

**WEB**

**Flask core-ui 탐구**

### 1. clone 받기

```
git clone https://github.com/twintechlabs/flaskdash
```

### 2. Virtualenv 설정

(리눅스) `mkvirtualenv -p PATH/TO/PYTHON my_app`  
(windows) pycharm을 통한 interpreter 설정

### 3. pip를 통한 패키지 설치

```
pip install -r requirements.txt
```

### 4. DB 초기화

```
python manage.py init_db
```

### 5. 실행하기

```
python manage.py runserver
```

### 6. 테스트 코드 실행

```
py.test tests/
```

# localhost:5000으로 접속 시, 첫 화면

The screenshot shows a web browser window with the address bar displaying "localhost:5000/user/sign-in". The page features a light gray background with a central white and blue sign-in/sign-up form. The form is divided into two main sections: a white "Login" section on the left and a blue "Sign up" section on the right.

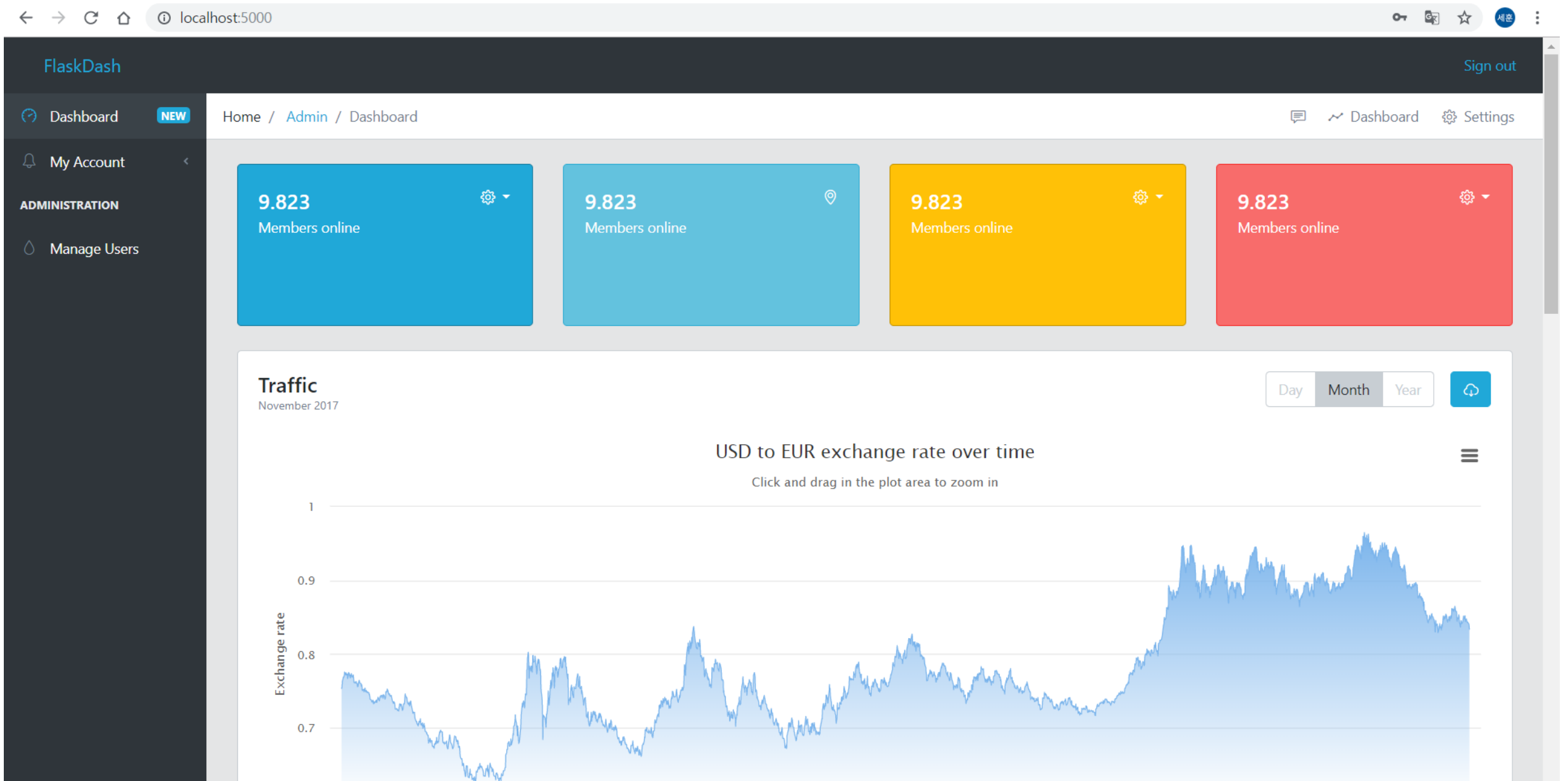
**Login Section:**

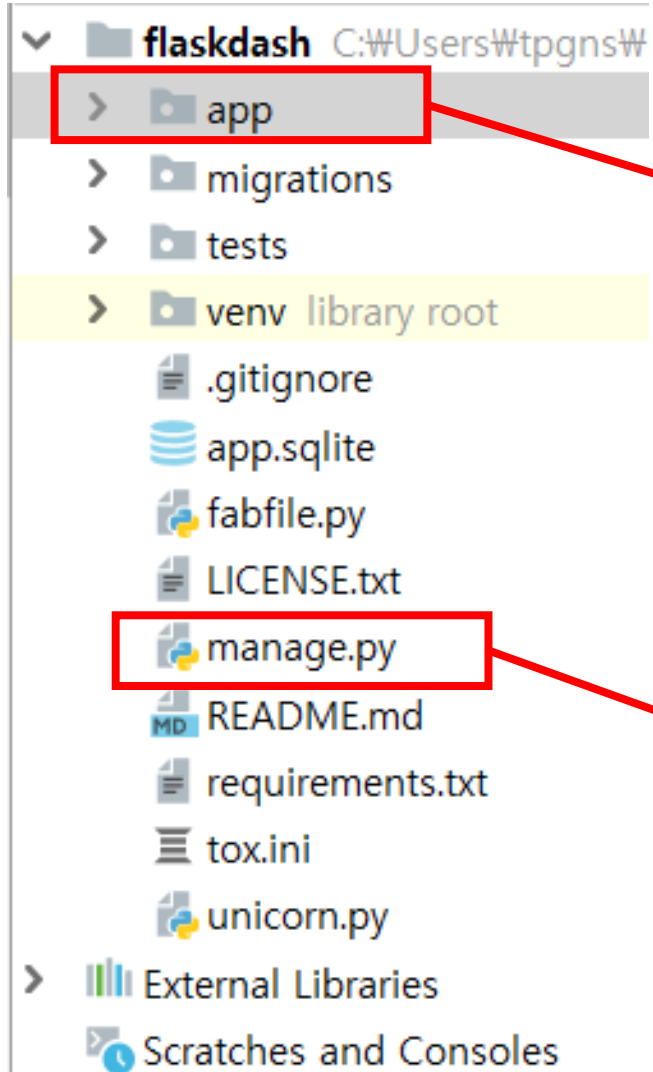
- Title: **Login**
- Subtitle: Sign In to your account
- Input fields: Two input fields, one for "Email" (with a person icon) and one for "Password" (with a lock icon).
- Buttons: A blue "Sign in" button and a link for "Forgot password?".

**Sign up Section:**

- Title: **Sign up**
- Text: Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
- Button: A blue "Register Now!" button.

# admin 계정으로 로그인 후의 화면



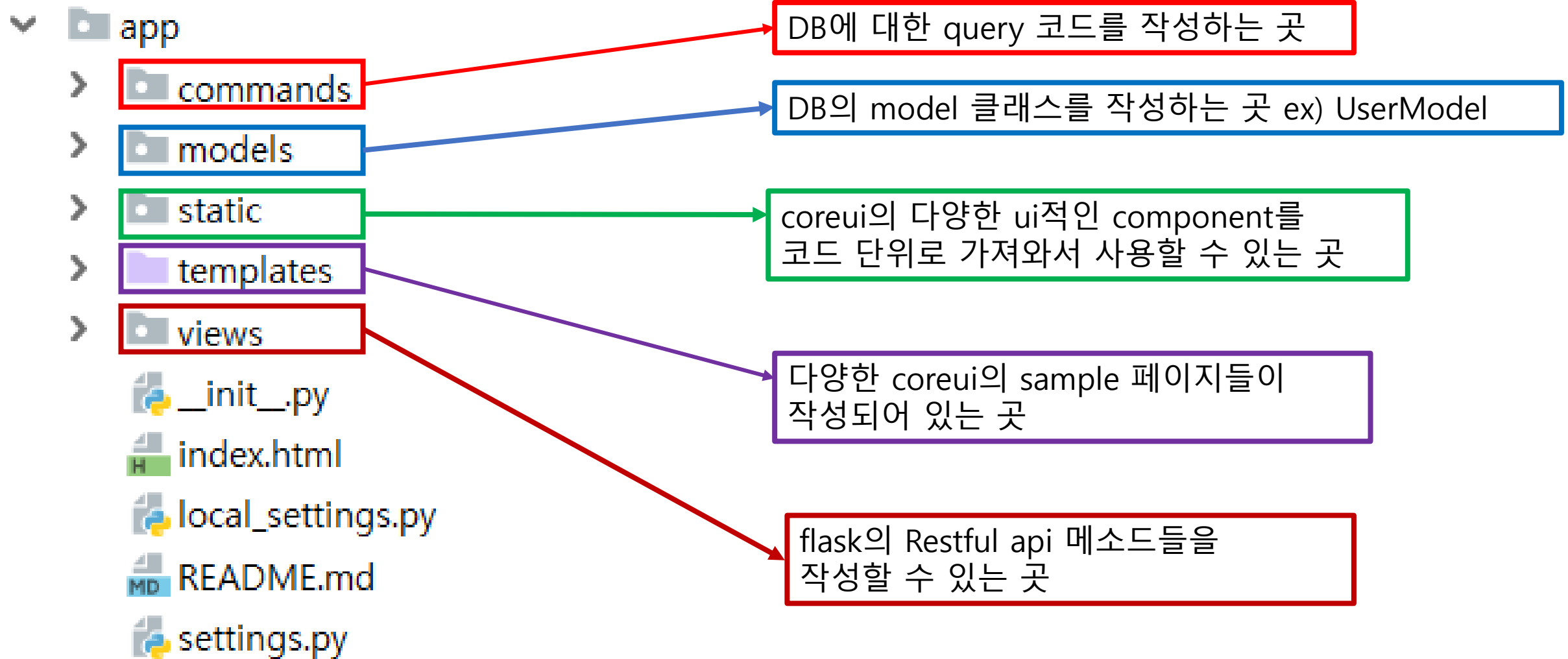


### 모든 코드가 담겨있는 곳

app 디렉토리 밑에 있는 코드 중에서  
불필요한 부분을 빼내어 사용하면 된다.

### 서버의 메인이 되는 파일

이 파일을 실행시킴으로써  
서버를 실행할 수 있다.



### React Core-UI

1. React 자체가 FrontEnd 프레임워크이기에 Restful API 작성이 불가능하다.
2. 또한 DB에 대한 어떠한 관여도 할 수가 없다.
3. 상당히 프로젝트 규모가 크기에 무겁다.
4. component 단위로 Chart, CardView 등의 다양한 기능을 사용할 수 있다.

### Flask Core-UI

1. Restful API 작성이 가능하다.
2. flask-coreui가 SQLAlchemy 패키지를 이용하여 mysql DB와 flask의 연동을 해두었다.
3. Front-Back-DB 모두 한번에 연동되었음에도 무겁지 않다.
4. stati과 templates에서 필요한 view를 가져와서 사용할 수 있다.



Flask-CoreUI를 사용해본 결과, “React가 이제는 필요 없겠구나”라는 생각을 했습니다.

한 프로젝트 내에서 view, rest API, DB를 모두 사용할 수 있다는 것에서 매력을 느꼈습니다.

DB의 User 모델도 이미 설계가 되어 있고 flask의 로그인 기능마저 coreui에서 구현해 놓았기에 정말 rest API 메소드 작성해만 치중하면 된다고 느꼈습니다. flask-coreui만 있으면 상당히 빠르게 웹 페이지를 만들 수 있습니다!