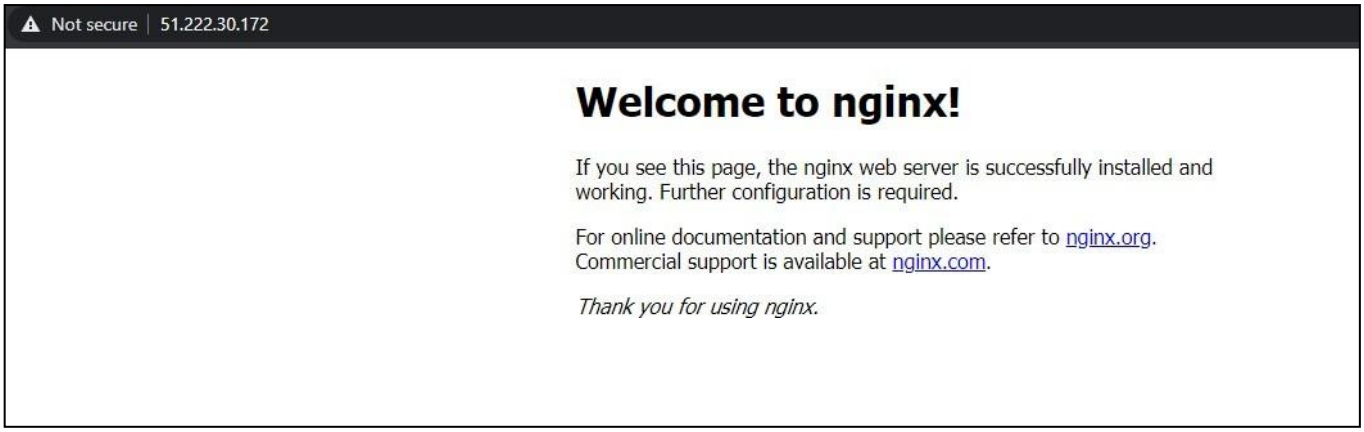
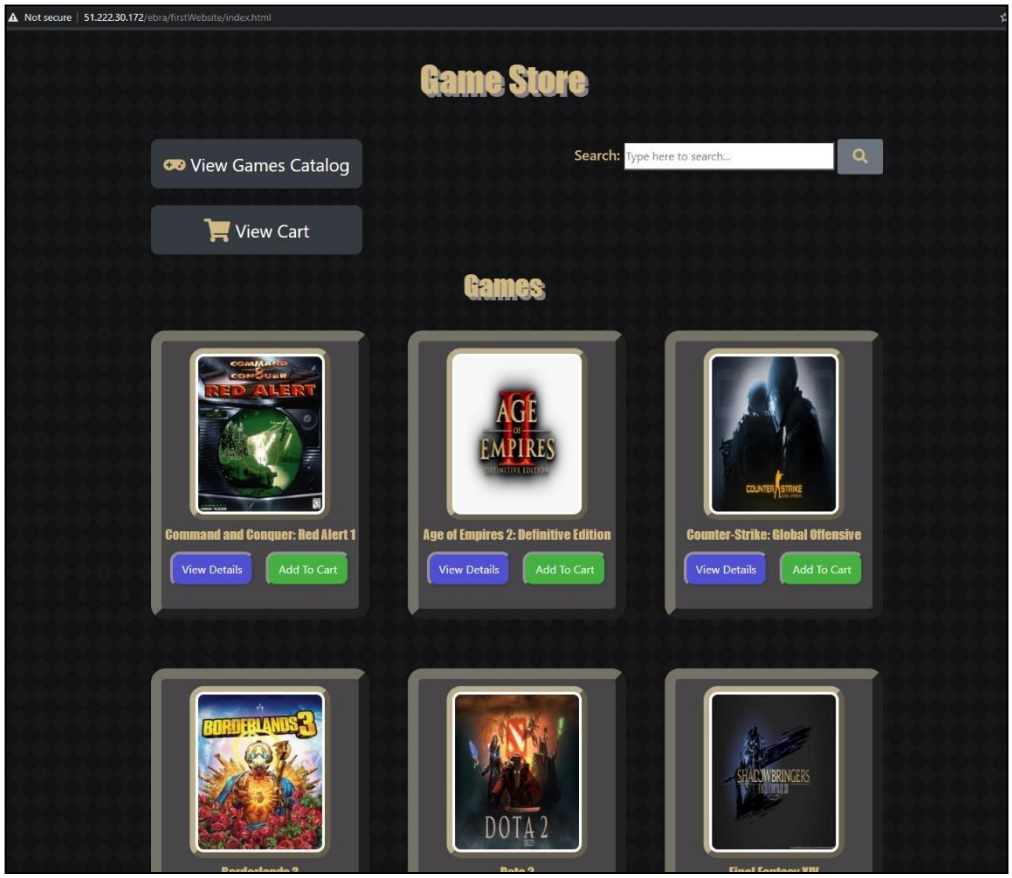
Ebrahim Vericain Unix (00002) Semester 3 Team: ACEUNIX
IP: 51.222.30.172

Git Repo: https://github.com/wornsoulbit/ACEUNIX

Screenshot of nginx welcome page on our server:

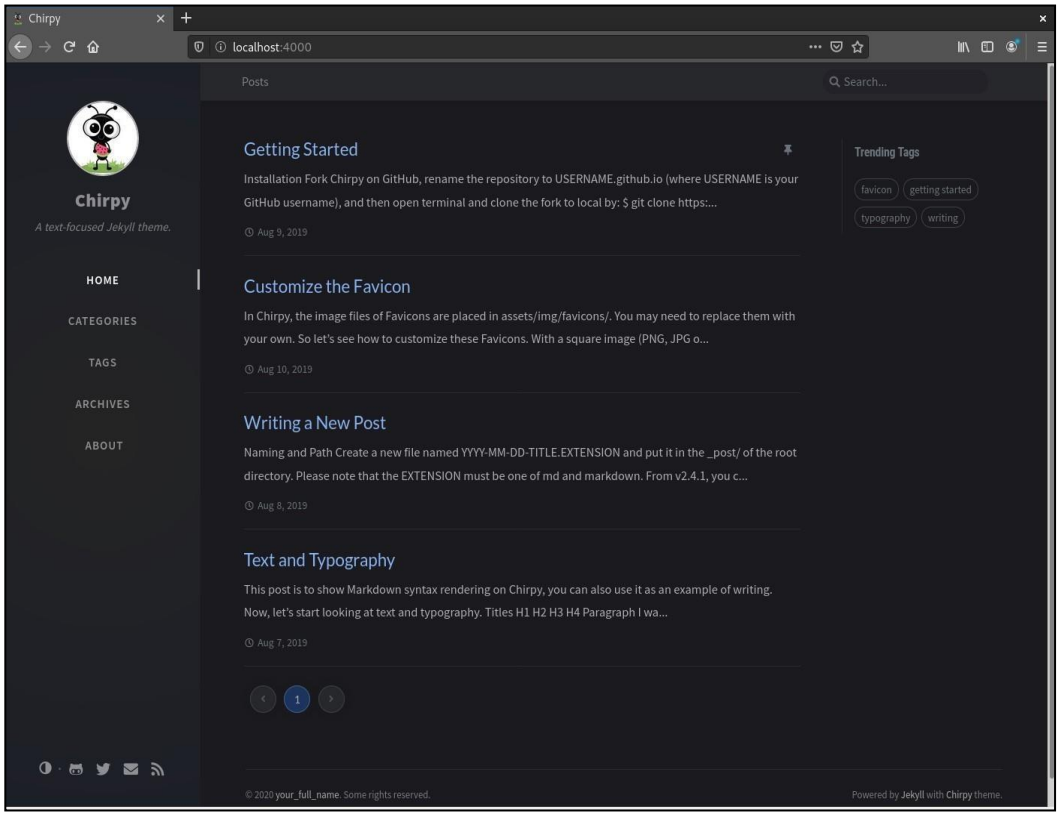


Screenshot of my client-side website deployed on our server:

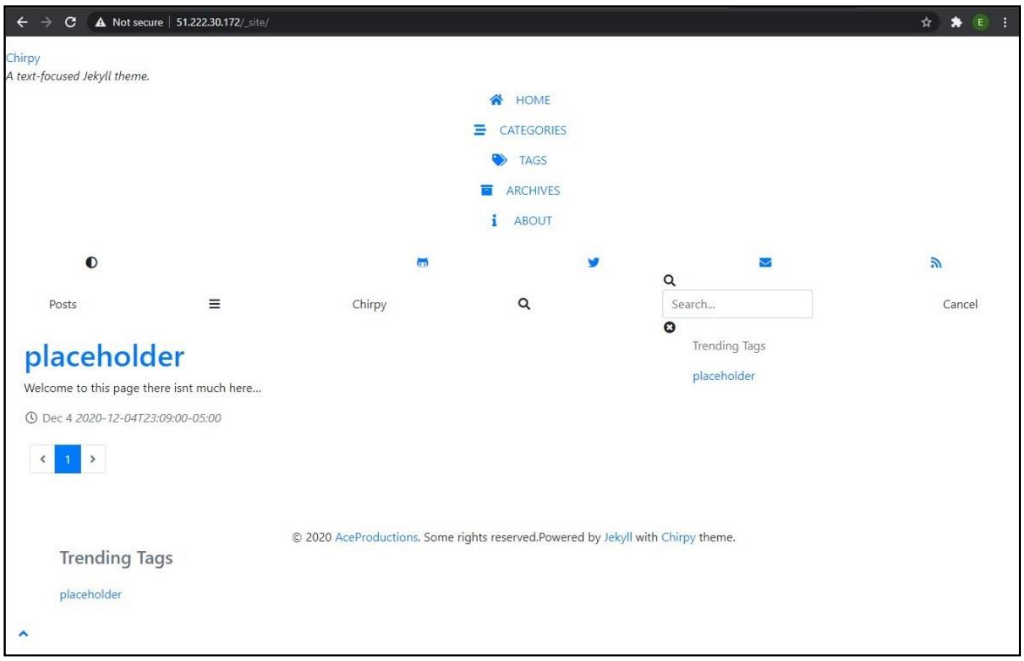


Git Repo: https://github.com/wornsoulbit/ACEUNIX

Screenshot my themed Jekyll site:



Screenshot of themed Jekyll group website:



Git Repo: https://github.com/wornsoulbit/ACEUNIX

2.

We were able to complete:

- Installing the nginx web server and checking if it works

- Deploying our individual client-side website to our server and checking if it works

- Setting up git (creating repository, adding collaborators, learning git commands, and seeing how to resolve conflicts with =========, >>>>>>>> and <<<<<<<<< within files)

- Installing Jekyll package then installing Jekyll bundler with the gems (individual)

- Creating a Jekyll site then checking if it works on my localhost:4000 (individual)

- Adding a theme to my Jekyll site (individual)

- One of my group members created a new Jekyll project for all of us to work on and edit

- We found a way to deploy our new Jekyll project to our server

- Automatic deployment (a script that gets the current version of out Jekyll site through git, checks if there are any changes, then deploys the current version of website project on to the server (_site))

- Execute the script every 5 mins by using systemd

3.

We are blocked at:

- When deploying the Jekyll site on our server, the layout of the theme we chose completely changed (as if some files were corrupted or something, I restarted the entire process and it did not change), as seen in the last screenshot above.

4.

For me, I would say our biggest challenge so far this week was trying to add and edit a theme on the Jekyll site. Also, creating the auto deployment script file took me a fairly long time.

Git Repo: https://github.com/wornsoulbit/ACEUNIX

5.

Commands for deploying my website:

*First downloaded and installed nginx

1. cd /var/www/html/
2. mkdir /var/www/html/ebra
3. cd /var/www/html/ebra
4. scp /firstWebsite.zip debian@51.222.30.172:/var/www/html/ebra
5. sudo unzip firstWebsite.zip
6. localhost:4000 in url to see if it's up and running

Setting up git repository:

1. Logged in into github
2. Created private repository
3. Chose GNU GPLv3 LICENSE
4. Added README file

Commands used for Git process (practice):

1. git pull
2. git push (me and my team member are working on different version of project (to intentionally create conflict))

*Went to file(s) that cause conflict then deleted =========, >>>>>>>> and <<<<<<<<

3. git push

Commands used for Jekyll process:

1. sudo apt-get install ruby-full build-essential zlib1g-dev (individual)
2. gem install jekyll bundler (individual)
3. jekyll new myStaticSite (individual)

   *downloaded website theme from jekyll site (jekyll-theme-chirpy), then put that theme folder into my jekyll project folder (individual)

4. bundle exec jekyll serve (in jekyll-theme project directory, to see if the theme worked with localhost:4000) (individual)

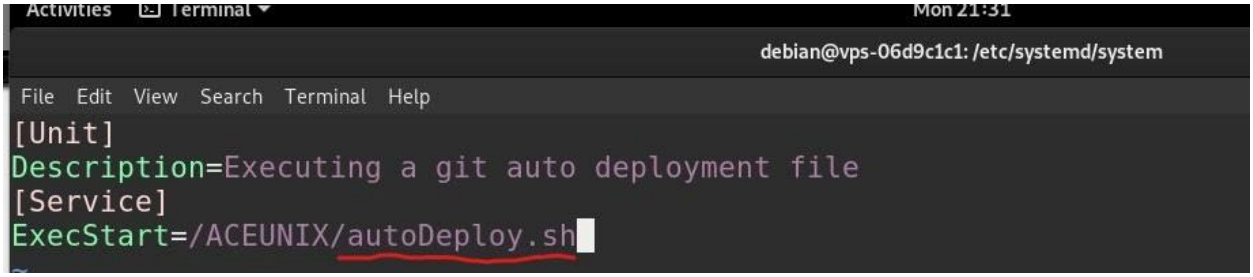
5. Here, one of my group members set up the group website then pushed it to github
6. git cloned our group website
7. edited files in the _posts folder in our website themed folder
8. cd jekyll-theme-chirpy/ (in server)
9. sudo bundle install (in server)
10. sudo bundle exec jekyll build (in server)
11. sudo cp _site /var/www/html (in server)

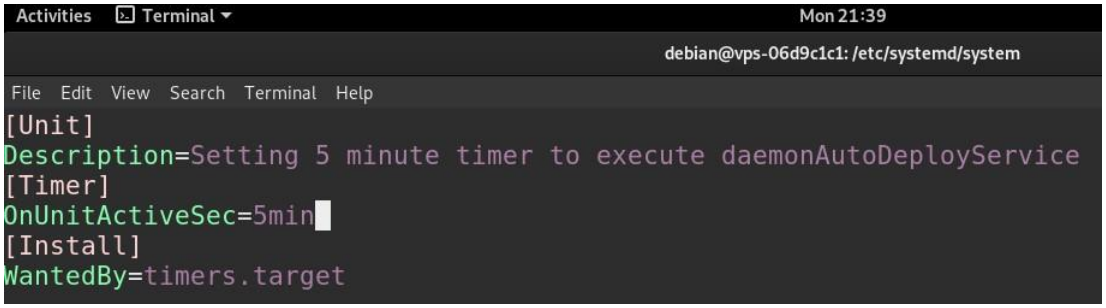
Git Repo: https://github.com/wornsoulbit/ACEUNIX

Creating auto deployment shell script:

```
#!/bin/bash

            #In order for this script to work you need to be in your local git
            repository (ACEUNIX directory)

            upToDateMessage=$(git pull)


            if ! [[ $upToDateMessage == "Already up to date." ]];
            then
                  #(1)building site
                  cd ~/ACEUNIX/jekyll-theme-chirpy
                  bundle exec jekyll build
                  #(2)copy static files to the correct location
                  sudo cp -R _site/* /var/www/html
            fi
```

Executing script every 5mins





systemctl enable daemonAutoDeployService.timer