

Project Documentation

Mohammed Showharwade Neshad
swnishat29@gmail.com
contact no. : 01689628044

Smart Data Insights Dashboard

Contents

1.	Project Overview & Purpose	2
2.	Methodology & Development Process	2
3.	Tools & Technologies.....	5
4.	Codebase Structure.....	6
5.	Key Functions Implemented	6
6.	System Architecture	7
7.	Challenges Faced.....	7
8.	Future Improvements	7
9.	Deployment Links.....	8

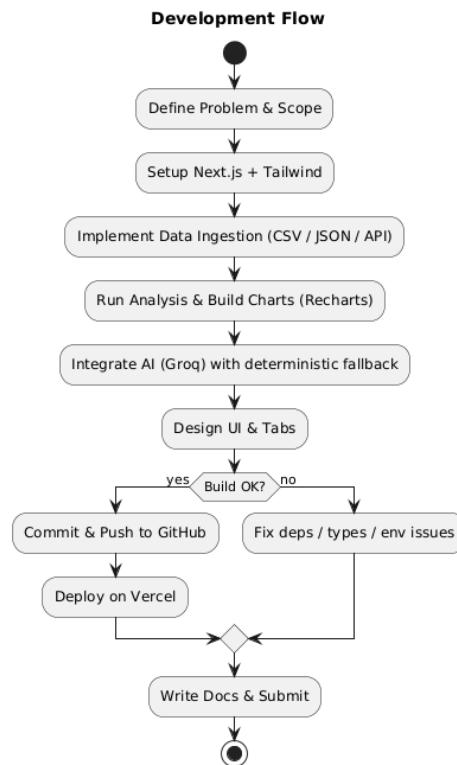
1. Project Overview & Purpose

The **Smart Data Insights Dashboard** is a full-stack web application built to transform raw data into actionable insights. Its purpose is to allow users to:

- Upload data from multiple sources (CSV, JSON, APIs).
- Automatically analyze the data, detect patterns, and highlight correlations, anomalies, or trends.
- Generate **interactive charts & graphs** (line, bar, scatter, histograms, box plots, heatmaps).
- Provide **AI-powered insights & recommendations** using Groq's LLM APIs.
- Deliver a modern, sleek, and responsive UI for easy exploration.

The system is designed to empower users (analysts, business stakeholders, students, etc.) to **understand data faster and make informed decisions** without manually writing statistical code.

2. Methodology & Development Process



Phase 1 – Setup & Initialization

- Initialized a **Next.js (App Router) project**.
- Configured project structure under `/src/app` with pages, API routes, and styles.
- Installed core dependencies:
 - react, next, tailwindcss for UI
 - papaparse for CSV parsing
 - recharts for data visualizations
 - html2canvas for chart exporting
 - dotenv for environment variables

Phase 2 – Core Features

1. Data Ingestion

- Implemented file upload (CSV, JSON) and API fetch.
- Used papaparse to parse CSV data and coerce it into a common Table format.

2. Automated Data Profiling

- Functions in `/src/lib/stats.ts` perform:
 - Missing value detection
 - Numeric summary stats (mean, median, IQR, outliers)
 - Categorical imbalance
 - Duplicate detection
 - Correlation & trend analysis

3. Interactive Visualizations

- Built chart utilities in `/src/lib/charts.ts`.
- Implemented multiple chart types in `/src/app/page.tsx`:
 - Line, Bar (Top-K), Scatter, Histogram, Box Summary, Correlation Heatmap.
- Added **export to PNG** for charts using html2canvas.

4. AI-Powered Insights

- Created /src/app/api/ai-insights/route.ts to call Groq's API.
- Fallback deterministic rule-based recommendations built in TypeScript.
- AI generates: narrative summaries, risks, recommendations, and suggested charts.

Phase 3 – UI & UX

- Customized **TailwindCSS theme** with dark background, neon gradients, and Poppins font.
- Improved **navigation tabs** (Load, Preview, Profile, Insights, Charts).
- Added **file pill preview** (shows uploaded filename).
- Ensured **responsive layout** across devices.

Phase 4 – AI Tools in Development

Throughout the process, I used:

- **ChatGPT (Sneha 😊)** → for coding guidance, debugging, and explanations.
- **DeepSeek** → for trying alternative approaches.
- **Copilot** → inline suggestions in VS Code.
- **Groq API** → powering AI insights in production.

Phase 5 – Deployment

- GitHub repo created: <https://github.com/SwNishad/xephyr-insights>
- Deployment via **Vercel**.
- Encountered multiple **build errors** (dependency conflicts, missing types, module not found).
- Fixed by:
 - Removing unnecessary openai dependency.
 - Adding @types/papaparse for TypeScript builds.
 - Cleaning node_modules and package-lock.json.
 - Updating .gitignore to exclude .env.local.
- Final environment variables configured in Vercel dashboard.

3. Tools & Technologies

Frontend

- **Next.js 15 (App Router)**
- **React**
- **TailwindCSS**
- **Recharts** (charts)
- **html2canvas** (export)

Backend

- **Next.js API Routes**
- **Groq LLM API** (AI insights)

Utilities

- **papaparse** → CSV parsing
- **TypeScript** → Strong typing
- **dotenv** → Env variable handling

Deployment

- **GitHub** → Version control
- **Vercel** → Hosting and CI/CD pipeline

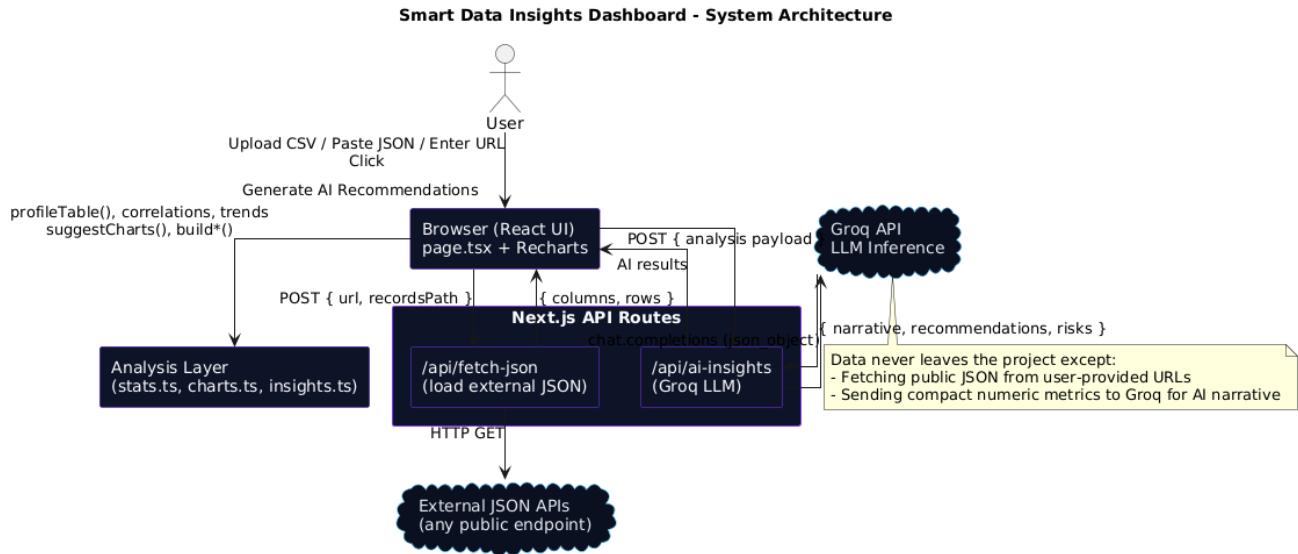
4. Codebase Structure

```
xephyr-insights/
|__ src/
|   __ app/
|   |__ api/
|   |   __ ai-insights/route.ts      # AI-powered insights
|   |   __ fetch-json/route.ts     # Load JSON data from API
|   |   __ page.tsx                # Main dashboard
|   |   __ layout.tsx              # Root layout (fonts, styles)
|   |   __ globals.css              # Global styling
|   __ lib/
|   |__ stats.ts                  # Profiling & analysis
|   |__ charts.ts                 # Chart building utils
|   |__ insights.ts               # AI payload builders
|   |__ types.ts                  # Shared TS types
|__ public/
|   __ sample-data/              # Example CSV/JSON files
|__ .env.local                   # Local env variables
|__ .gitignore
|__ package.json
|__ README.md
```

5. Key Functions Implemented

- `profileTable()` → column profiling
- `topCorrelations()` → strongest correlations
- `firstDateTrend()` → time-series trend detection
- `suggestCharts()` → recommends chart types
- `generateInsightsAll()` → structured insights from stats
- `buildAiAnalysisPayload()` → compact AI-ready JSON
- `ChartCard` → reusable chart container with export

6. System Architecture



7. Challenges Faced

1. Dependency Conflicts

- openai required Zod v3, but project had Zod v4.
- Fixed by removing openai package and only using Groq's REST API.

2. Build Errors on Vercel

- Missing types (papaparse).
- Fixed by installing @types/papaparse.

3. Environment Variables

- .env.local not pushed to GitHub.
- Fixed by creating .env.example for structure and setting actual keys in Vercel.

4. Time Consumption

- Although intended for 3–4 hours, due to repeated deployment/debugging cycles, total effort extended beyond **3-4 hours**.

8. Future Improvements

- Add **more visualization types** (Pie, Donut, Treemap, Sankey).

- Provide **export to PDF/Excel** for reports.
- Enhance **AI explanations** with chart-specific narratives.
- Add **user authentication** and private dashboards.
- Enable **real-time data streams** (e.g., APIs/websockets).
- Better **mobile UI optimization**.

9. Deployment Links

- **Live App:** <https://xephyr-insights-qhacgkwph-neshads-projects-cd619fd4.vercel.app>
- **GitHub Repo:** <https://github.com/SwNishad/xephyr-insights>