

College Management System

by

Deva Anand M & CS23B1020

Independent Project

[Computer science department]

ACKNOWLEDGEMENTS

I am very thankful to the Indian Institute of Information Technology, Raichur for allowing me to do this Independent Project. I am grateful to my guide Mr. Natesh BV support throughout the project. Their help improved my skills and knowledge a lot. The institute provided a good environment for research. This project was an important achievement for me.



**Indian Institute of Information Technology
Raichur (IIITR)**

Raichur, Karnataka – 584135

CONTENTS

ABSTRACT

1. INTRODUCTION - 5
2. AIM, MOTIVATION, and OBJECTIVES - 6
3. METHODOLOGY - 9
4. RESULTS - 20
5. DISCUSSION - 39
6. CONCLUSION - 40

REFERENCES

ABSTRACT

The goal of this project is to create a comprehensive software system to efficiently manage all major operations and record keeping for a college or university. The key modules of the College Management System will include Student Management, Faculty Management, Course Management and Fee Management.

The Student Management module will maintain records of all currently enrolled students, including personal details, fee payment status and other relevant data. This will allow administrators to easily access and update student information. The Faculty Management component will store faculty member data like personal details, educational qualifications, salary information, courses taught and other responsibilities. It will facilitate processes like hiring new faculty and assigning teaching loads.

The Fee Management module will integrate with Student Management to track due dates and payments of all types of fees such as tuition fees, Salary etc.

The unified College Management System will improve efficiency, organization and data accessibility for administrators. Overall, this system aims to modernize college operations through the power of integrated software systems and databases.

1. INTRODUCTION:

Efficiently managing all aspects of college operations is a significant challenge for institutions of higher education. The existing system is not user friendly because the retrieval of data is very slow, and data is stored manually. In this existing system the papers can miss placed and documents can be loss. This will cause extra work for the admin department stuffs. The project aims to address these issues by developing a comprehensive College Management System (CMS).

Key terms and concepts such as Relational Data base management system (RDBMS), CRUD, .NET Framework, C Sharp, SQL Query, Microsoft SQL Server, student information management, Faculty information management, and user interface design principles.

The CMS will greatly simplify and expedite management processes by eliminating paperwork. Administrators, faculty, and students can perform tasks effortlessly through a convenient digital interface. Centralized information and easy accessibility provide efficient control over operations. The system enhances productivity by reducing administrative workloads, allowing staff to focus on core duties.

The administrator and students are two major functional requirements in the system. The Administrator will be given more powers (enable/disable/ update) than other users. It will be ensured that the information entered is of the correct format. For example, name cannot contain numbers. In case if incorrect form of information is added, the user will be asked to fill the

information again. Students use the system to query and enter their information only.

The goal is to create a solution that can streamline operations, reduce workload on staff, and provide efficient services for students, faculty and college administrators.

2. AIM, MOTIVATION, and OBJECTIVES

AIM:

The key aim is to create a user-friendly, centralized software system that digitizes and efficiently manages core operations like student admissions, course management, Faculty Management, fee payments and Salary Management records. This system aims to improve the efficiency of college information management, and the main function is managing and maintaining information. The system aims to reduce manual workloads, minimize errors, improve information access and communications while ensuring data security.

MOTIVATION:

- To save costs and improve productivity by optimizing and integrating operations under one system.
- To provide better information access, communication, and services for students and faculty.

- To modernize the institution by adopting new technologies that meet current digital needs.
- To strengthen data security, reporting capabilities, and regulatory compliance.

OBJECTIVES:

- **Centralized data management:** To create a centralized database to store and manage all institutional data related to students and Faculty. This will ensure data consistency, integrity, and easy accessibility.
- **Scalability and flexibility:** To develop a system that can scale and adapt to the growing needs of the institution, with the ability to integrate new modules or functionalities as required.
- **Compliance and Reporting:** Facilitate compliance with regulatory requirements and accreditation standards by generating accurate reports, maintaining audit trails, and ensuring data security and privacy measures are in place to protect sensitive information.

3. METHODOLOGY:

3.1 SCOPE FOR DEVELOPMENT OF THIS PROJECT

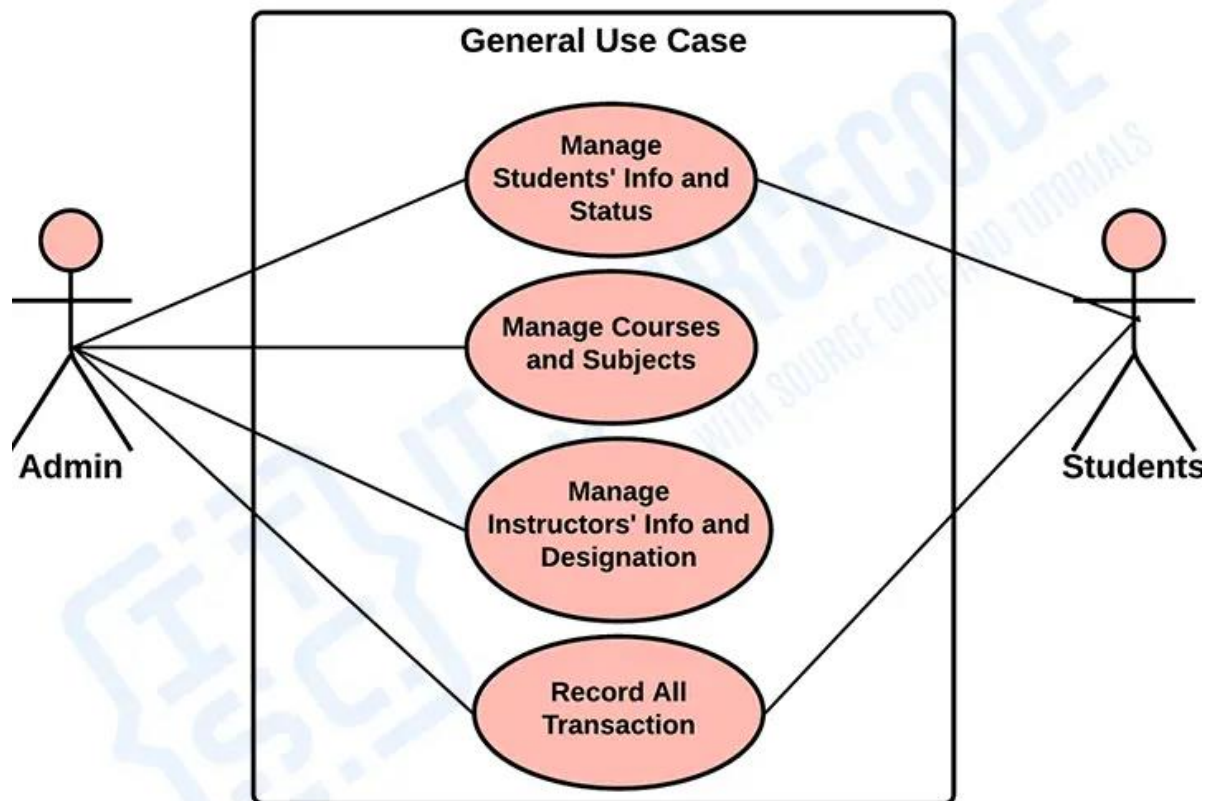
The requirement of the admin is to:

- Login to the system through the first page of the application.

- Access/ Search information from search student and search teacher portal.
- Create a new student information from New Admission portal.
- Fee can be updated through fee portal.
- Remove the student information from remove student portal.
- Upgrade the student's semester from upgrade semester portal.
- Create a new faculty information from add teacher portal.
- Salary can be credited for the faculty from salary portal.

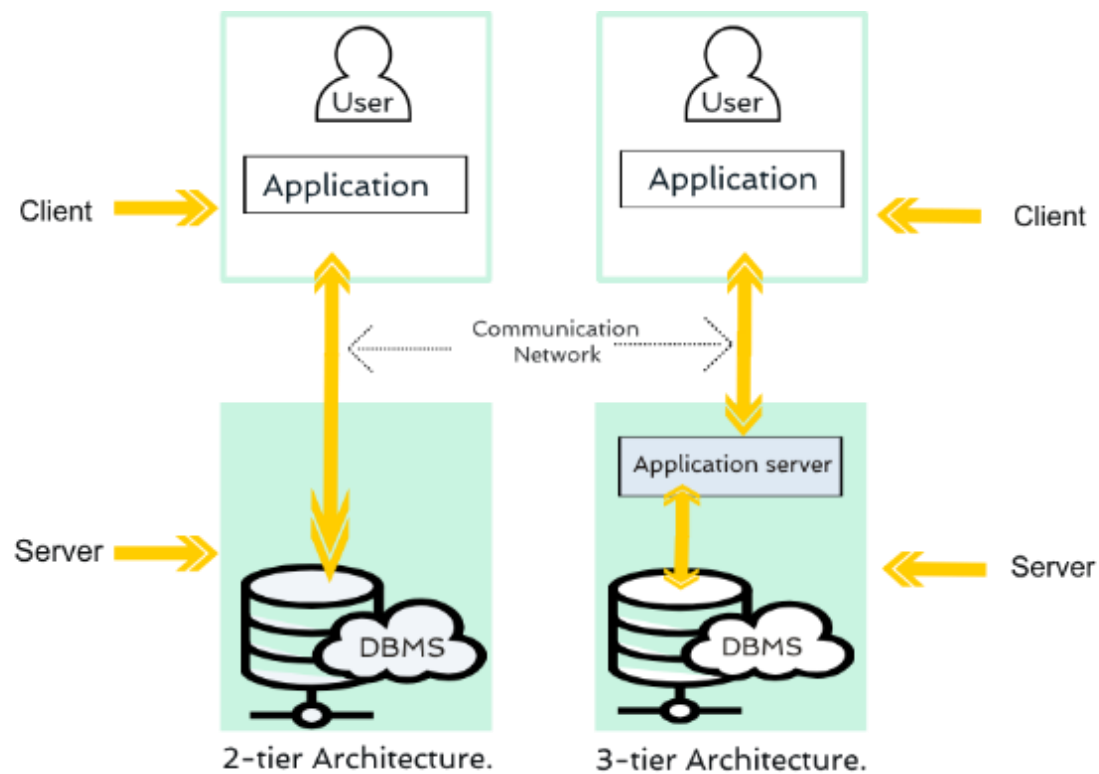
3.2 System Design:

COLLEGE MANAGEMENT SYSTEM



USE CASE DIAGRAM

3.2.1 ARCHITECTURE



2-tier vs 3 tier

- The college management system employs a **client-server** architecture following a **2-tier** model, consisting of the client application tier and the database server tier. This approach is suitable as the system's users (students, faculty, administrators) are considered **naive user**, meaning they lack expertise in directly interacting with databases using SQL.
- The client-server 2-tier model allows these naive users to access and manipulate data stored in the database server through an **intuitive software application** interface on the client side. The client application tier handles all the database interaction logic, translating user inputs into corresponding SQL queries and commands to be executed on the database server. It shields users from the complexities of SQL, providing a user-friendly experience focused on the relevant institutional processes and information management tasks. This architectural approach balances ease of use for naive users

while ensuring data integrity through controlled database access to the server.

3.2.2 Database and DBMS:

- Database management **CRUD** functionality encompasses Create, Read, Update, and Delete operations, fundamental for interacting with databases. **"Create"** involves adding new data entities, **"Read"** retrieves existing data, **"Update"** modifies data, and **"Delete"** removes data. Users insert new records, query existing ones, update attributes, and delete records as needed.

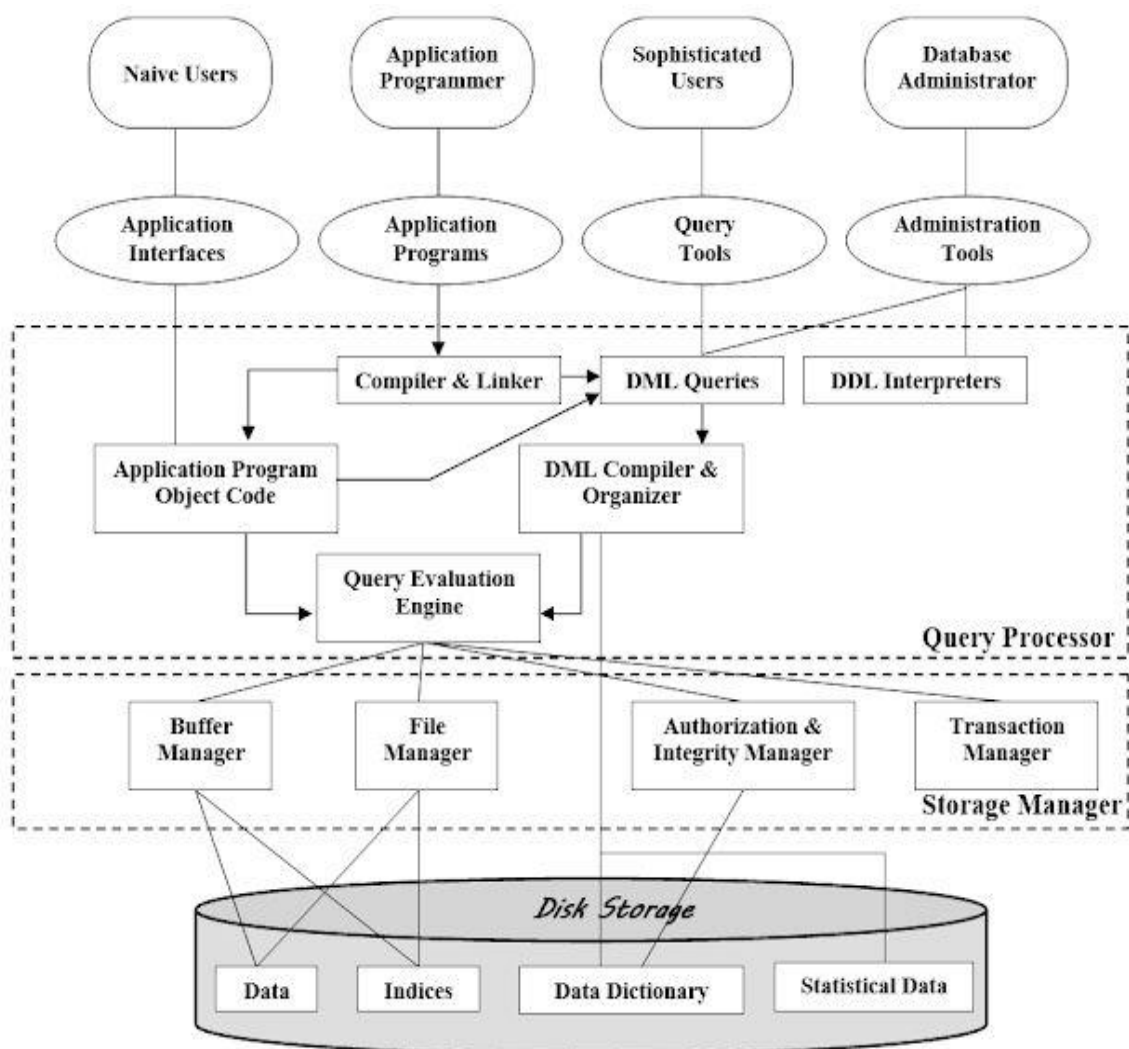


Figure: System Architecture

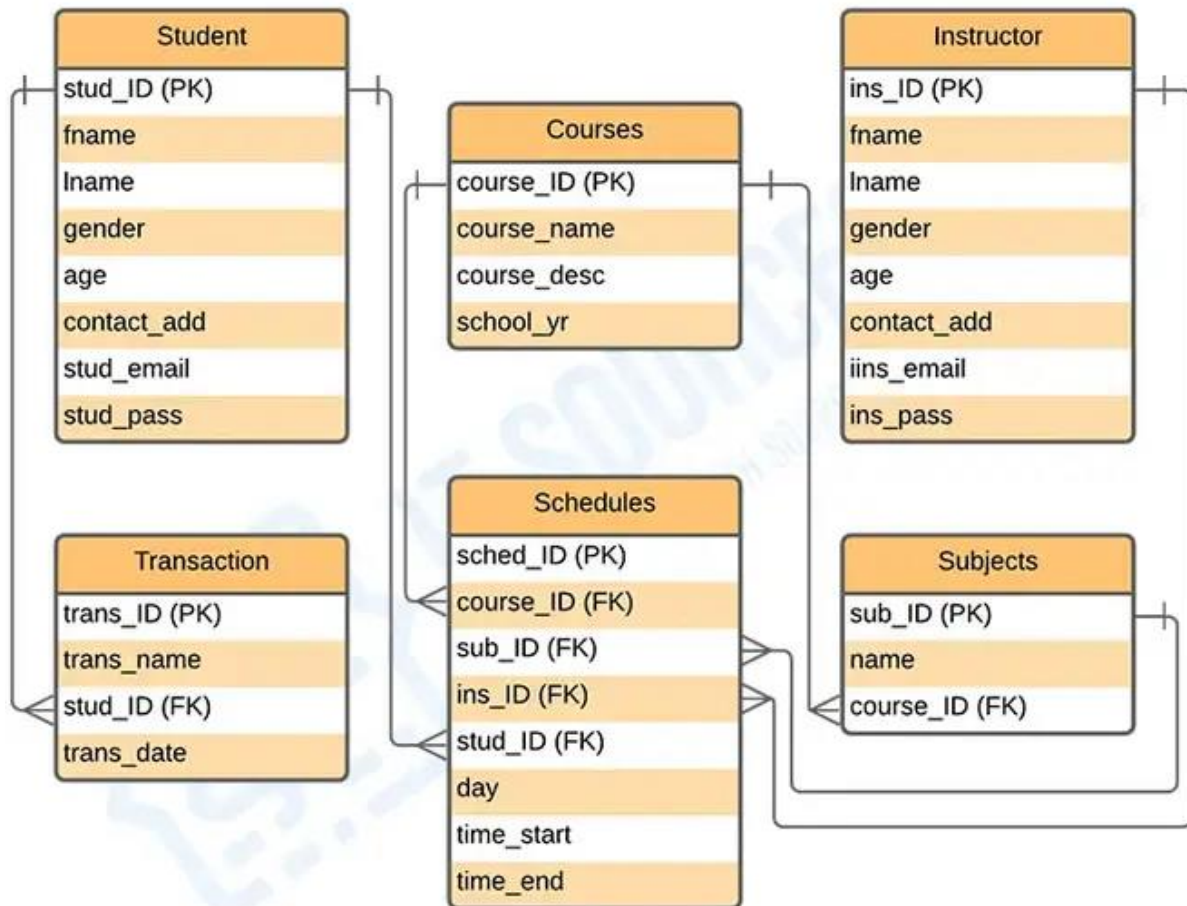
- In the college management system's database architecture, the Database Management System (DBMS) acts as an **intermediary** between the application and the physical storage of data.
- The DBMS consists of a query processor component that interprets and executes SQL queries received from the application. When a query is submitted, the query processor analyses its syntax, optimizes the execution plan, and retrieves or modifies data accordingly from the storage component.

- The storage component manages the physical storage and organization of data on disk, utilizing techniques like indexing and file management for efficient access.
- The DBMS ensures data integrity, security, and concurrent access control, shielding applications from low-level storage details. This architecture allows the application to interact with the database through high-level SQL statements processed by the DBMS's query processor and storage components.

3.2.3 Entity-Relationship (ER) diagram:

An ER diagram is a graphical representation that models the data requirements and relationships for a database system. It provides a high-level, conceptual view of the key data entities, their attributes, and the relationships between those entities

COLLEGE MANAGEMENT SYSTEM



ENTITY RELATIONSHIP DIAGRAM

This conceptual ER design allows us to analyse data objects and their relationships from a high level, without worrying about the physical implementation details initially. It helps identify potential issues like redundancy before translating the design into a logical database schema.

3.3 Technology Stack:

Frontend Development:

Platform: .NET Framework

Programming Language: C#

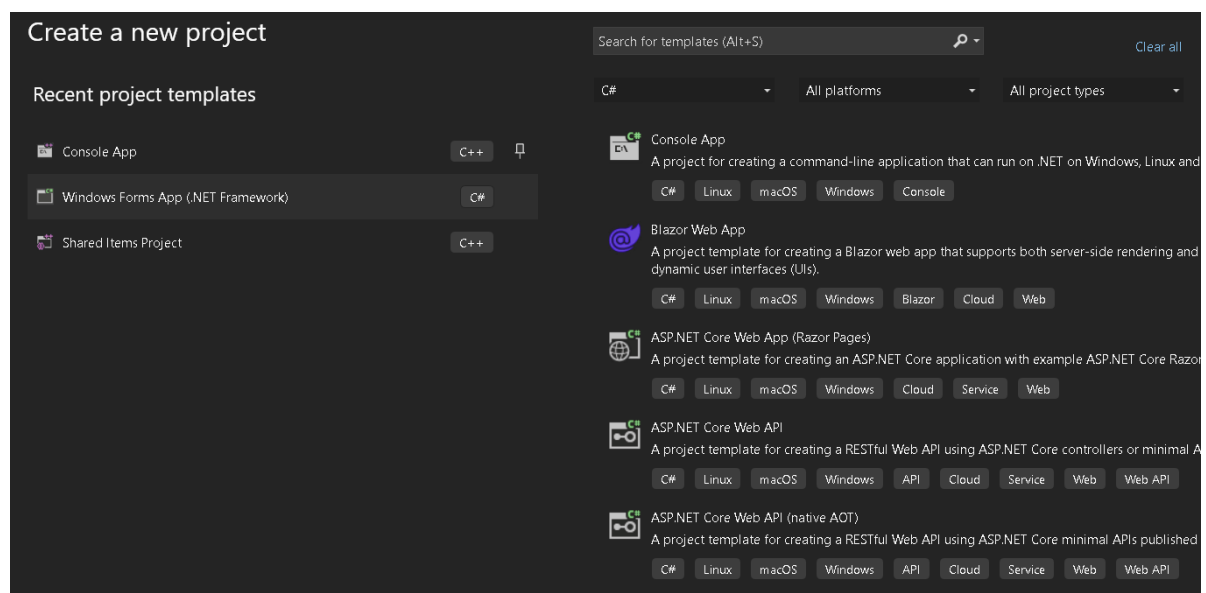
Development Environment: Visual Studio IDE

Backend Database:

Relational Database Management System (RDBMS): Microsoft SQL Server

Language: Structured Query Language (SQL)

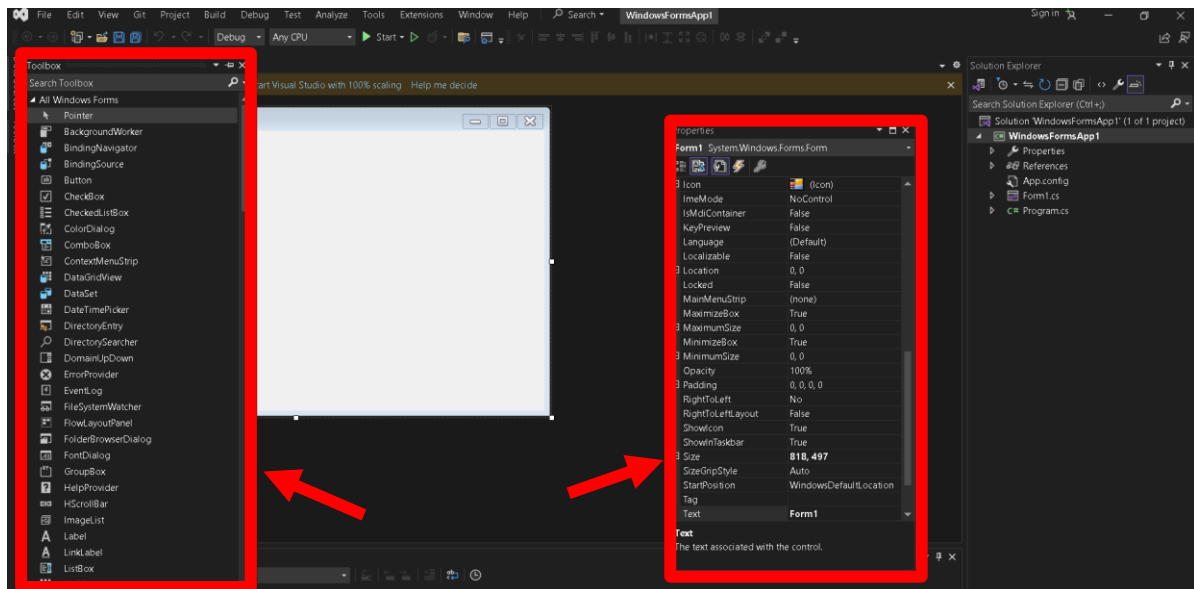
3.3.1 Frontend:



VISUAL STUDIO and C#

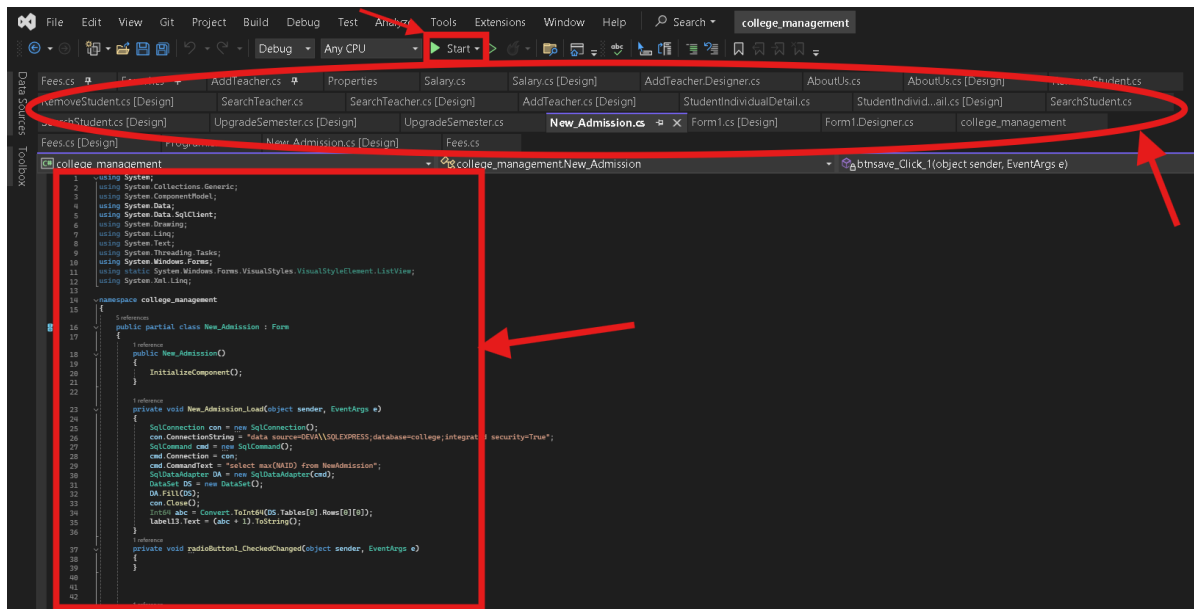
- For the frontend or user interface of the college management system, the .NET Framework was chosen as the development platform, with C# as the programming language. This decision was driven by several key factors.

- Firstly, the .NET Framework, developed by Microsoft, provides a comprehensive and robust software framework for building Windows desktop applications, web apps, and other software components. It offers a rich set of class libraries, development tools, and a runtime environment that simplifies the development process and enhances application performance.



WINDOW FORM and DESIGNING TOOLS

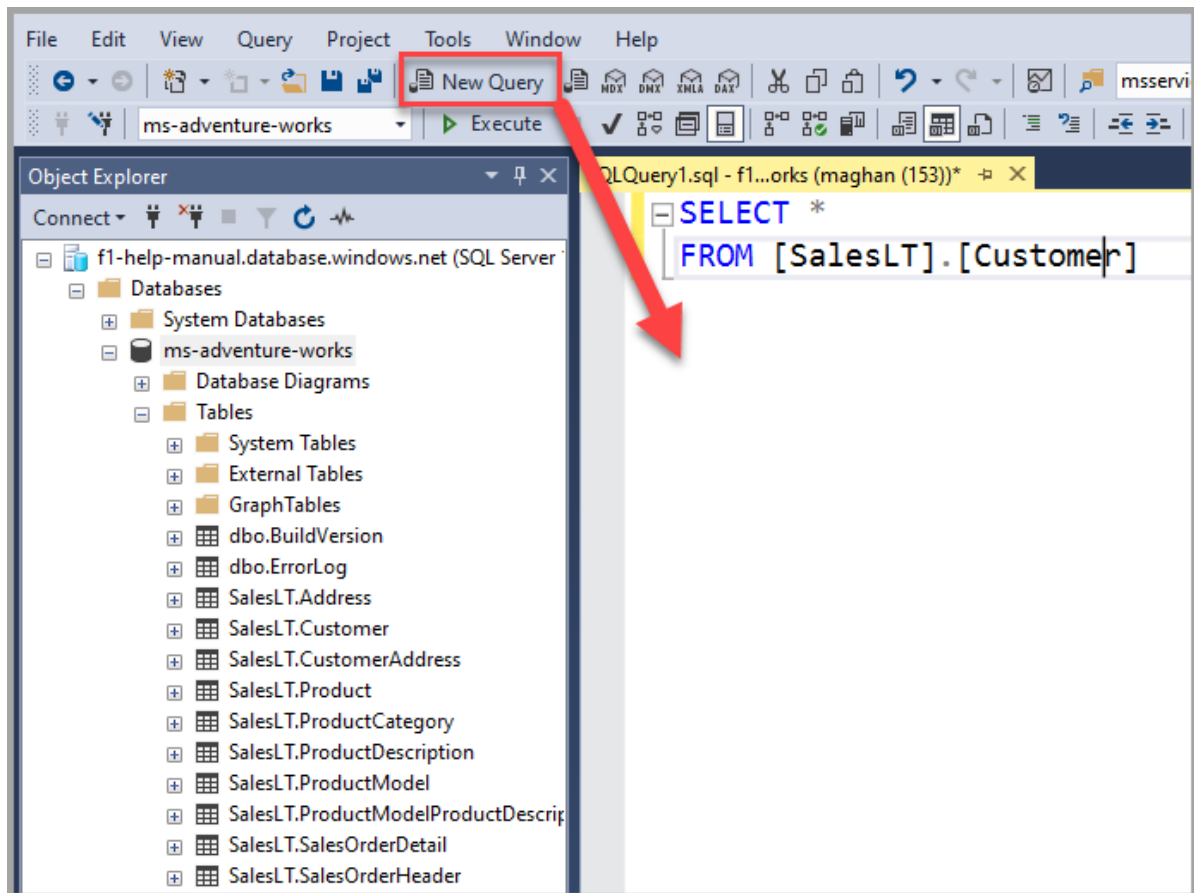
- Within the .NET ecosystem, as a language like C++, C# seamlessly supports Object-Oriented Programming (OOP) concepts like encapsulation, inheritance, and polymorphism. This facilitates the creation of reusable, maintainable, and extensible code, which is crucial for a complex system like a college management application.



CODING PART, ALL FORMS and START BUTTON

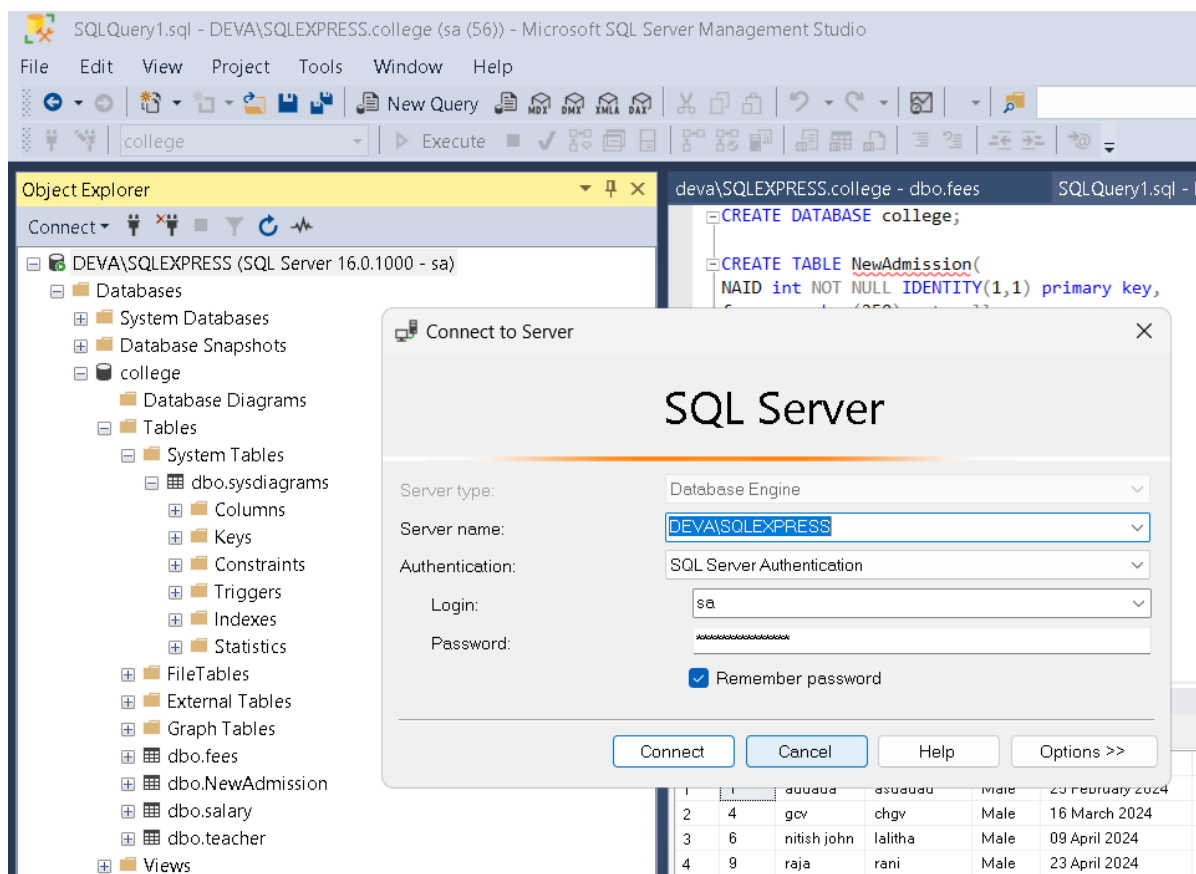
- Furthermore, the Visual Studio Integrated Development Environment (IDE) offered by Microsoft, along with its powerful debugging tools and code editors, further enhances the productivity of C# developers.

3.3.2 Backend:



SQL Server and SQL

- For the backend database component, Microsoft SQL Server was chosen as the Relational Database Management System (RDBMS). SQL Server is a highly reliable, scalable, and feature-rich database system developed by Microsoft, making it a natural choice for integration with the .NET Framework and C# frontend.



DATA TABLE and SQL SERVER AUTHENTICATION

- The decision to use SQL Server was it provides robust security features, such as user authentication, data encryption, SQL Server offers a user-friendly graphical interface for database administration and granular access controls, ensuring the protection of sensitive institutional data within the college management system. It also supports advanced features like stored procedures, triggers, and views, enabling the implementation of complex business logic and data validation rules directly within the database.
- To facilitate communication between the C# frontend and the SQL Server backend, the application utilizes SQL connections. This approach allows the frontend to execute SQL queries and commands on the database, enabling data retrieval, insertion, updates, and deletions as required by the various modules of the college management system.

Overall, the combination of the .NET Framework with C# for the frontend and Microsoft SQL Server for the backend database provides a powerful and cohesive technology stack for developing the college management system as a Windows application. This choice leverages the strengths of Microsoft's ecosystem, ensuring a consistent development experience, seamless integration, and the ability to create a robust, scalable, and secure solution to meet the institution's needs.

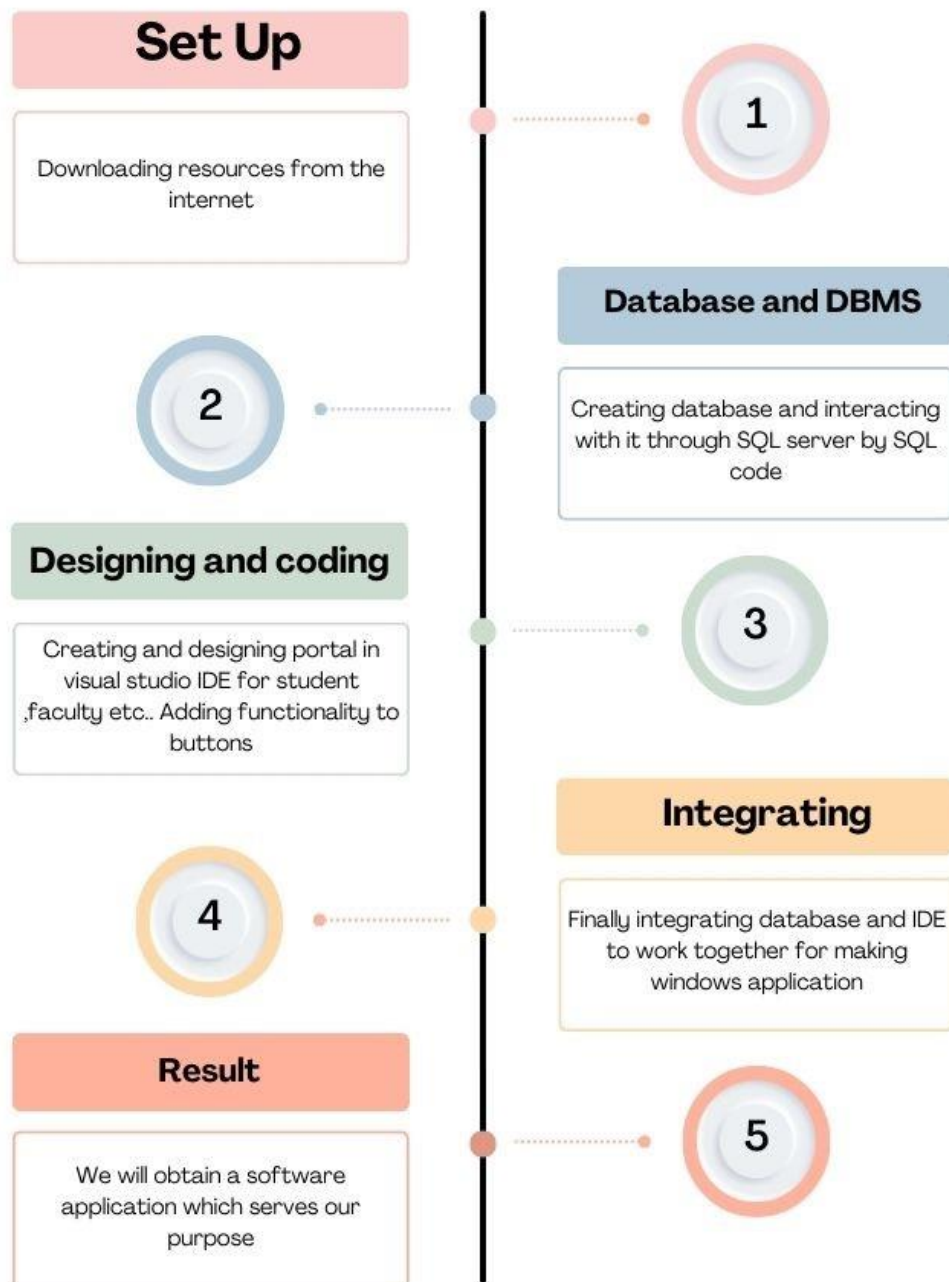
3.4 Integration and Testing:

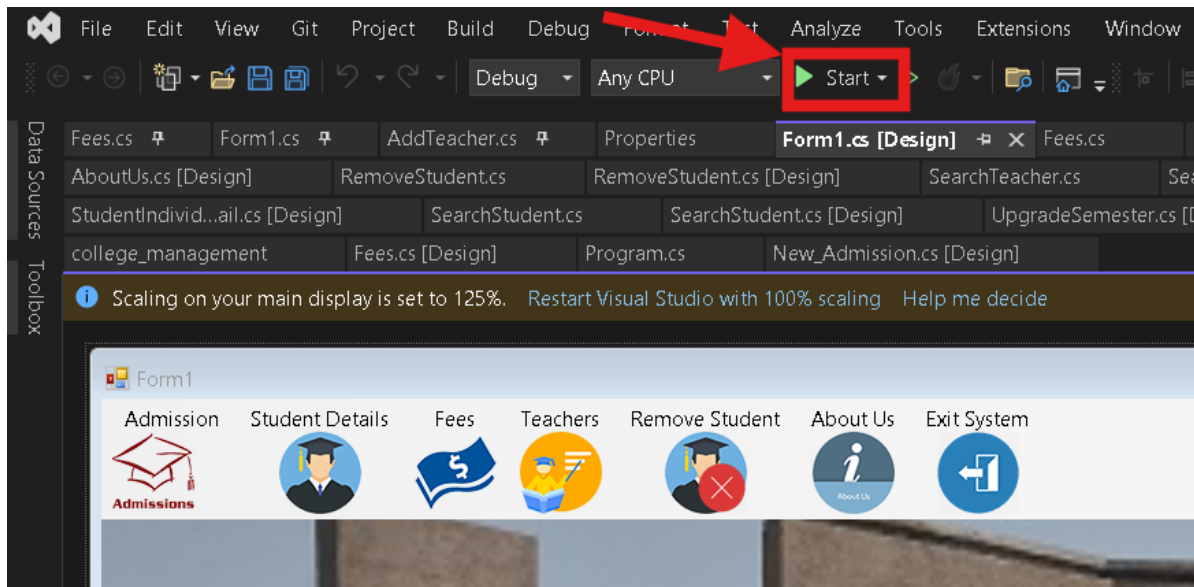
Developing a College Management System portal with .NET, C#, and SQL Server involves implementing a 2-tier architecture for frontend and backend integration. Database and DBMS principles guide the design of the database schema and data management. An Entity-Relationship (ER) diagram visually represents the database structure. During compilation and execution, .NET applications undergo translation, optimization, and execution phases to produce executable code.

4. RESULTS:

The process of creating a database and interacting with it through SQL Server using SQL code comprises several sequential steps. Initially, one must conceptualize and design the database schema, delineating tables, columns, and their relationships. Subsequently, SQL queries facilitate diverse operations including insertion, updating, deletion, and retrieval of data. Ultimately, interaction with the database is accomplished via SQL Server Management Studio or programmatically through execution of SQL commands from applications.

College management system

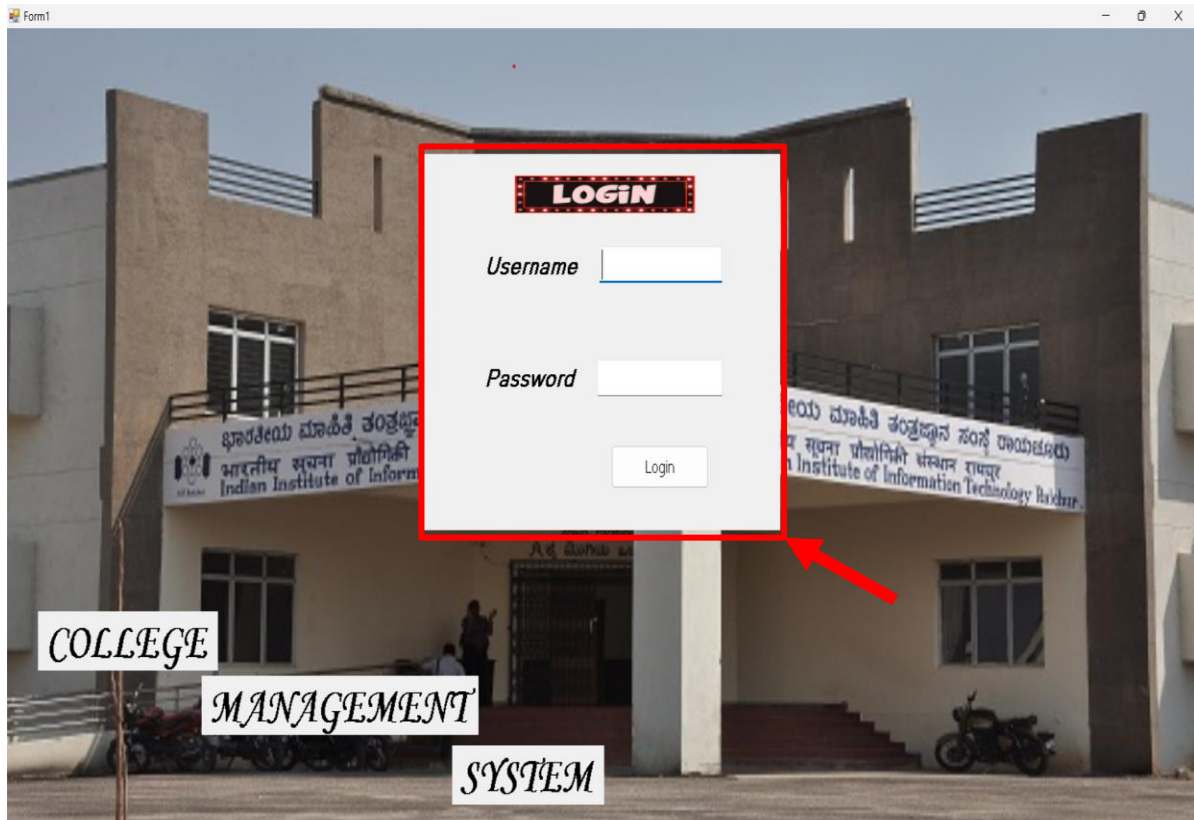




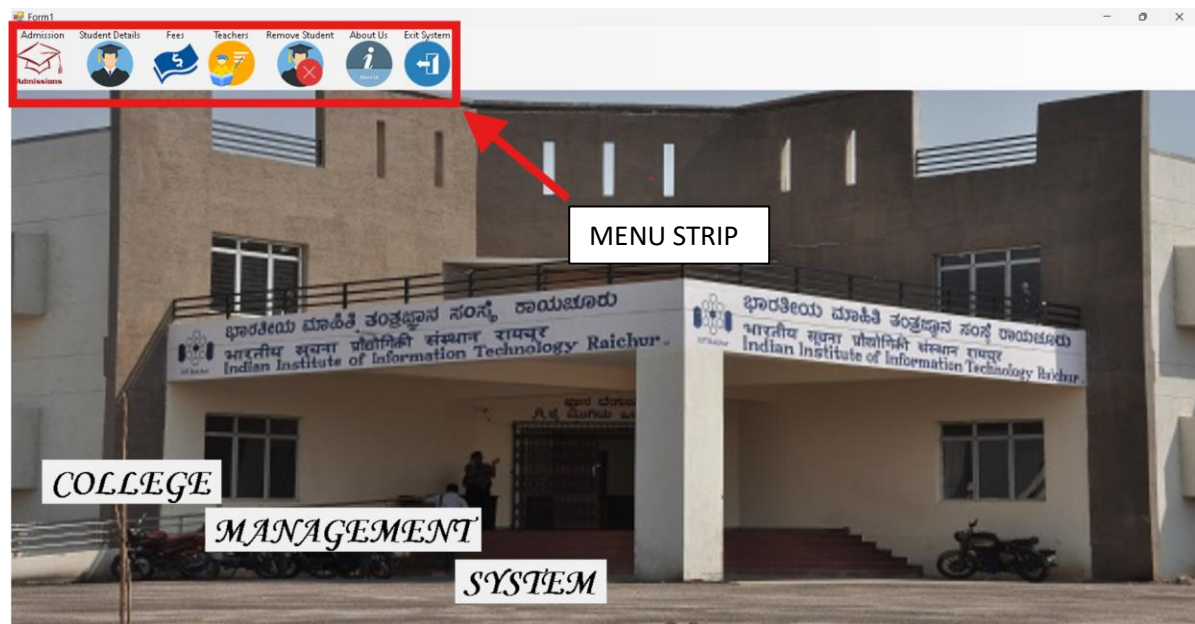
First thing to start the project is, we should click start button in the Visual Studio IDE.

4.1 Home Page:

The system requires a username and password for login. For students, the default credentials are "student" for both the username and password fields. Upon successful login, students are directed to the Home Page. This page prominently displays various portals like Admission, Student Details, Fees, Teachers, Remove Student, About Us and Exit. These portals allow admins to access different functionalities of the CMS such as applying for admission, viewing their details and fee records, information about faculty members, as well as an overview of the system.



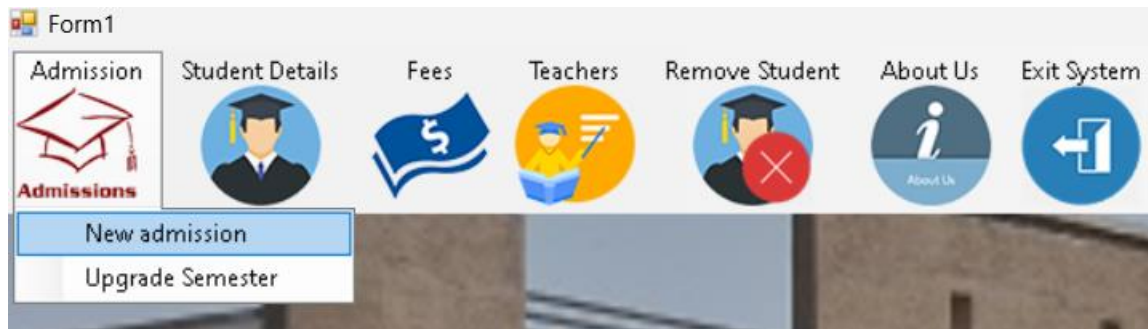
LOGIN INTERFACE



HOME PAGE

4.2 Admission

It contains new admission and upgrade semester portals.



ADMISSION

4.2.1 New Admission module:

New Admission Module:

This module handles the admission process for new students into the college. It consists of a form with input fields to capture the relevant details of the prospective student, such as name, date of birth, contact information, and any other required particulars.

Submit Button:

The Submit button is responsible for executing an INSERT command into the database table designated for storing new admission records. When clicked, it triggers the following actions:

1. Input Validation: The module first validates the user-entered data to ensure it meets the required criteria and format.
2. Data Insertion: If the input data is valid, an SQL INSERT statement is constructed dynamically, incorporating the values from the form fields.

3. **Registration ID Generation:** The module automatically generates a unique registration ID for the new admission record. This can be done using techniques like auto-incrementing primary keys or applying a specific algorithm to generate a unique identifier.

4. **Database Operation:** The SQL INSERT statement, along with the generated registration ID, is executed against the database table designated for new admissions. This results in a new record being inserted into the table with the provided details and the assigned registration ID.

Reset Button:

The Reset button is designed to clear all input fields on the New Admission form, allowing the user to start fresh with a new admission entry. When clicked, it resets all form fields to their default or empty state, ensuring no residual data remains from the previous entry.

deva\SQLEXPRESS....dbo.NewAdmission		deva\SQLEXPRESS.college - dbo.Table_2	
	Column Name	Data Type	Allow Nulls
	NAID	int	<input type="checkbox"/>
	fname	varchar(250)	<input type="checkbox"/>
	mname	varchar(250)	<input type="checkbox"/>
	gender	varchar(10)	<input type="checkbox"/>
	dob	varchar(50)	<input type="checkbox"/>
	moblie	bigint	<input type="checkbox"/>
	email	varchar(150)	<input type="checkbox"/>
	semester	varchar(100)	<input type="checkbox"/>
	prog	varchar(150)	<input type="checkbox"/>
	sname	varchar(150)	<input type="checkbox"/>
	duration	varchar(120)	<input type="checkbox"/>
	address	varchar(250)	<input type="checkbox"/>
			<input type="checkbox"/>

	NAID	fname	mname	gender	dob	moblie	email	semester	prog	sname	duration	address
1	1	addada	asdada	Male	25 February 2024	468644678	dadadadadad	4th Sem	C++	aadadad	2020-2024	aadadwd
2	4	gcv	chgv	Male	16 March 2024	524	fdgh	4th Sem	C Programming	retfgh	2019-2023	asdxfov
3	6	nitish john	lalitha	Male	09 April 2024	9441577383	nitish@gmail.com	1st Sem	Python	sri prakash	2023-2027	vizag,india
4	9	raja	rani	Male	23 April 2024	986532147	raja@gmail.com	6th Sem	Java	sri vignesh	2022-2026	Tamil Nadu,India
5	10	shyaam	shobanadevi	Male	18 October 1998	9685741230	shyaam@gmail.com	8th Sem	Java	RSK	2020-2024	Tamil Nadu,India
6	11	shyaam	shobanadevi	Male	18 October 1998	9685741230	shyaam@gmail.com	8th Sem	Java	RSK	2020-2024	Tamil Nadu,India
7	12	deva	shobanadevi	Male	03 December 2024	52836974102	deva@gmail.com	2nd Sem	C++	KNMS	2023-2027	Tamil Nadu,India

New Admission DATABASE

Admission Student Details Fees Teachers Remove Student About Us Exit System

Admissions

COLLEGE MANAGEMENT

ADMISSIONS

Registration NO: 13

Full Name

Mother's Name

Gender ☐ Male ☐ Female

Date Of Birth

Mobile No.

Email ID

Semester

Programming

School Name

Duration

Address

Submit Reset

New_Admission

ADMISSIONS

Registration NO: 13

Full Name: Devanand

Mother's Name: shobanadevi

Gender: ☒ Male ☐ Female

Date Of Birth: 10 January 2024

Mobile No.: 5642317890

Email ID: deva@gmail.com

Semester: 2nd Sem

Programming: C#(C Sharp)

School Name: KNMS

Duration: 2023-2027

Address: Tamil Nadu,India

Submit Reset

Data

✗ Data Saved.Remember the Registration ID

OK

4.2.2 Upgrade Semester module:

The Upgrade Semester module allows administrators to update the semester details for existing students in the database by clicking upgrade button. It displays a list of enrolled students, and the user can select a student and change their semester value, like from Semester 1 to Semester 2. The module validates the new semester input, constructs an SQL UPDATE statement, and executes it to modify the student's record in the database table with the

updated semester information.

Admission Student Details Fees Teachers Remove Student About Us Exit System

Admissions

COLLEGE MANAGEM

UpgradeSemester

QUARTER SYSTEM SEMESTER SYSTEM

Select Semester for Old Students Admission

From: --Select--

To: --Select--

UPGRADE

Select Semester for Old Students Admission

From: 3rd Sem

To: 4th Sem

UPGRADE

Confirm?



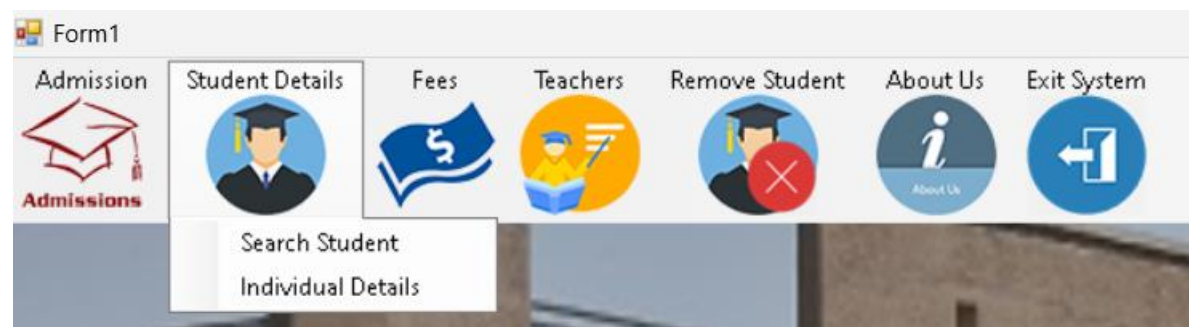
Semester Upgrade Warning!

OK

Cancel

4.3 Student Details

It has search student module and individual details module.

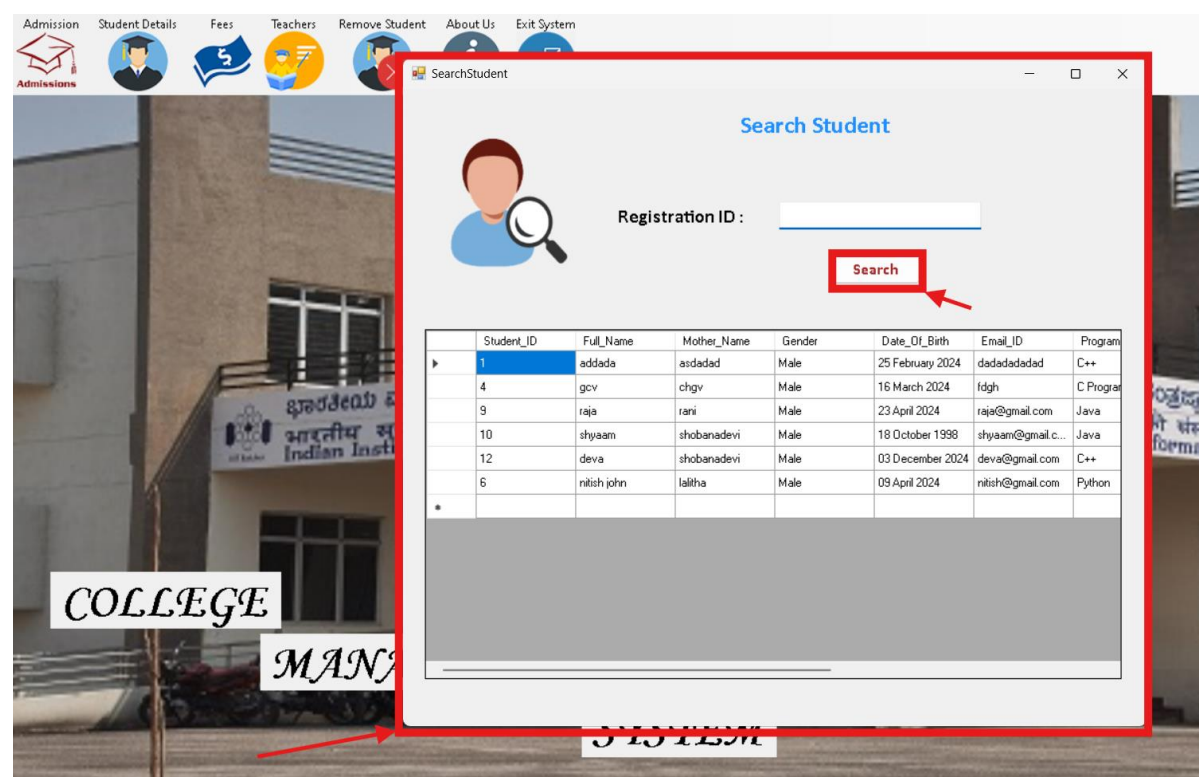


4.3.1 Search Student module:

- The Search Student module is designed to retrieve and display comprehensive student details by integrating data from multiple tables within the database. It leverages the concept of **foreign keys** to establish relationships between tables, enabling efficient data retrieval and cross-referencing of information.
- When a search query is initiated, the module constructs a SQL SELECT statement that joins the new admission table, containing core student information, with the fee table, which stores fee payment records. The join operation is facilitated by utilizing the foreign key relationship established between the two tables, typically linking the student identification fields.
- The SQL statement incorporates search parameters or constraints, such as filtering for students who have paid fees

in the fee module. This condition is implemented by adding appropriate WHERE clauses that check the existence of corresponding fee records linked to each student entry via the foreign key reference.

- The query results, comprising student details from the new admission table and associated fee payment information, are then retrieved from the database and presented to the user through a user-friendly interface within the Search Student module.



4.3.2 Individual Details module:

This also same as search student module but without concept of foreign key, here it retrieves data only based on registration id by clicking Show Details button.

Search Individual Student Details			
Registration ID :		11	
		Show Details	
Full Name:	shyaam	Standard:	8th Sem
Mother Name:	shobanadevi	Medium:	Java
Gender:	Male	School Name:	RSK
Date Of Birth	18 October 1998	Year:	2020-2024
Moblie:	9685741230	Address:	Tamil Nadu,India
E-mail:	shyaam@gmail.com		
Reset			

4.4 Fees Module

The fee module in the CMS facilitates insertion of fee data into the fee table, akin to the new admission module. It includes a foreign key from the new admission table, ensuring correlation between admission and fee records, streamlining data management and ensuring data integrity.

Column Name	Data Type	Allow Nulls
fid	int	<input type="checkbox"/>
NAID	int	<input type="checkbox"/>
fees	int	<input type="checkbox"/>

	fid	NAID	fees
1	1	1	50000
2	2	2	90000
3	3	3	50000
4	4	4	53333
5	5	9	50000
6	6	10	50000
7	7	12	100000
8	8	6	50000

Fee table

The screenshot displays a web application for college management. The top navigation bar includes links for Admission, Student Details, Fees, Teachers, Remove Student, About Us, and Exit System. The 'Fees' form is highlighted with a red border. It features a header image of a building with the text 'COLLEGE MANAGEMENT' overlaid. The form fields are: Registration No., Full Name, Mother Name, Duration, and Fees. A red arrow points to the 'Submit' button at the bottom right of the form.

COLLEGE
MANAGEMENT


Registration No:

Full Name:

Mother Name:

Duration:

Fees:



Registration No: 12

Full Name: deva

Mother Name: shobanadevi

Duration: 2023-2027

Fees: 50000

[Submit](#)

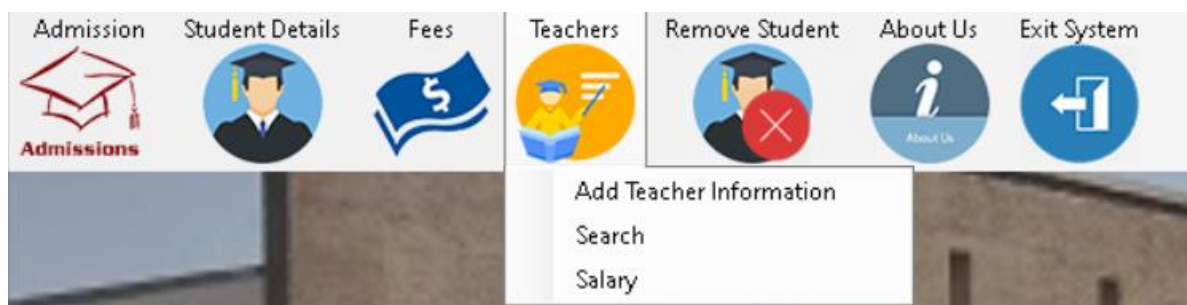
×

Fees was Already Submitted !!

OK

4.5 Teachers

It has add teacher information module, search teacher module and Salary module.



4.5.1 Add Teacher Information module:

The teacher module in the CMS mirrors the new admission module, operating similarly with a dedicated teacher table.

 A screenshot of the 'Add Teacher' form in the CMS. The form is titled 'Basic Information' and includes the following fields:

- Teacher ID: 6
- Full Name:
- Semester:
- Gender: ☐ Female ☐ Male
- Programming:
- Date Of Birth:
- Duration:
- Mobile No:
- Address:
- E-Mail ID:

 At the bottom of the form, there are two buttons: 'Submit' (green) and 'Reset' (orange). Red arrows point to these buttons. The background of the form shows a building with a sign that reads 'COLLEGE MAN...'.

4.5.2 Search module:

This module mirrors the individual details module.

Search Teacher

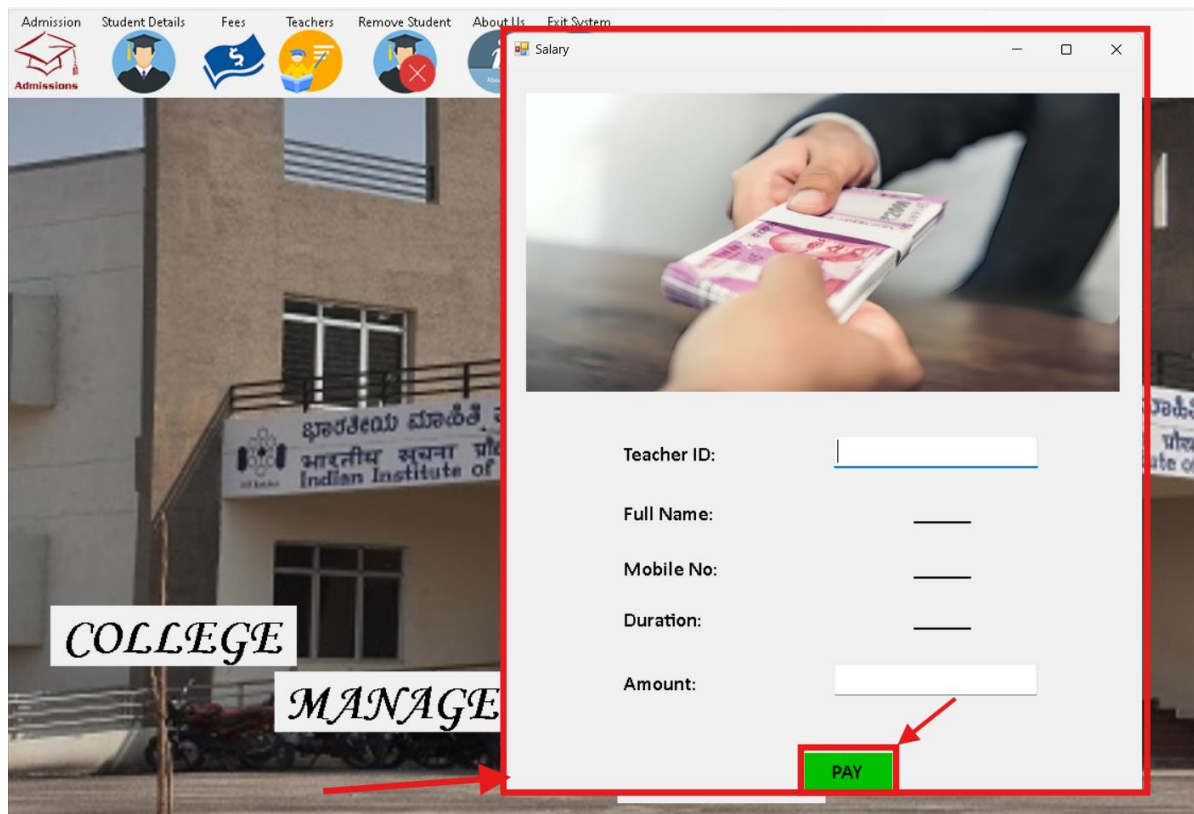
Registration ID :

Search

ID	fname	gender	dob	mobile	email	semester	prog	ye
1	jin	Male	03 September 20...	8520963852	jin@gmail.com	2nd Sem	Java	202
2	Hitler	Male	03 December 2024	5461237890	Hitler@gmail.com	4th Sem	C#(C Sharp)	202
3	Tinlin	Male	09 April 2024	8529637412	Tinlin@gmail.com	4th Sem	Java	202
4	mulhu	Male	18 April 2024	5236981470	mulhu@gmail.com	5th Sem	Java	202
5	devgi	Female	16 April 2024	741852096332	gbtg@gmailcom	7th Sem	JSP	202

4.5.3 Salary module:

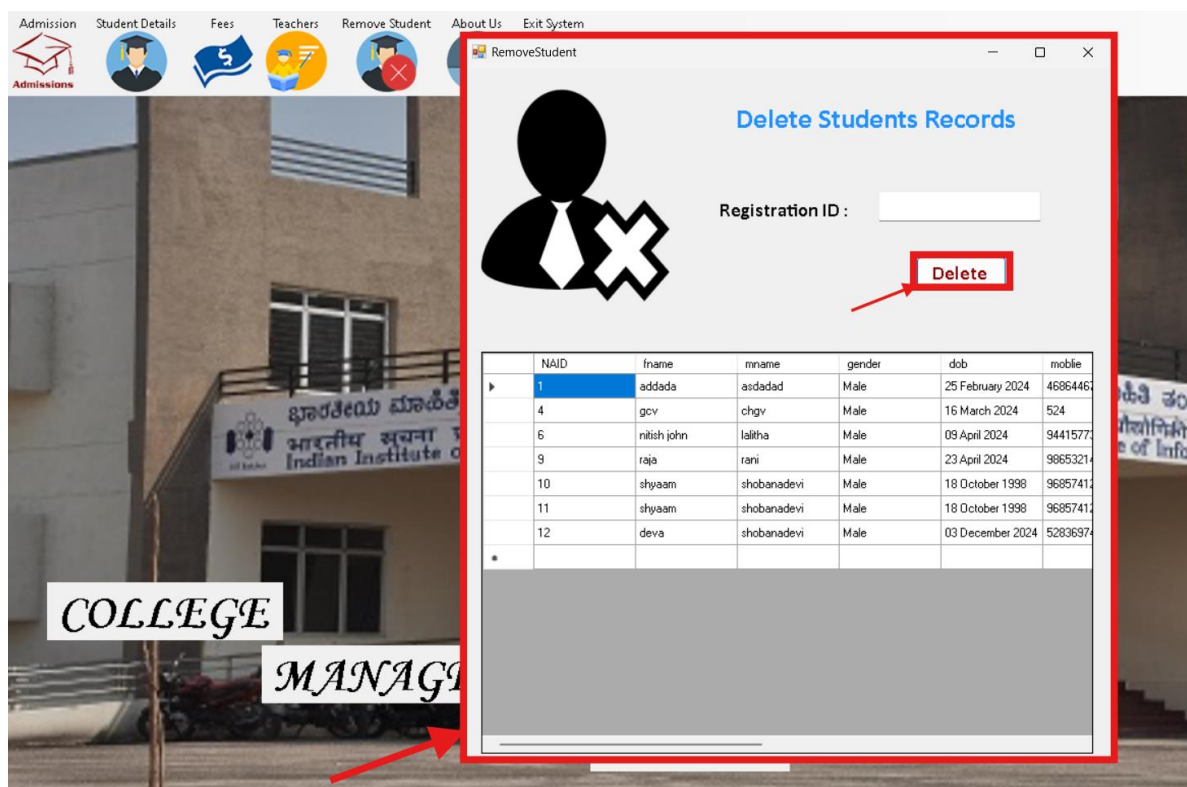
This module mirrors the fee module, operating similarly with a dedicated salary table.



4.6 Remove Student

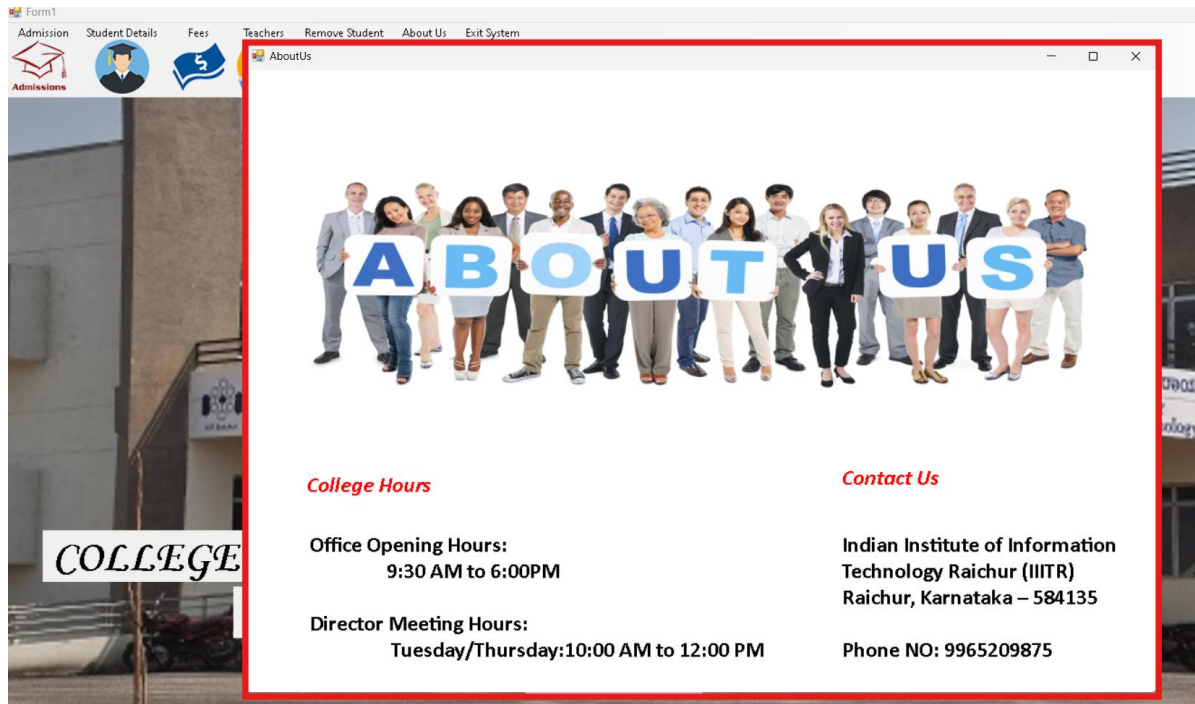
- The Remove Student module facilitates the permanent removal of student records from the core database upon their completion or departure from the institution. It executes a deletion operation on the student's entry in the new admission table, effectively purging their comprehensive data from the system.
- When initiating a removal, the module constructs a SQL DELETE statement targeting the specific record in the new admission table, typically identified by the unique registration ID assigned during the initial admission process. Upon successful execution of this statement, the student's record is permanently expunged from the table.
- Consequently, any attempt to interact with the deleted student's data, such as attempting fee payments through the fee portal, will result in an error or exception being raised.

This is because the underlying foreign key relationships and data integrity constraints within the database ensure that operations cannot be performed on non-existent or orphaned records.



4.7 About Us

This module provides essential information about the college, including office hours, address, contact number, and other pertinent details.



4.8 Exit System

It will just exit from application.



5. DISCUSSION:

- The development and implementation of the integrated college management system has yielded significant improvements in operational efficiency, data management, and stakeholder experiences across the institution. The key findings demonstrate the system's ability to streamline core academic and centralized data storage, and intuitive user interfaces.
- Utilizing a database over a file system or other storage systems offers several significant advantages in terms of data management, security, scalability, and accessibility. Firstly, databases provide a structured and organized approach to storing data, allowing for efficient retrieval and manipulation using queries and indexes. This structured format enables users to easily locate and retrieve specific data points, enhancing overall data management efficiency.
- Additionally, databases are designed to scale seamlessly with growing data volumes and user demands. They can handle large datasets and concurrent user requests efficiently, ensuring optimal performance even as the system expands over time. This scalability enables organizations to accommodate increasing data requirements without sacrificing performance or reliability.
- Overall, the structured approach, robust security measures, scalability, and accessibility provided by databases make them a superior choice over file systems or other storage systems for managing and storing data effectively in modern organizations.

6. CONCLUSION:

- A notable conclusion is the substantial reduction in manual paperwork and administrative workloads achieved by digitizing processes like student admissions, course enrollment and fee processing. This has resulted in cost savings and resource optimization for the college.
- The successful implementation showcases the potential for similar higher education institutions to adopt comprehensive management systems, driving digital transformation and positioning themselves as future-ready organizations in an increasingly competitive landscape.

REFERENCES

1. https://www.tutorialspoint.com/mysql/mysql_tutorial.pdf

2. https://edtechbooks.org/learning_mysql
3. <https://zetcode.com/csharp/mysql/>
4. <https://learn.microsoft.com/en-us/shows/on-net/how-do-i-use-sql-server-with-csharp-and-dotnet>
5. <https://learn.microsoft.com/en-us/dotnet/api/system.data.sqlclient.sqlconnection?view=dotnet-plat-ext-5.0&viewFallbackFrom=dotnet-plat-ext-8.0>
6. <https://learn.microsoft.com/en-us/visualstudio/designers/walkthrough-windows-forms-designer?view=vs-2022>