Authentication System Improvements

Overview

This document outlines the security and usability improvements made to the authentication system in the PSScript application. These enhancements focus on improving error handling, logging, security, and user experience.

Key Improvements

1. Enhanced User Model

- Added tracking for login attempts to detect and prevent brute force attacks
- Implemented secure password validation with proper error handling
- Added last login timestamp tracking for security auditing
- Improved error handling in password validation

2. Robust Authentication Routes

- Implemented structured error responses with consistent format
- Added detailed error codes for better client-side error handling
- Enhanced validation for registration and login requests
- Improved token generation with proper expiration handling
- Added request tracking with unique request IDs for better debugging
- Implemented progressive delays for failed login attempts to prevent brute force attacks

3. Comprehensive Logging

- Added detailed logging for all authentication operations
- Included request IDs in logs for request tracing
- Logged IP addresses and user agents for security monitoring
- Added performance metrics (processing time) for monitoring
- Implemented different log levels (debug, info, warn, error) for better filtering

4. Improved Frontend Error Handling

- Enhanced Login component with better error display
- Added support for structured error responses from the backend
- Implemented user-friendly error messages based on error codes
- Added field-level validation with visual feedback
- Improved form validation with real-time feedback

5. Security Enhancements

- Properly typed JWT token handling
- Added proper environment variable handling with secure defaults
- Implemented token expiration handling
- Added IP address and user agent tracking
- Enhanced refresh token security

Technical Details

Error Response Format

All authentication endpoints now return consistent error responses:

```
{
  "success": false,
  "message": "Human-readable error message",
  "error": "error_code",
  "requestId": "unique-request-id",
  "details": {
    // Optional additional error details
  }
}
```

Error Codes

The system now uses standardized error codes:

- `validation_error`: Input validation failed
- `email_already_exists`: Email is already registered
- `username_already_exists`: Username is already taken
- `user_not_found`: User not found
- `invalid_credentials`: Password is incorrect
- `server_error`: Internal server error
- `missing_token`: No authentication token provided
- `invalid_token_format`: Token format is invalid
- `token_expired`: JWT token has expired
- `refresh_token_expired`: Refresh token has expired
- `invalid_refresh_token`: Refresh token is invalid
- `missing_refresh_token`: No refresh token provided

Request Tracking

Each authentication request now includes:

- Unique request ID
- Timestamp
- IP address
- User agent
- Processing time

This information is included in logs and error responses for better debugging and security monitoring.

Testing Authentication

Using the Admin Account

The system includes a default administrator account with the following credentials:

- Email: `admin@psscript.com`
- Password: `ChangeMe1!`

To verify the authentication system is working correctly, run:

```
node test-login.js
```

This script tests: 1. Login with admin credentials 2. Retrieving user details using the authentication token 3. Verifies token generation and validation

Future Improvements

Potential future enhancements to consider:

1. Implement rate limiting for authentication endpoints
2. Add multi-factor authentication support
3. Implement account lockout after multiple failed attempts
4. Add password strength requirements
5. Implement session management with device tracking
6. Add support for OAuth providers (Google, GitHub, etc.)

Generated 2026-01-16 23:34 UTC