Phase 3 –Image Generation Implementation

1. Introduction

This documentation provides a complete explanation of the Image Generation module developed Phase 4 of the chatbot project. The objective of this module was to integrate AI-powered image generation into the chatbot, giving users the ability to generate, view, and download AI-generated images in an interactive and user-friendly interface.

2. Objectives

- Implement an AI-based image generation feature.
- Allow users to generate new images dynamically.
- Provide an option to download generated images.
- Design and position intuitive buttons for user interaction.
- Ensure a smooth integration of frontend (UI/UX) and backend (API calls).

3. Development Phases

3.1 Phase 4.1 - Initial Image Generation Setup

In this phase, we integrated the backend logic for AI image generation. We used Replicate API to generate images from user prompts. The backend handled API calls and returned the generated image URL and the model we used for the image generation is (black-forest-labs/flux-dev).

Example backend logic (Python):

```
import replicate
output = replicate.run("black-forest-labs/flux-dev", input={"prompt": "a cute cat"})
print(output)  # Returns image URL
```

3.3 Phase 4.2 - Frontend Integration

In this step, we updated the frontend to allow users to generate and view images directly. We added two major UI buttons:

1. Generate Image Button – triggers the backend to create a new image.
2. Download Image Button – allows the user to download the generated image.

The buttons were designed using SVG icons for a modern look and were placed in the bottom-left corner of the interface for accessibility.

Example frontend (JavaScript snippet):

```
const generateBtn = document.getElementById("generateBtn");
const downloadBtn = document.getElementById("downloadBtn");

generateBtn.addEventListener("click", () => {
  fetch("/generate-image")
    .then(res => res.blob())
    .then(blob => {
      const url = URL.createObjectURL(blob);
      document.getElementById("imageDisplay").src = url;
    });
});
```

3.4 Phase 4.3 - Button Design Improvements

Initially, the buttons were simple HTML buttons, but we later replaced them with SVG-based icons to enhance aesthetics. The sound button was also repositioned to maintain UI consistency.

3.5 Phase 4.4 – Remove Speech SVG from image chat bubble

In tha the task focuses on refining the image chat bubble interface by removing the speech (sound) SVG button specifically from all image-generated chat bubbles, while keeping it intact and functional for the remaining text-based chat bubbles. This adjustment ensures that the sound feature is only available where it makes sense—on text responses—since images do not require text-to-speech playback. By applying this selective removal, the interface becomes more logical, user-friendly, and visually consistent, preventing unnecessary or non-functional buttons from appearing on image-only outputs.

3.6 Phase 4.5 – How to prompt to Generate the image:

Unlike ChatGPT, this system works on a **rule-based approach**, meaning it does not automatically interpret prompts to decide whether to generate text or images. To generate an image, the user must explicitly start their prompt with **“generate image”**, as this rule clearly instructs the system to switch to image generation mode. Without this specific keyword at the beginning, the system will not create an image and will instead treat the input as a normal text request.

4. Conclusion

The Image Generation module has been successfully implemented. Users can now:

- Enter prompts to generate AI images.
- View generated images in the chatbot interface.
- Download the generated images with a single click.

NOTE

The API used for image generation is **not completely free**. When you first sign up on the Replicate website, it provides you with a token that allows a limited number of free trials, typically enough to generate around **10 to 15 images**. After these initial free attempts, continued usage of the model requires **payment**, as the free access is only provided at the beginninfor testing purposes.