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

|  |  |
| --- | --- |
| **Student Information** | |
| **Student Name** | Cian Scarborough  David O’Reilly  Deborah Rimei |
| **Student Number** | 3000011  3018591  3015579 |
| **Code Review #** | 1 |

# Project specifications

## Scope:

* The goal of your chatbot is to plan your clothing requirements for a trip that will visit 5 locations in 3 days.
* The bot should be able to review the weather for each of these locations and suggest appropriate clothing for each location for the day that you will visit.
* Your team must indicate which team member wrote each method and document its functionality.
* Each team needs to create a repository for the chatbot and add the lecturer to the project as a member.
* There must be evidence of a testing framework for each feature of the chatbot.
* There must be evidence that all members have both pulled code from the repository and committed changes to the project at each milestone review.

## Extra Features (Optional):

* Adding a GUI for your chatBot.
* Adding audio functionality for your chatBot.

# 

# Plan

## Code Languages

The chatbot will be created using HTML, CSS, PHP and JS.

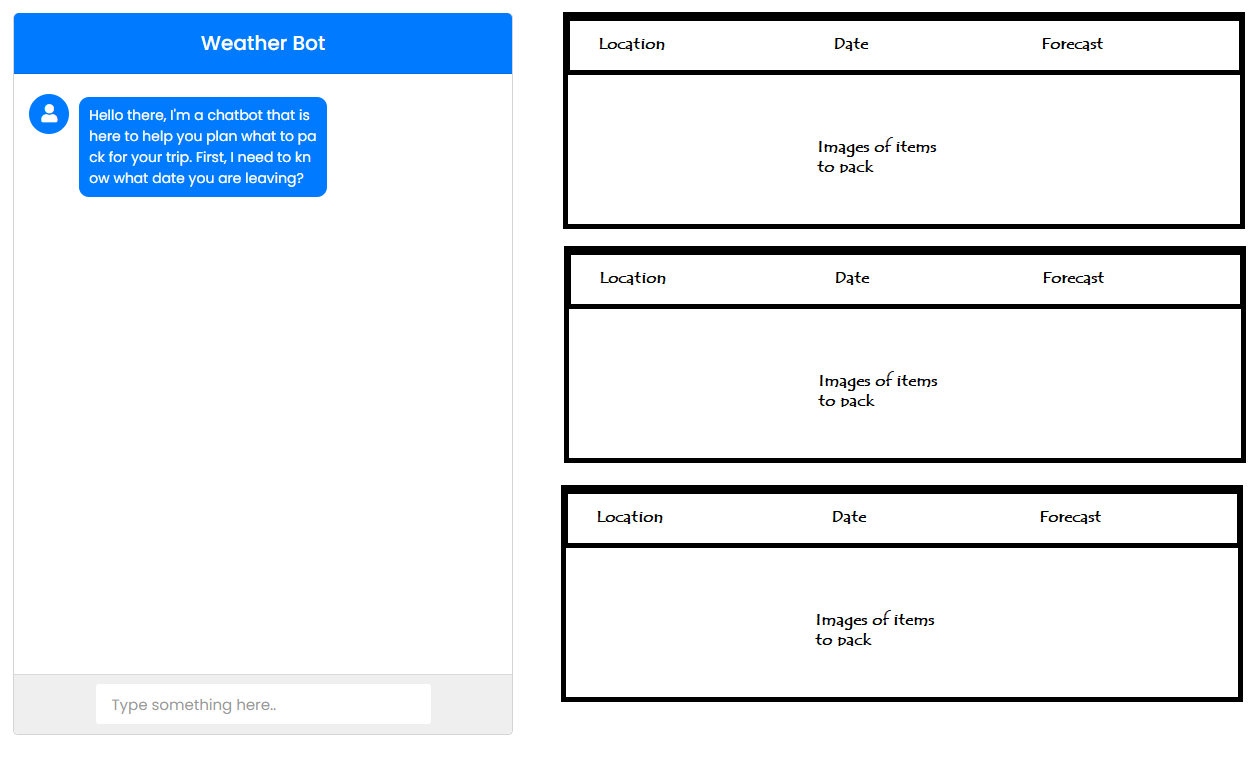
## 

## Design

The chatbot will be developed on three files, “Chatbot.html” for the chat interface, “Style.css” that contains the design and “Message.php” where the data sent from the interface are processed.

### Interface - Chatbot.html

Below shows a rough outline of how the interface will look and work.



1. Bot will prompt the user to provide the data required.
2. As the user answers the data will be passed to the message.php file for processing.
3. Once processed it will be returned and stored as a variable in the chatbot.html file.
4. Once the all data on a location is captured the message.php will contact the api and gather the data on that location.
5. This data will then run though a matrix that decides what clothes should be packed.
6. The data and images of the clothes as processed into the interface.
7. The bot will then prompt the user to repeat the process.

The interface will post the message entered by the user and then process the return message into the interface. The post will be done by jQuery to the message file. Example below.

$(document).ready(function() {

$("#send-btn").on("click", function() {

//Present value on interface.

$value = $("#data").val();

$msg = '<div class="user-inbox inbox"><div

class="msg-header"><p>' + $value + '</p></div></div>';

$(".form").append($msg);

$("#data").val('');

//Pass value to message.php

// start ajax code

$.ajax({

url: 'message.php',

type: 'POST',

data: 'text=' + $value,

//Process return.

success: function(result) {

$reply = '<div class="bot-inbox inbox">

<div class="icon"> <i class="fas fa-user">

</i></div>

<div class="msg-header">' + result + '</div></div>';

$(".form").append($replay);

// when chat goes down the scroll bar automatically comes to bottom

$(".form").scrollTop($(".form")[0].scrollHeight);

}

});

});

});

### Process data - Messages.php

This file will process the data sent from the interface and return a message.  
  
Currently we see us requiring the following methods to process the data.

First, we will capture what the message being passed is.

$getMesg = $\_POST['text'];

ProcessText{} //Decides what the data being sent is.

ProcessDate{} //Takes a date from the string and formats to send.

ProcessLocation{} //Takes the location from the string and formats to send.

QueryAPI{} //

Error{}

Contributors:

Cian Scarborough 3000011

David O’Reilly 3018591

Deborah Rimei 3015579