SW Engineering CSC648/848 FALL 2020

Let's Donate

Team 3

Milestone 4 v1

December 10, 2020

Team Lead: Himanshu Garg (hgarg@mail.sfsu.edu)

Front-End Lead: Milo Abril (mabril0526@gmail.com)

Back-End Lead: Marlon Bustamante (mbustama@mail.sfsu.edu)

Github Master: Suman Basaula (Sbasaula@mail.sfsu.edu)

Database Manager: Dipendra Dhoj Rana (drana1@mail.sfsu.edu)

Full Stack Developer: Kunyu Ruan (kruan1@mail.sfsu.edu)

Date	Version
Dec 10, 2020	Milestone 4, v1
Nov 30, 2020	Milestone 3, v2
Nov 19, 2020	Milestone 3, v1
Nov 12, 2020	Milestone 2 v2
Oct 29, 2020	Milestone 2 v1
Oct 8, 2020	Milestone 1 v2
Oct 1, 2020	Milestone 1 v1

Table of Contents

1. Product Summary

San Francisco is one of the most expensive cities to live in throughout the world. The housing and furniture prices are high and can be almost non-affordable for most. In addition to that, there are many other expenses like hospital bills or insurance and there are almost no options available to the public that can help with this pricey lifestyle. Hence there is definitely a need for a support service where any individual can get at least some of the essentials like furniture and financial support from people with some privileges.

Let's Donate comes to the rescue to provide support to the people in need of daily use products for free along with some financial support to get help during difficult times. We at Let's Donate will build a community that revolves around helping people in need.

While Let's Donate strives to be completely free compared to the competition, to guarantee the longevity of the product, there will be a number of ways that we can obtain income. To ensure that our product is not taken advantage of, users will have a monthly limit of claiming five instances of support whether it be a physical item or monetary donation. If a user would like to be able to accept more support in a given month, there will be a fee that will lift that limit for the month. Another way to ensure the financial support for the product would be through monetary donations that users can give directly to Let's Donate. When a user provides financial support to someone in need, they will also have the option to donate to Let's Donate. In addition to a pay-off to receive more products and accepting donations, with enough traction, Let's Donate may also use advertising from other businesses to gain revenue. Doing so can guarantee that Let's Donate can continue to operate.

Uniqueness:

Let's Donate is the only app promoting fundraisers and donation of products like table, chair, etc under the same roof with the service of direct communication facility with the donor. Apps like Facebook, Craigslist are filled with lots of overpriced used products. Our goal is to eliminate the wastage of products which are in good condition and otherwise would have been thrown away and we also focus on raising money for people in need.

Product Features:

Guests:

1. Guests shall be able to register themselves in the website
2. Guests shall be able to create their own accounts
3. Guests shall be able to become users
4. Guests shall be able to browse donations and fundraisers
5. Guests shall be able to view donations page
6. Guests shall be able to view fundraisers page
7. Guests shall be able to view user profiles
8. Guests shall be able to view other user posting history

Users:

1. Users shall be able to login to the website
2. Users shall have a username
3. Users shall have a password.
4. Users shall be given an editable profile
5. Users shall be able to edit their user page
6. Users shall be able to set up their profile picture
7. Users shall be able to see other people's current posts
8. Users shall be able to visit their profile
9. Users shall be able to view their current posts

10. Users shall be able to reset their password
11. Users shall be able to shop
12. Users shall be able to search browse items
13. Users shall be able to direct message other users
14. Users shall able to search items by name
15. Users shall able to endorse fundraisers
16. Users shall be able to make payments
17. Users shall be able to place items up for donation.
18. Users shall be able to set up description of items for donation
19. Users shall be able to remove their own product posts
20. Users shall be able to remove their own fundraiser posts
21. Users shall be able to delete their own profile
22. Users shall be able to log out of the website
23. Users shall be able to see terms and condition of app while signing up
24. User shall sign up only after accepting terms and conditions

Website:

1. The website shall be regularly updated regarding items.
2. The Website shall update whenever a new post is made.
3. The website shall be regularly updated regarding history.

Administrator:

1. Administrators shall have the ability to ban users from using features of the website
2. Administrators shall be able to delete the spam items.
3. Administrators shall be able to remove posted items.
4. Administrators shall be able to remove posted fundraisers.

Product Url:

https://letsdonate.xyz/

2. Usability Test Plan

Test Objectives:

1. Guests shall be able to browse the products page
   What? - The effectiveness of these pages: we want to test if the user is able to navigate to these pages successfully (including through search) and gauge how easy it was for them to get there and how much time it took them as well. And when they *are* on those pages, how satisfied are they with the layout and the method of getting there?
   Why? - The browsing pages are essential to our website in that they are how users are able to find products to ask for or fundraisers to help.
2. Guests shall be able to register an account from any page they are viewing
   What? - The effectiveness of these pages: we want to test if the user is able to sign into an account easily without any issues. regardless of where they are on the website.
   Why? - An account is necessary to access features of LetsDonate that will allow a user to interact with the community within it.
3. Users shall be able to place items up for donation.
   What? - This is our way of testing a user's ability to post a donation on the website for other users to view. We should test how effective the navigation through this process is along with how satisfied a user is with the process in general.
   Why? - This is another vital functionality to our product since it is the first half of users' ability to directly donate products to others.
4. Users shall be able to remove their own posts
   What? - Here, we are testing the effectiveness of the process of deleting a post.
   Why? - As a user, a product posted for donation is still owned by the user until someone else receives it from them and in some cases, the user may change their mind about the item and want to take it down as an offer for donation.
5. Users shall be able to message each other through website's messaging system
   What?- We need to test how effective the messaging system works for users. They should be able to notice whenever they get messages from other users and what those messages are for and message each other without any issues.
   Why? - This system is supposed to be vital to how users are able to give donations as it's how users are able to communicate their methods of exchanging items.

Test Description:

In terms of system setup, the server side is deployed on Amazon Web Service running on a Ubuntu server and its database is MySQL while the client side is using Windows 10, OSX, or a Linux system running on chromium with a browser of Google Chrome v86.0, Firefox v86.0, or Microsoft Edge v85.0.

The starting point is the home/landing page (the URL of which is located at the end of this section). From there, the user can use the search bar or the two "featured" buttons to make their way to the browsing pages or sign up/log in and view their

dashboard to access posts. On all pages, the user may be able to read messages so long as they are signed in.

For the purposes of simply browsing posts, the intended users are the general public, but for those that wish to receive or give donations through our platform, we are aiming to pull in users that are looking to give to those that need it or users that are seeking help, both in the forms of materialistic or monetary donations.

The URL of the system to be tested is http://letsdonate.xyz/. We will be looking to measure the amount of time testers take in performing tasks and how satisfied they are with their experience with them. With the current COVID circumstances, the measurements will be taken through a video call through a shared screen of the tester's process in performing a task. In addition to the questionnaire, the tester may also ask simple questions such as "What did [the tester] like most about the feature?"

Usability Task Description:

Task	**Description**
task	navigate to product browsing page
machine state	home/landing page of *Let'sDonate* http://letsdonate.xyz/ user is not registered/logged in
successful completion criteria	the tester's browser has reached the /products page
benchmark	<= 1 minute

Task	**Description**
task	create/register/sign up a new user account
machine state	home/landing page of *Let'sDonate* http://letsdonate.xyz/ user is not registered/logged in
successful completion criteria	the tester is able to reach the registration form and input information
benchmark	<= 1 minute

Task	**Description**
task	create a post for a donation

machine state	user's dashboard page https://letsdonate.xyz/User user is logged in
successful completion criteria	a new post is created on the /products page
benchmark	<= 5 minutes

Task	**Description**
task	delete an existing post that was created by the user
machine state	product page created in previous task https://letsdonate.xyz/ProductPost user is logged in
successful completion criteria	existing post is removed from /products page
benchmark	<= 5 minutes

Task	**Description**
task	send and receive a direct message with another user
machine state	home/landing page of *Let'sDonate* http://letsdonate.xyz/ user is logged in
successful completion criteria	a message is successfully sent to another user, the current user receives a message from the other user, and they notice a notification is given when that happens.
benchmark	<= 5 minutes

Usability Test Table:

Task	**Completed**	**Errors**	**Comments**
navigate to product browsing page	5/5	none	They thought the products page was intended for a different use than it was made

			for. Image in the back is too distracting.
create a new user account	5/5	the form does not follow UI mockup	users did not actually create accounts
create a post for a donation	2/5	users button	users button on navbar wasn't noticeable or didn't click on first attempt
delete an existing post that was created by the user	0/5	not implemented	not implemented
send and receive a direct message with another user	5/5	none	users did not actually send messages

Questionnaire:

	strongly agree	**agree**	**neither agree or disagree**	**disagree**	**strongly agree**
I am satisfied with how easy it is to use this system.		4	1		
The organization of information on the website's pages was clear.		4	1		
I can see myself using this website frequently.	1	4			

Comments: I thought it was a really interesting website. Better than craigslist for sure.

3. QA Test Plan

Test Objectives:

1. Guests shall be able to browse all existing items.
    a. In this non-functional requirement, what is being tested is that users shall be able to browse all items on the website. This means that users shall be able to view all products that are being donated and be able to view all fundraisers that users wish to receive monetary donations from.
2. Each post shall display an image of the item.
    a. What is being tested in this requirement is that when a user clicks on a product that is being donated, the user will be able to see an image of the actual item being donated.
3. The website shall not be overcrowded with ugly buttons and links.
    a. With this requirement, the ease of use and easy navigation is to be tested. With only the buttons and links necessary for a post, users shall be able to have an easier experience.
4. Website shall scale properly for mobile devices
    a. Every user will have a different method for viewing our website. This requirement will be tested to ensure users will have a user interface that will scale properly no matter if the user is using a laptop, desktop, or mobile device.
5. The font shall not be too small.
    a. What is to be tested with this requirement is that the text on the website shall be easily read. Users should not have a difficult time reading text on the screen.

Hardware and Software Setup:

Hardware:

Computer:	Custom PC with Ryzen 5 1600 CPU, 16GB DDR4 RAM, Radeon HD 7950 GPU, 24” 1080p IPS monitor
Mobile Device:	Motorola Z3 Play
Operating System:	Windows 10 Pro x64, Android 9

Software:

Desktop Browser:	Google Chrome Version 87.0.4280.66 (Official Build) (64-bit) Mozilla FireFox Version 83.0
Mobile Browser:	Google Chrome Version 87.0.4280.66

The website located at https://letsdonate.xyz/ will be tested using these hardware and software specifications. The testing shall be conducted as a guest meaning that logging in is not required to perform these tests.

Feature to be Tested:

1. Guests shall be able to browse all existing items.
2. Each post shall display an image of the item.
3. The website shall not be overcrowded with ugly buttons and links.
4. Website shall scale properly for mobile devices
5. The font shall not be too small.

QA Test Plan:

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results
1	Browse all existing items (Chrome)	Guests should be able to view all donated products and all fundraisers	Visit homepage and click either “View All Products” or “View All Fundraisers”	User is brought to a page with all donated products or all fundraisers	Pass
2	Browse all existing items (FireFox)	Guests should be able to view all donated products and all fundraisers	Visit homepage and click either “View All Products” or “View All Fundraisers”	User is brought to a page with all donated products or all fundraisers	Pass
3	Browse all existing items (Mobile)	Guests should be able to view all donated products and all fundraisers	Visit homepage and click either “View All Products” or “View All Fundraisers”	User is brought to a page with all donated products or all fundraisers	Pass
4	Display image of item (Chrome)	Guests should be shown an image of product that is being donated	Visit the homepage and click “View All Products”. Then select a product to view product post	Product displays the image of the product that is used for the thumbnail	Pass
5	Display image of item (FireFox)	Guests should be shown an image of product that is being donated	Visit the homepage and click “View All Products”. Then select a product to view product post	Product displays the image of the product that is used for the thumbnail	Pass
6	Display image of item (Mobile)	Guests should be shown an image of product that is	Visit the homepage and click “View All	Product displays the image of the product that is	Fail

		being donated	Products". Then select a product to view product post	used for the thumbnail	
7	Use of buttons and links (Chrome)	When navigating between pages, there should be only relevant buttons and links present	Visit the homepage and click "View All Fundraisers". Click on the post labeled "Recently turned Homeless".	Fundraiser post only has relevant information like a button to donate and send a message, image about post, and post description	Pass
8	Use of buttons and links (FireFox)	When navigating between pages, there should be only relevant buttons and links present	Visit the homepage and click "View All Fundraisers". Click on the post labeled "Recently turned Homeless".	Fundraiser post only has relevant information like a button to donate and send a message, image about post, and post description	Pass
9	Use of buttons and links (Mobile)	When navigating between pages, there should be only relevant buttons and links present	Visit the homepage and click "View All Fundraisers". Click on the post labeled "Recently turned Homeless".	Fundraiser post only has relevant information like a button to donate and send a message, image about post, and post description	Fail
10	User interface Scalability (Chrome)	Visiting the website should provide a usable and pleasant experience no matter the window size	Visit the homepage and change the window size of the browser.	Content on website will resize to fit the window size	Fail
11	User interface Scalability (FireFox)	Visiting the website should provide a usable and pleasant experience no matter the window size	Visit the homepage and change the window size of the browser.	Content on website will resize to fit the window size	Fail
12	User interface Scalability	Visiting the website should provide a usable	Visit the homepage and change the	Content on website will resize to fit the	Fail

	(Mobile)	and pleasant experience no matter the window size	window size of the browser.	window size	
13	Font Size (Chrome)	Font and text size should be a reasonable size for ease of reading	Visit the homepage and read the current texts. Visit a product or fundraiser page and read the text.	Font and text size are easily readable and are a good size	Fail
14	Font Size (FireFox)	Font and text size should be a reasonable size for ease of reading	Visit the homepage and read the current texts. Visit a product or fundraiser page and read the text.	Font and text size are easily readable and are a good size	Fail
15	Font Size (Mobile)	Font and text size should be a reasonable size for ease of reading	Visit the homepage and read the current texts. Visit a product or fundraiser page and read the text.	Font and text size are easily readable and are a good size	Fail

4. Code Review

Architecture

For the deployment of our application, we use a serverless architecture. The application consults third-party cloud infrastructure services provided by AWS (Amazon Web Services). The benefit of it is we have less work to deal with managing or supporting the server as well as the hardware needed for the application.

Reusability & Complexity

In order to make our code more reusable and maintainable, we use components to build up the user interface such as display dashboards, configuration settings, and etc. Since we can break down components into subcomponents in some cases, the reusability of our codes is going to increase.

Coding Style

For the coding style, we use UpperCamelCase for classes, lowerCamelCase for functions and variables, and uppercase characters separated by underscores for constants variables. Also, we use 1TBS (the one true space style) for brace placement.

However, there is one thing that we need to improve, which is the indentation spaces of our code. Some of us use two spaces to indent the code, and some of us use four spaces. Even though this is a very small problem, we should be consistent with the coding style in order to make our code more organized and professional.

Peer Review

Code review for the following feature in QA test:

- Each post shall display an image of the item
- The website shall not be overcrowded with ugly buttons and links
- The font shall not be too small

Code review for the following feature in usability test:

- Guests shall be able to browse donations page
- Guests shall be able to browse fundraisers page
- Users shall be able to place items up for donation
- Users shall be able to message each other through website's messaging system

Examples of Code Comments

```
{/* code review comment: code format - inconsistent indentation on the following 3 FundraiserCard components makes code hard to read.  */}
  <FundraiserCard title="Hospital Expenses"
                  description="Money required for the hospital and medicine expenses."
                  endorsements={4600}
                  requiredAmount="$10,000"
                  image={clinic}/>
  <FundraiserCard title="College Expenses"
                  description="Unable to pay tuition fees. Need money to pay all the money to the university."
                  endorsements={4200}
                  requiredAmount="$5,000"
                  image={college}/>
  <FundraiserCard title="Money for Candies"
                  description="Money required to buy whole lot of candies."
                  endorsements={40}
                  requiredAmount="$100,000"
                  image={candies}/>
```

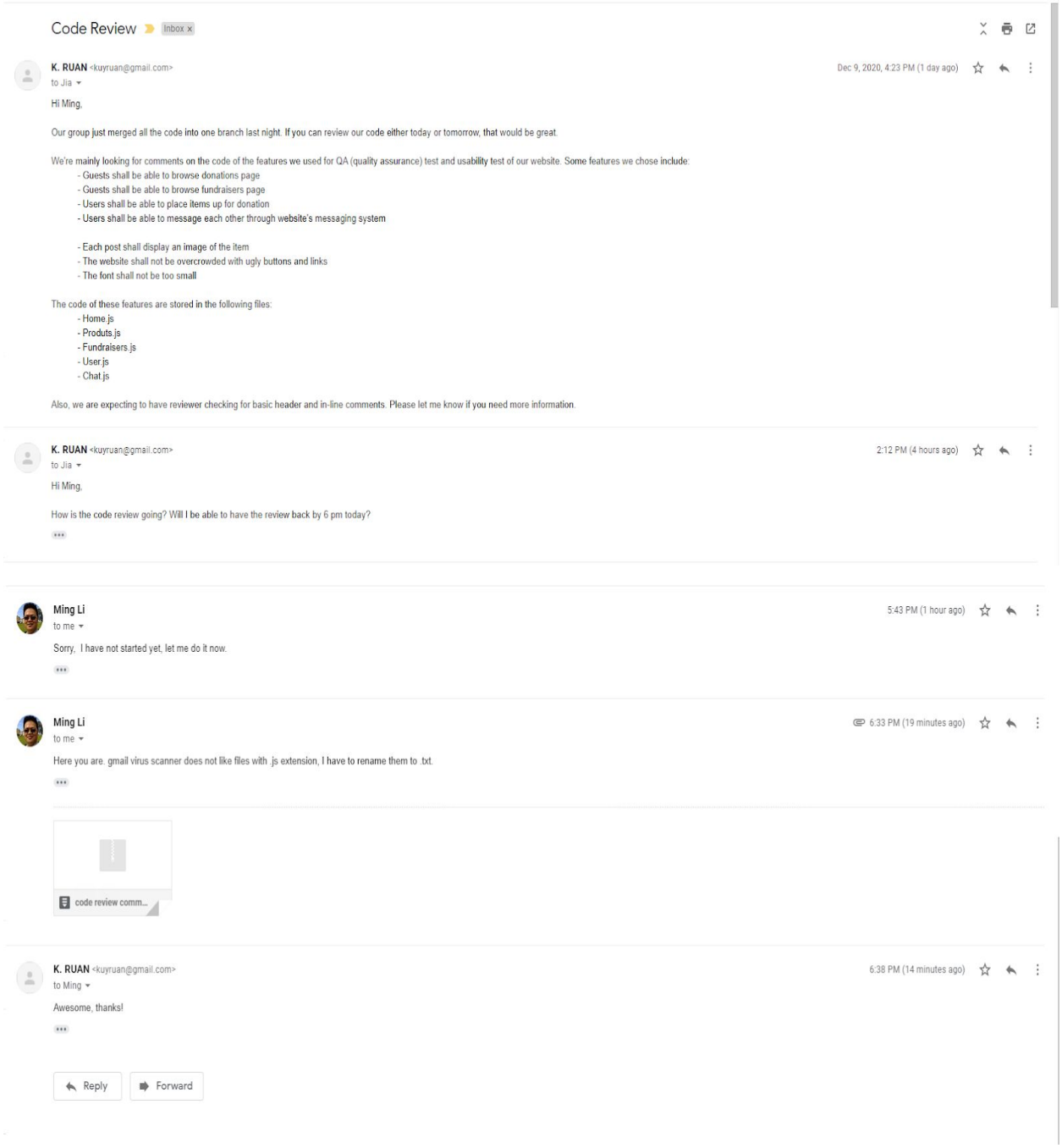
```
/* code review comment: code format:
    - inconsistent indentation: web format (css, html, js ..) recomment use 2 whitespaces indentation.
*/

class Chat extends Component {

    componentDidMount() {

    }

    openForm() {
        document.getElementById("myForm").style.display = "block";
    }

    closeForm() {
        document.getElementById("myForm").style.display = "none";
    }

    render() {
        const DUMMY_DATA = [
            {
              senderId: "perborgen",
              text: "who'll win?",
              user: false,
            },
            {
              senderId: "janedoe",
              text: "It's gonna be me of course.",
              user: true,
            }
        ]

```

```
/* code review comment: code format:
    - inconsistent indentation: web format (css, html, js ..) recomment use 2 whitespaces indentation.
*/

class Chat extends Component {

    componentDidMount() {

    }

    openForm() {
        document.getElementById("myForm").style.display = "block";
    }

    closeForm() {
        document.getElementById("myForm").style.display = "none";
    }

    render() {
        const DUMMY_DATA = [
            {
              senderId: "perborgen",
              text: "who'll win?",
              user: false,
            },
            {
              senderId: "janedoe",
              text: "It's gonna be me of course.",
              user: true,
            }
        ]

```

Peer Review Feedback

Based on the code we chose from QA and usability features, there is no big issue regarding algorithms or structural design. The main problem is probably the consistency of our coding conventions.

Email Communication History



Code Review Inbox x

K. RUAN <kuyruan@gmail.com> Dec 9, 2020, 4:23 PM (1 day ago)
to Jia

Hi Ming,

Our group just merged all the code into one branch last night. If you can review our code either today or tomorrow, that would be great.

We're mainly looking for comments on the code of the features we used for QA (quality assurance) test and usability test of our website. Some features we chose include:

- Guests shall be able to browse donations page
- Guests shall be able to browse fundraisers page
- Users shall be able to place items up for donation
- Users shall be able to message each other through website's messaging system

- Each post shall display an image of the item
- The website shall not be overcrowded with ugly buttons and links
- The font shall not be too small

The code of these features are stored in the following files:

- Home.js
- Produts.js
- Fundraisers.js
- User.js
- Chat.js

Also, we are expecting to have reviewer checking for basic header and in-line comments. Please let me know if you need more information.

K. RUAN <kuyruan@gmail.com> 2:12 PM (4 hours ago)
to Jia

Hi Ming,

How is the code review going? Will I be able to have the review back by 6 pm today?

Ming Li 5:43 PM (1 hour ago)
to me

Sorry, I have not started yet, let me do it now.

Ming Li 6:33 PM (19 minutes ago)
to me

Here you are. gmail virus scanner does not like files with .js extension, I have to rename them to .txt.

code review comm...

K. RUAN <kuyruan@gmail.com> 6:38 PM (14 minutes ago)
to Ming

Awesome, thanks!

Reply Forward

5. Self-Check on Best Practices for Security

Security has always been a major challenge while hosting or browsing the internet. There have been multiple instances where the security bugs of a website have been exposed and have been used for data leaks and data breaches. So, while developing a website or a web application it is necessary to take security aspects into outmost consideration and design the website with minimal security issues.

While taking on the security aspect of a website, there should be a distinction between sensitive data and non-sensitive data so that the sensitive data is handled and stored in a more secure way. In this web application, there are some data that are sensitive and need to be handled and stored more securely such as:-

- User data – the user data such as full name, email, zip code, phone number, username and password are considered to be sensitive data.
- Product information- entities of product such as name, description, images and the information about the user donating that product
- Payment information- since users are able to make payment for fundraisers, it is necessary to store payment information of the users in the most secure way.

It's never ideal to store passwords in a plain text in the database. If so and in case of data breach, the intruder may have the username and password of other users and have full access to the account of other users. Hence, passwords should be stored with encryption so that even in the case of the data breach, the intruders have access to encrypted passwords, which will be difficult to decrypt. In this web application, https will be used instead of http, since of course, https is more secure. The password will be sent as plain text from frontend to backend, and the backend then encrypts the password and stores the encrypted text password in the database while creating an account. Since, https will be used instead of http, the intruder won't be able to access the password even though the password is sent as plain text in the request header.

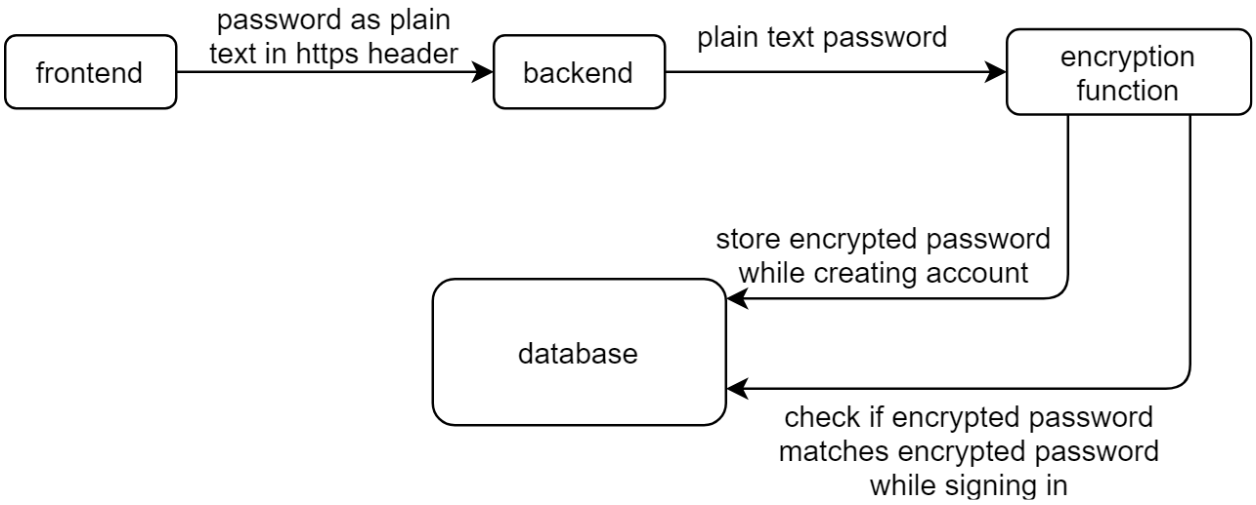


Fig: process of password encryption

No screenshot of the database is available now cause the backend/database of the website has not been made yet.

While creating a website, it needs to make sure that the data user is providing in the website is valid, hence comes the input data validation. Some of the data that is being validated are as:

- Email Address – since, an email address is used to send various emails to the user, it needs that the user while creating an account enters a valid email address. To validate email addresses, regular expressions will be used to search for symbols such as "@" and "." And furthermore a link will be sent to that email which when clicked verifies that the email is valid.
- Full Name / Username – the field such as full name and username can't be empty.
- Password – to make a website more secure, the user must create a strong password while creating the website. The password must contain at least 8 characters with at least one number, one uppercase letter, one lowercase letter and a symbol. Also, the password shouldn't contain the user's username or first name or surname.
- Product Name and description – the user must include some name of the product that she/he is donating along with description so that it will be easier for other users to search the product.
- Fundraiser description – it will be compulsory for the user to put on some description on why he/she needs the money while creating fundraisers so that the donors can donate with some reasons.

6. Self-check: Adherence to original Non-functional specs

Reliability

1. Database shall make a full backup every 24 hours.
2. The website shall provide services 24/7.

Response time

1. First Contentful Paint (FCP) shall be under 2.5 seconds.
2. Time to Interactive (TTI) shall be under 4 seconds.
3. Speed Index shall be under 3.5 seconds.
4. Total Blocking Time (TBT) shall be under 300 milliseconds.
5. Largest Contentful Paint (LCP) shall be under 3 seconds.
6. Cumulative Layout Shift (CLS) shall be under 0.1.

Hardware and networking

1. Data shall be stored in the MySQL database.
2. Website shall be deployed on Amazon Web Services.
3. Frontend shall be written using the React framework.
4. Backend shall be written using the Express framework.

Usability

1. Guests shall be able to browser all existing items.
2. 99% of users shall be able to find their items through searching for the first time without assistance.
3. 90% of users shall be able to find their items through category navigation for the first time without assistance.
4. Login is needed for posting donations.
5. Login is needed for requesting donations.
6. Website shall scale properly for mobile devices

Marketing, legal, licensing

1. The user needs to admit website policies in order to create an account.

2. Copyrighted material shall not be used against what the copyright holder allows.

Media content

1. Users' uploaded images should be compressed.
2. Each post shall display an image of the item.
3. Each post shall have a description of the item with maximum 10 words.
4. Uploaded item images shall be stored into the cloud

Privacy

1. Phone number is optional for creating an account.
2. Zip code is required when creating an account.
3. Email address is required when creating an account (username).
4. Only the database administrator can access users' personal information.
5. User's password shall be stored in the database after password encryption.
6. There shall be 2 factor authentications for user login.
7. Users shall not be allowed to login from a new device until 2 process authentication is processed.

Compatibility

1. The website shall be supported by Google Chrome.
2. The website shall be supported by Firefox.
3. The website shall be supported by Internet Explorer.

User Experience

1. The website shall not be overcrowded with ugly buttons and links.
2. The navigation between multiple pages shall be best.
3. The font shall not be too small.
4. The website shall provide small hints when using the website for the first time.
5. The colors of font and background shall be different.
6. The website shall avoid displaying fake posts.

Programming Methodology

1. A function shall not hold many responsibilities.
2. The app architecture shall be designed before writing the code.

3. Focus more on code quality and not on number of lines.
4. Error prone code shall not be pushed to github.
5. Proper comments shall be provided with the code.
6. Proper use of data structure shall be implemented.
7. The time complexity of different functions shall be regulated.
8. Users shall not see error messages on screen.
9. The tech stack shall be approved by all team members.
10. There shall be a proper use of design patterns.
11. The code shall not break when visiting a wrong endpoint

Storage

1. The system will have an overview of how many transactions it needs to handle at peak time
2. It monitors the volume of data the system will page
3. The system will have enough storage to scale up for increasing volume demands.
4. The system will have an overview of how much data to be stored

Security

1. The software must remain resilient in the face of attacks.
2. The behavior of the software must be correct and predictable.
3. The software must be available and behave reliably even under DOS attacks.
4. The software must ensure the integrity of the customer account information.
5. The server must not return a restricted web page to a user who is not authorized to access it.
6. The website must not accept overlong input data.
7. The application must not accept invalid URLs.
8. The system controls user access and sessions
9. It provides the secure channel for the data

Scalability

1. It supports the anticipated future number of open connections,
2. It acts to according to the response time per user request etc.
3. The system will be able to handle the highest workloads under which the system will still meet the performance requirements
4. It gives the system an ability to enlarge the architecture to accommodate more users.
5. It provides the system with an ability to meet the future needs of the business.

Team Contribution

Himanshu Garg: Worked on Product Summary with discussions from Front end and Bank end lead. Lead weekly team meetings every Monday.

Milo Abril: Worked on Usability Test plan. Regular to weekly group and after class meetings. Implemented the assigned work and more before the deadline.

Marlon Bustamate: Worked on many different sections - QA Test Plan, Title Page, Table of Contents. Highly attentive and active in all group activities and meetings. Completed assigned work and more before the deadline.

Kunyu Ruan: Worked on the code-review part. Always present in team meetings with great participation. Completed assigned work before the deadline.

Dipendra Dhoj Rana: Worked on Self-check: Adherence to original non-functional specs. Again, highly attentive and active in all groups meetings and discussions. Completed assigned work before the deadline.

Suman Basaula: Worked on Self-check on Best practices for security. Great Contributor. Always present in group meetings and always participates in group discussions. Completed assigned work before the deadline.