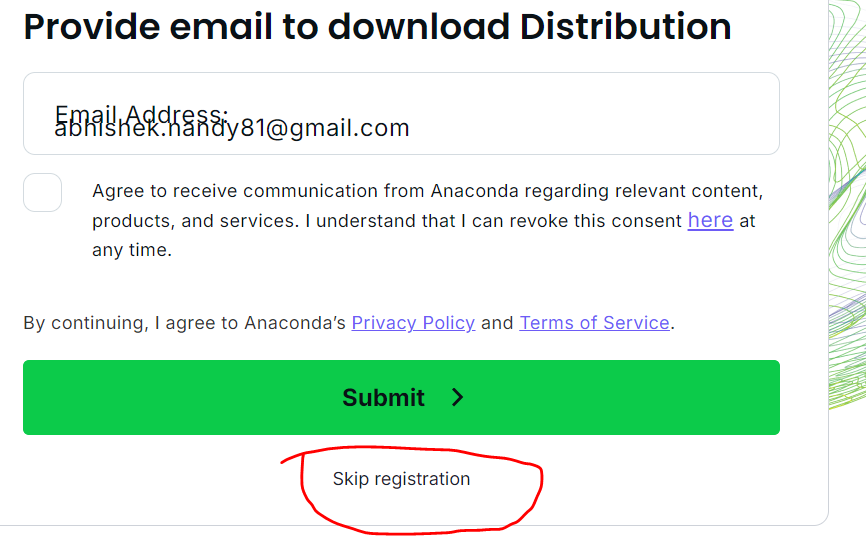
How to download Anaconda

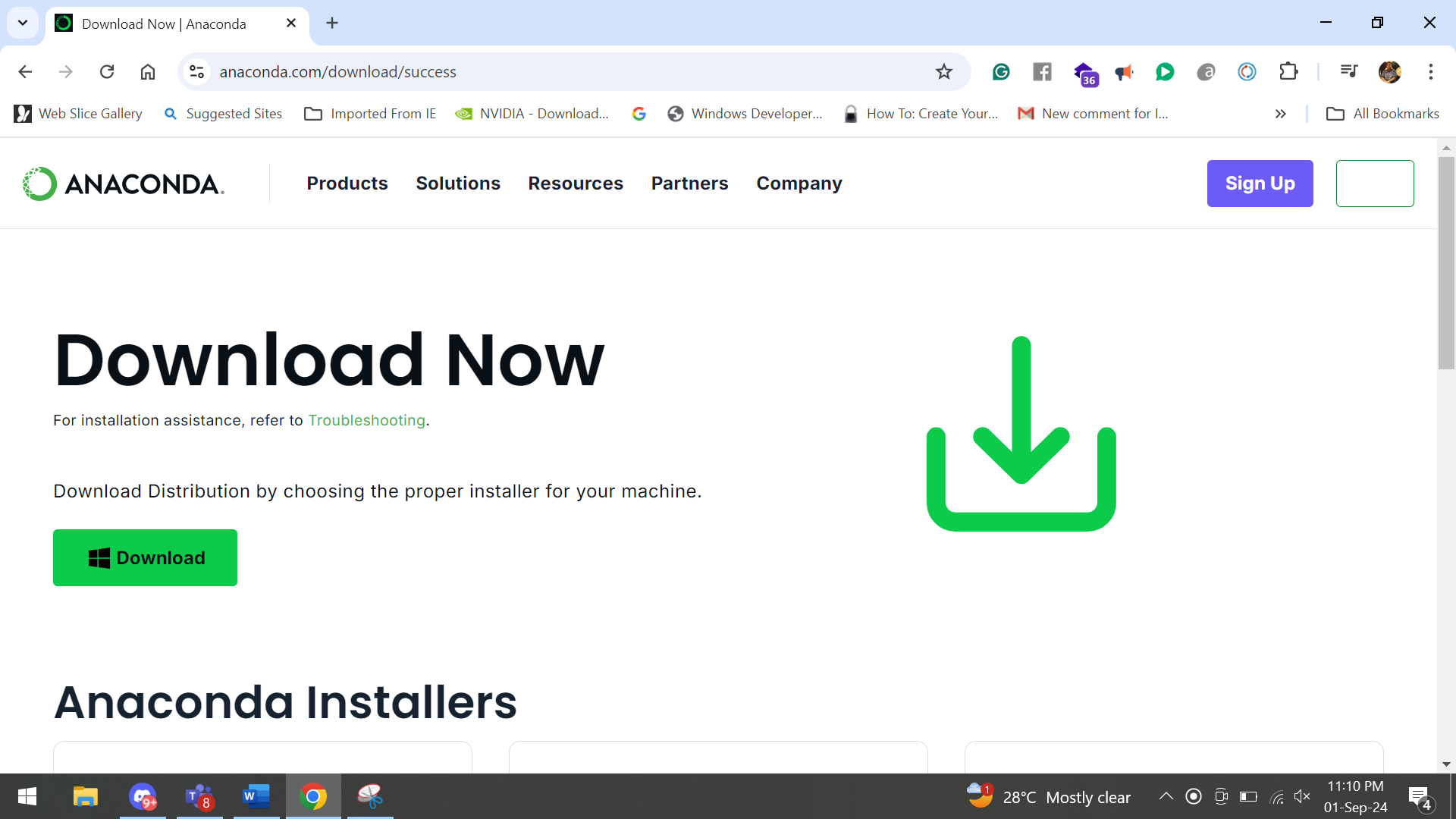
Link to download

<https://www.anaconda.com/download>

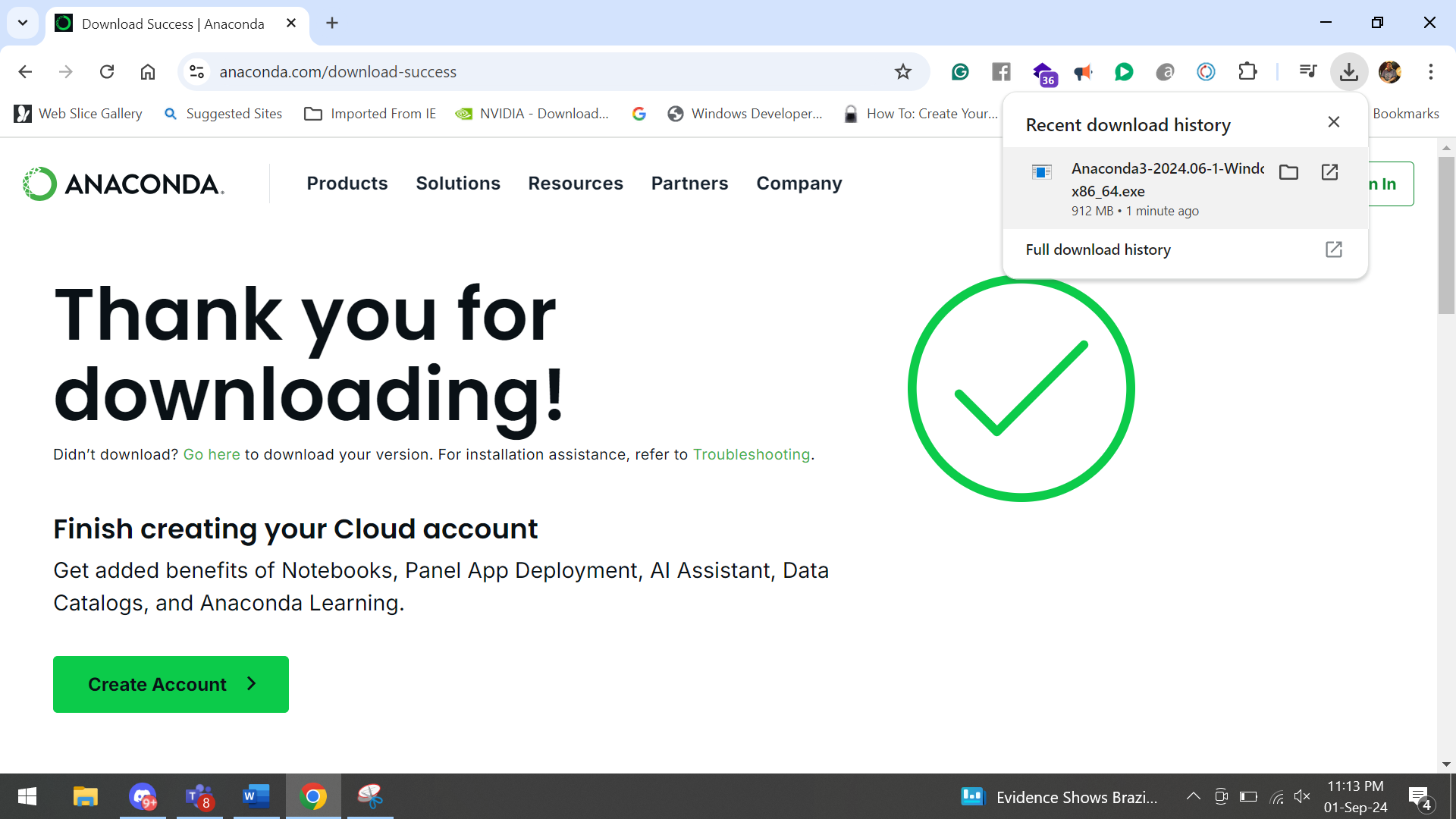
Skip the registration



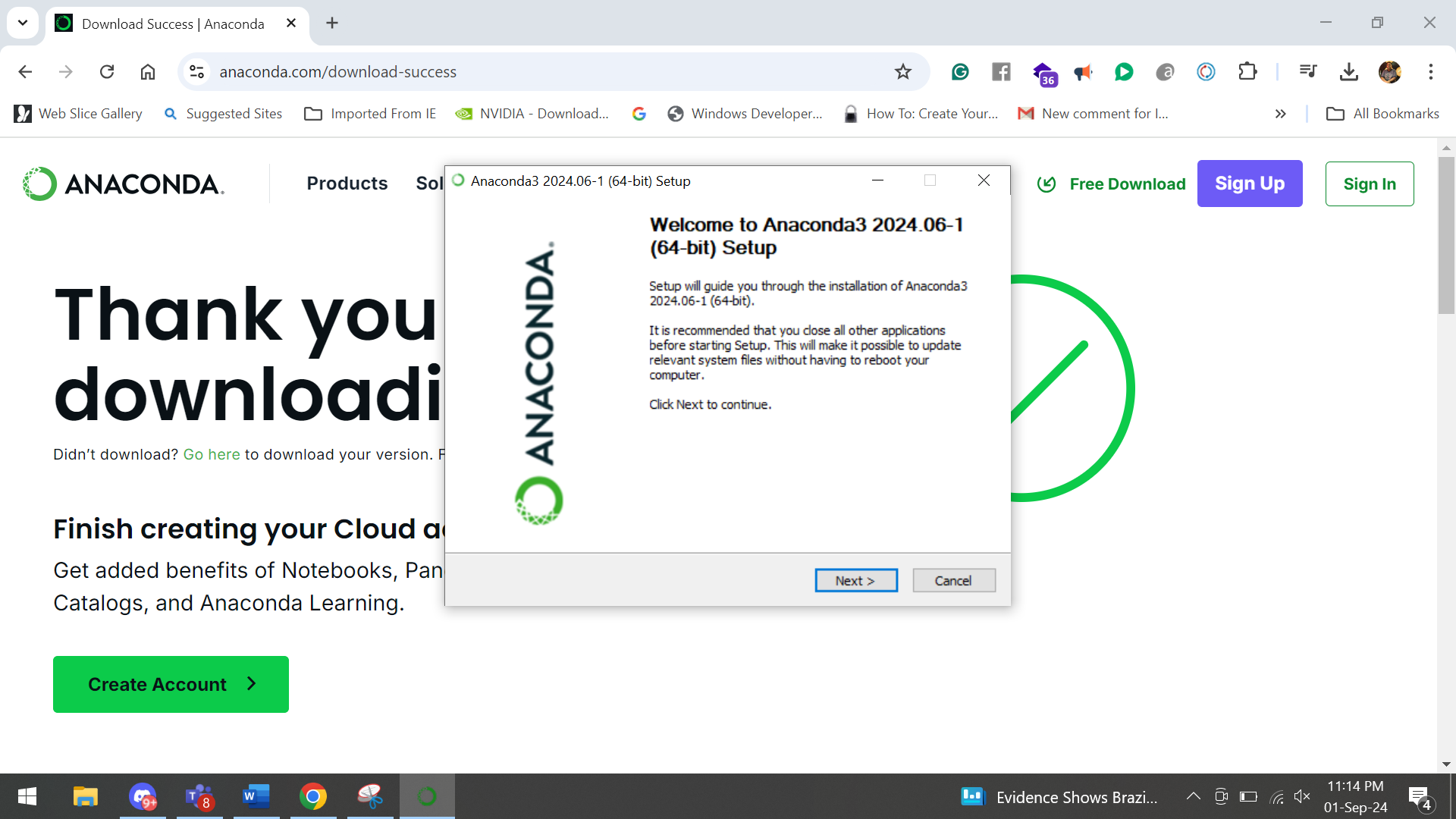
Download Windows version.



Find the downloaded file double click on it to install it.

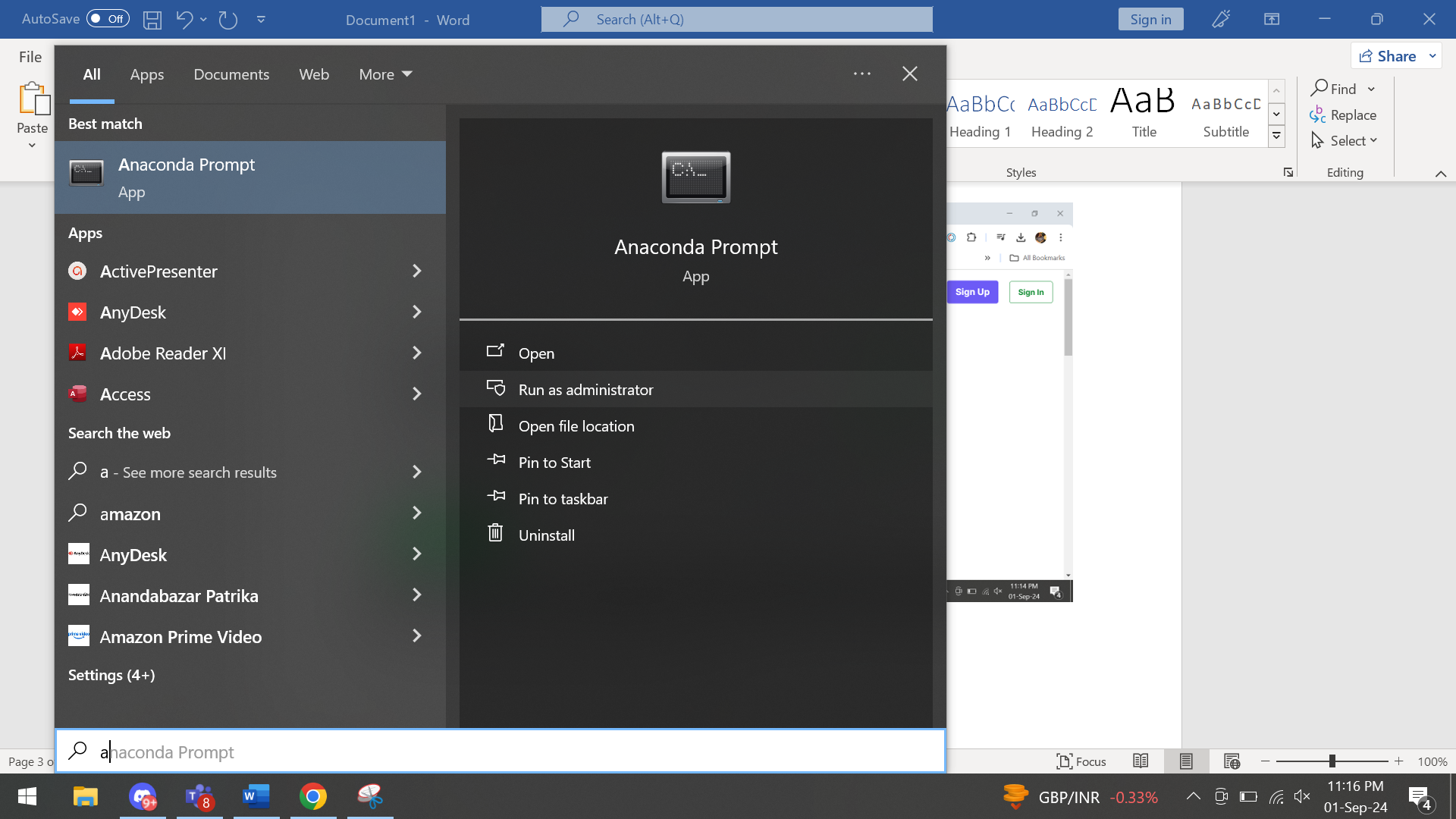


Install Anaconda



Follow the steps to complete the installation.

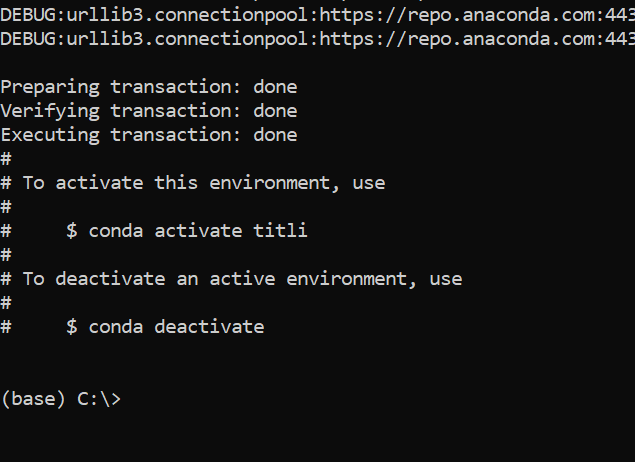
After installation is complete find the Anaconda prompt .



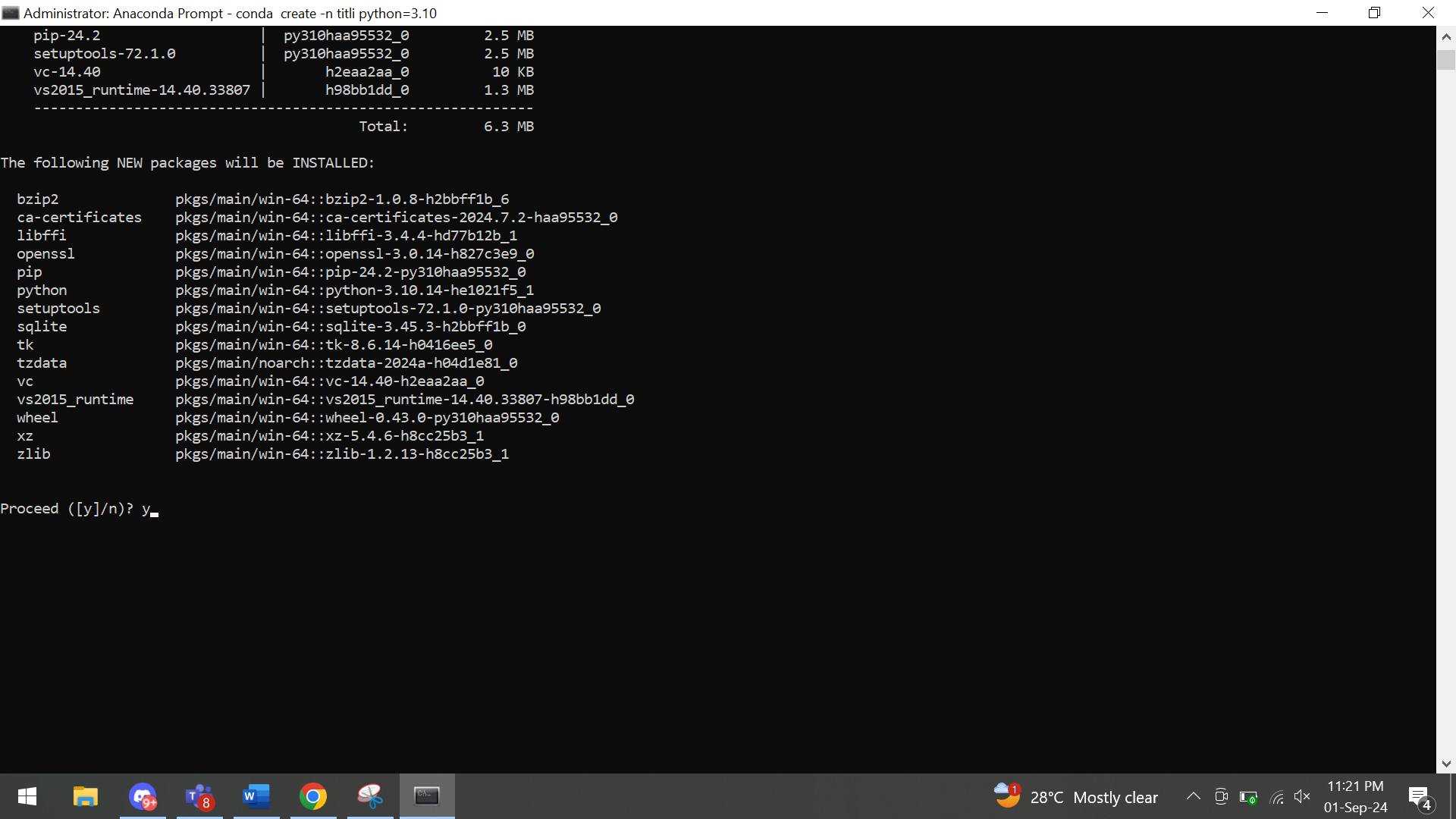
Run in Administrator mode

Run the following command to create a new environment.

conda create -n titli python=3.10



After installation happens click on yes



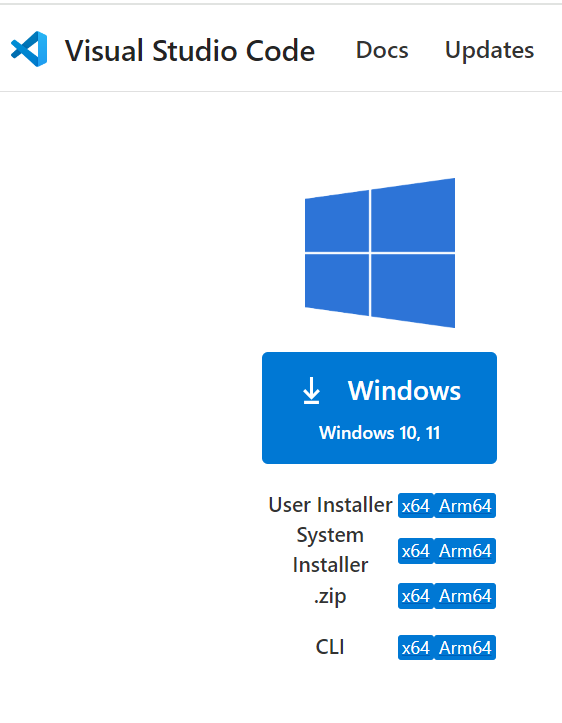
To activate the environment use

conda activate titli

To deactivate use command conda deactivate

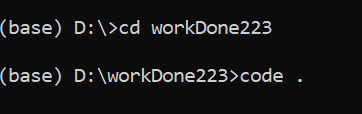
Let’s download Visual Studio Code

<https://code.visualstudio.com/download>



Create a Folder named Work or according to your liking

We will go inside the folder and do our coding exercise.



Using code . opens visual studio code inside the folder.

We have a simple dataset with two columns, x and y. The x column represents a sequence of values from 0 to 100, and the y column contains cumulative sums of normally distributed random values. Let's create an interactive visualization using Streamlit.

Here’s how you can set up a basic Streamlit app to visualize this data with both a scatter plot and a line chart:

**Step-by-Step Streamlit App**

1. **Install Streamlit** (if you haven’t already):

pip install streamlit

1. **Create a new Python script**, say simple\_visualization.py, and add the following code:

python

Copy code

import streamlit as st

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Generate a simple dataset

np.random.seed(42)

data = pd.DataFrame({

'x': np.linspace(0, 100, 200),

'y': np.random.normal(loc=0, scale=1, size=200).cumsum()

})

# Streamlit App

st.title('Simple Data Visualization with Streamlit')

st.write('This app visualizes a simple dataset using a scatter plot and a line chart.')

# Scatter Plot

st.subheader('Scatter Plot')

fig, ax = plt.subplots()

ax.scatter(data['x'], data['y'], alpha=0.7)

ax.set\_xlabel('X')

ax.set\_ylabel('Y')

st.pyplot(fig)

# Line Chart

st.subheader('Line Chart')

st.line\_chart(data)

# Interactive Filtering

st.subheader('Interactive Data Filter')

filter\_x = st.slider('Select X range', min\_value=float(data['x'].min()), max\_value=float(data['x'].max()), value=(0.0, 100.0))

filtered\_data = data[(data['x'] >= filter\_x[0]) & (data['x'] <= filter\_x[1])]

st.write('Filtered Data', filtered\_data)

# Re-display filtered scatter plot

st.subheader('Filtered Scatter Plot')

fig, ax = plt.subplots()

ax.scatter(filtered\_data['x'], filtered\_data['y'], alpha=0.7)

ax.set\_xlabel('X')

ax.set\_ylabel('Y')

st.pyplot(fig)

1. **Run your Streamlit app** with the following command in your terminal:

streamlit run simple\_visualization.py

**Explanation of the Code**

* **Title and Descriptions**: st.title() and st.write() are used to display the title and description of the app.
* **Scatter Plot**: Created using Matplotlib to visualize the relationship between x and y.
* **Line Chart**: Streamlit’s st.line\_chart() function provides a quick line chart of the data.
* **Interactive Filtering**: A slider is added to allow users to filter the data by the x value range. The filtered data is displayed and plotted.

Do the exercise till this point and we will continue after that.