In [ ]:

```python
import streamlit as st
import pandas as pd
from datetime import datetime
import os

File_Name = 'phone_directory.xlsx'


# -------------------- Initialize Excel File --------------------
def init_excel():
    if not os.path.exists(File_Name):
        contacts_df = pd.DataFrame(columns=['Name', 'Phone'])
        deleted_df = pd.DataFrame(columns=['Name', 'Phone', 'Deleted_On'])

        with pd.ExcelWriter(File_Name, engine='openpyxl') as writer:
            contacts_df.to_excel(writer, sheet_name='Contacts', index=False)
            deleted_df.to_excel(writer, sheet_name='Deleted_Contacts', index=False)


# -------------------- Load Data --------------------
def load_contacts():
    return pd.read_excel(File_Name, sheet_name='Contacts')


def load_deleted_contacts():
    return pd.read_excel(File_Name, sheet_name='Deleted_Contacts')


# -------------------- Save Data --------------------
def save_contacts(df):
    with pd.ExcelWriter(File_Name, engine='openpyxl', mode='a',
                        if_sheet_exists='replace') as writer:
        df.to_excel(writer, sheet_name='Contacts', index=False)


def save_deleted_contacts(df):
    with pd.ExcelWriter(File_Name, engine='openpyxl', mode='a',
                        if_sheet_exists='replace') as writer:
        df.to_excel(writer, sheet_name='Deleted_Contacts', index=False)
```

```python
# -------------------- Phone Validation --------------------
def valid_phone(phone):
    return phone.isdigit() and 7 <= len(phone) <= 15


# -------------------- Streamlit UI --------------------
st.set_page_config(page_title="Phone Directory", layout='centered')
st.title("📱 Phone Directory Manager")

init_excel()

menu = st.sidebar.radio(
    "Menu",
    ["Add Contact", "Search Contact", "Delete Contact",
     "View All Contacts", "View Deleted Contacts"]
)

contact_df = load_contacts()
deleted_df = load_deleted_contacts()

# ============================================================
# ADD CONTACT
# ============================================================
if menu == "Add Contact":
    st.subheader("➕ Add New Contact")

    name = st.text_input("Contact Name")
    phone = st.text_input("Phone Number")

    if st.button("Save Contact"):
        if not name or not phone:
            st.warning("Please enter both name and phone number.")

        elif not valid_phone(phone):
            st.error("Phone number must contain only digits (7–15 digits).")

        elif name in contact_df['Name'].values:
            st.error("Contact name already exists.")
```

```
        elif phone in contact_df['Phone'].values:
            st.error("Phone number already exists.")

        else:
            new_row = pd.DataFrame({'Name': [name], 'Phone': [phone]})
            contact_df = pd.concat([contact_df, new_row], ignore_index=True)
            save_contacts(contact_df)
            st.success("Contact added successfully!")


# ============================================================
# SEARCH CONTACT
# ============================================================
elif menu == "Search Contact":
    st.subheader("🔍 Search Contact")

    search_term = st.text_input("Enter Name or Phone")

    if st.button("Search"):
        if search_term:
            result = contact_df[
                (contact_df['Name'].str.contains(search_term, case=False, na=False)) |
                (contact_df['Phone'].astype(str).str.contains(search_term))
            ]

            if not result.empty:
                st.success(f"{len(result)} contact(s) found:")
                st.dataframe(result)
            else:
                st.error("No contact found.")
        else:
            st.warning("Please enter search term.")


# ============================================================
# DELETE CONTACT
# ============================================================
elif menu == "Delete Contact":
    st.subheader("🗑 Delete Contact")

    delete_name = st.selectbox("Select Contact to Delete", contact_df['Name'])
```

```
    if st.button("Delete"):
        if delete_name:
            deleted_contact = contact_df[contact_df['Name'] == delete_name]

            if not deleted_contact.empty:
                deleted_contact['Deleted_On'] = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

                # Remove from main contacts
                contact_df = contact_df[contact_df['Name'] != delete_name]

                # Add to deleted sheet
                deleted_df = pd.concat([deleted_df, deleted_contact], ignore_index=True)

                save_contacts(contact_df)
                save_deleted_contacts(deleted_df)

                st.success("Contact deleted successfully!")


# ============================================================
# VIEW ALL CONTACTS
# ============================================================
elif menu == "View All Contacts":
    st.subheader("📋 All Contacts")

    if contact_df.empty:
        st.info("No contacts available.")
    else:
        st.dataframe(contact_df)
        st.write(f"Total Contacts: {len(contact_df)}")


# ============================================================
# VIEW DELETED CONTACTS
# ============================================================
elif menu == "View Deleted Contacts":
    st.subheader("🗑️ Deleted Contacts History")

    if deleted_df.empty:
        st.info("No deleted contacts.")
```

```
    else:
        st.dataframe(deleted_df)
        st.write(f"Total Deleted Contacts: {len(deleted_df)}")
```