Reflective Essay

During the first coursework in which I had the chance to sketch the functionalities of my website this time, I had the possibility to use what designed and planned and try some features of the PHP language.

The aim of this project was to build a notebook to keep a record of work done for some projects and to keep logs of days and hours spent working on a project with the possibility to add a note. The use of a database allowed me to create separate notebooks for each project holding several logs. It is useful to note down and record thoughts, ideas and append when you had this idea or how much time you spent on it. It often happens that something comes to your mind, and you need to write it down before you forget, and now I have a tool to do it. Starting and ending dates are mandatory but for the time I used the actual time at the moment the log is written as default, so you only need to specify what day you worked. Some notes might be quick ideas, for example, that’s why I decided to be creative and insert now time as default and today buttons as default, the whole module is about being creative. I also added a rich-text editor to logs and project descriptions so users can also add images, links, and code. It is useful because it is possible to add personalised content with different sizes of font, images as well as links to share resources with other colleagues.

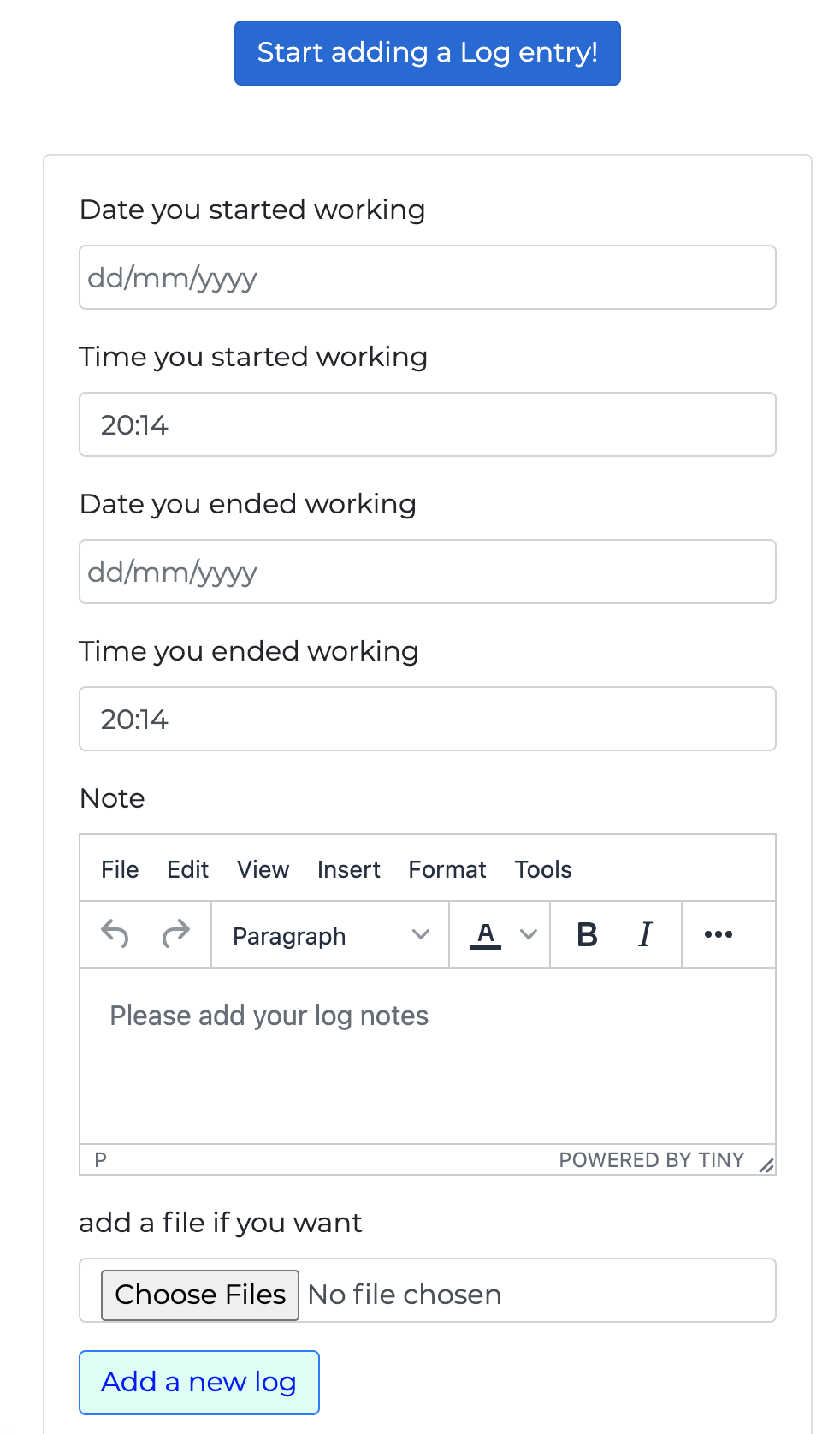
Graphical user interface, application, website

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Logs which are a list of notes taken by the user along with the time, date and files as we know are closely linked to projects so if a project gets deleted all the logs related to it must be deleted. This happens but the security issue I found is that the deletion of a log passing through the deletion process of the project does not remove from the memory a file attached to it. An error gets returned since there might be some files attached to logs posted by other people that are not the project's creator.

If you are the owner of a log and you enter inside the project to delete that log, the log gets deleted along with all the files attached because you are the owner of that file and log as well. But if you are the owner of the project it is impossible to delete the files since you don’t own them.

I tried it but a security error says that you are not allowed to perform that action:

if ($pps["creator\_id"] == $context -> user() -> getID() && $pps -> getID() == $restURL[0])

{

foreach ($pps -> logs() as $singleLog)

{

foreach ($singleLog -> files() as $upload)

{

$upload -> delete();

}

R ::trash($singleLog);

}

R ::trash($pps);

$context -> local() -> addval(["successDelete" => 1, "toast" => "1"]);

}

To overcome this kind of error I think that there is a need to implement some levels to distinguish the “owner” of a log and upload file to the owner of the project that should be able to remove everything related to the project.

Table

Description automatically generated

It is possible to share information with other people because this tool allows interaction with other people. You can have interactions in two ways: you can either ask a person to join your project and another user can ask to take part in your project. If any member of the project invites someone, they will need to accept it but if they ask to join the project, any member of the project can accept it. Each project has an owner, so the person who created it and also each project has co-workers. For this reason, I decided not to delete a project if the creator decides to delete their user account. For further implementation of the project, it would be appropriate to give the creator rights and power to a co-worker. It might be the oldest member added or someone the original creator decides to pick in advance. It might be convenient also to deactivate accounts for inactivity. It would be useful to pass project ownership to active members. However, if the project doesn’t have other members except for the owner, it might be deleted with no issue or deactivated with the owner's account in case the user would re-activate the account in the future.

A successor in this way might bring advantages in terms of more control since now only the owner of the project can delete a project to avoid people to join and bringing confusion and disorder.

Another handy future implementation to implement security would be the chance to kick people from a project or to block people so they cannot send you requests to join or to be added anymore (in both ways). I decided not to implement this kind of feature yet since I noticed it would have taken a great amount of time just to study a non-invasive change of database structure to edit users. I think that creating a sort of timer to know the period of activity of a user would be something advanced to implement so I preferred to avoid breaking the framework itself but the inactivity period to pass a project from the owner to a collaborator would be something needed.  
A very helpful feature in terms of security would be to have administrative powers for projects. When watching the video about security I heard a lot of advice on accessing with the lowest level of power possible in order not to do unrecoverable actions. When using the framework, I saw the different levels of users, admin, developer, and tester each with different powers and features. I would like to add something similar to the administrative powers for blogs or social networks pages/groups where there is a membership for people that can edit, others that can add things or mute people for a certain amount of time. Access rights are a powerful feature that needs to be implemented.

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

I wanted to get some ideas from modern social networks so I also implemented a profile in which users can see other users’ projects and also request to join.

Graphical user interface, application

Description automatically generated

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

Description automatically generatedGraphical user interface, application

Description automatically generated**

I must admit, it has not been easy creating this notebook using a framework and templates like Twig and red bean that I have never heard of much before. Still, I really enjoyed it because I had a bias on this module since web development is a computer science field that interests me a lot. I tried to start this coursework early because I was full of enthusiasm since I really wished to have a product to add to my personal portfolio. I used PHP once before since I wanted to apply for web development jobs, and I had the chance to put my hands on Laravel, so I knew what to expect. Besides this, I had difficulties because I tend to make things work using my own way (overcomplicating things sometimes) so it might happen that I code in a ‘bizarre’ way that works but it’s not coded clearly and efficiently.  
For this coursework, I had to follow strict rules and to follow principles due to the fact that for this project I had a customer, who was my employer and the Quality Assurance (QA) department all rolled into one so I must follow strict development QA guidelines. As I said I started quite early and I found a couple of bugs while using the framework and this gave me the ‘illusion’ to be in a real development environment like when contributing to GitHub projects.

Working in such a controlled environment has been challenging because I really enjoy independence, but it offered me an excellent product which I’m really proud to show in a portfolio with all the good practices that I will come across when working for an employer.  
My original plan was to do it on the second semester server-side as well, but the module has been cancelled as the lecturer has left. Hence, I’m glad I am doing PHP for the server-side components of this coursework. I know that PHP drives the majority of websites and I can definitely say that I have learnt a lot and now I am able to write code for other projects.

The framework is designed to build RESTful; I knew something about web requests because, for another project I used ‘postman’ to contact an API making requests and this time, I had an off-the-shelf method that could retrieve all the information for me. In the beginning, I thought it would have been easy to think to design a page for every operation like one to create, one to add logs, and one to show stuff but having something handy already implemented was much more straightforward.

I learnt the importance of using ORM to sanitise inputs and avoid malicious attacks like SQL injections. I didn’t know much about Red Bean, which is relatively simple and very powerful, so it is well-documented and widely used. The same approach has been used with Twig which I didn’t know to be a template engine to generate HTML outputs. I had to research a lot to understand how to use them and with this, I also learnt the importance of security to avoid using plain HTML to reduce errors and speed development. I managed not to mix HTML and PHP and to give useful comments and meaningful variables to be readable if I will work on this project in the future or if someone else would work on it.

I used Bootstrap once to make my website so I knew I could do nice tables and make it mobile-friendly. Having a mobile version is also important to reduce stuff to load for mobile users and user experience. Compatibility is important with other browsers and devices, and it is fundamental to know how to get the most out of every browser. It was also nice that the framework had font awesome already implemented and easily usable to be imported as the framework includes sets of macros that will help you create forms and models too.

Twigs and bootstrap gave shape to my ideas because my concept was already drawn in my storyboard but then I had to put in code. Basically, in this system, users have a list of projects and every project has a list of logs. So, I decided to give it a tabular representation. I knew that I had to keep things easy, so I opted for something minimal and clear using white backgrounds and light colours. I wanted the project page to be the most important page, accessible from every other page since connections to pages are important in SEO. I also designed log and project pages to be similar one to the other to give the user illusion to have never left the page. To increase this illusion, I made it RESTful.  
The first page when visiting my logging system is the index homepage, I designed it to be a static page that brings you to the logging page if you are not logged in or to the project page.

The navigation bar has similar features, if you are not logged in, the menu has links only to the About page, contact and login pages. The search bar is always present but to go to the profile page of the person you must be logged in. After accessing, it is possible to go to the project and log pages; instead of the login button, you can access the profile or log out.

The bootstrap features I implemented were similar to the ones I already used in my personal website, as a background-free image from ‘pixabay’ and some CSS was added to create new files. I also used the About page to give users an introduction to the tool and invite them to join. After the login, the user on the About page can read a sort of quick start guide with a brief explanation of what to click to make a start. In the About page, I added some static screenshots (you can find them in the assets folder).

To retrieve projects from the user I used the user model:

* projectNotCollaborating(string $user): array which passed a user as a parameter returns projects that the user creates in which you are not collaborating, I used this function in the profile page to return projects in which you can collaborate (ask to).
* project(): array which returns all the projects created by the user
* projectAskedToJoin(): array, this function took me a lot of time to optimise and make work efficiently since it gets all the projects the user logged in creates and all the projects the user is actually collaborating to and for each of them it returns the project itself and the user objects of the people who asked to collaborate. This method is used to accept requests from people who want to join projects.
* projectCollaborating(int $parameter): array this function gets 0,1,2 (for now) as a parameter which is the 3 status of a request and returns projects in which the user logged in has that status [0 if you asked someone to join your project, 2 if you asked the project owner to join their projects and 1 if the user is collaborating (accepted)].

The project page shows the user all the projects they own or are taking part in. Here the user can go to the log page related to that project, create a new project, edit the description, delete a project they own or add logs. Users on the project page can also invite colleagues to join their projects or accept requests from other users to join their project or accepts other users’ requests. The project page not only shows information but also accepts requests to create new projects (or edit them), to invite people or to create logs of projects. The project has a model to retrieve logs.

The log page also has a model to retrieve all the uploaded files related to each log.  
The log page is really like the project. It retrieves information from the URL to show the appropriate data. It accepts forms to delete logs (like on the project page for projects) to create a new one or even just to upload a file to it.

I have imported two JS scripts and CSS for the date picker and time picker to facilitate the user in the selection of the data and time for their logs. Any piece of information inserted is checked both on the client side when writing it and on the server side when validating it.

When implementing a new functionality, I always tested it and the beautiful and breath-taking part of web development is that you instantly see if what you wrote breaks everything or if it works. Debugging is hard since sometimes you have to correct a big block of code before seeing the result on the front end but still, you can do unit tests dividing functionalities into small bits.

When testing it I focussed on error checking:

I started from the base, from being sure that to perform any action the user must be logged in; then the user must have had the rights to perform certain actions or even only to visualise data like if you go to the log page related to a project you cannot see, an error is returned.

For validating inputs, I used where possible parsley (on twig pages) and then checked again using PHP in the back end. When accessing logs, I checked if the project id selected was a numeral otherwise if also trying to retrieve a non-existing or access-forbidden project rather than causing an internal error I created a specific 404 page for logs. Validation is important for this matter as well, not only to block malicious attempts but also to avoid unwanted errors. When retrieving something from the database I always check if the result is ok otherwise just return an error, I never ignore error messages, related to the server side so generate them from PHP code or from the developer console (from different browsers) to be sure what I was running was ok and now also the ide I used returned errors to facilitate debugging. Moreover, to extend basic security if someone tries to bypass validation on the front end I added a function to sanitise code when the user can add data to the database to avoid injections.

About security, I also learnt that it’s important to disable directory listing using .htaccess and that I can remove file extensions from my website address, or I can hide identification like php versions from the php.ini of the server. CSP has been really a challenge because I never came across it and it’s been hard to understand how to make it work even with the documentation. I used Subresource Integrity when I imported JavaScript files for date picker and time picker knowing that the browsers can check the checksum, rather than downloading the file or importing it directly, it has been good to use this security feature.

I had to format my code rewriting the rules because my ide automatically formatted my code using its own rules and then I had to rewrite the rules first and then to be sure it didn’t change the order while I was writing; it is still part of the experience. For the graphical part, I tried to make it looks similar to my storyboard because I really liked it and I was very proud of what I produced so I decided to keep the style and fonts.

This piece of coursework excited me also because I would like to work for a living doing web development. Since I planned to challenge my creativity while learning new skills, I tried to implement some nice functionality by reading the documentation I’ve been given for the constraints and practices set as default and mandatory. I implemented some smart checks to allow the user to perform something particular like asking people to join a project or for you to ask to join other’s people projects.

. I know I could have implemented it in a lot of different ways like sending an email or with a notification, but I wanted to try something that could have been stable to the website without relying on something “external” like opening an email to avoid the user to quit the website and with the lower number or interactions possible from the user.

When at the beginning I tried to implement something more complex to start understanding how the framework worked I came across good practices for a good user experience like not asking users to repeat emails when changing it and for privacy limiting what information the user was getting. I was risking having a privacy issue when I risked showing the name of people directly when randomly I thought to use checkboxes for co-workers. Now I have implemented something similar to a friend request in a social network, so the user types a name and it gets as result names of people that match what the user typed. All the results are then filtered to avoid adding users already added to the project. For a future implementation it would be useful to improve performances by not retrieving all the results matching the name but a limited number and then if the user is not happy, load more results and so on. Implementing multiple user interactions gave me a broad overview of all the tools from the database with the shared tables to parsley rules and checks in the front end.

Regarding security, I also learnt that when uploading files, the upload should not be public and that for best security it is important to block connections with external resources so for example to add the rich-text editor I had to add the domains to the CSP as specified in the Framework Security documentation.

I also learnt how to add appropriate content to have the best position in search engines possible, this also helps to improve load time and speed. I created a well-defined structure (e.g. ordered headings) with keywords, meta tags and description at the beginning so search engines can read it since they read only the beginning of the page. I learnt that there are multiple ways to include meta tags following Dublin core or Open graph rules. In order to improve speed, I have seen what was suggested to improve in some tools like Pingdom and power mapper which suggested for example checking for broken links, typos and for images that don’t have the ‘alt’ field. For page loading speed a great factor is “cacheability” reducing the load on the server, latency and improving performance and the ability to handle more queries. There are many types of cache where to keep copies of frequently accessed pages. I noticed that the framework was taking care of the cache for me when I had to change my function that displayed projects shared among co-workers returning from the same call with different results depending on the field inserted as a variable. To test it I had to clear the cache while developing it otherwise my method didn’t work successfully at first, so I had to make small changes each time and clear the cache to see the results. I noticed another example of the framework keeping data cached when I added some CSS directly on twig.

**To see the website at work:**

[**https://youtu.be/ENetnAoml1A**](https://youtu.be/ENetnAoml1A)