

LAB-2 .NET Programming Assignment Documentation

Name: Onur Oksuz **Course:** .NET Programming

Table of Contents

1. Task 1: Using Generic Collection Classes
 2. Task 2: HTTP Protocol and JSON Data Exchange
 3. Task 3: Asynchronous Programming
 4. Task 4: Creating Visual Applications
 5. Task 5: Visual Containers in UI Development
 6. Summary
-

Task 1: Using Generic Collection Classes

Description: This task demonstrates the use of `List<string>` and `Dictionary<string, int>` in C# to store, organize, and display data.

Screenshot:

```
PS C:\Users\iniro\Desktop\Vizja Projects\.NET\LAB-2\Task1_GenericCollections> dotnet run
List of cities:
- Warsaw
- Krakow
- Gdansk
- Lodz

List of countries:
- Poland
- Germany
- France

Updated list of cities:
- Warsaw
- Krakow
- Gdansk
- Lodz
- Poznan

Population of Warsaw: 1800000
```

Explanation: - The program lists cities and countries using `List<string>`.

- It displays the population of a city using `Dictionary<string, int>`.

- All requirements and modifications were implemented as instructed.

Task 2: HTTP Protocol and JSON Data Exchange

Description: This task demonstrates how to use `HttpClient` to retrieve JSON data from an API and display it in the console.

Screenshot:

```
PS C:\Users\iniro\Desktop\Vizja Projects\.NET\LAB-2\Task2_HttpJson> dotnet run
Post Title: qui est esse
Post ID: 2
```

Explanation: - The program retrieves a post from a JSON API and prints its title and ID.

- JSON deserialization is handled with case-insensitive property matching.
 - The output confirms successful data retrieval and display.
-

Task 3: Asynchronous Programming

Description: This task demonstrates asynchronous programming in C# using `async` and `await`.

Screenshot:

```
PS C:\Users\iniro\Desktop\Vizja Projects\.NET\LAB-2\Task3_Async> dotnet run
Downloading data...
Downloaded content (first 100 chars): [
  {
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excep...
```

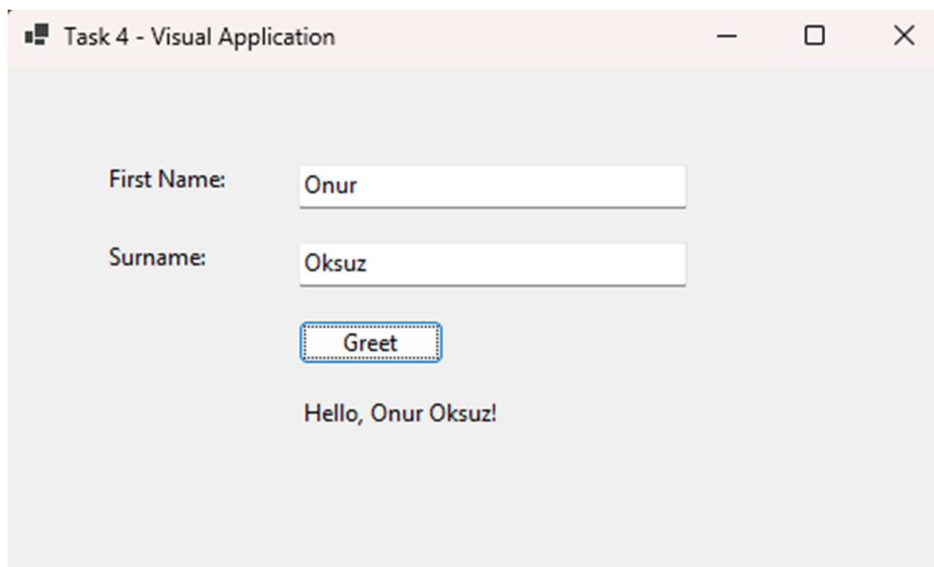
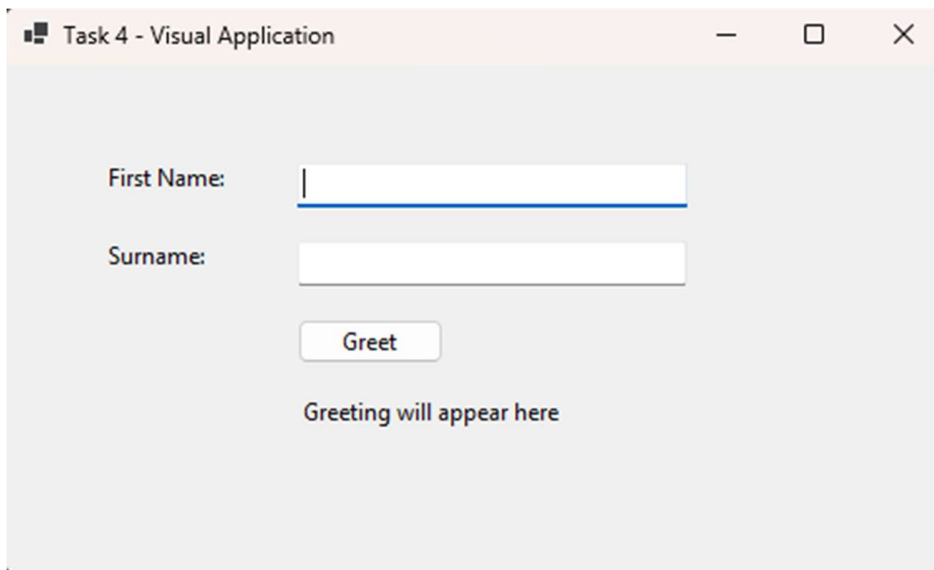
Explanation: - The program downloads data asynchronously and prints the first 100 characters.

- The console output verifies correct asynchronous behavior.
-

Task 4: Creating Visual Applications

Description: This task demonstrates building a Windows Forms application with text boxes, a button, and a label.

Screenshot:



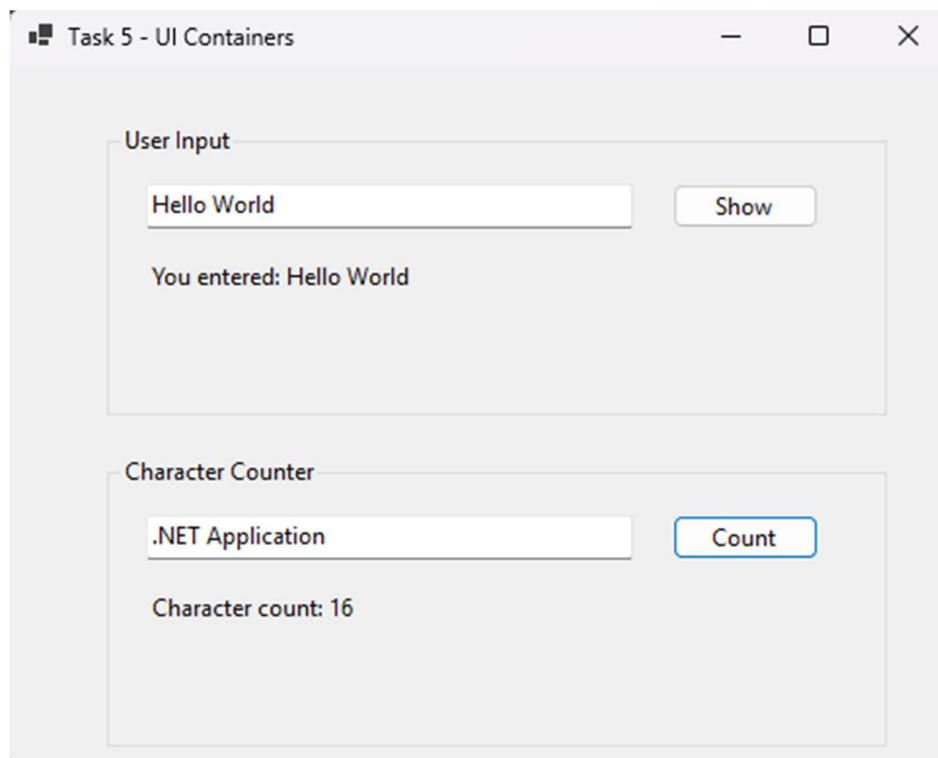
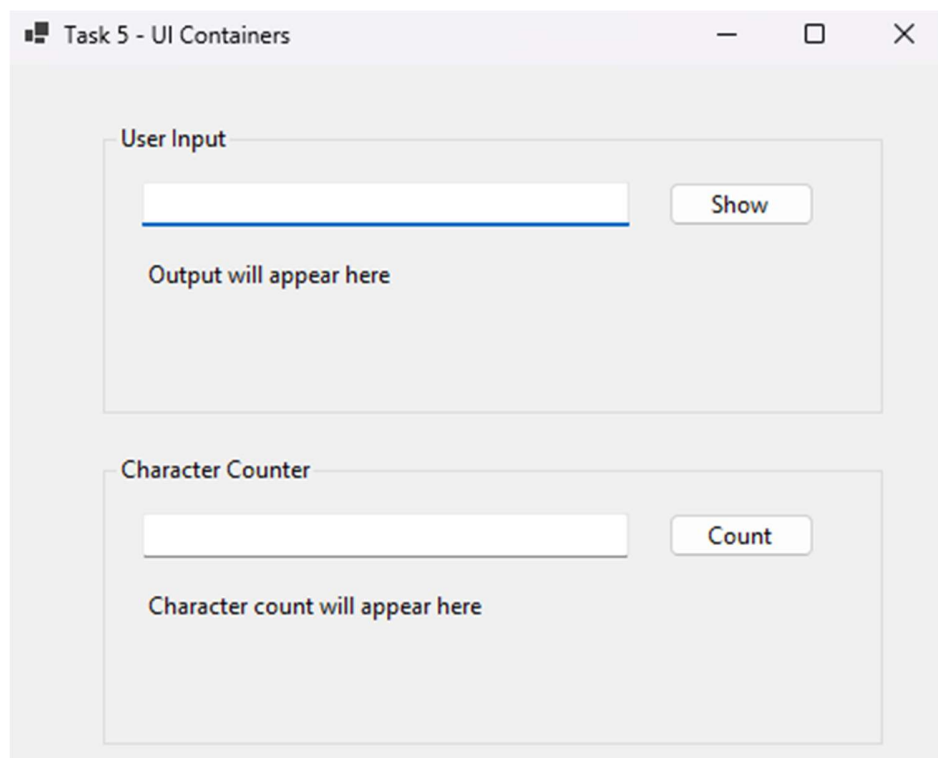
Explanation: - The app greets the user by their first name and surname.

- The screenshot shows the UI with the required controls and a sample greeting.

Task 5: Visual Containers in UI Development

Description: This task demonstrates organizing controls into visual groups using GroupBox in a Windows Forms application.

Screenshot:



Explanation: - The app uses two GroupBoxes: one for user input and one for counting characters.

- The screenshot shows both features working as required.

Summary

All tasks were completed as required. The screenshots above provide evidence of each working solution. Through this lab, I learned about collections, HTTP/JSON, async programming, and WinForms UI in .NET. Any challenges encountered were solved through debugging and research.