|  |  |
| --- | --- |
| https://www.griffith.ie/sites/all/themes/griffith/images/griffith-title-logo.png | |
| **Course and Module Information** | |
| **Academic Year** | 2021-2022 |
| **Semester** | 2 |
| **Course** | BSCH |
| **Year** | 2 |
| **Module** | Software Development 2 |
|  |  |
| **MILESTONE** | **REVIEW 1** |

|  |  |
| --- | --- |
| **Student Information** | |
| **Student Name** | Baraah Afana, Arthur Martins, Mert Bekar |
| **Student Number** | 3054399, 3028568,3050376 |
| **Code Review #** | 1 |

**Plan:**

Before starting this project, we came together as a group and outlined the general plan for our chatbot prior to the first review. We decided establishing a basic chatbot that takes input with a simple working design is the best. From that we could build on for future reviews.

For this review, our aim is to create a template of what our chat-bot will look like and have the chat-bot preform basic functionalities

In order to put our plan in action we needed to research more about chatbots and its functionalities. After much research and more understanding of what we wanted to implement for our chatbot we decided that using HTML, CSS and JavaScript is our best option of programming languages. We chose JavaScript because it easy to implement a GUI with JavaScript.

After deciding what we wanted to use to code the chat-bot, we used:

HTML – to create a webpage of the chat-bot that acts as the GUI

CSS – to style and design our GUI.

JavaScript:

* To output messages from the chat-bot as response to general user input.
* To Process the user input.
* To Communicate with the OpenWeatherMap API.
* To process the data that we receive from the API.
* To update the interface accordingly.

This plan is quite reserved and for future reviews we hope to add to the complexity of our GUI and chat-bot. By starting with a simple but well-built chat-bot, we could create a solid base from which we can build upon.

**Modifications:**

For this review we have made a chat bot from scratch. We have developed it to resemble the image we are expecting to have by the end of this project.

HTML (Baraah Afana).

Created a webpage that acts the GUI for our chat bot. The webpage consists of:

* A chatBox <div> that holds the messages sent by both the user and bot.
* A sendText <div> that holds the text box for user input and the send button.
* <input type = “text”> which intakes the user input which is used in script.js file.
* A send button used in script.js to process the user input.

CSS (Baraah Afana)

Added style and design to the webpage to make the GUI eye-friendly to the user.

JavaScript (Arthur and Mert)

Script.js implements the functionality of our chat-bot.

* Created a object called clotheRecommendations that stores the clothe recommendations that will be presented to the user depending on the weather in the requested location.
* Created a function that takes in userInput when the user clicks the send button.
* Created a function that takes in userInput when the user presses the enter key on their keyboard.
* getUserInput( ) – gets the user input from the html and returns it as a variable.
* updateChatBox( ) – takes in user message or bot message, updates html; creating a <div> element for the message and then presents it inside the chatBox <div> on the webpage.
* getBotReponse( ) – processes user input and outputs a bot response message accordingly.
* processUserInput( ) – processes user input, finds correct bot response for that input then displays the bot response inside the chatBox on webpage.
* getWeatherDataFromAPI( ) – requests the weather data from the API based on the user’s requested location and returns it as a json format object.
* processesWeatherData( ) – processes the weather data object and re-formats it into a more efficient object that is useful to our program.
* getNumberOfForecastUntilTomorrow( ) – returns the number of forecast from the time the user requests the location until the next day
* getWeatherDataForTheNextThreeDays( ) – gets the weather forecast for the next three days in the requested location of the user and returns an array of the forecasts.

Unit testing (Arthur and Mert)

Made unit tests to ensure that are functions are working as expected to.

**Work log:**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**Updates:**

What we need to work on and update:

* Get the bot send personalized messages as response to additional specific user inputs.
* Display the weather forecast to the user in a side box.
* Separate the functions and objects into classes.
* Implement a state machine framework for the bot.
* Provide precise instructions on what clothing the user should pack on their journey for each designated location.
* Add more CSS style to make GUI more sophisticated.