## Phase 7: Integration & External Access — Customized Guide

### Objective

Connect Salesforce with external services for:

- **Real-time candidate data fetch** from external APIs.
- **External system integration** (e.g., HR tools, job portals).
- **Secure and scalable architecture** for future integrations.

Your current integration example uses https://jsonplaceholder.typicode.com as a demo API to fetch candidate profiles.

---

### Step 1: Identify Integrations (Integration Catalog)

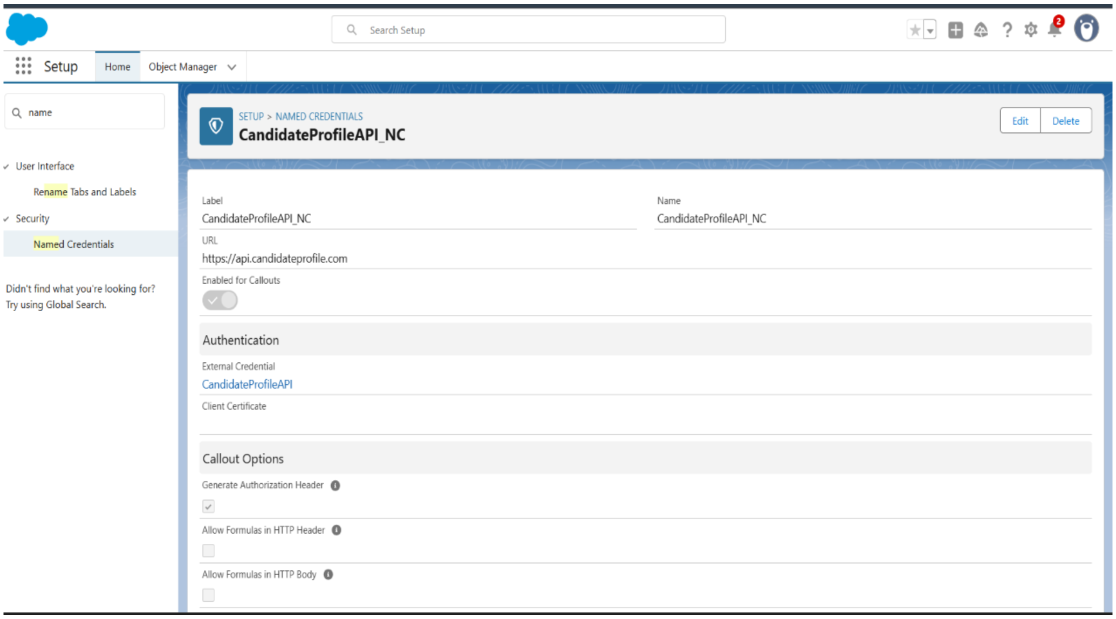
| Integration | Purpose (Sync Direction) | Volume (Calls/min) | Latency SLA | Key Data | Recommended Pattern | Auth Model |
|---|---|---|---|---|---|---|
| Candidate Data API (jsonplaceholder) | Fetch candidate profile details for demo | Low (1–10) | < 3 sec | Candidate ID, name, contact info, job title | **Synchronous** direct callout | **No Auth** (Public API) |
| Future HR System | Push selected candidate data | Medium (10–50) | < 5 sec | Candidate record, job role, department | **Asynchronous** (Queueable/Platform Events) | **Named Principal** |
| Job Portal | Fetch external job postings | Low (5–20) | < 3 sec | Job ID, description, posting date | **Synchronous** | **Named Principal** |

---

### Step 2: Configure Endpoint Access

### Option A — Preferred: Named Credential (Modern & Secure)

1. Go to **Setup → Named Credentials → New**.
2. Configure:
   - **Label**: CandidateProfileAPI
   - **URL**: https://jsonplaceholder.typicode.com
   - **Identity Type**: Named Principal
   - **Authentication Protocol**: No Authentication (public API demo)
3. Save.



CandidateProfileAPI_NC

Label: CandidateProfileAPI_NC

Name: CandidateProfileAPI_NC

URL: https://api.candidateprofile.com

Enabled for Callouts

Authentication

External Credential: CandidateProfileAPI

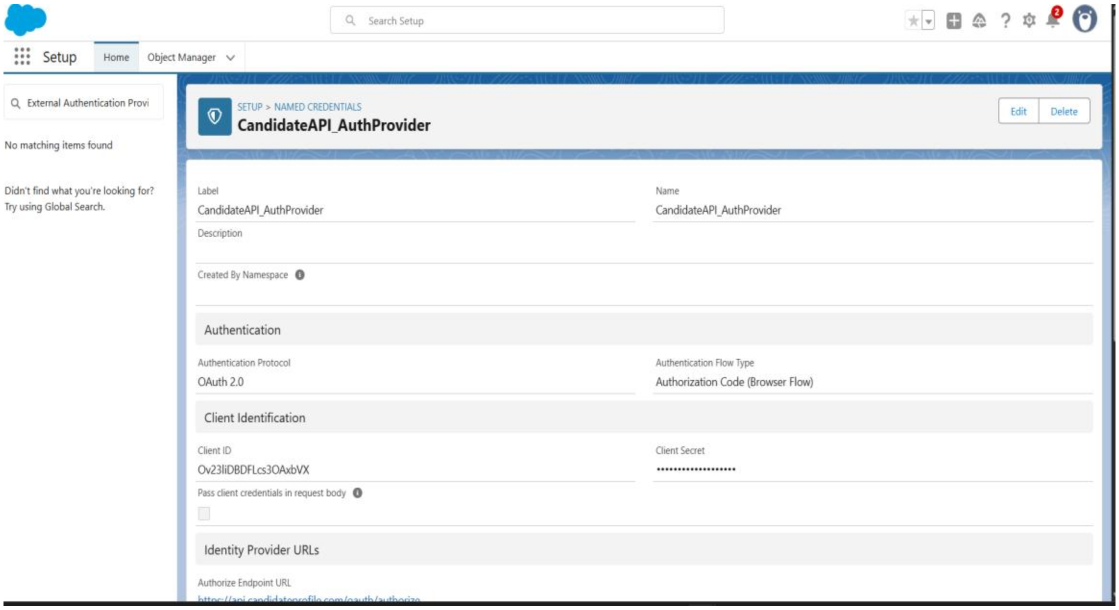
Client Certificate

Callout Options

Generate Authorization Header: ☑

Allow Formulas in HTTP Header: □

Allow Formulas in HTTP Body: □



CandidateAPI_AuthProvider

Label: CandidateAPI_AuthProvider

Name: CandidateAPI_AuthProvider

Description

Created By Namespace

Authentication

Authentication Protocol: OAuth 2.0

Authentication Flow Type: Authorization Code (Browser Flow)

Client Identification

Client ID: Ov23liDBDFLcs3OAxbVX

Client Secret: ••••••••••••••••••••

Pass client credentials in request body: □

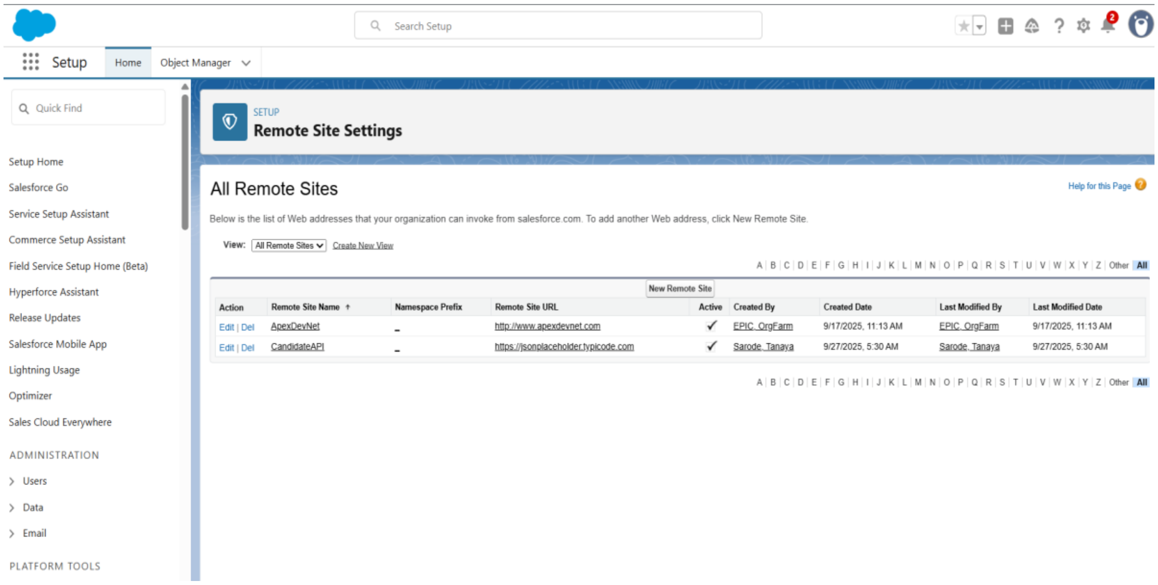
Identity Provider URLs

Authorize Endpoint URL: https://api.candidateprofile.com/oauth/authorize

### Option B — Remote Site Setting (Fallback)

*Use this only if Named Credential gives permission issues (like in your org).*

1. Go to **Setup** → **Remote Site Settings** → **New Remote Site**.
2. Fill:
   - **Remote Site Name**: CandidateAPI
   - **URL**: https://jsonplaceholder.typicode.com
3. Save.



All Remote Sites

Below is the list of Web addresses that your organization can invoke from salesforce.com. To add another Web address, click New Remote Site.

View: All Remote Sites Create New View

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

New Remote Site

| Action | Remote Site Name ↑ | Namespace Prefix | Remote Site URL | Active | Created By | Created Date | Last Modified By | Last Modified Date |
|---|---|---|---|---|---|---|---|---|
| Edit \| Del | ApexDevNet | _ | http://www.apexdevnet.com | ✓ | EPIC, OrgFarm | 9/17/2025, 11:13 AM | EPIC, OrgFarm | 9/17/2025, 11:13 AM |
| Edit \| Del | CandidateAPI | _ | https://jsonplaceholder.typicode.com | ✓ | Sarode, Tanaya | 9/27/2025, 5:30 AM | Sarode, Tanaya | 9/27/2025, 5:30 AM |

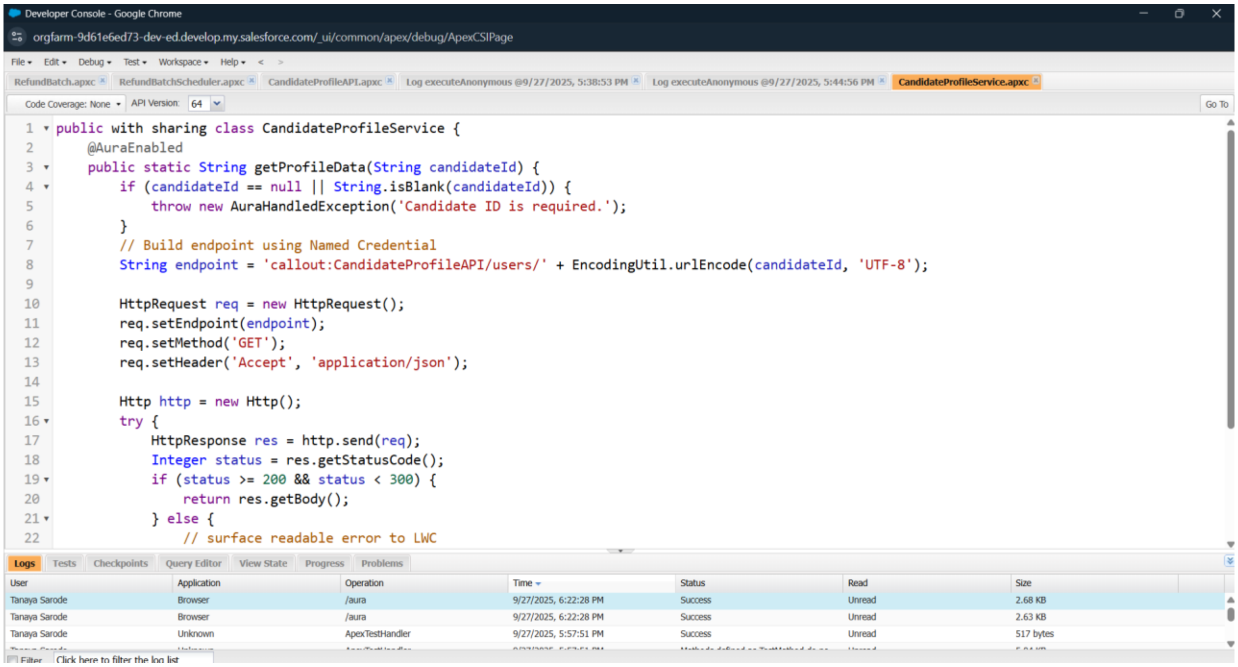
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

---

### Step 3: Build Backend Logic (Apex Callout)

Create an Apex class for the outbound REST API callout.

### Example Apex Code

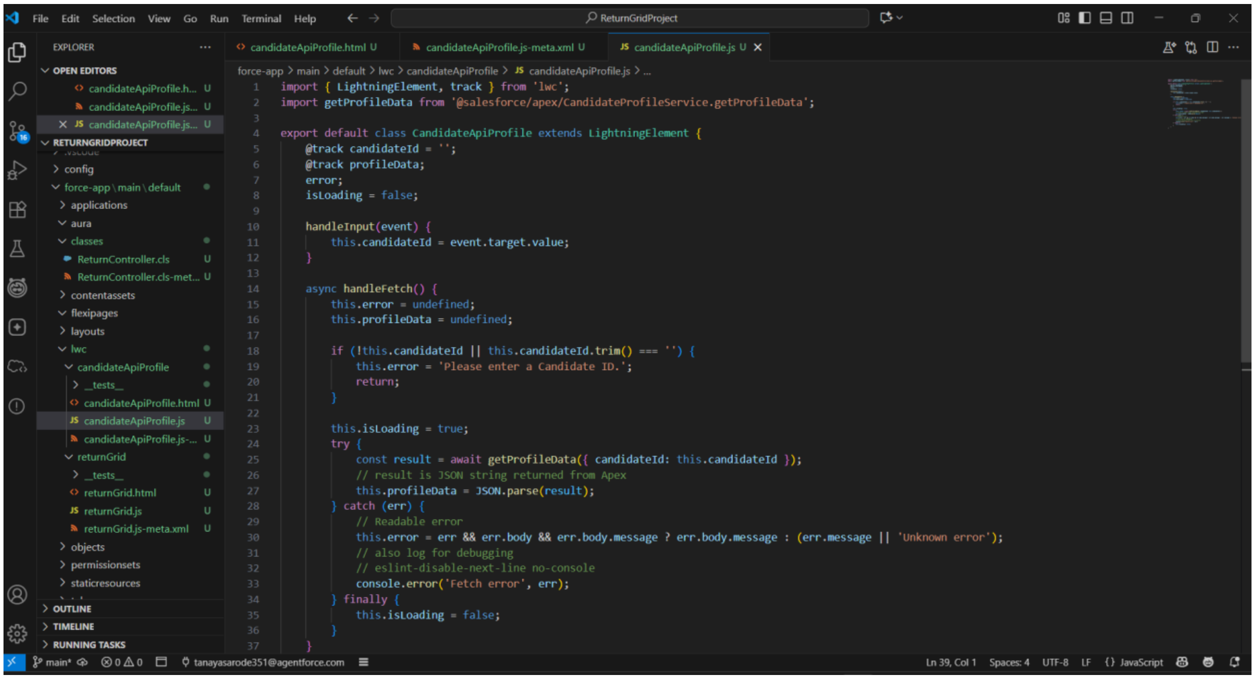

Developer Console - Google Chrome
orgfarm-9d61e6ed73-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File
Edit
Debug
Test
Workspace
Help
RefundBatch.apxc
RefundBatchScheduler.apxc
CandidateProfileAPI.apxc
Log executeAnonymous @9/27/2025, 5:38:53 PM
Log executeAnonymous @9/27/2025, 5:44:56 PM
CandidateProfileService.apxc
Code Coverage: None
API Version: 64
Go To
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Filter
Click here to filter the log list

```
public with sharing class CandidateProfileService {
    @AuraEnabled
    public static String getProfileData(String candidateId) {
        if (candidateId == null || String.isBlank(candidateId)) {
            throw new AuraHandledException('Candidate ID is required.');
        }
        // Build endpoint using Named Credential
        String endpoint = 'callout:CandidateProfileAPI/users/' + EncodingUtil.urlEncode(candidateId, 'UTF-8');

        HttpRequest req = new HttpRequest();
        req.setEndpoint(endpoint);
        req.setMethod('GET');
        req.setHeader('Accept', 'application/json');

        Http http = new Http();
        try {
            HttpResponse res = http.send(req);
            Integer status = res.getStatusCode();
            if (status >= 200 && status < 300) {
                return res.getBody();
            } else {
                // surface readable error to LWC
```

| User | Application | Operation | Time | Status | Read | Size |
|---|---|---|---|---|---|---|
| Tanaya Sarode | Browser | /aura | 9/27/2025, 6:22:28 PM | Success | Unread | 2.68 KB |
| Tanaya Sarode | Browser | /aura | 9/27/2025, 6:22:28 PM | Success | Unread | 2.63 KB |
| Tanaya Sarode | Unknown | ApexTestHandler | 9/27/2025, 5:57:51 PM | Success | Unread | 517 bytes |

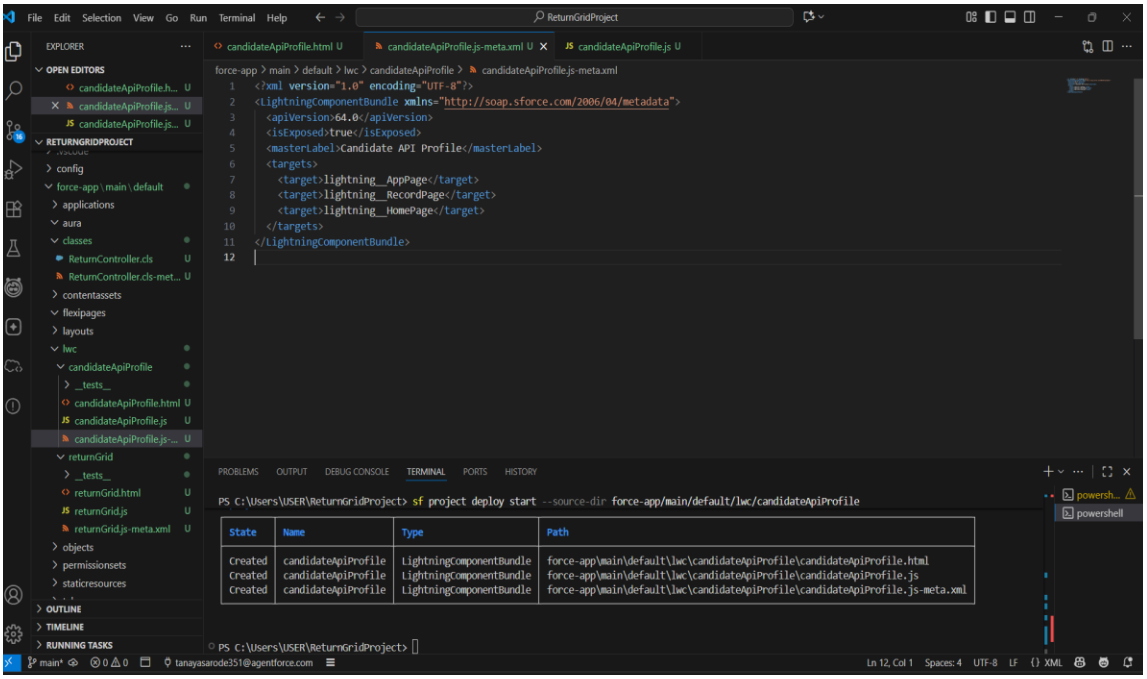
## Step 4: Build LWC (Frontend Integration)

1. **LWC folder structure**
2. candidateApiProfile/
3. candidateApiProfile.html
4. candidateApiProfile.js
5. candidateApiProfile.js-meta.xml
6. candidateApiProfile.css
7. **LWC JS Code Example**



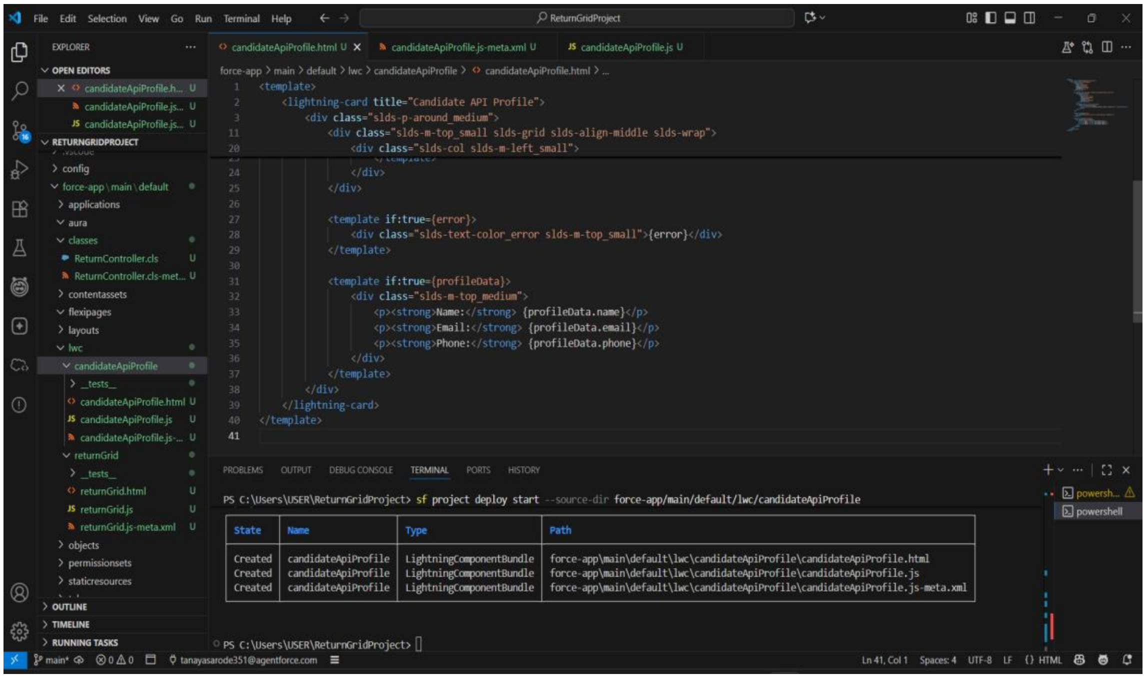
```javascript
import { LightningElement, track } from 'lwc';
import getProfileData from '@salesforce/apex/CandidateProfileService.getProfileData';

export default class CandidateApiProfile extends LightningElement {
    @track candidateId = '';
    @track profileData;
    error;
    isLoading = false;

    handleInput(event) {
        this.candidateId = event.target.value;
    }

    async handleFetch() {
        this.error = undefined;
        this.profileData = undefined;

        if (!this.candidateId || this.candidateId.trim() === '') {
            this.error = 'Please enter a Candidate ID.';
            return;
        }

        this.isLoading = true;
        try {
            const result = await getProfileData({ candidateId: this.candidateId });
            // result is JSON string returned from Apex
            this.profileData = JSON.parse(result);
        } catch (err) {
            // Readable error
            this.error = err && err.body && err.body.message ? err.body.message : (err.message || 'Unknown error');
            // also log for debugging
            // eslint-disable-next-line no-console
            console.error('Fetch error', err);
        } finally {
            this.isLoading = false;
        }
    }
```

8. **LWC HTML Example**



```xml
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>64.0</apiVersion>
  <isExposed>true</isExposed>
  <masterLabel>Candidate API Profile</masterLabel>
  <targets>
    <target>lightning__AppPage</target>
    <target>lightning__RecordPage</target>
    <target>lightning__HomePage</target>
  </targets>
</LightningComponentBundle>
```

```
PS C:\Users\USER\ReturnGridProject> sf project deploy start --source-dir force-app/main/default/lwc/candidateApiProfile

 State    Name                 Type                     Path
 Created  candidateApiProfile  LightningComponentBundle force-app\main\default\lwc\candidateApiProfile\candidateApiProfile.html
 Created  candidateApiProfile  LightningComponentBundle force-app\main\default\lwc\candidateApiProfile\candidateApiProfile.js
 Created  candidateApiProfile  LightningComponentBundle force-app\main\default\lwc\candidateApiProfile\candidateApiProfile.js-meta.xml

PS C:\Users\USER\ReturnGridProject>
```

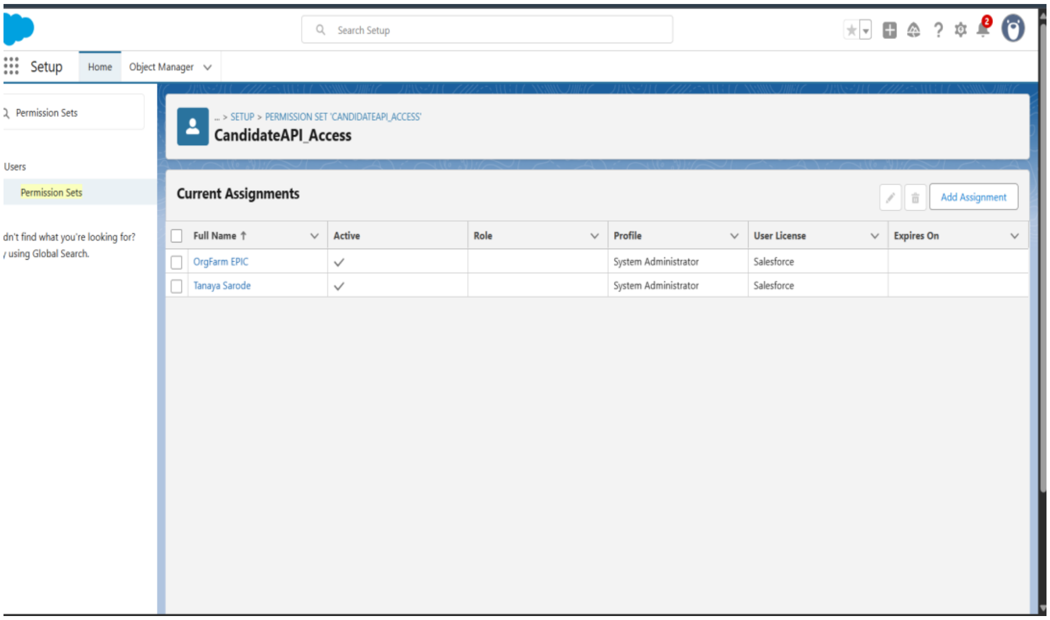


```html
<template>
    <lightning-card title="Candidate API Profile">
        <div class="slds-p-around_medium">
            <div class="slds-m-top_small slds-grid slds-align-middle slds-wrap">
                <div class="slds-col slds-m-left_small">
                </div>
            </div>

            <template if:true={error}>
                <div class="slds-text-color_error slds-m-top_small">{error}</div>
            </template>

            <template if:true={profileData}>
                <div class="slds-m-top_medium">
                    <p><strong>Name:</strong> {profileData.name}</p>
                    <p><strong>Email:</strong> {profileData.email}</p>
                    <p><strong>Phone:</strong> {profileData.phone}</p>
                </div>
            </template>
        </div>
    </lightning-card>
</template>
```

```
PS C:\Users\USER\ReturnGridProject> sf project deploy start --source-dir force-app/main/default/lwc/candidateApiProfile

| State   | Name                | Type                     | Path                                                                        |
|---------|---------------------|--------------------------|-----------------------------------------------------------------------------|
| Created | candidateApiProfile | LightningComponentBundle | force-app\main\default\lwc\candidateApiProfile\candidateApiProfile.html     |
| Created | candidateApiProfile | LightningComponentBundle | force-app\main\default\lwc\candidateApiProfile\candidateApiProfile.js       |
| Created | candidateApiProfile | LightningComponentBundle | force-app\main\default\lwc\candidateApiProfile\candidateApiProfile.js-meta.xml |

PS C:\Users\USER\ReturnGridProject>
```

## Step 5: Security & Permissions

- **Profiles / Permission Sets**:
    - Grant the Apex class CandidateProfileService access to relevant users.
    - Ensure the LWC is visible on assigned Lightning pages.



... > SETUP > PERMISSION SET 'CANDIDATEAPI_ACCESS'

CandidateAPI_Access

Current Assignments

| □ | Full Name ↑ | Active | Role | Profile | User License | Expires On |
|---|---|---|---|---|---|---|
| □ | OrgFarm EPIC | ✓ | | System Administrator | Salesforce | |
| □ | Tanaya Sarode | ✓ | | System Administrator | Salesforce | |

- **CSP Trusted Sites**:

- Add https://jsonplaceholder.typicode.com to **CSP Trusted Sites** to avoid browser security issues.

---

**Step 6: Error Handling & Logging**

| Failure Type | Action |
|---|---|
| API Down / 500 Error | Show user-friendly error message in LWC. |
| Timeout (>10 sec) | Retry up to **3 times** with exponential backoff. |
| Invalid Candidate ID | Display "Candidate not found" message. |
| Integration Failure | Create a record in Integration_Log__c object for tracking. |

---

**Step 7: Deployment Checklist**

| Step | Action |
|---|---|
| **1** | Move Apex class, LWC, and related metadata using **Change Set** or **SFDX Deploy**. |
| **2** | Deploy Remote Site Settings or Named Credentials to production. |
| **3** | Test callouts in **production** with correct endpoints. |
| **4** | Verify permissions for target profiles. |
| **5** | Validate monitoring dashboard for Integration logs. |

---

**Future Improvements**

1. **Switch to Named Credentials** when org issues are resolved.
2. Use **Platform Events** for asynchronous integration with HR systems.
3. Add **error dashboards** and automated reconciliation jobs.
4. Secure sensitive APIs with **OAuth 2.0**.