# Turso Integration Complete - Deployment Guide

## ✅ What's Been Implemented

Your mobile visualizer has been successfully upgraded to use Turso instead of Google Drive + SQLite:

### 1. **Database Successfully Uploaded**

* ✅ Database: caeser-water-monitoring
* ✅ URL: libsql://caeser-water-monitoring-benjaled.aws-us-east-1.turso.io
* ✅ 44 wells, 180,293 water level readings
* ✅ Connection tested and working

### 2. **Turso Service Created**

* ✅ src/lib/api/services/turso.ts - Complete service implementation
* ✅ All required functions: getWells(), getWell(), getWaterLevelData(), etc.
* ✅ Matches actual database schema
* ✅ Tested and functional

### 3. **Netlify Functions Updated**

* ✅ databases.ts - Updated to use Turso
* ✅ wells.ts - Updated to use Turso
* ✅ data.ts - Updated to use Turso
* ✅ All Google Drive dependencies removed
* ✅ Caching preserved for performance

### 4. **Dependencies Added**

* ✅ @libsql/client package installed
* ✅ Environment configuration ready

## 🚀 Deployment Steps

### Step 1: Set Environment Variables in Netlify

In your Netlify dashboard, go to **Site settings > Environment variables** and add:

TURSO\_DATABASE\_URL=libsql://caeser-water-monitoring-benjaled.aws-us-east-1.turso.io

TURSO\_AUTH\_TOKEN=eyJhbGciOiJFZERTQSIsInR5cCI6IkpXVCJ9.eyJpYXQiOjE3NTA4MDg2MDcsImlkIjoiNTgwNDQ1MDgtOGQwNi00M2QwLTg4NzItNGI5NGFhODY0OTY0IiwicmlkIjoiZGJiNjc3M2ItNjZiMi00YThkLWExMmItOGE4MmMwZDQxNDU1In0.zb9Z6iCYS4OUKeoRyajOeU4GRVh6xgFk4YErNYzR6M8DX0CFcJhifdMhXHYPCj3RmVkWJbrYfpWSZaFc3xNPBg

### Step 2: Deploy to Netlify

npm run build

# Deploy via Netlify CLI or push to your connected GitHub repo

### Step 3: Test API Endpoints

After deployment, test these endpoints:

* GET /.netlify/functions/databases - List databases
* GET /.netlify/functions/wells/caeser-water-monitoring - Get wells
* GET /.netlify/functions/data/caeser-water-monitoring/water/TN157\_000364 - Get water data

## 📊 What's Different Now

### Before (Google Drive + SQLite):

1. User uploads SQLite file to web app
2. Functions download file from Google Drive
3. Open SQLite file locally
4. Query data
5. Close file

### After (Turso):

1. User accesses web app directly
2. Functions query Turso cloud database instantly
3. No file uploads or downloads needed
4. Faster, more reliable, always up-to-date

## 🔧 Available API Functions

### Database Functions:

* getDatabaseStats() - Get wells count, readings count, last update
* listDatabases() - Returns info about the Turso database

### Wells Functions:

* getWells(params) - Get paginated wells list with search/filter
* getWell(wellNumber) - Get specific well details
* getWellFields() - Get unique field names

### Data Functions:

* getWaterLevelData(params) - Get time-series data with downsampling
* getRechargeResults(wellNumber) - Get calculation results
* getDataSummary(wellNumber) - Get well data summary

## 📋 Database Schema Used

**Wells Table:**

* well\_number, cae\_number, well\_field, cluster
* latitude, longitude, top\_of\_casing, aquifer

**Water Level Readings:**

* id, well\_number, timestamp\_utc, julian\_timestamp
* water\_level, temperature, baro\_flag, level\_flag

**Recharge Calculations:**

* rise\_calculations, mrc\_calculations, erc\_calculations

## 🎯 Next Steps

1. **Set Netlify environment variables** (Step 1 above)
2. **Deploy the application**
3. **Test the mobile visualizer app**
4. **Your mobile app now works with live cloud data!**

## 🛠️ Maintenance

* **Database updates**: Upload new data to Turso using turso db shell
* **New features**: Add functions to TursoService class
* **Schema changes**: Update type definitions and mapping functions

The mobile visualizer is now cloud-connected and ready for production use!