# Water Level Visualizer - Mobile Web App

A mobile-optimized web application for visualizing groundwater monitoring data, built as a companion to the desktop Water Level Monitoring system.

## 🎯 Overview

This mobile web app provides a simplified, touch-friendly interface for accessing and visualizing water level monitoring data on mobile devices. It's designed for field workers and researchers who need quick access to monitoring data while away from their desktop workstations.

## ✨ Key Features

### 📊 **Data Visualization**

* **Interactive Charts**: Touch-optimized time-series plots using Recharts
* **Multi-Data Types**: Water levels, temperature, and manual readings
* **Responsive Design**: Optimized for phones, tablets, and desktop browsers
* **Touch Gestures**: Pan, zoom, and tap interactions for mobile devices

### 🗄️ **Database Management**

* **File Upload**: Drag-and-drop SQLite database upload with validation
* **Multiple Databases**: Switch between different project databases
* **Data Persistence**: Local storage for offline access to uploaded databases
* **Smart Caching**: Efficient data loading and caching strategies

### 🔍 **Well Browser**

* **Advanced Search**: Filter wells by number, CAE, field, and data availability
* **Sortable Columns**: Sort by well number, CAE number, or last reading date
* **Pagination**: Handle large datasets with mobile-friendly pagination
* **Data Indicators**: Visual badges showing available data types

### 📈 **Chart Controls**

* **Date Range Selection**: Presets (30 days, 3 months, 1 year) and custom ranges
* **Data Type Toggles**: Show/hide water levels, temperature, manual readings
* **Navigation Tools**: Brush control for navigating large datasets
* **Zoom and Pan**: Interactive chart exploration

### 🧮 **Recharge Results** (Read-Only)

* **Method Display**: View RISE, MRC, and EMR calculation results
* **Historical Data**: Access to all previous calculations
* **Method Comparison**: Side-by-side comparison of different methods
* **Export Options**: Download results in multiple formats

## 🛠️ Technical Stack

### **Frontend**

* **Framework**: Next.js 14 with App Router
* **Language**: TypeScript for type safety
* **Styling**: Tailwind CSS with mobile-first design
* **Charts**: Recharts for interactive visualizations
* **State Management**: React hooks and context

### **Database**

* **Engine**: SQLite with sql.js for in-browser operation
* **Schema**: Compatible with desktop Water Level Monitoring system
* **Operations**: Read-only access with efficient querying
* **Caching**: Client-side caching for performance

### **Performance**

* **Code Splitting**: Automatic route-based code splitting
* **Data Optimization**: Smart downsampling for large datasets
* **Caching**: Multi-layer caching (browser, localStorage, memory)
* **Bundle Size**: Optimized for fast loading on mobile networks

## 🚀 Getting Started

### **Prerequisites**

* Node.js 18+ and npm
* SQLite database files from the Water Level Monitoring system

### **Installation**

# Install dependencies

npm install

# Start development server

npm run dev

# Build for production

npm run build

### **Development**

# Start dev server on custom port

npm run dev -- --port 3001

# Run type checking

npm run build

# Check for linting issues

npm run lint

## 📱 Usage Guide

### **1. Upload Database**

1. Open the app in your mobile browser
2. Drag and drop a .db or .sqlite file from your Water Level Monitoring system
3. Wait for validation and processing
4. The database will be stored locally for offline access

### **2. Browse Wells**

1. Select a database from the homepage
2. Use search and filters to find specific wells
3. View well metadata and data availability
4. Tap "View Data" to see charts

### **3. Visualize Data**

1. Adjust chart controls to customize the view
2. Select date ranges and data types
3. Use touch gestures to interact with charts
4. Export data when needed

### **4. View Recharge Results**

1. From a well's plot page, tap "Recharge"
2. View calculation summaries and detailed results
3. Compare different calculation methods
4. Export results for further analysis

## 🗂️ Project Structure

src/

├── app/ # Next.js App Router pages

│ ├── page.tsx # Homepage (database selection)

│ ├── wells/[id]/ # Wells browser

│ │ ├── page.tsx # Wells list and search

│ │ ├── plot/[wellNumber]/ # Data visualization

│ │ └── recharge/[wellNumber]/ # Recharge results

├── components/ # Reusable React components

│ ├── DatabaseUpload.tsx # File upload with validation

│ ├── DatabaseSelector.tsx # Database selection interface

│ ├── WellBrowser.tsx # Wells table with search/filter

│ ├── WaterLevelChart.tsx # Interactive chart component

│ ├── ChartControls.tsx # Chart configuration controls

│ └── LoadingSpinner.tsx # Loading states

├── lib/ # Core utilities and database logic

│ └── database.ts # SQLite operations with sql.js

├── types/ # TypeScript type definitions

│ └── database.ts # Database entity interfaces

├── utils/ # Helper functions

└── styles/ # Global styles and themes

## 💾 Database Compatibility

The app is fully compatible with SQLite databases created by the desktop Water Level Monitoring system:

### **Required Tables**

* wells - Well metadata and locations
* water\_level\_readings - Transducer and telemetry data
* manual\_readings - Manual water level measurements

### **Optional Tables**

* rise\_results - RISE method recharge calculations
* mrc\_results - MRC method recharge calculations
* emr\_results - EMR method recharge calculations
* transducers - Transducer metadata
* barologgers - Barometric pressure logger data

### **Data Types Supported**

* **Transducer Data**: Continuous water level and temperature readings
* **Manual Readings**: Field-measured water levels
* **Telemetry Data**: Remote monitoring system data
* **Recharge Calculations**: Results from various calculation methods

## 🔧 Configuration

### **Environment Variables**

Create a .env.local file for local development:

# Development settings

NEXT\_PUBLIC\_APP\_NAME="Water Level Visualizer"

NEXT\_PUBLIC\_MAX\_FILE\_SIZE="104857600" # 100MB in bytes

### **Build Configuration**

The app is configured for both development and production deployment:

* **Development**: Full Next.js server with hot reload
* **Production**: Optimized build with static generation where possible
* **Deployment**: Compatible with Netlify, Vercel, and other platforms

## 📊 Performance Features

### **Data Optimization**

* **Smart Downsampling**: Automatically reduces data points for large datasets
* **Lazy Loading**: Components and data loaded on demand
* **Caching**: Multiple levels of caching for fast repeat access
* **Compression**: Efficient data storage and transfer

### **Mobile Optimization**

* **Touch Targets**: Minimum 44px touch targets for accessibility
* **Responsive Design**: Fluid layouts that work on all screen sizes
* **Offline Capability**: Local storage for uploaded databases
* **Fast Loading**: Optimized bundle sizes and loading strategies

## 🔮 Future Enhancements

### **Phase 4: Data Export System**

* CSV export with custom filters
* JSON export for data interchange
* PDF report generation
* Batch export capabilities

### **Phase 5: Advanced Features**

* PWA (Progressive Web App) functionality
* Offline-first architecture
* Data synchronization with cloud systems
* Advanced statistical analysis tools

### **Phase 6: Integration**

* Direct cloud database connectivity
* Real-time data updates
* User authentication and permissions
* Integration with main monitoring system

## 🤝 Contributing

This mobile web app is designed to complement the desktop Water Level Monitoring system. For contributions:

1. Maintain compatibility with existing database schemas
2. Follow mobile-first design principles
3. Ensure accessibility and performance standards
4. Test on multiple devices and browsers

## 📄 License

This project is part of the Water Level Monitoring system and follows the same licensing terms.

## 🆘 Support

For technical support:

1. Check the console for error messages
2. Verify database file format and structure
3. Ensure browser compatibility (modern browsers required)
4. Clear browser cache if experiencing issues

**Note**: This mobile app provides read-only access to water level monitoring data. For data collection, processing, and advanced analysis, use the desktop Water Level Monitoring application.