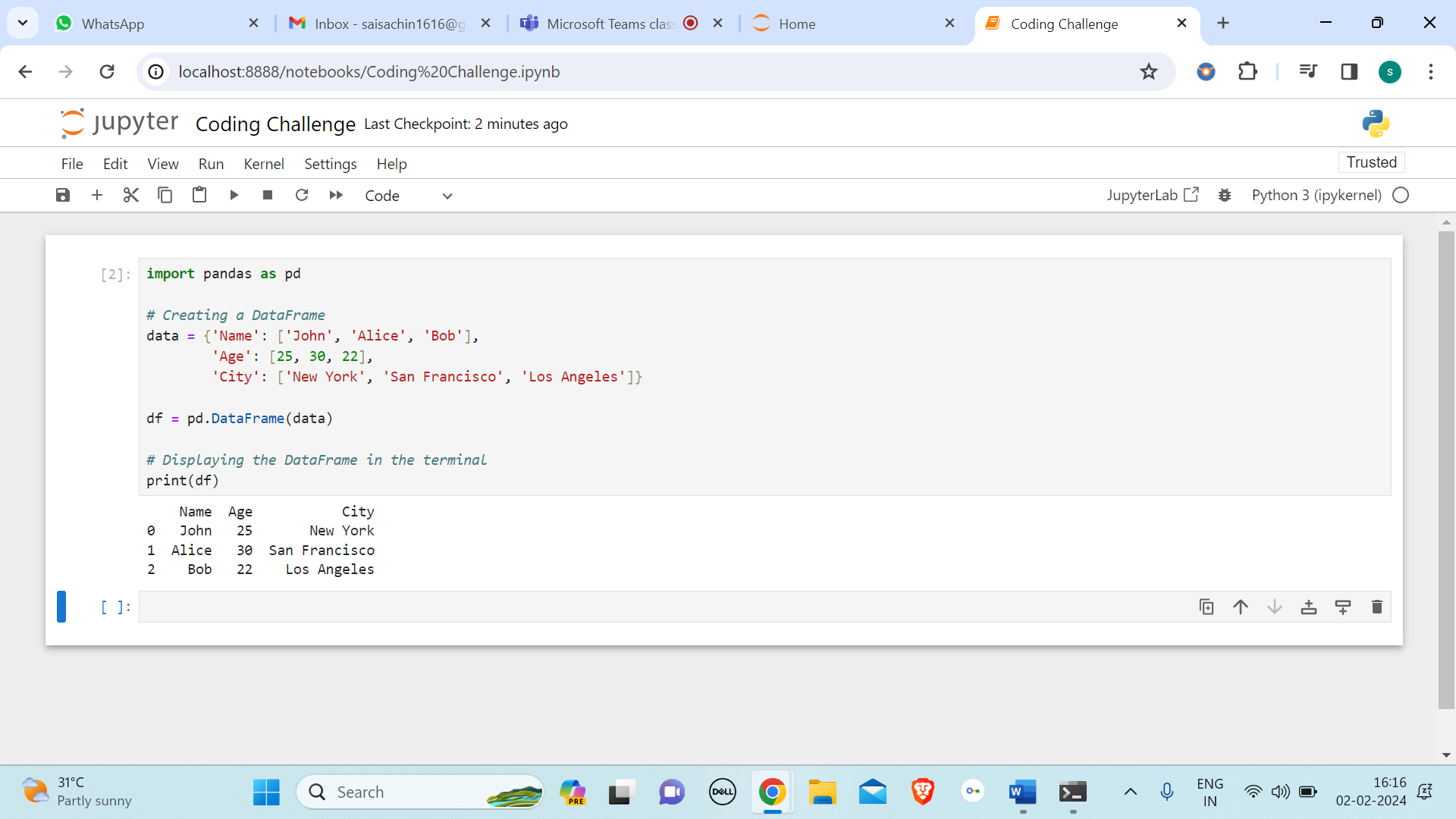
**Explain Pandas for Data Processing**

Pandas is a popular open-source Python library used for data manipulation and analysis. It provides data structures like DataFrame, which are powerful tools for working with structured data.

DataFrame:

* Two-dimensional labeled data structure.
* Consists of rows and columns.
* Can be thought of as a spreadsheet or SQL table.

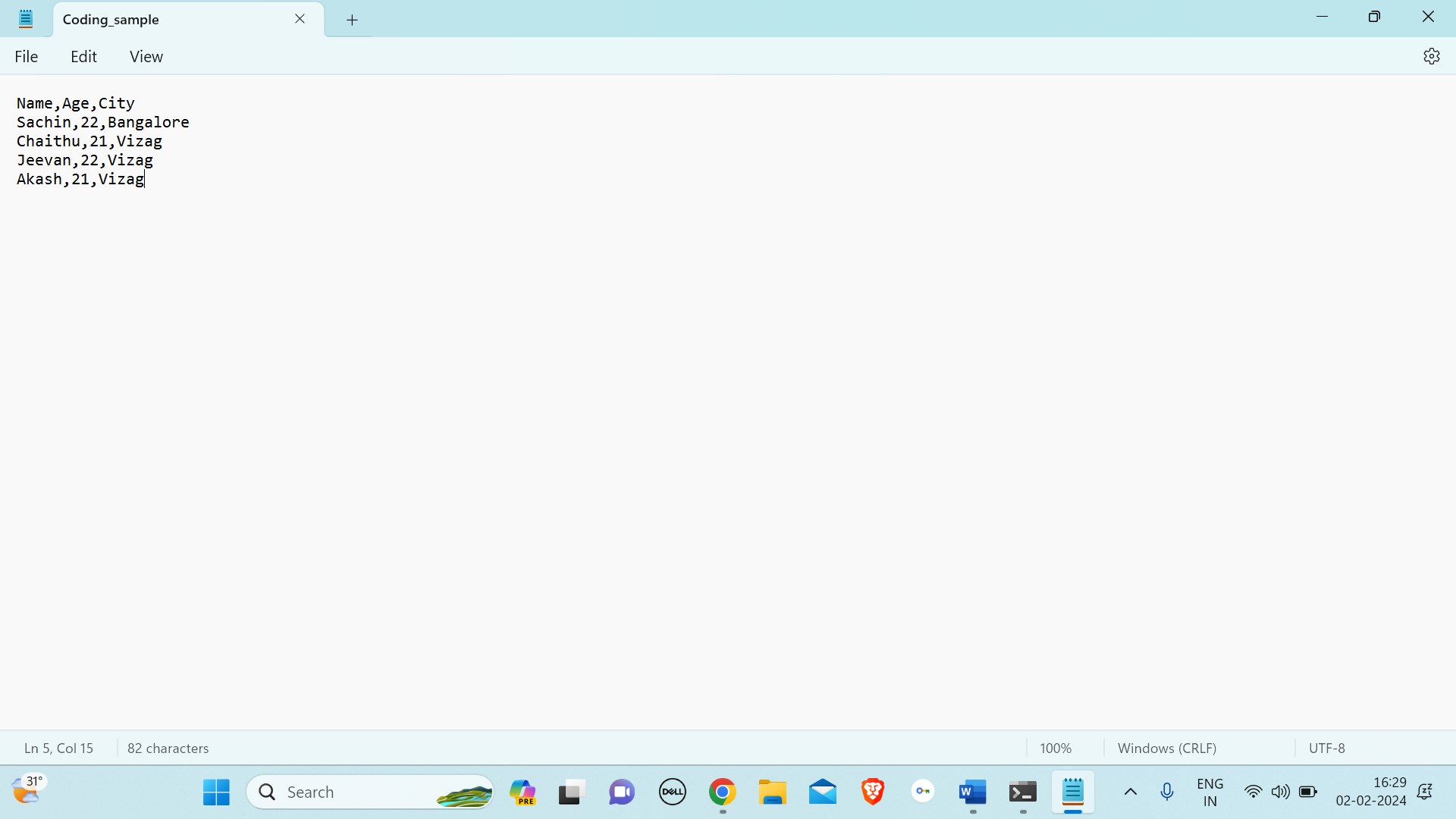


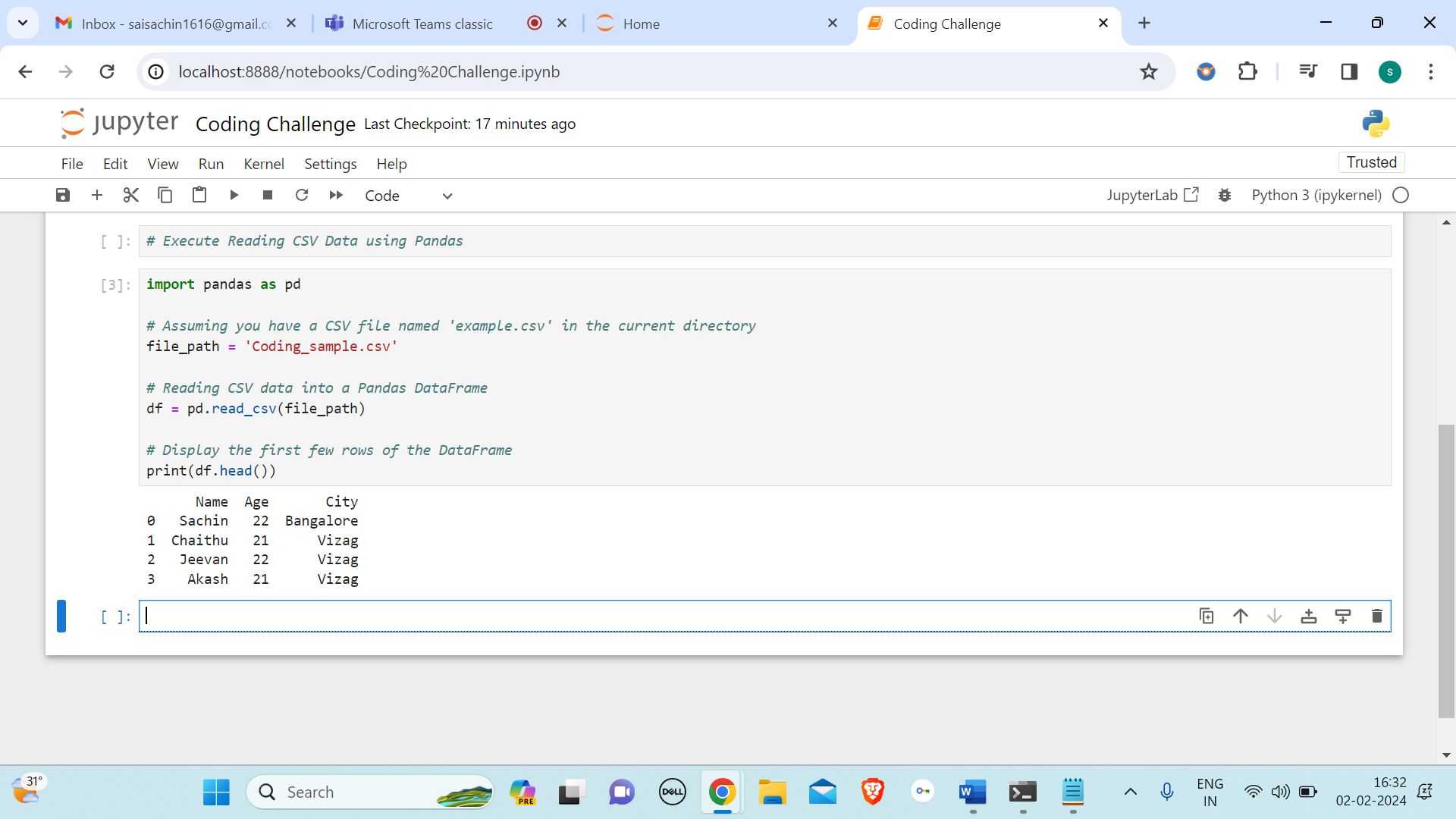
In the above example code uses the Pandas library in Python to create and display a table-like structure called a DataFrame. The code starts by importing the Pandas library and giving it a shorthand name **'pd'** for easier use. Next, a dictionary named **'data'** is created, acting like a blueprint for the DataFrame. This dictionary has keys ('Name', 'Age', 'City') representing columns, and the corresponding values are lists containing data for each column. Then, the Pandas library is used to convert this dictionary into a DataFrame using the line **df = pd.DataFrame(data)**. The resulting DataFrame (**'df')**, organizes the data into a neat tabular structure with rows and columns. Finally, **print(df)** is used to display this DataFrame in the terminal. When executed, you'll see a table with columns for 'Name', 'Age', and 'City', showing the provided information for each person. It's a concise way to organize and view structured data in Python.

**Execute Reading CSV Data using Pandas**

To execute reading CSV data using Pandas, you can use the **read\_csv** function. (CSV stands for comma-separated values). In the below code the read\_csv function automatically detects the delimiter and other parameters, making it convenient for reading CSV files. The resulting DataFrame will contain the data from the CSV file.

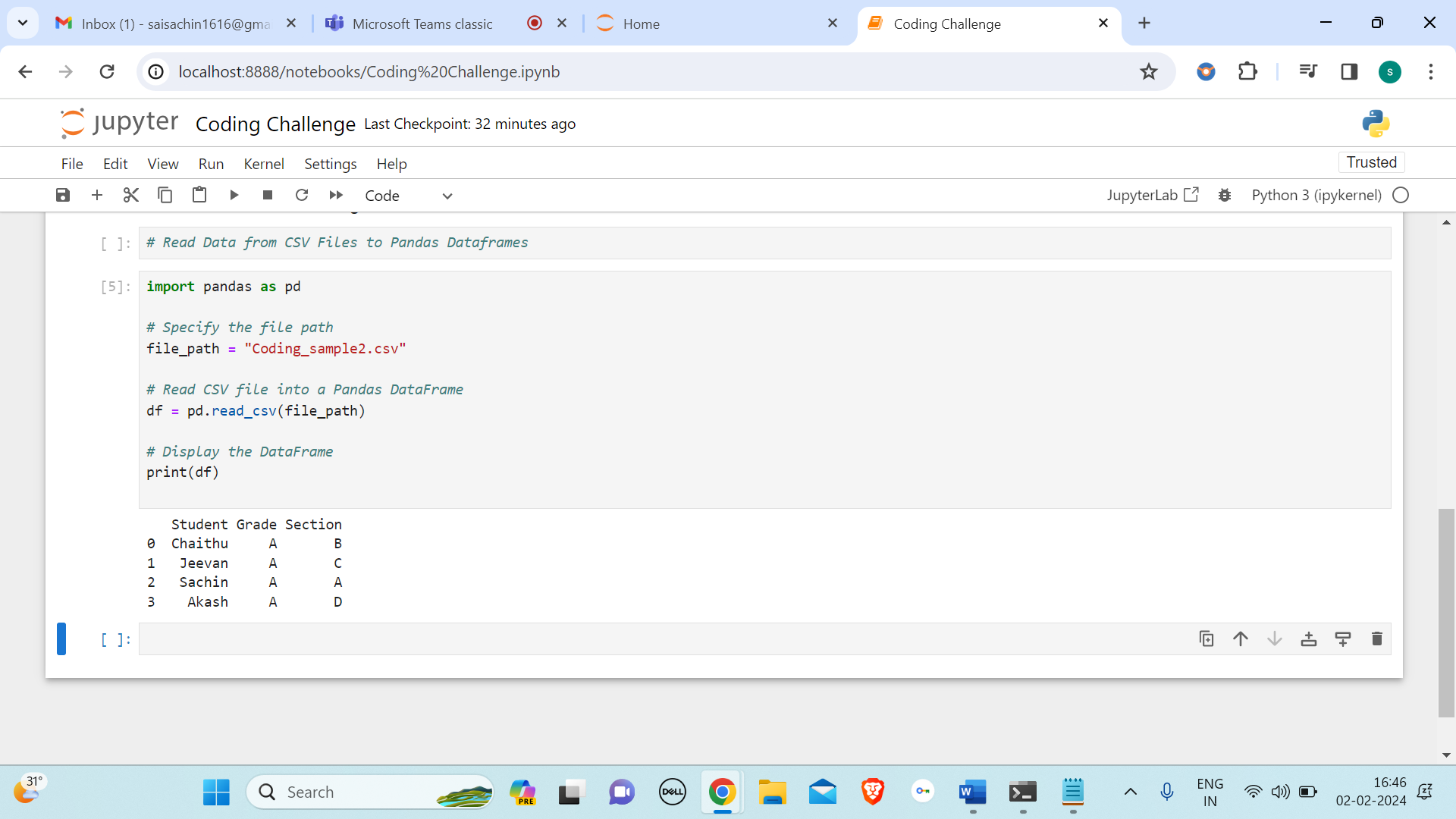
The below image shows us the csv file which I have created and named it as **‘Coding\_sample.csv’**





From the above image, code takes information from a CSV file and puts it into a table called a DataFrame using Python's Pandas library.

**Read Data from CSV Files to Pandas Dataframes**



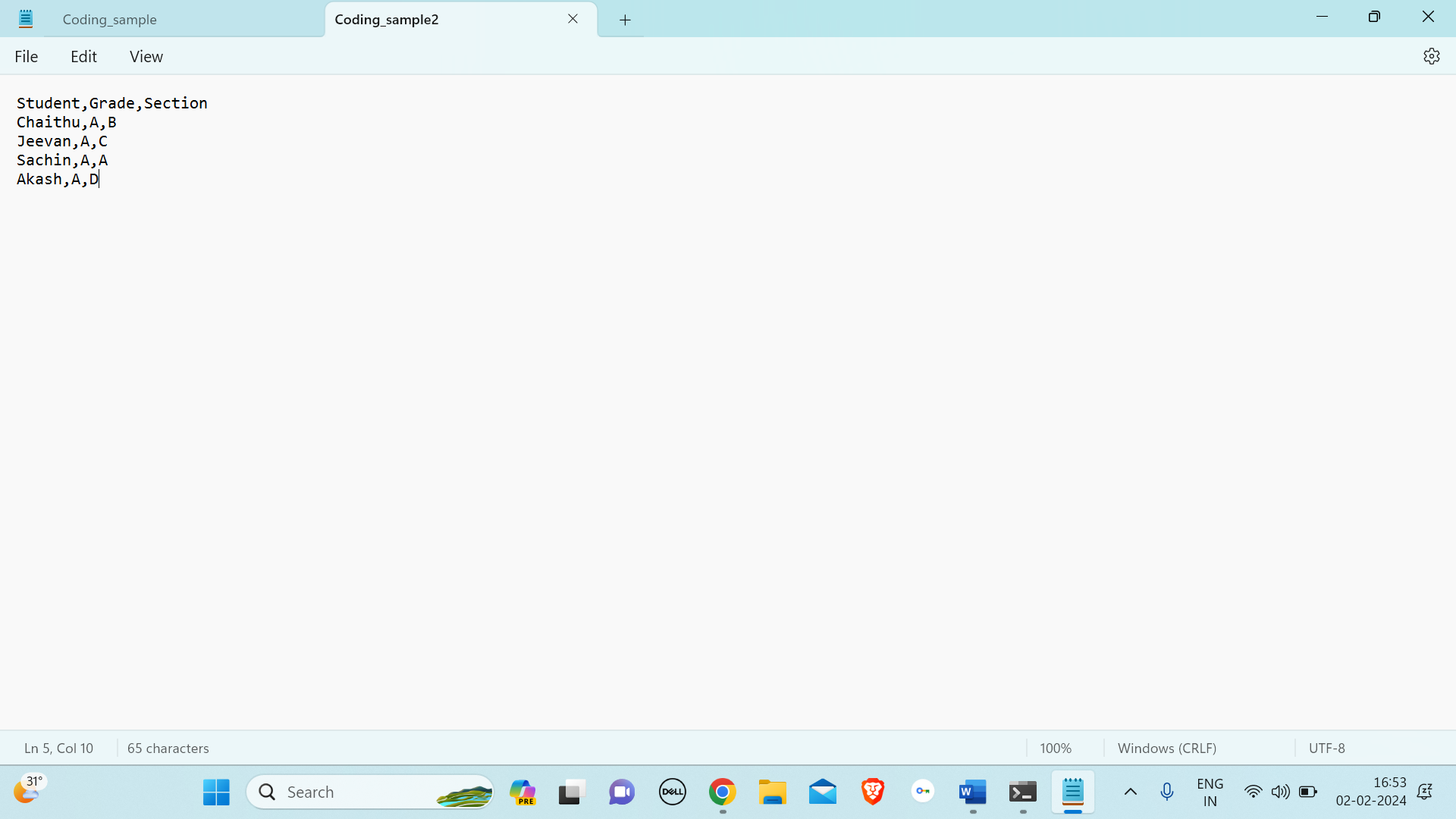
In this code demonstrates how to read data from CSV files and store it in a Pandas DataFrame. In the code, the first step involves importing the Pandas library using the import pandas as pd statement. This is a common practice, and the alias **'pd'** simplifies subsequent code.

Next, a file path is specified using **file\_path = "Coding\_sample2.csv"**, indicating the location of the CSV file named **"Coding\_sample2.csv"** This file path is crucial for Pandas to locate and read the CSV data.

The main action takes place with the **pd.read\_csv(file\_path)** line. Here, the **read\_csv()** function from Pandas is utilized to read the CSV file and create a DataFrame, denoted as **'df.'** The DataFrame is essentially a table that holds the data in a structured manner, making it easy to work with and analyze.

Finally, the code prints the DataFrame to the console using **print(df)**. This step visually displays the data in a table format, allowing you to inspect the contents of the CSV file.

The below image indicates the stored information in **“Coding\_sample2.csv”.**



**Filter Data in Pandas Dataframe using query.**

The **query** method in pandas allows you to filter data in a DataFrame using a query expression. In the below example, the query expression **is 'Age > 25 and Salary > 50000'**, which means we are selecting rows where the age is greater than 25 and the salary is greater than 50000.

We should remember that the column names should not contain spaces or special characters when using the query method directly. If your column names have spaces or special characters, you can enclose them with backticks in the query expression.

