# The Project

## Project Title

"Energy Consumption Benchmark: A Web Service for Energy Consumption Monitoring"

## Introduction

In this project, we'll embark on creating Energy Consumption Benchmark, a cutting-edge web service designed to monitor and analyze energy consumption from various devices. As the world moves towards sustainability, understanding and optimizing energy use has never been more important. Energy Consumption Benchmark aims to empower users by providing real-time data and insights into their energy consumption patterns, encouraging informed decisions towards energy efficiency and environmental sustainability.

## Project Overview

Energy Consumption Benchmark is a comprehensive platform that allows users to register their electrical devices, track energy consumption, and gain valuable insights into how to reduce their energy footprint. Utilizing Node.js for our backend and a modern, responsive frontend, we'll create a user-friendly web service accessible from various devices.

## Project Objectives

* **Develop a Full-Stack Web Service**: Learn the intricacies of full-stack development using Node.js, Express, and Tailwind CSS / Flowbite.
* **Implement Real-Time Data Processing**: Handle live data feeds of energy consumption, simulating real-world IoT data handling.
* **Create a User-Centric Design**: Focus on usability and user experience, ensuring our service is accessible and easy to use.

## Project Components

1. **Backend Development**: Using Node.js and Express, create a robust API that handles user management, device registration, and data processing.
2. **Frontend Development**: Design an intuitive and responsive user interface with HTML, CSS (Tailwind CSS / flowbit) and JavaScript.
3. **Database Management**: Design and implement a MySQL database schema for storing user data, device information, and energy consumption records.
4. **Data Visualization**: Implement interactive charts and reports for users to understand their energy consumption patterns.
5. **Security**: Ensure the application's security through proper authentication, authorization, and data validation techniques.

## Deliverables

* **Web Application**: A fully functional web service for energy consumption tracking.
* **Documentation**: Comprehensive documentation covering the API, database schema, and project setup.
* **Presentation**: A final presentation showcasing the project features, design choices, and lessons learned.

## Learning Outcomes

By the end of this project, students will have gained practical experience in:

* Full-stack web development using Node.js and modern frontend technologies.
* Developing and consuming REST APIs.
* Implementing user authentication and maintaining session states.
* Working with databases to store and retrieve data efficiently.
* Applying basic principles of software engineering, including version control, documentation, and collaborative development.

## Conclusion

Through teamwork, creativity, and technology, we will build a platform that not only enhances our understanding of software development but also contributes to a greener planet.

# Project Organization

## Team Organization

The project will be divided into 3 teams:

* Team 1 - Frontend team
* Team 2 - Backend team
* Team 3 - REST API team

Each team should have a defined team leader

## Project Timeline

The project will span 14 weeks, broken down into phases:

1. **Weeks 1**: Project planning, design, and setup.
2. **Weeks 2-13**: Development.
3. **Final Week**: Delivery and presentation.



## Milestones

### Frontend Team Milestones

#### User Interface Design Approval

* Finalize the design mockups for the user interface.
* Get approval on the design direction.

#### Frontend Prototype

* Develop a prototype of the user interface using HTML, CSS, Tailwind CSS and Flowbite.
* Ensure basic navigation and layout functionalities are implemented.

#### Data Visualization Integration

* Integrate data visualization libraries (e.g., Chart.js) into the frontend.
* Implement interactive charts and reports for energy consumption visualization.

#### Responsive Design Implementation

* Optimize the user interface for responsiveness across various devices.
* Test and ensure consistent user experience across different screen sizes.

### Backend Team Milestones

#### Database Schema Design and Setup

* Design an efficient database schema considering project requirements.
* Set up the MySQL database system based on the designed schema.
* Configure database connections, security settings, and initial data seeding.

#### Basic Setup

* Setup node.js and express.js
* Routing pages and static files

#### User and Device Management Implementation

* Develop endpoints for user registration, login, profile management, and device registration.
* Implement authentication, session management, validation, and error handling functionalities.

#### Data Processing Logic

* Develop backend logic for processing and analyzing energy consumption data.

#### Database Testing and Optimization

* Optimize database queries, indexing, and overall database performance.

### REST API Team Milestones

#### API Design and Documentation

* Collaborate with the frontend and Backend teams to design a clean and efficient API.
* Document the API specifications, including endpoint routes, request/response formats, and authentication mechanisms.

#### API Endpoint Development

* Implement CRUD endpoints for user data, device registration, and energy consumption records.
* Ensure endpoints follow REST principles and adhere to the API design specifications.

#### API Testing and Validation

* Conduct thorough testing of API endpoints using tools like Postman or Swagger.
* Validate endpoint functionality and data integrity.

#### API Documentation Completion

* Finalize comprehensive documentation covering all API endpoints.
* Include usage examples and sample requests/responses for each endpoint.

## Specifications and/or requisites

### Frontend Team

#### Homepage

A welcoming page with an overview of the web service, its benefits, and how it works.

##### Header

* **Logo:** Top left, featuring "EnergyConsumptionBenchmark" branding.
* **Navigation Menu:** Horizontally aligned on the top right, including links to "Home," "About," "Device Registration," "View Graphs," "Login/Signup."
* **Call-to-Action Button:** "Get Started" - prominent, leading new users to sign up or sign in.

##### Main Section

* **Hero Image:** A dynamic, engaging image symbolizing energy efficiency and technology.
* **Tagline:** A compelling, succinct statement summarizing the project's mission.
* **Brief Introduction:** A short paragraph about the significance of monitoring energy consumption and how "EnergyConsumptionBenchmark" can help.

##### Features Overview

* **Icons and Short Descriptions:** Summarize key features such as real-time data processing, user-centric design, and device registration.

##### How It Works

* **Step-by-Step Guide:** Illustrated steps on how users can register devices, monitor consumption, and analyze data.

##### Footer

* **Contact Information:** Email, phone number.
* **Quick Links:** To essential pages like FAQs, Support, Privacy Policy.

#### FAQ/Help Center

* **Description:** Provides answers to common questions and guidance on how to use the web service.
* **Key Features:** Categorized questions and answers, search functionality, and contact information for further support.

#### Contact Us Page

* **Description:** A page for users to contact the web service team for support or inquiries.
* **Key Features:** Contact form with fields for name, email, subject, and message, along with submission button. Optionally, include direct contact information like email or phone number.

#### Dashboard (After Login)

* **Description:** A personalized user dashboard displaying an overview of registered devices, recent energy consumption data, and quick access to key features.
* **Key Features:** Summary of energy consumption, list of devices, shortcuts to add a new device, view reports, and update user profile.

#### Device Registration Page

* **Description:** A page to add new devices to the user's account, specifying the device details and the connected electrical equipment.
* **Key Features:** Form fields for device name, type, and description of the connected electrical equipment, and a submit button. Optionally, include a unique device identifier if necessary.

#### Device Management Page

* **Description:** A central place for users to view, edit, or remove registered devices.
* **Key Features:** List of registered devices with options to edit their details or remove them from the account. Consider using a table or grid layout for clarity.

#### Energy Consumption Reporting Page

* **Description:** A page to view detailed reports on energy consumption for each device, with options to filter by date range or device.
* **Key Features:** Charts or graphs displaying energy usage, filter options for selecting specific devices or time frames, and possibly export functionality for reports.

#### User Profile Page

* **Description:** Allows users to view and edit their account information, such as email, password, and contact details.
* **Key Features:** Form fields for user information with edit functionality and options to delete the account or log out.

#### Some examples for inspiration

Homepage:

Dashboard:

<https://www.tailwindawesome.com/resources/flowbite-admin-dashboard/demo>

### Backend Team

#### Database Schema Overview

**MySQL Database Setup:** Install and configure MySQL. Design a schema that supports user data, device information, and energy consumption records.

**Seeding:** Implement seeding for initial data setup.

Users Table: Stores information about users who register on the platform.

user\_id: Primary Key, INT, AUTO\_INCREMENT

email: VARCHAR, unique

password\_hash: VARCHAR

name: VARCHAR

created\_at: DATETIME

updated\_at: DATETIME

Devices Table: Contains details of devices registered by users.

device\_id: Primary Key, INT, AUTO\_INCREMENT

user\_id: INT, Foreign Key referencing Users

name: VARCHAR

type: VARCHAR (e.g., HVAC, Refrigerator, etc.)

description: TEXT

created\_at: DATETIME

updated\_at: DATETIME

EnergyConsumptionRecords Table: Holds energy consumption data for devices.

record\_id: Primary Key, INT, AUTO\_INCREMENT

device\_id: INT, Foreign Key referencing Devices

energy\_consumption: FLOAT (energy consumed in a specific unit, e.g., kWh)

timestamp: DATETIME (when the record was generated)

Relationships

A User can have multiple Devices (One-to-Many relationship between Users and Devices).

A Device can have multiple Energy Consumption Records (One-to-Many relationship between Devices and EnergyConsumptionRecords)

#### Development Environment Setup

**Node.js and Express.js**: Setup the Node.js environment with Express.js framework for building the REST API.

**Routing and serving static files:** Setup the Node.js application to serve the web page and respective routings

**User and Device Management Implementation:** implement user authentication and management using helmet, cookie-parser middleware and bcrypt, jsonwebtoken libraries

#### Energy Graphs web Page

* **Description:** A dedicated section for users to access personalized energy consumption graphs and a comprehensive view of data from the open database. This dual perspective enables users to compare their consumption against broader trends.
* **Key Features:**
  + **Personal Energy Graphs:** Visual representations of the user's energy consumption over selectable time frames.
  + **Open Database Graphs:** Access to aggregated energy data from all users, allowing for benchmark comparisons.
  + **Filter Options:** Users can filter graphs by date, device, and comparison parameters.

### REST API Team

#### Sign Up Page

* **Description:** A page for new users to create an account.
* **Key Features:** Form fields for email, password, and possibly other relevant information (like name or contact info), and a submit button. Consider including validation feedback and a link to the login page for existing users.

#### Login Page

* **Description:** A page for existing users to access their accounts.
* **Key Features:** Form fields for email and password, a login button, and a link to the sign-up page for new users. Optionally, include a password reset link.

#### User Management

**Register User**

* + Endpoint: /api/users/register
  + Method: POST
  + Description: Registers a new user with the provided information.
  + Body: { email, password, name }

**Login User**

* + Endpoint: /api/users/login
  + Method: POST
  + Description: Authenticates a user and returns a token.
  + Body: { email, password }

**Get User Profile**

* + Endpoint: /api/users/me
  + Method: GET
  + Description: Retrieves the profile of the currently authenticated user.
  + Headers: Authorization: Bearer <token>

**Update User Profile**

* + Endpoint: /api/users/me/update
  + Method: PUT
  + Description: Updates the user's profile information.
  + Headers: Authorization: Bearer <token>
  + Body: { email, name }

#### Device Management

**Register Device**

* + Endpoint: /api/devices
  + Method: POST
  + Description: Registers a new device to the user's account.
  + Headers: Authorization: Bearer <token>
  + Body: { name, type, description }

**List Devices**

* + Endpoint: /api/devices
  + Method: GET
  + Description: Lists all devices registered by the user.
  + Headers: Authorization: Bearer <token>

**Update Device Information**

* + Endpoint: /api/devices/:deviceId
  + Method: PUT
  + Description: Updates information for a specific device.
  + Headers: Authorization: Bearer <token>
  + Body: { name, type, description }

**Delete Device**

* + Endpoint: /api/devices/:deviceId
  + Method: DELETE
  + Description: Removes a device from the user's account.
  + Headers: Authorization: Bearer <token>

#### Energy Consumption Data

**Submit Energy Data**

* + Endpoint: /api/data
  + Method: POST
  + Description: Submits energy consumption data for a specific device.
  + Headers: Authorization: Bearer <token>
  + Body: { deviceId, energyConsumption, timestamp }

**Get Energy Data**

* + Endpoint: /api/data/:deviceId
  + Method: GET
  + Description: Retrieves energy consumption data for a specific device.
  + Headers: Authorization: Bearer <token>
  + Query Parameters: startDate, endDate (optional for filtering)