**Mini-Project: (15 hours)**

The students have to form a team of 4 members and choose a problem. They need to prepare UML Diagrams for the same and use Architecture and Design Principles and Patterns to develop a solution for the chosen problem. And implement the same using java /C++.

1. **Mini Project Evaluation Policy:**

1: Presentation of the project will be team-based.

2: Each student will be evaluated individually.

1. **Marks Distribution:**

|  |  |
| --- | --- |
| Analysis and Design Models | 3 marks |
| Use of MVC Architecture Pattern | 1 marks |
| Use of Design Principles and Patterns (at least 1each per team member) | 4 marks |
| Presentation/ Demo/ Explaining the code | 2 marks |
| **Total** | **10 marks** |

**Note:** \* - Any 4 Design patterns and any 4 design principles in total for the project for a 4 member team

1. **Complexity of the project:**
   * + A typical team size of 4 members would be required to implement at least 4 major features/use cases and 4 minor features/use cases (1 of each type per team member).
     + There should be equal participation of each student in a team.

1. **Important Aspects to be followed:**
   * + The implementation should be done using any JAVA/C++ technologies that must include an MVC framework (like Java Spring Framework,CROW Framework) along with optional Frontend frameworks. However, only Desktop and Web applications are accepted. **Mobile applications are not accepted.**
     + Each student should own a use case completely and implement it (that is, avoid situations where a team member will own only UI, another will own Backend etc.)
     + The implemented use cases should be merged into a single application. The data related to the project should be persisted in any database of your choice.

1. **Submission:** - Report document (PDF) containing
   * Title page with course details, project title, and team member details - PESU template to be used.
   * Problem statement (synopsis etc.)
   * Models (Use Case and Class Models)
   * Architecture Patterns, Design Principles and Design Patterns used along with a short description which elaborates how that is applied to your problem domain.
   * Github link to the Codebase (Repository should be public and accessible to all)
   * Individual contributions of the team members
   * Screenshots with input values populated and output shown (Use white background screens)