## Project Scenario

The Ontology team has a dataset of clinical concepts called “Master Ontology” currently managed in CSV. They would like you to architect/implement an application for managing this dataset so that they can easily search, browse and update it. They plan to do a one-time load of the CSV data into the app and then manage it through the app moving forward.

## Requirements

* All employees have access to search and view the data, but only the Ontology team can create, update and delete entities in the system.
* Max users at any given time will be <= 30.
* Application will be hosted on AWS and you have access to any of their managed services
* Front-end of the application will be built in React.js
* Users are authenticated using SAML-SSO and groups help define a user’s access (admin, edit, view, etc…).

## Data

**Sample CSV data**

conceptId, displayName, description, parentIds, childIds, alternateNames  
1,"Diagnosis", "Entity domain", null, "2,3", null  
2,"Disease of Nervous System", "Diseases targeting the nervous system", "1", "4", null  
3,"Disease of Eye", "Diseases targeting the eye", "1", "8,9", null

4,"Physical Disorders", "Physical Disorders", "1", "8,9", null

5,"Multiple Sclerosis (MS)", "Multiple Sclerosis", "2,4", "5,6,7","MS,name1,name2"

**Sample Relationship Model**

Each row is its own clinical concept, but they are related in a hierarchy.

Diagnosis

Disease of Nervous System

Multiple Sclerosis

Disease of Eye

Glaucoma

## Task

1. Using AWS services and React.js create a proof of concept application that does the following:
   1. List, search and view details of clinical concepts
   2. Add, edit and delete concepts
2. Assuming this project moves beyond the POC stage, create a diagram to illustrate the target application architecture and data flow including how any code and infrastructure would be deployed.
3. Briefly summarize any proposals, concerns or questions that you may have if you were to meet with the Ontology team.

You are free to architect and implement the application as you see fit, but it should be a working application (including database) that meets the ask for Task 1. You can use AWS free tier services for hosting.

You must provide a working URL for us to browse the application and access to source code (Github, Gitlab, etc…).

**You do not need to implement security or transform the data from CSV format as part of Task 1.** Instead, they should be taken into account as part of Task 2.

Given that you do not have direct access to end users, you are free to make assumptions in your implementation, but please document those assumptions and describe how they influence decisions you made as part of Task 3.

## Success Criteria

Your project will be evaluated against the following criteria:

1. Whether the application satisfies business requirements or not
2. Ability to evaluate business requirements and translate them into a technical solution
3. Architecture patterns for scalability and reuse including usage and understanding of AWS services and best practices
4. Code quality and readability
5. Aesthetic design and usability of front-end
6. Additional features, if any, that are not required but add to overall end-user experience