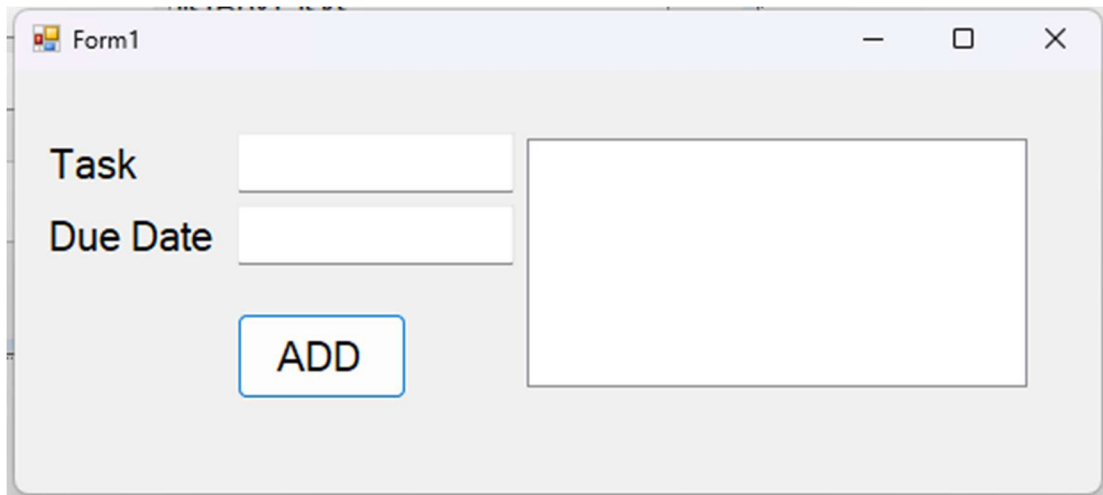


Practical Assessment

The application manages a simple to-do list with **(Form Design Elements)** :-

- TextBox: txtTask
- TextBox: txtDueDate
- Button: btnAddTask
- ListBox: listBoxTasks



Code for “btnAddTask”

```
private void btnAddTask_Click(object sender, EventArgs e)
{
    string taskName = txtTask.Text;
    string dueDate = txtDueDate.Text;

    // No validation in v1.0
    string taskEntry = taskName + " - Due: " + dueDate;
    listBoxTasks.Items.Add(taskEntry);

    // Clear inputs
    txtTask.Clear();
    txtDueDate.Clear();
}
```

Task 1: Corrective Maintenance

Tasks:

- Fix the following issues:
 1. Allows adding a task with empty name or due date.
 2. Accepts invalid due date formats (e.g., "abc").
- Use proper input validation and display errors using `MessageBox.Show()`.
- Submit fixed code and a bug tracking sheet that includes:
 - Original issue
 - Correction made

Corrected Code Snippet (`btnAddTask_Click`)

```
private void button1_Click(object sender, EventArgs e)
{
    string taskName = txtTask.Text.Trim();
    string dueDateText = txtDueDate.Text.Trim();

    if (string.IsNullOrEmpty(taskName) ||
string.IsNullOrEmpty(dueDateText))
    {
        MessageBox.Show("Task name and due date cannot be empty.", "Input
Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    DateTime dueDate;
    if (!DateTime.TryParse(dueDateText, out dueDate))
    {
        MessageBox.Show("Please enter a valid date in the format
MM/DD/YYYY.", "Invalid Date", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        return;
    }
}
```

```

        listBoxTasks.Items.Add(taskName + " - Due: " +
dueDate.ToShortDateString());
        txtTask.Clear();
        txtDueDate.Clear();
    }

```

Task 2: Perfective Maintenance

Tasks:

- Add features:
 1. Clear button to reset inputs
 2. Delete button to remove selected task
 3. GroupBox to organize form controls with descriptive Labels
- Submit improved application and perfective change log with justification

New Features Added

Feature	Description	Justification
Clear Button	Clears input fields (txtTask, txtDueDate).	Enhances user convenience during data entry.
Delete Button	Deletes the selected task from the ListBox.	Improves usability by allowing task removal.

The screenshot shows a Windows application window titled "Form1". Inside the window, there are two text input fields. The first is labeled "Task" and the second is labeled "Due Date". To the right of these fields is a large, empty rectangular box, which is likely intended for a list box to display tasks. Below the input fields, there are three buttons: "ADD", "Clear", and "Delete". The "Delete" button is currently selected, indicated by a blue border around it.

Code for “btnClear”

```
private void btnClear_Click(object sender, EventArgs e)
{
    txtTask.Clear();
    txtDueDate.Clear();
}
```

Code for “btnDelete”

```
private void btnDelete_Click(object sender, EventArgs e)
{
    if (listBoxTasks.SelectedIndex != -1)
    {
        listBoxTasks.Items.RemoveAt(listBoxTasks.SelectedIndex);
    }
    else
    {
        MessageBox.Show("Please select a task to delete.", "Delete Task",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
```

Question 1: Git Version Control and Change Tracking (25 Marks)

Tasks:

- Create Git repo and commit:
 - Initial version v1.0
 - Bug fixes v1.1
 - Final improved version v2.0
- Submit:
 - Git screenshots and logs
 - Short change impact analysis

Question 2: Testing and User Documentation (25 Marks)

Tasks:

- Manual test cases for:
 - Adding valid/invalid tasks
 - Clearing input
 - Deleting task
- Test results table (Pass/Fail)
- Brief user guide (screenshots included):
 - How to Add, Clear, and Delete tasks
 - Date format rules