File & Folder Management System

We are expecting the applicant to design and develop a web application which provides the users to have a file system of their own. There are required functionalities which are expected to be implemented. But for the sake of the evaluation process, you might choose to eliminate some of the functionalities in order to present us a working prototype.

Project Description

The client(a medium sized firm) wants his employers to have a file system where they can store their files in it and access from everywhere. Therefore a web application will satisfy the client.

Client wants his employers to be able to:

Login to the system without a registration process (users can be added to the db manually) => +

Empty file system at first login => +

Each user will only see their own files and folders => +

Files and folders will be listed in a table, showing creation dates, and sizes on related columns => +

Sort according to columns

Create folders => +

Navigate through folders - breadcrumbs (C > Users > Hp….)should be provided

Create folders in subfolders => +

Upload files under every subfolder he has created => +- drag&drop is a plus

Each level of folder is shown in a new page(not a tree structure on a single page)

Download single file => +

Download single folder(should be zipped) => +

View viewable files on the web application(pictures, videos, pdf files)

Rename files/folders => +

Delete files/folders - future plans might require the functionality to restore the deleted files (ONLY DELETED NOT STORED TRASH) => +

No sharing required for now - future plans might change this (\*) => -

Evaluation Process

We expect the applicant to:

Plan his/her the development process(including database design)in iterations and share the plan with us prior to start of the development process.

Provide a status report after each iteration.

Write extendable, maintainable and readable code. The use of design patterns is a huge plus.

Present a working prototype with all or some of the requirements satisfied. The robustness of the application is more important than the number of the functionalities provided.

Technologies

Server Side

.NET6 project with layered architecture

Core Layer (Db Entities, Common Codes, Extension Methods etc)

Data Access Layer (Repositories)

Business Layer

Web Api Layer

Unit Tests (x Unit)

Useful Libraries

Entity Framework Core with DB First

Microsft.AspNetCore.Authentication.JwtBearer

Fluent Validation.AspNetCore

AutoMapper

Moq

Client Side

React

Functional Components

Hooks

Context API

Axios

Please create a private repository in github and push your code there as you progress through project

Have Fun!