**PROJECT REPORT**

ON

**ERP SYSTEM**

**SUBMITTED TO**



**G.H.G. KHALSA COLLEGE, GURUSAR SADHAR**

In the partial fulfilment of Degree of BCA

**SUBMITTED BY**

RUKMANI KUMARI SHARMA

BCA – 6th Semester

Session - 2023-24



**PANJAB UNIVERSITY**

**CHANDIGARH**

**DECLARATION**

I declare that this written submission represents my ideas in my own words, and where others’ ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed**.**

RUKMANI KUMARI SHARMA

**ACKNOWLEDGEMENT**

I take this opportunity to express my acknowledgement and a deep sense of gratitude for rendering valuable assistance and guidance to me by the following personality for the successful completion of my project.

We express our heartfelt gratitude to the Principal Prof. Inderjeet Singh, G.H.G Khalsa College Gurusar Sadhar, and the management for providing the needed infrastructure. We wish to express our deep sense of gratitude to Our Guide Prof. Shubham Kumar and HOD Prof. Tarsem Singh for their able guidance and useful suggestions, which helped us in completing the project work, in time. Finally, yet importantly, we would like to express our heartfelt thanks to our beloved parents for their blessings, friends/classmates for their help, and wishes for the successful completion of this project.

We thank all these people fervently as these were the leaders and we simply followed them.

RUKMANI KUMARI SHARMA

BCA – 6th Semester

21047458

**CERTIFICATE**

This is to certify that the project entitled ERP SYSTEM has been submitted for the fulfillment of the requirement of the degree of Bachelors of Computer Applications of Panjab University, Chandigarh. This project is bona fide work of

RUKMANI KUMARI SHARMA no part of it has been submitted for any other Degree.

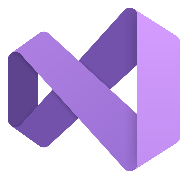
|  |  |
| --- | --- |
| **HEAD OF DEPARTMENT**  Prof. Tarsem Singh  Computer Science and Applications | **MENTOR**  BOHAR SINGH  Computer Science and Applications |

**INDEX OF PROJECT**

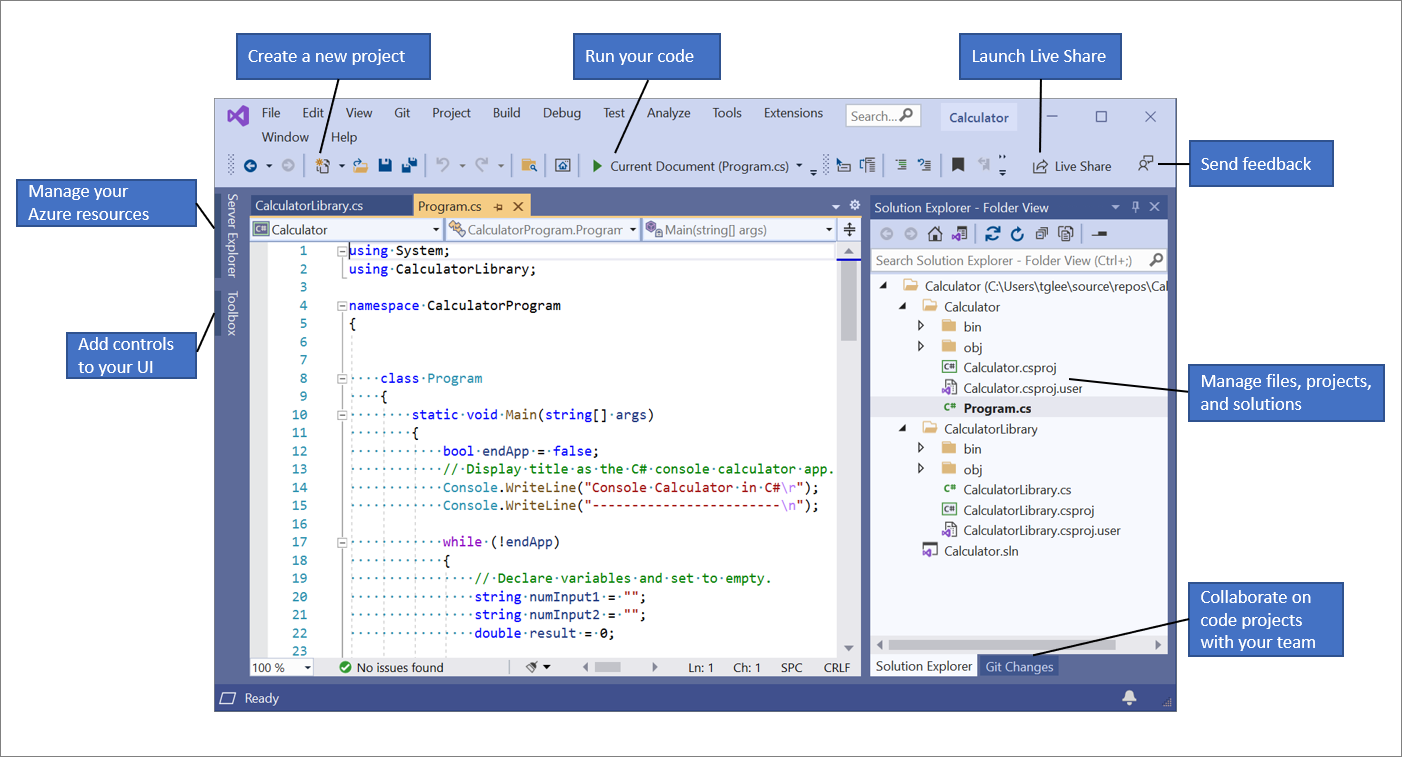
|  |  |  |
| --- | --- | --- |
| **S.no.** | **Title** | **Page No.** |
| **1** | **Chapter 1: (Introduction)** |  |
|  | 1.1Introduction to Visual Studio .NET |  |
|  | 1.2Introduction to .NET Framework |  |
|  | 1.3Features of .NET Framework |  |
|  | 1.4 Window Applications |  |
|  |  |  |
| **2** | **Chapter 2: (Introduction about Project)** |  |
|  | 2.1 About Project |  |
|  |  |  |
| **3** | **Chapter 3: ( System Requirements and Specifications)** |  |
|  | 3.1 Definition |  |
|  | 3.2 Hardware and Software Requirements |  |
|  |  |  |
| **4** | **Chapter 4: (System Design)** |  |
|  | 4.1 Definition |  |
|  | 4.2 Data Flow Diagram (DFD) |  |
|  | 4.3 Entity-Relation Diagram (ER) |  |
|  |  |  |
| **5** | **Chapter 5: (Source Code)** |  |
|  | 5.1 Source of Project |  |
|  |  |  |
| **6** | **Chapter 6: (Testing and Implementation)** |  |
|  | 6.1 Testing Definition and Importance |  |
|  | 6.2 Testing Report |  |
|  |  |  |
| **7** | **Chapter 7: (System Implementation)** |  |
|  | 7.1 Definition of Implementation |  |
|  | 7.2 Deployment of Project |  |
|  |  |  |
| **8** | **Chapter 8: (Output and Future Scope)** |  |
|  | 8.1 Output Screenshots |  |
|  | 8.2 Future Scope |  |
|  | 8.3 Bibliography |  |

Chapter 1:

Introduction

**Introduction to Visual Studio .NET**

Visual Studio .NET is a Microsoft-integrated development environment (IDE) that can be used for developing consoles, graphical user interfaces (GUIs), Windows Forms, Web services and Web applications.Visual Studio is used to write native code and managed code supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight. Visual Studio .NET’s code editor supports IntelliSense and code refactoring, while the Visual Studio .NET integrated debugger supports both source and machine-level debugging. Visual Studio .NET includes other built-in tools, like a form designer, which is useful when building GUI applications; a Web designer that creates dynamic Web pages; a class designer that is used to create custom libraries, and a schema designer for database support.

****

Tool windows

Tool windows are listed on the View menu and are defined by the current application and its add-ins. Developer can arrange tool windows in the IDE to:

* Show or hide automatically
* Tab link with other tool windows
* Dock against the edges of the IDE
* Float over
* Display on other monitors

In addition, developer can display more than one instance of certain tool windows at a time. For example, developer could display more than one Web browser window. Developer can create another instance of a tool window by choosing New Window on the Window menu. Also, developer can determine how the Close and Auto Hide buttons affect a group of tool windows docked together.

Document windows

Document windows are dynamically created when developer open or create files or other items. The list of open document windows appears in the Window menu in the current z-order, with the top-most window listed first.

The ways in which developer manage document windows depends greatly upon the interface mode selected on the General page of the Environment folder in the Options dialog box. Developer can choose to work in either Multiple Document Interface (MDI) or Tabbed Documents mode. Experiment with these settings to create a document editing environment that satisfies  needs and preferences.

**Introduction to .NET Framework**

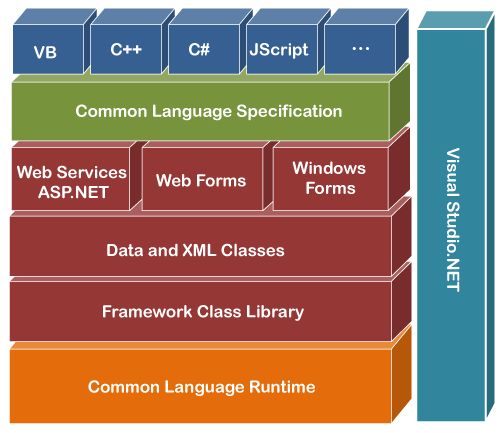
The **.NET Framework** is a software development framework developed by Microsoft that provides a runtime environment and a set of libraries and tools for building and running applications on Windows operating systems. The framework includes a variety of programming languages, such as C#, F#, and Visual Basic, and supports a range of application types, including desktop, web, mobile, and gaming applications.

1. The .NET Framework includes two main components: the Common Language Runtime (CLR) and the .NET Framework Class Library. The CLR is responsible for managing the execution of code written in any of the supported languages, while the class library provides a large set of pre-built functions and classes that can be used to create a wide range of applications.
2. One of the key advantages of the .NET Framework is its support for a variety of programming languages. This means that developers can choose the language that best fits their needs and expertise, while still being able to use the same set of libraries and tools provided by the framework.
3. Another advantage of the .NET Framework is its support for a variety of application types. The framework includes libraries and tools for creating desktop, web, mobile, and gaming applications, which makes it a versatile choice for developers working on a wide range of projects.
4. The .NET Framework also provides a number of features that help improve the security, reliability, and performance of applications. These include features such as code access security, automatic memory management, and just-in-time (JIT) compilation, which helps improve the speed of application execution.
5. The .NET Framework is also designed to integrate with other Microsoft technologies, such as Microsoft SQL Server, Microsoft SharePoint, and Microsoft Office, which can make it easier to build applications that work seamlessly with other Microsoft products.

Overall, the .NET Framework is a powerful and versatile development platform that provides a wide range of tools and libraries for building and running applications on Windows operating systems.

**.NET**is a software framework that is designed and developed by Microsoft. The first version of the .Net framework was 1.0 which came in the year 2002. In easy words, it is a virtual machine for compiling and executing programs written in different languages like [C#](https://www.geeksforgeeks.org/introduction-to-c/), VB.Net, etc.

It is used to develop Form-based applications, Web-based applications, and Web services. There is a variety of programming languages available on the .Net platform, VB.Net and [C#](https://www.geeksforgeeks.org/introduction-to-c/) being the most common ones. It is used to build applications for Windows, phones, web, etc. It provides a lot of functionalities and also supports industry standards.



**Features of .NET Framework**

**1. Language Interoperability:**Language Interoperability means code written in any language can be used by other languages and there is an access provided to COM components in the System. In other words, this is the ability of code to  interact with code which is written using a different programming language. It can help to maximize code reusability and therefore improve the efficiency of the development process.

**2. Portable:**The .NET Framework provides a common platform to write portable programs using different languages of .NET environment, based on open Internet standards which allow different devices, software’s and applications to work together across a wide and dispersed network of networks.

**3. Type Safety:**During program execution, the type checker ensures that all objects and values, and the references to those objects and values, have a valid type. For example, the type checker ensures that only an integer value is assigned to an integer variable. The type checker also ensures that only valid operations are performed on the objects or values.

**4. Managed Multithreading Support:**Applications running in the CLR can utilize the managed multithreading support. Multithreading requires careful programming For most tasks, you can reduce complexity by queuing requests for execution by thread pool threads.

**5. Performance:**Its CLR environment promotes safe execution of code, eliminates performance bottlenecks through its garbage collector, minimizes software development conflicts through namespaces and avoids versioning conflicts through assembly manifest.

**6. Side by Side Execution:**The NET Framework allows you to deploy multiple versions of an application on a system by using assemblies. Assemblies are the deployment units in the NET Framework. An assembly contains the IL code and metadata. The metadata contains information such as the version of the other assemblies on which the assembly depends. The Common Language Runtime uses the version lo information in the metadata to determine application dependencies and enables you to execute multiple versions of an application side-by-side.

**7. Common Type System:**This provides the necessary data types, values, and object types, which you need to develop applications in different languages. All the NET languages share Common Type System. This implies that a String in Visual Basic .NET is the same as a String in Visual C# or in Visual C++ NET. All the NET languages have access to the same class libraries. In addition, all languages are equally powerful. There is no superior language in .NET i.e. Visual Basic .NET is as powerful as Visual C# or Visual C++ NET.

**8. Parallel Computing:**The NET Framework 4.0 introduces a new programming model for writing multithreaded and asynchronous code that greatly simplifies the work of application and library developers.

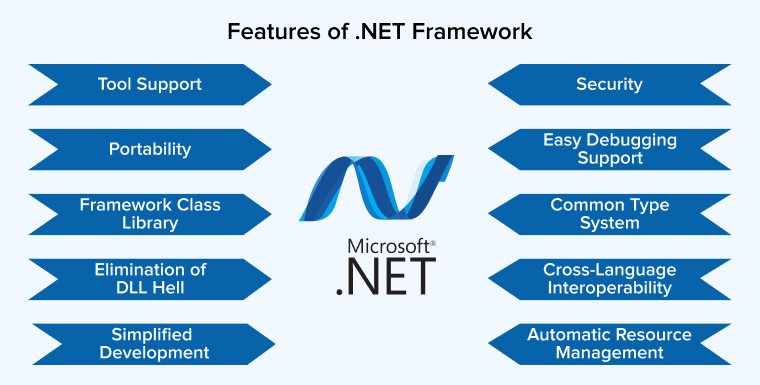
**9. Dynamic Language Runtime:**The dynamic language runtime is a new runtime environment that adds a set of services for dynamic languages to the Common Language Runtime. The CLR makes it easier to develop dynamic languages run on the .NET framework and to add dynamic features to statically typed language.

**10. Automatic memory management:**While developing applications developers had to develop an eye on system resource like memory. ..NET takes this worry away from developer by handling memory on its own. The garbage collector takes care of freeing unused objects at appropriate intervals.

**11.  Language Independence:**Language Independence makes use of the power of various languages in one platform. It means programmer’s can develop their applications in any of the language supported by NET framework .NET framework supports various programming languages such as Microsoft Visual  Basic NET, Microsoft Visual J# NET etc. After Compilation the source code of all these languages converted into Common Intermediate Language (CL) that enhances Language Independence.

**12. Base Class Library:**The Base Class Library (BCL) is a library of functionality available to all languages in the NET Framework. The BCL provides classes that encapsulate a number of common functions including file reading and writing graphic rendering, database interaction, XML document manipulation.

**13. Security:**Windows platform was always criticized for poor security mechanisms. Microsoft has taken great efforts to make .NET platform safe and secure for enterprise applications Features such as type safety, code access security etc.



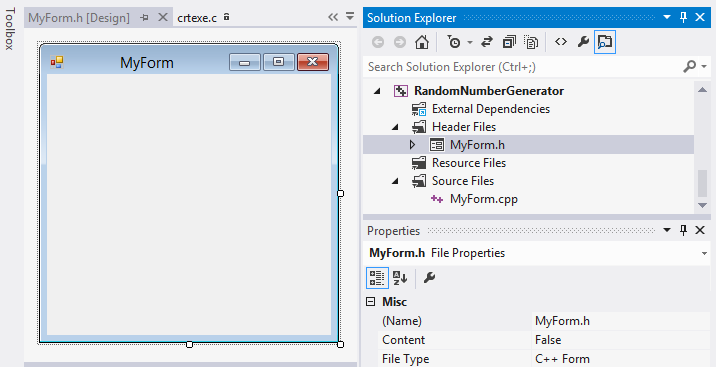
**Window Applications**

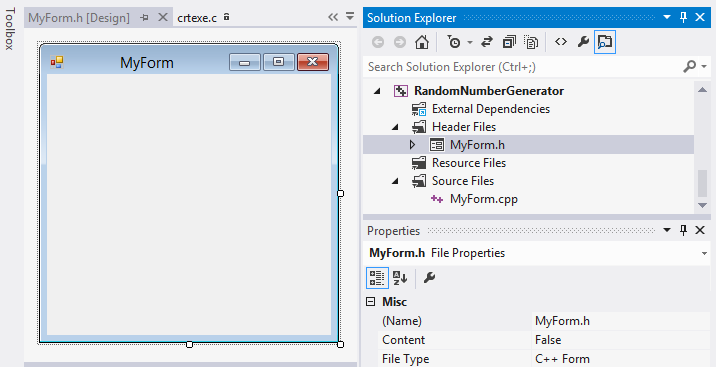
Window Applications (often referred to as Windows applications) are software programs designed to run on Microsoft Windows operating systems. These applications typically have a graphical user interface (GUI) that allows users to interact with them using windows, icons, menus, buttons, and other graphical elements.

Here are some key aspects and features of Windows applications:

1. **Graphical User Interface (GUI)**:
   * Windows applications have a visual interface that allows users to interact with them using graphical elements such as windows, buttons, text boxes, checkboxes, and more.
   * The GUI provides a user-friendly experience, enabling users to perform tasks intuitively without needing to understand complex commands or programming languages.
2. **Event-Driven Programming Model**:
   * Windows applications are often built using an event-driven programming model, where user actions (such as clicking a button or typing in a text box) trigger events that are handled by the application.
   * Developers write code to respond to these events, defining what actions the application should take in response to user input or other events.
3. **Windows Forms**:
   * Windows Forms (WinForms) is a graphical application programming interface (API) included in the .NET Framework for building Windows applications.
   * WinForms provides a set of pre-built controls and components that developers can use to create the user interface of their applications, making it relatively easy to design and customize the appearance of Windows applications.
4. **WPF (Windows Presentation Foundation)**:
   * WPF is a more modern and flexible UI framework for building Windows applications, also included in the .NET Framework.
   * WPF uses a declarative programming model based on Extensible Application Markup Language (XAML) to define the structure and behavior of the user interface, allowing for greater separation of design and logic.
5. **Integration with .NET Framework**:
   * Windows applications built using .NET Framework leverage its features, libraries, and runtime services for tasks such as memory management, data access, security, and more.
   * Developers can use languages such as C#, Visual Basic .NET, or C++/CLI to build Windows applications, taking advantage of the language interoperability provided by the .NET platform.
6. **Deployment**:
   * Windows applications can be distributed and installed on users' computers using various deployment methods such as setup installers, ClickOnce deployment, or Windows Store (for UWP apps).
   * Deployment typically involves packaging the application along with any required dependencies (such as .NET Framework runtime libraries) and installing them on the target system.

Overall, Windows applications provide a familiar and intuitive user experience, making them suitable for a wide range of desktop software needs, including productivity tools, multimedia applications, games, utilities, and more. With the capabilities of modern UI frameworks like WinForms and WPF, developers can create visually appealing and feature-rich Windows applications that meet the needs of their users.Top of Form

****

Bottom of Form

**Chapter 2:**

**Introduction about Project**

**Introduction about ERP SYSTEM Project**

Three types of users can use this ERP system project. These are administrator, student and teacher.

After logging in, the administrator will reach the admin page. There will be options to add, update and delete the details of new students and teachers.

A new student can be added by clicking on add student's details. By clicking on add button the student's details will be added to the database. While adding the student, a message box will be shown indicating that the details have been added.

A new teacher can be added by clicking on add teacher's details. By clicking on add button the teacher's details will be added to the database. While adding the teacher, a message box will be shown indicating that the details have been added.

Data of already registered students and teachers can also be updated and deleted.

Roll no. have been used to make students unique and Teacher id has been used to make teachers unique.

There is also an option of teacher's salary in the admin page. By clicking on it, the salary of the teacher has been given and the salary is pending.

After login the teacher will go to MainManu page. where the option of student details, fees, and teacher details is available.

Teachers can check student details and student fees and can also check their own details.

After logging in, the student will go to the Student Menu page, where he will get only the option of student details and fees.

Students will only have access to view their details and their fees.

Top of Form

**Chapter 3:**

**System Requirements and Specifications**

**Definition**

System requirements outline the technical specifications and configurations necessary for the software or system to operate effectively. They typically include:

1. Hardware Requirements:
   * Specifies the minimum and recommended hardware specifications, such as processor, memory (RAM), storage (hard disk space), and graphics capabilities.
   * Example: Intel Core i5 processor, 8GB RAM, 256GB SSD.
2. Software Requirements:
   * Lists the operating systems, software dependencies, and other supporting software required for the system to function properly.
   * Example: Windows 10 or later, .NET Framework 4.8, Microsoft SQL Server 2019.
3. Network Requirements:
   * Describes the network infrastructure requirements, including internet connectivity, bandwidth, and network protocols.
   * Example: Minimum internet speed of 10 Mbps, support for TCP/IP protocol.
4. Compatibility Requirements:
   * Specifies compatibility with other software, hardware devices, or systems that the software needs to interact with.
   * Example: Integration with Microsoft Office suite, compatibility with popular web browsers like Google Chrome and Mozilla Firefox.

Specifications:

Specifications define the functional and non-functional requirements of the software or system. They detail what the system should do and how it should perform. Specifications typically include:

1. Functional Requirements:
   * Describes specific features, functionalities, and capabilities of the software or system.
   * Example: User authentication, data entry forms, search functionality, reporting features.
2. Non-functional Requirements:
   * Defines the quality attributes, constraints, and performance characteristics of the system.
   * Example: Performance (response time, throughput), reliability, security, scalability, usability, accessibility.
3. User Interface (UI) Design:
   * Specifies the layout, navigation, and visual design of the user interface.
   * Example: Use of color schemes, typography, icons, and layout grids.
4. Data Requirements:
   * Describes the data entities, attributes, relationships, and constraints of the system.
   * Example: Database schema design, data validation rules, data migration requirements.
5. Testing Requirements:
   * Outlines the testing approach, methodologies, and criteria for verifying and validating the system.
   * Example: Unit testing, integration testing, performance testing, user acceptance testing (UAT).
6. Documentation Requirements:
   * Specifies the documentation needed for the system, including user manuals, technical documentation, and training materials.
   * Example: User guides, system architecture diagrams, API documentation, release notes.

By clearly defining system requirements and specifications, stakeholders can ensure that the development team builds a system that meets their needs and expectations. These documents serve as a blueprint throughout the development lifecycle, guiding design, implementation, testing, and deployment efforts.

Top of Form

**Hardware and Software Requirements**

**Hardware Requirements:**

1. **Processor (CPU)**:
   * Minimum: Intel Core i3 or equivalent
   * Recommended: Intel Core i5 or higher
2. **Memory (RAM)**:
   * Minimum: 4GB
   * Recommended: 8GB or higher
3. **Storage (Hard Disk Space)**:
   * Minimum: 20GB available space
   * Recommended: 50GB or higher for development environment and project files
4. **Graphics Card**:
   * Basic graphics card capable of supporting the monitor's resolution
5. **Monitor**:
   * Resolution: 1280x800 pixels or higher
6. **Input Devices**:
   * Keyboard and mouse or other pointing devices

**Software Requirements:**

1. **Operating System**:
   * Windows 7 SP1 or later
   * Windows Server 2012 or later (if developing server-side applications)
2. **Integrated Development Environment (IDE)**:
   * Visual Studio IDE (Community, Professional, or Enterprise edition) for VB.NET development
   * Version: Visual Studio 2017 or later recommended
3. **.NET Framework**:
   * Version: .NET Framework 4.7.2 or later
   * Required for running VB.NET applications
4. **Database Software** (if applicable):
   * Microsoft SQL Server Express Edition or higher
   * MySQL Community Server
   * SQLite
5. **Web Browser**:
   * Recommended for testing web-based applications:
     + Google Chrome
     + Mozilla Firefox
     + Microsoft Edge
6. **Additional Software** (depending on project requirements):
   * Microsoft Office Suite (for integration or automation)
   * Version control software (e.g., Git, SVN) for managing source code
   * Any third-party libraries or frameworks required for the project

**Development Environment:**

1. **Integrated Development Environment (IDE)**:
   * Visual Studio is the primary IDE for VB.NET development, providing features such as code editing, debugging, and project management.
2. **Configuration**:
   * Configure Visual Studio with appropriate settings and extensions for VB.NET development.
   * Install necessary SDKs and tools for building, testing, and deploying VB.NET applications.

By ensuring that your hardware and software meet these requirements, you can create a suitable development environment for VB.NET projects. Additionally, consider scalability and future project needs when determining hardware and software specifications.

Top of Form

**Chapter 4:**

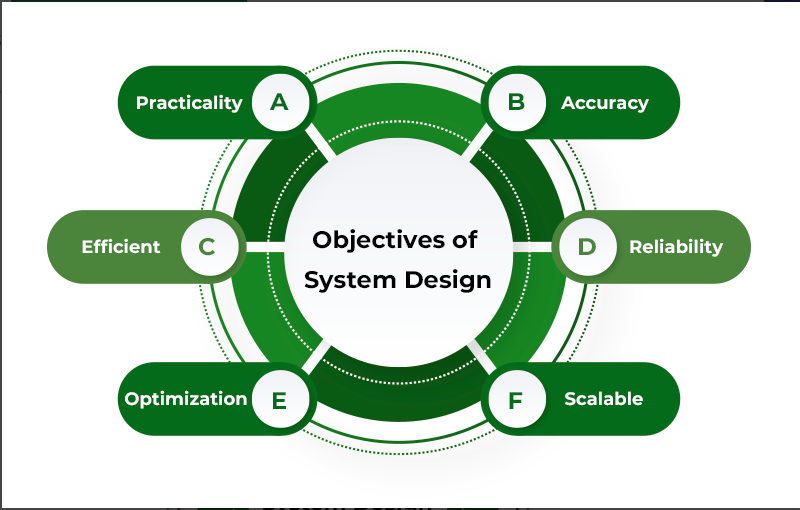
**System Design**

**Definition**

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It involves translating user requirements into a detailed blueprint that guides the implementation phase. The goal is to create a well-organized and efficient structure that meets the intended purpose while considering factors like scalability, maintainability, and performance.

**Objectives of System Design**

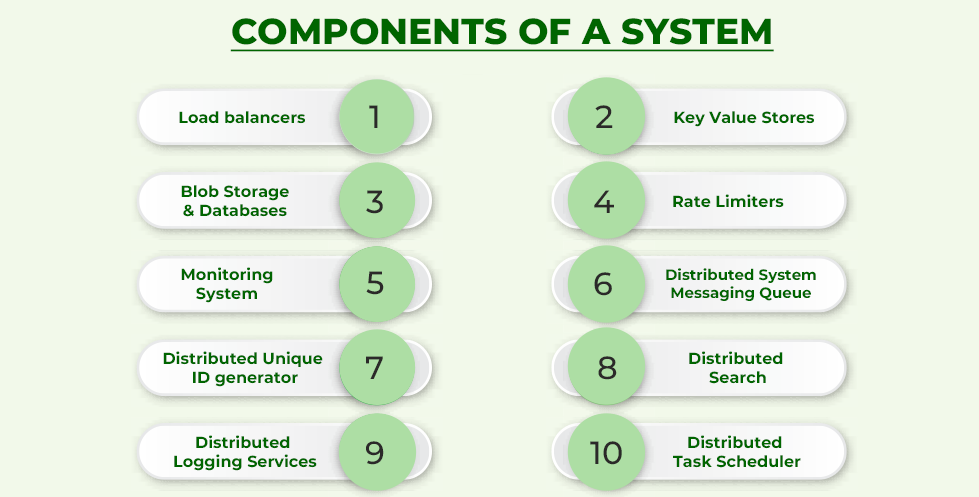
1. **Practicality**: We need a system that should be targetting the set of audiences(users) corresponding to which they are designing.
2. **Accuracy**: Above system design should be designed in such a way it fulfills nearly all requirements around which it is designed be it functional o non-functional requirements.
3. **Completeness**: System design should meet all user requirements
4. **Efficient**: The system design should be such that it should not overuse surpassing the cost of resources nor under use as it will by now we know will result in low thorough put (output) and less response time(latency).
5. **Reliability**: The system designed should be in proximity to a failure-free environment for a certain period of time.
6. **Optimization**: Time and space are just likely what we do for code chunks for individual components to work in a system.
7. **Scalable(flexibility)**: System design should be adaptable with time as per different user needs of customers which we know will keep on changing on time. The best example here out is the well-known firm: Nokia. It is the most important aspect while designing systems and is the result of why 1 of 100 startups succeed over the long run, the best example here out is GeeksforGeeks.



[Components of System Design](https://www.geeksforgeeks.org/what-are-the-components-of-system-design/)

Below are some of the major components of the System Design. discussed in brief. The detailed version of this will be discussed in different posts:

1. **Load balancers:**Most crucial component for scalability, availability, and performance measures for systems.
2. **Key Value Stores:** It is a storage system similar to hashtables where key-value stores are distributed hash tables.
3. **Blob Storage:** Blob stands for binary large objects, as the name suggests is storage for unstructured data such as YouTube, and Netflix.
4. **Databases:** It is an organized collection of data so that they can be easily accessed and modified.
5. **Rate Limiters:** These sets the maximum number of requests a service can fulfill.
6. **Monitoring System:** These are basically software where system administrator monitor infrastructures such as bandwidth, CPU, routers, switches, etc.
7. **Distributed System Messaging Queue:** Transaction medium between producers and consumers.
8. **Distributed Unique ID generator:** In the case of large distributed systems, every moment multiple tasks are occurring so in order to distinguish it assign a tag corresponding to every event.
9. **Distributed Search:** Over every website, crucial information that visitors will seek is put into the search bar.
10. **Distributed Logging Services:** Tracing sequences of events from end to end.
11. **Distributed Task Scheduler:**  Computational resources such as CPU, memory, storage, etc.



**Data Flow Diagram (DFD)**

ADD STUDENT ADD TEACHER

UPDATE STUDENT UPDATE TEACHER

ADMIN

DELETE TEACHER

DELETE STUDENT

SALARY DETAILS

STUDENT DETAILS FEES

TEACHER LOGIN

TEACHER DETAILS

FEES STUDENT DETAILS

STUDENT LOGIN

**Entity-Relation Diagram (ER)**

salary

name id teacherID email

fess

roll no phone no. TEACHER

class

STUDENT

course

admission employee

add

admin teacher

add update

student teacher

delete teacher/student update student

**Chapter 5:**

**Source Code**

**Source Code**

**connection**

Imports MySql.Data.MySqlClient

Public Class connection

Shared Function con()

Dim conn As New MySqlConnection("host=localhost; user=root; password=; database=erp")

Return conn

End Function

End Class

**Admin**

Imports MySql.Data.MySqlClient

Public Class Admin

Private Sub Admin\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim con As MySqlConnection = connection.con

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

first.Show()

first.TextBox3.Text = "student"

Me.Hide()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

first.Show()

first.TextBox3.Text = "teacher"

Me.Hide()

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

first.Show()

first.TextBox3.Text = "admin"

Me.Hide()

End Sub

End Class

**LOGIN PAGE**

Imports MySql.Data.MySqlClient

Public Class first

Dim con As MySqlConnection = connection.con

Dim cmd As Object

Dim com As MySqlCommand

Private Sub Form1\_Closing(sender As Object, e As EventArgs) Handles MyBase.FormClosing

Admin.Show()

End Sub

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

If con.State = ConnectionState.Closed Then

con.Open()

End If

cmd = "SELECT \* FROM logins WHERE username=@username AND Pass=@passw AND Type=@typ"

com = New MySqlCommand(cmd, con)

com.Parameters.AddWithValue("username", TextBox1.Text)

com.Parameters.AddWithValue("passw", TextBox2.Text)

com.Parameters.AddWithValue("typ", TextBox3.Text)

Dim res As Integer = Convert.ToInt32(com.ExecuteScalar())

If res > 0 Then

Dim usertype As String = TextBox3.Text

con.Close()

If usertype = "admin" Then

adminpage.Show()

Me.Hide()

ElseIf usertype = "teacher" Then

mainmenu.Show()

Me.Hide()

ElseIf usertype = "student" Then

studentmenu.Show()

Me.Hide()

Else

MsgBox("Invalid User Type!", MsgBoxStyle.Information, "Information")

End If

Else

MsgBox("Please check your username or password!", MsgBoxStyle.Critical, "Error!")

End If

End Sub

End Class

**Admin page**

Imports MySql.Data.MySqlClient

Public Class adminpage

Dim con As MySqlConnection = connection.con

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

salary.Show()

Me.Hide()

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

addstudent.Show()

Me.Hide()

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

addteacher.Show()

Me.Hide()

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

updatestudent.Show()

Me.Hide()

End Sub

Private Sub Button5\_Click(sender As Object, e As EventArgs) Handles Button5.Click

updateteacher.Show()

Me.Hide()

End Sub

Private Sub Button6\_Click(sender As Object, e As EventArgs) Handles Button6.Click

deletestudent.Show()

Me.Hide()

End Sub

Private Sub Button7\_Click(sender As Object, e As EventArgs) Handles Button7.Click

deleteteacher.Show()

Me.Hide()

End Sub

Private Sub adminpage\_FormClosing(sender As Object, e As EventArgs) Handles MyBase.FormClosing

Admin.Show()

End Sub

Private Sub adminpage\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

End Class

**Checkfees**

Imports MySql.Data.MySqlClient

Public Class checkfess

Private Sub checkfess\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim con As MySqlConnection = connection.con

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

End Class

**Addteacher**

Imports MySql.Data.MySqlClient

Public Class addteacher

Dim con As MySqlConnection = connection.con

Dim cmd As Object

Dim com As MySqlCommand

Private Sub addteacher\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

cmd = "INSERT INTO addteacher(TEACHERID,NAME,DOB,CONTACTNO,EMAILID,FATHERNAME,FATHEROCCUPATION,MOTHERNAME,MOTHEROCCUPATION,ADDRESS,HIGHESTQULIFICATION,SUBJECTSNAME,EXPERIENCE,PAYSCALE,AADHARCARDNO,SPECILIZATION) VALUES(@tid,@name,@dob,@cono,@id,@fname,@foccu,@mname,@moccu,@addr,@hq,@sname,@exp,@pay,@aadrar,@specil)"

com = New MySqlCommand(cmd, con)

com.Parameters.AddWithValue("tid", TextBox6.Text)

com.Parameters.AddWithValue("name", TextBox1.Text)

com.Parameters.AddWithValue("dob", DateTimePicker1.Text)

com.Parameters.AddWithValue("cono", TextBox3.Text)

com.Parameters.AddWithValue("id", TextBox22.Text)

com.Parameters.AddWithValue("fname", TextBox21.Text)

com.Parameters.AddWithValue("foccu", TextBox20.Text)

com.Parameters.AddWithValue("mname", TextBox19.Text)

com.Parameters.AddWithValue("moccu", TextBox16.Text)

com.Parameters.AddWithValue("addr", TextBox4.Text)

com.Parameters.AddWithValue("hq", TextBox5.Text)

com.Parameters.AddWithValue("sname", TextBox12.Text)

com.Parameters.AddWithValue("exp", TextBox18.Text)

com.Parameters.AddWithValue("pay", TextBox13.Text)

com.Parameters.AddWithValue("aadrar", TextBox14.Text)

com.Parameters.AddWithValue("specil", TextBox15.Text)

If com.ExecuteNonQuery Then

MsgBox("teacher added Success")

Else

MsgBox("Error!")

End If

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then

PictureBox1.ImageLocation = OpenFileDialog1.FileName

PictureBox1.SizeMode = PictureBoxSizeMode.StretchImage

End If

End Sub

End Class

**Delete student**

Imports MySql.Data.MySqlClient

Public Class deletestudent

Dim con As MySqlConnection = connection.con

Dim cmd As Object

Dim com As MySqlCommand

Private Sub deletestudent\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

Private Sub delete1\_Click(sender As Object, e As EventArgs) Handles delete1.Click

If con.State = ConnectionState.Closed Then

con.Open()

End If

cmd = "DELETE FROM addnewstudent WHERE ROLLNO=@rln"

com = New MySqlCommand(cmd, con)

com.Parameters.AddWithValue("rln", TextBox1.Text)

If com.ExecuteNonQuery Then

MsgBox("Student deleted Success")

Else

MsgBox("Student not exist!")

End If

End Sub

End Class

**Delete teacher**

Imports MySql.Data.MySqlClient

Public Class deleteteacher

Dim con As MySqlConnection = connection.con

Dim cmd As Object

Dim com As MySqlCommand

Private Sub deleteteacher\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

Private Sub delete2\_Click(sender As Object, e As EventArgs) Handles delete2.Click

If con.State = ConnectionState.Closed Then

con.Open()

End If

cmd = "DELETE FROM addteacher WHERE TEACHERID=@tid"

com = New MySqlCommand(cmd, con)

com.Parameters.AddWithValue("tid", TextBox1.Text)

If com.ExecuteNonQuery Then

MsgBox("Teacher deleted Success")

Else

MsgBox("Teacher not exist!")

End If

End Sub

End Class

**fees**

Imports MySql.Data.MySqlClient

Public Class fees

Dim con As MySqlConnection = connection.con

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

rollno.Show()

Me.Hide()

End Sub

Private Sub fees\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

End Class

**Feeslip**

Imports MySql.Data.MySqlClient

Public Class feeslip

Private Sub feeslip\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim con As MySqlConnection = connection.con

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

End Class

**Mainmenu**

Imports MySql.Data.MySqlClient

Public Class mainmenu

Private Sub Form2\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim con As MySqlConnection = connection.con

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

Private Sub Button5\_Click(sender As Object, e As EventArgs) Handles Button5.Click

first.Show()

Me.Close()

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

fees.Show()

Me.Close()

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

student.Show()

Me.Close()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs)

salary.Show()

Me.Close()

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

teacher.Show()

Me.Close()

End Sub

End Class

**Rollno**

Imports MySql.Data.MySqlClient

Public Class rollno

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

feeslip.Show()

Me.Hide()

MsgBox("fess deposited sucessfully")

End Sub

Private Sub rollno\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim con As MySqlConnection = connection.con

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

End Class

**Salary**

Imports MySql.Data.MySqlClient

Public Class salary

Private Sub Form3\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim con As MySqlConnection = connection.con

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

MsgBox("salary paid")

End Sub

End Class

**Student**

Imports MySql.Data.MySqlClient

Public Class student

Dim con As MySqlConnection = connection.con

Dim cmd As Object

Dim com As MySqlCommand

Private Sub Form7\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim con As MySqlConnection = connection.con

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

End Class

**Student menu**

Imports MySql.Data.MySqlClient

Public Class studentmenu

Dim con As MySqlConnection = connection.con

Dim cmd As Object

Dim com As MySqlCommand

Private Sub LineShape4\_Click(sender As Object, e As EventArgs)

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

If con.State = ConnectionState.Closed Then

con.Open()

End If

student.Show()

Me.Hide()

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

checkfess.Show()

Me.Hide()

End Sub

Private Sub teachermenu\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim con As MySqlConnection = connection.con

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

End Class

**Teacher**

Imports MySql.Data.MySqlClient

Public Class teacher

Private Sub teacher\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim con As MySqlConnection = connection.con

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

End Class

**Update student**

Imports MySql.Data.MySqlClient

Public Class updatestudent

Dim con As MySqlConnection = connection.con

Dim cmd As Object

Dim com As MySqlCommand

Private Sub updatestudent\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

Private Sub updatestudent\_FormClosing(sender As Object, e As EventArgs) Handles MyBase.FormClosing

'adminpage.Show()

'Me.Close()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

If con.State = ConnectionState.Closed Then

con.Open()

End If

cmd = "Update addnewstudent SET NAME=@name,CLASS=@cls,DOB=@dob,FATHERNAME=@fname,FATHEROCCUPATION=@foccu,MOTHERNAME=@mname,MOTHEROCCUPATION=@moccu,PHONENO=@phn,ADDRESS=@adrs,SUBJECTNAME=@sub,LASTPASSINGYEAR=@lpy,HIGHESTQULIFICATION=@hq,BLOODGROUP=@bg,LASTYEARPASINGMARKS=@lpym,TOTALFEES=@totf,AADHARCARDNO=@acn WHERE ROLLNO=@rln"

com = New MySqlCommand(cmd, con)

com.Parameters.AddWithValue("name", TextBox1.Text)

com.Parameters.AddWithValue("cls", TextBox2.Text)

com.Parameters.AddWithValue("dob", TextBox22.Text)

com.Parameters.AddWithValue("fname", TextBox21.Text)

com.Parameters.AddWithValue("foccu", TextBox20.Text)

com.Parameters.AddWithValue("mname", TextBox19.Text)

com.Parameters.AddWithValue("moccu", TextBox16.Text)

com.Parameters.AddWithValue("phn", TextBox4.Text)

com.Parameters.AddWithValue("adrs", TextBox5.Text)

com.Parameters.AddWithValue("sub", TextBox12.Text)

com.Parameters.AddWithValue("lpy", TextBox18.Text)

com.Parameters.AddWithValue("hq", TextBox13.Text)

com.Parameters.AddWithValue("bg", TextBox14.Text)

com.Parameters.AddWithValue("lpym", TextBox15.Text)

com.Parameters.AddWithValue("totf", TextBox24.Text)

com.Parameters.AddWithValue("acn", TextBox23.Text)

com.Parameters.AddWithValue("rln", TextBox3.Text)

If com.ExecuteNonQuery Then

MsgBox("Updated Success")

Else

MsgBox("Error!")

End If

con.Close()

End Sub

Private Sub TextBox3\_TextChanged(sender As Object, e As EventArgs) Handles TextBox3.TextChanged

If con.State = ConnectionState.Closed Then

con.Open()

End If

cmd = "SELECT \* FROM addnewstudent WHERE ROLLNO=@rln"

com = New MySqlCommand(cmd, con)

com.Parameters.AddWithValue("rln", TextBox3.Text)

Dim status As Integer = Convert.ToInt32(com.ExecuteScalar)

If (status > 0) Then

Dim res As Object = com.ExecuteReader

res.Read()

TextBox1.Text = res.Item(1)

TextBox2.Text = res.Item(2)

TextBox16.Text = res.Item(8)

TextBox19.Text = res.Item(7)

TextBox20.Text = res.Item(6)

TextBox21.Text = res.Item(5)

TextBox22.Text = res.Item(4)

TextBox4.Text = res.Item(9)

TextBox5.Text = res.Item(10)

TextBox12.Text = res.Item(11)

TextBox13.Text = res.Item(13)

TextBox14.Text = res.Item(14)

TextBox15.Text = res.Item(15)

TextBox18.Text = res.Item(12)

TextBox23.Text = res.Item(17)

TextBox24.Text = res.Item(16)

Else

clearfields()

End If

con.Close()

End Sub

Private Sub clearfields()

TextBox1.Clear()

TextBox2.Clear()

TextBox16.Clear()

TextBox19.Clear()

TextBox20.Clear()

TextBox21.Clear()

TextBox22.Clear()

TextBox4.Clear()

TextBox5.Clear()

TextBox12.Clear()

TextBox13.Clear()

TextBox14.Clear()

TextBox15.Clear()

TextBox18.Clear()

TextBox23.Clear()

TextBox24.Clear()

End Sub

End Class

**Update teacher**

Imports MySql.Data.MySqlClient

Public Class updateteacher

Dim con As MySqlConnection = connection.con

Dim cmd As Object

Dim com As MySqlCommand

Private Sub updateteacher\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim con As MySqlConnection = connection.con

If con.State = ConnectionState.Closed Then

con.Open()

End If

End Sub

Private Sub updateteacher\_FormClosing(sender As Object, e As EventArgs) Handles MyBase.FormClosing

Admin.Show()

Me.Close()

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

If con.State = ConnectionState.Closed Then

con.Open()

End If

cmd = "UPDATE addteacher SET NAME=@name,DOB=@dob,CONTACTNO=@cono,EMAILID=@id,FATHERNAME=@fname,FATHEROCCUPATION=@foccu,MOTHERNAME=@mname,MOTHEROCCUPATION=@moccu,ADDRESS=@addr,HIGHESTQULIFICATION=@hq,SUBJECTSNAME=@sname,EXPERIENCE=@exp,PAYSCALE=@pay,AADHARCARDNO=@aadrar,SPECILIZATION=@specil WHERE TEACHERID=@tid"

com = New MySqlCommand(cmd, con)

com.Parameters.AddWithValue("name", TextBox1.Text)

com.Parameters.AddWithValue("dob", DateTimePicker1.Text)

com.Parameters.AddWithValue("cono", TextBox3.Text)

com.Parameters.AddWithValue("id", TextBox22.Text)

com.Parameters.AddWithValue("fname", TextBox21.Text)

com.Parameters.AddWithValue("foccu", TextBox20.Text)

com.Parameters.AddWithValue("mname", TextBox19.Text)

com.Parameters.AddWithValue("moccu", TextBox16.Text)

com.Parameters.AddWithValue("addr", TextBox4.Text)

com.Parameters.AddWithValue("hq", TextBox5.Text)

com.Parameters.AddWithValue("sname", TextBox12.Text)

com.Parameters.AddWithValue("exp", TextBox18.Text)

com.Parameters.AddWithValue("pay", TextBox13.Text)

com.Parameters.AddWithValue("aadrar", TextBox14.Text)

com.Parameters.AddWithValue("specil", TextBox15.Text)

com.Parameters.AddWithValue("tid", TextBox6.Text)

If com.ExecuteNonQuery Then

MsgBox("Updated Success")

Else

MsgBox("Error!")

End If

con.Close()

End Sub

End Class

**Chapter 6:**

**Testing and Implementation**

**Testing Definition and Importance**

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

It depends on the process and the associated stakeholders of the project(s). In the IT industry, large companies have a team with responsibilities to evaluate the developed software in context of the given requirements. Moreover, developers also conduct testing which is called **Unit Testing**. In most cases, the following professionals are involved in testing a system within their respective capacities −

* Software Tester
* Software Developer
* Project Lead/Manager
* End User

Different companies have different designations for people who test the software on the basis of their experience and knowledge such as Software Tester, Software Quality Assurance Engineer, QA Analyst, etc.

Testing is done in different forms at every phase

* During the requirement gathering phase, the analysis and verification of requirements are also considered as testing.
* Reviewing the design in the design phase with the intent to improve the design is also considered as testing.
* Testing performed by a developer on completion of the code is also categorized as testing.

**WHY SOFTWARE TESTING IS IMPORTANT?**

**Software testing saves money**

There's a lot of importance of software testing. Cost-effectiveness is one of the top reasons why companies should go for [software and automation testing services](https://www.indiumsoftware.com/software-testing-services/). It helps avoid the extra costs that occur to fix issues after the product is released to the market.

‍

**Security**

It is reason why software testing is important as well as [automation testing](https://testsigma.com/blog/how-to-choose-the-right-open-source-test-automation-tool/), both should be taken into consideration. It is considered to be the most sensitive part. There are a bunch of situations in which the information and details of the users are stolen and they are used for the benefits. It is considered to be the reason why people look for tested and reliable products. As a specific product undergoes testing, the user can be ensured that they are going to receive a reliable product. Testing makes products more vulnerable.

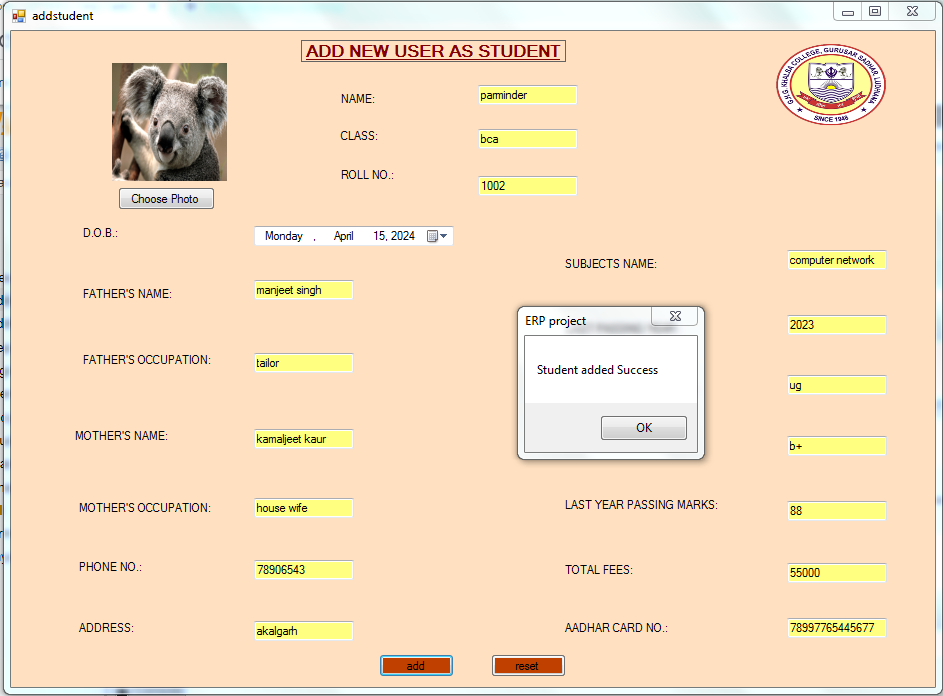
**Customer satisfaction**

The reason why testing is important is because it makes sure that the software is user-friendly and as per the user expectations. That makes it capable of being used by the customers it is intended for. Those who expertise in [software application pentesting](https://www.getastra.com/blog/security-audit/software-penetration-testing/) are familiar with the needs of customers, and unless the software can satisfy a customer’s needs, it would be a practically useless investment. Different kinds of software have different kinds of customers. That’s why just like developers, testers also tend to specialise in certain kinds of software designs. That’s what makes software testing all the more resourceful in gaining customer confidence.

**Performance**

If the performance of the software is low, you will find that it brings your reputation down in the market. Users are not going to trust any people that's why testing is important. There are chances that the reputation of your organization is going to suffer. Software testing also helps in determining the performance of the Product.

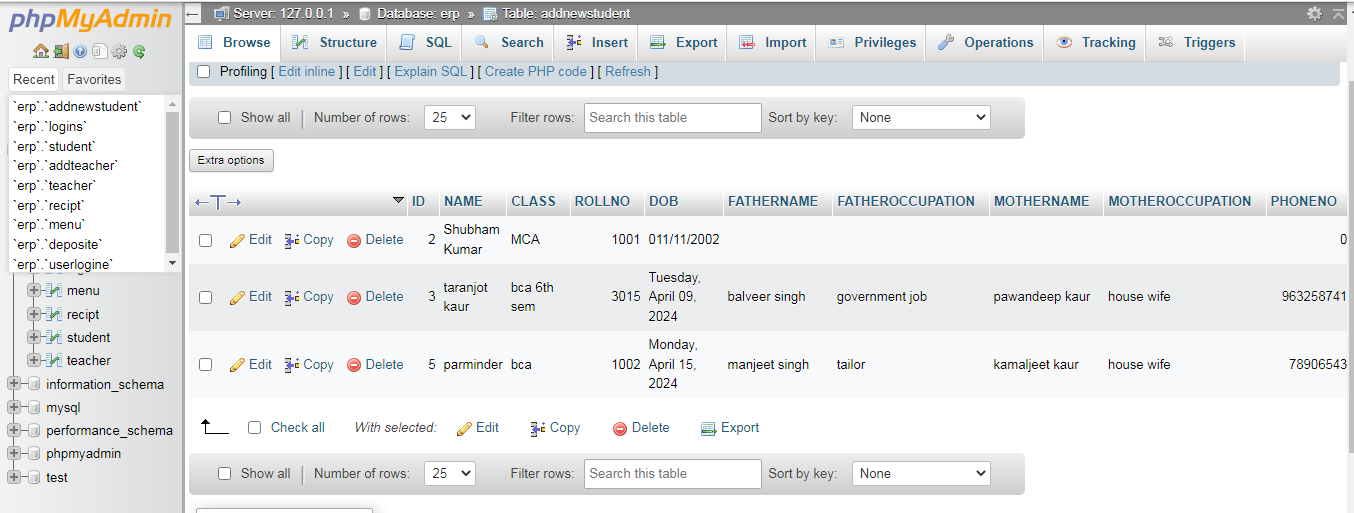
**Testing Report**

****

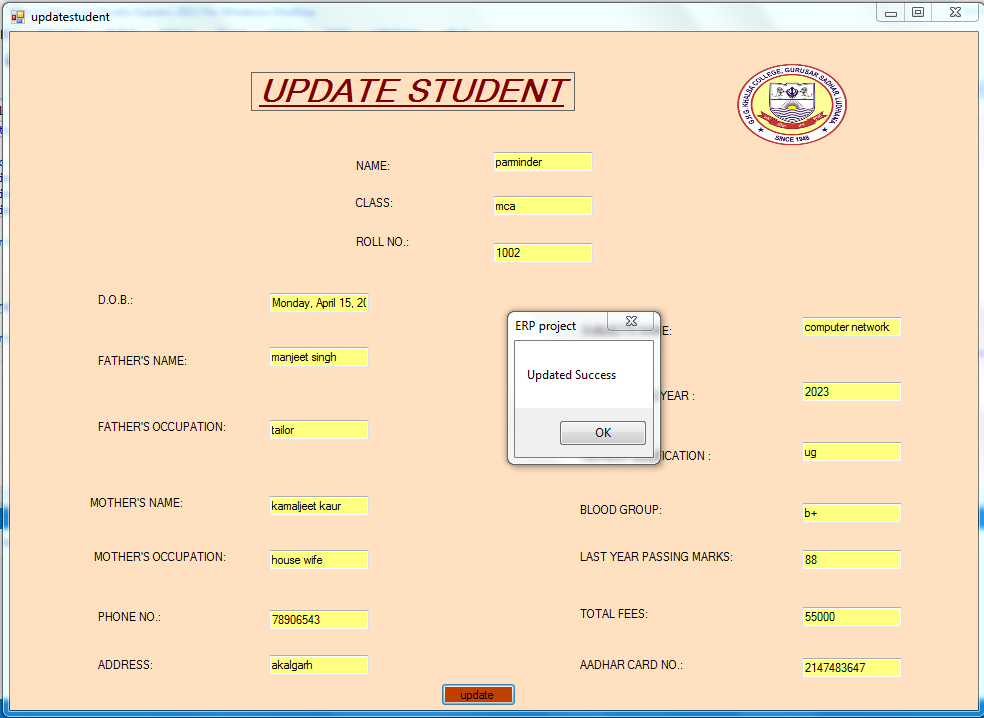
Details of students is added successfully in database.

****

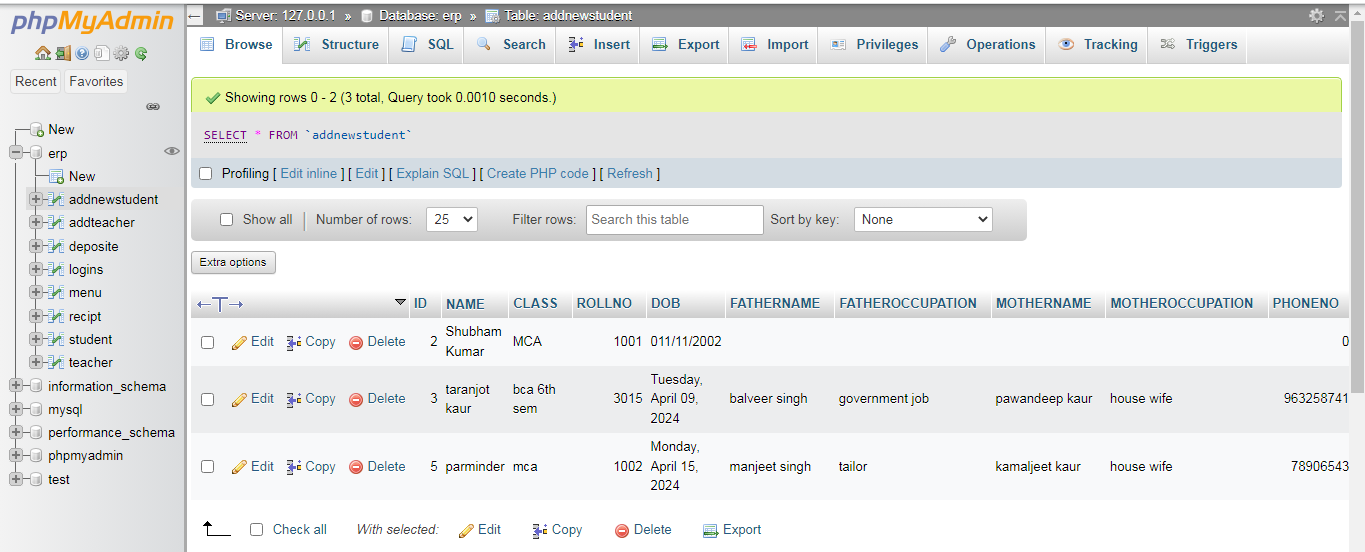
Details of teacher is added successfully in database.



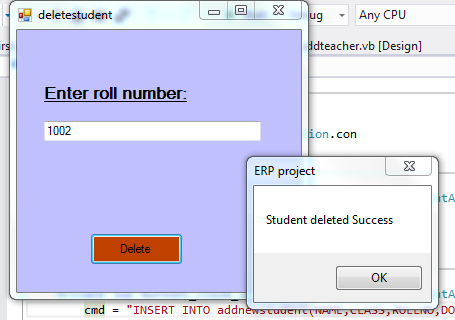
Record in database.



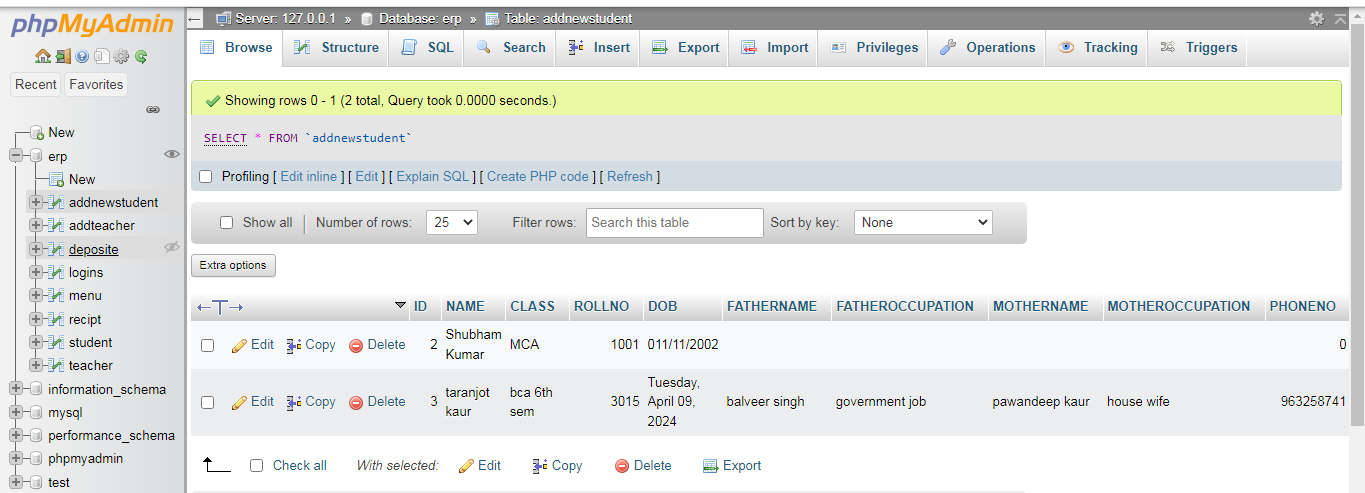
Details of students is updated successfully in database.

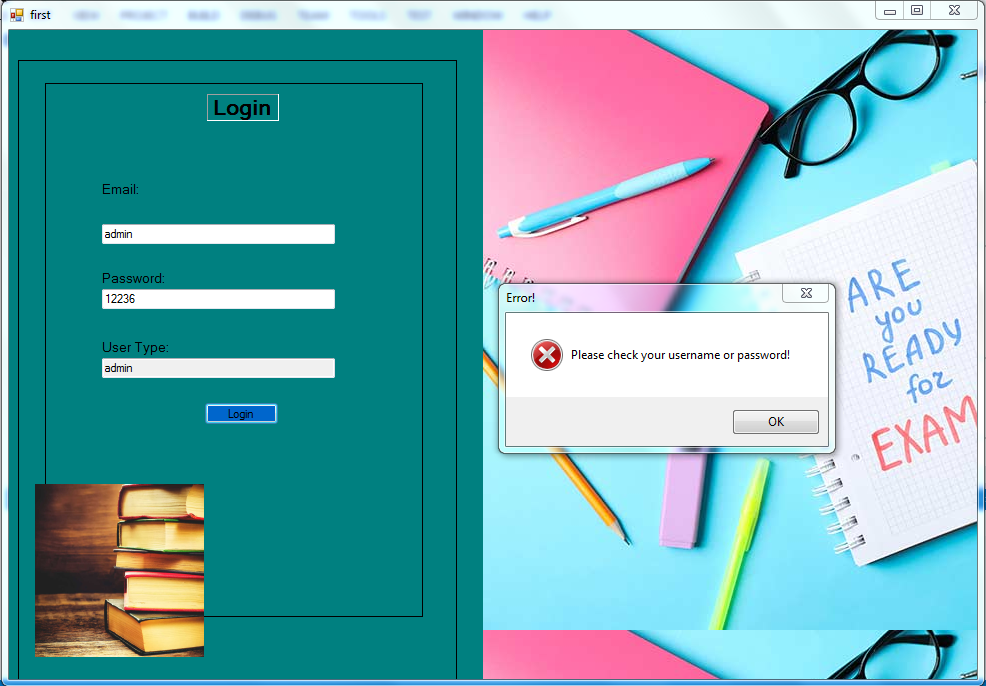


Record in database.



Record of students is deleted successfully from the database.





When we try to login with wrong id or password, A message box will display with message of please check your username or password .

**Chapter 7:**

**System Implementation**

**Definition of Implementation**

Software implementation is the process of taking on a new software application and incorporating it into existing business workflows.

The process may be small, applying only to a certain department or company-wide, and involve significant change management as part of an overall digital transformation strategy.

Software may be developed in-house and rolled out, or an external application may be purchased and used to replace outdated or inefficient workflows.

Software implementation is becoming more widespread – Gartner has forecast worldwide IT spending will grow 4.3% in 2023 as CIOs focus on technologies that enable automation and efficiency.

It’s time to program codes for the project. The implementation process takes huge importance itself. Sometimes it is time worthy step to determine step by a step implementation plan. To simplify the process, the work needs to go through with different modules and units. In the software development process, implementation is the longest process although. When developers have got the design prepared in their hands, they take steps to implementation. Following all the previous steps carefully will help you to implement the best result and make the program productive as well.

When the code gets ready in the implementation phase, then it needs to be tested in the testing phase. In the testing stage, the code prepared in the design step is checked against the specifications to ensure that the program answers all the issues for which it is created. All kinds of functional testing like,

* Integration testing
* Unit testing
* System testing
* Acceptance testing
* Design implementation testing

Are captured in this step. Software testing is a significant and critical step in software development. It finds out if any errors need to be recognized. Before deployment, a program needs multiple testing for errors and deficiencies. It is highly necessary to address these problems before the product meets the original specifications.

**Deployment of Project**

Deploying a project typically involves taking the codebase or application you've developed and making it accessible and operational for its intended users. Here's a general guide on how to deploy a project:

1. **Prepare Your Code**: Ensure that your code is ready for deployment. This includes thorough testing to catch any bugs and optimizing performance where necessary.
2. **Choose a Hosting Provider**: Decide where you want to host your project. This could be on your own servers, cloud platforms like AWS, Google Cloud Platform, or Microsoft Azure, or specialized hosting services like Heroku or Netlify for web applications.
3. **Set Up Your Environment**: Depending on your hosting choice, you'll need to set up your environment. This might involve configuring servers, installing dependencies, and setting up databases or other services your project relies on.
4. **Configure Your Deployment**: Configure your deployment settings. This includes things like specifying which branch or version of your code to deploy, setting environment variables, and configuring any necessary build steps.
5. **Deploy Your Code**: Depending on your setup, this might involve pushing your code to a remote repository (e.g., GitHub, GitLab) and triggering a deployment pipeline, running a deployment script, or using a deployment tool provided by your hosting service.
6. **Monitor and Test**: After deployment, monitor your application to ensure it's running smoothly. Set up monitoring tools to track performance, errors, and usage metrics. Perform thorough testing to catch any issues that may have arisen during deployment.
7. **Scale if Necessary**: If your project experiences increased demand, you may need to scale your infrastructure to handle it. This could involve adding more servers, optimizing code for better performance, or using auto-scaling features provided by your hosting provider.
8. **Regular Maintenance**: Regularly update your codebase and dependencies to ensure security and stability. Monitor for any vulnerabilities and apply patches as necessary. Additionally, consider implementing features based on user feedback to improve the overall user experience.

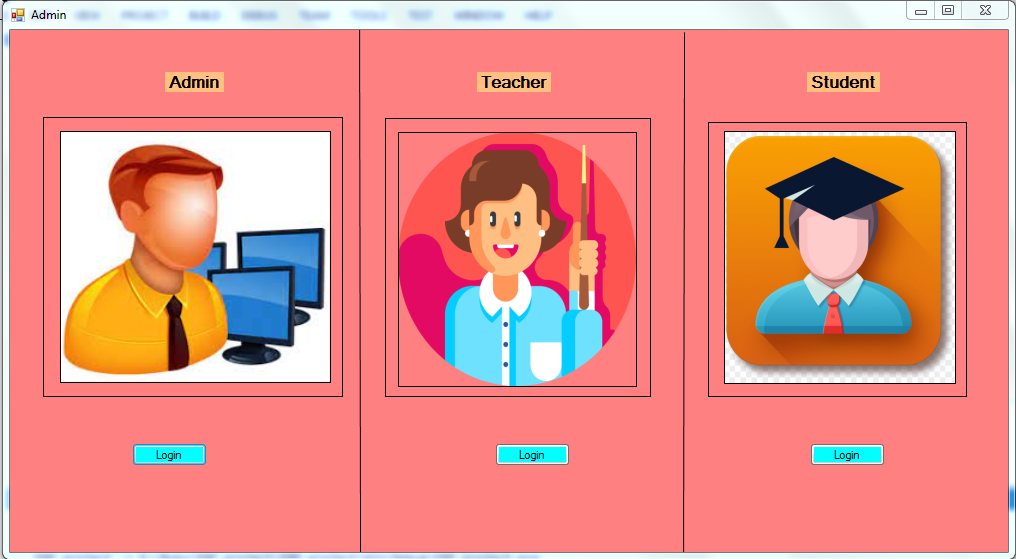
Remember that the deployment process can vary significantly depending on the specifics of your project and your chosen hosting environment. It's essential to thoroughly understand the requirements and best practices for your particular situation.

Top of Form

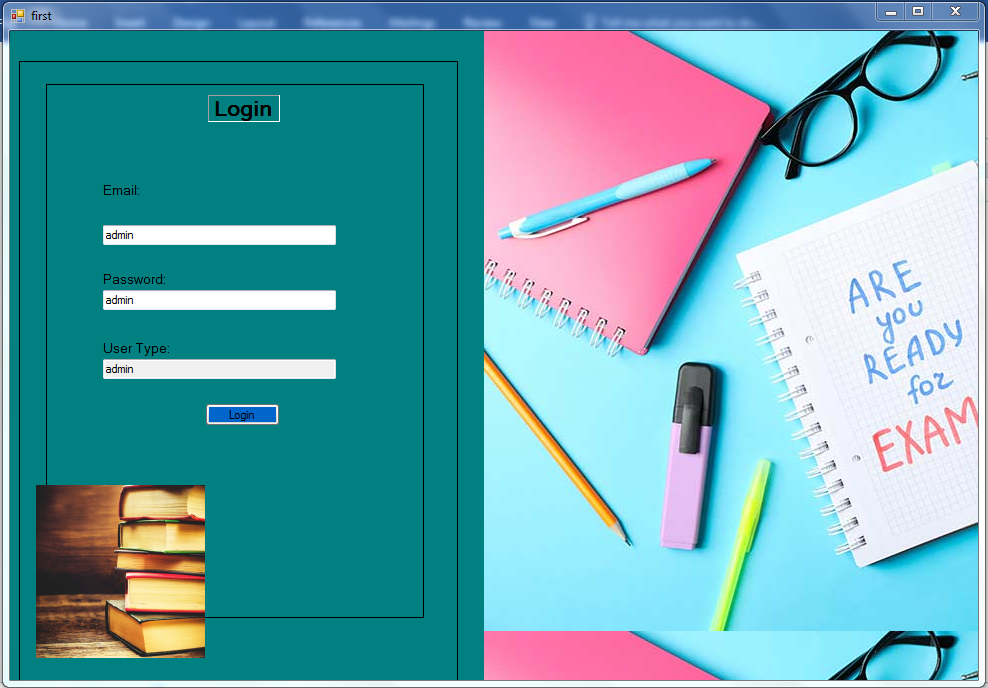
**Chapter 8:**

**Output and Future Scope**

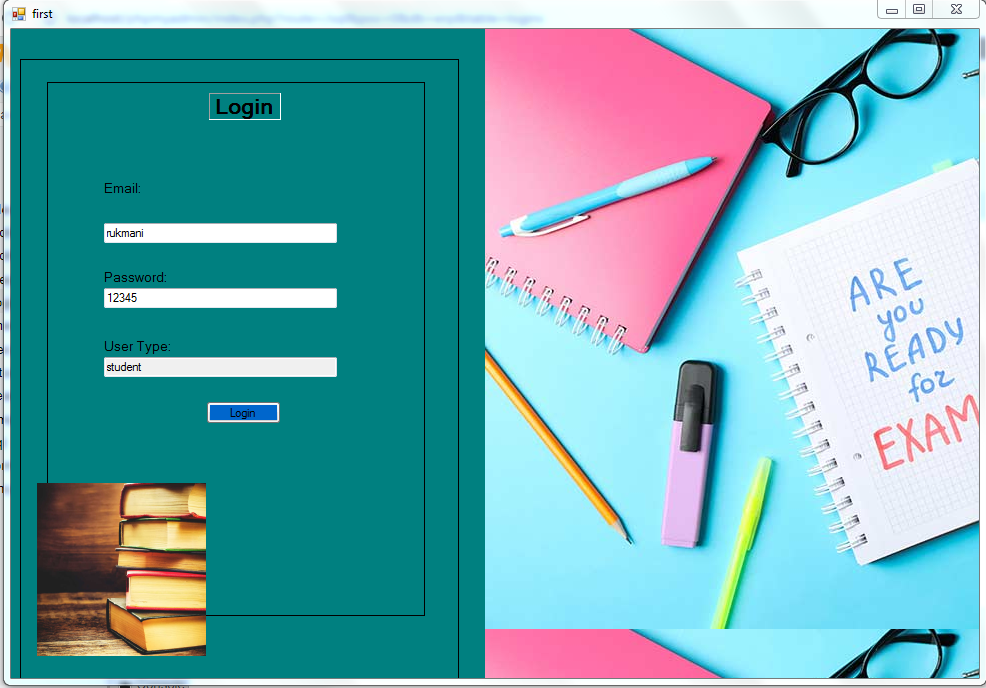
**Output Screenshots**



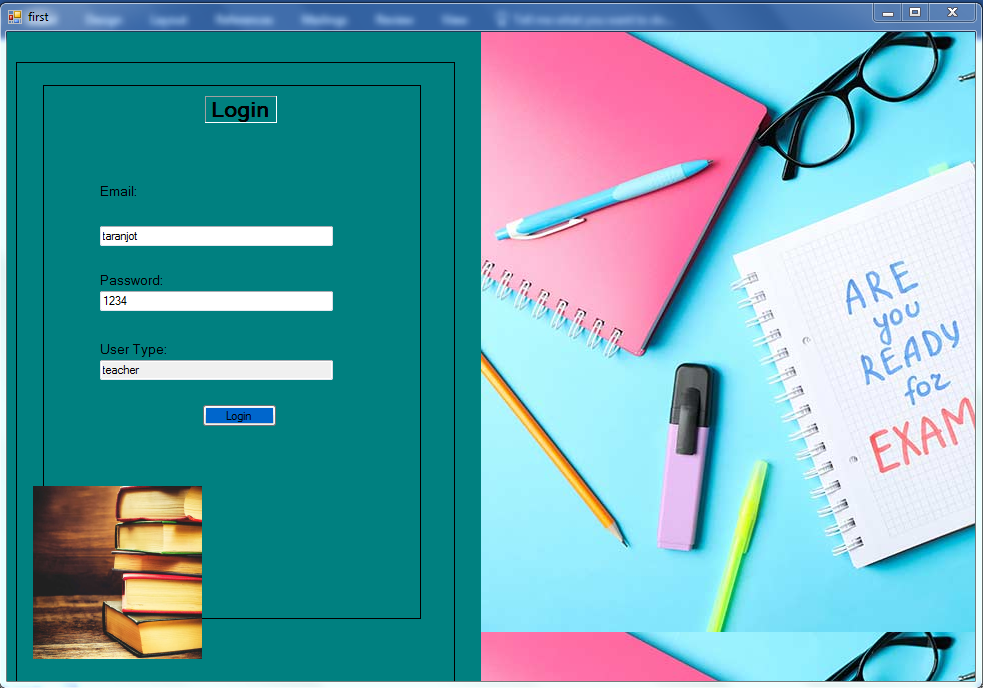
This admin page is used for selection of user type.



This is admin login page .

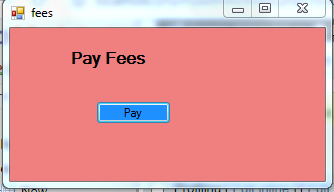


This is Student login page



This is Teacher login page

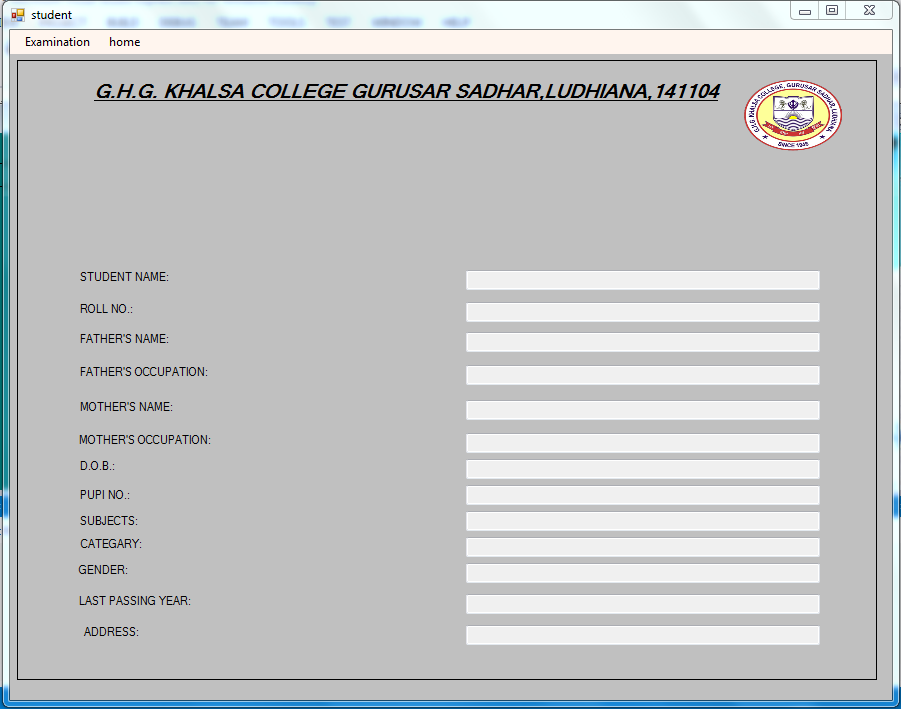
This is main menu page used by teacher after login



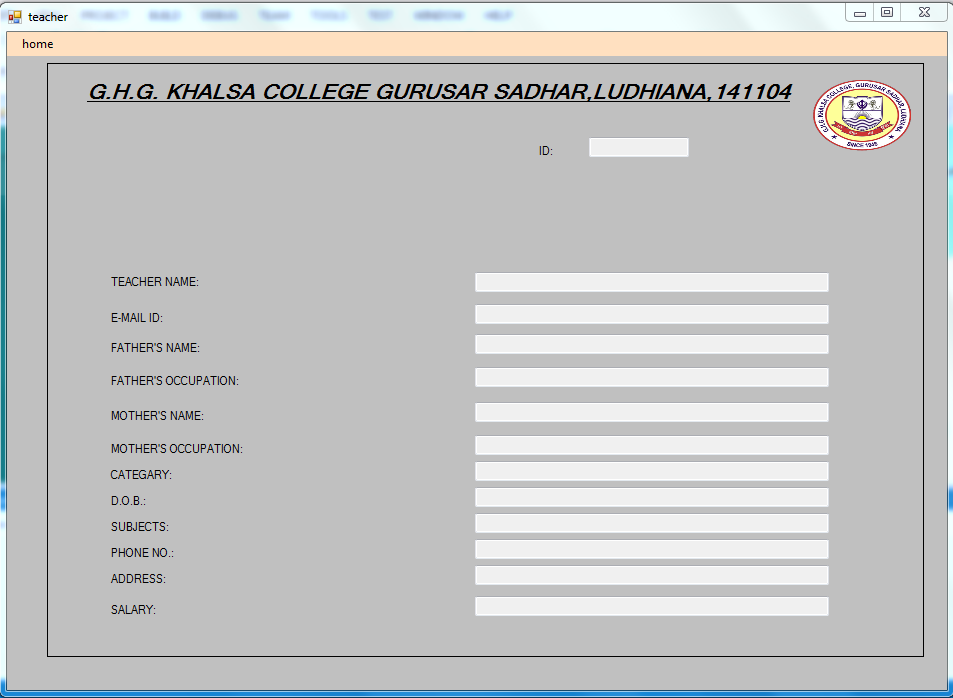
This is Fess page which is used by student to pay fess



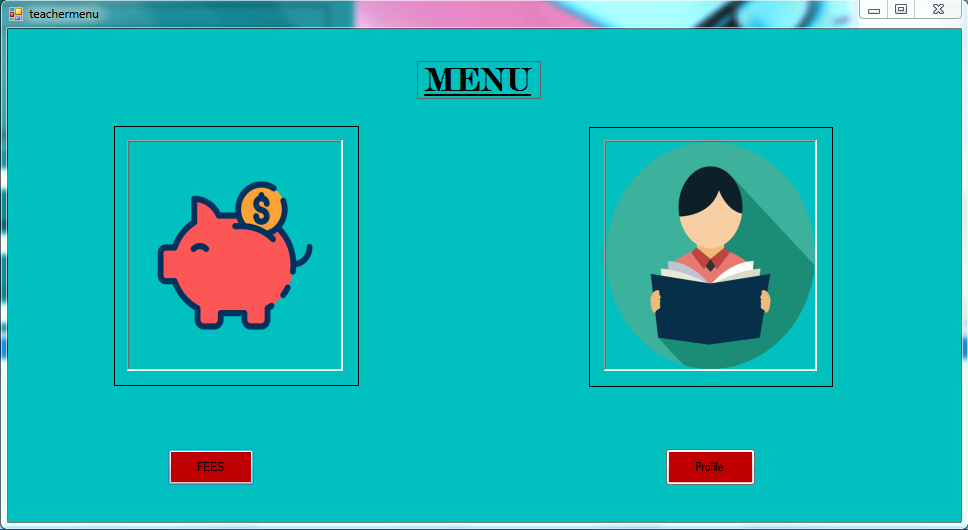
This is roll no page ,this page will show students fees details after entering roll no.



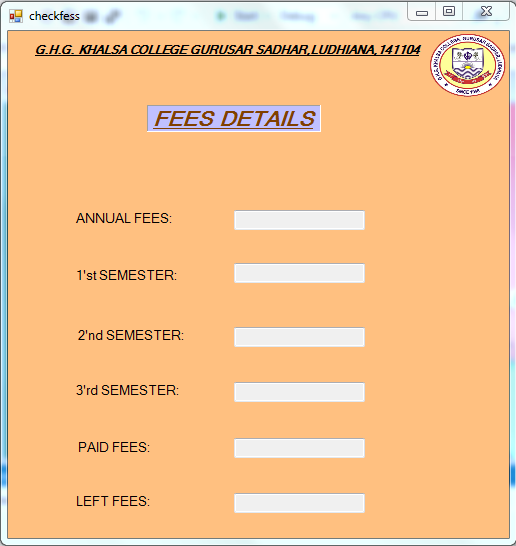
This is student page ,It is used to display Student details



This is teacher page .It is used to display teachers details



This is student menu page .It is student main menu.



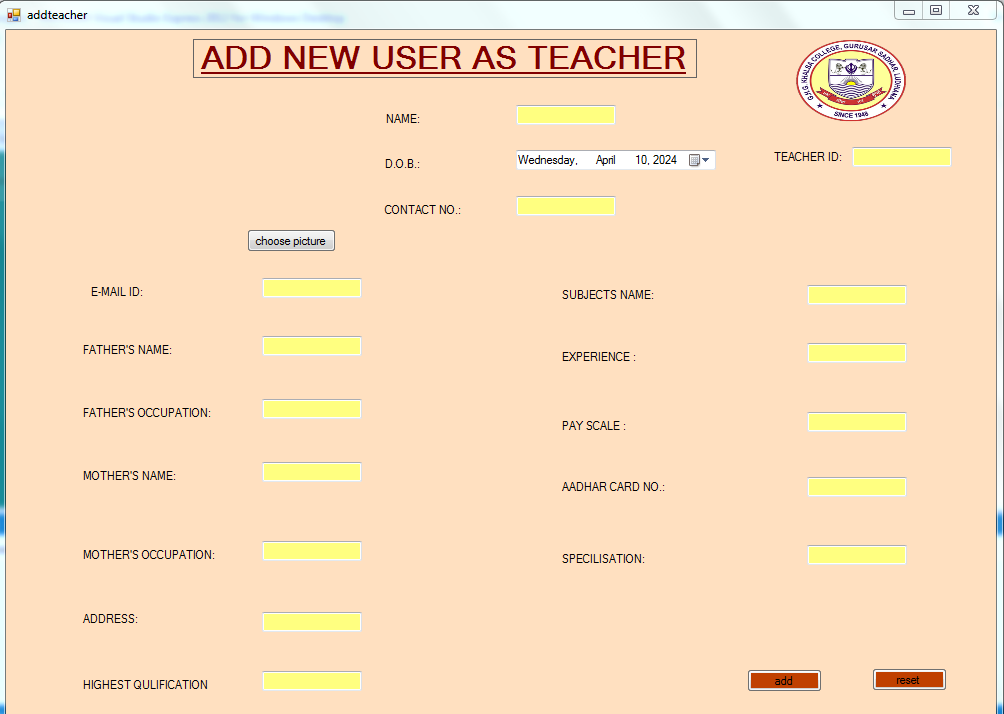
This is checkfees page .It is used to display fess details of student



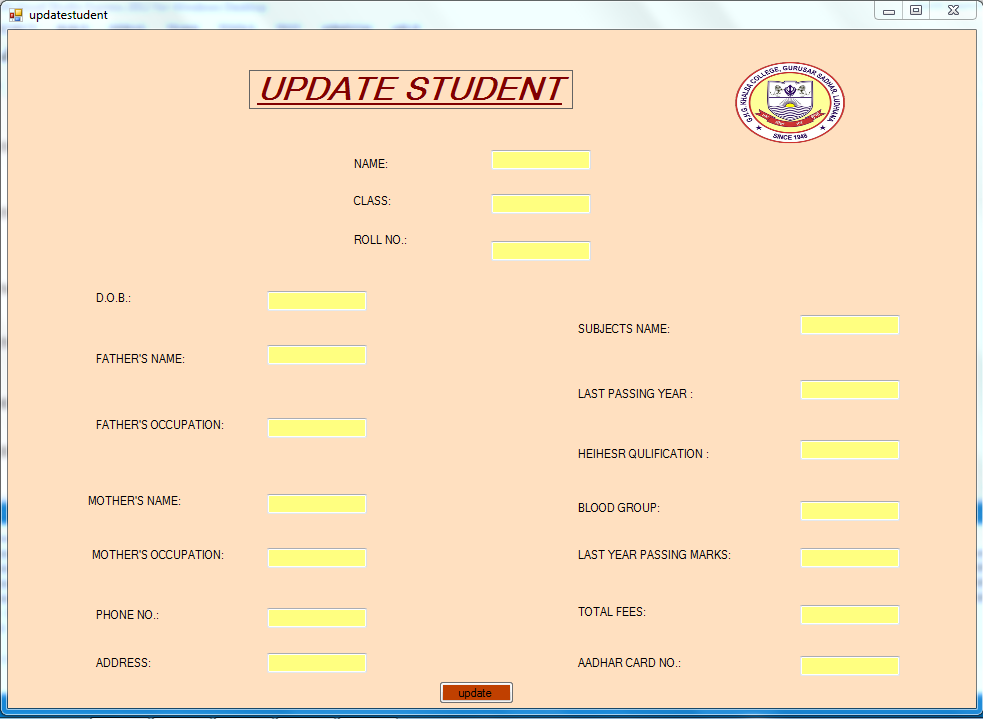
This is adminpage.It is admin page used by admin to add, update ,delete records.



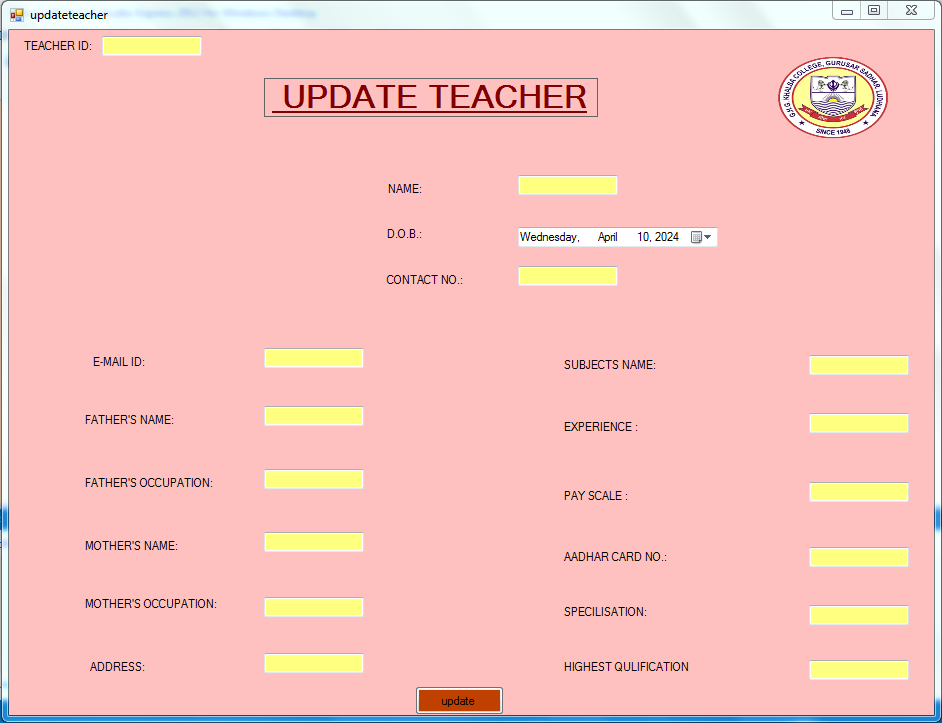
This is add student page .It is used by admin to add new student in database



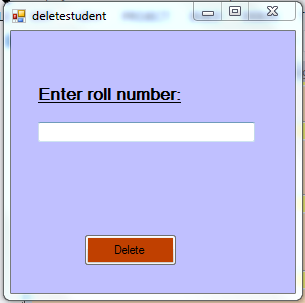
This is add teacher page .It is used by admin to add new student in database.



This is update student page.It is used by admin to update student record in database.



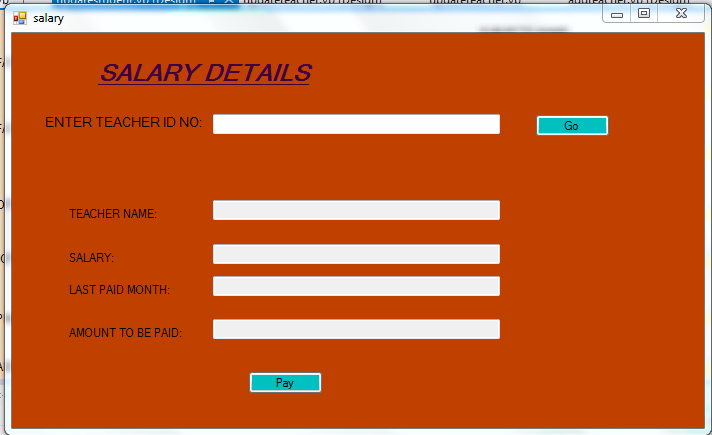
This is update teacher page .It is used by admin to update teacher record in database



This is delete student page .It is used by admin to delete student record in database by using roll no.



This is delete teacher page .It is used by admin to delete teacher record in database by using teacher id.



This is salary page .It us used to pay teacher’s salary.

**Future Scope**

The future scope of ERP (Enterprise Resource Planning) systems is exciting and expansive, driven by technological advancements and evolving business needs. Here are some key areas where the future of ERP systems is likely to focus:

1. **Cloud-Based ERP**: Cloud computing continues to dominate the IT landscape, offering scalability, flexibility, and cost-effectiveness. Future ERP systems will increasingly be delivered via the cloud, allowing for easier access, updates, and integration with other cloud services.
2. **AI and Machine Learning**: AI and machine learning technologies are poised to revolutionize ERP systems by automating routine tasks, providing intelligent insights, and enhancing decision-making processes. Expect ERP systems to incorporate AI-powered features for predictive analytics, demand forecasting, and process optimization.
3. **IoT Integration**: The Internet of Things (IoT) is transforming how businesses collect and analyze data from interconnected devices and sensors. ERP systems will leverage IoT integration to enable real-time monitoring of equipment, inventory management, and predictive maintenance.
4. **Blockchain for Transparency and Security**: Blockchain technology offers unparalleled transparency and security, making it an ideal solution for enhancing data integrity and trust in ERP systems. Expect to see blockchain integration in areas such as supply chain management, financial transactions, and compliance tracking.
5. **Mobile Accessibility**: With the proliferation of mobile devices, future ERP systems will prioritize mobile accessibility, allowing users to access critical business data and functionality from anywhere, at any time. Mobile ERP apps will become increasingly sophisticated, offering seamless integration with core ERP modules.
6. **Enhanced User Experience (UX)**: ERP vendors are investing in improving the user experience of their systems to enhance usability and productivity. Expect future ERP systems to feature intuitive interfaces, personalized dashboards, and role-based access controls tailored to the needs of different user groups.
7. **Industry-Specific Solutions**: As businesses become more specialized, ERP systems will offer industry-specific solutions tailored to the unique requirements of different sectors. Whether it's manufacturing, healthcare, retail, or finance, expect ERP vendors to develop specialized modules and features to address sector-specific challenges.
8. **Integration with Emerging Technologies**: ERP systems will continue to evolve alongside other emerging technologies such as augmented reality (AR), virtual reality (VR), and edge computing. These technologies will be integrated into ERP systems to improve collaboration, training, and decision support capabilities.

**Bibliography**

Beyond the guidance of the staff members we have consulted some books and websites while developing the project. Some of these books and websites are:

1.W3Schools.com

2.Tutorialpoint.com

3.geeksforgeeks.org