**Gemini Discord Bot**

Gemini Discord Bot is an advanced, multimodal Discord bot leveraging Google Generative AI capabilities. Besides text responses, this bot can read images, listen to audio files, watch videos, and interpret text files. This makes Gemini one of the most versatile AI assistants available for Discord.

**Features**

* **Multimodal Capabilities:** Interacts with images, videos, audio files, and text files.
* **Image Generation:** Generates images based on user prompts using multiple models like SD-XL, Playground, and more.
* **Story Generation:** Generates stories based on user prompts with a desired length.
* **Poem Generation:** Generates poems based on user prompts to give the theme of the poem.
* **Notifications:** Generates notifications that allows the user to keep track of events they wish to keep tabs on.
* **Alarms:** Generates alarm and messages based on user input with editing features such as viewing and deleting.
* **Speech Generation:** Converts text to speech in multiple languages.
* **Music Generation:** Generates music based on textual prompts.
* **Custom Personalities:** Supports creating custom personalities for users and servers.
* **Advanced Interaction:** Supports advanced user and server settings, including chat history management, response formatting, QNA, and help manual.

**Getting Started**

**Prerequisites**

* Node.js (v14 or later)
* Discord Bot account & token ([create a bot on Discord Developer Portal](https://discord.com/developers/applications))
* Google AI Studio & API key ([create a project on Google AI Studio](https://aistudio.google.com/apikey))

**Installation**

1. Clone the repository:

git clone https://github.com/hihumanzone/Gemini-Discord-Bot

cd Gemini-Discord-Bot

1. Install dependencies:

npm install @google/generative-ai axios cheerio discord.js dotenv eventsource fs sharp office-text-extractor youtube-transcript node-os-utils ws mathjs

1. Set up environment variables:

Edit the .env file in the root directory and add your Discord bot token and Google API key:

DISCORD\_BOT\_TOKEN=your\_discord\_bot\_token

GOOGLE\_API\_KEY=your\_google\_api\_key

CLIENT\_ID=your\_discord\_bot\_client\_ID

1. Setting up Discord Developer Portal

Refer to Discord Bot Set Up Instructions below

1. Configure the bot:

Edit the config.json file as per your preferences.

1. Start the bot:

npm start

**Discord Bot Set Up (Discord Developer Portal)**

* 1. Create New Application under Applications (Login to your Discord Account)

Name it “PEM PAL”

* 1. Finding Discord Token and Client ID:

Discord Token: Bot > Token > Reset Token

After resetting the token, save it in the .env file. You shouldn’t need to reset the token again after this step.

Client ID: OAuth2 > Client Information > Client ID

Save the Client ID in the .env file.

* 1. Setting up the Bot

Under Bot > Privileged Gateway Intents

Toggle “Presence Intent”, “Server Members Intent” and “Message Content Intent” from OFF to ON

* 1. Installing the Bot into your server

Go to Discord, create a NEW server in Discord

Return to the Discord Developer Portal, Under Installation > Install Link

**!!!IMPORTANT!!**

Using the Discord Provided Link, copy it and add this behind:

**“scope=bot%20applications.commands&permissions=2147614728”**

This line allows the bot to use slash commands and join servers as an Application along with the permission integers that is needed to run the Bot

Discord doesn’t add it by default and without it, the bot will not run in your Discord Server.

Your Install Link should look like this:

https://discord.com/oauth2/authorize?client\_id=YOUR\_BOT\_ID**&scope=bot%20applications.commands&permissions=2147614728**

Put this into your browser and select the Discord Server you just created.

* 1. Return to Installation Instructions

**Commands**

**AI Generation**

* /imagine [prompt] [model] [resolution]: Generate an image using a selected model and resolution.
* /speech [language] [prompt]: Generate speech from text in the specified language.
* /music [prompt]: Generate music based on a prompt.
* /write\_story [prompt] [length]: Generate a story based on the prompt with a choice of length
* /write\_poem [theme]: Generates a poem based on the theme given

**Chat Management**

* /respond\_to\_all: Enable the bot to respond to all messages in the channel.
* /clear\_memory: Clear the conversation history.
* /settings: Open user settings.
* /server\_settings: Open server settings.

**Admin Commands**

* /blacklist [user]: Blacklist a user from using certain interactions.
* /whitelist [user]: Remove a user from the blacklist.
* /status: Display bot's CPU and RAM usage in detail.

**Assistance Commands**

* /questions: Generates a list of frequently asked questions.
* /help: Generates the manual for PEM PAL.

**Functions Commands**

* /alarm [date time message]: Generates an alarm for the user.
* /view\_alarms: Allow users to view their alarms.
* /delete\_alarm [number]: Allow users to delete their alarms.
* /notify [message] [time]: bot sets a notification for an event

**Configuration**

**config.json**

This file contains various settings for the bot:

* **defaultResponseFormat**: The default response format for the bot. (embedded or normal)
* **hexColour**: The default hex color for embeds.
* **defaultImgModel**: The default image generation model.
* **defaultUrlReading**: Default URL reading preference (ON or OFF).
* **workInDMs**: Whether the bot should work in DMs (true or false).
* **shouldDisplayPersonalityButtons**: Whether to display personality buttons in settings (true or false).
* **SEND\_RETRY\_ERRORS\_TO\_DISCORD**: Whether to send retry errors to Discord channel (true or false).
* **activities**: An array of activities for the bot to display.
* **defaultPersonality**: The default personality instructions for the bot.
* **defaultServerSettings**: The default settings for the server.

**NSFW Word Filtering (nsfwWords.json)**

* Contains an array of words that should be filtered from prompts.

**Notes/Tips for Future Batches.**

1. Use Visual Studio Code, download Github by KnisterPeter under Extensions.

Helps with cloning the GitHub repository and allows you to commit etc.

1. All functions and commands work as listed in the manual (/help)
2. For Image and Music generation, give it 1 to 2 minutes to generate your prompt
3. Take note of NSFW filter (Might need testing/fixing again)
4. If the bot doesn’t work, try deleting node\_modules and reinstalling dependencies.
5. Console log may require tidying up (notify function)
6. This line may appear on the console log, it is fine.

“ (node:23612) Warning: Supplying "ephemeral" for interaction response options is deprecated. Utilize flags instead.

(Use `node --trace-warnings ...` to show where the warning was created) “