Kristu Jayanti College
AUTONOMOUS Bengaluru
Reaccredited 'A' Grade by NAAC | Affiliated to Bangalore University

"MOVIE STREAMING APP - BINGER"

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

BACHELOR OF COMPUTER APPLICATIONS (BCA)

Submitted By

Haseef Azhaan

19CS1H2143

Under the guidance of

Prof Hubert Shanthan B J

Prof Gladys Deborah

DEPARTMENT OF COMPUTER SCIENCE (UG)
BCA PROGRAMME
KRISTU JAYANTI COLLEGE (Autonomous)
K. Narayanapura, Kothanur P.O., Bangalore – 560077

Kristu Jayanti College
AUTONOMOUS Bengaluru
Reaccredited 'A' Grade by NAAC | Affiliated to Bangalore University

DEPARTMENT OF COMPUTER SCIENCE (UG)

CERTIFICATE OF COMPLETION

This is to certify that the project entitled **"Movie Streaming App - Binger"** has been satisfactorily completed by **Haseef Azhaan**, 19CS1H2143 in partial fulfillment of the award of the Bachelor of Computer Applications degree requirements prescribed by Kristu Jayanti College (Autonomous) Bengaluru (Affiliated to Bangalore University) during the academic year 2020 -21.

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

Valued by Examiners

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

2:_____ ***Date:***

DECLARATION

I, Haseef Azhaan, Regno:19CS1H2143 hereby declare that the project work entitled Movie Streaming App – 'Binger' is an original project work carried out by me, under the guidance of Prof. Margaret Mary T and Prof Hubert Shanthan B J.

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
(Haseef Azhaan)

Bengaluru
Date:

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**, and **BANGALORE** 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. Hubert Shanthan B J** 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 successfully.

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROBLEM DEFINITION

During This Pandemic, The Film/Cinema Industry Has A Huge Impact, There Was No Source of Entertainment That Could Reach A Solo Individual, Since All the Movie Theatre and Cinema Houses Were Shutdown. This Movie App 'BINGER' Provides a Platform for All the Cinematic Entertainment at Home, So That It Can Reach All the Individual During This Scenario. This App Also Helps Its User to Be Entertained and Reduces the Struggle of Searching the Movies by Its Special Algorithm to Provide A Quality Content to Each User.

1.2 OBJECTIVES OF THE APP

The Objective of This "BINGER" App Is to Provide Quality Entertainment At Home.

- Easy and highly interactive user interface.
- Easy to maintain all the records of movies that are been watched by the user.
- Discover movies based on special search algorithm
- Provide a useful insight to the admin to produce better and quality movies
- Since it is a streaming app it reduces the time of download

1.3 MODULES

- User Management
- Movie Recommendation
- Movie Information
- Movie Player
- User History And Collection Management
- User Detail Management
- Admin Page
- Visual Reports Generator

USER MANAGEMENT – This Module Helps to Validate User Based on Credentials Specified and To Register New User Using Their Details.

MOVIE RECOMMENDATION - This Module Helps the User to Find New Movies Based on Title of Movie, Genres or Ratings of The Movie.

In This, The User Can Also Get Movies Similar to The Genre, Cast, Director Which He Has Already Watched

MOVIE INFORMATION - It Displays the Description of a Movie Which Is Selected, This Description Include Title of Movie, Genres, Star Casts, Plot, Directors of The Movie

MOVIE PLAYER - Play's movie Selected by The User.

USER HISTORY AND COLLECTION MANAGEMENT - This Module helps in finding previously watched movies By the User, And Store the Movie If He Wishes To Watch Later.

USER DETAIL MANAGEMENT - Includes an O0verview Of the User, This Helps To update The Details of The User, such as Password, Phone Number, Country, Email, Account alive, etc.

ADMIN PAGE - This Module is Responsible for Adding, Removing and Updating

the Movies. This Page Is Only Restricted For Company For Handling The Application.

VISUAL REPORT GENERATOR - In This Module A Visual Report Is Generated to The Admin Handling the Movies Insertion and Deletion. This Module Helps to Provide A Useful Insight About the Movie Liked by User, Pattern of User Watching Movies, Time Spent By Each User In Application, etc.

2.SYSTEM STUDY

2.1 EXISTING SYSTEM

The Study Of Existing Involves Gathering Details To Find Out What Are The Requirements, How It Works, And Where Improvements Should Be Made, It's Basically a Preliminary Investigation.

In the Current System, The Movies are displayed in cinemas and the Theatre, where it is very unsecure and chances of spreading the pandemic is very high. In this type.

The user has to specifically visit a place where he can watch the movie, and the cost of maintaining such type of system is also high.

The user has to first buy movie ticket and then he has to visit the thether in the time allotted to view that specific movie, And the Movie ticket are also limited in numbers. If the tickets are sold out then the customer need to wait for the next round of allotment.

There is no flexibility in this type of system.

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. 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. Feasibility studies also allow a business to address where and how it will operate, potential obstacles, competition and the funding needed to get the business up and running.

TECHNICAL FEASIBILITY

Technical feasibility is providing the estimation for a minimum amount of technical skills required, Upon the estimation, a developer can predict whether they must gain knowledge

or not. Which will ultimately affect the time and economic feasibility? This is a cascading process.

In this estimation following are the requirements.

- This system is maintainable for new updates or release
- This system should be secure, scalable and optimum.
- This system is responsive.

When all the requirements are met the system is said to be technically feasible

OPERATIONAL FEASIBILITY

In this step, all different operational factors are determined of the proposed systems like work force, time, etc. to design, develop and maintain this system. whichever solution uses least amount of effort, is the best operationally feasible solution. Operational Feasibility determines whether the proposed system satisfies all clients' demands and functionalities. Work culture, style of executions, planning and the design of the product should be accepted by all the members. The proposed system will not cause any problem under any circumstances to the end-user and works properly .In this feasibility study the time and personal requirements are satisfied. so this solution is operationally feasible.

ECONOMIC FEASIBILITY

Economic feasibility is one of the major factors. In this project one say that this project is financially feasible because it requires no extra financial investment, this project can be completed in six months. In economic feasibility, existing projects are compared based on future financial benefits of the proposed system with the equity, return on investment is 8 calculated. Profits are expected on the investments. If a system gives more returns compared to investments then the system is financially feasible. Economic Feasibility will conclude whether the project can be done from available resource or not. Financial benefits should be greater then investment.

In this feasibility following are checked:

➢ Cost price for full system investigation.

➢ The installation and operational cost for the system.

➢ The cost during the development includes tools.

The returns of this project comparing to the investments is very good. So this project is economically feasible because it is giving more returns compare to investments.

LEGAL FEASIBILITY

Legal Feasibility encompasses a broad range of concern that includes contacts, liability, infringement and myriad other taps frequently unknown to the technical staff. A determination of any infringement, violation or liability that could result from the development of the app.

- **SCHEDULE:**

1.Time duration of this project requires 80 days covering 10 days for initiation phase, 7 days for definition phase, 25 days for Design phase, 25 days for implementation phase, 8 days for Testing phase and 5 days for documentation.

2.Resource persons are HASEEF AZHAAN and SAHANA.S

- **OPERATIONAL:**

❖ Reduction of paper work.

❖ Human effort or manual labour can be reduced drastically.

❖ Major operations that are done manually can be done within a matter of few seconds.

❖ History storing is easier and backup is available.

2.3 PROPOSED SYSTEM

This application works on the average function which runs in two stages. In first step, average function calculates the rating on each crew member which is stored in the dataset, the output generated in first stage is given as an input to the second stage.

In the final stage the application gives the verdict whether movie is hit or flop. Function is based on the average calculated on the team members and a result is generated. The version of vb.net should be greater than 16 and visual studio 2019. These requirements specified are the minimum, any version below will not support the application.

If a user demands the result of a movie and the data for the movie is not available in the historical dataset for some crew members. then, the user can add the rating for that missing crew member. Yet, if data is not available then it will not affect the output generated. After the result is produced it is saved in the database once for each movie and respective crew members. Over time dynamics in data keeps training the system.

ADVANTAGES OF PROPOSED SYSTEM

- ✔ Minimise the workload of users.
- ✔ It is user friendly.
- ✔ Easy to view histories stored in database.
- ✔ Feedback can be given by the watched users
- ✔ Good quality movies based on genres.

OBJECTIVES OF PROPOSED SYSTEM

The study of existing system is the exact nature of problem manual system. Also, if has been decided that problem in the existing system can be solved only through computerization.

The existing system has been studied through observing user validation like number, email, name, country and also Movie Id, name, ratings, release date, description, User history, favourite and star cast.

3. System Design

It Is Also One of The Most Crucial Step In Software Development. In the design phase, the architecture is established. This phase starts with the requirement document delivered by the requirement phase und maps the requirements into an architecture.

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.

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 system analysis, systems architecture and systems engineering,

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 viu 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 i**nto 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 hack to them.

Data Design is concerned with how the data is represented and stored within the system.

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.

3.1 ER 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. 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.

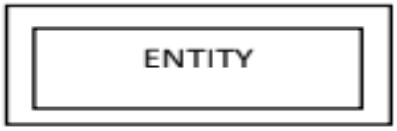
- Make effective use of colours. You can use colours to classify similar entities or to highlight key areas in your diagrams.

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 flow.io to craft visually engaging and precise ER diagrams. The ER Diagram Software offered by flow.io 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. 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.

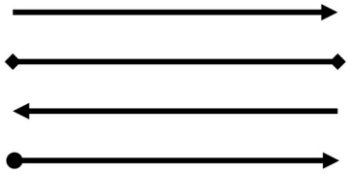


- **Relationship** , which are represented by diamond shapes, show how two entities share information in the database. In some cases, entities can be self-linked.

- **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity.

- A **multivalve attribute** can have more than one value. A derived attribute is based on another attribute.

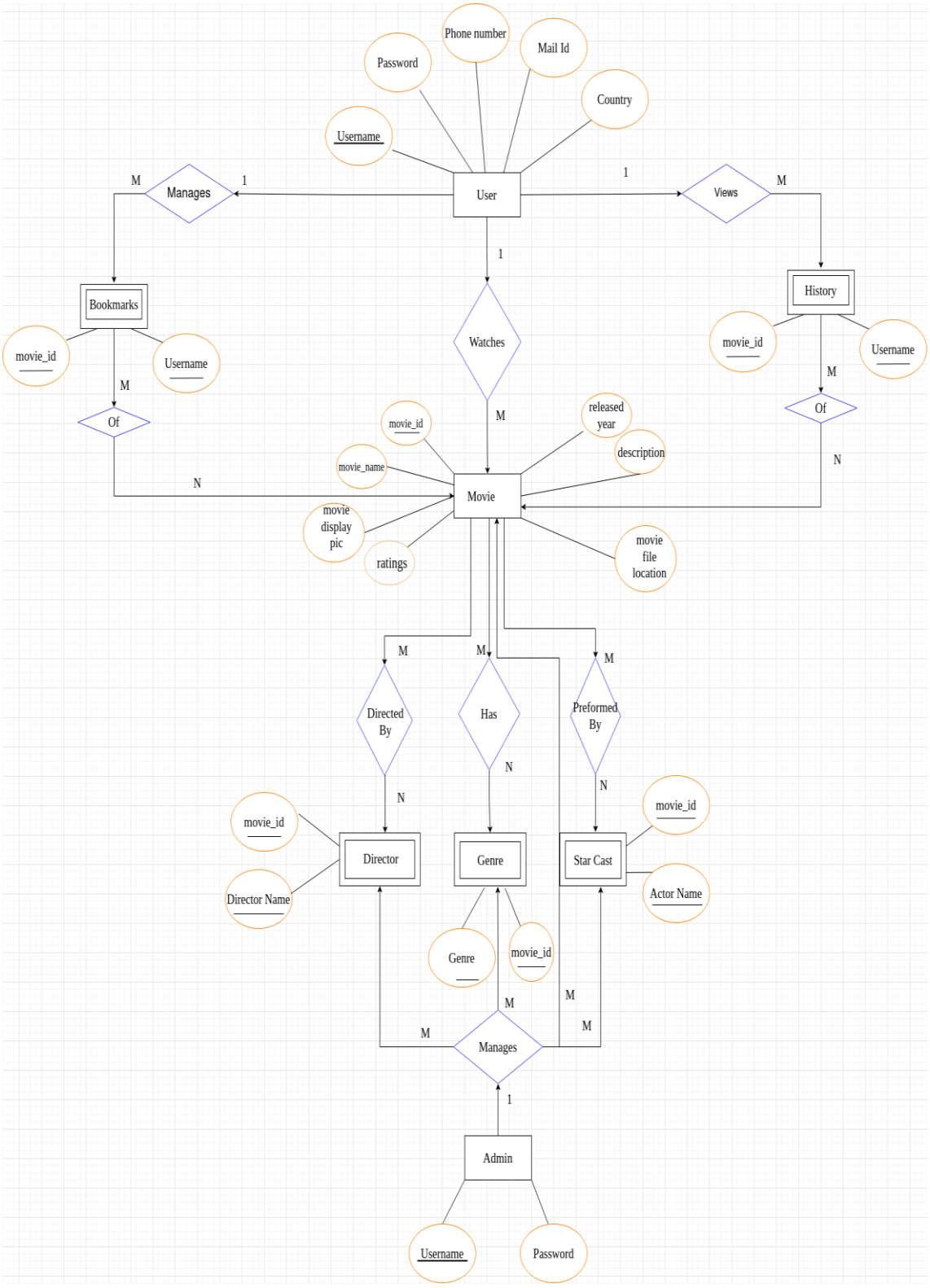
- **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the ER Diagram.



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


Phone number
Password
Mail Id
Country
Username
User
Manages
Views
Bookmarks
movie_id
Username
Of
Watches
History
movie_id
Username
Of
released year
description
movie_id
movie_name
Movie
movie display pic
ratings
movie file location
Directed By
Has
Preformed By
movie_id
Director
Director Name
Genre
Genre
movie_id
Star Cast
movie_id
Actor Name
Manages
Admin
Username
Password

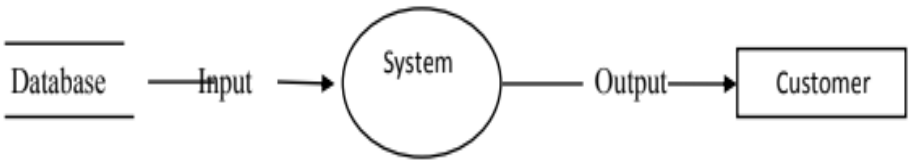
3.2 Data flow diagram (level 0 and level 1)

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.

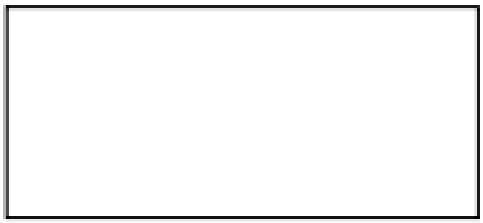
A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data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

Theory:

Data flow diagram example:



Entity



This Represents an **Entity.** A source of data or a destination for data.

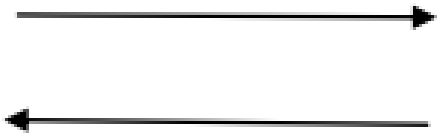
Process/Function

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.

Database

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.

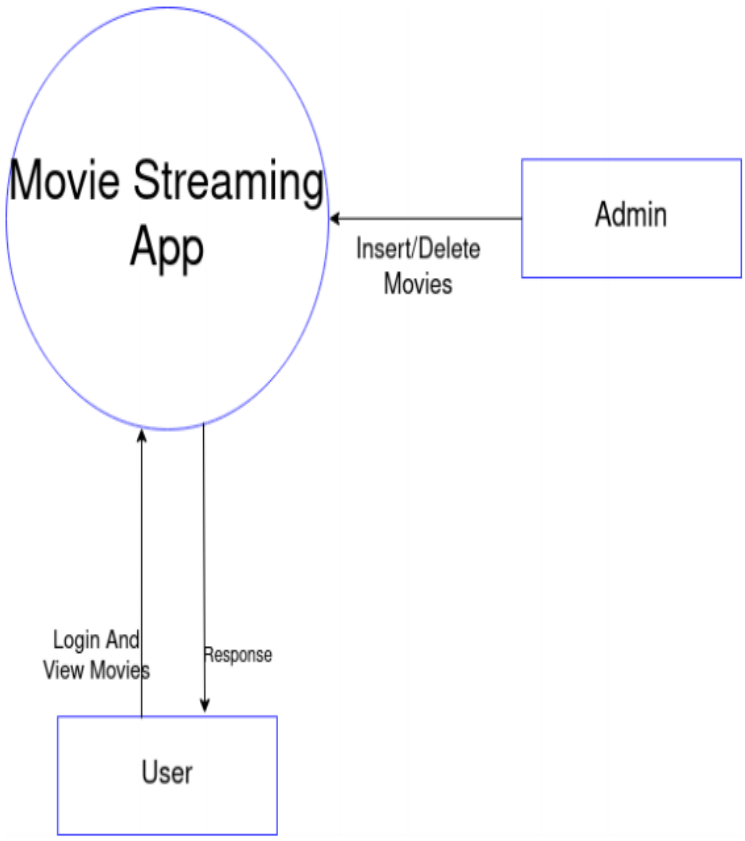
Data Flow



Data Flow: Represented by a unidirectional arrow. Data Flows show how data is moved through the System. Data Flows are labelled with a description of the data that is being passed through it.

Data flow diagrams are also known as **bubble charts**. DFD is a designing tool used in the top-down approach to Systems Design. This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modelled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present.In order for the system to do its job, and shows the flow of data between the various parts of the system. 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 identify the transaction data in the data model.

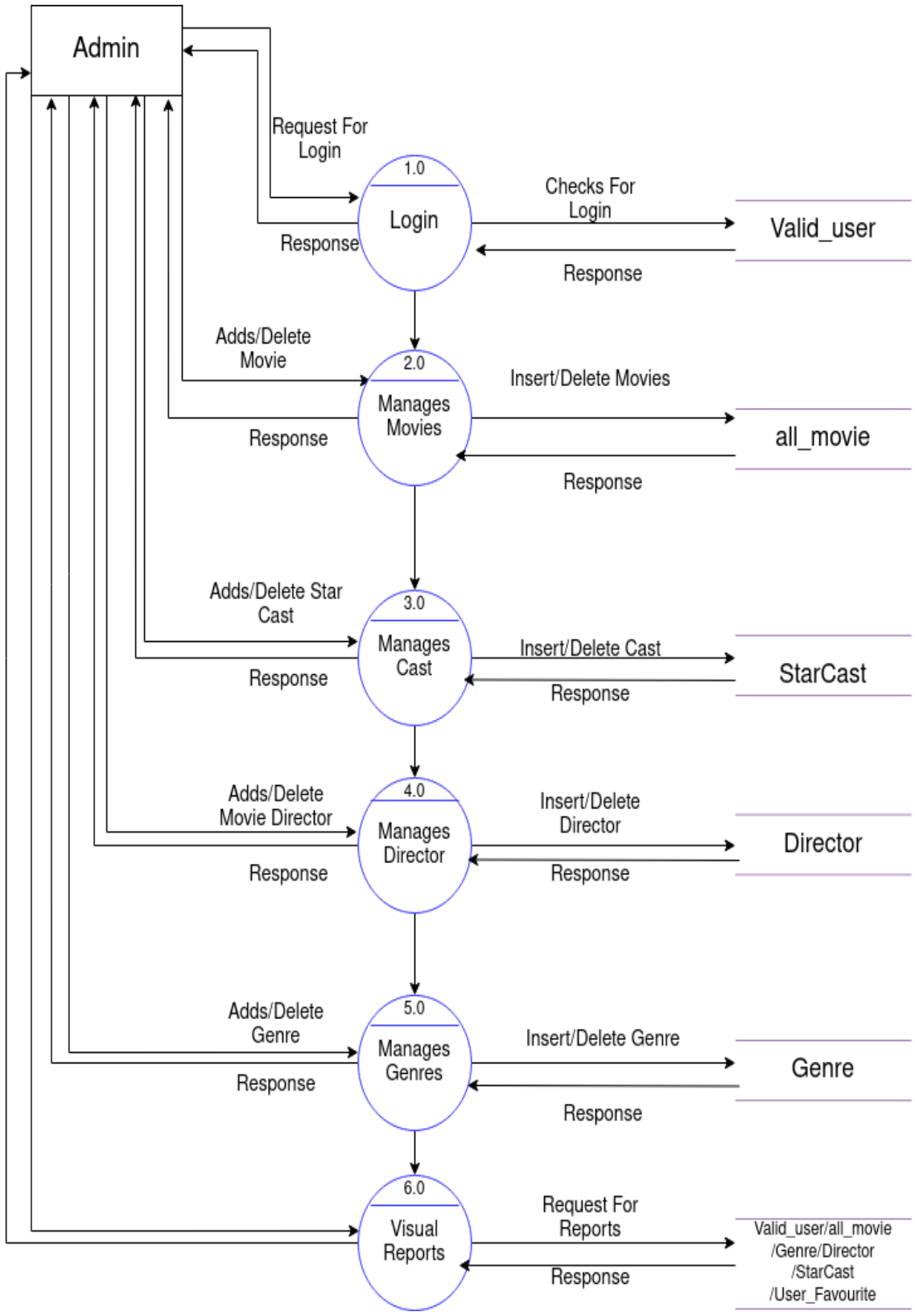
Level-0 DFD.



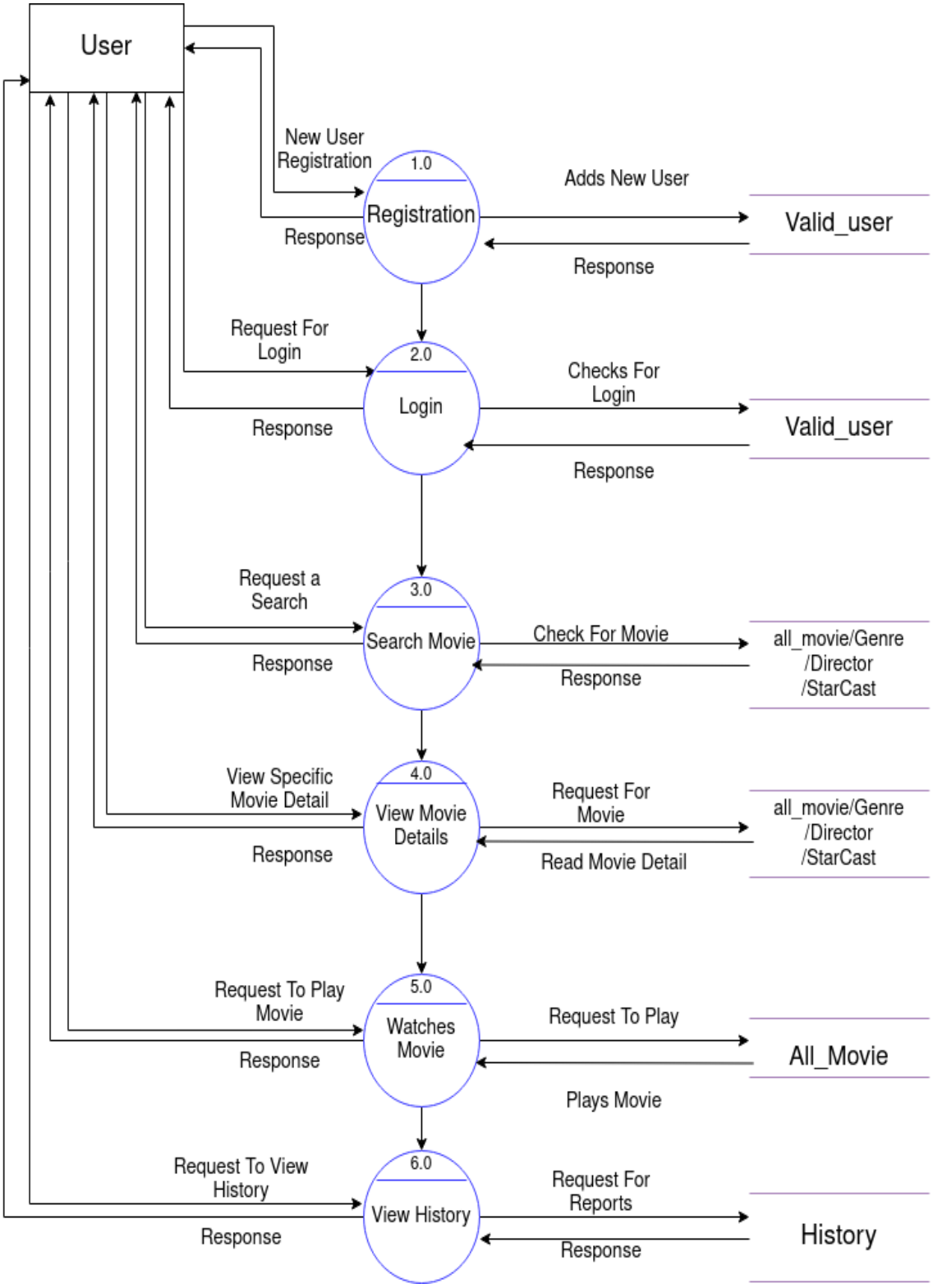
Level 1 DFD.

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 sub-processes. 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 sub-processes. 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 assummade from the text.

Level 1 DFD (FOR ADMIN)



DFD LEVEL -1 (FOR USER)



3.3 GANTT CHART

A Gantt chart is a type of bar chart, devised by Henry Gantt in the 1910s, 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.

GANTT CHART BENEFITS:

Clarity: One of the biggest benefits of a Gantt chart is the tool's 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.

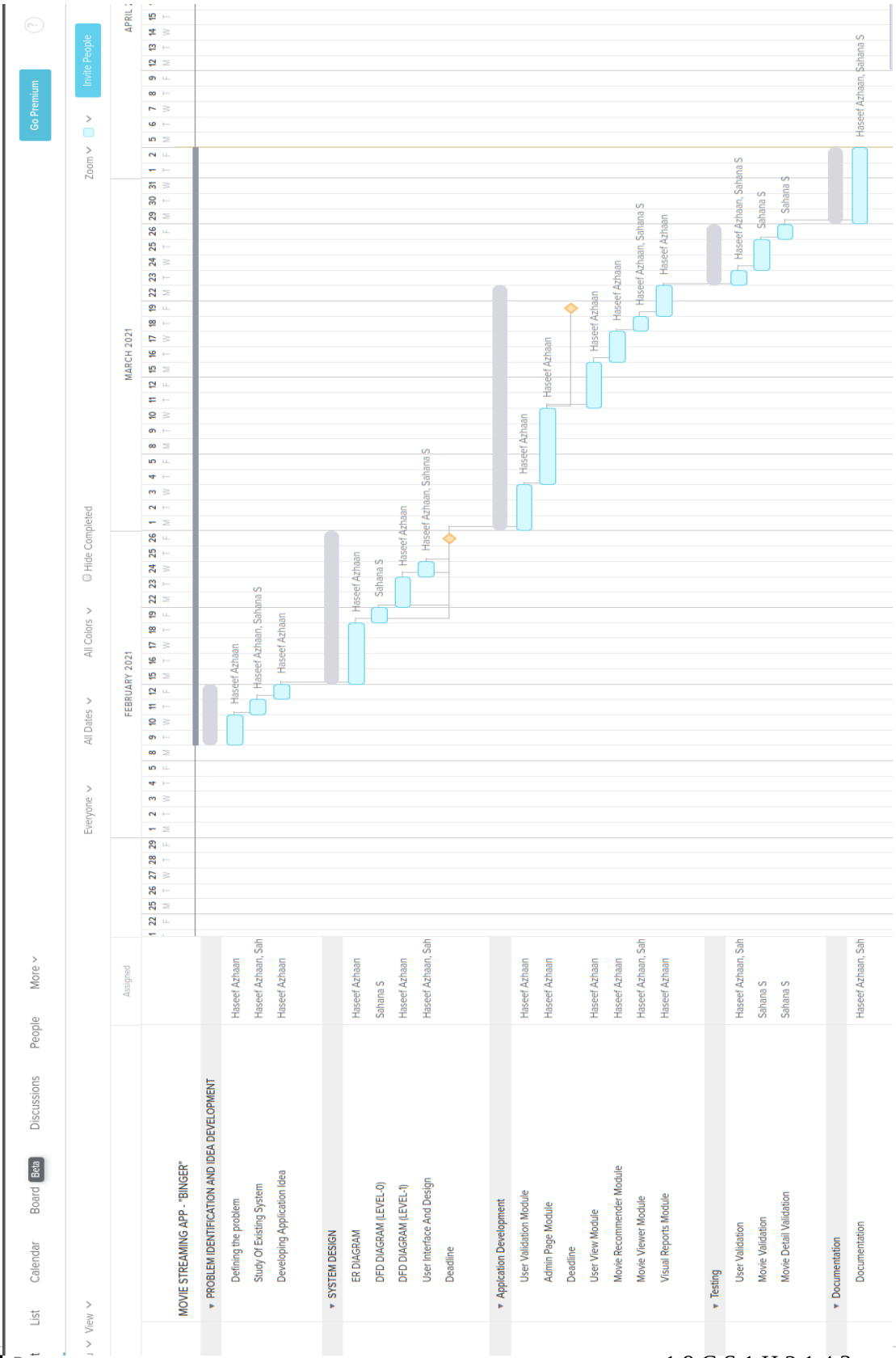
Communication: Teams can use Gantt charts to replace meetings and enhance other status updates. Simply clarifying chart positions offers an easy, visual method to help team members understand task progress.

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.

Efficiency: Another one of the benefits of Gantt charts is the ability for team‘s 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.

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.

Gantt Chart


Go Premium
Invite People
List
Calendar
Board
Beta
Discussions
People
More
Everyone
All Dates
All Colors
Hide Completed
Zoom
FEBRUARY 2021
MARCH 2021
APRIL
Assigned
MOVIE STREAMING APP - "BINGER"
PROBLEM IDENTIFICATION AND IDEA DEVELOPMENT
Defining the problem
Study Of Existing System
Developing Application Idea
SYSTEM DESIGN
ER DIAGRAM
DFD DIAGRAM (LEVEL-0)
DFD DIAGRAM (LEVEL-1)
User Interface And Design
Deadline
Application Development
User Validation Module
Admin Page Module
Deadline
User View Module
Movie Recommender Module
Movie Viewer Module
Visual Reports Module
Testing
User Validation
Movie Validation
Movie Detail Validation
Documentation
Documentation
Haseef Azhaan
Haseef Azhaan, Sahana S
Sahana S

3.4 Input/Output Design

Login Module

```
Imports System.Data.SqlClient

Public Class login_page
    Dim con As New SqlConnection
    Public usr
Private Sub login_page_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'connection string
con.ConnectionString="datasource=(localdb)\mssqllocaldb;attachdbfilename=c:\users\haseef\source\repos\
login_page\login_page\database1.mdf;integrated security=true"
        con.Open()
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        validation()
    End Sub
    Private Sub validation()
        Dim cmd As New SqlCommand
        Dim sql As String
        Dim dr As SqlDataReader
        'sql command to validate if user is in database
        sql = "select * from valid_user where username  = '" + (TextBox1.Text) + "'  AND Passwd ='" +
(TextBox2.Text) + "'"
        cmd.Connection = con
        cmd.CommandText = sql
        dr = cmd.ExecuteReader
        If dr.Read Then
            'checks if he is has admin privelage or not
            If dr("isadmin") = 1 Then
                admin_page.Show()
            Else
                main_page.user.Text = TextBox1.Text
                main_page.Show()
                main_page.Setuserdp()
                Me.Hide()
            End If
            Me.Hide()
```

```
        TextBox1.Text = "Username"
        TextBox2.Text = "Password"
        TextBox2.PasswordChar = ""
    Else
        invalidtext.Visible = True
        Panel1.BackColor = Color.Red
        Panel2.BackColor = Color.Red
    End If
    dr.Close()
End Sub
Private Sub Exit_button_Click(sender As Object, e As EventArgs) Handles Exit_button.Click
    Application.Exit()
End Sub
Private Sub Mini_button_Click(sender As Object, e As EventArgs) Handles Mini_button.Click
    Me.WindowState = FormWindowState.Minimized
End Sub
Private Sub Max_button_Click(sender As Object, e As EventArgs) Handles Max_button.Click
    If Me.WindowState = FormWindowState.Normal Then
        Me.WindowState = FormWindowState.Maximized
    Else
        Me.WindowState = FormWindowState.Normal
    End If
End Sub

'All Below is for ui design
Private Sub Button1_MouseHover(sender As Object, e As EventArgs) Handles Button1.MouseEnter
    ChangecoloronHover(Button1, 3, Color.FromArgb(105, 102, 102))
End Sub
Private Sub Button1_MouseLeave(sender As Object, e As EventArgs) Handles Button1.MouseLeave
    ChangecoloronHover(Button1, 0, Color.FromArgb(101, 101, 219))
End Sub
Private Sub TextBox2_Click(sender As Object, e As MouseEventArgs) Handles TextBox2.Click
    TextBox2.Text = ""
    TextBox2.PasswordChar = "*"
    Panel2.BackColor = Color.White
    invalidtext.Visible = False
End Sub
```

```
Private Sub TextBox1_Click(sender As Object, e As EventArgs) Handles TextBox1.Click
    TextBox1.Text = ""
    Panel1.BackColor = Color.White
    invalidtext.Visible = False
End Sub
Private Sub ChangecoloronHover(ByRef btn_name, ByVal border, ByVal color)
    btn_name.FlatAppearance.BorderSize = border
    btn_name.BackColor = color
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    TextBox1.Text = ""
    register.Show()
    Me.Hide()
End Sub
Private Sub Button2_MouseLeave(sender As Object, e As EventArgs) Handles Button2.MouseLeave
    ChangecoloronHover(Button2, 3, Color.FromArgb(40, 40, 41))
End Sub
Private Sub Button2_MouseHover(sender As Object, e As EventArgs) Handles Button2.MouseEnter
    ChangecoloronHover(Button2, 0, Color.FromArgb(200, 200, 204))
End Sub
Private Sub PictureBox4_MouseEnter(sender As Object, e As EventArgs) Handles PictureBox4.MouseEnter
    TextBox2.PasswordChar = ""
End Sub
Private Sub PictureBox4_MouseLeave(sender As Object, e As EventArgs) Handles PictureBox4.MouseLeave
    TextBox2.PasswordChar = "*"
End Sub
Private Sub TextBox2_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TextBox2.KeyPress
    TextBox2.PasswordChar = "*"
End Sub
Private Sub ShowOnHover(ByRef button, ByVal img)
    button.Image = Image.FromFile(img)
    'sets the image specified in the button
End Sub
```

```
    Private Sub Exit_button_MouseEnter(sender As Object, e As EventArgs) Handles Exit_button.MouseEnter
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Exit.png"
        ShowOnHover(Exit_button, img)
    End Sub
    Private Sub Exit_button_MouseLeave(sender As Object, e As EventArgs) Handles Exit_button.MouseLeave
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Red.png"
        ShowOnHover(Exit_button, img)
    End Sub
    Private Sub Mini_button_MouseEnter(sender As Object, e As EventArgs) Handles Mini_button.MouseEnter
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Minimise.png"
        ShowOnHover(Mini_button, img)
    End Sub
    Private Sub Mini_button_MouseLeave(sender As Object, e As EventArgs) Handles Mini_button.MouseLeave
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Yellow.png"
        ShowOnHover(Mini_button, img)
    End Sub
    Private Sub Max_button_MouseEnter(sender As Object, e As EventArgs) Handles Max_button.MouseEnter
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Maximise.png"
        ShowOnHover(Max_button, img)
    End Sub
    Private Sub Max_button_MouseLeave(sender As Object, e As EventArgs) Handles Max_button.MouseLeave
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Green.png"
        ShowOnHover(Max_button, img)
    End Sub
End Class
```

Register Module

```
Imports System.Data.SqlClient
Imports System.Globalization

Public Class register
```

```
    Dim con As New SqlConnection

    Dim confrugration_username = False
    Dim confrugration_phoneno = False
    Dim confrugration_password = False

    Private Sub register_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        con.ConnectionString = "data source=(localdb)\mssqllocaldb;attachdbfilename=c:\users\haseef\source\repos\login_page\login_page\database1.mdf;integrated security=true"
        con.Open()
        Country_list()
    End Sub
    Private Sub Country_list()
        For Each cl As CultureInfo In CultureInfo.GetCultures(CultureTypes.SpecificCultures)
            Dim reg As RegionInfo = New RegionInfo(cl.LCID)
            If (ComboBox1.Items.Contains(reg.EnglishName) = False) Then
                ComboBox1.Items.Add(reg.EnglishName)
            End If
            ComboBox1.Sorted = True
        Next
    End Sub
    Private Sub Button2_Click_1(sender As Object, e As EventArgs) Handles Button2.Click
        login_page.Show()
        Hide()
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Try
            Dim usrname = Me.username.Text
            Dim fulln = Me.f_name.Text
            Dim phoneno = Me.phno.Text
            Dim passwd = pass.Text
            Dim r_passwd = r_pass.Text
            Dim county = Me.ComboBox1.SelectedItem.ToString
            Dim mail = email.Text
            If confrugration_password And confrugration_username And confrugration_username Then
                Insert_new_user(usrname, fulln, passwd, phoneno, county, mail)
```

```
            result.Text = "Registered Sucessfully"
        Else
            result.Text = "Invalid Input"
        End If
        result_panel.Visible = True
    Catch ex As SqlException
        Excp_Sql(ex)
    End Try
    con.Close()
End Sub
Private Sub Excp_Sql(ByVal ex)
    MsgBox("error")
    If ex.Number = 2627 Then
        result.Text = "User Already Exist ," & vbCrLf & " please try with different" & vbCrLf & " User name"
        result_panel.Visible = True
    Else
        result.Text = "Somthing went wrong "
        Console.WriteLine(ex.ToString)
        result_panel.Visible = True
    End If
End Sub
Private Sub Insert_new_user(ByVal uname, ByVal fullname, ByVal passwd, ByVal phno, ByVal country, ByVal mail)
    Dim command As New SqlCommand
    Dim da As New SqlDataAdapter
    command.Connection = con
    command.CommandText = "INSERT INTO valid_user(username,fullname,passwd,phno,country,profile_pic,mail,isadmin) VALUES(@name,@fullname,@passwd,@phno,@country,@pp,@mail,@isadmin)"
     Dim param_name As New SqlParameter("@name", SqlDbType.VarChar, 50) With {
        .Value = uname}
    Dim param_full As New SqlParameter("@fullname", SqlDbType.VarChar, 50) With {
        .Value = fullname}
    Dim param_passwd As New SqlParameter("@passwd", SqlDbType.VarChar, 30) With {
        .Value = passwd}
    Dim param_phone As New SqlParameter("@phno", SqlDbType.BigInt) With {
```

```
        .Value = phno }
    Dim param_country As New SqlParameter("@country", SqlDbType.VarChar, 10) With {
        .Value = country }
    Dim param_pp As New SqlParameter("@pp", SqlDbType.VarChar, 150) With {
        .Value  =  "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\User-Account-PNG-Image(1)1.png"}
    Dim param_mail As New SqlParameter("@mail", SqlDbType.VarChar, 50) With {
        .Value = mail}
    Dim Isadmin As New SqlParameter("@isadmin", SqlDbType.Int, 5) With {
        .Value = 0}
    command.Parameters.Add(param_name)
    command.Parameters.Add(param_full)
    command.Parameters.Add(param_passwd)
    command.Parameters.Add(param_phone)
    command.Parameters.Add(param_country)
    command.Parameters.Add(param_pp)
    command.Parameters.Add(param_mail)
    command.Parameters.Add(Isadmin)
    da.InsertCommand = command
    Try
        da.InsertCommand.ExecuteNonQuery()
    Catch ex As SqlException
        Excp_Sql(ex)
    End Try
End Sub
Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles username.TextChanged
    If username.Text.Contains(" ") Then
        MsgBox("please Use Underscore '_' not space")
        username.Text = username.Text.Substring(0, username.Text.Length - 1)
    End If
    If username.Text.Length > 2 Then
        check_aval.Visible = True
        'Checks If username is Avaiable in database or not
        Check_Avaliablity(username.Text)
    Else
        check_aval.Visible = False
    End If
```

```
End Sub
Private Sub Check_Avaliablity(ByVal name)
    Dim cmd As New SqlCommand
    Dim sql As String
    sql = "select count(username) from valid_user where username = '" + name + "'"
    cmd.Connection = con
    cmd.CommandText = sql
    Dim ans = cmd.ExecuteScalar()
    If ans >= 1 Then
        check_aval.Text = "Not Available"
        check_aval.ForeColor = Color.Tomato
        confrugration_username = False
    Else
        check_aval.Text = "Available"
        check_aval.ForeColor = Color.Lime
        confrugration_username = True
    End If
End Sub
Private Sub phno_TextChanged(sender As Object, e As EventArgs) Handles phno.TextChanged
    Check_phno.Visible = True
    If phno.Text.Length >= 10 And phno.Text.Length < 13 Then
        If IsNumeric(phno.Text) Then
            Check_phno.Text = "Ok"
            Check_phno.ForeColor = Color.Lime
            confrugration_phoneno = True
        Else
            Check_phno.Text = "Not Valid"
            Check_phno.ForeColor = Color.Tomato
            confrugration_phoneno = False
        End If
    Else
        Check_phno.Text = "Not Valid"
        Check_phno.ForeColor = Color.Tomato
        confrugration_phoneno = False
    End If
End Sub
Private Sub email_TextChanged(sender As Object, e As EventArgs) Handles email.TextChanged
```

```
        Email_Validation.Visible = True
        If email.Text.Contains("@") And email.Text.Contains(".com") Then
            Email_Validation.Text = "Ok"
            Email_Validation.ForeColor = Color.Lime
        Else
            Email_Validation.Text = "Email Not Valid"
            Email_Validation.ForeColor = Color.Tomato
        End If
    End Sub
    Private Sub r_pass_TextChanged(sender As Object, e As EventArgs) Handles r_pass.TextChanged
        If pass.Text = r_pass.Text Then
            confrugration_password = True
            mismatch.Visible = False
        Else
            mismatch.Visible = True
            confrugration_password = False
        End If
    End Sub
    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Me.Close()
        login_page.Show()
    End Sub
End Class
```

Movie Selection Module

```
Imports System.Data.SqlClient
Public Class main_page
    Dim con As New SqlConnection
    Dim count As Integer = 0
    Dim file As String
    Dim director As String
    Dim stars As String
    Dim genre As String
    Public Userid As String
    Dim watched_movies As New List(Of String)()
    Dim clickedMovie As String
```

```
Private Sub Exit_button_Click(sender As Object, e As EventArgs) Handles Exit_button.Click
    Application.Exit()
End Sub
Private Sub Mini_button_Click(sender As Object, e As EventArgs) Handles Mini_button.Click
    Me.WindowState = FormWindowState.Minimized
End Sub
Private Sub Max_button_Click(sender As Object, e As EventArgs) Handles Max_button.Click
    Me.MaximizedBounds = Screen.GetWorkingArea(Me)
    If Me.WindowState = FormWindowState.Normal Then
        Me.WindowState = FormWindowState.Maximized
    Else
        Me.WindowState = FormWindowState.Normal
    End If
End Sub
Private Sub Main_page_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Userid = user.Text
    con.ConnectionString = "Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\HASEEF\source\repos\login_page\login_page\Database1.mdf;Integrated Security=True"
    con.Open()
    Fillhomescreen()
    add_btn_Click(sender, e)
End Sub
Private Sub Fillhomescreen()
    Dim sql_history = "SELECT * from allmovie where movie_id in(SELECT movie_id from history where history.user_id = '" + Userid + "' ) "
    Dim sql_actionmovies = "select * from allmovie,genre where genre.genre_name = 'Action' AND genre.movie_id = allmovie.movie_id"
    Dim sql_madeforyou = "select * from allmovie where rating >= '7.5' order by rating DESC"
    Dim sql_scific = "select * from allmovie,genre where genre.genre_name = 'Sci-fic' AND genre.movie_id = allmovie.movie_id"
    Dim sql_Romance = "select * from allmovie,genre where genre.genre_name = 'Romance' AND genre.movie_id = allmovie.movie_id"
    ReadFromDb(sql_history, pic_history)
    ReadFromDb(sql_actionmovies, Action_pnl)
    ReadFromDb(sql_madeforyou, made_for_you_panel)
    ReadFromDb(sql_scific, Scific_panel)
```

```
    ReadFromDb(sql_Romance, Romance_panel)
End Sub
Private Sub ShowPanel(ByRef pnl)
    HideAllPanel()
    pnl.visible = True
End Sub
Private Sub HideAllPanel()
    collection_panel.Visible = False
    Movie_info_pnl.Hide()
    trending.Hide()
    userhistory.Hide()
    favourite.Hide()
    discover.Hide()
    user_setting.Hide()
End Sub
Public Sub ReadFromDb(ByVal sql, ByRef pnl)
    Dim cmd As New SqlCommand
    Dim dr As SqlDataReader
    cmd.Connection = con
    cmd.CommandText = sql
    dr = cmd.ExecuteReader()
    Dim x As Integer = 5
    Dim y As Integer = 3
    Dim elem = 0
    While dr.Read
        elem += 1
        Dim pic As New PictureBox
        pic.Size = New Point(150, 210)
        pic.Name = "pic_box-" & (count + 1)
        pic.Location = New Point(x, y)
        pic.SizeMode = PictureBoxSizeMode.Zoom
        Dim movie_id = dr("movie_id")
        Dim movie_pic = dr("display_picture")
        Dim movie_name = dr("movie_name")
        Dim movie_rating = dr("rating")
        Dim movie_released = dr("released")
        Dim movie_desc = dr("movie_description")
```

```
        Dim movie_file = dr("movie_file")
        pic.Image = Image.FromFile(movie_pic)
        AddHandler pic.Click, AddressOf Me.ShowDetails
        Dim val() As String = {movie_pic, movie_name, movie_released, movie_rating, movie_desc,
movie_file, movie_id}
        pic.Tag = val
        pnl.Controls.Add(pic)
        x += 195
        count += 1
        If elem = 5 Then
            x = 5
            y = 250
        End If
    End While
    dr.Close()
  End Sub
  Public Sub GetOtherDetails(ByVal movie_id)
    clickedMovie = movie_id
    Dim sql_genre = "select genre_name from genre where movie_id = '" + movie_id + "'"
    Dim sql_director = "select director_name from director where movie_id = '" + movie_id + "'"
    Dim sql_actor = "select star_name from actor where movie_id = '" + movie_id + "'"
    genre = ""
    director = ""
    stars = ""
    Storeinvar(sql_genre, "genre_name", genre)
    Storeinvar(sql_director, "director_name", director)
    Storeinvar(sql_actor, "star_name", stars)
  End Sub
  Public Sub Storeinvar(ByVal sql, ByVal col_name, ByRef values)
    Dim command As New SqlCommand
    Dim dr As SqlDataReader
    command.Connection = con
    command.CommandText = sql
    dr = command.ExecuteReader()
    While dr.Read
      If values = String.Empty Then
        values = dr(col_name)
```

```
            Else
                values += " , " + dr(col_name)
            End If
        End While
        dr.Close()
    End Sub
    Public Sub ShowDetails(sender As Object, e As EventArgs)
        Dim picbox As PictureBox = CType(sender, PictureBox)
        Dim item() As String = picbox.Tag
        Show_box(item)
    End Sub
    Public Sub Show_box(ByRef details)
        picture.Image = Image.FromFile(details(0))
        mve_name.Text = details(1)
        rel_yr.Text = details(2)
        rat.Text = details(3)
        mve_details.Text = details(4)
        file = details(5)
        GetOtherDetails(details(6))
        mve_genre.Text = genre
        mve_directors.Text = director
        mve_stars.Text = stars
        Movie_info_pnl.Show()
        CheckCollection()
    End Sub
    Public Sub play_mve_Click(sender As Object, e As EventArgs) Handles play_mve.Click
        move_player.Show()
        Dim st = file
        move_player.pa.playlist.add("file:///" & st)
        Dim query = "insert into history(user_id,movie_id) values(@uid,@mid)"
        Savetohistory(query)
        updateTimeAndDate()
        pic_history.Controls.Clear()
        Dim sql_history = "SELECT * from allmovie where movie_id in(SELECT movie_id from history
where history.user_id = '" + Userid + "'  ) "
        ReadFromDb(sql_history, pic_history)
        Me.WindowState = FormWindowState.Minimized
```

```
    End Sub
    Private Sub updateTimeAndDate()
        Dim dt = Today.ToShortDateString

Dim update = "Update history set watched_time = 1,watched  = '12-03-2020' where user_id  = '" + Userid +
"' and movie_id = '" + clickedMovie + "'"
        Dim comm As New SqlCommand With {
            .Connection = con,
            .CommandText = update     }
        comm.ExecuteNonQuery()
    End Sub
    Private Sub Savetohistory(ByVal query)
        Dim command As New SqlCommand
        Dim da As New SqlDataAdapter
        command.Connection = con
        command.CommandText = query
        da.InsertCommand = command
        Dim param_mid As New SqlParameter("@uid", SqlDbType.VarChar, 10)
        param_mid.Value = Userid
        Dim param_name As New SqlParameter("@mid", SqlDbType.VarChar, 50)
        param_name.Value = clickedMovie
        command.Parameters.Add(param_mid)
        command.Parameters.Add(param_name)
        da.InsertCommand = command
        Try
            da.InsertCommand.ExecuteNonQuery()
        Catch ex As Exception
            Dim q = "update history set watched_time = watched_time + 1 where user_id = '" + Userid + "' and
movie_id = '" + clickedMovie + "'"
            command.CommandText = q
            command.ExecuteNonQuery()
        End Try
    End Sub
    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles info_btn.Click
        If Movie_info_pnl.Dock = DockStyle.Right Then
            Movie_info_pnl.Dock = DockStyle.Fill
            Home.Hide()
            info_btn.Text = "Show Less"
```

```
            more_movie_info_pnl.Visible = True
        Else
            Movie_info_pnl.Dock = DockStyle.Right
            Home.Show()
            info_btn.Text = "Show More"
            more_movie_info_pnl.Visible = False
        End If
    End Sub
    Private Sub CheckCollection()
        Dim cmd As New SqlCommand
        Dim dr As SqlDataReader
        cmd.Connection = con
        cmd.CommandText = "select * from favourite_movies where  user_id = '" + Userid + "'  AND movie_id ='" + clickedMovie + "'"
        dr = cmd.ExecuteReader()
        If dr.Read Then
            PictureBox3.Image = Image.FromFile("C:\Users\HASEEF\source\repos\login_page\login_page\Resources\icons8-checked-48.png")
            PictureBox3.Tag = 1
        Else
            PictureBox3.Image = Image.FromFile("C:\Users\HASEEF\source\repos\login_page\login_page\Resources\icons8-plus-48.png")
            PictureBox3.Tag = 0
        End If
        dr.Close()
    End Sub
    Private Sub close_pnl_Click_1(sender As Object, e As EventArgs) Handles close_pnl.Click
        Movie_info_pnl.Visible = False
        Home.Show()
        CheckCollection()
    End Sub
    Private Sub Add(ByRef pnl, ByRef Addhere)
        pnl.TopLevel = False
        pnl.TopMost = True
        pnl.FormBorderStyle = FormBorderStyle.None
        pnl.Dock = DockStyle.Fill
```

```
        Addhere.Controls.Add(pnl)
        Addhere.AutoScroll = True
        pnl.BringToFront()
        pnl.Show()
    End Sub
    Private Sub add_btn_Click(sender As Object, e As EventArgs) Handles Home_btn.Click
        ShowPanel(collection_panel)
        ClickedButton(Home_btn)
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles trending_btn.Click
        HideAllPanel()
        Add(trending, Home)
        ClickedButton(trending_btn)
    End Sub
    Private Sub Button4_Click_1(sender As Object, e As EventArgs) Handles History_btn.Click
        HideAllPanel()
        Add(userhistory, Home)
        ' userhistory.CheckIfUserStore(userhistory.history)
        ClickedButton(History_btn)
    End Sub
    Private Sub PictureBox3_Click(sender As Object, e As EventArgs) Handles PictureBox3.Click
        If PictureBox3.Tag = 0 Then
            Timer1.Start()
            PictureBox3.Image =
Image.FromFile("C:\Users\HASEEF\source\repos\login_page\login_page\Resources\icons8-plus.gif")
            Dim query = "insert into favourite_movies(user_id,movie_id) values(@uid,@mid)"
            Savetohistory(query)
            PictureBox3.Tag = 1
        Else
            MsgBox("Already Bookmarked")
        End If
    End Sub
    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        PictureBox3.Image =
Image.FromFile("C:\Users\HASEEF\source\repos\login_page\login_page\Resources\icons8-checked-
48.png")
        Timer1.Stop()
```

```
End Sub
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Collection_btn.Click
    HideAllPanel()
    favourite.favourite_Load(sender, e)
    Add(favourite, Home)
    ' userhistory.CheckIfUserStore(favourite.pic_fav)
    ClickedButton(Collection_btn)
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Discover_btn.Click
    HideAllPanel()
    Add(discover, Home)
    ClickedButton(Discover_btn)
End Sub
Private Sub Button1_Click_3(sender As Object, e As EventArgs) Handles More_btn.Click
    If Panel_morefunction.Visible = False Then
        Panel_morefunction.Visible = True
    Else
        Panel_morefunction.Visible = False
    End If
    ClickedButton(More_btn)
End Sub
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    HideAllPanel()
    Add(user_setting, Home)
End Sub
Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
    Me.Hide()
    login_page.Show()
    login_page.WindowState = FormWindowState.Normal
End Sub
Public Function Select_user()
    Dim q = "select * from valid_user where username = '" + Me.Userid + "'"
    Dim com As New SqlCommand With {
        .Connection = con,
        .CommandText = q}
    Dim dr = com.ExecuteReader
    Return dr
```

```
    End Function
    Public Sub Setuserdp()
        Userid = user.Text
        Dim dr = Select_user()
        While dr.read
            PictureBox2.Image = Image.FromFile(dr("profile_pic"))
        End While
        dr.close()
    End Sub
    'UI Design
    Private Sub ShowOnHover(ByRef button, ByVal img)
        button.Image = Image.FromFile(img)
    End Sub


    End Sub
    Private Sub Max_button_MouseLeave(sender As Object, e As EventArgs) Handles
Max_button.MouseLeave
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Green.png"
        ShowOnHover(Max_button, img)
    End Sub
    Private Sub ClickedButton(ByRef btn)
        StandardBtn()
        btn.backcolor = Color.FromArgb(11, 20, 150)
    End Sub
    Private Sub StandardBtn()
        Home_btn.BackColor = Color.FromArgb(11, 17, 37)
        trending_btn.BackColor = Color.FromArgb(11, 17, 37)
        Discover_btn.BackColor = Color.FromArgb(11, 17, 37)
        Collection_btn.BackColor = Color.FromArgb(11, 17, 37)
        History_btn.BackColor = Color.FromArgb(11, 17, 37)
        More_btn.BackColor = Color.FromArgb(11, 17, 37)
    End Sub
End Class
```

Admin module

```
Imports System.Data.SqlClient
Public Class admin_page
```

```
Public con As New SqlConnection
Dim dp, fileloc As String
Dim directors As New List(Of String)()
Dim genre As New List(Of String)()
Dim stars As New List(Of String)()
Dim val As String
Private Sub Admin_page_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ' Connects To The DataBase
    con.ConnectionString = "Data Source=  (LocalDB) \MSSQLLocalDB; AttachDbFilename=
C:\Users\HASEEF\source\repos\login_page\login_page\Database1.mdf;Integrated Security=True "
    con.Open()
End Sub
Private Sub ButtonExit_Click(sender As Object, e As EventArgs) Handles buttonExit.Click
    con.Close()
    Application.Exit()
End Sub
Private Sub Minimise_Click(sender As Object, e As EventArgs) Handles Minimise.Click
    Me.WindowState = FormWindowState.Minimized
End Sub

Private Sub SelectMovie_Click(sender As Object, e As EventArgs) Handles selectMovieFile.Click
    Dimfilter="AllMedia Files|*.mpg;*.mpeg; *.m3u;*.mp4;* .mov;*.3gp; *.midi;*.mkv;*.MPEG;*.MP4;
*.3GP; *.MIDI; *.MKV"
    fileloc = SelectFile(filter, display_vid_file)
End Sub
Private Sub SelectDp_Click(sender As Object, e As EventArgs) Handles SelectDp.Click
    Dim filter = "All Media Files|*.bmp;*.png;*.jpg;*.gif;*.tif"
    dp = SelectFile(filter, display_dp_file)
End Sub
Private Function SelectFile(ByVal string_filter, ByRef textbox)
    OpenFileDialog1.Filter = string_filter
    OpenFileDialog1.FilterIndex = 1
    OpenFileDialog1.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.Recent)
    If OpenFileDialog1.ShowDialog = DialogResult.OK Then
        val = OpenFileDialog1.FileName
    End If
    textbox.Text = val
```

```
        textbox.Visible = True
        Return val
    End Function
    Private Sub InsertNewMovie(sender As Object, e As EventArgs) Handles add_todatabase_movie.Click
        Try
            Dim movie_id As String = m_id.Text
            Dim name As String = m_name.Text
            Dim rating As String = Rating_1.Text
            Dim rel As String = released_year.Text
            Dim desc As String = m_desc.Text
            Dim dur As String = mve_dur.Text
            Dim command As New SqlCommand
            Dim query As String = "insert into allmovie ( movie_id , movie_name, rating, released,
movie_description, display_picture, movie_file,movie_duration) values (@mid, @mname, @ratings,
@released, @moviedesc, @dp,@mf,@mdur)"
            command.Connection = con
            command.CommandText = query
            Dim param_mid As New SqlParameter("@mid", SqlDbType.VarChar, 10)
            param_mid.Value = movie_id
            Dim param_name As New SqlParameter("@mname", SqlDbType.VarChar, 50)
            param_name.Value = name
            Dim param_ratings As New SqlParameter("@ratings", SqlDbType.Float)
            param_ratings.Value = rating
            Dim param_rel As New SqlParameter("@released", SqlDbType.Int, 4)
            param_rel.Value = rel
            Dim param_desc As New SqlParameter("@moviedesc", SqlDbType.VarChar, 1000)
            param_desc.Value = desc
            Dim param_dp As New SqlParameter("@dp", SqlDbType.VarChar, 150)
            param_dp.Value = dp
            Dim param_file As New SqlParameter("@mf", SqlDbType.VarChar, 150)
            param_file.Value = fileloc
            Dim param_duration As New SqlParameter("@mdur", SqlDbType.VarChar, 150)
            param_duration.Value = dur
            command.Parameters.Add(param_mid)
            command.Parameters.Add(param_name)
            command.Parameters.Add(param_ratings)
            command.Parameters.Add(param_rel)
```

```
        command.Parameters.Add(param_desc)
        command.Parameters.Add(param_dp)
        command.Parameters.Add(param_file)
        command.Parameters.Add(param_duration)
        Try
            Dim adapter As New SqlDataAdapter
            adapter.InsertCommand = command
            adapter.InsertCommand.ExecuteNonQuery()
            MoreDb(movie_id)
            result_panel.Visible = True
            result.Text = "Inserted Sucessfully"
        Catch ex As SqlException
            If ex.Number = 2627 Then
                result.Text = "Movie Already Exist ," & vbCrLf & " please try with different" & vbCrLf & "
Title"
                result_panel.Visible = True
            Else
                result.Text = "Somthing went wrong "
                result_panel.Visible = True
            End If
        End Try

    Catch ex As Exception
        result.Text = "Somthing went wrong "
        result_panel.Visible = True
    End Try
  End Sub
  Private Sub MoreDb(ByVal m_id)
    Dim query1 = "insert into genre(movie_id,genre_name) values(@mid,@name)"
    Dim query2 = "insert into director(movie_id,director_name) values(@mid,@name)"
    Dim query3 = "insert into actor(movie_id,star_name) values(@mid,@name)"
    For Each value As String In genre
        InsertIntoDb(query1, m_id, value)
    Next
  Private Sub InsertIntoDb(ByVal query, ByVal movie_id, ByVal value)
    Dim command As New SqlCommand
    Dim da As New SqlDataAdapter
```

```
        command.Connection = con
        command.CommandText = query
        da.InsertCommand = command
        Dim param_mid As New SqlParameter("@mid", SqlDbType.VarChar, 10)
        param_mid.Value = movie_id
        Dim param_name As New SqlParameter("@name", SqlDbType.VarChar, 50)
        param_name.Value = value
        command.Parameters.Add(param_mid)
        command.Parameters.Add(param_name)
        da.InsertCommand = command
        da.InsertCommand.ExecuteNonQuery()
    End Sub
    Public Sub Showpanel(ByRef pnl)
        Hidepanel()
        pnl.visible = True
        pnl.dock = DockStyle.Fill
    End Sub
    Private Sub Hidepanel()
        Panel_add_movie.Visible = False
        Panel_director.Dock = DockStyle.None
        DeleteMovie_panel.Visible = False
        DeleteMovie_panel.Dock = DockStyle.None
        stats.Hide()
        stats2.Hide()
        stats3.Hide()
        stats_details_user.Hide()
    End Sub
    Private Sub add_btn_Click(sender As Object, e As EventArgs) Handles add_btn.Click
        Showpanel(Panel_add_movie)
    End Sub
    Private Sub ShowOnHover(ByRef button, ByVal img)
        button.Image = Image.FromFile(img)
    End Sub

Private Sub Exit_button_MouseEnter(sender As Object, e As EventArgs) Handles Exit_button.MouseEnter
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Exit.png"
        ShowOnHover(Exit_button, img)
    End Sub
```

```
    Private Sub Exit_button_MouseLeave(sender As Object, e As EventArgs) Handles Exit_button.MouseLeave
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Red.png"
        ShowOnHover(Exit_button, img)
    End Sub
    Private Sub Mini_button_MouseEnter(sender As Object, e As EventArgs) Handles Mini_button.MouseEnter
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Minimise.png"
        ShowOnHover(Mini_button, img)
    End Sub
    Private Sub Mini_button_MouseLeave(sender As Object, e As EventArgs) Handles Mini_button.MouseLeave
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Yellow.png"
        ShowOnHover(Mini_button, img)
    End Sub
    Private Sub Max_button_MouseEnter(sender As Object, e As EventArgs) Handles Max_button.MouseEnter
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Maximise.png"
        ShowOnHover(Max_button, img)
    End Sub
    Private Sub Max_button_MouseLeave(sender As Object, e As EventArgs) Handles Max_button.MouseLeave
        Dim img = "C:\Users\HASEEF\source\repos\login_page\login_page\Resources\Green.png"
        ShowOnHover(Max_button, img)
    End Sub
    Private Sub Exit_button_Click(sender As Object, e As EventArgs) Handles Exit_button.Click
        Application.Exit()
    End Sub
    Private Sub Mini_button_Click(sender As Object, e As EventArgs) Handles Mini_button.Click
        Me.WindowState = FormWindowState.Minimized
    End Sub
    Private Sub Max_button_Click(sender As Object, e As EventArgs) Handles Max_button.Click
        Me.MaximizedBounds = Screen.GetWorkingArea(Me)
        If Me.WindowState = FormWindowState.Normal Then
            Me.WindowState = FormWindowState.Maximized
        Else
            Me.WindowState = FormWindowState.Normal
```

```
    End If
End Sub
Dim x1 As Integer = 0
Dim x2 As Integer = 0
Dim x3 As Integer = 0
Dim item As Integer = 0
Private Sub CreateVisibleValues(ByRef box, ByRef array, ByRef parent_panel, ByVal x, ByVal name)
    Dim txtbx As New TextBox With {
        .Name = name + item.ToString,
        .Text = box.text,
        .Location = New Point(x, 0),
        .ReadOnly = True   }
    array.add(txtbx.Text)
    parent_panel.Controls.Add(txtbx)
    box.Text = ""
    item += 1
End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
    CreateVisibleValues(m_genre, genre, Panel_genre, x1, "genre_")
    x1 += 120
End Sub
Private Sub Button3_Click_1(sender As Object, e As EventArgs) Handles Button3.Click
    CreateVisibleValues(m_director, directors, Panel_director, x2, "dir_")
    x2 += 120
End Sub
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    CreateVisibleValues(m_cast, stars, Panel_star, x3, "cast_")
    x3 += 120
End Sub
Private Sub RemoveControls(ByVal panel_name As Panel)
    panel_name.Controls.Clear()
End Sub

Private Sub ClearFromForm(sender As Object, e As EventArgs) Handles Button4.Click
    ClearTextBox(Me)
    m_desc.Text = ""
```

```
        RemoveControls(Panel_director)
        RemoveControls(Panel_genre)
        RemoveControls(Panel_star)
        x1 = 0
        x2 = 0
        x3 = 0
        result_panel.Visible = False
        directors.Clear()
        genre.Clear()
        stars.Clear()
    End Sub
    Public Sub ClearTextBox(ByVal root As Control)
        For Each ctrl As Control In root.Controls
            ClearTextBox(ctrl)
            If TypeOf ctrl Is TextBox Then
                CType(ctrl, TextBox).Text = String.Empty
            End If
        Next ctrl
    End Sub
    Private Sub ShowDelPanel(sender As Object, e As EventArgs) Handles Del_Movie.Click
        Showpanel(DeleteMovie_panel)
    End Sub
    Private Sub SearchButton_Click(sender As Object, e As EventArgs) Handles SearchButton.Click
        Dim query As String = "select * from allmovie where movie_id = '" + MovietoDelete.Text + "'"
        Dim sql_genre = "select genre_name from genre where movie_id = '" + MovietoDelete.Text + "'"
        Dim sql_director = "select director_name from director where movie_id = '" + MovietoDelete.Text +
"'"
        Dim command As New SqlCommand With {
            .Connection = con,
            .CommandText = query  }
        Dim dr As SqlDataReader
        dr = command.ExecuteReader()
        If dr.Read Then
            Movie_detail_panel.Visible = True
            Dim movie_pic = dr("display_picture")
            movie_name_todelete.Text = dr("movie_name")
            movie_ratings_todelete.Text = dr("rating")
```

```
            movie_year_todelete.Text = dr("released")
            movie_pic_todelete.Image = Image.FromFile(movie_pic)
            dr.Close()
            Search_otherDb(sql_director, movie_dir_todelete, "director_name")
            Search_otherDb(sql_genre, movie_gen_todelete, "genre_name")
        Else
            res_invalid.Visible = True
        End If
        dr.Close()
    End Sub
    Private Sub Search_otherDb(ByVal sql, ByRef control, ByVal col_name)
        Dim command As New SqlCommand
        Dim dr As SqlDataReader
        command.Connection = con
        command.CommandText = sql
        dr = command.ExecuteReader()
        If dr.Read Then
            control.text = dr(col_name)
        End If
        dr.Close()
    End Sub
    Dim success = 0
    Private Sub DeleteMovie_Click(sender As Object, e As EventArgs) Handles DeleteMovie.Click
        If MessageBox.Show("Do you really want to delete this record?", "Delete",
MessageBoxButtons.YesNo, MessageBoxIcon.Warning) =
            Windows.Forms.DialogResult.Yes Then
            DeleteMovieFromDb(CreateSqlString("allmovie", MovietoDelete.Text))
            DeleteMovieFromDb(CreateSqlString("director", MovietoDelete.Text))
            DeleteMovieFromDb(CreateSqlString("genre", MovietoDelete.Text))
            DeleteMovieFromDb(CreateSqlString("actor", MovietoDelete.Text))
            DeleteMovieFromDb(CreateSqlString("history", MovietoDelete.Text))
            DeleteMovieFromDb(CreateSqlString("favourite_movies", MovietoDelete.Text))
            If success > 0 Then
                MsgBox("Record Successfully Deleted")
            Else
                MsgBox("Record is not deleted")
            End If
```

```
            Movie_detail_panel.Visible = False
        End If
    End Sub

    Private Function CreateSqlString(ByVal table, ByVal movie_id)
        Dim query = "Delete From " + table + " Where movie_id = '" + movie_id + "'"
        Return query
    End Function
    Private Sub DeleteMovieFromDb(ByVal query)
        Dim cmd As New SqlCommand With {
            .Connection = con,
            .CommandText = query
        }
        Dim i As Integer = cmd.ExecuteNonQuery()
        If (i > 0) Then
            success += 1
        End If
    End Sub
Public Sub InsertInPanel(ByRef Pnl)
        Pnl.TopLevel = False
        Pnl.TopMost = True
        Showpanel(Pnl)
        Panel3.Controls.Add(Pnl)
    End Sub
    Private Sub LogOut_Click(sender As Object, e As EventArgs) Handles LogOut.Click
        login_page.Show()
        Me.Hide()
    End Sub
    Public Sub Fill_Page(ByVal query, ByVal Seriesname, ByRef Chartname)
        Dim cmd As New SqlCommand With {
          .Connection = con,
          .CommandText = query}
        Dim dr As SqlDataReader
        dr = cmd.ExecuteReader
        Chartname.ChartAreas(0).AxisX.Interval = 1
        While dr.Read
            Chartname.Series(Seriesname).Points.AddXY(dr.GetValue(0), dr.GetValue(1))
```

```
        End While
        dr.Close()
    End Sub
End Class
```

Report Module

```
Imports System.Data.SqlClient
Public Class stats
    Dim con
    Public totalwatched_Movie
    Private Sub stats_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        con = admin_page.con
        FillHomeStats()
    End Sub
    Private Sub FillHomeStats()
        Dim Query = "select count(username) from valid_user"
        user_count.Text = CountTotalInDb(Query)
        Dim Query2 = "select count(movie_id) from allmovie"
        Movie_count.Text = CountTotalInDb(Query2)
        GetMovieDetails()
        GetGenreWatched()
        totalwatched_Movie = CountTotalInDb("select  count(distinct movie_id) from history")
    End Sub
    Public Function CountTotalInDb(ByVal query)
        Dim cmd As New SqlCommand With {
            .Connection = con,
            .CommandText = query
        }
        Dim ans = cmd.ExecuteScalar()
        Return ans
    End Function
    Private Sub GetMovieDetails()
        Dim query = "select top 5 count(movie_id),movie_id  from history  group by movie_id ORDER BY
COUNT(movie_id) DESC "
        Dim cmd As New SqlCommand With {
            .Connection = con,
            .CommandText = query  }
```

```
        Dim dr As SqlDataReader
        dr = cmd.ExecuteReader
        Dim a = 0
        While dr.Read
            If a = 0 Then
                TopMovie.Text = dr.GetString(1)
                a = 1
            End If
            Chart1.Series("Movies").Points.AddXY(dr.GetString(1), dr.GetInt32(0))
        End While
        dr.Close()
        GetMovie_Pic(TopMovie.Text)
    End Sub
    Private Sub GetMovie_Pic(ByVal movie_id)
        Dim cmd As New SqlCommand With {
            .Connection = con,
            .CommandText = "select display_picture,movie_name from allmovie where movie_id = '" + movie_id + "'"}
        Dim dr2 As SqlDataReader
        dr2 = cmd.ExecuteReader
        If dr2.Read Then
            Dim img = dr2.GetValue(0)
            Top_MoviePic.Image = Image.FromFile(img)
            TopMovie.Text = dr2.GetValue(1)
        End If
        dr2.Close()
    End Sub
    Private Sub GetGenreWatched()
        Dim query = "Select genre_name,count(genre_name) from genre group by genre_name"
        admin_page.Fill_Page(query, "Genre", TotalMovie_Gen)
    End Sub
    Private Sub Guna2CircleButton2_Click(sender As Object, e As EventArgs) Handles Guna2CircleButton2.Click
        admin_page.InsertInPanel(stats2)
    End Sub
    Private Sub Guna2CircleButton3_Click(sender As Object, e As EventArgs) Handles Guna2CircleButton3.Click
```

```
        admin_page.InsertInPanel(stats3)
    End Sub
    Private Sub Guna2CircleButton1_Click(sender As Object, e As EventArgs) Handles Guna2CircleButton1.Click
        admin_page.InsertInPanel(stats_details_user)
    End Sub
End Class
```

Movie Recommendation Module

```
Public Class discover

    Dim cl = 0
    Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles TextBox1.Click
        If cl = 0 Then
            TextBox1.Text = ""
        End If
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim searchquery = "select * from allmovie where movie_name  LIKE '%" + TextBox1.Text + "%'"
        Me.PanelMovie.Controls.Clear()
        main_page.ReadFromDb(searchquery, Me.PanelMovie)
    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        fill("Action")
    End Sub
    Private Sub fill(ByVal gen)
        Dim searchquery = "select * from allmovie,genre where genre.genre_name = '" + gen + "' AND genre.movie_id = allmovie.movie_id"
        Me.PanelMovie.Controls.Clear()
        main_page.ReadFromDb(searchquery, Me.PanelMovie)
    End Sub
    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        fill("Comedy")
    End Sub
    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        fill("Romance")
    End Sub
```

```
    Private Sub Button5_Click_1(sender As Object, e As EventArgs) Handles Button5.Click
        fill("Sci-fic")
    End Sub
    Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
        fill("Thriller")
    End Sub
    Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
        fill("Mystery")
    End Sub
End Class

Imports System.Data.SqlClient
Public Class user_setting
    Dim con As New SqlConnection
    Dim pass_auth = False
    Private pass
    Private Sub LoadPage(sender As Object, e As EventArgs) Handles MyBase.Load
        con.ConnectionString = "Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\HASEEF\source\repos\login_page\login_page\Database1.mdf;Integrated Security=True"
        con.Open()
        Res_panel.BringToFront()
        GetUserDetail()
    End Sub
    Private Sub ChangeDp_Click(sender As Object, e As EventArgs) Handles ChangeDp.Click
        Dim changeimage = " "
        OpenFileDialog1.Filter = "All Media Files|*.bmp;*.png;*.jpg;*.gif;*.tif"
        OpenFileDialog1.FilterIndex = 1
        OpenFileDialog1.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.Recent)
        If OpenFileDialog1.ShowDialog = DialogResult.OK Then
            changeimage = OpenFileDialog1.FileName
        Else
            Exit Sub
        End If
        Dim query = "update valid_user set profile_pic = '" + changeimage + "' where username = '" + main_page.Userid + "'"
        Dim cm As New SqlCommand With {
            .Connection = con,
```

```
            .CommandText = query}
        If cm.ExecuteNonQuery() Then
            Res_panel.Visible = True
            Resultlab.Text = "You Have Changed Sucessfully"
        End If
        GetUserDetail()
        main_page.Setuserdp()
    End Sub
    Public Sub GetUserDetail()
        Dim dr = main_page.Select_user()
        While dr.Read
            full_name.Text = dr("fullname")
            usr_name.Text = dr("username")
            usr_email.Text = dr("mail")
            usr_country.Text = dr("country")
            pass = dr("passwd")
            usr_phno.Text = dr("phno")
            profile_picture.Image = Image.FromFile(dr("profile_pic"))
        End While
        dr.Close()
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        ShowSpecificPanel(Newpass_panel)
    End Sub
    Private Sub ValidateUpdate(ByRef value, ByVal updatehere)
        Dim query = "update valid_user set " + updatehere + " = '" + value.Text + "' where username = '" +
main_page.Userid + "'"
        If UpdateUserDetail(query) Then
            Res_panel.Visible = True
            Res_panel.BringToFront()
            Resultlab.Text = "You Have Changed Sucessfully"
            value.Text = ""
            HideAllPanels()
            GetUserDetail()
        Else
            Res_panel.Visible = True
            Resultlab.Text = "Somthing Went Wrong ." + vbCrLf + " Please Try Again!"
```

```
            value.Text = ""
        End If
    End Sub
    Private Sub ChangePassword_Click(sender As Object, e As EventArgs) Handles ChangePassword.Click
        Authenticate_pass()
        If pass_auth Then
            If newp.Text = r_newp.Text Then
                ValidateUpdate(newp, "passwd")
                GetUserDetail()
            Else
                Res_panel.Visible = True
                Resultlab.Text = "Password Mis Match "
            End If
        Else
            Res_panel.Visible = True
            Resultlab.Text = "Old Password Do Not " + vbCrLf + "Match With Our Database"
        End If
    End Sub
    Private Function UpdateUserDetail(ByVal query)
        Dim cm As New SqlCommand With {
            .Connection = con,
            .CommandText = query}
        Dim res = cm.ExecuteNonQuery()
        Return res
    End Function
    Private Sub Authenticate_pass()
        If pass = old.Text Then
            pass_auth = True
        End If
    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Res_panel.Visible = False
    End Sub
    Private Sub ShowEmail_panel_Click(sender As Object, e As EventArgs) Handles
ShowEmail_panel.Click
        ShowSpecificPanel(newEmail_panel)
    End Sub
```

```
    Private Sub ShowSpecificPanel(ByRef panel)
        If panel.visible = False Then
            HideAllPanels()
            panel.visible = True
        Else
            panel.visible = False
        End If
    End Sub
    Private Sub HideAllPanels()
        newEmail_panel.Visible = False
        Newcount_panel.Visible = False
        Newpass_panel.Visible = False
        NewPhno_panel.Visible = False
    End Sub
    Private Sub ShowPh_panel_Click(sender As Object, e As EventArgs) Handles ShowPh_panel.Click
        ShowSpecificPanel(NewPhno_panel)
    End Sub
    Private Sub ShowCount_panel_Click(sender As Object, e As EventArgs) Handles
ShowCount_panel.Click
        ShowSpecificPanel(Newcount_panel)
    End Sub
    Private Sub ChangeEmail_Click(sender As Object, e As EventArgs) Handles ChangeEmail.Click
        ValidateUpdate(New_Email, "mail")
    End Sub
    Private Sub ChangePhoneno_Click(sender As Object, e As EventArgs) Handles ChangePhoneno.Click
        ValidateUpdate(New_phone, "phno")
    End Sub
    Private Sub ChangeCountry_Click(sender As Object, e As EventArgs) Handles ChangeCountry.Click
        ValidateUpdate(New_Country, "Country")
    End Sub
End Class
```

4.SYSTEM CONFIGURATION

4.1 HARDWARE REQUIREMENTS

TECHONOLOGY	VB.NET
RAM	3.00 GB AND ABOVE
HARD DISK	500 GB AND ABOVE
PROCESSOR	INTEL i3 AND ABOVE
PROCESSOR SPEED	2.06 GHz

4.2 SOFTWARE REQUIREMENTS

OPERATING SYSTEM	WINDOWS 10
PROGRAMMING LANGAUGE	VB.NET 2019
DATABASE	SQL SERVER 2012
TOOL(S)	GOOGLE GANTTER
DOCUMENTATION	LIBRE OFFICE WRITER

5. DETAILS OF SOFTWARE

5.1 OVERVIEW OF FRONT END

MICROSOFT VISUAL STUDIO 2019

Microsoft Visual Studio 2019 is an integrated development environment (IDE) from Microsoft, ft 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 languagespecific 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.

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 is very simple, particularly as to the executable code.

- VB is not only a language but primarily an integrated, interactive development environment ("1DE").
- The VB-IDE has been highly optimized to support rapid application development ("RAD"). It is particularly easy to develop graphical user interfaces and 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.
- VB is a component integration language which is attuned to Microsoft's Component Object Model ("COM").
- 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 Visual languages, so there is very little that can't be done in Visual Basic that can be done in other languages (such as C#).

DISADVANTAGES

• Visual basic is a proprietary programming language written by Microsoft, so programs written in Visual basic cannot, easily, be transferred to other operating systems

• There are some, fairly minor disadvantages compared with C. C has better declaration of arrays - its possible to initialize an array of structures in C.at declaration time; this is impossible in VB.

5.2 OVERVIEW OF BACK-END

MICROSOFT SOL SERVER 2012:

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

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. The protocol layer implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoft-defined format, called Tabular Data Stream (TDS). TDS is an application layer protocol, used to transfer data between a database server and a client. A Relational Database 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.

5.3 ABOUT THE PLATFORM

Windows is a series of Operating Systems developed by Microsoft. Each version of Windows includes a Graphical User Interface, with a desktop that allows users to view files and folders in Windows. For 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). The current version, Windows 7, was released in 2009. The first business-oriented version of Windows, called Windows NT 3.1, was in 1993.

Therefore, it can be installed on multiple brands of hardware, such as Dell, HP, and Sony computers, as well as home-built PCs. Windows 7 also includes several touchscreen features, that allow the operating system to run on certain tablets and computers with touchscreen displays. Microsoft's mobile operating system,

Windows Phone, is designed specifically for smartphones and runs on several brands of phones, including HTC, Nokia, and Samsung.

.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 CFR together constitute the .NET Framework. FCL provides user interface, data access, database connectivity, cryptography, web application development, numeric

algorithms, and network communications. Programmers produce software by combining their source code with .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 largely for .NET software called Visual Studio. .NET Framework began as proprietary software, although the firm worked to standardize the software stack almost immediately, even before its first release. Despite the standardization efforts, developers, mainly those in the free and open-source software communities, expressed their unease with the selected terms and the prospects of any free and open-source implementation, especially regarding software patents. Since 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.

6. 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 lifecycle. 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.

Software development life cycle

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. 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're not done till the tests pass. 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 for this 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. Perform 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 testing - 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.

Compatibility testing - Testing how well software preforms 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.

During The Progress Of The Application , Unit Testing ,Regression Testing , Performance Testing were used.

7. CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The Movie Streaming App “Binger“ Is A platform which provides a quality of movie entertainment at home with less cost and less resources. It has Highly Interactive User Interface which makes easy to access and use the application. The user can watch any movie anytime, anywhere just by using their username and password. This Application also Recommend new movies to watch from the Genre, Title of Movie, or Other Detail Regarding the Movie. This Is an OTT Platform Application Which Can Be Used By Any User who Has Minimum or Basic Knowledge on Computer.

FUTURE ENHANCEMENT

The Movie Streaming App Provide a Quality Entertainment, But It can be Improved By Using complex machine learning algorithm by,

- Finding the Most Genre Watched Pattern and Provide movies asper the result to Each user
- Suggesting New Movies and Non-Repetitive Movies.
- Improve the User Interface and Experience of The Program
- Implement Machine Learning Algorithm To Recommend Movies

8.BIBLOGRAPHY

SL.NO	AUTHOR NAME	WEBSITE/ TITLE OF BOOK	PLACE	PUBLISHER
1		https://www.geeksforgeeks.org/	Uttar Pradesh	GeeksForGeeks
2	Tutorials point	https://www.tutorialspoint.com/	Hyderabad	Tutorials Point
3	Sonoo Jaiswal	https://www.javatpoint.com/	Delhi	JavaTpoint

9.APPENDICES A-TABLE STRUCTURE

valid_users

NAME	DATATYPE	ALLOW NULL
username	nchar(50)	Not null
fullname	nchar(50)	Not null
password	nchar(30)	Not null
phone_number	int	Not null
mail_id	nchar(50)	Not null
profile_pic	nchar(150)	Not null
country	nchar(50)	Not null

all_movies

NAME	DATATYPE	ALLOW NULL
movie_id	nchar(10)	Not null
movie_name	nchar(50)	Not null
ratings	Float	Not null
released_year	Int(4)	Not null
movie_description	nchar(150)	Not null
movie_picture	nchar(150)	Not null
movie_file	nchar(50)	Not null
movie_duration	Float	Not null

User_history

NAME	DATATYPE	ALLOW NULL
movie_id	nchar(10)	Not null
username	nchar(50)	Not null

User_Favourite

NAME	DATATYPE	ALLOW NULL
movie_id	nchar(10)	Not null
username	nchar(50)	Not null

Genre

NAME	DATATYPE	ALLOW NULL
movie_id	nchar(10)	Not null
genre_name	nchar(50)	Not null

Director

NAME	DATATYPE	ALLOW NULL
movie_id	nchar(10)	Not null
Director_name	nchar(50)	Not null

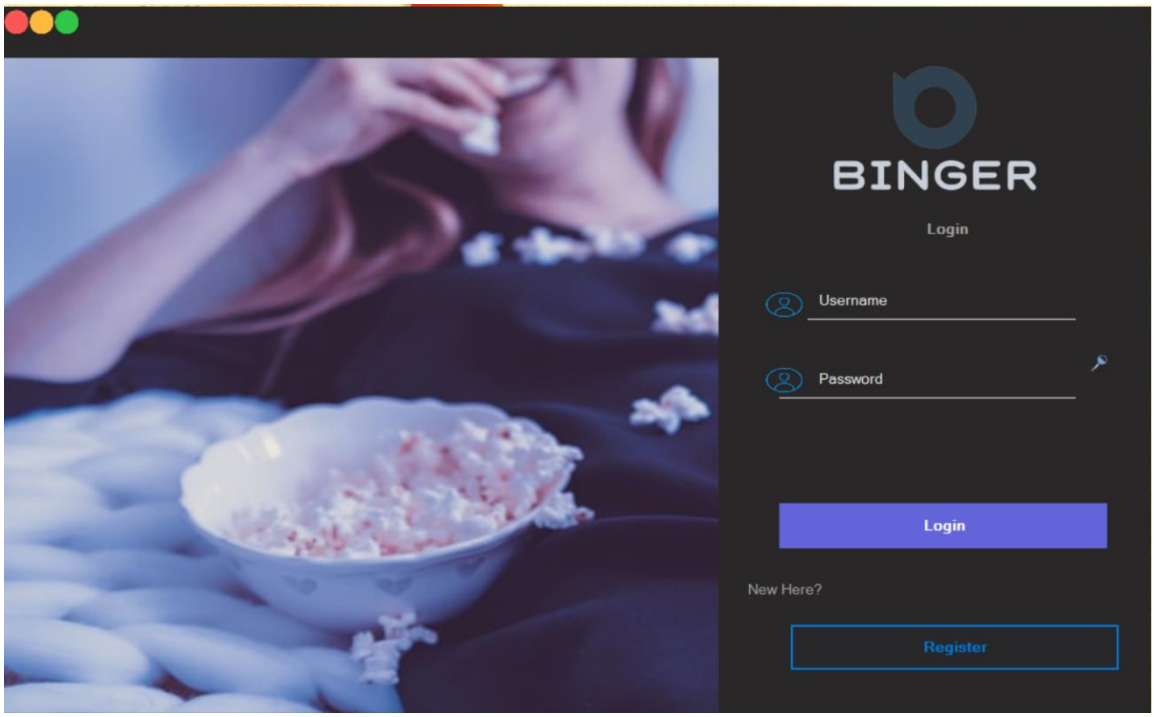
Star_cast

NAME	DATATYPE	ALLOW NULL
movie_id	nchar(10)	Not null
actor_name	nchar(50)	Not null

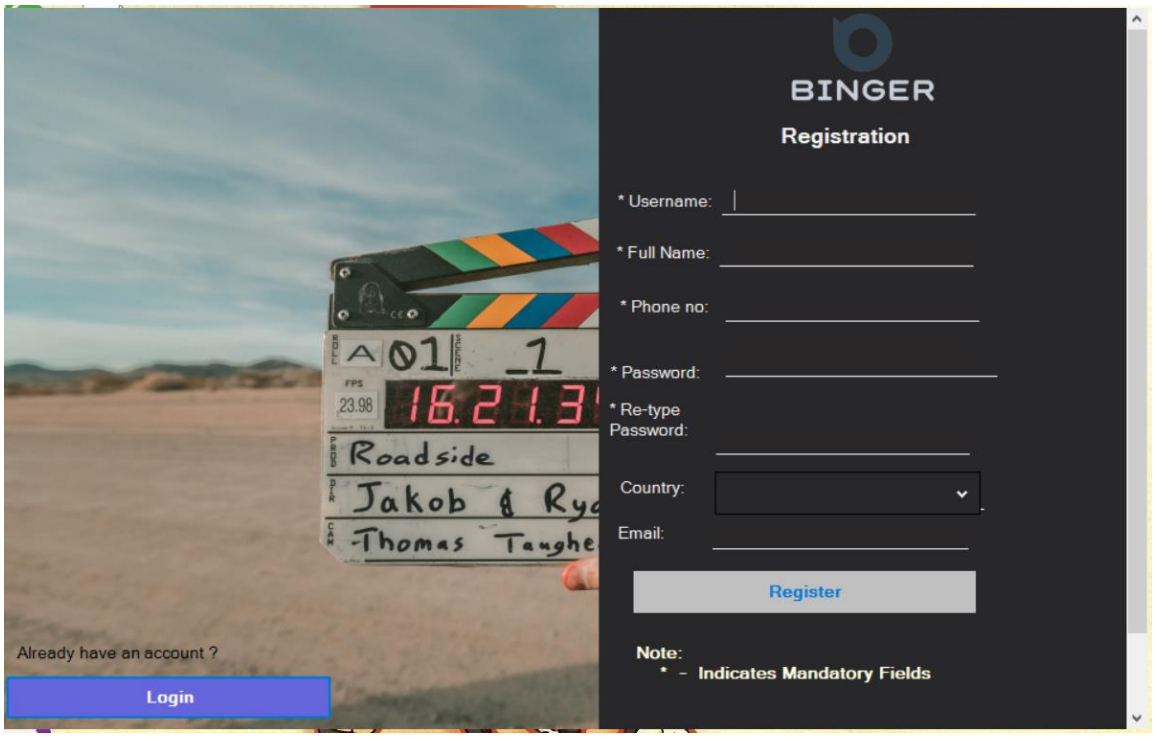
10.APPENDICES B-SCREENSHOTS

User Validation Module:

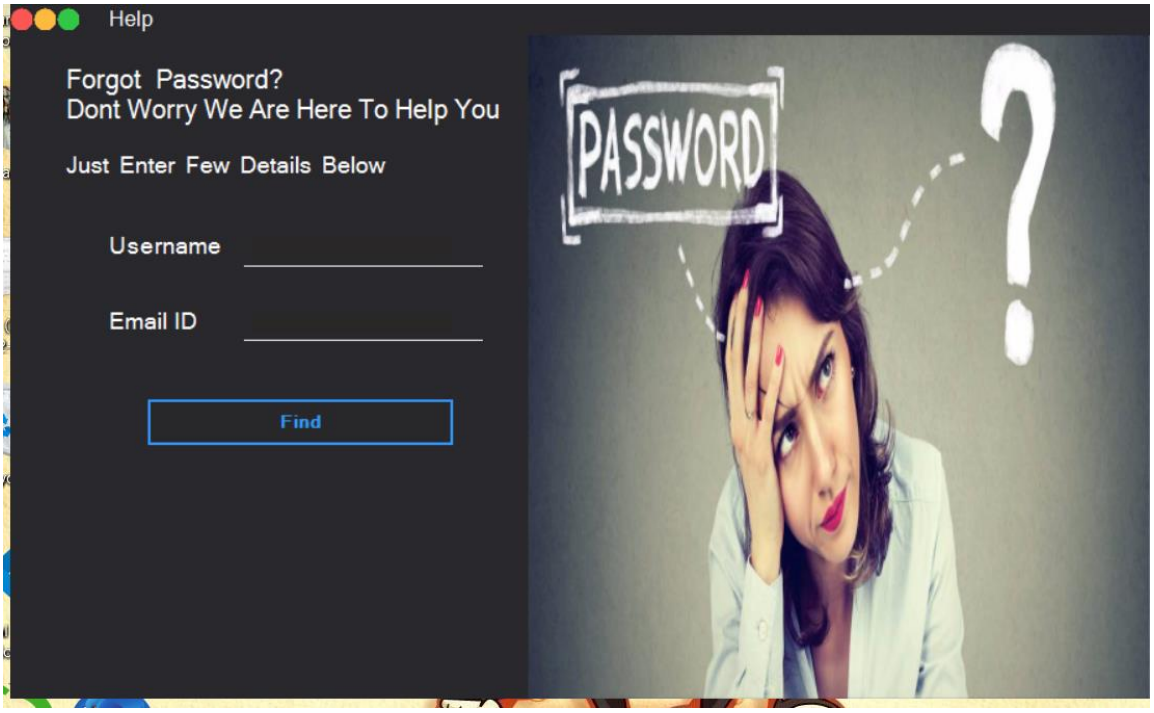
- **Login Page**



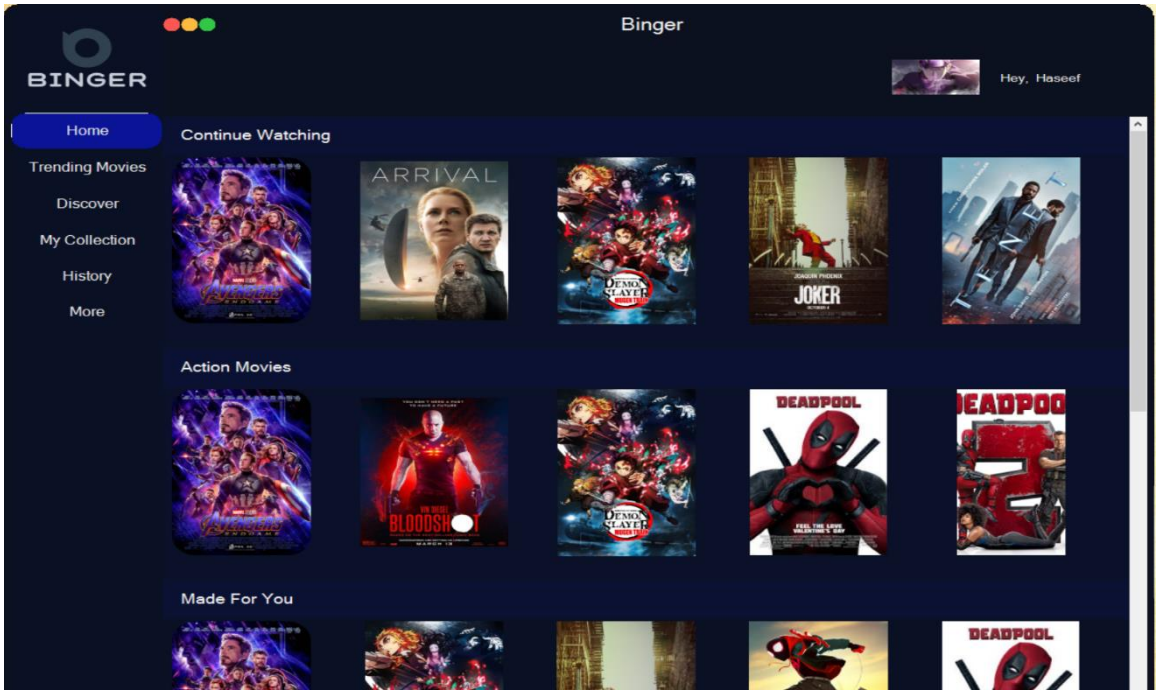
- **Registration Page**



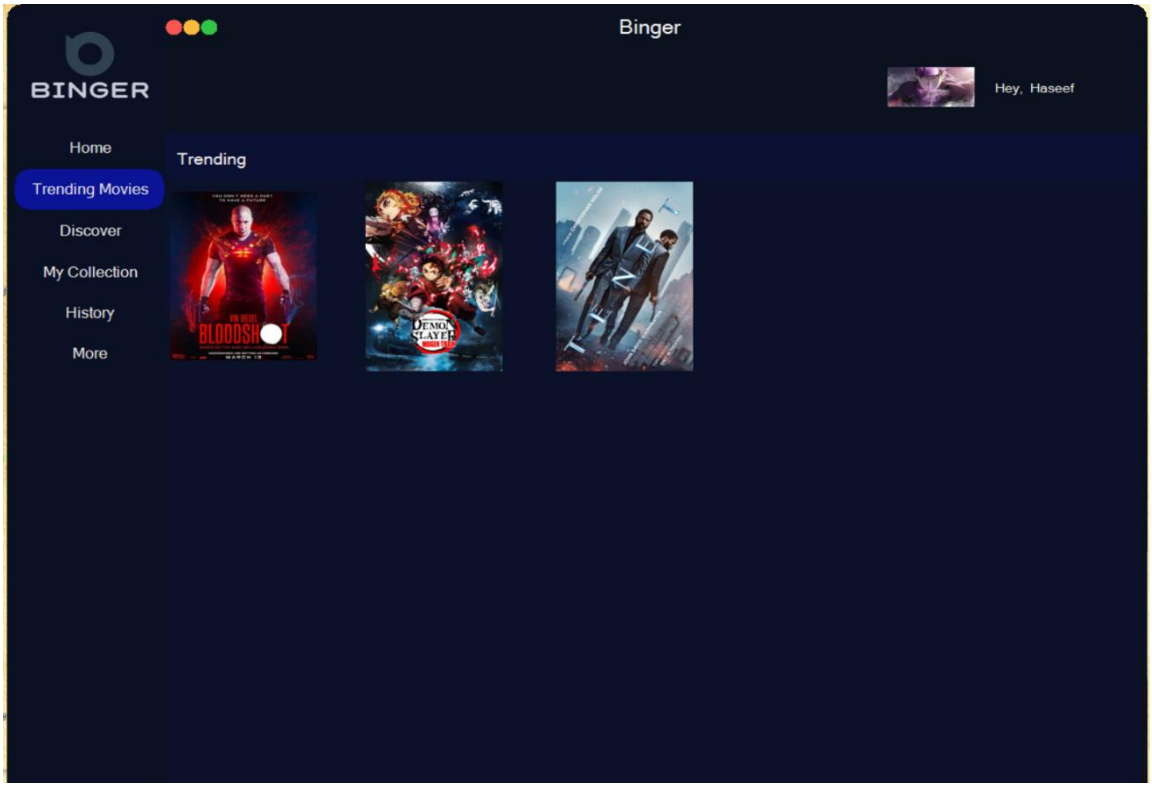
Forget Password



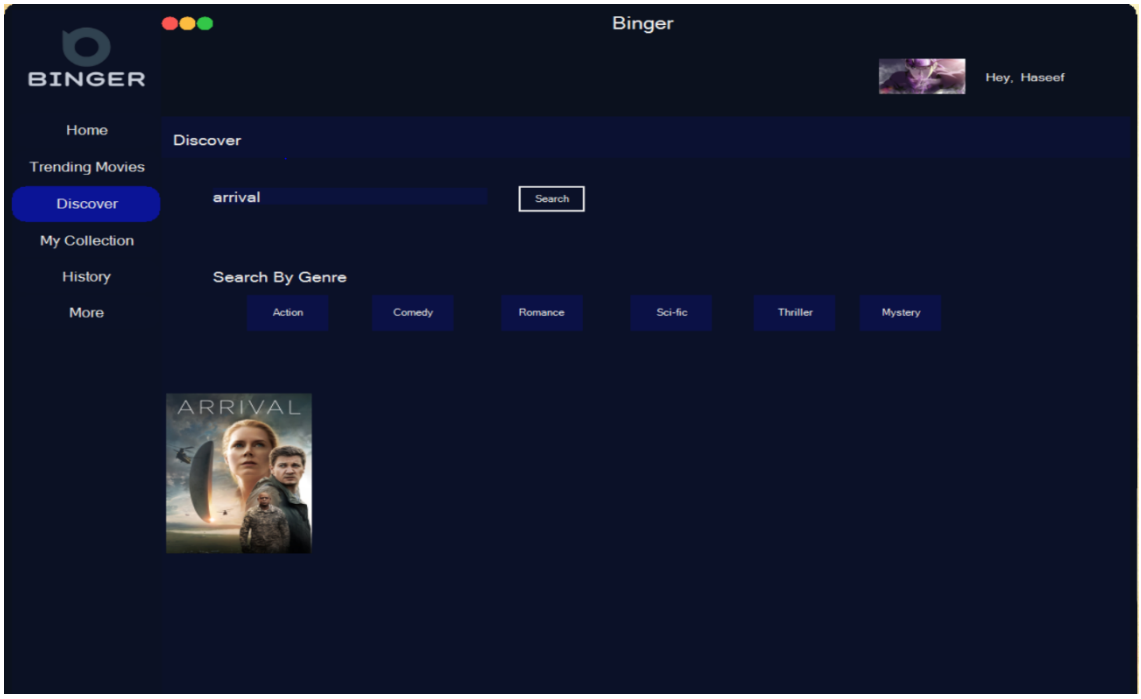
Main Page (Landing Page)



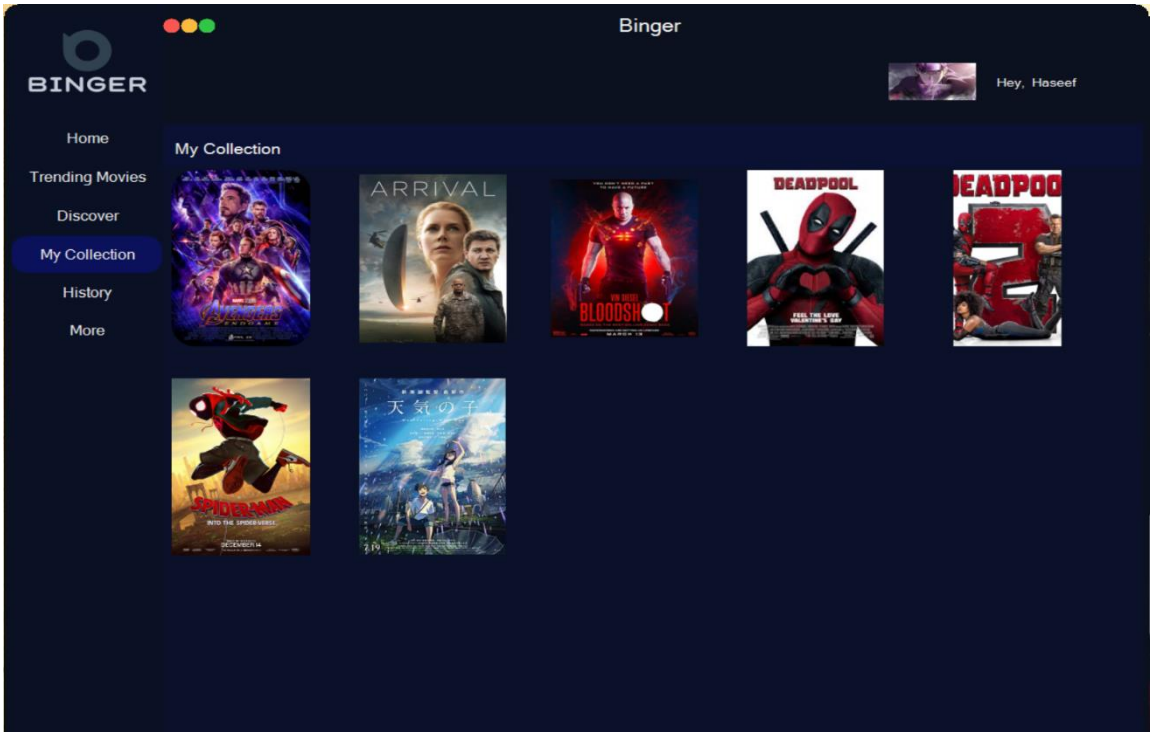
Trending Page



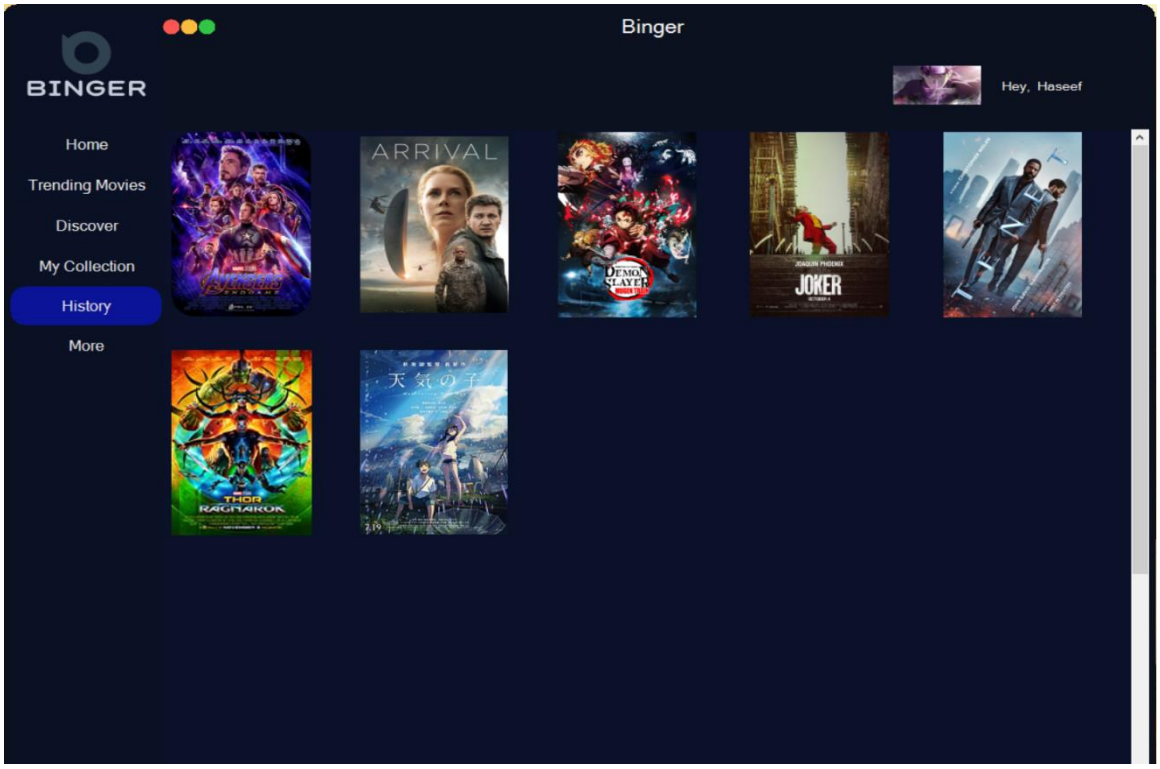
Discover New Movie Panel



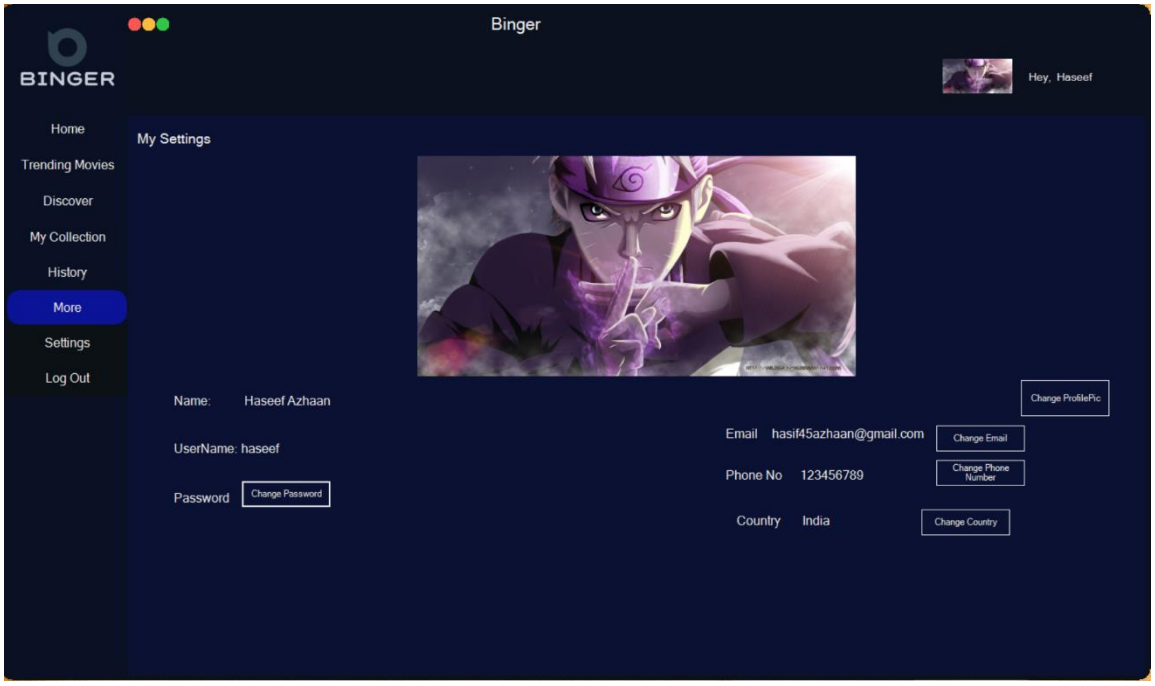
My Collection Panel



History Panel

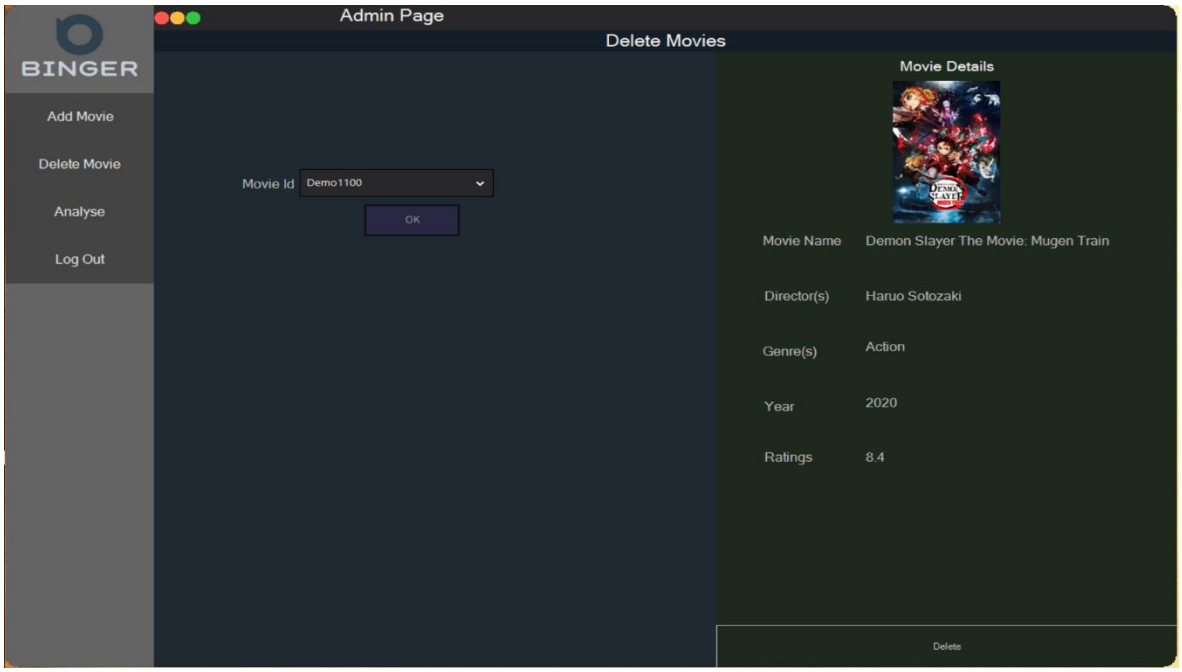


Settings

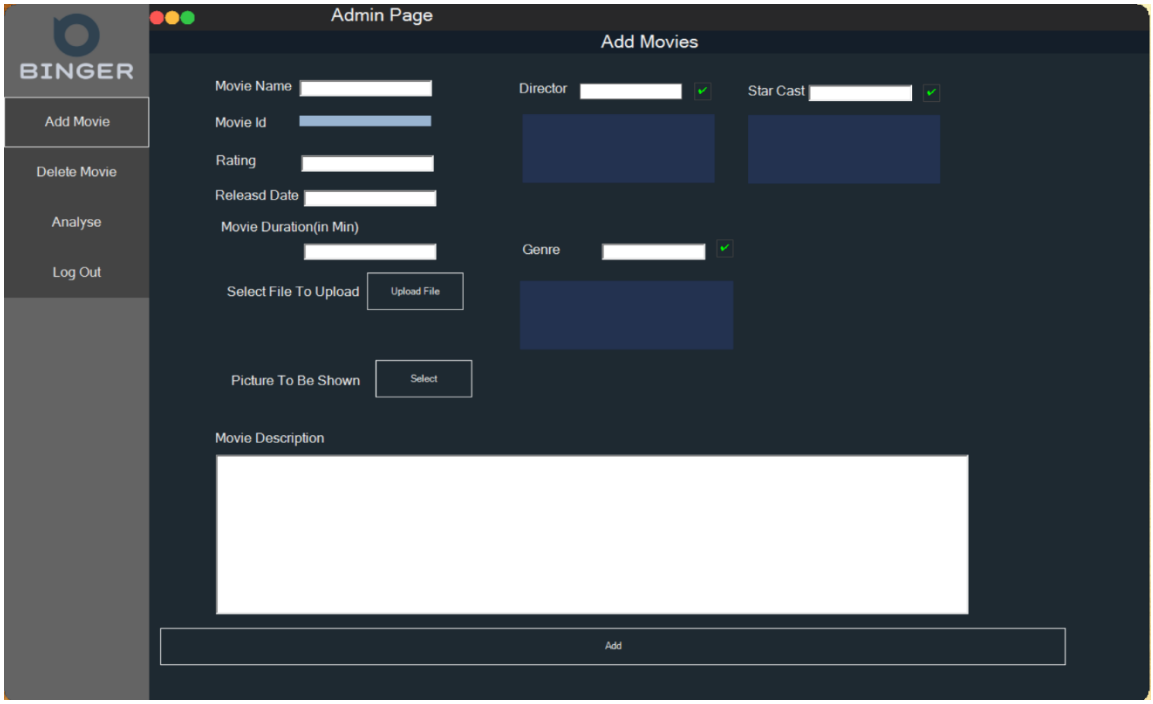


ADMIN PAGE

Add Movie



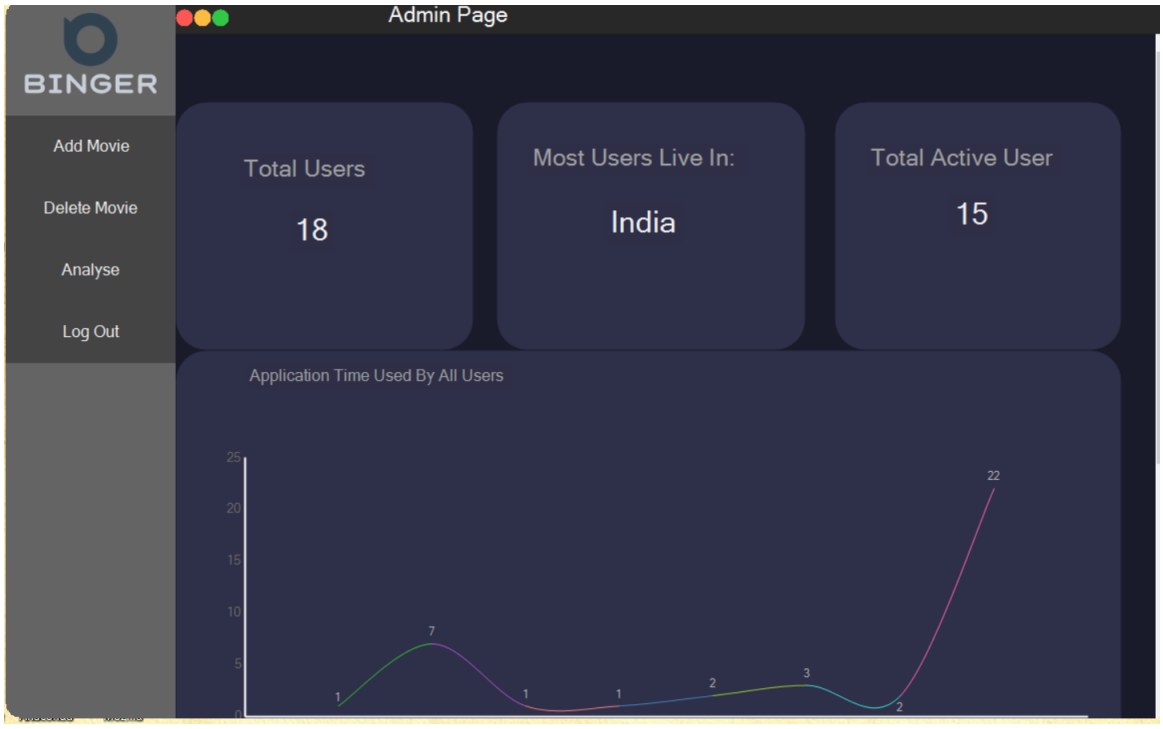
Delete Movie



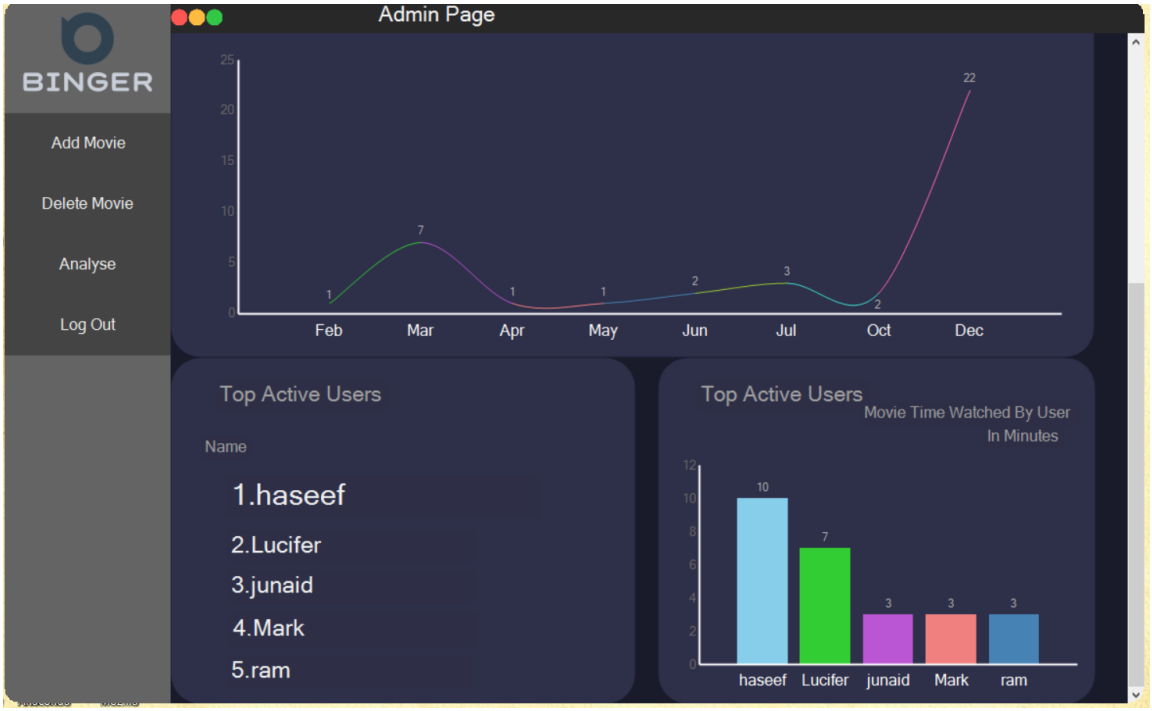
Analyse Page:

Genre Analyse Page

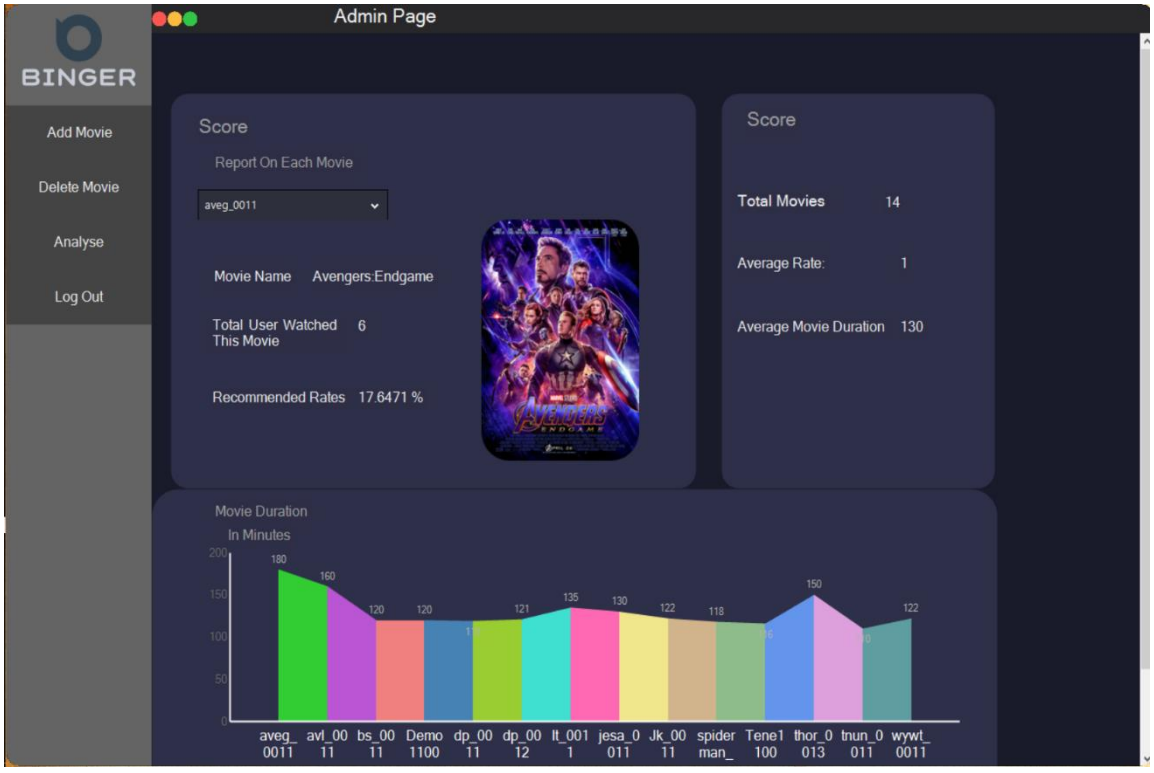
User Analyse Page 1



User Analysis Page2



Movie Analysis Page



11.APPENDINCES C-SAMPLE REPORTS OF TEST CASES

sl.no	Test Case Id	Test Description	Steps to Execute	Test Data	Expected Result	Actual Result	Status
01	Test case id 1	Correct username and password	Enters username and password in login form and clicks enter	Correct username and password	Successful login	Successful login	pass
02	Test case id 2	Correct username and Wrong password	Enters username and password in login form and clicks enter	Correct username and Wrong password	Login Failed-Invalid Credentials	Login Failed-invalid Credentials	Pass
03	Test case id 3	wrong username and correct password	Enters username and password in login form and clicks enter	Wrong username and Correct password	Login Failed-Invalid Credentials	Login Failed-invalid Credentials	Pass
04	Test case id 4	wrong username and wrong password	Enters username and password in login form and clicks enter	Wrong username and Wrong password	Login Failed-Invalid Credentials	Login Failed-Invalid Credentials	Pass
05	Test case id 5	Register New user	Enters Users Details	New User	registered Successfully	registered Successfully	Pass
06	Test case id 6	Register Existing User	Enters Users Details	Existing User	User Already Exist – Recheck username	User Already Exist – Recheck username	Pass
07	Test case id 7	View Movie Details	Clicks on Movie Picture	Click Using Mouse	Display Movie Details	Display Movie Details	Pass
08	Test case id 8	Watch movie	Clicks On ‘Watch movie’ Button	Click Using Mouse	Shows Selected Movie	Shows Selected Movie	Pass