

# README.md

## 전체 플로우 설명

### 1. 사용자가 관리자 로그인 시

- views/admin\_login.ejs ⇒ 로그인 페이지 (사용자가 계정 정보 입력)
- 입력 후, routes/auth.js → /login ⇒ 로그인 처리
  - 입력된 이메일과 비밀번호로 인증 후 JWT 토큰 발급 -> localStorage 저장
  - 로그인 성공 후, /main 페이지로 리다이렉트 (JWT인증 필요)

### 2. 회원가입

- views/admin\_register.ejs ⇒ 회원가입 페이지 (사용자가 이메일 입력)
- "admin\_verifyemail" → public/register.js → /signup/email → 이메일 인증 요청 :
  - middlewares/emailAuth.js : 이메일 전송(인증코드 포함)
  - 인증코드가 이메일로 발송됨
  - views/admin\_verifyemail.ejs : 인증 코드 입력 페이지 오픈

### 3. 이메일 인증

- public/register.js → 이메일 인증코드 입력 후 /signup/email/verify로 요청
- middlewares/emailAuth.js → 인증 코드 확인 :
  - 코드 검증 후 인증 완료
  - 인증 완료되면 admin\_register.ejs 폼으로 이동

### 4. 관리자 회원가입

- public/register.js → 회원가입 양식 -> register/admin ⇒ 회원가입 처리 :
  - middlewares/registerForm.js ⇒ 회원가입 양식 검증
  - middlewares/registerMail.js ⇒ 관리자 승인 요청 이메일 전송
  - 이메일 전송 후 관리자 승인 대기

### 5. 관리자 승인

- views/admin\_approve.ejs ⇒ 관리자 승인 페이지 (승인 버튼 ON);
- routes/adminApprove.js → /admin/approve ⇒ 관리자 승인 :
  - 승인시, 관리자가 이메일로 승인 완료 알림을 보냄
  - 승인된 사용자는 로그인 가능

## 미들웨어 및 라우터 연결

- adminAuth.js : JWT 인증을 통해 보호된 라우터 profile, adminApprove 등에 접근시 인증 확인
- emailAuth.js : 이메일 인증을 통해 회원가입 시 이메일 인증 코드 검증
- registerForm.js : 회원가입 양식의 유효성을 체크
- registerMail.js : 회원가입 후 관리자에게 승인 요청 이메일을 전송

## 최종 파일 연결

1. public/login.js : 사용자가 로그인 요청을 보내면, /auth/login 경로를 통해 auth.js 로그인 라우터로 이동
2. public/register.js : 사용자가 이메일을 입력하면 /signup/email로 이메일 인증을 시작하고, /signup/email/verify로 인증 진행
3. views/admin\_register.ejs : 회원가입 폼이 제출되면 /register/admin으로 요청을 보내고, middlewares/registerMail.js에서 관리자 승인 이메일 전송
4. routes/adminApprove.js : 관리자가 승인 버튼을 클릭하면 사용자의 승인을 진행하고, 승인된 이메일로 알림을 전송
5. views/admin\_approve.ejs : 관리자 승인