MMMAC Daddies - Membership Site

Contributors:

- Michael Moran: Project Manager
- Mikayla Stewart: UX Designer
- Malcolm Shepherd: Backend Developer
- Alysha McCullough: Frontend Developer
- Cameron Norton: Backend Developer

Executive Summary:

The website was made to represent a pretend organization with the purpose of homing left behind items with “mac” in their name. The website accomplishes the purpose of the organization by allowing for new members to join, allowing members to take these items, and informing them of the mission on the site.

Introduction:

The MMMAC Daddies website, as mentioned previously, was made to represent a pretend organization with the purpose of homing left behind items with “mac” in their name. The website was designed to be able to allow new members to join with a login feature, implement databases for the users as well as the “mac” items, and style it in a way that it is accessible to all users.

Initial Goals/Objectives:

Goals and objectives include meeting the final project specifications as listed below, and creating a usable site that is easily navigable and readable.

Objectives:

- The website will have an index page that welcomes returning users and includes a slideshow of sympathy bringing images. This index page will also share the mission of the organizations.
- A login page that will check the database for returning users to be able to login and stay logged in.
- A mock shop that includes basic merchandise.
- A join page that will allow users to create an account that is then stored in the database and will only function when the user inputs proper information.
- A mock donation page for the organization.
- A mock contacts page for the organization that includes the members of the group.
- An organization events page.
- An items page that allows logged in users to claim items and have those items in the database be associated with their user ID.
- An admin feature to add new items to the item page and the database.
- A page that introduces the main members of the group.

- A video on our homepage that represents our organizations.

Milestones Met/Timeline:

January: Get Repo set up in a way that is organized and that progress can be made.
February 1st-14th: Get basic site framework set up as well as basic docker and database.
February 20th: Create subpages and set up links to be able to navigate to them from any page.
February 21st: Start working on footer.
February 22nd: Work on the html of the subpages and the css of the index page.
February 23rd: Start getting the base css code down on subpages. Put things into the shop. Finish footer.
February 24th-27th: Fixing up the navigation bar. Added jQuery library.
February 28th: First database communication! New subpage (items) was created and the nav bar updated accordingly.
February 15th-28th: Create new subpages and styles them and existing ones. This includes things like navigation bar, color themes, and properly ordered pages. Made database scripts and eventually got them to communicate with the site.
March 1st-2nd: Replaced current Node application with basic ExpressJS setup. Finished up Login css and added a forgot password page. Login page now communicates with the database. Texted is replaced with icons for social media on the footer for all pages.
March 6th: Password and database related fixes. Added Node bcrypt library for more secure password hashing. Rewrite login.js to verify password input with bcrypt. Login.js now queries the database to verify user information.
March 7th-8th: Worked on cookies, banners for pages and made the contact page better.
March 1st-9th: Improved existing pages where necessary. Implemented Node for database. Added login and join verification.
March 11th-12th: Added and worked on logging table to database as well as implementation involving it.
March 13th: Worked on events and login page.Added events to the database and associated fetching and jQuery implementations to the site. Updated admin/login features.
March 14-15th: Finishing touches and checking over everything.

Project Deliverables:

Github Repo: contains all site content.
Docker image
Database.docx
databaseDiagram.png

Project Results

Project Specifications met:

1. You must use a javascript library (jquery is a great choice) (10%)

Jquery was added to the scripts folder and implemented mostly on the index.html page. Used jquery cycle2 plugin to create a slideshow of our pitiable mac images in black and white. Jquery was also used for text animations on the index.html banner.

2. You must use a stylesheet to style your site (10%)

Each of the main pages has its own styles.css file stored in the styles folder.

3. You must have a database with data, you will need to provide a creation script with instructions so I can run your final project on my local machine (10%)

Using MySql for the database and Node to communicate to that database. Creation script in root directory.

4. You must provide a login functionality with admin features (10%)

Users will have a join feature and a login feature. Admin features will simply include showing more info on donate, item, and event pages.

5. You will need to use cookies to manage state and visits and present a different page/message for returning users vs. new (10%)

Index page will change its main message to a welcome back message if cookie status has a logged in user. Login button will also change to a logout button. The join now button will change to a donate button on the main page. Users will not be able to adopt items unless they are logged in.

6. You must implement logging and write logs to your database this will be used for error management and site usage and anything else you deem necessary (10%)

A logging table in the database stores simple timestamp-message pairs. Log entries are written upon item adoption, item update, user registration, login, or log out.

7. You are required to use git and provide your repo info so I can view check in's etc (10%)

Github info provided.

8. You will need to provide a writeup at the end with the following information (20%)

This is the final writeup.

User Website Guide:

- Website setup guide on GitHub page.
- Admin may login using the username and password “root”

- New users will be greeted with the index.html homepage that includes our mission statement, and a big join button to present the user with the option to make an account (all links can also be accessed with the navigation bar). Upon account creation, the user may now be able to log in to the new created account with the login button on the navigation bar. The index page will include a jquery slideshow of sad mac images that will gain sympathy from the user.
- The user can access any desired page from the navigation bar at the top.
- The logo on the navigation bar will take you back to the home page.
- The shop page will display a mock shop for member merchandise.
- The login page should now become a logout button on the nav bar.
- The join page link will then change to the logged in username.
- The Donate page will lead to a page for information on where donations to the organization may be sent.
- The Contacts page will include basic organization email information, as well as drop down info boxes that will have the emails of each of the group members.
- The items page will have a short list of the currently available mac items that have the option to be claimed. Claiming an item will require the user to be logged in. Claiming an item is as simply as clicking the radio button option on the desired item and its respective submit button. The ‘MMMeet the Team’ link will simply take the user to a list of pictures of each of the group members.
  - Admins can update the stock of the items.
- At all times a bottom bar will be available with another logo that takes you to the home page and some links to social media pages.
- You can access the database at **localhost:8081**
  - The creating script and all of our database work is in the sitedb

Post Mortem

Project Results

Lessons Learned:

- Since there is a style sheet for each page, it is easiest to have a section of the site that exists on all pages (like the header and footer) to be exactly the way you want it before copying it to all the other pages rather than making small changes overtime and having to remember to change it on the other pages.
- It is easier to find things/navigate the style sheet when it is organized in a way that is similar to the page layout.
- It is better to start sooner rather than later.
- Should have kept track of all image sources from the beginning.
- Using something such as React may have made the project much easier.
- Reducing the number of Docker containers or layers of communication if possible in the future would probably make this easier.
- Laying out what a page should look like beforehand will make styling it much easier.

Challenges/Obstacles and Goals Not Met:

- An admin feature that lets an admin add items to the items page and the database seemed to be too difficult for the time we had left so we instead opted for an easier to implement admin feature.
- We instead implemented an admin feature to change the number in stock of items available.
- Making the page layout work on different sized computer screens.
- The header/nav bar. All aspects of this was a challenge. Getting some of them to float right was a battle, making the organization name a part of the logo was done because it fixed a bad formatting issue, making it extend to the edge of the page (making it not extend past the edge of the page), changing the text color of the link if we were currently on that page. All of it took way more work then it should have. We were originally planning on making this a floating header that scrolled with the user, sitting on the top of the page. However, it was too much work for not that big of a payoff, especially considering most of our pages aren’t very long.
- The footer. Some similar problems as the header. It was easier because we learned from the mistakes made in the header.
- Video was deemed not worth creating based on the time we have.
- We were unable to use Laragon due to it only being available on Windows machines.
- The actual adoption process is now done through logging, so when a user adopts an item it will instead be seen and processed through the logs as an order essentially.
- Having more interaction between users and shop, event, and item pages would have been cool, but we don't have the time to flesh those out.

What We Would Do Differently:

- Start sooner
- Have page layout in mind before creating the page for all pages and not just the homepage.
- More neatly arranged and overall cleaner code, especially since we are working in a group.
- Consider using something like React to make everything easier.
    - Use JQuery selectors over document.getElementById() in most places.
- Consider setting up a custom Dockerfile with both Apache and Node available so that we don’t have to do the weird back-and-forth message passing.
    - Or do everything from ExpressJS.
- Write more documentation along with the project rather than towards the end.

Conclusion:

Overall, the project was somewhat successful in achieving most of the main goals we set out to do. Due to time constraints we had to leave out a couple of extra features we had planned in order to prioritize meeting the main final specifications. It was a little evident that we started a bit later than we should have and getting started right away was a big lesson learned. Also with working in a group setting, neatly organized code, well written commit messages, writing documentation early, and good group communication are shown to be very important. Thankfully none of these things were a big issue in the group and communication went pretty well for the most part. Some of our features, such as the item adoption, were still implemented, but just not how we initially intended; however we still implemented these features in a functional manner. Keeping these in mind, I think we could have found more success in better implementing site features with more time/starting earlier and by laying out more of a framework of the plan before creating it. We set out to make a site where users may join a mock organization that allowed them to claim abandoned items. Our site allowed users to do that in, perhaps not the most efficient way, but a sufficient way.

Appendices:

Database.docx included in the root of the repo.
jQuery https://code.jquery.com/jquery-3.6.4.min.js
databaseDiagram.png
Documentation folder includes research from sources listed below.

Acknowledgements:

Thank you Scott Amack for answering any questions we had regarding the assignment. Thank you Mac Daddy’s for name inspiration.

Sources:

- MMMAC logo: Shutterstock.com
- Twitter Icon: Twitter - Free social icons (flaticon.com)
- Facebook Icon: Facebook, social media, fb, social icon - Free download (iconfinder.com)
- Instagram Icon: Instagram Logo Fill Vector SVG Icon - SVG Repo
- Mac n Cheese Image on Contacts page: https://realfood.tesco.com/recipes/miso-mac-n-cheese.html
- Join page inspiration: Sierra Club
- Banner Inspiration / Event Page: World Wildlife
- Collapsible content in contacts page: https://www.w3schools.com/
- jQuery https://code.jquery.com/jquery-3.6.4.min.js
- Images Used
  - Macadamia Nuts
  - Macaw
  - Macintosh apples
  - Mac Cosmetics
  - iMac
  - Mackerel
  - Macarons
  - Mac and Cheese