# 9530

**St.MOTHER THERESA ENGINEERING COLLEGE**

COMPUTER SCIENCE ENGINEERING

**NM-ID**: 47C946261E6C08EA89841788EAF123D5

**REG NO**: 953023104077

**DATE:**22-09-2025

**Completed the project named as**

**Phase 3**

FRONT END TECHNOLOGY

# INTERACTIVE QUIZ APP

**SUBMITTED BY,**

MUTHU SHIVANI S

9042058025

## Phase 3 — MVP Implementation :

The Interactive Quiz App is being developed in structured phases to ensure a robust and user-friendly product. Phase 3 represents one of the most crucial steps in the journey — the MVP Implementation. The goal of this phase is to create a working prototype of the app that contains all the core functionalities, ensuring that users can experience its value even before advanced features are added. This document provides an elaborate explanation of each task involved in Phase 3. By the end of this phase, the app will be fully functional at its most basic level, with user engagement, quiz management, and data handling features operational.

### 1.Project Setup :

The first step in Phase 3 is laying down the technical foundation. Project setup is crucial as it defines the structure and standards of the entire codebase. During this stage, the following tasks are completed: Setting up the development environment with necessary tools and frameworks such as React, Flutter, or Angular. Configuring package managers, build tools, and linters for consistency. Establishing GitHub repositories with well-defined branching strategies for collaboration. Defining coding standards and documentation guidelines for maintainability.

Integrating Continuous Integration/Continuous Deployment (CI/CD) pipelines for faster testing and deployment. By focusing on a robust setup, developers minimize technical issues and enable smooth collaboration throughout the development cycle.

### 2.Core Features Implementation :

The Interactive Quiz App is designed to deliver a fun, educational, and engaging experience. Core features form the backbone of the app and include the following:

- **User Authentication:** Registration and login functionalities allow users to create profiles and track progress. Quiz.
- **Creation & Management:** Admins or instructors can upload questions, define categories, and organize quizzes.
- **Question Types:** Support for multiple-choice, true/false, and short-answer formats.

- **Timed Quizzes:** Enhancing interactivity by introducing countdown timers for each quiz session.
- **Scoring System:** Automatic score calculation upon completion of quizzes.
- **Leaderboards:** Encouraging competition by ranking users based on their performance.
- **Responsive UI:** Mobile-friendly layouts ensuring accessibility across devices. Implementing these core features ensures the MVP is capable of serving its primary purpose: offering users an interactive quiz experience with measurable outcomes.

**3.Data Storage (Local State / Database) :**

Efficient data management is a critical requirement for the quiz app. The MVP balances simplicity with reliability in its storage strategy. Initially, local state management can be employed for lightweight functionality, while a database is integrated for long-term scalability. Local State: Temporary storage for user progress during a quiz session. Relational Database: SQL databases like MySQL/PostgreSQL to store quiz questions, answers, and results. NoSQL Option: MongoDB for flexibility in handling unstructured quiz data. Cloud Integration: Using Firebase or AWS DynamoDB for real-time synchronization. Data storage not only ensures persistent quiz records but also helps personalize user experiences, support analytics, and maintain high reliability as user numbers grow.

**4.Testing Core Features :**

Testing guarantees that the MVP runs smoothly across different devices and use cases. The app undergoes various levels of testing to eliminate errors and provide a bug-free experience: Unit Testing: Verifying individual functions such as question display, scoring, and authentication.

Integration Testing: Ensuring modules such as login, quiz selection, and leaderboard interact correctly. Performance Testing: Checking responsiveness under multiple concurrent users. User Acceptance Testing (UAT): Involving real users to test app usability and provide feedback.

Comprehensive testing ensures reliability, builds user trust, and sets the foundation for future upgrades.

## 5.Version Control (GitHub) :

Version control is essential in managing code changes, especially in collaborative projects. GitHub acts as the central hub for maintaining the Interactive Quiz App's codebase. Key practices include: Creating feature branches to isolate new functionality. Using pull requests for peer reviews and quality assurance. Maintaining a clean commit history for transparency. Tracking issues and milestones to manage project progress. Automating deployments with GitHub Actions. With GitHub, the team benefits from efficient collaboration, accountability, and better release management practices.