**Code Review of The Software Project:**

**Online-Exam-System : Study Outline**

**Project By:**

Student ID: 180238

Student ID: 170238

**Reviewed By:**

Student ID: 210203

# **Introduction**

This code review evaluates the productivity application: **Study Outline**. The review identifies areas for improvement, adherence to best practices, and suggestions for enhancing maintainability, security, and overall code quality. In this code review, bad smells of the code, architecture evaluation, modularity check, condition statements of the code & other related sections are evaluated.

# **Code Smells**

1. **Large or complex methods:**

There are not many large or complex methods which can be difficult to read and understand. The average function size is close to 10 lines of code which is close to the standard size of functions.

1. **Long parameter lists:**

There is no method with long parameter lists.

1. **Excessive comments:**

In some modules there are excessive comments and some doesn’t have any comment at all.

Compose comments convergently for a block of code, not for line of code.

1. **Duplicate code:**

There is no duplicate code in the methods. Code reusability has been done quite effectively in most of the modules.

1. **Inconsistent naming conventions:**

In most cases, naming conventions are standardized.

1. **Incomplete error handling:**

Errors are mostly handled in each of the modules. There are mainly scope of running into error while doing file operations which seems to be handled carefully.

1. **Too many if/else statements:**

There are no significance of redundant usage of if/else statements.

1. **Poor use of inheritance:**

There is no poor use of inheritance in this project which is particularly causing any problem.

1. **Unnecessary dependencies:**

No signs of unnecessary library usage.

1. **Magic numbers or hard-coded values:**

In most of the cases, there is no sign of hard coding or magic numbers. Instead, contents are introduced and used in the project.

# **Proposed Architecture Evaluation**

Although there exists modularity in the project but doesn’t satisfy the MVC pattern. Usually, web applications follow Model-View-Controller Architecture, In the code base no signs of the architecture are found.  
The modular components should be rearranged by following proper file-folder structure.

# **Modularity Check**

The Project is modular, but need a little bit brush up in arranging the files in accordance with their roles, as Model, View, and Controller.

# **If/else Condition to Switch statement**

There doesn’t exist any heavy usage of if/else condition. So, the introduction of switch case statement is not required.