Solution Challenge Hackathon

Domain: Quality Education

Contributors:

- Utsav Verma
- Nishant E. Ekka
- Shivam Kumar
- Sudip Mandal

Problem Statement:

Lacked the resources necessary to test the Knowledge one has acquired from various resources in the field of education.

Our Solution:

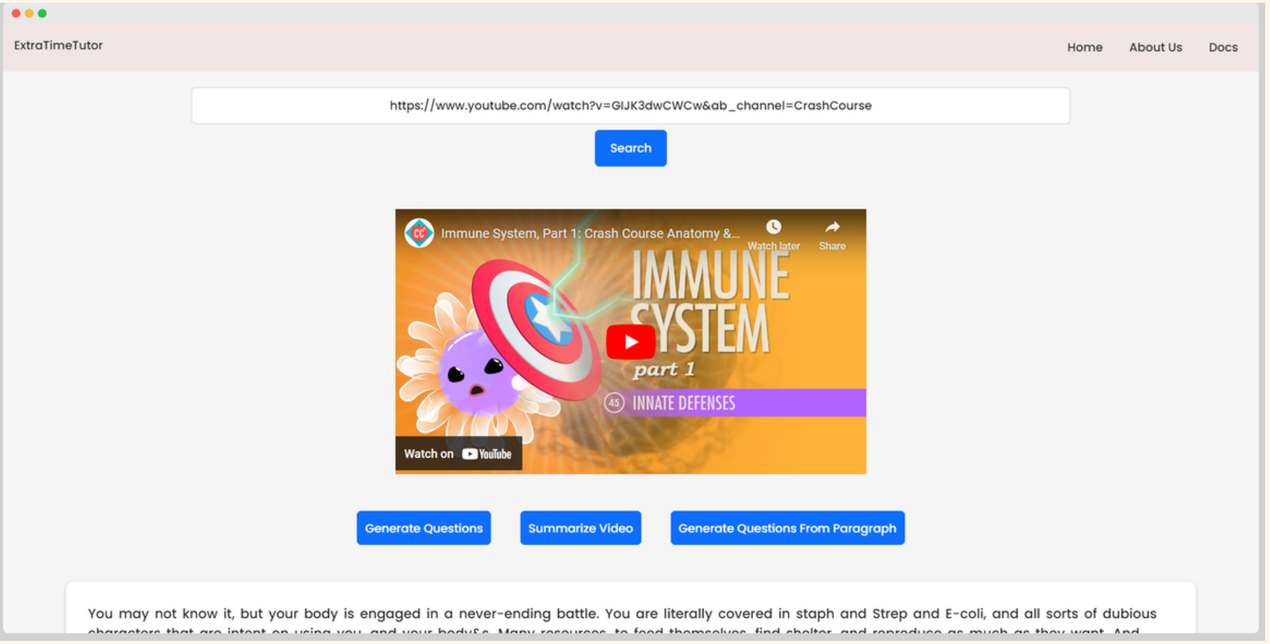
To provide an adequate amount of challenging problem statements for a certain individual to practice his/her knowledge related to some specific video Lecture or text content.

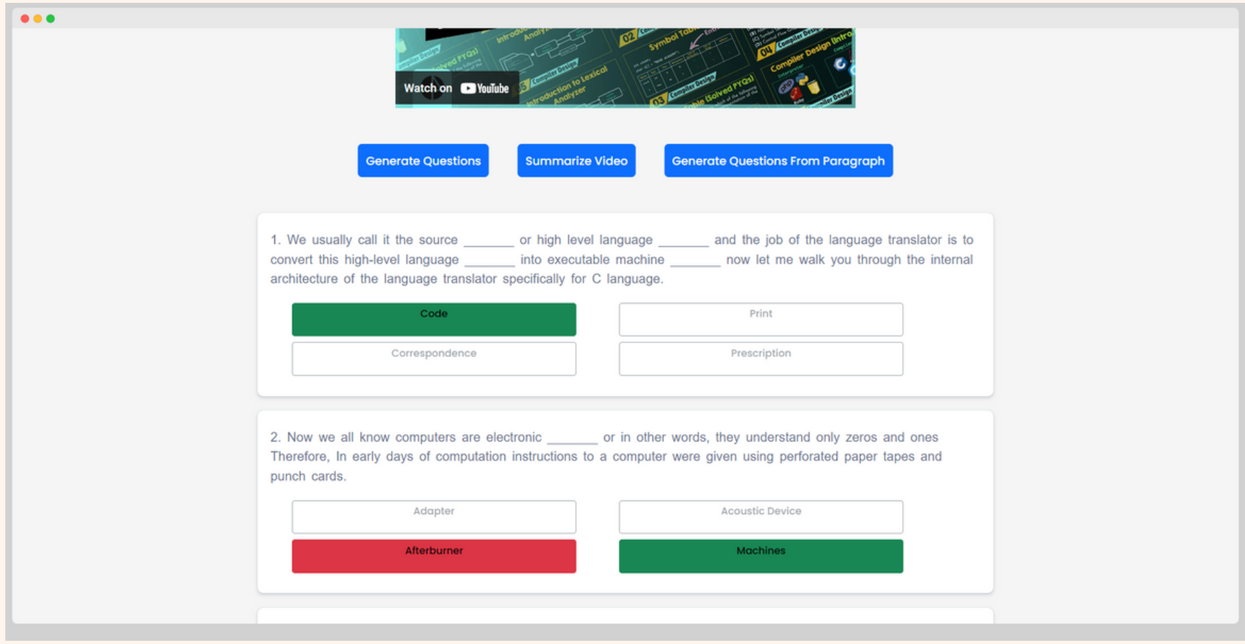
Extra Time Tutor

About Our Project:

Our algorithm begins by requesting a user-provided YouTube link, after which it uses that link to assess the video's subtitles.
Our NLP algorithm now makes use of the subtitles to produce a context that is punctuated, summarised, and ends with creating dynamic Multiple choice questions.
Those Multiple choice questions or the summarized content of the video is accessible to user in just one button click.

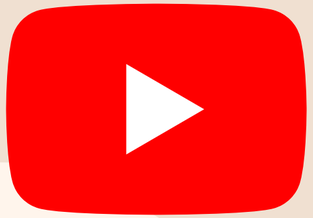
User Interference



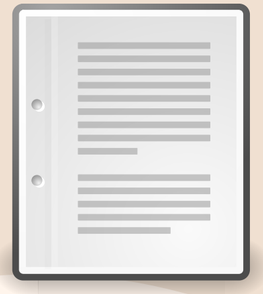


How it works?

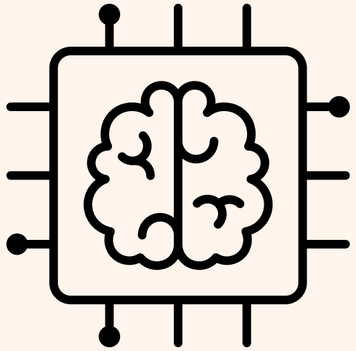
Getting subtitles



Summarized Subtitles



punctutation



NLP model Processing Content

Generated MCQs

Summarized Content

NLP model

To make our project what it is now, this model plays an important role, It is divided into following subparts:

- **Youtube Subtitles Extraction and Converting into a processable paragraph.**
- **Summarizing and just keeping the important points to reduce over-computation using model gpt2-finetuned-cnn-summarization-v2.**
- **Adding punctuation to our data.**
- **Important keyword extraction using Python Keyword Extractor(PKE) library.**
- **By using those keywords we get corresponding sentences from which we can generate questions.**
- **After generating questions, by using Wordnet of NLTK library we can find other distractors which will act as wrong options.**

Back-end and API Creation

In this project, we used the well-known Python Flask framework for backend development and creating API endpoints.

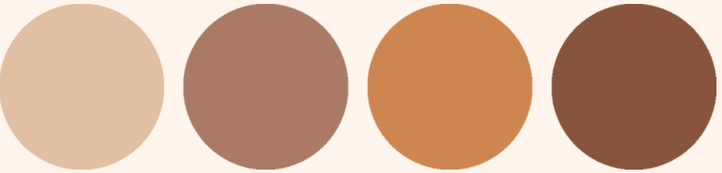
We divided our API into three endpoints with distinct functionalities in the backend to facilitate simple flow.

Our API endpoints are divided in the following manner:

- **It takes youtube video link and generate questions.**
- **It takes youtube video link and summarize it in text form.**
- **It takes text content and generates questions.**

Front-end and UI/UX

For frontend development, we have used standard tools like HTML, CSS, and Javascript along with React Framework.
A YouTube video window that functions just as regular YouTube is also provided. It displays the content of the Link we have given and the question we will create based on it.
In essence, our front end has a field for the YouTube Link and three buttons that may be used to create questions or summaries.



User-Friendly API

It is an application programming interface that is designed to be easily understood and used by developers. It provides a clear and consistent interface for accessing data or services, and its documentation is easy to understand and navigate. It has well-structured responses that make it easy for developers to interact. It should be easy to set up and use, and offer clear error messages that help developers diagnose and fix issues quickly. It can help reduce development time, increase developer satisfaction, and improve the overall user experience of the application or service.

Future Goals and Fields to Grow

Although we primarily designed this tool for MCQs, it has several potential growth areas that can be noted below:

- **We can also generate different formats of questions like True/False, Fill in the Blanks, one-word answer type questions, etc.**
- **We can also use the Wishper module to even analyze the audio part of the video lecture.**
- **Right now this model is only for English but it can also be programmed for other languages.**
- **This model can also be enhanced to summarize live google meets.**
- **For reaching a wider range of audience we are also in process of developing mobile application too.**

Thank you!