Project description:

1. Singularitybdweb

* This project is in .net core 3.1. It uses Microsoft identity Framework for security/login etc. with some customized class for Role based dynamic mane.
* This also has cache based Authorization
* I have used the default layout for time sortage
* This project also contains 2 api which can be authorized by ether identity or JWT token. (Not asked)

1. Please Attach the database Which is in under(Singularitybdweb

=>database folder) restore it on Sql Server 2019

And Update the connection string in both projects appconfig.json files

* If api Project Url changes please Change it In (Singularitybdweb=>apimodels=>JwtToken=>MyToken=> base Url variable)

Current url (http://localhost:17250/api/)

Functionality:

If a user registers the admin needs to Assign that user a role i.e. Employee

The he will get to see those menus that are binded with Employee role

Role can be added, edited, deleted

Menus cab be added, edited, deleted

Each role contains menus that are assigned to it.

The model I have chosen is for booklist

This Store a Book Detail. Have action (add, edit, delete)

For Trash and recycle I have created a menu which contains files

(add, edit, delete, restore)- It doesn’t take any file from user just take name and url. For limited time a had I could not implement any ftp server/sftp/or file upload download system.

The api provides Two Data Services Booklist and Files.

These are the already stored users

**manager@test.com**

**admin@test.com**

**User@gmail.com**

**hridoy@gmail.com**

[**employee@test.com**](mailto:employee@test.com)

**with common password: P@ssw0rd**

1. SingularityApi: Restful web api

This project is in .net 5.0.

Which Authenticated with JWT token Authentication.

Did not Implemented Cross-Origin Resource Sharing because the url of the Web project can change.

To access api via external source:

Need to first create a JWT token:

Using A post http call :

On this url: <http://localhost:17250/api/Account/>

NB: Change the root url if api url changes on your end.

which needs to pass user credential like: this data in body to get jwt token

{

    "UserName":"admin@test.com",

    "Password":"P@ssw0rd"

}