Chapter 1. Literature Review

The purpose of this literature review is to explore existing research and real-world platforms that inform the development of the Personal Accounting Climate Economics (P.A.C.E) website. The objective is to establish a foundation for building a user-focused platform that encourages all seventeen sustainable development goals [1].

**1.1 Best Practices in Sustainability Web Applications**

The P.A.C.E. System incorporates several best practices in sustainable web development. Currently, in website development, there are a few popular front-end technologies [2], including HTML/CSS, JavaScript [3], and frameworks like Angular [4] and Vue.js. Among these, React [5] was selected due to its efficiencyandflexibility in building interactiveuserinterfaces. React is particularly advantageous compared to Angular and Vue.js because it uses a virtualDOM(Document Object Model) [6], which allows for faster rendering of updates by minimizing the number of changes made to the actual webpage. This results in improvedperformance, especially for dynamic applications, like the P.A.C.E. System. Additionally, React’s component-based structure [7] enables reusability of code, reducing development time and making future updates more scalable and maintainable.

**Front-End:**

To support sustainability goals, the interface of the P.A.C.E. System was designed to be intentionally simplified. This decision was made to enhance user accessibility, ensuring that even users who are not technology-oriented can easily navigate the platform. By limiting the use of complex elements and animations, the design reduces the amount of processing required. A minimalistic desig also ensures that the platform is responsive across various devices, offering a better user experience without overloading mobile or desktop browsers. This approach not only makes the platform more user-friendly, but it also aligns with the goal of creating an energy-efficient website [8] that consumes fewer resources while maintaining functionality.

**BackEnd:**

On the backend, the system uses Supabase [9], which was selected over several other databases like MongoDB [10], MySQL [11], and Firebase [12] due to its unique combination of open-source nature. Unlike Firebase, which offers a closed system(which means there are some restrctions to access all features), Supabase is fully open-source [13], allowing for more customization and control over the database without being locked into a paid model. Compared to MongoDB, which is a NoSQL database, Supabase uses PostgreSQL, a relational database, making it easier to manage structured data with complex relationships and reliability in data handling.

Supabase also offers self-hosting capabilities [14], providing flexibility for future deployment and allowing the platform to be scaled based on growing user demand. addition to these advatages, Supabase enables optimized data handling with its built-in APIs [15], simplifying data exchange between the frontend and database.

**Design:**

In the design process, various sustainability-centered platforms were assessed to examine their navigation flows, UX strategies, and overall design. Examples include GoFundMe [16], with its highly regarded simple donation process, and Ecosia [17], a search engine focused on promoting eco-friendliness and a minimalistic approach. These platforms were assessed for their effectiveness and people-engagement strategies.

For example, the streamlined design and simple donation process at GoFundMe influenced P.A.C.E. to chose simplicity and minimalism to make sure that donors were able to make donations with ease and without any distractions. P.A.C.E. took over Ecosia’s concept of visual impact monitoring, where users are able to view the real-time effect of their contributions, hence driving engagement and openness.

Thes case studies influenced several P.A.C.E. design choices, including restricting animations to limit visual complexity and not having unnecessarily complicated layouts. Using them, the P.A.C.E. system is consistent with sustainable web design characteristics to make the platform accessible, efficient, and user-focused, with a priority placed on conserving resources.

**1.2. Case Studies:**

### **Case Study 1: React vs. Other Front-End Technologies** [18]

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **React** | **Angular** | **Vue.js** |
| **Virtual DOM** | **Yes** – Fast rendering with minimal updates. | **Yes**, but less efficient than React. | **No**, slower with frequent updates. |
| **Component-Based Structure** | **Yes** – Reusable and modular. | **Yes**, but less flexible. | **Yes**, but React has a stronger ecosystem. |
| **Performance with Dynamic Data** | **High** – Handles real-time updates efficiently. | **Moderate** – Can slow down with frequent updates. | **Moderate** – Lightweight but less efficient for complex apps. |
| **Ecosystem & Libraries** | **Extensive** – Large community, many libraries. | **Large**, but more built-in features. | **Growing**, but fewer libraries. |
| **Community Support** | **Very Large** – Frequent updates. | **Large**, but slower to adopt new features. | **Growing**, but smaller. |
| **Mobile App Development** | **Yes**, with **React Native** for mobile apps. | **Yes**, with **Ionic**, but not as seamless. | **Yes**, with **NativeScript**, but fewer resources. |
| **Resource Consumption** | **Low** – Efficient virtual DOM and small bundle size. | **High** – More features can increase consumption. | **Moderate** – Generally smaller but still uses more resources than React. |

When selecting the right front-end technology for the **P.A.C.E. System**, several options were considered, including **React**, **Angular**, and **Vue.js**. Each of these technologies offers unique advantages, but after careful analysis, **React** emerged as the best choice due to its **performance, flexibility**, and **developer-friendly features**. Below is a comparison of the three technologies based on key criteria: [18]

* **High Performance**: React’s **virtual DOM** [5]ensures **fast updates** and **minimal resource usage**, making it ideal for real-time, data-intensive applications like P.A.C.E.
* **Modular Structure**: React’s **component-based architecture** allows for **reusable code**, which simplifies maintenance and future updates.
* **Large Ecosystem**: With **extensive community support** and a wealth of libraries, React provides all the tools necessary for rapid development and scaling.
* **Low Resource Consumption**: React is highly **efficient**, meaning it consumes less **processing power**, aligning with the project’s goal of **minimizing energy usage**.

Given these benefits, React’s performance, scalability, and **developer-friendly features** made it the **best choice** for building the P.A.C.E. System.

**Case Study 2: Backend Technologies – Supabase vs. Firebase vs. MongoDB** [19]

When selecting the backend for the **P.A.C.E. system**, three databases were evaluated: **Supabase, Firebase**, and **MongoDB.** Below is a comparison of these technologies based on their key features:

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Supabase** | **Firebase** | **MongoDB** |
| **Database Type** | Relational (PostgreSQL) | NoSQL (Firestore) | NoSQL |
| **Self-Hosting** | **Yes** –Offers **self-hosting** capabilities. | **No** – Proprietary, cloud-hosted only. | **Yes** – Self-hosting is possible. |
| **Real-Time Data** | **Yes** – Built-in **real-time support**. | **Yes** – **Real-time data** with Firestore. | **Yes** – With **MongoDB Change Streams**. |
| **Ease of Use** | **High** – Simple API, automatic setup. | **High** – Easy setup with ready-to-use features. | **Moderate** – Requires additional tools and setup. |
| **Security** | **Built-in** – **Role-based access control** (RBAC) and **row-level security**. | **Built-in** – Simple security setup with **Firebase Authentication**. | **Requires external setup** – More flexible but needs manual security configuration. |
| **Data Structure** | **Structured** – Best for managing **complex relationships** (e.g., user data, donations). | **Flexible –** Bestfor **real-time, document-based data.** | **Flexible** – Best for **unstructured, large datasets**. |

For the P.A.C.E. system, a relational database was necessary to handle complex data relationships such as user profiles, donations, volunteer activities, and sustainability metrics. While Firebase and MongoDB offer advantages, such as real-time data handling and scalability, they do not support the level of data structuring required by P.A.C.E.

* Firebase is great for real-time applications but is NoSQL-based [12], which makes it harder to handle complex relationships like donations tied to user accounts and volunteer hours.
* MongoDB, also NoSQL, is suitable for large-scale, unstructured data but lacks the relational capabilities needed for managing structured data with multiple relationships.

On the other hand, Supabase uses PostgreSQL, a relational database that is well-suited for managing complex, structured data and supports real-time updates for tracking user actions like donations and volunteering. Thus, **Supabase** was chosen as the **most suitable option** for the P.A.C.E. system

**Case Study 3: Design Research Platforms – Best Practices from Sustainability Websites**

In the development of the P.A.C.E. System, several **sustainability-focused platforms** were reviewed to study their **design structures**, **navigation flows**, and **user experience strategies**. These platforms provided valuable insights into how **simplicity** and **clarity** in design can improve user engagement, especially in the context of sustainability initiatives.

**1. GoFundMe (Crowdfunding Platform)**

GoFundMe is widely recognized for its **clean layout** and **minimal steps** in the donation process, which was essential in ensuring ease of use for people unfamiliar with donation platforms.It features **real-time fundraising progress**, which directly aligns with P.A.C.E.'s aim to showcase user contributions and impacts in real-time.

**Influence on P.A.C.E**:

The **simple and direct user flow** of GoFundMe inspired the **P.A.C.E. donation process**, ensuring minimal barriers to entry and **enhanced user participation**. Like GoFundMe, P.A.C.E. uses **progress bars** to show donation and engagement progress, which motivates users to continue participating.

**2. Ecosia (Search Engine for Sustainability)**

Ecosia’s **live counter** showing how many trees have been planted helped create a **strong emotional connection** with users, encouraging repeat interactions. Ecosia keeps its interface clean with no unnecessary distractions, focusing users on their **goal of planting trees**.

**Influence on P.A.C.E**:

The **real-time tracking** of user actions in Ecosia was directly applied in P.A.C.E., where users can see how their **donations** and **volunteer activities** are contributing to specific sustainability targets.The **clean, minimalist design** of Ecosia inspired P.A.C.E. to limit excessive animations and unnecessary graphics, making the platform **more energy-efficient** and focused on the content.

These case studies showed that **simple, clear, and engaging design** encourages users to participate in sustainability. By using **minimalist design, real-time tracking,** and **easy-to-understand data**.

**1.3. Learnings:**

**Technology-Related Learnings**

**React**: I learned that **React’s virtual DOM** improves **performance** by updating only parts of the page, which is perfect for the **real-time tracking** in P.A.C.E. Its **component-based structure** also helps make the code more **modular** and **reusable**, which reduces development time and makes future updates easier.

**Supabase**: I chose **Supabase** over **Firebase** and **MongoDB** because it uses **PostgreSQL**, a **relational database** that’s better for handling **complex data relationships** like **user profiles** and **donations**. Supabase’s **real-time support** and **self-hosting** capabilities provide flexibility and control, while its **security features** align with P.A.C.E.’s need for **data protection**.

**Design Related Learings**

From GoFundMe, I learned the importance of a simple, efficient donation flow and how a minimalist design enhances user engagement. This inspired P.A.C.E. to adopt a simplified design with clear progress tracking for donations, ensuring ease of use and minimizing distractions.

From Ecosia, I learned how real-time impact tracking boosts user engagement by showing the tangible results of their actions. P.A.C.E. applied this by including real-time trackingfeatures for donations and volunteering, while also keeping the design clean and user-friendly.

**Learnings from Persona and Red Route Analysis:**

From the development of **user personas**, I learned how tailoring features for specific user groups, like **students** and **corporate users**, ensures a more **engaging experience**. The **student persona** guided the addition of **gamification** features, while the **corporate persona** led to the inclusion of **KPI tracking** and **performance dashboards**.

Through **red route analysis**, I identified the **critical user pathbs** that ensure key actions like **donating**, **tracking progress**, and **volunteering** are **simple and intuitive** for users and organizations.

# References:

|  |  |
| --- | --- |
| [1] | C. Wankel, "An examination of the divergent attention to the seventeen SDGs of the United Nations," 2020. |
| [2] | Naik and Arjun., "The Front-End Dilemma: How to Choose the Perfect Technology for your Application.," *Journal of Computer Science and Technology Studies ,* no. 6.1, pp. 211-216, 2024. |
| [3] | Dean and John. , "Web Programming with HTML, CSS, and JavaScript," 2025. |
| [4] | Geetha, "Interpretation and analysis of angular framework.," *International Conference on Power, Energy, Control and Transmission Systems (ICPECTS),* 2022. |
| [5] | Javeed and Arshad., "Performance optimization techniques for ReactJS.," *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT),* 2019. |
| [6] | Keshari and Pankaj, "Web Development Using ReactJS.," *5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N).,* 2023. |
| [7] | Pitt and Christopher, "React Components.," 2016. |
| [8] | Huma, Zillay and Junaid Muzaffar., "Sustainable Web Development: Creating Energy-Efficient and Eco-Friendly Applications.," *Pioneer Research Journal of Computing Science,* 2024. |
| [9] | "Supabase," [Online]. Available: https://supabase.com/. |
| [10] | "MongoDB," [Online]. Available: https://www.mongodb.com/. |
| [11] | "MySQL," [Online]. Available: https://www.mysql.com/. |
| [12] | "Firebase," [Online]. Available: https://firebase.google.com/. |
| [13] | Ayezabu and Amanuel Zewdie, "Supabase vs Firebase: Evaluation of performance and development of Progressive Web Apps.," 2022. |
| [14] | Molnar, David and Stuart E. Schechter, "Self Hosting vs. Cloud Hosting: Accounting for the Security Impact of Hosting in the Cloud," 2010. |
| [15] | Brajesh, "Introduction to APIs." API Management: An Architect's Guide to Developing and Managing APIs for Your Organization.," 2023. |
| [16] | "GoFundMe," [Online]. Available: https://www.gofundme.com/en-ie. |
| [17] | "Ecosia," [Online]. Available: https://www.ecosia.org/. |
| [18] | Cincović, Jelica and Marija Punt., "Comparison: Angular vs. React vs. Vue. Which framework is the best choice?.," 2020. |
| [19] | Kolesnikov O, Golovko G, Yastreba,V and Piaty, "LEVERAGING CLOUD TECHNOLOGIES AND SERVERLESS ARCHITECTURE FOR EFFICIENT WEB DEVELOPMENT: A CASE STUDY FROM REAL-WORLD APPLICATION," 2024. |
| [20] | Banks, Alex and Eve Porcello, " Learning React: modern patterns for developing React apps.," 2020. |
| [21] | Ayezabu and Amanuel Zewdie, "Supabase vs Firebase: Evaluation of performance and development of Progressive Web Apps.," 2022. |