FYP Management System

Session: 2022 - 2026

Submitted By:

Fasi-ur-Rehman 2022-CS-71

Supervised By:

Sir Nazeef Ul Haq

Department of Computer Science
University of Engineering and Technology
Lahore Pakistan

Contents

List of Figures

1 Introduction

1.1 Overview

The Final Year Management System is a project designed to streamline the management of final-year projects. It handles a range of tasks involved in project management, such as forming groups for the final year project or disbanding them as needed. Whether you're adding or removing students, keeping track of advisors, or supervising projects, this system is here to simplify every step.

One of its features is the ability to assign projects to specific groups, ensuring a well-organized approach to project distribution. Furthermore, the system allows for the assignment of multiple advisors to a single project, providing comprehensive guidance and support. Essentially, it acts as a complete solution, streamlining the complexities of managing final-year projects.

The system also offers a user-friendly feature for generating PDF reports. These reports help the committee to streamline various aspects of final-year projects.

1.2 Significance

This project is aimed at managing the activities of students, projects, advisors, and their relationships with each other. All the data related to these activities is stored in an SQL database. The significance of this project is that it uses a more powerful tool like an SQL database over traditional text files or CSV files to manage data. In the previous projects, data management was done through text files or CSV, which had limited features. However, an SQL database provides better features some of which are:

- It offers extensive storage capacity for data on secondary storage devices.
- Databases offer powerful querying languages (e.g., SQL) that enable complex searches, and filtering and make managing data more easy and efficient.

2 Functionalities

Now, let's delve into the detailed functionalities of the Final Year Project (FYP) Management System.

2.1 Manage Students

Within this module, we handle all aspects related to student information. This includes the addition of new students, updating their details, and deleting records while keeping records of those students within the database. Additionally, the system facilitates efficient searching for students based on specific criteria or information. This functionality streamlines the student management process and ensures the integrity and accessibility of their information within the database.

2.1.1 Add Student

We add information about students. This information involves their first name, last name (optional), Registration number contact(optional) information, email, and gender(optional).

They are added to the project database tables. The addition of data involves two tables: the **Student** table and the **Person** table. The student table consists of the student's registration number and ID. The rest of the information is stored in the person table. Both tables are mapped together using their IDs. There is also a **Look Up** table which has an id for the gender, which is 1 for male and 2 for female. The ID of the person table, which is auto-generated, is given to the student table for storing the registration number of relevant student information in the student table. In this way, We add student information to these two tables.

The **validation** in this functionality is that we can not add two students with the same registration numbers, and we can not leave compulsory attributes null.

2.1.2 Update Student

This functionality updates the information of specific students. The tuple is selected from the data grid view, and the text boxes are filled with that tuple's information. We can change the needed information and then update it.

In the backend, the ID of the selected tuple will be retrieved from the table, and the data will be updated from the database by matching that ID.

There is also a search feature that ensures that we can search for the needed student, so you don't have to scroll through extensive data.

In validation again, we can not leave the compulsory attributes null, and we can not change the registration number to the same number that is already in the database.

2.1.3 Delete Student

In this functionality, we ensure that the student is deleted from the user interface for the user, but it is kept in the records or database in case it is ever needed. The user have to put registration number of student to be deleted and press delete button then it will perform deletion.

For this purpose, we update the student's registration number and put a steric (*) in front of it. While showing data, we ignore the tuples with steric in front of it.

In this way, we delete the student for the user while keeping the actual record as it is.

2.2 Manage Advisor

Manage advisor functionality is similar to manage student functionality. It also involves the use of three tables. **Advisor table**, **Person Table** and **Look up table**.In this look-up table, have the designation of the advisor.

2.2.1 Add Advisor

In this, we map the person table and advisor table as we did in the add student functionality. We add the auto-generated ID of the person table to the advisor table. The advisor table has designation and salary(optional). All other attributes are the same as they were in student except for the registration number.

The **validation** of this functionality is that compulsory attributes can not be left alone. Other than that, only available designations can be selected. Data of the required data type of column can be added to the table.

2.2.2 Update Advisor

This functionality works the same as the updation of the students. Also in this we need to select the tuple of data we need to update and the text boxes will be filled with the data then it can be changed according to the need.

Search functionality is also added in this so that users don't have to scroll through a large database.

Validations of adding advisor hold even when we update the advisor, i.e., compulsory attributes can not be left alone. Other than that, only available designations can be selected. Data of the required data type of column is added to the table.

2.2.3 Delete Advisor

This work is the same as in the deletion of students. The data is deleted only from the user interface while it is kept in the record of the database. The user have to select the tuple to be deleted from the data grid view and then press delete button in this way deletion is perform on advisor.

This is achieved by putting steric (*) in front of the advisor's First name, as all other attributes of the Advisor table are integer types.

While displaying the data advisors with steric in front of their first name are ignored. In this way, deletion of Advisor is done.

2.3 Manage Projects

Projects are managed using only one table named Project. In this description and title attributes are present. There is also an id attribute, but that is auto-generated.

2.3.1 Add Project

In this projects are added into the table using insert SQL query just as it was done in previous functionalities.

In this validation implemented, the title can not be empty as it is not a null attribute in the database, and the description is kept optional as it can be null in the database. Other than that, extra validation

is added, which ensures that the project titles should be unique so that they will not cause any conflict in managing the projects.

2.3.2 Update Project

The updation of the Project is the same as it was in previous functionalities. Select the tuple to be deleted from the data grid view and then the text boxes will be filed by the data and it can be updated as needed.

While updation the validations are kept as it is.

2.3.3 Delete Project

This work is the same as in the deletion of previous functionalities. The user has to select the tuple to be deleted from the data grid view and then press the delete button. In this way, deletion is performed on the projects.

This is achieved by putting steric (*) in front of the project title. As it is a compulsory attribute while description can be null, so, we put steric in front of the title.

While displaying the data projects with steric in front of their titles are ignored. In this way, deletion of project is done.

2.4 Formation of Groups

In group formation, three tables are involved. **Group**, **StudentGroup** and **Student table**.

- **Group** table has attribute id, which is auto-generated, so it is not needed from the user. Another attribute is created-on, which is the date on which the group was created, and it is the current date when it was formed, so it is also not needed from the user.
- **GroupStudent** has an attribute GroupId which is given by the Group table. StudentId attribute is given by the Student table. Other than that it has status attribute which has id mapped on look up table. It can be active or inactive.

2.4.1 Formation of Group

By creating a Group table tuple, we give the value of its id to the GroupStudent table, and then we add multiple student IDs against each group ID in the GroupStudent table, and in this way, the group is formed.

In UI we ask the user to enter the registration number of multiple students in comma-separated form and in the backend, we split the array and then find the ids of those registration numbers and add them against a single group id in the GroupStudent table and in this way group is formed.

2.4.2 Removal of student from group

Students can be removed from the group by turning their status to inactive, and while displaying, we ignore the inactive students. In this way, the students are removed from the group, and the data about their previous groups are kept saved. Furthermore, when a student is deleted from the Student deletion functionality, as explained in section 2.1.3, they automatically become inactive in the group.

2.4.3 Deletion of a Group

For deleting a group we make every student of that group inactive so that they will not be displayed but they will be kept in the database for record.

2.5 Assignment of Project to a Group

In this **Project**, **GroupProject** and **Group** tables.

- **Project** Table has titles, descriptions, and IDs of the project.
- **Group** Table has the id and date on which the group was created.
- **GroupProject** Table has attributes of ProjectId, GroupId, and Assignment date. ProjectId is taken from the Project table and GroupId is taken from the Group table. The assignment date is the date on which the project is assigned, and that is taken when the group is created on the application.

2.5.1 Working

In this, when we want to assign the project to a group, the user needs to select an available group ID and project ID of available projects. Then, when both are selected, they are inserted into the GroupProject table.

The data is validated while inserting. The **validation** involves checking whether the group ID and project ID are valid and whether a project is assigned to a group or not. A group can not have more than one project, and more than one group can not have the same project.

2.6 Assignment of Multiple Advisors

Advisors are assigned to projects using **Advisor**, **ProjectAdvisor**, and **Project** tables.

- **Advisor** table has Advisor ID, designation, and salary.
- **Project** Table has Project ID, Description and title.
- **GroupAdvisor** Table has AdvisorId, ProjectId, AdvisorRole, and assignment date. AdvisorId is retrieved from the Advisor table, ProjectId is retrieved from the Project table, and the Advisor role is of integer type. It is mapped with the id of the lookup table, and its value can be retrieved from there. The assignment date is the date when the advisor was assigned.

2.6.1 Working

In this, when we want to assign an advisor to a project, the user needs to select a project and advisor. Then, the user selects the role of that advisor in the project, and then it is inserted into the GroupAdvisor table. In this way, multiple advisors can be assigned to a single project.

The **validation** of data while inserting involves checking that the same advisor should not be assigned to a group more than one time. Different advisors can be assigned to the same project, and the same advisors can be assigned to different projects, but the same advisor can not be assigned to the project more than one time.

2.7 Manage Evaluations

Manage Evaluation involves an Evaluation table. It has evaluation ID, which is an auto-generated attribute, the Name of the evaluation, the TotalMarks of evaluation, and its total weightage.

2.7.1 Working

In this, users have to enter the name of the evaluation, and then it will enter the total marks and weightage of the student.

Validation involves making sure that total marks and weightage are of integer type and none of the attributes is null. In this, we also make sure that weightage is greater and, less than 100. Total marks can not be negative.

2.8 Mark Evaluation

Marking of evaluation involves **Evaluation**, **GroupEvaluation**, and **Group** Tables.

- **Group** Table has the id and date on which the group was created.
- **Evaluation** Table has evaluation ID, which is an auto-generated attribute, the Name of the evaluation, the TotalMarks of evaluation, and its total weightage.
- **GroupEvaluation** Table has GroupId, EvaluationId, ObtainedMarks, and EvaluationDate. GroupId is taken from Group Table and EvaluationID is taken from Evaluation Table. Obtained Marks are entered by the user. EvaluationDate is the current date when the evaluation was marked.

2.8.1 Working

Data is inserted into the GroupEvaluation table by selecting a group and the evaluation, then entering the obtained marks of the group in that evaluation. The result is calculated by dividing the obtained marks by the total marks and then multiplying it by the weightage of that evaluation.

Validation involves checking that obtained marks should be less than the total marks of that evaluation. Other than that, a group can be evaluated against multiple evaluations, but it can not be evaluated against the same evaluation twice.

3 Tools Used

A set of powerful and versatile tools facilitated the development and implementation of this project.

3.1 Visual Studio

Description: Visual Studio served as the primary integrated development environment (IDE) for this project. Its user-friendly interface provided an excellent platform for coding, debugging, and project management.
Role: Visual Studio played a pivotal role in creating, editing, and organizing the code.

3.2 WinForms (C# .NET)

Description: The project utilized WinForms, a graphical user interface (GUI) toolkit within the Microsoft .NET framework using C#.
Role: WinForms in C# .NET enabled the creation of a visually appealing and interactive user interface, enhancing the overall user experience in the Final Year Project Management System.

3.3 PDFsharp

Description: PDFsharp, a popular open-source library for creating and processing PDF documents, played a key role in generating PDF reports within the project.
Role: The integration of PDFsharp allowed the project to dynamically create detailed SQL reports in the PDF format, offering a convenient and standardized way to analyze various aspects of the final year projects.

4 User Interface

The user interface of this application is structured with a multipage layout, and users can navigate between these pages using various buttons conveniently located in the side panel. The design ensures a user-friendly and organized experience. Additionally, some buttons in the side panel incorporate drop-down menus, providing users with extended options for more nuanced interactions.

Each button on the side panel corresponds to a specific page or user control, and by selecting a particular button, users are directed to the associated page.

4.1 Main Window

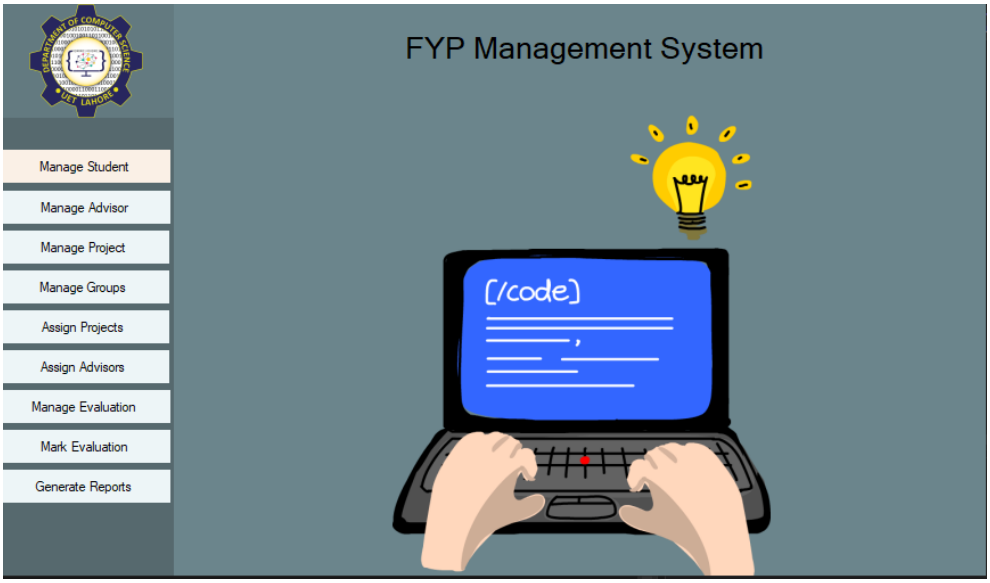

Figure 1: Main window

The main window consists of all the buttons, and by clicking them, the user's control opens. Some buttons have a drop-down menu to select sub-options for managing students, you can select any of the four options from there, which are added, delete, update, and view. The image of the main window is given below.

4.2 Manage Evaluation Screen

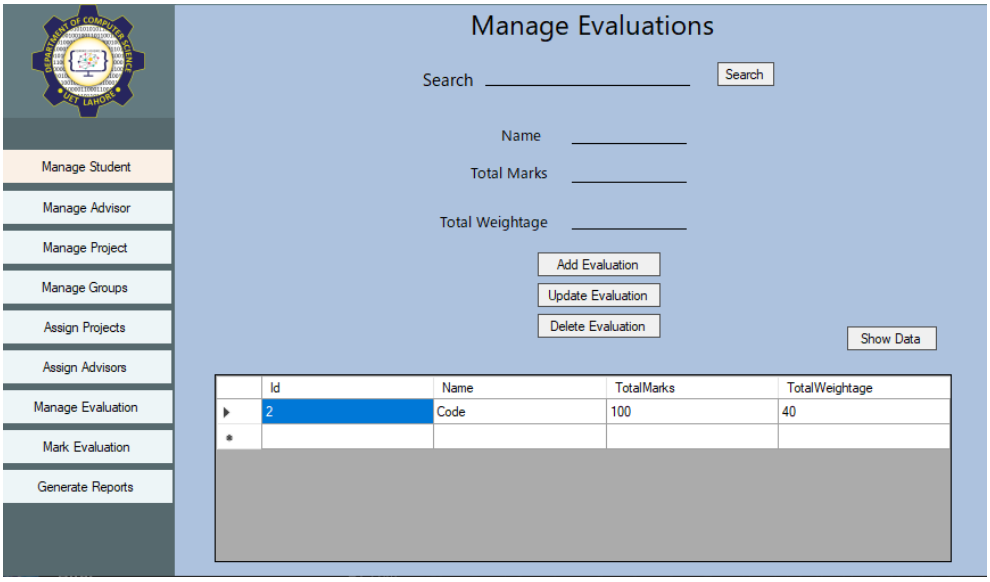

Figure 2: Manage Evaluation window

In this window, the user can add data about a specific evaluation data, including the name of the evaluation, the Total marks of that evaluation, and the weightage it has. Users can fill the text boxes with the information and then add it. There is also a grid view from where the user can select a tuple and update it or delete it.

4.3 Mark Evaluation

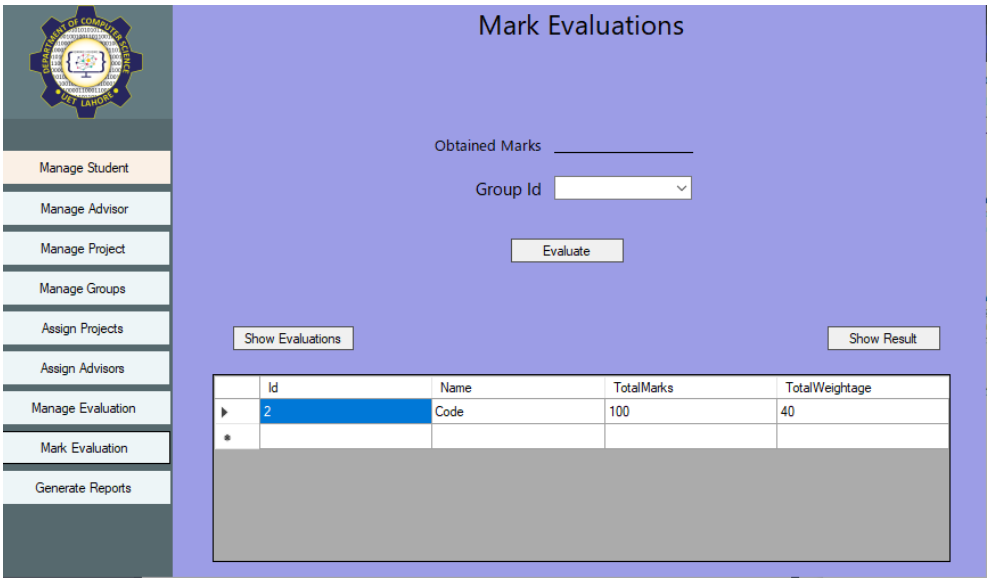

Figure 3: Manage Evaluation window

In this window, the user can group the marks obtained in each evaluation, which allows us to calculate the student's evaluation result. A single group can not be evaluated more than once. You have to write the obtained marks in the text box. Then, you can select a group ID from the combo box and select an evaluation against which you want to mark the student. Then, press the evaluate button to fill in the data in the Group Evaluation table. You can also switch the data in the grid to view evaluated data.

4.4 Form Group

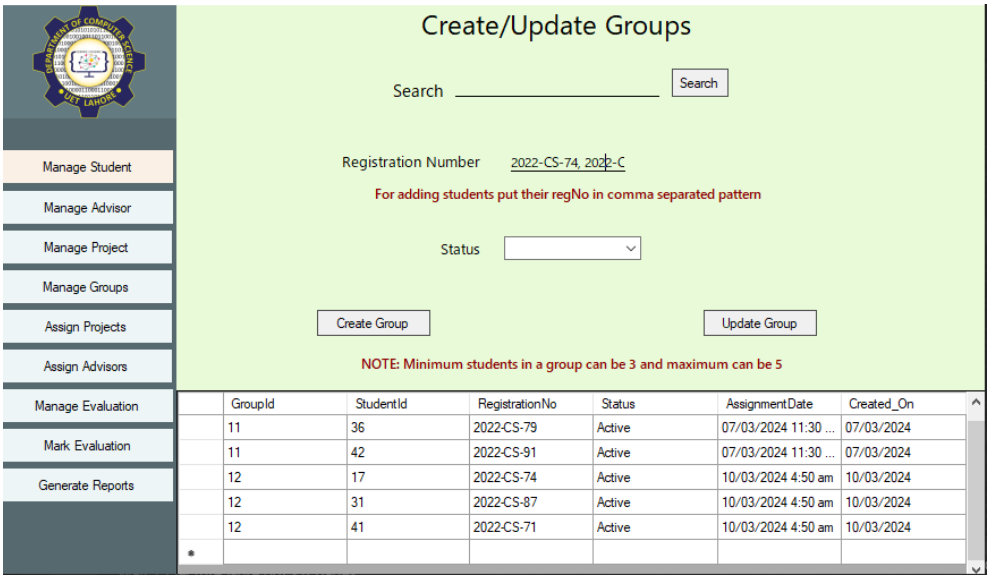

Figure 4: Manage Evaluation window

On this page, you can form a group of multiple students. The maximum number of students you can group together is 5, and the minimum is 3. To group students together, you have to put the registration numbers of students in comma-separated form in the text box and then press the Create button. You can also add more students or make them inactive through the update button. There is also a search button through which you can search all students.

5 Generated Reports

Following are the reports generated using the SQL queries, which could be useful for the Committee.

5.1 Report 1

List of projects along with advisory board and list of students.

List of projects along with advisory board and list of students

RegistrationNo	Student Name	Student Email	Project Title	Advisor Name	Advisor Email
2022-CS-56	Anas	anasmustafa123@gmail.com	Library Management System	Nazeef ul Haq	nazeefulhaq@uet.edu.pk
2022-CS-56	Anas	anasmustafa123@gmail.com	Library Management System	Irzam	irzamliaquat@uet.edu.pk
2022-CS-56	Anas	anasmustafa123@gmail.com	Library Management System	Laeeq	laeeqkhanniazi@uet.edu.pk
2022-CS-57	AbdulRehman	abdulrehmanirfan2238@gmail.com	Library Management System	Nazeef ul Haq	nazeefulhaq@uet.edu.pk
2022-CS-57	AbdulRehman	abdulrehmanirfan2238@gmail.com	Library Management System	Irzam	irzamliaquat@uet.edu.pk
2022-CS-57	AbdulRehman	abdulrehmanirfan2238@gmail.com	Library Management System	Laeeq	laeeqkhanniazi@uet.edu.pk
2022-CS-59	Muhammad	madnanbirmani@gmail.com	Library Management System	Nazeef ul Haq	nazeefulhaq@uet.edu.pk
2022-CS-59	Muhammad	madnanbirmani@gmail.com	Library Management System	Irzam	irzamliaquat@uet.edu.pk
2022-CS-59	Muhammad	madnanbirmani@gmail.com	Library Management System	Laeeq	laeeqkhanniazi@uet.edu.pk
2022-CS-60	Noor	noorulhuda@gmail.com	Library Management System	Nazeef ul Haq	nazeefulhaq@uet.edu.pk
2022-CS-60	Noor	noorulhuda@gmail.com	Library Management System	Irzam	irzamliaquat@uet.edu.pk
2022-CS-60	Noor	noorulhuda@gmail.com	Library Management System	Laeeq	laeeqkhanniazi@uet.edu.pk
2022-CS-61	Aaliya	aaliyakhalil2022@gmail.com	Library Management System	Nazeef ul Haq	nazeefulhaq@uet.edu.pk
2022-CS-61	Aaliya	aaliyakhalil2022@gmail.com	Library Management System	Irzam	irzamliaquat@uet.edu.pk
2022-CS-61	Aaliya	aaliyakhalil2022@gmail.com	Library Management System	Laeeq	laeeqkhanniazi@uet.edu.pk
2022-CS-61	Aaliya	aaliyakhalil2022@gmail.com	Blood Donation Management System	Awais	awaishasan@uet.edu.pk
2022-CS-62	Aqsa	aqsabatool6766@gmail.com	Blood Donation Management System	Awais	awaishasan@uet.edu.pk
2022-CS-63	Faisal	faisalilyas2005@gmail.com	Blood Donation Management System	Awais	awaishasan@uet.edu.pk
2022-CS-65	Wali	waliaslam2002@gmail.com	Blood Donation Management System	Awais	awaishasan@uet.edu.pk
2022-CS-66	Bisma	bismafajar816@gmail.com	Blood Donation Management System	Awais	awaishasan@uet.edu.pk

5.1.1 Report 2

Marks sheet of projects that shows the marks in each evaluation against each student and project

Marks Sheet

RegistrationNo	Student Name	Project Title	Evaluation Name	TotalMarks	TotalWeightage	ObtainedMarks	Result
2022-CS-56	Anas	Library Management System	Code formatation	100	10	90	9
2022-CS-56	Anas	Library Management System	Project Completion	100	40	91	36.4
2022-CS-56	Anas	Library Management System	Project Idea	100	10	79	7.9
2022-CS-57	AbdulRehman	Library Management System	Code formatation	100	10	90	9
2022-CS-57	AbdulRehman	Library Management System	Project Completion	100	40	91	36.4
2022-CS-57	AbdulRehman	Library Management System	Project Idea	100	10	79	7.9
2022-CS-59	Muhammad	Library Management System	Code formatation	100	10	90	9
2022-CS-59	Muhammad	Library Management System	Project Completion	100	40	91	36.4
2022-CS-59	Muhammad	Library Management System	Project Idea	100	10	79	7.9
2022-CS-60	Noor	Library Management System	Code formatation	100	10	90	9
2022-CS-60	Noor	Library Management System	Project Completion	100	40	91	36.4
2022-CS-60	Noor	Library Management System	Project Idea	100	10	79	7.9
2022-CS-62	Aqsa	Blood Donation Management System	Code formatation	100	10	70	7
2022-CS-62	Aqsa	Blood Donation Management System	Project Completion	100	40	77	30.8
2022-CS-62	Aqsa	Blood Donation Management System	Project Idea	100	10	69	6.9
2022-CS-63	Faisal	Blood Donation Management System	Code formatation	100	10	70	7
2022-CS-63	Faisal	Blood Donation Management System	Project Completion	100	40	77	30.8
2022-CS-63	Faisal	Blood Donation Management System	Project Idea	100	10	69	6.9
2022-CS-65	Wali	Blood Donation Management System	Code formatation	100	10	70	7
2022-CS-65	Wali	Blood Donation Management System	Project Completion	100	40	77	30.8
2022-CS-65	Wali	Blood Donation Management System	Project Idea	100	10	69	6.9
2022-CS-66	Bisma	Blood Donation Management System	Code formatation	100	10	70	7
2022-CS-66	Bisma	Blood Donation Management System	Project Completion	100	40	77	30.8
2022-CS-66	Bisma	Blood Donation Management System	Project Idea	100	10	69	6.9
2022-CS-67	Hamid	Election And Voting Management system.	Code formatation	100	10	60	6
2022-CS-67	Hamid	Election And Voting Management system.	Project Completion	100	40	87	34.8
2022-CS-67	Hamid	Election And Voting Management system.	Project Idea	100	10	89	8.9
2022-CS-69	Ruhab	Election And Voting Management system.	Code formatation	100	10	60	6
2022-CS-69	Ruhab	Election And Voting Management system.	Project Completion	100	40	87	34.8
2022-CS-69	Ruhab	Election And Voting Management system.	Project Idea	100	10	89	8.9
2022-CS-70	Laiba	Ecommerce Scapper	Code formatation	100	10	65	6.5
2022-CS-70	Laiba	Ecommerce Scapper	Project Completion	100	40	20	8
2022-CS-70	Laiba	Ecommerce Scapper	Project Idea	100	10	79	7.9
2022-CS-71	Fasi	Ecommerce Scapper	Code formatation	100	10	65	6.5

5.1.2 Report 3

List of students who left more than one group.

Students who changed group more than 1 times

Name	RegistrationNo	Email	Value	DateOfBirth	Contact
Aaliya	2022-CS-61	aaliyakhalil2022@gmail.com	Female	27/03/2004 12:00:00 am	3251620682
Leena	2022-CS-72	leenazaheer734@gmail.com	Female	11/08/2004 12:00:00 am	3344907146

5.2 Report 4

Advisors, along with a count of projects they are advising.

Advisors with number of project they are supervising

Advisor Name	Advisor Email	Number of Projects Advised
Awais	awaishasan@uet.edu.pk	2
Irzam	irzamliaquat@uet.edu.pk	1
Khaldoon	khaldoonkhurshid@uet.edu.pk	1
Laeeq	laeeqkhanniazi@uet.edu.pk	1
Nazeef ul Haq	nazeefulhaq@uet.edu.pk	2

5.3 Report 5

Groups that obtained the highest marks in each evaluation.

Group Which obtained highest marks in each evaluation

Name	TotalMarks	ObtainedMarks	GroupID
Project Idea	100	99	6
Project Completion	100	91	1
Code formatation	100	90	1

6 Conclusion

The Final Year Project Management System developed in this project is a comprehensive solution for efficiently managing final-year projects, including students, advisors, projects, evaluations, and group formations. The system offers a user-friendly interface implemented using WinForms in C# .NET, providing users an organized and interactive experience.

The project incorporates PDFsharp to generate PDF reports, offering valuable insights for the committee overseeing final-year projects. These reports cover project details, advisory board information, student marksheets, and other relevant data.

Overall, the project successfully addresses the complexities of managing final-year projects by using modern development tools and technologies. The user interface, generated reports, and implemented functionalities contribute to an efficient system for academic project management.