

Technical Guide to Relive System

System Components

Database:

The system Uses Docker to create container with a MariaDB instance which it uses to store the data. This instance uses a docker volume to store the data locally. MariaDB is based on the MySQL language and was designed to work smoothly with docker instances.

System Recommendation Model:

The Prediction Model is built using Python and uses the Random Forest Model to choose which of the existing dietary plans is best to recommend.

Backend:

The system uses a NodeJS server service. It has proven reliability and flexibility as an industry standard.

The system uses the npm package "mysql" for communicating with the Database. It also uses the "express" npm package to create an HTTP API layer to communicate with the Frontend.

Given that the Prediction Model is built with python, the Backend uses "child_process" with the "spawn" function to run the python files locally.

Server Management:

The Database being a docker container allows us to easily integrate it into any docker compatible cloud infrastructure. In our testing we found it was possible to combine the Database with our backend service quite easily.

This allows for flexibility regarding the hosting choice based on the customers needs.

Frontend:

The system uses Angular to build a frontend webapp. It also uses Tailwind and DaisyUI for CSS styling and a unified design philosophy.

Setup and Installation

The system can be run locally as a local development environment.

To run it requires the following programs:

- Docker
- NodeJS
- Angular
- Python

If those programs are installed you can run the project. Start by Extracting the Project files in a folder. The folder should contain these 3 folders: "relive-webapp", "relive-express", "relive-db".

You can use VSCode to open the folder and manage the terminals inside. This guide will continue assuming the user does not use VSCode.

Open a commandline/terminal with a path to the extracted folder. This can be done in windows by typing CMD in the folder path bar in file explorer.

Otherwise use the following command in a commandline `cd FULL_FOLDER_PATH`. Replace the `FULL_FOLDER_PATH` with the path to the folder you extracted into.

We will run each section seperately and in order. beginning with the Database and ending with the frontend.

Database

If the commandline is in the correct folder the command `cd relive-db` should run smoothly.

To start up the database container run:

```
docker compose up
```

This may take a minute to start up but after it does you can continue to the Backend section.

You can use Docker desktop from this point on to turn it off.

To switch off in the terminal use CTRL+C to turn it off.

For good practice use the following command after you turn the container off:

```
docker compose down
```

Backend

Open a second commandline/terminal much like the first but this time navigate to the folder "relive-express" using the command `cd relive-express`.

Requires NodeJS and npm. in npm it requires the packages 'express', 'mysql', 'child_process' and 'jsonwebtoken' to run. We recommend installing 'nodemon' from npm as well.

you can install Packages with the following command:

```
npm install PACKAGE_NAME
```

If you have nodemon installed simply run:

```
nodemon
```

if you dont run the following:

```
node server.js
```

This should use port 8080 by default and can be accessed via <http://localhost:8080>
if this port is changed then the file `..\relive-webapp\src\app\services\baseURL.ts` must be changed accordingly.

both can be stopped with Ctrl+C. And require no further steps.

Frontend

Requires NodeJS and npm, Along with Angular being installed on the machine.

Open a third commandline/terminal much like the first but this time navigate to the folder "relive-webapp" using the command `cd relive-webapp`.

If this is its first time you start it use the following command. Otherwise you can skip this step:

```
npm install
```

To start the website use the command:

```
ng serve
```

This should use port 4200 by default and can be accessed via <http://localhost:4200>

You can use Ctrl+C to stop it.