Course: EGDF20
Module: EGE202 Application Programming

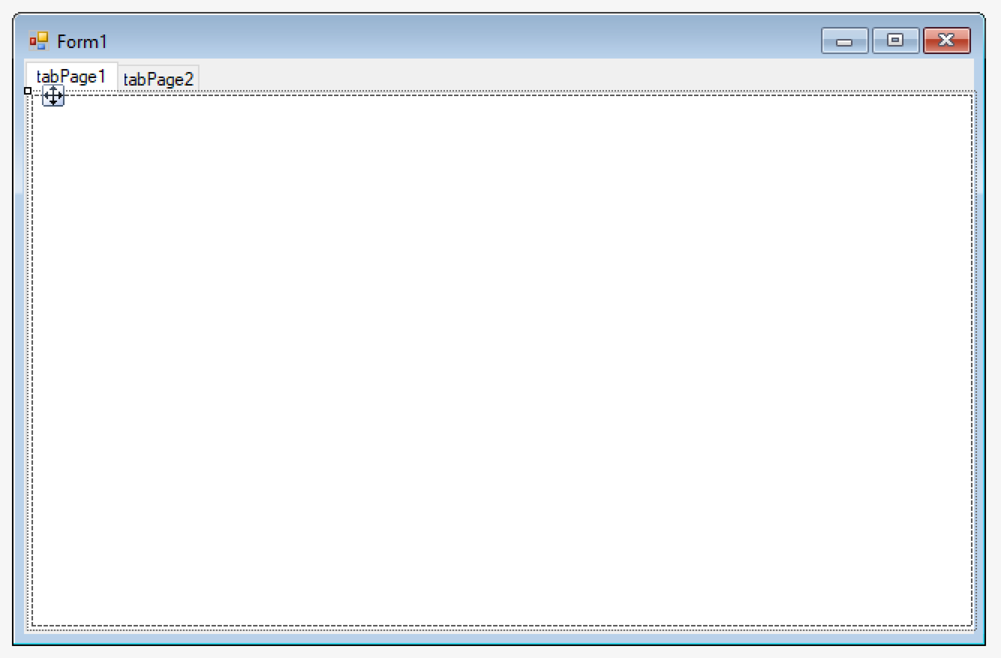
Practical 10: Integrating ChatBot to WinForm Application

Objectives: At the end of this lab, the student should be integrate browser and bot related technology into WinForm using 3rd party library. They will also experiment with container controls such as TabControl and other advanced controls such as WebBrowser control.

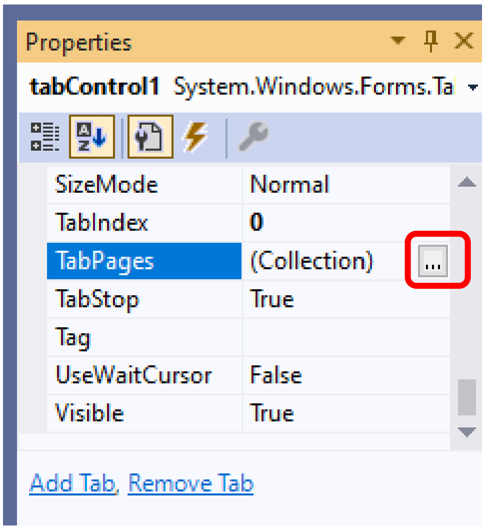
Exercise 1 – Integrating Browser Control

Part 1: UI Design with TabControl

1. Under the ***File*** menu, click ***New Project*** or use the ***New Project*** button to create a new project. Alternatively, use the ***Create New Project*** link in the ***Get Started*** *popup* dialog.
2. From the pop-up dialog, select **"C#"** for the ***Language filter***, **"Windows"** for ***the Platform filter*** and **"Desktop"** for the ***Project type filter***.
3. Then choose ***Windows Forms App (.Net Framework)*** and click the ***Next*** button.
4. Type the name of your new project as ***ChatBotServer*** and keep the Solution name the same as Project name.
5. Set the Location to put the project in your own created folder and finally click on the ***OK*** button.
6. Double click on "Form1.cs" ***Solution Explorer*** window to launch the ***Form Designer*** tab.
7. From the ***Toolbar***, drag in 1 *TabControl* control into the *Form1* window. Resize the control to fit the *Form1* client area.



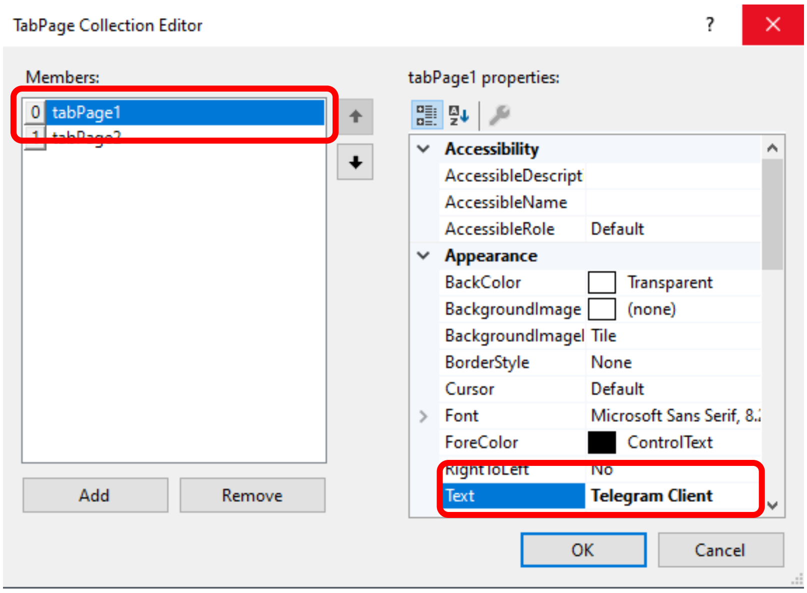
8. Select the *TabControl* and modify the *TabPages* properties by clicking the ***button*** with 3 dots.



Properties

tabControl1 System.Windows.Forms.Ta

SizeMode	Normal
TabIndex	**0**
TabPages	(Collection) ...
TabStop	True
Tag	
UseWaitCursor	False
Visible	True



TabPage Collection Editor

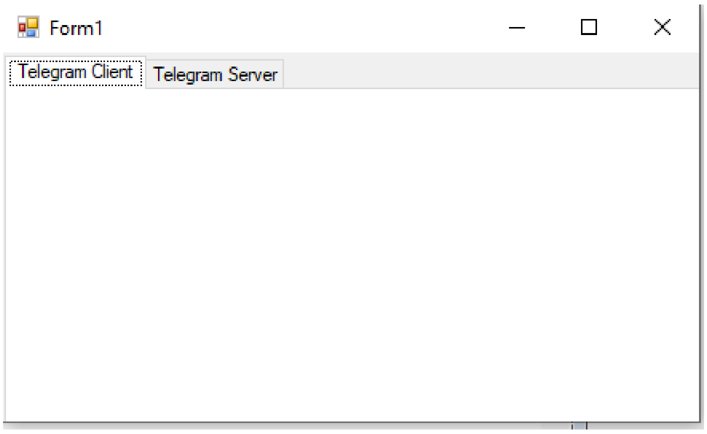
Members:

0 tabPage1
1 tabPage2

tabPage1 properties:

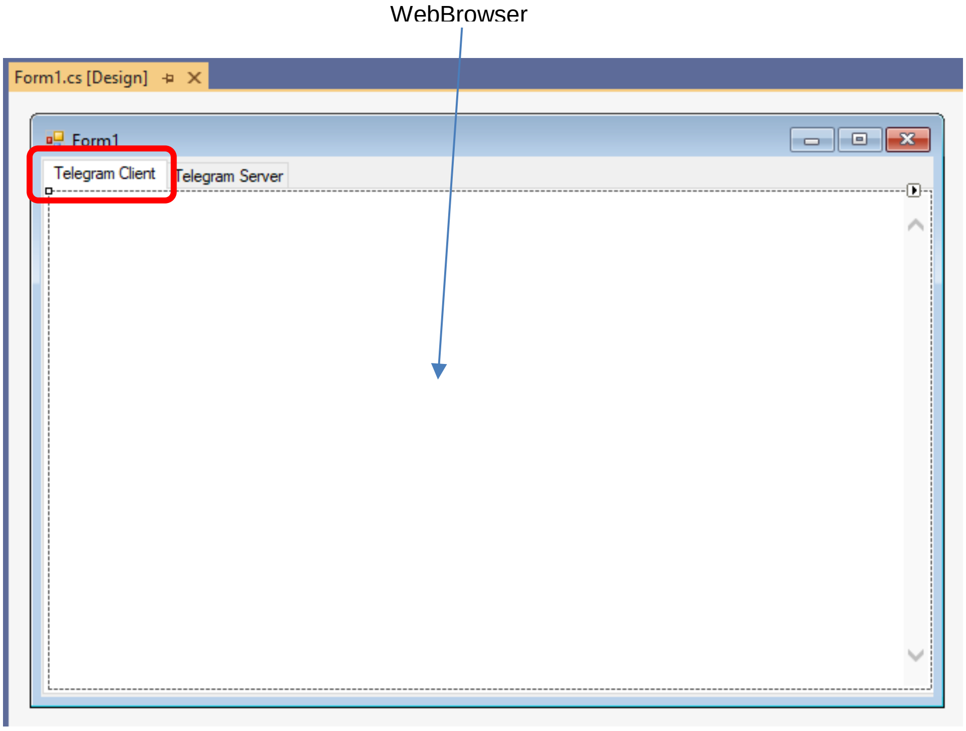
Accessibility	
AccessibleDescript	
AccessibleName	
AccessibleRole	Default
Appearance	
BackColor	□ Transparent
BackgroundImage	□ (none)
BackgroundImagel	Tile
BorderStyle	None
Cursor	Default
Font	Microsoft Sans Serif, 8.:
ForeColor	■ ControlText
RightToLeft	No
Text	**Telegram Client**

9. With the *tabPage1* selected, modify the *Text* property to ***Telegram Client.***
10. Similarly, with *tabPage2* selected, modify the *Text* property to ***Telegram Server.***
11. Build and test the application.

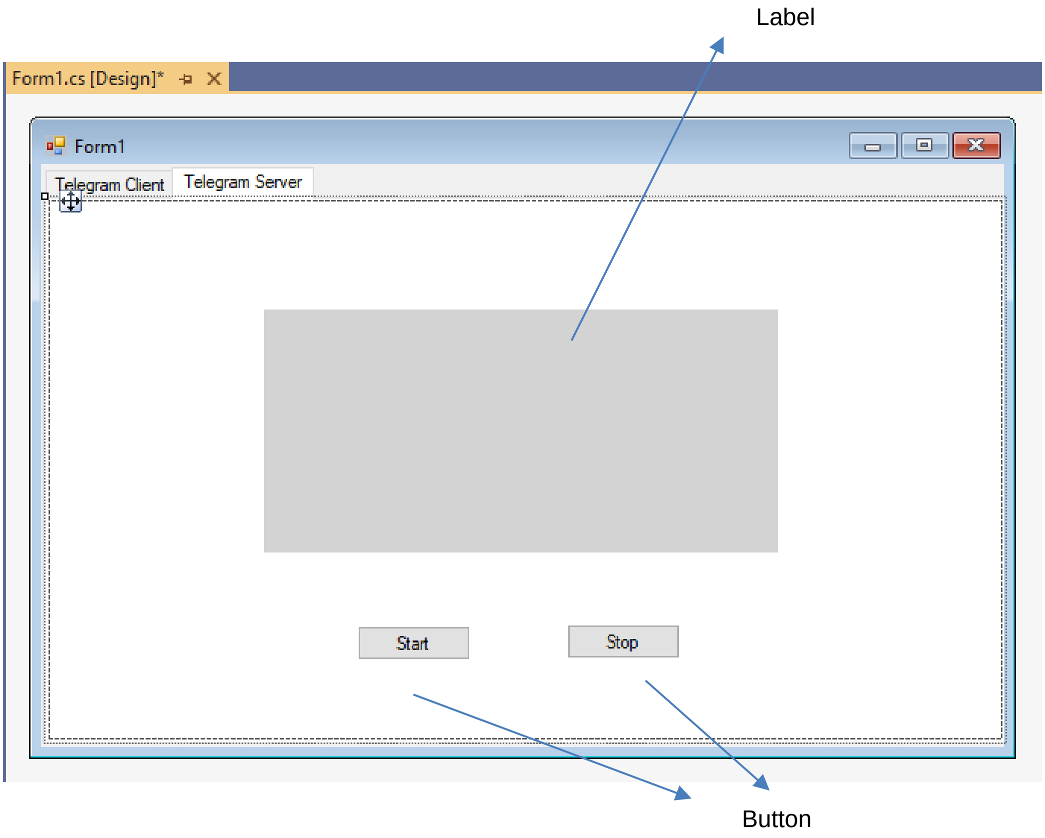


Part 2: Designing individual TabPages

1. Back to the *Form Designer*, select the *Telegram Client* tab. From the ***Toolbar***, drag in 1 *WebBrowser* control into the *Form1* window.



2. Next select the *Telegram Server* tab. From the ***Toolbar***, drag in *1 Label and 2 Button* controls into the *Form1* window.



3. Modify the properties of the controls as follows:

{Name} From	{Name} To	{Text}	{AutoSize}	{BackColor}
label1	lblMesg		False	Web->LightGray
button1	btnStart	Start		
button2	btnStop	Stop		

4. Resize the lblMesg to cover bigger area
5. Build and run your application and test out the tabpages.

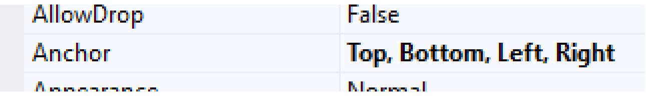
Part 3: Implementing Web Client using Browser Control

1. In the ***Form Designer*** double click on the ***Form1*** to create *Form1_Load(…)* event handler.

```
private void Form1_Load(object sender, EventArgs e)
{
    this.webBrowser1.Navigate("https://www.telegram.org/");
}
```

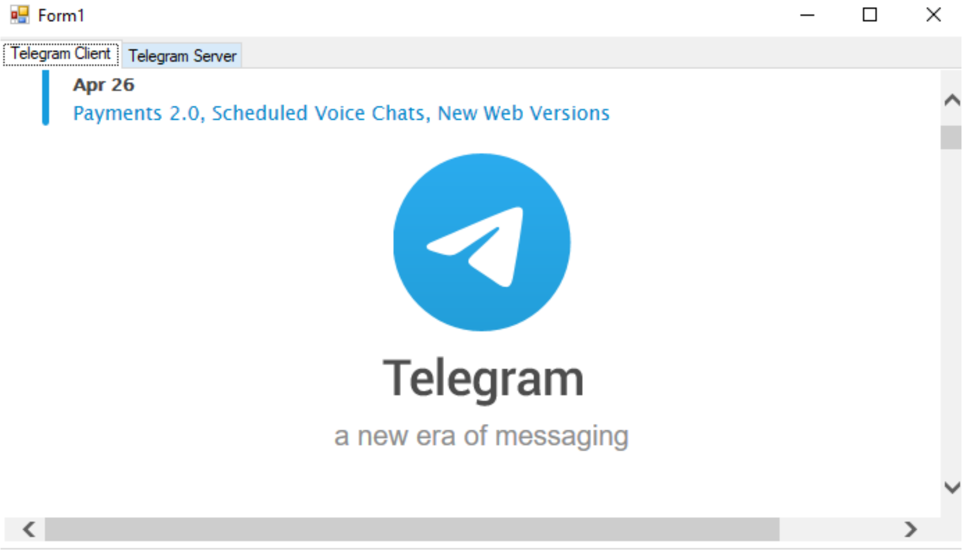
Note: web.telegram.org is needed to access the browser version of telegram. However, WebBrowser control is not compatible with website causing scripting error.

2. Build and run your application. Try resize the Form, did the browser control automatically resize?
3. Let's modify both the *Anchor* property for ***tabControl1*** and ***webBrowser1*** control to set it to ***Top, Bottom, Left, Right***



AllowDrop	False
Anchor	**Top, Bottom, Left, Right**
Appearance	Normal

4. Build and run your application. Try resize the Form, did the browser control automatically resize?



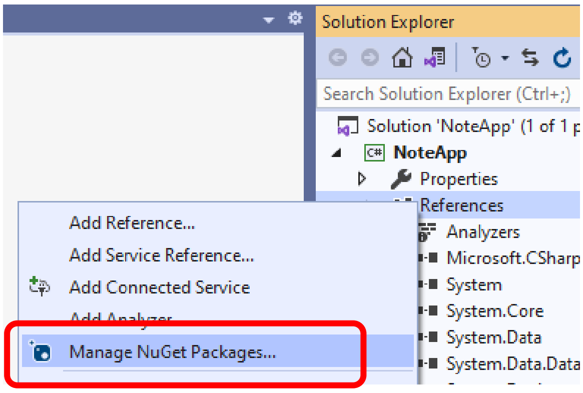
Apr 26
Payments 2.0, Scheduled Voice Chats, New Web Versions

Telegram
a new era of messaging

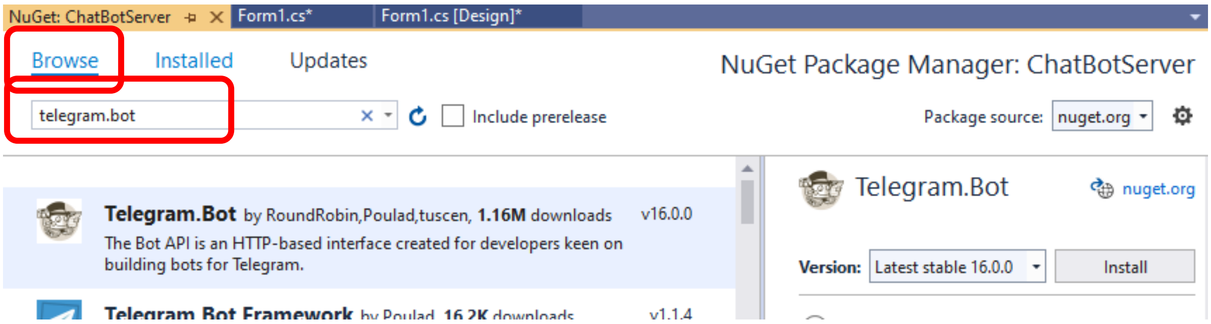
Exercise 2 – Integrating Telegram Bot Server

Part 1: Setting up Telegram.Bot Library

1. Using the same solution/project, at the ***Solution Explorer***, right click References and select ***Manage NuGet Packages***.

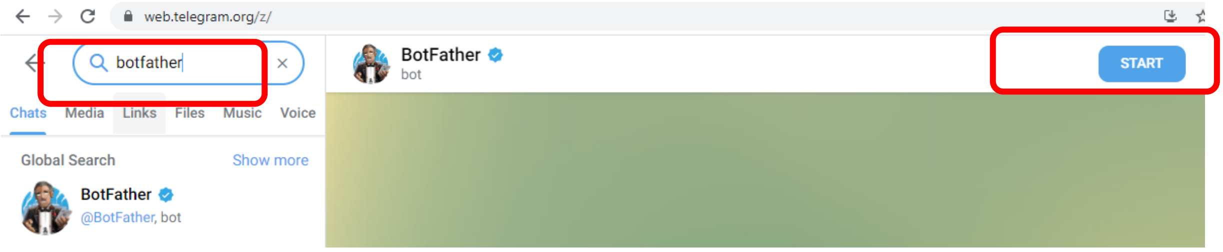


2. Then search for "***Telegram.Bot"*** in Browse tab and click install. ***Choose version 16.x.x or 15.x.x***


NuGet: ChatBotServer
Form1.cs*
Form1.cs [Design]*
Browse
Installed
Updates
NuGet Package Manager: ChatBotServer
telegram.bot
Include prerelease
Package source: nuget.org
Telegram.Bot by RoundRobin,Poulad,tuscen, 1.16M downloads
v16.0.0
The Bot API is an HTTP-based interface created for developers keen on building bots for Telegram.
Telegram.Bot Framework by Poulad, 16.2K downloads
v1.1.4
Telegram.Bot
nuget.org
Version: Latest stable 16.0.0
Install

Part 2: Create a Telegram Bot using Bot Father

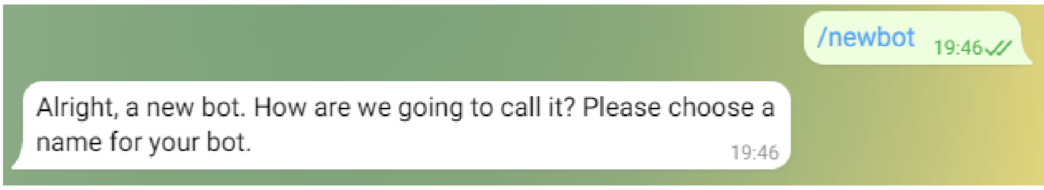
1. Create a Telegram Account (if you do not have one)
2. Before you start developing a bot, you need to talk to @BotFather on Telegram. Register a bot with him and get an access token.



3. Access token is a key used to identify and authorize your bot in API requests so keep it with yourself as a secret. It looks like this

```
1234567:4TT8bAc8GHUspu3ERYn-KGcvsvGB9u_n4ddy
```

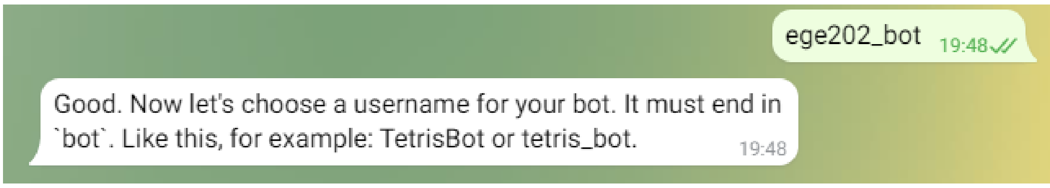
4. You need to create a bot with Bot Father to have your own access token
5. First use the command **/newbot** to create one. The bot name is only for reference purposes. Give it a name.



/newbot 19:46

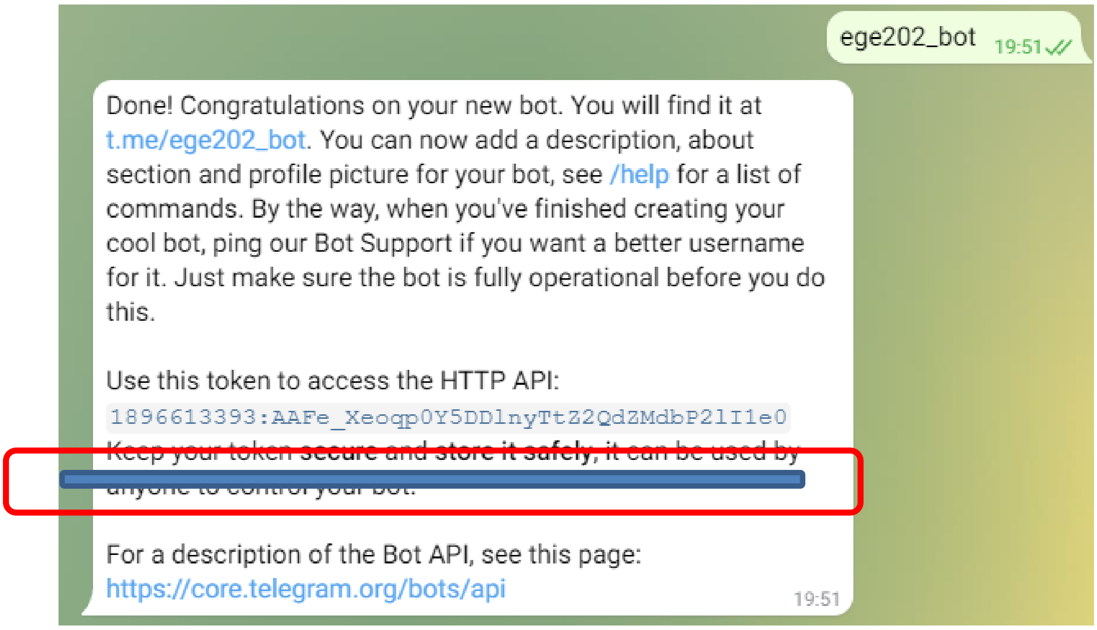
Alright, a new bot. How are we going to call it? Please choose a name for your bot. 19:46

6. Next you need to give the bot a username. ***This username must be unique and end with bot.***



ege202_bot 19:48

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 19:48



ege202_bot 19:51

Done! Congratulations on your new bot. You will find it at t.me/ege202_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
1896613393:AAFe_Xeoqp0Y5DDlnyTtZ2QdZMdbP21I1e0
Keep your token secure and store it safely, it can be used by [redacted]

For a description of the Bot API, see this page:
https://core.telegram.org/bots/api 19:51

7. Copy and save the access token, you will need it later.
8. Next you need to add the newly create bot to your own Telegram

Part 3: Create a Telegram Bot Server

1. Add the following namespaces to *Form1.cs*

```
using Telegram.Bot;
using Telegram.Bot.Args;
```

2. Add the following codes to the Form1 class

```
public partial class Form1 : Form
{
    ITelegramBotClient botClient;

    public Form1()
    {
        InitializeComponent();
    }
```

3. Now double click on the “Start” ***Button*** in the ***Form Designer***. That will automatically create *btnStart_Click (…)* function.
4. Modify *btnStart_Click (…)* to include the following codes:

```
private void btnStart_Click(object sender, EventArgs e)
{
    botClient = new TelegramBotClient("<use your own token>");

    var me = botClient.GetMeAsync().Result;
    lblMesg.Text = "I am user" + me.Id + " and my name is " + me.FirstName;

    botClient.OnMessage += Bot_OnMessage;
    botClient.StartReceiving();
}
```

5. **Replace the code above with your own token**
6. Add the following codes after the *btnStart_Click (…)* function

```
async void Bot_OnMessage(object sender, MessageEventArgs e)
{
    if (e.Message.Text != null)
    {

        lblMesg.Invoke((MethodInvoker)delegate {
        // Running on the UI thread
        lblMesg.Text = e.Message.From.FirstName + ":" + e.Message.Text;
    });

         await botClient.SendTextMessageAsync(
             chatId: e.Message.Chat,
             text: "You said:\n" + e.Message.Text
         );
    }
}
```

7. Now double click on the “Stop” ***Button*** in the ***Form Designer***. That will automatically create *btnStop_Click (…)* function.
8. Modify *btnStop_Click (…)* to include the following codes:

```
private void btnStart_Click(object sender, EventArgs e)
{
     botClient.StopReceiving();
     lblMesg.Text = "";
}
```