{Christina}

Hi everyone! I’m Christina from GitGoing, and I’m joined today by Michael, Travis, and Sarah. To recap from last time, the goal of our project is to develop a web-based code review tool integrated with Git. When last we left off, GitGoing was in-progress with several functionality requirements still needed, such as automatic retrieval of diffs when a file is updated, implementation of an approve and reject system for reviews, and a display of file history for projects. We also needed a way to comment on multiple lines of code as well as a notification system when inviting users to projects. Well today, we’re happy to present that many of these planned features are now live, and available for use.

We’ve talked about it before, so we won’t go too in-depth, but here is a quick recap of where we were previously. We had our log-in and registration systems in place, which is fully authenticated by Congito from AWS. These calls are defined in our javascript code through the AWS service Amplify, which is in turn paired with Amazon Cognito. Through these services, we may include other related services, such as the simple email service, and our simple queue service. Additionally, our projects page displays file contents, specific user data, and code diffs after securely retrieving from a DB using Cognito credentials. Here is a quick look at our database design. We’ve also successfully implemented our inline commenting, where a user is able to comment on a specific line of code. Using JavaScript react, a comment is developed as a single component where each component has a separate comment box which contains the line of code, comment list, and username. Once a button is clicked, the comment box will be returned with the proper information needed to post a comment.

Since then, we’ve developed a lot of new features into our site, so here’s a look at our progress and where we are now. Diffs are now automatically retrieved when a file is updated. Using a flask app, we are able to retrieve the diffs between the current file and the proposed updated file when the user uploads a new file. Users are now also able to approve and reject these changes since we’ve stored the diffs as a review status in the MySQL database. This allows users to access review changes, and approve a review which updates the file table. A display of the file history for each project can also be seen. As you can see, you can view the file name, file type, and the date of the modified file. Multiline commenting is also implemented into the site where users can highlight multiple lines of text and insert a comment, which is then stored in the database. Lastly, we have our notification system up and running where users will receive a notification when they are invited to join a project.

In order to achieve these new features on the site, we did a lot of research and testing. For example, I personally did a lot of research into how react components work in order to develop the multiline commenting and I know my teammates also did a lot of research into how to store information into the database, how to develop a flask app, and how to redesign the look and feel of the overall website through CSS. Altogether, we worked hard developing and testing what we have to finally have our new features deployed on the site. Now I’m going to hand it off to Sarah to go further in depth about our progress, achievements and our bumps along the way.

{Sarah}

{Michael}

{Travis}