



**9530**

**St. MOTHER THERESA ENGINEERING COLLEGE**

**COMPUTER SCIENCE ENGINEERING**

**NM-ID: 772987860203767908241637198AA868**

**REG NO: 953023104121**

**DATE:06-10-2025**

**Completed the project named as**

**Phase 5**

**FRONT END TECHNOLOGY**

**INTERACTIVE QUIZ APP**

**SUBMITTED BY,**

**SUBBULAKSHMI M**

**9150566495**

# **Interactive Quiz App**

# 1. Final Demo Walkthrough

The final demonstration showcased the complete functionality of the *Interactive Quiz App*. The walkthrough covered:

- **User Registration & Login:** Secure authentication allowing personalized quiz experiences.
- **Dashboard Interface:** Displays quiz categories, scores, and user progress.
- **Quiz Attempt Flow:** Real-time question rendering, timer-based answering, and instant score calculation.
- **Leaderboard:** Dynamic ranking system updating based on user performance.
- **Admin Controls:** Ability to add, edit, or delete quiz questions and view analytics.

During the demo, both desktop and mobile responsiveness were displayed, proving cross-platform compatibility.

## 2. Project Report

The report summarized the entire development lifecycle:

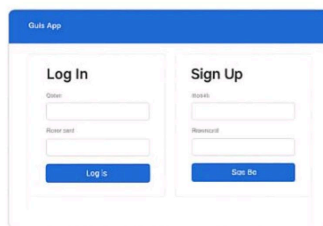
- **Objective:** To create an engaging quiz platform enhancing learning through gamification.
- **Technology Stack:**
  - *Frontend:* React.js / HTML, CSS, JavaScript
  - *Backend:* Node.js with Express
  - *Database:* MongoDB
  - *Deployment:* Netlify (Frontend) & Render/Heroku (Backend)
- **Key Features:** User authentication, randomized questions, result analytics, and leaderboard tracking.

- **Performance Evaluation:** Smooth loading times, optimized API calls, and strong data consistency.

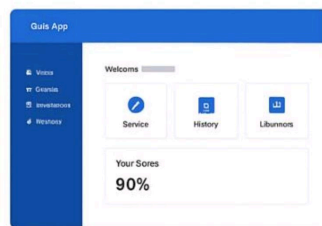
## 3. Screenshots / API Documentation

### Screenshots / API Documentation

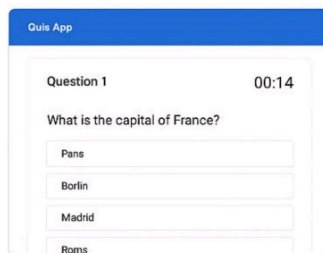
#### Screenshots Included:



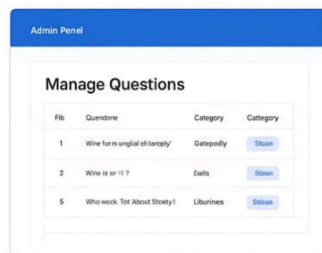
Login and Registration Page



Dashboard View



Quiz Interface with Timer



Admin Question Management Panel

#### API Documentation:

- **POST /api/register** –Registers new users
- **POST /api/login** – Authenticates users
- **GET /api/quizzes** –Fetches available quiz categories
- **POST /api/submit** –Submits quiz answers and calculates score
- **GET /api/leaderboard** –Returns top-performing users

## 4. Challenges & Solutions

Challenges	Solutions Implemented
Managing real-time quiz timer	Used React state hooks and <code>useEffect</code> for timer synchronization
Data inconsistency on quiz refresh	Implemented local storage to persist session data
Slow API response for large quiz sets	Optimized MongoDB queries and added pagination
UI design for multiple devices	Applied responsive design principles using CSS Flexbox and Grid
Deployment configuration issues	Resolved through environment variable setup and CI/CD debugging

## 5. GitHub README & Setup Guide

The **README.md** includes:

- Project Overview
- Features List
- Technology Stack
- Setup Instructions
  1. Clone the repository
  2. Run `npm install` for both client and server
  3. Add environment variables (`.env` file for DB and JWT key)
  4. Run the project using `npm start`

- Contribution Guidelines
- API Documentation Links
- Live Demo URL

## **6. Final Submission (Repo + Deployed Link)**

- **GitHub Repository:** <https://github.com/Subbulakshmi-11/SMTEC-NM-IBM>
- **Live Deployed Link:**