

**A
PROJECT
ON
“Gaming HUB”**

**Submitted to
Rashtrasant Tukadoji Maharaj Nagpur University,
NAGPUR
In the Partial Fulfillment of
B.Com. (Computer Application) Final Year**

Submitted by

Jayesh S. Manapure
Deepak Kumar

**Under the Guidance of
Pravin J. Yadao**



**G. S. College of Commerce & Economics
Nagpur
2019-2020**

G. S. COLLEGE OF COMMERCE & ECONOMICS

NAGPUR

CERTIFICATE

(2019 - 2020)

This is to certify that Mr. Jayesh S. Manapure & Deepak Kumar has completed their project on the topic of “Gaming HUB” prescribed by the Rashtrasant Tukadoji Maharaj Nagpur University for B.Com. (Computer Application) - III course in G. S. College of Commerce & Economics, Nagpur.

Date: 16/03/2020

Place: Nagpur

Pravin J. Yadao

Project Guide

External Examiner

Internal Examiner

ACKNOWLEDGEMENT

We take this opportunity to express our deep gra—titude and whole hearted thanks to project guide Prof. Pravin Yadao, Coordinator for his guidance throughout this work. We are very much thankful to him for his constant encouragement, support and kindness.

We are also grateful to our teachers Prof. Rahul Tiwari, Prof. Sushma Gawande, Prof. Preeti Rangari, Prof. Prajkta Deshpande and Prof. Haresh Naringe for their encouragement, help and support from time to time.

We also wish to express our sincere thanks to Principal Dr. N. Y. Khandait for providing us wide range of opportunities, facilities and inspiration to gather professional knowledge and material without which this project could not have been completed.

Jayesh S. Manapure

Deepak Kumar

Date: 16/03/2020

Place: Nagpur

DECLARATION

We **Jayesh S. Manapure & Deepak Kumar** hereby honestly declare that the work entitled **“Gaming HUB”** submitted by us at G.S. College of Commerce & Economics, Nagpur in partial fulfillment of requirement for the award of B.Com. (Computer Application) degree by Rashtrasant Tukadoji Maharaj, Nagpur University, Nagpur has not been submitted elsewhere for the award of any degree, during the academic session 2019-2020.

The project has been developed and completed by us independently under the supervision of the subject teacher and project guide.

Jayesh S. Manapure

Deepak Kumar

Date: 16/03/2020

Place: Nagpur

INDEX

Sr No.	PARTICULARS	PAGE No.	REMARKS
1	Introduction	6 - 8	
2	Objectives	9 - 11	
3	Preliminary Investigation 3.1 Identification of Need 3.2 Preliminary Investigation 3.3 Need for New System 3.4 Feasibility Study 3.5 Project Category	12 - 18	
4	Hardware & Software Requirement Specification	19 - 20	
5	Detailed System Analysis	21 - 24	
6	System Design 6.1 Form Design 6.2 Source Code 6.3 Input & Output Screens	25 - 65	
7	Testing & Validation Checks71	66- 69	
8	System Security Measures	70 - 71	
9	Implementation, Evaluation & Maintenance	72 - 77	
10	Future Scope of the Project	78 - 79	
11	Conclusion	80 - 81	
12	Bibliography	82 - 83	
13	Synopsis	84 - 85	

INTRODUCTION

The Gaming Hub is a collection of various kinds of interesting games. This includes game such as: **Car Racing, Hit Game, Tic Tac Toe & Puzzle.**

These games are structured form of play, usually used for enjoyment and also used as an educational tool. It consists of Four games and the user can choose a game which he wants to play. It is a very interesting and attracting platform to play the games so that the players or the user will not get bored.

This software enables the user to reduce operational time and costs and improve product development and scalability.

This proposed system provides good facilities to user with GUI forms to registration, enjoy the games and to make the project attractive. According to the system, there are two users one is admin who has control over updating the required updates time to time and also responsible for maintenance of customer information and data security. Second is customer who can access the system for fulfilling their requirements. Here, initially there is a requirement for users to register in order to use the system. Registered users can then enter the system by logging on. this requires

(1) User Id

(2) Password.

The password must be kept secret (and is not displayed as it is entered). After login, the home page will be displayed where user have to select the game as per his/her requirement there are 4 Games:

Car Racing:

Car Racing game is a very interesting game in this game user needs to focus with full concentration so that he/she score more. After selecting the car racing option main menu page opens which consists of 4 options: Start, High Score, Back and Quit. As soon as the user clicks on Start option the game starts, as the game goes on the speed of the car increases slightly and if the car crashes to another car the game gets over and the score automatically get stored in the database. If the user clicks on high score the list of high scores appears from th e database. And if the user clicks on back the home page appears and if the user clicks on quit then the application gets close.

Hit Game:

Hit game is a very interesting game in this game user needs to focus with full concentration so that he/she score more. After selecting the hit game option main menu page opens which consists of 4 options: Start, High Score, Back and Quit. As soon as the user clicks on Start option the game starts, as the game goes on the speed of the Ant increases slightly and the user had to click on the ant if the ant hits the top most window the game gets over and the score automatically get stored in the database. If the user clicks on high score the list of high scores appears from the database. And if the user clicks on back the home page appears and if the user clicks on quit then the application gets close.

Tic Tac Toe:

Tic Tac Toe game is a old game used to play in childhood. After selecting the tic tac toe option main menu page opens which consists of 3 options: Start, Back and Quit. The Player name window will appear over there the user has to give their name. As soon as the user clicks on Start option the game starts. if the user clicks on back the home page appears and if the user clicks on quit then the application gets close.

Puzzle:

Puzzle game is a very interesting game in this game user needs to focus with full concentration so that he/she score more. After selecting the puzzle option main menu page opens which consists of 4 options: Start, High Score, Back and Quit. As soon as the user clicks on Start option the game starts, it is a puzzle of numbers, the user has to arrange the numbers from 1 to 15, as soon as the user arranges it the game gets over and the time and moves he takes to complete the puzzle automatically get stored in the database. If the user clicks on high score the list of high scores appears from the database. And if the user clicks on back the home page appears and if the user clicks on quit then the application gets close.

OBJECTIVES

1.Entertainment:

Our main objective is to provide entertainment, fun and to refresh the mind of the user. It also increases the level of concentration, ability to think and focus of the user.

2.Reliability:

This system can be used again and again. It has reliability feature,

by this we can use this application anywhere at any time.

3.Interactive:

Different commands and menus are provided in the project to make it more interactive so that one operates it very easily and should feel it entertaining.

4. User Friendly:

It is important that the software which is developed should be user friendly. Our project meet this requirement as it is more interactive and easy to use. There is no chance of any error.

5. Flexible:

This software is quite flexible as it allows one to add-on new updations or changes that one find suitable. The updation can be made according to the requirement which makes this project more flexible.

6. Easy to Access:

In the project, various command buttons are given on each form. Every activity related to project can be performed at the click event of these command buttons which make this project easy to access.

7. Easy to Handling:

Handling such type of project is not a difficult task. All you need is some basic knowledge about the system and the software. For handling purpose, specified knowledge is not required.

8. Cost Effective:

This project does not cost much as we have taken special care that it should be cost effective. Bulky and complicated software application are not required to be purchased.

9. Affordable:

You do not need to spend much on this software as it is cost effective. It is affordable to a large extent. It can be easily handled and can be accessed by anyone who is having basic computer knowledge.

10. Maintenance:

Maintaining such type of project does not cost much and don't require any specialization and can be maintained easily.

11. Compatible:

This software is compatible for every system. It does not require any specifications. Installing lengthy and bulky software are not necessary for installing this software.

PRELIMINARY SYSTEM

ANALYSIS

- **Preliminary Investigation:**

The preliminary investigation in my case revealed that the manual system and maintaining project is a tedious process and time consuming. The application “The Gaming HUB” is very easy to handle even by the naive user and also alterations with this application is also very easy and user friendly.

In our day to day life, we come across various systems. Therefore, today’s condition is to manage each and everything in a proper system.

- **Present System in use:**

There are various kind of gaming applications available right now in the current world. There are many kind of gaming platforms in the current world like PSP, PC, X-Box, etc.

There are online as well as offline, both type of games available in the market. In offline games, you compete with the computer bots or you can say, you play the campaign part of the game whereas in online games, we compete with human players who play that game from whole world.

- **Flaws of Present System:**

The present system has a lot of good things but it also has some flaws like people having no facility of internet cannot play all kind of online games and hence he just left with offline games to satisfy himself. In the current online games, we also see that games are hacked by various players which makes the game unfair and it discourages the other players or the players who just came to know about that game to play and it also ruins the image of gaming community of that particular country of which the palyers hack.

- **Need of new System:**

As we all know, most of the people don't have internet connection, so they are unable to play online games. So we created this application "The Gaming HUB" which is an offline application and contains some of the best worldwide games like puzzle, tictactoe, racing game, etc. The user just need to signup if he don't have his account, and if he already have signed up then he just need to login and that's all, he will come to the home page of the application and from there he can select any game he want to play.

One also cannot hack into our system as we provided a strong security to the application and we do not provide the source code of the application openly, so there is no chance for the players to hack into the system.

- **Feasibility Study:**

Feasibility is the determination of whether or not a project is worth doing. The process followed in making the determination is called feasibility study. Once, it is determined that project is feasible, the analyst can go ahead and prepare the project specification which finalizes project requirements.

1.Technical Feasibility:

In this Proposed System, this is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include:

The facility to produce outputs in a given time.

- Response time under certain conditions.
- Facility to communicate data to distant location.

2.Economical Feasibility:

In this system, Development Cost:

- Once the required hardware and software requirements gets fulfilled, there is no need for the user of our system to spend for any additional overhead.
- Equipment's required for developing the software are easily available.

3.Social Feasibility:

Social Feasibility is a determination of whether this proposed system will be acceptable to the people or not. This determination typically examines the probability of the project accepted by the group directly affected by the proposed system change.

- It describes the effect on users from the introduction of the new system considering whether there will be need for retraining the workforce.
- It describes how you propose to ensure user co-operation before changes are introduced.

- **Project Category:**

Platform/Language:

Front-End : VB.Net

Back-End : MS-Access

Platform : Windows 10

In this proposed system “Gaming HUB”, we use VB.Net as front-end and Microsoft Access as back-end of this project. Brief review of the package:

Microsoft Visual Basic. Net

Visual Basic .Net (VB.Net) is an object-oriented computer programming language implemented on the .Net framework. Although it is an evolution of classic Visual Basic language, it is not backwards-compatible with VB6, and any code written in the old version does not compile under VB.NET.

VB.NET is implemented by Microsoft’s .NET Framework. Therefore, it has full access to all the libraries in the .Net Framework. it’s also possible to run VB.Net programs not only under Windows, but even Linux or Mac OSX.

This programming language provides support for language features like implementation inheritance, encapsulation and polymorphism etc. Visual Basic.Net provides a visual forms designer to help web developers create web forms easily.

VB.Net also provides the usage of GUI (Graphical User Interface) which generally consists of one or more forms that contain text boxes, labels, buttons, picture boxes, etc.

MS-Access

Data is a collection of information that is organized on a list. MS-Access is one of such packages that has increasingly gained acceptance as DBMS. Starting the small database feature, one can create a complex data handling application. Access is used to store data and manipulate large amount of data and automate respective task. For the requirement of our project it is very supported and successive.

In this proposed system, we require “Microsoft Access” as backend. As in today’s information technology age, there is a need to manage information effectively. MS-Access is a powerful tool to create and manage database. It has in-build features that assist you in construction and viewing information.

HARDWARE AND SOFTWARE **REQUIREMENT SPECIFICATION**

HARDWARE

The Hardware means basic physical component, which together with connectivity in them, forms the machine is called computer, without the computer can't brought through of:

- 4 GB RAM and Above
- 320 GB HARDDISK and Above
- Keyboard
- Mouse
- Processor (CPU) with 2 gigahertz (GHz) frequency or Above
- Monitor Resolution 1024 * 768 or Above

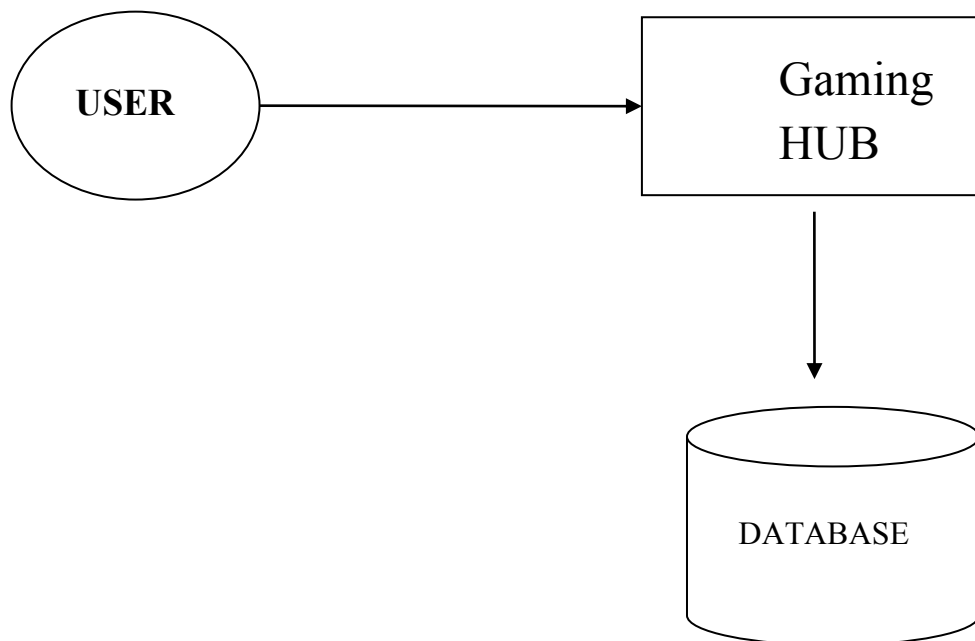
SOFTWARE

Software can be termed as the group of instruction or command used by the computer to accomplish the given task. In today's world generation of software is ever ending. It is an evolution of dignified technology.

- **OPERATING SYSTEM:** Microsoft Windows – 7 or Above
- **LANGUAGES (FRONT END):** Visual Basic 2010 Ultimate
- **DBMS (BACK END):** Microsoft Access 2010

DETAILED SYSTEM ANALYSIS

- **DATA FLOW DIAGRAM**



- **DATA STRUCTURES AND TABLES**

SIGNUP

Signup						
ID	P_Name	E-Mail	Ph_No	Password	Question	Answer
6	Shubham	shubham@gma	123456789	panda	What is your birth place?	Nagpur
7	deepak	deepak@gmail.	7545785	panda	What is your pet name?	panda
9	jayesh	jayesh@yahoo.	871545457	doggy	What is your pet name?	bingo

HIGH SCORE:

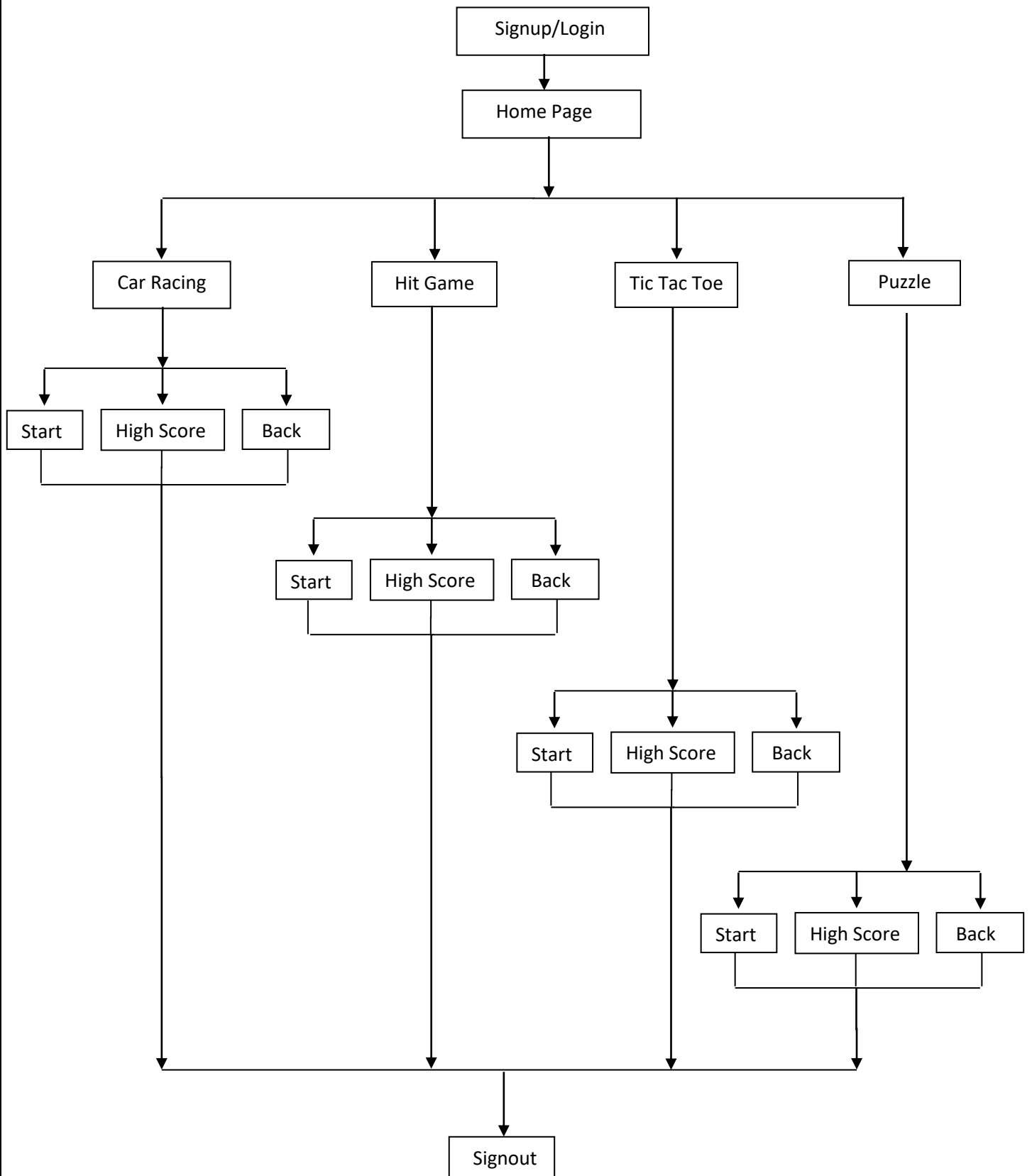
Car Racing

CarRacing	
PlayerName	HighScore
Deepak	82
jayesh	28
jayesh	16

Hit Game

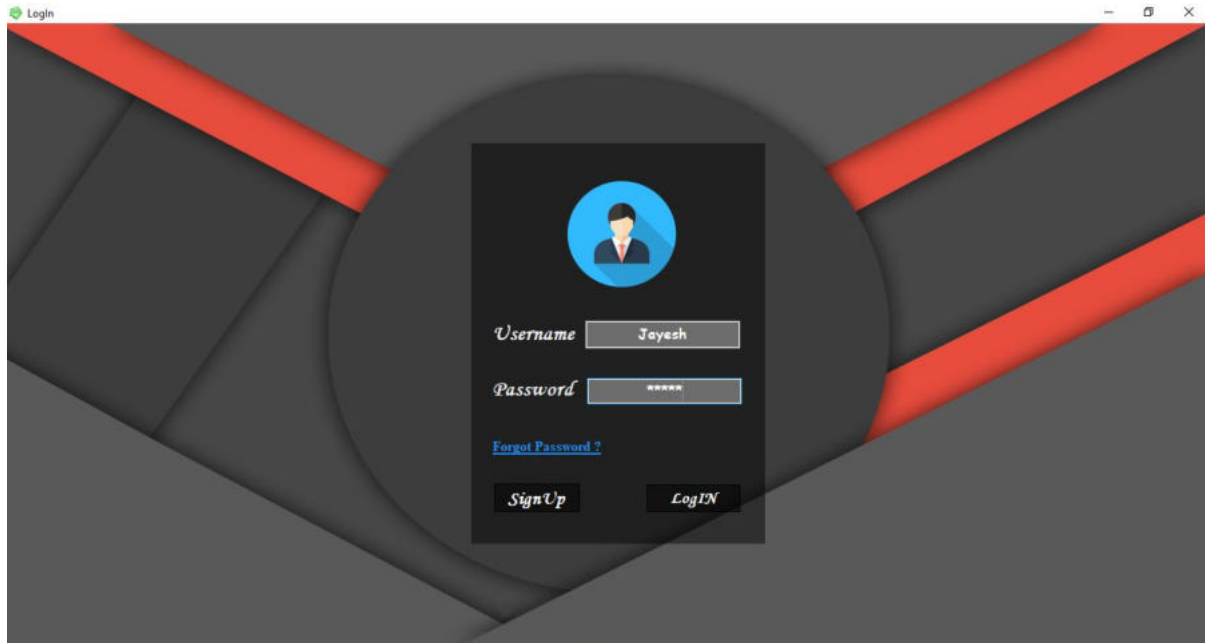
HitGame	
PlayerName	HighScore
jayesh	6
jayesh	38
jayesh	7

- DATA SYSTEM ANALYSIS



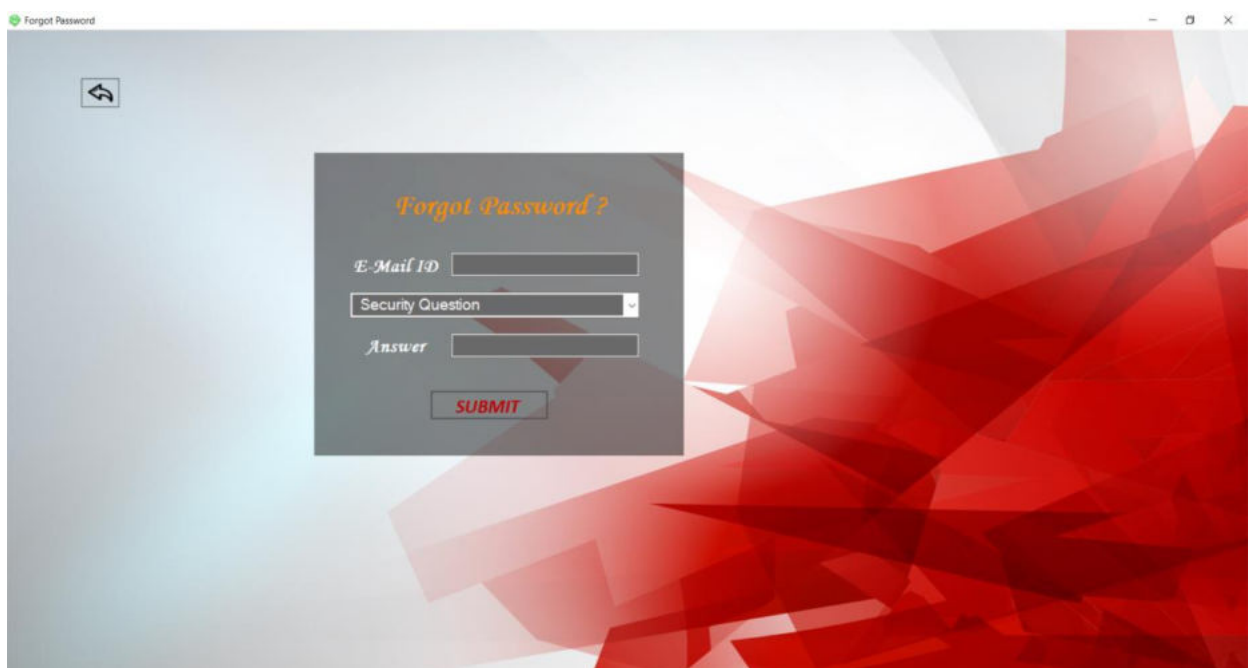
SYSTEM DESGIN

LOGIN FORM



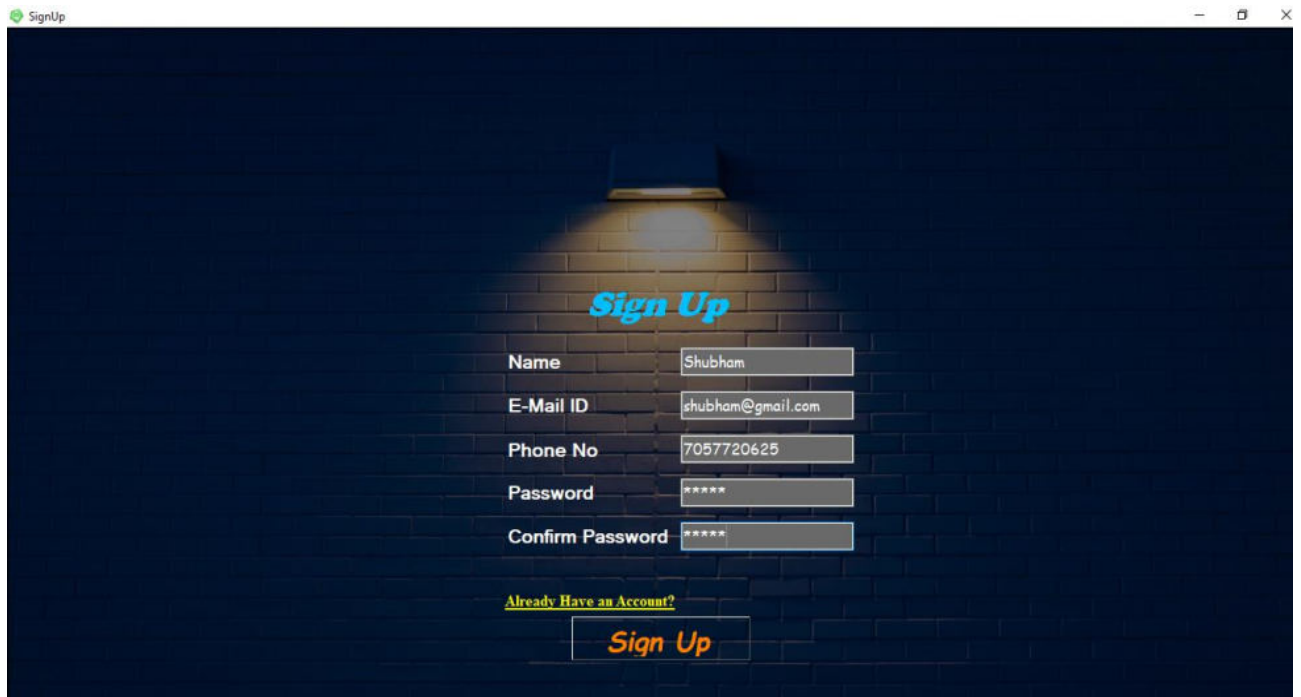
The screenshot shows a web application window titled "Login". The background is dark gray with red diagonal stripes. In the center, there is a dark gray rounded rectangle containing a blue circular profile picture of a man. Below the profile picture, the text "Username" is followed by a text input field containing "Joyesh". Below that, the text "Password" is followed by a password input field with six asterisks. A blue link "Forgot Password ?" is positioned below the password field. At the bottom of the form, there are two buttons: "Sign Up" and "Log IN".

FORGET PASSWORD



The screenshot shows a web application window titled "Forgot Password". The background is light gray with a red abstract geometric pattern on the right side. In the top left corner, there is a small icon of a house with a circular arrow. The main form is a dark gray rounded rectangle with the title "Forgot Password ?" in orange. It contains three input fields: "E-Mail ID", "Security Question" (with a dropdown arrow), and "Answer". A red "SUBMIT" button is at the bottom of the form.

SIGNUP



A screenshot of a web browser window titled "SignUp". The background is a dark blue brick wall with a spotlight illuminating the center. The text "Sign Up" is written in a stylized blue font. Below it, there are five input fields for registration: Name (Shubham), E-Mail ID (shubham@gmail.com), Phone No (7057720625), Password (*****), and Confirm Password (*****). Below the fields, there is a link "Already Have an Account?" and a button labeled "Sign Up" in orange text.

Sign Up

Name

E-Mail ID

Phone No

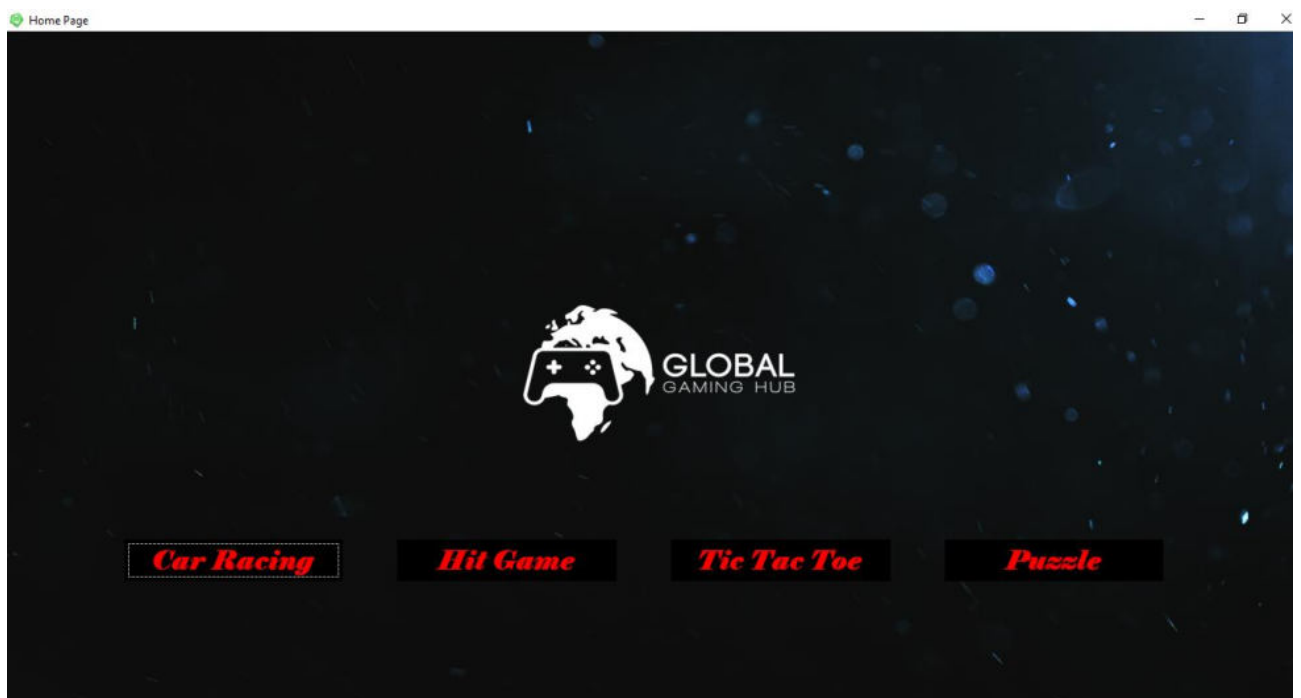
Password

Confirm Password

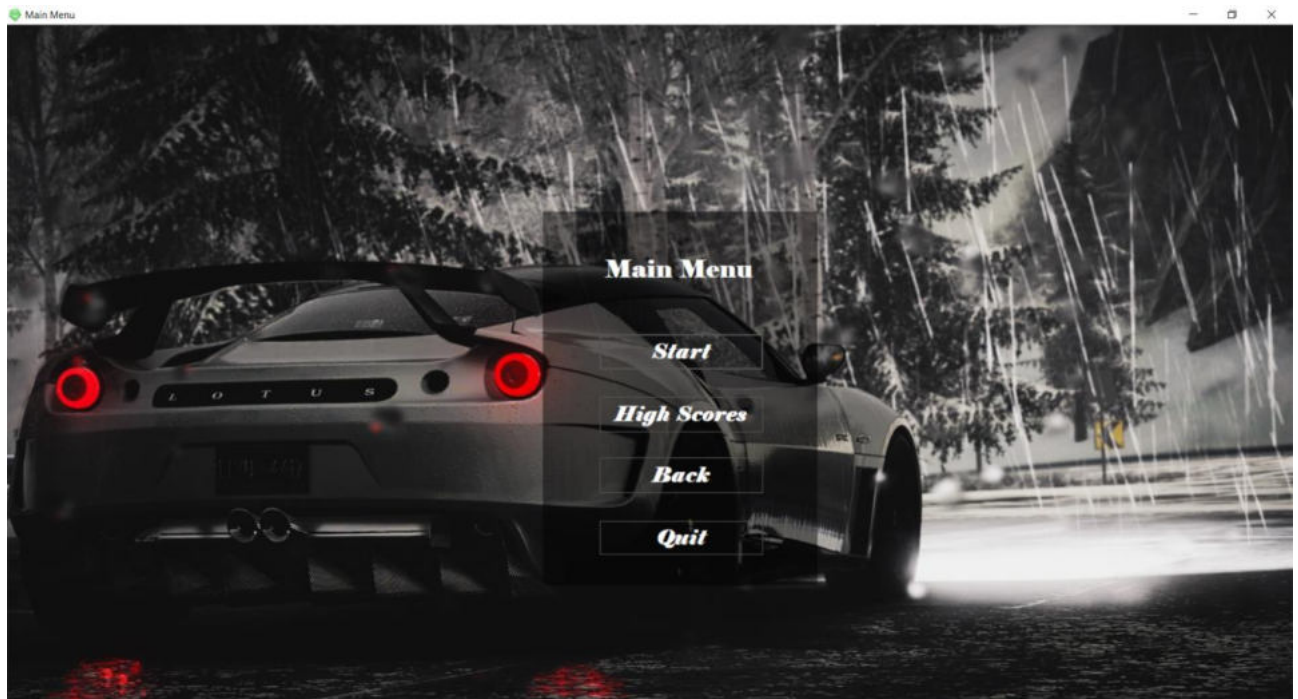
[Already Have an Account?](#)

Sign Up

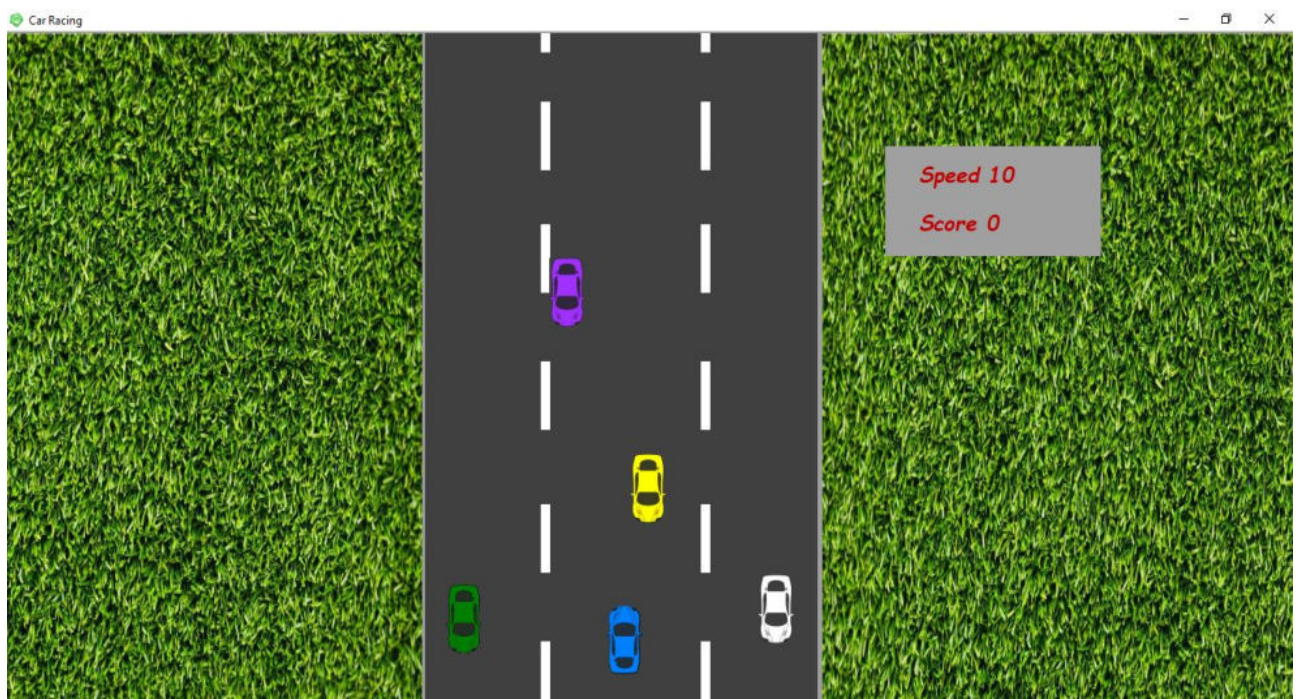
HOME



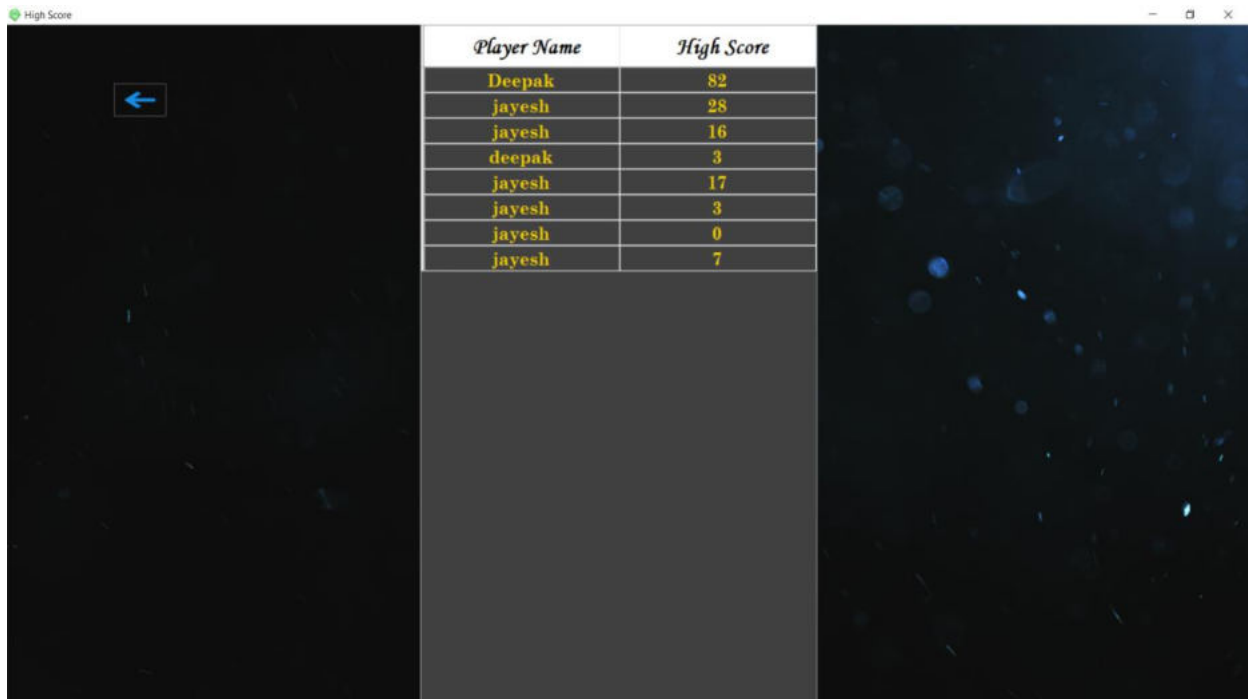
MAIN MENU



CAR RACING



RACING GAME HIGH SCORE

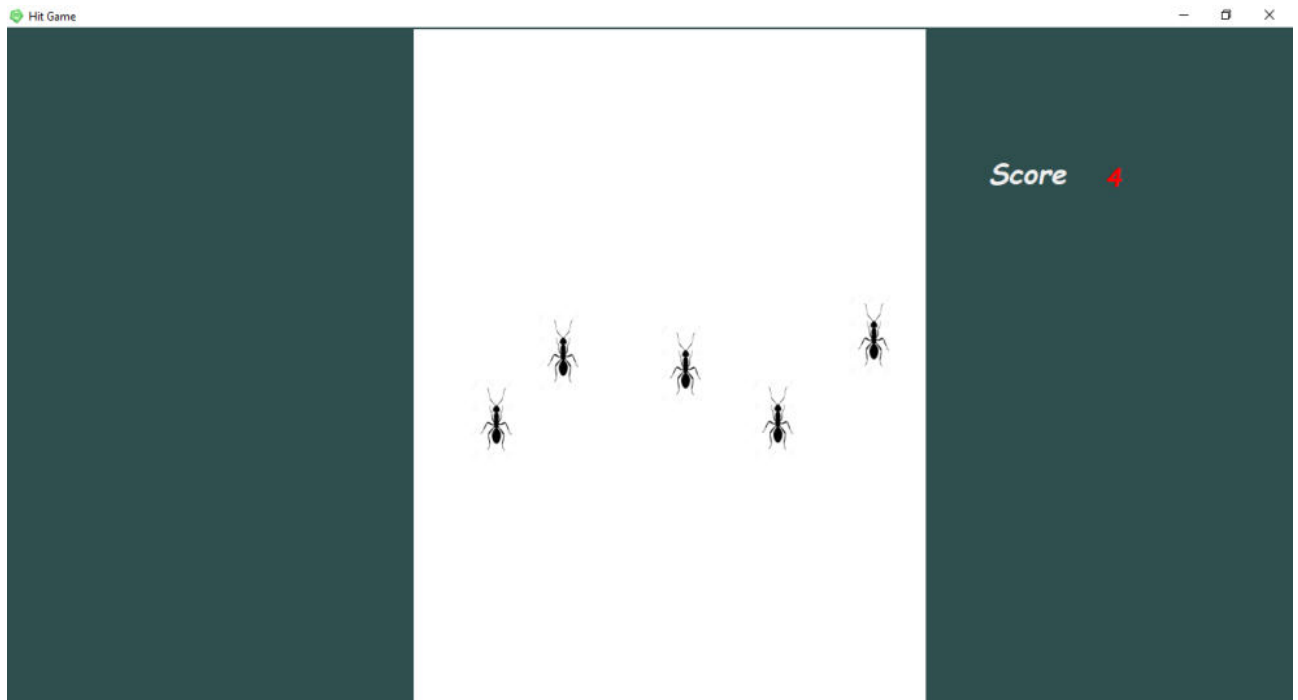


<i>Player Name</i>	<i>High Score</i>
Deepak	82
jayesh	28
jayesh	16
deepak	3
jayesh	17
jayesh	3
jayesh	0
jayesh	7

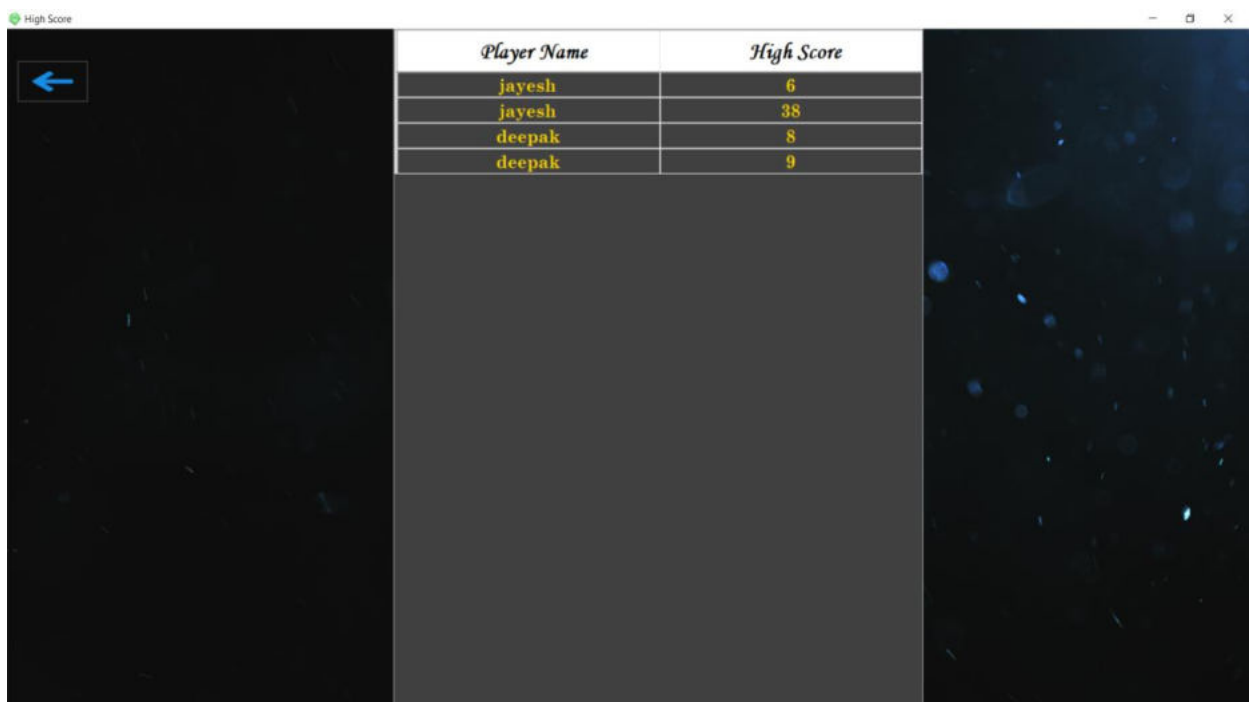
HIT GAME MAIN MENU



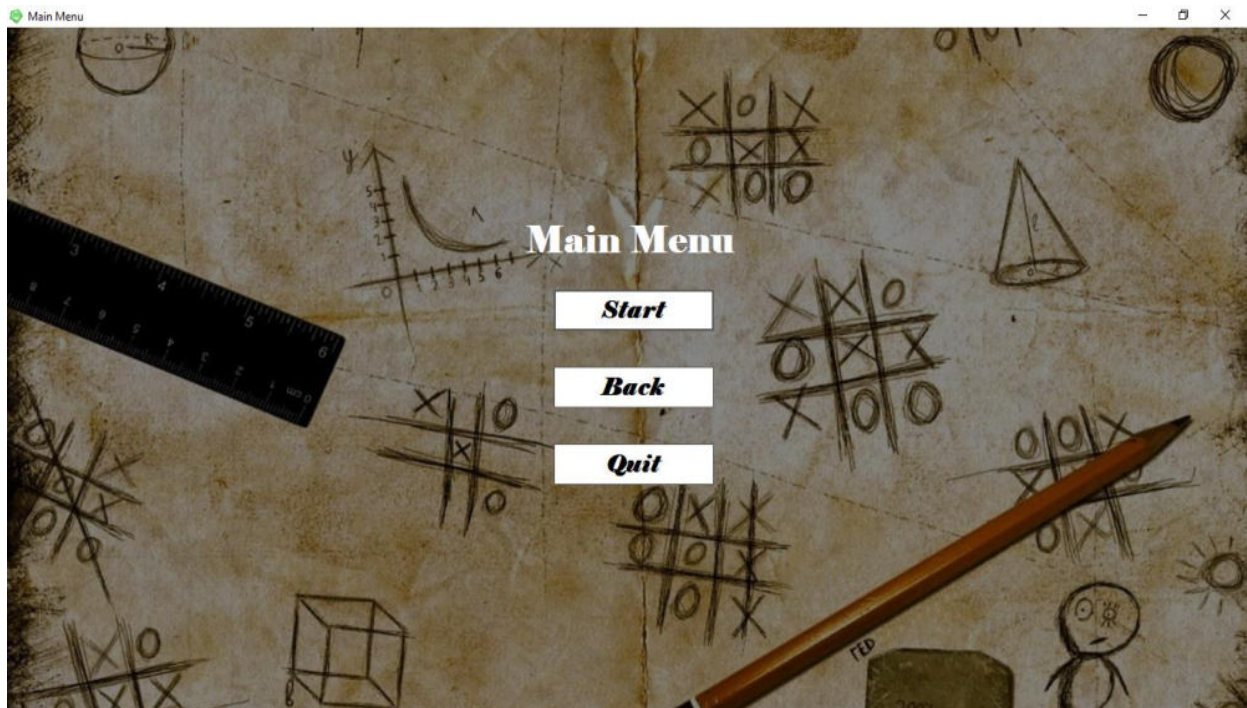
HIT GAME



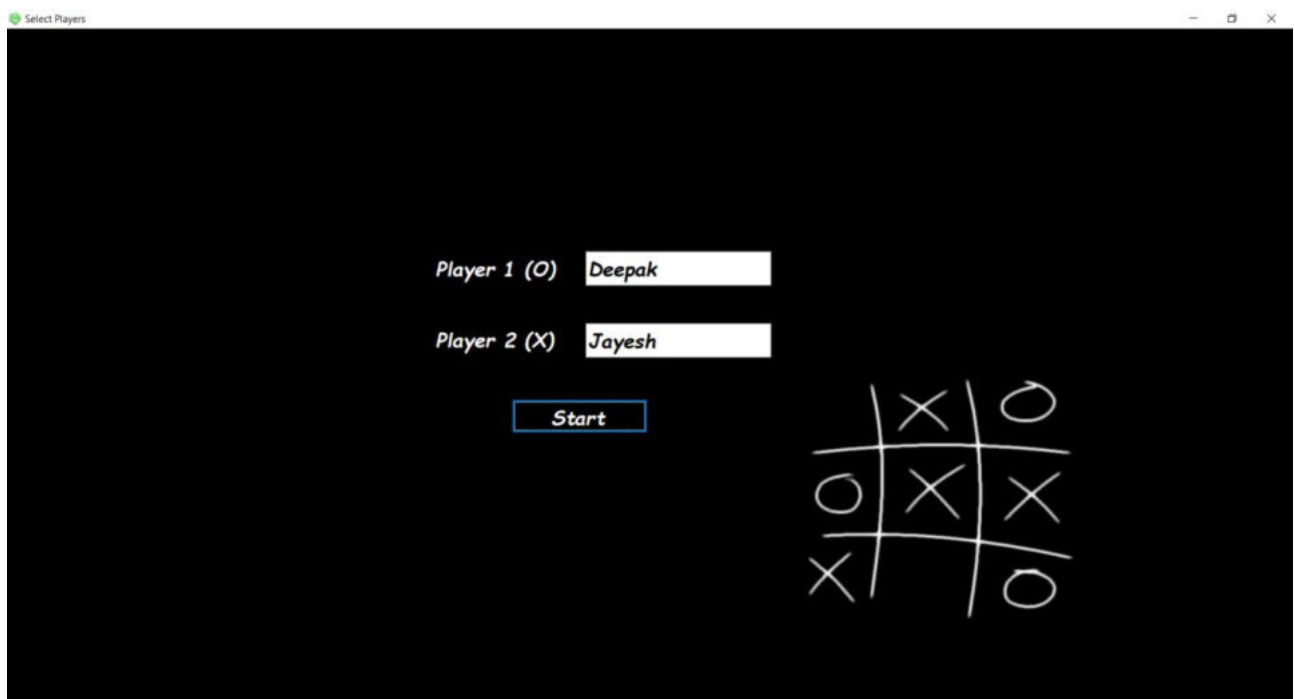
HIT GAME HIGH SCORE



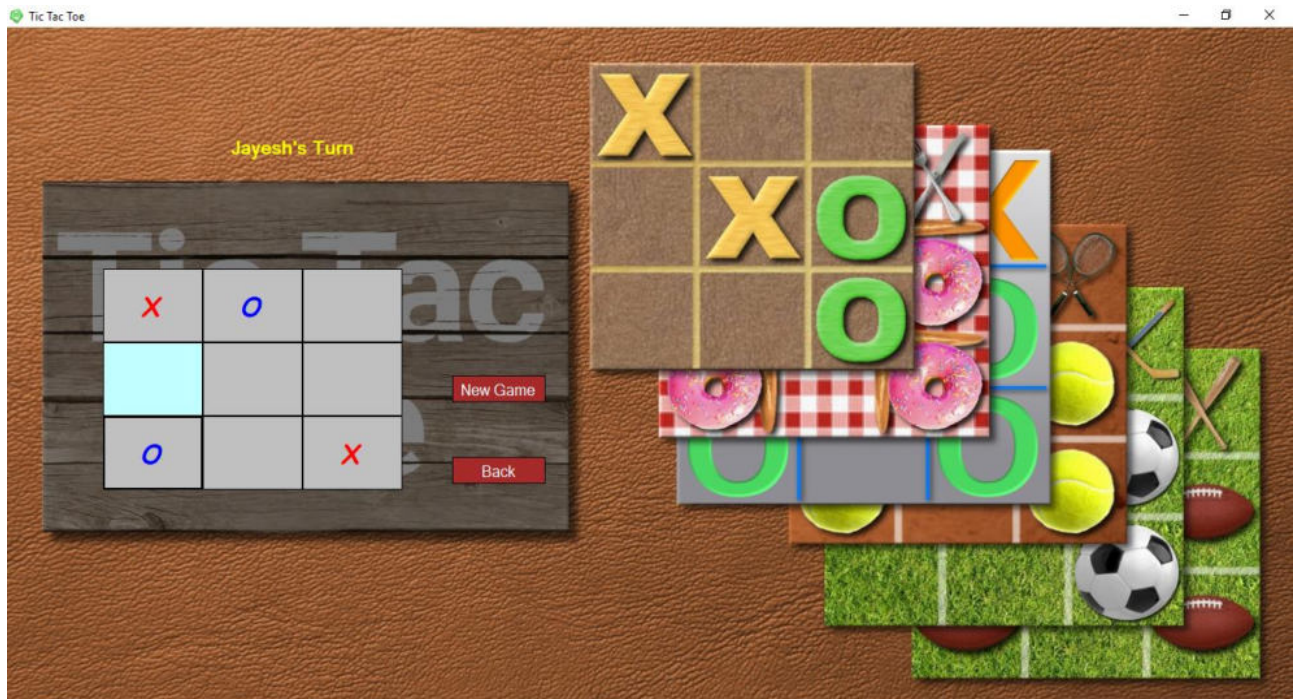
TIC TAC TOE MAIN MENU



SELECT PLAYERS



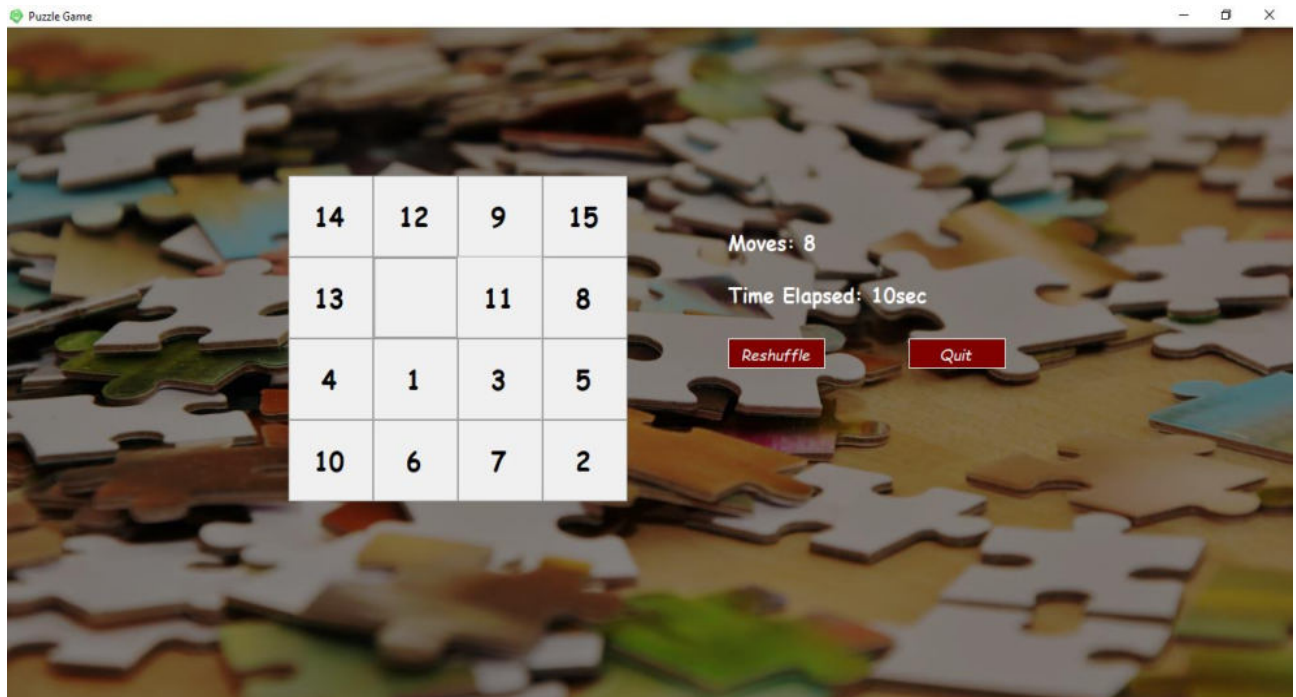
TIC TAC TOE GAME



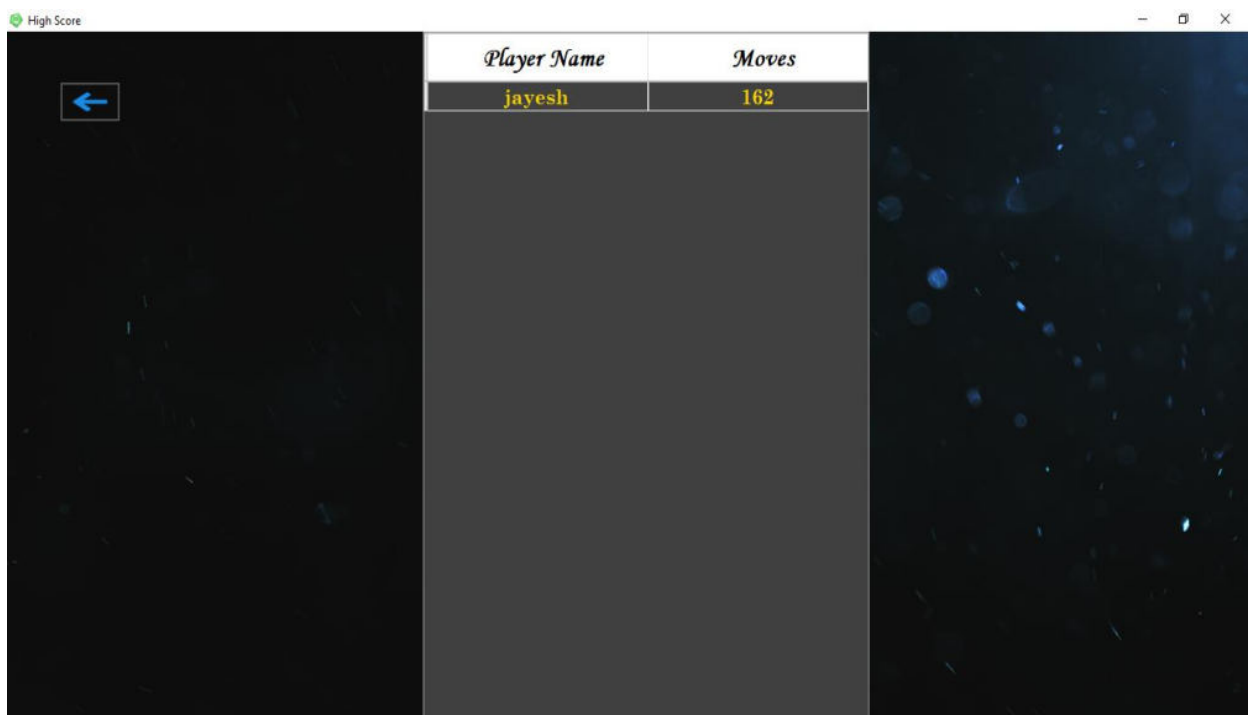
PUZZLE MAIN MENU



PUZZLE GAME



PUZZLE HIGH SCORE



• SOURCE CODE

SIGNUP

```
Imports System.Data.OleDb
Public Class SignUp
    Dim provider As String
    Dim datafile As String
    Dim connstring As String
    Dim myconnection As OleDbConnection = New
OleDbConnection
    Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
        Try
            If TextBox1.Text <> Nothing And TextBox2.Text <>
Nothing And TextBox3.Text <> Nothing And TextBox4.Text <>
Nothing And TextBox6.Text <> Nothing Then
                If TextBox2.Text.Contains("@gmail.com") Or
TextBox2.Text.Contains("@rediffmail.com") Or
TextBox2.Text.Contains("@yahoo.com") Then
                    If TextBox5.Text = TextBox4.Text Then
                        provider =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source="
                        datafile =
"C:\Users\admin\Desktop\New folder\Gaming
HUB(Feb)\Database\GamingHUB.accdb"
                        connstring = provider & datafile
myconnection.ConnectionString =
connstring

                        myconnection.Open()
                        Dim str As String
                        str = "Insert into
Signup([P_Name],[EMail],[Ph_No],[Password],[Question],[Answ
er]) values (?, ?, ?, ?, ?, ?)"
                        Dim cmd As OleDbCommand = New
OleDbCommand(str, myconnection)
                        cmd.Parameters.Add(New
OleDbParameter("P_Name", CType(TextBox1.Text, String)))
                        cmd.Parameters.Add(New
OleDbParameter("EMail", CType(TextBox2.Text, String)))
                        cmd.Parameters.Add(New
OleDbParameter("Ph_No", CType(TextBox3.Text, String)))
```

```

                                cmd.Parameters.Add(New
OleDbParameter("Password", CType(TextBox4.Text, String)))
                                cmd.Parameters.Add(New
OleDbParameter("Question", CType(ComboBox1.SelectedItem,
String)))
                                cmd.Parameters.Add(New
OleDbParameter("Answer", CType(TextBox6.Text, String)))
                                Try
                                    cmd.ExecuteNonQuery()
                                    cmd.Dispose()
                                    myconnection.Close()
                                    TextBox1.Clear()
                                    TextBox2.Clear()
                                    TextBox3.Clear()
                                    TextBox4.Clear()
                                    TextBox5.Clear()
                                    TextBox6.Clear()
                                Catch ex As Exception
                                    MsgBox(ex.Message)
                                End Try
                                MessageBox.Show("SignUp Successful",
"SignUp", MessageBoxButtons.OK, MessageBoxIcon.Information)
                                Me.Hide()
                                LogIn.Show()
                                Else
                                    MessageBox.Show("Password didn't
Match", "Error", MessageBoxButtons.RetryCancel,
MessageBoxIcon.Error)
                                End If
                                Else
                                    MessageBox.Show("Please enter valid E-
Mail ID", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                                End If
                                Else
                                    MessageBox.Show("Please fill all the
Fields", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                                End If
                                Catch ex As Exception
                                    MsgBox(ex.Message)
                                End Try
                            End Sub

```

```

    Private Sub Panel1_Paint(sender As Object, e As
PaintEventArgs) Handles Panel1.Paint
        Panel1.BackColor = Color.FromArgb(120, 0, 0, 0)
    End Sub

    Private Sub LinkLabel1_LinkClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel1.LinkClicked
        Me.Hide()
        LogIn.Show()
    End Sub

    Private Sub Button6_Click(sender As Object, e As
EventArgs) Handles Button6.Click
        End
    End Sub
End Class

```

LOGIN

```

Imports System.Data.OleDb
Public Class LogIn
    Private Sub Panel1_Paint(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.PaintEventArgs) Handles
Panel1.Paint
        Panel1.BackColor = Color.FromArgb(120, 0, 0, 0)
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
        Try
            Clipboard.SetText(TextBox1.Text)
        abc:
            Dim con As New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\admin\Desktop\New folder\Gaming
HUB(Feb)\Database\GamingHUB.accdb")

```

```

        Dim cmd As OleDbCommand = New
OleDbCommand("SELECT * FROM Signup WHERE P_Name = '" &
TextBox1.Text & "' AND Password = '" & TextBox2.Text & "' ",
con)

        Dim user As String = ""
        Dim pass As String = ""
        con.Open()
        Dim sdr As OleDbDataReader = cmd.ExecuteReader()
        If (sdr.Read() = True) Then
            MessageBox.Show("Login Successful!",
"Welcome", MessageBoxButtons.OK, MessageBoxIcon.Information)
            TextBox1.Clear()
            TextBox2.Clear()
            Me.Hide()
            Home.Show()
        Else
            MessageBox.Show("Invalid Username or
Password!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
        End If
        Catch ex As Exception
            GoTo abc
        End Try
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
        Me.Hide()
        SignUp.Show()
    End Sub

    Private Sub LinkLabel1_LinkClicked(sender As Object, e
As LinkLabelLinkClickedEventArgs) Handles
LinkLabel1.LinkClicked
        Me.Hide()
        ForgotPassword.Show()
    End Sub

    Private Sub Button6_Click(sender As Object, e As
EventArgs) Handles Button6.Click
        End
    End Sub
End Class

```

CAR RACING

```
Imports System.Data.OleDb
Public Class RacingGame
    Dim speed As Integer
    Dim road(11) As PictureBox
    Dim score As Integer = 0
    Dim provider As String
    Dim datafile As String
    Dim connstring As String
    Dim myconnection As OleDbConnection = New
OleDbConnection
    Private Sub Form3_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        speed = 1
        road(0) = PictureBox1
        road(1) = PictureBox2
        road(2) = PictureBox3
        road(3) = PictureBox4
        road(4) = PictureBox5
        road(5) = PictureBox6
        road(6) = PictureBox7
        road(7) = PictureBox8
        road(8) = PictureBox9
        road(9) = PictureBox10
        road(10) = PictureBox11
        road(11) = PictureBox12
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Road_Mover.Tick
        For x As Integer = 0 To 11
            road(x).Top += speed
            If road(x).Top >= Me.Height Then
                road(x).Top = -road(x).Height
            End If
        Next
        If score > 2 And score < 6 Then
            speed = 2
        End If
        If score > 6 And score < 15 Then
            speed = 3
        End If
    End Sub
End Class
```

```

End If
If score > 15 And score < 20 Then
    speed = 4
End If
If score > 20 And score < 30 Then
    speed = 5
End If
If score > 30 And score < 40 Then
    speed = 6
End If
If score > 40 And score < 60 Then
    speed = 7
End If
If score > 60 And score < 80 Then
    speed = 8
End If
If score > 80 And score < 100 Then
    speed = 9
End If
If score > 100 And score < 120 Then
    speed = 10
End If
Speed_text.Text = "Speed " & speed * 10
If (Car.Bounds.IntersectsWith(EnemyCar1.Bounds))
Then
    gameover()
End If
If (Car.Bounds.IntersectsWith(EnemyCar2.Bounds))
Then
    gameover()
End If
If (Car.Bounds.IntersectsWith(EnemyCar3.Bounds))
Then
    gameover()
End If
If (Car.Bounds.IntersectsWith(EnemyCar4.Bounds))
Then
    gameover()
End If
End Sub
Private Sub gameover()
    End_Text.Visible = True
    Replay.Visible = True
    Road_Mover.Stop()

```

```

        Enemy_Mover1.Stop()
        Enemy_Mover2.Stop()
        Enemy_Mover3.Stop()
        Enemy_Mover4.Stop()
        Exit_btn.Visible = True
        provider = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source="
        datafile = "C:\Users\admin\Desktop\New folder\Gaming
HUB(Feb)\Database\GamingHUB.accdb"
        connstring = provider & datafile
        myconnection.ConnectionString = connstring
        myconnection.Open()
        Dim str As String
        str = "Insert into
CarRacing([PlayerName],[HighScore]) values (?,?)"
        Dim cmd As OleDbCommand = New OleDbCommand(str,
myconnection)
        cmd.Parameters.Add(New OleDbParameter("PlayerName",
CType(Clipboard.GetText, String)))
        cmd.Parameters.Add(New OleDbParameter("HighScore",
CType(score, String)))
        cmd.ExecuteNonQuery()
        cmd.Dispose()
        myconnection.Close()
    End Sub

    Private Sub Form3_KeyDown(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.KeyEventArgs) Handles
MyBase.KeyDown
        If e.KeyCode = Keys.Right Then
            Move_left.Start()
        End If
        If e.KeyCode = Keys.Left Then
            Move_right.Start()
        End If
    End Sub

    Private Sub Timer2_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Move_left.Tick
        If (Car.Location.X < 375) Then
            Car.Left += 5
        End If
    End Sub

```



```

    Private Sub Timer3_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Move_right.Tick
        If (Car.Location.X > 17) Then
            Car.Left -= 5
        End If
    End Sub

    Private Sub Form3_KeyUp(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.KeyEventArgs) Handles
MyBase.KeyUp
        Move_left.Stop()
        Move_right.Stop()
    End Sub

    Private Sub Enemy_Mover1_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Enemy_Mover1.Tick
        EnemyCar1.Top += speed * 4 / 3
        If EnemyCar1.Top > Me.Height Then
            score += 1
            Score_text.Text = "Score " & score
            EnemyCar1.Top = -CInt(Math.Ceiling(Rnd() * 150))
+ EnemyCar1.Height
            EnemyCar1.Left = CInt(Math.Ceiling(Rnd() * 100))
+ 0
        End If
    End Sub

    Private Sub Enemy_Mover2_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Enemy_Mover2.Tick
        EnemyCar2.Top += speed * 2.5 / 2
        If EnemyCar2.Top > Me.Height Then
            score += 1
            Score_text.Text = "Score " & score
            EnemyCar2.Top = -CInt(Math.Ceiling(Rnd() * 150))
+ EnemyCar2.Height
            EnemyCar2.Left = CInt(Math.Ceiling(Rnd() * 100))
+ 80
        End If
    End Sub

```

```

    Private Sub Enemy_Mover3_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Enemy_Mover3.Tick
        EnemyCar3.Top += speed * 3 / 2
        If EnemyCar3.Top > Me.Height Then
            score += 1
            Score_text.Text = "Score " & score
            EnemyCar3.Top = -CInt(Math.Ceiling(Rnd() * 150))
+ EnemyCar3.Height
            EnemyCar3.Left = CInt(Math.Ceiling(Rnd() * 100))
+ 170
        End If
    End Sub
    Private Sub Enemy_Mover4_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Enemy_Mover4.Tick
        EnemyCar4.Top += speed * 5 / 3
        If EnemyCar4.Top > Me.Height Then
            score += 1
            Score_text.Text = "Score " & score
            EnemyCar4.Top = -CInt(Math.Ceiling(Rnd() * 150))
+ EnemyCar4.Height
            EnemyCar4.Left = CInt(Math.Ceiling(Rnd() * 100))
+ 270
        End If
    End Sub
    Private Sub Button1_Click_1(ByVal sender As Object,
ByVal e As EventArgs) Handles Exit_btn.Click
        Me.Close()
        RacingStrt.Show()
    End Sub
    Private Sub Panel2_Paint(ByVal sender As Object, ByVal e
As PaintEventArgs)
        Panel2.BackColor = Color.FromArgb(120, 0, 0, 0)
    End Sub
    Private Sub Replay_Click(sender As Object, e As
EventArgs) Handles Replay.Click
        score = 0
        Me.Controls.Clear()
        InitializeComponent()
        Form3_Load(e, e)
    End Sub
End Class

```

HIT GAME

```
Imports System.Data.OleDb
Public Class HitGame
    Dim provider As String
    Dim datafile As String
    Dim connstring As String
    Dim myconnection As OleDbConnection = New
OleDbConnection
    Dim score As Integer
    Private Sub Timer1_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer1.Tick
        Ant2.Top -= 8
        Ant1.Top -= 14
        Ant5.Top -= 12
        Ant3.Top -= 10
        Ant4.Top -= 8
    End Sub

    Private Sub Timer2_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer2.Tick
        If Ant2.Location.Y < -30 Or Ant1.Location.Y < -30 Or
Ant5.Location.Y < -30 Or Ant3.Location.Y < -30 Or
Ant4.Location.Y < -30 Then
            gameover()
        End If
    End Sub
    Private Sub gameover()
        Exit_btn.Visible = True
        Replay_Button.Visible = True
        End_Text.Visible = True
        Timer1.Stop()
        Timer2.Stop()
        Ant1.Enabled = False
        Ant2.Enabled = False
        Ant3.Enabled = False
        Ant4.Enabled = False
        Ant5.Enabled = False
        provider = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source="
        datafile = "C:\Users\admin\Desktop\New folder\Gaming
HUB(Feb)\Database\GamingHUB.accdb"
        connstring = provider & datafile
    End Sub
End Class
```

```

        myconnection.ConnectionString = connstring
        myconnection.Open()
        Dim str As String
        str = "Insert into HitGame([PlayerName],[HighScore])
values (?,?)"
        Dim cmd As OleDbCommand = New OleDbCommand(str,
myconnection)
        cmd.Parameters.Add(New OleDbParameter("PlayerName",
CType(Clipboard.GetText(), String)))
        cmd.Parameters.Add(New OleDbParameter("HighScore",
CType(score, String)))
        cmd.ExecuteNonQuery()
        cmd.Dispose()
        myconnection.Close()
    End Sub

    Private Sub PictureBox1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Ant2.Click
        score = score + 1
        Label1.Text = score
        Ant2.Location = New Point(140, 615)
    End Sub

    Private Sub PictureBox2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Ant1.Click
        score = score + 1
        Label1.Text = score
        Ant1.Location = New Point(60, 360)
    End Sub

    Private Sub PictureBox3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Ant5.Click
        score = score + 1
        Label1.Text = score
        Ant5.Location = New Point(461, 381)
    End Sub

    Private Sub Ant3_Click(ByVal sender As Object, ByVal e
As EventArgs) Handles Ant3.Click
        score = score + 1
        Label1.Text = score
        Ant3.Location = New Point(261, 463)

```

```

End Sub

Private Sub Ant4_Click(ByVal sender As Object, ByVal e
As EventArgs) Handles Ant4.Click
    score = score + 1
    Label1.Text = score
    Ant4.Location = New Point(386, 650)
End Sub

Private Sub Exit_btn_Click(ByVal sender As Object, ByVal
e As EventArgs) Handles Exit_btn.Click
    Me.Hide()
    HitStrt.Show()
End Sub

Private Sub Replay_Button_Click(ByVal sender As Object,
ByVal e As EventArgs) Handles Replay_Button.Click
    score = 0
    Me.Controls.Clear()
    InitializeComponent()
End Sub

End Class

```

TIC TAC TOE

```
Public Class TicTacToeGame
```

```
    Private Sub TicTacToeGame_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load
```

```
        Label1.Text = TicTacToePly.textbox1.Text + "'s Turn"
```

```
        For Each c As Control In Panel1.Controls
```

```
            If c.GetType() = GetType(Button) Then
```

```
                AddHandler c.Click, AddressOf btn_Click
```

```
            End If
```

```
        Next
```

```
    End Sub
```

```
    Dim Xor0 As Integer = 0
```

```
    Private Sub btn_Click(ByVal sender As Object, ByVal e As  
EventArgs)
```

```
        Dim btn As Button = sender
```

```
        If btn.Text.Equals("") Then
```

```
            If Xor0 Mod 2 = 0 Then
```

```
                btn.Text = "X"
```

```
                btn.ForeColor = Color.Red
```

```
                Label1.Text = TicTacToePly.textbox1.Text +
```

```
"'s Turn"
```

```
                getTheWinner()
```

```
            Else
```

```
                btn.Text = "O"
```

```
                btn.ForeColor = Color.Blue
```

```
                Label1.Text = TicTacToePly.textbox2.Text +
```

```
"'s Turn"
```

```
                getTheWinner()
```

```
            End If
```

```
            Xor0 += 1
```

```
        End If
```

```
    End Sub
```

```

Dim win As Boolean = False

Private Sub getTheWinner()

    If Not Button1.Text.Equals("") AndAlso
Button1.Text.Equals(Button2.Text) AndAlso
Button1.Text.Equals(Button3.Text) Then
        win = True
        winEffect(Button1, Button2, Button3)
    End If

    If Not Button4.Text.Equals("") AndAlso
Button4.Text.Equals(Button5.Text) AndAlso
Button4.Text.Equals(Button6.Text) Then
        win = True
        winEffect(Button4, Button5, Button6)
    End If

    If Not Button7.Text.Equals("") AndAlso
Button7.Text.Equals(Button8.Text) AndAlso
Button7.Text.Equals(Button9.Text) Then
        win = True
        winEffect(Button7, Button8, Button9)
    End If

    If Not Button1.Text.Equals("") AndAlso
Button1.Text.Equals(Button4.Text) AndAlso
Button1.Text.Equals(Button7.Text) Then
        win = True
        winEffect(Button1, Button4, Button7)
    End If

    If Not Button2.Text.Equals("") AndAlso
Button2.Text.Equals(Button5.Text) AndAlso
Button2.Text.Equals(Button8.Text) Then
        win = True
        winEffect(Button2, Button5, Button8)
    End If

    If Not Button3.Text.Equals("") AndAlso
Button3.Text.Equals(Button6.Text) AndAlso
Button3.Text.Equals(Button9.Text) Then
        win = True
        winEffect(Button3, Button6, Button9)

```

```

End If

If Not Button1.Text.Equals("") AndAlso
Button1.Text.Equals(Button5.Text) AndAlso
Button1.Text.Equals(Button9.Text) Then
    win = True
    winEffect(Button1, Button5, Button9)
End If

If Not Button3.Text.Equals("") AndAlso
Button3.Text.Equals(Button5.Text) AndAlso
Button3.Text.Equals(Button7.Text) Then
    win = True
    winEffect(Button3, Button5, Button7)
End If

If allbuttonsTextLength() = 9 AndAlso win = False
Then
    Label1.Text = "Drawn"
End If

End Sub

Function allbuttonsTextLength() As Integer

    Dim btnsTextLength As Integer = 0

    For Each c As Control In Panel1.Controls

        If c.GetType() = GetType(Button) Then
            btnsTextLength += c.Text.Length
        End If

    Next

    Return btnsTextLength
End Function

Private Sub winEffect(ByVal b1 As Button, ByVal b2 As
Button, ByVal b3 As Button)

    b1.BackColor = Color.Red
    b2.BackColor = Color.Red
    b3.BackColor = Color.Red

```



```

b1.ForeColor = Color.White
b2.ForeColor = Color.White
b3.ForeColor = Color.White

Button1.Enabled = False
Button2.Enabled = False
Button3.Enabled = False
Button4.Enabled = False
Button5.Enabled = False
Button6.Enabled = False
Button7.Enabled = False
Button8.Enabled = False
Button9.Enabled = False
Label1.Text = b1.Text + " Win"
End Sub

Private Sub Button10_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button10.Click
    Button1.Enabled = True
    Button2.Enabled = True
    Button3.Enabled = True
    Button4.Enabled = True
    Button5.Enabled = True
    Button6.Enabled = True
    Button7.Enabled = True
    Button8.Enabled = True
    Button9.Enabled = True
    Xor0 = 0
    win = False
    Label1.Text = TicTacToePly.textbox1.Text + "'s Turn"
    For Each c As Control In Panel1.Controls
        If c.GetType() = GetType(Button) Then
            c.BackColor = Color.Silver
            c.Text = ""
        End If
    Next
End Sub

```

```
Private Sub Button11_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button11.Click  
    Me.Close()  
    TicTacToeStrt.Show()  
End Sub  
  
Private Sub Panel3_Paint(sender As Object, e As  
PaintEventArgs) Handles Panel3.Paint  
    Panel3.BackColor = Color.FromArgb(120, 0, 0, 0)  
End Sub  
  
Private Sub Panel2_Paint(sender As Object, e As  
PaintEventArgs) Handles Panel2.Paint  
    Panel2.BackColor = Color.FromArgb(120, 0, 0, 0)  
End Sub  
End Class
```

PUZZLE

```
Public Class PuzzleGame
```

```
    Public count = 1, elapsed As Integer
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Button1.Click
```

```
        '2 5
```

```
        cblev1(Button1, Button2)
```

```
        cblev1(Button1, Button5)
```

```
        checkSolved()
```

```
    End Sub
```

```
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Button2.Click
```

```
        '1 6 3
```

```
        cblev1(Button2, Button1)
```

```
        cblev1(Button2, Button6)
```

```
        cblev1(Button2, Button3)
```

```
        checkSolved()
```

```
    End Sub
```

```
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Button3.Click
```

```
        '2 7 4
```

```
        cblev1(Button3, Button2)
```

```
        cblev1(Button3, Button7)
```

```
        cblev1(Button3, Button4)
```

```
        checkSolved()
```

```
    End Sub
```

```
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Button4.Click
```

```
        '3 8
```

```
        cblev1(Button4, Button3)
```

```
        cblev1(Button4, Button8)
```

```
        checkSolved()
```

```
    End Sub
```

```
    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Button5.Click
```

```
        '1 6 9
```

```
        cblev1(Button5, Button1)
```

```
        cblev1(Button5, Button6)
```

```
        cblev1(Button5, Button9)
```

```
        checkSolved()
```

```
    End Sub
```

```
    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Button6.Click
```

```
        '2 5 10 7
```

```
        cblev1(Button6, Button2)
```

```
        cblev1(Button6, Button5)
```

```
        cblev1(Button6, Button10)
```

```
        cblev1(Button6, Button7)
```

```
        checkSolved()
```

```
    End Sub
```

```
    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Button7.Click
```

```
        '3 6 8 11
```

```
        cblev1(Button7, Button3)
```

```
        cblev1(Button7, Button6)
```

```

        cblev1(Button7, Button8)
        cblev1(Button7, Button11)
        checkSolved()
    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button8.Click
        '4 7 12
        cblev1(Button8, Button4)
        cblev1(Button8, Button7)
        cblev1(Button8, Button12)
        checkSolved()
    End Sub

    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button9.Click
        '5 10 13
        cblev1(Button9, Button5)
        cblev1(Button9, Button10)
        cblev1(Button9, Button13)
        checkSolved()
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        shuffle()
        elapsed = 0
        Timer1.Enabled = True
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick
        elapsed += 1
        If elapsed < 60 Then
            Time.Text = "Time Elapsed: " & elapsed & "sec"
        ElseIf elapsed > 60 And elapsed < 3600 Then
            Time.Text = "Time Elapsed: " & Convert.ToUInt64(elapsed / 60) & "min"
        ElseIf elapsed > 3600 Then
            Time.Text = "Time Elapsed: " & Convert.ToUInt64(elapsed / 3600) & "hr"
        End If
    End Sub

    Private Sub Button18_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button18.Click
        Me.Close()
        PuzzleStrt.Show()
    End Sub

    Private Sub Button17_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button17.Click
        Timer1.Stop()
        shuffle()
        elapsed = 0
        Timer1.Start()
        count = 1
        Label1.Text = "Moves: " & count
    End Sub

    Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button10.Click
        '6 9 11 14
        cblev1(Button10, Button6)
        cblev1(Button10, Button9)
        cblev1(Button10, Button11)
        cblev1(Button10, Button14)
        checkSolved()

```

```

End Sub

Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button11.Click
    '7 10 12 15
    cblev1(Button11, Button7)
    cblev1(Button11, Button10)
    cblev1(Button11, Button12)
    cblev1(Button11, Button15)
    checkSolved()
End Sub

Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button12.Click
    '6 11 16
    cblev1(Button12, Button8)
    cblev1(Button12, Button16)
    cblev1(Button12, Button11)
    checkSolved()
End Sub

Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button13.Click
    '9 14
    cblev1(Button13, Button9)
    cblev1(Button13, Button14)
    checkSolved()
End Sub

Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button14.Click
    '13 10 15
    cblev1(Button14, Button13)
    cblev1(Button14, Button10)
    cblev1(Button14, Button15)
    checkSolved()
End Sub

Private Sub Button15_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button15.Click
    '16 11 14
    cblev1(Button15, Button16)
    cblev1(Button15, Button14)
    cblev1(Button15, Button11)
    checkSolved()
End Sub

Private Sub Button16_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button16.Click
    '12 15
    cblev1(Button16, Button12)
    cblev1(Button16, Button15)
    checkSolved()
End Sub

Private Sub Panel1_Paint(sender As Object, e As PaintEventArgs) Handles Panel1.Paint
    Panel1.BackColor = Color.FromArgb(120, 0, 0, 0)
End Sub
End Class

```

PUZZLE (MODULE)

```
Imports System.Data.OleDb
Module Module1
    Dim provider As String
    Dim datafile As String
    Dim connstring As String
    Dim myconnection As OleDbConnection = New
OleDbConnection
    Sub cblev1(ByRef Button1 As Button, ByRef Button2 As
Button)
        If Button2.Text = Nothing Then
            Button2.Text = Button1.Text
            Button1.Text = Nothing
        End If
    End Sub

    Sub checkSolved()
        If PuzzleGame.Button1.Text = "1" And
PuzzleGame.Button2.Text = "2" And PuzzleGame.Button3.Text =
"3" And PuzzleGame.Button4.Text = "4" And
        PuzzleGame.Button5.Text = "5" And
PuzzleGame.Button6.Text = "6" And PuzzleGame.Button7.Text =
"7" And PuzzleGame.Button8.Text = "8" And
        PuzzleGame.Button9.Text = "9" And
PuzzleGame.Button10.Text = "10" And PuzzleGame.Button11.Text
= "11" And PuzzleGame.Button12.Text = "12" And
        PuzzleGame.Button13.Text = "13" And
PuzzleGame.Button14.Text = "14" And PuzzleGame.Button15.Text
= "15" Then
            provider =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source="
            datafile = "C:\Users\admin\Desktop\New
folder\Gaming HUB(Feb)\Database\GamingHUB.accdb"
            connstring = provider & datafile
            myconnection.ConnectionString = connstring
            myconnection.Open()
            Dim str As String
            str = "Insert into Puzzle([Playername],[Moves])
values (?,?)"
            Dim cmd As OleDbCommand = New OleDbCommand(str,
myconnection)
```

```

        cmd.Parameters.Add(New
OleDbParameter("PlayerName", CType(Clipboard.GetText(),
String)))
        cmd.Parameters.Add(New OleDbParameter("Moves",
CType(PuzzleGame.count - 1, String)))
        cmd.ExecuteNonQuery()
        cmd.Dispose()
        myconnection.Close()
        MsgBox("Wow! You did it in " & PuzzleGame.count
- 1 & " Clicks", vbInformation)
        PuzzleGame.Timer1.Stop()
    End If
    PuzzleGame.Label1.Text = "Moves: " &
PuzzleGame.count
    PuzzleGame.count += 1
End Sub

Sub shuffle()
    Dim i, j, rn, a(15) As Integer
    Dim flag As Boolean
    flag = False
    i = 1
    a(j) = 1
    Do While i <= 15
        Randomize()
        rn = CInt(Int((15 * Rnd()) + 1))
        For j = 1 To i
            If (a(j) = rn) Then
                flag = True
                Exit For
            End If
        Next
        If flag = True Then
            flag = False
        Else
            a(i) = rn
            i = i + 1
        End If
    Loop
    PuzzleGame.Button1.Text = a(1)
    PuzzleGame.Button2.Text = a(2)
    PuzzleGame.Button3.Text = a(3)
    PuzzleGame.Button4.Text = a(4)
    PuzzleGame.Button5.Text = a(5)

```

```
PuzzleGame.Button6.Text = a(6)
PuzzleGame.Button7.Text = a(7)
PuzzleGame.Button8.Text = a(8)
PuzzleGame.Button9.Text = a(9)
PuzzleGame.Button10.Text = a(10)
PuzzleGame.Button11.Text = a(11)
PuzzleGame.Button12.Text = a(12)
PuzzleGame.Button13.Text = a(13)
PuzzleGame.Button14.Text = a(14)
PuzzleGame.Button15.Text = a(15)
PuzzleGame.Button16.Text = ""
End Sub
End Module
```

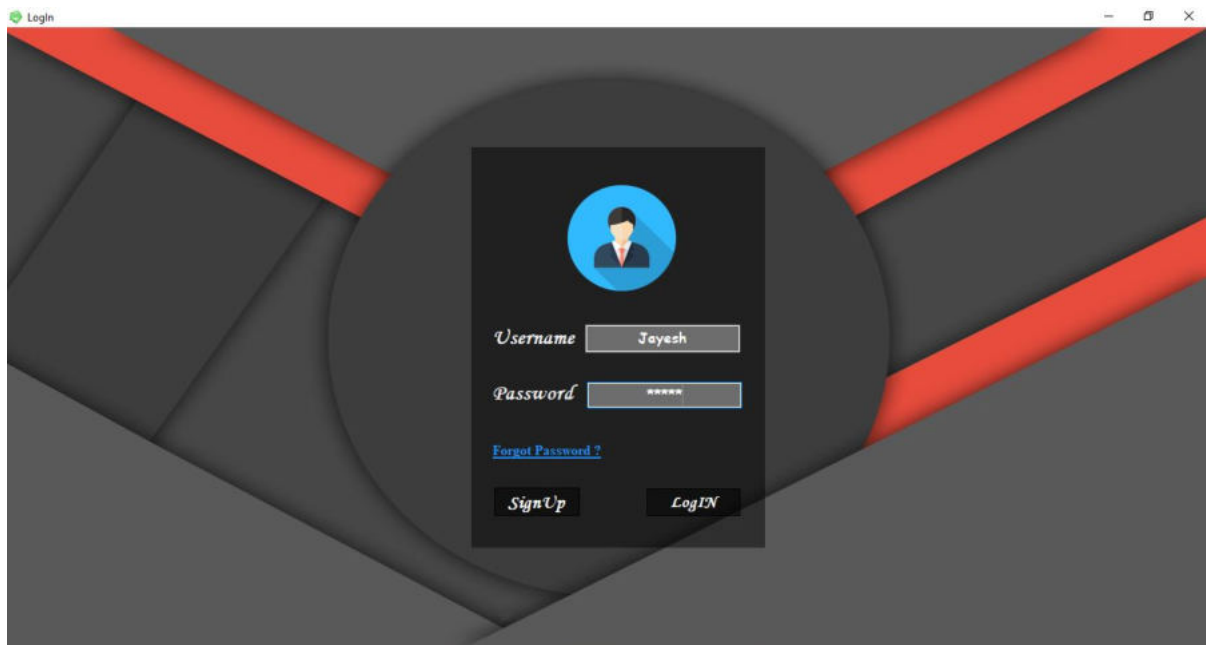

FORGOT PASSWORD

```
Imports System.Data.OleDb
Public Class ForgotPassword
    Private Sub Panel1_Paint(ByVal sender As System.Object,
        ByVal e As System.Windows.Forms.PaintEventArgs) Handles
        Panel1.Paint
        Panel1.BackColor = Color.FromArgb(120, 0, 0, 0)
    End Sub

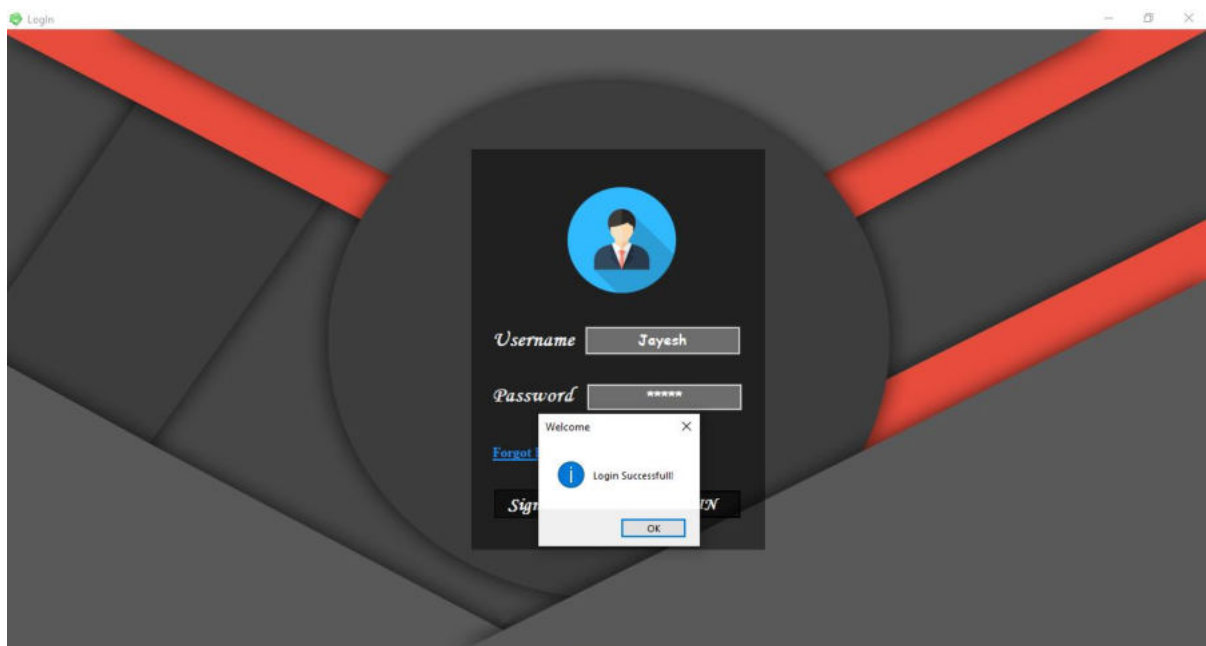
    Private Sub Button2_Click(sender As Object, e As
        EventArgs) Handles Button2.Click
        Me.Hide()
        LogIn.Show()
    End Sub

    Private Sub Button1_Click(sender As Object, e As
        EventArgs) Handles Button1.Click
        Dim con As New
        OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
        Source=C:\Users\admin\Desktop\New folder\Gaming
        HUB(Feb)\Database\GamingHUB.accdb")
        Dim cmd As OleDbCommand = New OleDbCommand("SELECT
        Password FROM Signup WHERE EMail = '" & TextBox1.Text & "'
        AND Question = '" & ComboBox1.SelectedItem & "' AND Answer
        = '" & TextBox2.Text & "' ;", con)
        con.Open()
        Try
            Dim sdr As OleDbDataReader = cmd.ExecuteReader()
            If (sdr.Read() = True) Then
                MessageBox.Show("Password: " &
                sdr("Password").ToString, "Password", MessageBoxButtons.OK,
                MessageBoxIcon.Information)
            Else
                MessageBox.Show("Invalid E-Mail or Security
                Answer", "Error", MessageBoxButtons.OK,
                MessageBoxIcon.Error)
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub
End Class
```

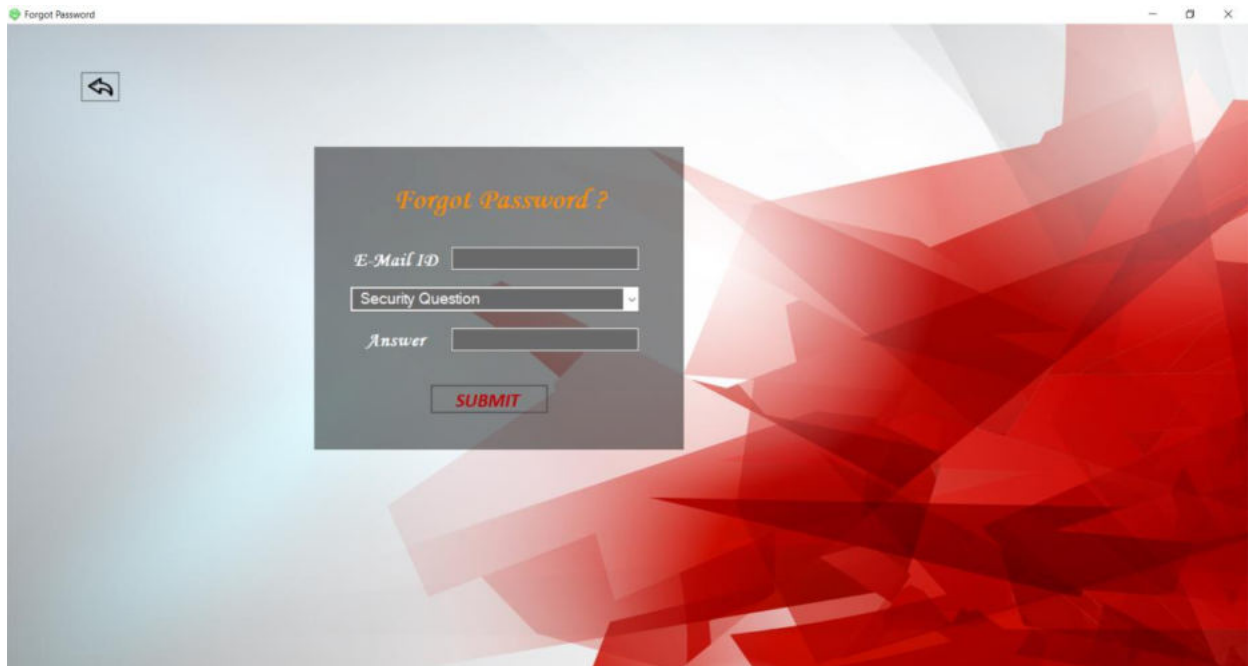
INPUT(LOGIN):



OUTPUT (LOGIN)

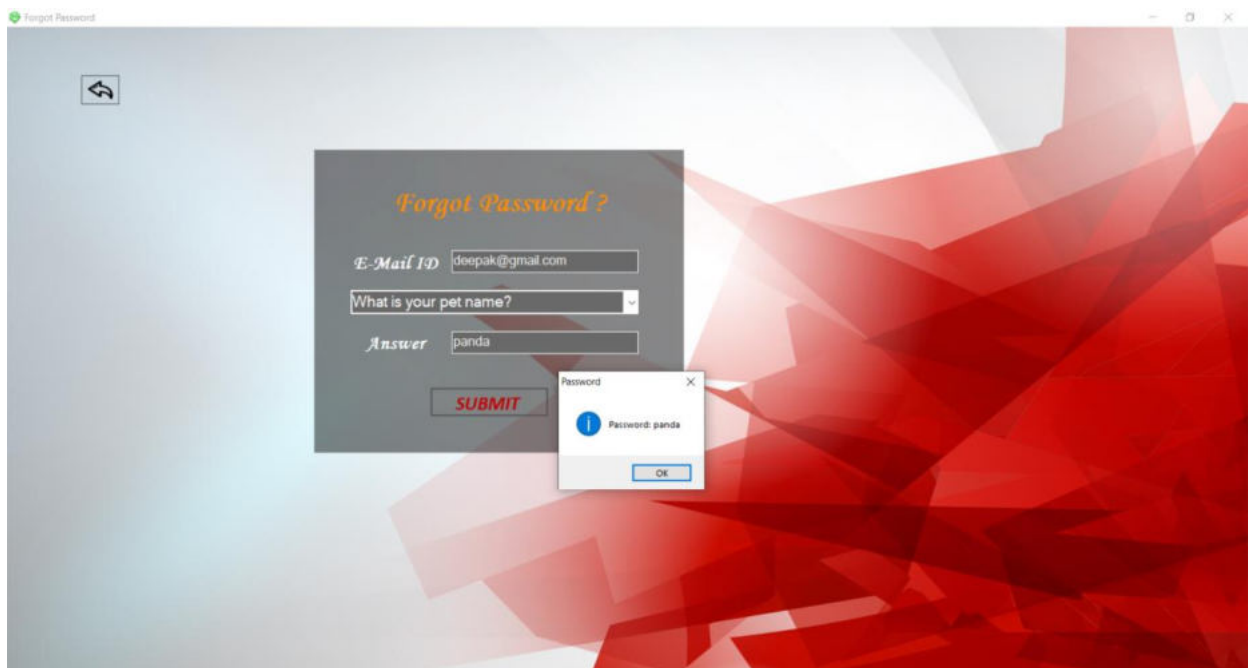


INPUT (FORGOT PASSWORD)



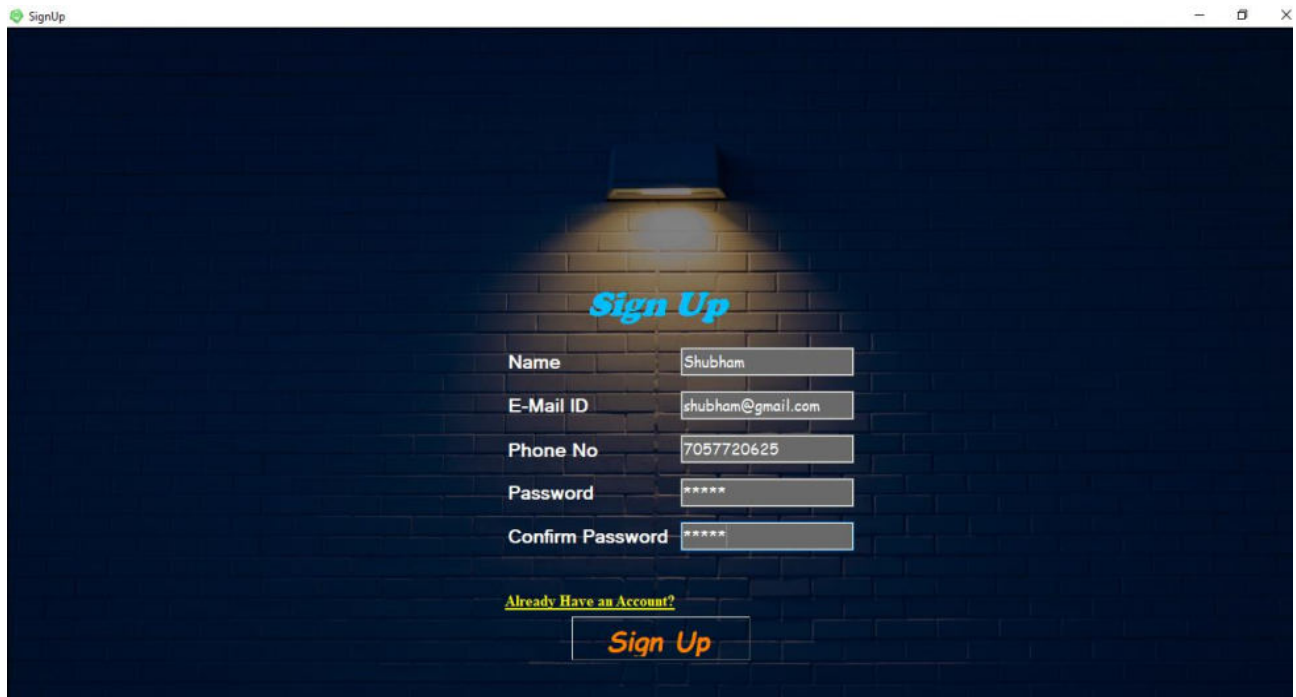
The screenshot shows a web application window titled "Forgot Password". The background features a red and white geometric pattern. In the top left corner, there is a small icon of a house with a circular arrow. The main form is a dark gray box with the title "Forgot Password?" in orange. It contains three input fields: "E-Mail ID" (a text box), "Security Question" (a dropdown menu), and "Answer" (a text box). Below these fields is a red "SUBMIT" button.

OUTPUT (FORGOT PASSWORD)



This screenshot shows the same "Forgot Password?" form as the previous one, but with data entered into the fields. The "E-Mail ID" field contains "deepak@gmail.com", the "Security Question" dropdown is set to "What is your pet name?", and the "Answer" field contains "panda". The red "SUBMIT" button is still visible. A small dialog box titled "Password" is open in the foreground, displaying an information icon, the text "Password: panda", and an "OK" button.

INPUT (SIGNUP)



A screenshot of a web application window titled "SignUp". The background is a dark blue brick wall with a spotlight effect. The "Sign Up" text is in a stylized blue font. Below it, there are five input fields: "Name" (containing "Shubham"), "E-Mail ID" (containing "shubham@gmail.com"), "Phone No" (containing "7057720625"), "Password" (containing "*****"), and "Confirm Password" (containing "*****"). Below the fields, there is a link "Already Have an Account?" and a "Sign Up" button with orange text.

Sign Up

Name Shubham

E-Mail ID shubham@gmail.com

Phone No 7057720625

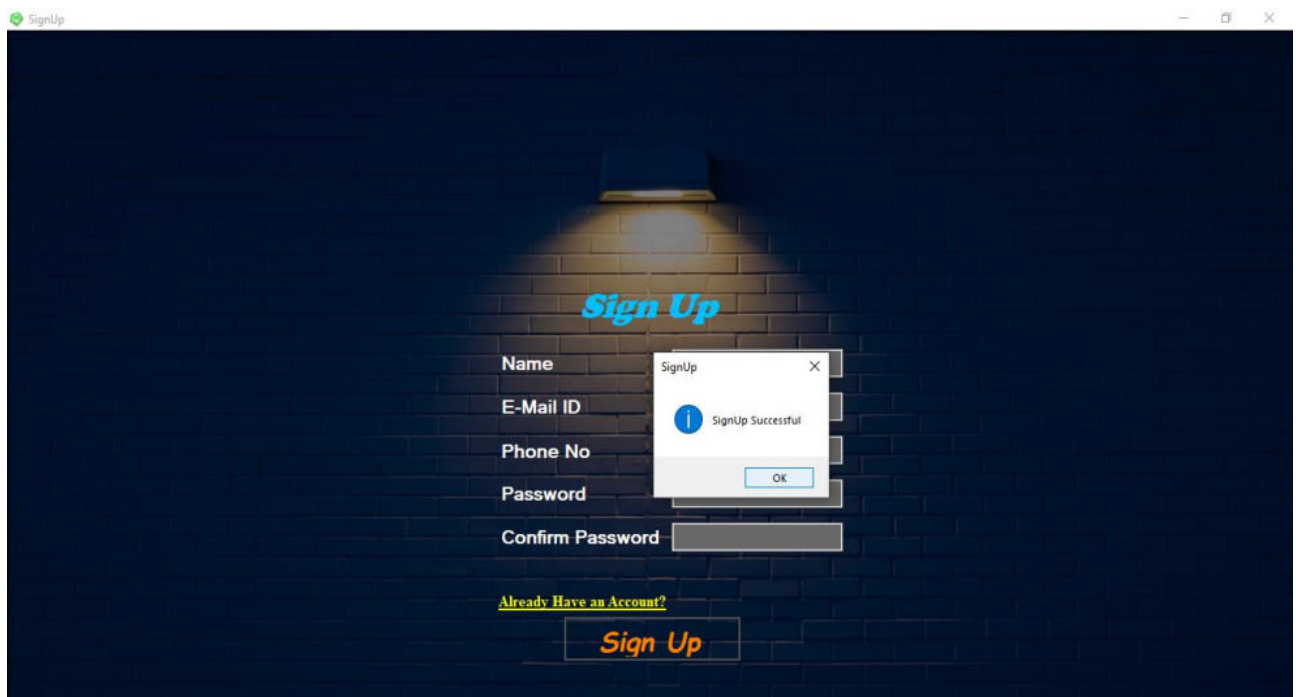
Password *****

Confirm Password *****

[Already Have an Account?](#)

Sign Up

OUTPUT (SIGNUP)



A screenshot of the same web application window, but now a modal dialog box is displayed over the form. The dialog box is titled "SignUp" and contains an information icon, the text "SignUp Successful", and an "OK" button. The form fields and the "Sign Up" button are still visible in the background.

Sign Up

Name Shubham

E-Mail ID shubham@gmail.com

Phone No 7057720625

Password *****

Confirm Password *****

[Already Have an Account?](#)

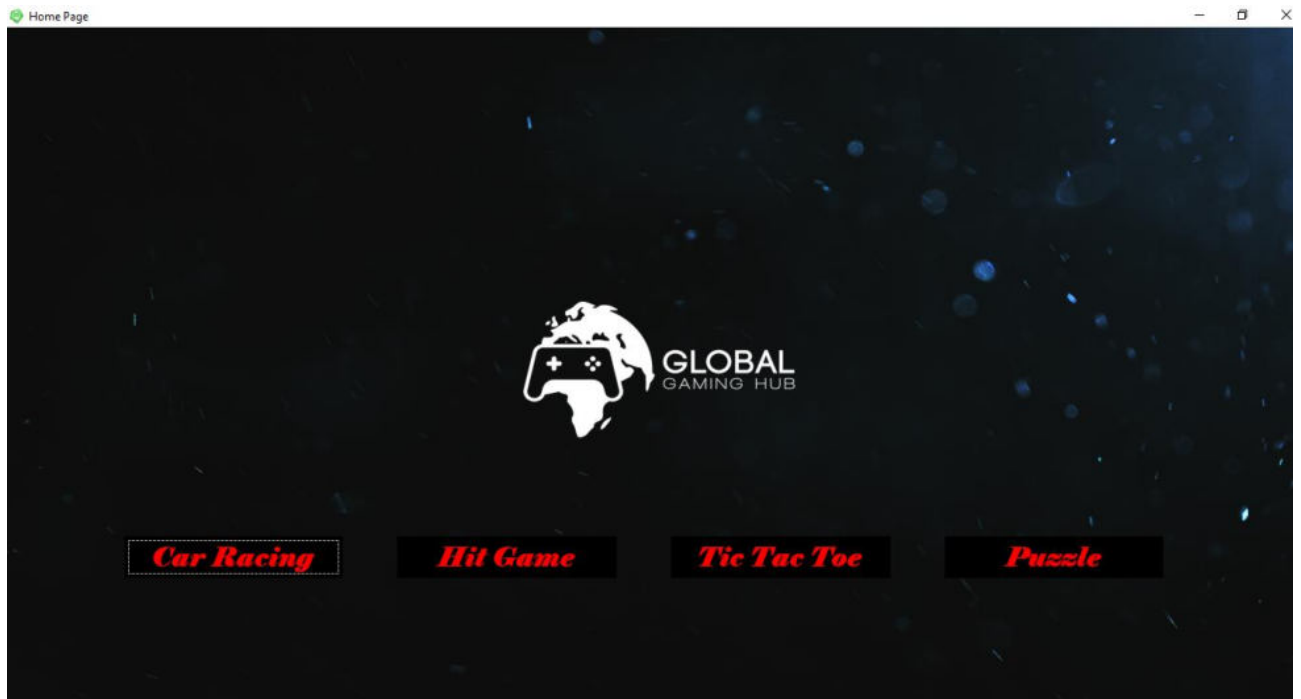
Sign Up

SignUp

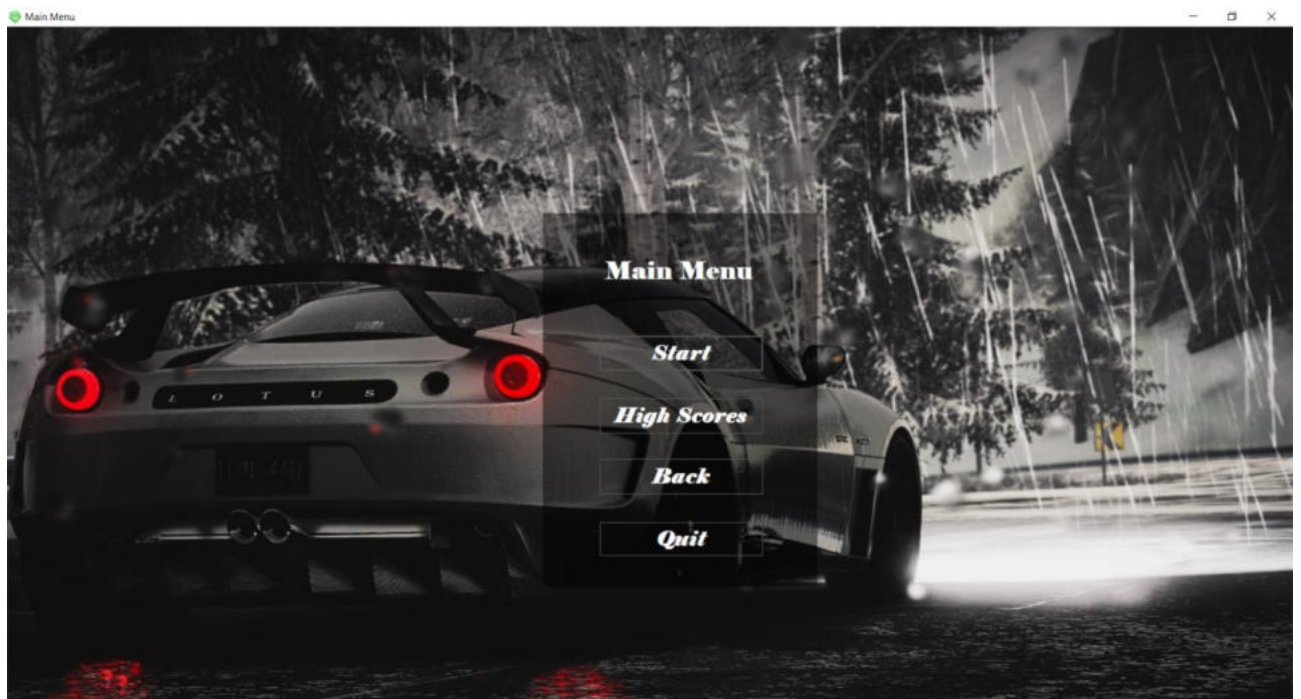
SignUp Successful

OK

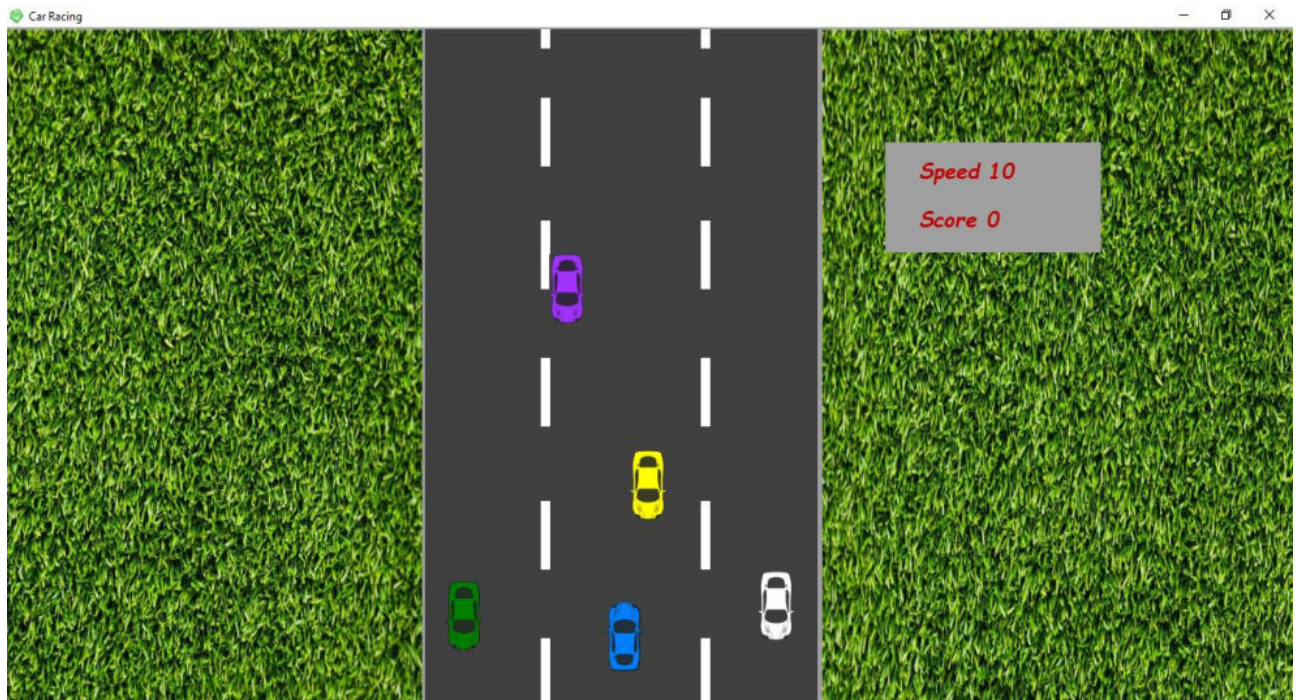
OUTPUT (HOME)



OUTPUT (CAR RACING MAIN MENU)



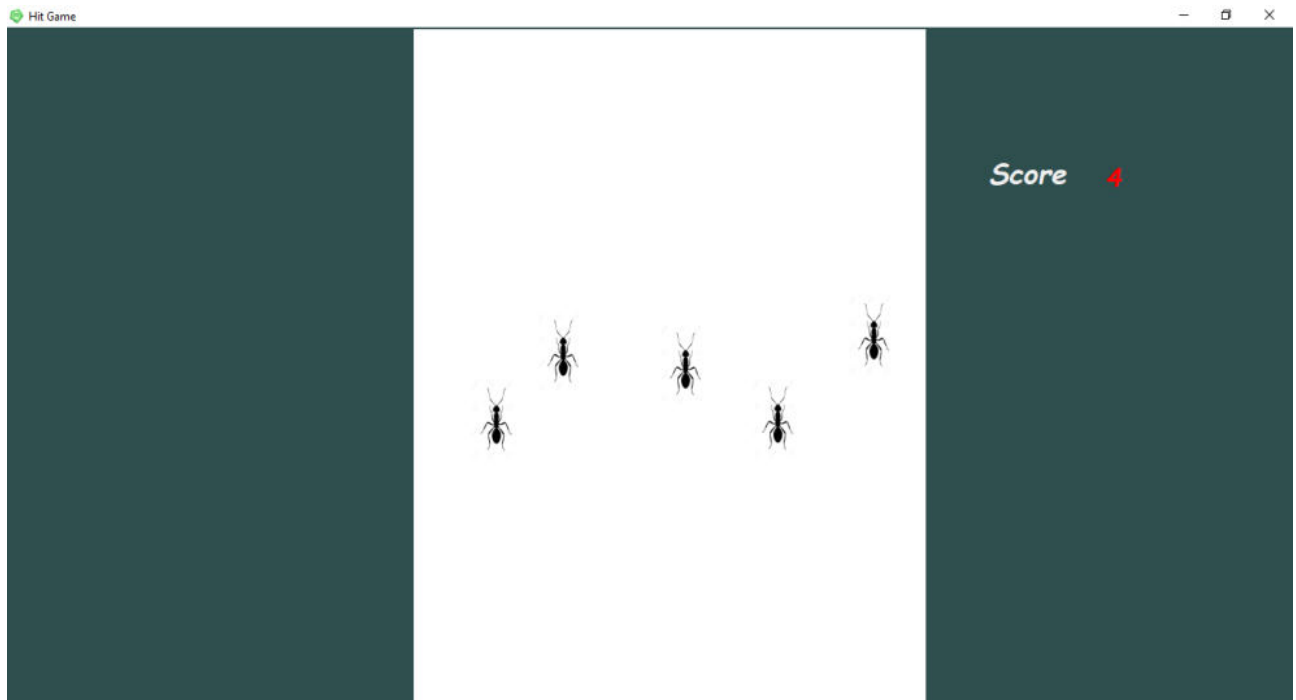
OUTPUT (CAR RACING)



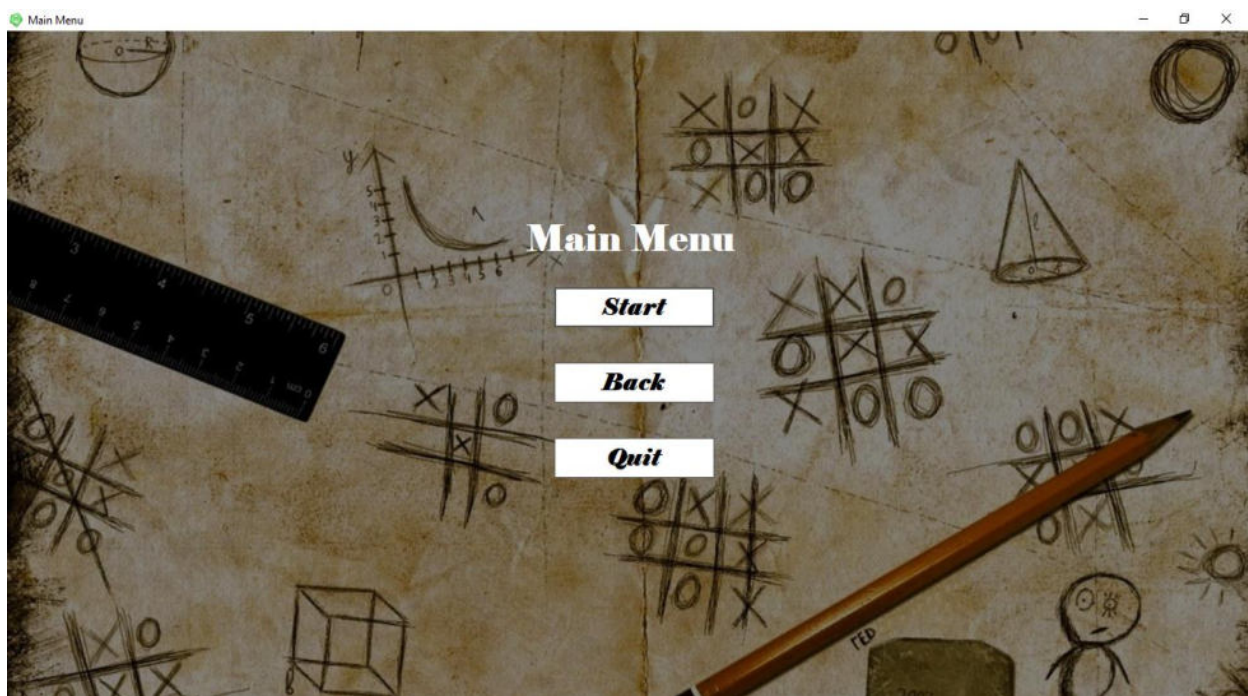
OUTPUT (HIT GAME MAIN MENU)



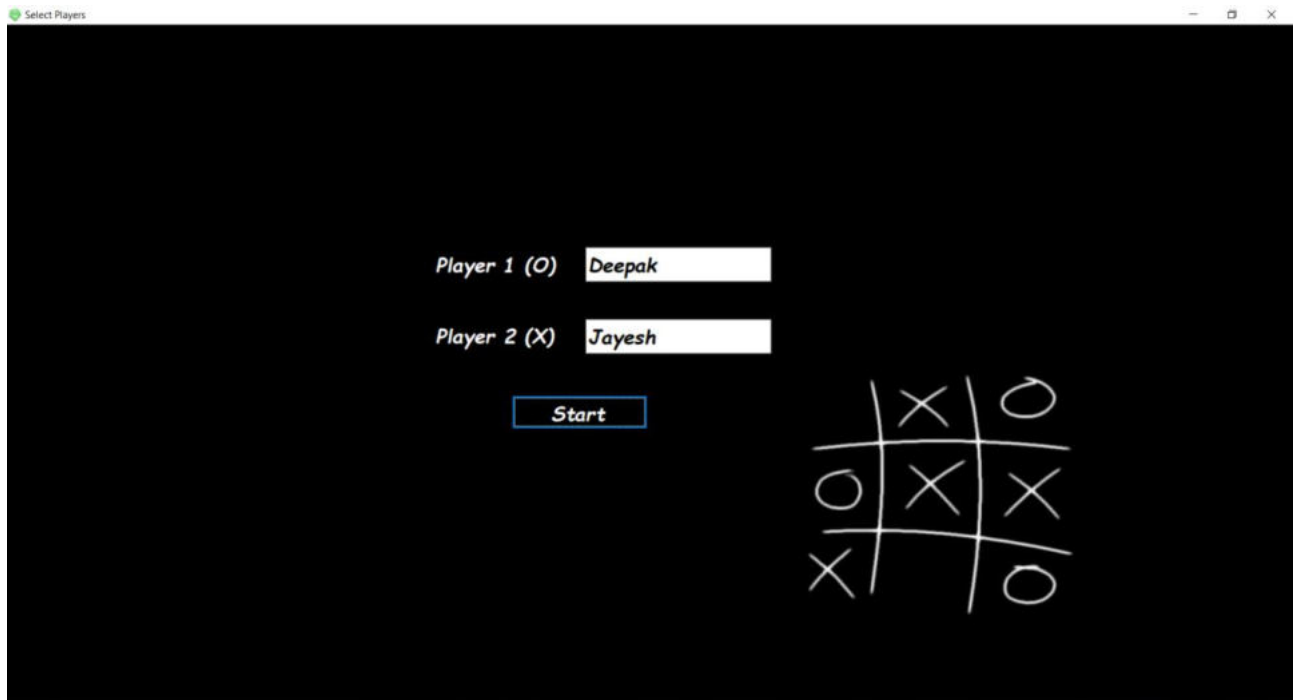
OUTPUT (HIT GAME)



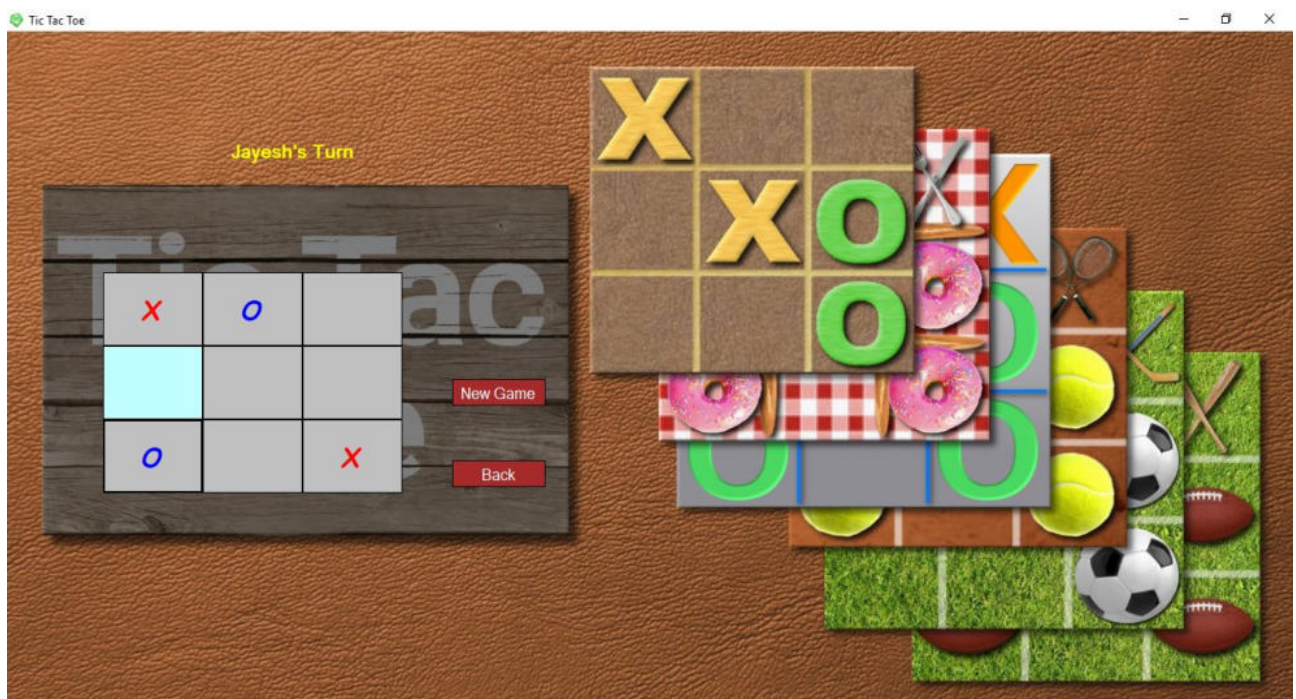
OUTPUT (TIC TAC TOE MAIN MENU)



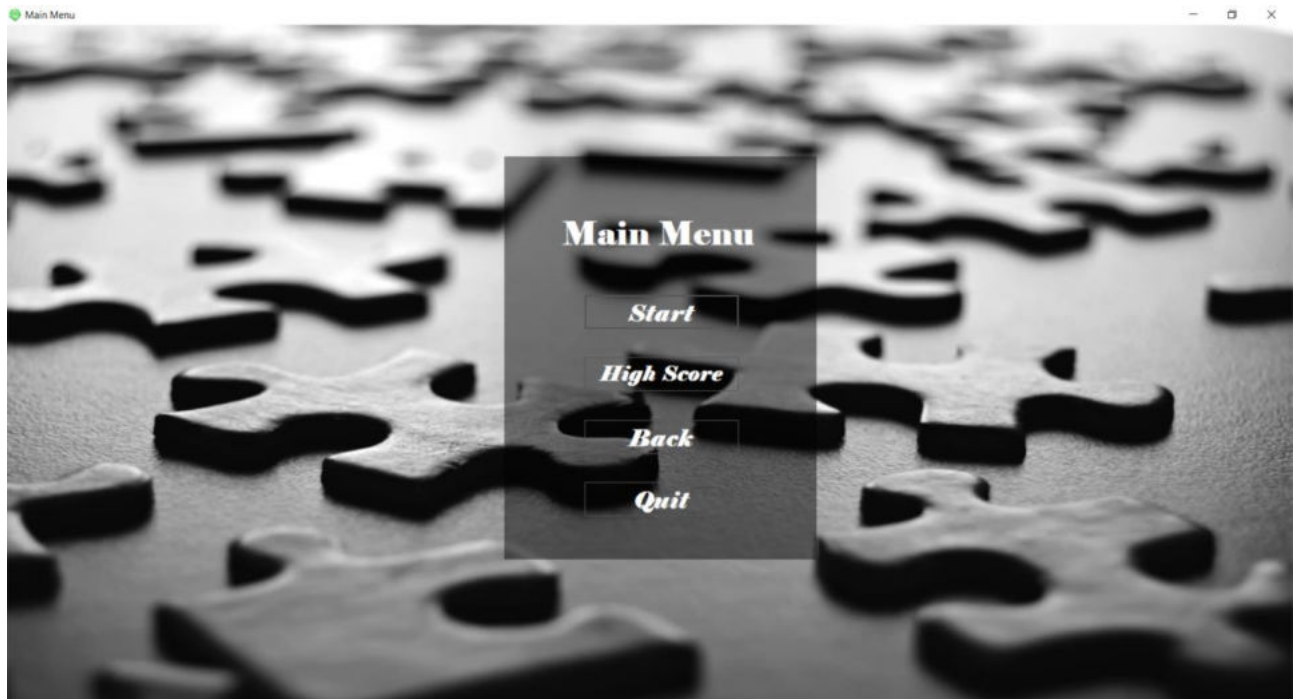
INPUT (SELECT PLAYERS)



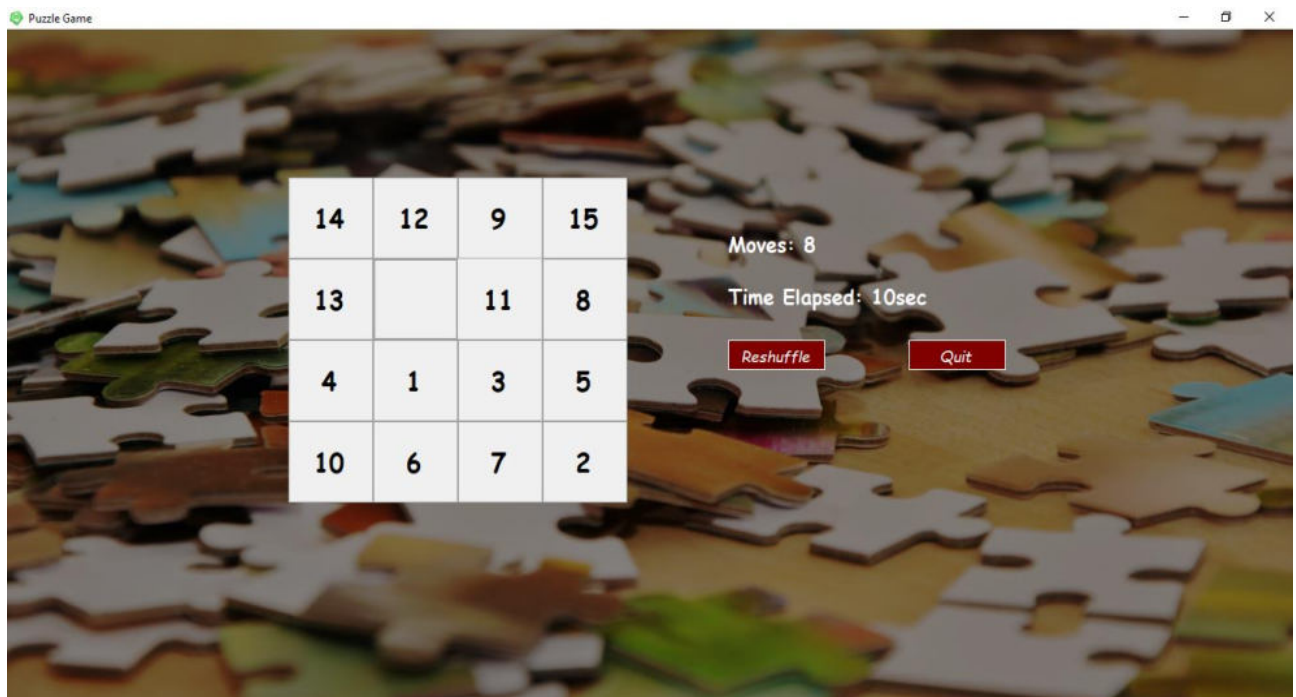
OUTPUT (TIC TAC TOE GAME)



OUTPUT (PUZZLE MAIN MENU)



OUTPUT (PUZZLE GAME)



TESTING AND VALIDATION **CHECKS**

TESTING:

Testing plays important role to identify the quality of any software. Testing actually refers to detecting errors in the system. Before testing can begin, a test plan needs to be developed. Test plan actually includes the type of testing that has to be performed on the code, resources for testing, how the software will be tested. There are several types of testing during the test phase, that includes quality assurance testing (QAT), System Integration testing (SIT), and user acceptance testing (UAT).

- **Quality Assurance (QA) Testing:** In this the procedures and processes are checked. This means whether the instructions are executed as per the user requirements and commands.
- **System Integration Testing (SIT):** It verifies proper execution of software components and proper interfacing between components within the solution. the objective behind this testing is to validate that all software module dependencies are functionally correct and that data integrity is maintained between separate modules for the entire solution.
- **User Acceptance Testing (UAT):** This is the last phase of the software testing procedure. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. UAT is one of the final and critical software project procedures that must occur before newly developed software is rolled out to actual use.

Before testing the system, we need to consider following questions in our mind:

- What is the actual problem?
- How critical the problem is?
- Measures should be taken for the upcoming problems or errors?

Testing gives chance to upgrade or to improve if any drawbacks prevails in the application. Testing is generally done at two levels,

testing of individual modules and

testing entire system.

During system testing, the system is used experimentally to ensure that the software does not fail. that it will run according to its specification and in the way users expect. Testing is done throughout system development at various stages.

Following are the type of testing done in the project:

1. Program Testing: In this, we have to concentrate on the software part, system software should be free from errors. whether it is syntax error or logical error. In this system, we have done software testing and the output of this test is satisfactory. It fulfills all the conditions, which was required for the program testing.
2. Stress Testing: this test is conducted to check the performance of the system in main hours. It finds out how much workload the system can bear. In stress testing of this system, we come to know that this software can work easily and accurately at any condition. The concentration is made on the performance of the system by checking the giving input and their expected outputs.
3. Documentation Testing: this testing work to find out that whatever document supplied is satisfactory or there is a need to supply further document. In this system, all the documents which are supplied are satisfactory.

VALIDATION CHECKS

Data validation is the process of ensuring, at least as far as is possible, that the data given to a program by a user or from a file (essentially, the system's input) is of the correct type, and in the correct format.

There are however measures that can be taken to restrict the program's input to valid data. such measures involve the application of validation rules to any data being input to the program. In this system, Data validation rules can also make this system more user friendly, since they enable the program to warn the user immediately when there is a problem rather than simply allowing them to continue entering data until the program crashes or some other problem occurs.

In this proposed system, we have introduced the following data validation rules:

1. Value entered check: this is used for things like required fields in online forms where the user must enter some data (for example their username and password) and must not leave the field blank.
2. Permitted character check: it is useful for determining whether an input string contains valid characters. For example, a phone number may include the digits 0-9.
3. Limit check: It is used for numeric values that must either be greater than or equal to some lower limit, or less than or equal to some upper limit. For example, the limited number that a user can enter as a phone number is 10.
4. Confirmation check: At the time of creating an account in this system, it is used for determining whether the enter password and confirm password are same or not.
5. Email address check: At the time of creating an account, the system can only accept a valid email id. For example, "@gmail.com".

SYSTEM SECURITY MEASURES

SYSTEM SECURITY MAJORS

The objective of system security is the protection of information and property from theft, corruption and other types of damage, while allowing the information and property to remain accessible and productive. System security includes the development and implementation of security countermeasures. There are a number of different approaches to system security.

In this proposed system, we have provided the following security majors:

1. **Password:** the most widely method to prevent unauthorized access is to use passwords. The password needs to be kept secret and is only intended for the specific user. In this system, each password is associated with a specific username since many individuals may be accessing the same system.
2. **Accessibility:** In the website, with the help of admin username and password, only the admin has the right to update the website such as add, delete and update the products.
3. **System Automation:** In this system, with the use of selenium we can automating web application. Web driver is the successor to selenium, which accepts commands and sends them to a browser.

IMPLEMENTATION, **EVALUATION & MAINTENANCE**

IMPLEMENTATION

Implementation refers to that stage of project during which the theory is turned into practice i.e. converting soft ideas into actual process. In this stage physical system specifications are converted into working and reliable solution. This is where the system is developed. It is followed by testing and then again implementation.

Implementation phases:

- Coding: this includes implementation of the design document into executable programming language code. The output of the coding phase is the source code for the software that acts as input to the testing and maintenance phase.
- Integration and Testing: It includes detection of errors in the software. The testing process starts with a test plan recognizes test-related activities , such as test case generation, testing criteria and resource allocation of testing. The code is tested and mapped against the design document created in the design phase.
- Installation: New system is installed and rolled out.

The steps involved in this phase are:

1. Acquisition and installation of hardware and software.
2. Conversion: It actually means to convert the old data to new format for proper functioning of the application in the new system.

3. User Training: User in this case has to be trained to use the system properly so that it is easy for them to grab control over the use of the application.
4. Documentation: This provides details of how to operate the given software, application and website.

The hardware and relevant software required for running the application must be installed and fully checked before implementation. In this phase conversion plays a crucial role. It actually means to convert the old data to a new format for proper functioning of the application in the new system. During the phase all the required programs are loaded onto user's computer. User must be trained.

The documentation is a complete description of the system from the users point of view as it provides details of how to operate the given software and application. It also includes certain error messages that a user is expected to encounter during its usage and solution to the expected problems. It involves detailed and step by step information of the project development so as to modify or update as per the new user requirements.

EVALUATION

Evaluation phase is the next to the implementation and it evaluates whether or not the system has met its requirements by comparing with the standards that were set before its actual development. The evaluation process includes the study of the current system and their drawbacks (if any) and various alternatives to improve and solve those prevailing problems. Evaluation is done by keeping the preliminary requirements of the user in mind.

Evaluation is included as a part of the final phase, but practically, evaluation takes place during each and every phase. The concentration should be on the satisfying the primary requirement of the users. The system is evaluated on the basis of following points:

- System Availability: whether the required system is available or not.
- Compatibility: whether the application is compatible with the system or not.
- Cost: whether the developed application is affordable and has low maintenance cost.
- Performance: it basically checks the efficiency of the application. Efficiency in handling the rush and fired queries simultaneously. It evaluates whether the application generates result at same speed when load is given to it as when it is stress free.
- Usability: whether the developed application is easily accessible and user-friendly.

Evaluation in this system is done as follows:

- The errors generated in the code due to compatibility issues are debugged.
- Ease of installation and training.
- Adequacy and cost of hardware maintenance.
- Performance and its efficiency to handle the stress.
- Low maintenance cost.

In this proposed system, evaluation is made on existing system, what are their drawbacks what improvement can be made to provide facility to users. Collecting the information required for improvement in the project and then implementing it in real use

MAINTENANCE

Maintenance is the final stage after the development process. After the system is installed, it must be maintained means that the computer programs must be modified and kept up to date. The average amount of time spent on maintenance is 60% of the total time. Estimates of the time spent by departments on maintenance have ranged from 48 to 60 percent of the total time spent developing systems. As the number of programs written increases, so does the amount of maintenance they require.

Maintenance covers a wide range of activities including correcting, coding, designing errors and updating user support. the project needs maintenance in further if any enhancements are made, maintenance of the hardware and software is also required.

The maintenance phase occurs once the system is operational. It includes implementation of changes that software might undergo over a period of time, or implementation of new requirements after the software is deployed at the customer location. The maintenance phase also includes handling the residual errors that may exist in the software even the testing phase.

The maintenance phase also monitors system performance, rectifies bugs and requested changes are made.

Maintenance is performed for two reasons:

- **First** is to correct software errors. no matter how thoroughly the system is tested, bugs or errors creep into the computer programs. Bugs in commercial PC software are often documented as “known anomalies” and are corrected when new versions of the software are released or in an interim release. In custom software (also called bespoke software), bugs must be corrected as they are detected.

- **Second**, for performing system maintenance is to enhance the software's capabilities in response to changing organizational needs, generally involving one of the following three situations:

1. Users often request additional features after they become familiar with the computer system and its capabilities.
2. The business changes over time.
3. Hardware and software are changing at an accelerated pace.

In summary, Maintenance is an ongoing process over the life cycle of a system. After the application is installed, maintenance usually takes the form of correcting previously undetected program errors. Once these are corrected, the system approaches a steady state, providing dependable service to its users.

FUTURE SCOPE OF THE PROJECT

FUTURE SCOPE OF THE PROJECT

- **User Friendly:**

The system could be made flexible in nature and method to record any data could be modified. Thus, the system can be made more friendly by modifying certain areas in the system.

- **Flexible:**

The basic scope of this application is to make user flexible with the system.

- **Extra Features:**

Success of this application provides the extra ordinary features to the user.

- **Reliability:**

This application can be run or expected into the current operating software's also.

- **Interactive GUI:**

This application provides the data into the graphical mode for elements with the help of VB.NET.

- **Simple and Easy:**

This project procedure is user-friendly, easy, simple and error free, it makes the software attractive.

CONCLUSION

CONCLUSION

- **Usefulness:**

The project is useful for all the persons no matter what the age or gender of the person is.

- **Refreshing and Enhances Mind Power:**

This project is made basically to refresh the mind of people as well as it also enhances the concentration power of the user.

- **Premium Quality:**

The main motive behind developing this project is to provide good quality software.

- **Fulfilment of User Satisfaction:**

The basic concern behind developing this software is to provide good quality of website and provide high level of satisfaction to the user.

BIBLIOGRAPHY

BIBLIOGRAPHY

References: -

- 1) <http://www.tutorialspoint.com/>**
- 2) <https://www.seleniumhq.org/download/>**
- 3) [www.Google.co.in](http://www.google.co.in)**
- 4) YouTube**

**A
PROJECT SYNOPSIS
ON
“ Gaming HUB”**

**Submitted to
Rashtrasant Tukadoji Maharaj Nagpur University,
NAGPUR
In the Partial Fulfillment of
B.Com. (Computer Application) Final Year**

**Synopsis Submitted by
Jayesh S. Manapure
Deepak Kumar**

**Under the Guidance of
Pravin J. Yadao**



**G. S. College of Commerce & Economics
Nagpur
2019-2020**

1. Introduction:

The Gaming Hub is a collection of various kinds of interesting games. This includes game such as : **Tic Tac Toe , Puzzle , Hit Game & Car Racing.**

These games are structured form of play, usually used for enjoyment and also used as an educational tool. It consists of FOUR games and the user can choose a game which he wants to play.

2. Objectives of the project:

1. For Fun and refreshment.
2. To play with friends (Multiplayer).
3. Puzzle helps to increase concentration, thinking ability.
4. To earn Money.

3. Project Category: Application

4. Tools/ Platform/ Languages to be used: Visual Basic.Net, MS Access

5. Scope of future application:

1. Introduce these old games to new generation.
2. Spread worldwide.
3. To organize competition among top players.
4. Platform independent.

Submitted by,

Jayesh S. Manapure

Deepak Kumar

Approved by,

Prof. Pravin Yadao

Project Guide