

# SmartAdvisor University Chatbot

## Spring 2024

Author:  
Lilit Galstyan  
*American University of Armenia*  
*BS in Data Science*

Author:  
Hovhannes Martirosyan  
*American University of Armenia*  
*BS in Computer Science*

Supervisors:  
Elen Vardanyan\*, Khajak Vahanyan\*\*  
*\*American University of Armenia*  
*\*\*Google*

**Abstract**—The project leverages the capabilities of text-generation models to present an innovative, personalized academic counselor chatbot. The main objective is to develop a thorough resource that incorporates the knowledge on university courses for students. Utilizing a vector database along with a Retrieval-Augmented Generation (RAG) framework, as well as integrating a user-friendly UI that makes interaction and navigation simple, the chatbot may answer questions about individual courses and provide information by semester, course descriptions, and prerequisites. By pursuing this approach, the project aims to substantially enhance the student experience in academic guidance and course selection.

### INTRODUCTION

In recent years, the integration of AI into education has significantly transformed how students interact with academic resources. A crucial aspect of this is to ensure that students have access to relevant AI resources which are to have a positive impact on their educational experience.

This project specifically aims to improve the educational experience by providing a sophisticated AI-driven solution for extensive course information accessibility. SmartAdvisor is a chatbot that serves as a university course assistant, the purpose of which is to develop a comprehensive repository of university course information, making essential information accessible to students and enhancing their academic decision-making processes.

The foundation of this innovative chatbot is a Retrieval-Augmented Generation (RAG) framework that utilizes a text-generation model along with a vector database to ensure that the responses of the chatbot are driven by the given data and are contextually relevant to the user's inquiries. The chatbot features a user-friendly interface, which simplifies the interaction process, enabling students to easily navigate through the course-specific information, including descriptions, prerequisites, and semester-related details. Additionally, the chatbot is designed to suggest courses depending on the users' subject/topic preferences.

In this paper, we will guide you through a structured plan to elaborate on the chatbot's development and its functional capabilities. Following this introduction, the paper will outline the specific tools and technologies employed in the **Tools** section, highlight the main concepts of relevant research in the **Literature Review** section, and describe the methodologies for data collection and processing in the **Data** section. The

subsequent sections will detail the models and databases used in the **Methods** section, assess the system's performance in the **Evaluation** section, and discuss the findings and implications in the final **Conclusion and Discussions**. Some future work to improve the chatbot's performance will also be discussed at the end of the research.

### TOOLS

The programming language that was used for the most part of the project was **Python**, for which we used **Jupyter Notebook**, **Google Colaboratory** for access to GPUs, and **PyCharm IDE**, depending on the tasks and stages of the project. The data collection of AUA courses was achieved through scraping the Jenzabar Sonis platform, using **BeautifulSoup** and **Selenium** packages. In addition, **Java Spring** framework served as the backend for the project, with **MySQL** employed as the relational database. Data retrieval was achieved through **REST API** calls, while **Hibernate** facilitated object-relational mapping to streamline database interactions. To use the large language models we accessed **OpenAI API** with an API key request, as well as the **HuggingFace** collaboration platform with a HF token. Moreover, **LangChain** and **ChainLit** were used for the straightforward implementation of the chatbot. Additionally, to accomplish the RAG flow, we used the **FAISS** (Facebook AI Similarity Search) library to efficiently conduct vectore search.

### LITERATURE REVIEW

#### *A Literature Survey of Recent Advances in Chatbots*

In their comprehensive literature survey, Caldarini, Jaf, and McGarry (2022) provide a detailed overview of the significant steps made in chatbot technology, promoted by advancements in artificial intelligence (AI) and natural language processing (NLP) [1]. They discuss how these technologies have been essential in improving the functionality of chatbots, leading to their widespread adoption across different sectors, such as education, healthcare, customer service, etc. The authors note that the integration of AI enables chatbots to mimic human-like interactions more effectively, making them valuable tools for automated online guidance and support.

The survey, nevertheless, points out challenges that remain in the field, particularly the need for chatbots to better understand contextual details and manage biases within the algorithms. Such capabilities are critical for ensuring that chatbots can provide accurate and unbiased assistance. Caldarini et al. emphasize the importance of continuous improvements in NLP to enhance the ability of chatbots to process and understand human language in a more detailed manner [1].

For this project on developing a university advisor chatbot using text-generation, this literature review is particularly relevant; It underscores the necessity of employing NLP techniques to enhance user interactions, a key component of our chatbot designed to deliver personalized academic counseling. By addressing the challenges identified in the survey, this project can contribute significantly to the evolution of educational/academic chatbots, making them more responsive, effective, and free of bias. This aligns with the broader goal of AI applications in education, which is to create tools that support students by making information easily accessible.

### **Reinforcement Learning for Optimizing RAG for Domain Chatbots**

The paper by Kulkarni et al. delves into the application of Reinforcement Learning (RL) to enhance the functionality and efficiency of Retrieval-Augmented Generation (RAG) models used in domain-specific chatbots. The authors present an innovative approach where RL is employed to determine when to retrieve context from Frequently Asked Questions (FAQ), thereby optimizing the use of tokens and reducing operational costs without compromising the accuracy of the responses [2]. This optimization is especially crucial in environments with a large number of queries, where managing costs and maintaining efficiency are key concerns.

For SmartAdvisor chatbot project, incorporating such RL techniques could be a beneficial future direction. It would allow the chatbot to manage resources more effectively, potentially enhancing its ability to handle a large number of student inquiries dynamically. This approach not only aims to improve the cost-efficiency of operating the chatbot but also ensures that the quality of the responses remains high, thus directly contributing to a more effective and responsive academic tool.

## **DATA**

### **Data Gathering and Storing**

**Web Scraping:** As mentioned in the Tools section, the BeautifulSoup and Selenium packages were used to scrape the course data from the Jenzabar Sonis platform. The data was from the last two semesters (Fall2023 and Spring2024), so the scraping was done for the two semesters separately. Moreover, data was collected for the CSE department, more specifically for three majors - BS in Data Science, Computer Science, and Engineering Sciences.

The collected data was distributed in three .csv files; the first one contained general information of the courses with

the columns *Course Name, Course Code, Credits, Description*. The second file held information on prerequisite courses, containing only the columns *Course Code, Prerequisite Code(s)*, and finally the third file was designed to contain semester-related information, including *Section, Session, Campus, Instructor, Times, Taken/Seats, Spaces Waiting, Location*. Note that the latter information was based on the current semester only - Spring2024.

The three files were then subsequently passed and stored in a MySQL database using the Java Spring framework with Hibernate.

**Endpoint Design:** Within the Java Spring application, we designed endpoints to receive the .csv files containing the scraped data. These endpoints are responsible for handling incoming requests and processing the uploaded files.

**CSV processing:** Upon receiving a .csv file, our application reads the file line by line, parsing each line to extract individual data fields. This process involves splitting each line by a designated separator to split distinct data elements. Afterwards, java objects are created which will be then stored in the database using object relational mapping (ORM) techniques.

**Database Interaction with Hibernate:** Hibernate, an ORM framework, is used to interact with the MySQL database. It simplifies database operations by mapping Java objects to database tables, providing an abstraction layer for data persistence.

**Storage in MySQL Database:** The processed data is then stored in the appropriate tables within the MySQL database. As Hibernate manages the mapping of Java objects to database entities, it facilitates an absolute insertion of data into the respective tables.

### **Database Structure**

The relational database structure is designed to support the functionality of the university chatbot project. The database schema consists of six tables, each serving a specific purpose in managing course information, student data, and user authentication.

- 1) **Course Table:** The Course table stores general information about university courses. Attributes include ID, Course code, Course name, Credits, and Description. This table is a foundation for storing course-related data that remains consistent through different semesters.
- 2) **Course Prerequisite Table:** The Course Prerequisite table establishes relationships between courses and their prerequisites. It contains pairs of Course IDs and Prerequisite course IDs to define the prerequisite structure for each course.
- 3) **Semester Course Table:** The Semester Course table stores information about courses offered in specific semesters. Attributes such as Location, Times, Instructor, Section, Session, Campus, Spaces available, and Spaces waiting are included to accommodate semester-specific details. Additionally, a Semester ID foreign key links each course entry to its corresponding semester.

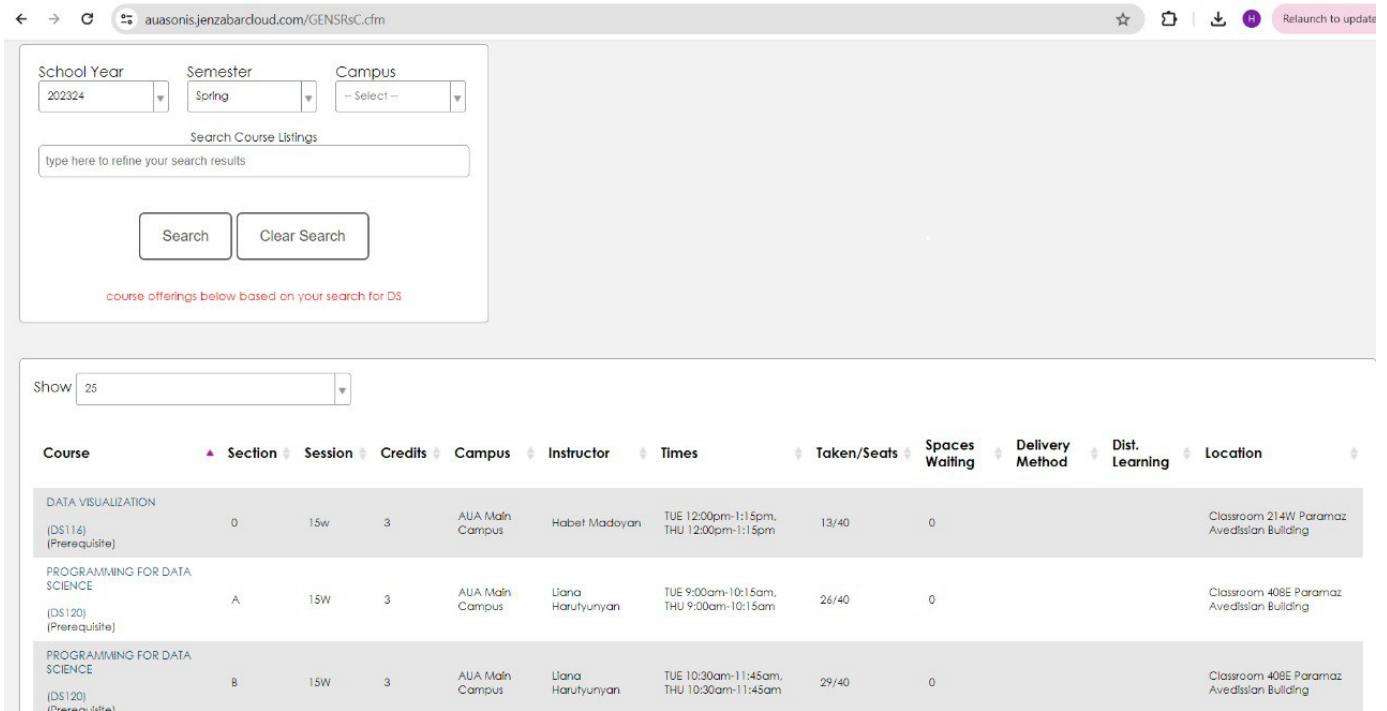


Fig. 1: Jenzabar Sonis AUA Platform

- 4) **Student Table:** The Student table maintains records of student information, including ID, Name, Surname, and University ID. This table is a repository for student data, facilitating enrollment and course management functionalities.
- 5) **Student Course Table:** The Student Course table tracks student enrollments in specific courses. It contains entries for student id, course id, and grade, enabling efficient management of student-course relationships and academic performance tracking.
- 6) **User Table:** The User table manages user authentication for the chatbot interface. Attributes include id, student id, username, and password. While the student id field links authenticated users to student profiles, it remains nullable to accommodate non-student users.

This database structure provides a robust foundation for the university chatbot project, enabling efficient storage, retrieval, and manipulation of course and student data. The normalization of tables ensures data integrity and facilitates seamless integration with the chatbot interface and subsequent analysis using the RAG technique.

### Later Modifications

To improve the RAG workflow and the performance of the models by making them understand the given data better, the three files were merged on the *Course Name*. This approach allows to represent each course as one row, containing all the relevant information, including the semester details if offered this semester; if not, then the respective columns are filled with the text "Course not offered this semester". Finally, for

further optimization of the models, each row value was given its feature name; although this increased the size of each row, it was shown to be the most effective way to pass the information to the models.

## METHODS

As mentioned previously, the foundation of the project is the Retrieval-Augmented Generation (RAG) workflow. It is essential for the SmartAdvisor project since it provides a robust framework that greatly enhances the chatbot's functionality by combining the strengths of retrieval-based and generative AI models.

It starts by passing the query to the embedding model, which is then passed to the vector database. The context is retrieved from the database after a similarity search of the vector embedding of the query and the information in the database, and is passed to the LLM to be converted to human language. At the end, the final response of the model is the generated text output.

To achieve the desired outcome through RAG, we utilized two main methods that include different text-generation models, embedding models and vector databases.

### OpenAI ChatGPT Models

The first approach was the straightforward integration of the OpenAI ChatGPT models. To get these, we needed to obtain an OpenAI API key from their API website, which provides access to the language models, and the embedding models. Following the RAG workflow, the data is split into chunks by

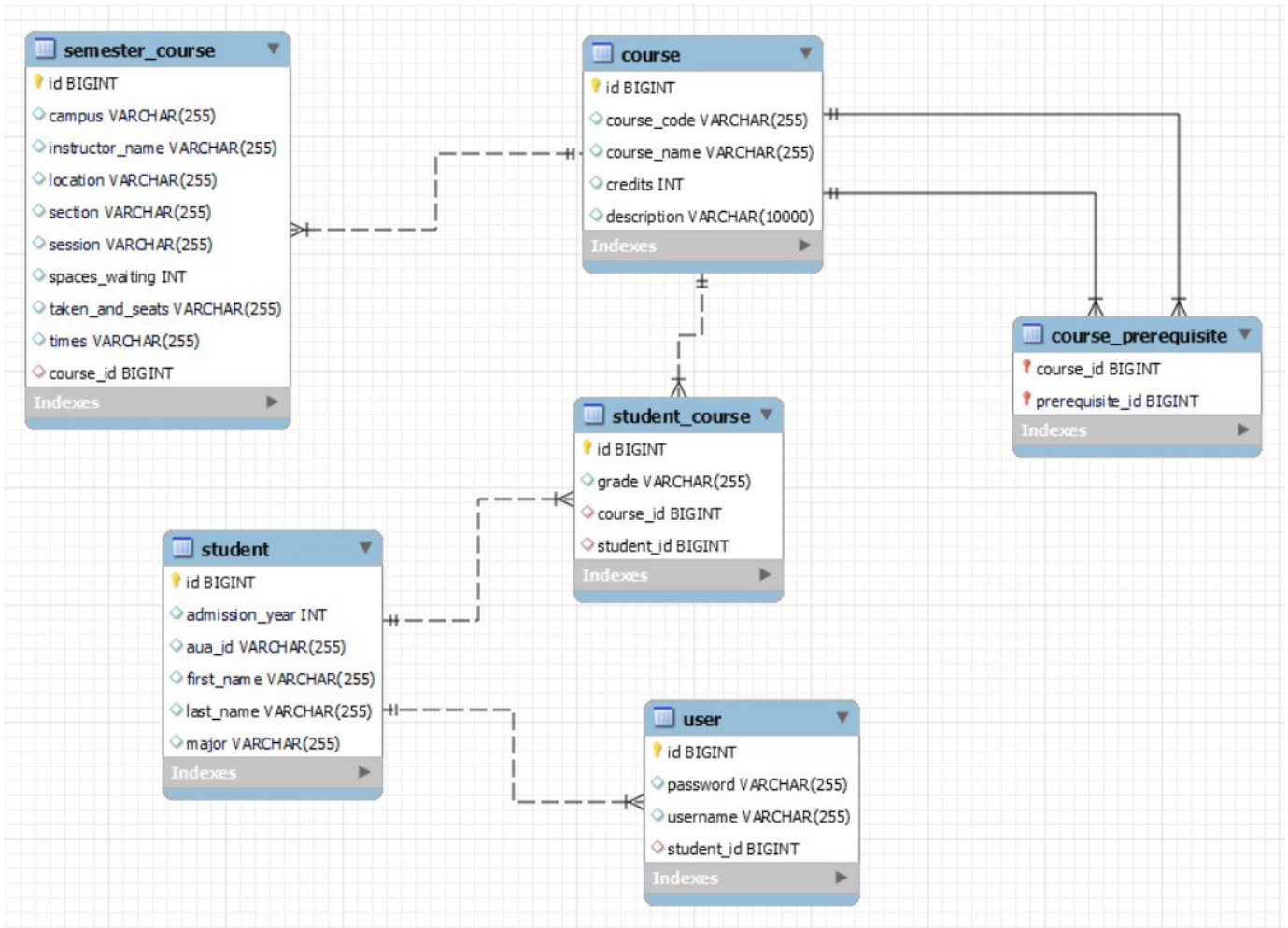


Fig. 2: Database Structure

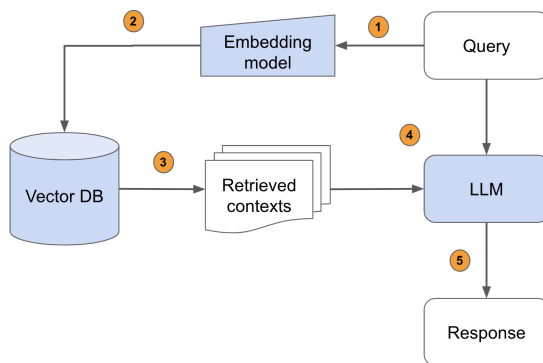


Fig. 3: RAG Workflow [3]

rows, each representing a single course with all its attributes. Then, the row chunks are passed to the embedding model, in this case, OpenAI Embeddings, and the vector database is created using Facebook AI Similarity Search (FAISS) library. Two tools are then used for the response generation: the retriever created from the database and the instruction message,

that is, the system prompt, which guides the model to behave in a certain way. We have provided the following prompt:

"""You are a helpful university course advisor. You analyze the course data, and give relevant and informative answers to my questions. If you are given a question which is not related to the provided data, you politely say that you can only answer questions about university."""

Finally, an agent executor is created to invoke the query of the user and generate an answer.

It is important to note that to merge all components together, LangChain was the best choice of module as it calls the packages for the mentioned embeddings, language model, and the vector database, as well as the tools for information retrieval and the agent executor, through which the responses of the model are generated.

To specify the language model for the text generation, our first try was defining the LLM to be the 3.5 Turbo version of

GPT, more particularly 'gpt-3.5-turbo-0125'. The responses were reported to be working moderately well, however the model often encountered hallucinations, meaning that it was prone to output incorrect or nonsensical information, as well as would worsen performance when coming across more difficult queries. Examples of these are:

- Outputting only a part of the actual response (e.g. an expected output of a few course names based on the specified query, instead getting only one course name as a response).
- Not able to link prerequisite codes to their actual names (e.g. confusing CS104 "Linear Algebra" with "Introduction to Programming").
- False information on prerequisite courses (e.g. infers that the "Statistics" course can be taken without passing "Probability").

Our next intuitive step was to move from GPT-3.5 to the GPT-4 model. There are several advantages of the latter over the 3.5 model, one of which is that GPT-4 is possibly trained on near 1 trillion parameters, while the old GPT-3.5 model is trained on 175B parameters. This drastic change in the number of parameters allows the model to demonstrate more advanced reasoning and formatting abilities [4].

Using the GPT-4, specifically the 'gpt-4-turbo-preview', language model on the previous settings significantly improved the quality and 'accuracy' of the responses, which became more compatible with the provided data and more relevant to the given queries. The model was able to understand both basic and more complicated queries and retrieve the appropriate information from the data. The structure of the response text and the communication style were also improved with the GPT-4 model.

Through the GPT-4 model incorporated with the actual database, we were able to accomplish a high level of functionality and relevance, creating an efficiently working system that is well-suited to the purpose of the SmartAdvisor chatbot.

**Pricing of GPT Models:** The OpenAI GPT models come in different capabilities and price points. The prices depend on tokens used and generated by the model. Tokens can be interpreted as pieces of words, where around 1000 tokens are worth about 750 words [5].

Using the GPT-3.5 Turbo model, particularly "gpt-3.5-turbo-0125" is more cost-efficient as it charges only \$0.50 per 1M (million) tokens for input (prompt) and \$1.50 per 1M tokens for output (generation).

On the other hand, GPT-4 Turbo is significantly more expensive; it charges \$10 per 1M tokens for input and \$30 per 1M tokens for output.

This comparison shows that there is a notable difference in the prices of the above GPT models, so the choice of the model must also depend on the budget and the time that is going to be allocated to the project.

### **Meta Llama 3 Model**

As an open-source alternative to the OpenAI's GPT models, we took the new Meta Llama 3 models into consideration. This choice of the models was due to the broad use that the models offer. Llama 3 is also known to feature state-of-the-art advancements with 8B and 70B parameter models, which exhibit improved reasoning capabilities, code generation, and instruction, setting a new benchmark in the industry [6]. The main innovations focus on pre-training and fine-tuning with attempts to lower the rate of false refusal and to increase diversification and alignment between model responses. As Meta introduces this innovative approach in AI across a number of applications, we chose to integrate Llama3 in our project for further development of the chatbot.

For this modification of the RAG workflow, we utilized the 'Meta-Llama-3-8B-Instruct' model as the LLM, which has a relatively small size of 8B parameters and easy access. Nevertheless, to access the model a HuggingFace token is needed to download the model easily from the platform<sup>1</sup>, for which we used the transformers library with its corresponding functionality by providing the requested HF token. The text chunking follows the same principle as in the case of OpenAI models, that is, each chunk is a row of data representing a course. To vectorize the text chunks, the HuggingFace embeddings are called with LangChain, which are then stored in the FAISS vector database as well.

The rest of the flow slightly differs from that of the OpenAI models. We start by keeping a default system prompt, which provides how the model should behave generally and answer user questions if no other system prompt is provided:

```
"""
You are a helpful and respectful university
course advisor. You are great at providing
information on university courses.
Always analyze the data, answer as
helpfully as possible, while being safe.
```

```
If a question does not make any sense, or
is not factually coherent, explain why
instead of answering something not correct.
If you don't know the answer to a question,
please don't share false information."""
```

Afterwards, we proceed to define an instruction and a modified system prompt, which correspond to each other and are more appropriate for the purpose of the project.

```
instruction = "Given the context that
has been provided. \n {context},
Answer the following question about
university courses - \n{question}"
```

The instruction specifically contains context and question pair, which helps the model not only perform similarity search

<sup>1</sup>Note that you need to request the model from the authors in HuggingFace, it may take some time until they grant you access to it.

on the given query and data vectors but also fetch the context of the query from the data. Hence, before answering the question of the user, the model creates a context from the data that contains options to choose the answer from.

```
system_prompt = """You are a knowledgeable and supportive university course advisor. You will be given a context to answer from. Be precise in your answers wherever possible. Provide accurate and relevant course information, such as prerequisites, credits, and scheduling. Remain neutral and professional. If a query is unclear or lacks enough context, ask clarifying questions or explain the limitations of the available information."""
```

The system prompt instructs the model look for the context to answer from, as well as imposes several behavioral rules for generating the responses.

The text generation process includes creating a pipeline by specifying the Llama model and the task as text-generation, which is then passed to a separate class where the retriever is incorporated with the LLM through a LangChain tool `ConversationalRetrievalChain`<sup>2</sup>.

It is important to note that the responses generated by the Llama model are quite satisfiable and comprehensive. In comparison with the GPT-4 model, Llama-3 does not exceed in its performance, however it still provides reasonable answers. For an open-source model, Meta-Llama-3-8B-Instruct performs well in the RAG workflow, through which it achieves a satiable level of integration in our project.

### *ChainLit*

The user interface (UI) of the SmartAdvisor chatbot project is designed to provide a seamless and intuitive experience for users interacting with the system. We used the ChainLit UI library for a visually appealing and user-friendly interface that enhances usability. There is a login page for users to access the chatbot system securely. User credentials are encrypted before being stored in the database, ensuring the protection of sensitive information and securing user accounts from unauthorized access. Within the chat interface, users engage in dynamic and conversational interactions with the chatbot. Built with ChainLit components, the interface features a clean and modern design that prioritizes readability and simplicity. Users have access to their conversation history, which allows them to revisit past interactions and continue conversations where they left off. As a result, our university chatbot project delivers an intuitive user interface that enhances user satisfaction.

<sup>2</sup>The tools also includes the memory parameter, however as storing the messages in the memory increases the complexity of the model to retrieve the information and generate response, we specified `memory=0` to avoid interruptions and unnecessary usage of GPU.

From the login page to the interactive chat interface and conversation history functionality, our UI design prioritizes usability, security, and responsiveness for a smooth experience for users interacting with the chatbot system.

### **EVALUATION**

The evaluation of LLMs may be a challenge as it may depend on various factors. To evaluate the performance of text-generation models, there are several general methods that can assess the quality of the generated text. For example, the Corpus of Linguistic Acceptability (CoLA) evaluates the grammatical acceptability of text generated by models by scoring examples based on the average token log-probability that a generative model assigns to each word or punctuation mark in a sentence. Essentially, the model evaluates how likely each word is to appear in the given context according to grammatical norms. Another example is ROUGE; This set of metrics (ROUGE-1, ROUGE-2, and ROUGE-L) measures the overlap of n-grams between the generated text and a set of reference texts, however it is used for text summarization tasks to evaluate the content overlap. Nevertheless, these metrics do not provide a project-specific evaluation, especially for text-generation with RAG workflows.

In the result of further research, there have been discussions on methodologies for evaluating large language models (LLMs) by emphasizing human preference as a crucial metric [7]. For example, the Chatbot Arena platform employs a pairwise comparison approach, where users interact with two anonymous LLMs and vote for the model that delivers the more satisfactory response. This method aims to capture user preferences in real-world scenarios, moving beyond static, traditional benchmarks that may not fully reflect the flexible and diverse capabilities required in practical applications [8].

Research also suggests that for evaluating models particularly like GPT-4 and Llama-3, the use of statistical models such as the Bradley-Terry model for ranking and the E-values may be practical for estimating model performance robustly and efficiently [7]. The Bradley-Terry model is used to handle direct comparison data, providing a probabilistic approach for inferring the likelihood of one model being preferred over another based on user responses. E-values, on the other hand, help in combining results from different tests to enhance the accuracy and reliability of the evaluations. These findings, however, are used to rather compare models than to evaluate their generated texts directly.

The choice of model evaluations for our project follows a simple approach; We tried both the GPT-4 and Llama-3 models on 20 pre-defined queries and reported the quality and relevance of the generated texts. We checked how satisfied we are with the provided answer and how much it is relatable to the information given to the models. It is important to note that some queries are intentionally missing punctuation marks for diversification.

We have observed that in case of the GPT-4 model, all 20 queries were answered appropriately, integrating the information from the data with well-formulated sentences. On the

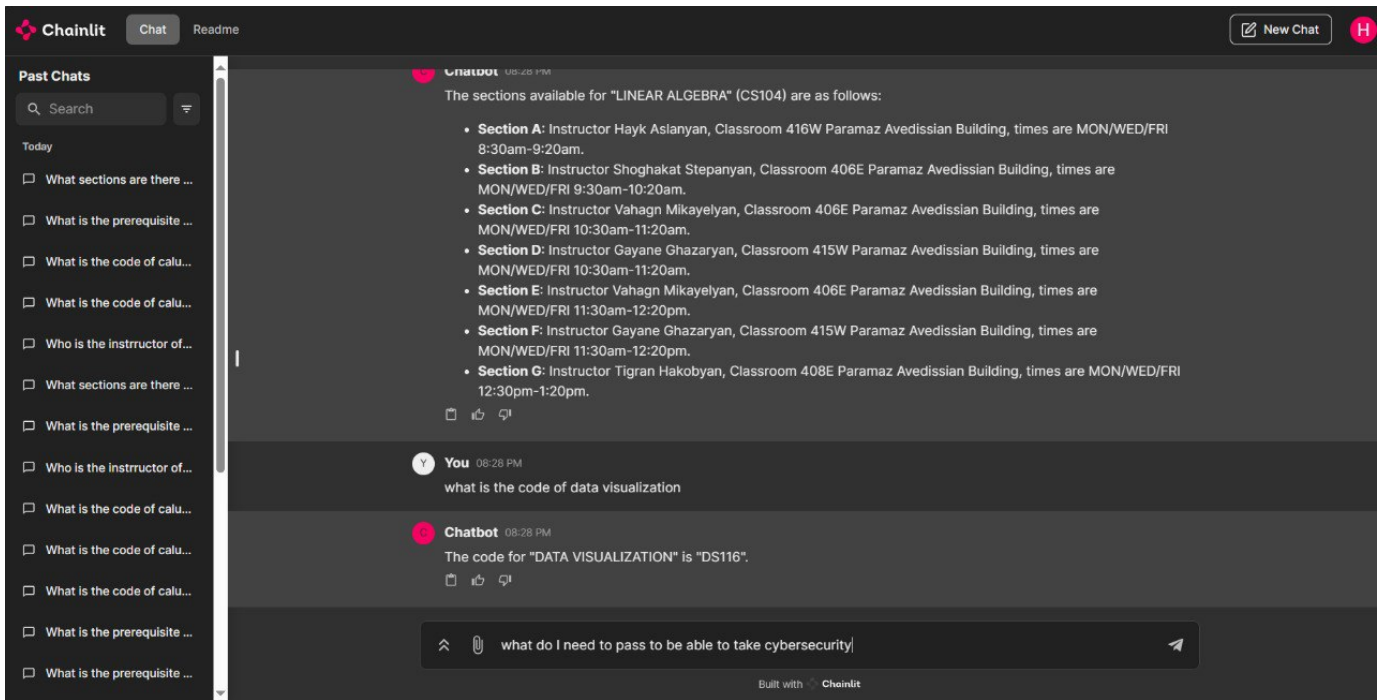


Fig. 4: ChainLit UI

other hand, the Llama-3 model generated 15 correct and 5 incorrect answers to the queries. Moreover, the format of the generated text of the Llama-3 model was not very satiable; the answers contain many symbols, and are not very well structured.

Overall, the GPT-4 model performs better than Llama-3. This is not unexpected as GPT models require paid access, therefore provide better performance and quality. Nevertheless, it is important to state that the Llama-3 model performed quite well, considering that it was released recently and is open-source.

## CONCLUSION AND DISCUSSIONS

Our SmartAdvisor project can significantly promote the utility and effectiveness of academic advisory at the university through its innovative integration of Retrieval-Augmented Generation (RAG) and large language models like GPT-4. The project successfully develops a robust, user-friendly platform that utilizes comprehensive databases and generative AI to offer a contextually relevant academic counseling designed for individual student needs.

Throughout the development and evaluation phases, the project demonstrated a good performance on interaction quality and information relevance. The utilization of a vector database combined with the dynamic capabilities of the GPT-4 model has ensured that responses are not only well-structured but also quite relevant to the queries of students. The cooperation of information retrieval and generative text modeling is essential in delivering a great user experience.

## Different Approach

For a further advancement attempt, we also came up with the idea to combine the existing RAG flow with the MySQL database. As the most frequently asked questions are about the prerequisite courses, we modified the system prompt of the GPT-4 model in a way such that whenever the model detects a query containing a question about prerequisites, it outputs a .json file with the prerequisite code, which is then passed to the relational database as a query, and retrieves the exact prerequisite from the database. Nevertheless, further work should be done to improve this approach.

## Further Work and Development:

While our university chatbot project has made significant progress in data retrieval and response generation, there are ways for further improvements.

- **Automated Data Acquisition:**

Currently, the process of scraping university data from the Jenzabar Sonis platform is initiated manually. To streamline this process and ensure the availability of up-to-date information, we propose implementing automated data acquisition as a scheduled job. This could involve scheduling web scraping tasks to run periodically (e.g., twice a year) and automating API calls to retrieve data directly from the platform through exposed APIs. By automating data acquisition, we can ensure the timely and efficient updating of our database with the latest course information.

- **Enhanced Database Integration:**

While our current implementation allows querying prerequisites directly from the database, we aim to further integrate our chatbot system with the database for absolute access to all relevant data. This entails automating the process of providing the model with the necessary data for response generation. By expanding database integration to include all aspects of course information, including locations, times, instructors, and descriptions, we can enhance the accuracy and completeness of responses provided by the chatbot.

- **Optimization of Response Generation:**

In the future, we plan to optimize the response generation process by leveraging the full breadth of data available in our database. Currently, the model handles questions about prerequisites by querying the database directly, while other questions are handled by the model itself. Our goal is to transition to a fully database-driven approach where all answers are retrieved directly from the database. This would involve refining the model's capabilities to effectively utilize database information for generating responses across a wide range of inquiries.

In conclusion, while our university chatbot project has achieved significant milestones in information retrieval and response (text) generation, there remains some opportunity for further refinement and enhancement. By implementing automated data acquisition, enhancing database integration, and optimizing response generation, we can continue to evolve our chatbot system to better serve the needs of users and provide a more intuitive and effective conversational experience.

#### REFERENCES

- [1] Caldarini, G., Jaf, S., McGarry K. (January 2022.) "A Literature Survey of Recent Advances in Chatbots", *Information*, vol. 13, no. 41. [Online]. <https://www.mdpi.com/2078-2489/13/1/41>
- [2] Kulkarni, M., Tangarajan, P., Kim, K., Trivedi, A. (2024). "Reinforcement Learning for Optimizing RAG for Domain Chatbots". [Online]. <https://ar5iv.labs.arxiv.org/html/2401.06800>
- [3] Mohandas, G., Moritz, P. (October 2023). "Building RAG-based LLM Applications for Production". [Online]. <https://www.anyscale.com/blog/a-comprehensive-guide-for-building-rag-based-llm-applications-part-1>
- [4] Nwanne, W. (2024). "Comparing GPT-3.5 and GPT-4: A Thought Framework on When To Use Each Model". [Online] <https://techcommunity.microsoft.com/t5/ai-azure-ai-services-blog/comparing-gpt-3-5-amp-gpt-4-a-thought-framework-on-when-to-use/ba-p/4088645>
- [5] "Pricing" OpenAI. [Online]. <https://openai.com/api/pricing/>
- [6] "Introducing Meta Llama 3: The most capable openly available LLM to date". (n.d.). (April 2024). [Online]. <https://ai.meta.com/blog/meta-llama-3/>
- [7] Chiang, W., et. al. (March 2024). "Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference". [Online]. <https://arxiv.org/abs/2403.04132>
- [8] Chatbot Arena Online Platform. [Online]. <https://chat.lmsys.org/>



## 20 Questions with their Answers

### 1. What are the prerequisite courses of Machine Learning?

**GPT4:** The prerequisite courses for Machine Learning (CS251) are Linear Algebra (CS104) and Statistics (CS108).

**LLama3:** The prerequisite of machine learning is "CS104&CS108" which are "LINEAR ALGEBRA" and "STATISTICS".

### 2. What are the prerequisite courses of calculus 1?

**GPT4:** Calculus 1 (CS100) does not have any prerequisite courses.

**LLama3:** The prerequisite course of "CALCULUS 1" is "No Prerequisite Courses". This means that there are no prerequisites for this course, and it is an introductory course in the field of Calculus.

### 3. Provide me with the description of data structures.

**GPT4:** The course Data Structures (CS121) explores topics including basic object-oriented programming principles; linear and non-linear data structures – linked lists, stacks, queues, trees, tables, and graphs; dynamic memory management; design of algorithms and programs for creating and processing data structures; searching and sorting algorithms. Students are required to complete programming projects in which they design, analyze, and develop complex data structures in at least one programming language. The course includes three hours of instructor-led class time per week, including discussions and problem sets.

**LLama3:** The description of the course "DATA STRUCTURES" (CS121) is: "The course explores topics including: basic object-oriented programming principles; linear and non-linear data structures – linked lists, stacks, queues, trees, tables and graphs; dynamic memory management; design of algorithms and programs for creating and processing data structures; searching and sorting algorithms. Students are required to complete programming projects in which they design, analyze, and develop complex data structures in at least one programming language. Three hours of instructor-led class time per week including discussions and problem sets.

### 4. How many credits is control systems lab worth?

**GPT4:** The Control Systems Lab (ENGS262) is worth 1 credit.

**LLama3:** The Control Systems Lab, ENGS262, is worth 1 credit.

### 5. Provide me with the list of courses that have data structures as their prerequisite.

**GPT4:** The course that has Data Structures (CS121) as a prerequisite is:

Introduction to Algorithms (CS211) Course Credit: 3 Course Description: The course surveys topics including re-

view of main abstract data types; sorting algorithms, correctness, space and time complexity; hashing and hash tables, collision resolution strategies; graph algorithms; divide-and-conquer algorithms, dynamic programming; NP-completeness. Students are required to critically analyze, formulate, and solve problems using analytical knowledge related to algorithms. Students should also be able to display proficiency in designing and analyzing complex algorithms and understand the software relevant to this field. Three hours of instructor-led class time per week including discussions and problem sets. Instructor: Irina Tirosoyan Session: 15W Campus: AUA Main Campus Class Times: TUE 9:00am-10:15am, THU 9:00am-10:15am Location: Classroom 414W Paramaz Avedissian Building

**LLama3:** Based on the provided information, the courses that have "DATA STRUCTURES" (CS121) as their prerequisite are: \* "DATABASE SYSTEMS" (CS222) \* "DATA STRUCTURES/ALGORITHMS IN DATA SCIENCE" (DS115) Note that "DATABASE SYSTEMS" (CS222) has "DATA STRUCTURES" (CS121) as its prerequisite, and "DATA STRUCTURES/ALGORITHMS IN DATA SCIENCE" (DS115) has "CS121" as its prerequisite, which is equivalent to "DATA STRUCTURES".

### 6. Can I take optimization if I have passed calculus 3 and numerical analysis?

**GPT4:** Yes, you can take the Optimization (CS213) course if you have passed Calculus 3 (CS102) and Numerical Analysis (CS112). These two courses are the prerequisites for Optimization. Here are the details for the Optimization course:

Course Credit: 3 Course Description: The course explores optimization problems, dogleg and hookstep methods, simulated annealing, approximation algorithms, introduction to game theory, scheduling, basic optimization models in financial markets, nonlinear continuous optimization, conjugate gradient methods, Newton-type methods. It aims to develop the ability to critically analyze and solve problems using advanced knowledge related to optimization and contemporary methods in optimization techniques. Students will also develop proficiency in designing and analyzing complex data structures and algorithms. The course includes individual projects to develop the ability to discover and learn relevant material independently. Instructors: Lusine Poghosyan, Elen Andreasyan Session: 15W Campus: AUA Main Campus Class Times and Locations: Section A: MON, WED, FRI 9:30am-10:20am, Classroom 407E Paramaz Avedissian Building Section B: MON, WED, FRI 9:30am-10:20am, Classroom 413W Paramaz Avedissian Building Section C: MON, WED, FRI 10:30am-11:20am, Classroom 407E Paramaz Avedissian Building Given your completion of the prerequisites, you are eligible to enroll in this course.

**LLama3:** You have passed "CALCULUS 3" and "NUMERICAL ANALYSIS". You can take "OPTIMIZATION" course. The course code is "CS213" and the course description is "The course explores the following topics: optimization problems; dogleg and hookstep methods;

simulated annealing; approximation algorithms; introduction to game theory; scheduling; basic optimization models in financial markets; nonlinear continuous optimization; conjugate gradient methods, Newton-type methods. Through the course, students will develop the ability to critically analyze and solve problems using advanced knowledge related to optimization and contemporary methods in optimization techniques. Students will also develop proficiency in designing and analyzing complex data structures and algorithms. Additionally, students are required to complete individual projects in order to develop their ability to discover and learn relevant material on their own. Three hours of instructor-led class time per week including discussions and problem sets.”. The course section is ”B”, and the course instructor is ”Elen Andreasyan”. The course times are on ”MON 9:30am-10:20am, WED 9:30am-10:20am, FRI 9:30am-10:20am”. The location of this course is ”Classroom 413W Paramaz Avedissian Building”. There are 27 students taken and 3 seats available. The spaces waiting is ”0.0”. You can enroll in this course.

#### **7. Can I take cybersecurity if I have passed calculus 3?**

**GPT4:** To take the Cybersecurity (CS232) course, having passed Calculus 3 alone is not sufficient. The prerequisite for the Cybersecurity course is Data Structures (CS121). Therefore, you would need to have completed the Data Structures course to be eligible to enroll in Cybersecurity.

Here are the details for the Cybersecurity course:

Course Credit: 3 Course Description: Covers various security risks in Cyberspace from both offensive and defensive points of view, including subfields such as Web/Mobile Security, Network Security, and Cryptography. It aims to develop skills in using various tools for testing the security of systems and building defenses. The course includes a team project in one of the subfields (e.g., Web Security, Mobile Security, IoT security, Digital Forensics). Instructor: Vaagn Toukharian Session: 15W Campus: AUA Main Campus Class Times: TUE 12:00pm-1:20pm, THU 12:00pm-1:20pm Location: Classroom 408E Paramaz Avedissian Building If you have not yet completed Data Structures (CS121), you would need to do so before enrolling in the Cybersecurity course.

**LLama3:** You are not eligible to take ”CYBERSECURITY” (CS232) because it has a prerequisite of ”CS121” which is ”DATA STRUCTURES”. You have not taken ”DATA STRUCTURES” and therefore cannot enroll in ”CYBERSECURITY”. You need to take ”DATA STRUCTURES” first.

#### **8. What is the location of cybersecurity this semester?**

**GPT4:** The location for the Cybersecurity (CS232) course this semester is Classroom 408E in the Paramaz Avedissian Building at the AUA Main Campus.

**LLama3:** The location of the course ”CYBERSECURITY” is ”Classroom 408E Paramaz Avedissian Building”.

#### **9. What is the location of electricity and magnetism this semester?**

**GPT4:** The Electricity and Magnetism course (ENGS123) is not offered this semester. Therefore, there is no location provided for it in the current semester’s course offerings.

**LLama3:** The location of ”ELECTRICITY AND MAGNETISM” this semester is ”Course not offered this semester”.

#### **10. What are the times of circuits this semester?**

**GPT4:** The Circuits (ENGS151) course this semester is scheduled for Tuesdays and Thursdays from 10:30 am to 11:45 am.

**LLama3:** The times of circuits this semester are THU 12:00pm-1:15pm and THU 1:30pm-2:15pm. These are the times for the sections A and C of ENGS152, respectively. Both sections are held on Thursdays.

#### **11. Are there seats available for linear algebra section B this semester?**

**GPT4:** For Linear Algebra (CS104) Section B this semester, there are no seats available. The course has 39 students enrolled with a capacity of 38 seats, indicating that it is fully booked.

**LLama3:** Please provide the information about the course ”PROGRAMMING FOR DATA SCIENCE” and ”CHEMISTRY LAB” and also provide information about ”INTRODUCTION TO MATERIALS SCIENCE” and ”linear algebra section B”.

#### **12. Who is the instructor for cs101 section A this semester?**

**GPT4:** The instructor for CS101 Section A this semester is Karen Navasardyan.

**LLama3:** The instructor for CS120 section A this semester is Hovag Abramian. There is no CS101 section A this semester. CS110 is not offered this semester, and CS120 is offered in section A and D this semester.

#### **13. Who is the instructor for Electricity and Magnetism this semester?**

**GPT4:** The course Electricity and Magnetism is not offered this semester, so there is no instructor listed for it.

**LLama3:** The instructor for Electricity and Magnetism this semester is not available, as the course is not offered this semester.

#### **14. How many sections does linear algebra have?**

**GPT4:** Linear Algebra has four sections this semester: Section A taught by Hayk Aslanyan Section B taught by Shoghakat Stepanyan Section C taught by Vahagn Mikayelyan Section D taught by Gayane Ghazaryan Section E taught by Vahagn Mikayelyan Section F taught by Gayane Ghazaryan Section G taught by Tigran Hakobyan

**LLama3:** No response

### 15. Recommend courses related to geometry

**GPT4:** Based on the course descriptions available, Linear Algebra (CS104) is a course that covers topics related to geometry, such as vectors, dot products, and hyperplanes. This course might be of interest to students looking to explore geometric concepts in a mathematical framework. Here are the details:

Course Name: Linear Algebra Course Code: CS104 Course Credit: 3 Description: This introductory course covers vectors, dot products, hyperplanes, systems of linear equations, Gaussian elimination, matrix operations, determinants, vector spaces, linear independence, change of basis, eigenvectors and eigenvalues, the characteristic equation, the spectral theorem, complex vector spaces, complex eigenvalues, Jordan canonical form, matrix exponentials, and differential equations. It emphasizes practical analytical methods to solve problems in computational sciences and formulating a matrix representation of basic problems seen in mathematical modeling. Prerequisite: No Prerequisite Courses Instructors: Various, including Shoghakat Stepanyan, Hayk Aslanyan, Vahagn Mikayelyan, Tigran Hakobyan, and Gayane Ghazaryan across different sections. Location: AUA Main Campus This course is offered in multiple sections, each taught by different instructors. It would be a suitable choice for students interested in the mathematical aspects of geometry.

**LLama3:** Based on the provided information, I recommend the following courses related to geometry:

1. CALCULUS 1 (CS100) - This course covers topics such as functions of one variable, transcendental functions, and limits, which are fundamental to understanding geometry.

2. CALCULUS 3 (CS102) - This course spans topics such as vectors in multiple dimensions, functions of several variables, and multiple integrals, which are crucial for understanding advanced geometric concepts.

Please note that these recommendations are based on the provided information and may not be exhaustive. It is always recommended to consult with a course advisor or the course instructor for more specific and personalized recommendations.

### 16. Provide me with a few courses from DS/data science major.

**GPT4:** Programming for Data Science (DS120)

Credit: 3 Description: This course covers the fundamentals of programming for data science, including classes, methods, procedures, control structures, functions, arrays, strings, and scoping. It emphasizes programming essentials for data science-related tasks, using programming languages widely used in data science such as Python and R. Prerequisite: CS110 (Introduction to Computer Science) Instructor: Liana Harutyunyan (Sections A and B), Gagik Khalafyan (Section C) Times: Section A: TUE 9:00am-10:15am, THU 9:00am-10:15am Section B: TUE 10:30am-11:45am, THU 10:30am-11:45am (Liana Harutyunyan) Section C: TUE 10:30am-11:45am, THU 10:30am-11:45am (Gagik Khalafyan) Location: Classroom 408E and 214W Paramaz Avedissian Building

Data Visualization (DS116)

Credit: 3 Description: This course focuses on the art and science of converting raw data into readable and useful visuals. It covers choosing appropriate visualizations for different types of data, principles of visualization, visualization of spatial and text data, and developing visual dashboards. Prerequisite: CS108 (Statistics) Instructor: Habet Madoyan Times: TUE 12:00pm-1:15pm, THU 12:00pm-1:15pm Location: Classroom 214W Paramaz Avedissian Building Was this information helpful? Can I assist you with anything else?

**LLama3:** The courses from DS/data science major that you are interested in are: 1. BUSINESS ANALYTICS FOR DATA SCIENCE (DS227) 2. DATA STRUCTURES/ALGORITHMS IN DATA SCIENCE (DS115) 3. PROGRAMMING FOR DATA SCIENCE (DS120)

These courses are part of the DS/data science major. You can also consider other courses that are prerequisites for these courses or have similar topics. For example, you may want to take "INTRODUCTION TO COMPUTER SCIENCE" (CS110) as a prerequisite for "PROGRAMMING FOR DATA SCIENCE" (DS120).

Let me know if you need more information or have any other questions.

### 17. Provide me with a few courses from CS/computer science major.

**GPT4:** Here are a few courses from the Computer Science major at AUA:

Special Topics in Applied Computer Science (CS290)

Credit: 3 Description: This course explores topics in applied computer science with an emphasis on current technologies and approaches. Topics will be announced prior to course registration. Prerequisite: CS121 (Data Structures) Instructor: Khachatur Virabyan Times: TUE 9:00am-10:15am, THU 9:00am-10:15am Location: Classroom 406E Paramaz Avedissian Building Cybersecurity (CS232)

Credit: 3 Description: Covers various security risks in cyberspace from both offensive and defensive points of view, including subfields such as Web/Mobile Security, Network Security, and Cryptography. Students will develop skills to test the security of systems and build defenses. Prerequisite: CS121 (Data Structures) Instructor: Vaagn Toukharian Times: TUE 12:00pm-1:20pm, THU 12:00pm-1:20pm Location: Classroom 408E Paramaz Avedissian Building Introduction to Object-Oriented Programming (CS120)

Credit: 3 Description: Surveys control structures, functions, arrays, strings, introduction to UML, classes and data abstraction, inheritance, introduction to polymorphism, abstract classes, and interfaces. Students will develop basic proficiency in utilizing and testing software systems. Prerequisite: CS110 (Introduction to Computer Science) Instructor: Varduhi Yeghiazaryan Times: TUE 10:30am-11:45am, THU 10:30am-11:45am Location: Classroom 414W Paramaz Avedissian Building Please note that "Introduction to Computer Science (CS110)" is not offered this semester.

Was this information helpful? Can I assist you with anything else?

**LLama3:** Here are a few courses from the CS/Computer Science major:

1. **CS110: Introduction to Computer Science** (3 credits) - This course provides students with a broad foundation in computer science, covering topics such as digital technology, logic gates, binary systems, and operating systems.

Prerequisite: No prerequisite courses Course section: Not offered this semester Course session: Not offered this semester Course campus: Not offered this semester Course instructor: Not offered this semester Course times: Not offered this semester Taken/seats: Not offered this semester Spaces waiting: Not offered this semester Location: Not offered this semester

2. **CS290: Special Topics in Applied Computer Science** (3 credits) - This course explores topics in applied computer science, with an emphasis on current technologies and approaches. Topics to be announced prior to course registration.

Prerequisite: CS121 (Data Structures) Course section: 0 Course session: 15W Course campus: AUA Main Campus Course instructor: Khachatur Virabyan Course times: Tuesday 9:00am-10:15am, Thursday 9:00am-10:15am Taken/seats: 20/30 Spaces waiting: 0.0 Location: Classroom 406E Paramaz Avedissian Building

3. **CS296: Capstone** (3 credits) - This course provides computer science majors the opportunity to develop the knowledge they have obtained from across the curriculum. Students can choose either a theory or applied project and work on the course-long project under the mentorship of the advising instructor.

Prerequisite: No prerequisite courses Course section: A, C Course session: 15W Course campus: AUA Main Campus Course instructor: Hayk Nersisyan Course times: TBD Taken/seats: 25/25, 17/25 Spaces waiting: 0.0 Location: Course not offered this semester

Please note that the availability of these courses may vary depending on the semester and other factors. I recommend checking with the university's course catalog or advising office for the most up-to-date information.

#### 18. Provide me with a few courses from ENGS/engineering major.

**GPT4:** Here are a few courses from the Engineering major at AUA that are offered this semester:

Computer-Aided Manufacturing (ENGS242)

Credit: 3 Description: The course introduces Computer-Aided Manufacturing technologies in Numerical Control and Reverse Engineering, covering topics such as 3D scanning/coordinate measuring, 3D printing, vacuum forming, composites, NC fundamentals, manual NC programming, and computer-aided part programming. It also touches on material removal processes like metal cutting fundamentals, design for manufacturability, design for machining, and process engineering. Prerequisite: ENGS241 (Computer-Aided Design) Instructor: Sargis Zeytunyan Times: TUE 1:30pm-2:50pm,

THU 1:30pm-2:50pm Location: Classroom 315E Paramaz Avedissian Building Capstone 2 (ENGS299)

Credit: 3 Description: This course provides Engineering Sciences majors the opportunity to develop the knowledge they have obtained from across the curriculum. Students work in teams on the implementation of an applied project, typically with industry partners on real-life engineering problems under the mentorship of the advising instructor. Projects are presented and demonstrated orally at the end of the course, and project reports are submitted in writing. Prerequisite: ENGS298 (Capstone 1) Instructor: Satenik Mnat-sakanyan Times: TBD Location: Course not offered this semester Please note that some courses like "Computer-Aided Design (ENGS241)" and "Introduction to Materials Science (ENGS181)" are not offered this semester.

Was this information helpful? Can I assist you with anything else?

**LLama3:** I can provide you with a few courses from the ENGS/Engineering major. Here are a few options:

1. **ENGS271: Systems Engineering** - This course is a transdisciplinary course that teaches about systems design principles and concepts. It has no prerequisites and is worth 3 credits. Unfortunately, it is not offered this semester.

2. **ENGS241: Computer-Aided Design** - This course is a fundamental course in computer-aided design tools and data structures. It has a prerequisite of **ENGS141: Engineering Statistics** and is worth 3 credits. Unfortunately, it is not offered this semester.

3. **ENGS299: Capstone 2** - This course is a capstone course that allows students to develop a project with industry partners. It has a prerequisite of **ENGS298: Capstone 1** and is worth 3 credits. It is offered this semester, but only has a few seats remaining.

4. **ENGS265: Mechatronic Design** - This course is an introduction to mechatronics and robotic systems. It has a prerequisite of **ENGS241: Computer-Aided Design** and is worth 3 credits. It is offered this semester, and has a few seats remaining.

Please let me know if you would like more information about any of these courses or if you have any other questions!

#### 19. Does calculus 1 have a project?

**GPT4:** The course Calculus 1 (CS100) at AUA does not specifically mention a project as part of its curriculum. The course description highlights topics such as functions of one variable, transcendental functions, introduction to complex numbers, polar coordinates, limits, continuity, derivatives, techniques of differentiation, differentiability, extrema of differentiable functions, applications of differentiation, indefinite and definite integrals, mean value theorem, related-rates problems, and the fundamental theorem of calculus. Students are required to complete weekly problem sets to develop basic proficiency in the mathematical foundations introduced in the field of Calculus. The course includes three hours of instructor-led class time per week, including discussions and problem sets, but does not explicitly mention a project component.

Instructor: Anush Vardanyan Times: MON 9:30am-10:20am, WED 9:30am-10:20am, FRI 9:30am-10:20am Location: Classroom 308E Paramaz Avedissian Building Was this information helpful? Can I assist you with anything else?

**LLama3:** I can provide information about the course, but I need to clarify that the question is asking about a project in the course "CALCULUS 1" with the course code "CS100". Based on the provided course description, there is no mention of a project being a part of the course. The course description mentions that students are required to complete weekly problem sets, but it does not mention a project. If you would like to know more about the course or its requirements, I can provide more information. Please let me know how I can assist you further.

#### 20. Does calculus 1 have an exam

**GPT4:** The course description for Calculus 1 (CS100) at AUA does not explicitly mention exams, but given the standard academic structure, especially for a foundational mathematics course like Calculus 1, it is highly likely that exams are included as part of the assessment criteria. Courses covering topics such as functions, limits, derivatives, and integrals typically assess understanding through mid-term and final exams, in addition to weekly problem sets and possibly quizzes. However, for specific details regarding the number, format, and scheduling of exams, it would be best to consult the course syllabus or contact the instructor directly.

Instructor: Anush Vardanyan If you need more detailed information about the assessment methods or any other aspect of the course, I recommend reaching out to the instructor or the department.

Is there anything else I can help you with?

**LLama3:** Yes, Calculus 1 has an exam. As per the course description, students are required to complete weekly problem sets in order to develop basic proficiency in the mathematical foundations introduced in the field of Calculus. This suggests that there will be an exam or a comprehensive assessment at the end of the course to evaluate student understanding and proficiency. However, the exact format and frequency of the exams are not specified in the provided information.

Please let me know if you need any further assistance.