**Sentiment Analysis of Amazon Product**

**Why**

1. Amazon hosts **millions of product reviews.** Manually reading them all is impossible — sentiment analysis helps **automatically determine** how customers feel (positive, negative, or neutral) about a product.
2. Track daily/weekly sentiment trends
3. Detect sudden spikes in negative feedback (e.g., after a defect batch

**What**

1. **Sentiment analysis** is a type of **Natural Language Processing (NLP)** where a computer system reads and interprets the **emotions or opinions** in written text
2. Helps businesses, customers, and researchers **understand how people feel** about products without reading every review.
3. This Application is a web Application which can be access over a LAN.

**How**

**Backend :**

**Data Collection : Google Sheets with Python**

**Data Organization : Pandas**

**Data Analysis :NLTK,vaderSentiment**

**Data Visualization : Plotly**

**Frontend:**

**Google Form**

**Web Application :Streamlit**

**Google Account**

**Google Project**

**Enable google sheet API**

**Create a consent application**

**Download credentials**

**CODE**

**import streamlit as st**

**from google\_auth\_oauthlib.flow import InstalledAppFlow**

**from googleapiclient.discovery import build**

**from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer**

**import pandas as pd**

**import plotly.express as px**

**st.title("Amazon Product Feedback Sentiment Analysis Application ")**

**choice=st.sidebar.selectbox("My Menu",("Home","Video","Analysis","Result"))**

**if(choice=="Home"):**

**st.image("https://miro.medium.com/v2/resize:fit:1200/1\*dSJyqhRaqcVFi3geHLHroA.png")**

**st.write("Sentiment analysis is the process of identifying and extracting emotions or opinions from text, typically using natural language processing (NLP), text analysis, and computational linguistics..")**

**elif(choice=="Video"):**

**st.video("https://youtu.be/6L6isZV6Etw")**

**elif(choice=="Analysis"):**

**sd=st.text\_input("Enter the google\_sheet ID")**

**p=st.text\_input("Enter the range between first column to last column")**

**b=st.text\_input("Enter the column name that is to be analyzed")**

**btn=st.button("start")**

**if(btn):**

**if "cred" not in st.session\_state:**

**f=InstalledAppFlow.from\_client\_secrets\_file("key.json",["https://www.googleapis.com/auth/spreadsheets"])**

**st.session\_state["cred"]=f.run\_local\_server(port=0)**

**mymodel=SentimentIntensityAnalyzer()**

**service=build("Sheets","v4",credentials=st.session\_state["cred"]).spreadsheets().values()**

**k=service.get(spreadsheetId=sd,range=p).execute()**

**a=k["values"]**

**df=pd.DataFrame(data=a[1:],columns=a[0])**

**l=[]**

**for i in range (0,len(df)):**

**t=df.\_get\_value(i,b)**

**pred=mymodel.polarity\_scores(t)**

**if(pred["compound"]>0.5):**

**l.append("postive")**

**elif(pred["compound"]<-0.5):**

**l.append("negative")**

**else:**

**l.append("netural")**

**df["sentiment"]=l**

**df.to\_csv("resultsfront.csv",index=False)**

**st.subheader("Done")**

**st.subheader("The Analysis result")**

**elif(choice=="Result"):**

**df=pd.read\_csv("resultsfront.csv")**

**choice2=st.selectbox("Choose Visualization",("NONE","DATA","PIE CHART","HISTOGRAM","SCATTER\_PLOT"))**

**if(choice=="DATA"):**

**df=pd.read\_csv("resultsfront.csv")**

**st.dataframe(df)**

**elif(choice2=="PIE CHART"):**

**posper=(len(df[df["sentiment"]== "positive"])/len(df))\*100**

**negper=(len(df[df["sentiment"]== "negative"])/len(df))\*100**

**netper=(len(df[df["sentiment"]== "netural"])/len(df))\*100**

**fig=px.pie(values=[posper,negper,netper],names=["postive","negative","netural"])**

**st.plotly\_chart(fig)**

**elif(choice2=="HISTOGRAM"):**

**k=st.selectbox("chose column",df.columns)**

**fig=px.histogram(x=df[k],color=df["sentiment"])**

**st.plotly\_chart(fig)**

**elif(choice2=="SCATTER\_PLOT"):**

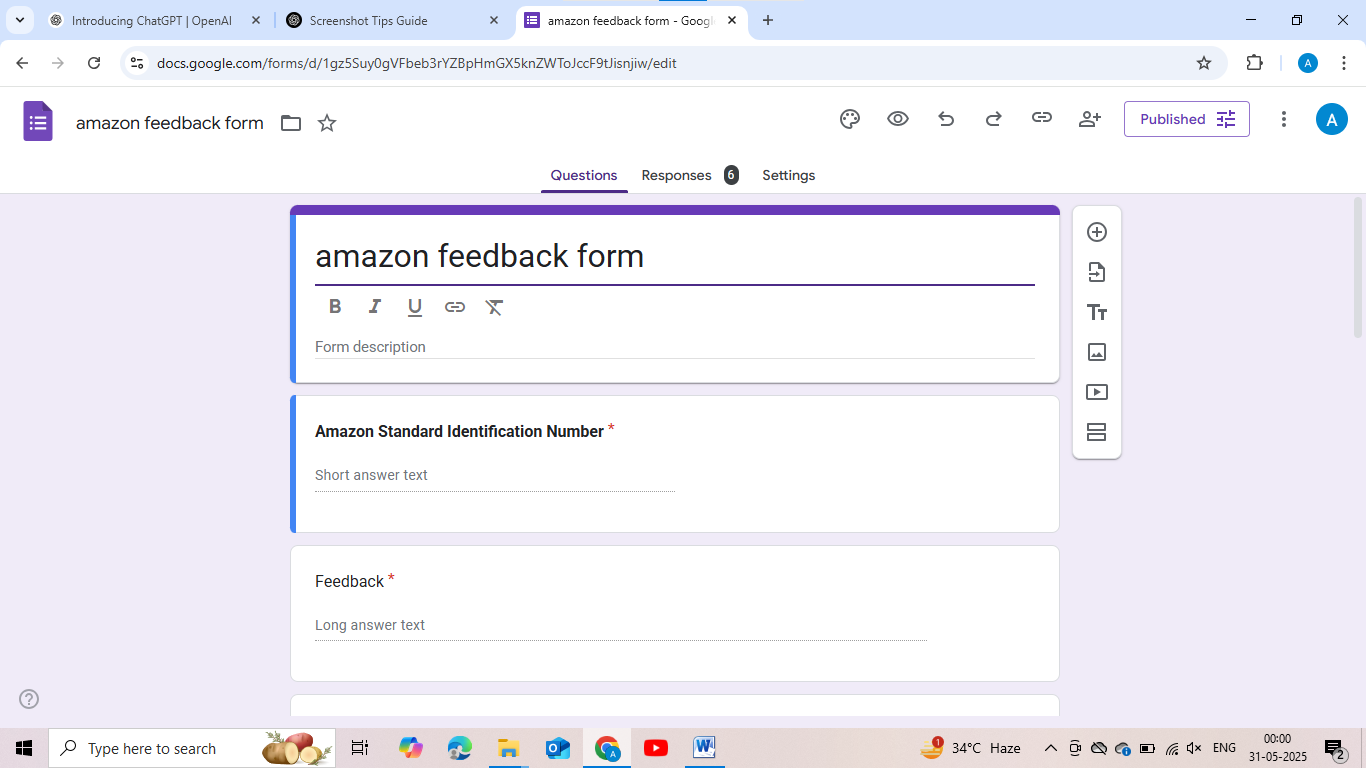
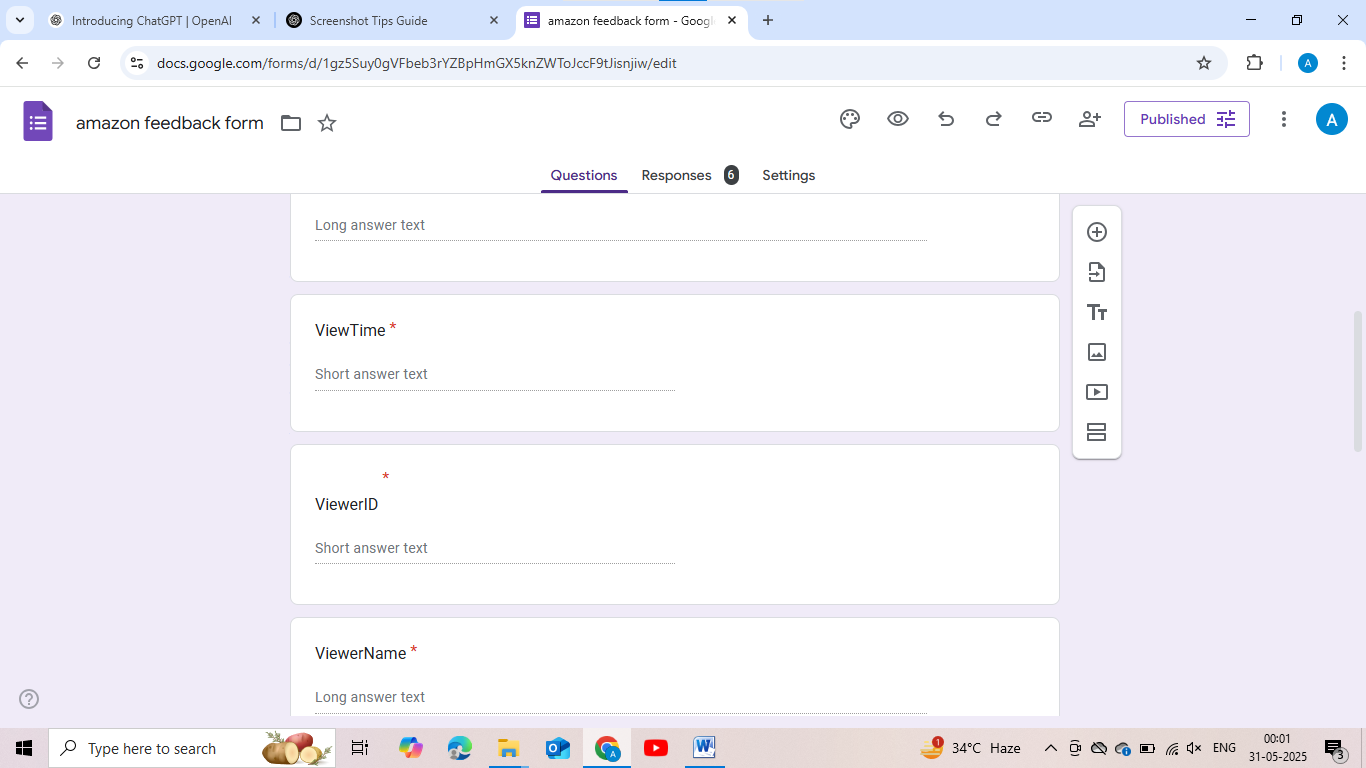
**s=st.text\_input("enter the continous column name")**

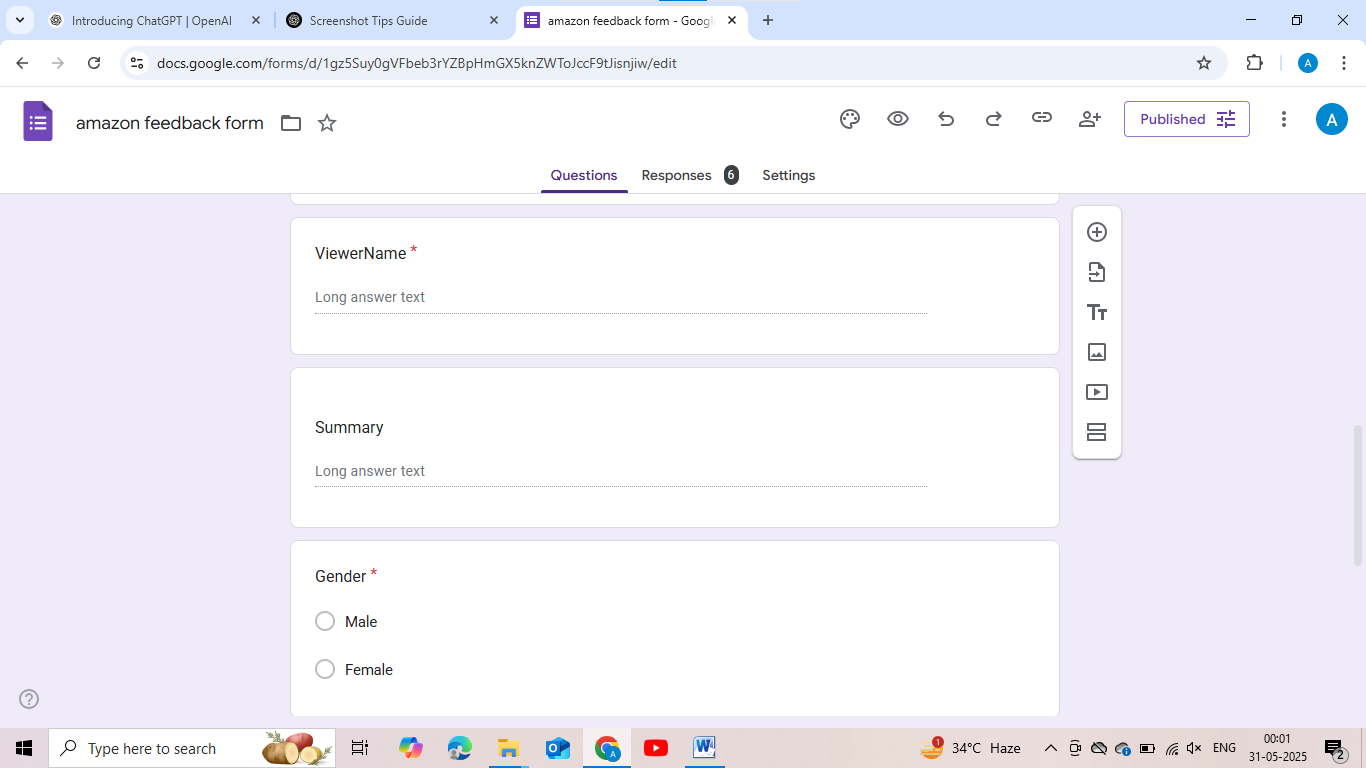
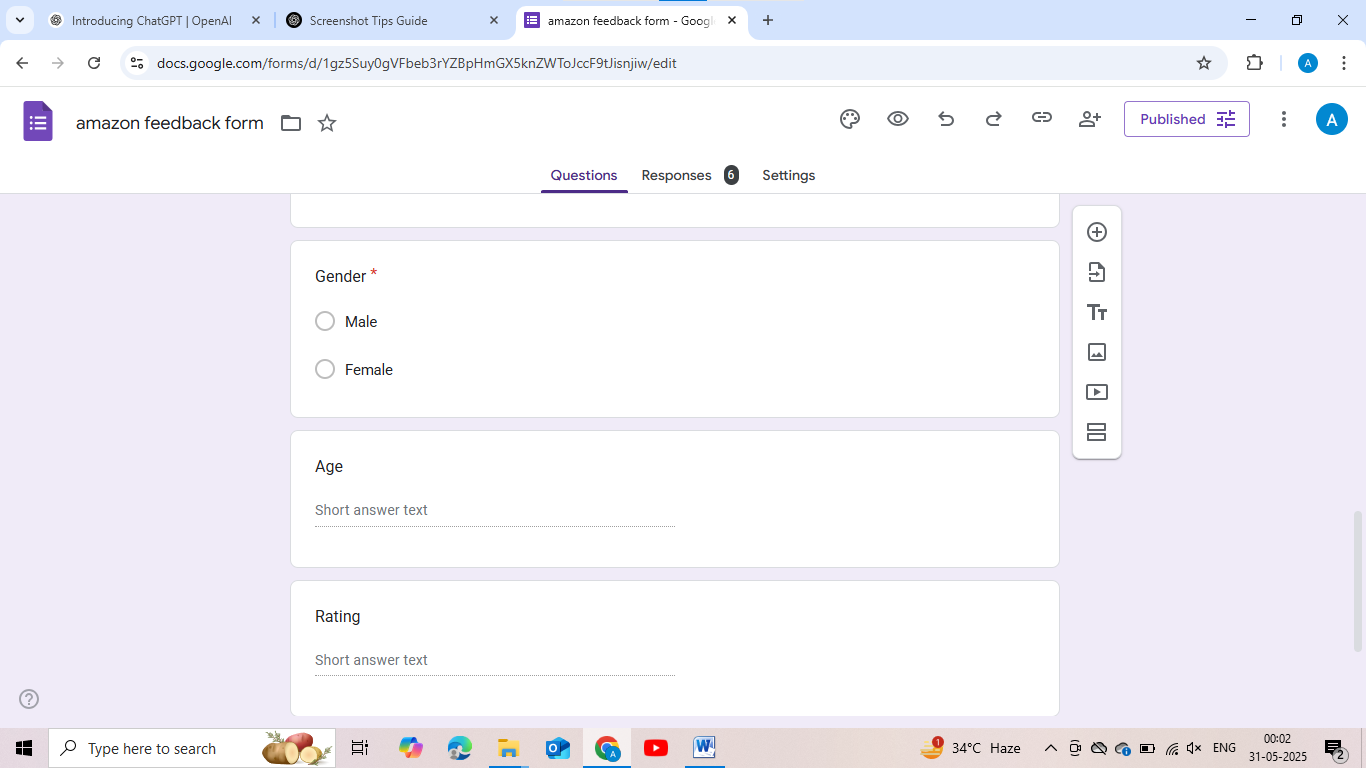
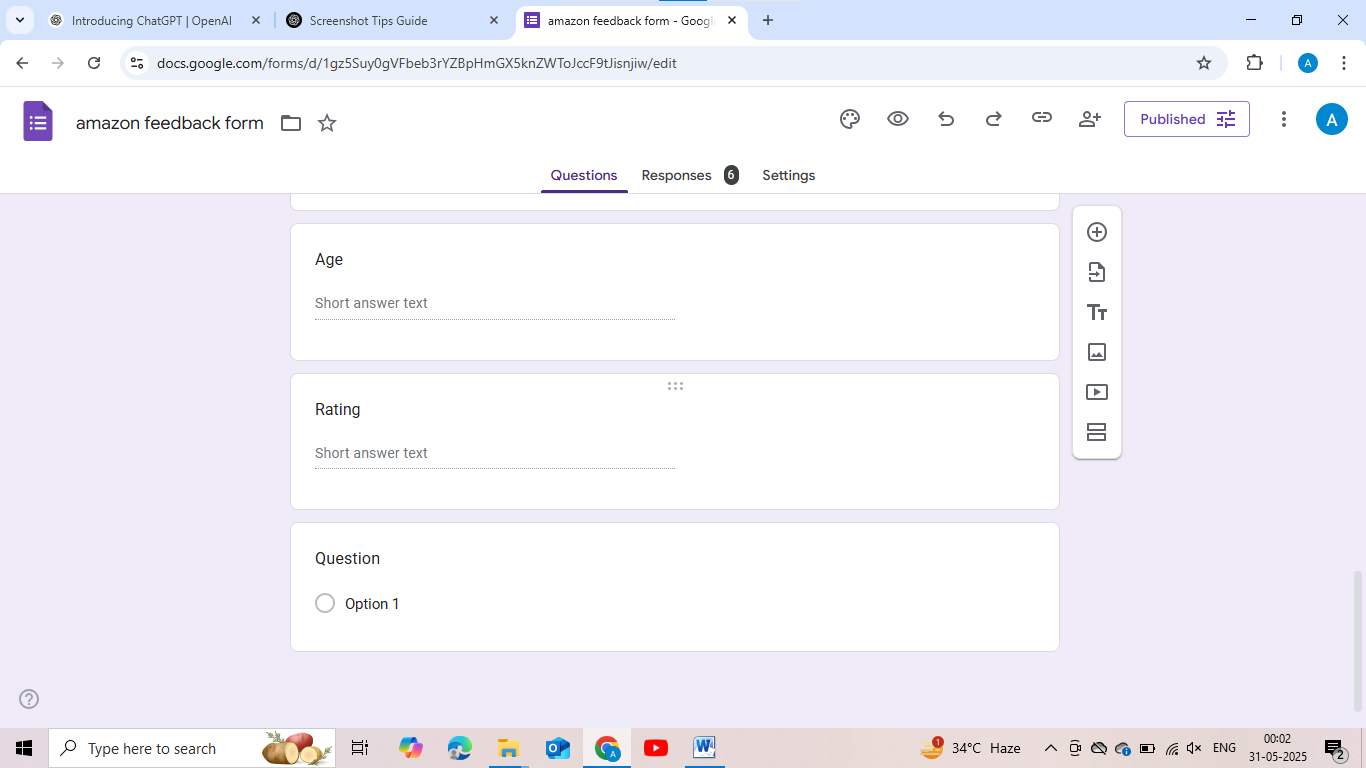
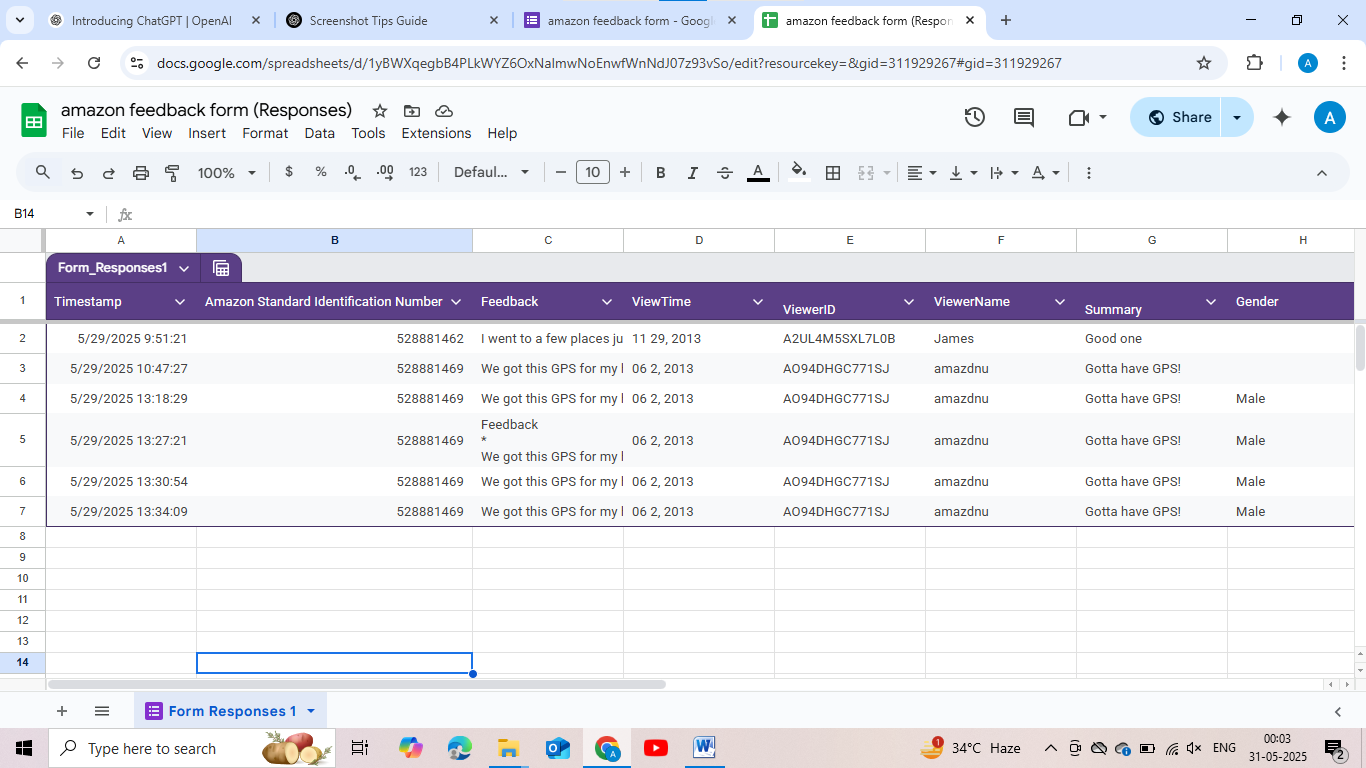
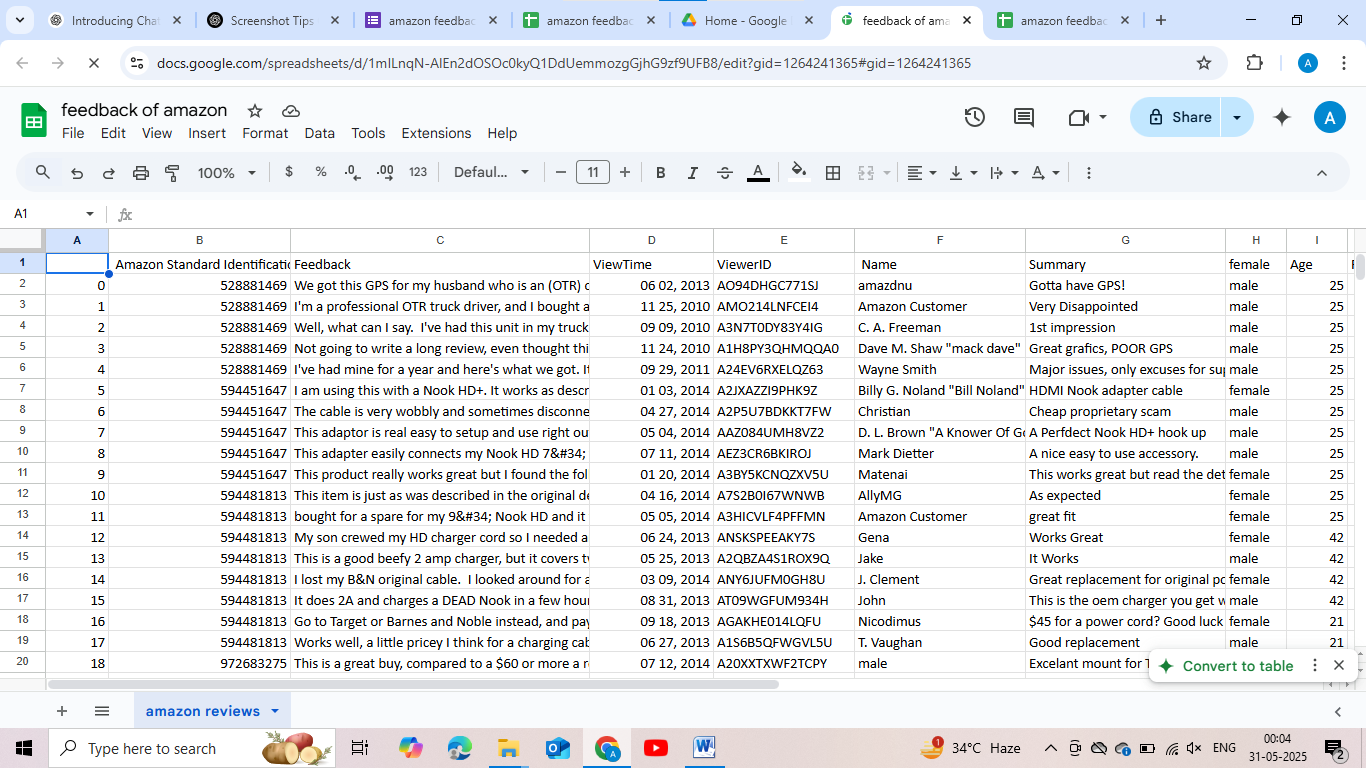
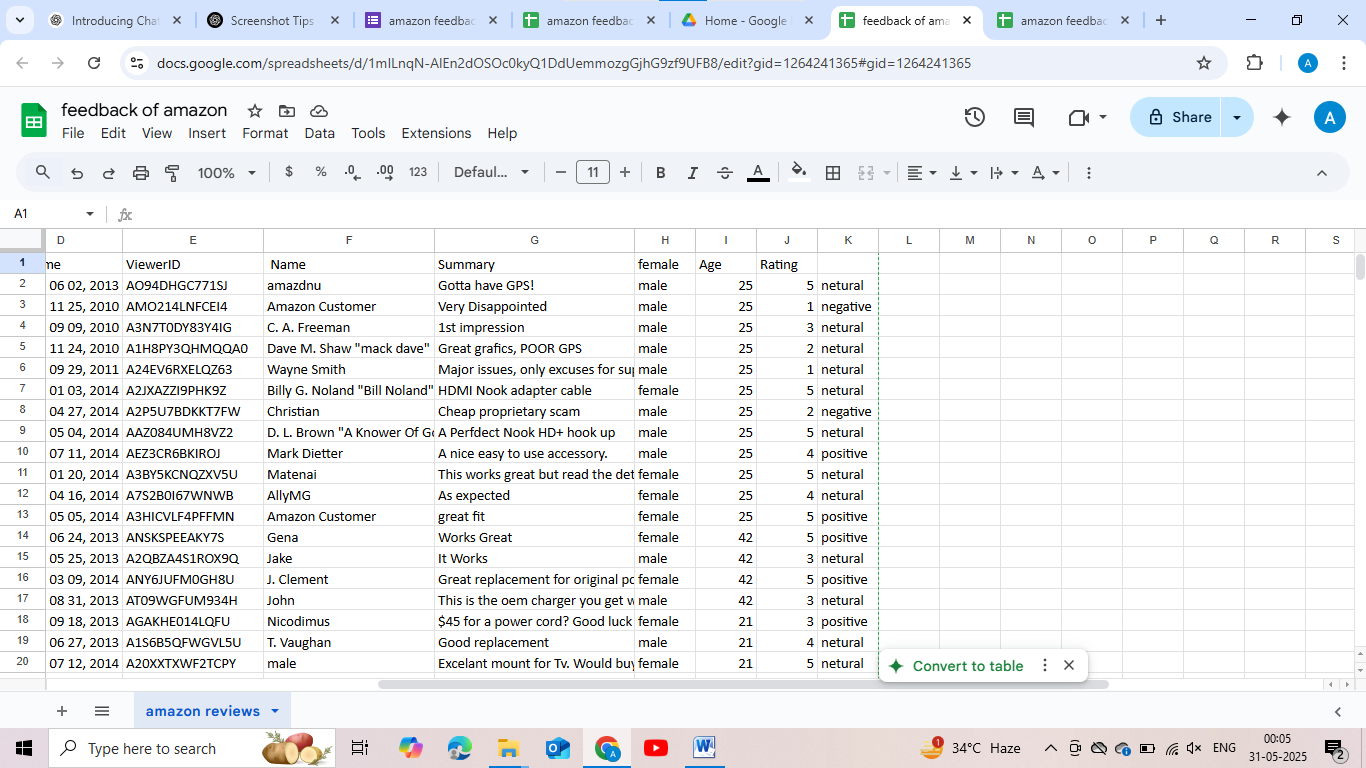
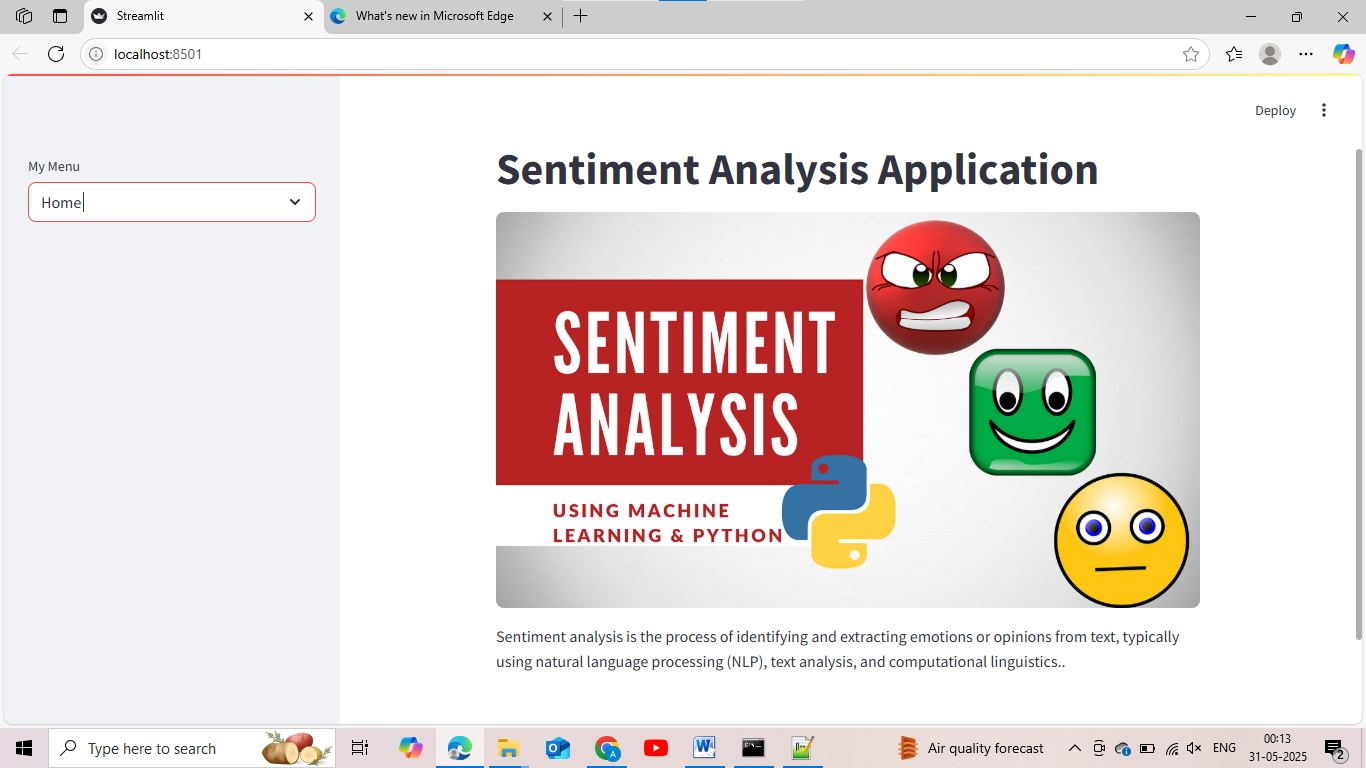
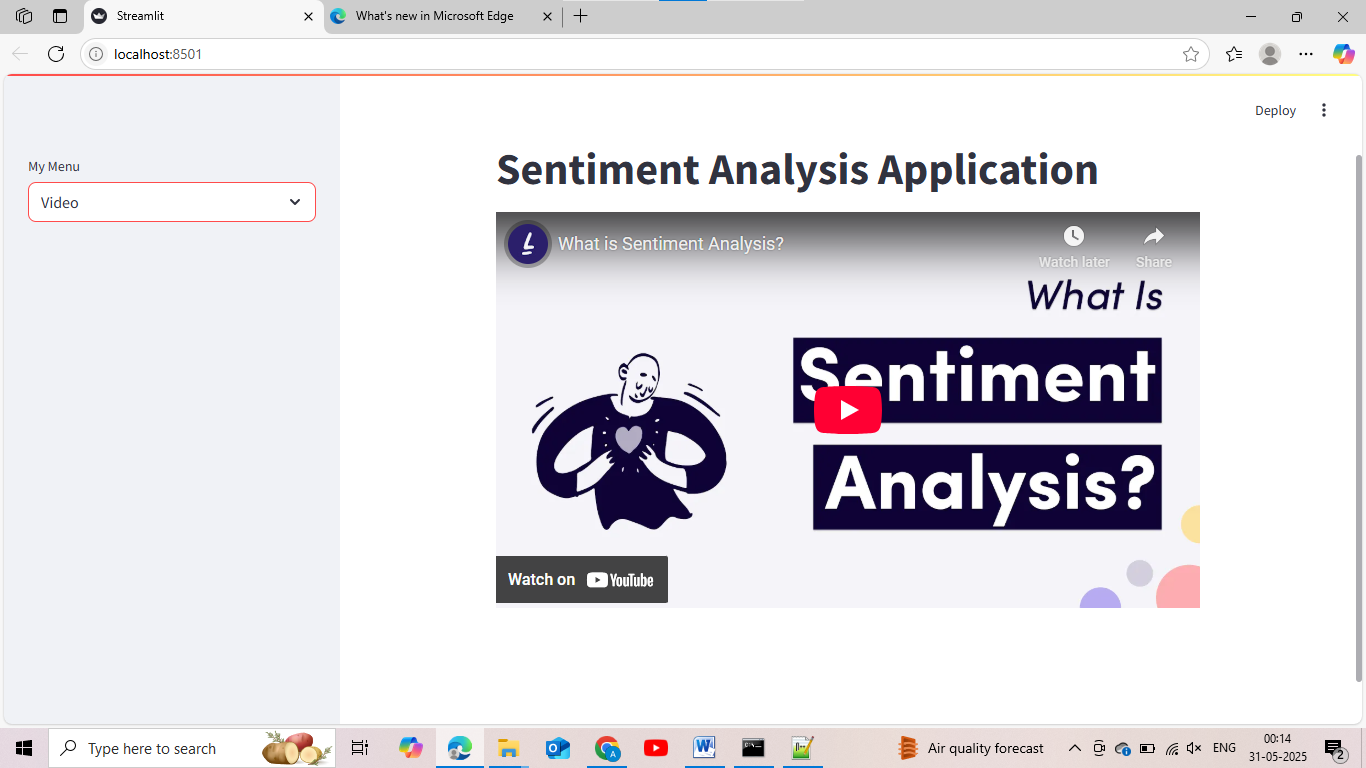
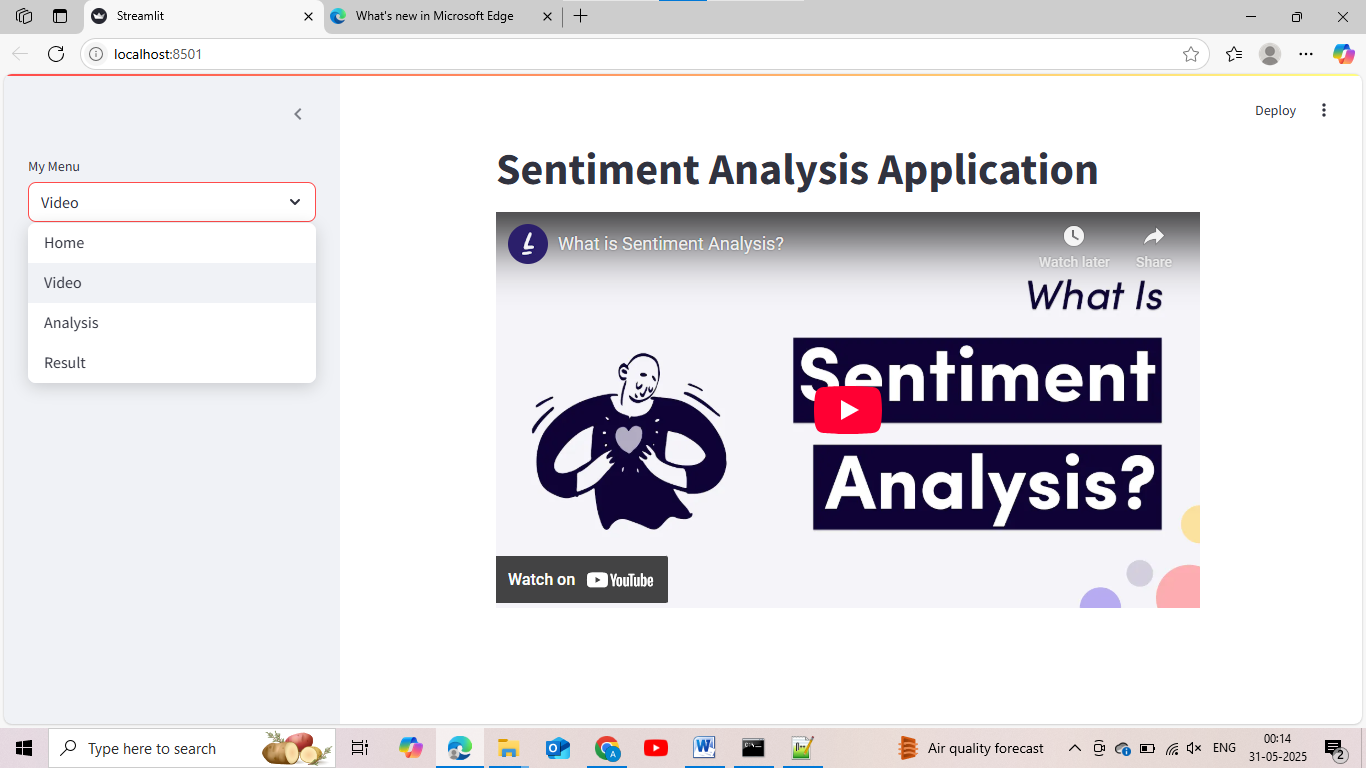
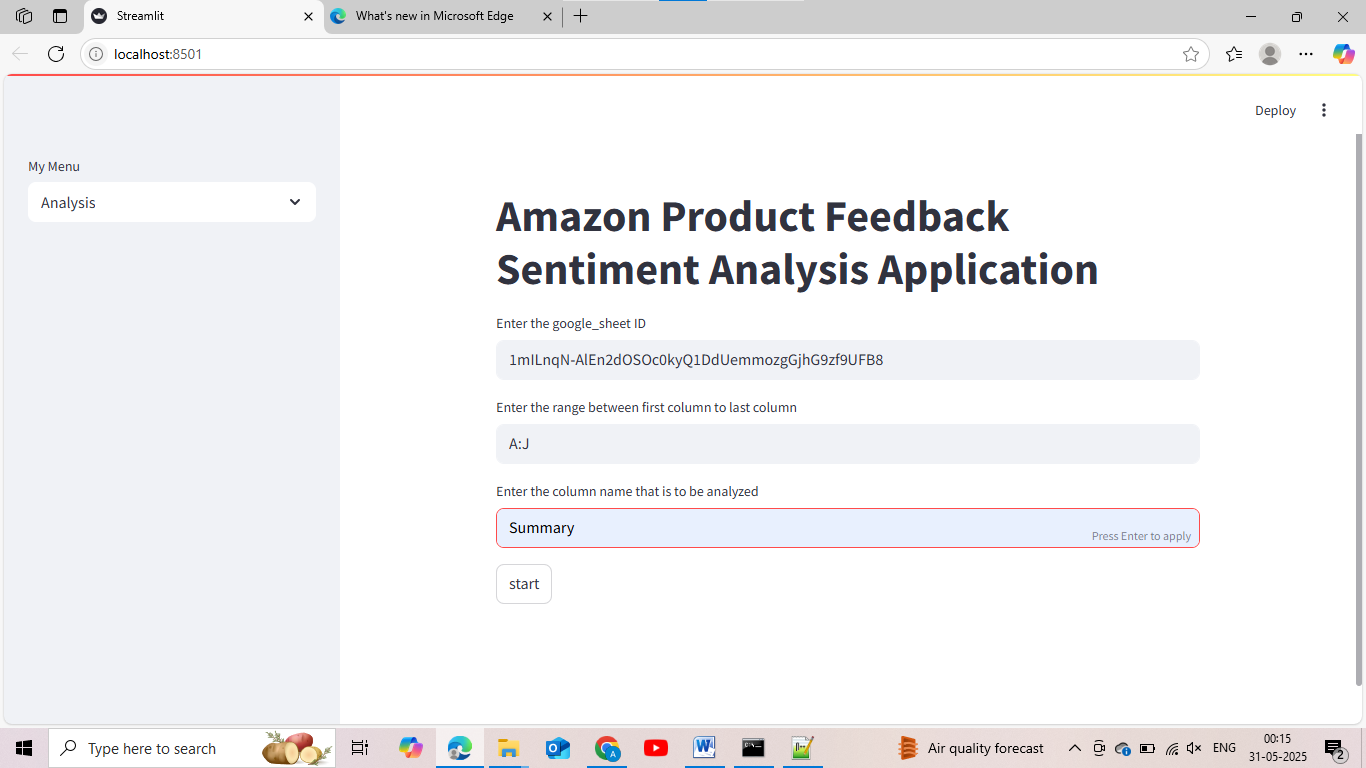
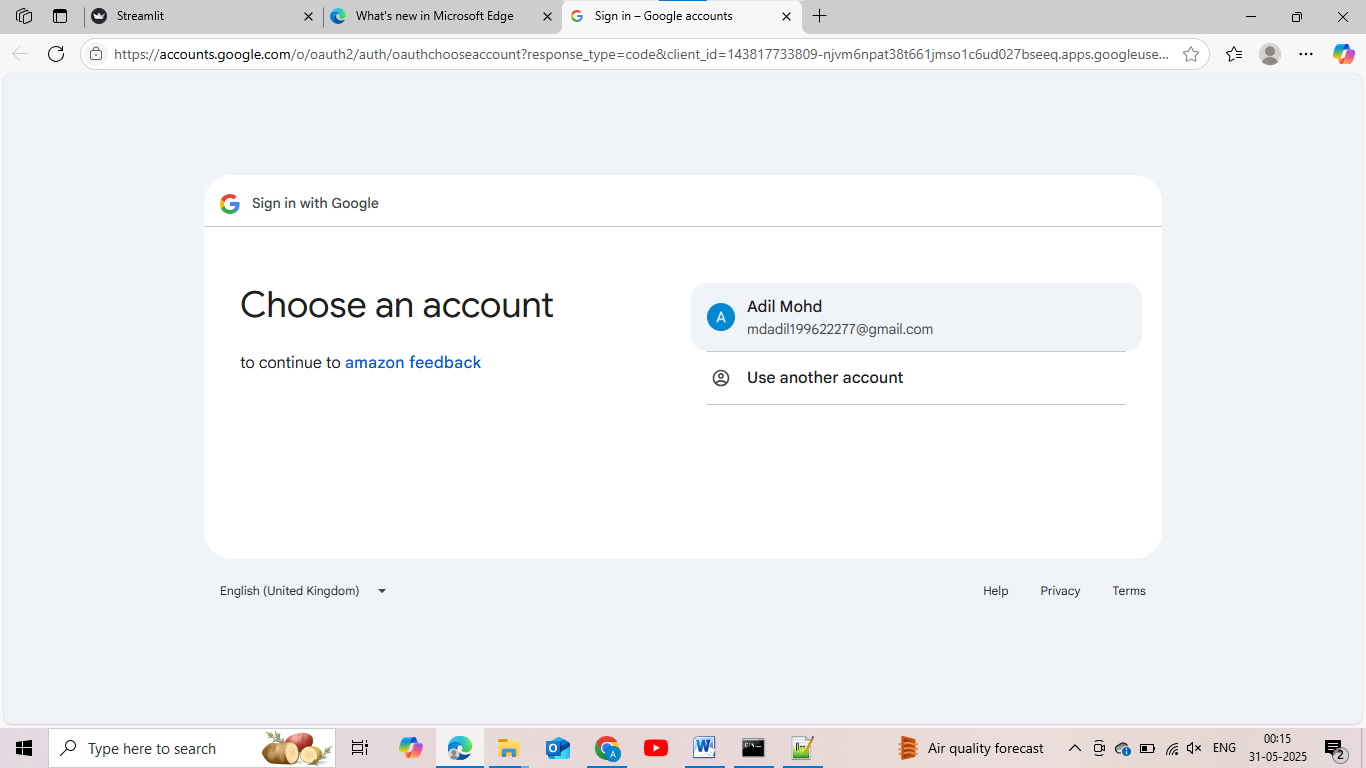
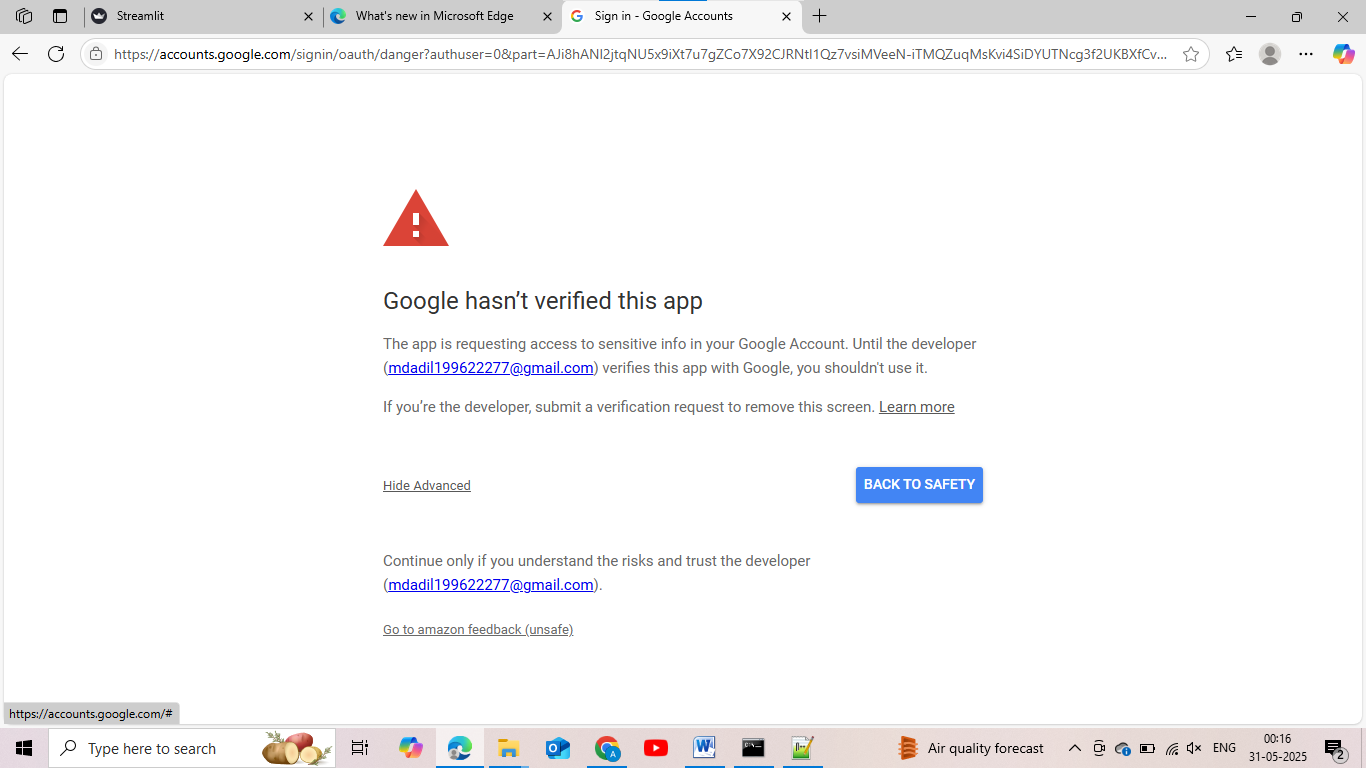
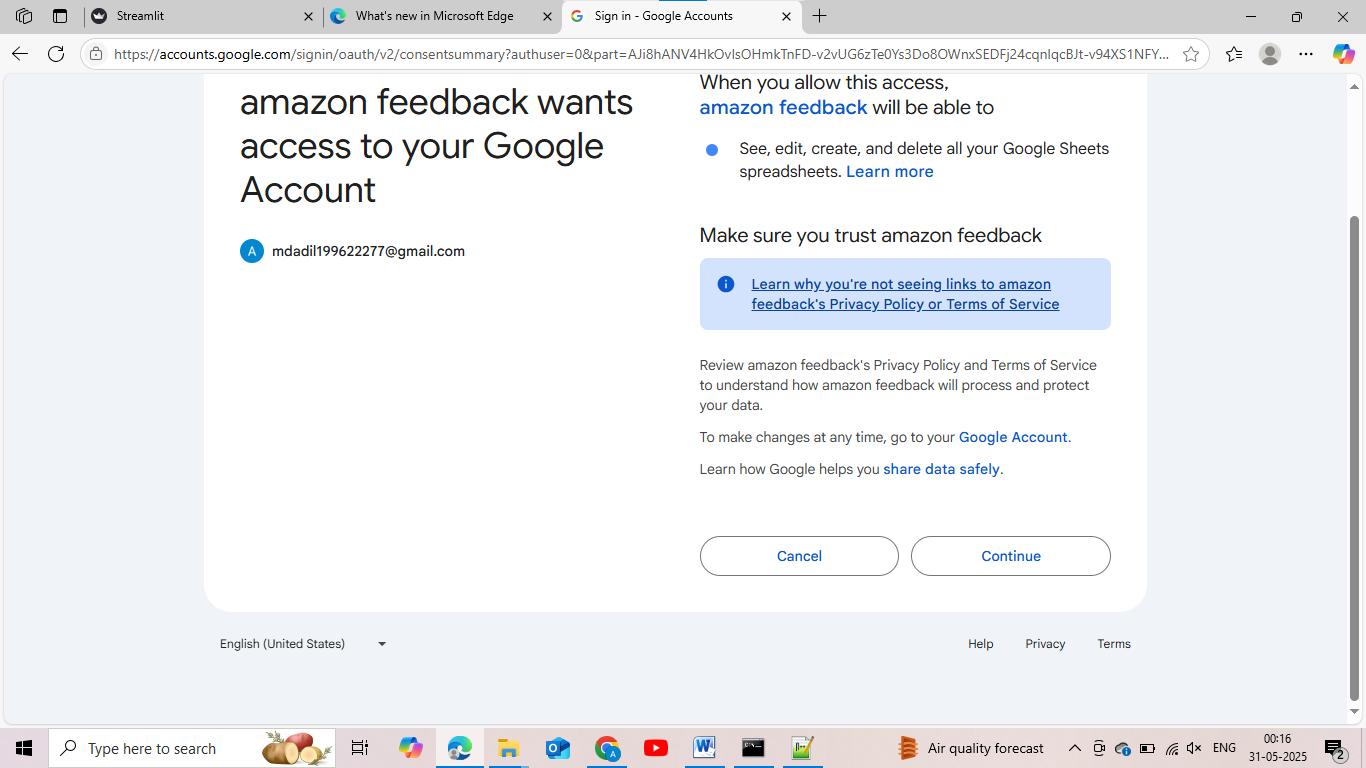
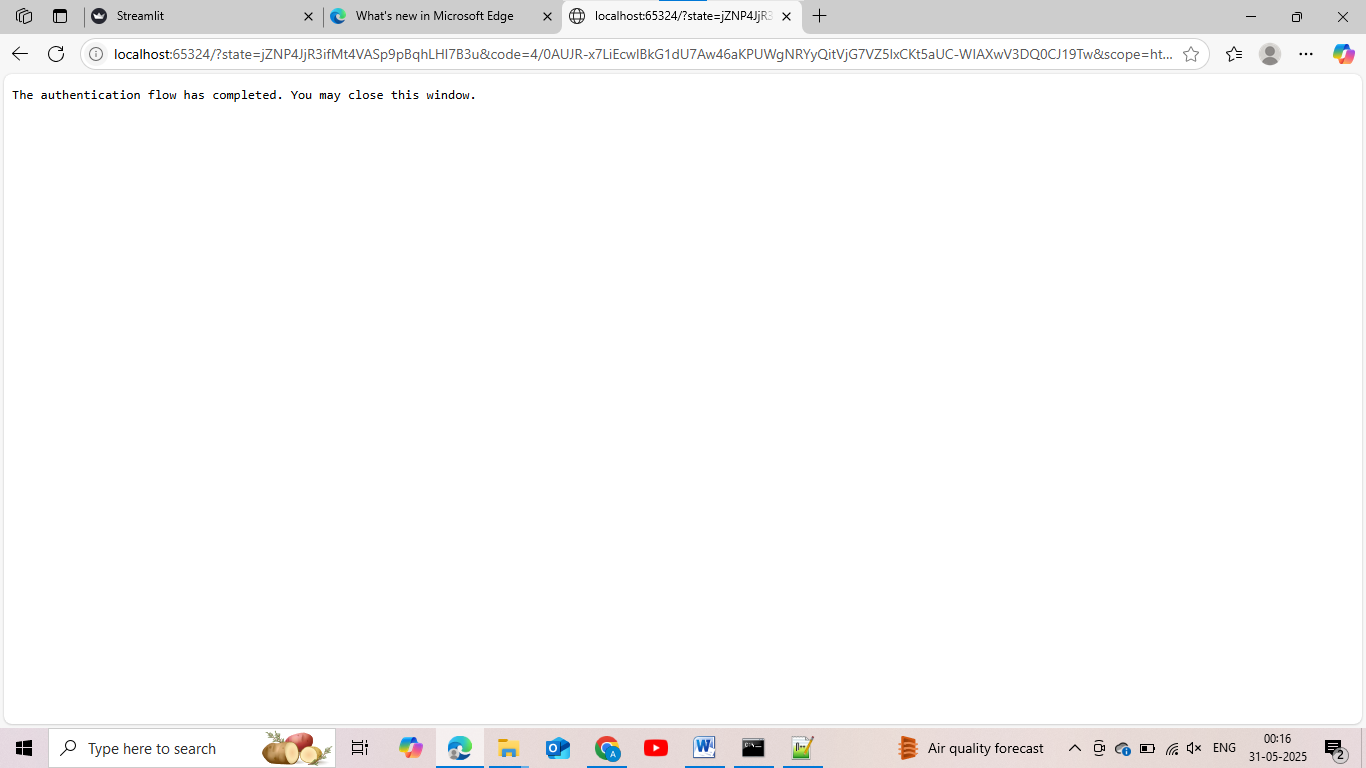
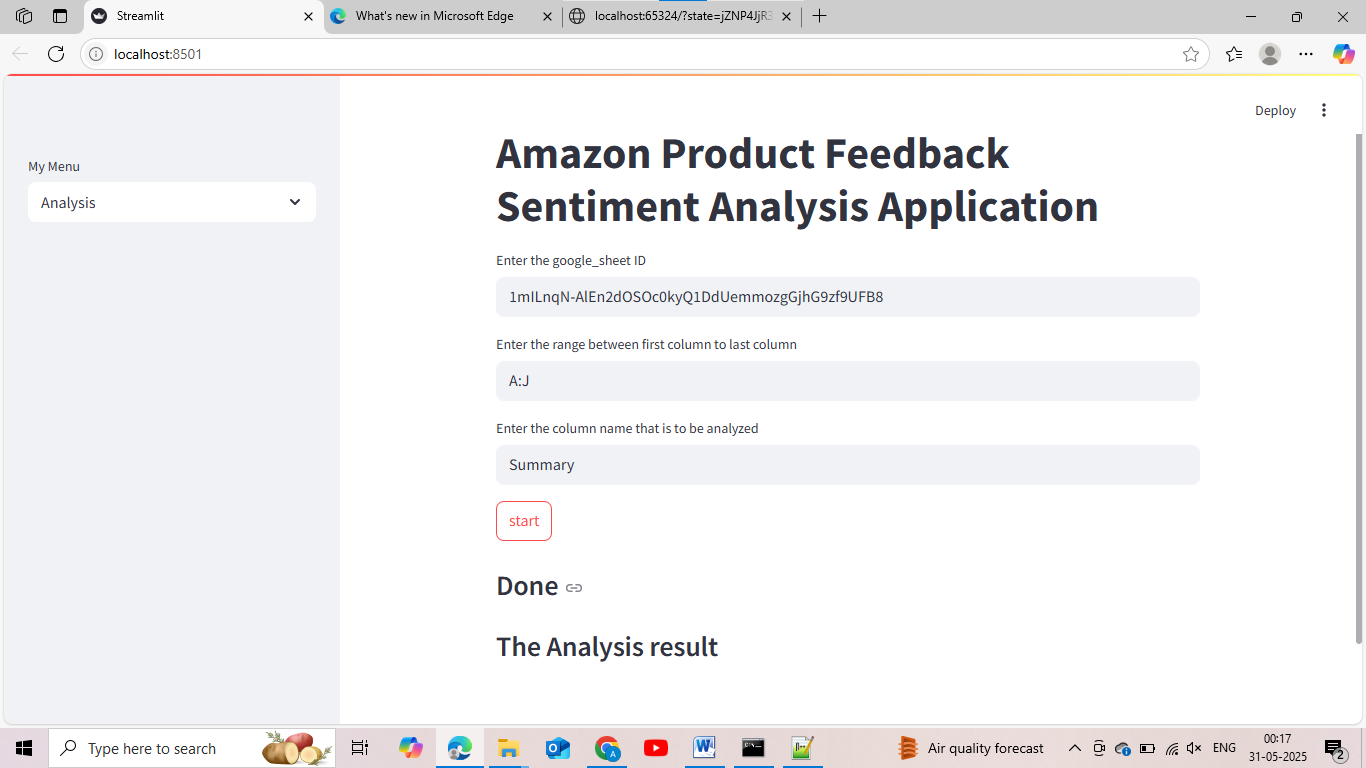
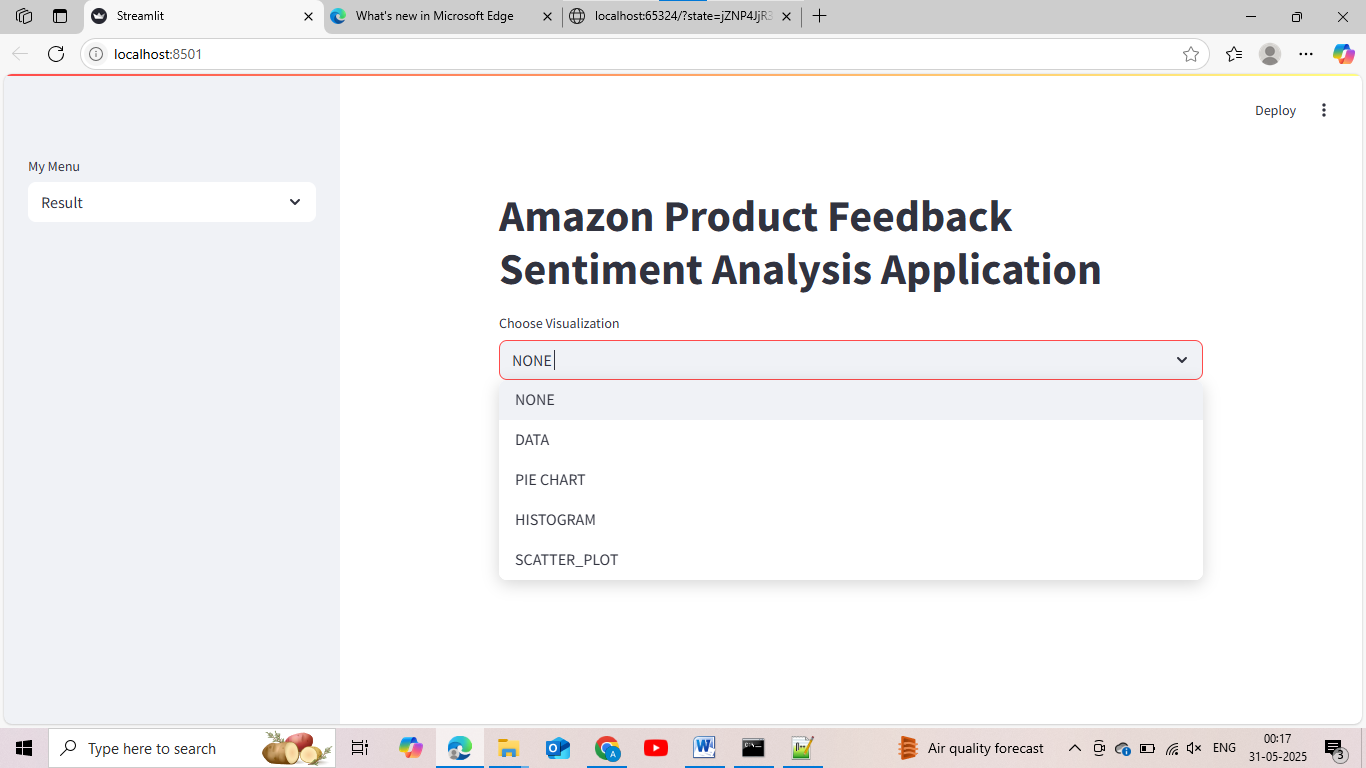
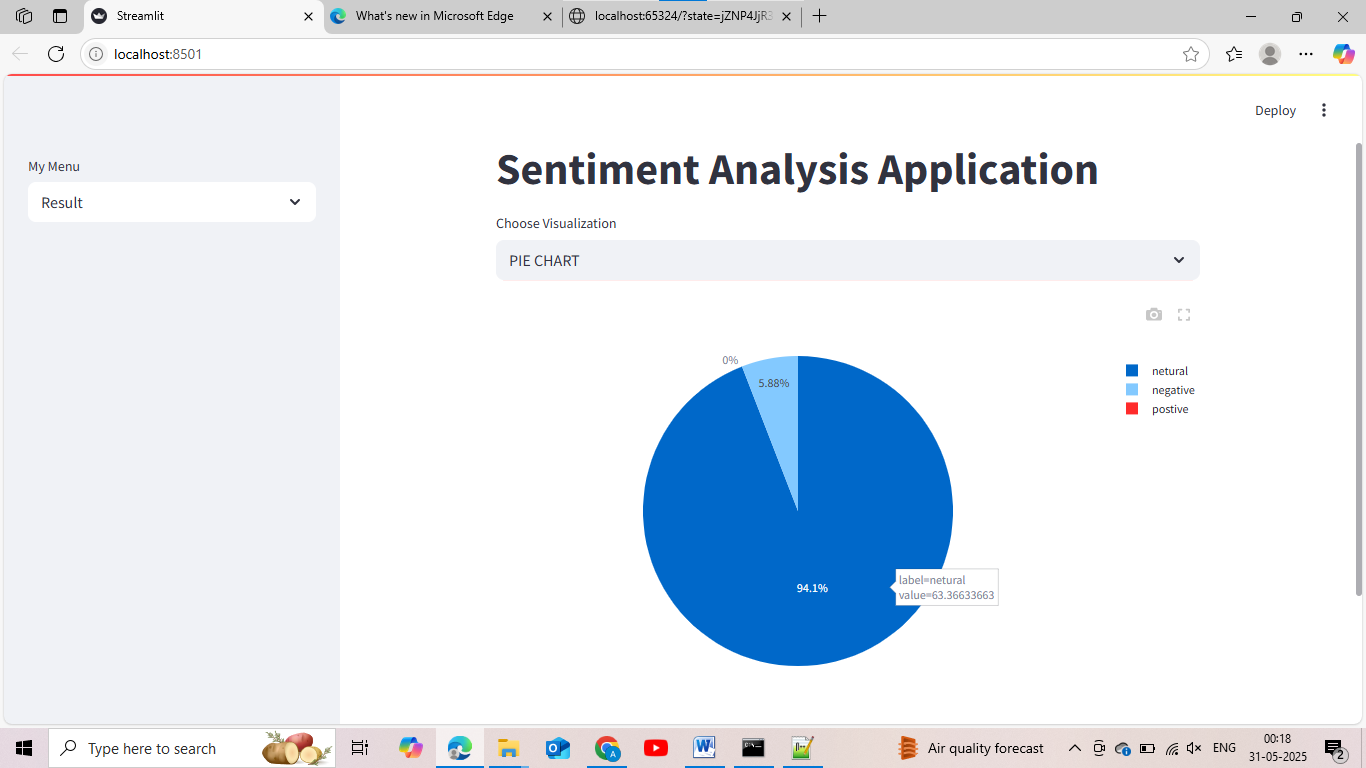
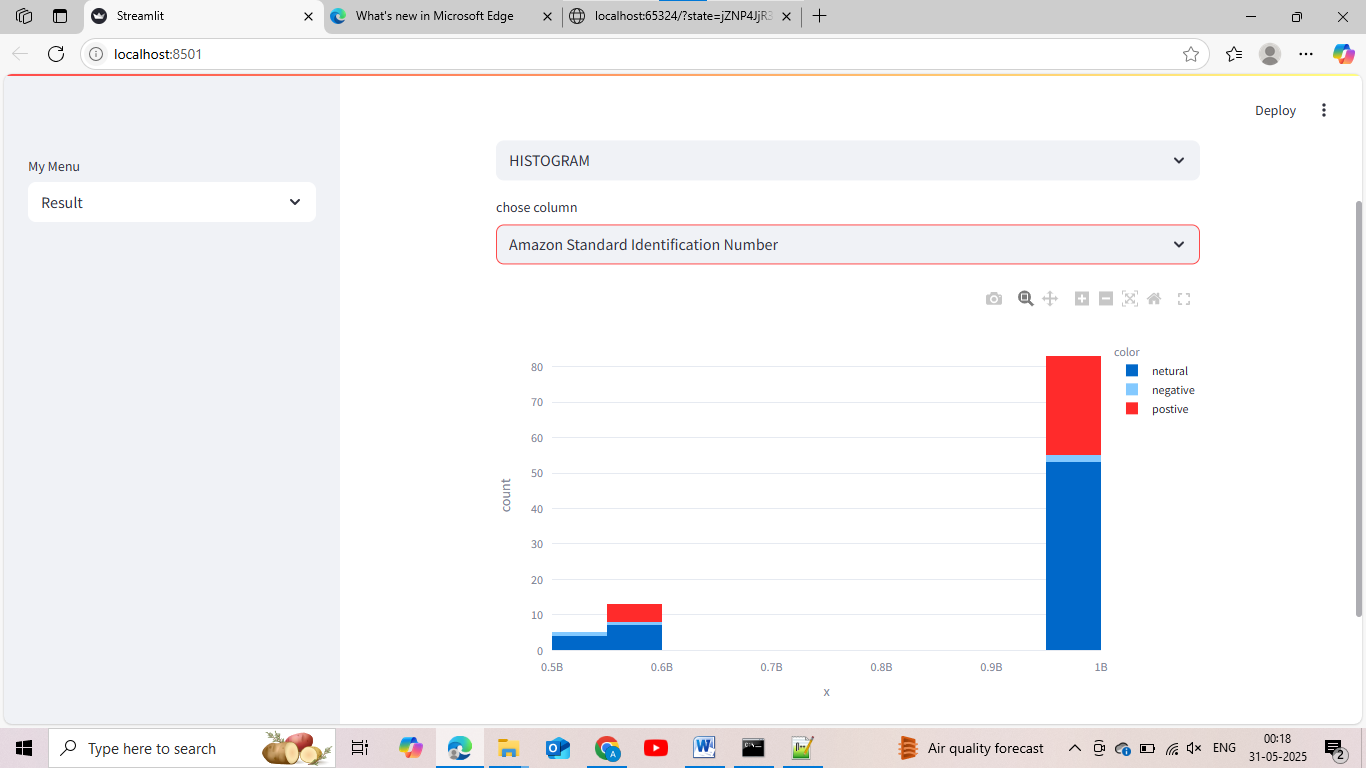
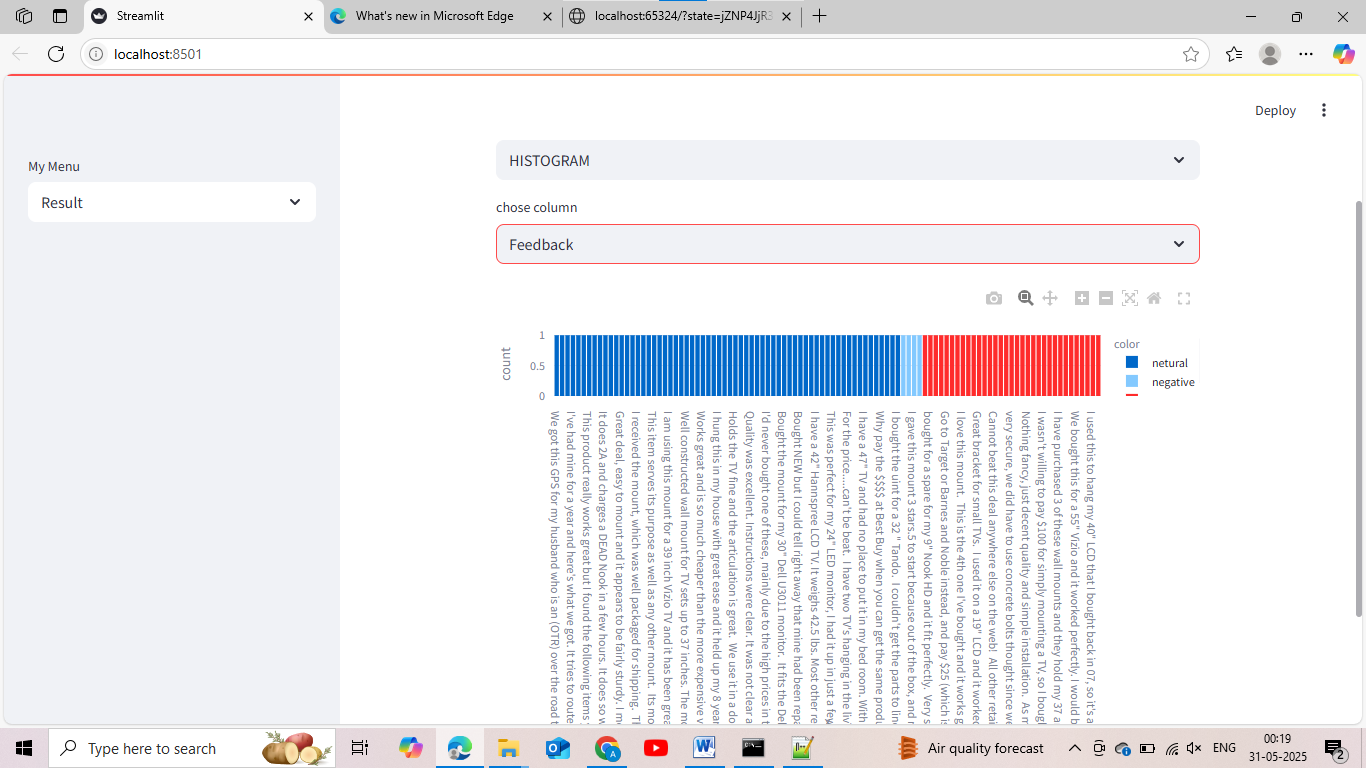
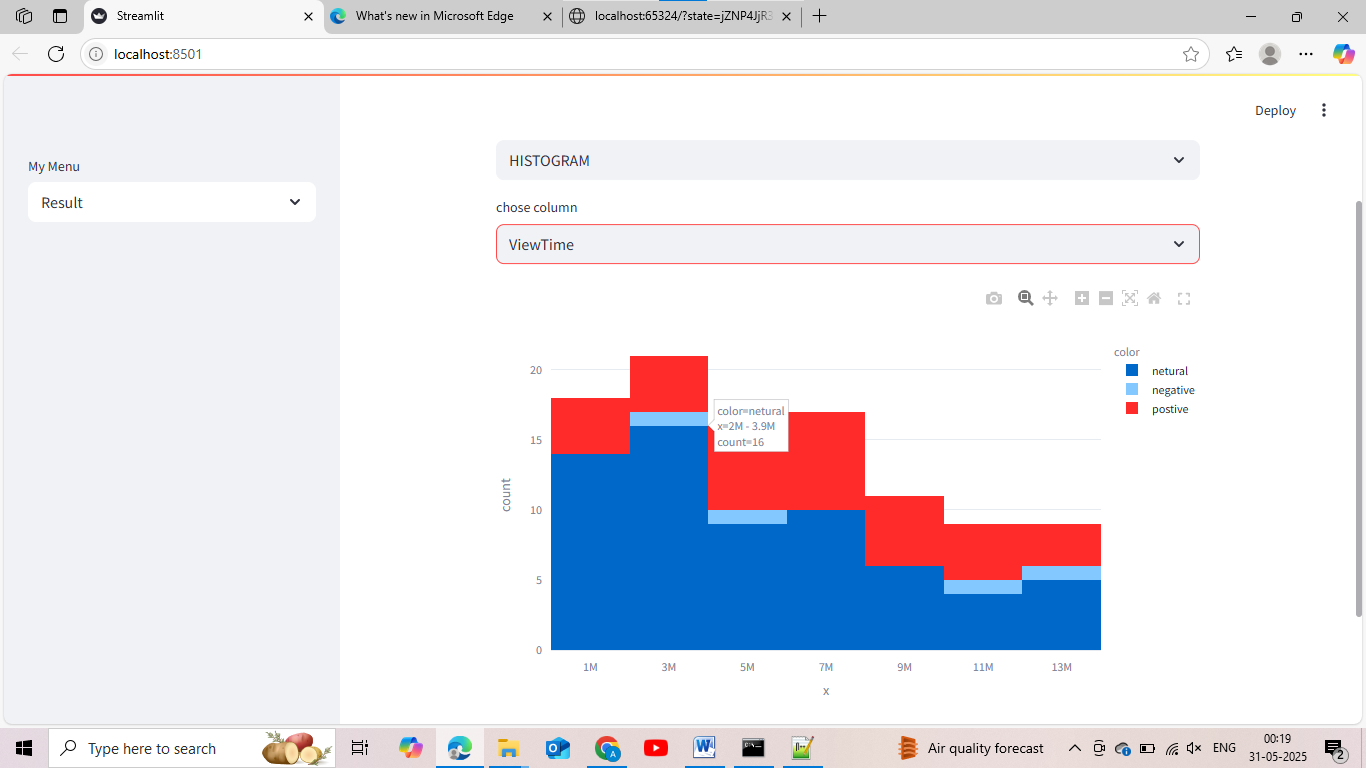
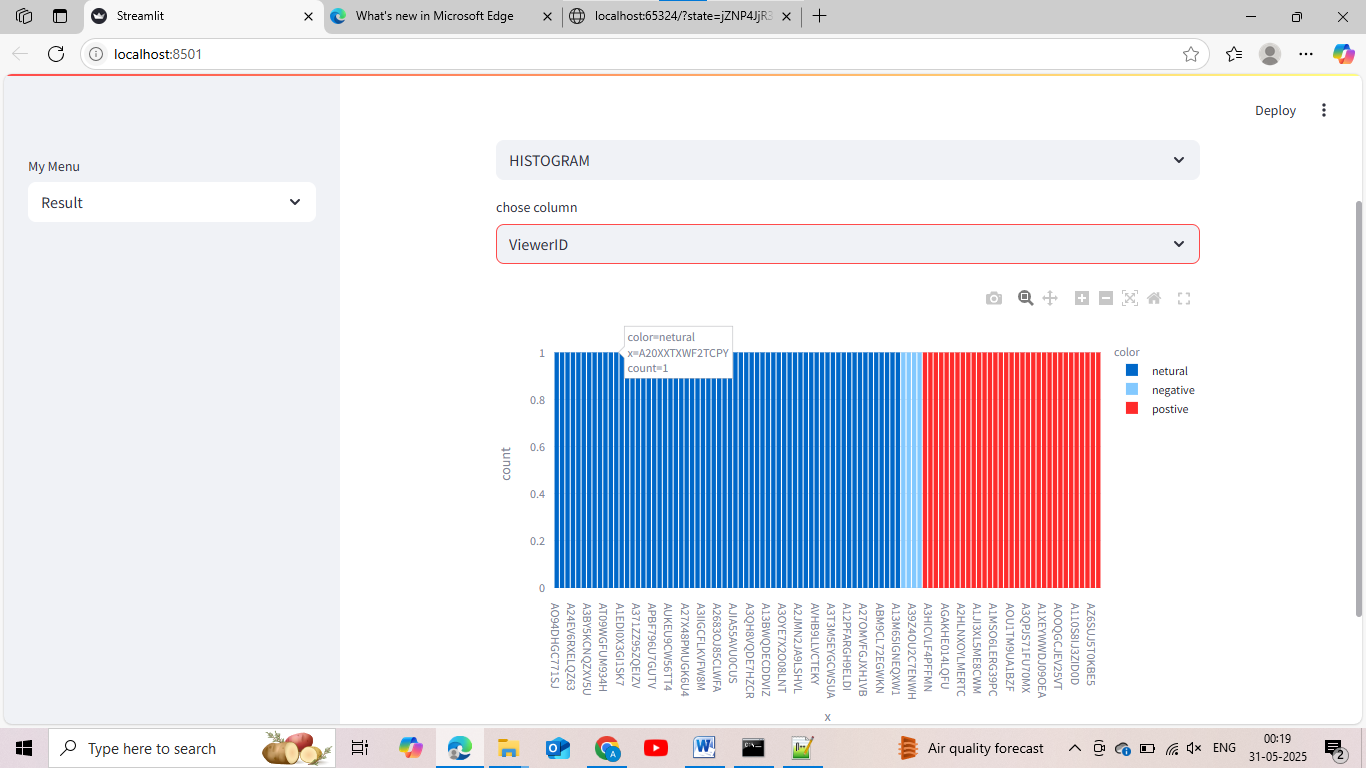
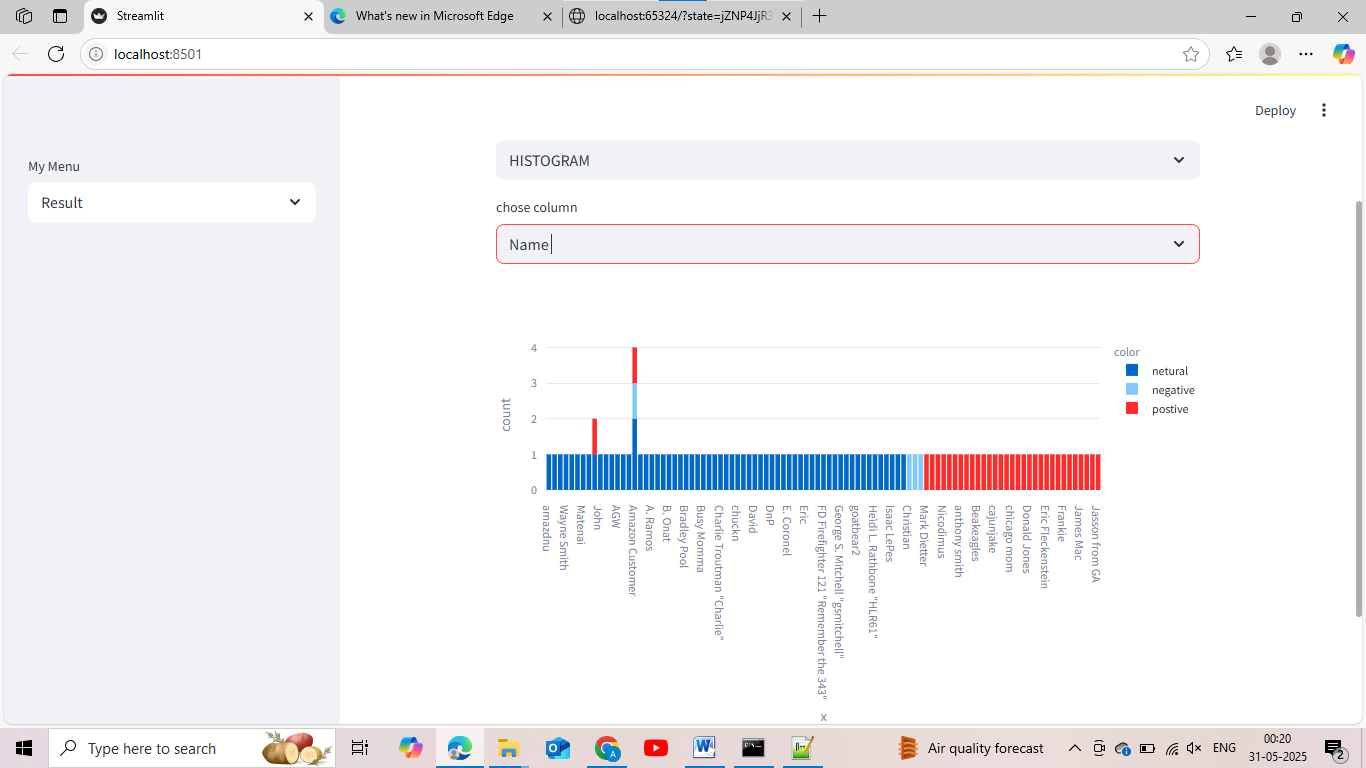
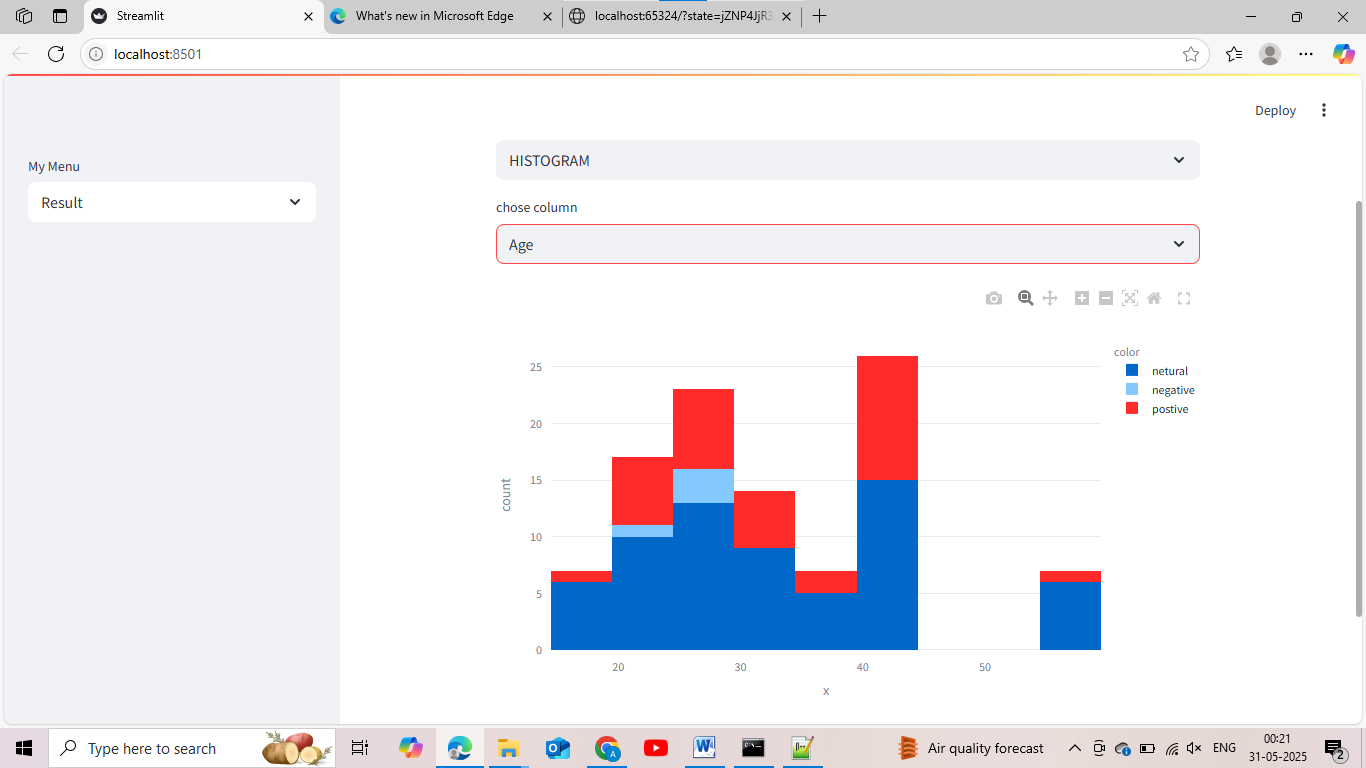
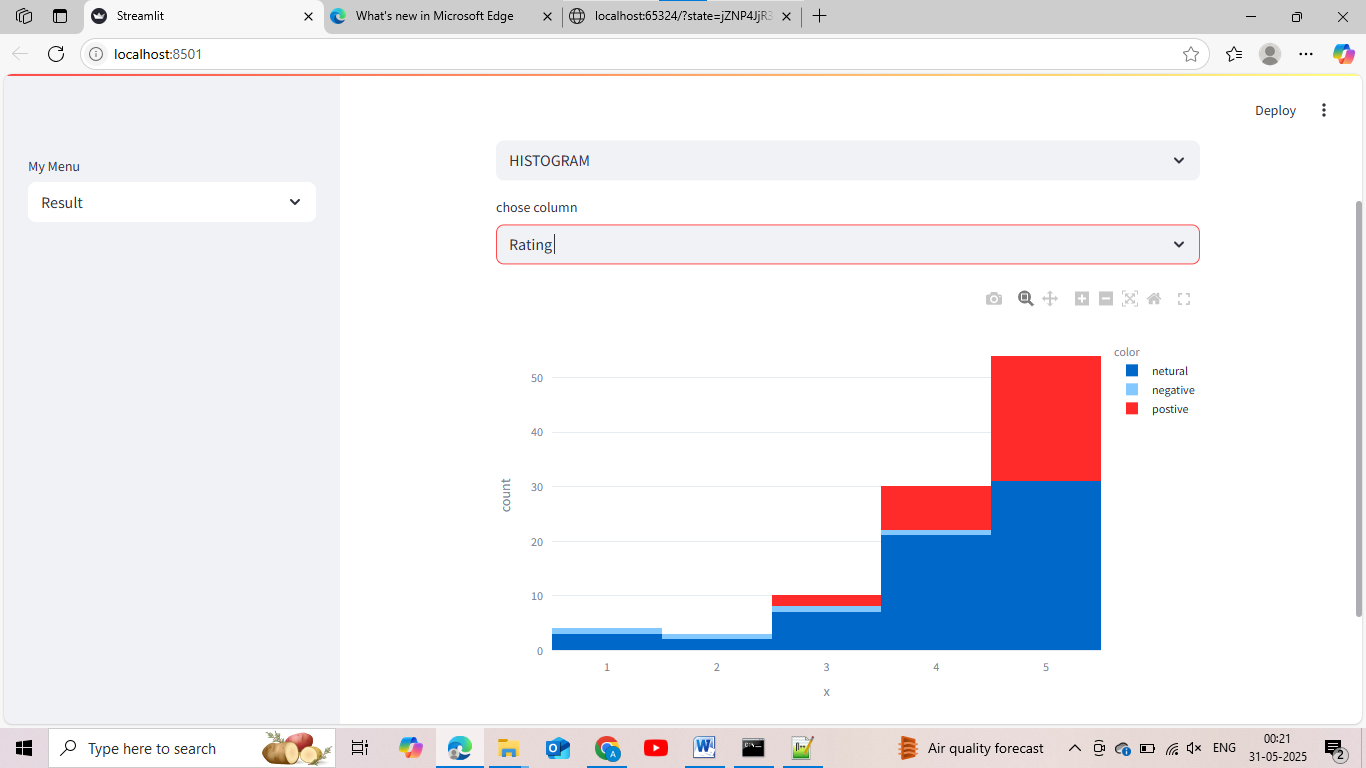
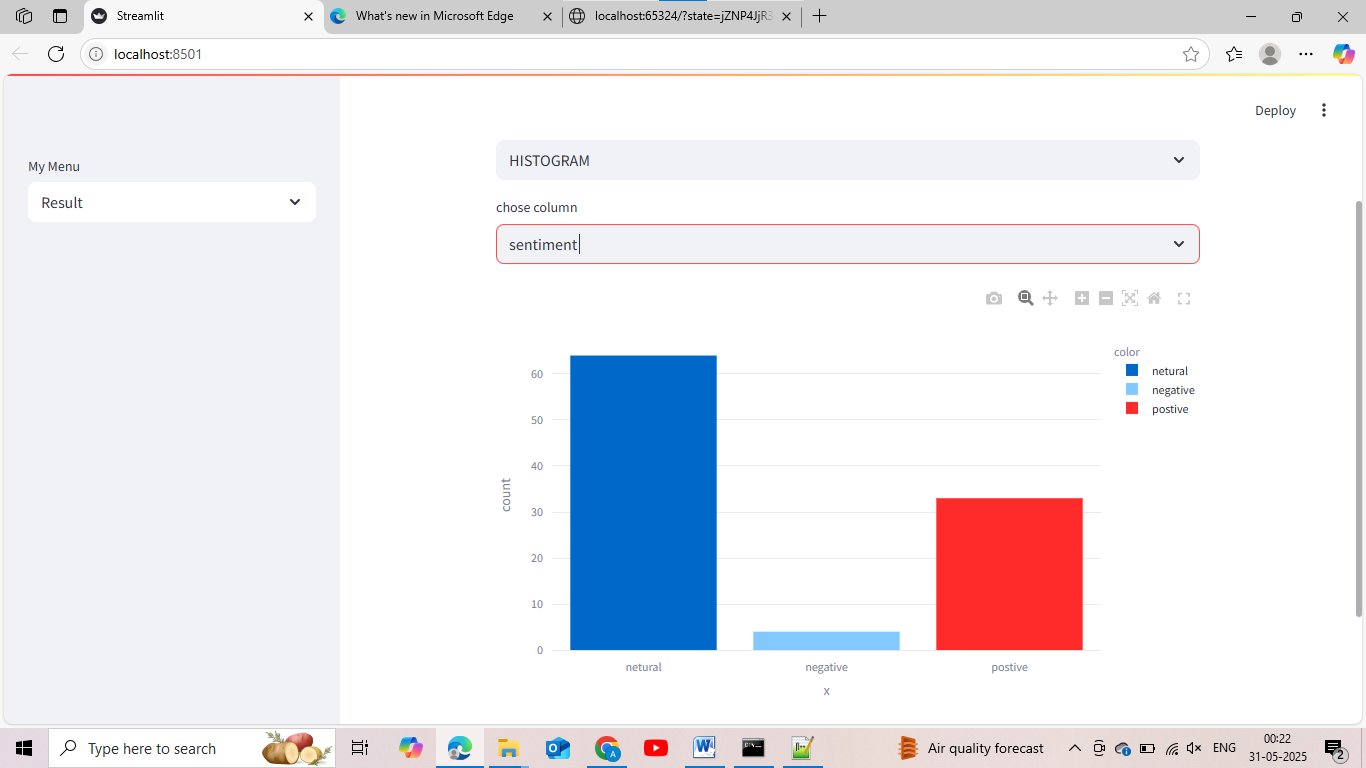
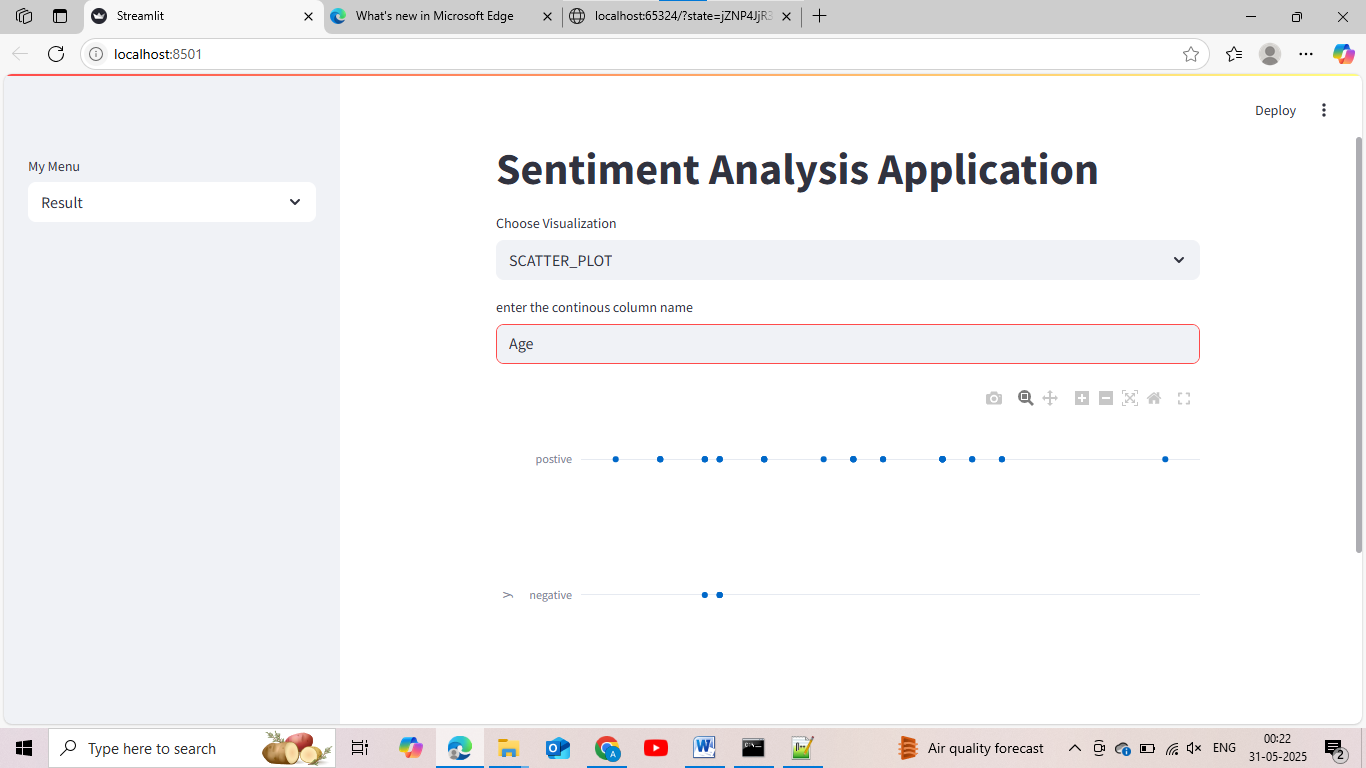
**if s:**

**fig=px.scatter(x=df[s],y=df["sentiment"])**

**st.plotly\_chart(fig)**

**Screenshot of Project**

**** ****

                         .