

# Karthikeyan Knowledge Transfer

## Freshgrc

Our FreshGRC product aims to democratize the GRC industry as a cost effective alternative to most GRC products. An automated GRC (Governance, Risk & Compliance) platform like ours, will make it easy to adopt, approach and automate the process flow of compliance. Our GRC solution helps organization in practicing Compliance as a culture, with improved productivity, better process, detect variances, fix gaps, proactive measures and avoid paying fines and reputation impacts.

## Front end technologies

Html, Css, Javascript, jQuery, Bootstrap, AngularJS.

## Version Control

Git.

## GitHub Repository Name

[https://github.com/fixnixinc/freshgrc\\_angular\\_bootstrap](https://github.com/fixnixinc/freshgrc_angular_bootstrap).

## Intall front end in local system

1. Open Terminal.
2. Install git into your machine.
3. Copy and paste the command => `git clone https://github.com/fixnixinc/freshgrc_angular_bootstrap.git`
4. Install npm => `sudo npm install`
5. Install grunt => `sudo npm install -g grunt-cli`
6. After all installation open ( `freshgrc_angular_bootstrap` ) folder in Terminal.
7. run the grunt to view app in local systyem => `grunt dev`
8. It will run on localhost:4000

## Freshgrc modules

1. Audit
2. Asset
3. Risk
4. Compliance
5. Policy
6. Fraud
7. Incident
8. Control
9. BCPM

## Git branches

- Perfect working Local branch => `develop-new`
- Perfect working Production branch => `azure-front`

## Merging branches in Git

1. First create new branch using following command => `git checkout -b`
2. After working merge the wored branch with `develop-new`, If there is any conflict resolve it otherwise merge the branch.
3. After merging check `develop-new` branch with others system.
4. After woking in others system merge the `develp-new` branch with `azure-front`
5. Make sure to pull in production server for added changes updated.

## Risk

- Risk module is used to prevent the risk for the assets.
- Eg: Software risk ==> renewal of the antivirus licence.

## Risk modules

1. Plan
2. Mitigate
3. Review
4. Report

## Credentials

To enter into risk module need to use the following credentials,

1. username for create plan => risk\_owner@fixnix.co | password => freshgrc\$123
2. username for mitigate risk => risk\_mitigator@fixnix.co | password => freshgrc\$123
3. username for review risk => risk\_reviewer@fixnix.co | password => freshgrc\$123

## Risk Plan: => Plan your risk

- app/modules/risk/manage/riskplan/riskplan.html
- app/modules/risk/manage/riskplan/riskplanCtrl.js
- app/modules/risk/manage/riskplan/riskplanModule.js
- app/modules/risk/manage/riskplan/riskplanRoute.js
- app/modules/risk/manage/riskplan/riskplanService.js

## Risk List: => Display all the risk in list

- app/modules/risk/risklist/risklist.html
- app/modules/risk/risklist/risklistCtrl.js
- app/modules/risk/risklist/risklistModule.js
- app/modules/risk/risklist/risklistRoute.js
- app/modules/risk/risklist/risklistService.js

## Risk Mitigation: => Mitigate the created risk

- app/modules/risk/manage/riskmitigation/riskmitigation.html
- app/modules/risk/manage/riskmitigation/riskmitigationCtrl.js
- app/modules/risk/manage/riskmitigation/riskmitigationModule.js
- app/modules/risk/manage/riskmitigation/riskmitigationRoute.js
- app/modules/risk/manage/riskmitigation/riskmitigationService.js

## Risk Review: => Review mitigated risk

- app/modules/risk/manage/riskreview/riskreview.html
- app/modules/risk/manage/riskreview/riskreviewCtrl.js
- app/modules/risk/manage/riskreview/riskreviewModule.js
- app/modules/risk/manage/riskreview/riskreviewRoute.js
- app/modules/risk/manage/riskreview/riskreviewService.js

## Risk Register: => Display all the reports for risk

- app/modules/risk/manage/riskregister/riskregister.html
- app/modules/risk/manage/riskregister/riskregisterCtrl.js
- app/modules/risk/manage/riskregister/riskregisterModule.js
- app/modules/risk/manage/riskregister/riskregisterRoute.js
- app/modules/risk/manage/riskregister/riskregisterService.js

## Working flow

\* --> Before creating risk need to create asset from asset module. Plan

1. Login as risk\_owner@fixnix.co
2. While add risk, Select risk i)owner as risk\_owner, ii)mitigator as risk\_mitigator, iii)reviewer as risk\_reviewer
3. Library need to select only ISO27001(other options are not work)
4. Scoring types select any one type and give your metrics. You have another option to update this in mitigator part.
5. POST --> http://localhost:3000/api/v1/risk/risk

## Mitigator

1. Login as risk\_mitigator@fixnix.co
2. Controls will be created based on include function.
3. Post request for control
4. POST --> http://localhost:3000/api/v1/risk/mitigation
5. Questions are displayed based on selected control and asset types.
6. Not necessary to answer all of the questions.
7. Update the scoring types.\*select the one previously selected in plan.
8. Post request url for submit mitigation
9. POST --> http://localhost:3000/api/v1/risk/mitigations/create\_mitigation\_control\_details

## Review

1. Login as risk\_reveiwer@fixnix.co
2. Make sure to submit Loss event with filled data.
3. Post request for loss event
4. POST --> [http://localhost:3000/api/v1/risk/reviews/create\\_loss\\_event](http://localhost:3000/api/v1/risk/reviews/create_loss_event)
5. From the plan selected Risk Kri will be displayed.
6. Need to update the current value for all Risk Kri.
7. Review process will be done infinite time.
8. After updated Risk Kri, it will be displayed on top.
9. Post url for submit Risk Kri
10. POST --> [http://localhost:3000/api/v1/risk/reviews/create\\_review](http://localhost:3000/api/v1/risk/reviews/create_review)

## Report

1. After reviewed risk, Report page will be generated.
2. It will show assets, control type, control nature, asset value and Responsible person name based on risk id.
3. There is option to get all the wrapped data as a pdf or zip file formats.

## Wordpress

Create a free website or easily build a blog on WordPress.com. Hundreds of free, customizable, mobile-ready designs and themes. Free hosting and support.

## Fixnix main website

1. <https://fixnix.co/>
2. Type the url to edit this webpage <https://fixnix.co/wp-admin>
3. Wordpress console will be open then use the following credentials
4. Username: karthikeyan@fixnix.co
5. Password: freshgrc\$123
6. Use the following documentation for further development process.
7. <http://themes.muffingroup.com/betheme/documentation/>

## Freshgrc Product page

1. <http://www.freshgrc.com/>
2. Type the url to edit this webpage <http://www.freshgrc.com/wp-admin>
3. Wordpress console will be open then use the following credentials
4. Username: karthikeyan@fixnix.co
5. Password: freshgrc\$123
6. Use the following documentation for further development process.
7. <https://vc.wpbakery.com/features/wordpress-editor/>

## Fixnix blog

1. <https://blog.fixnix.co/>
2. Type the url to edit this webpage <https://blog.fixnix.co/wp-admin>
3. Wordpress console will be open then use the following credentials
4. Username: karthikeyan@fixnix.co
5. Password: freshgrc\$123

## Initial screen design

1. Initial screen design has the following modules
  - Signin
  - Signup
  - Reset password
  - Confirm email
  - Login
  - Set password
2. For all the above modules has the common header and footer
3. They are mentioned as following files, also used common styles.
  - common-header.html
  - common-footer.html
  - style.css

## HTML webpages

1. <https://nixaudit.com/>
  - <https://github.com/fixnixinc/freshGRC.com>
2. <http://jumpstraps.com/>
3. <http://www.freshgrc.co/>
4. <http://shanmugavel.com/>
  - [https://github.com/fixnixinc/shan\\_website](https://github.com/fixnixinc/shan_website)
5. <https://nixrisk.com/>