មហាវិទ្យាល័យ ព័ត៌មានវិទ្យា និងវិទ្យាសាស្ត្រ
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCE

Subject: Software Project Management
by Mr. CHEN Sovann

Topic: Web Based AI-ChatBot

Our Team:

1. Seng Sokmeng
2. Yann Kosal
3. Lim Sok Panha
4. Neng Serey Rith
5. Doeuk Sothanroth

Agenda

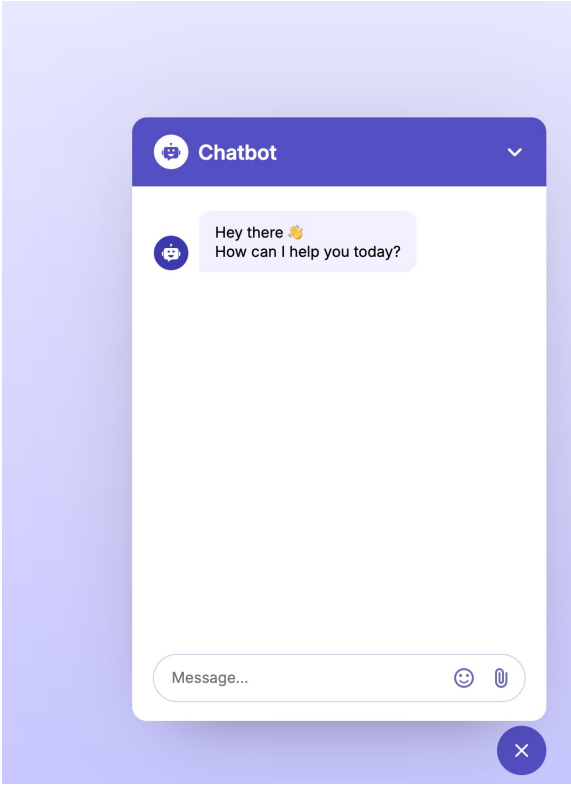
- Problem Statement
- Proposed Solution
- Key Features
- Technology Stack
- System Architecture
- Challenges & Solutions
- Testing Results
- Conclusion & Future

Problem Statement

- **Current Limitations:** Many existing websites rely on inefficient manual customer support or static FAQ pages.
- **User Frustration:** These methods fail to provide real-time assistance, leading to delayed responses and a poor user experience.
- **The Need:** Users increasingly expect instant responses and interactive communication when visiting web-based services.

Proposed Solution

- **AI Chatbot:** Development of a web-based AI Chatbot capable of real-time interaction.
- **Automation:** Integrates an AI language model API to generate intelligent, contextual, and automated responses.
- **Enhanced Experience:** Aims to provide a modern, user-friendly interface that supports multimedia inputs like text, emojis, and images.



Key Features

- **Real-Time Messaging:** Users can send messages and receive instant automated replies.
- **Multimedia Support:**
- **File Uploads:** Users can upload image files for the AI to analyze.
- **Emojis:** Integrated emoji picker to enhance expression.
- **Contextual Awareness:** The system maintains conversation context within a session for more natural interactions.
- **User Interface:** Features a typing indicator, auto-scrolling to the latest message, and a toggleable chat window.

Technology Stack

- **Frontend:**

- HTML & CSS: Used for structuring and designing the user interface.

- JavaScript: Handles client-side logic and API interactions.

- **AI Integration:**
- **Google Gemini API:** Utilized to generate intelligent AI responses.
- **Fetch API:** Used to establish connection and send requests to the AI service.
- **Libraries:**
- **Emoji Mart:** Integrated for emoji selection functionality.

System Architecture

- **Client-Server Model:** The system follows a modular client-server architecture.
  - **Client:** A web interface where users input text and files.
  - **Backend/API:** Processes input and communicates with the AI model to generate responses.
- **Data Flow:** User inputs are sent to the external API, which analyzes the data and returns a response to the UI.
- **Data Management:** The system relies on temporary session storage rather than a traditional relational database for chat history.

Challenges & Solutions

- **Challenge 1: File Handling:** processing file uploads and Base64 encoding was more complex than anticipated.
- **Challenge 2: Context Management:** Integrating the API while keeping the chat history relevant for context.
- **Challenge 3: UX/UI Issues:** Mobile responsiveness and emoji picker behavior required adjustments.

Testing Results

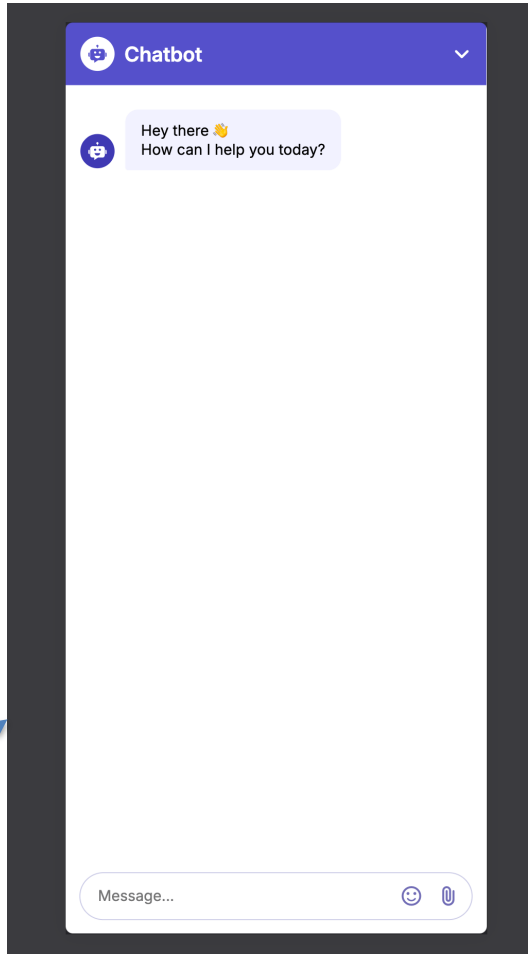
- **Methodology:** Manual testing of UI and functionality, plus browser console debugging.

Key Observations:

Success: All main functionalities (messaging, AI response, file upload, emojis) passed testing.

Performance: The chatbot loads within 2 seconds and works across desktop and mobile browsers.

Reliability: No critical bugs were found; minor styling issues were fixed immediately.



Responsive on mobile application

Conclusion & Future

- **Conclusion:** The project successfully delivered a fully functional, web-based AI chatbot that meets all defined functional requirements within the planned timeline and budget.
- **Future Recommendations:**
- **Dark Mode:** Implement a dark theme for better user accessibility.
- **History Storage:** Add database support to save chat history permanently.
- **Authentication:** specialized user logins for personalized experiences.
- **Voice Interaction:** Explore voice input/output features.