Currently

Project Overview

Overview of Currently

Currently I aim to design and implement a secure web application that can enable users to actively track, monitor, understand and hopefully reduce their household electricity consumption. The system will provide users with tools to record and track appliance usage, visualise energy consumption across different rooms of their house and generate estimated costs based on their usage patterns, all while ensuring user data is stored and processed securely.

What distinguishes this project is its **accessibility and emphasis on usability**. Unlike many existing solutions, this system does not require significant hardware investment. Instead, it leverages an appliance-level database and user-inputted data to generate practical insights. Features such as **"Map My House"** and **"Watch Your Watts"** provide intuitive visualizations and predictive simulations that are rarely available in consumer energy apps. The project also incorporates a **cybersecurity focus**, ensuring that user data is stored and processed securely.

Tech Stack

Frontend

- **Framework:** React **Styling:**CSS

Backend

- **Framework:** Java Spring Boot
- **Build Tool:** Maven
- **Security:** Spring Security (for user authentication and JWT token handling)
- **API Testing:** Postman / Insomnia

Database

- **Database: PostgreSQL**
    - Structured, easy to host, well-documented.
    - Pairs smoothly with Spring Boot via JPA (Hibernate).

Design and Documentation

- **Wireframing & UI Design:** wireframe.cc
- **System Diagrams:** PlantUML
- **Project Management & Documentation:** Notion

User Facing Pages

Public Frontend:

Landing Page

The Front page of Currently, consists of a legend and hero custom to Currently with information, the navbar consists of a sign Up/Log In button which takes you to that section

Sign Up/Log In Section

This section is where the user account will be created and stored within the DB or where the user successfully and securely logs into the Currently application and can use the features.

Public Components:

"public-nav": this is the navbar with two buttons "more Information" and "signup/login". Not to be used for private pages.

Private Frontend:

Dashboard

See your electricity usage at a glance. The dashboard gives you a clear estimate of your energy consumption and costs right when you log in.

Map My House

Build your home digitally. Start with one floor and add as many rooms as you like. Got a two- or three-story home? Add extra floors too. Choose from common room types or personalize your setup with custom names like "Liam's Room."

My Appliances

Bring your rooms to life by assigning appliances. Pick from our database of household appliances and link them to specific rooms. This makes it easy to see exactly where your energy is being used.

Watch Your Watts

Turn data into insight. See which rooms use the most electricity with interactive pie charts. See your "Biggest Eaters" — the top 5 appliances that drain the most power. Look ahead with predictive cost estimates, showing how much you could save by reducing usage.

Private Common components

"user-nav": This component is like the "public- header" but has interactive navigation links for each of the private functions (Dashboard, MapMyHouse etc)

Backend:

The backend of *Currently* is developed using **Java Spring Boot**, providing a secure, scalable and modular foundation for all user-facing functionality. It is responsible for managing user authentication, storing appliance and household data, handling analytics, and providing real-time responses to frontend requests through RESTful APIs.

Core Functions:

- **User Authentication & Authorization:**
  Uses **Spring Security** with **JWT tokens** to handle secure registration and login. Passwords are hashed using **BCrypt** before storage to protect user credentials.
- **Data Management:**
  Manages all user, room, and appliance data using **PostgreSQL**, accessed via **Spring Data JPA (Hibernate)**. This enables clean relational mapping (e.g., one user → many rooms → many appliances).
- **API Layer:**
  Exposes RESTful endpoints for the frontend, including /api/auth, /api/rooms, /api/appliances, and /api/usage. These endpoints are structured to ensure modularity and maintainability.
- **Predictive Insights Engine:**
  Contains logic for forecasting electricity costs and potential savings. This module processes user-inputted appliance data and estimated usage hours to generate daily, weekly, and monthly consumption predictions.
- **Security & Data Protection:**
  Integrates best practices such as **input validation**, **CORS configuration**, and **secure token-based sessions**. Sensitive user data is never exposed to the client-side.

Architecture Overview:

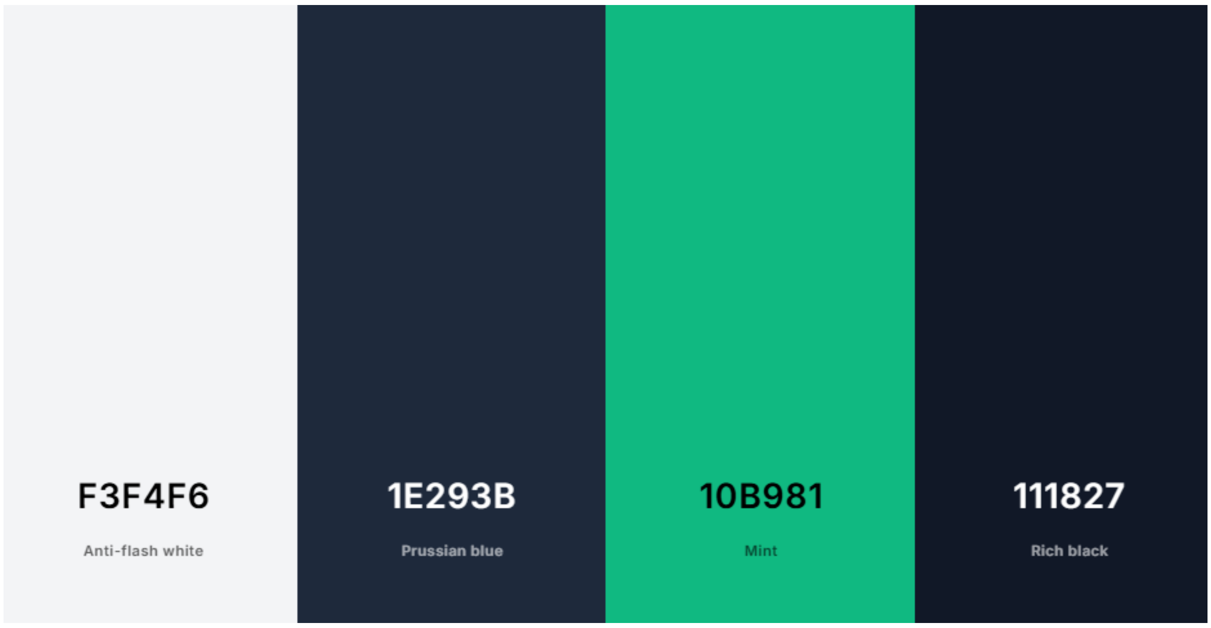
- **Controllers:** Handle HTTP requests and responses between the frontend and backend.
- **Services:** Contain business logic for authentication, appliance management, and energy analytics.
- **Repositories:** Manage database communication through JPA interfaces.
- **Models/Entities:** Define database structure (Users, Rooms, Appliances, UsageData).

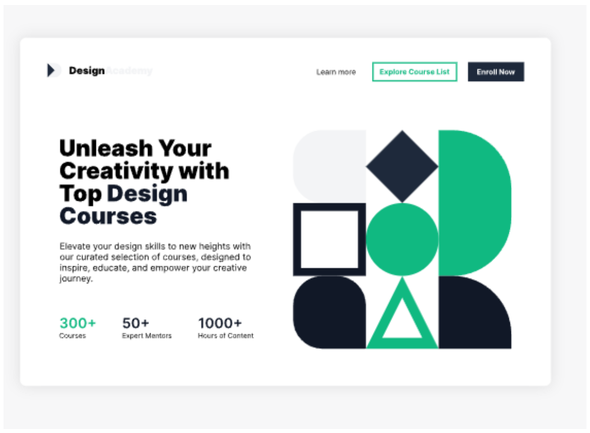
Documentation and Testing:

- API routes and functionality are documented in **Notion**, with endpoint testing performed via **Postman**.
- Unit and integration tests will be implemented using **JUnit and Mockito** to ensure backend reliability.

Variable.css styling/mood board

Colour scheme and mock web designs

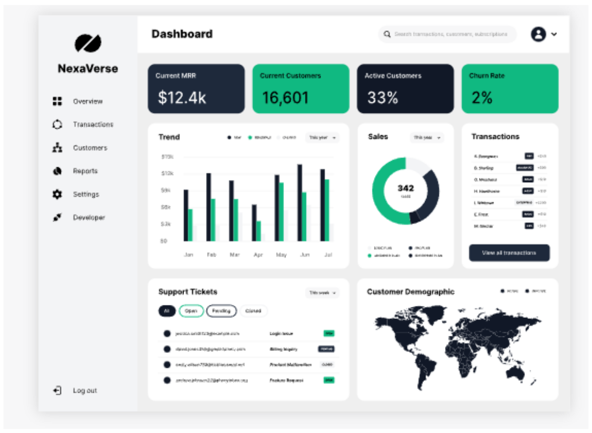




Unleash Your Creativity with Top Design Courses

Elevate your design skills to new heights with our curated selection of courses, designed to inspire, educate, and empower your creative journey.

300+ Courses
50+ Expert Mentors
1000+ Hours of Content



Root CSS colour scheme

```
:root {
      --color-white: #FFFFFF;
      --color-offwhite: #F3F4F6;
     --color-text: #111827;
     --color-accent: #10B981;
      --color-accent-dark: #1E293B;
}
```

Also using “Montserrat” google font for the main font throughout my project