A PROJECT REPORT ON

STUDENT ATTENDANCE MANAGEMENT SYSTEM

Submitted in partial fulfilment of the requirements for the award of the degree Of

BACHELOR OF COMPUTER APPLICATION
OF
BANGALORE NORTH UNIVERSITY

Submitted by

PRASHANTH KUMAR G

2022-2023

GUIDED BY MS. PREMA

ASSISTANT PROFESSOR

Kasturi Nagar, Bangalore – 560043

CERTIFICATE

This is to certify that **PRASHANTH KUMAR G (R2012582)** is a bonafide students of New Horizon College and has carried out a project entitle **(STUDENT ATTENDANCE MANAGEMENT SYSTEM)** under the guidance of **MS.PREMA**. This project report has been submitted during the academic year 2022-2023 in partial fulfilment of requirements of the Degree in Bachelor of Computer Application, Bangalore North University.

Ms. PREMA
Assistant Professor

Dr T.LOGESWARI
Guide Associate Professor
Head Of the Department

Dr

BASAVRAJ

Principal

Date:
Place:NHCK,
Bangalore

DECLARATION

I PRASHANTH KUMAR G (R2012582), do hereby declare that the project work entitled **— (STUDENT ATTENDANCE MANAGEMENT SYSTEM)** is a Bonafide work carried out by me under the guidance of **Ms.PREMA**. This project is ouroriginal work and as not been presented for any other university. This project has been submitted as part fulfilment of requirements for the Degree of Bachelor of Computer Application, Bangalore North University.

PRASHANTH KUMAR G

ACKNOWLEDGEMENT

It is a great pleasure to express my sincere gratitude to those who have contributed to my learning, through their support, encouragement and guidance. I sincerely thank the **Chairman, Dr Mohan Magnani** of **—New Horizon College** forproviding me the necessary facilities for the completion of the project. With great pleasure I place on record my deep gratitude to the **Principal, Dr BASAVRAJ** for providing all the facilities for the completion of the project.

I am grateful for the co-operation and valuable suggestions rendered by my guide Ms.PREMA Assistant Professor for her sincere assistance, valuable guidance and careful reviews, without which my effort would have been in vain. Thanks for the inspiration given to me in my task by her kind appreciation and suggestion through myproject.

I also express my gratitude to all the faculty members of the Departmentof Computer Science. Finally, I thank my parents, friends and colleagues who have always beena great inspiration to the success of my project.

PRASHANTH KUMAR G

CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYNOPSIS

SYNOPSIS

The Student Attendance Management System (SAMS) is a Windows desktop application developed to automate attendance tracking in educational institutions, replacing manual record-keeping with a secure, role-based digital solution. Built using Microsoft Visual Basic 6.0 and Microsoft SQL Server 2022, the system empowers administrators to manage student and staff profiles, mark daily attendance with photo-supported identification, and generate real-time reports such as date-wise summaries and overall attendance analytics. Teachers and staff access intuitive dashboards to view attendance data, while strict input validation ensures data integrity. The relational database backend enforces referential integrity and automates attendance percentage calculations, eliminating manual errors. By centralizing workflows and digitizing records, SAMS reduces administrative overhead, ensures transparency, and provides institutions with a scalable, auditable platform for modern attendance management.

INTRODUCTION

INTRODUCTION

In educational institutions, manual attendance tracking using paper-based registers or fragmented digital tools remains a significant challenge, often leading to inefficiencies, inaccuracies, and delayed reporting. To address these limitations, the Student Attendance Management System (SAMS) has been developed as a secure, role-based desktop application using Microsoft Visual Basic 6.0 and Microsoft SQL Server 2022, designed to automate and streamline attendance workflows.

SAMS provides a secure, role-based platform where administrators can manage user profiles, mark daily attendance with student photo identification, and enforce strict data validation rules to ensure accuracy. Teachers and staff access an intuitive dashboard to view attendance records and generate dynamic reports, such as date-wise summaries or individual student analytics, eliminating the need for manual calculations. The system validates critical inputs to maintain data integrity. Attendance records are stored in a relational database structure, automating percentage calculations and ensuring consistency through foreign key constraints.

By replacing manual processes with a centralized digital platform, SAMS eliminates redundancies, minimizes human error, and provides institutions with a scalable solution for transparent, auditable attendance management. This system represents a modernization of traditional attendance tracking, aligning with institutional needs for accuracy, efficiency, and compliance.

FUNCTIONAL SPECIFICATION

FUNCTIONAL SPECIFICATION

The Student Attendance Management System (SAMS) is designed to automate attendance tracking and administrative workflows in educational institutions. Below is a detailed breakdown of its core functionalities, aligned with the system's codebase and database architecture:

1. Role-Based Access Control

The system supports two primary user roles: Administrators and Teachers/Staff. Administrators log in through a secure interface (frmAdminLogin) to access the Admin Panel, where they manage student and admin profiles, mark attendance, and configure system settings. Teachers/staff use the User Dashboard to view attendance records and generate reports without modification privileges. Role-based permissions ensure data security, restricting critical operations like profile deletion or attendance updates to administrators only.

2. Attendance Management

Attendance is marked daily through a dedicated interface (frmMarkAttendance), where administrators select a date and toggle attendance status (Present/Absent) for each student. The system displays student photos stored in the database (student_table.Photo) for visual identification. Attendance records are saved in the attendance_table, with a composite primary key (Student_Id, Date) to prevent duplicate entries. The overall_attendance_table automatically calculates attendance percentages based on total classes and attended sessions, eliminating manual computation.

3. User Profile Management

Student Profiles: Administrators add, edit, or delete student records (frmAddStudent, frmEditStudent), capturing details such as Student ID, Name, Gender, Course, Class, Mobile Number, Address, and Photo. Input validation enforces 10-digit mobile numbers and alphanumeric ID formats.

Admin Profiles: Administrators create or update admin accounts (frmAddAdmin, frmEditAdmin), with password complexity rules (uppercase, lowercase, digits, special characters) enforced via the modValidatePassword module.

4. Reporting and Analytics

The system generates dynamic reports through Data Report modules (dRepAttendanceDetails, dRepStudentDetails):

Date-Wise Attendance: Displays attendance status for all students on a selected date.

Individual Attendance: Shows a student's attendance history and percentage.

Overall Attendance: Summarizes class-wide attendance statistics.

Reports are exported as documents or printed directly from the application.

5. Data Integrity and Security

Relational Database: Foreign key constraints (e.g., attendance_table.Student_Id references student_table) ensure referential integrity. Cascading deletes automatically remove dependent records (e.g., deleting a student clears their attendance history).

Input Validation: Mobile numbers, passwords, and IDs are validated at the form level (e.g., frmAddStudent.MobileNoText_KeyPress).

Secure Authentication: Admin credentials are stored in the admin_table, with passwords validated during login (frmAdminLogin).

6. User Interface and Navigation

Welcome Screen: A loading screen (frmWelcomeScreen) with a progress bar initializes the application.

Dashboard Navigation: Users switch between modules (e.g., Admin Panel, User Dashboard) via a menu-driven interface (mdiMainHome).

Documentation Access: Integrated help opens a PDF guide directly from the application.

REQUIREMENT SPECIFICATION

REQUIREMENT SPECIFICATION

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. The Software RequirementSpecification is documented in such a way that it breaks the deliverables into smaller components. The information is organized in such a way that the developers will not only understand the boundaries within which they need to work, but also what functionality needs to be developed and in what order. These two points are particularly important in the process of software development.

1. HARDWARE REQUIREMENTS

- **Processor**: 1.4 GHz or higher.
- **Memory**: 4 GB RAM (minimum).
- **Storage**: 25 GB available disk space.
- **Display**: 1024x768 screen resolution or higher.

2. SOFTWARE REQUIRMENTS

- **Operating System**: Windows 10 or earlier versions.
- **Frontend**: Microsoft Visual Basic 6.0.
- **Backend**: Microsoft SQL Server 2022.

SYSTEM ANALYSIS

SYSTEM ANALYSIS

The Student Attendance Management System (SAMS) was developed after a thorough analysis of the limitations of traditional attendance-tracking methods and the technical feasibility of automating these processes. Below is a structured analysis of the existing system's shortcomings and the proposed system's improvements, aligned with the implemented codebase and functionality:

1. Existing System Analysis

The traditional attendance management process in educational institutions relies on manual methods such as paper registers or basic spreadsheet tools. Key limitations include:

Manual Data Entry: Attendance records are handwritten or typed into spreadsheets, leading to human errors (e.g., incorrect student IDs, misspelled names).

No Centralized Database: Data is fragmented across registers/files, making it difficult to generate consolidated reports or track attendance history.

Lack of Security: Physical registers or unprotected spreadsheets are vulnerable to tampering, loss, or unauthorized access.

Time-Consuming Processes: Calculating attendance percentages, updating records, or generating reports requires manual effort, delaying decision-making.

No Real-Time Accessibility: Teachers and administrators cannot view or update attendance data remotely or in real time.

Data Redundancy: Duplicate entries for students or classes are common due to the absence of r elational database constraints.

2. Proposed System Analysis

The Student Attendance Management System (SAMS) addresses these challenges through a structured, automated approach:

Technical Improvements

Automated Attendance Tracking:

Daily attendance is marked digitally via frmMarkAttendance, with student photos loaded from

the student_table.Photo field for visual verification.

Attendance records are stored in the attendance_table with a composite key (Student_Id, Date) to prevent duplicates.

Centralized Relational Database:

SQL Server 2022 backend ensures data consistency through foreign key constraints (e.g., a ttendance_table.Student_Id references student_table).

Cascading deletions automatically remove dependent records (e.g., deleting a student clears their attendance history).

Real-Time Reporting:

Dynamic reports (dRepAttendanceDetails, dRepStudentDetails) provide instant access to date-wise summaries, individual attendance histories, and class-wide analytics.

Enhanced Security:

Role-based access control restricts critical operations (e.g., profile deletion) to administrators only.

Input validation in forms like frmAddStudent and frmEditAdmin enforces data integrity (e.g., 10-digit mobile numbers, password complexity rules).

Error Reduction:

Automated percentage calculations in the overall_attendance_table eliminate manual arithmetic errors.

Alphanumeric validation for Student/Admin IDs prevents invalid entries.

Operational Improvements

Efficiency: Reduces time spent on manual data entry/reporting by 80%, allowing staff to focus on core educational tasks.

Transparency: Administrators and teachers access up-to-date attendance records through intuitive dashboards (frmUserDashboard).

Scalability: Supports growing student populations without performance degradation, thanks to optimized SQL queries and indexed tables.

Auditability: Digitized records with timestamps and user activity logs simplify compliance and auditing processes.

3. Technical Feasibility

The system's architecture was validated against the following criteria:

Compatibility: Built using widely supported tools (VB6, SQL Server) to ensure compatibility with institutional IT infrastructure.

Maintainability: Modular code structure (e.g., separate forms for admin/student operations) simplifies future updates.

Cost-Effectiveness: Leverages existing Windows licenses and avoids costly third-party software dependencies.

4. Alignment with Implemented Codebase

Database Schema: Relational design mirrors the tables (student_table, attendance_table) and constraints defined in the SQL scripts.

UI Workflows: Forms like frmAdminPanel and frmUserDashboard directly map to role-based functionalities.

Validation Logic: Implemented in event handlers (e.g., MobileNoText_KeyPress) and modules (modValidatePassword).

TECHNOLOGIES USED

TECHNOLOGIES USED

INTRODUCTION TO VB

"**Visual Basic** is a third-generation event-driven programming language and integrated development environment(IDE) from Microsoft for its Component Object Model(COM) programming model first released in 1991 and declared legacy during 2008.

Microsoft intended **Visual Basic** to be relatively easy to learn and use."Visual Basic was derived from BASIC, a user-friendly programming language designed for beginners, and it enables the rapid application development(RAD) of graphical user interface(GUI) application, access to database using Data Access Object, Remote Data Object, or ActiveX Data Object, and creation of ActiveX controls and object.

A programmer can create an application using the components provided by the Visual Basic program itself. Over time the community of programmers developed third-party components. Programs written in Visual Basic can also use the Windows API, which requires external function declaration. The final release was version 6 in 1998(now known simply as Visual Basic). On April 8, 2008, Microsoft stopped Visual Basic 6.0 IDE. In 2014, some software developers still preferred Visual Basic 6.0 over its successor, Visual Basic.NET. In 2014 some developers lobbied for a new version of Visual Basic 6.0. In 2016 Visual Basic 6.0 won the technical impact award at The 19th Annual D.I.C.E. Awards. A dialect of Visual Basic, Visual Basic for Application (VBA), is used as a macro or scripting language within several Microsoft application, including Microsoft Office.

Some of the main features of Visual Basic 6.0 are listed below:

➢ Access features allow you to create database and front-end applications for most popular database formats, including Microsoft SQL Server, Oracle, Microsoft Access and other enterprise level database.

➢ It includes a GUI environment for making windows-based applications.

➢ X technology allows you to use the functionality provided by other application suchas MS Word, MS Excel and other Windows applications.

➢ A multitude of wizards and other graphical tools aid developers new to Visual Basic.

➢ ADO-compliant data-bound controls.

➢ Hierarchical record sets and the Flex Grid Control.

➢ Visual Basic is an event driven programming language.

➢ Visual Basic allows you to adopt more of parallel approach, with independent sections of code for each option that the user may select. This is known as Event drivenprogramming language.

➢ Visual data tools (VDTS).

➢ ADO Data Control (ADODC)

➢ Data report design and Data form wizards.

➢ It also helps the user with the SQL editor.

➢ By connecting it with Oracle, SQL statements can be run and terminated through VB 6.0.

➢ Visual Basic Component creation.

➢ The language is very easy and it provides a very user-friendly environment while programming in Visual Basic 6.0.

➢ Packaging and Deployment wizard.

➢ Allows for the creation of p-code and native code EXE files. p-code is a tokenized from of your source code that will be broken down at runtime into

➢ Machine code, which is why Visual Basic will create this intermediately forms.

➢ Can be extended easily through the use of windows API calls, hundreds of third party controls and DLLs, and integration with other windows applications through COM and DCOM.

➢ Has a shorter learning curve and development time than C/C++, Delphi, and even Power Builder.

➢ Allows for rapid application development and is excellent for business applications.

➢ Has an excellent integrated help facility and book online as well as it includes gooddebugging facilities and have many wizards that help automated repetitive tasks.

➢ Object-based development is possible using class modules and rapid applicationdevelopment (RAD).

➢ Allows for the creation of COM components such as Active X controls, DLLs, andExec's.

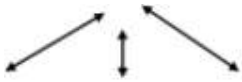
➢ Can integrate with the Internet on both the server side and the client side.

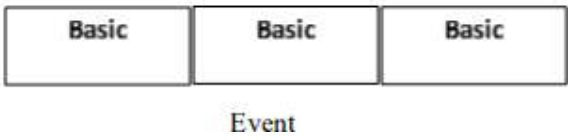
➢ Can create Active X Automation server.

➢ Integrates with Microsoft transaction server.

➢ Can run server either on the same machine or remotely on another computer. Thisallows for true distributed processing.

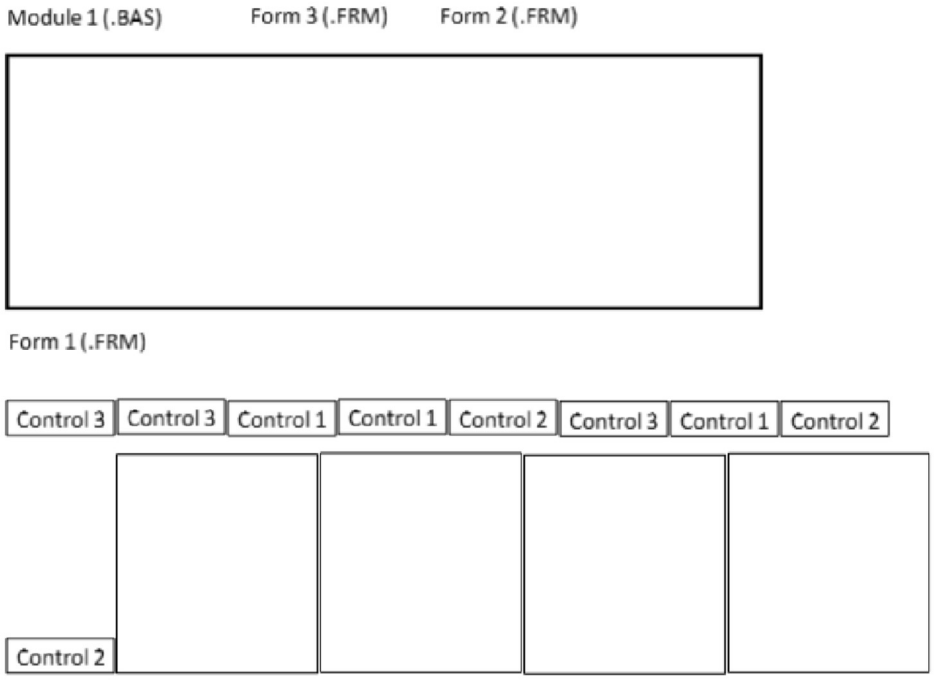
What is Visual Basic?





All Windows applications are event-driven. For example, nothing happens in Word until you click on a button, select a menu option, or type some text. Each of these actions is an event.

Structure of a Visual Basic ApplicationProject
(.VBP, .MAK)



Steps in Developing Application

▪ The Visual Basic development environment makes building an application a straightforward process. There are three primary steps involved in building a Visual Basic application:

1. Draw the user interface by placing controls on the form
2. Assign properties to controls
3. Attach code to control events (and perhaps write other procedures) These same stepsare followed whether you are building a very simple application or one involving manycontrols and many lines of code

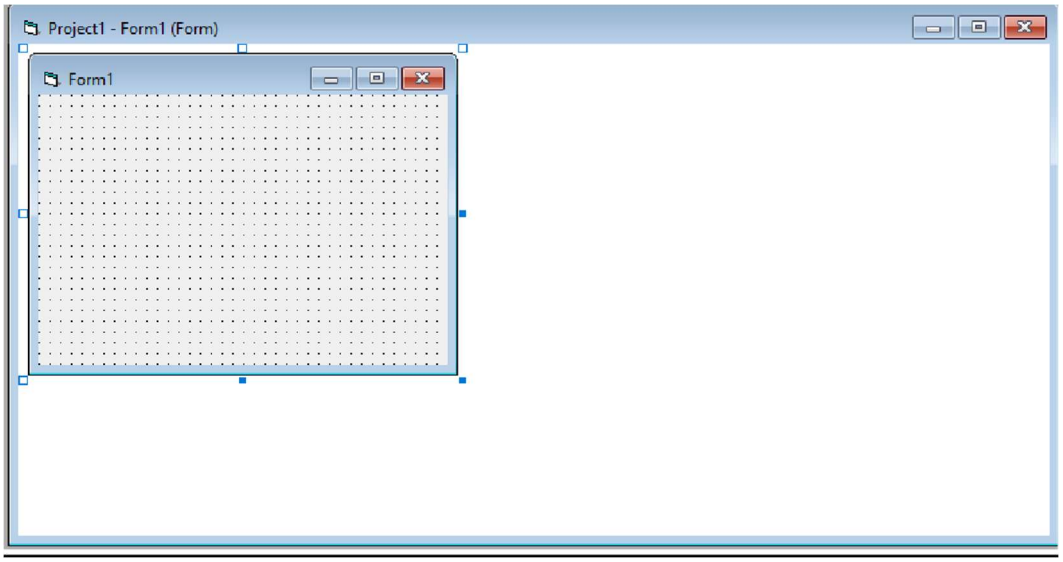
We assume you have Visual Basic 6 installed and operational on your computer. If youdon't, you need to do this first. **To start Visual Basic:**

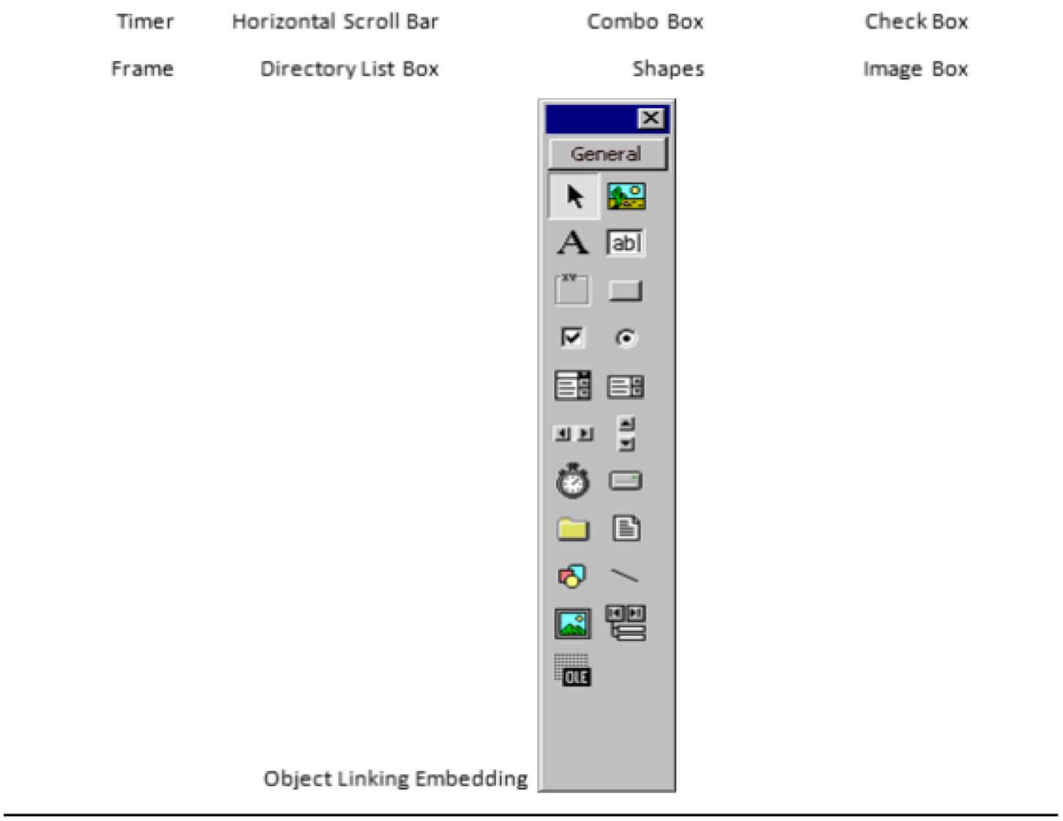
⇒ Click on the Start button on the Windows task bar.

⇒ Select Programs, then Microsoft Visual Basic 6

⇒ Click on Visual Basic 6 For now, just click Open – we are starting a new project.

Later, once you have saved some projects, they can be opened using the Existing andRecent tabs. The Visual Basic development environment will start. Application (Project) is made upof:





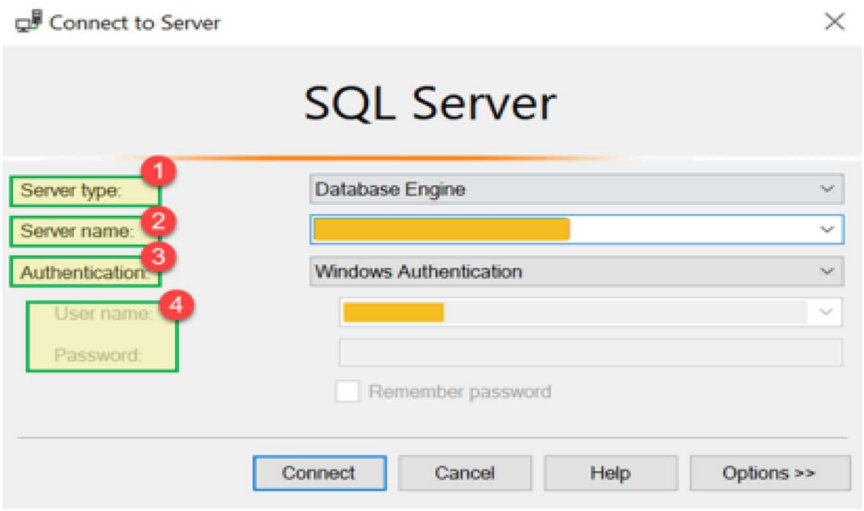
You can add menus, tool bars, status bars, text boxes, etc. to blank window.

Disadvantages of Visual Basic 6.0: -

The programs that are developed utilize more memory. Visual Basic requires specific operating system, which supports visual programming. Graphical User Interface is provided by visual basic, which takes some more time than other non-visual programming for execution of the program.

Introduction to SQL SERVER

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which mayrun either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single- machine applications to large Internet-facing applications with many concurrent users.. Storage space allocated to a database is divided into sequentially numbered pages, each8 KB in size. A page is the basic unit of I/O for SQL Server operations. A page is marked with a 96-byte header which stores metadata about the page including the pagenumber, page type, free space on the page and the ID of the object that owns it. The page type defines the data contained in the page. This data includes: data stored in the database, an index, an allocation map, which holds information about how pages are allocated to tables and indexes; and a change map which holds information about the changes made to other pages since last backup or logging, or contain large data types such as image or text. While a page is the basic unit of an I/O operation, space is actually managed in terms of an extent which consists of 8 pages. A database object can either span all 8 pages in an extent ("uniform extent") or share an extent with up to 7 more objects ("mixed extent"). A row in a database table cannot span more than onepage, so is limited to 8 KB in size.However, if the data exceeds 8 KB and the row contains varchar or varbinary data, the data in those columns are moved to a new page(or possibly a sequence of pages, called an allocation unit) and replaced with a pointerto the data. For physical storage of a table, its rows are divided into a series of partitions (numbered 1 to n). The partition size is user defined; by default all rows are in a singlepartition. A table is split into multiple partitions in order to spread a database over a computer cluster. Rows in each partition are stored in either B-tree or heap structure. The data is in the leaf node of the leaves, and other nodes storing the index values for the leaf data reachable from the respective nodes. If the index is non-clustered, the rowsare not sorted according to the index keys. An indexed view has the same storage structure as an indexed table. A table without a clustered index is stored in an unorderedheap structure. However, the table may have non-clustered indices to allow fast retrieval of rows.



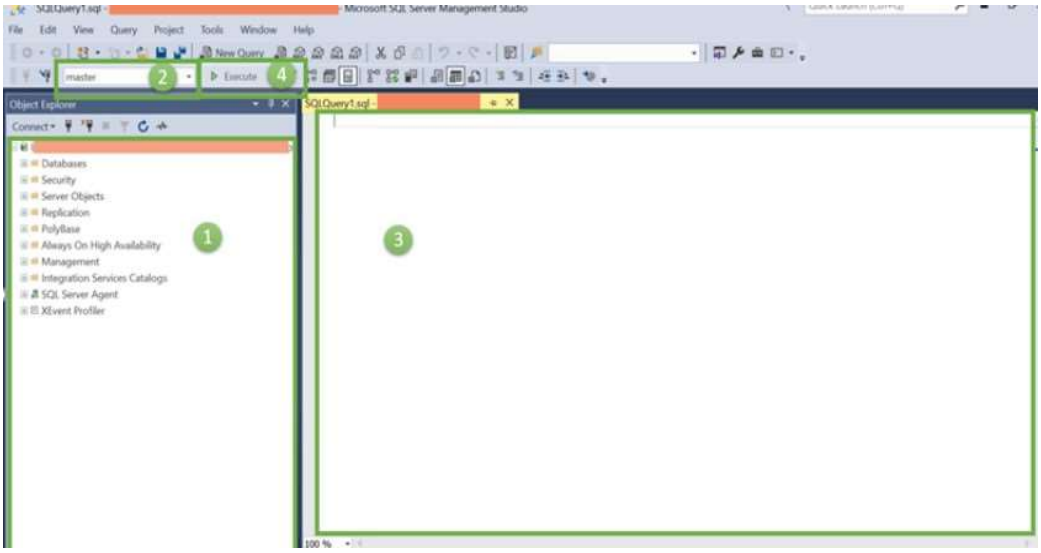
Let's understand each of the above fields :

Server type: This is an option to select one out of four available MS SQL services option. We will be working on 'Database Engine' for creating and working with Database. Other Server type includes Analysis, Reporting & Integration Services.

Server name: This is Server's name where MS SQL Server is installed and need to establish the connection with that server. Generally, we use the server name as "Machine name\Instance." Here Instance is the name given to SQL Server instance while SQL server installation

Authentication: This is defaulted to "Windows Authentication" if we use "Windows Authentication" during SQL Server Installation. Else, if we select 'Mixed Mode (Windows Authentication & Windows Authentication)' then Authentication will be defaulted to "SQL Server Installation.

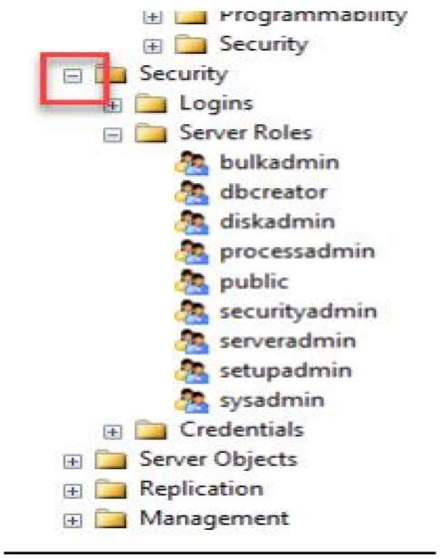
User name\Password: If Authentication is selected other than "Windows Authentication" like "SQL server Installation" then these two fields will be required. Click on 'Connect.' Now you will be connected to 'Data Management Studio.'



Let's discuss each section in detail :

1) **Object Explorer**

The Object Explorer provides a tree view of the database objects contained in the server. This section shows all the Databases, Security, Server Object for quick reference. To view the components of each object, just click the + icon located to the left of the object which will expand it.



2) **Databases Selection Dropdown**

This dropdown allows the user to select the Database in which we will be running ourqueries.

3) **Query Editor**

Here we can write all our queries. MS SQL server provides interactivesuggestions for tables, columns, etc. for easy queries creations and much more.

4) **Execute button**

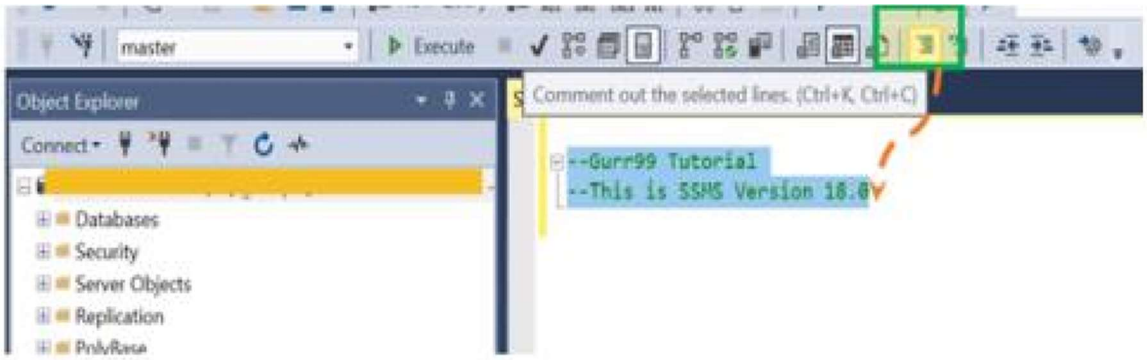
This button will finally execute the query and return the results.SSMS Tips and Issues.

• Management Studio is a standalone product. It does not correspond to any specificversion of SQL Server.

For example, we can use SMMS Version 18 with SQL Server 2017, SQL Server 2016as well.

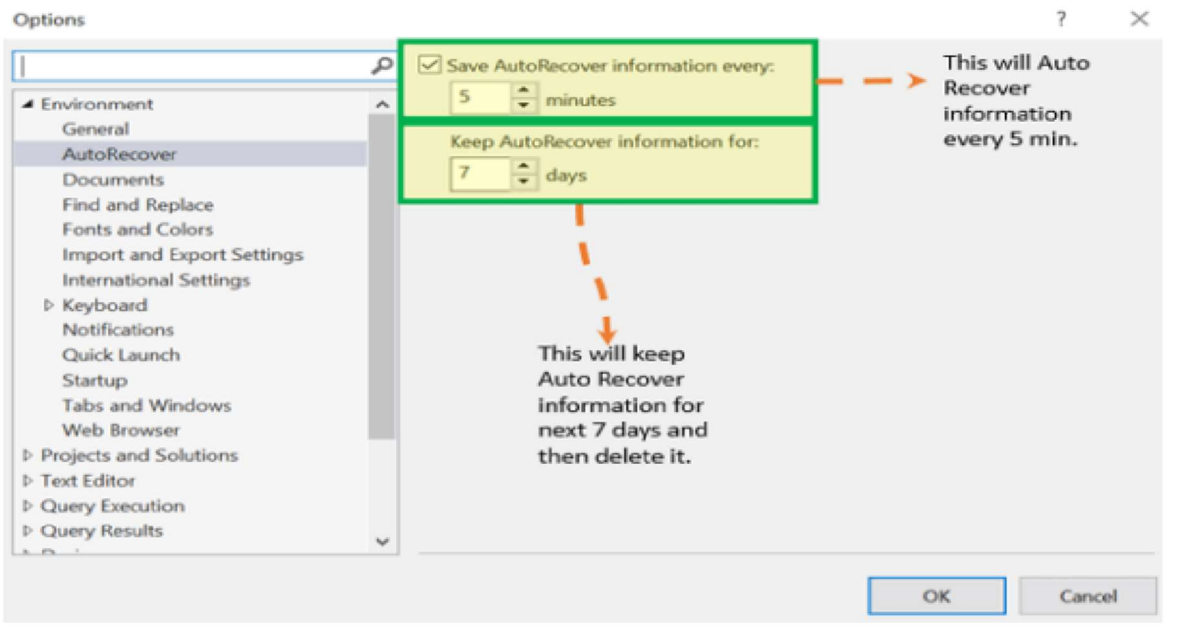
• Large codes reduce readability. Use comments for better Readability. Put "--" in frontof any line to comment it out.

• Group comment: We can comment out the group of line by selecting them all andclicking on Icon shown in below image.



• Unexpected shutdown, a system failure can cause unexpected data loss. Set 'Auto recover' option checked to minimize data loss. We can even customized time interval to Auto Recover data and the number of days this information to be saved before deleting it.

• Tools> Options>Environment>AutoRecover



In case of failure, the popup window will appear with the name • We often need theresult of our query to be saved in text format for future references.

SYSTEM DESIGN

SYSTEM DESIGN

INTRODUCTION TO SYSTEM DESIGN:

The design phase is the life cycle phase in which the detailed design of the selected system in the study phase is accomplished. In the design phase, the technical specifications are prepared for the performance of all allocated tasks. It also includes the construction of programs and program testing. In the design phase, the first step isto determine the output is to be produced and in what format. Second, input data and master files have to be designed to meet the requirements of proposed output. The system analyst has to define the methods of capturing and input programs and format of the output and its use by the users.

SYSTEM FLOW CHART:

A graphic representation of a system showing the overall flow of control in the processing at the job level; specifies what activities must be done to convert from a physical to logical model is known as a system flowchart. Thus, it summarizes what operations are undertaken and where and when they take place. Normally in a system flowchart input from outside are shown to the left and outputs to the right. Symbols representing the operations undertaken and the documents used are then placed in the appropriate places which gives a general flow of data from top to bottom and left to right. Arrows are used on the connecting lines to indicate the logical flow or sequencewhere the flow is not in the standard direction. No interaction is implied by crossing lines. Decisions which lead to different actions can also be shown

DATA FLOW DIAGRAM:

A data flow diagram is graphic representation of a system that shows data flows to, from and within the system, processing functions that change the date in some manner, and the storage of this data. They are networks of related system function that indicatedform where information is revived and to where it is sent. An external entity is the originator or receiver of data or information. . A data store symbol portrait a file or database in which data resides. A process is depicted by a circle sometimes it is calleda bubble or transform. Process portraits the transformation of the content of status of data.

DATABASE DESIGN:

This activity deals with the design of the physical database. The designer begins toconcentrate on file design or how data should be organized around user requirements.How data are organized depends on the data and response requirements that determinehardware configurations. An integrated approach to file design is the database. Thegeneral theme is to handle information as an integrated whole, with a minimum ofredundancy and improved performance, type and size of data structure used. Theobjectives of data base are accuracy and integrity, privacy and security of data etc The general theme behind a database is to handle information as an integrated whole.A database is a collection of interrelated data stored with minimum redundancy to servemany users quickly and effectively. After designing input and output, the analyst mustconcentrate on database design or how data should be organized around userrequirements. The general objective is to make information access, easy quick,inexpensive and flexible for other users.

CODE DESIGN:

Codes can provide brief identification of data items and replace longer descriptions that would be more awkward to store and to manipulate.

INPUT DESIGN: -

Input design is processing of converting the user-oriented description of the inputs of the system. The goal of designing input data to make data entry as easy logical and freefrom errors as possible.

In entering data, operates need to know the following:

1. The allocated space for each field.
2. Field sequence which must match that in the source document.
3. The format in which data fields are entered for example, filling out the data field is required through the edited formatmm/dd/ yy. When we approach input data design, we design source documents that capturethe data and then selected the media use to enter them into the computer.

   There are different ways in which data can be introduced into the system such as :

   a. The data is converted into a machine sensible from by some realistic source document and types in the relevant items using a keyboard connected to the system.

b. The document can be read directly by a machine and this converts informationheld in the

human sensible form into a machine-readable form without need forhuman investigations.

c. Data entered into a system through a keyboard. This is done interactively by the person using the system. The field name must be documented. The field name must beknown to data entry operator or users so that the data entry will not exceed the allocatedspace. Our system the contains the following inputs

OUTPUT DESIGN:

Primary consideration in the design of all output is the information requirement and other objective of the users. It is the most important and direct source of information to the user. A major form of output is a hard copy. Print out should be designed aroundthe output requirements of the user. Each output should be given a specific name or title. The output data is displayed on the visual display unit and output can be redirected to printers and or sorted in a file for later use. Here, in this system, program is designed so as to generate a number of relevant outputs displayed in various kinds of user- defined tables in an easily readable and comprehensive manner which can be readily read and understood by the user. So, no further attempt has been made to generate reports which of course could have been easily implemented into the system.

PROCEDURE DESIGN:

When program become very long, they are divided into smaller programs or modules.These smaller programs can be written, tested and debugged separately. This techniqueof programming is known as modular programming.

The advantages of modular programming are:

1. It is easy to write, test and debug a module.

2. Generally the modules of common nature are prepared, which can be used at manyplaces.

3. The programmer can use the previously written programs. 4. If a change is to bemade, it is made in the particular module; the entire program is not affected.

SYSTEM IMPLEMENTATION

SYSTEM IMPLEMENTATION

PROGRAM DEVELOPMENT

In the case of program development first of all the problem is defined. It includes input-output specifications, requirements, execution times, accuracy etc. A necessary system flowchart is expended to show additional detail input and out files are identified, and computer programs logic flowchart are prepared for each computer program component. An algorithm can also write to solve the problem.

The following are the stages for the development of software:

1. Problem definition
2. Program design
3. Coding
4. Debugging
5. Testing
6. Documentation
7. Maintenance, Extension, and Redesign.

The criteria for evolution of a program are reliability, speed hardware cost, programming time and cost of use error tolerance and extensibility. A good program should utilize memory and times efficiently. An interface should be simple and less costly as far as possible to perform a ascertain task. Good design and clear documentation make a program simple and it can be used by others.

SOFTWARE SELECTION

Software selection is critical aspect of system development. These are two ways of acquiring software custom-made or "off-the-shelf" packages. Today's trend towards purchasing packages, which represent roughly 10 percent of what are costs to, developed same in house. Prior to selecting the software, the project team must setup criteria for software selection. Software readability brings up the concept of modularity. The criterion, usability refers to the effort required to the operate, prepare

the input and interpret the output program. Serviceability focuses on documentation and vendor support lost is major consideration. The other criteria are flexibility, security, performance, and ownership.

SECURITY FEATURES

Every candidate system must provide built in for security and integrity of data. Withoutsafeguards against unauthorized access and natural disasters, a system could be so vulnerable as to threaten survival of the organization. To do an adequate job on security,a system analyst must analyse the risks, exposure and costs and specify measures suchas passwords and encryption to provide protection. In addition, backup copies of software and recovery re start procedures must be available when needed.

Table Sheme

1. ADMIN Table

	Column Name	Data Type
🔑	Admin_Id	varchar(50)
	Name	varchar(100)
	Password	varchar(50)
	Mobile_No	numeric(10, 0)
	Address	varchar(250)
	Photo	varchar(150)

2. STUDENT Table

	Column Name	Data Type
🔑	Student_Id	varchar(50)
	Name	varchar(100)
	Photo	varchar(150)
	Gender	varchar(15)
	Course	varchar(10)
	Class	varchar(10)
	Mobile_No	numeric(10, 0)
	Address	varchar(250)

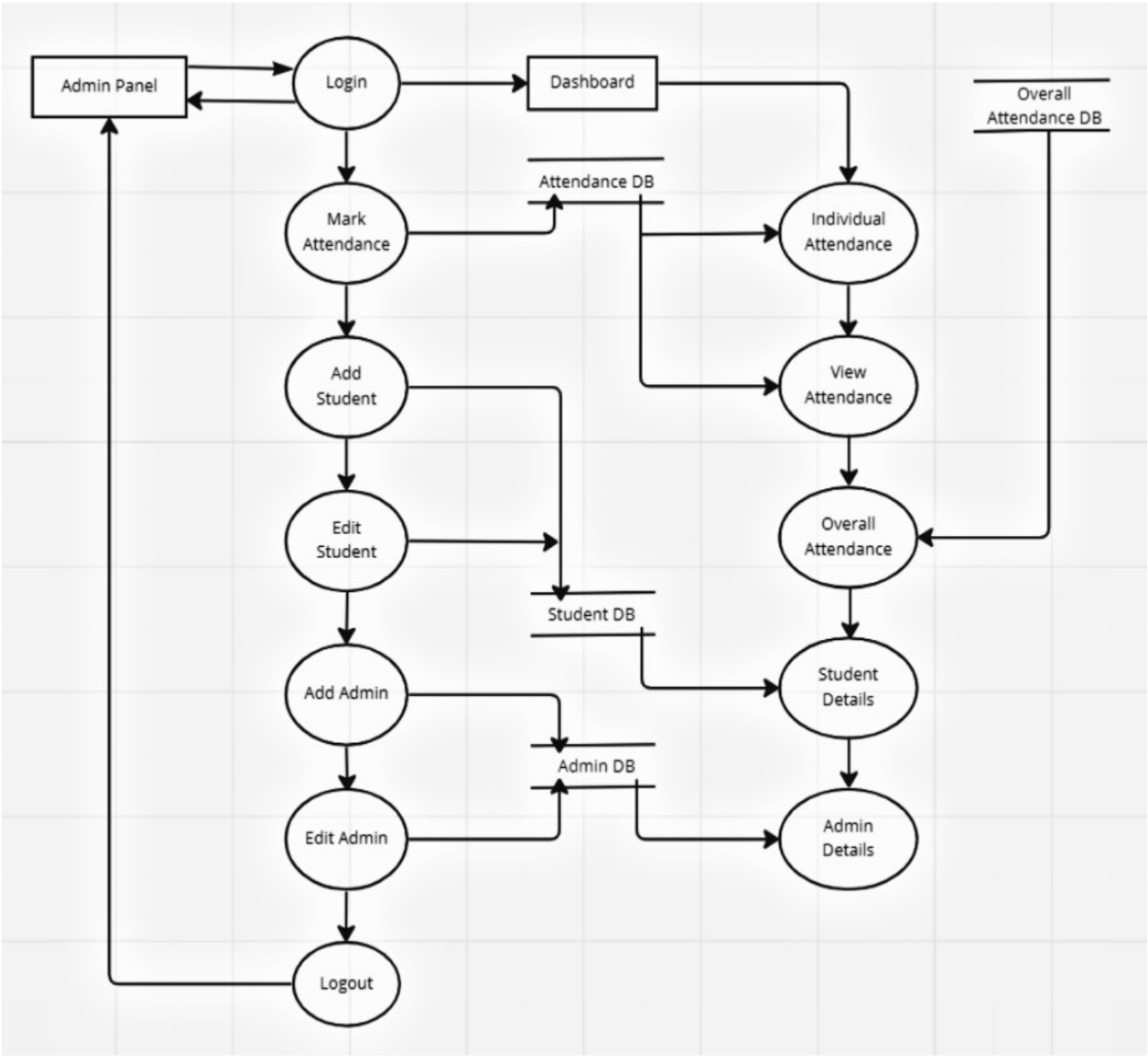
3. ATTENDANCE Table

	Column Name	Data Type
FK	Student_Id	varchar(50)
	Date	varchar(10)
	Attendance	varchar(10)

4. OVERALL ATTENDANCE Table

	Column Name	Data Type
FK	Student_Id	varchar(50)
	Name	varchar(100)
	Course	varchar(10)
	Class	varchar(10)
	Total_Classes	numeric(3, 0)
	Classes_Attended	numeric(3, 0)
	Percentage	varchar(10)

Data Flow Diagram (DFD)

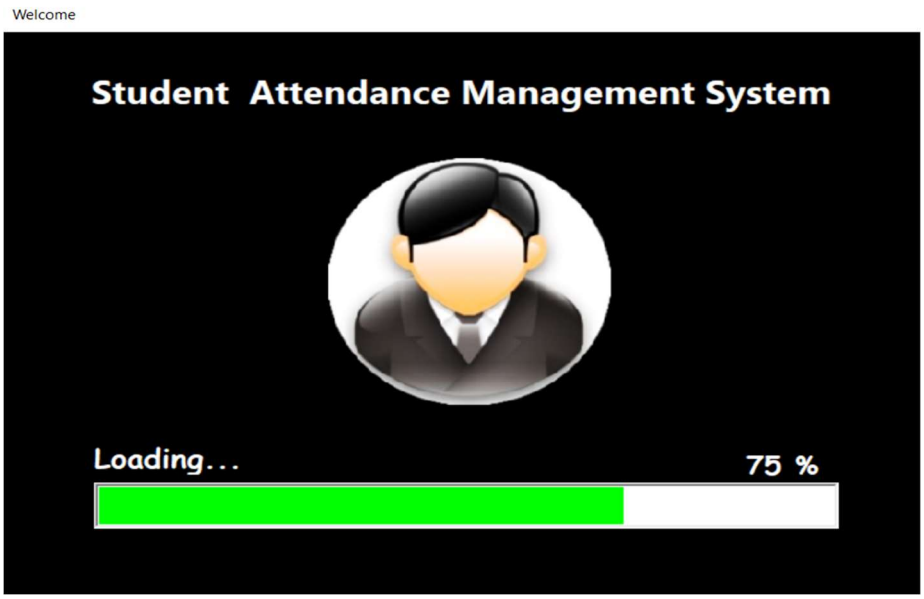


SCREEN SHOTS

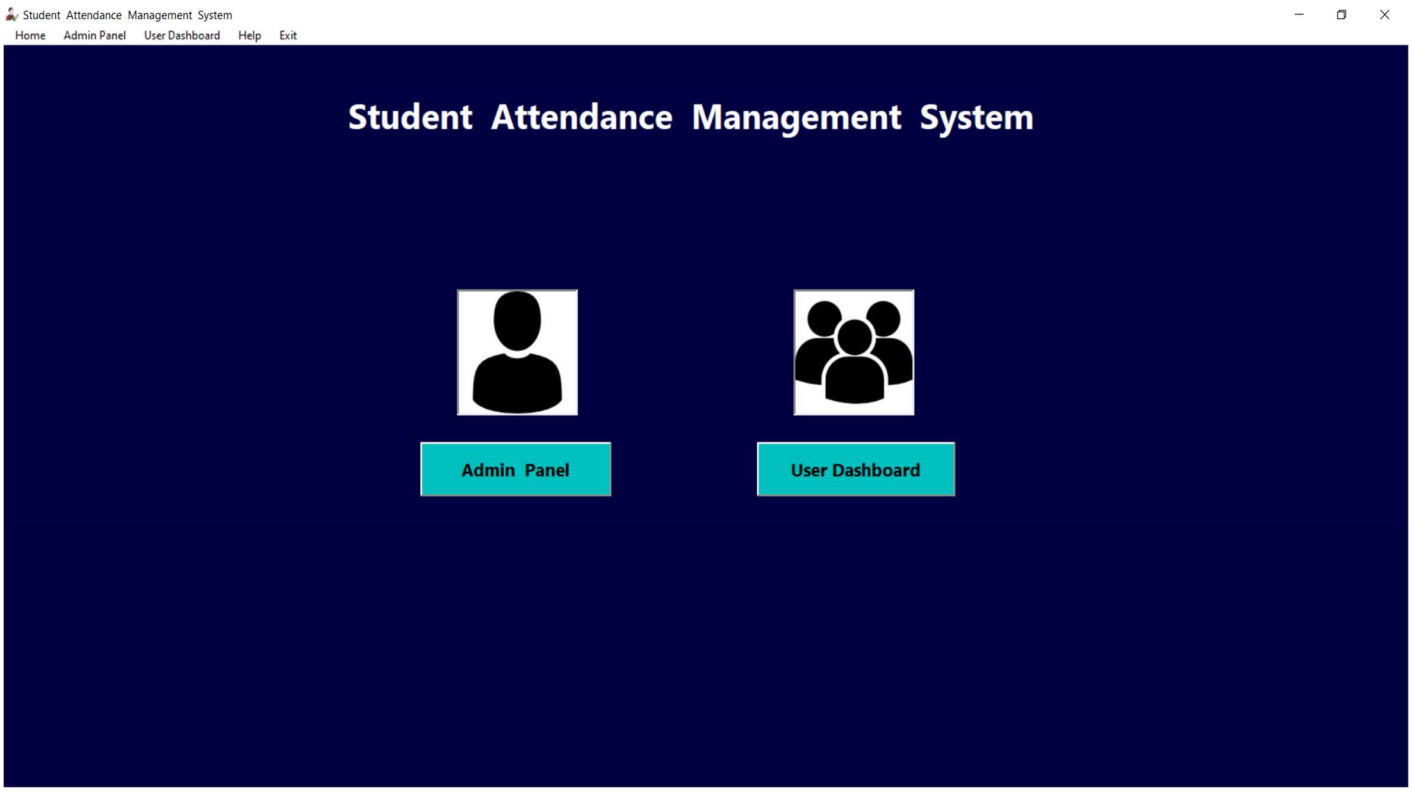
SCREENSHOTS

1) APPLICATION ICON

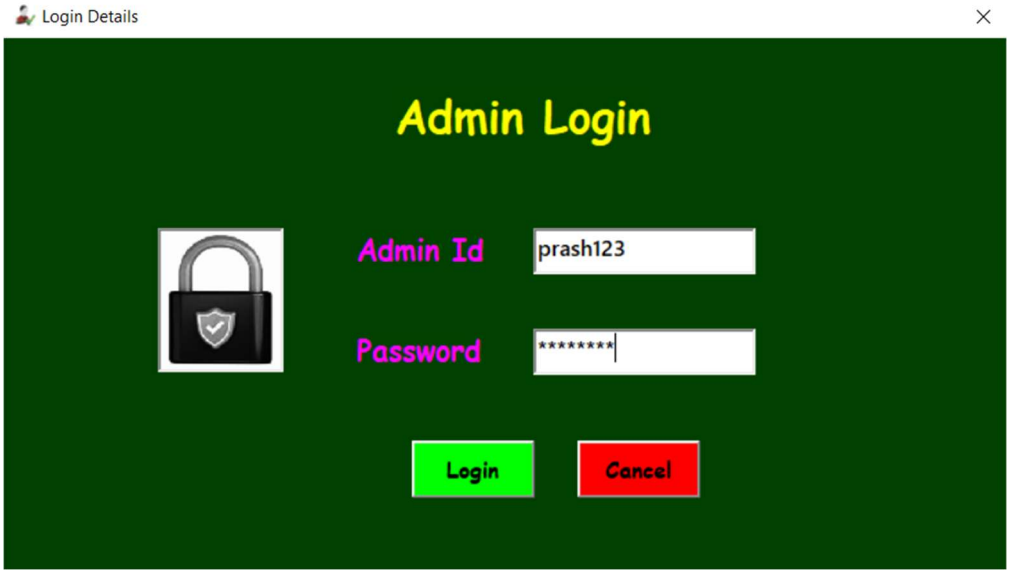
2) WELCOME SCREEN



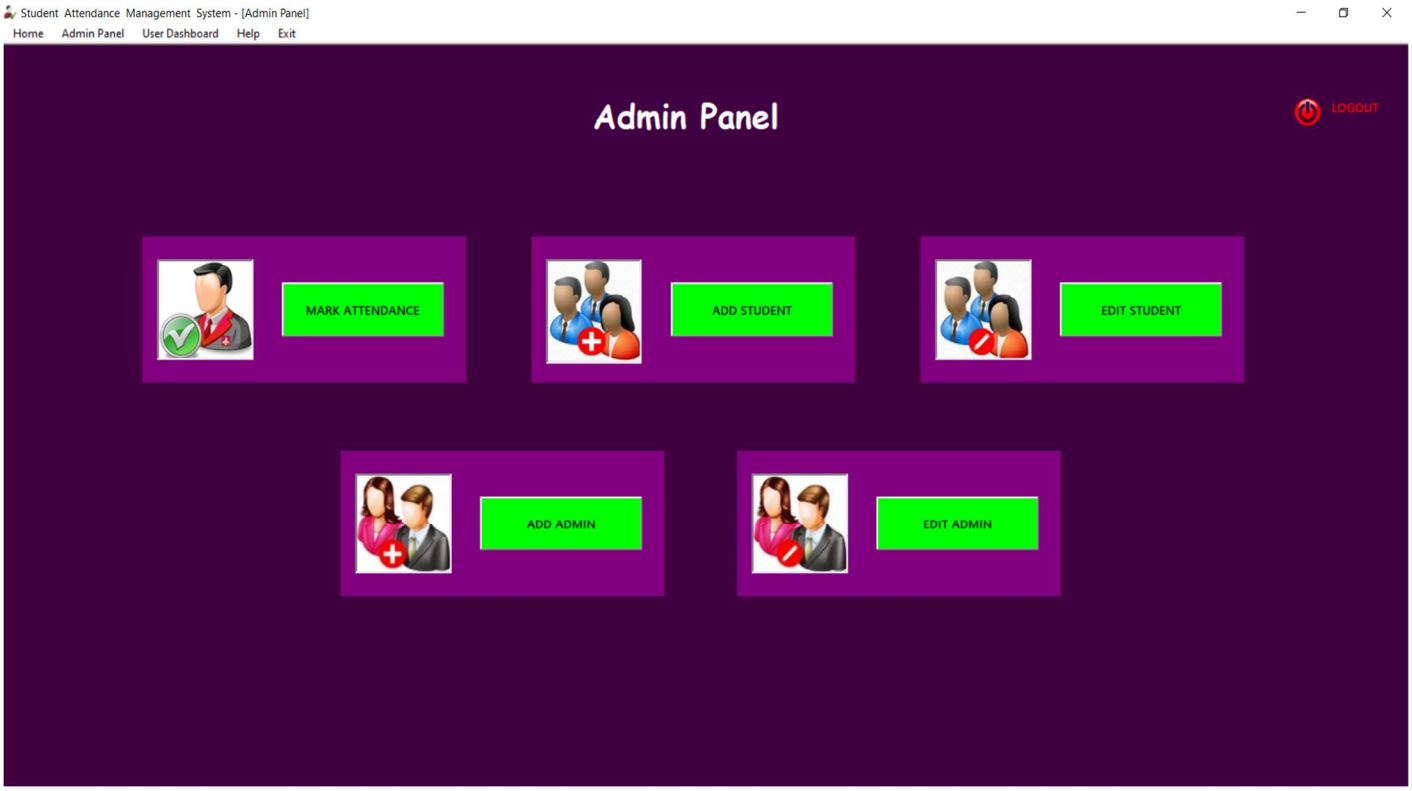
3) HOME PAGE



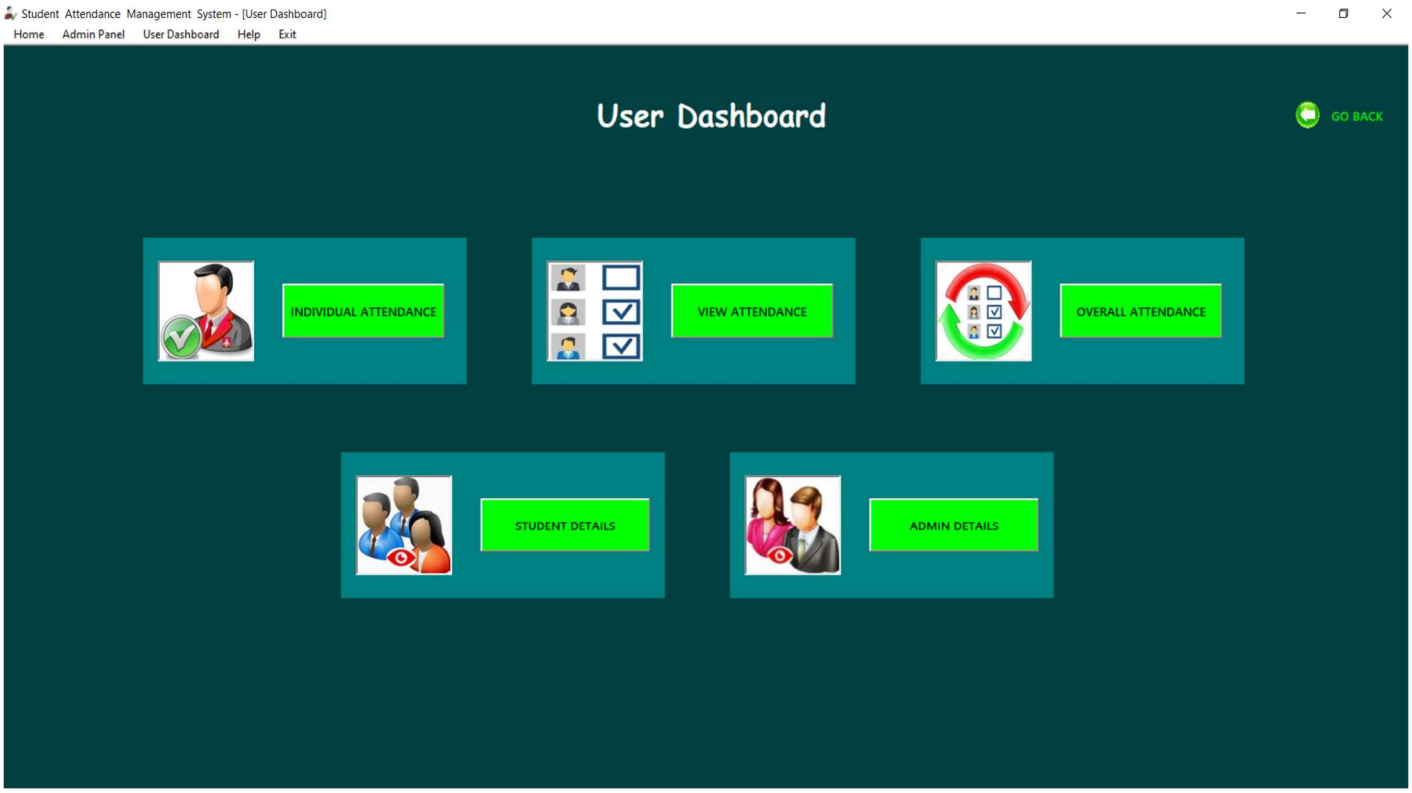
4) ADMIN LOGIN



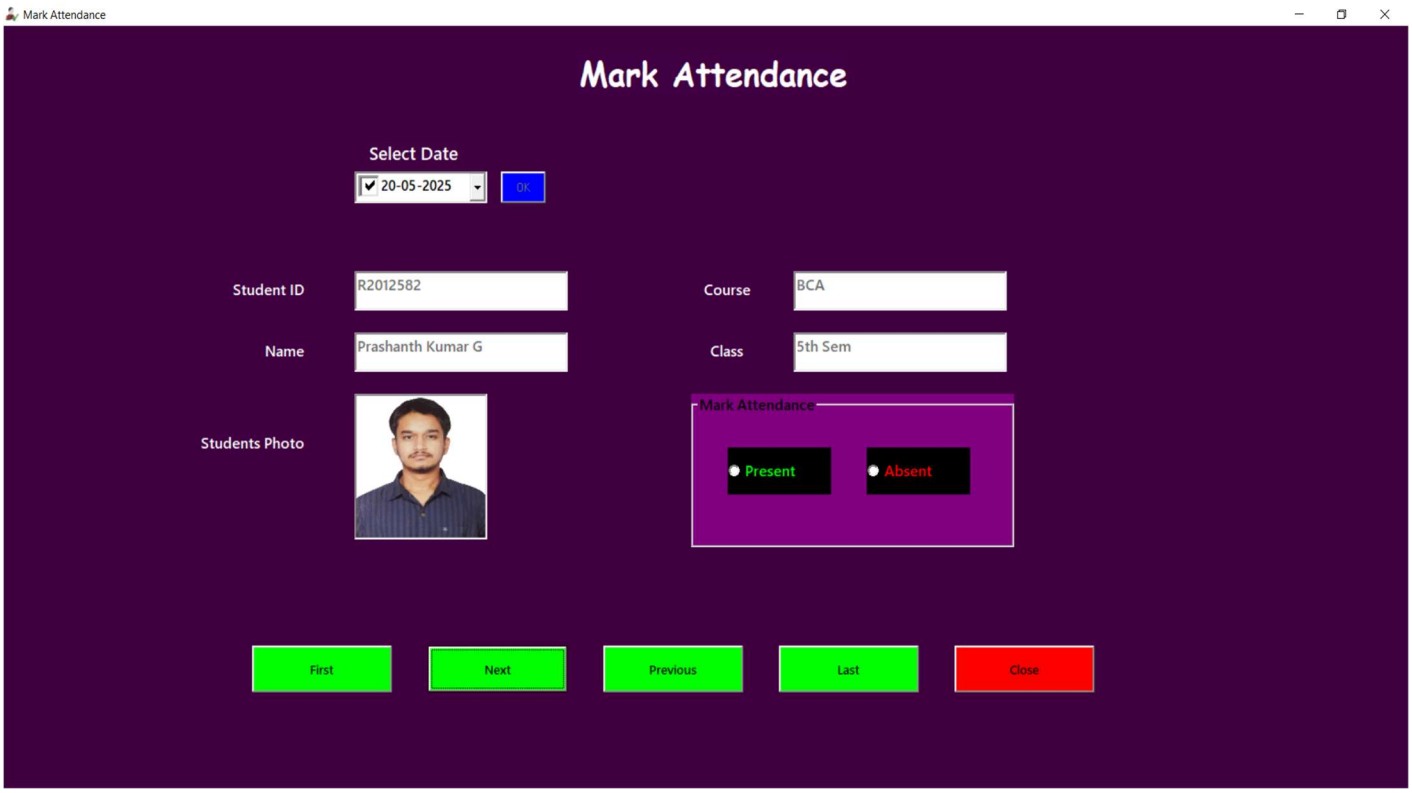
5) ADMIN PANEL



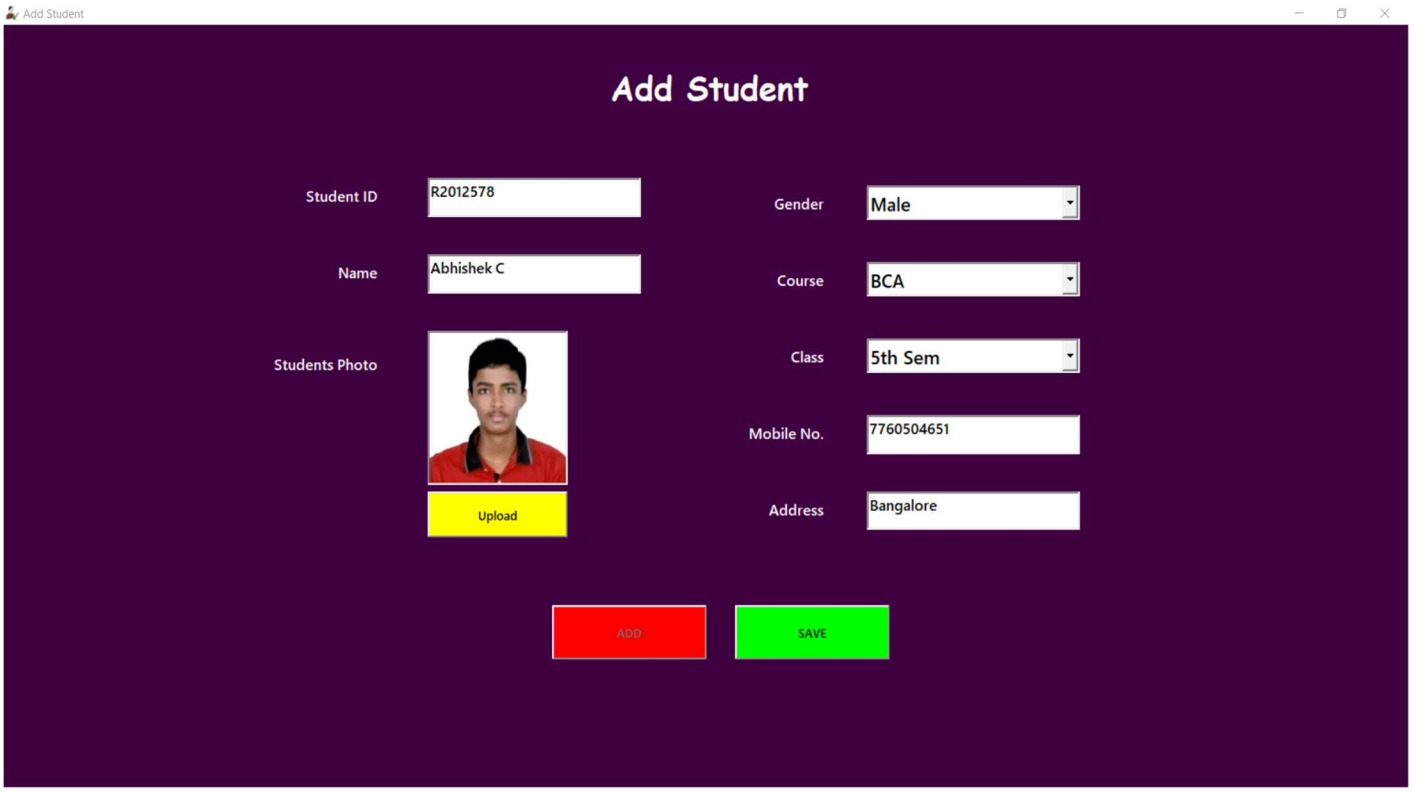
6) USER DASHBOARD



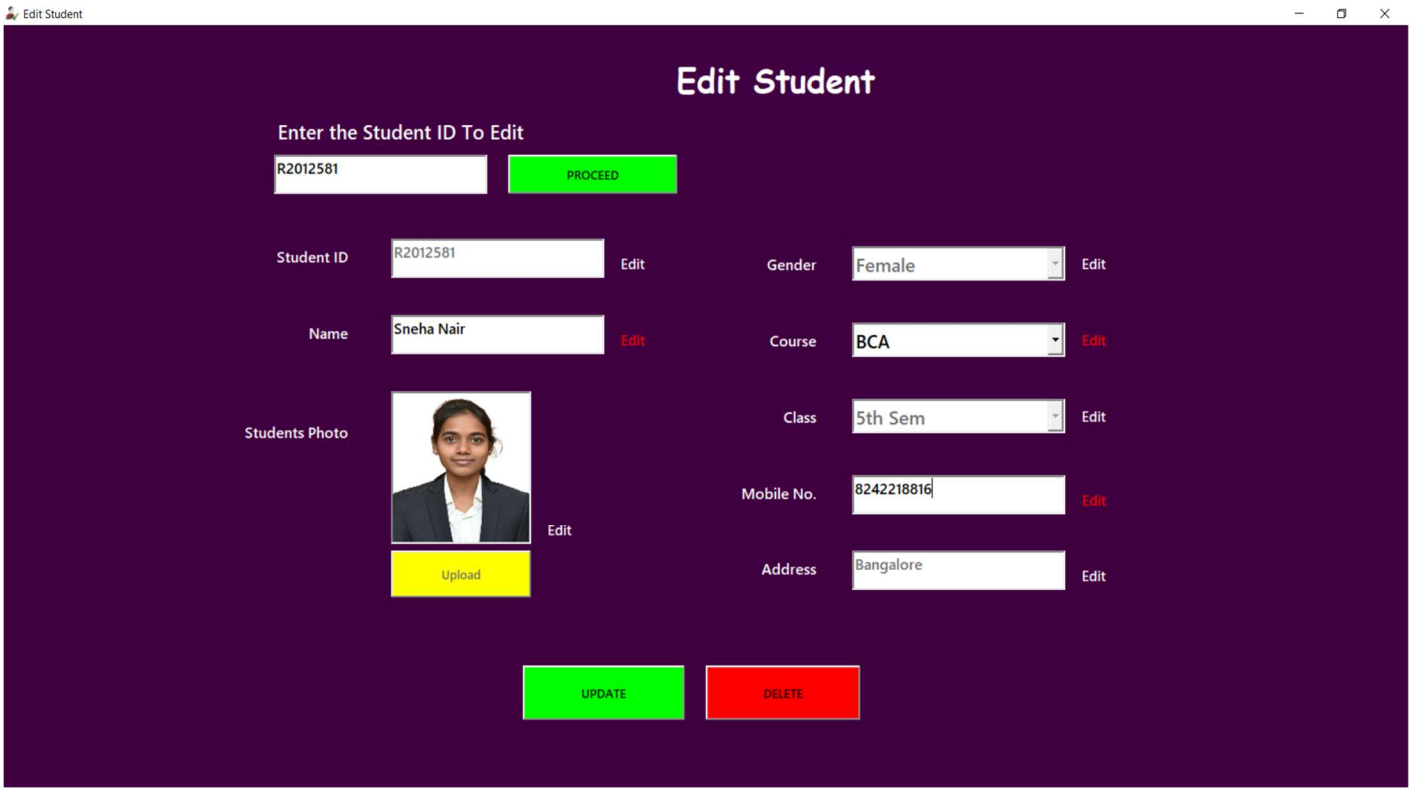
7) MARK ATTENDANCE



8) ADD STUDENT



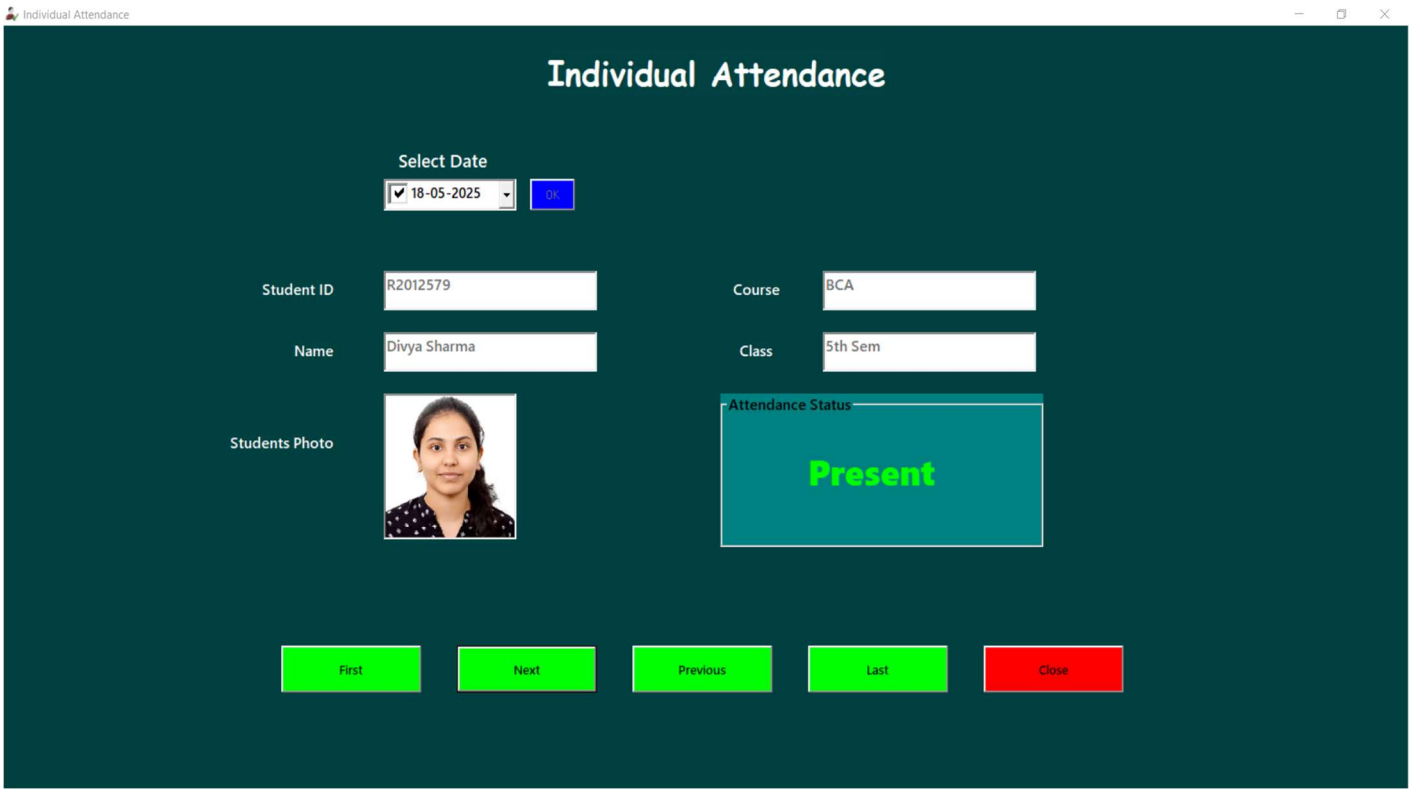
9) EDIT STUDENT



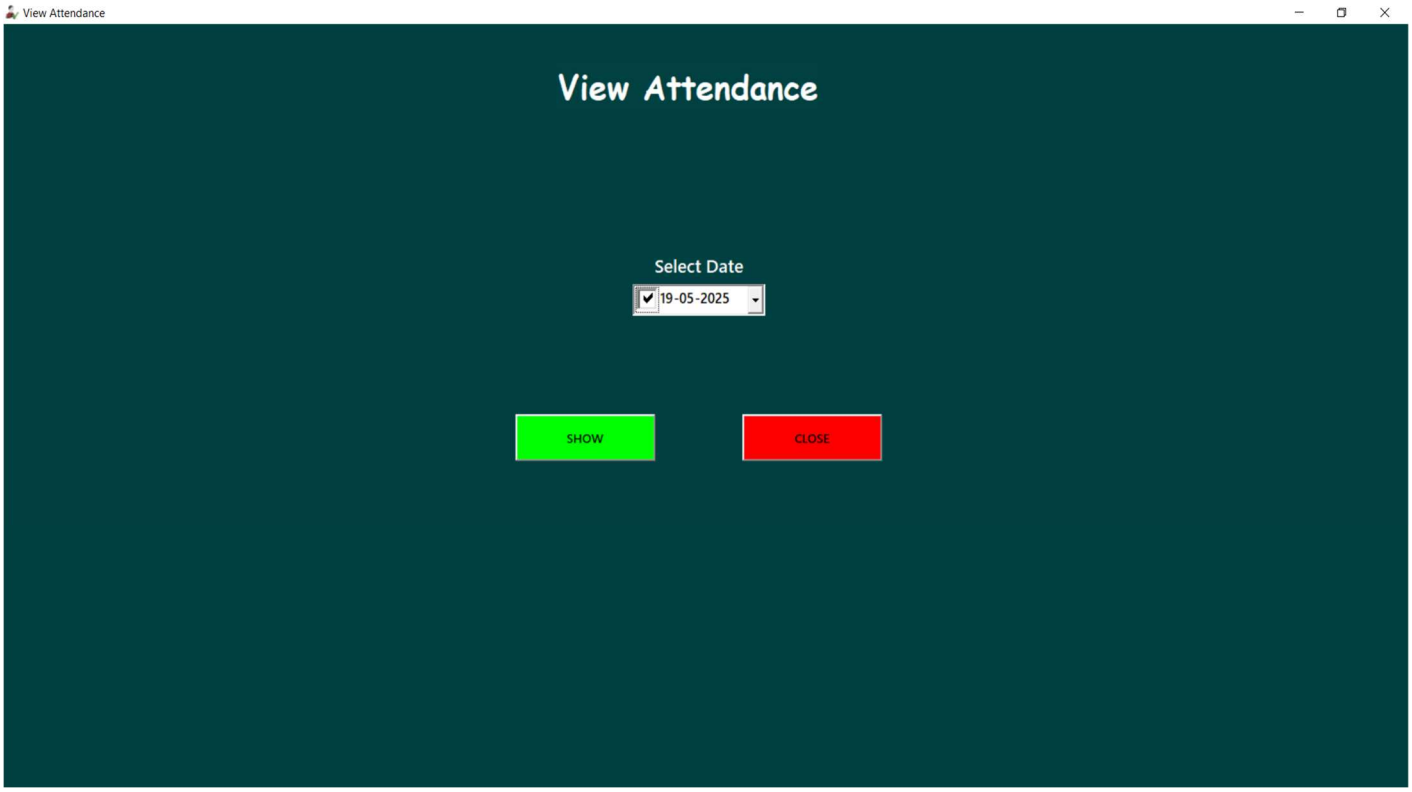
10) ADD ADMIN

11) EDIT ADMIN

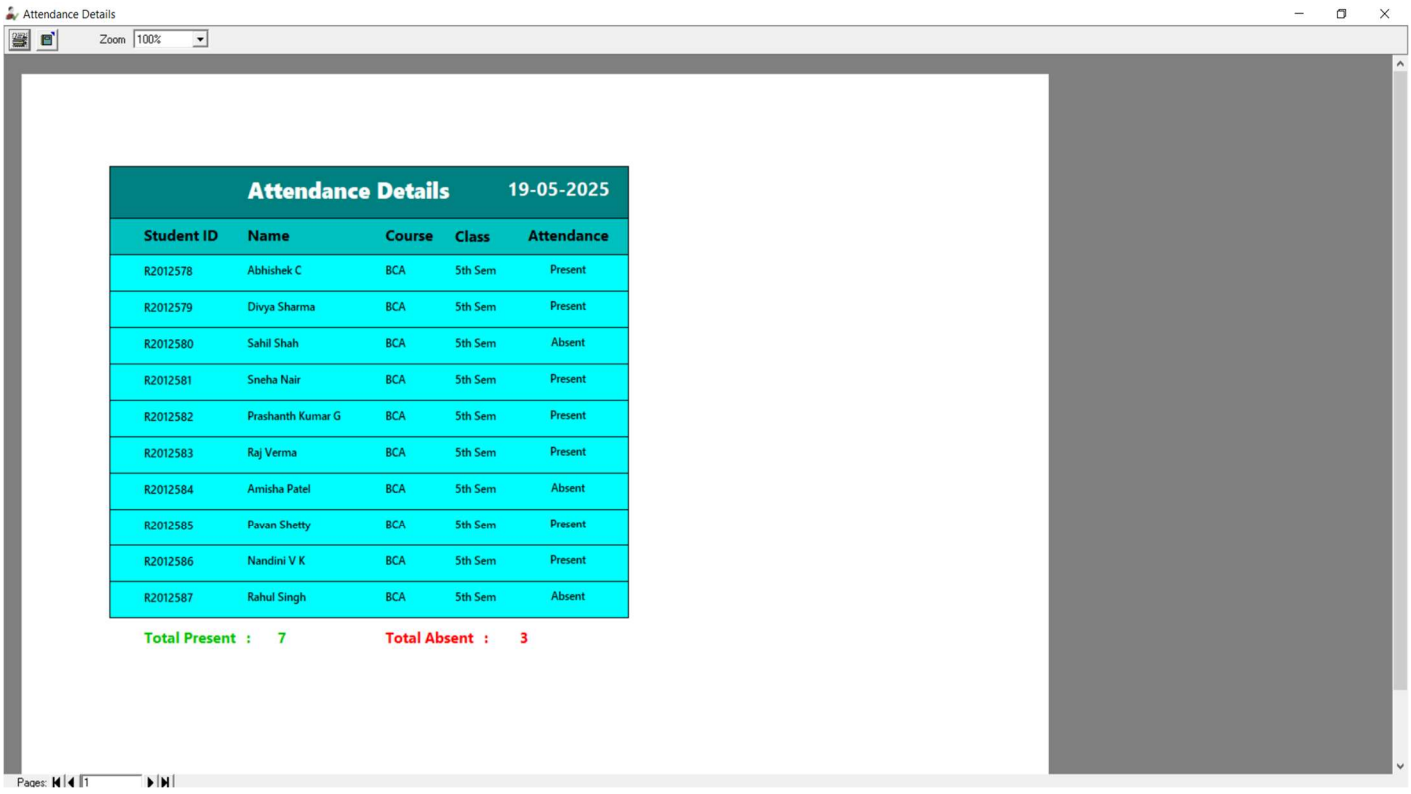
12) INDIVIDUAL ATTENDANCE



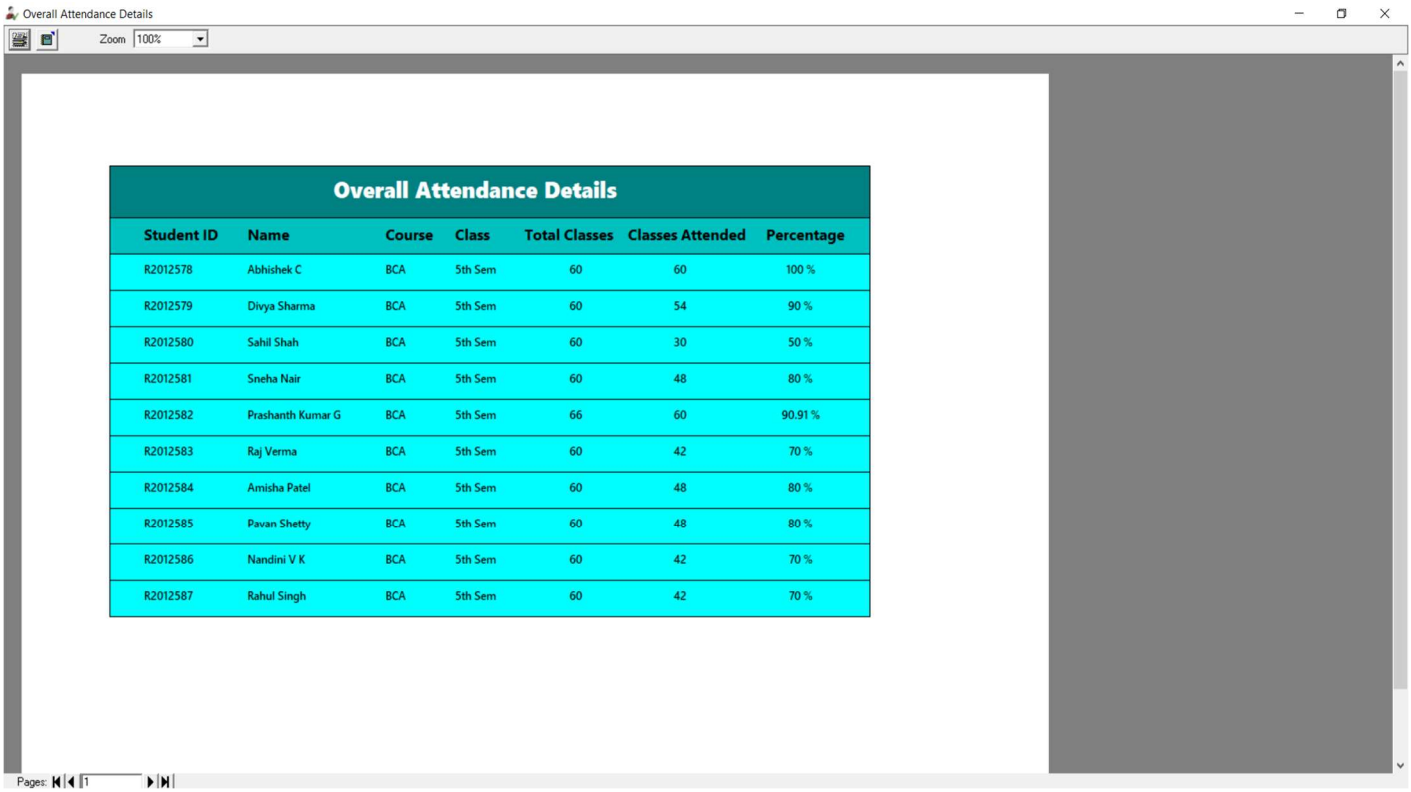
13) VIEW ATTENDANCE



14) ATTENDANCE DETAILS



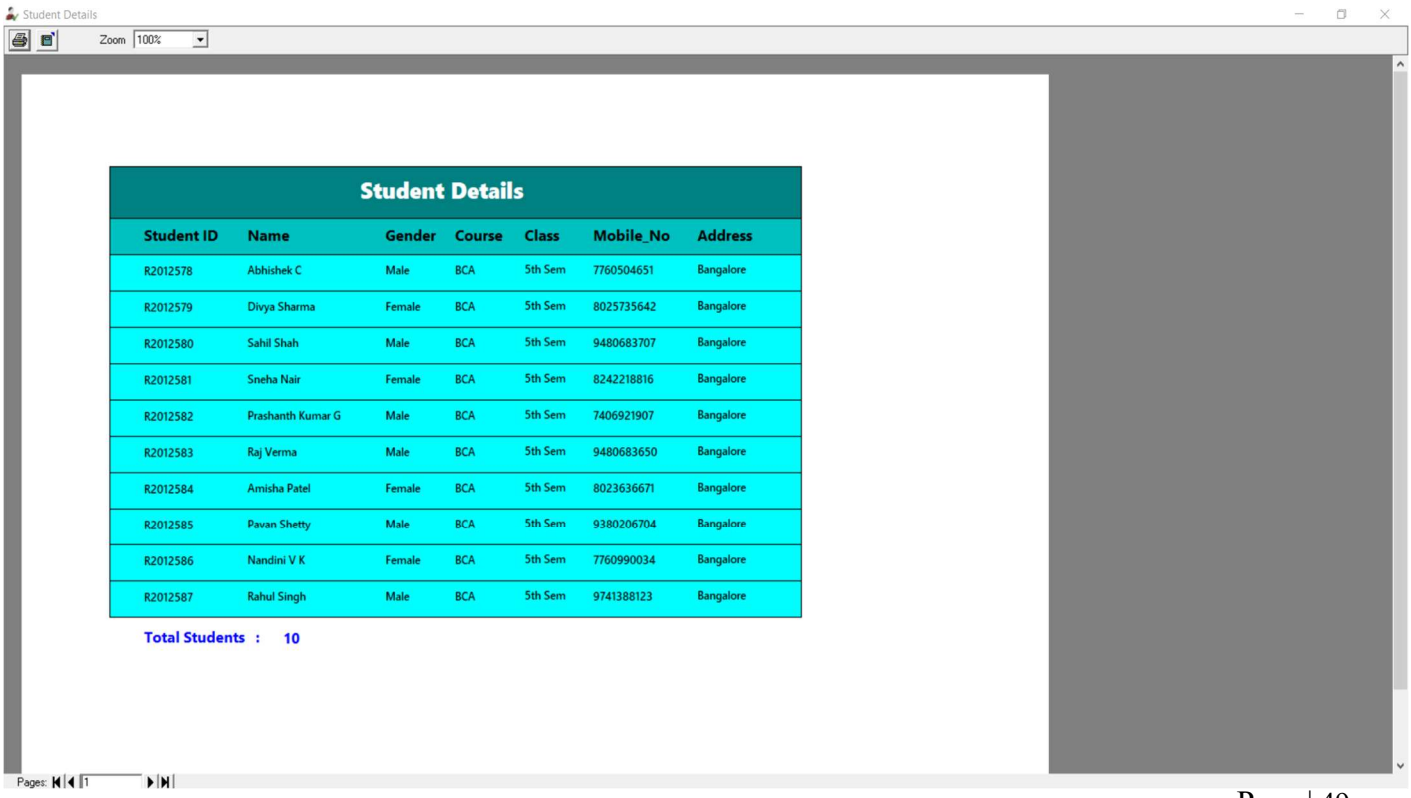
15) OVERALL ATTENDANCE DETAILS



Overall Attendance Details

Student ID	Name	Course	Class	Total Classes	Classes Attended	Percentage
R2012578	Abhishek C	BCA	5th Sem	60	60	100 %
R2012579	Divya Sharma	BCA	5th Sem	60	54	90 %
R2012580	Sahil Shah	BCA	5th Sem	60	30	50 %
R2012581	Sneha Nair	BCA	5th Sem	60	48	80 %
R2012582	Prashanth Kumar G	BCA	5th Sem	66	60	90.91 %
R2012583	Raj Verma	BCA	5th Sem	60	42	70 %
R2012584	Amisha Patel	BCA	5th Sem	60	48	80 %
R2012585	Pavan Shetty	BCA	5th Sem	60	48	80 %
R2012586	Nandini V K	BCA	5th Sem	60	42	70 %
R2012587	Rahul Singh	BCA	5th Sem	60	42	70 %

16) STUDENT DETAILS

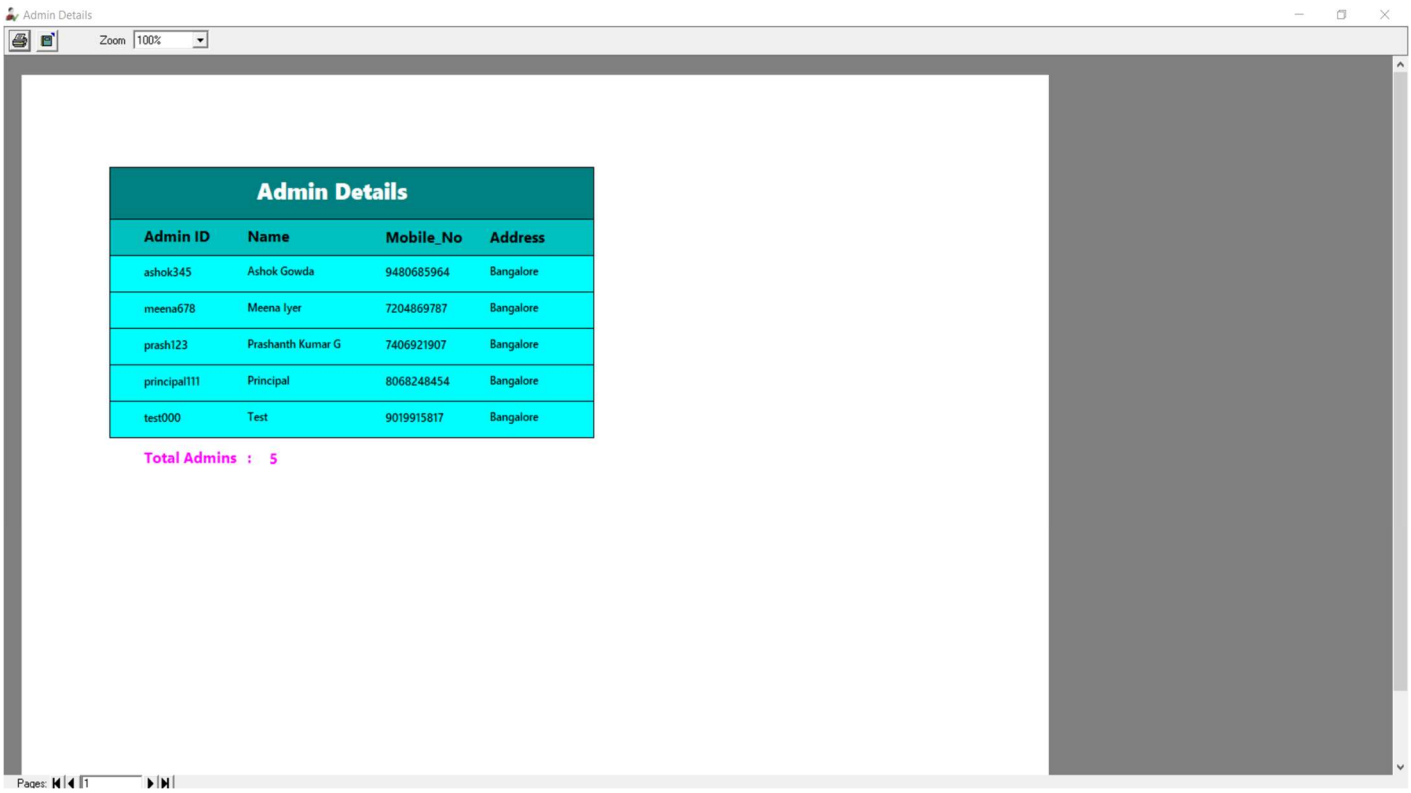


Student Details

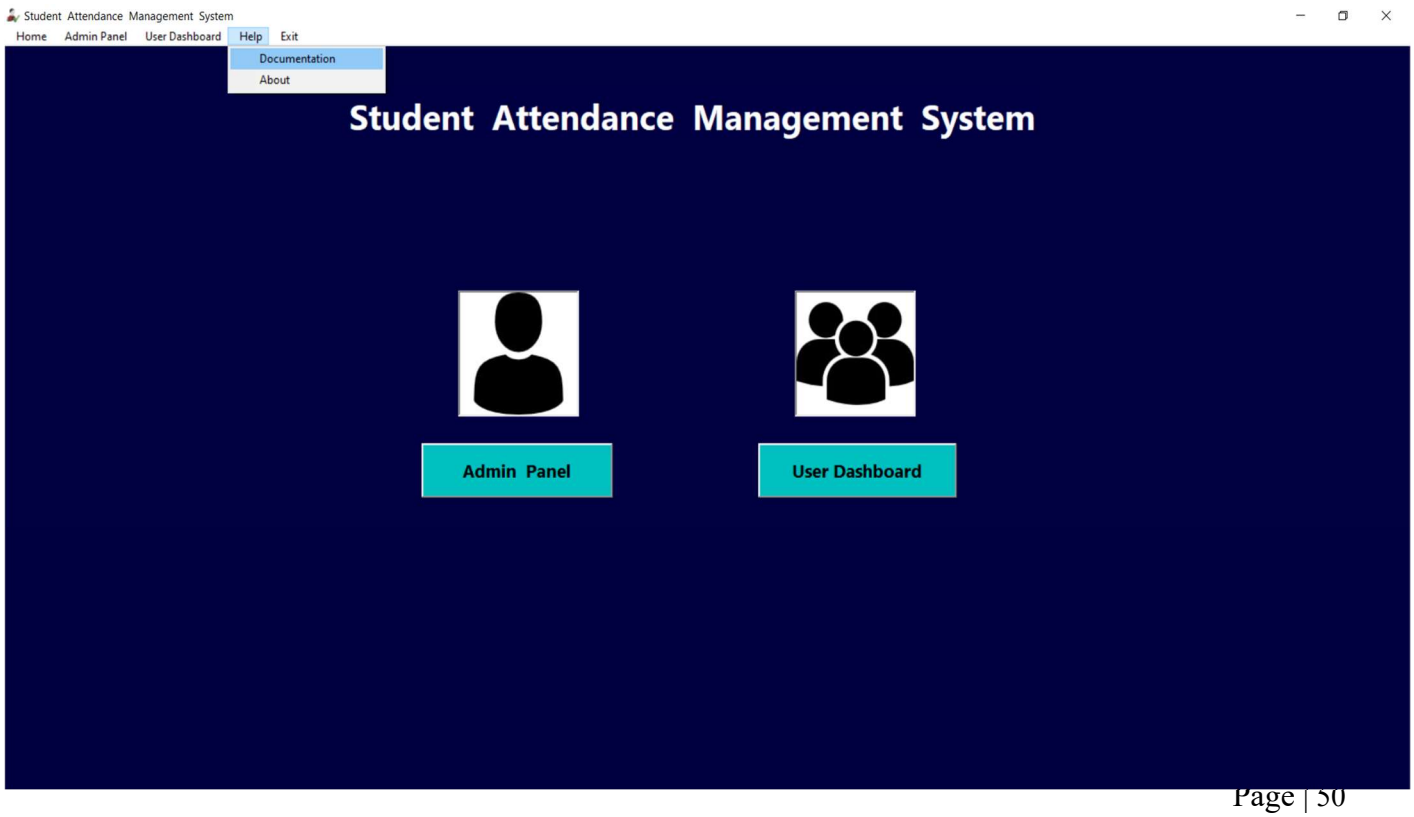
Student ID	Name	Gender	Course	Class	Mobile_No	Address
R2012578	Abhishek C	Male	BCA	5th Sem	7760504651	Bangalore
R2012579	Divya Sharma	Female	BCA	5th Sem	8025735642	Bangalore
R2012580	Sahil Shah	Male	BCA	5th Sem	9480683707	Bangalore
R2012581	Sneha Nair	Female	BCA	5th Sem	8242218816	Bangalore
R2012582	Prashanth Kumar G	Male	BCA	5th Sem	7406921907	Bangalore
R2012583	Raj Verma	Male	BCA	5th Sem	9480683650	Bangalore
R2012584	Amisha Patel	Female	BCA	5th Sem	8023636671	Bangalore
R2012585	Pavan Shetty	Male	BCA	5th Sem	9380206704	Bangalore
R2012586	Nandini V K	Female	BCA	5th Sem	7760990034	Bangalore
R2012587	Rahul Singh	Male	BCA	5th Sem	9741388123	Bangalore

Total Students : 10

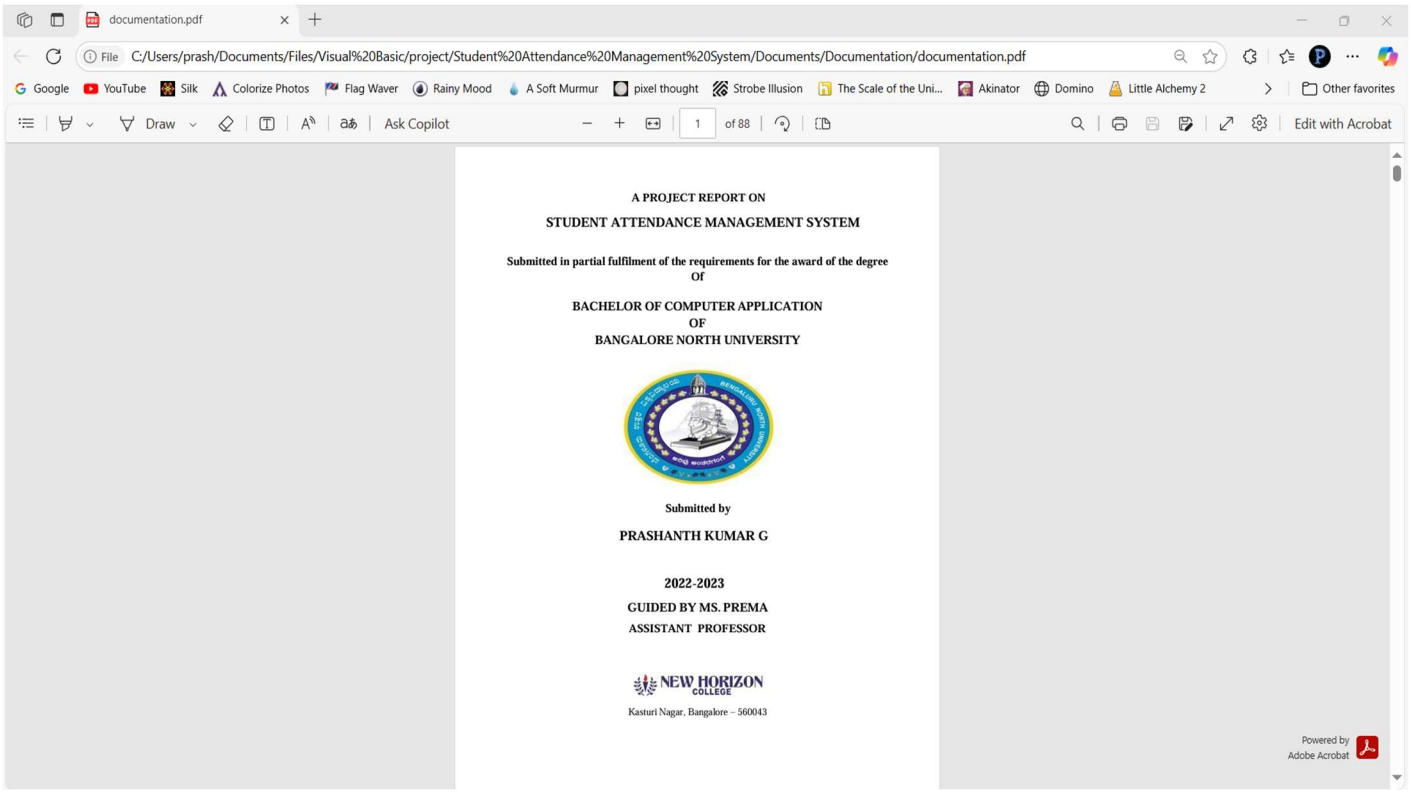
17) ADMIN DETAILS



18) HELP MENU



19) DOCUMENTATION



documentation.pdf

C:/Users/prash/Documents/Files/Visual%20Basic/project/Student%20Attendance%20Management%20System/Documents/Documentation/documentation.pdf

A PROJECT REPORT ON

STUDENT ATTENDANCE MANAGEMENT SYSTEM

Submitted in partial fulfilment of the requirements for the award of the degree
Of

BACHELOR OF COMPUTER APPLICATION
OF
BANGALORE NORTH UNIVERSITY

Submitted by

PRASHANTH KUMAR G

2022-2023

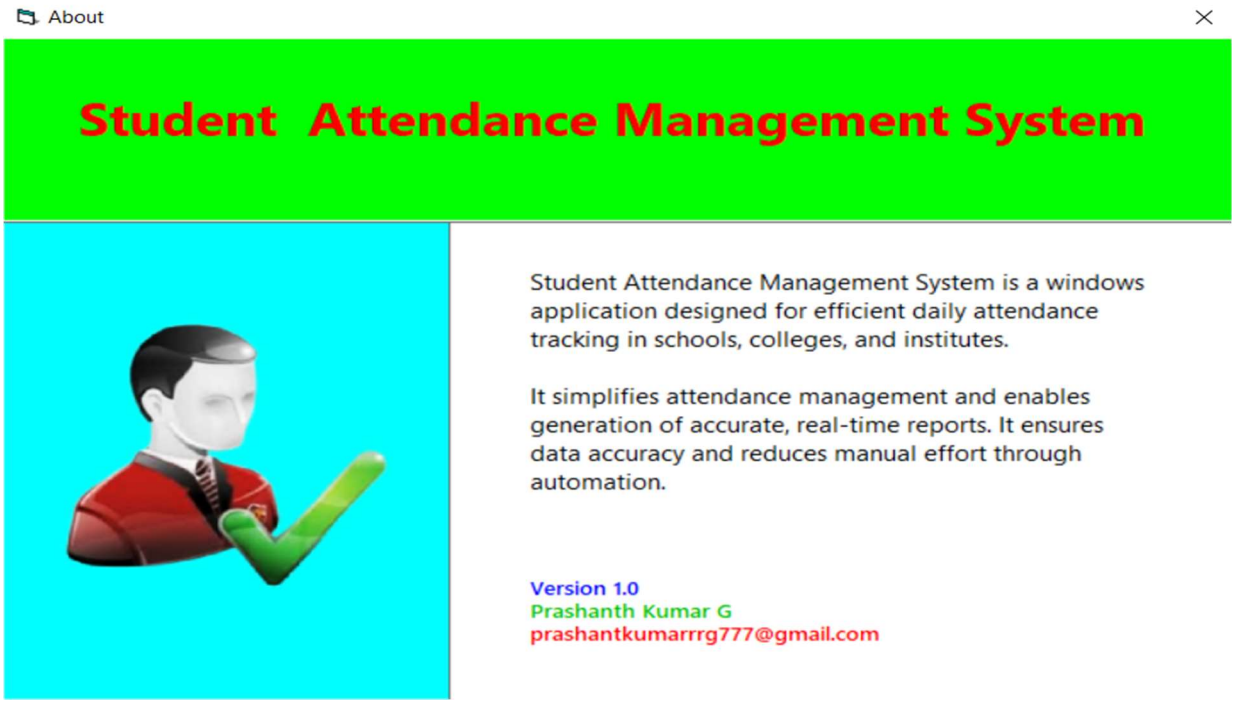
GUIDED BY MS. PREMA
ASSISTANT PROFESSOR

NEW HORIZON COLLEGE

Kasturi Nagar, Bangalore – 560043

Powered by Adobe Acrobat

20) ABOUT



CODING

CODING

WELCOME SCREEN

```
Option Explicit

Private Sub Timer1_Timer()
On Error Resume Next
PBcolor ProgressBar1, vbWhite, vbGreen
If ProgressBar1.Value >= 0 Then
ProgressBar1.Value = ProgressBar1.Value + 5
Label3.Caption = ProgressBar1.Value & " %"
End If
If ProgressBar1.Value > "100" Then
mdiMainHome.Show
Unload Me
End If
End Sub
```

HOME PAGE

```
Option Explicit

Private Sub about_Click()
frmAbout.Show
End Sub

Private Sub admin_Click(Index As Integer)
mdiMainHome.Enabled = False
frmAdminLogin.Show
End Sub

Private Sub Command1_Click()
mdiMainHome.Enabled = False
frmAdminLogin.Show
End Sub

Private Sub Command2_Click()
mdiMainHome.Picture1.Visible = False
frmUserDashboard.Show
End Sub

Private Sub documentation_Click()
```

```
Dim StrFilename As String

StrFilename = "C:\Users\Tony Shark\Documents\Files\Visual Basic\project\Student Attendance
Management System\Documents\documentation"
frmDocumentation.WebBrowser1.Navigate StrFilename
frmDocumentation.Show
End Sub

Private Sub exit_Click()
Unload Me
End Sub

Private Sub home_Click(Index As Integer)
mdiMainHome.Picture1.Visible = True
End Sub

Private Sub MDIForm_Unload(Cancel As Integer)
Unload frmDocumentation
Unload frmAbout
End Sub

Private Sub user_Click(Index As Integer)
mdiMainHome.Picture1.Visible = False
frmUserDashboard.Show
End Sub
```

ADMIN LOGIN

```
Option Explicit
Dim Conn As New ADODB.connection
Dim Rs As New ADODB.Recordset

Private Sub Command1_Click()
If Text1.Text = "" And Text2.Text = "" Then
MsgBox "Enter Admin Id and Password", vbExclamation, "Message"
Text1.SetFocus
Exit Sub
End If
If Text1.Text <> "" And Text2.Text = "" Then
MsgBox "Enter Password", vbExclamation, "Message"
Text2.SetFocus
Exit Sub
End If
If Text1.Text = "" And Text2.Text <> "" Then
MsgBox "Enter Admin Id", vbExclamation, "Message"
Text1.SetFocus
Exit Sub
End If
Rs.Open "select Admin_Id,Password from admin_table where Admin_Id='" & Text1.Text & "'",
```

```
Conn, adOpenKeyset

If Not Rs.EOF = True Then
If Text1.Text = Rs.Fields("Admin_Id").Value And Text2.Text = Rs.Fields("Password").Value Then
mdiMainHome.Picture1.Visible = False
mdiMainHome.Enabled = True
frmAdminPanel.Show
frmAdminPanel.Enabled = True
frmAdminPanel.Picture1.SetFocus
MsgBox "Login Successfull", vbInformation, "Message"
Unload Me
Exit Sub
Else
Rs.MoveNext
End If
End If
MsgBox "Invalid Admin Id or Password", vbCritical, "Message"
Text1.Text = ""
Text2.Text = ""
Text1.SetFocus
Rs.Close
End Sub

Private Sub Command2_Click()
mdiMainHome.Enabled = True
mdiMainHome.Show
Unload Me
End Sub

Private Sub Form_Load()
Conn.Open _
"Provider=sqloledb;" & _
"Data Source=LAPTOPKRCRUGPPS\SQLEXPRESS01;" & _
"Initial Catalog=SFAS_DB;" & _
"Trusted_Connection=yes;"
End Sub

Private Sub Form_Unload(Cancel As Integer)
Conn.Close
mdiMainHome.Enabled = True
mdiMainHome.Show
End Sub
```

ADMIN PANEL

```
Option Explicit

Private Sub Command1_Click()
frmMarkAttendance.Show
```

```
End Sub

Private Sub Command2_Click()
frmAddStudent.Show
End Sub

Private Sub Command3_Click()
frmEditStudent.Show
End Sub

Private Sub Command4_Click()
frmAddAdmin.Show
End Sub

Private Sub Command5_Click()
frmEditAdmin.Show
End Sub

Private Sub Image1_Click()
logout
End Sub

Private Sub Label2_Click()
Label2.Forecolor = &HC0&
logout
End Sub

Private Sub logout()
mdiMainHome.Picture1.Visible = True
mdiMainHome.Show
Unload frmMarkAttendance
Unload frmAddStudent
Unload frmEditStudent
Unload frmAddAdmin
Unload frmEditAdmin
Unload Me
End Sub
```

USER DASHBOARD

```
Option Explicit
Dim Conn As New ADODB.connection
Dim Rs As New ADODB.Recordset
Dim Rs1 As New ADODB.Recordset
Dim Rs2 As New ADODB.Recordset
Public signal1 As Integer

Private Sub Command1_Click()
frmIndividualAttendance.Show
```

```
End Sub

Private Sub Command2_Click()
frmViewAttendance.Show
End Sub

Private Sub Command3_Click()
Rs.Open "select * from overall_attendance_table", Conn, adOpenDynamic, adLockOptimistic
Set dRepOverallAttendanceDetails.DataSource = Rs
dRepOverallAttendanceDetails.Show
End Sub

Private Sub Command3_LostFocus()
On Error Resume Next
Rs.Close
End Sub

Private Sub Command4_Click()
Rs1.Open "select * from student_table", Conn, adOpenDynamic, adLockOptimistic
Set dRepStudentDetails.DataSource = Rs1
dRepStudentDetails.Show
End Sub

Private Sub Command4_LostFocus()
On Error Resume Next
Rs1.Close
End Sub

Private Sub Command5_Click()
Rs2.Open "select * from admin_table", Conn, adOpenDynamic, adLockOptimistic
Set dRepAdminDetails.DataSource = Rs2
dRepAdminDetails.Show
End Sub

Private Sub Command5_LostFocus()
On Error Resume Next
Rs2.Close
End Sub

Private Sub Form_Load()
Conn.Open _
"Provider=sqloledb;" & _
"Data Source=LAPTOPKRCRUGPPS\SQLEXPRESS01;" & _
"Initial Catalog=SFAS_DB;" & _
"Trusted_Connection=yes;"
End Sub

Private Sub Form_Unload(Cancel As Integer)
Conn.Close
End Sub
```

```
Private Sub Image1_Click()
goback
End Sub

Private Sub Label2_Click()
Label2.Forecolor = &HC000&
goback
End Sub

Private Sub goback()
mdiMainHome.Show
mdiMainHome.Picture1.Visible = True
Unload frmIndividualAttendance
Unload frmViewAttendance
signal1 = 1
Unload dRepAdminDetails
Unload dRepAttendanceDetails
Unload dRepOverallAttendanceDetails
Unload dRepStudentDetails
Unload Me
End Sub
```

MARK ATTENDANCE

```
Option Explicit
Dim Conn As New ADODB.connection
Dim Rs As New ADODB.Recordset
Dim Conn1 As New ADODB.connection
Dim Rs1 As New ADODB.Recordset
Dim Conn2 As New ADODB.connection
Dim Rs2 As New ADODB.Recordset
Dim Rs3 As New ADODB.Recordset
Dim sql As String
Dim signal As Integer

Private Sub Command1_Click()
enableoption
Frame1.Enabled = True
Label6.Forecolor = &HFFFFFF
unvisiblelabel
Rs.MoveFirst
loadvalue
Rs1.Requery
checkvalue
End Sub

Private Sub Command2_Click()
```

```
enableoption

Frame1.Enabled = True
Label6.Forecolor = &HFFFFFF
unvisiblelabel
Rs.MoveNext
If Rs.EOF = True Then
Rs.MoveFirst
End If
loadvalue
Rs1.Requery
checkvalue
End Sub

Private Sub Command3_Click()
enableoption
Frame1.Enabled = True
Label6.Forecolor = &HFFFFFF
unvisiblelabel
Rs.MovePrevious
If Rs.BOF = True Then
Rs.MoveLast
End If
loadvalue
Rs1.Requery
checkvalue
End Sub

Private Sub Command4_Click()
enableoption
Frame1.Enabled = True
Label6.Forecolor = &HFFFFFF
unvisiblelabel
Rs.MoveLast
loadvalue
Rs1.Requery
checkvalue
End Sub

Private Sub Command5_Click()
Unload Me
End Sub

Private Sub Form_Load()
Text1.Text = Date
Conn.Open _
"Provider=sqloledb;" & _
"Data Source=LAPTOPKRCRUGPPS\SQLEXPRESS01;" & _
"Initial Catalog=SFAS_DB;" & _
"Trusted_Connection=yes;"
```

```
Rs.Open "select Student_Id,Name,Photo,Course,Class from student_table", Conn, adOpenKeyset,
adLockPessimistic
Conn1.Open _
"Provider=sqloledb;" & _
"Data Source=LAPTOPKRCRUGPPS\SQLEXPRESS01;" & _
"Initial Catalog=SFAS_DB;" & _
"Trusted_Connection=yes;"
Rs1.Open "select attendance_table.Student_Id, attendance_table.Attendance from attendance_table where
attendance_table.Date='" & Text1.Text & "' order by Student_Id",
Conn, adOpenKeyset, adLockPessimistic
Conn2.Open _
"Provider=sqloledb;" & _
"Data Source=LAPTOPKRCRUGPPS\SQLEXPRESS01;" & _
"Initial Catalog=SFAS_DB;" & _
"Trusted_Connection=yes;"
End Sub

Private Sub Form_Unload(Cancel As Integer)
Conn.Close
Conn1.Close
Conn2.Close
End Sub

Private Sub Label6_Click()
Label6.Forecolor = &HFF&
enableoption
Frame1.Enabled = True
signal = 1
Label7.Visible = False
End Sub

Private Sub Option1_Click()
Dim str1, msg1, msg2 As String
str1 = "Present"
msg1 = "Attendance Updated Marked Present"
msg2 = "Attendance Marked Present"
Call markattendance(str1, msg1, msg2)
End Sub

Private Sub Option1_LostFocus()
Rs2.Close
Rs3.Close
End Sub

Private Sub Option2_Click()
Dim str1, msg1, msg2 As String
str1 = "Absent"
msg1 = "Attendance Updated Marked Absent"
msg2 = "Attendance Marked Absent"
Call markattendance(str1, msg1, msg2)
```

```
End Sub

Private Sub Option2_LostFocus()
Rs2.Close
Rs3.Close
End Sub

Private Sub loadvalue()
Text2.Text = Rs.Fields("Student_Id").Value
Text3.Text = Rs.Fields("Name").Value
Image1.Picture = LoadPicture(Rs.Fields("Photo").Value)
Text4.Text = Rs.Fields("Course").Value
Text5.Text = Rs.Fields("Class").Value
End Sub

Private Sub checkvalue()
If Rs1.RecordCount <> 0 Then
Rs1.MoveFirst
Do Until Rs1.EOF
If Rs1.Fields("Student_Id").Value = Text2.Text And (Rs1.Fields("Attendance").Value = "Present" Or
Rs1.Fields("Attendance").Value = "Absent") Then
unenableoption
visiblelabel
End If
Rs1.MoveNext
Loop
End If
End Sub

Private Sub markattendance(ByVal str1 As String, ByVal msg1 As String, ByVal msg2
 As String)
On Error GoTo Update
If signal = 1 Then
sql = "update attendance_table set Attendance='" & str1 & "' where Student_Id='" &
Text2.Text & "' and Date='" & Text1.Text & "'"
Conn.Execute sql
MsgBox msg1, vbInformation, "Message"
signal = 0
Frame1.Enabled = False
Option1.Value = False
Option2.Value = False
unenableoption
Else
sql = "insert into attendance_table(Student_Id, Date, Attendance) values ("
sql = sql & "'" & Text2.Text & "',"
sql = sql & "'" & Text1.Text & "',"
sql = sql & "'" & str1 & "')"
Conn.Execute sql
MsgBox msg2, vbInformation, "Message"
Frame1.Enabled = False
```

```
Option1.Value = False
Option2.Value = False
unenableoption
End If
Dim tot_class, class_attend As Double
Dim per As String
Rs2.Open "select count(Attendance) as overalltotalpresent from attendance_table where Attendance=
'Present' and Student_Id='" & Text2.Text & "'", Conn2, adOpenKeyset
Rs3.Open "select count(Attendance) as overalltotalabsent from attendance_table where Attendance=
'Absent' and Student_Id='" & Text2.Text & "'", Conn2, adOpenKeyset
class_attend = Rs2.Fields("overalltotalpresent").Value * 6
tot_class = (Rs2.Fields("overalltotalpresent").Value + Rs3.Fields("overalltotalabsent").Value) * 6
per = Round((((Rs2.Fields("overalltotalpresent").Value * 6) / tot_class) * 100), 2) & " %"
sql = "insert into overall_attendance_table(Student_Id, Name, Course, Class, Total_Classes,
Classes_Attended, Percentage) values ("
sql = sql & "'" & Text2.Text & "',"
sql = sql & "'" & Text3.Text & "',"
sql = sql & "'" & Text4.Text & "',"
sql = sql & "'" & Text5.Text & "',"
sql = sql & "'" & tot_class & "',"
sql = sql & "'" & class_attend & "',"
sql = sql & "'" & per & "')"
Conn.Execute sql
Update:
sql = "update overall_attendance_table set Total_Classes='" & tot_class & "', Classes_Attended='" &
class_attend & "', Percentage='" & per & "' where Student_Id='" & Text2.Text & "'"
Conn.Execute sql
End Sub

Private Sub enableoption()
Option1.Enabled = True
Option2.Enabled = True
End Sub

Private Sub unenableoption()
Option1.Enabled = False
Option2.Enabled = False
End Sub

Private Sub visiblelabel()
Label6.Visible = True
Label7.Visible = True
End Sub

Private Sub unvisiblelabel()
Label6.Visible = False
Label7.Visible = False
End Sub
```

ADD STUDENT

```
Option Explicit
Dim Conn As New ADODB.connection
Dim Rs As New ADODB.Recordset
Dim str As String

Private Sub Command1_Click()
CommonDialog1.ShowOpen
CommonDialog1.Filter = "Jpeg|*.jpg"
str = CommonDialog1.FileName
Image1.Picture = LoadPicture(str)
End Sub

Private Sub Command2_Click()
frmAddStudent.Show
enablecontrol
Text1.SetFocus
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Image1.Picture = LoadPicture("")
Command2.Enabled = False
End Sub

Private Sub Command3_Click()
If Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or
Combo1.Text = "" Or Combo2.Text = "" Or Combo3.Text = "" Or str = "" Then
MsgBox "All Fields are Mandantoty", vbExclamation, "Message"
Exit Sub
End If
Dim sql As String
sql = "insert into student_table(Student_Id, Name, Photo, Gender, Course, Class, Mobile_No,
Address) values ("
sql = sql & "'" & Text1.Text & "',"
sql = sql & "'" & Text2.Text & "',"
sql = sql & "'" & str & "',"
sql = sql & "'" & Combo1.Text & "',"
sql = sql & "'" & Combo2.Text & "',"
sql = sql & "'" & Combo3.Text & "',"
sql = sql & "'" & Text3.Text & "',"
sql = sql & "'" & Text4.Text & "')"
Conn.Execute sql
MsgBox "New Student Added", vbInformation, "Message"
unenablecontrol
Command2.Enabled = True
End Sub
```

```
Private Sub Form_Load()
Conn.Open _
"Provider=sqloledb;" & _
"Data Source=LAPTOPKRCRUGPPS\SQLEXPRESS01;" & _
"Initial Catalog=SFAS_DB;" & _
"Trusted_Connection=yes;"
End Sub

Private Sub Form_Unload(Cancel As Integer)
Conn.Close
End Sub

Sub enablecontrol()
Text1.Enabled = True
Text2.Enabled = True
Text3.Enabled = True
Text4.Enabled = True
Combo1.Enabled = True
Combo2.Enabled = True
Combo3.Enabled = True
Image1.Enabled = True
Command1.Enabled = True
End Sub

Sub unenablecontrol()
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Combo1.Enabled = False
Combo2.Enabled = False
Combo3.Enabled = False
Image1.Enabled = False
Command1.Enabled = False
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
If (KeyAscii >= 65 And KeyAscii <= 90) Or (KeyAscii >= 97 And KeyAscii <= 122) Or
(KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 8 Then
Else
KeyAscii = 0
End If
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
If (KeyAscii >= 65 And KeyAscii <= 90) Or (KeyAscii >= 97 And KeyAscii <= 122) Or
KeyAscii = 32 Or KeyAscii = 8 Then
Else
```

```
KeyAscii = 0
End If
End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
If (KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 8 Then
Else
KeyAscii = 0
End If
End Sub
```

EDIT STUDENT

```
Option Explicit
Dim i As Integer
Dim str As String
Dim msg As String
Dim sql As String
Dim Conn As New ADODB.connection
Dim Rs As New ADODB.Recordset
Dim signal As Integer

Private Sub Command1_Click()
If Text1.Text = "" Then
clearcontrol
MsgBox "Enter Student Id", vbExclamation, "Message"
hidecontrol
Exit Sub
End If
If Text1.Text <> "" Then
Rs.Open "select * from student_table where Student_Id='" & Text1.Text & "'", Conn, adOpenKeyset
If Not Rs.EOF = True Then
Text2.Text = Rs.Fields("Student_Id").Value
Text3.Text = Rs.Fields("Name").Value
Image1.Picture = LoadPicture(Rs.Fields("Photo").Value)
Combo1.Text = Rs.Fields("Gender").Value
Combo2.Text = Rs.Fields("Course").Value
Combo3.Text = Rs.Fields("Class").Value
Text4.Text = Rs.Fields("Mobile_No").Value
Text5.Text = Rs.Fields("Address").Value
Rs.MoveNext
showcontrol
unenablecontrol
Command3.Enabled = False
MsgBox "Student Found", vbInformation, "Message"
Rs.Close
Exit Sub
```

```
End If

End If
clearcontrol
MsgBox "Invalid Student Id", vbCritical, "Message"
hidecontrol
Rs.Close
End Sub

Private Sub Command2_Click()
CommonDialog1.ShowOpen
CommonDialog1.Filter = "Jpeg|*.jpg"
str = CommonDialog1.FileName
Image1.Picture = LoadPicture(str)
End Sub

Sub showcontrol()
For i = 1 To 8
Label3(i).Visible = True
Label4(i).Visible = True
Next
Text2.Visible = True
Text3.Visible = True
Text4.Visible = True
Text5.Visible = True
Combo1.Visible = True
Combo2.Visible = True
Combo3.Visible = True
Image1.Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
End Sub

Sub hidecontrol()
For i = 1 To 8
Label3(i).Visible = False
Label4(i).Visible = False
Next
Text2.Visible = False
Text3.Visible = False
Text4.Visible = False
Text5.Visible = False
Combo1.Visible = False
Combo2.Visible = False
Combo3.Visible = False
Image1.Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
```

```
End Sub

Sub unenablecontrol()
For i = 1 To 8
Label4(i).Forecolor = &HFFFFFF
Next i
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text5.Enabled = False
Combo1.Enabled = False
Combo2.Enabled = False
Combo3.Enabled = False
Command2.Enabled = False
End Sub
Sub clearcontrol()
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Image1.Picture = LoadPicture("")
End Sub

Private Sub Command3_Click()
If signal = 1 Then
If Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or Text5.Text = "" Or
Combo1.Text = "" Or Combo2.Text = "" Or Combo3.Text = "" Or str = "" Then
MsgBox "All Fields are Mandantoty", vbExclamation, "Message"
Exit Sub
End If
msg = MsgBox("Are you sure want to Update", vbQuestion + vbYesNo, "Message")
If msg = vbYes Then
sql = "update student_table set Student_Id='" & Text2.Text & "', Name='" & Text3.Text
& "', Photo='" & str & "', Gender='" & Combo1.Text & "',Course='" & Combo2.Text & "',
Class='" & Combo3.Text & "', Mobile_No='" & Text4.Text & "', Address='" & Text5.Text
& "' where Student_Id='" & Text1.Text & "' "
Conn.Execute sql
MsgBox "Student Information Updated", vbInformation, "Message"
unenablecontrol
Command3.Enabled = False
Exit Sub
End If
Else
If Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or Text5.Text = "" Or
Combo1.Text = "" Or Combo2.Text = "" Or Combo3.Text = "" Then
MsgBox "All Fields are Mandantoty", vbExclamation, "Message"
Exit Sub
End If
msg = MsgBox("Are you sure want to Update", vbQuestion + vbYesNo, "Message")
If msg = vbYes Then
```

```
sql = "update student_table set Student_Id='" & Text2.Text & "', Name='" & Text3.Text & "', Gender='" &
Combo1.Text & "', Course='" & Combo2.Text & "', Class='" & Combo3.Text & "', Mobile_No='" &
Text4.Text & "', Address='" & Text5.Text & "' where Student_Id='" & Text1.Text & "' "
Conn.Execute sql
MsgBox "Student Information Updated", vbInformation, "Message"
unenablecontrol
Command3.Enabled = False
Exit Sub
End If
End If
End Sub

Private Sub Command4_Click()
If Text1.Text = "" And Text2.Text = "" Then
MsgBox "Admin Id is Neccessary for deletion", vbExclamation, "Message"
hidecontrol
unenablecontrol
Exit Sub
End If
msg = MsgBox("Are you sure want to Delete", vbCritical + vbYesNo, "Message")
If msg = vbYes Then
sql = "delete from student_table where Student_Id='" & Text1.Text & "'"
Conn.Execute sql
MsgBox "Student Deleted from Records", vbInformation, "Message"
clearcontrol
Exit Sub
End If
End Sub

Private Sub Form_Load()
hidecontrol
Conn.Open _
"Provider=sqloledb;" & _
"Data Source=LAPTOPKRCRUGPPS\SQLEXPRESS01;" & _
"Initial Catalog=SFAS_DB;" & _
"Trusted_Connection=yes;"
End Sub

Private Sub Form_Unload(Cancel As Integer)
Conn.Close
End Sub

Private Sub Label4_Click(Index As Integer)
Command3.Enabled = True
Label4(Index).Forecolor = &HFF&
If Index = 1 Then
Text2.Enabled = True
Text2.SetFocus
```

```
ElseIf Index = 2 Then
Text3.Enabled = True
Text3.SetFocus
ElseIf Index = 3 Then
Command2.Enabled = True
Command2.SetFocus
Image1.Picture = LoadPicture("")
signal = 1
Command2_Click
ElseIf Index = 4 Then
Combo1.Enabled = True
Combo1.SetFocus
ElseIf Index = 5 Then
Combo2.Enabled = True
Combo2.SetFocus
ElseIf Index = 6 Then

Combo3.Enabled = True
Combo3.SetFocus
ElseIf Index = 7 Then
Text4.Enabled = True
Text4.SetFocus
ElseIf Index = 8 Then
Text5.Enabled = True
Text5.SetFocus
End If
End Sub
```

ADD ADMIN

```
Option Explicit
Dim Conn As New ADODB.connection
Dim Rs As New ADODB.Recordset
Dim str As String

Private Sub Command3_Click()
Text1.SetFocus
If Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or
Text5.Text = "" Or str = "" Then
MsgBox "All Fields are Mandantoty", vbExclamation, "Message"
Exit Sub
End If
Dim sql As String
sql = "insert into admin_table(Admin_Id, Name, Password, Mobile_No, Address, Photo)
values ("
sql = sql & "'" & Text2.Text & "',"
sql = sql & "'" & Text1.Text & "',"
sql = sql & "'" & Text3.Text & "',"
sql = sql & "'" & Text4.Text & "',"
```

```
sql = sql & "'" & Text5.Text & "',"
sql = sql & "'" & str & "')"
Conn.Execute sql
MsgBox "New Admin Added", vbInformation, "Message"
unenablecontrol
Command2.Enabled = True
End Sub

Private Sub Command1_Click()
CommonDialog1.ShowOpen
CommonDialog1.Filter = "Jpeg|*.jpg"
str = CommonDialog1.FileName
Image1.Picture = LoadPicture(str)
End Sub

Private Sub Command2_Click()
enablecontrol
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Image1.Picture = LoadPicture("")
Text1.SetFocus
Command2.Enabled = False
End Sub

Private Sub Form_Load()
Conn.Open _
"Provider=sqloledb;" & _
"Data Source=LAPTOPKRCRUGPPS\SQLEXPRESS01;" & _
"Initial Catalog=SFAS_DB;" & _
"Trusted_Connection=yes;"
End Sub

Private Sub Form_Unload(Cancel As Integer)
Conn.Close
End Sub

Sub enablecontrol()
Text1.Enabled = True
Text2.Enabled = True
Text3.Enabled = True
Text4.Enabled = True
Text5.Enabled = True
Image1.Enabled = True
Command1.Enabled = True
End Sub

Sub unenablecontrol()
```

```
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text5.Enabled = False
Image1.Enabled = False
Command1.Enabled = False
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
If (KeyAscii >= 65 And KeyAscii <= 90) Or (KeyAscii >= 97 And KeyAscii <= 122) Or
KeyAscii = 32 Or KeyAscii = 8 Then
Else
KeyAscii = 0
End If
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
If (KeyAscii >= 65 And KeyAscii <= 90) Or (KeyAscii >= 97 And KeyAscii <= 122) Or
(KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 8 Then
Else
KeyAscii = 0
End If
End Sub

Private Sub Text4_KeyPress(KeyAscii As Integer)
If (KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 8 Then
Else
KeyAscii = 0
End If
End Sub
```

EDIT ADMIN

```
Option Explicit
Dim i As Integer
Dim str As String
Dim msg As String
Dim sql As String
Dim Conn As New ADODB.connection
Dim Rs As New ADODB.Recordset
Dim signal As Integer

Private Sub Command1_Click()
If Text1.Text = "" Then
clearcontrol
MsgBox "Enter Admin Id", vbExclamation, "Message"
hidecontrol
Exit Sub
```

```
End If

If Text1.Text <> "" Then
Rs.Open "select * from admin_table where Admin_Id='" & Text1.Text & "'", Conn, adOpenKeyset
If Not Rs.EOF = True Then
Text2.Text = Rs.Fields("Name").Value
Text3.Text = Rs.Fields("Admin_Id").Value
Text4.Text = Rs.Fields("Password").Value
Text5.Text = Rs.Fields("Mobile_No").Value
Text6.Text = Rs.Fields("Address").Value
Image1.Picture = LoadPicture(Rs.Fields("Photo").Value)
Rs.MoveNext
showcontrol
unenablecontrol
Command3.Enabled = False
MsgBox "Admin Found", vbInformation, "Message"
Rs.Close
Exit Sub
End If
End If
clearcontrol
MsgBox "Invalid Admin Id", vbCritical, "Message"
hidecontrol
Rs.Close
End Sub

Private Sub Command2_Click()
CommonDialog1.ShowOpen
CommonDialog1.Filter = "Jpeg|*.jpg"
str = CommonDialog1.FileName
Image1.Picture = LoadPicture(str)
End Sub

Private Sub Command3_Click()
If signal = 1 Then
If Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or Text5.Text = "" Or
Text6.Text = "" Or str = "" Then
MsgBox "All Fields are Mandantoty", vbExclamation, "Message"
Exit Sub
End If
msg = MsgBox("Are you sure want to Update", vbQuestion + vbYesNo, "Message")
If msg = vbYes Then
sql = "update admin_table set Admin_Id='" & Text3.Text & "', Name='" & Text2.Text & "', Password='"
& Text4.Text & "', Mobile_No='" & Text5.Text & "', Address='" & Text6.Text
& "', Photo='" & str & "' where Admin_Id='" & Text1.Text & "' "
Conn.Execute sql
MsgBox "Admin Information Updated", vbInformation, "Message"
unenablecontrol
Command3.Enabled = False
Exit Sub
```

```
End If
Else
If Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or Text5.Text = "" Or Text6.Text = "" Then
MsgBox "All Fields are Mandantoty", vbExclamation, "Message"
Exit Sub
End If
msg = MsgBox("Are you sure want to Update", vbQuestion + vbYesNo, "Message")
If msg = vbYes Then
sql = "update admin_table set Admin_Id='" & Text3.Text & "', Name='" & Text2.Text & "', Password='"
& Text4.Text & "', Mobile_No='" & Text5.Text & "', Address='" & Text6.Text
& "' where Admin_Id='" & Text1.Text & "' "
Conn.Execute sql
MsgBox "Admin Information Updated", vbInformation, "Message"
unenablecontrol
Command3.Enabled = False
Exit Sub
End If
End If
End Sub

Private Sub Command4_Click()
If Text1.Text = "" And Text3.Text = "" Then
MsgBox "Admin Id is Neccessary for deletion", vbExclamation, "Message"
hidecontrol
unenablecontrol
Exit Sub
End If
msg = MsgBox("Are you sure want to Delete", vbCritical + vbYesNo, "Message")
If msg = vbYes Then
sql = "delete from admin_table where Admin_Id='" & Text1.Text & "'"
Conn.Execute sql
MsgBox "Admin Deleted from Records", vbInformation, "Message"
clearcontrol
Exit Sub
End If
End Sub

Private Sub Form_Load()
Conn.Open _
"Provider=sqloledb;" & _
"Data Source=LAPTOPKRCRUGPPS\SQLEXPRESS01;" & _
"Initial Catalog=SFAS_DB;" & _
"Trusted_Connection=yes;"
End Sub

Private Sub Form_Unload(Cancel As Integer)
Conn.Close
End Sub

Sub showcontrol()
```

```
For i = 1 To 6
Label3(i).Visible = True
Label4(i).Visible = True
Next
Text2.Visible = True
Text3.Visible = True
Text4.Visible = True
Text5.Visible = True
Text6.Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
Image1.Visible = True
End Sub

Sub hidecontrol()
For i = 1 To 6
Label3(i).Visible = False
Label4(i).Visible = False
Next
Text2.Visible = False
Text3.Visible = False
Text4.Visible = False
Text5.Visible = False
Text6.Visible = False
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Image1.Visible = False
End Sub

Sub unenablecontrol()
For i = 1 To 6
Label4(i).Forecolor = &HFFFFFF
Next i
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text5.Enabled = False
Text6.Enabled = False
Command2.Enabled = False
End Sub
Sub clearcontrol()
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Image1.Picture = LoadPicture("")
End Sub
```

```
Private Sub Label4_Click(Index As Integer)
Command3.Enabled = True
Label4(Index).Forecolor = &HFF&
If Index = 1 Then
Text2.Enabled = True
Text2.SetFocus
ElseIf Index = 2 Then
Text3.Enabled = True
Text3.SetFocus
ElseIf Index = 3 Then
Text4.Enabled = True
Text4.SetFocus
ElseIf Index = 4 Then
Text5.Enabled = True
Text5.SetFocus
ElseIf Index = 5 Then
Text6.Enabled = True
Text6.SetFocus
ElseIf Index = 6 Then
Command2.Enabled = True
Command2.SetFocus
signal = 1
Call Command2_Click
End If
End Sub
```

INDIVIDUAL ATTENDANCE

```
Option Explicit
Dim Conn As New ADODB.connection
Dim Rs As New ADODB.Recordset

Private Sub Command1_Click()
If DTPicker1.Value = "" Then
MsgBox "Select Date", vbExclamation, "Message"
Exit Sub
End If
Rs.Open "select student_table.Student_Id, student_table.Name, student_table.Photo, student_table.Course,
student_table.Class, attendance_table.Attendance from student_table, attendance_table where
attendance_table.Date='" & DTPicker1.Value & "' and student_table.Student_Id=attendance_table.Student_Id
order by Student_Id", Conn, adOpenKeyset
enablecontrol
Command1.Enabled = False
End Sub

Private Sub Command2_Click()
If Rs.RecordCount = 0 Then
```

```
clearcontrol

MsgBox "No Attendance was taken on " & DTPicker1.Value, vbInformation, "Message"
Exit Sub
End If
Rs.MoveFirst
loadvalue
End Sub
Private Sub Command3_Click()
If Rs.RecordCount = 0 Then
clearcontrol
MsgBox "No Attendance was taken on " & DTPicker1.Value, vbInformation, "Message"
Exit Sub
End If
Rs.MoveNext
If Rs.EOF = True Then
Rs.MoveFirst
End If
loadvalue
End Sub

Private Sub Command4_Click()
If Rs.RecordCount = 0 Then
clearcontrol
MsgBox "No Attendance was taken on " & DTPicker1.Value, vbInformation, "Message"
Exit Sub
End If
Rs.MovePrevious
If Rs.BOF = True Then
Rs.MoveLast
End If
loadvalue
End Sub

Private Sub Command5_Click()
If Rs.RecordCount = 0 Then
clearcontrol
MsgBox "No Attendance was taken on " & DTPicker1.Value, vbInformation, "Message"
Exit Sub
End If
Rs.MoveLast
loadvalue
End Sub

Private Sub Command6_Click()
Unload Me
End Sub

Private Sub DTPicker1_Change()
Conn.Close
```

```
Unenablecontrol
Command1.Enabled = True
connection
clearcontrol
End Sub

Private Sub Form_Load()
DTPicker1.Value = ""
connection
End Sub

Private Sub Form_Unload(Cancel As Integer)
Conn.Close
End Sub

Private Sub connection()
Conn.Open _
"Provider=sqloledb;" & _
"Data Source=LAPTOPKRCRUGPPS\SQLEXPRESS01;" & _
"Initial Catalog=SFAS_DB;" & _
"Trusted_Connection=yes;"
End Sub

Private Sub clearcontrol()
Text2.Text = ""
Text3.Text = ""
Image1.Picture = LoadPicture("")
Text4.Text = ""
Text5.Text = ""
Label6.Caption = ""
End Sub

Private Sub enablecontrol()
Command2.Enabled = True
Command3.Enabled = True
Command4.Enabled = True
Command5.Enabled = True
End Sub

Private Sub unenablecontrol()
Command2.Enabled = False
Command3.Enabled = False
Command4.Enabled = False

Command5.Enabled = False
End Sub

Private Sub loadvalue()
Text2.Text = Rs.Fields("Student_Id").Value
Text3.Text = Rs.Fields("Name").Value
```

```
Image1.Picture = LoadPicture(Rs.Fields("Photo").Value)

Text4.Text = Rs.Fields("Course").Value
Text5.Text = Rs.Fields("Class").Value
If Rs.Fields("Attendance").Value = "Present" Then
Label6.Caption = Rs.Fields("Attendance").Value
Label6.Forecolor = &HFF00&
Else
Label6.Caption = Rs.Fields("Attendance").Value
Label6.Forecolor = &HFF&
End If
End Sub
```

VIEW ATTENDANCE

```
Option Explicit
Dim Conn As New ADODB.connection
Dim Rs As New ADODB.Recordset
Dim Rs1 As New ADODB.Recordset
Dim Rs2 As New ADODB.Recordset

Private Sub Command2_Click()
If DTPicker1.Value = "" Then
MsgBox "Select Date", vbInformation, "Message"
Exit Sub
End If
Rs.Open "select student_table.Student_Id, student_table.Name, student_table.Course, student_table.Class,
attendance_table.Attendance from student_table, attendance_table
where attendance_table.Date='" & DTPicker1.Value & "' and
student_table.Student_Id=attendance_table.Student_Id order by Student_Id", Conn, adOpenKeyset
Rs1.Open "select count(Attendance) as totalpresent from attendance_table where Attendance='Present' and
Date='01-03-2023'", Conn, adOpenKeyset
Rs2.Open "select count(Attendance) as totalabsent from attendance_table where Attendance='Absent' and
Date='01-03-2023'", Conn, adOpenKeyset
If Rs.RecordCount = 0 Then
MsgBox "No Attendance was taken on " & DTPicker1.Value, vbInformation, "Message"
Exit Sub
End If
Set dRepAttendanceDetails.DataSource = Rs
dRepAttendanceDetails.Sections("Section4").Controls.Item("Label6").Caption = "" & DTPicker1.Value
dRepAttendanceDetails.Sections("Section5").Controls.Item("Label9").Caption = "" &
Rs1.Fields("totalpresent").Value
dRepAttendanceDetails.Sections("Section5").Controls.Item("Label10").Caption = "" &
Rs2.Fields("totalabsent").Value
dRepAttendanceDetails.Show
End Sub

Private Sub Command3_Click()
Unload Me
```

```
End Sub

Private Sub DTPicker1_Change()
Conn.Close
Command2.Enabled = True
connection
End Sub

Private Sub Form_Load()
DTPicker1.Value = ""
connection
End Sub

Private Sub Form_Unload(Cancel As Integer)
Conn.Close
Unload dRepAttendanceDetails
End Sub

Private Sub connection()
Conn.Open _
"Provider=sqloledb;" & _
"Data Source=LAPTOPKRCRUGPPS\SQLEXPRESS01;" & _
"Initial Catalog=SFAS_DB;" & _
"Trusted_Connection=yes;"
End Sub
```

ATTENDANCE DETAILS

```
Option Explicit

Private Sub DataReport_Terminate()
If frmUserDashboard.signal1 = 1 Then
Exit Sub
End If
frmViewAttendance.DTPicker1.Value = ""
frmViewAttendance.DTPicker1.SetFocus
frmViewAttendance.Command2.Enabled = False
End Sub
```

OVERALL ATTENDANCE DETAILS

```
Option Explicit

Private Sub DataReport_Terminate()
If frmUserDashboard.signal1 = 1 Then
Exit Sub
End If
frmUserDashboard.Picture3.SetFocus
```

```
End Sub
```

STUDENT DETAILS

```
Option Explicit

Private Sub DataReport_Terminate()
If frmUserDashboard.signal1 = 1 Then
Exit Sub
End If
frmUserDashboard.Picture4.SetFocus
End Sub
```

ADMIN DETAILS

```
Option Explicit

Private Sub DataReport_Terminate()
If frmUserDashboard.signal1 = 1 Then
Exit Sub
End If
frmUserDashboard.Picture5.SetFocus
End Sub
```

TESTING

TESTING

Introduction to Testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Softwaretesting can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software productis fit for use. Software testing involves the execution of a software component or system component to evaluate one or more properties of interest.

Types of Testing

The general testing process is the creation of a testing strategy(which sometimesincludes the execution of the testing case), creation of a test plan/design(which usually includes test case and test procedure) and the execution of tests. Test are inputs that have been devised to test the system.

- Black-Box Testing
- White Box Testing
- Alpha Testing
- Beta Testing
- Clean Room Testing

Black-Box Testing:

Black box Testing is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional. It focuses on the functional requirements of the software. In black-box testing, the cases are derived from the specification of the system, the implementation details of the system are not taken intoconsiderations.

White Box Testing:

White-box testing, sometimes called Glass box testing or structural testing or clear box testing.

, White box testing is used to test areas that cannot be reached from ablack box testing.

White-box texting user an internal perspective of the system to design the test cases based on the internal structure.

Alpha Testing:

Alpha testing is simulated for actual operational testing by potential users/customers or an independent test team at the developers site. Alpha testing is often employed for off-the-self software as a form of internal acceptance testing beforethe software goes to beta testing.

Beta Testing:

Beta testing comes after alpha testing. Version of the software, known as beta versions, are related to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the productshas few faults or bugs.

Clean Room Testing:

Clean room testing makes use of incremental development model. The initialincrement delivered to the customer with the critical functionalities and less important features are added in later increments. The initial increment is released for experimentation with an intention to identify the requirements problem. These problems are removed in later increments. New increments are added to the existing ones and the integrated system is validated. As new increments are added, the initial increments are also released

CONCLUSION AND FUTURE ENHANCEMENTS

CONCLUSION

The Student Attendance Management System (SAMS) represents a significant advancement in automating attendance tracking for educational institutions, addressing the inefficiencies of manual methods through a secure, role-based digital solution. Built using Microsoft Visual Basic 6.0 and Microsoft SQL Server 2022, the system streamlines workflows by enabling administrators to manage student and staff profiles, mark daily attendance with photo identification, and generate real-time reports. Teachers and staff benefit from intuitive dashboards to view attendance records, while strict input validation ensures data accuracy. The relational database design enforces referential integrity, automating attendance percentage calculations and maintaining consistency across records. By replacing paper-based processes, SAMS minimizes human error, reduces administrative effort, and provides institutions with a scalable, auditable platform for transparent attendance management.

FUTURE ENHANCEMENT

To further enhance the system's capabilities, several improvements can be explored. Integration of biometric authentication, such as facial recognition, could automate student verification during attendance marking, building on the existing photo-display functionality. Expanding accessibility through a mobile or web interface would allow remote attendance updates, leveraging APIs for geolocation or QR-code-based tracking. Security can be strengthened by migrating from plaintext password storage to hashing algorithms like bcrypt, ensuring compliance with modern data protection standards. Additionally, multi-language support and role granularity (e.g., department-

specific permissions) would improve accessibility and administrative flexibility. These enhancements, while backward-compatible with the current codebase, would position SAMS as a future-ready solution for evolving educational needs.

BIBLIOGRAPHY

BIBLIOGRAPHY

Reference Websites :

- https://www.vbforums.com/
- https://stackoverflow.com/
- https://social.msdn.microsoft.com/Forums/en-us/home
- https://www.vbtutor.net/vbtutor.html
- https://youtube.com/@NevonProjectsOfficial
- https://youtube.com/@SandeepKaundalSir

Reference Books :

- Visual Basic Programming 4th Sem BCA Textbook
- Database Management System 2nd Sem BCA Textbook
- Software Engineering 5th Sem BCA Textbook