

Article

Not peer-reviewed version

---

# History Test Answering Support System Based on RAG and LLM Applications in Vietnam

---

[Nguyen Dang](#)\*, Seijoon Kang, Jihong Park, Trang Le

Posted Date: 14 July 2025

doi: 10.20944/preprints202507.1153.v1

Keywords: LLM; RAG; History Education



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# History Test Answering Support System Based on RAG and LLM Applications in Vietnam

Nguyen Dang <sup>1,\*</sup>, Seijoon Kang <sup>1</sup>, Jihong Park <sup>1</sup>, Vy Truong <sup>2</sup>, An Vo <sup>2</sup> and Trang Le <sup>2</sup>

<sup>1</sup> The Brook Hill School, Bullard, Texas, United States

<sup>2</sup> Cam Le High School, Hoa Vang, Da Nang, Vietnam

\* Correspondence: Jobdang123@gmail.com

## Abstract

AI has gone through many stages of development with remarkable advances. Initially, AI applications mainly solved simple problems such as finding routes through symbolic AI and search algorithms. By the 1980s, expert systems became popular in fields such as medicine and diagnosis. However, due to limitations in data and computing power, AI's progress was still limited. The real breakthrough came in the early 2000s when machine learning algorithms were more widely applied, helping AI analyze data effectively, such as image and sound recognition. In particular, since 2012, deep learning with deep neural networks and large amounts of data has helped AI go further, especially in image, voice recognition and object classification. It is for these reasons that we always cherish the desire to create software. The system software can support high school students in studying and looking up history tests. With our passion, we have completed our idea in the form of a website and have had initial success when it was officially launched at <https://historyai-81jl.onrender.com/login>. With the research topic: "System to support answering history test based on RAG application and LLM".

**Keywords:** LLM; RAG; History Education

## I. Introduction

### A. Research Questions, Technical Objectives and Expected Results

#### 1. Research Question

In the current educational context, innovation in methods and integration of technology models are essential requirements to meet the diverse learning needs and comprehensive development of students. The application of technology models not only supports but also creates interest in learning and develops students' self-learning ability. Especially, for subjects that require precision and deep understanding, the application of technology models into the program helps to bring a richer and more effective learning experience.

So there are 11 questions:

1. How to improve the lack of accurate information about the history of high school students?
2. How to develop students' self-study ability in history?
3. Besides serving the needs of history subjects, can it be applied to other subjects?
4. What are the data collection and processing methods?
5. What is the appropriate RAG method for the problem? What are the techniques used?
6. How to generate data? Choose LLM large language model and other
7. Why combine both RAG and LLM models instead of choosing one or the other?
8. Why has large language modeling technology not been widely applied in Education in Vietnam?
9. How can students access and use this model?
10. Is the cost to implement high?

11. So how should we design a website to answer the above questions?
2. Technical objectives
- Create an AI model capable of answering multiple choice questions history
- Generate data to explain multiple choice questions.
- Build a website for students to use conveniently.
3. Expected results
- Generate explanatory data for multiple choice questions with answers (MCQA) based on the Retrieval-Augmented Generation (RAG) method.
- Fine-tune large language models (LLM) to answer multiple-choice questions with explanations.

II. Methodology

Detailed Description of Research Methods and Conclusions

1. Topic overview
- The topic “History test answering support system based on RAG and LLM application” includes 3 main contents.
- Generate explanatory data for multiple choice questions with answers based on the RAG method.
- Fine-tune LLM models to answer multiple-choice questions.
- Build a website to manage automatically updated data for management and a system to answer multiple choice questions with explanations for students.
2. Theoretical basis
- 2.1. Data collection and processing methods
- To collect test questions and book content online, the team used several libraries: Selenium; BeautifulSoup; re; csv; pytesseract.

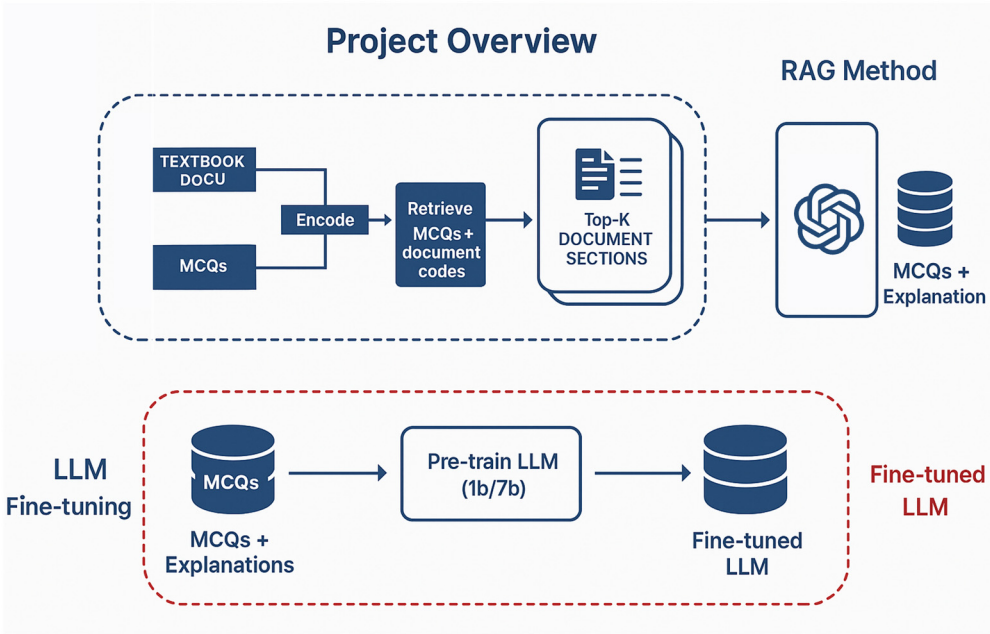


Figure 1. Overview of the topic.

2.2. Retrieval Augmented Generation (RAG) method

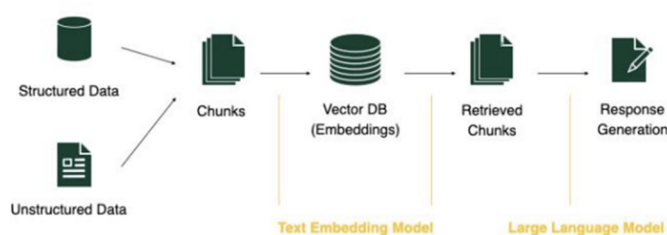
Retrieval-Augmented Generation (RAG) is a model designed to improve the text generation capabilities of a model by integrating the information retrieval process into the generation process. This model combines the power of the retrieval model to find and select relevant information from a

large database or document collection, and the generative model to generate text based on the retrieved information.

RAG works in the following steps:

- Indexing: Documents are divided into blocks, encoded into vectors and stored in a vector database.
- Retrieval: Retrieve the first blocks of text relevant to the problem based on textual similarity and semantic relativity.
- Generation: Input the original question and the retrieved fragments together into the LLM to generate the final answer.

### Simple RAG



**Figure 2.** Simple RAG combined LLM model.

#### 2.2.1. Text encoding (Embedding)

Text embedding is a technique used to transform text into the numerical representations used in the RAG method.

Specifically, “embedding” is the way in which individual words, phrases, or entire sentences are mapped into a high-dimensional vector space, where semantic and syntactic relationships between words are reflected through the distance and orientation of the vector

#### 2.2.2. Passage Retrieval

Passage Retrieval is a systematic process of searching for and returning passages from a set of documents based on a specific query. The goal of this process is to find the passages that are most relevant to the information the student is looking for. Prominent large language models such as OpenAI’s GPT (Generative Pre- trained Transformer) or Google’s BERT (Bidirectional Encoder Representations from Transformers) have demonstrated remarkable progress in the field of natural language processing (NLP).

#### 2.2.3. Chunking method

Chunking is a technique of dividing information into smaller blocks or units, chunking increases efficiency by dividing large data into small blocks for easy management and analysis. The method has two main parameters: Chunk Size (length of text) and Chunk Overlap (length of repetition between two chunks).

#### 2.3. Quantize Low-Rank Adapters (QLoRA) method

Fine-tuning LLM models improves the performance of the model in performing a number of tasks, but the huge scale of LLM models poses a very high computational cost. Traditional fine-tuning methods often require retraining the entire LLM on the target task data, which can be resource-intensive and time-consuming. QLoRA (Quantized Low-Rank Adaptation) is a resource-efficient solution in the fine-tuning process.

#### 2.4. Large Language Model (LLM)

Large language models are a type of artificial intelligence designed to understand and generate natural-sounding text based on learning from large amounts of linguistic data. Built on deep neural networks, these models are capable of processing and analyzing billions of words and phrases to capture the structure and meaning of language. This allows them to perform complex tasks such as

translation, writing, answering questions, summarizing text, and even engaging in conversations with humans.

- 2.5. Mern Stack
- Flexible backend and API with Express and Node.js makes it easy to build and manage endpoints.
  - Connects directly to MongoDB, allowing efficient data storage and easy scaling.
  - Simple client-side routing with React Router, allowing to create SPAs with multiple pages.
  - Flexible development environment with hot-reloading capability in React, allowing changes to be updated without reloading the page.
  - Easy to extend and customize for both backend and frontend. MERN stack helps leverage big data processing capabilities with MongoDB, build powerful APIs with Express, create fast interfaces with React, and deploy easily on the server with Node.js.

III. Result and Data Collection

3.1. Data

3.1.1. Data collection

Data was collected automatically using Python from three websites: Vietjack and Tech12h for collecting multiple choice questions with answers, and O-study for collecting history textbooks. Multiple Choice questions from Vietjack have answer explanations. In contrast, multiple choice questions from Tech 12h do not have answer explanations. The results of the collected multiple choice question data and textbook data are presented in Tables 1 and 2.

Table 1. Statistical description of multiple-choice question data.

Source	Vietjack	Tech12h	Total
Quantity	8912	2069	10981
Percentage	81.2%	18.8%	100%
Answer Ratio (A:B:C:D)	25:26:26:23	23:26:24:27	25:26:26:23

Table 2. Statistical description of textbook data.

Grade	10	11	12	Total
Pages	197	155	209	561
Percentage	35.1%	27.6%	37.3%	100%

3.1.2. Data processing

- Multiple choice question data after collection:
- With answers and explanations:
    - 8912 samples
    - 'id', 'question', 'answers', 'explanation', 'correct\_answer'
  - With answers and without explanations:
    - 2069 samples



- ‘id’, ‘question’, ‘answers’, ‘explanation’, ‘correct\_answer’  
Historical book data after collection:
- 561 samples
- ‘id’, ‘text’  
Preprocessing multiple choice question data:
- ‘question’ column:
- Remove the characters ‘Sentence 1:’, ‘Sentence 1.’, ‘Sentence 2:’, ... at the beginning of the text.
- Remove special characters and non-Vietnamese characters.
- ‘answers’ column:
- Keep the samples containing all 4 answers ‘A’, ‘B’, ‘C’, ‘D’.
- Format the answers in the ‘answers’ column starting with a letter (A, B, C or D) and a ‘.’.
- Remove special characters and non-Vietnamese characters.
- Column ‘correct\_answer’:
- Check and format data into 1 of 4 characters ‘A’, ‘B’, ‘C’, ‘D’.
- ‘explanation’ column:
- If there is no explanation data, replace it with the value ‘null’.
- If there is explanation data, remove special characters and non-Vietnamese characters.
- Preprocessing of historical book data:
- ‘text’ column:
- Unicode Normalization: Function using unicodedata.normalize (‘NFC’, text) to normalize text to a composite Unicode format.
- Standardize Vietnamese accents: Standardize Vietnamese accents, including converting exclamation marks and tone marks from typing style 1 and 2 to the old typing style.
- Convert text to lowercase: Convert all characters in text to lowercase.
- Sentence normalization: Remove unnecessary characters and extra spaces in the text.

3.1.3. Result

- Multiple choice data with explanation (8912 samples).

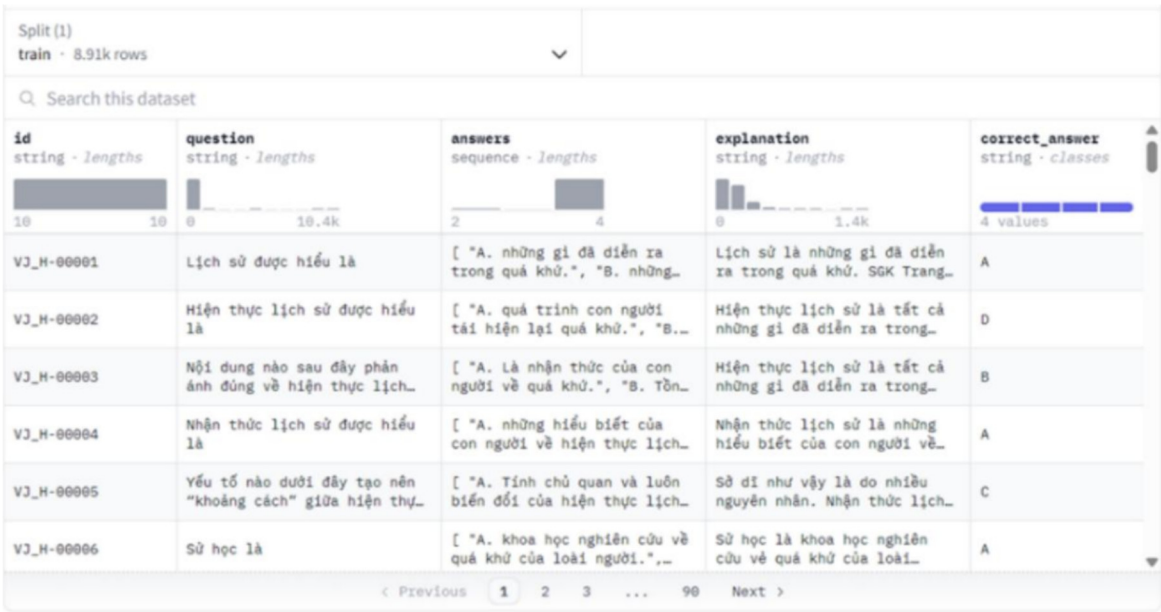


Figure 3. Multiple choice data with explanation.

- Multiple choice data without explanation (2069 samples).

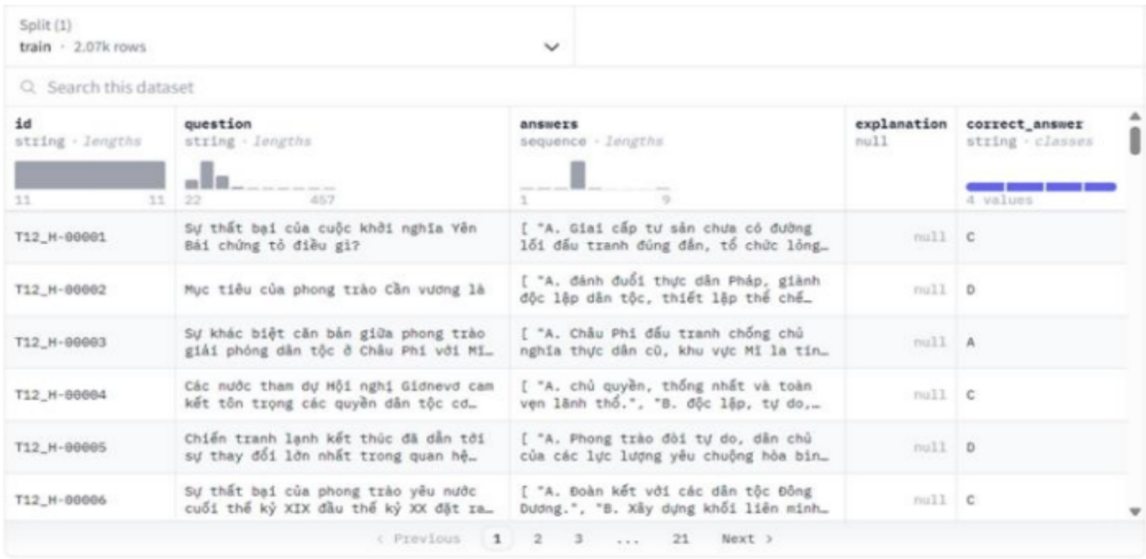


Figure 4. Multiple choice data without explanation.

- Historical book data (561 pages).



Figure 5. Textbook document data.

3.2. Students Explain Multiple Choice Answers Using the RAG Method

3.2.1. Text Encoding (Embedding)

- This problem aims to find the optimal embedding method for extracting relevant text. This process is very important for the ability of RAG to retrieve the most relevant information from the knowledge base for the query.
- The nature of the problem is Vietnamese language, so fine-tuned embedding models will be more suitable. However, the team will still test other optimization models to compare the results. The group got the following results:

Table 3. Comparison table of encoding models.

Model	Result
All-MiniLM-L6-v2	15.62%
sup-SimCSE-Vietnamese-phobert-base	35.83%
vietnamese-bi-encoder	33.94%
Doc2Vec	4.07%

3.2.2. Passage Retrieval

- This problem uses the best results of the text encoding process ( sup-SimCSE- VietNameese-phobert-base model). The retrieval mechanism scans the entire vector in the database to identify knowledge segments (which are passages) which are semantically similar to the student’s query. These passages are then fed into the LLM to add context to the answer generation process.

Table 4. Comparison table of text extraction methods.

Evaluation Method	Result
Cosine similarity	26.21%
Okapi BM25	33.52%
Semantic search	28.04%
Okapi BM25 and Semantic search (coefficient = 0.5)	35.80%

3.2.3. Generation

- Steps to generate an explanation for the question:
- From top-k text segments after the text extraction process, combine into a complete applied knowledge text.
  - Use OpenAI’s application programming interfaces (APIs) to support the process of generating explanations from the above questions and texts. Explanations are only used with information in the applicable knowledge text, if there is no relevant information, no results are returned, this helps ensure historical content is accurate.

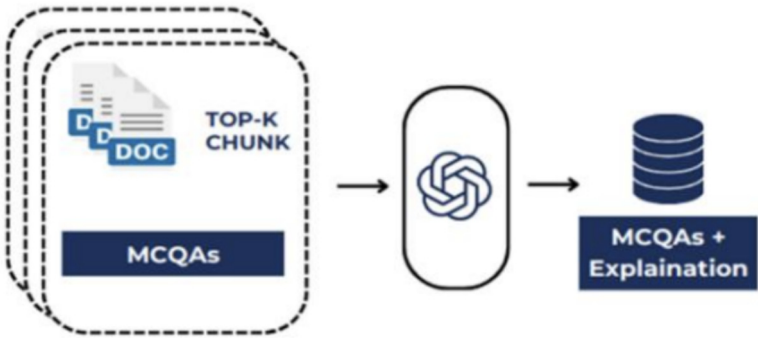


Figure 6. Explanation of the birth diagram.



Table 5. Prompt table explanation.

Case	Syntax	Example
With relevant text	<p>###System Message</p> <p>Applicable knowledge: {Applicable Knowledge}</p> <p>Only use the info in the applicable knowledge section and do not use any other knowledge to explain.</p> <p>If there is no data related to the question, print “null explanation”.</p> <p>###User Message</p> <p>Question: {Question} {Choices}</p> <p>Correct Answer: {Correct Answer}</p> <p>###Explanation (max 2 sentences):</p>	<p>###System Message</p> <p>Applicable knowledge: (Applicable Knowledge)</p> <p>Only use the info in the applicable knowledge section and do not use any other knowledge to explain. If there is no data related to the question, print “null explanation”.</p> <p>###User Message</p> <p>Question: The difference in skin color among races around the world reflects differences in: ['A. level of civilization' 'B. social class' 'C. economic level' 'D. biological characteristics']</p> <p>Correct Answer: D</p> <p>###Explanation (max 2 sentences):</p>
No relevant text	<p>###User Message</p> <p>###Question: {Question} {Choices}</p> <p>###Correct Answer: {Correct Answer}</p> <p>###Explanation (max 2 sentences):</p>	<p>###User Message</p> <p>###Question: The difference in skin color among races around the world reflects differences in: ['A. level of civilization' 'B. social class' 'C. economic level' 'D. biological characteristics']</p> <p>###Correct Answer: D</p> <p>###Explanation (max 2 sentences):</p>

Model achieved:

- Both results are obtained using knowledge from the top-k text segments after segmentation, where k = 4 and the segmentation uses chunk-size = 200 and overlap = 15, respectively.
- Correct knowledge application:
  - Explanation: The difference in skin color between races in the world is a manifestation of human’s long-term adaptation to different natural conditions, not a difference in civilization level, social class or economic level. This reflects the biological diversity of humans in the process of evolution and adaptation to the living environment.
- Wrong knowledge application:
  - Explanation: Null

3.2.4. Results of Generating Explanations for Multiple Choice Questions in History Using the RAG Method

- First execution with gpt-3.5-turbo and related documentation on the entire dataset (10981 questions) resulted in 7719 generated explanations, achieving 70.29%.
- Continued execution with ungenerated questions explained with gpt-4 and related documents, resulting in 1902 generated explanations corresponding to 58.31%.
- The remaining questions are continued to be executed with gpt-3.5-turbo but without the relevant documentation to achieve maximum results.

3.3. Refining the LLM Model for History Multiple Choice Questions

3.3.1. Overview

- From the data generated by the RAG method, the multiple choice question data with answers and explanations will be refined on large language models (LLM) using the Quantized LoRA (QLoRA) method. QLoRA is a method that combines quantization and LoRA (Learnable ReLU Activation) networks to reduce the memory burden and accelerate computation for deep learning models. This technique helps to lighten the Transformer model and improve performance on devices that support 16-bit computing. This topic provides practical results, refining LLM models (1 billion parameters and 7 billion parameters) to perform the task of answering and explaining history test questions.

3.3.2. Implementation Work

- Set up environmental parameters to fine-tune the model.
- Test prompts to improve model results.
- Compare the results from the refined LLM model.

3.3.3. Implementation method

- Fine-tune parameters of LLM model using QLoRA method:

Table 6. fine-tuning environment parameter table using QLoRA.

QLoRA Parameters	BitsandBytes Parameters	Training Arguments
lora_r: 64 lora_alpha: 16 lora_dropout: 0.1	use_4bit = True bnb_4bit_quant_type: nf4 bnb_4bit_compute_dtype: float16 use_nested_quant: False	per_device_train_batch_size: 1 gradient_accumulation_steps: 1 max_grad_norm: 0.3 learning_rate: 2e-4 weight_decay: 0.001 optim: paged_adamw_32bit lr_scheduler_type: cosine max_steps: 1 warmup_ratio: 0.03 logging_steps: 25

- Prompt board used

Prompt	Syntax	Example
Fine-tune	###QUESTION: {Question} {Choices} ###ANSWER: {Correct answer} EXPLAIN: {Explanation}	###QUESTION: History helps us understand... Choices: A, B, C, D ###ANSWER: D EXPLAIN: Explanation in max 2 sentences

- Finetuned LLM results

Table 7. Summary of fine tuned LLM results.

Model	Param	Data	Repeats	Time (h)	Train Val(%)	Test (%)
Bloomz	1B	TRAIN	10	5	31	26

Bloomz	1B	TEST	20	2.5	34	32
Bloomz	1B	TRAIN	5	2.5	32	26
Bloomz	1B	TRAIN	1	0.5	36	25
Bloomz	1B	TRAIN/10	1	0.2	28	25
Bloomz	7B1	TRAIN/10	1	0.5	33	28
Bloomz	7B1	TRAIN	1	2.5	49	42
Llama	7B	TRAIN/10	1	0.5	65	27
Llama	7B	TRAIN	1	2.5	52	43
VinaLlama	7B	TRAIN	1	2.5	50	30
PhoGPT	4B	TRAIN/10	1	6	28	23
PhoGPT-Chat	4B	TRAIN/10	1	6	29	24
Qwen	7B	TRAIN	1	2.5	70	55

Comment on results:

- The larger the model size, the higher the result but the cost of fine-tuning the model will be larger; The training time between Bloomz 1B and Bloomz 7B1 models with 1 epoch, TRAIN data has a difference when the time is 0.5h and 2.5h respectively.
- Increasing the number of iterations (epochs) during the fine-tuning process does not greatly affect the model results; In the Bloomz 1b model with iterations of 1, 5 and 10 on the TRAIN dataset, the accuracy on the TRAIN\_VAL and TEST datasets does not increase with the number of epochs.
- The more training data increases, the higher the model accuracy on the TRAIN\_VAL set; In the Llama 7B model or the Bloomz 7B1 model, the Bloomz 1B model with the TRAIN and TRAIN/10 data sets, there is a difference in the TRAIN\_VAL result when the accuracy is lower in the fine-tuned model on the TRAIN/10 set.
- LLM models that do not support the QLoRA method have significantly longer tuning times, which means consuming more computational resources; PhoGPT and PhoGPT-Chat models with TRAIN/10 data and 4B parameters have longer training times than models with 7B parameters and larger training data of 6h.
- The model with the best accuracy is Gwen 7B with 70% on the set TRAIN\_VAL and 55% on the TEST set.

3.4. Website Building

3.4.1. Back-End

The system’s backend is built using Node.js and Express, providing RESTful API for activities related to historical questions. The system allows students to register, log in, and perform Q&A activities, saving the history of questions asked. The main features include student management, question content management, and Q&A history. In addition, the team integrates APIs from Hugging Face to process and load AI models for Q&A.

Steps to follow:

- Set up the environment and install the necessary libraries.
- Configure Express application.

- Design MongoDB database models with Mongoose:
  - + User: Student information
  - + Conversation: History of Q&A conversations
  - + Content: Historical question data

### 3.4.2. Front End

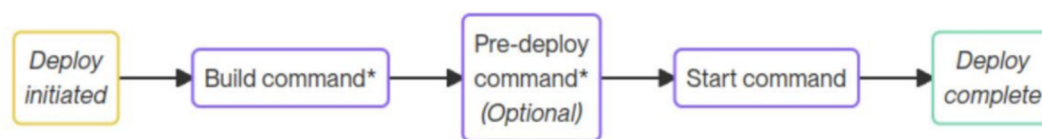
The front end of the system is built using React within the Node.js framework, allowing server-side rendering (SSR) and static page generation (SSG). The system provides a student-friendly interface so that students can register, log in, and ask and answer history questions intuitively and easily.

Steps to follow:

- Set up the environment and install the necessary libraries.
- Nodejs application configuration (libraries, config files, etc.).
- Student-friendly interface design.
- Integrate interface with API from backend to handle Q&A requests and student operations.

### 3.4.3. Server Configuration

Deploying a system onto a server has never been a simple matter, especially ensuring that the system runs stably when there are many students. Render is another cloud platform that offers simple and powerful web application deployment, including support for deploying Node.js applications like your MERN stack. Render makes it easy to deploy, scale, and manage applications with notable features to ensure high performance and stability in production environments.



**Figure 7.** Steps to deploy the system using Render.

### 3.5. Finished Product

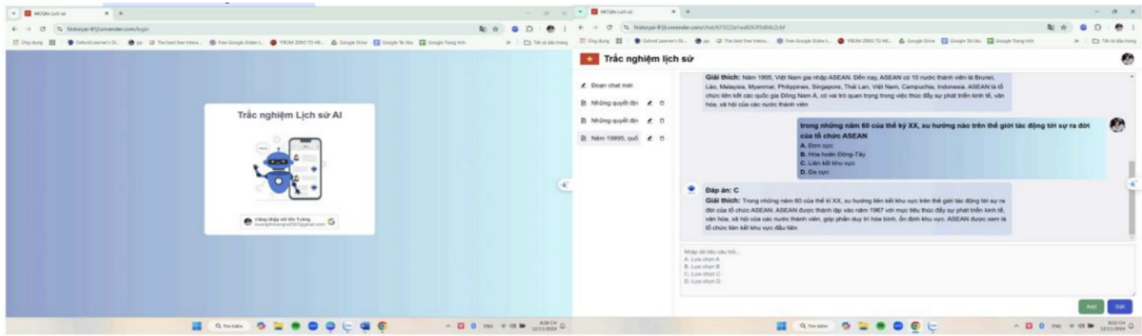
Students can access the Web in two ways:

**Method 1:** Scan QR Code



**Method 2:** Link: <https://historyai-81jl.onrender.com/chat/newchat>

- System website interface supports answering history quiz questions on computer.



- In addition, the website of the system that supports answering history quiz questions can also be used on phones, tablets...

IV. Conclusion

4.1. Novelty

- Use the RAG method to generate explanatory data.
- Using QLoRA method to fine-tune large language models LLM.
- Use new technologies to build websites.
- Control input data sources to ensure accuracy of history compared to other Chatbots.

4.2. Creativity

- Combine LLM model with RAG, can be linked with Websites to form a complete system.
- Build your own initial dataset.
- ChatGPT API integration to help with RAG.
- New questions that the system has not recorded will be saved to retrain the model to help increase the accuracy rate.

4.3. Replicability

- The product is easy to upgrade, maintain and has high replicability depending on the number of students accessing it, the number of students can be upgraded to more.

4.4. Product Existence

- The system is still capable of giving wrong answers in some special cases, such as missing punctuation or misspelled questions.
- The product uses a third party, HuggingFace, to store the model, so there is a fee to use it.

4.5. Development Direction

- Use a larger LLM model and add additional data to make the model more accurate.
- Build more LLM large language models to apply to more subjects in the Website.
- Upgrade system traffic and increase information processing speed believe
- Upgrade Website to application and add features smart like reading, scanning, recognizing.
- Upgrade and add more references from textbooks to increase accuracy and help students understand the answers more deeply.
- Integrate more AI Models so that students have more sources of answers to compare and refer to.

4.6. Conclusion

The trained model is able to answer and explain history multiple choice questions but there is still an error rate. The website has been used by students and is effective in learning history.



## References

1. Vailaya, A. (2002). Automatic image orientation detection. *IEEE Transactions on Image Processing*, 11(7), 746–755. <https://doi.org/10.1109/TIP.2002.801590>
2. Dang, A. T., Huynh, T. M. T., & Tran, V. L. (n.d.). A method for building an intelligent assistant application based on self-instruction techniques for fine-tuning LLMs. (Source not publicly available online—citation incomplete)
3. Ngo, V. T. H., Dao, D. T., & Tran, T. M. H. (2023). The application of federated learning with FedBN algorithm for breast cancer detection. In *Proceedings of the 16th National Conference on Fundamental and Applied Information Technology Research (FAIR'2023)*. [https://vap.ac.vn/Portals/0/TuyenTap/2024/2/21/64e13532907845ed9f5a2547dfec276f/15BB\\_FAIR2023\\_paper\\_4435.pdf](https://vap.ac.vn/Portals/0/TuyenTap/2024/2/21/64e13532907845ed9f5a2547dfec276f/15BB_FAIR2023_paper_4435.pdf)
4. Ninh, K. C., Tu, K. N., Vo, T. H., & Ninh, K. D. (2023). Applications of recurrent neural network in fake news classification. In *Proceedings of the 16th National Conference on Fundamental and Applied Information Technology Research (FAIR'2023)*. <https://elib.vku.udn.vn/handle/123456789/4138>
5. Nguyen, D.-V., & Nguyen, Q.-N. (2023). Evaluating the symbol binding ability of large language models for multiple-choice questions in Vietnamese general education. In *Proceedings of the 12th International Symposium on Information and Communication Technology* (pp. 177–183). <https://doi.org/10.1145/3628797.3628837>
6. Nguyễn, T. T. N., Bùi, T. L., & Nguyễn, Đ. D. (2017). Một phương pháp phân đoạn ký tự cho bài toán nhận dạng chữ viết tay online. In *FAIR - Nghiên cứu cơ bản và ứng dụng công nghệ thông tin*. (Source not found online—citation incomplete)
7. Võ, D. N., & Đinh, Đ. (2020). Các tiêu chí ngôn ngữ trong việc xây dựng kho ngữ liệu tiếng Việt. In *Proceedings of the 13th National Conference on Fundamental & Applied Information Technology Research*. [https://vap.ac.vn/Portals/0/TuyenTap/2021/6/18/363b8d0087164f4ca9882cae5d2230d0/4\\_FAIR2020\\_paper\\_118\\_637596274935007124.pdf](https://vap.ac.vn/Portals/0/TuyenTap/2021/6/18/363b8d0087164f4ca9882cae5d2230d0/4_FAIR2020_paper_118_637596274935007124.pdf)
8. Dương, T. L. (2023). Ứng dụng ChatGPT thúc đẩy dạy và học bậc đại học trong kỷ nguyên trí tuệ nhân tạo. *Tập san Khoa học và Kỹ thuật Trường Đại học Bình Dương*, 6(2). <https://doi.org/10.56097/binhduonguniversityjournalofscienceandtechnology.v6i2.160>
9. Daly, C. (2023). Learning to rank: Performance and practical barriers to deployment in enterprise search. In *2023 3rd Asia Conference on Information Engineering (ACIE)*. IEEE. <https://www.computer.org/csdl/proceedings-article/acie/2023/331200z005/1PONGtsTMtwo>

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.