**Script Documents**

Bash script environment setup:

Common Task

1. **nano ~/.bashrc**

2. add line **export TD\_PATH="/home/zahid/client-rnd"**

**PATH="$PATH:$TD\_PATH/dev-booster/scripts"**

3. **chmod -R 744 $TD\_PATH/dev-booster/scripts**

**Uses of bash-script**

**1. script-module-with-menu.sh**

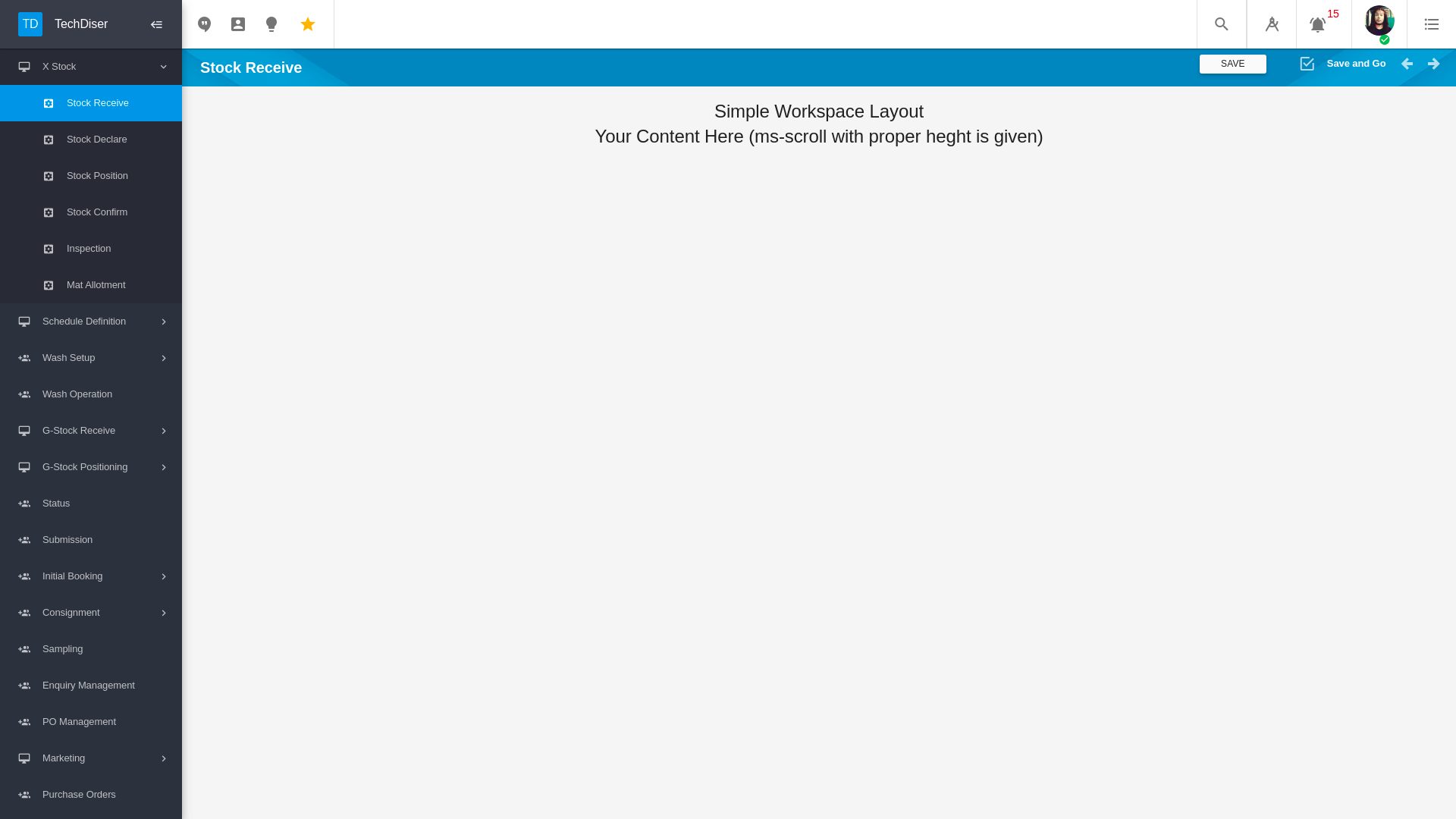
Use this script to make a menu group with multiple menus and sample layouts.

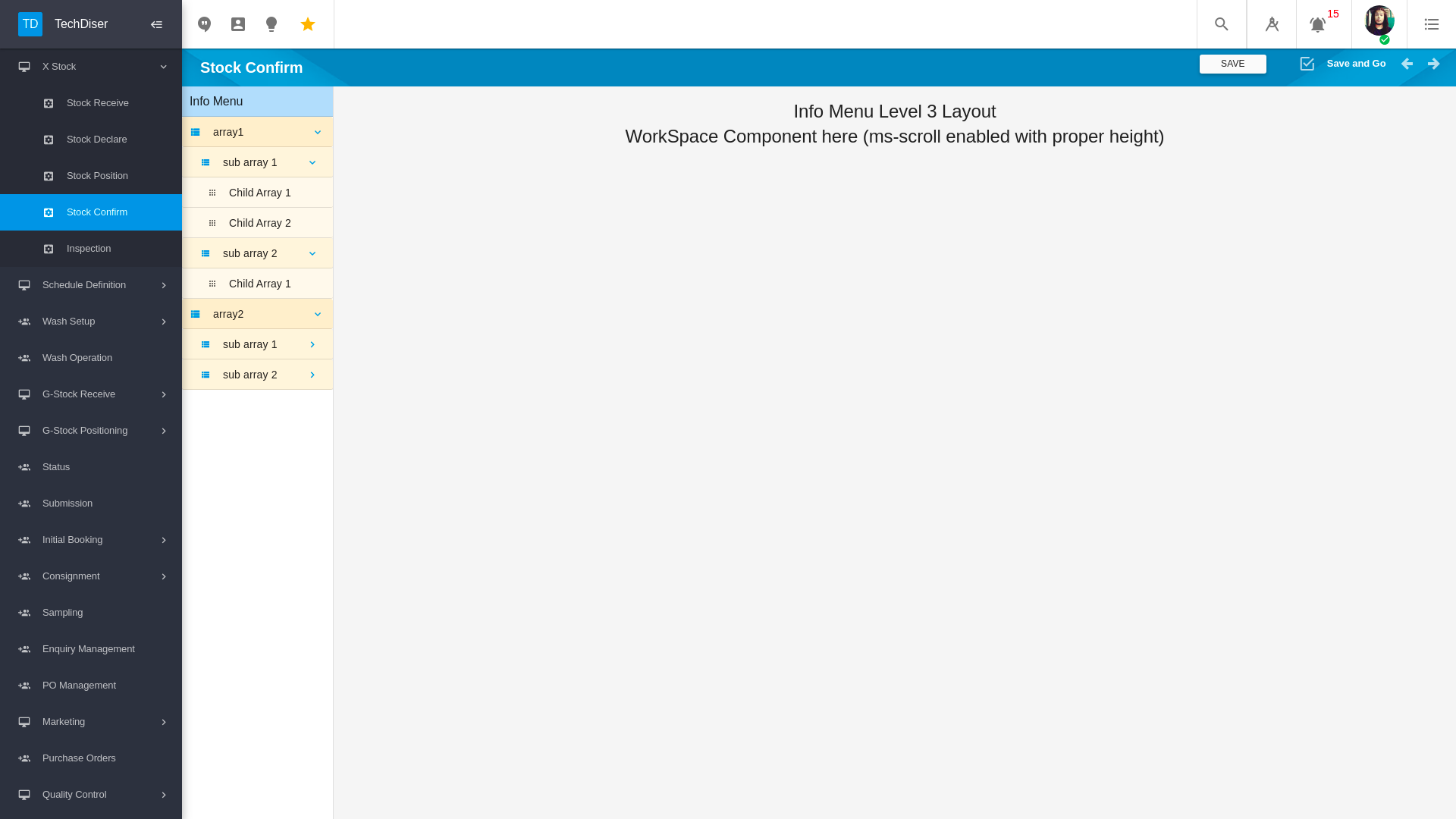
This script can cope up with any existing work

This script can be run several times for making new menus without interrupting existing one

***sample-layout***

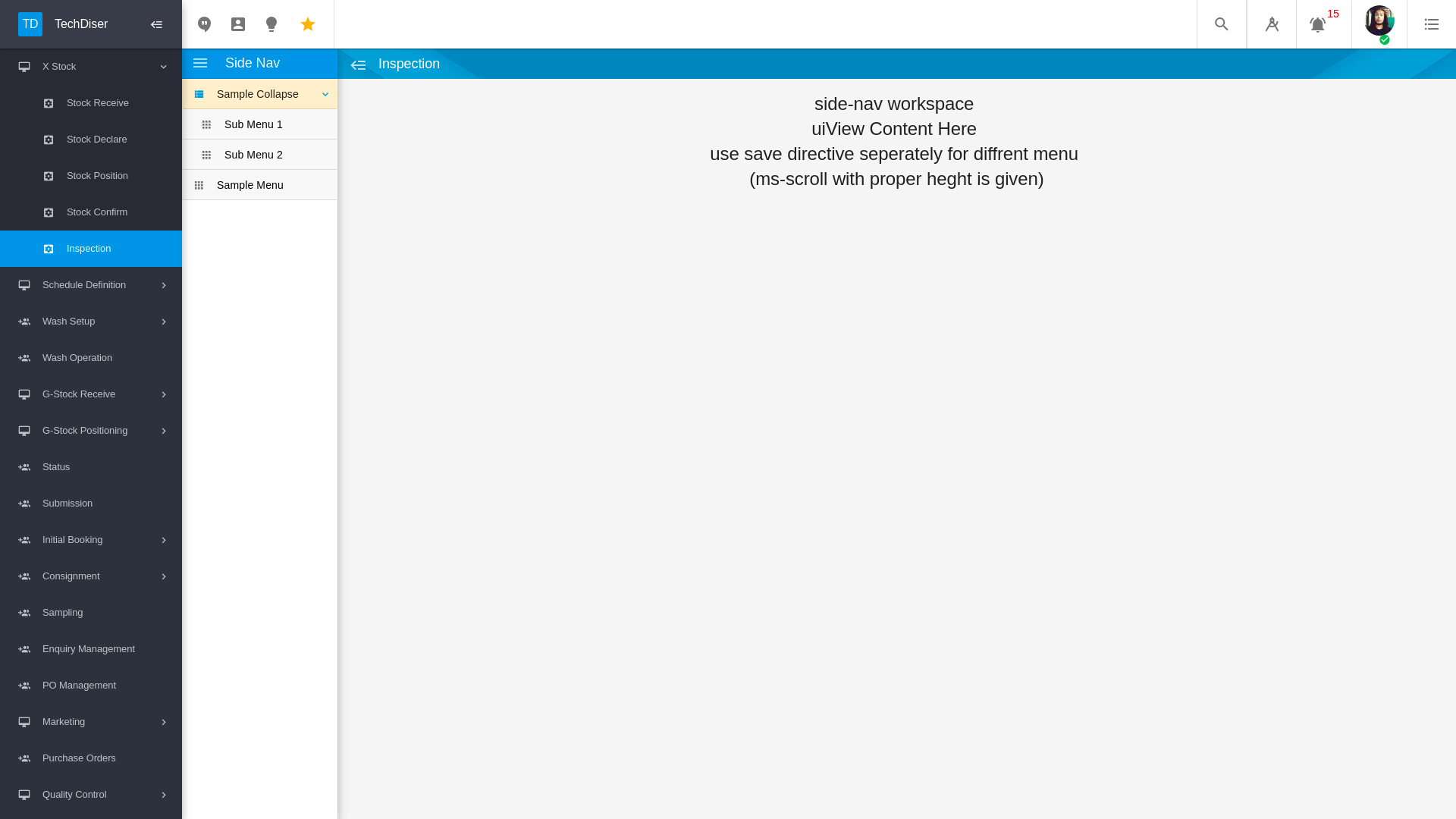
1. simple-workspace
2. info menu workspace-level(1-3)
3. side nav workspace
4. sub menu workspace
5. default-view

**simple-workspace:** This layout is consist of a header with title and save directive. Ms-scroll is given with proper height

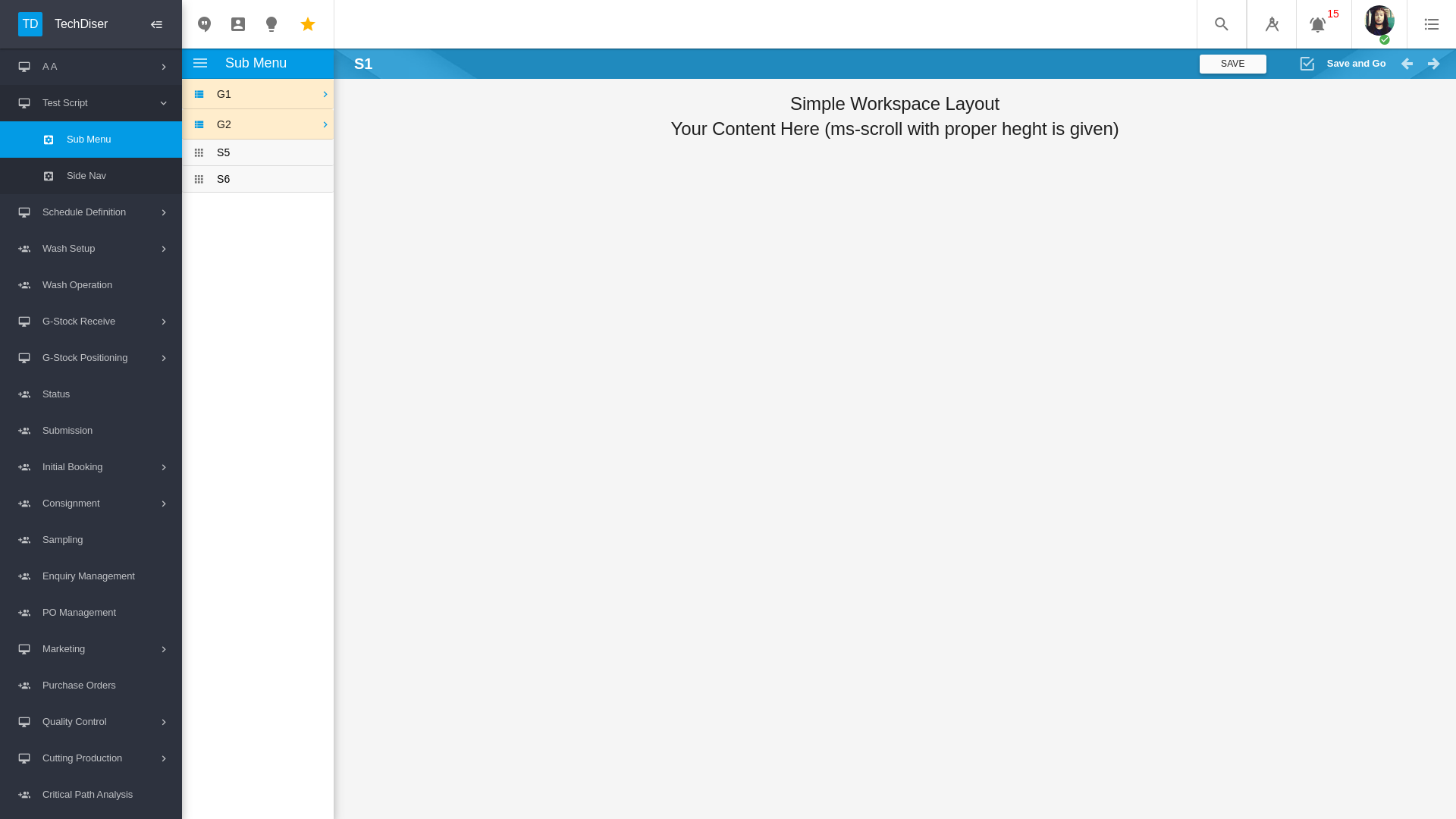
**info-menu-workspace:** This layout is consist of a header with title and save directive. Info menu with sample data structure and corresponding function. Available level options (1-3) Ms-scroll is given with proper height

**side-nav-workspace:** Collapsible side nav. Sample menu and state is given. No save directive because it will be added from sub menu individually.

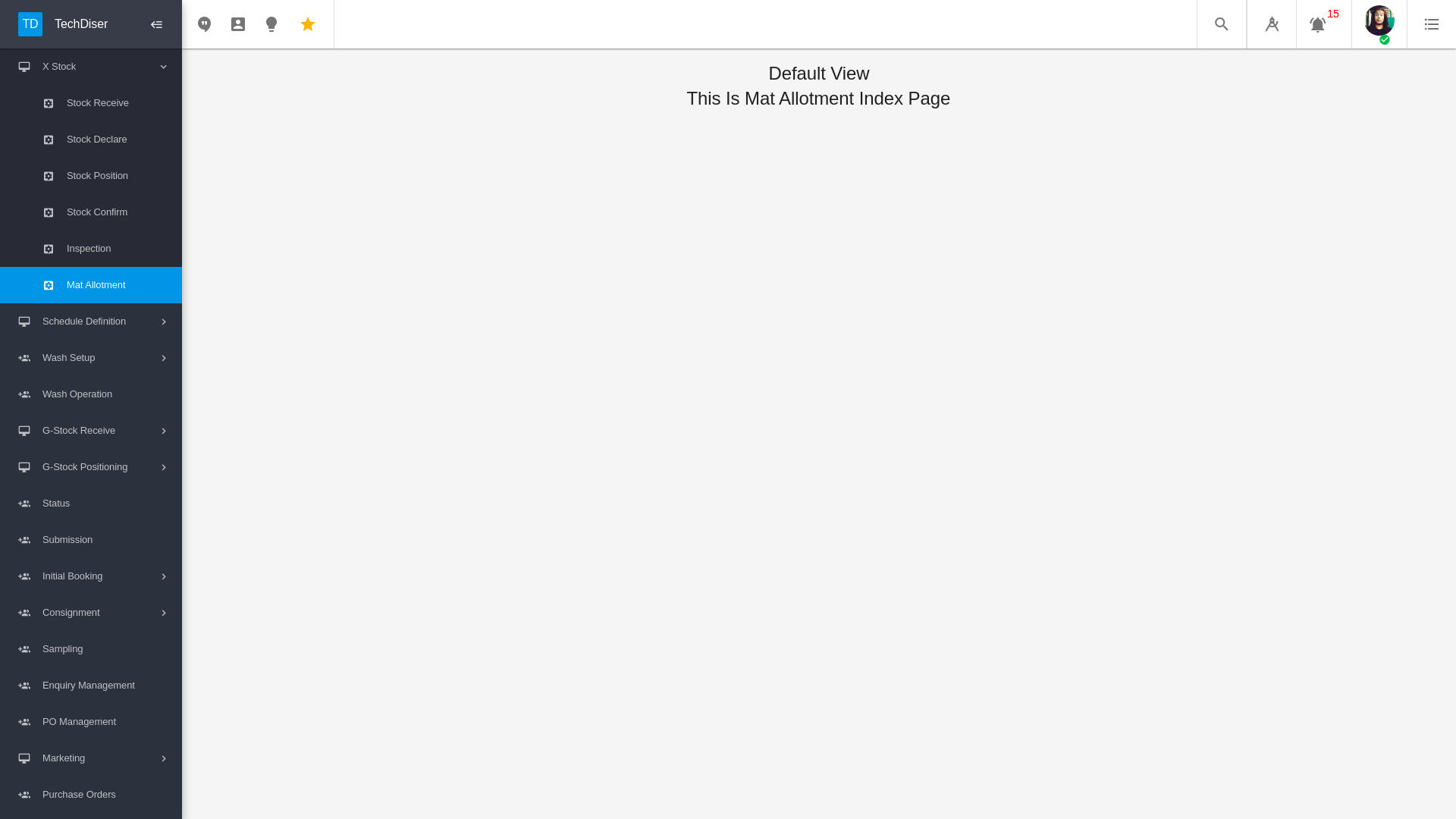
To add sub menu in this layout use script-sub-menu.sh



**sub-menu-workspace:** Fixed side nav. No save directive because it will be added from sub menu individually. To add sub menu in this layout use script-sub-menu.sh from menu directory



**default-view:** Simple html and controller file is given no layout design will be provided.



Required input

1. module name
2. menu names
3. layout for each menu

This script can be executed from any location

To add new menu in existing menu-group then only give module-name and new menu names.

No occurrence will be held if existing menu name is given

**Required Instructions**

|  |  |  |
| --- | --- | --- |
| Marker | Snippet | Where to place |
| //CODE\_GENERATOR\_MARKER\_STATE | snpt-instruction-state | module file if exists |
| //CODE\_GENERATOR\_MARKER\_SAVE\_ITEM | snpt-instruction-saveItem | module file if exists |
| //CODE\_GENERATOR\_MARKER\_APP\_NAME | snpt-instruction-constantAppName | index.module.js |

**Note: Script to be run from the main folder**

**2. script-sub-menu.sh**

This script is used for creating sub-menus of a menu.

It will creates

1. sub-menu-state
2. sub-menu folder
3. sub-menu html and controller file
4. add this menu to controller side-nav data structure (For side-nav/sub-menu workspace)

Run this script from folder where you want to add sub menus

To add new sub-menu in existing menu then only give new sub menu names.

No occurrence will be held if existing menu name is given

**Markers required**

|  |  |  |  |
| --- | --- | --- | --- |
| Marker | Snippet | Where to place | Reference |
| //CODE\_GENERATOR\_MARKER\_STATE | snpt-instruction-state | module file |  |
| //CODE\_GENERATOR\_MARKER\_SUB\_MENU | snpt-instruction-subMenuCtrl | controller file |  |
| <!-- CODE\_GENERATOR\_MARKER\_HTML\_SUB\_MENU --> | snpt-instruction-subMenuHtml | Html file where sub menu will be created |  |
| <!-- CODE\_GENERATOR\_MARKER\_HTML\_SIDE\_NAV --> | snpt-instruction-sideNavHtml | Html file where side nav will be created |  |

**3. script-id-gen.sh**

This script is used for **level 1** ID generation.

It will creates:

1. controller file with corresponding ID GEN code if required
2. data json file and data structure if not
3. json service file if not
4. Add service and task name in constants

Run this script from menu/sub-menu/tab directory in which you want to generate id

required inputs:

1. module name
2. menu/sub-menu/tab name
3. root id prefix
4. root id without prefix
5. attribute name

**Markers required**

|  |  |  |
| --- | --- | --- |
| Marker | Snippet | Where to place |
| //CODE\_GENERATOR\_MARKER\_CONSTANTS | snpt-instruction-constantAdd | index.constants.js |
| //CODE\_GENERATOR\_MARKER\_ID\_GEN | snpt-instruction-idGeneration | id generation controller file |
| //CODE\_GENERATOR\_MARKER\_ATTRIBUTE\_FUNCTION\_EXPOSE | snpt-instruction-specdefFunctionExpose | json service file if exists |
| //CODE\_GENERATOR\_MARKER\_ATTRIBUTE\_FUNCTION\_ADD | snpt-instruction-specdefFunctionAdd | json service file if exists |

**4. script-tab-generator.sh**

This script is used for creating tabs (separate state) for a menu.

It will creates

1. tab-state
2. tab folder
3. tab html and controller file
4. and tab item in html file

Run this script from folder where you want to add tabs

To add new tab in existing menu then only give new tab names.

No occurrence will be held if existing menu name is given

**Markers required**

|  |  |  |
| --- | --- | --- |
| Marker | Snippet | Where to place |
| //CODE\_GENERATOR\_MARKER\_STATE | snpt-instruction-state | module file |
| <!-- CODE\_GENERATOR\_MARKER\_TAB\_ITEM →  place this marker in between  <md-tabs> </md-tabs> | snpt-tabGeneratorHtml | Html file where tab will be created |

**Conventions those followed by scripts:**

**Naming conventions:**

if module name is menu-group

menu name is menu-one, menu-two

menu-one sub-menu is sub-menu-one, sub-menu-two

And sub-menu-one have ID generation

then file structure will be

Folder hierarchy

* *menu-group*
  + *menu-one*
    - *sub-menu-one*
      * *sub-menu-one.html*
      * *sub-menu.one.ctrl.js → controller name→ menuGroupMenuOneSubMenuOne*
    - *sub-menu-two*
      * *sub-menu-two.html*
      * *sub-menu.two.ctrl.js → controller name→ menuOneSubMenuTwo*
    - *menu-one.html*
    - *menu.one.ctrl.js → controller name- menuGroupMenuOne*
  + *menu-two*
    - *menu-two.html*
    - *menu.two.ctrl.js → controller name- menuGroupMenuTwo*
  + *menu-group.module.js*
* *data*
  + *menu-group*
    - *sub-menu-one.json*
* *specdef*
  + *menu-group*
    - *sub-menu-one.json.js → factory name -menu\_group\_sub\_menu\_one\_json*

***Item Confirmations:***

***app-name :*** *menu-group*

***menu-one state:*** *menu-group.menu-one*

***menu-two state:*** *menu-group.menu-two*

***sub-menu-one state:*** *menu-group.menu-one.sub-menu-one*

***sub-menu-two state:*** *menu-group.menu-two.sub-menu-two*

***index.constants***

*service-name: MENU\_GROUP*

*task-name: sub\_menu\_one*

All naming conventions can be editable during script execution.

**Others conventions**

No existing files will be modified without code\_generator\_marker

in case of file missing script will automatically create file and folders

All task works independently

Logs

[I] – Information regular color

[A] – Added cyan color

[M] – Modified blue color

[W] – Warning yellow color

[E] – Error red color

highlighted file/folder magenta color