Full Notes: OAuth 2 , JWT & Django Integration

1️⃣ OAuth 2 কী?

OAuth 2 হলো একটি authorization framework, যা দিয়ে কোনো অ্যাপ ইউজারের password না জেনে অন্য
ডাটা ব্যবহার করতে পারে।

Key Terms: - Resource Owner: ইউজার, যিনি ডাটা/permission দেন - Client: তোমার অ্যাপ/সার্ভিস যা acc
Authorization Server: যেখান থেকে token আসে (Google, Facebook) - Resource Server: যেখান থেকে pro
data আসে - Access Token: ডাটা access করার জন্য credential - Refresh Token: expired token পুনরায়
জন্য

2️⃣ OAuth 2 Flow Diagram (Authorization Code Flow):

User -> Django App -> Google OAuth URL -> Google Authorization Server -> User login + pe
Authorization Code -> Django App -> Token Endpoint -> Access Token -> User Info API

3️⃣ OAuth 2 এর প্রয়োজন - পাসওয়ার্ড না শেয়ার করেই login - Secure access token দিয়ে limited pe
Single Sign-On (SSO) - API & Mobile App friendly - Scope & Expiry control

---

4️⃣ Django-তে Google OAuth 2 Integration (django-allauth ব্যবহার করে)

Step 1 : Install pip install django-allauth

Step 2 : settings.py INSTALLED_APPS = [..., 'allauth', 'allauth.account', 'allauth.socialaccount',
'allauth.socialaccount.providers.google'] SITE_ID = 1 AUTHENTICATION_BACKENDS =
['django.contrib.auth.backends.ModelBackend', 'allauth.account.auth_backends.AuthenticationBackend']

Step 3 : urls.py path('accounts/', include('allauth.urls'))

Step 4 : Google Console Redirect URI: http:// 1 2.7 . 0 . 0 : 1 8 0 0/accounts/google/login/
callback/

Step 5 : Django Admin Social Applications -> Google -> Client ID & Secret -> Add site

Step 6 : Template login.html Login with Google

5️⃣ JWT (JSON Web Token)

Structure: Header.Payload.Signature - Header: algorithm, token type - Payload: user info, claims - Si
token verify করার জন্য

Use Case: - Mobile API authentication - SPA authentication

6️⃣ OAuth 2 vs JWT

বিষয়	OAuth 2	JWT
Purpose	Authorization framework	Token format (Identity / Claims)
Use	Third-party access	API authentication
Password	Not required	Not required
Server	Needs token store	Stateless (optional)
Expiry	Access + Refresh Token	Usually short expiry
Example	Google/Facebook login	Mobile API auth

7️⃣ OAuth 2 + JWT Together

Scenario: Mobile App calls Django API - User logs in via OAuth 2 - Django creates JWT toke
access - Mobile app uses JWT in Authorization header - Django validates JWT → grants API access

---

8️⃣ Best Practices - Always use HTTPS - Store tokens securely - Short-lived Access Token + Re
Use scope to limit access - Revoke tokens on logout - Combine OAuth 2 for login + JWT for API

End of Notes