

Name: Seidl	<b>WMC</b>	Datum: 17.09.2024
Klasse: 3AHIF		Gruppe: Y
Lehrer: Michel	<b>Protokoll</b>	Bewertung:

## WEBSITE

### Trivia Quiz

#### Quiz App

Kategorie wählen:

Which punk rock band released hit songs such as "Californication", "Can't Stop" and "Under the Bridge"?

Foo Fighters

Linkin Park

Green Day

Red Hot Chili Peppers

Richtig!

Neue Frage

Wenn eine Falsche Antwort gewählt wurde:

### Trivia Quiz

#### Quiz App

Kategorie wählen:

What was the name of the first artificial Earth satellite, launched by the Soviet Union in 1957?

Sputnik 1

Soyuz 7K-OK

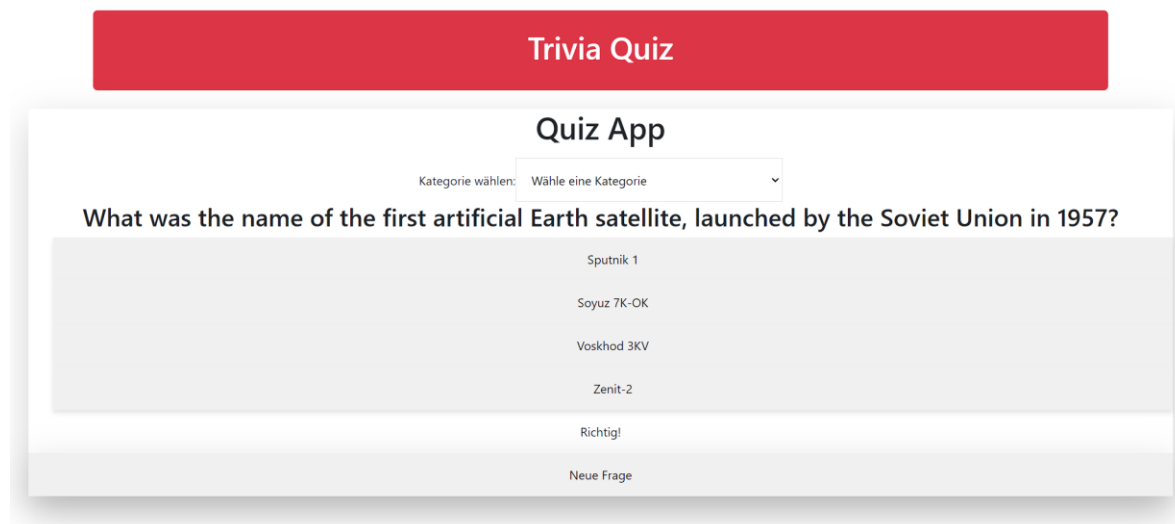
Voskhod 3KV

Zenit-2

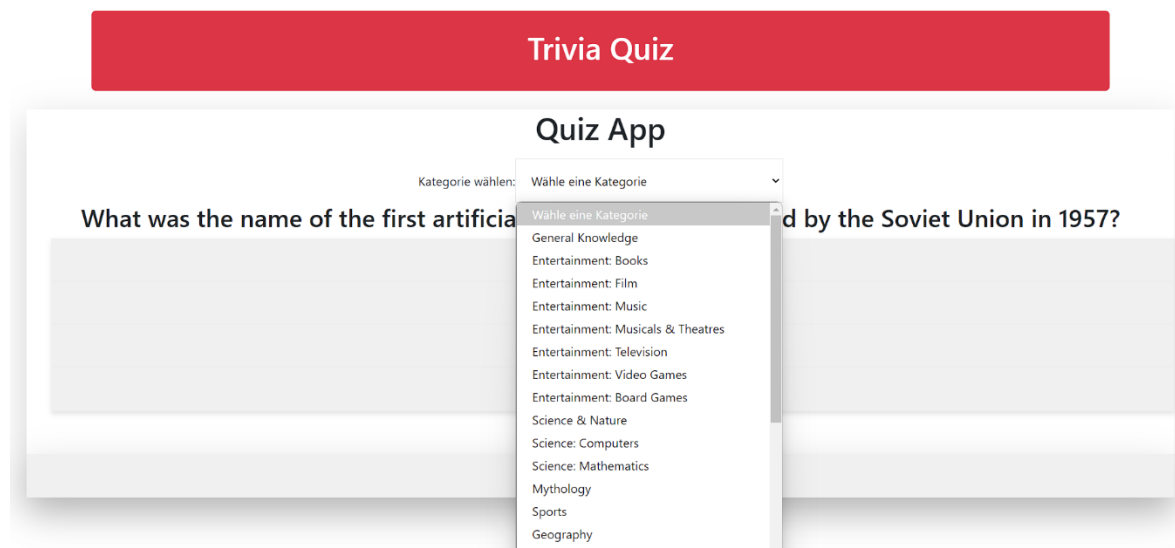
Falsch!

Neue Frage

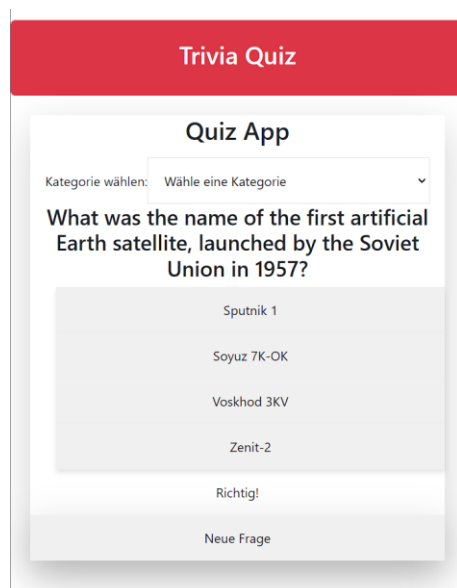
Wenn eine Richtige Antwort gewählt wird:



Wenn man das Kategorien Menu öffnet:



Mobile Ansicht:



## CODE:

Ich habe für die Website zum Stylen standart CSS und Tailwind verwendet. Jetzt werde ich ihnen ein paar der schwereren code snipits erklären.

```
<div class="min-h-screen bg-gradient-to-r from-blue-500 to-indigo-600 flex items-center"
  <div class="rounded-lg shadow-lg p-8 max-w-xl w-full">
    <h1 class="text-3xl font-bold text-center text-indigo-700 mb-6">Quiz App</h1>

    <!-- Category Dropdown -->
    <div v-if="categories.length" class="mb-6 text-center">
      <label for="category" class="block text-gray-700 font-medium mb-2">Kategorie wählen:
      <select
        v-model="selectedCategory"
        @change="loadQuestion"
        class="mt-2 block w-full p-3 border rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-500"
        <option value="" disabled>Wähle eine Kategorie</option>
        <option v-for="category in categories" :key="category.id" :value="category.id">
          {{ category.name }}
        </option>
      </select>
    </div>
```

```
methods: {
  async loadCategories() {
    const response = await fetch('https://opentdb.com/api_category.php');
    const data = await response.json();
    this.categories = data.trivia_categories;
  },
}
```

## Category Dropdown

- Diese Codesektion erstellt das Dropdownmenu, welches Usern erlaubt einen Schwierigkeitsgrad einstellen zu lassen.
- `v-if="categories.length"`: Stellt sicher das das Dropdown Menu nur gezeigt wird wen es gefüllt ist(also erfolgreich gefetcht wurde).
- **Label und Select Elements:**
  - `label for="category"`: Describes the select dropdown, labeled "Kategorie wählen".
  - `<select>`: Ist das Dropdownfeld bei welchem man die Kategorie auswählen kann.
    - `v-model="selectedCategory"`: Binds the value of the selected option to the selectedCategory data property.
    - `@change="loadQuestion"`: Triggers the loadQuestion method when a new category is selected.
    - `class="mt-2 ... focus:ring-2"`: Styling for padding, borders, and focus effect (highlighting when selected).

## Option Elements

- `<option v-for="category in categories" ...>`: Loops over the categories array and generates an option for each category.
  - `:key="category.id"`: A unique key for each option to help Vue optimize rendering.
  - `:value="category.id"`: Sets the value of each option to the category's ID.
  - `{{ category.name }}`: Displays the name of the category inside the option.

## Key Concepts:

- **Vue Directives:**
  - v-if: Conditionally renders an element.
  - v-model: Two-way data binding between the select input and the component's data.
  - v-for: Looping through an array to generate elements.
  - @change: Event binding for the select element.
- **Tailwind CSS Utility Classes:** Provide pre-built styles for spacing, layout, and appearance.

```
async loadQuestion() {
  const response = await fetch(
    `https://opentdb.com/api.php?amount=1&difficulty=easy&category=${this.selectedCategory}`
  );
  const data = await response.json();
  const questionData = data.results[0];
  this.question = questionData;
  this.shuffledAnswers = this.shuffleAnswers([questionData.correct_answer, ...questionData.incorrect_answers]);
  this.result = null;
},

shuffleAnswers(answers) {
  return answers.sort(() => Math.random() - 0.5);
},

checkAnswer(selectedAnswer) {
  if (selectedAnswer === this.question.correct_answer) {
    this.result = 'Richtig!';
  } else {
    this.result = 'Falsch!';
  }
},
},
```

## Laden der Frage

- Speichert fetch in die variable response
- (this.selectedCategory)= wenn eine Kategorie über geben ist das nur fragen mit diesen typen gefetched werden
- Nimmt die Fragen und fügt sie in einen Arrey ein und verwendet dann die shuffel methode um die richtigen und falschen Antworten zu mischen

## Fragen mischen

- Mischt die Answers mit einer Mathmethode und wird im laden der fragen verwendet

## Fragen checken

- Schaut nach ob die selected Answer richtig ist und returned true oder false danach ob die Antwort richtig oder nicht ist.

Größten Challenges:

Ich habe mir am Schwersten getan bei der Mischung des Arrays und beim selecten von denn Kategorien weil ich beim fetchen als erstes nicht wusste wie ich mache falls eine Kategorie ausgewählt wurde das ich nach dieser die richtigen fragen fetche