

DAILY TASK REMINDER APP CONNECTED WITH DATA BASE

CODE :

```
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 Daily_TaskReminder
{
    public partial class Form1 : Form
    {
        SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-
SJBQ00P\SQLEXPRESS01;
Initial Catalog=DailyTasksDB;Integrated Security=True");
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            LoadTasks();
        }
        private void LoadTasks()
        {
            con);
            SqlDataAdapter sda = new SqlDataAdapter("SELECT * FROM Tasks",
            DataTable dt = new DataTable();
            sda.Fill(dt);
            dgvTasks.DataSource = dt;
        }

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

        private void btnAddTask_Click(object sender, EventArgs e)
        {
            try
            {
```

```

        // Open the connection
        con.Open();

        // SQL insert command
        string query = "INSERT INTO Tasks (TaskName, TaskDate,
TaskTime ) VALUES (@name, @date, @time )";

        // Create SqlCommand object
        SqlCommand cmd = new SqlCommand(query, con);

        // Pass values from the form controls
        cmd.Parameters.AddWithValue("@name", txtTaskName.Text);
        cmd.Parameters.AddWithValue("@date",
dtpDueDate.Value.Date);
        cmd.Parameters.AddWithValue("@time",
dtpDueTime.Value.ToShortTimeString());

        // Execute the insert command
        cmd.ExecuteNonQuery();

        // Show success message
        MessageBox.Show("Task added successfully!");

        // Close the connection
        con.Close();

        // Refresh the task list in DataGridView
        LoadTasks();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message);
        con.Close();
    }
}

private void btnDeleteTask_Click(object sender, EventArgs e)
{
    try
    {
        if (dgvTasks.SelectedRows.Count > 0)
        {
            int taskId =
Convert.ToInt32(dgvTasks.SelectedRows[0].Cells["TaskID"].Value);

            con.Open();

            string query = "DELETE FROM Tasks WHERE TaskID =
@TaskID";

            SqlCommand cmd = new SqlCommand(query, con);
            cmd.Parameters.AddWithValue("@TaskID", taskId);
            cmd.ExecuteNonQuery();

            con.Close();

            MessageBox.Show("Task deleted successfully!");
            LoadTasks(); // Refresh DataGridView
        }
        else
        {
            MessageBox.Show("Please select a task to delete.");
        }
    }
}

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show("Error deleting task: " + ex.Message);
    }
}

private void dgvTasks_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (e.RowIndex >= 0)
    {
        DataGridViewRow row = dgvTasks.Rows[e.RowIndex];

        txtTaskName.Text = row.Cells["TaskName"].Value.ToString();
        dtpDueDate.Value =
Convert.ToDateTime(row.Cells["DueDate"].Value);
        dtpDueTime.Value =
Convert.ToDateTime(row.Cells["DueTime"].Value);
    }
}

private void btnUpdateTask_Click(object sender, EventArgs e)
{
    if (dgvTasks.SelectedRows.Count > 0)
    {
        int taskId =
Convert.ToInt32(dgvTasks.SelectedRows[0].Cells["TaskID"].Value);
        string taskName = txtTaskName.Text;
        DateTime dueDate = dtpDueDate.Value.Date;
        DateTime dueTime = dtpDueTime.Value;

        string query = "UPDATE Tasks SET TaskName = @TaskName,
DueDate = @DueDate, DueTime = @DueTime WHERE TaskID = @TaskID";

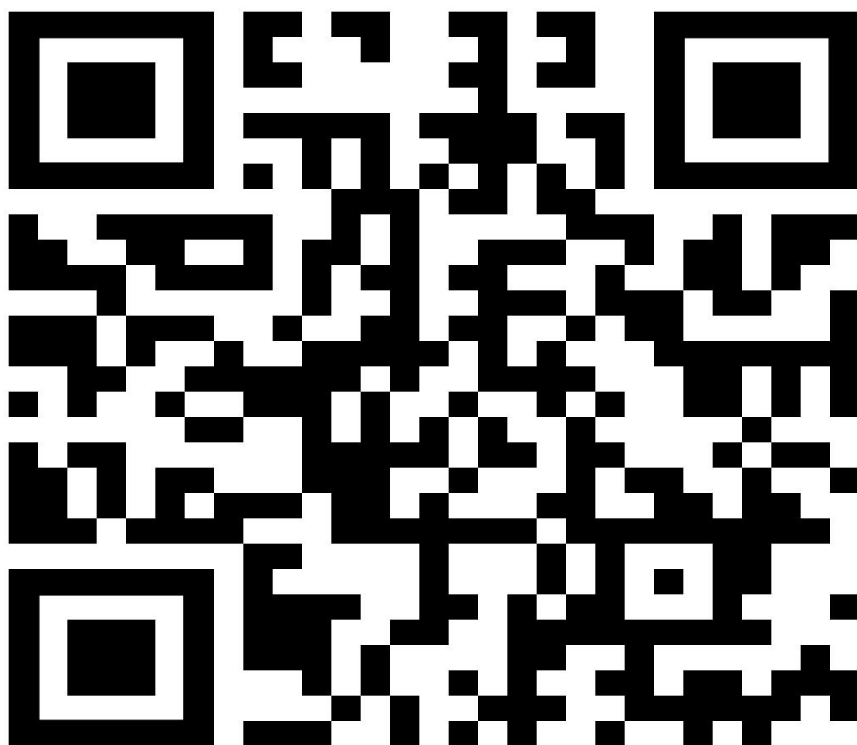
        using (SqlConnection conn = new SqlConnection(@"Data
Source=DESKTOP-SJBQ00P\SQLEXPRESS01;Initial
Catalog=DailyTasksDB;Integrated Security=True"))
        {
            conn.Open();
            SqlCommand cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@TaskName", taskName);
            cmd.Parameters.AddWithValue("@DueDate", dueDate);
            cmd.Parameters.AddWithValue("@DueTime", dueTime);
            cmd.Parameters.AddWithValue("@TaskID", taskId);

            cmd.ExecuteNonQuery();
        }

        MessageBox.Show("Task updated successfully!");
        LoadTasks();
    }
    else
    {
        MessageBox.Show("Please select a task to update.");
    }
}
}

```

QR CODE :



OUTPUT PICTURE :

