**1.INTRODUCTION**

**1.1 Abstract**

This project entitles as “**College Assignment Computerization**” is developed using Asp.Net as front end, C# as coding language and SQL Server as back end. All reports will be generated using data report.

The biggest challenge of a college office authority is to manage each and every student detail and their assignment submission. Often cases occur of changes in marks, pass list, name changes, data losses etc. In the existing scenario each student’s assignments are separately entered by their teachers first in a sheet of paper and then later assignments will be corrected and marks will be provided.  It is extremely tedious to do all the things manually and assign marks according to their performance; moreover there is every chance of entering wrong marks of the student. To overcome all the cons and disadvantages of the existing system, the proposed software is developed to make the entry and the retrieval of student data much easier.

This application is based on a web based application, which can be accessed my all department staffs, anywhere at any time. This application will enable as a mobile based application in future. All departments can be created in this application as well as student’s details can be created. The created details will be stored in the database for future reference. This application is developed especially for maintaining the assignments and marks of the student. Assignments are more important for the students to increase their percentage. After the submission of assignment all the student’s marks will be entered in this web application. This application will find out the late submission.

Students can able to submit their assignments through online itself. All the submitted assignment can be validated in online through staff login. Marks too updated on online. The most important feature in this project is, staff can able to select a best assignment in the class and they can share the assignment to the entire class. All the data updating can be done through web by the admin of this project. Reports will be furnished using data report.

**1.2 MODULES:**

* Students login
* Assignment submission
* Assignment Validation
* Marks Updates
* Assignment view & Late submission

**1.3 Modules Description**

**Students Login**

This is the initial module of this project, students plays the main role in this project. The corresponding staffs can create the student details. The staffs are responsible for maintaining their students list. While creating the students the staffs need to provide the basic information of the students like, Roll Number, Name, Department Name, Course name and etc. All the details will be centralized in the local server. Student’s details can be managed by the staffs in case if any updates. Once a student is created the student will be provided with a unique username and password. Students can get login into their account for further process.

**Assignment Submission**

Here students can get login into their account for assignment process. This is the core module in this project. Here multiple user interfaces can be done. The entire student can get login into this module. In staff side, the staffs need to provide the assignment name, Department, Course and year. The published details will be transferred to the corresponding student’s login. While the student getting login into their account, the student can view the assigned assignment. After the assignment preparation, students need to upload the assignment. The assignment can be in any formation. The data base will accept various formats for assignment.

**Assignment Validation**

After uploading the assignment, all the files will be centralized in the server for validation process. Here client server method with local host has been used. This method will separate the assignment contents and forward the details to the corresponding staff. The staff will receive the exact file uploaded by the student. So no more confusion will occur during the time of data transfer. In the assignment validation process, staff can view all the files uploaded by the student. The file can be either download by the staff or can be view directly on the browser itself.

**Marks updates**

This is the dependency module of the previous module. This is because both validation and transfer process has been used in this module. Here the staffs can able to provide the marks for the particular assignment. The mark details will be provided by the staffs to the student. Students can able to view the marks in their login itself. Staff can take consolidated mark report for the entire class.

**Best assignment & Late submission**

Being the centralization of all the details, all data come under data mining process. While submitting the assignment, data and time will be calculated. Late submission details will be highlighted in the grid. While staff come across the various assignment, in case the staff found any best assignment in the class means, the staff can able to select the best assignment and publish to the entire class room. The encourages the students to write or prepare good assignments.

**2. SYSTEM CONFIGURATION**

**2.1 Hardware Specification:**

PROCESSOR : Intel Core i5

MOTHERBOARD : Intel 915GVSR chipset board

RAM : 4 GB DDR 3RAM

HARD DISK DRIVE : 100 GB

MONITOR : 17” Color TFT Monitor

KEYBOARD : Multimedia Keyboard 108 Keys

MOUSE : Logitech Optical Mouse

BANDWIDTH : 100 mbps.

**2.2 SOFTWARE SPECIFICATION:**

FRONTEND : ASP.NET 2010

BACKEND : SQL Server 2008

CODING LANGUAGE : C#

OPERATING SYSTEMS : Microsoft windows 10

DOCUMENTATION : Microsoft word 2019

SCRIPTING LANGUAGE : Java Script

CLIENT SERVER TOOL : AJAX 4.0

**3. SYSTEM STUDY**

In the existing system all work is done on paper. The whole class need to submit assignment only through paper. This leads to more paper waste. This version provides a learning environment.

**3.1 DISADVANTAGES OF EXISTING SYSTEM**

**• Not User Friendly:**

The existing system is not user friendly because the retrieval of data is very slow and data is not maintained efficiently.

**• Difficulty in report generating:**

We require more calculations to generate the report so it is generated at the end of the session. And the student not gets a single chance to improve their attendance

**• Manual control:**

All calculations to generate report is done manually so there is greater chance of errors.

**• Lots of paperwork:**

Existing system requires lot of paper work. Loss of even a single register/record led to difficult situation because all the papers are needed to generate the reports.

**• Time consuming:**

Every work is done manually so we cannot generate report in the middle of the session or as per the requirement because it is very time consuming.

**3.2 Proposed System**

* **Enhancement:**

The main objective of Student admission System is to enhance and upgrade the existing system by increasing its efficiency and effectiveness. The software improves the working methods by replacing the existing manual system with the computer-based system.

* **Automation:**

The Student admission System automates each and every activity of the manual system and increases its throughput. Thus the response time of the system is very less and it works very fast.

* **Accuracy:**

The Student admission System provides the uses a quick response with very accurate information regarding the users etc. Any details or system in an accurate manner, as and when required.

* **User-Friendly:**

The Student admission System has a very user-friendly interface. Thus the users will feel very easy to work on it. The software provides accuracy along with a pleasant interface. Make the present manual system more interactive, speedy and user friendly.

* **Availability:**

The transaction reports of the system can be retried as and when required. Thus, there is no delay in the availability of any information, whatever needed, can be captured very quickly and easily.

* **Maintenance Cost:**

Reduce the cost of maintenance.

4 LANGUAGE SPECIFICATION

**4.1 INTRODUCTION TO ASP.NET**

ASP.NET is the .NET framework layer that handles Web requests for specific types of files, namely those with (.aspx or .ascx) extensions. The ASP.NET engine provides a robust object model for creating dynamic content and is loosely integrated into the .NET framework.

**WHAT IS ASP.NET?**

ASP.NET is part of the .NET framework. ASP.NET programs are centralized applications hosted on one or more Web servers that respond dynamically to client requests. The responses are dynamic because ASP.NET intercepts requests for pages with a specific extension (.aspx or .ascx) and hands off the responsibility for answering those requests to just-in-time (JIT) compiled code files that can build a response “on-the-fly.”

ASP.NET deals specifically with configuration (web.config and machine.config) files, Web Services (ASMX) files, and Web Forms (ASPX) files. The server doesn’t “serve” any of these file types—it returns the appropriate content type to the client. The configuration file types contain initialization and settings for a specific application or portion of an application. Another configuration file, called machine.web, contains machine-level initialization and settings. The server ignores requests for web files, because serving them might constitute a security breach.

Client requests for these file types cause the server to load, parse, and execute code to return a dynamic response. For Web Forms, the response usually consists of HTML or WML. Web Forms maintain state by round-tripping user interface and other persistent values between the client and the server automatically for each request.

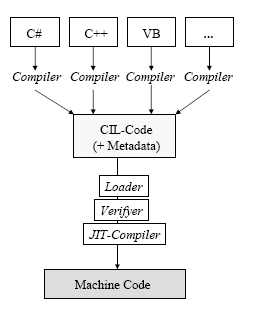
A request for a Web Form can use View State, Session State, or Application State to maintain values between requests. Both Web Forms and Web Services requests can take advantage of ASP. Net’s integrated security and data access through ADO.NET, and can run code that uses system services to construct the response. So the major difference between a static request and a dynamic request is that a typical Web request references a static file. The server reads the file and responds with the contents of the requested file.

ASP.NET uses .NET languages. ASP.NET code exists in multithreaded JIT compiled DLL assemblies, which can be loaded on demand. Once loaded, the ASP.NET DLLs can service multiple requests from a single in-memory copy.

ASP.NET supports all the .NET languages (currently C#, C++, VB.NET, and JScript, but there are well over 20 different languages in development for .NET), so you will eventually be able to write Web applications in your choice of almost any modern programming language.

In addition to huge increases in speed and power, ASP.NET provides substantial development improvements, like seamless server-to-client debugging, automatic validation of form data.

Fig2. Interoperability



**ASP.NET EVENTS are cool**

Every time an ASP.NET page is viewed, many tasks are being performed behind the scenes. Tasks are performed at key points ("events") of the page's execution lifecycle.

**The most common events are:**

**OnInit**

The first event in our list to be raised is OnInit. When this event is raised, all of the page's server controls are initialized with their property values. Post Back values are not applied to the controls at this time.

**On Load**

The next event to be raised is On Load, which is the most important event of them all as all the pages server controls will have their Post Back values now.

**Post Back Events**

Next all the Post Back events are raised. These events are only raised when the page view is the result of a Post Back. The order that these events are raised can't be defined or relied upon; the only consistency with the order that Post Back events are raised is that they are all raised between the Unload and OnPreRender events.

**OnPreRender**

This event is raised just prior to the page or server control's html output being written into the response stream that's sent to the client web browser. This is last chance you have to make any modifications. By this point, all the server controls on the page have the final data applied.

**On Unload**

This is the last event in our list to be raised and you should destroy any un-managed objects and close any currently open database connection at this point. It is not possible to modify any controls on the page at this point as the response stream has already been sent to the client web browser.

As each event of the page is raised it also automatically tells all its child controls to raise their own implementation of the same event. In turn each of those controls will tell its own child controls to do the same and so on down the control tree till all controls have done so. Then execution flow is passed back to the main page class to continue onto the next event and the process is repeated for that event.

**MAIN FEATURES OF ASP.NET**

Successor of *Active Server Pages* (ASP), but completely different architecture

• Object-oriented

• Event-based

• Rich library of Web Controls

• Separation of layout (HTML) and logic (e.g. C#)

• Compiled languages instead of interpreted languages

• GUI can be composed interactively with Visual Studio .NET

• Better state management

**NAMESPACES**

ASP.NET uses a concept called namespaces. Namespaces are hierarchical object models that support various properties and methods. For example, HTML server controls reside in "System.web.UI.HtmlControls" namespace, web server controls reside in “System.web.UI.WebControls" namespace and ADO+ resides in "System. Data" namespace.

**LANGUAGE INDEPENDENT**

An ASP.NET page can be created in any language supported by .NET framework. Currently .NET framework supports VB, C#, JScript and Managed C++.

**ASP.NET SERVER CONTROLS**

Using ASP.NET Server Controls, browser variation is handled because these controls output the HTML themselves based on the browser requesting the page.

**TYPES OF CONTROLS**

ASP.NET has two basic types of controls: HTML server controls and Web server controls.HTML Server Controls are generated around specific HTML elements and the ASP.NET engine changes the attributes of the elements based on server-side code that you provide. Web server controls revolve more around the functional you need on the page. The ASP.NET engine takes the extra steps to decide based upon the container of the requester, what HTML to output.

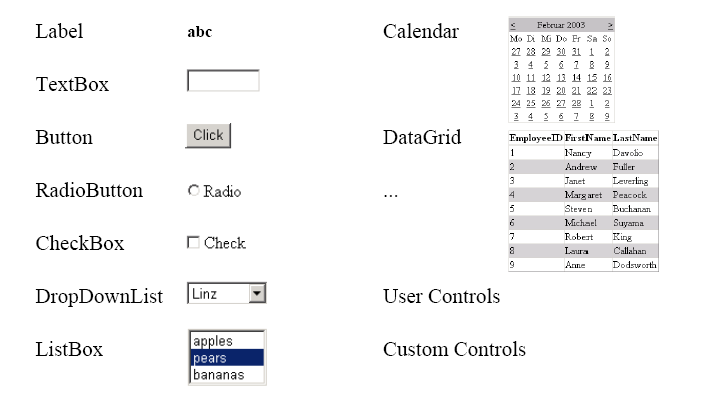


Fig 3. Web Controls

**Worldwide Field Training**

**ADO.NET**

ADO.NET provides a set of classes which a script can use to interact with databases. Scripts can create instances of ADO.NET data classes and access their properties and methods. A set of classes which work with a specific type of database is known as a **.NET Data Provider**. ADO.NET comes with two Data Providers, the SQL Server.NET Data Provider (which provides optimised access for Microsoft SQL Server databases) and the OLEDB.NET Data Provider, which works with a range of databases. The main ADO.NET OLEDB data access classes are OLEDBConnection, OLEDBCommand, OLEDBDataReader and OLEDBDataAdapter.

Visual Studio .NET is the single IDE that all the .NET languages can use. It makes everything available to all languages.

Visual Studio .NET is a great Multilanguage development environment and offers a complete set of tools to build Windows Forms , ASP.NET Web applications , and XML Web services.

###### Start Page

The Start page offers three tabs at the top of the window that enables to modify Visual Studio.NET as well as find important information. The tabs are

|  |  |
| --- | --- |
| **HTML Server Controls versus Web Server Controls** | |
| **Control Type** | **When to use this Control Type** |
| HTML Server Controls | When converting traditional ASP 3.0 Web pages to ASP.NET Web pages and speed of completion is a concern. It is a lot easier to change your HTML elements to HTML server controls than it is to change them to Web server controls.  When you prefer a more HTML-type programming model.  When you wish to explicitly control the code that is generated for the browser. |
| Web Server Controls | When you require a rich set of functionality to perform complicated page requirements.  When you are developing web pages that will be viewed by a multitude of browser types and that require different code based on these types. |

**Projects tab:**

This tab is the one to start new projects and launch projects that already exists. This tab lets you to create a new project or open an existing project.

**Online Resources tab:**

This tab provides a number of online resources when connected to the Internet.

**My Profile tab:**

This tab enables to customize the Visual Studio.NET environment to resemble the structured environment that is familiar with.

**Server Explorer**

This window enables to perform a number of functions such as database connectivity, performance monitoring, and interacting with event logs.

By using Server Explorer you can log on to a remote server and view database and system data about that server. Many of the functions that are performed with the Enterprise Manager in SQL Server can now be executed in the Server Explorer.

#### Solution Explorer

This provides an organized view of the projects in the application.The toolbar within the Solution Explorer enables to

* View code page of the selected item.
* View design page of the selected item.
* Refresh the state of the selected item.
* Copy the Web project between Web servers.
* Show all the files in the project, including the hidden files.
* See Properties of the selected item.

**Class View**

The Class View window can be viewed from the Start Page by clicking the Class View tab. The Class View shows all the classes that are contained within your solution.

The Class View shows the hierarchical relationship among the classes in your solution as well as the number of other items including methods, enumerations, namespaces, unions, and events. It is possible to organize the view of these items within the window by right-clicking anywhere in the Class View area and choosing how the items are sorted.

**Toolbox**

The Toolbox window enables to specify elements that will be part of the Windows Forms or Web Forms. It provides a drag and drop means of adding elements and controls to the pages or forms. The code snippets can also be stored within the Toolbox.

**Properties window**

This window provides the properties of an item that is part of the application. This enables to control the style and behavior of the item selected to modify.

**Dynamic Help**

This window shows a list of help topics. The help topics change based on the item selected or the action being taken.

The Dynamic Help window shows the help items displayed when you have a Button control on the page selected. After the item is selected, a list of targeted help topic is displayed. The topics are organized as a list of links. Clicking one of the links in the Dynamic Help window opens the selected help topic in the Document window.

**Document window**

The Document window is the main window within Visual Studio.NET where the applications are built.

The Document window shows open files in either Design or HTML mode. Each open file is represented by a tab at the top of the Document window. Any number of files can be kept open at the same time, and you can switch between the open files by clicking the appropriate tab.

**Design mode versus HTML mode**

Visual Studio.NET offers two modes for viewing and building files: Design and HTML. By clicking the Design tab at the bottom of the Document window, you can see how the page will view to the user. The page is built in the Design mode by dragging and dropping elements directly onto the design page or form. Visual Studio .NET automatically generates the appropriate code.

When the page is viewed in HTML mode, it shows the code for the page. It enables to directly modify the code to change the way in which the page is presented.

**Working with SQL Server through the Server Explorer**

Using Visual Studio.NET , there is no need to open the Enterprise Manager from SQL Server. Visual Studio.NET has the SQL Servers tab within the Server Explorer that gives a list of all the servers that are connected to those having SQL Server on them. Opening up a particular server tab gives five options:

* Database Diagrams
* Tables
* Views
* Stored Procedures
* Functions

**Database Diagrams**

To create a new diagram right click Database diagrams and select New Diagram. The Add Tables dialog enables to select one to all the tables that you want in the visual diagram you are going to create.

Visual Studio .NET looks at all the relationships between the tables and then creates a diagram that opens in the Document window.

Each table is represented in the diagram and a list of all the columns that are available in that particular table. Each relationship between tables is represented by a connection line between those tables.

The properties of the relationship can be viewed by right clicking the relationship line.

**Tables**

The Server Explorer allows to work directly with the tables in SQL Server. It gives a list of tables contained in the particular database selected.

By double clicking one of the tables, the table is seen in the Document window. This grid of data shows all the columns and rows of data contained in the particular table.

The data can be added or deleted from the table grid directly in the Document window. To add a new row of data , move to the bottom of the table and type in a new row of data after selecting the first column of the first blank row. You can also delete a row of data from the table by right clicking the gray box at the left end of the row and selecting Delete.

By right clicking the gray box at the far left end of the row, the primary key can be set for that particular column. The relationships to columns in other tables can be set by selecting the Relationships option.

To create a new table right-click the Tables section within the Server Explorer and selecting New Table. This gives the design view that enables to start specifying the columns and column details about the table.

To run queries against the tables in Visual Studio .NET, open the view of the query toolbar by choosing View->Toolbars->Query.

To query a specific table, open that table in the Document window. Then click the SQL button which divides the Document window into two panes-one for query and other to show results gathered from the query.

The query is executed by clicking the Execute Query button and the result is produced in the lower pane of the Document window.

#### Views

To create a new view, right-click the View node and select New View. The Add Table dialog box enables to select the tables from which the view is produced. The next pane enables to customize the appearance of the data in the view.

4.2 FEATURES OF SQL-SERVER

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six type of objects,

They are,

1. TABLE

2. QUERY

3. FORM

4. REPORT

5. MACRO

**TABLE:**

A database is a collection of data about a specific topic.

**VIEWS OF TABLE:**

We can work with a table in two types,

1. Design View

2. Datasheet View

**Design View**

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

**Datasheet View**

To add, edit or analyses the data itself we work in tables datasheet view mode.

**QUERY:**

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot(it cannot be edited).Each time we run query, we get latest information in the dynaset.Access either displays the dynaset or snapshot for us to view or perform an action on it ,such as deleting or updating.

**FORMS:**

A form is used to view and edit information in the database record by record .A form displays only the information we want to see in the way we want to see it. Forms use the familiar controls such as textboxes and checkboxes. This makes viewing and entering data easy.

**Views of Form:**

We can work with forms in several primarily there are two views,

They are,

**1. Design View**

**2. Form View**

**Design View**

To build or modify the structure of a form, we work in forms design view. We can add control to the form that are bound to fields in a table or query, includes textboxes, option buttons, graphs and pictures.

**Form View**

The form view which display the whole design of the form.

**REPORT:**

A report is used to vies and print information from the database. The report can ground records into many levels and compute totals and average by checking values from many records at once. Also the report is attractive and distinctive because we have control over the size and appearance of it.

**MACRO:**

A macro is a set of actions. Each action in macros does something. Such as opening a form or printing a report .We write macros to automate the common tasks the work easy and save the time.

**MODULE:**

Modules are units of code written in access basic language. We can write and use module to automate and customize the database in very sophisticated ways.

**WHAT IS “.NET”?**

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There’s no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

“.NET” is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

**THE .NET FRAMEWORK**

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).

2. A hierarchical set of class libraries.

The CLR is described as the “execution engine” of .NET. It provides the environment within which programs run. The most important features are:

* Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
* Memory management, notably including garbage collection.
* Checking and enforcing security restrictions on the running code.
* Loading and executing programs,with version control and other features.

The following features of the .NET framework are also worth description:

**Managed Code** - is code that targets .NET, and which contains certain extra information - “metadata” - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

**Managed Data** - With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you’re using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn’t get garbage collected but instead is looked after by unmanaged code.

**Common Type System** - The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn’t attempt to access memory that hasn’t been allocated to it.

**Common Language Specification** - The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

**THE CLASS LIBRARY**

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

**LANGUAGES SUPPORTED BY .NET**

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft’s old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family:

Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, custom attributes and also supports multi-threading.

Visual Basic .NET is also CLS compliant, which means that any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET.

Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework.

C# is Microsoft’s new language. It’s a C-style language that is essentially “C++ for Rapid Application Development”. Unlike other languages, its specification is just the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own.

Microsoft Visual J# .NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of other programming languages.

Active State has created Visual Perl and Visual Python, which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment. Visual Perl includes support for Active State’s Perl Dev Kit.

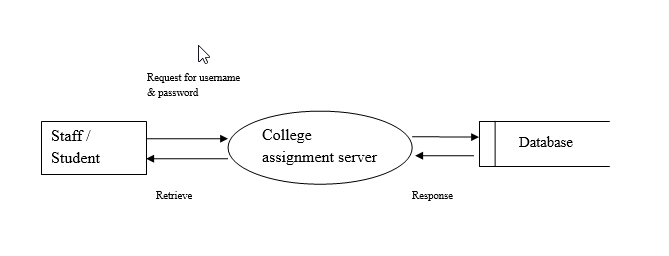
Other languages for which .NET compilers are available include:

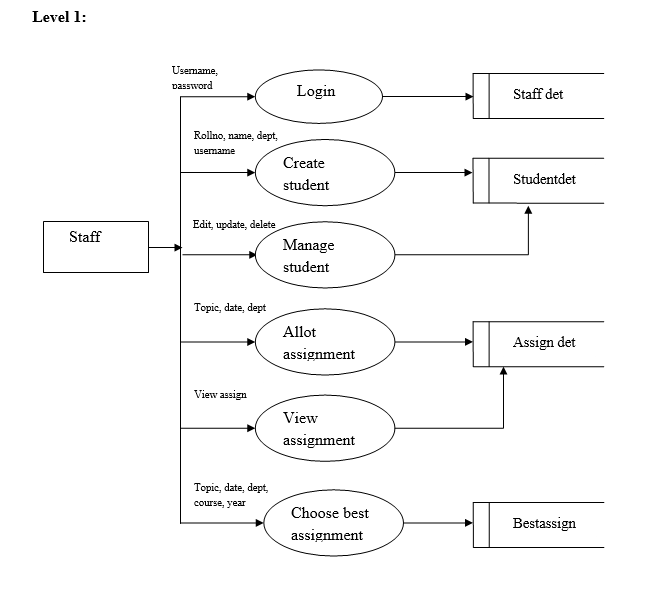
* FORTRAN
* COBOL
* Eiffel

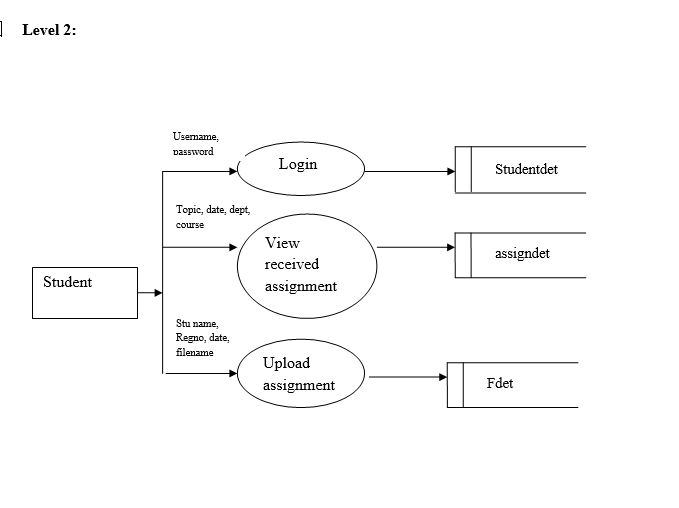
|  |  |
| --- | --- |
| ASP.NET  XML WEB SERVICES | Windows Forms |
| Base Class Libraries | |
| Common Language Runtime | |
| Operating System | |

**5. SYSTEM DESIGN**

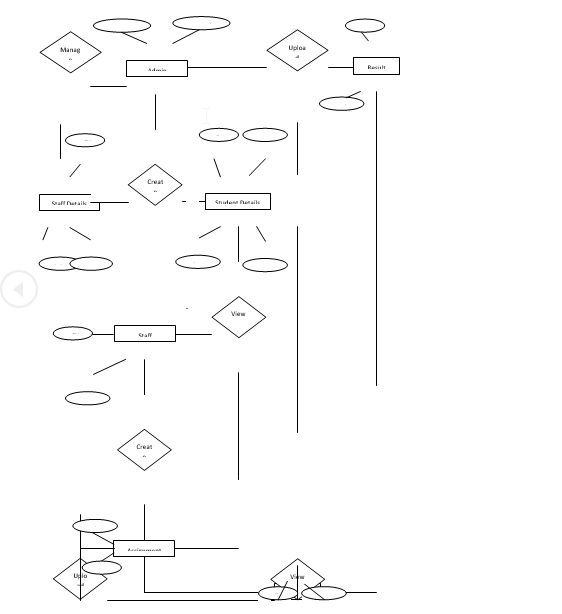
**5.1 DATAFLOW DIAGRAM**

****

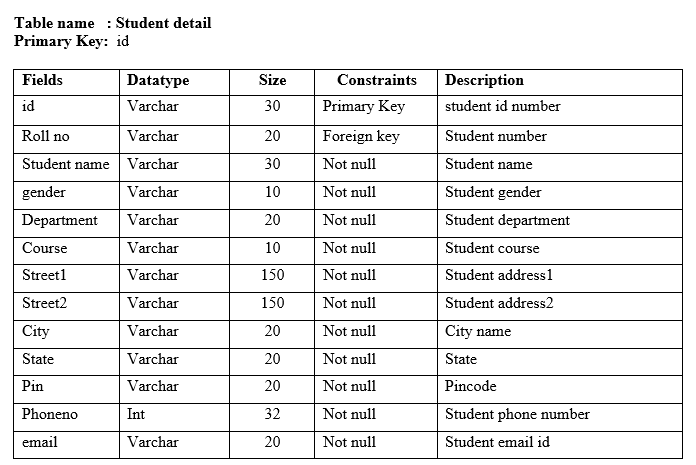
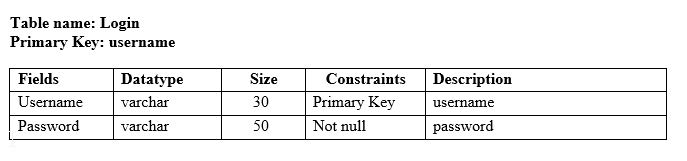
****

****

**5.2 ER DIAGRAM**

****

**5.3 TABLE DESIGN.**



**6. SYSTEM TESTING**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client/server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

### TESTING ISSUES

* Client GUI considerations
* Target environment and platform diversity considerations
* Distributed database considerations
* Distributed processing considerations

**6.1 TESTING METHODOLOGIES:**

System testing is the state of implementation, which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences. It certifies that the whole set of programs hang together. System testing requires a test plan that consists of several key activities and steps for run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system.

Testing is an important stage in software development. The system test in implementation stage in software development. The system test in implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification. Testing is vital to the achievement of the system goals. The objective of testing is to discover errors. To fulfill this objective a series of test step unit, integration, validations and system tests were planned and executed.

**6.2 TESTING OBJECTIVES:**

Testing is a process of executing a program with the intent of finding an error.

A good test case is one that has a high probability of finding on as yet undiscovered error.

A successful test is one that uncovers on as yet undiscovered error.

The above objectives imply a change in view. They move counter to the commonly held view that a successful list is one in which no errors are found. Any engineered product can be listed in one of two ways

1. Knowing the specified function that a product has been designed to perform tests can be conducted to demonstrate each function is fully operational.

2. Knowing the internal workings of a product, tests can be conducted to ensure that “all gear mesh” that is, the internal operation of the product performs according to specification and all internal components have been adequately exercised.

**Types of testing**

1. Unit Testing

2. Integration Testing

3. Validation Testing

4. Output Testing

5. User acceptance Testing

6. System Testing:-

**Unit Testing:-**

All modules were tested and individually as soon as they were completed were checked for there correct functionality. Unit testing is carried out by verify and recover errors within the boundary of the smallest unit or a module. In this testing step, each module was found to be working satisfactory per the expected output of the module. In the package development, each module is tested separately after it has been completed and checked with valid data.

**Integration Testing :-**

The entire project was split into small programs; each of these single programs gives a frame as an output. These programs were tested individually; at last all these programs where combined together by creating another program where all these constructions were used. It give a lot of problem by not functioning in an integrated manner.

The user interface testing is important since the user has to declare that the arrangements made in the frames are convenient and it is satisfied. When the frames are the test, the end user gave suggestion. Since they where much exposed to do the work manually.

**Validation Testing: -**

At the culmination of the black box testing software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of test i.e., validation succeeds when the software functions in a manner that can be reasonably accepted by the customer.

**Output Testing:-**

After performing the validation testing the next step is output testing of the proposed system. Since the system cannot be useful if it does not produce the required output.

Asking the users about the format in which the system is required tests the output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is on screen and another one is printed format. The output format on the screen is found to be corrected as the format was designed in the system phase according to the user needs. And for the hardcopy the output comes according to the specifications requested by the user. Here the output testing does not result in any correction in the system.

**User Acceptance Testing:-**

User acceptance testing of the system is the key factor the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of development and making change whenever required. This is done with regard to the input screen design and output screen design.

**System Testing:-**

This is to verify that all the system elements have been properly integrated and perform allocated functions. Testing is executing a program to test the logic changes made in it and with intention of finding errors. Tests are also conducted to find discrepancies between system and its original objective, current specification and documents.

**6.3 QUALITY ASSURANCE:**

Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confidence that product quality is meeting its goal.

**Quality Assurance Goals**

**Correctness**

The extent to which the program meets system specifications and user objectives.

**Reliability**

The degree to which the system performs its intended functions overtime.

**Efficiency**

The amount of computer resources required by a program to perform a function.

**Usability**

The effort required learning and operating a system.

**Maintainability**

To use with which program errors are located and corrected.

**Testability**

The effort required testing a program to ensure its correct performance.

**Portability**

The ease of transporting a program from one hardware configuration to another.

**Accuracy**

The required position in input editing, computations and output.

**Generic Risks**

The general risk involves two categories:

☼ **Uncertainty:**

The risk may or may not happen; that is, there are no 100% probable risks

☼ **Loss:**

If the risk becomes a reality, unwanted consequences or losses will occur.

**Different categories of risks are considered:**

**Project risks:-**

Project risks identify potential budgetary, schedule, personnel like staffing and organization, resource, customer, and requirements problems and their impact on a software project.

**Technical risks:-**

Technical risks identify potential design, implementation, interface, verification, and maintenance problems.

**Security Technologies and Policies**

Any system developed should be secured and protected against possible hazards. Security measures are provided to prevent unauthorized access of the database at various levels. At uninterrupted power supply should be so that the power failure or voltage fluctuations will not erase the data in the files.

Password protection and simple procedures to change the unauthorized access are provided to the users. The system allows the user to enter the system for product management and order status entry only through login utility. The user will have to enter the user name and password.

A multi-layered security architecture comprising firewalls, filtering routers, encryption and digital certification must be ensured in this project in real time that order and payment details protected from unauthorized access. The customer can access this order status only by using his customer code and order number.

**7. SYSTEM IMPLEMENTATION**

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

**The stage consists of**

* Testing the developed program with simple data.
* Detection’s and correction of error.
* Creating whether the system meets user requirements.
* Testing whether the system.
* Making necessary changes as desired by the user.
* Training user personnel.

**7.1 IMPLEMENTATION PROCEDURES:**

The implementation phase is less creative than system design. A system project may be dropped at any time prior to implementation, although it becomes more difficult when it goes to the design phase.

The final report to the implementation phase includes procedural flowcharts, record layouts, report layouts, and a workable plan for implementing the candidate system design into an operational one. Conversion is one aspect of implementation.

* The conversion portion of the implementation plan is finalized and approved.
* Files are converted.
* Parallel processing between the existing and the new system are logged on a special form.
* Assuming no problems, parallel processing is discontinued. Implementation results are documented for reference.

**7.2 USER TRAINING:**

User Training is designed to prepare the user for testing & convening the system. There are several ways to train the user. They are

1) User Manual.

2) Help Screens.

3) Training Demonstration.

**1. User Manual:**

The summary of important functions about the system and software can be provided as a document to the user.

**2. Help Screens:**

This features now available in every software package, especially when it is used with a menu. The user selects the “Help” option from the menu. The system accesses the necessary description or information for user reference.

**3. Training Demonstration:**

Another User Training element is a Training Demonstration. Live demonstrations with personal contact are extremely effective for Training Users.

**7.3 OPERATIONAL DOCUMENTATION:**

Documentation means of communication; it establishes design and performance criteria for phases of the project. Documentation is descriptive information that portrays the use and / or operation of the system.

**7.3.1 Documentation Tools**

Document production and desktop publishing tool support nearly ever aspect of software developers. Most software development organizations spend a substantial amount of time developing documents, and in many cases the documentation process itself is quite inefficient. It is not use unusual for a software development effort on documentation. For this reason, documentation tools provide an important opportunity to improve productivity.

**7.3.2 Document Restructuring**

Creating document is far too timed consuming. If the system work’s, we’ll live with what we have. In some cases, this is the correct approach. It is not possible to recreate document for hundreds of computer programs.

Documentation must be updated, but we have limited resources. It may not be necessary to fully redocument an application. Rather, those portions of the system that are currently undergoing change are fully documented.

The system is business critical and must be fully redocumented. Even in this case, an intelligent approach is to pare documentation to an essential minimum.

##### 7.4 SYSTEM MAINTANANCE:

Maintenance is actually the implementation of the review plan. As important as it is, many programmers and analysts are to perform or identify themselves with the maintenance effort. There are psychological, personality and professional reasons for this. Analysts and programmers spend far more time maintaining programs than they do writing them. Maintenance accounts for 50-80 percent of total system development.

**Maintenance is expensive. One way to reduce the maintenance costs are through maintenance management and software modification audits.**

* Maintenance is not as rewarding as exciting as developing systems. It is perceived as requiring neither skill not experience.
* Users are not fully cognizant of the maintenance problem or its high cost.
* Few tools and techniques are available for maintenance.
* A good test plan is lacking.
* Standards, procedures, and guidelines are poorly defined and enforced.
* Programs are often maintained without care for structure and documentation.
* There are minimal standards for maintenance.
* Programmers expect that they will not be in their current commitment by time their programs go into the maintenance cycle.

**8. CONCLUSION**

The Project titled **“College Assignment computerization”** is tested with sample data and found to be working well. The project has been completed successfully with the maximum satisfaction of the organization. It has been designed to meet the requirements. This software was tested and found to work satisfactorily without any problem. Various test factors are also followed; it was error free and was found to work satisfactorily.

Testing and implementation are done successfully. The system satisfies all requirements needed by the user. I conclude the software as best to my knowledge.

**9. SCOPE FOR FURURE ENHANCEMENT**

The web application has developed in such way that enhancement can be done easily. The changes can be updated with less time and have no risk to add a new modules. The system will be improved for further enhancements, whenever the user needs an additional feature. It also helps to enhance their web application and thus it is helpful for the growth of the organization.

The below mentioned enhancements are likely to happen in the near future. The system can be easily upgraded to hold these features.

After administrator authentication of user name and password for students, the user name and password can be sent via Mail and SMS to particular user.

This project can be further enhanced based on the future trends and strategies.

**10. BIBLIOGRAPHY**

**Books Referred:**

1. More ASP.NET (Teach Yourself) - Lowell Mauer
2. Guide to ASP.NET - Peter Norton,
3. Fundamentals of Database System - Ramez
4. Complete Guide to SQL server - Peter Norton.

**Web links referred:**

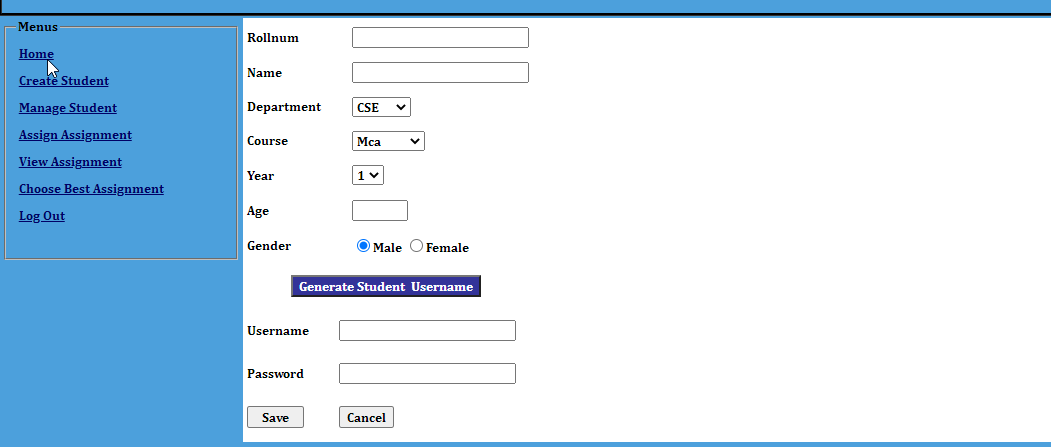
[**Http://www.Sourcecode.com**](Http://www.Sourcecode.com)

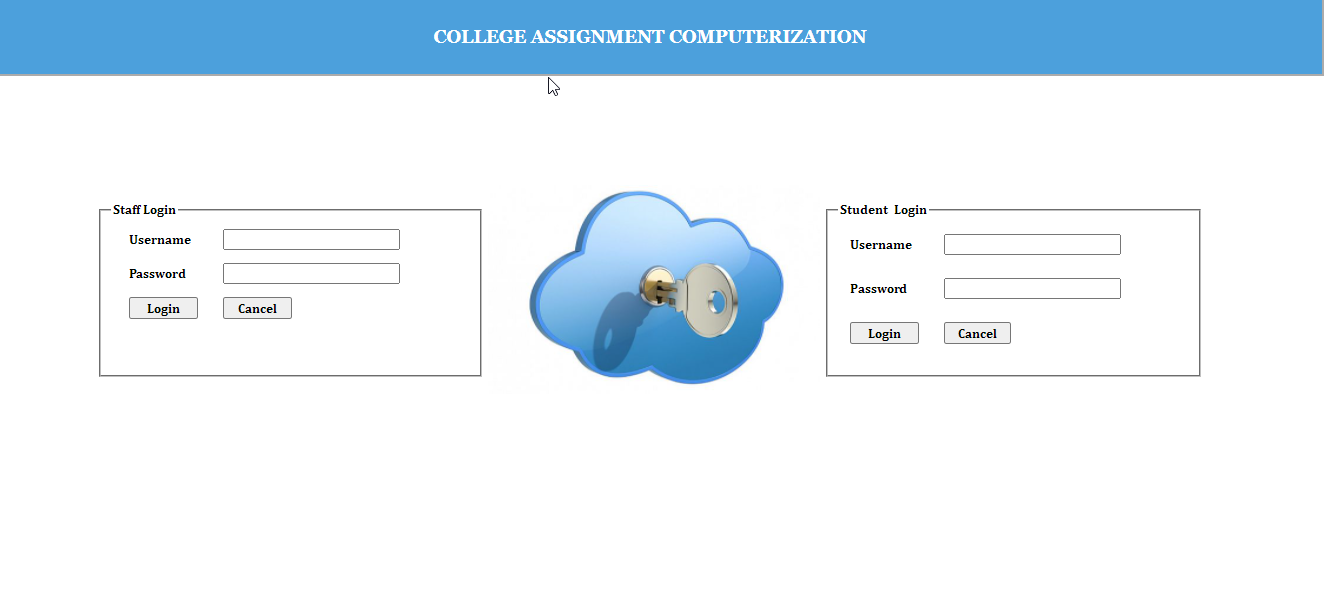
[**Http://www.dbms.co.in**](Http://www.dbms.co.in)

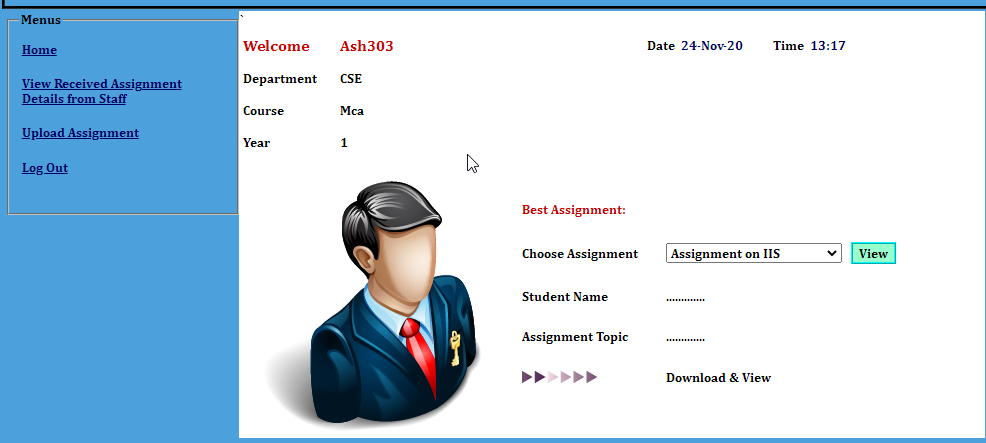
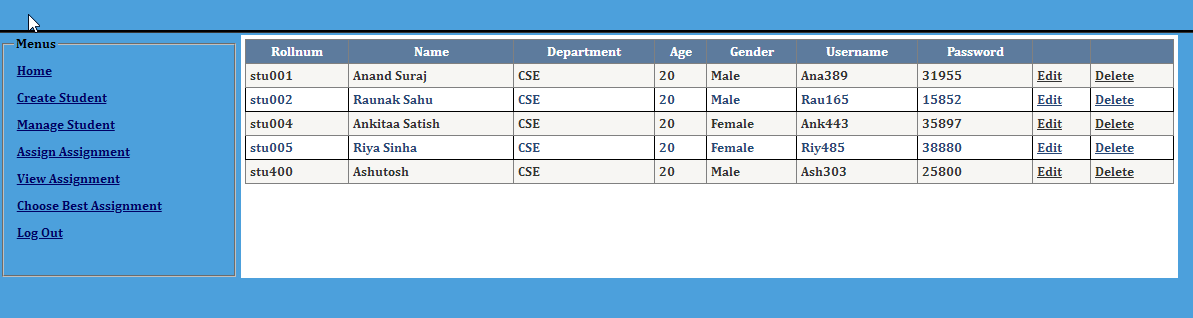
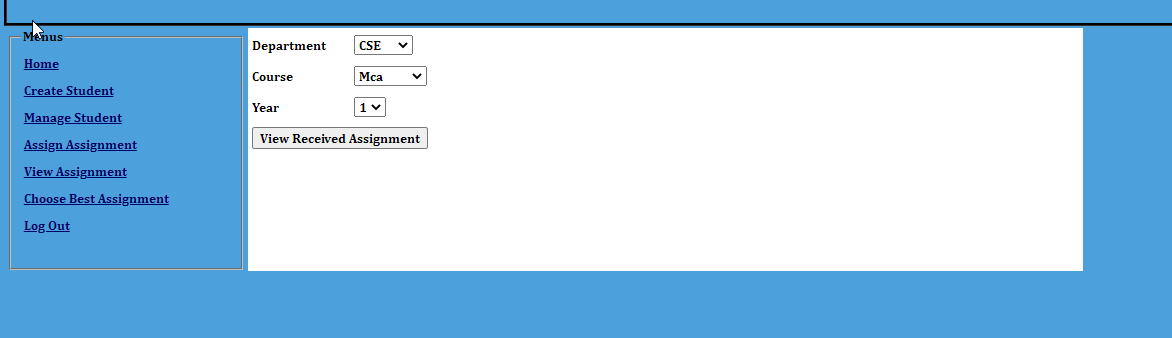
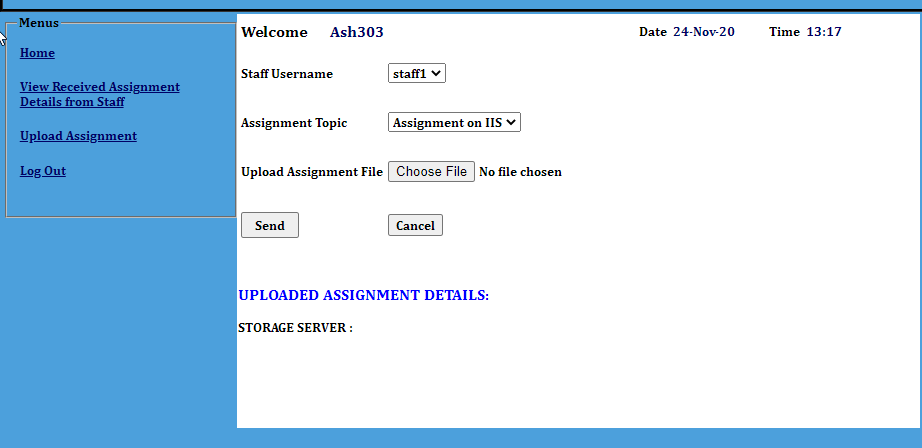
[**Http://A1code.com**](Http://A1code.com)

**11. APPENDIX**

**11.1 SCREEN SHOTS**

****

****



**11.2 SAMPLE CODE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.OleDb;

using System.Web.Configuration;

public partial class createstu : System.Web.UI.Page

{

OleDbConnection con;

OleDbCommand cmd;

string query, a;

protected void Page\_Load(object sender, EventArgs e)

{

}

public void data()

{

string connstring = WebConfigurationManager.ConnectionStrings["connect"].ConnectionString;

con = new OleDbConnection(connstring);

con.Open();

}

protected void Button3\_Click(object sender, EventArgs e)

{

int len = txtname.Text.Length;

if (len > 2)

{

int u = randomnumber(100, 888);

int p = randomnumber(11112, 67898);

txteusername.Text = txtname.Text.Substring(0, 3) + u.ToString();

txtepassword.Text = p.ToString();

}

else

{

}

}

private int randomnumber(int min, int max)

{

Random random = new Random();

return random.Next(min, max);

}

protected void Button4\_Click(object sender, EventArgs e)

{

if (RadioButton1.Checked == true)

{

a = "Male";

}

else

{

a = "Female";

}

data();

query = "insert into userdet(rollnum,name,dept,course,yearr,age,gender,uname,pwd)values('" + txtrollnum.Text + "','" + txtname.Text + "','" + dropdept.SelectedItem + "','" + DropDownList1.SelectedItem + "','" + dropyear.SelectedItem + "','" + txtage.Text + "','" + a.ToString() + "','" + txteusername.Text + "','" + txtepassword.Text + "')";

cmd = new OleDbCommand(query, con);

cmd.ExecuteNonQuery();

con.Close();

txtrollnum.Text = "";

txtname.Text = "";

txtage.Text = "";

txteusername.Text = "";

txtepassword.Text = "";

}

protected void txtrollnum\_TextChanged(object sender, EventArgs e)

{

data();

query = "select rollnum from userdet where rollnum='" + txtrollnum.Text + "'";

cmd = new OleDbCommand(query, con);

OleDbDataReader rd = cmd.ExecuteReader();

if (rd.Read())

{

lblerr.Visible = true;

}

else

{

lblerr.Visible = false;

}

rd.Close();

con.Close();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.OleDb;

using System.Web.Configuration;

public partial class choosebest : System.Web.UI.Page

{

OleDbConnection con;

OleDbCommand cmd;

string query,staffuname;

string assgntopic, fname, student;

protected void Page\_Load(object sender, EventArgs e)

{

staffuname = Session["staff"].ToString();

}

public void data()

{

string connstring = WebConfigurationManager.ConnectionStrings["connect"].ConnectionString;

con = new OleDbConnection(connstring);

con.Open();

}

protected void Button1\_Click(object sender, EventArgs e)

{

Panel1.Visible = true;

DropDownList2.DataBind();

}

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

{

staffuname = Session["staff"].ToString();

student = GridView1.SelectedRow.Cells[0].Text.ToString();

fname = GridView1.SelectedRow.Cells[1].Text.ToString();

assgntopic = GridView1.SelectedRow.Cells[2].Text.ToString();

data();

query = "insert into bestassign(assigntopic,fname,sentstaff,stuuname,dept,course,yearr)values('" + assgntopic + "','" + fname + "','" + staffuname + "','" + student + "','" + dropdept.SelectedItem + "','" + DropDownList1.SelectedItem + "','" + dropyear.SelectedItem + "')";

cmd = new OleDbCommand(query, con);

cmd.ExecuteNonQuery();

con.Close();

Panel1.Visible = false;

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.OleDb;

using System.Web.Configuration;

public partial class viewreceiveddet : System.Web.UI.Page

{

OleDbConnection con;

OleDbCommand cmd;

string query, a;

protected void Page\_Load(object sender, EventArgs e)

{

lblusername.Text = Session["user"].ToString();

data();

query = "select dept,course,yearr from userdet where uname='" + lblusername.Text + "'";

cmd = new OleDbCommand(query, con);

OleDbDataReader rd = cmd.ExecuteReader();

while(rd.Read())

{

lbldept.Text = rd[0].ToString();

lblcourse.Text = rd[1].ToString();

lblyear.Text = rd[2].ToString();

}

rd.Close();

con.Close();

}

public void data()

{

string connstring = WebConfigurationManager.ConnectionStrings["connect"].ConnectionString;

con = new OleDbConnection(connstring);

con.Open();

}

}