SOFTWARE REQUIREMENTS SPECIFICATION

for

Lab Evaluation System

Version 1.0

Prepared by:

Mehul Gupta (102316082)

Ishaan Bansal (102316064)

Tejas Chaudary (102316058)

Naaz Uppal (102316002)

Submitted to:

Dr. Komal Bharti

November 25, 2025

Contents

1 Project Selection Phase

1.1 Introduction

This document presents the official software development bid for the **StudyLMS: Lab Quiz & Exam Management System**. The proposal outlines the scope of work, deliverables, timeline, cost estimation, and responsibilities associated with the successful completion of the project.

1.2 Project Overview

The primary objective of the system is to digitalize and automate the following academic workflows:

- Lab exam scheduling and software-aware slot booking,
- Secure quiz/exam delivery with encryption and anti-cheating features,
- Automated evaluation of MCQ and coding-based assessments,
- Role-based access (Admin, Faculty, Student),
- Comprehensive dashboards for analytics and management.

The system is designed as a responsive, web-based application accessible across devices and integrated with institutional workflows.

1.3 Scope of Work

The software development scope includes the following modules:

1.3.1 1. Authentication and Access Control

- Secure login system for Admin, Faculty, and Students.
- Session management with role-based dashboards.
- Protection from unauthorized access and URL-level attacks.

1.3.2 2. Lab Scheduling and Booking System

- Real-time availability of labs.
- Software-aware booking (e.g., Python, Java, MATLAB).
- Prevention of double-booking and conflict detection.

1.3.3 3. Exam Creation and Management

- Faculty uploads exam PDFs (encrypted storage).
- Configuration of exam timing and permitted software.
- Watermarking and secure viewing mechanisms.

1.3.4 4. Student Exam Portal

- Timer-synced exam interface.
- Code editor with execution sandbox.
- MCQ and descriptive-style question support.

1.3.5 5. Auto-Grading & Result Generation

- Auto-grading of MCQs.
- Code execution with timeout and error handling.
- Exportable results for faculty as CSV/PDF.

1.3.6 6. Admin Dashboard & Monitoring

- System-wide monitoring of lab usage.
- Manage user accounts (Faculty/Student).
- View analytics and generate institutional reports.

1.4 Project Deliverables

The following deliverables will be submitted upon completion:

1. Fully functional web-based LMS application.
2. Source code repository (GitHub/Bitbucket).
3. Deployment documentation.

4. System requirement specification (SRS).
5. Test case documentation and testing reports.
6. User manual and training guide.
7. Final presentation and demonstration.

1.5 Development Timeline

Phase	**Tasks**	**Duration**
Requirement Analysis	SRS, architecture design	1 Week
UI/UX Design	Wireframes, prototypes	1 Week
Backend Development	API design, database setup	3 Weeks
Frontend Development	Dashboards, portals, exam UI	3 Weeks
Integration & Testing	System testing, bug fixes	2 Weeks
Deployment & Training	Hosting, documentation	1 Week
Total Estimated Duration		**11 Weeks**

1.6 Cost Estimate

The cost breakdown is calculated based on development hours, complexity, and integration needs.

Component	**Estimated Cost (INR)**
Frontend Development	35,000
Backend Development	45,000
Database Design & Deployment	20,000
Security + Encryption Layer	15,000
Testing & QA	10,000
Documentation & Training	5,000
Total Estimated Cost	**130,000 INR**

1.7 Assumptions & Exclusions

- Hosting costs and domain fees are excluded.
- Any integration with third-party systems (e.g., University ERP) is out of scope.
- Future enhancements beyond current requirements will incur additional cost.

1.8 Conclusion

This software bid outlines the development plan, cost, and timeline for delivering a robust and secure **Lab Quiz & Exam Management System**. The proposed solution meets the functional and technical requirements while ensuring scalability, reliability, and ease of use for Admins, Faculty, and Students.

2 Project Overview

2.1 Introduction

The **Lab Quiz & Exam Management System** is designed to digitize, streamline, and secure the management of laboratory examinations, quizzes, and scheduling within an academic environment. Traditional lab examinations suffer from scheduling conflicts, manual coordination, software mismatch, and security vulnerabilities during exam delivery. This system aims to provide a unified, secure, and automated platform enabling faculty, students, and administrators to efficiently conduct and monitor lab assessments.

The platform supports secure exam uploads, software-aware lab slot booking, automated grading of coding questions, and a student-safe exam environment with anti-cheating measures. Through role-based access, each stakeholder is provided with a personalized dashboard suited to their responsibilities and workflow.

2.2 Objectives

- To automate lab scheduling based on software and system availability.
- To ensure secure delivery of exam materials with encryption and restricted access.
- To provide a timed, reliable, and user-friendly environment for students to attempt quizzes and coding-based exams.
- To assist faculty in creating, managing, and auto-grading assessments.
- To provide administrators with control over user management, analytics, and system-wide monitoring.

2.3 System Users

The system supports three main user groups:

- **Admin:** Manages users, monitors system usage, and oversees institution-level operations.
- **Faculty:** Creates exams, uploads files, configures timing, views results, and monitors student submissions.
- **Students:** Books lab slots, attempts exams, views results, and interacts with the exam interface.

2.4 Project Scope

2.4.1 In-Scope

The following features and modules are included within the scope of this project:

- **Authentication & Role-Based Access**
  - Admin, Faculty, and Student login.
  - Session management and access restriction.
- **Lab Scheduling & Booking System**
  - Lab availability listing.
  - Software-specific filtering (e.g., Python, Java, MATLAB).
  - Conflict detection and double-booking prevention.
- **Exam Creation & Upload**
  - Upload encrypted exam PDFs.
  - Configure timing, allowed applications, and exam duration.
  - Watermarking and secure view restrictions.
- **Student Exam Interface**
  - Timer-synced exam environment.
  - MCQ, short-answer, and coding questions.
  - Code editor with sandbox execution and timeout limits.
- **Auto-Grading System**
  - Automatic evaluation of MCQs.
  - Code execution and result matching.
  - Error handling (SyntaxError, RuntimeError).
- **Result Generation & Download**
  - CSV/PDF export of results.
  - Faculty dashboard for result management.
- **Security & Anti-Cheating Features**
  - Disable right-click, copy-paste, and inspect access.
  - Encrypted exam files and limited visibility until start time.
  - Session timeout and auto-logout.
- **Admin Dashboard**
  - User management (Faculty/Student accounts).

  - System usage analytics.
  - Monitoring ongoing exams.

2.4.2 Out-of-Scope

The following components are **not included** in the scope of this project:

- **Integration with External University ERP Systems**
  - No automatic syncing with attendance, internal marks, or other academic modules.
- **Biometric-Based Authentication**
  - No fingerprint, retina scan, or RFID-based login.
- **AI-Based Cheating Detection**
  - No real-time webcam monitoring or face recognition.
- **Offline Desktop Application**
  - System does not support offline usage; requires internet connection.
- **Physical Lab Hardware Monitoring**
  - Does not track lab PC hardware health, OS-level logs, or system performance metrics.
- **Automated Question Paper Generation**
  - Questions must be manually uploaded by faculty; no AI-based generation.
- **Payment or Fee Management Modules**
  - No integration with financial systems or fee departments.

2.5 Conclusion

The project scope clearly defines the boundaries of the **Lab Quiz & Exam Management System**, ensuring that all stakeholders understand the system's capabilities and limitations. The in-scope features provide a complete, secure, and efficient digital examination workflow, while the out-of-scope items ensure clarity and prevent feature creep during development. This structured scope enables timely and high-quality delivery of the final system.

3 Analysis Phase

3.1 Use-Case Diagrams

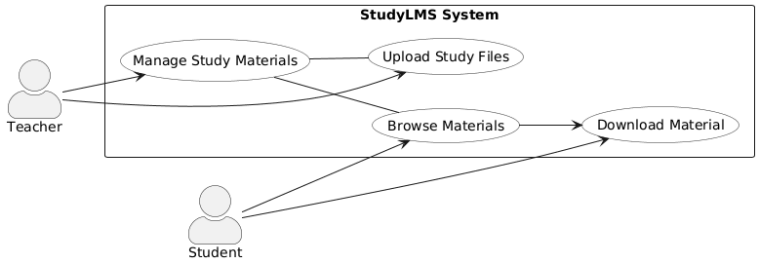


Figure 3.1: Teacher uploads → Students download

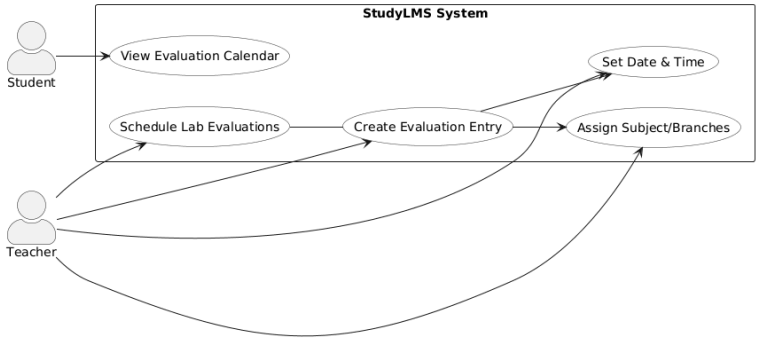


Figure 3.2: Teacher schedules → Students view

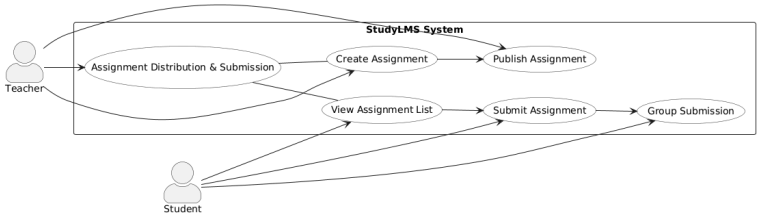
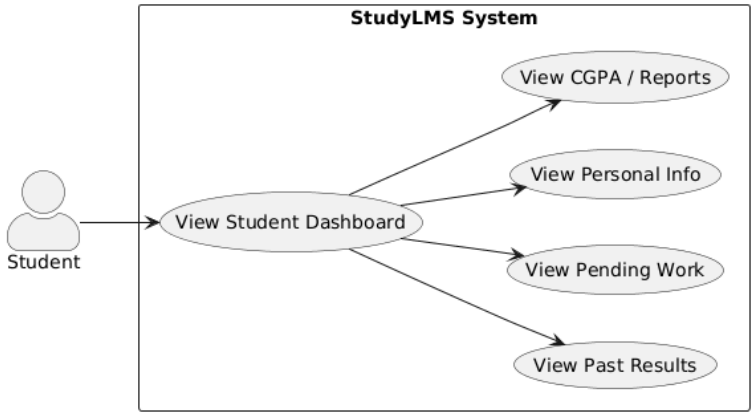
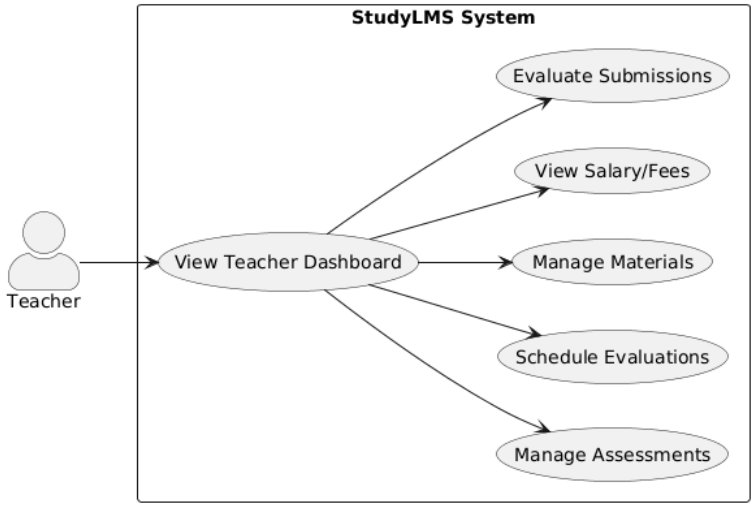


Figure 3.3: Teacher assigns → Students submit → Group submission





3.2 Use Case Templates

3.2.1 Use Case 1: User Registration and Authentication

Actors: Student, Teacher
Description: Users register using their official `@thapar.edu` email. Students provide roll number and branch, while teachers provide employee ID and department. The system validates the domain and creates a role-bound profile. Upon successful login, the user is redirected to their respective dashboard.
Trigger: User accesses the signup/login page.
Preconditions: User must possess a valid `@thapar.edu` email ID.
Postconditions: User account is created or authenticated and corresponding dashboard is displayed.

3.2.2 Use Case 2: Student Dashboard Access

Actor: Student
Description: Upon login, the student views upcoming lab evaluations, pending assignments, past results, and study materials. The dashboard displays CGPA, completed tasks, and exam countdowns.
Trigger: Successful student login.
Postconditions: Student-specific widgets and navigation are rendered.

3.2.3 Use Case 3: Teacher Dashboard Access

Actor: Teacher
Description: The teacher views tools for scheduling evaluations, creating assessments, uploading study materials, managing submissions, and viewing salary details.
Trigger: Successful teacher login.
Postconditions: Teacher-specific dashboard and navigation are displayed.

3.2.4 Use Case 4: Manage Study Materials

Actors: Teacher (primary), Student (consumer)
Description: Teachers upload PDFs, PPTs, and notes for subjects. Students browse subjects and download materials relevant to examinations.
Trigger: Teacher selects "Upload Material".
Preconditions: Teacher must be mapped to the subject.
Postconditions: Updated study material list accessible to students.

3.2.5 Use Case 5: Schedule Lab Evaluations

Actor: Teacher
Description: The teacher schedules lab evaluations by selecting date, time, subject, and applicable branches. Students see this schedule in the calendar.
Trigger: Teacher selects "Create Evaluation".
Preconditions: Teacher must be logged in.
Postconditions: Evaluation appears in student calendar.

3.2.6 Use Case 6: View Evaluation Calendar

Actor: Student
Description: Students view a calendar or timeline of upcoming evaluations with details such as subject, timing, venue, and instructions.
Trigger: Student opens the Lab Evaluation Calendar page.
Postconditions: Students gain awareness of upcoming assessments.

3.2.7 Use Case 7: Attempt Online Assessment

Actor: Student
Description: Students select an assessment, verify identity, enter a passcode, enable camera and microphone, and attempt the test. Assessments can be MCQ-based or coding-based. Timers, autosubmit mechanisms, and question navigation are provided.
Trigger: Student clicks "Start Assessment".
Preconditions: Valid assessment ID, passcode, and active test window.
Postconditions: Student responses are recorded and submitted.

3.2.8 Use Case 8: Create Assessments

Actor: Teacher
Description: Teachers create new assessments by specifying duration, passcode, evaluation window, and adding MCQ or coding questions.
Trigger: Teacher selects "Create Assessment".
Postconditions: Assessment becomes available at scheduled time.

3.2.9 Use Case 9: Assignment Distribution and Submission

Actors: Teacher (assigner), Student (submitter)
Description: Teachers publish assignments with deadlines and instructions. Students view tasks, upload submissions, and form groups if required.
Trigger: Teacher creates assignment.
Postconditions: Assignment appears in student view with submission tracking.

3.2.10 Use Case 10: Group Assignment Collaboration

Actor: Student
Description: Students form groups for specific assignments by selecting members from the allowed student list and submit work jointly.
Trigger: Student initiates group submission.
Preconditions: Assignment must permit group submissions.
Postconditions: Group submission metadata stored.

3.2.11 Use Case 11: Evaluation of Assessments

Actor: Teacher
Description: MCQ assessments are auto-evaluated, whereas coding assessments require manual review by teachers. The teacher assigns marks and publishes results.
Trigger: Teacher selects "Evaluate Submissions".
Postconditions: Final marks stored and reflected in student dashboards.

3.2.12 Use Case 12: Result Publishing and Viewing

Actors: Teacher (publisher), Student (viewer)
Description: Teachers publish assessment and assignment results. Students view detailed score breakdowns and semester-wise reports.
Trigger: Teacher updates results.
Postconditions: Student dashboard is updated with latest results.

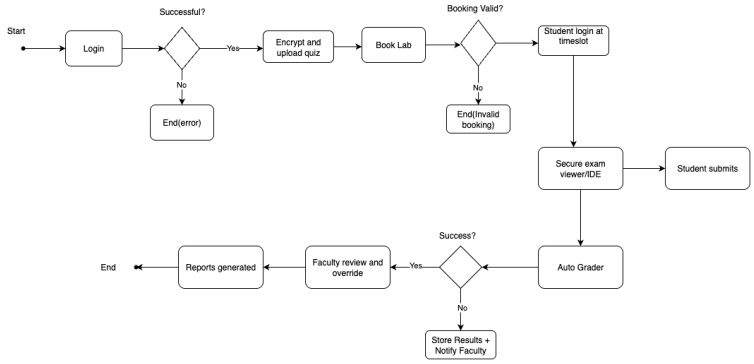
3.2.13 Use Case 13: Salary and Fee Record Viewing

Actor: Teacher
Description: Teachers view monthly salary breakdown, deductions, and related financial information.
Trigger: Teacher opens the Salary Dashboard.
Postconditions: Salary details are displayed.

3.2.14 Use Case 14: Logout and Session Handling

Actors: Student, Teacher
Description: Users log out, clearing local session information. The system restricts access to protected pages after logout.
Trigger: User clicks the Logout button.
Postconditions: User is returned to the login page and session is terminated.

3.3 Activity Diagram and Swimlane Diagrams



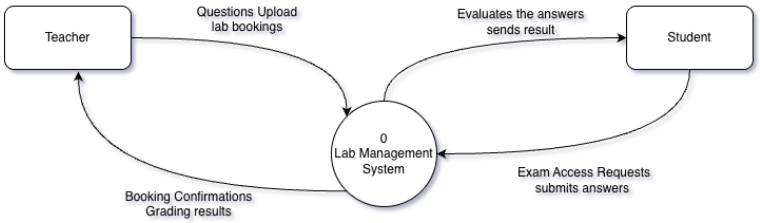
3.4 Data Flow Diagrams (DFDs)

The LES will be developed in multiple phases to ensure structured and testable releases:

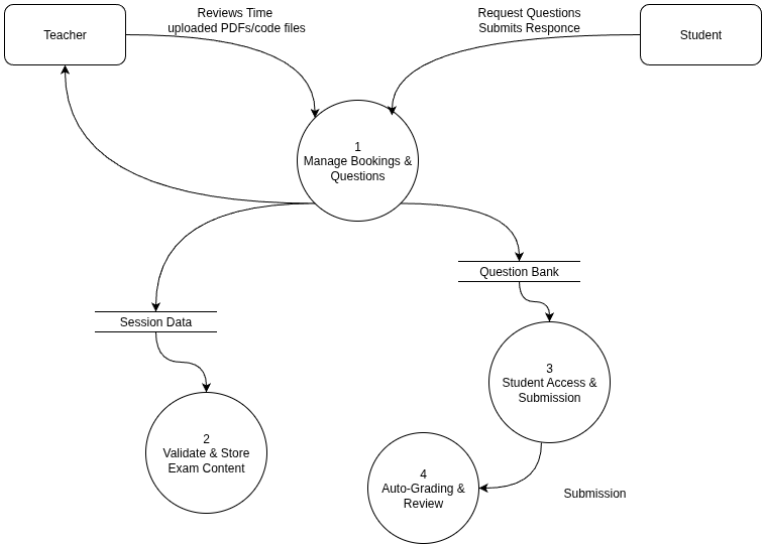
- **Phase 1 (MVP):** Authentication, lab scheduling, PDF upload and question extraction, basic grading, and result dashboard.

- **Phase 2:** Encrypted question delivery, sandboxed code execution, and advanced audit logging.
- **Phase 3:** Performance analytics, plagiarism detection, and detailed reporting modules.

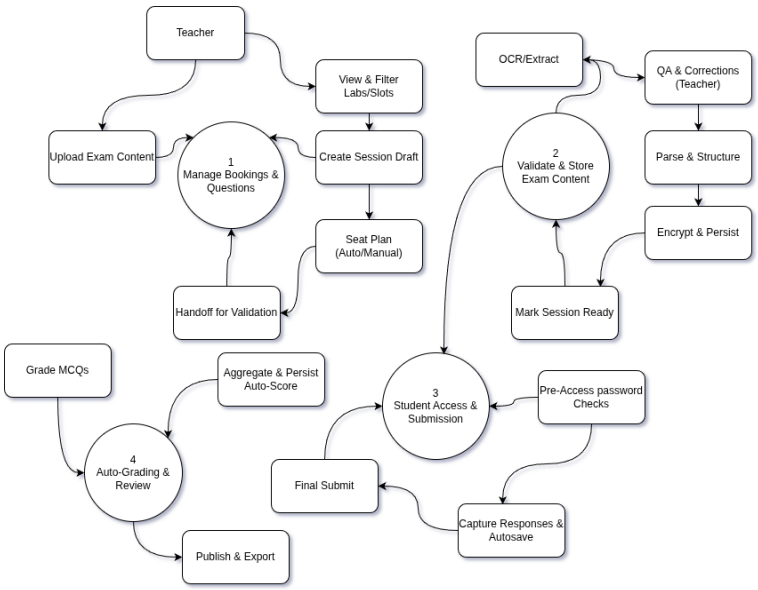
3.4.1 DFD Level 0



3.4.2 DFD Level 1



3.4.3 DFD Level 2



3.5 Software Requirement Specification in IEEE Format

https://drive.google.com/file/d/1RRqPvMq6iUfRgjsZ7CzMJNWDs60jXXAH/view?usp=sharing

3.6 User Stories and Story Cards

3.6.1 Section 1: User Stories

US-001: Report an Issue (Citizen)

Acceptance Criteria:

- App allows uploading images
- Location tagging available
- Issue type selection
- Add description
- Confirmation after submission

US-002: Automatic Tagging of Authorities (System)

Acceptance Criteria:

- System tags authority based on geo-location
- Notification sent to relevant department

US-003: Notify Authorities of New Issues

Acceptance Criteria:

- Notifications include issue type, description, location, images
- Authorities can open full details

US-004: City Dashboard for Issue Tracking

Acceptance Criteria:

- Dashboard lists all issues with status
- Filter by type, status, priority
- Mark issue as resolved
- Show images and descriptions

US-005: View Report History (Citizen)

Acceptance Criteria:

- Display history of reported issues
- Filter by open/resolved
- View details (images, status)

3.6.2 Section 2: Story Cards

Story Card: US-001

Title: Report an Issue (Citizen)
User Story: As a citizen, I want to report an issue with geo-location and images, so that I can help local authorities identify and fix the problem.

Acceptance Criteria:

- Upload images
- Location tagging
- Select issue type (pothole, broken light, garbage)
- Add description
- Receive confirmation message upon submission

Priority: High
Estimate: 5 story points (4 hours)
Notes/Comments: Ensure intuitive UX for uploading images and marking location.

Story Card: US-002

Title: Automatic Tagging of Authorities (System)
User Story: As a system, I want to automatically tag relevant municipal authorities based on issue type and location, so that the correct department is notified.

Acceptance Criteria:

- Tag correct authority based on geo-location
- Send notification to the tagged department

Priority: High
Estimate: 8 story points (6 hours)
Notes/Comments: Integrate map APIs and consider geo-fencing for accuracy.

Story Card: US-003

Title: Notify Authorities of New Issues
User Story: As a municipal authority, I want to receive notifications when new issues are reported, so that I can take prompt action.

Acceptance Criteria:

- Notification includes type, description, geo-location, images
- Authority can view issue details from notification

Priority: High
Estimate: 4 story points (3 hours)
Notes/Comments: Ensure reliable notification system (email or push notifications).

Story Card: US-004

Title: City Dashboard for Issue Tracking
User Story: As a city official, I want to view a dashboard to track the status and resolution of reported issues.

Acceptance Criteria:

- Dashboard shows all issues with status
- Filter by issue type, status, and priority
- Officials can mark issues as resolved
- Display issue details with images and status

Priority: Medium
Estimate: 13 story points (10 hours)
Notes/Comments: Dashboard must be responsive and user-friendly.

Story Card: US-005

Title: View Report History (Citizen)
User Story: As a citizen, I want to view my report history and track the status of my issues, so that I can follow up on unresolved issues.

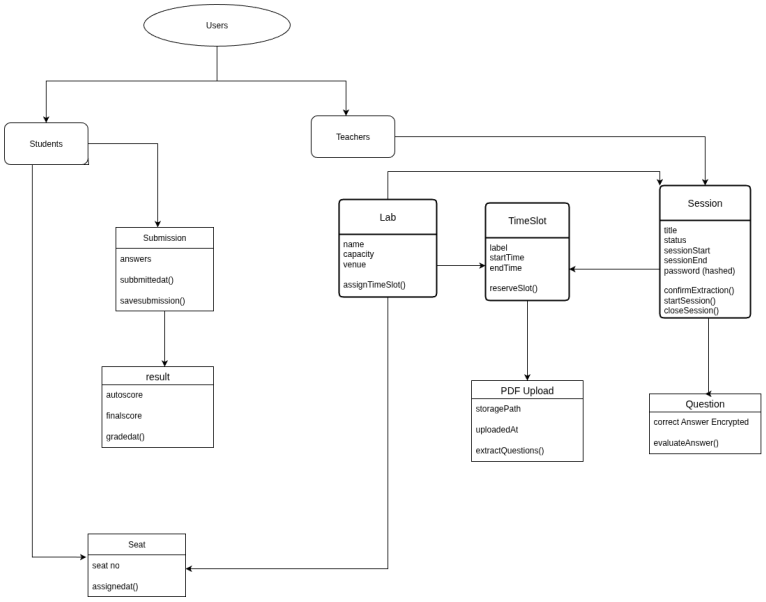
Acceptance Criteria:

- Show history of reported issues
- Filter by open/resolved status
- View issue details (images, description, status)

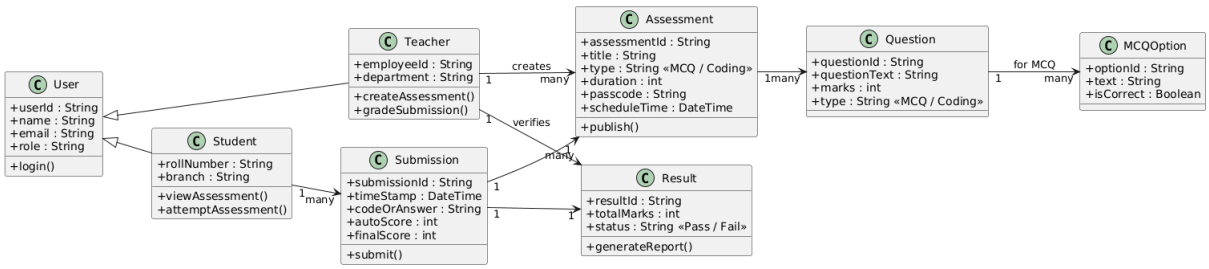
Priority: Medium
Estimate: 5 story points (4 hours)
Notes/Comments: Ensure easy access and filtering by status.

4 Design Phase

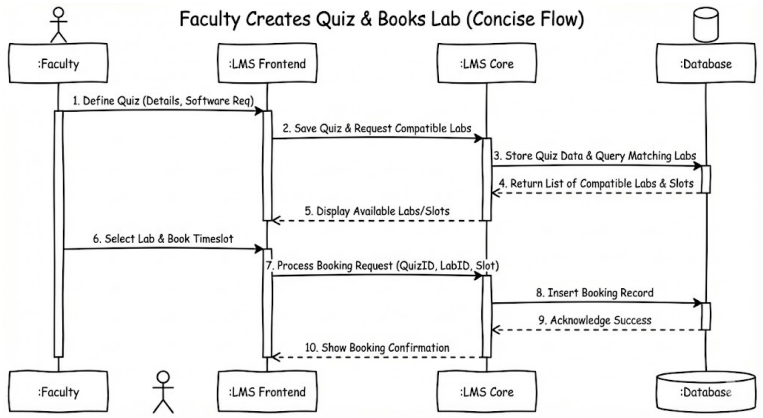
4.1 Class Diagram



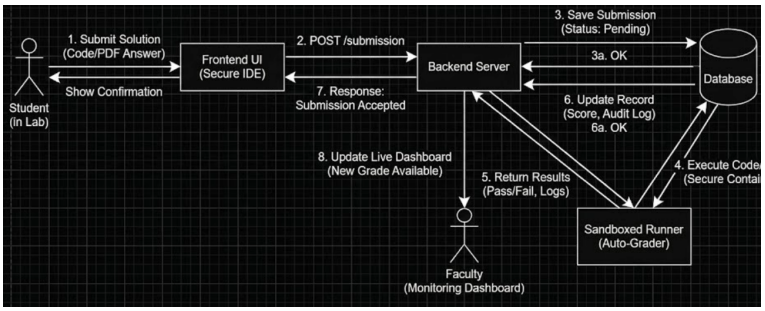
Class Diagram - Assessment Management Module



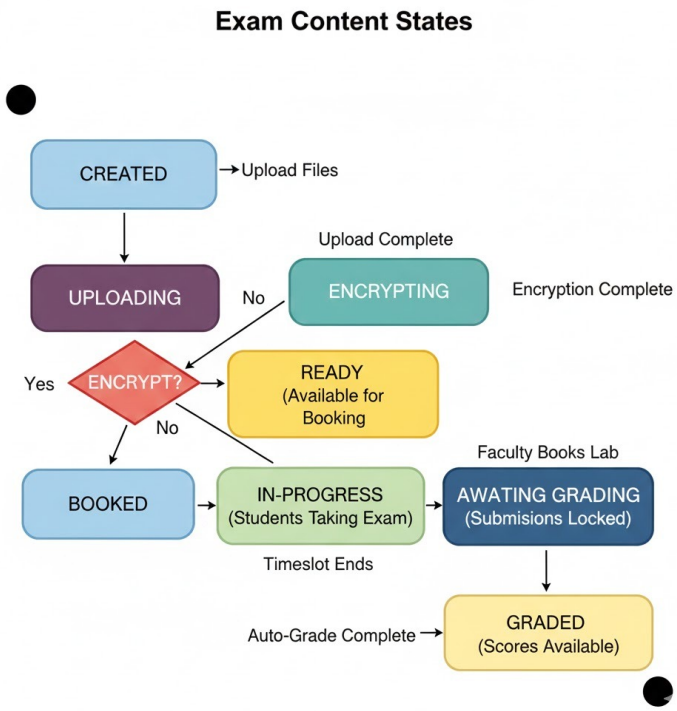
4.2 Sequence Diagram

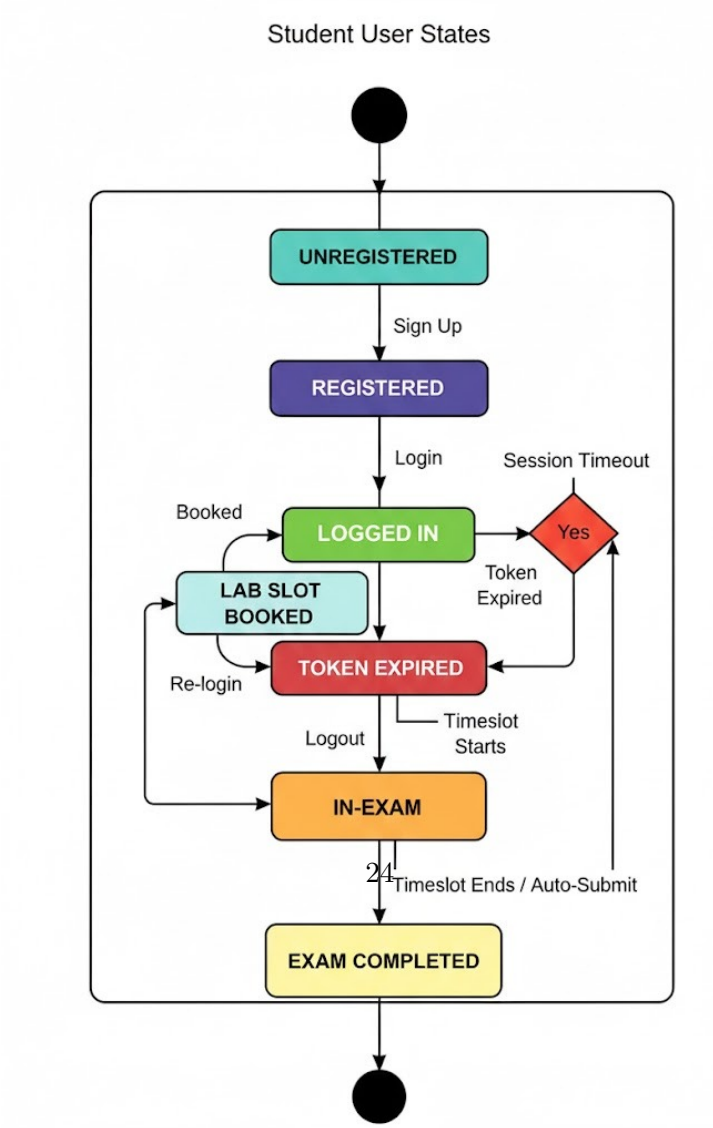


4.3 Collaboration Diagram



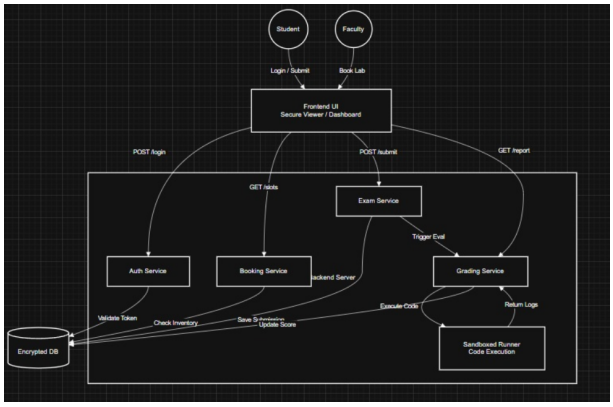
4.4 State Chart Diagrams



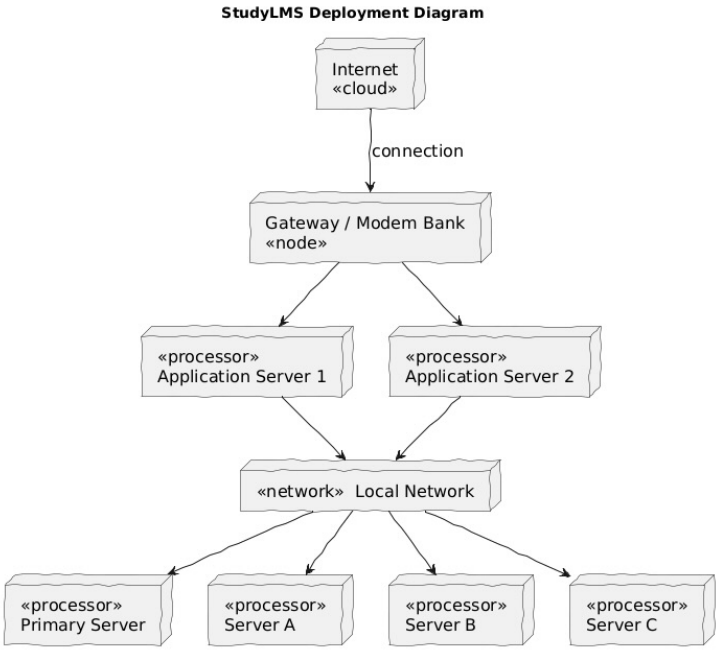


5 Implementation

5.1 Component Diagrams

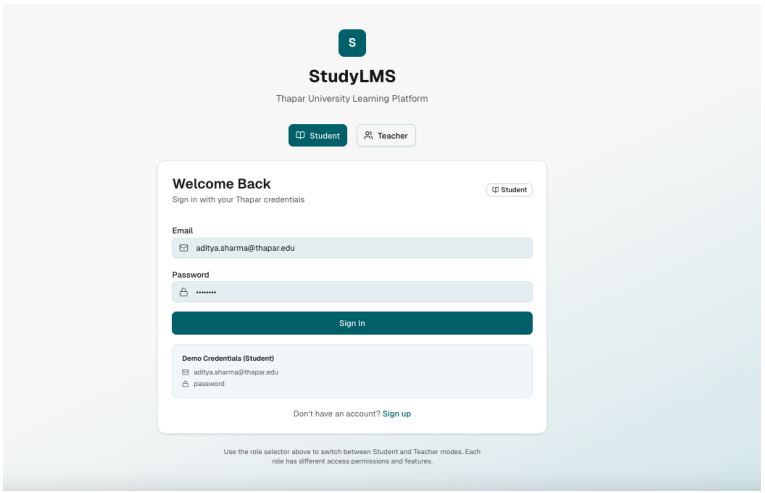


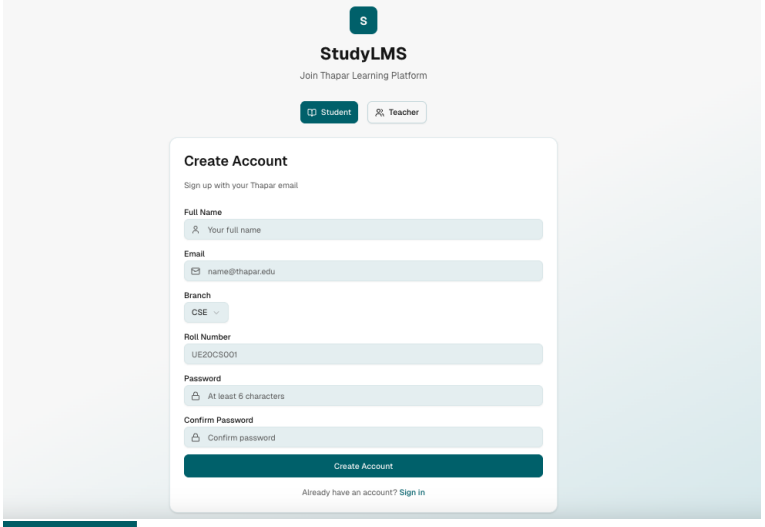
5.2 Deployment Diagrams

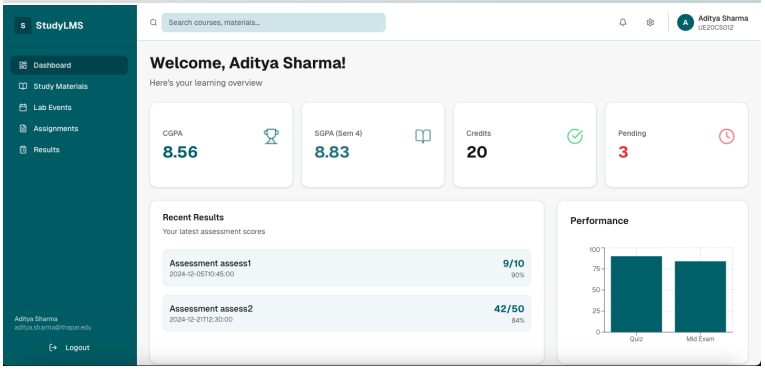


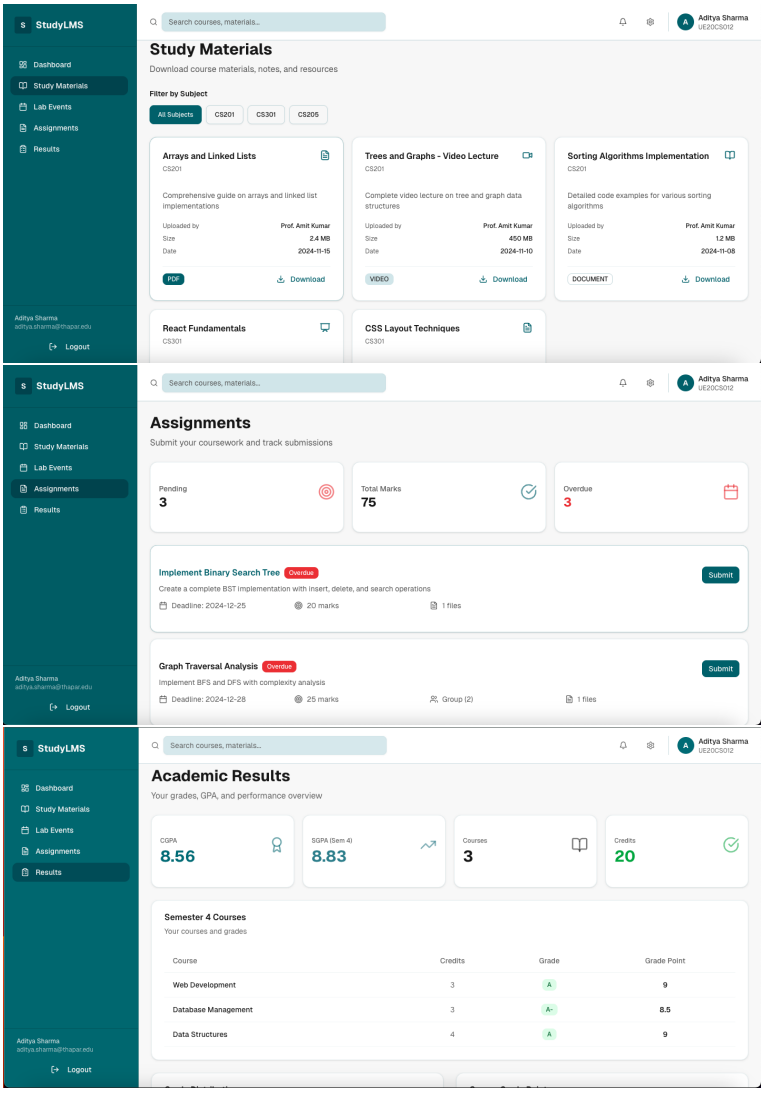
5.3 ScreenShots

Student

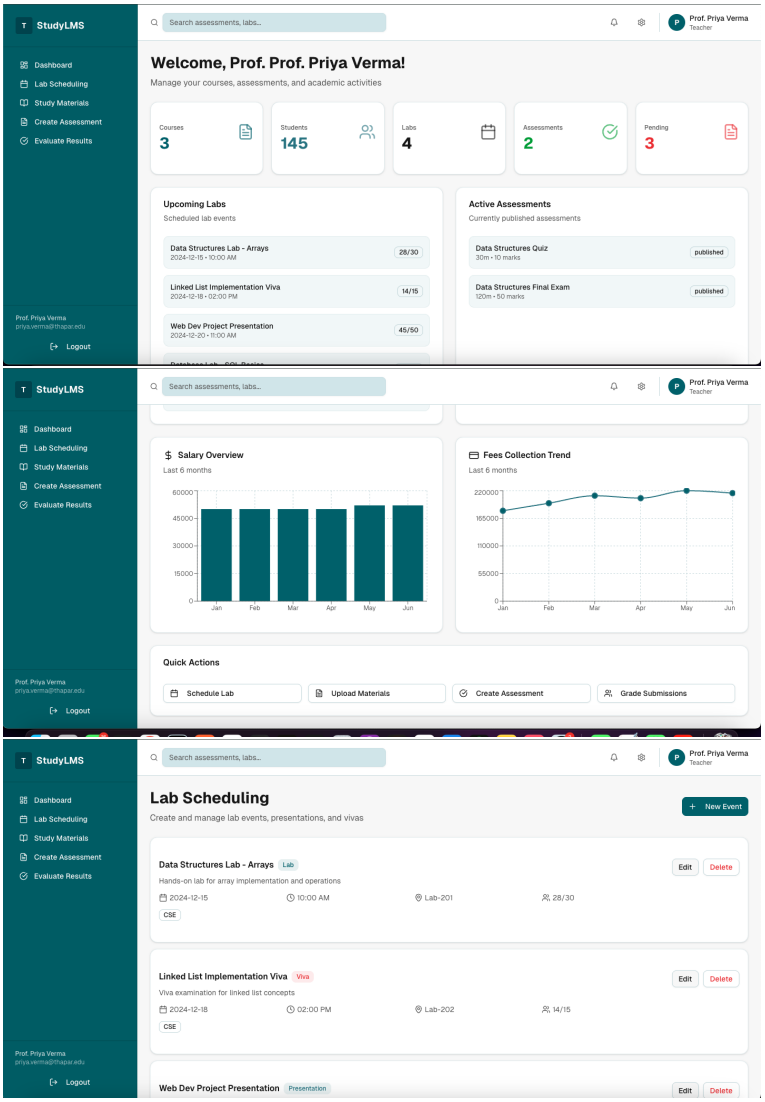


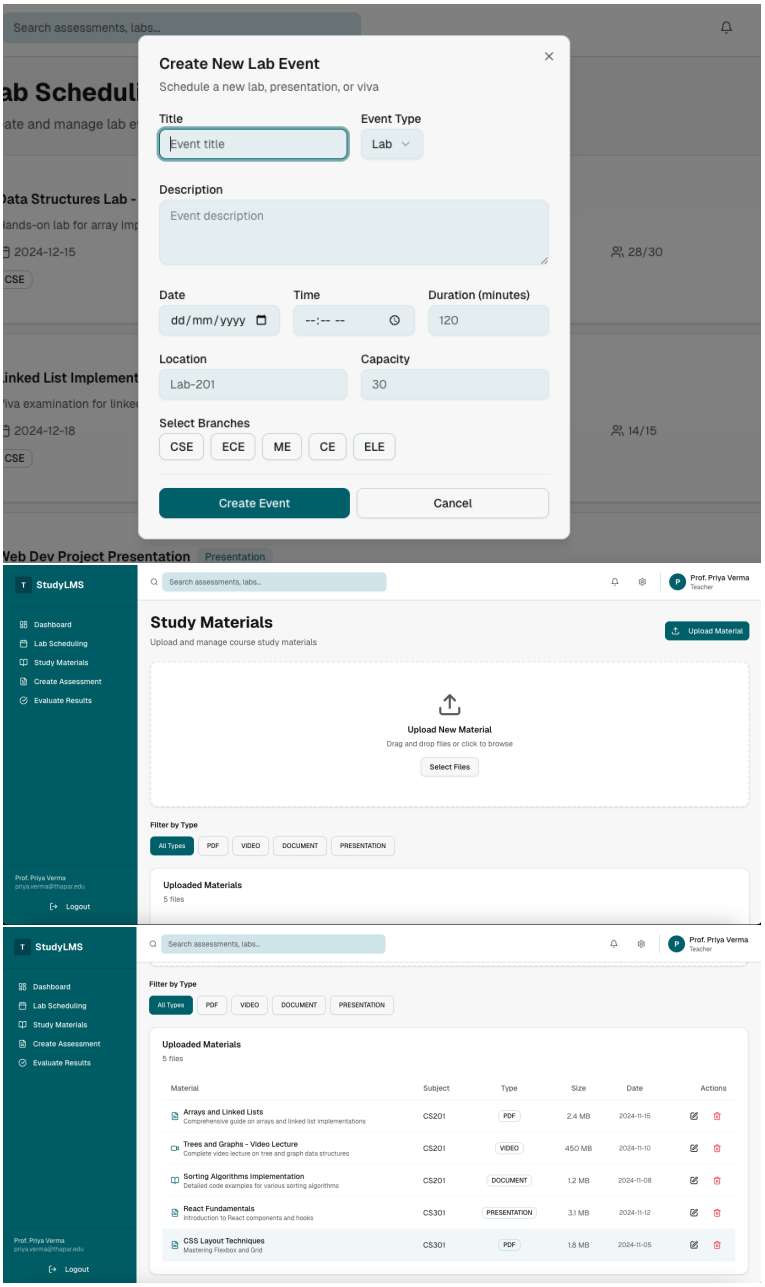


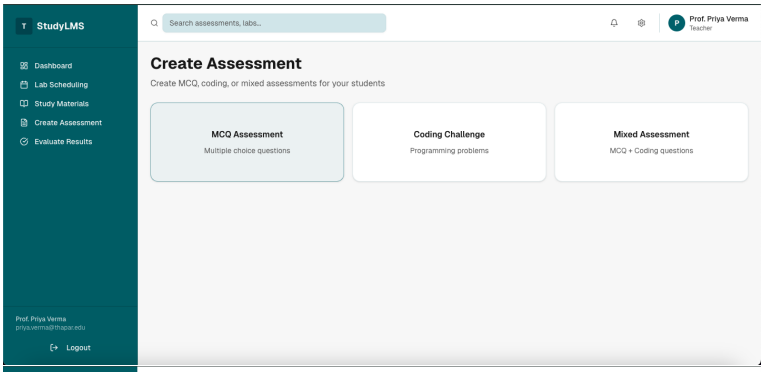


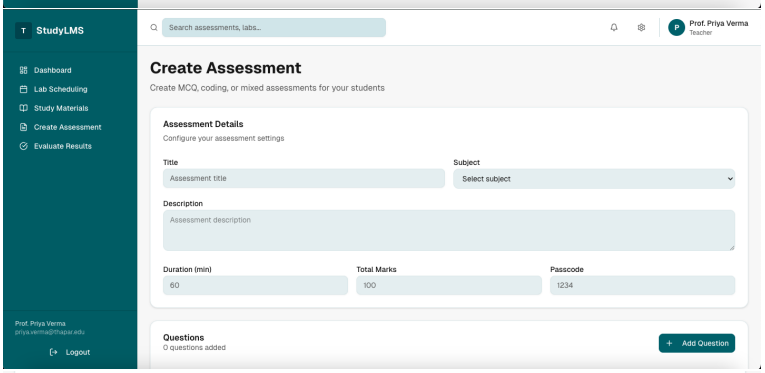

StudyLMS
Search courses, materials...
Aditya Sharma
UE20CS012
Dashboard
Study Materials
Lab Events
Assignments
Results
Study Materials
Download course materials, notes, and resources
Filter by Subject
All Subjects
CS201
CS301
CS205
Arrays and Linked Lists
CS201
Comprehensive guide on arrays and linked list implementations
Uploaded by
Prof. Amit Kumar
Size
2.4 MB
Date
2024-11-15
PDF
Download
Trees and Graphs - Video Lecture
CS201
Complete video lecture on tree and graph data structures
Uploaded by
Prof. Amit Kumar
Size
450 MB
Date
2024-11-10
VIDEO
Download
Sorting Algorithms Implementation
CS201
Detailed code examples for various sorting algorithms
Uploaded by
Prof. Amit Kumar
Size
1.2 MB
Date
2024-11-08
DOCUMENT
Download
React Fundamentals
CS301
CSS Layout Techniques
CS301
Aditya Sharma
aditya.sharma@thapar.edu
Logout
StudyLMS
Search courses, materials...
Aditya Sharma
UE20CS012
Dashboard
Study Materials
Lab Events
Assignments
Results
Assignments
Submit your coursework and track submissions
Pending
3
Total Marks
75
Overdue
3
Implement Binary Search Tree
Overdue
Create a complete BST implementation with insert, delete, and search operations
Deadline: 2024-12-25
20 marks
1 files
Submit
Graph Traversal Analysis
Overdue
Implement BFS and DFS with complexity analysis
Deadline: 2024-12-28
25 marks
Group (2)
1 files
Submit
Aditya Sharma
aditya.sharma@thapar.edu
Logout
StudyLMS
Search courses, materials...
Aditya Sharma
UE20CS012
Dashboard
Study Materials
Lab Events
Assignments
Results
Academic Results
Your grades, GPA, and performance overview
CGPA
8.56
SGPA (Sem 4)
8.83
Courses
3
Credits
20
Semester 4 Courses
Your courses and grades
Course
Credits
Grade
Grade Point
Web Development
3
A
9
Database Management
3
A-
8.5
Data Structures
4
A
9
Aditya Sharma
aditya.sharma@thapar.edu
Logout

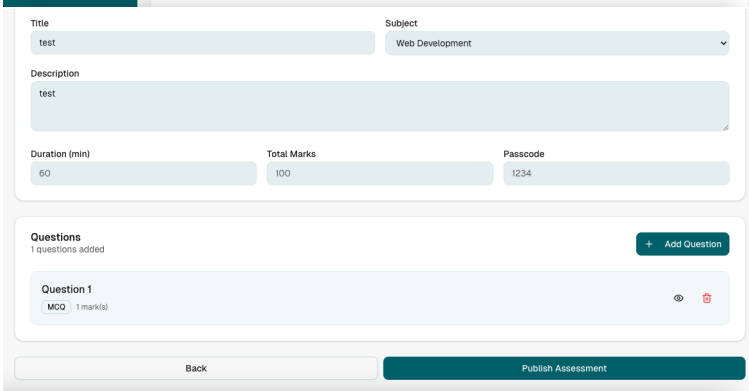
Teacher

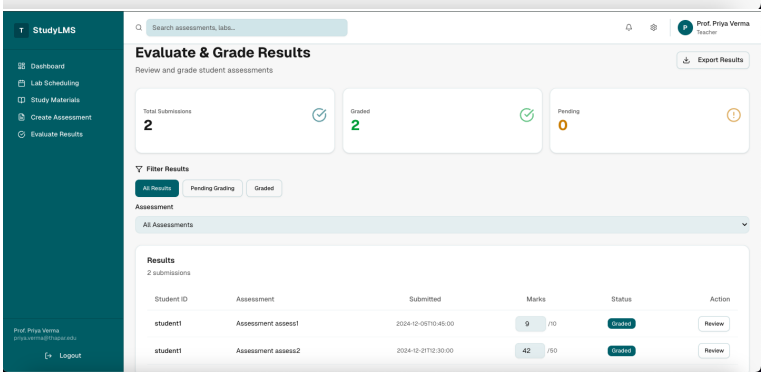

StudyLMS
Search assessments, labs...
Prof. Priya Verma
Teacher
Dashboard
Lab Scheduling
Study Materials
Create Assessment
Evaluate Results
Welcome, Prof. Prof. Priya Verma!
Manage your courses, assessments, and academic activities
Courses
3
Students
145
Labs
4
Assessments
2
Pending
3
Upcoming Labs
Scheduled lab events
Data Structures Lab - Arrays
2024-12-15 • 10:00 AM
28/30
Linked List Implementation Viva
2024-12-18 • 02:00 PM
14/15
Web Dev Project Presentation
2024-12-20 • 11:00 AM
45/50
Active Assessments
Currently published assessments
Data Structures Quiz
30m • 10 marks
published
Data Structures Final Exam
120m • 50 marks
published
Prof. Priya Verma
priya.verma@thapar.edu
Logout
Salary Overview
Last 6 months
60000
45000
30000
15000
0
Jan
Feb
Mar
Apr
May
Jun
Fees Collection Trend
Last 6 months
220000
165000
110000
55000
0
Quick Actions
Schedule Lab
Upload Materials
Create Assessment
Grade Submissions
Lab Scheduling
Create and manage lab events, presentations, and vivas
New Event
Data Structures Lab - Arrays
Lab
Hands-on lab for array implementation and operations
2024-12-15
10:00 AM
Lab-201
28/30
CSE
Edit
Delete
Linked List Implementation Viva
Viva
Viva examination for linked list concepts
2024-12-18
02:00 PM
Lab-202
14/15
CSE
Edit
Delete
Web Dev Project Presentation
Presentation
Edit
Delete


Search assessments, labs...
Create New Lab Event
Schedule a new lab, presentation, or viva
Title
Event title
Event Type
Lab
Description
Event description
Date
dd/mm/yyyy
Time
--:-- --
Duration (minutes)
120
Location
Lab-201
Capacity
30
Select Branches
CSE
ECE
ME
CE
ELE
Create Event
Cancel
Web Dev Project Presentation
Presentation
StudyLMS
Dashboard
Lab Scheduling
Study Materials
Create Assessment
Evaluate Results
Prof. Priya Verma
priya.verma@thapar.edu
Logout
Study Materials
Upload and manage course study materials
Upload Material
Upload New Material
Drag and drop files or click to browse
Select Files
Filter by Type
All Types
PDF
VIDEO
DOCUMENT
PRESENTATION
Uploaded Materials
5 files
Material
Subject
Type
Size
Date
Actions
Arrays and Linked Lists
Comprehensive guide on arrays and linked list implementations
CS201
PDF
2.4 MB
2024-11-15
Trees and Graphs - Video Lecture
Complete video lecture on tree and graph data structures
CS201
VIDEO
450 MB
2024-11-10
Sorting Algorithms Implementation
Detailed code examples for various sorting algorithms
CS201
DOCUMENT
1.2 MB
2024-11-08
React Fundamentals
Introduction to React components and hooks
CS301
PRESENTATION
3.1 MB
2024-11-12
CSS Layout Techniques
Mastering Flexbox and Grid
CS301
PDF
1.8 MB
2024-11-05


StudyLMS
Search assessments, labs...
Prof. Priya Verma
Teacher
Dashboard
Lab Scheduling
Study Materials
Create Assessment
Evaluate Results
Create Assessment
Create MCQ, coding, or mixed assessments for your students
MCQ Assessment
Multiple choice questions
Coding Challenge
Programming problems
Mixed Assessment
MCQ + Coding questions
Prof. Priya Verma
priya.verma@thapar.edu
Logout


StudyLMS
Search assessments, labs...
Prof. Priya Verma
Teacher
Dashboard
Lab Scheduling
Study Materials
Create Assessment
Evaluate Results
Create Assessment
Create MCQ, coding, or mixed assessments for your students
Assessment Details
Configure your assessment settings
Title
Assessment title
Subject
Select subject
Description
Assessment description
Duration (min)
60
Total Marks
100
Passcode
1234
Questions
0 questions added
Add Question
Prof. Priya Verma
priya.verma@thapar.edu
Logout


Title
test
Subject
Web Development
Description
test
Duration (min)
60
Total Marks
100
Passcode
1234
Questions
1 questions added
Add Question
Question 1
MCQ
1 mark(s)
Back
Publish Assessment


StudyLMS
Search assessments, labs...
Prof. Priya Verma
Teacher
Dashboard
Lab Scheduling
Study Materials
Create Assessment
Evaluate Results
Evaluate & Grade Results
Review and grade student assessments
Export Results
Total Submissions
2
Graded
2
Pending
0
Filter Results
All Results
Pending Grading
Graded
Assessment
All Assessments
Results
2 submissions
Student ID
Assessment
Submitted
Marks
Status
Action
student1
Assessment assess1
2024-12-05T10:45:00
9
/10
Graded
Review
student1
Assessment assess2
2024-12-21T12:30:00
42
/50
Graded
Review
Prof. Priya Verma
priya.verma@thapar.edu
Logout

6 Testing

6.1 Test Plan

6.1.1 Introduction

The objective of the testing phase was to ensure that the **Lab Quiz & Exam Management System** functions correctly under realistic lab conditions. The system was tested primarily across three core pillars:

- **Security** — Encryption, watermarking, restricted access.
- **Scheduling** — Software-aware lab booking and conflict prevention.
- **Evaluation** — Auto-grading accuracy for coding and MCQ exams.

6.1.2 Testing Strategy

A hybrid testing approach was adopted:

1. Manual Testing (Black Box)

- Used to validate User Interface (UI) flows such as booking a lab or attempting a quiz.
- Performed on Google Chrome and Mozilla Firefox for cross-browser compatibility.

2. Unit Testing (White Box)

- Basic assertions were implemented in the backend (Node.js/Python).
- Ensured accurate scoring and correct text-matching behavior during auto-grading.

3. Security Sanity Checks

- Manually attempted unauthorized access (e.g., Student accessing Faculty Dashboard).
- Verified exam content remains hidden from browser developer tools before exam start.

6.1.3 Test Environment

- **Hardware:** University Lab PCs (Windows 10, 8GB RAM) and personal laptops.
- **Network:** University Wi-Fi (restricted) and home Wi-Fi.
- **Browsers Tested:** Chrome v120+, Microsoft Edge (latest).

6.2 Test Cases

Table 1: Authentication & Access Control

Test ID	Description	Input Data	Expected Outcome	Status
TC-01	Admin Login	Valid Admin Username/Password	Redirect to Admin Dashboard	Pass
TC-02	Invalid Login Attempt	Wrong Password	Error message: "Invalid Credentials"	Pass
TC-03	Role Protection	Student accessing Faculty URL	Access Denied (403 Forbidden)	Pass
TC-04	Session Expiry	Idle for 15 minutes	Auto-logout and redirect to Login page	Pass

Table 2: Booking & Exam Management

Test ID	Description	Input	Expected Outcome	Status
TC-05	Software Filter	Select "Python 3.8"	Only labs supporting Python appear	Pass
TC-06	Double Booking	Book an existing reserved slot	System shows: "Slot Unavailable"	Pass
TC-07	Upload Exam File	Upload PDF	File gets encrypted and stored securely	Pass

Table 3: Student Execution & Grading

6.3 Test Reports by Peers

6.3.1 Peer Testing Methodology

A beta testing session was conducted with five peers from the Computer Science department.

- **Testers:** 5 students from CSE.

Test ID	Description	Input	Expected Outcome	Status
TC-08	Exam Timer	Start Exam	Timer counts down and auto-submits at 00:00	Pass
TC-09	Copy Protection	Right-click / Ctrl+C	Context menu disabled; copying blocked	Pass
TC-10	Code Execution	Valid Python Code	Output "Hello World" displayed	Pass
TC-11	Syntax Error Handling	Incorrect code	Displays "SyntaxError" log	Pass
TC-12	Infinite Loop Handling	`while(True): pass`	Runner terminates after 5 seconds	Pass

- **Duration:** 1-hour testing session.
- **Tasks Assigned:** Create a quiz, book a lab, attempt the quiz as a student.
- **Goal:** Identify bugs and UI/UX issues.

6.3.2 Summary of Feedback and Bugs Found

Tester	Role Tested	Observation / Bug	Severity	Action Taken
Peer A	Faculty	"Upload PDF" button too small	Low (UX)	Increased button size and added icon
Peer B	Student	Timer reset after page refresh	High	Timer now synced with server timestamp
Peer C	Student	Code editor font too small	Medium	Added font-size toggle
Peer D	Student	Infinite loop froze browser	High	Reduced sandbox timeout to 3 seconds
Peer E	Admin	CSV export feature works correctly	N/A	Feature validated

6.3.3 Conclusion

Peer testing identified two high-severity issues (Exam Timer Reset, UI Freeze) which were fixed before final submission. Additional UI improvements were also implemented based on medium/low priority feedback. Overall, the system is stable and ready for demonstration.