CS640 – Computer Science Project Dissertation

P.A.C.E: A Website for Tracking & Contributing to Sustainable Development Goals.

Student Name: Vamsi Krishna Kancharapu

Student Number: 24252051

Year: 2024 – 2025

Course: MSc Computer Science (Software Engineering)



Department of Computer Science

Maynooth University, Maynooth

Co. Kildare, Ireland

Head of Department: Dr. Aiden Mooney

Supervisors: Dr. Peter Jean Paul, Dr. Linzi Ryan

LIST OF ACRONYMS…………………………………………………..

LIST OF TABLES……………………………………………………….

LIST OF FIGURES………………………………………………………

# TABLE OF CONTENTS

## ABSTRACT…………………………………………………………..

## 1. INTRODUCTION…………………………………………………….

1.1 Background and Motivation………………………………………

1.2 Problem Statement………………………………………………..

1.3 Objectives and Scope…………………………………………….

## 2. LITERATURE REVIEW……………………………………………..

2.1 Best Practices in Sustainability Web Applications (Theory)…….

2.2 Case Studies (Application) ………………………………………

2.3 Learnings…………………………………………………………

## 3. DEVELOPMENT PROCESS…………………………………….......

3.1 Agile Backlog and Scrum Process Overview…………………….

3.2 Prioritization and Sprint Planning………………………………..

3.3 Technology Stack: React, JavaScript, and Database Systems……

## 4. SYSTEM DESIGN…………………………………………………….

4.1 Overview of P.A.C.E System Architecture……………………….

4.2 Updated Website Layout and Workflow………………………….

## 5. SYSTEM FUNCTIONALITY…....……………………………………

5.1 User Functionality Requirements…………...................................

5.1.1 User KPI’s………………………………………………………

5.2 Corporate Functionality Requirements …………………...............

5.2.1 Corporate KPI’s…………………………………………………

## 6. TECHNICAL IMPLEMENTATION………………………………….

6.1 Front-End Development…………………………………………..

6.1.1 User Front-End………………………………………………….

6.1.2 Corporate Front-End…………………………………………….

6.2 Backend Development and Data Flow (Database Setup)…………

6.2.1 Dataflow Between Front-End & Back-End……………………..

6.3 Key Features and Functionalities………………………………….

## 7. TESTING AND QUALITY ASSURANCE……………………………

7.1 Bug Tracking and Resolution………………………………….......

7.2 User Feedback and Iterative Improvements……………………….

7.3 Measuring Quality Assurance with Key Performance Indicators…

## 8. RESEARCH LIMITATIONS & CHALLENGES……………………...

8.1 Challenges in Transaction Management and Data Retrieval………

## 9. CONCLUSION……………………………………………………........

## REFERENCES……………………………………………………………

## APPENDICES…………………………………………………………….

1 Appendix A: Low-Fidelity Designs………………………………….

2 Appendix B: Detailed User and Corporate Personas………………...

3 Appendix C: Backlog and Process Details…………………………...

# Chapter 1 Introduction

## 1.1 Background and Motivation

In 2015, the United Nations launched the Sustainable Development Goals (SDGs) [1] a set of 17 global goals intended to end poverty, reverse biodiversity loss and environmental degradation, and ensure prosperity by 2030. However, the success of these goals depends heavily on the active participation of individuals, communities, and organizations.

In order to achieve these effectively, they must be localized within countries [2] meaning, sub national entities like local governments must contribute in implementing efforts to achieve the SDGs. A number of scholars have looked into this phenomenon. As an example, Krantz and Gustafsson [3] conducted a study of localizing SDGs in municipalities within Sweden through an integrated approach. They recognised that localization of the SDGs is still a novel realm and that localization may need a significantly large systems view on sustainability. Lmassah and Mohieldin [4] focused on localizing frameworks of SDGs in the African cities, whereas Buyana et al [4]. talked about localizing SDGs in the African cities. They discovered that numerous frameworks do not regard the interconnections and intersections between SDGs and that the nature and historical specificity of forces influencing urbanisation are highly variable. It is due to this realization that a number of groups have organized ad-hoc initiatives and grassroots projects specifically aimed at mobilizing the individual citizen to actively participate in the success of the SDGs on a daily basis. An example is the 170 Series that was introduced by the Perception Change Project of the UN whereby it also offers 10 recommendations per goal of the 17 SDGs of what an individual can do to achieve these goals on an everyday basis [5] . The initiative offered an easy way of empowering all the people to participate in realization of the SDGs. However, individuals did not have an incentive to act and do so collectively, coherently, sustainably to provide a positive large-scale contribution to the SDGs.

Despite these examples, many individuals [6] have great difficulty engaging meaningfully with sustainability efforts. Frequently individuals cite cost, inconvenience and the fact that sustainability activities are disconnected from day-to-day activities as being some of the primary reasons for disengagement and inability to see individual impact. [7]

## 1.2 Problem Statement

A global initiative to address resource depletion, climate change, and the unsustainable nature of contemporary life is known as the Sustainable Development Goals (SDGs). Under the "no one left behind" theme, the Sustainable Development Goals (SDGs) comprise 169 targets and 17 global goals. Early efforts following the 2015 adoption of the SDGs produced positive trends. However, over the last three years, the COVID-19 pandemic, the conflict in Ukraine, and climate-related catastrophes have slowed or even reversed the SDG's progress. According to an analysis of 140 targets for which data is available, more than 30% have either experienced slow movement or regressed below the 2015 baseline1, and half of these targets are moderately or severely off track. No nation is expected to meet its targets by 2030, demonstrating the universal lack of progress with the SDGs. The past three years have demonstrated that efforts to accomplish the SDGs need to be resilient to unforeseen negative global phenomena and sustainable in the face of them.

The urgency to meet the 2030 SDGs is at its peak, yet widespread public engagement in sustainability practices remains low. Traditional communication strategies and policies often fail to resonate with end-users who feel disconnected from global agendas. This results in low participation, minimal behavioral reinforcement, and missed opportunities to scale community-based sustainability initiatives.

The main issue is that there isn't a smooth, all-encompassing solution that considers people's financial and behavioral characteristics and is data-driven to enable people to take action in their daily lives to meet the SDGs. Given the complexity of the SDGs, the solution should be multidisciplinary, adaptable, and simple for the average person to understand. The solution should be simple to implement, adaptable enough to consider the circumstances of various nations, and lead to widespread, coordinated action. In order to support top-down and bottom-up sustainable initiatives and support data-driven policy, it should enable the transparent collection of data. Lastly, the solution should be resilient to unforeseen negative global phenomena and sustainable in the face of them.

## 1.3 Objectives and Scope

The primary objective of this research is to develop a web-based sustainability platform the Personal Accounting Climate Economic Service (P.A.C.E.). This is a novel financial system empowering individual to take customized action to achieve the sustainable development goals on a personal level whilst gaining economic benefits and ensuring cohesive large scale positive impacts on social, environmental, and economic sustainability. Individuals (Clients) would sign into the system and be given a list of daily activities to complete. Each activity will have associated with it a significant economic benefit that the individual would gain if they completed the specific activity. All activities are meant to be simple yet contribute to the overall achievement of the SDGs. The system itself will consist of:

1. A master database for storing client data and used for client analytics,
2. An artificial intelligence system used to design and optimize activities for clients in such a manner as to maximize economic benefits for all clients involved and use data from all clients to determine the next best activity to issue to achieve the SDGs,
3. A reward system for tracking the loyalty scores of clients and a blacklist database for tracking clients who are performing poorly and other clients of interest who have yet to be contracted.

System that bridges the gap between awareness and action. The system aims to:

* Encourage participation in sustainability through interactive tools for individuals and organizations.
* Track and display contributions toward SDG-aligned activities such as donations, volunteering and recycling.
* Provide feedback and rewards using gamification, analytics, and a leaderboard system.
* Support organizations (e.g., universities) with dashboards for ESG goal setting, KPI tracking, and performance comparison.

The scope includes both individual users (with features like personal dashboards and Leaderboards) and corporate users (offering analytics dashboards). The platform also focuses on energy-efficient design, ensuring its own sustainability in terms of technical architecture.

# Chapter 2 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 sustainability habits and the active participation of individuals in achieving all the sustainable development goals [1].

## 2.1 User Experience Design Process and Best Practices

User Experience (UX) design process is critical when developing digital platforms, particularly those that would further sustainability. The process helps in ensuring that the product is finished is user centred, intuitive and working across all platforms. A normal UX design cycle includes usability testing, wireframing, task prioritisation and user research.

User Research and Requirements Gathering is the initial step in the process, during which designers attempt to learn about what its users need and anticipate of the platform. It could include online surveys, personal interviews or discussion in focus groups with individuals having different backgrounds. The aim is to extract general user requirements and areas of pain, which will serve to inform the subsequent design choices.

The second phase is UX Strategy and Task Prioritization and the most essential features or user journeys are scheduled. An example is the platforms that concentrate on enhancing navigation ease, lightening the cognitive load with neutral coloring, and focusing on such features as AI-generated suggestions that enable users to promptly make high-impact actions without being confused by many options.

After the development of the strategy, designers transition to Wireframing and Prototyping. Wireframes are simple visual maps that map the layout of a site and then are developed into a real site. Low-fidelity wireframes are simple sketched diagrams that may be tested by hand to explore initial layout hypotheses, whereas high-fidelity prototypes, typically made with tools such as Figma, are more realistically detailed and interactive, and they can be used to explore user paths more deeply.

Lastly, Usability Testing is done in order to provide validation to these designs. At this phase, the prototype will be presented to people that have been part of the previous research groups to be compared to their needs and expectations. Feedback is also obtained regarding the ability of users to achieve critical tasks without having to struggle (so-called red routes) and changes are being done to clarify things, minimize friction and enhance accessibility.

This is an iterative process of UX that is becoming a standard in modern web design and is especially helpful in platforms whose design is intended to encourage ongoing user interaction and streamline decision-making. Within the framework of sustainability-oriented uses, this methodology is used to create transparent, easy to access, and simple interfaces that reduce entry barriers. Following these principles of design will help platforms direct users to actions that are meaningful, make them inclusive of people with different backgrounds, and keep them interested with the platforms because of their ease of use and the ability to see their progress.

## 2.2 The SCRUM Process for software Development

The SCRUM is a popular approach to the contemporary software development. It pays special attention to iterative development, joint planning and constant improvement. This methodology is especially useful in dealing with complicated projects in which requirements could change over time.

SCRUM has three prominent roles:

* The Product Owner states the product vision and identifies the features which add the most value.
* The Scrum Master makes sure that the development process works well and eliminates all barriers and leads the team towards Scrum principles.
* The Development Team will work on developing the product in a series of small steps, with their main emphasise being on small deliverables in every cycle.

The methodology contains also a number of artifacts:

* Product Backlog, which includes all tasks, features and fixes that are prioritized according to the importance and urgency.
* Sprint Backlog A Sprint Backlog, a subset of the product backlog the team hopes to accomplish within a single sprint.
* A product or feature that is produced at the end of a sprint cycle and which is the working product or feature.

Key events in the SCRUM process include:

* Sprint Planning, at which the work of the next sprint is chosen and determined.
* Short check-ins ( Daily Scrum ( Stand-Up) meetings ) to look over progress and spot blockers.

A Sprint Review, in which the work done is presented and feedback is received.

* A Sprint Retrospective, during which a reflection on the sprint is made to find out the areas of improvement and the achievements.

The approach can be particularly effective in both individual and team-based developmental settings. As an illustration, academic projects tend to separate tasks into specific modules such as user-facing parts and organizational dashboards and then subdivide them into smaller deliverables, such as the ability to log in, track donations, or analytics pages. Scheduling weekly review meetings with academic supervisors can be used in the same way sprint reviews, keeping the development process in-track and maintaining feedback. Here, the adherence to the SCRUM process gave the developer the ability to plan and deliver the features in an incremental manner, keep the momentum of the project going and modify the workflow with the changing requirements as the project progresses.

## 2.3 Current Best Practices in Sustainability Web Applications

The two components of Web applications are the frontend and the backend which are developed with the objective of creating an application. The front (or the client-side) is the visual component of the site that the user interacts with, i.e. buttons, menus, and pages they view in their browser. The backend (or server-side) refers to the back-room system, which manages data, user accounts, processing requests and data base interactions.

In the frontend, the modern web development frequently employs a set of tools (HTML, CSS, and JavaScript frameworks such as React, Angular, or Vue.js). React is one of them and is commonly known to be strong in terms of performance and adaptability to develop speedy and reactive user interfaces [2]. It has a Virtual DOM (Document Object Model) which is a browser-friendly object that reacts only to elements in the webpage that require modification, which can be considered to be its major strength compared to what happens with the full webpage [3]. React is also structured in terms of components, i.e. groups of reusable code which enable the reduction of repetition and enhances maintainability [4].

The other aspect that is significant in the modern frontend development is the styling technique. CSS (Cascading Style sheets) is the standard practice of giving a complete customization to the appearance of a web page. Certain frameworks such as Tailwind CSS provide pre-built styles, although plain CSS is frequently favored by developers who want more control and a lower visual hierarchy, particularly in sustainability-conscious designs where minimalism is a design best practice to conserve energy [5]. On the back end, the developers need a system that will handle data storage, processing and the movement between the front end and the database. Firebase, MongoDB, MySQL and Supabase are some of the common technologies used as backends database. Firebase is also characterized by real-time functionality and is a proprietary platform, which does not allow much customization unless one buys a plan. MongoDB is a NoSQL database which is handy in unstructured data but not in complex data relationships.

Conversely, Supabase is developed on the basis of PostgreSQL, which is a relational database, that enables the organisation of data (which can be helpful when dealing with user profiles, donations, and records of volunteers). Supabase can also be customized completely and is open-source, and has self-hosting and real-time data sync features. It also has inbuilt APIs (Application Programming Interfaces) -that are considered as messengers between the backend and front end-whereby it is easy to transmit and receive information between the database and the user interface [6].

The tools and practices are highly used in platforms that pursue sustainable and inclusive design. An example is that sustainability-oriented websites tend to use fewer heavy animations, flashy design and avoidable code to conserve energy and make content more accessible, particularly when using mobile devices or on low-bandwidth networks [5].

2.4 Case Designs for Sustainability Web Applications

Sustainable web application design choices tend to focus on promoting user interaction and energy-efficient and simple designs. The analysis of existing platforms assists in determining the trends in the behavior of people, the flow, and motivation strategies.

Among the popular options is a crowdfunding web site, GoFundMe, which enables people and groups to crowd source their personal or social money [16]. The design of the platform is on a streamlined layout that requires few steps to donate. The interface is not overloaded with visual clutters and users can therefore focus more on the important actions. Such simplicity promotes increased participation of users in the use as confusion is minimized, and accessibility is enhanced.

One more platform implemented sustainability-oriented approach is Ecosia a web search engine generating its advertising income to plant trees [17]. Ecosia gives the user live counters and impact visualizations indicating how many trees have been planted depending on user action. Such real-time feedback can allow users to interpret the immediate impact of their interaction, potentially encouraging users to engage in the process again and be more conscious of the environment.

The design features these platforms have witnessed, are the best practice on web development that is sustainability-oriented:

* Minimalist interfaces cut on distractions and energy usage.
* Real-time impact displays and progress bars make one feel that they contribute and are responsible.
* Easy to navigate flows allow users to make actions fast, particularly with donation-based or purpose-driven websites.

These characteristics can be canvassed as a broadly recommended practice in user experience (UX) and sustainability literature as tactics to create meaningful, and accessible digital environments [5].

# Chapter 3 Development Process

In this work, the development process consisted of two development stages: the user experience (UX) design process and the Scrum Process. The User Experience (UX) Design Process is a structured approach to creating products (especially digital ones like websites) that are easy, enjoyable, and effective for users. The Scrum process on the other hand is a framework used in Agile software development. It helps teams work together to deliver products in short, iterative cycles called sprints. The User Experience (UX) Design Process was used to design the website while the Scrum process was used to develop the website. This chapter outlines what was done during each development stage to bring the website from concept to implementation.

## 3.1 User experience (UX) design process

## 3.1.1 Research

The development commenced with a research phase to get insights into users, their needs, behavior and some of the barriers that are likely to hinder them to engage in sustainability actions. They were done in two main ways:

1. Interview: Interview by the Head of the Green Campus Initiative in the university conducted on Wednesday, 20 November, 11 AM, at the Rhetoric Building, 2 nd floor. This was done to know challenges that organizations experience in ensuring sustainability and to know the service gap. The barriers to adoption, reward systems and improvements required were addressed as questions. Some of the main insights were the absence of user-friendly platforms and the necessity of feedback and recognition of sustainable actions.
2. Focus Group: Held in the group of 17 people (8 boys and 9 girls), aged 19-46. The members of the public and university students were the participants. The group talked about what would make them act sustainably, or not, how they would like to do so, and what would they require in a digital platform. The majority of respondents said they wanted to do something but were not able to find available and visible platforms to follow their will.

The main insights were that individuals would like to participate in sustainable activities but they require a centralized, straightforward, and entertaining platform that enables them to easily do and monitor the impact. These lessons were used as basis in other design process.

## 3.1.2 Define Problem

Based on the results of the interviews and the focus groups, the ultimate problem was defined as “The collective, publicly available platform through which individuals or organizations can engage in sustainability targets in a transparent and measurable manner does not exist.” In order to interpret and solve this problem:

Two personas were created:

• A User Persona of ordinary persons who are interested in becoming contributing to sustainability.

• A Corporate Persona that applies to organizations that strive to have sustainable strategy in place.

Based on these personas, Red Routes were designed on both user and corporate journeys to determine the most valuable actions such as signing up, donating, viewing impact and volunteering. These helped prioritize the core features in development.

## 3.1.3 Ideate

After the identification of the problems and goals, the ideation phase started:

* Approximately 15 low-fidelity wireframes were drawn on A4 pieces of paper as a way of sketching layout, buttons, and structure.
* Such sketches assisted in discussion of the possibilities of streamlining the user experience and pointing out the possible flaws in the interface early.
* During the development, the design was changed and modified- e.g. changing a traditional navbar into one Start Here button to simplify navigation.
* Real websites such as GoFundMe and Ecosia were additionally used as inspiration, particularly due to uncomplicated design, light color palette, and straightforward user navigation.

## 3.1.4 Prototype

After sketching:

* Figma was used to create high-fidelity wireframes demonstrating end-result color schemes and components, as well as interactiveness.
* Lucidchart was also used to create a functional website layout diagram on how each page would expand.
* These tools have served the purpose of making the design consistent, simple and in accordance with what the users would have expected of the platform.

## 3.1.5 Test

The prototypes of the early websites were tested among 6-7 MSc Computer Science students, who were asked to provide feedback in terms of usability. Key learnings from testing:

* There were too many buttons, and this confused the users hence simplification of navigation.
* Feedback provided contributed to bringing AI-based recommendations to the activity, which directs the users based on their performance.
* A voice activated emoji welcome option was provided to increase engagement.

This feedback allowed to streamline the design into a nicer more usable design.

## 3.1.6 Implement

After finalizing the design:

* It was constructed with a React.js (Frontend), the Node.js and Express (Middleware), and Supabase (Database).
* The system made sure that all the design, functionality and features were correctly coded.
* The mobile app is being developed by another member of the team, and the projects have a common backend and database, which guarantees the seamless user experience on both platforms.

## 3.2 SCRUM process

## 3.2.1 Product Backlog.

A complete Product Backlog was developed (in the appendices), and all the user and corporate stories, technical tasks and even anticipated bugs are listed. The tasks were ranked on the basis of importance and technical complexity. The product backlog made sure development of key elements such as user login, donation pages, dashboards and analytics were implemented at the beginning.

## 3.2.2 Sprint Planning

Planning of stunts was carried out on a regular basis and involved meetings with the App Researcher and the project supervisor.

* The platform was divided into two big modules, User Side and Organization Side.
* The first was the User Side, which was then further subdivided into such individual features as Donation, Volunteering, Advocate, Reuse-Reduce-Recycle, Protect Wildlife and Strengthen Body-Mind-Sprit.
* Meetings were taken with Mobile App Developer and the supervisor twice a week to organize feature alignment and to share the responsibilities.
* The data obtained in the app and the site should be kept in sync hence the design of the backend database tables to cater to both platforms.

## 3.2.3 Weekly Scrum Meetings and Sprint Reviews

* Sprints were reviewed every two weeks with the supervisor, where the work done was presented and feedback obtained.
* Questions like What should we do next? and Is this feature clear enough? and How can we make it better? were discussed in these sessions.
* Then the changes were implemented in the following sprint cycle.
* Although the mobile app was a separate project, there was always direct communication with the team, which made coordination, compatibility, and understanding.

## 3.3 Final Product Testing and Evaluation

After development:

* The system was tested on 5 MSc students who were asked to complete simple activities such as registering, making a donation, looking at dashboards and viewing recommended actions.
* Feedback focused on:
  + - Button visibility and layout clarity
    - The helpfulness of AI recommendations
    - Speed and responsiveness of pages
* With this, slight CSS adjustments were performed in order to enhance font sizes, spacing of buttons and page alignment.

The analytics functions were also utilized to track the interaction of users after testing and this will still be used to update the future.

# Chapter 4 System Design

## 4.1 Overview of P.A.C.E System Architecture

The Website is built in 3 simple parts/layers that contains the frontend, Backend and Database. 

**Frontend (Client Side):** Frontend is the part users see and use like buttons, colours, forms and pages etc. client side means when a user wants to use the platform or website whatever he sees, and he do from his side it comes under the client-side process. This frontend is developed using the React.js, responsible for the user interface.  Users can:

* View and complete sustainability actions
* Earn points and rewards
* Compete on leaderboards
* Control their profile and their past.

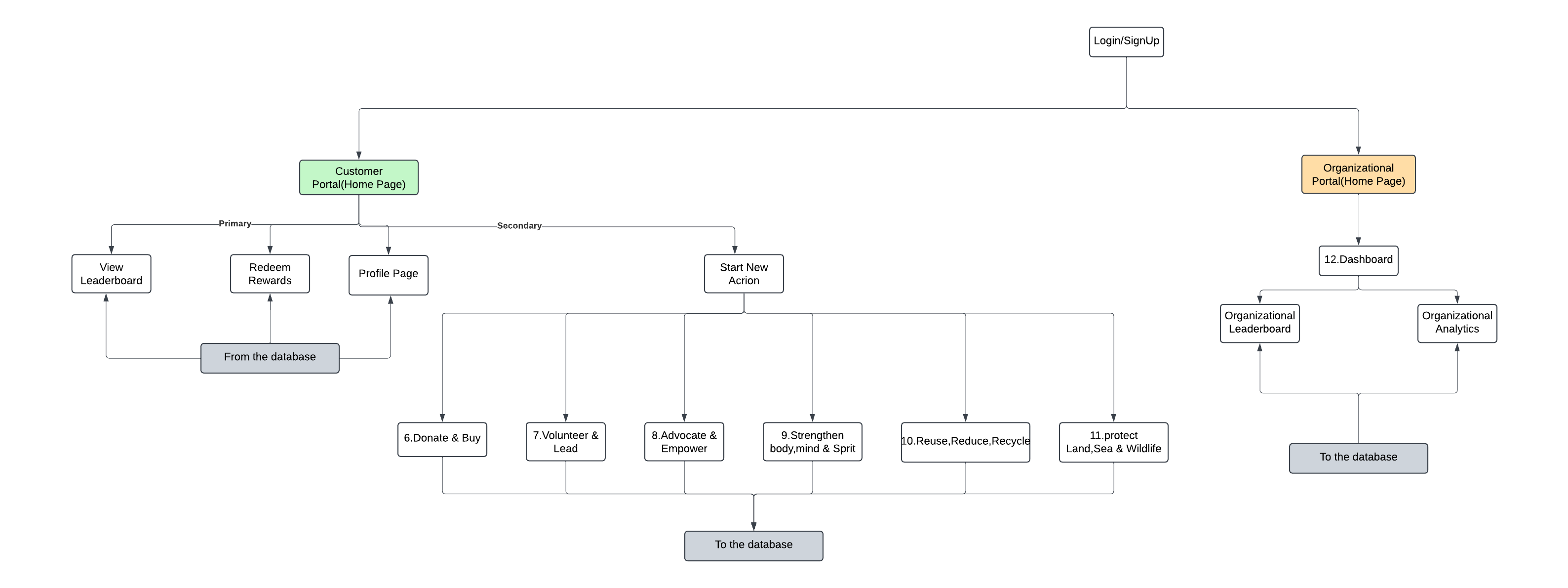
**Backend (Server Side):** It is the part that works in the background, handling tasks like saving data and checking login details. It is build using the Node.js and express.js. It is nothing but a mediator which connects the front end with the database and process requests from the front end to store in the database like checking login details, saving donations, and calculating points.  Authentication: A user is allowed to log in successfully with email and password. JWT (JSON Web Tokens) are provided with different access to regular users and admins.

* Database: Supabase (PostgreSQL) is used in the site and it is where structured data is stored in the form of users, actions, points and rewards.
* Live activity: It can be used to track and update the activity in real-time with WebSockets and Cloud Messaging.
* Analytics and Reports: Live reporting systems are used to generate SDG document.
* API Integration: APIs enable the mobile app and the site to have the same real-time data hence actions performed on one platform are reflected to the other.

**Database (Where data is stored):** The database is where all the information like donations, user account details, forms and page details will store safely in the form of tables. Every table contains table names, Rows, Columns. This website stores the data in Supabase, which helps manage the data properly and update things in real time.

**Security and Privacy:**

Security was a great point:

* All information is encrypted (while stored as well as while shared).
* Role-based access prevents unauthorized access.

## 4.2 Updated Website Layout and Workflow

The site is divided into two large parts, User Dashboard and Admin Dashboard

**User Dashboard Features:**

* Home Page: Introduces a welcome text with 6 Action circular buttons with good animation and with only 3 buttons-Start Here, View Leaderboard and Redeem Rewards to make user life simpler.
* Start Here: The user is given a list of all the available sustainability activities(2 activities per category/action).
* View Leaderboard: Displays rankings of top-performing users with Rank, Badge, Name, Email and points they earned in total. This page also have the Badge info which gives idea to users how they are consider as that particular badge holders.
* Rewards Page: This is a page that shows what one can get based on the number of activities they involved and the points they earned through it. Different rewards will be awarded for each activities.
* Profile: The user can view his/her profile and his/her achievements.

**Admin Dashboard Features:**

The Admin Dashboard will allow managers, team leads, and sustainability officers to track company-wide progress on SDG goals, view detailed reports, and monitor employee participation.

* Dashboard Overview: Shows SDG performance of the company at the company level in the form of a Pie Charts, Horizontal Bar Chart and mini trends (Graphs).
* Reports: Real-time performance reports and downloadable summaries in word format.
* User Management: Admins can also view all the information of the individual users.

Development Workflow:

The architecture was derived based on a SCRUM process:

* The entire project was split into user side and the corporate side.
* It was broken down into the smaller portions in each page and drawn sprint-by-sprint on every side.
* Weekly meetings with the professor and app developer were organized to keep track of the progress and rectify.
* Changes including: were guided by the many tests and feedback (user comments) by the users like sample students.
  + - * + Elimination of the conventional navigation bar.
        + To facilitate ease of navigation, an inclusion of one Start Here button.
        + A suggestion to use a smiley voice to introduce and AI-based activity recommendations.

Design Evolution

* The first designs were created in low-fidelity paper sketches (of the order of 15 pages).
* They have been built as high-fidelity mockups in Figma and layout tools like Lucid chart.
* Feedback was responded to in the process of test use and when a review of the design was being undertaken.

# Chapter 5 System Functionality

## 5.1 Authentication

Authentication refers to the process of verifying who a user is and subsequently granting him access to a system. A safe authentication system was developed in this project with Supabase Authentication and JWT (JSON Web Tokens) to manage the sessions. Users make an account using a username and a password. To provide more security, the password is hashed and then stored in the database that is, it is stored in a scrambled format which cannot be undone to save user credentials even when the database is retrieved. There are two types of users:

* The regular users are allowed to register freely with their email and password.
* Organizations are able to use a distinct Organization ID to access. The system administrators only issue this ID to known groups, and they are officially registered.

Although there was no implementation of traditional session mechanism (such as log out timer), after the user logs in, the system will remember the session of a user even between page loads, and therefore a user is not automatically logged out even when navigating or refreshing the page.

This method ensures that:

* Identities of users are authenticated.
* Sensitive data is protected.
* Only authorized users (e.g., verified organizations) can access certain features.

## 5.2 User Data Storage

Once a user is logged in successfully, the system will monitor the activity and progress of the user. This is carried out by saving user information in the back end with Supabase which is a secure online database. Each user has the following information saved:

* Name and email address
* Hashed password (to be secure)
* Points earned
* Time and date of registration
* Every sustainability activity done.
* Donation history
* Login timestamps

This information is applicable in two ways:

1. Personal use -The user is able to view their status, points, completed actions and donor counts on the dashboard.
2. Organization reporting - The same data is displayed (in summary form) in the organization administration dashboard. This aids organizations to know the number of people using it, the activities that are popular and people who are doing a good job.

Also, the leaderboard option runs on this stored information. It ranks all of them according to the number of total points and shows the ranking, which brings friendly competition, and makes users want to participate more. In general, this user data storage system will guarantee that:

* Every development is stored and presented properly.
* Organizations possess defined understanding of user engagement.
* Users are able to see and monitor their sustainability experience in real time.

## 5.3 Donation System

The donation system enables the user to contribute to sustainability causes using actual money. It is created with Stripe, a safe and popular online payment system. Here’s how it works:

**Integration with Stripe:** A payment form powered by Stripe allows users to donate with their card. It is a safe and easy procedure.

**What Data Is Stored:** A user can make a donation where the following information is stored safely in the database:

* Amount donated
* Time and date of donation
* User ID/email (to trace who is donating)

Note: Card information is not saved to provide privacy and security of users.

**User Profile Updates:** Once the user has donated, a profile displays the amount of donation. This will make users maintain their records on what they contribute and take more action.

**Impact on Points and Rewards:** Donations are also given more points than the normal activities. These points:

* Improve user rankings on the leaderboard
* Unlock better rewards
* Increase their sustainability badge tier

Such a system encourages users to repay, and, at the same time, provides organizations with useful data to measure engagement and impact via the admin dashboard.

## 5.4 Q-Learning AI Recommendation System

To enhance user engagement and personalize the experience, a **Q-learning-based AI recommendation system** was integrated into the P.A.C.E. platform. This smart system assists the users in being directed to most efficient sustainable steps depending on their past activity, making the participation simpler, more influential and more rewarding.

**What is Q-Learning?**

Q-learning represents a form of reinforcement learning algorithm, in which the system is reinforced to understand which actions are optimal to take under particular circumstances as time passes. It does not need a model of the environment but rather depends on experiences gained by means of interactions.

Every user action in the P.A.C.E. system is considered a potential choice (e.g., donating, recycling, volunteering). The system stores these actions and their outcomes in a Q-table, where:

* **Rows** represent user states or situations. Each action has 2 activities each(e.g. if in Action\_1 user already did an Activity\_1/Activity\_2 then the state flag is set to 1 )
* Columns represent available actions(e.g. see above- if Activity\_1/Activity\_2 is set as flag 1 then it is considered as that Action\_1 is completed, so flag 1 for Action\_1)
* The value in each cell in q-value denotes the number based on the below Q-algorithm formulae where it calculates based on most performed activity.

When the user interacts with the system, the Q-values are updated by the Bellman Equation:

Q(s,a)=Q(s,a)+α⋅[r+γ⋅maxQ(s′,a′)−Q(s,a)]

Where:

* s = current state,
* a = action taken,
* r = reward received,
* s′ = new state,
* alpha-α = learning rate,
* gamma-γ = discount factor.

**How the Q-Table Was Trained**

The process of the training was carried out in four significant steps:

Step 1: Initial Setup

* All possible states (e.g. low points, medium points, high points) and all possible actions (Recycle, Donate, Volunteer, etc.) were specified.
* A reward matrix was constructed -rewards on more significant action such as donation being assigned more rewards.

Step 2: Exploration

* The model started experimenting on the various actions in each state.
* It studied the extent to which action received a reward (points), under various circumstances.

Step 3: Learning

* Based on the reward obtained and the future reward that would be obtained, the Q-values (scores) were updated after every action using the Bellman Equation.
* The model was run through numerous cycles (referred to as episodes) and therefore had the opportunity to learn through several tries.

Step 4: Final Q-Table Generation

* A stable Q-table was produced after training. This table assists the system in determining what is the best step to take of a user according to his/her current point level.

It was implemented in Python and the resulting Q-table was saved in the Supabase database.

**Frontend Integration (Website UI)**

The proposal is present on the screen as a unique user interface:

* A bracket wraps around the most recommended activity.
* This bracket dynamically moves according to the way Q-table recommends it to.
* This visual signal is the guide of AI that helps the user pay attention to the most effective activity to get more points.

**Backend Integration**

* Since it is a background service, the Q-learning code is written in Python.
* It connects to Supabase database to access the activity logs and update the Q-table.
* The react frontend fetches the recommendation after its calculation and presents it through the bracket UI around the activity boxes.

**Impact and Benefits**

* It ensure that users do not need to second guess the next action to take- they have a clear call of action.
* The AI bases its suggestions on actual information, and this increases trust and makes people interested.
* It helps users **earn points faster**, and thus users can move up the leaderboard and access higher reward tiers much faster.This Q-learning solution achieves platform targets

# Chapter 6 Technical Implementation

## 6.1 Front-End Development

Front end is that portion of the web site that the user interacts with directly in the browser. It encompasses everything on the visual side such as buttons, navigation menu, text field, images and animation. The frontend of the P.A.C.E. is developed with React, a JavaScript library that is used to develop the fast and interactive user interfaces. React is made efficient and is fast because it relies on a concept known as Virtual DOM (Document Object Model) that only updates the modified sections of the webpage rather than loading the entire webpage.

### 6.1.1 User Front-End

The user interface (UI) is made as simple and as explicit as possible, and a central portion of the interface is a Start Here button, rather than a menu bar. The pages include:

* Home Page: Shows welcome message, user points and progress.
* Action Page: provides sustainable activities based on SDGs and ESGs.
* Leaderboard Page: Displays level of tiers and user rankings.
* Reward Page: Shows the number of points that are needed to redeem rewards.
* Notification Page: Displays action updates and user feedback.

Using APIs (Application Programming Interfaces) which are tools that enable the forward-end to call the backend to send or receive data are used to fetch dynamically a piece of data. The UI has a recommendation bracket that is AI powered. This animated bracket draws attention to the activity that has the largest Q-learning value to indicate what is expected of the users next.

### 6.1.2 Corporate Front-End

The Corporate Front-End is a dedicated web interface that specifically allows administrators, organizations, and sustainability managers to administer, control and assess the sustainability engagement of all the participants on the platform.

This is in contrast with the user interface, which offers data-oriented graphic representations and control capabilities to trace impact, adjust rewards, delegate duties, and produce automated reports.

**Dashboard Overview Page**

The dashboard is the hub of the entire platform providing admins with a bird-eye view of the platform activity. When the admin logs in, he/she can observe:

* No. of Participants- Number of people registered to the organization.
* No. of Completed Activities- Total Count of activities completed by all users
* Total Points Awarded - Sum of the points gained on all users.

**Category Overview (Donut Charts)**

* User by Category - Displays the distribution of user activities across the 6 categories:
  + - * 1. Donate & Buy
        2. Volunteer & Lead
        3. Advocate
        4. Body & Mind
        5. Reuse/Recycle
        6. Protect Wildlife
* Points by Category – Shows which category the total points were contributed.

**Mini Trends & Popular Activities**

* Mini Trends Graphs Line charts that indicate growth of activity in the previous two events that were tracked (e.g., Note Sharing, Reusable Mug). Two mini trends are displayed, one for categories and another for the activities.
* By Activity (Top 6)- A bar chart (a graphical presentation of data with rectangular bars) of the most done sustainability actions.

**Leaderboard**

* Shows the high achievers in the organization.
* Ranked in terms of overall points to promote competition.

**SDG Report Generator (powered by Ollama)**

The button of downloading SDG Report on the admin dashboard is one of the most innovative features. This button automatically transfers the data on the dashboard (activity completion, trends, points, etc.) to a local AI engine named Ollama, when it is clicked.

Ollama: Open-source AI language model engine, the same as ChatGPT, installed and executed on a local machine (offline). It works on the data without using cloud service or internet connection and is high-speed and entirely private. Ollama also processes received data to create a personalized SDG progress report in the Microsoft Word (.docx) format. Immediately the admin can download the report and present it or make use of it.

**Security**

* Role-based access control (RBAC) limits access to this dashboard. RBAC is a method that gives only the user with the role of administrator the ability to access or manipulate organizational data.
* Administration login will need a unique organization ID that is issued to only verified partners.

## 6.2 Backend Development and Data Flow (Database Setup)

The other side of the system that a user does not see is the backend. It contains the server, database and all the logic (such as computations, authentication and storing user actions).

The back-end is developed with the help of Node.js and Express. Node.js is a JavaScript runtime enabling the execution of JavaScript code in the server. Express.js is a flexible web application and API framework in Node.js. The entire information is saved safely in Supabase, which is an open-source application of authentication, database, and file storage. Supabase is a structured relational database (PostgreSQL).

The backend includes:

* Authentication: It is done by email and password. Rule-based validation is applied in Supabase which includes the use of the at sign(@) in an email address and at least 6 characters on a password.
* Role-based access: The users and organizations have different rules of logging in. Unique org ID is supplied in the form of organization login.
* Security: Supabase has JWT (JSON Web Tokens) to provide secure access and hashed passwords to secure user information.

### 6.2.1 Dataflow Between Front-End & Back-End

The sequence of data flow is the following:

1. User Action: This is when a user makes an action or sends in a donation.
2. API Call: API is used by the front end to send the data to the back end.
3. Database Update: The backend updates Supabase database.
4. Response: The backend reacts to the front end by transmitting updated data, to refresh UI elements.

A diagram of a software application

AI-generated content may be incorrect.Any user data is updated in real time to assist dashboards and leaderboards.

## 6.3 Key Features and Functionalities

The basic features used in the system are:

1. Authentication

* The users create accounts and log in by using email and password.
* Organizations require a distinct org ID to be able to enter the corporate portal.
* The passwords are hashed (encrypted).
* JWT tokens have secure sessions.

2. Data Storage

Supabase contains all user information such as:

* Personal details
* Points
* Activities completed
* Donation history

The data in these records are read by organization dashboards to display engagement statistics.

3. Donation System

* Stripe is a secure payment gateway that handles real-time transactions and is used to process donations.
* Donation gives the user additional points.
* Donation history (without card information) is stored in Supabase and are displayed in user profiles and organization dashboards.

4. AI-Based Q-Learning Recommendation

* Q-learning is a form of Reinforcement Learning algorithm. It does this by learning what the user does to suggest the next activity that is the most rewarding activity.
* The Q-table holds the state-action pair along with a value known as Q-value that grows with repetitive you are using that activity.
* Example: Q-value will increase with the use of reusable mug when the rewards are common.
* The bracket UI points out the action that is recommended based on the largest Q-value.
* Python Backend code computes the Q-values and then updates the database on-the-fly.

5. Audio Greeting

* Users are welcomed by a mini smiley emoji icon that plays an audio file when they are accessing the site.
* This enhances user experience with a sense of personalization and friendliness.

6. Real-Time Leaderboard & Analytics

* Leaderboard lists the users in order of points earned.
* The rewards are divided into levels according to the points.
* Admin dashboard includes live charts of user action, donation impact and SDG progress.

7. Shared Backend for Web & App

* The website shares the same Supabase database as the mobile app.
* Both platforms reflect user actions in both interfaces.

# Chapter 7 Testing and Quality Assurance

## 7.1 Bug Tracking and Resolution

## 7.2 User Feedback and Iterative Improvements

## 7.3 Measuring Quality Assurance with Key Performance Indicators