**Step1: Set up your Open AI account**

Sign up for an O[penAI account](https://platform.openai.com/signup) if you haven’t already

Get your API key from the OpenAI dashboard. This key will allow you to make requests to the GPT model API

**Step 2: Set Up Your Environment**

Create a new directory for your project and navigate to it using the terminal.

Install Python in your system. Visit the link [**https://www.python.org**](https://www.python.org/) to download the latest release of Python.

After Python installation, execute the below command to install virtualenv package inside the terminal:

pip install virtualenv

[Open in app](https://rsci.app.link/?%24canonical_url=https%3A%2F%2Fmedium.com%2Fp%2Fea39439cf9ed&%7Efeature=LoOpenInAppButton&%7Echannel=ShowPostUnderUser&source=---two_column_layout_nav----------------------------------)

Sign up

[Sign In](https://medium.com/m/signin?operation=login&redirect=https%3A%2F%2Fmedium.com%2F%40mailsushmita.m%2Fcreate-an-generative-ai-chatbot-using-python-and-flask-a-step-by-step-guide-ea39439cf9ed&source=post_page---two_column_layout_nav-----------------------global_nav-----------)



# Create an Generative-AI chatbot using Python and Flask: A step by step guide

[InnovatewithDataScience](https://medium.com/@mailsushmita.m?source=post_page-----ea39439cf9ed--------------------------------)

[InnovatewithDataScience](https://medium.com/@mailsushmita.m?source=post_page-----ea39439cf9ed--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2Fee46a1334f52&operation=register&redirect=https%3A%2F%2Fmedium.com%2F%40mailsushmita.m%2Fcreate-an-generative-ai-chatbot-using-python-and-flask-a-step-by-step-guide-ea39439cf9ed&user=InnovatewithDataScience&userId=ee46a1334f52&source=post_page-ee46a1334f52----ea39439cf9ed---------------------post_header-----------)

4 min read

·

Sep 4

[Listen](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2Fplans%3Fdimension%3Dpost_audio_button%26postId%3Dea39439cf9ed&operation=register&redirect=https%3A%2F%2Fmedium.com%2F%40mailsushmita.m%2Fcreate-an-generative-ai-chatbot-using-python-and-flask-a-step-by-step-guide-ea39439cf9ed&source=-----ea39439cf9ed---------------------post_audio_button-----------)

Share



Photo by [Andrew Neel](https://www.pexels.com/@andrew/) on Pexels

## Introduction

Chatbots have become an integral part of modern applications, enhancing user engagement and providing instant support. In this tutorial, we’ll walk through the process of creating a chatbot using the powerful GPT model from OpenAI and Python Flask, a micro web framework. By the end of this guide, you’ll have a functional chatbot that can hold interactive conversations with users.

## **Prerequisites**

1. Basic understanding of python programming.
2. Familiarity with web development and API’s.

**Step1: Set up your Open AI account**

Sign up for an O[penAI account](https://platform.openai.com/signup) if you haven’t already

Get your API key from the OpenAI dashboard. This key will allow you to make requests to the GPT model API.

**Step 2: Set Up Your Environment**

Create a new directory for your project and navigate to it using the terminal.

Install Python in your system. Visit the link [**https://www.python.org**](https://www.python.org/) to download the latest release of Python.

After Python installation, execute the below command to install virtualenv package inside the terminal:

pip install virtualenv

Create a virtual environment to isolate your project dependencies:

virtualenv chatbot\_env #chatbot\_env is the name of the environment

Activate the virtual environment:

source venv/bin/ chatbot\_env

**Step 3: Install Required Libraries**

Install Flask:

pip install Flask

Install the OpenAI library:

pip install openai

**Step 4: Create Flask App Structure**

Create a new Python file, e.g., app.py, to hold your Flask application.

Import necessary modules:

from flask import Flask,render\_template, request, jsonify

Initialize the Flask app:

#app.py  
#import files  
from flask import Flask, render\_template, request  
app = Flask(\_\_name\_\_)  
@app.route("/")  
def home():  
 return "Hello, This is Flask Application"  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run()

**Step 5: Create HTML Template:**

Create a folder static in your project directory, place the image that needs to be displayed in the UI and name it as **chatbot1.png**

Create a templates folder in your project directory.

Inside the templates folder, create an HTML file, e.g., index.html. This file will be the frontend of your chatbot.

<!DOCTYPE html>  
<html>  
<head>  
 <title>GenAI-Bot</title>  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  
 <style>  
 \* {  
 box-sizing: border-box  
 }  
/\* Set height of body and the document to 100% \*/  
 body, html {  
 height: 100%;  
 margin: 0;  
 font-family: Arial;  
 }  
 #chatbox {  
 margin-left: auto;  
 margin-right: auto;  
 width: 40%;  
 margin-top: 60px;  
 }  
 #userInput {  
 margin-left: auto;  
 margin-right: auto;  
 width: 40%;  
 margin-top: 60px;  
 }  
 #textInput {  
 width: 90%;  
 border: none;  
 border-bottom: 3px solid black;  
 font-family: monospace;  
 font-size: 17px;  
 }  
 .userText {  
 color: white;  
 font-family: monospace;  
 font-size: 17px;  
 text-align: right;  
 line-height: 30px;  
 }  
 .userText span {  
 background-color: #808080;  
 padding: 10px;  
 border-radius: 2px;  
 }  
 .botText {  
 color: white;  
 font-family: monospace;  
 font-size: 17px;  
 text-align: left;  
 line-height: 30px;  
 }  
 .botText span {  
 background-color: #4169e1;  
 padding: 10px;  
 border-radius: 2px;  
 }  
 #tidbit {  
 position: absolute;  
 bottom: 0;  
 right: 0;  
 width: 300px;  
 }  
 .boxed {  
 margin-left: auto;  
 margin-right: auto;  
 width: 78%;  
 margin-top: 60px;  
 border: 1px solid green;  
 }  
 </style>  
</head>  
<body>  
<div>  
 <h1 align="center"><b>AI-Gen ChatBot</b></h1>  
 <h4 align="center"><b>Please start your personalized interaction with the chatbot</b></h4>  
 <p align="center"><img src="static\\chatbot1.png" alt="Python-BOT" height="210" width="220"></p>  
 <div class="boxed">  
 <div>  
 <div id="chatbox">  
 <p class="botText">  
 <span>Hi! I'm your AI-Generative Chatbot</span>  
 </p>  
 </div>  
 <div id="userInput">  
 <input id="textInput" type="text" name="msg" placeholder="Message" />  
 </div>  
 </div>  
 <script>  
 function getBotResponse() {  
 var rawText = $("#textInput").val();  
 var userHtml = '<p class="userText"><span>' + rawText + "</span></p>";  
 $("#textInput").val("");  
 $("#chatbox").append(userHtml);  
 document  
 .getElementById("userInput")  
 .scrollIntoView({ block: "start", behavior: "smooth" });  
 $.get("/get", { msg: rawText }).done(function (data) {  
 var botHtml = '<p class="botText"><span>' + data + "</span></p>";  
 $("#chatbox").append(botHtml);  
 document  
 .getElementById("userInput")  
 .scrollIntoView({ block: "start", behavior: "smooth" });  
 });  
 }  
 $("#textInput").keypress(function (e) {  
 if (e.which == 13) {  
 getBotResponse();  
 }  
 });  
 </script>  
 </div>  
 </div>  
</body>  
</html>

**Step 6: Implement Backend Logic:**

In app.py create a route for the index page and return the HTML template

Import the openai library: import openai

Set up your OpenAI API key: openai.api\_key = “YOUR\_OPENAI\_API\_KEY”

Use the get\_completion() function to interact with the GPT-3.5 model and get the response for the user query.

#Final app.py   
#import files  
from flask import Flask, render\_template, request  
import openai  
app = Flask(\_\_name\_\_)  
openai.api\_key = "<place your openai\_api\_key>"  
  
def get\_completion(prompt, model="gpt-3.5-turbo"):  
 messages = [{"role": "user", "content": prompt}]  
 response = openai.ChatCompletion.create(  
 model=model,  
 messages=messages,  
 temperature=0, # this is the degree of randomness of the model's output  
 )  
 return response.choices[0].message["content"]  
@app.route("/")  
def home():   
 return render\_template("index.html")  
@app.route("/get")  
def get\_bot\_response():   
 userText = request.args.get('msg')   
 response = get\_completion(userText)   
 #return str(bot.get\_response(userText))   
 return response  
if \_\_name\_\_ == "\_\_main\_\_":

**Step 7: Run the Flask App:**

In the terminal, run your Flask app: python app.py

