

TriviaTrek: Journey through a World of Questions

A Project Report

Submitted by

Lingamaneni Ruthvik - AP23110010131

Yarra Khyathisree - AP23110010215

Bikki Bindu Venkata Priya - AP23110010365

Manchipalli Jahnavi - AP23110011379

Sumbitted To

Mr.Yatharth

In partial fulfilment for the requirements of the full stack development project

**BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRM UNIVERSITY-AP

NEERUKONDA

MANAGALAGIRI - 522503

ANDHRA PRADESH, INDIA

DECEMBER-2025

INTRODUCTION

TriviaTrek is an interactive, full-stack quiz-based web application designed to deliver an engaging, educational, and entertaining experience for users of all age groups. The platform allows players to test their knowledge in a wide array of categories: Science, History, Movies, Technology, Geography, Sports, and General Knowledge. Each of the quizzes has three difficulty levels: Easy, Medium, and Hard. Such organization allows both beginners and advanced learners to enjoy the challenge corresponding to their level. With its intuitive, responsive interface, TriviaTrek is optimized to seamlessly deploy across devices, maintaining a seamless flow between questions-and-answers, using timers, instantaneous feedback mechanisms, and clear navigation.

The application is designed for personalization and long-term engagement by allowing for user authentication, detailed progress tracking, and milestone-based achievements. Users will be able to securely sign up, log in, and access personalized dashboards displaying performance metrics on total quizzes taken, best scores, accuracy, and recently earned badges. TriviaTrek also features a global leaderboard system that fosters healthy competition among its users. Individuals can compare their performance globally or within particular categories, creating a lively and motivating learning environment. This achievement system enhances user interaction by awarding badges in recognition of milestones, such as answering questions quickly, getting perfect scores, or mastering certain categories over time.

From a technical point of view, TriviaTrek demonstrates the concepts of full-stack development with its structured architecture and efficient implementation. The frontend includes React.js with Vite for fast development builds, Tailwind CSS for responsive styling, and Axios for API communications. The simulated RESTful API environment on the backend side is handled by JSON Server, which manages users, questions, results, and leaderboard data. These bring together the capability to demonstrate key real-world development practices: routing, authentication, CRUD operations, state management, and gamification. All in all, TriviaTrek will be both an engaging quiz platform for users and a strong technical model for students and developers interested in full-stack web application development.

SCENARIO BASED INTRODUCTION

Picture a student getting ready for competitive exams or even just someone who wishes to challenge their general knowledge after a long day. Instead of tediously scrolling through PDFs or static questionnaires, they launch TriviaTrek—a bright, dynamic, and intuitive quiz platform that makes learning interactive and enjoyable. The minute they log in, a neat dashboard displays quiz categories, difficulty modes, recent achievements, and leaderboard highlights, setting the stage for a personalized quiz experience.

The user selects a preferred category — for example, Science — and TriviaTrek fetches a fresh set of questions from the backend right away. A countdown timer appears as the quiz begins, creating a sense of excitement and urgency. Each question displays four neatly designed options that allow the user to think, evaluate, and respond. Behind the scenes, TriviaTrek silently records the user's choices, calculates accuracy, and tracks speed for future achievement rewards. The seamless transitions, instantaneous updates, and responsive design make the quiz feel so engrossing and fluid on any device.

With each progression of the quiz, the user gets points for each correct answer and gradually unlocks harder difficulty levels. Badges like "Perfect Score" are awarded upon completing a quiz flawlessly, while achievements such as "Marathon Player" can be received upon the user playing the quiz regularly. After completion, the user is greeted with a detailed performance summary that includes correct/incorrect answers to be reviewed, percentage of score, time taken, and the overall accuracy of responses. They can compare their ranking with others on both the global and category-specific leaderboards.

Meanwhile, administrators can log into their interface, where they can manage questions, add new categories, update difficulty levels, and maintain the overall quality and diversity of the quiz database. Ensuring that TriviaTrek remains fresh, relevant, and continuously improving for all users.

TriviaTrek turns a simple quiz into a relevant, game-oriented, and rewarding experience that blends learning and entertainment in such a manner as to make the user eager to improve a bit every day..

Target Audience

TriviaTrek is designed for diverse users:

Students & Competitive Exam Aspirants

- Practice subject-wise quizzes
- Improve speed and accuracy
- Track progress over time

Knowledge Enthusiasts & Gamers

- Fun quiz challenges
- Unlock achievements and badges
- Compete on leaderboards

Educators & Trainers

- Use quizzes as learning tools
- Analyze student progress

Project Goals & Objectives

TriviaTrek aims to showcase complete full-stack development concepts:

Frontend Goals

- Build a clean, responsive, and intuitive UI
- Implement dynamic quiz screens with timers
- Provide smooth routing and transitions
- Display real-time progress, feedback, and stats

Backend Goals

- Create secure REST APIs for user, quiz, and leaderboard data
- Implement authentication & authorization
- Manage quiz logic and scoring
- Perform data validation and error handling

Database Goals

- Store users, questions, categories, results, and achievements
- Track long-term performance and quiz history
- Support filtering by category, difficulty, and user

TriviaTrek demonstrates real-world application structure and data flow from UI → API → DB and back.

Modern Tech Stack

Frontend

- React.js
- React Router
- Tailwind CSS
- Axios
- lucide-react
- Vite
- PostCSS

Backend

- JSON Server (Mock REST API)
- Custom routes via routes.json

Database

- db.json (JSON-based mock database)

Build & Dev Tools

- Concurrently
- ESLint
- Vite build tools
- Node.js + npm

Key Features

1. Category-Based Quiz Selection

Users can explore quizzes across well-defined categories such as Science, History, Movies, Technology, Geography, Sports, and General Knowledge. Each category displays options for selecting difficulty levels (Easy, Medium, Hard), ensuring a customized quiz experience that matches the user's skill level.

2. Difficulty-Level Based Quiz Flow

Every category supports structured difficulty tiers.

- Easy – simple questions for beginners
- Medium – intermediate conceptual questions
- Hard – advanced or competitive-level questions

The system fetches questions dynamically based on the selected category and difficulty, ensuring variety and progression

3. Interactive Quiz Gameplay Interface

- Smooth, responsive UI built using React.js
- Timed questions with countdown animation
- Four-option MCQ layout for every question
- Automatic answer validation and score calculation
- Instant navigation to the next question

This interactive interface ensures a fast, engaging, and distraction-free quiz experience.

4. Quiz Summary & Review System

After finishing a quiz, the user is presented with a detailed rundown, including:

- Final score
- Correct and incorrect answers
- Total time taken
- Percentage accuracy

Users may optionally review all questions again with correct answers highlighted, enabling deeper learning and self-evaluation.

5. User Authentication (Login & Signup)

Login and registration pages communicate with the JSON Server backend to store and validate user accounts.

Includes:

- Email existence check
- Password validation
- Secure storage in the users collection

User login status and account details are managed through React state, ensuring consistent access across pages.

6. Profile Management Dashboard

Logged-in users can view personalized information such as:

- Username and email
- Total quizzes attempted
- Best scores
- Category-wise performance
- Earned achievements

All data is fetched directly from db.json to ensure reliability and consistency in progression tracking.

7. Badges & Achievement System

TriviaTrek uses gamification to boost user engagement:

- Fast Thinker – answering in under 5 seconds
- Perfect Score – scoring 10/10
- Marathon Player – completing 10 quizzes
- Category mastery badges (e.g., Science Sage, History Explorer)

Achievements appear visually on the user's profile and motivate continued learning.

8. Global Leaderboards

TriviaTrek includes competitive ranking features:

- Overall ranking across all categories
- Category-specific leaderboards
- Recent performance ranking

Sorting is done using criteria such as score, accuracy, and completion time. This creates a competitive and motivating environment for players.

9. Admin Panel (Content Management)

Admin users can manage the question bank through CRUD operations:

- Add new questions
- Edit existing ones
- Update correct answers
- Delete outdated or wrong questions

Admins can also manage categories and adjust difficulty structures, ensuring content stays relevant and high-quality.

10. Responsive & Device-Friendly UI

- Tailwind CSS ensures smooth rendering across mobile, tablet, and desktop
- Consistent spacing, readable typography, and flexible layout
- Ideal for students who prefer learning from phones or laptops

The entire interface adapts seamlessly to different screen sizes.

11. Mock Backend Powered by JSON Server

JSON Server powers the backend with fast mock API functionality. Stored datasets include:

- Users
- Questions
- Categories
- Results
- Leaderboard data

Supports:

- GET (fetch quizzes, users, results)
- POST (store new users, results)
- Custom routing using routes.json

This setup allows realistic backend simulation during development.

12. Developer-Friendly Setup

- Vite enables ultra-fast builds and dev server
- Axios handles all REST API communication
- Clean folder structure improves readability and feature scalability

Overall, the project is easy to maintain, debug, and extend.

PRE-REQUISITES

Before running the TriviaTrek Quiz Application, certain tools, software, and technical knowledge are required. Since this project uses React.js for the frontend, Tailwind CSS for styling, Axios for API calls, and JSON Server as a lightweight mock backend, the setup remains simple, fast, and beginner-friendly. No heavy backend frameworks or SQL databases are needed.

1. Code Editor (VS Code Recommended)

A modern code editor is required to build, edit, and run the application. Recommended: Visual Studio Code (VS Code)

VS Code provides:

- Built-in support for JSX and JavaScript ES6+
- Syntax highlighting and IntelliSense for React
- Integrated terminal for running the development and backend servers
- Easy debugging features

2. Web Browser (Chrome / Firefox)

A modern browser is needed to run and debug the React application.

Recommended: Google Chrome

- Comes with powerful DevTools
- Supports React Developer Tools extension

Other compatible browsers

- Firefox
- Microsoft Edge
- Brave

These browsers can be used for UI testing, responsiveness checks, and performance monitoring.

3. Node.js & npm (Mandatory for React + Vite)

TriviaTrek relies heavily on Node.js for managing dependencies and running the development environment.

Node.js provides:

- JavaScript runtime needed to execute Vite and React scripts
- Environment for bundling, building, and serving the app

npm provides:

- Installation of all project dependencies including:
 - React
 - React Router
 - Axios
 - JSON Server
 - Tailwind CSS
 - lucide-react
 - Vite
 - ESLint
 - PostCSS & Autoprefixer

Without Node.js and npm, TriviaTrek cannot run.

4. React.js Framework Knowledge

Since TriviaTrek is built fully using React, developers should have a basic understanding of:

- Functional components & JSX
- React Hooks
 - useState
 - useEffect
- Props & component communication
- Conditional rendering
- Rendering lists (map)
- Event handling
- Component reusability

5. React Router (For Page Navigation)

TriviaTrek uses React Router for handling navigation between multiple views such as:

- Login
- Signup
- Quiz Selection
- Quiz Page
- Quiz Result
- Leaderboard
- Admin Panel
- Profile Page

Developers should understand route creation, dynamic URL parameters, and nested routing.

6. Axios (For API Requests)

TriviaTrek uses Axios to communicate with JSON Server.

Developers should know how to:

- Perform GET, POST, PUT, PATCH, DELETE operations
- Handle API responses and errors
- Work with async/await syntax
- Manage loading and error states

Axios is used widely across QuizPage, AdminPage, ProfilePage, and LeaderboardPage.

7. JSON Server (Local Mock Backend)

JSON Server acts as the backend for TriviaTrek.

Developers should know how to:

- Start JSON Server using:
- `json-server --watch public/db.json --port 3001 --routes public/routes.json`
- Understand the structure of db.json, which contains:
 - users
 - questions
 - categories
 - quiz results
 - leaderboard data

JSON Server Responsibilities:

- Acts like a real REST API
- Provides instant data retrieval
- Handles CRUD operations for admin features
- Makes the project easy to test and extend

8. Tailwind CSS (Styling Framework)

TriviaTrek uses Tailwind CSS for designing responsive UI components.

Developers should understand:

- Utility classes
- Responsive breakpoints
- Flexbox/Grid using Tailwind
- Customizing Tailwind config if required

Tailwind helps ensure consistency, speed, and professional UI design.

9. Local Development Server

Vite provides a fast development server for running the frontend.

To start the project:

```
npm run dev
```

This simultaneously runs:

- The front-end (React + Vite)
- JSON Server backend.

10. Basic Web Development Knowledge

Even with powerful tools, developers must understand the fundamentals:

HTML & JSX

- Semantic structure
- Component-based UI

CSS / Tailwind

- Styling layouts
- Positioning and spacing

- Responsive design

JavaScript ES6+

- Arrow functions
- Spread / rest operators
- Modules and imports
- Async/await and Promises
- Array methods (map, filter, reduce)

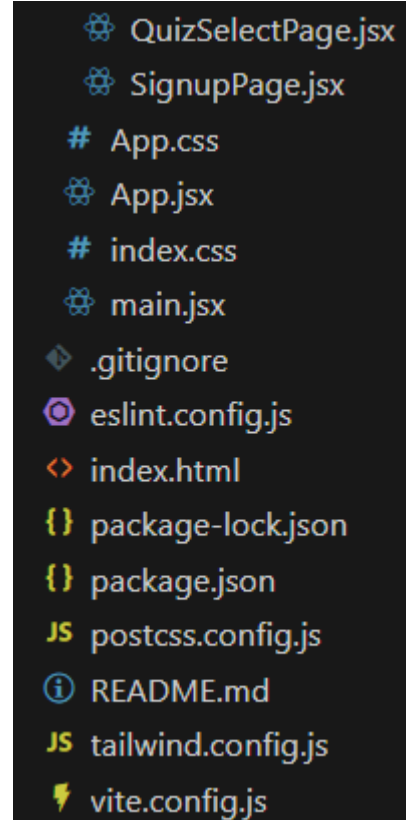
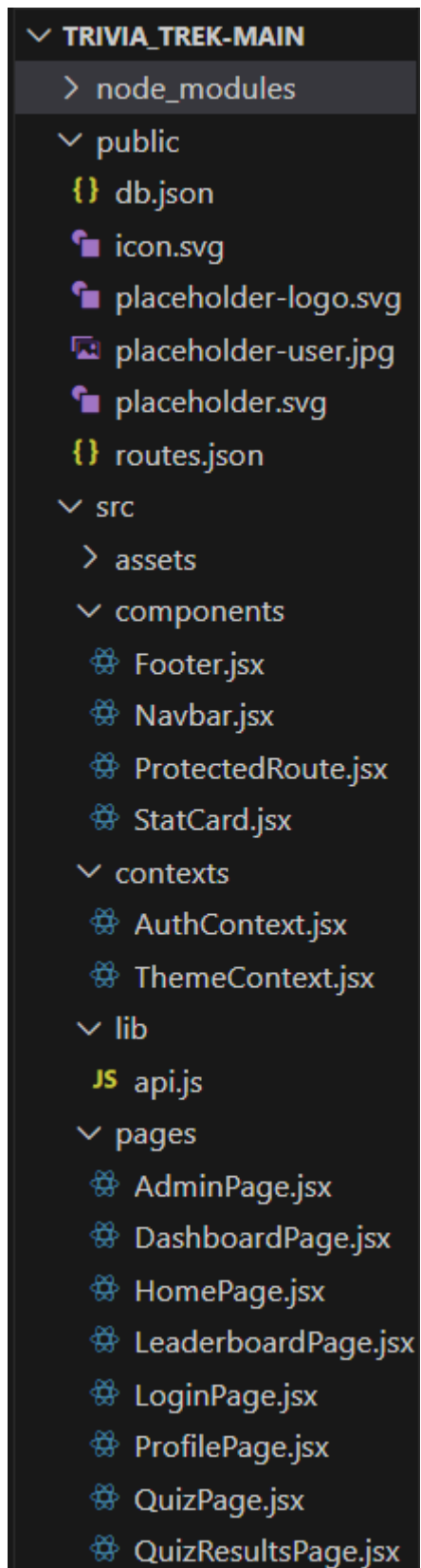
11. System Requirements

To run TriviaTrek smoothly:

- Minimum 4GB RAM (8GB recommended)
- Dual-core processor
- 2–5GB of free disk space
- Stable Internet connection (for installing npm packages)

These specs ensure fast builds, smooth UI, and quick server responses.

PROJECT STRUCTURE



PROJECT FLOW

Project demo:

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

Demo link:

<https://drive.google.com/file/d/1C1GhnzqKljk-1kwn27P3QCDyyxjic1kY/view?usp=sharing>

Code Explanation:

https://drive.google.com/file/d/1xBoPlqWjwR9BmRCpDCYO_HceA07eVkt6/view?usp=sharing

Code Description

The **App.jsx** file serves as the central routing configuration for the TriviaTrek application. It defines how different pages and UI components load based on the URL, ensuring smooth navigation throughout the quiz platform. This file acts as the backbone of the application's navigation system, linking together all major pages such as Login, Signup, Quiz, Results, Leaderboard, Admin Panel, and more.

1. Imports

The file begins by importing essential React Router modules along with project-specific pages.

React Router Imports

- Routes and Route — used to define navigation paths throughout the application.

Layout Component

- The Layout component provides the common UI structure shared across multiple pages, such as:
 - Navbar
 - Footer
 - Main Content Wrapper

Any page placed inside this layout automatically inherits these shared UI elements.

Page Components

These components represent different functional screens of the TriviaTrek application:

- *HomePage* — Overview of quiz categories and navigation options

- *LoginPage* — User login screen
- *SignupPage* — New user registration
- *QuizSelectPage* — Category & difficulty selection page
- *QuizPage* — Displays questions, options, timers, and handles gameplay
- *QuizResultsPage* — Shows score, accuracy, time, and review options
- *LeaderboardPage* — Displays global and category-wise rankings
- *ProfilePage* — Shows user details, badges, and quiz history
- *DashboardPage* — Logged-in user dashboard
- *AdminPage* — Allows administrators to manage quiz content (CRUD operations)

Each component represents a dedicated section of the TriviaTrek interface.

2. *App Component Overview*

The **App()** function maps each URL to its corresponding page using `<Routes>` and `<Route>`. It is responsible for:

- Defining which page appears for each URL
- Handling protected vs. public pages
- Structuring routes inside and outside the Layout
- Managing dynamic quiz paths like category and difficulty

This forms the core navigation logic of the application.

3. *Layout-Based Routing*

All primary user-facing screens are wrapped inside:

```
<Route path="/" element={<Layout />}>
```

Pages inside this block automatically display the Navbar, Footer, and main layout design.

Routes Inside Layout

- `"/"` → *HomePage*
- `"/dashboard"` → *DashboardPage*
- `"/profile"` → *ProfilePage*
- `"/leaderboard"` → *LeaderboardPage*
- `"/quiz/select"` → *QuizSelectPage*
- `"/quiz/:category/:difficulty"` → *QuizPage (dynamic)*
- `"/results"` → *QuizResultsPage*
- `"/admin"` → *AdminPage*
- `"*"` → *PageNotFound (fallback for invalid URLs)*

These routes inherit the common layout and maintain consistent UI structure.

4. Standalone Routes (Without Layout)

Certain pages do not require the Navbar or Footer, especially authentication pages.

Standalone Pages:

- `"/login" → LoginPage`
- `"/signup" → SignupPage`

These are kept outside the Layout wrapper for a clean and distraction-free user experience.

5. Dynamic Routes Explained

TriviaTrek uses dynamic URL parameters to handle quizzes efficiently:

- `"/quiz/:category/:difficulty"`
Loads quiz questions based on selected category and difficulty level.

This dynamic structure allows scalability and supports unlimited question sets.

Routing Summary

Main Routes

- Home → `/`
- Dashboard → `/dashboard`
- Profile → `/profile`
- Leaderboard → `/leaderboard`
- Quiz Selection → `/quiz/select`
- Quiz Gameplay → `/quiz/:category/:difficulty`
- Results → `/results`
- Admin Panel → `/admin`

Authentication Routes

- Login → `/login`
- Signup → `/signup`

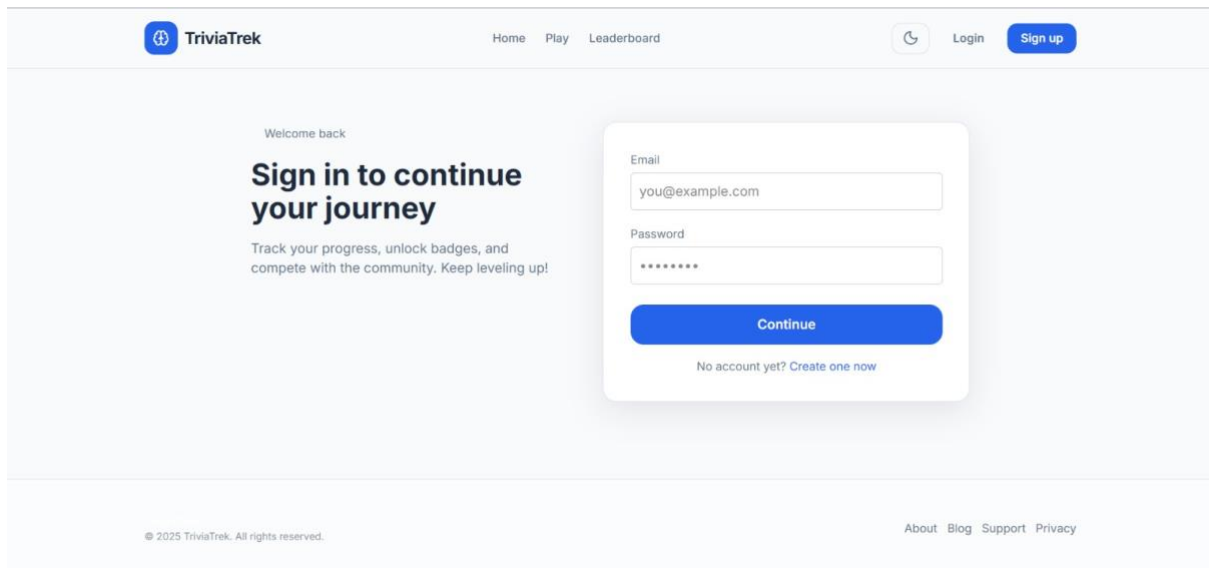
Fallback

- Page Not Found → `*`

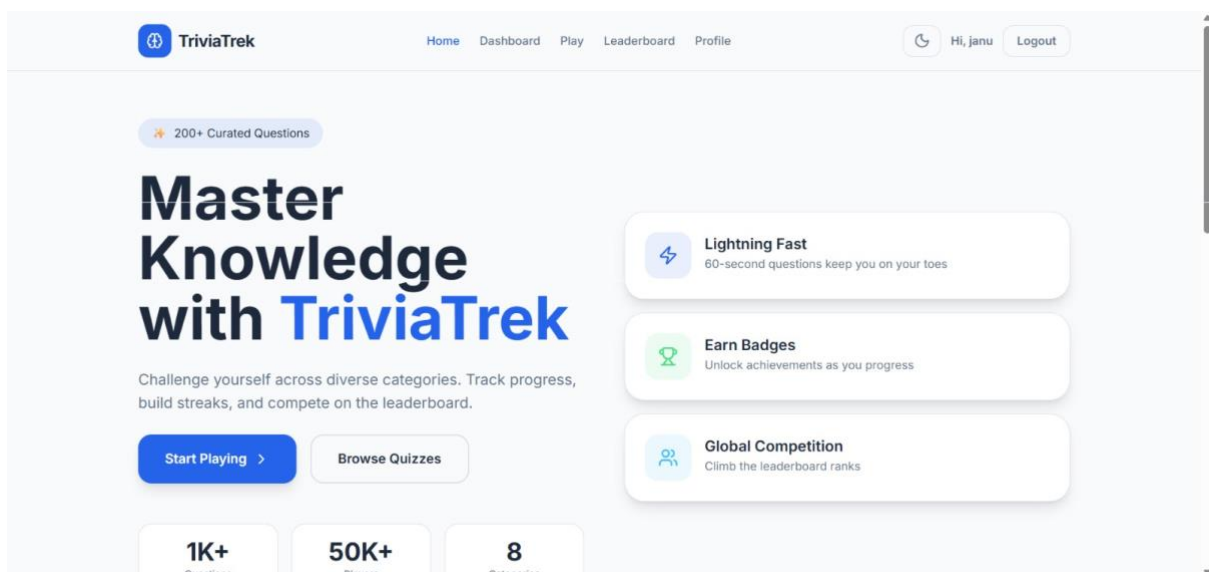
This routing configuration ensures smooth navigation from login to quiz gameplay, results, leaderboard, and admin management.

Project Execution

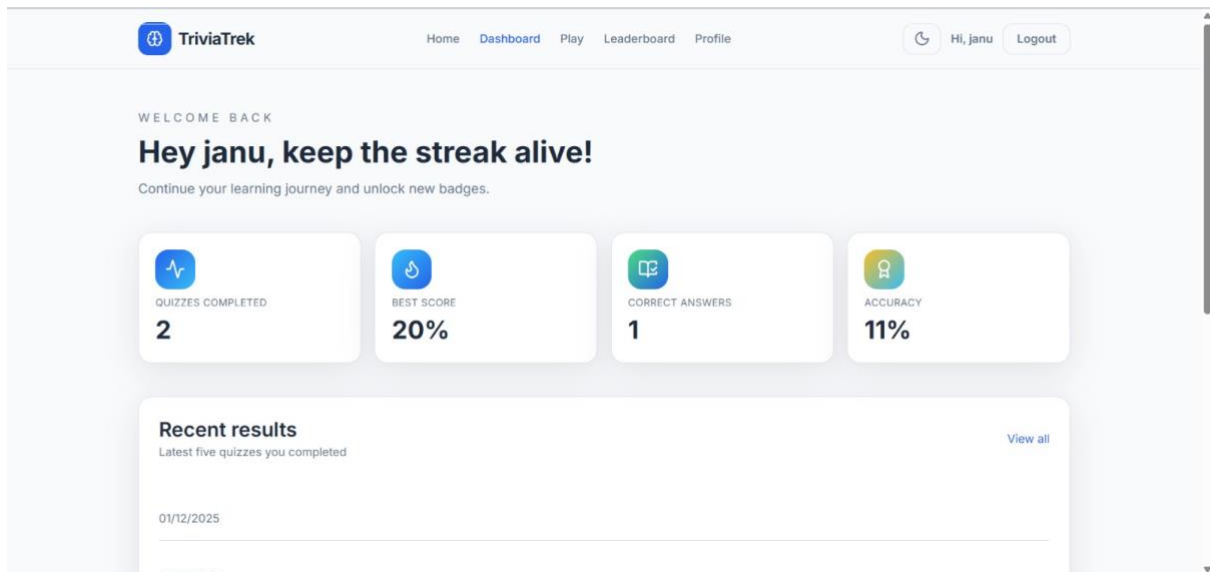
Login:



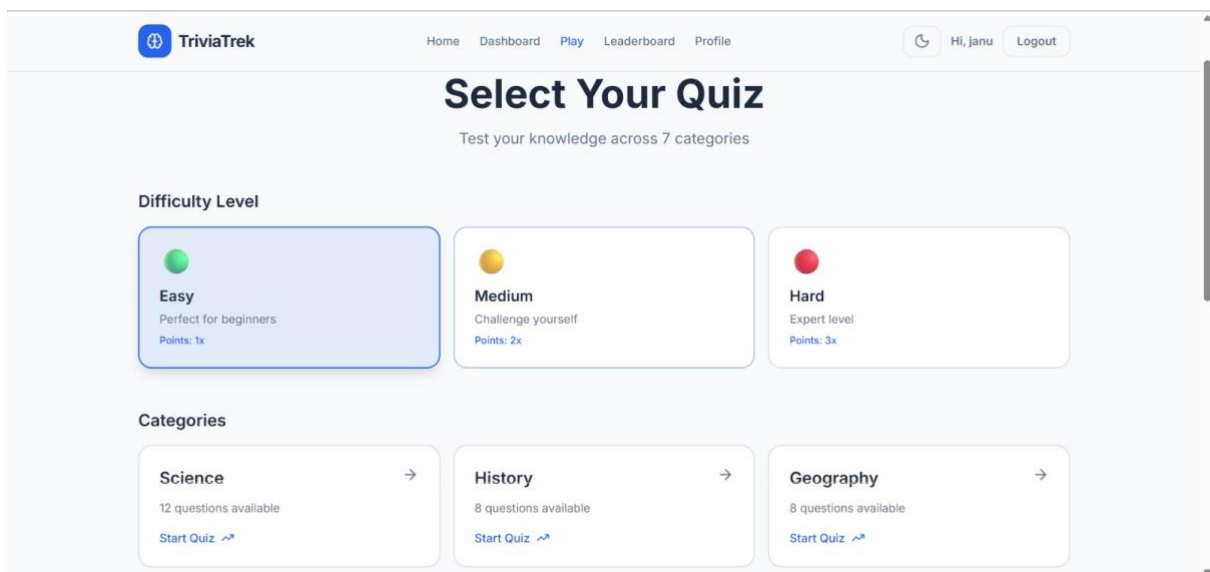
Home:



Dashboard:



Quiz Selection:



Quiz:

TriviaTrek

HomeDashboardPlayLeaderboardProfile

Hi, JanuLogout

Exit Quiz

QUIZ IN PROGRESS

Science · Easy

Question 1/5Time left: 57s

EASY DIFFICULTY

What is the chemical symbol for Gold?

A Au

B Ag

C Gd

LIVE SCORE

0/5

Accuracy 0%

0Correct

0Incorrect

Answered

QUESTION NAVIGATOR

1

2

3

4

5

Score:

TriviaTrek

HomeDashboardPlayLeaderboardProfile

Hi, JanuLogout

Quiz Complete!

Science · Easy Level

60%

3/5 Correct

Final You'll Score

CORRECT

3

INCORRECT

2

ACCURACY

Leaderboard:

TriviaTrek

HomeDashboardPlayLeaderboardProfile

Hi, JanuLogout

LEADERBOARD

Top explorers this week

Climb the leaderboard by achieving high accuracy.

TOP PLAYER

Jahnavi

Average score 83% across 4 quizzes

RANK	PLAYER	AVG. SCORE	RUNS
#1	Jahnavi	83%	4
#2	Ruthvik_I	71%	8
#3	bindu06	35%	3


Profile:

TriviaTrek


HomeDashboardPlayLeaderboardProfile


Hi, JanuLogout

EXPLORER PROFILE




janu
janu@gmail.com

janu@gmail.com



Joined 01/12/2025

Badges



First Step
Complete your first quiz