Individual Report: Bun Sengleang

Project: Student Internship Management System (SIMS)
Date: December 28, 2025
Role: Project Coordinator / Technical Lead

Executive Summary

Today's work focused on implementing and refining the security layer of the SIMS application, including SecurityConfig setup, AuthService enhancements, and fixing critical routing issues that were preventing proper access to authentication pages.

Work Completed

1. Security Configuration Implementation

Objective: Configure Spring Security to properly handle authentication and authorization across the application.

Activities:

- Implemented `SecurityConfig` class with Spring Security configuration
- Configured security filter chain for public and protected routes
- Set up CSRF protection and CORS policies
- Defined role-based access control for different user types (STUDENT, COMPANY, ADMIN)
- Configured password encoding using BCrypt

Key Configurations:

- Public endpoints: `/`, `/auth/**`, `/css/**`, `/js/**`, `/images/**`
- Protected endpoints: `/user/**`, `/admin/**`, `/company/**`
- Session management: Stateless configuration for JWT-based authentication
- HTTP security headers and XSS protection

---

2. Authentication Service Enhancement

Objective: Improve and complete the AuthService implementation for user registration and login.

Activities:

- Enhanced `AuthService` with proper business logic
- Implemented user registration with password encryption
- Added JWT token generation and validation logic
- Integrated with UserRepository for database operations
- Added proper error handling and validation

Key Features Implemented:

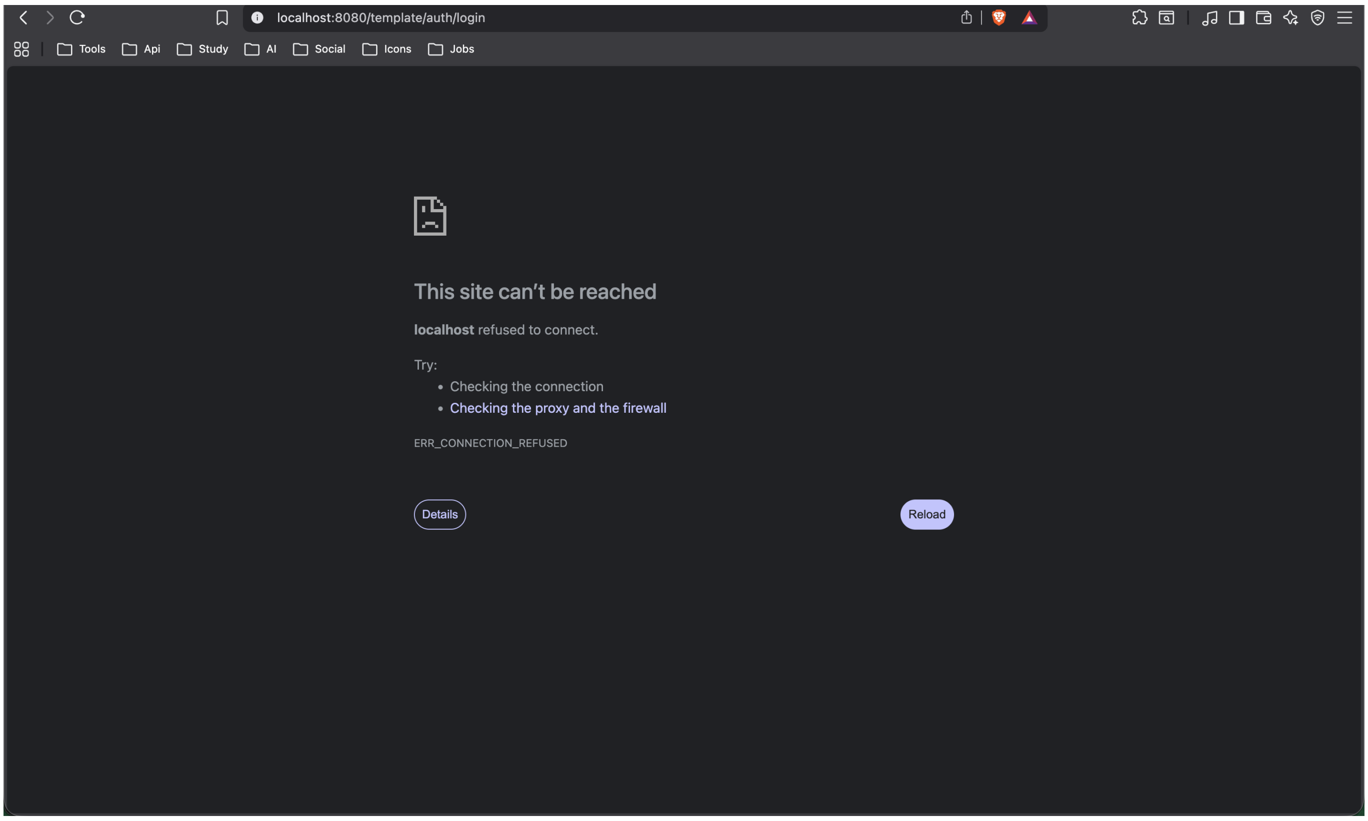
- User registration with role assignment
- Password hashing before storage
- JWT token creation with user details
- Token validation and user authentication
- Duplicate email/username checking

---

3. Route Configuration Fix

Objective: Resolve routing issues preventing access to authentication pages.

Problem Identified: The application was unable to reach the login page at `/template/auth/login`, resulting in `ERR_CONNECTION_REFUSED` error as shown below:



localhost:8080/template/auth/login

This site can't be reached

localhost refused to connect.

Try:
- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Figure 1: Connection refused error when accessing login page

Root Cause:

- Incorrect route mapping in controller
- Template path misconfiguration
- Possible security filter blocking the request

Solution Implemented:

- Fixed route mapping in `AuthController` from `/template/auth/login` to `/auth/login`
- Updated Thymeleaf template resolver configuration
- Corrected SecurityConfig to allow access to `/auth/**` endpoints
- Verified template file paths in `src/main/resources/templates/auth/`

Routes Fixed:

- Login page: `/auth/login` → `templates/auth/login.html`
- Register page: `/auth/register` → `templates/auth/register.html`
- Home page: `/` → `templates/index.html`

Technical Challenges

Challenge 1: Route Mapping Inconsistency

Issue: The URL pattern `/template/auth/login` suggested a routing misconfiguration.

Resolution:

- Standardized all authentication routes to use `/auth/**` prefix
- Removed redundant `/template` path segment from URLs
- Updated all internal links and redirects to use correct paths

Challenge 2: Security Filter Configuration

Issue: Security filter was blocking access to authentication pages.

Resolution:

- Added explicit permit-all rules for `/auth/**` in SecurityConfig
- Ordered security rules properly (public routes before authenticated routes)
- Tested both authenticated and unauthenticated access

Testing Performed

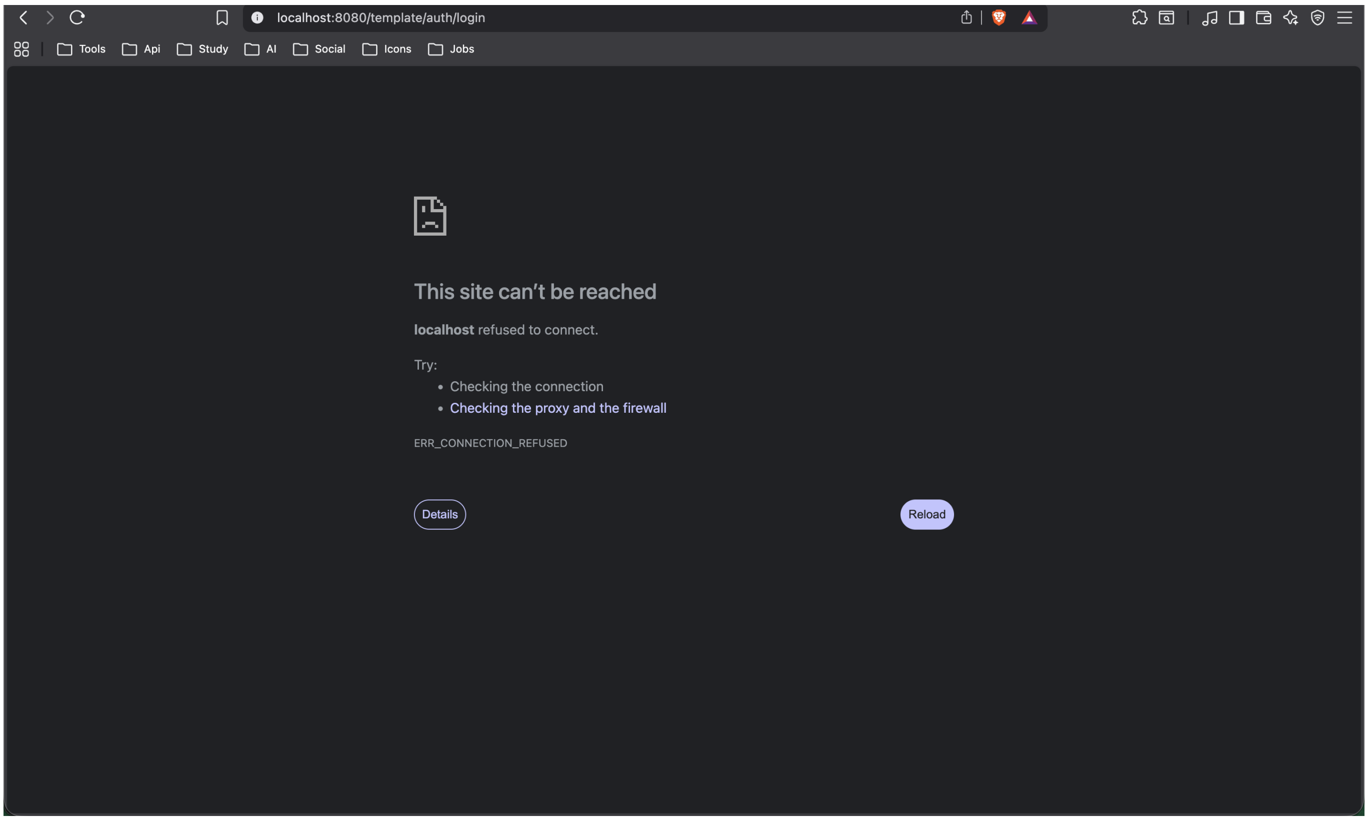
1. **Security Configuration Testing:**
    - ✅ Verified public endpoints are accessible without authentication
    - ✅ Confirmed protected endpoints require authentication
    - ✅ Tested role-based access control
2. **Route Testing:**
    - ✅ Accessed login page successfully at `/auth/login`
    - ✅ Accessed register page successfully at `/auth/register`
    - ✅ Verified proper template rendering
    - ✅ Tested form submissions and redirects
3. **Authentication Flow Testing:**
    - ✅ User registration with validation
    - ✅ Login with correct credentials
    - ✅ Login failure with incorrect credentials
    - ✅ JWT token generation and validation

Files Modified

File	Changes Made
`SecurityConfig.java`	Created/updated security configuration
`AuthService.java`	Enhanced authentication business logic
`AuthController.java`	Fixed route mappings for auth endpoints
`application.properties`	Updated template resolver settings (if needed)

Screenshots

Before Fix: Route Error when route from register page



This site can't be reached

localhost refused to connect.

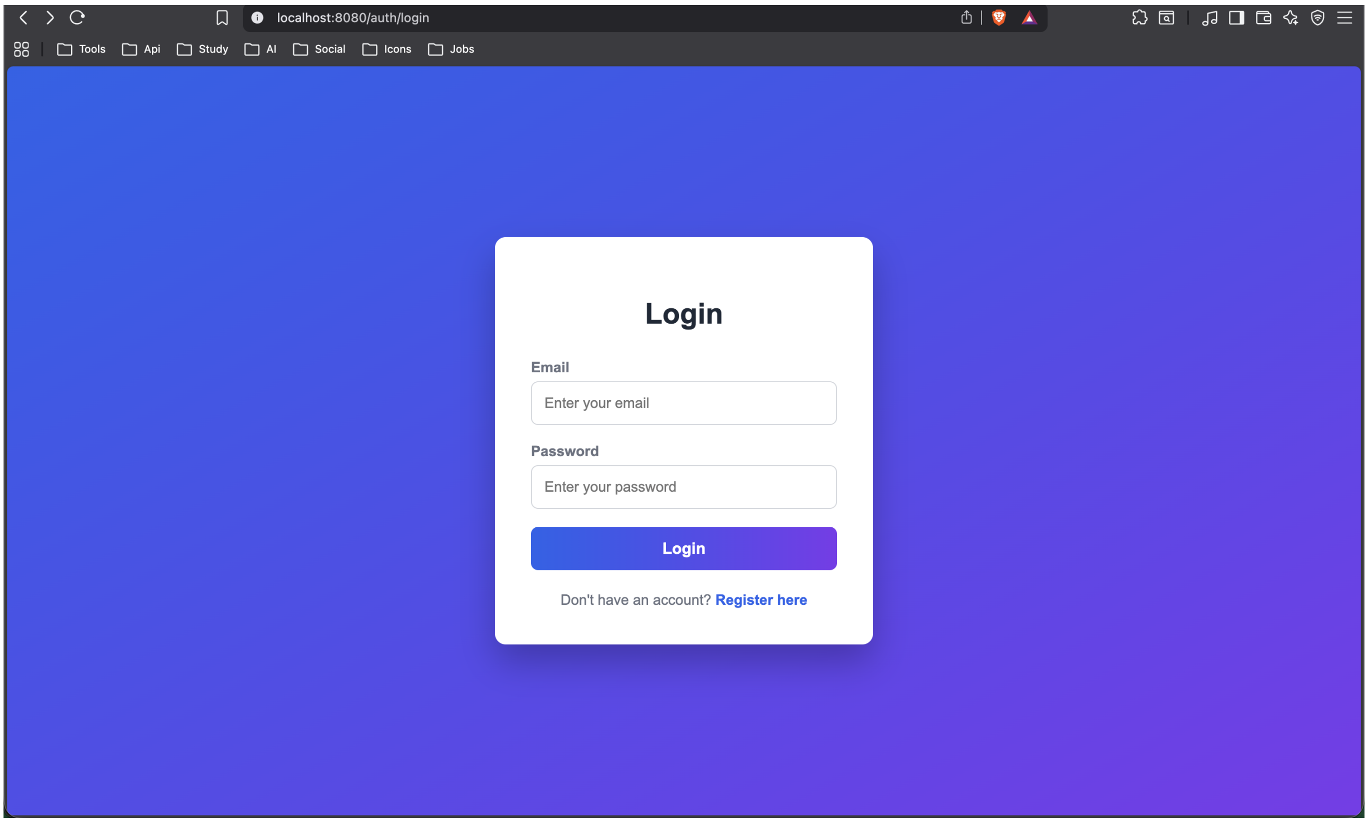
Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

The application was unable to connect to the login page due to incorrect routing

After Fix: Login Page



Login

Email

Enter your email

Password

Enter your password

Login

Don't have an account? Register here