

# HACKATHON

## ~API PROBLEM STATEMENT~

13th July, 2023

Location: Department of Computer Science and Engineering, BUET

### Scenario:

Currently, we are living in the world of AI and LLM (Large Language Model). We can hardly do any stuff right now without an AI model! But so far, AI chatbots still lack many useful features! We are pretty sure they will be available within some time. But in the meantime, let's take matters into our own hands! We will try to build something like chatGPT but add more functionalities to it like voice search, image search, storybook generation, etc!

### Pre-Phase:

This is not a phase but an overall guideline on best practices. You should maintain the best practices (e.g., **rate limiting, versioning, caching**), if possible, while exposing your API.

### Phase 1:

#### Building the Chatbot

Existing language models, such as GPT-3, utilize transformer networks and are trained on extensive text data. However, one drawback of the currently popular model is it can only take text input as a prompt.

Let's start with the basics. As we are building a chatbot that allows voice prompts too, your system should be able to communicate over voice command, i.e., it has to recognize voice commands and also be able to respond via speech.

Now integrate an external API for a language model. Using a language model and external API to modify the language model, **create a chatbot that can accept inputs from users in both voice and text formats. The bot should generate responses in both text and voice, depending on the user's preference.**

### Phase 2:

## Task-1: File upload

We have extended the functionality of a language model! Now, let's extend the functionality of the chatbot by allowing users to upload files containing prompts. The input files can be either images or text files. Incorporate Optical Character Recognition (OCR) to extract the text from the image files.

## Task-2: Generate a kid's book

In Stage 2, you will have to create an API that serves as the backend functionality for the chatbot. This API will be responsible for performing specific tasks based on the prompts provided by the users. One of the tasks to be implemented is generating PDF files for kids' books.

Participants will design and develop the API in a way that, when given a prompt related to a kids' book, it will utilize the available resources, like language models and image-generating API, to generate a visually appealing and well-formatted PDF file. The API will handle the process of converting the prompt into a PDF document, incorporating elements like text, images, and formatting to create an engaging kids' book.

You will have the freedom to choose your preferred tools and techniques to accomplish this task, such as leveraging libraries for PDF generation and integrating any necessary image processing or OCR functionalities. The ultimate goal is to provide an API endpoint that takes the prompt or file as input and returns a downloadable PDF file, ready to be shared or printed as a kids' book.

## Bonus Task: Extend the API endpoint

By now you have a pipeline to do a specific task such as only generating a kid's pdf book. Now the participants are expected to make the backend more modular. Example: say we have a similar pipeline that takes the text prompt related to traveling places (some places the user has visited with travel images) the pipeline can generate a travel blog for that trip. Your API should have only one endpoint to incorporate such modularity.

***N.B: Participants are not expected to build such a pipeline. They only need to have ways so that such a pipeline can be incorporated into the system using as minimal change as possible to the existing system.***

## Phase 3:

### Create a pdf sharing platform.

All this time you have consumed external APIs. Now it's time to write one yourself. In this stage, the main objective is to create a platform where users can share and upload PDF books that have been generated by the chatbot. You will design and develop a user-friendly interface that allows users to easily upload their generated PDF books to the platform.

The platform should provide essential features such as browsing, searching, viewing, and downloading PDF books. Users should be able to browse through the available books, search for specific titles or keywords of the book, and view detailed information about each book. They should also have the option to download the books directly from the platform.

***Extra marks will be given to the teams who will provide their own storage for the pdf. (Not relying on the CDN)***

## **Final Phase:**

### **Wrapping Up:**

Feel free to add features that might better improve your system. You'll receive additional points if you come up with creative features and can use relevant API or Cloud Services to solve them! When you are done with everything, deploy your system on a public server for the final task.

## **Bonus Phase:**

### ***Building an E-book:***

After you have successfully done everything, if you have more time to kill then you can tackle this problem! But be sure to wrap up your main tasks beforehand. Build an e-book with the pdf that you have made in phase 2 and voice with the one with phase 1. Build a feature on your platform where your system will read out the story books to the viewer. You can think of other features on your own.

## **General Instructions:**

These are some general instructions for participants of any segment:

- Participants will need to start a new GitHub repository onsite and submit it at the end of the competition.
- The project must be developed on the day of the Hackathon. Any project which was developed (even partially) in advance will not be accepted.
- If participants choose to use publicly available code, then they must properly attribute the source.
- Unless otherwise specified, participants can choose their own tech stack.

***Any unfair attempts, misbehaviour or breach of terms will result in disqualification. The organizers reserve the right to make final decisions. Participants will be notified if there is notable modification in existing rules.***