

# Florida Flood Gauge Monitor

## Real-Time Interactive Web Application with AI Analysis

Paul Fishwick and Claude Code

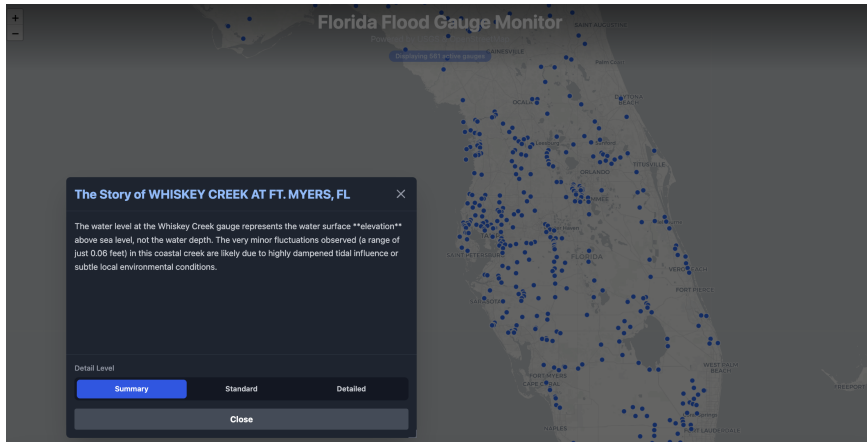
January 4, 2026

# Project Overview

- Web-based application for real-time flood gauge monitoring across Florida
- Integrates live data from USGS Water Services API
- Interactive map displaying active monitoring stations
- AI-powered explanations of water level fluctuations
- Built with modern web technologies (React, TypeScript, Vite)

**Purpose:** Transform raw sensor data into actionable insights accessible to both technical and non-technical users

# Interface Demonstration



## WHISKEY CREEK AT FT. MYERS, FL



# Key Features

## Data Integration

- Live USGS API data
- 7-day historical trends
- Active gauges statewide
- Real-time updates

## Visualization

- Interactive OpenStreetMap
- Time-series charts
- Responsive design
- Intuitive navigation

## AI-Powered Analysis

- Google Gemini integration
- Three detail levels
- Contextual explanations
- Cached for performance

## Factors Analyzed

- Water body type
- Tidal influences
- Precipitation patterns
- Regional topography

# System Architecture

## Component-Based Design:

- **App Component:** State management, data orchestration, error handling
- **Map Component:** Leaflet-based interactive mapping with gauge markers
- **GaugeChart Component:** Recharts visualization of historical water levels
- **GaugeStory Component:** AI-generated explanations with detail toggles

## Unidirectional Data Flow:

- 1 Fetch USGS data on initialization
- 2 Store in component state
- 3 Pass to child components via props
- 4 User interactions update parent state
- 5 Components re-render with new data

# Technical Stack

Category	Technology
Framework	React 19 with TypeScript 5
Build Tool	Vite 5 (fast development server)
Mapping	Leaflet 1.9 + React-Leaflet 5
Charts	Recharts 3
AI Integration	Google Gemini API (gemini-2.5-flash)
Data Source	USGS Water Services API
Deployment	Single-page application (SPA)

## Key Implementation Details:

- Environment variables via Vite (`import.meta.env`)
- Story caching for instant detail level switching
- Comprehensive error handling and loading states

# Conclusion

## Achievements:

- Successfully integrated real-time government data with modern web interface
- Enhanced raw sensor data with AI-generated contextual explanations
- Created accessible tool for both technical and non-technical users
- Demonstrated effective use of React, TypeScript, and AI technologies

## Future Enhancements:

- Additional data sources (weather, precipitation forecasts)
- Predictive flood modeling capabilities
- Extended historical trend analysis
- Mobile-optimized responsive interface
- User alerts for critical water level changes