

# Internship Progress Report

---

Name: Saiprakash Bollam

Internship Role: Research Intern

Duration: 3th September 2025 - 16th September 2025

Organization: Computer Science Department, Binghamton University

Supervisor: Zerksis Umrigar

Email: umrigar@binghamton.edu

## 1. Introduction

This week was dedicated to test stabilization and debugging. After implementing multiple features and their test cases in prior weeks, a number of failures persisted due to Spring Security misconfigurations, database schema mismatches, and Thymeleaf template dependencies. AI tools were used extensively to analyze stack traces, propose corrections, and automate iterative fixes. The end result was significantly improved test coverage and reliability.

## 2. Objectives

- Fix failing unit, integration, and security tests.
- Resolve Spring context loading issues (bean conflicts, missing configs).
- Correct schema.sql and test-data.sql mismatches with entity mappings.
- Mock dependencies (UserService, ContactService, SessionHelper) to reduce reliance on external systems.
- Achieve a higher test pass rate and maintainability.

## 3. Key Fixes and Features

- Security Tests:
  - Resolved bean conflicts with PasswordEncoder by consolidating TestSecurityConfig.
  - Updated test authentication flows to align with actual SecurityConfig.
  - Mocked UserService and ContactService to avoid real DB dependencies.
- Controller Tests:
  - Fixed ContactControllerTest by registering SessionHelper with correct bean name.
  - Added missing mocks for QRCodeGeneratorService.
  - Resolved PageControllerTest failures by mocking UserRepo.
- Integration Tests:
  - Skipped real email sending by mocking EmailService for test profile.
  - Fixed duplicate user email issues in test-data by assigning unique emails.

- Updated Contact entity to enforce proper @GeneratedValue strategy.

- Session Management Tests:

- Reconfigured login endpoints (/authenticate) and CSRF tokens.
- Adjusted schema.sql to match entity mappings (users table, roles, provider fields).
- Handled reserved keywords in H2 by renaming/escaping.

#### 4. AI's Role in Development

AI agents (Copilot & Gemini) played an essential role:

- Debugging: Interpreted Spring Boot test failures and proposed fixes (e.g., bean naming, duplicate configs).
- Configuration Management: Consolidated multiple overlapping TestSecurityConfig files into one working version.
- Schema & Data Fixes: Highlighted mismatches between entity annotations and schema.sql/test-data.sql.
- Automation: Iteratively ran tests, applied patches, and validated fixes.
- Optimization: Suggested cleanup of redundant files (e.g., duplicate mail configs, unused TestSessionHelper).

#### 5. Challenges Faced

- Frequent bean definition conflicts in Spring Boot test environments.
- Managing multiple conflicting test configuration files (e.g., TestSecurityConfig, TestMailConfig).
- Debugging integration test failures caused by Cloudinary and email properties.
- Ensuring AI-generated changes aligned with project conventions and real-world scenarios.

#### 6. Insights on AI Usage – Pros and Cons

Pros:

- Accelerated debugging and resolution of complex Spring Boot test failures.
- Reduced manual trial-and-error with precise patch suggestions.

- Helped enforce clean code practices by removing redundant/unused test files.
- Provided learning opportunities on Spring Security testing best practices.

Cons:

- AI occasionally generated verbose or redundant fixes, requiring manual refinement.
- At times, conflicting suggestions from different AI agents needed reconciliation.
- Could not always anticipate environment-specific errors (e.g., H2 vs. MySQL differences).

## 7. Outcomes & Learning

- Achieved a much higher test pass rate (most test suites now 100%).
- Established a more reliable test infrastructure with consistent mock configurations.
- Learned the importance of synchronizing entity, schema, and test-data.sql.
- Gained deeper insights into Spring Boot testing strategies: WebMvcTest, MockBean usage, and integration setups.