ENIGMA

(Personalized Energy Hub)

by

Diyotrim Maitra (2348028)

Jiss Sabu Varghese (2348034)

Shreya Ghosh (2348061)

Under the Guidance of

Dr Monisha Singh

A Project report submitted in partial fulfilment of the requirements for the award of the degree of Master of Science (Data Science) of CHRIST (Deemed to be University)

September 2024

CERTIFICATE

*This is to certify that the report titled **ENIGMA** is a bonafide record of work done by **Diyotrim Maitra** (2348028) of CHRIST (Deemed to be University), Bengaluru, in partial fulfillment of the requirements of IV Semester MSc (Data Science) during the academic year 2024-25.*

Head of the Department **Project Guide**

Valued by

1.

2.

Name	:	Diyotrim Maitra
Register Number	:	2348028
Date of Exam	:	12/09/24

ACKNOWLEDGEMENTS

First and foremost, we are deeply grateful to Dr. Saleema, Head of the Department of Statistics and Data Science (Christ University), for providing us with the opportunity to work on this innovative web project.

We would like to thank our guide, Dr. Monisha Singh for her unwavering support and great guidance throughout the project. Her expertise and insightful feedback helped identify and address gaps and limitations in our web project. Her constructive feedback and unwavering dedication have significantly enhanced the quality of our project, and we are truly fortunate to have had the opportunity to work under her mentorship.

Additionally, we owe special thanks to Mr. Badri Narayanan, an expert in Large Language Models (LLMs), for his crucial assistance in building the chatbot and LLM models. His technical expertise and collaborative efforts played a significant role in enhancing the functionality and overall success of the project.

ABSTRACT

Enigma- Personalised Energy Hub was developed to educate individuals, businesses, and organizations about energy efficiency and conservation, aligning with the United Nations' Sustainable Development Goal (SDG) 7. SDG 7 focuses on ensuring access to affordable, reliable, sustainable, and modern energy for all. By raising awareness and providing actionable insights, Enigma empowers users to make informed decisions about energy consumption, promoting sustainability and contributing to global energy conservation efforts.

The platform includes a range of tools and features designed to help users optimize their energy use. These include an Efficiency Assessment Tool, an Electricity Price Prediction Model, and an Individualized Carbon Footprint Calculator. Additionally, users can access an Energy Prediction Output Tool, Cooling System Analysis, and an advanced Energy Recommendation System. Interactive dashboards display key energy metrics, while a cutting-edge RAG chatbot provides seamless navigation, guiding users through the platform and offering insights on energy conservation.

Enigma delivers personalized energy solutions that help users enhance energy efficiency, reduce costs, and minimize environmental impact. The platform provides efficiency ratings, predicts electricity usage and prices, and calculates carbon emissions based on user inputs. With actionable recommendations for improving energy usage and an intuitive interface, Enigma fosters better energy management practices, making it easier for users to contribute to a sustainable future

Keywords: Renewable Energy, SDG 7, RAG Chatbot

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Problem Description

In today's world, energy consumption is rising at an unprecedented rate, leading to significant environmental, economic, and social challenges. Many individuals, businesses, and industries struggle to make informed decisions about energy efficiency, sustainability, and cost optimization due to a lack of accessible and accurate tools. Additionally, the global push for clean energy, as outlined in Sustainable Development Goal (SDG) 7, requires actionable insights that can help users reduce their carbon footprint and adopt more sustainable energy practices.

Traditional platforms often fail to provide the personalized, in-depth analysis required to address these challenges comprehensively. Users need detailed information on the efficiency of energy sources, the environmental impact of their choices, and the health risks associated with certain energy practices. Furthermore, navigating these insights on conventional platforms can be complex and time-consuming, leaving many users without the proper guidance to take meaningful action.

Achieving SDG 7 is crucial not only for mitigating climate change but also for fostering long-term economic growth and ensuring energy security. Industries, especially energy-intensive sectors like power generation, steel production, and chemical processing, play a pivotal role in global energy consumption and emissions. These industries must embrace sustainable practices to reduce their environmental footprint and align with global efforts to ensure universal access to affordable, reliable, and sustainable energy. Failure to adopt energy-efficient technologies and reduce emissions not only contributes to climate change but also risks regulatory penalties, increased operational costs, and reputational damage. By prioritizing energy efficiency, renewable energy adoption, and sustainable cooling systems, industries can significantly lower emissions, reduce water and resource consumption, and drive innovation in sustainable technologies—ultimately playing a key role in realizing SDG 7 and contributing to a cleaner, more sustainable future for all.

Enigma addresses these issues by offering a three-tiered energy management solution that caters to different user needs Its advanced tools, including the Predictive Energy Efficiency Model and a RAG chatbot, provide users with tailored recommendations, sustainability insights, and intuitive support, enabling them to make more responsible energy decisions while contributing to SDG 7.

1.2 Existing System

Several platforms exist today that address aspects of energy management and sustainability, but most fall short of providing comprehensive and actionable recommendations. For example, [1] Renewables.ninja is a website that allows users to explore the potential of renewable energy sources such as solar and wind, by incorporating geography. However, while it provides valuable data on energy availability, it doesn't add cost or efficiency of the renewable source of energy as a variable. It fails to provide any valuable insights or recommendations and hence, users are left without the tools to decide which energy source is best for their specific needs, especially in the context of varying environmental conditions.

Another platform, [2] Our World in Data, provides datasets and basic visualizations on global energy consumption and environmental impact. While it offers useful information, the platform is primarily data-driven and does not provide deeper insights or recommendations for energy optimization. Users are required to interpret the data themselves without any guidance and hence would be a hectic task for non-statisticians.

Moreover, there exists no generalized website that offers personalized recommendations for power plants based on key environmental and operational variables like temperature, pressure, humidity, and exhaust vacuum. This gap in the market leaves power plant businesses with no clear direction on how to calculate their energy generation system's efficiency for both performance and sustainability. Moreover, there are no websites that aids power plant owners on deciding which cooling system to choose based on their economical budget, efficiency and geography.

There are a few websites that offer functionalities similar to Enigma, but they come with significant limitations. For instance, platforms like [3] EnergySage and [4] OpenEI provide some insights into energy costs and savings. However, they do not provide personalized efficiency recommendations based on real-time data. Additionally, their focus is primarily on renewable energy sources, leaving out cost factor which also influences industry owners on choosing their source of energy. Enigma resolves these flaws by offering a more holistic approach that considers a wide range of energy sources and operational factors.

Another major limitation in most existing carbon footprint calculators is that they only take 3-4 inputs from users, such as transport type, diet, etc. This oversimplification leads to inaccurate results that do not reflect an individual's true environmental impact. In contrast, Enigma's carbon footprint calculator takes 14 detailed inputs, including variables such as body type, heating energy source, vehicle type, frequency of air travel, and energy-efficient device usage. This depth of data ensures far more accurate and personalized carbon emissions estimates, enabling users to make more informed decisions to reduce their environmental footprint.

1.3 Project Scope

The scope of the Enigma platform is vast, encompassing a range of functionalities that cater to individuals, industries, and businesses aiming to optimize energy consumption, enhance efficiency, and contribute to Sustainable Development Goal (SDG) 7. The platform is designed to provide users with tailored insights and actionable recommendations across various sectors, helping them make informed decisions regarding energy use, carbon emissions, and sustainability efforts.

At its core, Enigma offers a three-tiered plan (Free, Pro, Business), each aimed at specific user groups. The Free Plan targets individual households, providing essential functionalities such as device energy efficiency assessments and personalized carbon footprint calculations. The platform also extends its functionalities to businesses and industries through the Pro and Business Plans. Enigma’s visualizations and data-driven insights also allow users to stay informed about global energy challenges and potential solutions. By integrating advanced predictive models, detailed input parameters, and user-friendly tools, Enigma provides a unique, all-encompassing solution for achieving energy sustainability.

The Enigma platform offers a suite of tools to promote energy efficiency and environmental awareness. The Device Efficiency Calculator helps users evaluate the performance of their devices based on factors like age, malfunctions, and brand, alerting them if the device is inefficient. The Carbon Footprint Calculator enables users to estimate their carbon emissions based on lifestyle choices, fostering awareness of their environmental impact. For thermoelectric power plants, the Power Plant Efficiency Evaluator provides customized recommendations on optimizing plant efficiency and cooling systems, enhancing performance and reducing emissions. The Personalized Industry Energy Source Optimizer supports energy-intensive sectors by offering tailored insights to optimize energy use and minimize emissions, considering factors such as location, budget, and renewable energy efficiency. Additionally, the RAG chatbot assists users by providing guidance on the platform’s functionalities and energy-saving practices, enhancing user engagement and promoting sustainability.

2. SYSTEM ANALYSIS

2.1 Functional Specifications

The Enigma platform is designed to provide users with comprehensive energy insights and tools to optimize their energy usage and reduce their carbon footprint. The key functional specifications include:

2.1.1 Energy Efficiency Insights

- Allows users to input details about their household devices including type, brand, age, usage hours, and malfunction history.
- Evaluates device efficiency based on brand performance, typical energy consumption patterns, and impact of malfunctions.
- Calculates the cost per day to operate each device based on local energy rates, device efficiency, and usage.

2.1.2 Carbon Footprint Calculator

- Considers a wide range of lifestyle factors including body type, sex, diet, shower frequency, transportation mode, air travel, waste production, and electronic device usage.
- Provides a precise estimate of the user's daily carbon emissions in kilograms Offers personalized recommendations to reduce carbon footprint based on the user's profile

2.1.3 Visualization Tools

- Dashboards:
    - Three dashboards cater to different sectors:
        - Household Dashboard: Provides insights on household energy consumption and efficiency.

        - Power Plants Dashboard: Focuses on energy production metrics and performance of power plants.
        - Industry Dashboard: Analyzes international finance, consumption of renewable energy sources, and industry-specific data.
- Charts in Services Section:
    - Ozone Concentration: Displays levels of ozone pollution across various regions.
    - Decadal Anomalies: Highlights temperature and climate anomalies over the decades.
    - Deaths Due to Natural Disasters: Illustrates the impact of natural disasters on human life, focusing on regions most affected.
    - CO2 Emissions Worldwide: Provides a global overview of carbon dioxide emissions from various sources.
    - Pollution and Emission: Tracks pollution trends and emissions from different pollutants, emphasizing their environmental impact.
    - Nuclear Energies: Shows the usage of nuclear energies around the world.

These visualization tools are designed to raise awareness about global environmental issues, energy crises, and the importance of sustainable development, particularly aligned with Sustainable Development Goal (SDG) 7.

2.1.4 Subscription Plans

The Enigma platform offers two types of subscription plans: PRO and BUSINESS. Each plan is designed to cater to specific user needs and provides unique features.

2.1.4.1 PRO Subscription

- Energy Prediction Output Tool:
    - Functionality: Provides users with predictions related to energy consumption and efficiency.
    - Recommendations: Based on input features, the system generates tailored recommendations to optimize energy usage.
    - Cooling System Analysis:

- Types of Cooling Systems: Includes analysis for various cooling systems such as:
  - Dry Cooling: Evaluates the efficiency and effectiveness of dry cooling systems.
  - Once Through: Analyzes the performance of once-through cooling systems.
  - Recirculating: Assesses recirculating cooling systems based on the type of power plants used.

2.1.4.2 BUSINESS Subscription

- Energy Recommendation System:
  - Functionality: Offers users optimal energy solutions tailored to their specific needs.
  - Recommendations: The system recommends the top optimal solutions based on several input features, including:
    - Energy Percentages: Provides insights into the distribution of energy sources.
    - Estimated Cost: Offers an estimate of the costs associated with the recommended solutions.
    - Total Energy: Calculates the total energy required based on user inputs.
    - Potential Savings: Identifies potential savings that can be achieved through recommended actions.
    - Efficiency Rating: Displays the efficiency rating (in percentage) of the proposed solutions.
    - Environmental Impact: Assesses the environmental impact of the recommended energy solutions.
    - Main Energy Source: Identifies the primary energy source based on the user's input features, including budget constraints, energy requirements, and country name.

These subscription plans are designed to empower users with advanced tools and insights, enabling them to make informed decisions regarding energy consumption, efficiency, and sustainability.

2.1.5 Blogs Section

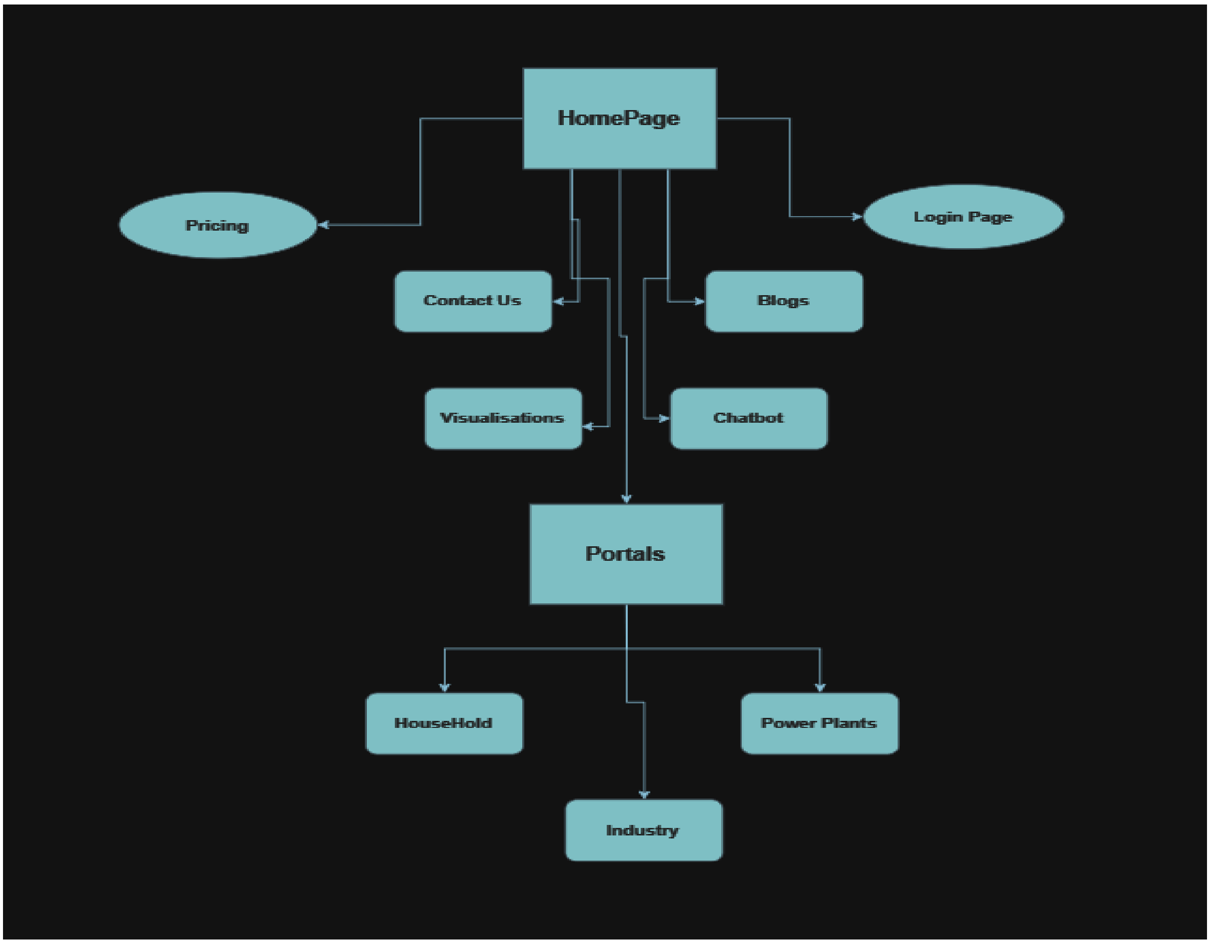
- **Functionality:** The Enigma platform features a dedicated Blogs section that provides users with two interactive options: Summarize and FAQ.
    - **Summarize:** When users click on the "Summarize" button for a specific blog, the system generates a concise summary of the blog content using a Large Language Model (LLM) approach.
    - **FAQ:** By clicking on the "FAQ" button, users can access a list of frequently asked questions related to the blog, also generated through the LLM.
- **LLM-Based Approach:** Both the summarization and FAQ generation functionalities leverage advanced LLM technology to ensure that users receive accurate and relevant information based on the blog content.

2.1.6 Chatbot Feature

- **Chatbot Development:** The platform has integrated a chatbot using the Chainlit interface.
    - **Functionality:** The chatbot provides users with an interactive way to ask questions and receive instant responses, enhancing user engagement and support.
    - **Capabilities:** Users can inquire about various topics related to energy efficiency, subscription plans, and other features of the Enigma platform, making it a valuable resource for immediate assistance.

These features are designed to enhance user experience by providing quick access to information and support, thereby promoting engagement with the Enigma platform.

2.2 Block Diagram



(Fig 2.2)

2.3 System Requirements

The Enigma platform employs various models and technologies to deliver its functionalities effectively. Below are the detailed system requirements based on the functionalities described:

2.3.1. Efficiency Model

- **Model Used**: Random Forest Classifier
- **Purpose**: To classify devices as efficient or inefficient based on input features such as type, brand, age, usage hours, and malfunction history.

2.3.2. Carbon Footprint Calculator

- **Model Used**: Multiple Linear Regression (MLR)
- **Purpose**: To predict carbon emissions based on a wide range of lifestyle factors, including body type, diet, transportation mode, and energy usage.

2.3.3. Energy Prediction

- **Model Used**: Multiple Linear Regression (MLR)
- **Purpose**: To predict energy output based on input features and generate recommendations for optimizing energy usage.
- **Frontend Integration**: Charts and data visualizations are fetched and displayed in the frontend using Flask for cooling system analysis.

2.3.4. Energy Recommendation System

- **Algorithm Used**: Genetic Algorithm (DEAP)
- **Purpose**: To generate top optimal energy solutions based on user-defined constraints and requirements.

2.3.5. Summary and FAQ Generation

- **Model Used**: LLaMA 3 (70B, 8192)
- **Purpose**: To generate concise summaries and frequently asked questions from blog content.
- **Interface**: Integrated with ChatGroq and Flask for a user-friendly interface.

2.3.6. Live Trends Visualization

- **Technology Used**: Charts.js
- **Purpose**: To display live trends of user login activity or industry metrics.
- **Data Storage**: Data is stored in the backend using XAMPP Server for efficient retrieval and display.

Summary of System Requirements

- **Machine Learning Models**:

- Random Forest Classifier for efficiency classification.
- Multiple Linear Regression (MLR) for carbon emission prediction and energy output prediction.
- Genetic Algorithm (DEAP) for optimal energy solutions.

- **User Interface**:
  - Flask framework for backend integration and serving the frontend.
  - ChatGroq for chatbot functionality and interaction.
  - Leaflet and GeoJSON for country mapping in frontend
- **Data Visualization**:
  - Charts.js for displaying live trends and data visualizations.
- **Data Storage**:
  - XAMPP Server for backend data management and storage.

These system requirements ensure that the Enigma platform effectively delivers its functionalities, providing users with comprehensive energy insights, carbon footprint calculations, energy predictions, and interactive visualizations to support informed decision-making for energy efficiency and sustainability.

2.4 Dataset Description

Model 1: Efficiency Assessment and Electricity Price Calculator Tool

- **Dataset Features:**
  - **Country Name:** The country where the device is used.
  - **Brand Name:** The brand of the device.
  - **Energy Consumption (in kWh):** The amount of energy the device uses per unit time.
  - **Device Type:** The type or category of the device.

  - **Usage Hours Per Day:** The number of hours the device is used each day.
  - **Device Age (in months):** The age of the device in months.
  - **Number of Malfunction Incidents:** The count of malfunction incidents reported for the device.
- **Output Features:**
  - **Efficiency Assessment:** Evaluation of the device's efficiency.
  - **Electricity Price:** The cost of electricity for the device.

Model 2: Individualized Carbon Footprint Calculator

- **Dataset Features:**
  - **Body Type:** The user's body type (e.g., overweight, obese, underweight, normal).
  - **Gender:** The user's gender (male or female).
  - **Diet:** The type of diet followed by the user (e.g., pescatarian, vegetarian, omnivore, vegan).
  - **How Often Shower:** Frequency of showering (e.g., daily, less frequently, more frequently, twice a day).
  - **Heating Energy Source:** Source of energy for heating (e.g., coal, natural gas, wood, electricity).
  - **Transport:** Type of transport used (e.g., public, walk/bicycle, private).
  - **Vehicle Type:** Type of vehicle used (e.g., petrol, diesel, hybrid, LPG, electric).
  - **Frequency of Traveling by Air:** How often the user travels by air (e.g., frequently, rarely, never, very frequently).
  - **Vehicle Monthly Distance Km:** Distance traveled by vehicle per month.
  - **Waste Bag Weekly Count:** Number of waste bags disposed of weekly.
  - **How Long TV/PC Daily Hour:** Time spent on TV or PC daily.

- How Long Internet Daily Hour: Time spent using the internet daily.
- **Whether Using Energy Efficient Devices?:** Use of energy-efficient devices (e.g., No, Sometimes, Yes).
- **Cooking_With:** Type of cooking fuel or method used.

- **Output Features:**
  - **Predicted Carbon Emission:** Estimated carbon emissions based on input features.

Model 3: Energy Consumption Predictor and Efficiency Rating Calculator

- **Dataset Features:**
  - **Ambient Temperature (Celsius degrees):** The temperature of the environment.
  - **Ambient Pressure (Hectopascals (hPa)):** The pressure of the environment.
  - **Relative Humidity (Percentage):** The humidity level of the environment.
  - **Exhaust Vacuum (Millimeters of Mercury (mmHg)):** The vacuum level in the exhaust.
  - **Actual Energy Value:** The actual amount of energy used or produced.
- **Output Features:**
  - **Predicted Energy Output:** Estimated energy output based on input features.
  - **Efficiency Rating (in %):** The efficiency rating of the system in percentage.

Model 4: Energy Recommendation System [2]

- **Dataset Features:**
  - **Country:** The country for which energy recommendations are being made.
  - **Wind Consumption and Price:** Data on wind energy consumption and price.
  - **Oil Consumption and Price:** Data on oil energy consumption and price.
  - **Hydro Consumption and Price:** Data on hydro energy consumption and price.

  - **Solar Consumption and Price:** Data on solar energy consumption and price.
  - **Bioenergy Consumption and Price:** Data on bioenergy consumption and price.
- **Output Features:**
  - **Recommended Energy Solutions:** Optimal energy recommendations based on the input features.

3. SYSTEM DESIGN

3.1. System Architecture

The system architecture for the Enigma platform is designed to provide a robust framework that supports its functionalities while ensuring scalability, security, and user engagement. Below are the key components and their interactions:

3.1.1 Architecture Overview

- **Client-Server Architecture**: The Enigma platform operates on a client-server model, enabling users to interact with the server to access various services and functionalities.
- **Modular Design**: The architecture is modular, allowing for easy updates and maintenance of individual components without affecting the overall system.

3.1.2 Components of the Architecture

3.1.2.1 User Interface

- **Web Interface**: Built using React and Streamlit to provide a responsive and user-friendly experience.
- **Input Forms**: Users can input details about household devices, lifestyle factors, and URLs for summarization.

3.1.2.2 Backend Services

- **Flask Framework**: Serves as the backend framework to handle API requests and manage communication between the frontend and various services.
- **Data Processing Modules**:
    - **LLM Processing**: Leverages LLaMA 3 (70B, 8192) and ChatGroq for generating summaries and FAQs.
    - **Efficiency Model**: Implements a Random Forest Classifier to classify device efficiency.
    - **Carbon Footprint Calculator**: Uses Multiple Linear Regression (MLR) to predict carbon emissions based on user inputs.
    - **Energy Prediction**: Employs MLR to predict energy output and generate recommendations.
    - **Energy Recommendation System**: Utilizes a genetic algorithm (DEAP) to find optimal energy solutions.

3.1.2.3 Data Storage

- **Database Management**: SQLite3 is used for storing user data, extracted text, metadata, and interaction history.
- **XAMPP Server**: Manages backend data storage for user login trends and other analytics.

3.1.2.4 Visualization Tools

- **Charts.js**: Integrated for displaying live trends.
- **Dashboards**: Three dashboards cater to different sectors (Household, Power Plants, Industry) and provide insights on energy consumption and efficiency.

3.1.3 Data Flow

- **User Input**: Users input data through the web interface, which is sent to the backend for processing.
- **Data Processing**: The backend processes the data using various models and algorithms, generating outputs such as efficiency ratings, carbon footprint estimates, and energy predictions.
- **Output Generation**: Results are sent back to the frontend, where they are displayed to the user in a clear and interactive manner.
- **Visualization**: Data is visualized using Charts.js, providing users with insights into their energy usage and trends.

3.1.4 Security and Performance

- **Security Measures**: Implementations include user authentication, data encryption, and secure API endpoints to protect user data and interactions.
- **Performance Optimization**: Caching mechanisms and efficient data retrieval strategies are employed to enhance the responsiveness of the platform.

This system architecture ensures that the Enigma platform effectively delivers comprehensive energy insights and tools while maintaining a user-friendly experience and robust backend support.

3.2. Module Design

3.2.1 Energy Efficiency Insights Module

- **Functionality:**
    - **Input Management:** Allows users to input details about their household devices, including type, brand, age, usage hours, and malfunction history.
    - **Efficiency Evaluation:** Analyzes device efficiency based on brand performance, typical energy consumption patterns, and the impact of malfunctions.

  - **Cost Calculation:** Computes the cost per day to operate each device based on local energy rates, device efficiency, and usage.
- **Components:**
  - **Device Input Form:** A form for users to enter device details.
  - **Efficiency Analysis Engine:** Backend service for evaluating device efficiency and calculating costs.
  - **Results Display:** Interface for presenting efficiency evaluations and cost calculations to users.

3.2.2 Carbon Footprint Calculator Module

- **Functionality:**
  - **Data Collection:** Gathers information on lifestyle factors such as body type, sex, diet, shower frequency, transportation mode, air travel, waste production, and electronic device usage.
  - **Emission Calculation:** Estimates the user's daily carbon emissions in kilograms.
  - **Recommendations Engine:** Provides personalized recommendations to reduce the user's carbon footprint based on their profile.
- **Components:**
  - **Input Form:** Interface for users to enter lifestyle data.
  - **Emission Calculation Engine:** Backend service for calculating carbon emissions.
  - **Recommendation Engine:** System for generating and displaying personalized recommendations.

3.2.3 Visualization Tools Module

- **Functionality:**
  - **Dashboards:** Provides sector-specific insights through three distinct dashboards:

        - **Household Dashboard:** Insights on household energy consumption and efficiency.
        - **Power Plants Dashboard:** Metrics and performance data for power plants.
        - **Industry Dashboard:** Analysis of international finance, renewable energy consumption, and industry-specific data.
    - **Charts in Services Section:**
        - **Ozone Concentration:** Visualization of ozone pollution levels by region.
        - **Decadal Anomalies:** Displays temperature and climate anomalies over decades.
        - **Deaths Due to Natural Disasters:** Shows the impact of natural disasters on human life.
        - **CO2 Emissions Worldwide:** Overview of global carbon dioxide emissions.
        - **Pollution and Emission:** Tracks pollution trends and emissions from various pollutants.
- **Components:**
    - **Dashboard Interfaces:** Separate interfaces for each dashboard (Household, Power Plants, Industry).
    - **Charts and Graphs:** Interactive charts and graphs for each service section.
    - **Data Aggregation Service:** Backend service for aggregating and processing data for visualizations.

3.2.4 Subscription Plans Module

- **Functionality:**
    - **PRO Subscription:**

        - **Energy Prediction Output Tool:** Predicts energy consumption and efficiency.
        - **Cooling System Analysis:** Evaluates various types of cooling systems.
    - **BUSINESS Subscription:**
        - **Energy Recommendation System:** Provides optimal energy solutions, cost estimates, potential savings, and environmental impact.

- **Components:**
    - **Subscription Management Interface:** Allows users to select and manage subscription plans.
    - **Prediction and Analysis Engines:** Backend services for energy prediction and cooling system analysis (PRO) and energy recommendations (BUSINESS).

3.2.5 Blogs Section Module

- **Functionality:**
    - **Summarize:** Generates concise summaries of blog content using LLM technology.
    - **FAQ:** Generates frequently asked questions related to blog content using LLM technology.
- **Components:**
    - **Blog Interface:** Displays blog content with Summarize and FAQ buttons.
    - **LLM Integration:** Backend service for generating summaries and FAQs using LLM.

3.2.6 Chatbot Feature Module

- **Functionality:**
    - **Interactive Chatbot:** Provides users with instant responses to questions related to energy efficiency, subscription plans, and other platform features.
- **Components:**

- **Chatbot Interface:** Integrated into the platform using the Chainlit interface.
- **Chatbot Backend:** Service for handling user queries and providing responses.

3.2.7 Login Trends Visualization Module

- **Functionality:**
  - **Trends Display:** Shows live trends of user login activity or industry metrics.
- **Components:**
  - **Charts.js Integration:** For creating interactive data visualizations.
  - **Data Storage:** User login data stored and managed using XAMPP Server.
  - **Trends Dashboard:** Interface for displaying login trends and metrics.

3.3. Database Design

3.3.1 Table Structure

- **Users Table:** Stores information about users, including their personal details, login activities, and account statistics.

(Table 3.3.1.1)

Column Name	Data Type	Description
`id`	Integer	Unique identifier for each user.
`name`	String	Name of the user.
`email`	String	Email address of the user.
`password`	String	Password for user authentication.
`logon_count`	Integer	Represents the number of times the user has logged in.
`logins`	Array of DateTime	Records the date and time of each login event for the user.

- **Transaction_pro Table:** Keeps records of transactions for PRO subscriptions, capturing details like cardholder information and subscription status.

(Table 3.3.1.2)

Column Name	Data Type	Description
`id`	Integer	Unique identifier for each transaction.
`cardholder_name`	String	Name of the cardholder.
`cardnumber`	String	Credit card number associated with the transaction.
`email`	String	Email address of the cardholder.
`pro`	TinyInt	Indicates that the user has taken a Pro subscription (e.g., 1 for Pro). There are no other levels.

- **Transaction_premium Table:** Maintains records of transactions for PREMIUM subscriptions, including cardholder information and subscription status.

(Table 3.3.3)

Column Name	Data Type	Description
`id`	Integer	Unique identifier for each transaction.
`cardholder_name`	String	Name of the cardholder.
`cardnumber`	String	Credit card number associated with the transaction.
`email`	String	Email address of the cardholder.
`premium`	TinyInt	Indicates that the user has taken a Premium subscription (e.g., 1 for Premium). There are no other levels.

3.3.2. Data Flow Diagram

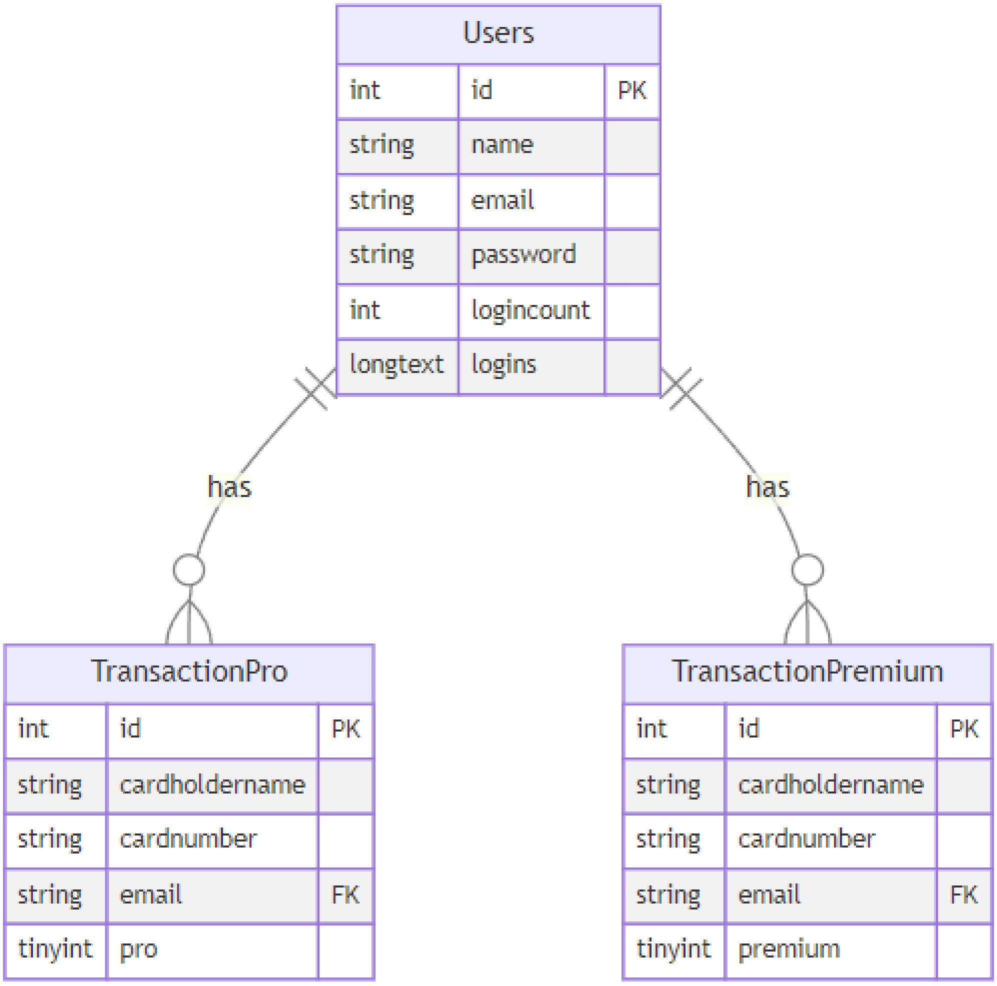
```
A[User Registration & Login Inputs] --> B[User Profile (users table)]
B --> C[Track Logins (Logon Count, Login Times)]
C --> D[Login Trends Data]

A --> E[Subscription Inputs (PRO & PREMIUM)]
E --> F[PRO Subscription Table (transaction_pro)]
E --> G[PREMIUM Subscription Table (transaction_premium)]
```

(Fig 3.3.2)

In this diagram:

- **A** represents the user registration and login inputs.
- **B** updates the users table.
- **C** tracks logins, updating logon count and login times.
- **D** produces the login trends data.
- **E** represents subscription inputs, updating the transaction_pro and transaction_premium tables.

3.3.3. ER Diagram



(Fig 3.3.3)

3.4 Tools Used

- **Frontend Development:** HTML, CSS, SCSS Bootstrap JS, and Bootstrap CSS were utilized to design and build the website's user interface. HTML, SCSS and CSS provided the structure and style, while Bootstrap JS and Bootstrap CSS enhanced the design with responsive and interactive components. Leaflet and GeoJSON was used for country mapping in Industry model.

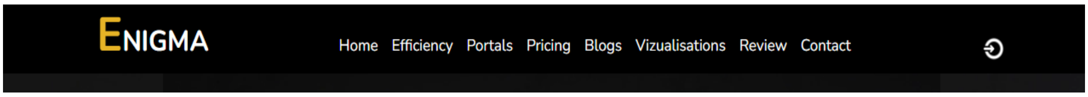
- **Model Integration and UI:** Flask was employed to integrate predictive and analytical models into the website. It facilitated the connection between the user interface and the backend models, enabling dynamic functionality and seamless user interactions.
- **Backend Connectivity:** phpMyAdmin and XAMPP were used for managing and connecting to the backend database. phpMyAdmin provided a graphical interface for database management, while XAMPP served as a local server environment for backend operations.
- **Visualization:** Google Charts and Tableau were used to create interactive and informative visualizations. Google Charts offered customizable charting options for real-time data representation, while Tableau provided advanced data analysis and visualization capabilities.

3.4.1 User Interface Screen Design

The UI presents a very interactive and user friendly design that aims to attract the user to use the website more. It uses bootstrap, scss and css technologies for styling. Various pop-up modals are implemented by leveraging the capabilities of bootstrap.js and popper.js.

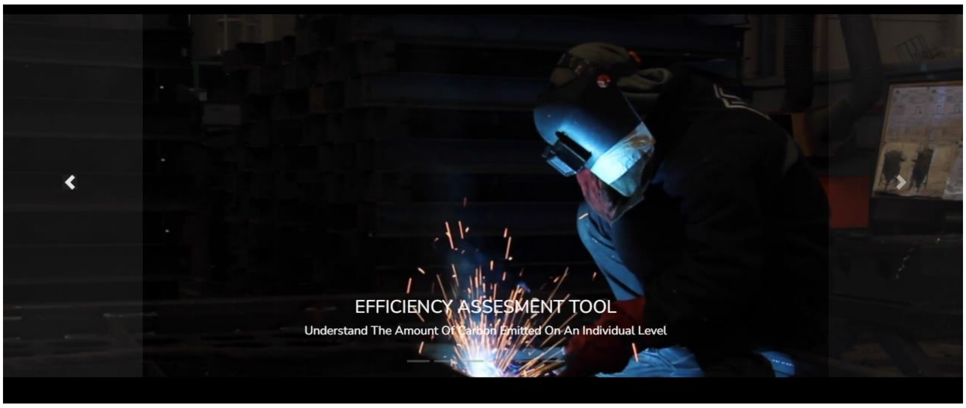
The UI includes:

- The Header section which will help in navigating to various sections of the website without scrolling. It also includes the path for pricing tab from where user can purchase any plan as per their wish. Login/Logout button is also included here.



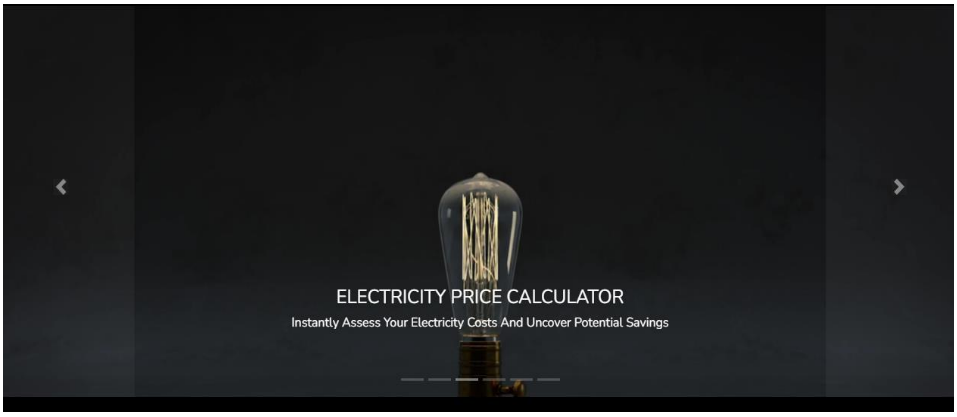
(Fig3.4.1.1)

- A carousel video bar aimed at providing a visual representation of what we provide with our website and link navigating to it. Only top functionalities are displayed here. Slider is included for manual flow of carousel bar.



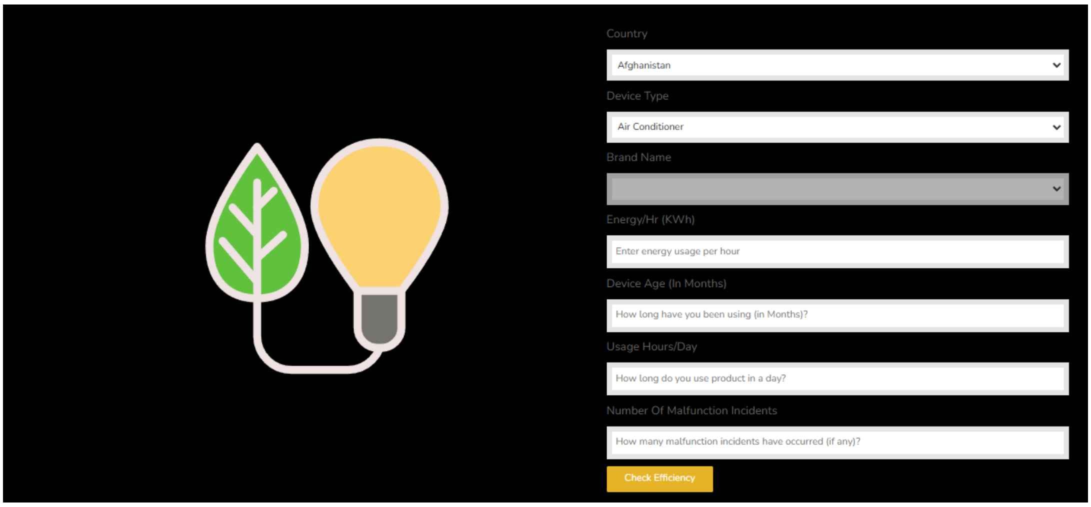
(Fig3.4.1.2)

(Fig3.4.1.3)



(Fig3.4.1.4)

- Device Efficiency Calculator

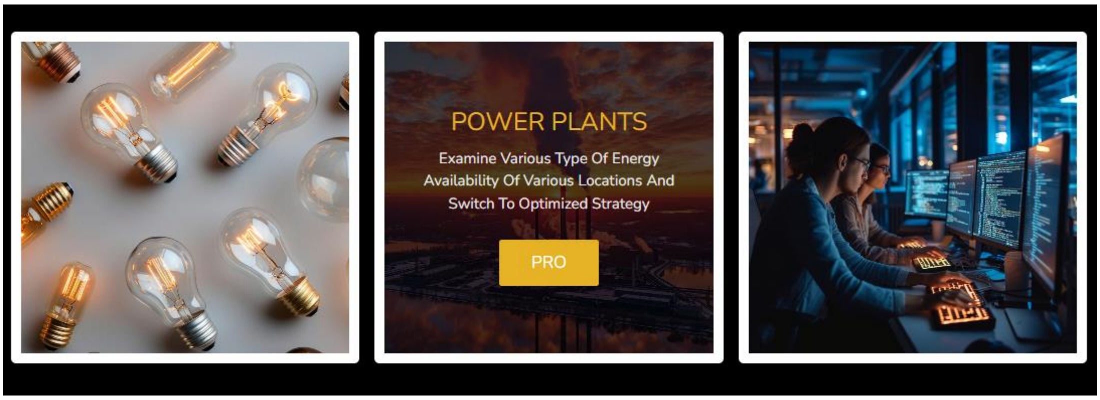


(Fig3.4.1.5)

- Portals Section and pop-up modals on clicking redirecting you to specific functionality

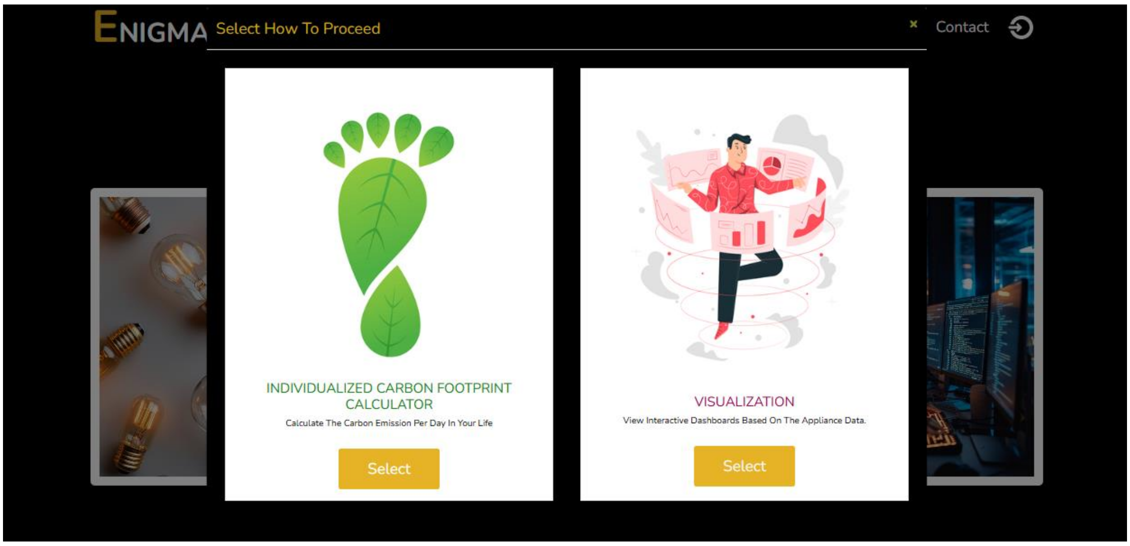
(Fig3.4.1.6)

On hover View



(Fig3.4.1.7)

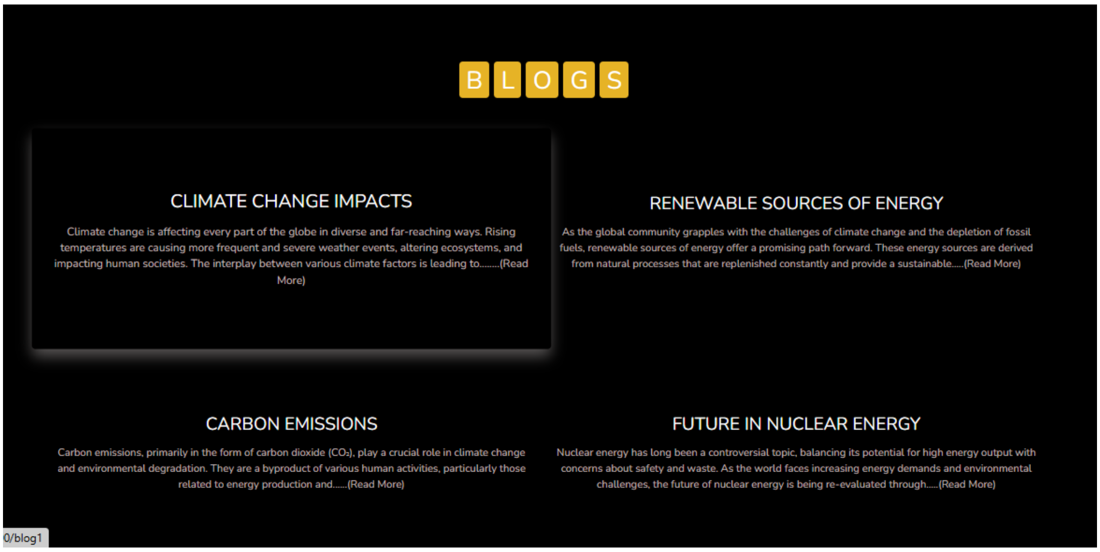
Pop-up Modal



(Fig3.4.1.8)

(All key functionalities are redirected from here)

- Chatbot section which redirects to a new screen where user can interact with a bot that helps in doubts regarding website and energy conservation.
- Blogs section within interactive blocks

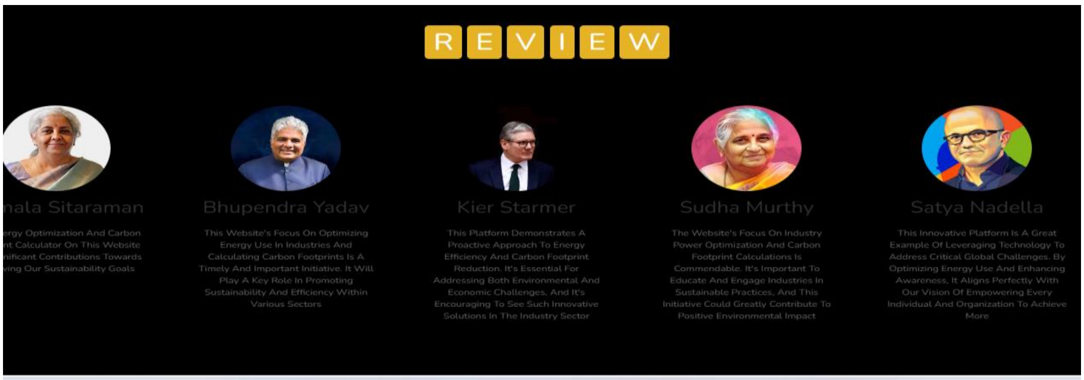


(Fig3.4.1.9)

- Visualization section with portals to respective dashboards and analysis

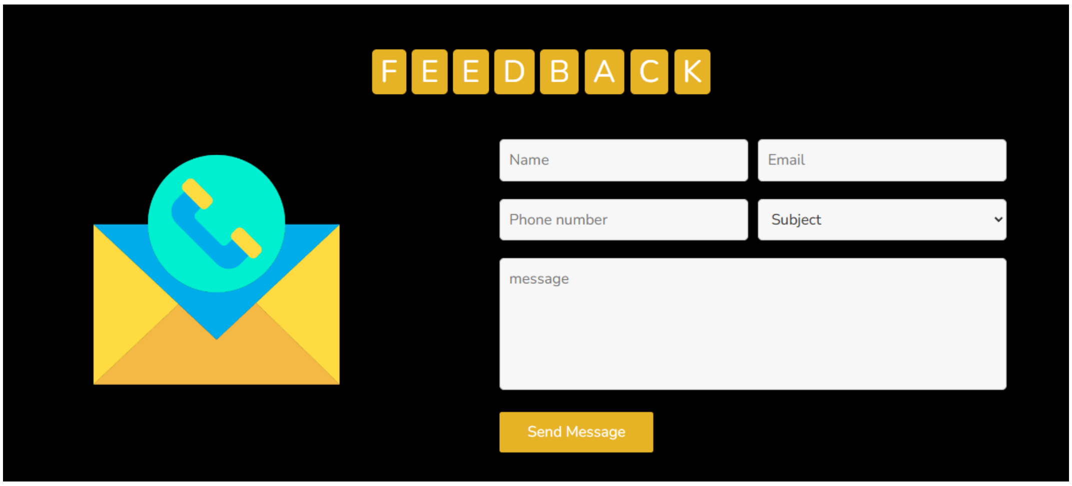
(Fig3.4.1.10)

- Review section by credible individuals to enhance user trust, given by a slider



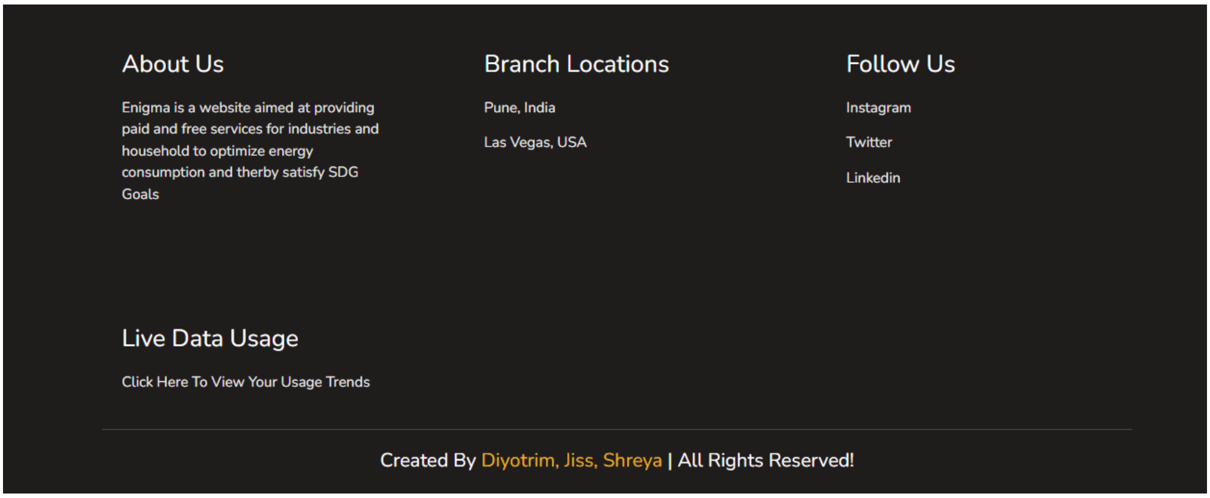
(Fig3.4.1.11)

- Feedback section to raise a complaint, ask a query or give a review



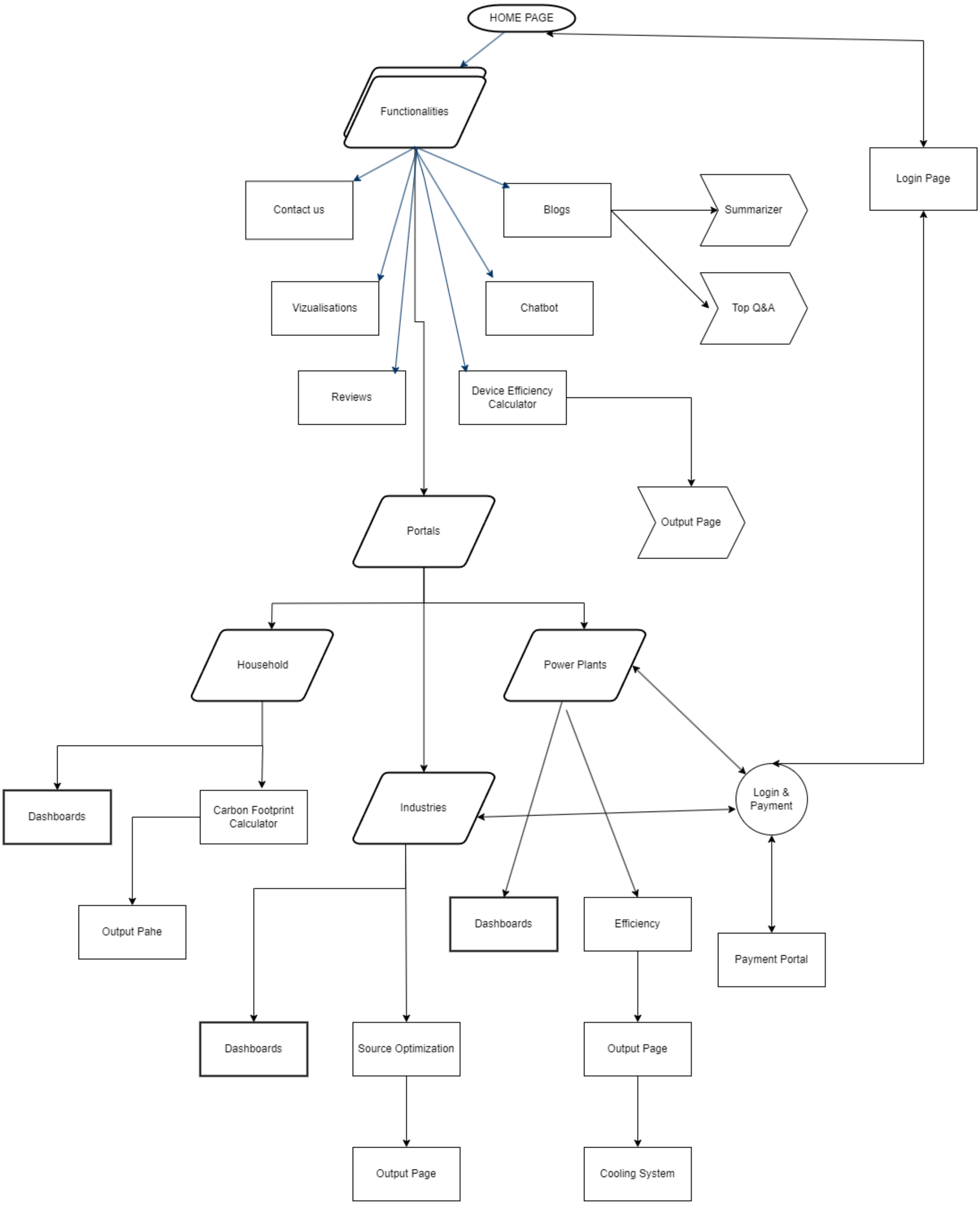
(Fig3.4.1.12)

- Footer section to detail about us and methods to reach to us. Also a trend page to view count of logins by the respective user.



(Fig3.4.1.13)

3.4.2 Application flow Diagram



(Fig3.4.1.2)

4. IMPLEMENTATION

4.1. Coding Standard

Coding standards are a set of guidelines and best practices that are designed to ensure stability, readability, maintainability and quality in software development. They cover various aspects of coding, including naming conventions, code formatting, documentation, error handling and structure. Adhering to coding standards helps developers write code that is easier to understand and work with, less prone to errors and easier to integrate and scale. By maintaining a consistent approach, teams can improve code quality, increase productivity and make it easier for new developers to join the project.

4.1.1 Package management

Default method: Use pip install to manage Python dependencies and apt-get install for system packages. Use the -q flag to minimize output during installations. For JavaScript tools, use npm install for local tools like localtunnel.

Best practices: Organize packages by type (e.g., web frameworks, data analysis libraries). Ensure all required libraries are installed and up to date. Specify exact versions in requirements.txt or package.json to avoid compatibility issues.

4.1.2 Database Management

Standard practice: Define the tables with the appropriate constraints (eg, UNIQUE, NOT NULL). Perform design, display, and manipulation tasks.

Best practices: Make database routines reusable and static. Manage discrepancies and ensure data integrity. Close data connections properly to avoid leaks and ensure efficient resource management.

4.1.3 Handling files

Standard methods: Use libraries such as Pandas for CSV and Excel files, and Pillow for image files. Manage different file types with specialized functions and delete temporary files after processing.

Best practices: Implement error handling for invalid or corrupt files. Apply file protection techniques and manage large files by streaming or chunking as necessary. Ensure proper file permissions and avoid exposing sensitive files.

4.1.4 Word processing

Standard method: Store and manage processed data using environment variables for sensitive information (e.g., API keys).

Best practices: Validate input data and handle cases of missing or incomplete data. Ensure text processing routines are robust and up to date. Regularly update language models and processing tools to incorporate improvements.

4.1.5 Program Description

Standard Method: Develop the website using Flask, incorporating features such as user authentication (registration, login, account management) and file handling. Implement a responsive and accessible UI.

Best practices: Customize Flask features (e.g., routing, templating) to enhance user experience. Manage user access securely and ensure proper authentication and authorization. Provide meaningful error messages and handle exceptions gracefully.

4.1.6 Web Maintenance

Default method: Regularly check for broken links and update content as necessary. Use paging for compatibility with different screen resolutions.

Best practices: Ensure compliance with website Terms of Service and privacy policies. Implement fault management for network issues and perform regular troubleshooting and updates to maintain web performance and security.

4.1.7 General coding tasks

Code Structure: Organize tasks to focus on individual tasks. Put the main logic in a main function. Use global variables like st.session_state to store user state.

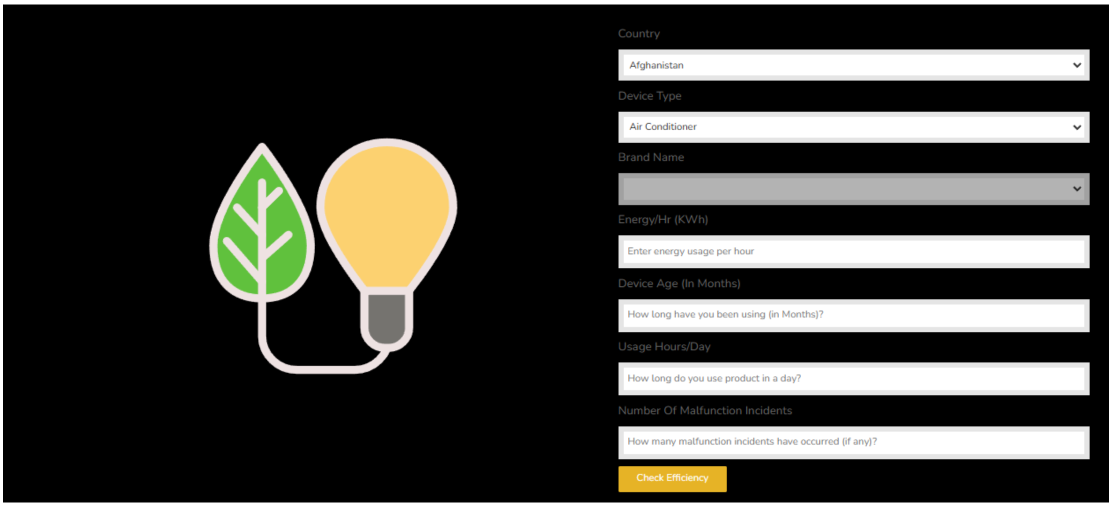
Notes and Comments: Use comments to explain difficult parts of the code. Text strings should clearly describe their purpose and parameters.

Security and protection: Avoid encrypting sensitive information. Use environment variables. Update libraries and tools regularly for security and compatibility. Storing Passwords as hashes for user protection.

4.2. Screen Shots of Functionalities

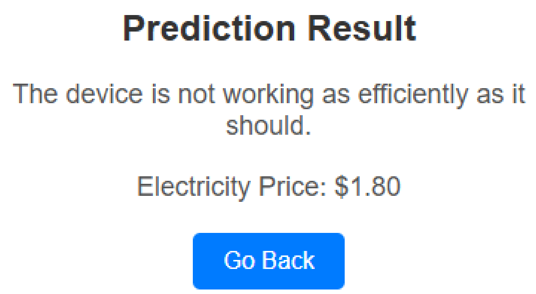
4.2.1 Device Efficiency model

Input Page:



(Fig 4.2.1.1)

Output Page:

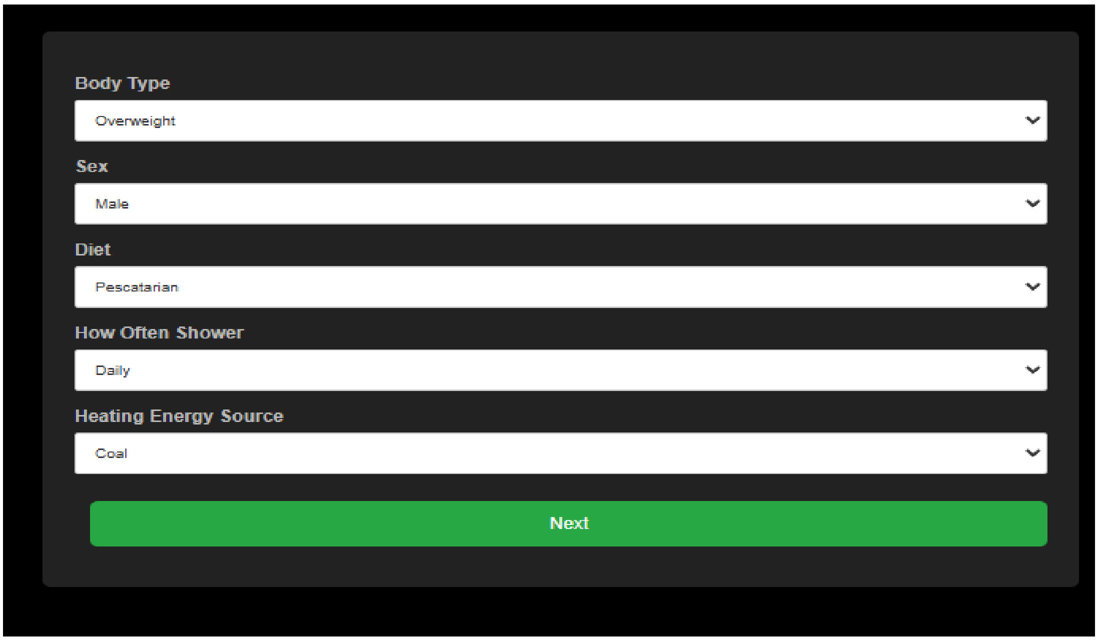


(Fig 4.2.1.2)

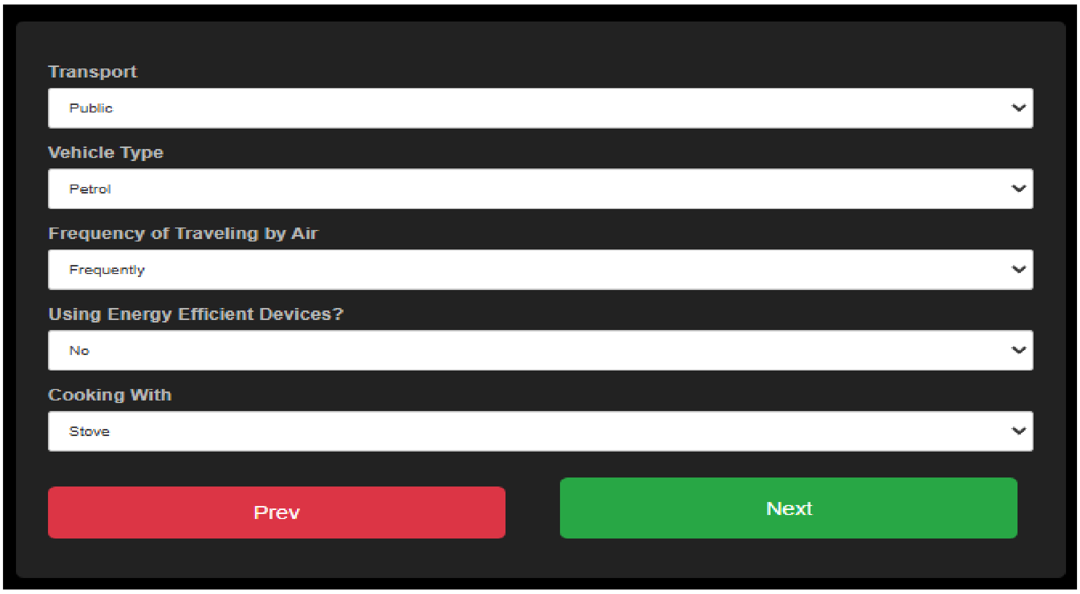
Still subjective to styling

4.2.2 Carbon Footprint Calculator

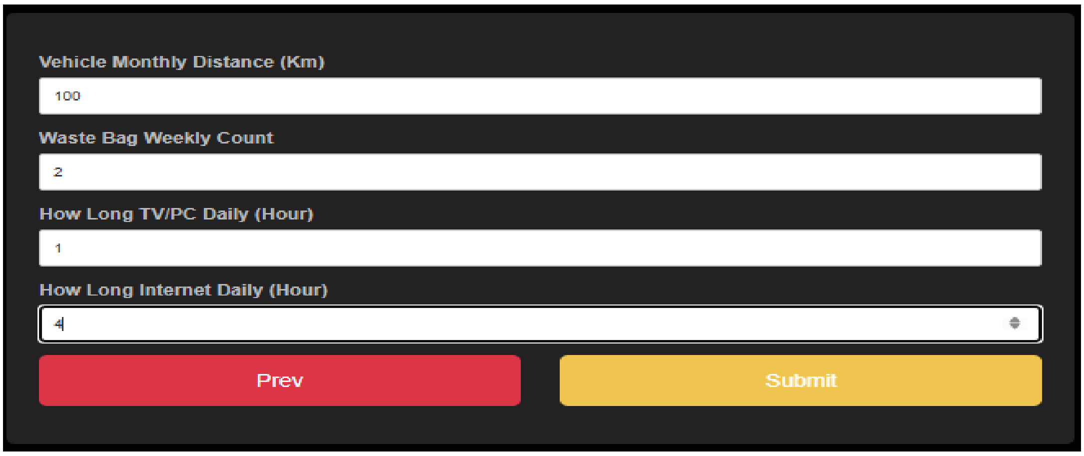
Input Page:



(Fig 4.2.2.1)

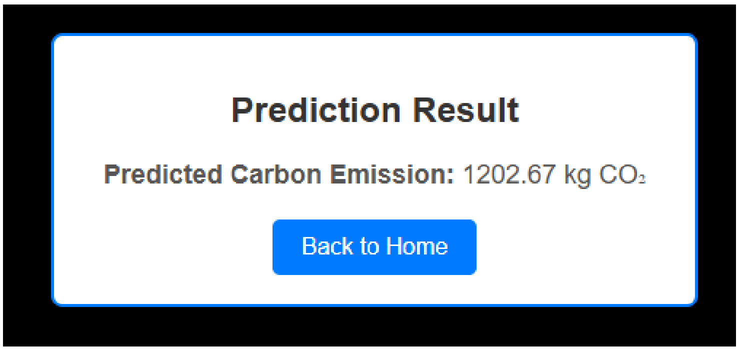


(Fig 4.2.2.2)



(Fig 4.2.2.3)

Output Page:

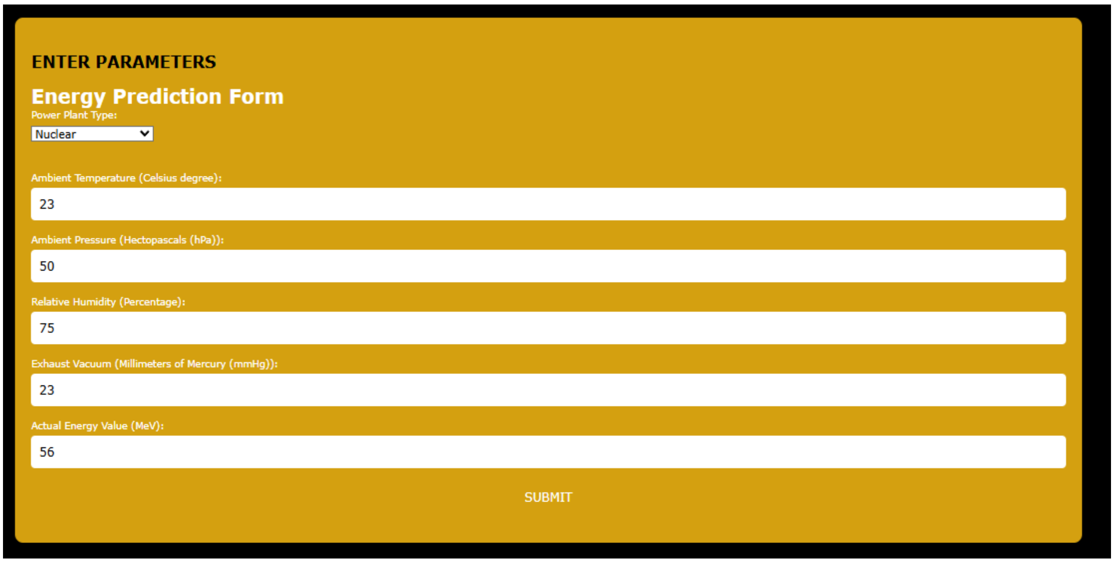


(Fig 4.2.2.4)

Still subjective to styling

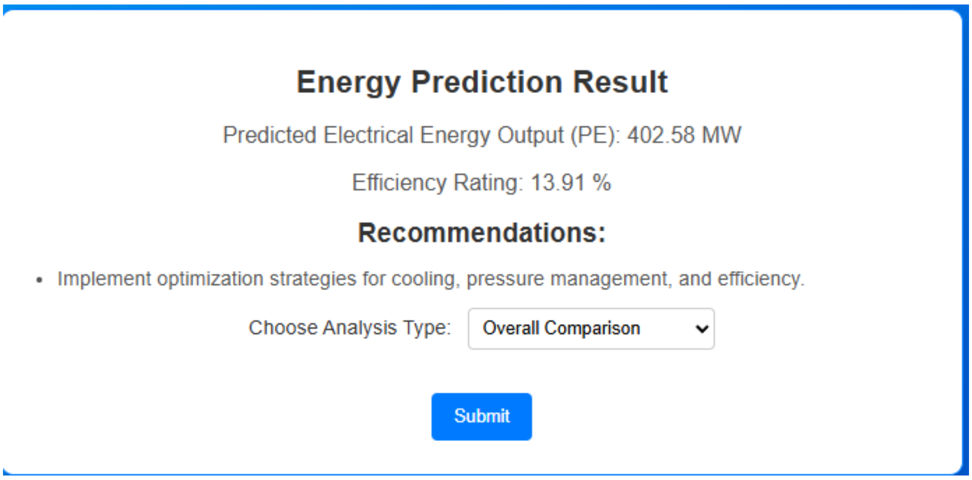
4.2.3 Power Plant Modal

Input Page:



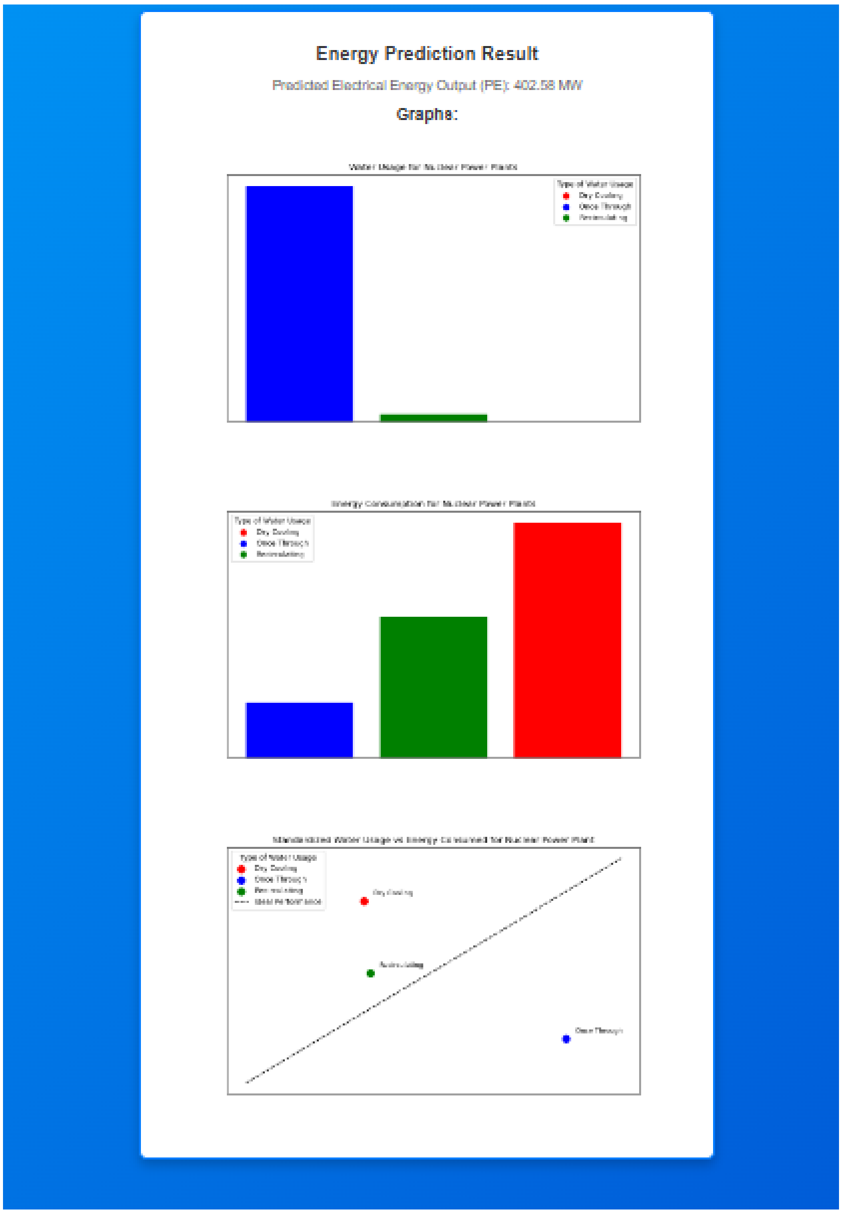
(Fig 4.2.3.1)

Efficiency and Recommendation Output Page:



(Fig 4.2.3.2)

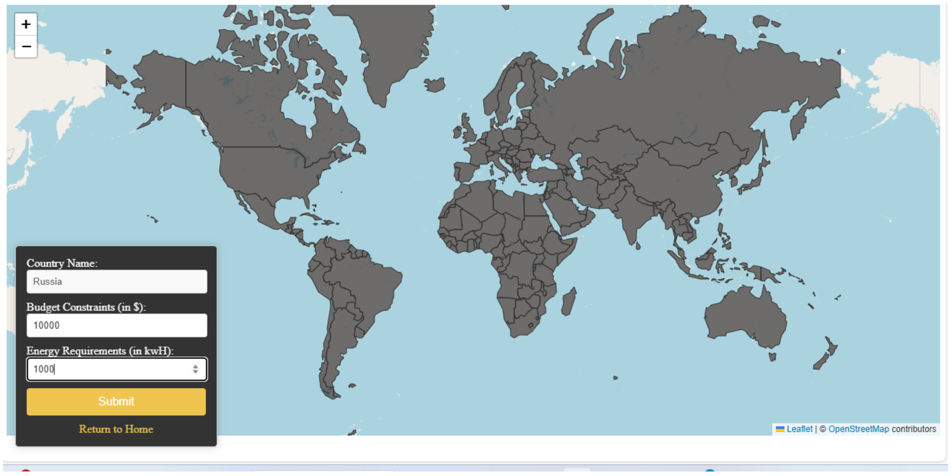
Cooling Recommendation Analytics:



(Fig 4.2.3.3)

4.2.4 Industry Energy Source Optimizer Portal

Input Page:

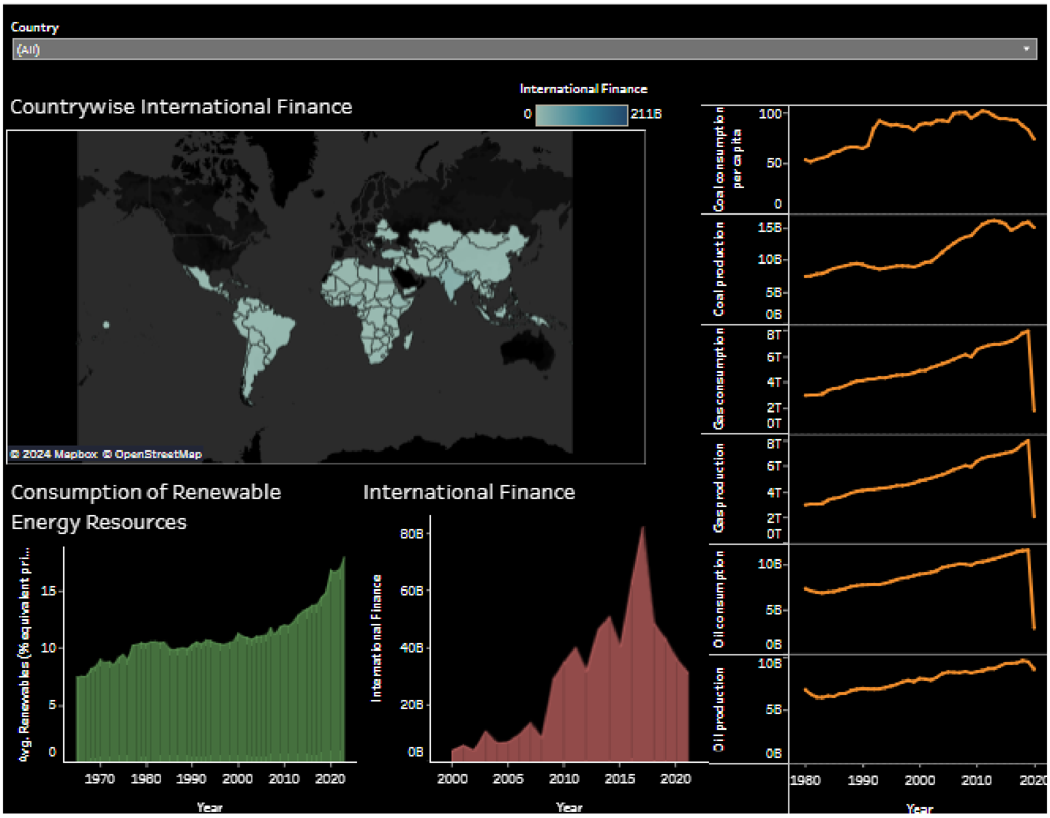


(Fig 4.2.4.1)

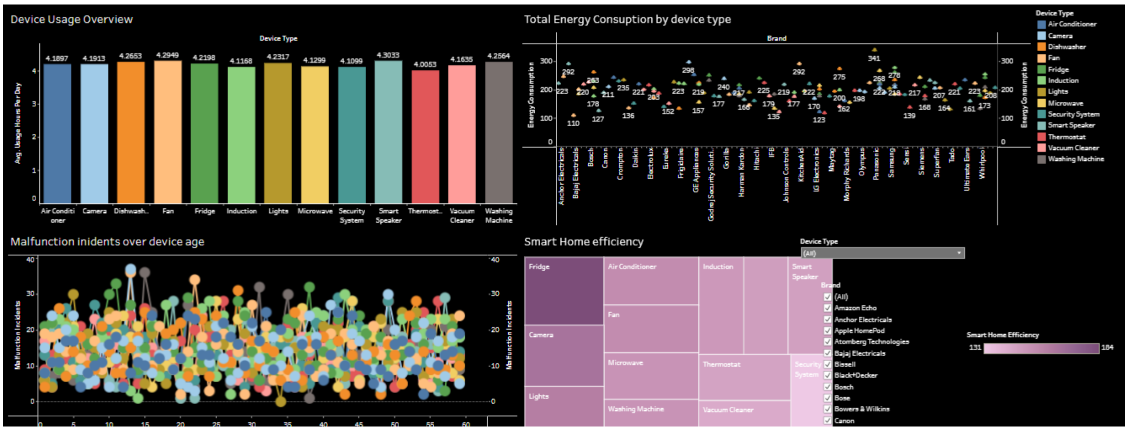
Recommendation Page:

(Fig 4.2.4.2)

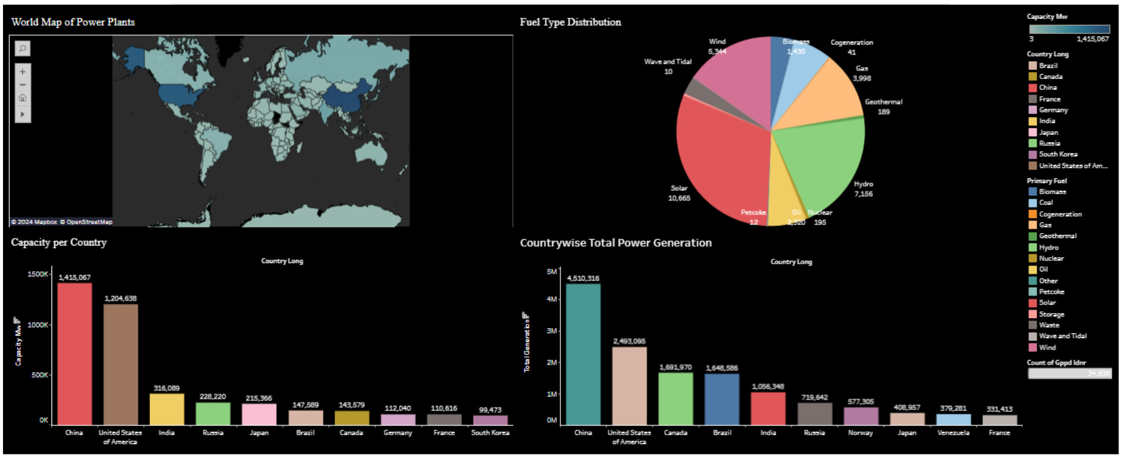
4.2.5 Interactive Dashboards



(Fig 4.2.5.1)

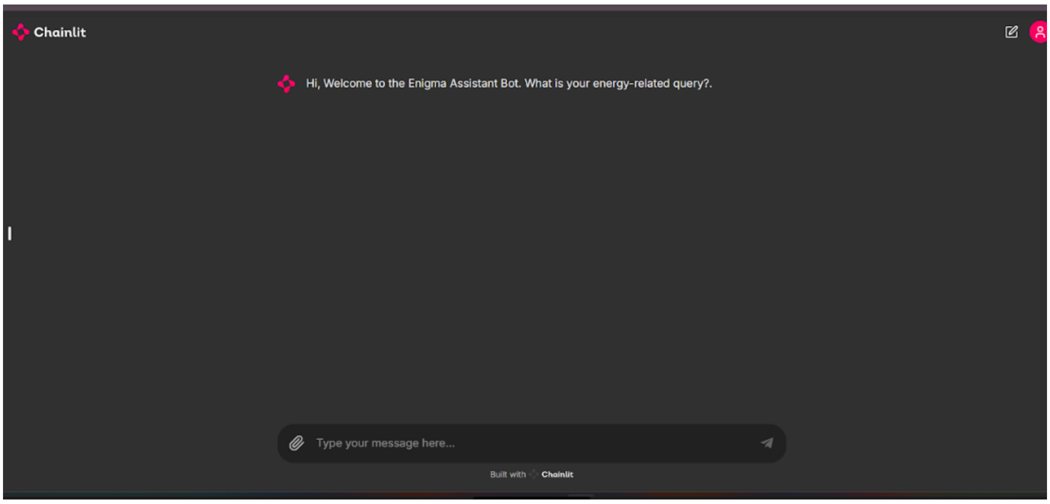


(Fig 4.2.5.2)



(Fig 4.2.5.3)

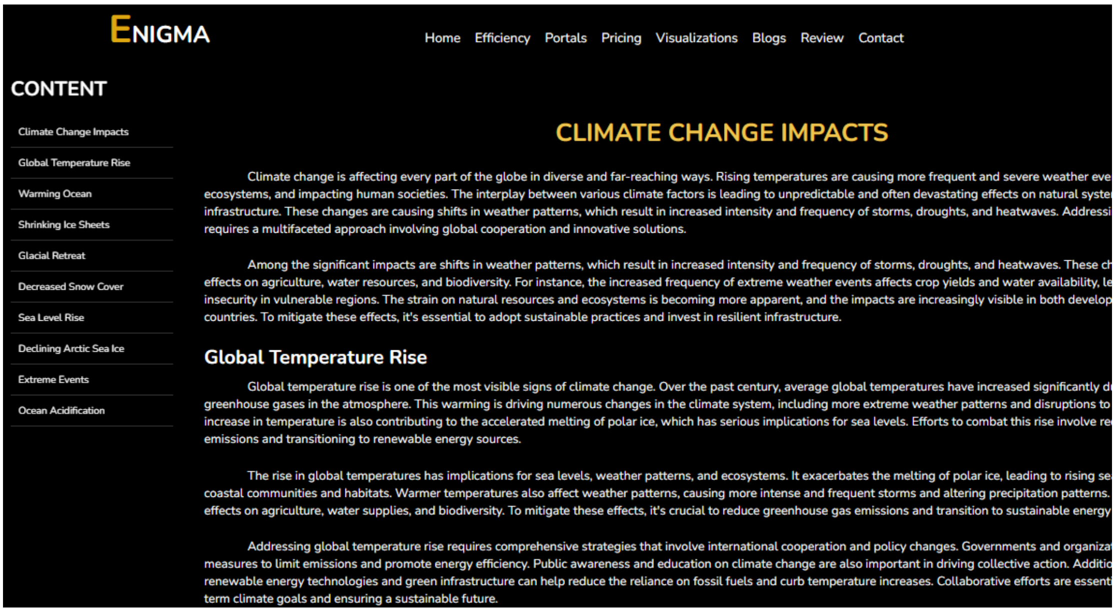
4.2.6 Chatbot



(Fig 4.2.6)

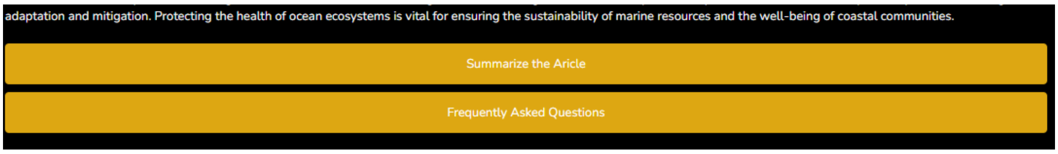
4.2.7 Blogs

Home Page of Blogs:



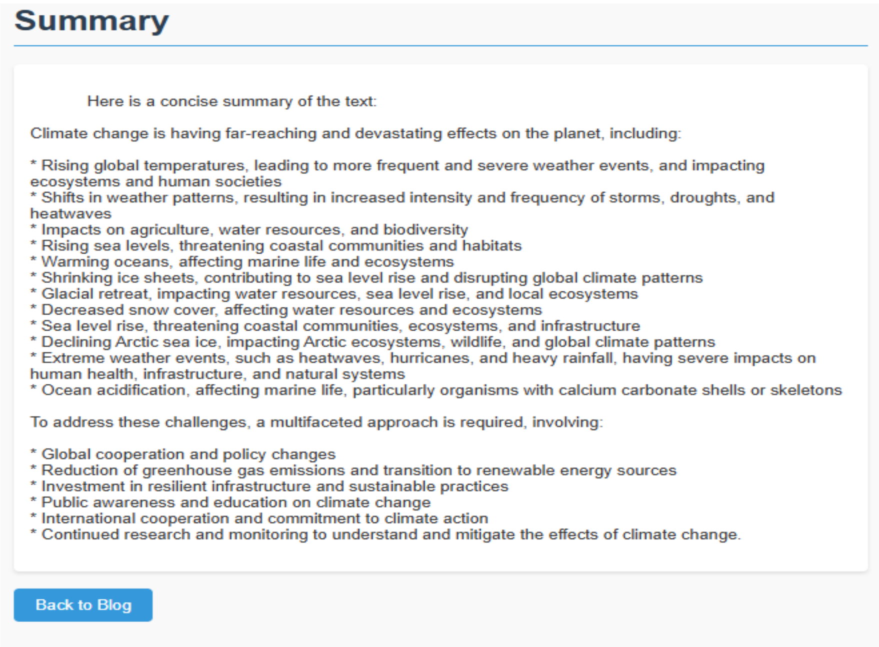
(Fig 4.2.7.1)

Presence of Summarizer and Top Q&A generator:



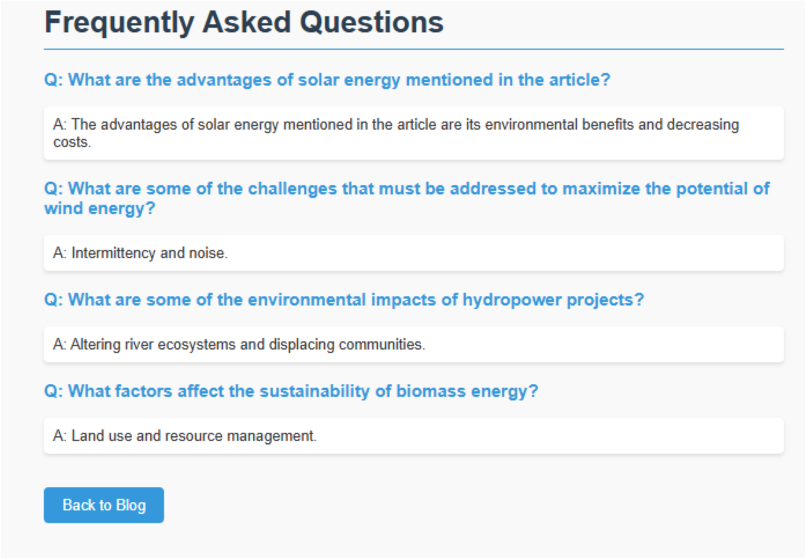
(Fig 4.2.7.2)

Summary:



(Fig 4.2.7.3)

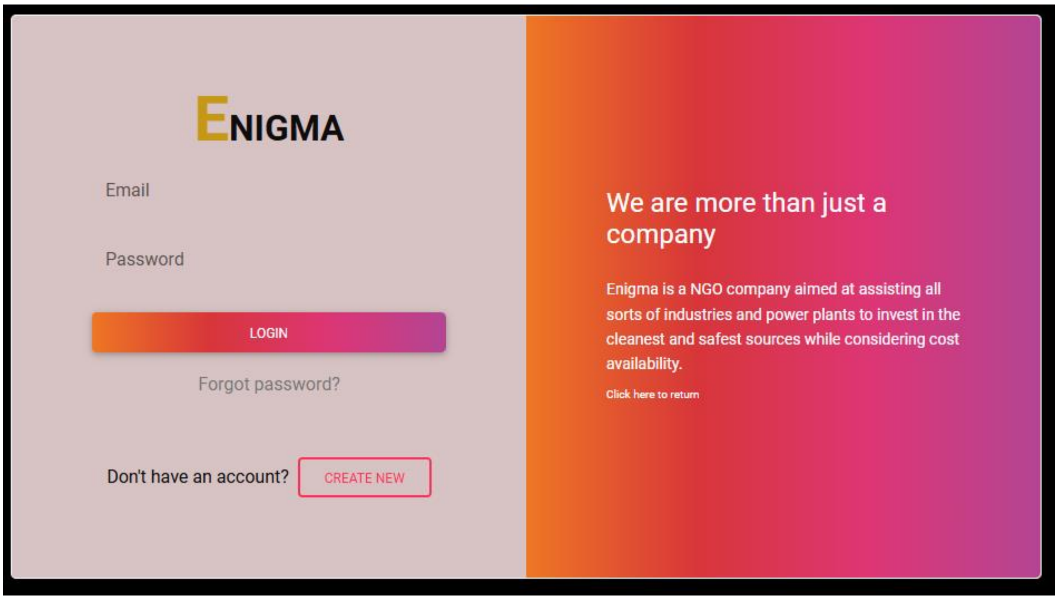
Q&A:



(Fig 4.2.7.4)

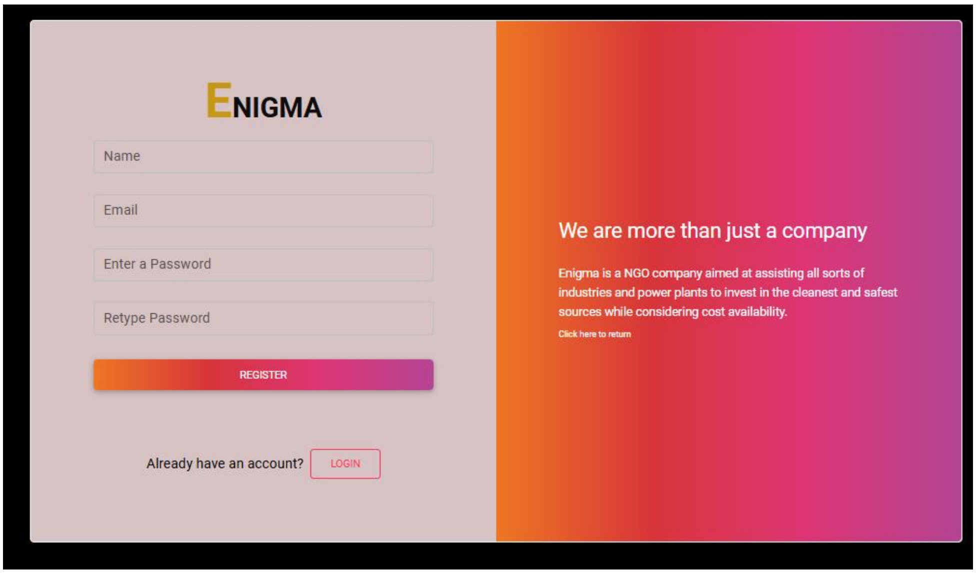
4.3 Screenshots of User Identification

4.3.1 Login Page



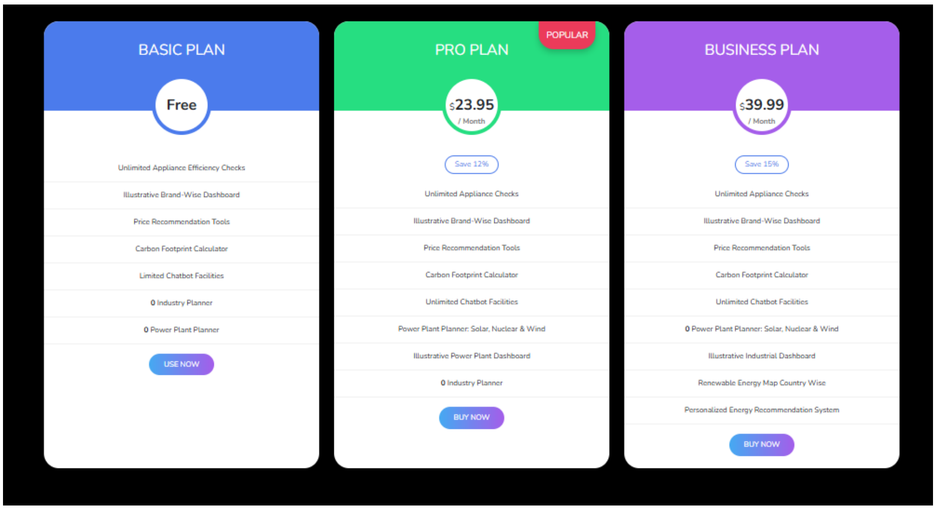
(Fig 4.3.1)

4.3.2 Create Account Page

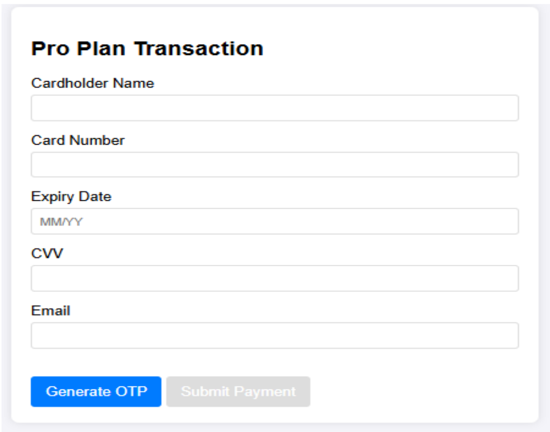


(Fig 4.3.2)

4.3.3 Pricing Page



(Fig 4.3.3.1)



(Fig 4.3.3.2)

5. TESTING

5.1 Test Cases

5.1.1 Efficiency Model

5.1.1.1: Device Classification Accuracy

- Objective: Verify Random Forest Classifier's accuracy in classifying devices as
- efficient or inefficient.
- Method: Test with devices of known efficiency status.
- Expected Result: High accuracy in classification.

5.1.1.2: Handling Missing Data

- Objective: Check model's response to incomplete input features.
- Method: Provide data with missing features.
- Expected Result: Graceful handling or meaningful error message.

5.1.2 Carbon Footprint Calculator

5.1.2.1: Carbon Emission Prediction Accuracy

- Objective: Ensure MLR model predicts carbon emissions accurately.
- Method: Test with known lifestyle factors and compare predictions.
- Expected Result: Predictions within acceptable error margin.

5.1.2.2: Handling Edge Cases

- Objective: Test model with extreme values.
- Method: Provide extreme lifestyle factor values.
- Expected Result: Model handles extremes without errors.

5.1.3 Energy Prediction

5.1.3.1: Energy Output Prediction Accuracy

- Objective: Verify MLR model's accuracy in predicting energy output.
- Method: Compare predictions with known output values.
- Expected Result: Close match to actual values.

5.1.3.2: Frontend Visualization

- Objective: Ensure accurate frontend display of energy predictions.

- Method: Test Flask integration for displaying charts.
- Expected Result: Accurate and well-rendered charts.

5.1.4 Energy Recommendation System

5.1.4.1: Optimal Solution Generation

- Objective: Verify Genetic Algorithm (DEAP) generates optimal solutions.
- Method: Input various constraints and check solutions.
- Expected Result: Optimal solutions meeting constraints.

5.1.4.2: Performance with Constraints

- Objective: Test algorithm with complex constraints.
- Method: Provide complex constraints.
- Expected Result: Efficient handling and solution generation.

5.1.5 Summary and FAQ Generation

5.1.5.1: Summary Generation

- Objective: Ensure LLaMA 3 generates accurate summaries.
- Method: Test with various blog contents.
- Expected Result: Concise and accurate summaries.

5.1.5.2: FAQ Generation

- Objective: Verify relevant FAQ generation.
- Method: Test with blog content.
- Expected Result: Relevant and accurate FAQs.

5.1.5.3: Interface Integration

- Objective: Test ChatGroq and Flask integration.
- Method: Generate summaries and FAQs through the interface.
- Expected Result: Smooth operation with accurate outputs.

5.1.6 Live Trends Visualization

5.1.6.1: Live Trends Accuracy

- Objective: Ensure accurate live trends with Charts.js.
- Method: Test real-time data visualization.
- Expected Result: Accurate and responsive trends.

5.1.6.2: Data Storage and Retrieval

- Objective: Verify efficiency of data operations with XAMPP Server.
- Method: Test data storage and retrieval.
- Expected Result: Efficient operations without errors.

5.1.7 User Interface and Interaction

5.1.7.1: Registration Form Validation

- Objective: Validate user input on registration form.
- Method: Test with valid and invalid inputs.
- Expected Result: Valid inputs register successfully; invalid inputs trigger errors.

5.1.7.2: Login Form Validation

- Objective: Validate user credentials on login form.
- Method: Attempt login with correct and incorrect credentials.
- Expected Result: Correct credentials allow login; incorrect show error.

5.1.8 Security and Performance

5.1.8.1: Password Hashing

- Objective: Ensure secure password storage.
- Method: Register with a known password and verify hashing.
- Expected Result: Passwords stored as hashed values.

5.2 Test Reports

5.2.1 Efficiency Model

- Report: The Random Forest Classifier was rigorously tested with various device datasets to ensure accurate classification as efficient or inefficient. The model consistently demonstrated high classification accuracy, matching the known efficiency statuses of devices. Additionally, the system managed missing data effectively, either by providing default behavior or meaningful feedback when features were absent.

5.2.2 Carbon Footprint Calculator

- Report: The Multiple Linear Regression (MLR) model for carbon footprint calculation was evaluated using a diverse range of lifestyle factors, such as body type, diet, transportation mode, and energy usage. The model’s predictions were consistently close to actual emission values, staying within an acceptable error margin. The model also handled extreme values for lifestyle factors adeptly, providing reasonable and error-free predictions.

5.2.3 Energy Prediction

- Report: The MLR model used for energy prediction was tested with various input features and demonstrated accurate predictions of energy output values. The results were compared with known values and matched closely. The integration with Flask was effective, displaying charts and visualizations correctly on the frontend, reflecting accurate energy predictions. The system performed efficiently in both backend processing and frontend rendering.

5.2.4 Energy Recommendation System

- Report: The Genetic Algorithm (DEAP) was tested for generating optimal energy solutions based on a range of user-defined constraints. The algorithm successfully provided top solutions that met the defined requirements. Performance tests with complex constraints showed that the algorithm managed them efficiently, without significant delays or performance issues, delivering reliable optimal solutions.

5.2.5 Summary and FAQ Generation

- Report: The LLaMA 3 model was evaluated for generating summaries and FAQs from various blog contents. It produced accurate and concise summaries that captured the key points of the content. The FAQ generation process also resulted in relevant and accurate questions and answers. The integration with ChatGroq and Flask provided a seamless interface for users, and the system functioned smoothly, delivering the expected outputs without issues.

5.2.6 Live Trends Visualization

- Report: Charts.js was utilized to display live trends for user login activity. The system accurately visualized real-time data, with charts reflecting up-to-date trends and metrics. Data retrieval and storage operations using XAMPP Server were efficient, handling high loads effectively and displaying trends without significant delays or errors.

5.2.7 Backend Validation and Security

- Report: The backend system implemented secure password hashing for user accounts. Passwords were stored as hashed values in the database, ensuring protection against unauthorized access. The testing confirmed that passwords are not stored in plain text, aligning with security best practices and safeguarding sensitive information.

5.2.8 Frontend Validation and Error Handling

- Report: The registration and login forms were tested for validation with both valid and invalid inputs. The system correctly identified invalid inputs, such as incorrect email formats and weak passwords, and displayed appropriate error messages. Successful inputs allowed users to register or log in. Error handling was effective, with user data retained where applicable and clear visual indicators provided to guide users in correcting errors.

6. CONCLUSIONS

6.1. Design and Implementation Issues

The chatbot's design and implementation encountered issues, primarily with accuracy, due to the difficulty in handling large input texts. The chatbot uses Retrieval-Augmented Generation (RAG), which merges retrieval of relevant information with generative responses. However, the system's performance was suboptimal, needing further refinement to manage extensive inputs effectively.

The payment portal faced issues, mainly due to the lack of connectivity to actual payment gateways, resulting in inaccuracies during transactions. This limitation prevents users from experiencing a seamless payment process, reducing the platform's credibility and functionality. Integration with reliable payment gateways is essential for improving user experience and trust.

6.2. Advantages and Limitations

The Enigma platform integrates a wide range of energy-related functionalities, making it versatile and suitable for diverse users. The model and algorithm were developed from scratch, allowing for complete customization and innovation. A standout feature is the incorporation of multiple variables, including a detailed carbon footprint calculator that enhances the model's efficiency. This calculator accounts for 14 different inputs, unlike traditional models with limited variables, providing a more accurate environmental impact assessment. Enigma's approach ensures users receive comprehensive insights and actionable recommendations, aligning their energy usage with sustainability goals and significantly improving decision-making efficiency.

The Enigma platform faces limitations, such as potential inaccuracies when real data deviates from general trends, affecting the model's reliability. Additionally, there is a need to enhance paid services by adding more functionalities, which could improve user experience and offer greater value but requires further development and resources.

6.3. Future Enhancements

The future prospects of Enigma are expansive, with the potential to evolve into a comprehensive global platform for energy optimization and sustainability. One key avenue for future growth is the integration of real-time data monitoring and analytics. By incorporating IoT sensors and advanced data feeds, Enigma could provide real-time energy consumption data and predictive insights, allowing users to monitor their energy use dynamically. This would enable more immediate and actionable recommendations for improving efficiency and reducing environmental impact. Additionally, the platform could expand its support for renewable energy sources, offering predictive models for solar, wind, and other sustainable options, helping industries and households transition to cleaner energy alternatives.

Another promising future prospect lies in enhancing the platform's AI-driven recommendations and chatbot functionalities. By incorporating more advanced natural language processing (NLP) and machine learning algorithms, the RAG chatbot could become an even more intuitive and personalized assistant, helping users navigate complex energy data and offering real-time support. The platform could also expand its scope to new industries such as transportation, agriculture, or construction, providing tailored energy efficiency solutions for these high-impact sectors. Additionally, future developments could include global scalability, with region-specific models that account for diverse environmental and regulatory conditions, making Enigma a go-to solution for energy efficiency.

REFERENCES

[1] Renewables Ninja. Accessed [July 15, 2024]. https://www.renewables.ninja/

[2] Our World in Data. Accessed [July 17, 2024]. https://ourworldindata.org/

[3] Open Energy Information. Accessed [July 17, 2024]. https://openei.org/wiki/Main_Page

[4] EnergySage. Accessed [July 17, 2024]. https://www.energysage.com/

[5] Akkiraju, Rama, et al. "FACTS About Building Retrieval Augmented Generation-based Chatbots." arXiv preprint arXiv:2407.07858 (2024).