

Legal Consultant AI Chatbot - Complete Setup Guide

Project Overview

This is a complete legal consultant chatbot that uses:

- **InLegalBERT** for understanding Indian legal text
- **RAG (Retrieval Augmented Generation)** for up-to-date legal information
- **Groq Llama 3.1** for generating human-readable advice
- **Streamlit** for the web interface

Quick Start (5 Minutes)

Step 1: Install Dependencies

```
bash

# Install Python packages
pip install -r requirements.txt
```

Step 2: Project Structure

Create the following folder structure:

```
legal-chatbot/
|
├── app.py           # Main application (provided)
├── requirements.txt  # Dependencies (provided)
├── legal_data/      # Your legal documents folder
|   ├── sample_ipc.json  # Sample data (provided)
|   └── [add more JSON/TXT files]
└── vector_db/       # Auto-created for storing index
```

Step 3: Add Legal Documents

Create `legal_data` folder and add the sample data:

```
bash

mkdir legal_data
# Copy sample_ipc.json to legal_data/
```

Document Format (JSON):

```
json

[
  {
    "text": "Your legal text here...",
    "metadata": {
      "source": "IPC",
      "section": "420",
      "category": "criminal"
    }
  }
]
```

Or use TXT files: Just put `.txt` files in `legal_data/` folder with legal content.

Step 4: Get Groq API Key (FREE)

1. Go to: <https://console.groq.com>
2. Sign up for free account
3. Create API key
4. Copy the key (starts with `gsk_...`)

Step 5: Run the Application

```
bash

streamlit run app.py
```

The app will open in your browser at `http://localhost:8501`

Step 6: Initialize the System

1. Paste your Groq API key in the sidebar
 2. Click "Initialize System"
 3. Wait for the system to load (1-2 minutes first time)
 4. Start chatting!
-

Adding Your Own Legal Documents

Option 1: JSON Format (Recommended)

Create files like `legal_data/ipc_sections.json`:

```
json
[
  {
    "text": "Section 302 IPC: Whoever commits murder shall be punished with death or imprisonment for life...",
    "metadata": {
      "source": "Indian Penal Code",
      "section": "302",
      "category": "criminal",
      "type": "bare_act"
    }
  },
  {
    "text": "Section 420 IPC: Whoever cheats and thereby dishonestly induces...",
    "metadata": {
      "source": "Indian Penal Code",
      "section": "420",
      "category": "criminal",
      "type": "bare_act"
    }
  }
]
```

Option 2: Plain Text Format

Create files like `legal_data/consumer_act.txt`:

Consumer Protection Act 2019

A consumer can file a complaint within two years from the date of cause of action.

District Commission has jurisdiction for claims up to Rs 1 crore.

State Commission for claims between Rs 1 crore to Rs 10 crore.

National Commission for claims exceeding Rs 10 crore.

Categories to Include

Add documents for these legal categories:

1. Criminal Law

- IPC sections
- CrPC procedures
- Criminal case laws

2. Civil Law

- CPC provisions
- Contract Act
- Property laws

3. Consumer Law

- Consumer Protection Act
- E-commerce rules
- Consumer forums

4. Labor Law

- Industrial Disputes Act
- Minimum Wages Act
- ESI/PF rules

5. Cyber Law

- IT Act 2000
- Data protection
- Cyber crimes

6. Family Law

- Hindu Marriage Act
- Divorce procedures
- Maintenance laws

System Configuration

Edit these settings in `app.py` if needed:

```
python
```

class Config:

```
GROQ_MODEL = "llama-3.1-70b-versatile" # Best free model
EMBEDDING_MODEL = "law-ai/InLegalBERT"
DATA_FOLDER = "legal_data"           # Your documents folder
VECTOR_DB_PATH = "vector_db"         # Vector database location
MAX_RETRIEVAL = 5                     # Number of relevant docs to retrieve
CHUNK_SIZE = 512                      # Characters per chunk
```

How to Use the Chatbot

Example Queries:

1. Criminal Law:

- "What is the punishment for cheating under IPC?"
- "How do I file an FIR?"
- "Can I get bail in a non-bailable offense?"

2. Consumer Rights:

- "I bought a defective product. What can I do?"
- "How do I file a consumer complaint?"
- "What is the time limit for consumer cases?"

3. Labor Law:

- "I was fired without notice. Is this legal?"
- "What are my rights regarding retrenchment?"
- "How much notice period is required?"

4. Cyber Crime:

- "Someone is using my photos without permission online"
 - "I received a threatening email. What should I do?"
 - "How to report online fraud?"
-

Updating Legal Information (RAG Feature)

To Add New Laws/Amendments:

1. Create new JSON file: `legal_data/new_amendment_2024.json`

```
json
[
  {
    "text": "Consumer Protection Amendment 2024: E-commerce platforms now liable for defective products...",
    "metadata": {
      "source": "Consumer Protection Act",
      "type": "amendment",
      "year": "2024",
      "effective_date": "2024-01-01"
    }
  }
]
```

2. In the Streamlit sidebar, check "Rebuild Vector Database"
3. Click "Initialize System" again
4. The new law is now included!

No retraining needed! Just add files and rebuild index.

Troubleshooting

Issue: "No documents found in legal_data/"

Solution: Make sure you have created `legal_data` folder with at least one `.json` or `.txt` file.

```
bash

mkdir legal_data
# Add sample_ipc.json to this folder
```

Issue: "Error: Invalid API key"

Solution:

1. Check your Groq API key is correct (starts with `gsk_`)
2. Get new key from <https://console.groq.com>

3. Make sure you're using the free tier model: `llama-3.1-70b-versatile`

Issue: "Model loading takes too long"

Solution:

- First time: InLegalBERT download takes 2-3 minutes (1.5GB)
- Subsequent runs: Loads from cache (30 seconds)
- Use GPU if available for faster inference

Issue: "Out of memory"

Solution:

- Reduce `CHUNK_SIZE` to 256
- Reduce `MAX_RETRIEVAL` to 3
- Process fewer documents at once
- Use CPU instead of GPU for embedding

Issue: "Responses are not accurate"

Solution:

- Add more relevant legal documents to `legal_data/`
- Increase `MAX_RETRIEVAL` to 7-10
- Use better quality source documents
- Include more case laws and precedents



System Requirements

Minimum:

- **RAM:** 4GB
- **Storage:** 5GB free space
- **Python:** 3.8 or higher
- **Internet:** Required for first-time model download

Recommended:

- **RAM:** 8GB+

- **GPU:** Not required but speeds up embedding
 - **Storage:** 10GB+ for large document collections
-

API Limits (Groq Free Tier)

- **Requests:** 30 requests per minute
- **Tokens:** 14,400 tokens per minute
- **Models:** Llama 3.1 70B, Mixtral, Gemma

More than enough for development and testing!

Data Sources (Where to Get Legal Documents)

Free Sources:

1. Indian Kanoon (<https://indiankanoon.org>)

- 10M+ judgments
- Free API
- Download scripts available

2. India Code (<https://indiacode.nic.in>)

- All Central Acts
- Official government source
- Downloadable PDFs

3. Supreme Court (<https://main.sci.gov.in>)

- Recent judgments
- Daily orders

4. Legal Blogs & Websites

- LiveLaw
- Bar & Bench
- Legal Service India

Sample Scraping Script:

```
python
```



```
import requests
from bs4 import BeautifulSoup
import json

def scrape_indian_kanoon(query, num_results=10):
    documents = []
    url = f"https://indiankanoon.org/search/?formInput={query}"

    # Add your scraping logic here
    # Parse results and save to JSON

    with open('legal_data/scraped_cases.json', 'w') as f:
        json.dump(documents, f, indent=2)
```

Security & Privacy

Important Notes:

1. **API Keys:** Never commit API keys to Git
2. **User Data:** All chats are local, not stored externally
3. **Legal Disclaimer:** Always show disclaimer to users
4. **Professional Advice:** Encourage consulting lawyers for specific cases

Recommended Disclaimer:

"This AI provides general legal information based on Indian law.
It is not a substitute for professional legal advice.
For specific legal matters, please consult a qualified lawyer.
Laws may vary by jurisdiction and may have been updated."

Advanced Features (Future Enhancements)

1. Multi-language Support

Add Hindi/Tamil/Bengali translations

2. Voice Input

Integrate speech-to-text for queries

3. Document Upload

Let users upload their legal documents for analysis

4. Case Tracker

Track ongoing cases and deadlines

5. Lawyer Directory

Connect users with verified lawyers

Support

Common Commands:

```
bash

# Install dependencies
pip install -r requirements.txt

# Run application
streamlit run app.py

# Update dependencies
pip install --upgrade -r requirements.txt

# Clear cache
rm -rf vector_db/
rm -rf ~/.cache/huggingface/
```

File Structure After Setup:

```
legal-chatbot/
├── app.py           # Main application
├── requirements.txt # Python dependencies
├── legal_data/      # Your legal documents
│   ├── sample_ipc.json
│   ├── consumer_act.json
│   └── labor_laws.txt
├── vector_db/       # Auto-generated
│   ├── index.faiss  # Vector index
│   └── documents.json # Document metadata
└── SETUP_AND_RUN.md # This file
```

✓ Checklist

- ☐ Python 3.8+ installed
 - ☐ Created `legal_data` folder
 - ☐ Added sample legal documents
 - ☐ Installed requirements.txt
 - ☐ Got Groq API key
 - ☐ Ran `streamlit run app.py`
 - ☐ System initialized successfully
 - ☐ Tested with sample queries
-

🎓 Learning Resources

Understanding RAG:

- Retrieval: Find relevant documents from database
- Augmentation: Add documents to LLM prompt
- Generation: LLM creates answer using documents

Why InLegalBERT?

- Pre-trained on 100K+ Indian legal documents
- Understands legal terminology
- Better than generic BERT for legal text

Why Groq?

- Free tier with powerful models
 - Fast inference (10x faster than OpenAI)
 - Llama 3.1 70B available for free
-

📈 Performance Tips

1. More Documents = Better Answers

- Start with 50+ documents minimum
- Add 1000+ for production quality

2. Quality Over Quantity

- Use official legal sources
- Include case laws and precedents
- Add explanations and procedures

3. Regular Updates

- Add new judgments monthly
- Update amendments quarterly
- Review and clean old data

4. Monitor Usage

- Check response accuracy
- Collect user feedback
- Iterate on document quality

You're Ready!

Your legal chatbot is now ready to use. Start by:

1. Testing with the sample queries above
2. Adding more legal documents to `legal_data/`
3. Sharing with friends for feedback
4. Iterating based on user needs

Remember: This is a tool to provide general legal information. Always recommend consulting a lawyer for specific legal matters.

License

This project is for educational and informational purposes. Always comply with:

- Groq API Terms of Service
 - HuggingFace Model Licenses
 - Legal practice regulations in your jurisdiction
-

Happy Coding! 🚖🤖