**OCTAVIA DOCUMENTATION**

**Title:** Weather APP (Octavia)

This **scope** of this project is to Create a Weather App that predicts the weather situation of an input Location within the Ireland Counties by Users and also give the idea of clothing and Accessories and if any Location aside these counties are inserted. Our App should be able to direct the user to Google for further help and assistance.

**Aim**: To design an app for weather location with the help of calling a weather **API** for a given Location.

On the Project: We are 3 students working as a team to achieve this and each team member have one or more responsibilities to ensure the success of the project.

* **Atilola Damilare**
* **Kennedy Onyemenam**
* **Araujo Janaina**

**Language Picked:** We have different choice of Programming Languages such as Java, PHP, Javascript, Python and so on. But on this Particular Project we chose **JavaScript** because we are much conversant with it and the Interface for Javascript include the Use of HTML (Hyper Text Markup Language and CSS (Cascading Sheet Styling) for the sole purpose of displaying and designing the Project.

API: Rapid Weather API which has a url and key was found at [API Marketplace - Free Public & Open Rest APIs | RapidAPI](https://rapidapi.com/marketplace)

**Testing:** There are various types of Testing Software for Testing Javascript Project but we chose CYPRESS which we find more popular and easily accessible.

How to install, run, test and so on was found at [JavaScript End to End Testing Framework | cypress.io](https://www.cypress.io/)

**Research & Gathering Information**

We spent the 1st 2 weeks of our Project Timeline to Make quality Research and findings which Involves

* Decision Tree.
* Discovering of Cypress and
* The Weather API used and so many examples that could help by giving the the Ideas needed to successcfully carry out the Project.

[JavaScript Tutorial (w3schools.com)](https://www.w3schools.com/js/default.asp)

[Build a Simple Weather App With Vanilla JavaScript (tutsplus.com)](https://webdesign.tutsplus.com/tutorials/build-a-simple-weather-app-with-vanilla-javascript--cms-33893)

**Project Charts**

**Graphical user interface

Description automatically generated with medium confidence**

**Creating Each Component parts of the Project**

The reason for this is to make the project Very easy to Join at the End, hence we spent quality time in creating each Parts which include

* Calling the API.
* Creating the Weather App Interface.
* Converting Date, Degrees.
* Creating Voice Recognition and Text to speech.
* Listing and Suggesting the Type of Clothing and Accessories to bring along.
* Listing out all the Counties.
* Wireframes of how the Project will Look like.

**PROJECT WIREFRAMES**

**Diagram

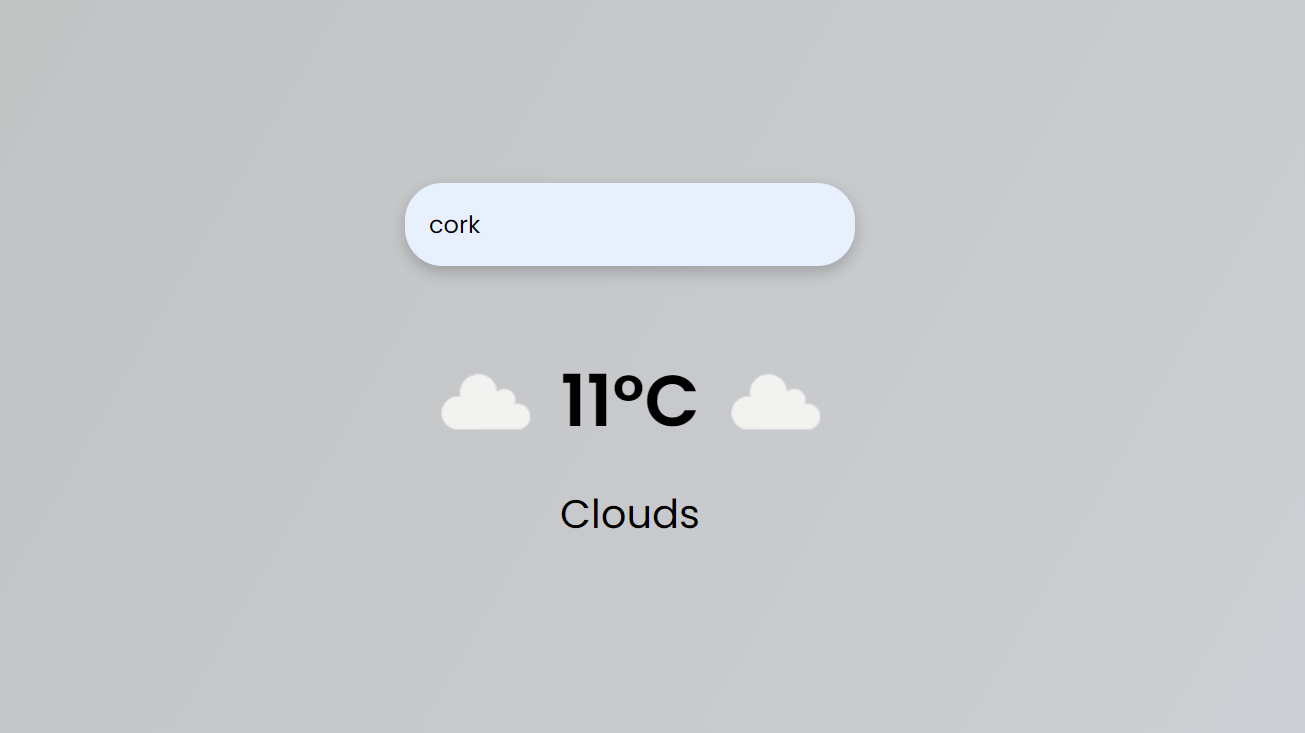
Description automatically generated**

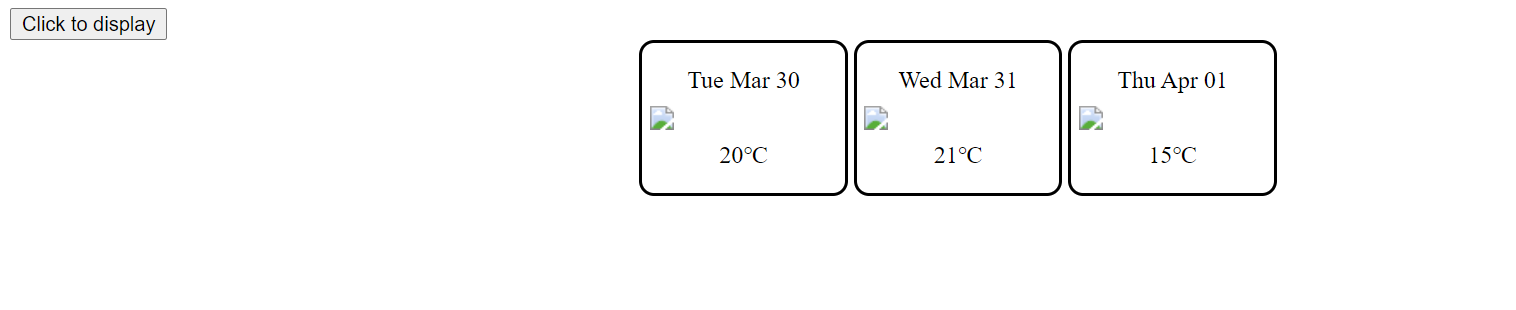
**Diagram

Description automatically generated**

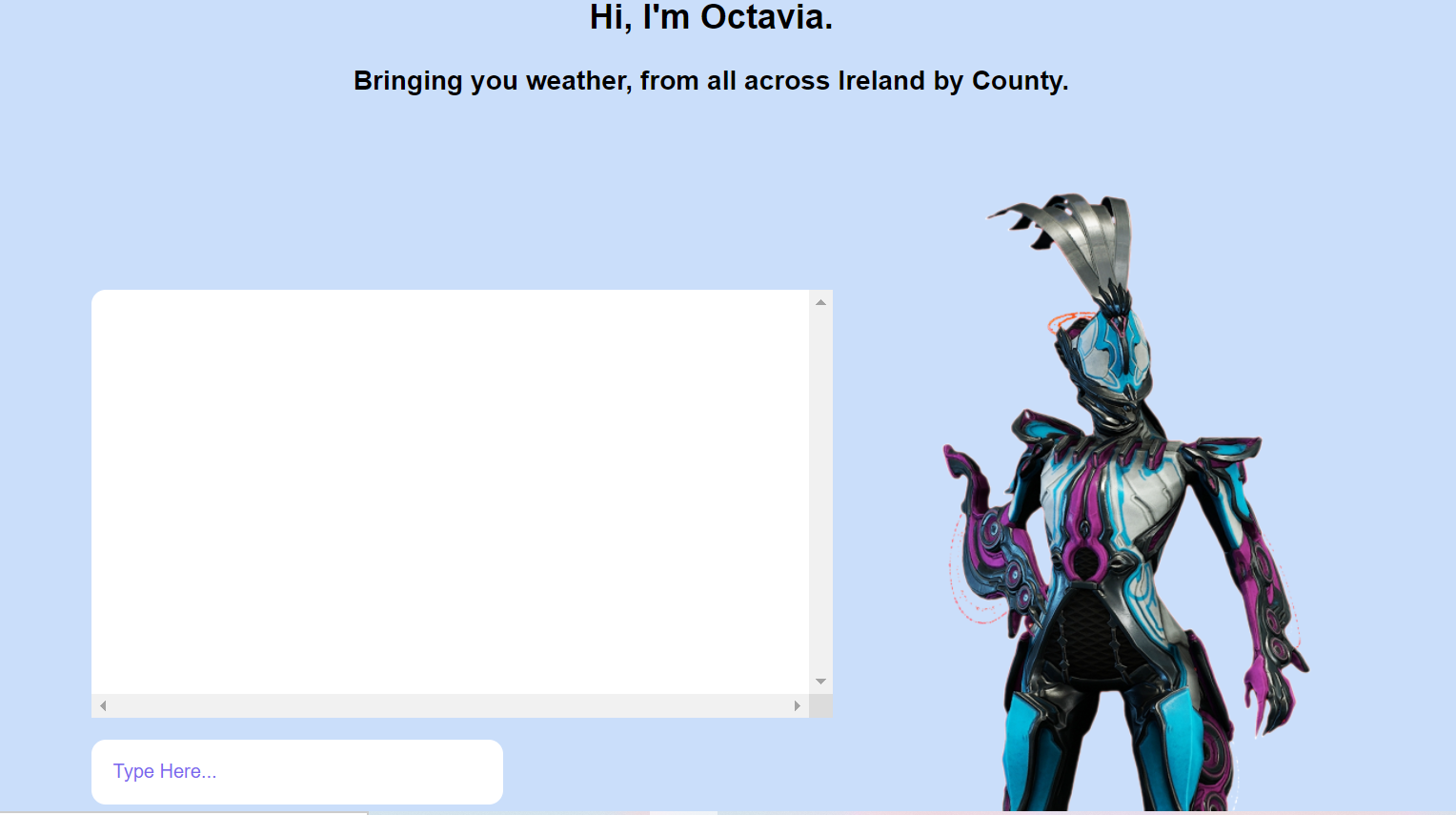
As the Project Progress we keep running the Test part of it.

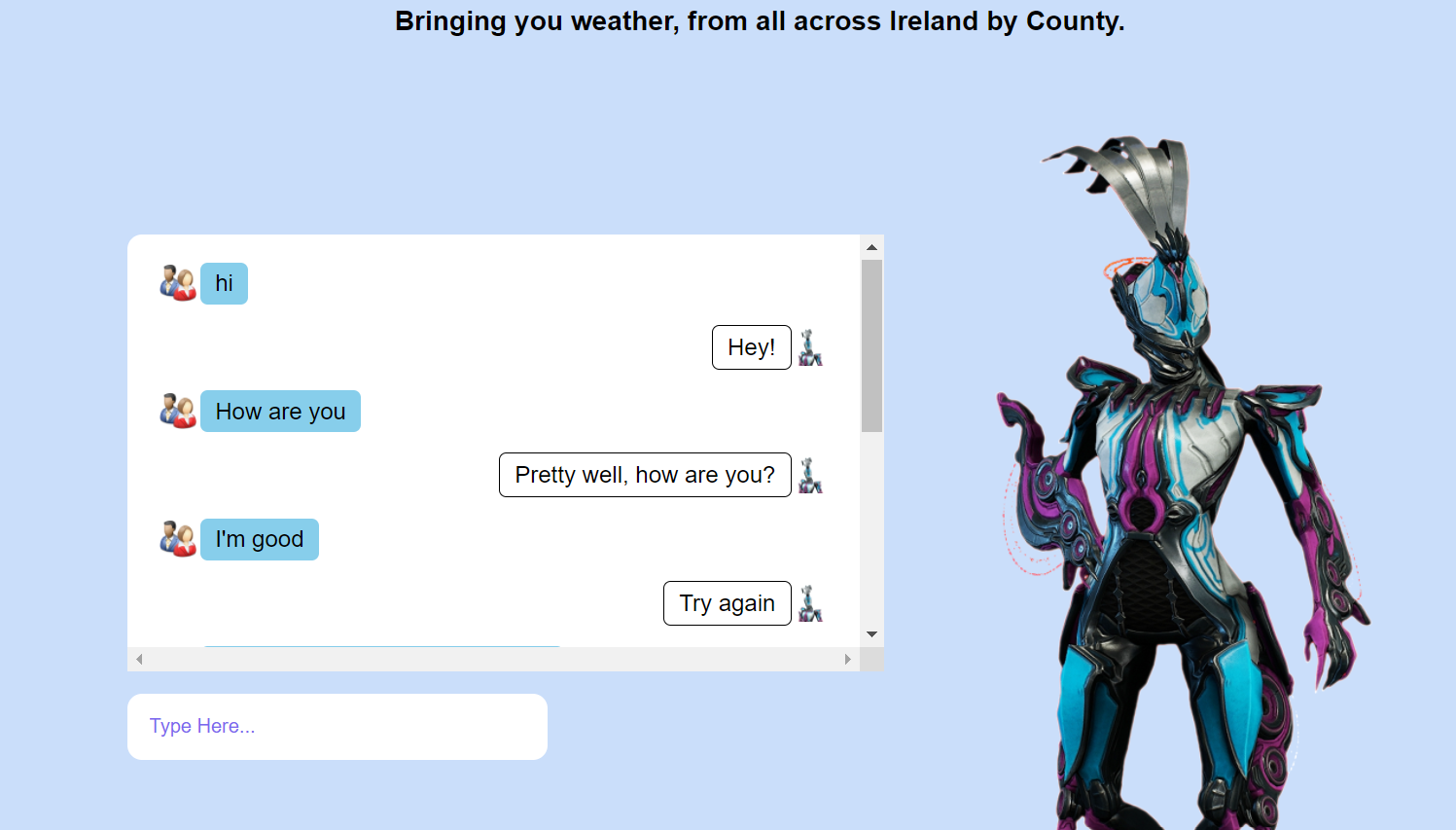
Before the 1st Milestone review, we have our project API working Perfectly by searching for Locations and Displaying the Weather with their Appropriate Icons. And they are displayed below as:

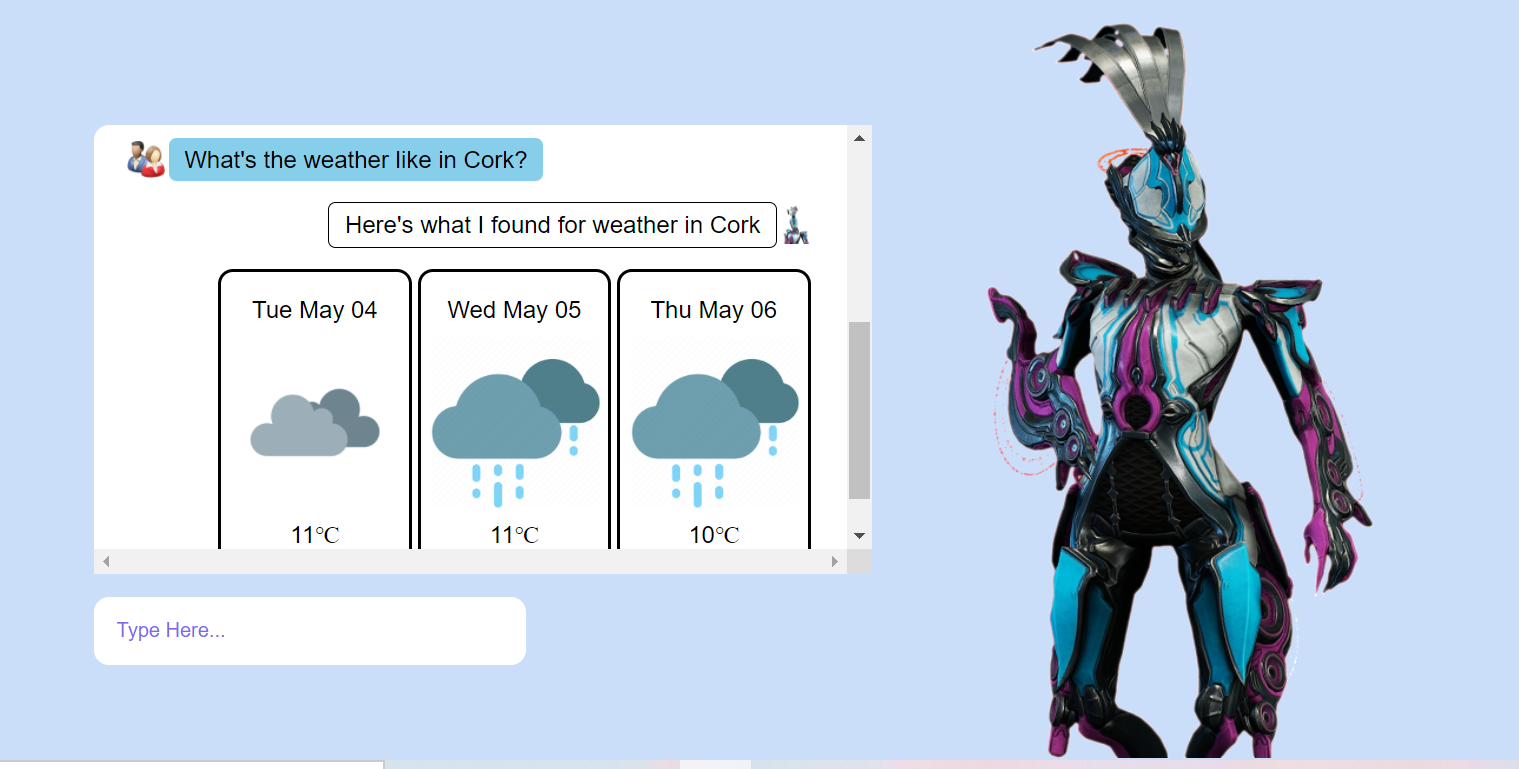




Then during 2nd Milestone Review we have our Project looking like this:





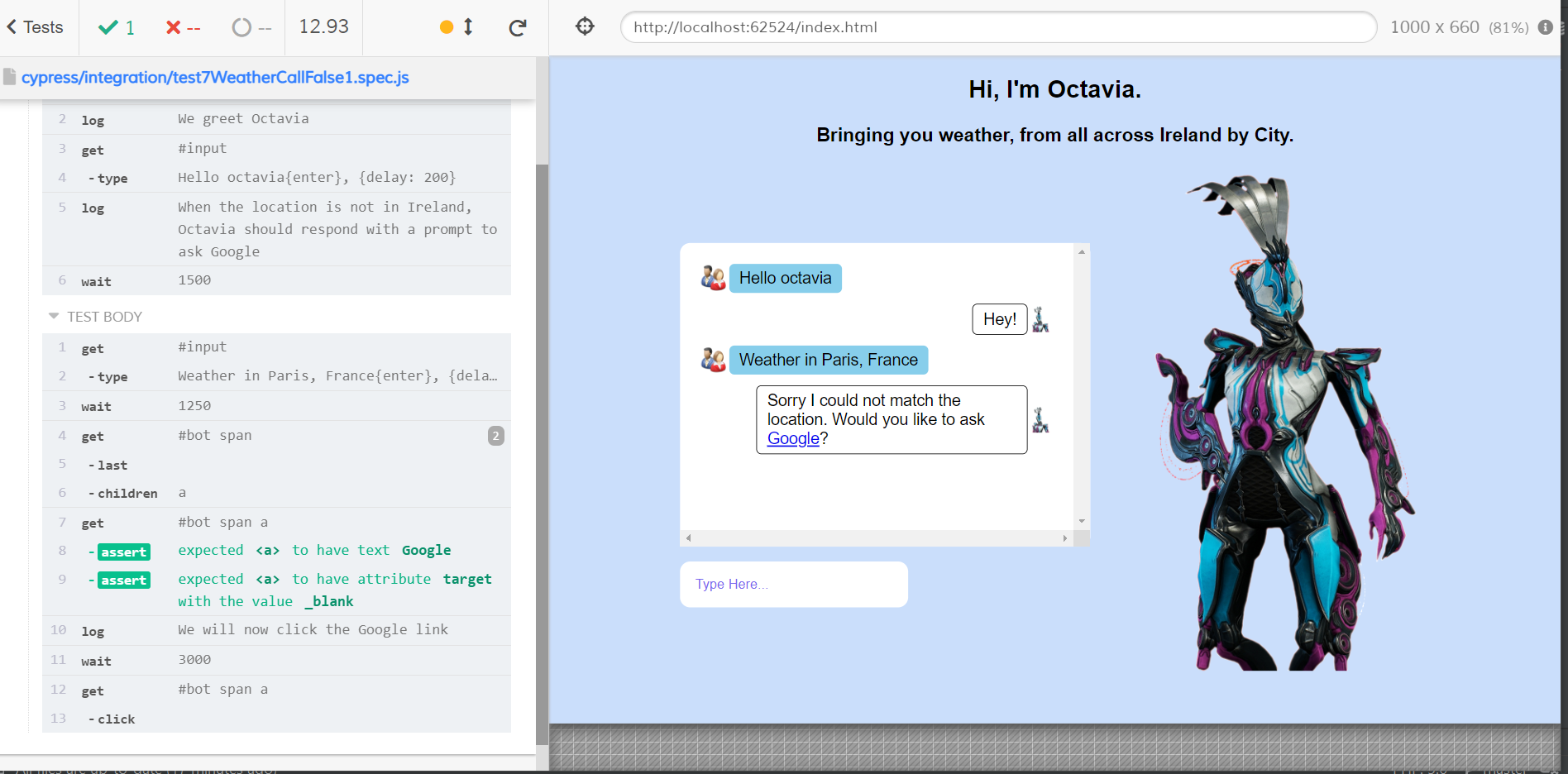


As these works are going on, so also their testing.

We had 5 tests Ran already and everything is working. The Next Phase is to update the project by adding Clothing Accessories to our Weather API response based on the Result given.

We had to change from counties to Cities because some counties are not recognized in our API calls.

We have had 7 Test Written and all are successful.



**Output (Interface) of One of the tests being ran using Cypress.**

**We** added Clothing to our Bot, which will display immediately after returning the Weather forecast for any requested location.

We did this by asking the Bot to give a quality suggestion of Materials, Shoes and Extra Accessories to take along with them based on the condition of the weather.

Graphical user interface, application

Description automatically generated