

“HUMAN RESOURCE MANAGEMENT SYSTEM”

Project Report submitted in partial fulfillment of the requirements for the award of the degree of

## BACHELOR OF COMPUTER APPLICATIONS (BCA)

****

*Submitted By*

**ISMAIL PM**

**22BCAC32**

Under the guidance of

## Prof. Dr M SUDHARSAN

**Department Of Computer Science (UG)**

**BCA Programme**

**Kristu Jayanti College (Autonomous)**

**K.Narayanapura,Kothanur P.O.,Bangalore-560077**



**DEPARTMENT OF COMPUTER SCIENCE (UG)**

# CERTIFICATE OF COMPLETION

This is to certify that the project entitled **“HUMAN RESOURCE MANAGEMENT SYSTEM”** has been satisfactorily completed by **ISMAIL PM, 22BCAC34** in partial fulfillment of the award of the Bachelor of Computer Applications degree requirements prescribed by Kristu Jayanti College (Autonomous) Bengaluru (Affiliated to Bangalore North University) during the academic year 2024 -25.

***Internal Guide Head of the Department***

***Valued by Examiners***

**1: *Centre:* Kristu Jayanti College**

**2: *Date:***



## DECLARATION

I, **ISMAIL PM**, **22BCAC34** hereby declare that the project work entitled ‘HUMAN RESOURCE MANAGEMENT SYSTEM’ is an original project work carried out by me, under the guidance of **PROF. DR M. SUDHARSAN**.

This project work has not been submitted earlier either to any University / Institution or any other body for the fulfillment of the requirement of a course of study.

Signature (ISMAIL PM.)

BengaluruDate:

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Topic** | **Page No** |
| 1  1.1  1.2  2  2.1  2.2  2.3  3  3.1  3.2  3.3  3.4  4  4.1  4.2  5  5.1  5.2  5.3  6  7  8  9  10  11 | Introduction  Problem Definition Scope of the Project  System Study  Existing System Feasibility Study Proposed System  System Design  ER Diagram DFD [lvl0, lvl1] Gantt Chart  Input / Output Design System Configuration  Hardware Requirements Software Requirements  Details of Software  Overview of Frontend Overview of Backend About the Platform  Testing  Conclusion and Future Enhancement Bibliography  APPENDICES A-Table Structure APPENDICES B-Screenshots APPENDICES C-Sample Report of test cases | 1-2  1  1  2  3  3  4  5- 7  7-11  12-18  19-22  23-51  52  52  52  53  53-55  55-58  59-60  61-64  65-66  67  68-71  72-76  77-80 |

**ACKNOWLEDGEMENT**

The success of the project depends upon the efforts invested. It’s my duty to acknowledge and thank the individuals who has contributed in the successful completion of the project.

I take this opportunity to express my profound and whole hearted thanks to **Rev. Fr. Dr. AUGUTINE GEORGE, PRINCIPAL, KRISTU JAYANTI COLLEGE** for providing ample facilities made to undergo my project successfully.

I express my deep sense of gratitude and sincere thanks to our Head of the Department **Prof. SEVUGA PANDIAN** for his valuable advice.

I feel immense pleasure to thank my respected guide **Prof. Dr. M SUDHARSAN** for sustaining Interest and providing dynamic guidance in aiding me to complete this project immaculately and impeccably and for being the source of my strength and confidence.

It is my duty to express my thanks to all Teaching and Non-Teaching Staff members of Computer science department who offered me help directly or indirectly by their suggestions.

The successful completion of my project would not have been possible without my parent’s Sacrifice, guidance and prayers. I take this opportunity to thank everyone for their continuous Encouragement. I convey my thankfulness to all my friends who were with me to share my happiness and agony.

Last but not the least I thank almighty God for giving me strength and good health throughout my project and enabling me to complete it successfull

# 1 INTRODUCTION

## 1PROBLEM DEFINITION:

The existing Human Resources Management System (HRMS) allows users to log in and access features based on their roles. Administrators can manage employee and department records, publish news and announcements, track employee performance, handle payroll and benefits administration, and integrate with other enterprise systems for data consistency and efficiency. Overall, it streamlines HR processes.Supports strategic HR decision making.

## SCOPE OF THE PROJECT:

## Functional Scope:

## Define the functional scope of the project by specifying the features and functionalities that the new HRMS will include. This could include modules for employee data management, payroll processing, benefits administration, attendance tracking, performance management, recruitment, and employee self-service.

## Technical Scope:

## Define the technical scope of the project by specifying the technologies, platforms, and infrastructure requirements for the new HRMS. This could include decisions regarding the programming languages, frameworks, databases, hosting environment (on-premises or cloud-based), and integration protocols.

## Inclusions and Exclusions:

## Clearly outline what is included and excluded from the project scope to manage stakeholder expectations and avoid scope creep. This could involve listing specific features or functionalities that will be developed or enhanced in the new HRMS and specifying any functionalities or integrations that will not be addressed in the current project phase.

## Timeline and Resource Constraints:

## Define the project timeline and resource constraints, including deadlines for key milestones, allocation of human and financial resources, and dependencies on external factors such as regulatory approvals or vendor contracts.

## Risk Management:

## Identify potential risks and uncertainties that could impact the project's success, such as technical challenges, resource constraints, changes in regulatory requirements, or unforeseen dependencies. Develop strategies to mitigate these risks and ensure project resilience.

## MODULES IN THE PROJECT:

## Admin

## Employee

## Managing Employee information

## Managing Performance, Salary

## Managing Department

## Managing News

**ADMIN:** The admin manages the whole system which include the information of Employee, Department, Performance, Salary, News.

**EMPLOYEE:** He can only view the Performance and News

**PERFORMANCE:** This module contains the performance of the employee.

**DEPARTMENT:** This module contains various information about department like dept\_id,dept\_name.

**NEWS:** This module contains about the updates and day to day announcement.

# SYSTEM STUDY

## 2.1 EXISTING SYSTEM:

## The existing Human Resources Management System (HRMS) serves as a centralized platform for managing various HR functions and processes. It includes modules for employee information management, recruitment, onboarding/offboarding, time and attendance tracking, performance management, payroll administration, benefits management, employee self-service, manager self-service, reporting and analytics, and compliance management. The HRMS streamlines HR processes, facilitates communication, supports decision-making, and ensures compliance with regulations. Overall, it plays a crucial role in optimizing HR operations and workforce management within the organization.

## 2.2 FEASIBILITY STUDY:

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it.

A feasibility study tests the viability of an idea, a project or even a new business. The goal of a feasibility study is to place emphasis on potential problems that could occur if a project is pursued and determine if, after all significant factors are considered, the project should be pursued.

The project “**HUMAN RESOURCE MANAGEMENT SYSTEM S”** can be designed and developed using .Net framework. The content, language is feasible to use and the portal can be developed based on the requirements.

Components of feasible study:

Technical feasibility

Operational feasibility

Schedule Feasibility

Every project is feasible for given unlimited resources and infinitive time. Feasibility study is an evaluation of the proposed system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources.

Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. Feasibility and risk analysis are related in many ways. The feasibility analysis in this project has been discussed below based on the above mentioned components of feasibility.

## SCHEDULE:

Time duration of this project requires 56 days covering 7 days for initiation phase, 10 days for definition phase, 7 days for Design phase, 18 days for implementation phase, 7 days for Testing phase and 7 days for documentation.

Resource person is ISMAIL PM.

## OPERATIONAL:

Reduction of paper work.

Human effort or Manual Labor can be reduced drastically.

Major operations that are done manually can be done within a matter of seconds. Data storing is easier.

## 2.3 PROPOSED SYSTEM:

The proposed system has got the following advantages over the existing system:-

Allows to access the data of Employee

Manages Department

Manages Performances of Employee.

Easy to handle.

And lastly user friendly

# 

# 3.SYSTEM DESIGN

In the design phase the architecture is established. This phase starts with the requirement document delivered by the requirement phase and maps the requirements into an architecture. The architecture defines the components, their interfaces and behaviours. The deliverable design document is the architecture. The design document describes a plan to implement the requirements. This phase represents the ``how'' phase.

Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established.

The design may include the usage of existing components. Analysing the trade-offs of necessary complexity allows for many things to remain simple which, in turn, will eventually lead to a higher quality product. The architecture team also converts the typical scenarios into a test plan. In our approach, the team, given a complete requirement document, must also indicate critical priorities for the implementation team.

A critical implementation priority leads to a task that has to be done right. If it fails, the product fails. If it succeeds, the product might succeed. At the very least, the confidence level of the team producing a successful product will increase. This will keep the implementation team focused. Exactly how this information is conveyed is a skill based on experience more than a science based on fundamental foundations. System design is the process of defining the architecture components, modules, interfaces, and data for a system to satisfy specified requirements.

Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis , systems architecture and systems engineering. If the broader topic of product development blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

Until the 1990s, systems design had a crucial and respected role in the data processing industry. In the 1990s, standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software enginee[ring.](https://en.wikipedia.org/wiki/Software_engineering)

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modelling software systems and is increasingly used for high designing non software systems and organizations.

### ARCHITECTURAL DESIGN:

The architectural design of a system emphasizes the design of the system architecture that describes the structure, behavior and more views that system and analysis.

### LOGICAL DESIGN:

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams).

### PHYSICAL DESIGN:

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided.

Input requirement, Output requirements, Storage requirements, Processing requirements,

System control and backup or recovery.

Put another way, the physical portion of system design can generally be broken down into three sub- tasks:

User Interface Design. Data Design.

Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system.

Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system.

At the end of the system design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase. Physical design, in this context, does not refer to the tangible physical design of an information system.

To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc.

It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc.

It involves a detailed design of a user and a product database structure processor and a control processor. The H/S personal specification is developed for the proposed system.

# 3.1 E-R DIAGRAM:

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique. Because this ER tutorial focuses on beginners.

Below are some tips that will help you build effective ER diagrams:

Identify all the relevant entities in a given system and determine the relationships among these entities.

An entity should appear only once in a particular diagram.

Provide a precise and appropriate name for each entity, attribute, and relationship in the diagram.

Terms that are simple and familiar always beats vague, technical-sounding words.

In naming entities, remember to use singular nouns. However, adjectives may be used to distinguish entities belonging to the same class (part-time employee and full -time employee, for example).

Meanwhile attribute names must be meaningful, unique, system independent, and easily understandable.

Remove vague, redundant or unnecessary relationships between entities. Never connect a relationship to another relationship.

You can use colours to classify similar entities or to highlight key areas in your diagrams. You can draw entity relationship diagrams manually, especially when you are just informally showing simple systems to your peers.

However, for more complex systems and for external audiences, you need diagramming software such as Creately’s to craft visually engaging and precise ER diagrams.

The ER Diagram Software offered by Creately as an online service is pretty easy to use and is a lot more affordable than purchasing licensed software.

It is also perfectly suited for development teams because of its strong support for collaboration.

## THE HISTORY OF ENTITY RELATIONSHIP DIAGRAMS:

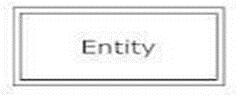
Peter Chen developed ERDs in 1976. Since then Charles Bachman and James Martin have added some slight refinements to the basic ERD principles.

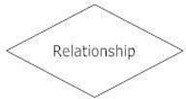
## STRUCTURE OF ENTITY RELATIONSHIP DIAGRAM WITH COMMON ERD:

**Notations:** An entity relationship diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

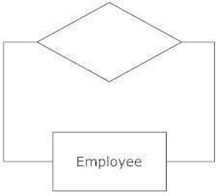
**Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information.

A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



 **Actions**, which are represented by diamond shapes, show how two entities share information in the database.

In some cases, entities can be self-linked. For example, employees can supervise other employees.



**Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.



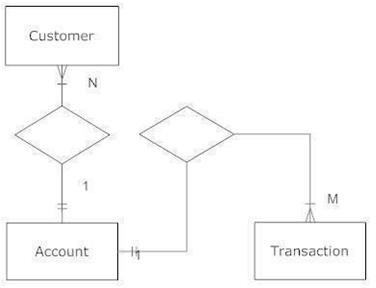
A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.

A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

**Connecting lines**, solid lines that connect attributes to show the relationships of entities in the ER Diagram.

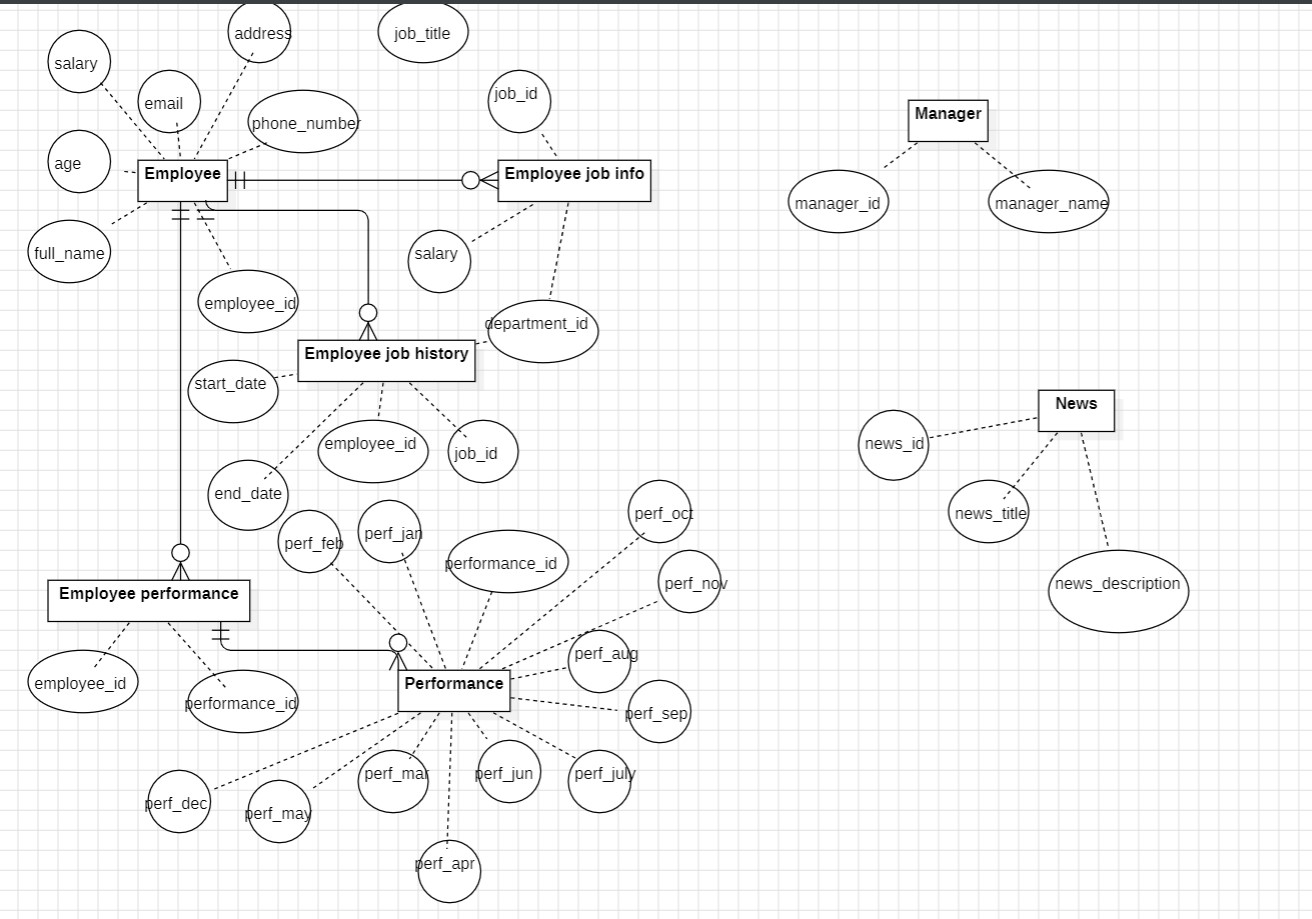
**Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinality specifies the absolute minimum number of relationships.

There are many notation styles that express cardinality.



# 

# ER DIAGRAM FOR HUMAN RESOURCE MANAGEMENT SYSTEM

****

## DATA FLOW DIAGRAM (LEVEL 0 AND LEVEL 1):

The Data Flow Diagrams (DFDs) are used for structure analysis and design. DFDs show the flow of data from external entities into the system. DFDs also show how the data moves and is transformed from one process to another, as well as its logical storage. The following symbols are used within DFDs. For clarity, a key has been provided at the bottom of this page.

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

### HISTORY:

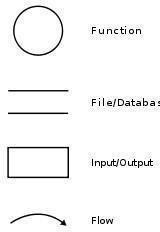
[Larry Constantine,](https://en.wikipedia.org/wiki/Larry_Constantine) the original developer of structured design, based on Martin and Estrin's "Data Flow Graph" model of computation.

Starting in the 1970s, data flow diagrams (DFD) became a popular way to visualize the major steps and data involved in software system processes. DFDs were usually used to show data flow in a computer system, although they could in theory be applied to [business process modelling. D](https://en.wikipedia.org/wiki/Business_process_modeling)FD were useful to document the major data flows or to explore a new high-level design in terms of data flow.

## THEORY:

****

**DATA FLOW DIAGRAM EXAMPLE:**

****

The Data Flow Diagrams (DFDs) are used for structure analysis and design. DFDs show the flow of data from external entities into the system. DFDs also show how the data moves and is transformed from one process to another, as well as its logical storage. The following symbols are used within DFDs. For clarity, a key has been provided at the bottom of this page.

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

Data flow diagrams are one of the three essential perspectives of the structured-systems analysis and design method. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented.

The old system's dataflow diagrams can be drawn up and compared with the new system's data flow diagrams to draw comparisons to implement a more efficient system. Data flow diagrams can beused to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data flow diagram model.

In the course of developing a set of levelled data flow diagrams the analyst/designer is forced to address how the system may be decomposed into component sub-systems, and to ide[ntify the transaction data](https://en.wikipedia.org/wiki/Transaction_data) [in t](https://en.wikipedia.org/wiki/Transaction_data)he [data model.](https://en.wikipedia.org/wiki/Data_model)

Data flow diagrams can be used in both Analysis and Design phase of the [SDLC.](https://en.wikipedia.org/wiki/Systems_development_life_cycle)

There are different notations to draw data flow diagrams (Yourdon & Coad and [Gane](https://en.wikipedia.org/wiki/Chris_Gane_(computer_scientist)) [&](https://en.wikipedia.org/wiki/Chris_Gane_(computer_scientist)) [Sarson),](https://en.wikipedia.org/wiki/Trish_Sarson) defining different visual representations for processes, data stores, data flow, and external entities.

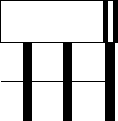
## PHYSICAL VS LOGICAL DFD:

A logical DFD captures the data flows that are necessary for a system to operate. It describes the processes that are undertaken, the data required and produced by each process, and the stores needed to hold the data. On the other hand, a physical DFD shows how the system is actually implemented, either at the moment (Current Physical DFD), or how the designer intends it to be in the future (Required Physical DFD).

Thus, a Physical DFD may be used to describe the set of data items that appear on each piece of paper that move around an office, and the fact that a particular set of pieces of paper are stored together in a filing cabinet. It is quite possible that a Physical DFD will include references to data that are duplicated, or redundant, and that the data stores, if implemented as a set of [database tables, w](https://en.wikipedia.org/wiki/Database_table)ould constitute an un-normalised (or denormalised) relational database. In contrast, a Logical DFD attempts to capture the data flow aspects of a system in a form that has neither redundancy nor duplication.

## 

## DATA FLOW SYMBOLS AND THERE MEANINGS:

****

An entity. A source of data or a destination for data.

**Source/Sink:** Represented by rectangles in the diagram. Sources and Sinks are external entities which are sources or destinations of data, respectively.



**Process:** Represented by circles in the diagram. Processes are responsible for manipulating the data. They take data as input and output an altered version of the data.



**Data Store:** Represented by a segmented rectangle with an open end on the right. Data Stores are both electronic and physical locations of data. Examples include databases, directories, files, and even filing cabinets and stacks of paper.



In our course, we need to understand and be able to draw 2 types of Data Flow Diagrams, they are Level-0 and Level 1 DFD’s. In this blog, I will hopefully make it easier to understand the differences between the two types of DFD’s and help understand how to draw a DFD for the exam.

Firstly we will look at level-0 DFD’s and give an example. Then we will look at Level 1 DFD’s and

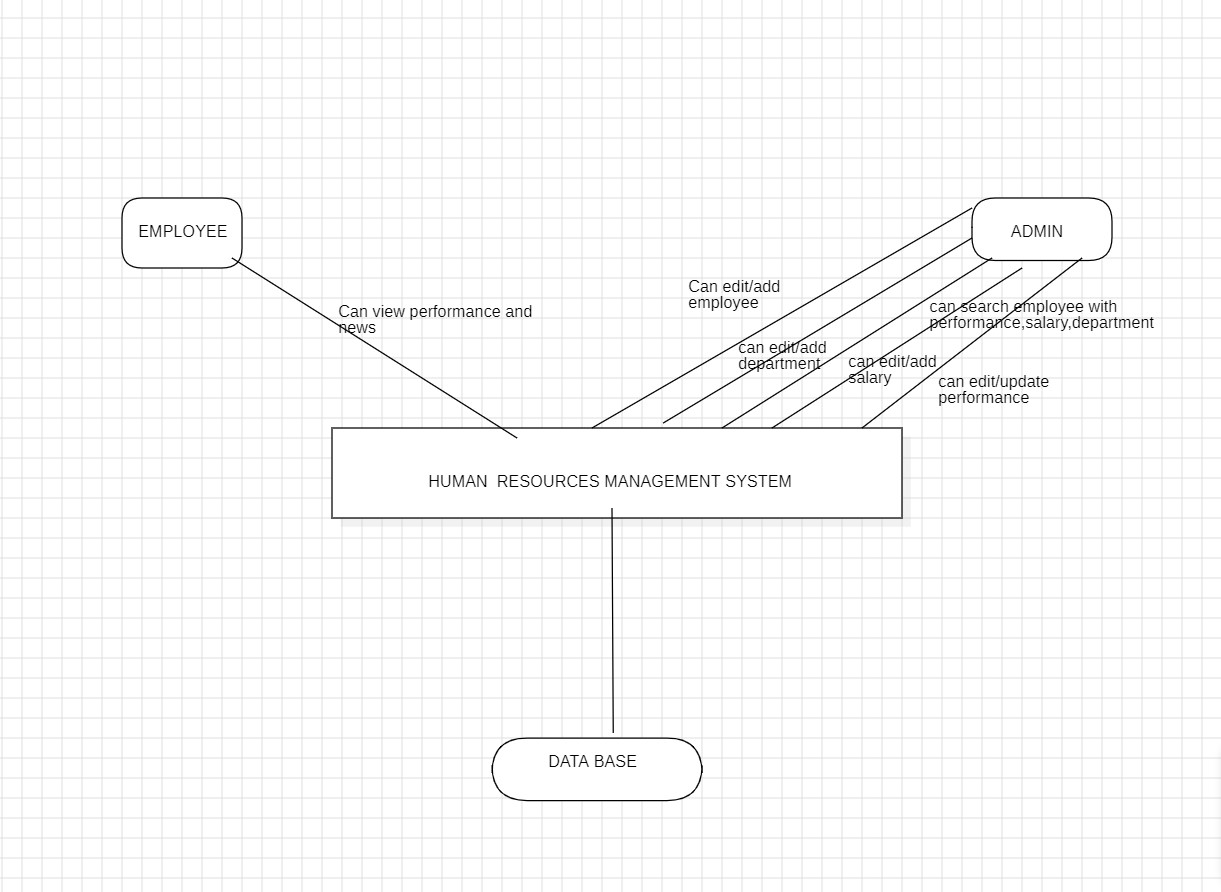
give an example.

A level-0 DFD is the most basic form of DFD. It aims to show how the entire system works at a glance. There is only one process in the system and all the data flows either into or out of this process.Level- 0 DFD’s demonstrates the interactions between the process and external entities. They do not contain Data Stores.

When drawing Level-0 DFD’s, we must first identify the process, all the external entities and all the data flows. We must also state any assumptions we make about the system. It is advised that we draw the process in the middle of the page. We then draw our external entities in the corners and finally connect our entities to our process with the data flows.

## 

## LEVEL ZERO DIAGRAM FOR HUMAN RESOURCE MANAGEMENT SYSTEM:

****

**LEVEL 1 DFD’s:**

Level 1 DFD’s aim to give an overview of the full system. They look at the system in more detail. Major processes are broken down into subprocesses. Level 1 DFD’s also identifies data stores that are used by the major processes. When constructing a Level 1 DFD, we must start by examining the Context Level DFD.

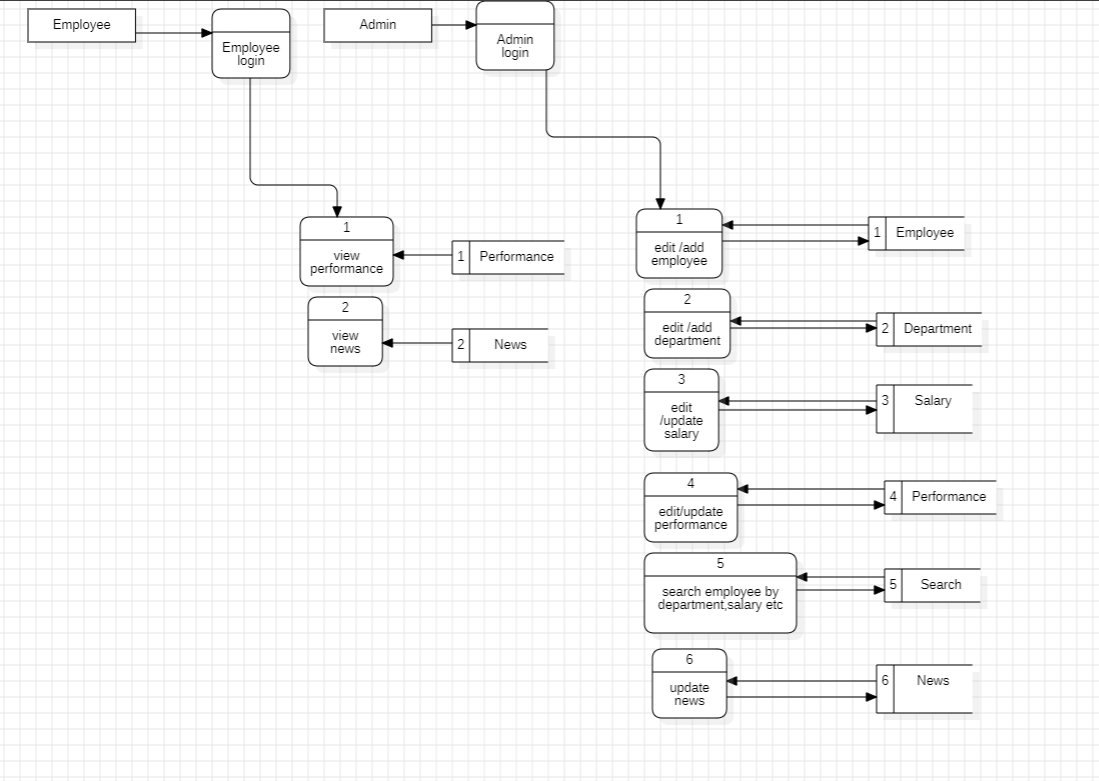
We must break up the single process into its subprocesses.

We must then pick out the data stores from the text we are given and include them in our DFD.

Like the Context Level DFD’s, all entities, data stores and processes must be labelled.

We must also state any assumptions made from the text.

This page has the Level 1 DFD of HUMAN RESOURCE MANAGEMENT SYSTEM



## 

# 3.3 GANTT CHART:

A Gantt chart is a type of bar chart, devised by Henry Gantt in th10s, that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project.

Modern Gantt charts also show the dependency (i.e., precedence network) relationships between activities. Gantt charts can be used to show current schedule status using percent-complete shadings and a vertical "TODAY" line as shown here.

## Although now regarded as a common charting technique, Gantt charts were considered revolutionary when first introduced. This chart is also used in information technology to represent data that has been collected.

## HISTORICAL DEVELOPMENT:

The first known tool of this type was developed in 1896 by Karol Adamiecki, who called it a harmonogram , Adamiecki did not publish his chart until 1931, however, and only in Polish, which limited both its adoption and recognition of his authorship.

The chart is named after Henry Gantt (1861– 1919), who designed his chart around the years 1910– 1915.

One of the first major applications of Gantt charts was by the United States during World War I, at the instigation of General William Crozier

In the 1980s, personal computers allowed widespread creation of complex and elaborate Gantt charts. The first desktop applications were intended mainly for project managers and project schedulers.

With the advent of the Internet and increased collaboration over networks at the end of the 1990s, Gantt charts became a common feature of web-based applications, including collaborative group

## GANTT CHART BENEFITS:

**CLARITY:** One of the biggest benefits of a [Gantt chart is](http://www.brighthubpm.com/templates-forms/3418-using-excel-to-create-a-gantt-chart/) the tool's ability to boil down multiple tasks and timelines into a single document. Stakeholders throughout an organization can easily understand where teams are in a process while grasping the ways in which independent elements come together toward project completion.

**COMMUNICATION:** Teams can use Gantt charts to replace meetings and enhance other status updates. Simply clarifying chart positions offers an easy, visual method to help team members understand task progress.

**MOTIVATION:** Some teams or team members become more effective when faced with a form of external motivation. Gantt charts offer teams the ability to focus work at the front of a task timeline, or at the tail end of a chart segment. Both types of team members can find Gantt charts meaningful as they plug their own work habits into the overall project schedule.

**CO ORDINATION:** For project managers and resource schedulers, the benefits of a Gantt chart include the ability to sequence events and reduce the potential for overburdening team members. Some project managers even use combinations of charts to break down projects into more manageable sets of tasks.

**CREATIVITY:** Sometimes, a lack of time or resources forces project managers and teams to find creative solutions. Seeing how individual tasks intertwine on Gantt charts often encourages new partnerships and collaborations that might not have evolved under traditional task assignment systems.

**TIME MANAGEMENT**: Most managers regard scheduling as one of the major benefits of Gantt charts in a creative environment. Helping teams understand the overall impact of project delays can foster stronger collaboration while encouraging better task organization.

**FLEXIBILITY:** Whether you [use Excel to generate Gantt charts or](http://www.brighthubpm.com/templates-forms/31029-a-microsoft-excel-project-planning-form/) you load tasks into a more precise chart generator, the ability to issue new charts as your project evolves lets you react to unexpected changes in [project scope or](http://www.brighthubpm.com/project-planning/10057-defining-the-project-scope-tips-amp-free-template/) timeline. While revising your project schedule too frequently can eliminate some of the other benefits of Gantt charts, offering a realistic view of a project can help team members recover from setbacks or adjust to other changes.

**MANAGEABILITY:** For project managers handling complex assignments, like software publishing or event planning, the benefits of Gantt charts include externalizing assignments. By visualizing all of the pieces of a project puzzle, managers can make more focused, effective decisions about resources and timetables.

**EFFICIENCY:** Another one of the benefits of Gantt charts is the ability for teams members to leverage each other’s deadlines for maximum efficiency. For instance, while one team member waits on the outcome of three other tasks before starting a crucial piece of the assignment, he or she can perform other project tasks. Visualizing resource usage during projects allows managers to make better use of people, places, and things.

## ACCOUNTABILITY: When project teams face major organizational change, documenting effort and outcomes becomes crucial to career success. Using Gantt charts during critical projects allows both project managers and participants to track team progress, highlighting both big wins and major failures. During professional review periods, team members who frequently exceed expectations can leverage this documentation into larger raises or bonuses.

## GANTT CHART IMPORTANCE:

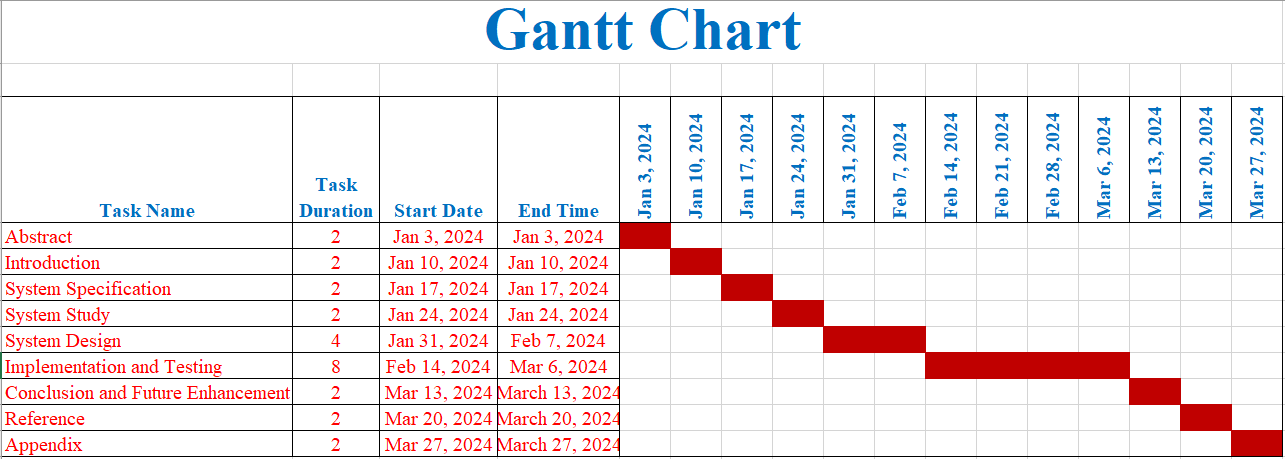
The project's summary and terminal elements, which combine to form the project's internal structure, are shown on the Gantt chart. Many charts will also depict the precedence rankings and dependencies of various tasks within the project. The charts can illustrate the start and finish project terminal elements in project management.

It can also show summary elements and terminal dependencies. The smallest task tracked as part of the project effort is known as a terminal element. Gantt chart represents the tasks in most modern project scheduling packages. However other management applications use simpler communication tools such as message boards, to-do lists and simple scheduling etc., therefore, they do not use Gantt charts as heavily.

The way to create this chart begins by determining and listing the necessary activities. Next, sketch out how you expect the chart to look. List which items depend on others and what activities take place when. For each activity, list how many man-hours it will require, and who is responsible.

Lastly, determine the throughput time. This technique's primary advantage is its good graphical overview that is easy to understand for nearly all project participants and stakeholders. Its primary disadvantage is its limited applicability for many projects, since projects are often more complex than can be effectively communicated with this chart.

Here is the Gantt chart for HUMAN RESOURCE MANAGEMENT SYSTEM



# INPUT/OUTPUT DESIGN:

## LOGIN PAGE:

Imports MySql.Data.MySqlClient

Imports System.Security.Cryptography

Imports System.Text

Public Class Voter\_Login

Private voterId As Integer

Public Function HashPassword(password As String) As String

Dim sha256 As SHA256 = SHA256.Create()

Dim bytes As Byte() = Encoding.UTF8.GetBytes(password)

Dim hash Imports System.Data.SqlClient

Public Class Frm\_Login

Dim emp\_id As Integer = 0

Private Sub formClear()

txtUserID.Clear()

txtPass.Clear()

End Sub

Private Sub btnCancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCancel.Click

Me.Hide()

Frm\_Main.Show()

End Sub

Private Sub cmboxLoginType\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmboxLoginType.SelectedIndexChanged

If cmboxLoginType.Text.Equals("Employee") Then

lblName.Text = "Employee ID"

lblPass.Visible = False

txtPass.Visible = False

lblLogin.Location = New Point(49, 130)

cmboxLoginType.Location = New Point(157, 129)

ElseIf cmboxLoginType.Text.Equals("Admin") Then

lblName.Text = "Admin Name"

lblLogin.Location = New Point(49, 166)

cmboxLoginType.Location = New Point(159, 166)

lblPass.Visible = True

txtPass.Visible = True

End If

End Sub

Public Function verifyEmployeeId() As Integer

Dim query = "Select employee\_id From Employee Where employee\_id = '" & Convert.ToInt32(txtUserID.Text) & "'"

Dim dr As SqlDataReader = getDataReader(query)

dr.Read()

emp\_id = dr("employee\_id")

dr.Close()

Return emp\_id

End Function

Private Sub btnLogin\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLogin.Click

Try

If cmboxLoginType.Text.Equals("Employee") AndAlso Convert.ToInt32(txtUserID.Text) = verifyEmployeeId() Then

disableRights(Frm\_Main.btnAdmin, Frm\_Main.btnDeptt, Frm\_Main.btnSearch, Frm\_Main.btnEmployee, Frm\_Main.MenuStrip)

Frm\_Employee\_Performance.emp\_id = Convert.ToInt32(txtUserID.Text)

Frm\_Main.flag = 0

Frm\_Main.flag3 = 0

Frm\_Main.Show()

formClear()

ElseIf cmboxLoginType.Text.Equals("Admin") AndAlso txtUserID.Text.Equals("admin") AndAlso txtPass.Text.Equals("admin") Then

enableRights(Frm\_Main.btnAdmin, Frm\_Main.btnDeptt, Frm\_Main.btnSearch, Frm\_Main.btnEmployee, Frm\_Main.MenuStrip)

Frm\_Main.Show()

Frm\_Main.flag = 1

Frm\_Main.flag3 = 1

Frm\_Main.btnLogout.Visible = True

Frm\_Main.btnAdmin.Enabled = False

formClear()

Else

MsgBox("Must Enter A Valid Login Details")

End If

Catch ex As Exception

If cmboxLoginType.Text.Equals("Employee") Then

MsgBox("Must Enter A Valid Employee ID")

End If

End Try

End Sub

Private Sub txtPass\_KeyPress(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtPass.KeyPress

If (Asc(e.KeyChar) = 13) Then

btnLogin\_Click(sender, e)

End If

End Sub

Private Sub txtUserID\_KeyPress(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtUserID.KeyPress

If (Asc(e.KeyChar) = 13) Then

btnLogin\_Click(sender, e)

End If

End Sub

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

End Sub

End ClassAs Byte() = sha256.ComputeHash(bytes)

Dim stringBuilder As StringBuilder = New StringBuilder()

For i As Integer = 0 To hash.Length - 1

stringBuilder.Append(hash(i).ToString("x2"))

Next

Return stringBuilder.ToString()

End Function

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

Dim connStr As String = "server=127.0.0.1;user=root;password=MySQL@6947;database=project"

Dim voter\_id As String = TextBox1.Text

voterId = voter\_id

Dim hashed\_pass As String = HashPassword(TextBox2.Text)

Dim password As String = ""

Try

Using conn As New MySqlConnection(connStr)

conn.Open()

Dim sql As String = $"select password from voters where voter\_id={voter\_id}"

Using cmd As New MySqlCommand(sql, conn)

Using reader As MySqlDataReader = cmd.ExecuteReader()

While reader.Read()

password = reader.GetString("password")

End While

End Using

End Using

End Using

Catch ex As Exception

MessageBox.Show("Error retrieving data: " & ex.Message)

End Try

If password = hashed\_pass Then

MessageBox.Show("Login Successful")

Dim voterPage As New Voter\_page(voterId)

voterPage.Show()

Else

MessageBox.Show("Login Unsuccessful")

End If

TextBox1.Text = ""

TextBox2.Text = ""

End Sub

End Class

## ADMIN LOGIN PAGE:

Public Class Frm\_Admin\_Login

Private Sub btnOk\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOk.Click

If txtUserName.Text.Equals("Admin") AndAlso txtPassword.Text.Equals("Admin") Then

Frm\_Main.btnLogout.Visible = True

Frm\_Main.flag2 = 1

Me.Close()

Else

MsgBox("Must Enter A Valid Login Details")

End If

End Sub

Private Sub btnCancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCancel.Click

Me.Close()

End Sub

Private Sub txtPassword\_KeyPress(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtPassword.KeyPress

If (Asc(e.KeyChar) = 13) Then

btnOk\_Click(sender, e)

End If

End Sub

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

End Sub

End Class

## MAIN FORM:

## Public Class Frm\_Main

## Public Shared flag = 0

## Public Shared flag2 = 0

## Public Shared flag3 = 0

## Private Sub Frm\_Main\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

## btnLogout.Visible = False

## Panel4.Visible = False

## Timer1.Start()

## End Sub

## Private Sub btnEmployee\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEmployee.Click

## Frm\_Employee.Show()

## End Sub

## Private Sub btnDeptt\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDeptt.Click

## Frm\_Department.Show()

## End Sub

## Private Sub btnPerformance\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPerformance.Click

## If (flag = 0) Then

## Frm\_Employee\_Performance.Show()

## ElseIf (flag = 1) Then

## Frm\_PerformanceAll.Show()

## End If

## End Sub

## Private Sub btnLogout\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLogout.Click

## If (flag2 = 0) Then

## Me.Close()

## ElseIf (flag2 = 1) Then

## Me.Close()

## Frm\_Login.Show()

## End If

## End Sub

## Private Sub NewToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

## Frm\_Department.Show()

## End Sub

## Private Sub AddNewToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

## Frm\_Employee.Show()

## End Sub

## Private Sub btnAdmin\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAdmin.Click

## Frm\_Admin\_Login.Show()

## End Sub

## Private Sub btnSearch\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSearch.Click

## Frm\_Search.Show()

## End Sub

## Private Sub btnNews\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNews.Click

## If (flag3 = 0) Then

## loadNews()

## Panel4.Visible = True

## ElseIf (flag3 = 1) Then

## Frm\_News.Show()

## End If

## End Sub

## Private Sub AllEmployeePerformanceToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

## Frm\_PerformanceAll.Show()

## End Sub

## Private Sub AddNewsToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

## Frm\_News.Show()

## End Sub

## Private Sub loadNews()

## Dim news As String = String.Empty

## Dim query = "Select news\_title, news\_description From News"

## Dim dr As SqlClient.SqlDataReader = getDataReader(query)

## While dr.Read AndAlso dr.HasRows

## news &= "-> " & dr("news\_description").ToString()

## End While

## Label1.Text = news

## Label1.BackColor = Color.SkyBlue

## Label1.AutoSize = True

## Label1.Left = Panel1.Width

## Timer1.Interval = 25

## End Sub

## Private Sub Panel1\_Paint(sender As Object, e As PaintEventArgs) Handles Panel1.Paint

## End Sub

## Private Sub Panel4\_Click(sender As Object, e As EventArgs) Handles Panel4.Click

## End Sub

## Private Sub Label1\_Click(sender As Object, e As EventArgs) Handles Label1.Click

## End Sub

## Private Sub Panel4\_Paint(sender As Object, e As PaintEventArgs) Handles Panel4.Paint

## End Sub

## End Class

## EMPLOYEE FORM:

## Public Class Frm\_Employee

## Dim emp\_id

## Function emptyFields()

## If txtName.Text = "" OrElse txtAge.Text = "" OrElse txtEmail.Text = "" OrElse txtAddress.Text = "" OrElse txtSalary.Text = "" Then

## Return True

## End If

## Return False

## End Function

## Sub formClear()

## txtName.Clear()

## txtAge.Clear()

## txtEmail.Clear()

## txtPhone.Clear()

## txtAddress.Clear()

## txtSalary.Clear()

## End Sub

## Private Sub add()

## If emptyFields() = True Then

## MsgBox("Must Enter In All Fields ", MsgBoxStyle.Information)

## Return

## Else

## dgvEmployee.Rows.Add()

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpID").Value = txtID.Text

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpName").Value = txtName.Text

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpAge").Value = txtAge.Text

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpEmail").Value = txtEmail.Text

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpPhone").Value = txtPhone.Text

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpAddress").Value = txtAddress.Text

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpSalary").Value = txtSalary.Text

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpDepttID").Value = cmboxDeptt.SelectedValue

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpDeptt").Value = cmboxDeptt.Text

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).DefaultCellStyle.ForeColor = Color.Blue

## End If

## formClear()

## End Sub

## Private Sub edit()

## If dgvEmployee.SelectedRows.Count = 0 Then

## Return

## End If

## If btnEdit.Text = "Edit" Then

## txtID.Text = dgvEmployee.SelectedRows(0).Cells("EmpID").Value

## txtName.Text = dgvEmployee.SelectedRows(0).Cells("EmpName").Value

## txtAge.Text = dgvEmployee.SelectedRows(0).Cells("EmpAge").Value

## txtEmail.Text = dgvEmployee.SelectedRows(0).Cells("EmpEmail").Value

## txtPhone.Text = dgvEmployee.SelectedRows(0).Cells("EmpPhone").Value

## txtAddress.Text = dgvEmployee.SelectedRows(0).Cells("EmpAddress").Value

## txtSalary.Text = dgvEmployee.SelectedRows(0).Cells("EmpSalary").Value

## cmboxDeptt.SelectedValue = dgvEmployee.SelectedRows(0).Cells("EmpDepttID").Value

## cmboxDeptt.Text = dgvEmployee.SelectedRows(0).Cells("EmpDeptt").Value

## dgvEmployee.Enabled = False

## btnEdit.Text = "Update"

## Else

## dgvEmployee.SelectedRows(0).Cells("EmpID").Value = txtID.Text

## dgvEmployee.SelectedRows(0).Cells("EmpName").Value = txtName.Text

## dgvEmployee.SelectedRows(0).Cells("EmpAge").Value = txtAge.Text

## dgvEmployee.SelectedRows(0).Cells("EmpEmail").Value = txtEmail.Text

## dgvEmployee.SelectedRows(0).Cells("EmpPhone").Value = txtPhone.Text

## dgvEmployee.SelectedRows(0).Cells("EmpAddress").Value = txtAddress.Text

## dgvEmployee.SelectedRows(0).Cells("EmpSalary").Value = txtSalary.Text

## dgvEmployee.SelectedRows(0).Cells("EmpDepttID").Value = cmboxDeptt.SelectedValue

## dgvEmployee.SelectedRows(0).Cells("EmpDeptt").Value = cmboxDeptt.Text

## dgvEmployee.Enabled = True

## btnEdit.Text = "Edit"

## End If

## dgvEmployee.SelectedRows(0).DefaultCellStyle.ForeColor = Color.Blue

## End Sub

## Private Sub remove()

## If dgvEmployee.SelectedRows.Count = 0 Then

## Return

## End If

## dgvEmployee.SelectedRows(0).DefaultCellStyle.ForeColor = Color.Red

## End Sub

## Private Sub apply()

## Dim query As String = String.Empty

## For Each row As DataGridViewRow In dgvEmployee.Rows

## If row.Cells("FromDB").Value = True Then

## If row.DefaultCellStyle.ForeColor = Color.Black Then

## Continue For

## ElseIf row.DefaultCellStyle.ForeColor = Color.Blue Then

## query = "Update Employee set full\_name = '" & row.Cells("EmpName").Value & "',age = '" & row.Cells("EmpAge").Value & "',email = '" & row.Cells("EmpEmail").Value & "',phone\_number = '" & row.Cells("EmpPhone").Value & "',address = '" & row.Cells("EmpAddress").Value & "',salary = '" & row.Cells("EmpSalary").Value & "', department\_id = '" & row.Cells("EmpDepttID").Value & "' Where employee\_id = '" & row.Cells("EmpID").Value & "'"

## ElseIf row.DefaultCellStyle.ForeColor = Color.Red Then

## query = "Delete From Employee Where employee\_id = " & row.Cells("EmpID").Value

## End If

## Else

## If row.DefaultCellStyle.ForeColor = Color.Blue Then

## query = "Insert Into Employee (full\_name,age,email,phone\_number,address,salary,department\_id) Values ('" & row.Cells("EmpName").Value & "','" & row.Cells("EmpAge").Value & "','" & row.Cells("EmpEmail").Value & "','" & row.Cells("EmpPhone").Value & "','" & row.Cells("EmpAddress").Value & "','" & row.Cells("EmpSalary").Value & "','" & row.Cells("EmpDepttID").Value & "')"

## ElseIf row.DefaultCellStyle.ForeColor = Color.Red Then

## dgvEmployee.Rows.Remove(row)

## Continue For

## End If

## End If

## executeQuery(query)

## Next

## reload()

## End Sub

## Private Sub reload()

## Dim query As String

## query = "Select E.\*, D.department\_name, D.department\_id"

## query = query + " From Employee E "

## query = query + " Inner Join Department D On E.department\_id = D.department\_id"

## dgvEmployee.Rows.Clear()

## Dim dt As DataTable = getDataTable(query)

## For Each row As DataRow In dt.Rows

## dgvEmployee.Rows.Add()

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpID").Value = row("employee\_id")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpName").Value = row("full\_name")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpAge").Value = row("age")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpEmail").Value = row("email")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpPhone").Value = row("phone\_number")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpAddress").Value = row("address")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpSalary").Value = row("salary")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpDepttID").Value = row("department\_id")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpDeptt").Value = row("department\_name")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("FromDB").Value = True

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).DefaultCellStyle.ForeColor = Color.Black

## Next

## txtID.Text = autoId().ToString()

## End Sub

## Private Sub fillCmboxDepartment() ' for getting deparment names from database

## Dim query As String = "Select department\_id, department\_name From Department order by department\_name"

## Dim dtDeptt As DataTable = getDataTable(query)

## cmboxDeptt.DataSource = dtDeptt

## cmboxDeptt.DisplayMember = "department\_name"

## cmboxDeptt.ValueMember = "department\_id"

## If cmboxDeptt.Items.Count > 0 Then

## cmboxDeptt.SelectedIndex = 0

## End If

## End Sub

## Private Function autoId() As Integer ' generating auto employee id for a new employee

## Dim query = "Select IsNull(Max(employee\_id+1),0) employee\_id From Employee"

## Dim dr As SqlClient.SqlDataReader

## dr = getDataReader(query)

## dr.Read()

## emp\_id = dr("employee\_id")

## dr.Close()

## Return emp\_id

## End Function

## Private Sub Frm\_Employee\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

## Try

## fillCmboxDepartment() ' loading all department names from database to combobox department

## loadAllEmployees() ' loading all employee names from database to combobox employee for searching purpose

## loadAllJobTitles() ' loading all job titles from database to combobox job titles

## reload()

## txtID.Text = autoId().ToString()

## dateTimeEnd.MinDate = Now

## dateTimeJoin.MinDate = Now

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub loadAllEmployees()

## Dim query As String = "Select employee\_id, full\_name From Employee order by full\_name"

## Dim dtEmp As DataTable = getDataTable(query)

## cmboxNames.DataSource = dtEmp

## cmboxNames.DisplayMember = "full\_name"

## cmboxNames.ValueMember = "employee\_id"

## If cmboxNames.Items.Count > 0 Then

## cmboxNames.SelectedIndex = 0

## End If

## End Sub

## Private Sub lstboxNames\_DoubleClick(ByVal sender As System.Object, ByVal e As System.EventArgs)

## Dim name = lstboxNames.SelectedItem.ToString()

## Dim query As IEnumerable(Of DataGridViewRow) = \_

## From row As DataGridViewRow In dgvEmployee.Rows \_

## Where row.Cells("EmpName").Value.ToString().Contains(name) \_

## Select row

## For Each row As DataGridViewRow In query

## row.Selected = True

## Next

## End Sub

## Private Sub loadAllJobTitles()

## Dim query As String = "Select job\_id, job\_title From Employee\_Job\_Info order by job\_title"

## Dim dtEmp As DataTable = getDataTable(query)

## cmboxTitle.DataSource = dtEmp

## cmboxTitle.DisplayMember = "job\_title"

## cmboxTitle.ValueMember = "job\_id"

## If cmboxTitle.Items.Count > 0 Then

## cmboxTitle.SelectedIndex = 0

## End If

## End Sub

## Private Sub btnAddJob\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAddJob.Click

## Try

## add()

## apply()

## Dim query = "Insert Into Employee\_Job\_History(employee\_id,start\_date,end\_date,job\_id,department\_id) Values ('" & Convert.ToInt32(emp\_id - 1) & "','" & dateTimeJoin.Value & "','" & dateTimeEnd.Value & "','" & cmboxTitle.SelectedValue & "','" & cmboxDeptt.SelectedValue & "')"

## executeQuery(query)

## MsgBox("Employee And Job Added Successfully")

## TabControl1.TabPages("Employee Details").Show()

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub btnAdd\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAdd.Click

## Try

## add()

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub btnEdit\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEdit.Click

## Try

## edit()

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub btnRemove\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRemove.Click

## Try

## remove()

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub btnApply\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnApply.Click

## Try

## apply()

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub btnOk\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOk.Click

## Try

## apply()

## Me.Close()

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub btnCancel\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCancel.Click

## Me.Close()

## End Sub

## Private Sub btnSubmit\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSubmit.Click

## lstboxNames.Items.Add(cmboxNames.Text)

## End Sub

## Private Sub cmboxNames\_SelectedIndexChanged\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmboxNames.SelectedIndexChanged

## Dim name = cmboxNames.Text

## Dim query As IEnumerable(Of DataGridViewRow) = \_

## From row As DataGridViewRow In dgvEmployee.Rows \_

## Where row.Cells("EmpName").Value.ToString().Contains(name) \_

## Select row

## For Each row As DataGridViewRow In query

## row.Selected = True

## Next

## End Sub

## Private Sub txtName\_KeyPress(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtName.KeyPress

## Dim ch As Char = e.KeyChar

## If Char.IsDigit(ch) Then 'Ristricting TxtName To Input Only Characters

## e.Handled = True

## End If

## End Sub

## Private Sub txtAge\_KeyPress(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtAge.KeyPress

## Dim ch As Char = e.KeyChar

## If Char.IsLetter(ch) Then 'Ristricting age To Input Only Digits(any number)

## e.Handled = True

## End If

## End Sub

## Private Sub txtPhone\_KeyPress(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtPhone.KeyPress

## Dim ch As Char = e.KeyChar

## If Char.IsLetter(ch) Then 'Ristricting TxtPhone To Input Only Digits(any number)

## e.Handled = True

## End If

## End Sub

## Private Sub txtSalary\_KeyPress(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtSalary.KeyPress

## Dim ch As Char = e.KeyChar

## If Char.IsLetter(ch) Then 'Ristricting salary To Input Only Digits(any number)

## e.Handled = True

## End If

## End Sub

## Private Sub GroupBox2\_Enter(sender As Object, e As EventArgs) Handles GroupBox2.Enter

## End Sub

## Private Sub TabPage1\_Click(sender As Object, e As EventArgs) Handles TabPage1.Click

## End Sub

## End Class

## EMPLOYEE PERFORMANCE FORM:

Public Class Frm\_Employee\_Performance

Public Shared emp\_id = 0

Dim monthName As String = String.Empty

Private Function employeePerformance(ByVal employeeID As Integer, ByVal monthName As String) As DataTable

Dim query = "Select E.employee\_id , E.full\_name, P.performance\_id, P." & monthName

query = query & " From Employee\_Performance\_Table Ept"

query = query & " Inner Join Employee E on E.employee\_id = Ept.employee\_id"

query = query & " Inner Join Performance P on P.performance\_id = Ept.performance\_id"

query = query & " Where E.employee\_id ='" & employeeID & "'"

Return getDataTable(query)

End Function

Private Sub showPerfInDataGrid()

Dim dt As DataTable = employeePerformance(emp\_id, monthName)

dgvPerformance.Rows.Clear()

For Each row As DataRow In dt.Rows

dgvPerformance.Rows.Add()

dgvPerformance.Rows(dgvPerformance.RowCount - 1).Cells("PerfID").Value = row("performance\_id")

dgvPerformance.Rows(dgvPerformance.RowCount - 1).Cells("EmpID").Value = row("employee\_id")

dgvPerformance.Rows(dgvPerformance.RowCount - 1).Cells("EmpName").Value = row("full\_name")

dgvPerformance.Rows(dgvPerformance.RowCount - 1).Cells("PerfMonth").Value = row(monthName)

Next

End Sub

Private Sub cmboxMonths2\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmboxMonths2.SelectedIndexChanged

If cmboxMonths2.Text.Equals("January") Then

monthName = "perf\_january"

ElseIf cmboxMonths2.Text.Equals("Feburary") Then

monthName = "perf\_feburary"

ElseIf cmboxMonths2.Text.Equals("March") Then

monthName = "perf\_march"

ElseIf cmboxMonths2.Text.Equals("April") Then

monthName = "perf\_april"

ElseIf cmboxMonths2.Text.Equals("May") Then

monthName = "perf\_may"

ElseIf cmboxMonths2.Text.Equals("June") Then

monthName = "perf\_june"

ElseIf cmboxMonths2.Text.Equals("July") Then

monthName = "perf\_july"

ElseIf cmboxMonths2.Text.Equals("August") Then

monthName = "perf\_august"

ElseIf cmboxMonths2.Text.Equals("September") Then

monthName = "perf\_september"

ElseIf cmboxMonths2.Text.Equals("October") Then

monthName = "perf\_october"

ElseIf cmboxMonths2.Text.Equals("November") Then

monthName = "perf\_november"

ElseIf cmboxMonths2.Text.Equals("December") Then

monthName = "perf\_december"

End If

End Sub

Private Sub btnOk\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOk.Click

Me.Close()

End Sub

Private Sub btnSubmit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSubmit.Click

showPerfInDataGrid()

End Sub

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

End Sub

End Class

## PERFORMANCE ALL FORM:

## Public Class Frm\_PerformanceAll

## Dim flag As Integer = 0

## Dim monthName As String = String.Empty

## Private Function monthPerformance(ByRef monthName As String) As DataTable

## Dim query = "Select E.employee\_id , E.full\_name , " & monthName

## query = query & " From Employee\_Performance\_Table Ept"

## query = query & " Inner Join Employee E On E.employee\_id = Ept.employee\_id"

## query = query & " Inner Join Performance P On P.performance\_id = Ept.performance\_id"

## Return getDataTable(query)

## End Function

## Private Sub cmboxMonths\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmboxMonths.SelectedIndexChanged

## If cmboxMonths.Text.Equals("January") Then

## dgvPerformance.DataSource = monthPerformance("perf\_january")

## ElseIf cmboxMonths.Text.Equals("Feburary") Then

## dgvPerformance.DataSource = monthPerformance("perf\_feburary")

## ElseIf cmboxMonths.Text.Equals("March") Then

## dgvPerformance.DataSource = monthPerformance("perf\_march")

## ElseIf cmboxMonths.Text.Equals("April") Then

## dgvPerformance.DataSource = monthPerformance("perf\_april")

## ElseIf cmboxMonths.Text.Equals("May") Then

## dgvPerformance.DataSource = monthPerformance("perf\_may")

## ElseIf cmboxMonths.Text.Equals("June") Then

## dgvPerformance.DataSource = monthPerformance("perf\_june")

## ElseIf cmboxMonths.Text.Equals("July") Then

## dgvPerformance.DataSource = monthPerformance("perf\_july")

## ElseIf cmboxMonths.Text.Equals("August") Then

## dgvPerformance.DataSource = monthPerformance("perf\_august")

## ElseIf cmboxMonths.Text.Equals("September") Then

## dgvPerformance.DataSource = monthPerformance("perf\_september")

## ElseIf cmboxMonths.Text.Equals("October") Then

## dgvPerformance.DataSource = monthPerformance("perf\_october")

## ElseIf cmboxMonths.Text.Equals("November") Then

## dgvPerformance.DataSource = monthPerformance("perf\_november")

## ElseIf cmboxMonths.Text.Equals("December") Then

## dgvPerformance.DataSource = monthPerformance("perf\_december")

## End If

## End Sub

## Private Sub Frm\_PerformanceAll\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

## Me.Size = New Size(605, 582)

## Me.lblPerf.Location = New Point(123, 6)

## End Sub

## Private Sub btnUpdate\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnUpdate.Click

## If flag = 0 Then

## Me.btnUpdate.Text = "<< Update Performance"

## Me.Size = New Size(1068, 582)

## Me.lblPerf.Location = New Point(382, 6)

## flag = 1

## ElseIf flag = 1 Then

## Me.btnUpdate.Text = "Update Performance >>"

## Me.Size = New Size(605, 582)

## Me.lblPerf.Location = New Point(123, 6)

## flag = 0

## End If

## End Sub

## Private Function employeePerformance(ByVal employeeID As Integer, ByVal monthName As String) As DataTable

## Dim query = "Select E.employee\_id , E.full\_name, P.performance\_id, P." & monthName

## query = query & " From Employee\_Performance\_Table Ept"

## query = query & " Inner Join Employee E on E.employee\_id = Ept.employee\_id"

## query = query & " Inner Join Performance P on P.performance\_id = Ept.performance\_id"

## query = query & " Where E.employee\_id ='" & employeeID & "'"

## Return getDataTable(query)

## End Function

## Private Sub cmboxMonths2\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmboxMonths2.SelectedIndexChanged

## If cmboxMonths2.Text.Equals("January") Then

## monthName = "perf\_january"

## ElseIf cmboxMonths2.Text.Equals("Feburary") Then

## monthName = "perf\_feburary"

## ElseIf cmboxMonths2.Text.Equals("March") Then

## monthName = "perf\_march"

## ElseIf cmboxMonths2.Text.Equals("April") Then

## monthName = "perf\_april"

## ElseIf cmboxMonths2.Text.Equals("May") Then

## monthName = "perf\_may"

## ElseIf cmboxMonths2.Text.Equals("June") Then

## monthName = "perf\_june"

## ElseIf cmboxMonths2.Text.Equals("July") Then

## monthName = "perf\_july"

## ElseIf cmboxMonths2.Text.Equals("August") Then

## monthName = "perf\_august"

## ElseIf cmboxMonths2.Text.Equals("September") Then

## monthName = "perf\_september"

## ElseIf cmboxMonths2.Text.Equals("October") Then

## monthName = "perf\_october"

## ElseIf cmboxMonths2.Text.Equals("November") Then

## monthName = "perf\_november"

## ElseIf cmboxMonths2.Text.Equals("December") Then

## monthName = "perf\_december"

## End If

## End Sub

## Private Sub btnSubmit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSubmit.Click

## Try

## Dim dt As DataTable = employeePerformance(Convert.ToInt32(txtID.Text), monthName)

## For Each row As DataRow In dt.Rows

## dgvEmployee.Rows.Add()

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("PerfID").Value = row("performance\_id")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpID").Value = row("employee\_id")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("EmpName").Value = row("full\_name")

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("PerfMonth").Value = row(monthName)

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).Cells("FromDB").Value = True

## dgvEmployee.Rows(dgvEmployee.RowCount - 1).DefaultCellStyle.ForeColor = Color.Black

## Next

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub btnSave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click

## 

## Try

## Dim query = "Update Performance set " & monthName & " = " & dgvEmployee.SelectedRows(0).Cells("PerfMonth").Value & " Where performance\_id = " & dgvEmployee.SelectedRows(0).Cells("PerfID").Value & ""

## executeQuery(query)

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## End Class

## SEARCH FORM:

## Public Class Frm\_Search

## Private Sub fillCmboxDepartment() ' for getting deparment names from database

## Dim query As String = "Select department\_id, department\_name From Department order by department\_name"

## Dim dtDeptt As DataTable = getDataTable(query)

## cmboxDeptt.DataSource = dtDeptt

## cmboxDeptt.DisplayMember = "department\_name"

## cmboxDeptt.ValueMember = "department\_id"

## If cmboxDeptt.Items.Count > 0 Then

## cmboxDeptt.SelectedIndex = 0

## End If

## End Sub

## Private Sub fillCmBoxJobTitles()

## Dim query As String = "Select job\_id, job\_title From Employee\_Job\_Info order by job\_title"

## Dim dtEmp As DataTable = getDataTable(query)

## cmboxTitle.DataSource = dtEmp

## cmboxTitle.DisplayMember = "job\_title"

## cmboxTitle.ValueMember = "job\_id"

## If cmboxTitle.Items.Count > 0 Then

## cmboxTitle.SelectedIndex = 0

## End If

## End Sub

## Private Sub Frm\_Search\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

## Try

## Me.DoubleBuffered = True

## Me.Invalidate()

## fillCmboxDepartment()

## fillCmBoxJobTitles()

## cmboxDeptt.Text = "Select - Department"

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub cmboxDeptt\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmboxDeptt.SelectedIndexChanged

## Try

## Dim query = "Select E.\* "

## query = query & " From Employee E"

## query = query & " Inner Join Department D on E.department\_id = D.department\_id"

## query = query & " Where department\_name = '" & cmboxDeptt.Text & "'"

## dgvEmployee.DataSource = getDataTable(query)

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub cmboxTitle\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmboxTitle.SelectedIndexChanged

## Try

## Dim query = "Select E.employee\_id, E.full\_name, J.start\_date , J.end\_date, D.department\_name"

## query = query & " From Employee\_Job\_History J"

## query = query & " Inner Join Employee E on E.employee\_id = J.employee\_id "

## query = query & " Inner Join Department D on D.department\_id = J.department\_id"

## query = query & " Inner Join Employee\_Job\_Info Ji on Ji.job\_id = J.job\_id"

## query = query & " Where Ji.job\_title = '" & cmboxTitle.Text & "'"

## dgvEmployeeJob.DataSource = getDataTable(query)

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub btnSubmit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSubmit.Click

## Try

## Dim query = "Select E.employee\_id, E.full\_name, E.salary, D.department\_name"

## query = query & " From Employee E"

## query = query & " Inner Join Department D on D.department\_id = E.department\_id"

## query = query & " Where E.salary > " & Convert.ToInt32(txtStart.Text) & " And E.salary < " & Convert.ToInt32(txtEnd.Text) & ""

## dgvSalary.DataSource = getDataTable(query)

## Catch ex As Exception

## MsgBox(ex.Message, MsgBoxStyle.Critical)

## End Try

## End Sub

## Private Sub grpDeptt\_Paint(sender As Object, e As PaintEventArgs) Handles grpDeptt.Paint

## End Sub

## Private Sub grpSalary\_Paint(sender As Object, e As PaintEventArgs) Handles grpSalary.Paint

## End Sub

## End Class

# 4.SYSTEM CONFIGURATION:

## 4.1HARDWARE REQUIREMENTS:

PROCESSOR : INTEL CORE SPEED : 1.50GHZ

RAM : 4GB

## 4.2 SOFTWARE REQUIREMENTS:

FRONTEND : VB.NET (VISUALSTUDIO) 2022

BACKEND : MY SQL DOCUMENTATON : MS WORD 2019 OPERATING SYSTEM : WINDOWS 11

# 5.DETAILS OF SOFTWARE:

**5.1OVERVIEW OF FRONT-END:**

Microsoft Visual Studio 2022 is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio supports different programming languages and allows the code editor and debugger to support nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, Visual C++ and VB.NET. Support for other languages such as Python, Ruby, Node.js, and M among others is available via language services installed separately.

It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Java (and J#) were supported in the past**.** Microsoft provides a free version of Visual Studio called the Community edition that supports plugins and is available at no cost for all users. Support for programming languages isadded by using a specific VSPackage called a Language Service.

A language service defines various interfaces which the VSPackage implementation can implement to add support for various functionalities. Functionalities that can be added this way include syntax coloring, statement completion, brace matching, parameter information tooltips, member lists and error markers for background compilation.

If the interface is implemented, the functionality will be available for the language. Language services are implemented on a per-language basis. The implementations can reuse code from the parser or the compiler for the language. Language services can be implemented either in native code or managed code. For native code, either the native COM interfaces or the Babel Framework can be used. For managed code, the MPF includes wrappers for writing managed language services.

## FEATURES:

Boolean Conditions Automatic Garbage Collection Standard Library

Assembly Versioning Properties and Events

Delegates and Events Management Easy-to-use Generics

Indexers

Conditional Compilation Simple Multithreading

## ADVANTAGES:

The structure of the [Basic programming language](http://en.wikipedia.org/wiki/BASIC) [is](http://en.wikipedia.org/wiki/BASIC) very simple, particularly as to the executable code.

VB is not only a language but primarily an integrated, interactive development environment

(“[IDE“).](http://en.wikipedia.org/wiki/Integrated_development_environment)

The VB-IDE has been highly optimized to support rapid application development(“RAD”). It is particularly easy to develop [graphical user interfaces](http://en.wikipedia.org/wiki/Graphical_user_interface) [an](http://en.wikipedia.org/wiki/Graphical_user_interface)d to connect them to handler functions provided by the application.

The graphical user interface of the VB-IDE provides intuitively appealing views for the management of the program structure in the large and the various types of entities (classes, modules, procedures, forms).

VB provides a comprehensive interactive and context-sensitive online help system.

When editing program texts the “[IntelliSense”](http://en.wikipedia.org/wiki/IntelliSense) [t](http://en.wikipedia.org/wiki/IntelliSense)echnology informs you in a little popup window about the types of constructs that may be entered at the current cursor location.

VB is a component integration language which is attuned to Microsoft’s Component Object Model (“COM”).

COM components can be written in different languages and then integrated using VB.

Interfaces of COM components can be easily called remotely via Distributed COM (“DCOM”), which

makes it easy to construct distributed applications.

COM components can be embedded in / linked to your application’s user interface and also in/to

stored documents ([Object Linking and Embedding](http://en.wikipedia.org/wiki/Object_Linking_and_Embedding) “OLE”, “Compound Documents”). There is a wealth of readily available COM components for many different purposes. Visual Basic is built around the .NET environment used by all Microsoft

# OVERVIEW OF BACK-END

MySQL is a relational database management system developed by Oracle. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).

Oracle markets at least a dozen different editions of MySQL, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. The protocol layer implements the external interface to SQL Server. All operations that can be invoked on MySQL are communicated to it via a Oracle- defined format, called Tabular Data Stream (TDS).

TDS is an application layer protocol, used to transfer data between a database server and a client. Initially designed and developed by Sybase Inc. for their Sybase SQL Server relational database engine in 1984, and later by Microsoft in Microsoft SQL Server, TDS packets can be encased in other physical transport dependent protocols, including TCP/IP, named pipes, and shared memory. Consequently,

access to SQL Server is available over these protocols. In addition, the SQL Server API is also exposed over web services.

A Relational Data Base Management System (RDBMS) is software that: Enables you to implement a database with tables, columns and indexes. Guarantees the Referential Integrity between rows of various tables. Updates the indexes automatically. Interprets an SQL query and combines information from various tables. MySQL is a leading open source database management system.

It is a multiuser, multithreaded database management system. MySQL is especially popular on the web. It is one of the parts of the very popular LAMP platform. Linux, Apache, MySQL and PHP. MySQL database is available on most important OS platforms. It runs on BSD Unix, Linux, Windows or Mac. Wikipedia, YouTube, Facebook use MySQL. These sites manage millions of queries each day. MySQL comes in two versions. MySQL server system and MySQL embedded system. The MySQL server software and the client libraries are dual-licensed.

## ADVANTAGES:-

MySQL is becoming so popular because of many good reasons:

MySQL is released under an open-source license. So you have nothing to pay to use it. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form of the well-known SQL data language. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. MySQL works very quickly and works well even with large data sets. MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB). MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

## FEATURES:

Written in C and C++.

Tested with a broad range of different compilers.

Works on many different platforms. .

For portability, uses **C Make** in MySQL 5.5 and up. Previous series use GNU Automake,

Auto conf, and Libtool.

Tested with Purify (a commercial memory leakage detector) as well as with Val grind, a GPL tool.

Uses multi-layered server design with independent modules.

Designed to be fully multi-threaded using kernel threads, to easily use multiple CPUs if they are available.

Provides transactional and non transactional storage engines.

Fixed-length and variable-length string types.

**Statements and Functions** mysql> SELECT CONCAT(first\_name, ' ', last\_name) -> FROM citizen

-> WHERE income/dependents > 10000 AND age > 30;Full support for SQL GROUP BY and ORDER BY clauses. Support for and [GROUP\_CONCAT()).](https://dev.mysql.com/doc/refman/5.7/en/group-by-functions.html)

Support for LEFT OUTER JOIN and RIGHT OUTER JOIN with both standard SQL and ODBC syntax. Support for aliases on tables and columns as required by standard SQL. Some were changed (affected) to, return the number of rows matched instead by setting a flag when connecting to the server. Support for MySQL-specific [SHOW st](https://dev.mysql.com/doc/refman/5.7/en/show.html)atements that retrieve information about databases, storage engines, tables, and indexes. Support for the INFORMATION\_SCHEMA database, implemented according to standard SQL.

An [EXPLAIN](https://dev.mysql.com/doc/refman/5.7/en/explain.html) [st](https://dev.mysql.com/doc/refman/5.7/en/explain.html)atement to show how the optimizer resolves a query.

Independence of function names from table or column names. For example, ABS is a valid column name. The only restriction is that for a function call, no spaces are permitted between the function name and the “(” that follows it. You can refer to tables from different databases in the same statement.

### SECURITY:

A privilege and password system that is very flexible and secure, and that enables host-based verification.

Password security by encryption of all password traffic when you connect to a server.

### SCALABILITY AND LIMITS:

Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.

Support for up to 64 indexes per table. Each index may consist of 1 to 16 columns or parts of columns. The maximum index width for [InnoDB ta](https://dev.mysql.com/doc/refman/5.7/en/innodb-storage-engine.html)bles is either 767 bytes or 3072 bytes. The maximum index width for [My ISAM](https://dev.mysql.com/doc/refman/5.7/en/myisam-storage-engine.html) [ta](https://dev.mysql.com/doc/refman/5.7/en/myisam-storage-engine.html)bles is 1000 bytes. An index may use a prefix of a column for [CHAR,](https://dev.mysql.com/doc/refman/5.7/en/char.html) [VARCHA](https://dev.mysql.com/doc/refman/5.7/en/char.html)[R,](https://dev.mysql.com/doc/refman/5.7/en/blob.html) [BLOB,](https://dev.mysql.com/doc/refman/5.7/en/blob.html) [or](https://dev.mysql.com/doc/refman/5.7/en/blob.html) [TEXT](https://dev.mysql.com/doc/refman/5.7/en/blob.html) [co](https://dev.mysql.com/doc/refman/5.7/en/blob.html)lumn types.

### CONNECTIVITY:

Clients can connect to MySQL Server using several protocols:

Clients can connect using TCP/IP sockets on any platform.

On Unix systems, clients can connect using Unix domain socket files.

### LOCALIZATION:

The server can provide error messages to clients in many languages.

Full support for several different character sets, including latin1 (cp1252), german, big5, ujis, several Unicode character sets, and more. For example, the Scandinavian characters “å”, “ä” and “ö” are permitted in table and column names. All data is saved in the chosen character set.

Sorting and comparisons are done according to the chosen character set and collation (using latin1 and Swedish collation by default). It is possible to change this when the MySQL server is started. To see an example of very advanced sorting, look at the Czech sorting code. MySQL Server supports many different character sets that can be specified at compile time and runtime.

The server time zone can be changed dynamically, and individual clients can specify their own time zone.

### CLIENTS AND TOOLS:

MySQL includes several client and utility programs. These include both command-line programs and graphical programs.

MySQL Server has built-in support for SQL statements to check, optimize, and repair tables.

These statements are available from the command line through the my sqlcheck client. MySQL also includes myisamch, a very fast command-line utility for performing these operations on MyISAM tables.

MySQL programs can be invoked with the --help or -? option to obtain online assis

## 5.3 ABOUT THE PLATFORM

Windows is a series of [Operating Systems dev](https://techterms.com/definition/operating_system)eloped by Microsoft. Each version of Windows includes a [Graphical User Interface, wit](https://techterms.com/definition/gui)h a [desktop tha](https://techterms.com/definition/desktop)t allows users to view files and folders in [Windows. For](https://techterms.com/definition/window) the past two decades, Windows has been the most widely used operating system for personal computers PCs.

Microsoft Windows is designed for both home computing and professional purposes. Past versions of Windows home editions include Windows 3.0 (1990), Windows 3.1 (1992), Windows 95 (1995),

Windows 98 (1998), Windows Me (2000), Windows XP (2001), and Windows Vista (2006), windows

7(2009), windows 8(2012), windows 8.1(2013).

Windows 11 is a series [of operating systems](https://en.wikipedia.org/wiki/Operating_system) developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) and released as part of its [Windows NT](https://en.wikipedia.org/wiki/Windows_NT) family of operating systems. It is the successor to [Windows 8.1](https://en.wikipedia.org/wiki/Windows_8.1), released nearly two years earlier, and was [released](https://en.wikipedia.org/wiki/Software_release_cycle#Release_to_manufacturing_(RTM)) to manufacturing on July 15, 2015, and broadly released for the general public on July 29, 2015 .Windows Vista and Windows 7 have followed the same release strategy. Windows is designed to run on standard [x86 har](https://techterms.com/definition/x86)dware, such as Intel and AMD [processors. Th](https://techterms.com/definition/processor)erefore, it can be installed on multiple brands of hardware, such as Dell, HP, and Sony computers, as well as home-built PCs.

During upgrades, Windows 11 licenses are not tied directly to a product key. Instead, the license status of the system's current installation of Windows is migrated, and a "Digital license" (also known as "Digital entitlement" in version 1511 or earlier) is generated during the activation process.

# .NET FRAMEWORK:

.NET Framework (pronounced as "dot net") is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages.

Programs written for .NET Framework execute in a software environment (in contrast to a hardware environment) named Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling.

As such, computer code written using .NET Framework is called "managed code". FCL and CLR together constitute the .NET Framework. FCL provides [user interface,](https://en.wikipedia.org/wiki/User_interface) [data access,](https://en.wikipedia.org/wiki/User_interface) [database](https://en.wikipedia.org/wiki/Database_connection) [connectivity,](https://en.wikipedia.org/wiki/Database_connection) [cryptography,](https://en.wikipedia.org/wiki/Database_connection) [web](https://en.wikipedia.org/wiki/Web_application) [a](https://en.wikipedia.org/wiki/Web_application)pplication d[ev](https://en.wikipedia.org/wiki/Web_application)elopment, numeric algorithms, a[nd](https://en.wikipedia.org/wiki/Algorithm) [network](https://en.wikipedia.org/wiki/Computer_networking) [communications. Pro](https://en.wikipedia.org/wiki/Computer_networking)grammers produce software by combining their [source](https://en.wikipedia.org/wiki/Source_code) [code wit](https://en.wikipedia.org/wiki/Source_code)h .NET Framework and other libraries.

The framework is intended to be used by newest applications created for the Windows platform. Microsoft also produces an [integrated development environment lar](https://en.wikipedia.org/wiki/Integrated_development_environment)gely for .NET software called [Visual Studio**.**](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio) **.**NET Framework began as [proprietary software, alt](https://en.wikipedia.org/wiki/Proprietary_software)hough the firm worked to [standardize th](https://en.wikipedia.org/wiki/Software_standard)e software stack almost immediately, even before its first release.

Despite the standardization efforts, developers, mainly those in the [free](https://en.wikipedia.org/wiki/Free_and_open-source_software) [and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) [co](https://en.wikipedia.org/wiki/Free_and_open-source_software)mmunities, expressed their unease with the selected terms and the prospects of any free and open- source implementation, especially regarding [software patents. Sin](https://en.wikipedia.org/wiki/Software_patent)ce then, Microsoft has changed

.NET development to more closely follow a contemporary model of a community developed software project, including issuing an update to its patent promising to address the concerns.

# TESTING:

Testing is a vital part of software development, and it is important to start it as early as possible, and to make testing a part of the process of deciding requirements. To get the most useful perspective on your development project, it is worthwhile devoting some thought to the entire lifecycle including how feedback from users will influence the future of the application.

The tools and techniques we've discussed in this book should help your team to be more responsive to changes without extra cost, despite the necessarily wide variety of different development processes. Nevertheless, new tools and process improvements should be adopted gradually, assessing the results after each step.

Testing is part of a life cycle. The software development lifecycle is one in which you hear of a need, you write some code to fulfil it, and then you check to see whether you have pleased the stakeholders— the users, owners, and other people who have an interest in what the software does.

Hopefully they like it, but would also like some additions or changes, so you update or augment your code; and so the cycle continues. This cycle might happen every few days, as it does in Fabrikam's ice cream vending project, or every few years, as it does in Contoso's carefully specified and tested healthcare support system. Software development lifecycle Testing is a proxy for the customer.

You could conceivably do your testing by releasing it into the wild and waiting for the complaints and compliments to come back. Some companies have been accused of having such a strategy as their business model even before it became fashionable. But on the whole, the books are better balanced by trying to make sure that the software will satisfy the customer before we hand it over.

We therefore design tests based on the stakeholders' needs, and run the tests before the product reaches the users. Preferably well before then, so as not to waste our time working on something that isn't going to do the job.

In this light, two important principles become clear:

**TESTS REPRESENT REQUIREMENTS**: Whether you write user stories on sticky notes on the wall, or use cases in a big thick document, your tests should be derived from and linked to those requirements. And as we've said, devising tests is a good vehicle for discussing the requirements.

**WE ARE NOT DONE TILL THE TEST IS PASSED:** The only useful measure of completion is when tests have been performed successfully.

Those principles apply no matter how you develop your software.

## SOFTWARE TESTING TYPES:

**BLACK BOX TESTING** – Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

**WHITE BOX TESTING** – This testing is based on knowledge of the internal logic of an application’s code. Also known as Glass box Testing. Internal software and code working should be known forthis type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

**UNIT TESTING** – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.

**INCREMENTAL INTEGRATION TESTING** – Bottom up approach for testing i.e continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately. done by programmers or by testers.

**INTEGRATION TESTING** – Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

**FUNCTIONAL TESTING** – This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

**SYSTEM TESTING** – Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

**END-TO-END TESTING** – Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate

**SANITY TESTING** – Testing to determine if a new software version is performing well enough to accept it for a major testing effort. If application is crashing for initial use then system is not stable enough for further testing and build or application is assigned to fix.

**REGRESSION TESTING** – Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

**ACCEPTANCE TESTING** -Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer do this testing to determine whether to accept application.

**LOAD TESTING**– Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

**STRESS TESTING**– System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.

**PERFORMANCE TESTING** – Term often used interchangeably with ‘stress’ and ‘load’ testing. To check whether system meets performance requirements. Used different performance and load tools to do this.

**USABILITY TESTING** – User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically system navigation is checked in this testing.

**INSTALL/UNINSTALL TESTING**– Tested for full, partial, or upgrade install/uninstall processes on different operating systems under different hardware, software environment.

**RECOVERY TESTIG** – Testing how well a system recovers from crashes, hardware failures, or other catastrophic problems.

**SECURITY TESTING** – Can system be penetrated by any hacking way. Testing how well the system protects against unauthorized internal or external access.

Checked if system, database is safe from external attacks.

**COMPABILITY TESTING**– Testing how well software performs in a particular hardware/software/operating system/network environment and different combination s of above.

**COMPARISON TESTING**– Comparison of product strengths and weaknesses with previous versions or other similar products.

**ALPHA TESTING**– In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing.

**BETA TESTING** – Testing typically done by end-users or others. Final testing before releasing application for commercial purpose.

# CONCLUSION AND FUTURE ENHANCEMENT

## CONCLUSION:

In conclusion, the Human Resources Management System (HRMS) project offers a comprehensive solution for streamlining HR processes, managing employee data, facilitating communication, and supporting strategic decision-making within the organization. Through the implementation of various modules such as employee management, recruitment, performance evaluation, payroll administration, and compliance management, the HRMS aims to improve operational efficiency, enhance employee engagement, and ensure compliance with regulatory requirements.

Furthermore, the HRMS facilitates communication and collaboration among employees, managers, and HR personnel through features such as self-service portals, internal messaging systems, and news and announcement modules. This enables seamless communication, dissemination of information, and collaboration across the organization, fostering a culture of transparency and engagement.

Moreover, with access to real-time data and insights, decision-makers can make informed and strategic decisions. HR analytics tools and customizable reports provide valuable insights into workforce trends, performance metrics, and compliance status, enabling organizations to identify areas for improvement and make data-driven decisions that align with business objectives.

Overall, the HRMS project aims to improve operational efficiency, enhance employee engagement, and ensure compliance with regulatory requirements. By leveraging technology, automation, and data-driven insights, the HRMS enables organizations to effectively manage their most valuable asset – their people – and drive sustainable growth and success.

## 

## FUTURE ENHANCEMENT: Future enhancements for the Human Resource Management System could include:

## 1. Advanced Analytics and Predictive Insights: Incorporate advanced analytics capabilities to derive insights from HR data, predict workforce trends, and identify opportunities for improvement.

## 2. Enhanced Self-Service Functionality: Expand employee and manager self-service capabilities to empower users with greater control over HR processes, such as leave management, performance feedback, and training enrollment.

## 3. Integration with Emerging Technologies: Explore integration with emerging technologies such as artificial intelligence (AI), machine learning (ML), and natural language processing (NLP) to automate routine tasks, personalize user experiences, and improve decision-making.

## 4. Mobile Application Development: Develop a mobile application for the HRMS to enable employees and managers to access HR functionalities on-the-go, enhancing accessibility and convenience.

## 5. Employee Well-being and Engagement Initiatives: Introduce features focused on employee well-being and engagement, such as wellness programs, social recognition platforms, and employee feedback mechanisms, to foster a positive work culture.

## 6. Enhanced Security Measures: Implement additional security measures, such as biometric authentication, data encryption, and advanced threat detection, to safeguard sensitive HR data from cybersecurity threats.

## 7. Integration with External Systems: Strengthen integration capabilities with external systems, such as financial management, procurement, and customer relationship management (CRM) systems, to enable seamless data exchange and holistic business insights.

# BIBLIOGRAPHY

* Geeks for Geeks
* w3schools
* Programmz
* Stack Overflow
* Tutorial spoint
* Java point

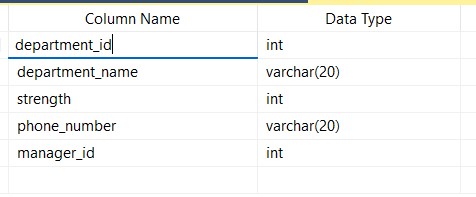
# APPENDICES A-TABLE STRUCTURE:

## EMPLOYEE:

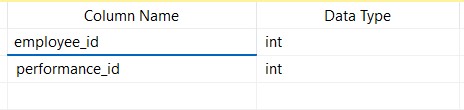
## 

## 

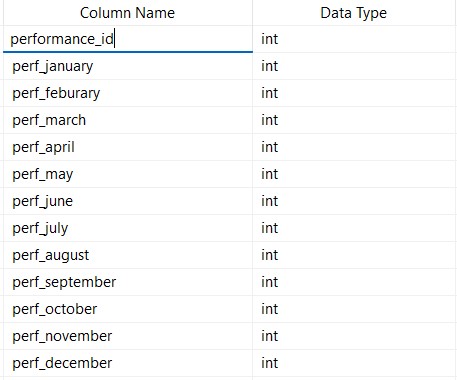
**DEPARTMENT:**



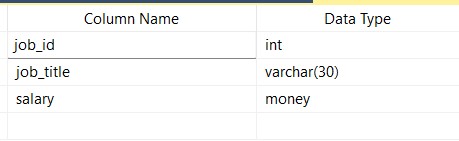
## PERFORMANCE:



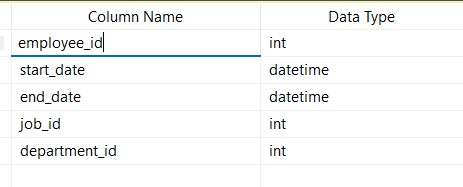
**PERFORMANCE ALL:**



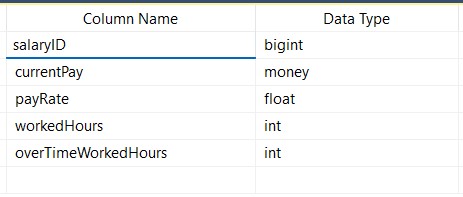
**EMPLOYEE JOB INFO :**



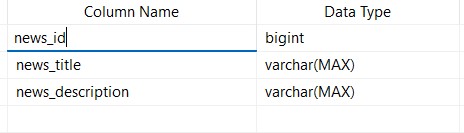
## EMPLOYEE JOB HISTORY:



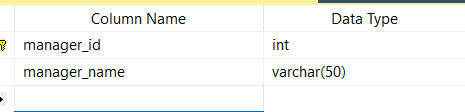
**EMPLOYEE SALARY INFO:**



**NEWS :**

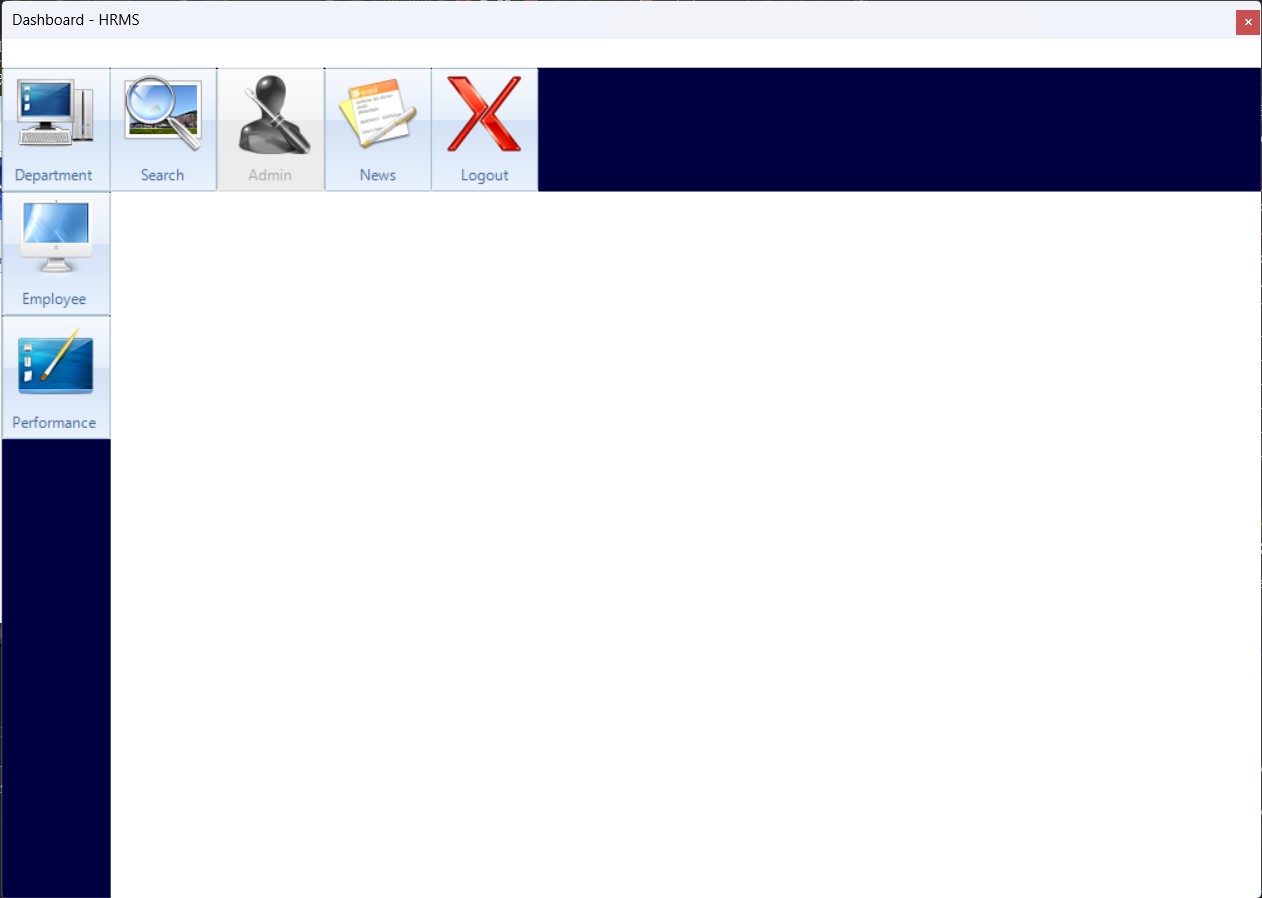


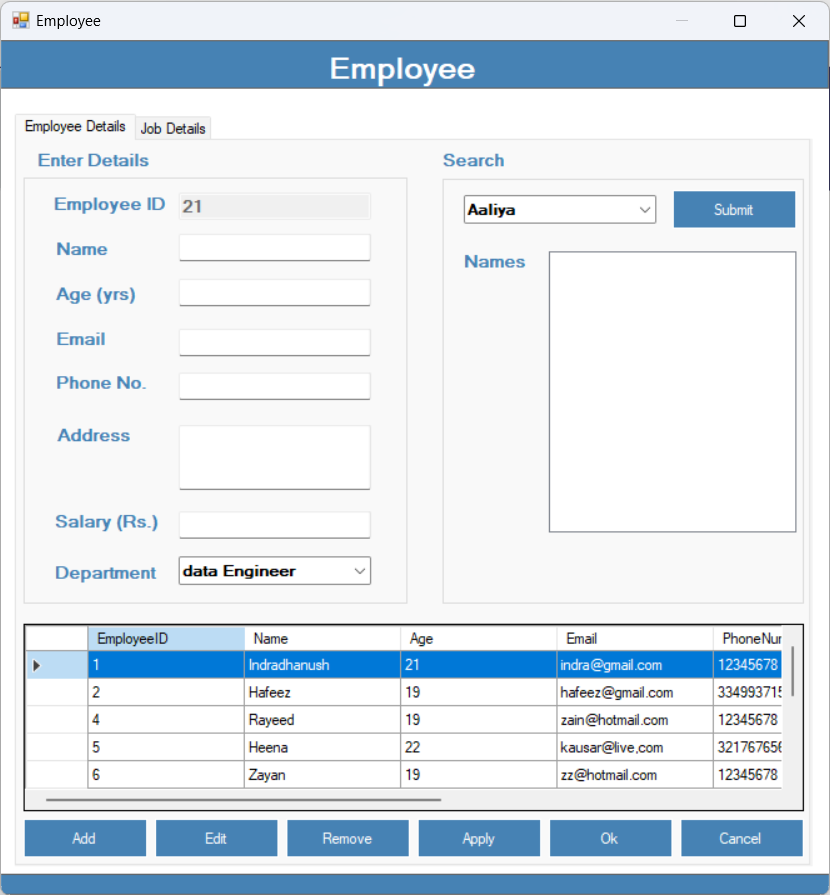
**MANAGER:**

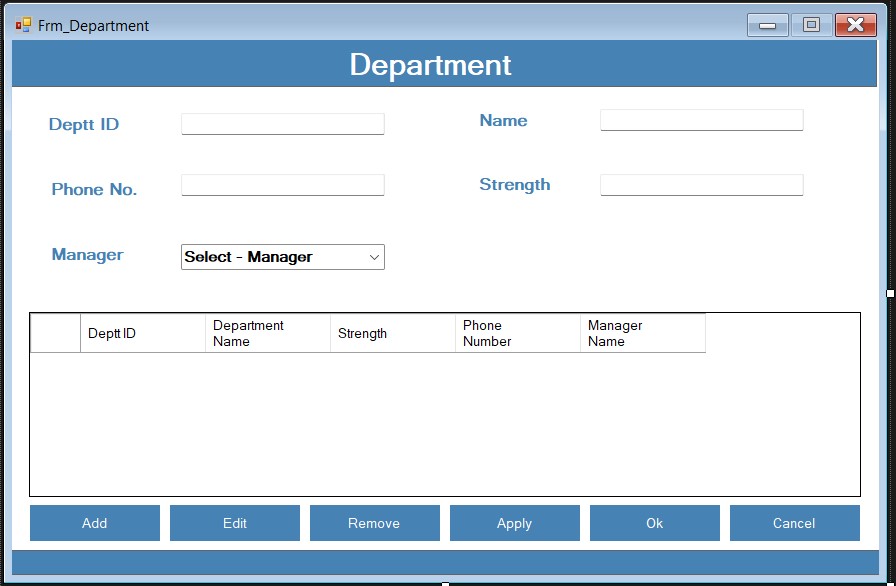
****

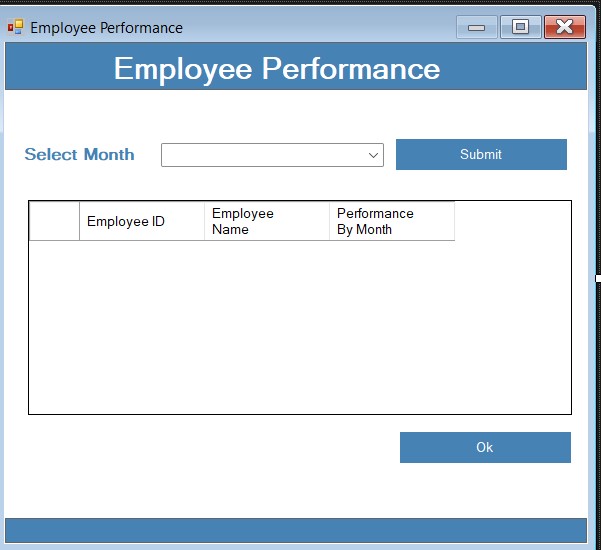
# APPENDICES-B SCREENSHOTS:

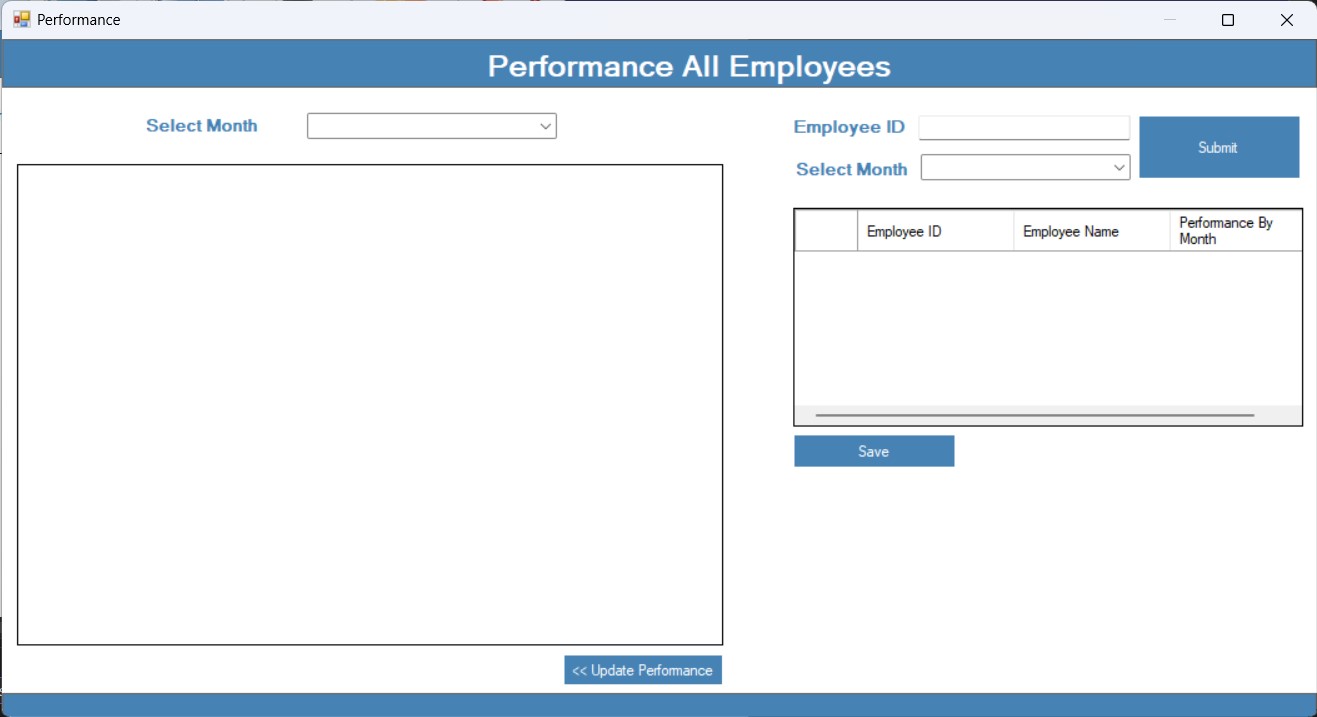






****



****

****



**11.APPENDICES C-SAMPLE REPORT OF TEST CASES:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl**  **No.** | **Test**  **Case Id** | **Test**  **Description** | **Steps to**  **execute** | **Test data** | **Expected**  **result** | **Actual result** | **Status** |
| 1 | Test  Case Id 1 | Correct user  name and password | Start-enter  the username, password  login | Correct  username and password | Straight to the dashboard | Straight to the dashboard | Pass |
| 2 | Test  Case Id 2 | Wrong user  name and password | Start-enter  the username, password  login | Correct  username and wrong Password | Must Enter the Valid login  details | Must Enter the Valid login  details | Pass |
| 3 | Test  Case Id 3 | Adding an employee | Start-enter,  enter the  details, and add | Enter all the  details. | Successfully  added. | Successfully  added. | Pass |
| 4 | Test  Case Id 4 | updating  employee performance | Start-login,  enter the  details and submit. | The employee id is wrong whereas the month is correct | Input string invalid | Input string invalid. | Pass |

