CAMPUS EVENT MANAGEMENT SYSTEM

NAME: SHAIK RAQUEEBUL ISLAM

Phase 7: Integration & External Access

Objective

To demonstrate Salesforce's ability to integrate with external systems, we implemented a real-time API integration. The Campus Event Management System connects to an external REST API endpoint to fetch live data (e.g., event location weather or external event info) and displays it directly within the Salesforce UI using Lightning Web Components (LWC).

Integration Strategy

- An **outbound REST API callout** was designed.
- An **LWC** initiates the request → calls an **Apex class** → Apex performs an HTTP request to an external endpoint → returns the JSON response to the LWC → UI renders the results.
- The integration showcases **Salesforce's external connectivity** and **real-time API consumption**.

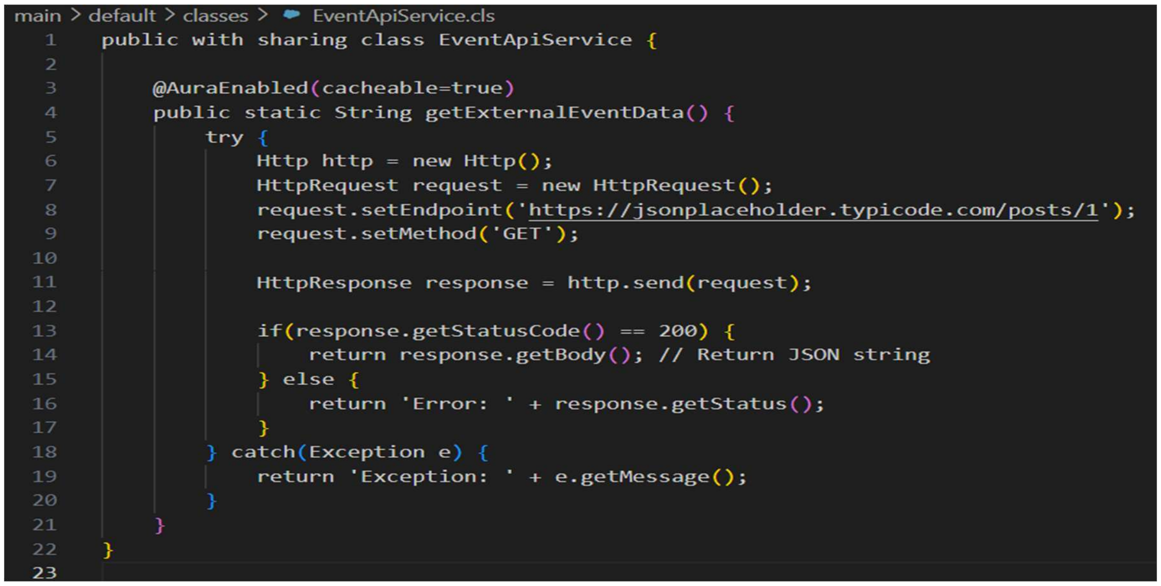
Endpoint Configuration

1. **Initial Approach (Named Credential):**
   - We configured an External Credential with **No Authentication**.
   - Created a **Named Credential** for the endpoint https://jsonplaceholder.typicode.com.
   - The Named Credential was referenced in Apex for secure callouts.
2. **Troubleshooting & Pivot:**
   - Due to restrictions in the developer org, Named Credential calls returned authorization errors.
3. **Final Approach (Remote Site Settings):**

- We configured **Setup → Security → Remote Site Settings**.
- Added the external site https://jsonplaceholder.typicode.com to whitelist the domain for Apex callouts.

Backend Development (Apex Callout)

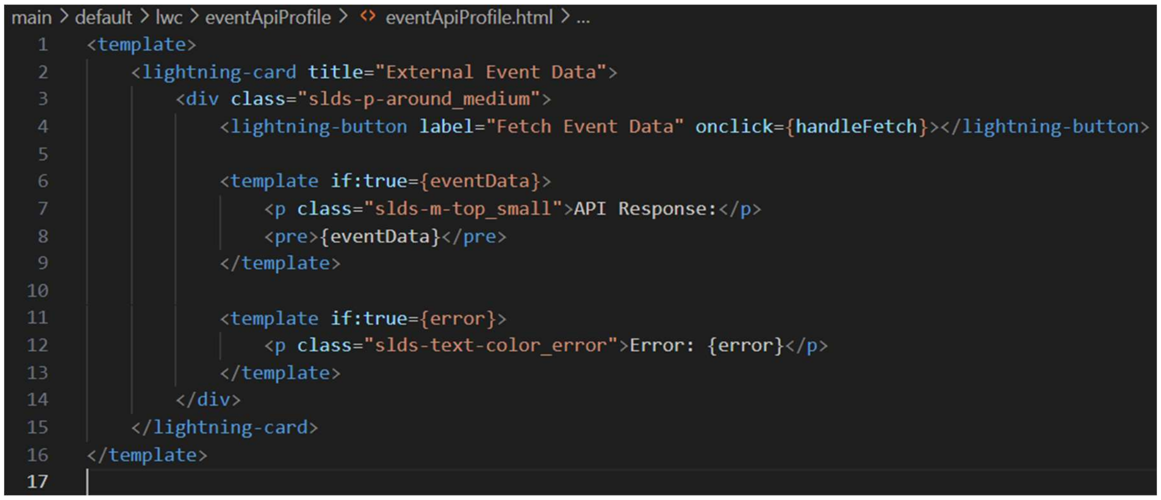
- The EventApiService Apex class is responsible for handling outbound REST API callouts from Salesforce. It exposes a method getExternalEventData, annotated with @AuraEnabled, which makes it accessible to Lightning Web Components. This method sends a GET request to the external API https://jsonplaceholder.typicode.com/posts/1, retrieves the JSON response, and returns it to the frontend for display. The logic is wrapped in error handling to ensure smooth execution, returning meaningful error messages if the API call fails. This implementation highlights Salesforce's capability to integrate with external systems in real time, extending the functionality of the Campus Event Management System.


main > default > classes > EventApiService.cls

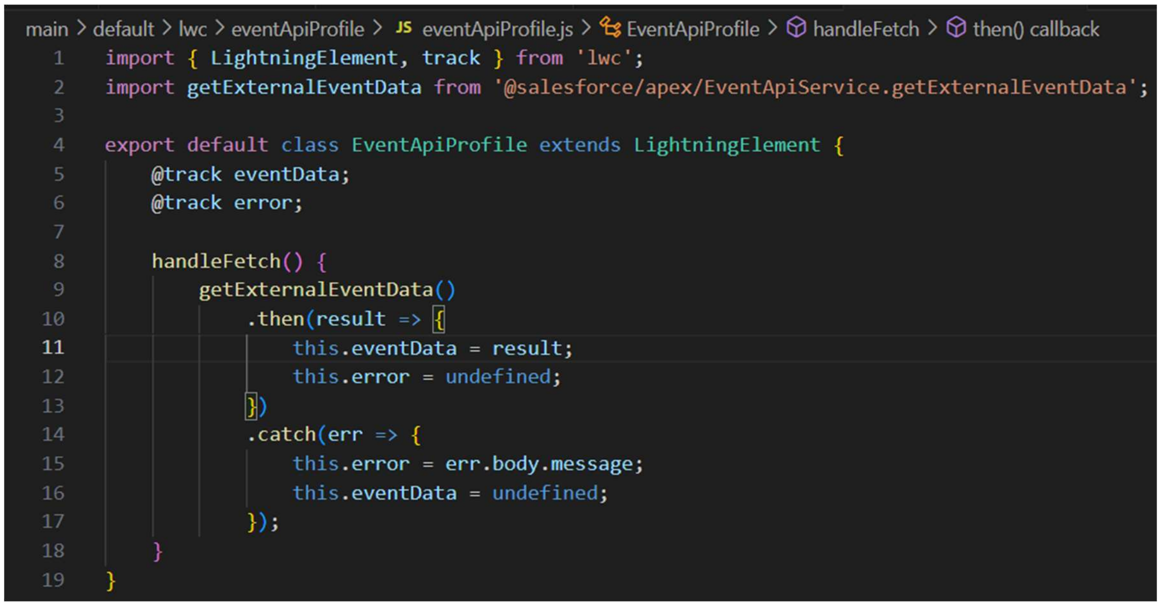
```
public with sharing class EventApiService {

    @AuraEnabled(cacheable=true)
    public static String getExternalEventData() {
        try {
            Http http = new Http();
            HttpRequest request = new HttpRequest();
            request.setEndpoint('https://jsonplaceholder.typicode.com/posts/1');
            request.setMethod('GET');

            HttpResponse response = http.send(request);

            if(response.getStatusCode() == 200) {
                return response.getBody(); // Return JSON string
            } else {
                return 'Error: ' + response.getStatus();
            }
        } catch(Exception e) {
            return 'Exception: ' + e.getMessage();
        }
    }
}
```

Frontend Development (LWC Integration)

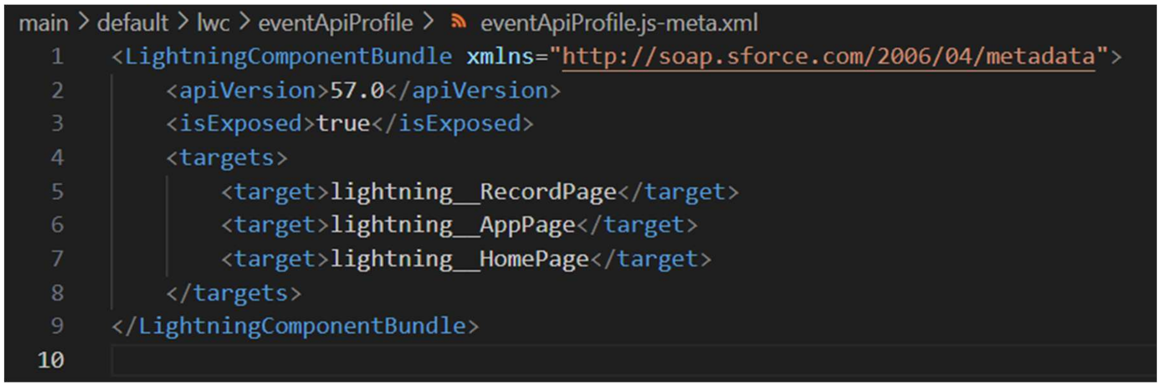
- The eventApiProfile Lightning Web Component (LWC) provides the user interface for interacting with the external API. It includes a button labeled "Fetch Event Data," which, when clicked, calls the Apex method getExternalEventData. The component tracks the API response and displays it dynamically on the page. If an error occurs during the callout, the component shows an error message to the user. The LWC is structured with an HTML template for layout, a JavaScript controller for logic, and a metadata XML file to expose it on Salesforce Home, App, and Record Pages. This frontend integration allows users to view live external data seamlessly within the Salesforce UI, enhancing the Campus Event Management System's interactivity and real-time capabilities.


main > default > lwc > eventApiProfile > eventApiProfile.html > ...

```html
<template>
    <lightning-card title="External Event Data">
        <div class="slds-p-around_medium">
            <lightning-button label="Fetch Event Data" onclick={handleFetch}></lightning-button>

            <template if:true={eventData}>
                <p class="slds-m-top_small">API Response:</p>
                <pre>{eventData}</pre>
            </template>

            <template if:true={error}>
                <p class="slds-text-color_error">Error: {error}</p>
            </template>
        </div>
    </lightning-card>
</template>
```


main > default > lwc > eventApiProfile > eventApiProfile.js > EventApiProfile > handleFetch > then() callback

```javascript
import { LightningElement, track } from 'lwc';
import getExternalEventData from '@salesforce/apex/EventApiService.getExternalEventData';

export default class EventApiProfile extends LightningElement {
    @track eventData;
    @track error;

    handleFetch() {
        getExternalEventData()
            .then(result => {
                this.eventData = result;
                this.error = undefined;
            })
            .catch(err => {
                this.error = err.body.message;
                this.eventData = undefined;
            });
    }
}
```


main > default > lwc > eventApiProfile > eventApiProfile.js-meta.xml

```xml
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>57.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__RecordPage</target>
        <target>lightning__AppPage</target>
        <target>lightning__HomePage</target>
    </targets>
</LightningComponentBundle>
```