



# Indian Law Information Chatbot

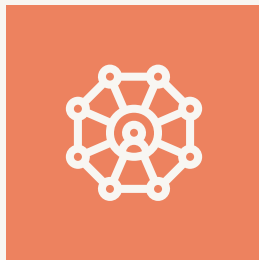
---

Based on Llama Index & ChatGPT

Piyush Dubey | Team - Innovate

# What are we working on

---



## Creating Custom Knowledge based bots

Dataset is Given to the model to train  
GPT on the top of it

# About the project

---

1. The Law Information ChatBot is designed to assist users in retrieving relevant information about legal sections and information about it. It utilises the Indian Penal Code (IPC) and Code of Criminal Procedure (CRPC) as dataset on which chatgpt is trained. It combines the power of Llama Index, a dataset indexing library, and ChatGpt, a language model, to create an interactive conversational experience.
2. The project repository contains the necessary code and resources to run the ChatBot. Additionally, the dataset used for indexing is also provided.

# 1.

**Indian Law Information Chatbot.ipynb**

File Edit View Insert Runtime Tools Help *Cannot save changes*

+ Code + Text Copy to Drive

RAM Disk

## Chat Build-a-thon

### Introduction

Open AI's ChatGpt Based Chat Bot on Custom Knowldege Base using FaceBook's Llama Index

This Project is under evaluation of Chat Build-a-thon on the premises of Lovely Professional University.

Team Name - **Innovate Members - 01 - Piyush Dubey(12017082)**

Github Link : [https://github.com/piyushdubey490/LPU\\_chat\\_build-a-thon](https://github.com/piyushdubey490/LPU_chat_build-a-thon)

Download the data for your custom knowledge base

Executing (24m 52s) <cell line: 1> > ask\_ai() > raw\_input() > \_input\_request() > select()

# 2.

Safari File Edit View History Bookmarks Develop Window Help

colab.research.google.com/github/piyushdubey490/... Creating a Chatb... Channel content...

Indian Law Information Chatbot.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

Files

- Files
- LPU\_chat\_build-a-thon
  - Context data
  - Home Cooking
    - Legal
      - IPC.pdf
      - crpc.pdf
  - Indian Law Information Chatbot...
  - README.md
  - img.png
  - sample\_data
  - index.json

### Download the data for your custom knowledge base

For the demonstration purposes we are going to use — as our knowledge base. You can download them to your local folder from the github repository by running the code below. Alternatively, you can put your own custom data into the local folder.

```
[1] 1 ! git clone https://github.com/piyushdubey490/LPU_chat_build-a-thon.git
```

Cloning into 'LPU\_chat\_build-a-thon'...  
remote: Enumerating objects: 30, done.  
remote: Counting objects: 100% (30/30), done.  
remote: Compressing objects: 100% (24/24), done.  
remote: Total 30 (delta 8), reused 19 (delta 4), pack-reused 0  
Unpacking objects: 100% (30/30), 3.49 MiB | 12.58 MiB/s, done.

### Install the dependencies

Run the code below to install the dependencies we need for our functions

```
[2] 1 !pip install llama-index==0.5.6  
2 !pip install langchain==0.0.148  
3 !pip install PyPDF2
```

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>  
Collecting llama-index==0.5.6  
Downloading llama\_index-0.5.6.tar.gz (165 kB)  
----- 165.0/165.0 kB 4.2 MB/s eta 0:00:00  
Preparing metadata (setup.py) ... done  
Collecting dataclasses\_json (from llama-index==0.5.6)  
Downloading dataclasses\_json-0.5.8-py3-none-any.whl (26 kB)  
Executing (24m 56s) <cell line: 1> > ask\_ai() > raw\_input() > \_input\_request() > select()

Disk 83.59 GB available

Chat Build-a-thon

# 3.

The screenshot shows a JupyterLab environment with the following components:

- File Browser (Left):** Displays a file tree with folders like 'Context data', 'Legal', and 'sample\_data'. The 'sample\_data' folder is expanded, showing 'index.json'.
- Code Editor (Center):** Contains Python code for defining functions to index data and query it. The code is as follows:

```
1 from llama_index import SimpleDirectoryReader, GPTListIndex, readers, GPTSimpleVectorIndex, LLMPredictor, PromptHel
2 from langchain import OpenAI
3 import sys
4 import os
5 import PyPDF2
6 from IPython.display import Markdown, display
7
8 def construct_index(directory_path):
9     # set maximum input size
10     max_input_size = 4096
11     # set number of output tokens
12     num_outputs = 2000
13     # set maximum chunk overlap
14     max_chunk_overlap = 20
15     # set chunk size limit
16     chunk_size_limit = 600
17
18     # define prompt helper
19     prompt_helper = PromptHelper(max_input_size, num_outputs, max_chunk_overlap, chunk_size_limit=chunk_size_limit)
20
21     # define LLM
22     # change the temperature according to your model's requirement, temperature helps in creativeness of the answer
23     llm_predictor = LLMPredictor(llm=OpenAI(temperature=0.5, model_name="text-davinci-003", max_tokens=num_outputs)
24
25     documents = SimpleDirectoryReader(directory_path).load_data()
```
- Terminal (Bottom):** Shows the execution status: 'Executing (24m 45s) <cell line: 1> > ask\_ai() > raw\_input() > \_input\_request() > select()'. The status bar indicates '83.59 GB available'.

# 4.

**Indian Law Information Chatbot.ipynb**

File Edit View Insert Runtime Tools Help [Cannot save changes](#)

RAM 100% Disk 100%

**Files**

- ..
- LPU\_chat\_build-a-thon
  - Context data
    - Home Cooking
    - Legal
      - IPC.pdf
      - crpc.pdf
  - Indian Law Information Chatbot...
  - README.md
  - img.png
  - sample\_data
  - index.json

**Set OpenAI API Key**

You need an OPENAI API key to be able to run this code. As API limit reaches fast so you need to use your API but for evaluation matter i am giving my API which i will later retract .

Then run the code below and paste your API key into the text input.

My API :- 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 If any error , use your API as mine must have been over used

```
[4] 1 os.environ["OPENAI_API_KEY"] = input("Paste your OpenAI key here and hit enter:")
```

Paste your OpenAI key here and hit enter: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

**Construct an index**

We are now prepared to build the index. Every file in the folder "data" will be extracted, divided into smaller pieces, and then embedded using OpenAI's embeddings API.

```
[5] 1 construct_index("/content/LPU_chat_build-a-thon/Context data/Legal")
```

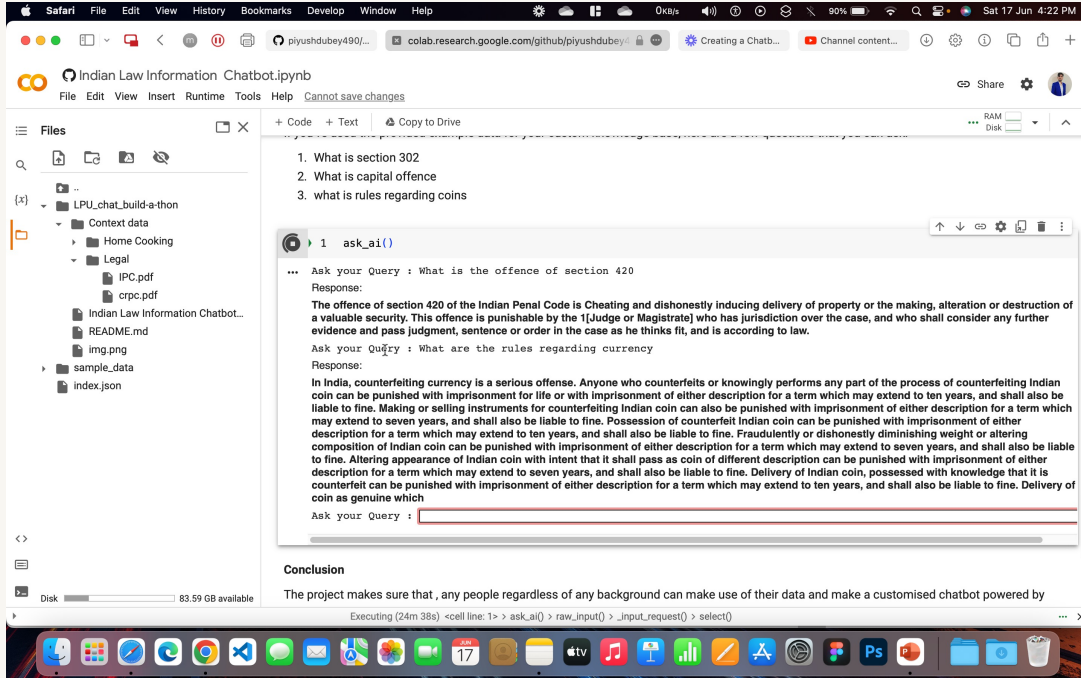
```
<llama_index.indices.vector_store.vector_indices.GPTSimpleVectorIndex at 0x7efca4545d20>
```

**Ask questions**

It's time to have fun and test our AI. Run the function that queries GPT and type your question into the input.

Executing (25m 2s) <cell line: 1> > ask\_ai() > raw\_input() > \_input\_request() > select()

# 5.





# SOCIETY BENEFITS

- It provides easy access to legal information, empowering individuals to understand their rights and responsibilities.
- The chatbot simplifies legal concepts, making them more understandable for non-legal professionals.
- It offers guidance on legal procedures, helping individuals navigate processes such as filing complaints or applying for licenses.
- The chatbot raises awareness about legal rights and obligations, promoting a more informed and empowered society.
- It assists legal professionals by providing quick access to legal resources and saving time in research.
- The chatbot promotes legal literacy, enabling individuals to make informed decisions and actively participate in legal discussions.
- However, it is important to remember that the chatbot is not a substitute for professional legal advice, and consulting a qualified lawyer is recommended for specific legal matters.

**THANKS**