# DSCS Project

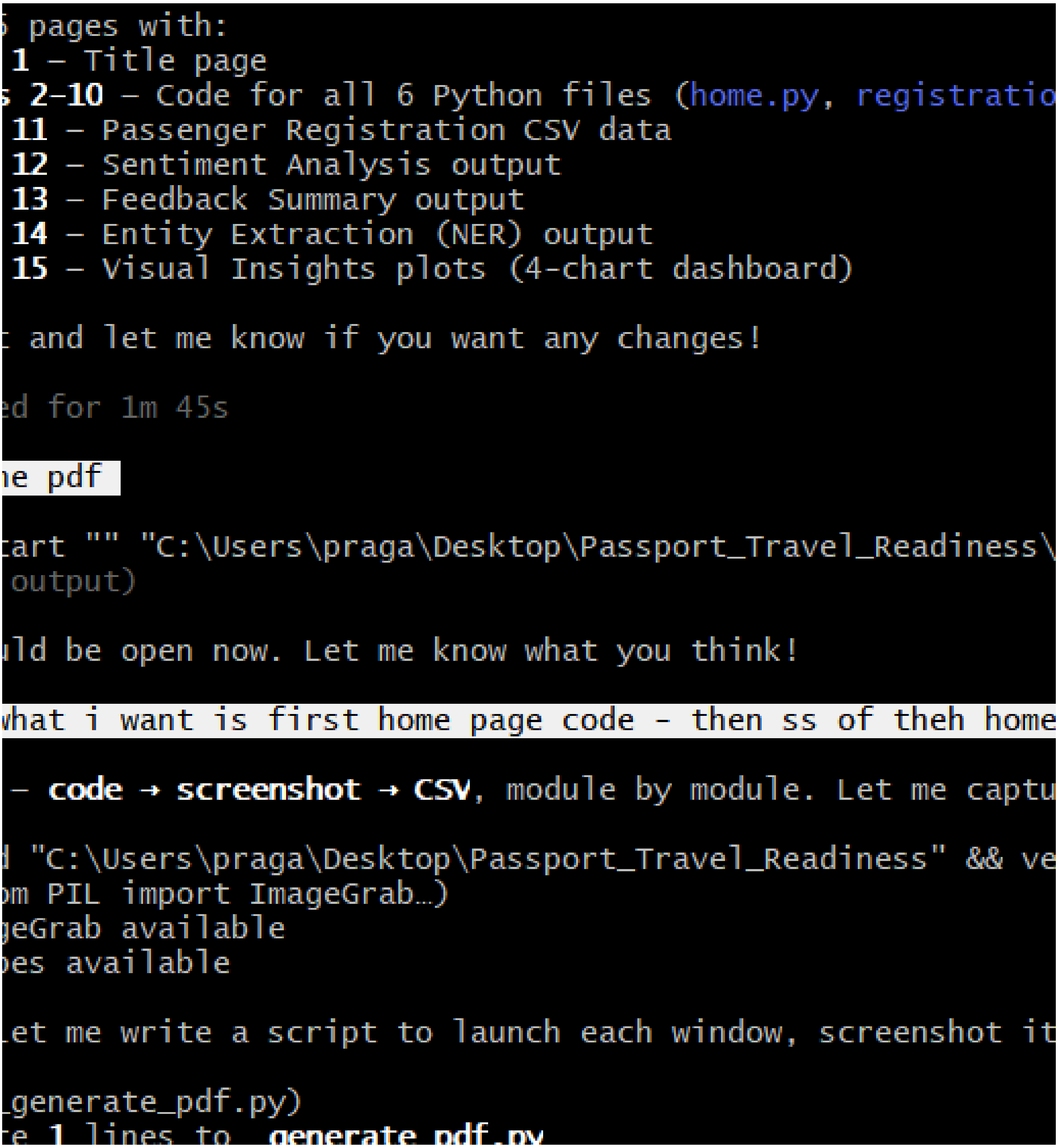
## Passport Travel Readiness System

Data Science - NLP - Visualization

# home.py

```
import tkinter as tk
import subprocess
import sys
import os

BASE = os.path.dirname(os.path.abspath(__file__))
VENV_PYTHON = os.path.join(BASE, "venv", "Scripts", "python.exe")
PYTHON = VENV_PYTHON if os.path.exists(VENV_PYTHON) else sys.executable

def run(file):
    subprocess.Popen([PYTHON, os.path.join(BASE, file)])

root = tk.Tk()
root.title("Passport Travel Readiness System")
root.geometry("500x520")
root.resizable(False, False)

tk.Label(
    root,
    text="Passport Travel Readiness\nData Science Case Study",
    font=("Arial", 18, "bold"),
    pady=20
).pack()

btn = {"font": ("Arial", 12), "width": 30, "height": 2}

tk.Button(root, text="Passenger Entry",
          command=lambda: run("registration.py"), **btn).pack(pady=6)

tk.Button(root, text="Feedback Insights",
          command=lambda: run("feedback_gui.py"), **btn).pack(pady=6)

tk.Button(root, text="Text Summarization",
          command=lambda: run("summary_gui.py"), **btn).pack(pady=6)

tk.Button(root, text="Entity Extraction (NER)",
          command=lambda: run("ner_gui.py"), **btn).pack(pady=6)

tk.Button(root, text="Visual Insights",
          command=lambda: run("plot_gui.py"), **btn).pack(pady=6)

tk.Button(root, text="Exit", bg="red", fg="white",
          command=root.destroy, **btn).pack(pady=20)

root.mainloop()
```

## Home Dashboard - Window



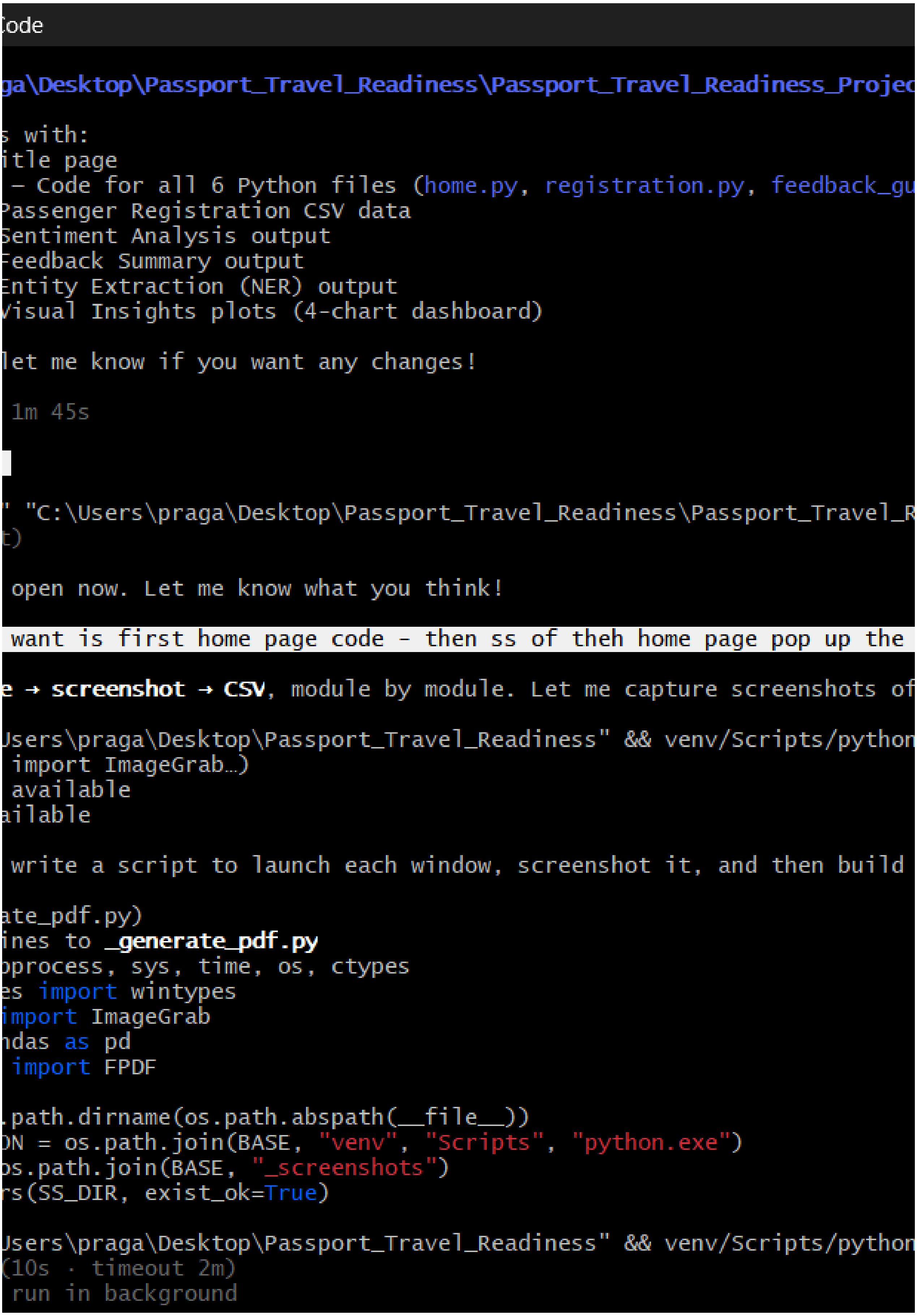
```
 pages with:
 1 – Title page
s 2–10 – Code for all 6 Python files (home.py, registratio
 11 – Passenger Registration CSV data
 12 – Sentiment Analysis output
 13 – Feedback Summary output
 14 – Entity Extraction (NER) output
 15 – Visual Insights plots (4-chart dashboard)

 and let me know if you want any changes!

ed for 1m 45s

e pdf

art "" "C:\Users\praga\Desktop\Passport_Travel_Readiness\
 output)

ld be open now. Let me know what you think!

hat i want is first home page code - then ss of theh home

 – code → screenshot → CSV, module by module. Let me captu

 "C:\Users\praga\Desktop\Passport_Travel_Readiness" && ve
om PIL import ImageGrab…)
eGrab available
es available

et me write a script to launch each window, screenshot it

generate_pdf.py)
e 1 lines to  generate_pdf.py
```

## registration.py

```
import tkinter as tk
from tkinter import messagebox, ttk
import csv, os

FILE = "passenger_registration.csv"

if not os.path.exists(FILE):
    with open(FILE, "w", newline="", encoding="utf-8") as f:
        csv.writer(f).writerow([
            "Full_Name","Passport_Number","Date_of_Birth","Destination_Country",
            "Travel_Purpose","Group_Size","Number_of_Bags",
            "Children_Count","Elderly_Count",
            "Accessibility_Needs","Medical_Conditions","Dietary_Requirements",
            "Language_Preference","First_Time_Traveler","Emergency_Contact",
            "Travel_Insurance","Vaccination_Status","Travel_Feedback"
        ])

def submit():
    row = [
        name.get(), passport.get(), dob.get(), dest.get(), purpose.get(),
        group.get(), bags.get(), children.get(), elderly.get(),
        accessibility.get(), medical.get(), dietary.get(),
        language.get(), first_time.get(), emergency.get(),
        insurance.get(), vaccination.get(),
        feedback.get("1.0", tk.END).strip()
    ]
    if not row[0] or not row[1]:
        messagebox.showerror("Error", "Name & Passport required")
        return

    with open(FILE, "a", newline="", encoding="utf-8") as f:
        csv.writer(f).writerow(row)

    messagebox.showinfo("Saved", "Data saved successfully")

    # Clear fields
    name.delete(0, tk.END)
    passport.delete(0, tk.END)
    dob.delete(0, tk.END)
    dest.delete(0, tk.END)
    group.delete(0, tk.END)
    bags.delete(0, tk.END)
    children.delete(0, tk.END)
    elderly.delete(0, tk.END)
    medical.delete(0, tk.END)
    dietary.delete(0, tk.END)
    emergency.delete(0, tk.END)
    feedback.delete("1.0", tk.END)

root = tk.Tk()
root.title("Passport Travel Readiness - Inclusive Entry Form")
root.geometry("600x850")

canvas = tk.Canvas(root)
scrollbar = ttk.Scrollbar(root, orient="vertical", command=canvas.yview)
scrollframe = tk.Frame(canvas)

scrollframe.bind(
    "<Configure>",
    lambda e: canvas.configure(scrollregion=canvas.bbox("all"))
)

canvas.create_window((0, 0), window=scrollframe, anchor="nw")
canvas.configure(yscrollcommand=scrollbar.set)

canvas.pack(side="left", fill="both", expand=True)
scrollbar.pack(side="right", fill="y")

tk.Label(scrollframe, text="Passport Travel Readiness System", font=("Arial", 16, "bold")).pack(pady=10)
tk.Label(scrollframe, text="Inclusive Travel Information", font=("Arial", 10, "italic"), fg="gray").pack()

def lbl(t): tk.Label(scrollframe, text=t, anchor="w").pack(fill="x", padx=10)
def ent():
    e = tk.Entry(scrollframe, width=40)
    e.pack(padx=10, pady=3)
    return e

# Personal Information
tk.Label(scrollframe, text="?? PERSONAL INFORMATION ??", font=("Arial", 11, "bold"), fg="#2c3e50").pack(fill="x", padx=10, pac
lbl("Full Name *"); name = ent()
lbl("Passport Number *"); passport = ent()
lbl("Date of Birth"); dob = ent()

# Destination
tk.Label(scrollframe, text="?? TRAVEL DETAILS ??", font=("Arial", 11, "bold"), fg="#2c3e50").pack(fill="x", padx=10, pady=8)
lbl("Destination Country"); dest = ent()

lbl("Travel Purpose")
purpose = tk.StringVar(value="Tourism")
ttk.Combobox(scrollframe, textvariable=purpose,
             values=["Tourism","Business","Education","Work","Visit Family"],
             width=37).pack(padx=10, pady=3)

lbl("Group Size"); group = ent()
lbl("Number of Bags"); bags = ent()

# Family Info
tk.Label(scrollframe, text="?? GROUP COMPOSITION ??", font=("Arial", 11, "bold"), fg="#2c3e50").pack(fill="x", padx=10, pady=8
lbl("Children Count"); children = ent()
lbl("Elderly Count"); elderly = ent()

# Inclusive & Accessibility
tk.Label(scrollframe, text="?? ACCESSIBILITY & HEALTH ??", font=("Arial", 11, "bold"), fg="#2c3e50").pack(fill="x", padx=10, p

lbl("Accessibility Needs (Wheelchair, Mobility Aid, Visual/Hearing Impairment, etc.)")
accessibility = ent()

lbl("Medical Conditions (if any)")
medical = ent()

lbl("Dietary Requirements (Vegetarian, Vegan, Halal, Kosher, Allergies, etc.)")
dietary = ent()

lbl("Language Preference")
language = tk.StringVar(value="English")
ttk.Combobox(scrollframe, textvariable=language,
             values=["English","Spanish","French","Mandarin","Hindi","Arabic","Other"],
             width=37).pack(padx=10, pady=3)

# Travel Experience
tk.Label(scrollframe, text="?? TRAVEL EXPERIENCE ??", font=("Arial", 11, "bold"), fg="#2c3e50").pack(fill="x", padx=10, pady=8

lbl("First Time Traveler?")
first_time = tk.StringVar(value="No")
ttk.Combobox(scrollframe, textvariable=first_time,
             values=["Yes","No"],
             width=37).pack(padx=10, pady=3)

# Important Info
tk.Label(scrollframe, text="?? IMPORTANT INFO ??", font=("Arial", 11, "bold"), fg="#2c3e50").pack(fill="x", padx=10, pady=8)

lbl("Emergency Contact (Name & Phone)")
emergency = ent()

lbl("Travel Insurance?")
insurance = tk.StringVar(value="No")
ttk.Combobox(scrollframe, textvariable=insurance,
             values=["Yes","No"],
             width=37).pack(padx=10, pady=3)

lbl("Vaccination Status")
vaccination = tk.StringVar(value="Complete")
ttk.Combobox(scrollframe, textvariable=vaccination,
             values=["Complete","Partial","Not Vaccinated","Prefer Not to Say"],
             width=37).pack(padx=10, pady=3)

# Feedback
tk.Label(scrollframe, text="?? ADDITIONAL FEEDBACK ??", font=("Arial", 11, "bold"), fg="#2c3e50").pack(fill="x", padx=10, pady
lbl("Travel Feedback/Comments")
feedback = tk.Text(scrollframe, height=4, width=40)
feedback.pack(padx=10, pady=3)
```

```
# Submit Button
tk.Button(scrollframe, text="Submit", bg="#27ae60", fg="white",
          width=40, font=("Arial", 12, "bold"), command=submit).pack(pady=15)

root.mainloop()
```

## Passenger Entry - Window



```
ga\Desktop\Passport_Travel_Readiness\Passport_Travel_Readiness_Projec

s with:
itle page
 - Code for all 6 Python files (home.py, registration.py, feedback_gu
Passenger Registration CSV data
Sentiment Analysis output
Feedback Summary output
Entity Extraction (NER) output
Visual Insights plots (4-chart dashboard)

let me know if you want any changes!

 1m 45s


' "C:\Users\praga\Desktop\Passport_Travel_Readiness\Passport_Travel_R
t)

 open now. Let me know what you think!

 want is first home page code - then ss of theh home page pop up the

e → screenshot → CSV, module by module. Let me capture screenshots of

Users\praga\Desktop\Passport_Travel_Readiness" && venv/Scripts/python
 import ImageGrab…)
 available
ailable

 write a script to launch each window, screenshot it, and then build

ate_pdf.py)
ines to _generate_pdf.py
bprocess, sys, time, os, ctypes
es import wintypes
import ImageGrab
ndas as pd
 import FPDF

.path.dirname(os.path.abspath(__file__))
ON = os.path.join(BASE, "venv", "Scripts", "python.exe")
os.path.join(BASE, "_screenshots")
rs(SS_DIR, exist_ok=True)

Users\praga\Desktop\Passport_Travel_Readiness" && venv/Scripts/python
(10s · timeout 2m)
 run in background
```

# passenger_registration.csv

| Full_Name | Passport_Number | Date_of_Birth | Destination_Country | Travel_Purpose | Group_Size | Number_of_Bags | Children_Count | Elderly_Count |
|---|---|---|---|---|---|---|---|---|
| Rahul Sharma | P1234567 | 12-04-1995 | France | Tourism | 4 | 5 | 1 | 1 |
| Ananya Verma | P2345678 | 25-08-1998 | Japan | Education | 1 | 2 | 0 | 0 |
| Mohit Khan | P3456789 | 09-06-1992 | UAE | Business | 3 | 4 | 0 | 1 |
| Sneha Kulkarni | P4567890 | 14-02-1996 | Germany | Tourism | 2 | 3 | 0 | 0 |
| Amit Patel | P5678901 | 03-01-1989 | USA | Business | 5 | 6 | 2 | 0 |
| Pooja Nair | P6789012 | 17-11-1997 | Canada | Education | 1 | 2 | 0 | 0 |
| Rohit Mehta | P7890123 | 30-09-1994 | UK | Tourism | 3 | 4 | 1 | 0 |
| Neha Gupta | P8901234 | 11-07-1999 | Australia | Work | 2 | 3 | 0 | 1 |
| Kunal Shah | P9012345 | 21-05-1991 | Singapore | Business | 1 | 1 | 0 | 0 |
| Kavya Iyer | P0123456 | 28-08-1995 | Italy | Tourism | 3 | 4 | 1 | 0 |

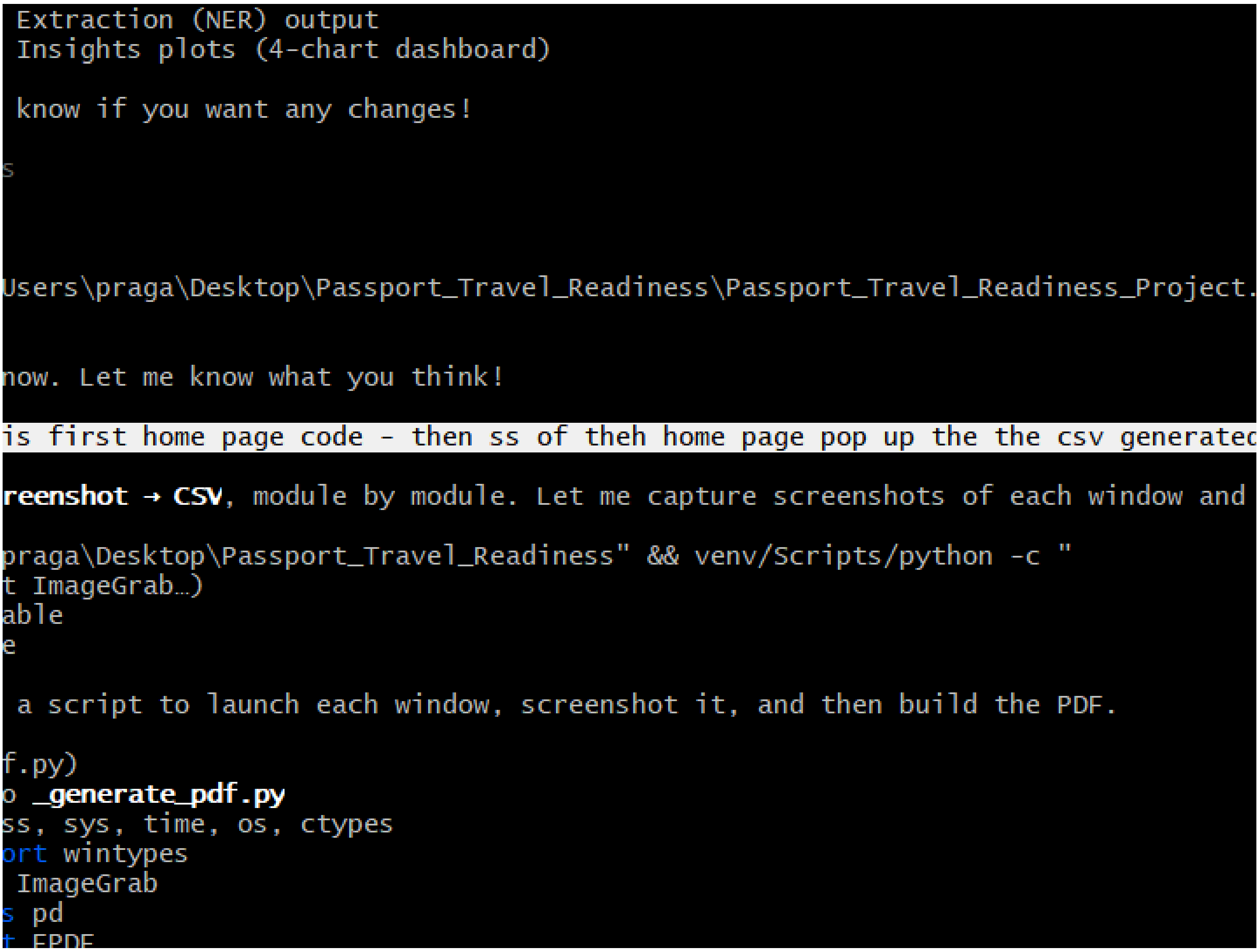
# passenger_registration.csv (continued)

| Full_Name | Accessibility_Needs | Medical_Conditions | Dietary_Requirements | Language_Preference | First_Time_Traveler | Emergency_Contact | Travel_Insurance | Vaccination_Status |
|---|---|---|---|---|---|---|---|---|
| Rahul Sharma | Wheelchair | Asthma | Vegetarian | English | No | 9876543210 | Yes | Complete |
| Ananya Verma | No Special Needs | Thyroid | Vegan | English | Yes | 9123456780 | Yes | Complete |
| Mohit Khan | Hearing Aid | Diabetes | Halal | Hindi | No | 9988776655 | Yes | Complete |
| Sneha Kulkarni | No Special Needs | Migraine | Vegetarian | English | No | 9090909090 | No | Complete |
| Amit Patel | No Special Needs | Hypertension | Jain | Gujarati | No | 8899776655 | Yes | Complete |
| Pooja Nair | Visual Impairment | Allergy | Gluten-Free | English | Yes | 9812345678 | Yes | Complete |
| Rohit Mehta | No Special Needs | Cholesterol | No Restriction | Hindi | No | 9001122334 | No | Complete |
| Neha Gupta | Mobility Aid | BP Low | Kosher | English | Yes | 9870011223 | Yes | Complete |
| Kunal Shah | No Special Needs | No Conditions | No Restriction | English | No | 9988112233 | Yes | Complete |
| Kavya Iyer | Wheelchair | Allergy | Vegetarian | English | Yes | 9876501234 | No | Complete |

# feedback_gui.py

```
import tkinter as tk
from tkinter import scrolledtext, messagebox
import pandas as pd
from textblob import TextBlob
import os
import time

def sentiment(text):
    p = TextBlob(str(text)).sentiment.polarity
    if p > 0: return "Positive"
    if p < 0: return "Negative"
    return "Neutral"

def analyze():
    try:
        df = pd.read_csv("passenger_registration.csv")

        if len(df) == 0:
            messagebox.showwarning("No Data", "Please enter travel data first!")
            return

        df["Sentiment"] = df["Travel_Feedback"].fillna("").apply(sentiment)

        # Try to save CSV with retry
        max_retries = 3
        for attempt in range(max_retries):
            try:
                if os.path.exists("sentiment_analysis.csv"):
                    try:
                        os.remove("sentiment_analysis.csv")
                    except:
                        time.sleep(0.5)

                df.to_csv("sentiment_analysis.csv", index=False)
                break
            except PermissionError:
                if attempt < max_retries - 1:
                    time.sleep(0.5)
                else:
                    messagebox.showwarning("File Locked", "Could not save CSV (file locked), but analysis displayed below.")

        messagebox.showinfo("Success", "Feedback analysis complete!")

        out.delete(1.0, tk.END)
        out.insert(tk.END, "?? FEEDBACK SENTIMENT ANALYSIS ??\n\n")

        for idx, (_, r) in enumerate(df.iterrows(), 1):
            feedback = str(r['Travel_Feedback'])[:100] if pd.notna(r['Travel_Feedback']) else "No feedback"
            sentiment_val = r['Sentiment']
            out.insert(tk.END, f"{idx}. Feedback: {feedback}\n   ? Sentiment: {sentiment_val}\n\n")

        # Summary
        sentiment_counts = df['Sentiment'].value_counts()
        out.insert(tk.END, "\n?? SUMMARY ??\n")
        for sentiment_type, count in sentiment_counts.items():
            out.insert(tk.END, f"{sentiment_type}: {count}\n")

    except FileNotFoundError:
        messagebox.showerror("Error", "passenger_registration.csv not found! Enter data first.")
    except Exception as e:
        messagebox.showerror("Error", f"Error: {str(e)}")

root = tk.Tk()
root.title("Feedback Analysis")
root.geometry("700x500")

tk.Button(root, text="Re-Analyze Feedback", command=analyze, font=("Arial", 12, "bold"),
          bg="#27ae60", fg="white", padx=10, pady=8).pack(pady=10)
out = scrolledtext.ScrolledText(root, width=85, height=25, font=("Arial", 10))
out.pack()

root.after(100, analyze)
root.mainloop()
```

## Feedback Analysis - Window



```
 Extraction (NER) output
 Insights plots (4-chart dashboard)

 know if you want any changes!

s



Users\praga\Desktop\Passport_Travel_Readiness\Passport_Travel_Readiness_Project.


now. Let me know what you think!

is first home page code - then ss of theh home page pop up the the csv generatec

reenshot → CSV, module by module. Let me capture screenshots of each window and

praga\Desktop\Passport_Travel_Readiness" && venv/Scripts/python -c "
t ImageGrab…)
able
e

 a script to launch each window, screenshot it, and then build the PDF.

f.py)
o _generate_pdf.py
ss, sys, time, os, ctypes
ort wintypes
 ImageGrab
s pd
t FPDF
```

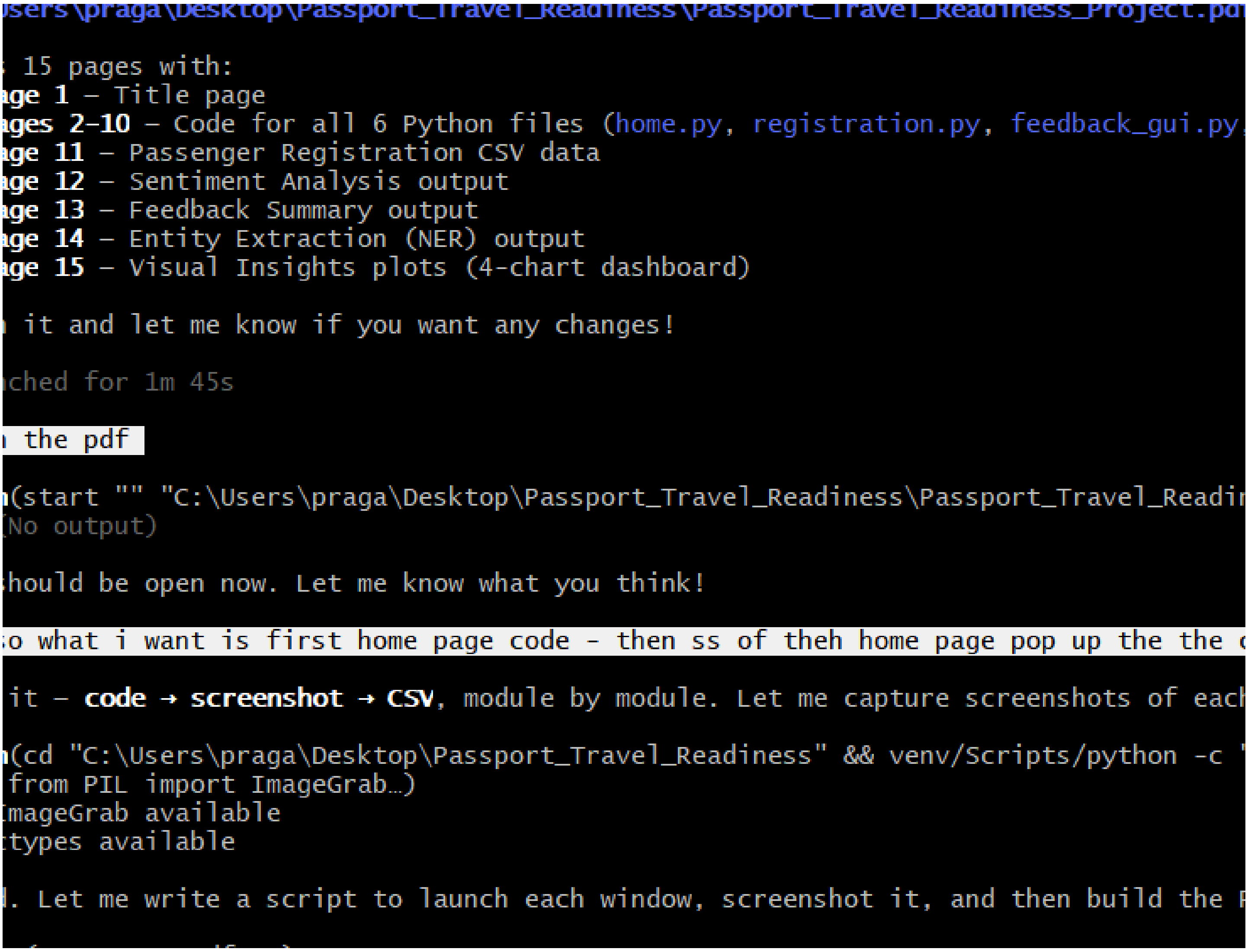
# sentiment_analysis.csv

| Full_Name | Destination_Country | Travel_Feedback | Sentiment |
|---|---|---|---|
| Rahul Sharma | France | Travelling with family was s | Positive |
| Ananya Verma | Japan | Solo travel experience was w | Neutral |
| Mohit Khan | UAE | Business travel was efficien | Neutral |
| Sneha Kulkarni | Germany | Couple travel was comfortabl | Positive |
| Amit Patel | USA | Group travel took longer due | Negative |
| Pooja Nair | Canada | Student travel process was s | Positive |
| Rohit Mehta | UK | Travelling with a child requ | Neutral |
| Neha Gupta | Australia | Elderly assistance was provi | Positive |
| Kunal Shah | Singapore | Fast check-in and smooth ver | Positive |
| Kavya Iyer | Italy | Family travel was good but b | Positive |

# summary_gui.py

```
import tkinter as tk
from tkinter import scrolledtext
import pandas as pd

keywords = [
    "smooth","delay","security","baggage",
    "slow","quick","helpful","stressful","waiting"
]

def summarize(text):
    text = str(text).lower()
    found = [k for k in keywords if k in text]
    if not found:
        return "General travel experience."
    return "Key points: " + ", ".join(found)

def generate():
    df = pd.read_csv("passenger_registration.csv")
    df["Summary"] = df["Travel_Feedback"].fillna("").apply(summarize)
    df.to_csv("feedback_summary.csv", index=False)

    out.delete(1.0, tk.END)
    for i, r in df.iterrows():
        out.insert(tk.END,
            f"Original:\n{r['Travel_Feedback']}\n"
            f"Summary:\n{r['Summary']}\n\n")

root = tk.Tk()
root.title("Text Summarization")
root.geometry("720x520")

tk.Button(root, text="Re-Generate Summary", command=generate).pack(pady=10)
out = scrolledtext.ScrolledText(root, width=85, height=25)
out.pack()

root.after(100, generate)
root.mainloop()
```

# Text Summarization - Window



```
sers\praga\Desktop\Passport_Travel_Readiness\Passport_Travel_Readiness_Project.pdf

 15 pages with:
age 1 – Title page
ages 2-10 – Code for all 6 Python files (home.py, registration.py, feedback_gui.py,
age 11 – Passenger Registration CSV data
age 12 – Sentiment Analysis output
age 13 – Feedback Summary output
age 14 – Entity Extraction (NER) output
age 15 – Visual Insights plots (4-chart dashboard)

 it and let me know if you want any changes!

ched for 1m 45s

 the pdf

(start "" "C:\Users\praga\Desktop\Passport_Travel_Readiness\Passport_Travel_Readin
(No output)

hould be open now. Let me know what you think!

o what i want is first home page code - then ss of theh home page pop up the the c

 it – code → screenshot → CSV, module by module. Let me capture screenshots of each

(cd "C:\Users\praga\Desktop\Passport_Travel_Readiness" && venv/Scripts/python -c "
 from PIL import ImageGrab…)
ImageGrab available
types available

. Let me write a script to launch each window, screenshot it, and then build the P
```

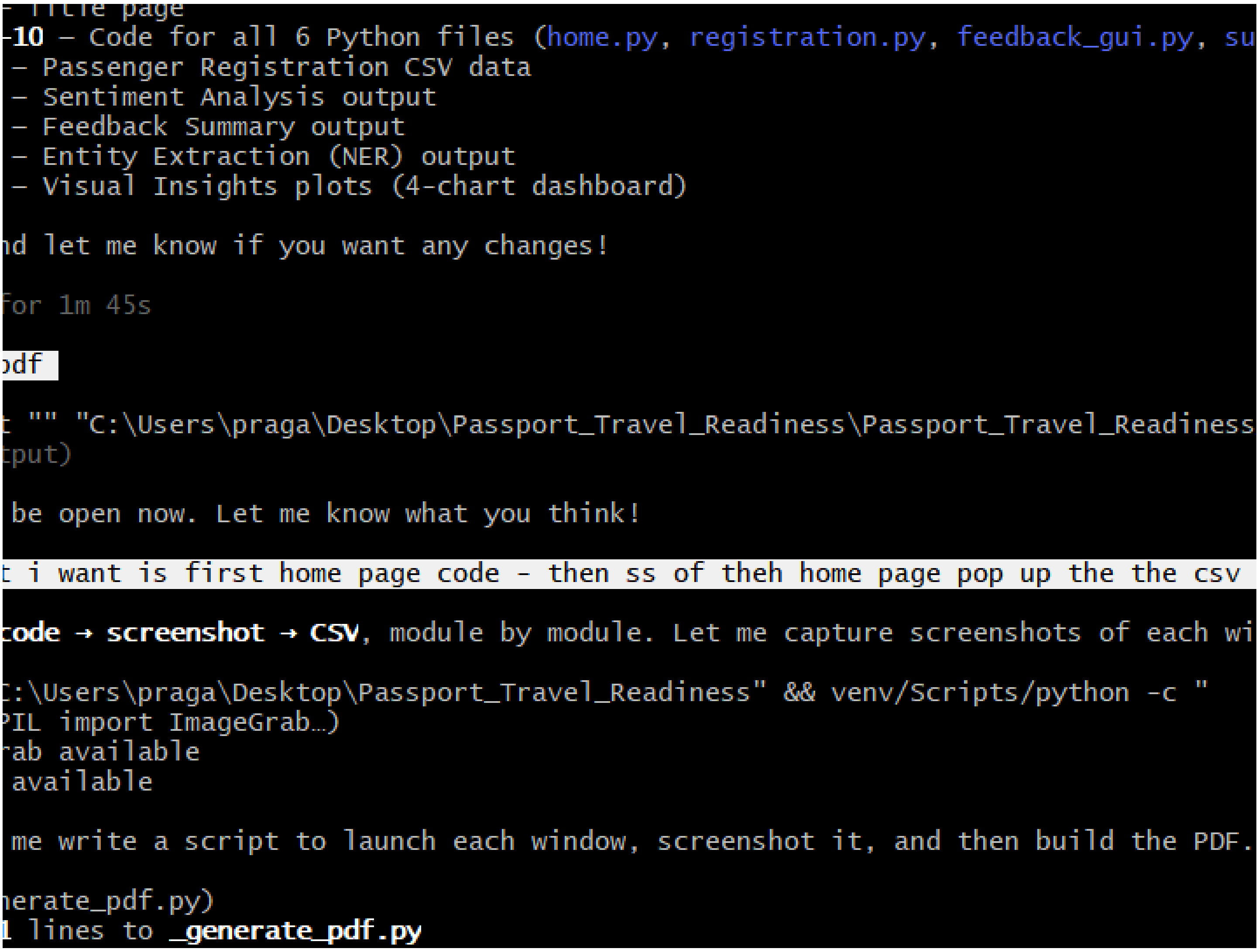
# feedback_summary.csv

| Full_Name | Destination_Country | Travel_Feedback | Summary |
|---|---|---|---|
| Rahul Sharma | France | Travelling with family was s | Key points: smooth, security |
| Ananya Verma | Japan | Solo travel experience was w | Key points: helpful |
| Mohit Khan | UAE | Business travel was efficien | Key points: baggage |
| Sneha Kulkarni | Germany | Couple travel was comfortabl | Key points: quick |
| Amit Patel | USA | Group travel took longer due | Key points: baggage |
| Pooja Nair | Canada | Student travel process was s | Key points: smooth |
| Rohit Mehta | UK | Travelling with a child requ | Key points: security |
| Neha Gupta | Australia | Elderly assistance was provi | Key points: waiting |
| Kunal Shah | Singapore | Fast check-in and smooth ver | Key points: smooth |
| Kavya Iyer | Italy | Family travel was good but b | Key points: baggage, slow |

# ner_gui.py

```
import tkinter as tk
from tkinter import scrolledtext, messagebox
import pandas as pd
import re

try:
    import spacy
    nlp = spacy.load("en_core_web_sm")
except ImportError:
    messagebox.showerror("Error", "Spacy not installed. Install with: pip install spacy")
    nlp = None

def run_ner():
    try:
        if nlp is None:
            messagebox.showerror("Error", "Spacy model not loaded. Restart the application.")
            return

        df = pd.read_csv("passenger_registration.csv")

        if len(df) == 0:
            messagebox.showwarning("No Data", "Please enter travel data first!")
            return

        rows = []

        for _, r in df.iterrows():
            text = f"{r.get('Full_Name', '')} travelled to {r.get('Destination_Country', '')}. {r.get('Travel_Feedback', '')}"
            doc = nlp(text)

            for e in doc.ents:
                rows.append([e.text, e.label_])

            if 'Passport_Number' in r and pd.notna(r['Passport_Number']):
                for p in re.findall(r"[A-Z0-9]{6,10}", str(r["Passport_Number"])):
                    rows.append([p, "PASSPORT_NUMBER"])

        if not rows:
            messagebox.showinfo("Info", "No entities found in the data")
            out.delete(1.0, tk.END)
            out.insert(tk.END, "No named entities detected in travel data.")
            return

        ner_df = pd.DataFrame(rows, columns=["Entity", "Type"])

        try:
            ner_df.to_csv("entity_extraction.csv", index=False)
        except:
            pass  # File might be locked, continue anyway

        out.delete(1.0, tk.END)
        out.insert(tk.END, "?? NAMED ENTITY RECOGNITION (NER) RESULTS ??\n\n")
        for e, t in rows:
            out.insert(tk.END, f"{e} ? {t}\n")

        messagebox.showinfo("Success", "NER + Regex completed")

    except FileNotFoundError:
        messagebox.showerror("Error", "passenger_registration.csv not found!")
    except Exception as e:
        messagebox.showerror("Error", str(e))

root = tk.Tk()
root.title("NER + Regex")
root.geometry("700x500")

tk.Button(root, text="Re-Run NER + Regex", command=run_ner).pack(pady=10)
out = scrolledtext.ScrolledText(root, width=85, height=25)
out.pack()

root.after(100, run_ner)
root.mainloop()
```

## Entity Extraction (NER) - Window



```
- Title page
-10 — Code for all 6 Python files (home.py, registration.py, feedback_gui.py, su
 — Passenger Registration CSV data
 — Sentiment Analysis output
 — Feedback Summary output
 — Entity Extraction (NER) output
 — Visual Insights plots (4-chart dashboard)

nd let me know if you want any changes!

for 1m 45s

pdf

t "" "C:\Users\praga\Desktop\Passport_Travel_Readiness\Passport_Travel_Readiness
tput)

 be open now. Let me know what you think!

t i want is first home page code - then ss of theh home page pop up the the csv

code → screenshot → CSV, module by module. Let me capture screenshots of each wi

C:\Users\praga\Desktop\Passport_Travel_Readiness" && venv/Scripts/python -c "
PIL import ImageGrab…)
rab available
 available

 me write a script to launch each window, screenshot it, and then build the PDF.

nerate_pdf.py)
1 lines to _generate_pdf.py
```

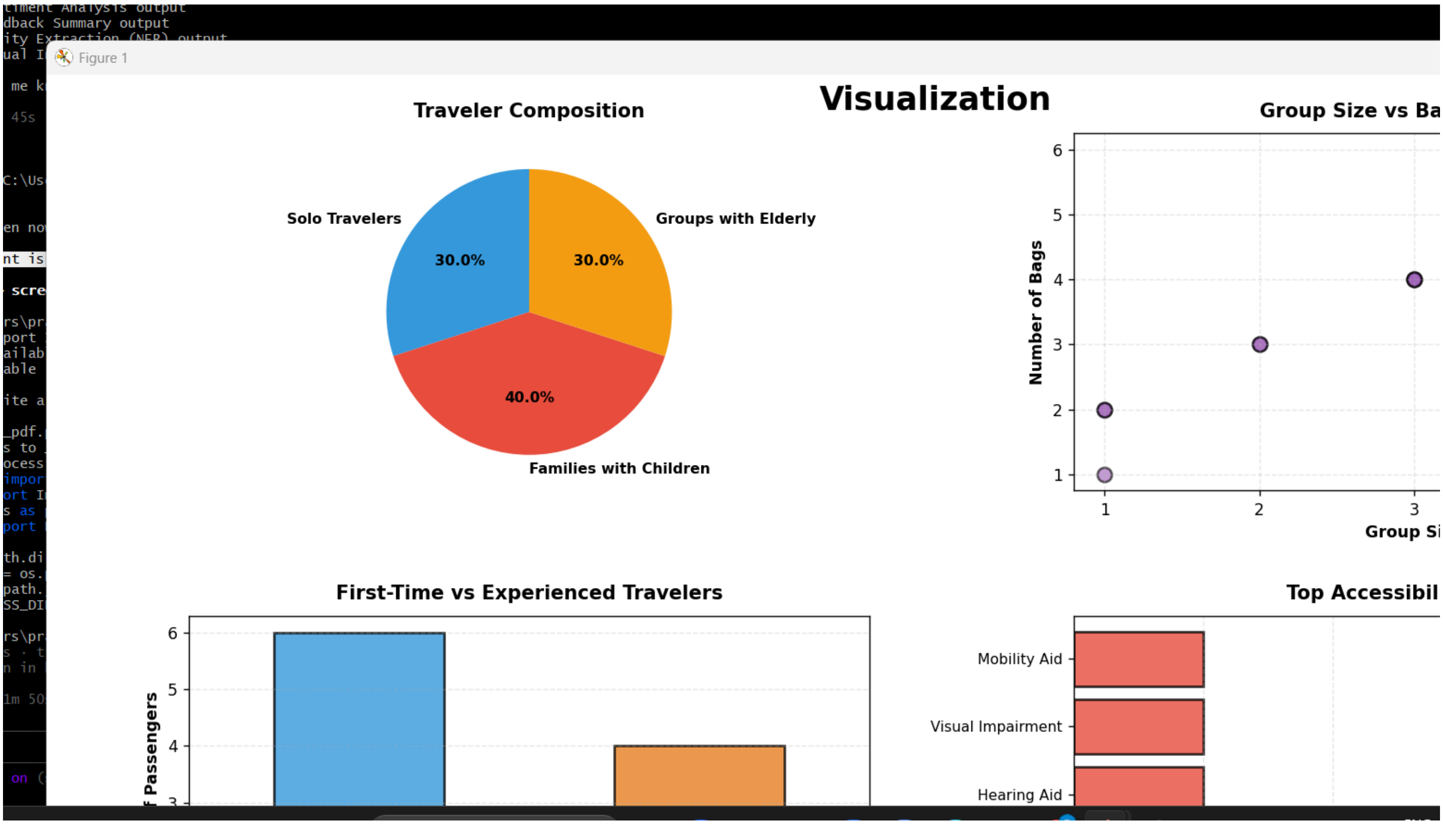
# entity_extraction.csv

| Entity | Type |
|---|---|
| Rahul Sharma | PERSON |
| France | GPE |
| P1234567 | PASSPORT_NUMBER |
| Ananya | GPE |
| Japan | GPE |
| P2345678 | PASSPORT_NUMBER |
| Mohit Khan | PERSON |
| UAE | GPE |
| P3456789 | PASSPORT_NUMBER |
| Sneha Kulkarni | PERSON |
| Germany | GPE |
| P4567890 | PASSPORT_NUMBER |
| USA | GPE |
| P5678901 | PASSPORT_NUMBER |
| Pooja Nair | PERSON |
| Canada | GPE |
| P6789012 | PASSPORT_NUMBER |
| Rohit Mehta | PERSON |
| UK | GPE |
| P7890123 | PASSPORT_NUMBER |
| Neha Gupta | PERSON |
| Australia | GPE |
| P8901234 | PASSPORT_NUMBER |
| Kunal Shah | PERSON |
| Singapore | GPE |
| P9012345 | PASSPORT_NUMBER |
| Kavya Iyer | PERSON |
| Italy | GPE |
| P0123456 | PASSPORT_NUMBER |

## plot_gui.py

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

try:
    df = pd.read_csv("passenger_registration.csv")

    if len(df) == 0:
        print("No data found. Please enter travel data first!")
        exit()

    # Data preprocessing
    df['Group_Size'] = pd.to_numeric(df['Group_Size'], errors='coerce').fillna(1).astype(int)
    df['Children_Count'] = pd.to_numeric(df['Children_Count'], errors='coerce').fillna(0).astype(int)
    df['Elderly_Count'] = pd.to_numeric(df['Elderly_Count'], errors='coerce').fillna(0).astype(int)
    df['Number_of_Bags'] = pd.to_numeric(df['Number_of_Bags'], errors='coerce').fillna(0).astype(int)
    df['First_Time_Traveler'] = df['First_Time_Traveler'].fillna('No')
    df['Accessibility_Needs'] = df['Accessibility_Needs'].fillna('None')

    # Create 2x2 subplot grid
    fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(16, 12))
    fig.suptitle("Visualization", fontsize=20, fontweight='bold', y=0.99)

    # PLOT 1: TRAVELER COMPOSITION
    solo = len(df[df["Group_Size"] == 1])
    family = len(df[df["Children_Count"] > 0])
    elderly = len(df[df["Elderly_Count"] > 0])

    colors1 = ["#3498db", "#e74c3c", "#f39c12"]
    ax1.pie(
        [solo, family, elderly],
        labels=["Solo Travelers", "Families with Children", "Groups with Elderly"],
        autopct="%1.1f%%",
        colors=colors1,
        startangle=90,
        textprops={'fontsize': 9, 'weight': 'bold'}
    )
    ax1.set_title("Traveler Composition", fontsize=12, fontweight='bold', pad=10)

    # PLOT 2: GROUP SIZE vs BAGS
    ax2.scatter(df["Group_Size"], df["Number_of_Bags"], s=80, alpha=0.6, color='#9b59b6', edgecolors='black', linewidth=1.5)
    ax2.set_xlabel("Group Size", fontsize=10, fontweight='bold')
    ax2.set_ylabel("Number of Bags", fontsize=10, fontweight='bold')
    ax2.set_title("Group Size vs Baggage Load", fontsize=12, fontweight='bold', pad=10)
    ax2.set_xticks(sorted(df["Group_Size"].unique()))
    ax2.grid(True, alpha=0.3, linestyle='--')

    # PLOT 3: FIRST-TIME vs EXPERIENCED TRAVELERS
    counts = df["First_Time_Traveler"].value_counts()
    colors3 = ['#3498db', '#e67e22'][:len(counts)]
    counts.plot(kind="bar", ax=ax3, color=colors3, alpha=0.8, edgecolor='black', linewidth=1.5)
    ax3.set_xlabel("Traveler Type", fontsize=10, fontweight='bold')
    ax3.set_ylabel("Number of Passengers", fontsize=10, fontweight='bold')
    ax3.set_title("First-Time vs Experienced Travelers", fontsize=12, fontweight='bold', pad=10)
    ax3.set_xticklabels(ax3.get_xticklabels(), rotation=0, ha='center', fontsize=9)
    ax3.grid(True, alpha=0.3, axis='y', linestyle='--')

    # PLOT 4: ACCESSIBILITY NEEDS
    access = df["Accessibility_Needs"].value_counts().head(8)
    ax4.barh(range(len(access)), access.values, color='#e74c3c', alpha=0.8, edgecolor='black', linewidth=1.5)
    ax4.set_yticks(range(len(access)))
    ax4.set_yticklabels(access.index, fontsize=9)
    ax4.set_xlabel("Number of Passengers", fontsize=10, fontweight='bold')
    ax4.set_title("Top Accessibility Needs", fontsize=12, fontweight='bold', pad=10)
    ax4.grid(True, alpha=0.3, axis='x', linestyle='--')

    # Adjust spacing to prevent label overlap
    plt.subplots_adjust(top=0.94, bottom=0.08, left=0.08, right=0.96, hspace=0.35, wspace=0.3)

    # Remove toolbar for cleaner look
    mng = plt.get_current_fig_manager()
    if hasattr(mng, 'toolbar'):
        mng.toolbar.pack_forget()
    plt.show()

except FileNotFoundError:
    print("Error: passenger_registration.csv not found!")
except Exception as e:
    print(f"Error: {str(e)}")
```

# Visual Insights - Plots



```
timent Analysis output
dback Summary output
ity Extraction (NER) output
ual I
 me k
 45s

C:\Us

en no
nt is
 scre
rs\pr
port
ailab
able

ite a

_pdf.
s to
ocess
impor
ort I
s as
port

th.di
= os.
path.
SS_DI

rs\pr
s · t
n in

1m 50

 on (
```