**TRƯỜNG ĐẠI HỌC** **SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

HCMC University of Technology and Education

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

NGÀNH CÔNG NGHỆ THÔNG TIN



***PROJECT’S REPORT :***  **Manage Student Project**

Member’s group:

1/ Nguyễn Cảnh Dũng 18110010

2/ Nguyễn Ngọc Hải Đăng 18110011

3/ Nguyễn Anh Vũ 18110068

Lecture: TS. Huỳnh Xuân Phụng

**HO CHI MINH City, June**

SCORE

|  |  |  |  |
| --- | --- | --- | --- |
| CRITERIA | CONTENT | PRESENT | TOTAL |
| SCORE |  |  |  |

ASSESSMENT OF THE TEACHERS:

……………………………………………………………………………………………………………………………...

……………………………………………………………………………………………………………………………...

……………………………………………………………………………………………………………………………...

……………………………………………………………………………………………………………………………...

……………………………………………………………………………………………………………………………...

……………………………………………………………………………………………………………………………...

……………………………………………………………………………………………………………………………...

……………………………………………………………………………………………………………………………...

……………………………………………………………………………………………………………………………...

……………………………………………………………………………………………………………………………...

……………………………………………………………………………………………………………………………...

……………………………………………………………………………………………………………………………...

*ACKNOWLEDGEMENTS*

*We would like to sincerely thank the High Quality Training Department, Ho Chi Minh City University of Technical Education for creating favorable conditions for us to study and complete this project.*

*We would like to express our deep gratitude to Huynh Xuan Phung for giving us wholehearted guidance and guidance during the implementation process.*

*We would like to thank the attention and help of friends during the implementation.*

*Although I have tried to complete the project within the scope and ability to allow but certainly inevitable shortcomings.*

*We hope to receive the sympathy, suggestions and devoted guidance of teachers and friends.*

HCMC, 23rd June 2020

Group’s members

Nguyễn Cảnh Dũng 18110010

Nguyễn Ngọc Hải Đăng 18110011

Nguyễn Anh Vũ 18110068

Table of Contents:

# Catalogue of shapes ………………………………………………

# List of tables………………………………………………………

# Chapter 1: Program Overview……………………………………

## General Introduction

## Software specification

# Chapter 2: Plan……………………………………………………

## Subdivision at work:

# Chapter 3: Software Design………………………………………

## Algorithm

## Design Class

# Chapter 4: Install and Test…………………………………………

# Chapter 5: Conclusion and Development…………………………

## Reference Material**Chapter 1 : Program Overview**

1. *Genneral Introduction*

Request: Manage student project

Manage: Information of Student (name, id, birthday,….)

+ Information of project (name, day begin, day end,…)

Analysis: Creat database to save information and use control datagridview on winform to show data of studne, project and access project score,…

*Software Specfication*

* Program allow us to enter the information of student and project and give them a score. In addition, program also has some features: add, edit, delete,…
* **Input:** Student (name, ID, birthday,…), Project ( name, day begin, day end,…)
* **Output :**
* Requested by the proposed project.
* Adopt object Oriented programming and new software technologies.
* Lightweight software, stable running.
* Tool and Technology used:

Entity framework

Visual studio 2019 (.NET framework)

Microsoft SQL server management studio 18

Chapter 2: Plan

1. *Subdivision at work*

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Fullname | Desciption work | Contribution (%) |
| 1 | Nguyễn Cảnh Dũng | Design Program  Annalysis Interface.  Write code.  Design and connect to database | 40 |
| 2 | Nguyễn Ngọc Hải Đăng | Design Program  Annalysis Interface.  Desgin Graphic | 30 |
| 3 | Nguyễn Anh Vũ | Design Program  Annalysis Interface  Write code  Report | 30 |

Chapter 3: Software Design

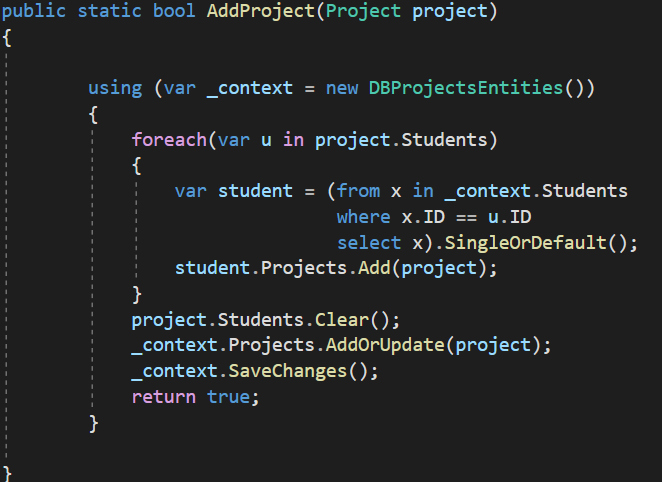
**How to do:**

Need to solve 1 important problem: How to add and access project score for student

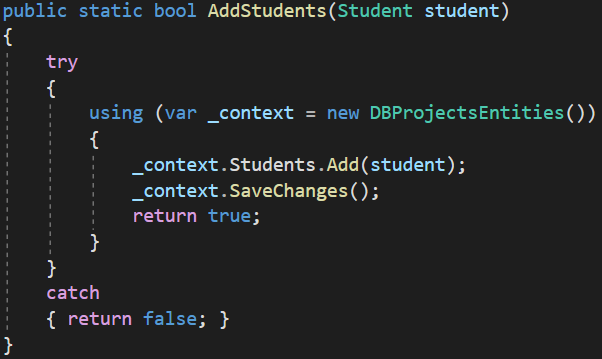
Solution: use SQL commands

**Alogrithm:**

### Save Information from Form to database



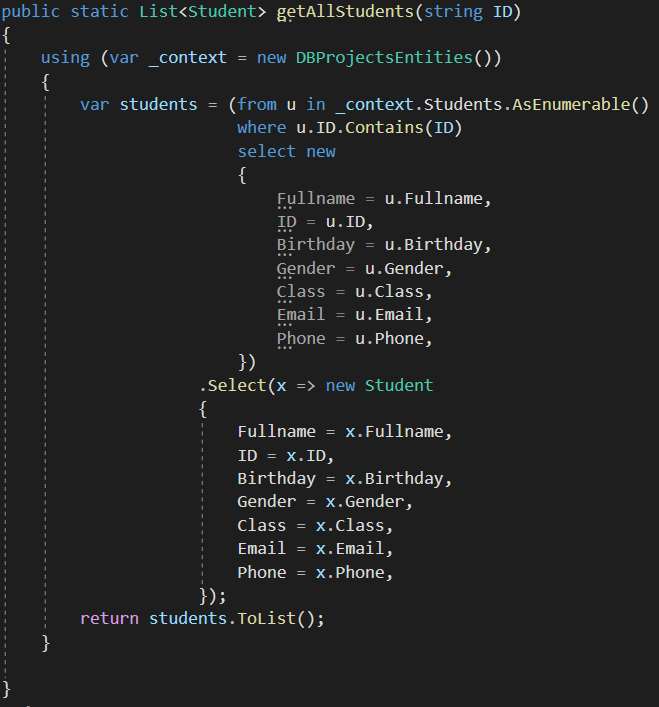
Add Project

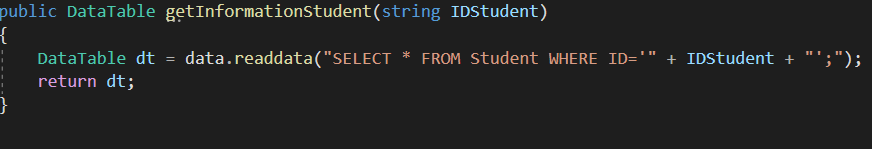


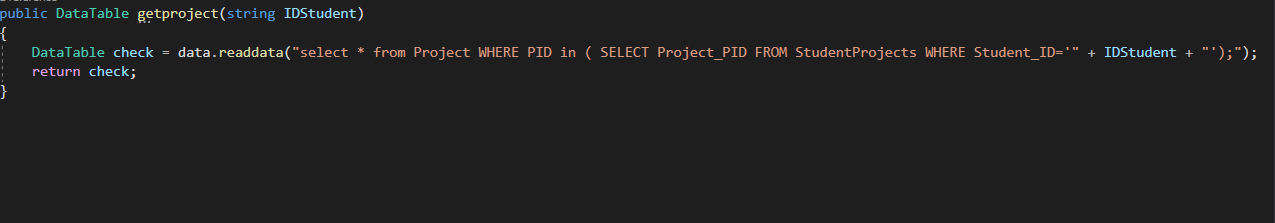
Add Student

### Show data on datagridview





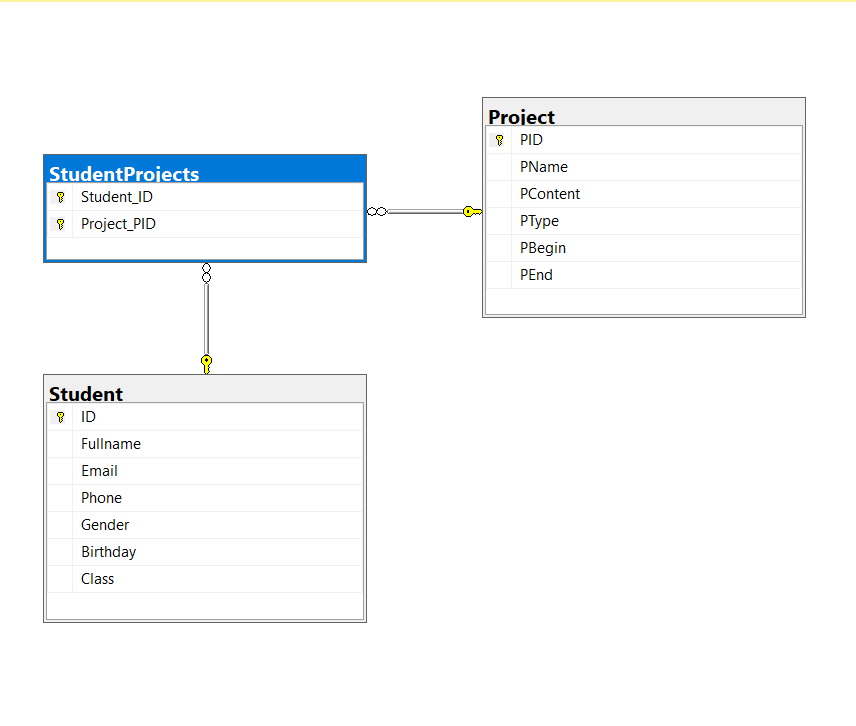




**Design Class:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Code** | **Function** |
|  | **StudentForm** |  | **To Add and Delete** |
| 1 | btnAdd\_Click | if (this.txtFullname.Text.Trim().Length <= 0)  {  erFullname.SetError(this.txtFullname, "Please enter fullname");  }  if (this.txtID.Text.Trim().Length <= 0)  {  erID.SetError(this.txtID, "Please enter ID");  }  else  {  if (StudentController.checkExistStudentName(this.txtID.Text.Trim()) == true)  {  erID.SetError(this.txtID, "This ID is available in database");  return;  }  }  Student nStudent = new Student();  nStudent.Fullname = this.txtFullname.Text.Trim();  nStudent.ID = this.txtID.Text.Trim();  nStudent.Birthday = this.dtpBirthday.Value;  nStudent.Gender = this.txtGender.Text.Trim();  nStudent.Class = this.txtClass.Text.Trim();  nStudent.Email = this.txtEmail.Text.Trim();  nStudent.Phone = this.txtPhone.Text.Trim();  if (StudentController.AddStudents(nStudent) == false)  {  MessageBox.Show("Error in adding a new user", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  BindingSource source = new BindingSource();  source.DataSource = StudentController.getAllStudents(); ;  this.dataStudent.DataSource = source; | When user click Add, the information you enter to the form will be added to the database |
| 2 | btnDelete\_Click | if (this.dataStudent.SelectedRows.Count <= 0)  {  return;  }  string ID = this.dataStudent.SelectedRows[0].Cells[0].Value.ToString();  if (StudentController.DeleteStudent(ID) == false)  {  MessageBox.Show("Cannot delete student");  }  else  {  BindingSource source = new BindingSource(); // add an empty row.  source.DataSource = StudentController.getAllStudents();  this.dataStudent.DataSource = source;  } | This button will clear all the information of that student |
|  | **ProjectForm** |  | **To Add, Delete, Update and Display for Project** |
|  | DisplayToListView | List<Project> lstProject = ProjectController.getAllProject();  string displayIDStudents;  this.listViewProject.Items.Clear();  foreach (Project project in lstProject)  {  displayIDStudents = "";  foreach (Student s in project.Students)  {  displayIDStudents = displayIDStudents + s + " ";  }    ListViewItem eVent = new ListViewItem(project.PID.ToString());  eVent.SubItems.Add(new ListViewItem.ListViewSubItem(eVent, project.PName));  eVent.SubItems.Add(new ListViewItem.ListViewSubItem(eVent, project.PContent));  eVent.SubItems.Add(new ListViewItem.ListViewSubItem(eVent, project.PType));  eVent.SubItems.Add(new ListViewItem.ListViewSubItem(eVent, project.PBegin.HasValue ? project.PBegin.GetValueOrDefault().ToString("dd/MM/yyyy"): "NULL"));  eVent.SubItems.Add(new ListViewItem.ListViewSubItem(eVent, project.PEnd.HasValue ? project.PEnd.GetValueOrDefault().ToString("dd/MM/yyyy") : "NULL"));  eVent.SubItems.Add(new ListViewItem.ListViewSubItem(eVent, displayIDStudents));  this.listViewProject.Items.Add(eVent); | This will show the information of project |
| 4 | btnAddP\_Click | Project NProject = new Project();  NProject.PID = ProjectController.getIDfromDB();  NProject.PName = this.txtPName.Text;  NProject.PContent = this.txtContent.Text;  NProject.PType = this.ccbType.Text.Trim();  NProject.PBegin = this.dateBegin.Value; //dd/MM/YYYY  NProject.PEnd = this.dateEnd.Value; //dd/MM/YYYY  NProject.Students = new List<Student>();  string displayIDStudents = "";  for (int i = 0; i < lstIDs.Items.Count; i++)  {  //displayIDStudents = displayIDStudents + this.lstIDs.Items[i].ToString() + " ";  NProject.Students.Add(this.lstIDs.Items[i] as Student); //ID  }  if (checkbegin(NProject) && checkend(NProject) && checkname(NProject))  {  MessageBox.Show("This Project has already exist");  return;  }  if (NProject.PEnd <= NProject.PBegin)  {  MessageBox.Show("Project's deadline must be greater than its begin");  return;  }  if (ProjectController.AddProject(NProject) == false)  {  MessageBox.Show("Error in adding a new project", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  this.DisplayToListView(); | This button will add the project. |
| 5 | btnDeleteP\_Click | if (this.listViewProject.SelectedItems.Count <= 0)  {  return;  }  int idproject = int.Parse(this.listViewProject.SelectedItems[0].SubItems[0].Text);  if (ProjectController.DeleteProject(idproject) == false)  {  MessageBox.Show("Cannot delete user!!!");  }  else  {  MessageBox.Show("Delete success", "Note", MessageBoxButtons.OK);  this.DisplayToListView();  } | This button will delete the selected project |
| 6 | btnUpdate\_Click | this.listViewProject.SelectedItems[0].SubItems[1].Text = this.txtPName.Text;  this.listViewProject.SelectedItems[0].SubItems[2].Text = this.txtContent.Text;  this.listViewProject.SelectedItems[0].SubItems[3].Text = this.ccbType.Text;  this.listViewProject.SelectedItems[0].SubItems[4].Text = this.dateBegin.Value.ToString("dd/MM/yyyy");  this.listViewProject.SelectedItems[0].SubItems[5].Text = this.dateEnd.Value.ToString("dd/MM/yyyy");  this.listViewProject.SelectedItems[0].SubItems[6].Text = this.lstIDs.Text;  Project project = new Project();  project.PID = int.Parse(this.listViewProject.SelectedItems[0].SubItems[0].Text.Trim());  if (project.PID <= 0)  return;  if (this.listViewProject.SelectedItems[0].SubItems[1].Text is null)  {  this.listViewProject.SelectedItems[0].SubItems[1].Text = "";  }  project.PName = this.txtPName.Text;  if (this.listViewProject.SelectedItems[0].SubItems[2].Text is null)  {  this.listViewProject.SelectedItems[0].SubItems[2].Text = "";  }  project.PContent = this.txtContent.Text;  if (this.listViewProject.SelectedItems[0].SubItems[3].Text is null)  {  this.listViewProject.SelectedItems[0].SubItems[3].Text = "";  }  project.PType = this.ccbType.Text.Trim();  if (this.listViewProject.SelectedItems[0].SubItems[4].Text is null)  {  this.listViewProject.SelectedItems[0].SubItems[4].Text = DateTime.Now.ToString();  } project.PBegin = this.dateBegin.Value;  if (this.listViewProject.SelectedItems[0].SubItems[5].Text is null)  {  this.listViewProject.SelectedItems[0].SubItems[5].Text = DateTime.Now.ToString();  } project.PEnd = this.dateEnd.Value;  if (this.listViewProject.SelectedItems[0].SubItems[6].Text is null)  {  this.listViewProject.SelectedItems[0].SubItems[6].Text = " ";  }  else // Can not update ID  {  //bool IsListIDClicked = IsEventFired(lstIDs\_DoubleClick);  for (int i = 0; i < lstIDs.Items.Count; i++)  {  project.Students.Add(this.lstIDs.Items[i] as Student); //ID  }  //this.lstIDs.DoubleClick += delegate { Is = true; };  }  if (ProjectController.UpdateProject(project) == false)  {  MessageBox.Show("Cannot update!!!");  }  else  {  MessageBox.Show("Update success", "Note", MessageBoxButtons.OK);  } | This button update the information needed |
|  | **SearchForm** |  | **Search and access project of student** |
| 7 | btnSearch\_Click | BindingSource bs = new BindingSource();  DataTable dt= studentProjectController.getproject(this.txtID.Text);  bs.DataSource=dt;  this.dataSearchID.DataSource = bs;  dt = studentController.getInformationStudent(this.txtID.Text);  foreach (DataRow row in dt.Rows)  {  this.label5.Text = "Name: " + row[1].ToString();  this.label6.Text = "Email: " + row[2].ToString();  this.label7.Text = "Phone: " + row[3].ToString();  this.label8.Text = "Gender:" + row[4].ToString();  this.label9.Text = "BirthDay: " + DateTime.ParseExact(row[5].ToString(), "dd-MMM-yy hh:mm:ss tt", CultureInfo.InvariantCulture).ToString("dd-MM-yyyy"); //"dd-MM-yyyy"  this.label10.Text = "Class: " + row[6].ToString();  } | This button allow to search student by ID |
| 8 | btnAssess\_Click | Project project = new Project();  project.Score = this.txtScore.Text;  project.Comment = this.txtComment.Text;  //if (project.AddStudents(nStudent) == false)  //{  // MessageBox.Show("Error in adding a new user", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  //}  BindingSource source = new BindingSource();  //source.DataSource = ProjectController.getExistProject();  this.dataSearchID.DataSource = source; | This button allow to access score |

**Design Database:**

****

|  |  |  |
| --- | --- | --- |
| STT | Name of Table | Purpose |
| 1 | Student | Save information of students |
| 2 | Project | Save projects |

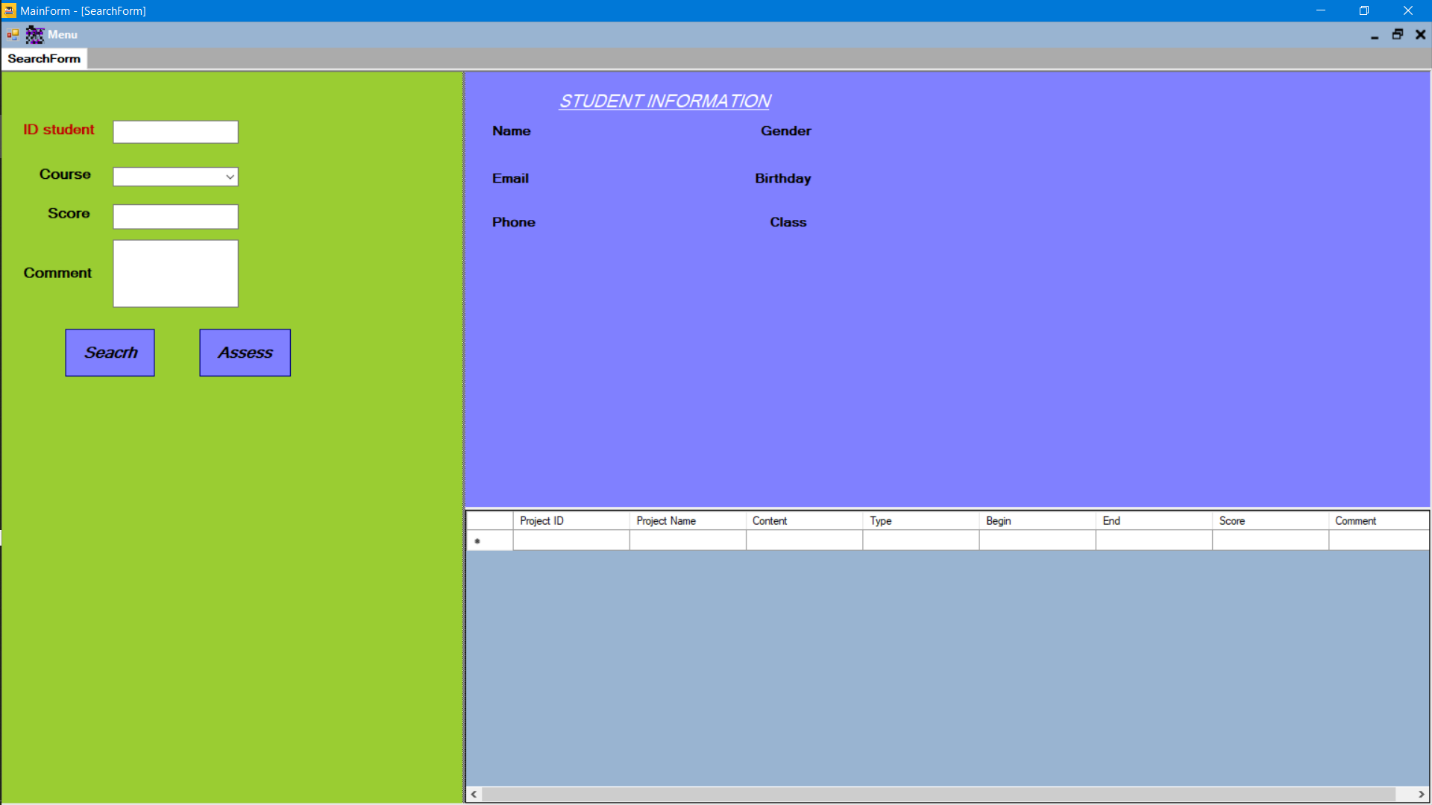
Student table:

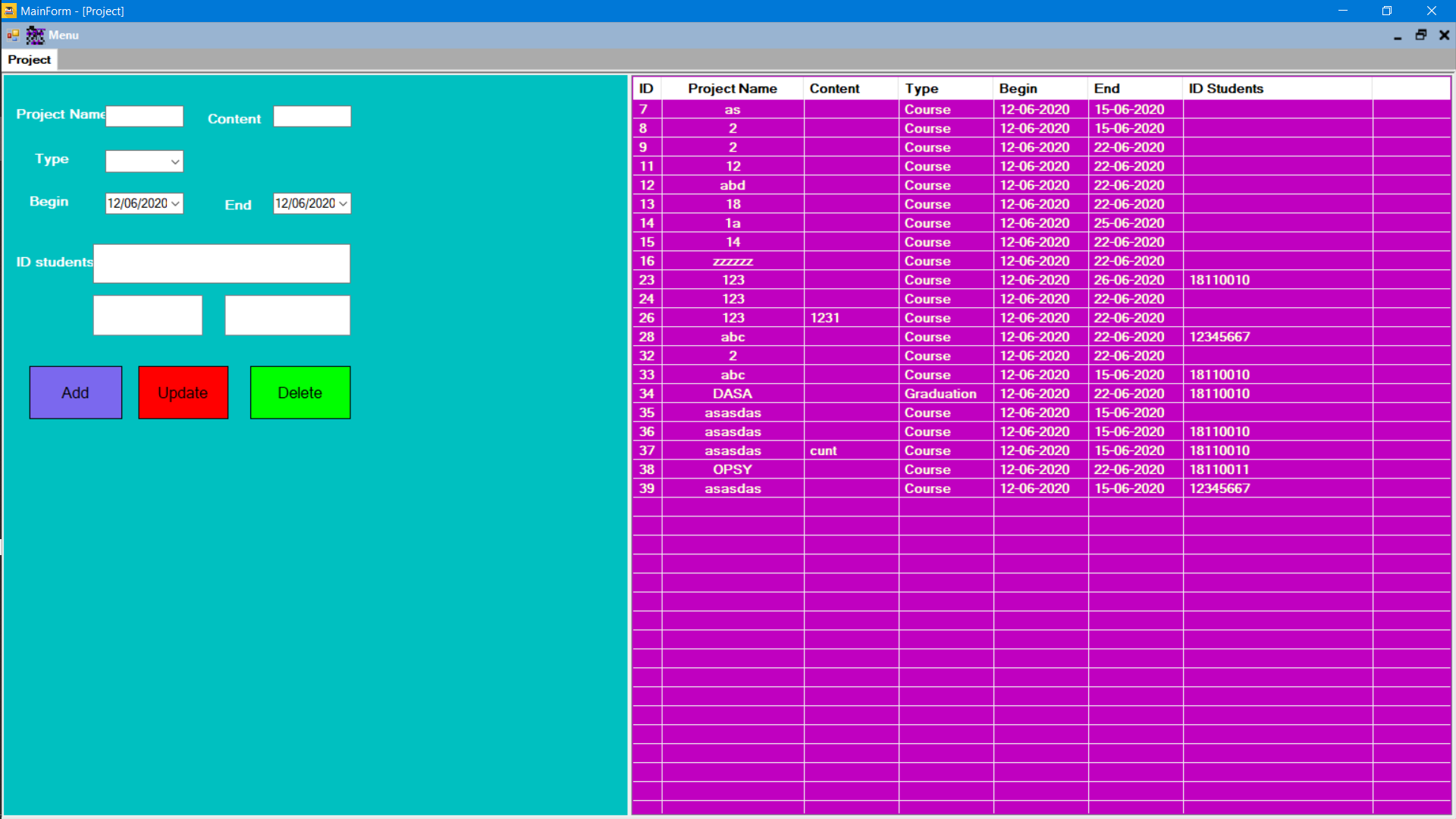
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Field | Type | Purpose |
| 1 | Fullname | Nvarchar(MAX) | Save student’s fullname |
| **2** | **ID** | **Nvarchar(128)** | **Save student’s id** |
| 3 | Birthday | datetime | Save student’s birthday |
| 4 | Gender | Nvarchar(MAX) | Save student’s gender |
| 5 | Class | Nvarchar(MAX) | Save student’s class |
| 6 | Email | Nvarchar(MAX) | Save student’s email |
| 7 | Phone | Nvarchar(MAX) | Save student’s phone |

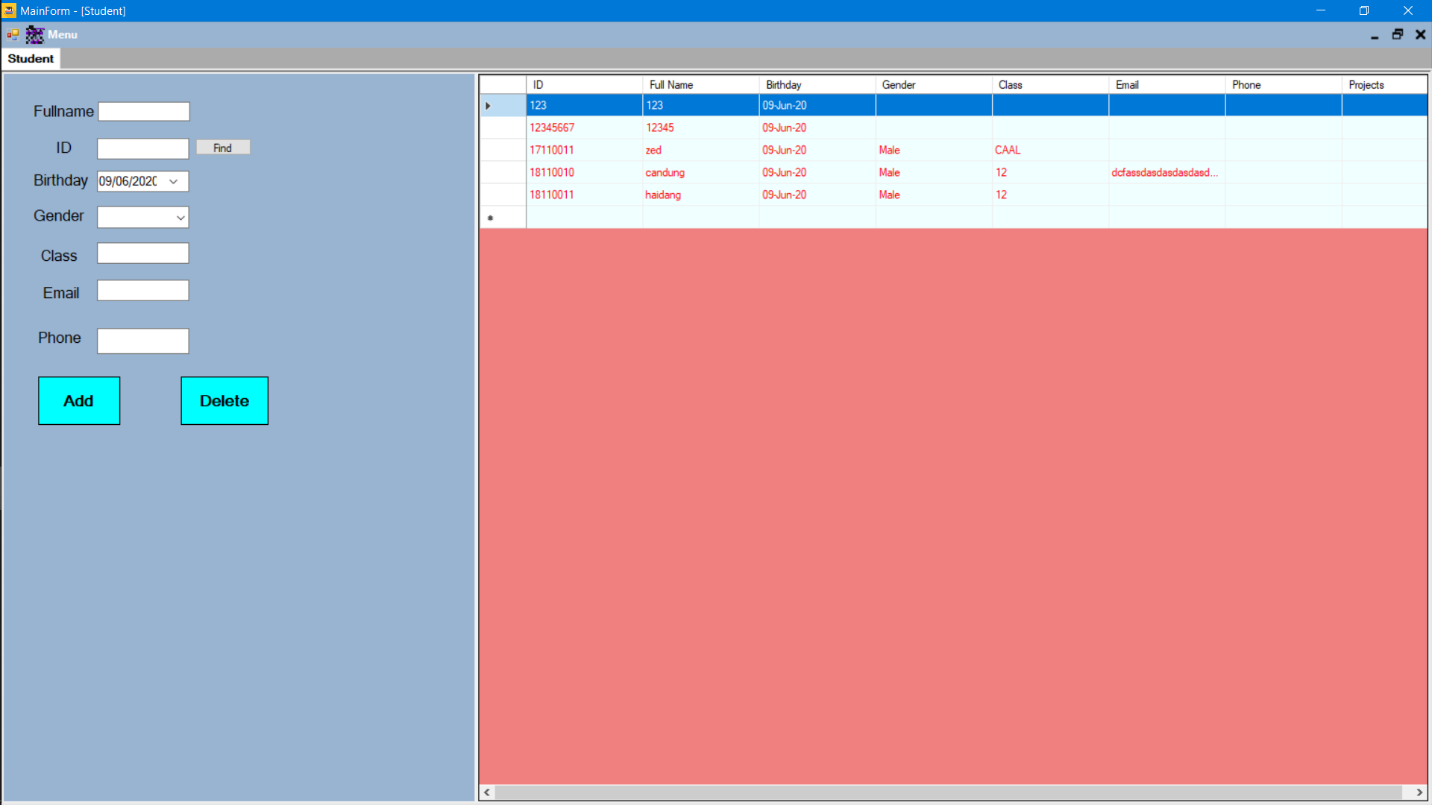
Project table:

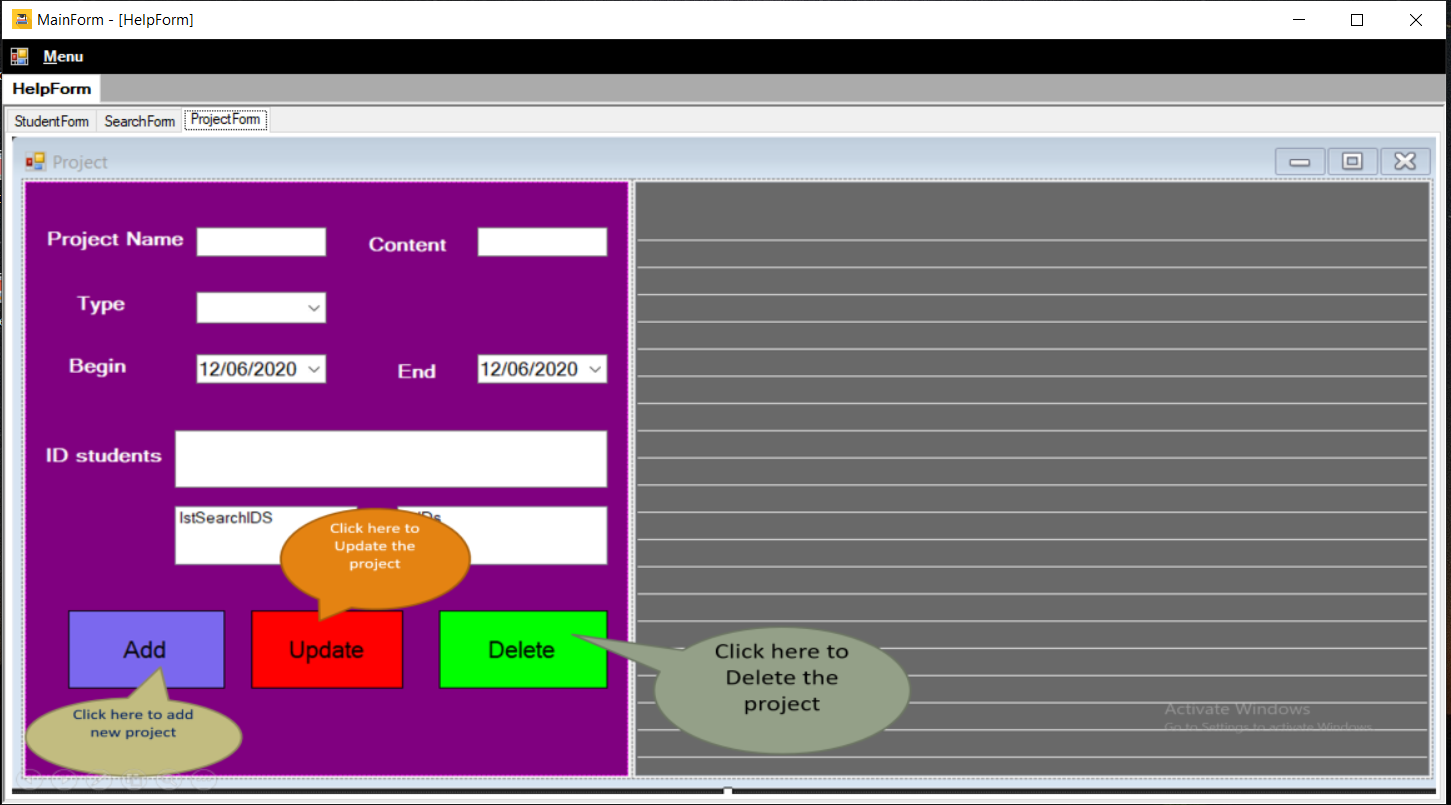
|  |  |  |  |
| --- | --- | --- | --- |
| TT | Field | 28 | Purpose |
| 1 | PName | Nvarchar(MAX) | Save project’s name |
| **2** | **PID** | int | **Save project’s ID** |
| 3 | PContent | Nvarchar(MAX) | Save project’s content |
| 4 | PType | Nvarchar(100) | Save project’s type |
| 5 | PBegin | datetime | Save project’s begin day |
| 6 | PEnd | datetime | Save project’s end day |

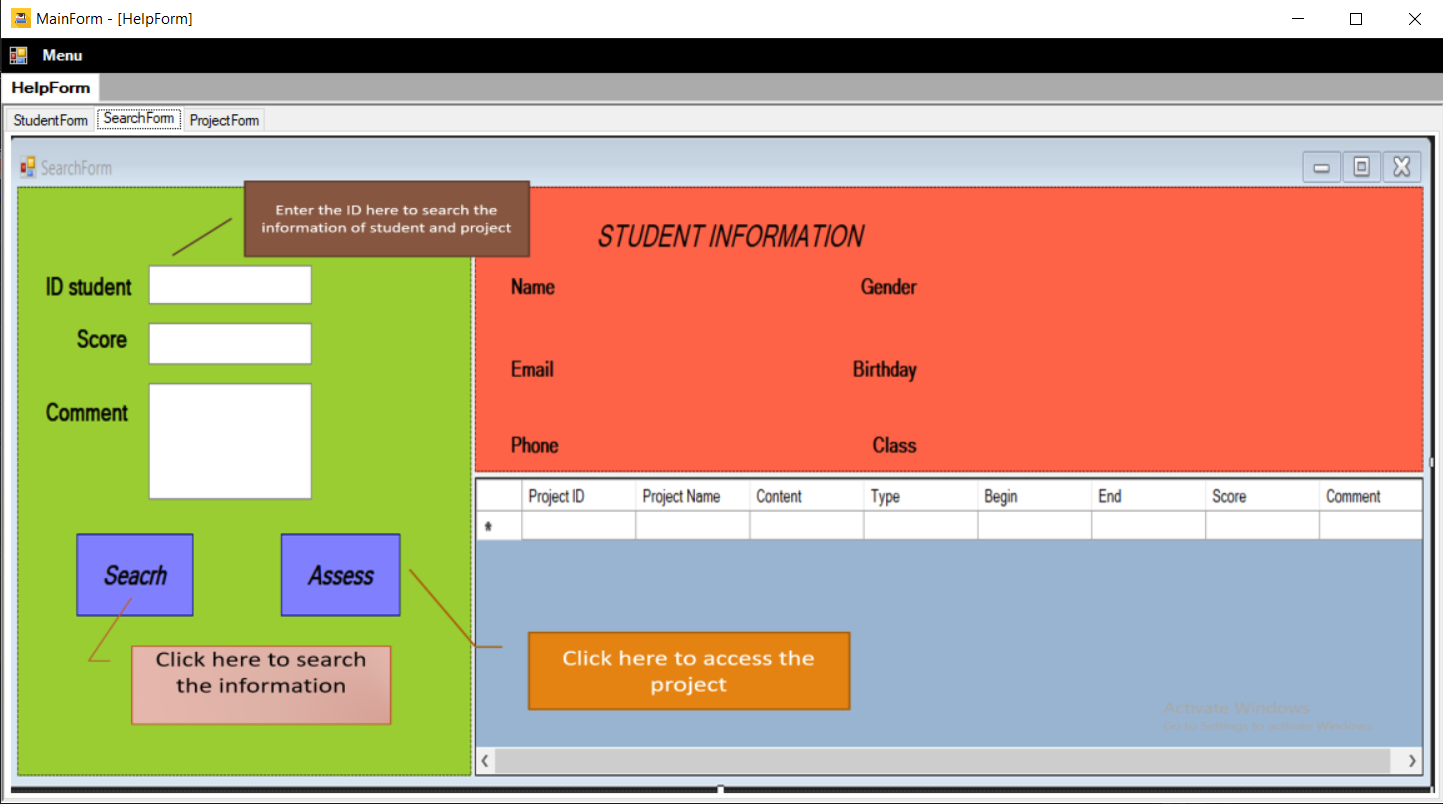
Chapter 4 : Install and Test

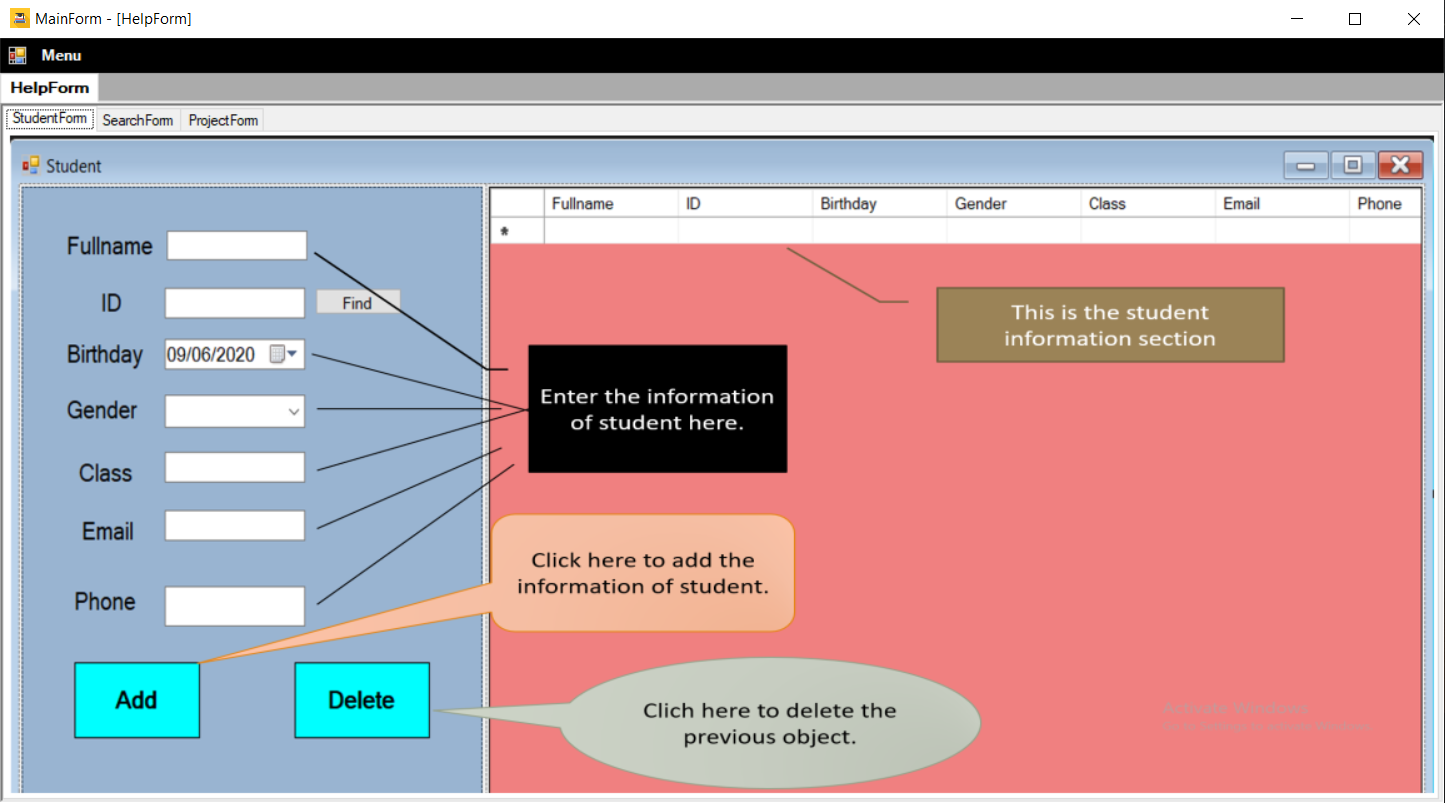












Chapter 5 : Development

* Add teacher login and only that teacher can access the project that they guide
* Teacher can find all information about students, projects. Then teacher can grade and assess.
* This program helps teacher easier in manage the students and project.

***References:***

[1]Entity Framework (Microsoft) <https://docs.microsoft.com/en-us/ef/>

[2] C#: Learn C# in One Day and Learn It Well. C# for Beginners with Hands-on Project. (Learn Coding Fast with Hands-On Project Book 3) Kindle Edition by Jamin Chan <https://www.amazon.com/Beginners-Hands-Project-Coding-Project-ebook/dp/B016Z18MLG>

[3]C# 8.0 and NET.Core 3.0 – Modern Cross Platform Development Fourth Edition(2019).pdf

[4] Documentation windown forms’ on Website Microsoft <https://docs.microsoft.com/en-us/dotnet/framework/winforms/>

[5] Some videos from : <https://www.youtube.com/channel/UCWwPuel7ja3fi1TyXWpHzXg/videos>