



🏛️ Indian Penal Code Chatbot: Your Legal Assistant 🧑

```
In [ ]: # Step-1: Document Loading & Text Extraction
```

```
In [ ]: # Step-1: Document Loading
```

```
# Install required dependencies
!pip install PyPDF2 langchain wget chromadb openai

import wget
from PyPDF2 import PdfReader

# Download the IPC PDF file
url = "https://raw.githubusercontent.com/venkatareddykona/Datasets/master/
file_name = wget.download(url)

# Load and extract text
reader = PdfReader(file_name)
full_text = ""

for page in reader.pages:
    full_text += page.extract_text()

# Print first 1000 characters
print("First 1000 characters:\n")
print(full_text[:1000])

# Print statistics
num_lines = full_text.count("\n")
num_words = len(full_text.split())
num_characters = len(full_text)

print("\n--- Document Statistics ---")
print("Lines:", num_lines)
print("Words:", num_words)
print("Characters:", num_characters)
```

```
In [ ]: # Step-2: Split the Text into Chunks
```

```
In [ ]: # Step-2: Chunking the data
```

```
from langchain.text_splitter import RecursiveCharacterTextSplitter

text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=1000,
    chunk_overlap=200
)

chunks = text_splitter.split_text(full_text)

print("Total Chunks:", len(chunks))
print("\nFirst Chunk:\n", chunks[0])
```

```
In [ ]: # Step-3: Creating Embeddings & Storing in Vector Store
```

```
In [ ]: # Step-3: Create Embeddings & Store Vector DB
```

```
from langchain.embeddings import OpenAIEMBEDDINGS
from langchain.vectorstores import Chroma

# Create embeddings
embeddings = OpenAIEMBEDDINGS()

# Create persistent vector store
db = Chroma.from_texts(chunks, embeddings, persist_directory="IPC_db")

db.persist()

print("Vector Store created successfully!")
```

```
In [ ]: # Step-4: Retrieval + Conversation Setup
```

```
In [ ]: # Step-4: Retrieval and Conversation Chain
```

```
from langchain.chat_models import ChatOpenAI
from langchain.memory import ConversationBufferMemory
from langchain.chains import ConversationalRetrievalChain

# Load existing DB
vectorstore = Chroma(persist_directory="IPC_db", embedding_function=embeddings)

# Create retriever
retriever = vectorstore.as_retriever()

# Test query
query = "What is the section related to take part in an unlawful assembly or r
result = retriever.get_relevant_documents(query)

# Print retrieved documents
for doc in result:
    print("\nRelevant Text:\n", doc.page_content)

# Setup Conversation AI
llm = ChatOpenAI(model="gpt-3.5-turbo", temperature=0)

memory = ConversationBufferMemory(memory_key="chat_history", return_messages=True)

conversational_RAG = ConversationalRetrievalChain.from_llm(
    llm=llm,
    retriever=retriever,
    memory=memory
)

print("Conversational Chain Ready!")
```

```
In [ ]: # Step-5: User Conversation
```

```
In [ ]: # Step-5: Chatbot Loop
```

```
print("\nIPC Chatbot Ready! Type 'quit' to exit.\n")

while True:
    user_input = input("You: ")

    if user_input.lower() == "quit":
        print("Chatbot Ended. Goodbye!")
        break

    response = conversational_RAG({"question": user_input})
    print("\nBot:", response["answer"], "\n")
```

```
In [ ]: # Example Queries You Can Try:
```

```
In [ ]: What is the section related to take part in an unlawful assembly or riot?
What is the punishment for that?
What is the Punishment for using a false property mark?
Which section deals with that?
Which section deals with Counterfeiting currency-notes or bank-notes?
Tell me more about the section and punishment for Counterfeiting currency-note
```