

# Secure Auth RS - Non-Functional Test Script

## Three-Phase Authentication System Testing Guide

Version: 1.0

Date: November 13, 2025

Tester: \_\_\_\_\_

## 1 Test Environment Setup

### 1.1 Prerequisites

- Web browser (Chrome, Firefox, Safari, or Edge)
- Internet connection
- TOTP authenticator app (Aegis, Authy, etc.) MUST SUPPORT SHA512
- Rust must be installed on System. Follow the instructions here: <https://rust-lang.org/tools/install/>
- If using unix-like OS (Linux, MacOS, etc.), you can run this:  
`--proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh`
- Git clone repo if you haven't already
- `git clone https://github.com/patrickhaahr/secure_auth_rs.git`
- `cd secure_auth_rs/`
- `cargo run`
- Test URL: <https://127.0.0.1:3443>

### 1.2 Browser Security Settings

- Ensure JavaScript is enabled
- Accept TLS certificates if using local development
- Disable password managers for accurate testing

## 2 Test Scenarios

### 2.1 1. User Registration Flow

#### 2.1.1 Step 1: Navigate to Registration

1. Open browser and navigate to the application URL
2. Click on “Create Account” button
3. Verify registration page loads correctly

#### 2.1.2 Step 2: Complete Registration

1. Click “Generate Account ID” button
2. Copy the generated Account ID
3. Click on the “Login” button

### 2.2 2. User Login Flow

#### 2.2.1 Step 1: Navigate to Login

1. Open browser and navigate to application URL

2. Click on “Continue” button

#### **2.2.2 Step 2: Enter Credentials**

1. Enter your Account Id
2. Click “Login” button

#### **2.2.3 Step 3: First time TOTP authentication Setup**

1. click on the “Setup TOTP” button
2. Verify TOTP setup page loads correctly and can see QR code
3. Scan the QR code with your authenticator app
4. Enter the 6-digit TOTP code from your authenticator app
3. Click “Verify” button

#### **2.2.4 Step 4: First time CPR number authentication setup**

1. Enter your CPR number - format: DDMMMYY-XXXX
2. Click “Submit CPR” button

#### **2.2.5 Step 5: Successful Login**

1. Verify login page loads correctly
2. Click “Admin Panel” button
3. Verify “Access Denied” message appears
4. Click “Back to Dashboard” button
5. Verify dashboard loads correctly
6. Click “Logout” button
7. Verify logout page loads correctly

#### **2.2.6 Expected Results**

- Login form validates correctly
- TOTP prompt appears after credential validation
- Successful authentication redirects to dashboard
- JWT token is issued (check browser storage)

### **2.3 3. Verified Account Login Flow**

#### **2.3.1 Step 1: Navigate to Login**

1. Open browser and navigate to application URL
2. Make sure you are in index.html page: <https://127.0.0.1:3443/>

#### **2.3.2 Step 2: Enter Account ID**

1. Enter your Account ID (from a previously verified account)
2. Click “Login” button

#### **2.3.3 Step 3: Verified User Authentication**

1. Verify login page detects account as verified
2. Verify TOTP and CPR input fields appear simultaneously
3. Enter the 6-digit TOTP code from your authenticator app
4. Enter your CPR number - format: DDMMMYY-XXXX
5. Click “Confirm” button

#### **2.3.4 Expected Results**

- Login form recognizes verified account status
- Both TOTP and CPR fields are required
- No setup prompts appear (unlike first-time login)
- Successful authentication with both factors redirects to dashboard

- JWT token is issued (check browser storage)
- Session is established correctly

## **2.4 4. Admin Account Setup**

### **2.4.1 Step 1: Grant Admin Role via SQLite**

1. Open terminal/CLI on the system running the application
2. Navigate to the application directory
3. Run the following command (replace ACCOUNTID\_CHANGE\_ME with your Account ID):

```
sqlite3 auth.db "INSERT INTO account_roles (account_id, is_admin) VALUES ('ACCOUNTID_CHANGE_ME', 1) ON CONFLICT(account_id) DO UPDATE SET is_admin = 1;"`
```

4. Verify command executes without errors

### **2.4.2 Step 2: Access Admin Panel**

1. Log out of the application (if already logged in)
2. Clear browser cache and cookies
3. Log in again with your verified account credentials (TOTP + CPR)
4. Click “Admin Panel” button
5. Verify admin dashboard loads successfully
6. Verify user list is displayed with all registered accounts

### **2.4.3 Step 3: Admin User Management**

1. Verify page shows list of all users
2. Locate a test user in the list
3. Click “Delete User” button next to the test user
4. Confirm deletion when prompted
5. Verify user is removed from the list
6. Verify deletion is permanent (user cannot log in after deletion)

### **2.4.4 Expected Results**

- Admin role is successfully assigned via SQLite command
- Admin panel is accessible after re-login and cache clear
- User list displays all accounts with delete functionality
- User deletion works correctly and persists
- Deleted user accounts cannot authenticate

## **2.5 5. Security Testing**

### **2.5.1 CSRF Protection**

1. Login to the application
2. Open new browser tab/window
3. Attempt to submit form via external site or direct POST request
4. Verify CSRF token validation prevents attack

### **2.5.2 Rate Limiting**

1. Attempt multiple failed login attempts (>5)
2. Verify rate limiting activates
3. Check error messages and time delays

### **2.5.3 TLS Security**

1. Verify HTTPS connection is established
2. Check certificate validity
3. Ensure no mixed content warnings

#### 2.5.4 Session Management

1. Login successfully
2. Close browser and reopen
3. Verify session persistence/expiration as expected
4. Test logout functionality

### 3 Test Results

#### 3.1 Pass/Fail Criteria

Test	Description	Status	Notes
Registration	Complete user registration flow	<input type="checkbox"/> <input type="checkbox"/>	
Login	Authentication with TOTP	<input type="checkbox"/> <input type="checkbox"/>	
CPR Auth	CPR validation and authorization	<input type="checkbox"/> <input type="checkbox"/>	
CSRF	CSRF protection verification	<input type="checkbox"/> <input type="checkbox"/>	
Rate Limit	Rate limiting effectiveness	<input type="checkbox"/> <input type="checkbox"/>	
TLS	Secure connection verification	<input type="checkbox"/> <input type="checkbox"/>	
Performance	Response time benchmarks	<input type="checkbox"/> <input type="checkbox"/>	