

**Dr.Babasaheb Ambedkar Marathwada University, Aurangabad**



A MAJOR PROJECT REPORT ON :

**“ DATA SECURITY ”**

**Guided By:**

**Mrs.PRIYANKA HIWRALE  
Mrs.PALLAVI SALUNKE**

*In partial Fulfilment for the  
Degree Of B.Sc.(Computer  
Science)*

*in Faculty Of Science*

**Submitted By:**

**GOVIND MADAN AMBADE**

**PRAJWAL MOHAN CHAPKE**

**VIKAS GAJANAN KAKDE**

**Submitted To:**



**Vivekanand Arts, Sardar Dalip Singh Commerce & Science College, Aurangabad  
Academic Year 2021-22**

## VIVEKANAND SHIKSHAN SANSTHA'S



### **VIVEKANAND ARTS, SARDAR DALIPSINGH COMMERCE & SCIENCE COLLEGE, AURANGABAD**

#### CERTIFICATE

This is to certify that a major project report entitled "**DATA SECURITY**" has been successfully completed and submitted by **GOVIND MADAN AMBADE , PRAJWAL MOHAN CHAPKE , VIKAS GAJANAN KAKDE** as per requirement of Dr.Babasaheb Ambedkar Marathwada University, Aurangabad , in fulfillment of **Bachelor of Computer Science 3rd Year for Academic Year 2021-22.**

**(Guided By)**

**Mrs.Priyanka Hiwrale**

**Mrs.Pallavi Salunke**

**(Head of Dept)**

**Dr.Ashok Gaikwad**

**Examiner**

## **ACKNOWLEDGEMENT**

With the complete work undertaken. I would like to take this opportunity to thank all those who helped me to complete this project with their guidance, co-operation and support. This project would not have been materialized without the help from many quarters.

I would like to thanks **Dr. Ashok T. Gaikwad (Head of Dept), Mr Narayan Kulkarni (Co-Ordinator), Mrs.Priyanka Hiwrale & Mrs.Pallavi Salunke** and all my faculty members.

I also cannot forget thanking my classmates for providing the friendly and cordial environment to achieve the goal. Without their valuable contribution this project would not has been a success.

At last but not the least I owe my success in accomplishing this task of completing the project successfully to my parents and family member who by one or more ways have encouraged me to carry out the project and put in the best of my efforts.

**Submitted by:**

**GOVIND MADAN AMBADE, Roll No: 05, Div : A**  
**PRAJWAL MOHAN CHAPKE, Roll No: 18, Div : A**  
**VIKAS GAJANAN KAKDE, Roll No: 43, Div : A**

# INDEX

<b>SR.NO</b>	<b>CONTENT</b>	<b>PAGE NO</b>
	<b>A. INTRODUCTION</b>	
	University Certificate	1
	College Certificate	2
	Acknowledgement	3
1.	Introduction Of Project	5
2.	Scope Of Work	6
3.	Proposed System	7
	<b>B.SYSTEM ANALYSIS</b>	
1.	Hardware And Software Requirements	8
2.	Existing System & Limitations	9
3.	Need For The System	10
4.	Objective Of System	11
	<b>C.FEASIBILITY REPORT</b>	
1.	Technical Feasibility	12
2.	Operational Feasibility	12
3.	Economic Feasibility	12
	<b>D.SELECTED SOFTWARE</b>	
1.	C# : Microsoft Visual Studio	13
2.	Microsoft SQL Server	14
	<b>E.SYSTEM DESIGN</b>	
1.	Introduction	15
2.	E-R Diagram	16
3.	Data Flow Diagram	
	<b>F.ANALYSIS &amp; REPORT</b>	
1.	Testing	17
2.	Strategic Approaches to Software Testing	18
3.	Source Code	19
4.	Output Screens	99
	<b>G.CONCLUSION</b>	108
	<b>H.BIBLIOGRAPHY</b>	109

## **A.INTRODUCTION**

### **1. Introduction Of Project**

**Data security** is the practice of protecting digital information from unauthorized access, corruption, or theft throughout its entire lifecycle. It's a concept that encompasses every aspect of information security from the physical security of hardware and storage devices to administrative and access controls, as well as the logical security of software applications. It also includes organizational policies and procedures.

When properly implemented, robust data security strategies will protect an organization's information assets against cybercriminal activities, but they also guard against insider threats and human error, which remains among the leading causes of data breaches today. Data security involves deploying tools and technologies that enhance the organization's visibility into where its critical data resides and how it is used. Ideally, these tools should be able to apply protections like encryption, data masking, and redaction of sensitive files, and should automate reporting to streamline audits and adhering to regulatory requirements.

**Deadlock** is a full-suite data security solution for Windows 7, Windows 8, Windows 10 and Windows

11. In **Deadlock** lets you lock, hide, Encrypt and password-protect files, folders and drives.

It is a Multi-user software. **Deadlock** contains two type of accessing modes, Administrator and User.

Administrator has full rights for accessing software.

In User mode User have a specific rights provided by administrator. Basic rights of User as follows:

Lock Folder , Hide folder, Encrypt Folder etc.

## **2. Scope Of Work**

The future scope of this project is very broad

Few of them are :-

- ✓ **Deadlock** is the most comprehensive data protection suite with seven (6) tools; Lock Files, Encrypt Files, Disable USB, Hide Files, Lock Drive And Hide Drive.
- ✓ Option to either Encrypt or Lock, choices useful for both power users and casual users. Those who want ultimate security can choose encryption. Those who want ultimate speed can choose locking
- ✓ Use Hack Attempt Monitoring feature to maximize protection against incorrect password attempts. Trigger **Deadlock** to Exit Application, Log-Off Windows, and System Shutdown upon multiple incorrect password attempts.

### **3. Proposed System**

The main goal of the system is secure data from friends ,co-worker ,family and data-theft. Some Of goals of the system are listed below:

- ◆ Encrypt Or Decrypt Folders
- ◆ Lock Or Unlock Folders / Drives
- ◆ Hide Or Un-Hide Folders / Drives
- ◆ Block USB Devices
- ◆ Block New Application Install
- ◆ Protect Folder And Files from Modifying or Deleting
- ◆ Two Step Verification

## **B.SYSTEM ANALYSIS**

### **1. System Requirements**

#### **Hardware Configuration**

- Memory :- 4 GB RAM
- Hard Disk :- 30 GB
- Processor :- Intel I-3 and Above

#### **Software Configuration**

- Front End :- Microsoft Visual Studio ( C# )
- Back End :- Microsoft SQL Server
- O.S :- Windows 7 or Above

## **2.Existing System & Limitations:**

- User friendliness is provided in the application with various controls provided by system rich user interface.
- The system makes the overall project Experience much easier and flexible.
- The user information can be stored in centralized database which can be maintained by the system.
- This can give good security for user information because data is not in client machine.
- There is no risk of data management at any level while the project development is under process.

### **3. Need For The System**

Think about companies like Facebook, Dropbox, Google, Microsoft - Now think about how much you trust them with your data knowing full well that millions of records are breached by hackers and third-party apps every year.

If you don't want to think that far; think about your co-workers, your boss or even your spouse who is curious about what you do online or what type of files you have on your Computer.

Now picture yourself going about your usual day mindlessly saving passwords in a spreadsheet you keep in 'Documents' folder, surfing scam websites and unknowingly installing malware.

Files and folders left unprotected can lead to theft, data-loss, and breaches of your privacy, costing you embarrassment, money, lost data, and maybe even your job!

## **4.Objective Of System**

- ◆ The main objective of the system intended for people who wants to lock, hide and password-protect your personal pictures, videos, private files, etc. from family and friends who would share your computer or have a prying eye on you, you need to browse those files in Lock Folder.
- ◆ Integrity is aimed at protecting data from modification by unauthorized users and improper modification by authorized users.
- ◆ Keep data safe.
- ◆ It's user-friendly

## **C.FEASIBILITY REPORT**

- After doing this Data Security project in C#, we study and analysing all the existing or required functionalities of the system, the next task is to do that the feasibility study for the project. All projects are feasible – given unlimited resources and infinite time. Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be that future changes can be easily done based on the future upcoming requirements.

### **1.Economical Feasibility:-**

This is a very important aspect to be considered while developing a project.

We decided the technology based on minimum possible cost factor.

1. All hardware and software cost has to be borne by the organization.
2. Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs.

### **2.Technical Feasibility:-**

This includes the study of function, performance and constraints that may affect the ability to achieve to an acceptance system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification(SRS), and checked if everything was possible using different type of fronted and backend platform.

### **3.Operational Feasibility:-**

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken are self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system.

## **D.SELECTED SOFTWARE**

### **➤ C# :-**

C# is a general purpose object oriented programming language with multiple paradigms. It was designed for Common Language Infrastructure (CLI) in 2000 by Microsoft. for its .NET framework and also approved by ECMA(European Computer Manufacturers Association ) and ISO(International Organization for Standardization ).

C# was originally developed as C-Like Object Oriented Language (COOL) but this name was later changed to avoid any trademark issues. This language has been influenced by various languages such as Java, Pascal, C++, Eiffel etc.

C# has multiple points that make web development solutions faster as well as easier. Some of these are garbage collection, scalability support, type safety, easier type declarations etc.

### **Some of the major features of C# are as follows:**

#### **1. Object Oriented Language**

C# is an object oriented language. This means that development is easier if there are long codes in large projects as compared to procedure oriented programming languages.

#### **2. Structured programming language**

The programs in C# can be structured by breaking them into smaller parts using functions. This makes the program easier to understand.

#### **3. Simple to use**

It is quite simple to use C# as it has various features and functionalities. Moreover, it provides a structured approach that makes the program easier to understand.

#### **4. Scalable language**

C# is an automatically scalable as well as updatable language. The old files are regularly updated and replaced with new ones.

There are many applications that can be created using C#. Some of these are given as follows:

1. Window applications
2. Web applications
3. Database applications
4. Distributed applications
5. Web service applications

➤ **Microsoft SQL Server**

SQL Server is a relational database management system (RDBMS) developed by Microsoft. It is primarily designed and developed to compete with MySQL and Oracle database. SQL Server supports ANSI SQL, which is the standard SQL (Structured Query Language) language. However, SQL Server comes with its own implementation of the SQL language, T-SQL (Transact-SQL).

T-SQL is a Microsoft propriety Language known as Transact-SQL. It provides further capabilities of declaring variable, exception handling, stored procedure, etc.

SQL Server Management Studio (SSMS) is the main interface tool for SQL Server, and it supports both 32-bit and 64-bit environments.

## **E.SYSTEM DESIGN**

### **● INTRODUCTION**

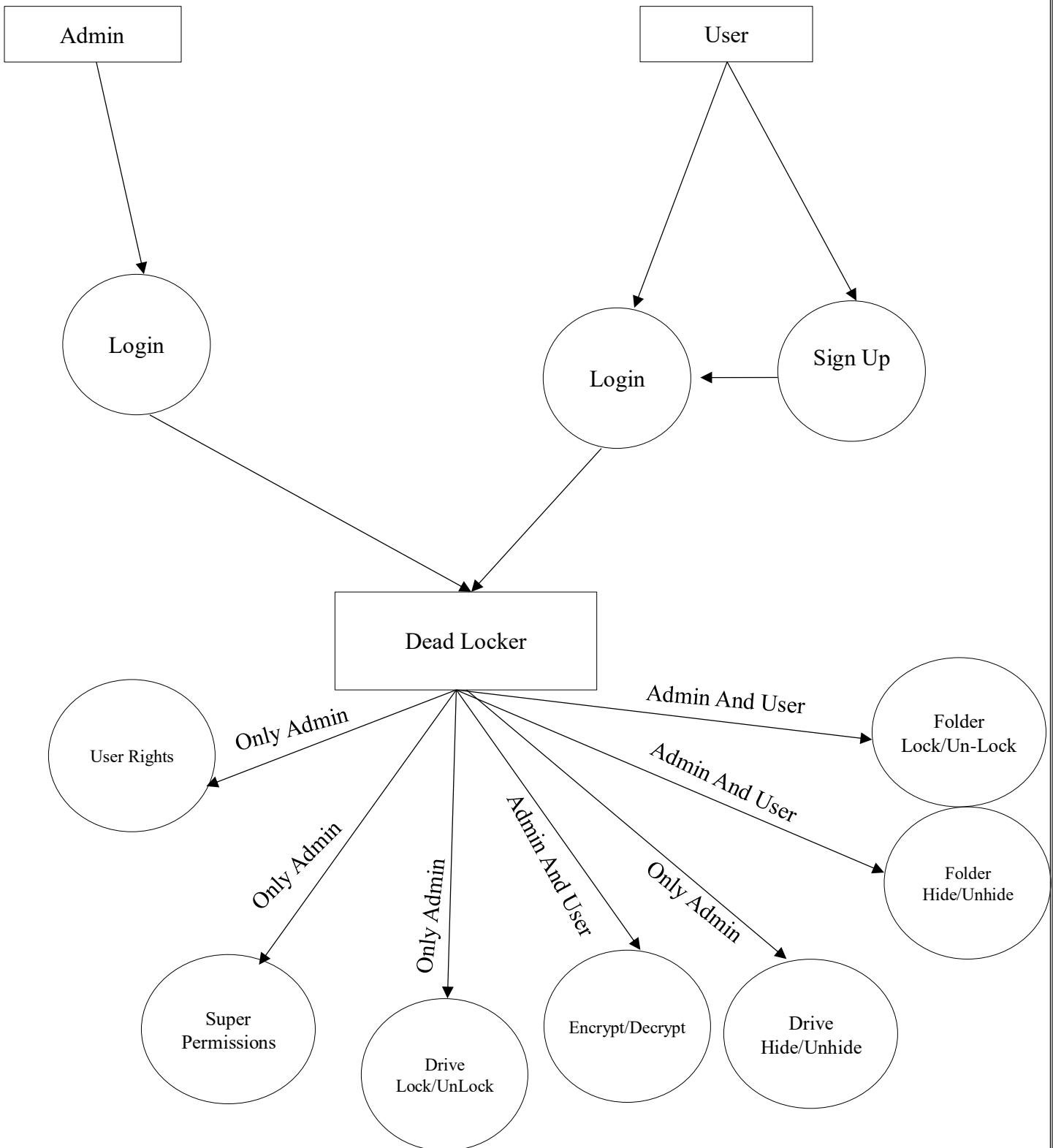
In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the client's requirements into A logically working system. Normally, design is performed in the following in the following two steps:

- Primary Design Phase: In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimizing the information flow between blocks. Thus, all activities which require more interaction are kept in one block.
- Secondary Design Phase: In the secondary phase the detailed design of every block is performed.

**The general tasks involved in the design process are the following:**

- ➔ Design various blocks for overall system processes.
- ➔ Design smaller, compact and workable modules in each block.
- ➔ Design various database structures.
- ➔ Specify details of programs to achieve desired functionality.
- ➔ Design the form of inputs, and outputs of the system.
- ➔ Perform documentation of the design.
- ➔ System reviews.

## ER-DIAGRAM



## **F.ANALYSIS & REPORT**

### **➤ Testing :-**

#### **◦ Introduction:-**

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing presents an interesting anomaly for the software engineer.

#### **Testing Objectives include:**

- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has a probability of finding an as yet undiscovered error
- A successful test is one that uncovers an undiscovered error

#### **Testing Principles:**

- All tests should be traceable to end user requirements.
- Tests Should be planned long before testing begins
- Testing should begin on a small scale and progress towards testing in large
- Exhaustive testing is not possible
- To be most effective testing should be conducted by an independent thirdparty.

## **Strategic Approaches to Software Testing :-**

- Testing is a set of activities that can be planned in advance and conducted systematically.
- For this reason a template for software testing- a set of steps into which you can place specific test case design techniques and testing methods- should be defined for the software process.
- A number of software testing strategies have been proposed in the literature.
- All provide you with a template for testing and all have the following generic characteristics:
  - By doing this, many errors will be eliminated before testing Commence. Testing begins at the component level and works “outward” toward the integration of the entire computer-based system.
  - Different testing techniques are appropriate for different software engineering approaches and at different points in time. Testing is conducted by the developer of the software and an independent test group. Testing and debugging are different activities, but debugging must be accommodated in any testing any testing strategy.
  - To perform effective testing, you should conduct effective technical reviews.

## ● Source Code

### ❖ LOGIN.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Media;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class Login : Form
    {
        SqlConnection con = new SqlConnection("Data Source=DESKTOP-
9QTIITB\\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();
        public Login()
        {
            InitializeComponent();
        }
        public string str_, str1_;
        public int count_, count1_;
        public SoundPlayer Player = new SoundPlayer();

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void guna2ToggleSwitch1_CheckedChanged(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {
```

```

    }

private void twostep()
{
    if(guna2ToggleSwitch1.Checked)
    {
        //Its User Check

        // con.Open();
        SqlDataAdapter dt = new SqlDataAdapter("select Twostep_status from [NewUser] where Twostep_status='"
+ label9.Text + "'", con);
        DataTable d1 = new DataTable();
        dt.Fill(d1);
        if(d1.Rows.Count == 1)
        {
            //two step is enabled
            securitykey sw = new securitykey();
            //passing value of username
            sw.passingvalue = guna2TextBox1.Text;
            //passing admin or user value
            if(guna2ToggleSwitch1.Checked)
            {
                //user
                sw.passingtype = label7.Text;

            }
            else
            {
                //admin
                sw.passingtype = label4.Text;
            }
            sw.Show();

            this.Hide();
        }
        else
        {
            //two step is disabled
            Master2 ss = new Master2();
            ss.passingvalue = guna2TextBox1.Text;//username
            if(guna2ToggleSwitch1.Checked)
            {
                ss.passingtype = label7.Text;//user

            }
            else

```

```

        {
            ss.passingtype = label4.Text;//admin
        }
        ss.Show();

        this.Hide();

    }
    // con.Close();

}

else
{
    //Its Admin Check
    // con.Open();
    SqlDataAdapter dt = new SqlDataAdapter("select Twostep_status from [SuperAdmin] where
Twostep_status=" + label9.Text + "", con);
    DataTable d1 = new DataTable();
    dt.Fill(d1);
    if (d1.Rows.Count == 1)
    {

        securitykey sw = new securitykey();
        sw.passingvalue = guna2TextBox1.Text;//paasing username
        if (guna2ToggleSwitch1.Checked)
        {
            sw.passingtype = label7.Text;//passing user

        }
        else
        {
            sw.passingtype = label4.Text;// passing admin
        }
        sw.Show();

        this.Hide();

    }
    else
    {
        Master2 ss = new Master2();
        ss.passingvalue = guna2TextBox1.Text;
        if (guna2ToggleSwitch1.Checked)
        {
            ss.passingtype = label7.Text;

        }
        else
    }
}

```

```

        {
            ss.passingtype = label4.Text;
        }
        ss.Show();

        this.Hide();

    }
    // con.Close();

}

//Two Step End

}

private void Login_Load(object sender, EventArgs e)
{
    loop();
}

private void guna2GradientButton1_Click(object sender, EventArgs e)
{
}

int r=0;
private void btnlogin_Click(object sender, EventArgs e)
{
    if(guna2ToggleSwitch1.Checked)
    {
        try
        {
            //user pass check
            con.Open();
            SqlDataAdapter dt = new SqlDataAdapter("select Username from [NewUser] where Username=" + guna2TextBox1.Text + "and Password=" + guna2TextBox2.Text + "", con);
            DataTable d1 = new DataTable();
            dt.Fill(d1);
            if(d1.Rows.Count == 1)
            {

                MessageBox.Show(this, "LOGIN SUCESSFULLY...:", "Sucess...", MessageBoxButtons.OK, MessageBoxIcon.Question);
                this.Player.Stop();

                twostep(); //calling twostep for checking two step verification
            }
        }
    }
}

```

```

        }
    else
    {

        MessageBox.Show(this, "Wrong Username And Password....:", "Error...", MessageBoxButtons.OK, MessageBoxIcon.Error);
        guna2TextBox1.Clear();
        guna2TextBox2.Clear();
        r = r + 1;
        if (r == 3)
        {
            Application.Exit();
        }
        else if (r == 2)
        {
            this.Player.SoundLocation = @"D:\DeadLock\DeadLock\Resources\Alarm.wav";
            this.Player.PlayLooping();
        }
        else
        {

        }

    }
    con.Close();
}
catch (Exception ew)
{
    var rr = 0;
    MessageBox.Show(this, "Wrong Username And Password...:", "Error...", MessageBoxButtons.OK, MessageBoxIcon.Error);
    guna2TextBox1.Clear();
    guna2TextBox2.Clear();
    rr = rr + 1;
    if (rr == 3)
    {
        Application.Exit();
    }
}
// it's admin login entry
else
{
    try
    {
        con.Open();
        SqlDataAdapter dt = new SqlDataAdapter("select Username from [SuperAdmin] where Username=" + guna2TextBox1.Text + "and Password=" + guna2TextBox2.Text + "", con);

```



```

}

private void label6_Click(object sender, EventArgs e)
{
    //displaying new user form
    User sw = new User();
    sw.Show();
    this.Hide();
}

private void label6_MouseHover(object sender, EventArgs e)
{
    label6.ForeColor = Color.Red;
}

private void label6_MouseLeave(object sender, EventArgs e)
{
    label6.ForeColor = Color.White;
}

private void Swtshow_CheckedChanged(object sender, EventArgs e)
{
}

private void Swtshow_CheckedChanged_1(object sender, EventArgs e)
{//password view
    if (Swtshow.Checked)
    {
        guna2TextBox2.UseSystemPasswordChar = false;
    }
    else
    {
        guna2TextBox2.UseSystemPasswordChar = true;
    }
}

private void label5_Click(object sender, EventArgs e)
{
    Forgetpass sw = new Forgetpass();
    sw.Show();
    this.Hide();
}

private void label5_MouseHover(object sender, EventArgs e)
{
}

```

```

        label5.ForeColor = Color.Red;
    }

    private void label5_MouseLeave(object sender, EventArgs e)
    {
        label5.ForeColor = Color.White;
    }
    public void loop()
    {
        Label1.Text = "";
        count_ = 1;
        str_ = "DEAD LOCKER";
        timer1.Enabled = true;
        //-----
        label2.Text = "";
        count1_ = 1;
        str1_ = "T H E";
        timer2.Enabled = true;
    }

    private void panel1_Paint(object sender, PaintEventArgs e)
    {

    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        if (Label1.Text.Length == str_.Length)
        {
            timer1.Enabled = false;
            loop();
            return;
        }
        Label1.Text = str_.Substring(0, count_);
        count_++;
    }

    private void timer2_Tick(object sender, EventArgs e)
    {
        if (label2.Text.Length == str1_.Length)
        {
            timer2.Enabled = false;
            return;
        }
        label2.Text = str1_.Substring(0, count1_);
        count1_++;
    }
}

```

## ❖ MASTER2.CS

```
using SpeechLib;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class Master2 : Form
    {
        SqlConnection con = new SqlConnection("Data Source=DESKTOP-
9QTIITB\\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();
        public Master2()
        {
            InitializeComponent();
            customizeDesign();
        }
        public string name,type;
        public string passingvalue
        {
            get { return name; }
            set { name = value; }
        }
        public string passingtype
        {
            get { return type; }
            set { type = value; }
        }
        private void customizeDesign()
        {
            panelSubMenu1.Visible = false;
            panelSubMenu2.Visible = false;
            panelSubMenu3.Visible = false;
```

```

        panelSubMenu4.Visible = false;

    }

    private void hideSubMenu()
    {
        if (panelSubMenu1.Visible == true)
            panelSubMenu1.Visible = false;
        if (panelSubMenu2.Visible == true)
            panelSubMenu2.Visible = false;
        if (panelSubMenu3.Visible == true)
            panelSubMenu3.Visible = false;
    }

    private void ShowSubMenu(Panel subMenu)
    {
        if (subMenu.Visible == false)
        {
            hideSubMenu();
            subMenu.Visible = true;
        }
        else
        {
            subMenu.Visible = false;
        }
    }

    private void Super_Button1_Click(object sender, EventArgs e)
    {
        ShowSubMenu(panelSubMenu1);
    }

    private void Lock_Folder_Click(object sender, EventArgs e)
    {
        openChildFormInPanel(new Lock_Folder(label1.Text,label4.Text));

        hideSubMenu();

    }

    private void Super_Button2_Click(object sender, EventArgs e)
    {
        ShowSubMenu(panelSubMenu2);
    }

    private void Lock_Drive_Click(object sender, EventArgs e)
    {
        openChildFormInPanel(new DriveLock());
        hideSubMenu();
    }

```

```

private Form activeForm = null;
private void openChildFormInPanel(Form childForm)
{
    if (activeForm != null)
        activeForm.Close();
    activeForm = childForm;
    childForm.TopLevel = false;
    childForm.FormBorderStyle = FormBorderStyle.None;
    childForm.Dock = DockStyle.Fill;
    panelChieldForm.Controls.Add(childForm);
    panelChieldForm.Tag = childForm;
    childForm.BringToFront();
    childForm.Show();
}

//Fetching Name
string nnmmee;
private void fetchname()
{
    if(label4.Text == "User")
    {
        cmd = new SqlCommand("select Name from NewUser where Username='"+ label1.Text + "'"
        , con);
        con.Open();
        SqlDataReader sdr = cmd.ExecuteReader();
        while (sdr.Read())
        {
            nnmmee = sdr["Name"].ToString();
        }
        con.Close();
    }
    else
    {
        cmd = new SqlCommand("select Name from SuperAdmin where Username='"+ label1.Text
        + "' ", con);
        con.Open();
        SqlDataReader sdr = cmd.ExecuteReader();
        while (sdr.Read())
        {
            nnmmee = sdr["Name"].ToString();
        }
        con.Close();
    }
}

// Checking Rights For USer
public void retrive()

```

```

{
    if(label4.Text == "User")
    {
        string drivelock,rights,folderlock,hidefolder,encrypt,twostep,changeppass;
        cmd = new SqlCommand("Select * from rightss ", con);
        con.Open();
        SqlDataReader sdr = cmd.ExecuteReader();
        while (sdr.Read())
        {
            drivelock = sdr["Drive_Lock"].ToString();
            rights = sdr["Rights"].ToString();
            folderlock = sdr["Lock_Folder"].ToString();
            hidefolder = sdr["Hide_Folder"].ToString();
            encrypt = sdr["Encrypt_Folder"].ToString();
            twostep = sdr["Two_step"].ToString();
            changeppass = sdr["Change_Password"].ToString();

            if(drivelock == "Deny")
                Super_Button2.Visible = false;
            if(rights == "Deny")
                btnRights.Visible = false;
            if(folderlock == "Deny")
                Lock_Folder.Visible = false;
            if(hidefolder == "Deny")
                btnHide.Visible = false;
            if(encrypt == "Deny")
                Super_Button4.Visible = false;
            if(twostep == "Deny")
                guna2GradientButton11.Visible = false;
            if(changeppass == "Deny")
                guna2GradientButton10.Visible = false;
        }
        con.Close();
    }
}

private void Master2_Load(object sender, EventArgs e)
{
    label4.Text = "" + type;
    label1.Text = "" + name;
    retrive();
}

```

```

    fetchname();

}

private void guna2ControlBox1_Click_1(object sender, EventArgs e)
{
}

private void guna2Button1_Click(object sender, EventArgs e)
{
    //Application.Exit();
    if (DialogResult.Yes == MessageBox.Show("Are You Sure Want To Exit..?", "Confirmation",
    MessageBoxButtons.YesNo, MessageBoxIcon.Warning))
    {
        Application.Exit();
    }
}

private void button1_Click(object sender, EventArgs e)
{
    // ll.Show();
}

private void btnHide_Click(object sender, EventArgs e)
{
    openChildFormInPanel(new HideFolder(label1.Text));
    hideSubMenu();
}

private void btntwostep_Click(object sender, EventArgs e)
{
}

private void Super_Button3_Click(object sender, EventArgs e)
{
}

private void btnRights_Click(object sender, EventArgs e)
{
}

private void guna2GradientButton11_Click(object sender, EventArgs e)

```

```

{
}

private void guna2GradientButton2_Click(object sender, EventArgs e)
{
}

private void Super_Button3_Click_1(object sender, EventArgs e)
{
    ShowSubMenu(panelSubMenu3);
}

private void btnEncryption_Click(object sender, EventArgs e)
{
}

private void Super_Button4_Click(object sender, EventArgs e)
{
    ShowSubMenu(panelSubMenu4);
}

private void btnenccc_Click(object sender, EventArgs e)
{
    openChildFormInPanel(new Encrypt2());
    hideSubMenu();
}

private void pictureBox1_Click(object sender, EventArgs e)
{
}

private void timer1_Tick(object sender, EventArgs e)
{
    SpVoice voice = new SpVoice();
    voice.Speak("Welcome To The World Of Dead Lock. Master "+ nmmee,
    SpeechVoiceSpeakFlags.SVSFlagsAsync);
    voice.WaitUntilDone(30000);
    timer1.Stop();
}

private void panelLogo_Paint(object sender, PaintEventArgs e)
{
}

```

```
}

private void guna2GradientButton10_Click(object sender, EventArgs e)
{
    openChildFormInPanel(new Change_Password(label1.Text, label4.Text));
    hideSubMenu();
}

private void guna2GradientButton11_Click_1(object sender, EventArgs e)
{
    openChildFormInPanel(new TwoStep(label1.Text, label4.Text));
    hideSubMenu();
}

private void btnRights_Click_1(object sender, EventArgs e)
{
    openChildFormInPanel(new Rights());
    hideSubMenu();
}

private void guna2Button2_Click(object sender, EventArgs e)
{
    if(this.WindowState == FormWindowState.Maximized)
    {
        this.WindowState = FormWindowState.Minimized;
    }
}

private void panelChieldForm_Paint(object sender, PaintEventArgs e)
{

}
}
```

## ❖ ADMIN.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net;
using System.Net.Mail;
using System.Net.NetworkInformation;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Data.SqlClient;
using System.Configuration;

namespace DeadLock
{
    public partial class Admin : Form
    {
        SqlConnection con = new SqlConnection("Data Source=DESKTOP-
9QTIITB\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();

        public Admin()
        {
            InitializeComponent();
        }

        private void txtpass_Leave(object sender, EventArgs e)
        {
            groupBox1.Visible = false;
        }

        private void txtpass_TextChanged(object sender, EventArgs e)
        {
            groupBox1.Visible = true;
            //Validating groupbox condition.............
            if (!(txtpass.TextLength >= 8))
            {
                // MessageBox.Show("Password Contain at least 8 Character ");
                label4.ForeColor = Color.Red;
                btnCreateacc.Enabled = false;
            }
        }
    }
}
```

```

        }
    else
    {
        label4.ForeColor = Color.SpringGreen;

    }
//-----
if (!txtpass.Text.Any(char.IsUpper))//checking one upper char
{
    label2.ForeColor = Color.Red;
    btnCreateacc.Enabled = false;
}
else
{
    label2.ForeColor = Color.SpringGreen;

}
//-----
if (!txtpass.Text.Any(char.IsLower))//checking one lower char
{
    label3.ForeColor = Color.Red;
    btnCreateacc.Enabled = false;
}
else
{
    label3.ForeColor = Color.SpringGreen;

}
//-----
if (!txtpass.Text.Any(char.IsNumber))//checking number
{
    label5.ForeColor = Color.Red;
    btnCreateacc.Enabled = false;
}
else
{
    label5.ForeColor = Color.SpringGreen;

}
//-----
if (txtpass.Text.Contains(" "))//no blank spaces in pass
{
    label8.ForeColor = Color.Red;
    btnCreateacc.Enabled = false;
}
else
{
    label8.ForeColor = Color.SpringGreen;
}

```

```

        }

//-----
String test_string = txtpass.Text;
if (Regex.IsMatch(test_string, "^[a-zA-Z0-9\\x20]+$"))//using regular expression for checking
symbol in pass
{
    label7.ForeColor = Color.Red;
    btnCreateacc.Enabled = false;
}
else
{
    label7.ForeColor = Color.SpringGreen;
}

//Checking Password and Retype Password
passcheck();
}

private void passcheck()
{
    if (txtpass.Text == TextBox5.Text)
    {
        btnCreateacc.Enabled = true;

    }
    else
    {
        btnCreateacc.Enabled = false;
    }
}

private void label7_Click(object sender, EventArgs e)
{

}

private void txtpass_Click(object sender, EventArgs e)
{

}

private void txtpass_MouseClick(object sender, MouseEventArgs e)
{
    groupBox1.Visible = true;
}

//Foram Load
private void Admin_Load(object sender, EventArgs e)
{
}

```

```

    TextBox1.Enabled = true;
    TextBox2.Enabled = true;
    txtotp.Enabled = true;
    TextBox3.Enabled = false;
    TextBox5.Enabled = false;
    txtpass.Enabled = false;

    if(TextBox1.Text == "" && TextBox2.Text == "" && TextBox3.Text == "" && txtpass.Text
    == "" && TextBox5.Text == ""&&txtotp.Text=="")
    {
        btnCreateacc.Enabled = false;
    }
    else
    {
        btnCreateacc.Enabled = true;
    }
    PictureBox3.Visible = false;
    pictureBox1.Visible = false;
    //Checking Internet Connection...
    if(NetworkInterface.GetIsNetworkAvailable())
    {
    }
    else
    {
        MessageBox.Show("Please Connect Internet, To Proceed... :( ", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        this.Close();
    }
    //-----
}

```

```

private void TextBox5_TextChanged(object sender, EventArgs e)
{
    //Checking Password and Retype Password
    passcheck();
}
private void email()
{
    //Generate Random OTP
    string otpcode;
    Random rand = new Random();

```

```

        otpcode = (rand.Next(9999999)).ToString();
        label9.Text = otpcode;

        //Sending Mail
        try
        {
            TextBox1.Enabled = false;
            TextBox2.ReadOnly = true;

            TextBox3.Enabled = false;
            TextBox5.Enabled = false;
            txtpass.Enabled = false;

            MailMessage mail = new MailMessage();
            mail.From = new MailAddress("Gmail-Id");
            mail.To.Add(TextBox2.Text);
            mail.IsBodyHtml = true;
            mail.Subject = "OTP Verification...";
            mail.Body = "Your OTP Code is :- " + otpcode;
            SmtpClient smtp = new SmtpClient("smtp.gmail.com");
            smtp.Port = 587;
            smtp.Credentials = new NetworkCredential("Gmail-Id", "Password");
            smtp.EnableSsl = true;
            smtp.Send(mail);
            MessageBox.Show("Email Sent Sucessfully | Check Email... ", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        catch(Exception e)
        {
            MessageBox.Show(e.Message, "", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void TextBox2_Leave(object sender, EventArgs e)
    {

    }

    private void txtotp_TextChanged(object sender, EventArgs e)
    {
        pictureBox1.Visible = false;
    }

    private void PictureBox3_Click(object sender, EventArgs e)
    {
        //Validating OTP

        if(label9.Text == txtotp.Text)
        {

```

```

        MessageBox.Show("OTP Verified Sucessfully ", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        TextBox1.Enabled = true;
        TextBox2.ReadOnly = true;
        txtotp.ReadOnly = true;
        TextBox3.Enabled = true;
        TextBox5.Enabled = true;
        txtpass.Enabled = true;
        PictureBox3.Visible = false;

    }
    else
    {
        MessageBox.Show("Wrong Verification Code :( ", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        txtotp.Clear();
    }
}

private void pictureBox1_Click(object sender, EventArgs e)
{
    //Checking Domain And Sending Mail

    if (TextBox2.Text.Contains("@gmail.com"))
    {
        MessageBox.Show("Please Enter OTP To Continue ", "", MessageBoxButtons.OK,
MessageBoxIcon.Question);
        email();
        PictureBox3.Visible = true;
        MessageBox.Show("Please Check Your Gmail ", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);

    }
    else
    {
        MessageBox.Show("Please Enter Only Gmail Id ", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        TextBox2.PlaceholderText = "E.g :- abcpqr@gmail.com";
    }
}

private void TextBox2_TextChanged(object sender, EventArgs e)
{
    pictureBox1.Visible = true;
}

private void picshowpass_Click(object sender, EventArgs e)
{

```

```

//displaying or hiding password
if (txtpass.UseSystemPasswordChar == false && TextBox5.UseSystemPasswordChar == false)
{
    txtpass.UseSystemPasswordChar = true;
    TextBox5.UseSystemPasswordChar = true;
}
else
{
    txtpass.UseSystemPasswordChar = false;
    TextBox5.UseSystemPasswordChar = false;
}

}

private void btnCreateacc_Click(object sender, EventArgs e)
{
    if (TextBox1.Text != "" && TextBox2.Text != "" && TextBox3.Text != "" && txtpass.Text != ""
        && TextBox5.Text != "" && txtotp.Text != "")
    {
        cmd = new SqlCommand("insert into
SuperAdmin(Name,Email,Username,Password,Twostep_status,Two_Step,Form_Status)
values(@Name,@Email,@Username,@Password,@Twostep_status,@Two_Step,@Form_Status)",

con);
        con.Open();
        cmd.Parameters.AddWithValue("@Name", TextBox1.Text);
        cmd.Parameters.AddWithValue("@Email", TextBox2.Text);
        cmd.Parameters.AddWithValue("@Username", TextBox3.Text);
        cmd.Parameters.AddWithValue("@Password", txtpass.Text);
        cmd.Parameters.AddWithValue("@Twostep_status", "0");
        cmd.Parameters.AddWithValue("@Two_Step", "0");
        cmd.Parameters.AddWithValue("@Form_Status", "1");
        cmd.ExecuteNonQuery();
        MessageBox.Show("Super User Added Sucessfully ", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        con.Close();
        Login ss = new Login();
        ss.Show();

        Form1 sq = new Form1();
        sq.Close();
        this.Hide();
    }
    else
    {
        MessageBox.Show("Please Fill All Fields ", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}

```

## ❖ **LOCK\_FOLDER.CS**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Security.AccessControl;
using System.Security.Principal;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class Lock_Folder : Form
    {
        SqlConnection abc = new SqlConnection("Data Source=DESKTOP-
9QTIITB\\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();

        int id = 0;

        public Lock_Folder(string getname, string gettype)
        {
            InitializeComponent();
            label1.Text = getname; //Getting value from master form
            label7.Text = gettype;
            arr = new string[6];
            status = "";
            arr[0] = ".{2559a1f2-21d7-11d4-bdaf-00c04f60b9f0}";
        }

        public static string fc;
        public string status;
        public static string sp;
        string[] arr;
        // private string _pathkey;
        //public string pathkey
        //{
        //    get { return _pathkey; }
        //    set { _pathkey = value; }
        //}
    }
}
```

```

private void btnBrowse_Click(object sender, EventArgs e)
{
    status = arr[0];
    if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)
    {
        textBox1.Text = folderBrowserDialog1.SelectedPath;
        sp = folderBrowserDialog1.SelectedPath;
        DirectoryInfo d = new DirectoryInfo(folderBrowserDialog1.SelectedPath);
        string selectedpath = d.Parent.FullName + d.Name;//taking complete parent path
        label4.Text = selectedpath;
        BtnLock.Enabled = true;
    }
}

private void BtnLock_Click(object sender, EventArgs e)
{
    //calling random pass
    string password = GetRandomPassword(8);
    label2.Text = password + label1.Text;

    BtnLock.Enabled = false;
    if (textBox1.Text.LastIndexOf(".".) == -1)
    {

        DirectoryInfo d = new DirectoryInfo(sp);
        string selectedpath = d.Parent.FullName + d.Name;
        if (!d.Root.Equals(d.Parent.FullName))
            d.MoveTo(d.Parent.FullName + "\\\" + d.Name + status);

        else d.MoveTo(d.Parent.FullName + d.Name + status);

        sp = sp + ".{2559a1f2-21d7-11d4-bdaf-00c04f60b9f0}";
        FileSecurity fileSecurity = File.GetAccessControl(sp);//getting list of access control
        fileSecurity.SetOwner(WindowsIdentity.GetCurrent().User);//getting current user and set
        current user as owner
        fileSecurity.AddAccessRule(new FileSystemAccessRule(Environment.UserName,
        FileSystemRights.FullControl, AccessControlType.Deny));//removing all access rule of directory
        File.SetAccessControl(sp, fileSecurity);

        MessageBox.Show("The Selected Folder Has Been Locked Successful...", "",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

//-----
cmd = new SqlCommand("insert into
lock(path,username,rndpass)values(@path,@username,@rndpass)", abc);
abc.Open();

```

```

        cmd.Parameters.AddWithValue("@path", textBox1.Text);
        cmd.Parameters.AddWithValue("@username", label1.Text);
        cmd.Parameters.AddWithValue("@rndpass", label2.Text);
        cmd.ExecuteNonQuery();
        // MessageBox.Show(this, "ADDED SUCESSFULLY.", "Sucess...", 
MessageBoxButtons.OK, MessageBoxIcon.Question);
        display();
        abc.Close();
        textBox1.Text = "";

    }
    else
    {
        MessageBox.Show("Folder Is Already Locked", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}

private void display()
{
    SqlDataAdapter adapt = new SqlDataAdapter("select * from lock where username like '" +
label1.Text + "%'", abc);
    DataTable dt = new DataTable();
    adapt.Fill(dt);
    dataGridView1.DataSource = dt;
}

private string getstatus(string stat)
{
    for (int i = 0; i < 6; i++)
        stat = stat.Substring(stat.LastIndexOf("."));
    return stat;
}

/// Generate random password....
public static string GetRandomPassword(int length)
{
    const string chars =
"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!@#$%^*()_+=-";
    StringBuilder sb = new StringBuilder();
    Random rnd = new Random();

    for (int i = 0; i < length; i++)
    {
        int index = rnd.Next(chars.Length);
    }
}

```

```

        sb.Append(chars[index]);
    }

    return sb.ToString();
}

private void fetched()
{
    SqlCommand cmd = new SqlCommand("SELECT rndpass FROM lock WHERE rndpass='" +
label3.Text + "'", abc);
    abc.Open();
    SqlDataReader re = cmd.ExecuteReader();

    if (re.Read())
    {
        label6.Text = re["rndpass"].ToString();
    }
    abc.Close();
    // dataGridView1.DataSource = dt;
}

private void guna2TextBox2_TextChanged(object sender, EventArgs e)
{
    abc.Open();
    SqlDataAdapter adapt = new SqlDataAdapter("select * from lock where path like%" +
guna2TextBox2.Text + "%'", abc);
    DataTable dt = new DataTable();
    adapt.Fill(dt);
    dataGridView1.DataSource = dt;
    abc.Close();
}

private void Lock_Folder_Load(object sender, EventArgs e)
{
    // TODO: This line of code loads data into the 'deadLockDataSet1._lock' table. You can move,
    or remove it, as needed.
    this.lockTableAdapter1.Fill(this.deadLockDataSet1._lock);
    // TODO: This line of code loads data into the 'deadLockDataSet1._lock' table. You can move,
    or remove it, as needed.
    this.lockTableAdapter1.Fill(this.deadLockDataSet1._lock);
    //label77.Text = "" + getname;
    display();
    BtnLock.Enabled = false;

}

private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
}

```

```

id = Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());
//textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString() + ".{2559a1f2-
21d7-11d4-bdaf-00c04f60b9f0}";
textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();

label3.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();

if(e.ColumnIndex == 4)
{
    label5.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString() + ".{2559a1f2-
21d7-11d4-bdaf-00c04f60b9f0}";

    label3.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
    // textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
    fetched();
    status = arr[0];
    textBox1.Text = folderBrowserDialog1.SelectedPath;
    if(folderBrowserDialog1.ShowDialog() == DialogResult.OK)
    {
        textBox1.Text = folderBrowserDialog1.SelectedPath;
        sp = folderBrowserDialog1.SelectedPath;
        DirectoryInfo d = new DirectoryInfo(folderBrowserDialog1.SelectedPath);
        string selectedpath = d.Parent.FullName + d.Name;
        label4.Text = selectedpath;

    }
}

//-----
if(label5.Text == textBox1.Text && label3.Text == label6.Text)
{
    // MessageBox.Show("Matcheddddd");
    if(sp.LastIndexOf(".".) == -1)
    {
        MessageBox.Show("Folder Is Already Unlocked", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    else
    {
        fc = null;
    }
}

```

```

DirectoryInfo d = new DirectoryInfo(sp);

string selectedpath = d.Parent.FullName + d.Name;
FileSecurity fileSecurity = File.GetAccessControl(sp);
// fileSecurity.SetOwner(WindowsIdentity.GetCurrent().User);
fileSecurity.RemoveAccessRule(new FileSystemAccessRule(Environment.UserName,
FileSystemRights.FullControl, AccessControlType.Deny));
File.SetAccessControl(sp, fileSecurity);

MessageBox.Show("The Selected Folder Has Been Unlocked
Successful","", MessageBoxButtons.OK, MessageBoxIcon.Information);
status = getstatus(status);
d.MoveTo(sp.Substring(0, sp.LastIndexOf(".")));
string nsp = sp.Replace(".{2559a1f2-21d7-11d4-bdaf-00c04f60b9f0}", "");
sp = nsp;
textBox1.Text = nsp;

//-----



if (id != 0)
{
    cmd = new SqlCommand("delete lock where ID=@id", abc);
    abc.Open();
    cmd.Parameters.AddWithValue("@id", id);
    cmd.ExecuteNonQuery();
    display();
    abc.Close();
    textBox1.Clear();
}
else
{
    MessageBox.Show("Please Select Record to Delete", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    textBox1.Clear();
}

}

else
{
    MessageBox.Show("Please Select Proper Folder Path....", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    textBox1.Clear();
}

```

```
        }

    }

private void panel2_Paint(object sender, PaintEventArgs e)
{
}

private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    }

}
```

## ❖ HIDE FOLDER.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Diagnostics;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class HideFolder : Form
    {
        SqlConnection abc = new SqlConnection("Data Source=DESKTOP-9QTIITB\\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();
        public HideFolder(string getname)
        {
            InitializeComponent();
        }
    }
}
```

```

        label1.Text = getname;
    }

    private void panel2_Paint(object sender, PaintEventArgs e)
    {

    }

    private void btnBrowse_Click(object sender, EventArgs e)
    {
        if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)
        {
            // Select the folder to lock
            textBox1.Text = folderBrowserDialog1.SelectedPath;
            textBox1.Text = $"\"{textBox1.Text}\\"";
            BtnLock.Enabled = true;
        }
    }

    private void display()
    {

        SqlDataAdapter adapt = new SqlDataAdapter("select * from hiden where username like '" + label1.Text + "%'", abc);
        DataTable dt = new DataTable();
        adapt.Fill(dt);
        dataGridView1.DataSource = dt;
    }

    private void BtnLock_Click(object sender, EventArgs e)
    {
        if (textBox1.Text != "")
        {

            BtnLock.Enabled = false;
            ProcessStartInfo ps = new ProcessStartInfo();
            ps.WindowStyle = ProcessWindowStyle.Hidden;
            ps.FileName = "cmd.exe";
            ps.Arguments = @"/c attrib +s +h /s /d " + textBox1.Text;
            ps.Verb = "runas";
            Process.Start(ps);
            MessageBox.Show("Hiden Sucessfully", "Sucess...", MessageBoxButtons.OK,
MessageBoxIcon.Question);
            //-----
            cmd = new SqlCommand("insert into hiden(path,username)values(@path,@username)", abc);
            abc.Open();
        }
    }
}

```

```

cmd.Parameters.AddWithValue("@path", textBox1.Text);
cmd.Parameters.AddWithValue("@username", label1.Text);
cmd.ExecuteNonQuery();
// MessageBox.Show(this, "ADDED SUCESSFULLY.", "Sucess...", 
MessageBoxButtons.OK, MessageBoxIcon.Question);
display();
abc.Close();
textBox1.Text = "";
}
else
{
    MessageBox.Show("Please Select Folder For Hide.....", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
}
int id = 0;
private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    id = Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());
    textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

    if (e.ColumnIndex == 2)
    {

        if (textBox1.Text != "")
        {
            ProcessStartInfo ps = new ProcessStartInfo();
            ps.WindowStyle = ProcessWindowStyle.Hidden;
            ps.FileName = "cmd.exe";
            ps.Arguments = @"/c attrib -s -h /s /d " + textBox1.Text.ToString();
            ps.Verb = "runas";
            Process.Start(ps);
            MessageBox.Show("Folder Un-Hide Successfully", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
        //-----
        if (id != 0)
        {
            cmd = new SqlCommand("delete hiden where ID=@id", abc);
            abc.Open();
            cmd.Parameters.AddWithValue("@id", id);
            cmd.ExecuteNonQuery();

            // MessageBox.Show("Record Deleted Successfully!");
            display();
            textBox1.Clear();
            abc.Close();
        }
        else
        {
    }
}

```

```
        MessageBox.Show("Please Select Record to Delete", "", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}
else
{
    MessageBox.Show("Please Select Folder For Un-Hide....", "", MessageBoxButtons.OK,
MessageBoxIcon.Error);
}

}

private void HideFolder_Load(object sender, EventArgs e)
{
    // TODO: This line of code loads data into the 'deadLockDataSet1.hiden' table. You can move,
or remove it, as needed.
    this.hidenTableAdapter1.Fill(this.deadLockDataSet1.hiden);

    BtnLock.Enabled = false;
    display();

}

private void guna2TextBox2_TextChanged(object sender, EventArgs e)
{
    abc.Open();
    SqlDataAdapter adapt = new SqlDataAdapter("select * from hiden where path like '%" +
guna2TextBox2.Text + "%'", abc);
    DataTable dt = new DataTable();
    adapt.Fill(dt);
    dataGridView1.DataSource = dt;
    abc.Close();
}
```

## ❖ DRIVE\_LOCK.CS

```
using Microsoft.Win32;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Security.Principal;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class DriveLock : Form
    {
        private RegistryKey regkey;
        private string Regpath = @"Software\Microsoft\Windows\CurrentVersion\Policies\Explorer";
        private string OsDrive;
        private Int32 DriveDword, Loc;
        public DriveLock()
        {
            InitializeComponent();
        }

        private void panel2_Paint(object sender, PaintEventArgs e)
        {

        }

        private void BtnLock_Click(object sender, EventArgs e)
        {
            OsDrive = Path.GetPathRoot(Environment.SystemDirectory); //getting operating system path
            if (comboBox1.Text.Length > 0)
            {
                if (!(OsDrive == comboBox1.Text))
                {
                    //registry path
                    RegistryKey key =
Registry.CurrentUser.OpenSubKey(@"Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
"); //current user
```

```

        int ss = int.Parse(key.GetValue("NoViewOnDrive").ToString());//getting value from
registry
        DriveDword = Driv(comboBox1.Text) + ss;//adding value from combobox
        regkey = Registry.CurrentUser.OpenSubKey(Regpath, true);

        regkey.SetValue("NoViewOnDrive", DriveDword, RegistryValueKind.DWord);//setting
value
        MessageBox.Show("Drive Lock Successfully", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        foreach (var myp in Process.GetProcessesByName("explorer"))//killing all processes
myp.Kill();
        Process process = Process.Start("explorer.exe");//launching new process of explorer

    }
    else
        MessageBox.Show("You Cannot Lock This Drive Because It Contains Your Operating
System", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
        MessageBox.Show("Please Select A Drive To Lock", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
}

private void DriveLock_Load(object sender, EventArgs e)
{
    // Creating Registry File With 0 Values
    regkey = Registry.CurrentUser.OpenSubKey(Regpath, true);
    regkey.SetValue("NoViewOnDrive", 0, RegistryValueKind.DWord);//creating file in registry
currentuser
    regkey.SetValue("NoDrives", 0, RegistryValueKind.DWord);//creating file in registry
currentuser

    // Checking Admin And Putting Values On ComboBox
    if(this.IsAdministrator())
    {
        foreach (var drive in Environment.GetLogicalDrives())
        {
            DriveInfo InfoDrive = new DriveInfo(drive);//checking all available drives in system
            if(InfoDrive.DriveType == DriveType.Removable | InfoDrive.DriveType ==
DriveType.Fixed)
                comboBox1.Items.Add(drive);
        }
        comboBox1.SelectedIndex = 0;
        comboBox1.Text = comboBox1.SelectedItem.ToString();
    }
    else
    {
}

```

```

        MessageBox.Show("Administrators Access Rights Are Required.", "",  

MessageBoxButtons.OK, MessageBoxIcon.Error);  

        System.Environment.Exit(0);  

    }  

}  
  

private void comboBox1_KeyPress(object sender, KeyPressEventArgs e)  

{  

    e.Handled = true;  

}  
  

private void btnunlock_Click(object sender, EventArgs e)  

{  

    OsDrive = Path.GetPathRoot(Environment.SystemDirectory);  

    if (comboBox1.Text.Length > 0)  

    {  

        if (!(OsDrive == comboBox1.Text))  

        {  

            //getting registry value  

            RegistryKey key =  

Registry.CurrentUser.OpenSubKey(@"Software\Microsoft\Windows\CurrentVersion\Policies\Explorer  

");  

            if (key != null)  

            {  

                int ss = int.Parse(key.GetValue("NoViewOnDrive").ToString());//if file present getting  

value of it  

                DriveDword = ss - Driv(comboBox1.Text);//subtract from combobox value  

                regkey = Registry.CurrentUser.OpenSubKey(Regpath, true);  

                regkey.DeleteValue("NoViewOnDrive");  

                regkey.SetValue("NoViewOnDrive", DriveDword, RegistryValueKind.DWord);  

                MessageBox.Show("Drive UnLocked Successfully", "", MessageBoxButtons.OK,  

MessageBoxIcon.Information);  

                foreach (var myp in Process.GetProcessesByName("explorer"))//killing all process  

                    myp.Kill();  

                Process process = Process.Start("explorer.exe");//launch file exlorer  

            }  

            else  

            {  

                MessageBox.Show("No Drive Locked", "", MessageBoxButtons.OK,  

MessageBoxIcon.Error);  

            }  

        }  

        else  

        {  

            MessageBox.Show("You cannot Hide this drive because it contains your operating  

system", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  

        }  

    else

```

```

        MessageBox.Show("Please select a drive to Hide", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }

public bool IsAdministrator()//method for checking administrator
{
    bool isAdmin;

    using (WindowsIdentity identity = WindowsIdentity.GetCurrent())
    {
        WindowsPrincipal principal = new WindowsPrincipal(identity);
        isAdmin = principal.IsInRole(WindowsBuiltInRole.Administrator);
    }

    return isAdmin;
}

private void guna2Button1_Click(object sender, EventArgs e)
{

}

private void btnHide_Click(object sender, EventArgs e)
{
    OsDrive = Path.GetPathRoot(Environment.SystemDirectory);
    if (comboBox1.Text.Length > 0)
    {
        if (!(OsDrive == comboBox1.Text))
        {
            //getting registry value
            RegistryKey key =
Registry.CurrentUser.OpenSubKey(@"Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
");

            int ss = int.Parse(key.GetValue("NoDrives").ToString());
            DriveDword = Driv(comboBox1.Text) + ss;
            regkey = Registry.CurrentUser.OpenSubKey(Regpath, true);

            regkey.SetValue("NoDrives", DriveDword, RegistryValueKind.DWord);
            MessageBox.Show("Drive Hide successfully", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
        foreach (var myp in Process.GetProcessesByName("explorer"))
            myp.Kill();
        Process process = Process.Start("explorer.exe");
    }
    else
        MessageBox.Show("You cannot Hide This Drive Because It Contains Operating
System...", "Error...", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

```

```

        }
    else
        MessageBox.Show("Please Select A Drive To Hide", "", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
    }

private void btnunhide_Click(object sender, EventArgs e)
{
    OsDrive = Path.GetPathRoot(Environment.SystemDirectory);
    if (comboBox1.Text.Length > 0)
    {
        if (!OsDrive == comboBox1.Text)
        {
            //getting registry value
            RegistryKey key =
Registry.CurrentUser.OpenSubKey(@"Software\Microsoft\Windows\CurrentVersion\Policies\Explorer");
            if (key != null)
            {
                int ss = int.Parse(key.GetValue("NoDrives").ToString());
                DriveDword = ss - Driv(comboBox1.Text);
                regkey = Registry.CurrentUser.OpenSubKey(Regpath, true);
                regkey.DeleteValue("NoDrives");

                regkey.SetValue("NoDrives", DriveDword, RegistryValueKind.DWord);
                MessageBox.Show("Drive Un-Hide Successfully", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                foreach (var myp in Process.GetProcessesByName("explorer"))
                    myp.Kill();
                Process process = Process.Start("explorer.exe");
            }
            else
            {
                MessageBox.Show("No Drive Hidden", "Error..", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        else
            MessageBox.Show("You Cannot Hide This Drive Because It Contains Your Operating
System", "", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
        MessageBox.Show("Please select a drive to Hide", "", MessageBoxButtons.OK,
MessageBoxIcon.Error);
}

private void groupBox1_Enter(object sender, EventArgs e)
{
}

```

```

    }

public Int32 Driv(string Drive)
{
    //it's default values of local drive
    if(Drive == @"A:\")
        Loc = 0;
    else if((Drive == @"B:\"))
        Loc = 2;
    else if((Drive == @"C:\"))
        Loc = 4;
    else if((Drive == @"D:\"))
        Loc = 8;
    else if((Drive == @"E:\"))
        Loc = 16;
    else if((Drive == @"F:\"))
        Loc = 32;
    else if((Drive == @"G:\"))
        Loc = 64;
    else if((Drive == @"H:\"))
        Loc = 128;
    else if((Drive == @"I:\"))
        Loc = 256;
    else if((Drive == @"J:\"))
        Loc = 512;
    else if((Drive == @"K:\"))
        Loc = 1024;
    else if((Drive == @"L:\"))
        Loc = 2048;
    else if((Drive == @"M:\"))
        Loc = 4096;
    else if((Drive == @"N:\"))
        Loc = 8192;
    else if((Drive == @"O:\"))
        Loc = 16384;
    else if((Drive == @"P:\"))
        Loc = 16384 * 2;
    else if((Drive == @"Q:\"))
        Loc = 65536;
    else if((Drive == @"R:\"))
        Loc = 131072;
    else if((Drive == @"S:\"))
        Loc = 262144;
    else if((Drive == @"T:\"))
        Loc = 262144 * 2;
    else if((Drive == @"U:\"))
        Loc = 1048576;
    else if((Drive == @"V:\"))

```

```
    Loc = 1048576 * 2;
else if((Drive == @"W:\"))
    Loc = 1048576 * 4;
else if((Drive == @"X:\"))
    Loc = 1048576 * 8;
else if((Drive == @"Y:\"))
    Loc = 1048576 * 16;
else if((Drive == @"Z:\"))
    Loc = 1048576 * 32;
return Loc;
}

}
```

❖ **ENCRYPT.CS**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.IO.Compression;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class Encrypt2 : Form
    {
        public Encrypt2()
        {
            InitializeComponent();
        }

        private void Encrypt2_Load(object sender, EventArgs e)
        {
            btnEncrypt.Visible = false;
            btnDecrypt.Visible = false;
            progressBar1.Visible = false;
            btnbrowse_opt.Enabled = false;
        }

        private void btnBrowse_inp_Click(object sender, EventArgs e)
        {
            if (!swtcheck.Checked)
                {//folder browse dialog for inpute box
                    if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)
                    {
                        // Select the folder to lock
                        textBox1.Text = folderBrowserDialog1.SelectedPath;
                        btnbrowse_opt.Enabled = true;
                        progressBar1.Value = 0;
                    }
                }
            else
            {
        }
```

```

//selected decrypt the open file dialog
 OpenFileDialog openFileDialog1 = new OpenFileDialog
{
    InitialDirectory = @"E:\",
    Title = "Browse Text Files",

    CheckFileExists = true,
    CheckPathExists = true,

    DefaultExt = "txt",
    Filter = "txt files (*.txt)|*.txt | All files (*.*)|*.*",
    FilterIndex = 2,
    RestoreDirectory = true,

    ReadOnlyChecked = true,
    ShowReadOnly = true
};

if(openFileDialog1.ShowDialog() == DialogResult.OK)
{
    textBox1.Text = openFileDialog1.FileName;
    btnbrowse_opt.Enabled = true;
    btnDecrypt.Visible = true;

}
}

private void btnbrowse_opt_Click(object sender, EventArgs e)
{
    if(!swtcheck.Checked)
    {
        if(folderBrowserDialog1.ShowDialog() == DialogResult.OK)
        {
            // display dialogbox for output
            textBox2.Text = folderBrowserDialog1.SelectedPath;
            if(File.Exists(textBox2.Text + "\\Encrypt.zip"))
            {
                File.Delete(textBox2.Text + "\\Encrypt.zip");
            }
            btnEncrypt.Visible = true;
        }
    }
}

```

```

    else
    {
        if(folderBrowserDialog1.ShowDialog() == DialogResult.OK)
        {

            textBox2.Text = folderBrowserDialog1.SelectedPath;
            if(Directory.Exists(textBox2.Text + "\\Decrypted"))
            {
                MessageBox.Show("Please Change Decrypted Directory Name | Otherwise Change The
Output Location...");
                textBox2.Clear();
            }
        }
    }

private void btnEncrypt_Click(object sender, EventArgs e)
{
    if(txtpass.Text != "" && textBox1.Text!="" && textBox2.Text!="")
    {
        string input = textBox1.Text;

        string op = textBox2.Text + "\\Encrypt.zip";

        string password = txtpass.Text;

        ZipFile.CreateFromDirectory(input, op);//creating zip file
        progressBar1.Style = ProgressBarStyle.Blocks;
        progressBar1.Value = 40;
        FileEncrypt(op, password);//calling encrypt method

        progressBar1.Value = 50;
        File.Delete(op);

        progressBar1.Value = 100;
        MessageBox.Show("Encrypted | Sucessfully....:)", "", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    else
    {
        progressBar1.Visible = false;
        MessageBox.Show("Please Enter Secret Key....", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

```

```

        }

    }

private void btnDecrypt_Click(object sender, EventArgs e)
{
    if (txtpass.Text != "" && textBox1.Text != "" && textBox2.Text != "")
    {
        string input = textBox1.Text;

        string output = textBox2.Text + "\\Encrypt.zip";

        string password = txtpass.Text;

        string ex = textBox2.Text + "\\Decrypted";
        progressBar1.Style = ProgressBarStyle.Blocks;
        progressBar1.Value = 30;

        FileDecrypt(input, output, password); //calling decypt method

        progressBar1.Value = 80;
        ZipFile.ExtractToDirectory(output, ex); //extracting zip file

        string[] directoryFiles = Directory.GetFiles(ex, "*.aes", SearchOption.AllDirectories);
        // DirectoryInfo di = new DirectoryInfo("YourPath");
        foreach (string directoryFile in directoryFiles)
        {
            File.Delete(directoryFile);
        }
        File.Delete(output);
        progressBar1.Value = 100;
        MessageBox.Show("Decrypted | Sucessfully...:", "", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    else
    {
        progressBar1.Visible = false;
        MessageBox.Show("Please Enter Secret Key....", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

public static byte[] GenerateRandomSalt()
{
    byte[] data = new byte[32];

    using (RNGCryptoServiceProvider rng = new RNGCryptoServiceProvider()) //generating high
quality random number
    {

```

```

        for (int i = 0; i < 10; i++)
    {
        // Fille the buffer with the generated data
        rng.GetBytes(data);
    }
}

return data;
}

private void FileEncrypt(string inputFile, string password)
{
    //generate random salt
    byte[] salt = GenerateRandomSalt();

    //create output file name
    FileStream fsCrypt = new FileStream(inputFile + ".aes", FileMode.Create);
    //convert password string to byte array
    byte[] passwordBytes = System.Text.Encoding.UTF8.GetBytes(password);
    RijndaelManaged AES = new RijndaelManaged(); //define aes
    AES.KeySize = 256;
    AES.BlockSize = 128;
    AES.Padding = PaddingMode.PKCS7;
    var key = new Rfc2898DeriveBytes(passwordBytes, salt, 50000); //generating key encryption
decryption
    AES.Key = key.GetBytes(AES.KeySize / 8);
    AES.IV = key.GetBytes(AES.BlockSize / 8);
    AES.Mode = CipherMode.CFB;
    fsCrypt.Write(salt, 0, salt.Length);

    CryptoStream cs = new CryptoStream(fsCrypt, AES.CreateEncryptor(),
CryptoStreamMode.Write);

    FileStream fsIn = new FileStream(inputFile, FileMode.Open);

    //create a buffer (1mb) so only this amount will allocate in the memory and not the whole file
    byte[] buffer = new byte[1048576];
    int read;

    try
    {
        while ((read = fsIn.Read(buffer, 0, buffer.Length)) > 0)
        {
            Application.DoEvents(); // -> for responsive GUI, using Task will be better
            cs.Write(buffer, 0, read);
        }

        // Close up
        fsIn.Close();
    }
}

```

```

        }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message);
    }
    finally
    {
        cs.Close();
        fsCrypt.Close();
    }
    // File.Delete = textBox1.Text;
    // MessageBox.Show("Done");
}

private void FileDecrypt(string inputFile, string outputFile, string password)
{
    byte[] passwordBytes = System.Text.Encoding.UTF8.GetBytes(password);
    byte[] salt = new byte[32];

    FileStream fsCrypt = new FileStream(inputFile, FileMode.Open);
    fsCrypt.Read(salt, 0, salt.Length);

    RijndaelManaged AES = new RijndaelManaged();
    AES.KeySize = 256;
    AES.BlockSize = 128;
    var key = new Rfc2898DeriveBytes(passwordBytes, salt, 50000);
    AES.Key = key.GetBytes(AES.KeySize / 8);
    AES.IV = key.GetBytes(AES.BlockSize / 8);
    AES.Padding = PaddingMode.PKCS7;
    AES.Mode = CipherMode.CFB;

    CryptoStream cs = new CryptoStream(fsCrypt, AES.CreateDecryptor(),
    CryptoStreamMode.Read);

    FileStream fsOut = new FileStream(outputFile, FileMode.Create);
    // FileStream fsOut = new FileStream(outputFile + ".zip", FileMode.Create);

    int read;
    byte[] buffer = new byte[1048576];

    try
    {
        while ((read = cs.Read(buffer, 0, buffer.Length)) > 0)
        {
            Application.DoEvents();
            fsOut.Write(buffer, 0, read);
        }
    }
}

```

```

        }
    }
    catch (CryptographicException ex_CryptographicException)
    {
        MessageBox.Show("CryptographicException error: " +
ex_CryptographicException.Message);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message);
    }

    try
    {
        cs.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error by closing CryptoStream: " + ex.Message);
    }
    finally
    {
        fsOut.Close();
        fsCrypt.Close();
    }
}

private void txtpass_Leave(object sender, EventArgs e)
{
    progressBar1.Visible = true;
    progressBar1.Value = 15;

}

private void panel2_Paint(object sender, PaintEventArgs e)
{
}
}
}

```

❖ **RIGHTS.CS**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class Rights : Form
    {
        SqlConnection con = new SqlConnection("Data Source=DESKTOP-
9QTIITB\\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();
        public Rights()
        {
            InitializeComponent();
        }

        public void retrive()
        {
            cmd = new SqlCommand("Select * from rightss ", con);
            con.Open();
            SqlDataReader sdr = cmd.ExecuteReader();
            while (sdr.Read())
            {
                comboBox1.Text = sdr["Drive_Lock"].ToString();
                comboBox2.Text = sdr["Rights"].ToString();
                comboBox3.Text = sdr["Lock_Folder"].ToString();
                comboBox4.Text = sdr["Hide_Folder"].ToString();
                comboBox5.Text = sdr["Encrypt_Folder"].ToString();
                comboBox6.Text = sdr["Two_step"].ToString();
                comboBox7.Text = sdr["Change_Password"].ToString();
            }
            con.Close();
        }

        private void btn_Add_Click(object sender, EventArgs e)
```

```
        {
            cmd = new SqlCommand("update rightss set Drive_Lock='" + comboBox1.Text +
",Lock_Folder='" + comboBox3.Text + ",Hide_Folder='" + comboBox4.Text + ",Encrypt_Folder='" +
comboBox5.Text + ",Two_step='" + comboBox6.Text + ",Change_Password='" + comboBox7.Text +
"', con);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("Rights Update Sucessfully...:", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }

private void Rights_Load(object sender, EventArgs e)
{
    retrive();
}

private void groupBox1_Enter(object sender, EventArgs e)
{
}

}
```

❖ **TWOSTEP.CS**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class TwoStep : Form
    {
        SqlConnection con = new SqlConnection("Data Source=DESKTOP-
9QTIITB\\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();
        private static string correctHash = @"";
        public TwoStep(string getname, string gettype)
        {
            InitializeComponent();
            label2.Text = getname;
            label3.Text = gettype;
        }
        public string OsDrive;
        private void Btnstp1_Click(object sender, EventArgs e)
        {
            OsDrive = Path.GetPathRoot(Environment.SystemDirectory);

            var drives = DriveInfo.GetDrives().Where(drive => drive.IsReady && drive.DriveType ==
DriveType.Removable).ToList();
            if (drives.FirstOrDefault() != null)
            {
                Btnstp1.Enabled = false;
                btnstp3.Enabled = false;
                btnstp2.Enabled = true;
                label4.Text = drives.FirstOrDefault().Name.ToString();
                int i = 1;
                while (i <= 1)
                {

```

```

        if (uussbb.GetUSBDevices().Any(x => drives.ToString() == correctHash))
            i++;
            break;
    }
    string path = OsDrive + "TextFile111.txt";
    richTextBox1.Text = File.ReadAllText(path);
}
else
{
    MessageBox.Show("Please Connect Usb Drive.....", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);

}

private void btnstp2_Click(object sender, EventArgs e)
{
    OsDrive = Path.GetPathRoot(Environment.SystemDirectory);

    var drives = DriveInfo.GetDrives().Where(drive => drive.IsReady && drive.DriveType ==
DriveType.Removable).ToList();
    if (drives.FirstOrDefault() != null)
    {
        MessageBox.Show("Please Disconnect Pendrive...", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
    else
    {
        btnstp2.Enabled = false;
        btnstp3.Enabled = true;
        int q = 1;
        while (q <= 1)
        {

            if (uussbb.GetUSBDevices().Any(x => x.ToString() == correctHash))
                q++;
                break;
        }
        string path = OsDrive + "TextFile111.txt";
        string path1 = OsDrive + "TextFile222.txt";
        richTextBox1.Text = File.ReadAllText(path1);

        string[] names1 = File.ReadAllLines(path);
        string[] names2 = File.ReadAllLines(path1);

        // Create the query. Note that method syntax must be used here.
        IEnumerable<string> differenceQuery = names1.Except(names2);

        // Execute the query.
    }
}

```

```

        foreach (string s in differenceQuery)
            richTextBox2.Text = (s);

    }

}

private void TwoStep_Load(object sender, EventArgs e)
{
    btnstp2.Enabled = false;
    btnstp3.Enabled = false;
    OsDrive = Path.GetPathRoot(Environment.SystemDirectory);
    string path = OsDrive + "TextFile111.txt";
    string path1 = OsDrive + "TextFile222.txt";
    try
    {
        File.Delete(path);
        File.Delete(path1);
    }
    catch
    {

    }

    using (FileStream fs = File.Create(path));
    using (FileStream fs = File.Create(path1));

}

private void btnstp3_Click(object sender, EventArgs e)
{
    if (label3.Text == "Admin")
    {

        cmd = new SqlCommand("UPDATE SuperAdmin set
Twostep_status=@Twostep_status,Two_Step=@Two_Step where Username=@Username ", con);
        cmd.Parameters.AddWithValue("@Twostep_status", 1);
        cmd.Parameters.AddWithValue("@Two_Step", richTextBox2.Text);
        cmd.Parameters.AddWithValue("@Username", label2.Text);
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();

        MessageBox.Show("Two Step Verification Enabled Sucessfully :)", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
        Btnstp1.Enabled = false;
        btnstp2.Enabled = false;
        btnstp3.Enabled = false;
    }
}

```

```

OsDrive = Path.GetPathRoot(Environment.SystemDirectory);
string path = OsDrive + "TextFile111.txt";
string path1 = OsDrive + "TextFile222.txt";
try
{
    File.Delete(path);
    File.Delete(path1);
}
catch
{
}

}
this.Close();

}

else
{

    cmd = new SqlCommand("UPDATE NewUser set
Twostep_status=@Twostep_status,Two_Step=@Two_Step where Username=@Username ", con);
    cmd.Parameters.AddWithValue("@Twostep_status", 1);
    cmd.Parameters.AddWithValue("@Two_Step", richTextBox2.Text);
    cmd.Parameters.AddWithValue("@Username", label2.Text);
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
    MessageBox.Show("Two Step Verification Enabled Sucessfully :)", "",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    Btnstp1.Enabled = false;
    btnstp2.Enabled = false;
    btnstp3.Enabled = false;
    string path = OsDrive + "TextFile111.txt";
    string path1 = OsDrive + "TextFile222.txt";
    try
    {
        File.Delete(path);
        File.Delete(path1);
    }
    catch
    {

    }
    this.Close();
}

}

```

```
        }
    }
}
```

## ❖ FORGETPASS.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Net;
using System.Net.Mail;
using System.Net.NetworkInformation;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class Forgetpass : Form
    {
        SqlConnection con = new SqlConnection("Data Source=DESKTOP-
9QTIITB\\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();
        public Forgetpass()
        {
            InitializeComponent();
        }

        private void guna2TextBox2_TextChanged(object sender, EventArgs e)
        {

        }

        private void btn_Add_Click(object sender, EventArgs e)
        {
            if (comboBox1.SelectedIndex == 0)
            {
```

```

con.Open();
cmd = new SqlCommand("select Email from [SuperAdmin] where Email=@Email", con);
cmd.Parameters.AddWithValue("@Email", TextBox1.Text);
var da = new SqlDataAdapter(cmd);
var dt = new DataTable();
da.Fill(dt);
if ((dt.Rows.Count == 1))
{
    string otpcode;
    Random rand = new Random();
    otpcode = (rand.Next(9999999)).ToString();
    Label1.Text = otpcode;

    try
    {
        MailMessage mail = new MailMessage();
        mail.From = new MailAddress("davidsonr882@gmail.com");
        mail.To.Add(TextBox1.Text);
        mail.IsBodyHtml = true;
        mail.Subject = "Password Recovery...";
        mail.Body = "Your OTP Code is :- " + otpcode;
        SmtpClient smtp = new SmtpClient("smtp.gmail.com");
        smtp.Port = 587;
        smtp.Credentials = new NetworkCredential("Gmail-Id", "Password");
        smtp.EnableSsl = true;

        smtp.Send(mail);
        MessageBox.Show("Email Sent Sucessfully | Check Email... ", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
        TextBox2.Focus();
        comboBox1.Enabled = false;
        TextBox1.Enabled = false;
        BTN_check.Enabled = true;
        TextBox2.Enabled = true;
    }
    catch (SmtpException ex)
    {
        MessageBox.Show(ex.Message);
    }
}
else
    MessageBox.Show("Enter A Valid Email :( ", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    con.Close();

}
else if(comboBox1.SelectedIndex==1)
{

```

```

con.Open();
cmd = new SqlCommand("select Email from [NewUser] where Email=@Email", con);
cmd.Parameters.AddWithValue("@Email", TextBox1.Text);
var da = new SqlDataAdapter(cmd);
var dt = new DataTable();
da.Fill(dt);
if ((dt.Rows.Count == 1))
{
    string otpcode;
    Random rand = new Random();
    otpcode = (rand.Next(9999999)).ToString();
    Label1.Text = otpcode;

    try
    {
        MailMessage mail = new MailMessage();
        mail.From = new MailAddress("Gmail-Id");
        mail.To.Add(TextBox1.Text);
        mail.IsBodyHtml = true;
        mail.Subject = "Password Recovery...";
        mail.Body = "Your OTP Code is :- " + otpcode;
        SmtpClient smtp = new SmtpClient("smtp.gmail.com");
        smtp.Port = 587;
        smtp.Credentials = new NetworkCredential("Gmail-Id", "Passwor");

        smtp.EnableSsl = true;

        smtp.Send(mail);
        MessageBox.Show("Email Sent Sucessfully | Check Email... ", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
        TextBox2.Focus();
        comboBox1.Enabled = false;
        TextBox1.Enabled = false;
        BTN_check.Enabled = true;
        TextBox2.Enabled = true;
    }
    catch (SmtpException ex)
    {
        MessageBox.Show(ex.Message);
    }
}
else
    MessageBox.Show("Enter A Valid Email :( ", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
con.Close();
}
else
{

```

```

        MessageBox.Show("Please select user type :( ", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }

}

private void panel1_Paint(object sender, PaintEventArgs e)
{
}

private void TextBox1_TextChanged(object sender, EventArgs e)
{
}

private void BTN_check_Click(object sender, EventArgs e)
{
    var r = 0;
    if (Label1.Text == TextBox2.Text)
    {

        Newpass qw = new Newpass();
        qw.passingtype = comboBox1.Text;
        qw.passingvalue = TextBox1.Text;
        qw.Show();
        this.Hide();
    }
    else
    {
        MessageBox.Show("Wrong Verification Code :( ", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        r = r + 1;
        if (r == 3)
            this.Close();
    }
}

private void Forgetpass_Load(object sender, EventArgs e)
{
    if (NetworkInterface.GetIsNetworkAvailable())
    {
}

```

```
        else
        {
            MessageBox.Show("Please Connect Internet, To Proceed... :( ", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            this.Close();
        }
        BTN_check.Enabled = false;
        TextBox2.Enabled = false;
    }

private void guna2Button1_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void groupBox1_Enter(object sender, EventArgs e)
{
}

}
```

## ❖ CHANGE PASSWORD.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class Change_Password : Form
    {
        SqlConnection con = new SqlConnection("Data Source=DESKTOP-
9QTIITB\\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();

        public Change_Password(string getname,string gettype)
        {
            InitializeComponent();
            //getting username and usertype
            label9.Text = getname;
            label10.Text = gettype;
        }

        private void guna2Button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void Change_Password_Load(object sender, EventArgs e)
        {
            txtnew.Enabled = false;
            txtretype.Enabled = false;
            btn_Add.Enabled = false;
        }

        private void txtold_TextChanged(object sender, EventArgs e)
        {
            if(label9.Text == "User")
            {
                con.Open();
            }
        }
}
```

```

SqlDataAdapter dt = new SqlDataAdapter("select Password from [NewUser] where
Username=" + label9.Text + "and Password=" + txtold.Text + " ", con);

DataTable d1 = new DataTable();
dt.Fill(d1);
if(d1.Rows.Count == 1)
{

    txtnew.Enabled = true;
    txtretype.Enabled = true;
    btn_Add.Enabled = true;

}
else
{
    txtnew.Enabled = false;
    txtretype.Enabled = false;
    btn_Add.Enabled = false;
}

con.Close();
}
else
{
    con.Open();
    SqlDataAdapter dt = new SqlDataAdapter("select Password from [SuperAdmin] where
Username=" + label9.Text + "and Password=" + txtold.Text + " ", con);

    DataTable d1 = new DataTable();
    dt.Fill(d1);
    if(d1.Rows.Count == 1)
    {

        txtnew.Enabled = true;
        txtretype.Enabled = true;
        btn_Add.Enabled = true;

    }
    else
    {
        txtnew.Enabled = false;
        txtretype.Enabled = false;
        btn_Add.Enabled = false;
    }
}

```

```

        }
        con.Close();
    }

}

private void btn_Add_Click(object sender, EventArgs e)
{
    //updating password
    if(label10.Text == "Admin")
    {
        if(txtnew.Text == txtretype.Text)//checking password
        {
            cmd = new SqlCommand("UPDATE SuperAdmin set Password=@Password where Username=@Username ", con);
            cmd.Parameters.AddWithValue("@Password", txtnew.Text);
            cmd.Parameters.AddWithValue("@Username", label9.Text);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("Password Reset Sucessfully :)", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
            this.Close();
        }
        else
            MessageBox.Show("Password Do Not Match Try Again..... :( ", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        if(txtnew.Text == txtretype.Text)
        {
            cmd = new SqlCommand("UPDATE NewUser set Password=@Password where Username=@Username ", con);
            cmd.Parameters.AddWithValue("@Password", txtnew.Text);
            cmd.Parameters.AddWithValue("@Username", label9.Text);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("Password Reset Sucessfully :)", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
            this.Close();
        }
        else
    }
}

```

```

        MessageBox.Show("Password Do Not Match Try Again..... :( ", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void txtnew_TextChanged(object sender, EventArgs e)
{
    //checking password condition
    if (!(txtnew.TextLength >= 8))
    {

        label6.ForeColor = Color.Red;
        btn_Add.Enabled = false;
    }
    else
    {
        label6.ForeColor = Color.SpringGreen;

    }
//-----
if (!txtnew.Text.Any(char.IsUpper))
{
    label4.ForeColor = Color.Red;
    btn_Add.Enabled = false;
}
else
{
    label4.ForeColor = Color.SpringGreen;

}
//-----
if (!txtnew.Text.Any(char.IsLower))
{
    label3.ForeColor = Color.Red;
    btn_Add.Enabled = false;
}
else
{
    label3.ForeColor = Color.SpringGreen;

}
//-----
if (!txtnew.Text.Any(char.IsNumber))
{
    label5.ForeColor = Color.Red;
    btn_Add.Enabled = false;
}
else
{
}

```

```

        label5.ForeColor = Color.SpringGreen;

    }
//-----
if (txtnew.Text.Contains(" "))
{
    label8.ForeColor = Color.Red;
    btn_Add.Enabled = false;
}
else
{
    label8.ForeColor = Color.SpringGreen;
}

}

//-----
String test_string = txtnew.Text;
if (Regex.IsMatch(test_string, "^[a-zA-Z0-9\\x20]+$"))
{
    label7.ForeColor = Color.Red;
    btn_Add.Enabled = false;
}
else
{
    label7.ForeColor = Color.SpringGreen;
}

}
passcheck();
}

private void passcheck()
{
    if (txtnew.Text == txtretype.Text)
    {
        btn_Add.Enabled = true;
    }
    else
    {
        btn_Add.Enabled = false;
    }
}

private void txtretype_TextChanged(object sender, EventArgs e)
{
    passcheck();
}

```

```
private void panel1_Paint(object sender, PaintEventArgs e)
{
}
}
```

❖ **CRIPT.CS**

```
using System;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Text;

namespace DeadLock
{
    static class Cript
    {

        public static string HashSHA512(this string toEncrypt)
        {
            byte[] toEncryptBytes = Encoding.UTF8.GetBytes(toEncrypt);
            byte[] encryptedBytes;

            using (SHA512 shaManaged = new SHA512Managed())
                encryptedBytes = shaManaged.ComputeHash(toEncryptBytes);

            return (encryptedBytes);
        }
    }
}
```

❖ **USER.CS**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Net;
using System.Net.Mail;
using System.Net.NetworkInformation;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class User : Form
    {
        SqlConnection con = new SqlConnection("Data Source=DESKTOP-
9QTIITB\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();
        public User()
        {
            InitializeComponent();
        }

        private void btnCreateacc_Click(object sender, EventArgs e)
        {
            if (TextBox1.Text != "" && TextBox2.Text != "" && TextBox3.Text != "" && txtpass.Text !=
"" && TextBox5.Text != "" && txtotp.Text != "")
            {
                cmd = new SqlCommand("insert into
NewUser(Name,Email,Username,Password,Twostep_status,Two_Step)
values(@Name,@Email,@Username,@Password,@Twostep_status,@Two_Step)", con);
                con.Open();
                cmd.Parameters.AddWithValue("@Name", TextBox1.Text);
                cmd.Parameters.AddWithValue("@Email", TextBox2.Text);
                cmd.Parameters.AddWithValue("@Username", TextBox3.Text);
                cmd.Parameters.AddWithValue("@Password", txtpass.Text);
                cmd.Parameters.AddWithValue("@Twostep_status", "0");
                cmd.Parameters.AddWithValue("@Two_Step", "0");
                cmd.ExecuteNonQuery();
                MessageBox.Show("User Added Sucessfully ", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
        }
    }
}
```

```

        con.Close();
        Login ss = new Login();
        ss.Show();

        Form1 sq = new Form1();
        sq.Close();
        this.Hide();
    }
    else
    {
        MessageBox.Show("Please Fill All Fields ", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

private void txtpass_TextChanged(object sender, EventArgs e)
{
    groupBox1.Visible = true;

    if (!(txtpass.TextLength >= 8))
    {
        // MessageBox.Show("Password Contain at least 8 Character ");
        label4.ForeColor = Color.Red;
        btnCreateacc.Enabled = false;
    }
    else
    {
        label4.ForeColor = Color.SpringGreen;

    }
//-----
if (!txtpass.Text.Any(char.IsUpper))
{
    label2.ForeColor = Color.Red;
    btnCreateacc.Enabled = false;
}
else
{
    label2.ForeColor = Color.SpringGreen;

}
//-----
if (!txtpass.Text.Any(char.IsLower))
{
    label3.ForeColor = Color.Red;
    btnCreateacc.Enabled = false;
}
else
{
}

```

```

label3.ForeColor = Color.SpringGreen;

}

//-----
if (!txtpass.Text.Any(char.IsNumber))
{
    label5.ForeColor = Color.Red;
    btnCreateacc.Enabled = false;
}
else
{
    label5.ForeColor = Color.SpringGreen;

}
//-----
if (txtpass.Text.Contains(" "))
{
    label8.ForeColor = Color.Red;
    btnCreateacc.Enabled = false;
}
else
{
    label8.ForeColor = Color.SpringGreen;

}

//-----
String test_string = txtpass.Text;
if (Regex.IsMatch(test_string, "^[a-zA-Z0-9\\x20]+$"))
{
    label7.ForeColor = Color.Red;
    btnCreateacc.Enabled = false;
}
else
{
    label7.ForeColor = Color.SpringGreen;

}
//Checking Password and Retype Password
passcheck();
}
private void passcheck()
{
    if (txtpass.Text == TextBox5.Text)
    {
        btnCreateacc.Enabled = true;
    }
}

```

```

        else
    {
        btnCreateacc.Enabled = false;
    }
}
private void txtpass_Leave(object sender, EventArgs e)
{
    groupBox1.Visible = false;
}

private void txtpass_MouseClick(object sender, MouseEventArgs e)
{
    groupBox1.Visible = true;
}

private void User_Load(object sender, EventArgs e)
{

    TextBox1.Enabled = true;
    TextBox2.Enabled = true;
    txtotp.Enabled = true;
    TextBox3.Enabled = false;
    TextBox5.Enabled = false;
    txtpass.Enabled = false;

    if(TextBox1.Text == "" && TextBox2.Text == "" && TextBox3.Text == "" && txtpass.Text
    == "" && TextBox5.Text == "" && txtotp.Text == "")
    {
        btnCreateacc.Enabled = false;
    }
    else
    {
        btnCreateacc.Enabled = true;
    }
    PictureBox3.Visible = false;
    pictureBox1.Visible = false;
    //Checking Internet Connection...
    if(NetworkInterface.GetIsNetworkAvailable())
    {
    }
    else
    {
        MessageBox.Show("Please Connect Internet, To Proceed... :( ", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        //this.Close();
    }
}

```

```

private void TextBox5_TextChanged(object sender, EventArgs e)
{
    passcheck();
}

private void email()
{
    //Generate OTP
    string otpcode;
    Random rand = new Random();
    otpcode = (rand.Next(9999999)).ToString();
    label9.Text = otpcode;

    //Sending Mail
    try
    {
        TextBox1.Enabled = false;
        TextBox2.ReadOnly = true;

        TextBox3.Enabled = false;
        TextBox5.Enabled = false;
        txtpass.Enabled = false;

        MailMessage mail = new MailMessage();
        mail.From = new MailAddress("Gmail-Id");
        mail.To.Add(TextBox2.Text);
        mail.IsBodyHtml = true;
        mail.Subject = "OTP Verification...";
        mail.Body = "Your OTP Code is :- " + otpcode;
        SmtpClient smtp = new SmtpClient("smtp.gmail.com");
        smtp.Port = 587;
        smtp.Credentials = new NetworkCredential("Gmail-Id", "Password");
        smtp.EnableSsl = true;
        smtp.Send(mail);

        MessageBox.Show("Email Sent Sucessfully | Check Email... ", "", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message);
    }
}

private void txtotp_TextChanged(object sender, EventArgs e)
{
    pictureBox1.Visible = false;
}

```

```

private void PictureBox3_Click(object sender, EventArgs e)
{
    //Validating OTP

    if (label9.Text == txtotp.Text)
    {
        MessageBox.Show("OTP Verified Sucessfully ", "", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        TextBox1.Enabled = true;
        TextBox2.ReadOnly = true;
        txtotp.ReadOnly = true;
        TextBox3.Enabled = true;
        TextBox5.Enabled = true;
        txtpass.Enabled = true;
        PictureBox3.Visible = false;

    }
    else
    {
        MessageBox.Show("Wrong Verification Code :( ", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        txtotp.Clear();
    }
}

private void pictureBox1_Click(object sender, EventArgs e)
{
    if (TextBox2.Text.Contains("@gmail.com"))
    {
        MessageBox.Show("Please Enter OTP To Continue ", "", MessageBoxButtons.OK,
        MessageBoxIcon.Question);
        email();
        PictureBox3.Visible = true;
        //MessageBox.Show("Please Check Your Gmail ", "", MessageBoxButtons.OK,
        MessageBoxIcon.Information);

    }
    else
    {
        MessageBox.Show("Please Enter Only Gmail Id ", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

private void TextBox2_TextChanged(object sender, EventArgs e)
{
    pictureBox1.Visible = true;
}

```

```

}

private void picshowpass_Click(object sender, EventArgs e)
{
    if (txtpass.UseSystemPasswordChar == false && TextBox5.UseSystemPasswordChar == false)
    {
        txtpass.UseSystemPasswordChar = true;
        TextBox5.UseSystemPasswordChar = true;
    }
    else
    {
        txtpass.UseSystemPasswordChar = false;
        TextBox5.UseSystemPasswordChar = false;
    }
}

private void TextBox3_Validate(object sender, EventArgs e)
{
    cmd = new SqlCommand("select Username from NewUser where Username=@Username",
con);
    cmd.Parameters.AddWithValue("@Username", TextBox3.Text);
    con.Open();
    var da = new SqlDataAdapter(cmd);
    var dt = new DataTable();
    da.Fill(dt);
    if ((dt.Rows.Count == 0))
        btnCreateacc.Enabled = true;
    else
    {
        MessageBox.Show("Username Already Exist | Take Another One", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        btnCreateacc.Enabled = false;
    }
    con.Close();
}

private void TextBox2_Leave(object sender, EventArgs e)
{
    cmd = new SqlCommand("select Email from NewUser where Email=@Email", con);
    cmd.Parameters.AddWithValue("@Email", TextBox3.Text);
    con.Open();
    var da = new SqlDataAdapter(cmd);
    var dt = new DataTable();
    da.Fill(dt);
    if ((dt.Rows.Count == 0))
        btnCreateacc.Enabled = true;
    else
    {
}

```

```

        MessageBox.Show("Email Already Exist | Take Another One", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        btnCreateacc.Enabled = false;
    }
    con.Close();
}

private void panel2_Paint(object sender, PaintEventArgs e)
{
}
}
}
}

```

❖ **NEWPASS.CS**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class Newpass : Form
    {
        SqlConnection con = new SqlConnection("Data Source=DESKTOP-
9QTIITB\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();
        public Newpass()
        {
            InitializeComponent();
        }

        public string mail, type;
        public string passingvalue
        {

            get { return mail; }
            set { mail = value; }
        }
    }
}

```

```

        }

    public string passingtype
    {
        get { return type; }
        set { type = value; }

    }
    private void btn_Add_Click(object sender, EventArgs e)
    {
        if (label2.Text == "Admin")
        {
            if ((TextBox3.Text == TextBox2.Text))
            {
                cmd = new SqlCommand("UPDATE SuperAdmin set Password=@Password where Email=@Email ", con);
                cmd.Parameters.AddWithValue("@Password", TextBox3.Text);
                cmd.Parameters.AddWithValue("@Email", label1.Text);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Password Reset Sucessfully :)", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                this.Hide();
                Login sw = new Login();
                sw.Show();
            }
            else
                MessageBox.Show("Password Do Not Match Try Again..... :( ", "", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
        {
            if ((TextBox3.Text == TextBox2.Text))
            {
                cmd = new SqlCommand("UPDATE NewUser set Password=@Password where Email=@Email ", con);
                cmd.Parameters.AddWithValue("@Password", TextBox3.Text);
                cmd.Parameters.AddWithValue("@Email", label1.Text);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Password Reset Sucessfully :)", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                this.Hide();
                Login sw = new Login();
                sw.Show();
            }
        }
    }
}

```

```

        else
            MessageBox.Show("Password Do Not Match Try Again..... :( ", "", 
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

}

private void Newpass_Load(object sender, EventArgs e)
{
    label2.Text = "" + type;
    label1.Text = "" + mail;
    btn_Add.Enabled = false;
}

private void TextBox3_TextChanged(object sender, EventArgs e)
{

    if (!(TextBox3.TextLength >= 8))
    {
        // MessageBox.Show("Password Contain at least 8 Character ");
        label6.ForeColor = Color.Red;
        btn_Add.Enabled = false;
    }
    else
    {
        label6.ForeColor = Color.SpringGreen;

    }
//-----
    if (!TextBox3.Text.Any(char.IsUpper))
    {
        label4.ForeColor = Color.Red;
        btn_Add.Enabled = false;
    }
    else
    {
        label4.ForeColor = Color.SpringGreen;

    }
//-----
    if (!TextBox3.Text.Any(char.IsLower))
    {
        label3.ForeColor = Color.Red;
        btn_Add.Enabled = false;
    }
    else
    {

```

```

label3.ForeColor = Color.SpringGreen;

}

//-----
if (!TextBox3.Text.Any(char.IsNumber))
{
    label5.ForeColor = Color.Red;
    btn_Add.Enabled = false;
}
else
{
    label5.ForeColor = Color.SpringGreen;

}
//-----
if (TextBox3.Text.Contains(" "))
{
    label8.ForeColor = Color.Red;
    btn_Add.Enabled = false;
}
else
{
    label8.ForeColor = Color.SpringGreen;

}

//-----
String test_string = TextBox3.Text;
if (Regex.IsMatch(test_string, "^[a-zA-Z0-9\x20]+$"))
{
    label7.ForeColor = Color.Red;
    btn_Add.Enabled = false;
}
else
{
    label7.ForeColor = Color.SpringGreen;

}
passcheck();
}

private void passcheck()
{
    if (TextBox3.Text == TextBox2.Text)
    {
        btn_Add.Enabled = true;
    }
    else

```

```

        {
            btn_Add.Enabled = false;
        }
    }

private void TextBox2_TextChanged(object sender, EventArgs e)
{
    passcheck();
}

private void panel1_Paint(object sender, PaintEventArgs e)
{
}

private void guna2Button1_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}
}

```

❖ **PROGRAM.CS**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

```

❖ **SECURITYKEY.CS**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    public partial class securitykey : Form
    {
        SqlConnection con = new SqlConnection("Data Source=DESKTOP-
9QTIITB\\SQLEXPRESS;Initial Catalog=DeadLock;Integrated Security=True");
        SqlCommand cmd = new SqlCommand();
        private static string _correctHash ;
        public securitykey()
        {
            InitializeComponent();
        }
        public string name, type;
        public string passingvalue
        {

            get { return name; }
            set { name = value; }

        }
        public string passingtype
        {

            get { return type; }
            set { type = value; }

        }
        private void guna2Button1_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void fetchday()
```

```

{
    if (label3.Text == "Admin")
    {
        try
        {
            cmd = new SqlCommand("Select * from SuperAdmin where Username = '" + label2.Text
+ "'", con);
            con.Open();
            SqlDataReader sdr = cmd.ExecuteReader();
            while (sdr.Read())
            {
                _correctHash = sdr["Two_Step"].ToString();

            }

            con.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
    else
    {
        try
        {
            cmd = new SqlCommand("Select * from NewUser where Username = '" + label2.Text + "'",
con);
            con.Open();
            SqlDataReader sdr = cmd.ExecuteReader();
            while (sdr.Read())
            {
                _correctHash = sdr["Two_Step"].ToString();

            }

            con.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```

```
private void btnHide_Click(object sender, EventArgs e)
{
    while (true)
    {
        if (uussbbb.GetUSBDevices().Any(x => x.ToString() == _correctHash))
            break;
        Thread.Sleep(1000);
    }
    Master2 sq = new Master2();
    sq.passingtype = label3.Text;
    sq.passingvalue = label2.Text;
    sq.Show();
    this.Hide();
}

private void panel1_Paint(object sender, PaintEventArgs e)
{
}

private void securitykey_Load(object sender, EventArgs e)
{
    label3.Text = "" + type;
    label2.Text = "" + name;
    fetchday();
}

}
```

❖ **USB.CS**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Management;
using System.Runtime.InteropServices;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DeadLock
{
    class uussbbb
    {

        public uussbbb()
        {

        }

        public uussbbb(string deviceId, string pnpDeviceId)
        {
            DeviceId = deviceId;
            PnpDeviceId = pnpDeviceId;
            string a = (DeviceId + PnpDeviceId).HashSHA512();

            var drives = DriveInfo.GetDrives().Where(drive => drive.IsReady && drive.DriveType ==
DriveType.Removable).ToList();
            if (drives.FirstOrDefault() != null)
            {

                string path = OsDrive + "TextFile111.txt";
                using (StreamWriter writer = new StreamWriter(path, true)) /// true to append data to the
file
                {
                    writer.WriteLine(a);
                }
            }
            else
            {
            }
        }
    }
}
```

```
string path1 = OsDrive + "TextFile222.txt";

// using (FileStream fs = File.Create(path1))
using (StreamWriter writer = new StreamWriter(path1, true)) /// true to append data to the
{
    writer.WriteLine(a);
}

}

public override string ToString()
{
    return (DeviceId + PnpDeviceId).HashSHA512();
}

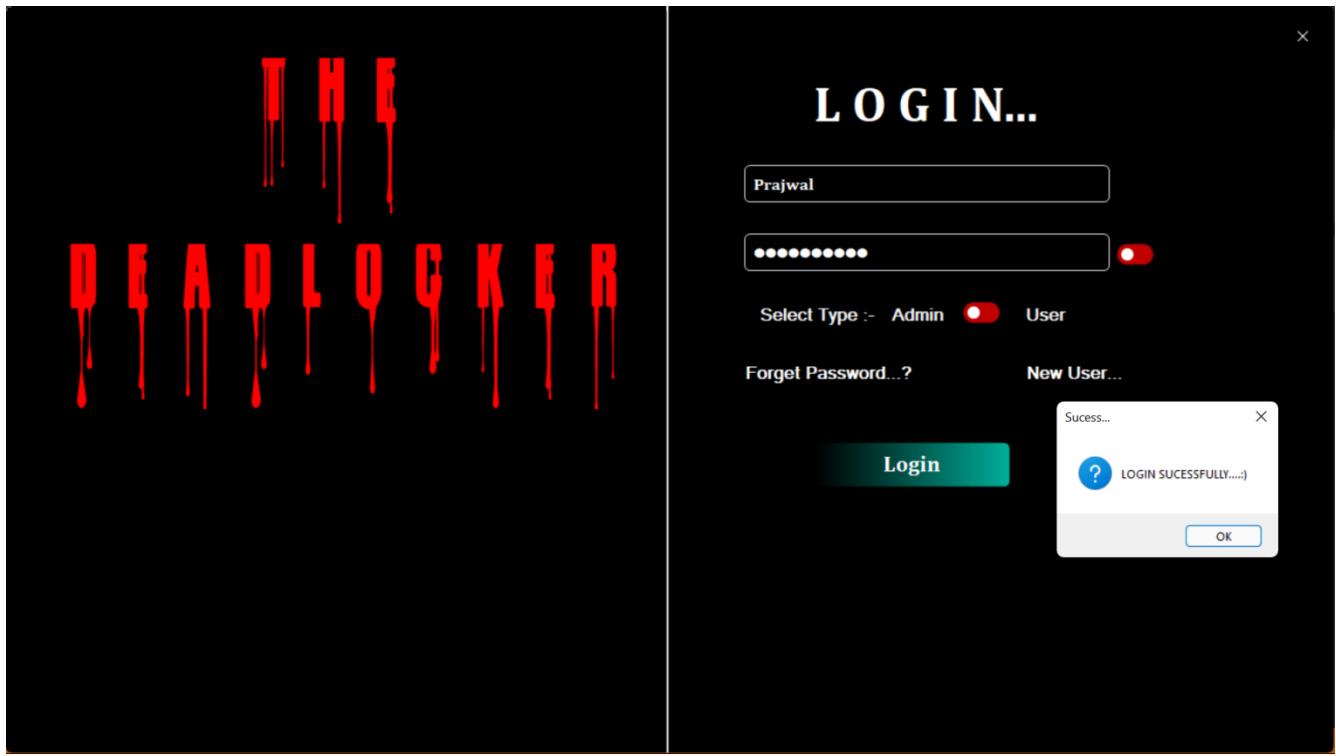
));

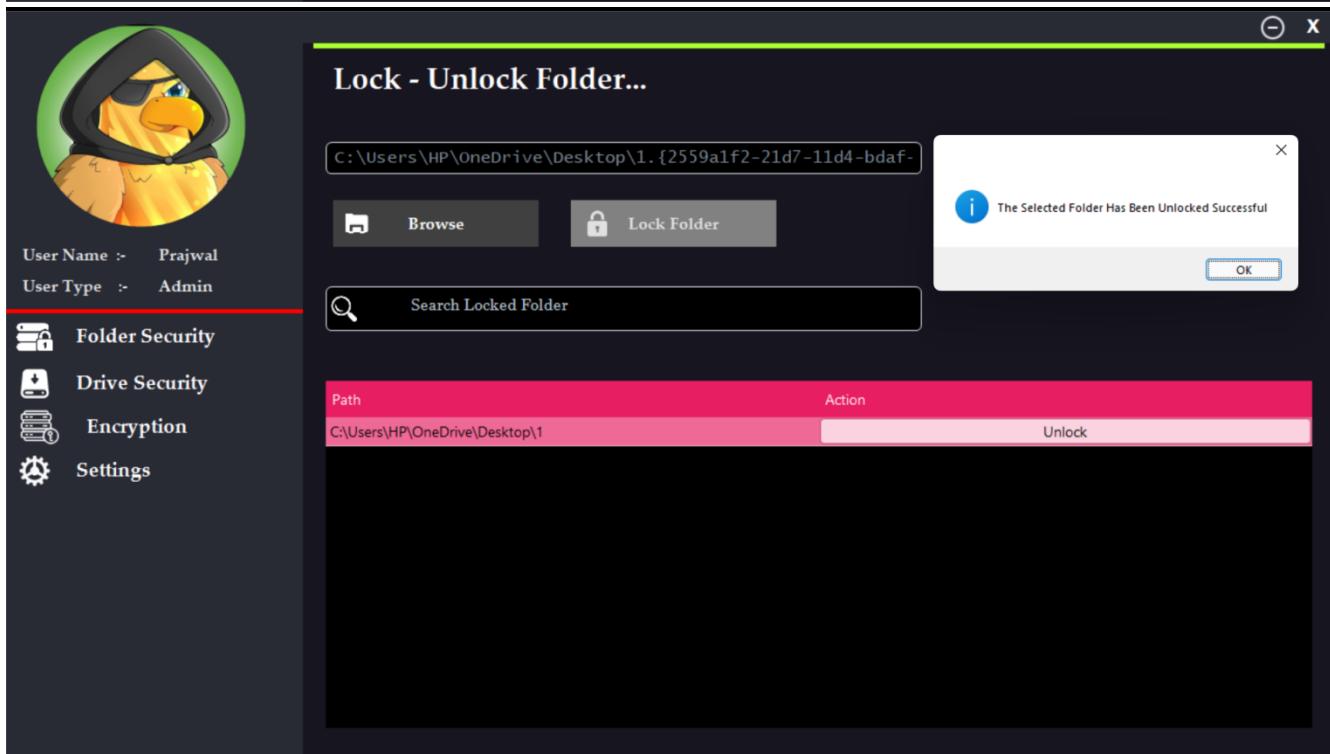
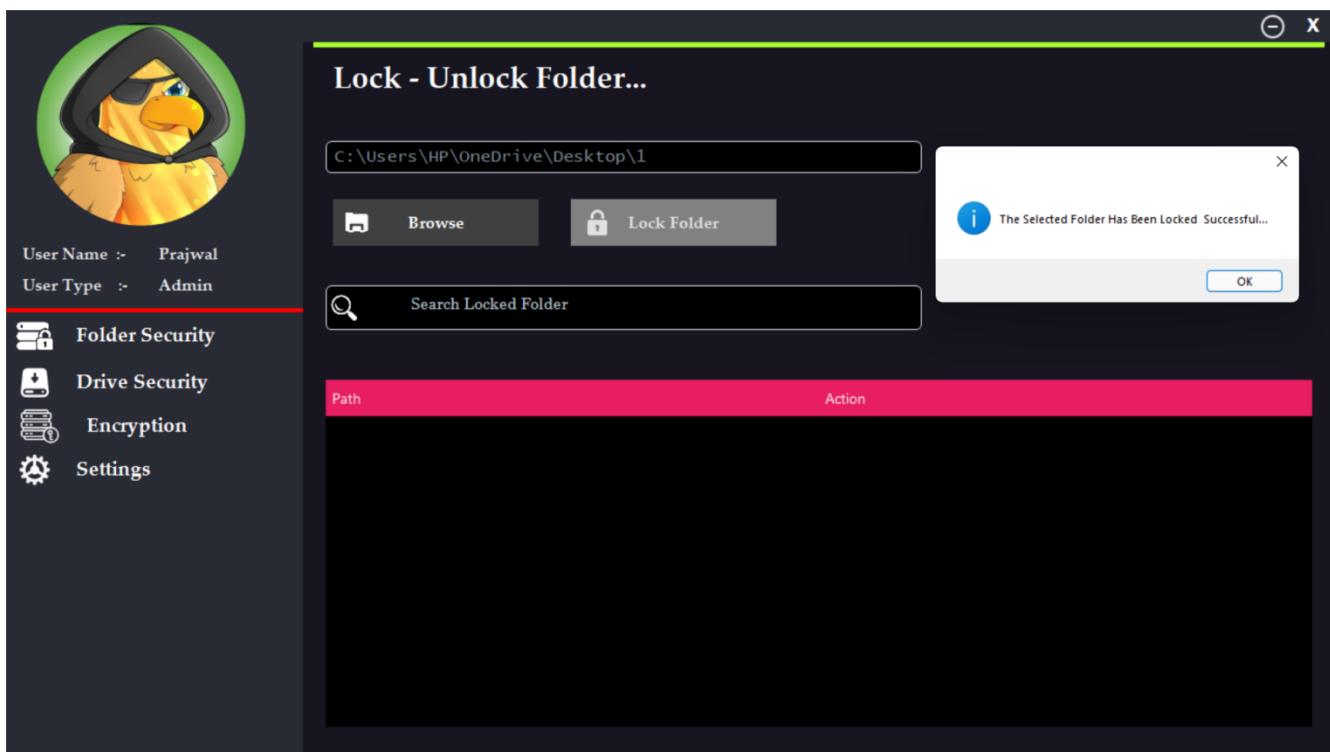
collection.Dispose();
return devices;
}

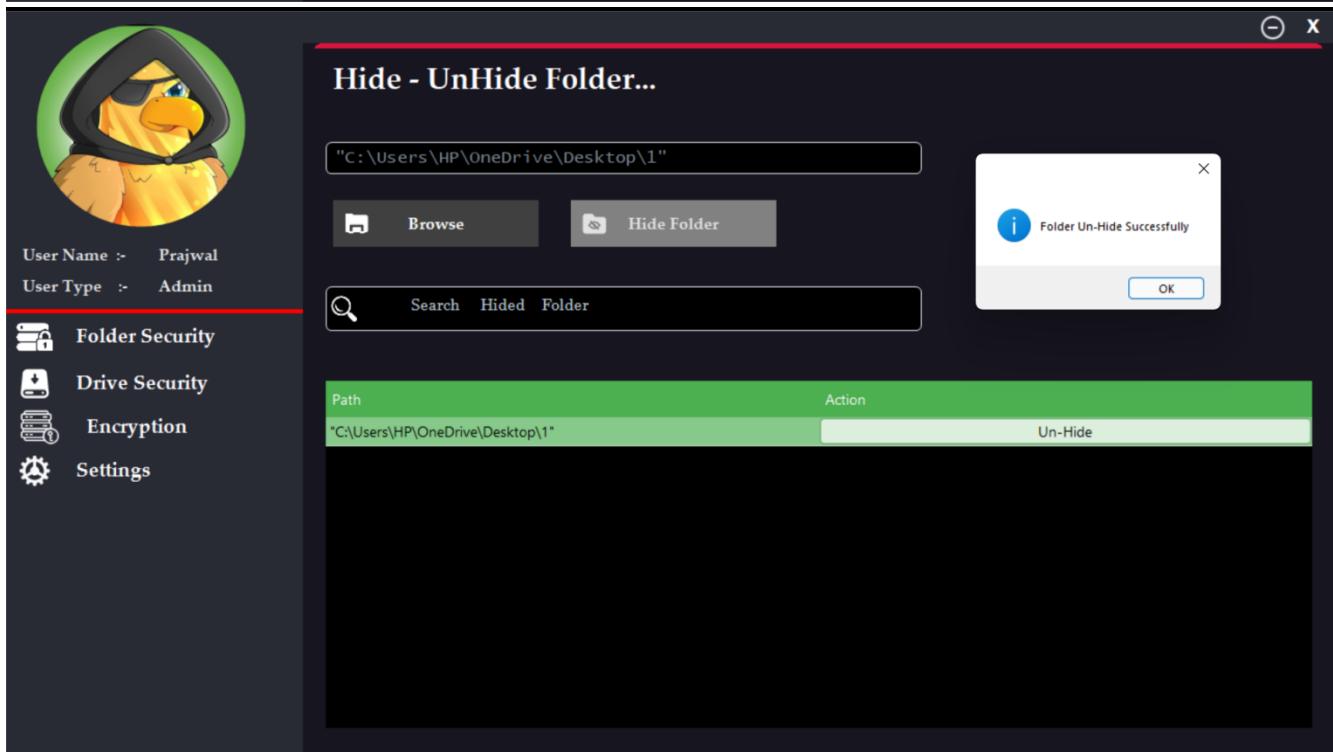
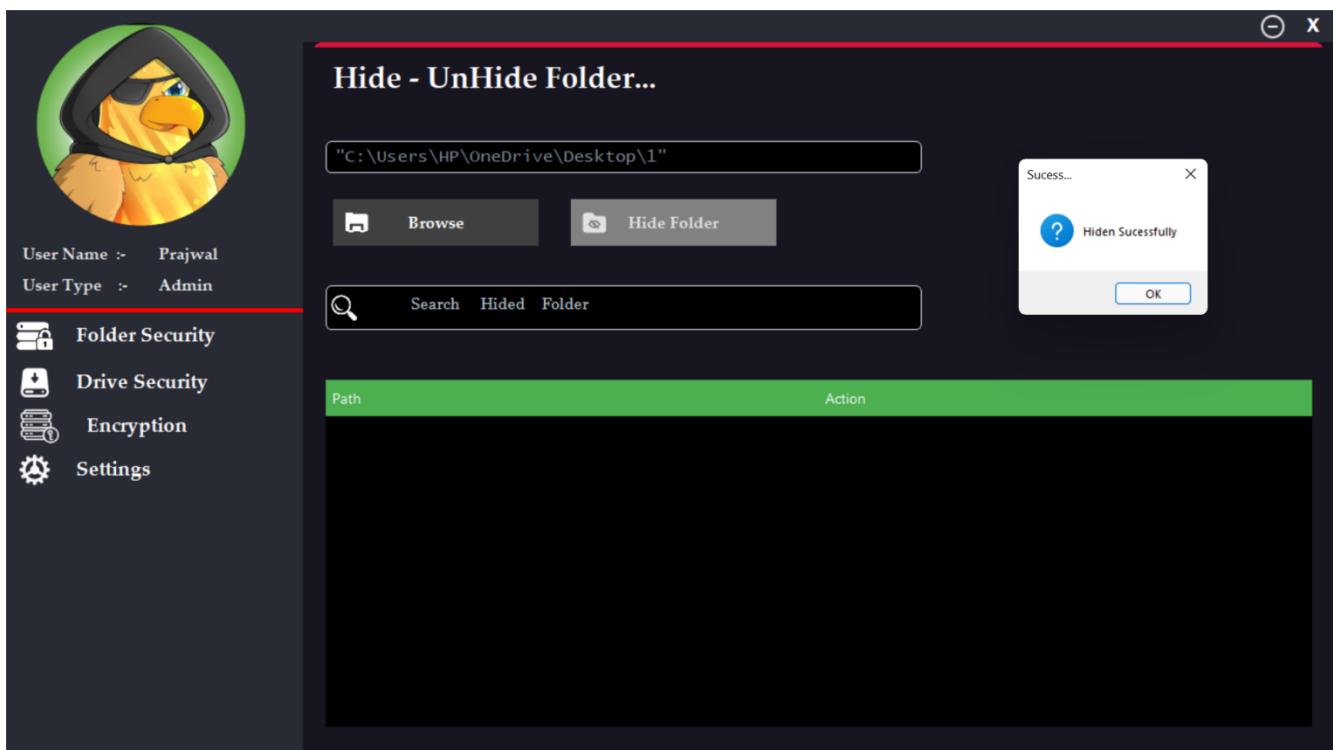
}

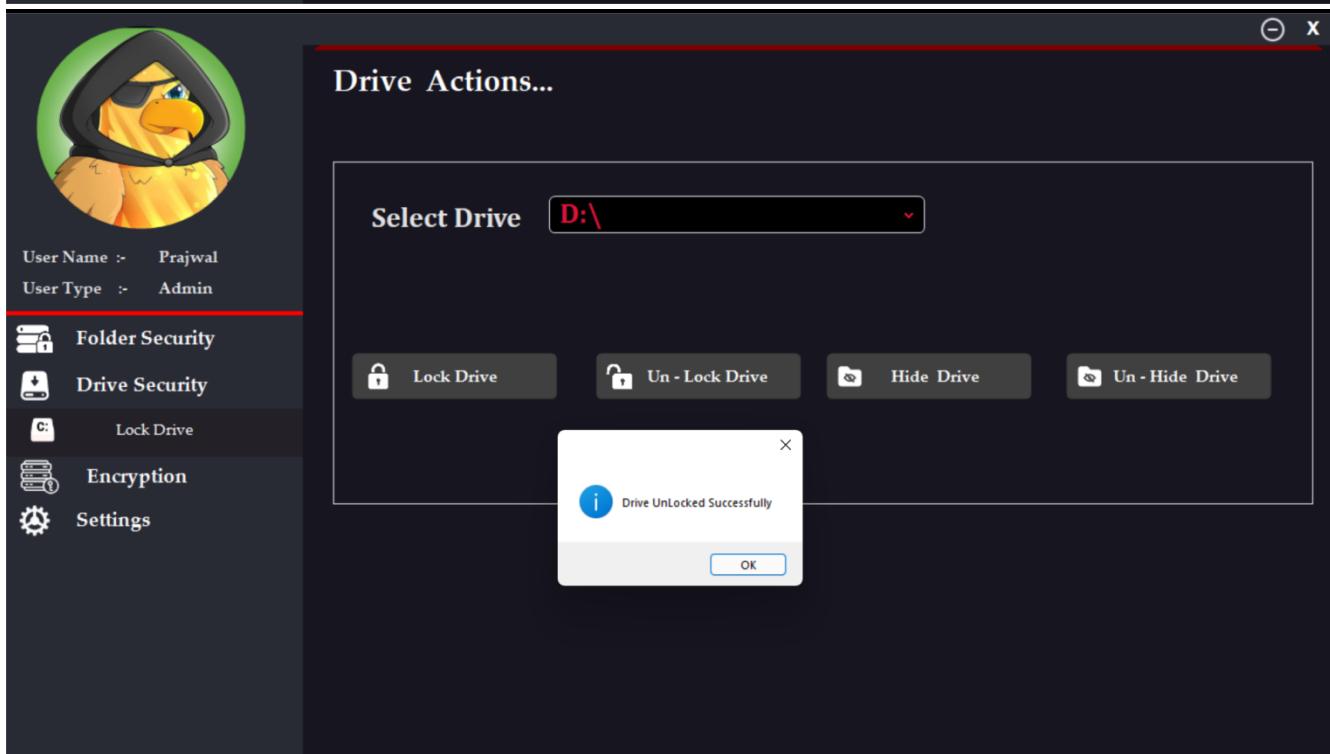
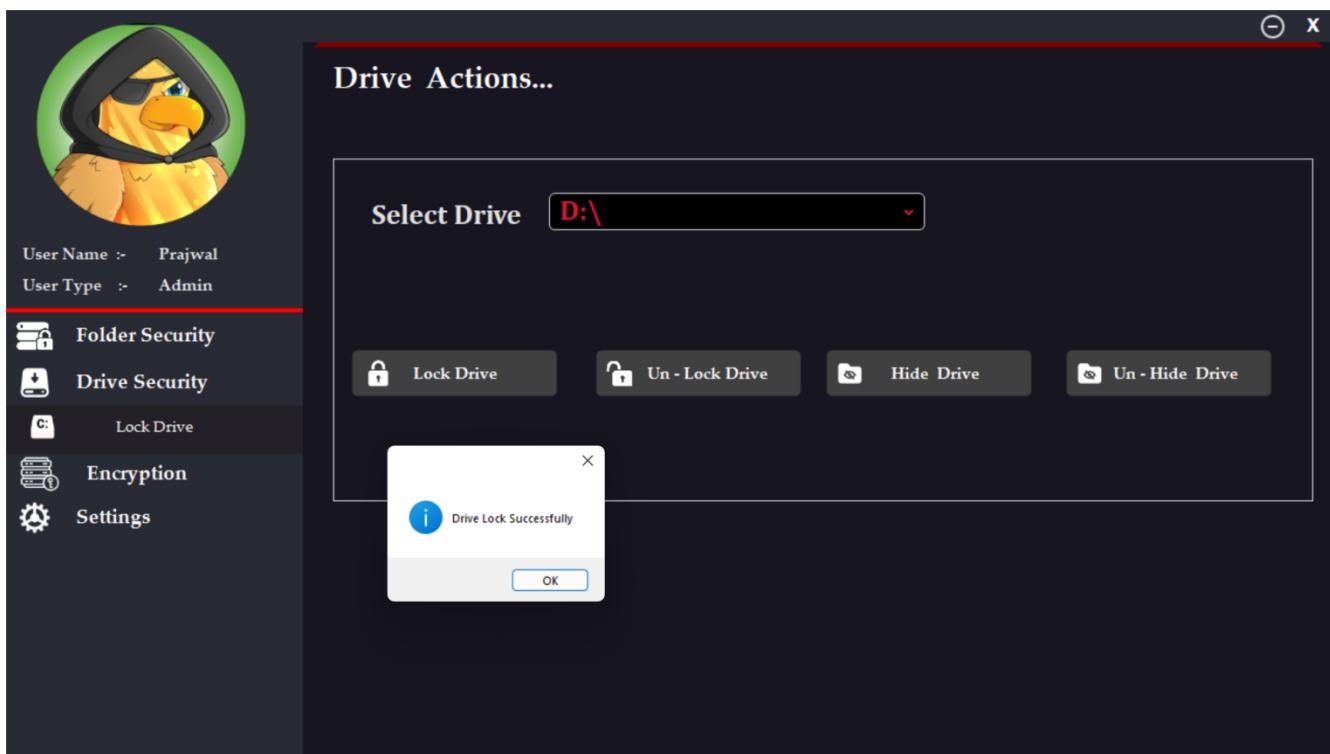
}
```

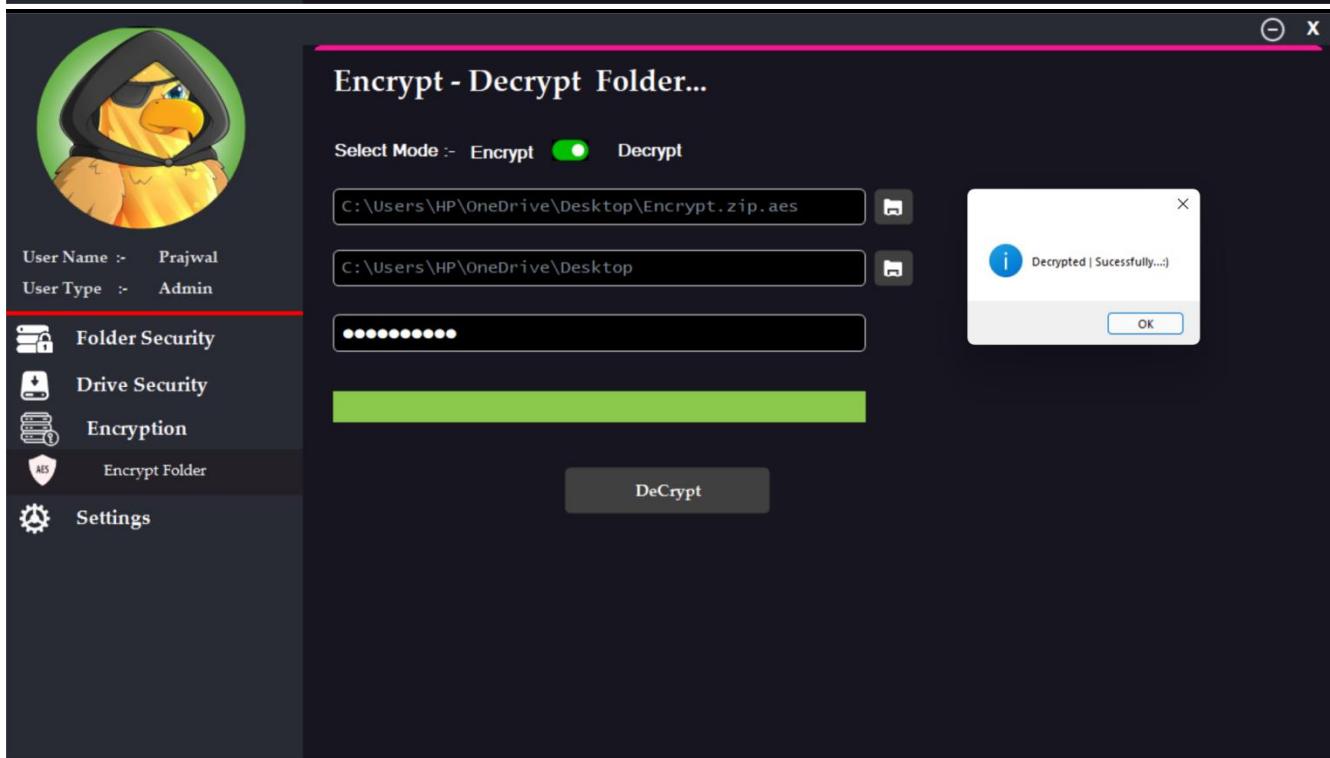
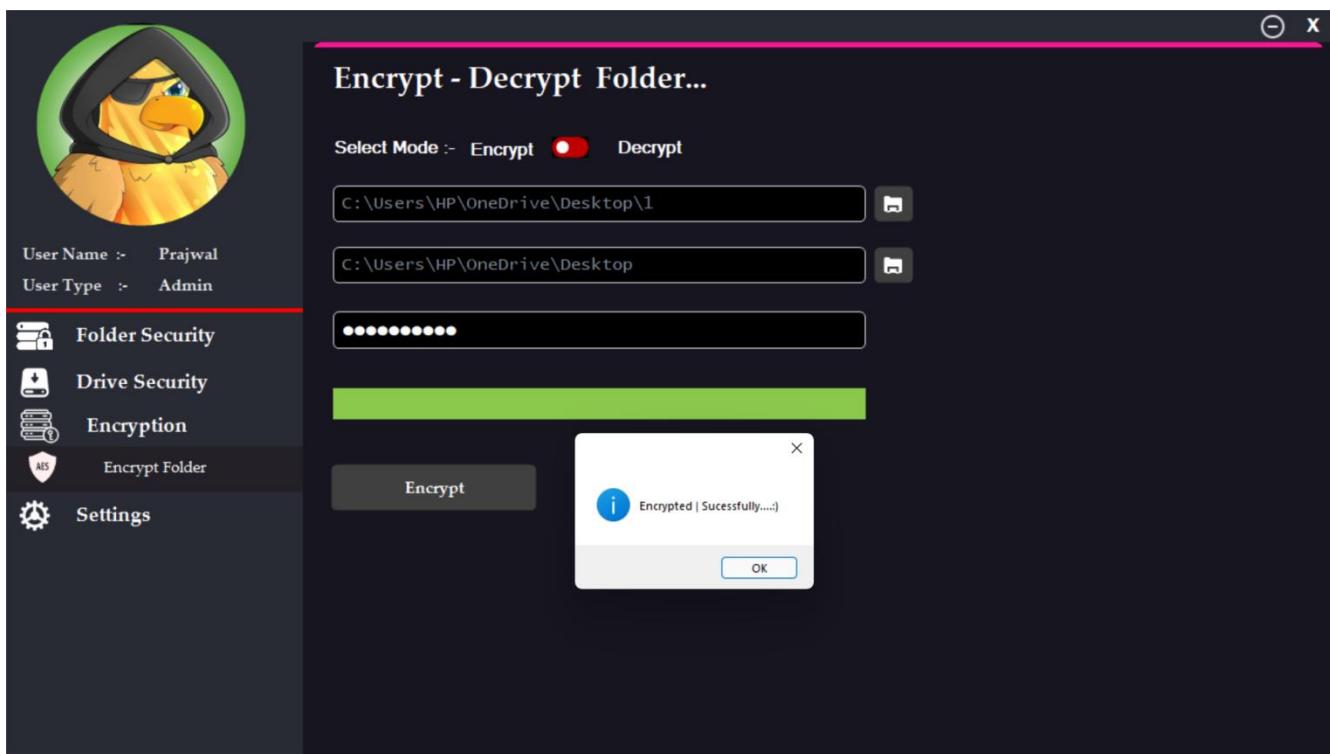
## OUTPUTSCREENS













User Name :- Prajwal  
User Type :- Admin

**Folder Security**

**Drive Security**

**Encryption**

**Settings**

**Rights**

**TwoStep Verification**

**Change Password**

### Provide Rights To User...

Select Rights

Lock Folder	Allow	Encrypt Folder	Deny
Hide Folder	Deny	TwoStep Verification	Deny
Lock Drive	Deny	Change Password	Deny
Rights		Deny	

**OK**

**Save**

 Rights Update Sucessfully...:)



User Name :- Prajwal  
User Type :- Admin

**Folder Security**

**Drive Security**

**Encryption**

**Settings**

### Generate Security Key...

USB Drive Name :- F:\

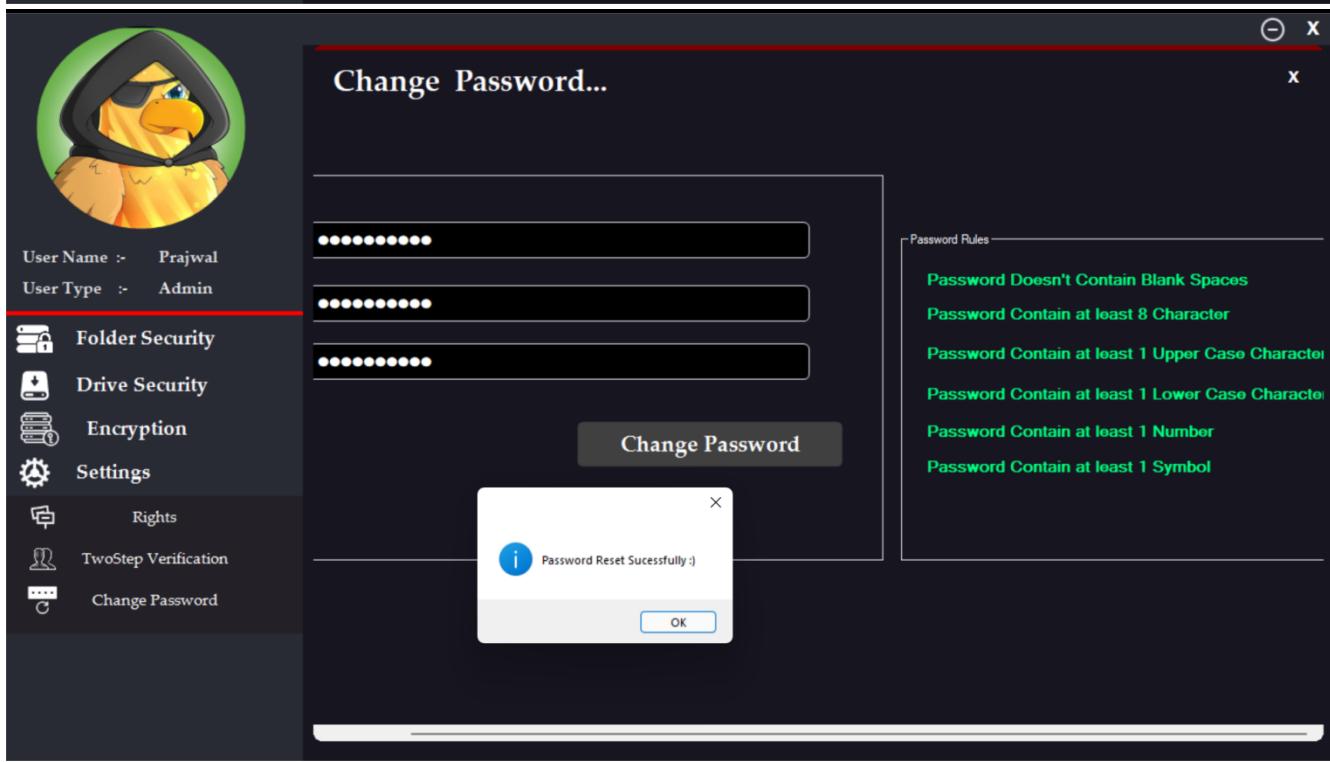
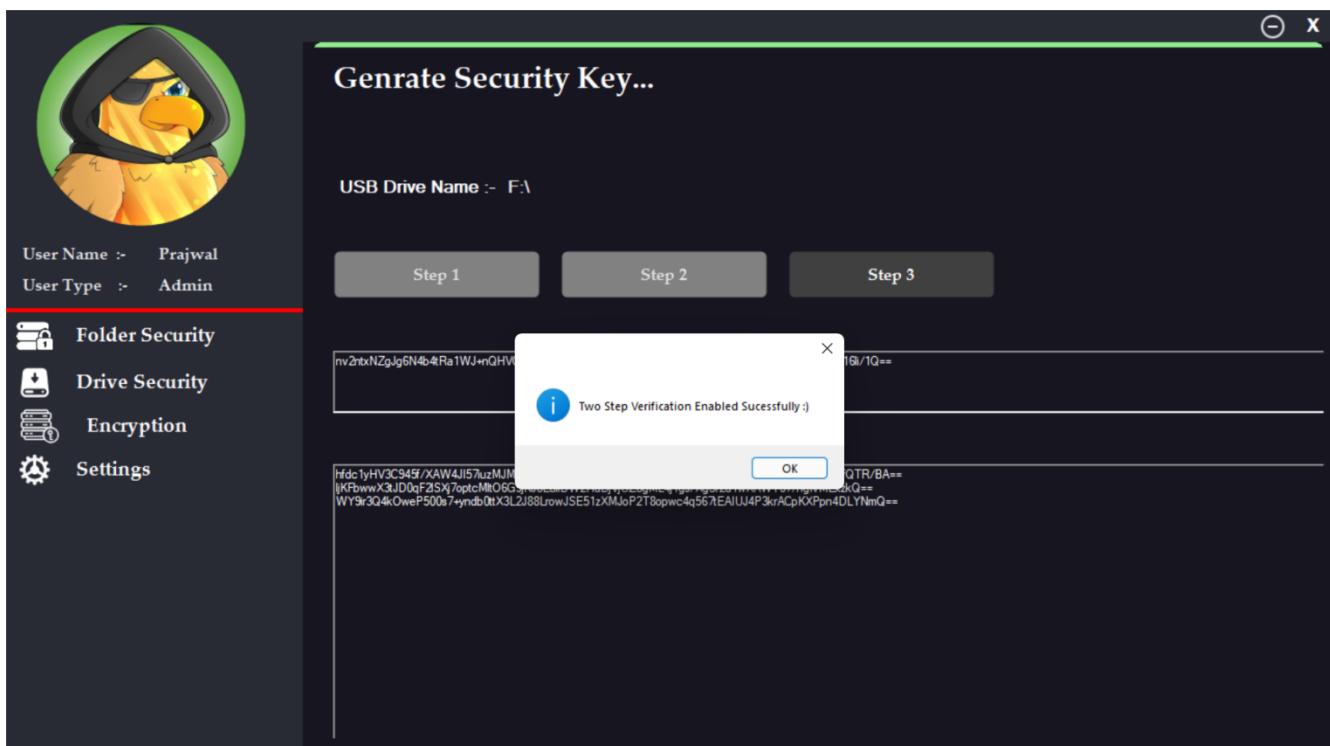
Step 1 Step 2 Step 3

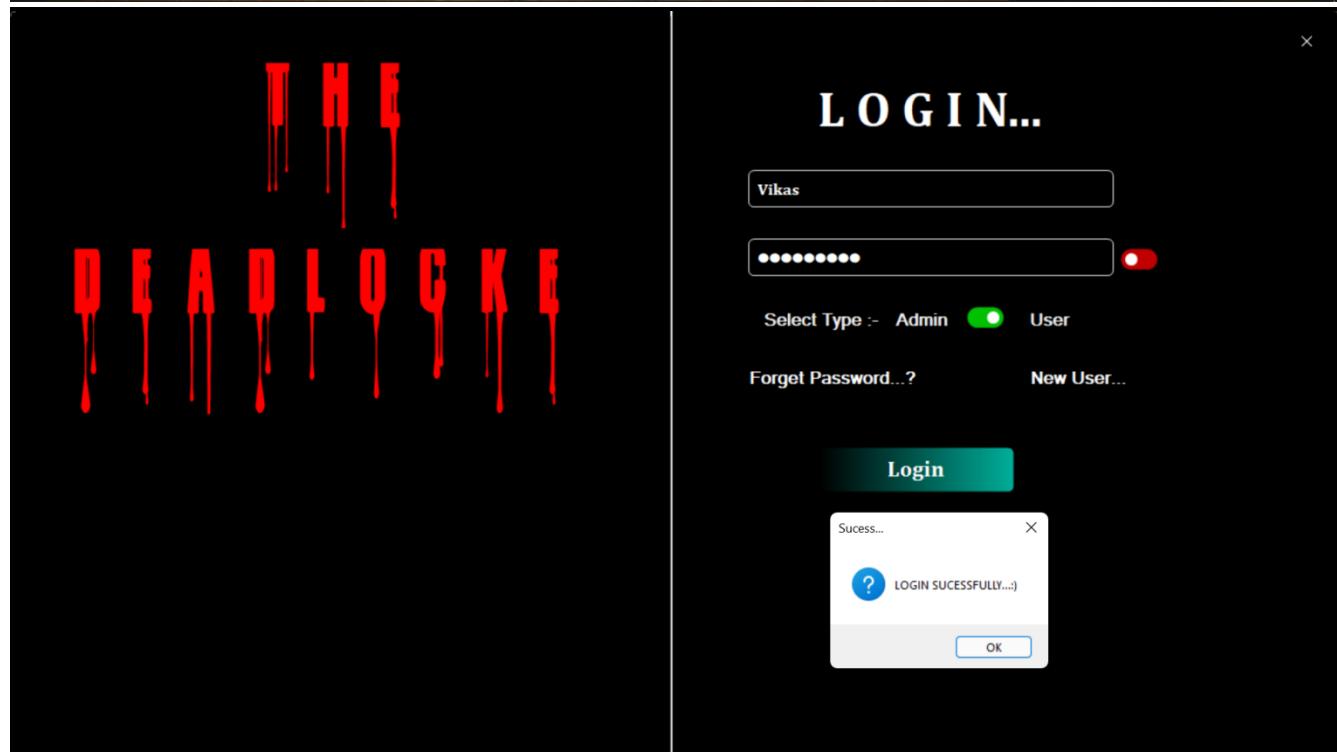
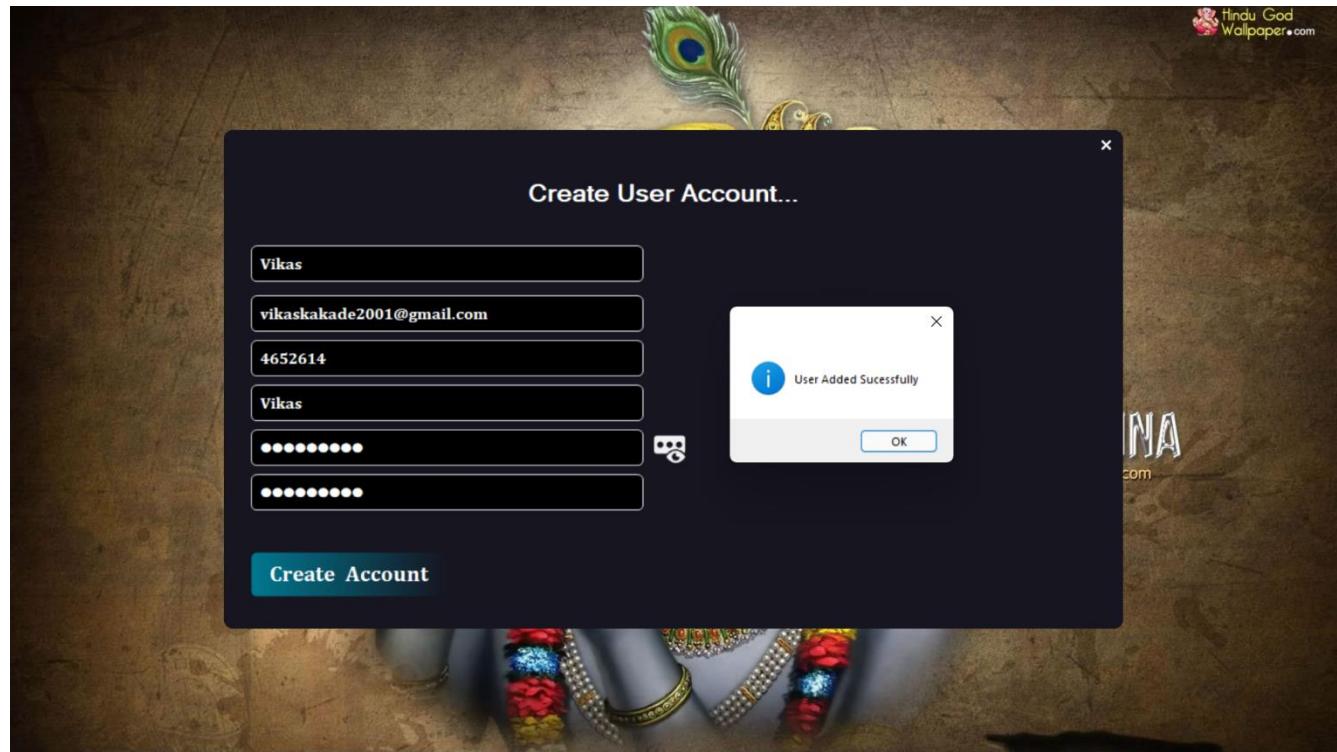
**Error**

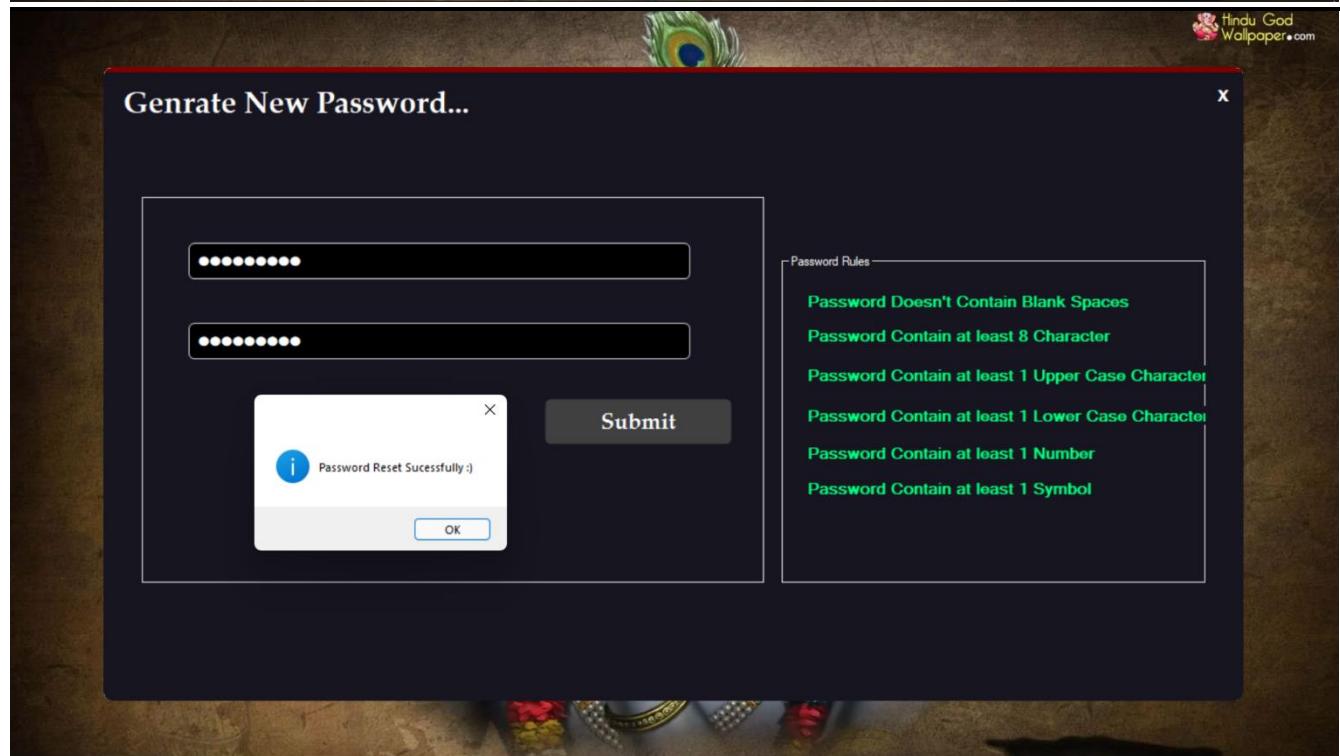
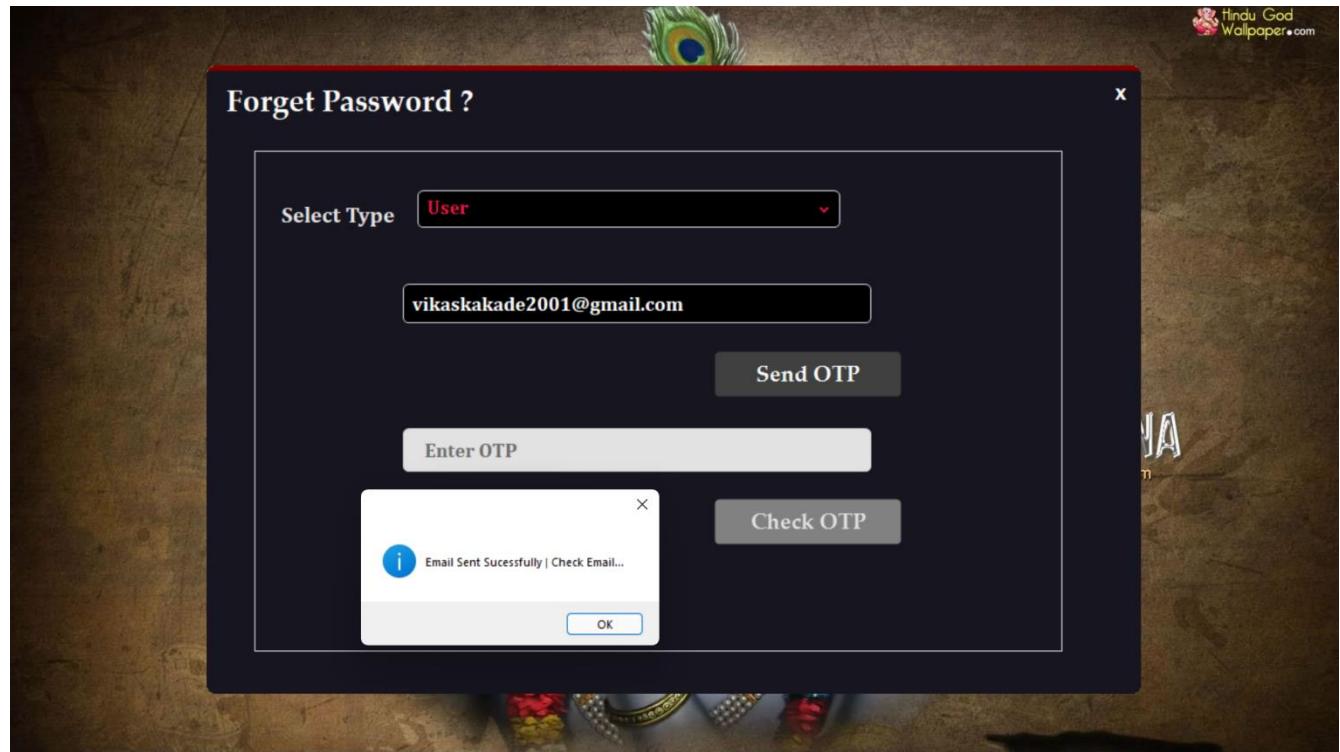
Please Disconnect Pendrive...

nv2nbxNzgJg6N4b4Ra1WJ-nnQHVOzxiWE  
hdc:yHV3C949/XAW4i5I7uzJMzbTfKS67S0mLcAjgV3HvEY7m3G6oD10smtWV35XReTQTR/BA==  
|KfbwwX3JD0gF2SXj7optcMnO6G9KJ8EuDW2HdDlUJZbgM\_4Igs/YgSr2d1wXAWY67/hgVmExzkQ==  
WY9r3Q4kOweP500s7+ynob0tX3L2J88RowJSE51zXMjoP2g8opwc4q567EAUJ4P3rACpKXPPn4DLYNmQ==

**OK**







## **G.CONCLUSION**

This is the new era of technology and we are providing you with the new facilities and the new innovative way for learning. Now-a-days, Students are becoming more faster than the technology so this is the new way for the students to practice and learn new things and explore themselves.

This will help students to improve their syllabus and also extra knowledge which is needed for them. This project is to satisfy students need to manage their self-study material. User friendly design and code is adopted here. The objective of my project is to provide bunch of notes which include all data useful for three years degree course.

## ***H. BIBLIOGRAPHY***

- ➔ C# Manual- docs.microsoft.com
- ➔ C# Refrance :- <https://www.javatpoint.com/c-sharp-tutorial>
- ➔ <https://www.ibm.com/topics/data-security>
- ➔ Lynda.com - Learning C#
- ➔ Tutorials Point - Basic and Advanced C#
- ➔ <https://www.youtube.com/watch?v=7AbgM9rNc00> - Hacked
- ➔ <https://youtu.be/SuLiu5AK9Ps> - CodeWithHarry