# **Web Dev Quiz - Documentation**

## **1. Overview**

**Web Dev Quiz** is an interactive quiz application designed to test and improve web development knowledge. It features:

* **Timed quizzes** on various topics (HTML, CSS, JavaScript, React, etc.).
* **AI-generated questions** using Google's Gemini API.
* **Study resources** with links to W3Schools tutorials.
* **Progress tracking** with score summaries and explanations.

## **2. Features**

### **Quiz Features**

✅ **Multiple Topics:** HTML, CSS, JavaScript, React, Python, Java, C#, C++, and Software Development Concepts.  
✅ **Difficulty Levels:** Easy, Medium, Hard.  
✅ **Timed Questions:** 60 seconds per question.  
✅ **Instant Feedback:** Correct/incorrect answers with explanations.  
✅ **Score Tracking:** Final results with performance breakdown.

### **Study Features**

📚 **Topic Selection:** Choose a topic to study.  
🔗 **Curated Resources:** Direct links to W3Schools tutorials, references, and examples.

### **Technical Features**

⚡ **React.js:** Frontend built with React.  
🤖 **Gemini API:** AI-powered question generation.  
🚀 **Netlify Deployment:** Hosted on Netlify for easy access.

## **3. Installation & Setup**

### **Prerequisites**

* Node.js (v16+).
* npm / Yarn.
* Google Gemini API Key (for question generation).

### **Steps to Run Locally**

1. **Clone the repository:**

bash

git clone <https://github.com/your-repo/web-dev-quiz.git>cd web-dev-quiz

1. **Install dependencies:**

bash

npm install

1. **Set up environment variables:**  
   Create a .env file:

env

VITE\_GEMINI\_API\_KEY=your\_api\_key\_here

1. **Run the app:**

bash

npm run dev

1. **Build for production:**

bash

npm run build

## **4. Deployment (Netlify)**

### **Manual Deployment**

1. **Build the app:**

bash

npm run build

1. **Drag & drop** the build folder into [Netlify](https://app.netlify.com/drop).

### **Automated Deployment (GitHub/GitLab)**

1. Connect your repo to Netlify.
2. Set:
   1. **Build command:** npm run build.
   2. **Publish directory:** build.
3. Add environment variables (if using Gemini API).

## **5. File Structure**

text

src/  
├── App.js # Main app with routing  
├── About.js # About page  
├── Study.js # Study topics selection  
├── StudyTopic.js # Study resources per topic  
├── Footer.js # Footer component  
├── index.js # React entry point  
public/  
├── index.html # Base HTML  
.env # Environment variables

## **6. API Usage (Gemini AI)**

The app uses **Google's Gemini API** to generate quiz questions dynamically.

### **Prompt Structure**

js

`Generate 5 ${difficulty} level MCQs about ${topic} for web development.  
Format as JSON array with:  
{  
 "question": "Question text",  
 "options": ["Option1", "Option2", "Option3", "Option4"],  
 "correctAnswer": "CorrectOption",  
 "explanation": "Brief explanation"  
}`

## **7. Future Improvements**

🔹 **User Accounts:** Save progress and scores.  
🔹 **Leaderboard:** Compare scores with others.  
🔹 **More Topics:** Expand to backend (Node.js, SQL).  
🔹 **Dark/Light Mode:** Better theme support.