# Proof of Concept

## Project Name: Bharatlaw Text-to-speech

## Introduction:

In the realm of law, the abundance of legal documents, including court judgements, poses a significant challenge for legal professionals and researchers. The sheer volume of textual information makes it difficult and time-consuming to extract key insights from these complex legal texts efficiently. Professionals in the legal domain face a critical challenge in efficiently processing extensive legal documents to extract essential details, impeding overall productivity.

**Problem Statement:** Efficient Consumption of Key Points through Concise Audio Summaries

## Objectives:

The primary objective is to develop a proof-of-concept (PoC) solution that specifically caters to the legal domain, enabling professionals to efficiently process lengthy legal documents, extract crucial and concise details swiftly and convert them into audio summaries. The system should offer a streamlined approach, allowing legal practitioners to grasp key points of the legal summary, implications, and nuances without being involved in manual document scrutiny which is time consuming. Implementation of TTS – Text-to-Speech for the given legal document summary into using python while researching on different TTS – (Text-to-Speech) technologies that are available to be used in the system. The primary objective of the TTS conversion should be contextual and concise enough to understand easily, instead of just word to word conversion from the summary.

## Resources:

Python
Google Text to Speech Library
HuggingFace
Tensorflow
Natural Language Processing

# Proposed Solution:

## I. Requirements:

1. **Document Types:** The document types will preferably be in the form of pdf, doc, txt files and raw text.
2. **User Input:** User will have the option to upload a file and copy-paste text.
3. **Summary Length:** The length of the audio summary will depend on the size of the legal summaries. The lengthier the textual summaries, the more will be the key points that need to be extracted and the lengthier will be the audio summaries. Even though efforts will be made to keep the audio summaries the generated audio summary is less than 5 minutes on average.
4. **Target Audience:** The target audience are dvided into various sectors such as:

*A. Legal Professionals:*

- Lawyers, attorneys, and legal practitioners who need to review and understand legal documents efficiently.
- Legal researchers looking to quickly grasp the key points and implications of legal texts.

*B. Corporate Entities:*

- In-house legal teams within corporations that deal with contracts, agreements, and legal documents.
- Executives and decision-makers who require a high-level understanding of legal documents without delving into the details.

*C. Students and Researchers:*

- Law students and researchers who may find audio summaries beneficial for rapid comprehension of legal concepts and cases.

*D. Business Owners:*

- Small business owners who handle legal agreements and contracts but may not have legal expertise. They could use audio summaries for a quick overview.

*E. Government Agencies:*

- Government officials and agencies dealing with legal documents and policies, seeking a time-efficient way to understand important details.

*F. Legal Tech Professionals:*

- Professionals working in legal tech who can integrate the solution into legal tech platforms or applications for broader use.

*G. Accessibility Users:*

- Individuals with visual impairments or other accessibility needs who can benefit from audio summaries as an alternative to reading lengthy legal texts.

*H. General Audience:*

- Anyone interested in staying informed about legal matters without having to read lengthy documents, such as journalists, policymakers, or general knowledge seekers.

# II. Text Extraction:

## 1. Document Preprocessing:

- ***Text Extraction:*** In the step, extraction of text content from different document formats like pdf and docx is initiated as input for further processing.
- ***Tokenization:*** Tokenization involves breaking down the text into individual words or phrases, known as tokens. This step is essential for subsequent NLP tasks, enabling the analysis of the document at a more granular level.
- ***Sentence Segmentation:*** Breaking the text into sentences is important for tasks like summarization, where understanding the structure of the text at the sentence level is crucial.

## 2. Data Cleaning:

- ***Removing Formatting and Special Characters:*** Stripping away unnecessary formatting, such as font styles, sizes, and special characters, ensures a clean and uniform text representation.
- ***Handling Line Breaks and Whitespace:*** Addressing line breaks and excessive whitespace helps in maintaining consistent spacing and structure within the text, making it easier for subsequent processing steps.
- ***Lowercasing:*** Converting all text to lowercase ensures consistency and avoids duplication of words due to case variations.
- ***Spell Checking:*** Implementing a spell-checking mechanism helps correct any typographical errors that might affect the accuracy of subsequent NLP tasks.

- ***Handling Abbreviations and Acronyms:*** Expanding or standardizing abbreviations and acronyms ensures that they are interpreted correctly during the summarization process.

# III. Natural Language Processing:

### 1. *Tokenization:*

Tokenization is a crucial step in natural language processing (NLP) tasks, including summarization. Different models and approaches may require different tokenization methods. Below, I'll provide a general overview of tokenization for BART, Distil-BART, TF-IDF, and frequency-based summarization:

1. BART and Distil-BART Tokenization: BART and Distil-BART typically use subword tokenization, breaking words into subword units. SentencePiece or Byte Pair Encoding (BPE) is commonly employed. Special tokens like <s> (start of sequence), </s> (end of sequence), and <pad> (padding) are used. Tokenization libraries such as the Hugging Face transformers library provide easy-to-use tokenizers for BART and Distil-BART.
2. TF-IDF Tokenization: For TF-IDF-based summarization, tokenization involves breaking the document into individual terms (words or n-grams). Common English stop words are often removed to focus on meaningful terms. Tokenization can be performed using libraries like scikit-learn's **TfidfVectorizer**.
3. Frequency-Based Summarization Tokenization: Frequency-based summarization may involve tokenization based on word frequencies. Tokenizing the document into words or phrases and counting their frequencies is a common approach.

### *2. Text Summarization:*

Text summarization involves condensing a large piece of text into a shorter version while retaining its key information and meaning. There are two main approaches:

- *Extractive Summarization:* It is a technique of selecting and extracting key sentences or phrases from the original text based on their importance. I have used TF-IDF and Frequency Based approaches.
- *Abstractive Summarization:* It is a technique of generating a summary that may contain rephrased sentences and new wording. I have Facebook's BART and Distil-BART for Abstractive Summarization.

## IV. Audio Conversion:

***Text-to-Speech (TTS):*** Convert the summarized text either through Extractive or Abstractive Summarization into audio summaries using a TTS. I have used Google's Text To Speech for converting the textual summaries to audio summaries. The audio summaries are not a direct word to word conversion of the given textual summaries.

## V. Result:

The summaries generated using BART and Distil-BART are more relevant with the text thereby providing better results.
In Extractive Summarization techniques, TF-IDF technique provided a better summary rather than Frequency based only.

## VI. Submission Links:

1. BART Summarizer – https://github.com/souravbiswas19/BART-Abstractive-Text-Summarization
2. Distil-BART – https://github.com/souravbiswas19/Distill-BART-Abstractive-Text-Summarization
3. TF-IDF – https://github.com/souravbiswas19/TF-IDF-Extractive-Summarization
4. Frequency Based – https://github.com/souravbiswas19/Frequency-Based-Extractive-Text-Summarization

The demonstration videos are uploaded in the github repositories