**Progress Report 2**

**Backlog and changes report:**

**<https://github.com/Darylwanji/COSC4P02---Software-Engineering-2/projects/1>**

**Sprint planning meeting:** Sprint goal

* **Front-end:** Host website on AWS, create to API gateway and setup lambda function for future back-end logic.
* **Back-end:** Basic Version of AI for chat-bot with No UI and Testing, scraping Data for AI, writing unit test and error handling for programs.

**Scrum meeting:** Front-end

* Created and tested the API gateway and dummy Lambda function.
* Updated HTML version by adding FAQ responses.
* Upload the website files to an AWS S3 bucket and set appropriate access control list (ACL) permissions.

|  |  |  |
| --- | --- | --- |
| **Test** | **Input** | **Output** |
| **Tested lambda function output**  Tested that the lambda function output was working using testing feature in AWS. | Invoked our function with a template test event.  {  "key1": "value1",  "key2": "value2",  "key3": "value3"  } | Response  "Hello World!"  Function Logs  START RequestId: 2b41ce3b-f062-4494-b561-397354cce304 Version: $LATEST.  END RequestId: 2b41ce3b-f062-4494-b561-397354cce304  REPORT RequestId: 2b41ce3b-f062-4494-b561-397354cce304 Duration: 11.77 ms Billed Duration: 12 ms Memory Size: 128 MB Max Memory Used: 64 MB Init Duration: 221.10 ms  Request ID  2b41ce3b-f062-4494-b561-397354cce304 |
| **Tested lambda function when calling from JavaScript.**  Tested how the JavaScript chatbot window function retrieves the lambda return value. | Setup the return function in our JavaScript to be printed on the console on the browser. | First output was “undefined”. But after AJAX call the result became expected value “Hello world”. |
| **Tested HTML + CSS + JavaScript files on AWS S3**  Tested the permissions on website files contained in the S3 bucket and confirmed website displayed correctly. | Uploaded HTML + CSS + JavaScript files on AWS S3 bucket with following input access control list (ACL) permissions:  Object owner: Read, Write.  Everyone (public access): Read, Read  Authenticated users' group (anyone with an AWS account): Read, Read. | Working website that can be accessed by anyone with the following URL: **<https://ourwebsite.s3.us-east-2.amazonaws.com/Mobi.html>** |
| **Tested python web-scraping code**  Tested the web scrape code whether is work as expected and return the correct output. | Tested the program with a python unittest library.  Created test methods and classes that are inherited from the web scrape code.  Using assertEqual()  And assertIsNotNone()  Method to test the code. | All the methods passed the test,  The test results are True. |
| **Tested Python AI Bot**  Tested various inputs from the pattern for each query. | Uniitest was used to test each input from each tag and compare the results with any from the responses section. | All methods Passed the test.  Test results: OK. |

**Scrum meeting:** Back-end

* Created python scripts to scrape information from both brocku.ca and niagara2022games.ca.
* Wrote Test classes for scrape scripts.

**Sprint deliverables**

* **Front-end:**
  + API Gateway Invoke URL:

<https://4r6bkh99fk.execute-api.us-east-2.amazonaws.com/dev/chat>

* + Website endpoint: **<https://ourwebsite.s3.us-east-2.amazonaws.com/Mobi.html>**
  + Lambda function ARN:

***arn: aws: lambda:us-east-2:692749162245: function:getChatbotAnswer***

* **Backend**:
  + Translated a portion of the scraped information into a .json file.
  + All the python web scraping code that scraped information from brocku.ca and niagara2022games.ca

**Sprint review and retrospect:**

* **Front-end:**
  + Need to change HTML hyperlink tags to a distinct colour for visibility.
  + Setting up the AWS lambda function was quite challenging but managed to get it working with Node.js 14.x runtime for JavaScript code. This may need to change if chatbot logic is implemented in python.
  + **Future release:** 
    - Connect front-end user input to backend lambda function logic.
    - Retrieve back-end response from lambda function, via API gateway and display it to the user.
    - Add FAQ for Niagara games
* **Backend:**
  + Scraping information from Brocku.ca was challenging as the web pages that were formatted the same often needed to be checked manually to see if the correct information was scraped and occasionally needed the creation of new or specific scraping to get the correct information.
  + **Future release:** 
    - Need to combine all python scrapers (brocku.ca and niagara2022games.ca) into one.
    - We need to convert all scraped data into a .json file.