# Online Quiz Application

## Phase:5 (project Demostration and Documentation)

## 1 Final Demo Walkthrough:

This phase demonstrates the working model of your application. You should perform a step-by-step live demo or a recorded video walkthrough showing how each feature of your quiz system works.

### 💻 Demo Flow:

**Step 1: User page**

- Front pages is displaying
- Select no if question
- Select category
- Select difficulty

**Step 2:**

- After select the above items
- Press start quiz button

**Step3: Quiz Page**

- Demonstrate fetching quiz questions dynamically from the backend REST API.

**Step4: Quiz Attempt**

- Show timer functionality (if available), navigation between questions, and answer selection.

**Step5: Submission & Result**

- Submit answers → show correct/incorrect responses.

**Step6: Admin Panel (optional)**

- Display how the admin can add, update, or delete quiz questions.

# 2 Project Report:

## 1. Introduction:

The Online Quiz Application is a web-based system that allows users to take quizzes and view their results instantly. It is designed to replace traditional paper-based assessments with a digital, efficient, and interactive platform.
This project uses Node.js and Express.js for backend API development and MongoDB for data storage. The system enables smooth communication between the client and server using RESTful APIs, ensuring secure and efficient performance.

## 2. Objective:

**The main objective of this project is to:**

- Develop a dynamic and scalable online quiz system.
- Allow users to register, log in, and attempt quizzes.
- Provide instant evaluation and result generation.
- Enable admin control for managing quiz questions.
- Integrate REST API for backend communication with frontend.

## 3. Scope of the Project:

**The project can be used in:**

- Timepass as well as knowledge
- Checking the knowledge about methology and sports

**It supports:**

- Multi-user login
- Multiple-choice quizzes
- Result tracking and analytics
- REST API integration for external systems

**4. System Requirements:**

**Software Requirements:**

| Component | Description |
|---|---|
| ●Frontend | ●HTML, CSS, EJS |
| ●Backend | ●Node.js, Express.js |
| ●Database | ●MongoDB(trivia) |
| ●Tools | ●VS Code, GitHub |

**Hardware Requirements:**

| Component | Description |
|---|---|
| ●Processor | ●Lenovo |
| ●RAM | ●4 GB or higher |
| ●Storage Minimum | ●250 GB |
| ●Internet | ●Required for API and deployment |

**5.System Design:**

- **Architecture Diagram:**

  [User Interface] → [Express.js Server] → [REST API] → trivia [MongoDB Database]

- **Flowchart:**

  Start → Fetch Quiz → Attempt Questions → Submit Answers → Display Result → End

**6. Modules Description:**

**User Module:**

- User registration and login with validation.
- Session management using tokens.
- Allows users to access quiz content.

**Quiz Module:**

- Fetches quiz questions from API.
- Displays one question at a time.
- Records selected answers.

**Result Module:**

- Evaluates answers automatically.
- Stores results in the database.
- Displays user performance summary.

**Admin Module:**

- Admin can create, update, and delete quiz questions.
- Can view user performances and manage data.

## 7.Result:

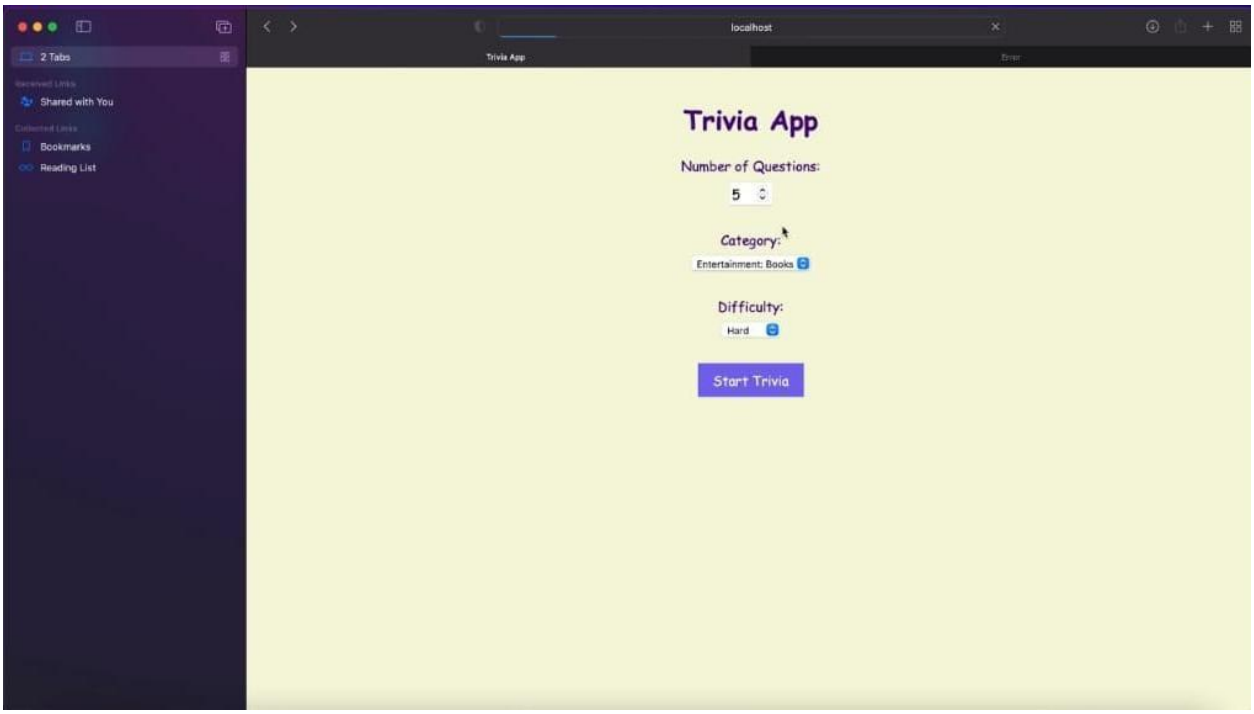
**The project successfully allows users to:**

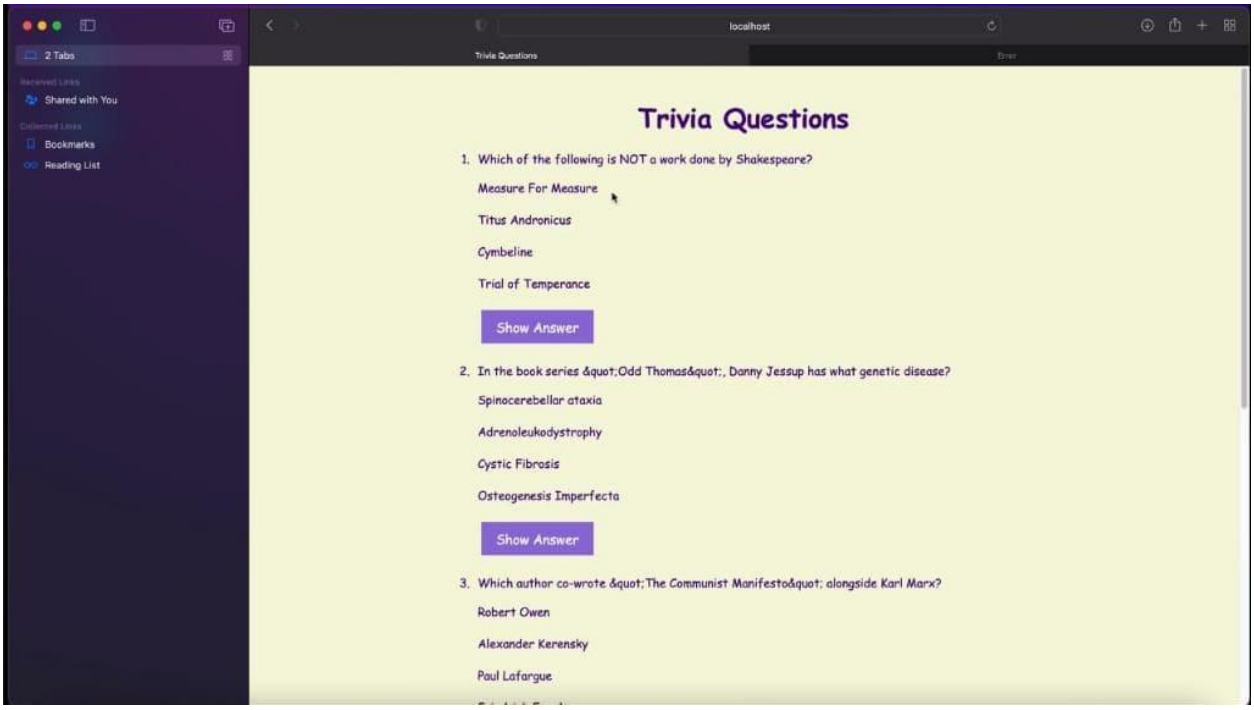
- Attempt quizzes fetched dynamically from the API.
- Receive instant score results.
- Provides admin management of quiz data.

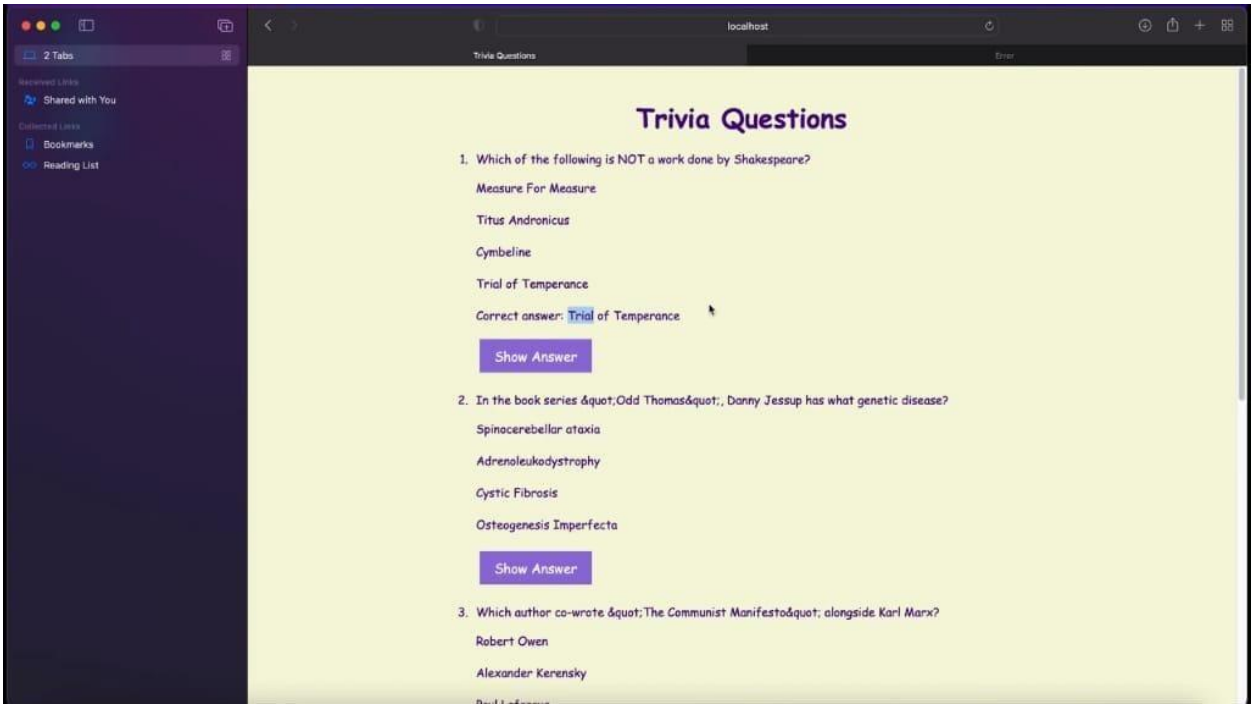
## 8.Conclusion:

The Online Quiz Application is a reliable and efficient platform for conducting online assessments. It automates question management, user participation, and scoring through RESTful API architecture. The system can easily be scaled and integrated into educational or corporate testing platforms.

## 3 Screenshots/API Documentation:






## 4 Challenges & Solutions:

| Challenges | Solutions |
|---|---|
| ●Rendering EJS file threw “missing catch error. | ● Added catch block after try in trivia.ejs. |
| ●Handling asynchronous API cals | ● Used async/await and proper error handling. |
| ●Data not saving to MongoDB | ● Checked connection string and used mongoose.connect() correctly. |
| ●CORS issue during frontend-backend connected. | ● Enabled CORS middleware in Express |
| ●Slow loading during deployment Minimize. | ● Optimized server routes and Assets. |

# 5 Git hug README & Setup guide:

A Online Quiz Application built using Node.js, Express.js, MongoDB, and REST API.
This app allows users to register, log in, take quizzes, view scores, and track progress in real-time.

## Tech Stack:

| Component | Technology Used |
|---|---|
| ●Frontend | ●HTML, CSS, JavaScript (or React if applicable) |
| ●Backend | ●Node.js, Express.js |
| ●Database | ●MongoDB Atlas |
| ●Authentication | ●JWT (JSON Web Token) |
| ●Hosting (optional) | ●Render / Vercel / Heroku |

## Installation and Setup Guide:

**1.Prerequisites:**

**Make sure you have installed:**

- Node.js (v16 or higher)
- MongoDB Atlas account (or local MongoDB)
- Git

**2.Clone the Repository:**

- git clone https://github.com/your-username/online-quiz-application.git

cd online-quiz-application

**3.Install Dependencies:**

- npm install

4.**Configure Environment Variables:**

- Create a .env file in your root directory and add:PORT=5000
- MONGO_URI=your_mongodb_connection_string
- JWT_SECRET=your_jwt_secret_key

**5.Run the Server:**

- npm start
- Or in development mode (with live reload):
- npm run dev
- Server will start at:http://localhost:5001

**6.Test the API using Postman:**

**Endpoints examples:**

| Method | Endpoint | Description |
|---|---|---|
| POST | /api/auth/register | Register a new user |
| POST | /api/auth/login | Login user |
| GET | /api/quiz/questions | Fetch quiz questions |
| POST | /api/quiz/submit | Submit quiz answers |
| GET | /api/score/:id | Get user score |

## Folder Structure:

```
online-quiz-application/
│
├── backend/
│   ├── server.js
│   ├── routes/
│   ├── controllers/
│   ├── models/
│   ├── middleware/
│   └── config/
│
├── frontend/
│   ├── index.html
│   ├── quiz.js
│   └── styles.css
│
├── .env
├── package.json
└── README.md
```

**Common Issues & Solutions:**

| Challenge | Solution |
| --- | --- |
| ●MongoDB connection failed. | ● Check MONGO_URI and enable IP access in MongoDB Atlas. |
| ●Server crashes on invalid inputs. | ●Wrap API calls in try...catch blocks. |
| ●CORS error | ●Enable CORS in Express using npm i cors |
| ●JWT not verified | ●Ensure correct Authorization header format: Bearer token. |

## 6 Final submission:

**Reposipory link:**

https://github.com/lakshitha2206/Online_Quiz-Application-.git