“Score Record”

Group Members:

Hira Tahir (20K-0374)

Naima Jamal (20K-0404)

Lailumah Hiba Nadeem (20K-0431)

Overview:

Our main goal was to develop a program that could take data from a file and store it in a hash table and through linear chaining we are storing the data for a particular column.
If the user wants to find out who did the submission at a particular date, they can access the file through dates. They can also find out who submitted it late.The second challenge was to maintain data of millions of users and to search it efficiently.

Project Details:

How does it work?

Our program reads the data from the csv file, Submissions.txt, then stores it into an object of vector type, and this vector contains all the records. It pops the menu on the screen, and from here the user can select any one of the 4 options. If the user enters anything other than these options, then it will again ask to select the valid option.
On pressing 1, it will ask you the date of submission from the records, and then after storing these records into the hash table then after searching, it will display the records of that particular date.

On pressing 2, it will store all the values into the hash table and then after searching, it will show all the records of late submission.

On pressing 3, it will first ask you totenter the number of records you want to store into the hashtable and then displays all the records you want to display.

On pressing 4, if the user does not want to enter or search any record then he can exit.

Before performing any of the above actions , it will first clear the screen.

Benefits:

- Searching is made easier and faster.
- Fast retrieval of information.
- Large data sets can be handled

This program uses:

- Hashing
- Hash Tables
- Linear chaining
- Classes/Objects
- Filing
- Vectors

Libraries:

- #include <iostream>
- #include <string>
- #include <fstream>
- #include <sstream>
- #include <vector>
- #include <math.h> // floor()
- #include <iostream>
- #include <bits/stdc++.h>

Resources:

Dataset taken form Kaggle.com