CVE Database & API

Overview

This project is a full-stack application that integrates with the National Vulnerability Database (NVD) CVE API to retrieve, store, and display Common Vulnerabilities and Exposures (CVE) details. The project consists of two parts:

1. **Backend**: An Express.js API that retrieves CVE data from the NVD, stores it in a MongoDB database, and provides API endpoints to access and filter CVE data.
2. **Frontend**: A React.js application that fetches and displays CVE data, implements pagination, and shows detailed views for individual CVEs.

Features

- **Data Synchronization**: Periodically syncs CVE data from the NVD API into the MongoDB database.
- **Data Filtering**: Provides API endpoints to filter CVE data by CVE ID, publication year, CVSS score, and last modified date.
- **Pagination & Sorting**: Implements pagination and sorting on both the backend and frontend.
- **Detailed View**: Displays detailed information for individual CVEs when a row is clicked.
- **Responsive UI**: A clean, user-friendly interface to view and interact with CVE data.

Tech Stack

- **Backend**: Node.js, Express.js, MongoDB, Axios, Node-Cron
- **Frontend**: React.js, React Router, Axios
- **Database**: MongoDB
- **Other**: CORS, dotenv, Mongoose

Setup

Prerequisites

1. **Node.js**: Ensure that Node.js is installed on your machine. You can download it from here.
2. **MongoDB**: Install MongoDB locally or use a cloud MongoDB provider like MongoDB Atlas. The default MongoDB URI in the `.env` file assumes local MongoDB.

Installation

- **Clone the repository**:

  git clone https://github.com/yourusername/cve-database-api.git

  cd cve-database-api

- **Install backend dependencies**:

  cd backend

  npm install

- **Install frontend dependencies**:

  cd frontend

  npm install

- **Set up environment variables**:

  Create a `.env` file in the **backend** folder and add the following:

  MONGODB_URI=mongodb://localhost:27017/cve_database

  PORT=5001

Start the Application

- **Start the backend server**:

  cd backend

  npm start

1. The backend will run on **http://localhost:5001**.

- **Start the frontend server**:

  cd frontend

  npm start

2. The frontend will run on **http://localhost:3000**.

Data Synchronization

The backend synchronizes CVE data with the NVD API at **midnight every day**. You can manually trigger the sync by running the following command in the **backend** folder:

- npm run sync

Accessing the API

The API provides several endpoints:

1. Get all CVEs

- **URL**: `/api/cves`
- **Query parameters**:
  - `page`: Page number (default: `1`).
  - `limit`: Number of records per page (default: `10`).
  - `id`: CVE ID (optional).
  - `year`: Publication year (optional).
  - `score`: CVSS score (optional).
  - `modifiedDays`: Number of days since last modified (optional).

Example request:

- GET http://localhost:5001/api/cves?page=1&limit=10&year=2021

2. Get CVE by ID

URL: `/api/cves/:id`

Example request:

- GET http://localhost:5001/api/cves/CVE-2021-1234

Frontend

The frontend is built using **React.js**. It allows users to:

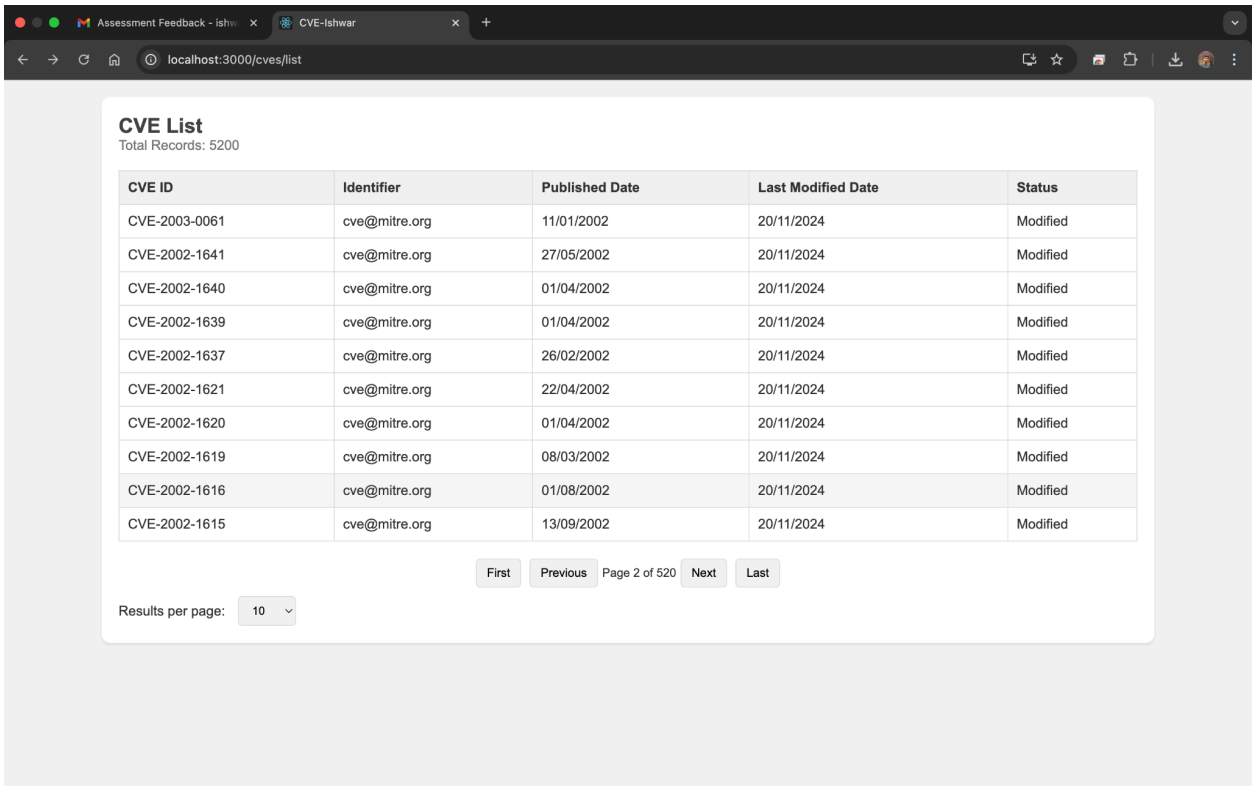
- View a list of CVEs with pagination and sorting.
- Click on a CVE to view detailed information about it.

Routes:

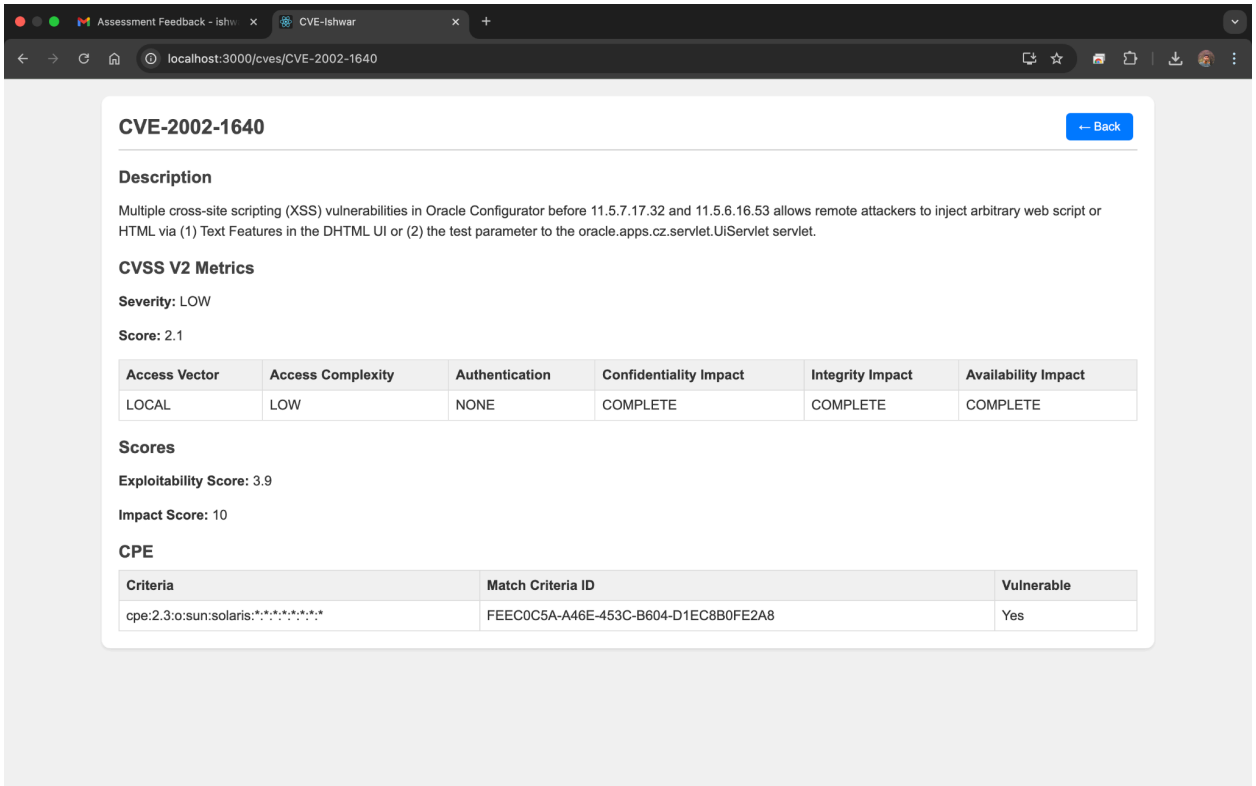
- **CVE List Page**: `/cves/list`
  - Displays a list of CVEs with pagination and sorting by publication date.
- **CVE Detail Page**: `/cves/:id`
  - Displays detailed information for a selected CVE.

Sample UI

- **CVE List**: Displays a table of CVEs with columns for CVE ID, identifier, published date, last modified date, and vulnerability status.



- **CVE Detail**: Displays detailed information about a CVE, including description, CVSS metrics, and associated CPEs.



API Documentation

Base URL: `http://localhost:5001/api`

1. Fetch a List of CVEs

Endpoint: `/api/cves`

Method: `GET`

Description: Fetches a paginated list of CVEs from the MongoDB database. You can filter the results by CVE ID, publication year, CVSS score, and last modified date.

Query Parameters:

- `page` (optional): The page number (default: `1`).
- `limit` (optional): The number of records per page (default: `10`).

- `id` (optional): CVE ID to filter by (e.g., `CVE-2021-1234`).
- `year` (optional): Publication year to filter by (e.g., `2021`).
- `score` (optional): CVSS score to filter by (greater than or equal to).
- `modifiedDays` (optional): Number of days since last modification (greater than or equal to).

Example Request:

- GET http://localhost:5001/api/cves?page=1&limit=10&year=2021

Example Response:

- {
- "cves": [
- {
- "id": "CVE-2021-1234",
- "published": "2021-01-01T00:00:00Z",
- "lastModified": "2021-01-10T00:00:00Z",
- "vulnStatus": "Known",
- "descriptions": [
- {
- "lang": "en",
- "value": "Description of CVE-2021-1234"
- }
-],
- "metrics": {
- "cvssMetricV2": {
- "cvssData": {
- "baseScore": 7.5,
- "severity": "High"
- }
- }
- }
- }
-],
- "total": 100,
- "totalPages": 10,
- "currentPage": 1,
- "hasNextPage": true,
- "hasPrevPage": false
- }

2. Fetch a Specific CVE by ID

Endpoint: `/api/cves/:id`

Method: `GET`

Description: Fetches detailed information for a specific CVE by its ID.

Parameters:

- `id` (required): The unique CVE ID (e.g., `CVE-2021-1234`).

Example Request:

- GET http://localhost:5001/api/cves/CVE-2021-1234

Example Response:

- {
- "id": "CVE-2021-1234",
- "published": "2021-01-01T00:00:00Z",
- "lastModified": "2021-01-10T00:00:00Z",
- "vulnStatus": "Known",
- "descriptions": [
- {
- "lang": "en",
- "value": "Detailed description of CVE-2021-1234"
- }
-],
- "metrics": {
- "cvssMetricV2": {
- "cvssData": {
- "baseScore": 7.5,
- "severity": "High",
- "vectorString": "AV:N/AC:L/Au:N/C:P/I:P/A:C",
- "exploitabilityScore": 8.0,
- "impactScore": 6.0
- }
- }
- },

- "cpe": [
- {
- "criteria": "cpe:2.3:a:vendor:product:version",
- "matchCriteriaId": "1234",
- "vulnerable": true
- }
-]
- }

3. Manually Trigger Data Synchronization

Endpoint: `/api/sync` (Optional endpoint if triggered via manual command, not via HTTP request)

Method: `POST`

Description: Manually triggers the synchronization of CVE data from the NVD API into the local MongoDB database.

Usage:

- This endpoint is typically invoked from the backend using a cron job, running daily at midnight. However, developers or administrators can manually trigger the sync using the following command:
  npm run sync

Backend Synchronization (Cron Job)

Cron Job: Runs every day at **midnight** to fetch the latest CVE data from the NVD API and store it in the MongoDB database.

Cron Expression: `0 0 * * *`

- **Trigger Time**: Every day at 12:00 AM (midnight).

Error Responses

1. **Invalid CVE ID**:

   HTTP Status Code: `400 Bad Request`

- **Response**:

```
{
  "error": "Invalid CVE ID format"
}
```

2. **CVE Not Found**:

   HTTP Status Code: `404 Not Found`

- **Response**:

```
{
  "error": "CVE not found"
}
```

3. **Internal Server Error**:

   HTTP Status Code: `500 Internal Server Error`

- **Response**:

```
{
  "error": "Internal server error"
}
```

Pagination and Sorting

- **Pagination**: All endpoints that return lists of CVEs support pagination.
  - **`page`**: Specifies the page number (default is `1`).
  - **`limit`**: Specifies the number of records per page (default is `10`).
- **Sorting**: The CVE list is sorted by the CVE ID in ascending order and the last modified date in descending order.

API Rate Limiting: The backend may include basic rate limiting to prevent abuse, but this should be configured according to specific application needs.