	Marwadi University **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: CP (01CT0715)	**Aim: Deployment and Operations**	
	Date:	**Enrollment No: 92200133036**

1. Deployment Process

The Acadexa *Student App* was developed as a mobile application using **Flutter** (front-end), **PHP** (API layer), and **MySQL** (database). At this stage, the application has not been deployed to a public app store (Google Play or Apple App Store) or a live cloud server. Instead, the complete source code and backend configuration files are hosted on **GitHub**, which serves as the deployment medium beyond localhost. This ensures that the project is accessible, reproducible, and can be built and run on any compatible system.

Platform Selection Rationale:

- **GitHub Repository**: Selected for version control, collaboration, and accessibility. It allows evaluators and developers to clone the project and run it locally or on an emulator.
- **Play Store/App Store (Future Plan)**: The intended deployment environment for the app, since it is designed for student use on mobile devices.

Deployment Steps Completed:

1. Created a GitHub repository and pushed the Flutter app, PHP APIs, and MySQL schema.
2. Documented setup instructions in the README file for easy replication.
3. Verified that the app can be cloned, built, and run successfully on another system/emulator.

Future Deployment Plan (Play Store):

- Generate a signed APK (Android) or IPA (iOS).
- Configure app metadata (name, description, screenshots).
- Upload to Google Play Console / Apple Developer Account.
- Conduct beta testing before public release.

Evidence of Deployment:

- GitHub Repository Link: *(insert your repo link here)*
- Screenshots of the repository and emulator build are included in the Appendix.

2. Monitoring Strategy

Since the app is not yet live on an app store, real-time monitoring tools have not been implemented. However, a **monitoring strategy** has been designed for future deployment to ensure system reliability and user satisfaction.

Proposed Monitoring Tools:

- **Firebase Crashlytics**: For real-time crash reporting and error tracking.
- **Google Analytics for Firebase**: To monitor user engagement, active sessions, and retention.
- **API Monitoring (Postman Monitors / Grafana)**: To track backend API response times and error rates.

Key Performance Indicators (KPIs):

1. **API Response Time** – Average time for login, attendance fetch, and assignment upload.
2. **Crash-Free Sessions** – Percentage of app sessions without crashes.
3. **User Engagement** – Daily active users (DAU) and session duration.

Baseline Results (from local testing):

- Login API response time: **1.6s**
- Assignment upload success rate: **100% under normal load**
- Chat latency: **0.4s**

These KPIs will be continuously tracked once the app is deployed on the Play Store.

3. Maintenance Plan

A structured maintenance plan has been outlined to ensure long-term reliability once the app is deployed.

Planned Activities:

- **Weekly Backups**: Database backups to prevent data loss.
- **Monthly Security Audits**: Review PHP APIs for vulnerabilities.
- **Dependency Updates**: Regularly update Flutter packages and PHP libraries.
- **Bug Fixes**: Address issues reported via Firebase Crashlytics.

Potential Issues & Mitigation:

- **Scalability**: If user load increases, migrate backend to AWS or Azure with load balancing.
- **Software Dependencies**: Outdated libraries may cause compatibility issues → schedule quarterly updates.
- **Server Downtime**: Use cloud hosting with redundancy to minimize downtime.

Maintenance Schedule:

- Weekly: Database backup and error log review.
- Monthly: Security audit and dependency updates.
- Quarterly: Performance optimization and scalability review.

4. Challenges

The main challenge was the **lack of live deployment** to Play Store/App Store due to administrative and financial constraints (developer account fees, compliance requirements). To address this, the project was hosted on GitHub, ensuring accessibility and reproducibility for evaluators.

Another challenge was ensuring **cross-platform compatibility**. This was mitigated by testing the app on both Android Emulator and a physical Android device to confirm stability.