Florida Flood Gauge Monitor: A Real-Time Interactive Web Application

Paul Fishwick and Claude Code

January 4, 2026

Abstract

The Florida Flood Gauge Monitor is a web-based application that provides real-time visualization and AI-powered analysis of flood gauge data across Florida. The system integrates live data from the United States Geological Survey (USGS) Water Services API with interactive mapping capabilities and Google Gemini AI to deliver contextual explanations of water level fluctuations. This report describes the system architecture, key features, and technical implementation of the application.

1 Introduction

Flood monitoring is critical for public safety, infrastructure management, and environmental research in Florida, where water levels can fluctuate significantly due to tides, rainfall, and seasonal variations. The Florida Flood Gauge Monitor addresses the need for accessible, real-time flood data visualization by providing an interactive web interface that displays active USGS monitoring stations across the state.

The application, shown in Figure 1, combines modern web technologies with artificial intelligence to transform raw sensor data into actionable insights. Users can explore gauge locations on an interactive map, view historical water level trends through dynamic charts, and receive AI-generated explanations that contextualize the observed fluctuations based on local geography, weather patterns, and tidal influences.

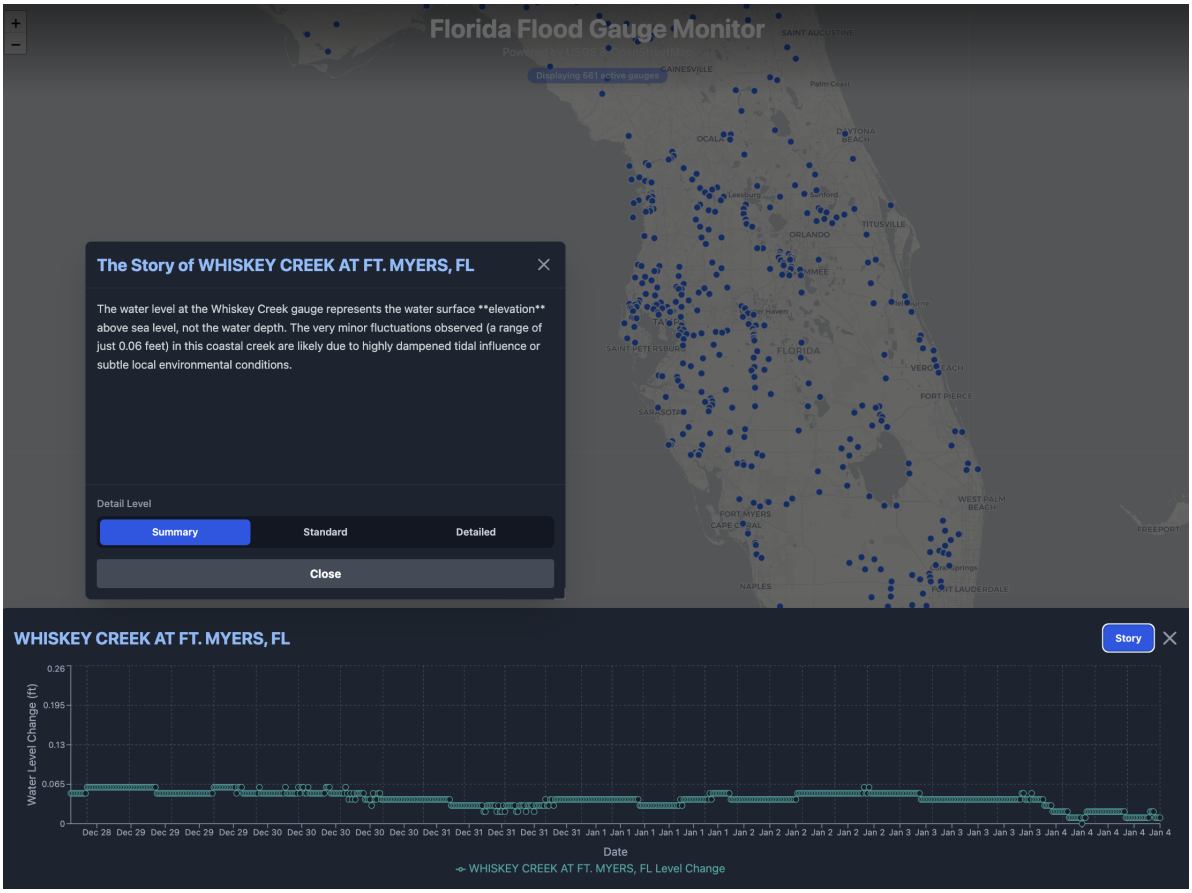

Figure 1: The Florida Flood Gauge Monitor interface showing the interactive map with gauge locations, a selected gauge's historical data chart, and the AI-generated story panel explaining water level fluctuations.

2 System Architecture

The Florida Flood Gauge Monitor is built as a modern single-page application using React and TypeScript, with Vite serving as the build tool and development server. The architecture follows a component-based design pattern that separates concerns and promotes code reusability.

2.1 Frontend Components

The application consists of four primary React components:

- **App Component**: The main application container that manages global state, including the list of active gauges, selected gauge information, loading states, and error handling. It orchestrates data fetching and coordinates interactions between child components.
- **Map Component**: Renders an interactive OpenStreetMap using the Leaflet library and React-Leaflet bindings. Each active gauge is displayed as a marker at its geographic coordinates. User clicks on markers trigger gauge selection events that propagate to the parent component.

- **GaugeChart Component**: Visualizes historical water level data using the Recharts library. The component displays a time-series line chart showing gage height measurements over the previous seven days, allowing users to identify trends and patterns in water elevation.
- **GaugeStory Component**: Presents AI-generated explanations of water level behavior. Users can toggle between three detail levels (summary, standard, detailed) to receive explanations ranging from one-sentence summaries to multi-paragraph analyses.

2.2 Data Flow

The application follows a unidirectional data flow pattern. On initialization, the App component fetches gauge data from the USGS API through the `geminiService` module. The retrieved data is stored in component state and passed down to child components as props. User interactions, such as clicking a gauge marker, trigger callback functions that update the parent state, causing dependent components to re-render with updated information.

3 Key Features

3.1 Live USGS Data Integration

The application retrieves real-time data from the USGS Water Services API, specifically requesting gage height measurements (parameter code 00065) for all active monitoring sites in Florida. The API provides the last seven days of data for each gauge, enabling trend analysis and pattern recognition.

The USGS response is a complex nested JSON structure that is transformed into a simplified internal data model. Each gauge record includes a unique identifier, site name, geographic coordinates (latitude and longitude), and an array of historical data points consisting of timestamps and water level measurements.

3.2 Interactive Mapping

The map interface provides an intuitive way to explore gauge locations across Florida. OpenStreetMap tiles serve as the base layer, offering familiar geographic context including coastlines, rivers, lakes, and urban areas. Gauge markers are clustered when zoomed out and expand to individual points as users zoom in, preventing visual clutter while maintaining comprehensive coverage.

3.3 Historical Data Visualization

When a user selects a gauge, a slide-up panel displays a line chart of water level measurements over the previous week. The chart uses a responsive design that adapts to different screen sizes, with the x-axis showing dates and the y-axis representing gage height in feet above sea level. This visualization helps users quickly identify trends such as tidal patterns, flood events, or seasonal variations.

3.4 AI-Powered Story Generation

The Story feature represents a significant enhancement over traditional flood monitoring systems. When activated, the application sends the selected gauge's historical data to Google Gemini AI along with a carefully crafted prompt that instructs the model to act as a hydrologist and science communicator.

The AI analyzes the data and generates explanations that consider multiple factors:

- Water body type (river, coastal lagoon, lake, swamp)
- Tidal influences for coastal systems
- Recent precipitation or drought conditions
- The magnitude of observed fluctuations
- Regional topography affecting base elevation

Three detail levels are available: *summary* (1-2 sentences), *standard* (single paragraph), and *detailed* (2-3 paragraphs). The application pre-fetches all three levels concurrently when the Story button is clicked, ensuring instant toggling between detail levels without additional API calls.

4 Technical Implementation

4.1 Technology Stack

The application leverages the following technologies:

- **React 19**: Component-based UI framework with hooks for state management
- **TypeScript 5**: Static type checking for improved code reliability
- **Vite 5**: Fast build tool and development server with hot module replacement
- **Leaflet 1.9**: Open-source JavaScript library for interactive maps
- **React-Leaflet 5**: React bindings for Leaflet
- **Recharts 3**: Composable charting library built on React components
- **Google Gemini API**: AI model for natural language generation

4.2 Environment Configuration

The application uses Vite's environment variable system to manage sensitive configuration such as API keys. Environment variables prefixed with `VITE_` are exposed to the client-side code through `import.meta.env`. The Gemini API key is stored in a `.env.local` file and accessed as `import.meta.env.VITE_GEMINI_API_KEY`, ensuring the key is embedded during build time while remaining excluded from version control.

4.3 Error Handling and Loading States

The application implements comprehensive error handling to provide a robust user experience. Network failures, API errors, and invalid responses trigger error messages displayed in modal overlays. Loading indicators appear during data fetching operations, providing visual feedback that operations are in progress. A retry mechanism allows users to re-attempt failed requests without reloading the entire application.

4.4 Caching Strategy

To minimize API calls and reduce latency, the application caches AI-generated stories in component state. The cache is structured as a nested object keyed by gauge ID and story level, allowing instant retrieval of previously generated content. This approach significantly improves responsiveness when users switch between different detail levels or revisit previously explored gauges.

5 Conclusion

The Florida Flood Gauge Monitor demonstrates the potential of combining real-time government data, interactive web technologies, and artificial intelligence to create accessible and informative public resources. By transforming raw sensor measurements into visual representations and human-readable explanations, the application makes flood monitoring data more accessible to non-specialists while maintaining the detail required for technical users.

Future enhancements could include additional data sources, predictive flood modeling, historical trend analysis beyond seven days, and mobile-optimized interfaces. The modular architecture and modern technology stack position the application for continued evolution and feature expansion.