សកលវិទ្យាល័យ ប៊ែលធីអន្តរជាតិ

BELTIE INTERNATIONAL UNIVERSITY

គុណភាព ប្រសិទ្ធភាព ឧត្តមភាព សីលធម៌ គុណធម៌

Quality Efficiency Excelence Morality Virtue

មហាវទ្យាល័យ ព័ត៌មានវទ្យា និងវទ្យាសាស្ត្រ

Faculty of Information Technology & Science

មុខវិទ្យា ៖ SOFTWARE PROJECT MANAGEMENT

ប្រធានបទ ៖ AI ChatBot

សាស្ត្រាចារ្យ ៖ Chen Sovann

ជំនាញ ៖ Software Engineering (SE)

និស្សិតឆ្នាំទី ៖ ៤ ឆមាសទី ៖ ២ វេនសិក្សា៖ ល្ងាច

បន្ទប់លេខ៖ ២១៧ ក្រុម៖ ក្រុមទី ១

Group #1:

Seng Sokmeng

Douek Sothanroth

Neng Sereyrith

Yann Kosal

Lim Sopanha

Acknowledgements

We would like to acknowledge the individuals whose contributions and support have enabled us to successfully complete this research report.

First and foremost, we would like to express our sincere gratitude to **H.E. Ly Navuth**, President of **BELTEI International University**, for providing us with the valuable opportunity to pursue our studies at BELTEI International University under the **Faculty of Information Technology and Science**, majoring in **Software Engineering**.

Secondly, we would like to extend our heartfelt appreciation to **Mr. Oem Chanthorn**, Dean of the Faculty of Information Technology and Science, for granting us the opportunity to participate in this research program. His support has greatly contributed to enhancing our knowledge and strengthening our research skills, which are essential for academic and professional development.

We would also like to express our sincere thanks to **Vice Dean Kroeng Vannak** for his guidance, encouragement, and valuable advice throughout the research process, which played a significant role in the successful completion of this report.

Furthermore, we are deeply grateful to **Lecturer Chen Sovann** for his continuous guidance, constructive feedback, and mentorship. His support has been invaluable in shaping our understanding of **Management Information Systems and** , and in guiding us toward the successful completion of this research.

Finally, we would like to thank our families, classmates, and friends for their constant encouragement, support, and motivation, which have been a great source of strength throughout this academic journey.

Table of Content

A. Project Proposal

1. Introduction

With the rapid growth of web-based services, users increasingly expect instant responses and interactive communication on websites. However, many existing websites still rely on manual customer support or static FAQ pages, which are inefficient and unable to provide real-time assistance. This project proposes the development of a **web-based AI Chatbot** that can interact with users in real time, answer questions, and support multimedia inputs such as text, emojis, and images. The chatbot aims to enhance user experience by providing fast, automated, and intelligent responses through a modern and user-friendly interface.

2. Objectives

The main objectives of this project are:

- To design and develop a web-based chatbot interface using HTML, CSS, and JavaScript.
- To integrate an AI language model API to generate intelligent and contextual responses.
- To allow users to send text messages, emojis, and image attachments to the chatbot.
- To improve user interaction and engagement through real-time chat functionality.
- To demonstrate the practical application of AI-powered conversational systems in web development

3. Scope of the Project

Includes:

- Web-based chatbot user interface.
- Real-time messaging between user and chatbot.
- Integration with an AI API for automated responses.
- Support for emojis and image uploads.
- Responsive design for desktop and mobile devices.

Excludes:

- Mobile native applications (Android / iOS).
- User authentication and login system.
- Database storage of chat history (future enhancement).
- Voice input or speech recognition features.

4. Methodology

This project follows a **simple iterative development approach**, suitable for academic coursework. The development process includes planning the chatbot interface, implementing frontend components using HTML, CSS, and JavaScript, and integrating the AI API for response generation. Features are tested incrementally to ensure correct functionality. Minor improvements and bug fixes are applied throughout development based on testing results.

5. Expected Outcomes

Upon completion, the project will deliver a fully functional AI-powered chatbot that can be embedded into a website. The chatbot will be able to respond intelligently to user queries, handle multimedia inputs, and provide a smooth and interactive user experience. The project will also demonstrate the student's understanding of web development concepts and AI API integration.

6. References

- Google Generative Language API Documentation
- HTML, CSS, and JavaScript official documentation
- Emoji Mart Documentation
- Basic concepts of conversational AI systems

.

B. Software Requirement Specification (SRS)

1. Purpose

This Software Requirements Specification (SRS) document describes the functional and non-functional requirements of the **Web-Based AI Chatbot System**. The purpose of this document is to clearly define what the system will do, how it will perform, and the constraints under which it operates. This document serves as a reference for development, testing, and evaluation of the chatbot system.

2. System Overview

The AI Chatbot System is a web-based application that allows users to interact with an AI-powered chatbot through ~ chat interface. The system uses frontend web technologies and an external AI API to generate intelligent responses. Users can send text messages, emojis, and image files, while the chatbot provides automated replies in real time.

3. Functional Requirements

R1. User Interaction

- **R1.1:** The system shall allow users to open and close the chatbot interface.
- **R1.2:** The system shall allow users to send text messages to the chatbot.
- **R1.3:** The system shall display chatbot responses in real time.

R2. Multimedia Support

- **R2.1:** The system shall allow users to upload image files in the chat interface.
- **R2.2:** The system shall display uploaded images within the chat conversation.
- **R2.3:** The system shall allow users to insert emojis into messages.

R3. AI Response Generation

- **R3.1:** The system shall send user messages to the AI API for processing.
- **R3.2:** The system shall receive and display AI-generated responses.
- **R3.3:** The system shall maintain conversation context during a chat session.

R4. User Interface Management

- **R4.1:** The system shall display a typing indicator while the AI is generating a response.
- **R4.2:** The system shall automatically scroll to the latest message.

4. Non-Functional Requirement

Performance Requirements

- **NFR1:** The chatbot interface shall load within 2 seconds on a standard web browser.
- **NFR2:** AI responses should be displayed within a reasonable time depending on network conditions.

Usability Requirements

- **NFR3:** The user interface shall be intuitive and easy to use for first-time users.
- **NFR4:** The chatbot shall be accessible on both desktop and mobile devices.

Reliability Requirements

- **NFR5:** The system shall handle API errors gracefully and display error messages when necessary.

Security Requirements

- **NFR6:** The system shall not store user messages permanently.
- **NFR7:** Uploaded files shall only be used for generating AI responses during the session.

5. Use Case Descipstions

Use Case 1: Chat with AI Bot

- **Actor:** User
- **Precondition:** Chatbot interface is opened.
- **Main Flow:**

1. User types a message or uploads an image.
2. System sends input to the AI API.
3. AI generates a response.
4. System displays the response in the chat window.

- **Postcondition:** Conversation continues until the user closes the chatbot.

Use Case 2: Toggle Chatbot Interface

- **Actor:** User
- **Precondition:** Website is loaded.
- **Main Flow:**

1. User clicks the chatbot button.
2. Chatbot interface opens or closes.

- **Postcondition:** Chatbot state changes accordingly.

(Use Case Diagrams can be added to visually represent User and AI interactions.)

6. Assumptions and Constraints

- The system requires an active internet connection.
- The AI API must be available for response generation.
- The chatbot runs only in supported web browsers.

7. References

- IEEE 830 Software Requirements Specification Template
- Google Generative Language API Documentation
- Web Development Standards (HTML, CSS, JavaScript)

C. Project Plan

Purpose

The purpose of this project plan is to describe how the development work of the project is organized, managed, and tracked. It explains the major tasks involved in the project and provides a basic schedule showing when each task was carried out during the semester.

Work Breakdown Structure (WBS)

The project work is divided into the following main phases:

1.0 Planing

This phase focused on understanding the project requirements and defining the project scope. Activities included requirements gathering, topic approval, and selection of development tools and technologies such as programming language, framework, and libraries

2.0 Design

In this phase, the overall system design was prepared. This included designing the user interface layout, defining system flow, and preparing basic diagrams such as use case descriptions and system architecture.

3.0 Development

This phase involved implementing the actual system based on the design. Activities included frontend development, backend logic implementation, AI model integration, and connecting all components together to make the system functional.

4.0 Testing

The testing phase ensured that the system worked correctly. Unit testing and simple integration testing were performed to identify bugs and verify that features behaved as expected. Minor issues found during testing were fixed immediately.

Project Schedule

The project was developed as part of a class assignment during the **2025 academic year**. The schedule below summarizes the approximate timeline of each phase.

. **Planing:** September 2025

. **Design:** Late September – Early October 2025

. **Development:** October - November 2025

. **Testing & Refinement:** November – December 2025

A simple Gantt chart was created using spreadsheet software to visually represent the project timeline and task durations.

D. System Design Document (SDD)

Purpose

The System Design Document (SDD) describes the **overall architecture and technical design** of the AI Chatbot system. It explains how different system components interact with each other, how data flows through the system, and how the user interfaces with the chatbot. This document helps readers understand the internal structure of the system without going into detailed source code.

Key Section Required

. System Architecture

. Database Design

. Interface Design

System Architecture

The AI Chatbot system is designed using a **client–server architecture**. The user interacts with the chatbot through a simple chat interface, where text input is provided. The backend system processes the input and communicates with the AI model developed using **PyTorch**. The AI model analyzes the user message and generates an appropriate response, which is then sent back to the user interface.

The system is modular, meaning each component has a specific responsibility. This design improves maintainability and allows future improvements, such as enhancing the AI model or modifying the user interface, without affecting the entire system.

Database Design

The AI Chatbot does not rely on a traditional relational database like web-based management systems. Instead, it uses **training data files and model-related data** to operate.

The main data component include:

- **Training Dataset**: Contains predefined patterns, intents, and responses used to train the chatbot.
- **Model Files**: Store the trained neural network parameters generated using PyTorch.
- **Configuration Files**: Define model settings and chatbot behavior.

This design is suitable for an AI-based system where the focus is on learning from data rather than managing structured business records.

Interface Design

The chatbot interface is designed to be **simple and user-friendly** to ensure ease of use. The main interface components include:

- A **text input field** for users to enter messages.
- A **send button** to submit user input.
- A **chat display area** showing the conversation between the user and the chatbot.

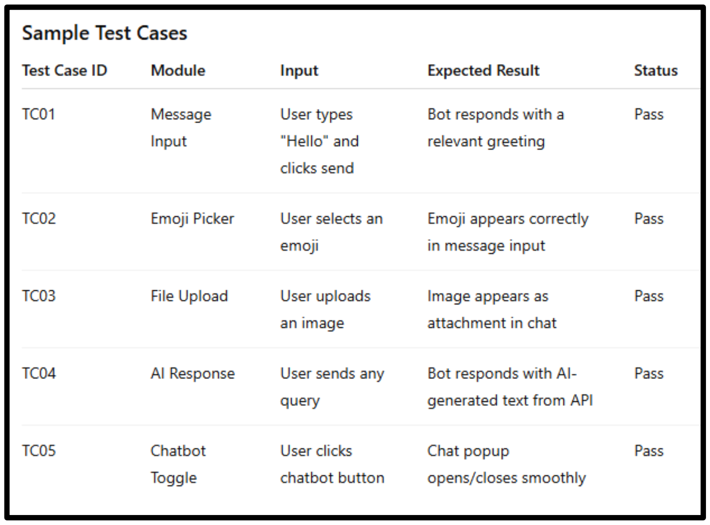
The interface allows users to interact naturally with the AI chatbot, making the system accessible even for users with minimal technical knowledge. Screenshots or basic wireframe designs can be included to visually demonstrate the interface layout

E. Test Plan and Test Summary Report

Purpose

To verify that the chatbot works correctly — sends messages, receives AI responses, handles file uploads, emojis, and displays correctly

Tools: Manual testing (for UI), Browser DevTools (console errors), and optionally automated tests with **Selenium** for UI flow



Sample Test Cases

Test Case ID	Module	Input	Expected Result	Status
TC01	Message Input	User types "Hello" and clicks send	Bot responds with a relevant greeting	Pass
TC02	Emoji Picker	User selects an emoji	Emoji appears correctly in message input	Pass
TC03	File Upload	User uploads an image	Image appears as attachment in chat	Pass
TC04	AI Response	User sends any query	Bot responds with AI-generated text from API	Pass
TC05	Chatbot Toggle	User clicks chatbot button	Chat popup opens/closes smoothly	Pass

- All main functionalities tested: **message sending, AI response, file upload, emoji, toggle, scrolling**.

- **No critical bugs found**; minor styling issues fixed immediately.

- Chatbot works on **desktop and mobile** browsers.

- Confirms that the system meets functional and non-functional requirements (response time, UI display, interactions).

F. Project Closing Report

Project Overview:

The AI Chatbot project was successfully completed on time, delivering a web-based chatbot that can send and receive messages, handle emojis, attachments, and respond using the AI API. The system meets the defined functional and non-functional requirements and works on both desktop and mobile browsers

Planned vs. Actual:

- **Time:** Planned 3 months → Actual 3 months

- **Budget:** Planned $0 → Actual $0 (using free tools and APIs)

- **Scope:** All planned modules (message input, AI response, emoji picker, file upload, chat toggle) were implemented.

Lessons Learned:

- Handling file uploads and base64 encoding was more complex than initially expected.

- Integrating with the AI API required careful management of chat history for context.

- Mobile responsiveness and emoji picker behavior needed extra adjustments for smooth UX.

Future Recommendations:

- Add **Dark Mode** for user interface.
- Include **chat history saving** so users can resume conversations.
- Implement **user authentication** for personalized chatbot experience.

- Explore **voice input/output** for a more interactive experience.

G. User Manual/Guide

1. Getting Started

To run the AI Chatbot project, users should ensure they have a modern web browser and an active internet connection. The project can be accessed by cloning the GitHub repository or downloading the source files. Once the files are available locally, open the index.html file in the browser to launch the chatbot interface. The chatbot utilizes the Google Gemini API for AI responses, and all necessary JavaScript and CSS files are included in the repository. Users can interact with the chatbot by typing messages in the input box, uploading image files if needed, or using emojis. The system automatically handles user input and generates AI responses, providing a fully interactive chat experience.

2. Feature & How to use them

The AI Chatbot project provides an interactive chat experience with two main modes of interaction:

1. **User Mode:** Users can type messages into the chat input to communicate with the AI. They can also enhance their messages with emojis or upload images, which the system can process as part of the conversation.
2. **AI Response Mode:** The AI agent generates real-time responses using the Google Gemini API. The system continuously updates the chat window with the AI's replies, maintaining the conversation context and delivering meaningful interactions.

The interface is designed to be intuitive, automatically scrolling to display the latest messages and resetting attachments or input fields after each submission. Users can open or close the chatbot popup at any time, and all chat interactions are stored temporarily for session-based context management.

3. Troubleshooting / FAQ

Q1: The chatbot doesn't respond when I send a message.
A: Make sure your internet connection is stable. The AI relies on the Google Gemini API, so an active connection is required. Also, ensure the API key is correctly set in the `script.js` file.

Q2: My uploaded image is not showing.
A: Check that the file type is supported (images only: `.jpg`, `.png`, `.gif`). Large files may take longer to load; try resizing or compressing the image.

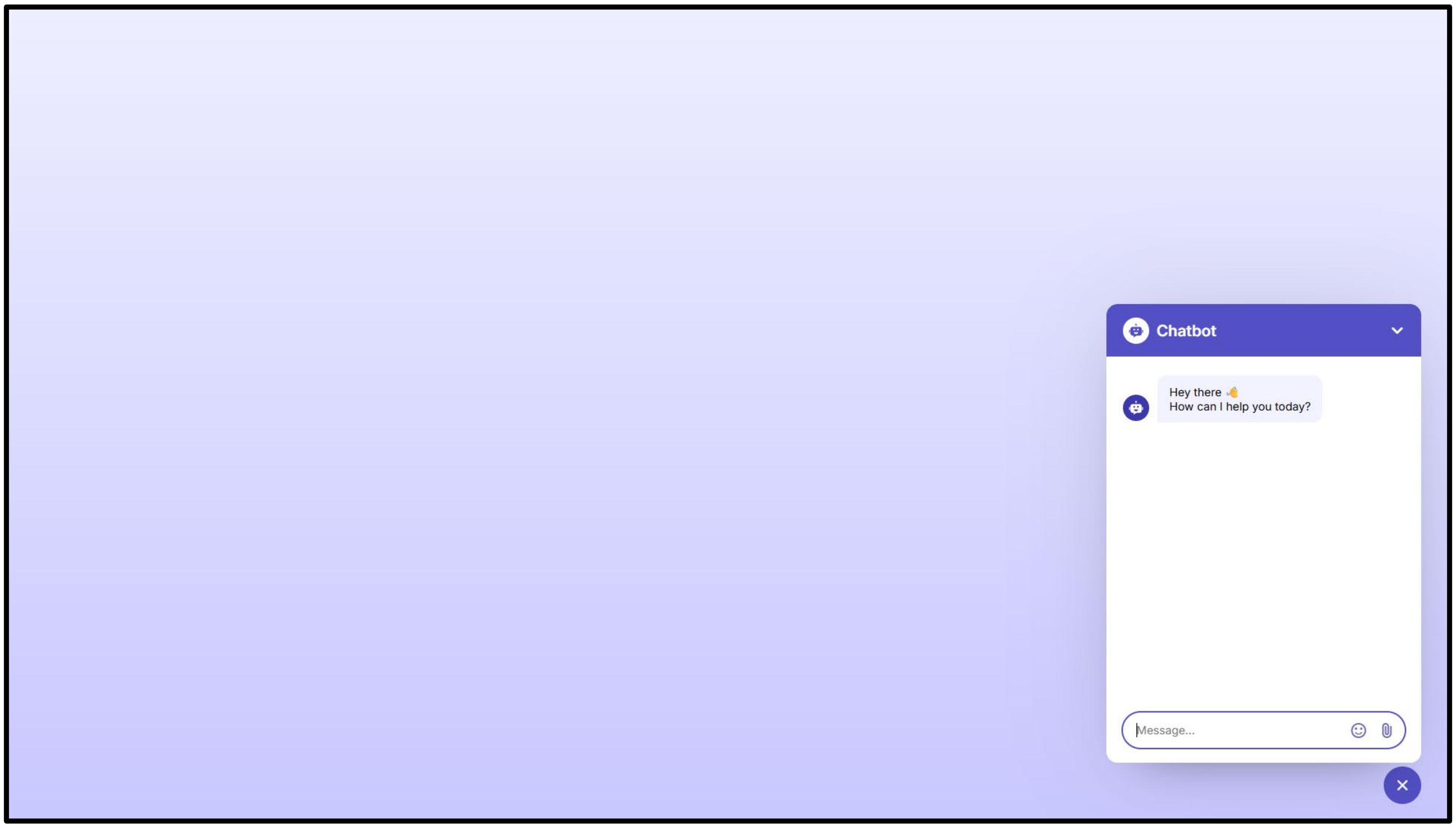
Q3: Emojis are not appearing in the message.
A: Ensure the emoji picker is visible by clicking the smiley icon. If it still doesn’t work, refresh the page to reload the emoji library.

Q4: Chatbot popup won’t open or close.
A: Confirm that JavaScript is enabled in your browser. If the popup still doesn’t respond, try clearing your browser cache or using a different browser.

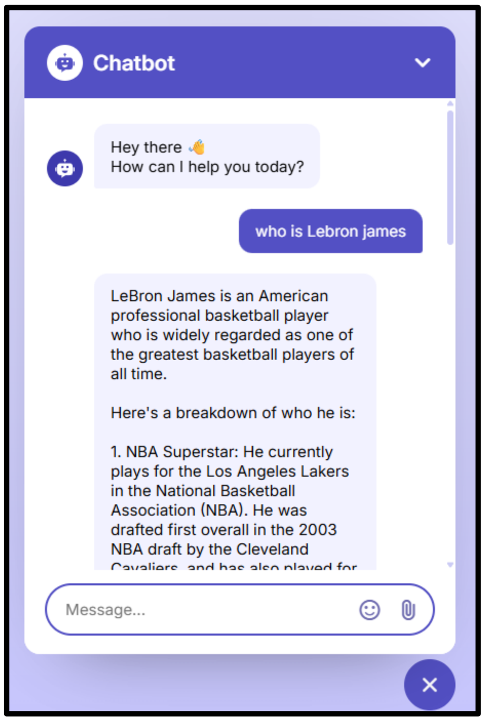
Q5: The AI gives incorrect or irrelevant answers.
A: The AI responds based on session context and the API model. If responses are inconsistent, try rephrasing your question or starting a new session by closing and reopening the chatbot.

Q6: I see error messages in red text.
A: These indicate a problem with the API request or connectivity. Check your API key, internet connection, and that the API quota has not been exceeded.

4 Visual / Screenshots



Hey there 👋
How can I help you today?



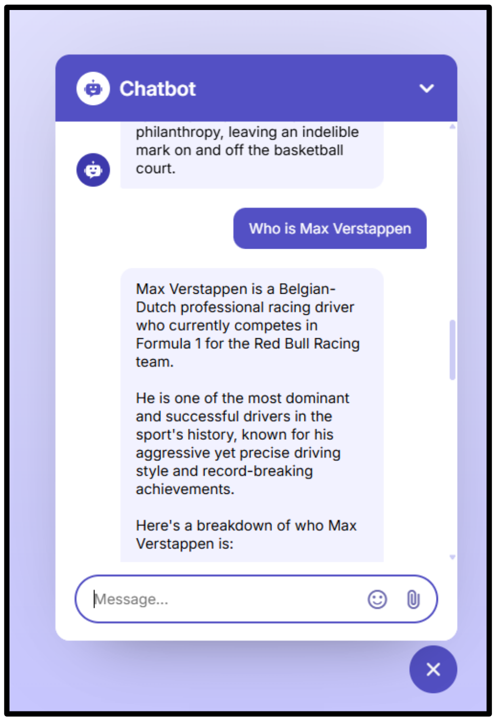
Hey there 👋
How can I help you today?

who is Lebron james

LeBron James is an American professional basketball player who is widely regarded as one of the greatest basketball players of all time.

Here's a breakdown of who he is:

1. NBA Superstar: He currently plays for the Los Angeles Lakers in the National Basketball Association (NBA). He was drafted first overall in the 2003 NBA draft by the Cleveland Cavaliers, and has also played for



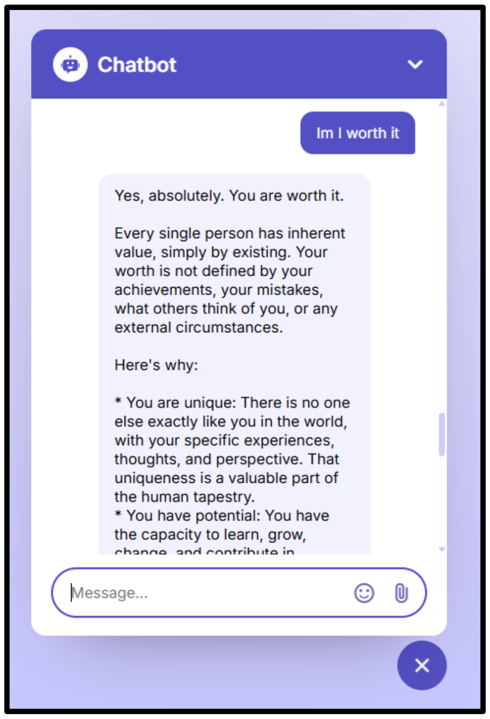
philanthropy, leaving an indelible mark on and off the basketball court.

Who is Max Verstappen

Max Verstappen is a Belgian-Dutch professional racing driver who currently competes in Formula 1 for the Red Bull Racing team.

He is one of the most dominant and successful drivers in the sport's history, known for his aggressive yet precise driving style and record-breaking achievements.

Here's a breakdown of who Max Verstappen is:



Im I worth it

Yes, absolutely. You are worth it.

Every single person has inherent value, simply by existing. Your worth is not defined by your achievements, your mistakes, what others think of you, or any external circumstances.

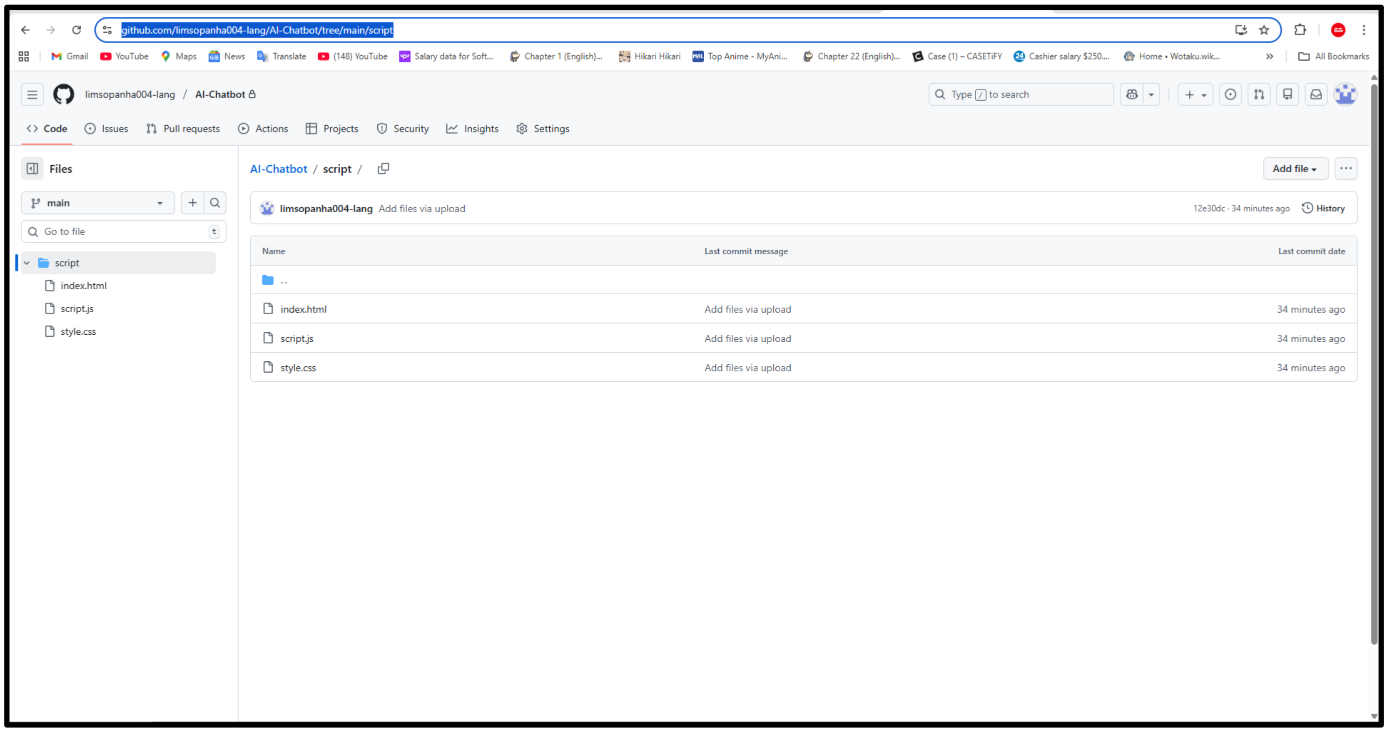
Here's why:

* You are unique: There is no one else exactly like you in the world, with your specific experiences, thoughts, and perspective. That uniqueness is a valuable part of the human tapestry.
* You have potential: You have the capacity to learn, grow, change, and contribute in

H. Code and Software

1. Link and Source Code Repository

https://github.com/limsopanha004-lang/AI-Chatbot/tree/main/script



limsopanha004-lang Add files via upload 12e30dc · 34 minutes ago

Name	Last commit message	Last commit date
..		
index.html	Add files via upload	34 minutes ago
script.js	Add files via upload	34 minutes ago
style.css	Add files via upload	34 minutes ago

The repository is organized to ensure clarity and ease of use:

- `src/` → Contains all frontend files (`index.html`, `style.css`, `script.js`).
- `docs/` → Contains project documents, diagrams, and this report.
- `README.md` → Provides detailed instructions on how to run the chatbot locally, project overview, features, and screenshots.

2. Final Software Application

The AI Chatbot is a **web-based application** and can be accessed live at:
[Live Demo URL]

Users can interact with the chatbot directly from their browser. For offline use, the project can be downloaded from the GitHub repository and run locally by opening `index.html` in a web browser. No additional installation is required other than having an internet connection for API calls.

Dependencies and Libraries Used:

- EmojiMart → For emoji picker integration.
- JavaScript `fetch()` API → For connecting with AI API and generating responses.
- HTML & CSS → For user interface design.