**AI Dataset Insights Generator**

**Project Overview**

The AI Dataset Insights Generator is an innovative web-based application that transforms raw datasets into actionable insights using artificial intelligence and advanced data visualization techniques.

It enables users to upload datasets and automatically obtain statistical summaries, visual representations, and AI-driven recommendations — all within a clean, interactive, and responsive interface.

**Technical Specifications**

**Core Technologies**

| Component | Technology |
| --- | --- |
| Programming Language | Python 3.8+ |
| Web Framework | Flask 3.0+ |
| Deployment Platform | Vercel (Serverless) |

**Dependencies**

| Library | Version | Purpose |
| --- | --- | --- |
| pandas | 2.2+ | Data manipulation and analysis |
| numpy | 2.0+ | Numerical computing |
| matplotlib | 3.8+ | Data visualization |
| openai | 1.30+ | AI-powered insights generation |

**Frontend Technologies**

* HTML5 – Application structure
* CSS3 – Styling and responsive design
* JavaScript (ES6+) – Interactivity and chart rendering
* Pico CSS – Lightweight CSS framework
* Font Awesome – Icon library

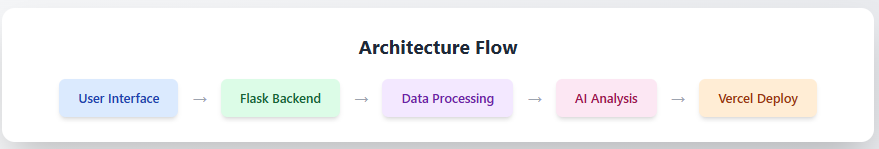
**System Architecture**

**Architecture Type**

* Serverless Web Application

**Main Components**

1. Frontend Layer
   * Interactive UI built with HTML, CSS, JavaScript
   * File upload (drag-and-drop)
   * Visualization and results display
   * Theme customization (Light/Dark)
2. Backend Layer
   * Flask web server
   * Data processing engine
   * AI integration module
   * File management and cleanup
3. Data Processing Layer
   * Pandas for data manipulation
   * NumPy for computations
   * Matplotlib for chart generation
4. AI Integration Layer
   * OpenAI GPT-based insight generation
   * Rule-based fallback for offline use
   * Automated pattern and anomaly detection
5. Deployment Layer
   * Serverless hosting on Vercel
   * Environment configuration
   * Global CDN for static asset delivery

****

**Installation and Setup**

**Prerequisites**

* Python 3.8 or higher
* pip package manager
* Git version control
* OpenAI API key

**Local Setup**

Step 1: Clone the repository

* git clone https://github.com/ajithram2003/Project-AI-Dataset-Insights.git
* cd Project-AI-Dataset-Insights

Step 2: Create a virtual environment

* python -m venv venv

Step 3: Activate environment

# Windows

* venv\Scripts\activate

# macOS/Linux

* source venv/bin/activate

Step 4: Install dependencies

* pip install -r requirements.txt

Step 5: Configure environment variables  
Create a .env file in the project root:

* OPENAI\_API\_KEY=your\_openai\_api\_key\_here
* FLASK\_SECRET\_KEY=your\_secret\_key\_here

Step 6: Run the application

* python src/app.py

Step 7: Access locally  
Open browser → <http://localhost:5000>

**API Documentation**

**Endpoints**

GET /

* Description: Home page with file upload interface
* Response: HTML upload form

POST /analyze

* Description: Process uploaded dataset
* Parameters:
  + dataset (file): CSV/XLS/XLSX file
* Response:
  + Statistical summary
  + AI-generated insights
  + Visualizations
  + Data preview table

**Deployment Guide (Vercel)**

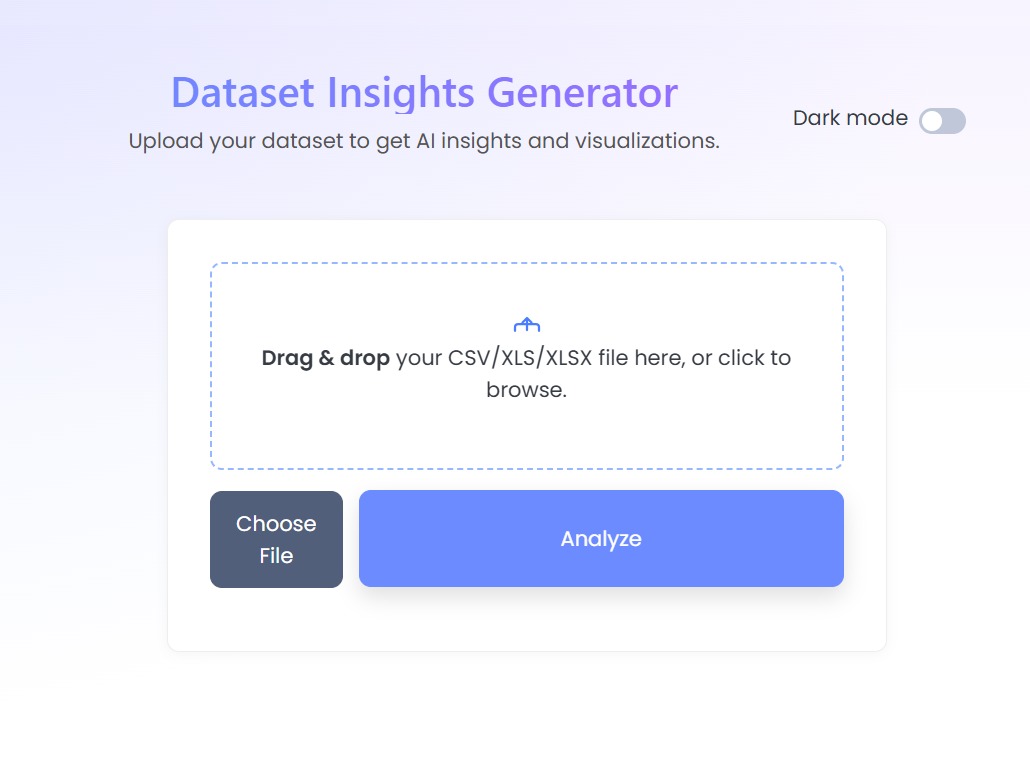
**Prerequisites**

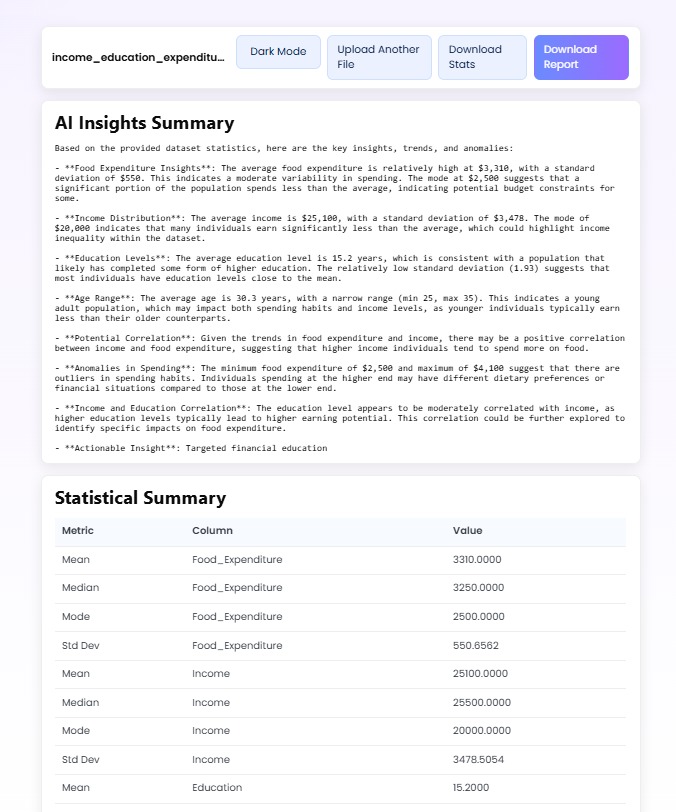
* Vercel account linked to GitHub
* OpenAI API key
* Properly structured Flask project

**Steps**

1. Visit [vercel.com](https://vercel.com)
2. Log in using GitHub
3. Click New Project → Import repository
4. Set environment variables:
   * OPENAI\_API\_KEY
   * FLASK\_SECRET\_KEY
5. Deploy the application

**Output**

****

****

****