# **Summary Report**:

## Project Overview

The goal of this phase was to prepare for the Final Year Project (FYP) by establishing a quality-driven backend API project. The focus was on creating a CRUD Web API with temporary in-memory data storage using Spring Boot.

## Tasks Completed (Task Allocation on Notion)

### Task 1: Integration of Static Analysis Tools

#### Choice of Linters to Add:

* + - * **ESLint** for **TypeScript**
      * **Pylint** or **Flake8** for **Python**
      * **Checkstyle** for **Java, (Chosen)**

#### SAST Scanner:

* + - * Integrated **Semgrep** for static analysis across the following languages
        + Java
        + Javascript
        + Python
        + Go
        + Ruby
        + C
        + CSharp
        + Objc
        + Kotlin
        + Swift
    - Custom Configurations:
      * Implemented at least 2 custom configurations for both linters and SAST
      * Resolved identified issues in the code flagged by the linters/SAST (Not done)

### Task 2: Web API Performance Testing Solution

#### Performance Testing Framework:

* + - * + Established a performance testing framework in the `tests/performance` folder in cloned Repo
        + Conducted basic web performance testing using Jmeter
        + Executed and documented performance tests for one of the CRUD operations, Read, Get Data to be exact.

### Task 3: Testable Non-Functional Requirements

#### Generic Requirements Documents:

* + - * + Developed generic testable non-functional requirements for a web API for the following documents provided in the repo;

Performance Requirements

Reliability Requirements

Security Requirements

Developer Guidelines

#### Compliance with ISO 25010:

* + - * + Ensured coverage of all sub-attributes of ISO 25010 for quality characteristics

#### Developer Guidelines:

* + - * + Explored various contributor guidelines from open-source projects
        + Provided an insightful explanation of the project structure based on the MVC framework

## Team Collaboration

* + Allocated tasks efficiently based on the team size
  + Ensured each team member contributed through individual commits
  + Maintained a focus on high-quality deliverables

## Challenges and Solutions

* + Overcame time constraints by prioritizing essential tasks
  + Collaborated effectively to achieve task objectives
  + Resolved issues identified by static analysis tools promptly

## Next Steps (Extra)

* + Review and refine the documentation and testing techniques for the FYP.
  + Conduct a thorough review of the codebase
  + Discuss for any additional features or improvements based on project requirements

## Conclusion

The completion of these tasks marks a significant milestone in preparing for the FYP project. The integration of static analysis tools, performance testing, and the establishment of non-functional requirements contribute to the foundation of a robust and quality-driven backend API, or at least, a head start on what to do.