

INDUSTRIAL PRACTICAL TRAINING
PROJECT REPORT

INDEX

S No.	HEADING	PAGE
1.1	Description of the project	3
1.2	Objective of the project	3
1.3	Software used	4
1.4	Application Setup	8
1.5	Working	19

1.1. Description of the project

The project assigned to me was to create a Work Management System. I created it on Windows Forms App on Visual Studio (2019). The back-end coding was done in C# .Net language. The database is stored separately in SQL Server Management and is connected to the code in visual studio.

1.2. Objective of the project

“It is our attitude at the beginning of a difficult task which, more than anything else, will affect its successful outcome.”
William James

Work management is more than just a task manager checking off items on a to-do list! It's an organized system for identifying, monitoring, and managing your work. Work management involves:

- Tracking task progress
- Setting deadlines
- Adjusting work schedules

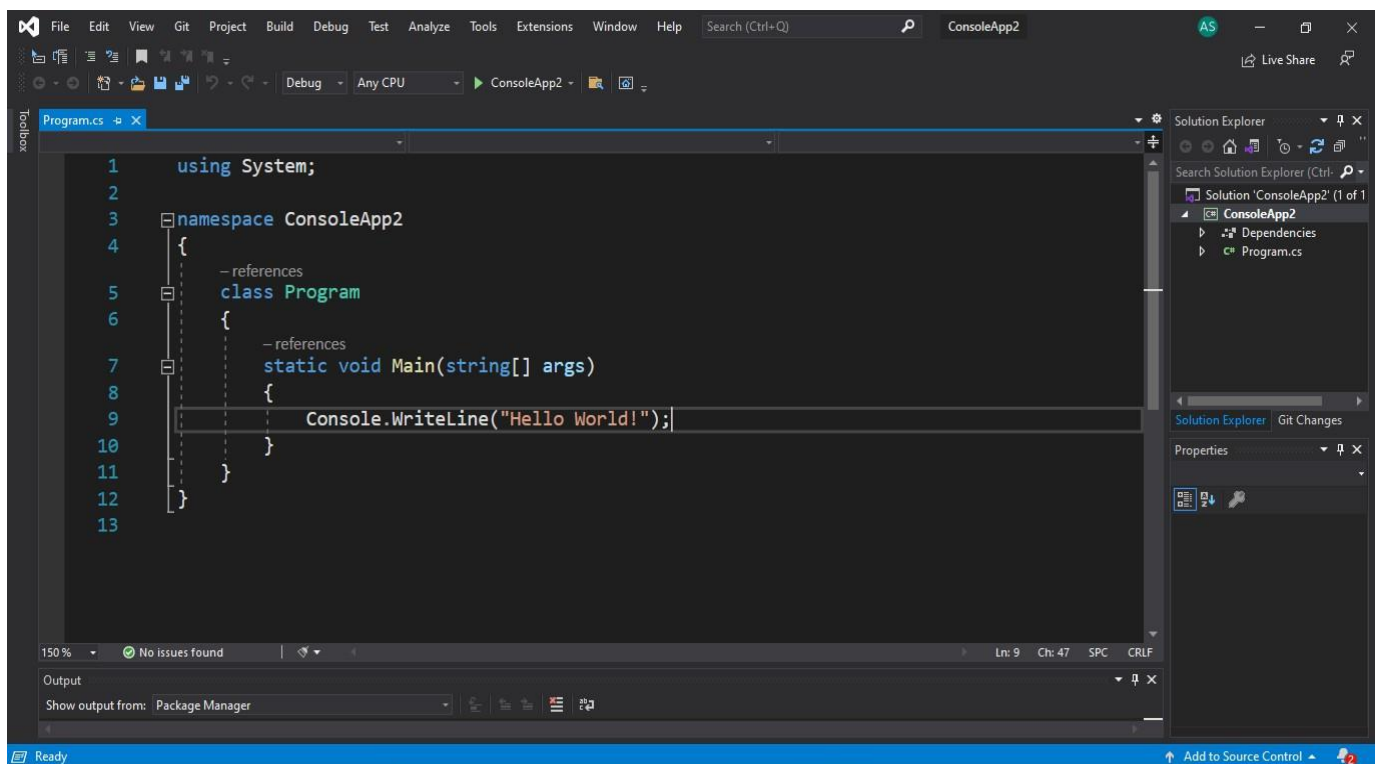
This Work Management system helps organize our everyday tasks and keeps track of project deadlines.

1.3. Software

1.3.1. Visual Studio

Visual Studio is an Integrated Development Environment (IDE) developed by Microsoft to develop GUI (Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code.

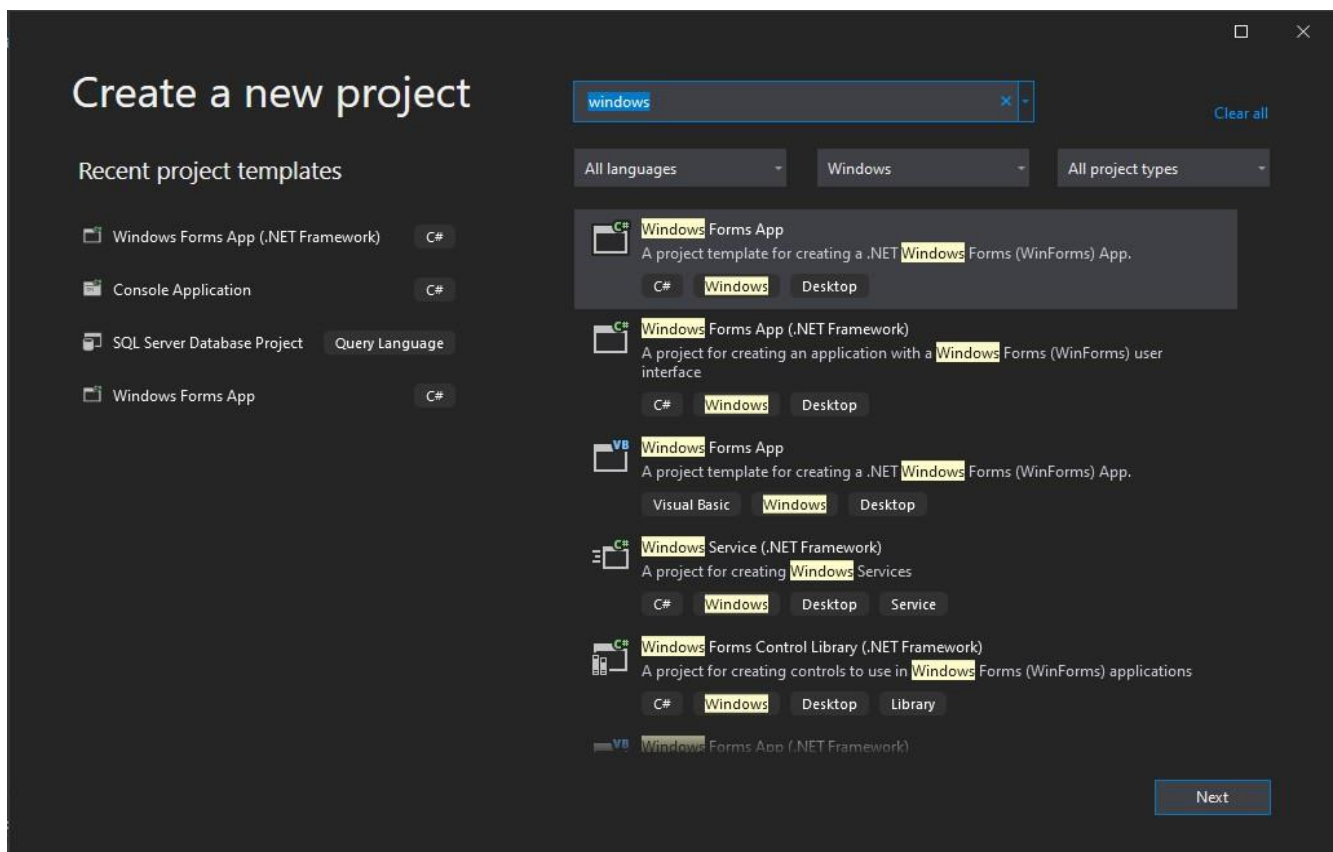
It is not a language-specific IDE as you can use this to write code in C#, C++, VB (Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages.

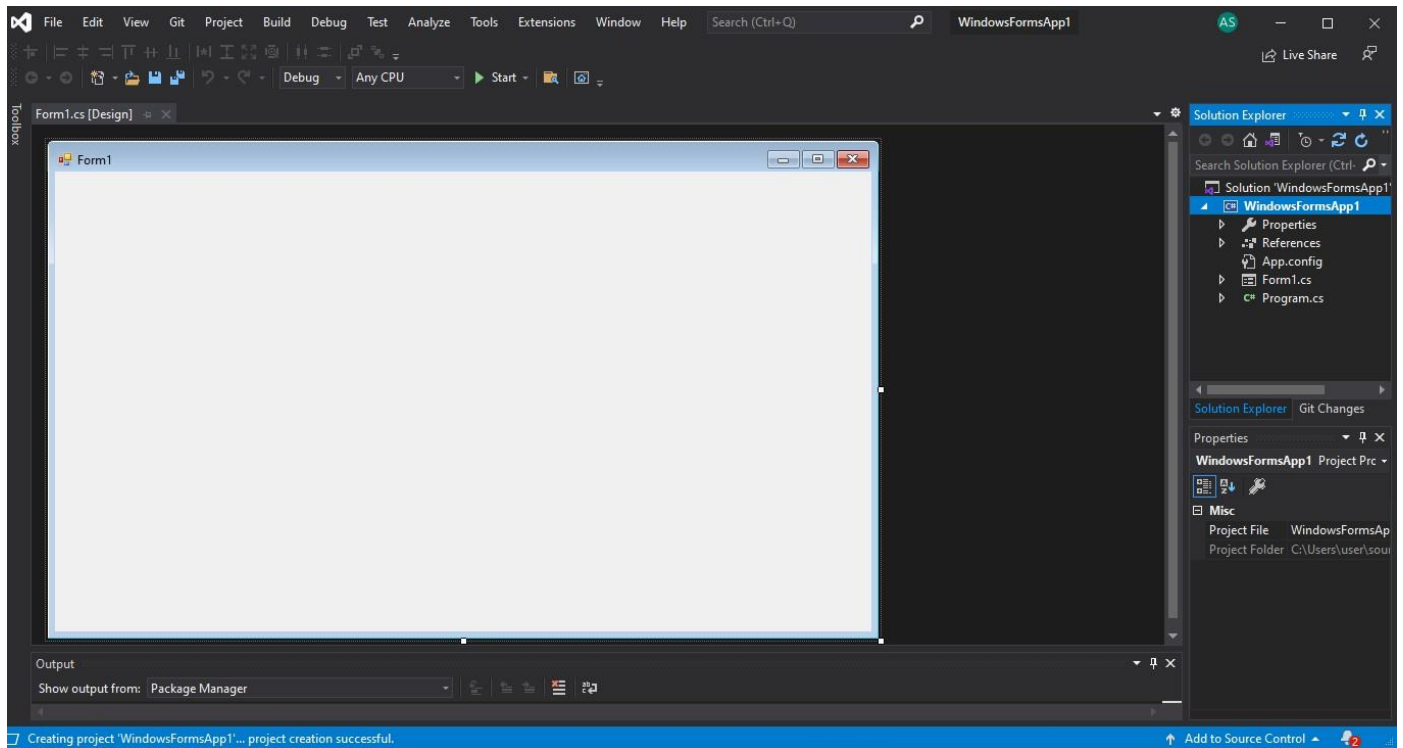


1.3.2. Introduction to C# Windows Forms Applications

Windows Forms is a Graphical User Interface (GUI) class library which is bundled in .Net Framework. Its main purpose is to provide an easier interface to develop the applications for desktop, tablet, PCs. It is also termed as the WinForms.

The applications which are developed by using Windows Forms or WinForms are known as the Windows Forms Applications that runs on the desktop computer. WinForms can be used only to develop the Windows Forms Applications not web applications. WinForms applications can contain the different type of controls like labels, list boxes, tooltip etc.





1.3.3. SQL Server Management Studio

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure. Use SSMS to access, configure, manage, administer, and develop all components of SQL Server, Azure SQL Database, and Azure Synapse Analytics. SSMS provides a single comprehensive utility that combines a broad group of graphical tools with a number of rich script editors to provide access to SQL Server for developers and database administrators of all skill levels.

Connect to Server

SQL Server

Server type: Database Engine

Server name: DESKTOP-NGS8Q66

Authentication: Windows Authentication

User name: DESKTOP-NGS8Q66\user

Password:

☐ Remember password

Connect Cancel Help Options >>

SQLQuery1.sql - DESKTOP-NGS8Q66.TaskManager (DESKTOP-NGS8Q66\user (56)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

TaskManager

Object Explorer

Connect

DESKTOP-NGS8Q66 (SQL Server 15.0)

Databases

System Databases

Database Snapshots

TaskManager

Database Diagrams

Tables

System Tables

FileTables

External Tables

Graph Tables

dbo.customerDB

dbo.techs

dbo.users

Columns

Keys

Constraints

Triggers

Indexes

Statistics

Views

External Resources

Synonyms

Programmability

Service Broker

Storage

Security

Server Objects

SQLQuery1.sql - DE...NGS8Q66\user (56)

Script for SelectTopNRows command from SSMS

```
SELECT TOP (1000) [userID]  
    , [username]  
    , [password]  
FROM [TaskManager].[dbo].[users]
```

Results

	userID	username	password
1	1	Aish2001	aish2001
2	2	user	visualstudio

Query executed successfully.

DESKTOP-NGS8Q66 (15.0 RTM) | DESKTOP-NGS8Q66\user (56) | TaskManager | 00:00:00 | 2 rows

1.4. Application Setup

This task manager system was created in Windows Forms App in Visual studio. It has two screens:

- a login screen with database level authentication
- once login is successful , user will be taken to the next screen which is a dashboard which displays the scheduled tasks and completed tasks

1.4.1 Login



This is the login screen where users will have to enter their Username and Password. Once the password is correct and verified, users will be directed to the dashboard with the contents.

LOGIN

x

TASK MANAGEMENT SYSTEM

LOGIN

User ID : Aish2001

Password : *****

SUBMIT

BUSINESS

25 great jobs for people who love to travel

Economy of the European Union

LOGIN

TASK MANAGEMENT SYSTEM

LOGIN

User ID : Aish2001

Password : *****

SUBMIT

BUSINESS

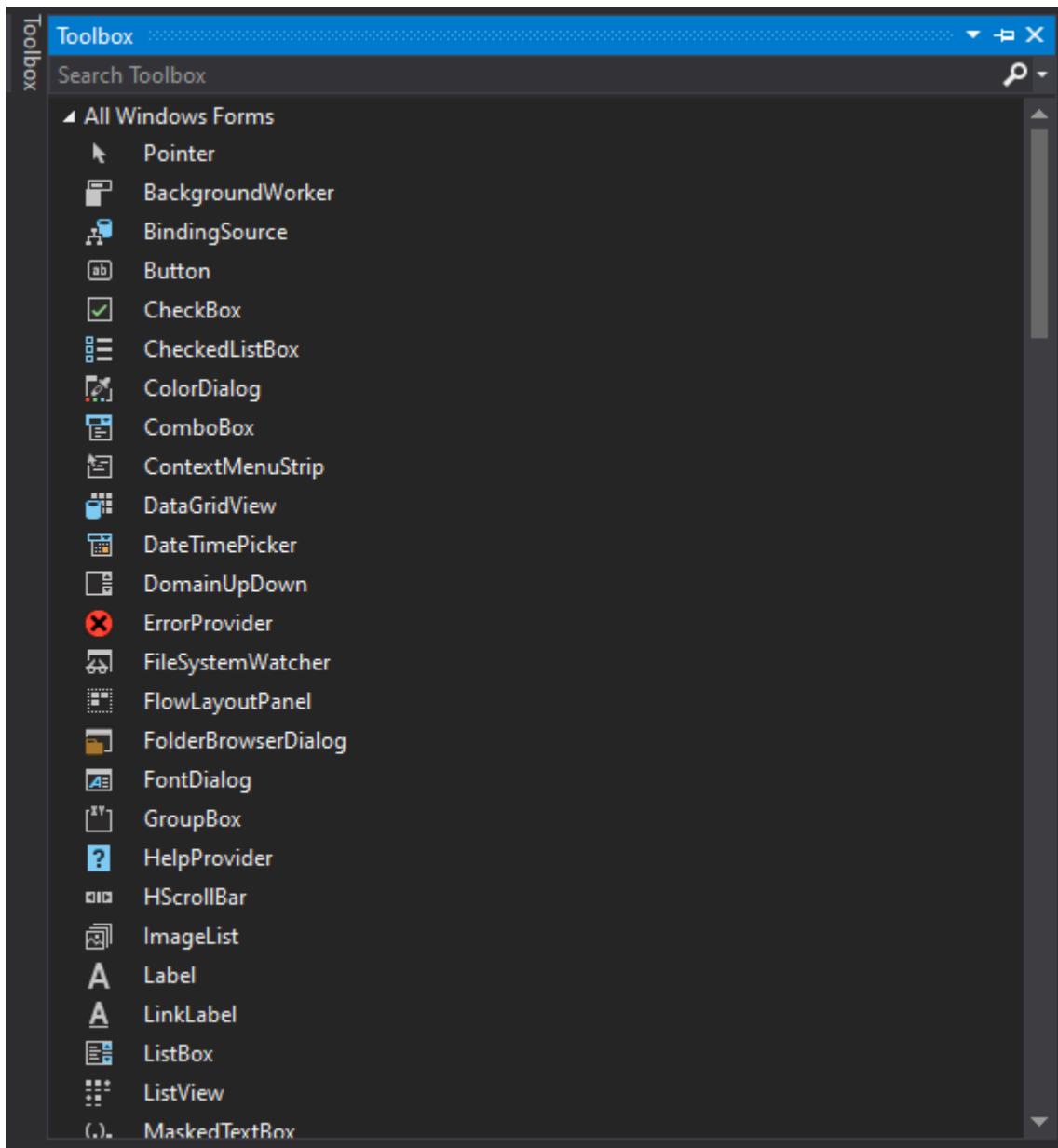
25 great jobs for people who love to travel

Economy of the European Union

Hey Aish2001! Welcome to Task Manager

OK

It's front end was designed with help of the toolbox available in Windows forms App.



The desired tools were selected and their properties were altered to create the design of the Login page.

The backend was coded in C# language.

Code :

```
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.SqlClient;

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

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

        private void btn_Submit(object sender, EventArgs e)
        {
            SqlConnection sqlcon = new SqlConnection(@"Data Source=DESKTOP-
NGS8Q66;Initial Catalog=TaskManager;Integrated Security=True");
            String query = "select* from users where username = '" +
comboBox1.Text.Trim() + "' and password = '" + textBox1.Text.Trim() + "'";
            SqlDataAdapter sda = new SqlDataAdapter(query, sqlcon);
            DataTable dtbl = new DataTable();
            sda.Fill(dtbl);
            if(dtbl.Rows.Count == 1)
            {
                MessageBox.Show("Hey " + comboBox1.Text + "! Welcome to Task
Manager");
                Main objMain = new Main();
                this.Hide();
                objMain.Show();
            }
            else
        }
    }
}
```

```

        {
            MessageBox.Show("Incorrect Username or Password Entered!");
        }
    }

    private void Login_Load_1(object sender, EventArgs e) { }
}

```

1.4.2. Dashboard

Dashboard shows the scheduled tasks (as live tickets) and the completed tasks (as closed tickets).

The front end was designed in the same way using the labels and the backend code is written in C#.

The screenshot shows a web application interface with a blue header and sidebar. The main content area has three tabs: 'TARGETS', 'CLOSED TARGETS', and 'CUSTOMER DATABASE'. The 'LIVE TICKETS' section is highlighted. It contains a large, empty rectangular box for displaying tickets. To the right of this box is a 'DATE DUE' section with a calendar for July 2021. The calendar shows the 11th as the selected date. Below the calendar are several input fields labeled 'ID:', 'CLIENT:', 'POST CODE:', 'ADDRESS:', 'PHONE:', 'TECHNICIAN:', and 'DESCRIPTION:'. At the bottom of the dashboard, there are two red buttons labeled 'DELETE TASK' and 'CLOSE', an 'ESTIMATE:' label with an input field, and a green 'SAVE' button.

Dashboard

TARGETSCLOSED TARGETSCUSTOMER DATABASE

CLOSED TICKETS

LOG OUT

TOTAL INVOICED:



DELETE

Dashboard also gives access to the customer database showing a user all their clients.

Dashboard

TARGETSCLOSED TARGETSCUSTOMER DATABASE

CUSTOMER DATABASE

LOG OUT

ADD A NEW CUSTOMER:

ID:

CLIENT:

POST CODE:

ADDRESS:

PHONE:

EMAIL:

SAVE

Code:

```
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.SqlClient;

namespace TASK_MANAGER
{
    public partial class Main : Form
    {
        SqlConnection sh = new SqlConnection(@"C:\Users\user\source\repos\TASK
MANAGER\SQLHelper.cs");
        string path1 = @"C:\Users\user\source\repos\TASK
MANAGER\App1.config";
        BindingSource bs = new BindingSource();

        SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-
NGS8Q66;Initial Catalog=TaskManager;Integrated Security=True");
        public Main()
        {
            InitializeComponent();
        }

        private void btnSaveClient_Click(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand cmd = con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "insert into customerDB(ID, client, post code,
address, phone, email) Values(@ID:, @CLIENT:, @POST CODE:, @ADDRESS:, @PHONE:,
@EMAIL:)";
            cmd.Parameters.AddWithValue("@ID:", textBox14.Text);
            cmd.Parameters.AddWithValue("@CLIENT:", textBox15.Text);
            cmd.Parameters.AddWithValue("@POST CODE:", textBox13.Text);
            cmd.Parameters.AddWithValue("@ADDRESS:", textBox12.Text);
            cmd.Parameters.AddWithValue("@PHONE:", textBox10.Text);
            cmd.Parameters.AddWithValue("@EMAIL:", textBox11.Text);

            cmd.ExecuteNonQuery();
        }
    }
}
```

```

        con.Close();
        disp_data();

        MessageBox.Show("Record Updated");

        textBox14.Text = "";
        textBox15.Text = "";
        textBox13.Text = "";
        textBox12.Text = "";
        textBox10.Text = "";
        textBox11.Text = "";

    }

    public void disp_data()
    {
        con.Open();
        SqlCommand cmd = con.CreateCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "select * from customerDB";
        System.Data.DataSet ds = new System.Data.DataSet();
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(ds);
        dataGridView3.DataSource = ds;

        con.Close();
    }

    private void Main_Load(object sender, EventArgs e)
    {
        disp_data();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        dataGridView1.Rows.Add(textBox8.Text, textBox7.Text, textBox1.Text,
        textBox3.Text, textBox5.Text, textBox6.Text, textBox4.Text, textBox2.Text,
        monthCalendar1.SelectionRange.Start.ToShortDateString());
        textBox8.Text = "";
        textBox7.Text = "";
        textBox1.Text = "";
        textBox3.Text = "";
        textBox5.Text = "";
        textBox6.Text = "";
        textBox4.Text = "";
        textBox2.Text = "";
    }

```

```

    }

    private void button3_Click(object sender, EventArgs e)
    {
        foreach (DataGridViewRow selRow in
dataGridView1.SelectedRows.OfType<DataGridViewRow>().ToArray())
        {
            dataGridView1.Rows.Remove(selRow);
            dataGridView2.Rows.Add(selRow);

            textBox9.Text = (from DataGridViewRow rows in
dataGridView2.Rows
                                where rows.Cells[7].FormattedValue.ToString()
!= string.Empty
                                select
Convert.ToDouble(rows.Cells[7].FormattedValue)).Sum().ToString();

        }

    }

    private void button1_Click(object sender, EventArgs e)
    {
        foreach (DataGridViewRow selRow in
dataGridView1.SelectedRows.OfType<DataGridViewRow>().ToArray())
        {
            dataGridView1.Rows.Remove(selRow);
        }

    }

    private void button5_Click(object sender, EventArgs e)
    {
        Microsoft.Office.Interop.Excel._Application app = new
Microsoft.Office.Interop.Excel.Application();
        Microsoft.Office.Interop.Excel._Workbook workbook =
app.Workbooks.Add(Type.Missing);
        Microsoft.Office.Interop.Excel._Worksheet worksheet = null;
        app.Visible = true;
        worksheet = workbook.Sheets["Sheet1"];
        worksheet = workbook.ActiveSheet;
        worksheet.Name = "Exported from Task Manager";

        for(int i = 1; i<dataGridView2.Columns.Count + 1; i++)
        {

```



```

        worksheet.Cells[1, i] = dataGridView2.Columns[i -
1].HeaderText;
    }
    for(int i=0; i < dataGridView2.Rows.Count - 1; i++)
    {
        for(int j=0; j < dataGridView2.Columns.Count; j++)
        {
            worksheet.Cells[i + 2, j + 1] =
dataGridView2.Rows[i].Cells[j].Value.ToString();
        }
    }

    worksheet.SaveAs("c:\\CLOSEDTARGETS.xls",
        Type.Missing,
        Type.Missing,
        Type.Missing,
        Type.Missing,
        Type.Missing,

Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlExclusive,
        Type.Missing,
        Type.Missing,
        Type.Missing
    );
}
private void button4_Click(object sender, EventArgs e)
{
    foreach (DataGridViewRow selRow in
dataGridView2.SelectedRows.OfType<DataGridViewRow>().ToArray())
    {
        dataGridView2.Rows.Remove(selRow);

        textBox9.Text = (from DataGridViewRow rows in
dataGridView2.Rows
                        where rows.Cells[7].FormattedValue.ToString()
!= string.Empty
                        select
Convert.ToDouble(rows.Cells[7].FormattedValue)).Sum().ToString();

    }
}

private void button8_Click(object sender, EventArgs e)
{
    this.Hide();
    Login loginPg = new Login();

```

```

        loginPg.Show();
    }

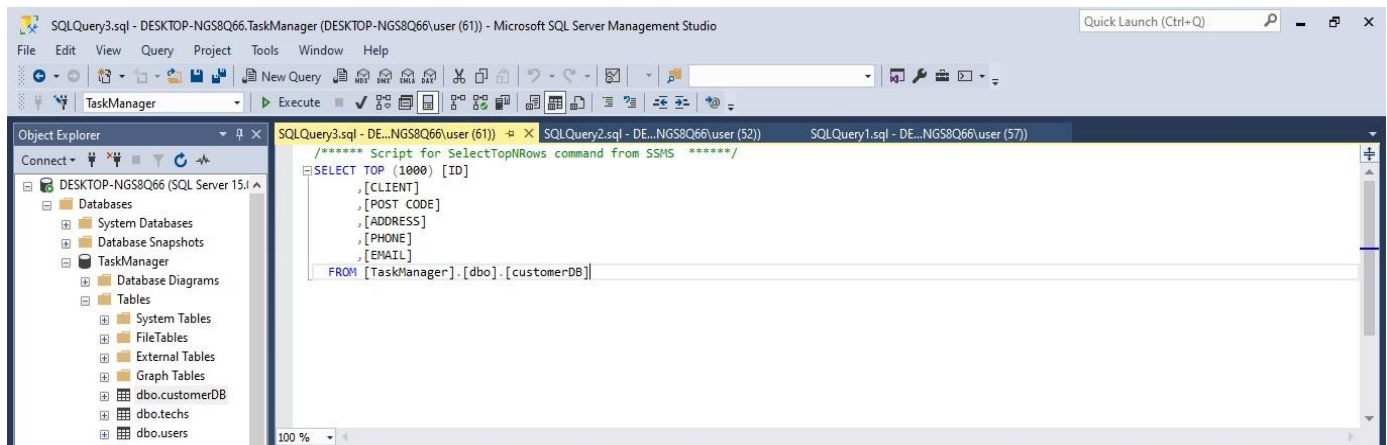
    private void button7_Click(object sender, EventArgs e)
    {
        this.Hide();
        Login loginPg = new Login();
        loginPg.Show();
    }

    private void button6_Click(object sender, EventArgs e)
    {
        this.Hide();
        Login loginPg = new Login();
        loginPg.Show();
    }
}
}

```

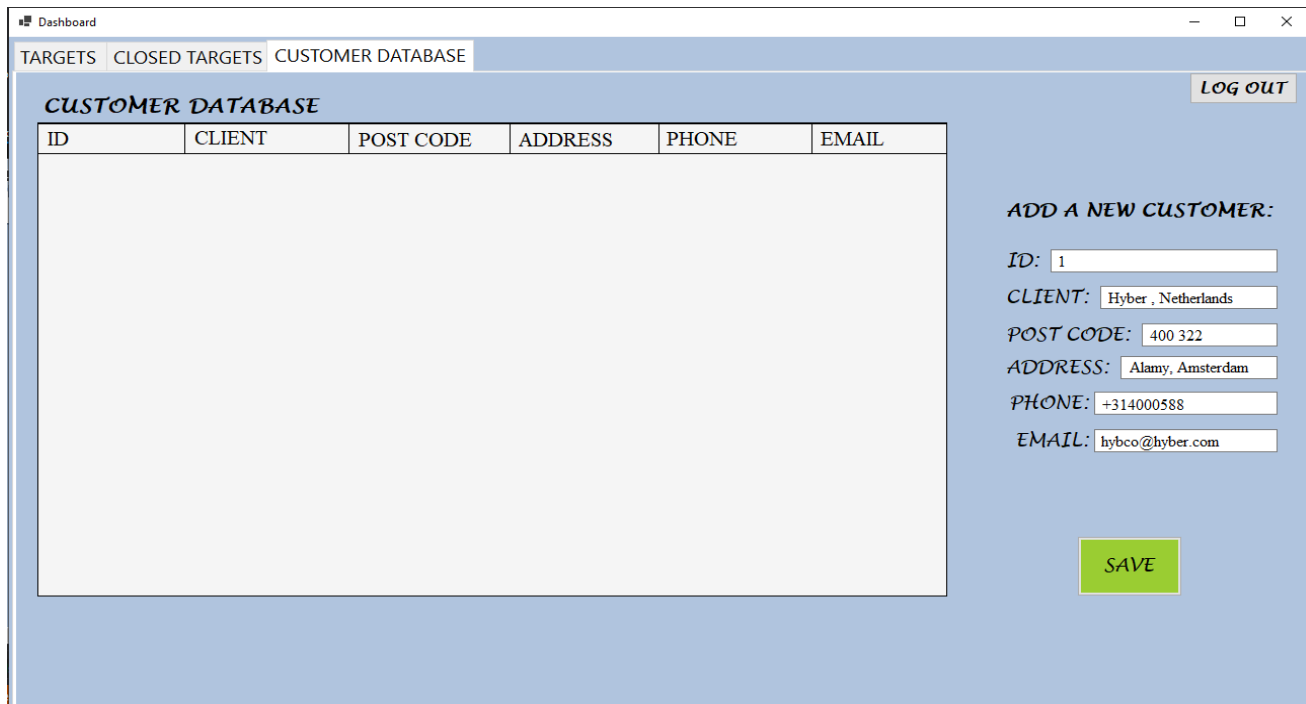
1.4.3. SQL Database

All tasks entered by the user is stored in a database in SQL Server. The details are arranged in the form of a table under various columns.



1.5. Working

The user enters Client details in the customer database.



The screenshot shows a web application window titled 'Dashboard' with three tabs: 'TARGETS', 'CLOSED TARGETS', and 'CUSTOMER DATABASE'. The 'CUSTOMER DATABASE' tab is active. On the right, there is a 'LOG OUT' button. The main area is titled 'CUSTOMER DATABASE' and contains a table with six columns: ID, CLIENT, POST CODE, ADDRESS, PHONE, and EMAIL. The table is currently empty. To the right of the table is a form titled 'ADD A NEW CUSTOMER:' with input fields for ID (1), CLIENT (Hyber , Netherlands), POST CODE (400 322), ADDRESS (Alamy, Amsterdam), PHONE (+314000588), and EMAIL (hybco@hyber.com). A green 'SAVE' button is located below the form.

ID	CLIENT	POST CODE	ADDRESS	PHONE	EMAIL
----	--------	-----------	---------	-------	-------

ADD A NEW CUSTOMER:

ID:

CLIENT:

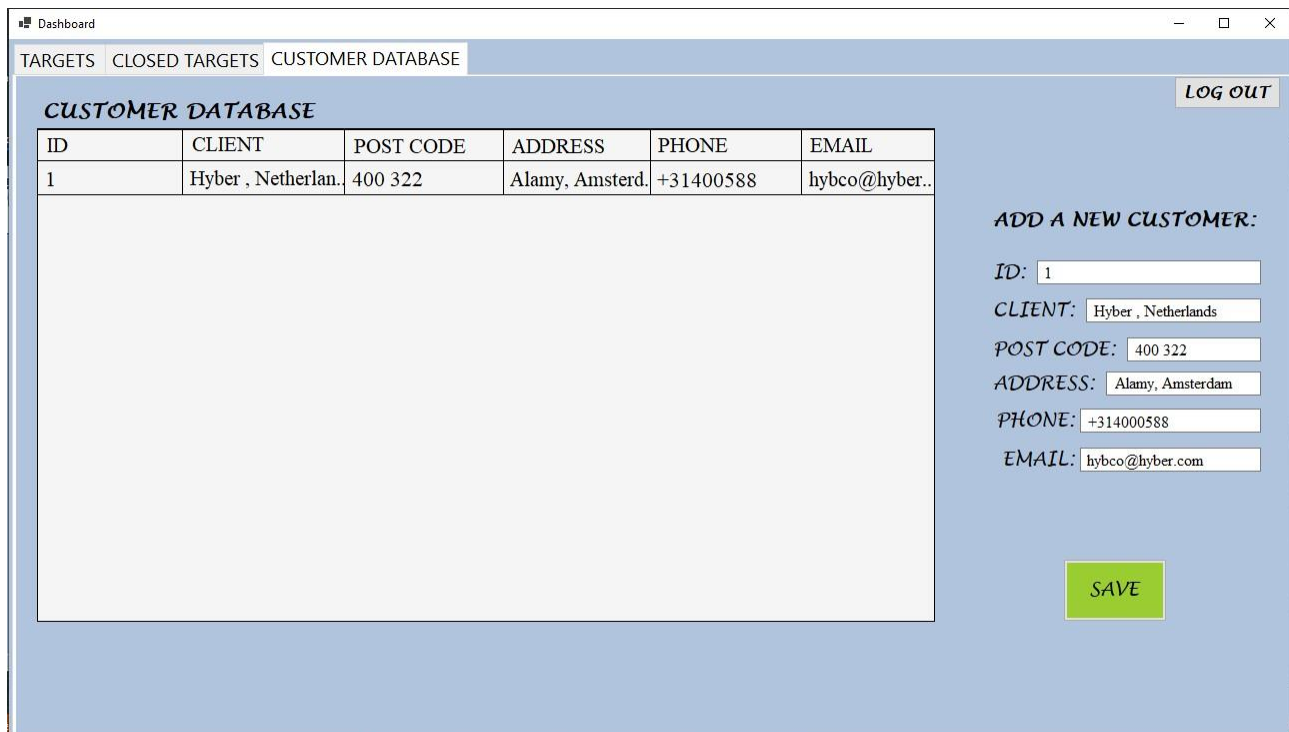
POST CODE:

ADDRESS:

PHONE:

EMAIL:

On clicking 'SAVE', the details are shown in the table.



The screenshot shows the same web application window, but now the table in the 'CUSTOMER DATABASE' tab has one row of data. The input fields on the right are still populated with the same values as in the previous screenshot. The green 'SAVE' button is still present.

ID	CLIENT	POST CODE	ADDRESS	PHONE	EMAIL
1	Hyber , Netherlan..	400 322	Alamy, Amsterd..	+31400588	hybco@hyber..

ADD A NEW CUSTOMER:

ID:

CLIENT:

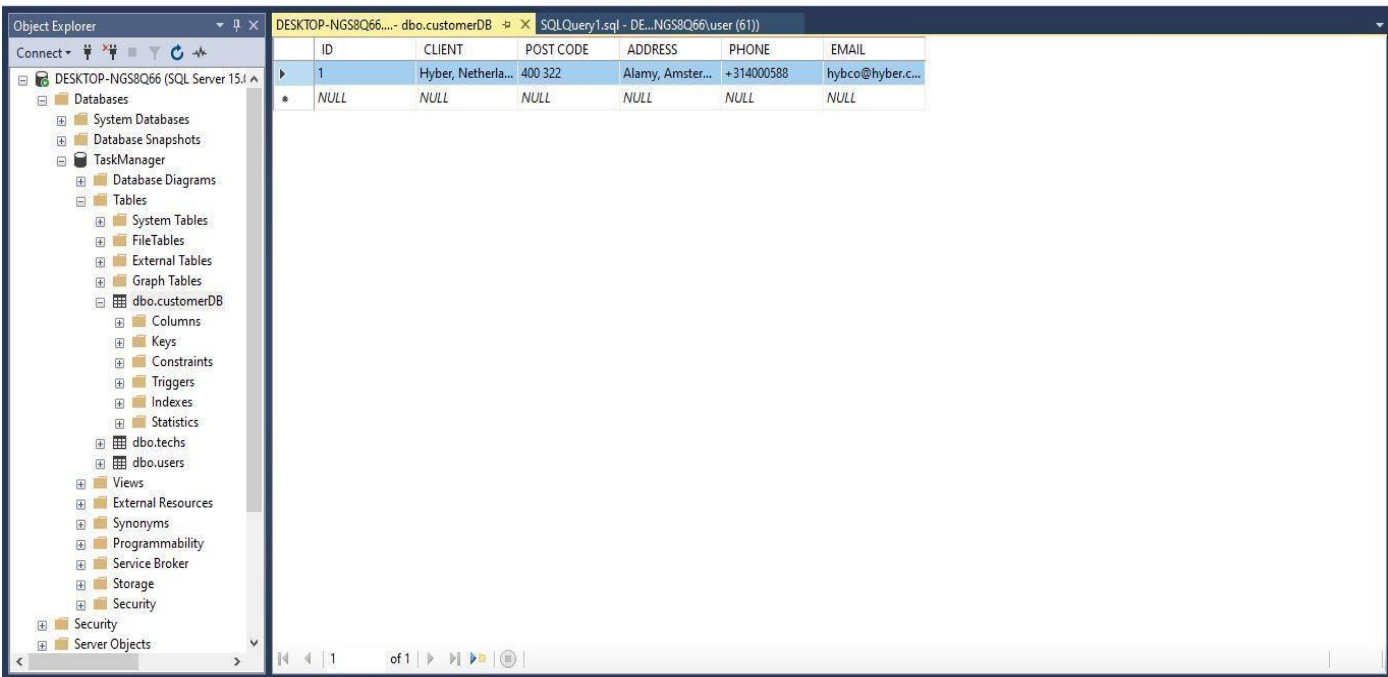
POST CODE:

ADDRESS:

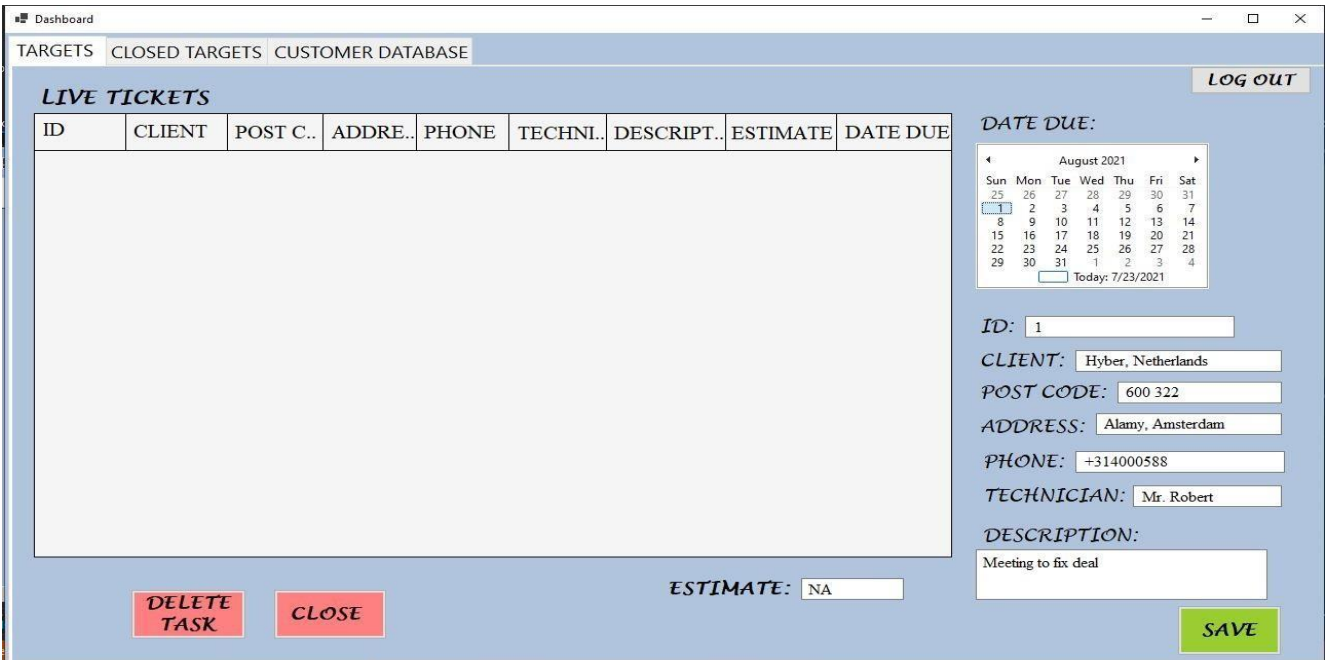
PHONE:

EMAIL:

Simultaneously, this also gets updated within the database in SQL Server.



Since this is not a deleted/closed task, it shows up under the ‘LIVE TARGETS’ bar. The Due Date can be set by the user and gets saved once the save button is clicked. In this case we select August 1st.



Pressing 'SAVE' also clears the textboxes which enables us to enter new tasks.

Dashboard

TARGETS

CLOSED TARGETS

CUSTOMER DATABASE

LOG OUT

LIVE TICKETS

ID	CLIENT	POST C..	ADDRE..	PHONE	TECHNI..	DESCRIPT..	ESTIMATE	DATE DUE
1	Hyber, Neth..	600 322	Alamy, Ams..	+314000588	Mr.Robert	Meeting to fix de..	NA	8/01/2021

DATE DUE:

August 2021

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Today: 7/23/2021

ID:

CLIENT:

POST CODE:

ADDRESS:

PHONE:

TECHNICIAN:

DESCRIPTION:

ESTIMATE:

SAVE

DELETE TASK

CLOSE

Once we are done with a task, we can either delete/close it. 'DELETE TASK' permanently deletes a task whereas 'CLOSE' closes a task and it shows up in the 'CLOSED TARGETS' bar.

Dashboard

TARGETS

CLOSED TARGETS

CUSTOMER DATABASE

LOG OUT

CLOSED TICKETS

ID	CLIENT	POST CODE	ADDRESS	PHONE	TECHNIC..	DESCRIPT..	ESTIMAT..
1	Hyber, Netherlan.	600 322	Alamy, Amster..	+314000588	Mr. Robert	Have a discussio..	NA

TOTAL INVOICED:

NA

DELETE

An excel sheet of the closed targets can be made available by clicking on the Excel icon to the right. This can be saved for the user’s future reference.

FILEHOMEINSERTPAGE LAYOUTFORMULASDATA REVIEWVIEW

Cut

Copy

Paste

Format Painter

Clipboard

Calibri11A⁺A⁻

B

I

U

Font

Alignment

Wrap Text

Merge & Center

Number

General

\$%‰.00→.0

Conditional Formatting

A2

X

✓

fx

1

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ID	CLIENT	POST COD	ADDRESS	PHONE	TECHNICIAN	DESCRIPTION	ESTIMATE	Column1				
2		1 Hyber, Ne	600 322	Alamy, An	314000588	Mr.Robert		NA	#####				
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													