

The Teaspot

Group: HTTP 418

UX Design: [Ian Dodson](#)

UI Implementation: [Roger Benjume](#)

Backend Implementation: [Alex Moomaw](#)

Backend Implementation: [Kevin Nguyen](#)

Project Manager: [Amy Washington](#)

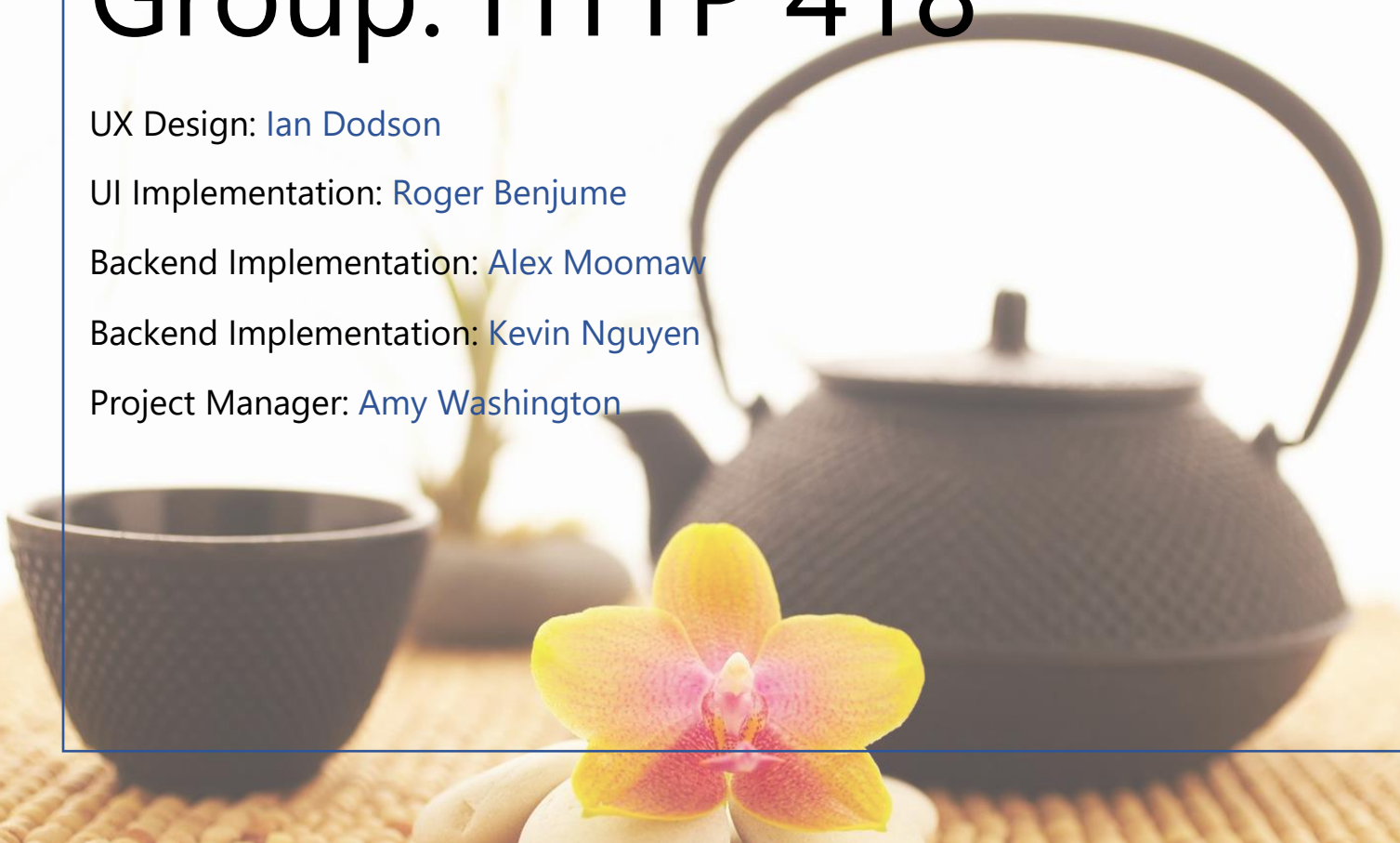


TABLE OF CONTENTS

Executive Summary.....	3
Introduction	4
Project Objectives	5
Project Timeline (Milestones and Deliverables)	6
Milestone #1: Research	6
Milestone #2: Set up GitHub Repository	6
Milestone #3: Initial UX Design	6
Milestone #4: UI Front End Functionality	7
Milestone #5: API Back End Functionality.....	7
Milestone #6: Database Design and Implementation.....	7
Milestone #7: Accessibility Research and Implementation	8
Milestone #8: Implement Dark Mode Design	8
Milestone #9: Testing/Refactoring	9
Project Deliverables	10
Original design mockups.....	10
Finished Product	11
Project Results	13
Objective	13
Outcome	13
Lessons Learned	15
Challenges and Obstacles	15
User Guide	17



EXECUTIVE SUMMARY

We were tasked with working as a group to create a personal blog page that incorporates the languages, techniques, and coding practices taught in this class including but not limited to php, javascript, html, css, and react. With this in mind, we created a personal blog page about teapots, tea, and not much else.

INTRODUCTION

With this project, we aim to satisfy the following requirements:

- Use at least one JavaScript library
 - We planned to use JQuery to implement tooltips on our blog page
- Incorporate a CSS style sheet
 - We planned to use a CSS JQuery style sheet
- Incorporate a database with data, provide a database creation script with instructions for running locally
 - We planned to use php, PHPMyAdmin, and MySQLi extension to bridge the connection between database, backend, and frontend code. Blog comments will be stored in a database, comments will be queried and displayed on their corresponding blog posts upon page loading. Each blog post will be assigned a unique page id to differentiate which comments should populate with each post to.
- Utilize cookies for state management and visits, present a different page/message for returning users vs. new users
 - We planned to use php to check whether the admin user is logged in. If the admin user is logged in there will be a “delete comment” option visible on blog post comments, additionally the login page display will change to show an option for logging out.
- Implement logging (write logs to database for error management and site usage)
 - Logging is designed in such a way that when backend interactions occur, the logger copies the query and stores it in a separate database table
- Utilize Git for version control
 - Link to GitHub repo: [GitHub](#)

PROJECT OBJECTIVES

We will be creating a personal blog page about teapots that incorporates the following features:

- About Me page
- Login/logout page for admin user (admin credentials will be stored in a database)
- Blog post page showcasing previews of different posts
- Filtering options to show blog posts by type and date posted
- Option for visitors to read comments on individual blog posts
- Option for visitors to post comments on individual blog posts
- Option for admin user to be able to delete blog posts
- Option for admin user to be able to delete blog comments
- Option for admin user to be able to create and post new blog entries from the GUI
- Option for visitors to toggle dark/light mode
- Option to sign up for an email list (email addresses will be stored in a database)
- Option to navigate through posts chronologically ("Next Post" and "Previous Post" links at the bottom of each blog post page)
- Custom navigation icons

PROJECT TIMELINE (MILESTONES AND DELIVERABLES)

Milestone #1:

Research

Met/Unmet: Met

Explanation: Conducted independent research followed by team discussion, needs assessment, site outline, and designed framework for implementation

Lessons: It's important to conduct thorough research to help inform the planning/design process.

Milestone #2:

Set up GitHub Repository

Met/Unmet: Met

Explanation: Verified all team members had access to the repository

Lessons: Make sure everyone on the team is prepared and able to use the resource effectively to ensure the individual work stays on track with group work.

Milestone #3:

Initial UX Design

Met/Unmet: Met

Explanation: Utilized Figma to create initial UX design for landing page

Lessons: Pre-planning allows for a seamless UI implementation, waiting for the initial UX design to be completed prevents repetition.

Milestone #4:

UI Front End Functionality

Met/Unmet: Met

Explanation: Implemented design for website

Lessons: It was essential to continually reevaluate the product during the implementation process to ensure that the UI wasn't writing cheques that the code couldn't cash, and to ensure that both frontend and backend code were being written in a complementary way. Frequent meetings and strong communication between departments was essential in ensuring the success of this implementation.

Milestone #5:

API Back End Functionality

Met/Unmet: Met

Explanation: Implemented code to support site functionality

Lessons: Learning how to connect the backend code to the frontend code in an elegant and performant way was an interesting challenge. As mentioned in milestone #4, effective communication was essential to the success of this project, especially since we had two team members working on this aspect of the work. We were able to divide up tasks, collaborate and support each other when needed, and communicate in a way that allowed us to accomplish almost all of our functionality goals.

Milestone #6:

Database Design and Implementation

Met/Unmet: Incomplete

Notes/Explanation: Standing up database for storing admin credentials, storing emails and guest comments, storing logging info, and storing guest

comments were successful. Storing blog post content in the database for rendering pages was not met due to time constraints.

Our initial hope was that the admin user would be able to create new blog posts from within the UI, however this would have required a more advanced skillset and a longer timeline. We made the decision to pivot and create our individual blog pages in the code knowing that this solution would not be scalable for an actual blog site.

Lessons: Using the database to store blog post content was a much larger and more involved undertaking than we initially expected. If we were creating this for production, we would have taken the time to follow through with this design to make the blog more scalable.

Milestone #7:

Accessibility Research and Implementation

Met/Unmet: Met

Notes/Explanation: Researched WCAG guidelines for web accessibility

Presented findings to class, plan is to incorporate WCAG guidelines into work moving forward (added tooltips and made sure all media had appropriate alt tags)

Lessons: We did not actively consider accessibility with our initial design so this research phase was eye opening. We learned about simple changes we could implement to make our site more accessible to visitors. There is such a wide range of accessibility needs that we would not be able to address them all within the scope of this project, however moving forward we will take accessibility into consideration in the design and implementation of this project as well as future projects.

Milestone #8:

Implement Dark Mode Design

Met/Unmet: Incomplete

Notes/Explanation: The plan was to include an option that would allow the user to toggle between a light and dark color scheme

We made the decision to focus on functional design aspects with the understanding that aesthetic changes such as implementing a dark mode feature may not be achievable within our production timeline.

Lessons: We were unfortunately unable to complete our plans for a dark mode design within the timeframe. This was unfortunately a lesson in over promising, under delivering. It might have been better to complete a more simple, pared down implementation of our design and then add on additional features at the end of the project.

Milestone #9:

Testing/Refactoring

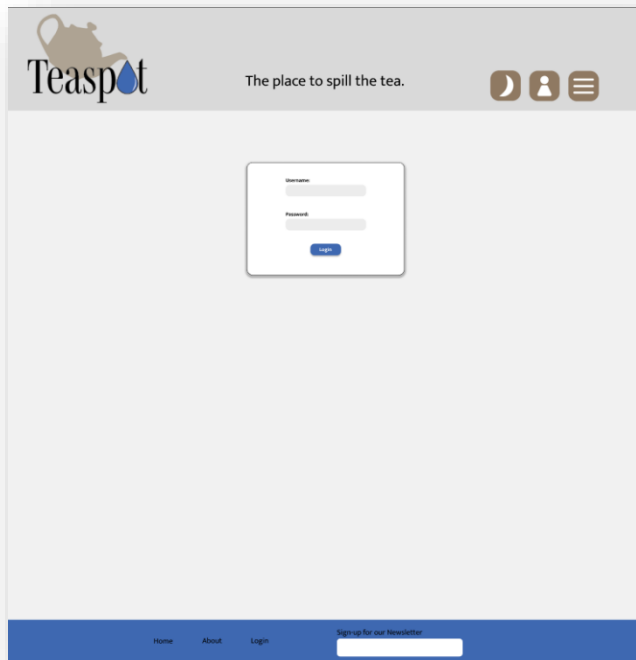
Met/Unmet: Met

Explanation: Once the individual components of the site were complete, we linked them all together and tested the function of the site to look for areas of improvement. We felt it was more useful to spend the last week of the design process conducting testing to ensure our deliverables were working as anticipated.

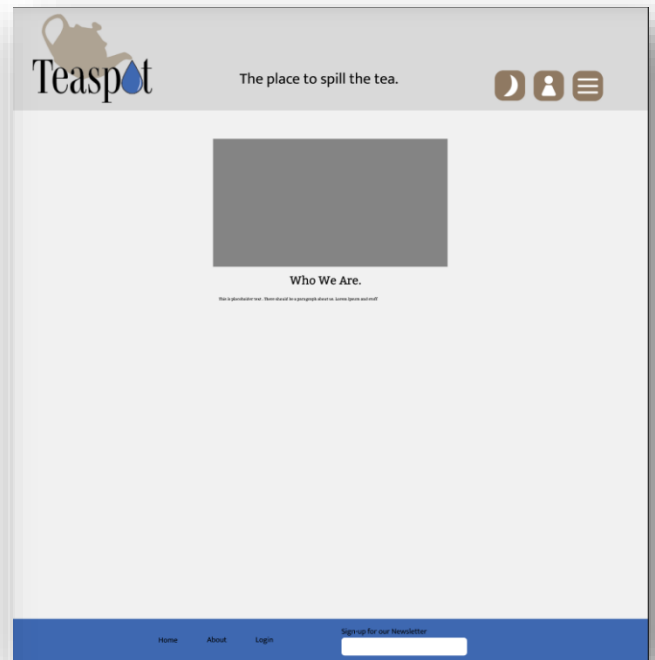
Lessons: We felt testing and refactoring was an important part of the process because it is better to deliver a product that works but is slightly less flashy than a big fancy product that doesn't function properly.

PROJECT DELIVERABLES

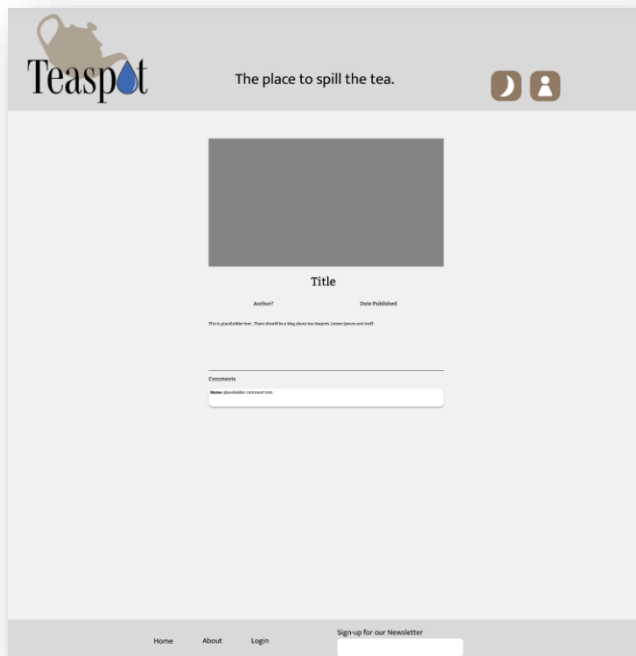
Original design mockups



Login



About

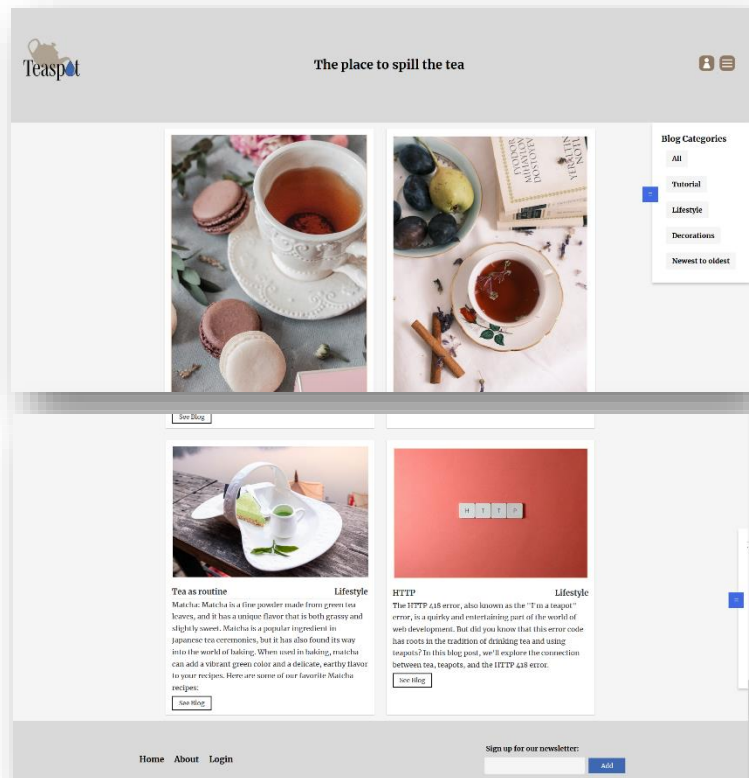


Posts

Finished Product

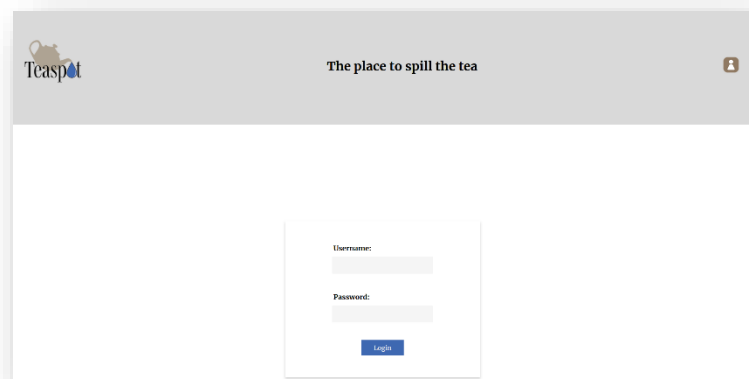
Landing page:

- Header with custom logo that persists on all pages
 - Teapot logo links to Landing Page
 - Custom icon links to Login page
 - Custom logout icon when admin user is logged in
- Drawer that slides out with filtering/sorting options to show blog posts by category
- Previews for blog posts
- Custom footer with navigation links to Home page, About page, and Login Page
- Input field for email list
 - Links to database



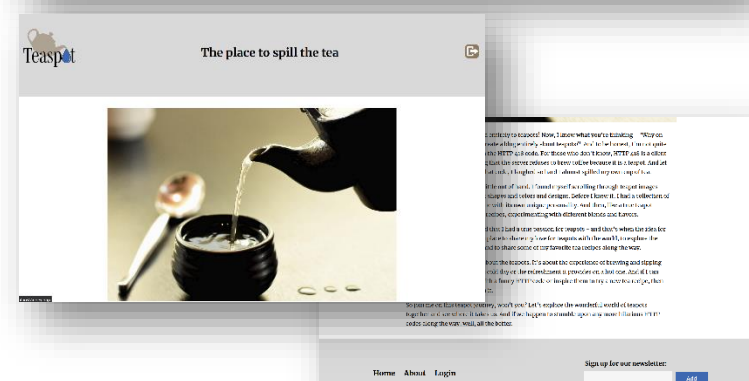
Login Page:

- Login credentials are stored in a database schema
- Incorrect login/password will return an error message



About Page:

- Logout icon when admin user is logged in
- Nav footer with email list field
- Teaspot logo navigates to home



Blog Posts:

- Comments are specific to each blog post
- Admin user will have the option to delete blog posts (will present as an icon next to each comment)
- Visitors can add comments without logging in
- Footer navigation routes back to Home, About, or Login page



Tea as routine

3/15/2023

Baking with tea can be a fun and unique way to add flavor to your baked goods. Matcha and Earl Grey are two of the most popular teas to use in baking, and they can add a distinct and delicious flavor to your recipes. In this blog post, we'll explore the world of baking with tea, and we'll provide you with some delicious recipes for Matcha and Earl Grey baked goods.

Matcha: Matcha is a fine powder made from green tea leaves, and it has a unique flavor that is both grassy and slightly sweet. Matcha is a popular ingredient in Japanese tea ceremonies, but it has also found its way into the world of baking. When used in baking, matcha can add a vibrant green color and a delicate, earthy flavor to your recipes. Here are some of our favorite Matcha recipes:

Matcha Cupcakes: These cupcakes are made with a fluffy and tender cake that is infused with matcha powder. Top them with a fluffy matcha frosting for a perfectly sweet and refreshing treat.

Matcha Cookies: These chewy and crunchy cookies are a great way to enjoy matcha in a more portable form. They're easy to make and they're perfect for on-the-go snacking.

Matcha Donuts: These donuts are made with a soft and tender dough that is infused with matcha powder. They're perfect for a special breakfast treat or for a sweet and indulgent dessert.

Comments

No comments yet

Username:

Comments:

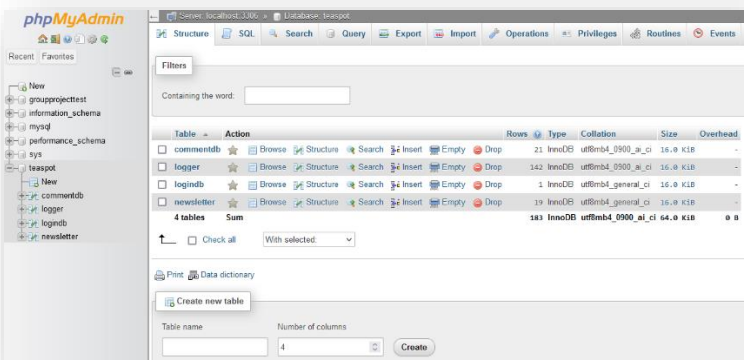
Add

[Home](#) [About](#) [Login](#)

Sign up for our newsletter:

Database:

- Table for storing blog comments
- Table for storing logging events
- Table for storing login credentials
- Table for storing email addresses for mailing list



PROJECT RESULTS

OBJECTIVE	OUTCOME
About Me page	Completed
Login/logout page for admin user (admin credentials will be stored in a database)	Completed – shows an error message for incorrect login credentials, correct credentials are stored in the database. Once logged in, the admin has the ability to delete blog post comments
Blog post page showcasing previews of different posts	Completed - this is the landing page for our blog site
Filtering options to show blog posts by type and date posted	Completed – this can be accessed through the “hamburger” icon on the right side of the landing page, and from the similar icon at the top right of the page.
Option for visitors to read comments on individual blog posts	Completed – each blog post has a section for comments at the bottom of the page
Option for visitors to post comments on individual blog posts	Completed – there is a text input field below the comments section on each blog post for site visitors to post additional comments. These comments are stored in the database and displayed comments are posted
Option for admin user to be able to delete blog posts	Not completed – We made the decision to build out each sample page in the code due to complexity and time limitations
Option for admin user to be able to delete blog comments	Completed – comments can be deleted by the admin user. A “delete” icon is displayed next to each individual comment when the admin user is logged in, deleted comments are removed from the database.
Option for admin user to be able to create and post new blog entries from the GUI	Not completed – We weren’t able to complete this goal, blog posts have to be added through the source code

Option for visitors to toggle dark/light mode	Not completed – We made the decision to focus on functional implementations and the dark mode option had to be shelved due to time constraints
Option to sign up for an email list (email addresses will be stored in a database)	Completed – there is a text field at the bottom of the landing page that accepts user emails as inputs. *Note* in future iterations we would add regex to validate email addresses
Option to navigate through posts chronologically (“Next Post” and “Previous Post” links at the bottom of each blog post page)	Not completed
Custom navigation icons	Completed – Site Logo, Login/Logout Icon, Delete Icon were all unique designs created by our UX team.

LESSONS LEARNED

- Good communication between team members is essential for a successful outcome
- Incorporate accessibility into the initial design plan to prevent tedious refactoring
- Team members should be encouraged to choose roles that play to their strengths/aptitudes
- It's important to have a plan in place before a single line of code is written
- The first iteration of a product should be simple, fancy features can be added once the minimum viable product has been achieved

CHALLENGES AND OBSTACLES

- Access to resources (trying to complete design aspects that hadn't been covered by the curriculum yet)
- Not having a solid design plan/schedule in place at the beginning of the project
- In the beginning of the project, we didn't have a lot of communication as a group. About halfway through we started placing more of an emphasis on collaboration and found the project moved much more easily.



CONCLUSION

Designing a functional website is a multi-faceted process that requires careful planning and consideration to be executed properly. Given the opportunity to do this project again from the beginning, we would have established a clearer workflow so that each department understood what resources their counterparts needed/would be providing. This would have simplified the design process. Our approach was more akin to building each section individually and then cobbling them together into one complete product, it might have been more effective to build out smaller sections of the website in stages with more coordination between our "departments".

USER GUIDE

Steps for running The Teaspot in Laragon's local environment:

1. Install Laragon
2. Clone our [github](#) repo
3. Set Laragon root folder to our project folder
4. Download [phpMyAdmin](#)
5. Extract phpMyAdmin into Laragon/etc/apps
6. Change the name of "phpMyAdmin-5.2.1-all-languages" to "phpMyAdmin"
7. Launch Laragon
8. Set root folder to our project's root folder
9. Open database,
 - Username = root
 - no password
10. Create database "teaspot"
11. Select database "teaspot"
12. Select the import option and import teaspot.sql from our project: DB/teapot.sql
13. Click 'Web' button in Laragon to launch
14. All php files will need to be opened in phpstorm or Laragon via right click web->www->folder.
15. Keep in mind that your server will have to be started in order to view our pages.

Note:

All php files will need to be opened in phpstorm or Laragon via right click web->www->folder. Keep in mind that your server will have to be started to view our pages.

Note:

Login credentials for the admin user are

- Username = admin
- Password = admin