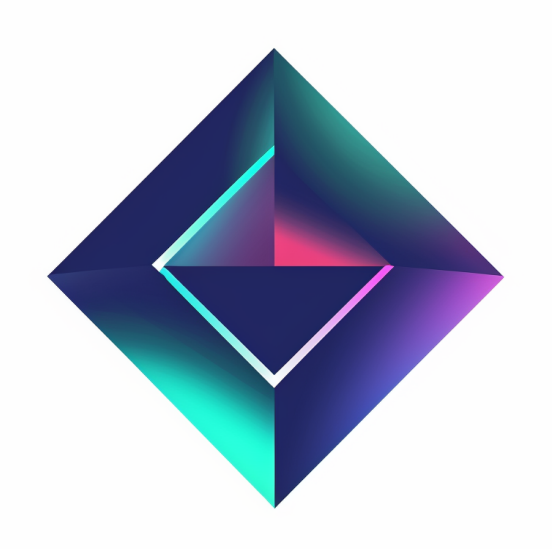

Econimate

Final Report

Jason Yan

Kelly Hong

Victoria Ye

Yukta Butala

Zhu Yuchen

Project Description

The rise of remote learning amid the COVID-19 pandemic has sparked a demand for alternative tutoring methods. In light of continuous advancements in artificial intelligence (AI), AI-based tutoring systems emerge as a viable solution to challenges students face, such as constraints related to the availability, cost, and location of traditional human tutoring (Sadler 2023). Today, users can easily delve into new subjects by posing questions to AI chatbots like ChatGPT and receive instant answers tailored to their preferences. Simultaneously, in the ever-evolving landscape of the contemporary job market, continuous skill development is imperative. Choosing to explore the field of economics provides individuals with the ability to comprehend intricate markets and cultivates robust analytical and problem-solving skills, complemented by essential business insights crucial for success in the professional realm. Notably, the significance of economics extends beyond the business sphere, offering advantages to individuals across diverse industries and in their everyday lives (Chladek 2017).

Our Experiential Learning team embarked on the creation of an educational powerhouse in economics named "Econimate." The objective is to harness the capabilities of Large Language Models (LLMs) and revolutionize economics education. Acknowledging the often daunting nature of economic concepts, Econimate employs an advanced AI video generator powered by GPT-4.0 Turbo. This innovative application not only addresses conceptual questions related to economics but also transforms lengthy and tedious text into dynamic, personalized video tutorials enriched with multimedia elements, providing users with an engaging and accessible learning experience.

Functioning as an AI tutor, Econimate offers personalized learning experiences, 24/7 accessibility, and adaptive assessment and feedback. It caters to the needs of individuals across

the spectrum, including students, professionals, and anyone curious about economics, facilitating their ability to stay competitive by acquiring and refining their economic knowledge. Moreover, Econimate contributes to the promotion of equal and accessible education resources. For instance, it proves particularly beneficial for students with disabilities like ADHD by delivering interactive and engaging content with the integration of multimedia elements (Roldan et al., 2021).

The AI tutor's flexibility accommodates users' busy schedules, proving valuable for those studying economics in a self-paced learning environment. Currently specializing in Economics tutoring, our team believes that focusing on one subject allows for a more profound exploration and understanding of the concepts. In navigating the future of work and recognizing the growing role of technology in education, Econimate emerges as a crucial tool, empowering individuals to master the complexities of economics.

Project Goals

Through the Google Hackathon, our team successfully refined the outputs from LLM (Google PaLM2 API) by employing prompt engineering techniques to enhance both text and code outputs. We successfully transformed textual information into audio and code into user-friendly visuals, culminating in the generation of videos to offer a comprehensive learning experience. In the second half of Mini 2, we replaced the PaLM2 API with the GPT-4.0 Turbo to enhance overall performance and content quality. Furthermore, we improved the model through additional training with a focus on applying advanced prompt engineering techniques identified during the hackathon. Additionally, we developed a user-friendly interface for direct interaction with the LLM model, eliminating the need for users to independently execute code for video

output and streamlining the learning process. Lastly, we explored the production of videos with multiple displaying methods.

Methodologies

Our project is segmented into two main parts: AI Training and Product Implementation. For the first segment, we trained our LLM to provide us with accurate responses to be implemented in the product's backend. The second segment involves creating a front-end website and refining our code on the backend. The project overview can be seen in *Figure 1*. Note that the PaLM2 API here is just an example and can be replaced with any other LLM API.

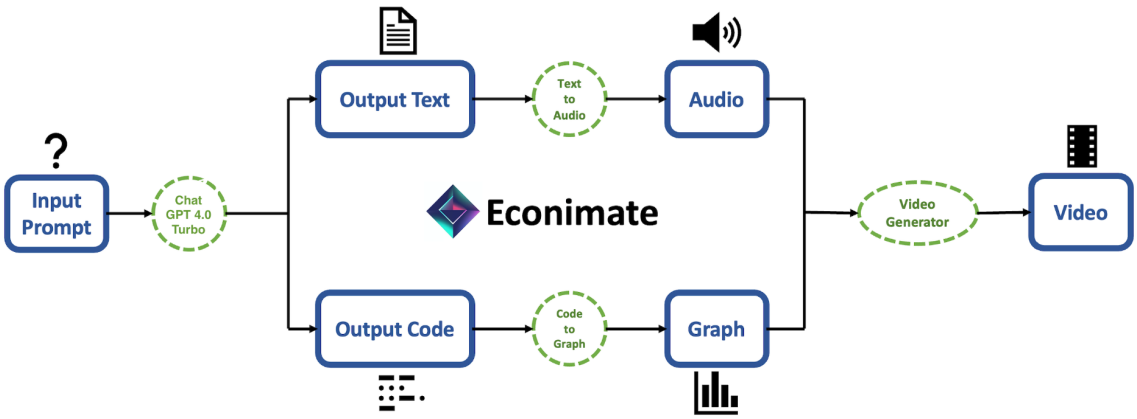

Figure 1: Project Implementation Overview

AI Training: Prompt Engineering

Our prompt engineering strategy involves creating prompts that consider structure, accuracy, and details. We have also employed certain techniques to ensure that the results align with our desired outcomes. For one, we asked the AI to assume the identity of an Economics tutor, which can provide tone and explanation of what we required (Gewirtz 2023). Furthermore, we asked the AI to engage in a "chain of thought" analysis using the "step-by-step" framework

(Gao 2022), which is proven to be able to guide the AI into producing results even by simply just including the sentence "Let's think step-by-step." (Kojima et al, 2022). Beyond that, we also explored the difference between few-shot prompting and zero-shot prompting. We have tested the efficacy of various prompts, which we will explore in the next section.

AI Training: Testing with Different APIs

During the implementation phase of our Google Hackathon, we faced multiple challenges in the usage of the PaLM2 API, such as the limited amount of tokens available and the inability of the API to produce desirable results. Hence, we have expanded our testing with resources from OpenAI, namely GPT3.5 Turbo and GPT4.0 Turbo. We chose OpenAI as it is the forerunner of Large Language Models in the market, with a much lower error rate but potentially lower processing speed as compared to other models (Wu et al, 2023; MindsDB, 2023). As our product focuses on accuracy, testing OpenAI's APIs is critical for us to optimize our product.

Product Implementation: Website and Integration

We created a website for Econimate. The front end is designed and operated on a local server using Flask, integrated with the Econimate API. The process can be seen in *Figure 2*.

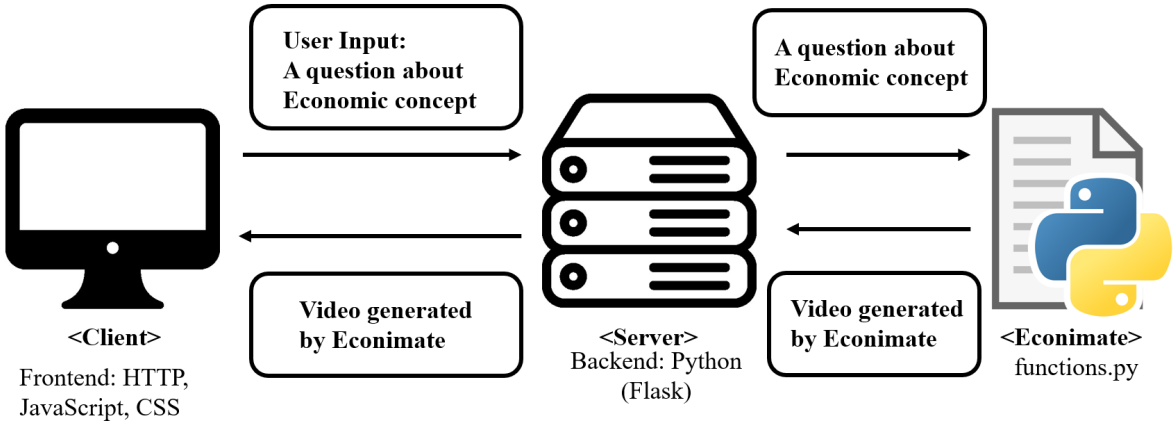

Figure 2: Development of Web Application

Front End Development: HTML, CSS, JavaScript

We implemented the frontend of our website using HTML, JavaScript, and CSS. The Econimate website uses a well-structured HTML document for the layout and uses CSS stylesheet to enhance the visual appeal. To offer users a sense of familiarity, the design incorporates colors, layout, and elements similar to ChatGPT's.

The JavaScript code enhances the dynamic behavior of the webpage. The “fetch_answer” function triggers the display of user questions in the chat interface, fetching answers from the backend. For responsive interactions, we implemented event listeners to ensure that, upon clicking the “ASK” button, the question is promptly displayed on the page along with an acknowledgement from Econimate saying, “Generating Video. Please wait.”

Backend Development: Python with Flask

1. **Flask Framework**

Our backend relies on the Flask web framework for its robust capabilities in handling routes and facilitating communication between the frontend and backend components.

2. **Python Code Integration**

OpenAI and gTTS Integration: The code integrates OpenAI’s GPT-4 Turbo 1106 model for generating comprehensive responses. Then, Google Text-to-Speech (gTTS) is utilized to convert text-based explanations into audio form, enhancing the multimedia learning experience.

Figure and Explanation Processing: The ‘query_genapi’ function processes user input, retrieves responses, and intelligently segments them into figure codes and corresponding explanations. The system ensures consistency by validating the number of figures and explanations and providing detailed and accurate content.

Audio and Figure Generation: Functions like ‘generate_audio’ and ‘process_figures’ are responsible for creating audio explanations and generating figures on the received responses. The system ensures the accurate representation of economic concepts through both visual and auditory means.

Video Clip Creation: The ‘create_video_clips’ function combines the generated figures with corresponding audio explanations, creating informative and engaging video clips.

Background Image and Final Video Composition: The ‘combine_and_save_video’ function incorporates a background image, providing context to the video clips, and produces a final video ready for users.

Product Design & Analysis

The final prompt engineered for Econimate can be seen in Appendix A. Here are several considerations in the final selection of the prompt:

(i) Suitability of desired response: The prompt is selected to ensure that the content produced is accurate and sensible to a teenage audience who is learning Economics, by asking the LLM to play the role of a tutor. A further emphasis is placed on reporting content which is not related to Economics, using the *“step-by-step”* method to filter out irrelevant questions.

(ii) Structure of response: The structure of response is of critical importance to the project as it ensures the smooth generation of the images through the effective separation of sections [Figure] and [Explanation], which ensures that the later separation process is smooth

(iii) Speed of response generation: The speed of response generation also matters as a more complex prompt might result in slower response time or more complex code, which would not lead to a slower video generation.

We also tested our prompts and trained our LLMs using a variety of examples which cover content from A-level economics textbooks. Various concepts from Microeconomics and Macroeconomics are included to ensure a sufficient breadth of the Economic concepts. Various prompts were used and adjusted with different APIs to ensure the most efficient.

We further tested different prompts with different APIs to optimize the prompt-API combination. Unsurprisingly, the prompts' efficiency depends on the APIs used. For the PaLM2 API, we find that shorter prompts tend to produce better results as longer prompts lead to a greater rate of hallucination, and in some cases, generation of non-output. For GPT-3.5 Turbo and GPT-4.0 Turbo, we find that longer and more complex prompts tend to give better results as long as they are structured well. For both cases, an important finding is that *zero-shot prompting* provided us with a better preservation of structure in the responses compared to *few-shot prompting,* which ensures the accurate separation of images and explanations in the implementation phase.

To improve our prompting techniques for future development, we propose utilizing Consistency-based Self-adaptive Prompting (COSP) for *zero-shot prompting*, minimizing the entropy of responses to improve efficiency (Wan et al, 2023).

Beyond prompt engineering, we tested our trained model on different APIs using 20 different test cases. Our findings show that PaLM2 API suffers in accuracy across all prompts, especially in the generation of accurate code to produce images. However, their results greatly improved with more examples inputted. The GPT APIs have greater accuracy, and interestingly, do not show great improvement after training with examples. A study by Radford et al. (2019) confirmed that in a *zero-shot* setting, GPT APIs can act as efficient multitask learners and

produce efficient results despite unsupervised methods. Results on the test cases of different APIs can be found in Appendix B.

Finally, we tried tuning methods for each API, but we had limited success with the low token limit we have available. This can be a potential area for future improvement.

Features Developed:

1. Visual and Audio integration that supports multimedia learning

The standout feature of Econimate is the ability to transform textual responses into dynamic, personalized video tutorials. This multimedia approach caters to a wide range of learning preferences and helps reinforce the understanding of complex economic concepts. Visual elements such as graphs and charts will help illustrate and break down complex data or theories, making them more digestible. The audio component, often in the form of narrations or explanations, complements these visuals by providing context and additional information. This combination helps in creating a more immersive and engaging learning environment, which can lead to better retention of information and a deeper understanding of the subject matter. The use of multimedia is particularly effective in catering to diverse learning needs, including those of auditory and visual learners, and enhances the overall educational value of Econimate.

2. Video file generated available locally

A significant feature of Econimate is its ability to generate video files that are available for local download and access. This functionality is essential for users who prefer or need to review educational content offline. The local availability of video tutorials ensures that learning is not interrupted by internet connectivity issues and allows users to have a personal library of educational content. This feature is particularly beneficial for users who may have limited internet access or those who want to refer back to these tutorials without needing to be online. It

also supports a self-paced learning style, where users can revisit complex topics at their convenience.

3. 24/7 Accessibility and Adaptive Learning:

Compared to the time limitations of traditional tutoring, Econimate provides 24/7 accessibility. This has offered the flexibility of personalized learning experiences, available around the clock. Econimate provides a modern, flexible alternative to traditional economic tutoring, especially valuable in a fast-paced world where learning needs constantly evolve.

Areas of Improvement:

1. Deploying the website on a Cloud Server

Currently, our website operates in a local environment. Transitioning to a cloud server deployment would significantly enhance accessibility and speed for users. This strategic move allows for a broader user base, accommodating multiple simultaneous users. The deployment on a cloud server aligns with our objective of serving diverse user needs, providing an optimized and responsive experience.

2. Improving Audio and Visual Interaction

Econimate's tutoring videos feature a graph accompanied by audio generated using gTTS, reading explanations formulated by the GPT model. However, the current audio might sound somewhat machine-like, lacking the natural tone of a human tutor. To address this, we propose training a model to emulate the characteristics of a real person's voice, creating a more authentic and user-friendly auditory experience. Additionally, incorporating a moving avatar for the AI tutor will add a visual element, making the interaction feel more like a genuine tutor engaging with the users. These enhancements aim to significantly elevate the overall learning

experience for our users. A demo of the project can be seen here. The GitHub repository of our project can be seen here.

Limitations, Mitigations & Future Scope

Speed and Scalability

One notable limitation of Econimate is the extended time required for video output generation, ranging from 4 to 7 minutes. This delay primarily stems from the local execution of the model and is amplified by variations in users' computing power and available RAM. To address this, we are actively planning to migrate the application to a cloud server infrastructure. By leveraging superior computing capabilities in the cloud, we aim to significantly reduce video generation times, ensuring a more efficient and consistent user experience for all users.

User Experience and Mobile Application

Currently, Econimate is accessible solely through a desktop interface. Recognizing the importance of enhancing user accessibility, we envision the development of a dedicated mobile application. This expansion would empower users to engage with Econimate seamlessly from any location, at any time, thus enriching the overall user experience.

Scalability to Other Subjects

While Econimate is presently specialized in handling economics queries, we acknowledge the potential to broaden its scope to encompass other quantitative subjects. This expansion aligns with our vision to create a versatile learning tool capable of addressing a diverse range of academic and professional queries. By diversifying the subjects covered, Econimate aims to become an invaluable resource for users seeking video explanations across various domains.

Multilingual Capability

Econimate currently operates exclusively in English, reflecting the language proficiency of ChatGPT 4.0 Turbo. However, recognizing the global user base with diverse linguistic preferences, we aim to implement multilingual support. This involves integrating an AI model capable of understanding and responding to queries in different languages. Consequently, users will have the flexibility to choose their preferred language for both input queries and video output, thereby enhancing the inclusivity and usability of Econimate.

Conclusion

Econimate is an educational software based on GPT-4.0 Turbo, using LLM to produce responses to Economic queries before translating them to a video. Currently, Econimate runs locally and can produce educational videos for a vast variety of economics concepts. To build Econimate, the team trained the AI using prompt engineering techniques and examples, before integrating the LLM's response using the *gTTS* and *moviepy* library to create a final video product. For further extension of our project, we aim to deploy our product into a cloud server and improve the efficiency of video generation such that it is easily accessible to the public.

Work Cited

Chladek, Natalie. “7 Reasons Why You Should Study Economics: HBS Online.” *Business Insights Blog*, 30 Nov. 2017, online.hbs.edu/blog/post/5-reasons-why-you-should-study-economics.

Gao, Andrew. “Prompt Engineering for Large Language Models.” *SSRN*, 17 July 2023, papers.ssrn.com/sol3/papers.cfm?abstract_id=4504303.

Gewirtz, David. “How to Write Better CHATGPT Prompts for the Best Generative AI Results.” *ZDNET*, www.zdnet.com/article/how-to-write-better-chatgpt-prompts/. Accessed 1 Dec. 2023.

Kojima, Takeshi, et al. “Large Language Models Are Zero-Shot Reasoners.” *arXiv.Org*, 29 Jan. 2023, arxiv.org/abs/2205.11916.

Molina Roldán, Silvia, et al. “How Inclusive Interactive Learning Environments Benefit Students without Special Needs.” *Frontiers*, Frontiers, 7 Apr. 2021, www.frontiersin.org/articles/10.3389/fpsyg.2021.661427/full.

“Navigating the LLM Landscape: A Comparative Analysis of Leading Large Language Models.” *AI Development Cloud Platform*, mindsdb.com/blog/navigating-the-llm-landscape-a-comparative-analysis-of-leading-large-language-models. Accessed 3 Dec. 2023.

Radford, Alec. *Language Models Are Unsupervised Multitask Learners - Openai*, cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Accessed 4 Dec. 2023.

Sadler, Chris. “The Future of AI Tutoring in Higher Ed.” *New America*, 4 Apr. 2023, www.newamerica.org/oti/briefs/the-future-of-ai-tutoring-in-higher-ed/#:~:text=The%20increase%20in%20remote%20learning,face%20with%20obtaining%20human%20tutoring.

Wan, Xingchen. *Better Zero-Shot Reasoning with Self-Adaptive Prompting - ACL Anthology*, aclanthology.org/2023.findings-acl.216.pdf. Accessed 3 Dec. 2023.

Wu, Sean. *A Comparative Study of Open-Source Large Language Models*, arxiv.org/pdf/2308.04709.pdf. Accessed 9 Dec. 2023.

Appendix A - Final Prompt

You are an economic tutor. Let's think step-by-step. Firstly, check if there are any economic concepts and questions within the prompt. If there are no economic concepts or questions, respond with 'This is not Economics related. Ask me anything about Economics instead!'.

If an economic concept or question is mentioned, explain the concept using figures and explanations for each part of the concept. The figures should be labeled [Figure - concept name] and the explanations should be labeled [Explanation - concept name].

All figures in the response should be displayed as either Python code for figures (graphs/tables/matrix/equations/more). Each explanation should refer to each and all of the figures and real-life scenarios.

You can have multiple [Figure] and [Explanation]. Have the response displayed in the order of [figure - a], [explanation - a], [figure - b], [explanation - b], etc.

Appendix B - Test Results for Different APIs

API	Training?	Test Cases Passed	Major Issue
PALM2	No	4/20 (20%)	● Inability to distinguish Economics concepts from other concepts
	Yes	8/20 (40%)	● Explanation generated inaccurate ● Massive errors in the code that result in inability to generate images
GPT-3.5 Turbo	No	14/20 (70%)	● Some errors in the explanation of Economic concepts
	Yes	16/20 (80%)	● Some structural errors that result in difficulty in the separation of *[Explanation]* and *[Figure]*
GPT-4.0 Turbo	No	19/20 (95%)	● A minor error in the accuracy of the Economic concept, and in linking the figure to the Explanation
	Yes	20/20 (100%)	● Nil