Here's a general approach to conduct white box testing on the authentication module:

1. **Code Review:**
    - Examine the source code of the authentication module.
    - Look for potential security vulnerabilities such as SQL injection, cross-site scripting (XSS), insecure password storage, inadequate session management, etc.
2. **Input Validation:**
    - Ensure that all input received from users (e.g., username, password) is properly validated and sanitized to prevent injection attacks.
    - Test the authentication module with different types of input (valid, invalid, and malicious) to ensure it behaves as expected and handles errors gracefully.
3. **Authentication Process:**
    - Verify that passwords are securely hashed using a strong cryptographic hashing algorithm (e.g., crypts) and that salts are used to prevent rainbow table attacks.
    - Check for proper implementation of password complexity requirements (e.g., minimum length, character types).
    - Test for account lockout mechanisms to prevent brute force attacks.
    - Verify that sessions are managed securely, with appropriate expiration times and protection against session fixation attacks.
4. **Authorization Checks:**
    - Ensure that authorization checks are performed properly to prevent unauthorized access to sensitive resources.
    - Test various scenarios to verify that users can only access the functionalities and data they are authorized to access.
5. **Error Handling:**
    - Test error handling mechanisms to ensure that sensitive information is not leaked in error messages.
    - Verify that appropriate error messages are displayed to users in case of authentication failures or other errors.
6. **Session Management:**
    - Check for session fixation vulnerabilities by testing session management functionality.
    - Verify that sessions are invalidated properly upon logout or when they expire.
7. **Cross-Site Request Forgery (CSRF) Protection:**
    - Ensure that the authentication process is protected against CSRF attacks by implementing and testing CSRF tokens.

8. **Logging and Monitoring:**
    - Verify that relevant authentication events are logged for monitoring and auditing purposes.
    - Test logging mechanisms to ensure they capture relevant information without exposing sensitive data.
9. **Integration Testing:**
    - Test the integration of the authentication module with other components of the Shop Ease platform to ensure seamless functionality.
10. **Security Headers:**
    - Check if appropriate security headers (e.g., Content Security Policy, X-Content-Type-Options, X-Frame-Options) are set to enhance the security of the authentication process.
11. **Security Standards Compliance:**
    - Ensure that the authentication module complies with relevant security standards (e.g., OWASP Top 10, PCI DSS) and best practices.