Pause: Menopause Tracking App Project Documentation

- Cohort: Spring 2024, fullstack-1
- Group: Group 5
- Project title: Pause
- Group member:
    - Katie Kennedy
    - Emma Jourzac
    - Lingsay Wong
    - Emma Sheils

Contents

1. Introduction

Aims and Objectives of the Project

The primary aim of this project is to develop a full-stack web app called Pause. Pause is a Menopause Tracking App to empower women by providing a comprehensive tool to track and manage menopause symptoms. The app aims to fill a significant gap in the health-tech segment, offering personalised insights and data-driven support for women experiencing menopause. The specific objectives include:

- Enabling users to log in and monitor symptoms daily.
- Providing visualisations like calendar views for easy tracking.
- Ensuring a user-friendly interface with accessibility considerations.
- Leveraging modern web technologies to provide a seamless user experience.

Roadmap of the Report

This document will cover:

- Introduction
- Background
- Specifications and Design
- Implementation and Execution
- Testing and Evaluation
- Conclusion

2. Background

Menopause is a significant phase in a woman's life, where they stop having periods, often accompanied by a variety of physical and emotional symptoms. Menopause symptoms can occur due to certain treatment or surgeries such as for cancer treatment, and transgender men, non-binary people or intersex people may experience menopause (NHS Inform, 2024)[1]. According to the Menopause Charity[2] 90% of women experience menopausal symptoms, and 10% of women leave their jobs because of it.

Despite its impact, this area of health is underserved in the tech industry. Our Pause app is designed to address this gap by providing a dedicated platform for symptom tracking, which can lead to better management and understanding of the menopause. Users will be able to log their symptoms on a daily basis on the app. This data can then be leveraged to provide the user with insights into their symptom patterns.

[1] https://www.nhsinform.scot/menopause

[2] https://www.themenopausecharity.org

3. Specifications and Design

Requirements

Technical requirements

- **Frontend:** React.js, Redux, React Router
- **Backend:** Node.js, Express.js
- **Database:** MySQL
- **Testing:** Jest, Supertest (backend), React Testing Library (frontend)
- **Styling:** CSS, Bootstrap

Non- Technical requirements

- Users should be able to log in securely.
- The calendar interface should display symptoms logged each day.
- The system should support CRUD operations for symptoms.
- The design should be intuitive and user-friendly. Users should be able to log their symptoms quickly and effortlessly.

Architecture

The Pause application follows a client-server architecture and Create, Read, Update, and Delete (CRUD) operations, with a React.js-based frontend communicating with a Node.js backend via RESTful APIs. The backend interacts with a MySQL database to store and retrieve user data and symptom logs.

Design Principles

- **Intuitive interface:** The app's design prioritises simplicity and ease of use, ensuring a seamless experience for users. They can log their symptoms quickly. This is the main function of the app, and we expect users to do this on a daily basis. With automatic sign in, users will be able to log their daily symptoms in two clicks. One click to check the checkbox for their symptom, and one click to save. From the main dashboard users can access modals that open and close with clear buttons. They can log symptoms, edit logged symptoms, delete logged symptoms, and access insights to gain a bigger picture of their experience over time. Each of these things are accessed from the main dashboard with just one click.
- **Visual clarity and simplicity:** The design will be minimalist to make the interface appear uncluttered and user-friendly.

Design Choices

- **Colours:** Pink, red and white. Simple, feminine, fresh.

- **Typography:** Clean, simple, and consistent typography, with the use of bold for headings and regular weights for body text. This consistency aids in readability and information organisation.
- **Layout:** Grid-based layout.

App Pages

1. **Landing Page:** Introduction to the app with a short overview of its features. From this page users can either log into the app or register.
2. **User Registration:** Form to register new users. On successful completion users are redirected to the main dashboard.
3. **Login:** Form to authenticate existing users. On successful completion users are redirected to the main dashboard.
4. **Main dashboard:** A calendar-style interface so users can easily access their symptoms on different dates or log their symptoms for that day. Straight away, the modal appears prompting users to log their symptoms for the current day. Any date can be accessed from the calendar and symptoms will be displayed on the right. Users will have the option to edit or delete logged symptoms. This functionality can be accessed from the main dashboard easily. If no symptoms have been logged for that day, users can open the modal to enter and save their symptoms. The insights area will be at the foot of the app providing personalised and meaningful data of their symptoms over time e.g. their most frequent symptoms for the week/month, and how frequently they have been logged. Information and support on how to manage symptoms can also be accessed from here. The insights will appear as individual components that users can click on to open as a modal.

4. Implementation and Execution

Development Approach and Team Member Roles

Our team will use GitHub for version control and collaboration. We followed agile methodologies using Jira, Slack and daily standup to track our progress. We worked in short sprints to ensure continuous progress and collaboration. We communicated on a daily basis via Slack and met most days to review code, merge pull requests and decide on the next backlog of tasks. We realised early that we had different expectations, working styles and pace. By communicating we were able to adapt and work together effectively as a team. Our workload distribution is as follows:

Emma Sheils: Login styling page
Katie Kennedy: App concepts, Figma wireframes, frontend React components, user interface styling, design and development, user testing.

Emma Jourzac: App concepts, design and draft wireframes, backend (API development and database setup) and frontend design and development, user testing.
Lingsay Wong: App concepts, Miro design and agile project management setup, Github setup and documentation, frontend and backend development, unit and user testing.

Tools and Libraries

React.js: JavaScript library for building user interfaces
Redux: State management library for React applications
React Router: Routing library for React applications
React Modal: Accessible modal dialog component
Bootstrap: CSS framework for styling and responsive design
Axios: A promise-based HTTP Client for node.js and the browser
React Calendar: For calendar functionalities
Node.js: JavaScript runtime environment
Express.js: Web application framework for Node.js
Cors: A Connect/Express middleware that can be used to enable cross origin resource sharing
Dotenv: To store configuration in the environment separate from code
MySQL: Relational database management system
Jest: Testing framework for JavaScript
Supertest: HTTP assertion library for testing Node.js applications
React Testing Library: Utilities for testing React components

Implementation Process

The implementation process began with the design and wireframing phase, where the team defined the user interface and overall application flow. Following this, the backend API and database were set up, allowing for the development of the frontend components. The team worked together on the frontend, taking responsibility for different areas e.g. appearance, functionality and testing. Towards the end of the project, we looked to refactor our code where possible. We removed unnecessary code and files (apart from one component to be developed in future iterations), and made sure that files were well organised.

Achievements

- Creating a viable minimum product that met our design specifications.
- Implementing Redux to manage states globally, enhancing the functionality of the product.
- Overcoming obstacles such as losing team members.
- Overcoming differences through communication and a commitment to teamwork.

Implementation Challenges

The team faced challenges due to the loss of team members and scheduling conflicts, which impacted the agile approach and testing practices. Despite these obstacles, the core functionality of user registration, login, symptom logging and display was implemented. It was not possible to achieve all the functionality as planned in the design phase, and so the Insights section was put in the backlog for future iterations.

- **Team Dynamics:** The loss of team members and scheduling conflicts posed challenges in maintaining consistent progress and aligning development efforts. A lot of lessons were learned regarding working effectively as a team and we all feel grateful to have had the experience and work with each other.
- **Git Version Control**: As a learning experience, the team encountered difficulties in effectively collaborating and managing code changes using Git. This was overcome by team members supporting each other with meetings to walk through using Git, and sharing resources such as online tutorials.
- **Testing and Debugging:** With limited resources, comprehensive testing and debugging efforts were hindered, potentially leading to unresolved issues or bugs.

User Guide

The Pause application includes a comprehensive User Guide (USER_GUIDE.md) that provides step-by-step instructions for using the application. The User Guide covers the following topics:

5. Getting Started
6. Creating an Account
7. Logging In
8. Logging Symptoms
9. Viewing Logged Symptoms
10. Editing or Deleting Entries

The User Guide is designed to help users navigate the application smoothly and make the most of its features.

5. Testing and Evaluation

Testing Strategy

The Pause app follows a comprehensive testing strategy to ensure a seamless user experience and reliable functionality. Functional testing is conducted by implementing unit tests and integration tests to verify the correctness of individual components and their interactions. These tests are implemented using testing frameworks Jest and React Testing

Library, as well as Eslint to help us format code consistently and quality, allowing for efficient test execution and maintenance. The testing approach includes:

- **Backend Testing**: Unit tests for the backend API endpoints using Jest and Supertest.
- **Frontend Testing**: Unit tests for React components using Jest and React Testing Library.
- **Manual Testing**: Functional testing and user acceptance testing to ensure the application provides a seamless user experience.

We also conducted regular user testing to evaluate the app's usability and identify potential areas for improvement or modification. We had regular group discussion to gather valuable feedback to iterate our app's useability, design and functionality.

Backend Testing

The backend API endpoints are thoroughly tested using Jest, a JavaScript testing framework, and Supertest, a library for testing HTTP assertions. The *register.test.js* file demonstrates the testing of the user registration endpoint, covering scenarios such as successful registration, duplicate email handling, and server error handling.

Frontend Testing:

The frontend components are tested using Jest and React Testing Library, which provide utilities for rendering and interacting with React components in a test environment. The *ButtonOpenLogSymptoms.test.js* and *Header.test.js* files demonstrate the testing of a functional component that renders a button element. This test verifies that the onClick prop (a function passed as a prop) of the component is called when the button is clicked, using the fireEvent utility from React Testing Library.

Manual Testing

Tests were performed for components that have interactions or event handlers to simulate those interactions and ensure the expected behaviour. The team conducted manual functional testing on the following components and an end-to-end flow (register → login → log symptom → view symptoms):

- User Registration
- Login
- Symptom Logging
- Symptom Display (Calendar and List Views)

Limited user testing was performed by gathering feedback from team members and a small group of potential users to evaluate the application's usability and identify areas for improvement.

Postman

API testing is an integral part of the testing strategy, ensuring the reliability and correctness of the application's communication with the backend services. The team utilises Postman, a popular API testing tool, to manually test the API endpoints. This includes verifying the expected responses for various HTTP methods (GET, POST) and validating the data formats and structures.

6. System Limitations & Future Iteration

The minimum viable product was achieved with the core functionality of logging and tracking menopause symptoms.

- **Data Visualisation:** The initial version of the application provides basic symptom display in the list view. Advanced data visualisation and analytics features are planned for future iterations. It was planned for the calendar to be colour coded based on how many symptoms had been logged for that date to provide a clear and instant insight into the user's experience, month by month.
- **Insights Area:** In future iterations, the insights area will be developed to leverage user data to give meaningful insights and ways to manage the menopausal experience. The current implementation includes a limited set of symptoms management tips. Expanding this with an additional section with more comprehensive and personalised content on resources, professional help and community forum for peer support is a future goal.
- **Logging out:** In future iterations this feature will be added.
- **JSON web tokens:** In future iterations this feature would be implemented to ensure secure authentication during login.
- **Accessibility Considerations:** While responsive design was implemented, further improvements to accessibility features are necessary to ensure the application is inclusive for users with diverse needs. For example making sure all html tags are semantic. Testing our app for accessibility. Developing our app to be accessible on different types of devices.

7. Conclusion

The Pause project aimed to create a user-friendly platform for individuals going through menopause to track, log their symptoms and access tips to manage symptoms. The Pause app project has laid a strong foundation for addressing a critical need in women's health technology. By providing a user-friendly platform for tracking menopause symptoms, the app empowers individuals to better understand and manage their health.

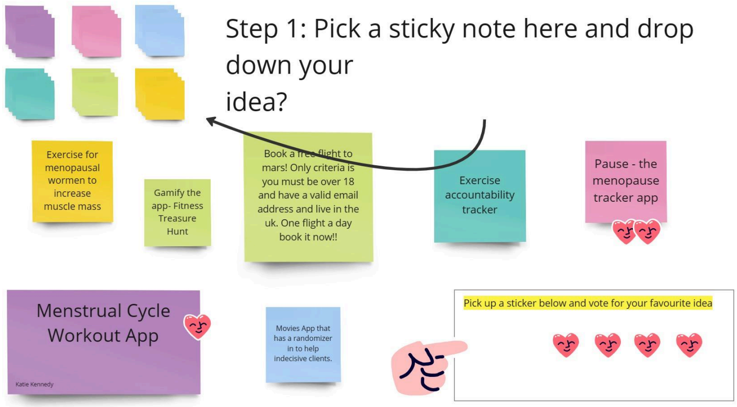
Despite facing challenges related to our team dynamics and resource constraints, the core functionality of user registration, login, symptom logging and display was successfully implemented. Throughout the project, the team gained valuable experience in agile development practices, collaborative coding, and adapting to changing circumstances. The lessons learned from this project will inform future iterations, which will focus on enhancing data visualisation, expanding educational resources, improving accessibility, and implementing comprehensive testing strategies.

The Pause application has the potential to positively impact the lives of individuals experiencing menopause by providing a centralised platform for symptom management and education. With continued development and user feedback, the application can evolve into a powerful tool that empowers women and improves their overall well-being during this transitional phase of life.

8. Appendix

Evidence of Development Approach

We used Miro in the idea phase to generate ideas and then vote as a team for which one we wanted to use.



Step 1: Pick a sticky note here and drop down your idea?

Exercise for menopausal wormen to increase muscle mass

Gamify the app- Fitness Treasure Hunt

Book a free flight to mars! Only criteria is you must be over 18 and have a valid email address and live in the uk. One flight a day book it now!!

Exercise accountability tracker

Pause - the menopause tracker app

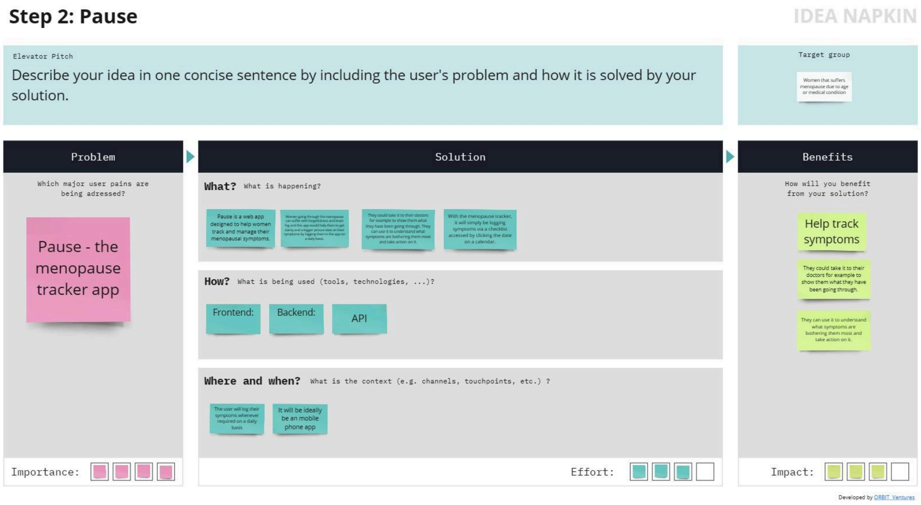
Menstrual Cycle Workout App

Katie Kennedy

Movies App that has a randomizer in to help indecisive clients.

Pick up a sticker below and vote for your favourite idea

We used Idea Napkin to expand on our initial idea



Step 2: Pause

IDEA NAPKIN

Elevator Pitch

Describe your idea in one concise sentence by including the user's problem and how it is solved by your solution.

Target group

Women that suffers menopause due to age or medical condition

Problem

Which major user pains are being adressed?

Pause - the menopause tracker app

Importance: ■ ■ ■ ■

Solution

What? What is happening?

Pause is a web app designed to help women track and manage their menopausal symptoms.

Women going through the menopause can suffer with forgetfulness and brain fog and this app would help them to get clarity and a bigger picture view on their symptoms by logging them in the app on a daily basis.

They could take it to their doctors for example to show them what they have been going through. They can use it to understand what symptoms are bothering them most and take action on it.

With the menopause tracker, it will simply be logging symptoms via a checklist accessed by clicking the date on a calendar.

How? What is being used (tools, technologies, ...)?

Frontend:

Backend:

API

Where and when? What is the context (e.g. channels, touchpoints, etc.) ?

The user will log their symptoms whenever required on a daily basis

It will be ideally be an mobile phone app

Effort: ■ ■ ■ □

Benefits

How will you benefit from your solution?

Help track symptoms

They could take it to their doctors for example to show them what they have been going through.

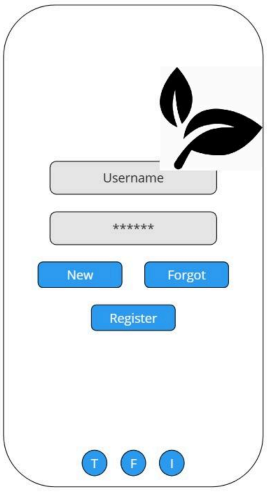
They can use it to understand what symptoms are bothering them most and take action on it.

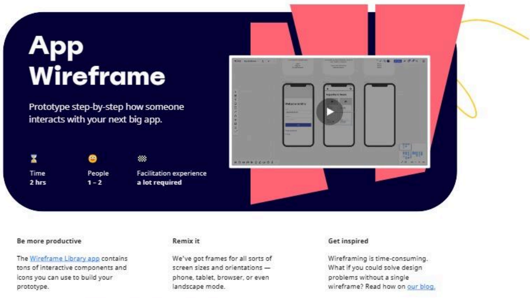
Impact: ■ ■ ■ □

Developed by ORBIT Ventures

We developed our designs starting off with simple wireframes and then making prototypes in Figma

Step 3: Sample Wireframes





App Wireframe

Prototype step-by-step how someone interacts with your next big app.

Time
2 hrs

People
1 – 2

Facilitation experience
a lot required

Be more productive

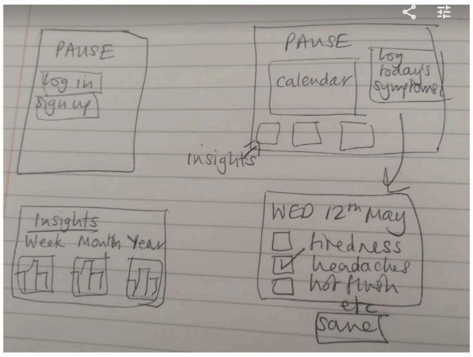
The Wireframe Library app contains tons of interactive components and icons you can use to build your prototype.

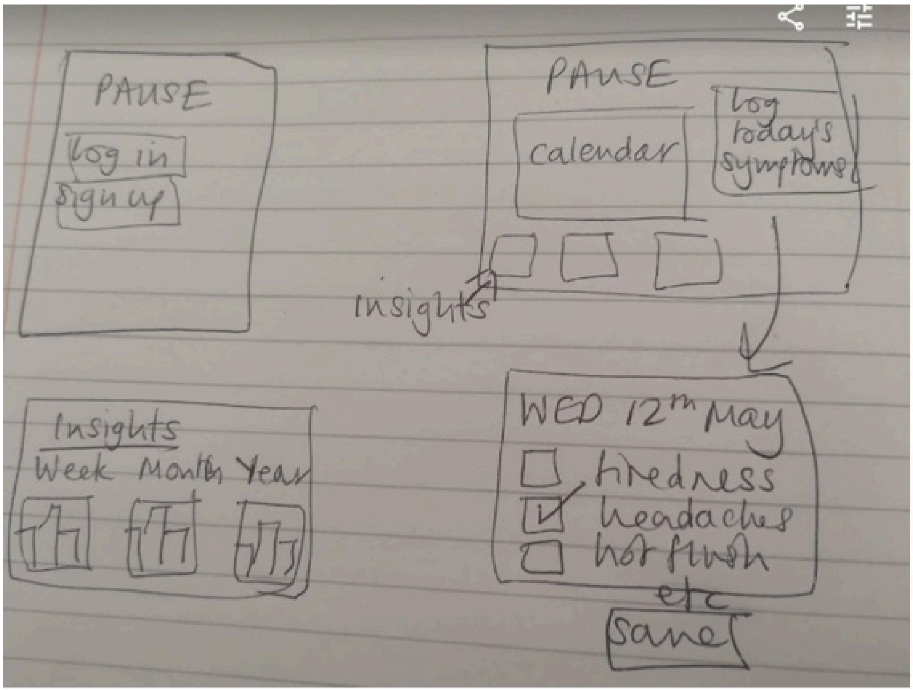
Remix it

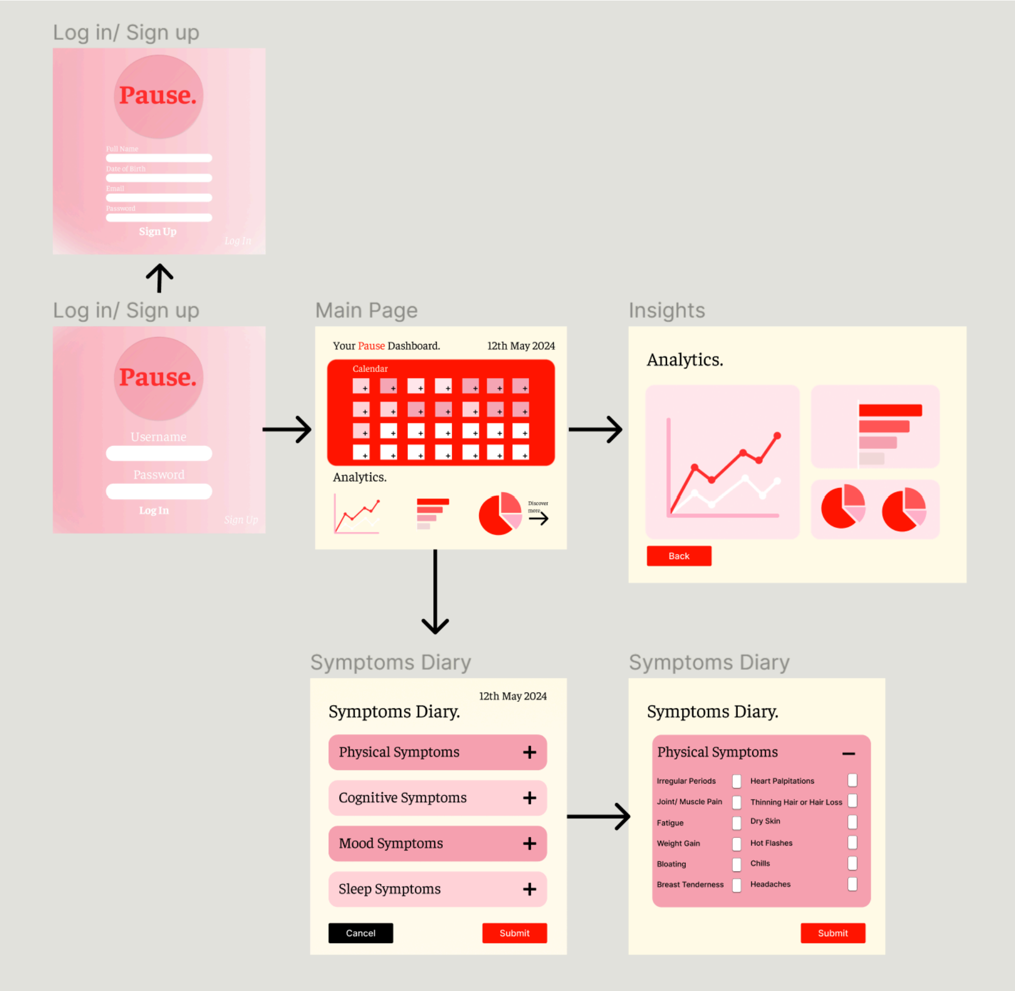
We've got frames for all sorts of screen sizes and orientations — phone, tablet, browser, or even landscape mode.

Get inspired

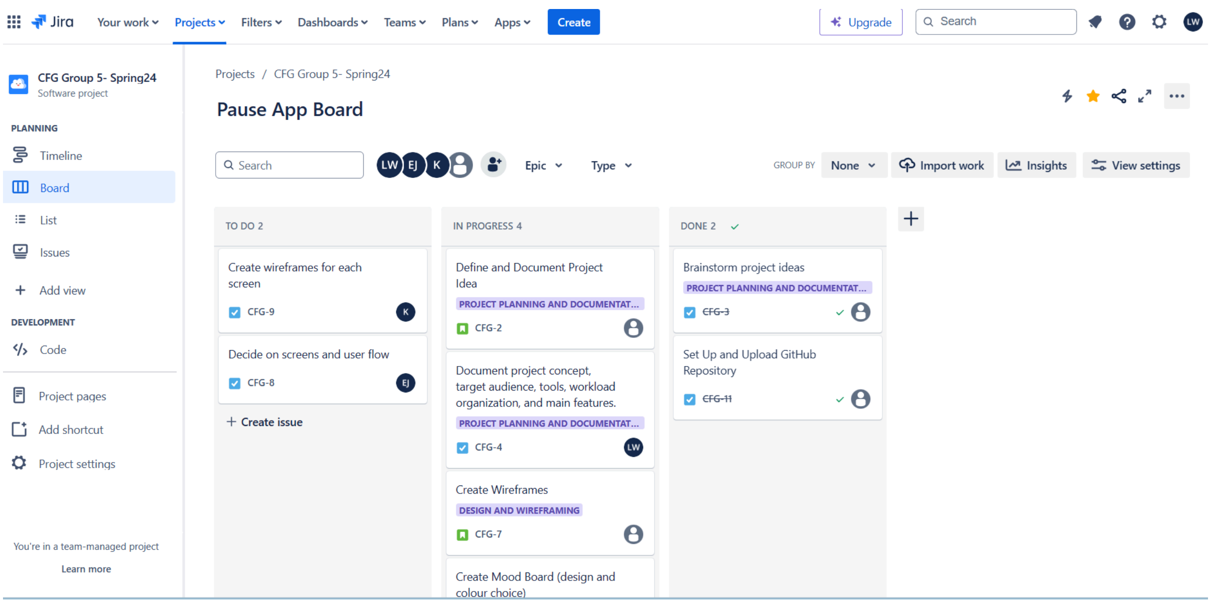
Wireframing is time-consuming. What if you could solve design problems without a single wireframe? Read how on our blog.




PAUSE
log in
Sign up
PAUSE
calendar
log today's symptoms
insights
Insights
Week Month Year
WED 12th May
tiredness
headaches
hot flush
etc
Save


Log in/ Sign up
Pause.
Full Name
Date of Birth
Email
Password
Sign Up
Log In
Log in/ Sign up
Pause.
Username
Password
Log In
Sign Up
Main Page
Your Pause Dashboard.
12th May 2024
Calendar
Analytics.
Discover more
Insights
Analytics.
Back
Symptoms Diary
12th May 2024
Symptoms Diary.
Physical Symptoms
Cognitive Symptoms
Mood Symptoms
Sleep Symptoms
Cancel
Submit
Symptoms Diary
Symptoms Diary.
Physical Symptoms
Irregular Periods
Joint/ Muscle Pain
Fatigue
Weight Gain
Bloating
Breast Tenderness
Heart Palpitations
Thinning Hair or Hair Loss
Dry Skin
Hot Flashes
Chills
Headaches
Submit

We tried to use Jira for agile project management but at the beginning of the project communication in the team made this difficult to implement.



Pause App Board

TO DO 2

- Create wireframes for each screen — CFG-9 — K
- Decide on screens and user flow — CFG-8 — EJ
- + Create issue

IN PROGRESS 4

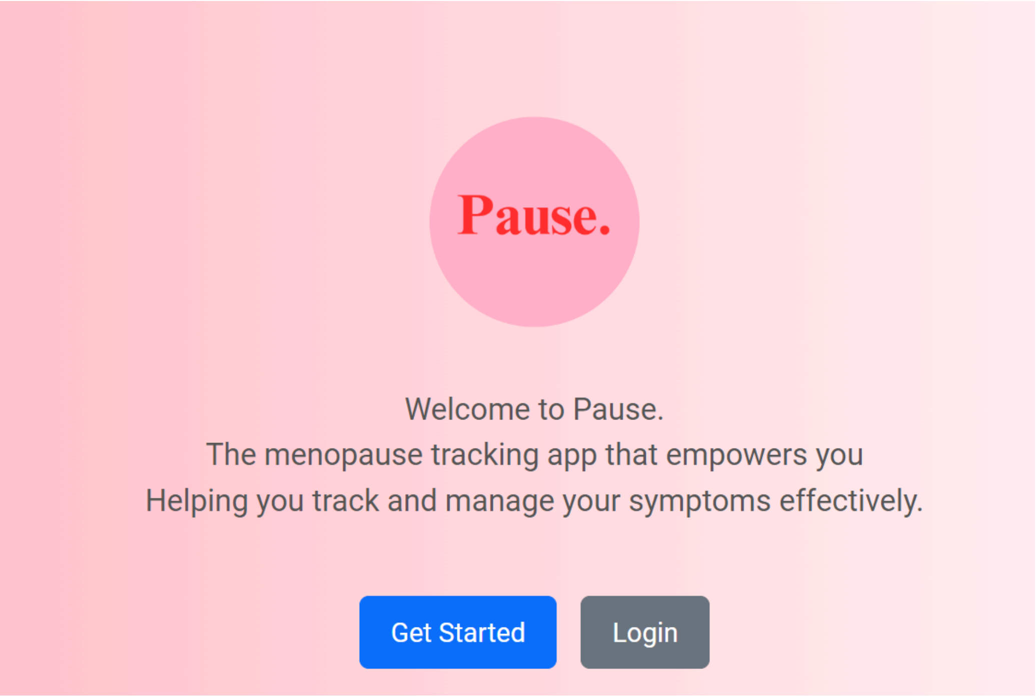
- Define and Document Project Idea — PROJECT PLANNING AND DOCUMENTAT... — CFG-2
- Document project concept, target audience, tools, workload organization, and main features. — PROJECT PLANNING AND DOCUMENTAT... — CFG-4 — LW
- Create Wireframes — DESIGN AND WIREFRAMING — CFG-7
- Create Mood Board (design and colour choice)

DONE 2 ✓

- Brainstorm project ideas — PROJECT PLANNING AND DOCUMENTAT... — ~~CFG-3~~ ✓
- Set Up and Upload GitHub Repository — ~~CFG-11~~ ✓

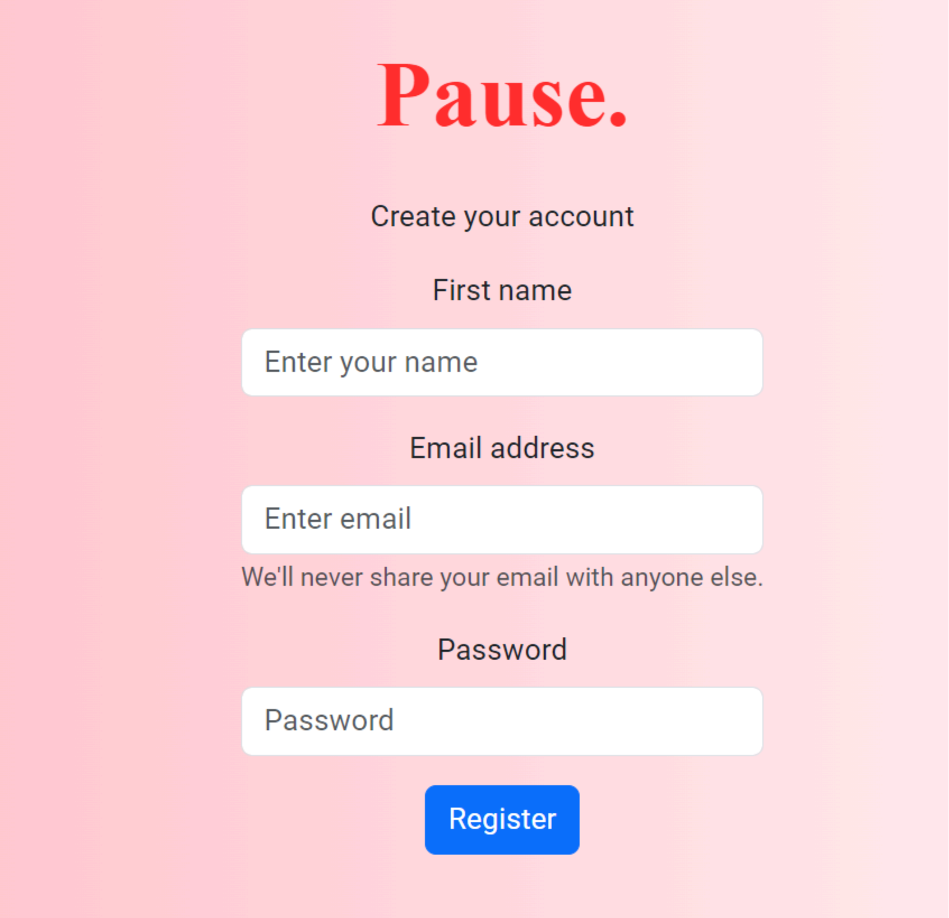
Images of Final Product

Landing page



Welcome to Pause.
The menopause tracking app that empowers you
Helping you track and manage your symptoms effectively.

Registration page



Pause.

Create your account

First name

Enter your name

Email address

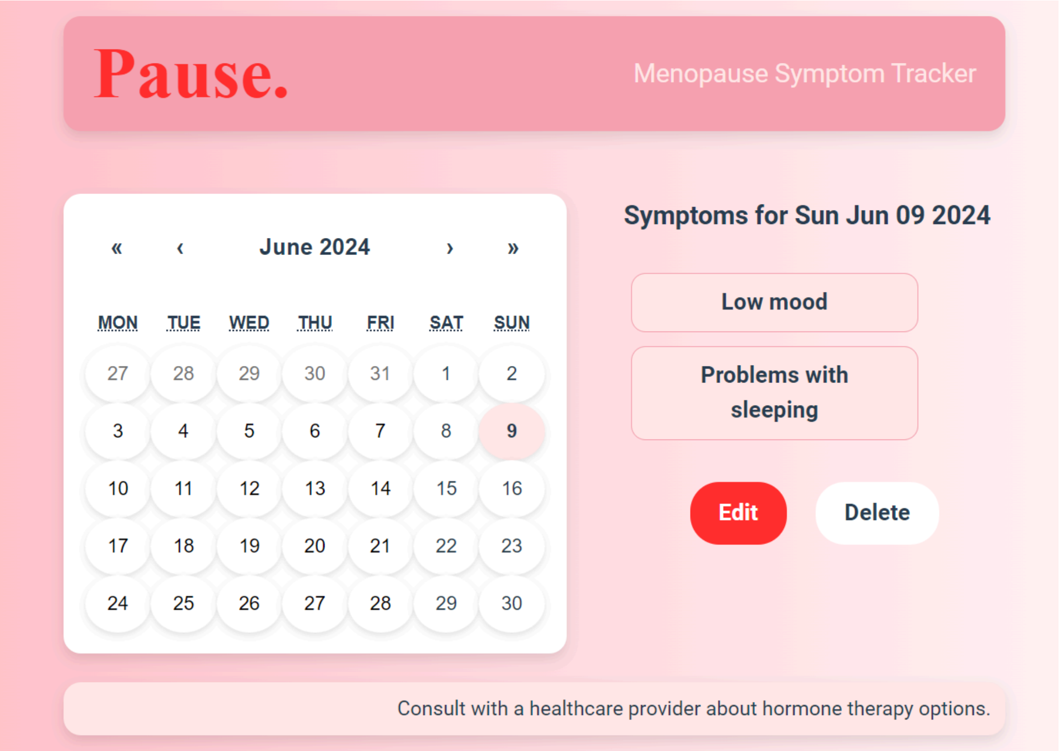
Enter email

We'll never share your email with anyone else.

Password

Password

Main dashboard



Pause.

Menopause Symptom Tracker

June 2024

MON	TUE	WED	THU	FRI	SAT	SUN
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

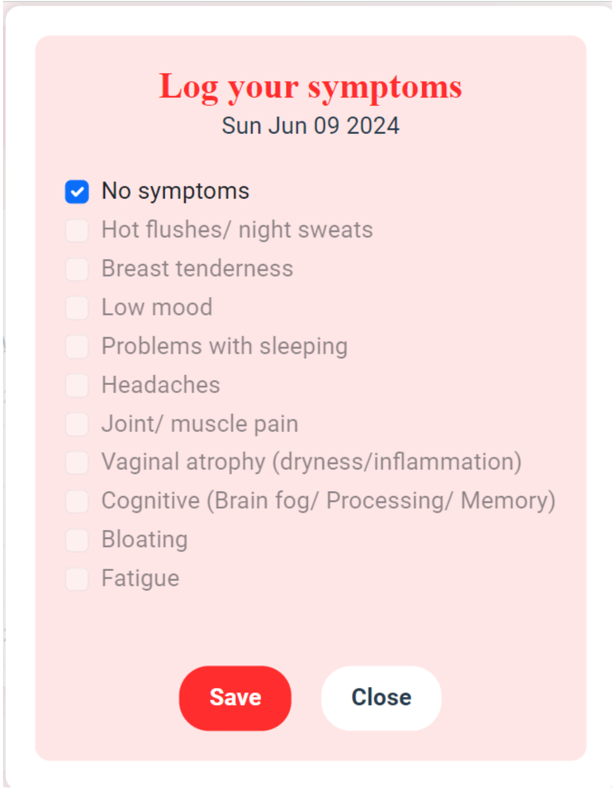
Symptoms for Sun Jun 09 2024

Low mood

Problems with sleeping

Consult with a healthcare provider about hormone therapy options.

Log symptoms modal



Log your symptoms

Sun Jun 09 2024

☑ No symptoms

□ Hot flushes/ night sweats

□ Breast tenderness

□ Low mood

□ Problems with sleeping

□ Headaches

□ Joint/ muscle pain

□ Vaginal atrophy (dryness/inflammation)

□ Cognitive (Brain fog/ Processing/ Memory)

□ Bloating

□ Fatigue