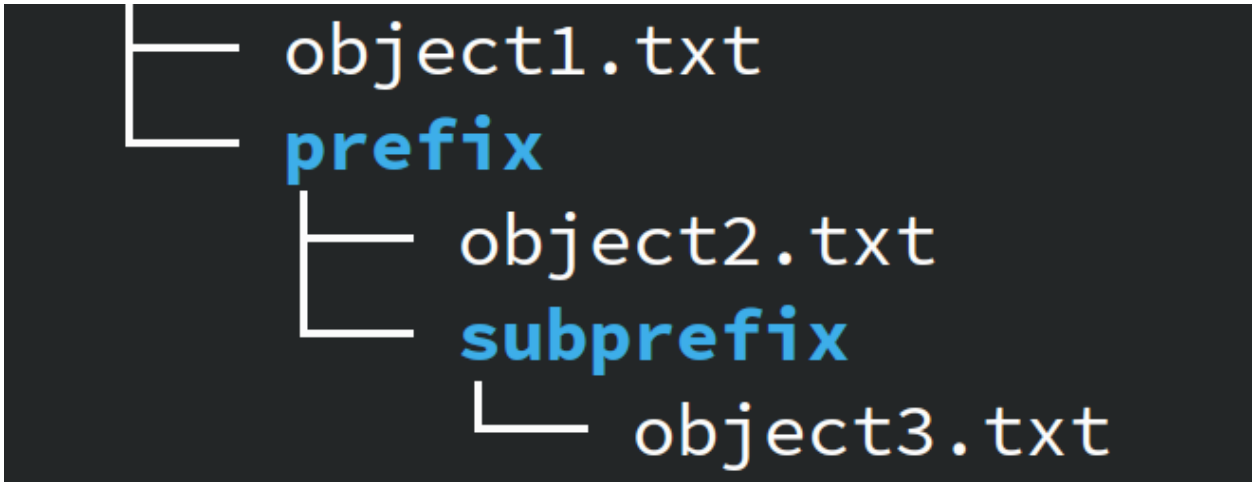
Create a “single page” react app that implements a simple browser for an AWS S3 bucket.

1. Non-functional requirements.
    - The presented solution should be as close as possible to production ready.
    - Your implementation can be a partial solution. It’s up to you where you will draw the line and send it to us.
    - The task shouldn’t require backend implementation. It should be possible with frontend-only (in browser).
    - You should use react and typescript.
    - You should not use a CSS framework (e.g. Bootstrap, Foundation, Bulma)
    - You should not use a component library (e.g.Material UI, React-Bootstrap, Fluent UI).
    - We recommend using the S3 module of aws-sdk for communication with the s3 bucket.

2. Data - use the S3 API in order to list, create and delete objects in a bucket.
    - To connect to S3, use the credentials provided in the email with this task
    - S3 is not a filesystem, but you can treat prefixes in object names as directories.
        - You can assume that all objects are text files.
        - An object name of the form **prefix/subprefix/object.txt** can be interpreted as an absolute path to the file name **object.txt**.
        - Note that there is no concept like an empty directory in s3, and part of the task is to devise a way of handling that.
        - To represent the structure given below, you will need those three objects:
            - **object1.txt**
            - **prefix/object2.txt**
            - **prefix/subprefix/object3.txt**



- You should know that the S3 API allows for such operation on a bucket:
  - List all objects in a bucket.
  - List objects in a bucket starting with some prefix.
  - Creating an object with a given name.
  - Getting object data given its name.
  - Deleting an object given its name.

3. Functional spec
   - The main goal is to enable users to:
     - Browse an S3 bucket as a filesystem.
     - Create, Read and Delete text files represented as objects in the s3 bucket. No need to update existing**.**
4. UI/UX
   - The user interface can be minimalistic but yet should showcase your understanding of UI which is easy to work with and look at.
   - On first interaction the user should be asked to configure some settings:
     - S3 Secret Key
     - Access Key ID
     - The bucket name
   - The config should be remembered for subsequent use of the system.
   - It should present a simple, file browser-like interface that organizes the bucket by prefixes.
   - The file browser should consist of two parts::
     - Currently working directory view (on the right) that:

            - Displays the entire content (files and directories) of the so-called “current working directory.”
            - Initially the root directory is considered “current”
            - It should allow you to navigate up and down the directory structure in some way.
            - It should allow you to read the content of a file
            - It should allow you to create new subdirectories.
            - It should allow you to create new files with a given content.
            - It should allow you to delete files.
        - Tree view (on the left) that:
            - Shows only directories.
            - Initially it will show only the root subdirectories.
            - A directory containing subdirectories should be marked so the user can understand that it can be expanded.
            - When a marked directory is clicked, it should be expanded, and its subdirs should be displayed.
            - If an expanded directory is clicked, it should be collapsed, and all its subdirs should be hidden.
            - If an expanded directory is clicked, it should be collapsed, and all its sub-dirs should be hidden.
            - If a directory is double-clicked, it should be selected as the current directory and displayed in the other view.
            - The current working directory should always be visible in the tree view and should be “decorated” in some way.

    - The web app should be able to detect changes to the bucket and react appropriately by removing non-existent or displaying new directories or files.