Creating a Containerised Development Environment

Setting-up Environment for Python Development

Installing Prerequisites

Installing Python and Python Virtual Environment

```
brew install python3
pip3 install virtualenv
```

Initialising The Environment

Creating a Isolated Python Environment

```
virtualenv AI-Chatbot
```

Activating the Environment

```
source AI-Chatbot/bin/activate
```

Installing Python Libraries in the Dev Environment

```
pip3 install numpy
pip3 install matplotlib
pip3 install seaborn
pip3 install tensorflow
```

Setting-up Environment for Web Development

Installing Prerequisites

```
brew install node
brew install tailwindcss tailwindcss-language-server
```

Installing Web Libraries in Dev Environment

```
pip3 install Flask
npm install -D tailwindcss
```

Initialising The Environment

Inside the **AI-Chatbot** Folder

```
npm init -y
# Creates a package.json with default values
npm tailwind init
# Initialises Tailwind
```

Building User Interface(With Flask)

Building a Simple Web UI

Let Us Build a Simple UI with only html, css and javascript

HTML

index.html

```
<!DOCTYPE html>
<html lang="en" class="scroll-smooth">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Paradise</title>
<link rel="stylesheet" href="style.css" />
<script src="index.js" defer></script>
<style>
        .chat-container {
                max-height: 400px;
                overflow-y: auto;
        }
</style>
</head>
<body>
<div class="container">
<div
        id="navbar"
        class="flex bg-orange-100 text-green-800 px-12 py-5 shadow-md justify-between items-
center"
>
<div class="text-3xl font-bold">Paradise</div>
<div class="text-xl">
<ul>
<a
        href="#chatbot"
        class="mx-4 hover:text-green-900 hover:underline"
        >Home</a
>
<a
        href="#chatbot"
        class="mx-4 hover:text-green-900 hover:underline"
        >Booking</a
>
<a
        href="#chatbot"
```

```
        class="mx-4 hover:text-green-900 hover:underline"
        >Contact</a
>
<a
        href="#chatbot"
        class="mx-4 hover:text-green-900 hover:underline"
        >About us</a
>
</ul>
</div>
<div class="text-xl">
<a
href="#chatbot"
class="bg-green-800 text-white px-6 py-2.5 rounded-full hover:bg-green-600"
><button>AI Assistant</button></a
>
</div>
</div>

<div id="hero-section" class="relative">
<img
src="background.jpeg"
alt=""
class="h-[38rem] w-full object-cover brightness-50 relative"
/>
<div
class="absolute top-1/2 left-1/2 -translate-x-1/2 -translate-y-1/2 text-white text-8xl font-
bold"
>
One Step Close <br />
<span class=""
>To
<span
class="text-green-700"
style="-webkit-text-stroke: 1px #fff"
>Paradise</span></span>
</div>
</div>
<!-- Chatbot section -->
<div id="chatbot">
<div class="bg-orange-100 mx-12 my-14 rounded-xl pb-10">
<div class="py-8 px-6 flex flex-col items-center">
<div class="text-3xl font-bold pb-4">AI Assistant</div>
<div class="text-xl">
        Meet your virtual assistant, where smart meets chat. I'm
        here to help!
</div>
</div>
<div class="max-w-xl mx-auto bg-slate-100 rounded shadow-lg">
<div class="chat-container p-4" id="chat-messages"></div>
<div class="flex items-center border-t p-2">
<input
        type="text"
        id="user-input"
        class="flex-1 py-2 px-4 rounded-full mr-2 outline-none"
        placeholder="Type a message..."
/>
```

```
<button
        id="send-button"
        class="bg-green-500 hover:bg-green-700 text-white font-bold py-2 px-4 rounded-full"
>
<svg
        xmlns="http://www.w3.org/2000/svg"
        fill="none"
        viewBox="0 0 24 24"
        stroke-width="1.5"
        stroke="currentColor"
        class="w-6 h-6"
>
<path
        stroke-linecap="round"
        stroke-linejoin="round"
        d="M6 12L3.269 3.126A59.768 59.768 0 0121.485 12 59.77 59.77 0 013.27 20.876L5.999
12zm0 0h7.5"
/>
</svg>
</button>
        </div>
</div>
                </div>
        </div>
        <!-- Footer -->
        <div
                class="flex justify-center bg-gray-200 border-t border-gray-200 shadow-lg"
        >
                <div class="text-lg py-3">
                        Copyright 2023 Pardise. All rights reserved.
                </div>
                </div>
        </div>
        </body>
</html>
```

Javascript

index.js

```
const chatMessages = document.getElementById("chat-messages");
const userInput = document.getElementById("user-input");
const sendButton = document.getElementById("send-button");

function appendMessage(sender, message) {
        const messageDiv = document.createElement("div");
        messageDiv.classList.add("mb-2", "p-2", "rounded", "max-w-fit", "ml-auto");
        if (sender === "user") {
                messageDiv.innerHTML = `<div class="bg-green-500 text-white px-4 py-2
rounded-md">${message}</div>`;
        } else {
                messageDiv.classList.toggle("ml-auto");
                messageDiv.innerHTML = `<div class="bg-gray-300 px-4 py-2 rounded-
md">${message}</div>`;
        }
```

```
        chatMessages.appendChild(messageDiv);
        chatMessages.scrollTop = chatMessages.scrollHeight;
}

function sendMessage() {
        const message = userInput.value;
        if (message.trim() === "") return;
        appendMessage("user", message);
        userInput.value = "";
        // Simulate response from the bot (in this example, a simple echo)
        setTimeout(() => {
                appendMessage("bot", message);
        }, 500);
}

sendButton.addEventListener("click", sendMessage);
userInput.addEventListener("keydown", (e) => {
        if (e.key === "Enter") {
                sendMessage();
        }
});
```

Setting-Up Flask

Checking the Flask Install

```
python -m flask
```

```
Usage: python -m flask [OPTIONS] COMMAND [ARGS]...

  A general utility script for Flask applications.

  An application to load must be given with the '--app' option, 'FLASK_APP'
  environment variable, or with a 'wsgi.py' or 'app.py' file in the current
  directory.

Options:
  -e, --env-file FILE   Load environment variables from this file. python-
                        dotenv must be installed.
  -A, --app IMPORT      The Flask application or factory function to load, in
                        the form 'module:name'. Module can be a dotted import
                        or file path. Name is not required if it is 'app',
                        'application', 'create_app', or 'make_app', and can be
                        'name(args)' to pass arguments.
  --debug / --no-debug  Set debug mode.
  --version             Show the Flask version.
  --help                Show this message and exit.

Commands:
  routes  Show the routes for the app.
```

```
    run     Run a development server.
    shell   Run a shell in the app context.
```

As we can see the flask was properly installed.

Running Flask

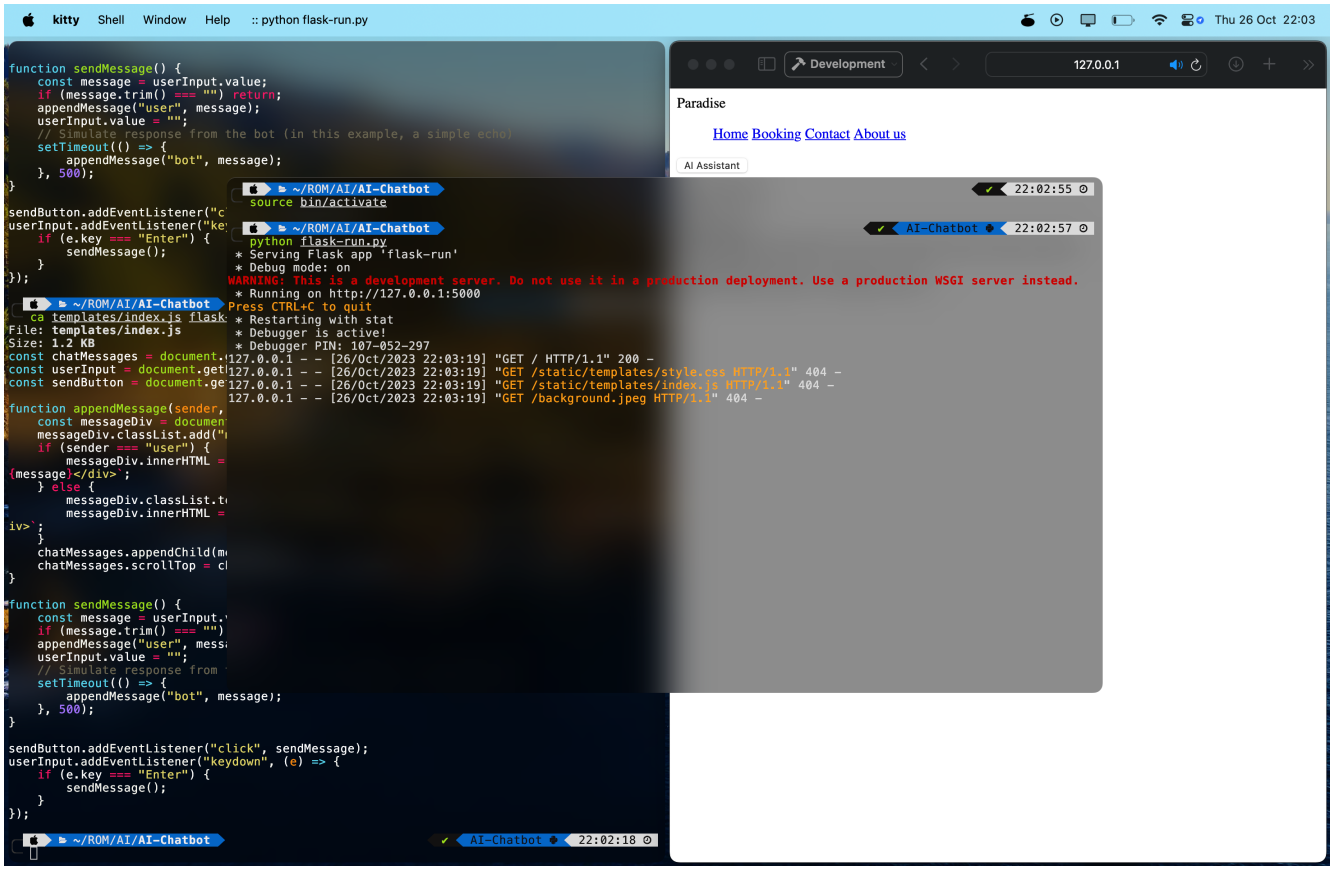
To use Flask to develop the back-end of a web application, we will need to create a Flask application object. We can then use the Flask application object to register routes, which are the paths that users can request to access different pages of our web application.
Once we have registered routes, we can use the Flask application object to generate responses to requests. The responses can be HTML, CSS, JavaScript, or any other type of data that we want to send to the user.

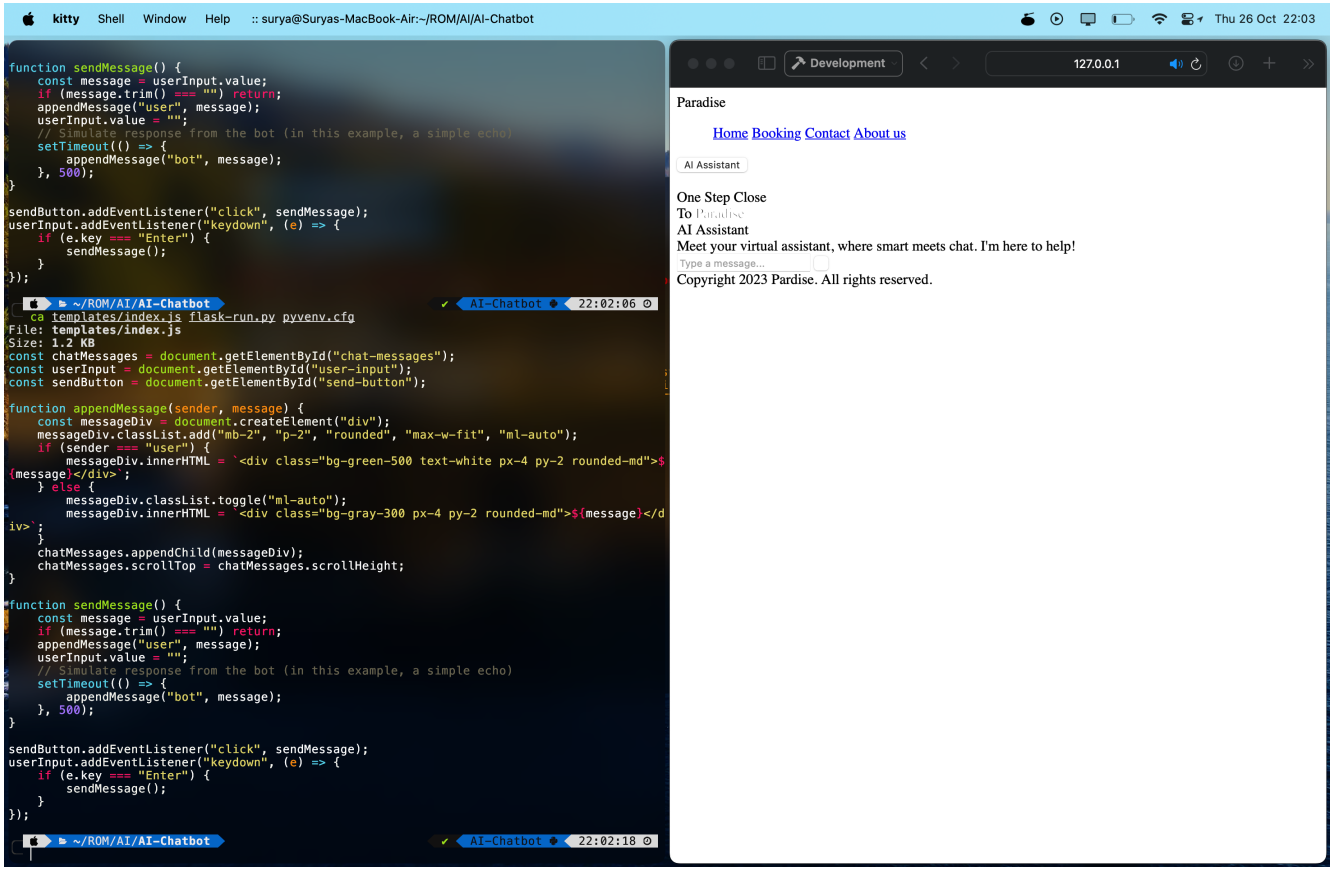
```python
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/")
def index():
    return render_template("index.html")

if __name__ == "__main__":
    app.run(debug=True)
```





Thus we were able to successfully able to Display our HTML in the Flask generated Portal.