

MOBILE COMPUTING & APPLICATIONS LAB REPORT

Advanced OOP-Based Quiz Web Application (Student–Teacher System)

Team Members

1. Abu Bakkar Siddik – ID: 233003122E (All Coding & System Development)
2. Tipu Sultan – ID: 2330029222E (Documentation, Report, Diagrams)

Project Overview

This project is an advanced quiz management web application designed using HTML, CSS, JavaScript, and OOP principles. It includes Student Login, Teacher Login, dynamic subjects, time customization, randomized questions, and a teacher-only backend marks section similar to Google Forms.

Technologies Used

- HTML5 – structure of pages
- CSS3 – styling & responsive UI
- JavaScript (ES6) – logic, OOP classes, role system,
- Local Storage – storing student quiz results backend-like for the teacher
- Browser DOM API – dynamic UI rendering

Why OOP Was Used

OOP makes the system modular, scalable, and maintainable.

- Class Question – stores question text, options, and correct index

- Class Quiz – handles question flow, navigation, and answer storage
- Class Timed Quiz (extends Quiz) – adds timer functionality
- Class Result – stores score, grade, analytics
- Class Subject Manager – dynamic subject creation
- Class QuizService – central logic to load quiz, randomize questions, calculate score

Why JavaScript

JavaScript allows building full logic inside the browser without a backend server. It was used to:

- Handle login system (student/teacher)
- Create OOP classes
- Store marks in local Storage like backend
- Dynamically render UI like Google Forms.

Major Code Components

```
class Question {...}
```

```
class Quiz {...}
```

```
class TimedQuiz extends Quiz {...}
```

```
class Result {...}
```

```
class SubjectManager {...}
```

```
class QuizService {...}
```

Each class performs specific tasks, making system modular.

Student Features

- Login with Name & ID
- Choose subject
- Choose number of questions
- Custom or fixed timer
- Randomized questions & options

- Submit & get detailed result
- Full answer review

Teacher Features

- Login (username: teacher, password: teacher123)
- View backend Marks Table (stored using localStorage)
- Add subjects dynamically
- Add questions dynamically (like Google Forms)
- Review performance of all students

Why This System Is Advanced

- Multiple roles implemented (Student + Teacher)
- Dynamic subject & question management
- Timer customization
- Randomization logic
- Full OOP implementation
- Teacher-only backend marks section
- Works fully offline

The over view:

Advanced Mobile Quiz App
Student & Teacher login • Dynamic subjects • Custom timer • OOP-based engine

Welcome

Select your role to continue: **Student** or **Teacher**.

Student Login
Students can attempt quizzes and see their own result at the end.

Teacher Login
Teachers can see all stored marks and manage subjects & questions (like Google Forms).

Student Name
e.g., teacher

Teacher Username
e.g., teacher

Student ID
e.g., 233003122E

Password
e.g., teacher123

Demo credentials – Username: teacher, Password: teacher123

Continue as Student

Login as Teacher

Advanced Quiz App – Mobile Computing & Applications Lab Project (Student + Teacher + OOP)

Student login:

The screenshot shows the 'Advanced Mobile Quiz App' interface. At the top, a blue header bar displays the app's name and features: 'Advanced Mobile Quiz App', 'Student & Teacher login • Dynamic subjects • Custom timer • OOP-based engine'. Below the header, a 'Welcome' message and a prompt to 'Select your role to continue: Student or Teacher.' are visible. A central 'Student Login' section contains fields for 'Student Name' (with 'ab' entered) and 'Student ID' (with '233003122e' entered). A large blue button at the bottom right of this section says 'Continue as Student'.

Student Quiz view:

The screenshot shows the 'Advanced Mobile Quiz App' interface from a student's perspective. At the top, a blue header bar displays the app's name and features: 'Advanced Mobile Quiz App', 'Student & Teacher login • Dynamic subjects • Custom timer • OOP-based engine', 'Logged in as Student ab (233003122e)', and a 'Logout' link. Below the header, a 'Start a Quiz' section allows the student to choose a subject ('C Programming'), number of questions ('5'), and time limit ('60s'). A 'Start Quiz' button is present. To the right, a 'Student Features' sidebar lists: 'Attempt quizzes in different subjects', 'Change time limit (fixed or custom)', 'Randomized questions & options each attempt', and 'See detailed result and answer review at the end'.

Teacher login:

Teacher Login

Teachers can see all stored marks and manage subjects & questions (like Google Forms).

Teacher Username

Password

Demo credentials – Username: teacher , Password: teacher123

Login as Teacher

ications Lab Project (Student + Teacher + OOP)

Teacher Dashboard:

Teacher Dashboard

Only logged-in teachers can see this part. This works similar to a simplified Google Forms + response sheet: you can create subjects and questions, and view stored quiz marks of students.

Stored Quiz Results (Marks)

These results are stored locally in this browser (demo mode). Only teachers can see them.

Student Name	Student ID	Subject	Score	Percentage	Grade	Attempted	Time Used	Taken At
ab	233003122e	C Programming	1/5	20.00%	F	5	14s / 60s	12/18/2025, 1:32:39 AM

Question & Subject Management

Create subjects and add questions dynamically (similar to Google Forms question creation).

Add New Subject
Subject Key (no spaces, e.g., python)
e.g., python_basics
name) e.g., Python Basics

Add New Question
Select Subject C Programming
Enter the question
Question Text
Option 1
Option 2
Option 3
Option 4
Correct Option Index (0-3)

Conclusion

This project successfully meets Mobile Computing lab standards with advanced features. Team effort:

1. Abu Bakkar Siddik – Developed full codebase
2. Tipu Sultan – Created documentation & report

The system is scalable, secure for demo, and similar to Google Forms.