Miniproject Evaluation Rubrics

Slot No	Parameter	Excellent (8 - 10)	Good (6 - 7)	Fair (0 - 5)
1.	Problem Statement and Role of Technology	• Innovative ideas, skill of proposing various solutions, convincing the proposed solution with the clear concepts, statement is excellent and there is scope for future enhancement. • The project's use of technology helps students achieve learning objectives. And use of appropriate and creative courses studied/studying. It should address minimum of 4 subjects and justify the need for that subject.	• The concepts are clear. Able to frame the problem statement. • The project's use of technology is focused but does not take full advantage of the courses. Students use technology but do not learn to manipulate the technology to express ideas or concepts. Moderately address the relevance of subjects.	• Statement is not clear due to lack of concept building. • The project's use of technology treats students as passive recipients of information, is not well-defined, does not support student learning, or is a trivial or inappropriate use of the medium. No subject relevance.
2.	Requirement Collection	• Understand and applying the knowledge of software engineering course which helps in specifying the requirements precisely. • Identifying functional and non functional requirements. • Analyze how the requirements can be met and provide the supporting documents.	• Specifying the requirements. • Unable to differentiate the functional and non functional requirements. • No sufficient data to support the identified requirements.	• Few requirements are specified. • No proper understanding of the requirement. • No data to support the requirements.
3.	**Presentation Of SRS**	• **Level of understanding the SRS is clear.**	• **Level of understating the SRS is to some extent.**	• **Not understood the SRS.** • **Limited focus on the subject relevance.**

		• **Strong focus on the subject relevance.** • **Presentation was interactive and creative.** • **Was convincing, confident, professional and provides sufficient information about the SRS.** • **According to the given template, in time submission.**	• **Focus on the subject relevance.** • **Presentation was engaging.** • **According to the given template, in time submission.** • **Was convincing, professional and provides sufficient information about the SRS.**	• **Presentation was not engaging.** • **According to the given template.** • **Was convincing and but provides less information about the SRS**
4.	Working with High Level Design and Alternate solution	• Provide High level design which involves architectural design, use cases/DFD, class diagram, system model and block diagram by using the principles of software engineering course. • Provide alternate solution with high level and low level design. • Justify and convince the selected solution.	• Provide High level design which involves architectural design, class diagram, system model and block diagram by using the principles of software engineering course. • Provide alternate solution with less supporting design document. • Justify but unable to convince the alternate solution.	• Provide High level design which involves system model and block diagram. • Provide alternate solution but no clarity. • No justification of the alternate solution.
5.	Identifying the modules and test cases for each	• Complete mapping of the entire specified functional and non functional requirements to the modules. • Identify the two categories of test cases for each module to check the validity and non validity (exception handling) of each module. And proposing the solution for invalid inputs (exception handling).	• Complete mapping of the entire specified functional requirements to the modules. • Identify the two categories of test cases for each module to check the validity and non validity (exception handling) of each module.	• Moderately mapping of the specified functional requirements to the modules. • Unable to distinguish between the valid and invalid inputs.
6.	Working with low level design	• Provide Use case/ Data Flow and Class diagram for each module and also for whole	• Provide Use case/ Data Flow and Class diagram, Activity/ Sequence diagram, ER module.	• Improper representation of Use case/ Data Flow and Class diagram, Activity/ Sequence diagram, ER module.

		project, Activity/ Sequence diagram, ER module. • Data Structure and Algorithm selection for each module. • Represent the asymptotic notation for the selected algorithm. • Pseudo code for each module.	• Data Structure and Algorithm selection for each module. • Pseudo code for each module.	• Data Structure and Algorithm selection not appropriate. • Pseudo code for each module.
7.	**Presentation of Design**	• **Presentation was well aligned according to high level, low level and alternative designs proposed.** • **Presentation was interactive and creative.** • **Was convincing, confident, professional and provides sufficient information about the design.**	• **Presentation was mostly aligned according to high level, low level and alternative designs proposed.** • **Presentation was engaging.** • **Was convincing, professional and provides sufficient information about the design.**	• **Presentation was not aligned according to high level, low level and alternative designs proposed.** • **Presentation was engaging.** • **Was convincing but provides less information about the design.**
8.	Coding of the modules and debugging	• Code is readable and proper use of either top-down approach / bottom-up approach. • The program fulfils the requirement of the question and is correct. • Maintaining Error log file module wise.	• The program adheres to part of the question and is correct. • No proper documentation of Error Log files.	• The program does not solve the original problem or is incorrect.
9.	Coding and Integrating the modules	• The program is completely modular, more than one level of function calls. • Modules are integrated smoothly with minimal errors.	• The program is somewhat modular with several function calls. • Modules are integrated with errors.	• The whole program consists of the main module only.

10.	Testing	• Test-cases identified are clear and realistic. • Unit testing is applied for all the modules and integration test for the project.	• Test cases identified are clear, but may not be realistic. • Unit testing is applied for some modules and integration test for the project.	• Test cases are not clear. • Not used any proper testing methodologies.
11.	**Project Demo, Modifications and viva voce**	• **Provides interface which is neat and well organised.** • **Runs smoothly without glitches.** • **Program is according to the original question and gives correct output.** • **Presents whole project in an organised manner.** • **Skilfully address and handles questions from audience.**	• **Interface is reasonably neat.** • **Runs but with some glitches.** • **Program partially follows the original question or gives some incorrect output.** • **Presents about the program only and in an unclear manner.** • **Addressing and handling of questions from audience is satisfactory.**	• **Interface is untidy and not organised.** • **Still contains errors and warnings.** • **Program is completely incorrect.** • **Presentation is unclear or not understandable.** • **Lacked in answering and handling questions.**
12.	Project Report submission	• All the necessary information is included. • Report organised according to the given template and submitted within the deadline.	• Necessary information is included with some irrelevant information. • Report partially follows the given template and submission exceeds the given deadline.	• Important information is left out. • Not followed the template guidelines.