Take a look at the sample data in the attachment. You need to build an API (use HTTP POST) to search some data entries from the attached zip files.

At first, your nodejs program must be able to load multiple zip files like these attached after booted. As you see, the naming convention follow the rule that only the year number changes. There are more files like these. I'm just listing these 3 as an example. Your function must be designed to be able to load a series of these files.

After that, you need to create a route. The route will take an array of strings as the input, in the req.body. Each element in the array will follow CPE 2.3 format. Take a look at this if you don't understand CPE: https://cpe.mitre.org/specification/. An element in the input array looks like the value stored in the "cpe23Uri" property in the attached data files. You need to build a logic to match each element from the input array to check whether it has one or more CVE records in the data. The output of your search is the value of cve.CVE_data_meta.ID of the matching record(s).

Note that your searching algorithm must be able to follow the "configurations" property of each record in the data files. In other words, you are not simply matching the "cpe23Uri" value, but your logic shall consider other properties in the "configuration" property, including the "operator" property and version-related properties such as "versionEndExcluding", "versionStartingIncluding", "versionEndIncluding", and "versionStartExcluding".

In addition, your search logic must be able to perform a search even if the information from the input is incomplete. For example, if an input element only has the vendor name, but doesn't have a product name, your logic must return all records from that vendor regardless of the product name. Another example, if an input element doesn't have version information, your logic must return all records of that vendor and product name regardless of version number. Truncate the returned value of each input element to a maximum of 10 records, if the input element is mapped to too many records. The returned value shall be an array of arrays, each sub-array matching the index of the string element of the input array.

Throughout the project, you must consult me before using any external packages, and only proceed if approved. In general, you are only allowed to use an external package if it has a MIT license. But remember, you need to consult me for approval regardless what packages you want to use.

You can refer to the cpe2cve command in this open source project (written in Golang) for some guidance: https://github.com/facebookincubator/nvdtools#cpe2cve
But you are not allowed to copy their algorithm, for copyright reasons. Also, we don't require that many configuration settings. Don't over-design it.

The deliverables are: 1) the function to load a series of
such .json.zip files; 2) the searching logic inside an API route.