

Healthy Competition Deployment Plan

1. Deployment Summary

Deployment Date	11/20/2020
Deployment Time	3:00 PM

2. Deployment Risks

Description	Probability High Med Low	Impact High Med Low	Mitigation
Application requires 70MB of disk space	n/a	Low	n/a
Lack of technical training / Having one lead developer slowing process of deployment	35 % med	Medium	Tasks assigned to each team member have specific due dates. Progress is being reported daily via Discord and by weekly meetings in order to spot possible issues before the section of project is due.
Anticipated components being used not being as easy to implement as originally thought	20%	Low	Extensive research being completed before implementation. Small testing between selected components and existing code is performed to ensure time is not wasted on something that will not work long term with desired functionality of the website.if selected tool is found to be compatible small and frequent pushes are done to ensure code works with new changes made.
Time constraints	5%	Low	Expectations were set at the beginning of the project start date to ensure

Budget constraints	35%	Low	<p>each member provided their availability to meet and work on the project section assigned.</p> <p>Anything being used was discussed among team members to ensure monetary resources were allocated appropriately. All purchases had to be approved by all members in order to be implemented.</p>
Lack of time to receive customer feedback after product is launched	20%	Low	<p>Requirements page will be used to ensure client required functionality is met by the deadline of the project.</p>
Error in code after sections for each component had been tested together.	5 %	Low	<p>Code merged to Github is kept small and frequent. Reviews must be approved by all members of the team in order for the new section of code to be merged to the main branch.</p> <p>All members must pull new changes to their local machine in order to have the latest code updated to the repo, this ensures compatibility issues are kept at a minimum.</p> <p>When a team member is ready to push their section to github a separate branch is created temporarily, a review is requested and must be approved/request changes within 4 hours, after this step a message is sent to all team members that a review request had been sent via Discord.</p>

3. Deployment Tools

Software:

- Heroku was used to host the site.
- A GitHub was used to upload all written code and documentation for the site. This was then uploaded to Heroku for hosting services.

4. Deployment Steps

1. Create and sign into Heroku account. This is where the site will be hosted.
2. Connect the Heroku website to the GitHub repository containing the website's code and information.
3. Heroku can automatically detect the language framework our code uses. For our project, this was node.js.
4. Heroku then applies a buildpack to install commonly used packages. This is important to note, because if we were to move the site to a new hosting site, we must ensure that all appropriate packages are installed on that service for npm install to properly run.
5. Select a unique domain for the project. Our domain name is healthy-competition-412.herokuapp.com This is a subdomain of Heroku's domain.
6. Because Heroku is connected to the GitHub project, any changes made and pushed to the repository will automatically deploy via Heroku.
7. After the GitHub repository has been uploaded to Heroku, all necessary packages are installed, and the site has a unique domain name, simply deploy the website by pressing the "Deploy" button.

5. Deployment Communication Plan

Communicate with	How often	How
Name Project Developers (Seth Johnson, Romina Charles, Elizabeth Miller, Hanna Chang, Chris Pastor, and Bradley Kurzawa)	Developers (the whole group) should communicate weekly. This can be done in our regular Wednesday meetings at 6:00 pm.	Developers should discuss any changes that should be made to the website, collect and analyze user feedback, and any new features that should be added to the site. Furthermore, developers should discuss how any new features would be implemented, as well as their benefits and risks. The same analysis should be conducted if any features on the website are to be removed.
Name: Website Testers (Chris Pastor and Seth Johnson)	Whenever any new features or functionality are added to the website. If no new features have been added, testing should still be conducted bi-weekly.	Testers should communicate regularly to ensure any new features added to the site are tested appropriately. They should also regularly test the preexisting components on the site to ensure that they still work as originally designed.
Name: Management-User communication (Bradley Kurzawa)	Whenever users provide Romina with feedback on the website.	Communication between the project manager, Bradley, and the end users should be maintained. The site should allow users to provide feedback on their experience using our website, and comment on things they like or dislike. These comments may then be considered and discussed by the development team to plan features that should be added, removed, or changed on the site.