# FastAPI Project Requirement Document

# FastAPI Project Requirement Document

## \*\*Project Title:\*\* FastAPI Data Management System using Pandas and SQLite

### \*\*Objective:\*\*

Develop a simple web application using \*\*FastAPI\*\* that serves data from two different sources:

1. \*\*Pandas DataFrame\*\* - Contains a small product dataset stored in memory.

2. \*\*SQLite Database\*\* - Stores user information using SQLAlchemy.

The project will also include a web interface to test the APIs.

---

## \*\*Project Features and Requirements:\*\*

### ✅ \*\*1. API Features\*\*

* - \*\*Home Page\*\*: Serve a simple HTML page to test the API.
* - \*\*Dataframe API\*\*: Endpoint to return data from a Pandas DataFrame in JSON format.

- \*\*GET\*\* `/dataframe`

* - \*\*User Management APIs\*\*:

- \*\*GET\*\* `/users` → Retrieve all users from SQLite.

- \*\*POST\*\* `/users` → Add a new user (requires `name` and `email`).

### ✅ \*\*2. Data Management\*\*

* - \*\*Pandas DataFrame\*\*: The app will create and store a sample product dataset using Pandas.

- Fields: `id`, `product`, `price`

* - \*\*SQLite Database\*\*: SQLAlchemy ORM will be used to manage user data.

- Fields: `id`, `name`, `email`

### ✅ \*\*3. Web Interface\*\*

* - Simple frontend using HTML and CSS.
* - Provides interactive buttons to test API responses.

---

## \*\*Tech Stack\*\*

* - \*\*Backend\*\*: FastAPI
* - \*\*Frontend\*\*: HTML, CSS (Bootstrap optional)
* - \*\*Database\*\*: SQLite
* - \*\*ORM\*\*: SQLAlchemy
* - \*\*Data Management\*\*: Pandas

---

## \*\*Folder Structure\*\*

FastAPI-Project/

├── main.py # FastAPI application with routes

├── database.py # SQLite database setup using SQLAlchemy

├── data.csv # Optional: Data for Pandas DataFrame

├── templates/

│ └── index.html # Frontend HTML for testing

├── static/

│ └── style.css # Optional: Stylesheet for frontend

└── requirements.txt # List of project dependencies

---

## \*\*API Endpoints Summary\*\*

| HTTP Method | Endpoint | Description | Parameters |

|--------------|----------------|-----------------------------------------|-------------------|

| GET | `/` | Serve the HTML homepage | None |

| GET | `/dataframe` | Get product data from Pandas DataFrame | None |

| GET | `/users` | Retrieve all users from SQLite | None |

| POST | `/users` | Add a new user to SQLite | `name`, `email` |

---

## \*\*Submission Guidelines:\*\*

* - Provide a zip file containing the complete project folder.
* - Ensure the app runs without errors using the command:

bash

uvicorn main:app --reload

* - Provide API screenshots for each endpoint response.

---

## \*\*Bonus Tasks (Optional)\*\*

* - Add error handling for invalid data submissions.
* - Implement an API to delete users.
* - Display API responses on the web interface.

Good luck with your project!