# Chapter 6

# 6.1 System Design

### 6.1.1 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):** This frontend consist of two dashboards that was is developed using the React.js. These dashboards are the student dashboard and the organization dashboard. With the student dashboard users can:

* View their total and redeemable points
* View the actions to be completed
* View user history of actions already completed
* Received advice from an AI avatar as to the next best action to be completed
* Make donations
* View the leaderboard showing top scoring users
* Receive assistance from an AI Avatar

With the organization dashboard the organization can:

* View aggregate performance of students across all six categories.
* Receive SDG report from OpenAI (Ollama Ai-Engine) as a downloadable pdf.
* View top performing student by category

**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 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 WebSocket’s and Cloud Messaging.
* Analytics and Reports: Live reporting systems are used to generate SDG documents for organization user.
* 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 are store securely in the form of tables. This website stores the data in Supabase, which helps manage and update data in real time.

### 6.1.2 Website Layout and Workflow

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

**User Dashboard Features**

A screenshot of a computer

AI-generated content may be incorrect.

TALK about features

**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.

A screenshot of a computer

AI-generated content may be incorrect.

* 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.

Talk about features

# 6.2 Special System Functionality Features for Users

The website has five unique functional features that are important to encourage user interaction. These are:

1. Authentication for security and privacy

2. User Data Storage

3. Donation System

4. Q-Learning AI Recommendation System

5. AI Avatar

### 6.2.1 Authentication for security and privacy

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 can use a distinct Organization ID to access site. The system administrators only issue this ID to known groups that 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.

### 6.2.2 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. 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.

### 6.2.3 Donation System

The donation system enables the user to contribute money to sustainability causes using Stripe a secure, popular online payment system that allows users to donate with their card.

**What Data Is Stored:** When users donate the following information is stored securely 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 more privacy and security for 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 make more donations, and, at the same time, provides organizations with useful data to measure engagement and impact via the admin dashboard.

### 6.2.4 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 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:

|  |  |
| --- | --- |
| Where: | |
| s | current state | |
| a | action taken | |
| r | reward received | |
| s′ | new state | |
| α | learning rate | |
| γ | discount facto | |

**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:

* The most recommended activity will be shown to user on the dashboard clearly.
* This recommendation is according to the way Q-table updates on backend.
* This is a visually guide of AI that helps the user pay attention to the most effective activity to get more points.
* On the other hand it also shows users the lowest q-value activity, which means not started since long time to cover all the activities.

**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 ensures 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

### 6.2.5 AI Avatar

To enhance user engagement and create a more personalized experience, an AI-based avatar was integrated into the user dashboard. This AI Avatar acts as a virtual assistant and greets the user upon entering the platform. The implementation of this avatar involved combining multiple AI tools and technologies in a structured manner. The voice was generated using [ElevenLabs](https://elevenlabs.io/?utm_source=chatgpt.com" \t "_new), a platform that specializes in **text-to-speech (TTS)** technology. Text-to-speech refers to a system that converts written text into spoken audio. Using ElevenLabs, a realistic and natural-sounding welcome message was recorded and exported as a pre-recorded audio file.

Next, a lifelike avatar image was sourced from public AI-generated image databases. To animate this image and bring it to life with mouth movements and facial expressions synced to the audio, the **HeyGen-AI platform** was used. HeyGen allows users to **generate talking avatars** by uploading both an image and a voice file, and it synchronizes them to simulate natural human speech and expressions.

The final animated avatar file (in video format with transparent background) was embedded into the **React-based** user dashboard UI. This avatar appears when the user logs in and provides a warm welcome, along with instructions to click the **"Need Assistance"** button if any support is required. This small yet powerful interaction adds a human-like touch to the interface, making the digital experience more interactive and user-friendly. Such use of AI enhances accessibility and usability while also aligning with the platform's goal of making sustainability engagement simple and welcoming for all users.

# 6.3 Special System Functionality Features for Corporate Users

## The website has two unique functional features that are important to encourage corporate user interaction. These are:

1. SDG Report Generator (powered by Ollama)
2. Security authentication

### 6.3.1 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.

### 6.3.2 Security authentication

* 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.

# 6.4 Backend Development and Data Flow

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 authentication and storing user actions). The backend of the platform is built using **Node.js** with the **Express.js** framework. **Node.js** is a JavaScript runtime that allows executing JavaScript code on the server side, while **Express.js** is a web framework for Node.js used to create API endpoints and manage routing between the frontend and the backend.

The main entry point for the backend is the index.js file, which initializes the Express server and manages API endpoints that interact with the database and frontend. When the frontend (built with React) sends a request such as a user logging in, starting an activity, or making a donation these requests are routed to the correct endpoint in the backend. Each request is processed and then connected to **Supabase**, the platform’s backend-as-a-service provider. Supabase manages the database operations and provides **real-time syncing**, **authentication**, and **secure storage** of user data.

The **data flow** between frontend, backend, and database works as follows:

1. **Frontend sends a request** (e.g., complete activity).
2. **Express backend receives** the request via API.
3. **Supabase client (used in backend)** sends a query to the relevant table (e.g., activity log, q\_table) to fetch, update, or store data.
4. **The backend sends a response** back to the frontend with the result (e.g., updated or confirmation message).

In addition to standard data exchange, the backend also includes an **AI engine** powered by **Python’s FastAPI**, running separately from the Node server. This Python-based system calculates **Q-values** for activity recommendations and updates the q\_table in the Supabase database.

A diagram of a software application

AI-generated content may be incorrect.

Finally, “**Ollama”**, a locally installed AI model server, is used when an organization clicks the “Download SDG Report” button. Data from the dashboard is sent to Ollama, which processes the content and returns a **generated Word report**. This combination of Node.js, Express, Python-based FastAPI, and Supabase ensures smooth, secure, and intelligent data processing between the user interface and the database.

# 6.5 Database Setup

The platform uses **Supabase**, which is a fully managed backend platform built on **PostgreSQL**, a powerful and reliable open-source relational database. The database is connected to both the web platform and mobile application, ensuring shared data access and real-time synchronization.

The following tables were created to support the full functionality of the platform:

* **Users** - Stores registered user data including ID, name, and email. Used by both Supabase Auth and backend analytics.
* **Activities** - Stores all sustainable activities, including their category, name, description, reward points, QR code values, estimated time, and which SDGs they affect. This is the master reference table for all user activities.
* **activity\_log** - Tracks each time a user completes or interacts with an activity, storing timestamps, rewards, and scanned QR codes. This supports analytics and performance tracking.
* **activity\_sessions** - Manages the “in progress” and “completed” status of an activity. If a user starts an activity on the website or app, it is marked as "in progress." Once verified (e.g., via QR scan), the status changes to "completed," and the website reflects this in the dashboard.
* **Categories** - Maps category codes to names (e.g., “Donate & Buy,” “Reuse/Recycle,” “Protect Wildlife”). Used to organize activities.
* **q\_table** - Contains the Q-learning matrix. It stores the current state, action, and calculated Q-value for every user-action pair. These values are updated based on user activity selections, completions and used to generate AI-based activity suggestions.
* **Rewards** - Tracks when a user redeems points for rewards. It stores user ID, total points gained, and points spent .
* **donate\_buy -** Stores user donation details including amount, transaction ID. Used to track donations and link them to user accounts for rewards and analytics.
* **strengthen\_body\_mind\_sprit** - Captures user mood ratings (1–5), feedback, and whether wellness suggestions were used. Helps monitor user wellbeing and engagement with mental health features.
* **need\_assistance** - Logs user-submitted support queries with their name, time, and message content. Allows the platform to respond to user issues and improve support services.

The **Supabase authentication system** works alongside these tables, enforcing strong security rules. For example, email addresses must contain “@”, and passwords must be at least 6 characters. Passwords are **hashed** (securely encrypted) before being stored. This schema ensures a robust and scalable structure for handling user actions, real-time updates, AI calculations, and personalized dashboards.