

Internship Progress Report – 3

Name: Saiprakash Bollam

Internship Role: Research Intern

Duration: 21st January 2026 - 3rd February 2026

Organization: Computer Science Department, Binghamton University

Supervisor: Zerksis Umrigar

Email: umrigar@binghamton.edu

1. Introduction

During this week, the primary focus was on evaluating and implementing development decisions derived from AI-assisted technical discussions. Specifically, the workflow followed “**Path A**”, which emphasized structured AI-guided implementation, prioritizing maintainability, scalability, and security while developing new application improvements.

This phase involved reviewing multiple AI-generated design alternatives, comparing implementation approaches, and selecting the most efficient development pathway. The process demonstrated how AI can support decision-making during software development rather than solely assisting with coding.

2. Objectives

- Analyze multiple AI-generated implementation strategies.
 - Select the most scalable and maintainable development pathway.
 - Implement features following structured AI recommendations.
 - Validate AI-generated architecture suggestions against existing project requirements.
 - Improve overall project reliability and development workflow efficiency.
-

3. Activities Performed

3.1 AI-Assisted Decision Analysis

Multiple implementation paths were generated through AI consultation. These included:

- Alternative controller handling strategies
- Different service-layer abstraction approaches
- Multiple validation and security handling mechanisms

Each solution was evaluated based on:

- Code maintainability
- Security compliance
- Compatibility with existing Spring Boot architecture
- Long-term scalability
- Development complexity

Following comparative analysis, **Path A** was selected due to its structured architecture and strong alignment with best software engineering practices.

3.2 Implementation and Refinement

After selecting the preferred approach, implementation involved:

- Refactoring existing components to align with selected architecture.
- Ensuring proper separation of concerns between controllers, services, and repositories.
- Improving validation flow to reduce redundancy.
- Enhancing modular design to simplify future feature additions.

AI tools were used to:

- Generate optimized implementation patterns.
 - Suggest improvements to controller-service communication.
 - Identify potential runtime issues before execution.
-

3.3 Testing and Validation

Once implementation was completed:

- Functional tests were executed to verify correct behavior.
- Edge cases were tested to ensure system reliability.
- Integration tests confirmed compatibility with existing features.

AI-assisted debugging helped interpret test failures and recommend targeted corrections, which reduced manual troubleshooting time.

4. AI's Role in Development

AI tools such as ChatGPT and GitHub Copilot provided significant support throughout this week:

- Generated multiple architectural alternatives.
- Assisted in evaluating trade-offs between different implementation approaches.
- Provided explanations for recommended design patterns.
- Suggested debugging strategies during validation testing.
- Helped maintain consistency with Spring Boot best practices.

AI acted as a **technical consultant**, supporting both design and implementation phases.

5. Insights on AI Usage

Advantages

- Accelerated architectural decision-making.
- Reduced development time through automated pattern suggestions.
- Improved confidence in selecting scalable solutions.
- Enhanced understanding of software design trade-offs.

- Assisted in early detection of potential errors.

Limitations

- AI recommendations required developer verification.
 - Some solutions needed adaptation to match project-specific requirements.
 - AI occasionally suggested overly generalized implementations requiring refinement.
-

6. Outcomes and Learning

By the end of this week:

- A structured and scalable implementation approach was successfully integrated.
- The project architecture became more modular and maintainable.
- Development decision-making skills improved through comparative AI analysis.
- Greater understanding was gained on using AI as a **design support tool** rather than solely as a coding assistant.

This week emphasized how AI can enhance **software architecture planning and decision-making** processes.

7. Future Work

- Expand Path A architecture into additional project modules.
- Strengthen automated testing for newly implemented components.
- Introduce documentation for new architectural standards.
- Continue leveraging AI for design validation and performance optimization.

Conclusion

This week demonstrated the effectiveness of integrating AI into software development decision-making. By selecting and implementing Path A, the project achieved improved modularity, scalability, and maintainability. The experience reinforced the importance of combining AI-generated insights with human technical judgment to achieve optimal software engineering outcomes.