EcoTrack Web Development

Table of Content

Introduction

EcoTrack is a project developed to meet a rising demand when it comes to easy to use, digital methods that could make users of those tools proactive when it comes to assessing their environment impact. This is normal to find large organizations depending on a complex system when it comes to sustainability tracking, individuals and the small to medium enterprises may lack either the technical facilities or resources to have such a system. The project will serve to eliminate that shortcoming by offering a user-friendly, but powerful platform that is easy to use without compromising fundamental functionality and safety of data. This app is aimed at providing a clean and friendly user experience with the help of a strong backend framework. The main characteristics is the user registration system, safe sign-in with coded credentials, dashboard interface that should be customized, and long-term and flexible NoSQL database storage of data (Bakke et al., 2022). Using these elements, EcoTrack does not only allow the user to leave an account of how much energy they consumed, how much carbon dioxide they released, and how much they recycled it allows them to establish sustainable habits and monitor their achievements over time. This deployment is unique in the sense that it has been done along the line of client accessibility. Opposed to GitHub access or use of command lines, EcoTrack is packaged as a ZIP file and may be easily distributed. The platform can be locally installed with a few steps of guided usage by users irrespective of their technicality. This report will become a technical document and a working guide to facilitate an easy work during installation, configuration and interim usage of the platform.

Methodology

The methodology of the development of EcoTrack was a well-thought and regulated one. This foundation was the identification of the most important user requirements and the establishment of the essence. The process of data entry and visualization of sustainability information in the form of a secure, clear, and accessible interface.

Planning

The initial stage was planning in which major flows like sign-up, sign-in, and dashboard handling were drawn. Security prerogative was considered, particularly in password processing, and a modular design that allows a higher-scale expansion without an architecture of the codebase.

Designing

The consistent visual theme was used during design phase, which is based on environmental colours mainly shades of green and off-white. The Fonts, containers, Cards, and buttons have been created keeping a simple modern appearance without frameworks (Santoro et al., 2024).

The frontend logic is sparse and specific-tuned and, as such, achieves the form validation and data submission with very few dependencies.

Development

The process of development was done in modular levels. The backend was developed using the Express.js, which deals with the RESTful routes of registering and logging in users (Anh et al., 2021). Mongoose is a very strong ODM that was used to integrate MongoDB, making it possible to define features such as User and communicate with the database in an easy manner. HTML, CSS and JavaScript files are in public folder and manage communication between the frontend and the server via fetch requests.

Testing

During development, certain aspects were carried out with manual testing, including proving form behaviour, connecting to MongoDB, and testing error cases, duplicate accounts, invalid logins, etc. The backend was designed with several backends of error handling to give useful feedback to users (Dudjak et al., 2020).

Documentation

Documentations were also written in a clear manner, such as the overall report and the in-code comments specifying the purpose of each module. Each stage the possibility of the ZIP delivery model was taken into consideration, and the guidelines on how to set it up, deploy it at runtime. Utilize the features of the library made no implicit prior knowledge of GitHub or sophistication in development pipelines and practices.

Tools and Technologies

EcoTrack harnesses a well-chosen set of tools and technological buildouts to construct a robust, high-performance, and infinitely scalable sustainability tracking capability. The idea was to make the toolset remain low weight but still strong enough to meet real cases in the world like user authentication, data manipulation, and user experiences across devices and browsers.

Frontend

HTML5

- Used to structure each webpage semantically, making the content readable by both browsers and screen readers. This forms the foundational markup of the app.

CSS3

- Makes custom styling and layout of the interface. With Flexbox and media queries, the UI dynamically adjusts to different screen sizes, ensuring a responsive design (Israpil et al., 2025).

JavaScript

- Client-side Interaction and validation of powers. This is applied to accept form data, manage buttons, and talking to back-end API by fetch. There is no use of frameworks making the frontend light and simple to debug.

Backend

Node.js

- Serves as the core JavaScript runtime on the backend, known for its non-blocking, event-driven architecture. This allows EcoTrack to handle multiple simultaneous requests efficiently.

Express.js

- Makes route management, middleware integration and API logic simple. All its backend functionality is sorted into clean and RESTful /api endpoints to Sign In, Sign Up and future actions.

Modular Routing

- Authentication and logic are separated into routes and model files, improving maintainability and scalability.

Database & Data Modelling

MongoDB

- The document-based NoSQL database used to store user profiles and environmental records. This supports flexible schema design, making it ideal for evolving data models (Brahmia et al., 2024).

Mongoose

- The Node.js ORM used for defining models like User, connecting to MongoDB, and performing validation and queries. It simplifies database interactions and helps enforce schema rules.

Security & Configuration

Bcryptjs

- Supplies password hashing with salting, which introduces a much-needed extra user credential verification. Passwords are not stored in the form of plaintext, thus avoiding potential security breaches.

dotenv

- Used to manage sensitive environment variables like the MongoDB URI. This allows credentials and configuration to be stored securely outside the source code.

Features

EcoTrack delivers a comprehensive feature set that addresses the essential needs of a modern sustainability tracking platform. This emphasizes usability, security, and modularity in its design.

User Registration

- Users can create an account using their full name, email, and password (Thomas et al., 2019).
- Frontend and backend validation make all inputs meet requirements.
- Friendly error messages guide users through failed sign-ups.

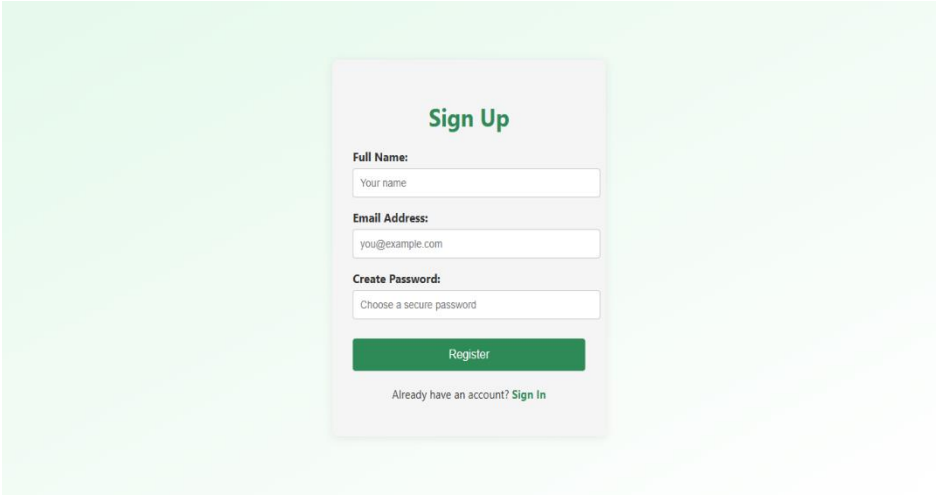

Figure 1: Signup Page

Secure Login

- Authentication follows best practices using bcryptjs to hash and verify passwords.
- Users are redirected to the dashboard on successful login.

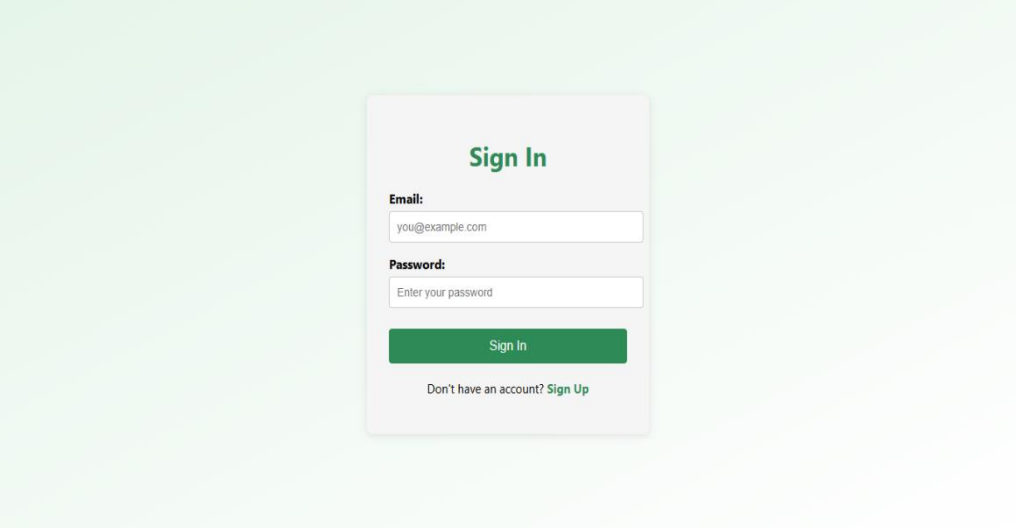

Figure 2: Signing Page

Personalized Dashboard

- After login, users can access their personalized dashboard to track sustainability metrics.
- Electricity usage in kilowatt-hours.
- Emissions measured in kilograms of CO_2.
- Recycling rates as percentages or volume.
- Simple form-based submission for daily or monthly tracking.

Data Storage & Management

- Entries are stored in MongoDB via Mongoose models (Erraji et al., 2023).
- Schema structure is designed for expansion (editing, viewing history, exports).
- Supports data persistence and scalability.

User Interface

- Unified, green-themed colour palette reflects the platform's environmental mission.
- Responsive design adapts to screens of all sizes.
- Clean input forms with intentional spacing for readability.
- Interactive buttons with hover feedback for improved UX.
- Global navigation links include Home, About, Contact, Sign In, and Sign Up.

Scalability & Modularity

- Backend organized using a modular folder structure (models, routes, config).
- Easily expandable to support.
- Chart-based visualizations using Chart.js or D3.js.

Pages

The EcoTrack has 6 primary pages written in HTML and are the front facing elements of the application. All these pages are dynamically functional though with backend data handling that is static in form.

Dashboard Page

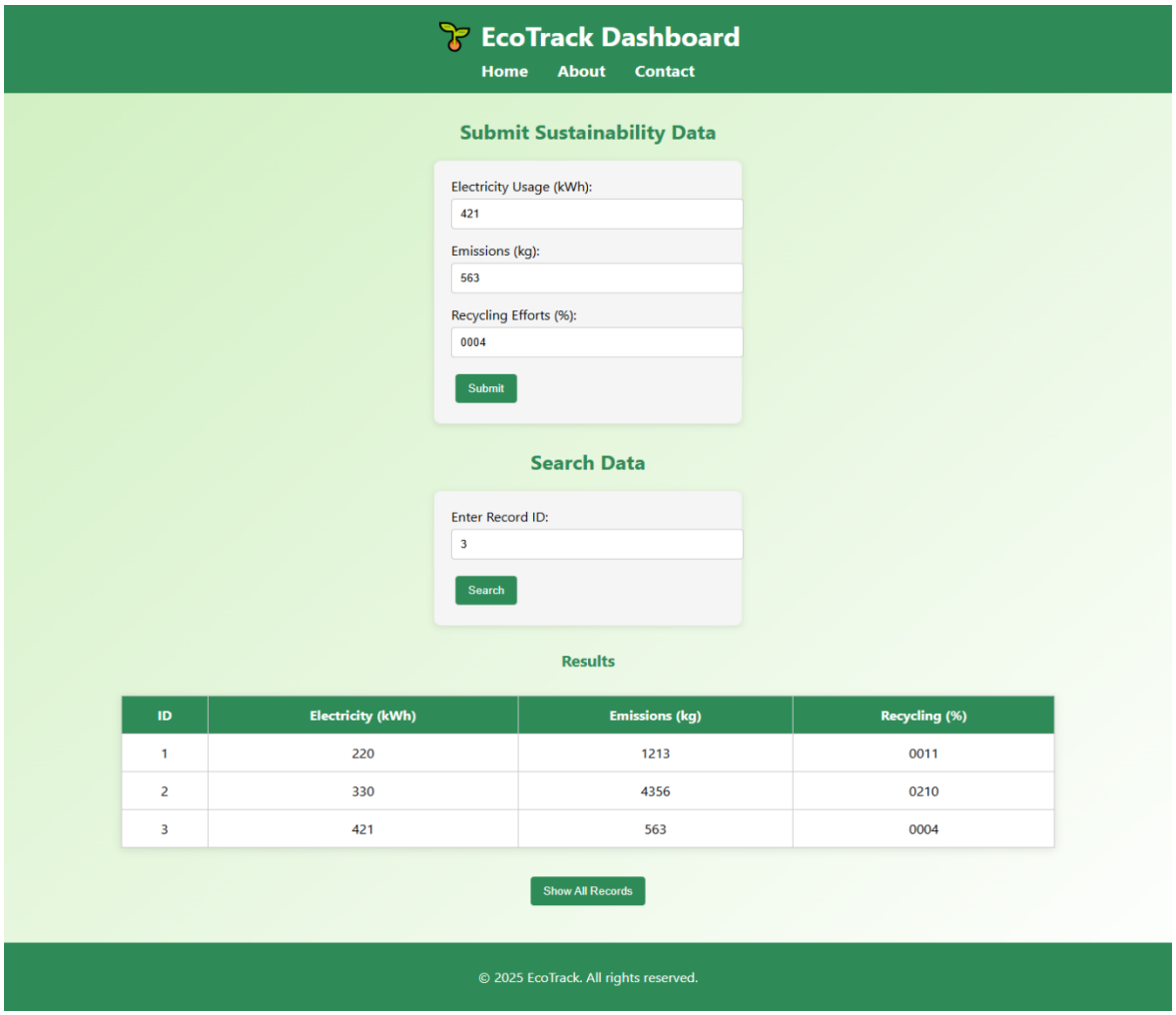

Figure 3: dashboard

The Dashboard page is an exclusive section, and it should be opened after the verification process. By the session handling is not made compulsory in this version, the page is structured in such a way to receive and print the data reported by using form. This layout is foresighted and is expecting improvements such as charts, summaries, and export facility in the future.

HomePage

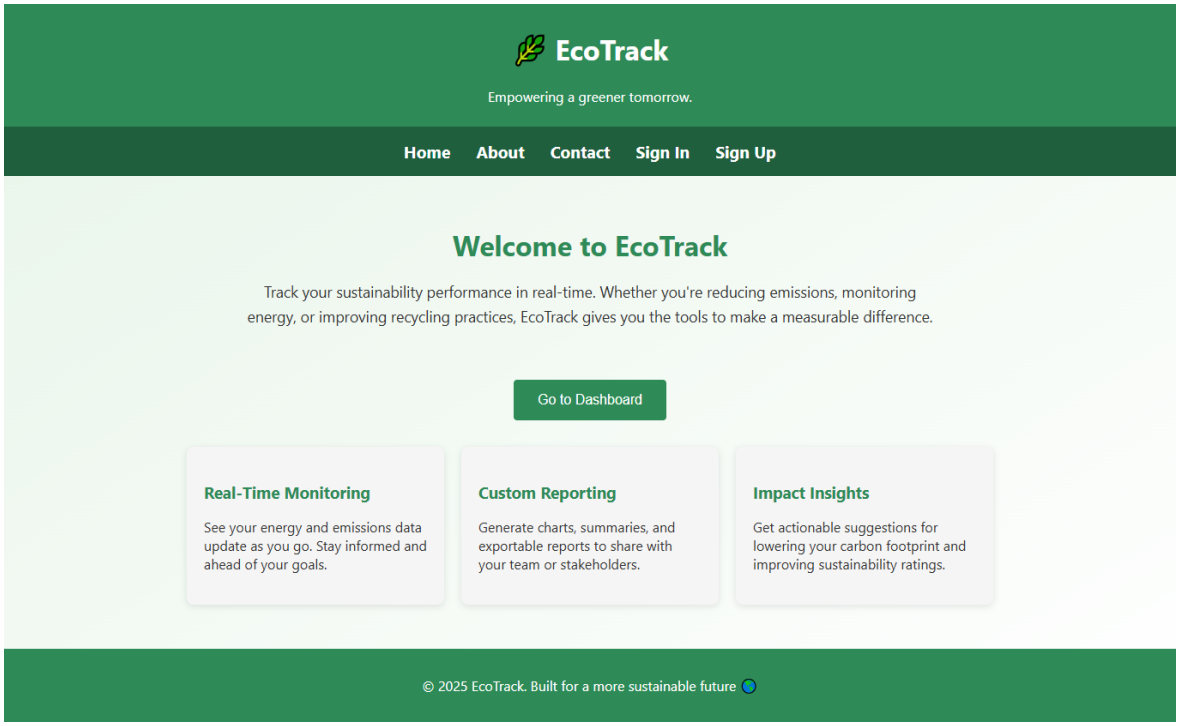

Figure 4: Hone Page

The homepage welcomes the user in the EcoTrack and has a call-to-action in the form of buttons that redirect to the registration and log-in page. The About content provides a mission statement and the purpose and the potential of the site. Cards summing up each of the key strengths of EcoTrack (Live Data Monitoring, User Reporting Tools, Environmental Suggestions, Planned Collaboration Features, and so on) are also present on this page.

Contact Page

The Contact page, there is formatted form in which information such as email address and a message can be entered. This is not connected to an email backend in the present form but can be integrated with systems such as EmailJS or Formspree (Arslan et al., 2020).

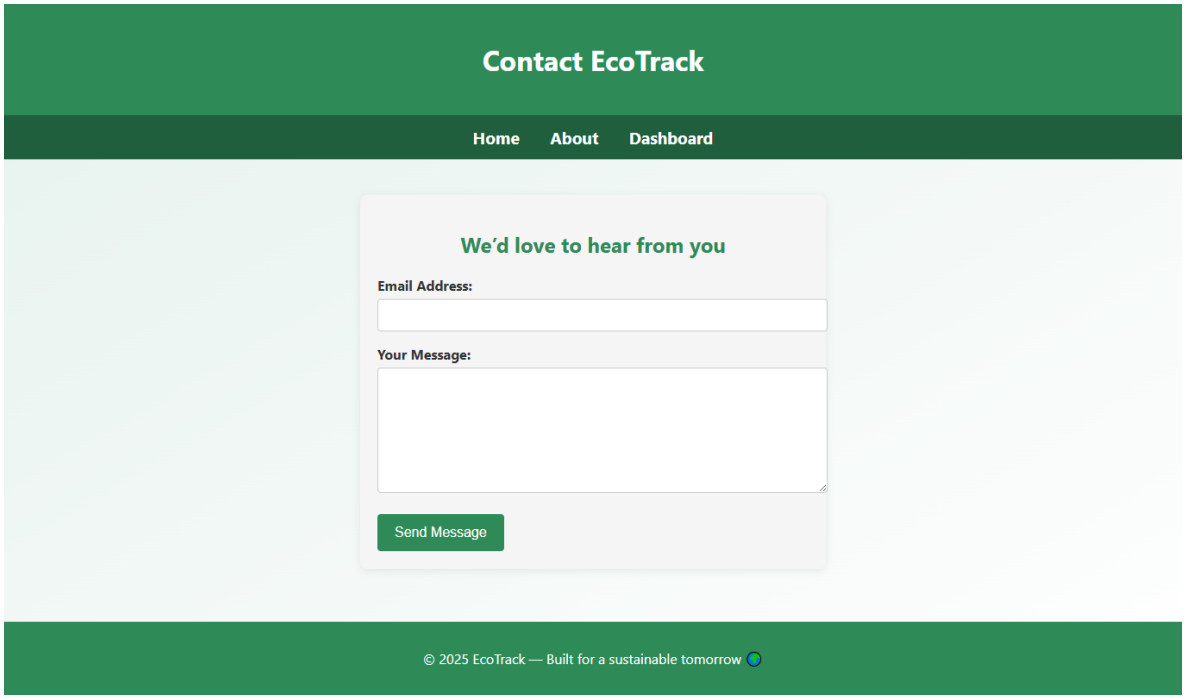

Figure 5: Contact Page

About Page

The About page provides an introduction of EcoTrack, which describes its mission statement of being a sustainability tracking tool. This explains the ultimate purpose of the platform and essential functions, and why it is important to keep track of the environmental data associated with a person.

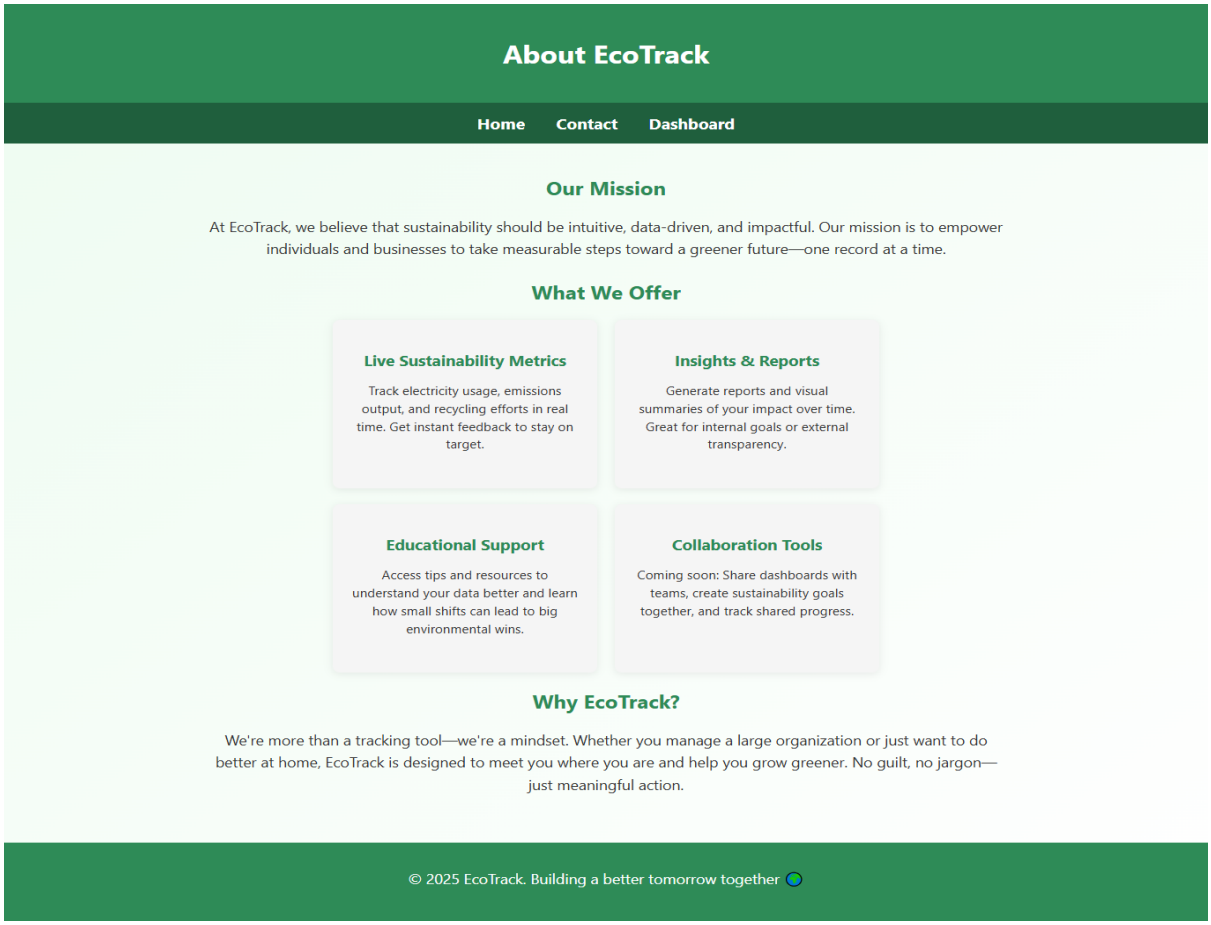

Figure 6 About Us Page

Sign up Page

The Sign Up and Sign In pages are not only well designed but also easy to use. They provide input validation and on-submit feedback, and their forms relate to the Express backend. Successful registration or login will take a user to the Dashboard.

Functional and Non-Functional Requirements

Functional Requirements

- Users can create the account using their full name, email, and password.
- Input fields are validated on both the frontend and backend to make accuracy.
- Duplicate registration using the same email is prevented with real-time error feedback.
- Passwords are hashed using bcryptjs before being stored securely in MongoDB.
- Users can log in using valid credentials that are matched against hashed records (Thomas et al., 2019).
- Upon successful login, users are redirected to a personalized dashboard.

Users can submit sustainability data, including.

- Electricity usage in kilowatt-hours.
- Carbon emissions in kilograms of CO_2.
- Recycling percentage or quantity.

Non-Functional Requirements

- The application performs efficiently with local server response times under one second.
- MongoDB and Express.js offer fast I/O with low memory consumption.
- Modular file structure supports long-term scalability and maintenance.
- Application is cross-platform and portable across systems with Node.js and MongoDB (Sharma et al., 2022).

Conclusion

EcoTrack is a well-crafted sustainability tracking software, a combination of accessibility and technical complexity. The modular full-stack architecture built on the ground up allows the user to take charge and contribute effectively to the cause of environmental awareness by tracking and reflecting daily impact through energy spent, emissions, and recycling activities. Focusing on the intuitive design, understandable workflows, and good security offered by using bcryptjs hashing and configuration based on an environment, EcoTrack brings environmentally friendly living to a level where it can be tracked and accomplished. This strict separation between the frontend, backend, and database roles guarantees that the system can be maintained and scaled, even when new features, e.g. data visualization or session-based authentication, or system-administration dashboards will be added to the application. The adoption of open, well-known web technologies by the platform reduces the threshold of customizing it, being thus an appropriate one to implement into numerous client requests and learning environments. Thev source of long-term sustainability, EcoTrack can serve as a learning project, prototype of solutions, as well as a 10-year background tool of more extensive data-driven ecological efforts. This transparency, flexibility and adaptivity are its key asset in further innovation which is based on increased demand of accessible environmental products of the digital era. This does not only follow impact, but helps its users comprehend and influence it.

References

Anh, V. (2021). Real-time backend architecture using Node. js, Express and Google Cloud Platform.

Arslan, R. C., Walther, M. P., & Tata, C. S. (2020). formr: A study framework allowing for automated feedback generation and complex longitudinal experience-sampling studies using R. Behavior research methods, 52(1), 376-387.

Bakke, T., Strøm, S. A., & Tengs Hafsø, H. (2022). System development of dashboard application-Visualizing customer energy data (Bachelor's thesis, NTNU).

Brahmia, Z., Grandi, F., & Oliboni, B. (2024). A literature review on schema evolution in databases. Computing Open, 2, 2430001.

Dudjak, M., & Martinović, G. (2020). An API-first methodology for designing a microservice-based Backend as a Service platform. Information Technology and Control, 49(2), 206-223.

Erraji, A., Maizate, A., & Ouzzif, M. (2023). An integral approach for complete migration from a relational database to MongoDB. Journal of the Nigerian Society of Physical Sciences, 1089-1089.

Israpil, R. (2025). Approaches to Creating Adaptive Design in Mobile Applications Using React Native. Universal Library of Engineering Technology, 2(1).

Santoro, F. (2024). Optimizing Industrial Operations: A Web Application for QR Code-Based Machinery Information Management (Doctoral dissertation, Politecnico di Torino).

Sharma, M. (2022). Full Stack Development with MongoDB: Covers Backend, Frontend, APIs, and Mobile App Development Using PHP, NodeJS, ExpressJS, Python and React Native. BPB Publications.

Thomas, K., Pullman, J., Yeo, K., Raghunathan, A., Kelley, P. G., Invernizzi, L., ... & Bursztein, E. (2019). Protecting accounts from credential stuffing with password breach alerting. In 28th USENIX Security Symposium (USENIX Security 19) (pp. 1556-1571).

Thomas, K., Pullman, J., Yeo, K., Raghunathan, A., Kelley, P. G., Invernizzi, L., ... & Bursztein, E. (2019). Protecting accounts from credential stuffing with password breach alerting. In 28th USENIX Security Symposium (USENIX Security 19) (pp. 1556-1571).