This documentation covers how to install and run the solution, how to see and use the results, the architecture overview, and possible next steps.

### Installation and Setup

To install and run the solution, follow the steps below:

1. Ensure that you have Docker installed and running on your machine.

2. Clone or download the code repository to your local machine(<https://github.com/Aswinjoseph2011/Data_Enginnering_Test.git>).

3. Open a terminal or command prompt and navigate to the directory containing the code files.

4. Build the Docker image by running the following command:

```

docker build -t my-fastapi-app .

```

5. After the image is built successfully, run the Docker container with the provided FastAPI parameters using the following command:

```

docker run -p 8000:8000 my-fastapi-app

```

Make sure to replace the values of the `-e` options (`HOST`, `PORT`, `DATABASE`, `USERNAME`) as per your MySQL server configuration.

6. Once the container is running, you can access the FastAPI application at http://localhost:8000.

### Viewing and Using the Results

The solution processes JSON files, extracts specific patient data, and populates a MySQL database table with the extracted information.

To view and use the results, follow these steps:

1. Place the JSON files you want to process in the same directory as the code files.

2. Use web browser to send an HTTP POST request to ` http://127.0.0.1:8000/docs `.

3. Include the required parameters in the request:

- `host`: The hostname or IP address of the MySQL server.

- `port`: The port number on which the MySQL server is listening.

- `database`: The name of the MySQL database where the table will be created.

- `username`: The username to authenticate with the MySQL server.

Attaching my screenshot for easy reference.

A screenshot of a computer

Description automatically generated

4. Send the request and wait for a response. The response will indicate whether the data conversion and database population were successful.

5. Check your MySQL database to see the populated table. The table name will be `patient\_data`.

Attaching an image for reference

A screenshot of a computer

Description automatically generated

### Architecture Overview

The solution is built using Python, FastAPI, Pandas, SQLAlchemy, and Docker. Here's an overview of the architecture:

1. FastAPI: The FastAPI framework is used to create a RESTful API for handling HTTP requests.

2. DataFrameToMySQL: The `DataFrameToMySQL` class converts JSON files to a Pandas DataFrame, extracts specific patient data, and populates a MySQL database table using SQLAlchemy.

3. MySQL Server: The MySQL database server stores the patient data in a table named `patient\_data`.

4. Docker: The solution is containerized using Docker, which provides a consistent and isolated environment for running the application.