"""

token\_manager.py

Manages JWT and other tokens.

"""

def generate\_token(data):

"""

Generates a JWT with the given payload data.

Steps:

1. Define the token's payload (e.g., user ID, expiry time).

2. Use a library like `pyjwt` to encode the token with a secret key.

3. Return the token.

"""

def decode\_token(token):

"""

Decodes and validates a JWT token.

Steps:

1. Decode the token using the secret key.

2. Verify its signature and expiry time.

3. Return the decoded payload or raise an error if invalid.

"""

def refresh\_token(token):

"""

Refreshes an expired token by generating a new one with the same payload.

Steps:

1. Decode the expired token to extract the payload.

2. Generate a new token with an updated expiry time.

3. Return the refreshed token.

"""