# ONECLICK WORKFLOW



| Prerequisite | |
|---|---|
| Install & Access | Path |
| VS Code | Link |
| NPM & Node | npm & node |
| GitHub Repository | Link |
| Twinit Extension | Latest version |

## Steps to Instruction

1. Initially Install VS Code (Link)
2. Then Install npm and Node. (npm & node) using command
   a. npm command: **npm i node@<version>**
3. NPM Registry setup for Configuring Environment
   a. In the User Variables section, click **New** and enter the name and value for the environment variable:

      i. Variable Name: GITHUB_TOKEN
      ii. Variable Value: E20******2d
      iii. Click **OK**
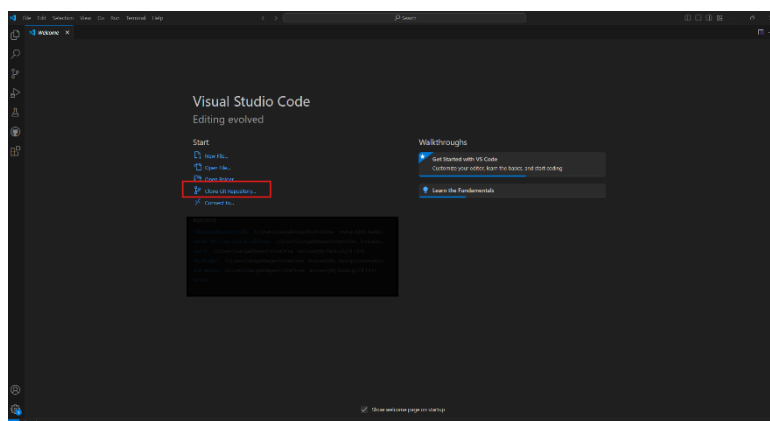   b. Use the %GITHUB_USER% and %GITHUB_TOKEN% anywhere that supports environment variables.

---

# ONECLICK WORKFLOW
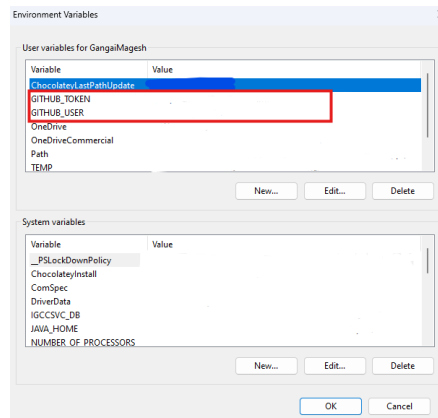


c.   Get access to GITHUB repository "Twinit-IPA-Core-Digital-Buildings "

4.   Clone the Repository from Link
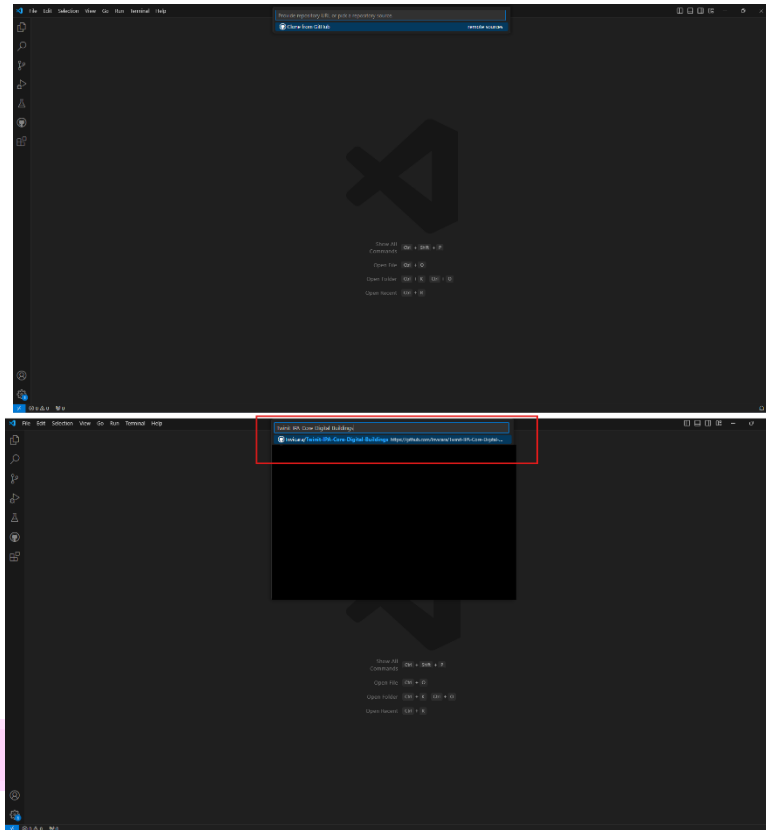   a.   Signup with you GitHub in VS code using the profile Icon.
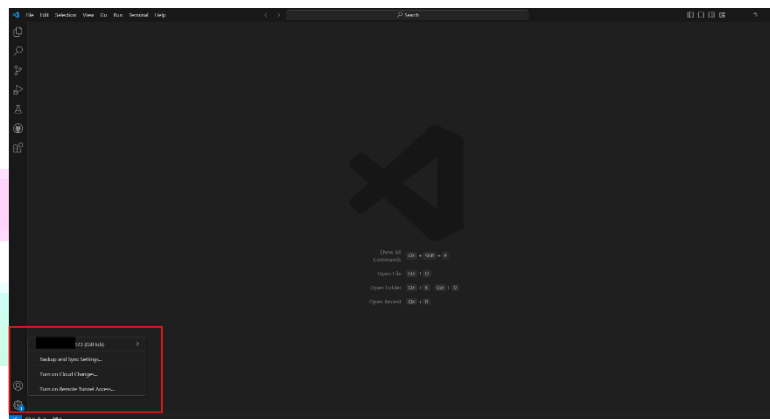


   b.   From welcome page Select "Clone Git Repository".
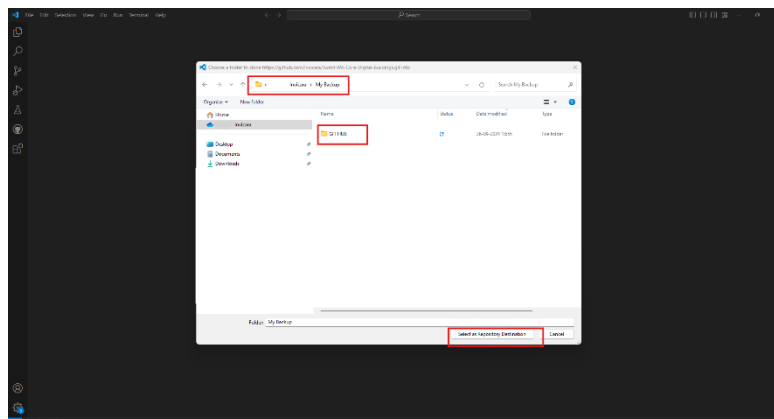
# ONECLICK WORKFLOW

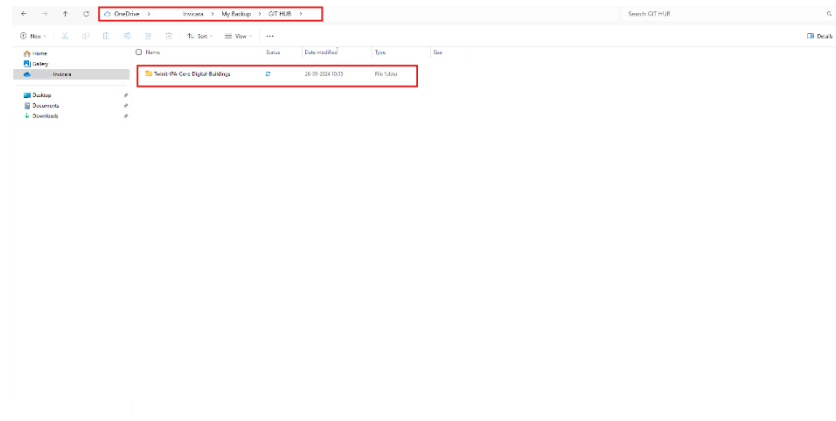c. From the Context Drop drown search for repository name as "Twinit-IPA-Core-Digital-Buildings" and select the same.



d. Now the window will prompt for the location in your local machine.
e. Select the path, click on "Select as Repository Destination".



f. Now the Repository will clone in the selected location.

# ONECLICK WORKFLOW



g. Clone the repository under local path of your machine.
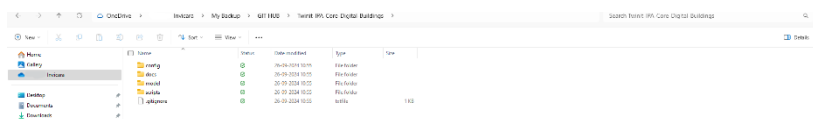


5. Opening the repository in VS Code
   a. Open VS Code and Select File > Open Folder option.
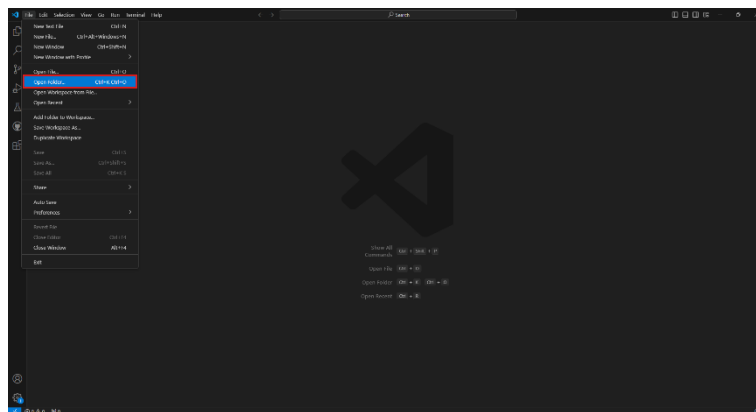


b. Select the folder where repository is cloned.

# ONECLICK WORKFLOW
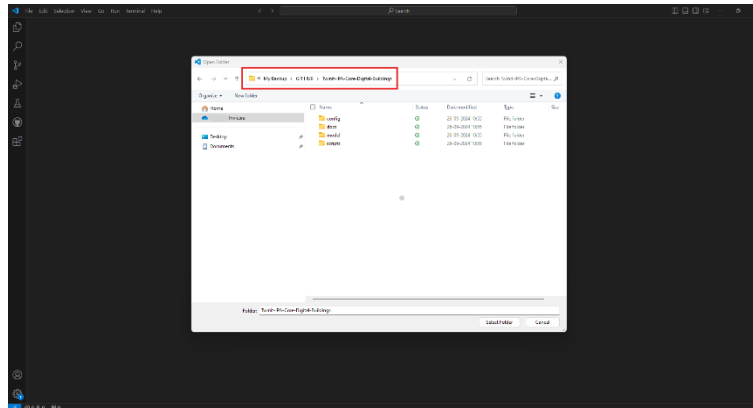


    c.   Selected repository will open in the right panel of VS Code.
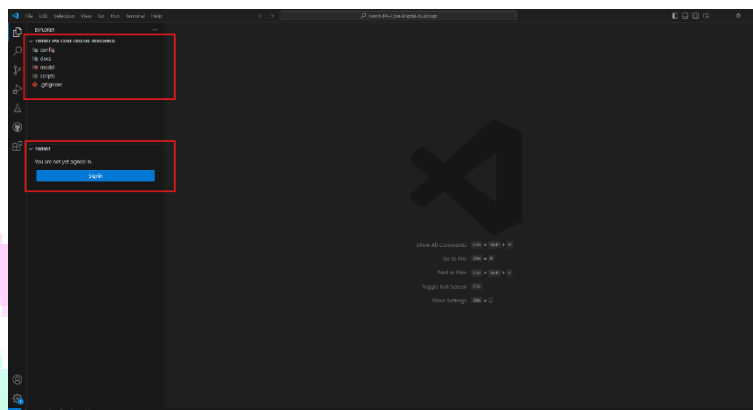


6.   Upload Twinit version via extension
    a.   Navigate to Extension in VS Code
    b.   Select 3 dots on right corner of extension and select **"Install from VSIX"**
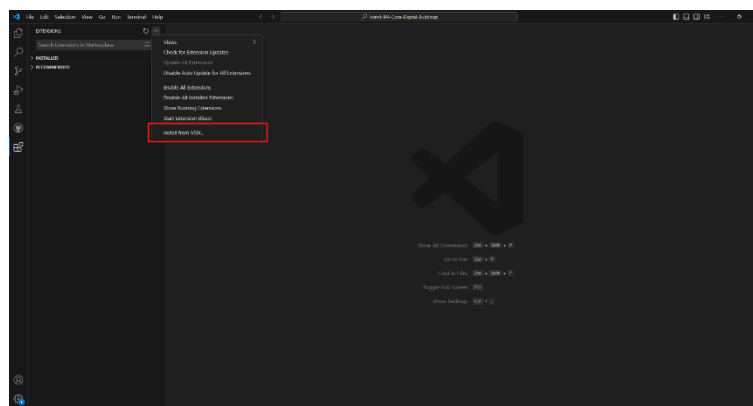


    c.   Select recent version of "**twinit-dev-x.x.x.vsix**" file that saved in local path.

# ONECLICK WORKFLOW
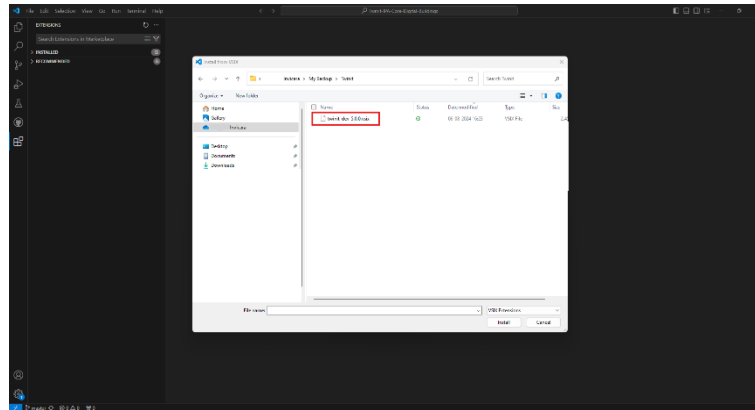


d. VS Code should display the Twinit-Dev with Twinit Extension Developer command tool along with repository .
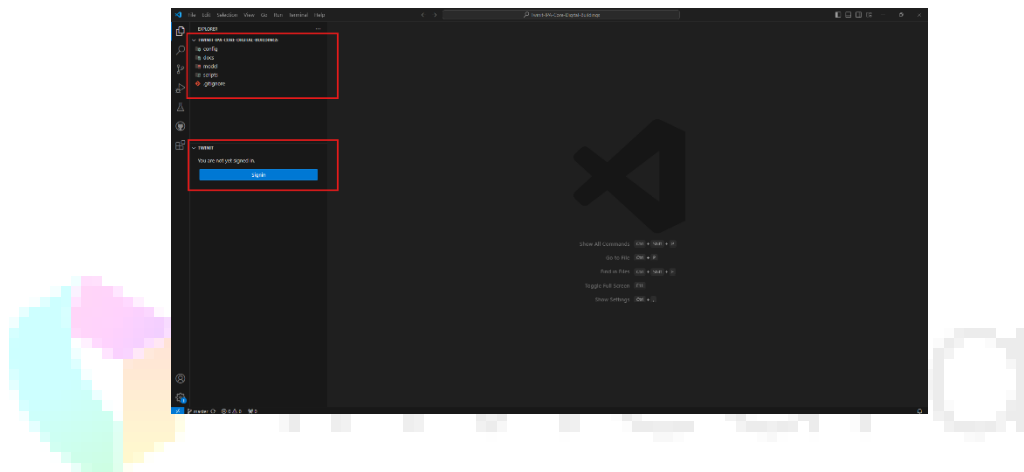


7. Add environmental setup in setting.json
    a. Select setting icon > settings
    b. From settings, select Application and from right panel select "Edit setting.json"
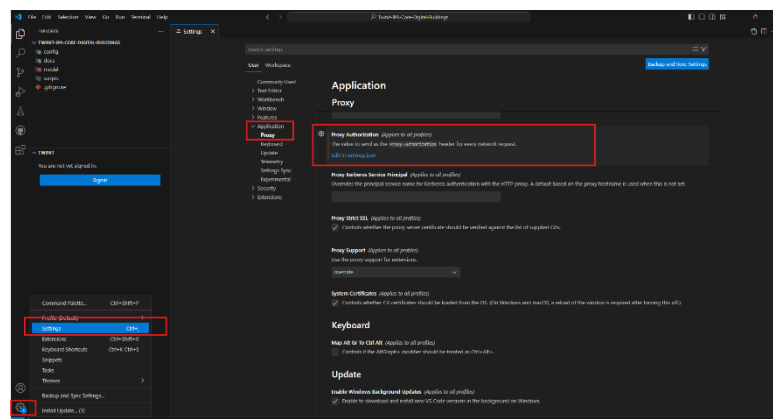


c. Update the script with required environment details for signing **in "env, name, access key and secret key"** of your profile.

# ONECLICK WORKFLOW



SettingJSON.txt
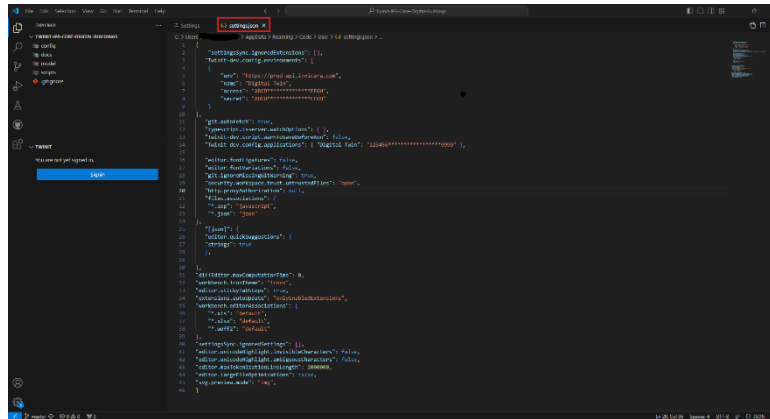


8. Click on "Sign-in" From "Twinit-Dev" from right corner of 3 dots and select relevant environment.





---
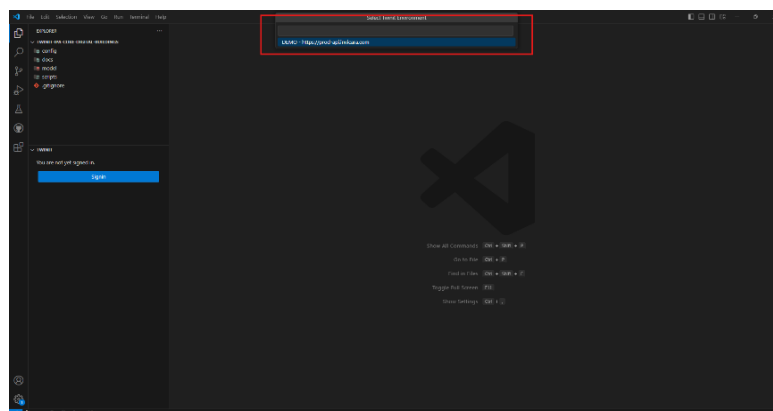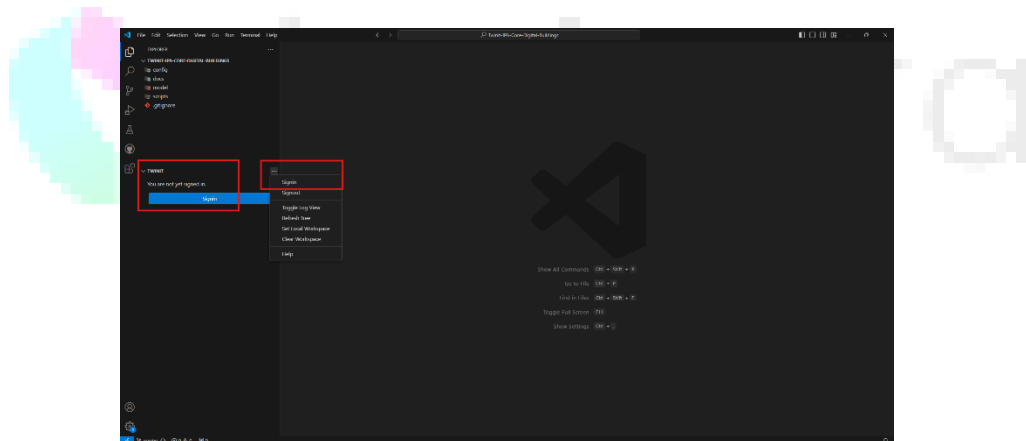
# ONECLICK WORKFLOW

9. Create Project Setup using Twinit-Dev
    a. Right click and select Twinit: New Project
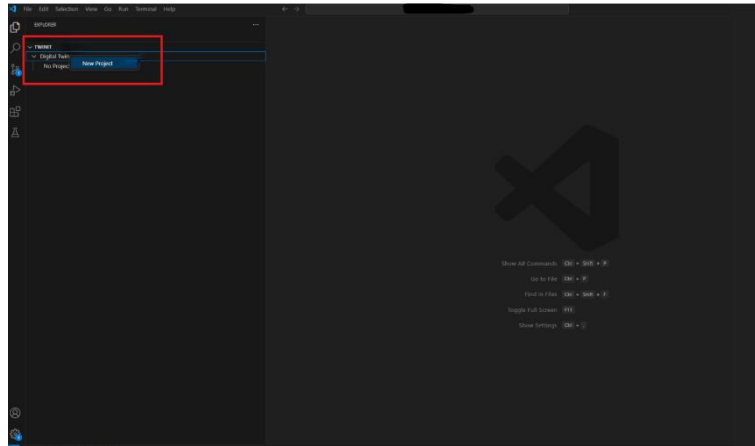


    b. Enter "Project Name" [Eg: Asset Twin Demo]
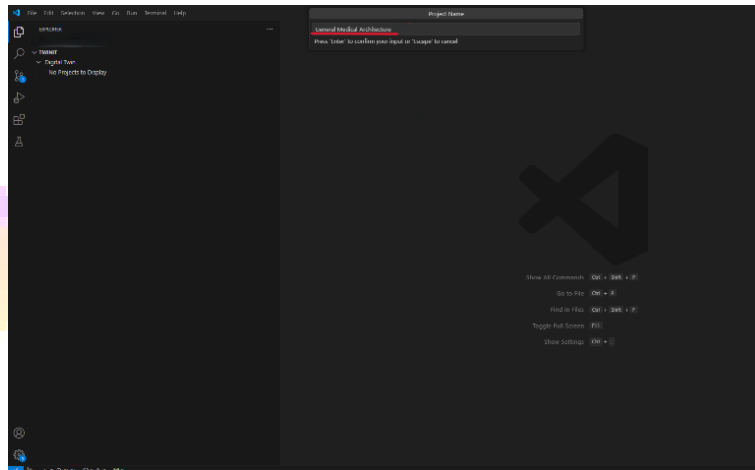


    c. Enter the "Project Description" [Eg: Asset Twin Demo Project]
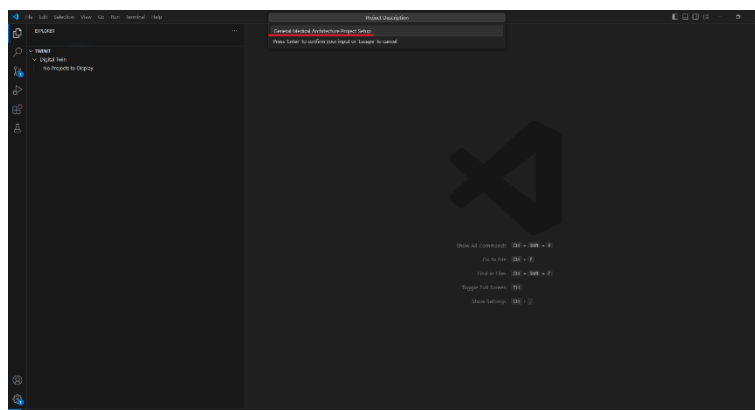


    d. Enter the "Project Short Name" [Eg: ATOD]

# ONECLICK WORKFLOW
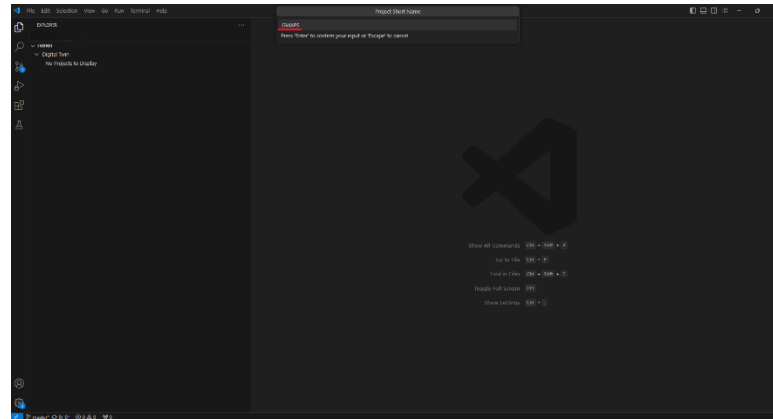


e. Observe your Project is created under **Digital Twin** also your scripts, userconfigs, API configs must be blank.
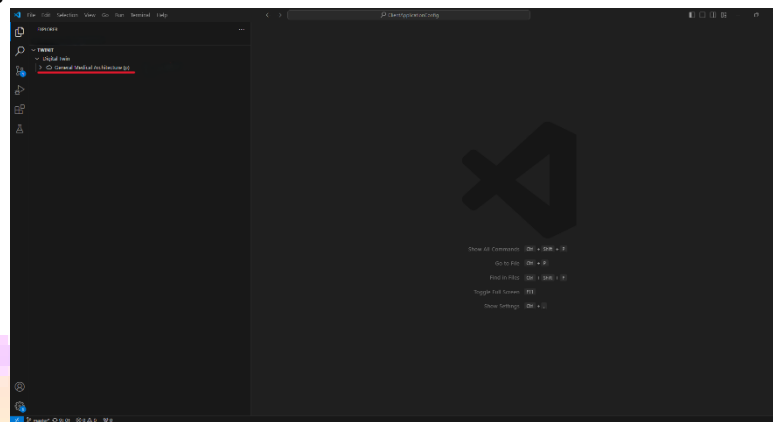


f. Creating a projectsetup,
   i. Under Oneclick project > right click on Script
   ii. Select > Create New Script > Enter Script name as "ProjectSetup"
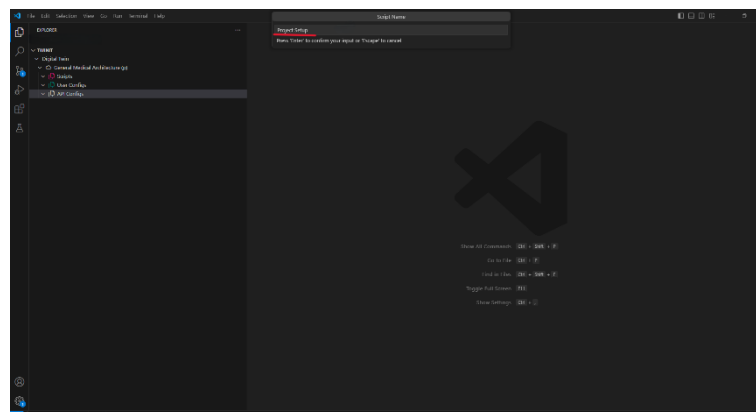


   iii. Enter Project description [Eg: Creating a project setup]

---

# ONECLICK WORKFLOW
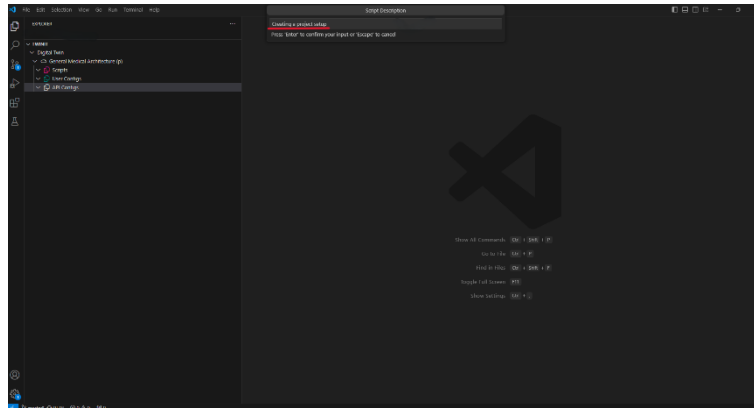


iv. Enter Project script short name
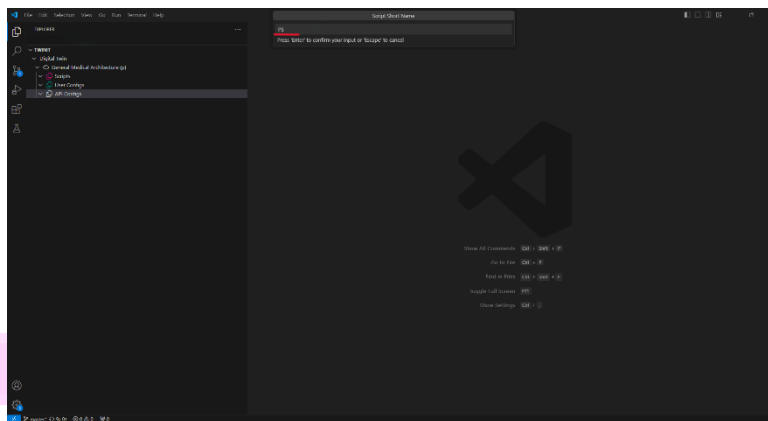


v. Enter Project Script User Type
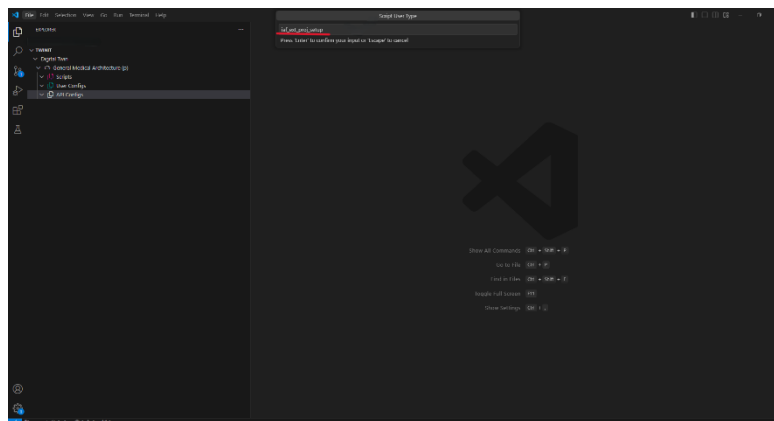


vi. Verify the Projectsetup is display under scripts in "Project Name".

---

# ONECLICK WORKFLOW
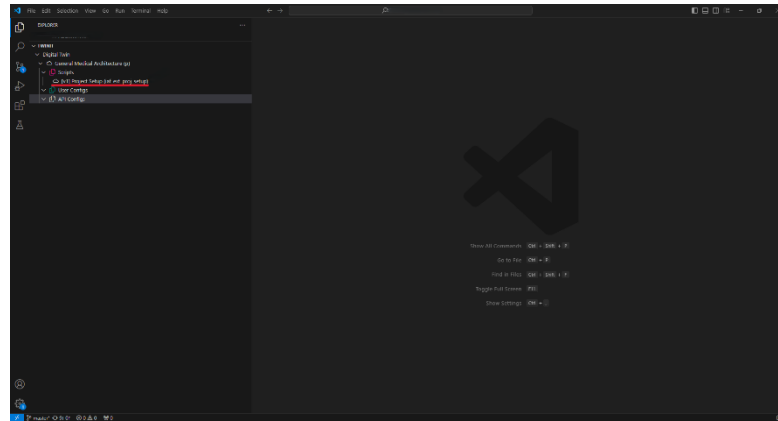


vii. Create a copy of project setup script from follow the below navigation
1. Path : [Twinit-IPA-Core-Digital-Buildings/scripts/ProjectSetup.js at master · Invicara/Twinit-IPA-Core-Digital-Buildings (github.com)](https://github.com)
Name: "Projectsetup.js"
2. Copy the entire script and paste in your project setup script (iaf_ext_proj_setup) and save.
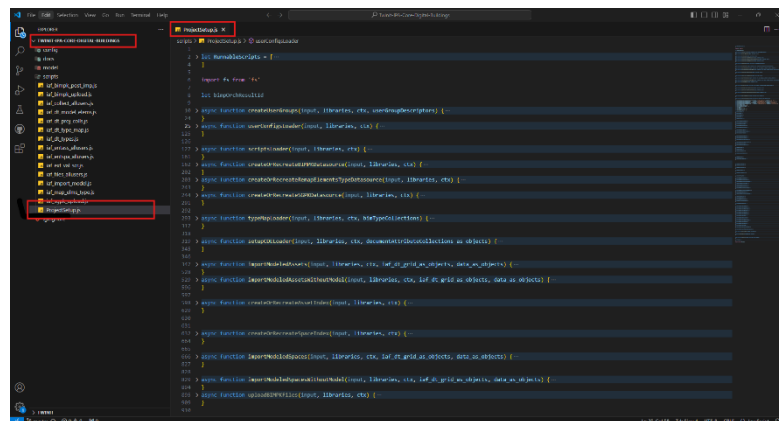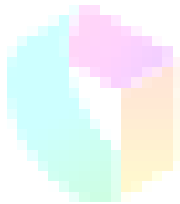


viii. Now Run the ProjectSetup script
1. Right click > Select Run Script
2. Select Oneclick Project Setup option from Select a Script to Run panel.

---

# ONECLICK WORKFLOW
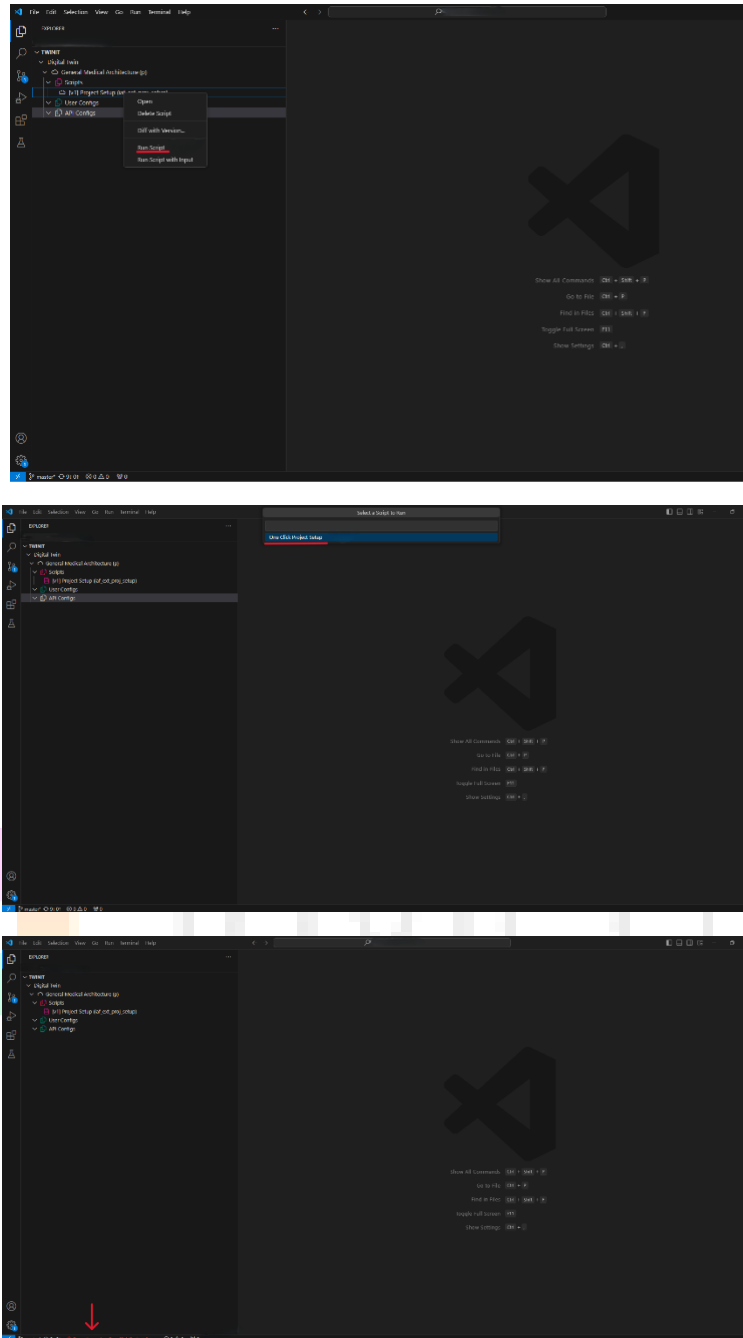






3. Select Config_sheet from folder repository location
   Path : [Twinit-IPA-Core-Digital-Buildings/config/Config_Sheet.xlsx at master · Invicara/Twinit-IPA-Core-Digital-Buildings (github.com)](https://github.com)
   Name: "Config_sheet"

---

# ONECLICK WORKFLOW
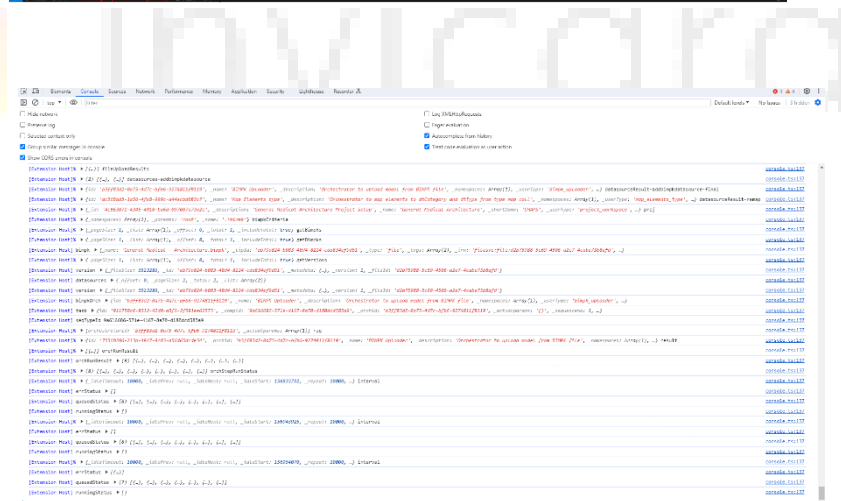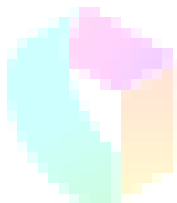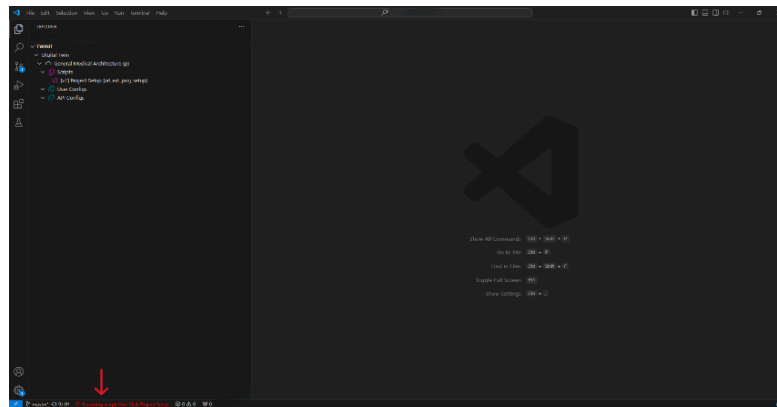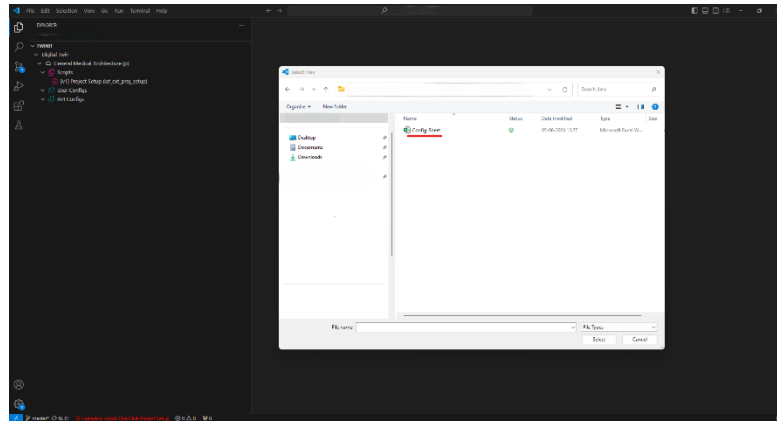






4. Once Project is successfully completed, it will load all your scripts and Userconfig that has been mentioned in config_sheet will be displayed under project.

---

# ONECLICK WORKFLOW
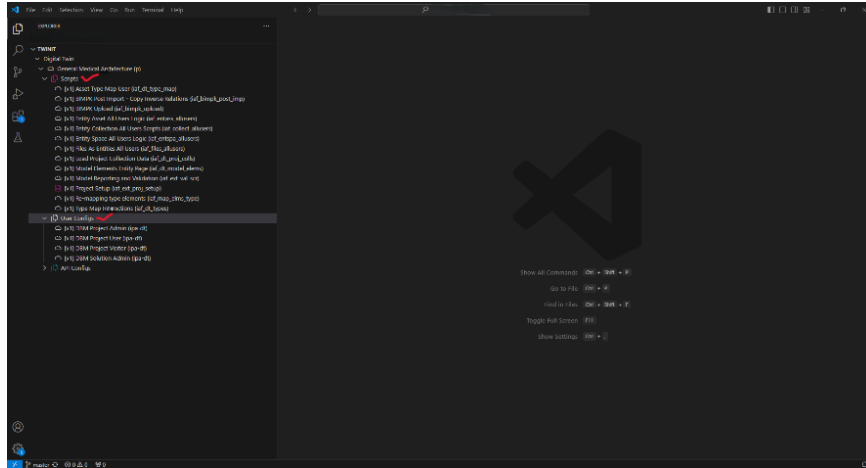


10. Project Visual in front end with multiple user role and all feature in "OneClick".
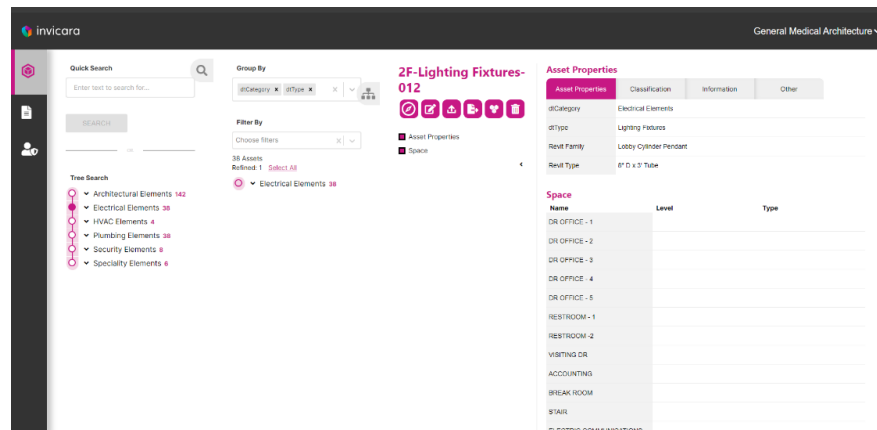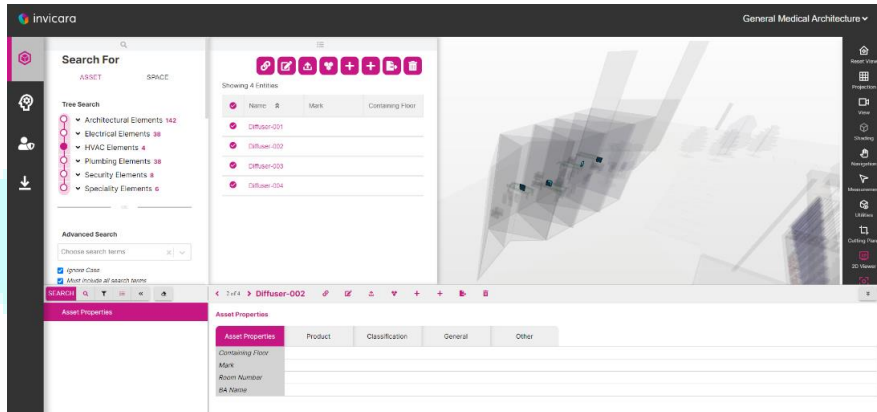
# ONECLICK WORKFLOW
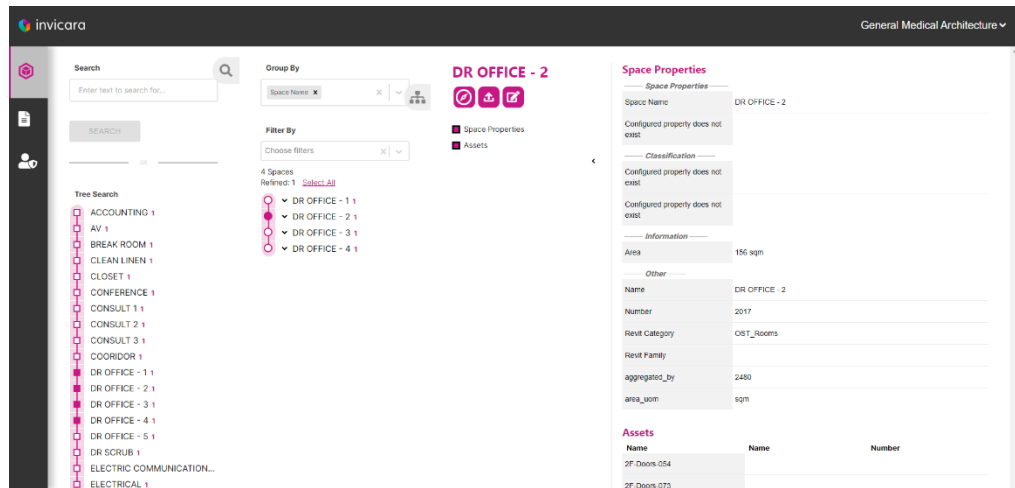


11. Config Sheet Preparation

    a. Add path "Script Path" and "BIMPK Path" under Path Sheet.
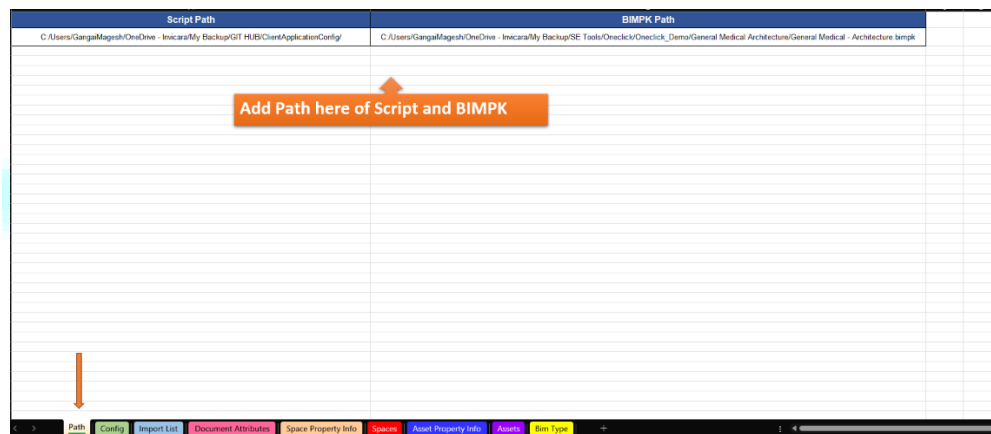


    b. Update the "Config" sheet with usergroup access and other handler access.
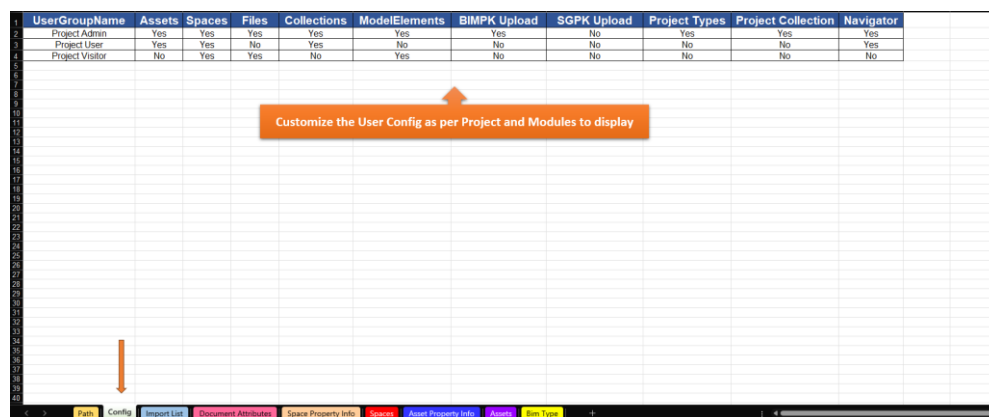
c. Update "Import List" sheet for the desired Asset Twin setup.



d. Fill the BIMType, Asset, Space, Asset Property, Space Property and Document Attributes as per the condition needs.

# ONECLICK WORKFLOW



Add Project Asset Properties here as Asset sheet



Add Project Asset List here



Add Project Space Properties here as Space sheet

# ONECLICK WORKFLOW



12. List of Use Cases Conditions for Config Sheet preparations

   i. If the ModelImport is marked as "No", BIMType should be "No" and under Path sheet "BIMPK Path" should be "Empty".

   ii. If the ModelImport is marked as "Yes", BIMType should be "Yes" " and under Path sheet "BIMPK Path" should be "Filled".

   iii. If the Asset is marked as "No", Asset and Asset Property should be "Empty"

   iv. If the Asset is marked as "Yes", Asset and Asset Property should be "Filled"

   v. If the Space is marked as "No", Space and Space Property should be "Empty"

   vi. If the Space is marked as "Yes", Space and Space Property should be "Filled"

   vii. If the "DocumentAttributes" is marked as "No", DocumentAttributes should be "Empty"

   viii. If the "DocumentAttributes" is marked as "Yes", DocumentAttributes should be "Filled"