Assessment - 1

NVD - CVE API:

The CVE API is used to easily retrieve information on a single CVE or a collection of CVE from the NVD. Pls refer to the below NVD CVE documentation to get more information.

https://nvd.nist.gov/developers/vulnerabilities

Problem Statement:

1. Consume the CVE information from the CVE API for all the CVE's and store it in a Database of your choice. - (API BaseURL - https://services.nvd.nist.gov/rest/json/cves/2.0)
2. Hint for accessing all the CVE's from API - Through a series of smaller “chunked” responses controlled by an offset startIndex and a page limit resultsPerPage users may page through all the CVE in the NVD.
3. Apply data cleansing & de-duplication, ensure data quality wherever applicable
4. CVE details should be synchronized into Database periodically in batch mode in a specific time period - (This can be full data refresh or incremental refresh for modified data alone)
5. Develop API’s to read & filter the CVE details by below parameters -
    - CVE ID
    - CVE ID’s belogs to a specific year
    - CVE Score (Field to ref - metrics.cvssMetricV2.cvssData.baseScore or metrics.cvssMetricV3.cvssData.baseScore)
    - last Modified in N days
6. Read the API and visualise it in UI using HTML, CSS and Javascript
7. Prepare the API documentations for each operations.
8. Write well defined unit test cases for the functionalities.
9. Code should be clear, vulnerable free & well tested and should follow the best practices and standards.

The first page should contain:

1. The route path should be /cves/list.
2. Read the API and display its results in a table with a "Total Records" count.
3. Include "Results Per Page" below the table, offering options of "10", "50", and "100", with a default selection of "10". Whenever an option is chosen, execute the respective API call to retrieve the records.(Good to have)

4. Add server side “Pagination” functionality (Optional - Added advantage)
5. Add server side “sorting” for “Dates” (Optional - Added advantage)

Sample UI:

CVE LIST

Total Records: 29999

CVE ID	IDENTIFIER	PUBLISHED DATE	LAST MODIFIED DATE	STATUS
CVE-1999-0334	cve@mitre.org	16 Dec 1999	17 Oct 2022	Analyzed
CVE-1999-0334	cve@mitre.org	16 Dec 1999	17 Oct 2022	Modified
CVE-1999-0334	cve@mitre.org	16 Dec 1999	17 Oct 2022	Rejected
CVE-1999-0334	cve@mitre.org	16 Dec 1999	17 Oct 2022	Analyzed
CVE-1999-0334	cve@mitre.org	16 Dec 1999	17 Oct 2022	Modified
CVE-1999-0334	cve@mitre.org	16 Dec 1999	17 Oct 2022	Rejected
CVE-1999-0334	cve@mitre.org	16 Dec 1999	17 Oct 2022	Analyzed
CVE-1999-0334	cve@mitre.org	16 Dec 1999	17 Oct 2022	Modified
CVE-1999-0334	cve@mitre.org	16 Dec 1999	17 Oct 2022	Rejected
CVE-1999-0334	cve@mitre.org	16 Dec 1999	17 Oct 2022	Analyzed

Results per page: 10 ▲

1 - 10 of 29999 records ◀ 1 2 3 4 5 ▶

1. When a row is clicked, navigate to the second page /cves/cve-1999-0334. The second page should include the following:
2. Perform an API call to retrieve the data of the selected CVE and display it in the user interface.

Sample UI:

CVE-1999-0334

Description:

In Solaris 2.2 and 2.3, when fsck fails on startup, it allows a local user with physical access to obtain root access.

CVSS V2 Metrics:

Severity: LOW **Score:** 7.2

Vector String AV:L/AC:L/Au:N/C:C/I:C/A:C

Access Vector	Access Complexity	Authentication	Confidentiality Impact	Integrity Impact	Availability Impact
LOCAL	LOW	NONE	COMPLETE	COMPLETE	COMPLETE

Scores :

Exploitability Score: 3.9

Impact Score: 10

CPE:

Criteria	Match Criteria ID	Vulnerable
cpe:2.3:o:sun:solaris:*:*:x86:*:*:*:*	FEEC0C5A-4A6E-403C-B929-D1EC8B0FE2A8	Yes
cpe:2.3:o:sun:solaris:*:*:x86:*:*:*:*	FEEC0C5A-4A6E-403C-B929-D1EC8B0FE2A8	Yes
cpe:2.3:o:sun:solaris:*:*:x86:*:*:*:*	FEEC0C5A-4A6E-403C-B929-D1EC8B0FE2A8	Yes