

AutoDBA Web – Technical Architecture

1. System Architecture Overview

The AutoDBA Web MVP is structured into three main layers:

- React frontend (TypeScript + MUI) for the user interface.
- Node.js + Express backend that exposes RESTful APIs.
- Python 3 audit engine that performs analysis on the connected MySQL database.

Each layer communicates securely and cleanly to ensure modularity and scalability.

2. Tool Stack

Frontend:

- React + TypeScript
- @mui/material, @mui/icons-material
- react-router-dom, react-hook-form
- axios, jwt-decode, notistack, recharts

Backend:

- Express.js, mysql2
- jsonwebtoken, bcryptjs, dotenv, cors
- child_process, winston (optional), eslint + prettier

Audit Engine (Python):

- Python 3.11+, pymysql, pandas
- json, collections, typing (built-in modules)

Development Tools:

- Vite, Nodemon, Postman, Git + GitHub, .env, Prettier + ESLint

3. Final Folder Structure

AutoDBA-Web/

```
├── frontend/           # React App
|   ├── public/
|   └── src/
|       ├── assets/, components/, pages/, sections/, context/, services/
|       ├── hooks/, types/, router/, App.tsx, main.tsx
├── backend/           # Express API
|   ├── routes/, controller/, services/, middlewares/, models/, utils/
|   ├── app.js, server.js, .env
├── audit-engine/      # Python Audit Scripts
|   ├── audit_schema.py, audit_indexes.py, audit_constraints.py, utils.py
├── docs/              # Documentation, ERDs, Screenshots
├── .gitignore, README.md
```

4. Environment Variables

.env (backend):

- PORT
- JWT_SECRET
- PYTHON_ENGINE_PATH

.env (frontend):

- VITE_API_URL

5. Summary

This technical architecture ensures modular, testable, and scalable implementation. Each component has a clear responsibility and minimal coupling, enabling quick iteration and future extensibility. All tools chosen are open-source, actively maintained, and compatible.