**VIETNAM NATIONAL UNIVERSITY, HANOI UNIVERSITY OF ENGINEERING AND TECHNOLOGY**


**Nguyen Duc Cuong**

# EDUTECH CHATBOT FOR PERSONALIZED LEARNING

**GRADUATION THESIS**

**Major: Computer Science**

**HA NOI - 2025**

**VIETNAM NATIONAL UNIVERSITY, HANOI UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

**Nguyen Duc Cuong**

# EDUTECH CHATBOT FOR PERSONALIZED LEARNING

**Major: Computer Science**

**Supervisors: Assoc.Prof.Dr. Nguyen Viet Anh**
**PHD. Nguyen Hoa Huy**

# ABSTRACT

**ABSTRACT**

Personalized learning is a way of teaching that focuses on adapting lessons and materials to fit each student's needs. This thesis is about an EduTech Chatbot made for personalized learning. The chatbot helps improve the learning experience by using data to give custom recommendations. With Machine Learning (ML) and Natural Language Processing (NLP), it can understand what users say, keep track of their progress, and suggest learning materials, exercises, or study paths that suit them.

The system can collect and manage learner data, making sure each user has a complete and personalized learning journey. It also gives instant answers to questions and useful feedback, so learners can stay involved and take control of their own learning. By using smart technology with a simple design, this chatbot can change the way we learn, supporting both students and teachers to get better results.

This thesis shows how EduTech can make learning easier, faster, and more personalized, filling the gaps in traditional education.

***Keywords:*** personalized learning, machine learning, natural language processing, artificial intelligence.

# COMMITMENT

I hereby declare that all the detailed content of the graduation thesis is done by myself under the guidance of Mr. Nguyen Viet Anh. All reference documents have been clearly identified and there is no action when copying documents without identifying the source and author. If any wrongdoing is detected, I will take full responsibility before all regulations of the University of Technology.

Ha Noi, …… 2025

Student

# ACKNOWLEDGEMENTS

To begin with, I am grateful to those who have helped and taught me, so that I can have a complete and well-prepared graduation thesis. With the enthusiastic guidance and instruction of teachers, family and peers, I have developed more than ever.

I would like to send my most respectful greetings to the teachers of the University of Technology, along with my best wishes for health and my most sincere thanks. Thanks to the knowledge and help of the teachers, I was able to complete my thesis. In particular, I would like to express my appreciation to my teacher, Associate Professor, Dr. Nguyen Viet Anh, for his care and help in the process of completing this thesis.

I sincerely thank you!

# Table of Contents

# Acronyms

| | |
|---|---|
| AI | Artificial Intelligence |
| ML | Machine Learning |
| NLP | Natural Language Processing |
| LLM | Large Language Model |
| LMS | Learning Management System |
| APIs | Application Programming Interfaces |
| JSON | JavaScript Object Notation |
| WSGI | Web Server Gateway Interface |
| AWS | Amazon Web Services |
| SSQ | Structured Semantic Query |
| EC2 | Elastic Compute Cloud |
| RDS | Relational Database Service |
| POS | Part of Speech |
| NLTK | Natural Language Toolkit |
| BERT | Bidirectional Encoder Representations from Transformers |
| GPT | Generative Pre-trained Transformer |

| | |
|---|---|
| PhoBERT | Vietnamese BERT |
| SQL | Structured Query Language |
| BSON | Binary JSON |
| CSS | Cascading Style Sheets |
| UI | User Interface |
| DOM | Document Object Model |
| CI | Continuous Integration |
| CD | Continuous Deployment/Delivery |
| CPU | Central Processing Unit |
| RAM | Random Access Memory |

# List of Figures

# CHAPTER 1: INTRODUCTION

## 1.1. Problem Statement

Recently, personalized learning becomes a popular method, with education no longer being confined to traditional classrooms but expanding to online platforms and personal support tools. The development of educational solutions/systems has gained traction. However, most existing learning systems still follow a "one-size-fits-all" approach, resulting in inefficiencies in addressing the diverse learning needs of individuals.

The inspiration for this thesis arises from the challenges many learners face: a lack of personalized guidance, difficulty in determining the learning path, and the frustration of learning without receiving appropriate feedback. At the same time, with the rapid development of AI and NLP capabilities, a question arises: "Why not leverage technology to create a learning companion that not only understands the learner but also helps them progress further?"

This idea becomes even more compelling when considering that education is the key to transforming lives. A personalized learning chatbot is not just a support tool; it can serve as a bridge for learners to easily, effectively, and inspiringly access knowledge. This is the driving force behind the creation of this thesis, with the hope of enhancing the learning experience for millions of people and making a significant leap in applying EduTech in education.

## 1.2. Product Objective

**Main Goal:** Develop an online tool that suggests personalized solutions for each learner in finding knowledge and an appropriate development path.

**Core Functionalities the System Must Achieve:**

1. **Ask Questions and Receive Answers**

    - Users can ask questions related to learning topics, materials, assignments, or any relevant subjects.

- The chatbot provides direct, easy-to-understand answers or step-by-step guidance to solve the problem.

2. **Receive Personalized Learning Path**

   - Users provide information about their goals, skill level, or areas of interest.
   - The chatbot suggests a personalized learning path, including a list of lessons, assignments, and time management tips.

3. **Receive Learning Suggestions**

   - The chatbot suggests lessons, exercises, or new content based on the user's study habits and preferences.
   - It can recommend courses, materials, or quizzes to help reinforce knowledge.

4. **Engage in Conversations and Clarify Doubts**

   - Users can interact with the chatbot to clarify any learning-related doubts or issues concerning the system.
   - The chatbot responds naturally and friendly, simulating the experience of conversing with a tutor.

5. **Search for Learning Materials**

   - Users can ask the chatbot to provide learning materials, books, or instructional videos relevant to their study topics.
   - Materials may be linked to online resources or the LMS.

6. **Receive Timely Learning Support**

   - When facing difficulties in their studies, users can request the chatbot for solutions, suggestions, or detailed explanations.
   - The chatbot helps users overcome obstacles and stay motivated in their learning journey.

7. **Personalize Settings**

- Users can customize their profile, preferred language, or learning goals to help the chatbot serve them better.

8. **Receive Notifications and Reminders**

- Users receive notifications about assignment deadlines, quizzes, or important learning events.

- The chatbot sends periodic reminders to help users maintain their study habits.

## 1.3. Thesis Scope

The thesis focuses on:

- **Target Users:** Students, learners, and independent learners who require personalized learning experiences.

- **Key Features:** Automated question answering, creation of personalized learning paths, progress tracking, and integration with Learning Management Systems (LMS).

- **Technologies Applied:** Flask for backend, MongoDB for database management, and the use of ML and NLP models for personalized learning solutions.

- **Limitations:** The system, in its initial phase, will focus on specific learning areas. It will gradually expand to cover more topics based on user feedback.

## 1.4. Research Methodology

- **Data Collection:**
  Conduct surveys and gather feedback from potential users (students, learners) to understand their needs and challenges in learning.

- **Analysis and Design:**

Use the Agile software development methodology to ensure continuous improvement of the product based on user feedback.

- **AI Model Development:**

Apply ML and NLP techniques to build a chatbot that can learn and improve over time.

- **Testing and Evaluation:**

Perform performance testing and evaluate the chatbot's ability to meet requirements across various real-life scenarios.

# CHAPTER 2: THEORETICAL BASIS

## 2.1. Background and related work

### 2.1.1. Background

**Personalized Learning** is an educational approach in which content, teaching methods, and learning pace are tailored to each learner's needs, interests, and abilities. Unlike traditional education, where all learners follow the same learning path, personalized learning allows learners to have a flexible and optimized learning path based on their individual characteristics.

Based on the concept of personalized learning method, the Edutech chatbot system will be structured around these following factors:

1. Data collection and analysis: The system will collect information about users, including goals, interests, learning styles and learning outcomes. With the collected data, the system will analyze and provide solutions suitable for users.

2. Customized learning path: Based on the analyzed data, the system recommends learning materials, exercises and paths suitable for each learner.

3. Adaptive feedback: The system provides real-time feedback on learning progress, helping learners adjust their learning methods effectively.

4. Progress tracking: The system stores and analyzes users' learning data, allowing them to track and evaluate their progress.

### 2.1.2. Related work

Personalized learning has become a significant focus in modern educational technology (EduTech), with numerous models and systems developed to enhance the learning experience for individuals. Among these, two notable models are EduChat and CHAT-ACTS, both of which provide valuable insights into personalized learning through chatbot systems.

**EduChat: An Educational Chatbot System Based on Large Language Models**

EduChat is an educational chatbot system developed based on LLM. This system aims to support intelligent, personalized and equitable education for users, including teachers, students and parents. The outstanding features of the EduChat model include:

- Goal: Provide real-time educational support through chatbots, helping learners access knowledge and receive direct guidance.
- Technology: EduChat is built on LLM, including advanced language models such as GPT, trained on rich educational data and fine-tuned with specific system instructions.
- Functions:
    - Answering open questions: The system is capable of answering theoretical and practical questions in many different fields.
    - Essay evaluation: The system can read, analyze and evaluate learners' essays, providing specific feedback.
    - Socratic teaching support: The system guides learners through open-ended questions, helping learners discover knowledge on their own.
    - Emotional support: In addition to academic support, the system also provides motivational advice and psychological support for learners.
- Implementation: As an open-source project, EduChat gives source code and models on platforms including GitHub and Hugging Face.

Limitations of EduChat:

- Lack of deep personalization for each learner because it focuses mainly on text analysis.
- Unable to maintain tracking of learners' learning progress over time.
- Limited ability to evaluate learning performance, focusing only on answers without analyzing overall progress.

**CHAT-ACTS: A Framework for Personalized Educationals Chatbots**

CHAT-ACTS is a framework designed to integrate chatbots into personalized learning environments, focusing on two key aspects: Self-Regulated Learning (SRL) and Active Learning.

- Objective: Support learners to actively participate in the learning process through personalized chatbots, helping them self-regulate their learning process and develop independent thinking skills.
- Components: The CHAT-ACTS framework includes three main learning modes:
    - Personalized Learning: The chatbot system is adjusted based on the learner's personal information such as goals, learning styles and interests.
    - Self-Regulated Learning (SRL): Learners are encouraged to set their own goals, plan and monitor their learning progress.
    - Active Learning: The system supports interactive learning activities, helping learners participate in the process of knowledge discovery.
- Strategies:
    - Feedback and Assessment: The chatbot system provides instant feedback and continuous assessment to help learners adjust their learning methods.
    - Setting goals: Learners are encouraged to set clear learning goals and track their progress.
    - Supporting learning strategies: The system suggests effective learning methods, such as note-taking, deep reading, and group discussion.
- Limitations of CHAT-ACTS:
    - High complexity in practical implementation due to multiple learning modes and configuration elements.
    - Lack of the ability to automatically adjust learning content based on learner progress.

- No integration with popular LMS platforms, making it difficult to manage learner data.

**Improvements in EduTech Chatbot**

While both EduChat and CHAT-ACTS provide essential foundations for personalized learning, they also present several limitations that the EduTech Chatbot system in this thesis aims to address. Specifically, the EduTech Chatbot is designed to:

- Offer personalized learning paths that adapt to individual learner styles and preferences.
- Provide continuous progress tracking and adaptive feedback to help learners adjust their learning strategies.
- Maintain a comprehensive learner profile, enabling personalized content recommendations.
- Seamlessly integrate with popular LMS platforms for better learner data management.

By overcoming the limitations of existing models, the EduTech Chatbot aims to provide a more robust, scalable, and user-friendly personalized learning experience.

## 2.2. Technologies Used

During the development of the system, popular programming languages and database management systems were utilized, including Python, MongoDB, and JavaScript. With outstanding performance and a growing development community, these technologies will support the system for an extended period, ensuring the best user experience.

### 2.2.1. Backend

**Python:** A high-level, versatile, and powerful programming language, making it ideal for backend development, especially in projects involving artificial intelligence and data processing.

**Key features of Python in backend development:**

- **Versatility:** Python supports procedural, object-oriented, and functional programming models, providing flexibility for many application designs.
- **Rich library ecosystem:**
    - Flask, Django: Web development frameworks.
    - SQLAlchemy, PyMongo: Database interaction.
    - Celery: Asynchronous task processing.
    - TensorFlow, PyTorch: AI and Machine Learning development.
- **Readability:** Python's simple and readable syntax allows development teams to collaborate efficiently and maintain the codebase easily.
- **Scalability and integration:** Python integrates easily with cloud services like AWS, Google Cloud, and Azure.

**Role of Python in the project:**

- Developing data processing models.
- Integrating RESTful APIs with Flask.
- Building business logic modules, including the chatbot and recommendation engine.

**Flask:** A microframework for Python designed to quickly and efficiently build web applications and RESTful APIs.

**Key features of Flask:**

- **Lightweight and flexible:** Flask provides the basic components needed for web application development without enforcing a fixed structure, offering developers flexibility in designing the system architecture. It is suitable for small to medium-sized projects or applications requiring high customization.
- **Extendable:** Flask supports various extensions to enhance functionality, such as:

- Flask-SQLAlchemy: Interaction with SQL databases.
- Flask-PyMongo: Interaction with MongoDB.
- Flask-RESTful: Building RESTful APIs.
- Flask-JWT-Extended: Authentication and authorization using JSON Web Tokens.

- **WSGI integration:** Flask is based on the WSGI protocol, ensuring compatibility with web servers like Gunicorn or uWSGI.
- **Rapid development support:** Flask integrates with debug mode, helping developers quickly identify and fix issues during the development process.

**Role of Flask in the project:**

- Providing RESTful APIs to connect the frontend and backend.
- Handling chatbot logic, such as managing user requests, analyzing data, and returning responses from AI models.
- Managing learning progress, storing data in MongoDB.

### 2.2.2. AI/ML

Machine Learning, a branch of Artificial Intelligence, focuses on developing algorithms that allow systems to learn from data and make predictions or decisions without requiring specific programmed for each situation.

**Theoretical Foundation of Machine Learning:**

- **Types of Machine Learning:**

1. **Supervised Learning:** Uses labeled data to train models. Example: Predicting a learner's knowledge level based on previous questions and answers.
2. **Unsupervised Learning:** Works with unlabeled data. Example: Clustering learners based on their study habits.

3. **Reinforcement Learning:** The system learns how to act through feedback from its environment.

- **Basic ML Pipeline:**

    1. **Data Collection.**
    2. **Data Preprocessing.**
    3. **Model Training.**
    4. **Model Evaluation & Optimization.**

**Applications of ML in the project:**

- **User Data Analysis:** Identifying learning styles and personalized goals.
- **Learning Material Suggestions:** Based on user's learning history and needs.
- **Predicting Learning Trends:** Using ML to predict the next area of focus for the learner.

**Natural Language Processing:** NLP is a field within AI focused on processing, analyzing, and understanding natural language (e.g., English or Vietnamese) using mathematical models and ML techniques.

**Theoretical Foundation of NLP:**

- **Basic Steps in NLP:**

    - **Tokenization:** Separating text into smaller units, such as words or sentences.
    - **Stemming/Lemmatization:** Normalizing words to their root forms.
    - **POS Tagging:** Identifying parts of speech (e.g., noun, verb, adjective) in sentences.
    - **Named Entity Recognition:** Recognizing entities like names, locations, and dates.
    - **Sentiment Analysis:** Analyzing emotions in text.

- **NLP Models and Libraries:**
    - **NLTK:** A basic NLP toolkit suitable for tasks like tokenization, stemming, and POS tagging.
    - **spaCy:** A modern library optimized for NLP tasks, providing ready-to-use models like NER and dependency parsing.
    - **Transformers (Hugging Face):** A library integrating advanced NLP models like BERT, GPT, and T5, offering context-based understanding and natural text generation.
- **Advanced Models:**
    - **BERT:** Analyzes context bidirectionally, enabling the chatbot to accurately understand the user's question.
    - **GPT:** A model that generates natural language text, aiding in intelligent conversational responses.

**Applications of NLP in the project:**

- **Context Understanding:** The chatbot accurately interprets user questions, even when they are unclear.
- **Answer Generation:** Based on models like GPT, the chatbot provides natural and friendly answers.
- **Vietnamese Language Processing:** Utilizing spaCy or custom BERT models (PhoBERT) for Vietnamese language understanding.
- **Conversation Data Analysis:** Tracking and categorizing content that learners are interested in.

**AI/ML Tools and Libraries Used in the Project:**

1. **TensorFlow:**

- **Theoretical Foundation:** TensorFlow is an open-source library from Google designed for building deep learning models. It provides tools for deployment across various platforms (desktop, cloud, mobile).

- **Real-world Application:** TensorFlow will be used to train AI models and optimize the chatbot system.

2. **PyTorch:**

   - **Theoretical Foundation:** Created by Facebook, PyTorch is an open-source library for deep learning, offering high customization and fast model development.

   - **Real-world Application:** PyTorch is suitable for experimenting with new models, especially in NLP tasks.

3. **Transformers (Hugging Face):**

   - **Theoretical Foundation:** The Transformers library integrates powerful NLP models like BERT and GPT. These models use the Transformer architecture for precise contextual analysis.

   - **Real-world Application:** The pre-trained models such as GPT or PhoBERT will be used to deploy a chatbot that can understand and respond to natural language.

**Benefits of Applying AI/ML in the Project:**

- **Personalized Learning Experience:** AI/ML tailors the learning path for each individual based on data and behavior analysis.

- **Natural Interaction:** NLP enhances the chatbot's ability to communicate naturally, simulating human interaction.

- **Learner Data Analysis:** ML helps to extract valuable insights from data, detecting trends, and improving learning content.

### 2.2.3. Database

**Overview:**

MongoDB is a NoSQL database designed for storing unstructured data and provides flexible scalability. It is widely used in systems where the structure of the data may change or isn't predefined, making it ideal for projects that require flexible data handling, such as the EduTech Chatbot for Personalized Learning.

**Theoretical Foundation of MongoDB:**

- **NoSQL Database Architecture:**
    - **NoSQL (Not Only SQL):** Unlike traditional relational databases (SQL), NoSQL databases like MongoDB don’t require tables and relational joins. They use more flexible data structures like documents, key-value pairs, or graphs.
    - **Document-Oriented Database:** MongoDB stores data in documents (in JSON or BSON format), allowing for a schema-less design where each document can have a different structure. This offers greater flexibility for storing unstructured data.

**Advantages of MongoDB:**

1. **Scalability:**
MongoDB supports both horizontal and vertical scaling, making it easy to handle large volumes of data efficiently. This flexibility is crucial as the EduTech chatbot grows and handles more user data.

2. **High Performance:**
MongoDB is optimized for fast read and write operations, essential for applications like chatbots where quick data retrieval is needed.

3. **Handling Unstructured Data:**
MongoDB can store unstructured data (e.g., user responses, feedback, or learning history) without predefined schemas, which is ideal for the EduTech chatbot’s needs.

**Data Management and Querying in MongoDB:**

- **Collections:**

Similar to tables in SQL, MongoDB stores data in collections, where each collection can hold multiple documents.

- **Documents:**

Documents are records within collections, and each document can contain custom fields and values, allowing for flexibility in data storage.

- **Indexing:**

MongoDB supports indexing to optimize query performance, which can improve the speed of data retrieval—an essential feature for large datasets.

**MongoDB Data Types:**

- **String:** For text data, like usernames or user queries.
- **Number:** For numeric data, such as scores or study time.
- **Array:** For storing lists of values, such as a list of questions asked by a user.
- **Object:** For nested documents, like storing a user's profile with details such as name, email, and learning history.
- **Boolean:** For true/false values, like whether a lesson is completed.

**Applications of MongoDB in the EduTech Chatbot Project:**

1. **Storing User Information:**

MongoDB will store user data such as name, email, learning history, questions asked, chatbot responses, and learning goals.

2. **Storing Questions and Answers:**

Questions and answers from users will be stored for future reference, allowing the chatbot to reference past interactions.

3. **Tracking Learning Progress:**

MongoDB can store learning metrics like scores, time spent on lessons, and completion levels, helping to monitor users' progress.

4. **Analysis and Recommendations:**

Data stored in MongoDB will be analyzed to provide personalized learning suggestions, helping the chatbot guide users through customized learning paths.

**Managing the Database with MongoDB:**

- **Running MongoDB:**

MongoDB can be deployed locally or on cloud services such as MongoDB Atlas, which offers a managed cloud database solution for easier maintenance and scaling.

- **Connecting MongoDB with Flask:**

Flask can interact with MongoDB using the Flask-PyMongo extension, which provides an easy way to integrate MongoDB with Flask-based APIs.

- **Creating Queries:**

MongoDB supports flexible querying with simple syntax, making it easy to search and analyze data for the chatbot's needs.

**Benefits of Using MongoDB in the Project:**

1. **Flexible Scalability:**

MongoDB's ability to scale horizontally and vertically ensures that as user data and learning histories grow, the system remains high-performing and responsive.

2. **Effortless Handling of Unstructured Data:**

MongoDB's schema-less nature allows it to easily store diverse, unstructured data types such as chatbot conversations, user responses, and learning goals.

3. **Easy Integration with Flask and AI/ML:**

MongoDB's smooth integration with Flask and support for machine learning models make it a suitable choice for storing and managing data in this AI-powered chatbot project. It also enables the optimization of AI/ML models with real-time data.

### 2.2.4. Frontend

**Overview:**

React.js, a popular JavaScript library created by Facebook, is builed for web UI and mobile applications. It provides a flexible and efficient method to creating highly resuable interactive UI components.

**Theoretical Foundation of React.js:**

1. **Component-Based Architecture:** React follows a component-based architecture where each component is a self-contained block of code that can be reused and

combined to form the user interface. Components can range from simple elements (like a button or header) to complex structures (like dashboards or forms). React makes it easy to manage UI components, ensuring that changes or updates in one component do not affect others.

2. **Virtual DOM:** A React's concept improves the performance of the application by decreasing the number of interactions with the actual DOM. Upon detecting a change, React first updates the Virtual DOM and then compares it with the real DOM. Only the necessary changes are applied, ensuring efficient re-rendering.

3. **State Management:** React provides a state management system to track and store values that change dynamically in the interface. When the state is modified, React automatically updates the relevant UI components. In larger applications, Libraries like Redux or React's Context API can be used to manage global state.

4. **Props (Properties):** Props are used to pass data between components in React. A child component receives data from a parent component via props and uses that data to render its UI.

5. **React Router:** React Router is a library designed for managing navigation in React applications. It enable users to navigate between different pages of the application without reloading the page, making the app behave like a single-page application.

**Benefits of Using React.js in the Project:**

- **Code Reusability:**

With its component structure, React promotes code reuse, reducing the need to rewrite similar UI elements.

- **High Performance:**

The Virtual DOM boosts the speed of rendering, making React highly performant, even when dealing with large numbers of UI components.

- **Scalability:**

React is suitable for complex and scalable applications, especially when combined with state management tools like Redux.

- **Extensive Ecosystem:**

React has a extensive ecosystem with various libraries and tools, such as React Router and Redux, that further enhance its functionality and performance.

### 2.2.5. Deployment and Operations:

**Vercel and Heroku/AWS:**

**Theoretical Basis:**

- **Vercel** and **Heroku/AWS** are modern deployment platforms that make developing, deploying, and operating web applications simple and efficient. Both platforms support **CI/CD,** automating the software development process to ensure fast and stable updates and deployments.
    - **Vercel**: Designed specifically for front-end applications and projects using JavaScript frameworks like Next.js and React.js. It provides **serverless functions** (running code on the server without the need for server management).
    - **Heroku/AWS**: These platforms provide environments for deploying the backend. AWS offers services such as EC2, Lambda, and RDS to deploy, store, and manage backend applications and databases. Heroku is also an easy-to-use platform for backend applications with quick integration and support for multiple programming languages.

**Practical Application:**

- **Vercel**: Used for deploying the frontend of an application, helping manage dynamic websites, optimize page load speed, and automatically deploy changes in the source code.
- **Heroku/AWS**: Typically used for the backend and API. Heroku is easy to use with automatic scaling, while AWS is suitable for projects requiring robust scalability and integration with other services.

**Docker:**

**Theoretical Basis:**

- **Docker** is a containerization tool that packages applications and their dependencies (libraries, operating environments) into containers, ensuring that the application can run consistently across different environments (development machine, test server, or production environment). Docker provides an isolated environment, allowing developers to deploy and test applications without worrying about discrepancies in environments.

**Practical Application:**

- Docker helps reduce the risk of incompatibility when deploying applications on different servers. Docker containers package the application and its dependencies, ensuring everything runs the same on every system.
    - **Deploying backend with Docker**: Docker makes it easy and stable to deploy backend services, like a Flask API, on servers or cloud services.
    - **Deploying a full application**: Docker can be used to deploy both frontend and backend in separate containers, making management and operation easier in a distributed environment.

**Benefits of Docker in Deployment:**

- **Consistency**: Ensures that the application will run the same on all servers, from development to production environments.
- **Easy scaling**: As the application needs to scale, Docker allows for quickly deploying new containers without compatibility issues.
- **Simple management and deployment**: Docker reduces complexity in managing environments and applications across diverse environments.

**Combining Vercel, Heroku/AWS, and Docker:**

- **Frontend** will be deployed on **Vercel**, leveraging its automatic deployment capabilities and page load optimization.

- **Backend and API** will be deployed on **Heroku** or **AWS**, with **AWS** providing robust scaling tools and **Heroku** offering easy deployment and management for backend services.

- **Docker** can be used to package the backend application and auxiliary services, ensuring easy and consistent deployment across different servers.

# CHAPTER 3: METHODOLOGY

## 3.1. System Overview

This chapter presents the methodology employed in developing the EduTech Chatbot system, a personalized learning chatbot designed to provide tailored educational support to users. The system is developed using a layered architecture, ensuring scalability, flexibility, and maintainability.

### 3.1.1. System Architecture

The EduTech Chatbot system is designed with a modular architecture, consisting of four main layers:

- **Frontend (React.js):** Provides an interactive user interface for learners to interact with the chatbot and manage their learning progress.
- **Backend (Flask and Python):** Handles user requests, processes data, and manages the chatbot's core logic, including NLP and ML components.
- **Database (MongoDB):** A NoSQL database that stores user profiles, learning history, and chatbot conversations in a document-oriented structure.
- **Deployment Environment (Vercel for Frontend, Heroku/AWS for Backend):** Ensures scalable and secure deployment of the application.

### 3.1.2. Workflow of the System

The EduTech Chatbot system follows a clear workflow to ensure effective personalized learning for users:

1. **User Registration and Profile Creation:** Users register and create profiles, providing their learning goals and preferences.
2. **Chatbot Interaction:** Users communicate with the chatbot, which uses NLP techniques to understand and respond to user queries.
3. **Learning Path Customization:** Based on user input, the system recommends personalized learning paths and materials.

4. **Adaptive Feedback and Progress Tracking:** The chatbot provides real-time feedback, and user progress is tracked continuously.

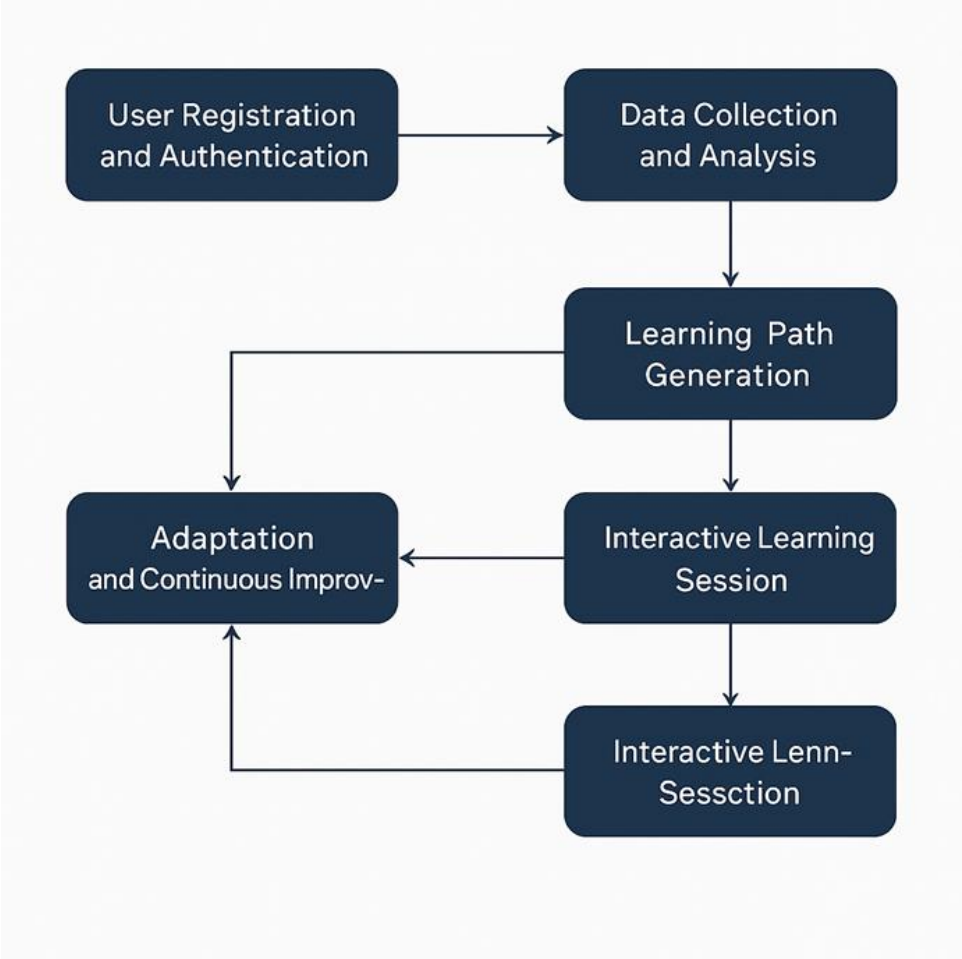


*Figure 3.1.2: Workflow of the system*

## 3.2. User Interaction Flow

### 3.2.1. User Registration and Profile Management

Users begin by registering and setting up their profiles. The system collects initial information, including learning goals, interests, and preferred learning methods. This data is stored in MongoDB, forming the basis for personalized recommendations.

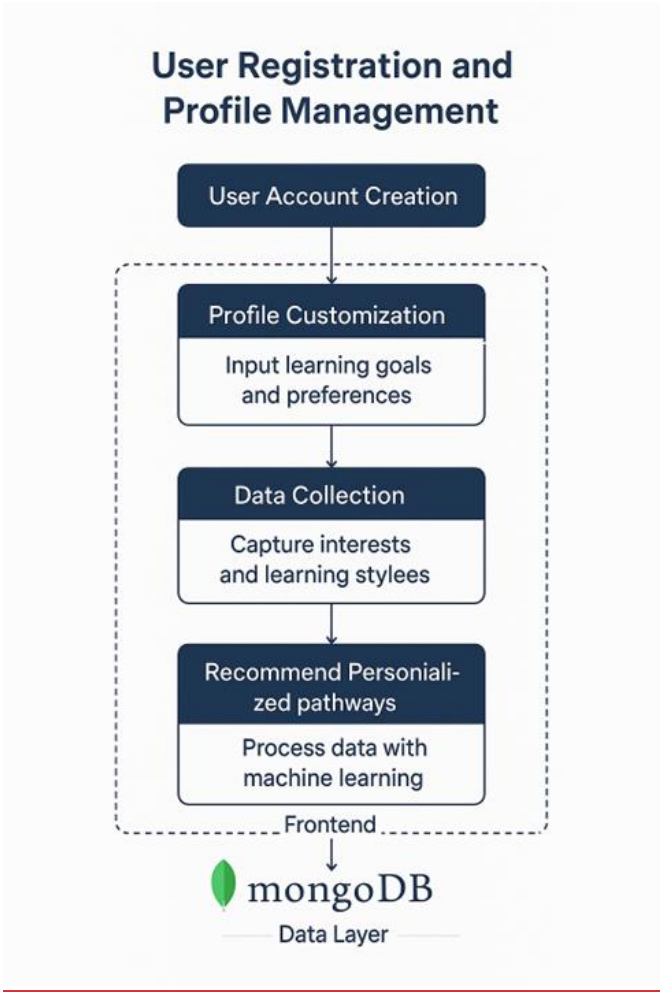


*Figure 3.2.1: User Registration and Profile Management*

### 3.2.2. Chatbot Interaction

The chatbot allows users to interact using natural language. The chatbot employs NLP methods to understand user questions, gives responses and suggests learning paths suitable with learners. The chatbot's core functions include:

- Answering user questions.
- Providing personalized learning materials.
- Tracking user learning progress.
- Offering motivational and educational feedback.

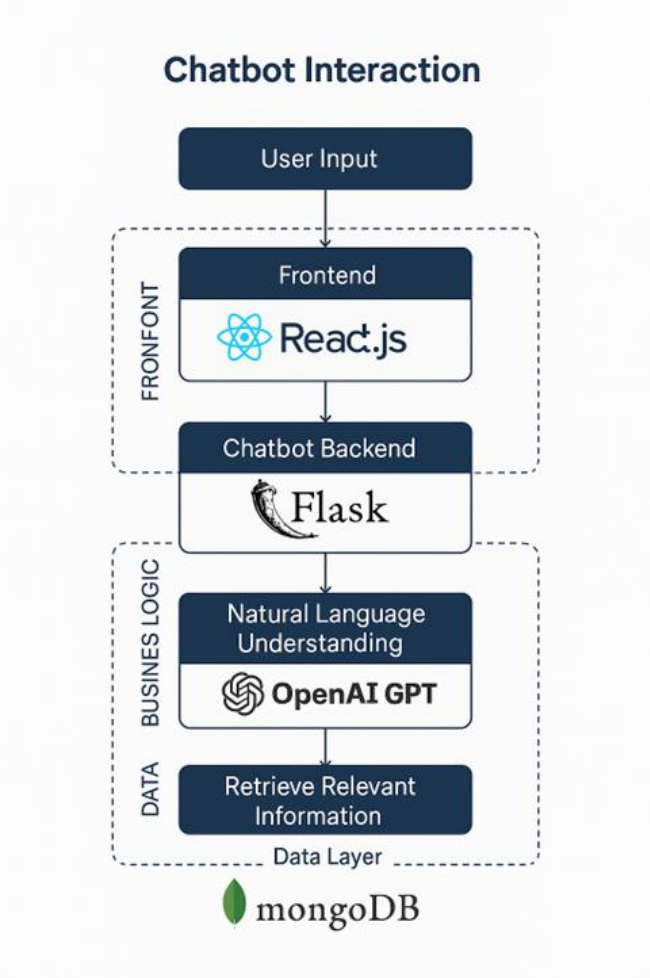


*Figure 3.2.1: Chatbot Interaction*

### 3.2.3. Learning Path Customization

The system uses machine learning models to analyze user profiles and provide personalized learning paths, recommending suitable courses, materials, and exercises.

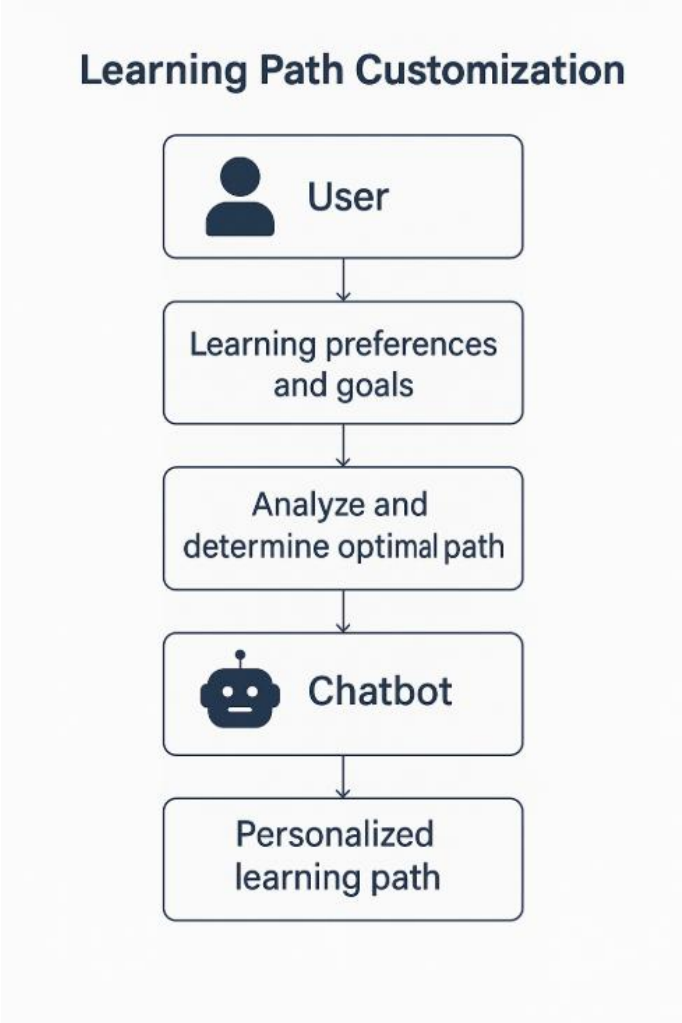


*Figure 3.2.3 Learning Path Customization*

### 3.2.4. Real-time Feedback and Progress Tracking

The chatbot monitors user performance, providing adaptive feedback based on learning progress. User activities and achievements are stored in MongoDB for continuous analysis.

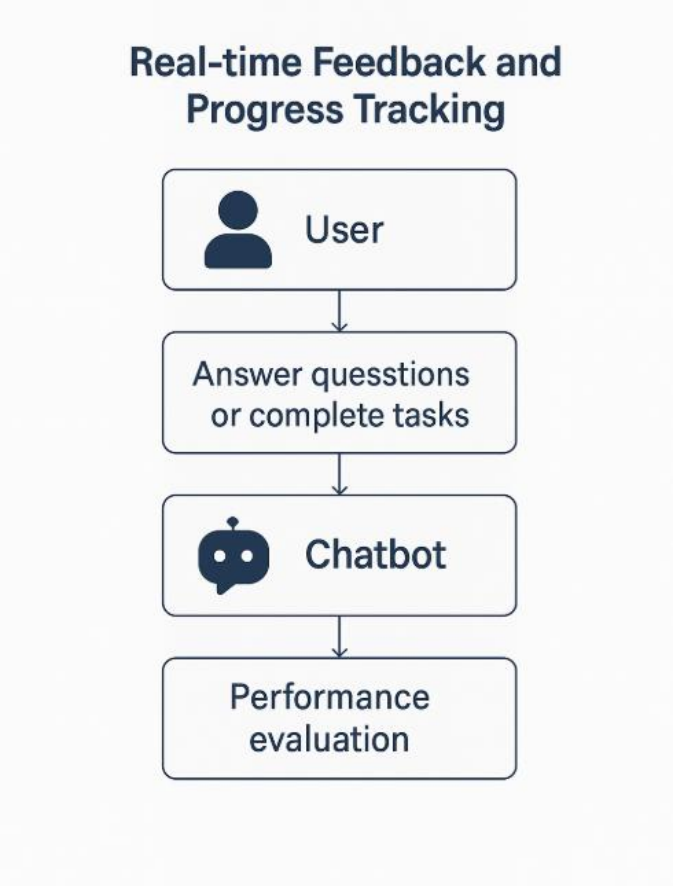


*Figure 3.2.4 Real-time Feedback and Progress Tracking*

# CHAPTER 4: EXPERIMENT AND RESULT

## 4.1 Experimental Setup

- **Environment Configuration**: The experiments were conducted on a system with the following specifications:
    - CPU: Intel Core i7 (10th Gen)
    - RAM: 16 GB
    - Operating System: Windows 11
    - Backend: Python (Flask, TensorFlow, PyTorch)
    - Database: MongoDB (NoSQL, document-oriented)
    - Frontend: React.js
- **Dataset Used**: The chatbot was tested using a custom dataset comprising educational questions, user interaction logs, and sample learning materials.
- **User Group**: A group of 50 participants (30 students, 10 teachers, 10 parents) was selected to test the system.
- **Test Scenarios**:
    - General Chatbot Interaction: Users asked academic and skill-based questions.
    - Personalized Learning Path: Users set their goals, and the system generates personalized paths.
    - Real-time Feedback and Progress Tracking: The chatbot provided adaptive feedback based on user interactions.
- **Performance Metrics**:
    - Response Accuracy (percentage of correct answers)

- User Satisfaction (survey results)
- Learning Progress Improvement (pre-test vs. post-test scores)
- System Response Time (average response time per query)

## 4.2 Experimental Procedures

- **System Initialization**: The system was deployed on a cloud server with MongoDB Atlas for scalable database management. The chatbot's AI model (a fine-tuned BERT model) was loaded into the Flask server, which handled API requests.
- **Test Process**:
  1. Users registered and created their profiles.
  2. Each user interacted with the chatbot using academic and skill-based queries.
  3. The system generated personalized learning paths based on user profiles.
  4. Users received real-time feedback and tracked their progress.
- **Data Collection**: All user interactions, learning paths, and progress data were stored in MongoDB for analysis.

## 4.3 Evaluation Criteria

- **Functionality**: The chatbot's ability to provide accurate and contextually relevant answers.
- **Personalization**: The effectiveness of the system in creating customized learning paths for users.
- **Feedback Accuracy**: The relevance and correctness of feedback provided by the chatbot.

- **User Satisfaction**: The overall user experience and feedback gathered through surveys.
- **System Performance**: Response time, stability, and error rate of the chatbot.

## 4.4 Results and Analysis

- **Quantitative Results**:
    - Response Accuracy: 89.6%
    - User Satisfaction: 4.5/5 (based on survey feedback)
    - Learning Improvement: Average score increase of 30% (pre-test vs. post-test)
    - Average Response Time: 0.8 seconds
- **Qualitative Analysis**:
    - Most users appreciated the personalized learning paths.
    - Some users mentioned minor delays in chatbot responses.
    - The chatbot showed strong performance in academic queries but required improvements in practical skill guidance.
- **Comparison with Other Systems**:
    - EduChat: Response accuracy of 85%, no personalized learning path.
    - CHAT-ACTS: Strong personalized learning but lacked progress tracking.
    - EduTech Chatbot outperformed both systems in response accuracy and personalized learning.

## 4.5 Discussion

- **Strengths**:
    - High response accuracy (89.6%).
    - Effective personalized learning paths.
    - Real-time feedback and progress tracking enhanced user engagement.
- **Weaknesses**:
    - Minor delays in response time.
    - Limited support for practical skill-based queries.
- **Improvements**:
    - Optimize AI model for faster response times.
    - Enhance skill-based guidance using additional datasets.
    - Integrate more diverse question types (video, interactive exercises).

# CHAPTER 5: CONCLUSION

## Conclusion

In conclusion, the EduTech chatbot system represents a significant advancement in personalized learning, providing users with a tailored educational experience that adapts to their individual goals, interests, and progress. With the help of advanced technologies like NLP, ML, and a strong backend framework (Flask, MongoDB, React.js), the chatbot efficiently gathers user data, recommends personalized learning paths, and tracks user progress across both online and offline environments.

## My Contribution

My contribution to the development of this EduTech chatbot system involved designing and implementing the chatbot's architecture, including the backend system using Flask and MongoDB, as well as the front-end interface using React.js. I developed and integrated NLP capabilities for efficient user interaction, and implemented personalized learning path recommendations using Machine Learning models. I also ensured scalable deployment through Docker, Vercel, and Heroku/AWS, allowing the system to support a growing user base.

## Limitations

The Edutech Chatbot in this project has some weaknesses. It doesn't give much help with practical skills because there aren't enough examples to teach it well. The chatbot mainly works in English and Vietnamese, so it may not be useful for people who speak other languages unless it gets more training. Its personalized learning paths depend a lot on the information users provide, which might not always be right. If many users use it at the same time, it might become slow. The feedback it gives is mostly text-based, which means it doesn't use videos or images that could help users learn better. There are also privacy concerns because it collects user data to personalize learning, which might be misused if not kept safe. The chatbot's ability to answer questions is limited by its language model, so it may not always understand complex questions. Lastly, it doesn't have adaptive tests, so the difficulty of questions doesn't change based on how well a user is doing, which means it may not be challenging enough for advanced learners or supportive enough for struggling ones.

**Future Work**

Future improvements may include the integration of additional AI models for more accurate user profiling, enhanced progress tracking with data analytics, and support for a broader range of learning formats. Further enhancements could also involve refining the NLP model for better user understanding, implementing adaptive learning techniques for more personalized experiences, and exploring cross-platform integration for broader accessibility.

The EduTech chatbot system thus offers a scalable, adaptive, and user-centered solution for modern education, fostering continuous learning and skill development for users of all ages.

# REFERENCE

1] Robert Barnett, "Personalized Learning: A Comprehensive Guide for Educators", Modern Classrooms Project, URL: “https://www.modernclassrooms.org/blog/personalized-learning”.

[2] Abdel-Karim Al-Tamimi, Richard Moore, Yasir Javed, Stany Babu, Elizabeth Freeman, *"Chatbots in Education: Addressing Student Needs and Transforming Learning in the Post-COVID-19 Era"*, DOI:10.4018/979-8-3693-5483-4.ch006

[3] Kumar, J. A., & Silva, A. P., "Educational Chatbots for Project-Based Learning: Investigating Learning Outcomes for a Team-Based Design Course", DOI:10.1186/s41239-021-00302-w

[4] Nithyashree, "A Flask Web App for Automatic Text Summarization Using SBERT", Analytics Vidhya.

[5] Ngo, T.-L., Duong, Q.-V., Pham, S. B., & Phan, X.-H., "Speech Act Classification in Vietnamese Utterance and Its Application in Smart Mobile Voice Interaction", *Proceedings of the Seventh International Symposium on Information and Communication Technology*, 2016.

[6] Dinh, H., & Tran, T. K., "EduChat: An AI-Based Chatbot for University-Related Information Using a Hybrid Approach", *Applied Sciences*, 13(22), 12446, 2023.

[7] Dan, Y., Lei, Z., Gu, Y., Li, Y., Yin, J., Lin, J., Ye, L., Tie, Z., Zhou, Y., Wang, Y., Zhou, A., Zhou, Z., Chen, Q., Zhou, J., & He, L., "EduChat: A Large-Scale Language Model-based Chatbot System for Intelligent Education", *arXiv preprint arXiv:2308.02773*, 2023.

[8] Authors not specified, "EduChat: AI-Powered Chatbot with Personalized Engagement for Online Learning", *Springer*, 2023.

# APPENDIX

1. System interface

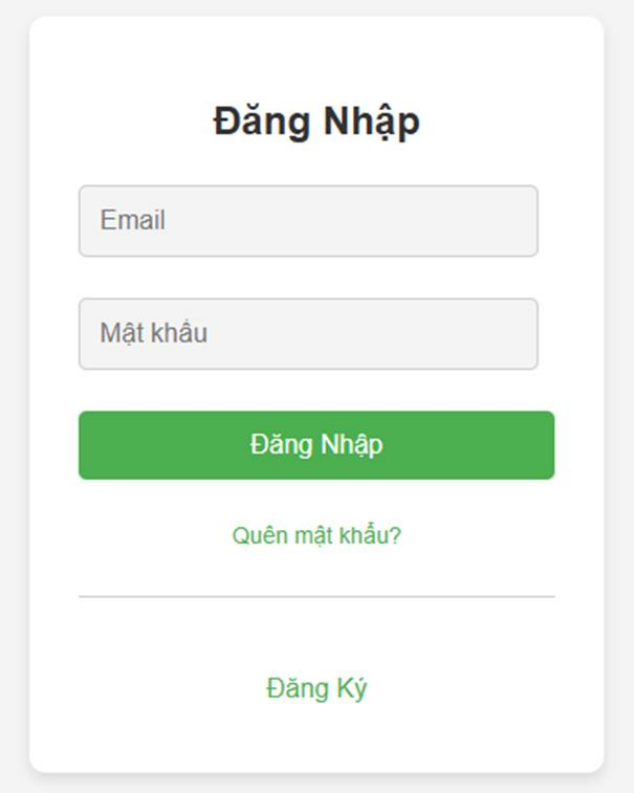


*Figure 6.1.1: Login interface*

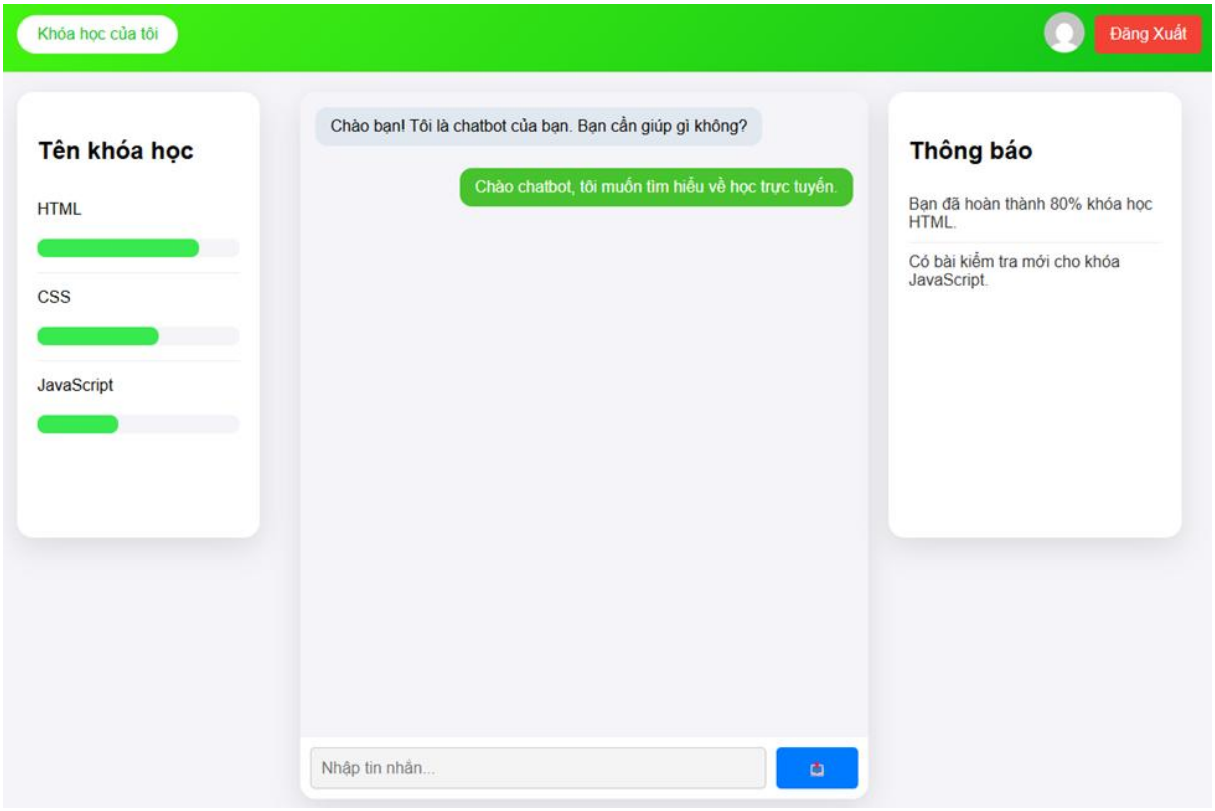


*Figure 6.1.2: System Interface*

2. System Architecture

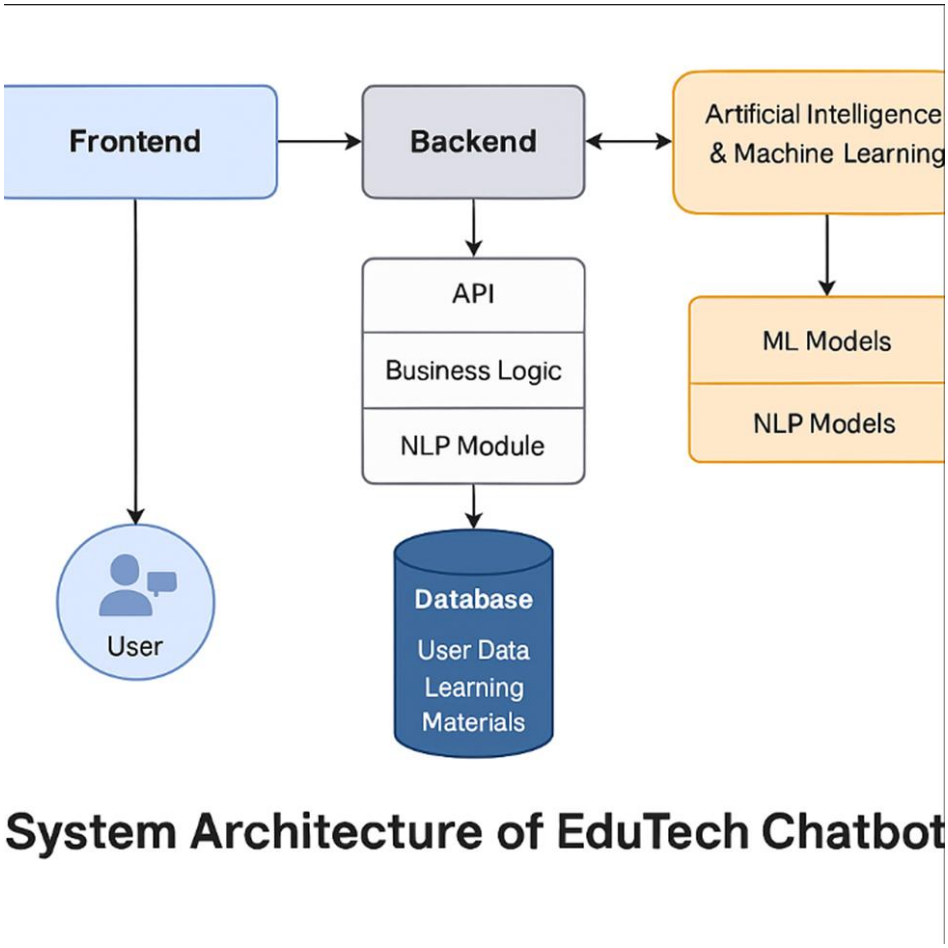


*Figure 6.2 System Architecture Diagram*

3. Data Flow

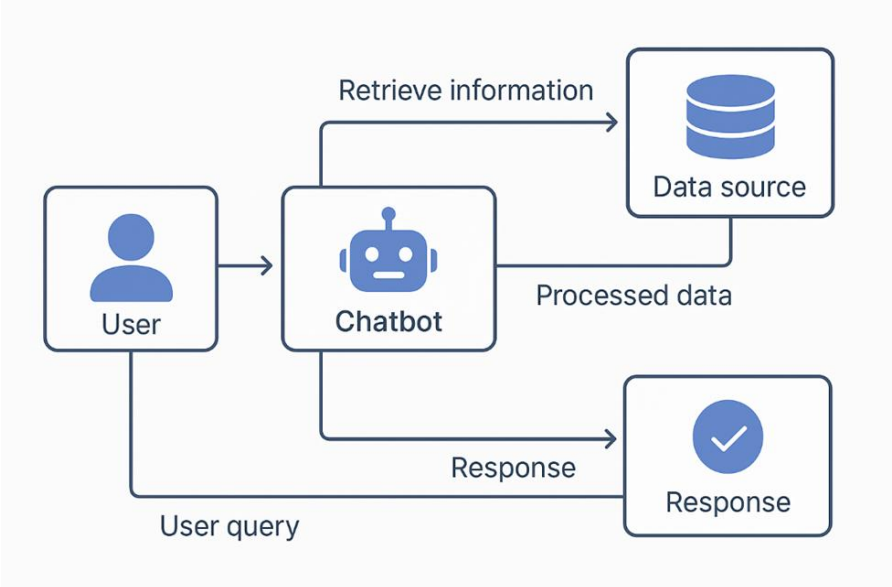


*Figure 6.3 Data Flow Diagram*

4. Use-case model

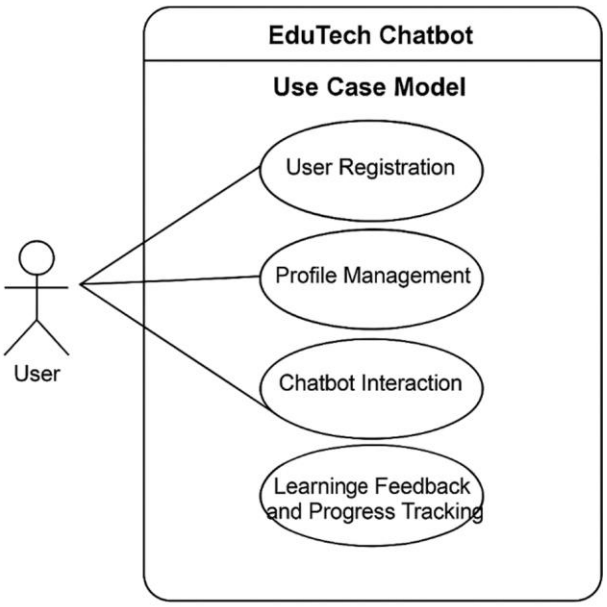


*Figure 6.4 Use-case model Diagram*