

**Title: Trivia Trek: Journey through a World of Questions**

**A PROJECT REPORT**

For

**“SEC160: FULL STACK”**

Submitted by

**Tarun Adithya - AP23110010451**

**Harsha - AP23110010622**

**Manikanta - AP23110010362**

**Sowjanya – AP23110010614**

**SEM: V**

**SECTION: J**

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

SCHOOL OF ENGINEERING AND SCIENCES



**SRM**  
UNIVERSITY AP

**Andhra Pradesh**

**SRM University-AP, Neerukonda, Andhra Pradesh 522240**

**December 2025**

# Introduction

The evolution of e-learning platforms and competitive quiz systems has transformed the way students practice, learn, and evaluate themselves. Instead of static question papers or manual evaluation, modern quiz applications provide instant scoring, feedback, time-tracking, and personalized insights into performance.

TriviaTrek is an advanced, responsive, and interactive quiz application developed using React.js, JSON Server, Tailwind CSS, and Context API. It provides a complete ecosystem for both users (students attempting quizzes) and admins (who create quizzes, add questions, and manage content).

TriviaTrek includes features such as category-wise quizzes, a real-time timer, animated UI using Framer Motion, dynamic scoring, question navigation, and a fully functional admin dashboard with CRUD operations.

The project demonstrates how modern web technologies can be combined to build an interactive assessment platform suitable for e-learning, competitive exam preparation, and training environments.

## Scenario-Based Introduction

Imagine a student preparing for a placement exam. She wants an interactive, fast, and personalized quiz system where she can attempt quizzes, track her score, review mistakes, and gradually improve.

However, most existing systems are either:

- Not interactive
- Do not provide instant analytics

- Do not allow admins to dynamically add/update questions
- Lack UI/UX sophistication

**TriviaTrek** solves these problems by creating a seamless experience for both learners and administrators.

For example:

A student logs in → chooses a category → starts a timed quiz → receives instant scoring → views mistakes → downloads PDF summary → tracks progress via charts.

At the same time, an admin can:

Create quizzes → add/edit/delete questions → view quiz statistics → manage categories → visualize data through charts.

TriviaTrek bridges the gap between usability and performance, ensuring a smooth, modern quiz-taking experience.

## **Target Audience :**

**The TriviaTrek Quiz Management System is thoughtfully designed to cater to a wide range of users across education, technology, assessment, and learning sectors. Each audience group benefits uniquely from the interactive features, analytics, and intuitive design of the platform.**

**Below is a more detailed and comprehensive breakdown of the target users:**

## **1) Students & Competitive Exam Aspirants**

**Students preparing for examinations—be it internal unit tests, semester examinations, national-level competitive exams, coding assessments, or placement readiness—can leverage TriviaTrek as a self-assessment and improvement tool.**

### **Why students benefit:**

- Ability to practice quizzes across multiple categories (C, OS, CN, DSA, AI-ML, etc.)**
- Instant feedback to help identify strengths and weaknesses**
- Timer-based tests simulate real exam environments**
- Score tracking and attempt history help measure improvement over time**
- PDF reports allow offline review and revision**
- Personalized analytics (charts & graphs) promote better learning decisions**

### **Ideal for:**

- B.Tech students**
- Intermediate students**
- Competitive exam aspirants (GATE, EAMCET, CAT, placement exams)**
- Students revising concepts before interviews**

## **2) Teachers, Trainers & Academic Instructors**

**Educators can use the Admin Panel to create digital assessments quickly without relying on manual question paper creation or offline evaluation.**

**Benefits for teachers:**

- Create unlimited quizzes with custom time limits**
- Add any number of questions with easy-to-use forms**
- Edit or delete quizzes dynamically based on class requirements**
- View analytics to understand student difficulty levels**
- Save time by automating grading and evaluation**

**Suitable for:**

- School teachers**
- College faculty**
- Coaching institutions**
- Corporate trainers conducting internal assessments**

## **3) Developers & Learners Exploring Full Stack Development**

**TriviaTrek is an excellent resource for students and developers exploring practical implementation of full-stack concepts.**

**What they learn:**

- Frontend development using React.js**

- **Implementing CRUD operations (Create, Read, Update, Delete) with JSON Server**
- **Global state management using Context API**
- **Routing and navigation using React Router**
- **User authentication flow (Admin/User separation)**
- **Component reusability and modular architecture**
- **Integrating charts, animations, and UI enhancements**
- **Persisting quiz results using LocalStorage**

**Best for:**

- **Beginners transitioning into full-stack development**
- **Students building portfolio projects**
- **Developers preparing for MERN stack interviews**
- **Hackathon participants creating assessment platforms**

## **4)E-Learning Platforms & EdTech Startups**

**TriviaTrek's modular architecture makes it ideal as a plug-and-play solution in larger educational ecosystems.**

**Why EdTech companies benefit:**

- **Can integrate quiz module with existing LMS systems**
- **Fully customizable: categories, subjects, difficulty levels**
- **Backend flexibility (JSON Server → Node.js → Firebase → MongoDB)**

- **Mobile-responsive user interface suitable for app integration**
- **Scalable quiz engine with analytics**

**Examples of use cases:**

- **Online learning apps**
- **Coaching institute platforms**
- **Course marketplaces (Coursera/Udemy-style)**
- **Internal training modules for companies**

## **5) Evaluators, Mentors & Educational Administrators**

**People responsible for measuring student performance can use TriviaTrek as an automated evaluation tool.**

**Advantages:**

- **Eliminates manual checking**
- **Generates performance insights instantly**
- **Shows strength/weakness patterns via graphs**
- **Allows comparison of attempts across time**
- **Provides downloadable reports for documentation**

**Useful for:**

- **Academic coordinators**
- **College mentors**
- **Placement training teams**
- **Internship evaluators**

- HR teams assessing candidate skill levels

## 6) Parents and Guardians (Indirect Users)

Parents can use the quiz reports to monitor their child's learning outcomes through downloadable performance summaries.

## 7) General Knowledge Enthusiasts & Lifelong Learners

Anyone who enjoys learning new concepts or testing their knowledge can use the platform for daily mental workouts.

### Key Features:

#### ◆ User Features

- Category-based quizzes
- Timer-enabled questions
- Real-time score calculation
- Option highlighting (correct/wrong)
- Progress tracking
- Review answers post-quiz
- PDF report generation
- Beautiful animations using Framer Motion
- Personalized history and attempts dashboard

## ◆ Admin Features

- Add Quiz
- Edit Quiz
- Delete Quiz
- Add / Edit / Delete Questions
- Bar chart & Pie chart analytics
- Dashboard statistics
- JSON-server-based persistent storage

## ◆ Technical Features

- Context API-based global state management
- Fully responsive UI (Tailwind CSS)
- Component-driven architecture
- LocalStorage for storing user history
- JSON Server for backend simulation
- Recharts for analytics

## Pre-requisites:

### Technical Knowledge

- React.js
- JavaScript (ES6+)
- Tailwind CSS
- JSON Server (for backend)
- Routing & State Management

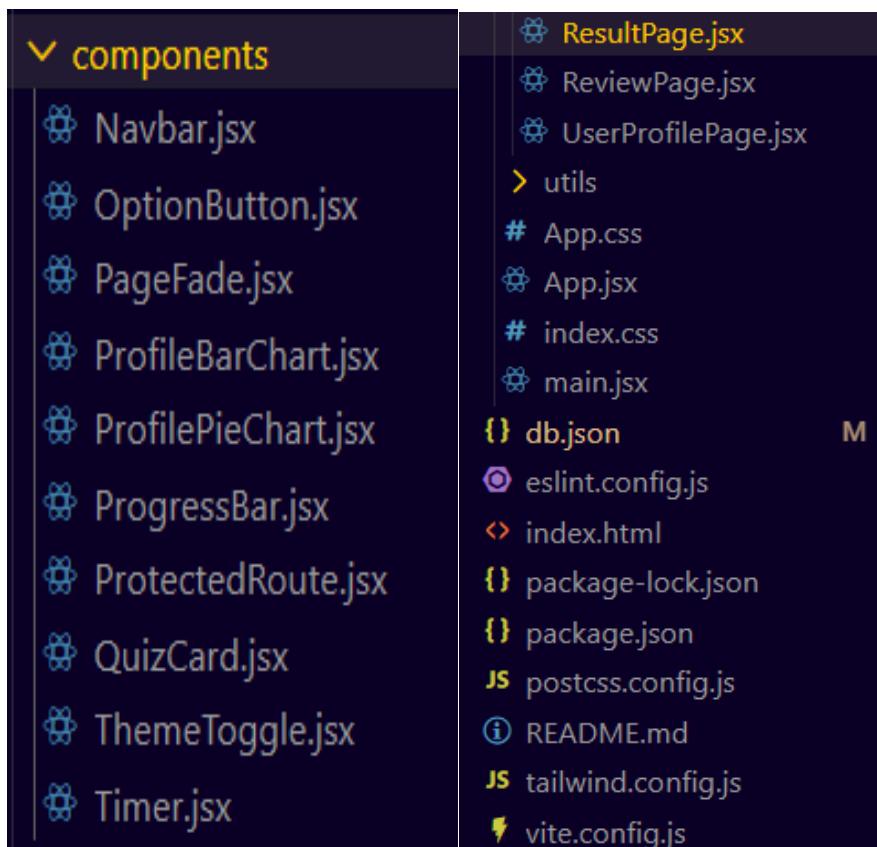
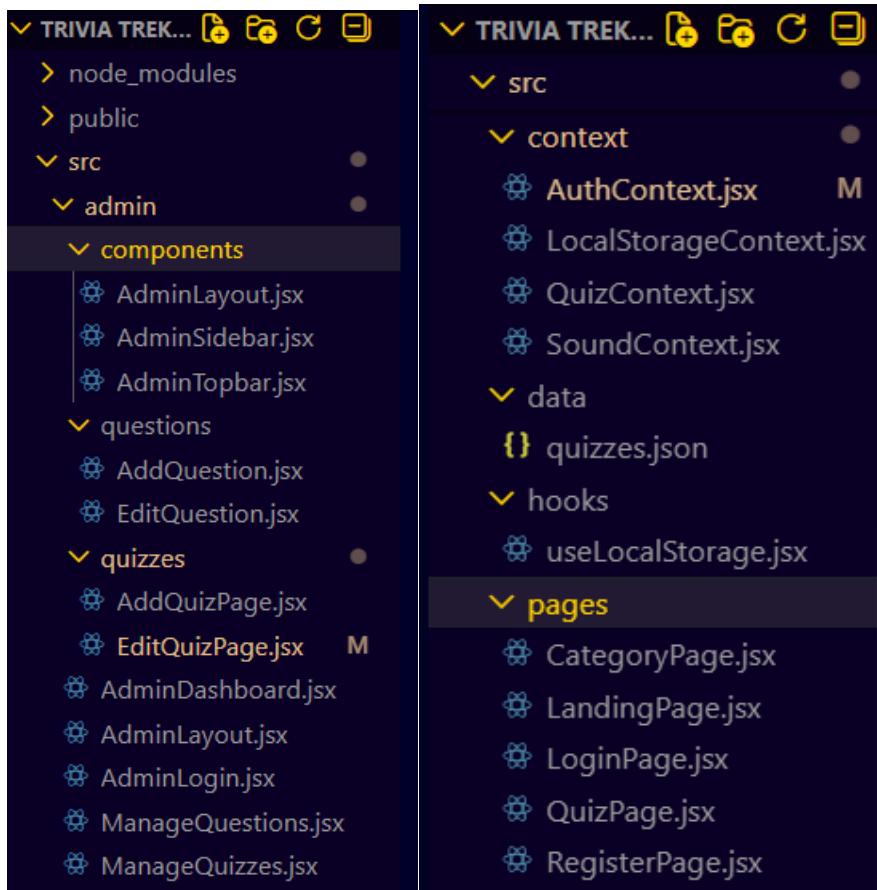
### Software Requirements

- Node.js
- Code Editor (VS Code)
- Browser (Chrome/Edge)
- NPM

## Libraries Used

- React Router
- React Hot Toast
- Axios
- Tailwind CSS
- Recharts
- Framer Motion

## Project Structure



## **1. components/**

This folder contains **reusable UI components** used across the entire TriviaTrek application.

Examples include:

- Buttons (OptionButton, ThemeToggle)
- Progress indicators (Timer, ProgressBar)
- Animations (PageFade)
- Charts (ProfileBarChart, ProfilePieChart)
- Navigation helpers (Navbar)

These components ensure:

- ✓ Visual consistency
- ✓ Reusability
- ✓ Faster development

## **2. pages/**

This directory holds **all user-facing pages**, where each file represents a separate route in the app.

Examples of pages:

- **LandingPage** → Welcome page
- **CategoryPage** → List of quiz categories
- **QuizPage** → Quiz-taking screen
- **ResultPage** → Displays user score
- **ReviewPage** → Shows answer review
- **UserProfilePage** → User history, attempts, graph analytics
- **LoginPage / RegisterPage** → Authentication screens

Each page serves as a full screen and connects the UI with context & backend logic.

### **3. admin/**

This folder contains everything related to the **Admin Portal**, including:

#### **admin/components/**

- AdminSidebar
- AdminTopbar
- AdminLayout (wrapping all admin routes)

#### **admin/quizzes/**

- AddQuizPage
- EditQuizPage

#### **admin/questions/**

- AddQuestion
- EditQuestion

#### **Top-level Admin Files**

- AdminDashboard
- ManageQuizzes
- ManageQuestions

Admin pages support complete CRUD operations using JSON Server:

- ✓ Add / Edit / Delete Quizzes
- ✓ Add / Edit / Delete Questions
- ✓ View total questions, stats, and analytics

## **4. context/**

Implements **global state management** using Context API.

**Includes Contexts such as:**

- **QuizContext** → Handles all Quiz CRUD operations using JSON Server
- **LocalStorageContext** → Stores quiz attempt history
- **AuthContext** → User authentication (Admin/User roles)
- **SoundContext** → Button click/correct/wrong audio

The Context layer:

- Centralizes the app logic
- Avoids prop drilling
- Keeps data synchronized across pages

## **5. public/**

Stores static assets accessible directly by the browser:

- Logos
- Background images
- Sound files (click.mp3, correct.mp3, wrong.mp3)
- Fallback icons

These files do not require Webpack bundling and load faster.

## **6. utils/**

Utility functions that simplify complex tasks.

Includes:

- **GeneratePDF.js** → Creates downloadable attempt reports
- **Helper functions**

- **Formatting utilities**
- **Reusable constants**

This keeps business logic separate from UI code.

## 7. App.jsx & main.jsx

These are the **core entry points** of the React application.

### main.jsx

- Renders the root <App /> component
- Wraps global providers:
  - QuizProvider
  - AuthProvider
  - LocalStorageProvider
  - SoundProvider

### App.jsx

Handles:

- All routing (User + Admin)
- ProtectedRoute checks
- Navigation integration
- Page-level layout structure

It defines complete navigation between:

- ✓ Public pages
- ✓ User-protected pages
- ✓ Admin-protected pages

## **PROJECT FLOW:-**

Project demo:

Before starting to work on this project, let's see the demo.

Demo link :

Milestone 1: Project Setup and Configuration:

### 1. Install Required Tools and Software

To begin the development of **Trivia Trek**, the essential software tools and

dependencies must be installed. These tools ensure smooth development, testing,

and execution of the frontend application.

Installation of Required Tools:

- Node.js & npm – Required to run React and install project dependencies
- Vite – Lightweight development bundler used to create React project
- Visual Studio Code – Code editor for writing and maintaining source code
- Google Chrome / Any Browser – For running and testing the application
- Git (optional) – For version control and repository management

For Further Reference, Use the Following Resources

- Official React documentation: <https://react.dev>
- Vite documentation: <https://vitejs.dev>
- React Router documentation: <https://reactrouter.com>
- MDN Web Docs for JavaScript & Web APIs

## Milestone 2: Web Development:

### 1. Setup React Application

The development of **Trivia Trek** begins with setting up the core React application using Vite. After initializing the project, essential folders such as components, pages, context, hooks, and utils are created to maintain a modular structure. The most important part of this milestone is the App.jsx file, which connects the entire application together.

```
src > @ App.jsx > ...
1  import React from "react";
2  import { Routes, Route } from "react-router-dom";
3
4  import Navbar from "./components/Navbar";
5  import LandingPage from "./pages/LandingPage";
6  import LoginPage from "./pages/LoginPage";
7  import RegisterPage from "./pages/RegisterPage";
8  import CategoryPage from "./pages/CategoryPage";
9  import Profile from "./pages/UserProfilePage";
10 import QuizPage from "./pages/QuizPage";
11 import ResultPage from "./pages/ResultPage";
12 import ReviewPage from "./pages/ReviewPage";
13 import ProtectedRoute from "./components/ProtectedRoute";
14 import { LocalStorageProvider } from "./context/localStorageContext";
15 /* ----- ADMIN IMPORTS ----- */
16 import AdminLayout from "./admin/AdminLayout";
17 import AdminDashboard from "./admin/AdminDashboard";
18 import AdminLogin from "./admin/AdminLogin";
19 import ManageQuizzes from "./admin/ManageQuizzes";
20 import AddQuizPage from "./admin/quizzes/AddQuizPage";
21 import EditQuizPage from "./admin/quizzes/EditQuizPage";
22 import ManageQuestions from "./admin/ManageQuestions";
23 import AddQuestion from "./admin/questions/AddQuestion";
24 import EditQuestion from "./admin/questions/EditQuestion";
25 /* ----- */
26
27 export default function App() {
28   return (
29     <div className="min-h-screen bg-gray-100 dark:bg-gray-900 transition-colors duration-300">
30       <LocalStorageProvider>
31         <Navbar />
32
33         <Routes>
34           {/* PUBLIC ROUTES */}
35           <Route path="/" element={<LandingPage />} />
36           <Route path="/login" element={<LoginPage />} />
37           <Route path="/register" element={<RegisterPage />} />
38           <Route path="/admin-login" element={<AdminLogin />} />
39
40           {/* USER PROTECTED ROUTES */}
41           <Route path="/categories" element={<ProtectedRoute><CategoryPage /></ProtectedRoute>}/>
42           <Route path="/quiz/:id/:attemptId" element={<ProtectedRoute><QuizPage /></ProtectedRoute>}/>
43           <Route path="/result/:id/:attemptId" element={<ProtectedRoute><ResultPage /></ProtectedRoute>}/>
44           <Route path="/review/:id/:attemptId" element={<ProtectedRoute><ReviewPage /></ProtectedRoute>}/>
45           <Route path="/profile" element={<ProtectedRoute><Profile /></ProtectedRoute>}/>
46           {/* ----- ADMIN ROUTES ----- */}
47           <Route path="/admin" element={<AdminLayout />} />
48           <Route index element={<AdminDashboard />} />
49           {/* QUIZZES */}
50           <Route path="quizzes" element={<ManageQuizzes />} />
51           <Route path="quizzes/add" element={<AddQuizPage />} />
52           <Route path="quizzes/edit/:id" element={<EditQuizPage />} />
53
54           {/* QUESTIONS */}
55           <Route path="questions" element={<ManageQuizzes />} />
```

2)

## A. Importing Required Modules (Updated for TriviaTrek Quiz App)

In the TriviaTrek application, the main App component imports all the essential modules required to build and run the project. These imports establish the structure, routing, authentication, and user/admin workflows of the application.

### 1. React & Core Hooks

The application imports React along with commonly used hooks like:

- **useState** – for managing component-level state
- **useEffect** – for handling side effects such as fetching quizzes

These hooks help maintain reactive UI updates throughout the application.

### 2. React Router Modules

The project uses **react-router-dom** to manage routing:

- **Routes** and **Route** → Define navigation paths
- **useNavigate**, **useParams** → Read parameters and navigate programmatically
- **ProtectedRoute** → Prevents unauthorized access to user-only pages

This allows seamless navigation between:

- Login Page
- Quiz Page
- Result Page
- Admin Dashboard
- Edit/Add Quiz pages

### 3. Animation Library (Framer Motion)

Imported for:

- Smooth page transitions (PageFade)

```
import React from "react";
import { motion } from "framer-motion";

export default function PageFade({ children }) {
  return (
    <motion.div
      initial={{ opacity: 0, y: 20 }}
      animate={{ opacity: 1, y: 0 }}
      exit={{ opacity: 0, y: -20 }}
      transition={{ duration: 0.4 }}
    >
      {children}
    </motion.div>
  );
}
```

```
import js from '@eslint/js'
import globals from 'globals'
import react from 'eslint-plugin-react'
import reactHooks from 'eslint-plugin-react-hooks'
import reactRefresh from 'eslint-plugin-react-refresh'
import { defineConfig, globalIgnores } from 'eslint/config'

export default defineConfig([
  globalIgnores(['dist']),

  {
    files: ['**/*.{js,jsx}'],

    extends: [
      js.configs.recommended,
    ],

    plugins: {
      react,
      'react-hooks': reactHooks,
      'react-refresh': reactRefresh,
    },

    languageOptions: {
      ecmaVersion: 'latest',
      sourceType: 'module',
      globals: globals.browser,
      parserOptions: {
        ecmaFeatures: {
          jsx: true,
        }
      }
    }
  }
])
```

- Animated buttons and elements
- Enhancing user experience with micro-interactions

Framer Motion makes the UI modern and interactive.

## 4. UI Components

Frequently used UI components are imported globally:

- **Navbar** – visible on all user pages
- **ThemeToggle** – switches between light/dark mode

```
import React, { useEffect, useState } from "react";

export default function ThemeToggle() {
  const [dark, setDark] = useState(() => {
    return localStorage.getItem("theme") === "dark";
  });

  useEffect(() => {
    if (dark) {
      document.documentElement.classList.add("dark");
      localStorage.setItem("theme", "dark");
    } else {
      document.documentElement.classList.remove("dark");
      localStorage.setItem("theme", "light");
    }
  }, [dark]);
}

return (
  <button
    onClick={() => setDark(!dark)}
    className="px-3 py-1 border rounded-md text-sm hover:bg-gray-200 dark:hover:bg-gray-700"
  >
  {dark ? "🌙 Dark" : "☀️ Light"}
  </button>
);
}
```

- **Timer, ProgressBar, OptionButton** – essential components for quiz-taking UI
- **Charts** (BarGraph, PieChart) – display performance analytics

These components ensure reusability and UI consistency.

## 5. Context Providers (Global State Management)

The app imports Context APIs to handle global data:

- **AuthContext** → login, register, admin login, logout
- **QuizContext** → CRUD operations for quizzes & questions (JSON Server)
- **LocalStorageContext** → stores each quiz attempt history

- **SoundContext** → handles click/correct/wrong sound effects

```

/* eslint-disable react-refresh/only-export-components */
import React, { createContext, useContext, useEffect, useState } from "react";
import axios from "axios";

const QuizContext = createContext();
export const useQuiz = () => useContext(QuizContext);

export function QuizProvider({ children }) {
  const API_URL = "http://localhost:3001/quizzes";

  const [quizzes, setQuizzes] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState("");

  /* -----
   | | USER ANSWERS (LOCAL STORAGE)
   ----- */
  const [answers, setAnswers] = useState(() => [
    const saved = localStorage.getItem("triviatrek_answers");
    return saved ? JSON.parse(saved) : {};
  ]);

  useEffect(() => {
    localStorage.setItem("triviatrek_answers", JSON.stringify(answers));
  }, [answers]);

  const saveAnswer = (quizId, attemptId, questionId, optionId) => {
    const key = `${quizId}-${attemptId}`;
    setAnswers((prev) => ({
      ...prev,
      [key]: {
        ...prev[key],
        [questionId]: optionId,
      },
    }));
  };
}

```

This allows data to be shared across all pages without prop drilling.

## 6. Admin Panel Files

The admin module imports:

- **AdminLayout** → wraps all admin routes
- **AdminSidebar, AdminTopbar** → administrative navigation
- **AdminDashboard** → analytics overview
- **ManageQuizzes, ManageQuestions** → CRUD pages for admins

- **AddQuizPage, EditQuizPage** → quiz creation and editing

```

import React from "react";
import { useQuiz } from "../context/QuizContext";
import {
  BarChart,
  Bar,
  XAxis,
  YAxis,
  Tooltip,
  ResponsiveContainer,
  PieChart,
  Pie,
  Cell,
  Legend
} from "recharts";

export default function AdminDashboard() {
  const { quizzes } = useQuiz();

  const totalQuizzes = quizzes.length;
  const totalQuestions = quizzes.reduce(
    (sum, quiz) => sum + quiz.questions.length,
    0
  );

  /* ----- PIE CHART DATA ----- */
  const pieData = quizzes.map((q) => ({
    name: q.title,
    value: q.questions.length,
  }));
  const COLORS = [
    "#4F46E5",
    "#0EAE9",
    "#1B8981",
    "#F59E0B",
    "#EF4444",
    "#A855F7",
    "#1488A6",
    "#F43F5E",
  ];
  /* ----- BAR GRAPH DATA ----- */
  const barData = quizzes.map((quiz) => ({
    name: quiz.title.length > 12 ? quiz.title.slice(0, 10) + "..." : quiz.title,
    questions: quiz.questions.length,
  }));

  return (
    <div className="p-8">
      <h1 className="text-3xl font-bold text-primary dark:text-indigo-300 mb-6">
        Admin Dashboard
      </h1>

      {/* Stat Cards */}
      <div className="grid grid-cols-1 md:grid-cols-2 gap-6 mb-10">
        {/* Total Quizzes */}
        <div className="bg-white dark:bg-gray-800 p-6 rounded-md shadow-md">
          <h2 className="text-xl font-semibold text-gray-800 dark:text-gray-200">
            Total Quizzes
          </h2>
          <p className="text-4xl font-bold text-indigo-500 mt-2">
            {totalQuizzes}
          </p>
        </div>

        {/* Total Questions */}
        <div className="bg-white dark:bg-gray-800 p-6 rounded-md shadow-md">

```

- **AddQuestion, EditQuestion** → question creation and editing

These enable complete backend-like quiz management directly through JSON server.

## 7. Global Styles

Finally, the app loads global CSS:

- **index.css / App.css** – base styles, TailwindCSS utilities, theme styles

```
@tailwind base;
@tailwind components;
@tailwind utilities;

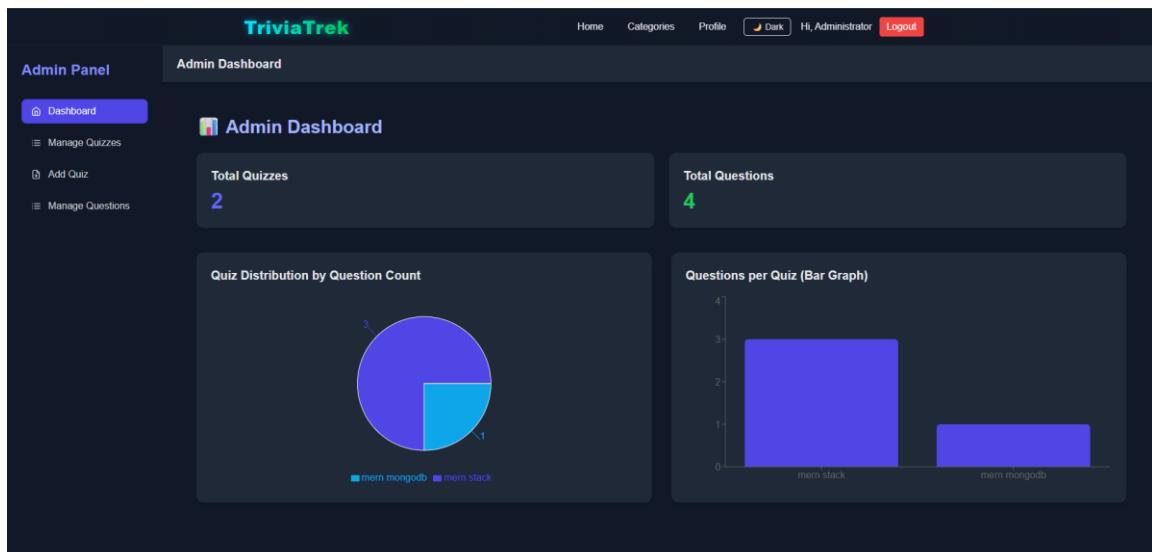
html, body, #root {
  height: 100%;
  background-color: hsl(var(--background));
}

body {
  @apply bg-gray-100 text-gray-900;
  font-family: Inter, sans-serif;
}
.dark {
  --background: 222.2 84% 4.9%;
```

This ensures a consistent design across all pages.

## Output:

### Admin panel outputs:



### + Add New Quiz

Quiz Title  
Enter quiz title...

Description  
Enter quiz description...

Category  
Enter category (e.g., C Programming)

Time Limit (in seconds)  
60

Create Quiz Cancel

## Manage Quizzes

**C Language Comprehensive Quiz**  
A set of 30 multiple-choice questions to practice C programming skills covering basics, data types, control structures, functions, arrays, pointers, and more.

Edit Questions Delete

**+ Add Quiz**

[Dashboard](#)

[Manage Quizzes](#)

[Add Quiz](#)

[Manage Questions](#)

### Edit Quiz

Quiz Title  
C Language Comprehensive Quiz

Description  
A set of 30 multiple-choice questions to practice C programming skills covering basics, data types, control structures, functions, arrays, pointers, and more.

Update Quiz

**Admin Panel**

**Admin Dashboard**

**C Language Comprehensive Quiz — Manage Questions**

**Add Question**

Which of the following is a correct data type in C?

- int
- symbol
- underscore
- doubledouble

**Edit** **Delete**

What is the size of an int data type in most systems?

- 2 bytes
- 4 bytes
- 8 bytes
- 1 byte

**Edit** **Delete**

Which keyword is used to declare a variable?

- var
- let
- int
- define

**Edit** **Delete**

What does printf() function do?

- Reads input
- Prints output
- Declares variables
- Loops

**Edit** **Delete**

**TriviaTrek**

Home Categories Profile **Dark** Hi, Administrator Logout

# TriviaTrek

Journey through a world of questions — fun, fast, and beautiful.

**Start Trek** **Browse**

**Quiz World**

**Why TriviaTrek?**

**Interactive Fun**  
Play beautifully designed quizzes with smooth animations.

**Track Progress**  
Monitor attempts, scores, charts, and performance easily.

**Grow Knowledge**  
Learn while enjoying hundreds of fun and fast questions.

**TriviaTrek**

Home Categories Profile **Dark** Hi, Administrator Logout

## Choose a Quiz Category

**General Knowledge Quiz**  
Test your knowledge about world facts, geography, history and more.  
40 Questions **Start**

**Science Fundamentals**  
Test your knowledge of basic science concepts across physics, chemistry, and biology.  
40 Questions **Start**

**Artificial Intelligence & Machine Learning**  
Comprehensive quiz on AI and ML concepts, algorithms, and applications.  
40 Questions **Start**

**Data Structures and Algorithms**  
Comprehensive quiz on data structures, algorithms, and computational complexity.  
40 Questions **Start**

**Computer Networks**  
Comprehensive quiz on computer networking concepts, protocols, and architectures.  
40 Questions **Start**

**Operating Systems**  
Comprehensive quiz on operating system concepts, processes, memory management, and file systems.  
40 Questions **Start**

**Amazon Web Services (AWS)**  
Comprehensive quiz on AWS cloud services, architecture, and best practices.  
40 Questions **Start**

**Full Stack Development**  
Comprehensive quiz on full stack web development including frontend, backend, databases, and DevOps.  
40 Questions **Start**

**TriviaTrek**

Home Categories Profile  Dark Hi, Administrator Logout

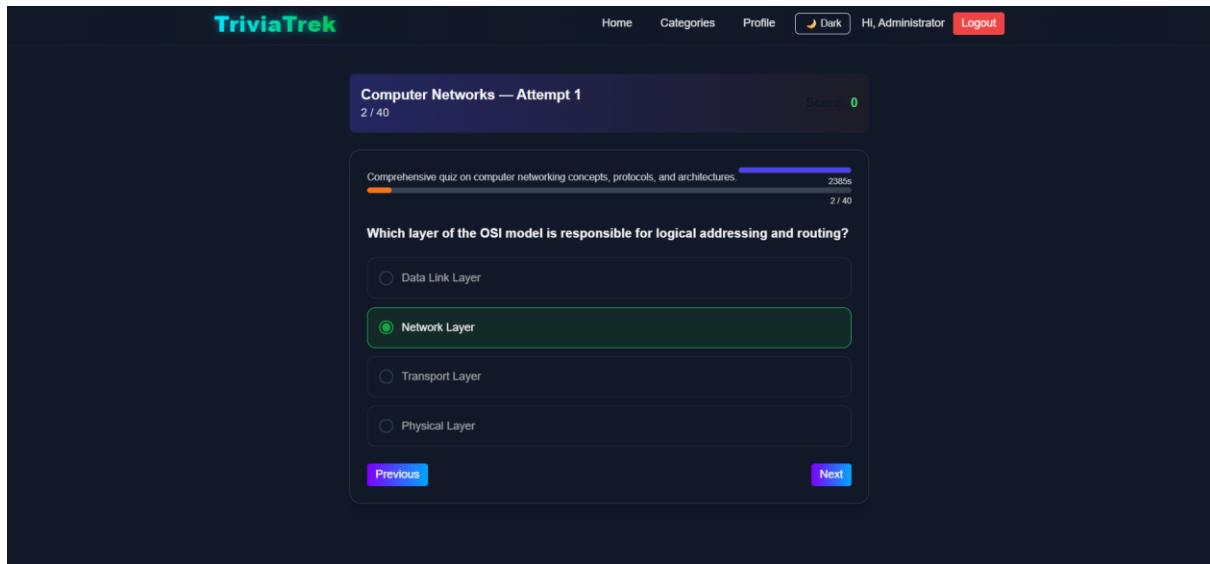
**Computer Networks — Attempt 1** Score: 0  
2 / 40

Comprehensive quiz on computer networking concepts, protocols, and architectures. 23855  
2 / 40

Which layer of the OSI model is responsible for logical addressing and routing?

Data Link Layer  
 Network Layer  
 Transport Layer  
 Physical Layer

[Previous](#) [Next](#)



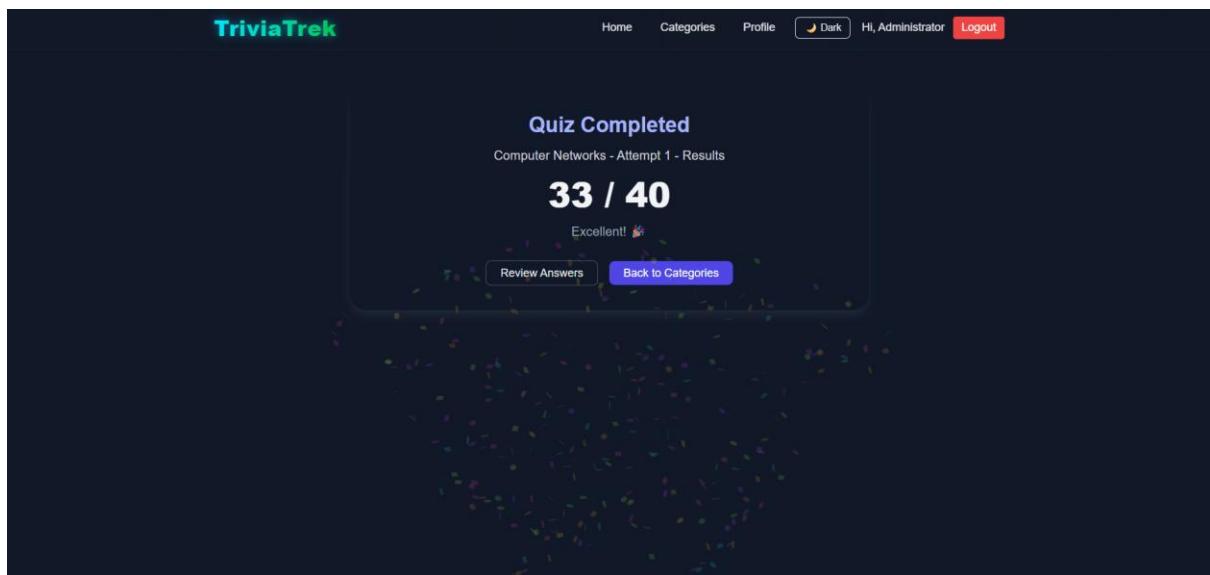
**TriviaTrek**

Home Categories Profile  Dark Hi, Administrator Logout

**Quiz Completed**  
Computer Networks - Attempt 1 - Results

**33 / 40**  
Excellent! 🎉

[Review Answers](#) [Back to Categories](#)



## Your Profile

### Quiz Performance

14      33  
Attempts      Best Score

### Quiz History

Computer Networks - Attempt 1 12/7/2025	33 / 40	<a href="#">Review</a>	<a href="#">PDF</a>
mem stack - Attempt 3 12/4/2025	1 / 3	<a href="#">Review</a>	<a href="#">PDF</a>
mem stack - Attempt 2 12/4/2025	0 / 3	<a href="#">Review</a>	<a href="#">PDF</a>
mem stack - Attempt 1 12/4/2025	2 / 3	<a href="#">Review</a>	<a href="#">PDF</a>
qoepiuw - Attempt 5 12/4/2025	1 / 2	<a href="#">Review</a>	<a href="#">PDF</a>
qoepiuw - Attempt 4 12/4/2025	0 / 1	<a href="#">Review</a>	<a href="#">PDF</a>
qoepiuw - Attempt 3 12/3/2025	1 / 1	<a href="#">Review</a>	<a href="#">PDF</a>
qoepiuw - Attempt 2 12/3/2025	0 / 1	<a href="#">Review</a>	<a href="#">PDF</a>
qoepiuw - Attempt 1 12/3/2025	1 / 1	<a href="#">Review</a>	<a href="#">PDF</a>
C Language - Attempt 4 12/3/2025	1 / 2	<a href="#">Review</a>	<a href="#">PDF</a>
data structure - Attempt 1 12/2/2025	1 / 1	<a href="#">Review</a>	<a href="#">PDF</a>
C Language - Attempt 3 12/2/2025	2 / 2	<a href="#">Review</a>	<a href="#">PDF</a>
C Language - Attempt 2 11/30/2025	2 / 2	<a href="#">Review</a>	<a href="#">PDF</a>
C Language - Attempt 1 11/30/2025	0 / 1	<a href="#">Review</a>	<a href="#">PDF</a>

### Score Comparison (Bar Chart)



### Overall Performance (Pie Chart)



[Home](#)