Story Generator Bot

We created this bot for children aged 4 to 12 years old to give them the opportunity to read a book with their favorite characters from movies, cartoons, or games. Nowadays, children are not reading books; they are simply playing computer games or watching cartoons. Our bot provides parents with a stimulus to encourage their children to read books with their beloved characters.

Here's how our code works:

1. Necessary modules and libraries are imported:

- `asyncio` for asynchronous programming.

- `aiogram` for creating and managing a Telegram bot.

- `openai` for interacting with the OpenAI API.

- `dotenv` for loading environment variables from the `.env` file.

- `os` for accessing operating system functionality.

2. Environment variables are loaded from the `.env` file using `dotenv.load\_dotenv()`. This allows secure storage of confidential information such as API keys.

3. The OpenAI API key is set by assigning its value to `openai.api\_key` from the environment variable `AI\_TOKEN`.

4. An instance of the bot is created using the Telegram bot token from the environment variable `BOT\_TOKEN`.

5. An instance of the Dispatcher is created to handle incoming bot messages.

6. Message handlers are defined for the `/start` and `/help` commands:

- The `send\_welcome` function is called when the user sends any of these commands.

- It sends a welcome message to the user, addressing them by name.

7. An asynchronous `generate\_story` function is defined to generate a story based on user input:

- The function takes a message as input.

- First, the entered data is checked by sending a request to OpenAI for verification.

- If the entered data passes the verification as a character name, a request is sent to OpenAI to generate a story.

- The generated story is sent back to the user.

8. A message handler is defined for general messages:

- The `handle\_message` function is called when the user sends any message other than the `/start` and `/help` commands.

- It sends a message to the user indicating that the story generation process is in progress.

- Then, the `generate\_story` function is called to generate a story based on the user's message.

9. The `main` function is defined as the entry point to run the bot:

- It starts the Dispatcher to listen for incoming messages.

10. `asyncio.run(main())` is called to run the bot and start receiving messages.

Our code sets up a Telegram bot using the aiogram library, interacts with the OpenAI API to generate stories based on user input, and sends the generated stories back to the users.