

**ASSIGNMENT COVER SHEET**

|  |  |
| --- | --- |
| **Student Names & ID:** | 1.  Chua Xue Ning Joey (2002230)  2.  Koh Huan Yin (2001376)  3. Kareem Mohamed Ashiq (2000923)  4. Aloysius Yeo Wei Jie (2002452)  5. See Rui Ling (2002110) |
| **Module Code:** | ICT1004 - Web Systems and Technologies |
| **Group:** | P4-6 |
| **Tutor’s Name:** | Goh Hui Lin Serena |
| **Assignment Title:** | heartBeat - Charity Organization |

1. **Introduction**

The team created a website for a fictitious non-profit organization called “heartBeat”. The organization supports 4 key features; allows users to donate to various fundraisers it hosts, allows for users to create fundraisers, allows for users to volunteer for various volunteering events and allows for users to create volunteering events themselves.

1. **Methodology**

During the initial phases of the project, the team was unsure as to how to process. To get things started and establish how things were going to work, the team decided to employ the AGILE – SCRUM framework as we believed that with a tight deadline and with the anticipated changes, we had to be flexible with how we managed the project. As how SCRUM was intended to be used, we conducted daily meetings discussing what we had accomplished since the previous meeting what we planned to do moving forward and discussed the issues each of us faced and requested for help if we were unable to solve it by ourselves. These meetings helped us tremendously and kept us on track.

1. **Design**

The team placed a lot of emphasis on the design of a website. We had to design such that users can identify the contents and nature of the webpage immediately upon glance.

We wanted the key features: volunteering events, fundraisers and donate to have their own respective processes and structured a flow that is straightforward yet simple to understand.

To execute the presentation of our webpage, we wanted the key features to be obvious enough to locate while making it look accommodating and welcoming to anyone visiting the webpage.

The team discussed the database tables together which covers all the CRUD of each table.

* 1. **Frontend**
     1. **Prototypes**

We started with a brief idea on how we wanted the web page to look like. Users can easily locate the individual features of interest by displaying it as the main content of the webpage. A navigation bar is placed to enable easy navigation through the webpage, alongside a log-in and log-out option to regulate users and participants. (Refer to Figure 1 and Figure 2 in the appendix)

However, we decide to change the UI of the prototype project after discovering inconsistencies in the UI design of the prototype. The current UI skeleton looks something like this with much lesser inconsistencies in the integration of other pages’ UI design. The design of the webpage does not follow through to our idea of a welcoming and vibrant page, whereby it looks clunky and dreary. Most importantly, the webpage was not responsive. As such, we decided to revise the UI design and take references to more modern webpages.

* + 1. **Final design**

We wanted to deliver the purpose of this project development, which is a non-profit organization that focuses on community-based projects and giving back to society. The first impression of the webpage is vital in capturing the interest or attention of users. Simple and welcoming and delivers a short and powerful message. The navigation bar was also revised to have a more aesthetically pleasing and straightforward design, which features all the functionalities to expect on our webpage.

The prototype UI was also revised to be more modern looking and describes what to expect upon clicking the navigation text.

As such, our webpage undergone a huge upgrade from the previous models and is comparable to other public sites that will provide an aesthetically pleasing and memorable impact on the user experience for everyone.

(Reference to Figure 3 in appendix)

* 1. **Backend**
     1. **Libraries and Frameworks**

We made use of a LAMP stack as it was what we used in the lab and we were familiar with it.

* + 1. **SQL Server**

Our webpage uses 5 main tables, each having its key features in the webpage. We could see how each table are all interlinked and related to one another, showing that no one part is independent and exclusive in the database. This requires us to collaborate and combine our efforts to make the project a success. (Reference to

1. **Implementation**

Once the design stage of the project was completed, we proceeded to implement the webpage. We finalized the content of all the pages on the website, pushed the pages onto the server and tested them. More details on the testing process can be found at the section labelled as “Testing” in this report.

* 1. **Content Management**

The content in our website was mostly made up by us, as this webpage belonged to a fictitious non-profit organization. However, we did refer to other fundraising pages to see what kind of fundraisers were being hosted on the website. We also delegated each member of the website a subsection of the webpage, which allowed for full control of the content produced and prevented contradictions being made.

* 1. **Technical Aspects**

As mentioned in “Design” section above, we used a LAMP Stack which had the necessary open-source applications to get us started almost immediately. To host the webpage, we used Amazon servers via its AWS Educate Program. To transfer the files to the web server, we used NetBeans and FileZilla. FileZilla also allowed for us to manage the files on the server easily, as without it we had to manually go into the server through command-line every time.

1. **Testing**
   1. **Quality and Standards**

The W3School Validation Test passed with no errors or warnings to show.

(Refer to Figure 4)

* 1. **Accessibility**

The following is an existing error found from our website & the reason they are unfixed:

* The colour and the background are contrasting as there is opacity over-laying the images in the carousel. This is checked according to the colour validation website provided by the axes test as shown below.
  1. **Responsiveness**

The images shown below are the screenshots of the footer and the navigation bar which is shared in all the pages running on Safari portrait and landscape view.

(Refer to Figure 6)

1. **Additional Features**

Apart from the basic features, the team managed to implement certain advanced features into our web project. We managed to implement Stripe on the frontend, Google Maps and session management on both the front and backend.

* 1. **Stripe API**

Stripe API is mainly used for the transaction of the donation made via debit/credit card. We use this API as a simulation to validate credit/debit card numbers that they are using to donate.

* 1. **Google Maps API**

Google maps API is used under the contact us section of the main page to display where our fictitious organisation is situated at. In this case, we set the coordinates at SIT@NYP.

* 1. **Sessions**

Sessions is mainly used for retrieving essential information of the user that is logged in and it is used in all the pages and it will kill the session when logged out. It is important because some pages you’ll need to be logged in in order to access and have a database and the knowledge on what the user does on the Web page. So, if there happen to have any errors, it will be easier for us to identify and remedy it.

1. **Contributions**

All the team members played an integral role in developing and creating the website. The team came together frequently to brainstorm various ideas, resolve issues and work on the web project itself. We gave ourselves distinct roles and responsibilities to focus on based on our expertise. Focusing on our strong points has helped us to work better and efficiently. We also helped one another out along the way and complemented each other’s weakness. A detailed explanation of the PHP pages on our web project can be found in the Appendix below.

|  |  |  |
| --- | --- | --- |
| **Name** | **Tasks Delegated** | **% Completion** |
| Joey | Created the following files:   1. wellwishes.php 2. about.php 3. fundraiser\_wellwishes.php 4. helped out 5-10% for some pages or functionalities 5. UI aspects   **Features yet to be completed:**   1. Ajax function 2. contact.php   **Reason:**  Tried to implement Ajax function as an additional feature to improve the website such as at the fundraiser\_wellwishes and donate page. This is not taught in school but the team thought that it would be a good feature to add to retrieve or search data without reloading the entire page. We did attempt it but the team could not figure out the problem and time constraint is another problem.  Contact page is a impromptu page the team decided on. The JS script and php 90% done. | 95% |
| Huan Yin | Created the following files:   1. Donate.php 2. Donation\_display.php 3. Fundraiser.php 4. Process\_donation 5. Makedonation.php 6. wellwishes.php 7. Footer.inc.php 8. Nav.inc.php 9. fundraiser\_.php 10. Contact.php 11. Index.php   Features yet to be completed:   1. move\_upload\_files: Unable to upload img files to MySQL. | 99% |
| Ashiq | Set up the following backend things:   1. LAMP Stack 2. SQL Database 3. Database Tables   Created the following files:   1. header.inc.php 2. nav.inc.php 3. footer.inc.php 4. index.php 5. donation\_display.php 6. Stripe Implementation | 100% |
| Aloysius | Created the following files:   1. display\_registered\_events.php 2. event\_creation.php 3. process\_event\_creation.php 4. process\_event\_register.php 5. volunteer.php 6. volunteer\_display.php   Features yet to be completed:   1. move\_upload\_files: Unable to upload img files to MySQL to be displayed in volunteer.php and volunteer\_display.php | 95% |
| Rui Ling | Created the following files:   1. register.php 2. process\_register.php 3. login.php 4. process\_login.php 5. session.php 6. logout.php 7. profile.php 8. profile\_update.php 9. deactivate.php | 100% |

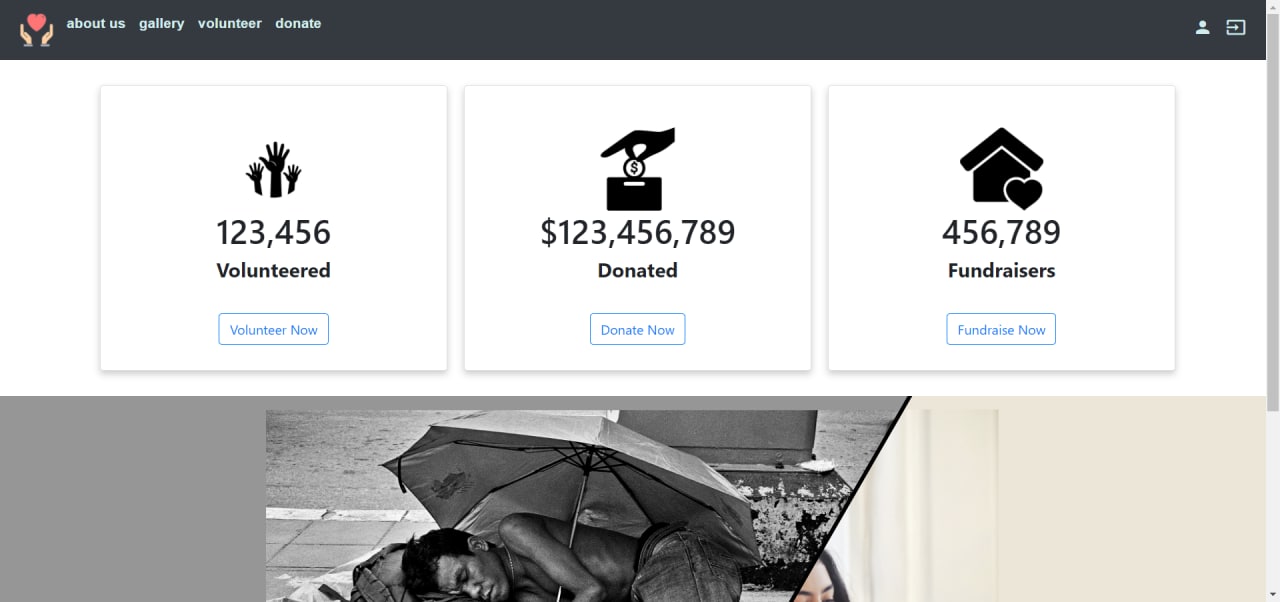
**Appendix**

**Figure 1: UI Prototype**

Graphical user interface, text, email

Description automatically generated

**Figure 2: UI Skeleton**



**Figure 3: Finalized UI**

A picture containing grass, outdoor, playing, young

Description automatically generated

**Figure 4: W3 Validation Test**

Graphical user interface, text, application, email

Description automatically generated

*W3school Validation Test*

**Figure 5: Database Design**

**A picture containing diagram

Description automatically generated**

**Figure 6: Responsive Design**

Graphical user interface, text, application, chat or text message

Description automatically generated

**Figure 7: PHP Pages**

|  |  |
| --- | --- |
| **Page** | **Description** |
| about.php | Brings you to about us page |
| contact.php | Contact us page |
| contact\_process.php | To inform whether the query sent to us are successful |
| db.inc.php | Php file containing database connection credentials and path |
| deactivate.php | Deactivates and delete account that was created |
| display\_registered\_events.php | Display events that are registered by the user via their profile, allowing them to see more into detail or delete their participation of the event. |
| donate.php | Display List of fundraisers causes saved from database |
| donation\_display.php | Display the details of the fundraiser events that user has selected and prompt user to donate which will then insert user info and donate amount to database |
| event\_creation.php | A form page that allows user to create volunteering event if they are logged in. Events can only span from 2020 to 2022 to regulate the frequency of current events. |
| footer.inc.php | display the footer which includes the navigation links, contact details and our mission statement. |
| fundraiser.php | A form page that allows user to create fundraisers event if they are logged in. |
| funraiser\_wellwishes.php | Display a list of well wishes according to the corresponding fundraiser ID. |
| wellwishes.php | Display a wall of well wishes comments retrieved from the database |
| head.inc.php | Php file that contains all the meta tags and libraries |
| index.php | Main page of website |
| login.php | Login page |
| logout.php | Kill session and logs you out |
| makedonation.php |  |
| nav.inc.php | Php file to include the navbar php codes |
| process\_donation.php |  |
| process\_event\_creation | The process page for event creation which sanitizes the user input and check for form errors like start date after end date or wrong contact number format. The process page will show successful event creation if it passes all the validation. |
| process\_event\_register | The process page for event creation which sanitizes the user input and check for form errors like users registering for the same event twice, or users that is already participating in an event on the same day. The process page will show successful event registration if it passes all the validation. |
| process\_fund.php | The process page to show the registration of fundraiser creation is successful. |
| process\_login.php | Retrieves data from database, shows the login requirements that the user input does not meet. Once login successful, it will redirect the user to home page, with session implemented. |
| process\_register.php | Insert data into database, shows the register requirements that the user input does not meet. Once registered successful, it will redirect the user to login page. |
| profile.php | Profile page that contains the updating of account details and deactivation of account. And as well as the list of events that the user has signed up for. Includes a deactivate button for users who wants to delete their account. Popup box will appear for user to confirm the deactivation. |
| profile\_update.php | Update the account details according to what user edits on profile.php |
| register.php | Account creation page |
| session.php | Commonly used as a include php file to implement whether a session has started for those pages that needed to be logged in. The file includes session timeout (one hour of inactivity) as well. |
| volunteer.php | Display List of volunteer events retrieved from database |
| volunteer\_display.php | Shows the details of the volunteer events when the user selects it and prompt user to volunteer to event. Which will then input user account info and volunteered event counter in database. The event counter is used to track the number of participants that have registered and the desired number of participants for the event. If the number of participants has been reached, the page will remove the register button. |