NVD_CVE_API_Project Documentation

1. Project Overview

The NVD_CVE_API_Project is a Python-based application that retrieves Common Vulnerabilities and Exposures (CVE) data from the National Vulnerability Database (NVD) API, stores it in a database (MySQL or MongoDB), and presents the data to users via a web-based UI. The application enables users to filter, query, and visualize CVE data, offering functionality like pagination, sorting, and detailed views for individual CVEs.

Key Features:

- Fetches CVE data from NVD API.
- Stores CVE data in MySQL.
- Synchronizes data periodically with Celery.
- Provides backend APIs for querying CVE data.
- Offers a web-based UI for filtering and visualization of CVEs.

2. Technology Stack

The system utilizes the following technologies:

- **Backend**: Python, Flask, Celery (for periodic tasks)
- **Database**: MySQL
- **Frontend**: HTML(Fetch API for AJAX)
- **Testing**: Pytest, Unittest
- **Deployment**: Docker (Optional)

3. System Architecture

3.1 Fetching and Storing CVE Data

The system fetches CVE data from the NVD API using the requests module. The fetched data is paginated and stored in a MySQL or MongoDB database. Data cleansing and deduplication are applied to ensure the database contains only relevant and unique CVE data.

3.2 Periodic Data Sync

Periodic data synchronization is implemented using **Celery** with **Redis**. This ensures that the CVE database remains up-to-date with the latest data from the NVD API, either through a full or incremental refresh based on the last modified date.

3.3 Backend APIs

The Flask-based backend exposes RESTful APIs for querying the CVE data. The following endpoints are available:

1. **Get All CVEs**
   - GET /api/cves
   - Returns a list of all stored CVEs.

2. **Get CVE by ID**
   - GET /api/cves/{cve_id}
   - Returns details of a specific CVE based on its ID.
3. **Get CVEs by Year**
   - GET /api/cves?year=2023
   - Filters CVEs based on the specified year.
4. **Get CVEs by Score**
   - GET /api/cves?min_score=7.0
   - Filters CVEs with a CVSS base score above a given threshold.
5. **Get Recently Modified CVEs**
   - GET /api/cve/modified?days=30
   - Fetches CVEs modified in the last N days.

3.4 Frontend UI

The frontend UI is built using **HTML**, **CSS**, and **JavaScript**. The system uses **Flask** to serve templates that display CVE data in a table. The following features are implemented:

- **Pagination**: Limits the number of records displayed per page.
- **Sorting**: Allows sorting of CVE records by date.
- **Clickable Rows**: Each row in the table is clickable, providing detailed information on the CVE when selected.

3.5 Testing and Security

Unit tests are implemented using **pytest** to ensure the system works as expected. The system also employs basic security practices such as input validation and prevention of SQL injections.

---

4. Installation and Setup

To set up and run the project, follow these steps:

4.1 Clone the Repository

bash

CopyEdit

```
git clone https://github.com/gnanithag5/NVD_CVE_API_Project

cd cve-visualizer
```

4.2 Install Dependencies

Install the required Python dependencies using the requirements.txt file:

bash

CopyEdit

pip install -r requirements.txt

4.3 Set Up MySQL Database

1. Modify the database.py file to ensure correct credentials for MySQL connection.
2. Use the mysql_code.sql file to define the database schema.

bash

CopyEdit

# Example command to create the database schema

mysql -u username -p < mysql_code.sql

4.4 Run Fetch CVE Data Script

Run the script fetch_CVE_data.py to fetch CVE data from the NVD API.

bash

CopyEdit

python fetch_CVE_data.py

4.5 Synchronize Data Periodically

Run full_syn.py to synchronize data on a periodic basis.

bash

CopyEdit

python full_syn.py

4.6 Start the API

Run the api_fetch.py file to expose the backend API.

bash

CopyEdit

python api_fetch.py

4.7 Start the Flask Web Application

Run app2.py to start the Flask application and serve the frontend UI.

bash

CopyEdit

python app2.py

4.8 Run Unit Tests

To run the unit tests:

bash

CopyEdit

```
pytest tests/unit_tests.py
```

---

5. Project Structure

bash

CopyEdit

```
NVD_CVE_API_Project/
│── app2.py                 # Main Flask application to integrate the frontend
│── fetch_CVE_data.py       # Fetches CVE data from NVD API
│── full_syn.py             # Synchronizes data periodically
│── api_fetch.py            # API for filtering CVEs
│── database.py             # DB connection logic
│── mysql_code.sql          # Defines database schema
│── requirements.txt        # Dependencies for the project
│
├── templates/              # HTML templates
│   ├── cve_list.html       # Displays CVE list in a table
│   ├── cve_detail.html     # Shows details of a selected CVE
│
├── tests/                  # Contains unit tests
│   └── unit_tests.py       # Unit tests for API & database
│
├── .pytest_cache/
├── CACHEDIR.TAG
├── .gitignore              # Specifies files to ignore in Git
├── README.md               # Project overview & instructions
├── venv/                   # Virtual environment directory
└── v/                      # Placeholder or temporary files
```

---

6. API Documentation

Available Endpoints:

1. **Get All CVEs**

   - GET /api/cves
   - Returns a list of all CVEs stored in the database.
2. **Get CVE by ID**
   - GET /api/cves/{cve_id}
   - Returns details of a specific CVE.
3. **Get CVEs by Year**
   - GET /api/cves?year=2023
   - Filters CVEs based on the year.
4. **Get CVEs by Score**
   - GET /api/cves?min_score=7.0
   - Filters CVEs with a score above a threshold.
5. **Get Recently Modified CVEs**
   - GET /api/cve/modified?days=30
   - Fetches CVEs modified in the last N days.

---

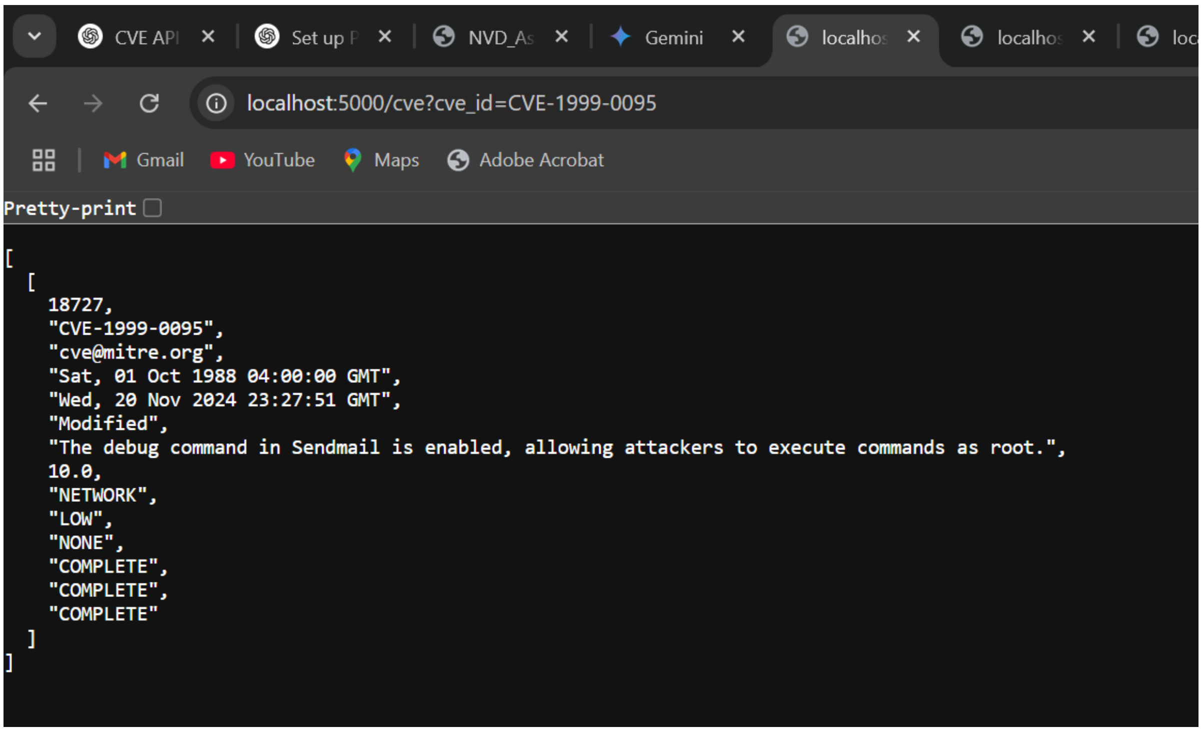
7. Troubleshooting

Common Issues and Solutions:

1. **MySQL connection errors:**
   - Verify that the database credentials in database.py are correct.
   - Ensure that MySQL server is running.
2. **API not running:**
   - Check installed dependencies with pip list.
   - Ensure Flask is installed and the script api_fetch.py is executed properly.

---

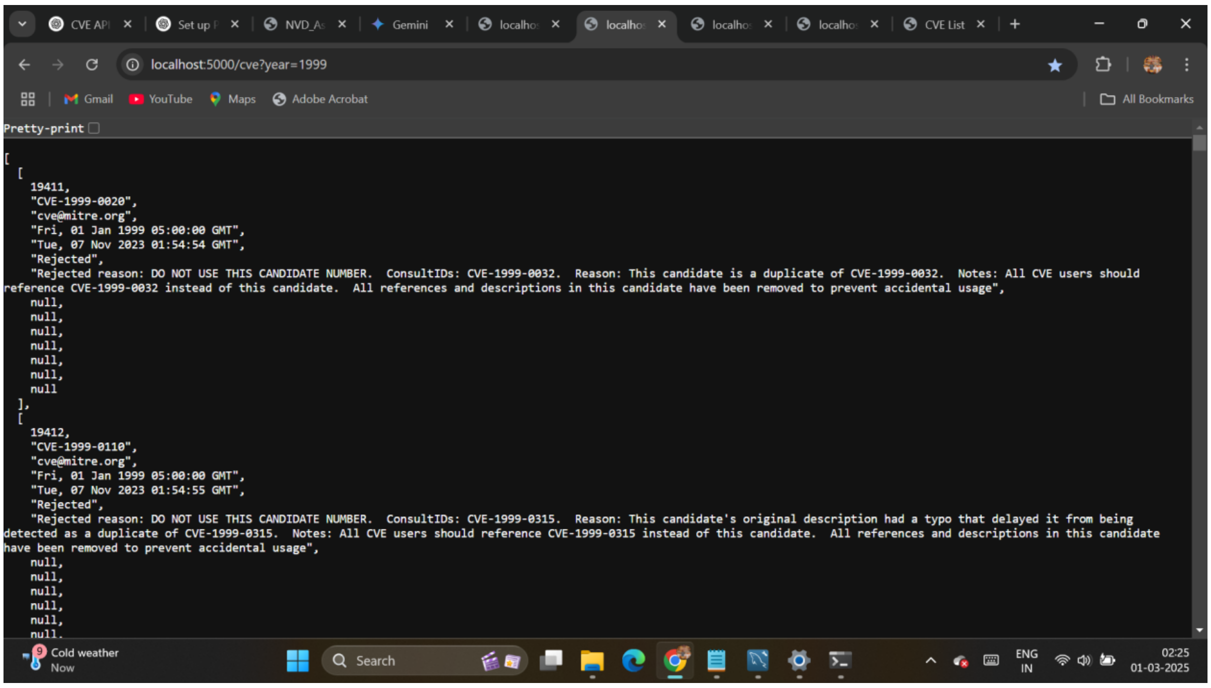
8. Results



```
localhost:5000/cve?cve_id=CVE-1999-0095
```

```
[
  [
    18727,
    "CVE-1999-0095",
    "cve@mitre.org",
    "Sat, 01 Oct 1988 04:00:00 GMT",
    "Wed, 20 Nov 2024 23:27:51 GMT",
    "Modified",
    "The debug command in Sendmail is enabled, allowing attackers to execute commands as root.",
    10.0,
    "NETWORK",
    "LOW",
    "NONE",
    "COMPLETE",
    "COMPLETE",
    "COMPLETE"
  ]
]
```

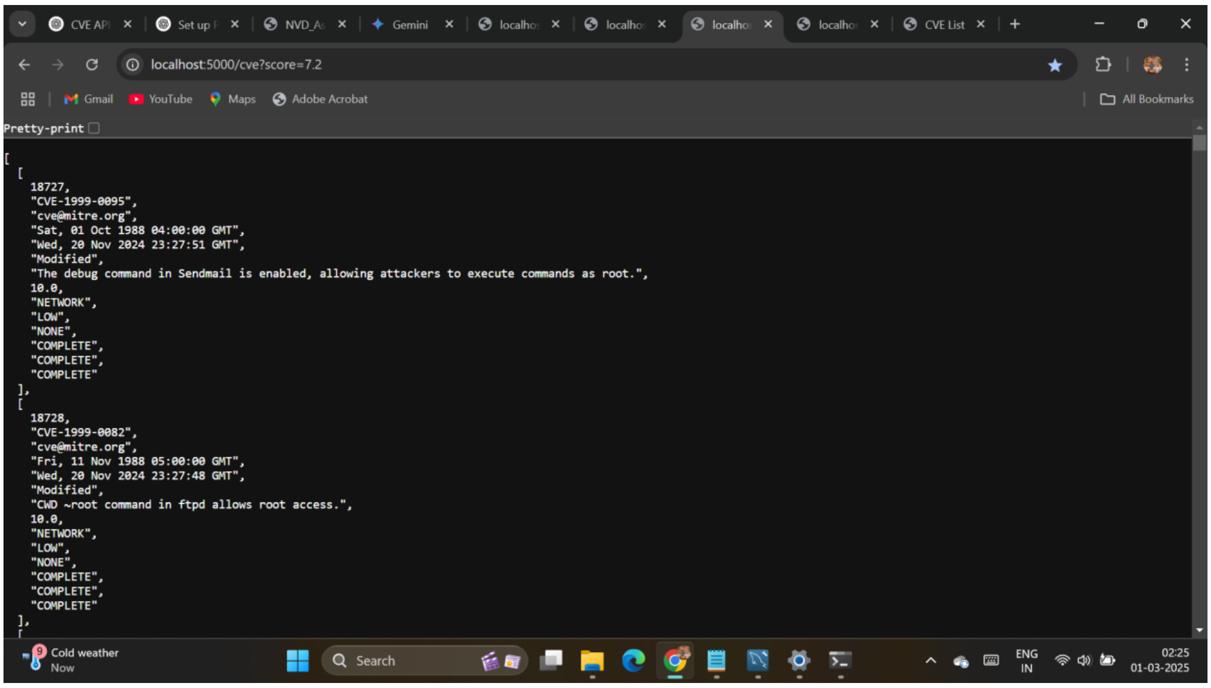
Figure 1: API filtering based on CVE_ID



```
localhost:5000/cve?year=1999
```

```
[
  [
    19411,
    "CVE-1999-0020",
    "cve@mitre.org",
    "Fri, 01 Jan 1999 05:00:00 GMT",
    "Tue, 07 Nov 2023 01:54:54 GMT",
    "Rejected",
    "Rejected reason: DO NOT USE THIS CANDIDATE NUMBER.  ConsultIDs: CVE-1999-0032.  Reason: This candidate is a duplicate of CVE-1999-0032.  Notes: All CVE users should
reference CVE-1999-0032 instead of this candidate.  All references and descriptions in this candidate have been removed to prevent accidental usage",
    null,
    null,
    null,
    null,
    null,
    null,
    null
  ],
  [
    19412,
    "CVE-1999-0110",
    "cve@mitre.org",
    "Fri, 01 Jan 1999 05:00:00 GMT",
    "Tue, 07 Nov 2023 01:54:55 GMT",
    "Rejected",
    "Rejected reason: DO NOT USE THIS CANDIDATE NUMBER.  ConsultIDs: CVE-1999-0315.  Reason: This candidate's original description had a typo that delayed it from being
detected as a duplicate of CVE-1999-0315.  Notes: All CVE users should reference CVE-1999-0315 instead of this candidate.  All references and descriptions in this candidate
have been removed to prevent accidental usage",
    null,
    null,
    null,
    null,
    null,
    null,
```

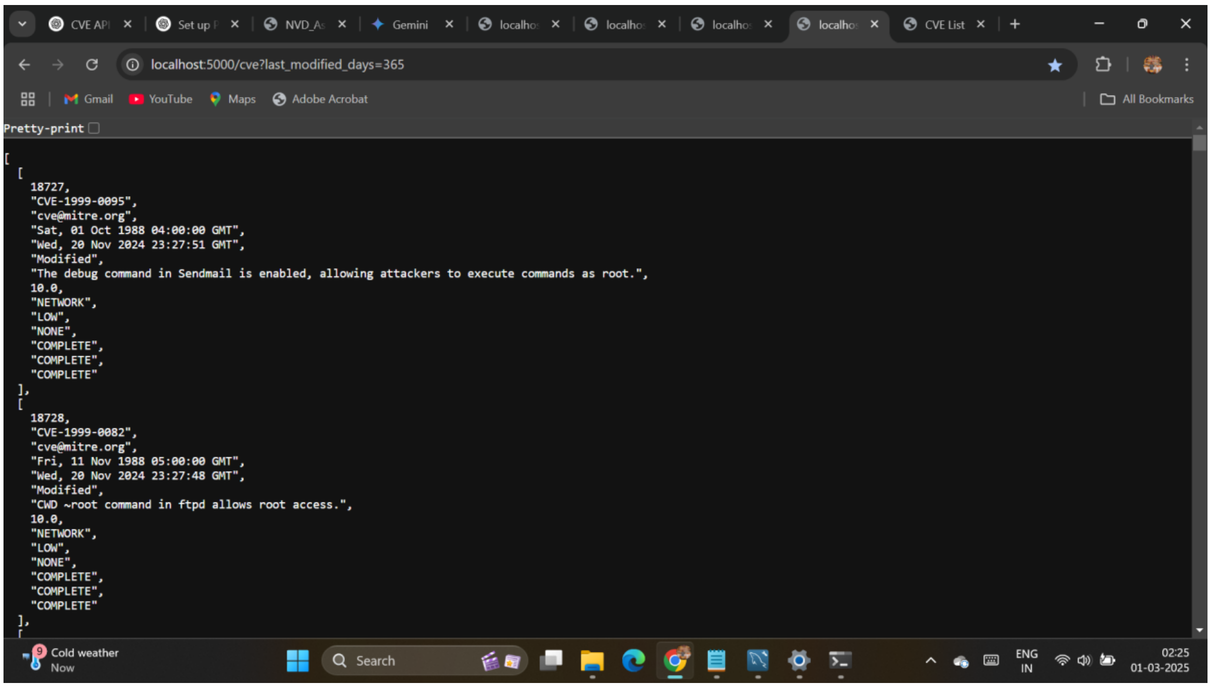
Figure 2: API Filtering based of year


```
localhost:5000/cve?score=7.2

Pretty-print

[
  [
    18727,
    "CVE-1999-0095",
    "cve@mitre.org",
    "Sat, 01 Oct 1988 04:00:00 GMT",
    "Wed, 20 Nov 2024 23:27:51 GMT",
    "Modified",
    "The debug command in Sendmail is enabled, allowing attackers to execute commands as root.",
    10.0,
    "NETWORK",
    "LOW",
    "NONE",
    "COMPLETE",
    "COMPLETE",
    "COMPLETE"
  ],
  [
    18728,
    "CVE-1999-0082",
    "cve@mitre.org",
    "Fri, 11 Nov 1988 05:00:00 GMT",
    "Wed, 20 Nov 2024 23:27:48 GMT",
    "Modified",
    "CWD ~root command in ftpd allows root access.",
    10.0,
    "NETWORK",
    "LOW",
    "NONE",
    "COMPLETE",
    "COMPLETE",
    "COMPLETE"
  ],
  [
```

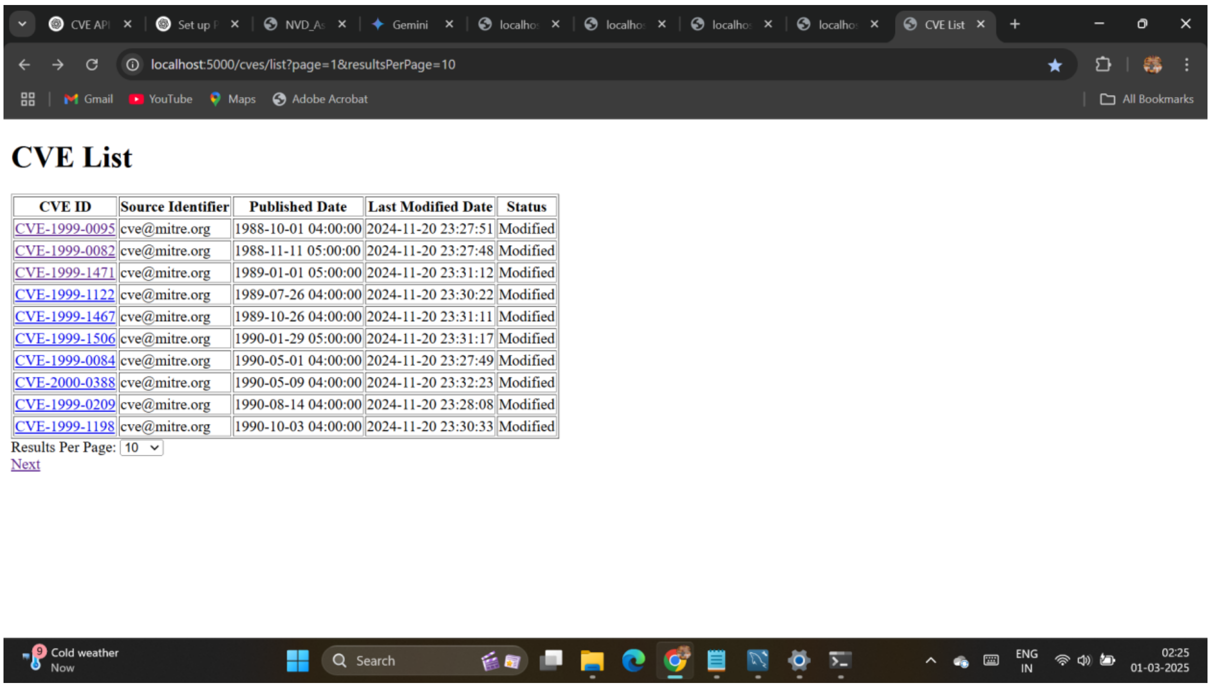
Figure 3: API filtering based on score


```
localhost:5000/cve?last_modified_days=365

Pretty-print

[
  [
    18727,
    "CVE-1999-0095",
    "cve@mitre.org",
    "Sat, 01 Oct 1988 04:00:00 GMT",
    "Wed, 20 Nov 2024 23:27:51 GMT",
    "Modified",
    "The debug command in Sendmail is enabled, allowing attackers to execute commands as root.",
    10.0,
    "NETWORK",
    "LOW",
    "NONE",
    "COMPLETE",
    "COMPLETE",
    "COMPLETE"
  ],
  [
    18728,
    "CVE-1999-0082",
    "cve@mitre.org",
    "Fri, 11 Nov 1988 05:00:00 GMT",
    "Wed, 20 Nov 2024 23:27:48 GMT",
    "Modified",
    "CWD ~root command in ftpd allows root access.",
    10.0,
    "NETWORK",
    "LOW",
    "NONE",
    "COMPLETE",
    "COMPLETE",
    "COMPLETE"
  ],
  [
```

Figure 4: API Filitering based of N number of days modifications made

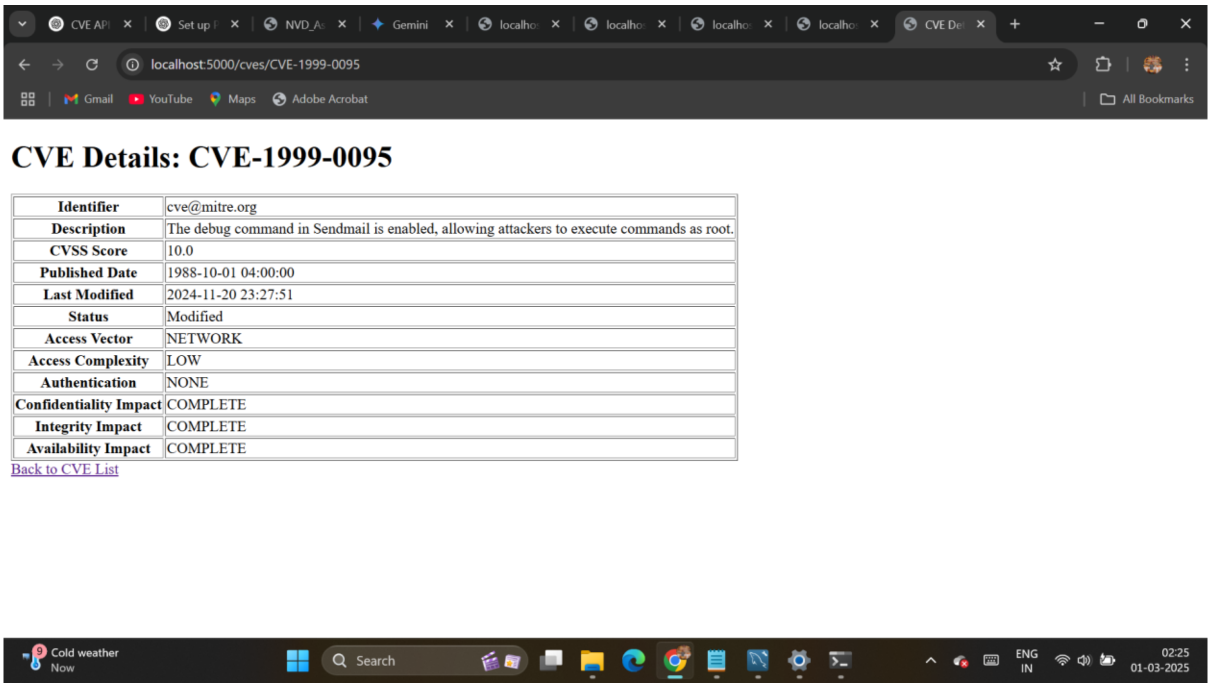


CVE List

CVE ID	Source Identifier	Published Date	Last Modified Date	Status
CVE-1999-0095	cve@mitre.org	1988-10-01 04:00:00	2024-11-20 23:27:51	Modified
CVE-1999-0082	cve@mitre.org	1988-11-11 05:00:00	2024-11-20 23:27:48	Modified
CVE-1999-1471	cve@mitre.org	1989-01-01 05:00:00	2024-11-20 23:31:12	Modified
CVE-1999-1122	cve@mitre.org	1989-07-26 04:00:00	2024-11-20 23:30:22	Modified
CVE-1999-1467	cve@mitre.org	1989-10-26 04:00:00	2024-11-20 23:31:11	Modified
CVE-1999-1506	cve@mitre.org	1990-01-29 05:00:00	2024-11-20 23:31:17	Modified
CVE-1999-0084	cve@mitre.org	1990-05-01 04:00:00	2024-11-20 23:27:49	Modified
CVE-2000-0388	cve@mitre.org	1990-05-09 04:00:00	2024-11-20 23:32:23	Modified
CVE-1999-0209	cve@mitre.org	1990-08-14 04:00:00	2024-11-20 23:28:08	Modified
CVE-1999-1198	cve@mitre.org	1990-10-03 04:00:00	2024-11-20 23:30:33	Modified

Results Per Page: 10
Next

Figure 5: UI interface

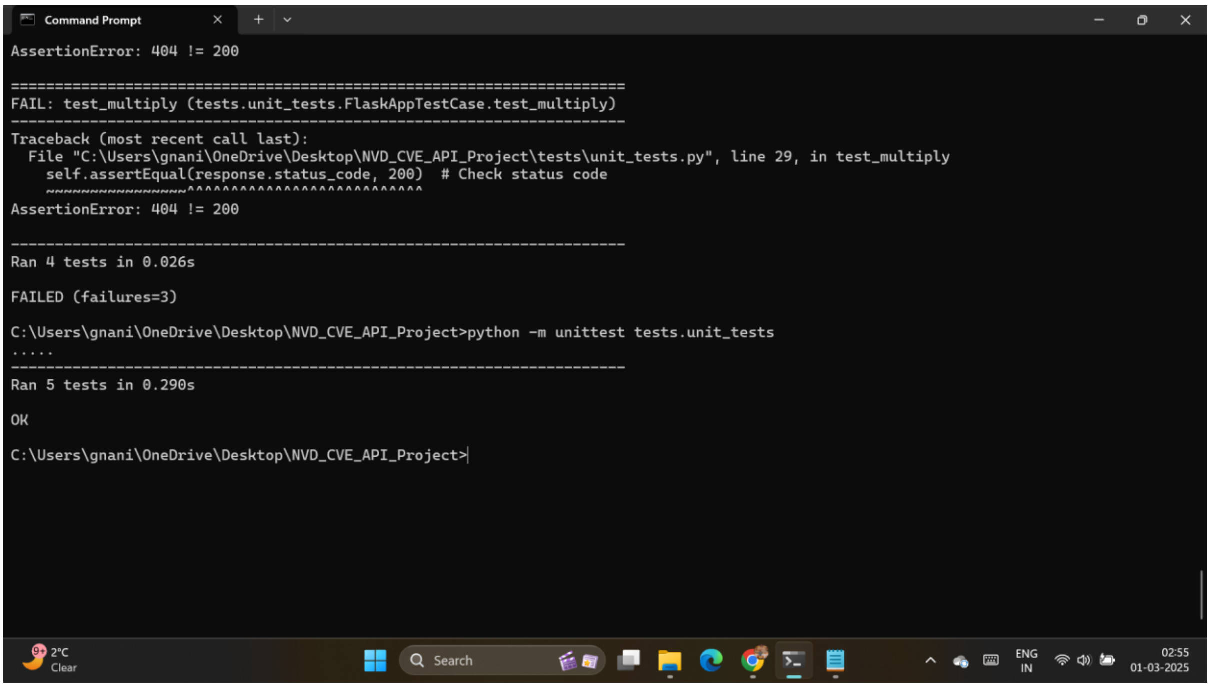


CVE Details: CVE-1999-0095

Identifier	cve@mitre.org
Description	The debug command in Sendmail is enabled, allowing attackers to execute commands as root.
CVSS Score	10.0
Published Date	1988-10-01 04:00:00
Last Modified	2024-11-20 23:27:51
Status	Modified
Access Vector	NETWORK
Access Complexity	LOW
Authentication	NONE
Confidentiality Impact	COMPLETE
Integrity Impact	COMPLETE
Availability Impact	COMPLETE

Back to CVE List

Figure 6: UI interface details of CVE



```
AssertionError: 404 != 200

======================================================================
FAIL: test_multiply (tests.unit_tests.FlaskAppTestCase.test_multiply)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\gnani\OneDrive\Desktop\NVD_CVE_API_Project\tests\unit_tests.py", line 29, in test_multiply
    self.assertEqual(response.status_code, 200)  # Check status code
    ~~~~~~~~~~~~~~~~^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: 404 != 200

----------------------------------------------------------------------
Ran 4 tests in 0.026s

FAILED (failures=3)

C:\Users\gnani\OneDrive\Desktop\NVD_CVE_API_Project>python -m unittest tests.unit_tests
.....
----------------------------------------------------------------------
Ran 5 tests in 0.290s

OK

C:\Users\gnani\OneDrive\Desktop\NVD_CVE_API_Project>
```

Figure 7: Output after executing unit tests