 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering & Technology Department of Information and Communication Technology	
Subject: CP (01CT0715)	Aim: Testing and Validation	
	Date:	Enrollment No: 92200133036

Testing and Validation

1. Testing Methodology

To make sure the *Acadexa Student App* works as intended, I followed a structured testing approach. Since the app was built using **Flutter** for the front-end, **PHP** for the backend APIs, and **MySQL** for the database, I divided testing into three levels:

- **Unit testing** for checking individual modules like login, attendance, and chat.
- **Integration testing** for verifying smooth communication between the app, APIs, and database.
- **Performance testing** for measuring speed, responsiveness, and stability under load.

For this, I used Flutter's built-in flutter test package, **Postman** for API checks, and **JMeter** for performance evaluation. In addition, I also did manual black-box testing to simulate how a student would actually use the app.

2. Unit Tests

Unit tests were carried out on the most critical modules. A few representative cases are listed below:


Test Case	Component	Input	Expected Output	Actual Result	Status
UT-01	Login API	Valid email & password	Success token returned	Token generated	Pass
UT-02	Attendance Module	Student ID = 102	Attendance record fetched	Correct record displayed	Pass
UT-03	Fees Record	Student ID = 105	Outstanding fees shown	Correct balance displayed	Pass
UT-04	Assignment Upload	PDF file < 10MB	File uploaded & stored	Upload successful	Pass
UT-05	Chat Module	Message "Hello"	Message stored & displayed in batch chat	Message visible instantly	Pass

3. Integration Tests

Integration testing was important because the app depends heavily on backend APIs and database queries. A few key scenarios were tested:

Test Case	Scenario	Expected Result	Actual Result	Status
IT-01	Student logs in → Dashboard loads batch details	Batch details displayed within 2s	Displayed in 1.8s	Pass
IT-02	Student submits assignment → File stored in DB & visible to faculty	Assignment visible in faculty portal	Verified successfully	Pass
IT-03	Student sends chat message → Message stored in DB & broadcast to batch	Message visible to all batch members	Delivered instantly	Pass

The results showed that the app, backend, and database are working together smoothly.

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering & Technology Department of Information and Communication Technology	
Subject: CP (01CT0715)	Aim: Testing and Validation	
	Date:	Enrollment No: 92200133036

4. Performance Metrics

To check how the app performs under different conditions, I measured a few key metrics:

Metric	Target	Observed Result	Analysis
API Response Time (Login)	< 2s	1.6s avg	Meets requirement
Assignment Upload	< 5s for 10MB file	4.2s	Acceptable
Chat Latency	< 1s	0.4s	Excellent
Concurrent Users (Load Test)	100 users	Stable up to 120 users	Scalable
DB Query Time (Attendance Fetch)	< 1s	0.7s	Efficient

5. Validation Against Objectives

The main objectives of the project were to provide students with a mobile app where they can:

- View batch details, attendance, and fees
- Submit assignments and give exams
- Chat with batchmates
- Access study material and faculty details

Testing confirmed that all these objectives were met. Students can log in, view their academic records, submit assignments, and interact with peers without issues. The only limitation observed was a slight delay in assignment uploads under very heavy load, which has already been addressed with proposed optimizations.