# FastAPI CVE Dashboard – Detailed API Documentation & Test Cases

## 1. Introduction

The FastAPI CVE Dashboard is an application built with FastAPI that helps users view and analyze Common Vulnerabilities and Exposures (CVE) data. It fetches CVEs from the NVD API, stores them in a SQLite database using SQLAlchemy, and serves them via a web interface and NVD API.

## 2. Data Model

The application defines two primary models:  
- CVE: Stores vulnerability details like ID, description, severity scores, and vectors.  
- CPE: Represents configuration platforms related to CVEs.

## 3. API Endpoints

* Endpoint: /cves/list

Method: GET

Description: Returns paginated list of CVEs with sorting options.

Parameters: skip (int), limit (int), sort\_by (published\_date/last\_modified\_date), sort\_order (asc/desc)

Response: JSON with 'records' and 'total'

* Endpoint: /cves/{cve\_id}/json

Method: GET

Description: Returns full details of a single CVE in JSON format.

Parameters: cve\_id (str)

Response: Full CVE with metrics and configurations

* Endpoint: /cve-detail/{cve\_id}

Method: GET (UI)

Description: Serves a detailed HTML page for a specific CVE.

Parameters: cve\_id (str)

Response: HTML page rendered via Jinja2

* Endpoint: /cves/year/{year}

Method: GET

Description: Returns all CVEs from a given year.

Parameters: year (int)

Response: List of CVEs

* Endpoint: /cves/score

Method: GET

Description: Returns CVEs filtered by score range.

Parameters: min\_score (float), max\_score (float), skip (int), limit (int)

Response: Paginated CVEs matching score range

* Endpoint: /ui

Method: GET (UI)

Description: Serves the main HTML page with search, sort, pagination.

Parameters: None

Response: HTML page

## 4. Testing

### 4.1 Unit Test Cases

- Test DB connection and model creation  
- Test API response codes for each endpoint  
- Test pagination and sorting logic  
- Test score-based filtering (min\_score, max\_score)  
- Test CVE detail JSON and UI render correctly

### 4.2 Manual Test Scenarios

- Open /ui in browser and verify table loads with pagination  
- Change results per page and verify update  
- Click CVE ID to open detailed view  
- Sort columns and verify order  
- Check CPE data and CVSS vectors are present in detail page  
- Check error handling for invalid CVE ID

## 5. Conclusion

The CVE Dashboard provides an easy and efficient way to track and analyze vulnerabilities using FastAPI. With both REST and web UI, users can interact and inspect details effectively. The architecture supports future extensions such as search, advanced filters, and visualizations.