

Frontend Coding Problem:

## Directory Structure Implementation

Your task is to create a Frontend app which will provide a desktop like folder management environment, where you can create and traverse between various directories and subdirectories. You are required to create a single page application, where you can perform basic operations like

1. Create folders
2. Back or Up button
3. Navigate easily between parent folders and subfolders in UI
4. Show current pwd (Present Working Directory) in top

This directory implementation is similar to an operating system directory structure, where you can create directory and browse through them at various levels.

Name ▲



folder\_1



folder\_2

---

*Above is just a representation. Basically, your application must keep running and keep accepting user interaction.*

## Instructions

At start, application would be at ROOT directory '/'. Application should be provided with options to create and traverse through directories. Your task is to successfully handle all events. If there is a scope for error, handle it with proper error message.

You are required to provide us the source code & install documentation (github repository preferred).

**You don't have to 'actually' create/remove the directories. But your application should keep a logical track of directories in a running session.**

## Example:

1. You start with root "/" directory .
2. There should be a button to create new folder & on creating the first folder it must be added to root directory. Suppose on creating a folder named 'question', it should look something like.
3. Child directories when created on 'question' folder should be visible as shown
4. Your implementation should support multi level directories  
e.g.

```
➤ "/"
  ➤ "folder_1"
    ➤ "folder_2"
  ➤ "other_folder"
    ➤ "child_folder"
      ➤ "example_folder"
      ➤ "sample_folder"
```

## **Application evaluation parameters**

Your application would be evaluated on following parameters.

- How extendable your code is. You may be asked to add more features in the following rounds
- Corner cases handling and proper success/error message outputs