**Progress Report**

Sprint 1:

* We created a requirements document and a document that contains a use case and use case diagram. Made group repository for Sprints and branches for each sprint. Shared fork of Augur with team. Created README.md for Sprints repository.
* Installed augur on the live server! A screenshot of a computer

  Description automatically generated with medium confidence
* This included connecting to the following database and creating a schema for it as well as succesfully installing the frontend dependencies/NTLK dictionary.

Text

Description automatically generated

* These are the values that we had in both augur.config.json and frontend.config.json so that the instance is correctly linked.

Text

Description automatically generated

* Set up a privileged user for everyone on the server, configured nginx, successfully ran “npm run serve” to start the development Vue frontend. I switched off most of the workers in augur.config.json to improve server performance. I went pretty closely off the [documentation](https://oss-augur.readthedocs.io/en/main/deployment/nginx-configuration.html) and this [video](https://vimeo.com/477794321). I turned Augur’s backend and Vue off for now because the server was lagging like crazy.

Text

Description automatically generatedText

Description automatically generated

(Left pic: /etc/nginx/sites-enabled/ec2-23-20-84-43.compute-1.amazonaws.com)

Sprint 2:

* More work on the server: configured an elastic IP and a domain name that you can now login with. Updated config files to account for domain name.
* Did some serious debugging/research. “npm run build” and “npm run serve” are compiling successfully but nothing showing up at augur.resumespace.tech or locally.
* Wrote a GitHub gist on how to deploy code on the server and added it to the README.md in our team repository.
* Connected the team Zapier account with Slack
* Successfully got Zapier to get data from the Chaoss API, wrote Hello World Document
* Created Design Document

Sprint 3:

* Fixed the server so now we can access Augur API endpoints (like for example <http://augur.resumespace.tech/api/unstable/repo-groups/10/top-insights>)
* Created a new Zapier trigger to check for Top Insights on repos
  + We ran into some problems when creating the new trigger. One issue is that Zapier does not allow an easy way to pass parameters that replace part of the URL, which is what we need in order to allow the endpoints to work for any given repo\_id or repo\_group\_id. We also had to change the JSON results so that each item returned has a property called “id”
* Updated triggers and API key authentication for our Zapier integration with Augur by updating the domain to our working server augur.resumespace.tech
* Looked into incorporating Augur Visualizations and Zapier Manager
* Added repos/repo groups to our instance to help with testing

Sprint 4:

* Fixed the Top Insights trigger so that parameters can be passed
  + The methods we used to allow for parameter passing and creating an id property are easy to use with other triggers should we make more.
* Created a working Zap that integrates with Slack and will post new top insights when our instance of Augur updates the endpoint (looks for this every 15 minutes.
* Created a presentation to showcase Zapier and how it works with Augur.
* We didn’t have any code to add to Augur so instead of making a pull request we decided to comment our work on this GitHub issue that talks about an Augur Zapier Notification System: <https://github.com/chaoss/augur/issues/1323>

The Progress Report will be some kind indication about whether or not you need to update your scope. For each sprint, include any explanation of obstacles you encountered, reflections, and goals