Global University Bangladesh

Department of Computer Science and Engineering

A project report on

Remote Academic Data Management System

Submitted by:

Ahmad Nafis Hasan
ID No.: 173-011-002
Program: B.Sc. in CSE, Regular

Supervised by:

Syed Imran Hossain
Lecturer, Department of CSE, GUB

Department of Computer Science and Engineering
Global University Bangladesh

August, 2021

Global University Bangladesh

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project entitled "**Remote Academic Data Management System" by "Ahmad Nafis Hasan", ID No.: 173-011-002,** has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering on **August, 2021.**

Signature of Supervisor

Syed Imran Hossain

Lecturer

Department of Computer Science and Engineering

Global University Bangladesh

Global University Bangladesh

Department of Computer Science and Engineering

DECLARATION

I hereby declare that my project entitled "**Remote Academic Data Management System"** is the result of my work. I also ensure that it was not previously submitted or published elsewhere for the award of any degree or diploma.

The work has been accepted for the degree of Bachelor of Science in Computer Science and Engineering at Global University Bangladesh (GUB).

Author

………………………….
(Name)

ACKNOWLEDGMENT

I am honestly grateful to our honorable teacher and supervisor Syed Imran Hossain, Lecturer, Department of Computer Science and Engineering (CSE), Global University Bangladesh (GUB).

With my gratitude I also want to say thank you and wish to express my heart full thanks to all of our honorable teachers of the Department of Computer Science and Engineering (CSE), Global University Bangladesh (GUB).

I am strongly thankful to Syed Imran Hossain Our Honorable Teacher and former Lecturer, Md. Abu Saleh Ovi whom has enlightened me with their vast knowledge in Database Management System, which helped me build the base of this project.

I am also thankful to online community like Stack Overflow, Code Project, for getting solution for most of the problem I faced developing this project using C# as a beginner.

ABSTRACT

Remote Academic Data Management System is a windows-based desktop software, which helps faculties, exam controller, register remotely manage, view, generate report related to course, offered course, result, course registration, student admission. Front-end of this software has been designed using .NET WinForms and Back-end has been developed using C#. In this system Data can be stored in both remotely using MySQL and locally in SQLite DBMS system. For remote database, software will automatically export the database in user given server at initial state of the software, or user can configure database manually.

List of contents

List of tables

List of figures

Abbreviations and List of Symbols

DBMS	Database Management System
PRI	Primary Key
MUL	Multiple Key-Foreign and Primary
DPAPI	Data Protection API
DGV	Data Greed View

Chapter 1

Introduction

1.1 Background

In current system academics or exam controller officers have to input data manually or sometimes they have to insert the same data multiple times. As exam result data are not stored in an online database system, they have to deliver data hand to hand. As there are no automated data management system data recovery or exporting also difficult. All this matter makes the result publish task difficult and time consuming.

In Remote Academic Data Management System, data will be stored in an online server, which will able its user to access data from anywhere they have internet access.
Data will be organized automatically and stored in database.
Course data such as course code, course name, credit hour will be automatically added as user select specific student.
Data can be stored in both local and remote host.
Data manipulation like summation or calculation of GPA will be done programmatically.

1.2 Objectives

a. Prepare remotely accessible server
b. Create Database Model
c. Choose specific platform for the software
d. Design Interactive User Interface
e. Develop automated data manipulation and storing system
f. Technical and user training

1.3 Benefits

a. Less time consuming and easy process
b. Reliable method
c. Remotely accessible
d. Automatic data manipulation
e. Reliable data backup system
f. Easy to access, recover or store data

1.4 Platform

This system can be developed for web, mobile or computer platform.
Though currently the software will be available for only windows (personal computer) platform.

1.5 Tools and Resources

Linux, ubuntu server OS has been used for the remote server management.
MySQL has been used for data management system.
The software has been developed using C# programming language for windows platform.
.NET Framework has been used for development process.
Visual Studio IDE will be used for development, publishing and windows form UI design.

1.6 Development and maintenance Cost

Cost for retail Virtual private server with Dedicated IP will be around 80-100$ in a Year.
This yearly cost can be reduced by building personal server with public IP address.
Using Shared server can farther reduce the cost

Chapter 2
Database Model

2.1 Student Information Table

```
+------------------+--------------+------+-----+-------------------+-----------------------------+
| Field            | Type         | Null | Key | Default           | Extra                       |
+------------------+--------------+------+-----+-------------------+-----------------------------+
| studentId        | int(11)      | NO   | PRI | NULL              |                             |
| batchId          | int(11)      | YES  | MUL | NULL              |                             |
| programId        | int(11)      | YES  | MUL | NULL              |                             |
| name             | varchar(60)  | YES  |     | NULL              |                             |
| contact          | varchar(20)  | YES  |     | NULL              |                             |
| father           | varchar(60)  | YES  |     | NULL              |                             |
| fatherContact    | varchar(20)  | YES  |     | NULL              |                             |
| fatherProfession | varchar(50)  | YES  |     | NULL              |                             |
| mother           | varchar(60)  | YES  |     | NULL              |                             |
| motherContact    | varchar(20)  | YES  |     | NULL              |                             |
| motherProfession | varchar(50)  | YES  |     | NULL              |                             |
| mailingAddress   | varchar(150) | YES  |     | NULL              |                             |
| permanentAddress | varchar(150) | YES  |     | NULL              |                             |
| email            | varchar(50)  | YES  |     | NULL              |                             |
| gender           | varchar(20)  | YES  |     | NULL              |                             |
| reilgion         | varchar(20)  | YES  |     | NULL              |                             |
| bloodGroup       | varchar(10)  | YES  |     | NULL              |                             |
| NID              | varchar(20)  | YES  |     | NULL              |                             |
| birthReg         | varchar(20)  | YES  |     | NULL              |                             |
| nationality      | varchar(20)  | YES  |     | NULL              |                             |
| dob              | date         | YES  |     | NULL              |                             |
| remark           | varchar(150) | YES  |     | NULL              |                             |
| created          | timestamp    | NO   |     | CURRENT_TIMESTAMP |                             |
| modified         | timestamp    | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+------------------+--------------+------+-----+-------------------+-----------------------------+
```

Table 2.1: Student Information Table

This Database entity will store data related to student's personal Information

2.2 Guardian or Emergency Contact Information Table

```
+---------------------+-------------+------+-----+------------------+----------------------------+
| Field               | Type        | Null | Key | Default          | Extra                      |
+---------------------+-------------+------+-----+------------------+----------------------------+
| studentId           | int(11)     | NO   | PRI | NULL             |                            |
| name                | varchar(60) | YES  |     | NULL             |                            |
| contact             | varchar(20) | YES  |     | NULL             |                            |
| mailingAddress      | varchar(200)| YES  |     | NULL             |                            |
| permanentAddress    | varchar(200 | YES  |     | NULL             |                            |
| relationWithStudent | varchar(30) | YES  |     | NULL             |                            |
| remark              | varchar(30) | YES  |     | NULL             |                            |
| created             | timestamp   | NO   |     | CURRENT_TIMESTAMP|                            |
| modified            | timestamp   | NO   |     | CURRENT_TIMESTAMP| on update CURRENT_TIMESTAMP|
+---------------------+-------------+------+-----+------------------+----------------------------+
```

Table 2.2: Guardian or Emergency Contact Information Table

This Database entity will store data related to student's local guardian or emergency contact information.

2.3 Educational Qualification Data Table

```
+-------------+--------------+------+-----+-------------------+-----------------------------+
| Field       | Type         | Null | Key | Default           | Extra                       |
+-------------+--------------+------+-----+-------------------+-----------------------------+
| educationId | int(11)      | NO   | PRI | NULL              |                             |
| exam        | varchar(50)  | NO   | PRI | NULL              |                             |
| year        | int(11)      | NO   |     | NULL              |                             |
| rollID      | varchar(25)  | NO   |     | NULL              |                             |
| department  | varchar(50)  | YES  |     | NULL              |                             |
| baord       | varchar(80)  | NO   |     | NULL              |                             |
| resultType  | varchar(20)  | NO   |     | NULL              |                             |
| resultClass | int(11)      | YES  |     | NULL              |                             |
| resultGPA   | decimal(6,2) | YES  |     | NULL              |                             |
| outOf       | decimal(6,2) | YES  |     | NULL              |                             |
| institute   | varchar(80)  | YES  |     | NULL              |                             |
| created     | timestamp    | NO   |     | CURRENT_TIMESTAMP |                             |
| modified    | timestamp    | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+-------------+--------------+------+-----+-------------------+-----------------------------+
```

Table 2.3: Educational Qualification Data Table

This database entity will store data related to student’s or other member’s previous educational qualification.

2.4 Course Table

```
+-------------+---------------+------+-----+-------------------+-----------------------------+
| Field       | Type          | Null | Key | Default           | Extra                       |
+-------------+---------------+------+-----+-------------------+-----------------------------+
| courseCode  | varchar(20)   | NO   | PRI | NULL              |                             |
| courseTitle | varchar(80)   | NO   |     | NULL              |                             |
| credit      | decimal(2,1)  | NO   |     | NULL              |                             |
| type        | varchar(25)   | NO   |     | NULL              |                             |
| created     | timestamp     | NO   |     | CURRENT_TIMESTAMP |                             |
| modified    | timestamp     | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
| shortDesc   | varchar(1000) | YES  |     | NULL              |                             |
+-------------+---------------+------+-----+-------------------+-----------------------------+
```

Table 2.4: Course Table

This database entity will store data related to course such as course code, course title, credit.

2.5 Course and Program Relational Table

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| programId  | int(11)     | NO   | PRI | NULL    |       |
| courseCode | varchar(20) | NO   | PRI | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

Table 2.5: Course and Program Relational Table

This databases entity was created to make relation with course and program table and reduce redundancy in course table. For example, a course like Digital Logic Design can be offered for both CSE and EEE, regular or evening program. If we want to be sure which courses are available for a particular program, we have to store program data alongside course related data in course table which will increase repeated data in course table. So, to avoid this data redundancy the course-program table has been created.

2.6 Semester Table

```
+----------------+-------------+------+-----+-------------------+-----------------------------+
| Field          | Type        | Null | Key | Default           | Extra                       |
+----------------+-------------+------+-----+-------------------+-----------------------------+
| semesterId     | int(11)     | NO   | PRI | NULL              |                             |
| name           | varchar(30) | NO   |     | NULL              |                             |
| SemesterLength | int(11)     | YES  |     | NULL              |                             |
| created        | timestamp   | NO   |     | CURRENT_TIMESTAMP |                             |
| modified       | timestamp   | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+----------------+-------------+------+-----+-------------------+-----------------------------+
```

Table 2.6: Semester Table

This database entity will store data related to a particular semester like Spring-2020

2.7 Offered Course Table

```
+-----------------+-------------+------+-----+-------------------+-----------------------------+
| Field           | Type        | Null | Key | Default           | Extra                       |
+-----------------+-------------+------+-----+-------------------+-----------------------------+
| courseCode      | varchar(30) | NO   | PRI | NULL              |                             |
| batchId         | int(11)     | NO   | PRI | NULL              |                             |
| semesterId      | int(11)     | NO   | PRI | NULL              |                             |
| programId       | int(11)     | NO   | PRI | NULL              |                             |
| facultyMemberId | int(11)     | YES  | MUL | NULL              |                             |
| created         | timestamp   | NO   |     | CURRENT_TIMESTAMP |                             |
| modified        | timestamp   | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+-----------------+-------------+------+-----+-------------------+-----------------------------+
```

Table 2.7: Offered Course Table

This database entity will store data related to offered courses in a particular semester.

2.8 Course Registration and Marks Table

```
+-----------------+--------------+------+-----+-------------------+-----------------------------+
| Field           | Type         | Null | Key | Default           | Extra                       |
+-----------------+--------------+------+-----+-------------------+-----------------------------+
| registrationId  | varchar(15)  | NO   |     | NULL              |                             |
| studentId       | int(11)      | NO   | PRI | NULL              |                             |
| courseCode      | varchar(20)  | NO   | PRI | NULL              |                             |
| semesterId      | int(11)      | NO   | PRI | NULL              |                             |
| remarkFaculty   | varchar(50)  | YES  |     | NULL              |                             |
| facultyMemberId | int(11)      | YES  |     | NULL              |                             |
| reamrkRegister  | varchar(50)  | YES  |     | NULL              |                             |
| staffId         | int(11)      | YES  |     | NULL              |                             |
| regDate         | datetime     | NO   |     | NULL              |                             |
| regType         | varchar(20)  | YES  |     | NULL              |                             |
| attendance      | decimal(4,2) | YES  |     | NULL              |                             |
| assignment      | decimal(4,2) | YES  |     | NULL              |                             |
| classMark       | decimal(4,2) | YES  |     | NULL              |                             |
| midViva         | decimal(4,2) | YES  |     | NULL              |                             |
| final           | decimal(4,2) | YES  |     | NULL              |                             |
| created         | timestamp    | NO   |     | CURRENT_TIMESTAMP |                             |
| modified        | timestamp    | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+-----------------+--------------+------+-----+-------------------+-----------------------------+
```

Table 2.8: Course Registration and Marks Table

This database entity will store course registration related data such as course code, semester, faculty member ID, Student ID, registration date at the time student register for a course. Later this database entity will be updated with course marks such as attendance, assignment, class test, midterm, final, viva, when course teacher input marks for a particular course.

2.9 Faculty Table

```
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| facultyId | int(11)     | NO   | PRI | NULL    |       |
| name      | varchar(50) | NO   |     | NULL    |       |
| dean      | int(11)     | YES  | MUL | NULL    |       |
| contact   | varchar(20) | NO   |     | NULL    |       |
| created   | datetime    | NO   |     | NULL    |       |
| modified  | datetime    | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

Table 2.9: Faculty Table

This database entity will store data relate to faculty.

2.10 Department Table

```
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| departmentId | int(11)      | NO   | PRI | NULL    |       |
| name         | varchar(50)  | NO   |     | NULL    |       |
| head         | int(11)      | YES  | MUL | NULL    |       |
| facultyId    | int(11)      | NO   | MUL | NULL    |       |
| location     | varchar(150) | NO   |     | NULL    |       |
| contact      | varchar(20)  | NO   |     | NULL    |       |
| created      | datetime     | NO   |     | NULL    |       |
| modified     | datetime     | NO   |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
```

Table 2.10: Department Table

This database entity will store data relate to department.

2.11 Batch Table

```
+----------------+-------------+------+-----+-------------------+-----------------------------+
| Field          | Type        | Null | Key | Default           | Extra                       |
+----------------+-------------+------+-----+-------------------+-----------------------------+
| batchId        | int(11)     | NO   | PRI | NULL              |                             |
| name           | varchar(20) | NO   |     | NULL              |                             |
| circularDate   | date        | YES  |     | NULL              |                             |
| deadline       | date        | YES  |     | NULL              |                             |
| classStartDate | date        | YES  |     | NULL              |                             |
| month          | date        | YES  |     | NULL              |                             |
| created        | timestamp   | NO   |     | CURRENT_TIMESTAMP |                             |
| modified       | timestamp   | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+----------------+-------------+------+-----+-------------------+-----------------------------+
```

Table 2.11: Batch Table

This database entity will store data relate to batch such batch id, circular date, class start date Batch ID example: 173, which represents third of year 2017.

2.12 Program Table

```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| programId    | int(11)     | NO   | PRI | NULL    |       |
| departmentId | int(11)     | NO   | MUL | NULL    |       |
| name         | varchar(50) | NO   |     | NULL    |       |
| shift        | varchar(20) | NO   |     | NULL    |       |
| created      | datetime    | NO   |     | NULL    |       |
| modified     | datetime    | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

Table 2.12: Program Table

This database entity will store data relate to program such program id, department ID, name of the program, regular or evening.

2.13 Faculty Member Table

```
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| facultyMemberId    | int(11)      | NO   | PRI | NULL    |       |
| name               | varchar(60)  | NO   |     | NULL    |       |
| departmentId       | int(11)      | YES  | MUL | NULL    |       |
| father             | varchar(60)  | NO   |     | NULL    |       |
| mother             | varchar(60)  | NO   |     | NULL    |       |
| contact            | varchar(20)  | NO   |     | NULL    |       |
| alternativeContact | varchar(20)  | YES  |     | NULL    |       |
| mailingAddress     | varchar(150) | NO   |     | NULL    |       |
| permanentAddress   | varchar(150) | NO   |     | NULL    |       |
| email              | varchar(50)  | YES  |     | NULL    |       |
| gender             | varchar(20)  | NO   |     | NULL    |       |
| reilgion           | varchar(20)  | NO   |     | NULL    |       |
| bloodGroup         | varchar(10)  | NO   |     | NULL    |       |
| NID                | varchar(20)  | NO   |     | NULL    |       |
| birthReg           | varchar(20)  | NO   |     | NULL    |       |
| nationality        | varchar(20)  | NO   |     | NULL    |       |
| dob                | date         | NO   |     | NULL    |       |
| created            | datetime     | YES  |     | NULL    |       |
| modified           | datetime     | YES  |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
```

Table 2.13: Faculty Member Table

This database entity will store data relate to faculty member’s personal information. Faculty member table is related with offered course table by using faculty member ID as foreign key in offered course table. This will help shortlist course taken by particular course teacher. It will also help to suggest only the courses a course teacher have in a semester with a batch, when a faculty member access this system for updating marks of a student.

2.14 Stuff Table

```
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| stuffId            | int(11)      | NO   | PRI | NULL    |       |
| position           | varchar(50)  | YES  |     | NULL    |       |
| name               | varchar(60)  | NO   |     | NULL    |       |
| father             | varchar(60)  | NO   |     | NULL    |       |
| mother             | varchar(60)  | NO   |     | NULL    |       |
| contact            | varchar(20)  | NO   |     | NULL    |       |
| alternativeContact | varchar(20)  | YES  |     | NULL    |       |
| mailingAddress     | varchar(150) | NO   |     | NULL    |       |
| permanentAddress   | varchar(150) | NO   |     | NULL    |       |
| email              | varchar(50)  | YES  |     | NULL    |       |
| gender             | varchar(20)  | NO   |     | NULL    |       |
| reilgion           | varchar(20)  | NO   |     | NULL    |       |
| bloodGroup         | varchar(10)  | NO   |     | NULL    |       |
| NID                | varchar(20)  | NO   |     | NULL    |       |
| birthReg           | varchar(20)  | NO   |     | NULL    |       |
| nationality        | varchar(20)  | NO   |     | NULL    |       |
| dob                | date         | NO   |     | NULL    |       |
| created            | datetime     | NO   |     | NULL    |       |
| modified           | datetime     | NO   |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
```

Table 2.14: Stuff Table

This database entity will store data relate to stuffs personal information, who are working in different official sections.

2.15 User Table

```
+----------+-------------+------+-----+-------------------+-----------------------------+
| Field    | Type        | Null | Key | Default           | Extra                       |
+----------+-------------+------+-----+-------------------+-----------------------------+
| userId   | varchar(30) | NO   | PRI | NULL              |                             |
| password | char(35)    | NO   |     | NULL              |                             |
| role     | varchar(20) | NO   |     | NULL              |                             |
| created  | timestamp   | NO   |     | CURRENT_TIMESTAMP |                             |
| modified | timestamp   | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+----------+-------------+------+-----+-------------------+-----------------------------+
```

Table 2.14: User Table

This database entity will store data related to system access control such User ID, Password, Role as user. Password will be stored as hash string of 32 character.

Most of this tables have two common attributes created and modified to store the date in which the rows will be added and later modified. Default value of created and modified attribute is CURRENT_TIMESTAMP, which automatically capture and store current date and time form server OS. Modified has one extra function on update CURRENT_TIMESTAMP, which capture and store current date and time from server OS every time data of a particular row is being modified.

Chapter 3
User Interface

User interface of this system has been designed using WinForms .NET Frameworks 4.7.2 in visual studio IDE. Windows Forms, a UI framework that creates rich desktop client apps for Windows. The Windows Forms development platform supports a broad set of app development features, including controls, graphics, data binding, and user input. Windows Forms features a drag-and-drop visual designer in Visual Studio to easily create Windows Forms apps. Windows size of this system is 1386 by 788 pixel. Auto scale mode has been set to dpi, which determines how the form or control will scale when screen resolution fonts changes.

When a user does something to form or one of its controls, the action generates an event. Software then reacts to these events with code, and processes the events when they occur. Windows Forms contains a variety of controls that can be added to forms: controls that display text boxes, buttons, drop-down boxes, radio buttons, and even webpages.

3.1 Front Panel

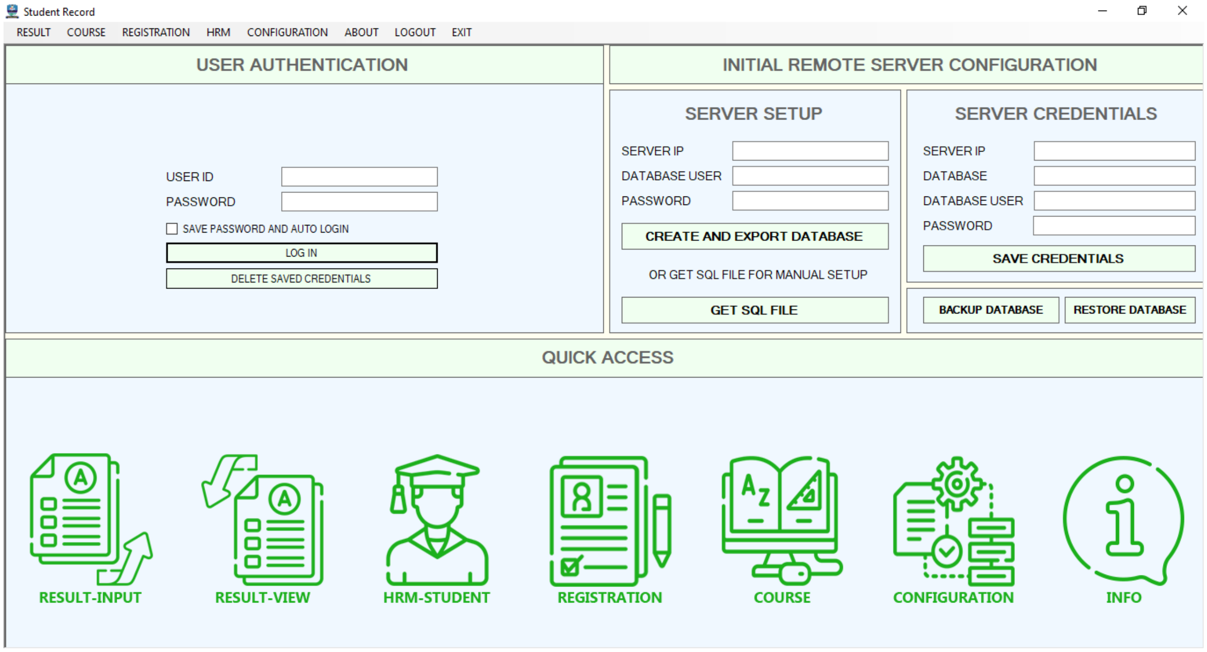


Figure 3.1: Front Panel

Front panel consist of three sections which are user authentication, Initial remote server configuration and quick action.
User authentication section will verify user identity upon launching the software. User can click save password and auto login checkbox, which will save user password in application

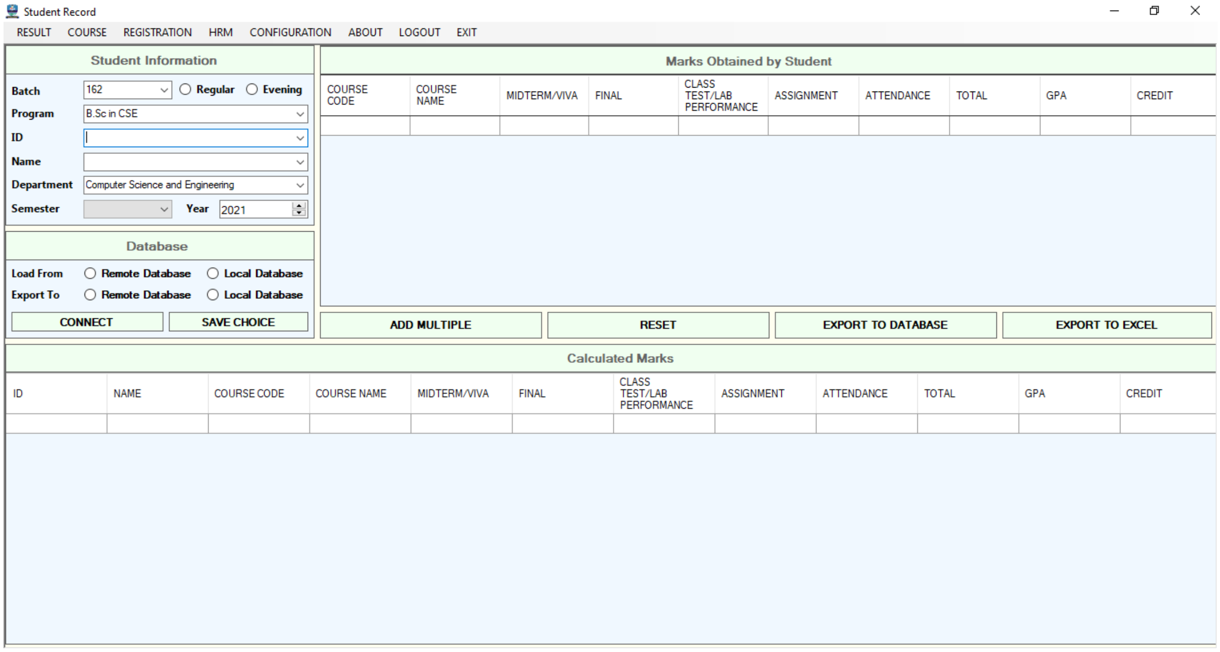
setting and in future launch it will automatically verify user identity with saved credentials. User alco can remove this saved credential by clicking delete credentials button.

Initial remote server configuration section consists of two different sections. Server setup section will enable its user automatically create and export database in user provided server.

In this case user provided database user must have permission to create database and table from remote host. User can also click on GET SQL FILE button to get "RADMS.sql" file and manually setup their server. In server credential section user can input server IP, database name, database user and password and click save credential button, which will allow user use their specified server for data retrieval. Use can also click on BACKUP DATABASE and RESTORE DATABASE button if they need to backup or restore old database. In this cases user specified database user must have remote server access permission.

User can use both toolbar or quick access section to open other panels available in this software. This section will be inaccessible if user open this software in log out state or click on logout toolbar button.

3.2 Result Input Panel



Student Information

Batch: 162 ○ Regular ○ Evening
Program: B.Sc in CSE
ID:
Name:
Department: Computer Science and Engineering
Semester: Year: 2021

Database

Load From ○ Remote Database ○ Local Database
Export To ○ Remote Database ○ Local Database

Marks Obtained by Student

COURSE CODE	COURSE NAME	MIDTERM/VIVA	FINAL	CLASS TEST/LAB PERFORMANCE	ASSIGNMENT	ATTENDANCE	TOTAL	GPA	CREDIT

Calculated Marks

ID	NAME	COURSE CODE	COURSE NAME	MIDTERM/VIVA	FINAL	CLASS TEST/LAB PERFORMANCE	ASSIGNMENT	ATTENDANCE	TOTAL	GPA	CREDIT

Figure 3.2: Result Input Panel

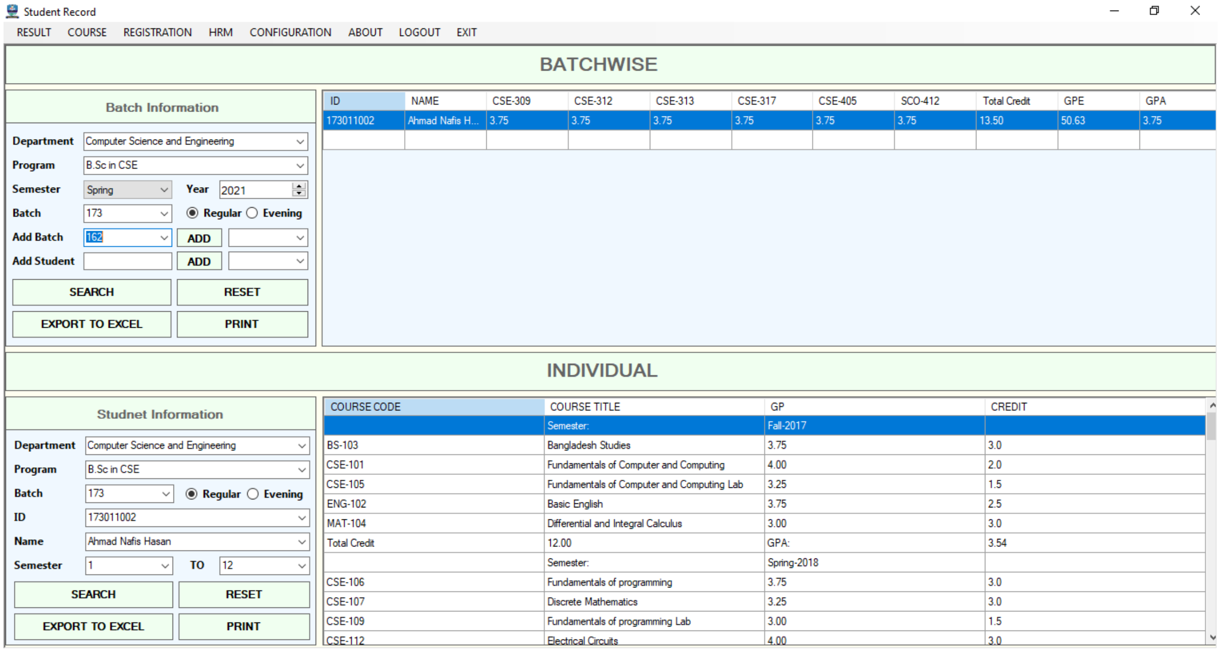
Result input panel consists of four section which are student information, marks obtained by student, calculated marks and database.

In student information common data like batch, program and department will be loaded automatically. If user select shift, batch and program, software will suggest all students fulfill all that selected requirements and populated ID and Name combo box. Later if user select a particular student from suggested students and select semester and year, this software will suggest and populate marks obtained by student data table with course code, course name and credit fetched from registration table. As a result, only the courses that particular student has registered in that specified semester will show. User can only type numeric values in this data table.

Clicking on ADD MULTIPLE button will enable its user to add multiple student's data in calculated marks data table. Reset button will clear out all filled input fields. Clinking on Export to database button will export data from calculated marks data table into database specified in application setting file. Clicking on export to excel will open a excel workbook contained the data from calculated marks data table.

Database section will enable it user to choose from which database data will be loaded and exported. Here user can choose local or remote database.

3.3 Result View Panel



Department	Computer Science and Engineering
Program	B.Sc in CSE
Semester	Spring
Year	2021
Batch	173 ● Regular ○ Evening
Add Batch	162
Add Student	

ID	NAME	CSE-309	CSE-312	CSE-313	CSE-317	CSE-405	SCO-412	Total Credit	GPE	GPA
173011002	Ahmad Nafis H...	3.75	3.75	3.75	3.75	3.75	3.75	13.50	50.63	3.75

Department	Computer Science and Engineering
Program	B.Sc in CSE
Batch	173 ● Regular ○ Evening
ID	173011002
Name	Ahmad Nafis Hasan
Semester	1 TO 12

COURSE CODE	COURSE TITLE	GP	CREDIT
	Semester:	Fall-2017	
BS-103	Bangladesh Studies	3.75	3.0
CSE-101	Fundamentals of Computer and Computing	4.00	2.0
CSE-105	Fundamentals of Computer and Computing Lab	3.25	1.5
ENG-102	Basic English	3.75	2.5
MAT-104	Differential and Integral Calculus	3.00	3.0
Total Credit	12.00	GPA:	3.54
	Semester:	Spring-2018	
CSE-106	Fundamentals of programming	3.75	3.0
CSE-107	Discrete Mathematics	3.25	3.0
CSE-109	Fundamentals of programming Lab	3.00	1.5
CSE-112	Electrical Circuits	4.00	3.0

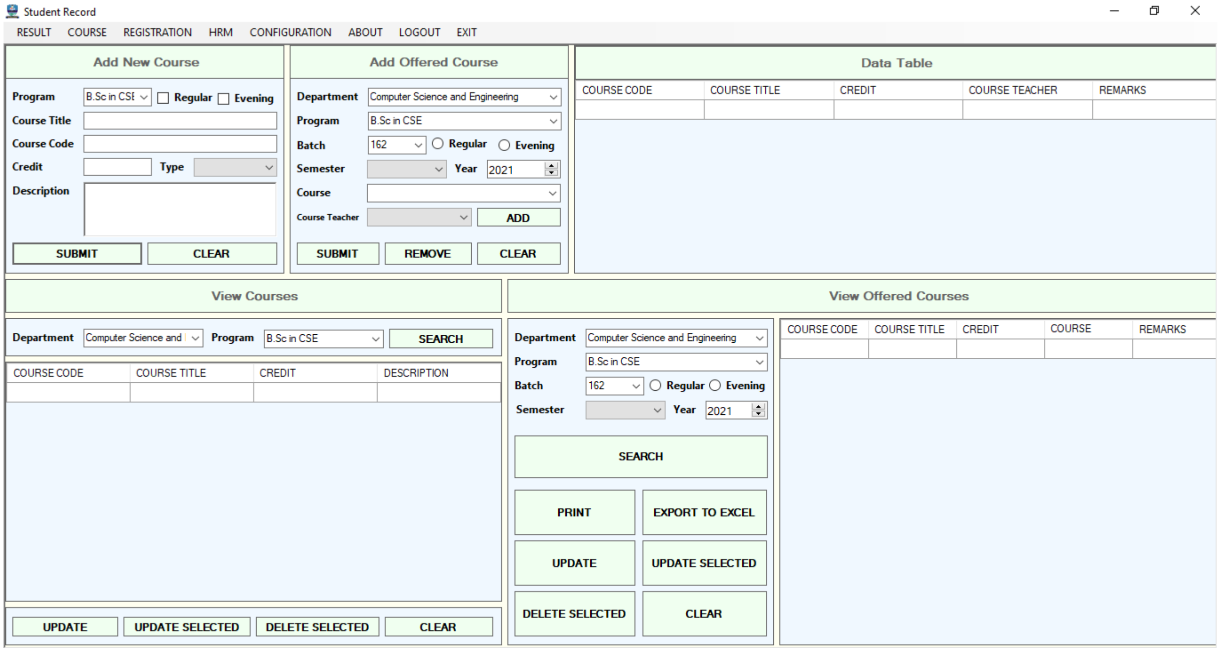
Figure 3.3: Result View Panel

Result view panel consist of two section which are Batchwise and Individual.

Batchwise section will show result of all student of a selected batch of specified semester. User can add additional batch or individual student, if necessary, from add batch combo box and add student textbox. Batchwise data table is a dynamic data table which add columns as course code automatically as necessary. In this case software will search for all courses including retake courses for selected batch and additional batch and student then add distinct columns for all of the courses found in search. User will able to generate report using this data as excel sheet or print generated report directly from software.

In individual section data table will show course code, course title, credit and grade point of a student grouped by semester. It will also show credit earned in a particular semester as well as total earned credit in user specified semesters. Also, in this section user will able to generate report using this data as excel sheet or print generated report directly from software. Student ID, name, department name, program details will be added in report automatically.

3.4 Course Panel



Add New Course

Program: B.Sc in CSE ☐ Regular ☐ Evening
Course Title:
Course Code:
Credit: Type:
Description:

Add Offered Course

Department: Computer Science and Engineering
Program: B.Sc in CSE
Batch: 162 ○ Regular ○ Evening
Semester: Year: 2021
Course:
Course Teacher:

Data Table

COURSE CODE	COURSE TITLE	CREDIT	COURSE TEACHER	REMARKS

View Courses

Department: Computer Science and Program: B.Sc in CSE

COURSE CODE	COURSE TITLE	CREDIT	DESCRIPTION

View Offered Courses

Department: Computer Science and Engineering
Program: B.Sc in CSE
Batch: 162 ○ Regular ○ Evening
Semester: Year: 2021

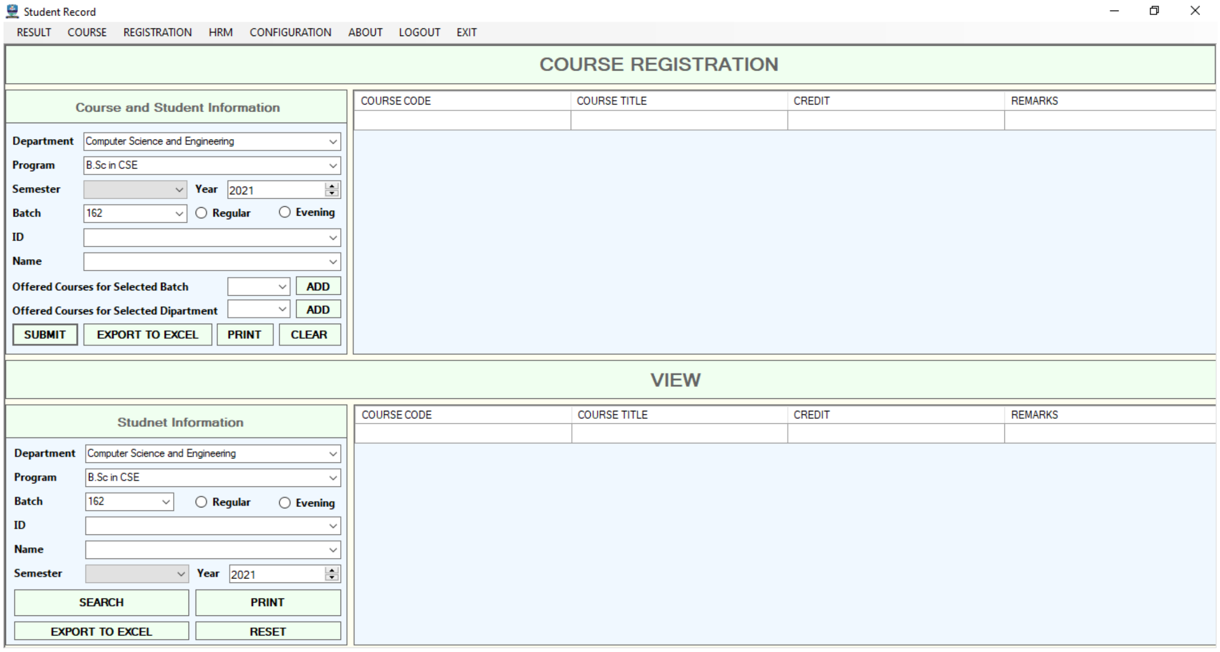
COURSE CODE	COURSE TITLE	CREDIT	COURSE	REMARKS

Figure 3.4: Course Panel

Course panel consist of four section which are Add new course, add offered course, View Courses and View offered course.

In view course and offered course view section user will able to update or delete data as necessary along with viewing.

3.5 Registration Panel



COURSE REGISTRATION

Course and Student Information

Department: Computer Science and Engineering
Program: B.Sc in CSE
Semester: Year: 2021
Batch: 162 ○ Regular ○ Evening
ID:
Name:
Offered Courses for Selected Batch
Offered Courses for Selected Dipartment

COURSE CODE	COURSE TITLE	CREDIT	REMARKS

VIEW

Studnet Information

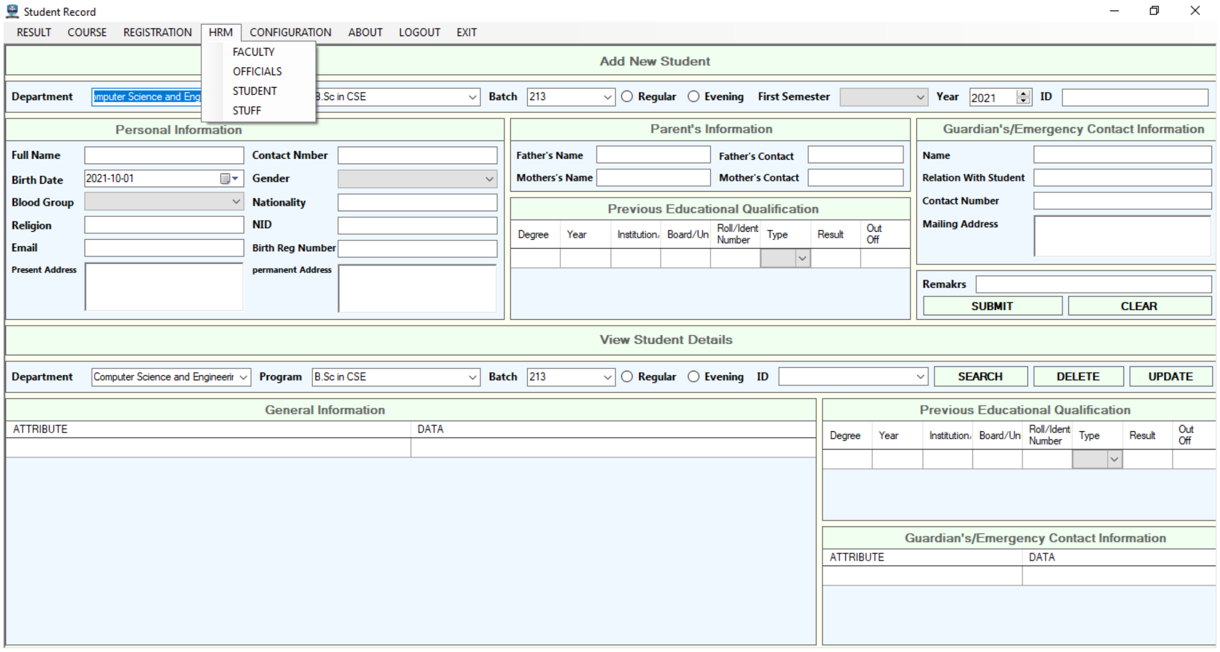
Department: Computer Science and Engineering
Program: B.Sc in CSE
Batch: 162 ○ Regular ○ Evening
ID:
Name:
Semester: Year: 2021

COURSE CODE	COURSE TITLE	CREDIT	REMARKS

Figure 3.5: Registration Panel

Registration panel consist of two section which are course registration and view.
Course registration section will suggest courses from offered courses based on batch for specified semester. User also can choose from all offered courses for specified department.
In view section user can search for previous registration data.
User will able to generate excel and direct printed report from both of these sections.

3.6 Student Panel



Department: mputer Science and Eng | B.Sc in CSE | Batch: 213 | Year: 2021

Birth Date: 2021-10-01

Department: Computer Science and Engineerin | Program: B.Sc in CSE | Batch: 213

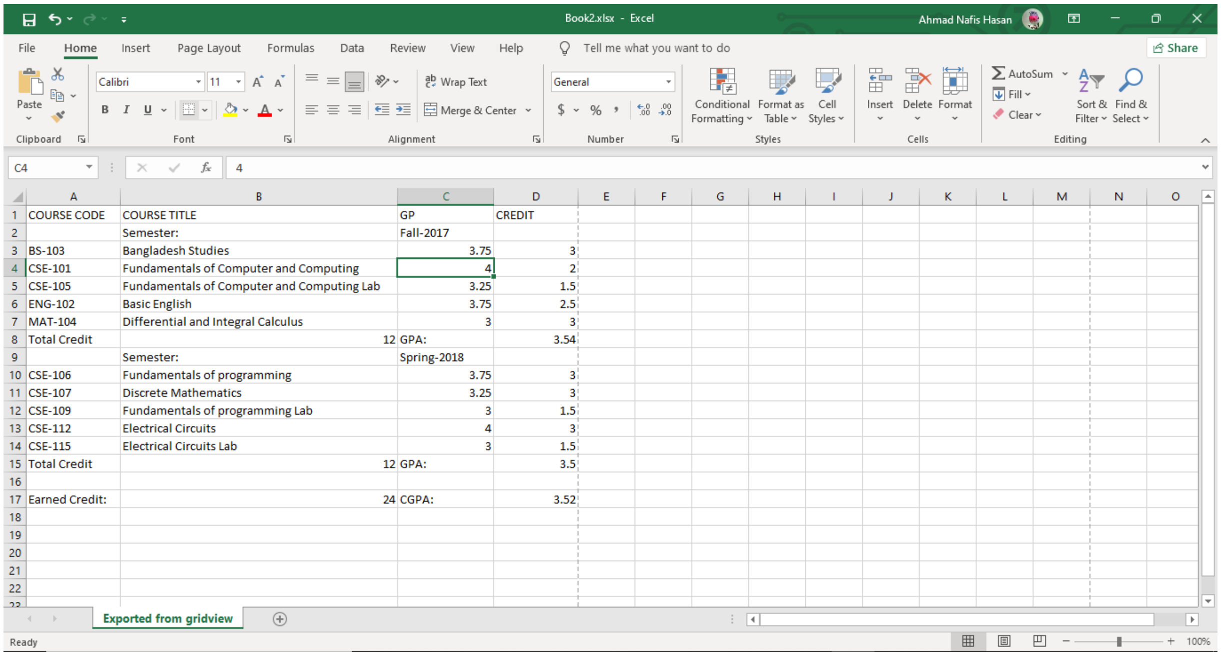
Figure 3.6: Student Panel

Student panel consist of two section which are add new student and view student details.
Add new student section divided into four section which are personal information panel, parents' information panel, emergency contact information panel and previous educational qualification details panel.
If user select program, batch, shift, first semester and year field, this software will automatically suggest Student ID.

In view student details section user can view, delete and update already exist data.

Chapter 4
Report Generation

4.1 Excel Sheet Report



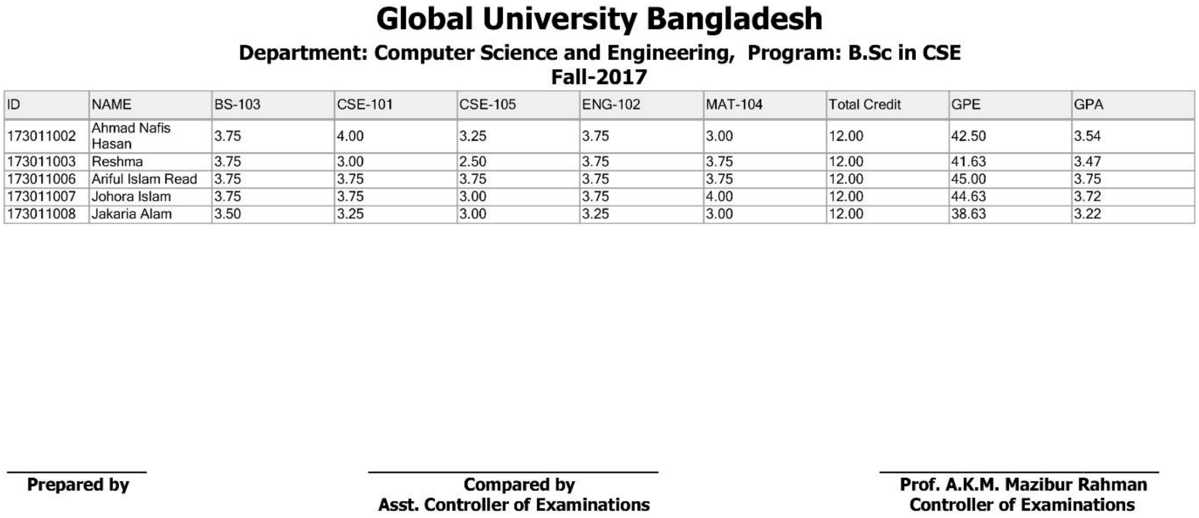
C4 | 4

	A	B	C	D
1	COURSE CODE	COURSE TITLE	GP	CREDIT
2		Semester:	Fall-2017	
3	BS-103	Bangladesh Studies	3.75	3
4	CSE-101	Fundamentals of Computer and Computing	4	2
5	CSE-105	Fundamentals of Computer and Computing Lab	3.25	1.5
6	ENG-102	Basic English	3.75	2.5
7	MAT-104	Differential and Integral Calculus	3	3
8	Total Credit	12	GPA:	3.54
9		Semester:	Spring-2018	
10	CSE-106	Fundamentals of programming	3.75	3
11	CSE-107	Discrete Mathematics	3.25	3
12	CSE-109	Fundamentals of programming Lab	3	1.5
13	CSE-112	Electrical Circuits	4	3
14	CSE-115	Electrical Circuits Lab	3	1.5
15	Total Credit	12	GPA:	3.5
16				
17	Earned Credit:	24	CGPA:	3.52

Figure 4.1: Excel Sheet Report

User will able to generate excel sheet report for result, registration and course section. Clicking on Export to Excel will open a excel book window with exported data within.

4.2 Direct Printed Report



Global University Bangladesh

Department: Computer Science and Engineering, Program: B.Sc in CSE

Fall-2017

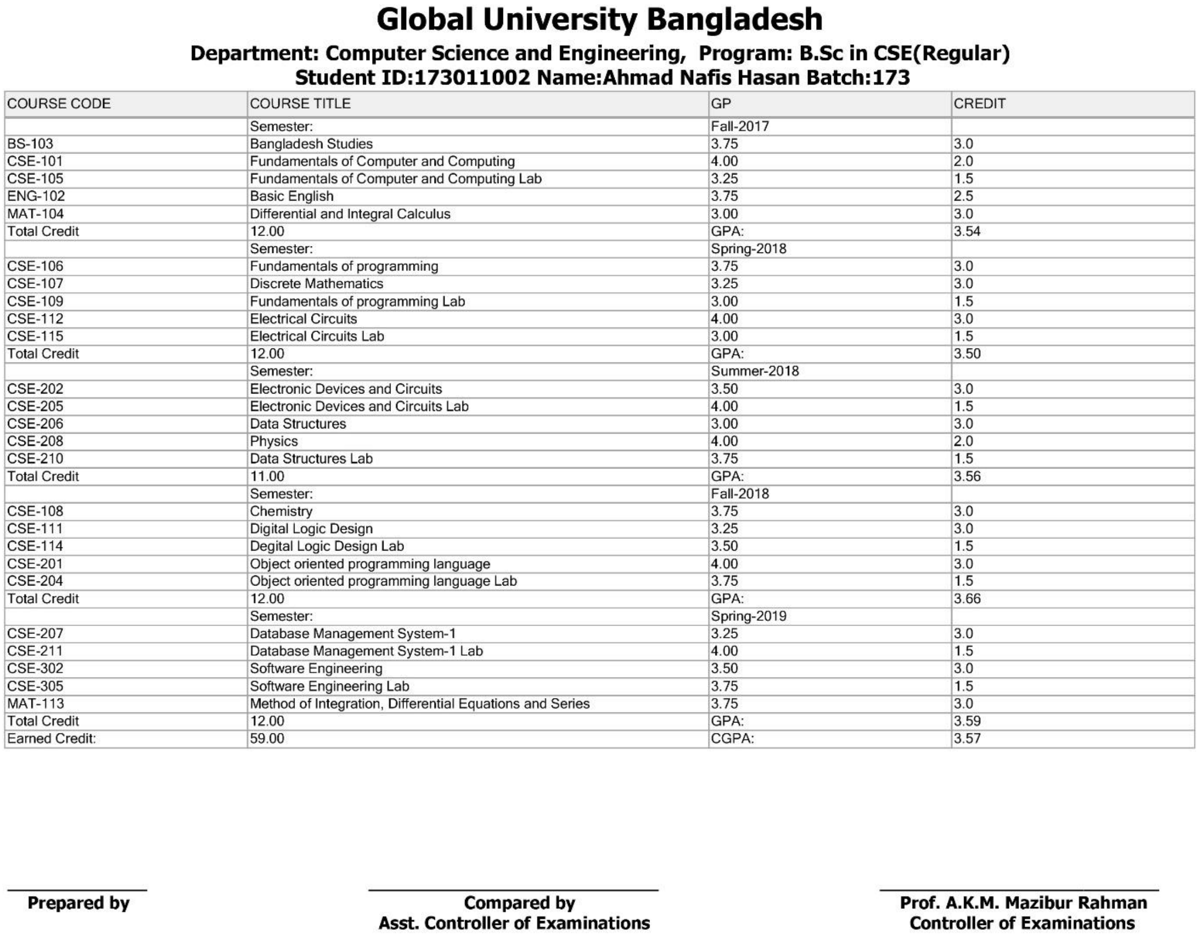
ID	NAME	BS-103	CSE-101	CSE-105	ENG-102	MAT-104	Total Credit	GPE	GPA
173011002	Ahmad Nafis Hasan	3.75	4.00	3.25	3.75	3.00	12.00	42.50	3.54
173011003	Reshma	3.75	3.00	2.50	3.75	3.75	12.00	41.63	3.47
173011006	Ariful Islam Read	3.75	3.75	3.75	3.75	3.75	12.00	45.00	3.75
173011007	Johora Islam	3.75	3.75	3.00	3.75	4.00	12.00	44.63	3.72
173011008	Jakaria Alam	3.50	3.25	3.00	3.25	3.00	12.00	38.63	3.22

Prepared by

Compared by
Asst. Controller of Examinations

Prof. A.K.M. Mazibur Rahman
Controller of Examinations

Figure 4.2.1: Batchwise Direct Printed Report



Global University Bangladesh

Department: Computer Science and Engineering, Program: B.Sc in CSE(Regular)

Student ID:173011002 Name:Ahmad Nafis Hasan Batch:173

COURSE CODE	COURSE TITLE	GP	CREDIT
	Semester:	Fall-2017	
BS-103	Bangladesh Studies	3.75	3.0
CSE-101	Fundamentals of Computer and Computing	4.00	2.0
CSE-105	Fundamentals of Computer and Computing Lab	3.25	1.5
ENG-102	Basic English	3.75	2.5
MAT-104	Differential and Integral Calculus	3.00	3.0
Total Credit	12.00	GPA:	3.54
	Semester:	Spring-2018	
CSE-106	Fundamentals of programming	3.75	3.0
CSE-107	Discrete Mathematics	3.25	3.0
CSE-109	Fundamentals of programming Lab	3.00	1.5
CSE-112	Electrical Circuits	4.00	3.0
CSE-115	Electrical Circuits Lab	3.00	1.5
Total Credit	12.00	GPA:	3.50
	Semester:	Summer-2018	
CSE-202	Electronic Devices and Circuits	3.50	3.0
CSE-205	Electronic Devices and Circuits Lab	4.00	1.5
CSE-206	Data Structures	3.00	3.0
CSE-208	Physics	4.00	2.0
CSE-210	Data Structures Lab	3.75	1.5
Total Credit	11.00	GPA:	3.56
	Semester:	Fall-2018	
CSE-108	Chemistry	3.75	3.0
CSE-111	Digital Logic Design	3.25	3.0
CSE-114	Degital Logic Design Lab	3.50	1.5
CSE-201	Object oriented programming language	4.00	3.0
CSE-204	Object oriented programming language Lab	3.75	1.5
Total Credit	12.00	GPA:	3.66
	Semester:	Spring-2019	
CSE-207	Database Management System-1	3.25	3.0
CSE-211	Database Management System-1 Lab	4.00	1.5
CSE-302	Software Engineering	3.50	3.0
CSE-305	Software Engineering Lab	3.75	1.5
MAT-113	Method of Integration, Differential Equations and Series	3.75	3.0
Total Credit	12.00	GPA:	3.59
Earned Credit:	59.00	CGPA:	3.57

Prepared by

Compared by
Asst. Controller of Examinations

Prof. A.K.M. Mazibur Rahman
Controller of Examinations

Figure 4.2.2: Individual Direct Printed Report

User can directly print report for result, registration and course section within the software. Clicking on Print button will open windows print window where user can choose printer device or print to PDF if available. This auto generated report will add student, batch, department, program, institution name automatically where necessary.

Chapter 5
Backend Development

5.1 Class

Backend of this software has been developed using C#. Event handling block are automatically generated by .NET Framework.

This software consists of twelve class files which are DBconnect.cs, DGVPrinter.cs, EncDec.cs, Form1.cs to Form8.cs and Program.cs

Program.cs Class run at beginning when user run this software. Then it starts executing tread and open Form1.cs.

Fomr1 is a Mdi Container form which allows other forms load within it. It also contains the main menu strip.

DBconnect.cs handles all database connectivity related functionalists such as opening connection, closing connection, checking internet connection availability, select, delete, update data in database, create database in server.

DGVPrinter.cs handles all print related requests. It takes data from DataGrid view as input.
This class can handle document's layout, title, subtitle, font, footer, font color, font size, column size, page number, adding image in background, showing print preview etc.

EncDec.cs contains functions for encrypting, decrypting data. It also has functionality to convert normal string to secure string and return back to normal string.

Form2.cs to Form8.cs handles all event generated by user's interaction with UI and data manipulation and representation works.

5.2 Library References

This system contains several important library references such as MySql.Data, MySqlBackup, Microsoft.Office.Interop.Excel, System.Data.SQLite, System.Security, System.Drawing.

MySql.Data library handles all incoming and outgoing database connection query.

MySqlBackup library handles backup and restoring database properties.

Microsoft.Office.Interop.Excel handles direct data exporting to installed Microsoft excel software in user machine.

System.Drawing helps generating graphical view for printing report generated from DataGrid view.

Chapter 6
Data Security

6.1 Authenticated User Access

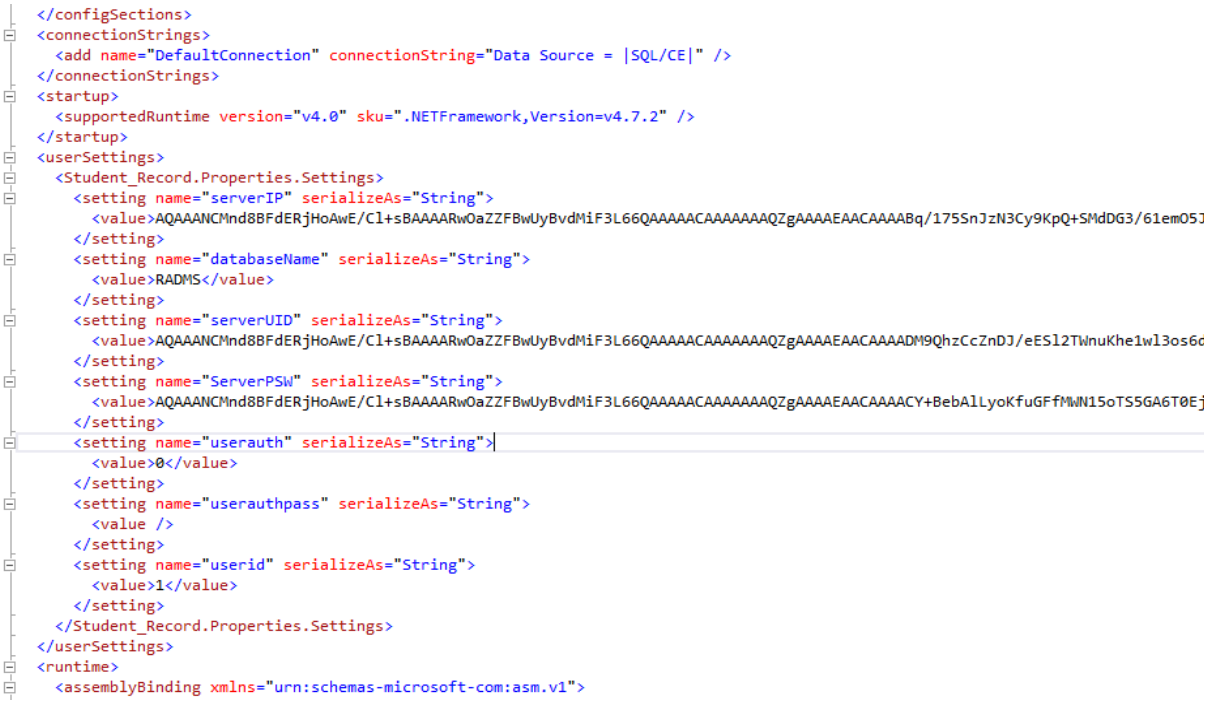
User will only able to access software functionalities after successful log in to the system. This will prevent unauthorized data entry or modification. Only admin user account will have permission to create new user with password, user will be able to change that password from configuration panel. Admin user account will also have permission to delete normal user account.

6.2 User Privilege

Admin can provide user with database access which have limited privilege or only have privilege in specific table. Admin also can control access according to user's role. Which will give admin control over data modification by authorized users.

6.3 Storing User or Database Credential

User password will be stored as MD5 hash string both in server and client machine. That will prevent someone with admin privilege to reveal user credential.
Database credential will be encrypted using DPAPI before storing in application configuration file. DPAPI is a service that is provided by the operating system and does not require additional libraries. It provides protection using the user or machine credentials to encrypt or decrypt data. The encrypted data can only be decrypted on the same machine on which is encrypted.



```
</configSections>
<connectionStrings>
  <add name="DefaultConnection" connectionString="Data Source = |SQL/CE|" />
</connectionStrings>
<startup>
  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
</startup>
<userSettings>
  <Student_Record.Properties.Settings>
    <setting name="serverIP" serializeAs="String">
      <value>AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAARwOaZZFBwUyBvdMiF3L66QAAAAACAAAAAAAQZgAAAAEAACAAAABq/175SnJzN3Cy9KpQ+SMdDG3/61emO5J
    </setting>
    <setting name="databaseName" serializeAs="String">
      <value>RADMS</value>
    </setting>
    <setting name="serverUID" serializeAs="String">
      <value>AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAARwOaZZFBwUyBvdMiF3L66QAAAAACAAAAAAAQZgAAAAEAACAAAADM9QhzCcZnDJ/eES12TWnuKhe1wl3os6d
    </setting>
    <setting name="ServerPSW" serializeAs="String">
      <value>AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAARwOaZZFBwUyBvdMiF3L66QAAAAACAAAAAAAQZgAAAAEAACAAAACY+BebAlLyoKfuGFfMWN15oTS5GA6T0Ej
    </setting>
    <setting name="userauth" serializeAs="String">
      <value>0</value>
    </setting>
    <setting name="userauthpass" serializeAs="String">
      <value />
    </setting>
    <setting name="userid" serializeAs="String">
      <value>1</value>
    </setting>
  </Student_Record.Properties.Settings>
</userSettings>
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
```

Figure 6.1 Storing User or Database Credential

Chapter 7
Conclusion

7.1 Discussion

COVID-19 pandemic situation has proved the necessity and importance of working from home or working remotely. This pandemic situation not only affected companies and industries but also academic sector, forced many to work from their homes, taken exam in online. Even without pandemic situation, working remotely allows for increased flexibility and autonomy for employees. In fact, many of today's top companies offer full, half, or partial remote work. Working remotely doesn't always mean "from home," either; it can apply anytime an employee works off-site—whether that's at a coffee shop or a flexible workspace.

Remote academic data management system will allow employees working in academic areas, managing academic related data remotely. This project has some lacks in many features which needed to be developed for farther advanced remote work but it is still workable in many ways for primary academic data management.

7.2 Feature Improving and New Feature Addition

a. Improve input filed data validation.
b. Full automated data synchronization with remote server and client device.
c. Adding feature for managing data related to account section, waiver, scholarship.

This project was a great learning carve to convert my theorical knowledge into practical work. Also helped find out my weakness and improve from there.

References

a. Windows Forms documentation.
https://docs.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-5.0

b. Encrypting Passwords in a .NET app.config File.
https://weblogs.asp.net/jongalloway/encrypting-passwords-in-a-net-app-config-file

c. DataGridView Printer.
https://www.codeproject.com/Articles/18042/Another-DataGridView-Printer

d. MySQL Documentation.
https://dev.mysql.com/doc/

e. SQLite Documentation.
https://www.sqlite.org/docs.html

f. MySQL Connector/NET Developer Guide.
https://dev.mysql.com/doc/connector-net/en/

g. C# documentation.
https://docs.microsoft.com/en-us/dotnet/csharp/