



**COLLEGE CODE : 9623**

**COLLEGE NAME : Amrita College of Engineering And Technology**

**DEPARTMENT : Computer Science and Engineering**

**STUDENT NM-ID: 9878F197076CE6DB4E74BA1EB71SFF81**

**ROLL NO : 23CS047**

**DATE : 25-09-2025**

**Completed the project named as**

**Phase 3 MVP Implementation**

**PROJECT NAME: INTERACTIVE QUIZ APP**

**SUBMITTED BY,**

**NAME : Kamalesh.T**

**MOBILE NO: 9043586106**

# Phase 3 – MVP Implementation (Deadline – Week 8)

## 1. Project Setup

The project setup marks the foundation for the MVP (Minimum Viable Product) development.

### 1.1 Environment Setup

- Installing Node.js, React.js, and MongoDB drivers.
- Using npm/yarn for dependency management.
- Structuring project directories for modularity.

### 1.2 Frontend Setup

- Creating React.js app with Tailwind CSS.
- Configuring routing with React Router.
- Implementing state management (Context API / Redux).

### 1.3 Backend Setup

- Initializing Express.js server with REST APIs.
- Middleware setup for logging, authentication, and validation.
- Creating routes for users, quizzes, and results.

### 1.4 Database Setup

- MongoDB Atlas cluster setup.
- Designing collections for users, quizzes, results, and sessions.
- Connecting securely using .env configurations.

### 1.5 GitHub & Deployment Setup

- Repository creation with branch protection.
- CI/CD workflows for automated deployment.
- Hosting frontend on Vercel/Netlify, backend on Render/AWS/Heroku.

## 2. Core Features Implementation

Core features define the first working version of the application.

### 2.1 User Authentication

- JWT-based signup/login.
- Password hashing with bcrypt.
- Token expiration and refresh handling.

### 2.2 Quiz Management (Admin)

- CRUD for quizzes and questions.
- Bulk upload using CSV/Excel.
- Analytics dashboard for admins.

### 2.3 Quiz Attempt (User)

- Randomized questions per attempt.
- Timer with auto-submit.
- Preventing cheating (e.g., disabling copy-paste).

### 2.4 Feedback & Scoring

- Real-time score calculation.

- Display correct/incorrect answers.
- Leaderboards for competition.

#### 2.5 Progress Tracking

- History of past attempts stored in database.
- Graphical insights using charts.
- Weekly/monthly reports for users.

### 3. Data Storage (Local State / Database)

Data is stored both locally and in MongoDB Atlas.

#### 3.1 Local State Management

- Storing temporary answers, timer data, and session state.
- Managed with React hooks and Context API.

#### 3.2 Database Collections

- Users Collection: name, email, password, role, history.
- Quizzes Collection: title, category, difficulty, questions.
- Results Collection: userId, quizId, score, answers, date.

#### 3.3 Example Schema (MongoDB)

```
{
  "quizId": "12345",
  "title": "JavaScript Basics",
  "questions": [
    {
      "question": "What is closure?",
      "options": ["Function inside function", "Loop", "Variable", "None"],
      "correctAnswer": "Function inside function"
    }
  ]
}
```

### 4. Testing Core Features

Testing ensures functionality and reliability.

#### 4.1 Unit Testing

- Jest for frontend components.
- Mocha/Chai for backend APIs.

#### 4.2 Integration Testing

- Full user workflow: login → attempt quiz → submit → result.

#### 4.3 Performance Testing

- JMeter for load testing (simulate 1000 users).

#### 4.4 Bug Tracking

- GitHub Issues for bug reports.
- Kanban board for assignments.

## 5. Version Control (GitHub)

GitHub ensures collaboration and code integrity.

### 5.1 Repository Management

- README, license, and guidelines setup.

### 5.2 Branching Strategy

- main branch: production-ready.
- dev branch: feature integration.
- feature branches: isolated development.

### 5.3 Collaboration Practices

- Pull requests with code reviews.
- Conventional commit messages.
- GitHub Actions for CI/CD pipelines.

## 6. Challenges in MVP Implementation

Challenges faced during development:

- Scalability for larger quizzes and data.
- Ensuring real-time performance with many users.
- Efficient database indexing for fast queries.
- Security threats like SQL injection, XSS, CSRF.

## 7. Conclusion

By the end of Phase 3, the Interactive Quiz App MVP will:

- Support authentication, quiz management, and attempts.
- Provide instant feedback and progress tracking.
- Be deployed with version control and testing.

This serves as a strong foundation for Phase 4.