

نظم المتنبي

“The Poetry of Al-Mutanabbi “

**Group members**

| Email | Name |
| --- | --- |
| [aomaghraby@gmail.com](mailto:aomaghraby@gmail.com) | Ashwag Oamr maghraby |
| [azzahalhashmi@hotmail.com](mailto:azzahalhashmi@hotmail.com) | Azzah mohammad mahjoob |
| [shahadkulaibi@gmail.com](mailto:shahadkulaibi@gmail.com) | Shahad ahmed kulaibi |
| [raghadalshmrani3@gmail.com](mailto:raghadalshmrani3@gmail.com) | Raghad abdullah alshmrani |

**Application Overview**

The application integrates artificial intelligence with the art of Arabic poetry, specifically emulating the style of the prominent Arab poet Al-Mutanabbi. It leverages generative AI by utilizing large language models (LLMs) to create poetic verses that closely align with the creativity of Al-Mutanabbi, showcasing an advanced interaction between art and technology.

**Primary Features of the Application**

The application performs three main tasks, all of which are designed to reflect Al-Mutanabbi's unique poetic style:

1. **Completing Poetic Verses**:

The application analyzes the first hemistich of a verse provided by the user and, using linguistic and stylistic models inspired by Al-Mutanabbi’s divan, generates the second hemistich in a manner that harmonizes with the original style.

1. **Answering Questions with Poetic Verses**:

The application interprets the user’s colloquial questions and responds with poetic verses crafted in Al-Mutanabbi’s style.

1. **Writing poetic verses on a selected topic**:

Based on a word or topic provided by the user, the application generates original poetic verses in the style and manner of Al-Mutanabbi.

**Prompt Engineering**

The application utilizes prompt engineering to guide the ALLaM model in generating responses that accurately reflect Al-Mutanabbi's style. Instead of altering the model’s internal weights or structure, carefully crafted prompts are used to optimize the model’s output for each task.

**How Prompt Engineering is Applied:**

1. **Task-Specific Prompts**:

For each of the three main tasks, specific prompts are designed to steer the model in generating the desired output. These prompts help the model understand the context and structure expected in the responses.

1. **Iterative Refinement**:

By refining the prompts through iterative testing, the application ensures that the model produces accurate, coherent, and stylistically appropriate poetic verses.

1. **Contextual Cues**:

Prompts are augmented with **contextual cues** derived from Al-Mutanabbi’s divan, ensuring that the generated responses closely mimic his style, structure, and linguistic patterns.

**Evaluation and Testing Framework**

To ensure the model's adherence to Al-Mutanabbi's poetic style and the quality of generated poetry, a comprehensive evaluation and testing framework was implemented.

**1. Human Evaluation:**

* **Google Form Surveys:** Users were presented with app generated poems and asked to rate their quality (style, meaning, and meter) and overall satisfaction using a Likert scale. (see Appendix 1)
* **Expert Review:** Poetry experts with a deep understanding of Al-Mutanabbi's work evaluated the generated poems for their adherence to the poet's stylistic elements, themes, and imagery.

**2-User Usability:**

* **Google Form Surveys:** Users were presented with app generated poems and asked to rate their sability, design, content variety and overall satisfaction using a Likert scale. (see Appendix 2)
* **User Review:** Poetry lovers assessed the app and their overall satisfaction through a Likert scale survey. Participants rated aspects such as usability, design, content variety

**Result discussion**

**1.Human Evaluation:**

The results from the expert reviews and poetry lovers provided valuable insights into the model's performance. Users generally expressed satisfaction with the generated poetry's quality, style, and meaning.

## Why Choose Our Solution?

Our application stands out for several reasons:

1. **Unique Concept**:

We combine Arabic poetry with personal imitation, using advanced AI to emulate the style of one of the greatest Arab poets, Al-Mutanabbi.

1. **Cultural Relevance**:

The year 2023 was designated as the "Year of Arabic Poetry" to revive poetry in Arab culture. Our application, particularly the feature of responding with poetic verses, can be linked to social media platforms to encourage young people to engage with poetry daily.

1. **Creative Tool for Poets**:

The application can serve as a valuable tool for poets, offering inspiration and creative assistance in the process of composing poetry.

## Target Audience

Our application is designed to appeal to a wide range of users, including:

1. **Poetry and Culture Enthusiasts**:

Individuals who appreciate Arabic poetry and its cultural significance.

1. **Beginning Writers and Poets**:

Aspiring poets who are looking for inspiration and guidance in crafting their own verses.

1. **Those Seeking Poetic Expression**:

Anyone interested in expressing their thoughts or emotions through poetic language.

**How Our Solution Supports the ALLaM Model**

The ALLaM model can be enhanced and supported by our application in the following ways:

**1. Prompt Engineering for Specific Tasks**

By applying prompt engineering techniques, we can optimize the ALLaM model’s performance for generating poetic verses and customized responses. These tailored prompts help the model better understand the specific requirements of each task, resulting in more accurate and stylistically appropriate outputs.

**2. Integrated User Feedback**

Our application will implement mechanisms to capture user feedback on the generated poetry and customized responses. This feedback will be used to further adjust and enhance the ALLaM model, bringing its performance closer to user preferences and expectations.

**Technologies Used for the Project**

**1.User Interface Design:**

* **Figma:** A collaborative design platform that enables designers to create intuitive and visually appealing user interfaces, ensuring a positive user experience.

**2.Prompt Engineering:**

* **WatsonX:** A cloud-based AI platform from IBM that offers tools for prompt engineering, helping to optimize the model's performance and generate more relevant and informative responses.

**3.Documentation and Presentation Tools:**

* **Microsoft PowerPoint:** A versatile presentation software for creating professional presentations to communicate project goals, progress, and outcomes.
* **Animaker:** An online animation tool that allows users to create engaging and informative animations to visually represent project concepts and processes.

**User Interfaces**

Application were divided into 4 interfaces: one for the home page and three for each task.

**Home Page**

The home page, as shown in figure 1, serves as the entry point for the application. It features a visually appealing logo and a concise title that accurately reflects the app's purpose. A brief description outlines the key functionalities: completing poetic verses, answering questions with poetry, and writing original verses.





Figure 1: Home page

**Completing Poetic Verses**

This interface allows users to input the first hemistich of a verse. Upon clicking the "Send"/"أرسل" button, the application utilizes linguistic and stylistic models inspired by Al-Mutanabbi's divan to produce a harmonious second hemistich. The completed verse is then displayed to the user. Figure 2 shows a visual sample of an interaction performed by a user:

**User Interaction Example:**

1. **User:** Opens the application and clicks the "Complete Verse" button.
2. **Interface:** Displays an input field for the first hemistich.
3. **User:** Enters the first hemistich: "كنا وصرنا فأين سنكون".
4. **User:** Clicks the "Send"/"أرسل" button.
5. **Interface:** Displays the completed verse: see Figure 2.







Figure 2: Completing Poetic Verses

**Answering Questions with Poetic Verses**

Users can enter a question in colloquial Arabic, and the application will generate a poetic response crafted in Al-Mutanabbi's style. This feature enables users to explore the application's ability to interpret and respond to prompts in a creative and poetic manner. Figure 3 shows a visual sample of an interaction performed by a user:

**User Interaction Example:**

1. **User:** Opens the application and clicks the "Answer Questions with Poetic Verses" button.
2. **Interface:** Displays an input field for the user to enter their question.
3. **User:** Enters the question: "ماهو الوفاء؟".
4. **User:** Clicks the "Send" or "أرسل" button.
5. **Interface:** Displays the generated poetic response: see Figure 3.

**Writing Poetic Verses on a Selected Topic**

This interface prompts users to enter a word or topic. The application then generates an original poetic verse in Al-Mutanabbi's style based on the provided input. This feature allows users to experiment with different themes and explore the application's versatility in creating diverse poetic content. Figure 4 shows a visual sample of an interaction performed by a user:

**User Interaction Example:**

1. **User:** Opens the application and clicks the "Write Poetic Verses on a Selected Topic" button.
2. **Interface:** Displays an input field for the user to enter a word or topic.
3. **User:** Enters the word "حب الوطن".
4. **User:** Clicks the "Send" or "أرسل" button.
5. **Interface:** Displays the generated poetic verse: see Figure 4.







Figure 3: Answering Questions with Poetic Verses





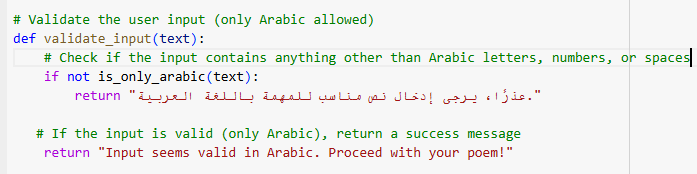
Figure 4: Writing Poetic Verses on a Selected Topic

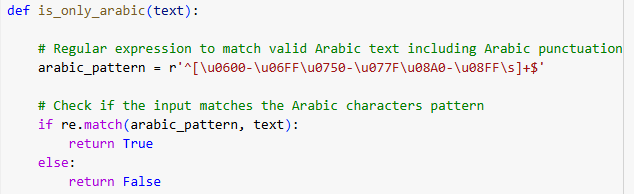
**Code Elements**

The project is implemented using FLASK API. It contains 4 HTML files. 4 JavaScript files, and 2 CSS files. It is hosted on the web using PythonAnyWhere.

* **Input Validation Check**

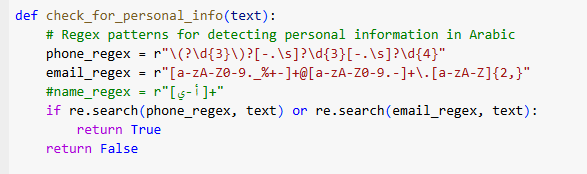
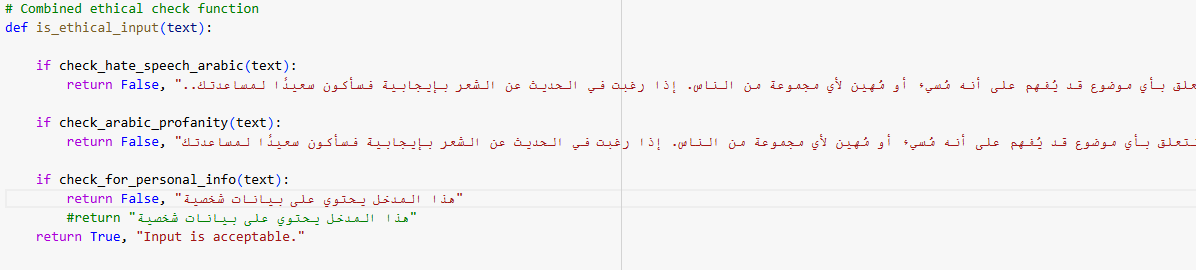
Check if the user input contains valid text before sending it to ALLAM**.**

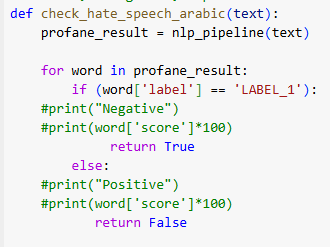
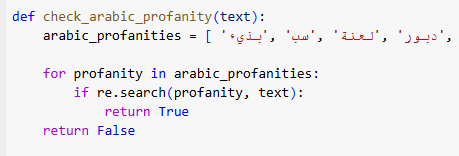
****

****

* **Ethical and responsible AI check**

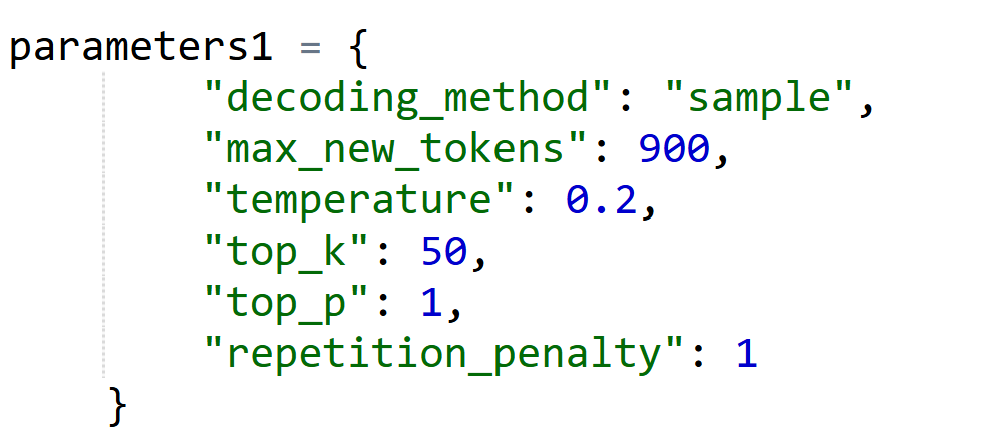
Check the input for personal info, profanity, and hate speech(using a trained model from hugging face).

****

****

* **Define ALLAM parameters**

Sample instead of greedy for creative tasks.



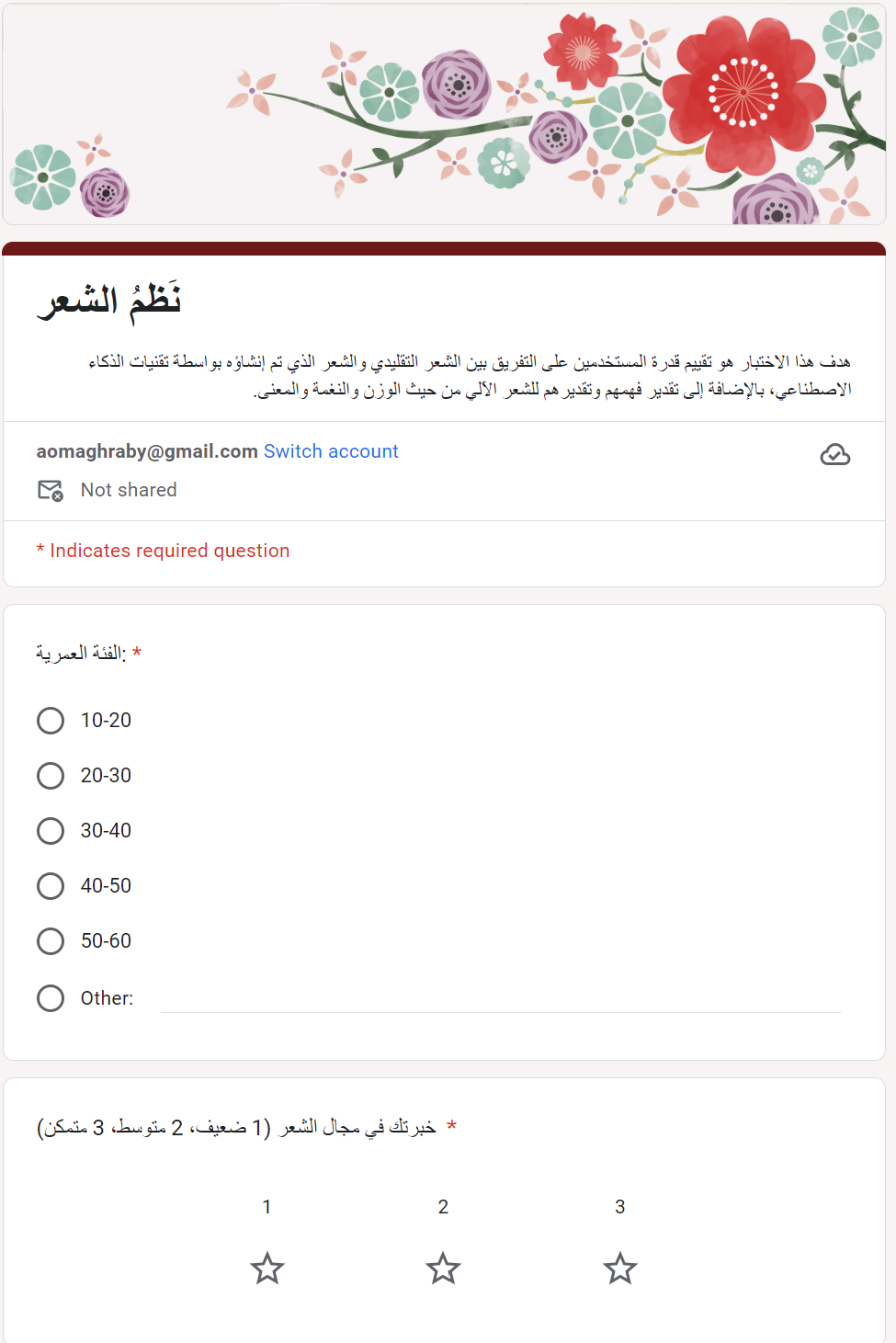
* **Maintain conversation**

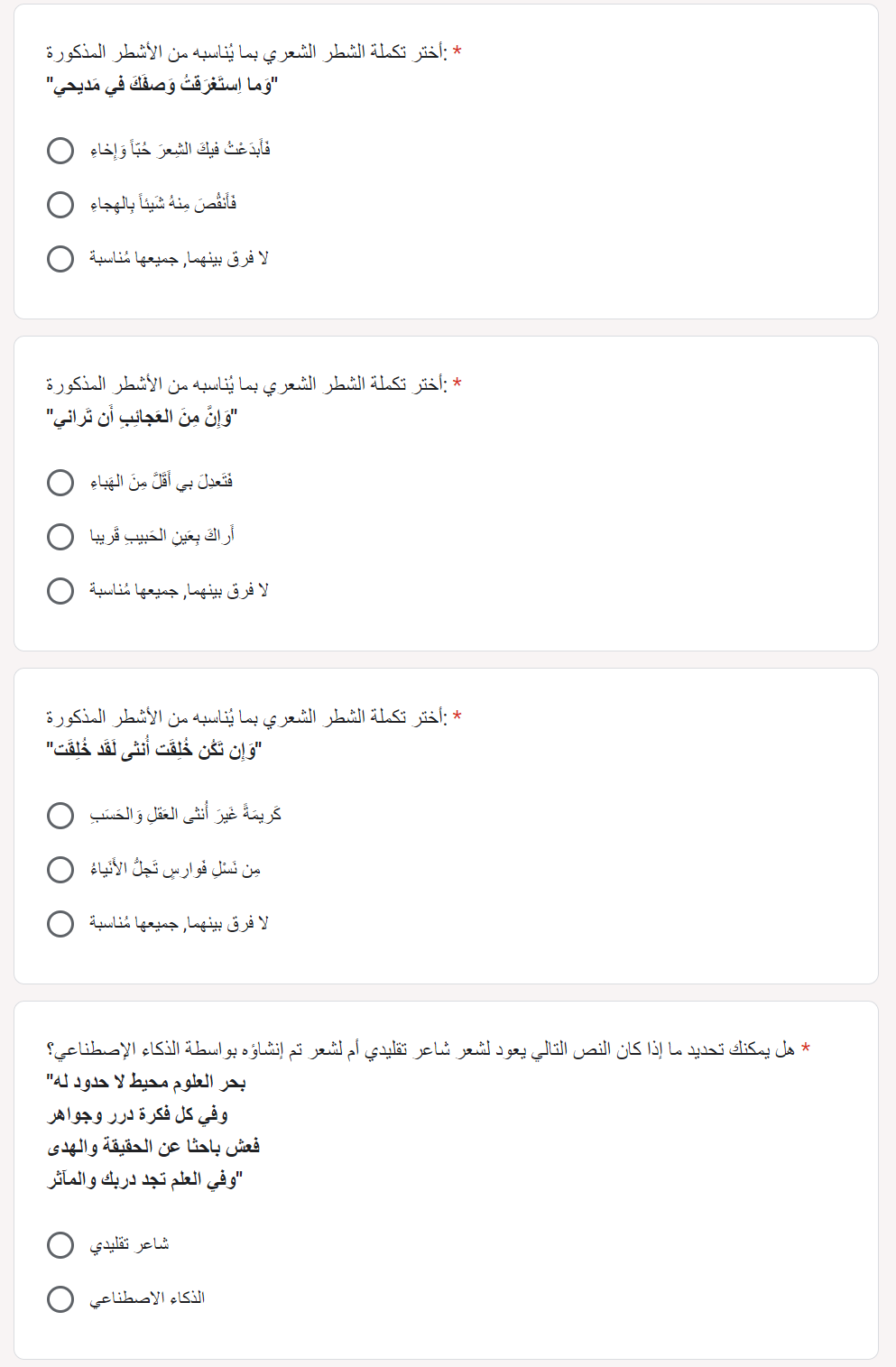
In a session, each question from the user and response from ALLAM are appended to the prompt text. So, each request to ALLAM contains the complete conversation with the user.

* **Check input length for ALLAM limit**

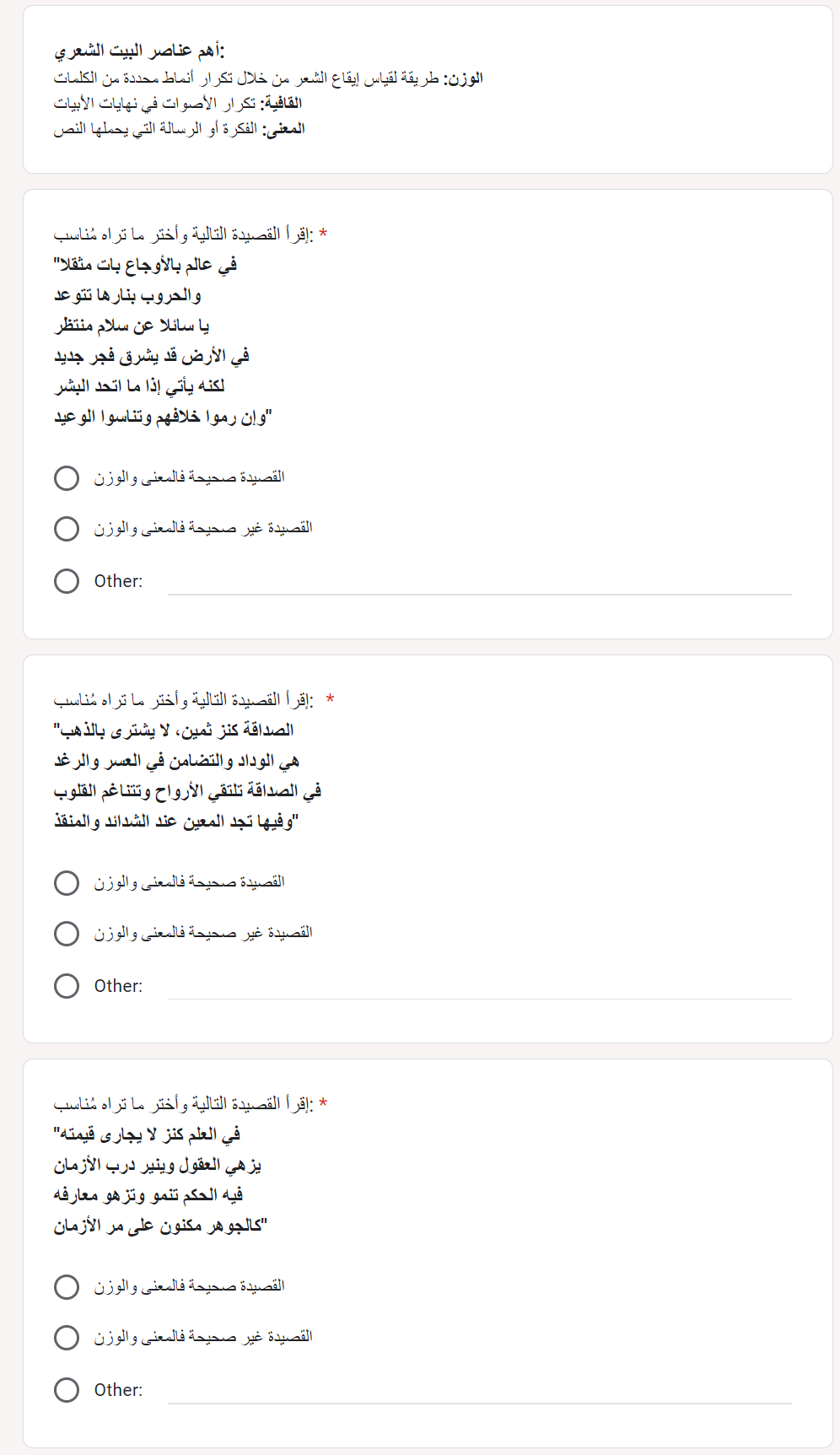
Since ALLAM has a limit for the input (4000 tokens). We check the input length and delete the conversation when the length is about to reach the limit. Therefore, no error from ALLAM will appear.

**Appendix 1**

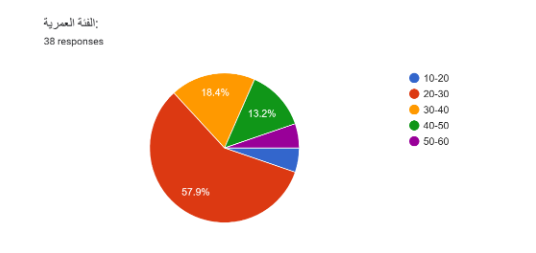
****

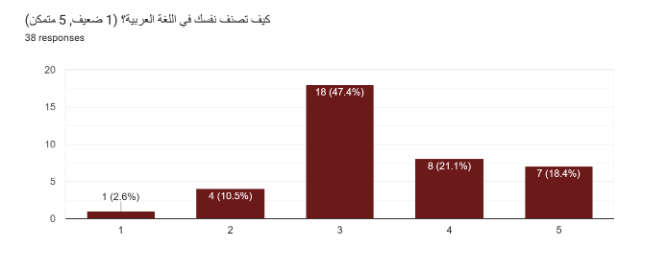
****

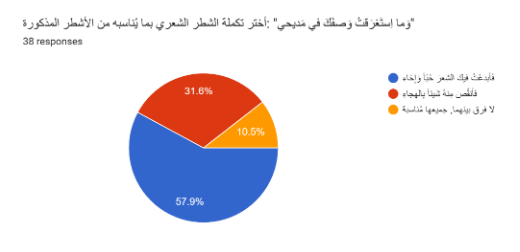
****

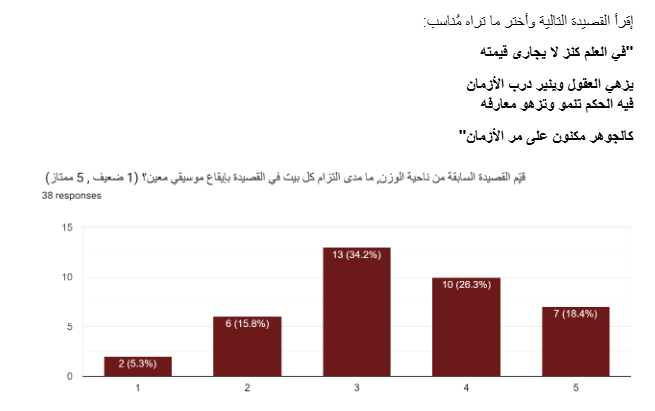
****

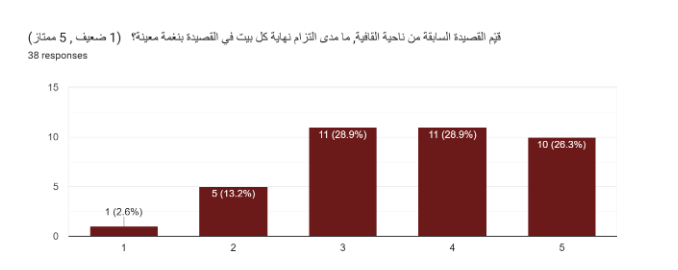
**Some results**

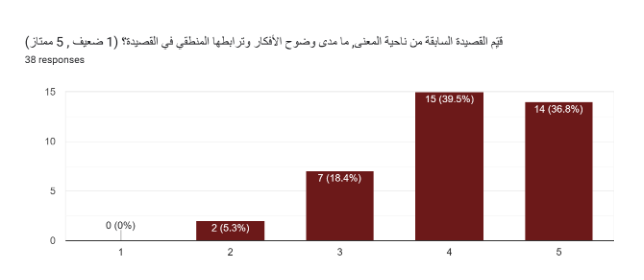
****

****

****

****

****

**Appendix 2**

****

****

****

****

****

**Some results**

