**Code sanitation rules**

**Git and Github:**

The following process must be adhered to strictly for contributing to our code repo. This will help maintain the sanity of the code.

* Fork the repo - https://github.com/YogeshBarai/GL-techDocs2.0 (Fork button is on the right hand side on top)
* This should show up in your Github account home page.
* Use this fork to work on and make any code changes that you would be making.
* Click on the green “Code” button and use the https link to clone it to your computer.
* At all times, please work on this cloned repo on your computer.
* We have 6 branches - main (the production code will reside here), front\_end (front end team will add their code here), back\_end (back end team will add their code here), deployment (deployment team will add their code here), dev (this will be our development pipeline where we will merge all three code branches) and finally test (we will first test our code in this branch by setting up a test pipeline).
* Please switch to the branch that you are a part of - front\_end, back\_end, deployment.
* Make all your code changes here in this branch.
* Also before you start making any changes, please pull all the updates from the [main repo](https://github.com/prernawaghray/GL-TechDocs.git) in your branch to avoid any conflicts.
* Once the work is done, please push all the commits to your fork (the one you set up initially).
* From your fork, please submit a Pull Request to the main repo to the branch that you are working on i.e; if you are working on front\_end code, then please push your code to <https://github.com/YogeshBarai/GL-techDocs2.0> repo’s front\_end branch only and not to the main branch.
* If everyone is good with the code, we can have a peer reviewed merge.
* How to update your clone each time you code?
  + Go to the folder where all the git code is present
  + git remote add <https://github.com/YogeshBarai/GL-techDocs2.0>
  + git pull upstream main (this will sync your cli with the main repo)
  + git push origin main (this will sync your gui with the main repo)

**Code:**

* Include comments, docstrings and explanations wherever necessary to comprehend your code easily
* Add input and output for your functions in the docstrings so that everyone knows what to expect from that particular function
* Follow [PEP-8 rules](https://realpython.com/python-pep8/) for Python coding

**For Frontend:**

TechDocs fronted uses a number of open source projects

* [Bootstrap](https://getbootstrap.com/) : great UI boilerplate for modern web apps
* [Python Flask](https://flask.palletsprojects.com/en/2.2.x/) : For URL Based Routing & Rendering HTML Templates
* [jQuery](http://jquery.com/) - UI/UX Actions/Events on HTML Documents
* [LatexJS](https://latex.js.org/) - JS Library to preview latex document

**Installation**

* git clone <https://github.com/YogeshBarai/GL-techDocs2.0>
* cd GL-TechDocs/frontend
* # Check the code in start.sh and make sure if port is 56733 for dev purpose and 80/443 for production purpose
* sudo bash start.sh
* # Docker container[techdocs-frontend] will be built & it will run. Get into docker and install node packages
* docker exec -it techdocs-frontend bash
* npm install
* exit
* Check [http://localhost:56733](http://localhost:56733/) in the browser.

**For Backend:**

* Please make a folder under the services folder according to your task (example: “FileManager” folder), and add your code to the respective folders.
* Make necessary changes to your code to make it a flask blueprint (<https://flask.palletsprojects.com/en/1.1.x/blueprints/>) and import & register your blueprint in app.py.
* Please refer to sampleBlueprint.py for further clarity.(in services/SampleBlueprint)

Requirements:

TechDocs backend is built on these open source softwares:

* [Python] : General purpose coding
* [Python Flask](https://flask.palletsprojects.com/en/2.2.x/) : For API endpoint handling
* [SQLAlchemy] : SQL Operations
* [MySQL] : Database
* [Mailjet] : SMTP server

#On your AWS EC2 instance, run the following commands:

* cd /home/ec2-user
* mkdir -p /techdocs\_filesystem/log
* touch /techdocs\_filesystem/log/filemanager.log
* git clone <https://github.com/YogeshBarai/GL-techDocs2.0>
* cd GL-TechDocs/backend
* #Update the docker-compose.yml file line#11 with below:
* /home/ec2-user/techdocs\_filesystem:/tmp

docker-compose up --build --scale app=3 -d