Software Engineering
CSC648/848
Section 4 | Team 2

Artemis

3/18/2022

Brain Adams	badams2@mail.sfsu.edu	Team Lead
Joseph Kois	jkois@mail.sfsu.edu	GitHub Master
Brandon Cruz-Youll	bcruz5@mail.sfsu.edu	Scrum Master
Sherman Wong	swong42@mail.sfsu.edu	Backend Lead
Arslan Alimov	aalimov@mail.sfsu.edu	Backend Support
Jade Simien	jsimien@sfsu.edu	Frontend Lead

History Table:

4/18/22	Went over M3 (Appendix I and II) feedback from presentation. Designated P1 functions to commit to for the rest of the semester. Began going over Milestone 4 and the tasks needed to fulfill those requirements.
4/19/22	Short stand up for progress on individual team members piece of the project.Discussed Pull Requests in further detail.
4/25/22	Went over M4 tasks again and refreshed everyone memory and what is needed to stay on task. Also, discussed end of project needs and the work involved.
5/2/22	Discussed needs for M4 and finished up what was needed to complete.

1. **Product Summary**

   Artemis is the goddess of the hunt. We have reincarnated Artemis to set her loose on the bugs of software. Artemis makes it easy to track bugs and manage projects with an elegant, efficient design in order to boost productivity for all of your teams. The User Interface is organic and seamless so you can devote your attention to your software production. At Artemis, we value safety and security first and foremost, and we understand how important it is to protect the data of your projects, employees, and customers. All sensitive information is either encrypted or hashed. Artemis respects your privacy and does not unnecessarily track your behavior. Your teams can create and delete workspaces that hold all their projects on different boards of ticket collections. Your teams can view the number of open tickets and closed tickets in the boards, giving you a dashboard for the workspace. Your team members can subscribe to the different workspaces for their projects. When your team views a project board, they can quickly ascertain what tickets are unassigned, assigned and their status—if the tickets are opened, closed and their priority. Managing the tickets could not be easier—simply drag a ticket from the unassigned column to one of your team member's column to assign the ticket. Click on a ticket to view/edit the details or delete a ticket. You can integrate with production to autogenerate tickets into a project board's unassigned tickets.

2. **Q&A Testing**

   I. **Unit Test**

      A. Our chosen framework for testing is Jest.

      B. 53.49% of statements are covered by our unit tests.

   II. **Integration Test**

      A. The link to the completed forms for integration Testing: https://docs.google.com/spreadsheets/d/17H-oyTKQAht21F01UdQ8Fc-HRHbLpdA4/edit?usp=sharing&ouid=113896827507729493729&rtpof=true&sd=true

      B. All P1 features were tested and performed as expected except user profile. This feature is not complete for M4, but an ongoing

feature that will be completed for M5.

3. Code Review

A. Coding Style

We have several systems in place to enforce code style. The first being strict compiler setting for both our front and backend. The TypeScript compiler is run for both the front and backend, which ensures that specific standards are met, this includes no implicit any, no isolated modules and strict null checks. We also use prettier and have a shared config (the prettier.rc in the project root) that will automatically reformat out code. This allows us to easily enforce a consistent format, some of our personal modifications include always inserting semicolons, using ' over ", 4 space tab width, and spaces over tabs. There are however plenty of standard formatting options that cannot be changed with prettier that are enforced as well. We have also banned using class based components in the frontend. Every component is a function, and if any logic needs to be consumed in 3 or more components it will be exported to a hook. We are also using SCSS, and CSS modules for the frontend to help us maintain class naming conventions between components. As an example we use a .wrapper class around wrapping divs throughout the frontend without running into name space collisions due to the hashing provided my CSS modules. This allows us to use a predictable naming scheme which helps those who didn't write the code better understand the intent of any given class.

B. Pull Requests

1. https://github.com/CSC-648-SFSU/csc648-spring22-04-team02/pull/2
2. https://github.com/CSC-648-SFSU/csc648-spring22-04-team02/pull/3
3. https://github.com/CSC-648-SFSU/csc648-spring22-04-team02/pull/4
4. https://github.com/CSC-648-SFSU/csc648-spring22-04-team02/pull/5
5. https://github.com/CSC-648-SFSU/csc648-spring22-04-team02/pull/6
6. https://github.com/CSC-648-SFSU/csc648-spring22-04-team02/pull/7
7. https://github.com/CSC-648-SFSU/csc648-spring22-04-team02/pull/8

8. https://github.com/CSC-648-SFSU/csc648-spring22-04-team02/pull/9
9. https://github.com/CSC-648-SFSU/csc648-spring22-04-team02/pull/10

4. Self-Check: Adherence to original Non-Functional specs

I. Accessibility and Compliance

A. The application is useable for those with vision impairments (correct contrast, predictable interface, readable text etc)

✅ **DONE**

B. The application respects user privacy and does not unnecessarily track users behavior. ✅ **DONE**

II. Security

A. All sensitive user information is either encrypted or hashed

✅ **DONE**

B. Application uses HTTPS ❗ **ISSUE**

We believe there is an issue with a firewall in our vps that is blocking the ability for https usage. We have this set up as something to implement should we hav time to do so before the end of the semester.

C. Use HTTP only and secure cookies ✅ **DONE**

III. Performance

A. Assets are small and lazy loaded ✅ **DONE**

B. Use a bundler that tree shakes, minifies and transpiles code that is compatible with older browsers. ✅ **DONE**

IV. Design

A. Create a UI that users can intuitively navigate ✅ **DONE**

B. Use color to place emphasis on specific items of interest ✅ **DONE**

C. Use readable text ✅ **DONE**

D. Avoid bloating the UI with options that are not relevant to the current view ✅ **DONE**