```
import pandas as pd
import numpy as np
import streamlit as st

data = pd.read_excel("Delhi.xlsx", usecols=('#Data','Unnamed: 1'))
data = data.rename(columns={'#Data':'Date', 'Unnamed: 1': 'PM'})
data = data.drop([0,1])
data.sort_values("Date", inplace = True)
data.reset_index(drop=True, inplace=True)
data.dropna(inplace=True)
data[['PM']] = data[['PM']].replace('-', np.nan)
data['PM'] = data.PM.interpolate(method = 'nearest')
data1 = data.copy()


import datetime as dt
data1['Date'] =  pd.to_datetime(data1['Date'],format='%Y-%m-%d %H:%M:%S')
data1["year"] = data1.Date.dt.strftime("%Y")
data1["month"] = data1.Date.dt.strftime("%b")
data1["Day"] = data1.Date.dt.strftime("%d")
data1['hour'] = data1.Date.dt.strftime("%H")
data1['dayofweek'] = data1.Date.dt.day_name()
data1['month'] = pd.Categorical(data1['month'], categories=data1.month.unique())
data1['dayofweek'] = pd.Categorical(data1['dayofweek'],
categories=data1.dayofweek.unique())

import statsmodels.api as sma
from statsmodels.tsa.statespace.sarimax import SARIMAX
mod_full = sma.tsa.statespace.SARIMAX(data1['PM'], trend='ct', order=(2,1,1),
seasonal_order=(2,0,2,4))
results_full = mod_full.fit(disp=-1)
print (results_full.summary())

x = np.arange(2375,2400,1)
future = pd.DataFrame(index=x, columns= data1.columns)
data1 = pd.concat([data1, future])

data1['forecast'] = results_full.predict (2365, 2400, dynamic=False, typ='levels')
data1['forecast'] = data1['forecast'].shift(-1)
datax = data1.loc[:,('PM','forecast')]

st.title('Delhi Particulate matter')
from PIL import Image
st.sidebar.header("Heat Map Of 4 Months")
image = Image.open('heat map.png')
st.sidebar.image(image, caption='Heat Map')

st.sidebar.header("Box Plot Hourly")
image = Image.open('PM_Hour.png')
```

```
st.sidebar.image(image, caption='Box Plot Hourly')

st.sidebar.header("Line Plot Of 4 Months")
image = Image.open('line_plot.png')
st.sidebar.image(image, caption='Line Plot Of 4 Months')

st.sidebar.header("ACF")
image = Image.open('ACF.png')
st.sidebar.image(image, caption='Auto-correlation')

st.sidebar.header("PACF")
image = Image.open('PACF.png')
st.sidebar.image(image, caption='Partial Auto-correlation')

st.sidebar.header("RMSE")
image = Image.open('RMSE.png')
st.sidebar.image(image, caption='RMSE comparison')
st.subheader('forecast for 24 hours')
st.line_chart(datax[['PM', 'forecast']][2000:], width=24, height=6)
# datax[2370:2397]

import datetime
from dateutil.relativedelta import relativedelta, MO
start = datetime.datetime.strptime("2018-04-20 01:00:00", "%Y-%m-%d %H:%M:%S")
date_list = [start + relativedelta(hours=x) for x in range(0,25)]
df = pd.DataFrame({'x':x, 'y':date_list})

st.header('User Input Parameters')
date_and_hour = st.selectbox('date / Time',(df.y))
a = df['x'][df['y'] == date_and_hour].values[0]
st.write(datax['forecast'][a])
```