Context

Since the arrival of the green invader, new pathogenic species have exploded. The medical teams, for what remains, are overwhelmed and not enough to meet the demand. To reduce the workload on their shoulders, you'll need to create a simple chatbot aimed at identifying health issues based on symptoms. This automated triage will make it possible to redirect patients to the right resources and thus avoid overloading an already strained system.

Objective

Your challenge is to create a simple chatbot that allows you to make a preliminary triage of potential future patients according to their symptoms. The user must collect information related to the user's condition, use it to question an AI model and provide a potential diagnosis as well as recommend actions to be taken if necessary.

You will be evaluated on your tool's ability to guide the user, question them relevantly, make a diagnosis and communicate it to them.

Scenario

The user is experiencing symptoms that they believe are due to a respiratory infection. Unsure of the seriousness of the situation, he uses the tool to determine the severity of his symptoms and the actions to take.

Parameters

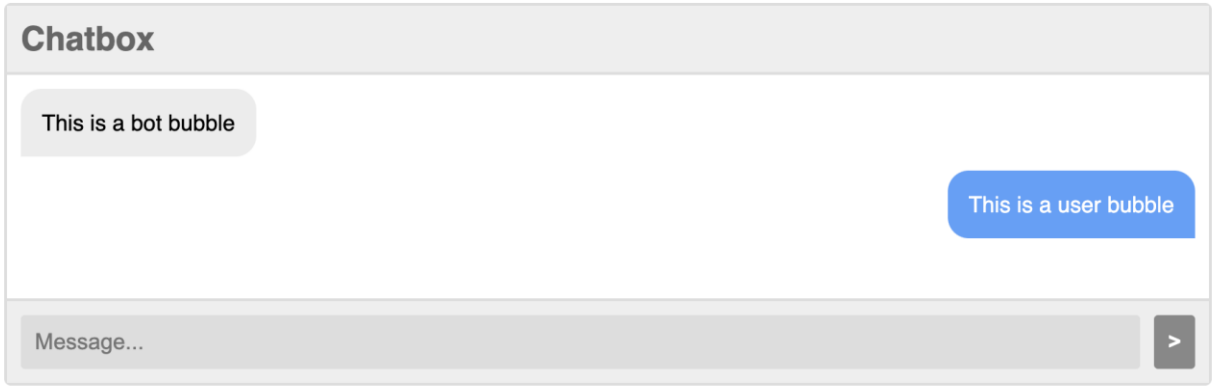
Interface

The chat interface is made in HTML/JavaScript.

Seed files are provided;

- Index.html contains the chat DOM
- Style.css set the chat UI
- Index.js contains the basic features of the chat

The visual and basic functionality of the chat are included in the seed files.



Refer to MDN Web Docs if you are not familiar with these technologies.

Model
The model used for the challenge is medalpaca-7b.

The model is pre-trained on medical data.

The model understands natural language.

The model is hosted in a cloud and does not have to be installed on-premises.

Code snippets to make calls to the template's API are provided in the query.md file.

Refer to the documentation to choose and configure your settings.

Chatbot
The bot is exchanging in natural language with the user via chat.

The bot is exchanging with the model via the API.

The bot's logic can be done in the language of your choice, but we recommend JavaScript or Python.

Restrictions
The use of any other model is prohibited.

The use of generation tools (chatGPT, Copilot, etc.) is prohibited.

Only the web pages listed in this document can be accessed during the Challenge.

Expectations
The bot should...

- Provide the context of the tool
- Guide the user through the use
- Gather the necessary information
- Questioning the AI model

The user should understand that ...

- The exchange is done with a bot and not a human
- The context is medical
- The information obtained is used to perform a triage
- The diagnosis is for illustrative purposes only (unofficial)

The participant must ...

- Complete the chat code (index.js) to allow the exchange between the bot and the user
- Program the bot's logic to harvest the user's information
- Query the model for diagnosis and triage
- Explain the process for establishing the information to be collected

Strategy
You will have to do *prompt engineering* to adjust the bot's exchanges with the user to obtain the best possible results.

All exchanges should be made in natural language.

Prompt *engineering* is the process by which you guide a generative artificial intelligence (*AI)* solution to generate the desired outcomes.

Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken and written.

Considerations

Examples of information to be collected:

- List of symptoms
- How long have the symptoms been present?
- Presence of high fever

Examples of actions to take:

- Using over-the-counter medication to help with symptoms
- Consult a professional if symptoms worsen
- Presenting to the emergency room

Examples of experiments:

- Questioning the model in different ways on the same topic
- Query the model on a topic to get the details
- Test different parameter configurations

Submission

You must submit before the end of the challenge:

- Your project's code
  - Create a public Git repo and send us the link
- A detailed README explaining how to execute the project
  - The project must be able to be launched
- A report detailing your prompt engineering process
  - The strategy(s) used
  - Configured settings
  - The Experimentation Process
  - Any other tests or reflections leading to refinement of the exchange
- Two different scenarios to test
  - Describe the exchange to be carried out
  - Specify the expected result
  - One scenario should lead to consulting a professional and the other should not

Score

The user experience, the report and the logic of the bot will be evaluated.

Here are the different evaluation criteria:

- Features and Experience (UX): 300 points

- Approach and experiments (prompt engineering): 400 points
- Relevance, Ability and Appropriateness (code): 700 points
- Score maximum possible: 1400 points