ISAD251 Database Applications Development, Resit.

Links to coursework:

GitHub repository

https://github.com/DigSwine/ISAD251-Resit

Application

https://github.com/DigSwine/ISAD251-Resit/tree/master/Application

Documentation

https://github.com/DigSwine/ISAD251-Resit/tree/master/Documentation

SQL

https://github.com/DigSwine/ISAD251-Resit/tree/master/SQL

Hosted web pages

Webpage URL

http://web.socem.plymouth.ac.uk/isad251/MWilson-Slider/Application/Website/index.php

Server

\\CENT-5-534.uopnet.plymouth.ac.uk\ISAD251\MWilson-Slider

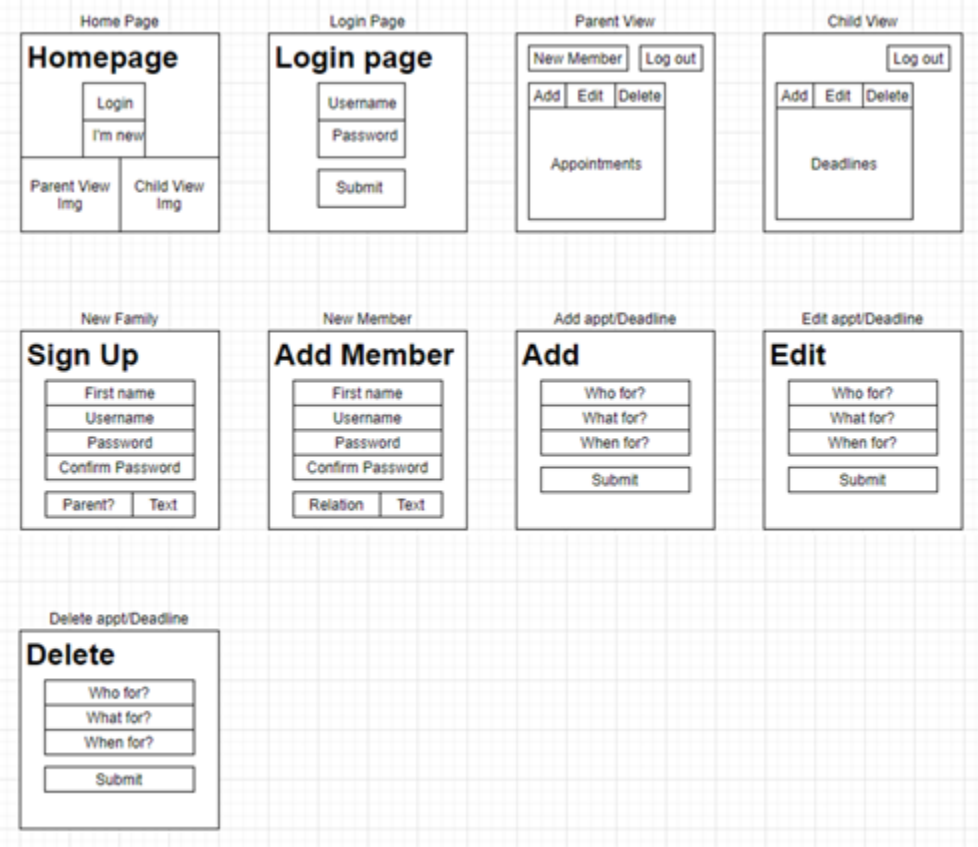
YouTube Video

https://youtu.be/m8K-2s13D3Y

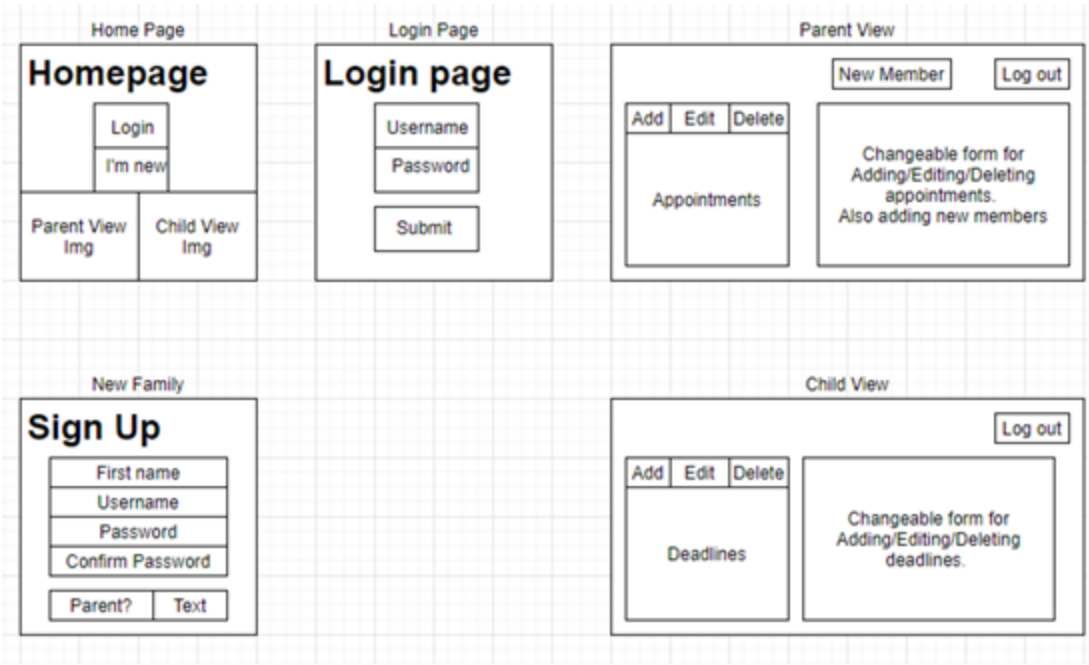
Application design document

Storyboard

Initial

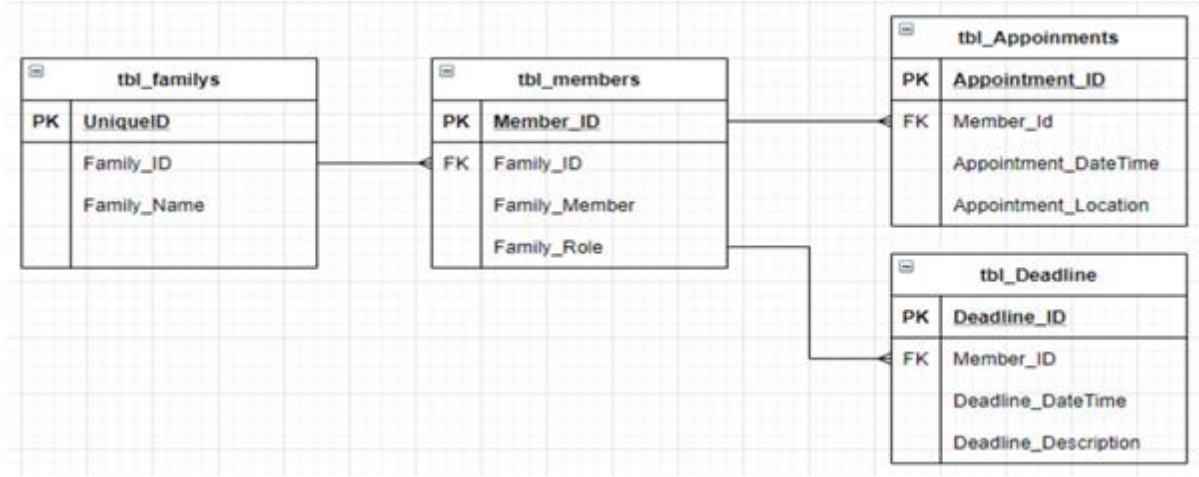


Final

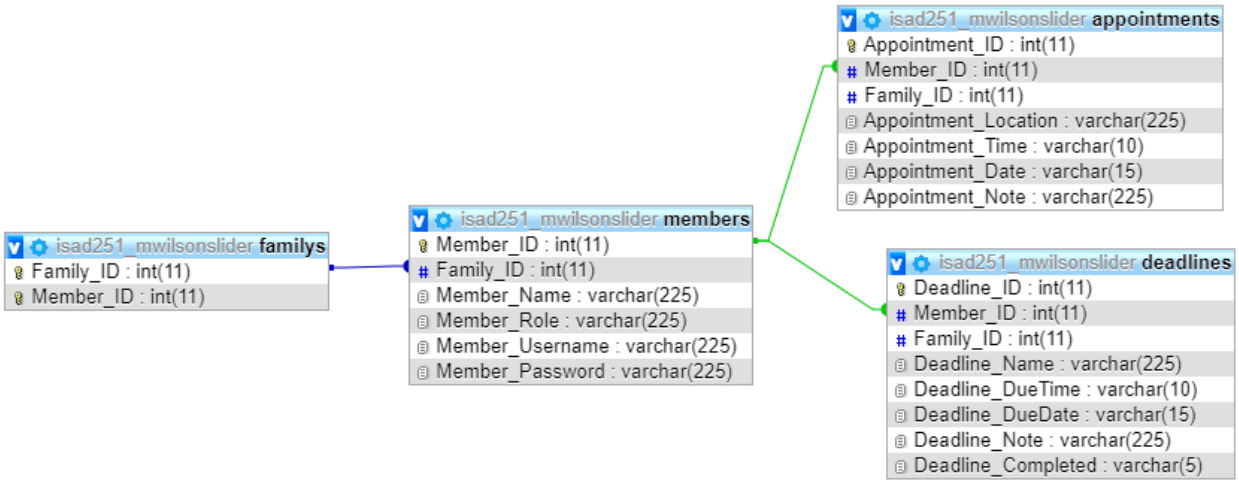


Entity Relationship Diagram

Initial

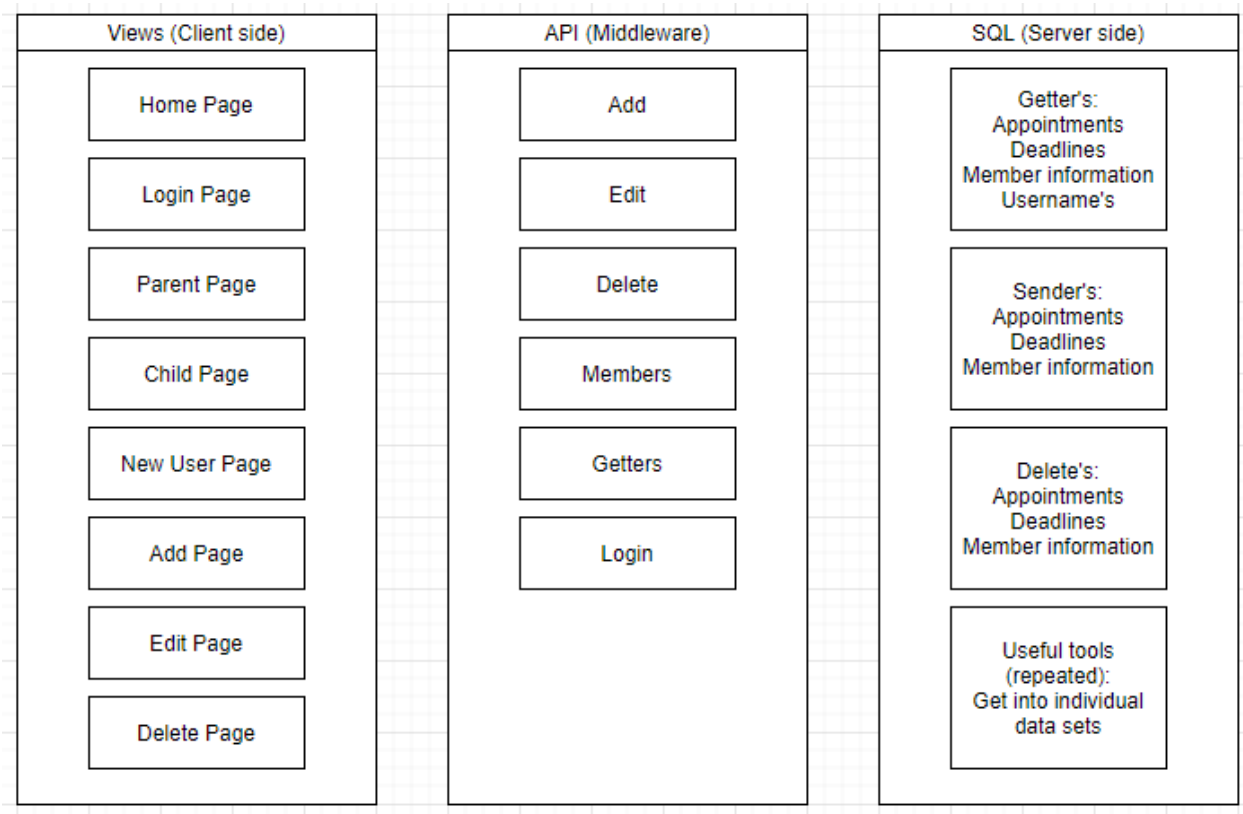


Final

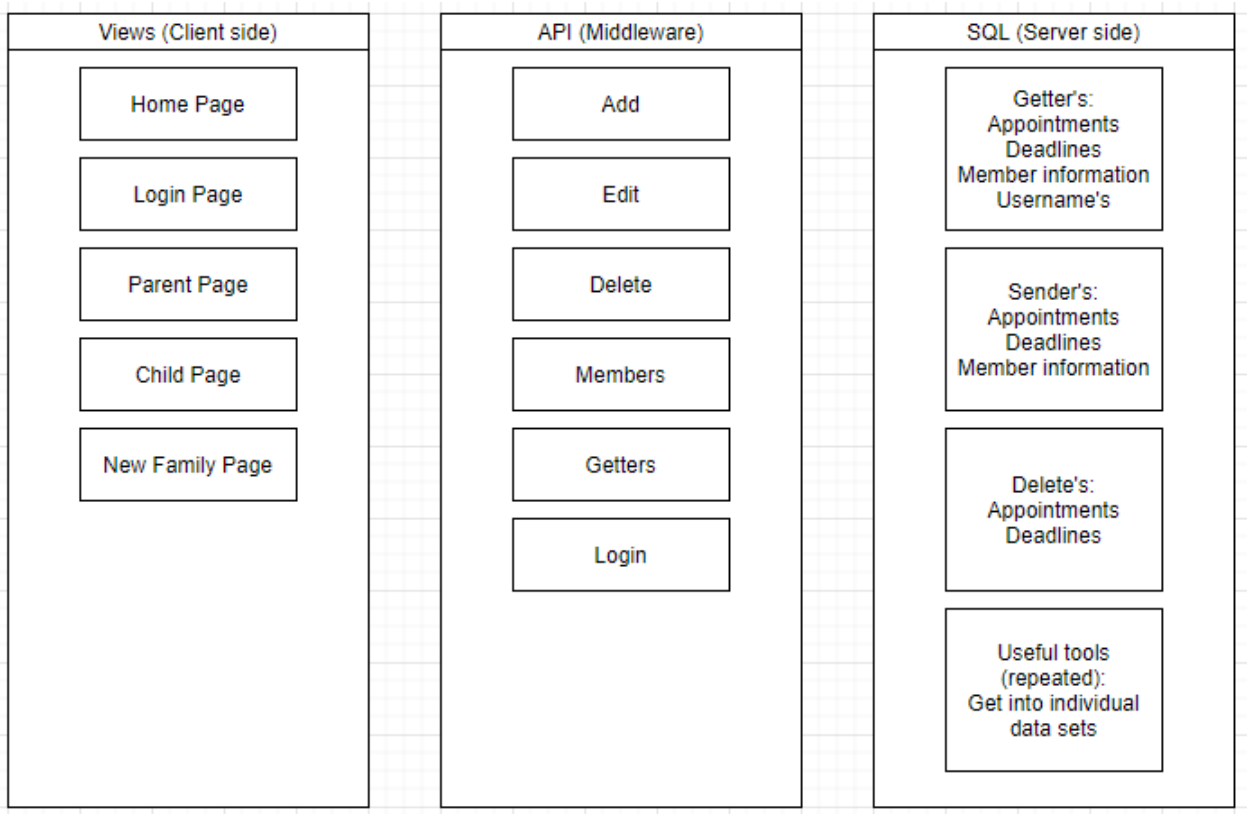


UML Diagrams

Initial UML diagram



Final UML Diagram

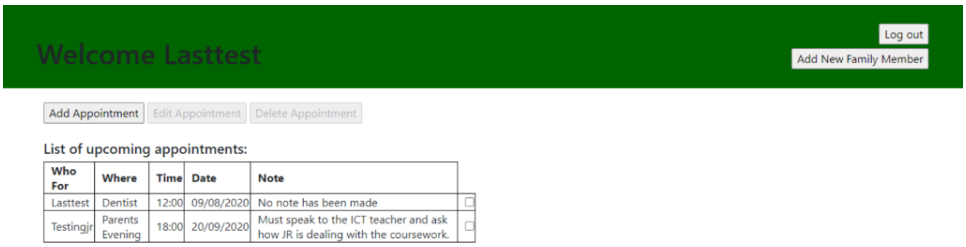


Settings

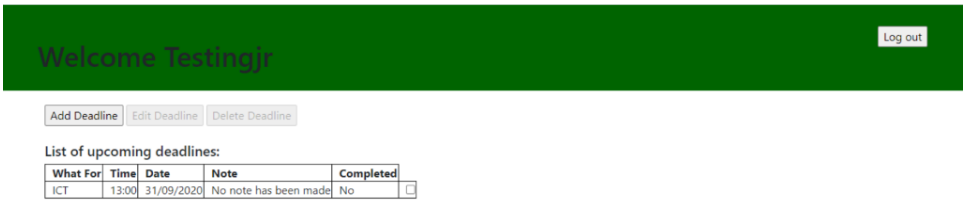
The webpage will run with basic settings active, working best with Google Chrome. When using the webpage to add, edit or delete an appointment or a deadline the user will have to log out and relog in to view changes, this is due to AJAX not being used. There are preset usernames in the system (see README.md).

Images

Parent Image

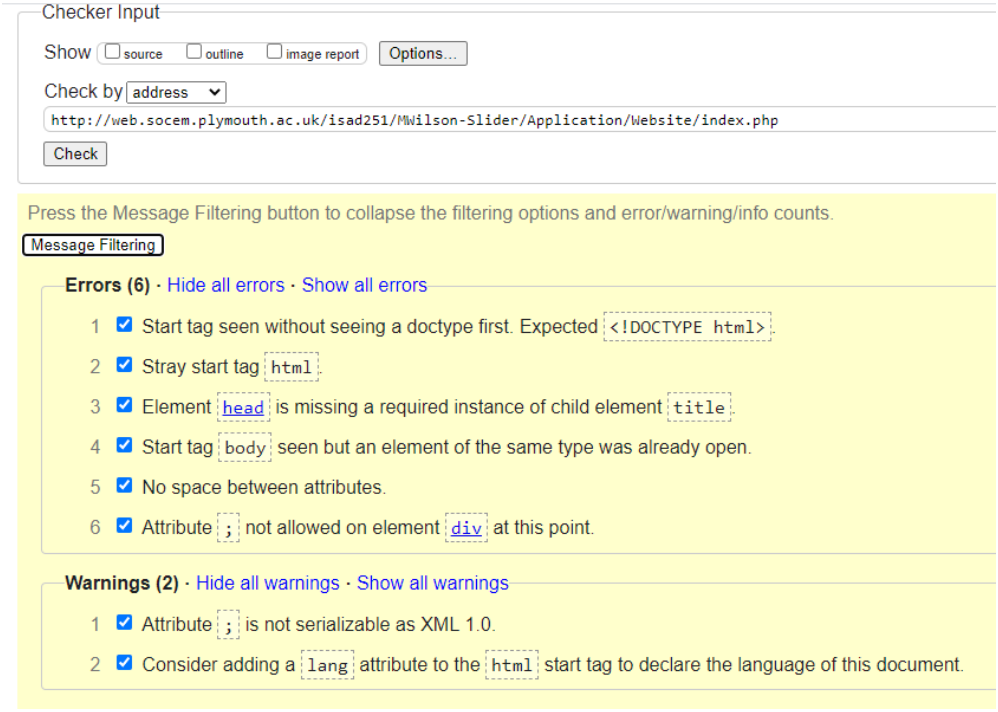


Child Image



Testing using WAI guidelines

Home page – index.php



Checker Input

Show source outline image report Options…

Check by address

http://web.socem.plymouth.ac.uk/isad251/MWilson-Slider/Application/Website/index.php

Check

Press the Message Filtering button to collapse the filtering options and error/warning/info counts.

Message Filtering

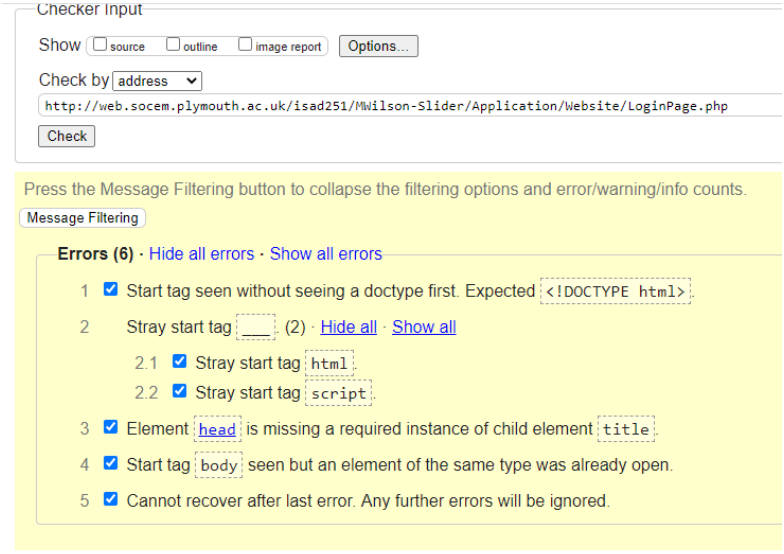
Errors (6) · Hide all errors · Show all errors

1. Start tag seen without seeing a doctype first. Expected `<!DOCTYPE html>`.
2. Stray start tag `html`.
3. Element `head` is missing a required instance of child element `title`.
4. Start tag `body` seen but an element of the same type was already open.
5. No space between attributes.
6. Attribute `;` not allowed on element `div` at this point.

Warnings (2) · Hide all warnings · Show all warnings

1. Attribute `;` is not serializable as XML 1.0.
2. Consider adding a `lang` attribute to the `html` start tag to declare the language of this document.

Login page – LoginPage.php



Checker Input

Show source outline image report Options…

Check by address

http://web.socem.plymouth.ac.uk/isad251/MWilson-Slider/Application/Website/LoginPage.php

Check

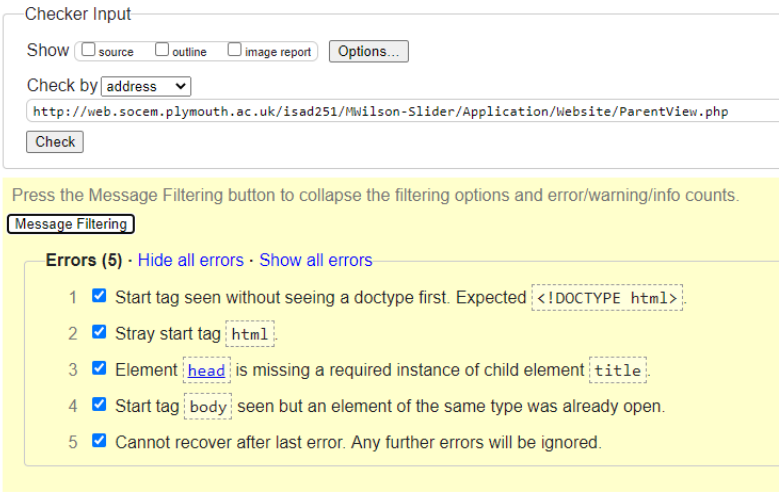
Press the Message Filtering button to collapse the filtering options and error/warning/info counts.

Message Filtering

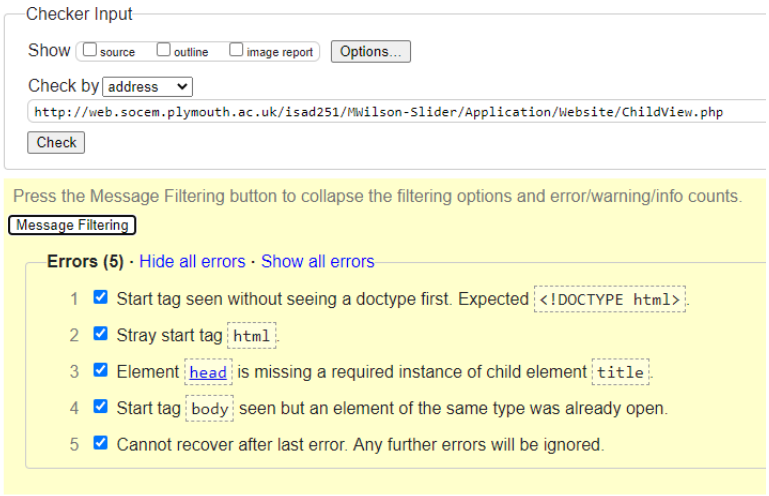
Errors (6) · Hide all errors · Show all errors

1. Start tag seen without seeing a doctype first. Expected `<!DOCTYPE html>`.
2. Stray start tag `___`. (2) · Hide all · Show all
   - 2.1 Stray start tag `html`.
   - 2.2 Stray start tag `script`.
3. Element `head` is missing a required instance of child element `title`.
4. Start tag `body` seen but an element of the same type was already open.
5. Cannot recover after last error. Any further errors will be ignored.

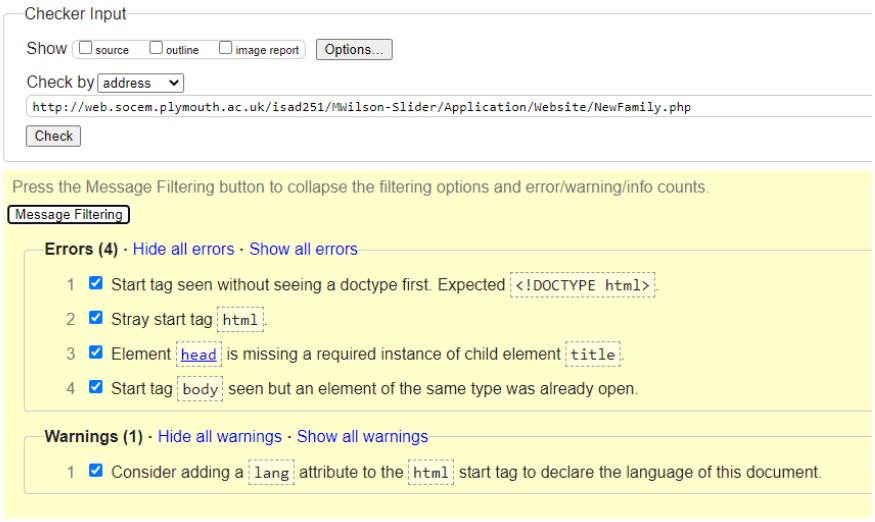
Parent page – ParentView.php



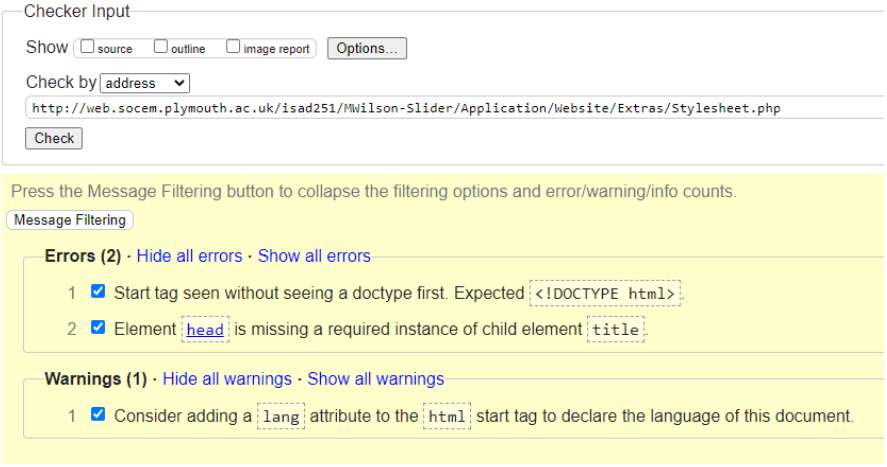
Child page – ChildView.php



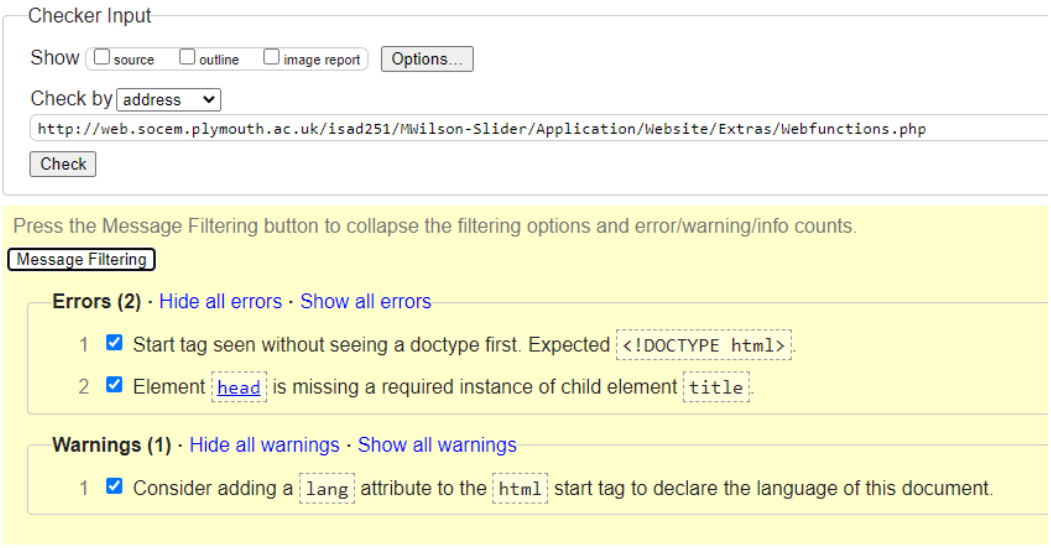
Add new family – NewFamily.php



Stylesheet



Webfunctions



Changes I would make and why

Home page

I would of like to develop this further and use this as more of an introduction by using outputs based on the Test user data.

New User page

This page could have been improved by having parents and children to make separate accounts instead of a parent having to create a child's login.

This page also has another issue that I would of like to fix, when details are rejected and edited so they do work, the alert box's appear multiple times often saying that the user exists then finally saying that the log in has been made.

Login page

This page works quite well, but I would of like to add a recover account option, by using an email-based password rest.

User pages

The parent and child pages both share the same problems and solutions.

When a user restarts the page, the website breaks as the session variables have been deleted, this can be fixed by using AJAX and I would have liked to produce a website that incorporates this.

When the user add's, edit's, or delete's an appointment/deadline then the p

Demo/Peer Review Feedback

Feedback given

1. Sort out file structure of the API.
2. Rename "Homepage.php" to "index.php", as the default of a server looks for "index.php" unless it has been programmed otherwise.
3. The login page uses GET instead of POST, this is a massive security risk as the login details are put into the URL, and the page currently refresh's so this is visible.
4. You'd solve the issue of the appointments not showing by using AJAX, but this is outside the scope of the system so don't fuss with that.
5. Instead of two separate Varchars for the time and date use DateTime, but that's something that again I wouldn't fuss about, as long you can argue why you have done it this way.
6. There is no specific requirements for how the page looks as long as it works as expected, but the select boxes, they would be huge if a user has been using the website for over a year, so maybe checkboxes might be the best way to deal with this.
7. The UML is not a proper UML but is more of a data flow diagram, I'd suggest that this get looked at, there is an example UML diagram on the DLE for you to look at.
8. And obviously the child page needs to be finished but remember I'm looking for quality more than quantity.

Changes made using feedback

1. The API has been put into separate folders, except the 3 that needed to be outside of their designated folders, due to the session.ID being different inside the folder.
2. "Homepage.php" was renamed to "index.php".
3. GET was changed to POST and the login page no longer refreshes, so immediate login is achieved.

4. I investigated AJAX briefly to get my head around it, but this was not implemented.
5. Due to time restraints this was not changed, I did change both of the Varchars into DateTime, and managed to split it and display the data on the website, however adding and editing the data was not quite right, I spent 2 days on this without much success.
6. The drop box was removed, and the checkboxes were added on the right side of the appointment/deadline
7. .
8. I finished the parent page fully, and implemented all the above before moving on to the child page ensuring that any repeated code was moved onto the "webfunctions.php" file as they can both be shared, the function to send form data however could not be moved into this file.

Suggested changes I would make if I had more time

I would of like to implement AJAX so that when the page gets refreshed or data gets edited then the website automatically shows this without breaking anything.

I would like to have been able to figure out the DateTime submission as that was the only part that I could not figure out, and in doing this implement the code for it.

When I added the checkboxes and made it so the buttons to edit and delete become enabled only when a checkbox has been checked, I encountered an issue, if one checkbox is checked then the buttons are enabled, but if another checkbox is check then the buttons become disabled. This can be redone to fix this issue, but I would have liked to solve this issue or added a troubleshooting page/pop up.

Finally, I would have liked to figure out how to pass the id of a session to the files in the API or investigated this further to develop a way to allow the same session to be picked up.

Links

GitHub repository

https://github.com/DigSwine/ISAD251-Resit

Application

https://github.com/DigSwine/ISAD251-Resit/tree/master/Application

Documentation

https://github.com/DigSwine/ISAD251-Resit/tree/master/Documentation

SQL

https://github.com/DigSwine/ISAD251-Resit/tree/master/SQL

Hosted web pages

Webpage URL

http://web.socem.plymouth.ac.uk/isad251/MWilson-Slider/Application/Website/index.php

Server

\\CENT-5-534.uopnet.plymouth.ac.uk\ISAD251\MWilson-Slider

YouTube Video

https://youtu.be/m8K-2s13D3Y