



VISUAL PROGRAMMING

COSC 31112



FINAL REPORT

PROJECT MANAGEMENT SYSTEM
BUILDING & CONSTRUCTION



GROUP NO : 01



UNIVERSITY OF KELANIYA
DEPARTMENT OF STATISTICS & COMPUTER SCIENCE
ACADEMIC YEAR 2021/2022

COSC 31112 – VISUAL PROGRAMMING
FINAL PROJECT REPORT

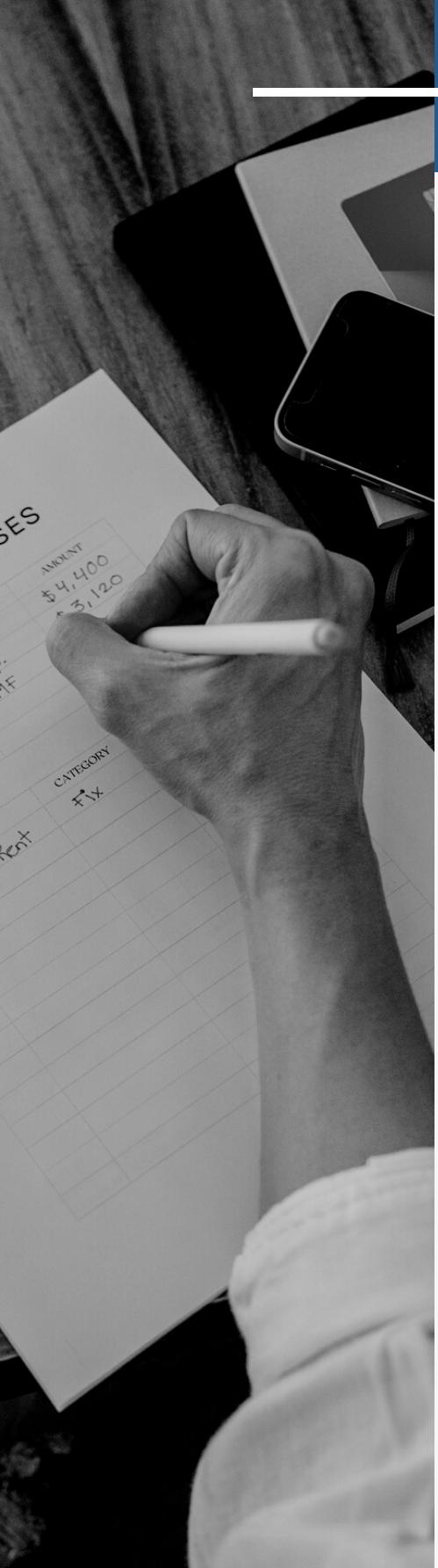
PROJECT MANAGEMENT SYSTEM
GROUP 01



Contents

Topics	Page No
[01]. Introduction	02
[02]. Problem Identification	03
[03]. Our Goals & Objectives	04
[04]. Functional & Non functional requirements	07
[05]. Diagram explanation	12
[06]. Design & code implementation of each form in the application	19
[07]. Tools and things used for design	68
[08]. Initial plan vs. Actual plan	69
[09]. Work contribution & Chalengers	70
[10]. Future enhancements	73
[11]. Our team	76

Introduction



A project management system is a digital tool designed to help individuals and teams efficiently plan, execute, and track the progress of projects. It provides a centralized platform where users can define project goals, allocate tasks, set deadlines, assign responsibilities, and monitor the overall project timeline. This system facilitates collaboration by allowing team members to communicate, share files, and update their progress in real-time. Additionally, it offers features such as milestone tracking, resource management, and reporting to ensure projects are completed on time and within scope.

In this introduction, we provide an overview of such a system, its key components, and how the .NET Framework can be leveraged to build a visual project management tool.

.NET Framework and C# language have been used to create a project management system for building and construction. A user-friendly visual interface is used to make the project management system efficient. We use Visual Studio software for that.

The project consists of three main interfaces, each tailored to the unique needs of customers, employees, and management. These interfaces act as portals to the system, giving users access to specific functionalities that align with their roles and responsibilities. In addition, nine additional models are accessible through these primary interfaces, providing additional functionality and enhanced data management capabilities as needed.

01



Problem identification

- **Complexity and Scale**

Construction projects involve numerous stakeholders, intricate processes, and extensive data.

Managing all these aspects manually can lead to errors, delays, and cost overruns.

- **Communication Challenges**

Effective communication between contractors, subcontractors, architects, engineers, and clients is critical.

Inefficient communication can lead to misunderstandings and project delays.

- **Resource Allocation**

Efficiently allocating and managing resources, including labor, materials, and equipment, is vital.

Poor resource management can lead to inefficiencies and increased costs.

We will give a brief introduction to the identified problems.

02



Our Goals and Objectives

03

- **Efficient Project Planning**

Enable project managers to efficiently plan construction projects, including defining tasks, allocating resources, setting timelines, and establishing dependencies.

- **User-friendly interface design**

Create intuitive and user-friendly interfaces for all modules to ensure ease of use for customers, employees, and management.

Prioritize clear navigation, consistent layouts, and easy access to relevant information on each interface.

Conduct user testing to gather feedback and improve the interface.

Our Goals and Objectives

03

- **Authentication and Access Control**

Implement secure login and authentication mechanisms for customers, employees, and management to ensure that only authorized users can access the system.

Define and enforce access controls based on user roles to restrict access to specific interfaces and functionalities.

Include password encryption and protection against common security vulnerabilities.

- **Security and Data Protection**

Guarantee the security and confidentiality of sensitive project data and documents.

- **Scalability and Flexibility**

Design the system to be scalable, accommodating projects of various sizes and complexities.

Our Goals and Objectives

03

- **Document and Data Management**

Streamline document management, including contracts, blueprints, permits, and other project-related files.

- **Data Analytics and Reporting**

Provide insights into project performance through data analytics and reporting tools.

- **Resource Optimization**

Develop optimization algorithms to make efficient use of resources, minimizing waste and costs.

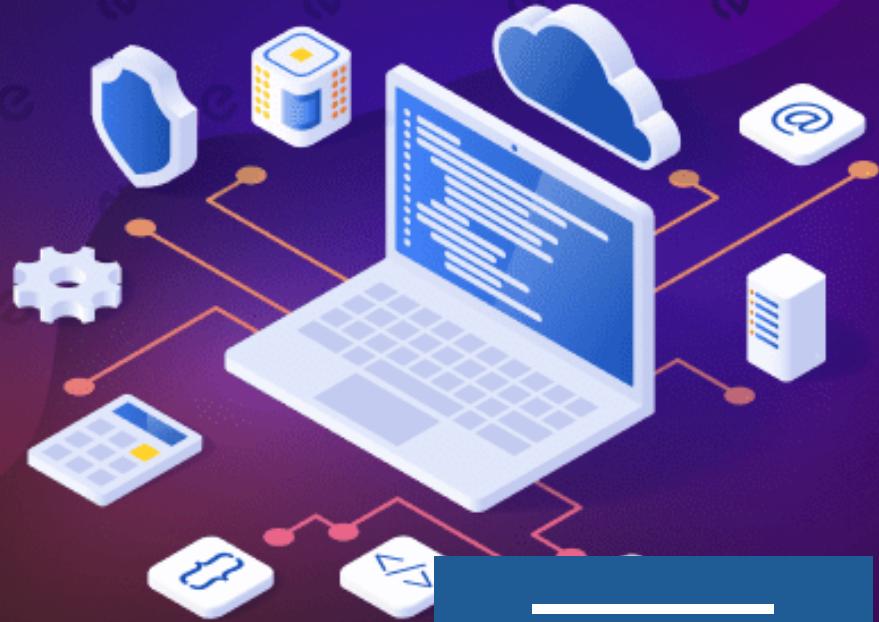
- **Cost Estimation**

Build applications for project managers to track progress, allocate resources, and manage budgets effectively.

- **Real-Time Communication**

Enhance communication between project stakeholders, ensuring that contractors, subcontractors, architects, and clients can share information and updates in real time.

Functional vs Non-functional Requirements



Functional Requirement

- **Project Planning and Scheduling**

Allow users to create and manage project plans.

Support task allocation, dependencies, and timelines.

Enable the creation of Gantt charts for visual project scheduling.

- **Resource Management**

Provide tools for resource allocation, including labor, equipment, and materials.

Allow users to assign resources to specific tasks.

Generate resource utilization reports.

Functional requirements are explicit descriptions of what a software system or product must do to meet user needs and perform its intended tasks.

04

Functional Requirements

• Reporting and Analytics

Generate customizable project performance reports. Provide data visualization tools, including charts and graphs. Offer dashboards for real-time project monitoring.

04

• Document and Data Management

Create a centralized repository for project-related documents. Implement version control for documents. Allow users to set access permissions and manage document revisions.

• Risk Management

Allow users to identify, assess, and manage project risks. Provide risk mitigation strategies and risk response planning tools. Generate risk assessment reports and risk dashboards.

• Budget Control and Cost Tracking

Allow users to set project budgets and track expenses. Generate cost reports and alerts for budget overruns. Enable users to manage change orders and their impact on costs.

• Communication and Collaboration

Offer real-time messaging and discussion forums for project teams. Facilitate document sharing and collaboration on blueprints, plans, and project-related files. Provide comment and annotation features for document collaboration.

Functional Requirements

04

- **Reporting and Analytics**

Generate customizable project performance reports.
Provide data visualization tools, including charts and graphs.
Offer dashboards for real-time project monitoring.

- **Document and Data Management**

Create a centralized repository for project-related documents.
Implement version control for documents.
Allow users to set access permissions and manage document revisions.

- **Risk Management**

Allow users to identify, assess, and manage project risks.
Provide risk mitigation strategies and risk response planning tools.
Generate risk assessment reports and risk dashboards.

- **Budget Control and Cost Tracking**

Allow users to set project budgets and track expenses.
Generate cost reports and alerts for budget overruns.
Enable users to manage change orders and their impact on costs.

- **Communication and Collaboration**

Offer real-time messaging and discussion forums for project teams.
Facilitate document sharing and collaboration on blueprints, plans, and project-related files.
Provide comment and annotation features for document collaboration.

Non Functional Requirement

• Security and Privacy

Data Security: Ensure strong encryption mechanisms to protect sensitive customer and financial data in transit and at rest.

Authentication Security: Implement strict security measures for facial recognition to prevent unauthorized access.

Compliance: Adhere to data protection regulations and industry standards to protect customer privacy.

• Performance:

Response time: Ensure that the system responds quickly to user interactions, including facial recognition, data retrieval, and transaction processing.

• Reliability:

Availability: ensuring high system availability, minimizing downtime and disruption to management operations.

Fault Tolerance: Implement mechanisms to properly recover from system failures or errors.

Non-functional requirements are specifications that define how a software system or product should perform, including attributes like speed, reliability, security, and usability, rather than specific functionalities.

04

Non Functional Requirement

- Usage

User-friendly interfaces: Design intuitive and user-friendly interfaces for all types of users, including clear navigation and helpful error messages.

Accessibility: Ensure that the system is accessible to users with disabilities, in accordance with applicable accessibility standards.

- Maintainability

Code maintainability: Write well-documented, modular, and maintainable code to facilitate future updates and enhancements.

Database maintenance: Develop routines for regular database maintenance, including backups and optimization.

- Compatibility:

Platform compatibility: Ensure that the application works seamlessly on different operating systems and browsers, if applicable.

Database Compatibility: Ensure compatibility with the selected database management system.

- User-friendly interface

Create intuitive and user-friendly interfaces for all modules to ensure ease of use for customers, employees, and management.

Prioritize clear navigation, consistent layouts, and easy access to relevant information on each interface.

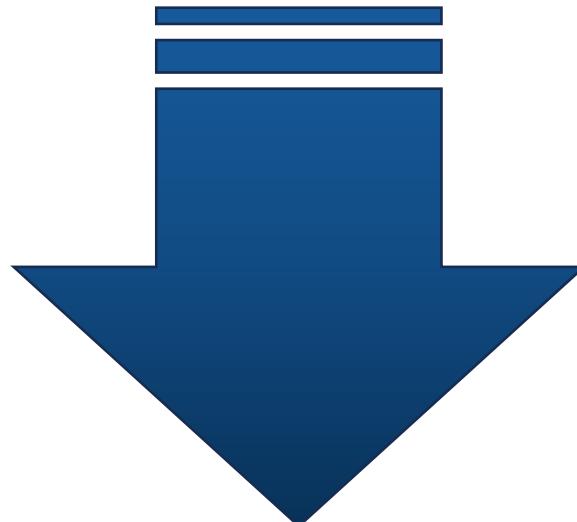
Conduct user testing to gather feedback and improve the interface.

04

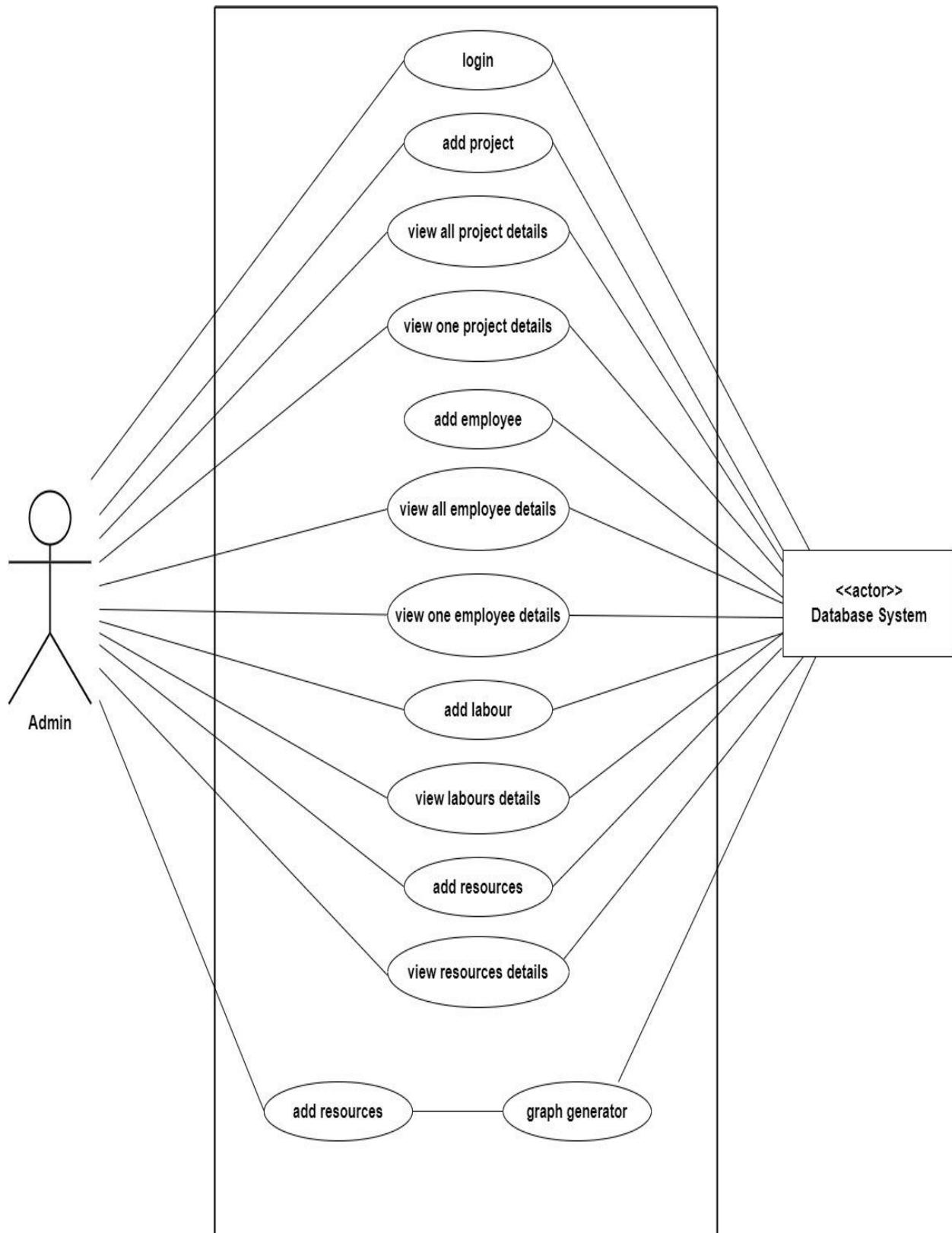


Diagram explanation

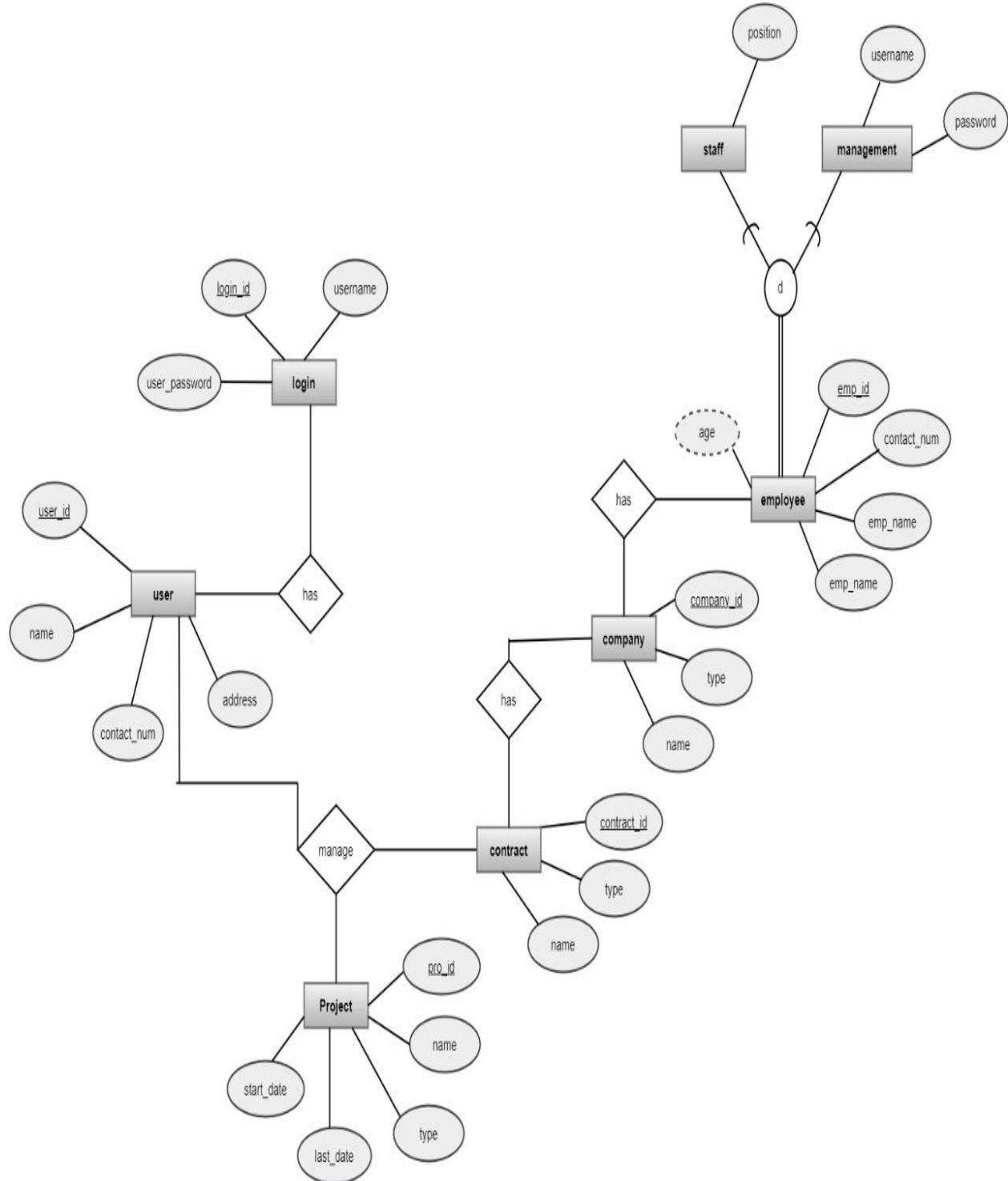
- User case diagram
- ER diagram
- Sequence diagram
- Class diagram



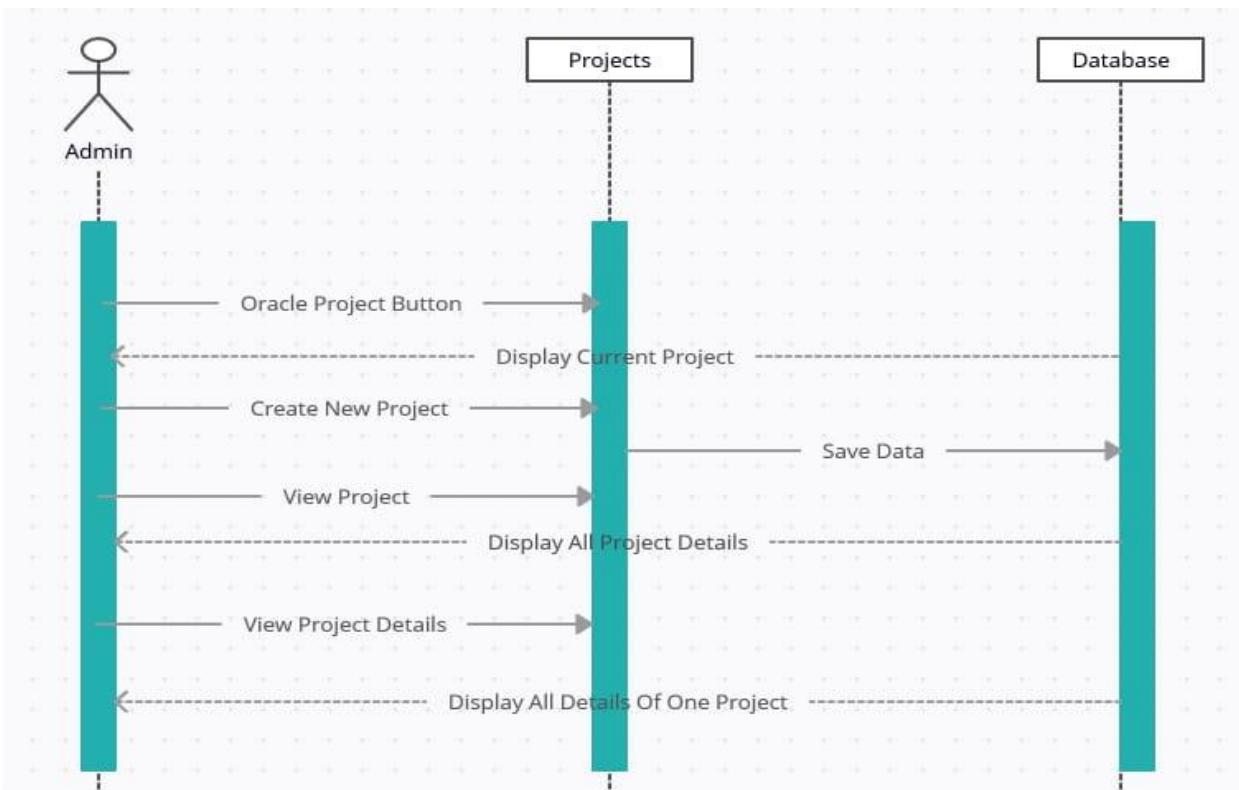
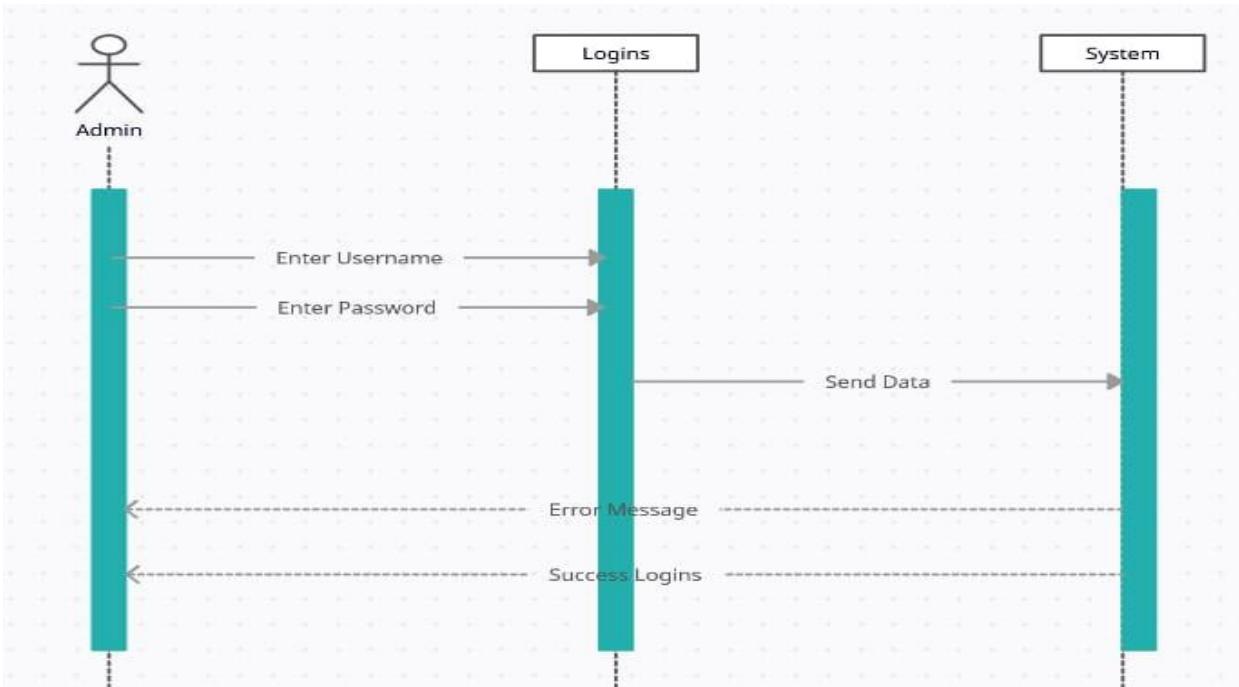
Use case diagram

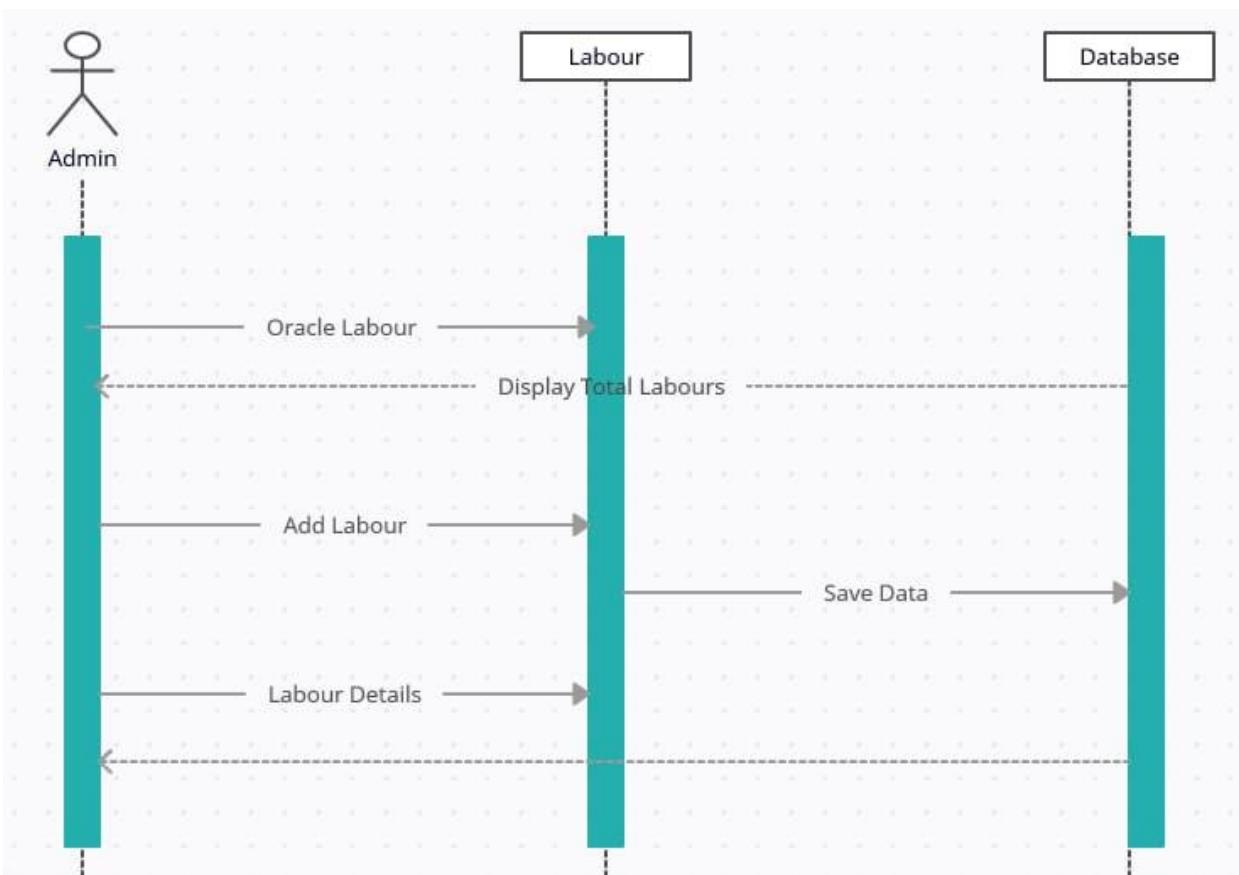
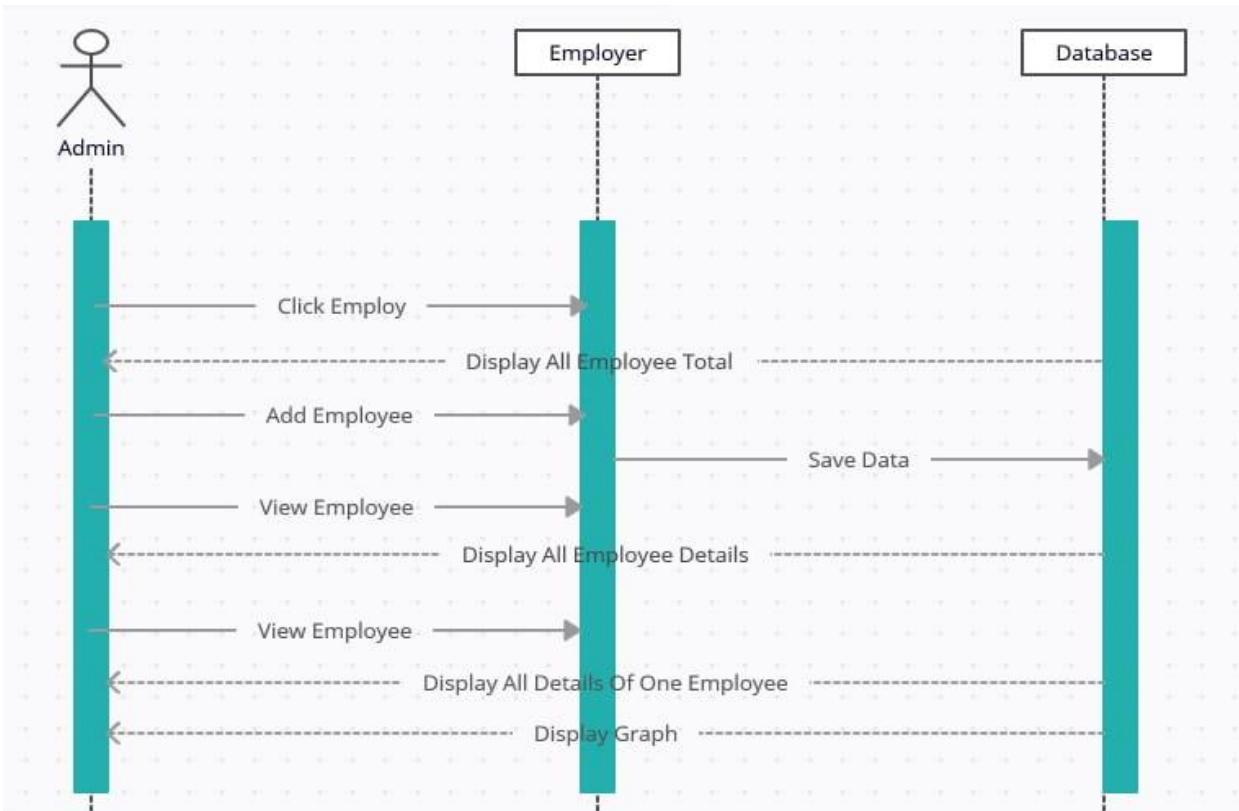


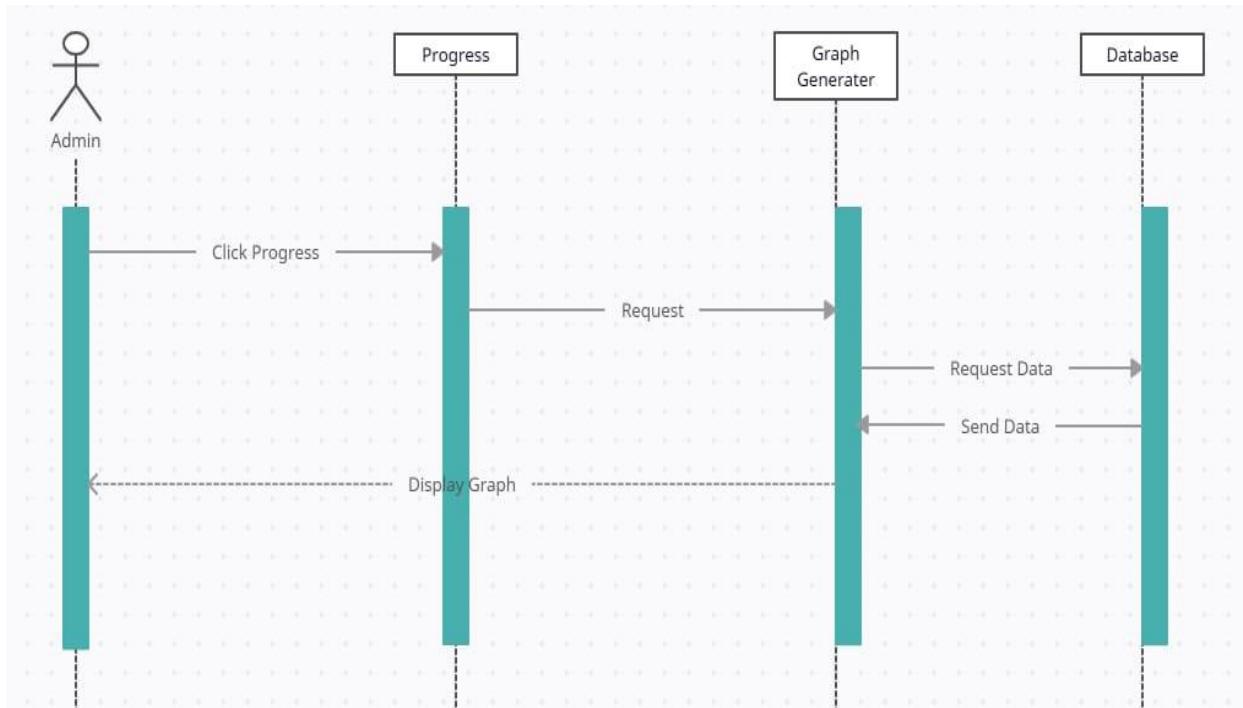
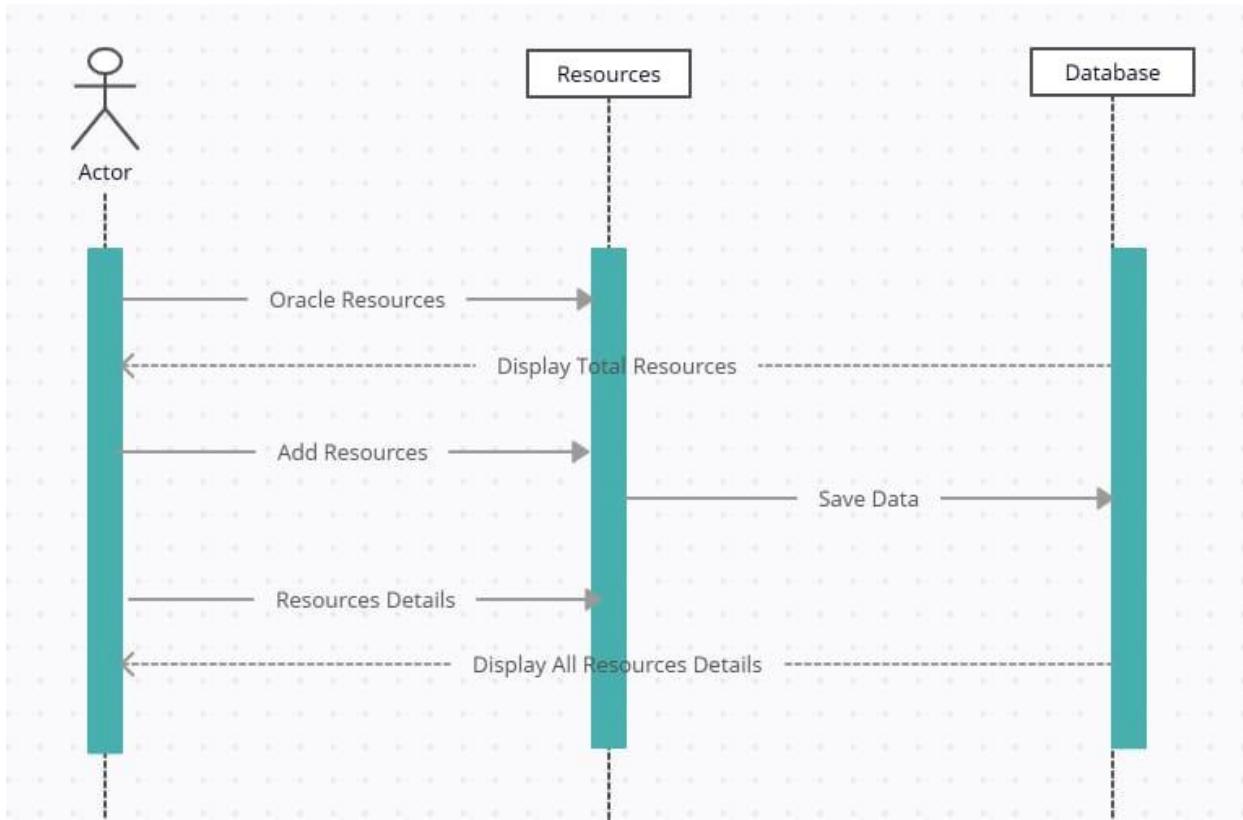
ER diagram



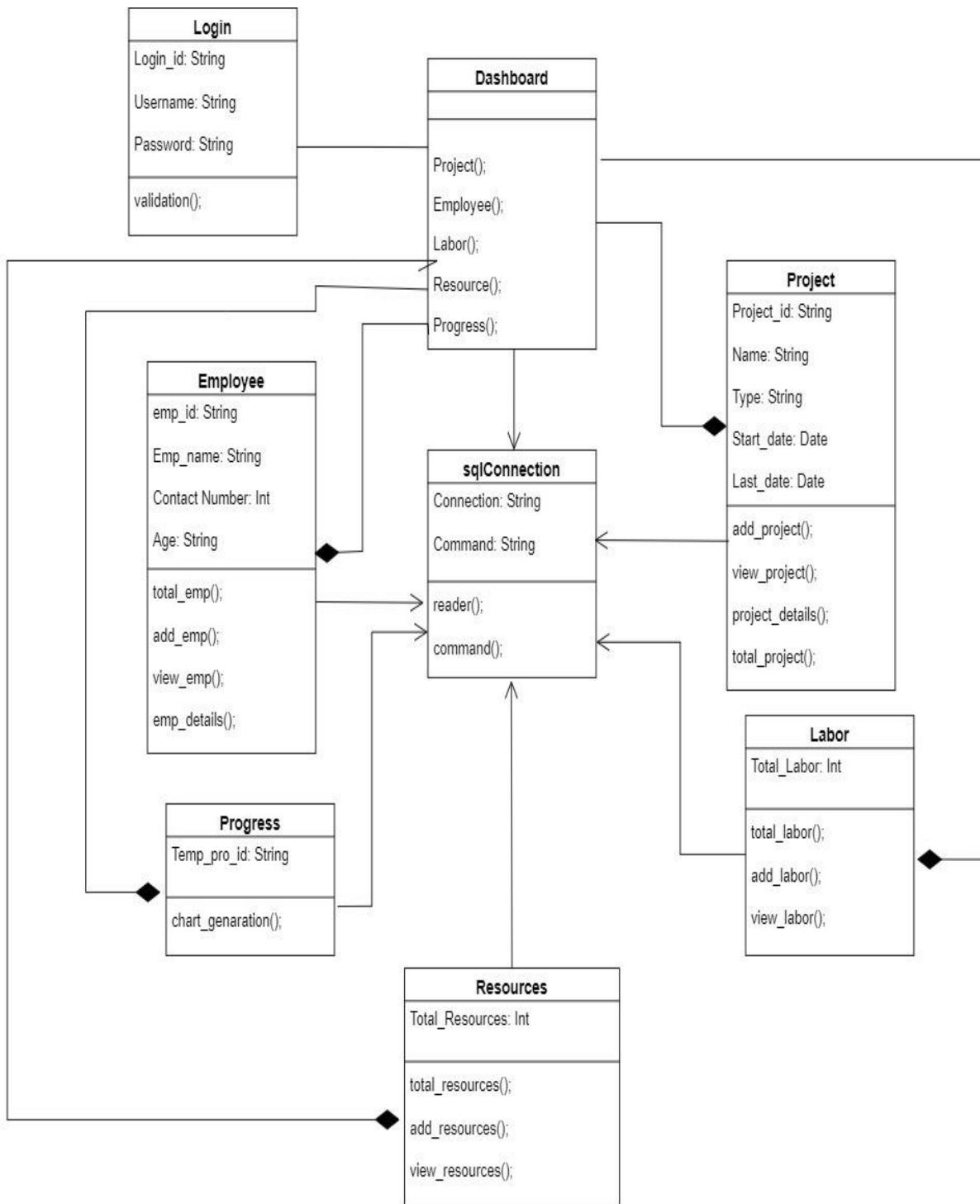
Sequence diagram







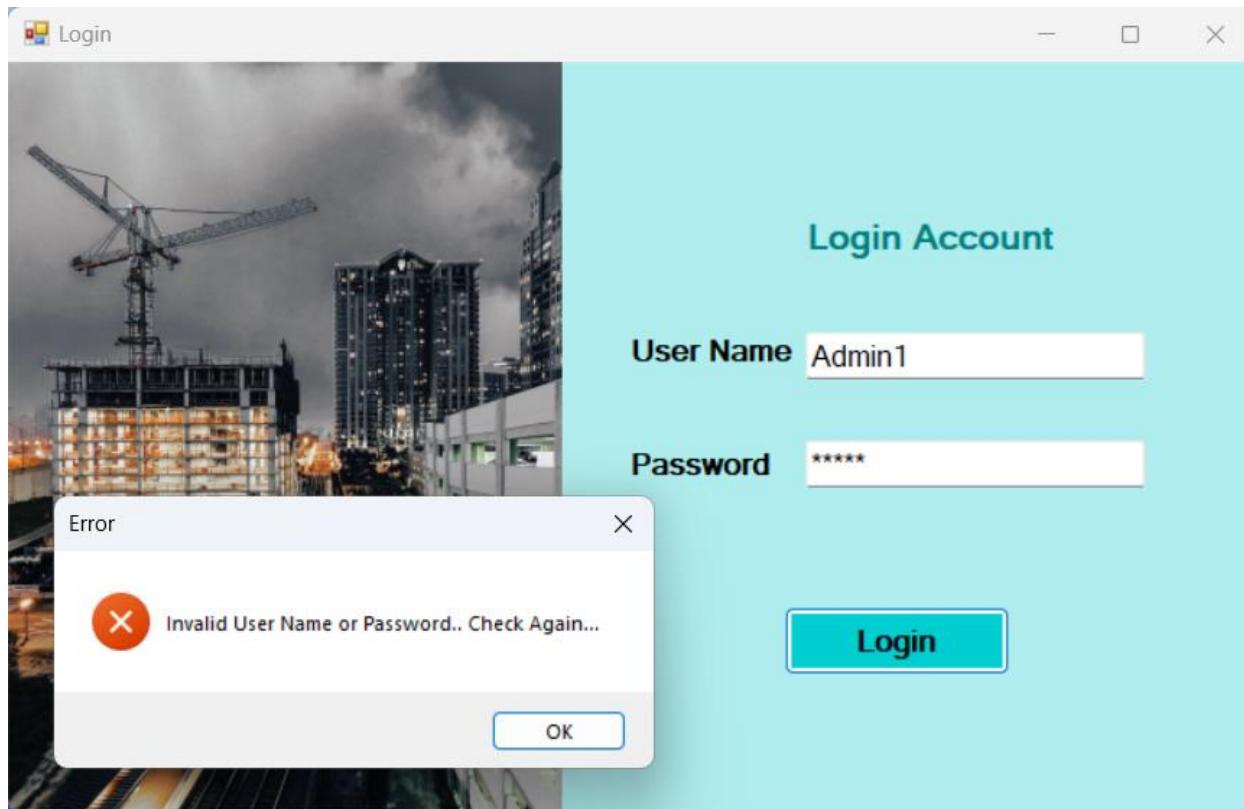
Class diagram

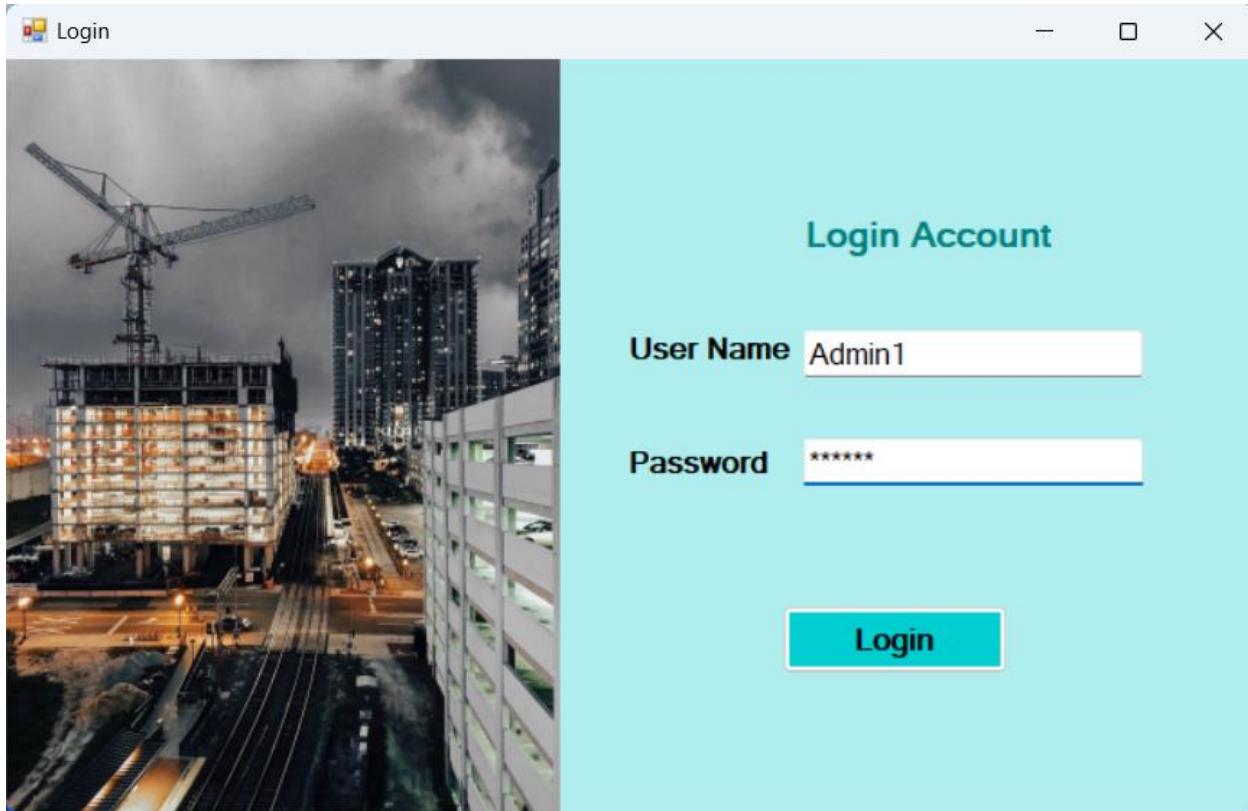




Design of each form in the application & code implementation

Login Page





```
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;

namespace ProjectManagementSystem
{
    public partial class Login : Form
    {
        public Login()
        {
            InitializeComponent();
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
            if (textUser.Text == "Admin1" && textPassword.Text == "admin1")
            {
                Home h1 = new Home();
                this.Hide();
                h1.ShowDialog();
                this.Close();
            }
        }
    }
}
```

```
        }
        else
        {
            MessageBox.Show("Invalid User Name or Password.. Check
Again...", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            textUser.Clear();
            textPassword.Clear();
        }
    }
}
=====
```

Dashboard



```

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;

namespace ProjectManagementSystem
{
    public partial class Home : Form
    {
        public Home()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            button1.BackColor = Color.Cyan;
            button2.BackColor = Color.Coral;
            button3.BackColor = Color.Cyan;
            button4.BackColor = Color.Cyan;
            button5.BackColor = Color.Cyan;
            pictureBox2.Visible = false;
            Employees e1 = new Employees();
            e1.MdiParent = this;
            e1.Dock = DockStyle.Fill;
            e1.Show();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            button1.BackColor = Color.Coral;
            button2.BackColor = Color.Cyan;
            button3.BackColor = Color.Cyan;
            button4.BackColor = Color.Cyan;
            button5.BackColor = Color.Cyan;
            pictureBox2.Visible = false;
            Projects p1 = new Projects();
            p1.MdiParent = this;
            p1.Dock = DockStyle.Fill;
            p1.Show();
        }

        private void panel3_Paint(object sender, PaintEventArgs e)
        {

        }

        private void Home_Load(object sender, EventArgs e)
        {
            date.Text = DateTime.Now.ToString("yyyy-MM-dd");
        }
    }
}

```

```

}

private void button5_Click(object sender, EventArgs e)
{
    button1.BackColor = Color.Cyan;
    button2.BackColor = Color.Cyan;
    button3.BackColor = Color.Cyan;
    button4.BackColor = Color.Cyan;
    button5.BackColor = Color.Coral;
    pictureBox2.Visible = false;
    Progress p2 = new Progress();
    p2.MdiParent = this;
    p2.Dock = DockStyle.Fill;
    p2.Show();
}

public void timer5_Tick(object sender, EventArgs e)
{
    // Update the label with the current date and time
}

private void panel2_Paint(object sender, PaintEventArgs e)
{

}

private void button3_Click(object sender, EventArgs e)
{
    button1.BackColor = Color.Cyan;
    button2.BackColor = Color.Cyan;
    button3.BackColor = Color.Coral;
    button4.BackColor = Color.Cyan;
    button5.BackColor = Color.Cyan;
    pictureBox2.Visible = false;
    Labour l1 = new Labour();
    l1.MdiParent = this;
    l1.Dock = DockStyle.Fill;
    l1.Show();
}

private void button4_Click(object sender, EventArgs e)
{
    button1.BackColor = Color.Cyan;
    button2.BackColor = Color.Cyan;
    button3.BackColor = Color.Cyan;
    button4.BackColor = Color.Coral;
    button5.BackColor = Color.Cyan;
    pictureBox2.Visible = false;
    Resources r1 = new Resources();
    r1.MdiParent = this;
    r1.Dock = DockStyle.Fill;
    r1.Show();
}
}
=====
```

Projects Page



```
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 ProjectManagementSystem
{
    public partial class Projects : Form
    {
        public Projects()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
```

```

}

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

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

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

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

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

private void Projects_Load(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection("Data
Source=localhost\\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
    conn.Open();
    SqlCommand cmd1 = new SqlCommand("Select Count(ProjectId) From
Projects", conn);
    int projectCount = (int)cmd1.ExecuteScalar();
    total.Text = projectCount.ToString();

    SqlCommand cmd2 = new SqlCommand("Select Count(ProjectId) From Projects
Where (Status = 'Completed' or Status = 'Closed')", conn);
    int CompleteCount = (int)cmd2.ExecuteScalar();
    complete.Text = CompleteCount.ToString();
    conn.Close();
}

private void label3_Click(object sender, EventArgs e)
{
}

private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    Home h1 = new Home();
    this.Hide();
    h1.ShowDialog();
    this.Close();
}

```

```
    private void viewAllProjectsToolStripMenuItem_Click(object sender, EventArgs e)
    {
    }

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

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

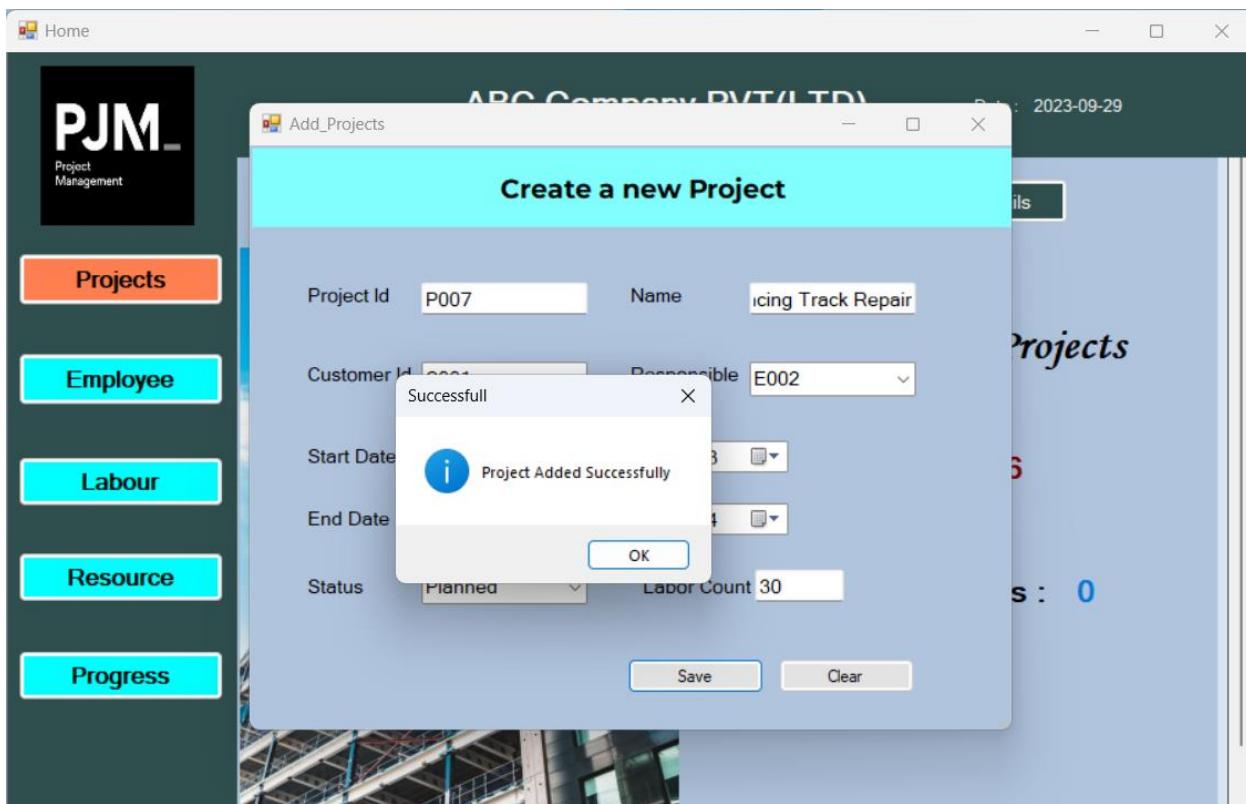
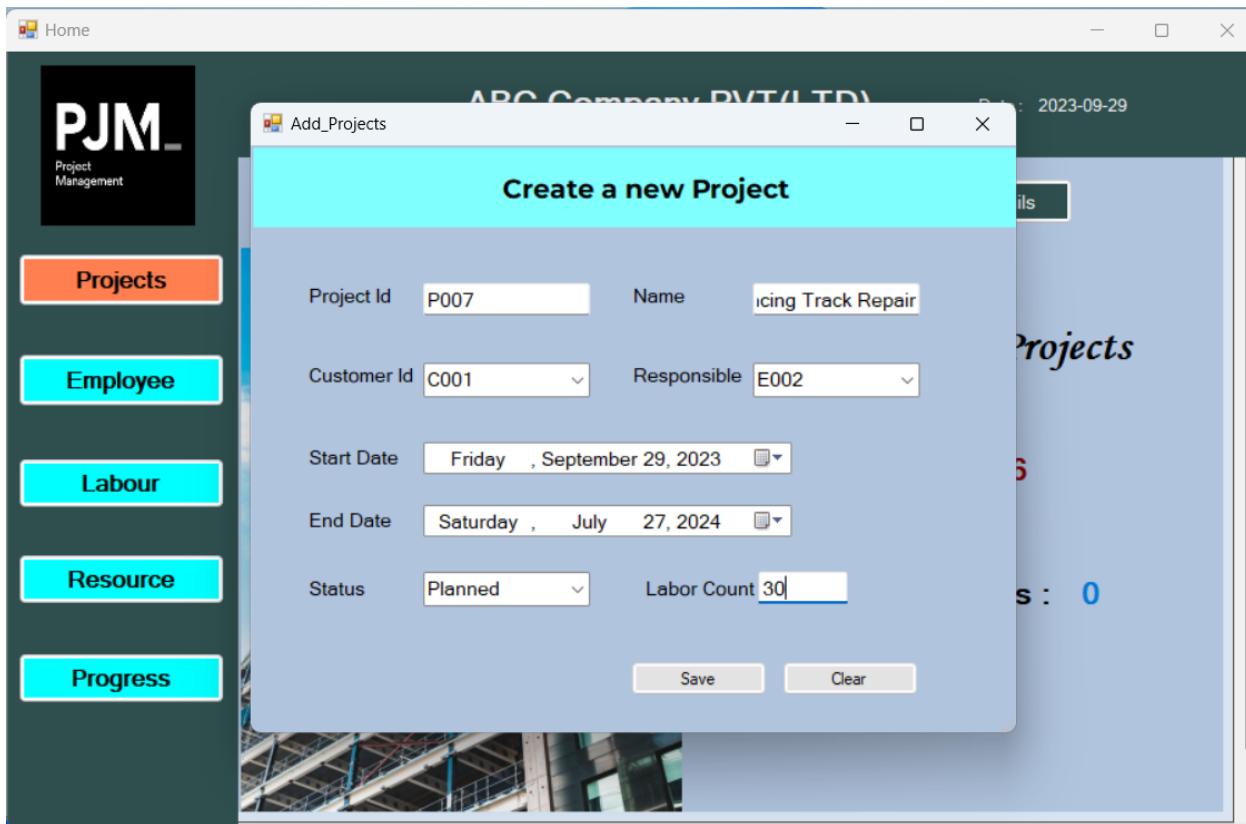
    private void searchProjectsToolStripMenuItem_Click(object sender, EventArgs e)
    {
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Add_Projects ap1 = new Add_Projects();
        ap1.ShowDialog();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        View_Projects vp = new View_Projects();
        vp.ShowDialog();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        Search_Project sp = new Search_Project();
        sp.ShowDialog();
    }
}
```

Add Project Form



```

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 ProjectManagementSystem
{
    public partial class Add_Projects : Form
    {
        public Add_Projects()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Projects p1 = new Projects();
            this.Hide();
            p1.ShowDialog();
            this.Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection conn = new SqlConnection("Data
Source=localhost\SQLEXPRESS;Initial Catalog=PMSSDB;Integrated Security=True");
            conn.Open();
            SqlCommand cmd = new SqlCommand("Insert into Projects values
(@ProjectId,@Name,@StartDate,@EndDate,@Responsible,@CustomerId,@LaborCount,@Status)"
, conn);
            cmd.Parameters.AddWithValue("@ProjectId", txtId.Text);
            cmd.Parameters.AddWithValue("@Name", txtName.Text);
            cmd.Parameters.AddWithValue("@StartDate", DateTime.Parse(date1.Text));
            cmd.Parameters.AddWithValue("@EndDate", DateTime.Parse(date2.Text));
            cmd.Parameters.AddWithValue("@Responsible", resBox.Text);
            cmd.Parameters.AddWithValue("@CustomerId", customerBox.Text);
            cmd.Parameters.AddWithValue("@LaborCount", int.Parse(txtLabor.Text));
            cmd.Parameters.AddWithValue("@Status", statusBox.Text);
            cmd.ExecuteNonQuery();

            conn.Close();

            MessageBox.Show("Project Added Successfully", "Successfull",
MessageBoxButtons.OK, MessageBoxIcon.Information);

            txtId.Clear();
            txtName.Clear();
            date1.Text = DateTime.Now.ToString();
            date2.Text = DateTime.Now.ToString();
            resBox.Text = "";
            customerBox.Text = "";
            txtLabor.Clear();
        }
    }
}

```

```

        statusBox.Text = "";
    }

    private void button3_Click(object sender, EventArgs e)
    {
        txtId.Clear();
        txtName.Clear();
        date1.Text = DateTime.Now.ToString();
        date2.Text = DateTime.Now.ToString();
        resBox.Text = "";
        customerBox.Text = "";
        txtLabor.Clear();
        statusBox.Text = "";
    }

    private void Add_Projects_Load(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection("Data
Source=localhost\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
        conn.Open();
        SqlCommand cmd1 = new SqlCommand("Select EmployeeId From EmpTemp Where
Role = 'Manager'", conn);
        List<string> EmpIds = new List<string>();

        using (SqlDataReader reader1 = cmd1.ExecuteReader())
        {
            while (reader1.Read())
            {
                string empId = reader1["EmployeeId"].ToString(); // Assuming
ProjectId is a string
                EmpIds.Add(empId);
            }
        }

        // Convert the list to an array if needed
        string[] empIdsArray = EmpIds.ToArray();

        // Populate a ListBox or another control with the array
        resBox.Items.AddRange(empIdsArray);

        SqlCommand cmd2 = new SqlCommand("Select CustomerId From Customer",
conn);
        List<string> custIds = new List<string>();

        using (SqlDataReader reader2 = cmd2.ExecuteReader())
        {
            while (reader2.Read())
            {
                string custId = reader2["CustomerId"].ToString(); // Assuming
ProjectId is a string
                custIds.Add(custId);
            }
        }

        // Convert the list to an array if needed
        string[] custIdsArray = custIds.ToArray();
    }
}

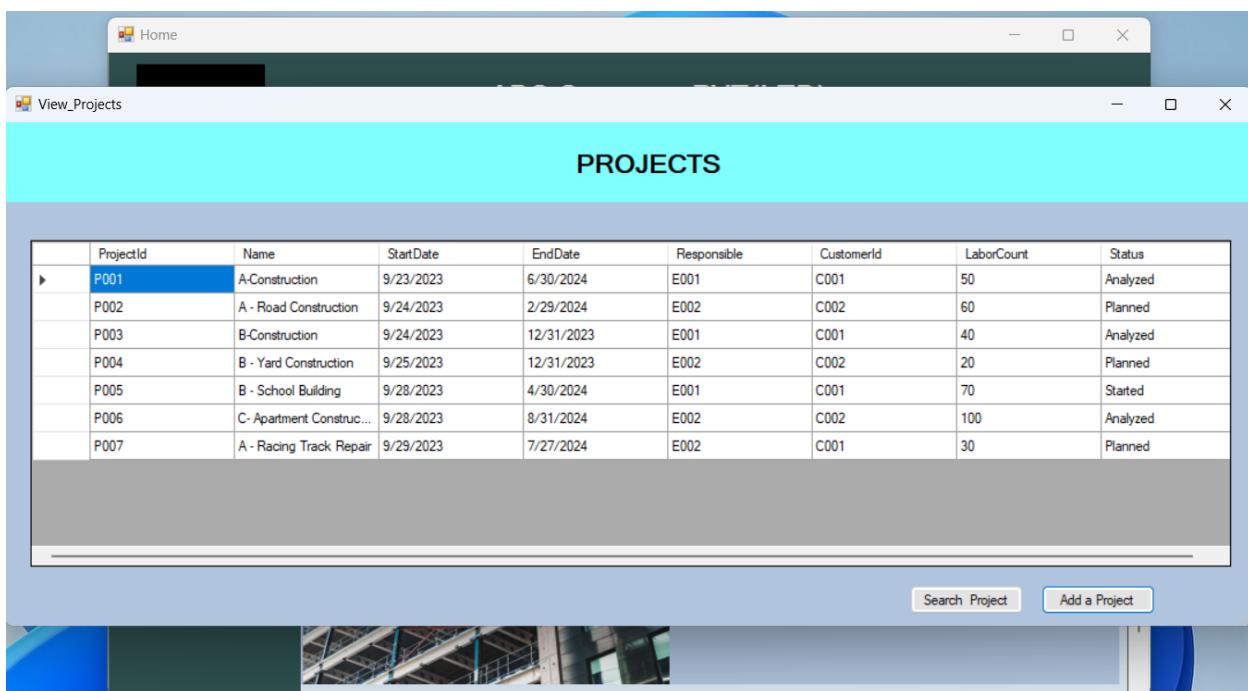
```

```

        // Populate a ListBox or another control with the array
        customerBox.Items.AddRange(custIdsArray);

        conn.Close();
    }
}
=====
```

View Project Form



```

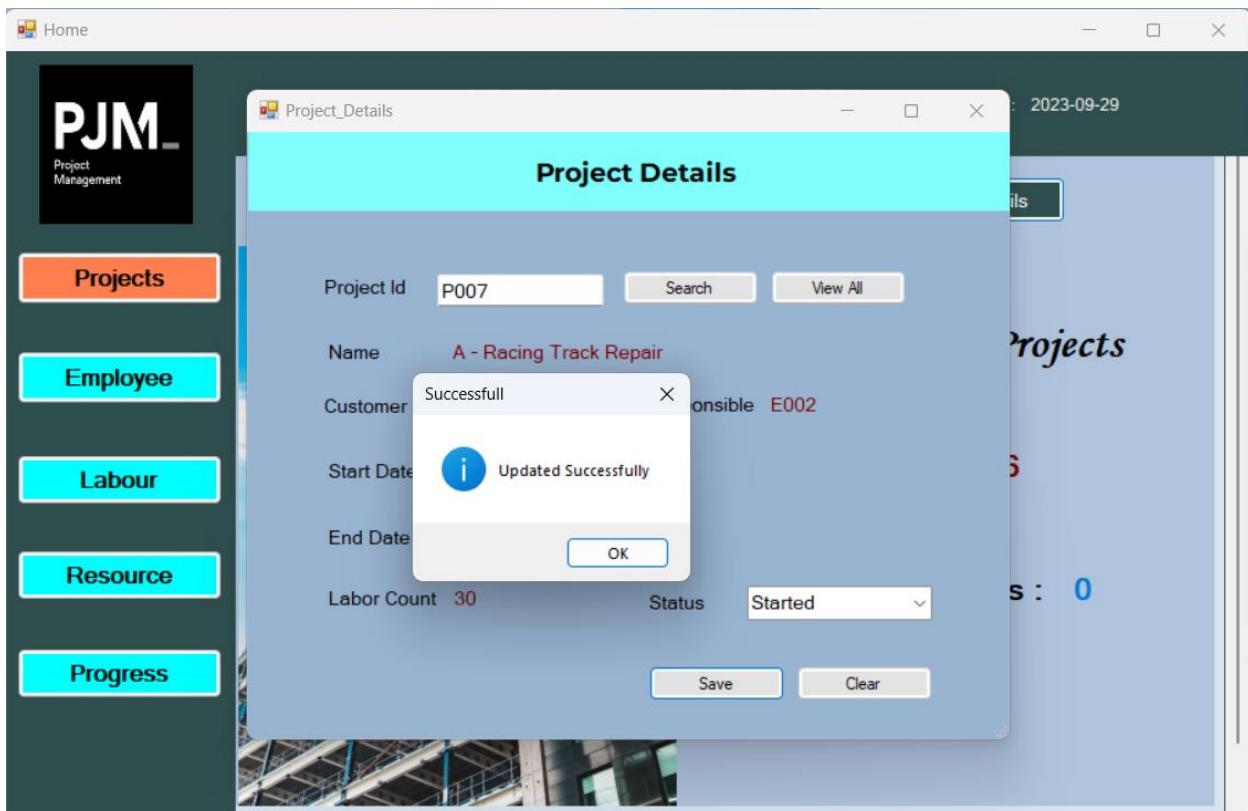
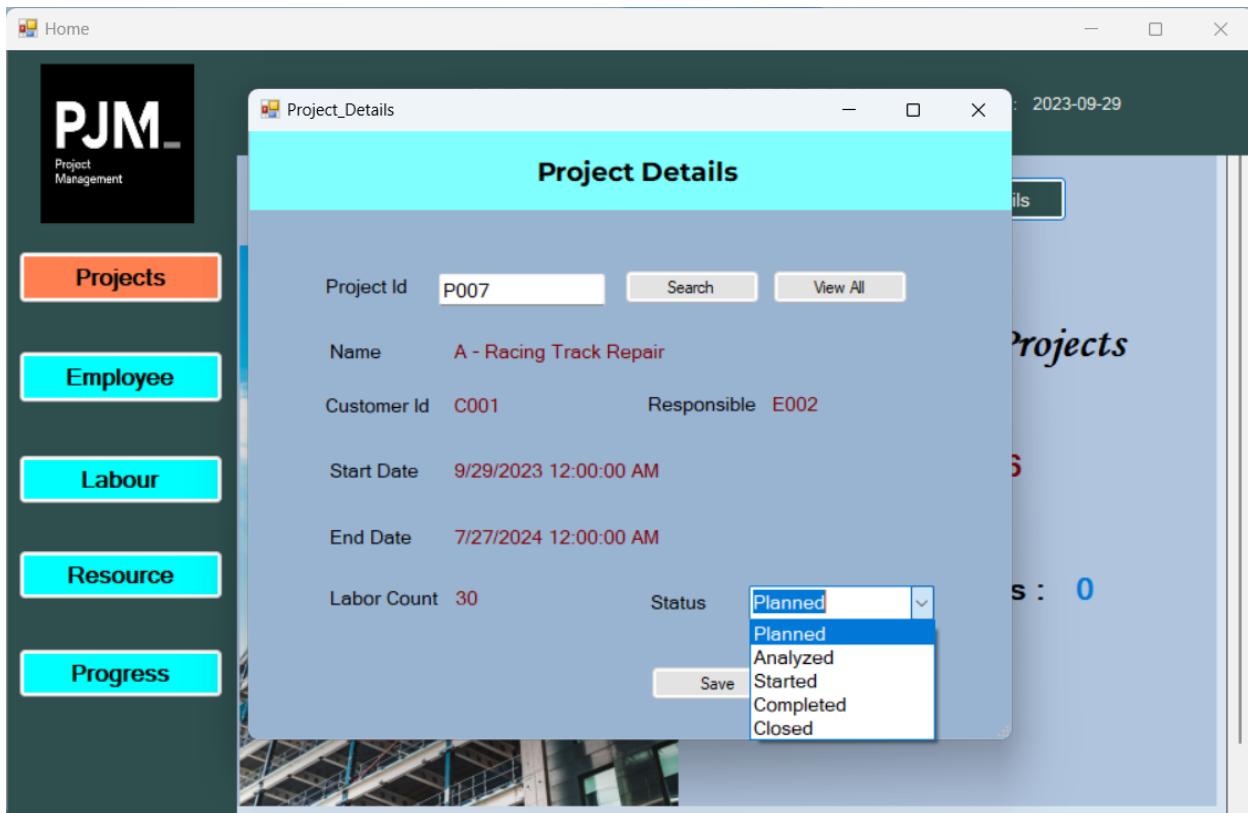
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ProjectManagementSystem
{
    public partial class View_Projects : Form
```

```
{  
    public View_Projects()  
    {  
        InitializeComponent();  
    }  
  
    private void button1_Click(object sender, EventArgs e)  
    {  
        Add_Projects ap1 = new Add_Projects();  
        ap1.ShowDialog();  
    }  
  
    private void button2_Click(object sender, EventArgs e)  
    {  
        Projects p1 = new Projects();  
        this.Hide();  
        p1.ShowDialog();  
        this.Close();  
    }  
  
    private void label1_Click(object sender, EventArgs e)  
    {}  
  
    private void View_Projects_Load(object sender, EventArgs e)  
    {  
        this.sqlDataAdapter1.Fill(this.projectsSet1.Projects);  
    }  
  
    private void button2_Click_1(object sender, EventArgs e)  
    {  
        Search_Project sp1 = new Search_Project();  
        sp1.ShowDialog();  
    }  
}
```

=====

Project Details Form



```

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 System.Data.OleDb;
using System.Data.SqlClient;

namespace ProjectManagementSystem
{
    public partial class Search_Project : Form
    {
        public Search_Project()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Home h1 = new Home();
            this.Hide();
            h1.ShowDialog();
            this.Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection conn = new SqlConnection("Data
Source=localhost\\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
            conn.Open();
            SqlCommand cmd = new SqlCommand("Select * from Projects where
ProjectId=@id", conn);
            cmd.Parameters.AddWithValue("@id", txtId.Text);
            SqlDataReader reader1;
            reader1 = cmd.ExecuteReader();
            if(reader1.Read())
            {
                txtName.Text = reader1["Name"].ToString();
                Responsible.Text = reader1["Responsible"].ToString();
                CustomerId.Text = reader1["CustomerId"].ToString();
                StartDate.Text = reader1["StartDate"].ToString();
                EndDate.Text = reader1["EndDate"].ToString();
                LaborCount.Text = reader1["LaborCount"].ToString();
                statusBox.Text = reader1["Status"].ToString();
            }
            else
            {
                MessageBox.Show("Invalid Project ID Check Again..", "Data Not
Found", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void button3_Click(object sender, EventArgs e)
    }
}

```

```

    {
        txtId.Clear();
        txtName.Text = "";
        Responsible.Text = "";
        CustomerId.Text = "";
        StartDate.Text = "";
        EndDate.Text = "";
        LaborCount.Text = "";
        statusBox.Text = "";
    }

    private void button2_Click_1(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection("Data
Source=localhost\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
        conn.Open();
        SqlCommand cmd = new SqlCommand("update Projects set Status=@Status
Where ProjectId=@id ", conn);
        cmd.Parameters.AddWithValue("@id", txtId.Text);
        cmd.Parameters.AddWithValue("@Status", statusBox.Text);
        cmd.ExecuteNonQuery();

        conn.Close();

        MessageBox.Show("Updated Successfully", "Successfull",
MessageBoxButtons.OK, MessageBoxIcon.Information);

        txtId.Clear();
        txtName.Text = "";
        Responsible.Text = "";
        CustomerId.Text = "";
        StartDate.Text = "";
        EndDate.Text = "";
        LaborCount.Text = "";
        statusBox.Text = "";
    }

    private void button4_Click(object sender, EventArgs e)
    {
        View_Projects vp1 = new View_Projects();
        this.Hide();
        vp1.ShowDialog();
        this.Close();
    }
}

```

Employee Page



```
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 ProjectManagementSystem
{
    public partial class Employees : Form
    {
        public Employees()
        {
            InitializeComponent();
        }

        private void Employees_Load(object sender, EventArgs e)
        {
            SqlConnection conn = new SqlConnection("Data
Source=localhost\\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
        }
    }
}
```

```
        conn.Open();
        SqlCommand cmd1 = new SqlCommand("Select Count(EmployeeId) From
EmpTemp", conn);
        int empCount = (int)cmd1.ExecuteScalar();
        total.Text = empCount.ToString();

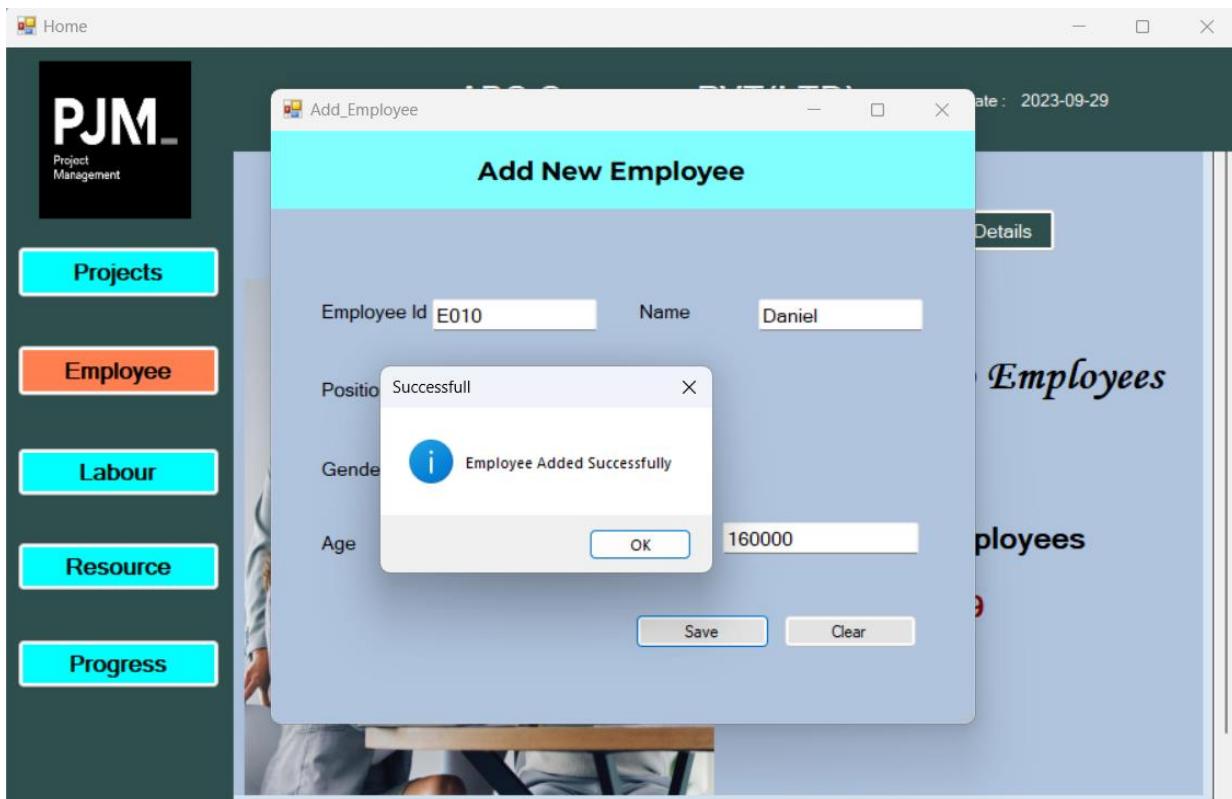
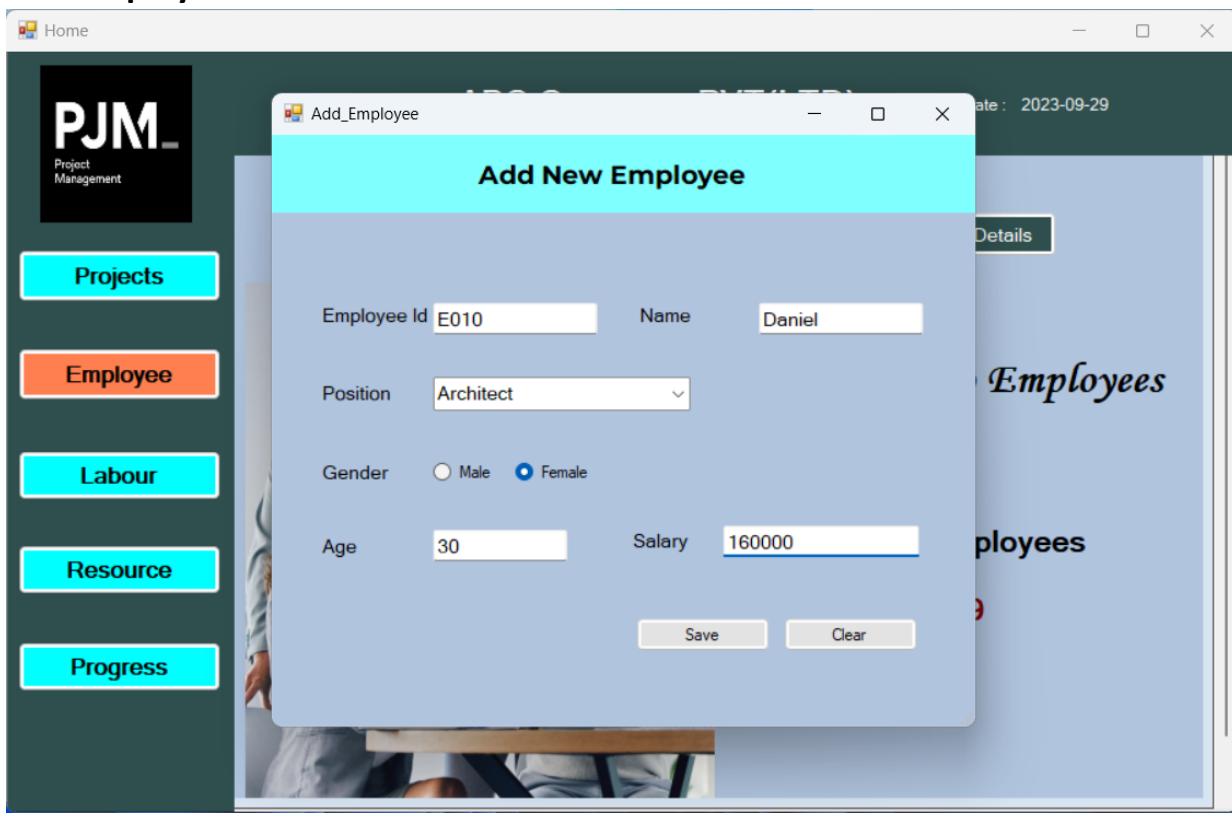
        conn.Close();
    }

    private void button1_Click(object sender, EventArgs e)
{
    Add_Employee ae1 = new Add_Employee();
    ae1.ShowDialog();
}

private void button3_Click(object sender, EventArgs e)
{
    Employee_Details ed1 = new Employee_Details();
    ed1.ShowDialog();
}

private void button2_Click(object sender, EventArgs e)
{
    View_Employees ve1 = new View_Employees();
    ve1.ShowDialog();
}
}
```

Add Employee Form



```

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 ProjectManagementSystem
{
    public partial class Add_Employee : Form
    {
        public Add_Employee()
        {
            InitializeComponent();
        }

        private void Add_Employee_Load(object sender, EventArgs e)
        {

        }

        private void button3_Click(object sender, EventArgs e)
        {
            txtId.Clear();
            txtName.Clear();
            roleBox.Text = "";
            radioButton1.Checked = false;
            radioButton2.Checked = false;
            age.Clear();
            empSalary.Clear();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection conn = new SqlConnection("Data
Source=localhost\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
            conn.Open();
            SqlCommand cmd = new SqlCommand("Insert into EmpTemp values
(@EmployeeId,@EmpName,@Role,@Age,@Gender,@Salary)", conn);
            cmd.Parameters.AddWithValue("@EmployeeId", txtId.Text);
            cmd.Parameters.AddWithValue("@EmpName", txtName.Text);
            cmd.Parameters.AddWithValue("@Role", roleBox.Text);
            cmd.Parameters.AddWithValue("@Age", int.Parse(age.Text));
            if (radioButton1.Checked)
            {
                cmd.Parameters.AddWithValue("@Gender", "Male");
            }
            if (radioButton2.Checked)
            {
                cmd.Parameters.AddWithValue("@Gender", "Female");
            }
            cmd.Parameters.AddWithValue("@Salary", int.Parse(empSalary.Text));
            cmd.ExecuteNonQuery();
        }
    }
}

```

```
conn.Close();

MessageBox.Show("Employee Added Successfully", "Successfull",
MessageBoxButtons.OK, MessageBoxIcon.Information);

txtId.Clear();
txtName.Clear();
roleBox.Text = "";
radioButton1.Checked = false;
radioButton2.Checked = false;
age.Clear();
empSalary.Clear();

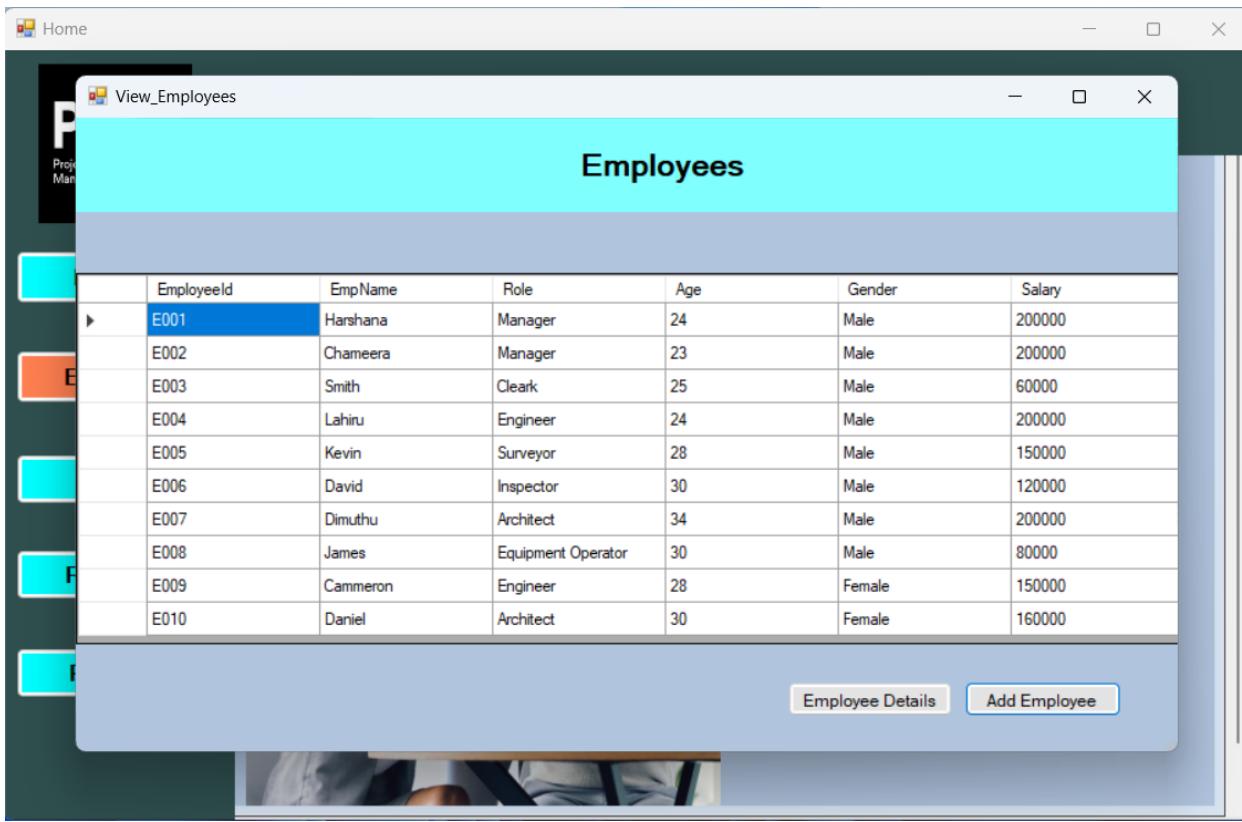
this.Refresh();
}

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

}

=====
```

View Employee Form



```
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;

namespace ProjectManagementSystem
{
    public partial class View_Employees : Form
    {
        public View_Employees()
        {
            InitializeComponent();
        }

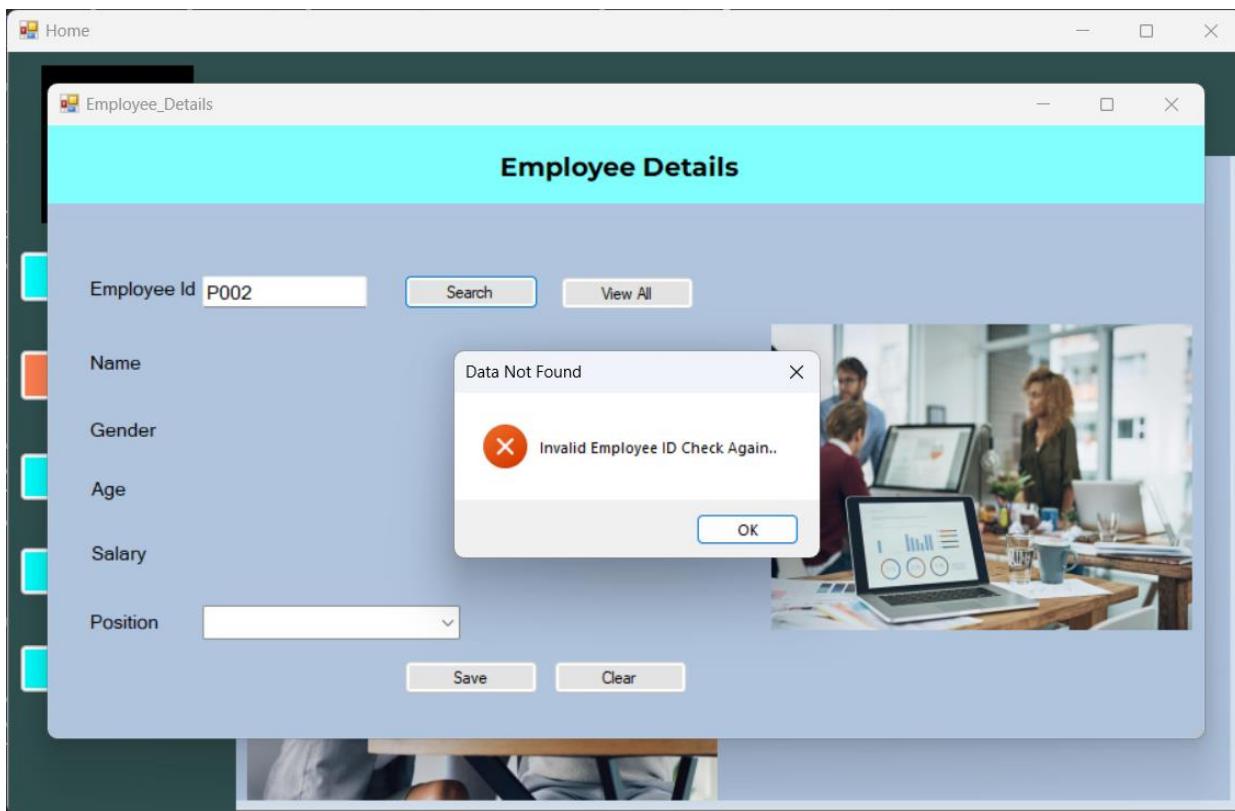
        private void button1_Click(object sender, EventArgs e)
        {
            Add_Employee ae1 = new Add_Employee();
            ae1.ShowDialog();
        }
    }
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    Employee_Details ed1 = new Employee_Details();
    ed1.ShowDialog();
}

private void View_Employees_Load(object sender, EventArgs e)
{
    this.sqlDataAdapter1.Fill(this.employeeSet.EmpTemp);
}
}

=====
```

Employee Details Form



Home

Employee_Details

Employee Details

Employee Id E002

Name Chameera

Gender Male

Age 23

Salary 200000

Position Manager

Search View All

Save Clear

Progress

Project	Progress
P002	10
P004	10
P007	60
P008	35

Data Series

Home

Employee_Details

Employee Details

Employee Id E010

Name Daniel

Gender Female

Age 30

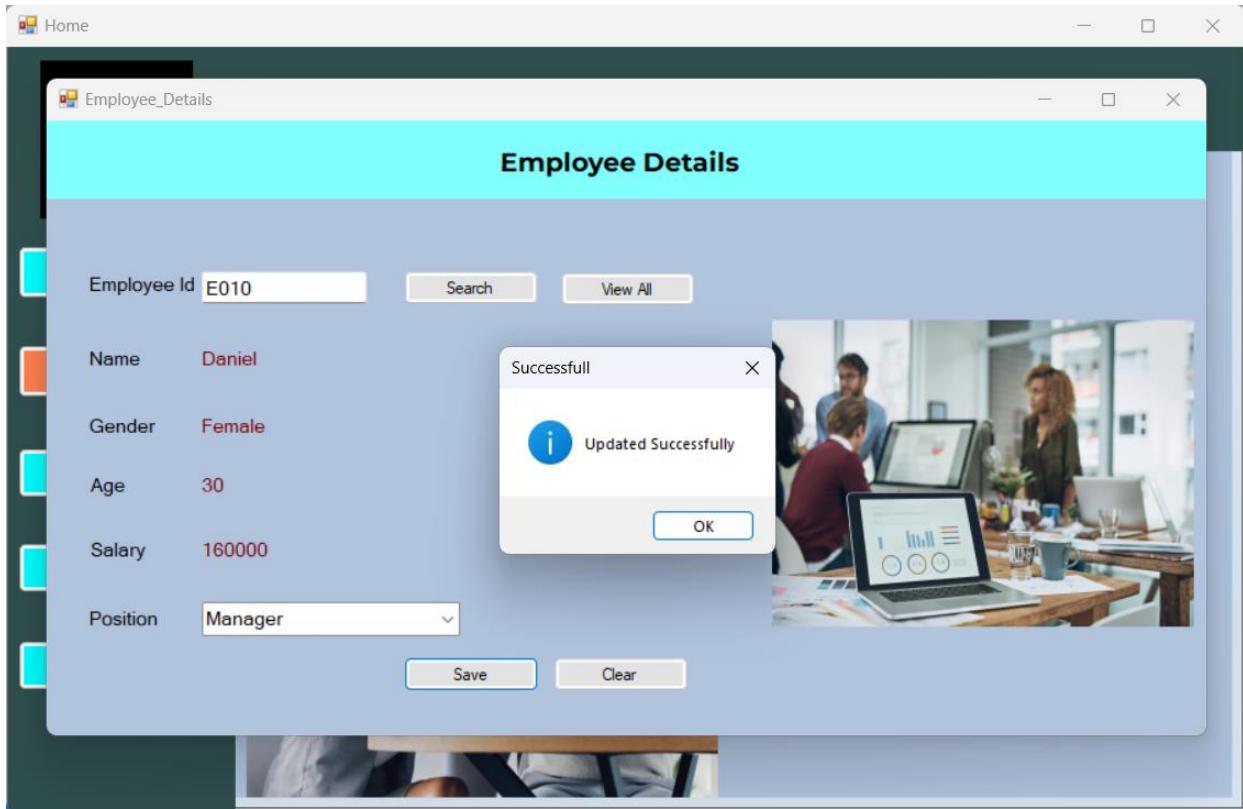
Salary 160000

Position Architect

Search View All

Save Clear

A photograph showing a group of four people in an office setting. One person is standing and pointing at a laptop screen, while others are seated or standing around a table with papers and a coffee cup.



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Runtime.InteropServices.ComTypes;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;
using System.Xml.Linq;

namespace ProjectManagementSystem
{
    public partial class Employee_Details : Form
    {
        public Employee_Details()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            SqlConnection conn = new SqlConnection("Data
Source=localhost\\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
        }
    }
}
```

```

conn.Open();
SqlCommand cmd = new SqlCommand("Select * from EmpTemp where EmployeeId=@id", conn);
cmd.Parameters.AddWithValue("id", txtId.Text);
SqlDataReader reader1;
reader1 = cmd.ExecuteReader();
if (reader1.Read())
{
    empName.Text = reader1["EmpName"].ToString();
    roleBox.Text = reader1["Role"].ToString();
    empAge.Text = reader1["Age"].ToString();
    empGender.Text = reader1["Gender"].ToString();
    Salary.Text = reader1["Salary"].ToString();

}
else
{
    MessageBox.Show("Invalid Employee ID Check Again..", "Data Not Found", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
reader1.Close();
conn.Close();
if (roleBox.Text == "Manager")
{
    pictureBox1.Visible = false;
    // Create a series
    Series series = new Series("Data Series");
    series.ChartType = SeriesChartType.Column;
    chart1.Series.Add(series);

    // Connect to the database
    string connectionString = "Data Source=localhost\\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True";
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        SqlCommand cmd1 = new SqlCommand("SELECT ProjectId, Status FROM Projects Where Responsible=@eId", connection);
        cmd1.Parameters.AddWithValue("eId", txtId.Text);

        // Read data from the database and add data points to the series
        SqlDataReader reader2 = cmd1.ExecuteReader();
        while (reader2.Read())
        {
            string Id = reader2["ProjectId"].ToString();
            int value = 0;

            switch (reader2["Status"].ToString())
            {
                case "Planned":
                    value = 10;
                    break;
                case "Analyzed":
                    value = 35;
                    break;
                case "Started":

```

```

                value = 60;
                break;
            case "Completed":
                value = 80;
                break;
            case "Closed":
                value = 100;
                break;
                // Handle other statuses as needed
        }

        // Add data points based on the "Name" column value
        series.Points.AddXY(Id, value);

        // Label the X-axis directly
        chart1.ChartAreas[0].AxisX.Title = "Project";

        // Label the Y-axis directly
        chart1.ChartAreas[0].AxisY.Title = "Progress";

        // Refresh the chart to display the changes
        chart1.Refresh();

    }

}

else
{
    pictureBox1.Visible = true;
}

private void button1_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection("Data
Source=localhost\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
    conn.Open();
    SqlCommand cmd = new SqlCommand("update EmpTemp set Role=@Role Where
EmployeeId=@id ", conn);
    cmd.Parameters.AddWithValue("@id", txtId.Text);
    cmd.Parameters.AddWithValue("@Role", roleBox.Text);
    cmd.ExecuteNonQuery();

    conn.Close();

    MessageBox.Show("Updated Successfully", "Successfull",
MessageBoxButtons.OK, MessageBoxIcon.Information);

    txtId.Clear();
    empName.Text = "";
    empAge.Text = "";
    empGender.Text = "";
    roleBox.Text = "";
    Salary.Text = "";
    pictureBox1.Visible = true;
}

```

```
private void button3_Click(object sender, EventArgs e)
{
    txtId.Clear();
    empName.Text = "";
    empAge.Text = "";
    empGender.Text = "";
    roleBox.Text = "";
    Salary.Text = "";
    pictureBox1.Visible = true;
}

private void button4_Click(object sender, EventArgs e)
{
    View_Employees ve1 = new View_Employees();
    ve1.ShowDialog();
}
}
```

Labour Page



```
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 ProjectManagementSystem
{
    public partial class Labour : Form
    {
        public Labour()
        {
            InitializeComponent();
        }

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

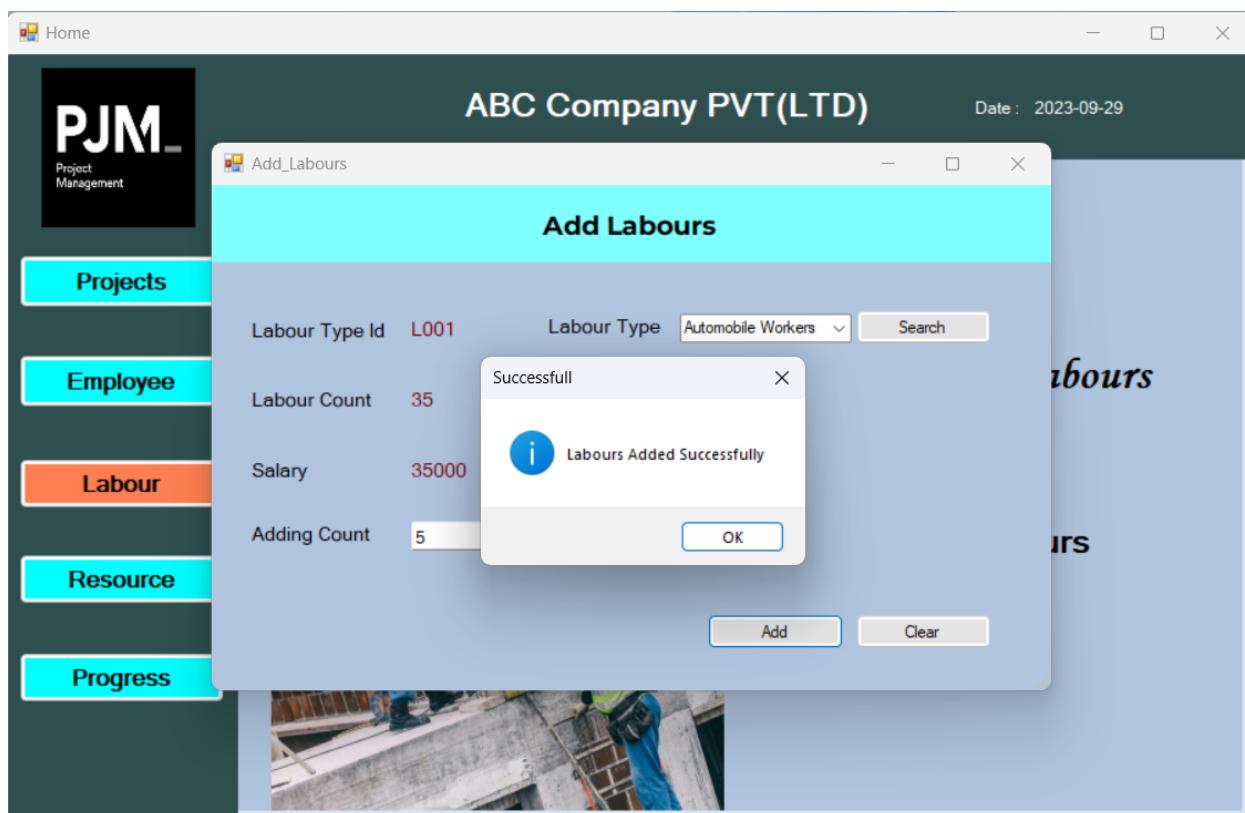
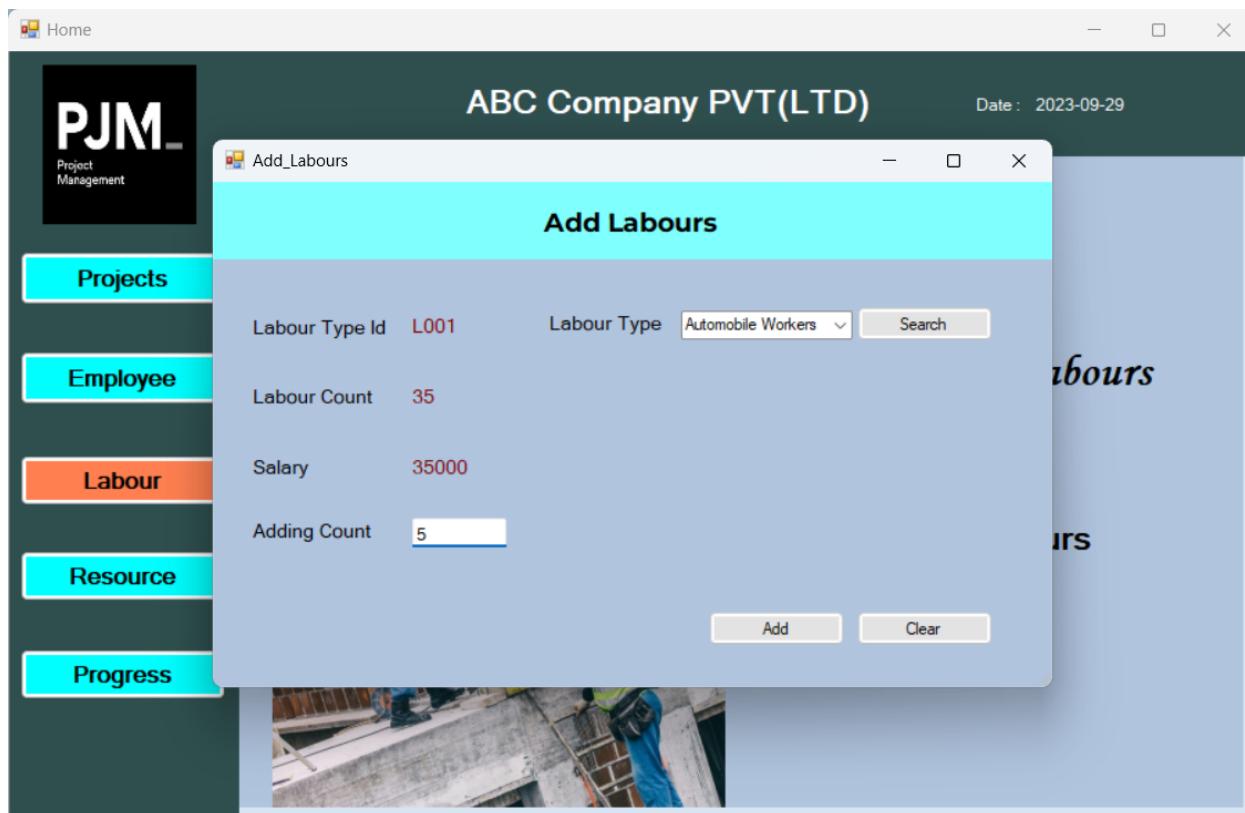
```
    SqlConnection conn = new SqlConnection("Data
Source=localhost\\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
    conn.Open();
    SqlCommand cmd1 = new SqlCommand("Select SUM(Count) From Labours",
conn);
    int lbrCount = (int)cmd1.ExecuteScalar();
    total.Text = lbrCount.ToString();

    conn.Close();
}

private void button1_Click(object sender, EventArgs e)
{
    Add_Labours ae1 = new Add_Labours();
    ae1.ShowDialog();
}

private void button2_Click(object sender, EventArgs e)
{
    Labour_Details ld1 = new Labour_Details();
    ld1.ShowDialog();
}
}
```

Add Labours Form



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Runtime.InteropServices.ComTypes;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Xml.Linq;

namespace ProjectManagementSystem
{
    public partial class Add_Labours : Form
    {
        public Add_Labours()
        {
            InitializeComponent();
        }

        private void Add_Labours_Load(object sender, EventArgs e)
        {
            if(txtId.Text == "")
            {
                textCountBox.Visible = false;
            }
            else
            {
                textCountBox.Visible = true;
            }

            SqlConnection conn = new SqlConnection("Data
Source=localhost\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
            conn.Open();
            SqlCommand cmd1 = new SqlCommand("Select LabourType From Labours",
conn);
            List<string> LabourTypes = new List<string>();

            using (SqlDataReader reader1 = cmd1.ExecuteReader())
            {
                while (reader1.Read())
                {
                    string labourType = reader1["LabourType"].ToString();
                    LabourTypes.Add(labourType);
                }
            }

            string[] labourTypeArray = LabourTypes.ToArray();
            comboBox1.Items.AddRange(labourTypeArray);

            conn.Close();
        }

        private void button3_Click(object sender, EventArgs e)
        {

```

```

        txtId.Text = "";
        txtCount.Text = "";
        txtSalary.Text = "";
        comboBox1.Text = "";
        textCountBox.Clear();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection("Data
Source=localhost\\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
        conn.Open();
        SqlCommand cmd = new SqlCommand("Select * from Labours where
LabourType=@type", conn);
        cmd.Parameters.AddWithValue("type", comboBox1.Text);
        SqlDataReader reader1;
        reader1 = cmd.ExecuteReader();
        if (reader1.Read())
        {
            txtId.Text = reader1["LabourTypeId"].ToString();
            txtSalary.Text = reader1["Salary"].ToString();
            txtCount.Text = reader1["Count"].ToString();
        }
        else
        {
            MessageBox.Show("Invalid Labour Check Again..", "Data Not Found",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }

        if (txtId.Text == String.Empty)
        {
            textCountBox.Visible = false;
        }
        else
        {
            textCountBox.Visible = true;
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection("Data
Source=localhost\\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
        conn.Open();
        SqlCommand cmd = new SqlCommand("UPDATE Labours SET Count=@Count WHERE
LabourTypeId=@id", conn);
        cmd.Parameters.AddWithValue("@id", txtId.Text);
        cmd.Parameters.AddWithValue("@Count", int.Parse(txtCount.Text) +
int.Parse(textCountBox.Text));
        cmd.ExecuteNonQuery();

        conn.Close();

        MessageBox.Show("Labours Added Successfully", "Successfull",
MessageBoxButtons.OK, MessageBoxIcon.Information);

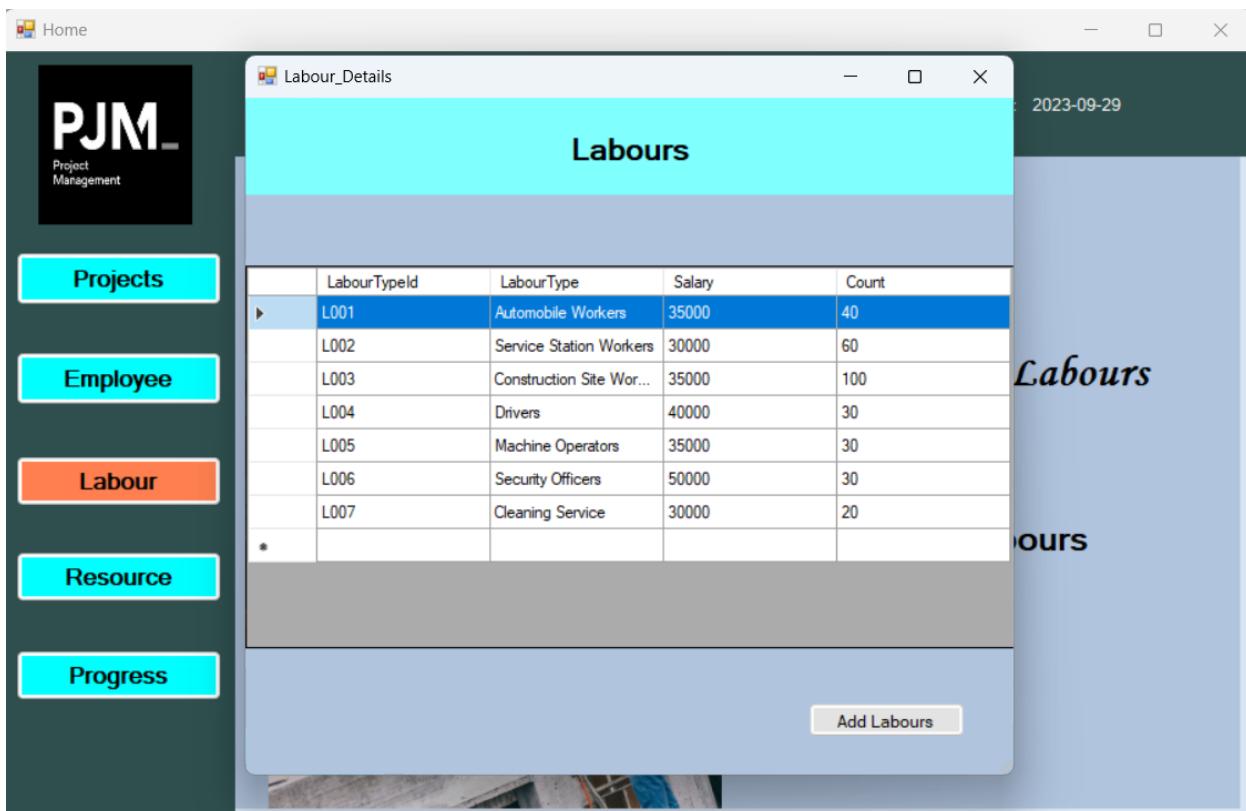
        txtId.Text = "";
        txtCount.Text = "";
    }
}

```

```

        txtSalary.Text = "";
        comboBox1.Text = "";
        textCountBox.Clear();
    }
}
=====
```

Labours Details Form



```

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;

namespace ProjectManagementSystem
{
    public partial class Labour_Details : Form
```

```

{
    public Labour_Details()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Add_Labours al1 = new Add_Labours();
        al1.ShowDialog();
    }

    private void Labour_Details_Load(object sender, EventArgs e)
    {
        this.sqlDataAdapter1.Fill(this.laboursSet1.Labours);
    }

    private void label1_Click(object sender, EventArgs e)
    {
    }
}
=====

```

Resource Page



```

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 ProjectManagementSystem
{
    public partial class Resources : Form
    {
        public Resources()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Add_Resources ar1 = new Add_Resources();
            ar1.ShowDialog();
        }

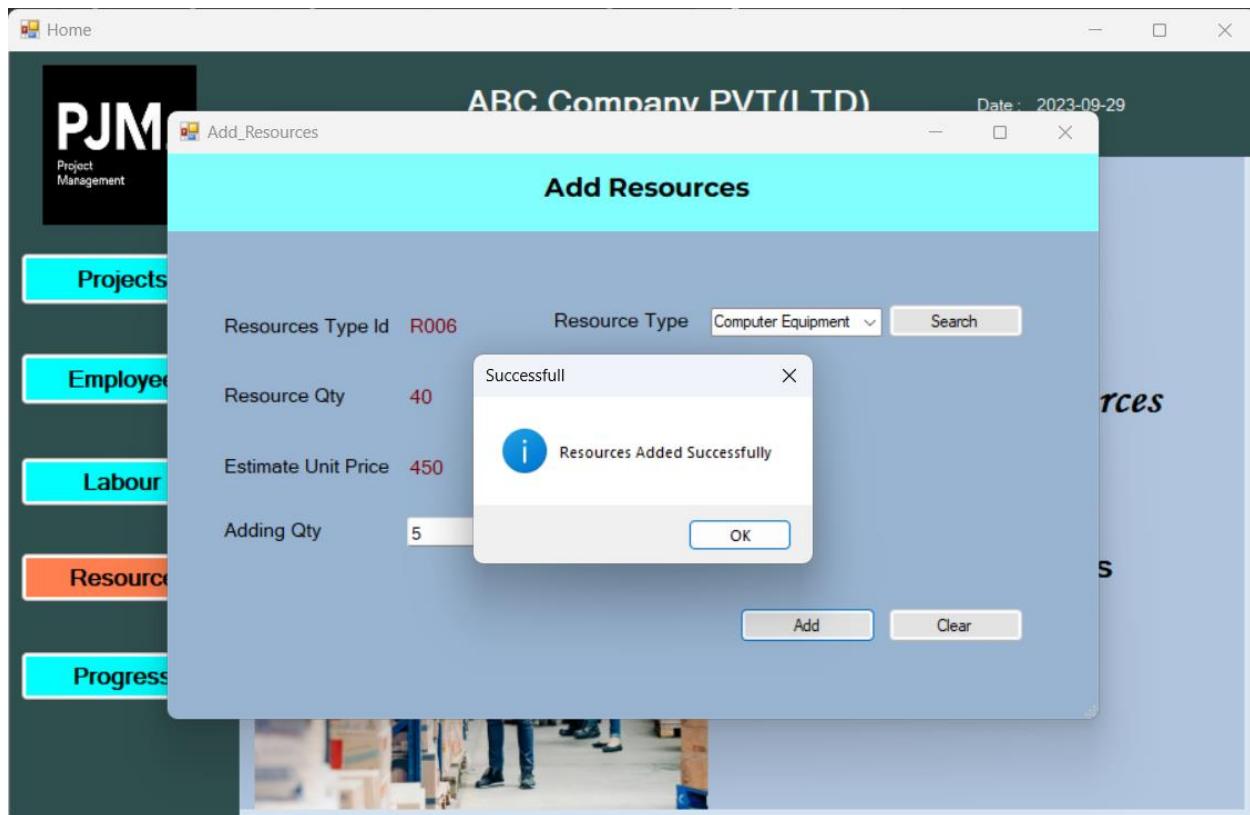
        private void Resources_Load(object sender, EventArgs e)
        {
            SqlConnection conn = new SqlConnection("Data
Source=localhost\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
            conn.Open();
            SqlCommand cmd1 = new SqlCommand("Select SUM(Qty) From Resources",
conn);
            int rsCount = (int)cmd1.ExecuteScalar();
            total.Text = rsCount.ToString();

            conn.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Resources_Details rd1 = new Resources_Details();
            rd1.ShowDialog();
        }
    }
}
=====
```

Add Resources Form

The screenshot shows a software interface for project management. On the left, there is a vertical sidebar with the logo 'PJM Project Management' and several menu items: Projects, Employees, Labour, Resources (which is highlighted in orange), and Progress. The main window title is 'ABC Company PVT(LTD)' with a date stamp 'Date : 2023-09-29'. A sub-window titled 'Add_Resources' is open in the center. The sub-window has a teal header bar with the title 'Add Resources'. Inside, there are input fields: 'Resources Type Id' with value 'R006', 'Resource Type' with a dropdown menu set to 'Computer Equipment' and a 'Search' button, 'Resource Qty' with value '40', 'Estimate Unit Price' with value '450', and 'Adding Qty' with a text input field containing '5'. At the bottom right of the sub-window are 'Add' and 'Clear' buttons.



```
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 ProjectManagementSystem
{
    public partial class Add_Resources : Form
    {
        public Add_Resources()
        {
            InitializeComponent();
        }

        private void Add_Resources_Load(object sender, EventArgs e)
        {
            if (txtId.Text == "")
            {
                textCountBox.Visible = false;
            }
            else
```

```

        {
            textCountBox.Visible = true;
        }

        SqlConnection conn = new SqlConnection("Data
Source=localhost\\SQLEXPRESS;Initial Catalog=PMSSDB;Integrated Security=True");
        conn.Open();
        SqlCommand cmd1 = new SqlCommand("Select ResourceTypeName From
Resources", conn);
        List<string> ResourceTypes = new List<string>();

        using (SqlDataReader reader1 = cmd1.ExecuteReader())
        {
            while (reader1.Read())
            {
                string ResourceType = reader1["ResourceTypeName"].ToString();
                ResourceTypes.Add(ResourceType);
            }
        }

        string[] ResourceTypeArray = ResourceTypes.ToArray();
        comboBox1.Items.AddRange(ResourceTypeArray);

        conn.Close();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection("Data
Source=localhost\\SQLEXPRESS;Initial Catalog=PMSSDB;Integrated Security=True");
        conn.Open();
        SqlCommand cmd = new SqlCommand("Select * from Resources where
ResourceTypeName=@type", conn);
        cmd.Parameters.AddWithValue("type", comboBox1.Text);
        SqlDataReader reader1;
        reader1 = cmd.ExecuteReader();
        if (reader1.Read())
        {
            txtId.Text = reader1["ResourceId"].ToString();
            txtPrice.Text = reader1["UnitPrice"].ToString();
            txtCount.Text = reader1["Qty"].ToString();
        }
        else
        {
            MessageBox.Show("Invalid Resources Check Again..", "Data Not Found",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }

        if (txtId.Text == String.Empty)
        {
            textCountBox.Visible = false;
        }
        else
        {
            textCountBox.Visible = true;
        }
    }
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection("Data
Source=localhost\\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
    conn.Open();
    SqlCommand cmd = new SqlCommand("UPDATE Resources SET Qty=@Qty WHERE
ResourceTypeId=@id", conn);
    cmd.Parameters.AddWithValue("@id", txtId.Text);
    cmd.Parameters.AddWithValue("@Qty", int.Parse(txtCount.Text) +
int.Parse(textBoxCount.Text));
    cmd.ExecuteNonQuery();

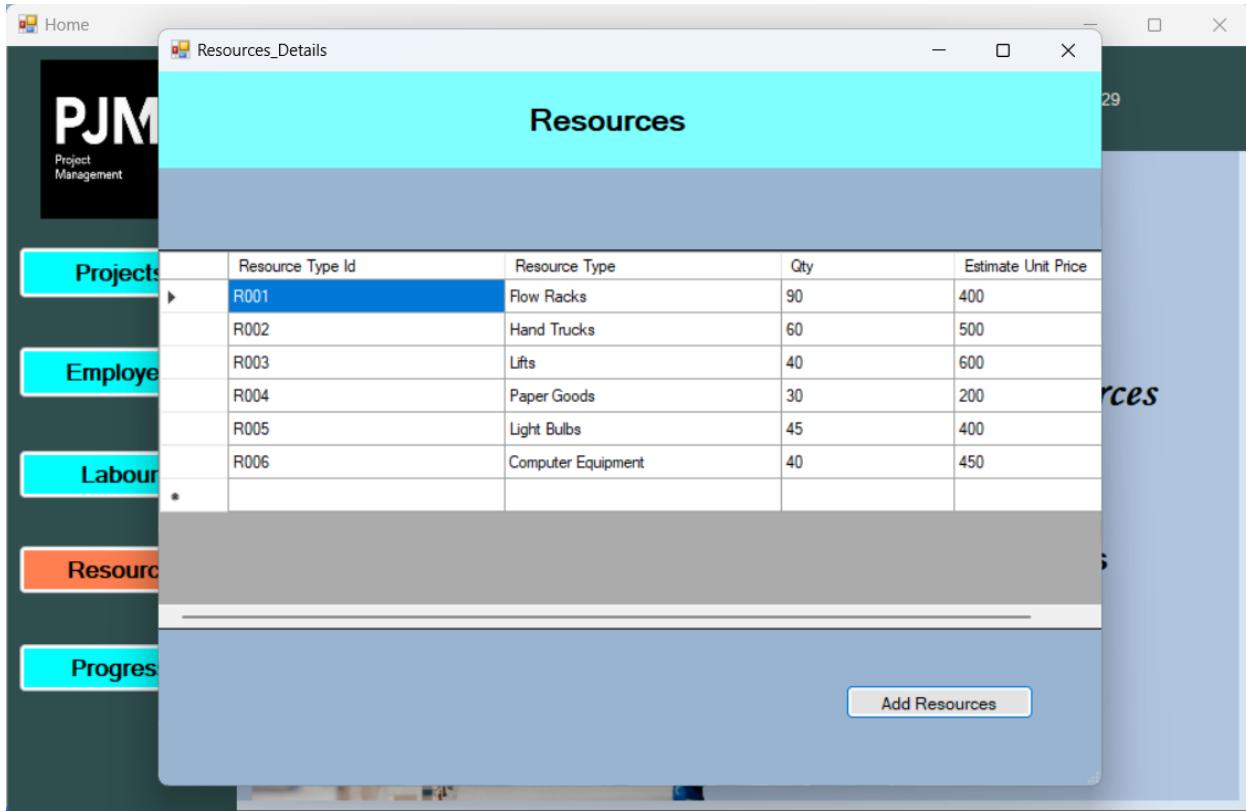
    conn.Close();

    MessageBox.Show("Labours Added Successfully", "Successfull",
MessageBoxButtons.OK, MessageBoxIcon.Information);

    txtId.Text = "";
    txtCount.Text = "";
    txtPrice.Text = "";
    comboBox1.Text = "";
    textBoxCount.Clear();
}

private void button3_Click(object sender, EventArgs e)
{
    txtId.Text = "";
    txtCount.Text = "";
    txtPrice.Text = "";
    comboBox1.Text = "";
    textBoxCount.Clear();
}
}
=====
```

Resources Details Form



```
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;

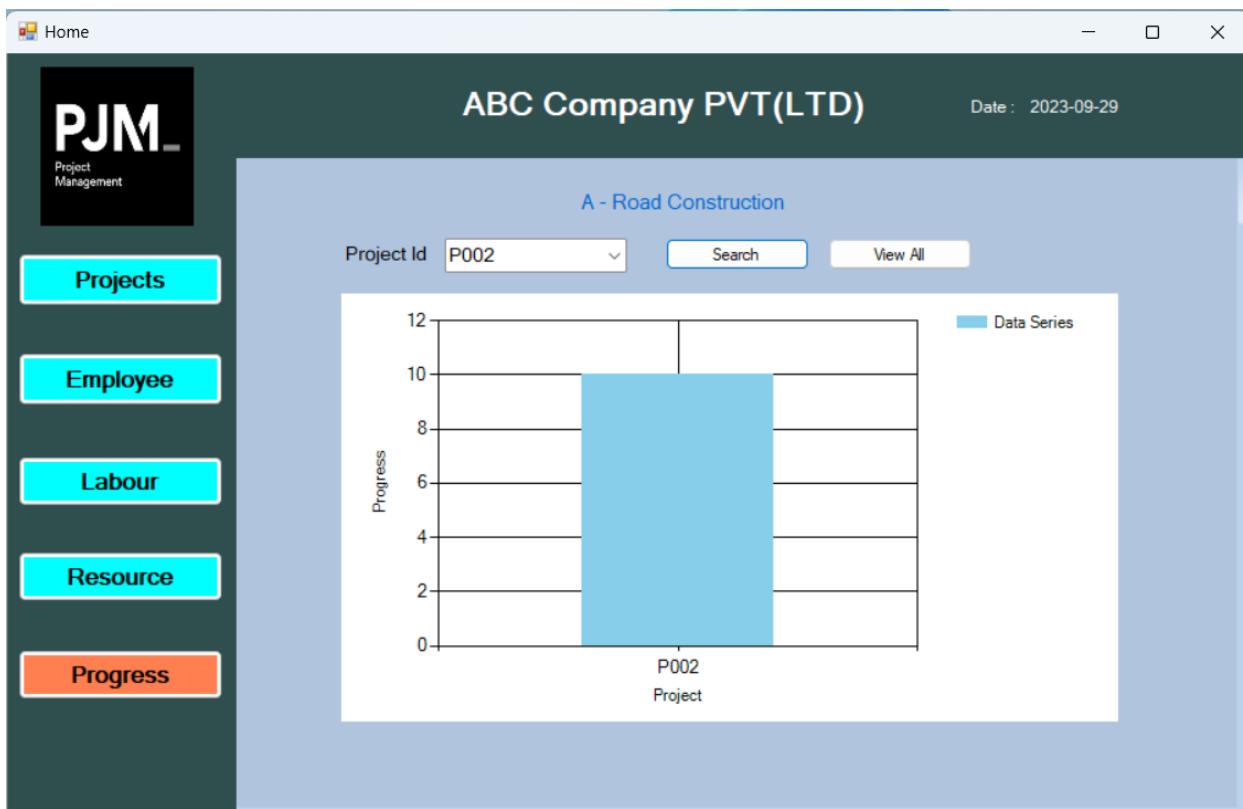
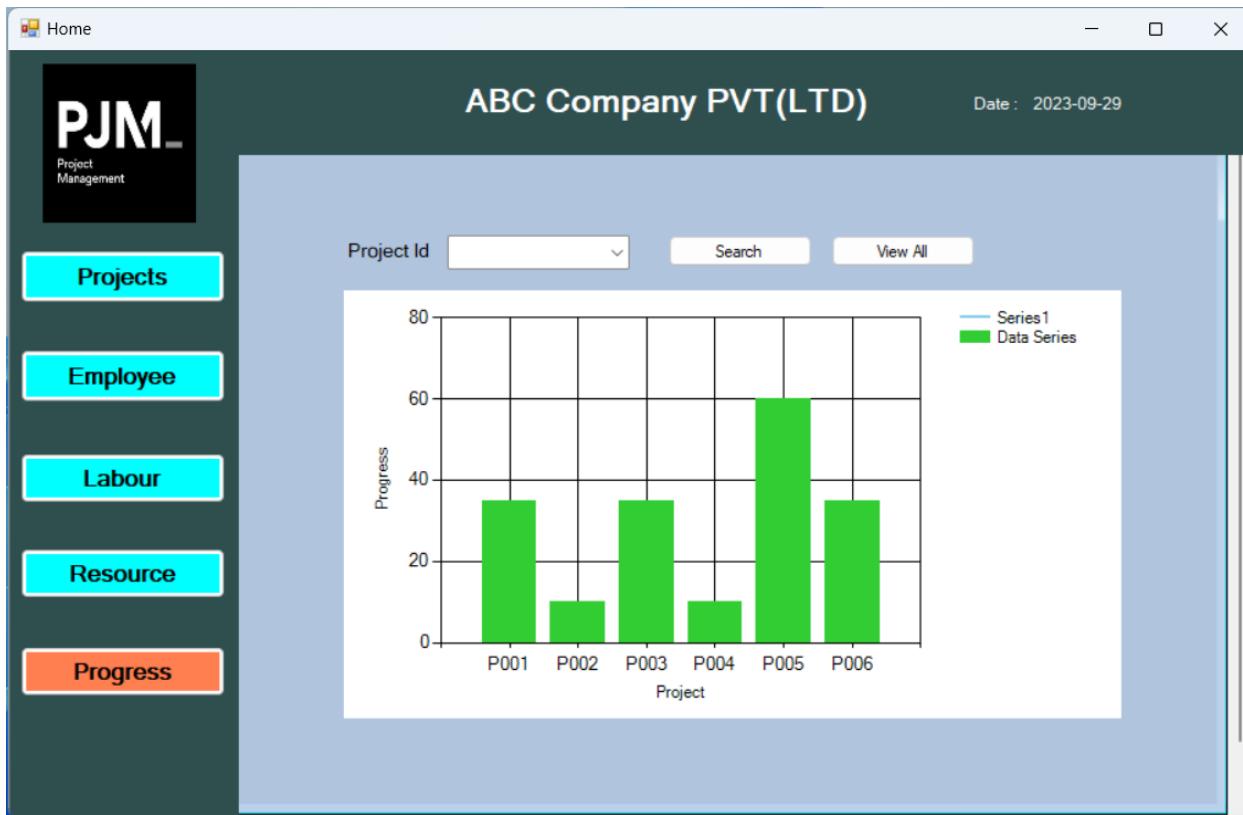
namespace ProjectManagementSystem
{
    public partial class Resources_Details : Form
    {
        public Resources_Details()
        {
            InitializeComponent();
        }

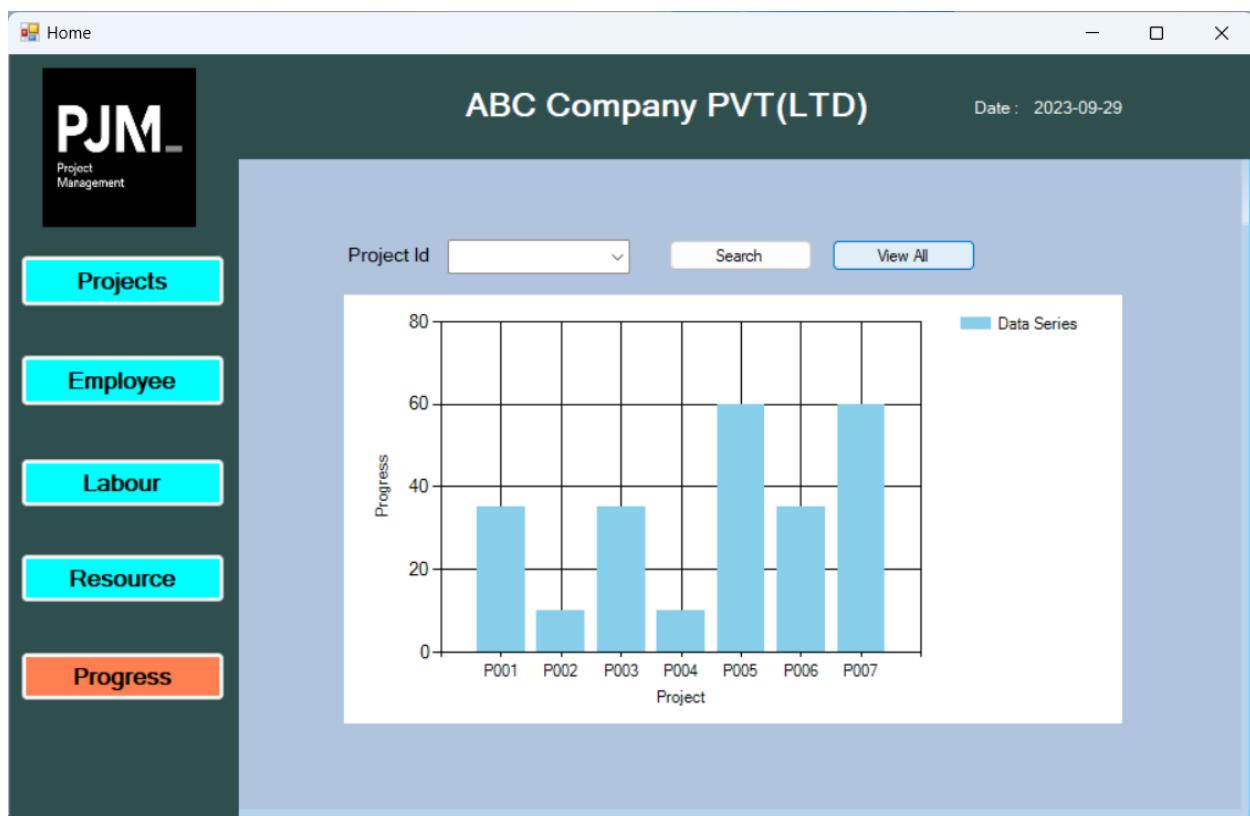
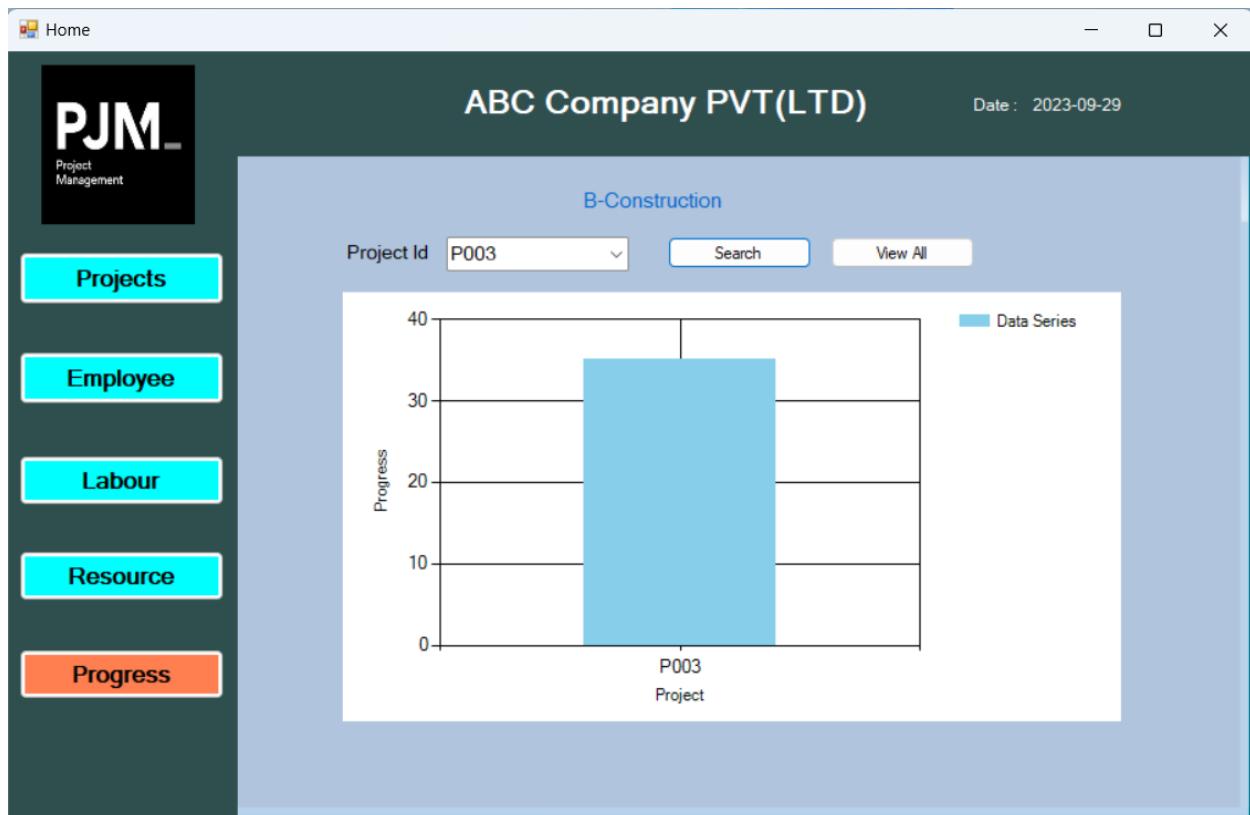
        private void button1_Click(object sender, EventArgs e)
        {
            Add_Resources ar1 = new Add_Resources();
        }
    }
}
```

```
        ar1.ShowDialog();
    }

    private void Resources_Details_Load(object sender, EventArgs e)
    {
        this.sqlDataAdapter1.Fill(this.resourceSet1.Resources);
    }
}
```

Progress Page





```

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 System.Windows.Forms.DataVisualization.Charting;
using System.Xml.Linq;
using static System.Windows.Forms.AxHost;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace ProjectManagementSystem
{
    public partial class Progress : Form
    {
        public Progress()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Projects p1 = new Projects();
            this.Hide();
            p1.ShowDialog();
            this.Close();
        }

        private void Progress_Load(object sender, EventArgs e)
        {

            // Create a series
            Series series = new Series("Data Series");
            series.ChartType = SeriesChartType.Column;
            chart1.Series.Add(series);

            // Connect to the database
            string connectionString = "Data Source=localhost\\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True";
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                connection.Open();
                SqlCommand cmd = new SqlCommand("SELECT ProjectId, Status FROM Projects", connection);

                // Read data from the database and add data points to the series
                SqlDataReader reader1 = cmd.ExecuteReader();
                while (reader1.Read())
                {
                    string Id = reader1["ProjectId"].ToString();
                    int value = 0;

                    switch (reader1["Status"].ToString())
                    {

```

```

        case "Planned":
            value = 10;
            break;
        case "Analyzed":
            value = 35;
            break;
        case "Started":
            value = 60;
            break;
        case "Completed":
            value = 80;
            break;
        case "Closed":
            value = 100;
            break;
        // Handle other statuses as needed
    }

    // Add data points based on the "Name" column value
    series.Points.AddXY(Id, value);

    // Label the X-axis directly
    chart1.ChartAreas[0].AxisX.Title = "Project";

    // Label the Y-axis directly
    chart1.ChartAreas[0].AxisY.Title = "Progress";

    // Refresh the chart to display the changes
    chart1.Refresh();
}
reader1.Close();

}

SqlConnection conn = new SqlConnection(connectionString);
conn.Open();
SqlCommand cmd1 = new SqlCommand("Select ProjectId From Projects",
conn);
List<string> ProjIds = new List<string>();

using (SqlDataReader reader1 = cmd1.ExecuteReader())
{
    while (reader1.Read())
    {
        string projId = reader1["ProjectId"].ToString(); // Assuming
        ProjectId is a string
        ProjIds.Add(projId);
    }
}

// Convert the list to an array if needed
string[] empIdsArray = ProjIds.ToArray();

// Populate a ListBox or another control with the array
comboBox1.Items.AddRange(empIdsArray);

```

```

    }

    private void button2_Click(object sender, EventArgs e)
    {
        Series series = new Series("Data Series");
        series.ChartType = SeriesChartType.Column;
        chart1.Series.Clear();
        chart1.Series.Add(series);

        SqlConnection conn = new SqlConnection("Data
Source=localhost\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True");
        conn.Open();
        SqlCommand cmd = new SqlCommand("Select ProjectId,Name,Status from
Projects where ProjectId=@id", conn);
        cmd.Parameters.AddWithValue("@id", comboBox1.Text);

        SqlDataReader reader1 = cmd.ExecuteReader();

        while (reader1.Read())
        {
            string Id = reader1["ProjectId"].ToString();
            ProjName.Text = reader1["Name"].ToString();

            int value = 0;

            switch (reader1["Status"].ToString())
            {
                case "Planned":
                    value = 10;
                    break;
                case "Analyzed":
                    value = 35;
                    break;
                case "Started":
                    value = 60;
                    break;
                case "Completed":
                    value = 80;
                    break;
                case "Closed":
                    value = 100;
                    break;
                // Handle other statuses as needed
            }

            // Add data points based on the "Name" column value
            series.Points.AddXY(Id, value);

            // Label the X-axis directly
            chart1.ChartAreas[0].AxisX.Title = "Project";

            // Label the Y-axis directly
            chart1.ChartAreas[0].AxisY.Title = "Progress";

            // Refresh the chart to display the changes
            chart1.Refresh();
        }
    }

```

```

        reader1.Close();
    }

    private void button1_Click_1(object sender, EventArgs e)
    {
        comboBox1.Text = "";
        ProjName.Text = " ";
        chart1.Series.Clear();
        Series series = new Series("Data Series");
        series.ChartType = SeriesChartType.Column;
        chart1.Series.Add(series);

        // Connect to the database
        string connectionString = "Data Source=localhost\\SQLEXPRESS;Initial Catalog=PMSDB;Integrated Security=True";
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();
            SqlCommand cmd = new SqlCommand("SELECT ProjectId, Status FROM Projects", connection);

            // Read data from the database and add data points to the series
            SqlDataReader reader1 = cmd.ExecuteReader();
            while (reader1.Read())
            {
                string Id = reader1["ProjectId"].ToString();
                int value = 0;

                switch (reader1["Status"].ToString())
                {
                    case "Planned":
                        value = 10;
                        break;
                    case "Analyzed":
                        value = 35;
                        break;
                    case "Started":
                        value = 60;
                        break;
                    case "Completed":
                        value = 80;
                        break;
                    case "Closed":
                        value = 100;
                        break;
                    // Handle other statuses as needed
                }

                // Add data points based on the "Name" column value
                series.Points.AddXY(Id, value);

                // Label the X-axis directly
                chart1.ChartAreas[0].AxisX.Title = "Project";

                // Label the Y-axis directly
                chart1.ChartAreas[0].AxisY.Title = "Progress";

                // Refresh the chart to display the changes
            }
        }
    }
}

```

```
        chart1.Refresh();
    }
    reader1.Close();

}
}

}

===== Screenshots & Code implementation end ... =====
```

Demo video of our project



Click me

<https://drive.google.com/file/d/1Bty0FCPX91jddkMx1wVGunjNbm2dikQo/view?usp=sharing>

Tools and Things used for design

07



- **Programming Language - C#**

The programming language used is **C#**, which is commonly used for Windows Forms applications and integrates with various libraries and frameworks.

- **Development Environment**

The project is likely developed using Visual Studio, a popular integrated development environment (**IDE**) for C# and **.NET** development. Visual Studio provides a user-friendly interface for building Windows Forms applications and offers debugging and design tools.

- **Windows Forms**

The user interface (**UI**) for the project is built using **Windows Forms**. Windows Forms is a graphical user interface framework provided by **Microsoft** for developing desktop applications in C#.

- **Canva**

Canva is a user-friendly online graphic design tool that helps people create a wide range of visual content easily, even without prior design experience. It is used to create **project reports**.

- **SQL Server Database**

The project connects to a **SQL Server database** using **System.Data.SqlClient**. This is evident in the code where SQL Server connections and commands are used for **data storage and retrieval**.

- **Database**

The project appears to interact with a database to store and retrieve information related to projects, employees, labor, resources, and progress. The exact **database management system (DBMS)** is **SQL Server**.

- **Draw.io**

Draw.io is a web-based diagramming tool that allows users to create **diagrams** and **visual representations** of ideas and concepts.



Initial plan vs Actual plan

Initial plan

In the initial plan for the project management system for building construction, that a user could log in and get details about the project.

Further primary objective was to create a functional system with a focus on simplicity.

Initially, there was an intention to have just four forms for each of the main user categories, dashboard, employee, labor, and project.

Actual plan

In the initial plan for the project management system for building construction, a user could log in and get details about the project.

Further primary objective was to create a functional system with a focus on simplicity.

Initially, there was an intention to have just four forms for each of the main user categories, dashboard, employee, labor, and project.

And not limited to four forms, it is made in a wide range of forms. This application allows you to add a project, employee, or resource and get their details

Furthermore, can see the progress of all projects, and get a correct understanding of them, by the graph giving details.

08

WORK CONTRIBUTION & CHALENGERS

Work Contribution

NAME

C.P.K.Wijeratne

H.G.H. Eshan

N.P.K.D.R.S.Gunarathna

D.H.G.Lahiru

J.M.N.C.Jayamanne

CONTRIBUTION

Dashboard & Employee

Progress & Login

Labor

Project

Resource



WORK CONTRIBUTION & CHALLENGERS

09

Challenges

- **Time Management:**

Balancing individual tasks with team responsibilities was challenging. Meeting deadlines and managing time efficiently is critical.

- **Online Collaboration:**

Adapting to and effectively using online collaboration tools can be a learning curve for some group members.

- **Communication Gaps:**

Misunderstandings and lack of clarity can occur due to differing schedules, leading to communication gaps.

- **Motivation and Accountability:**

Keeping all group members motivated and accountable, especially when working remotely, poses a challenge.

- **Conflict Resolution:**

Addressing disagreements or conflicts within the group requires effective resolution strategies.

Challenges

09

- **Workload Management:**

Balancing project work with individual commitments and responsibilities can lead to stress and time management challenges.

- **Flexibility:**

Adapting to changing schedules and unforeseen circumstances requires flexibility and adaptability.

- **Progress Tracking:**

Tracking project progress, especially when group members work at different times, can be complex.

- **Scheduling Conflicts:**

Coordinating the schedules of four group members is challenging, making it difficult to find suitable meeting times.

- **Task Management:**

Handling multiple tasks requires clear responsibilities and deadlines to prevent overlaps or omissions.

- **Coordination issues:**

Ensuring alignment between the contributions of different members can be tricky.



FUTURE ENHANCEMENTS

.....

- **Artificial Intelligence (AI) and Machine Learning:**

Integration of AI and machine learning algorithms for predictive analytics, task automation, and intelligent decision support. AI can assist in resource allocation, risk assessment, and project forecasting.

- **Advanced Reporting and Analytics:**

More sophisticated reporting and analytics capabilities, including real-time dashboards, customizable KPIs, and data visualization tools for better project insights and decision-making.

FUTURE ENHANCEMENTS

10

- **Blockchain Integration:**

Blockchain technology for enhanced security, transparency, and traceability in project management, particularly in industries like supply chain management and finance.

- **IoT Integration:**

Integration with the Internet of Things (IoT) devices to collect real-time data from project equipment and assets, enabling proactive maintenance and monitoring.

- **Automation of Routine Tasks:**

Automation of repetitive tasks such as status updates, document approvals, and task assignments to free up project managers for more strategic work.

- **Enhanced Mobile Experience:**

Mobile apps with improved functionality and offline capabilities to allow team members to work on projects from anywhere.

FUTURE ENHANCEMENTS

10

- **AI-driven Risk Management:**

AI-powered risk assessment and mitigation tools that can predict and proactively address potential project risks.

- **Security and Privacy:**

Enhanced security measures to protect sensitive project data and compliance with evolving data privacy regulations.

- **Resource Optimization:**

Advanced resource management features for optimizing the allocation of human and material resources based on skill sets and availability.

11

Our Team



Chameera Peshan

PS / 2019 / 265

wijerat-ps19265@stu.kln.ac.lk

+94 76 007 8842



Harshana Eshan

PS / 2019 / 137

eshanhg-ps19137@stu.kln.ac.lk

+94 71 604 8560

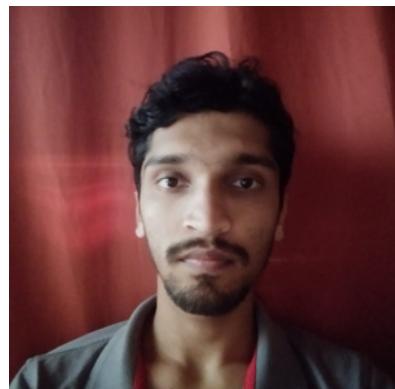


Randila Shehari

PS / 2019 / 225

gunarat-ps19225@stu.kln.ac.lk

+94 71 696 4666



D.H.G.Lahiru

PS / 2019 / 022

lahirud-ps19022@stu.kln.ac.lk

+94 76 927 9527



J.M.N.C.Jayamanne

PS / 2019 / 244

jayaman-ps19244@stu.kln.ac.lk

+94 71 568 6673



