**MULTI COMPANY PAYROLL SYSTEM**

**A MINI PROJECT REPORT**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE**

**REQUIREMENTS FOR THE AWARD OF THE DEGREE OF**

**MASTER OF SCIENCE**

**IN**

**INFORMATION TECHNOLOGY**

**Submitted by**

**M.PRIYADHARSHINI**

**REG. NO: I19171028**

**Under the Guidance of**

**DR.T. RATHIMALA M.C.A., M.Phil., Ph.D.,**

**Assistant Professor / Programmer**

**ANNAMALAI****UNIVERSITY**

**FACULTY OF SCIENCE**

**DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE**

**ANNAMALAINAGAR – 608 002**

**MAY - 2022**

**MULTI COMPANY PAYROLL SYSTEM**

**A MINI PROJECT REPORT**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE**

**REQUIREMENTS FOR THE AWARD OF THE DEGREE OF**

**MASTER OF SCIENCE**

**IN**

**INFORMATION TECHNOLOGY**

**Submitted by**

**M.PRIYADHARSHINI**

**REG. NO: I19171028**

**Under the Guidance of**

**DR.T.RATHIMALA M.C.A., M.Phil., Ph.D.,**

**Assistant Professor / Programmer**

**ANNAMALAI****UNIVERSITY**

**FACULTY OF SCIENCE**

**DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE**

**ANNAMALAINAGAR – 608 002**

**MAY - 2022**

**ANNAMALAI**  **UNIVERSITY**

**FACULTY OF SCIENCE**

**DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE**

**BONAFIDE CERTIFICATE**

This is to certify that the Project work entitled **“MULTI COMPANY PAYROLL SYSTEM”** is a bonafide work done by **Mr/Ms**. **M.PRIYADHARSHINI, Reg. No: I19171028** of **Master of Science** in **Information Technology (6THSemester) [Integrated]** during the academic year December 2021 - May 2022.

**Signature of the Guide**

**[T.RATHIMALA]**

Internal Examiner External Examiner

Place: Annamalai Nagar

Date:

Head of the Department

**DECLARATION**

I hereby declare that the project entitled **“[MULTI COMPANY PAYROLL SYSTEM]”**, submitted for the award of the degree of Master of Science in Information Technology [Integrated] is my original work. I declare that the project has not formed the basis for the award of any Degree, Diploma, Associate ship, Fellowship or similar other titles. It has not been submitted to any University or Institution for the award of any Degree or Diploma.

Signature of the Student

**M.PRIYADHARSHINI**

**Reg . No: I19171028**

Place: Annamalai Nagar

Date:

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to **Dr. N. PUVIARASAN, Professor & Head,** Department of Computer & Information Science, Annamalai University for his consistent co-operation and encouragement throughout the completion of the degree.

It gives me immense pleasure to express my deep sense of gratitude and sincere thanks to **Dr. K. SEETHARAMAN, Registrar**, Annamalai University and **Dr. V. RAMASWAMY, Dean**, Faculty of Science, Annamalai University who has given me the opportunity to pursue my degree.

I express my heartfelt thanks to my project coordinator **Dr**. **ARANGA. ARIVARASAN**, Assistant Professor / Programmer, Department of Computer & Information Science, Annamalai University for the encouragement and valuable guidance in the most demanding times to overcome the difficulties throughout the completion of this project work.

I express my heartfelt thanks forever to my project guide **T.RATHIMALA ,**Assistant Professor / Programmer, Department of Computer & Information Science, Annamalai University for suggesting this work. She also provided valuable and constant guidance, encouragement, reliable support and appreciated criticism throughout my completion of this project work.

I wish to record my sincere thanks to all the faculty members and non-teaching staff working in the Department. I also express my heartfelt thanks to my friends and family members whose blessings made this task possible for me.

**M.PRIYADHARSHINI**

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLES** | **PAGE NO** |
|  | **ABSTRACT** | 8 |
| 1 | **INTRODUCTION**   * 1. About the Project | 9 |
| 2 | **SYSTEM ANALYSIS**  2.1 Existing System    2.2 Proposed System | 10 |
| 3 | **SYSTEM REQUIREMENT**  3.1 Hardware Requirements  3.2 Software Requirements | 12 |
| 4 | **LITERATURE**  4.1 ASP.NET  4.2 SQL Database | 13 |
| 5 | **SYSTEM STUDY**  5.1 Feasibility study  5.1.1 Technical Feasibility    5.1.2 Operation Feasibility    5.1.3 Economical Feasibility | 21 |
| 6 | **SYSTEM DESIGN**  6.1 Use Case Diagram | 29 |
| 7 | **SYSTEM IMPLEMENTATION**  7.1 Module Description  7.2 Testing  7.2.1 Unit Testing  7.2.2 Integration Testing  7.2.3 System Testing  7.2.4 Acceptance Testing | 30 |
| 8 | **CONCLUSION** | 35 |
|  | **REFERENCES** | 36 |
|  | **APPENDIX**  SOURCE CODE  SCREEN SHOTS | 37  64 |

**ABSTRACT**

**MULTI COMPANY PAYROLL SYSTEM**

* Multi-company payroll system is designed to make the existing manual system automatic with the help of computerized equipment and full-edged computer software application, fulfilling their requirements, so that their valuable data and information can be stored for a longer period with easy access and manipulation of the same.
* The required software is easily available and easy to work. This web application can maintain and view computerized records without getting redundant entries. The project describes how to manage user data for good performance and provide better services for the client.

**CHAPTER-I**

**INTRODUCTION**

**1.1 ABOUT THE PROJECT**

This application is built to eliminate and, in some cases, reduce the hardships faced by the existing system. Moreover, this system is designed for particular need of the company to carry out its operations in a smooth and effective manner.

This web application is reduced as much as possible to avoid errors while entering data. It also provides error message while entering invalid data.

It is user-friendly as no formal knowledge is required to use the system. Human resource challenges are faced by every organization which has to be overcome by the organization. Every organization has different employee and payroll management needs.

**CHAPTER-II**

**SYSTEM ANALYSIS**

**2.1 EXISTING SYSTEM**

* Required a lot of paperwork.
* Everything is done on paper and these are highly prone to damages and require a good amount of security and space to store.
* Required Buying of goods more frequent as compared to the online system.
* If these files are less in number then it is not that big problem but if the number increases then searching for a particular document can give a personal nightmare as these are more prone to damages and even it gets the searching job done the time required to do the job is not ideal at all.
* Require more physical work and manpower.
* Editing is another level of problem in those paper documents the only option is to make new documents if any correction needs to be made that is not good at all for the environmental credentials.
* Where all the drawbacks of the pre-existing system have been overcome in the proposed system. It definitely improves the speed by automating most of the tasks required to be done manually previously.
* Automated systems are more reliable and efficient now a day for these types of work.

**2.2 PROPOSED SYSTEM**

The purpose of this document is to describe the functionality and specifications of the design of a web application for Managing Employees and their payroll. The expected audiences of this document are the developers and the admin of the web application. Now with the help of this system the admin has the information on his finger tips and can easily prepare a good record based on their requirements. Finally, we can say that this system will not only automate the process but save the valuable time of the manager or the admin, which can be well utilized buy his institute. This will be an additional advantage and management of power based on their free time from his normal duty.

**CHAPTER-III**

**SYSTEM REQUIREMENT**

The system requires the following as minimum recommended configuration for processing.

**3.1 Hardware Requirements**

Processor : Pentium IV

Hard Disk : 250GB

RAM : 3GB

**3.2** **Software Requirements**

Operating System : Windows 7

User Interface : ASP.Net

Platform : Microsoft Visual Studio 2010

Database : SQL Server2008

Server Deployment : IIS (Internet Information Server)6.0

Frame Work : .Net Framework 4.0

**CHAPTER-IV**

**LITERATURE**

**OVERVIEW OF TECHNOLOGIES USED**

**4.1 Front End – Asp.NET**

ASP.Net is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web application for PC, as well as mobile devices. ASP.Net works on top of the HTTP protocol and uses the HTTP commands and policies to set a browser-to-server two-way communication and cooperation.

ASP.Net is a part of Microsoft .Net platform. ASP.Net applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework.

The ASP.Net application codes could be written in either of the following languages:

C#

Visual Basic .Net

Jscript

J#

ASP.Net is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls like text boxes, buttons and labels for assembling, configuring and manipulating code to create HTML pages.

**ASP.Net Web Forms Model:**

ASP.Net web forms extend the event-driven model of interaction to the web applications. The browser submits a web form to the web server and the server returns a full markup page or Html page in response. All client-side user activities are forwarded to the server for state full processing. The server processes the

output of the client actions and triggers the reactions. Now, HTTP is a statelessprotocol. Asp.Net framework helps in storing the information regarding the state of the application.

**ASP.Net Component model**

The ASP.Net component model provides various building blocks of ASP.Net pages. Basically, it is an object model, which describes:

Server-side counter parts of almost all HTML elements or tags, like <form> and <input>.Server controls, which help in developing complex user-interface for example the Calendar control or the Grid view control.

ASP.Net web application is made of pages. When a user requests an ASP.Net page, the IIS delegates the processing of the page to the ASP.Net runtime system.

**.NET platform:**

The .Net platform is an integral component of the Microsoft Windows operating system for building and running next generation software applications and web services. The .Net development framework provides a new and simplified model for programming and developing applications on the Windows platform. It provides such advantages as multiplatform applications, automatic resource management, and simplification of application development.

As security is an essential part of .Net, it provides security support, such as code authenticity check, resources access authorizations, declarative and imperative security, and cryptographic security methods for embedding into the user’s application. In addition, it significantly extends the development platform by providing tools and technologies to develop internet-based distributed application.

The .NET framework is an integral windows component that supports building and running the next generation of applications and XML web services. The key component of the .NET frame work are the common language run time and the .NET frame work class library, which includes ADO.NET, ASP.NET and windows forms. The .NET framework provides a managed execution environment simplified development and deployment and integration with a wide variety of programming languages.

This framework is made up of the following parts:

* The common language runtime(CLR)
* The base class libraries.
* Object oriented internet development with ASP.NET
* Rich client user interface using windows forms
* RAD for the internet using web forms

The .NET framework is a new computing platform that simplifies application development in the highly distributed environment of the internet. The .NET framework is designed to fulfill following objectives:

* To provide a consistent object -oriented programming environment whether object code is stored and executed locally but internet- distributed or executed remotely.
* To provide a code execution environment that minimizes software deployment and versioning conflicts.
* To provide a code execution environment that guarantees safe execution of code, including code created by an unknown or semi trusted third party.
* To provide a code execution environment that eliminates the performance problem of scripted or interpreted environments.
* To make the developer experience consistent across widely types of application, such as windows-based applications and web- based applications.
* To build all communication on industry standards to ensure that code based on the .NET framework can integrate with any other code.

The .NET framework has two main components: the common language runtime and the .Net framework class library. The common language runtime is the foundation of the .NET framework. You can think of the runtime as an agent that manages code at execution time, and remoting while also enforcing strict type safely and other forms of code accuracy that ensure security and robustness in fact the concept of code management is a fundamental principle of the runtime.

Code that targets the runtime is known as managed code, while code that does not target the runtime

is known as unmanaged code. The class library, the other main component of the .NET frameworks is a

comprehensive, object-oriented collection reusable types that you can use to develop applications ranging

from traditional command line or graphical user interface (FGUI) applications to application base d on the

latest innovations provided by ASP.NET, such as web forms and XML web services.

The .NET framework can be hosted by unmanaged component that load the common language runtime into their processes and initiate the execution of managed code.

ASP.NET works directly with the runtime to enable ASP.NET application and XML web services, both of which are discussed later in this topic, Internet explorer is an example of unmanaged application that hosts the runtime (in the form of a MIME type extension).

Using internet explorer to the host runtime enables you to embed managed components or windows forms controls in HTML documents. Hosting the runtime in this way makes mobile code 9similar to Microsoft ActiveX controls) possible, but with significant improvement that only managed code can offer, such as semi-trusted execution and secure isolated file storage.

The following illustration shows the relationship of the common language runtime and the class library to your application and to the overall system. The illustration also shows how managed code operated with in a larger architecture.

We can use the .NET framework to develop the following types of application and services:

* Console applications
* Window GUI application (Windows Forms) ASP.NET applications
* XML Web services
* Windows services

**Common Language Runtime (CLR)**

The common language runtime (CLR) is responsible for runt-time services such as language integration; security enforcement; and memory, process and thread management. In addition, it has a roll at development time when features such as life cycle management strong type naming, cross-language exception handling, dynamic binding and so on, reduce the amount of code that a developer must write to turn the business logic the reusable component. The runtime can be hosted by high performance, server-side applications, such a s Microsoft Internet Information Services (IIS) for building web applications with ASP.NE and the next release of Microsoft SQL Server. This infrastructure enables you to use code “managed” by the .NET framework to write your business logic, while still enjoying the superior performance of the industry’s best enterprises servers that support runtime hosting.

**ASP.NET**

ASP.NET is a set of Microsoft.NET framework technologies used for building web applications and XML Web services. ASP.NET page execute on the server and generate markup such as HTML, WML or

XML that is sent to a desktop or mobile browser. ASP.NET pages use a compiled, event-driven programming model that improves performance and enables the separation of application logic and user interface. Both ASP.NET pages and ASP.NET web services files contain server-side (as opposed to client -side logic) written in Visual basic .NET, C#.NET or any .NET compatible language, Web applications and XML Web Services take advantage of the features of the common language runtime, such as type safety, inheritance, language, interoperability, versioning, and integrated security.

**IIS**

Microsoft Internet Information Services (IIS; formerly called Server is a set of Internet-based services for servers using Microsoft Windows. It is the world’s second most popular web server in terms of overall websites. As ofSeptember2007, it served 34.94% of all websites and 36.63% of all active websites according to Net craft. The servers currently include FTP, SMTP, NNTOP, and HTTP/HTTPS. **ADO.NET**

ADO.NET provides consistent access to data sources such a Microsoft SQL Server and XML, as well as to data sources exposed through OLE DB and ODBC. Data sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, manipulate and update the data that they contain.

ADO.NET separates data access from data manipulation into discrete components that can be used separately or in tandem. ADO.NET includes .NET Frame work data providers for connecting to a database, executing commands and retrieving results. Those results are either processed directly, placed in and ADO.NET Dataset objects in order to be exposed to the used in an ad hoc manner, combined with data from multiple sources or remote between tiers. The ADO.NET Dataset object can also be used independently of a .NET Framework data provider to manage data local to the application or sourced from XML.

The ADO.NET classes are found in System.Data.dll and are integrated with the XML classes found in System.Xml.dll. When compiling code that uses the System. Data, namespace reference both System.Data.dll and System.Xml.dll.

ADO.NET provided functionality to developers writing managed code similar to the functionality provided to native component object model (COM) developers by ActiveX Data Objects (ADO).

**ADO.NET COMPONENTS**

There are two components of ADO.NET that you can use to access and manipulate data:

* .NET Framework data providers.
* The Dataset

**.NET FRAMEWORK DATA PROVIDERS**

The .NET Framework Data providers are components that have been explicitly designed for data manipulation and fast, forward-only, read-only access to data. The connection object provides connectivity to a data source. The command object enables access to database commands to return data, modify data, run stored procedures and send or retrieve parameter information. The Data Adapter provides a high-performance stream of data from the data source. Finally, the Data Adapter provides the bridge between the Dataset object and the data source.

The Data Adapter uses command object to execute SQL commands at the data source to both load the Dataset with data and reconcile changes made to the data in the Dataset back to the data source.

**DATASET**

The ADO.NET Dataset is explicitly designed for data access independent of any data source. As a result, it can be used with multiple and differing data sources used with XML data or used to manage data local to the application. The Dataset contains a collection of one or more Data Table objects made up to rows and columns of data as well as primary key, foreign key, constraint and relation information about the data in the Data Table objects.

**BENEFITS OF ADO.NET**

ADO.NET offers several advantages over previous versions of ADO and over other data access components. These benefits fall into the following categories:

1. Interoperability
2. Maintainability
3. Programmability
4. Scalability

**C# language**

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with c, C++ or java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time.

As an object-oriented language, the C# supports the concepts of encapsulation, inheritance, data binding and polymorphism. All variables and methods, including the Main method, the application’s entry point, are encapsulated within class definitions.

**4.2 Back End- SQL Database 2008**

SQL stands for structured Query Language. SQL is used to communicate with a database. According to ANSI, it is the standard language for relational database management systems. SQL statements are used to perform task such as update data on a database, or retrieve data from a database.

However, the standard SQL commands such as “select”, “insert”, “update” “delete”, “Create”, and “Drop” can be used to accomplish almost ever

**CHAPTER-V**

**SYSTEM STUDY**

**5.1 FEASIBILITY STUDY**

Feasibility study is conducted once the problem is clearly understood. Feasibility study is a high-level capsule version of the entire system analysis and design process. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility is not to solve the problem but to determine if the problem is worth solving.

The system has been tested for feasibility in the following points.

1. Technical Feasibility

2. Operational Feasibility

3. Economical Feasibility.

**5.1.1. Technical Feasibility**

The project entitles "Web based tourism and travel planner” is technically feasibility because of the below mentioned feature. The project was developed in Java which Graphical User Interface.

It provides the high level of reliability, availability and compatibility. All these make Java an appropriate language for this project. Thus, the existing software Java is a powerful language.

**5.1.2. Operational Feasibility**

In this project, the management will know the details of each project where he may be presented and the data will be maintained as decentralized and if any inquires for that particular contract can be known as per their requirements and necessaries.

**5.1.3. Economical Feasibility**

The computerized system will help in automate the selection leading the profits and details of the organization. With this software, the machine and manpower utilization are expected to go up by 80-90% approximately. The costs incurred of not creating the system are set to be great, because precious time can be wanted by manually.

**Implementation**

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new success full system and the giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

**Testing**

The testing phase is an important part of software development. It is the computerized system will help in automated process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

Software testing is carried out in three steps:

**1**. The first includes unit testing, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether the objectives have been met. Errors are noted down and corrected immediately. Unit testing is the important and major part of the project. So, errors are rectified easily in particular module and program clarity is increased. In this project entire system is divided into several modules and is developed individually. So, unit testing is conducted to individual modules.

**2.** The second step includes Integration testing. It need not be the case, the software whose modules when run individually and showing perfect results, will also show perfect results when run as a whole. The individual modules are clipped under this major module and tested again and verified the results. This is due to poor interfacing, which may result in data being lost across an interface. A module can have inadvertent, adverse effect on any other or on the global data structures, causing serious probable

**3.** The final step involves validation and testing which determines which the software functions as the user expected. Here also some modifications were. In the completion of the project, it is satisfied fully by the end user.

**Maintenance and environment**

As the number of computer- based systems, grieve libraries of computer software began to expand. In house developed projects produced tones of thousand soft program source statements. Software products purchased from the outside added hundreds of thousands of new statements. A dark cloud appeared on the horizon. All of these programs, all of those source’s statements-had to be corrected when false were detected, modified as user requirements changed, or adapted to new hardware that was purchased. These activities were collectively called software Maintenance.

The maintenance phase focuses on change that is associated with error correction, adaptations required as the software's environment evolves, and changes due toenhancements brought about by changing customer requirements. Four types of changes are encountered during the maintenance phase.

* Correction
* Adaptation
* Enhancement
* Prevention

**Correction**

Even with the best quality assurance activities is lightly that the customer will uncover defects in the software. Corrective maintenance changes the software to correct defects.

Maintenance is a set of software Engineering activities that occur after software has been delivered to the customer and put into operation. Software configuration management is a set of tracking and control activities that began when a software project begins and terminates only when the software is taken out of the operation.

We may define maintenance by describing four activities that are undertaken after a program is released for use:

* Corrective Maintenance
* Adaptive Maintenance
* Perfective Maintenance or Enhancement
* Preventive Maintenance or reengineering

Only about 20 percent of all maintenance works are spent "fixing mistakes". The remaining 80 percent are spent adapting existing systems to changes in their external environment, making enhancements requested by users, and reengineering an application for use.

**Adaptation**

Over time, the original environment (E>G., CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate change to its external environment.

**Enhancement**

As software is used, the customer/user will recognize additional functions that will provide benefit. Perceptive maintenance extends the software beyond its original function requirements.

**Prevention**

Computer software deteriorates due to change, and because of this, preventive maintenance, often called software engineering, must be conducted to enable the software to serve the needs of its end users. In essence, preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted, and enhanced. Software configuration management (SCM) is an umbrella activity that is applied throughout the software process. SCM activities are developed to

**Software Methodology**

The software methodology followed in this project includes the object-oriented methodology and the application system development methodologies. The description of these methodologies is given below. Application System Development – A Life cycle - Approach

Although there are a growing number of applications (such as decision support systems) that should be developed using an experimental process strategy such as prototyping, a significant amount of new development work continues to involve major operational applications of broad scope. The application systems are large highly structured. User task comprehension and developer task proficiency is usually high. These factors suggest a linear or iterative assurance strategy. The most common method for this stage class of problems is a system development life cycle modal in which each stage of development is well defined and has straightforward requirements for deliverables, feedback and sign off. The system development life cycle is described in detail since it continues to be an appropriate methodology for a significant part of new development work.

The basic idea of the system development life cycle is that there is a well-defined process by which an

application is conceived and developed and implemented. The life cycle gives structure to a creative

process. In order to manage and control the development effort, it is necessary to know what should have

been done, what has been done, and what has yet to be accomplished. The phrases in the system

development life cycle provide a basis for management and control because they define segments of the

flow of work, which can be identified for managerial purposes and specifies the documents or other

deliverables to be produced in each phase.

The phases in the life cycle for information system development are described differently by different writers, but the differences are primarily in the amount of necessity and manner of categorization. There is a general agreement on the flow of development steps and the necessity for control procedures at each stage.

The information system development cycle for an application consists of three major stages.

1) Definition.

2) Development.

3) Installation and operation

The requirements are then translated into a physical system of forms, procedures, programs etc., by the system design, computer programming and procedure development. The resulting system is test and put into operation. No system is perfect so there is always a need for maintenance changes. To complete the cycle, there should be a post audit of the system to evaluate how well it performs and how well it meets the cost and performance specifications. The stages of definition, development and installation and operation can therefore be divided into smaller steps or phrases as follows.

**Definition**

Proposed definition : preparation of request for proposed applications.

Feasibility assessment : evaluation of feasibility and cost benefit of proposed system.

Information requirement analysis : determination of information needed.

**Design**

Conceptual design : User-oriented design of application development.

Physical system design : Detailed design of flows and processes in applications processing system and preparation of program specification.

**Development**

Program development : coding and testing of computer programs.

Procedure development : design of procedures and preparation of user instructions.

**Installation and operation**

Conversion : final system test and conversion.

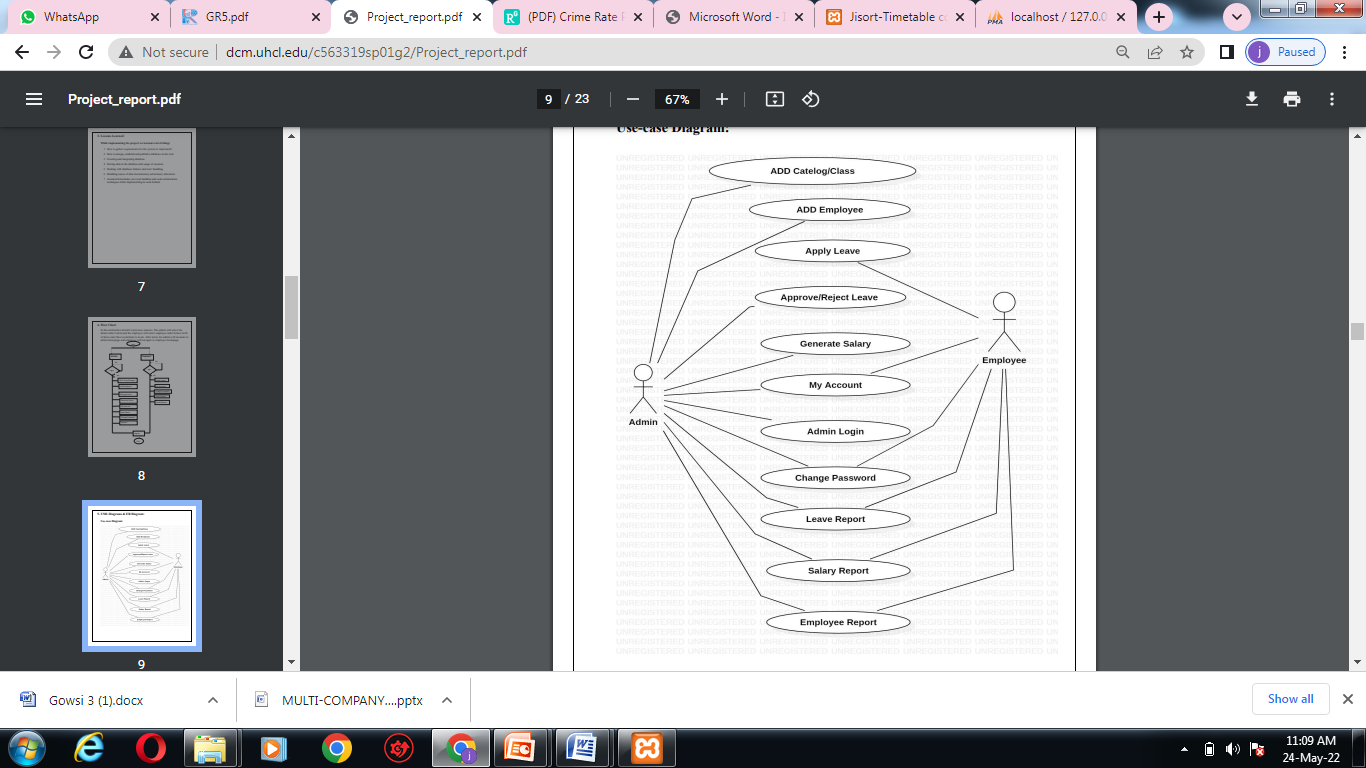
Operation and maintenance : Month to month operation and maintenance

Post audit : Evaluation of development process, application system and results of use at the completion of the each phase, formal approval sign-off is required from the users as well as from the manager of the project development.

**CHAPTER -VI**

**SYSTEM DESIGN**

**6.1 USECASE DIAGRAM**

****

**CHAPTER-VII**

**SYSTEM IMPLEMENTATION**

**7.1 MODULE DESCRIPTION**

Modules

♣Administrator

♣Student

**ADMIN:**

* The admin gets logged in by valid username and password. Admin can add new Employee, add new Department, add new Pay Grade for the employees. Admin can set the ‘from’ and ‘to’ date worked by an employee in a department with specific pay grade. The admin can generate an automated monthly salary of an employee. The admin can view all the past records of any recorded employee. The admin can check leave and he can approve and reject employee leave. According to approve leave the leave deduction will be deducted from salary.
* **HR:**

HR has a unique username and password to login in to system. After login in to system all employee can apply for a leave with valid reason. HR can add employee details and leave details.

**7.2 TESTING**

Testing is a process of executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding. System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing. A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

**Testing Objectives**

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

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

3. A successful test is one that uncovers an undiscovered error.

**Testing Principle**

1. All tests should be traceable to end user requirements.

2. Tests should be planned long before testing begins.

3. Testing should begin on a small scale and progress towards testing in large.

4. Exhaustive testing is not possible.

5. To be most effective testing should be conducted by an independent third party.

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are

* White box testing
* Black box testing

**White-box testing**

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

**Black-box testing**

Black box testing is designed to validate functional requirements without regard to the internalworkings of a program. Black box testing mainly focuses on the information domain of the software,deriving test cases by partitioning input and output in a manner that provides through test coverage.Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

**Testing strategies**

A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

**Testing fundamentals**

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully, it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

**Testing Information flow**

Information flow for testing flows the pattern. Two class of input provided to test the process. The software configuration includes a software requirements specification, a design specification and source code. Test configuration includes test plan and test cases and test tools. Tests are conducted and all the results are evaluated. That is test results are compared with expected results. When erroneous data are uncovered, an error is implied and debugging commences.

**7.2.1 Unit testing**

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors within the boundary of the modules. These tests were carried out during the programming stage itself. All units of SQL were successfully tested.

**7.2.2 Integration testing**

Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.

**7.2.3 System testing**

System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and its original objective, current specification and system documentation. The primary concern is the compatibility of individual modules. Entire system is working properly or not will be tested here, and specified path ODBC connection will correct or not, and giving output or not are tested here these verifications and validations are done by giving input values to the system and by comparing with expected output. Top-down test implementing here.

**7.2.4 Acceptance Testing**

This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

Tools to special importance during acceptance testing include:

Test coverage Analyzer – records the control paths followed for each test case.

Timing Analyzer – also called a profiler, reports the time spent in various regions of the code are areas to concentrate on to improve system performance.

Coding standards – static analyzers and standard checkers are used to inspect code for deviations from standards and guidelines.

**Test Cases**

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Using White-Box testing methods, the software engineer can drive test cases that

Guarantee that logical decisions on their true and false sides.

Exercise all logical decisions on their true and false sides.

Execute all loops at their boundaries and within their operational bounds.

Exercise internal data structure to assure their validity.

The test case specification for system testing has to be submitted for review before system testing commences.

**CHAPTER-VIII**

**CONCLUSION**

This project is built keeping in mind that it is to be used by only one user that is the admin. It is built for

use in small scale organization where the number of employees is limited. According to the requested

requirement the admin can add, manipulate, update and delete all employee data in his organization. The

admin can add new departments and delete them. The Admin can also add predefined pay grades for the

employees. The required records can be easily viewed by the admin anytime time he wants in an instant.

The payment of the employee is based on monthly basis. Numerous validations implemented would enable

the admin to enter accurate data. The main objective of this framework is to save time, make the system cost

effective and management records efficiently.

**REFERENCES**

**TEXTBOOK**

**1.** Andrew Troelsen , C# and .NetPlatform,Apress,1st edition,2001.

1. StephenWalther (2008), ASP.NET Unleashed, India: Dorling Kindersley.
2. Ramesh Bangia, ASP.NET and C#,5th edition,2004.
3. Sivaprasathkoirala, C# and ASP.NET Projects, BPB Publication.
4. J.G.R. Sathiaseelan, N. Sasikaladevi, Programming with C# .NET

**WEBSITES**

**1.**<https://www.w3schools.com/ASP>

**2.**[www.tutorialteacher.com](http://www.tutorialteacher.com)

**3**. [www.tutorialpoint.com](http://www.tutorialpoint.com)

**4.**[www.csharpcorner.com](http://www.csharpcorner.com)

**5**.[www.dotnettricks.com](http://www.dotnettricks.com)

**APPENDIX**

**SOURCE CODE**

**Home**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile=”[Home.aspx.cs](http://Home.aspx.cs)”

Inherits="Home" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>MULTI-COMPANY PAYROLL SYSTEM</title>

<meta name="keywords" content="free css template, orange bar, XHTML, CSS" />

<meta name="description" content="Orange Bar is a free CSS template from

templatemo.com" />

<link href="templatemo\_style.css" rel="stylesheet" type="text/css" />

<script language="javascript" src="templatemo\_image\_fader.js"

type="text/javascript"></script>

<script language="javascript" type="text/javascript">

function clearText(field) {

if (field.defaultValue == field.value) field.value = '';

else if (field.value == '') field.value = field.defaultValue;

}

</script>

<style type="text/css">

.style1

{

width: 100%;

}

</style>

</head>

<body>

<div class="templatemo\_container">

<div id="templatemo\_header">

<div id="templatemo\_logo\_area">

<div id="templatemo\_logo">

<a target="\_blank"><center>MULTI-COMPANY PAYROLL

SYSTEM</center></a></div>

<div id="templatemo\_slogan">

</div>

<div class="cleaner"></div>

</div>

<div id="templatemo\_menu">

<ul>

<li class="current"><a href="Home.aspx">Home</a></li>

<li ><a href="Login.aspx">Login</a></li>

<li><a href="About Us.aspx">About Us</a></li>

<li><a href="Contact Us.aspx">Contact Us</a></li>

</ul>

</div>

</div><!-- end of header -->

<div id="templatemo\_image\_fader">

<script type="text/javascript">

//new fadeshow(IMAGES\_ARRAY\_NAME, slideshow\_width, slideshow\_height, borderwidth, delay, pause (0=no, 1=yes), optionalRandomOrder)

new fadeshow(fadeimages, 960, 240, 1, 2000, 1, "R")

</script>

</div>

<div id="templatemo\_content\_area">

<div style="border: medium groove #808080">

<table class="style1">

<tr>

<td colspan="5">

<a href="#">

<img src="imag/Untitled\_design\_0ddc3c2b-42d6-4bb9-aec7-

c16fd67fb371.png"

alt="Yellow Rose" width="920" height="300" /></a>&nbsp;</td>

</tr>

<tr>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

</tr>

<tr>

<td colspan="4" rowspan="2">

<p style="font-family:'Times New Roman',Times,serif; font size: large; color: #000000; font-weight: bold;">If you’ve ever hired an employee, considered hiring one, or been employed by someone else, you’ve probably come across the term “payroll.”</p></td>

<td>

&nbsp;</td>

</tr>

<tr>

<td>

&nbsp;</td>

</tr>

<tr>

<td colspan="4">

<p style="font-family: 'Times New Roman', Times, serif; font-size: large; color: #000000; font-weight: bold;"> But as a business owner, you may be unsure of where to begin when it comes to the payroll process. Topics like how to do payroll and how to calculate payroll taxes can quickly feel overwhelming and intimidating, particularly if you’re hiring your first employee. And with so much information out there, it can be hard to figure out where to start.</p></td>

<td>

&nbsp;</td>

</tr>

<tr>

<td colspan="4">

<p style="font-family: 'Times New Roman', Times, serif; font-size: large; color: #000000; font-weight: bold;"> This guide to the payroll process for retailers will walk through some of the most common questions and advice for getting started.</p></td>

<td>

&nbsp;</td>

</tr>

<tr>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

</tr>

<tr>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

<td>

&nbsp;</td>

</tr>

</table>

</div>

</div>

&nbsp;

<!-- End Of right-->

<div class="cleaner"></div>

</div<!—End Of Content area -->

<div id="templatemo\_bottom\_area">

<div class="templatemo\_container”>

<div id="templatemo\_footer">

Copyright © 2048 <a href="#">Automated payroll System with location based services</a>

</div>

</div>

</div>

<!-- Free CSS Templates by TemplateMo.com -->

</body>

</html>

ADMIN LOGIN

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.IO;

public partial class Default8 : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString);

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

{

}

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

{

string Candidatename = TextBox1.Text;

string Age = TextBox2.Text;

string Address = TextBox3.Text;

string Year = TextBox4.Text;

string Partyname = TextBox5.Text;

string Election = DropDownList1.Text;

string Area = TextBox6.Text;

string Date = TextBox7.Text;

byte[] bytes;

using (BinaryReaderbr = new BinaryReader(FileUpload1.PostedFile.InputStream))

{

bytes = br.ReadBytes(FileUpload1.PostedFile.ContentLength);

}

//string constr=ConfigurationManager.ConnectionStrings["constr"].ConnectionString;

using (SqlConnection con = new SqlConnection(ConfigurationManager ["ConnectionString"].ConnectionString))

{

string sql = "INSERT INTO nac VALUES(@Candidatename,@Age,@Address,@Year,@Partyname,@Election,@Area,@Date,@Partyimage,@Imagedata)";

using (SqlCommandcmd = new SqlCommand(sql, con))

{

cmd.Parameters.AddWithValue("@Candidatename", Candidatename);

cmd.Parameters.AddWithValue("@Age", Age);

cmd.Parameters.AddWithValue("@Address", Address);

cmd.Parameters.AddWithValue("@Year", Year);

cmd.Parameters.AddWithValue("@Partyname", Partyname);

cmd.Parameters.AddWithValue("@Election", Election);

cmd.Parameters.AddWithValue("@Area", Area);

cmd.Parameters.AddWithValue("@Date", Date);

cmd.Parameters.AddWithValue("@Partyimage", Path.GetFileName(FileUpload1.PostedFile.FileName));

cmd.Parameters.AddWithValue("@Imagedata", bytes);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

//Response.Write("<script>alert(' Registration Successfully')</script>");

}

}

Response.Redirect(Request.Url.AbsoluteUri);

}

}

ADD EMPLOYEE

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.IO;

public partial class Add\_Employee : System.Web.UI.Page

{

SqlConnection con = new

SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString);

string empid = "Emp000";

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

{

GenerateAutoID();

Button5.Visible = false;

}

private void GenerateAutoID()

{

con.Open();

SqlCommandcmd = new SqlCommand("select count(Emp\_ID) from employee", con);

int i = Convert.ToInt32(cmd.ExecuteScalar());

con.Close();

i++;

TextBox1.Text = empid + i.ToString();

}

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

{

string FileName = Path.GetFileName(FileUpload1.PostedFile.FileName);

//Save files to images folder

FileUpload1.SaveAs(Server.MapPath("Upload/" + FileName));

this.Image1.ImageUrl = "Upload/" + FileName;

using (Stream fs = FileUpload1.PostedFile.InputStream)

{

using (BinaryReaderbr = new BinaryReader(fs))

{

byte[] bytes = br.ReadBytes((Int32)fs.Length);

using (SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString))

{

using (SqlCommandcmd = new SqlCommand("insert into employee(Emp\_ID,F\_Name,L\_Name,Gender,DOB,Address,Contact,Position,DOJ,Basic,Photo) values('" + TextBox1.Text + "','" + TextBox2.Text + "','" + TextBox3.Text + "','" + DropDownList1.Text + "','" + TextBox4.Text + "','" + TextBox5.Text + "','" + TextBox6.Text + "','" + TextBox7.Text + "','" + TextBox8.Text + "','" + TextBox9.Text + "',@Image)"))

{

cmd.Connection = con;

cmd.Parameters.AddWithValue("@Image", bytes);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

}

}

}

Response.Redirect(Request.Url.AbsoluteUri);

}

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

{

string FileName = Path.GetFileName(FileUpload1.PostedFile.FileName);

//Save files to images folder

FileUpload1.SaveAs(Server.MapPath("Upload/" + FileName));

this.Image1.ImageUrl = "Upload/" + FileName;

}

protected void Button2\_Click(object sender, EventArgs e)

{

con.Open();

SqlCommandcmd = new SqlCommand("update employee set

F\_Name='"+TextBox2.Text +"',L\_Name='"+TextBox3.Text

+"',Gender='"+DropDownList1.Text +"',DOB='"+TextBox4.Text

+"',Address='"+TextBox5.Text +"',Contact='"+TextBox6.Text

+"',Position='"+TextBox7.Text +"',DOJ='"+TextBox8.Text +"',Basic='"+TextBox9.Text +"'

where Emp\_ID='"+TextBox1.Text +"'",con);

int i= cmd.ExecuteNonQuery();

if (i == 1)

{

string script = "alert('Employee details updated Successfully....');";

ClientScript.RegisterClientScriptBlock(this.GetType(), "Alert", script, true);

}

else

{

string script = "alert('Employee details not updated ....');";

ClientScript.RegisterClientScriptBlock(this.GetType(), "Alert", script, true);

}

}

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

{

con.Open();

SqlCommandcmd = new SqlCommand("delete from employee where Emp\_ID='" + TextBox1.Text + "'", con);

int i = cmd.ExecuteNonQuery();

if (i == 1)

{

string script = "alert('Employee details Deleted Successfully....');";

ClientScript.RegisterClientScriptBlock(this.GetType(), "Alert", script, true);

}

else

{

string script = "alert('Employee details not Delete ....');";

ClientScript.RegisterClientScriptBlock(this.GetType(), "Alert", script, true);

}

}

protected void Button5\_Click(object sender, EventArgs e)

{

con.Open();

SqlCommandcmd = new SqlCommand("select \* from employee where Emp\_ID='" + TextBox1.Text + "'", con);

SqlDataReaderrdr = cmd.ExecuteReader();

while (rdr.Read())

{

TextBox1.Text = rdr["Emp\_ID"].ToString().Trim();

TextBox2.Text = rdr["F\_Name"].ToString().Trim();

TextBox3.Text = rdr["L\_Name"].ToString().Trim();

DropDownList1.Text = rdr["Gender"].ToString().Trim();

TextBox4.Text = rdr["DOB"].ToString().Trim();

TextBox5.Text = rdr["Address"].ToString().Trim();

TextBox6.Text = rdr["Contact"].ToString().Trim();

TextBox7.Text = rdr["Position"].ToString().Trim();

TextBox8.Text = rdr["DOJ"].ToString().Trim();

TextBox9.Text = rdr["Basic"].ToString().Trim();

}

con.Close();

}

}

PAYROLL

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient ;

using System.Configuration;

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

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString);

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

{

}

protected void TextBox6\_TextChanged(object sender, EventArgs e)

{

}

protected void TextBox11\_TextChanged(object sender, EventArgs e)

{

int a = Convert.ToInt32(TextBox7.Text);

int b = Convert.ToInt32(TextBox8.Text);

int c = Convert.ToInt32(TextBox9.Text);

int d = Convert.ToInt32(TextBox10.Text);

int f = a - b;

int g = f - c;

int h = g - d;

TextBox12.Text = Convert.ToString(h);

}

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

{

SqlCommandcmd = new SqlCommand("insert into

payroll(Emp\_Id,F\_Name,L\_Name,Position,DOJ,Working\_Days,Basic\_Pay,Cross\_Pay,Cash\_Advance,PF,Insurance,Deduction,Netpay)values('" + DropDownList1.Text + "','"

TextBox1.Text + "','" + TextBox2.Text + "','" + TextBox3.Text + "','" + TextBox4.Text + "','" + TextBox5.Text + "','" + TextBox6.Text + "','" + TextBox7.Text + "','" + TextBox8.Text + "','" + TextBox9.Text + "','" + TextBox10.Text + "','" + TextBox11.Text + "','" + TextBox12.Text + "')", con);

con.Open();

int i = cmd.ExecuteNonQuery();

if (i == 1)

{

string script = "alert('Payroll added Successfully');";

ClientScript.RegisterClientScriptBlock(this.GetType(), "Alert", script, true);

}

else

{

string script = "alert('Payroll not added ');";

ClientScript.RegisterClientScriptBlock(this.GetType(), "Alert", script, true);

}

}

protected void DropDownList1\_SelectedIndexChanged1(object sender, EventArgs e)

{

GenerateAutoID();

GenerateAuto();

con.Open();

SqlCommandcmd = new SqlCommand("select \* from employee where Emp\_ID='" + DropDownList1.Text + "'", con);

SqlDataReaderrdr = cmd.ExecuteReader();

while (rdr.Read())

{

TextBox1.Text = rdr["F\_Name"].ToString().Trim();

TextBox2.Text = rdr["L\_Name"].ToString().Trim();

TextBox3.Text = rdr["Position"].ToString().Trim();

TextBox4.Text = rdr["DOJ"].ToString().Trim();

TextBox6.Text = rdr["Basic"].ToString().Trim();

}

con.Close();

}

private void GenerateAutoID()

{

con.Open();

SqlCommand cmd1 = new SqlCommand("select \* from expanses where Emp\_ID='" + DropDownList1.Text + "'", con);

SqlDataReader rdr1 = cmd1.ExecuteReader();

while (rdr1.Read())

{

TextBox9.Text = rdr1["PF"].ToString().Trim();

TextBox10.Text = rdr1["Insurance"].ToString().Trim();

}

con.Close();

}

private void GenerateAuto()

{

con.Open();

SqlCommand cmd1 = new SqlCommand("select \* from advance where Emp\_ID='" + DropDownList1.Text + "'", con);

SqlDataReader rdr1 = cmd1.ExecuteReader();

while (rdr1.Read())

{

TextBox8.Text = rdr1["Amount"].ToString().Trim();

}

con.Close();

}

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

{

}

protected void Button5\_Click(object sender, EventArgs e)

{

int a = Convert.ToInt32(TextBox6.Text);

int b = a / 30;

int d = Convert.ToInt32(TextBox5.Text);

int f = b \* d;

TextBox7.Text = Convert.ToString(f);

}

protected void TextBox5\_TextChanged(object sender, EventArgs e)

{

int a = Convert.ToInt32(TextBox6.Text);

int b = a / 30;

int d = Convert.ToInt32(TextBox5.Text);

int f = b \* d;

TextBox7.Text = Convert.ToString(f);

}

protected void Button2\_Click(object sender, EventArgs e)

{

SqlCommandcmd = new SqlCommand("update payroll set F\_Name='" +

TextBox1.Text + "',L\_Name='" + TextBox2.Text + "',Position='" + TextBox3.Text +

"',DOJ='" + TextBox4.Text + "',Working\_Days'" + TextBox5.Text + "',Basic\_Pay='" +

TextBox6.Text + "',Cross\_Pay='" + TextBox7.Text + "',Cash\_Advance='" + TextBox8.Text + "',PF='" + TextBox9.Text + "',Insurance='" + TextBox10.Text + "',Deducion='" + TextBox11.Text + "',Netpay='" + TextBox12.Text + "' where Emp\_Id='" + DropDownList1.Text + "')", con);

con.Open();

int i = cmd.ExecuteNonQuery();

if (i == 1)

{

string script = "alert('Payroll updated Successfully');";

ClientScript.RegisterClientScriptBlock(this.GetType(), "Alert", script, true);

}

else

{

String script=”alert(‘Payroll updated Successfully’);”;

ClientScript.RegisterClientScriptBlock(this.GetType(), "Alert", script, true);

}

}

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

{

SqlCommandcmd = new SqlCommand("delete from payroll where Emp\_Id='" + DropDownList1.Text + "')", con);

con.Open();

int i = cmd.ExecuteNonQuery();

if (i == 1)

{

string script = "alert('Payroll deleted Successfully');";

ClientScript.RegisterClientScriptBlock(this.GetType(), "Alert", script, true);

}

else

{

string script = "alert('Payroll not delete ');";

ClientScript.RegisterClientScriptBlock(this.GetType(), "Alert", script, true);

}

}

}

VIEW

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

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

{

SqlConnection con = new

SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString);

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

{

}

protected void DropDownList1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

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

{

string constr = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;

using (SqlConnection conn = new SqlConnection(constr))

{

using (SqlDataAdaptersda = new SqlDataAdapter("SELECT \* FROM camera where Emp\_ID='" + DropDownList1.Text + "'", conn))

{

DataTable dt = new DataTable();

sda.Fill(dt);

gvImages.DataSource = dt;

gvImages.DataBind();

}

}

}

protected void OnRowDataBound(object sender, GridViewRowEventArgs e)

{

if (e.Row.RowType == DataControlRowType.DataRow)

{

DataRowViewdr = (DataRowView)e.Row.DataItem;

string imageUrl = "data:image/jpg;base64," + Convert.ToBase64String((byte[])dr["Photo"]);

(e.Row.FindControl("Image1") as Image).ImageUrl = imageUrl;

}

}

}

Records

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data;

using System.Configuration;

using iTextSharp.text;

using iTextSharp.text.pdf;

using iTextSharp.text.html.simpleparser;

using System.IO;

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

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString);

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

{

}

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

{

using (con)

{

using (SqlCommandcmd = new SqlCommand())

{

cmd.CommandText = "select \* from employee";

cmd.Connection = con;

con.Open();

GridView1.DataSource = cmd.ExecuteReader();

GridView1.DataBind();

con.Close();

}

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

using (con)

{

using (SqlCommandcmd = new SqlCommand())

{

cmd.CommandText = "select \* from payroll";

cmd.Connection = con;

con.Open();

GridView1.DataSource = cmd.ExecuteReader();

GridView1.DataBind();

con.Close();

}

}

}

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

{

using (con)

{

using (SqlCommandcmd = new SqlCommand())

{

cmd.CommandText = "select \* from advance";

cmd.Connection = con;

con.Open();

GridView1.DataSource = cmd.ExecuteReader();

GridView1.DataBind();

con.Close();

}

}

}

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

{

ExportGridToPDF();

}

public override void VerifyRenderingInServerForm(Control control)

{

//required to avoid the runtime error "

//Control 'GridView1' of type 'GridView' must be placed inside a form tag with runat=server."

}

private void ExportGridToPDF()

{

Response.ContentType = "application/pdf";

Response.AddHeader("content-disposition", "attachment;filename=Vithal\_Wadje.pdf");

Response.Cache.SetCacheability(HttpCacheability.NoCache);

StringWritersw = new StringWriter();

HtmlTextWriterhw = new HtmlTextWriter(sw);

GridView1.RenderControl(hw);

StringReadersr = new StringReader(sw.ToString());

Document pdfDoc = new Document(PageSize.A4, 10f, 10f, 10f, 0f);

HTMLWorkerhtmlparser = new HTMLWorker(pdfDoc);

PdfWriter.GetInstance(pdfDoc, Response.OutputStream);

pdfDoc.Open();

htmlparser.Parse(sr);

pdfDoc.Close();

Response.Write(pdfDoc);

Response.End();

GridView1.AllowPaging = true;

GridView1.DataBind();

}

}

Output





