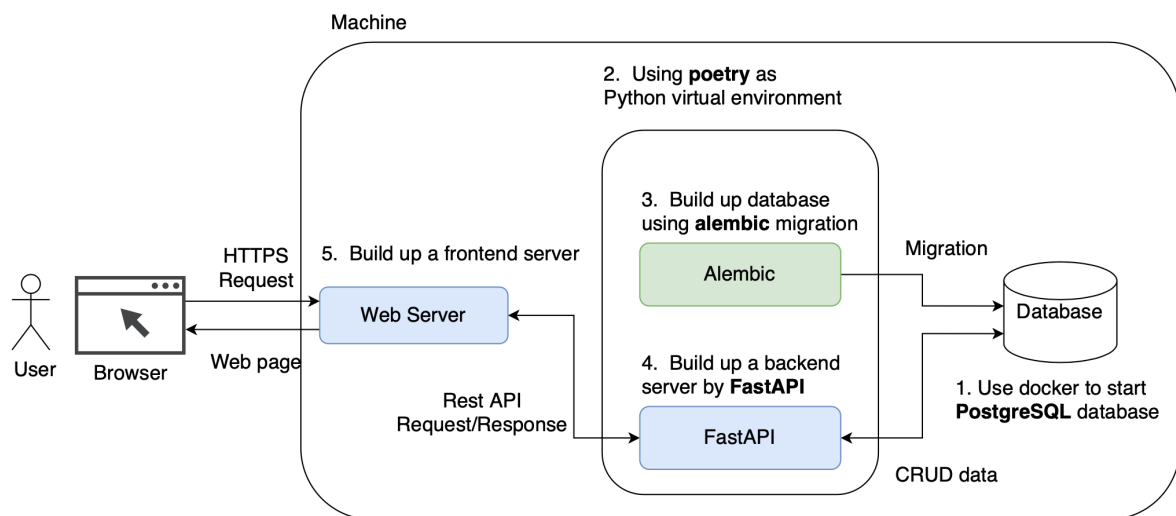




Practice



1. Use docker to start PostgreSQL database

- 目標: 使用 Docker 啟動 PostgreSQL 資料庫並建立資料庫
- Tutorial: <https://migueldoctor.medium.com/how-to-run-postgresql-pgadmin-in-3-steps-using-docker-d6fe06e47ca1>

2. Using poetry as Python virtual environment

- 目標: 使用 poetry 建立 python 虛擬環境
- Tutorial: <https://blog.kyomind.tw/python-poetry/>

3. Build up database using alembic migration

- 目標: 使用 alembic 建立 user table 並塞入資料, 格式如下:
 - username(varchar), primary key
 - password(varchar), not nullable
 - birthday(date)

- iv. `create_time(datetime)`, default: `datetime.utcnow()`
 - v. `last_login(datetime)`, nullable
 - b. Alembic Tutorial: <https://medium.com/@acer1832a/使用-alembic-來進行資料庫版本管理-32d949f7f2c6>
 - c. SQLAlchemy Tutorial: <https://www.maxlist.xyz/2019/10/30/flask-sqlalchemy/>
- 4. Build up a backend server by FastAPI
 - a. 目標：使用 FastAPI 建立 API server，提供 frontend 所需的相關 API，並使用 JWT 認證，可以對 `user` table 進行 CRUD，JWT 可透過 refresh API 進行更新
 - b. FastAPI Tutorial: <https://fastapi.tiangolo.com/>
 - c. JWT Tutorial: <https://fastapi.tiangolo.com/tutorial/security/oauth2-jwt/>
- 5. Build up a frontend server to query backend server and display it
 - a. 目標：建立一個前端伺服器，並可以跟透過 backend server 取得 user 資料並加以編輯
 - b. 操作：
 - i. 登入
 - ii. 讀取 user 資料並顯示在畫面上，其中 password 和 birthday 為可編輯狀態
 - iii. 按下 [Update] 按鈕更新 password 和 birthday 到後端 DB
 - iv. 登出
- 6. 提示
 - a. API 命名盡可能遵循 Restful 原則
 - b. 此練習重點在後端技術能力，前端 UI 從簡即可
 - c. (Optional) 為後端 API 加上 Unit Test