{Christina}

Hello everyone! I’m Christina from GitGoing, and I’m joined today by Michael, Travis, and Sarah. When last we left off, GitGoing was in-progress with several functionality requirements still needed. We talked about certain features that still needed implementation, such as authentication on log-in and registration, inline commenting, and messaging regarding reviews. We also needed a more robust database API that securely sent and retrieved information. Well today, we’re happy to present that many of these planned features are now live, and available for use.

{so this will address what we talked about last quarter}

//maybe brief images of the code that makes these things work. But will be displayed later so idk

We’ve talked about it before, so we won’t go to in-depth, but our log-in and registration systems are all fully authenticated by Congito from AWS

// Maybe 1/3 paragraph length

Our projects page displays file contents after securely retrieving from a DB

//maybe images of the route for the get(review), and images of the code on react.

Our projects have in-line commenting

//same as above

And our DB api is a robust secure system paired with the cognito-express AUTH library

//images of some of the routes and the auth set-up

And here’s Travis to talk about some of the other neat things we’ve done

-------------

{Travis}

Thanks Christina! So, in addition to all of these awesome goals we’ve achieved, we also took things quite a bit further in-regards to making the site, y’know, usable.

As y’all will be shown in our demo, once a project has been created users can be invited to collaborate on the project through our UI system on each individual project page. After a user logs in, they are redirected to the main site’s homepage, where any notifications they may have received are currently being displayed. These notifications display details about any invites they may have been sent, and give users the option to accept, or decline as they prefer. It’s also important that notifications are informative, so they include details regarding which project or file they’ve been invited to work, as well as the date/time the invite was issued. If a user declines an invite, it is removed from their notification queue. If a user accepts instead, they are then able to view the project, invite other collaborators, delete the project, or make comments on the file. They also have the option of making local changes and uploading a new file! After a user is brought on as a collaborator, or a file has changed in an existing project, you may want to alert others on the project that a review is required, so users can also invite others to review any changes. Users are given a similar notification as before, and if they accept the invite they are automatically redirected to the specific file so they can start reviewing!

//code snippets

Now, all of this would be worthless without access, and we are proud to announce that Git Going is 100% live, available for use in its current Alpha state. Through multiple AWS services, the site is reachable at our specific domain name, and it processes requests as any normal webpage might, though perhaps more securely than some others. The gist of it is that we have a domain name registered with Route 53, and a certificate, issued by AWS, attached to the domain name that provides SSL/TLS (secure sockets layer/transport layer security) transmission. The build of the project sits inside an AWS simple storage service bucket (AKA S3 bucket). A bucket is simply a container for objects stored in S3, and its meta-data. Basically, the bucket “serves” specific files on received requests. Then, a global content delivery network, CloudFront, services requests by catching traffic heading to our specific domain name, converting that request from HTTP to HTTPS if its not already, and pulling the specific files from our S3 bucket. It then sends those files to an edge location, which is just a geographically close-to-requester data centers. The requester receives the files from the edge location, which allows for incredibly fast, and secure requests to our site!

-----------

So that’s the front-end. Now here’s Michael to talk about the back-end.

{Michael}

{EC2 stuff}

And here’s Sarah with a demo of our live site!

{Sarah}

Thank you Michael!

{demo}

//flow of all working functionality

Conclusion: still to come, feelgood, everybody clapped