

The Leftovers

Weekly Update

Nicholas Chiapputo, Khaemon Edwards, Caleb Halter,
Jacob Robbins, Ryan Vanek, Saidat Babatunde,
Ephraim Emilimor, Jordan Simmons

March 27, 2020

1 Remote Setup

As a remote team, we have scheduled meetings through Zoom twice weekly. Our first meeting is after our TA meeting on Mondays at 3:00pm. During this meeting we dissect the information given to use from the TAs and modify our weekly plan accordingly. We have separated into four sub-teams with two members each. Each subteam is given a weekly responsibility for them to work on.

We then meet again on Fridays after the online course has ended to discuss our progress. During this meeting, each team showcases what they have implemented during the week. We also ask questions about each showcase to ensure that each member fully understands each part of the project and that we do not become segmented.

Most of the teams focus on front-end design using HTML, CSS, and JavaScript. One team focuses on the back-end implementation setting up the data and source structure for the server. The server is hosted using a DigitalOcean droplet running a clean install of Ubuntu 18.04.

In addition to scheduled Zoom sessions, the team communicates on a regular basis throughout the week through GroupMe. This communication consists of members asking questions for clarification to ensure that there is one vision and one goal for the project. This prevents us from realizing days later during Zoom sessions that our group has two different ideas of how something should look or work.

2 Implementation Status

Currently, our implementation is in the beginning stages. We have a GitHub repository at [1]. The source files for the server are contained in the top-level directory `src/`. In this directory, the exact structure of the website is kept so that members do not need to directly access the website to change files. Additionally, this allows us to simply create a pull request on the server to push the changes live, instead of having to manually merge, rename, and fix references in each of the source files. The domain for the website itself is located at

[2]. The pages of the actual restaurant implementation will be located with specific paths. The home page will simply have directed links to each location for testing purposes. In a real implementation, the device would use a locked-down browser that would prevent URL address bar tampering. The URL is simply an IP address as otherwise we would have to pay out of pocket. In a real implementation, this would of course be a better looking URL.

The back-end is using Node.js to communicate with the front-end pages. The source for the back-end, along with front-end scripts, is located in the `js/` directory. Currently, there exists only a script for sending the inventory list and a work-in-progress script to edit the inventory list. The inventory list and, in the future, the other data is stored using JSON format in the `data/` directory. This change was made from the original plan of MongoDB as setting up a full database is outside of our experience. Coupling this with the short turnaround time for the project and given the small size of the data needed, we decided it was unnecessary for a full database to be setup as this would cost significant amounts of time and would degrade the quality of the website itself. In a full implementation with experienced workers and more time, MongoDB would likely be implemented in this scenario.

3 Challenges and Minimizing Risk

The biggest challenge is our lack of experience. Most of our team members have little to no experience with front- or back-end development. Because of this, our first week of implementation was spent catching up on the relevant technologies, languages, and ideas. Another major challenge is lack of time. Despite the extra week of Spring Break, many of us were adapting to the changes caused by lockdowns and social distancing. Some of us also had to take time off due to illness. Because of these drawbacks, we began the implementation stage already behind. To remedy this, we are spending a significant amount of time focused on the project. However, we still have other obligations, creating a time-management challenge.

To minimize the risk of these challenges affecting the quality of our work, we are constantly communicating to ensure we are doing things right the first time. This prevents us having to completely rework features, wasting significant time. We are also spending more time on our implementations ensuring that it works instead of trying to rush through and likely causing many errors that we would then need to test, debug, and fix.

4 Status Update

4.1 Accomplishments

In our first week of work, we created our web server and got it online and running. We have begun work on the wireframes for the majority of pages for each of the four different user types (table, server, manager, and kitchen). We have created a standardized header and stylesheet in order to keep the look of the site consistent and clean. We have also settled on a project hierarchy to allow us to easily reuse implementations between our subgroups. This prevents each team from having to rewrite stylesheets, scripts, headers, and footers.

On the backend, Node.js scripts along with Nginx working as a reverse proxy have been implemented. Currently, the scripts are able to be queried by the live website to return the current inventory list in the form of a JSON list. The site is also able to send key-value pairs back to the server to attempt to update the inventory list. The back-end implementation of actually updating the list has not yet been completed, but likely will be by the time of our next update meeting on Monday 3/30.

4.2 Goals

For the next week, our goal is to have the wireframes of each page for each user completed. These pages will have their own URLs (i.e., 64.255.29.130/server.html for the server, etc.). This means that the display will look almost complete. The back-end will have full inventory and menu functionality implemented. It will be able to serve the menu to servers, managers, and the table as well as send the inventory list to the manager. The manager will be able to add, edit, and remove items from both the inventory list and the menu. If this is completed early enough, we also hope to have some functionality in the ordering process. Once the menu is able to be served to the table and servers, those users should be able to add items to an order. It is likely that the actual placement of orders may not be completed yet, however. Finally, we hope to have some work done on requesting help and refills. We do not anticipate this being a difficult task, so it should be done over the next week.

4.3 Interference With Success

The main potential challenge over the next week is the unknown. Due to the remote work, some members may lose internet access and, because of lockdowns, may not be able to go to a separate location with access. We may also have issues with sickness. While it is not likely, it is possible that some members may not be able to work the full week. This type of risk mostly can not be minimized. Instead, we can only do what we can to ensure that the work load is evenly balanced so that some members may be able to help others in a time of need.

Another potential interference is our lack of knowledge with these technologies. While we are optimistic with what we can do and have done so far, there are likely roadblocks that we do not know about yet with languages and technologies we use. For example, we originally did not know the XML HTTP requests had issues with accessing different domains. To minimize the chance of these issues, we will continue to perform significant research before attempting to actually implement a feature. This allows us to see potential issues ahead of time and plan accordingly.

References

- [1] The Leftovers, *GitHub*. [Online].Available: <https://github.com/NickChiapputo/TheLeftovers>
- [2] The Leftovers, *The Leftovers*. [Online].Available: <http://64.225.29.130/>