

ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

Revas - Webová aplikace pro jednoduché vyrábění testů



Autor: Patrik Tomašík
Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na programování
Třída: IT4
Školní rok: 2024/25

Obsah

Úvod	9
1 Teoretická část	11
1.1 Historie a vznik testování	11
1.1.1 Počátky moderního testování	11
1.1.2 Rozvoj testování ve 20. století	11
1.2 Druhy testů	11
1.3 Význam testování ve vzdělávání	12
1.4 Současné trendy v testování	12
2 Použité technologie	15
2.1 FileSaver.js (Pro ukládání testů)	15
2.2 XML	15
2.3 kalendar.js	16
2.4 JSON	16
3 Architektura aplikace	17
3.1 Celková architektura	17
3.2 Datový model	18
4 Implementace frontendu a backendu	21
4.1 Frontend implementace	21
4.1.1 Uživatelské rozhraní	21
4.2 Backend implementace	22
4.3 Ukládání otázek do souborů	23
5 Testy	27
5.1 Rozhodovací test	27
5.2 Ano, ne test	27
5.3 Psací test	28
5.4 Programovací test	29

5.5	Spojovací test	30
5.6	tabulkový test	30
5.7	Správné odpovídací test	31
6	Herní testy	33
6.1	Pexeso	33
6.2	Oběšenec	33
6.3	Hra s dveřmi	34
6.4	Snake	35
6.5	Kámen, nůžky, papír	36
6.6	Maraton	37
7	Herní testy k procvičování	39
7.1	Lov pokladů	39
7.2	Labyrint	40
7.3	Flappy bird	40
7.4	Klikačka	41
7.5	Miny	42
7.6	Tetris	42
8	Výsledky	45
8.1	Splněné cíle	45
8.2	Nedostatky a budoucí zlepšení	45
8.3	Přínosy	45
9	Závěr	47

Poděkování

Děkuji panu učiteli Mgr. Markovi Lucnemu a Ing. Petru Grussmanovi za rady při vytváření tohoto projektu.

Prohlášení

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 6. 1. 2025

.....
Podpis autora

Abstrakt

Abstrakt Revas je moderní webová aplikace navržená pro efektivní vytváření, správu a sdílení testů. Aplikace umožňuje uživatelům snadno vytvářet testy v různých formátech, které mohou zahrnovat různé typy otázek, jako jsou rozhodovací testy, spojovačky, psací testy, testy ano, ne a další běžné testy. Revas, ale zahrnuje i testy, které se dají dát do podoby her neboli takzvané herní testy. Mezi ně patří například pexeso, oběšenec, snake, hra s dvěma a mnoho dalších herních testů, které dělá z dělání testů zábavu. Dále obsahuje herní testy pro procvičování což tyto testy se od herních moc neliší jen v tom, že se neznámkuje, ale můžete v nich jen dostat body a tyto testy se berou hlavně, proto aby si žák zabavil a při tom něco i přiučil. Revas nabízí uživatelům flexibilitu ve tvorbě testů, čímž poskytuje ideální nástroj pro různé scénáře – od vzdělávacích institucí po profesionální certifikační procesy.

Klíčová slova

vzdělávání, testy, vyrábění testů, herní testy, zábava, herní testy pro procvičování, procvičování

Abstract

Revas is a modern web application designed for efficient test creation, management and sharing. The application allows users to easily create tests in various formats, which can include different types of questions, such as decision-making tests, dot-to-dot tests, writing tests, yes, no tests and other common tests. Revas also includes tests that can be put into the form of games, or so-called game tests. These include, for example, memory games, hangman, snake, the door game and many other game tests that make taking tests fun. It also includes game tests for practice, which are not much different from game tests, only that they are not marked, but you can only get points in them and these tests are mainly taken so that the student has fun and learns something at the same time. Revas offers users flexibility in test creation, thus providing an ideal tool for various scenarios - from educational institutions to professional certification processes.

Keywords

education, tests, test creation, game tests, entertainment, game tests for practice, practice

ÚVOD

Cíle projektu

Hlavním cílem této práce je vytvořit jednoduchou aplikaci pro vyrábění jednoduchých testů, které přináší i prvky zábavy a motivaci se něco i přiučit.

Představení projektu

Revas je aplikace, která umožňuje vytváření testů a kvízů, ale i zábavných herních testů, které kombinují otázky a odpovědi s herními prvky. Tento nástroj je navržen tak, aby byl uživatelsky přívětivý a snadno použitelný, a to jak pro učitele, studenty, tak pro širokou veřejnost, která chce spojit zábavu a učení.

Motivace

Revas vznikl z potřeby zjednodušit proces tvorby testů a zároveň přidat do učení prvek zábavy. Motivací byla frustrace z tradičních a nudných způsobů hodnocení a učení. Chtěl jsem vytvořit nástroj, který by učitelům a lektorům umožnil snadno vytvářet personalizované testy, které zároveň zaujmou studenty. Spojení herních prvků, jako je Pexeso, s otázkami a odpověďmi, oběšenec nebo dokonce studenty milovaná hra snake, přináší nový, interaktivní způsob učení, který je motivující a zároveň efektivní. Takový přístup pomáhá zvýšit zájem o učivo, což je klíčové pro lepší zapamatování a pozitivní vztah k učení.

Struktura aplikace

Aplikace se skládá z přihlašovací stránky, která vám nabídne se přihlásit či zaregistrovat se do aplikace. Jak tak učiníte, tak se dostanete na hlavní stránku, na kterém se nachází kalendář se slidem, kde na každém slidu se můžete seznámit s různými druhy testů, ve kterých můžete vyrábět různé testy. Dále se můžete dostat na stránky, kde jsou nabídky pro výrobů testů, či uložené od jiných uživatelů. Nachází se zde i chat a profilová část se statistikami.

1 TEORETICKÁ ČÁST

Testování je důležitým nástrojem nejen v oblasti vzdělávání, ale i v psychologii, medicíně a dalších oblastech, kde je třeba kvantifikovat a analyzovat různé schopnosti a stavy. Testy mohou mít formu otázek s výběrem odpovědí, esejí, úkolů nebo praktických zkoušek, přičemž volba formy závisí na cíli testování a povaze zkoumané oblasti.

1.1 Historie a vznik testování

Testování, jak ho známe dnes, má dlouhou historii, která sahá až do starověkých civilizací. Již ve starověkém Egyptě, Číně a Řecku se používaly formy testů pro výběr jednotlivců na různé pozice, například pro vojenské nebo vládní služby. V průběhu staletí se metody hodnocení vyvíjely, ale teprve na přelomu 19. a 20. století došlo k modernizaci testování díky rozvoji psychologie.

1.1.1 Počátky moderního testování

Za počátky moderního testování je považován vývoj psychometrie, tedy vědy o měření duševních schopností a charakteristik. Na konci 19. století vytvořil francouzský psycholog Alfred Binet první testy inteligence. Binetův test, známý jako Binet-Simon test, byl zaměřen na identifikaci dětí, které měly potíže se školním učením, a byl základem pro vznik dnešních IQ testů.

1.1.2 Rozvoj testování ve 20. století

V průběhu 20. století se testování stalo klíčovým nástrojem nejen ve vzdělávání, ale také v pracovních, vojenských a psychologických výzkumech. V roce 1916 byl ve Spojených státech vyvinut Stanford-Binet test, který rozšířil původní Binetův test a poskytl lepší nástroje pro měření inteligence. Tento test byl použit pro širokou škálu účelů, od školních zkoušek po výběr do armády.

Ve 20. letech 20. století začaly vznikat také první standardizované testy pro hodnocení výkonu studentů v širším měřítku. Testy byly zaměřeny na akademické dovednosti a vědomosti, například matematiku, čtení nebo gramatiku. V roce 1926 byl vyvinut test Scholastic Aptitude Test (SAT), který se stal jedním z nejznámějších a nejpoužívanějších nástrojů pro hodnocení výkonu studentů v USA.

1.2 Druhy testů

V současnosti existuje mnoho různých typů testů, které se liší jak formou, tak účelem. Některé z nejběžnějších typů testů zahrnují:

- Testy s výběrem odpovědí: Nejrozšířenější formou testování. Studenti vybírají jednu nebo více odpovědí z předem daných možností.
- Esejové testy: Testy, které vyžadují, aby studenti napsali odpověď v textové formě. Tyto testy hodnotí schopnost kritického myšlení a organizování informací.
- Praktické testy: Používají se k hodnocení praktických dovedností, například při výuce technických nebo uměleckých oborů.
- Standardizované testy: Tyto testy mají pevně stanovený formát a jsou používány k měření výkonu studentů ve srovnání s širokou populací. Příkladem jsou SAT, ACT nebo GRE.
- Formativní testy: Cílem těchto testů není pouze hodnocení, ale i podpora učení. Jsou používány v průběhu vzdělávacího procesu, aby učitelé zjistili, jaký pokrok studenti dělají a kde je potřeba zlepšení.

1.3 Význam testování ve vzdělávání

Testování má v současném vzdělávacím systému mnoho funkcí. Mezi hlavní patří:

- Hodnocení pokroku studentů: Testy umožňují měřit, jak dobře studenti zvládli učivo a jaký pokrok udělali během studia.
- Identifikace potřeb studentů: Testy mohou pomoci identifikovat studenty, kteří potřebují další podporu nebo kteří se nacházejí nad průměrem.
- Motivace k učení: Testování může sloužit jako motivace pro studenty, kteří se chtějí zlepšit nebo dokázat své schopnosti.
- Standardizace hodnocení: Standardizované testy poskytují objektivní způsob hodnocení, který umožňuje porovnávat výkon studentů na širší úrovni.
- Zpětná vazba pro učitele: Testy poskytují učitelům důležité informace o tom, jak efektivně vyučují a které oblasti je třeba zlepšit.

1.4 Současné trendy v testování

V současnosti se testování stále vyvíjí a přizpůsobuje novým technologiím a pedagogickým metodám. Mezi současné trendy patří:

- Online testování: S nástupem digitálních technologií se testování přesunulo do online prostředí, což umožňuje flexibilitu a dostupnost testů pro širokou veřejnost.

- Gamifikace testů: V některých případech je testování spojeno s herními prvky, což má za cíl zvýšit motivaci a angažovanost studentů.
- Testování na bázi analýzy dat: Využití pokročilé analytiky a algoritmů k hodnocení výkonu studentů a přizpůsobení testů na míru.
- Automatizované hodnocení: Pokroky v umělé inteligenci umožnily automatizaci hodnocení testů, což zrychluje proces a zajišťuje větší objektivitu.

Z tabulky je patrné, že každá aplikace má své silné a slabé stránky. Zatímco Anki vyniká v pokročilých funkcích a statistikách, postrádá moderní uživatelské rozhraní a gamifikační prvky. Quizlet nabízí dobrou rovnováhu mezi funkcionalitou a použitelností, ale mnoho pokročilých funkcí je dostupných pouze v placené verzi. Memrise má propracovaný systém gamifikace, ale je omezen především na jazykové učení a vyžaduje stálé připojení k internetu.

2 POUŽITÉ TECHNOLOGIE

2.1 FileSaver.js (Pro ukládání testů)

FileSaver.js je jednoduchá knihovna, která umožňuje ukládání datových souborů přímo na uživatelském zařízení. Tato funkce je užitečná zejména v situacích, kdy není k dispozici serverové řešení. Knihovna podporuje různé formáty, což usnadňuje export a správu dat.

Výhody:

- Umožňuje ukládání souborů na uživatelském zařízení bez potřeby serveru
- Podpora pro exportování dat do různých formátů (.txt, .json, .csv)
- Snadná implementace do webových aplikací

Nedostatky:

- Omezená podpora pro pokročilé operace s daty
- Problémy s velkými datovými sadami
- Závislost na prohlížeči, což může způsobovat nekompatibilitu

2.2 XML

Pro převod textových formátů do formátu Moodle XML byla použita technologie, která umožňuje automatizované generování testů, otázek a odpovědí pro platformu Moodle. XML je standardní formát používaný pro výměnu dat, ale může být poněkud těžkopádný pro ruční úpravy.

Hlavní výhody:

- Automatizované generování testových otázek a odpovědí
- Podpora různých typů otázek (výběr z více možností, textové odpovědi, přiřazování)
- Efektivní správa testového obsahu a jeho export na Moodle platformu

Nedostatky:

- Verbose formát (obsahuje velké množství redundantních dat)
- Obtížné na čtení a ruční úpravy
- Méně flexibilní ve srovnání s modernějšími formáty, jako je JSON

2.3 kalendar.js

Knihovna **kalendar.js** slouží k práci s kalendářními daty a správě událostí, což je užitečné například při nastavování termínů testů a jiných důležitých aktivit. Umožňuje vizualizaci dat v kalendářním formátu a nabízí základní funkce pro plánování.

Hlavní výhody:

- Jednoduchá implementace pro základní použití
- Možnost spravovat a zobrazovat události přímo v kalendáři
- Kompatibilita s jinými knihovnami pro rozšíření funkcionality

Nedostatky:

- Omezené možnosti přizpůsobení vzhledu kalendáře
- Komplexní scénáře plánování mohou vyžadovat další úpravy nebo integrace
- Závislost na externích knihovnách při pokročilém použití

2.4 JSON

JSON (JavaScript Object Notation) je lehký datový formát, který byl použit pro ukládání a přenos testových dat. Díky své jednoduché struktuře je snadno čitelný a podporovaný napříč mnoha programovacími jazyky.

Hlavní výhody:

- Jednoduchý a čitelný formát, který je snadno pochopitelný
- Velká podpora napříč programovacími jazyky a nástroji
- Kompaktnější než XML, což zlepšuje efektivitu přenosu dat

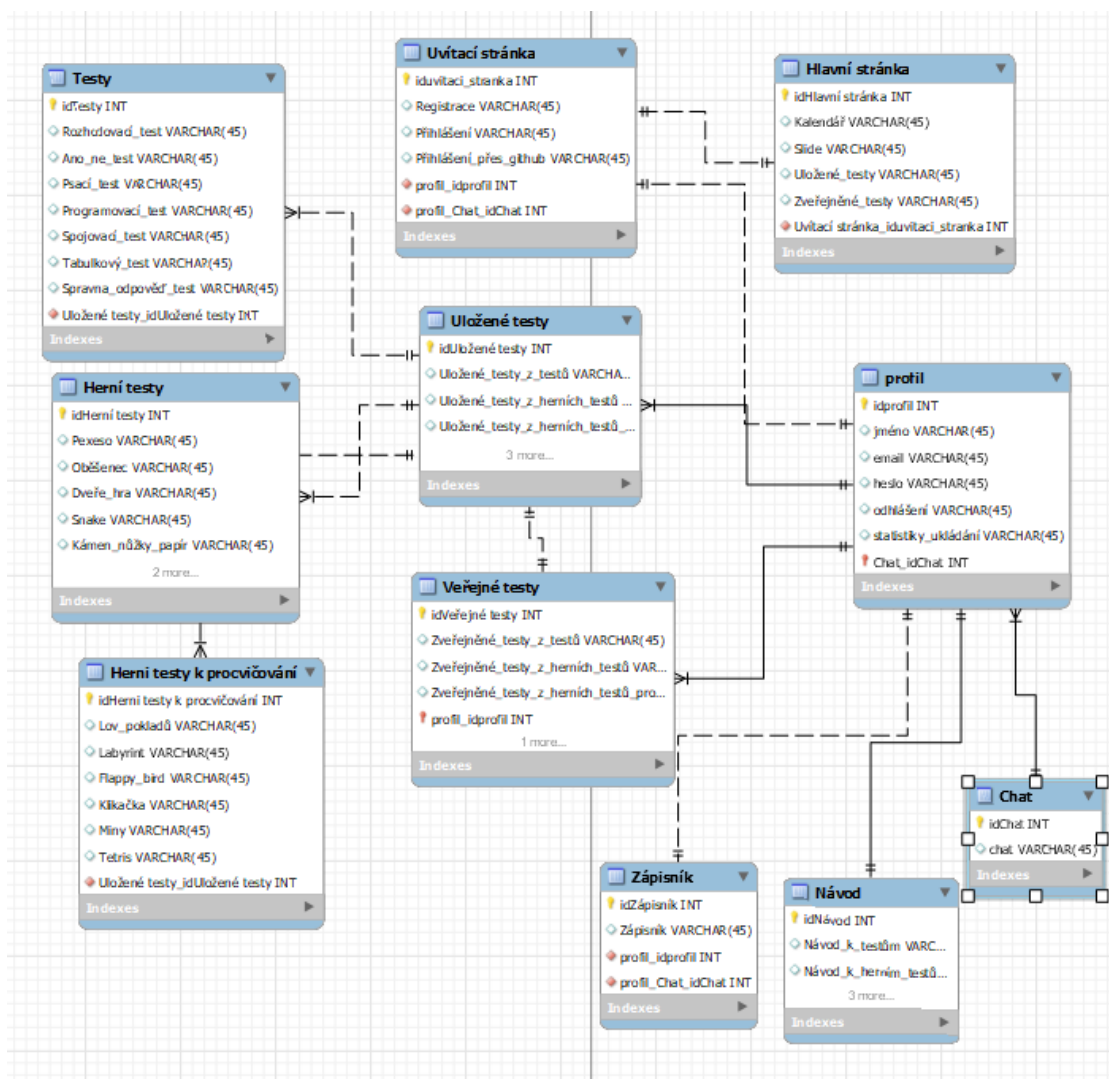
Nedostatky:

- Nevhodný pro složitější datové struktury s atributy
- Nepodporuje standardizované schéma jako XML
- Při velkých datových objemech může být méně efektivní

3 ARCHITEKTURA APLIKACE

3.1 Celková architektura

Webová aplikace se skládá několika stránek, které na sebe navazují. Mezi ně patří hlavně stránky s testy, herními testy a herní testy pro procvičování.



Obrázek 3.1: Architektura aplikace

Jednotlivé části aplikace

- Uvítací stránka
- registrační stránka
- Přihlašovací stránka
- Přihlašovací stránka přes github

- Hlavní stránka
- Stránka s testami
- Stránka s herními testami
- Stránka s herními testami pro procvičování
- Stránka s veřejnými testy od uživatelů
- Stránka s uloženými testy
- Zápisník
- Návod
- Chat
- Profil

3.2 Datový model

Pro reprezentaci dat v aplikaci pro ukládání jsem použil Django.

```

1 # models.py
2 from django.contrib.auth.models import User
3 from django.db import models
4 from django.contrib.auth.models import User
5 from django.conf import settings
6 from django.contrib.auth.models import User
7 from django.contrib.auth.models import AbstractUser
8
9
10 class Message(models.Model):
11     sender = models.ForeignKey(User, on_delete=models.CASCADE,
12                               related_name='sent_messages')
13     recipient = models.ForeignKey(User, on_delete=models.CASCADE,
14                                  related_name='received_messages', null=True, blank=True)
15     text = models.TextField()
16     timestamp = models.DateTimeField(auto_now_add=True)
17
18     def __str__(self):
19         return f"{self.sender} -> {self.recipient or 'Public'}:
20             {self.text[:30]}"

```

```
21
22
23 class TestType(models.Model):
24     name = models.CharField(max_length=100, unique=True)
25     description = models.TextField()
26
27     def __str__(self):
28         return self.name
29
30
31 class UserStatistics(models.Model):
32     user = models.OneToOneField(User, on_delete=models.CASCADE)
33     total_tests_created = models.PositiveIntegerField(default=0)
34     test_statistics = models.JSONField(default=dict)
35
36     def __str__(self):
37         return f"Statistiky pro {self.user.username}"
38
39     def update_statistics(self, test_type_name):
40         # Zvětšíme počet pro daný typ testu
41         if test_type_name in self.test_statistics:
42             self.test_statistics[test_type_name] += 1
43         else:
44             self.test_statistics[test_type_name] = 1
45         self.total_tests_created += 1
46         self.save()
47
48 class Profile(models.Model):
49     user = models.OneToOneField(User, on_delete=models.CASCADE)
50     image = models.ImageField(upload_to='avatars/', null=True, blank=True)
51     displayname = models.CharField(max_length=20, null=True, blank=True)
52     info = models.TextField(null=True, blank=True)
53
54 class Test(models.Model): # Zkontroluj, že třída má tento název
55     title = models.CharField(max_length=100)
56     description = models.TextField()
57
58 from django.db import models
59
```

```
60 class TestModel(models.Model):
61     title = models.CharField(max_length=100)
62     description = models.TextField()
63     questions = models.TextField()
64
65 class Question(models.Model):
66     title = models.CharField(max_length=200)
67     content = models.TextField()
68
69 class Comment(models.Model):
70     question = models.ForeignKey(Question, related_name='comments',
71     on_delete=models.CASCADE)
72     user = models.ForeignKey(User, on_delete=models.CASCADE)
73     content = models.TextField()
74
75
76
```

Kód 3.1: Definice datových modelů

4 IMPLEMENTACE FRONTENDU A BACKENDU

4.1 Frontend implementace

4.1.1 Uživatelské rozhraní

Vzhled uživatelského rozhraní je velmi důležité, protože dává první dojem ze stránky. Jako fronted jsem používat html a Django.

```
1 <div class="container">
2     <h1>Vytvoření Testu - Spojení otázek a odpovědí</h1>
3
4     <input type="text" id="testName" placeholder="Název testu" />
5     <textarea id="testDescription" placeholder="Popis testu" rows="4"></textarea>
6
7     <!-- Nastavení bodování -->
8     <h2>Bodování</h2>
9     <div class="grade-settings">
10         <label for="grade1">Známka 1 (minimální body):</label>
11         <input type="number" id="grade1" name="grades[1]" min="1" value="90">
12         <label for="grade2">Známka 2 (minimální body):</label>
13         <input type="number" id="grade2" name="grades[2]" min="1" value="70">
14         <label for="grade3">Známka 3 (minimální body):</label>
15         <input type="number" id="grade3" name="grades[3]" min="1" value="50">
16         <label for="grade4">Známka 4 (minimální body):</label>
17         <input type="number" id="grade4" name="grades[4]" min="1" value="30">
18         <label for="grade5">Známka 5 (minimální body):</label>
19         <input type="number" id="grade5" name="grades[5]" min="1" value="0">
20     </div>
21
22     <input type="text" id="questionInput" placeholder="Zadejte otázku" />
23     <input type="text" id="answerInput" placeholder="Zadejte odpověď" />
24     <input type="number" id="pointsInput" placeholder="Bodová hodnota" />
25     <button class="add-btn" onclick="addPair()">Přidat pár</button>
26
27     <div id="pairsList">
28         <!-- Seznam párů otázek a odpovědí -->
29     </div>
30
31     <button class="add-btn" onclick="startTest()">Vyzkoušet test</button>
```

```

32 <button onclick="saveToHTML()">Uložit do HTML</button>
33 <button onclick="saveToJSON()">Uložit do JSON</button>
34 <button onclick="saveToXML()">Uložit do XML</button>
35
36
37 <div id="testSection" style="display: none;">
38   <h2>Test: Spojte správné páry</h2>
39   <div id="testContent"></div>
40
41   <button class="add-btn" onclick="checkTest()">Odeslat odpovědi</button>
42   <button class="add-btn" onclick="restartTest()">Restartovat test</button>
43 </div>
44
45
46 <div id="resultSection" class="result-container" style="display: none;">
47   <h3>Výsledky testu:</h3>
48   <div id="resultTables"></div>
49   <p id="testResult"></p>
50   <p id="testGrade"></p>
51 </div>
52
53 </div>
54

```

4.2 Backend implementace

Backend aplikace jsem postavil na Django a při výrobě testů jsem použil Javascript.

- Databáze SQL (Ukládání testů a uživatelů)
- Django
- Javascript
- XML
- JSON


```

38     </head>
39     <body>
40         <h1>${testName}</h1>
41         <p>${testDescription}</p>
42         <form id="testForm">
43     `;
44
45 // Generování otázek
46 questions.forEach((question, index) => {
47     htmlContent += `
48         <div class="question">
49             <p><strong>Otázka ${index + 1}</strong>
50             ${question.text}</p>
51             <label for="answer_${index}">Odpověď:</label>
52             <input type="text" name="answer_${index}"
53             placeholder="Zadejte odpověď">
54         </div>
55     `;
56 });
57
58 htmlContent += `
59     <button type="button" onclick="evaluateTest()">
60     Vyhodnotit test</button>
61     <div id="resultsContainer"></div>
62 </form>
63 </body>
64 </html>
65 `;
66
67 const blob = new Blob([htmlContent], { type: 'text/html' });
68 saveAs(blob, `${testName}.html`);
69 }
70
71
72 function saveTest(type) {
73     const testName = document.getElementById('testName').value;
74     const testDescription =
75     document.getElementById('testDescription').value;
76     const testImage = document.getElementById('testImage').files[0];

```



```

77     const isPublic = (type === 'verejne_testy') ? true : false;
78
79     const formData = new FormData();
80     formData.append('name', testName);
81     formData.append('description', testDescription);
82     formData.append('image', testImage);
83     formData.append('is_public', isPublic);
84
85     // Odeslání dat na server pomocí AJAX
86     fetch('/api/save_test/', {
87         method: 'POST',
88         body: formData
89     })
90     .then(response => response.json())
91     .then(data => {
92         if (data.success) {
93             showModal('spěch', 'Test byl úspěšně uložen.');
```

```

94         } else {
95             showModal('Chyba', 'Nastala chyba při ukládání testu.');
```

```

96         }
97     })
98     .catch(error => {
99         showModal('Chyba', 'Došlo k chybě při komunikaci se serverem.');
```

```

100     });
101 }
102
103 function saveTestToXml() {
104     const testName = document.getElementById('name').value;
105     const testDescription =
106     document.getElementById('description').value;
107     const questions = gatherQuestions();
108
109     let xmlContent = `<?xml version="1.0" encoding="UTF-8"?>
110 <quiz>
111     <name>${testName}</name>
112     <description>${testDescription}</description>`;
113
114     // Generování otázek
115     questions.forEach((question, index) => {

```

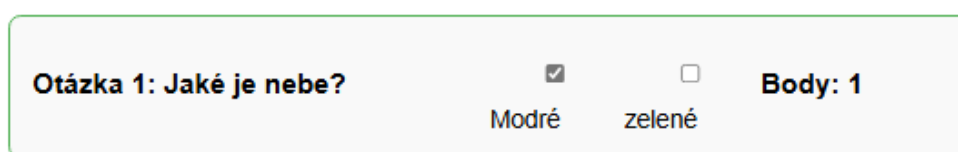
```
116     xmlContent += `
117 <question type="multichoice">
118     <name>
119         <text>Otázka ${index + 1}</text>
120     </name>
121     <questiontext format="html">
122         <text>${question.text}</text>
123     </questiontext>
124     <answer>
125         <text>${question.correctAnswer}</text>
126         <feedback>
127             <text>Správná odpověď!</text>
128         </feedback>
129     </answer>
130 </question>`;
131 });
132
133     xmlContent += `
134 </quiz>`;
135
136     const blob = new Blob([xmlContent], { type: 'application/xml' });
137     saveAs(blob, `${testName}.xml`);
138 }
139
```

5 TESTY

Zde se nacházejí klasické testy, s kterými se můžeme setkat normálně ve škole, maturitě či kvízu. Uživatel si může nastavit jméno testu, popisek, časovač, známky a hlavně otázky a odpovědi. Poté je může vyzkoušet v náhledu, kde se ukážou aji výsledky testu a jde se nakonec stáhnout do počítače ve formátu HTML, JSON a XML pro Moodle.

5.1 Rozhodovací test

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání známkování
- Přidání časovače
- Přidání otázek a body.
- Přidávání odpovědí v podobě odpověď a zaškrtování
- Náhled testu
- Výsledky testů v podobě celkový počet bodů, procentuálně, známka a počet špatných skoro správných a správných odpovědí
- Ukládání testu do formátu HTML, JSON a XML fromát Moodle



The screenshot shows a light gray rectangular box with a thin green border. Inside, on the left, is the text "Otázka 1: Jaké je nebe?". To the right of this text are two checkboxes. The first checkbox is checked and has the word "Modré" below it. The second checkbox is unchecked and has the word "zelené" below it. To the right of these options is the text "Body: 1".

Obrázek 5.1: Rozhodovací test

5.2 Ano, ne test

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání známkování
- Přidání časovače

- Přidání otázek a body
- Přidávání odpovědí v ano, ne
- Náhled testu
- Výsledky testů v podobě celkový počet bodů, procentuálně, známka a počet špatných a správných odpovědí a vypsání správných odpovědí a označení správných
- Ukládání testu do formátu HTML, JSON a XML formát Moodle

Je nebe modré?

Ano

Ne



Obrázek 5.2: Ano, ne test

5.3 Psací test

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání známkování
- Přidání časovače
- Přidání otázek a body
- Přidávání odpovědí v podobě psací odpovědi
- Náhled testu
- Výsledky testů v podobě celkový počet bodů, procentuálně, známka a počet špatných a správných odpovědí a vypsání správných odpovědí a označení správných
- Ukládání testu do formátu HTML, JSON a XML formát Moodle

Otázka 1:

Napište hlavní město České republiky.

Počet bodů:

1

Odpověď:

Praha

Obrázek 5.3: Psací test

5.4 Programovací test

- Přidání názvu testu.
- Přidání popisku a obrázek testu
- Přidání známkování
- Přidání časovače
- Přidání otázek pro delší text a body
- Přidávání odpovědí v podobě delšího textu či přidání souborů
- Náhled testu
- Výsledky ukazuje jen to co uživatel vložil
- Ukládání testu do formátu HTML, JSON a XML formát Moodle

Otázka 1:

Zadejte otázku

Přidat soubor k otázce:

Zvolit soubor

Nebyl vybrán žádný soubor

Body:

1

Zadejte textovou odpověď

Smazat textovou odpověď

Přidat soubor k odpovědi:

Zvolit soubor

Nebyl vybrán žádný soubor

Smazat souborovou odpověď

Obrázek 5.4: Programovací test

5.5 Spojovací test

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání známkování
- Přidání časovače
- Přidání páru, které se mají spojit a body
- Náhled testu
- Výsledky testů v podobě celkový počet bodů, procentuálně, známka a počet špatných a správných odpovědí a vypsání správných odpovědí a označení správných
- Ukládání testu do formátu HTML, JSON a XML formát Moodle

Test: Spojte správné páry

Hlavní město Německa?

Hlavní město ČR?

Hlavní město Slovenska?

Berlín

Praha

Bratislava

Odeslat odpovědi

Obrázek 5.5: Spojovací test

5.6 tabulkový test

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání známkování
- Přidání časovače
- Přidání otázek v podobě tabulek
- Přidávání odpovědí v podobě odpověď do tabulky, která v ukázce se má přetáhnout
- Náhled testu

- Výsledky testů v podobě celkový počet bodů, procentuálně, známka a počet špatných a správných odpovědí a vypsání správných odpovědí a označení správných
- Ukládání testu do formátu HTML, JSON a XML formát Moodle

Název tabulky:

Barvy

Přidat odpověď

Odpověď	Akce
Modrá	Smazat
Zeulená	Smazat

Název tabulky:

Hlavní města

Přidat odpověď

Odpověď	Akce
Praha	Smazat

Obrázek 5.6: tabulkový test

5.7 Správné odpovídací test

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání známkování
- Přidání časovače
- Přidání otázek a body
- Přidávání odpovědí v podobě psací odpověď a zaškrtnutí správné odpovědi
- Náhled testu
- Výsledky testů v podobě celkový počet bodů, procentuálně, známka a počet špatných a správných odpovědí a vypsání správných odpovědí a označení správných
- Ukládání testu do formátu HTML, JSON a XML formát Moodle

Otázka 1:	
Zodpověz na otázky a zaškrtni stejnou barvu s citrónem: Jaké je nebe? Jaká je barva slunce?	<div><div></div><div></div><div></div></div>
Tato otázka musí mít text.	
Odpovědi:	
modrou	
	<input type="checkbox"/>
Správná odpověď	
žlutou	
	<input checked="" type="checkbox"/>

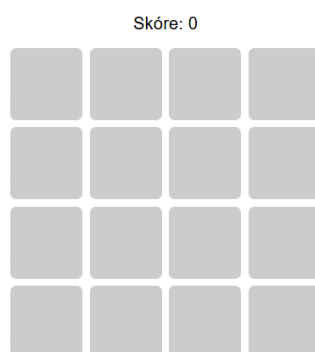
Obrázek 5.7: Správné odpovídací test

6 HERNÍ TESTY

Zde se nacházejí herní testy v podobě klasických a známých her. Uživatel si může nastavit jméno testu, popis, časovač, známky a hlavně otázky a odpovědi. Poté je může vyzkoušet v náhledu, kde se ukážou aji výsledky testu a jde se nakonec stáhnout do počítače ve formátu HTML, JSON a XML pro Moodle. Každý herní test je jedinečný a zábavný. U některých testů se známky vypočítávají automaticky a nemusí se používat bodování.

6.1 Pexeso

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání známkování
- Přidání časovače
- Přidání párů a body
- Náhled testu v podobě pexesa
- Výsledky testů v podobě celkový počet bodů a známka
- Ukládání testu do formátu HTML, JSON a XML formát Moodle



Obrázek 6.1: Pexeso

6.2 Oběšenec

- Přidání názvu testu
- Přidání popisku a obrázek testu.

- Přidání známkování
- Přidání časovače
- Přidání otázky a hádané odpovědi a body
- Náhled testu v oběšence, kde musí hádat písmena slova hádaného a má radu v podobě otázku ohledně toho slova. Pokud uhádne písmenko, tak zezelená, pokud ne zčervená. Má celkově 8 pokusů na uhádnutí.
- Výsledky testů v podobě celkový počet bodů a známka
- Ukládání testu do formátu HTML, JSON a XML fromát Moodle

Otázka: Jakou barvu má nebe?

H

I

Í

J

K

L

M

N

O

Ó

P

Q

Pokusy: 0/8

Obrázek 6.2: Oběšenec

6.3 Hra s dveřmi

- Přidání názvu testu
- Přidání popisku a obrázek testu.

- Přidání známkování
- Přidání časovače
- Přidání otázky a k odpovědi a body
- Náhled testu v podobě, hrajeme za červenou kuličku za pomocí šipek a naším cílem je projít těmi správnými dveřmi. Pokažé máme otázku a každé dveře mají odpovědi, pokud projdeme správnými dveřmi, tak máme body pokud ne, tak nemáme body a v obou případech jdeme na další otázku pokud je. Hráč může mít navýběr i z více dveří pokud výrobce testu udělá více odpovědí, ale maximum dveří je 6.
- Výsledky testů v podobě celkový počet bodů a známka
- Ukládání testu do formátu HTML, JSON a XML fromát Moodle



Obrázek 6.3: Hra s dveřmi

6.4 Snake

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání známkování
- Přidání časovače
- Přidání otázky a čtyř odpovědí a body

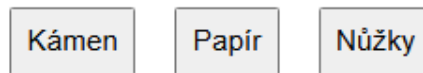
- Náhled testu hry snake a nahoře se objevují otázky a odpovědi, na které musíme odpovídat přitom jak hrajeme hru snake a pokud se dotkneme zdí, tak test se ukončí a pokud odpovíme na otázku, tak po každé otázce se snake zvětšuje a nesmí se dotknout sama sebe.
- Výsledky testů v podobě celkový počet bodů a známka
- Ukládání testu do formátu HTML, JSON a XML formát Moodle



Obrázek 6.4: Snake

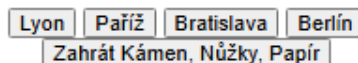
6.5 Kámen, nůžky, papír

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání známkování
- Přidání časovače
- Přidání otázek, odpovědí a nastavení pokusu hraní hry
- Náhled testu je jako normální testy, kde musíme klikat na správné odpovědi, ale při tom budeme mít možnost si zahrát kámen nůžky papír proti programu. Účelem je, že pokud se dostaneme na otázku, kterou opravdu nevíme a nechceme riskovat, tak můžeme si zahrát kámen nůžky papír, kde budeme muset hrát proti programu, kdo získá první tři vítězné, tak vyhrál a remízi se nepočítají. Pokud vyhraje žák, tak se mu otázka přeskočí a dá se mu za dobře a pokud vyhraje program, tak otázku přeskočí a dá se jí za špatně.
- Výsledky testů v podobě celkový počet bodů a známka
- Ukládání testu do formátu HTML, JSON a XML formát Moodle



Výhry: 0 | Ztráty: 0 | Zbývající pokusy: 3

Hlavní město Francie?

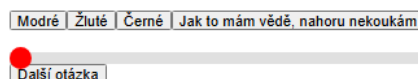


Obrázek 6.5: Kámen, nůžky, papír

6.6 Maraton

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání známkování
- Přidání časovače
- Přidání otázek a čtyř odpovědí, body
- Náhled testu, můžeme vidět našeho běžce a naším cílem je ho dostat nakonec běžecké čáry
- Výsledky testů v podobě celkový počet bodů a známka
- Ukládání testu do formátu HTML, JSON a XML formát Moodle

Jaké je nebe?



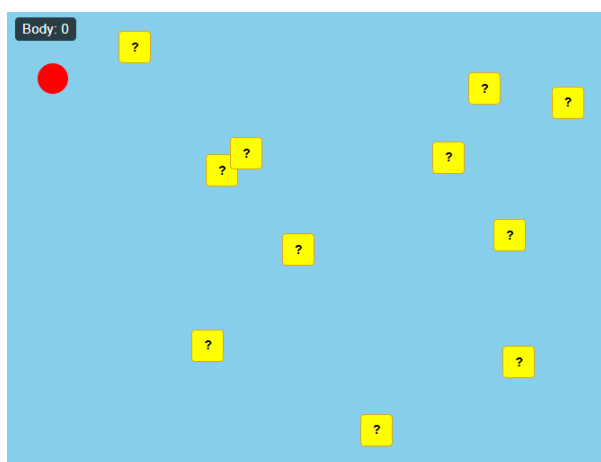
Obrázek 6.6: Maraton

7 HERNÍ TESTY K PROCVIČOVÁNÍ

Zde se nacházejí herní testy v podobě ne moc obvyklých her. Uživatel si může nastavit jméno testu, popis, časovač a hlavně otázky a odpovědi. Poté je může vyzkoušet v náhledu, kde se ukážou aji výsledky testu, ale jen v podobě bodech a jde se nakonec stáhnout do počítače ve formátu HTML, JSON a XML pro Moodle. Tyto testy se neznámkují, ale spíš cílem je hlavně procvičení dané látky. Každý herní test je jedinečný a zábavný.

7.1 Lov pokladů

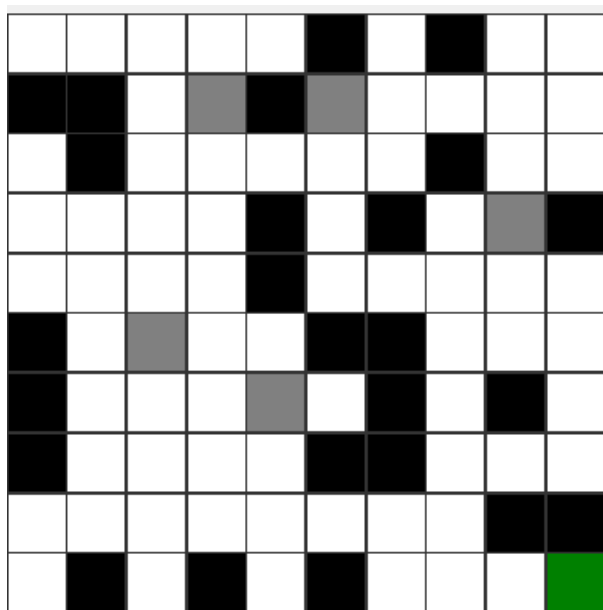
- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání časovače
- Přidání otázek a odpovědí, body
- Náhled testu, hrajeme za červenou kuličku, která hledá poklady. Na mapě můžeme vidět několik pokladů, ale většina je jich falešná. Pokud najdeme pravý poklad, tak nám vyskočí otázka a my musíme zodpovědět správně aby jsme dostali body. Hra končí po najití všech pokladů.
- Výsledky testů v podobě celkový počet bodů
- Ukládání testu do formátu HTML, JSON a XML fromát Moodle



Obrázek 7.1: Lov pokladů

7.2 Labyrint

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání časovače
- Přidání otázek a odpovědí, body
- Náhled testu, Hrajeme za červenou kostku, která je vedle labyrintu, který nám pomáhá jako mapa a mi musíme poslepu získat všechny otázky v podobě kostek a dostat se do cíle co je zelený čtverec a povolí nám to ukončit jen tehdy, kdy budeme mít všechny otázky a budeme v cíli.
- Výsledky testů v podobě celkový počet bodů
- Ukládání testu do formátu HTML, JSON a XML formát Moodle

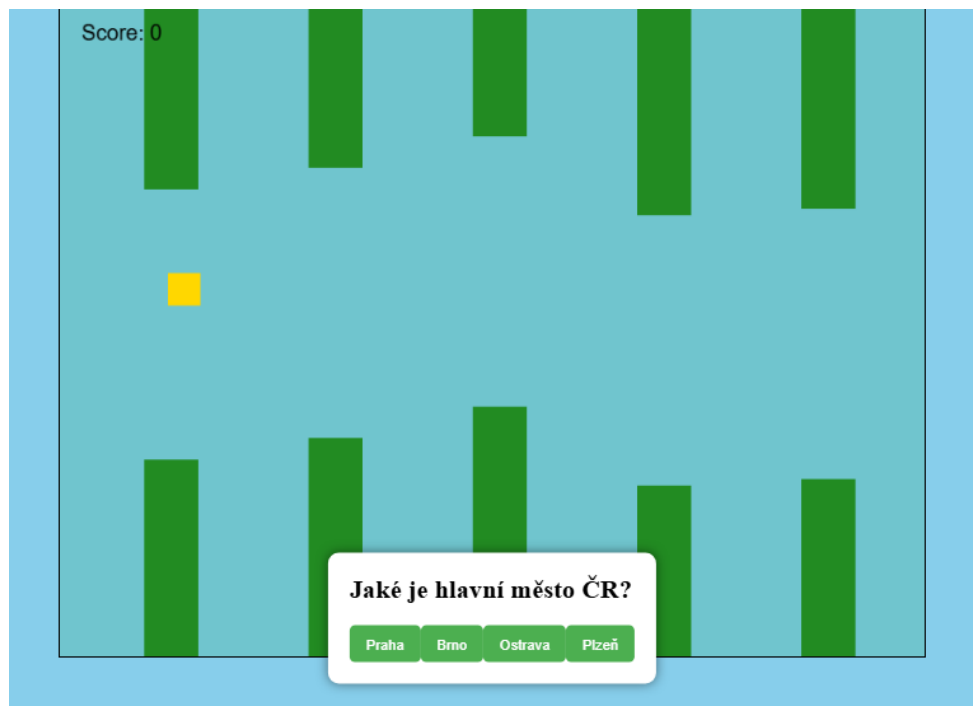


Obrázek 7.2: Labyrint

7.3 Flappy bird

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání časovače
- Přidání otázek a odpovědí, body

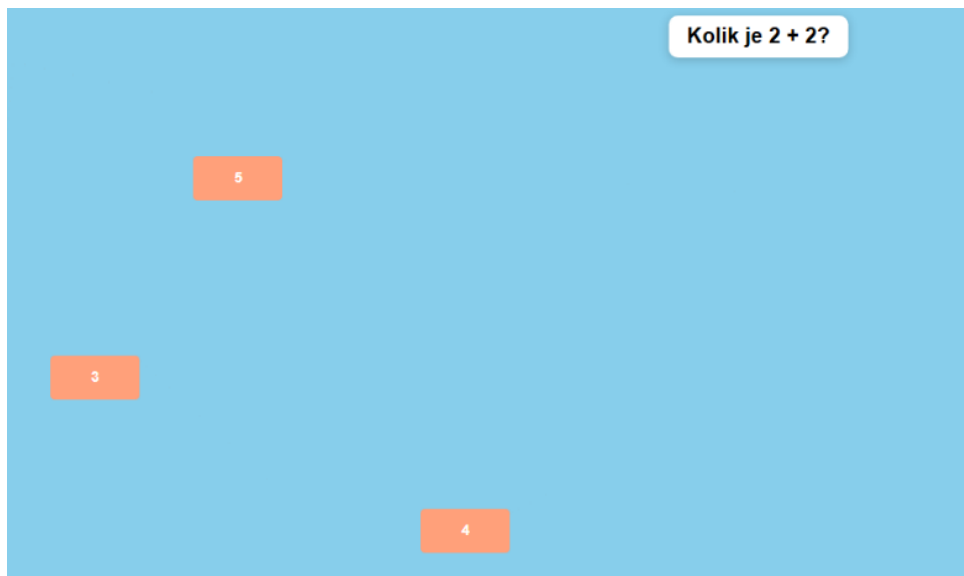
- Náhled testu, klasický Flappy bird, kde musíme přeskakovat mezi trubkami a pokaždé co proskočíme trubkou, tak budeme muset odpovídat na otázky a body dostaneme jen tehdy, kdy je zodpovíme správně. Hra končí po dotknutí trubky či země.
- Výsledky testů v podobě celkový počet bodů
- Ukládání testu do formátu HTML, JSON a XML fromát Moodle



Obrázek 7.3: Flappy bird

7.4 Klikačka

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání časovače
- Přidání otázek a odpovědí, body
- Náhled testu, ukáže se nám otázka a odpovědi se nám náhodně ve velké rychlosti pohybují po mapě a naším cílem je chytit tu správnou odpověď. Hra končí po zodpovězení všech otázek.
- Výsledky testů v podobě celkový počet bodů
- Ukládání testu do formátu HTML, JSON a XML fromát Moodle



Obrázek 7.4: Klikačka

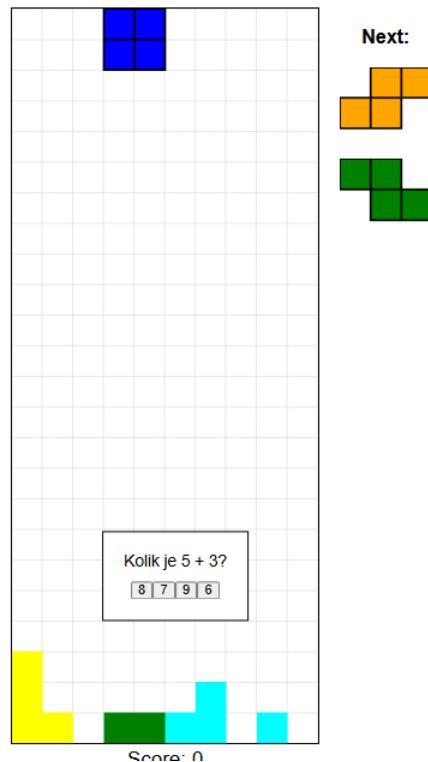
7.5 Miny

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání časovače
- Přidání otázek a odpovědí, body a počet kliků
- Náhled testu, máme minové pole a po každém kliku se nám objeví číslo, jestli je poblíž otázka a pokud je nějaká v okolí otázka, tak ukáže počet v okolí. Pokud je to otázka a zodpovíme jí dobře, tak políčko se zbarví dozelena a máme body a pokud špatně, tak do červena a nemáme body.
- Výsledky testů v podobě celkový počet bodů
- Ukládání testu do formátu HTML, JSON a XML formát Moodle

7.6 Tetris

- Přidání názvu testu
- Přidání popisku a obrázek testu.
- Přidání časovače
- Přidání otázek a odpovědí, body

- Náhled testu je klasický tetris, kde padají z nebe tvary a naším cílem je spojit ve řádcích. Pokud je spojíme, tak se nám ukáže otázka a pokud ji zodpovíme dobře či ne, tak podle toho dostaneme body. Vpravo se ukazují budoucí tvary a hra končí pokud nepůjdou padat další tvary.
- Výsledky testů v podobě celkový počet bodů
- Ukládání testu do formátu HTML, JSON a XML fromát Moodle



Obrázek 7.5: Tetris

8 VÝSLEDKY

8.1 Splněné cíle

- Vyrobení jak základních, tak i testů v herní a procvičovací podobě her
- Základní upravování testů
- Přihlašování do aplikace a registrace nových uživatelů
- Fuunkční chat.
- Fuunkční ukládání testů v různých formátech do počítače a do aplikace
- Statistiky vyrobených testů

8.2 Nedostatky a budoucí zlepšení

- Přidání více testů
- Přidání větší volnosti při vytváření testů
- Platby či nějaký obchod a měna, která by byla možná za odměnu vyrobeného testu
- Přidání dalších druhů testů v podobě kahootu a nebo denních výzev

8.3 Přínosy

- Efektivnější děláání testů.
- Zábavu
- Zkušenosti jak zacházet s časem a plánování
- Zkušenosti do budoucna

9 ZÁVĚR

Aplikace Revas představuje moderní nástroj, který výrazně usnadňuje tvorbu a správu testů v různých formách. Díky flexibilnímu přístupu a intuitivnímu uživatelskému rozhraní nabízí uživatelům širokou škálu možností pro tvorbu jak klasických testů, tak i v podobě her a na procvičení. Celkově jsem si tvorbu této aplikace užil. Přiučil jsem se mnoho nových věcí a osvěžil jsem si práci s Django a Javascriptem což jsou mé dva jazyky, které nemám moc v lásce. Nejvíce jsem si užíval děláni herních testů, kde jsem se snažil udělat klasické hry, které každý zná a rád si zahraje. Nachází se zde určitě mnoho nedostatků a věcí, kterých se dají změnit či udělat úplně jinak, ale snažil jsem se nejvíce o to aby to bylo hlavně funkční. Dalo mi to jak hodně zkušeností, tak hlavně poučení o tom, že toho času opravdu moc není a je opravdu důležité s ním zacházet a plánovat lépe. I přes všechny moje útrapy a nervy, které jsem stratil při děláni tohoto projekt, který jsem několikrát chtěl vzdát a dělat úplně něco jiného jsem nakonec vyrobil webovou aplikaci pro děláni testů, kde jsem si splnil nakonec i své základní cíle., Nakonec mám webovou aplikaci, na kterou můžu být hrdý, protože jsem do ní dal hodně práce a určitě v budoucnu v ní budu pokračovat.

Zdrojový kód projektu je na GitHubu (<https://github.com/Patrik1T/Maturitniprojekt?tab=readme-ov-file#classquiz>).

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

Seznam použitých informačních zdrojů

[1] 10.9. Quiz Flask App by simboli: Open-source webová aplikace postavená na Pythonu a Flasku s MySQL databází, umožňuje vytváření a správu kvízů. Dostupné z: <https://github.com/thepasterover/flask-quiz-app>

[2] 10.9. Quiz Flask App Open-source webová aplikace postavená na Pythonu a Flasku s MySQL databází, umožňuje vytváření a správu kvízů. Dostupné z: <https://github.com/simboli/quiz-flask-app>

[3] 10.9. ClassQuiz: Open-source aplikace podobná Kahoot!, umožňuje učitelům vytvářet kvízy, které mohou studenti hrát na dálku. Dostupné z: <https://github.com/mawoka-myblock/ClassQuiz>

[4] 9.10. Moosh: Nástroj pro správu Moodle z příkazové řádky, který může pomoci při exportu testů do Moodle. Dostupné z: <https://github.com/tmuras/moosh>

[5] 9.10. Moodle XML format: Nástroj pro import a export testových otázek a odpovědí do platformy Moodle ve strukturovaném XML formátu. Dostupné z: https://docs.moodle.org/405/en/Moodle_XML_format

[6] 9.10. JSON: Lehký formát pro výměnu a ukládání dat, ideální pro jednoduché a efektivní manipulace s daty mezi aplikacemi. Dostupné z: <https://zdrojak.cz/clanky/json-jednotny-form>