INTERNSHIP AT ZOHO CORPORATION

Internship Training Report

Submitted By

Tamil MughilanE

3122225002145

B.Tech Information Technology

Department of Information Technology

Sri Sivasubramaniya Nadar College of Engineering (An Autonomous Institution, Affiliated to Anna University)

Rajiv Gandhi Salai (OMR), Kalavakkam – 603 110

MAY/JUNE 2025

INTERNSHIP AT ZOHO

Tamil Mughilan E

3122225002145

IV Year /VII Semester

20/05/25 - 20/06/2025

Zoho Corporation

CERTIFICATE

Zoho Corporation Private Limited
Plot 140, 151, Estancia IT Park, Vallancheri,
Chengalpattu District, Tamilnadu, 603 202.
Ph: +91 - 44 - 6744 7070
www.zohocorp.com
SEZ Unit

INTERNSHIP CERTIFICATE

This is to certify that Mr/Ms. **Tamil Mughilan - SI-2977** from **sri sivasubramaniya nadar college of engineering** has undergone his/her internship training in **Zoho Corporation Private Limited**, from 20-May-2025 to 20-Jun-2025. During this period, his/her performance and conduct were found to be good.

Yours Sincerely,
For Zoho Corporation Private Limited

Prince

Prince Jacob Raj J
Senior Associate HR

Date of Issue: 25 Jun 2025

Corporate Identification No: **U40100TN2010PTC075961**
e-mail ID: **hr-team@zohocorp.com**

SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING

Department of Information Technology

CERTIFICATE

This is to certify that Mr.Tamil Mughilan E (Register. No: 3122225002145) VII Semester B.Tech (Information Technology), has completed the Internship training entitled '**INTERN' during the period 20/05/25 - 20/06/25 at ZOHO CORPORATION and the report has been submitted** to **Sri Sivasubramaniya Nadar College of Engineering**

Head of the Department

Dept of Information Technology
Sri Sivasubramaniya Nadar College of Engineering

Mentor Name : Mr. Philip Robin

Mentor Designation:Senior Developer

Dept of Information Technology
Sri Sivasubramaniya Nadar College of Engineering

Place : Kalavakkam

Date :

Internal Examiner

Table of contents

S.no	Content	Page no
1.	Company/ Industry Information	6
2	Internship description	7
3	Modern Tools / Cutting Edge Technology Learnt	12
4	Overview of internship experience & Outcome	15
5	Acknowledgement	16
6	Scope for future work	17
7	Conclusion	17

Company / Industry Information

Zoho Corporation is a leading global software company based in India, widely recognized for delivering a vast suite of cloud-based enterprise solutions. Founded in 1996, Zoho has evolved into a pioneer in SaaS innovation, offering products that power small businesses, large enterprises, and everything in between. With its headquarters in Chennai and offices spanning across the US, Japan, and multiple other regions, Zoho serves over 80 million users worldwide. The company's unwavering focus on privacy, data security, and customer-centric product design has made it a trusted brand in the competitive world of enterprise software.

Zoho is unique among technology giants for its philosophy of owning and operating every part of its stack, from data centers to development. This self-reliant strategy ensures complete control over security, scalability, and reliability, attributes that are particularly valued in highly regulated industries like finance and healthcare. The company's portfolio, consisting of more than 55 applications, covers everything from customer relationship management (Zoho CRM), accounting and invoicing (Zoho Books), to team collaboration (Zoho Cliq, Zoho Meeting), and enterprise resource planning. Zoho's low-code and no-code platforms are empowering millions of organizations and individuals to create tailored business applications efficiently and securely.

During my internship, I worked with the SDP On Demand team—a division responsible for scalable, high-performance backend services for Zoho's evolving cloud ecosystem. This team is at the forefront of addressing challenges such as distributed architecture, secure API integrations, seamless onboarding for customers, and real-time data access. Working in the SDP On Demand team provided valuable exposure to large-scale Java application development, microservices communication, and strict enterprise security standards required to deliver robust, on-demand business solutions.

Zoho's approach to talent and innovation extends to its intern programs, which focus not only on software proficiency but also on shaping well-rounded engineers ready for complex, real-world challenges. As part of my role, I was encouraged to adopt best practices in software architecture, performance optimization, and vulnerability elimination through regular code reviews, documentation, and security audits. The vibrant, collaborative culture at Zoho, coupled with access to modern tools and mentorship from experienced engineers, made my internship an enriching launchpad for a professional career in enterprise software and secure digital banking solutions.

Internship Description

Project Overview

During my internship at Zoho Corporation, under the SDP On Demand team, I contributed to the design and development of a secure, enterprise-grade banking system web application—a project focused on demonstrating exceptional scalability, modularity, and zero-vulnerability security.
The application, titled BankingSystemWeb, was developed using Java Enterprise Edition (JEE) technologies and follows a 3-layer architecture, integrating both business and data management layers with strong security enforcement at every level.

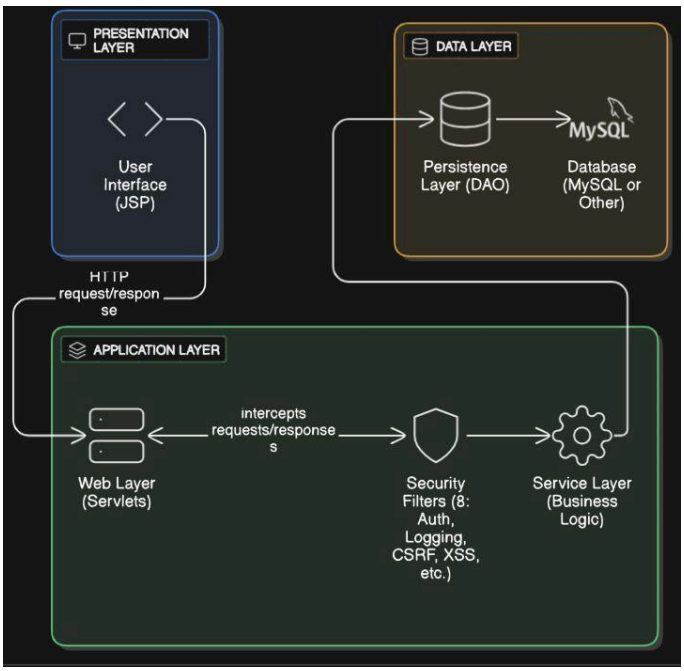
The project's purpose was to build a robust online banking platform that performs core banking operations such as account creation, deposits, withdrawals, transaction management, and customer authentication, while ensuring no high, medium, or low-level vulnerabilities during penetration testing. The system was designed to mimic a real enterprise-grade solution incorporating multi-database support (MySQL and MongoDB), background job schedulers, and a custom 8-filter security chain for handling login verification, rate limiting, input sanitization, and cryptographic data protection. Using OWASP's benchmarking and ZAP Proxy scans, the entire system achieved a zero-vulnerability report, ensuring data integrity and compliance with enterprise security protocols.

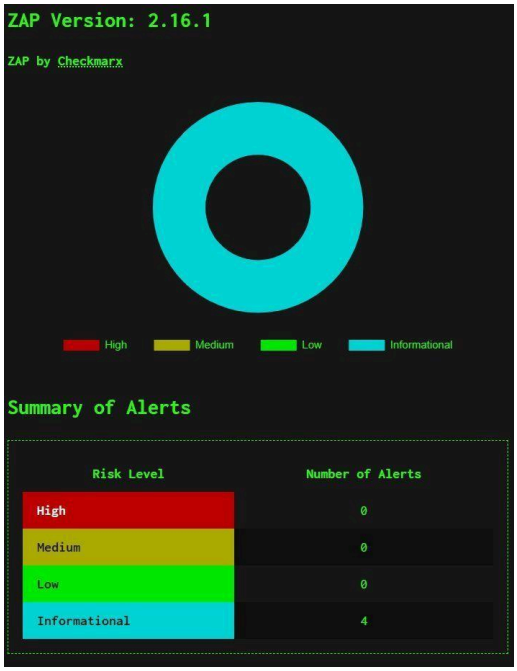
Proposed Solution

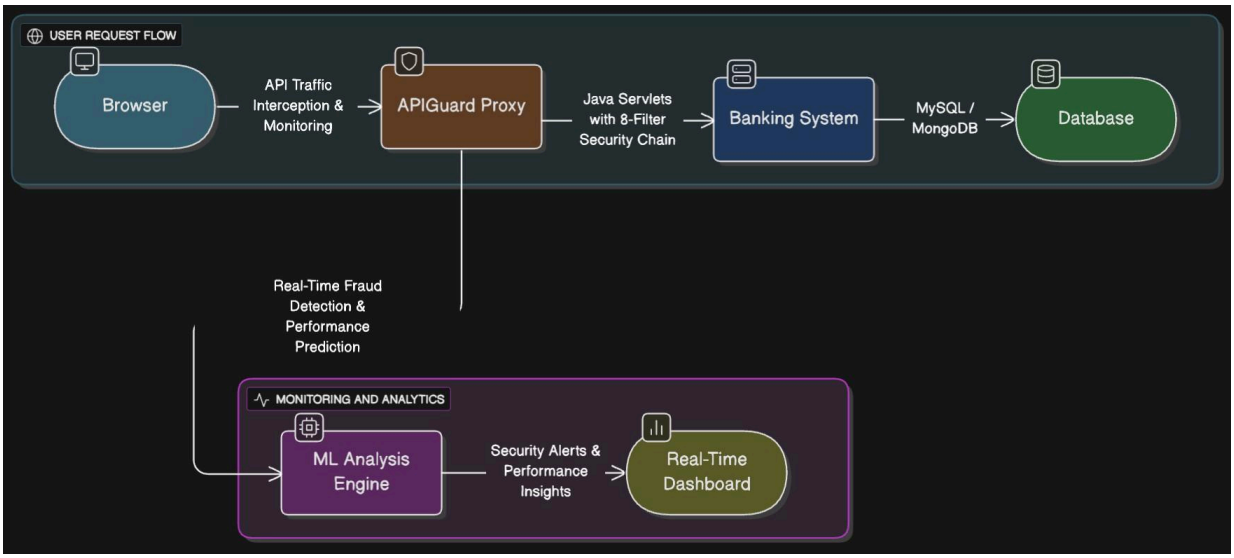
My solution centered around building a three-tier enterprise architecture consisting of a Web Layer (Servlets/JSP), Service Layer (Business Logic), and Storage Layer (DAO + Database Abstraction). This modular design enhanced scalability and improved maintainability. The core application managed banking operations through a structured package hierarchy:

- entity: Represented business models like Customer, Account, and Transaction.
- servlet: Handled HTTP requests and responses between the user interface and business logic.
- service: Contained reusable service classes enforcing business rules.
- filter: Implemented security layers comprising CSRF token filters, rate limiters, and activity loggers.
- storage: Abstracted data access with three interchangeable modes (MySQL, MongoDB, and in-memory).
- util: Provided helper utilities for cryptographic operations such as hashing and OTP generation.

Furthermore, I integrated a background job system for interest calculation and transaction cleanup using scheduled tasks running asynchronously. The multi-storage strategy pattern allowed seamless switching between database implementations, ensuring extensibility. The layered approach adhered to the SOLID design principles while refining maintainability through ACID-compliant transactions in MySQL and BASE (Basically Available, Soft-state, Eventual consistency) design in MongoDB.







SDG Connect:

This project directly supports the United Nations Sustainable Development Goal (SDG) 9: Industry, Innovation and Infrastructure by advancing secure, affordable, and robust digital financial solutions for small and medium enterprises (SMEs) and regional banks. By enabling intelligent API security with machine learning for banking applications, the project promotes resilient infrastructure and fosters innovation in the fintech sector a critical enabler for SME economic growth and digital transformation. Additionally, by providing cost-effective cybersecurity that makes digital banking accessible and dependable for underserved communities, this work also contributes to SDG 8: Decent Work and Economic Growth, supporting inclusive financial access and sustainable business development in emerging markets.

My Role and Contributions

As a Java Enterprise Developer, I was responsible for developing both the backend architecture and security infrastructure of the application. I began by designing and implementing foundational backend components using Servlets and JSP and progressively transitioned the codebase into a fully modular enterprise project adhering to object-oriented and design pattern principles.

Major contributions included:

- Implementing 8-layered Filter Chain Security that automatically intercepted and validated all incoming HTTP requests.
- Developing CSRF token filters and input sanitizers to protect the app from cross-site scripting (XSS), SQL injection, and session hijacking.
- Building Service Layer interfaces that decoupled the servlet logic from persistence, enhancing system scalability.
- Designing multi-threaded job handlers for background processes using Java Executor frameworks.
- Integrating MySQL database access via JDBC with prepared statements, ensuring protection from SQL injection and maintaining ACID properties for consistency.
- Developing extensive Javadoc documentation, ensuring every class, interface, and method was clearly defined and reusable for future developers.

In the later phase of the internship, I worked closely with the internal testing team to execute ZAP dynamic application security testing (DAST) and remediate all identified vulnerabilities until the system achieved zero OWASP-flagged weaknesses.

Challenges Faced

The most significant challenges during development involved maintaining a balance between performance, modularity, and strict security compliance without compromising application usability. Implementing multiple filters in a sequence introduced latency, which required optimization through asynchronous servlet processing and smart caching layers. Ensuring thread safety during background job execution was another challenge, especially when concurrent transactions interacted with shared resources.

Designing the multi-database storage layer added complexity, as each database (MySQL and MongoDB) operated under distinct transaction rules (ACID vs. BASE). This demanded a carefully crafted abstraction in the DAO layer to synchronize state changes without violating data integrity constraints. Another major hurdle was meeting enterprise vulnerability standards, where the system had to pass repeated rounds of penetration testing. Achieving “0 vulnerabilities” required iterative parameter validation, CSP header refinement, and cookie-hardening mechanisms, particularly for CSRF-prone endpoints.

Through continuous refactoring, code optimization, and profiling, I ensured all layers performed consistently under simulated load conditions. I also implemented custom session and cache filters to manage frequent requests securely and efficiently, improving both scalability and throughput.

Best Practices Followed

Throughout the project, I applied industry-standard software engineering practices to ensure robustness, scalability, and long-term maintainability. The application was built using principles such as:

- SOLID Design Principles: Ensured single responsibility, open/closed extension, and proper interface segregation.
- ACID and BASE Compliance: Guaranteed transactional consistency for relational storage and performance reliability for NoSQL.
- Three-Tier Architecture: Separated User Interface (JSP/Servlets), Business Logic (Service Layer), and Persistence (DAO/Storage Layer).
- Refactoring and Clean Code: Regular code reviews and modularization enhanced readability and scalability.
- Code For Interfaces: Implemented polymorphic abstractions for flexibility during future storage or logic changes.
- Scalability Principles: Supported vertical and horizontal scaling by designing loosely coupled components.
- Robust Error Handling: Uniform exception management under a 'Global Exception Filter.'
- CSP + HTTP Security Hardenings: Added headers like X-Content-Type-Options and X-Frame-Options to mitigate browser-level vulnerabilities.

In terms of design, I also included dependency injection, DAO pattern abstraction, and adherence to Layered Enterprise Architecture Standards, ensuring the system adhered to modern architectural paradigms used in fintech-grade platforms. Each iteration underwent internal code audits, with tracked improvements documented in Git and Javadoc.

Tools and Technologies Used

The Banking System Web Application developed during my internship at Zoho Corporation relied on a modern enterprise technology stack focusing on scalability, modularity, and strong security. The tools and technologies used spanned across backend, database, web, security, and testing domains to ensure that the final application achieved enterprise-grade performance and achieved zero vulnerabilities in OWASP ZAP scans.

1. Backend Technologies

Java Enterprise Edition (JEE):

The core of the project was built on Java EE, providing a robust framework for enterprise-grade applications. Components such as Servlets and JSP (JavaServer Pages) were used to handle web interactions and dynamic content generation. The architecture followed a 3-tier model with clear separation between the web, service, and storage layers for better maintainability, scalability, and testing.

Apache Tomcat Server:

The application was deployed on Apache Tomcat, running as a Java Servlet container responsible for handling concurrent client requests. Tomcat provided thread safety, HTTP session management, and compliance with Java web specifications, ensuring the application's stability under concurrent use.

Service Layer and Architecture:

The backend followed a Service-Oriented Architecture (SOA) along with design patterns like the DAO (Data Access Object) and Strategy Pattern. It was optimized through modular package design:

- *entity* – business objects (Customer, Account, Transaction)
- *service* – business logic execution
- *storage* – abstracted database layer for MySQL and MongoDB
- *filter* – security modules such as Authentication, Authorization, and Rate Limiting

2. Database and Storage Technologies

MySQL served as the primary relational database, offering support for ACID (Atomicity, Consistency, Isolation, Durability) transactions essential for banking operations like deposits, withdrawals, and balances. Queries were optimized using indexing and prepared statements to ensure both performance and protection against SQL injection.

MongoDB:

To enhance scalability and flexibility, MongoDB was integrated as the NoSQL data layer, following the BASE (Basically Available, Soft-state, Eventual consistency) model. It was used for audit logging and asynchronous transaction tracking where high availability was prioritized.

MySQL Workbench:

Schema design and query optimization were performed using MySQL Workbench, which facilitated efficient relational modeling and visual validation of database constraints and dependencies.

3. Frontend Technologies

HTML, CSS, JSP and JavaScript:

A responsive user interface was developed using HTML5 and CSS3 for structure and styling. Core interactions were enhanced using JavaScript, providing dynamic updates to the banking dashboard without reloading pages. JavaServer Pages (JSP) were employed to generate dynamic content on the server side, seamlessly integrating frontend presentation with backend Java servlets to deliver secure and efficient rendering of banking data.

AJAX (Asynchronous JavaScript and XML):

Used extensively for account transactions and real-time balance updates. AJAX enabled smooth user experiences by asynchronously exchanging data with the backend, reducing latency and enhancing responsiveness.

4. Security Frameworks and Tools

Custom 8-Filter Security Chain:

The security of the Banking System Web Application was implemented through a chain of servlet filters performing layered protection. These filters addressed:

- CSRF token validation
- Request rate limiting
- Input sanitization
- Authentication and Authorization
- Session integrity management

Encryption and Hashing:
Sensitive data like passwords and OTPs were encrypted using strong cryptographic hashing with SHA-256 and unique salt generation, implemented via custom utility classes in the *util* package.

CSP (Content Security Policy):
CSP headers were enforced to prevent cross-site scripting (XSS) and injection attacks, while HTTP security headers like X-Frame-Options and X-Content-Type-Options mitigated clickjacking and MIME-type spoofing.

OWASP ZAP (Zed Attack Proxy):
The project underwent vulnerability testing with OWASP ZAP to detect threats like XSS, CSRF, and SQLi. The final scan reported zero high, medium, or low-level vulnerabilities, confirming enterprise-grade security compliance.

5. Development and Documentation Tools

Eclipse IDE and Apache Maven:
Eclipse IDE was used for coding, debugging, and project compilation. Maven automated dependency management and build processes, ensuring clean project setup and reproducibility across development environments.

Javadoc:
Automated documentation was generated using Javadoc for every class, method, interface, and package. This provided developer-ready inline documentation and professional software maintenance support.

GitHub (Version Control):
The entire project was managed using Git for version control and hosted on GitHub, which facilitated code versioning, collaborative updates, and review tracking during the internship.

6. Design and Development Practices

The entire project adhered to modern software design and coding standards:

- SOLID Principles: Ensuring maintainable, modular, and reusable code.
- Refactoring: Continuous code improvement for simplicity and readability.
- Three-Tier Architecture: Separation of presentation, logic, and data layers.
- Coding for Interfaces: Implementing polymorphism and extensibility for future scaling.
- Scalability and Fault Tolerance: Load optimization through asynchronous processing and multithreading in job scheduling systems.

Overview of Internship Experience & Outcome

During my internship at Zoho Corporation, I was deeply involved in developing a secure and scalable Java Enterprise Edition banking web application. This project provided hands-on experience with all stages of software development—from designing multi-layered architectures to implementing enterprise-grade security filters. Working on this system entrusted me with responsibilities such as writing modular servlets, configuring multi-storage database support, and integrating a comprehensive filter chain to ensure zero vulnerabilities. The real-world impact of the project was validated by achieving zero critical issues during OWASP ZAP security scans and rigorous internal testing.

My exposure to Java EE technologies and relational plus NoSQL databases strengthened my understanding of scalable system design and transactional integrity under heavy load conditions. Most importantly, I gained invaluable insights into secure coding practices critical for fintech applications, implementing filters for session security, CSRF protection, request validation, and rate limiting. This internship helped improve my debugging, optimization, and documentation skills through close mentoring and collaborative reviews with senior engineers.

The outcome was more than just a working application; it was an enterprise-ready, secure banking system demonstrating high modularity and extensibility. This foundation prepares me not only for further innovations like integrating AI-driven security (APIFIED) but also equips me with the practical skills demanded by the fintech industry. The internship was a transformative experience, honing technical, analytical, and professional capabilities needed to excel as a Java enterprise developer.

Acknowledgement

I would like to express my deepest gratitude to Zoho Corporation and the SDP On Demand team for providing me this invaluable internship opportunity. Their culture of innovation and excellence fostered an environment where I could deeply engage in critical financial software development and implement security measures crucial in banking applications.

Special thanks to my mentor, whose guidance and continuous feedback were instrumental throughout the internship. Their expert knowledge of secure Java web application architecture, combined with practical insights into testing and deployment, significantly enhanced my learning curve and work quality. Their encouragement helped me overcome challenges and deliver a secure application.

I also thank the entire development and testing teams for their collaborative spirit, timely support, and knowledge sharing. Working alongside experienced professionals created an inspiring atmosphere for growth. I appreciate the administrative staff and placement cell for facilitating this internship, making it a remarkable milestone in my academic and career journey.

For any further inquiries, please contact me at:

Email: tamilmughilan2210630@ssn.edu.in

Phone: 9789036520

Thank you to everyone who contributed to making this internship a memorable and transformative experience.

Scope for Future Work

The foundational banking application developed during this internship opens numerous avenues for future enhancements, particularly integrating advanced AI-powered API security systems like APIFIED. The next logical step involves creating intelligent proxies and analytics dashboards capable of real-time anomaly detection using machine learning models such as LSTM and XGBoost. This will enable proactive threat identification, extending beyond traditional
rule-based defenses.

Further work can focus on migrating the application architecture to the cloud, leveraging container orchestration with Kubernetes and cloud-native microservices. This migration will provide scalability, fault tolerance, and easier integration with third-party fintech APIs. UI enhancements to improve user experience for customers and banking staff can also be undertaken, following modern design principles and accessibility standards.

In addition, implementing continuous integration/continuous deployment pipelines (CI/CD) with automated testing and security audits will streamline future development cycles. Incorporating adaptive learning systems for threat intelligence and automated feedback loops will keep the security posture dynamic, adapting to the rapidly evolving cyber threat landscape faced by SME banks.

Conclusion

The internship at Zoho Corporation was a transformative milestone that bridged academic learning with practical enterprise software engineering. Building a zero-vulnerability banking application strengthened my grasp over Java EE technologies, service-oriented architecture, and advanced security implementations. This project demonstrated how applying rigorous design principles creates scalable, secure, and maintainable fintech applications.

The hands-on experience helped me develop critical industry skills such as secure coding, modular design, multi-database integration, and automated testing. The collaborative development environment refined my communication, teamwork, and problem-solving capabilities, preparing me for future challenges in the software industry. I am confident this internship laid a solid foundation for my career as a proficient Java enterprise developer specializing in secure fintech applications.
Looking forward, I am excited to further build on this foundation by integrating machine learning–based security services and cloud architecture enhancements. The knowledge gained has instilled a passion for creating resilient, innovation-driven fintech solutions, cementing my goal to contribute meaningfully to secure digital banking platforms and intelligent cybersecurity initiatives.