Tribhuvan University

Faculty of Humanities and Social Science

Backend Developer – Intern

At

Eshare Solution Pvt. Ltd

An Internship Report

Submitted to

Department of Computer Application

Pascal National College

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Bikash Shrestha | 6-2-1226-5-2020

February 2025

Under Supervision of

Ashok Kumar Pant

ESHARE
solutions

MENTOR’S RECOMMENDATION LETTER

TO WHOM IT MAY CONCERN

I am pleased to recommend **Mr. Bikash Shrestha**, who successfully completed their internship at **Eshare Solution Pvt. Ltd**. During this time, he made a significant impact, especially through his work on the Central Authentication System. He demonstrated excellent technical skills by implementing features like token-based authentication, OTP services, and API integrations, applying best practices such as OAuth 2.0 and SOLID principles.

Mr. Bikash worked well with the team, collaborating with frontend, mobile, and senior developers to deliver robust and secure solutions. His proactive approach, quick learning, and problem-solving abilities stood out throughout the internship.

I am confident in Mr. Bikash’s abilities and recommend them for any opportunity. He will be a valuable addition to any team. Feel free to contact me at +977 9860874940 for more details.

Sincerely,

Ramesh Bhusal
Project Manager
Eshare Solution Pvt. Ltd.

Tribhuvan University

Faculty of Humanities and Social Science

Pascal National College

SUPERVISOR’S RECOMMENDATION

I hereby recommend that this internship report prepared under my supervision by **Bikash Shrestha** entitled “**Backend Developer - Intern**” at “**Eshare Solution Pvt. Ltd.”** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

Ashok Kumar Pant

SUPERVISOR

Principal, Faculty Member
Department of Computer Application
Pascal National College, Satdobato, Lalitpur

Tribhuvan University

Faculty of Humanities and Social Science

Pascal National College

EXAMINER’S APPROVAL LETTER

This is to certify that this internship report prepared by **Bikash Shrestha** entitled “**Backend Developer - Intern**” at “**Eshare Solution Pvt. Ltd.”** in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

________________________ **Ashok Kumar Pant** Principal, Faculty Member Department of Computer Application Pascal National College Satdobato, Lalitpur	________________________ **Suresh Thapa** BCA Coordinator Department of Computer Application Pascal National College Satdobato, Lalitpur
________________________ (Internal Examiner)	________________________ (External Examiner)

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude and appreciation to all those who have supported and guided me throughout my internship.

First and foremost, I would like to extend my deepest thanks to my mentor **Shoaib Manandhar**, my project supervisor **Ashok Kumar Pant** and our BCA coordinator **Suresh Thapa**, for their invaluable guidance, expertise, and constant support. Their mentorship and constructive feedback have been instrumental in shaping the project and pushing us to deliver my best.

Lastly, I would like to acknowledge the countless individuals who have shared their knowledge and expertise, either through discussions, feedback, or research materials. Your contributions have greatly enriched my understanding and helped shape the final outcome of this project.

Sincerely,

Bikash Shrestha

ABSTRACT

This report outlines my internship experience at Eshare Solution Pvt. Ltd., an organization specializing in capital market solutions, including trading platforms, analytics tools, and back-office systems for brokers. During the internship, I worked in the Development Unit, where my primary responsibility was the design and implementation of the Central Authentication System. This system ensures secure, centralized user authentication for multiple services, leveraging advanced protocols such as OAuth 2.0. I contributed to key features such as token-based authentication, OTP services, and API integrations. Additionally, I worked on log segregation and management for the API Gateway. My role also involved synchronizing Market Close Stats in the Exchange Adaptor system and collaborating with cross-functional teams to deliver robust solutions. Through these tasks, I applied industry best practices, such as Gitflow and SOLID principles, and gained valuable insights into secure software development. Overall, the internship provided a rich learning experience, enhancing both my technical skills and my ability to work on complex, real-world projects in the financial technology sector.

Keywords: Central Authentication System, Authorization Server, Request Owner Authorization Flow

TABLE OF CONTENTS

MENTOR'S RECOMMENDATION LETTER i
SUPERVISOR'S RECOMMENDATION ii
EXAMINER'S APPROVAL LETTER iii
ACKNOWLEDGEMENT iv
ABSTRACT v
TABLE OF CONTENTS vi
LIST OF ABBREVIATIONS viii
LIST OF FIGURES ix
CHAPTER 1 1
INTRODUCTION 1
1.1 Introduction 1
1.2 Problem Statement 1
1.3 Objectives 1
1.4 Scopes and Limitation 2
1.4.1 Scopes 2
1.4.2 Limitations 2
1.5 Report Organization 2
CHAPTER 2 3
INTRODUCTION TO ORGANIZATION 3
2.1. Organization Details 3
2.2. Organizational Hierarchy 3
2.3. Working Domains of Organization 4
2.4. Description of Intern Department / Unit 5
CHAPTER 3 6
BACKGROUND STUDY AND LITERATURE REVIEW 6
3.1 Background Study 6
3.2 Literature Review 7
CHAPTER 4 9
INTERNSHIP ACTIVITIES 9
4.1. Roles and Responsibilities 9
4.2. Weekly log 10
4.3. Description of the Project(s) Involved During Internship 14

4.4. Tasks / Activities Performed 16

CHAPTER 5 18

CONCLUSION AND LEARNING OUTCOMES 18

5.1 Conclusion 18

5.2 Learning outcomes 18

REFERENCES 19

LIST OF ABBREVIATIONS

API	Application Programming Interface
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CAS	Central Authentication System
CORS	Cross Site Request Forgery
JAR	Java Archive
JDK	Java Development Kit
KYC	Know Your Customer
MFA	Multi-Factor Authentication
NEPSE	Nepal Stock Exchange
NOTS	NEPSE Online Trading System
OAuth	Open Authorization
OTP	One Time Password
QA	Quality Assurance
REST	Representational State Transfer
SMS	Short Message Service
SSRF	Same Site Request Forgery
TDD	Test Driven Development
TMS	Trade Management System
XSS	Cross Site Scripting

LIST OF FIGURES

Figure 2. 1 Organization hierarchy of Eshare Solution Pvt. Ltd. 3

Figure 3. 1 Resource Owner Password (ROP) Flow implemented in CAS 7

Figure 4. 1 Modules of Central Authentication System 14

Figure 4. 2 High-level system architecture of projects involved.......................................15

CHAPTER 1

INTRODUCTION

1.1 Introduction

During my internship at Eshare Solution, I worked on building important software for the company. This company creates tools for the stock market, like systems for trading stocks, helping brokers with technical support, and giving them tools to analyze data. My main project was creating a Central Authentication System (CAS), which is very important for keeping user access secure and organized. The Central Authentication System was developed to provide a unified and secure authentication mechanism for users across multiple brokerage services. After users log in, they can access different services like placing orders or managing their funds. My work focused on making sure the system was safe, easy to use, and reliable. I focused on implementing the OAuth 2.0 framework, specifically the Resource Owner Password Flow, to ensure secure authentication through access and refresh tokens. Additionally, I developed features like login, logout, password management, OTP-based authentication, and CAPTCHA integration. I also integrated SMS and email services, addressed CORS issues, and implemented cookie-based security measures, creating a robust system for secure user access.

This project gave me hands-on experience in creating secure systems and allowed me to learn more about how the capital market industry works.

1.2 Problem Statement

The existing broker's trading system require a unified and secure Central Authentication System (CAS) to address fragmented processes, enhance scalability, and protect against security threats. There is a need for a unified system to handle challenges like managing secure user access, combating threats such as unauthorized access and attacks, and improving integration with modern authentication standards.

1.3 Objectives

The objectives of this project are follows as:

- To design and implement a secure and centralized authentication system for managing user access across various brokerage services.
- To address security challenges and improve user experience by incorporating modern authentication standards and features like OAuth 2.0, OTP, and CAPTCHA.

1.4 Scopes and Limitation

1.4.1 Scopes

The scopes of this project are follows as:

- The development of a Central Authentication System (CAS) for secure user access management across various brokerage services.
- Implementation of OAuth 2.0 framework and related authentication features like token-based authentication, OTP, CAPTCHA, and password management.
- Integration of SMS, email services, and security measures such as CORS and cookie-based security to enhance functionality and user experience.

1.4.2 Limitations

The limitations of this project are follows as:

- The system relies on third-party services (SMS, email, CAPTCHA) that may cause dependency issues or delays during outages.
- The system relies on the client's security practices and device integrity for safeguarding sensitive operations.

1.5 Report Organization

The **first chapter** of the report contains the summarized introduction of the whole project report. It includes the overview, scope and limitation, problem statement and objectives of this project.

The **second chapter** includes the organization details, organization hierarchy, working domains of organization and description of department I worked on.

The **third chapter** includes background study i.e., description of fundamental theories, general concepts and terminology related to the project. It also includes the literature review i.e. review of the similar projects, research and theories done by others.

The **fourth chapter** contains internship activities with my roles and responsibilities, weekly logs, description of projects involved during internship and related activities performed.

The **fifth chapter** includes conclusion and learning outcome. It contains the final paragraphs of the report and in this phase the overall outcome and the developer point of view is written. The lesson learned through all the phases is included in this chapter.

CHAPTER 2

INTRODUCTION TO ORGANIZATION

2.1. Organization Details

Eshare Solution is a financial technology company that specializes in automating capital markets of Nepal. Eshare Solution is founded by group of professionals from Technology, Capital Market and Financial Service industry in 2021 to provide centralized Financial Technology solutions and related services in Nepal. Eshare Solution is the technology leaders in the Nepalese Stock Market with 98% of the stockbrokers as valued customers.

Organization's contact details:

- **Website:** https://esharesolution.com/
- **Address:** Pani Pokhari Heights, Kathmandu, Nepal
- **Mail:** support@esharesolution.com
- **Contact no.:** +977 9851027322

2.2. Organizational Hierarchy

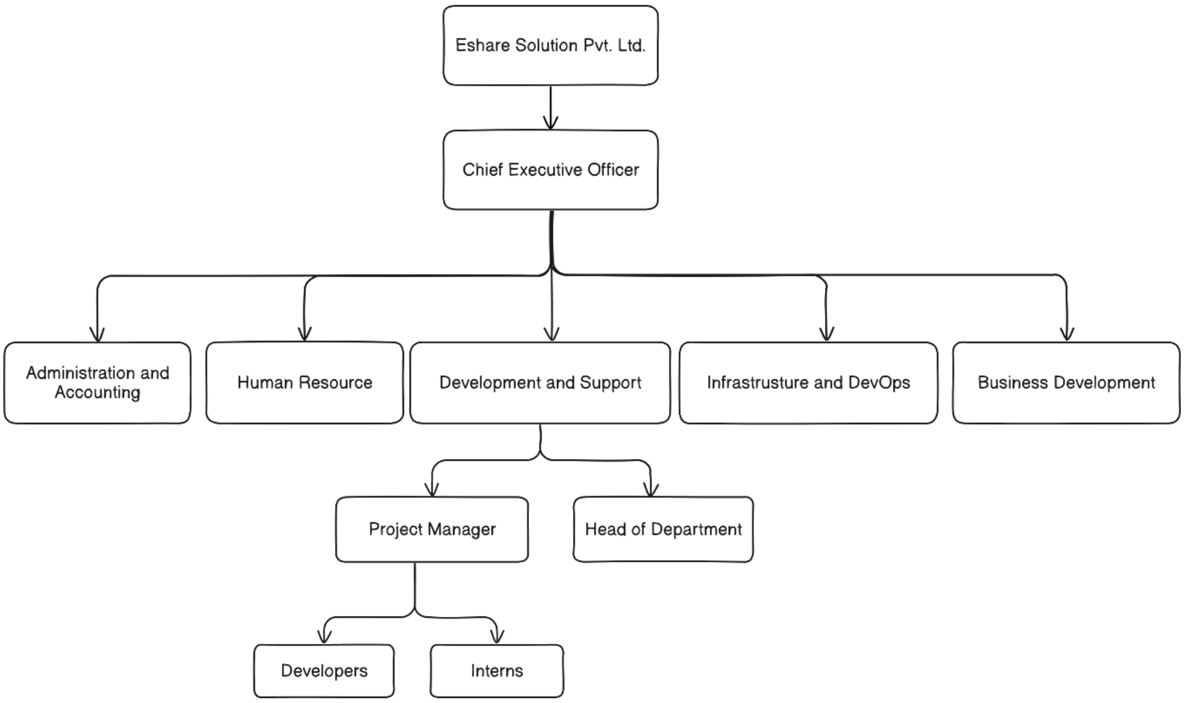

Figure 2. 1 Organization hierarchy of Eshare Solution Pvt. Ltd.

2.3. Working Domains of Organization

The Eshare Solution operates in the capital markets sector, offering innovative and comprehensive solutions tailored to meet the needs of brokers and investors. The organization focuses on delivering state-of-the-art technology and services to enhance market efficiency, streamline operations, and provide exceptional user experiences. Its working domains include:

Trading Platforms: The company provides a highly personalized and intelligent trading platform accessible via web-based, workstation-based, and mobile-based interfaces. This platform empowers market participants to invest and trade seamlessly across different mediums.

Data Analytics and Multi-Asset Trading Features: A dedicated research team delivers near real-time data analysis on listed companies, enabling informed decision-making for market participants. The system supports multi-asset trading with integrated features such as multiple market watches, order management, and real-time updates on fund and security balances. It accommodates diverse trading approaches, including margin trading and cash-and-carry trading.

Portfolio Information and Management: The organization offers real-time portfolio management tools, providing investors with profit and loss details on individual securities, thus fostering transparency and better investment tracking.

Broker Back Office Solutions: The organization offers post-trade operations, including KYC management, billing, integrated accounting, role-based security, and investor notifications. It includes a client portal and app for investors to track transactions, bills, and reports.

2.4. Description of Intern Department / Unit

Development and Support Department:

I worked in the Development and Technical Support Department, with a specific focus on the Development Unit. This unit plays a crucial role in designing and building innovative software solutions, aligning with the organization's working domains such as trading platforms, data analytics tools, portfolio management systems, and the broker back-office solutions.

The Development Unit is composed of a collaborative team that includes senior engineers, UI designers, frontend developers, QA specialists, and mobile developers. Together, they are responsible for developing, testing, and refining the organization's products to ensure they are secure, user-friendly, and aligned with client needs.

As part of the Development Unit, I had the opportunity to work on key projects, contributing to the creation and enhancement of robust software systems that address the dynamic requirements of the capital market industry. This role allowed me to gain valuable hands-on experience while collaborating with a skilled and dedicated team.

Internship Details:

- **Start date:** December 4th, 2025
- **End date:** February 4th, 2025
- **Duration:** 3 months
- **Office hours:** 10:15 am – 5:00 pm

CHAPTER 3

BACKGROUND STUDY AND LITERATURE REVIEW

3.1 Background Study

A capital market is a financial market where long-term securities like stocks and bonds are bought and sold. It's different from a money market, where short-term debt is traded (Majaski, 2024).

Capital markets channel savings into productive investments, which helps the economy grow. Capital markets provide market-based financing, which helps support financial stability. Capital markets help businesses access capital, which enables them to grow. Capital markets help households manage their savings (OECD, 2023).

Stock exchange is a marketplace where stocks, bonds and other securities are bought and sold. Stock exchanges have licensed brokerage firms in which traders must register to participate. The online trading system enables remote stock trading in the secondary market Treasury bonds, municipal bonds, corporate bonds, debentures, and shares or stocks issued by companies are traded in capital markets (Adam Hayes, 2024).

3.2 Literature Review

The literatures related to best security practices, authentication, authorization protocols, MFAs, logging conventions, exception handlings, caching, version control conventions, programming principles and design patterns were reviewed for internship projects. While working on the project, I applied these principles to improve maintainability and flexibility. This approach resulted in efficient and well-structured code (Bhardwaj, 2024).

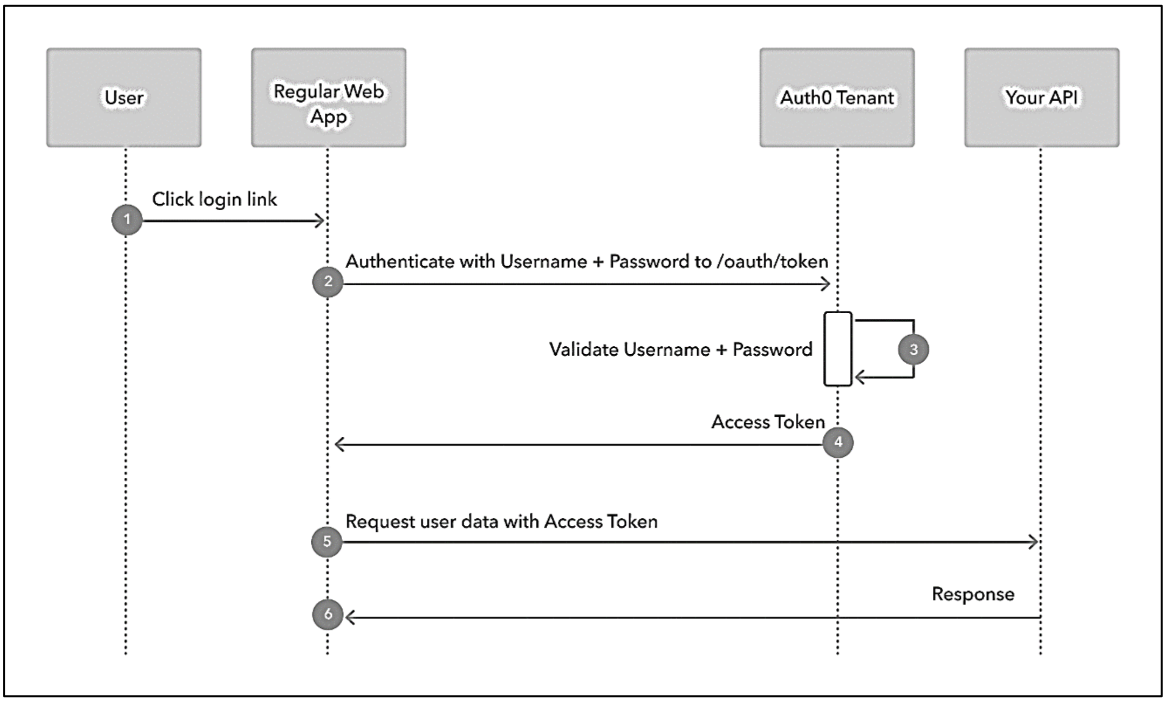
User
Regular Web App
Auth0 Tenant
Your API
1
Click login link
2
Authenticate with Username + Password to /oauth/token
Validate Username + Password
3
Access Token
4
Request user data with Access Token
5
Response
6

Figure 3. 1 Resource Owner Password (ROP) Flow implemented in CAS

Through studying the OAuth 2.0 authorization framework, Resource Owner Password Flow is ideal for the Central Authentication System. This flow allows users to authenticate securely by providing credentials to the client, which then obtains an access token from the authorization server to access protected resources. This approach is well-suited for centralizing user authentication via APIs, ensuring security and scalability while minimizing exposure of credentials. Applying this flow strengthened the system's authentication mechanisms and enhanced my understanding of OAuth 2.0’s practical applications (IETF, 2025) (IETF, 2024) (Dan & Bellen, 2021) (PortSwigger, 2025) (Parecki, 2019) .

The study of OTP best practices highlighted their importance in enhancing security against credential-based attacks like phishing and keylogging. OTPs are widely used for password-less login, multi-factor authentication, user verification, account recovery, and payment confirmations. Key principles I adopted include ensuring appropriate OTP complexity (6-10 characters), secure infrastructure, reliable service providers, retry mechanisms, and rate limiting. Implementing these practices in the CAS improved both security and user experience while reinforcing my understanding of secure authentication methods (Singh, 2022).

I gained insights into Gitflow, a structured version control workflow. Gitflow emphasizes branches like `master`, `develop`, `feature`, `release`, and `hotfix`, ensuring better organization, collaboration, and reduced code conflicts. Practicing Gitflow during my project allowed me to manage development efficiently, integrate changes seamlessly, and maintain code quality across different stages (Driessen, 2010).

CHAPTER 4

INTERNSHIP ACTIVITIES

4.1. Roles and Responsibilities

My responsibilities focused on the design, development, and implementation of secure and efficient systems, as well as addressing critical functionalities required in the capital market sector.

The key roles and responsibilities included are as follows:

- Work closely with frontend and mobile development teams to ensure seamless integration between software components and services.
- Collaborate with senior developers to design and develop robust software solutions, adhering to best practices and project requirements.
- Design and implement central authentication system to ensure seamless authentication mechanisms to optimize service efficiency and security.
- Add robust logging mechanism so that it will be easy find issues, by segregation and truncations by error levels.
- Create configurations to maintain logs in separate folders for different services and implement rollover conditions and log file deletion policies to manage storage.
- Dynamically schedule job for synchronizing Market Close Stats (MCS) to reconcile stock prices and managed price resets after holidays and pre-open periods, ensuring proper updates for market re-opening.

Through these responsibilities, I gained valuable hands-on experience in system design, API management, authentication protocols, logging configurations, and job scheduling. This role significantly enhanced my technical expertise while enabling me to contribute to impactful solutions in the capital market industry.

4.2. Weekly log

Week One (Dec 4 – Dec 10):

- Completed onboarding and set up the development environment, including the installation of JDK 17/22 and IntelliJ IDEA.
- Worked on the **Authentication Interceptor** using Spring Security and tested the generated JAR file.
- Attempted to configure security settings within the package and integrated them into the authentication service.
- Successfully implemented the authentication service, documented the interceptor, and wrapped up the task.

Week Two (Dec 11 – Dec 17):

- Began development of the **Centralized Authentication Module** to connect with the API Gateway.
- Learned to use debugging tools effectively and worked on understanding the authentication code structure.
- Attended a TMS information overview meeting for further context on system requirements.
- Debugged key components of the Centralized Authentication System, including `OAuthValidationRequestDispatcher` and `AbstractAuthenticationRealm`
- Focused on the login, ReCAPTCHA, and update password modules of the authentication system.

Week Three (Dec 19 – Dec 25):

- Researched CORS and Cookies to address cross-origin challenges in the system.
- Continued work on the **Centralized Authentication System**, including:
    - Update Password module
    - OTP generation and validation
    - Configuration for the k8 branch
    - Running the server with the database down
    - Enabling fake user login and third-party login/logout functionalities.

Week Four (Dec 26 – Jan 1):

- Enhanced the **Centralized Authentication System** with additional features:
    - Reset password functionality
    - Mail server configuration for notifications
    - API documentation for better developer reference.
- Created a merge request for the developer branch and integrated it successfully.
- Tested email setup for OTP delivery and developed the **System Options** and **OTP Service**.

Week Five (Jan 2 – Jan 8):

- Studied and implemented **Redis Cache** for the Centralized Authentication System to enhance performance.
- Resolved `ClassCastException` and researched vulnerabilities like Log4Shell, XSS, CSRF, and SSRF for security hardening.
- Implemented logging in the **API Gateway** for segregation, truncation, and formatting using Log4j.
- Worked on token management, including refresh tokens, access tokens, CSRF tokens, and HttpOnly cookies.

Week Six (Jan 9 – Jan 15):

- Conducted a revision of Spring Security concepts and continued work on resetting password functionality.
- Generated encrypted JWT tokens and tested the reset password service.
- Documented changes made to the system and resolved unresolved threads and issues in the system.

Week Seven (Jan 16 – Jan 22):

- Developed job scheduling for **Market Close Stats (MCS)** synchronization in the Exchange Adaptor.
- Tested and resolved issues with MCS sync job scheduling and implemented dynamic retries for up to 10 attempts.
- Created test cases for unique OTP generation and added expiration checks.
- Followed Test Driven Development (TDD) and SOLID principles while refining project components.

- Resolved issues in the `mcs-sync` branch and documented the Centralized Authentication System.

Week Eight (Jan 23 – Jan 29):

- Resolved database reconnection retries and added a secret key in headers for the Exchange Adaptor.
- Studied OTP best practices and implemented them using **RedisTemplate** for secure OTP delivery.
- Revised and refactored the OTP service with input from senior developers.
- Worked on integrating the SMS-service and addressing WebClient headers issues in MCS sync.

Week Nine (Jan 30 – Feb 4):

- Implemented reset password functionality for the mobile application and coordinated with mobile developers.
- Integrated the SMS-service into client-login and resolved client-login errors in the TMS_v2 module.
- Refactored token services and implemented a resend OTP API endpoint.
- Separated login and logout logic from `OAuthValidationDispatcher` and standardized responses across APIs.

Week Ten (Feb 5 – Feb 11):

- Enhanced exception handling and added an endpoint for expired passwords.
- Collaborated with senior developers to finalize password expiration logic and documented API requests and responses.
- Merged SMS-service and other branches into a code-refactor branch, implemented Swagger API documentation, and handled refresh token exceptions.
- Deployed and fixed deployment errors in the `TMS_v2` branch.

Week Eleven (Feb 12 – Feb 18):

- Tested dynamic data-source endpoints and resolved issues with duplicate member code fields.
- Added validation for new and confirmed passwords and included memberCode fields in payloads to distinguish users from different brokers.
- Merged the `tms_v2` branch into `feature/api-response`.

Week Twelve (Feb 19 – Feb 25):

- Updated custom exceptions from runtime exceptions to standard exceptions for better error management.
- Studied OAuth 2.0 Security Best Practices and applied relevant improvements to the CAS.
- Refactored CAS code, separating access and refresh token logic into `OAuthService`.
- Improved CSRF token and security mechanisms based on architectural discussions.

Week Thirteen (Feb 26 – Mar 5):

- Refined token services by separating caching token dependencies and renaming tokens to keys with added member-code handling.
- Worked on password update logic while ensuring secure member code checks.
- Delivered an internship presentation summarizing my contributions and technical learning.

4.3. Description of the Project(s) Involved During Internship

Central Authentication System (CAS)

The Central Authentication System was a pivotal project aimed at providing secure, centralized user authentication across multiple services. My work involved designing and implementing features such as login, logout, password management, OTP generation and validation, ReCAPTCHA integration, and token-based authentication using OAuth 2.0. Additionally, I implemented Redis Cache to optimize performance and refactored the system to improve API responses and support seamless mobile integration. This project emphasized security, scalability, and user-friendliness while addressing vulnerabilities and ensuring robust authentication mechanisms. This project aimed to design and implement a secure OTP service following best practices. I developed unique OTP generation, expiration checks, and retry mechanisms, ensuring reliability and security. The OTP service was integrated into the CAS with additional support for SMS-service configuration, further enhancing user verification and authentication processes.

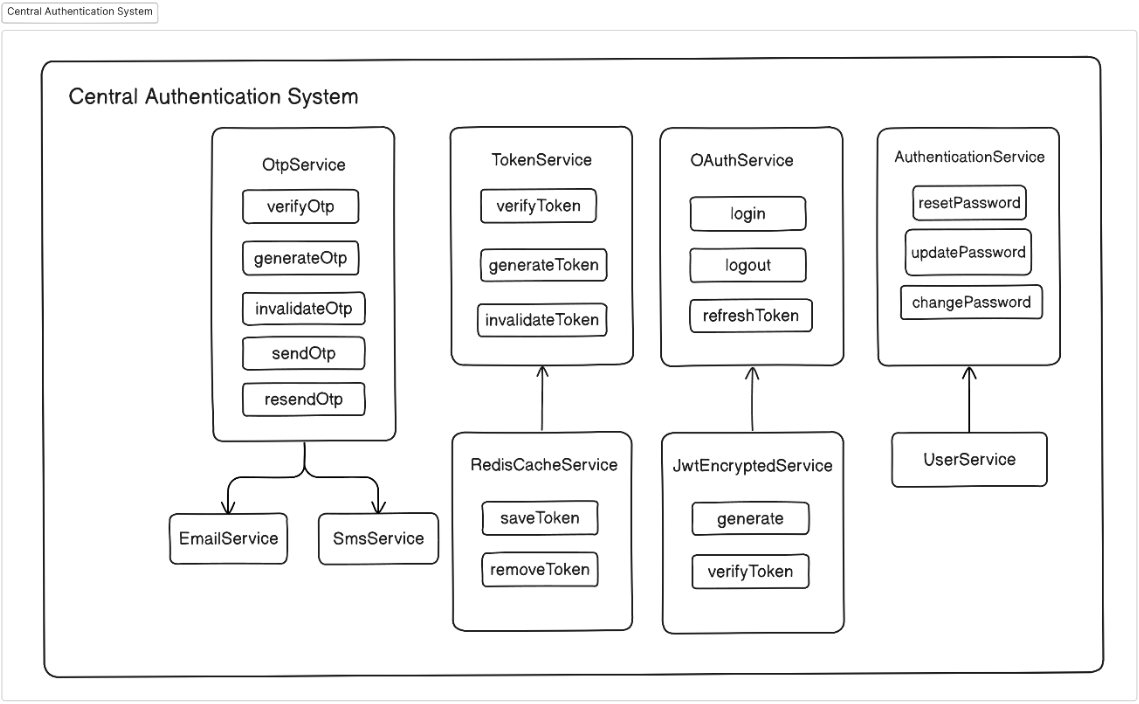

Figure 4. 1 Modules of Central Authentication System

Authentication Interceptor

The Authentication Interceptor project aimed to standardize the validation of API requests across multiple services. I created a reusable interceptor JAR file using Spring Security, which significantly reduced code duplication and improved security consistency. The solution was tested thoroughly and documented for seamless integration into the organization’s services.

API Gateway Logging and Management

This project focused on enhancing the API Gateway's logging and response management. I configured log segregation, truncation, and formatting using Apache Log4j, ensuring clear organization of logs by level (errors, info, warnings). Additionally, I implemented standardized exception handling, dynamic log management, and generic API responses to enhance system reliability and maintainability.

Exchange Adaptor – Market Close Stats (MCS) Synchronization

The Exchange Adaptor project involved developing job scheduling mechanisms to synchronize Market Close Stats. I ensured the system handled tasks such as resetting stock prices after holidays or pre-open sessions and reconciling stock prices with NOTS to manage packet loss during trading hours. The project focused on ensuring the accurate multicast of stock prices to brokers, improving system reliability and data accuracy.

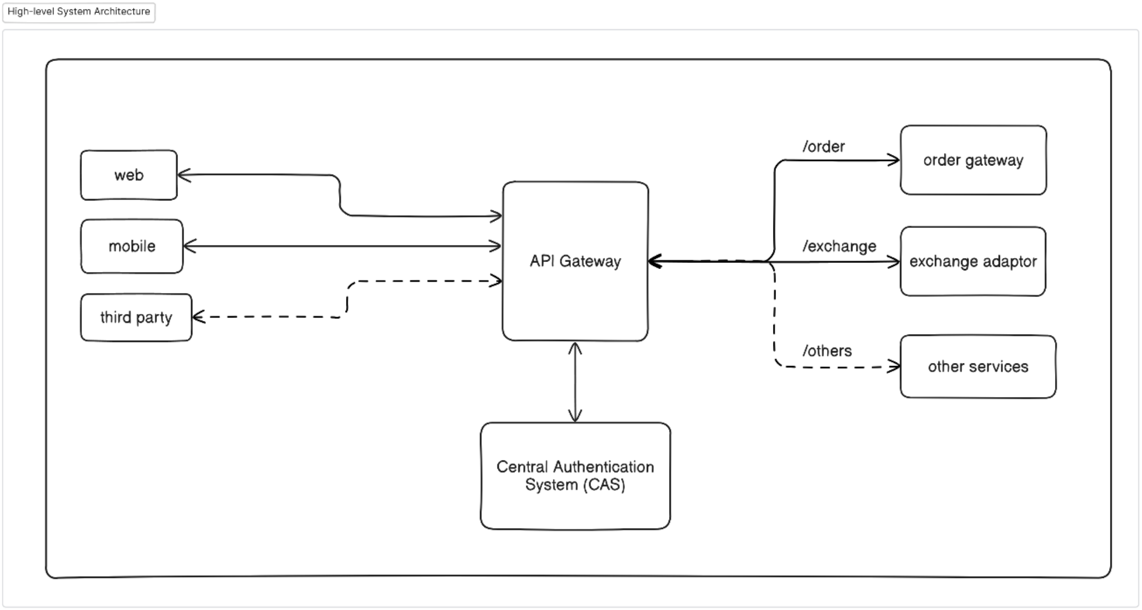

Figure 4. 2 High-level system architecture of projects involved

4.4. Tasks / Activities Performed

Centralized Authentication System (CAS)

- Developed and implemented features such as login, logout, password management, OTP generation, and ReCAPTCHA integration.
- Configured Redis Cache for enhanced performance and implemented token-based authentication (access tokens, refresh tokens, CSRF tokens).
- Conducted refactoring and code optimizations for CAS, including separating access and refresh token logic and improving API responses for mobile integration.
- Managed tasks like reset password services, API documentation, SMS-service configuration, and system options.
- Ensured robust security by addressing vulnerabilities like Log4Shell and implementing CSRF token mechanisms.

Authentication Interceptor

- Created a reusable JAR file to authenticate API requests across multiple services using Spring Security, reducing code duplication.
- Tested and documented the authentication interceptor for effective implementation.

API Gateway

- Configured log segregation, truncation, and formatting using Log4j.
- Implemented standardized exception handling, logging mechanisms, and generic API responses to enhance system reliability.

Exchange Adaptor

- Developed job scheduling mechanisms for Market Close Stats (MCS) synchronization to multicast stock prices to brokers.
- Managed synchronization after-market closures, handled packet loss, and introduced retry mechanisms for job scheduling.
- Resolved issues related to web client headers and refined logic for MCS updates.

OTP Service

- Designed and implemented a secure OTP service following best practices, including unique OTP generation, expiration checks, and retry mechanisms.
- Refactored OTP service to improve efficiency and integrated it with SMS-service for seamless delivery.

Collaboration and Knowledge Building

- Coordinated with frontend, mobile, and senior developers for seamless integration of services.
- Gained proficiency in debugging, using IntelliJ IDEA, and configuring development tools.
- Applied concepts like SOLID principles, Gitflow, and OAuth 2.0, enhancing code quality and security.

Documentation and Reporting

- Documented implementation details, API requests, and changes made to the systems.
- Prepared the internship report and delivered an internship presentation summarizing my contributions.

CHAPTER 5
CONCLUSION AND LEARNING OUTCOMES

5.1 Conclusion

The internship at Eshare Solution Pvt. Ltd provided me with an invaluable learning experience, allowing me to contribute to meaningful projects in the capital market sector. Working in the Development Unit, I had the opportunity to design and implement the Central Authentication System focusing on secure and efficient user authentication. Through tasks such as token-based authentication, OTP services, and integrating APIs, I applied industry best practices and enhanced my technical knowledge. Additionally, I collaborated with cross-functional teams, honed my problem-solving skills, and gained hands-on experience in applying concepts like OAuth 2.0, SOLID principles, and Gitflow in real-world scenarios. This experience not only deepened my technical expertise but also enhanced my understanding of software development workflows and their importance in delivering robust and scalable solutions.

Overall, the internship was a transformative journey that prepared me for future challenges in software development, enabling me to contribute effectively to complex systems in dynamic industries.

5.2 Learning outcomes

API Development: Gained experience in designing, implementing, and testing RESTful APIs, ensuring efficient data communication between the server and client.

Security Best Practices: Acquired knowledge of best practices for securing backend systems, including authentication, authorization, and data encryption techniques.

Debugging and Troubleshooting: Enhanced my ability to identify and resolve issues in backend code, improving the overall stability and performance of applications.

Version Control: Familiarized myself with version control systems like Git, enabling effective collaboration and code management within a team environment.

Team Collaboration: Developed strong communication and teamwork skills by working closely with front-end developers, project managers, and other stakeholders to deliver high-quality applications.

REFERENCES

Adam Hayes. (2024, June 23). *Capital Markets: What They Are and How They Work*. Retrieved from Investopedia: https://www.investopedia.com/terms/c/capitalmarkets.asp

Bhardwaj, A. (2024, June 24). *Introduction to SOLID principles*. Retrieved from What are the SOLID Principles in Java? Explained With Code Examples: https://www.freecodecamp.org/news/introduction-to-solid-principles/

Dan, A., & Bellen, S. (2021, October 7). *Blog: What Are Refresh Tokens and How to Use Them Securely*. Retrieved from Auth0: https://auth0.com/blog/refresh-tokens-what-are-they-and-when-to-use-them/

Driessen, V. (2010, January 5). *Posts: A successful Git branching model*. Retrieved from Nvie: https://nvie.com/posts/a-successful-git-branching-model/

IETF. (2024, June 3). *OAuth 2.0 Security Best Current Practice*. Retrieved from Web Authorization Protocol: https://datatracker.ietf.org/doc/html/draft-ietf-oauth-security-topics

IETF. (2025, March 05). *OAuth 2.0 for Browser-Based Applications*. Retrieved from Web Authorization Protocol: https://datatracker.ietf.org/doc/html/draft-ietf-oauth-browser-based-apps

Majaski, C. (2024, October 29). *Money Markets vs. Capital Markets: What's the Difference?* Retrieved from Investoperdia: https://www.investopedia.com/articles/investing/052313/financial-markets-capital-vs-money-markets.asp

OECD. (2023, 12 28). *Captial Markets*. Retrieved from OECD (Organisation for Economic Co-operation and Development): https://www.oecd.org/en/topics/capital-markets.html

Parecki, A. (2019, December 12). *It's Time for OAuth 2.1*. Retrieved from aaronparecki.com: https://aaronparecki.com/2019/12/12/21/its-time-for-oauth-2-dot-1

PortSwigger. (2025, February 29). *How to prevent CSRF vulnerabilities*. Retrieved from PortSwigger.net: https://portswigger.net/web-security/csrf/preventing

Shingala, A. (2023, August 4). *Spring Boot Caching with Redis*. Retrieved from Simform Engineering: https://medium.com/simform-engineering/spring-boot-caching-with-redis-1a36f719309f

Singh, P. (2022, April 17). *Best practices for one-time passwords*. Retrieved from MojoAuth - Go Passwordless: https://blog.mojoauth.com/best-practices-for-otp-authentication/