|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| GreenPath.  Sustainable Products Website with AI ChatBot Integration.  Instructor: Dr. Sarif Bambang. |
| |  |  |  | | --- | --- | --- | | Awoyale, Francis  (300356743) | Fall 2025 | Applied Research Project(CSIS 4495) | |

**1. Introduction and Background.**

GreenPath is an applied research project that adapts the concept of Happy Nutrition Riipen project brief to the broader sustainability domain. The aim is to develop a modern, cloud‑hosted website that educates users about sustainable product choices. The system combines a React frontend, a Node.js backend, and a PostgreSQL database deployed on Amazon Web Services. An AI‑powered chatbot provides conversational guidance, and a sustainability analytics dashboard that presents real world emissions data in an accessible, interactive format. The project emphasizes accessibility, performance, and transparency of data use so that results are meaningful to a general audience.

**2. Problem Definition.**

Sustainability information is plentiful but difficult to navigate. Consumers frequently encounter fragmented sources, outdated interfaces, and static charts that do not support exploration. This reduces engagement and limits informed decision‑making. GreenPath addresses these limitations by offering a single interface that unifies three elements: an intuitive explorer for sustainable product categories, an AI assistant that can answer practical questions in natural language, and a data driven analytics module that visualizes sector‑level carbon emissions. The combined approach lowers the barrier to understanding sustainability trends and connects those trends to actionable choices.

**3. Literature Review and Research Gap.**

Contemporary sustainability dashboards provide valuable statistics yet remain largely static and detached from conversational support. Well‑curated repositories such as Kaggle host emissions datasets that can be used for analysis and forecasting; however, end‑user tools that combine chart‑based exploration with a lightweight machine learning component and a guided conversational layer are uncommon. At the same time, web development literature underlines the importance of accessibility and performance, and the machine learning literature shows that interpretable techniques such as linear regression can provide clear, defensible insights. GreenPath targets this gap by integrating an accessible web interface, interpretable analytics, and an AI assistant into a single, deployable system.

**4. Research Assumptions and Hypothesis**

This project assumes that sector‑level carbon emissions for a country evolve in patterns that can be modeled with simple, interpretable techniques and that such patterns are sufficiently stable over medium time horizons to support short‑term prediction or categorical grouping. It further assumes that interactive charts and a conversational interface increase users’ understanding and retention compared to static pages. The central hypothesis is that a combined approach consisting of interactive analytics, a focused machine learning task, and an AI assistant, will provide clearer and more engaging sustainability insights than a conventional static website.

**5. Proposed Research Project (Design, Objective, Method, Data, Analysis)**

Design: GreenPath delivers three user‑facing capabilities: a category explorer that presents examples of eco‑friendly products with details, a chatbot that provides sustainability guidance, and an analytics dashboard that exposes sector‑level emissions trends. The system is implemented with React and TypeScript on the client, a Node.js and Express backend running on AWS Lambda behind API Gateway, and a PostgreSQL database hosted on Amazon RDS with RDS Proxy to manage connections. The front end is hosted through AWS Amplify and served via CloudFront. Secrets such as the OpenAI API key are stored in AWS Secrets Manager.

Objectives: The objective of GreenPath is to deliver a deployed and accessible web application that helps users understand sustainability topics through an integrated experience of product exploration, conversational assistance, and data‑driven insight. Concretely, the project aims to build a responsive React interface hosted on AWS Amplify, implement a serverless Node.js backend that secures keys and serves compact APIs, integrate an AI‑powered chatbot for sustainability question‑and‑answer interactions, and provide an analytics dashboard with at least three interactive charts that present sector‑level CO₂ emissions over time in gigatonnes. A focused machine learning component will be trained and embedded into the dashboard to generate short‑term forecasts for a selected country and sector, with results presented transparently and accompanied by interpretation notes. The application will adhere to WCAG 2.1 Level AA practices and target a strong baseline performance so that the experience is fast, readable, and keyboard‑friendly for all users.

Method: The project begins by preparing the data and establishing a stable deployment pipeline. The React application is scaffolded, and the core pages are created for the home view, the categories, the analytics dashboard, and the chatbot. The backend exposes a small set of REST endpoints for categories and analytics, and a single endpoint that relays chatbot prompts to the OpenAI API. Once the pipeline is in place the focus shifts to data ingestion, analysis, and presentation. Accessibility and performance are evaluated throughout using a combination of automated tools and manual checks.

Data collection and preparation: The analytics module uses the CO₂ Emissions Dataset by Saloni1712 on Kaggle. The dataset includes the country or region, the year, a sector designation, a numerical value, and a timestamp. The working copy is normalized by renaming columns to country, year, sector, and value; removing the timestamp; filtering to a practical subset of countries; and indexing the database for efficient queries. A magnitude check indicates that values should be interpreted as gigatonnes of CO₂, and charts and labels are annotated accordingly so users understand what the numbers represent.

Sample size and data scope: For a realistic and manageable analysis during the project, the minimum viable sample will focus on a representative subset of countries such as Canada, the United States, the United Kingdom, Germany, Brazil, Nigeria, India, and China, spanning approximately 2019 to 2023 across all six sectors. This scope yields several hundred to more than a thousand records, which is ample for descriptive analytics and for training a simple, interpretable model while retaining headroom for validation. The full dataset remains available in PostgreSQL for extended exploration, but the forecasting model will be trained on the defined subset using a chronological split that reserves the most recent years as a holdout period to approximate real‑world prediction.

Analysis techniques: The dashboard exposes interpretable views that support exploration. A line chart shows how a sector’s emissions evolve over time for a given country. A stacked bar or grouped bar enables a yearly comparison of sector contributions. A small machine learning component is integrated into the dashboard to demonstrate applied analytics. For the minimum viable product the preferred option is a simple regression that predicts the next year’s sector value for a selected country using recent history; alternatively, a compact classifier can assign countries to high, medium, or low emission categories based on historical distributions. Results are presented alongside the charts with clear statements of limitations.

Technology justification: React with TypeScript provides a fast, accessible interface and strong developer ergonomics. Node.js on Lambda minimizes operational overhead while API Gateway and RDS Proxy provide reliable connectivity. PostgreSQL offers mature querying, indexing, and aggregation suitable for the analytics workload. Recharts integrate cleanly with React for accessible visualizations.

Expected results: The outcome will be a working system that combines a friendly interface, meaningful analytics, and a modest predictive capability. Users will browse sustainable product categories, ask questions through the chatbot, and explore emissions data with interactive charts that clarify sector contributions and year‑over‑year changes. The model will produce short‑term forecasts for a chosen country and sector, visualized alongside historical trends and accompanied by plain‑language explanations of uncertainty, assumptions, and limitations. From an engineering perspective, the system is expected to achieve a Lighthouse performance score in the mid‑eighties or higher, pass core accessibility checks aligned to WCAG 2.1 practices, and maintain low API latency under typical classroom loads. From an analytics perspective, the regression baseline should reach a reasonable fit on the validation period for at least one country‑sector pair, emphasizing interpretability and defensibility over complexity.

**6. Riipen External Partners or Affiliate**

There is no external Riipen affiliate for this project. The work is student‑designed and executed as an individual applied research project.

**7. Project Planning and Timeline**

The project plan follows a conservative sequence so that each component becomes demonstrable as early as possible. Weeks one and two cover requirement confirmation, dataset selection, repository setup, and a baseline accessibility and performance measurement. Weeks three and four deliver the React scaffold, the category prototype, Amplify hosting, and an initial accessibility pass. Weeks five and six establish the back end, the database schema, secure key storage, and the ingestion of the Kaggle dataset into PostgreSQL. Weeks seven and eight integrate the chatbot and the front end, including basic logging. Weeks nine and ten complete the analytics dashboard with two to three charts that use gigatonne annotations. Week Eleven delivers the machine learning component integrated into the dashboard. Week twelve focuses on quality improvements, final reporting, and the demonstration script.

Below is the project Gantt chart that corresponds to the timeline described above. The first task ends at week two, and each subsequent task matches the stated two‑week duration until the final two single‑week tasks.

A graph with colored squares

AI-generated content may be incorrect.

**8. Project Contract**

I, Francis Awoyale (Student ID: 300356743), agree to complete this project and to deliver the scope outlined in this proposal. I will maintain a work log with a minimum of one hundred and twenty hours, submit progress and final reports on schedule, and follow the instructor’s directions and the course policies.

Signature: F.O.A Date: 21ST of September 2025.

**9. Work Date/Hour Log**

The work log records the date, hours spent, and description of work done in each milestone as shown in the work log below. The log will be extended through the duration of the project to reach at least one hundred and twenty hours.

|  |  |  |
| --- | --- | --- |
| Date | Number of Hours | Description of Work Done |
| 2025-09-12 | 1 | Created GitHub repo, added instructor as collaborator. |
| 2025-09-13 | 3 | Initial research on the topic.  Selected Kaggle dataset (Saloni1712) and verified units. |
| 2025-09-14 | 3 | Drafted initial project proposal with architecture. |

The complete log will continue in this format for all subsequent sessions, including back‑end development, data ingestion, analytics, machine learning, testing, and documentation.

**10. Closing**

GreenPath offers a realistic and educational path to demonstrating applied research outcomes. The project integrates accessible web design with cloud architecture, curated sustainability data, and a focused machine learning task.

**11. References**

W3C. Web Content Accessibility Guidelines (WCAG) 2.1. World Wide Web Consortium.

Google. Web Vitals and Lighthouse documentation. Google Developers.

Amazon Web Services. Amplify, API Gateway, Lambda, Amazon RDS for PostgreSQL, Secrets Manager, and CloudWatch service documentation.

OpenAI. API Documentation for text generation and chat completions.

Kaggle – *CO₂ Emissions Dataset by Saloni1712*. Retrieved from: [https://www.kaggle.com/datasets/saloni1712/co2-emissions](https://www.kaggle.com/datasets/saloni1712/co2-emissions?utm_source=chatgpt.com)

Géron, A. (2022). Hands‑On Machine Learning with Scikit‑Learn, Keras, and TensorFlow (3rd ed.). O’Reilly Media.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning Springer.

Sheehan, B., et al. (2020). The role of AI‑powered chatbots in health promotion: a scoping review. Journal of Medical Internet Research.