

```
import uuid
```

```
from swarms.telemetry.user_utils import (  
    generate_unique_identifier,  
    generate_user_id,  
    get_machine_id,  
    get_system_info,  
)
```

```
# Helper functions tests
```

```
def test_generate_user_id():  
    # Generate user IDs and ensure they are UUID strings  
    user_id = generate_user_id()  
    assert isinstance(user_id, str)  
    assert uuid.UUID(user_id, version=4)
```

```
def test_get_machine_id():  
    # Get machine ID and ensure it's a valid SHA-256 hash  
    machine_id = get_machine_id()  
    assert isinstance(machine_id, str)  
    assert len(machine_id) == 64  
    assert all(char in "0123456789abcdef" for char in machine_id)
```

```
def test_get_system_info():

    # Get system information and ensure it's a dictionary with expected keys

    system_info = get_system_info()

    assert isinstance(system_info, dict)

    expected_keys = [

        "platform",

        "platform_release",

        "platform_version",

        "architecture",

        "hostname",

        "ip_address",

        "mac_address",

        "processor",

        "python_version",

    ]

    assert all(key in system_info for key in expected_keys)
```

```
def test_generate_unique_identifier():

    # Generate unique identifiers and ensure they are valid UUID strings

    unique_id = generate_unique_identifier()

    assert isinstance(unique_id, str)

    assert uuid.UUID(

        unique_id, version=5, namespace=uuid.NAMESPACE_DNS

    )
```

```
def test_generate_user_id_edge_case():  
    # Test generate_user_id with multiple calls  
    user_ids = set()  
    for _ in range(100):  
        user_id = generate_user_id()  
        user_ids.add(user_id)  
    assert len(user_ids) == 100 # Ensure generated IDs are unique
```

```
def test_get_machine_id_edge_case():  
    # Test get_machine_id with multiple calls  
    machine_ids = set()  
    for _ in range(100):  
        machine_id = get_machine_id()  
        machine_ids.add(machine_id)  
    assert len(machine_ids) == 100 # Ensure generated IDs are unique
```

```
def test_get_system_info_edge_case():  
    # Test get_system_info for consistency  
    system_info1 = get_system_info()  
    system_info2 = get_system_info()  
    assert (  
        system_info1 == system_info2  
    ) # Ensure system info remains the same
```

```
def test_generate_unique_identifier_edge_case():  
    # Test generate_unique_identifier for uniqueness  
    unique_ids = set()  
    for _ in range(100):  
        unique_id = generate_unique_identifier()  
        unique_ids.add(unique_id)  
    assert len(unique_ids) == 100 # Ensure generated IDs are unique
```