Semester 4 Report Layout – Cloud-Based IDE / Virtual Labs Project

# 1. Cover Page

Include: Title, Department, Your Name, Roll No., Guide’s Name, Institute Name, and Semester.

# 2. Declaration

# 3. Acknowledgements

# 4. Abstract

Brief summary of progress since last semester – focus on Module 2 (Tests) and Module 3 (Playground).

# 5. Table of Contents

Auto-generated once the content is filled.

# 6. List of Figures

Screenshots of updated UI, diagrams of test module and playground, etc.

# 7. List of Tables

Tech stack comparisons, module timelines, etc.

# 8. Introduction

- Recap of the problem and motivation

- Highlights of Semester 3 outcomes

- What will be achieved in Semester 4

- Structure of the report

# 9. Module 2 – Test Environment

- Objective

- Features:

- Controlled access during tests

- Test creation by faculty (UI)

- Timer, auto-save

- Real-time cheating prevention

- Screenshots: Test panel (student + admin)

- Backend flow

- Tech stack used

- Challenges faced

# 10. Module 3 – Playground Environment

- Objective

- Features:

- Free coding space

- AI-powered helper tools (planned/implemented)

- File save/load

- Docker execution logic

- AI tools (if added): auto-suggestions, explanations

- Screenshots

- Backend/API handling

- Advantages over existing playgrounds like Replit

# 11. Updated System Architecture

Diagram showing new modules and how they integrate with existing ones.

# 12. Implementation Status

- What’s fully implemented

- What’s in progress

- Gantt chart/timeline showing semester-wise planning

# 13. Tech Stack (Reused from Previous Semester)

- \*\*Frontend\*\*: Next.js, TypeScript, Tailwind CSS

- \*\*Backend\*\*: Node.js, Express.js

- \*\*Database\*\*: MongoDB with Atlas

- \*\*Authentication\*\*: Firebase

- \*\*Containerization\*\*: Docker

Reasons: Real-time updates, scalable design, support for language-specific containers, and secure environments.

# 14. Testing & Evaluation

- Test case coverage

- Example test flows

- Screenshots of grading interface

- Output from student side

# 15. Future Enhancements (Post Sem 4)

- Mobile interface

- Advanced AI integration

- Adaptive learning

- ML-based student performance tracking

# 16. References

IEEE format for documentation, research papers, and tools used.

# 17. Annexures

- Login screenshots

- Figma links

- Schema diagrams

- Roles and responsibilities of team members

# 18. Features Already Implemented (from Previous Report)

- Authentication via Firebase with role-based access

- Assignment creation and grading interface

- Code editor using React Monaco with test case integration

- Centralized file storage

- Role-based dashboards for students and faculty

- Real-time discussion panels and notifications

# 19. Faculty Feedback (Carried Forward)

- Emphasis on secure exam environments and anti-cheating tools

- Need for automated evaluation and random test cases

- Desire for scalable, cross-institutional compatibility

- AI recommendations for plagiarism, feedback, and adaptive learning