|  |
| --- |
| **Web design with solid fill**  **SOURCE CODE** |
| * **FORM1**   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using Library\_System;  using Library\_System.Forms;  using Library\_System.Data\_Model;  namespace Library\_System  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  User.Deserialize();  Student.Deserialize();  Author. Deserialize();  }  private void userToolStripMenuItem\_Click(object sender, EventArgs e)  {  ListofUsers u = new ListofUsers();  u.ShowDialog();  }  private void Form1\_Load(object sender, EventArgs e) // Kapag nagload na yung form, Una itong bubukas, yung LOGIN form  {  Login login = new Login();  if (login.ShowDialog() != DialogResult.OK) //Kapag Hindi OK ang pinili, magkoClose ang Application  Application.Exit();  }  private void authorToolStripMenuItem\_Click(object sender, EventArgs e)  {  ListOfAuthor a = new ListOfAuthor();  a.ShowDialog();  }  private void studentsToolStripMenuItem\_Click(object sender, EventArgs e)  {  ListOfStudent s = new ListOfStudent();  s.ShowDialog();  }  }  }   * **UTILITIES**   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace Library\_System  {  internal class Utilities  {  //Purpose: Open DialogBox then Pick an Image  public static string BrowseImage(PictureBox pb)  //Ang naopen niyang image ay iseset niya into picture box  {  try  {  OpenFileDialog dialog = new OpenFileDialog();  dialog.Filter = "JPEG Files (\*.jpeg)|\*jpg|PNG Files (\*.png)|\*png| JPG Files (\*.jpg)|\*jpg| GIF Files (\*.gif) |\*.gif| All Files (\*.\*)| \*.\* ";  if (dialog.ShowDialog() == DialogResult.OK)  {  string imagePath = dialog.FileName.ToString();  pb.ImageLocation = imagePath; //Yung location na makukuha naten, yun yung Image na iaassign natin kay PictureBox  dialog.Dispose();  return imagePath;  }  else return null;  }  catch { return null; }  }  }  }  **DATA MODELS**   1. **AUTHOR.CS**   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Runtime.Serialization.Formatters.Binary;  using System.IO;  namespace Library\_System.Data\_Model  {  [Serializable]  public class Author  {  public int Id { get; set; } // Unique Id or Primary Key  public string Name { get; set; }  public string About { get; set; }  public static List<Author> ListOfAuthor = new List<Author>();  public static bool Add(Author author) //Means magpasa ka ng OBJECT na ang Parameter Type is User  {  ListOfAuthor.Add(author);  return Serialize();  }  public static bool Update(Author author)  {  foreach (var item in ListOfAuthor.Where(a => a.Id == author.Id)) //Hahanapin yung specific Id para hindi sabay sabay mag Update if ever na Name ang hanapin  //Magiging kinalabasan is WHERE Id ="\_\_\_\_\_"; ganyan parang sa database  {  item.Name = author.Name;  item.About = author.About;    }  return Serialize();  }  public static bool Delete(Author author) //Delete is Method  {  ListOfAuthor.Remove(author);  return Serialize();  }  public static bool Serialize()  {  try  {  FileStream fs = new FileStream("Author", FileMode.Create, FileAccess.Write, FileShare.None);  BinaryFormatter formatter = new BinaryFormatter();  formatter.Serialize(fs, ListOfAuthor);  fs.Dispose();  fs.Close();  return true;  }  catch { return false; }  }  public static bool Deserialize()  {  try  {  if (System.IO.File.Exists("Author")) // Is the file exists? if exists it will deserialize and it will be assigned to the ListofUser  {  FileStream fs = new FileStream("Author", FileMode.Open);  BinaryFormatter formatter = new BinaryFormatter();  ListOfAuthor = ((List<Author>)(formatter.Deserialize(fs)));  fs.Dispose();  fs.Close();  return true;  }  return false;  }  catch { return false; }  }  }  }   1. **USER.CS**   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Runtime.Serialization.Formatters.Binary;  using System.IO;  namespace Library\_System.Data\_Model  {  [Serializable]  public class User // User is an object  {  //PROPERTIES  public int Id { get; set; } // Unique Id or Primary Key  public string Name { get; set; }  public System.Drawing.Image Image { get; set; } = null;  public string Username { get; set; }  public string Password { get; set; }  public int PositionId { get; set; }  public bool Status { get; set; } = true;  //--  public static List<User> ListofUser = new List<User>();  public static bool Add (User user) //Means magpasa ka ng OBJECT na ang Parameter Type is User  {  ListofUser.Add(user);  return Serialize();  }  public static bool Update (User user)  {  foreach(var item in ListofUser.Where(a=>a.Id == user.Id)) //Hahanapin yung specific Id para hindi sabay sabay mag Update if ever na Name ang hanapin  //Magiging kinalabasan is WHERE Id ="\_\_\_\_\_"; ganyan parang sa database  {  item.Name = user.Name;  item.Password = user.Password;  item.PositionId = user.PositionId;  item.Image = user.Image;  item.Status = user.Status;  }  return Serialize();  }  public static bool Delete(User user) //Delete is Method  {  ListofUser.Remove(user);  return Serialize();  }  public static bool Serialize()  {  try  {  FileStream fs = new FileStream("User", FileMode.Create, FileAccess.Write, FileShare.None);  BinaryFormatter formatter = new BinaryFormatter();  formatter.Serialize(fs, ListofUser);  fs.Dispose();  fs.Close();  return true;  }  catch { return false; }  }  public static bool Deserialize()  {  try  {  if (System.IO.File.Exists("User")) // Is the file exists? if exists it will deserialize and it will be assigned to the ListofUser  {  FileStream fs = new FileStream("User", FileMode.Open);  BinaryFormatter formatter = new BinaryFormatter();  ListofUser = ((List<User>)(formatter.Deserialize(fs)));  fs.Dispose();  fs.Close();  return true;  }  return false;  }  catch { return false; }  }  }  }   1. **STUDENT.CS**   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Runtime.Serialization.Formatters.Binary;  using System.IO;  using System.Drawing;  namespace Library\_System.Data\_Model  {  [Serializable]  public class Student  {  public int Id { get; set; } // Unique Id or Primary Key  public Image Image { get; set; }  public string Name { get; set; }  public bool Status { get; set; } = true;  public static List<Student> ListOfStudent = new List<Student>();  public static bool Add(Student student) //Means magpasa ka ng OBJECT na ang Parameter Type is User  {  ListOfStudent.Add(student);  return Serialize();  }  public static bool Update(Student student)  {  foreach (var item in ListOfStudent.Where(a => a.Id == student.Id)) //Hahanapin yung specific Id para hindi sabay sabay mag Update if ever na Name ang hanapin  //Magiging kinalabasan is WHERE Id ="\_\_\_\_\_"; ganyan parang sa database  {  item.Name = student.Name;  item.Image = student.Image;  item.Status = student.Status;  }  return Serialize();  }  public static bool Delete(Student student) //Delete is Method  {  ListOfStudent.Remove(student);  return Serialize();  }  public static bool Serialize()  {  try  {  FileStream fs = new FileStream("Student", FileMode.Create, FileAccess.Write, FileShare.None);  BinaryFormatter formatter = new BinaryFormatter();  formatter.Serialize(fs, ListOfStudent);  fs.Dispose();  fs.Close();  return true;  }  catch { return false; }  }  public static bool Deserialize()  {  try  {  if (System.IO.File.Exists("Student")) // Is the file exists? if exists it will deserialize and it will be assigned to the ListofUser  {  FileStream fs = new FileStream("Student", FileMode.Open);  BinaryFormatter formatter = new BinaryFormatter();  ListOfStudent = ((List<Student>)(formatter.Deserialize(fs)));  fs.Dispose();  fs.Close();  return true;  }  return false;  }  catch { return false; }  }  }  }   1. **BOOK BORROWED**   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Library\_System.Data\_Model  {  [Serializable]  internal class Book\_Borrowed  {  public int Id { get; set; } // Unique Id or Primary Key  public int StudentId { get; set; }  public int UserId { get; set; } //Kung sino yung gumamit ng system na nagpahiram  //Para malaman rin kung sino ang nagpahiram  public int BookId { get; set; }  public DateTime Date { get; set; }  public DateTime DateReturn { get; set; }  public string Remarks { get; set; } //To know if there is a damage in the book  }  }   1. **BOOK CATEGORY.CS**   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Library\_System.Data\_Model  {  [Serializable]  internal class Book\_Category  {  public int Id { get; set; } // Unique Id or Primary Key  public string Name { get; set; }  public string Description { get; set; }  }  }   1. **BOOK.CS**   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Library\_System.Data\_Model  {  [Serializable]  internal class Book  {  public int Id { get; set; } // Unique Id or Primary Key  public string Title { get; set; }  public int AuthorId { get; set; }  public DateTime DatePublished { get; set; }  public string Description { get; set; }  public int CategoryId { get; set; }  public bool Status { get; set; }  }  }   1. **POSITION.CS**   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Library\_System.Data\_Model  {  [Serializable]  internal class Position //Position of User  {  public int Id { get; set; } // Unique Id or Primary Key  public string Name { get; set; }  public string Description { get; set; }  }  }  **FORMS**   1. **ADD EDIT AUTHOR.CS**   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using Library\_System.Data\_Model;  using Library\_System.Forms;  namespace Library\_System.Forms  {  public partial class AddEditAuthor : Form  {  public AddEditAuthor(Author author = null)  {  InitializeComponent();  Author = author;  if (author != null)  DisplaySelectedAuthor();  }  Author Author;  private void btnOk\_Click(object sender, EventArgs e)  {  if (Author == null)  {  Author author = new Author()  {  Id = NewId2(),  Name = tbName2.Text,  About = tbAbout.Text  };    if (Author.Add(author))  DialogResult = DialogResult.OK;  }  else  {  if (Author.Update(Author)) // Mag uupdate  DialogResult = DialogResult.OK; // Pag nag OK mag coclose na siya  }  }      private void btnCancel\_Click(object sender, EventArgs e)  {  this.DialogResult = DialogResult.Cancel; //Automatic close  }  int NewId2()  {  try  {  //x => \_\_\_ (Anong property ang gusto kunin - edi yung Id sa User)  return Author.ListOfAuthor.Max(x => x.Id) + 1; // +1 means everytime na magkecreate ng bago  }  catch { return 0; }  }  void DisplaySelectedAuthor() // Kapag Mag EEDIT, tatawagin ito ---->  {  tbName2.DataBindings.Add("Text", Author, "Name");  tbAbout.DataBindings.Add("Text", Author, "About");  }  }  }   1. **ADD EDIT STUDENT.CS**   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using Library\_System.Data\_Model;  namespace Library\_System.Forms  {  public partial class AddEditStudent : Form  {  public AddEditStudent(Student student = null)  {  InitializeComponent();  Student = student;  if (student != null)  DisplaySelectedStudent();  }  Student Student;  private void btnOK\_Click(object sender, EventArgs e)  {  if (Student == null)  {  Student student = new Student()  {  Id = NewId3(),  Name = tbName3.Text,  Status = cbActive2.Checked, //Boolean  Image = pictureBox1.Image  };  if (Student.Add(student))  DialogResult = DialogResult.OK;  }  else  {  Student.Image = pictureBox1.Image; //Para mag update ren yung Image, the sinet rin yung Iamge na property ni User sa laman ng PictureBox  if (Student.Update(Student)) // Mag uupdate  DialogResult = DialogResult.OK; // Pag nag OK mag coclose na siya  }  }  private void btnCancel\_Click(object sender, EventArgs e)  {  this.DialogResult = DialogResult.Cancel; //Automatic close  }  int NewId3()  {  try  {  //x => \_\_\_ (Anong property ang gusto kunin - edi yung Id sa User)  return Student.ListOfStudent.Max(x => x.Id) + 1; // +1 means everytime na magkecreate ng bago  }  catch { return 0; }  }  void DisplaySelectedStudent() // Kapag Mag EEDIT, tatawagin ito ---->  {    pictureBox1.Image = Student.Image;  tbName3.DataBindings.Add("Text", Student, "Name");  cbActive2.DataBindings.Add("Checked", Student, "Status");  }  private void pictureBox1\_DoubleClick(object sender, EventArgs e)  {  Utilities.BrowseImage(pictureBox1); //User to Open Dialog Box  }    }  }   1. **ADD EDIT USER.CS**   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using Library\_System.Data\_Model;  namespace Library\_System.Forms  {  public partial class AddEditUser : Form  {  public AddEditUser(User user = null)  {    InitializeComponent();  User = user;  if (user != null)  DisplaySelectedUser();  }  User User;  private void btnOk\_Click(object sender, EventArgs e)  {  if (ComparePassword())  {  //Dito na magdadagdag ng info  if (User == null)  {  User user = new User()  {  Id = NewId(),  Name = tbName.Text,  Username = tbUsername.Text,  Password = tbPassword.Text,  PositionId = cbPosition.SelectedIndex,//Selected Index siya dahil may choices,  Status = cbActive.Checked, //Boolean  Image = pictureBox1.Image  };  if (User.Add(user))  DialogResult = DialogResult.OK;  }  else  {  User.Image = pictureBox1.Image; //Para mag update ren yung Image, the sinet rin yung Iamge na property ni User sa laman ng PictureBox  if (User.Update(User)) // Mag uupdate  DialogResult = DialogResult.OK; // Pag nag OK mag coclose na siya  }  }  }  private void btnCancel\_Click(object sender, EventArgs e)  {  this.DialogResult = DialogResult.Cancel; //Automatic close  }  int NewId()  {    try  {  //x => \_\_\_ (Anong property ang gusto kunin - edi yung Id sa User)  return User.ListofUser.Max(x => x.Id) + 1; // +1 means everytime na magkecreate ng bago  }  catch { return 0; }  }  void DisplaySelectedUser() // Kapag Mag EEDIT, tatawagin ito ---->  {  //pictureBox1.DataBindings.Add("Image", User, "Image");  pictureBox1.Image = User.Image;  tbName.DataBindings.Add("Text", User, "Name");  tbUsername.DataBindings.Add("Text", User, "Username");  tbPassword.DataBindings.Add("Text", User, "Password");  cbPosition.DataBindings.Add("SelectedValue", User, "PositionId");  cbPosition.SelectedIndex = User.PositionId; //Ginagamit para hindi yung default ang nag aapper, yung mismong pinili talaga like if Librarian 1, dapat ayun ang lalabas  cbActive.DataBindings.Add("Checked", User, "Status");  }  bool ComparePassword() // For password  {  if (tbPassword.Text == tbRetypePassword.Text)  return true;  return false;  }  private void pictureBox1\_DoubleClick(object sender, EventArgs e)  {  Utilities.BrowseImage(pictureBox1); //User to Open Dialog Box  }  }  }   1. **LIST OF AUTHOR.CS**   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using Library\_System.Data\_Model;  using System.Windows.Forms;  using Library\_System;  namespace Library\_System.Forms  {  public partial class ListOfAuthor : Form  {  public ListOfAuthor()  {  InitializeComponent();  Reload2();  }  private void btnNew2\_Click(object sender, EventArgs e)  {  AddEditAuthor add = new AddEditAuthor();  if (add.ShowDialog() == DialogResult.OK)  Reload2(); //Function itong Reload()  }  void Reload2() //Irereset lang yung laman ng data grid view  {  dataGridView2.DataSource = null; // Gawing null para di madoble  dataGridView2.DataSource = Author.ListOfAuthor;    }  private void btnEdit2\_Click(object sender, EventArgs e)  {  Author selectedAuthor = Author.ListOfAuthor.First(y => y.Id == int.Parse(dataGridView2.SelectedCells[0].Value.ToString()));  //kung ano yung number ang pinili ayun ang ma eedit  //Ang kinukuha dito ay yung Selected Cells, whichi means yung Isang ROW  AddEditAuthor add = new AddEditAuthor(selectedAuthor); //For edit  if (add.ShowDialog() == DialogResult.OK)  Reload2();  }  private void btnDelete2\_Click(object sender, EventArgs e)  {  if (MessageBox.Show("Sigurado ka ba?", "Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)  {  Author selectedAuthor = Author.ListOfAuthor.First(y => y.Id == int.Parse(dataGridView2.SelectedCells[0].Value.ToString()));  // Yung [0] ito ay tinatawag na index which means ito yung ID  if (Author.Delete(selectedAuthor))  Reload2();  }  }  private void textBox1\_KeyDown(object sender, KeyEventArgs e)  {  if (e.KeyCode == Keys.Enter)  {  List<Author> x = Author.ListOfAuthor.Where(a => a.Name.ToString().ToUpper().Contains(textBox1.Text.ToUpper())).ToList(); // Kung anong nahanap niya since sa SEARCH ito, ayun yung ipapakita  dataGridView2.DataSource = null;  dataGridView2.DataSource = x;  }  }    }  }   1. **LIST OF STUDENT.CS**   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using Library\_System.Data\_Model;  using System.Windows.Forms;  using Library\_System;  namespace Library\_System.Forms  {  public partial class ListOfStudent : Form  {  public ListOfStudent()  {  InitializeComponent();  Reload3();  }  private void btnNew3\_Click(object sender, EventArgs e)  {  AddEditStudent add = new AddEditStudent();  if (add.ShowDialog() == DialogResult.OK)  Reload3(); //Function itong Reload()  }  void Reload3() //Irereset lang yung laman ng data grid view  {  dataGridView3.DataSource = null; // Gawing null para di madoble  dataGridView3.DataSource = Student.ListOfStudent;  }  private void btnEdit3\_Click(object sender, EventArgs e)  {  Student selectedStudent = Student.ListOfStudent.First(y => y.Id == int.Parse(dataGridView3.SelectedCells[0].Value.ToString()));  //kung ano yung number ang pinili ayun ang ma eedit  //Ang kinukuha dito ay yung Selected Cells, whichi means yung Isang ROW  AddEditStudent add = new AddEditStudent(selectedStudent); //For edit  if (add.ShowDialog() == DialogResult.OK)  Reload3();  }  private void btnDelete3\_Click(object sender, EventArgs e)  {  if (MessageBox.Show("Sigurado ka ba?", "Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)  {  Student selectedStudent = Student.ListOfStudent.First(y => y.Id == int.Parse(dataGridView3.SelectedCells[0].Value.ToString()));  // Yung [0] ito ay tinatawag na index which means ito yung ID  if (Student.Delete(selectedStudent))  Reload3();  }  }  private void textBox1\_KeyDown(object sender, KeyEventArgs e)  {  if (e.KeyCode == Keys.Enter)  {  List<Student> x = Student.ListOfStudent.Where(a => a.Name.ToString().ToUpper().Contains(textBox1.Text.ToUpper())).ToList(); // Kung anong nahanap niya since sa SEARCH ito, ayun yung ipapakita  dataGridView3.DataSource = null;  dataGridView3.DataSource = x;  }  }  }  }   1. **LIST OF USERS.CS**   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using Library\_System.Data\_Model;  using System.Windows.Forms;  namespace Library\_System.Forms  {  public partial class ListofUsers : Form  {  public ListofUsers()  {  InitializeComponent();  Reload();  //dataGridView1.DataSource = User.ListofUser; //Iassign kay DATAGRID VIEW yung list of users, then automatically icoconvert niya na ito into rows and columns  }  private void btnNew\_Click(object sender, EventArgs e)  {  AddEditUser ad = new AddEditUser();  if (ad.ShowDialog() == DialogResult.OK)  Reload(); //Funation itong Reload()  }  void Reload() //Irereset lang yung laman ng data grid view  {  dataGridView1.DataSource = null; // Gawing null para di madoble  dataGridView1.DataSource = User.ListofUser;  // dataGridView1.Columns [2].Visible = false; //Ihahide yung Column 2 sa form which is yung Image kasi hindi naman siya nakikita ng buo  }  private void btnEdit\_Click(object sender, EventArgs e)  {  User selectedUser = User.ListofUser.First(x=> x.Id == int.Parse(dataGridView1.SelectedCells[0].Value.ToString()));  //kung ano yung number ang pinili ayun ang ma eedit  //Ang kinukuha dito ay yung Selected Cells, whichi means yung Isang ROW  AddEditUser ad = new AddEditUser(selectedUser); //For edit  if (ad.ShowDialog() == DialogResult.OK)  Reload();  }  private void btnDelete\_Click(object sender, EventArgs e)  {  if(MessageBox.Show("Sigurado ka ba?", "Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)  {  User selectedUser = User.ListofUser.First(x => x.Id == int.Parse(dataGridView1.SelectedCells[0].Value.ToString()));  // Yung [0] ito ay tinatawag na index which means ito yung ID  if (User.Delete(selectedUser))  Reload();  }  }  private void textBox1\_KeyDown(object sender, KeyEventArgs e)  {  if (e.KeyCode == Keys.Enter)  {  List<User> x = User.ListofUser.Where(a => a.Name.ToString().ToUpper().Contains(textBox1.Text.ToUpper())).ToList(); // Kung anong nahanap niya since sa SEARCH ito, ayun yung ipapakita  dataGridView1.DataSource = null;  dataGridView1.DataSource = x;  }  }  }  }   1. **LOGIN.CS**   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using Library\_System.Data\_Model;  namespace Library\_System.Forms  {  public partial class Login : Form  {  public Login()  {  InitializeComponent();  }  private void btnOk\_Click(object sender, EventArgs e)  {  if (CheckUser()) // Ichecheck kung nag eexist si User, if nag eexist at tama password, mag O-OK  DialogResult = DialogResult.OK;  }  private void btnCancel\_Click(object sender, EventArgs e)  {  DialogResult = DialogResult.Cancel;  }  bool CheckUser() // Para malaman kung yung username ay nag eexist na sa database  {  try  {  if (User.ListofUser.Count > 0) //Para malaman kung nag eexist na yung Name sa ating DATA, AND ANG DATA natin ay yung ListofUser  {  User u = User.ListofUser.First(t => t.Username == tbUsername.Text);  //Hahanapin yung first element na kung saan ang Username ay equal sa linagay na Username sa TEXTBOX    if (u.Password == tbPassword.Text) //Para malaman kung same yung password  return true; // if same ang password  MessageBox.Show("Incorrect Password"); // Kapag hindi same  }  return false; // kapag wala pang laman yung User mag fafalse siya  }  catch  {  MessageBox.Show("Wala Naman"); //Kapag hindi mahanap or nag eexist, ito lalabas  return false;  }  }  }  } |
|  |
| **Images with solid fill**  **CODE OUTPUT (IMAGE)** |
|  |
|  |
| **Warning with solid fill**  **NEW ERROR ENCOUNTERED AND SOLUTION (if any)** |
| |  |  | | --- | --- | | **Error/s Encountered** | **Solution** | | 1. N/A | 1. N/A | |
|  |

|  |
| --- |
| **Head with gears with solid fill**  **WHAT I LEARNED**  *It could be new code, terms, techniques, procedures, and/or tips in programming.* |
| 1. I discovered that we can use a variety of tools in the toolbox, such as buttons, textboxes, and so on. 2. I learned that we can customize buttons and textboxes by changing their forecolor, background color, and so on. 3. I learned that we must take extra care when removing methods because they may cause an error, and to avoid this, we must remove them from the Designer.cs file. 4. I learned that using VScode studio, we can create a system and also add folders to it. |
|  |
| **Clipboard Checked with solid fill**  **SELF-EVALUATION** |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **PROGRAM INDICATORS *(90 Points)*** | | | | | | | | | | | **Programming Logic / Algorithmic Analysis** | *45* | | *33.75* | | *22.5* | | *11.25* | | ***Points*** | | The program compilers have no I/O or logic errors and meets specifications. | | The program compiles and has no I/O or logic errors and meets most specifications. | | The program compiles, some minimal I/O or logic errors and meets most specifications. | | The program produces incorrect results and/or does not compile at and fails to meet majority of specifications. | | **45** | | **Documentation** | *10 Points* | | *7.5 Points* | | *5 Points* | | *2.25 Point* | | ***10*** | | The documentation is well written and clearly explains what the code is accomplishing and how. | | The documentation is terse but gives a brief overview of what the code is accomplishing and how. | | The documentation is vague or does not give a brief overview of what the code is accomplishing and how. | | The documentation is vague and does not give a brief overview of what the code is accomplishing and how. | | | **Delivery** | *15 Points* | | *11.25 Points* | | *7.5 Points* | | *3.75 Points* | | ***15*** | | The program was delivered on time | | The program was delivered within 24 hours of the due date. | | The program was delivered within 72 hours of the due date. | | The program was delivered within 48 hours of the due date. | | | **Efficiency** | *20 Points* | | *15 Points* | | *10 Points* | | *5 Points* | | ***20*** | | The code is extremely efficient without sacrificing readability and understanding | | The code is fairly efficient without sacrificing readability and understanding. | | The code is brute force and unnecessary long. | | The code is huge and appears to be patched together | | | **RATING** | | | | | | | | | **90** | | **LEARNED INDICATORS *(10 Points)*** | | | | | | | | | | | **What I learned.** | | *10 Points* | | *7.5 Points* | | *5 Points* | | *2.5 Points* | ***Points*** | | I learned 4 or more new Code, Terms, Techniques, Procedure, and/or Tips in Programming | | I learned 3 new Code, Terms, Techniques, Procedures, and/or Tips in Programming | | I learned 2 new Code, Terms, Techniques, Procedures, and/or Tips in Programming | | I learned 1 new Code, Term, Technique, Procedures, and/or Tips in Programming | **10** | | **RATING** | | | | | | | | | **10** | | **FINAL RATING** | | | | | | | | | 100 | |
|  |