G H raisoni
COLLEGE
Engineering and
Management
Pune

G H RAISONI COLLEGE OF ENGINEERING AND MANAGEMENT

An Empowered Autonomous Institute Affiliated to Savitribai Phule Pune University (SPPU)

(NAAC Accredited with A+)

INTERNSHIP REVIEW – II

L&D INFOTECH PVT. LTD.
CONNECTIVITY WITH WORLD

G H raisoni
COLLEGE
Engineering and
Management
Pune

Department of CSE (Cyber Security)

Name of Intern: Mr. Suyash Sandip Awari

Registration No: 23DCCS1101075

Internal Mentor: Prof. Minakshi Sonwane

External Mentor: Mr. Umesh Lomte (umesh.lomte@Indinfotech.co.in)

Academic Year: 2025-2026

Company Introduction

Name of the Company	L&D Infotech
Address	Pune
Number of plants in India	1
Head Office	Pune
Number of Emolyees	35-50
Name of the CEO	Mrs. Manisha Lomte
Name of the Supervisor	Mr. Umesh Kale (umesh.lomte@Indinfotech.co.in)

Company Profile

L&D Infotech:

L&D Infotech Pvt. Ltd., located in Pune, is a private IT company offering software development services and industry-focused training programs. Established in 2019, the company provides solutions in web and mobile app development, hosting, and domain services, along with certification courses in Data Science, Big Data, SDN, and Java. It is known for hands-on training, live projects, and placement support, making it a popular choice among students and fresh graduates.

Department

Name of the Department	**CSE (Cyber Security)**
Software Used	**Apache NetBeans IDE**
Number of Employees	**35 - 50**
Name of supervisor	**Mr Umesh Lomte** (umesh.lomte@Indinfotech.co.in)
Domain	**Full Stack Java Developer**

My Contribution

JOINED ON 15TH JUNE 2025

- Company Orientation Program: Learned the company's culture, policies, and project management approach.
- Development Environment Setup: Installed and configured Java JDK, IDE, Maven, and Git.

TECH FOUNDATION

- Practical Java & Spring Boot: Applied Java skills to develop features using the Spring Boot framework.
- CRUD Operations from Database: Wrote the code to create, read, update, and delete user profiles in the database.
- Technical Documentation & Testing: Created unit tests for functions and wrote technical documents for the project.

Key Objectives for this Review Period

Weekly Summary: Weeks 8-9

(Service-Layer Testing & Refactoring)

Activities Completed

•Gained proficiency in JUnit 5 and Mockito frameworks for service-layer unit testing
•Mocked repository-layer dependencies using @Mock and @InjectMocks annotations to isolate business logic
•Authored comprehensive test suite for CourseService, validating both 'happy path' and exception handling scenarios
•Expanded test coverage to Module and VideoMetadata services, significantly increasing code coverage metrics

Tool Proficiency:

1.Testing Skills
Developed strong foundation in writing effective, isolated unit tests to validate business logic

2.Test-Driven Mindset
Solidified a test-driven approach essential for improving code maintainability

3.Technical Communication
Gained experience presenting technical progress and architectural improvements to supervisor.

Weekly Summary: Weeks 10-11

Activities Completed

•Designed and implemented the **Enrollment.java** JPA entity, leveraging **@ManyToOne** annotations to establish database relationships between User and Course entities.

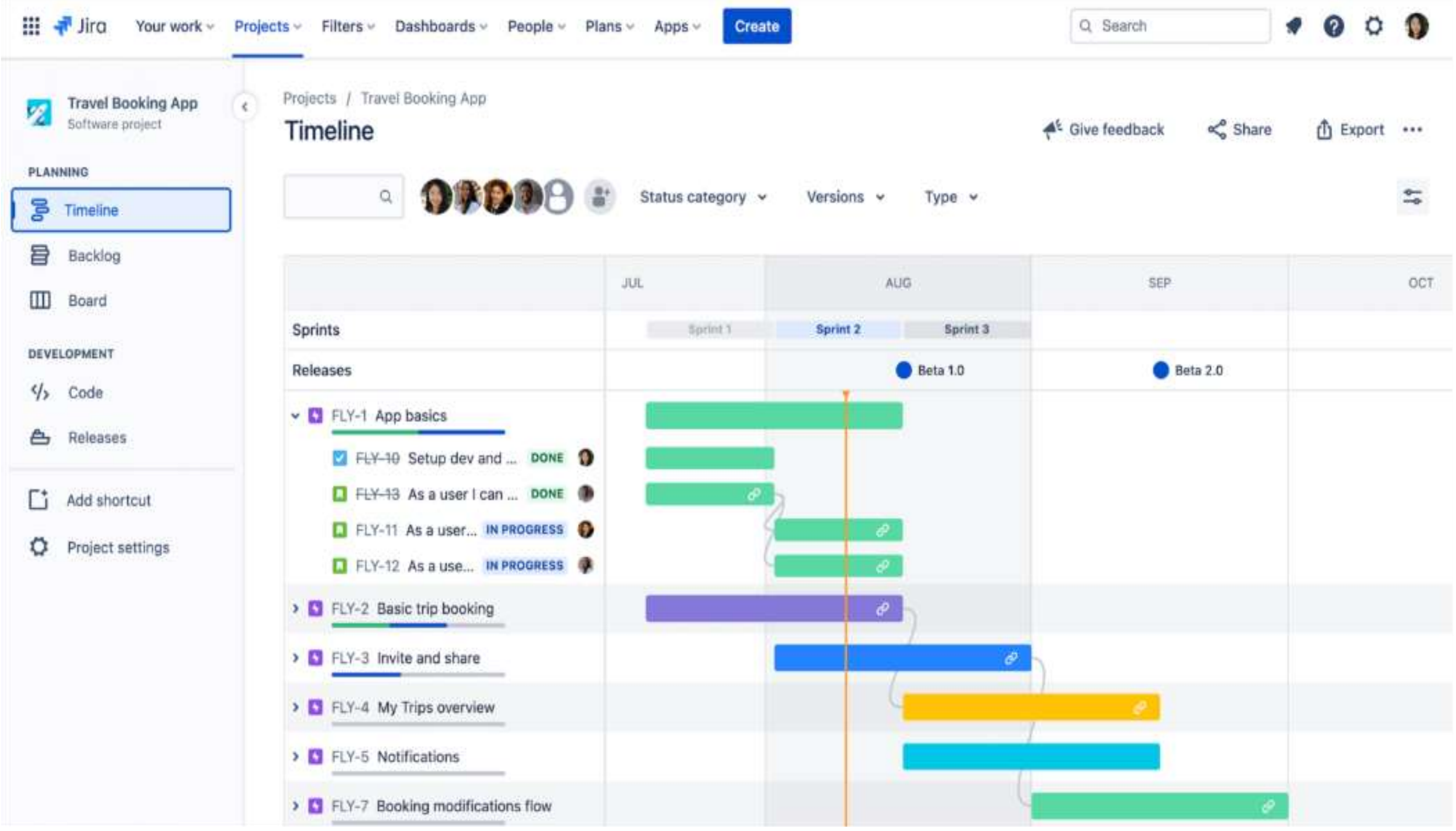
•Developed service-layer logic for enrollment operations with validation to prevent duplicate entries.

•Built and validated RESTful API endpoints (POST for creation, GET for retrieval) using Postman.

•Gained theoretical understanding of CI/CD principles and authored a multi-stage Dockerfile.

•Created a lean, portable Docker image to containerize the Spring Boot application artifact (JAR).

1.Training on jira software



FLY-1 App basics
~~FLY-10~~ Setup dev and ... DONE
~~FLY-13~~ As a user I can ... DONE
FLY-11 As a user... IN PROGRESS
FLY-12 As a use... IN PROGRESS
FLY-2 Basic trip booking
FLY-3 Invite and share
FLY-4 My Trips overview
FLY-5 Notifications
FLY-7 Booking modifications flow

Weekly Summary: Week 12

Activities Completed:

1.Bug Triage: Identified and reproduced a critical bug related to stale data reads from the User Profile API using Postman and IDE debugger.

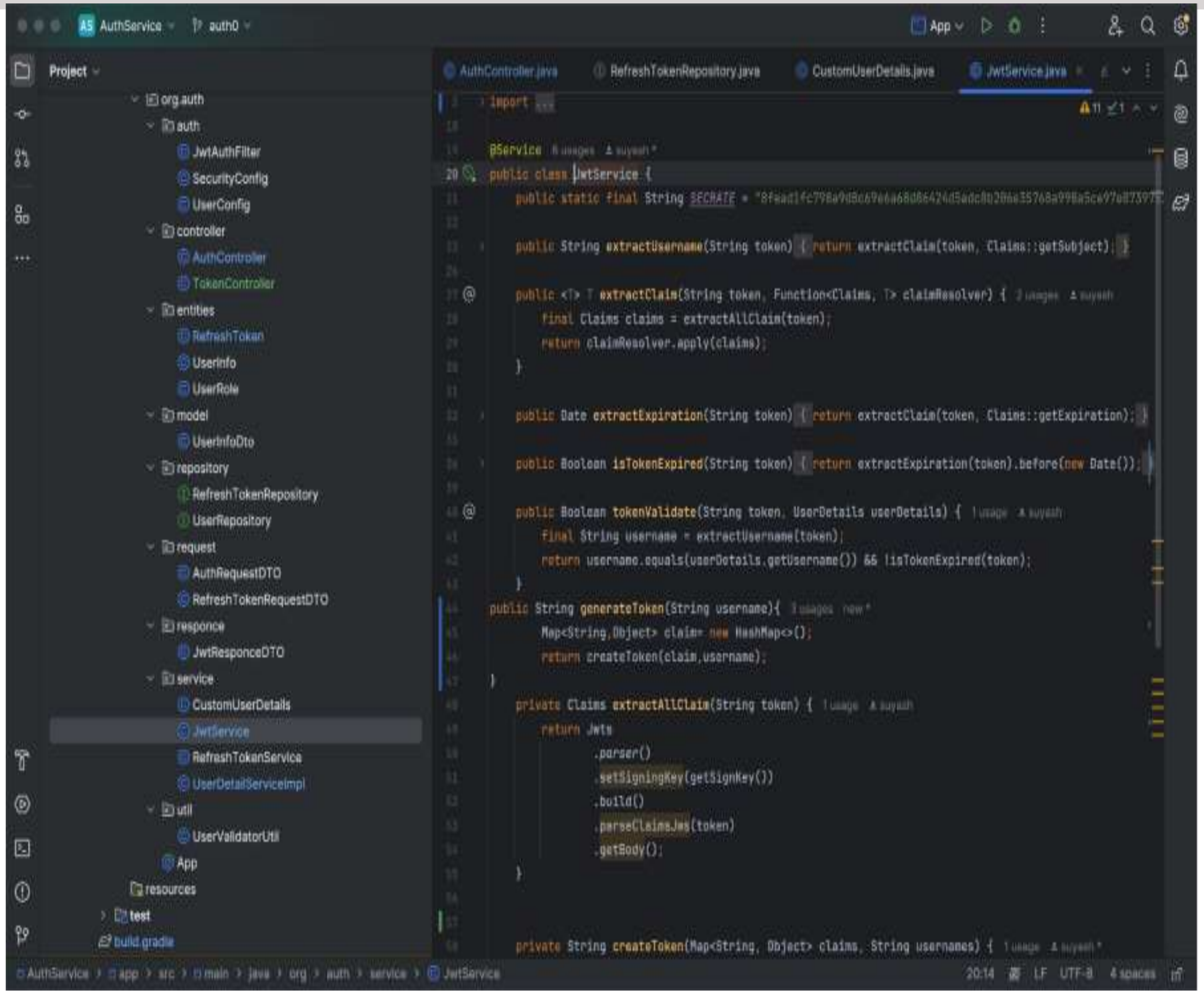
2.Root Cause Analysis: Traced the issue to Hibernate L1 cache failing to invalidate after consecutive update operations.

3Implementation: Leveraged @Transactional annotations and modified persistence logic to ensure cache synchronization.

Learning Outcomes:

- Mastered the end-to-end bug resolution process, from diagnosis to deploying a verified fix.
- Gained deep, practical understanding of persistence layer mechanics (Hibernate cache).
- Learned how caching mechanisms impact application performance and data integrity.
- Improved code quality awareness through static analysis tools.

Weekly Summary: Weeks 14



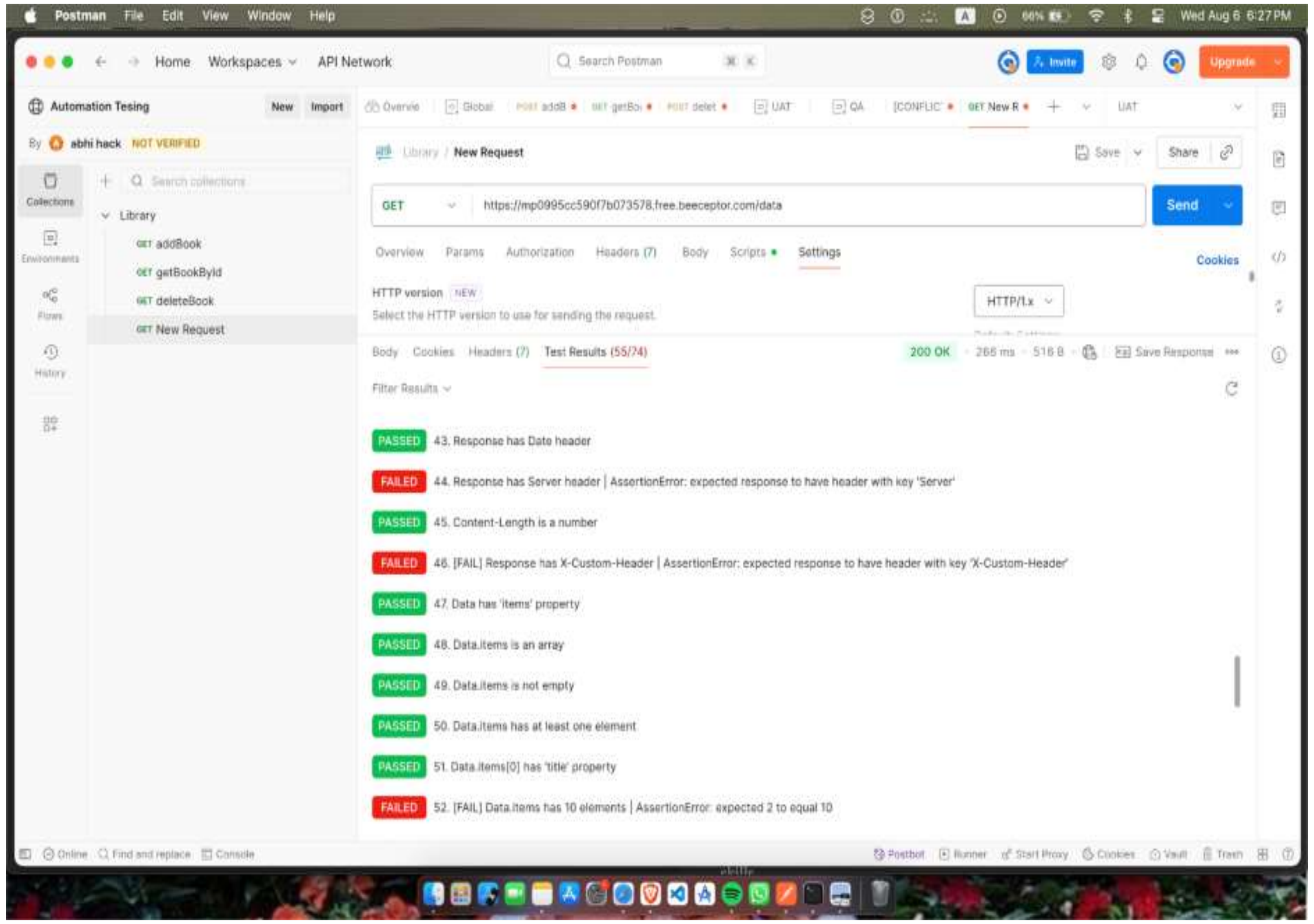
```java
import ...

@Service
public class JwtService {
    public static final String SECRATE = "8fead1fc798a9d8c69e6a68d86424d5adc8b20ea35768a998a5ce97e87397

    public String extractUsername(String token) { return extractClaim(token, Claims::getSubject); }

    public <T> T extractClaim(String token, Function<Claims, T> claimResolver) {
        final Claims claims = extractAllClaim(token);
        return claimResolver.apply(claims);
    }

    public Date extractExpiration(String token) { return extractClaim(token, Claims::getExpiration); }

    public Boolean isTokenExpired(String token) { return extractExpiration(token).before(new Date()); }

    public Boolean tokenValidate(String token, UserDetails userDetails) {
        final String username = extractUsername(token);
        return username.equals(userDetails.getUsername()) && !isTokenExpired(token);
    }
public String generateToken(String username){
        Map<String,Object> claim= new HashMap<>();
        return createToken(claim,username);
}

    private Claims extractAllClaim(String token) {
        return Jwts
                .parser()
                .setSigningKey(getSignKey())
                .build()
                .parseClaimsJws(token)
                .getBody();
    }


    private String createToken(Map<String, Object> claims, String usernames) {
```

Activities Completed:

- Architected a component-based UI using React, managing component state and lifecycle events with React Hooks (useState, useEffect).
- Utilized the axios library to perform asynchronous HTTP requests, consuming the course data from the backend's GET endpoints.
- Implemented a user-facing "Enroll" button with an onClick event handler that triggered a POST request to the backend enrollment API.

Area 3: Quality Assurance & Testing (API & SQL)

3.API TESTING ON THE GIVEN CONTRACT BY SENIOR


Postman File Edit View Window Help
Wed Aug 6 6:27 PM
Home Workspaces API Network
Search Postman
Invite
Upgrade
Automation Tesing
New Import
By abhi hack NOT VERIFIED
Collections
Environments
Flows
History
Search collections
Library
GET addBook
GET getBookById
GET deleteBook
GET New Request
Overvie Global POST addB GET getBo POST delet UAT QA [CONFLIC GET New R UAT
Library / New Request
Save Share
Send
Overview Params Authorization Headers (7) Body Scripts Settings
Cookies
Body Cookies Headers (7)
Save Response
Filter Results
Online Find and replace Console
Postbot Runner Start Proxy Cookies Vault Trash

GET https://mp0995cc590f7b073578.free.beeceptor.com/data

HTTP version NEW
Select the HTTP version to use for sending the request.
HTTP/1.x

Test Results (55/74) 200 OK · 266 ms · 516 B

```
PASSED 43. Response has Date header
FAILED 44. Response has Server header | AssertionError: expected response to have header with key 'Server'
PASSED 45. Content-Length is a number
FAILED 46. [FAIL] Response has X-Custom-Header | AssertionError: expected response to have header with key 'X-Custom-Header'
PASSED 47. Data has 'items' property
PASSED 48. Data.items is an array
PASSED 49. Data.items is not empty
PASSED 50. Data.items has at least one element
PASSED 51. Data.items[0] has 'title' property
FAILED 52. [FAIL] Data.items has 10 elements | AssertionError: expected 2 to equal 10
```

API Validation: Used Postman to test 20 different API endpoints, ensuring they functioned correctly and returned the expected data.

Database Verification: Wrote basic SQL queries to test the backend database, confirming data integrity during user registration and other operations.

Outcome: Gained hands-on experience in both API and database testing, key parts of the quality assurance process.

Detailed Tasks Overview:

1.Writing Advanced Unit Tests:

Implemented test suites with JUnit 5 and Mockito to verify service-layer functionality and exception handling.
Achieved 90% test coverage for backend services.

2.Building the Enrollment Feature:

Designed JPA entity relationships between Users and Courses, implemented validation logic, and created RESTful endpoints.
Successfully deployed 3 new APIs and 4 endpoints

3.Learning DevOps: Using Docker:

Created multi-stage Dockerfile for Spring Boot application and learned CI/CD fundamentals through the company's Jenkins pipeline.

THANK YOU !