# Internship Progress Report – 6

**Name:** Saiprakash Bollam
**Internship Role:** Research Intern
**Duration:** 4th December 2025 - 17th December 2025
**Organization:** Computer Science Department, Binghamton University
**Supervisor:** Zerksis Umrigar
**Email:** umrigar@binghamton.edu

# 1. Introduction

This week focused on implementing **new security improvements** identified during the previous security audit. The work aimed to strengthen authentication, authorization, session handling, and configuration security across the Smart Contact Manager application. Emphasis was placed on aligning the system with modern **Spring Security 6.x**, **Spring Boot 3.x**, and **OWASP best practices**, while ensuring backward compatibility and system stability.

AI-assisted tools were extensively used to validate configurations, detect misconfigurations, and recommend secure design patterns.

---

# 2. Objectives

- Enhance application security based on findings from the prior audit.
- Strengthen authentication and authorization mechanisms.
- Improve session management and endpoint protection.
- Eliminate remaining insecure configurations and weak defaults.
- Validate security improvements using AI-assisted analysis.

---

# 3. New Security Improvements Implemented

## 3.1 Authentication & Authorization Enhancements

- Strengthened role-based access control (RBAC) across protected endpoints.
- Ensured consistent usage of `GrantedAuthority` mappings for authenticated users.
- Validated secure password encoding mechanisms using modern `PasswordEncoder` implementations.

## 3.2 Session & Access Control Improvements

- Improved session management policies to prevent unauthorized reuse of sessions.
- Verified proper handling of authenticated vs. public endpoints.
- Ensured secure default behavior for unauthenticated requests.

### 3.3 Configuration Hardening

- Removed weak or unnecessary default security configurations.
- Enforced secure HTTP headers using modern Spring Security defaults.
- Verified absence of hardcoded secrets and sensitive configuration values.

### 3.4 Codebase Security Cleanup

- Removed redundant or legacy security-related code fragments.
- Improved consistency in security-related annotations and configuration classes.
- Standardized constructor-based dependency injection across security components.

---

## 4. AI's Role in Security Improvement

AI tools such as ChatGPT and GitHub Copilot supported this phase by:

- Reviewing security configurations for misconfigurations or weak defaults.
- Suggesting secure alternatives for deprecated or risky patterns.
- Validating the correctness of role-based access logic.
- Highlighting potential edge cases in authentication and session handling.
- Providing explanations for security best practices and their practical implications.

The AI-assisted approach significantly reduced the time required to validate security changes and increased confidence in the final implementation.

---

# 5. Challenges Faced

- Ensuring stricter security rules did not unintentionally block legitimate access.
- Balancing usability with enhanced security controls.
- Verifying changes across multiple user roles and access paths.
- Reviewing AI-generated recommendations to filter out non-applicable suggestions.

---

# 6. Insights on AI Usage

## Advantages

- Accelerated validation of security configurations.
- Helped identify overlooked edge cases in access control logic.
- Improved understanding of advanced Spring Security concepts.
- Reduced manual effort during security refinement.

## Limitations

- Some recommendations required contextual judgment before implementation.
- AI occasionally proposed overly restrictive configurations.
- Manual testing was still essential to confirm runtime behavior.

---

# 7. Outcomes & Learning

By the end of this week, the application achieved a **stronger and more consistent security posture**. The new improvements reduced potential attack surfaces, improved access control clarity, and ensured alignment with modern Spring Security standards.

This phase reinforced the importance of **iterative security enhancement** and demonstrated how AI can act as a powerful assistant when combined with human expertise and testing discipline.

---

## Conclusion

This week marked a transition from identifying vulnerabilities to **actively strengthening and future-proofing the application's security architecture**. AI-assisted development played a key role in accelerating improvements while maintaining system reliability.