WEBKIOSK PROTOTYPE

December – 2020

Submitted by: Utkarsh Garg 19103108

Tanansh Ahuja 19103101

Bhavya Kohli 19103098

Submitted in partial fulfilment of the degree of Bachelor of Technology

Department of Computer science and Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

Problem statement

In today's world managing data has become a very difficult task. And managing data is the most important task with which all the big tach giants are dealing with right now. Through this project we tried to manage teachers and student's data and have also tried to manage some connectivity between them.

Introduction

This project is based on the institutional web kiosk including teachers and student's data. As an institution has different people having different access on data this project will focus on that and will try to systematically organise the data of teachers and students under their respective teachers.

This project deals with the data of teachers and students and their relations with each other. All the details of teachers like their id's, the batches they teach, the courses they teach, their salaries, their performances, their posts and details about the students like their roll number, courses they have opted, teachers they are associated with, their academic performances, their marks, etc.

The project can be used by a single teacher to a small institute containing a few teachers and students. The data will be monitored by the admin and only he will have the total control over all the data.

The teachers can read their data while read and writing their respective student's data. The student will be able to read his data only.

List of data structure used

1. **Vector:** One of the most basic and commonly used data structure, for temporary storage of data for quick approval before finally saving it in a file.
2. **Hashing using maps:** To read and save the data of all student's data that a teacher needs to see, for easy and fast access.

3. **Searching:** Searching techniques to look for the specific data the teacher needs to see of a specific student, or when the admin to see records of a specific teacher or student.

4. **Sorting:** Will be constantly be used will writing data in file for fast retrieval in later stages.

Something we will try to implement

Other than using classes and object, inheritance, operator overloading (to compare between two students or teachers), we will try to implement file handling by connecting our project with MySQL or Excel to make something new.

Environment of project: CodeBlocks

Detailed design

Class 1 : Extra

- *void printmap(map<string,string> userdata)* : prints the map with key value that is passed inside the function.
- *string passwrd()* : helps show '*' instead of your password when you enter password in system.
- *string timestamp()* : generates the timestamp of system time in date, month and year format;
- *string generate_password() :* generates a random password of 10 characters and return back to the desired function.
- *string enrollment_generator() :* generates the Enrollment number of student.
- *string id_generator() :* Generates the desired teacher's ID according to the chronological order.

Class 2 : Auth [Derived class of class 1 public inherited**]**

private:[Variables**]**

- *vector< map<string,string> > userlist* : will save the data of the student of a specific teacher when she will login and will be accessible by all functions.
- *map<string,string> loginuser :* When the user will login, his/her critical data that needs to be used multiple times will be stored here.
- *string temp1,temp2,temp3* : will hold values that can't be passed between functions but still may come in handy while calculation or searching;
- *vector<string> name , enroll , password , role:* These 4 vectors will hold all the values from registry table, for faster access for future usage.

Public [Functions]

- *void appendinregister(map<string,string> userdata)* : When the register process is done, this function will save the data in the map inside the registery.csv file.
- *void apnd_in_student(string name,string enrol):* if registration of a student is being done, then student's data will be saved in the student table.
- *void apnd_in_teacher(string name, string id)* : if registration of a teacher is being done, then teachers's data will be saved in the teacher's table.
- *int registeruser() :* The function which is called if the admin wants to register a new user.
- *int checkcredential(string enrol, string pasw, string r)* : will take value and search the 4 vectors declared above and match the credentials of student, teacher. Return 0 if match is found, 1 in any other case.
- *int checkcredentialadmin(string enrol, string pasw) :* will specifically find admin's data from the vectors.

- *int readregistry() :* Will run the moment the code runs first time, and will assign the 4 vectors their respective values;
- *int updateadminpassword() :* Changes the admin’s password.
- *void updateadminid()* : Changes the admin’s userid.
- *void get_data(string id, string r):* Print the data of 1 student or all student, or, 1 teacher or all teachers given the parametrs and conditions.
- *void viewdata():* The function that asks who’s data the admin wants to see.
- *void Updatemarks():* updates the mark of the student in the subject in which the teacher likes.
- *void entertime(string r):* This function enters the value in logintime.csv table, and only works when a user successfully logs in, and note the time in the file.

class 3 : box [public inherited class Auth]

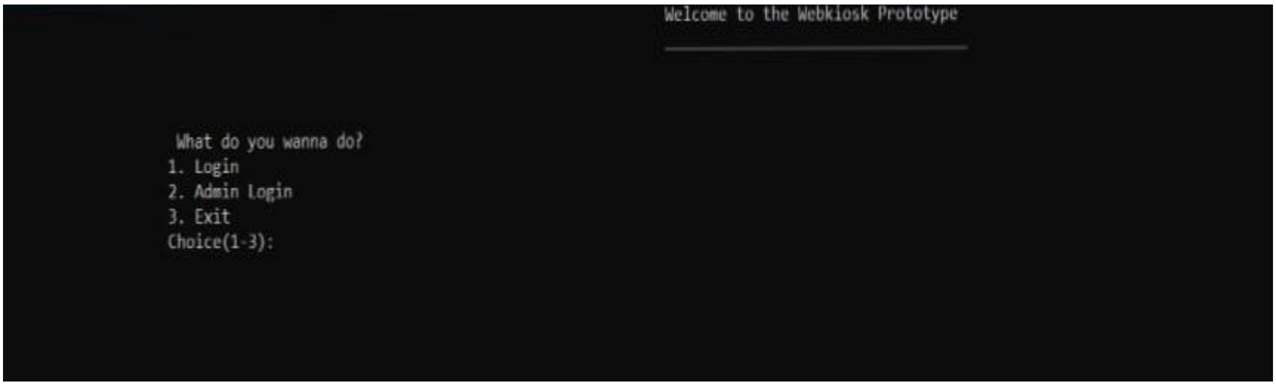
Public

- *void frontpage() :* This is the first function that is being called. The flow of control starts from here.
- *void Login() :* Asks the user his/her credentials and role(student or teacher), and cheack those credentials using other function.
- *void Studentpage():* Creates the interface a student would see if he is logged in.
- *void Teacherpage():* Creates the interface the teacher would see if she is logged in the system.
- *int Adminlogin() :* Takes admin credentials and check them and give admin the acces of Adminpage();
- *void Adminpage()* : Creates the interface of what admin will be able to see.

The above mentioned were just the functions. In this page we there are screenshots of the output to describe more about the functions in a little brief .

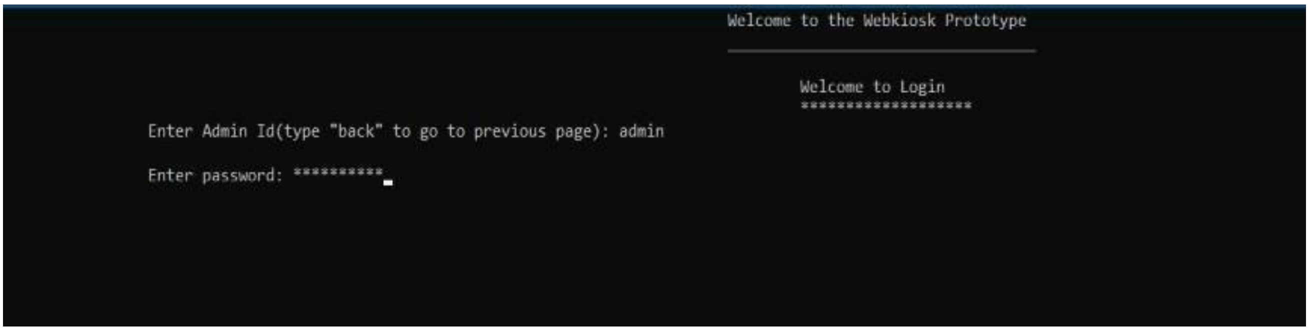
Implementations and results

1.) First page : void frontpage() is responsible for this output . it uses switch case and this pointer to let the user go to any page he wants depending upon the constraints



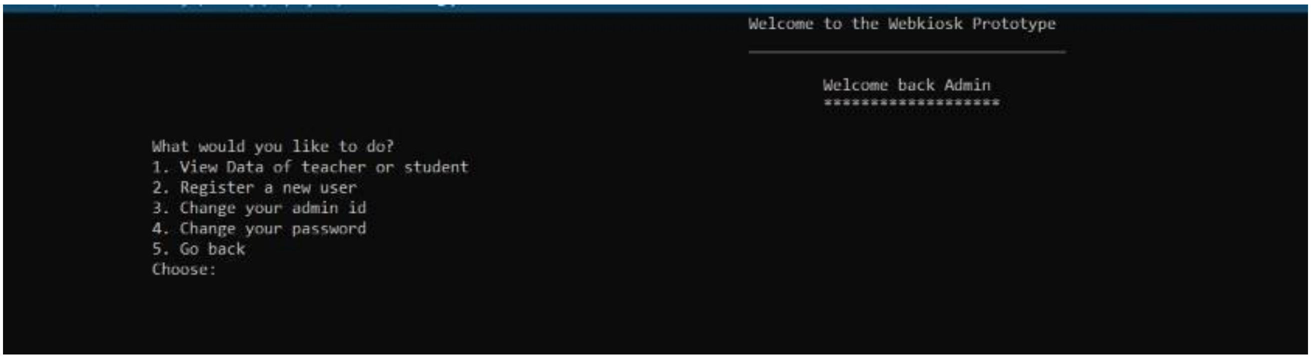
```
                                                   Welcome to the Webkiosk Prototype
                                                   _________________________________



           What do you wanna do?
          1. Login
          2. Admin Login
          3. Exit
          Choice(1-3):
```

2.) Second page : For option 2 the below shown output screen is the result . In this the admin has to enter his credentials .



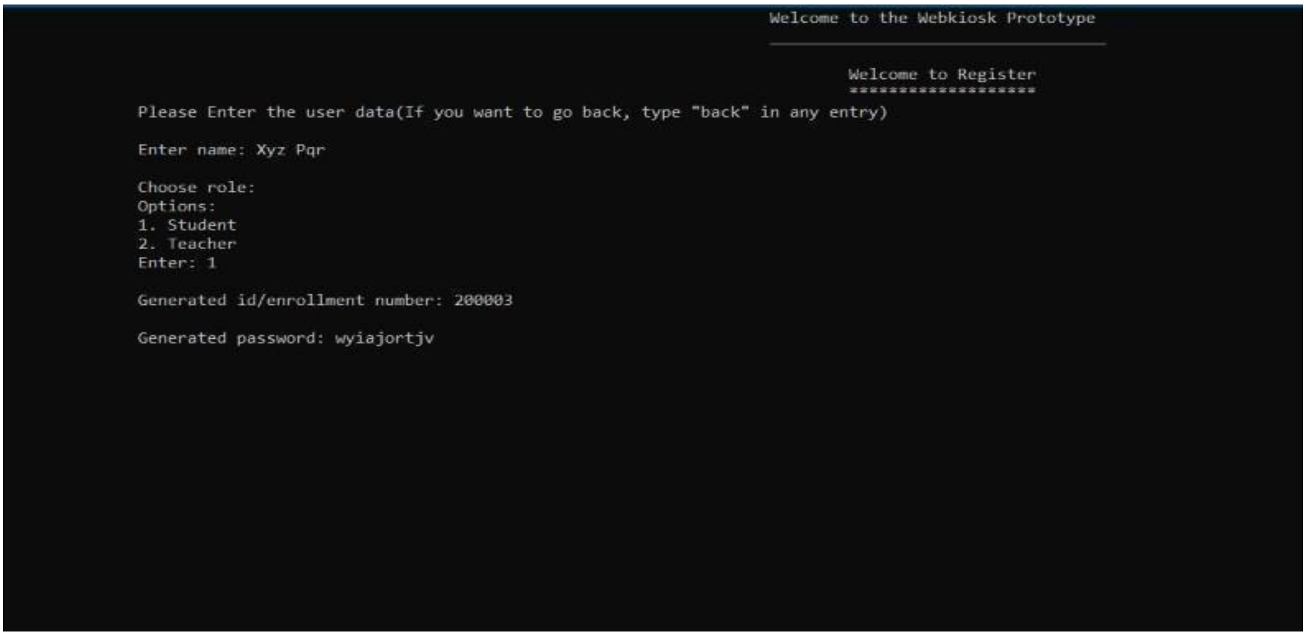
```
                                                       Welcome to the Webkiosk Prototype
                                                       _________________________________

                                                            Welcome to Login
                                                            ********************
          Enter Admin Id(type "back" to go to previous page): admin

          Enter password: **********_
```

3.) Third page : This page appears when the admins credentials matches with what entered in the database . if the credentials does not match it goes back to second page .



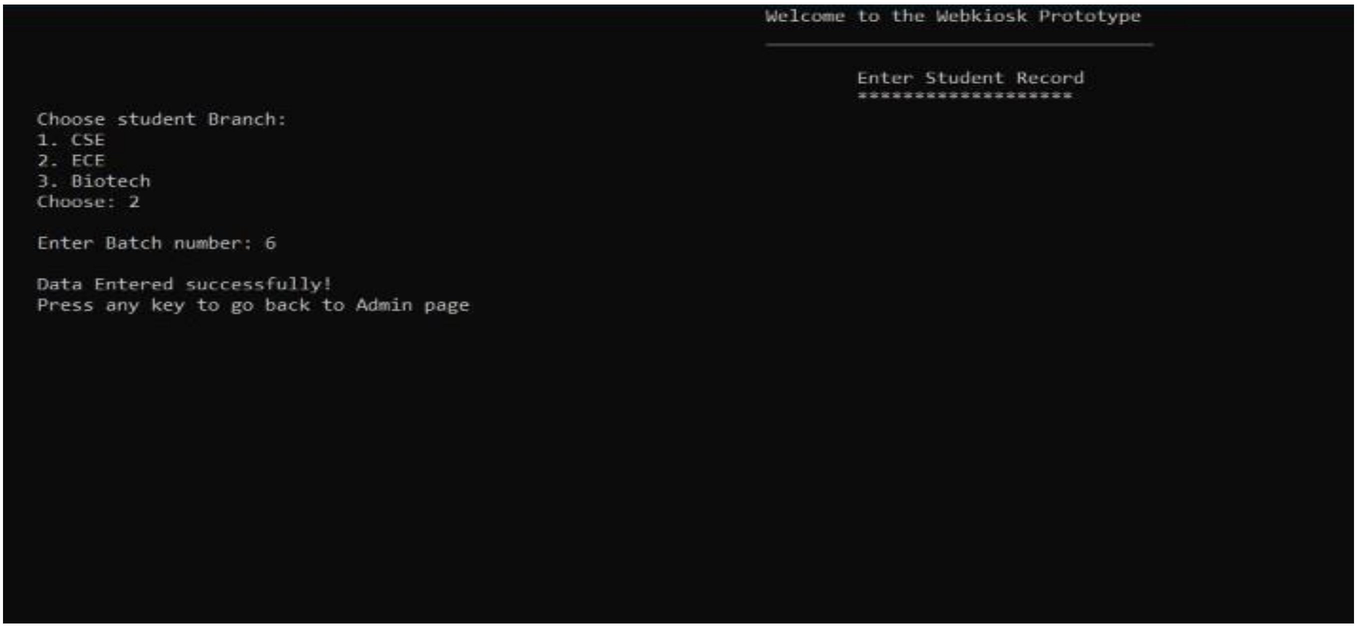
```
                                                            Welcome to the Webkiosk Prototype
                                                            _________________________________

                                                                  Welcome back Admin
                                                                  ******************

            What would you like to do?
            1. View Data of teacher or student
            2. Register a new user
            3. Change your admin id
            4. Change your password
            5. Go back
            Choose:
```

4.) Fourth page : This option opens when the admin wants to register the new user .



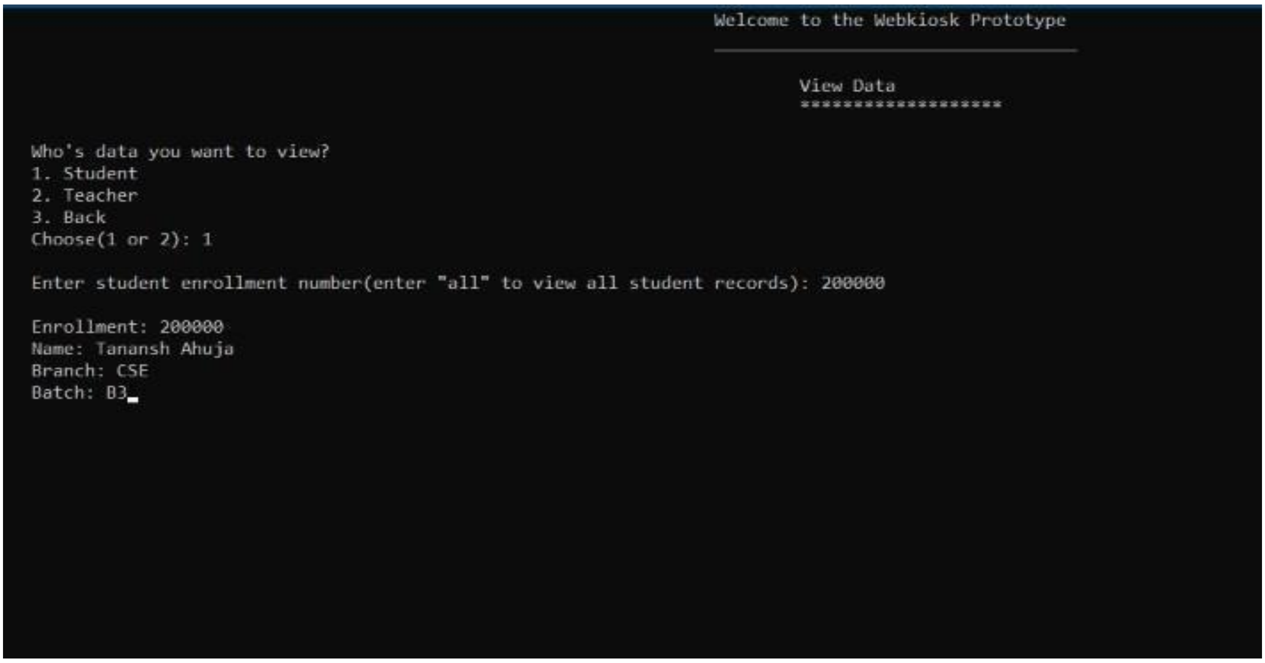
```
                                                              Welcome to the Webkiosk Prototype
                                                              _________________________________

                                                                    Welcome to Register
                                                                    *******************
          Please Enter the user data(If you want to go back, type "back" in any entry)

          Enter name: Xyz Pqr

          Choose role:
          Options:
          1. Student
          2. Teacher
          Enter: 1

          Generated id/enrollment number: 200003

          Generated password: wyiajortjv
```

5.) Fifth page : This option opens when students has to be registered . And in the previous page new enrol number and passwords are generated with the help of string generate_password() and string enrollment_generator() .



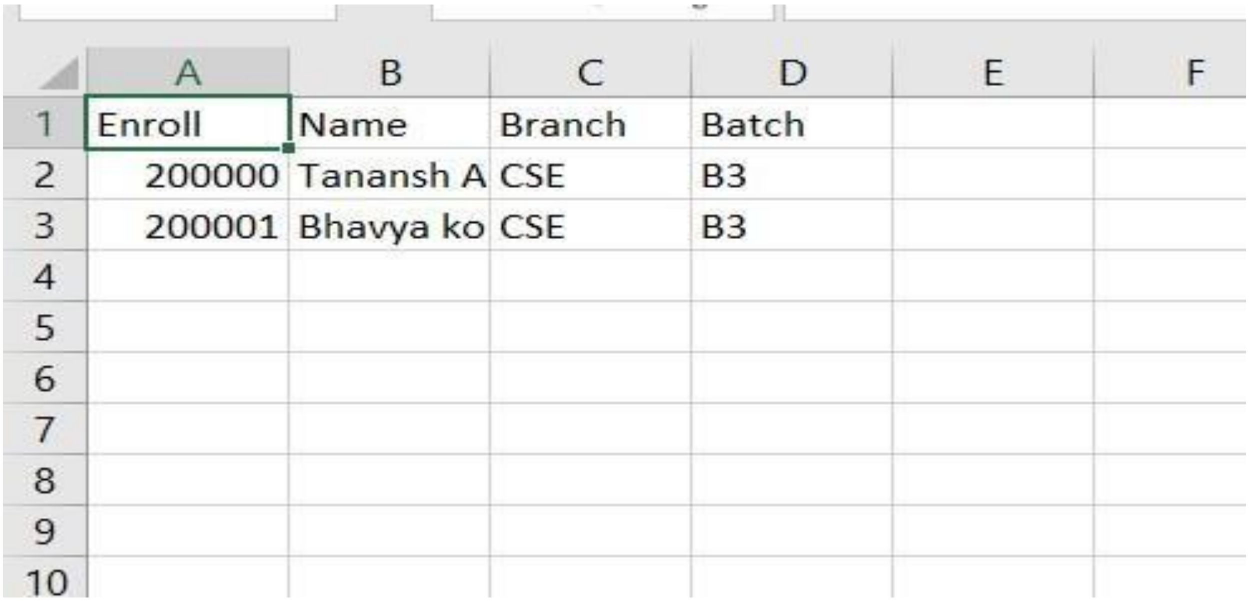
```
                                                  Welcome to the Webkiosk Prototype
                                                  __________________________________

                                                        Enter Student Record
                                                        ********************

  Choose student Branch:
  1. CSE
  2. ECE
  3. Biotech
  Choose: 2

  Enter Batch number: 6

  Data Entered successfully!
  Press any key to go back to Admin page
```

6.) Sixth page : When the student’s data is entered it gets stored in the excel file .



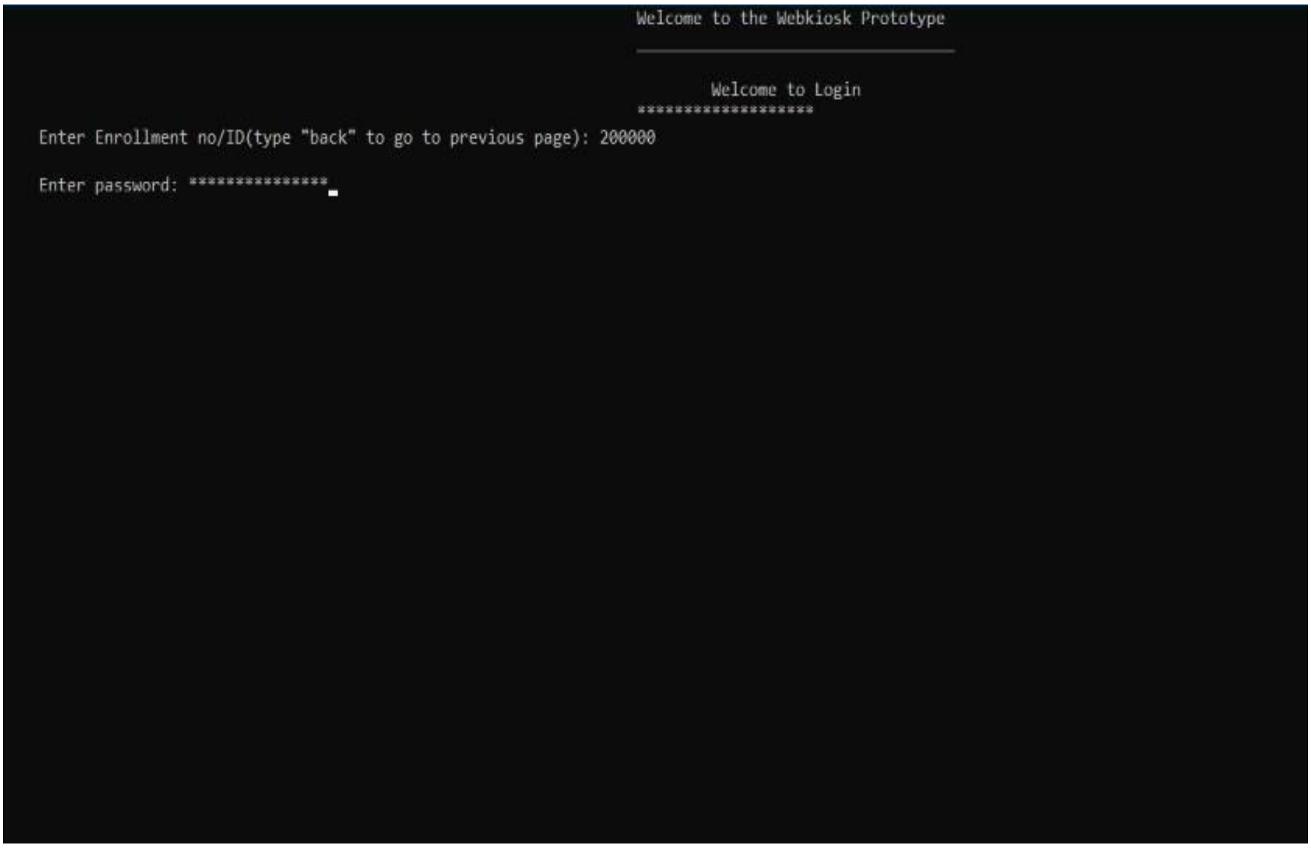
```
                                                  Welcome to the Webkiosk Prototype
                                                  __________________________________

                                                       View Data
                                                       ********************

  Who's data you want to view?
  1. Student
  2. Teacher
  3. Back
  Choose(1 or 2): 1

  Enter student enrollment number(enter "all" to view all student records): 200000

  Enrollment: 200000
  Name: Tanansh Ahuja
  Branch: CSE
  Batch: B3_
```

Corresponding excel file is :



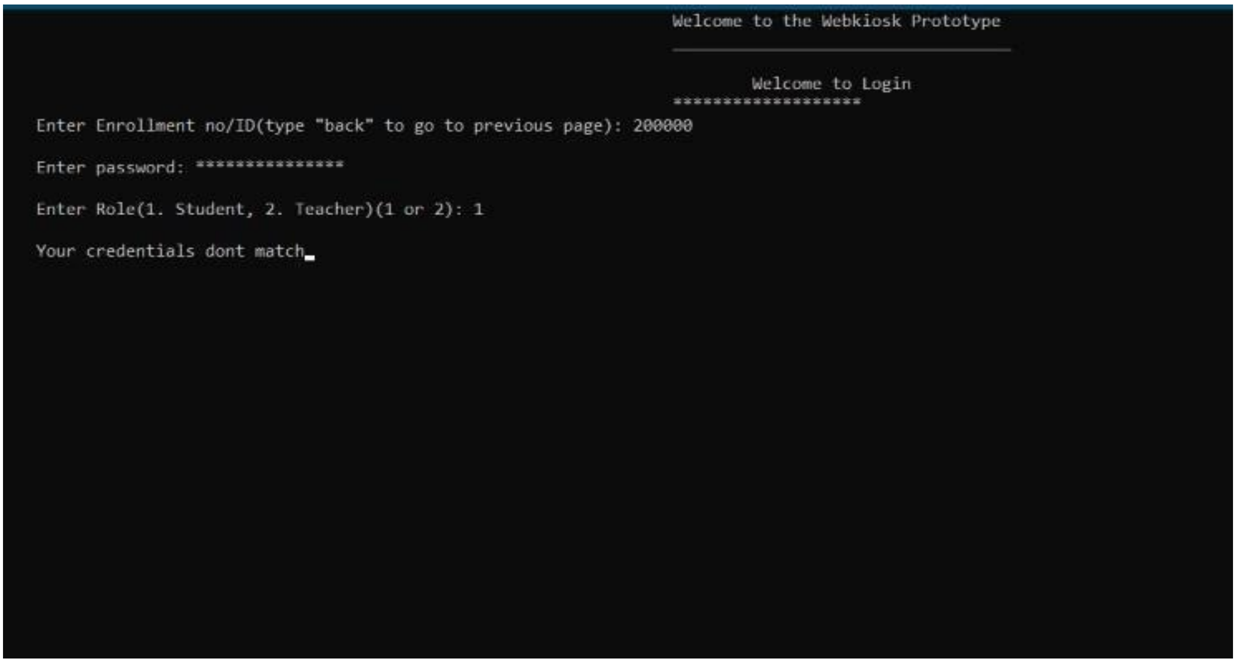
	A	B	C	D	E	F
1	Enroll	Name	Branch	Batch		
2	200000	Tanansh A	CSE	B3		
3	200001	Bhavya ko	CSE	B3		
4						
5						
6						
7						
8						
9						
10						

7.) This page is from students point of view that how his credentials are going to be entered .



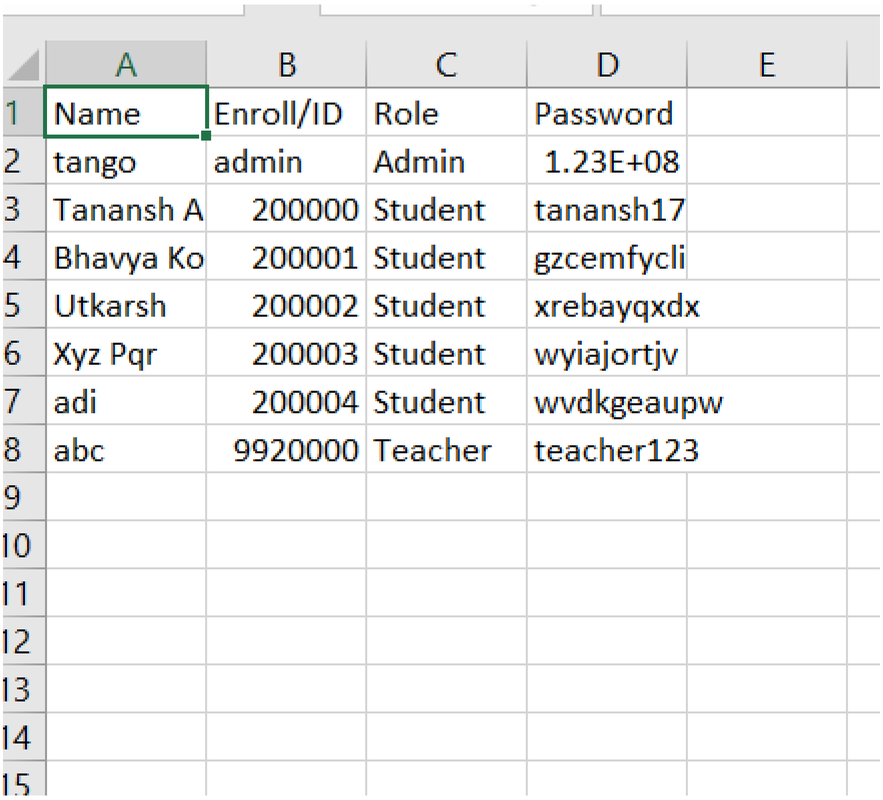
```
                                                                                 Welcome to the Webkiosk Prototype
                                                                                 _________________________________

                                                                                          Welcome to Login
                                                                                 ********************
    Enter Enrollment no/ID(type "back" to go to previous page): 200000

    Enter password: ***************
```

8.) This also from students point of view that what if the student enters wrong credentials .



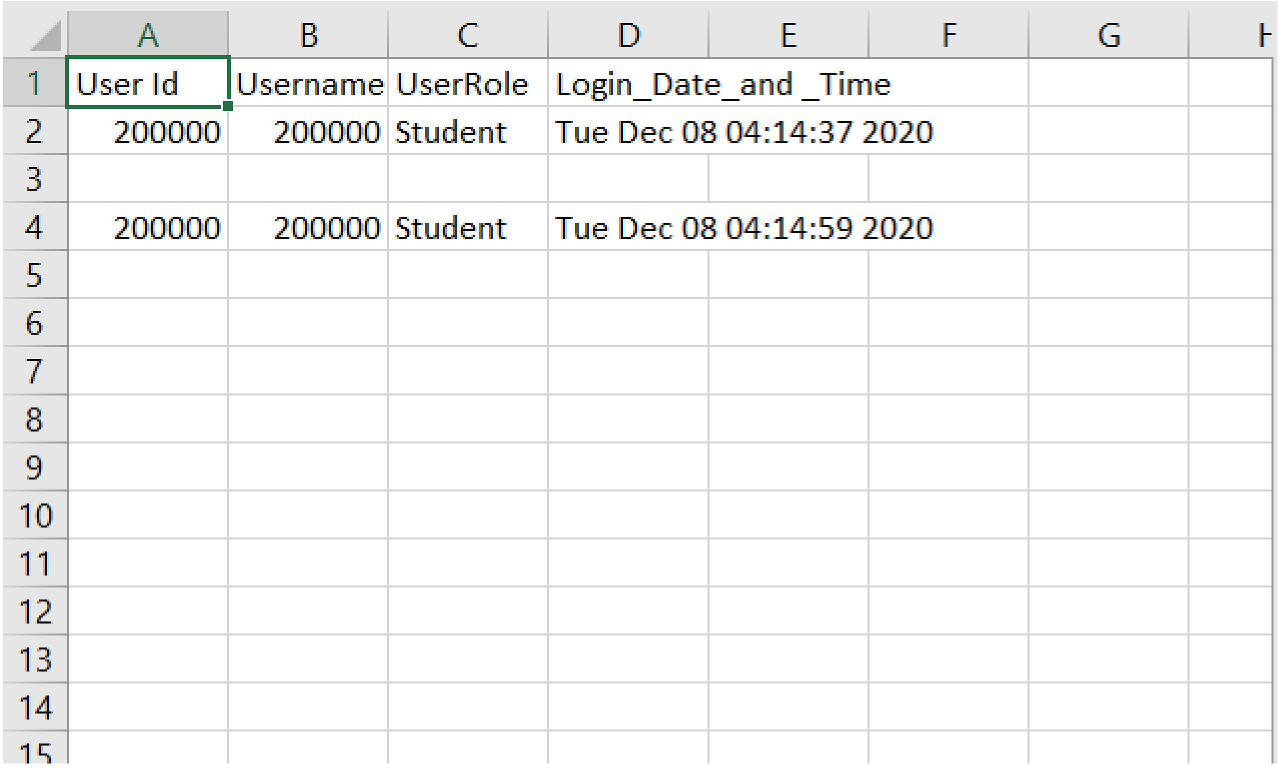
```
                                                                Welcome to the Webkiosk Prototype


                                                                        Welcome to Login
                                                                *******************
   Enter Enrollment no/ID(type "back" to go to previous page): 200000

   Enter password: ***************

   Enter Role(1. Student, 2. Teacher)(1 or 2): 1

   Your credentials dont match_
```

9.) These are all the entries in registry.



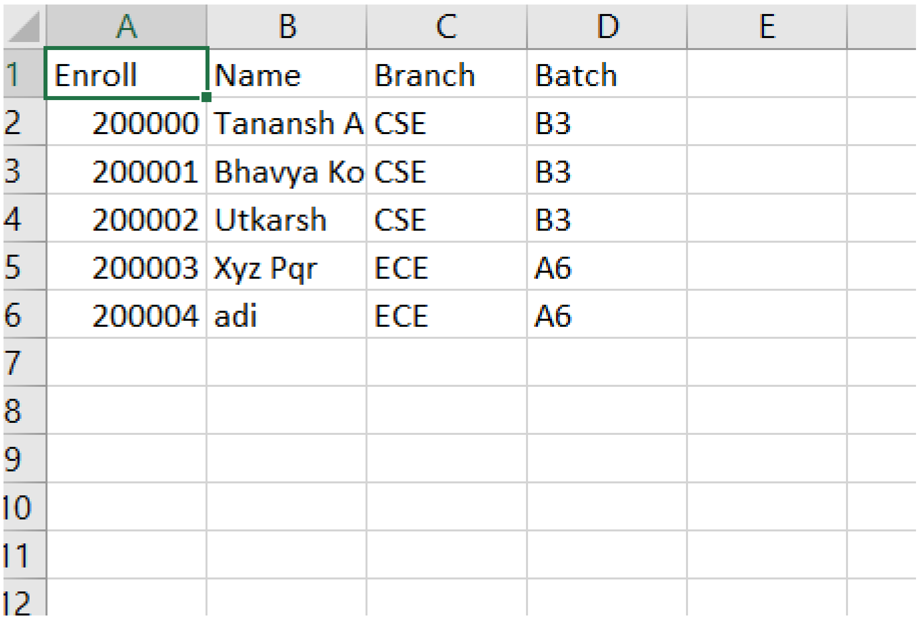
	A	B	C	D	E
1	Name	Enroll/ID	Role	Password	
2	tango	admin	Admin	1.23E+08	
3	Tanansh A	200000	Student	tanansh17	
4	Bhavya Ko	200001	Student	gzcemfycli	
5	Utkarsh	200002	Student	xrebayqxdx	
6	Xyz Pqr	200003	Student	wyiajortjv	
7	adi	200004	Student	wvdkgeaupw	
8	abc	9920000	Teacher	teacher123	
9					
10					
11					
12					
13					
14					
15					

10.) Function updatetime() writing in file of each login



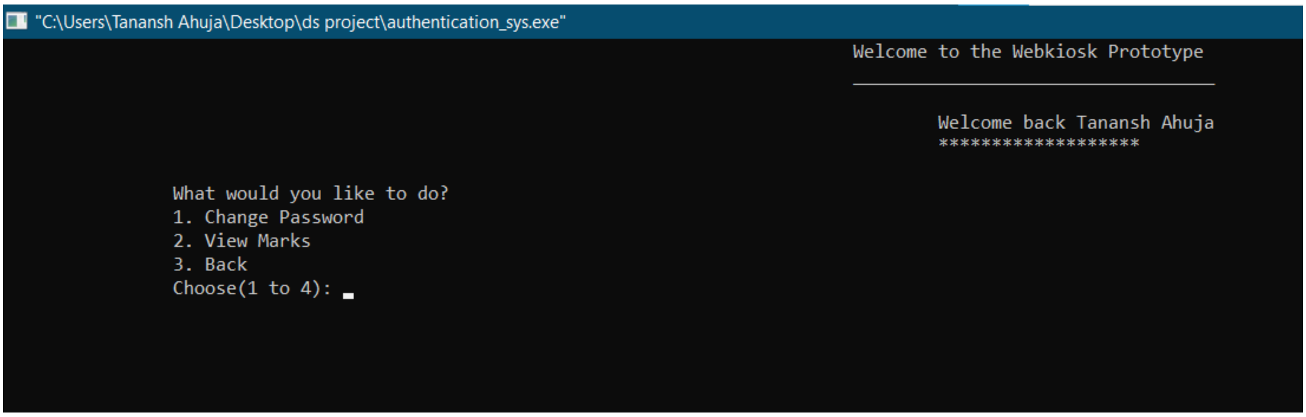
	A	B	C	D	E	F	G	H
1	User Id	Username	UserRole	Login_Date_and _Time				
2	200000	200000	Student	Tue Dec 08 04:14:37 2020				
3								
4	200000	200000	Student	Tue Dec 08 04:14:59 2020				
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

11.) Student table contains extra details of the student that was not in registry.



	A	B	C	D	E
1	Enroll	Name	Branch	Batch	
2	200000	Tanansh A	CSE	B3	
3	200001	Bhavya Ko	CSE	B3	
4	200002	Utkarsh	CSE	B3	
5	200003	Xyz Pqr	ECE	A6	
6	200004	adi	ECE	A6	
7					
8					
9					
10					
11					
12					

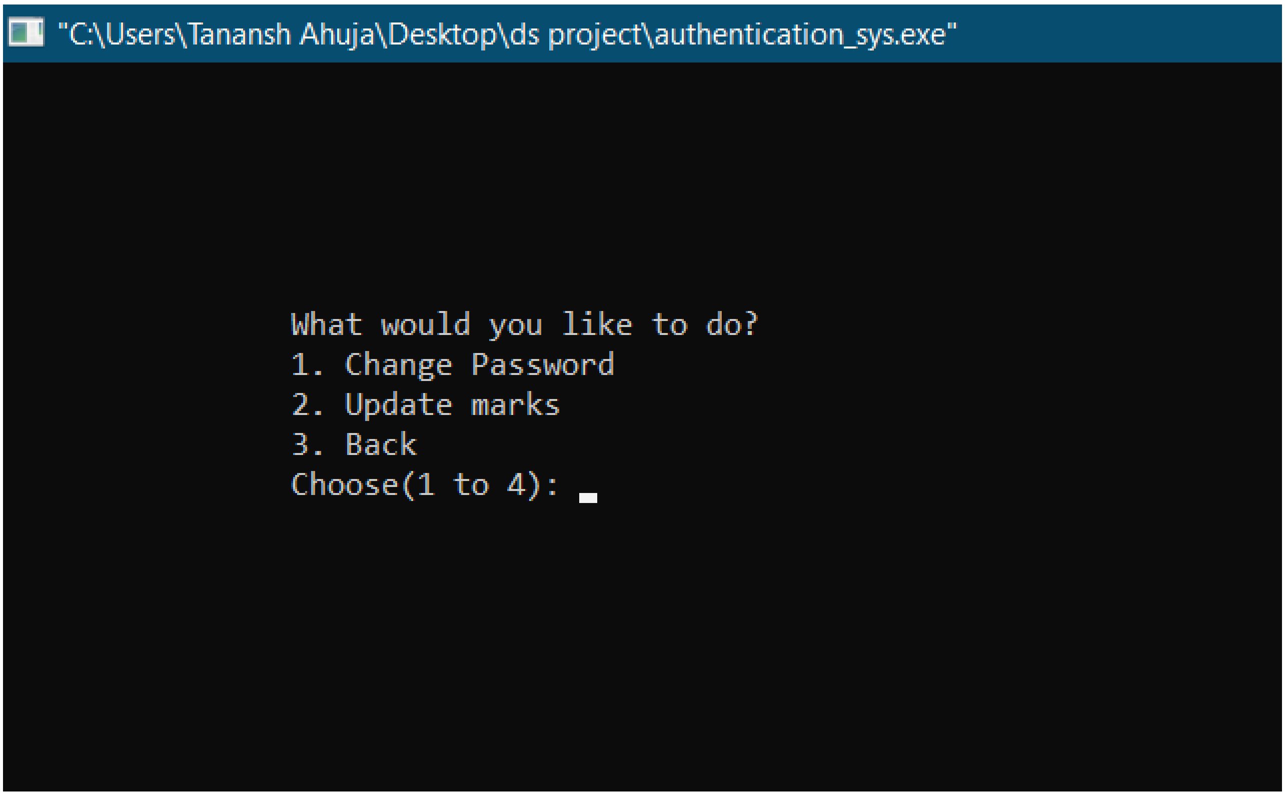
12.) Layout of what a student would see/



"C:\Users\Tanansh Ahuja\Desktop\ds project\authentication_sys.exe"

```
                                                                             Welcome to the Webkiosk Prototype
                                                                             _________________________________

                                                                                     Welcome back Tanansh Ahuja
                                                                                     *******************



               What would you like to do?
               1. Change Password
               2. View Marks
               3. Back
               Choose(1 to 4): _
```

13.) Layout of what a teacher would see



"C:\Users\Tanansh Ahuja\Desktop\ds project\authentication_sys.exe"

```
               What would you like to do?
               1. Change Password
               2. Update marks
               3. Back
               Choose(1 to 4): _
```

Conclusion:

Through the project we tried to create a webkiosk’s prototype through which we managed the data. We even used how to create excel file through file handling. We analysed how the authorisation is done in actual Webkiosk is also. All the operations example updating of any one’s data and deletion insertion generating new password for every teacher and student etc. are also applied in the project.