

# **CS3216: Assignment 3**

## **GivingCoupons**

*Group 1*

Bryann Yeap Kok Keong A0217687N

Lee Yong Ler A0219859J

Stuart Long Chay Boon A0217528B

Tee Weile Wayne A0149290B

# Table of content

<b>Table of content</b>	<b>2</b>
<b>URLs</b>	<b>3</b>
<b>Phase 1: Design</b>	<b>3</b>
Milestone 0	3
Milestone 1	3
Milestone 2	5
Milestone 3	6
Milestone 4	7
Milestone 5	8
<b>Phase 2: API Server</b>	<b>9</b>
Milestone 6	9
<b>Phase 3: Mobile Client</b>	<b>12</b>
Milestone 7	13
Milestone 8	14
Milestone 9	15
Milestone 10	16
Milestone 11	17
Milestone 12	19
Milestone 13	28
Milestone 14	30
Milestone 15	35
<b>References</b>	<b>36</b>

# URLs

Github: <https://github.com/cs3216/2022-a3-2022-a3-group-1>

Production: <https://giving-coupons.vercel.app/>

Admin email and password (to login and manage the campaigns):

Email: giving.coupons.sg@gmail.com

Password: unclesoo-please-give-us-A

Note: This is our production environment, please do not redeem coupons from the production campaigns we created.

Relevant app URLs to try:

1. To log into admin page: <https://giving-coupons.vercel.app/admin/>
2. To manage campaigns (for admin only, hidden from users):  
<https://giving-coupons.vercel.app/admin/campaigns>

## Phase 1: Design

### Milestone 0

Describe the problem that your application solves. (Not graded)

How can we encourage people to give back to society, to participate in charitable giving?

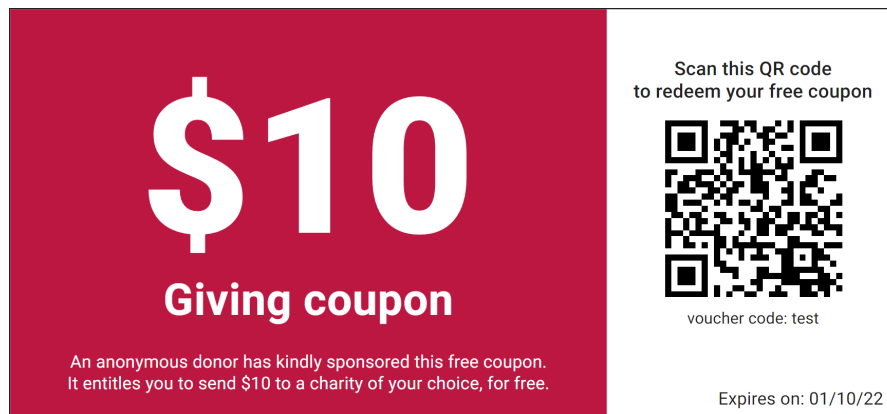
### Milestone 1

Describe your application and explain how you intend to exploit the characteristics of mobile cloud computing to achieve your application's objectives, i.e. why does it make the most sense to implement your application as a mobile cloud application?

Donors already give significant sums of money to various charities. These donors include philanthropists, corporations, charitable foundations, and governments. Our "giving coupon" idea aims to leverage these donations to raise awareness of these causes among ordinary individuals and encourage them to donate their money too.

For example, a donor might want to donate \$1000 to five charities. Instead of donating the money directly, they can instead work with us. We will split the \$1000

donation into a hundred \$10 "giving coupons". These coupons can be randomly distributed physically or electronically to a target audience such as NUS students.



*Coupon design for a campaign*

The coupon informs the user who picked it up that they have been given the opportunity to choose which charity receives this \$10. The user coupon directs the user to our webapp where they can learn about the different charities (e.g. malaria prevention programs, cancer research, animal welfare, local social services) and choose which cause they believe is the most worthy.

The objective of this project is to raise awareness and encourage users to donate their own money as well. This leads to a "money multiplier effect" on the original sum of \$1000.

As we intend for the vouchers to be distributed physically (as a start), users would clearly want a seamless experience that they can access from their mobile phones. A QR code linking to a webapp achieves this objective, making it fast, convenient and easy for users to access without having to download native apps or take out their laptops.

Note that we also have secondary, "admin" users who are able to login and create and manage campaigns. Our app allows these users to generate vouchers and monitor their redemption.

## Milestone 2

Describe your target users. Explain how you plan to promote your application to attract your target users.

Our target user is anyone that is currently not participating in charitable giving but can afford to do so. We think that many people are kind at heart, but do not have the awareness about different charities, and have no motivation to make the effort to find out and donate. It may be simply our culture that we do not really think about charitable giving.

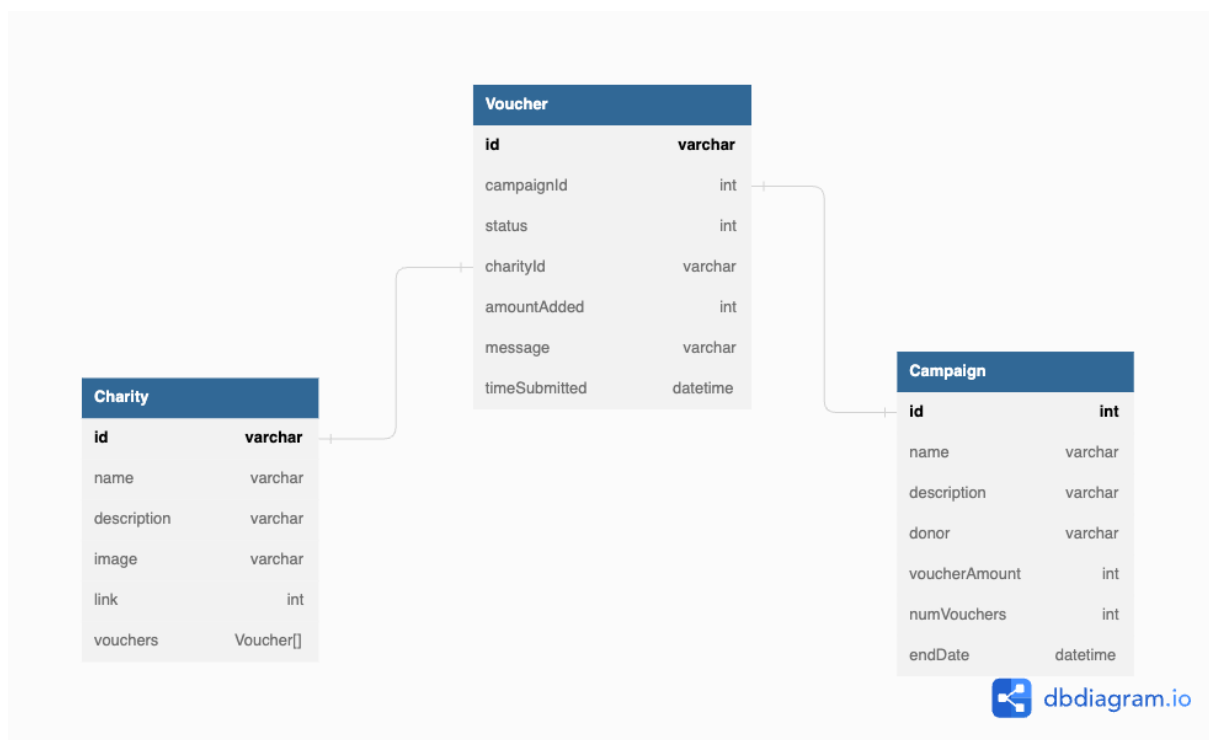
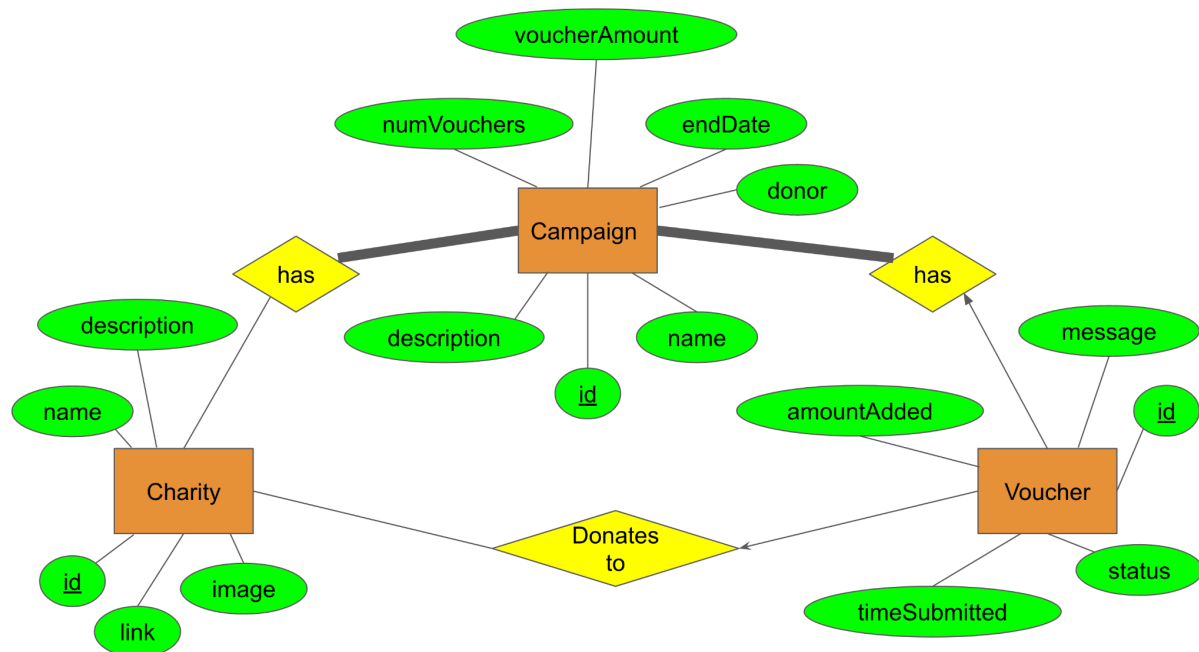
Our app does not target users who are interested in philanthropy, it targets users who are not interested or not aware and nudges them towards it. We think that our app is a small nudge to people to donate and a small reminder to them of the worthy causes around them. And when combined with other nudges and reminders, can make a significant impact.

To reach these users, we can start by physically distributing coupons to students directly or by leaving them at public areas (legally). To scale it up further we can give users who complete the form 5 more coupons to give to their friends. Distributing the coupons via email could also allow us to reach more users at lower cost.

To reach a larger number of users, we plan to pitch to and work with organisations. This also improves the legitimacy of our app, which is extremely important to acquiring users. For example a corporation could use this to encourage employees to participate in CSR donations. A school could implement this as part of educational programmes to encourage social responsibility. These coupons can also be given out at events as part of prizes or goody bags. Finally, the ultimate grantmaker is the government. This program could be implemented nationwide similarly to CDC voucher distribution.

## Milestone 3

Draw an Entity-Relationship diagram for your database schema.



## Milestone 4

Here are some popular alternatives to REST

GraphQL

gRPC (based on RPC architecture)

Explore one alternative to REST API (may or may not be from the list above). Give a comparison of the chosen alternative against REST (pros and cons, the context of use, etc.).

Between REST and your chosen alternative, identify which might be more appropriate for the application you are building for this project. Explain your choice.

### **Alternative to REST API:**

GraphQL

### **Comparison of GraphQL against REST API:**

#### **Pros:**

1. Gives frontend consumers of the API the power to explore fields and data they need.
2. GraphQL is also faster due to the above point.
3. Eliminates the issue in REST API where data might be under fetched (needing multiple entities at once) or over-fetched (using only a part of the fetched data)

#### **Cons:**

1. Not the conventional standard of designing APIs
2. Query complexity is high
3. Support and stability does not compare to REST API
4. Steep learning curve

Rest is more appropriate for our application as our team is more experienced in REST API from past projects, eliminating the buffer needed to learn a new tech stack. Furthermore, the flexibility offered by GraphQL (defining what fields of the data needed front the front end consumer) is not really needed by our application as most fetched data, if not all will be rendered in the frontend. The tradeoff between having the flexibility or saving data payload and the complexity needed to integrate this new tech stack is thus not justified. Hence we proceeded with using the REST API for our application.

## Milestone 5

Design and document all your REST API. If you already use Apiary to collaborate within your team, you can simply submit an Apiary link. The documentation should describe the requests in terms of the triplet mentioned above. Do provide us with an explanation of the purpose of each request for reference. Also, explain how your API conforms to the REST principles and why you have chosen to ignore certain practices (if any). You will be penalised if your design violates principles for no good reason.

Link to our Apiary: <https://givingcoupons.docs.apiary.io/#>

Below explains how our API conforms to the REST principles, and why we have chosen to ignore some of the practices.

Design practices:

- Accepting and responding with JSON: As mentioned in the coursework JSON is the most common format to send data in requests and responses. It is easy to parse and is supported by most frameworks. All of our requests and responses are in JSON.
- Stateless: Every request in our API contains all the information that is required to process that request.
- Using nouns in endpoint paths: Since the HTTP request methods are already verbs (GET, POST, PUT, etc), all of our endpoints are nouns (vouchers, campaigns, charities) that refer to the resource being manipulated. Additionally, the actions of each endpoint follow the semantic meaning of each HTTP request method.
- Using plural naming conventions: It shows that each group of APIs form a collection. We conform to this with resources such as charities, vouchers and campaigns.
- Handles errors and returns the appropriate status codes: Our API conforms to this practice very clearly.
  - 200 OK: Indicates a successful request (i.e. get requests)
  - 201 Created: Indicates that a new resource is created (i.e. Creating a campaign, creating a charity, etc)
  - 400 Bad Request: Indicates that the request cannot be processed due to a perceived client error. (i.e. Redeeming an expired coupon, or a coupon that has already been redeemed)



- 401 Unauthorized: Indicates that the client has to be authenticated to perform this request, but isn't. (i.e. Creating a new campaign, which can only be done by admins)
- 404 Not Found: The requested resource is not recognized.
- 405 Method Not Allowed: Indicates that the request method is not supported by the target resource. (i.e. Trying to send a DELETE request to /api/campaigns)
- 500 Internal Server Error: An error in the server when processing a request. This is the default status code from our API when an unanticipated error occurs.
- Good security practices: People are not able to access more information than requested. Additionally, APIs that require admin privileges are protected, and require a token to be set in the Authorization header of the requests.

A REST API practice we did not follow is resource nesting, such as '/users/123/orders/'. Since our database relationships, as well as our requests and responses, are relatively simple, we did not feel any need to do any resource nesting for our APIs. However, in our page routes, we did follow such conventions. For example, "/campaigns/1/print".

## **Phase 2: API Server**

### **Milestone 6**

Share with us some queries (at least 3) in your application that require database access. Provide the actual SQL queries you use (if you are using an ORM, find out the underlying query and provide both the ORM query and the underlying SQL query). Explain what the query is supposed to be doing.

In our application, we used an ORM called Prisma. Thus, we will provide both the Prisma query, as well as the SQL query that it generates.

The following query requests for a voucher object from the "vouchers" table that has an id of "15-002Y7". Since the prisma query includes querying for the campaign that the voucher belongs to, as well as the charities chosen by that donor in that campaign, it results in multiple SQL queries being generated. This query can happen when a user scans a QR code on a coupon.

## Prisma

```
const voucher = await prisma.voucher.findFirst({
  where: {
    id: voucherId,
  },
  include: {
    campaign: {
      include: {
        charitiesChosenByDonor: true,
      },
    },
  },
})
```

## SQL

```
SELECT "public"."vouchers"."id", "public"."vouchers"."campaignId",
"public"."vouchers"."status", "public"."vouchers"."charityId",
"public"."vouchers"."amountAdded", "public"."vouchers"."message",
"public"."vouchers"."timeSubmitted" FROM "public"."vouchers" WHERE
"public"."vouchers"."id" = "15-002Y7" LIMIT 1 OFFSET 0
```

```
SELECT "public"."campaigns"."id", "public"."campaigns"."name",
"public"."campaigns"."description", "public"."campaigns"."donor",
"public"."campaigns"."voucherAmount", "public"."campaigns"."numVouchers",
"public"."campaigns"."endDate" FROM "public"."campaigns" WHERE
"public"."campaigns"."id" IN (15) OFFSET 0
```

```
SELECT "public"."_CampaignToCharity"."A", "public"."_CampaignToCharity"."B"
FROM "public"."_CampaignToCharity" WHERE "public"."_CampaignToCharity"."A"
IN (15)
```

```
SELECT "public"."charities"."id", "public"."charities"."name",
"public"."charities"."description", "public"."charities"."image",
"public"."charities"."link" FROM "public"."charities" WHERE "public"."charities"."id"
IN ("make-a-wish", "sos") OFFSET 0
```

The next query occurs as a result of a PATCH request in order to modify the voucher fields. After scanning the QR code on the coupon, the user can choose to redeem the coupon, thus resulting in this query being run. The SQL queries that are generated basically searches the “vouchers” table for the voucher with the id of

“15-002Y7”, as well as searches the “campaigns” table to ensure that the id of the charity exists, then updates the fields with the user’s inputs from the form. This is all done in a single atomic transaction.

Let’s say that the user picks these options from the coupon:

I would like the money to be given to:

- ☒ Make-A-Wish Singapore  
☐ Samaritans of Singapore

We encourage you to donate your own money directly to these charities as well. This is **completely optional**. If you intend to do so, please tell us how much you plan to donate:

\$ 10

If you have any **feedback or questions** for us, or are interested in joining us, let us know here and include your contact details. Alternatively, email us at [giving.coupons.sg@gmail.com](mailto:giving.coupons.sg@gmail.com)

The UI of the form is very sleek!

## Prisma

```
const voucher = await prisma.voucher.update({
  where: {
    id: voucherId,
  },
  data: {
    status: redeemed,
    charity: {
      connect: {
        id: charityId,
      },
    },
    amountAdded,
    message,
    timeSubmitted: new Date(),
  },
})
```

## SQL

BEGIN

```
SELECT "public"."charities"."id" FROM "public"."charities" WHERE  
"public"."charities"."id" = "make-a-wish" OFFSET 0
```

```
SELECT "public"."vouchers"."id" FROM "public"."vouchers" WHERE  
"public"."vouchers"."id" = "15-002Y7"
```

```
UPDATE "public"."vouchers" SET "status" = 1, "amountAdded" = 10, "message" =  
"The UI of the form is very sleek!", "timeSubmitted" = "2022-09-23 15:02:54.026  
UTC", "charityId" = "make-a-wish" WHERE "public"."vouchers"."id" IN  
("15-002Y7")
```

```
SELECT "public"."vouchers"."id", "public"."vouchers"."campaignId",  
"public"."vouchers"."status", "public"."vouchers"."charityId",  
"public"."vouchers"."amountAdded", "public"."vouchers"."message",  
"public"."vouchers"."timeSubmitted" FROM "public"."vouchers" WHERE  
"public"."vouchers"."id" = "15-002Y7" LIMIT 1 OFFSET 0
```

COMMIT

## **Phase 3: Mobile Client**

## Milestone 7

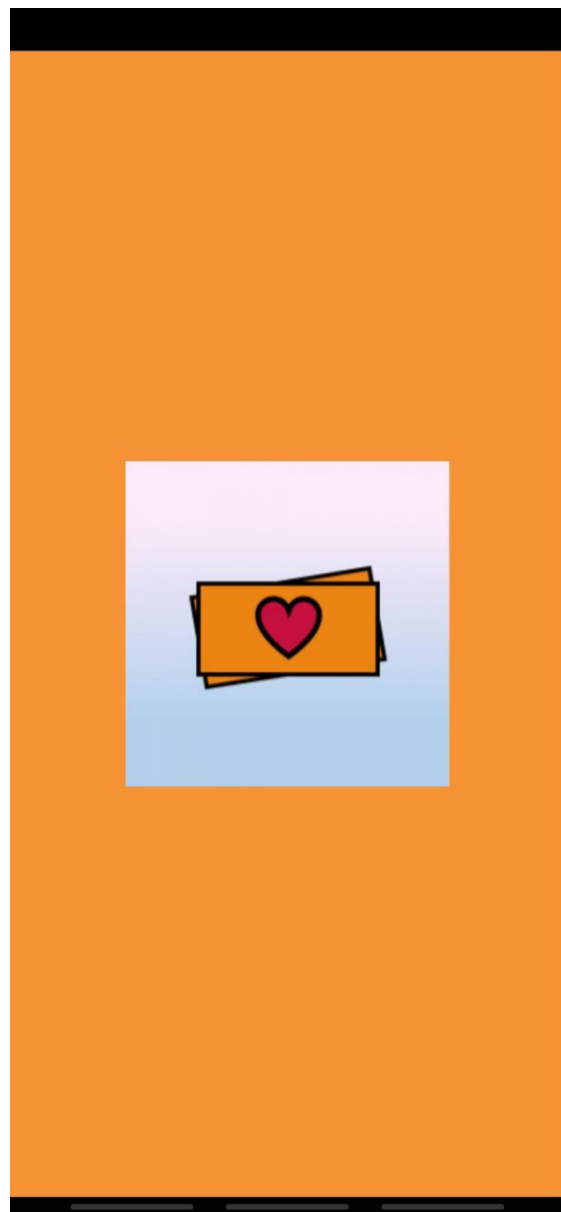
Create an attractive icon and splash screen for your application. Try adding your application to the home screen to make sure that they are working properly. Include an image of the icon and a screenshot of the splash screen in your write-up. If you did not implement a splash screen, justify your decision with a short paragraph. Add your application to the home screen to make sure that they are working properly. Make sure at least Safari on iOS and Chrome on Android are supported.

Our attractive icon and splash screen for our application:

*Icon*



*Splash screen*



Our inspiration behind the icon design is 2 coupons stacked on top of each other, representing the concept behind our application of utilising coupons and a multiplier stacking effect to spread awareness in Singapore of the charities available as well as the causes that they are serving. An eye-catching heart symbol supplements the icon, signifying our belief and mission of providing more warmth and love among the society.

## **Milestone 8**

Style different UI components within your application using CSS in a structured way (i.e. marks will be deducted if you submit messy code). Explain why your UI design is the best possible UI for your application. Choose one of the CSS methodologies (or others if you know of them) and implement it in your application. Justify your choice of methodology.

We will be using UI libraries as a standardised and aesthetic choice of styling, explained in detail in milestone 12. Hence, we do not need to rely heavily on CSS to do styling of our components. CSS styling is only needed when we overwrite the components from the UI library that is styled out of the box. I.e., minor tweaking using CSS only.

Our UI design is the best for our application due to the minimalistic and clean look we give to the users. We believe in giving more with less, and over styling components (with shadows, pop ups, colours) etc will only confuse the users. Other than that, we selected a uniform primary colour for our application (orange) to ensure a smooth transition between pages, leaving users feeling satisfied by the sleek look of our application. Also, components are styled in a way that conforms to Google's Material Design, making it seamless for users to use our app as if they are using other native apps.

In line with the ideal of charitable giving, we created a design that is warm, yet professional. We hope that users find our website to be trustworthy and legitimate. In our form design, key phrases were highlighted to aid understanding of our project to busy users who might not be interested in reading every word.

Our choice of methodology is the closest to Scalable and Modular Architecture for CSS. We chose this precise architecture compared to the others because the scalability of BEM CSS is complicated and unnecessary for our relatively simple application while OOCSS has some limitations in terms of customising each individual instance of the object and the CSS file might be too cluttered. SMACSS is

the best choice as it utilises the technique and clever use of wrappers and containers to make the component approach easier. Modularity, in this sense, is our main priority in choosing the CSS methodology. By abstracting design elements into modular components and their styling to individual files, we make development easier and achieve a more consistent style throughout our app.

## **Milestone 9**

Set up HTTPS for your application, and also redirect users to the https:// version if the user tries to access your site via http://. HTTPS doesn't automatically make your end-to-end communication secure. List 3 best practices for adopting HTTPS for your application.

We are hosting our next js application via vercel, which automatically serves it with HTTPS out of the box, providing secure browsing of our platform for our users. The SSL certificates are also automatically generated free of charge, saving developers time.

There are many best practices for adopting HTTPS for our application. One of them is in which we redirect users to the https:// version if the user tries to access our site via http://. After making this switch, we also visited each link in our application to test whether the HTTPS redirection is working properly in the deployment.

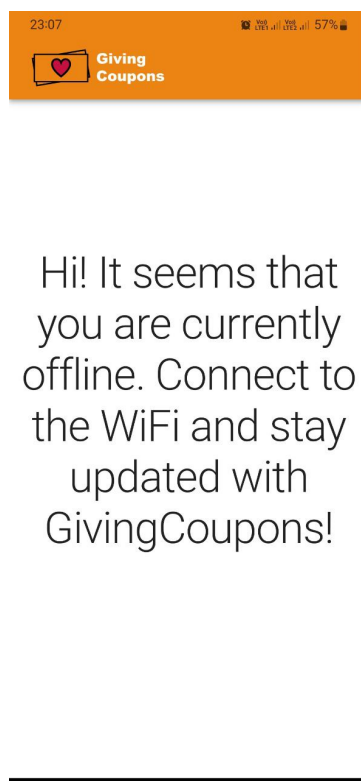
We also used HTTPS URLs for intersite URLs and relative intrasite URLs. Chrome developer tools were used to determine a successful HTTPS setup and redirection. Canonical links are also inserted into the application code to tell search engines that our application is best accessed through HTTPS. The setup is included in `_app.js`.

## Milestone 10

Implement and briefly describe the offline functionality of your application. Explain why the offline functionality of your application fits users' expectations. Implement and explain how you will keep your client synchronised with the server if your application is being used offline. Elaborate on the cases you have taken into consideration and how they will be handled.

With the library Next-PWA, we implemented automatic caching of the pages which caches the responses from requests. Upon a subsequent request, the cache will be used instead of requesting through the network. Users are able to see the home page, as well as their empty donate form / form response that is cached in the application.

As our application mainly allows users to scan QR code and submit a form / explore charities, this does not support offline functionality. An offline page is added such that the application fails gracefully when the user device is offline. The offline page is shown below:





## Milestone 11

Compare the advantages and disadvantages of token-based authentication against session-based authentication. Justify why your choice of authentication scheme is the best for your application.

Both session based and token based authentication enables the server to trust requests from users.

Advantages of token-based authentication against session-based authentication

1. The token based authentication method is more secure to use as the tokens are stored in users' computers and it cannot be tampered with. I.e Session based authentication has the sessions stored on the server, meaning that administrators are in control over the users like logging users out, changing their details etc.
2. There are scalability issues with session based authentication as the server has to look up and confirm session IDs when the users send a request with a cookie.
3. Using tokens helps to reduce the number of times the users need to send their credentials to the server.

Disadvantages of token-based authentication against session-based authentication

1. As the token is stored with the client, the server cannot carry out admin functionalities such as logging users out, changing their details etc.
2. By using session based authentication, API development cost for server side will be lower than the client side since token based authentication is more typical for clients with the rise of frameworks such as OAuth2 or OpenID Connect.

Our choice of authentication scheme: Token-based authentication

For our application where admins perform generally simple CRUD operations, we wanted the admin portal's performance to be relatively fast. Additionally, since this is a PWA, there are use cases that involve mobile phones. For instance, while distributing coupons, admins might want to constantly see real-time updates on their mobile phones of how the campaign is going (i.e. How many vouchers are redeemed, Whether any additional money was donated, etc). This was a use case that we experienced when performing our trial run of a campaign.

However, if we were to use cookie-based session authentication, the server would have to run a database query for every request that requires authentication. Alternatively, by using token-based authentication, there only needs to be a single database access upon signing in, to ensure the user is an admin. After which, every protected API request gets sent with a token that simply requires verification. This is much faster than a database lookup.

Additionally, we decided to use Firebase Authentication to implement our authentication, as they provide an easy to use SDK to integrate authentication into our application. As such, we used token-based authentication as Firebase authenticates using JWTs. Using token based authentication also allows us to more easily integrate with other SSO providers in the future.

## Milestone 12

Justify your choice of framework/library by comparing it against others. Explain why the one you have chosen best fulfils your needs. Lastly, list down some (at least 5) of the mobile site design principles and which pages/screens demonstrate them.

Material UI is being used as most of the team members are experienced in it, as well as the capability of creating beautiful styling in a short time that follows Google's Material Design. It is also a well maintained and popular library with a large user base. The ease of use of Material UI is one of the main motivating factors.

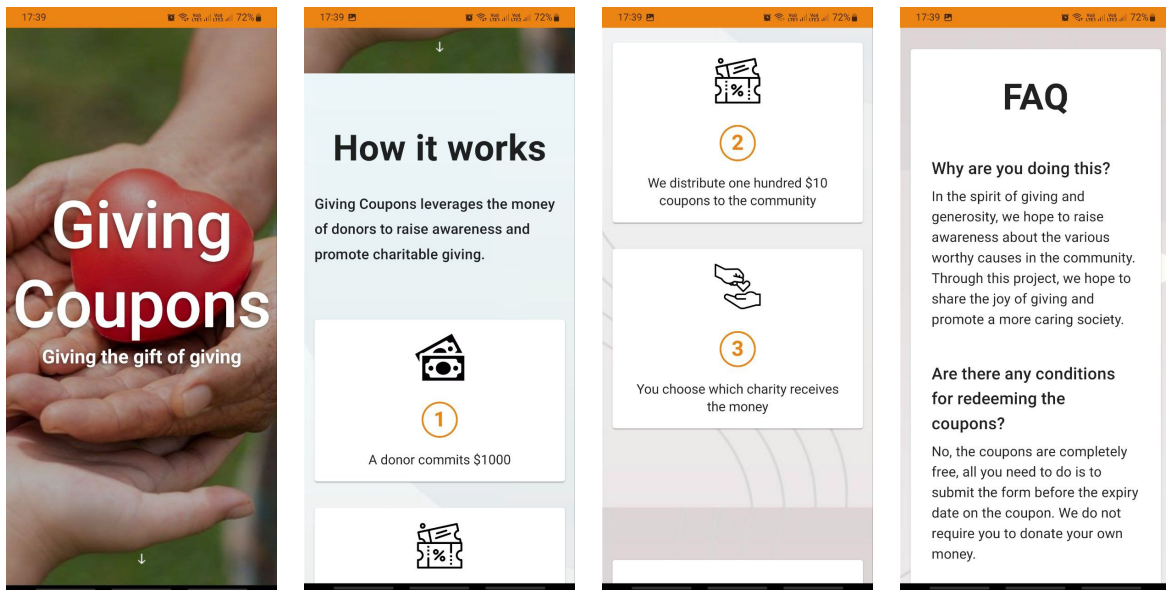
Material UI best fits our needs in particular for assignment 3 as it offers components that are well designed out of the box, while offering the flexibility to change the styling with ease. This is crucial for us to bring up an app within a short time frame while still providing native look and feel to the users.

Mobile site design principles along with corresponding screenshots:

## 1. Simple mobile page layout

Only vertical scrolling is present, with a single column layout. No horizontal scrolling is used. We made extensive efforts to declutter the layout as a mobile phone screen is typically small and we aim to make an impact on the users without complicating things. A minimalist homepage design with a seamless experience is curated for users to understand what we do better and to encourage them to donate.

Home page:



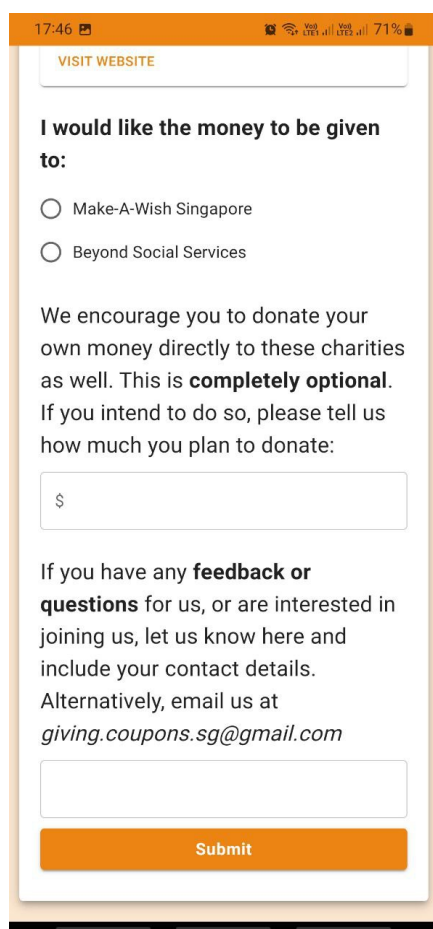
## 2. Responsive design + Mobile friendly sizes for components

Pages are designed with relative width and positioning such that it scales to the user's devices. This ensures a uniform experience across all devices.

Users will be frustrated if they need to zoom in and out to view a particular text or button. We designed the layout of our application such that each components' sizes scale well with each other and there is no need to zoom as everything is comfortably visible to the users. This also ensures that the users will not miss any important details when they zoom in and out.

User donate form page in different devices size:

*User donate form page in an android    User donate form page in an ipad view  
mobile phone view*



17:46 VISIT WEBSITE

I would like the money to be given to:

☐ Make-A-Wish Singapore

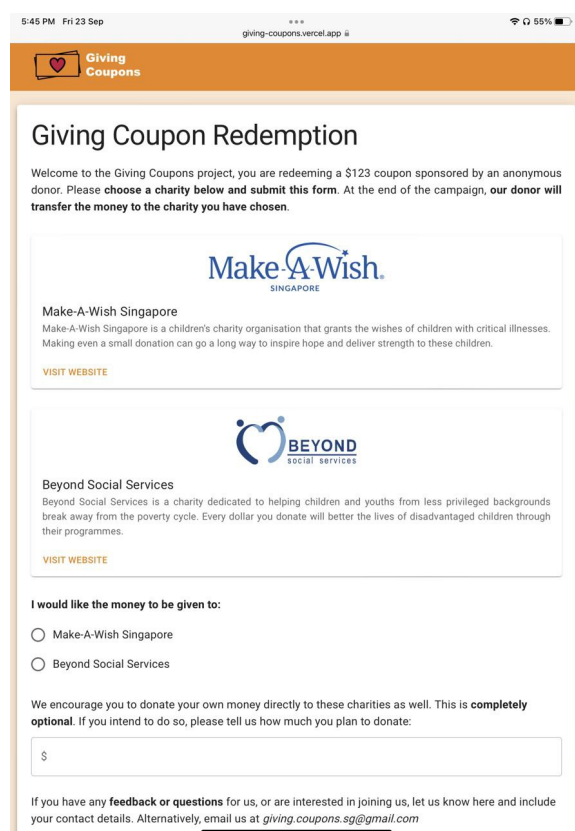
☐ Beyond Social Services

We encourage you to donate your own money directly to these charities as well. This is **completely optional**. If you intend to do so, please tell us how much you plan to donate:

\$

If you have any **feedback or questions** for us, or are interested in joining us, let us know here and include your contact details. Alternatively, email us at [giving.coupons.sg@gmail.com](mailto:giving.coupons.sg@gmail.com)

Submit



5:45 PM Fri 23 Sep giving-coupons.vercel.app

Giving Coupons

### Giving Coupon Redemption

Welcome to the Giving Coupons project, you are redeeming a \$123 coupon sponsored by an anonymous donor. Please **choose a charity below and submit this form**. At the end of the campaign, our donor will transfer the money to the charity you have chosen.

**Make-A-Wish SINGAPORE**

Make-A-Wish Singapore is a children's charity organisation that grants the wishes of children with critical illnesses. Making even a small donation can go a long way to inspire hope and deliver strength to these children.

VISIT WEBSITE

**BEYOND SOCIAL SERVICES**

Beyond Social Services is a charity dedicated to helping children and youths from less privileged backgrounds break away from the poverty cycle. Every dollar you donate will better the lives of disadvantaged children through their programmes.

VISIT WEBSITE

I would like the money to be given to:

☐ Make-A-Wish Singapore

☐ Beyond Social Services

We encourage you to donate your own money directly to these charities as well. This is **completely optional**. If you intend to do so, please tell us how much you plan to donate:

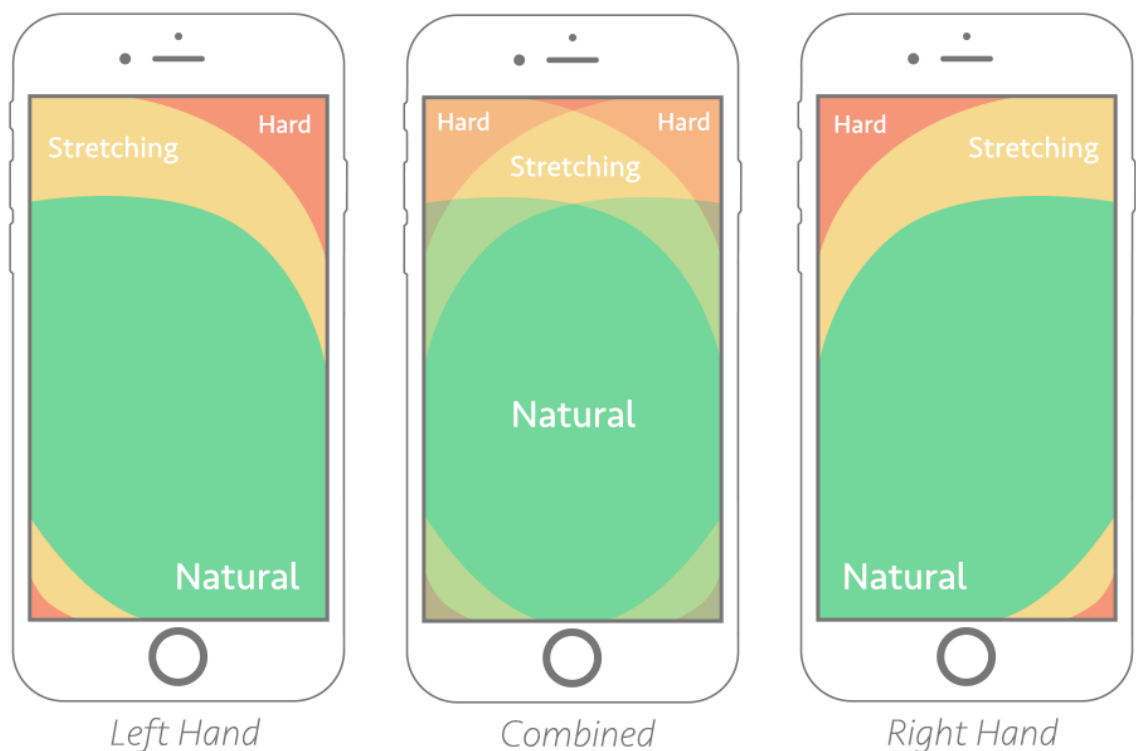
\$

If you have any **feedback or questions** for us, or are interested in joining us, let us know here and include your contact details. Alternatively, email us at [giving.coupons.sg@gmail.com](mailto:giving.coupons.sg@gmail.com)

### 3. Thumb friendly design on touchscreen mobile phones + Finger-Friendly Touch Targets

We ensure that the minimum target touch size is at least the size of a fingertip. Other than that, the touch targets are also not too close to each other, where we leave a margin in between to prevent the users from accidentally clicking the wrong thing. This maintains customers' satisfaction as they will not be constantly frustrated by pressing the wrong button when using our application.

We customise our designs to the typical thumb zones of touchscreen mobile users and place the common components, inputs as well as CTA accordingly in the combined natural region as shown in the picture below.



In the user donate form shown below, the radio buttons text are also configured to be clickable after taking the natural thumb zone into account. Also, notice that all other components are centred within the comfort zone, ensuring no thumb strains for the users.

17:46 VISIT WEBSITE

I would like the money to be given to:

☐ Make-A-Wish Singapore

☐ Beyond Social Services

We encourage you to donate your own money directly to these charities as well. This is **completely optional**. If you intend to do so, please tell us how much you plan to donate:

\$

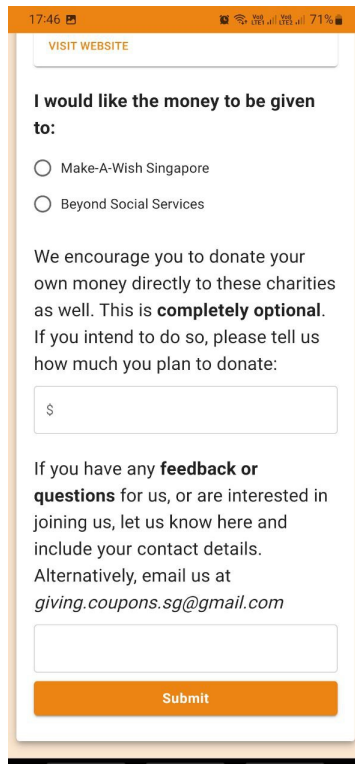
If you have any **feedback or questions** for us, or are interested in joining us, let us know here and include your contact details. Alternatively, email us at *giving.coupons.sg@gmail.com*

Submit

#### 4. Visual and functional consistency between mobile and desktop

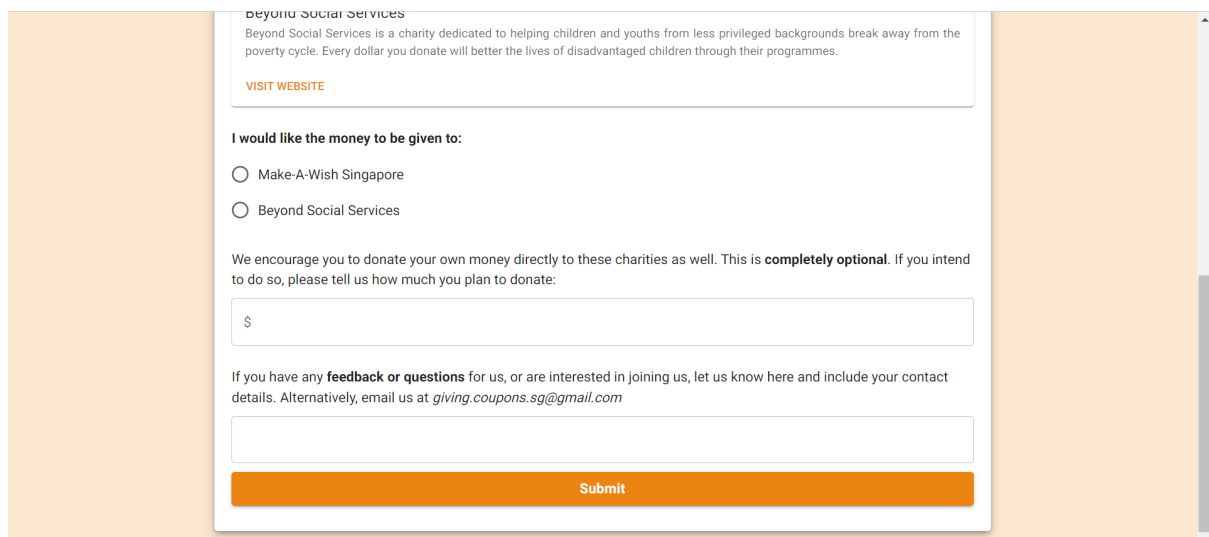
As our application is created as a PWA, the layout of the application in desktop and in mobile is similar to each other, thus eliminating any confusion from the users of any discrepancies. Users are able to switch between mobile and desktop with ease as all the components are in the same right place.

Mobile view of a user donate form:



The image shows a mobile view of a donation form. At the top, there is a status bar with the time 17:46 and battery level 71%. Below the status bar is a header with a 'VISIT WEBSITE' link. The main content area has a title 'I would like the money to be given to:' followed by two radio button options: 'Make-A-Wish Singapore' and 'Beyond Social Services'. Below this, there is a paragraph encouraging users to donate their own money directly to these charities, stating it is 'completely optional'. A text input field is provided for the donation amount, with a '\$' symbol. Below the input field, there is a paragraph asking for feedback or questions, with a text input field and an email address 'giving.coupons.sg@gmail.com'. At the bottom, there is an orange 'Submit' button.

Desktop view of a user donate form:



The image shows a desktop view of the same donation form. The layout is consistent with the mobile view, but the text is larger and the form is wider. The header 'VISIT WEBSITE' is at the top. The title 'I would like the money to be given to:' is followed by two radio button options: 'Make-A-Wish Singapore' and 'Beyond Social Services'. Below this, there is a paragraph encouraging users to donate their own money directly to these charities, stating it is 'completely optional'. A text input field is provided for the donation amount, with a '\$' symbol. Below the input field, there is a paragraph asking for feedback or questions, with a text input field and an email address 'giving.coupons.sg@gmail.com'. At the bottom, there is an orange 'Submit' button.



## 5. Use Only High Quality Assets

Our assets such as the images and icons are all designed with high resolution and placed strategically in the application. This is to capture the attention of users and drive calls to actions.

Home page:



## 6. Create a solid hierarchy of information

When users first land into our user donate form page, we tell them what we do and why we do it in short sentences. Also, there is a call to action to donate written in the very centre of the page. In other words, we are selective of the information we display such that they only need to know a little to perform an action. Within each charity, there is a link for users to learn more about them as well.

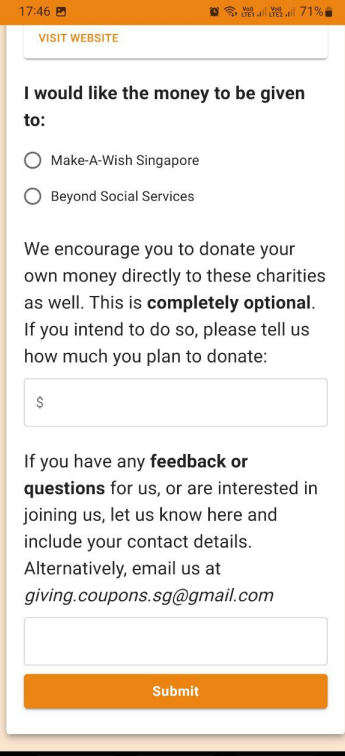
User donate form page:



## 7. Minimise user effort for data input

We only ask for the bare minimum information from the users, ensuring that it is not a hassle for them when filling up the forms.

As seen in our user donate form page below, we only ask for the minimum information from users. First is to reduce the effort required to donate, thus encouraging them to do so. Secondly, this ensures that users do not feel like their information is being collected too much, increasing the privacy of our users.



The image shows a mobile application interface for a donation form. At the top, there is an orange status bar with the time 17:46 and battery level 71%. Below this is a white header with a link labeled "VISIT WEBSITE". The main content area is white and contains the following text: "I would like the money to be given to:", followed by two radio button options: "Make-A-Wish Singapore" and "Beyond Social Services". Below these options, there is a paragraph: "We encourage you to donate your own money directly to these charities as well. This is **completely optional**. If you intend to do so, please tell us how much you plan to donate:". This is followed by a text input field with a dollar sign (\$) as a placeholder. Below the input field, there is another paragraph: "If you have any **feedback or questions** for us, or are interested in joining us, let us know here and include your contact details. Alternatively, email us at [giving.coupons.sg@gmail.com](mailto:giving.coupons.sg@gmail.com)". This is followed by another text input field. At the bottom of the form is an orange button labeled "Submit".

## Milestone 13

Describe 3 common workflows within your application. Explain why those workflows were chosen over alternatives with regards to improving the user's overall experience with your application.

A typical workflow would be as such: We / The admins pitch the concept to a donor, explaining the donation multiplier effect and that 100% of his money would be donated to the charities of their choice. If the donor agrees, we set up a new campaign based on the donor's details and generate the associated coupons for distribution. Upon distribution, a user obtains the coupons and scans the QR code to be redirected to our user donate form. When the users fill in the form and indicate the charities, the admin will be able to view it via our admin page.

A typical workflow for users who receive the vouchers would be as such: a user receives a voucher from us, and scans the QR code with his/her mobile phone. After which, it would redirect the user to our form, where the user can see the amount given to them to donate, as well as the participating charities and their descriptions, which he/she can choose to send the money to. He/she would be able to visit these charities websites by clicking the links in our form, and donate additional money to them if they would like. Lastly, they would be able to leave us a message and their contact if they would like to know more about our project, join us or even ask for proof of donation. The user can choose to install our PWA into his/her mobile phone as well.

This workflow was chosen over others such as allowing users to join via links shared online or through email from us was because we feel that these other methods might give users the impression that the link is a spam, due to the amount of spam messages forwarded online these days, as well as the chance that the email being flagged as spam. As our forms are unique and only allow one user to fill in per form, by creating shareable links, it would lead to a poorer user experience if another user tries to go to the form which has already been filled in. Moreover, by giving out coupons, we feel that it creates a more interactive and enjoyable experience for the users, enhancing the overall user experience, enticing the users to scan the QR code. It is easy to ignore a message and forget an email, but having a physical coupon serves as a physical reminder to users, and hopefully users will think twice before throwing a \$10 coupon away.

Legitimacy is very important for a project that seems to be taking money from people. Users are used to people soliciting money from them and we want to

emphasise that our project is different. Starting with the coupon design, we mentioned the word "free" three times. We continued to assure users of this in our form and FAQ section. We want users to feel like we are giving them money, and not trying to take their money.

Having the voucher code on the coupon assures users that their coupon is unique and valuable, the big \$10 text also entices users. We purposefully put less details on the coupon to encourage users to scan the QR to satisfy their curiosity.

We believe that by giving users a choice of charities, it gives them the opportunity to participate in a meaningful way, and feel that their actions have impact even though they do not actually give money. We hope that users feel that all of the causes are worthy and have a difficult time choosing, this engages them and encourages them to donate their own money.

As for admin workflow, we provide a functional dashboard to allow them to create and manage campaigns. A logout feature also improves security. The form is designed to allow admins to create campaigns efficiently.

On the campaign view, admins can click the view coupons button and see the coupons rendered in printable form, where they can directly print from their browsers into sheets of 4 coupons per A4 page. The size of the coupons is chosen to be roughly the size of a \$10 note, which fits nicely into a wallet. Admins can also click on the coupon id in the campaign dashboard to view or print or send an individual coupon.

The campaign view also allows admins to easily see relevant information like the amount of money per charity, the overall number of coupons redeemed, the redemption status and message from users if necessary. We also included the time submitted to allow admins to understand redemption trends.

## **Milestone 14**

Embed Google Analytics or equivalent alternatives in your application and give us a screenshot of the report. Make sure you embed the tracker at least 48 hours before the submission deadline as updates for Google Analytics are reported once per day.

Google analytics was chosen as the analytics tool for our application as it has many benefits compared to other alternatives. It is the most popular analytics tool, free of charge, has strong support and a large community as well as easy to integrate.

We integrated Google Analytics to track page views of our application, what type of devices they were using to access our application such that we can curate the experience of our application to them in addition to the demographics of the users. Also, since our application relies on the network effect to support the donation multiplier, the analytics is crucial for us to understand the reach of our network effect.

Other than that, event tracking was also set up for important user interactions such as users selecting a radio button of a charity (shown below) to donate to in the user donate form after they scan the QR code. This is to track whether they are actually interested in donating.

An example is shown below.

User selected the charity Transient Workers Count Too

Screenshot of the event being tracked by Google Analytics

Service Organisation in Singapore, dedicated to supporting and serving individuals with autism towards maximising their potential, helping them lead meaningful and quality lives in society.

VISIT WEBSITE

I would like the money to be given to:

☒ Transient Workers Count Too

☐ Handicaps Welfare Association

☐ Autism Association (Singapore)

We encourage you to donate your own money directly to these charities as well. This is **completely optional**. If you intend to do so, please tell us how much you plan to donate:

\$

If you have any **feedback or questions** for us, or are interested in joining us, let us know here and include your contact details. Alternatively, email us at [giving.coupons.sg@gmail.com](mailto:giving.coupons.sg@gmail.com)

Submit


Event count by Event name	
← selectedCharity	4
EVENT PARAMETE...	EVENT COUNT
Transient ... Count Too	4

1-1 of 1 < >

Besides, we implemented event tracking for the charities as well. Whenever a user clicks into the charity to learn more about them, it will be logged by Google Analytics. Both of these event tracking is aligned with our aim to spread awareness of the charities within Singapore.

User clicks ‘visit website’ for the charity  
Transient Workers Count Too to learn  
more about them


Screenshot of the event being tracked  
by Google Analytics



**Giving  
Coupons**

## Giving Coupon Redemption


Welcome to the Giving Coupons project, you are redeeming a \$10 coupon sponsored by an anonymous donor. Please **choose a charity below and submit this form**. At the end of the campaign, **our donor will transfer the money to the charity you have chosen**.



**Transient Workers Count Too**

Transient Workers Count Too extends help to low-wage migrant workers who have been injured, unpaid, or abused by employers. Doing low-wage work, migrant workers do not have savings to tide them through difficulties. Your donation helps TWC2 extend a lifeline to those in need.

[VISIT WEBSITE](#)



**HWA**

Event count by Event name

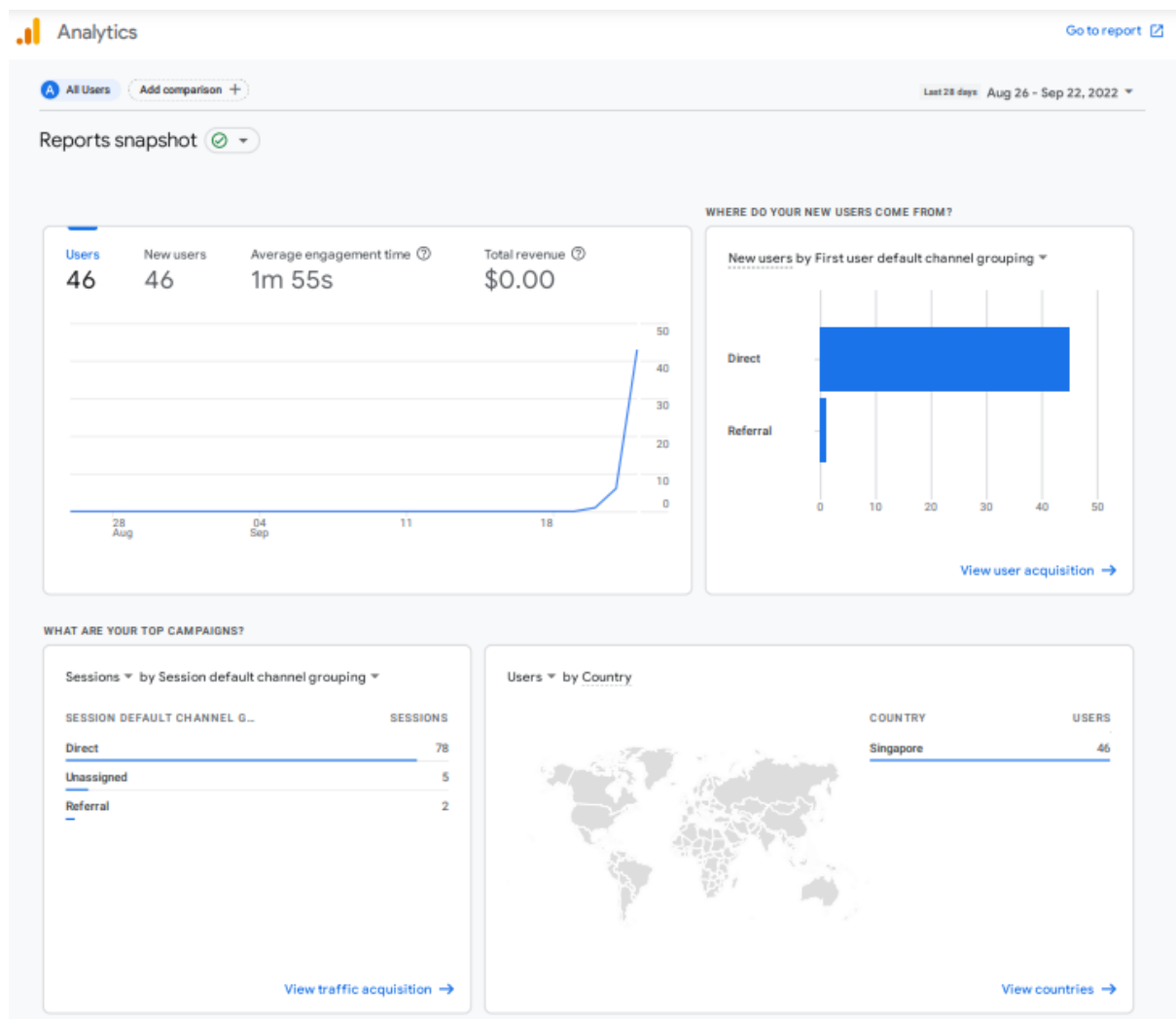
← link1

EVENT PARAMETE...	EVENT COUNT
<a href="https://twc2.org.sg/">https://twc2.org.sg/</a>	1

1-1 of 1 < >

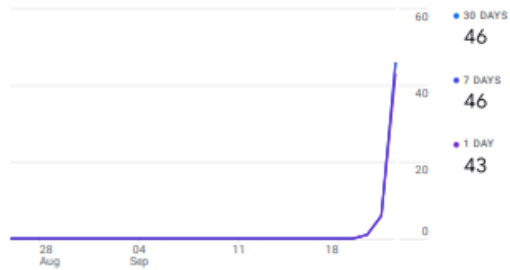


Attached below is a screenshot of our Google Analytics report:



#### HOW ARE ACTIVE USERS TRENDING?

User activity over time



#### HOW WELL DO YOU RETAIN YOUR USERS?

User activity by cohort

Based on device data only

	Week 0	Week 1	Week 2	Week 3	Week 4	Week 5
All Users	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Aug 7 - Aug 13						
Aug 14 - Aug 20						
Aug 21 - Aug 27						
Aug 28 - Sep 3						
Sep 4 - Sep 10						
Sep 11 - Sep 17						

6 weeks ending Sep 17

[View retention →](#)

#### WHICH PAGES AND SCREENS GET THE MOST VIEWS?

Views by Page title and screen class

PAGE TITLE AND SCREEN CLASS	VIEWS
<a href="#">Giving Coupons</a>	234
<a href="#">GivingCoupons.sg</a>	143
404: This page could not be found	14
500: Internal Server Error	14

[View pages and screens →](#)

#### WHAT ARE YOUR TOP EVENTS?

Event count by Event name

EVENT NAME	EVENT COUNT
<a href="#">page_view</a>	471
<a href="#">user_engagement</a>	340
<a href="#">session_start</a>	80
<a href="#">first_visit</a>	46

[View events →](#)

#### WHAT ARE YOUR TOP CONVERSIONS?

Conversions by Event name

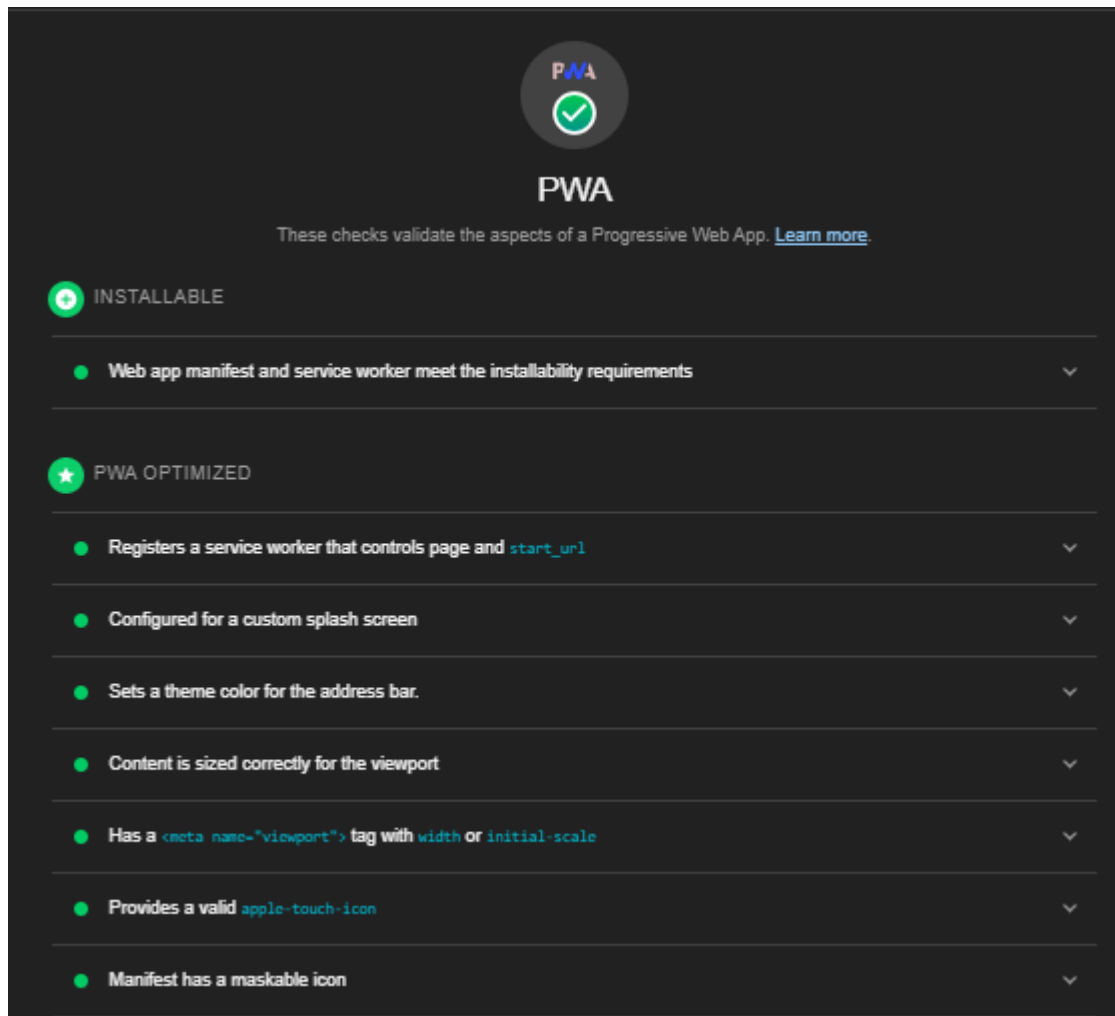
EVENT NAME	CONVERSIONS
No data available	

[View conversions →](#)

## Milestone 15

Achieve a score of at least 8/9 for the Progressive Web App category on mobile (automated checks only) and include the Lighthouse HTML report in your repository.

We achieved a score of 8/8 for the Progressive Web App category on mobile (automated checks only). The screenshot is attached here:



## References

- [1] Thumb zone design for mobile users:  
<https://www.smashingmagazine.com/2016/09/the-thumb-zone-designing-for-mobile-users/>