Project 3: Automatic login your FB account.

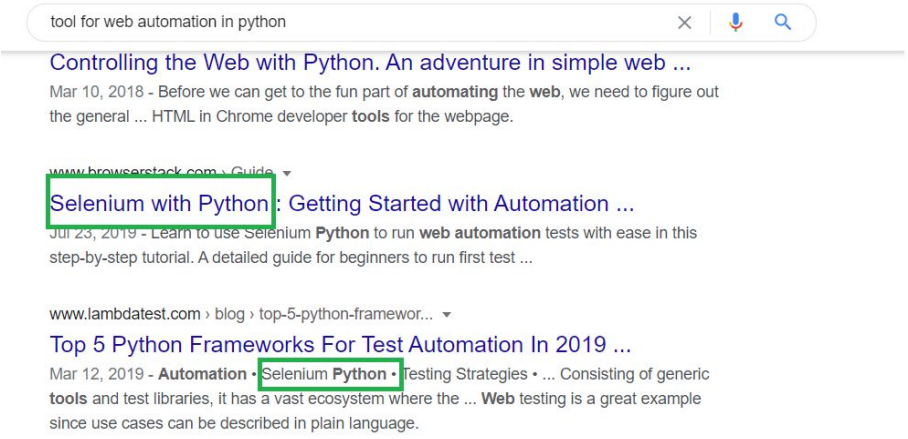
In this project we will be automating the Facebook login. I.e. whenever you run this code, the program will automatically log in to your FB account.

This project is an example of **web automation**. Also known as **browser automation.** Web automation simply means automating any task which we do manually. In web automation for any task done by humans, a program is written to do the same task. In this project we are going to automate the FB login process.

So continuing the approach that we followed for our previous projects, when you have got ideas, you need to search for resources such as tools, modules or functions over the internet.

So now I know that this is web automation so I will search for a module or a tool in python which can help me with the task.

When I searched for a *tool for web automation in python* on Google these are some of the results.



The search results directed me to the python library selenium. **Selenium** is a library which helps in website automation with python. With this library you can make **connections to any website or webpage over the internet.** Now to get this tool, we need to install it. Installing selenium is pretty easy. To install it, follow the steps given below.

Step1: Open CMD as Administrator.

Step2: type pip install selenium

Step3: press enter and wait for the download.

Step4: Once the download is completed, type pip list and press enter

Step5: If you can see selenium there then you are good to go.

Selenium actually requires a **webdriver** to interact with the web page. These interactions are simply reading, finding information and giving information to the webpage. For different search engines different web drivers are available. Some of them are

- Firefox driver.
- Internet explorer driver.
- **Chrome driver.**
- Safari driver.
- Opera driver.
- Android Driver.
- iPhone Driver.
- HTMLUnit driver.

Currently I am going to work with Chromedriver. So to work with it, I need to download it. To download it, follow the steps given below.

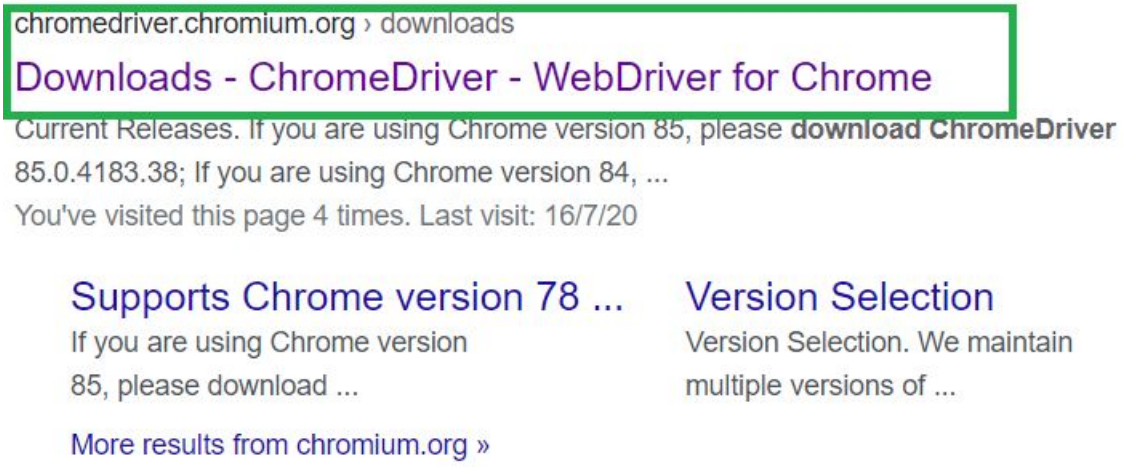
Step1: Search for *Download Chromedr*

After installing both selenium and chrome driver, let's start to write the code.

Let's start with the step that we take while doing this task manually. First of all we **open Google.**

So to do the same task here, we need to import the required libraries. Here we need to **import webdriver from s***iver*

Step2: Click on the first result which will lead you to the website chromedriver.chromium.org



Step3: Here you will be asked to select the version suitable with your chrome version

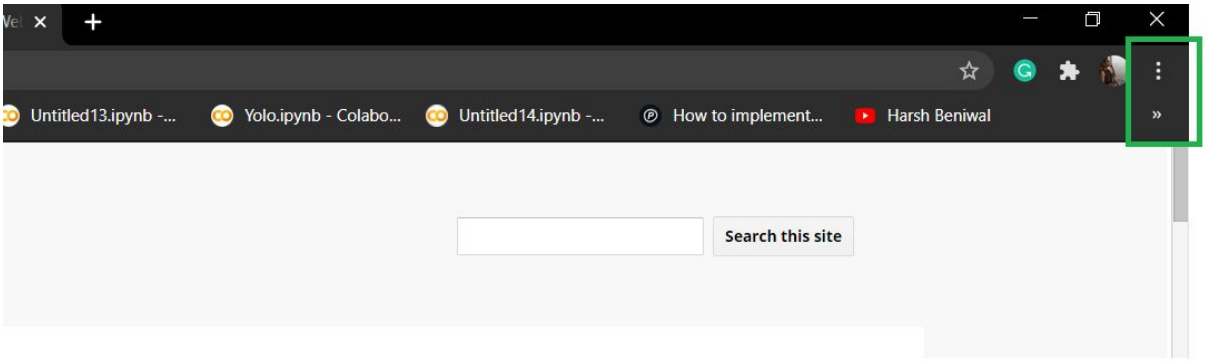
Downloads

Current Releases

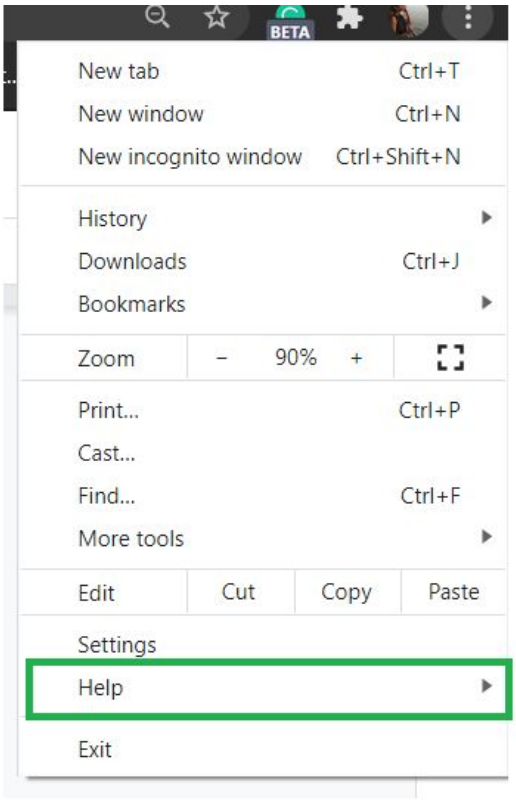
- If you are using Chrome version 85, please download ChromeDriver 85.0.4183.38
- If you are using Chrome version 84, please download ChromeDriver 84.0.4147.30
- If you are using Chrome version 83, please download ChromeDriver 83.0.4103.39
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

If you are using Chrome from Dev or Canary channel, please following instructions on the ChromeDriver Canary p

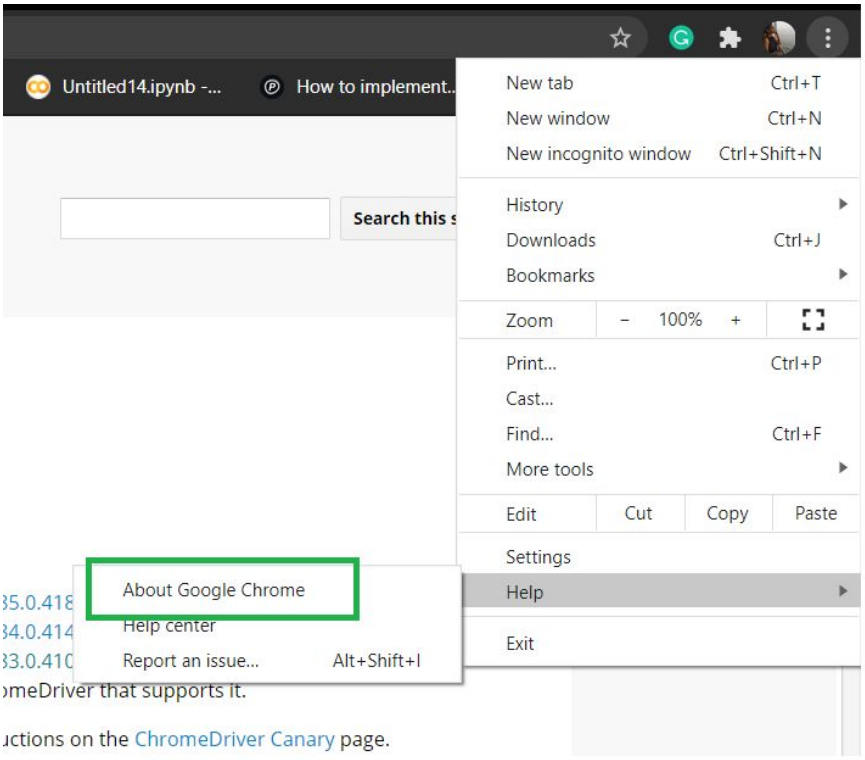
Step4: Now to select a version, you need to know which chrome version you are using. To know this, click on the 3 vertical dots present at the top right of your screen.



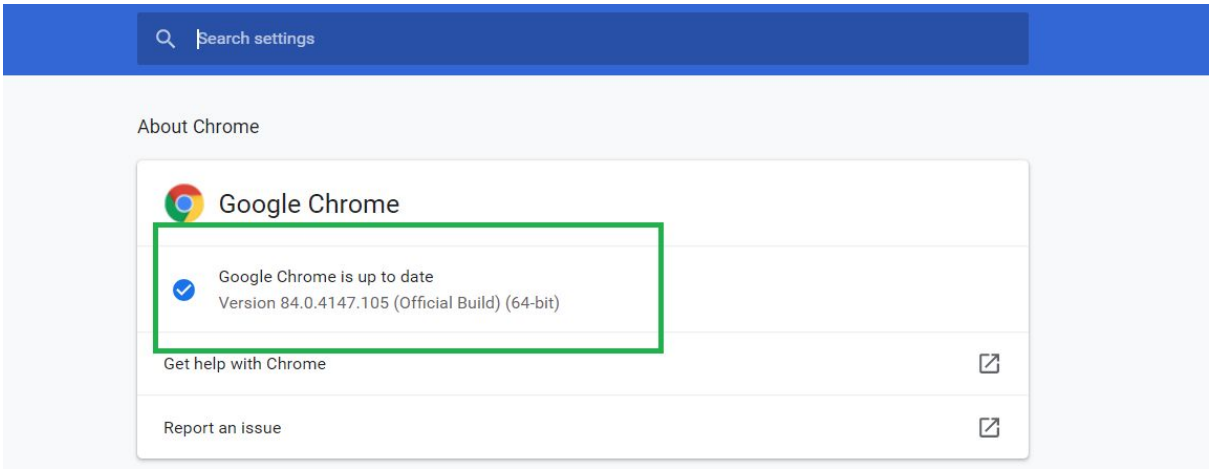
Step5: Go to Help.



Step6: Click on About Google Chrome.

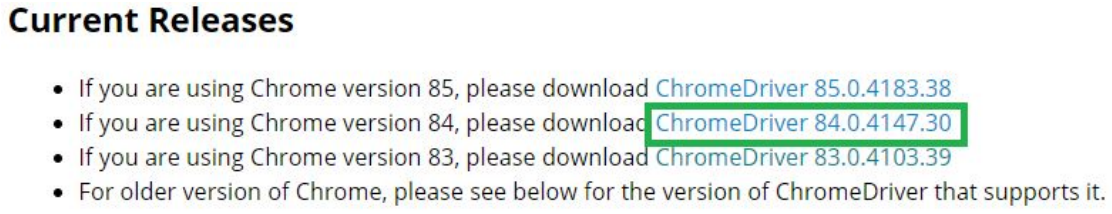


Step7: This is the version of your google chrome.

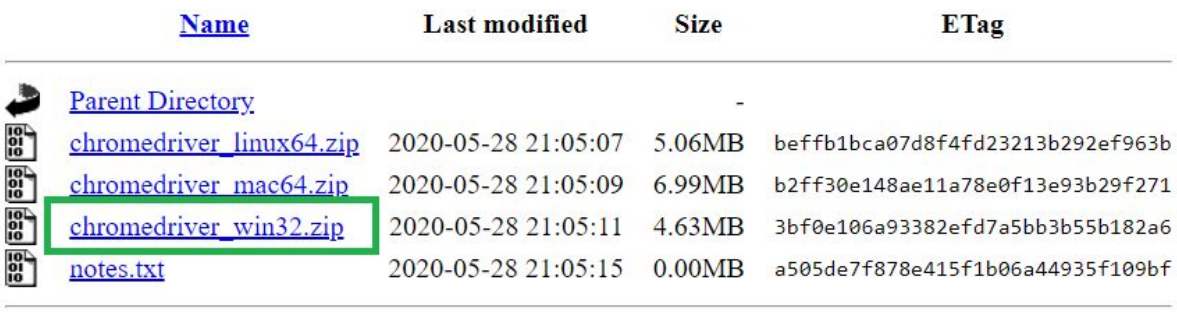


If this is **83.something**, then your version is **83.** If this is **84.something**, then your version is **84**, and the same goes for **85**.

Step8: Now come back to the download page and select the suitable Chrome driver. Here I am selecting ChromeDriver 84.0.4147.30.



Step9: On clicking on it, you will be sent to another page where you will get an option to select from some zip files. Select the one which suits your device. I selected the windows one.



Once clicking on the file a download will start

Step10: Unzip the file and store the location of chromedriver.exe.

elenium. To do that, run the following code.

```
from selenium import webdriver
```

After this, we need to open google from our program. To do this firstly we need to make an object of the webdriver. This newly made object is the thing using which we will interact with the web.

To make a webdriver object, run the following code:

```
browser= webdriver.Chrome(c). Yy
```

You will be giving your path to the chromedriver.exe as the argument. Here **browser** is my chromedriver object using which I will be interacting with any website or webpage. This will be **opening the google website.**

Now to open facebook**,** we will be using .get() method here. Any this function opens any website, whose link is given to it, through the program itself.

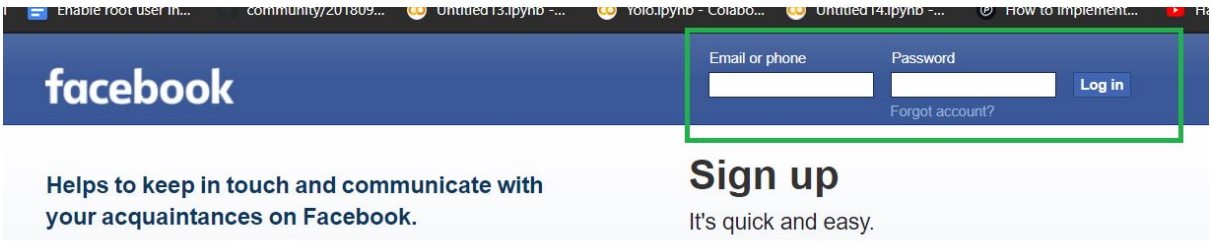
Syntax:

browser_object.get(link_of_the_webpage_to_be_opened): It opens the given webpage.

To open facebook from the program, run the code given below.

```
browser.get('https://www.facebook.com/')
```

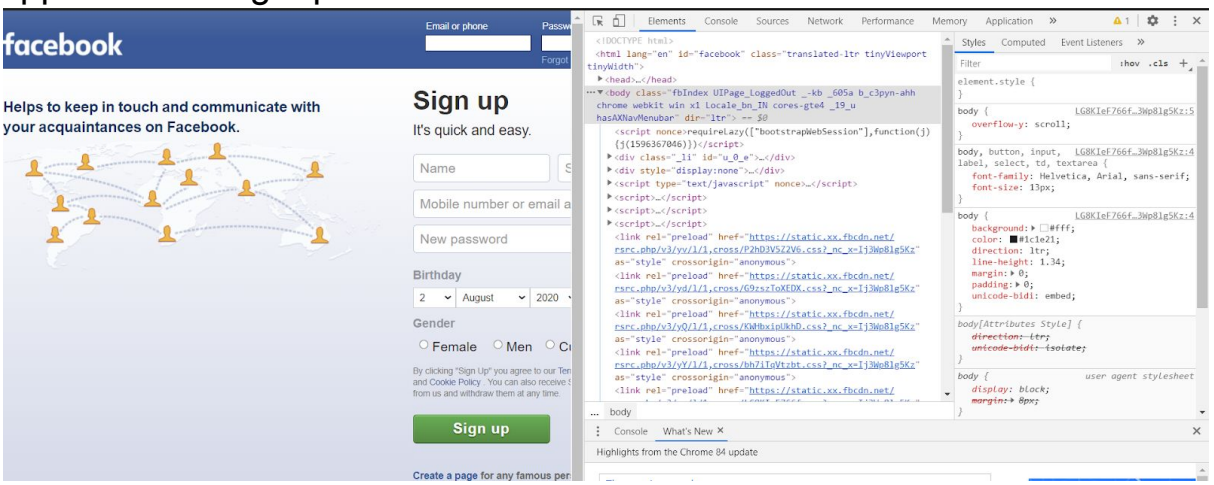
Once you run this, FB will open up in your browser. When this website has been opened you will be seeing the login portal at the top right of the page.



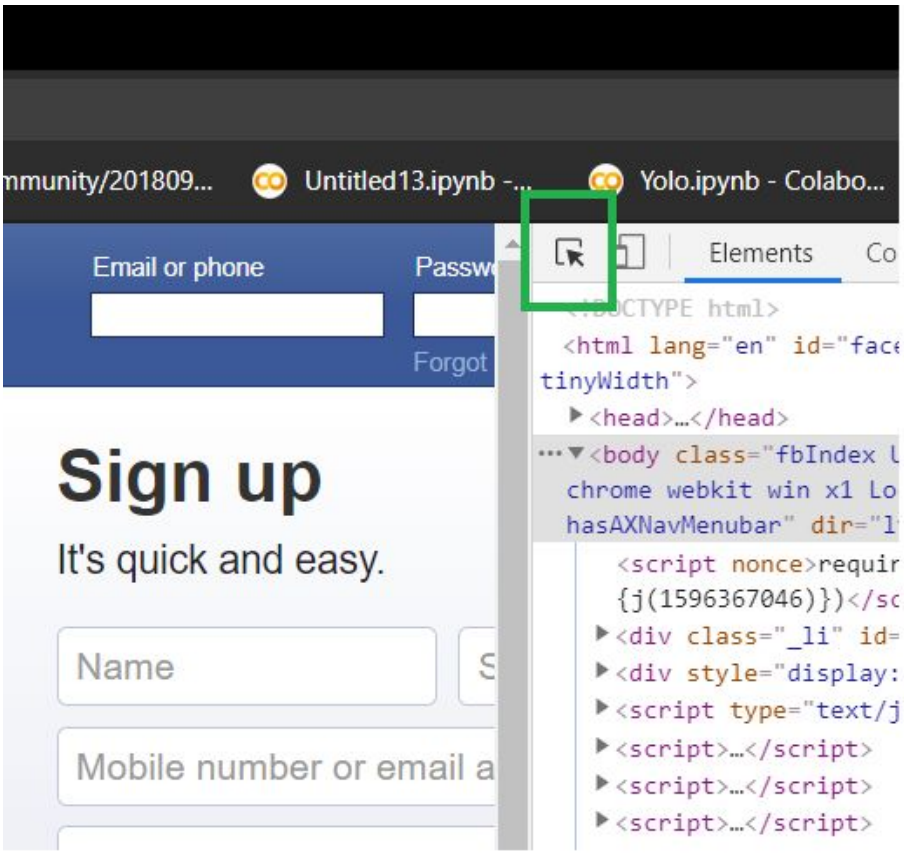
In order to login, we must fill up our credentials and then press Log in.

So our **next step is clicking on the Email or phone box and filling up our Email credential.** To find this text box through our program, we need to look at the **source code** of the page and see how this text box is stored. The source code of a file holds the code or programming done in order to build the page. These codes are generally written in **HTML(HyperText Markup Language)** and are stored as an html file. A **html file** is a file written in HTML and stored with .html extensions.

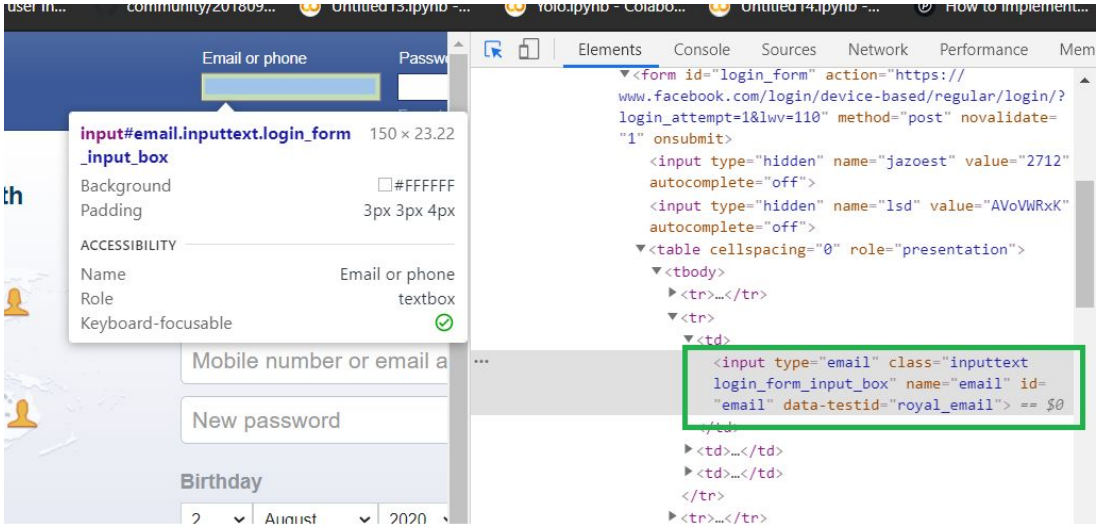
Now to open the source code of the page, press Ctrl+Shift+I while staying at the page. Once you press these keys the source code will appear at the right part of the screen.



Here click on the leftmost top **‘select button’** on the source code part of the screen. Now whenever you hover over any part of the page, its corresponding source code will highlight.



Now hover over this Email or phone text box and see how it is stored.



Here you will find out it is stored as an **input element.** Let's understand **what an input element in HTML is.** Input element specifies an **input field** where the user can enter data. Here in this field, users can input their *email or phone.*

Now to get this element through our code we need to have some attribute related to this field and then we can search for the element through our program. Here you will see the input field is having an **id as "email"**

With this information, we can search this element on the browser using the method **.find_element_by_id()**. This method finds an element by the id given to it.

Syntax for this method is:

element=browser.find_element_by_id("id_of_the_element")

This method will return an object of the element with id as id_of_the_element.

In our code, to find the text box for Email or Phone Number, run the following line of code.

```
user_box = browser.find_element_by_id("email")
```

Here, **user_box** is the object of the input text box where any user enters its Email ID or Phone number.

Once we get this, we need to type the data for this box. Here to do the same from our program, we have two options. **1st to save the User_ID** and **2nd to take it input from the user.** Let's look at the 2nd option.

To take any input in python, we use the method **input().**

Syntax for this method is:

a=input("Message_to_be_shown_to_the_screen")

Here, the message that you write as the argument will be shown to the user and the method will accept an input from the user. Once the user presses ENTER, the taken input is stored in a.

So to take the USER ID and PASSWORD as input run the following code:

```
user_id=input('Enter User Id of your Fb Account :')
```

```
password=input('Enter the password :')
```

This will store the USER ID and PASSWORD in the variables user_id and password respectively.

Now our task is to send the USER ID to the box whose object is user_box.

To do this, **.send_keys()** function is going to help us. When called with any object, this function will send the keys, given to it as input, as an object.

So to send the USER ID to the element user_box, run the following code.

```
user_box.send_keys(user_id)
```

Nextly we will be doing the same stuff with the **password**. i.e. We will search for the element holding the text box for password and then we will send the password to that box. First of all, let's see how the password text box is stored in the source code. To do this we will be following the same approach as we did in the case of Email ID. And when i did the same thing, this is what the password element looked like.

This was also an **input element with id as ‘pass’.** So we will run the below code to find this box and send the password to it.

```
password_box = browser.find_element_by_id("pass")
password_box.send_keys(password)
```

Ab

After filling up the USER ID and PASSWORD, we will have to click the **Login button.** So now, we need to find the login button. Here also you will be following the same approach. Looking at the source code and hovering over the Login button, this is what i found.

This is another **input element with id = “u_0_b”.**

So to find this we will run the following code.

```
login_box = browser.find_element_by_id("u_0_b")
```

So login_box is actually holding the login button here. Now we need to click this button. And this can be simply done by running the following code.

```
login_box.click()
```

After running the above code, you will be logged in to your FB.

Now the last step is to close the browser. For that simply run:

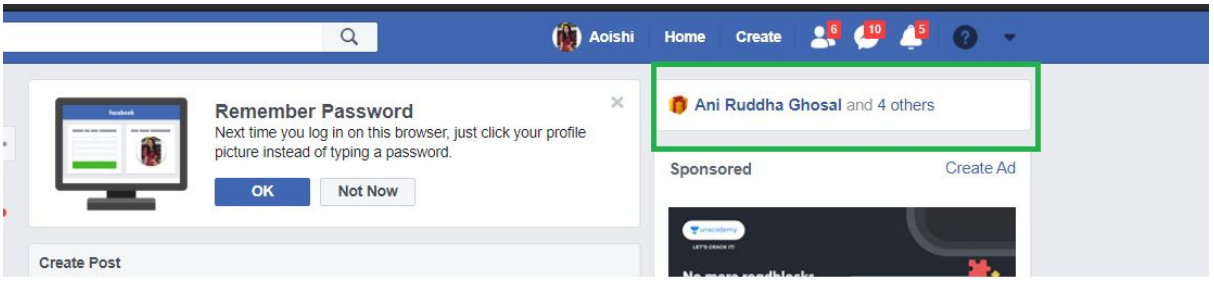
```
browser.quit().
```

Project 4: FB Birthday bot.

In this project we will be making a Facebook Bot which will wish to your facebook friends for their birthday. And again as it is another project on website automation so no human intervention.

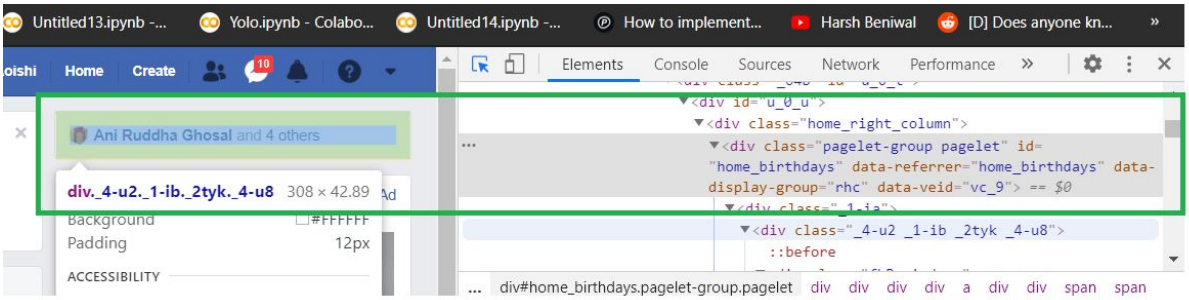
To begin with this, the first thing that we would do is **login to our FB account.** This is what we did in Project 3 so without repeating the same stuff again, I will proceed further.

Once you have logged into your account, then a notification for people whose birthday is today is seen at the top right of your news feed.



So from here you can see **how many people have birthdays today.** For example in the above screenshot, it's showing It's Ani + 4 other's birthday today. So if we can somehow retrieve this number and then add 1 to it so we will be able to find out how many people have birthdays today. So let's find out how this element is stored.

When I opened the source code and hovered over this item, this is what i found.



This is saved as a **div element** with **id 'home_birthdays'.** Now **what is a dive element?**

A div element specifies a **division or section** of an HTML document. It holds other elements. Now out of all these things, I am interested in the text '4others'. When i keep expanding the div element. I see that it is hid inside the elements like this

That means even after finding the top most div element, we will have to iterate to the last span element and extract the text '4 others'.

To iterate, we will have to run the following code:

```
k='//*[@id="home_birthdays"]/div/div/div/div/a/div/div/span/span[2]'

n=browser.find_element_by_xpath(k)
```

Now **n** is the object of the last span element. To extract the text element, we will have to run the following code:

```
n=n.get_attribute('textContent')
```

This will return the text ‘4 others’ and store it to **n.**

Now we just want the number which is present at the first location. So to retrieve this we will run the following code:

```
num=n[0]
```

And to convert it to integer, the following code:

```
num=int(num).
```

This num is now holding the number **4**.

Now we will have to **go to the page** **https://www.facebook.com/events/birthdays/** because this is the place from where we will be sending the message.

To do this, run the following code:

```
browser.get('https://www.facebook.com/events/birthdays/')
```

Now here we can send messages to every person present here. But we don't want to do that right. We just want to send the wishes to those people whose birthday is today. To implement this we are going to apply a simple concept. The number of people having the birthday today is **num+1**. So we will only be sending the message to the 1st num+1(5 in this case) people. This can be controlled by **using a for loop.** The for loop will run for num+1 times and each time the next person will be getting the message starting from the first person.

So now we are clear with how we can send messages to only those whose birthday is today. But we are still to find out what is the procedure to send the message.

To figure that out Let's **go to the page** https://www.facebook.com/events/birthdays/ and look at the source code.

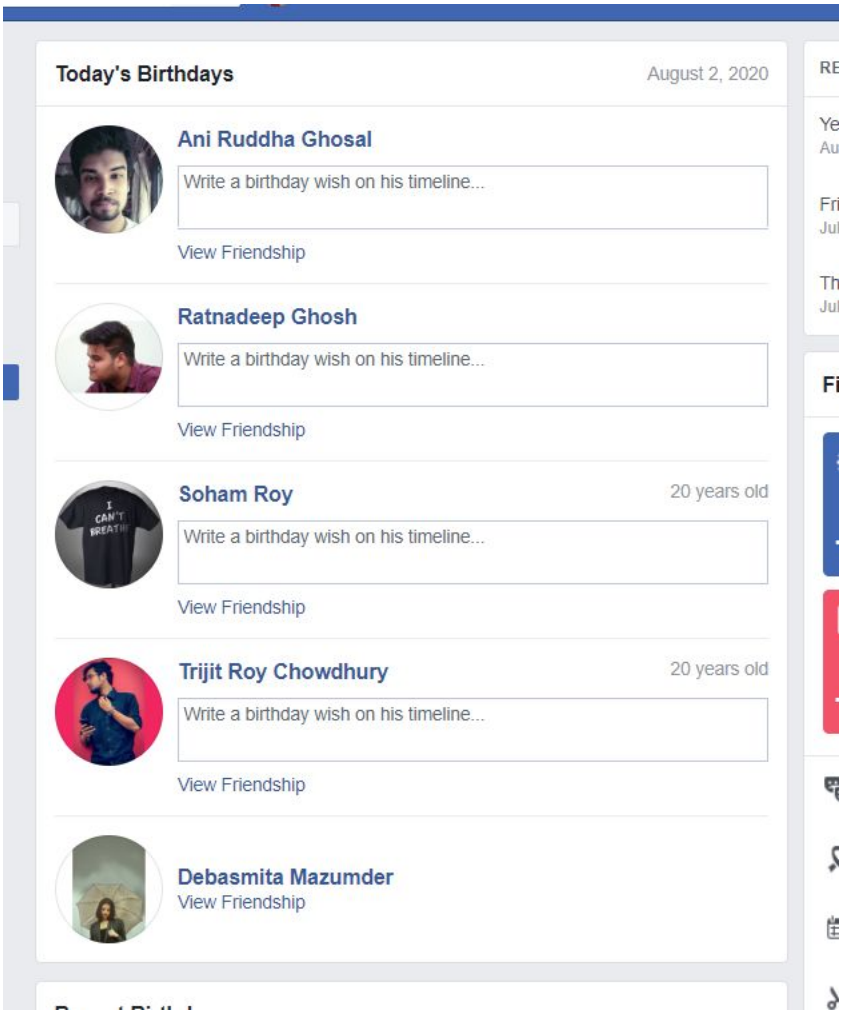
This is how the text field of the write post is stored.

It has a **class name "enter_submit……..textInout"** So we can now find out these text boxes just the way we have been finding out other elements by their Xpath and store them in a list. To do this run the following code.

```
bday_list=browser.find_elements_by_xpath("//*[@class ='enter_submit uiTextareaNoResize uiTextareaAutogrow uiStreamInlineTextarea inlineReplyTextArea mentionsTextarea textInput']")
```

This will be returning the list of all the boxes present there.

But we have already seen how to tackle this. So now we will be running a loop for num+1 times and writing the elements in the box and then sending them. But just look at the boxes once.



None of them has any **button** to send the message. Generally in this case, the message gets sent when we press ENTER. So to do the same here, will be using the class Keys of selenium.webdriver.common.keys.

We can use any one between Keys.ENTER and Keys.RETURN to fulfil this task.

So after getting the list of text boxes we run the following code.

```
c=0
```

```
for element in bday_list:
    element_id = str(element.get_attribute('id'))
    XPATH = '//*[@id ="' + element_id + '"]'
    post = browser.find_element_by_xpath(XPATH)
    post.send_keys(message)
    post.send_keys(Keys.RETURN)
    c=c+1
    if(c>num):
        break
```

Before the for loop, I am initializing the variable c as 0. This for loop is going to run for all the elements of the bday_list which is nothing but all the **'Write the birthday wish on his timeline'** boxes. Inside the for loop we are going to take the id of the boxes and store them in **element_id** as string. Now we will be searching for another element by the XPATH created using the help of ***XPATH = '//*[@id ="' + element_id + '"]'*** . This element is nothing but one particular box from all of them whose object is stored as **post**. Now we will send our message. For this case, the message is already saved as **'Happy Birthday'**. Now to send this, We will use **Keys.RETURN.** That's all what we need to do for sending the wishes.

Now to control this loop so that it only sends the message to those who have birthdays today we have used the variable **'c'.** So after each time the loop has run, we increase the value of **'c' by 1.** So what will happen here, when the loops run for 5 times, c's value becomes 5. Now when it will check the condition **where c>num** it is true, because num was 4, so in this case, when the loop has been run for 5 times, the loop will go inside the if case, where it is encountering a break. Hence stopping the loop.

Project 5: Automatic Wifi Connector bot

In this project, we are going to build a python bot which will connect your device to any saved wifi network.

To do this project, We are going to take help of the **netsh.** Netsh is a **command line scripting utility** which allows you to display or modify the network configuration of a computer.

So whatever task we are going to do, we will be using netsh commands for them.

But running netsh commands from the program is not possible. So we will be taking the help of the function **os.popen().** It lets you run any command given as its argument. Then to read what was returned by that command, we can use **.read()** function with that.

The first task we are going to do here is **finding the saved networks.** To do this, we will be using the command **netsh wlan show profiles.** This shows the currently saved networks. So to find the saved networks, run the following code:

```
saved_profiles = os.popen('netsh wlan show profiles').read()
print(saved_profiles)
```

Here we are going to use the **os.popen()** method. This is used when we want to run any command through our program. **read()** is going to return the values of the output of the command.

Now the second task is **finding the available networks.** Here we are going to use the command **netsh wlan show networks.** This shows the currently available networks. So to find the currently available network run the following code:

```
available_profiles = os.popen('netsh wlan show networks').read()
print(available_profiles)
```

So now, **saved_profiles** and **available_profiles** are holding the values of saved and available networks respectively.

Now we will ask the user to input their preferred choice of network. They are supposed to enter the correct name of the preferred network. We will then save it to some variable. To do this, run the following code:

```
preferred_ssid=input('Enter the preferred SSID for your connection : ')
```

So **preferred_ssid** will be holding the user's choice.

Now at this stage we need to disconnect the current wifi network of the user.(if any).

For this we will be using another netsh command and this time we have **netsh wlan disconnect.** This disconnects the connection of the device with the current network. To do this, run the following code.

```
response = os.popen("netsh wlan disconnect").read()
print(response)
```

After all this, we need to connect with the **preferred_ssid.** But before that we need to check 2 things. **Whether this network is saved or not** and **whether this network is available or not.** If this network is saved then only we will check for its availability. We will keep on checking for this network to be available. Once the network is available, then we will make a connection with it. So let's first check whether this network is saved or not.

To do this, we will have to use **if-else statement.** This statement checks for the truthness of a statement. If that statement is true, something happens, else something else happens.

Syntax of if-else:

```
if(condition):
        do something.
else:
        do some other thing.
```

Here is an **example.**

```
if(a>5):
        print(greater)
```

else:

print(smaller)

Here is how we can use if-else to check whether the preferred_ssid is in saved networks or not.

```
if preferred_ssid not in saved_profiles:
    print("Profile for "+preferred_ssid+" is not saved in system")
    print("Sorry but can't establish the connection")
    sys.exit()
else:
    print("Profile for "+preferred_ssid+" is saved in system")
```

Let's break down the code and understand what are we doing here. First of all we are checking if preferred_id **not in** saved_profiles. Here the condition is preferred_id **not in** saved_profiles. This condition will be true when the **preferred_id is not in saved_profiles.** When this is the case, we are printing the message, and running the code **sys.exit().** This will stop the code execution right at this moment. And nothing further will be executed. That means If the preferred_id is not in saved_profiles then we will not run the code any further.

But if this condition is false, then instead of entering into the *if case*, The code will enter into the *else case.* Where we are just printing the message.

Because of this block of code, we are making sure that the code is not connecting for those networks which are not saved. In other ways, we are making sure the connection is made with saved networks only.

After this it is now the turn to check for network availability. To do that we are going to run the following code.

```
while True:
    avail = os.popen('netsh wlan show networks').read()
    if preferred_ssid in avail:
    print('Found')
    break
```

Here using a while loop we are running an infinite loop. Inside this loop we are checking for the *currently* available network. If we find the preferred network to be available then we break this loop. Breaking a

loop means coming out of the loop. But if the preferred network is not available then it will keep on checking this condition. This will keep on running until the network is available. This simply is going to hold the code inside this loop until the network is available.

Now the code will come out this loop only if the network is available. If the network is available, then we will connect with this network. The command for doing this is To do this we will run the following code. *netsh wlan connect name='+""+preferred_ssid+"".* To do this, run the following code.

```
resp = os.popen('netsh wlan connect
name='+""+preferred_ssid+"").read()
```