WiserPause

CFG Full Stack Group Project

Autumn 2023

Tania Alessi, Selina Lehane, Ruth McGowan, Caitlin Oddy

Contents:

INTRODUCTION:

Thank you for visiting WiserPause. The following report documents how we created a tool for people to test their knowledge of the peri/menopause and helps them learn more too.

We are four women working together as part of the Full Stack Specialisation course. None of us have currently undergone the menopause but all felt it was something we ought to know more about, and that that was likely not to be a unique experience.

BACKGROUND:

Problem we're addressing:

A study in Post Reproductive Health in 2021 found that 49% of participants did not feel informed about the menopause and more than 50% got their information from sources other than health professionals and official websites (51.1% and 50.5% respectively). 33.1% turned to social media and 49.8% to friends for advice [1].

Clearly there is a need for greater awareness about the peri/menopause, not just by those experiencing it themselves but for family and friends and for businesses wishing to better support their employees. With the pension age set to rise to 68, more and more women will be experiencing the menopause while working full time. Greater awareness from employers would help women feel comfortable sharing issues they are having at work relating to their symptoms and ensure employers know how best to support them.

What our app should do:

We are building an interactive quiz which will store user's answers and then at the end return a count of how many questions they answered correctly, along with all the questions listed with the answer they gave and the correct answer so they can learn. After this there is a button for the user to click to find out more information, including stories from women with lived experiences; a link to a reputable podcast and information from the NHS on the menopause.

We will leverage React to create our app and will utilise CSS to create a visually appealing UI for our quiz.

SPECIFICATIONS AND DESIGN:

Requirements technical and non-technical

Our app relies on React and react-router-dom, as well as useState to enable the user to progress from one page to another. '.map' is also used to map the answers users give and the correct ones to display on the results page.

As part of future development, we would wish to incorporate a database to store the age ranges of respondents and potentially other demographic data to gain an insight into whether knowledge is equally spread across ages and other demographic groups.

The following are the main non-technical requirements we considered:

- Useability: We ensured the app was easy and logical for a user to interact with, including featuring 'Page Not Found' page if the user enters an incorrect link which accessing the app.
- Reliable resources:As one of our primary goals is dispelling misinformation we made sure to use reputable sources when offering further information to users. This included trusted medical sources like the NHS website and a registered charity (The Menopause Charity) as a signpost for more informal support.
- Accessibility: It was important to us to make sure the app could be used by screen reader users correctly and that the reading ease score was appropriate for the intended age range.

Design and architecture

We began by creating some simple wireframes for the app, which can be viewed here:

Figma wireframes

We decided on our eventual colour scheme after carefully considering the anticipated age of our audience, e.g. mid 30's onwards. We showed a small selection of friends and family in that age range some different colour schemes and this one scored the highest with positive feedback around the traditional colours we picked out. We wanted to create a quiz that was fun yet more mature than an average article to more effectively target our intended audience,

We wanted to make sure it was also easy to use, with minimal clicks required to get through it, and that it was accessibility friendly.
There will be clear ways to move back and forward, and text and background will have an appropriate level of contrast to be readable.

A link to a screen recording of our app in action can be found here and in our project submission.

IMPLEMENTATION AND EXECUTION:

Agile approach

We worked in an agile way, running the project as a single sprint. We held scrums up to twice a week where we each discussed our progress, identified any blockers and worked through them as a group. We gave ourselves interim deadlines along the way to ensure we were on track and held each other to account.

We employed iterative techniques in that once we had a working prototype we considered how to improve it and redesigned the structure in a more efficient way.

Workflow tools used

Outside of our scrum meetings we kept in contact using a dedicated Slack workspace, with separate channels for front end design, project administration (such as meeting dates and deadlines), react support and report writing.

To enable collaborative working we created a GitHub repository that was shared amongst us to enable us to work on the project. In addition we used Google Suite to create meeting minutes, write the report and design the project presentation. The Google Drive was shared in common so we could all see the changes being made and add to them ourselves.

Division of tasks

After an initial SWOT analysis in our first meeting we initially divided the project work among ourselves as below. This was informed partially based on our existing knowledge and what knowledge would be most useful for us each to develop in our everyday roles.

Area	Team member
Project management (including completing documentation and project administration)	Caity and Ruth
Wireframe design	Selina
Back end logic	Tania
Front end design	Tania and Ruth
User Acceptance Testing UAT	All

Additional Testing	Caity and Selina
Presentation	All

Implementation challenges

We encountered a number of challenges in the implementation process, a few of which we have highlighted below:

1. Initially the development team encountered some issues working with react-router-dom which was our preferred package. Because of this we initially used state changes to allow the user to progress through pages and save the results. After support from instructors we were able to leverage react-router-dom later in the project but due to time constraints were not able to refactor the existing code.

2. We had intended to feature a navigation bar as part of our app and attempted to build this using bootstrap. However we found that including a navigation bar created issues with the formatting of the rest of our app. Having spent time on the design of the website we decided this was the priority, but if we were creating another app we would work with bootstrap from the beginning to enable us to integrate it with the CSS better.

3. Some of the team encountered issues working with GitHub on this project. Initially we found there were too many files and the project would not upload, making it difficult for us to collaborate. We solved that issue but the problem persisted for some. We worked together and shared best practices to enable us all to commit changes to the project and see those of colleagues.

4. We encountered several issues with unit testing. This was begun by one team member and then all other team members worked to troubleshoot, but ultimately unsuccessfully. It was decided at this point to reach out to our supervisor who kindly worked through the issues with us and showed us the way forward. Unfortunately we then encountered issues with a team member's IDE that was ultimately the source of the issues. Once this was resolved we worked as a group in a code review session to get the tests back in a working format and pushed to GitHub so all members had the code, safeguarding against any further issues with IDEs.

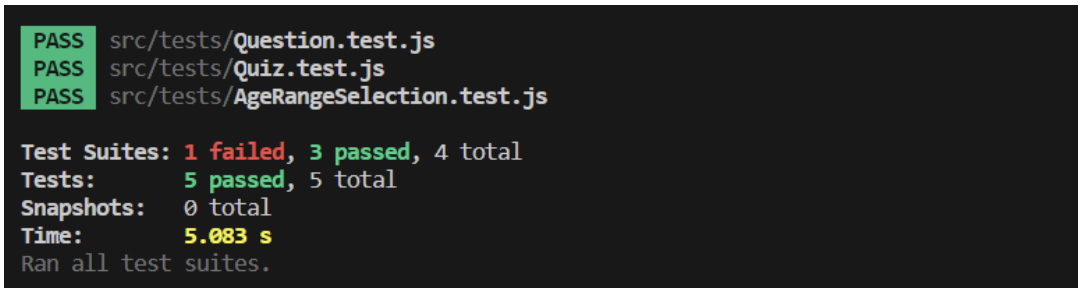
TESTING AND EVALUATION:

Unit Testing

We will create unit tests for the key elements of the app, namely: the age range selection, whether it can identify if a user selects a correct answer and if a new page renders when an answer is selected.

Our unit tests make use of *render* and *fireEvent* from React Testing Library as well as mock questions to test for successful renderings of pages and events occurring.

React testing library creates an additional test suite called App.test.js on initialising the tests that fails but this is not testing core functionality and our tests for that all run successfully, as evidenced below.



User Acceptance Testing (UAT):

All team members conducted initial UAT once we had a working app and provided feedback to the others about possible improvements. We also asked friends and family to go through it and give us user feedback. This led to the following changes:

- Users reported the quiz was too long so we reduced the number of questions to those we felt were the most important.
- The text on some buttons was not as clear as it could be to a user unfamiliar to the app and so we changed the text to provide greater clarity.

Accessibility testing

We felt it very important to make sure our app was accessible to people with health conditions or impairments and so we decided to conduct accessibility testing. A key aspect of this was ensuring our app worked for those that use screen readers. For this we researched and while JAWS is the most commonly used screen reader, it requires a licence and is not freely available so we chose to use NVDA to conduct our testing. [2]

After installing NVDA we navigated to the browser and checked every page, ensuring that it correctly reads out the type of each area of the page (button, heading, link etc) and that the page is fully navigable using keyboard shortcuts rather than the mouse.

This testing was successful, with the app having full compatibility with the screen reader. The only aspects that are inaccessible is the /aboutus page. This is due to issues adding a navigation bar, as discussed previously. This is an area of future development, however the core functionality of the app is accessible.

We also ran the text from certain sections of our app through a readability test checker, which gave us an average 'reading ease' score of around 47.1 out of 100 . We are content with this as our app is aimed at an older audience. [3]

CONCLUSION

Areas for expansion and improvement

We are very proud of the app we have created but of course development is never finished and there are some features we would like to work on in the future to add even more value for users.

One area would be, alongside the quiz and resources, develop a method for users to track their own peri/menopause symptoms. We discussed how we would initially wish to build this and agreed on a calendar with the ability to add an instance of a symptom from an existing list. This could then be further developed to allow a user to get a report of how many instances of a particular symptom occurred over a defined time period.

A feature we have mentioned before is a navigation bar using bootstrap. If we had time for further development we might try working with bootstrap from the outset instead of later in the project and then building the app on top. This would improve the accessibility of our app as well as the useability.

Final thoughts

As a team we had very little experience of React before this project - and some had never touched any Javascript before - so this experience has been a very good and challenging one. We have all developed our skills and hope to use this again in the future on our own projects.

Our thanks to our supervisor, Rehab Emad for her advice and support as we were progressing with our project and to the entire CFG team.

Sources:

1. https://www.ucl.ac.uk/news/2023/apr/nine-ten-women-were-never-educated-about-menopause
2. https://www.siteimprove.com/glossary/screen-reader-testing/
3. https://www.webfx.com/tools/read-able/smog-index/