Deliverable #2

Group 27

Henry Sazo, Hugo Chacon, Neil Verma, Alice Dao

Table of Contents

Group 27 Information

Authors
Henry Sazo [hsazo@usc.edu]
Hugo Chacon [chaconac@usc.edu]
Neil Verma [neilverm@usc.edu]
Alice Dao [alicedao@usc.edu]

Tasks and Updates

1. The developers should choose their roles and responsibilities for the project from: scrum master, developers, infrastructure.
    a. Status: Completed
        i. Scrum Master: Alice
        ii. Developers: Henry, Neil
        iii. Infrastructure: Hugo
2. The developers should create a working repository.
    a. Status: Completed
        i. We have a working repository and it is shared with all of our team members and stakeholder.
3. The developers should create a pipeline for the project.
    a. Status: In Progress
        i. We are currently working on setting up the Azure instance, and it is on track to be completed by the end of the sprint.
4. The developers should create a Reactjs/ .NET template project.
    a. Status: In Progress
        i. Our developers have completed laying out the template for the frontend and backend.
        ii. The frontend was able to mimic the version of the website the stakeholders showed us using Reactjs and Bootstrap. The frontend is 25% complete and this task is on track to be completed by the end of the sprint.
        iii. The backend has been testing sending dummy data to the front end.
        iv. The developers plan to meet and connect the frontend to the backend sometime soon.
5. The developers should research how to render REST and WMTS data in Reactjs.
    a. Status: In Progress
        i. All of our group members have been doing more research on the project as a whole and how to render REST and WMTS data in Reactjs.

Source Code

Frontend:

PROVIDED AS A ZIP FILE ON UPLOAD (OR EMAIL)

Backend:

```
using Microsoft.AspNetCore.Mvc;
using System.Net.Http;
using System.Threading.Tasks;

namespace MyApplication.Controllers;

public class MyApiController : ControllerBase
{
private readonly IHttpClientFactory _httpClientFactory;

public MyApiController(IHttpClientFactory httpClientFactory)
{
_httpClientFactory = httpClientFactory;
}

[HttpGet]
public async Task<IActionResult> GetExternalData()
{
var client = _httpClientFactory.CreateClient();
var response = await
client.GetAsync("https://sampleserver6.arcgisonline.com/arcgis/rest/services/Earthquakes_Since1970/MapServer/0/query?where=1%3D1&outFields=*&outSR=4326&f=json");

if (response.IsSuccessStatusCode)
{
var content = await response.Content.ReadAsStringAsync();
Console.WriteLine("content");
return Ok(content);
}

return StatusCode(500, "Error accessing external service");
}
}
```