BluElephant Water Analyzer

BluElephant Water Analyzer is a web-based tool supporting the BluElephant decentralized wastewater treatment system by enabling intuitive tracking and analysis of water sample data.  
  
Developed as part of the European Project Semester (EPS) at Saxion University, this app contributes to improving the usability and performance of the experimental BluElephant solution: https://bluelephant.global.

# Features

* Store and manage samples from influent, effluent, and sludge.
* Dynamic parameter analysis with charts and filters.
* Export data to Excel or CSV.
* Prototype prediction module (Random Forest) to flag samples exceeding legal thresholds.
* JWT-secured login with multiple user accounts.
* Historical measurements view with legal limit overlays.

# Technologies

* Backend: FastAPI (Python)
* Database: PostgreSQL
* Machine Learning: Random Forest Classifier (prototype feature)
* Frontend: React + Tailwind CSS

Note: The prediction module is a prototype and should not be used for critical decisions.

# Installation Guide

## Prerequisites

* Python 3.x
* Node.js with npm
* PostgreSQL with pgAdmin

## Step 1 – Create Database

Using pgAdmin 4 create new database called BluElephantApp.

## Step 2 – Clone the Repository

If you don’t have the project folder on your computer yet, you can clone it from GitHub:

git clone https://github.com/MiloszDalek/water\_quality\_app.git  
cd water\_quality\_app

## Step 3 – Backend Setup (FastAPI)

Run the following commands in PowerShell:  
cd backend  
python -m venv venv  
source venv/bin/activate (On Windows: venv\Scripts\activate)  
pip install -r requirements.txt

In backend directory create a .env file with the following configuration:

DATABASE\_URL=postgresql://user:password@localhost:5432/BluElephantApp  
SECRET\_KEY=supersecretkeysupersecretkeysupersecretkey  
ALGORITHM=HS256

*Replace ‘user’ and ‘password’ with your login credentials to postgreSQL.*

Run the backend:  
uvicorn main:app --reload --host 0.0.0.0 --port 8000  
Backend available at http://localhost:8000

## Step 4 – Frontend Setup (React)

cd frontend

In frontend directory create a .env file with the following configuration:

VITE\_API\_URL=http://localhost:8000

npm install  
npm start  
Frontend available at http://localhost:5173

## Step 5 – Run Application (windows only)

After completing all setup steps, you can start the application by running *run-BE-App.bat*. Before using this automatic start for the first time, you need to enable script execution by running the following command in PowerShell:

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned

# Usage Guide

1. Login/Register – First user can be created via API or seed script. JWT authentication is required for adding or modifying samples.
2. Add Samples – Choose a category: influent, effluent, or sludge. Fill in parameters and save.
3. Analyze Data – Use filters and charts to explore parameters. Toggle legal limits overlay for validation.
4. Export Data – Export the current dataset to Excel or CSV.
5. Prediction (Prototype) – Use the ML module to check if samples are close to exceeding legal thresholds.

# Developer Guide

## Project Structure

water\_quality\_app/  
├── backend/ # FastAPI + ML logic  
├── frontend/ # React + Tailwind UI  
├── docs/ # Project documentation  
└── README.md

## Updating the Database

• Modify models inside /backend/models/  
• Apply migrations (e.g., with Alembic)  
• Update frontend forms if parameters change

## Updating the Machine Learning Module

• Model training code is inside /model\_training\_scripts directory  
• Retrain with new datasets  
• Replace the saved model file

## JWT and Authentication

• The secret key is configured in the .env file  
• Tokens are issued upon login and attached to frontend requests

# Project Background

This application was developed during the European Project Semester (EPS) at Saxion.  
  
Future development suggestions:  
• Improve UI/UX  
• Enhance prediction accuracy with better datasets and better model training  
• Extend the database schema to support additional parameters