# src/user\_interactions.py

def complete\_survey(user, survey\_id, points\_per\_survey=30):

    """

    Award points for completing a survey.

    :param user: The user completing the survey.

    :param survey\_id: The identifier of the survey completed.

    :param points\_per\_survey: The number of points awarded per survey.

    :return: A message indicating the points awarded.

    """

    # Simulate completing a survey

    # In a real application, this would involve recording the survey completion in a database

    points\_awarded = points\_per\_survey

    return f"User {user} completed survey {survey\_id} and earned {points\_awarded} points."

def update\_profile(user, profile\_fields\_updated, points\_per\_field=5):

    """

    Award points for updating profile information.

    :param user: The user updating their profile.

    :param profile\_fields\_updated: The number of profile fields updated.

    :param points\_per\_field: The number of points awarded per field updated.

    :return: A message indicating the points awarded.

    """

    # Simulate updating profile information

    # In a real application, this would involve updating the user's profile in a database

    points\_awarded = profile\_fields\_updated \* points\_per\_field

    return f"User {user} updated {profile\_fields\_updated} profile fields and earned {points\_awarded} points."

def check\_daily\_offers(user, points\_per\_check=10):

    """

    Award points for checking daily offers.

    :param user: The user checking the daily offers.

    :param points\_per\_check: The number of points awarded per check.

    :return: A message indicating the points awarded.

    """

    # Simulate checking daily offers

    # In a real application, this would involve logging the user's interaction with the offers

    points\_awarded = points\_per\_check

    return f"User {user} checked daily offers and earned {points\_awarded} points."