

# Напреден Веб Дизајн

## Репорт за развој на квиз апликација со Vue.js и Bootstrap

### Вовед

Креирањето на образовни и интерактивни апликации се развива како значаен тренд во софтверската индустрија. Едни од најпопуларните образовни апликации се Квиз Апликациите, кои овозможуваат корисниците да ги тестираат своите знаења и да се забавуваат истовремено. Во овој текст, ќе го разгледаме процесот на развој на Квиз Апликација користејќи го Vue.js и дизајнот кој го реализирав со помош на Bootstrap.

### Зошто Квиз Апликација?

Идејата зад развојот на Квиз Апликација е да се креира интерактивен и забавен начин за учење и развивање на знаење во ИТ областа. Со денешните брзи технолошки напредоци, знаењето во оваа област станува неопходно и потребно е константно обновување и усовршување на знаењата. Квиз апликациите се одличен начин да се поттикнат луѓето да учат и да се тестираат.

### Vue.js и Bootstrap: Зашто овие технологии?

#### Vue.js

Vue.js е лесен за учење и користење JavaScript framework кој нуди многу предности за развивачите. Неговата реактивна природа го прави одличен избор за изградба на современи веб апликации. Една од најголемите предности на Vue.js е што овозможува брз и ефикасен развој на кориснички интерфејси и интерактивни апликации.

Vue.js овозможува развивачите да го раздвојат кодот во компоненти, што го прави организиран и лесно одржуван. Со ова, може да се создаде компонентна архитектура за Квиз Апликацијата, каде што секое прашање, одговор и функционалност може да биде имплементирано како одделен компонент. Ова олеснува тестирање, подобрување и додавање на нови функции.

#### Bootstrap

Bootstrap е најпопуларниот CSS framework кој овозможува брз и ефикасен дизајн на веб апликации. Со помош на Bootstrap, можно е да се создаде елегантен и респонзивен дизајн, кој ќе ги задоволи потребите на различни корисници и уреди.

Дизајнот на Квиз Апликацијата ќе биде разработен со помош на Bootstrap компоненти, што ќе го разубави изгледот на Квиз Апликацијата. Со Bootstrap, може да се создаде современ и привлечен дизајн, кој ќе го направи корисничкото искуство удобно и пријатно.

## Функционалности на Квиз Апликацијата

### 1. Приказ на прашања

Квиз Апликацијата ќе овозможи приказ на прашањата едно по едно. Ова ќе обезбеди корисниците да ги видат и одговорот на прашањата во последователен редослед. Секое прашање ќе биде прикажано на еден дел од екранот, со соодветни контроли за избор на одговор.

### 2. Feedback

Секое одговарање на прашање ќе резултира со feedback до корисникот. Доколку корисникот избере точен одговор, ќе добие потврда за точниот одговор, можеби со дополнителен текст. Доколку избере грешен одговор, ќе добие информативна порака за грешката и правилниот одговор.

### 3. Ограничување на изборот

Квиз Апликацијата ќе го ограничи изборот на корисниците на само еден одговор. Ова гарантира дека корисниците нема да можат да даваат повеќе од еден одговор на исто прашање. Ова ќе го направи искуството на играње чесно и справедливо.

### 4. Score

Апликацијата ќе прикажува поени (score) на корисникот во реално време. Секое точно одговарање ќе му додаде поен, додека секое грешно одговарање ќе ги намали поенките. Ова го поттикнува корисникот да го подобрува својот резултат и да се натпреварува со себеси.

### 5. Тајмер

За секое прашање, апликацијата ќе има вграден тајмер. Оваа функционалност ќе ги изненади корисниците и ќе го подобри искуството на играње. Корисниците ќе имаат одредено време за одговор на прашањето, што ќе ги предизвика да бидат брзи и точни во своите одговори, а истовремено и да немаат време да препишуваат.

## 6. Дополнителни факти

При точно одговарање на прашање, апликацијата ќе ги прикаже дополнителни факти или информации поврзани со тоа прашање. Ова ги обогатува знаењето на корисниците и го прави искуството поинтересно и образовно.

## 7. Грешка при избор на одговор

Доколку корисникот избере грешен одговор, апликацијата ќе му прикаже порака со грешка и ќе му обезбеди информации за точниот одговор. Ова им овозможува на корисниците да научат од своите грешки и да ги подобрат своите знаења во ИТ областа.

## 8. Резултат и опција за повторно започнување

На крајот од играта, апликацијата ќе го прикаже резултатот на корисникот. Корисникот ќе има опција да ја започне играта од почеток, да ја подобре својата игра или да ја сподели својата резултат со други корисници.

# Имплементација на Кодот

## HTML и Bootstrap

Првиот дел од кодот вклучува HTML структура и користење на Bootstrap за создавање на модерен и респонзивен дизајн. Вградени се потребните мета-тагови и вградени се Bootstrap стилови преку CDN (Content Delivery Network).. Ова додава стилови, бои и класи за креирање на привлечен и современ интерфејс.

## Vue.js и Vue Instance

Кодот продолжува со внесување на Vue.js со имплементацијата на Vue instance. Vue instance се креира со помош на `new Vue({})` конструкторот и се поврзува со елементот со `el` опцијата.

Vue.js е моќна библиотека за развој на кориснички интерфејси која овозможува лесна и ефикасна манипулација на податоците и обновување на интерфејсот. Во овој код, Vue.js се користи за да се одржуваат сите релевантни состојби на апликацијата, како што се тековното прашање, изборите на корисникот и бројот на поени.

Пример од код

```

new Vue({
  el: '#app',
  data: {
    questions: [
      {
        question: 'What does "CPU" stand for in the context of computing?',
        options: ['Central Process Unit', 'Central Programming Unit', 'Central Processing Unit'],
        correctIndex: 2,
        image: 'https://img.freepik.com/free-vector/realistic-style-microchip-processor-background_52683-58932.jpg?w=1060&t=st=1694668487~exp=1694669087~hmac=4f9e4546e10a5200a3a5e38fee3af9fc6d59cecdeb092896d54e0e2a801b50b8',
        funFact: 'The Central Processing Unit (CPU) is the primary component of a computer that acts as its "control center."',
      },
    ],
  },

```

## Data и Методи

Долниот дел од кодот определува различни променливи и методи кои го управуваат состојбата и функционалноста на апликацијата.

**questions:** Оваа променлива содржи список со сите прашања во формат на објекти, каде секое прашање има текст на прашањето, опции за одговори, индекс на точниот одговор, слика и интересен факт.

**currentQuestionIndex:** Го чува индексот на моменталното прашање во листата со прашања.

**selectedOption:** Оваа променлива ги следи одбраните одговори од корисниците.

**score:** Го чува моменталниот резултат на корисникот.

**initialTime** и **timeLeft:** Ги следи времето за одговор на тековното прашање.

**showQuiz**, **started**, **showFunFact**, и **showErrorMessage:** Се користат за управување на приказот на различни делови од апликацијата, во зависност од тековната состојба.

**startQuiz()**, **selectOption(index)**, **nextQuestion()**, **restartQuiz()**, и **resetTimer()**: Овие методи се поврзани со настани како кликување на копчиња или изминување на времето и го управуваат текот на играта и одговорите на корисниците.

Дел од кодот:

```

currentQuestionIndex: 0,
  selectedOption: null,
  score: 0,
  initialTime: 10,
  timeLeft: 10,

```

```

    showQuiz: false,
    started: false,
    showingFunFact: false,
    showErrorMessage: false,
    timer: null, // Added timer variable
  },
  methods: {
    startQuiz() {
      this.started = true;
      this.showQuiz = true;
      this.resetTimer();
      this.startTimer();
    },
    startTimer() {
      this.timer = setInterval(() => {
        if (this.timeLeft > 0) {
          this.timeLeft--;
        } else {
          clearInterval(this.timer);
          if (this.selectedOption === null) {
            this.showingFunFact = true;
            setTimeout(() => {
              this.showingFunFact = false;
              this.nextQuestion();
            }, 5000);
          }
        }
      }, 1000);
    },
    selectOption(index) {
      if (this.selectedOption === null) {
        this.selectedOption = index;
        if (index === this.currentQuestion.correctIndex) {
          this.score++;
          this.showingFunFact = true;
          setTimeout(() => {
            this.showingFunFact = false;
            this.nextQuestion();
          }, 5000);
        } else {
          this.showErrorMessage = true;
          setTimeout(() => {
            this.showErrorMessage = false;
            this.nextQuestion();
          }, 1000);
        }
      }
    },
    nextQuestion() {
      if (this.currentQuestionIndex < this.questions.length - 1) {
        this.currentQuestionIndex++;
      }
    }
  }
}

```

```

        this.selectedOption = null;
        clearInterval(this.timer); // Clear the previous timer
        this.resetTimer();
        this.startTimer(); // Start a new timer for the next question
    } else {
        this.showQuiz = false;
    }
},
restartQuiz() {
    this.currentQuestionIndex = 0;
    this.selectedOption = null;
    this.score = 0;
    this.showQuiz = true;
    this.showingFunFact = false;
    this.showErrorMessage = false;
    clearInterval(this.timer); // Clear the previous timer when restarting
    this.resetTimer();
    this.startTimer();
},
resetTimer() {
    this.timeLeft = this.initialTime;
},
},
},

```

## Computed Property

Мојот код содржи и computed property currentQuestion кој овозможува лесно добивање на тековното прашање од листата со прашања.

```

computed: {
    currentQuestion() {
        return this.questions[this.currentQuestionIndex];
    },
},

```

## HTML Дел и Динамичко Прикажување на Содржини:

Во HTML делот на кодот, се користат директиви како v-if и v-for за динамичко прикажување на содржини во зависност од состојбата на апликацијата.

v-if се користи за прикажување или скривање на одредени делови од интерфејсот во зависност од состојбата на started и showQuiz.

v-for се користи за повторување на елементите во листата со опции за тековното прашање.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
  <title>Quiz App</title>
</head>
<body>
  <div id="app" class="container bg-org">
    <div v-if="!started" class="text-center">
      <h1 class="display-4">Welcome to the Quiz App</h1>
      <button @click="startQuiz" class="btn btn-primary btn-lg" style="background-color:
green;">Start Quiz</button>
    </div>
    <div v-else>
      <div v-if="showQuiz" class="row mt-3">
        <h1 class="display-4 mb-4">Quiz App</h1>
        <div class="row pt-3 pb-3">
          <div class="col-4 border border-dark border-1 mb-3">
            <p class="lead"><strong>Question {{ currentQuestionIndex + 1 }} / {{
questions.length }}</strong></p>
          </div>
          <div class="col-4 border border-dark border-1 mb-3">
            <p class="lead text-center"><strong>Score: {{ score }}</strong></p>
          </div>
          <div class="col-4 border border-dark border-1 mb-3">
            <p class="lead text-right"><strong style="color: red;">Time Left: {{ timeLeft
}} seconds</strong></p>
          </div>
        </div>
        <h3 class="mb-3">{{ currentQuestion.question }}</h3>
        
        <ul class="list-group pt-3 bg-org">
          <li
            v-for="(option, index) in currentQuestion.options"
            :key="index"
            @click="selectOption(index)"
            class="list-group-item mb-3"
            :style="{
              backgroundColor: selectedOption === index ? 'orange' : 'transparent',
              border: selectedOption === index ? 'none' : '1px solid black'
            }"
            :class="{ active: selectedOption === index, 'orange-text': selectedOption ===
index, 'dark-orange-bg': selectedOption === index }"
          >
            {{ option }}
          </li>
        </ul>
      </div>
    </div>
  </div>

```

```

    <div v-if="showingFunFact" class="mt-3">
      <p class="text-success" style="background-color: green; font-weight: bold;
color: white !important; padding: 5px;">
        {{ currentQuestion.funFact }}
      </p>
    </div>
    <div v-if="showErrorMessage">
      <p class="text-danger">Wrong choice. Try again!</p>
    </div>
  </div>
  <div v-else class="text-center">
    <h1 class="display-4">Quiz Over</h1>
    <p class="lead">Your Score: {{ score }}</p>
    <button @click="restartQuiz" class="btn btn-primary btn-lg" style="background-
color: green;">Restart Quiz</button>
  </div>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/vue@2.6.14/dist/vue.js"></script>
<script src="main.js"></script>
<style>
  .quiz-image {
    width: 200px;
    height: auto;
  }
  body {
    background-color: rgba(255, 165, 0, 0.5);
  }
</style>
</body>
</html>

```

## Временска Ограниченост

Играта има временска ограниченост на секое прашање. За тоа се користат горенаведените функции од JavaScript. Со ова, корисникот има одредено време за да одговори на прашањето, и по истекот на времето, автоматски се прикажува следното прашање.

## Податоци за Прашањата и Одговорите:

Прашањата и одговорите се чуваат во листа во облик на објекти. Секое прашање има текст на прашањето, опции и индекс на точниот одговор. Ова ги олеснува додавањето на нови прашања во апликацијата.



```
question: 'What does "CPU" stand for in the context of computing?',
  options: ['Central Process Unit', 'Central Programming Unit', 'Central Processing Unit'],
  correctIndex: 2,
  image: 'https://img.freepik.com/free-vector/realistic-style-microchip-processor-background_52683-58932.jpg?w=1060&t=st=1694668487~exp=1694669087~hmac=4f9e4546e10a5200a3a5e38fee3af9fc6d59cecdeb092896d54e0e2a801b50b8',
  funFact: 'The Central Processing Unit (CPU) is the primary component of a computer that acts as its "control center."',
},
```

Вградените функционалности како feedback, тајмер и резултати ја прават апликацијата интерактивна и забавна. Bootstrap дизајнот обезбедува модерен и прилагодлив изглед на апликацијата, што ги задоволува потребите на различни корисници.

## Што може да се подобри?

Првичната имплементација на квиз апликацијата е успешна и ги има основните функционалности како што се прикажување на прашања, тајмер, оценка на корисничкиот резултат и приказ на интересни факти. Но, со секој нов додаток и подобрување, квиз Апликацијата со Vue.js и Bootstrap може да стане уште попопуларна и кориснички ориентирана. Како дел од процесот на развој на софтвер, важно е постојано да се работи на подобрувањето и унапредувањето на апликациите. Меѓутоа, искуството на корисникот може да се подобри со следните карактеристики :

### 1. Дизајн и Визуелен Изглед

И покрај тоа што Bootstrap обезбедува добар основен дизајн, може да се работи на персонализацијата и стилизацијата на апликацијата за да се креира уникатен и атрактивен визуелен изглед.

### 2. Подобрување на Интеракцијата

Подобрување на интеракцијата со можност за корисникот да избере одговор одеднаш и потоа да го потврди со клик на копчето ќе ја подобри искуството.

### 3. Повеќе Прашања и Категории

Додавање на повеќе прашања и категории ќе го зголеми обемот на контент и атрактивноста на апликацијата. Корисниците ќе имаат можност да изберат различни области и ниви на интерес.

#### **4. Управување со Корисниците**

Имплементација на систем за управување со корисниците, како регистрација, логирање и водење на резултатите за секој корисник, ќе додаде социјален аспект и конкуренција меѓу играчите.

#### **5. Мултимедија и Интерактивни Содржини**

Додавање на мултимедија како видеа и аудио, како и интерактивни содржини и анимации, ќе ги зголеми визуелната и аудио атрактивност на апликацијата.

#### **6. Локализација и Преводи**

Овозможување на локализација и преводи на апликацијата во различни јазици ќе го прошири профилот на корисници.

#### **7. Разнообразни прашања:**

Иако апликацијата има 10 прашања од ИТ област, може да се размисли за варијација во типовите прашања. Може да се вклучат текстуални одговори, слики, или доработени прашања со повеќе избори.

#### **8. Тежина на прашањата:**

Може да се внесе ниво на тежина на прашањата, со почетни, средни и напредни нивоа. Ова може да го прилагоди искуството за различни корисници со различни нивоа на знаење.

#### **9.Оценка и рангирање на играчите:**

Може да се додаде систем за оценување и рангирање на играчите. Ова може да додаде елемент на конкуренција и мотивација за корисниците да продолжат да играат.

#### **10.Додатни информации за точни одговори:**

Освен дополнителни факти, може да се вклучи и објаснувањата на точните одговори, за да се образуваат корисниците.

#### **11. Сценарио за крај на играта:**

Може да се даде опција на корисниците како да продолжат по завршувањето на квизот. Може да се вклучи и можност за споделување на резултатите на социјалните мрежи или зачувување на резултатите на нивниот профил.

#### **12.Мобилна поддршка:**

Оваа апликацијата исто така има можност да стане респонзивна и се прилагодува на различни екрани и уреди, вклучувајќи ги мобилните телефони и таблетите.

### **13. Заштита и безбедност:**

Заштитата на оваа апликација исто така е работа што може да се подобри.

## **Заклучок**

Оваа Квиз Апликација со користење на Vue.js и Bootstrap ќе овозможи забава и образование на корисниците, истовремено давајќи им можност за тестирање на нивните знаења во областа на информатичките технологии. Квиз Апликацијата со Vue.js и Bootstrap има значителен потенцијал за подобрување и унапредување. Со употреба на некои од горенаведените препораки, може да се развие и подобри апликацијата, додека се задржува интересот и забавата за корисниците.

Изработи: Станковска Петрина, 201073