This Python code sets up a custom chatbot using the GPT-3.5 Turbo model from OpenAI and deploys it with Gradio, a library for creating user interfaces for machine learning models. I'll provide a detailed line-by-line explanation of the code:

1. `import openai`: This line imports the OpenAI Python library, which is used to interact with OpenAI's GPT-3 models.

2. `import gradio`: This line imports the Gradio library, which is used to create a user interface for the chatbot.

3. `openai.api\_key = "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"`: Here, you set the OpenAI API key to authenticate with OpenAI's servers. This key is necessary to make requests to the GPT-3 model. Make sure to keep this key secure, as it provides access to your OpenAI account.

4. `messages = [{"role": "system", "content": "You are a doctor"}]`: This initializes a list called `messages` with a single message. This message specifies that the chatbot is being initialized as a doctor, and it's a system message.

5. `def CustomChatGPT(user\_input)`: This line defines a Python function named `CustomChatGPT`, which takes a `user\_input` parameter. This function will be responsible for processing user inputs and generating responses.

6. `messages.append({"role": "user", "content": user\_input})`: This line appends a user's input message to the `messages` list, marking it as a "user" message. It includes the text input provided by the user.

7. `response = openai.ChatCompletion.create(...)`: In this line, the code sends a request to the GPT-3.5 Turbo model for generating a response based on the accumulated `messages`. It uses the `openai.ChatCompletion.create` method with the model and messages as parameters.

- `model = "gpt-3.5-turbo"`: Specifies the GPT-3.5 Turbo model to use for generating the response.

- `messages = messages`: Passes the list of messages, which includes both user and system messages, to the model.

8. `ChatGPT\_reply = response["choices"][0]["message"]["content"]`: This line extracts the content of the assistant's response from the response returned by the GPT-3 model.

9. `messages.append({"role": "assistant", "content": ChatGPT\_reply})`: The generated assistant response is appended to the `messages` list, marking it as an "assistant" message.

10. `return ChatGPT\_reply`: The function returns the assistant's response, which will be used to display the response to the user.

11. `demo = gradio.Interface(...)`: This block creates a Gradio user interface for the chatbot.

- `fn=CustomChatGPT`: Specifies the function to be used for generating responses, which is the `CustomChatGPT` function defined earlier.

- `inputs = "text"`: Specifies that the input to the chatbot will be a text input.

- `outputs = "text"`: Specifies that the output from the chatbot will be text.

- `title = "DocPro"`: Sets the title of the Gradio interface to "DocPro."

12. `demo.launch(share=True)`: This line launches the Gradio interface, making the chatbot accessible through a web interface. The `share=True` option allows you to share the interface with others, enabling them to interact with the chatbot via a web browser.

With this code, you can interact with a chatbot that takes user input, generates responses using the GPT-3.5 Turbo model, and displays the results through a web interface created with Gradio.