**NVD CPE Assessment: Final Report**

GitHub link: -

**1. Project Overview**

This project implements a system to retrieve, store, and serve Common Platform Enumeration (CPE) data from the National Vulnerability Database (NVD) XML file. The system includes a backend built with Flask (Python) and a frontend built with React. The goal was to parse the XML, store the required fields in a PostgreSQL database, and provide API endpoints to allow efficient querying and pagination of the data. A frontend UI was also developed to render the data and support interactive filtering and pagination.

**2. Implementation Details**

* **Backend:**
  + **Framework:** Flask (Python)
  + **Database:** PostgreSQL, with SQLAlchemy ORM
  + **Modules Used:** flask, flask\_sqlalchemy, flask\_cors, requests, lxml
  + **Functionality:**
    - Download and parse the official CPE dictionary XML file.
    - Extract fields: cpe\_title, cpe\_22\_uri, cpe\_23\_uri, reference\_links, cpe\_22\_deprecation\_date, cpe\_23\_deprecation\_date.
    - Store data in the database.
    - Provide two RESTful API endpoints:
      * /api/cpes: Paginated retrieval of CPEs.
      * /api/cpes/search: Filtered search by cpe\_title, cpe\_22\_uri, cpe\_23\_uri, and deprecation\_date.
* **Frontend:**
  + **Framework:** React
  + **Libraries Used:** Material-UI, Axios
  + **Features:**
    - Table/grid view of CPE data.
    - Filters for title, URIs, and deprecation date.
    - Pagination with selectable rows per page (15, 20, 30, 50).
    - Truncated long titles and tooltips for full text.
    - Popover to display extra reference links beyond the first two.
    - Fallback message if no data is found.

**3. Testing and Verification**

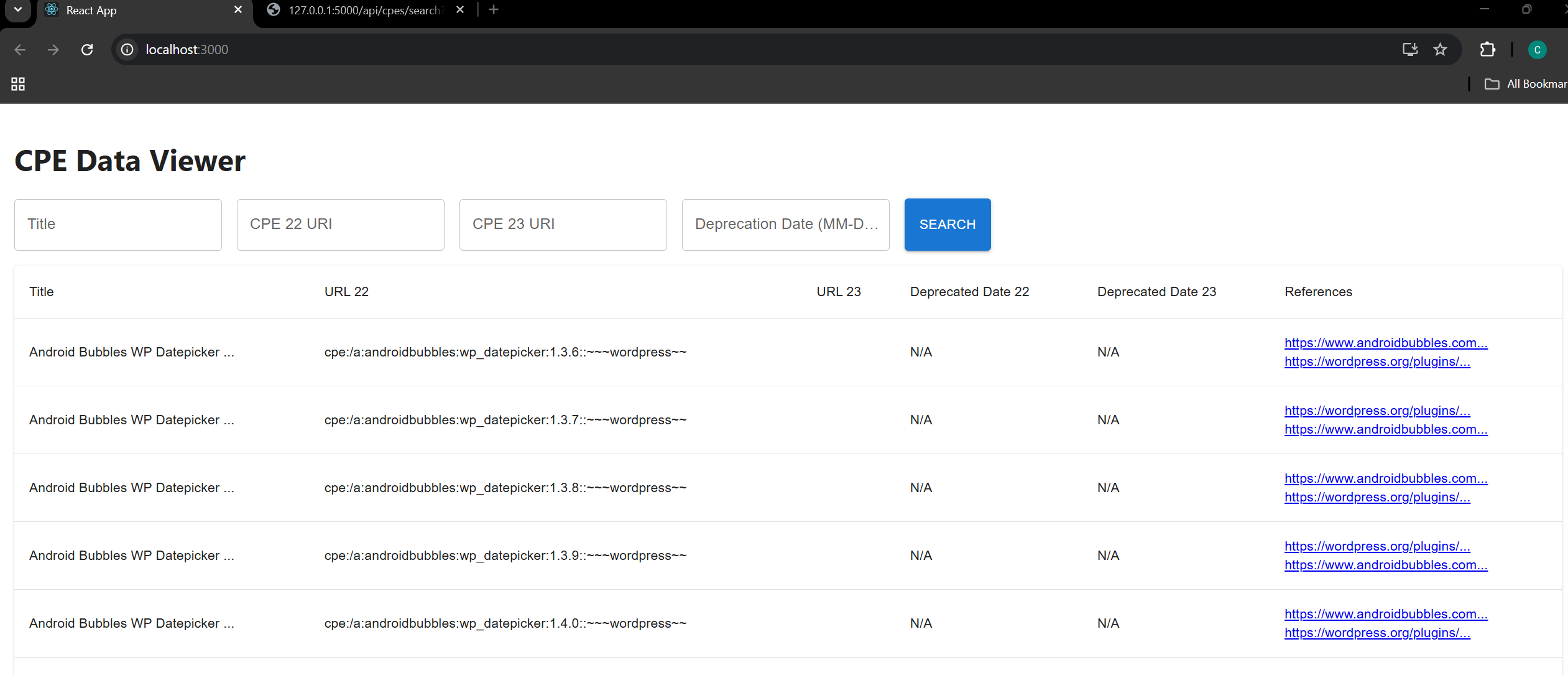
* **Backend Testing:**
  + Verified that data is correctly parsed and stored by querying the database directly.
  + Confirmed API functionality using Postman and browser-based tests.
  + Search functionality tested with various parameters and combinations.
* **Frontend Testing:**
  + Successfully retrieved and displayed data from the backend.
  + Verified filtering and pagination work as expected.
  + Tested popover feature for reference links and fallback message for no results.

**5. Conclusion**

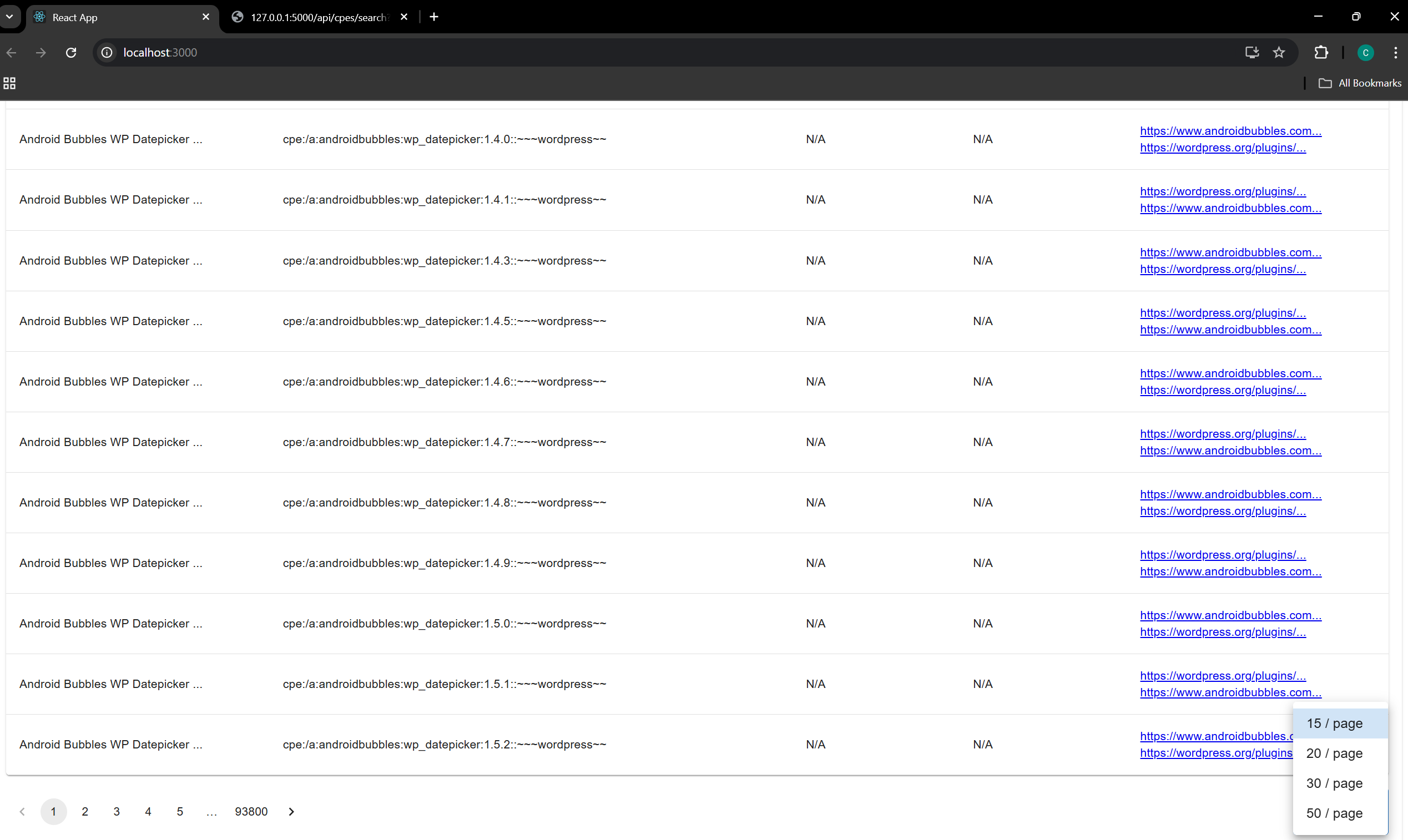
The project successfully meets all requirements outlined in the evaluation criteria. Both the backend and frontend are fully functional, and the system provides an efficient and user-friendly way to view and search CPE data. I have attached all screenshots to prove its working. I have almost tried and completed as you asked and most of them are also mentioned in Screenshot.

All for Frontend

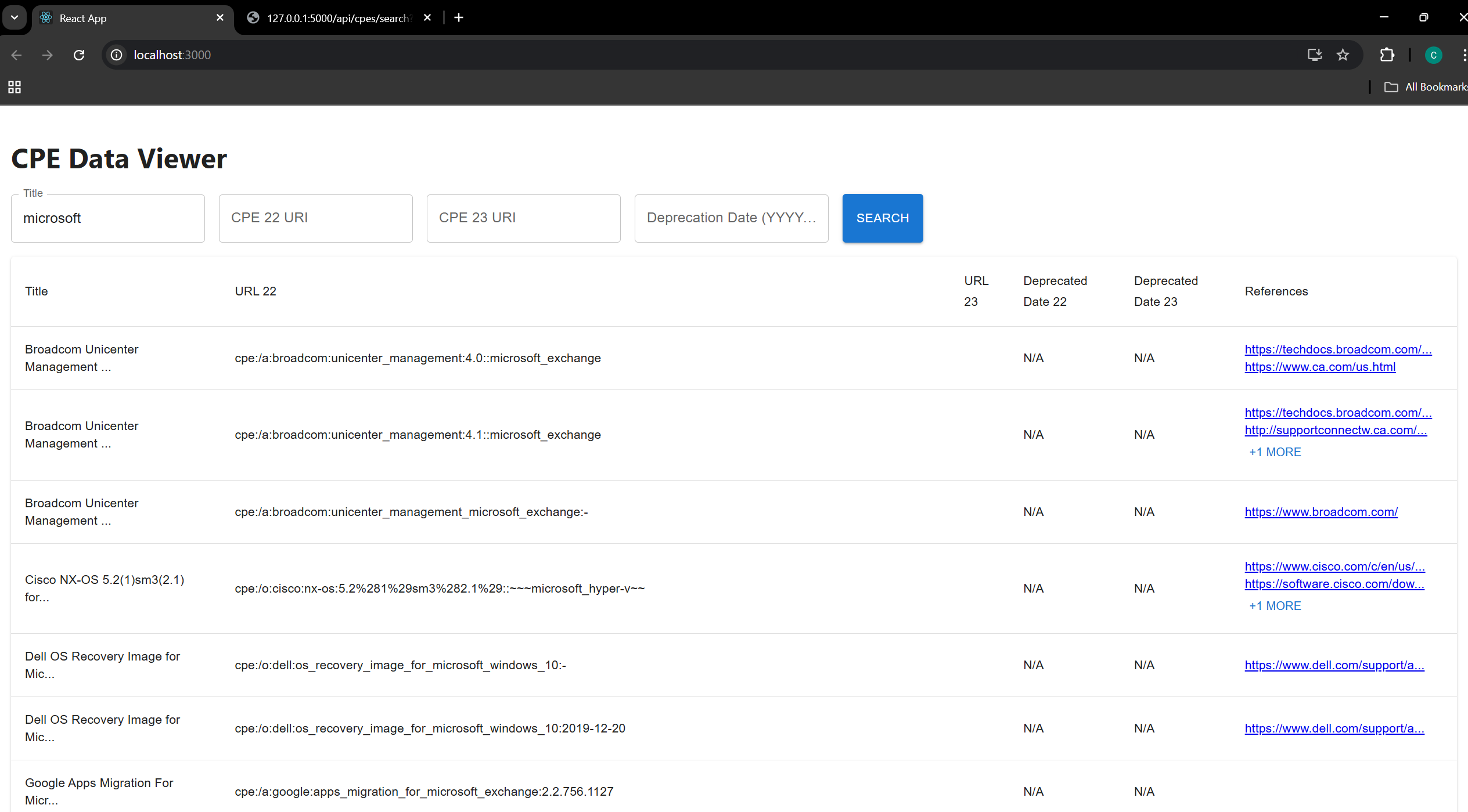
1] Search filter

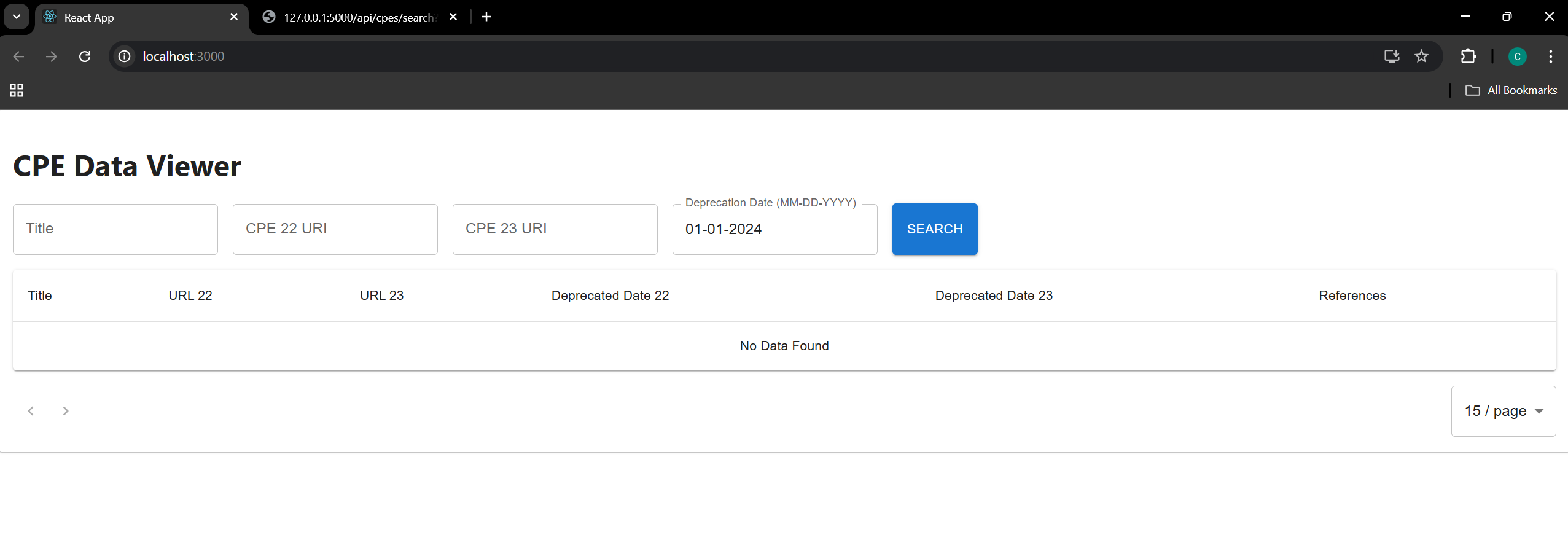


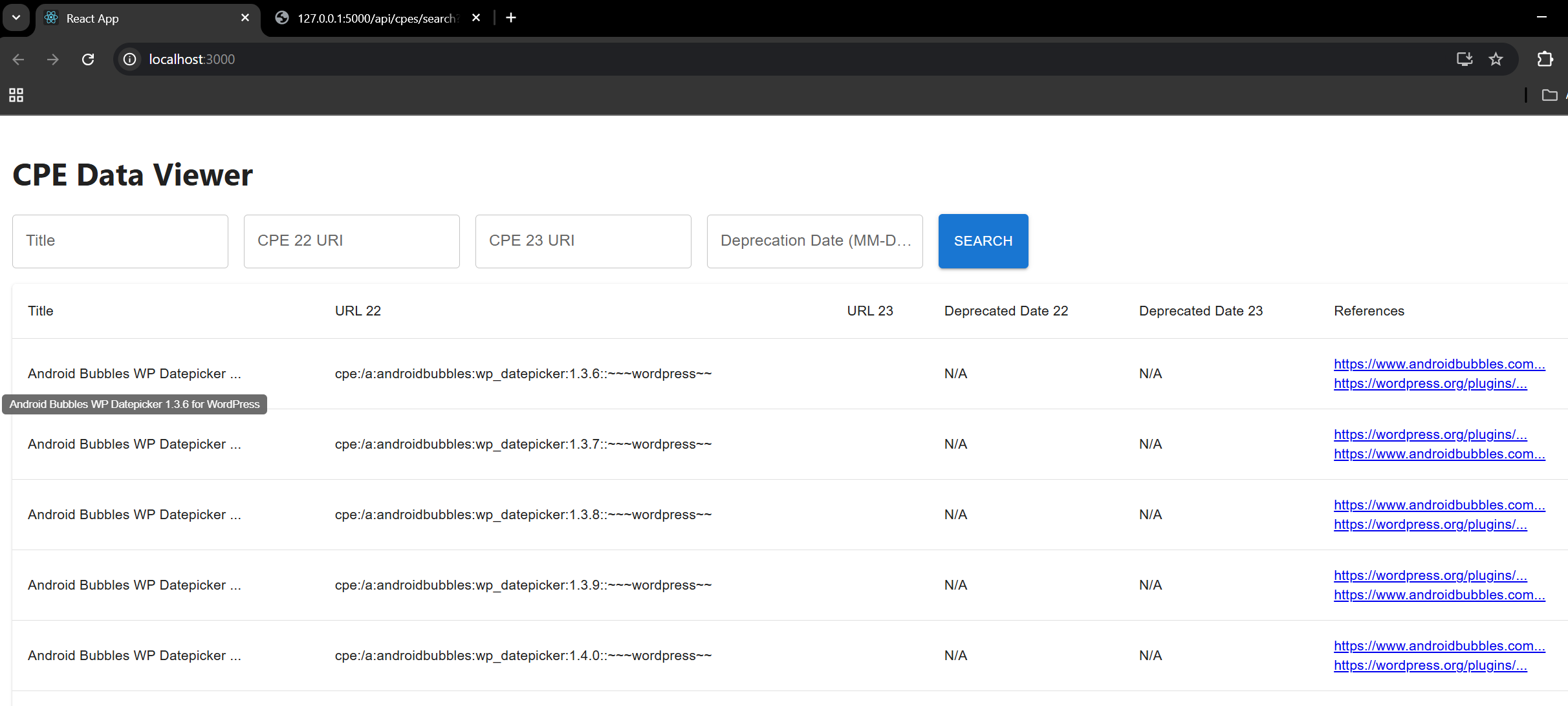
2] the page filer 15 to 50



3] Filter working







All for backend

