

Internship Report

Integration of AI in “Forgot Password” Module Development Using Java Spring Boot

Name: Saiprakash Bollam

Internship Role: Research Intern

Duration: 25th June 2025 - 8th July 2025

Organization: Computer Science department, Binghamton University

Supervisor: Zerkis Umrigar

1. Introduction

As part of my internship, I undertook the task of implementing the "Forgot Password" functionality in a partially completed Java Spring Boot project. This feature included password reset link generation, token validation, email integration, and password update mechanisms. To assist with development, I integrated OpenAI's conversational AI tools (specifically, ChatGPT) into my development workflow. This report outlines the experience, the technical aspects, and an evaluation of AI's role in the development process.

2. Objective

To design and implement a secure, user-friendly "Forgot Password" workflow that aligns with existing project standards, and to leverage AI tools to assist in:

- Code generation
- Debugging
- Validation implementation
- UI integration
- Email handling

3. Development Summary

3.1 Features Implemented:

- "Forgot Password?" link on the login page
- Email input form for reset request
- Reset token generation and email dispatch
- Secure password reset form with validation
- Token validation and expiry check
- Password update in the database
- Email confirmation after successful reset
- Integration into the existing project UI and architecture

3.2 Technologies Used:

- Java 17, Spring Boot 3
- Thymeleaf for server-side templating

- JPA (Hibernate), MySQL
- Tailwind CSS (existing styles)
- Jakarta Mail for email services

4. Role of AI in Development

4.1 Planning & Architecture:

- Generated structured development plans.
- Proposed token-based security mechanism with expiry.
- Identified necessary database and controller changes.

4.2 Code Generation:

- Created entity classes (e.g., PasswordResetToken)
- Generated repository, service interface, and implementation.
- Developed and validated Spring controllers for form handling.
- Produced Thymeleaf-compatible HTML templates.

4.3 Validation & UX:

- Introduced form validation using Java Bean Validation
- Diagnosed Thymeleaf binding issues and corrected template errors.
- Ensured error messages were properly displayed to users.

4.4 Debugging & Support:

- Analyzed and explained stack traces.
- Suggested corrections for fragment loading issues in templates.
- Helped resolve BindingResult errors and missing model attributes.

4.5 Email Service Integration:

- Integrated password reset email logic.
- Diagnosed limitations in using demo SMTP accounts.
- Provided alternatives for development and production environments.

5. Prompt Engineering Insights

Prompt Strategy	Effectiveness
Clear task-specific prompts	Precise implementation
Stack trace input for debugging	High accuracy in diagnosis
Step-by-step refinement	Maintained codebase structure
Asking for validation rules and UI consistency	Ensured error-free and user-friendly interface

6. Advantages of AI Assistance

Area	Benefit
Development Speed	Drastically reduced time to implement full module
Accuracy	Produced code matching Spring and Java standards
Debugging	Resolved complex template and binding issues
UI Consistency	Helped maintain project styling and structure
Learning Support	Explained reasoning behind validation and controller logic

7. Challenges and Limitations

Issue	Description
Email Sending	Restricted by SMTP provider (only test email allowed)
Context Retention	AI needed reminders about project structure or naming
Field Binding	Required manual inspection of error field names
Overgeneration	Occasionally produced more code than required (needed trimming)

8. Conclusion

The integration of AI in my development workflow significantly improved both productivity and learning. Tasks that might have taken several hours were completed in a fraction of the time, with the added benefit of real-time feedback and suggestions. From planning to debugging, the AI provided structured guidance that ensured code quality and maintainability.

This experience demonstrated how AI can augment software development, particularly for interns and junior developers. With responsible use and human oversight, AI tools can become invaluable technical collaborators in modern development environments.