pandas

matplotlib

seaborn

streamlit

import streamlit as st

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from datetime import datetime, timedelta

from collections import defaultdict

# Function to detect schedule flaws

def detect\_schedule\_flaws(schedule\_df):

flaws = []

# Check for overlapping tasks

for i, row\_i in schedule\_df.iterrows():

for j, row\_j in schedule\_df.iterrows():

if i != j:

overlap = max(0, min(row\_i['End Date'], row\_j['End Date']) -

max(row\_i['Start Date'], row\_j['Start Date']))

if overlap > 0:

flaws.append({

'type': 'Overlap',

'task1': row\_i['Task ID'],

'task2': row\_j['Task ID'],

'overlap\_days': overlap.days

})

# Check for negative durations

negative\_durations = schedule\_df[schedule\_df['Duration (days)'] <= 0]

for \_, row in negative\_durations.iterrows():

flaws.append({

'type': 'Negative Duration',

'task\_id': row['Task ID'],

'duration': row['Duration (days)']

})

return flaws

# Main app function

def main():

st.title("Construction Schedule Detector")

# File uploader

uploaded\_file = st.file\_uploader("Upload your schedule file (CSV or Excel)",

type=['csv', 'xlsx', 'xls'])

if uploaded\_file is not None:

try:

# Read file

if uploaded\_file.name.endswith('.xlsx') or uploaded\_file.name.endswith('.xls'):

schedule\_df = pd.read\_excel(uploaded\_file)

else:

schedule\_df = pd.read\_csv(uploaded\_file)

# Ensure date columns are properly formatted

schedule\_df['Start Date'] = pd.to\_datetime(schedule\_df['Start Date'])

schedule\_df['End Date'] = pd.to\_datetime(schedule\_df['End Date'])

# Detect flaws

flaws = detect\_schedule\_flaws(schedule\_df)

# Show results

st.subheader("Detection Results")

if not flaws:

st.success("✅ No major flaws detected in your schedule!")

else:

st.warning("⚠ Flaws detected:")

for flaw in flaws:

if flaw['type'] == 'Overlap':

st.write(f"- Tasks {flaw['task1']} and {flaw['task2']} overlap by {flaw['overlap\_days']} days")

elif flaw['type'] == 'Negative Duration':

st.write(f"- Task {flaw['task\_id']} has negative duration ({flaw['duration']} days)")

# Show schedule visualization

st.subheader("Schedule Timeline")

fig, ax = plt.subplots(figsize=(12, 6))

for \_, row in schedule\_df.iterrows():

ax.barh(row['Task ID'],

width=(row['End Date'] - row['Start Date']).days,

left=row['Start Date'],

height=0.5)

ax.set\_xlabel('Date')

ax.set\_ylabel('Tasks')

ax.set\_title('Construction Schedule Timeline')

ax.grid(True, axis='x', linestyle='--', alpha=0.7)

plt.tight\_layout()

st.pyplot(fig)

except Exception as e:

st.error(f"Error processing file: {str(e)}")

if \_name\_ == "\_main\_":

main()