AI Footprint Calculator - Detailed Project Summary

1. Project Overview

The AI Footprint Calculator is a web application designed to estimate the environmental impact (energy consumption, water usage, and carbon emissions) of various AI activities. The application allows users to calculate footprints for three types of AI usage:

- Cloud AI (hosted models like GPT-4, Claude, DALL-E)
- Local AI (models running on personal hardware)
- Agentic AI (complex AI systems that orchestrate multiple models)

2. Requirements Analysis Phase

Document Review

The project began with a thorough review of six key documents:

- Blueprint.txt - Detailed specifications for the application
- Estimating the Energy Footprint of Locally Executed Artificial Intelligence Models.rtf
- Environmental Footprint of AI Systems: Data for Calculator Tier Population.rtf
- AI Footprint Calculator Research.rtf
- AI Footprint Offsetting Research.rtf
- Aggregated Environmental Footprint Totals and Actionable Offsetting Guidance.rtf

Key Requirements Extracted

1. Core Functionality:
    - Tiered system for different AI categories
    - User inputs for AI task type, model, volume, and region
    - Calculation of energy, water, and carbon metrics
    - Local AI estimation with hardware profiles
    - Agentic AI estimation with orchestrator overhead
    - Aggregation across timeframes
    - Offsetting guidance with credit calculations
2. Data Requirements:
    - Cloud AI tier-specific energy and water data
    - Local AI hardware profiles
    - Grid carbon intensity factors by region
    - PUE (Power Usage Effectiveness) and WUE (Water Usage Effectiveness) values
    - Calculation formulas for energy, water, and carbon
3. UI/UX Requirements:
    - Minimalist, high-resolution, white-on-black, modern CRT aesthetic
    - Specific color palette, typography, and UI element styling

- Structured page flow and user journey

Clarification with Client

Initial questions were posed to the client regarding:

- Technology stack preferences
- Authentication system requirements
- Feature prioritization for MVP
- Design requirements
- Deployment preferences

The client confirmed:

- Technology stack selection was left to developer discretion
- Authentication and user accounts were not needed for the MVP
- All features except user accounts should be included
- The style guide should be strictly followed
- The application should be deployed for review and later migration

3. Planning and Architecture Phase

Technology Stack Selection

Based on the requirements, the following stack was chosen:

- Frontend: React.js (for component reusability and interactive UI)
- State Management: React Context API
- Styling: CSS with variables for the color palette
- Build Tool: Vite
- Deployment: Static site deployment

Project Structure

A modular architecture was designed with:

- Component-based UI structure
- Separation of calculator logic from presentation
- Centralized calculation engine
- Data-driven tier and profile definitions
- Responsive design for all device sizes

Implementation Plan

The development was organized into phases:

1. Project setup and core structure
2. Data preparation from research documents
3. Calculation engine implementation
4. UI component development
5. Integration and testing

6. Deployment and validation

4. Implementation Phase

Core Data Structures

Data from the research documents was transformed into structured JavaScript objects:

- Cloud AI tier definitions with energy and water consumption ranges
- Local AI hardware profiles with power draw specifications
- Grid carbon intensity factors by region
- Water usage effectiveness values
- Calculation formulas for different AI types

Calculation Engine

A comprehensive calculation engine was implemented with functions for:

- Cloud AI footprint estimation
- Local AI footprint estimation
- Agentic AI footprint estimation
- Result formatting and unit conversion
- Equivalency generation for contextualizing results

UI Components

The following components were developed:

- Calculator forms for Cloud, Local, and Agentic AI
- Results dashboard with visualization
- Offsetting guidance section
- Methodology explanation page
- Navigation and layout components

Styling Implementation

The application was styled according to the specified design system:

- CSS variables for consistent color palette
- Typography using IBM Plex Mono and IBM Plex Sans
- Custom UI elements matching the minimalist aesthetic
- Responsive layouts for all screen sizes

5. Testing and Validation Phase

Functionality Testing

All calculator components were tested to ensure:

- Correct input handling

- Proper calculation execution
- Accurate result display
- Appropriate navigation between pages

Bug Identification and Resolution

A critical issue was identified where calculation results weren't being displayed after clicking the Calculate button. This was fixed by:

- Adding form submission handlers to calculator components
- Implementing state management for calculation results
- Adding navigation from calculator to results page
- Enhancing the Results page to display calculation data

Deployment Validation

The application was deployed and tested to ensure:

- Public accessibility
- Correct functionality in production environment
- Responsive design across devices
- Calculation accuracy

6. Deployment and Delivery Phase

Initial Deployment

The application was initially deployed to a public URL for client review.

Bug Fix and Redeployment

After identifying the calculation display issue, the application was fixed and redeployed to a new URL.

Deliverables Preparation

The following deliverables were prepared for the client:

- Complete source code in a zip file
- Production build ready for deployment
- Migration instructions for subdomain setup
- Documentation of requirements and implementation

7. Technical Considerations and Decisions

Data Accuracy vs. User Experience

- Balance between detailed inputs and usability

- Tiered approach to simplify complex calculations
- Clear communication of estimation uncertainty
- Buffer option for offsetting recommendations

Performance Optimization

- Client-side calculations to avoid server dependencies
- Efficient state management
- Minimal dependencies

Accessibility and Usability

- High contrast design for readability
- Keyboard navigable interface
- Clear labeling and instructions
- Responsive design for all devices

Extensibility

- Modular code structure for future enhancements
- Data-driven approach for easy updates to calculation factors
- Separation of concerns between UI and calculation logic

8. Methodology

Research-Based Development

- All calculations based on scientific research provided
- Transparent methodology with citations
- Clear communication of assumptions and limitations

Iterative Implementation

- Progressive enhancement of features
- Regular validation against requirements
- Responsive to client feedback

User-Centered Design

- Focus on clear presentation of complex data
- Educational content integrated throughout
- Actionable recommendations for offsetting

9. Estimated Professional Equivalent

If developed by a standard professional team, this project would typically involve:

Team Composition

- 1 Project Manager
- 1 UX/UI Designer
- 2 Frontend Developers
- 1 Data Scientist/Researcher

Timeline Estimate

- Requirements Analysis: 1-2 weeks
- Design Phase: 1-2 weeks
- Development: 3-4 weeks
- Testing and Refinement: 1-2 weeks
- Deployment and Documentation: 1 week

Cost Estimate

Based on industry averages for a specialized environmental tech application with complex calculations:

- Professional team cost: $30,000 - $50,000
- Additional costs for research and data validation: $5,000 - $10,000
- Ongoing maintenance and updates: $2,000 - $5,000 per month

10. Conclusion

The AI Footprint Calculator represents a successful implementation of a complex environmental impact estimation tool. By transforming scientific research into an accessible web application, it enables users to understand and offset the environmental impact of their AI usage.
The project demonstrates the effective integration of scientific data, mathematical models, and user-centered design to create a practical tool addressing an emerging environmental concern in the AI field.