

Test Suite Documentation

Overview

This project includes a comprehensive PHPUnit test suite that validates all requirements from the problem statement.

Test Cases Implemented

1. Successful Login with Valid Credentials

Test: `testSuccessfulLoginWithValidCredentials()`

- Creates a user with valid credentials
- Attempts to login with correct username and password
- Verifies login succeeds and returns correct user data

2. Unsuccessful Login with Incorrect Password

Test: `testUnsuccessfulLoginWithIncorrectPassword()`

- Creates a user with valid credentials
- Attempts to login with correct username but wrong password
- Verifies login fails

3. Unsuccessful Login with Nonexistent Username

Test: `testUnsuccessfulLoginWithNonexistentUsername()`

- Attempts to login with a username that doesn't exist
- Verifies no user is returned

4. Successful Login After Creating New User

Test: `testSuccessfulLoginAfterCreatingNewUser()`

- Creates a new user

- Immediately attempts to login with the new credentials
- Verifies both registration and login succeed

5. Unsuccessful Login with Empty Password

Test: `testUnsuccessfulLoginWithEmptyPassword()`

- Attempts to register with empty password (fails validation)
- Creates valid user and tries to login with empty password
- Verifies both scenarios fail appropriately

Additional Tests

6. Username Uniqueness

Test: `testUsernameUniqueness()`

- Verifies duplicate usernames are rejected

7. Secure Password Storage

Test: `testSecurePasswordStorage()`

- Verifies passwords are hashed with Argon2id
- Confirms passwords are not stored in plain text
- Validates password verification works correctly

Setup Instructions

1. Create Test Database

Run the SQL script to create the test database:

```
mysql -u root -p < tests/setup_test_db.sql
```

Or manually in MySQL:

```
source tests/setup_test_db.sql;
```

2. Install Dependencies

Install PHPUnit and other dependencies via Composer:

```
composer install
```

3. Configure Database

Update `phpunit.xml` with your MySQL credentials if different from defaults:

- DB_HOST: localhost
- DB_NAME: your_test_db_name
- DB_USER: root //change to your user
- DB_PASS: your_MYSQL_password

Running Tests

Run Automated Tests

To run all automated tests quickly, use the test runner script:

```
php run_tests.php
```

Run All Tests

```
vendor/bin/phpunit
```

Run with Verbose Output

```
vendor/bin/phpunit --verbose
```

Run with Colors

```
vendor/bin/phpunit --colors=always
```

Run Specific Test

```
vendor/bin/phpunit --filter testSuccessfulLoginWithValidCredentials
```

Run with Coverage (requires Xdebug)

```
vendor/bin/phpunit --coverage-html coverage
```

Test Database

- **Database Name:** `login_system_test`
- **Isolation:** Each test cleans up data before and after execution
- **Structure:** Identical to production database
- **Purpose:** Separate from production to prevent data corruption

Expected Results

All 7 tests should pass:

```
PHPUnit 10.x
.....
(100%)

Time: 00:00.123, Memory: 10.00 MB

OK (7 tests, 25 assertions)
```

7 / 7

Project Requirements Compliance

Requirement	Status	Implementation
Create User Function	✓	AuthService::register()
Login Function	✓	findUserByUsernameOrEmail() + password_verify()
Unique Usernames	✓	Database constraint + validation
Secure Password Storage	✓	Argon2id hashing

Requirement	Status	Implementation
Empty Password Handling	✓	Validation in Validator class
Test Case 1: Valid Login	✓	testSuccessfulLoginWithValidCredentials()
Test Case 2: Wrong Password	✓	testUnsuccessfulLoginWithIncorrectPassword()
Test Case 3: Nonexistent User	✓	testUnsuccessfulLoginWithNonexistentUsername()
Test Case 4: New User Login	✓	testSuccessfulLoginAfterCreatingNewUser()
Test Case 5: Empty Password	✓	testUnsuccessfulLoginWithEmptyPassword()

Troubleshooting

Issue: Database Connection Failed

- Verify MySQL is running on XAMPP
- Check credentials in `phpunit.xml`
- Ensure test database exists: `mysql -u root -p -e "SHOW DATABASES;"`

Issue: Composer Not Found

- Install Composer: <https://getcomposer.org/download/>
- Add to PATH or use full path: `php composer.phar install`

Issue: Tests Fail

- Check PHP version (requires PHP 8.0+)
- Verify PDO extension is enabled
- Ensure test database schema matches production

Notes

- Tests automatically clean up data between runs
- Each test is isolated and independent
- Tests can be run in any order
- Session management is handled by bootstrap