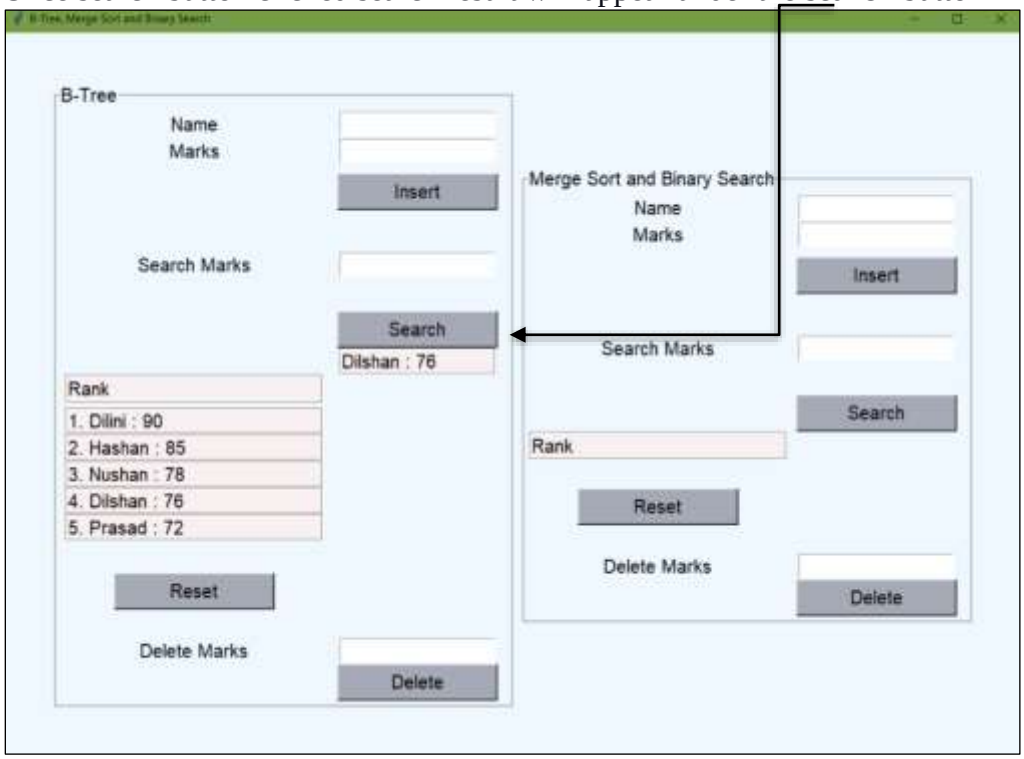
Field adds to the correct rank



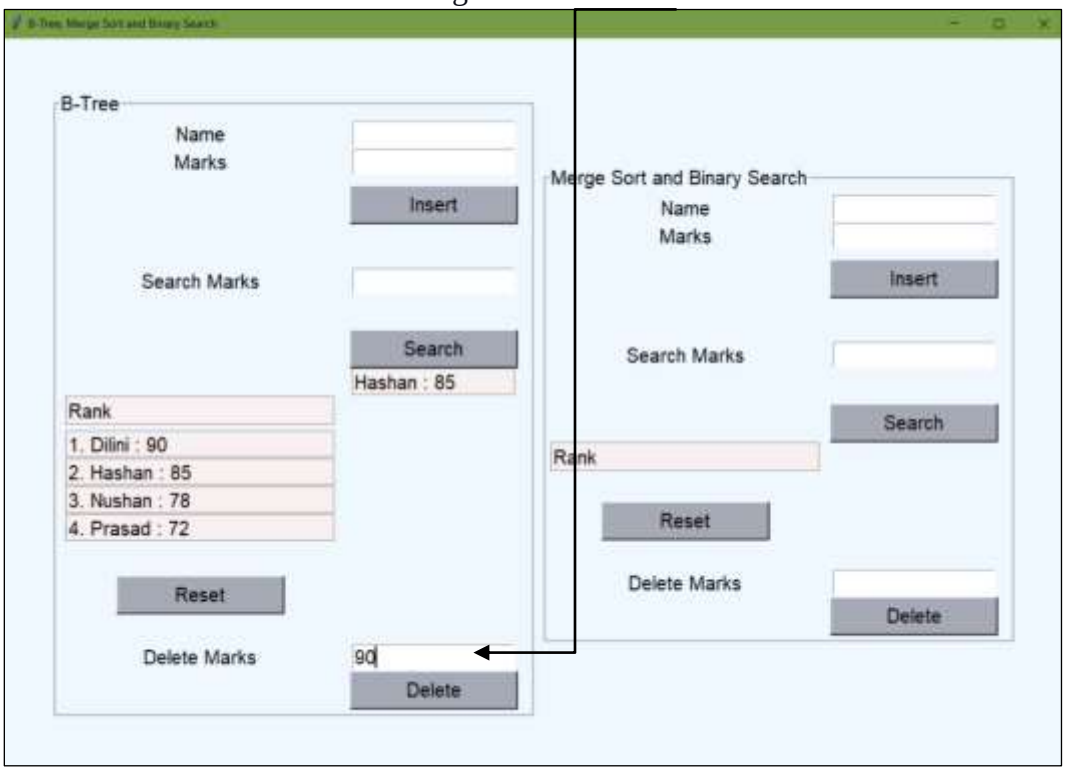
- To search the **marks** values and find a field for a given **marks** value. (Search by marks)



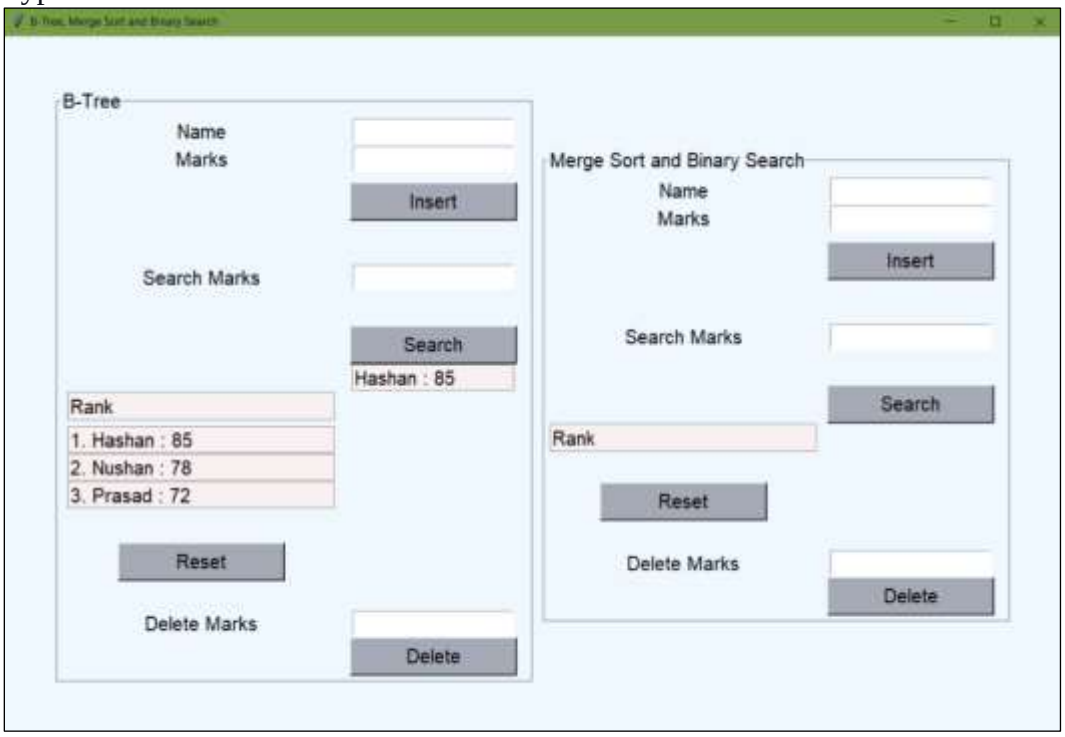
Once search button clicked search result will appear under the **search** button



- To delete a field from the table when given the **marks** value.

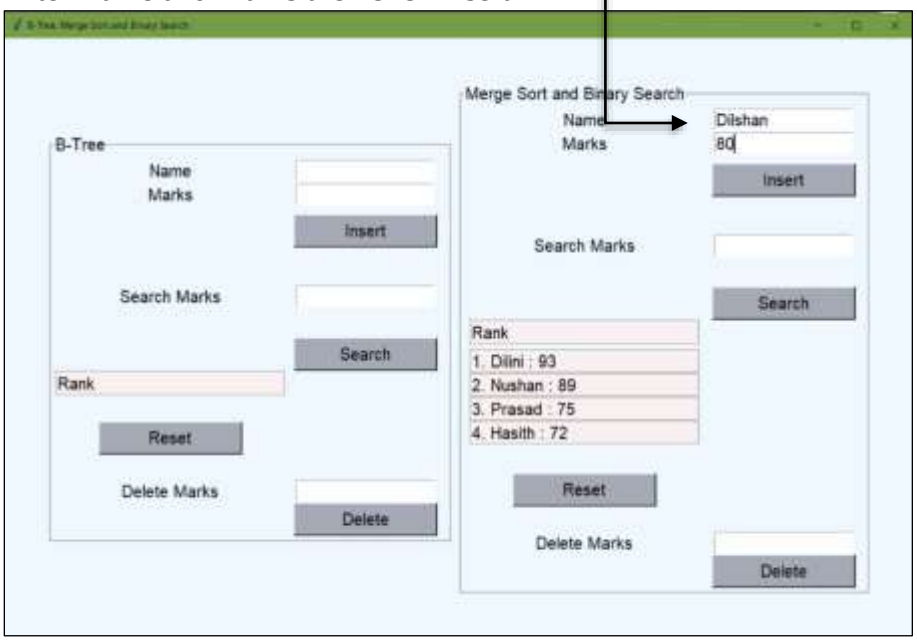


Type the mark value to be deleted. Click **Delete**. Marks will be deleted.

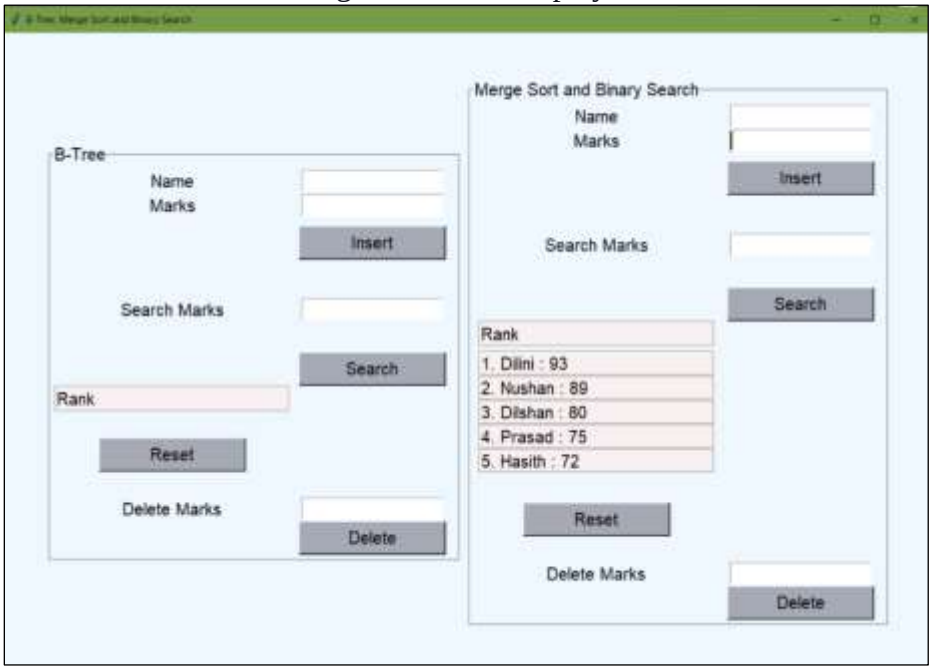


The functionality of the Merge Sort and Binary Search application as follows.

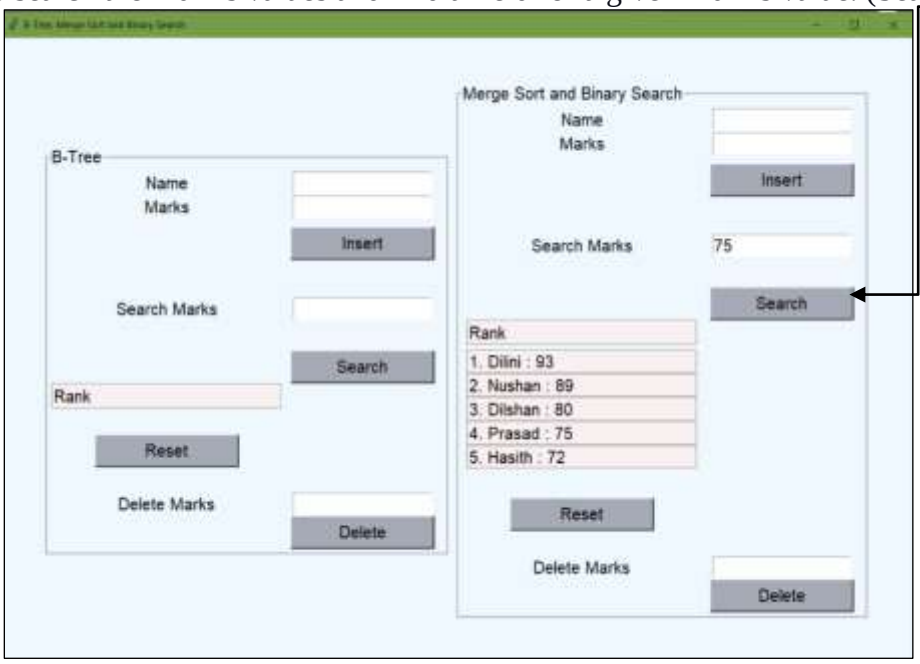
- Insert the **name** of a person and their **marks** for a particular subject.
  Enter Name and Marks then click insert



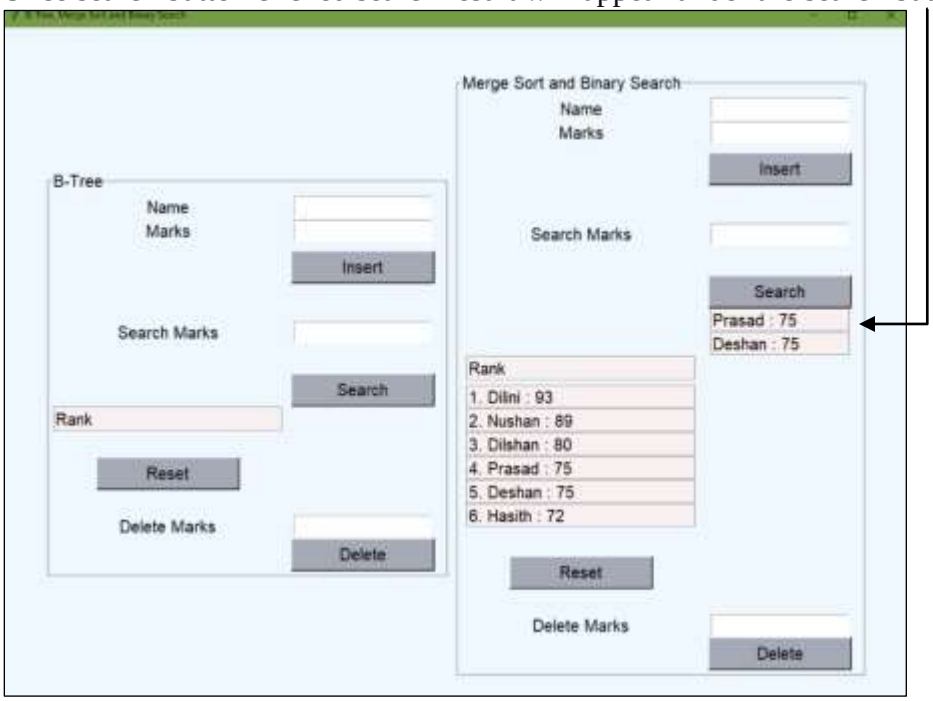
Name and marks according to rank will display under **rank** label



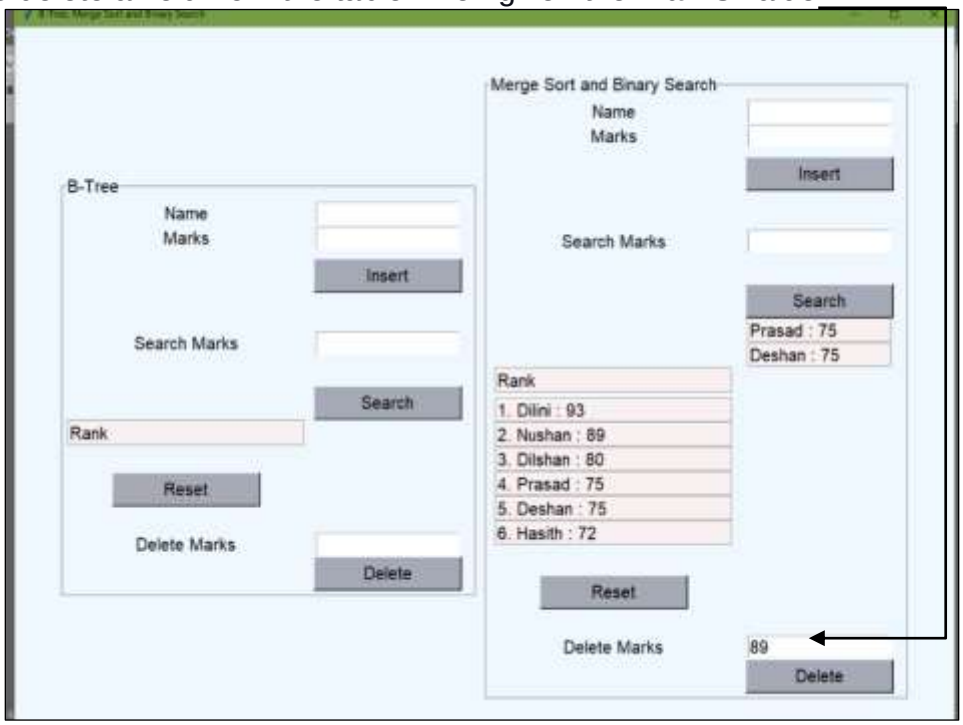
- To search the **marks** values and find a field for a given **marks** value. (Search by marks)



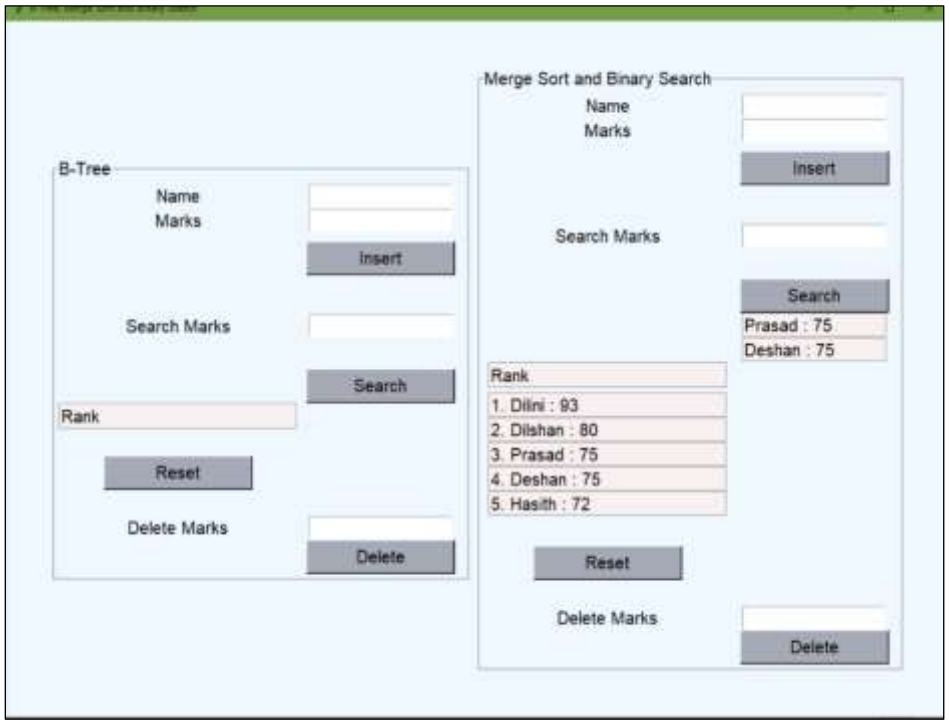
Once search button clicked search result will appear under the search button



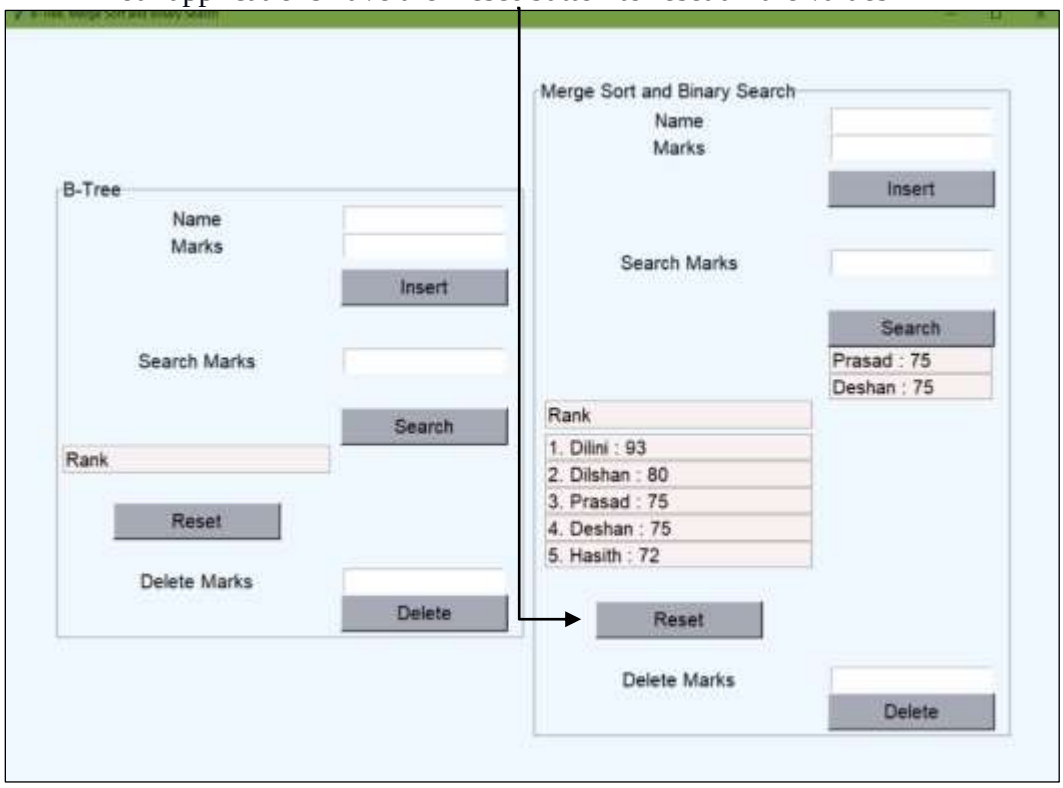
- To delete a field from the table when given the **marks** value.



Type the mark value to be deleted. Click **Delete**
The Field will be deleted.



- Both applications have the **Reset** button to reset all the values



Once Clicked each reset button resets the sorted values to be empty