Final Year Project Management System

Session: 2021 – 2025

Submitted by:

Saqib Shehzad 2021-CS-187

Supervised by:

Sir Samayan

Department of Computer Science and Engineering

University of Engineering and Technology

Lahore Pakistan

Acknowledgments

I am grateful to Allah Almighty that He provided me with the strength and power to complete this project in time.

I am thankful to Mr. Nauman Babar for the supervision, guidance, and care he provided throughout the project for making out the best project I could, by enlightening me with the techniques to manage my work by making a schedule beforehand.

I would also like to thank Mr. Samyan Qayyum Wahla for his dedication towards Data Base Theory and presenting concepts in a manner that helped me complete this project with great ease.

Moreover, I am fortunate to have friends who helped me in the hardest times of the project.

Contents

List of Figures

Abstract

The prime objective of the final term project was the visualization and critical analysis regarding the use and working of databases in real-world problems. Databases help the data to be durable and secure enough that the probability of data loss becomes very less in case of any mishap. In the case of file systems, data can be stored in CSV or JSON format but the risk of data loss is high. Moreover, access to data can be provided to only a single user at a particular time, whereas, databases provide isolation to users helping multiple users to access the data at a single time. The scenario which was implemented to grasp the concepts of databases was the management of final year projects of the Computer Science Department of UET Lahore. The requirements of the management system were thoroughly explained in the case study where it was explained that current management of final year students is being done manually. Dealing this much large amount of data manually on papers is a tiresome job and loss of paper. Our task was to automate the whole process of managing final year students through a desktop application. CRUD operations were to be performed on the relations provided in the database schema. Moreover, PDF reports were to be generated that gave useful information to the advisory board in streamlining the process. ...

Chapter 1

Introduction

1.1 Description

The Computer Science Department at UET Lahore plans to automate their current manual final year project management system with a desktop application integrated with a database. The new system will eliminate manual interventions, reducing the risk of errors and ensuring accurate and efficient project management. The system will include all students in the session, with groups formed for specific projects, and will be supervised by the advisory board consisting of Main Advisor, Co-Advisor, and Industry Advisor. The system will keep records according to specific dates, and faculty members will be classified according to their designations to facilitate easy assignment of roles. The new system will streamline the final year project management process at UET Lahore.

1.2 Motivation

The main motivation for this project was to help to learn querying data inserted in a database. Learning how to extract specific information from different relations of the database. Simple queries along with complex queries were written to make this project a successful one. The project was implemented in C# .NET Framework and SQL Server was integrated as a database. Moreover, PDF reports were generated using iTextSharp containing useful information for the department to make this management system more effective.

1.3 Target Audience

The target audience for the project are the universities where final year projects are managed manually. This desktop application will help them to enter the student's

details for the whole department and then form groups through the application. Projects that are finalized by the department can be added to the system after which the projects are assigned their advisors. An advisor can supervise multiple projects and thus evaluate those groups. The students are evaluated in groups and thus marks are evenly distributed among them. Thus, the main audience includes the students, advisors, and the faculty that will be allocating groups to the students. Students can use this application to view their evaluations.

Chapter 2

Operational Details

Till now the management system consists of a single admin that will be acting as the only user.Following are the details that the admin can perform.

1. The admin will be having the right to enter all details of students in the system. Moreover, he would be able to add or update any student record

2. Project details will be entered by the admin and updation of records will be possible.

3. Advisors can be added and updated to the system.

4. Advisors can be added and updated to the system.

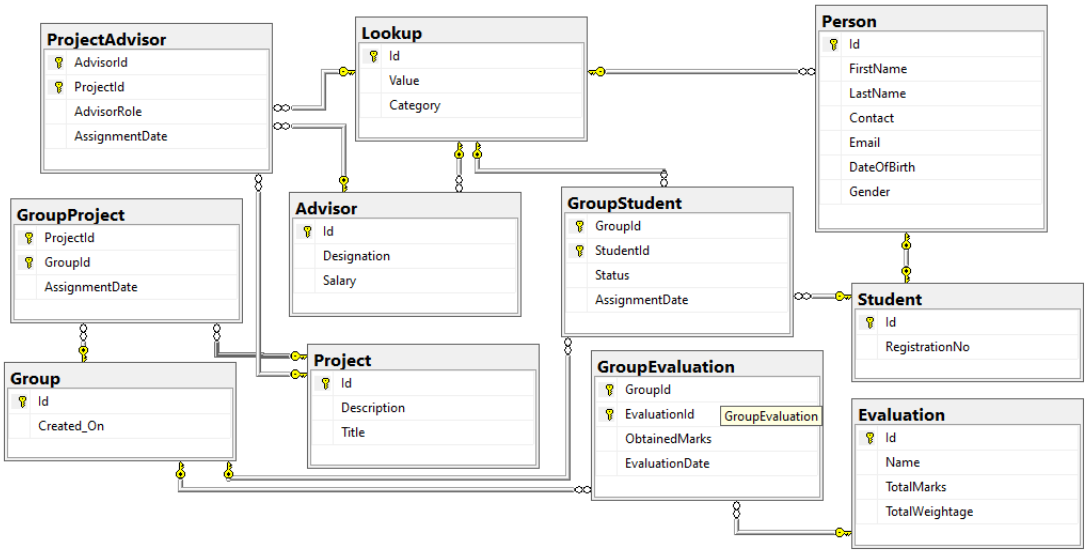
5. Advisors will be allocated to projects by the admin.

6. Groups will be assigned with their projects by the admin. Updation of any information will be possible.

7. Evaluation details will be added by the admin and the evaluated group's marks will be entered by the admin.

8. PDF reports can be generated by the admin

FIGURE 2.1: Database Diagram



Chapter 3

DataBase Design

3.1 Lookup

Lookup relation ?? is a supporting relation for other relations. It consists of advisor roles, designations and student's status.Lookup.ID is its primary key

3.2 Person

Peron consists of the data needed for a student. The Person.ID acts as a foreign key for the Student relation. If a student needs to be deleted then correspondingly the data in the Person relation should be deleted as the Student information is present in the Person relation.

3.3 Student

Student relation consists of the student registration and information of the person through Id which is a foreign key and a primary key. Person tuple corresponding to the Id of the Student relation needs to be deleted for complete deletion.

3.4 GroupStudent

The group student consists of the GroupId that is assigned to a StudentId. Both GroupId and StudentId acts as a primary key for the relation.

3.5 Group

Group consists of the Group.Id and the date of creation. The ID acts as a primary key. If the group is deleted then the relations that consist of the Group Id needs to be deleted. The group being deleted is then deleted from the GroupStudent relation, GroupEvaluation and GroupProject.

3.6 GroupProject

The GroupProject consists of the ProjectId assigned to a GroupId and thus both these attributes act as a primary key for the relation.

3.7 GroupEvaluation

GroupEvaluation has EvaluationId and GroupId as its primary key. It stores the marks obtained by a group in a specific evaluation.

3.8 Evaluation

Evaluation has Id as its primary attribute. It consists of the evaluation details that will be conducted throughout the final year project.

3.9 Project

Project has its Id as its primary key. It consists the title and description of the project.

3.10 ProjectAdvisor

Project advisor consists of the advisor that is assigned to a project along with the role that is assigned to the advisor. The ProjectId and the AdvisorId acts as the primary key for the relation. Advisor role acts as a foreign key for the parent Lookup table.

3.11 Advisor

The Advisor relation consists of the advisor's designation and the salary. Id acts as the primary key for the relation. The designation is a foreign key for the parent Lookup table. Deleting an advisor from the system will cause the associated details from the ProjectAdvisor to be deleted.

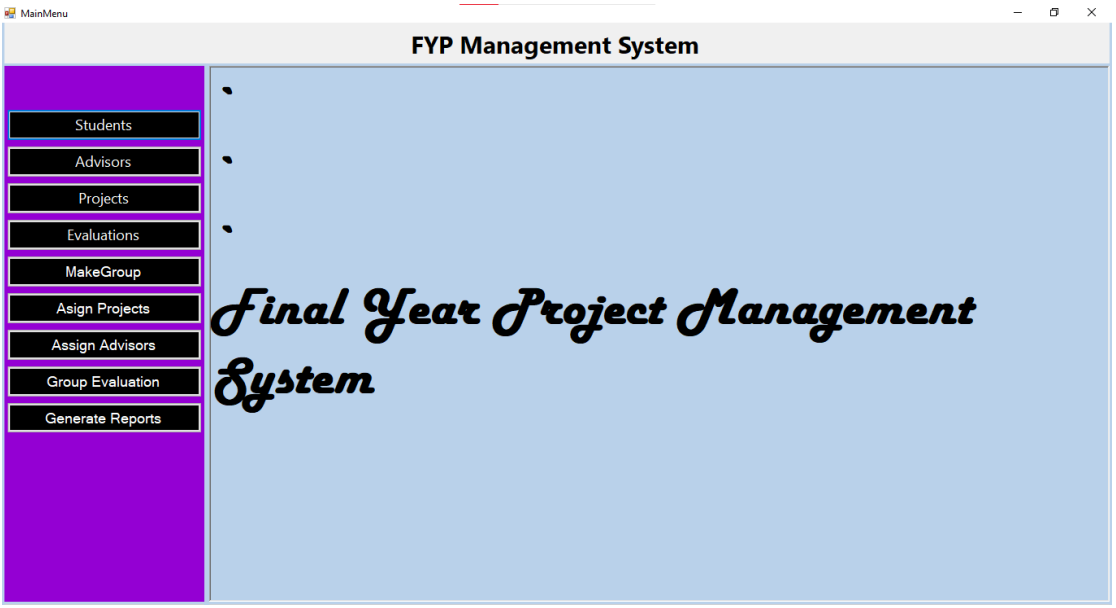
Chapter 4

GUI

4.1 Main Menu Page

Main Menu page Displays all the Available options that a user can perform.

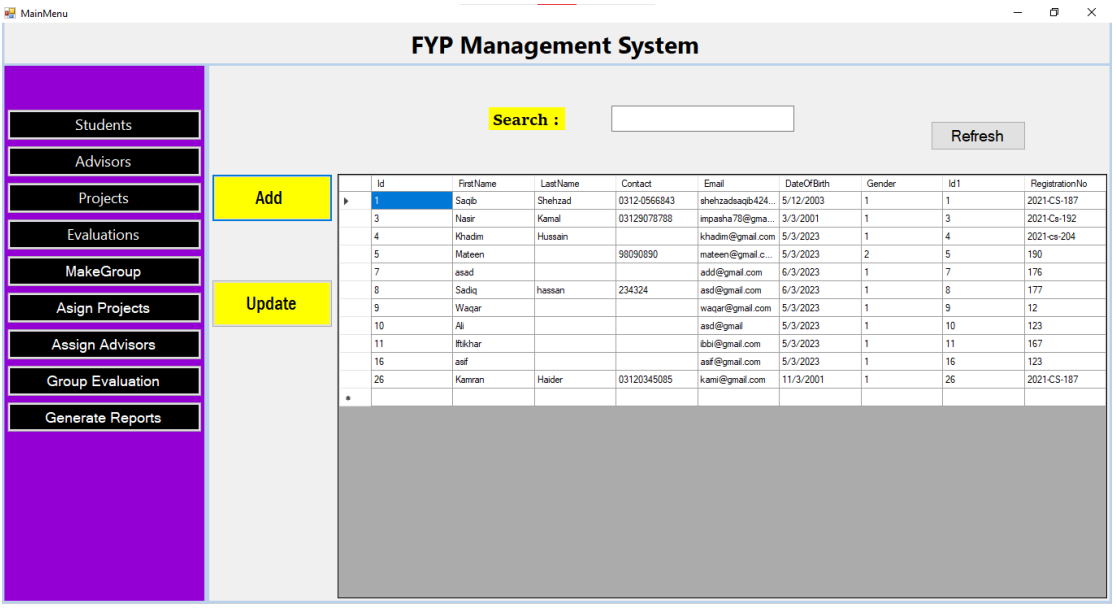
FIGURE 4.1: Main Menu



4.2 Manage Student Page

On this page All the options related to Student are available.Add Student,Update and Search the Student from the Existing Recods.All Students are viewed from this page

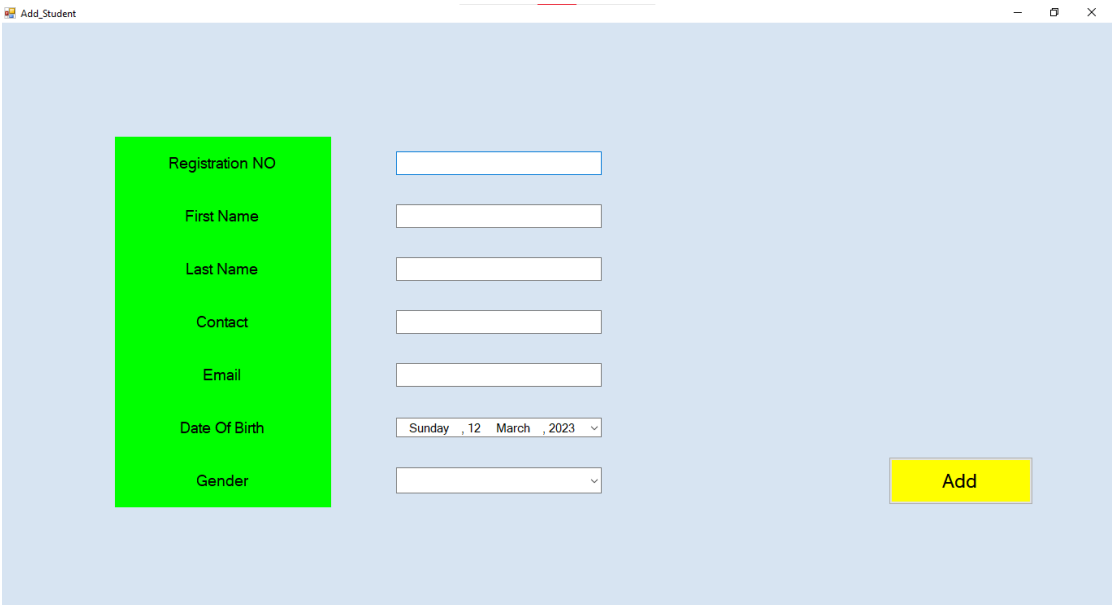
FIGURE 4.2: Manage Students



4.3 Add Student Page

The add student page asks from the user to enter the details of students through text boxes and combo boxes. After adding the information the user clicks on the Add button to insert the data in the system.

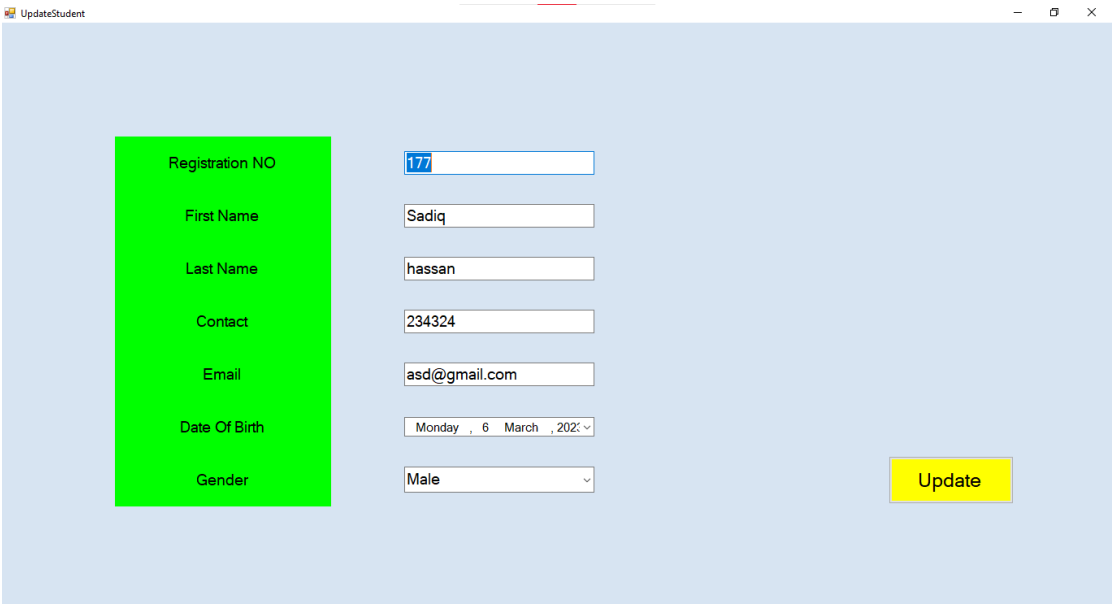
FIGURE 4.3: Add Students



4.4 Update Student

The update students page can be used to update some information of the student by using his registration number.

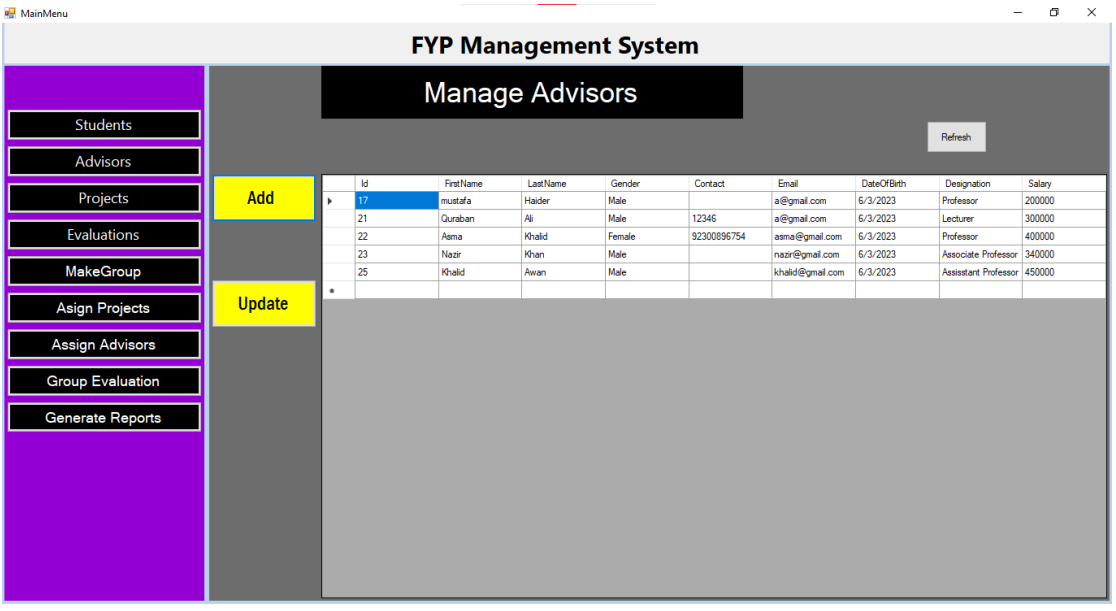
FIGURE 4.4: Update Students



4.5 Manage Advisors Page

On this page All the options related to Adviosrs are available.Add Advisors and Update Existing Recods.All Advisors are viewed from this page.

FIGURE 4.5: Manage Advisor



4.6 Add Advisor

The add advisor page asks from the user to enter the details of advisors through text boxes and combo boxes. After adding the information the user clicks on the Add button to insert the data in the system.

FIGURE 4.6: Add Advisor

4.7 Update Advisor

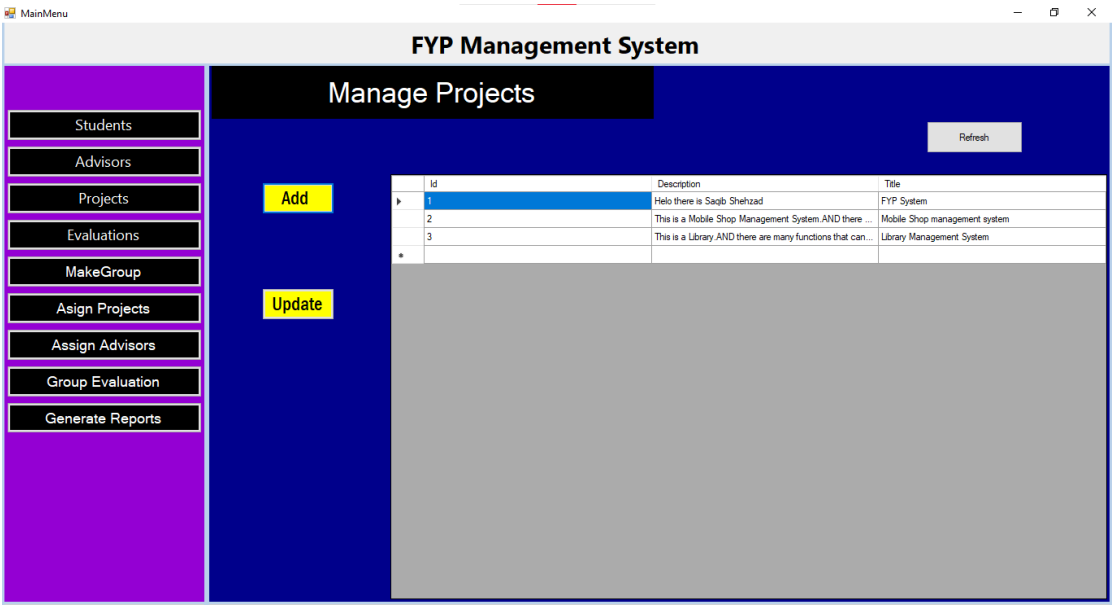
The update adviosrs page can be used to update some information of the advisors.

FIGURE 4.7: Update Advisor

4.8 Manage Projects Page

From this page all the projects can be viewwd and also the others buttons are available to add and update.

FIGURE 4.8: Manage Projects



4.9 Add Projects

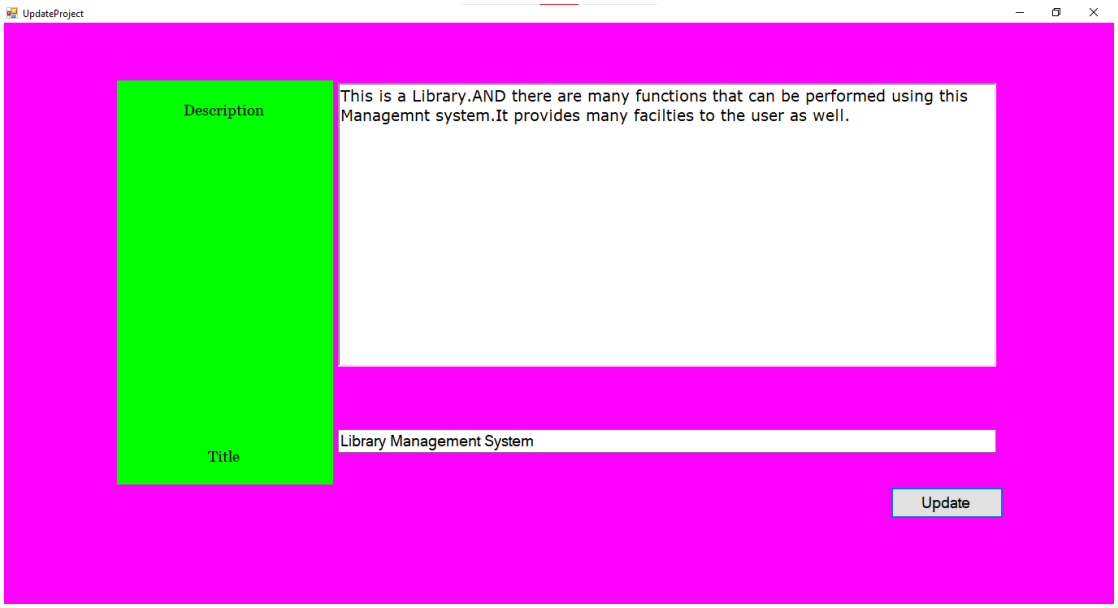
The add Projects page asks from the user to enter the details of projecs through text boxes.

FIGURE 4.9: Add Projects

4.10 Update Projects

The update Projects page can be used to update some information of the projects.

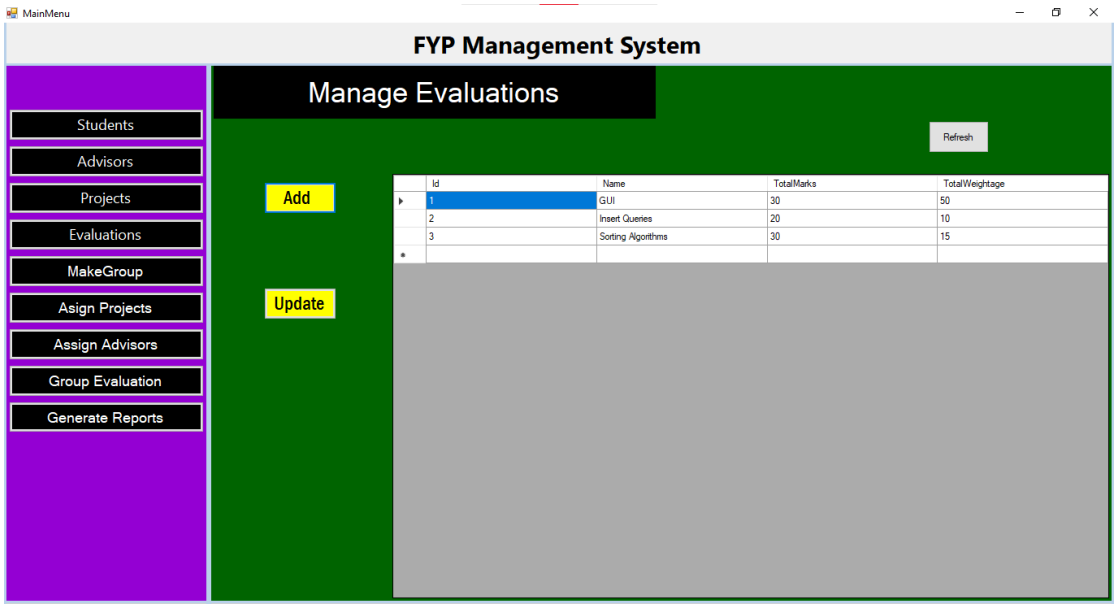
FIGURE 4.10: Update Projects



4.11 Manage Evaluations

From this page all the Evalutions can be viewwd and also the others buttons are available to add and update.

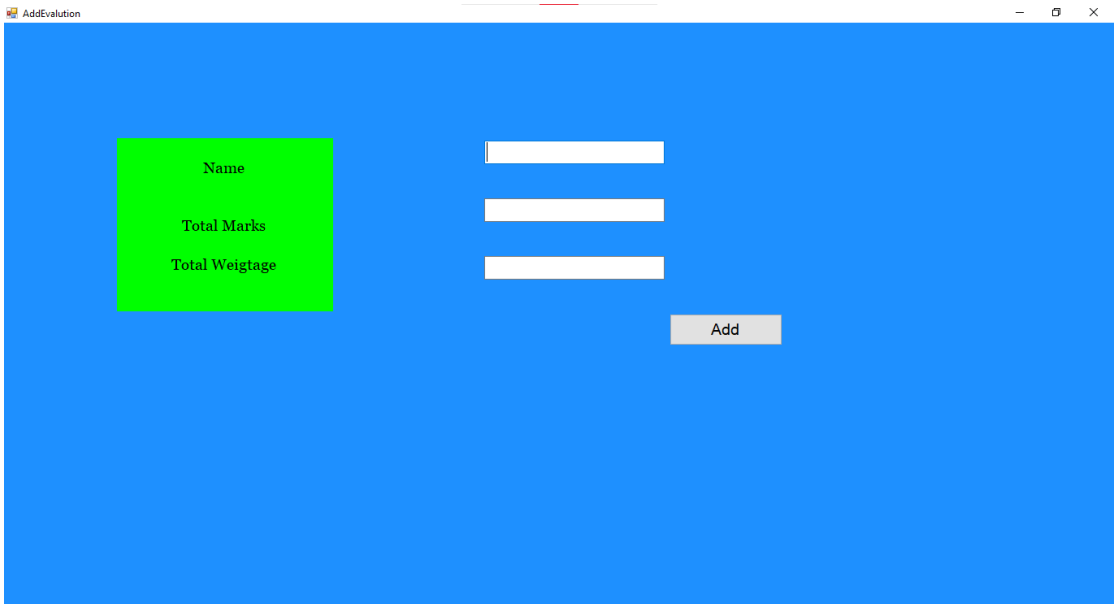
FIGURE 4.11: Manage Evaluations



4.12 Add Evaluations

The add Evaluations page asks from the user to enter the details of evaluations through text boxes.

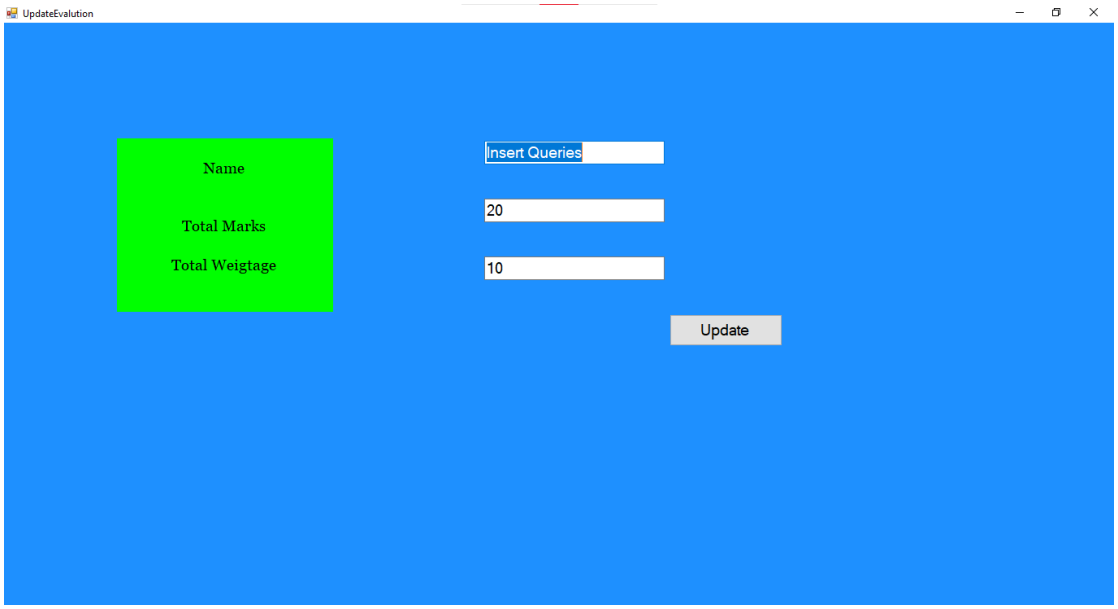
FIGURE 4.12: add Evaluations



4.13 Update Evaluations

The update Evaluations page can be used to update some information of the evaluations.

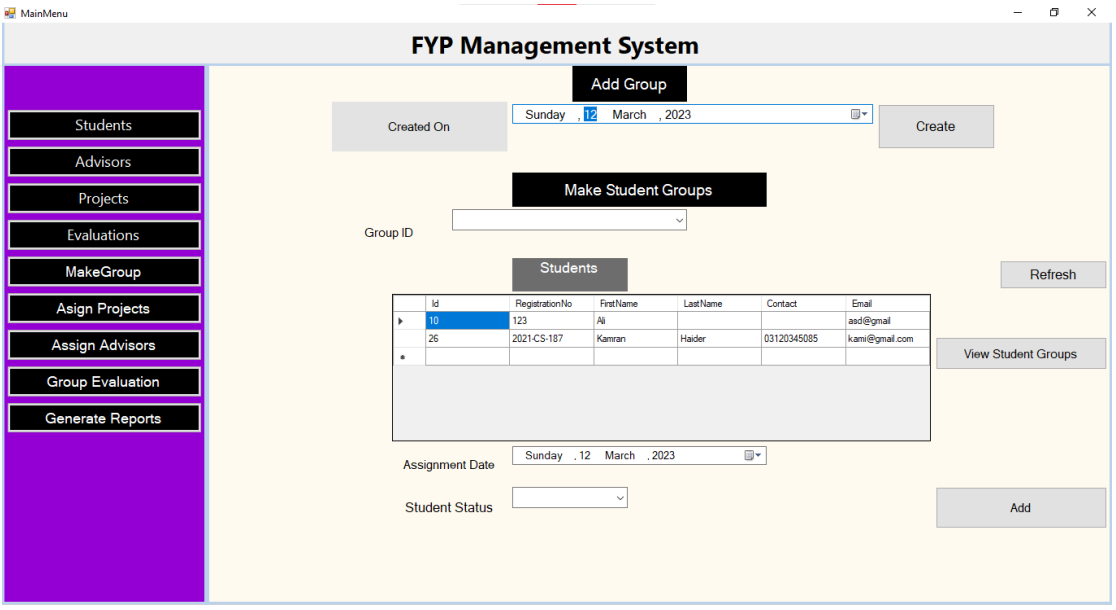
FIGURE 4.13: Update Evaluations



4.14 Create and Assign Group to Students

In this page first the group is created and then the groups are assigned to students with their ID and Status.

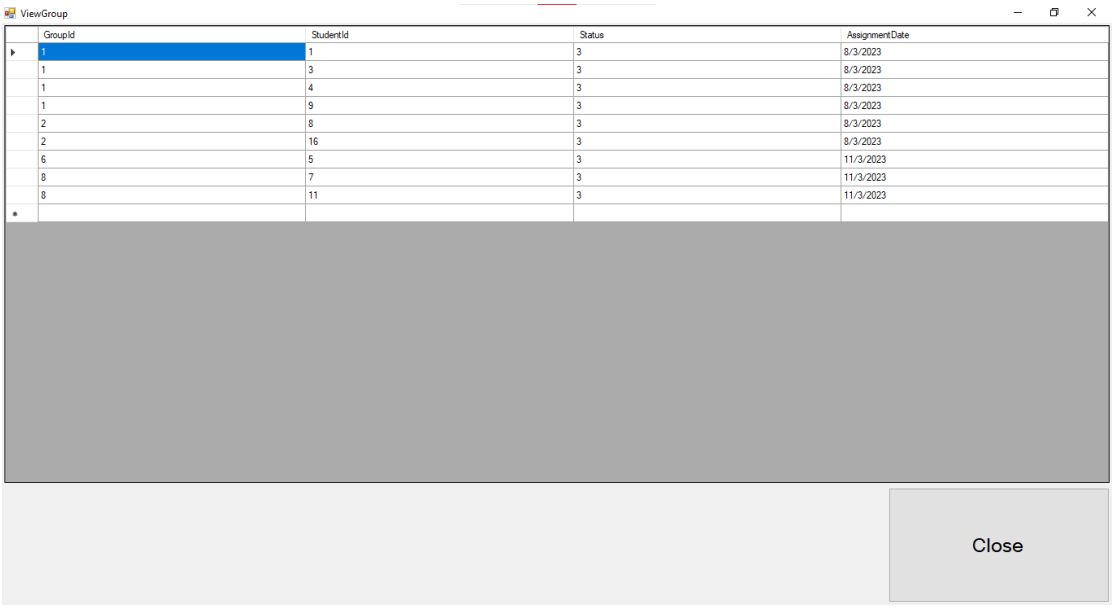
FIGURE 4.14: Create and Assign Groups



4.15 View Student Groups

In this page All the groups are shown with the student ids.

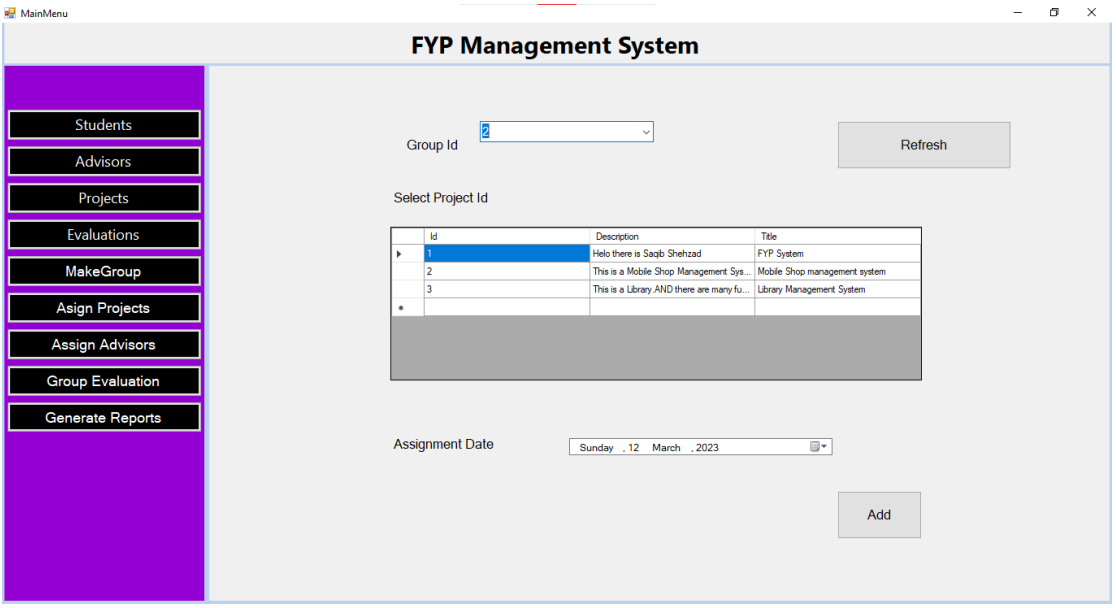
FIGURE 4.15: View Groups



4.16 Assign Projects Page

In this page Projects are assigned to the groups with the assignment date.

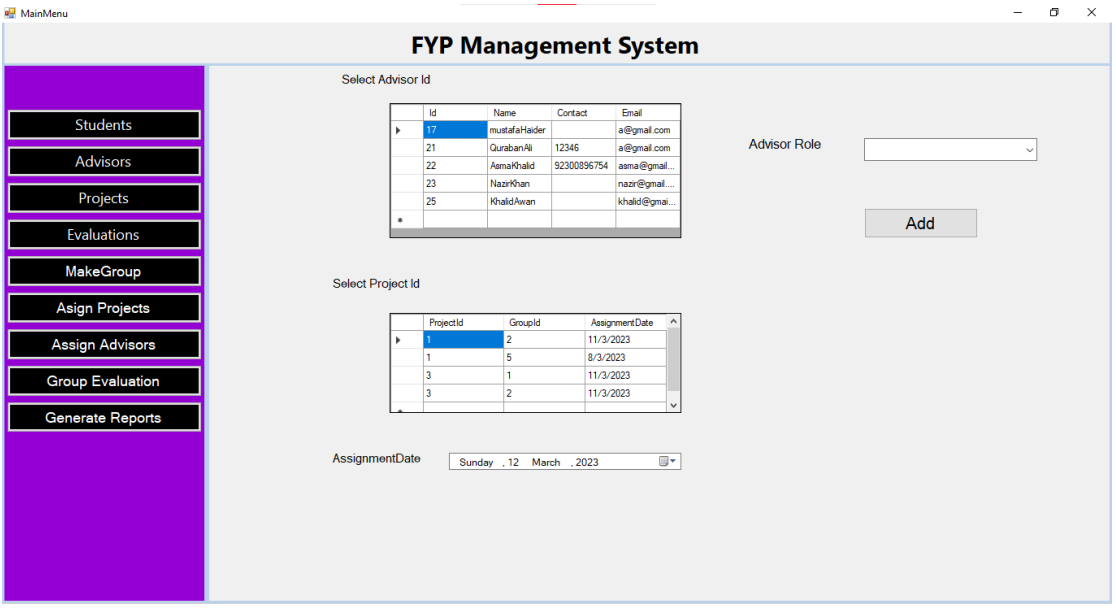
FIGURE 4.16: Assign Projects



4.17 Assign Advisors page

In this page Advisors are assigned to the projects with the adviosr ids and project ids and the advisor roles.

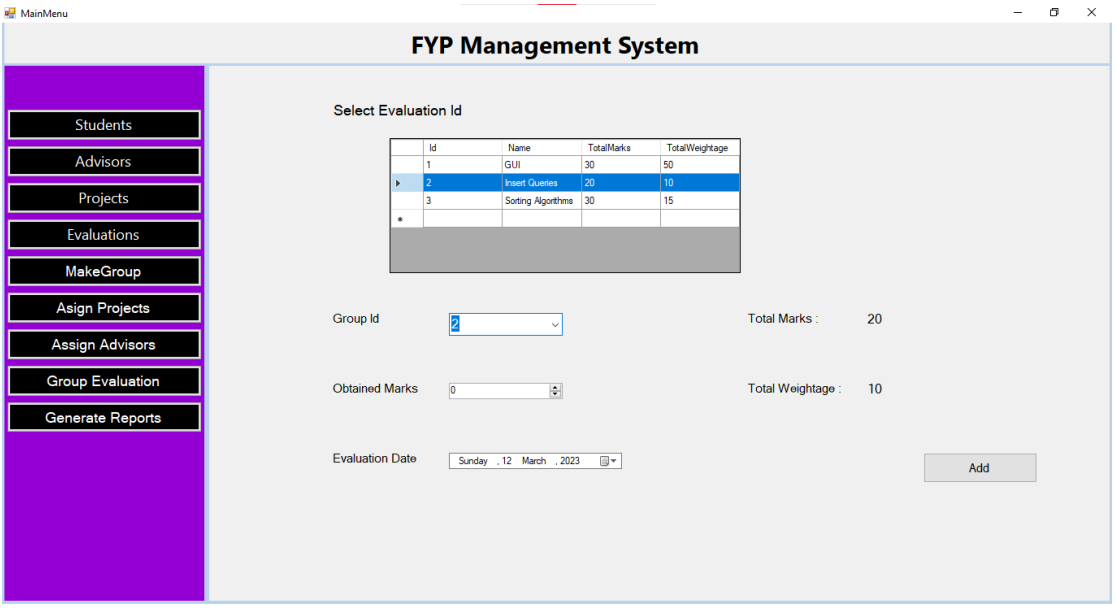
FIGURE 4.17: Assign Advisor



4.18 Group Evaluation

In this page Evaluations of the groups are marked with their obtained marks.

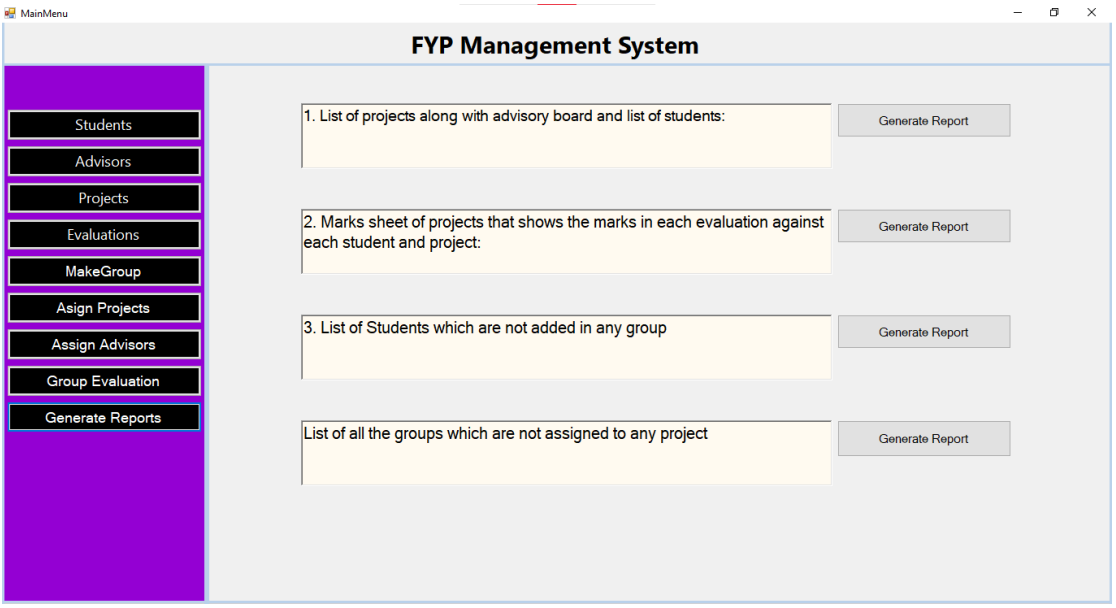
FIGURE 4.18: Group Evaluations



4.19 Generate PDF Reports

There are different buttons that create different PDFs with Different Information.

FIGURE 4.19: Generate Reports



Chapter 5

Queries

5.1 Select Queries

5.1.1 Select the Ids of the Students that are not assigned to any Groups.

```
SELECT S.Id, S.RegistrationNo, FirstName, LastName, Contact, Email
FROM Student AS S
JOIN Person AS P ON S.Id = P.Id
EXCEPT
SELECT S.Id, S.RegistrationNo, P.FirstName, P.LastName, P.Contact, P.Email
FROM [dbo].[Group] AS G
JOIN [dbo].GroupStudent AS Gs ON Gs.GroupId = G.Id
JOIN Lookup AS L ON L.Id = Gs.Status
JOIN Person AS P ON P.Id = Gs.StudentId
JOIN Student AS S ON S.Id = P.Id
WHERE L.Value = 'Active'
```

5.1.2 Select Only those Groups in which Students are assigned

```
SELECT DISTINCT Id
FROM dbo.[Group] g
JOIN GroupStudent gs ON g.Id = gs.Groupid
```

5.1.3 Select all the Projects in the system

```
SELECT * FROM Project;
```

5.1.4 Select Selected From the Lookup table using value

```
SELECT id FROM lookup WHERE value = @value;
```

5.1.5 List of projects along with advisory board and list of students:

```
SELECT P1.Id, P1.FirstName, P1.LastName, L.Value AS Gender, P1.Contact, P1.Email, P1.DateOfBirth
,P1.Designation, P1.Salary
FROM Lookup AS L
JOIN (
SELECT A.Id, A.Salary, P.FirstName, P.LastName, P.Contact, P.Email, P.DateOfBirth,
L.Value AS Designation, P.Gender
FROM Advisor AS A
JOIN Person AS P ON P.Id = A.Id
JOIN Lookup AS L ON L.Id = A.Designation
) AS P1 ON P1.Gender = L.Id;
```

Listing 5.1: Advisor Information

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

```
SELECT
p.id AS project_id,
p.title AS project_title,
s.id AS student_id,
CONCAT(per.firstname, ' ', per.lastname) AS student_name,
e.id AS evaluation_id,
e.name AS evaluation_name,
ge.obtainedmarks AS marks_obtained,
SUM(ge.obtainedmarks) OVER(PARTITION BY p.id, s.id) AS total_marks
FROM
project AS p
INNER JOIN groupProject AS gp ON p.id = gp.ProjectId
INNER JOIN groupStudent AS gs ON gp.groupId = gs.groupId
INNER JOIN student AS s ON gs.studentId = s.id
INNER JOIN person AS per ON s.id = per.id
INNER JOIN GroupEvaluation AS ge ON gp.groupId = ge.groupid
INNER JOIN evaluation AS e ON ge.evaluationID = e.id
ORDER BY
p.id, s.id, e.id;
```

Listing 5.2: Project Evaluation Information

5.2 Insert Queries

5.2.1 Insert Values in Group Student

```
INSERT INTO dbo.[GroupStudent] VALUES (@GroupId, @StudentId, @Status, @AssignmentDate);
```

Listing 5.3: Inserting Data into GroupStudent Table

5.2.2 Insert values into person

```
INSERT INTO Person VALUES (@FirstName, @LastName, @Contact, @Email, @DateOfBirth,
@Gender);
```

Listing 5.4: Inserting Data into Person Table

5.2.3 Insert Values into Advisors

```
INSERT INTO Advisor VALUES (@Id, @Designation, @Salary);
```

Listing 5.5: Inserting Data into Advisor Table

5.2.4 Insert into projects

```
INSERT INTO Project VALUES (@Description, @Title);
```

Listing 5.6: Inserting Data into Project Table

5.3 Update Queries

5.3.1 Update Person where selected id is equal to selected ID

```
UPDATE [dbo].[Person] SET FirstName = @FirstName, LastName = @LastName, Contact =
@Contact, Email = @Email, DateOfBirth = @DOB, Gender = @Gender WHERE Id = @Id;

UPDATE [dbo].[Advisor] SET Designation = @Designation, Salary = @Salary WHERE Id =
@Id;
```

Listing 5.7: Updating Data in Person and Advisor Tables

5.3.2 Update Evaluation

```
UPDATE Evaluation SET Name = @Name, TotalMarks = @TotalMarks, TotalWeightage =
@TotalWeightage WHERE ID = @Id;
```

Listing 5.8: Updating Data in Evaluation Table

5.3.3 Update Project

```
UPDATE Project SET Description = @Description, Title = @Title WHERE Id = @Id;
```

Listing 5.9: Updating Data in Project Table

Chapter 6

Generated Reports

6.1 Report 1

6.1.1 List of projects along with advisory board and list of students:

```
SELECT P1.Id, P1.FirstName, P1.LastName, L.Value AS Gender, P1.Contact, P1.Email, P1.DateOfBirth
,P1.Designation, P1.Salary
FROM Lookup AS L
JOIN (
SELECT A.Id, A.Salary, P.FirstName, P.LastName, P.Contact, P.Email, P.DateOfBirth,
L.Value AS Designation, P.Gender
FROM Advisor AS A
JOIN Person AS P ON P.Id = A.Id
JOIN Lookup AS L ON L.Id = A.Designation
) AS P1 ON P1.Gender = L.Id;
```

Listing 6.1: Advisor Information

6.2 Report 2

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

```
SELECT
p.id AS project_id,
p.title AS project_title,
s.id AS student_id,
CONCAT(per.firstname, ' ', per.lastname) AS student_name,
e.id AS evaluation_id,
e.name AS evaluation_name,
ge.obtainedmarks AS marks_obtained,
SUM(ge.obtainedmarks) OVER(PARTITION BY p.id, s.id) AS total_marks
FROM
project AS p
INNER JOIN groupProject AS gp ON p.id = gp.ProjectId
```

```
INNER JOIN groupStudent AS gs ON gp.groupId = gs.groupId
INNER JOIN student AS s ON gs.studentId = s.id
INNER JOIN person AS per ON s.id = per.id
INNER JOIN GroupEvaluation AS ge ON gp.groupId = ge.groupid
INNER JOIN evaluation AS e ON ge.evaluationID = e.id
ORDER BY
p.id, s.id, e.id;
```

Listing 6.2: Project Evaluation Information

6.3 Report 3

6.3.1 List of Students which are not added in any group:

```
SELECT S.Id, S.RegistrationNo, FirstName, LastName, Contact, Email
FROM Student AS S
JOIN Person AS P ON S.Id = P.Id
EXCEPT
SELECT S.Id, S.RegistrationNo, P.FirstName, P.LastName, P.Contact, P.Email
FROM [dbo].[Group] AS G
JOIN [dbo].GroupStudent AS Gs ON Gs.GroupId = G.Id
JOIN Lookup AS L ON L.Id = Gs.Status
JOIN Person AS P ON P.Id = Gs.StudentId
JOIN Student AS S ON S.Id = P.Id
WHERE L.Value = 'Active'
```

6.4 Report 4

6.4.1 List of all the groups which are not assigned to any project

```
SELECT GroupId, StudentId FROM GroupStudent
EXCEPT
SELECT GroupId, ProjectId FROM GroupProject;
```

Listing 6.3: Selecting Data from GroupStudent Except GroupProject

Chapter 7

Limitations

Following are the limitations of the project:

1: Limited User Access: The system is designed to only allow one admin user to access and manage the data. This can be a limitation if there are multiple stakeholders who need to access or contribute to the project.

2: Lack of Collaboration: While the system allows the admin to create groups of students, it does not provide any collaboration features such as real-time messaging or document sharing. This can be a limitation if the student groups need to collaborate closely on their projects.

3: Limited Functionality: While the system covers the basic functions of project management and evaluation, it may not be suitable for more complex scenarios or special requirements. For instance, it may not support advanced analytics or integrations with other systems.

4: Security Risks: While the system has some security measures such as user authentication and authorization, it may still be vulnerable to security risks such as data breaches or attacks. The system should undergo regular security assessments and updates to mitigate these risks.

Chapter 8

Future Work

The future work that can be done on the project are enlisted below:

1: Multiple User Roles: Instead of having a single admin user, the system could be designed to support multiple user roles such as project managers, advisors, and students. This would allow for more collaboration and delegation of responsibilities.

2: Enhanced Collaboration: The system could be upgraded to include features such as real-time messaging, document sharing, and task assignments to improve collaboration among student groups and advisors.

3: Integration with External Systems: The system could be upgraded to integrate with other systems such as learning management systems, project management tools, and collaboration platforms to improve workflow and productivity.

4: Mobile App Support: The system could be upgraded to include a mobile app that would allow students, advisors, and admin to access the system from their smartphones or tablets.

5: Improved Security: The system could be upgraded to include more advanced security features such as data encryption, intrusion detection, and continuous security monitoring to protect against cyber threats.

Chapter 9

Conclusion

In conclusion, the Final Year Management System is a promising project that addresses the needs of managing student projects and evaluations in a centralized and efficient manner. The system provides essential features that simplify the process of creating, assigning, and evaluating student projects while ensuring data security.

Overall, this project is a powerful tool that has the potential to become even more effective with future upgrades and enhancements. By continuously evaluating the needs of the users and incorporating cutting-edge technologies, the system can continue to adapt and evolve to meet the ever-changing demands of academia.