|  |
| --- |
| **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;  using Libray\_System;  using Libray\_System.Forms;  using Libray\_System.Data\_Model;  using Libray\_System.Views;  namespace Library\_System  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  User.Deserialize();  Author.Deserialize();  Student.Deserialize();  Book.Deserialize();  Book\_Borrowed.Deserialize();  Book\_Category.Deserialize();  Position.Deserialize();  var book = Book.Lists;  var aut = Author.Lists;  var cat = Book\_Category.Lists;  //Language Integrated Query  var newEntity = from b in book  join a in aut on b.AuthorId equals a.Id  join c in cat on b.CategoryId equals c.Id  select new Books  {  Id = b.Id,  //Names based in Id  Title = b.Name,  AuthorName = a.Name,  Category = c.Name,  DatePublished = b.DatePublished,  Status = b.Status    };  dataGridView1.DataSource = newEntity.ToList();  }  private void userToolStripMenuItem\_Click(object sender, EventArgs e)  {  ShowList<Users> s = new ShowList<Users>()  {  Lists = Users.View,  AddEditForm = new AddEditUser(),  DeleteMethod = User.Delete,  SearchMethod = Users.SearchView,  ObjectType = Libray\_System.Type.User,  RefreshView = Users.RefreshMethod,  isView = true  };  s.Show();  }  private void Form1\_Load(object sender, EventArgs e)// LOGIN form  {  Login login = new Login();  if (login.ShowDialog() != DialogResult.OK)  Application.Exit();  }  private void authorToolStripMenuItem\_Click(object sender, EventArgs e)  {  ShowList<Author> s = new ShowList<Author>()  {  Lists = Author.Lists,  AddEditForm = new AddEditAuthor(),  DeleteMethod = Author.Delete,  SearchMethod = Author.Search,  ObjectType = Libray\_System.Type.Author  };  s.Show();  }  private void studentsToolStripMenuItem\_Click(object sender, EventArgs e)  {  ShowList<Student> s = new ShowList<Student>()  {  Lists = Student.Lists,  AddEditForm = new AddEditStudent(),  DeleteMethod = Student.Delete,  SearchMethod = Student.Search,  ObjectType = Libray\_System.Type.Student    };  s.Show();  }  private void bOOKToolStripMenuItem\_Click(object sender, EventArgs e)  {  ShowList<Books> s = new ShowList<Books>()  {  Lists = Books.View,  AddEditForm = new AddEditBook(),  DeleteMethod = Book.Delete,  SearchMethod = Books.SearchView,  ObjectType = Libray\_System.Type.Book,  RefreshView = Books.RefreshMethod,  isView = true  };  s.Show();  }  private void bookBorrowedToolStripMenuItem\_Click(object sender, EventArgs e)  {  ShowList<BookBorrowed> s = new ShowList<BookBorrowed>()  {  Lists = BookBorrowed.View,  AddEditForm = new AddEditBookBorrowed(),  DeleteMethod = Book\_Borrowed.Delete,  SearchMethod = BookBorrowed.SearchView,  ObjectType = Libray\_System.Type.Book\_Borrowed,  RefreshView = BookBorrowed.RefreshMethod,  isView = true  };  s.Show();  }  private void bookCategoryToolStripMenuItem\_Click(object sender, EventArgs e)  {  ShowList<Book\_Category> s = new ShowList<Book\_Category>()  {  Lists = Book\_Category.Lists,  AddEditForm = new AddEditBookCategory(),  DeleteMethod = Book\_Category.Delete,  SearchMethod = Book\_Category.Search,  ObjectType = Libray\_System.Type.Book\_Category  };  s.Show();  }  private void positionToolStripMenuItem\_Click(object sender, EventArgs e)  {  ShowList<Position> s = new ShowList<Position>()  {  Lists = Position.Lists,  AddEditForm = new AddEditPosition(),  DeleteMethod = Position.Delete,  SearchMethod = Position.Search,  ObjectType = Libray\_System.Type.Position  };  s.Show();  }  }  }   * **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; }  }  }  }   * **CRUD.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 Library\_System.Data\_Model;  namespace Libray\_System.Data\_Model  {  [Serializable]  public class CRUD<c> where c : class //GENERIC - pwede mainherit ng ibang class  {  public static List<c> Lists = new List<c>();  public static string FileName { get; set; } = typeof(c).Name;// Kinukuha nito yung pangalan ng Data Model, kung sino man ang mag iinherit nitong CRUD  public static bool Add(c newData) //Means magpasa ka ng OBJECT na ang Parameter Type is User  {  Lists.Add(newData);  return Serialize();  }  public static bool Update(c newData)  {    return Serialize();  }  public static bool Delete(int id) //Delete is Method  {  var x = Lists.First(t => t.GetType().GetProperty("Id").GetValue(t).Equals(id)); // 1.Anong type ba ang ipinasa natin kay c 2. Then, kukunin yung property nung type na yun (like "Meron bang Id si user/author/student?) 3. Get the Value 4. Icocompare siya kung equal ba yung Id sa value na pinasa  Lists.Remove(x);  return Serialize();  }  public static List <c> Search (string name)  {  return Lists.Where(t => t.GetType().GetProperty("Name").GetValue(t).ToString().Contains(name)).ToList();  }  public static int NewId()  {  try  {  if(File.Exists(FileName))  {  string id = Lists.Max(x=> x.GetType().GetProperty("Id").GetValue(x)).ToString();  return int.Parse(id) + 1;  }  return 0;  }  catch (Exception) { return 0; }  }  public static bool Serialize()  {  try  {  FileStream fs = new FileStream(FileName, FileMode.Create, FileAccess.Write, FileShare.None);  BinaryFormatter formatter = new BinaryFormatter();  formatter.Serialize(fs, Lists);  fs.Dispose();  fs.Close();  return true;  }  catch { return false; }  }  public static bool Deserialize()  {  try  {  if (System.IO.File.Exists(FileName)) // Is the file exists? if exists it will deserialize and it will be assigned to the ListofUser  {  FileStream fs = new FileStream(FileName, FileMode.Open);  BinaryFormatter formatter = new BinaryFormatter();  Lists = ((List<c>)(formatter.Deserialize(fs)));  fs.Dispose();  fs.Close();  return true;  }  return false;  }  catch { return false; }  }  }  }   * **SHOW LIST.CS**   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using Library\_System.Forms;  using Library\_System.Data\_Model;  using Libray\_System.Forms;  namespace Libray\_System  {  public enum Type { User, Author, Student, Book, Book\_Borrowed, Book\_Category, Position } //List of Data, with no particular data type  internal class ShowList<c> where c : class  {  public delegate bool delDelete(int id);//Kpag tinatawag ito ay parang tinatawag na rin yung method ng ibang class  public delegate List<c> delSearch(string name);  public delegate List<c> delRefreshView();  ListForm lf = new ListForm();  public List<c> Lists { get; set; }  public Form AddEditForm { get; set; } // Used to call all add edit forms such as author and student  public delSearch SearchMethod { get; set; }  public delDelete DeleteMethod { get; set; }  public delRefreshView RefreshView { get; set; }  public bool isView { get; set; } = false;  public Type ObjectType { get; set; }  public void Show()  {  lf.btnNew.Click += BtnNew\_Click;  lf.btnDelete.Click += BtnDelete\_Click;  lf.btnEdit.Click += BtnEdit\_Click;  lf.tbSearch.KeyDown += TbSearch\_KeyDown;  Reload();  lf.ShowDialog();  }  private void TbSearch\_KeyDown(object sender, KeyEventArgs e)  {  if (e.KeyCode == Keys.Enter)  lf.dg.DataSource = SearchMethod(lf.tbSearch.Text);  }  private void BtnEdit\_Click(object sender, EventArgs e)  {  if (ObjectType == Type.Author)  EditAuthor();  else if (ObjectType == Type.User)  EditUser();  else if (ObjectType == Type.Student)  EditStudent();  else if (ObjectType == Type.Book)  EditBook();  else if (ObjectType == Type.Book\_Category)  EditBook\_Category();  else if (ObjectType == Type.Book\_Borrowed)  EditBook\_Borrowed();  else if (ObjectType == Type.Position)  EditPosition();  }  private void BtnDelete\_Click(object sender, EventArgs e)  {  if (MessageBox.Show("Sigurado ka ba?", "Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)  {  int id = int.Parse(lf.dg.SelectedCells[0].Value.ToString());  // Yung [0] ito ay tinatawag na index which means ito yung ID , so start siya lagi sa UNANG DATA  if (DeleteMethod(id))  Reload();  }  }  private void BtnNew\_Click(object sender, EventArgs e)  {  if (AddEditForm.ShowDialog() == DialogResult.OK)  Reload();  }  void Reload()  {  lf.dg.DataSource = null;  if (isView)  lf.dg.DataSource = RefreshView(); //Used to refresh data  else  lf.dg.DataSource = Lists;  }  void EditUser()  {  User selectedUser = User.Lists.First(x => x.Id == int.Parse(lf.dg.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();  }  void EditAuthor()  {  Author selectedUser = Author.Lists.First(x => x.Id == int.Parse(lf.dg.SelectedCells[0].Value.ToString()));  AddEditAuthor ad = new AddEditAuthor(selectedUser); //For edit  if (ad.ShowDialog() == DialogResult.OK)  Reload();  }  void EditStudent()  {  Student selectedUser = Student.Lists.First(x => x.Id == int.Parse(lf.dg.SelectedCells[0].Value.ToString()));  AddEditStudent ad = new AddEditStudent(selectedUser); //For edit  if (ad.ShowDialog() == DialogResult.OK)  Reload();  }  void EditBook()  {  Book selectedUser = Book.Lists.First(x => x.Id == int.Parse(lf.dg.SelectedCells[0].Value.ToString()));  AddEditBook ad = new AddEditBook(selectedUser); //For edit  if (ad.ShowDialog() == DialogResult.OK)  Reload();  }  void EditBook\_Category()  {  Book\_Category selectedUser = Book\_Category.Lists.First(x => x.Id == int.Parse(lf.dg.SelectedCells[0].Value.ToString()));  AddEditBookCategory ad = new AddEditBookCategory(selectedUser); //For edit  if (ad.ShowDialog() == DialogResult.OK)  Reload();  }  void EditBook\_Borrowed()  {  Book\_Borrowed selectedUser = Book\_Borrowed.Lists.First(x => x.Id == int.Parse(lf.dg.SelectedCells[0].Value.ToString()));  AddEditBookBorrowed ad = new AddEditBookBorrowed(selectedUser); //For edit  if (ad.ShowDialog() == DialogResult.OK)  Reload();  }  void EditPosition()  {  Position selectedUser = Position.Lists.First(x => x.Id == int.Parse(lf.dg.SelectedCells[0].Value.ToString()));  AddEditPosition ad = new AddEditPosition(selectedUser); //For edit  if (ad.ShowDialog() == DialogResult.OK)  Reload();  }  }  }   * **IADDEDIT.CS**   using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Libray\_System  {  internal interface IAddEdit  {  void DisplaySelectedObject();  void Clear();  }  }  **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;  using Libray\_System.Data\_Model;  namespace Library\_System.Data\_Model  {  [Serializable]  public class Author:CRUD<Author>  {  public int Id { get; set; } = NewId();  public string Name { get; set; }  public string About { get; set; }    }  }   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;  using Libray\_System.Data\_Model;  namespace Library\_System.Data\_Model  {  [Serializable]  public class User:CRUD<User> // User is an object  {  //PROPERTIES  public int Id { get; set; } = NewId();  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;    }  }   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 Libray\_System.Data\_Model;  namespace Library\_System.Data\_Model  {  [Serializable]  public class Student:CRUD<Student>  {  public int Id { get; set; } = NewId();  public System.Drawing.Image Image { get; set; } = null;  public string Name { get; set; }  public bool Status { get; set; } = true;    }  }   1. **BOOK BORROWED**   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 Libray\_System.Data\_Model;  namespace Library\_System.Data\_Model  {  [Serializable]  public class Book\_Borrowed:CRUD<Book\_Borrowed>  {  public int Id { get; set; } = NewId();  public int StudentId { get; set; }  public int UserId { get; set; }  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;  using System.Runtime.Serialization.Formatters.Binary;  using System.IO;  using Libray\_System.Data\_Model;  namespace Library\_System.Data\_Model  {  [Serializable]  public class Book\_Category:CRUD<Book\_Category>  {  public int Id { get; set; } = NewId();  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;  using System.Runtime.Serialization.Formatters.Binary;  using System.IO;  using Libray\_System.Data\_Model;  namespace Library\_System.Data\_Model  {  [Serializable]  public class Book:CRUD<Book>  {  public int Id { get; set; } = NewId();  public string Name { 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;  using System.Runtime.Serialization.Formatters.Binary;  using System.IO;  using Libray\_System.Data\_Model;  namespace Library\_System.Data\_Model  {  [Serializable]  public class Position:CRUD<Position> //Position of User  {  public int Id { get; set; } = NewId();// 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;  namespace Libray\_System.Forms  {  public partial class AddEditAuthor : Form, IAddEdit  {  public AddEditAuthor(Author author = null)  {  InitializeComponent();  this.author = author;  if (author != null)  DisplaySelectedObject();  }  public Author author;  public void DisplaySelectedObject() // Kapag Mag EEDIT, tatawagin ito ---->  {  tbName.DataBindings.Add("Text", author, "Name");  tbAbout.DataBindings.Add("Text", author, "About");  }  private void btnCancel\_Click(object sender, EventArgs e)  {  this.DialogResult = DialogResult.Cancel;  }  public void Clear()  {  tbName.Clear();  tbAbout.Clear();  }  private void btnOk\_Click(object sender, EventArgs e)  {  if (author == null)  {  Author a = new Author()  {    Name = tbName.Text,  About = tbAbout.Text  };  if (Author.Add(a))// Add na sa list  {  Clear(); //Para mawala yung laman sa loob ng form kapag nag add  DialogResult = DialogResult.OK;  }    }  else  {  if (Author.Update(author))  {  Clear(); //Para mawala yung laman sa loob ng form kapag nag add  DialogResult = DialogResult.OK;  }  }  }  }  }   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;  using Library\_System.Data\_Model;  namespace Libray\_System.Forms  {  public partial class AddEditStudent : Form, IAddEdit  {  public AddEditStudent(Student student = null)  {  InitializeComponent();  this.student = student;  if (student != null)  DisplaySelectedObject();  }  public Student student;  int NewId()  {  try  {  //x => \_\_\_ (Anong property ang gusto kunin - edi yung Id sa User)  return Student.Lists.Max(x => x.Id) + 1; // +1 means everytime na magkecreate ng bago  }  catch { return 0; }  }  public void DisplaySelectedObject() // Kapag Mag EEDIT, tatawagin ito ---->  {  pictureBox1.Image = student.Image;  tbName.DataBindings.Add("Text", student, "Name");  cbActive.DataBindings.Add("Checked", student, "Status");  }  public void Clear()  {  tbName.Clear();  }  private void btnOk\_Click(object sender, EventArgs e)  {  if (student == null)  {  Student a = new Student()  {  Id = NewId(),  Name = tbName.Text,  Status = cbActive.Checked,  Image = pictureBox1.Image  };  if (Student.Add(a))// Add na sa list  {  Clear(); //Para mawala yung laman sa loob ng form kapag nag add  DialogResult = DialogResult.OK;  }  }  else  {  student.Image = pictureBox1.Image;  if (Student.Update(student))  {  Clear(); //Para mawala yung laman sa loob ng form kapag nag add  DialogResult = DialogResult.OK;  }  }  }  private void btnCancel\_Click(object sender, EventArgs e)  {  this.DialogResult = DialogResult.Cancel;  }  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;  using Libray\_System;  namespace Library\_System.Forms  {  public partial class AddEditUser : Form, IAddEdit  {  public AddEditUser(User user = null)  {    InitializeComponent();  cbPosition.ValueMember = "Id"; //Yung value member ng Data Source manggagaling kay Id  cbPosition.DisplayMember = "Name"; // Nandito yung name property ng list of position  cbPosition.DataSource = Position.Lists;  User = user;  if (user != null)  DisplaySelectedObject();  }  User User;  private void btnOk\_Click(object sender, EventArgs e)  {  if (ComparePassword())  {    if (User == null)  {  User user = new User()  {    Name = tbName.Text,  Username = tbUsername.Text,  Password = tbPassword.Text,  PositionId = int.Parse(cbPosition.SelectedValue.ToString()),//Isasave na yung Value, hindi na yung Text property  Status = cbActive.Checked, //Boolean  Image = pictureBox1.Image  };  if (User.Add(user))  DialogResult = DialogResult.OK;  }  else  {  User.Image = pictureBox1.Image; //Image Update  if (User.Update(User)) // Update data  DialogResult = DialogResult.OK;  }  }  }  private void btnCancel\_Click(object sender, EventArgs e)  {  this.DialogResult = DialogResult.Cancel; //Automatic close  }    public void DisplaySelectedObject()  {  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;  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  }  public void Clear()  {  pictureBox1.Image = null;  tbName.Clear();  tbUsername.Clear();  tbPassword.Clear();  cbPosition.SelectedIndex = 0;  tbPassword.Clear();  tbRetypePassword.Clear();  cbActive.Checked = true;  }  private void AddEditUser\_FormClosing(object sender, FormClosingEventArgs e)  {  Clear();  }  }  }   1. **LIST FORM.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 ListForm : Form  {  public ListForm()  {  InitializeComponent();    }    }  }   1. **ADD EDIT BOOK.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;  using Library\_System;  using Library\_System.Data\_Model;  namespace Libray\_System.Forms  {  public partial class AddEditBook : Form, IAddEdit  {  Book book;  bool isNew = true;  public AddEditBook(Book book = null)  {  InitializeComponent();  cbAuthor.ValueMember = "Id"; //Yung value member ng Data Source manggagaling kay Id  cbAuthor.DisplayMember = "Name"; // Nandito yung name property ng list of position  cbAuthor.DataSource = Author.Lists;  cbAuthor.AutoCompleteMode = AutoCompleteMode.SuggestAppend; //Para macomplete yung laman ng Authors comboBox  cbAuthor.AutoCompleteSource = AutoCompleteSource.ListItems;  cbCategory.ValueMember = "Id"; //Yung value member ng Data Source manggagaling kay Id  cbCategory.DisplayMember = "Name"; // Nandito yung name property ng list of position  cbCategory.DataSource = Book\_Category.Lists;  cbCategory.AutoCompleteMode = AutoCompleteMode.SuggestAppend; //Para macomplete yung laman ng Authors comboBox  cbCategory.AutoCompleteSource = AutoCompleteSource.ListItems;  this.book = book;  if (book != null)  {  isNew = false;  DisplaySelectedObject();  }  }  public void Clear()  {  tbTitle.Clear();  cbAuthor.Text = string.Empty;  cbCategory.Text = string.Empty;  tbDescription.Clear();  tbDate.Value = DateTime.Now;  tbDescription.Clear();  cbActive.Checked = true;  }  public void DisplaySelectedObject() // Kapag Mag EEDIT, tatawagin ito ---->  {  tbTitle.Text = book.Name;  cbAuthor.SelectedValue = book.AuthorId;  tbDate.Value = DateTime.Now;  tbDescription.Text = book.Description;  cbCategory.SelectedValue = book.CategoryId;  cbActive.Checked = book.Status;    }  private void btnOk\_Click(object sender, EventArgs e)  {  if (isNew)  book = new Book();  book.Name = tbTitle.Text;  book.AuthorId = int.Parse(cbAuthor.SelectedValue.ToString());  book.DatePublished = tbDate.Value;  book.Description = tbDescription.Text;  book.CategoryId = int.Parse(cbCategory.SelectedValue.ToString());  book.Status = cbActive.Checked;  if (isNew)  {  if (Book.Add(book))  DialogResult = DialogResult.OK;  }  else  if (Book.Update(book))  DialogResult = DialogResult.OK;  }  private void btnCancel\_Click(object sender, EventArgs e)  {  this.DialogResult = DialogResult.Cancel; //Automatic close  }  }  }   1. **BOOK BORROWED. 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;  using Library\_System;  using Library\_System.Data\_Model;  namespace Libray\_System.Forms  {  public partial class AddEditBookBorrowed : Form, IAddEdit  {  Book\_Borrowed BookBorrowed;  bool isNew = true;  public AddEditBookBorrowed(Book\_Borrowed bookBorrowed = null)  {  InitializeComponent();  cbStudent.ValueMember = "Id";  cbStudent.DisplayMember = "Name";  cbStudent.DataSource = Student.Lists;  cbStudent.AutoCompleteMode = AutoCompleteMode.SuggestAppend;  cbStudent.AutoCompleteSource = AutoCompleteSource.ListItems;  cbUser.ValueMember = "Id";  cbUser.DisplayMember = "Name";  cbUser.DataSource = User.Lists;  cbUser.AutoCompleteMode = AutoCompleteMode.SuggestAppend;  cbUser.AutoCompleteSource = AutoCompleteSource.ListItems;  cbBook.ValueMember = "Id";  cbBook.DisplayMember = "Name";  cbBook.DataSource = Book.Lists;  cbBook.AutoCompleteMode = AutoCompleteMode.SuggestAppend;  cbBook.AutoCompleteSource = AutoCompleteSource.ListItems;  this.BookBorrowed = bookBorrowed;  if (BookBorrowed != null)  {  isNew = false;  DisplaySelectedObject();  }  }    public void DisplaySelectedObject()  {    cbStudent.SelectedValue = BookBorrowed.StudentId;  cbUser.SelectedValue = BookBorrowed.UserId;  cbBook.SelectedValue = BookBorrowed.BookId;  dtDate.Value = DateTime.Now;  dtDateReturn.Value = DateTime.Now;  tbRemarks.Text = BookBorrowed.Remarks;  }  private void btnOk\_Click(object sender, EventArgs e)  {  if (isNew)  BookBorrowed = new Book\_Borrowed();  BookBorrowed.StudentId = int.Parse(cbStudent.SelectedValue.ToString());  BookBorrowed.UserId = int.Parse(cbUser.SelectedValue.ToString());  BookBorrowed.BookId = int.Parse(cbBook.SelectedValue.ToString());  BookBorrowed.Date = dtDate.Value;  BookBorrowed.DateReturn = dtDateReturn.Value;  BookBorrowed.Remarks = tbRemarks.Text;    if (isNew)  {  if (Book\_Borrowed.Add(BookBorrowed))  DialogResult = DialogResult.OK;  }  else  if (Book\_Borrowed.Update(BookBorrowed))  DialogResult = DialogResult.OK;  }  public void Clear()  {  cbStudent.Text = string.Empty;  cbUser.Text = string.Empty;  cbBook.Text = string.Empty;  dtDate.Value = DateTime.Now;  dtDateReturn.Value = DateTime.Now;  tbRemarks.Clear();  }  private void btnCancel\_Click(object sender, EventArgs e)  {  this.DialogResult = DialogResult.Cancel; //Automatic close  }  private void AddEditBookBorrowed\_FormClosing(object sender, FormClosingEventArgs e)  {  Clear();  }  }  }   1. **BOOK CATEGORY. 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;  using Library\_System;  using Library\_System.Data\_Model;  using Libray\_System;  namespace Libray\_System.Forms  {  public partial class AddEditBookCategory : Form, IAddEdit  {  public Book\_Category category;  bool isNew = true;  public AddEditBookCategory(Book\_Category category = null)  {  InitializeComponent();  this.category = category;  if (category != null)  {  isNew = false;  DisplaySelectedObject();  }    }  public void Clear()  {  tbName.Clear();  tbDescription.Clear();  }  private void btnOk\_Click(object sender, EventArgs e)  {  if (isNew)  category = new Book\_Category();  category.Name = tbName.Text;  category.Description = tbDescription.Text;  if(isNew)  {  if(Book\_Category.Add(category))  DialogResult = DialogResult.OK;  }  else  if(Book\_Category.Update(category))  DialogResult = DialogResult.OK;  }  public void DisplaySelectedObject() // Kapag Mag EEDIT, tatawagin ito ---->  {  tbName.Text = category.Name;  tbDescription.Text = category.Description;  }  private void btnCancel\_Click(object sender, EventArgs e)  {  this.DialogResult = DialogResult.Cancel;  }  private void AddEditBookCategory\_FormClosing(object sender, FormClosingEventArgs e)  {  Clear();  }  }  }   1. **POSITION. 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;  using Library\_System;  using Library\_System.Data\_Model;  using Libray\_System;  namespace Libray\_System.Forms  {  public partial class AddEditPosition : Form, IAddEdit  {  Position position;  bool isNew = true;  public AddEditPosition(Position position = null)  {  InitializeComponent();  this.position = position;  if(position != null)  {  isNew = false;  DisplaySelectedObject();  }  }  public void Clear()  {  tbName.Clear();  tbDescription.Clear();  }  public void DisplaySelectedObject()  {  tbName.Text = position.Name;  tbDescription.Text = position.Description;  }  private void btnOk\_Click(object sender, EventArgs e)  {  if (isNew)  position = new Position();  position.Name = tbName.Text;  position.Description = tbDescription.Text;  if (isNew)  {  if (Position.Add(position))  DialogResult = DialogResult.OK;  }  else  if (Position.Update(position))  DialogResult = DialogResult.OK;  }  private void AddEditPosition\_FormClosing(object sender, FormClosingEventArgs e)  {  Clear();  }  }  }   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.Lists.Count > 0) //Para malaman kung nag eexist na yung Name sa ating DATA, AND ANG DATA natin ay yung ListofUser  {  User u = User.Lists.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)** |
| **FORM** |
|  |
| **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 | |
|  |