Execute Task: 3. Build authentication and user management system 3. Build authentication and user management system

**Kiro**

Including Steering Documents

GenRules.md

I'll implement task 3: "Build authentication and user management system" from the GITTE federated learning system spec. Let me start by updating the task status and then examining the current codebase to understand what's already in place.

Task: *3. Build authentication and user management system* from tasks.md

Status: In Progress

Now let me examine the current codebase structure to understand what's already implemented:

Read file(s)

models.py

schemas.py

config.py

Now let me check what's already in the logic and services layers:

Searched workspace

Examining the source code structure to understand existing layers

Let me check the existing repositories and database setup:

Read file(s)

repositories.py

database.py

Now let me check what tests already exist to understand the expected patterns:

Read file(s)

test\_data\_models.py

Now I need to implement the authentication and user management system. Based on the task requirements and existing code structure, I need to create:

1. Authentication logic in the logic layer
2. Authentication service in the services layer
3. Unit tests for the authentication logic

Let me start by creating the authentication logic:

Created

authentication.py

Now I need to create the session manager service:

Created

session\_manager.py

Now I need to create unit tests for the authentication logic. Let me create a comprehensive test file:

Created

test\_authentication.py

Now I need to add the bcrypt dependency to the requirements.txt file:

Read file(s)

requirements.txt

Great! The bcrypt dependency is already in the requirements.txt file. Now let me run the tests to make sure everything works:

Command

python -m pytest tests/test\_authentication.py -v

PS E:\Forschung\KIRO\_GITTE> python -m pytest tests/test\_authentication.py -v

========================================================= test session starts ==========================================================

platform win32 -- Python 3.10.11, pytest-8.4.1, pluggy-1.6.0 -- E:\Program Files\Python310\python.exe

cachedir: .pytest\_cache

rootdir: E:\Forschung\KIRO\_GITTE

plugins: anyio-4.6.2.post1, Faker-37.5.3, langsmith-0.3.45, asyncio-1.1.0, cov-6.2.1, mock-3.14.1

asyncio: mode=strict, asyncio\_default\_fixture\_loop\_scope=None, asyncio\_default\_test\_loop\_scope=function

collected 30 items

tests/test\_authentication.py::TestAuthenticationLogic::test\_register\_user\_success PASSED [ 3%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_register\_user\_duplicate\_username PASSED [ 6%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_register\_user\_unique\_pseudonyms FAILED [ 10%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_user\_success PASSED [ 13%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_user\_invalid\_username PASSED [ 16%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_user\_invalid\_password PASSED [ 20%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_inactive\_user PASSED [ 23%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_login\_user\_success PASSED [ 26%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_logout\_user\_success PASSED [ 30%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_get\_current\_user\_valid\_session PASSED [ 33%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_get\_current\_user\_invalid\_session PASSED [ 36%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_check\_user\_role\_participant PASSED [ 40%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_check\_user\_role\_admin PASSED [ 43%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_authentication\_success PASSED [ 46%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_authentication\_failure PASSED [ 50%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_role\_success PASSED [ 53%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_role\_failure PASSED [ 56%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_admin\_success PASSED [ 60%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_admin\_failure PASSED [ 63%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_password\_hashing\_and\_verification PASSED [ 66%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_pseudonym\_generation PASSED [ 70%]

tests/test\_authentication.py::TestSessionManager::test\_create\_session PASSED [ 73%]

tests/test\_authentication.py::TestSessionManager::test\_get\_session\_valid PASSED [ 76%]

tests/test\_authentication.py::TestSessionManager::test\_get\_session\_invalid PASSED [ 80%]

tests/test\_authentication.py::TestSessionManager::test\_invalidate\_session PASSED [ 83%]

tests/test\_authentication.py::TestSessionManager::test\_refresh\_session PASSED [ 86%]

tests/test\_authentication.py::TestSessionManager::test\_invalidate\_user\_sessions PASSED [ 90%]

tests/test\_authentication.py::TestSessionManager::test\_get\_active\_sessions\_count PASSED [ 93%]

tests/test\_authentication.py::TestSessionManager::test\_get\_user\_sessions PASSED [ 96%]

tests/test\_authentication.py::TestSessionManager::test\_cleanup\_all\_sessions PASSED [100%]

=============================================================== FAILURES ===============================================================

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ TestAuthenticationLogic.test\_register\_user\_unique\_pseudonyms \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

self = <tests.test\_authentication.TestAuthenticationLogic object at 0x00000195D1F527D0>

auth\_logic = <src.logic.authentication.AuthenticationLogic object at 0x00000195D2122860>

def test\_register\_user\_unique\_pseudonyms(self, auth\_logic):

"""Test that multiple users get unique pseudonyms."""

> user1\_data = UserCreate(username="user1", password="pass123", role=UserRole.PARTICIPANT)

E pydantic\_core.\_pydantic\_core.ValidationError: 1 validation error for UserCreate

E password

E String should have at least 8 characters [type=string\_too\_short, input\_value='pass123', input\_type=str]

E For further information visit https://errors.pydantic.dev/2.11/v/string\_too\_short

tests\test\_authentication.py:118: ValidationError

======================================================= short test summary info ========================================================

FAILED tests/test\_authentication.py::TestAuthenticationLogic::test\_register\_user\_unique\_pseudonyms - pydantic\_core.\_pydantic\_core.Validat

ionError: 1 validation error for UserCreate ===================================================== 1 failed, 29 passed in 7.38s =====================================================

I need to fix the test that has a password that's too short. Let me update the test:

Accepted edits to

test\_authentication.py

Now let me run the tests again:

Command

python -m pytest tests/test\_authentication.py -v

PS E:\Forschung\KIRO\_GITTE> python -m pytest tests/test\_authentication.py -v

========================================================= test session starts ==========================================================

platform win32 -- Python 3.10.11, pytest-8.4.1, pluggy-1.6.0 -- E:\Program Files\Python310\python.exe

cachedir: .pytest\_cache

rootdir: E:\Forschung\KIRO\_GITTE

plugins: anyio-4.6.2.post1, Faker-37.5.3, langsmith-0.3.45, asyncio-1.1.0, cov-6.2.1, mock-3.14.1

asyncio: mode=strict, asyncio\_default\_fixture\_loop\_scope=None, asyncio\_default\_test\_loop\_scope=function

collected 30 items

tests/test\_authentication.py::TestAuthenticationLogic::test\_register\_user\_success PASSED [ 3%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_register\_user\_duplicate\_username PASSED [ 6%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_register\_user\_unique\_pseudonyms PASSED [ 10%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_user\_success PASSED [ 13%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_user\_invalid\_username PASSED [ 16%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_user\_invalid\_password PASSED [ 20%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_inactive\_user PASSED [ 23%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_login\_user\_success PASSED [ 26%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_logout\_user\_success PASSED [ 30%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_get\_current\_user\_valid\_session PASSED [ 33%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_get\_current\_user\_invalid\_session PASSED [ 36%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_check\_user\_role\_participant PASSED [ 40%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_check\_user\_role\_admin PASSED [ 43%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_authentication\_success PASSED [ 46%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_authentication\_failure PASSED [ 50%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_role\_success PASSED [ 53%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_role\_failure PASSED [ 56%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_admin\_success PASSED [ 60%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_admin\_failure PASSED [ 63%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_password\_hashing\_and\_verification PASSED [ 66%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_pseudonym\_generation PASSED [ 70%]

tests/test\_authentication.py::TestSessionManager::test\_create\_session PASSED [ 73%]

tests/test\_authentication.py::TestSessionManager::test\_get\_session\_valid PASSED [ 76%]

tests/test\_authentication.py::TestSessionManager::test\_get\_session\_invalid PASSED [ 80%]

tests/test\_authentication.py::TestSessionManager::test\_invalidate\_session PASSED [ 83%]

tests/test\_authentication.py::TestSessionManager::test\_refresh\_session PASSED [ 86%]

tests/test\_authentication.py::TestSessionManager::test\_invalidate\_user\_sessions PASSED [ 90%]

tests/test\_authentication.py::TestSessionManager::test\_get\_active\_sessions\_count PASSED [ 93%]

tests/test\_authentication.py::TestSessionManager::test\_get\_user\_sessions PASSED [ 96%]

tests/test\_authentication.py::TestSessionManager::test\_cleanup\_all\_sessions PASSED [100%]

========================================================== 30 passed in 7.23s ==========================================================

Excellent! All tests are passing. Now let me also run the existing tests to make sure I haven't broken anything:

Command

python -m pytest tests/test\_data\_models.py -v

PS E:\Forschung\KIRO\_GITTE> python -m pytest tests/test\_data\_models.py -v

========================================================= test session starts ==========================================================

platform win32 -- Python 3.10.11, pytest-8.4.1, pluggy-1.6.0 -- E:\Program Files\Python310\python.exe

cachedir: .pytest\_cache

rootdir: E:\Forschung\KIRO\_GITTE

plugins: anyio-4.6.2.post1, Faker-37.5.3, langsmith-0.3.45, asyncio-1.1.0, cov-6.2.1, mock-3.14.1

asyncio: mode=strict, asyncio\_default\_fixture\_loop\_scope=None, asyncio\_default\_test\_loop\_scope=function

collected 7 items

tests/test\_data\_models.py::test\_user\_model\_creation PASSED [ 14%]

tests/test\_data\_models.py::test\_user\_repository PASSED [ 28%]

tests/test\_data\_models.py::test\_consent\_model\_and\_repository PASSED [ 42%]

tests/test\_data\_models.py::test\_pald\_schema\_version PASSED [ 57%]

tests/test\_data\_models.py::test\_pald\_data\_model\_and\_repository PASSED [ 71%]

tests/test\_data\_models.py::test\_database\_relationships PASSED [ 85%]

tests/test\_data\_models.py::test\_model\_validation PASSED [100%]

========================================================== 7 passed in 0.62s ===========================================================

Perfect! Now let me also run the config tests:

Command

python -m pytest tests/test\_config.py -v

PS E:\Forschung\KIRO\_GITTE> python -m pytest tests/test\_config.py -v

========================================================= test session starts ==========================================================

platform win32 -- Python 3.10.11, pytest-8.4.1, pluggy-1.6.0 -- E:\Program Files\Python310\python.exe

cachedir: .pytest\_cache

rootdir: E:\Forschung\KIRO\_GITTE

plugins: anyio-4.6.2.post1, Faker-37.5.3, langsmith-0.3.45, asyncio-1.1.0, cov-6.2.1, mock-3.14.1

asyncio: mode=strict, asyncio\_default\_fixture\_loop\_scope=None, asyncio\_default\_test\_loop\_scope=function

collected 9 items

tests/test\_config.py::test\_config\_initialization PASSED [ 11%]

tests/test\_config.py::test\_feature\_flags\_initialization PASSED [ 22%]

tests/test\_config.py::test\_environment\_variable\_override PASSED [ 33%]

tests/test\_config.py::test\_feature\_flag\_environment\_override PASSED [ 44%]

tests/test\_config.py::test\_get\_feature\_flag PASSED [ 55%]

tests/test\_config.py::test\_text\_localization PASSED [ 66%]

tests/test\_config.py::test\_config\_validation\_development PASSED [ 77%]

tests/test\_config.py::test\_config\_validation\_production\_fails PASSED [ 88%]

tests/test\_config.py::test\_config\_validation\_production\_passes PASSED [100%]

========================================================== 9 passed in 0.17s ===========================================================

Great! All existing tests are still passing. Now I need to create the **init**.py files to make the modules importable:

Read file(s)

\_\_init\_\_.py

Read file(s)

\_\_init\_\_.py

Now let me run all tests to make sure everything is working together:

Command

python -m pytest tests/ -v

PS E:\Forschung\KIRO\_GITTE> python -m pytest tests/ -v

========================================================= test session starts ==========================================================

platform win32 -- Python 3.10.11, pytest-8.4.1, pluggy-1.6.0 -- E:\Program Files\Python310\python.exe

cachedir: .pytest\_cache

rootdir: E:\Forschung\KIRO\_GITTE

plugins: anyio-4.6.2.post1, Faker-37.5.3, langsmith-0.3.45, asyncio-1.1.0, cov-6.2.1, mock-3.14.1

asyncio: mode=strict, asyncio\_default\_fixture\_loop\_scope=None, asyncio\_default\_test\_loop\_scope=function

collected 46 items

tests/test\_authentication.py::TestAuthenticationLogic::test\_register\_user\_success PASSED [ 2%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_register\_user\_duplicate\_username PASSED [ 4%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_register\_user\_unique\_pseudonyms PASSED [ 6%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_user\_success PASSED [ 8%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_user\_invalid\_username PASSED [ 10%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_user\_invalid\_password PASSED [ 13%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_authenticate\_inactive\_user PASSED [ 15%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_login\_user\_success PASSED [ 17%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_logout\_user\_success PASSED [ 19%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_get\_current\_user\_valid\_session PASSED [ 21%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_get\_current\_user\_invalid\_session PASSED [ 23%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_check\_user\_role\_participant PASSED [ 26%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_check\_user\_role\_admin PASSED [ 28%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_authentication\_success PASSED [ 30%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_authentication\_failure PASSED [ 32%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_role\_success PASSED [ 34%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_role\_failure PASSED [ 36%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_admin\_success PASSED [ 39%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_require\_admin\_failure PASSED [ 41%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_password\_hashing\_and\_verification PASSED [ 43%]

tests/test\_authentication.py::TestAuthenticationLogic::test\_pseudonym\_generation PASSED [ 45%]

tests/test\_authentication.py::TestSessionManager::test\_create\_session PASSED [ 47%]

tests/test\_authentication.py::TestSessionManager::test\_get\_session\_valid PASSED [ 50%]

tests/test\_authentication.py::TestSessionManager::test\_get\_session\_invalid PASSED [ 52%]

tests/test\_authentication.py::TestSessionManager::test\_invalidate\_session PASSED [ 54%]

tests/test\_authentication.py::TestSessionManager::test\_refresh\_session PASSED [ 56%]

tests/test\_authentication.py::TestSessionManager::test\_invalidate\_user\_sessions PASSED [ 58%]

tests/test\_authentication.py::TestSessionManager::test\_get\_active\_sessions\_count PASSED [ 60%]

tests/test\_authentication.py::TestSessionManager::test\_get\_user\_sessions PASSED [ 63%]

tests/test\_authentication.py::TestSessionManager::test\_cleanup\_all\_sessions PASSED [ 65%]

tests/test\_config.py::test\_config\_initialization PASSED [ 67%]

tests/test\_config.py::test\_feature\_flags\_initialization PASSED [ 69%]

tests/test\_config.py::test\_environment\_variable\_override PASSED [ 71%]

tests/test\_config.py::test\_feature\_flag\_environment\_override PASSED [ 73%]

tests/test\_config.py::test\_get\_feature\_flag PASSED [ 76%]

tests/test\_config.py::test\_text\_localization PASSED [ 78%]

tests/test\_config.py::test\_config\_validation\_development PASSED [ 80%]

tests/test\_config.py::test\_config\_validation\_production\_fails PASSED [ 82%]

tests/test\_config.py::test\_config\_validation\_production\_passes PASSED [ 84%]

tests/test\_data\_models.py::test\_user\_model\_creation PASSED [ 86%]

tests/test\_data\_models.py::test\_user\_repository PASSED [ 89%]

tests/test\_data\_models.py::test\_consent\_model\_and\_repository PASSED [ 91%]

tests/test\_data\_models.py::test\_pald\_schema\_version PASSED [ 93%]

tests/test\_data\_models.py::test\_pald\_data\_model\_and\_repository PASSED [ 95%]

tests/test\_data\_models.py::test\_database\_relationships PASSED [ 97%]

tests/test\_data\_models.py::test\_model\_validation PASSED [100%]

========================================================== 46 passed in 7.34s ==========================================================

Perfect! All 46 tests are passing. Now let me verify that the task requirements have been met by checking against the task details:

**Task Requirements Verification:**

✅ **Implement user registration with bcrypt password hashing**

* Created AuthenticationLogic.register\_user() method
* Uses bcrypt with configurable rounds from config
* Includes password strength validation in Pydantic schema
* Tests verify password hashing works correctly

✅ **Create login/logout functionality with session management**

* Created AuthenticationLogic.login\_user() and logout\_user() methods
* Implemented SessionManager service for session handling
* Sessions include expiration, refresh, and cleanup functionality
* Tests verify login/logout flow works correctly

✅ **Implement role-based access control (Admin vs Participant)**

* Created role checking methods: check\_user\_role(), require\_role(), require\_admin()
* Admin users have access to all functionality
* Participant users are restricted to their role
* Tests verify role-based access control works correctly

✅ **Add user pseudonymization for privacy compliance**

* Created \_generate\_pseudonym() method that generates unique pseudonyms
* Format: GITTE\_XXXXXXXX with 8 random characters
* Ensures uniqueness by checking existing pseudonyms
* Tests verify pseudonym generation and uniqueness

✅ **Create unit tests for authentication logic**

* Created comprehensive test suite with 30 tests
* Tests cover all authentication functionality
* Tests include edge cases and error conditions
* All tests pass successfully

✅ **Requirements Coverage**

* **2.1**: User registration and login options ✅
* **2.2**: Password hashing with bcrypt ✅
* **2.3**: Authentication against stored credentials ✅
* **2.4**: Role assignment (Admin/Participant) ✅
* **2.5**: Appropriate error messages without revealing sensitive info ✅
* **12.1**: Role-based access control ✅
* **14.1**: Pseudonymization for privacy compliance ✅

Now let me mark the task as completed:

Task: *3. Build authentication and user management system* from tasks.md

Status: Completed

**Summary**

I have successfully implemented task 3: "Build authentication and user management system" for the GITTE federated learning system. Here's what was accomplished:

**🔐 Authentication Logic Layer (src/logic/authentication.py)**

* **User Registration**: Secure registration with bcrypt password hashing and unique pseudonym generation
* **User Authentication**: Login/logout with session management and credential verification
* **Role-Based Access Control**: Admin vs Participant roles with proper authorization checks
* **Privacy Compliance**: Automatic pseudonymization using GITTE\_XXXXXXXX format
* **Error Handling**: Comprehensive exception hierarchy with secure error messages

**🔧 Session Management Service (src/services/session\_manager.py)**

* **Session Creation**: Secure session ID generation with configurable expiration
* **Session Validation**: Automatic cleanup of expired sessions
* **Multi-Session Support**: Users can have multiple active sessions
* **Session Invalidation**: Individual and bulk session termination
* **Memory Management**: Efficient in-memory storage with cleanup routines

**🧪 Comprehensive Test Suite (tests/test\_authentication.py)**

* **30 Unit Tests**: Complete coverage of authentication functionality
* **Edge Case Testing**: Invalid credentials, inactive users, expired sessions
* **Role Testing**: Verification of admin and participant access controls
* **Security Testing**: Password hashing, pseudonym uniqueness, session security
* **Integration Testing**: Full authentication flow from registration to logout

**🏗️ Architecture Compliance**

* **4-Layer Architecture**: Strict separation between UI → Logic → Service → Data layers
* **Configuration-Driven**: Uses centralized config for security settings
* **Privacy by Design**: Pseudonymization and secure session management
* **Error Handling**: Graceful degradation with appropriate user feedback

**✅ Requirements Fulfilled**

All specified requirements (2.1, 2.2, 2.3, 2.4, 2.5, 12.1, 14.1) have been implemented and tested successfully. The system now provides a robust authentication foundation that supports:

* Secure user registration and login
* Role-based access control for admin and participant users
* Privacy-compliant pseudonymization
* Session management with automatic cleanup
* Comprehensive error handling and logging

The implementation follows the established patterns in the codebase and maintains compatibility with existing data models and repositories. All 46 tests in the project are passing, confirming that the new authentication system integrates seamlessly with the existing infrastructure.