Informatics College Pokhara



Application Development

CS6004NP

Coursework 1

**Submitted by Submitted to**

Name: Kriti Gurung Name: Ishwor Sapkota

ID: 17030720 Application Development

Group: L3C1

Date: 2020/01/10

**Table of Contents**

[**1.** **Introduction** 1](#_Toc29588171)

[**2.** **User Manual** 2](#_Toc29588172)

[**2.1** **Execution of program** 2](#_Toc29588173)

[**2.2** **Login** 2](#_Toc29588174)

[**2.3** **Home Window** 4](#_Toc29588175)

[**2.4** **Enroll Student** 5](#_Toc29588176)

[**2.5** **Import Record** 6](#_Toc29588177)

[**2.6** **Generate Report** 7](#_Toc29588178)

[**2.7** **Generate Weekly Report** 8](#_Toc29588179)

[**2.8** **Generate Chart Diagram** 9](#_Toc29588180)

[**3.** **Flow Chart** 10](#_Toc29588181)

[**4.** **Method Description** 11](#_Toc29588182)

[**5.** **Testing** 15](#_Toc29588183)

[**5.1** **Testing 1** 15](#_Toc29588184)

[**5.2** **Testing 2** 18](#_Toc29588185)

[**5.3** **Testing 3** 19](#_Toc29588186)

[**5.4** **Testing 4** 20](#_Toc29588187)

[**5.5** **Testing 5** 22](#_Toc29588188)

[**5.6** **Testing 6** 25](#_Toc29588189)

[**5.7** **Testing 7** 26](#_Toc29588190)

[**5.8** **Testing 8** 27](#_Toc29588191)

[**5.9** **Testing 9** 28](#_Toc29588192)

[**5.10** **Testing 10** 29](#_Toc29588193)

[**6.** **Conclusion** 30](#_Toc29588194)

[**7.** **Appendix** 30](#_Toc29588195)

**List of Figures**

[Figure 1: Login Page of system 2](#_Toc29588231)

[Figure 2: Invalid username error 3](#_Toc29588232)

[Figure 3: Invalid password error 3](#_Toc29588233)

[Figure 4: Home page 4](#_Toc29588234)

[Figure 5: Student registration page 5](#_Toc29588235)

[Figure 6: Import CSV file 6](#_Toc29588236)

[Figure 7: Report generation 7](#_Toc29588237)

[Figure 8: Weekly report 8](#_Toc29588238)

[Figure 9: Display Pie chart 9](#_Toc29588239)

[Figure 10: Flowchart of enrollment of students 10](#_Toc29588240)

[Figure 11: Testing 1 15](#_Toc29588241)

[Figure 12: Testing 1.1 16](#_Toc29588242)

[Figure 13: Testing 1.2 17](#_Toc29588243)

[Figure 14: Testing 2 18](#_Toc29588244)

[Figure 15: Testing 3 19](#_Toc29588245)

[Figure 16: Testing 4 20](#_Toc29588246)

[Figure 17: Testing 4.1 21](#_Toc29588247)

[Figure 18: Testing 5 22](#_Toc29588248)

[Figure 19: Testing 5.1 24](#_Toc29588249)

[Figure 20: Testing 5.2 24](#_Toc29588250)

[Figure 21: Testing 6 25](#_Toc29588251)

[Figure 22: Testing 7 26](#_Toc29588252)

[Figure 23: Testing 8 27](#_Toc29588253)

[Figure 24: Testing 9 28](#_Toc29588254)

[Figure 25: Testing 10 29](#_Toc29588255)

# **Introduction**

Student Information System is a desktop application which is designed and implemented in C#. This system is responsible for tracking the student details, module selected registration, registration date, weekly reports and so on. The application allows the user to input the student personal detail including registration date so that a system can generate a weekly enrolment report of the student. System includes detail like name, address, contact no, email, program enroll, registration date. Moreover, this system brings ease to keep the record of the students.

Some of the key features of this system are listed below:

* A login window.
* Register or enroll new students.
* Create a new xml file consisting of student registration details.
* Display registered students in grid view.
* Import a CSV file and update the xml file of student registration details.
* Generate reports on the basis of requirements.
* Report could be sorted by student first name or registration date.
* Display the total number of students enrolled in a program.
* Display a pie chart of total number of student on each program.

# **User Manual**

For using the system properly instructions are given below with pictorial descriptions.

## **Execution of program**

In order to run the program click on ‘StudentInformationSystem.exe’ file. This file is located inside ‘StudentInformationSystem/bin/debug’ folder.

## **Login**

When the system loads a login form is displayed on the screen. The form asks the user to enter username and password. The username and password for this system is admin. When the username or password is entered incorrectly, the system displays a dialogue box for the error.

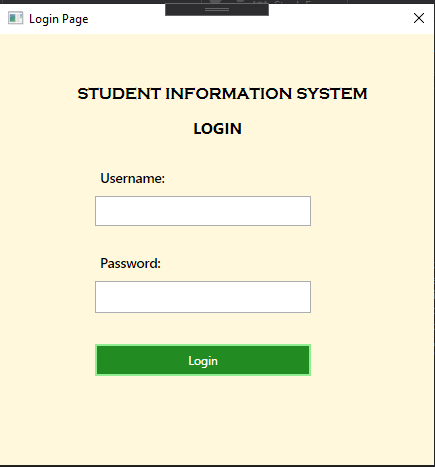


Figure 1: Login Page of system

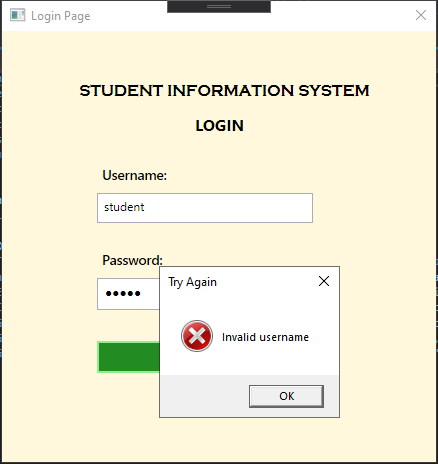


Figure 2: Invalid username error

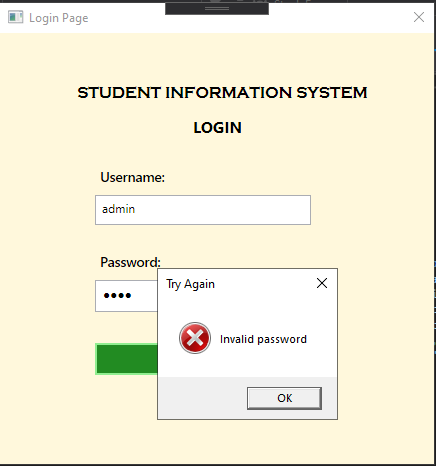


Figure 3: Invalid password error

## **Home Window**

This is the main form of the system. It consists of a group of button to navigate through next forms. There are five buttons to navigate through pages which are: enroll student, import record, generate record, generate report, generate weekly report and generate pie chart diagram. Each have their own functions.

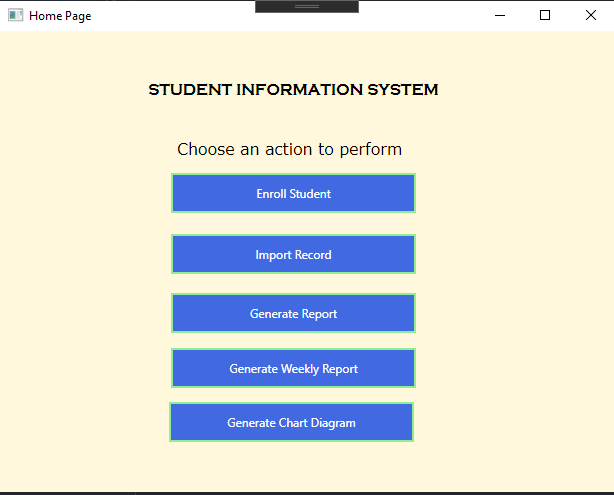


Figure 4: Home page

## **Enroll Student**

This form is responsible for registration of new students. It contains text input text boxes for registration id, registration date, student id, student name, student address, student contact number and module selected. All the fields are must except course. There is also a reset button which is to clear all the values of the fields. After the entering of genuine data, the data is displayed in the grid below the form.

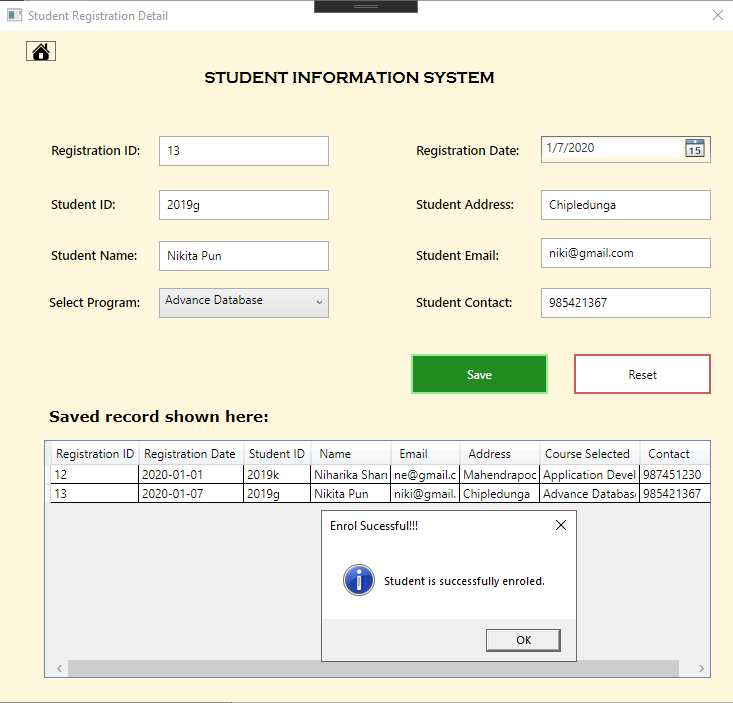


Figure 5: Student registration page

## **Import Record**

Inside this page, a text file consisting of a bulk data which is in the CVS format can be imported. A green button named as import file helps to import the file. The data is updated to the existing XML file of student details. If the file is not in CVS format then it is not able to import the details. So the file should be in CSV format.

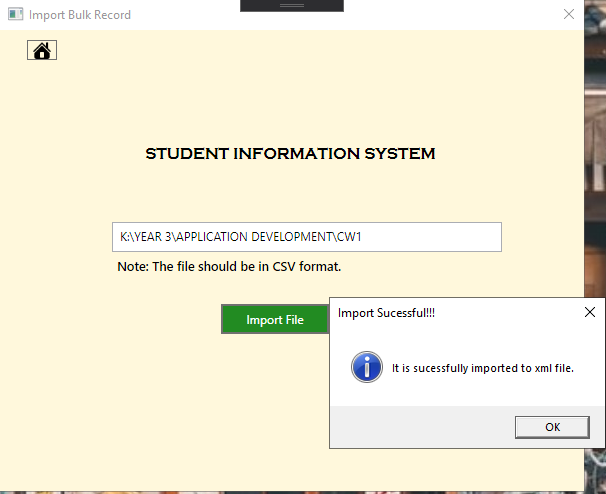


Figure 6: Import CSV file

## **Generate Report**

After the registration of a student, a XML file is created. With the help of XML the report is created otherwise it is unable to show the report. The report is shown in grid view. The display can be done by two methods. One is sorted by the first name and another by registration date. For sorting two radio buttons are provided and a button for generating report.

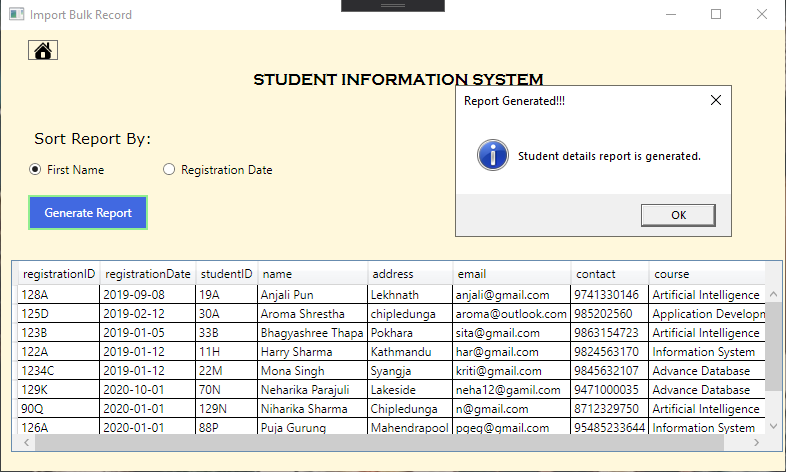


Figure 7: Report generation

## **Generate Weekly Report**

It also displays a report of the students with the help of XML data. It shows the weekly tabular report showing total number of students enrolled so far in each program offered by the institution. The data is retrieved and shown in the grid view. If there is no XML file then it shows an error message for not being able to generate the record.

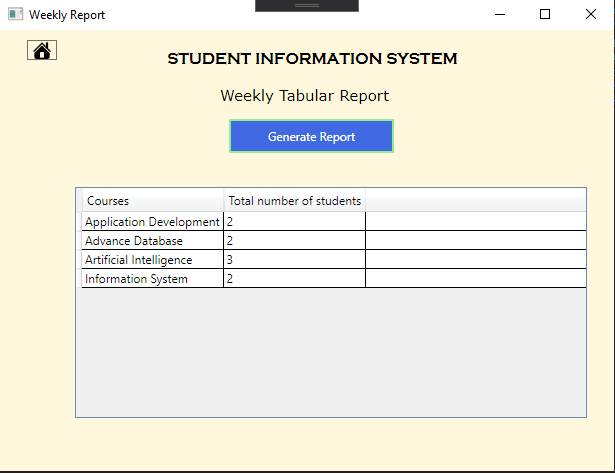


Figure 8: Weekly report

## **Generate Chart Diagram**

With the help of the data from XML, a pie chart is displayed showing total number of student on each program. When the cursor is hovered on the chart one can see the total number of students and percentage of student enrolled in the course.

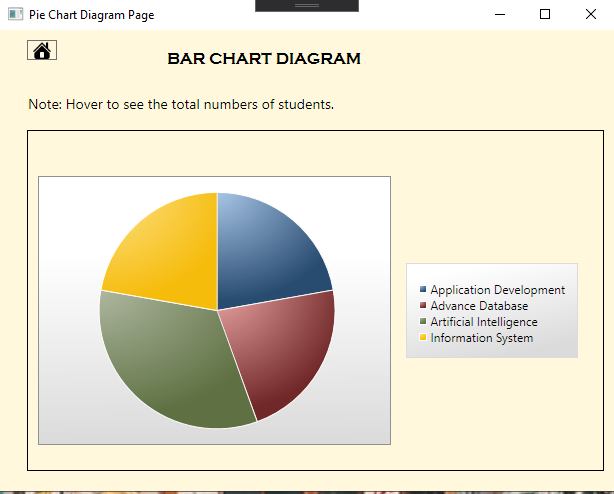


Figure 9: Display Pie chart

# **Flow Chart**

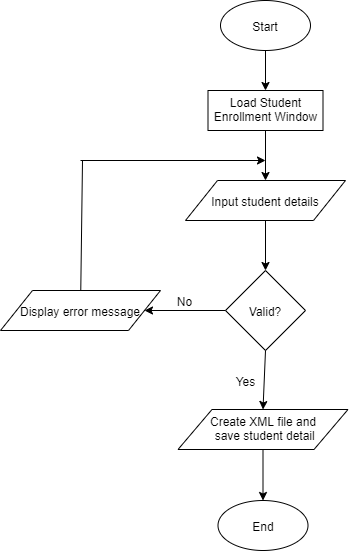


Figure 10: Flowchart of enrollment of students

In the above figure, a flow chart of student enrollment is given. Firstly, the system displays the student enrollment form. The user will input the details. If the details are left blank or not valid then the system displays a dialog box of the error message. If the details are valid, the system creates a XML file and saves the details to the file.

# **Method Description**

* **Login Page**

|  |  |
| --- | --- |
| **Method** | **Description** |
| login\_button | If the username and password matches then it navigates to home page. When the username and password mismatches then an error message is displayed. |

* **Home Page**

|  |  |
| --- | --- |
| **Method** | **Description** |
| navigate\_to\_registration | This method navigates to student registration page. |
| import\_bulk\_record | This method navigates to CSV file import page. |
| generate\_report | This method navigates to report generation page. |
| generate\_weekly\_report | This method navigates to weekly report generation page. |
| generate\_bar\_diagram | This method navigates to pie chart generation page. |

* **Student Registration Details Page**

|  |  |
| --- | --- |
| **Method** | **Description** |
| saveStdDetails | Firstly, it gets the user input details and checks whether the data is null or not. After the data is validated a XML file is created and the data is displayed to data grid. |
| AppendData | It appends the data to the current existing XML file. The data is de serialized first and serialized. After that it is added to the existing XML file. |
| Button\_Click | It navigates to home page. |
| btn\_reset\_Click | Clears the text fields. |

* **Import Bulk Record**

|  |  |
| --- | --- |
| **Method** | **Description** |
| btn\_save\_csv\_file\_Click | It opens the dialogue box and the csv file is read by the stream reader. Then is breaks it and adds to defined tags in the existing XML file. |
| Button\_Click | It navigates to home page. |

* **Generate Report**

|  |  |
| --- | --- |
| **Method** | **Description** |
| generateStudentReport | It reads the XML file with the help of data set and displays in grid. The data is also sorted by first name and registration date. There are two radio buttons for selecting the sorting process. |
| Button\_Click | It navigates to home page. |

* **Weekly Report**

|  |  |
| --- | --- |
| **Method** | **Description** |
| generate\_weekly\_report | It reads the XML file with the help of data set. It counts the total number of students on each course. And it is displayed to data grid. |
| Button\_Click | It navigates to home page. |

* **Chart Diagram**

|  |  |
| --- | --- |
| **Method** | **Description** |
| ChartDiagram | It reads the XML file with the help of data set. It counts the total number of students on each course. And it is generates a pie chart. The pie chart is created with the help of installing a package of data visualization. |
| Button\_Click | It navigates to home page. |

# **Testing**

## **Testing 1**

**Objective:** To display error message when the username or password is not valid.

**Output:** Displays error message for the invalid username or password.

**Result:** Successful.

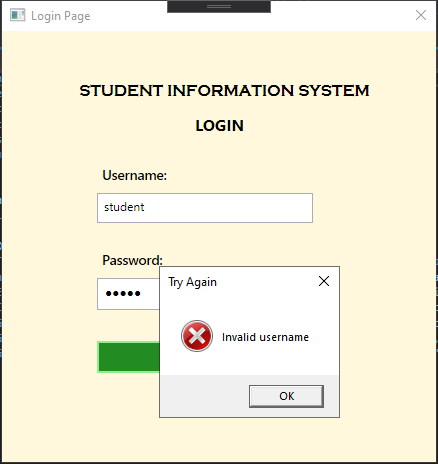


Figure 11: Testing 1

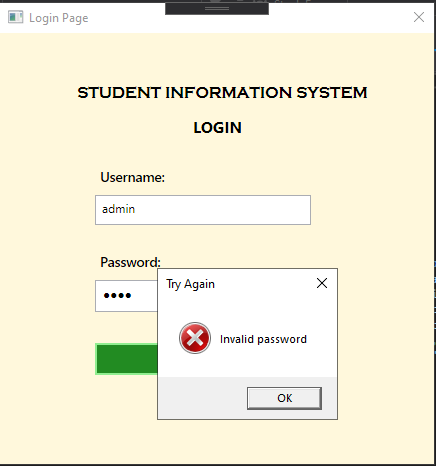


Figure 12: Testing 1.1

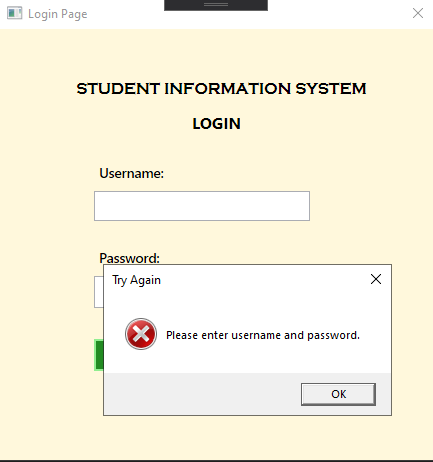


Figure 13: Testing 1.2

## **Testing 2**

**Objective:** To display success message and navigate to home page when the username and password matches.

**Output:** Displays successful message and navigates to home page.

**Result:** Successful.

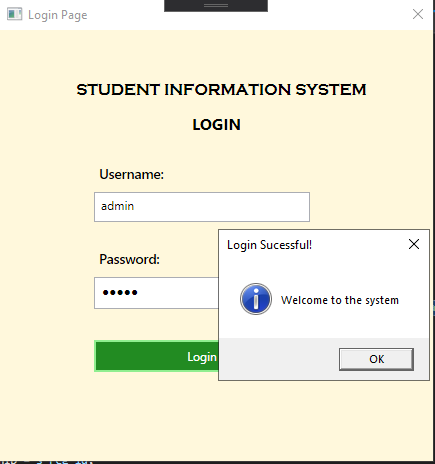


Figure : Testing 2

## **Testing 3**

**Objective:** To display error message when the user leaves text fields empty in student enroll page.

**Output:** Displays error message for the invalid input.

**Result:** Successful.

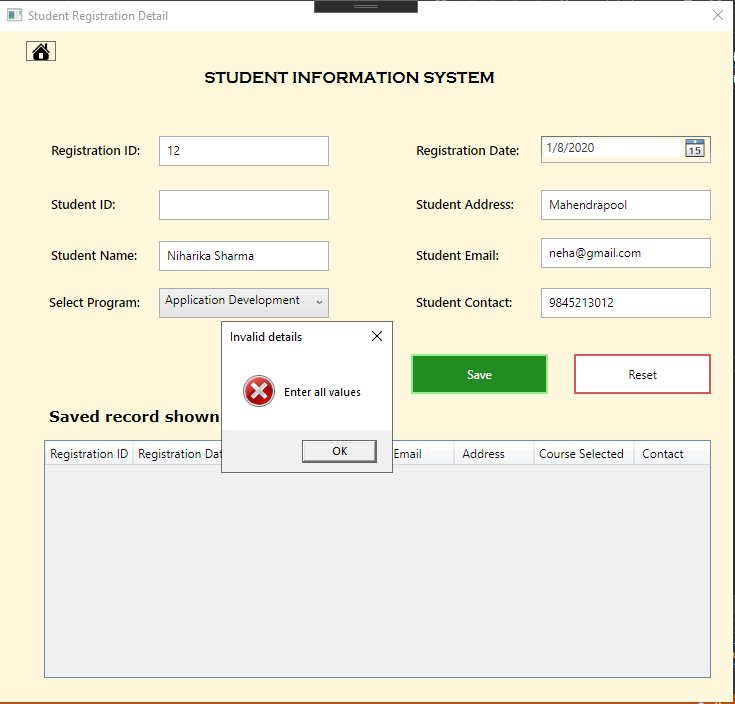
****

Figure 15: Testing 3

## **Testing 4**

**Objective:** To check whether XML file consisting data of student’s details or not.

**Output:** XML file has been created with the details provided by user.

**Result:** Successful.

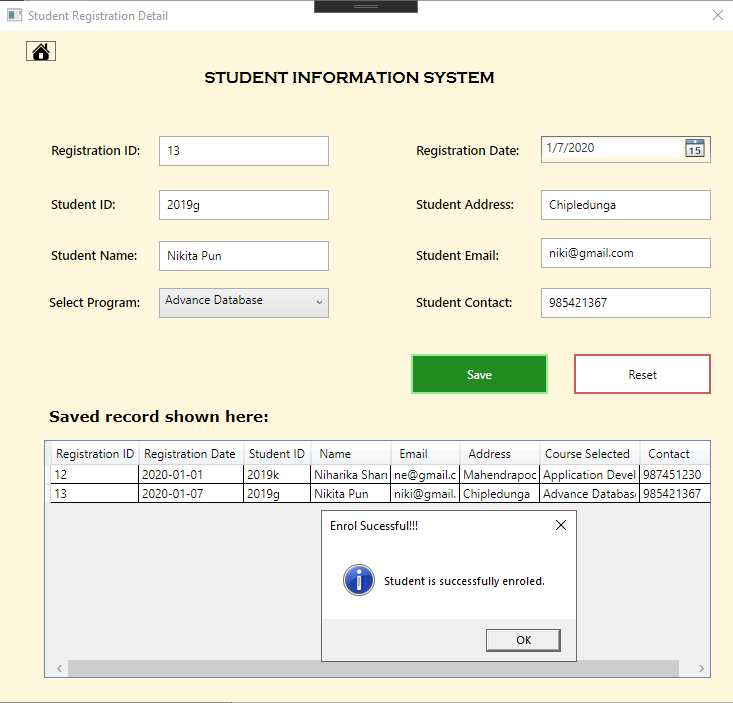


Figure 16: Testing 4

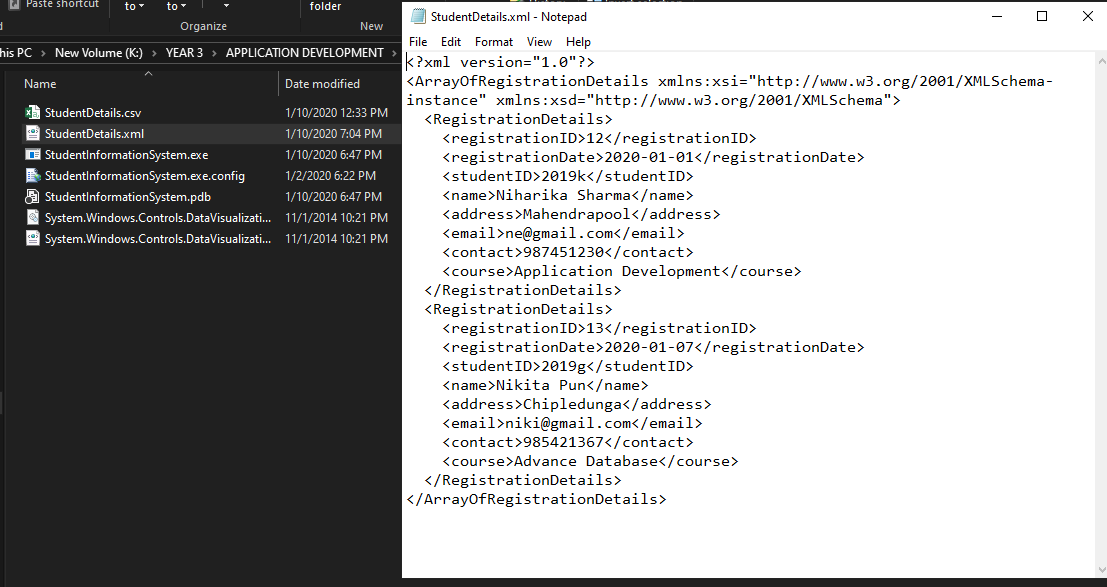


Figure 17: Testing 4.1

## **Testing 5**

**Objective:** Check the bulk data of csv file is added to existing XML.

**Output:** The bulk data is added to XML successfully.

**Result:** Successful.

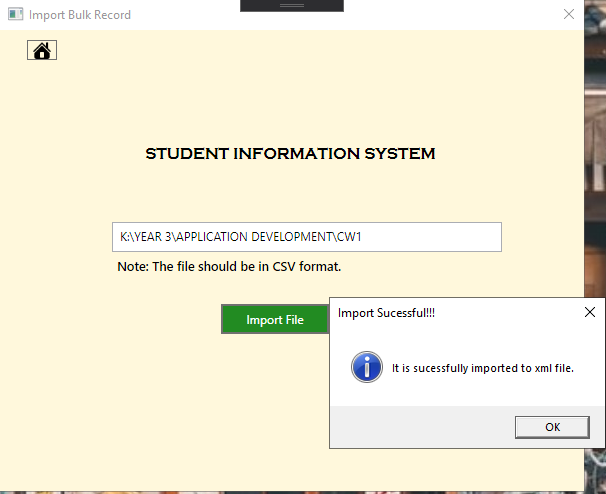
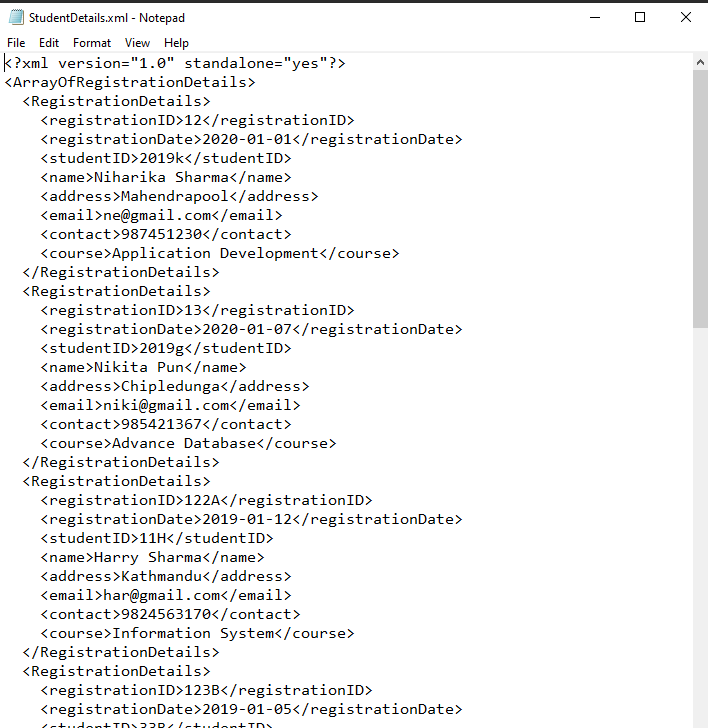


Figure 18: Testing 5



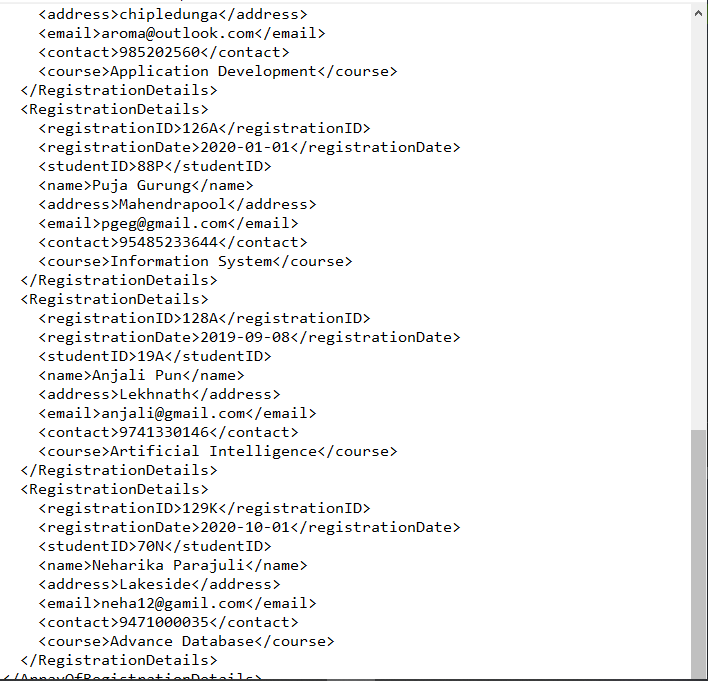


Figure 19: Testing 5.1

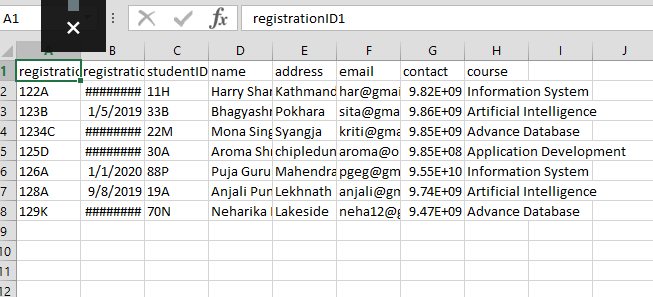


Figure 20: Testing 5.2

## **Testing 6**

**Objective:** Check if the report is sorted by first name and displayed to grid.

**Output:** Displays the data with the first name sorting.

**Result:** Successful.

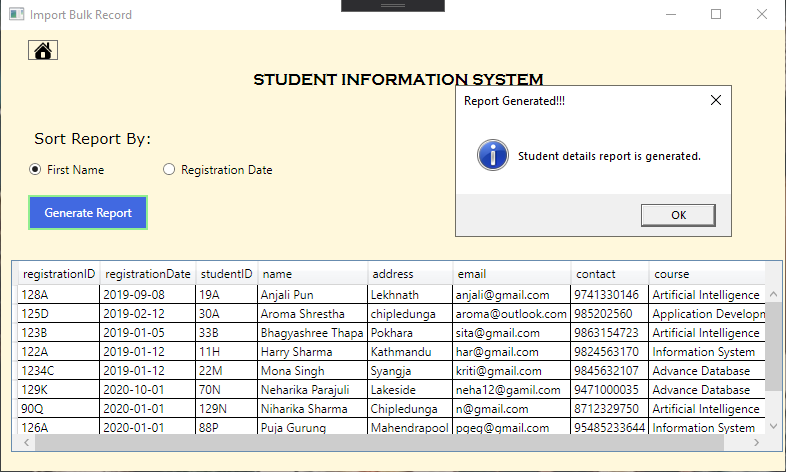


Figure 21: Testing 6

## **Testing 7**

**Objective:** Check if the report is sorted by date and displayed to grid.

**Output:** Displays the data with the date sorting.

**Result:** Successful.

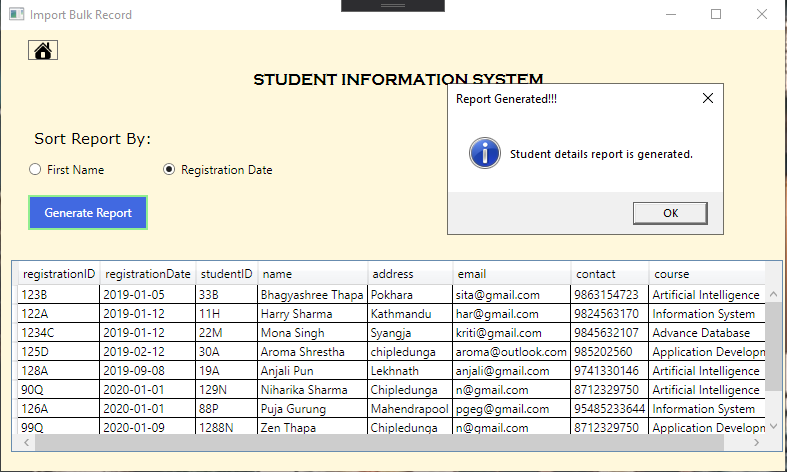


Figure 22: Testing 7

## **Testing 8**

**Objective:** Check if pie chart is created with the xml data.

**Output:** Displays pie chart.

**Result:** Successful.

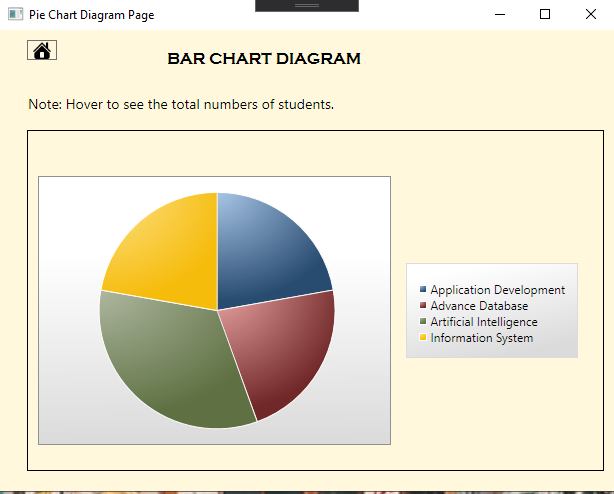


Figure 23: Testing 8

## **Testing 9**

**Objective:** Display error message when there is no xml file for generating report.

**Output:** Displays error message.

**Result:** Successful.

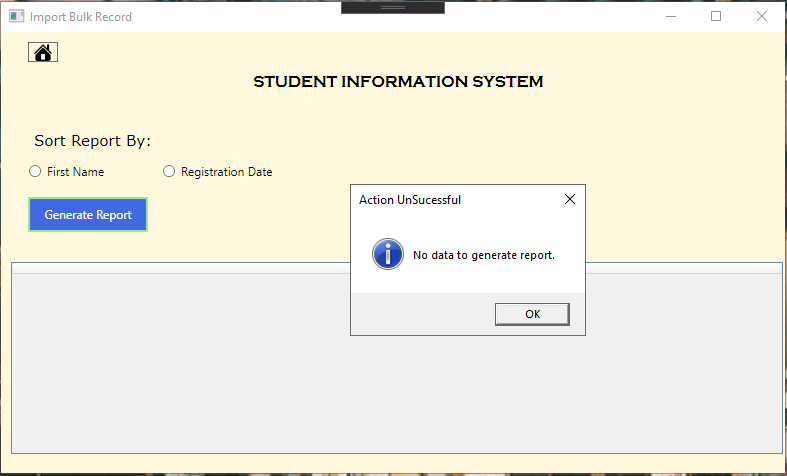


Figure 24: Testing 9

## **Testing 10**

**Objective:** Display error message when there is no xml file for generating pie chart.

**Output:** Displays error message.

**Result:** Successful.

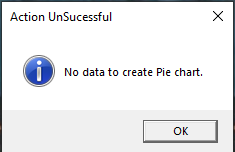


Figure 25: Testing 10

# **Conclusion**

# **Appendix**