Faculty Kit

The faculty kit contains the evaluation strategy for the different milestones of the project and any other documents/links that may aid in the evaluation process (like sample quizzes on technologies etc)

Evaluation Strategy/Tips for the different milestones of the project

Objective

These guidelines provides a detailed evaluation strategy for each milestone of the project, along with tips and guidelines to ensure comprehensive assessment. It includes recommendations for testing and verification, as well as useful resources for evaluators.

Requirements Specification

1. Requirement Specification

Evaluation Points:

- Clarity: Is the requirement document clear, concise, and easy to understand?

- Assumptions: Are the assumptions valid and documented appropriately?

- Team Understanding: Do all team members demonstrate a unified understanding of the requirements?

- Presentation Quality: How well is the document or presentation structured and delivered?

Suggested Activities:

- Require the team to submit a formal document.

- Conduct a team presentation followed by a Q&A session to test understanding.

---

2. Technology Familiarization

Evaluation Points:

- Research Depth: Has the team explored the chosen technologies sufficiently (e.g., React, Node.js, MySQL)?

- Applicability: Are the selected technologies suitable for the project objectives?

- Team Competence: Do team members show adequate understanding of the technologies?

Suggested Activities:

- Team presentation on technologies.

- Conduct a short quiz on relevant topics (e.g., React fundamentals, database relationships).

---

3. Database Creation

Evaluation Points:

- Design Quality: Does the schema include appropriate keys, relationships, and constraints?

- Redundancy: Are there redundant fields or tables?

- Backup/Recovery: Has a backup and recovery strategy been considered?

- Storage Calculations: Are data storage requirements calculated and justified?

Suggested Activities:

- Review the submitted database schema document.

- Evaluate clarity of explanations through a short write-up or viva.

---

4. High-Level and Detailed Design

Evaluation Points:

- Requirement Coverage: Does the design address all requirements from the requirement specification?

- Clarity: Are flowcharts or pseudocode detailed enough to support implementation?

- Alternative Designs: Has the team explored alternative approaches?

- Error Handling: Are error messages and scenarios well-defined?

Suggested Activities:

- Require a design document submission.

- Conduct a viva or Q&A session with focus on design details.

---

5. Front-End Implementation

Evaluation Points:

- User Interface: Is the interface intuitive, visually appealing, and user-friendly?

- Help Features: Are instructions available and aligned with the design?

- Error Handling: Are error messages clear, non-technical, and helpful?

- Navigation: Is the layout logical and easy to navigate?

Suggested Activities:

- Evaluate the team's demo of the front-end.

- Provide feedback on user experience and interface design.

---

6. Integration with the Database

Evaluation Points:

- System Stability: Does the system function without crashing or errors?

- Feature Coverage: Are all specified features implemented and demonstrated effectively?

- Database Operations: Is the team able to create, update, and delete entries in real time?

Suggested Activities:

- Conduct a hands-on demo with live interactions.

- Test critical features like seat booking, login/logout, and database updates.

---

7. Test-Plan Review

Evaluation Points:

- Requirement Coverage: Are all functional requirements included in the test cases?

- Error and Exception Testing: Are edge cases and exceptions accounted for?

- Clarity: Are the test cases detailed and easy to execute?

Suggested Activities:

- Review the submitted test plan document.

- Ask for a live demonstration of select test cases.

---

8. Final Review

Evaluation Points:

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

- Completeness: Is the project complete with all features and functionalities as per the specification?

- Performance: Does the application perform efficiently under typical and edge-case scenarios?

- Documentation: Is the final report comprehensive, clear, and professional?

- Presentation: How well does the team articulate their work in the final demo?

Suggested Activities:

- Conduct a final demo for stakeholders and evaluators.

- Review all intermediate deliverables (presentations, design documents, test cases).

1. **React Official Documentation**
2. **Node.js Official Documentation**
3. **MySQL Tutorials and Quizzes**
4. **Testing Tools Documentation: Selenium, Cypress**