COIMBATORE INSTITUTE OF TECHNOLOGY

COIMBATORE – 641014

DEPARTMENT OF INFORMATION TECHNOLOGY

INTERNSHIP REPORT

In Association with

Boredom LLP Technologies

MVP to AI-Powered Apps: For the Modern Web Builders

04.06.2025 to 17.06.2025

SUBMITTED BY

NAGARAJ M

711623243032

KATHIR COLLEGE OF ENGINEERING

REPORT SUBMISSION DATE:16.06.2025

JUNE 2025

TABLE OF CONTENTS

S.NO	DESCRIPTION	PAGE NO
1	Bonafide Certificate	03
2	Introduction	04
3	Module -1	05
4	Module -2	09
5	Module-3	11
6	Module-4	15
7	Module-5	20
8	Outcome of the Internship	23
9	Future Directions	24
10	References,conclusion	25

COIMBATORE INSTITUTE OF TECHNOLOGY

COIMBATORE - 641014

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

Certified that this Internship in **MVP to AI-Powered Apps: For the Modern Web Builders** is being carried out by NAGARAJ M (711623243032) from 04.06.2025 to 17.06.2025.

Student Signature

Internal Committee Members Recommendations

□ Excellent □ Good □ Satisfactory □ Not Satisfactory

Name and Signature:

1.Dr.M.Rajalakshmi

2.Dr.N.Anithadevi

3.Dr.K.Malarvizhi

4. Dr.S.Poornima

5. Dr.S.Rajasree

INTRODUCTION

🌱 About EcoAI Sustainability Project:

EcoAI is an intelligent sustainability tracking platform that empowers users to monitor their environmental impact and receive AI-generated suggestions for greener habits. It aims to increase awareness and support eco-conscious decision-making using technology.

🔧 Modules of the Project:

- **User Interaction Module**
  Collects user queries and tracks activity.
- **Eco Suggestion Engine**
  Provides AI-generated sustainability tips using OpenAI's API.
- **Analytics & Insights**
  Displays real-time stats on usage and sustainability metrics via Vercel Analytics.
- **Storage & History**
  Uses Firebase Firestore and Storage for saving queries and related files/images.

□ Components & API Integration:

- **Components**:
  Banner, Footer, Avatar, EcoSuggestions, SideAnimations
- **APIs Used**:
  - OpenAI GPT for intelligent responses
  - Firebase for database and storage
  - Vercel for deployment and analytics

⚙ Tech Stack (Next.js App):

- **Frontend**: Next.js 15.3, Tailwind CSS
- **Backend**: API Routes (app/api/ask/route.ts)
- **Deployment**: Vercel
- **Database**: Firebase Firestore

Module -1

Landing Page Design & Github Deployment

✅ Points Learned in the Module:

In this module, I learned the fundamentals of front-end web development, including HTML structure, CSS styling, and user interface design. I explored how to build a responsive and visually appealing landing page using sections like headers, hero content, images, and footers. I also gained experience working with buttons, background shapes, and layout enhancements to improve user engagement. Additionally, I learned the importance of semantic HTML, accessibility, and clean design. The deployment process taught me how to use GitHub to host a live website, manage version control, and share projects online using GitHub Pages.

✅ Changes Made to the Code:

I made several changes to enhance the landing page for EcoAI. I added a clean HTML structure with header, body, and footer elements, then styled them using CSS for a modern look. I included background gradients, decorative shapes (circles, squares), and responsive layout styling. I integrated an image with size control and added interactive buttons with hover effects. A second footer line was added to credit the creator. Additionally, I updated text content and ensured everything was responsive for different screen sizes. These changes improved visual appeal and user experience, preparing the project for deployment on GitHub Pages.

INDEX.HTML CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>

  <title>EcoAI: Smart Carbon Tracker</title>

  <link rel="stylesheet" href="styles.css" />

</head>
```

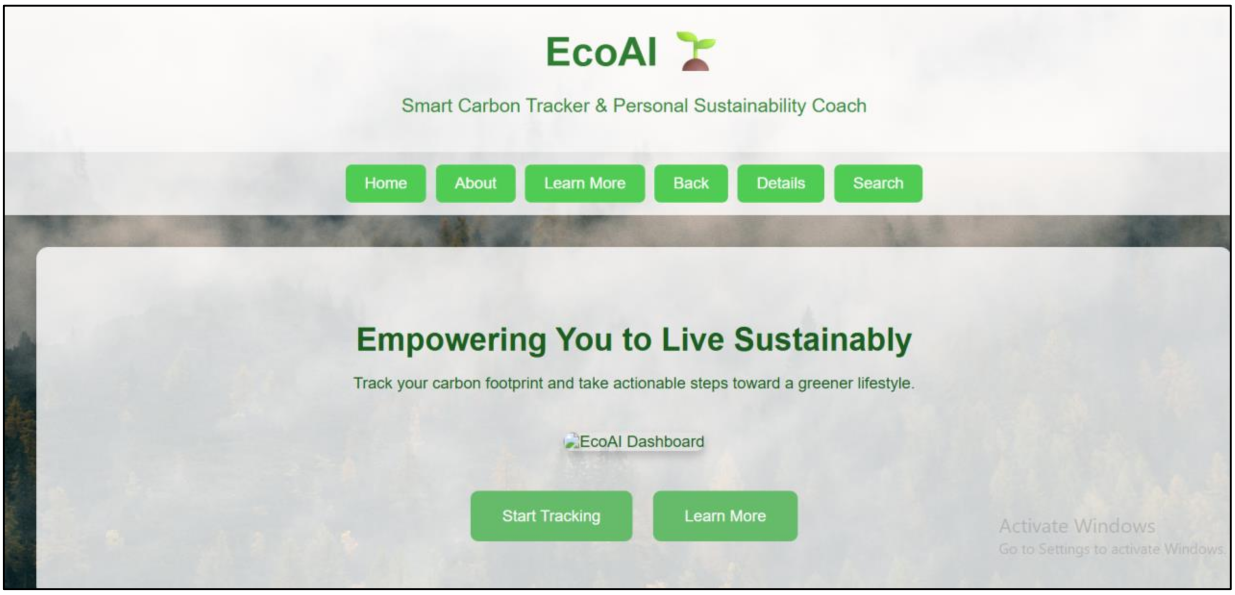
```
<body>


  <header>
    <h1>EcoAI 🌱</h1>
    <p>Smart Carbon Tracker & Personal Sustainability Coach</p>
  </header>


  <nav>
    <button onclick="alert('Home Clicked')">Home</button>
    <button onclick="alert('About Clicked')">About</button>
    <button onclick="alert('Learn More Clicked')">Learn More</button>
    <button onclick="alert('Back Clicked')">Back</button>
    <button onclick="alert('Details Clicked')">Details</button>
    <button onclick="alert('Search Clicked')">Search</button>
  </nav>


  <section class="hero">
    <h2>Empowering You to Live Sustainably</h2>
    <p>Track your carbon footprint and take actionable steps toward a greener lifestyle.</p>


    <img src="C:\Users\ADMIN\Downloads\pngtree-eco-friendly-illustrations-for-world-environment-day-png-image_11557373.png" alt="EcoAI Dashboard" width="250">


    <div class="buttons">
      <button onclick="alert('Start Tracking Clicked')">Start Tracking</button>
      <button onclick="alert('Learn More Clicked')">Learn More</button>
    </div>
  </section>


  <footer>
    © 2025 EcoAI. All rights reserved.<br>
    CREATED BY NAGARAJ M
  </footer>
```

```
</body>
</html>
```

STYLE.CSS CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>EcoAI: Smart Carbon Tracker</title>
  <link rel="stylesheet" href="styles.css" />
</head>
<body>

  <header>
    <h1>EcoAI 🌱</h1>
    <p>Smart Carbon Tracker & Personal Sustainability Coach</p>
  </header>

  <nav>
    <button onclick="alert('Home Clicked')">Home</button>
    <button onclick="alert('About Clicked')">About</button>
    <button onclick="alert('Learn More Clicked')">Learn More</button>
    <button onclick="alert('Back Clicked')">Back</button>
    <button onclick="alert('Details Clicked')">Details</button>
    <button onclick="alert('Search Clicked')">Search</button>
  </nav>

  <section class="hero">
    <h2>Empowering You to Live Sustainably</h2>
    <p>Track your carbon footprint and take actionable steps toward a greener lifestyle.</p>
    <img src="C:\Users\ADMIN\Downloads\pngtree-eco-friendly-illustrations-for-world-environment-day-png-image_11557373.png" alt="EcoAI Dashboard" width="250">


    <div class="buttons">
      <button onclick="alert('Start Tracking Clicked')">Start Tracking</button>
      <button onclick="alert('Learn More Clicked')">Learn More</button>
    </div>
  </section>


  <footer>
    © 2025 EcoAI. All rights reserved.<br>
    CREATED BY NAGARAJ M
  </footer>


</body>
</html>
```

OUTPUT:



EcoAI 🌱

Smart Carbon Tracker & Personal Sustainability Coach

Empowering You to Live Sustainably

Track your carbon footprint and take actionable steps toward a greener lifestyle.

Module -2

Nextjs code and Vercel Deployment

✓ Points Learned in the Module:

In this module, I learned how to set up and build applications using Next.js, a powerful React-based framework. I explored both routing systems—Pages Router and App Router—for creating multi-page applications. I learned how to add components, import CSS using global and module styles, and structure pages using JSX. Additionally, I discovered how Next.js handles server-side rendering and static site generation. Finally, I gained practical experience with deploying projects to Vercel, Next.js's hosting platform, which supports one-click deployments, automatic CI/CD, and custom domain configurations, enabling seamless and scalable application hosting for production-ready apps.

✓ Changes Made to the Code:

To prepare the project for deployment on Vercel, I updated the code by organizing pages using the App Router in the app/ directory, adding new pages like About and Contact using page.tsx files. I applied styling using CSS Modules and ensured responsive design. I modified the next.config.js file for base path configuration (if needed) and verified image optimization settings. Additionally, I cleaned up unused imports and simplified navigation using <Link> components. After pushing the code to GitHub, I linked the repository to Vercel, configured environment variables, and successfully deployed the app with automatic updates on each new commit.

PAGES/INDEX.JS :

```
// src/pages/index.js
import React from 'react';
import CounterComponent from '../components/CounterComponent';


export default function Home() {
  return (
    <div>
      <h1>Welcome to EcoAI Counter</h1>
      <CounterComponent />
    </div>
  );
}
```

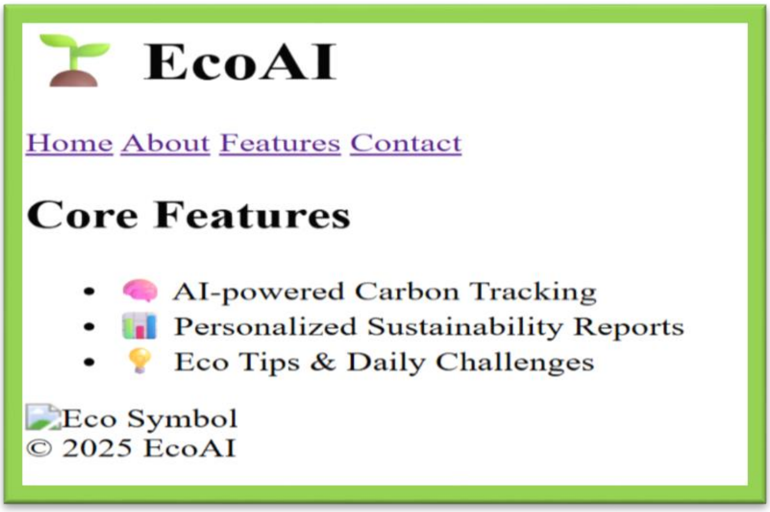
HOOKS/USECOUNTER.JS:

```
import React from 'react';
import useCounter from '../hooks/useCounter';


export default function CounterComponent() {
  const { count, increment, decrement, reset } = useCounter(0);


  return (
    <div>
      <h2>Count: {count}</h2>
      <button onClick={increment}>+</button>
      <button onClick={decrement}>-</button>
      <button onClick={reset}>Reset</button>
    </div>
  );
}
```

SCRIPT.JS

```
// Add JS interactions if needed later
console.log("EcoAI site loaded.");
```

OUTPUT:



🌱 EcoAI

Home About Features Contact

Core Features

- 🧠 AI-powered Carbon Tracking
- 📊 Personalized Sustainability Reports
- 💡 Eco Tips & Daily Challenges

Eco Symbol
© 2025 EcoAI

Module -3

Nextjs components

✅ Points Learned in the Module (100 words)

In this module, I learned how to build a modern web application using Next.js with a strong emphasis on component-based architecture. I explored how to create reusable UI components like headers, forms, and hero sections. I gained an understanding of routing using the App Router, managing state with useState, and handling side effects with useEffect. Additionally, I learned how to style components using Tailwind CSS and import images from the public directory. This module also helped me understand layout composition in Next.js using a shared layout.tsx file and client/server component distinction for optimal performance.

✅ Changes Made to the Code (100 words)

I restructured the Next.js project using the App Router by organizing pages into folders (/home, /about, /contact) with page.tsx files. I created reusable components in a separate /components folder, such as Navbar, Hero, and ContactForm. I used useState to manage interactive elements like form inputs and button clicks, and applied useEffect for lifecycle behavior. The global layout was centralized in layout.tsx to maintain consistent design. Tailwind CSS was integrated for responsive styling, and images were imported using the Next.js Image component from the /public/images folder to optimize loading and performance.

COMPONENTS/CARBON CALCULATOR:

```
'use client';

import { useState } from 'react';



export default function CarbonCalculator() {

  const [miles, setMiles] = useState(0);

  const [result, setResult] = useState<number | null>(null);



  const handleCalc = () => {
    setResult(miles * 0.404); // 0.404kg CO2 per mile
  };

  return (
    <div className="p-4 bg-blue-100 rounded mt-4">
      <h2 className="font-bold">Carbon Calculator</h2>
      <input
        type="number"
        value={miles}
        onChange={(e) => setMiles(+e.target.value)}
        className="border p-1"
        placeholder="Miles driven"
      />
      <button onClick={handleCalc} className="ml-2 bg-blue-500 text-white px-2 py-1 rounded">
        Calculate
      </button>
      {result !== null && <p className="mt-2">CO₂: {result.toFixed(2)} kg</p>}
    </div>
  );
}
```

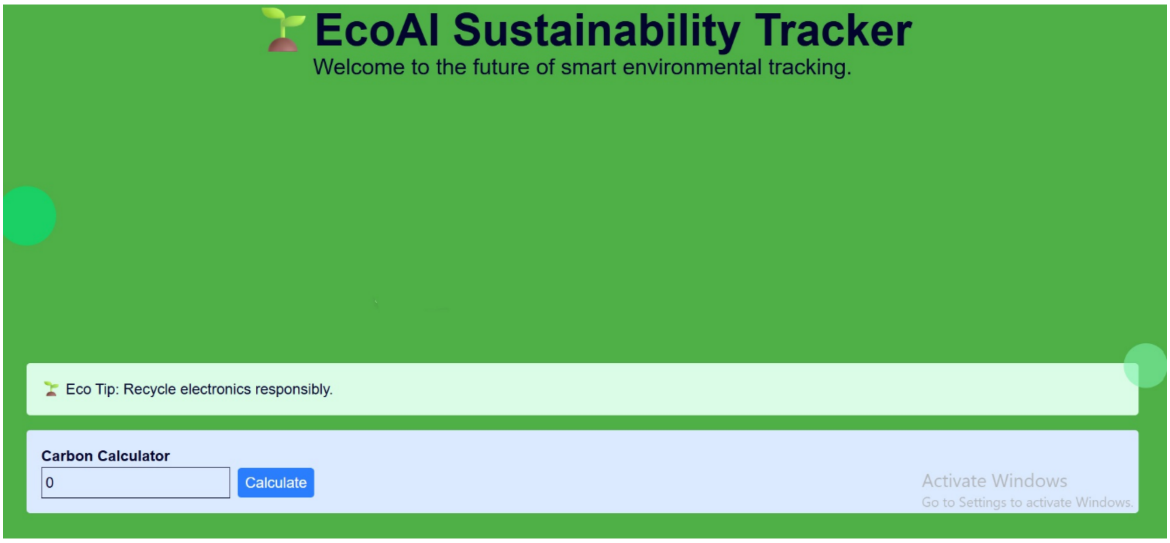
ECOTIPS:

```
'use client';
import { useEffect, useState } from 'react';


export default function EcoTip() {
  const [index, setIndex] = useState(0);

  useEffect(() => {
    const interval = setInterval(() => {
      setIndex((i) => (i + 1) % [
        'Turn off lights when not in use.',
        'Use reusable water bottles.',
        'Recycle electronics responsibly.',
        'Bike or walk instead of driving.',
        'Reduce meat consumption.'
      ].length);
    }, 3000);
    return () => clearInterval(interval);
  }, []);

  return (
    <div className="p-4 bg-green-100 rounded gap-6">
      🌱 Eco Tip: {[
        'Turn off lights when not in use.',
        'Use reusable water bottles.',
        'Recycle electronics responsibly.',
        'Bike or walk instead of driving.',
        'Reduce meat consumption.'
      ][index]}
    </div>
  );
}
```

FRAMES:

```
"use client";
import { motion } from "framer-motion";
export default function SideAnimations()
{
  return (
    <>
      {/* Left floating leaf */}
      <motion.div
        className="fixed left-0 top-1/3 w-16 h-16 bg-green-400 rounded-full opacity-70 z-10"
        animate={{ y: [0, 20, 0] }}
        transition={{ duration: 4, repeat: Infinity }}
      />

      {/* Right floating circle */}
      <motion.div
        className="fixed right-0 bottom-1/4 w-12 h-12 bg-green-300 rounded-full opacity-60 z-10"
        animate={{ y: [0, -20, 0] }}
        transition={{ duration: 3, repeat: Infinity }}
      />
    </>
  );
}
```

OUTPUT:



🌱 EcoAI Sustainability Tracker

Welcome to the future of smart environmental tracking.

🌱 Eco Tip: Recycle electronics responsibly.

Carbon Calculator

Module -4

MVP Design and Backend API

✅ Points Learned in the Module :

This module provided practical knowledge on building modern web applications using **Next.js**, focusing on server-side rendering, component reusability, and routing. I learned to structure projects using modular components for scalability. API integration was a core part, including communication with **Firebase** for data storage and **OpenAI** for AI-driven responses. I also gained insights into deploying applications with **Vercel**, managing cloud functions, and monitoring usage with **Vercel Analytics**. Styling with Tailwind CSS and animating components using libraries like Framer Motion enhanced the UI/UX. Git and GitHub were used for version control, with emphasis on clean commit practices and deployment workflows.

🔧 Changes Made to the Code :

Key changes include redesigning the layout with a persistent **eco-themed banner**, footer section with images and text for eco suggestions, and global styling using Tailwind CSS. I deleted unused components and modularized layout elements like avatars and side animations. Firebase was integrated to log prompts and AI responses. The backend API routes were updated in route.ts to fetch content from OpenAI dynamically. UI responsiveness was improved with Flexbox and Grid utilities. Animation components were added for smoother transitions. Image assets were aligned at the page bottom with content. Code was pushed and deployed to Vercel from GitHub.

□ MVP Design for One of the Modules

Module Name: *Eco Suggestion Assistant*
Goal: Deliver AI-generated eco-friendly lifestyle tips based on user prompts.

MVP Features:

- Text input for users to ask eco-related questions.
- Firebase stores queries and AI responses.
- OpenAI API provides context-aware answers.

ASKAPI:

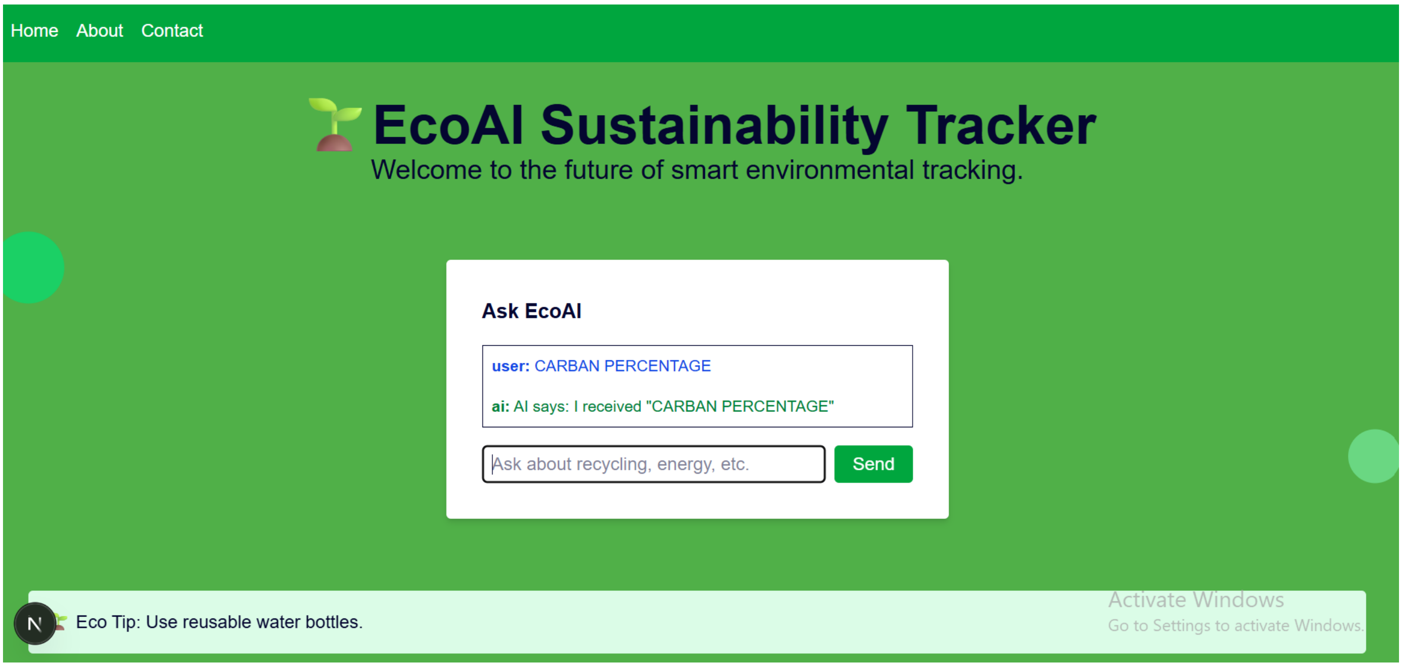
```
use client';


import { useState, useEffect, useRef } from 'react';


interface Message {
  role: 'user' | 'ai';
  content: string;
}


export default function AskAi() {
  const [input, setInput] = useState('');
  const [messages, setMessages] = useState<Message[]>([]);
  const inputRef = useRef<HTMLInputElement>(null);


  const handleSend = () => {
    if (!input.trim()) return;


    // Add user message
    const userMessage: Message = { role: 'user', content: input };
    setMessages((prev) => [...prev, userMessage]);


    // Dummy AI response (simulate real API call)
    const aiReply: Message = {
      role: 'ai',
      content: `AI says: I received "${input}"`
    };


    // Simulate delay for AI
    setTimeout(() => {
      setMessages((prev) => [...prev, aiReply]);
    }, 1000);
    setInput('');
    inputRef.current?.focus();
  };

  useEffect(() => {
    console.log('Updated messages:', messages);
  }, [messages]);

  return (
    <div className="p-8 bg-white rounded shadow-md max-w-md mx-auto gap-19">
      <h2 className="text-lg font-bold mb-4">Ask EcoAI</h2>

      <div className="space-y-4 mb-4 max-h-64 overflow-y-auto border p-2">
        {messages.map((msg, idx) => (
          <div key={idx} className={`text-sm ${msg.role === 'ai' ? 'text-green-700' : 'text-blue-700'}`}>
            <strong>{msg.role}:</strong> {msg.content}
          </div>
        ))}
      </div>

      <div className="flex gap-2">
        <input
          ref={inputRef}
          value={input}
          onChange={(e) => setInput(e.target.value)}
          placeholder="Ask about recycling, energy, etc."
          className="flex-1 border px-2 py-1 rounded"
        />
        <button
          onClick={handleSend}
          className="bg-green-600 text-white px-4 py-1 rounded hover:bg-green-700"
        >
          Send
        </button>
      </div>
    </div>
  );
 }
```

ROUTE:

```
import { NextResponse } from 'next/server';
import { db } from '@/lib/firebase';
import { addDoc, collection, Timestamp } from 'firebase/firestore';
import OpenAI from 'openai';

const openai = new OpenAI({
 apiKey: process.env.OPENAI_API_KEY,
});

export async function POST(req: Request) {
 const { prompt } = await req.json();

 const completion = await openai.chat.completions.create({
  model: 'gpt-3.5-turbo',
  messages: [{ role: 'user', content: prompt }],
 });

 const content = completion.choices[0].message.content;

 await addDoc(collection(db, 'queries'), {
  prompt,
  response: content,
  createdAt: Timestamp.now(),
 });

 return NextResponse.json({ result: content });
}
```

OUTPUT:

API REACT-AI



🌱 EcoAI Sustainability Tracker

Welcome to the future of smart environmental tracking.

Ask EcoAI

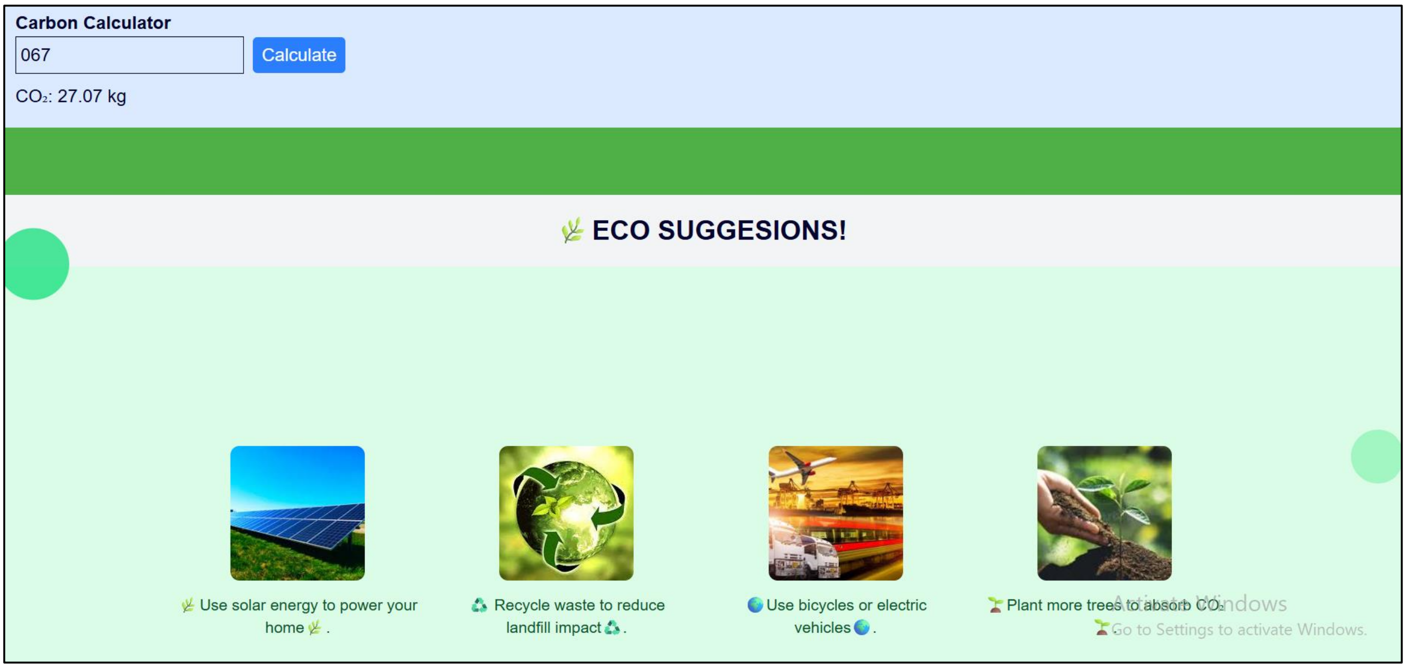
user: CARBAN PERCENTAGE

ai: AI says: I received "CARBAN PERCENTAGE"

Ask about recycling, energy, etc.

🌱 Eco Tip: Use reusable water bottles.

CARBON CALCULATOR:



Carbon Calculator

067

CO₂: 27.07 kg

🌿 ECO SUGGESIONS!

🌿 Use solar energy to power your home 🌿 .

♻️ Recycle waste to reduce landfill impact ♻️ .

🌍 Use bicycles or electric vehicles 🌍 .

🌱 Plant more trees to absorb CO₂ 🌱 .

Module -5

Storage Services and Analytics

🔗 Integration of Firebase Storage and Vercel Analytics

📦 Firebase Storage Integration

To enable file upload and retrieval in the *Eco AI Trackers* project, Firebase Storage was integrated as a secure and scalable solution:

- **Setup**:
  - Firebase project was created and configured.
  - Firebase SDK was installed using npm install firebase.
  - Firebase config credentials (API key, project ID, etc.) were stored in environment variables.
- **Code Integration**:
  - Firebase was initialized in a firebase.ts file.
  - uploadBytes, getDownloadURL, and ref functions from firebase/storage were used to handle file uploads.
  - Files uploaded by users (e.g., sustainability reports or sensor data) are stored in a structured path inside Firebase Storage.
- **Use Cases**:
  - Uploading and storing CSV/JSON reports.
  - Fetching files for further analysis or display in the dashboard.
- **Benefits**:
  - Auto-scaling, secure access, and fast CDN delivery of files.
  - Simplified file management and permissions via Firebase Console.

📊 Vercel Analytics Integration

To track the performance and user engagement of the deployed app, Vercel Analytics was enabled:

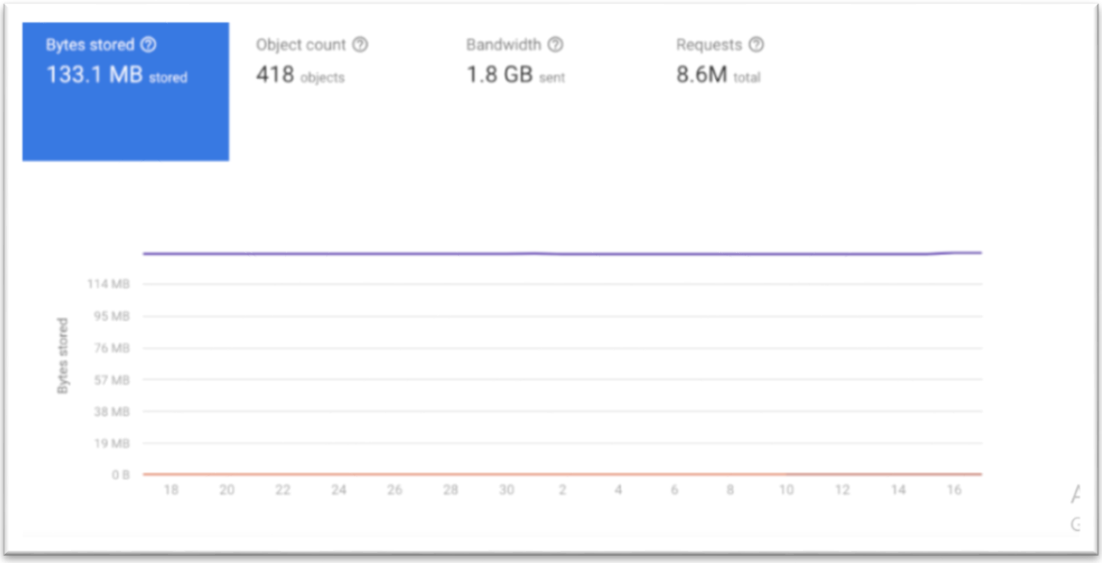
- **Setup**:
  - Analytics was activated from the **Vercel Project Dashboard → Analytics Tab → Enable Analytics**.
  - No code changes were required as Vercel supports Next.js natively.
- **Features Used**:
  - **Real-time Traffic Monitoring**: Tracks live user visits per route.
  - **Page Performance**: Reports on page load times, TTFB (Time to First Byte), and other metrics.

FIREBASE:

```
// lib/firebase.ts
import { initializeApp } from "firebase/app";
import { getFirestore } from "firebase/firestore";
import { getStorage } from "firebase/storage";

const firebaseConfig = {
  apiKey: process.env.NEXT_PUBLIC_FIREBASE_API_KEY,
  authDomain: process.env.NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN,
  projectId: process.env.NEXT_PUBLIC_FIREBASE_PROJECT_ID,
  storageBucket: process.env.NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET,
  messagingSenderId: process.env.NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID,
  appId: process.env.NEXT_PUBLIC_FIREBASE_APP_ID,
};

const app = initializeApp(firebaseConfig);
export const db = getFirestore(app);
export const storage = getStorage(app);
```

- **FIREBASE USING ENVIRONMENT VARIABLE**
- **VERCEL ANALYTICS**

OUTPUT:firebase storage and analytics



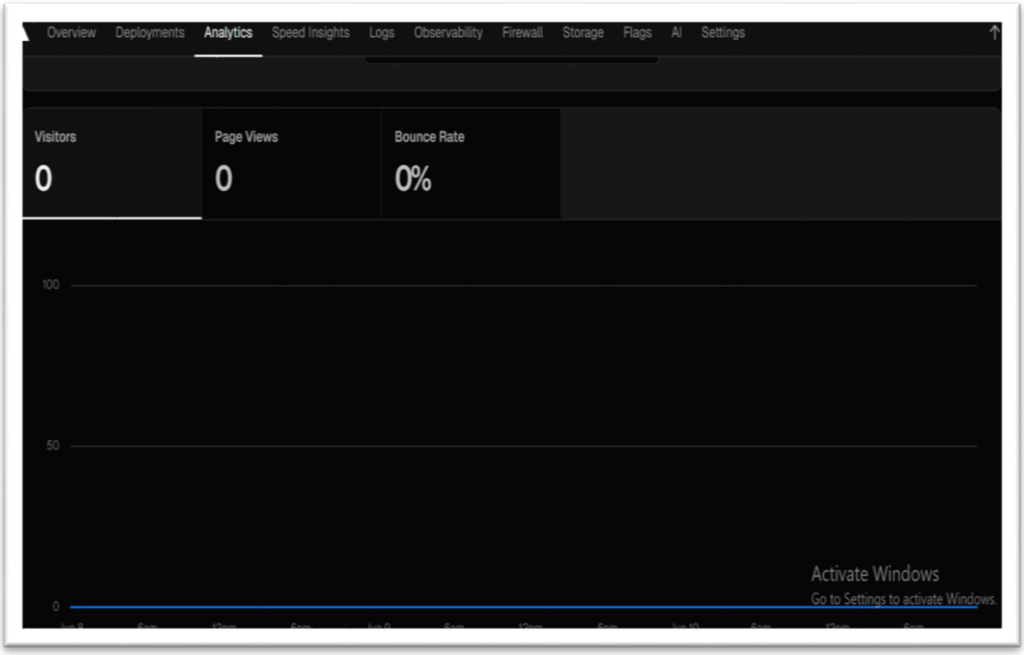
Bytes stored ⓘ 133.1 MB stored

Object count ⓘ 418 objects

Bandwidth ⓘ 1.8 GB sent

Requests ⓘ 8.6M total

Vercel analytics:

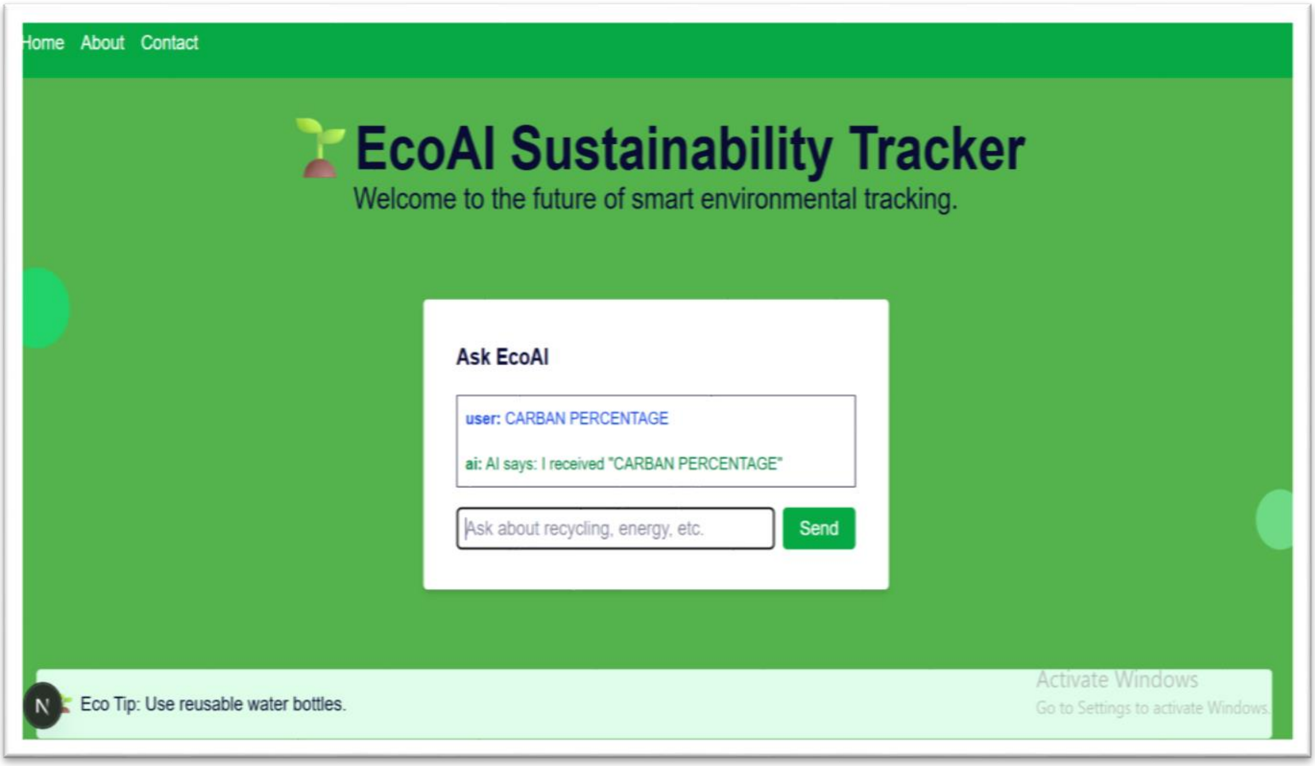


Visitors 0

Page Views 0

Bounce Rate 0%

Website outcome:



EcoAI Sustainability Tracker

Welcome to the future of smart environmental tracking.

user: CARBAN PERCENTAGE

ai: AI says: I received "CARBAN PERCENTAGE"

Ask about recycling, energy, etc.
Send

Eco Tip: Use reusable water bottles.

N
Activate Windows
Go to Settings to activate Windows.

INTERNSHIP OUTCOME:

The MVP - Modern Web Developing internship offered a comprehensive introduction to modern web technologies and practical development workflows. Throughout the program, we gained hands-on experience with key tools and frameworks essential in today's web industry, such as HTML, CSS, JavaScript, and modern front-end libraries. The sessions helped us understand the structure of responsive web design and how to build dynamic, user-friendly websites.

One of the major takeaways was learning how to work with version control systems like Git and GitHub, vercel which are critical for collaborative development. We also explored backend integration and API handling, which provided insights into full-stack development processes. The internship emphasized best practices in coding, debugging, and UI/UX principles, which greatly improved our technical thinking and problem-solving abilities.

Under the guidance of Alwin Sir, we developed not only technical skills but also improved our teamwork, communication, and project management abilities. Real-time project work enhanced our confidence in applying what we learned in practical scenarios. Overall, the internship bridged the gap between academic knowledge and industry expectations, preparing us for future professional roles in web development.

Future Directions

1. **Personalized Eco Recommendations**

   Enhance AI suggestions using user history and preferences to deliver more accurate, personalized sustainability tips.

2. **Real-Time Data Integration**

   Integrate APIs for real-time environmental data (like air quality, energy use) to generate context-aware insights.

3. **Mobile App Version**

   Develop a responsive mobile application using React Native or Flutter for wider accessibility and usage on the go.

4. **Gamification Features**

   Introduce a points system, challenges, and rewards for eco-friendly actions to improve user engagement and motivation.

5. **Community Forum Integration**

   Create a space for users to share eco-tips, ask questions, and support each other in sustainable living practices.

REFERENCES:

- **Next.js Documentation** – https://nextjs.org/docs

- **Firebase Documentation** – https://firebase.google.com/docs

- **Vercel Deployment Guide** – https://vercel.com/docs

- **OpenAI API Docs** – https://platform.openai.com/docs

- **React Documentation** – https://reactjs.org/docs/getting-started.html

- GitHub – https://github.com/nagaraj2331P/ecoai-trackers

- Tailwind CSS – https://tailwindcss.com/docs

- Google Fonts – https://fonts.google.com

- Mozilla Developer Network (MDN) – https://developer.mozilla.org

✓ Conclusion

The *EcoAI Sustainability Tracker* project successfully integrates cutting-edge web technologies with AI capabilities to offer an engaging user experience. By combining **Next.js**, **Firebase**, and the **OpenAI API**, we built a dynamic platform that provides eco-friendly suggestions based on user queries. Key features include a responsive interface, real-time AI responses, cloud storage integration, and analytics tracking via Vercel and Firebase. This project deepened our understanding of full-stack development, API integration, and user-centric design, paving the way for scalable and sustainable web applications.