

# **Gymnázium, Praha 6, Arabská 14**

*Obor Programování*



## **Ročníková práce**

Jakub Švéda, Egor Čekunov, Nella Carraro

## **Online učebnice matematiky**

Duben 2023

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V..... dne.....

.....

Jakub Švéda

.....

Egor Čekunov

.....

Nella Carraro

**Název práce:** Online učebnice matematiky

**Autoři:** Jakub Švéda, Egor Čekunov, Nella Carraro

**Abstrakt:**

Program bude webová aplikace. Aplikace bude obsahovat rozsah učiva dle učebního plánu 6.-9. třídy ZŠ, tzn. procvičování lineárních rovnic, geometrie, poměrů, logických úloh, zlomků, procent, mocnin, rovnic o dvou neznámých, funkcí. Procvičování bude probíhat prostřednictvím různých forem her. Aplikace bude generovat příklady k danému tématu. Úkoly budou mít různé obtížnosti. Každý uživatel bude mít účet, kam bude ukládat své skóre.

# Obsah

<b>Obsah.....</b>	<b>3</b>
<b>1. Úvod.....</b>	<b>5</b>
<b>2. Architektura.....</b>	<b>6</b>
2.1. Princip komunikace.....	6
2.2. Oprávnění.....	6
<b>3. Použité technologie.....</b>	<b>7</b>
3.1. JavaScript.....	7
3.2. Css.....	7
3.3. Bootstrap.....	7
3.4. React.....	7
3.5. Next.js.....	8
<b>4. Hry.....</b>	<b>9</b>
4.1. Signalizace správnosti odpovědi.....	9
4.2. Generace her.....	9
4.2.1. Negeometrické hry.....	9
4.2.2. Geometrické hry.....	9
4.3. Volba správné odpovědi.....	10
4.3.1. Popis.....	10
4.3.2. Obtížnosti.....	10
4.4. Hledání dvojic.....	10
4.4.1. Popis.....	10
4.4.2. Obtížnosti.....	10
4.5. Pravda nebo lež.....	11
4.5.1. Popis.....	11
4.5.2. Obtížnosti.....	11
4.6. Otáčení karet.....	11
4.6.1. Popis.....	11
4.6.2. Obtížnosti.....	11
4.7. Řazení kartiček.....	11
4.7.1. Popis.....	11
4.7.2. Obtížnosti.....	11
4.8. Geometrie.....	12
4.8.1. Popis.....	12
4.8.2. Obtížnosti.....	12
<b>5. Frontend.....</b>	<b>13</b>
5.1. Třídy.....	13
5.2. Uživatelské portfolio.....	13
5.3. Základní stránky.....	13
5.4. Stavba her.....	13
5.5. UI a UX.....	14
<b>6. Backend.....</b>	<b>15</b>
6.1. Autentifikace.....	15

6.2. Uživatel.....	15
6.2.1. Základní informace [/api/user/info/get].....	15
6.2.2. Přidání skóre [/api/user/score/add].....	15
6.2.3. Získání skóre [/api/user/score/get].....	15
6.2.4. Resetování skóre [/api/user/score/reset].....	15
6.3. Hry.....	16
6.3.1. Generace [/api/games/generation].....	16
6.3.2. Volba správné odpovědi [/api/games/getChooseCorrectStage].....	16
6.3.3. Stejně dvojice [/api/games/getEqualPairs].....	16
6.3.4. Řazení kartiček [/api/games/getSorterGame].....	16
<b>7. Databáze.....</b>	<b>17</b>
<b>8. Závěr.....</b>	<b>18</b>
<b>9. Bibliografie.....</b>	<b>19</b>

# 1. Úvod

Cílem projektu je vytvořit webovou aplikaci, která bude sloužit jako cvičebnice matematiky pro základní školy formou různých typů cvičení, potažmo her. Aplikace by měla zahrnovat obtížnost příkladů odpovídající úrovni šestého, sedmého, osmého a devátého ročníku a to včetně geometrie. Díky vedeným statistikám budou moci žáci snadno rozpoznat jejich progres.

Samotný web by měl být responzivní tak, aby bylo možné většinu her hrát na všech často používaných zařízeních, jako jsou mobilní telefony, tablety a počítače s monitory různých velikostí.

Záměr pro vytvoření této práce vychází z osobních zkušeností při doučování žáků základní školy. Často se totiž stává, že žáci neumí základní počty, nebo v nich pravidelně chybují. Řešením pro tuto situaci se nabízí vytvořit právě takovou aplikaci.

Druhým záměrem pro vytvoření této práce je čistě osobní zájem vyzkoušet si technologie jako je React, Next.js a další.

## 2. Architektura

Aplikaci je možné rozdělit podle různých kritérií. Například dle vývojových částí jako jsou backend, frontend, ale také podle jednotlivých obrazovek samotné webové aplikace nebo principu komunikace mezi různými částmi aplikace. V této kapitole je k nalezení právě třetí způsob.

### 2.1. Princip komunikace

Protože je celý tento projekt napsaný v programovacím jazyce Javascript, ke komunikaci mezi backendem a frontendem se používá takzvané rozhraní REST API. Zároveň však díky možnostem *Next.js* dokáží fungovat některá částí stránky jako statická stránka, tudíž se její obsah vygenerujeme pomocí serveru.

Lehce popsatelné je to na příkladu stránky některé ze tříd. O takové stránce víme, že na ni nemá přístup nepřihlášený uživatel a zároveň využívá data z databáze. V normálním případě by to znamenalo, že při dotazu na stránku třídy by server vrátil stránku, která by musela při renderování poslat další dotaz na server a zjistit, zda je daný uživatel přihlášený. Na základě toho by pak bylo rozhodnuto, jestli uživatel může dostat data stránky. Naopak při použití kombinace statické a dynamické stránky je postup jiný. Při prvním dotazu na server, je uživatel rovnou ověřen a v případě, že není, tak je mu vrácena přihlašovací stránka. Navíc díky tomu, že všechny rozhodovací procesy probíhají na serveru, je celkový čas menší, než v prvním případě. Zbytek dat je nicméně posílán standardně skrze REST API.

### 2.2. Oprávnění

Z hlediska oprávnění má aplikace pouze dva stavy. Uživatel je buď přihlášený nebo ne. Pokud není přihlášený, má přístup pouze ke kořenové stránce projektu, tedy k obsahu komponenty Home v rámci souboru `index.jsx`, a přihlašovací stránce. V případě, že se pokouší dostat na jinou stránku projektu, je automaticky přesměrován na přihlašovací stránku.

V opačném případě, kdy je přihlášený, má přístup na všechny stránky kromě kořenové stránky, neboť při pokusu o její získání je přesměrován na domovskou stránku v souboru `homepage.jsx`.

## 3. Použité technologie

V této kapitole jsou popsány technologie, které byly použity v projektu. Je popsáno, o jaké technologie se jedná a jak fungují. Také je popsáno, jak byly v tomto projektu využity.

### 3.1. JavaScript

JavaScript je programovací jazyk používaný při vytváření webových stránek a zároveň nezbytnou součástí většiny moderních webových stránek. Jde o interpretovaný jazyk, což znamená, že kód se vykonává na straně klienta, takže k jeho spuštění není třeba překladač. Interpretaci tedy provádí samotný prohlížeč uživatele.

JavaScript, zkráceně JS, se používá pro interaktivní prvky stránek. To jsou například animace, tlačítka, validace formulářů aj. Pro vývoj komplexnějších webových aplikací existuje mnoho JavaScriptových frameworků například React, Node.js, Angular a Vue. Tyto frameworky usnadňují tvorbu webových aplikací a dávají prostor pro vývoj komplexnějších projektů.

### 3.2. Css

CSS (*Cascading Style Sheets*) je jazyk používaný pro popis a úpravu vzhledu webových stránek a webových aplikací. Stejně jako JS je nutnou součástí moderních webových stránek. Umožňuje oddělit upravování obsahu od vzhledu stránky. Nejčastěji je používán s HTML (*HyperText Markup Language*) nebo XML (*eXtensible Markup Language*), kde HTML a XML udává strukturu i obsah, ale CSS udává vzhled.

Pomocí CSS se dají měnit vlastnosti jako jsou barva, velikost, typ písma, pozice a další vlastnosti pro různé prvky na stránce. Atributy se dají přiřadit určitým tagům HTML a nebo lze vytvořit vlastní novou třídu se specifickými designovými vlastnostmi.

### 3.3. Bootstrap

Bootstrap je open-source framework, který obsahuje předem vytvořené CSS styly. To umožňuje jejich lehkou implementaci a zároveň poskytuje responzivitu webových stránek. Velká část projektu byla provedena právě pomocí Bootstrapu, což zjednodušilo a urychlilo designování uživatelského rozhraní (UI), například navigační menu, tlačítka, rozdělení stránky aj.

Stránka je určena pro děti, proto byly základní barvy Bootstrapu změněny pomocí souboru *customBootstrap.scss* na více barevnější a vibrantní barvy z předchozích neutrálních. Bootstrap ale obsahuje pouze prvky pro běžné webové stránky, a proto bylo zapotřebí použít CSS soubory pro netypické nebo specifické části stránek.

### 3.4. React

Nejpoužívanější open-source JavaScript knihovna pro tvorbu UI je *React.js*, i proto byla použita v tomto projektu. Hlavní složkou Reactu jsou komponenty, což jsou samostatné znovupoužitelné části UI, které se skládají z HTML-like a JavaScriptového kódu. Komponenty pak lze poskládat do složitějších bloků a vytvořit tak rozsáhlé UI designy.



Jednotlivé komponenty jsou uloženy ve složce *components*. Například *NavBar.jsx*, což je komponenta představující navigační lištu, která je navrhnutá pomocí Bootstrapu. Tato komponenta se dá spojit s dalšími a jejich složením vznikne webová stránka.

### 3.5. Next.js

Next.js je open-source framework pro React, který umožňuje webovým aplikacím *server side rendering* a generování statických nebo inkrementálního generování stránek. Inkrementální generování stránek umožňuje generování stránek na vyžádání s cílem generovat stránky dynamicky a efektivně, což pro webovou stránku s hrami je velmi důležité proto, aby se obsah rychle načítal a hry tak byly plynulé bez velkého sekání a načítání.

## 4. Hry

Následující podkapitoly popisují jednotlivé typy her, které učebnice nabízí. Společně s nimi jsou popsány také dílčí obtížnosti her.

### 4.1. Signalizace správnosti odpovědi

Pokud určil uživatel výsledek správně, vybraná odpověď se označí zelenou barvou a hra postoupí na další příklad v pořadí. V opačném případě se tlačítko zbarví červeně, což je symbolem špatné odpovědi. Danou chybu pak lze vždy opravit zpět na správnou.

### 4.2. Generace her

V případech, kdy je třeba jednu z her vygenerovat, jsou vytvořeny soubory, které se nachází ve složce `/lib/generation/`.

#### 4.2.1. Negeometrické hry

První z dvojice souborů je *equationGeneration.js*, který obsahuje generaci negeometrických úloh.

Jedna z takových úloh je tvorba objektu pro hru *Volba správné odpovědi*. Metoda vytvoří objekt s otázkou, tedy rovnicí dle požadovaného tvaru, pole možných odpovědí, včetně té správné, a také správnou odpověď. To vše je vytvořeno dle požadovaných parametrů.

Druhá úloha slouží pro vytvoření dvojic příkladu a správné odpovědi. Konkrétně vrací objekt Map, který obsahuje jako klíče správné odpovědi a jako hodnoty klíčů příklady s různými operacemi, jejichž výsledek je roven klíči. I zde je generace ovlivněna vstupními parametry metody.

Třetí úloha tvoří pole, které slouží hře *Řazení kartiček*. Pole obsahuje čísla, kde pokud toto pole rozdělíme na trojice a vykonáme operace prvních dvou čísel z trojice, dostaneme třetí číslo. Pro rozpoznání, že je třetí číslo výsledkem, je mu na začátek připsán operátor rovná se (=).

Dále soubor obsahuje utilitní funkce, které pomáhají při generaci i v jiných částech aplikace, jimiž jsou *getNumberDecomposition(n)*, jež vrací pole prvočinitelů vstupního čísla, dále *getRandomInt(rangeStart, rangeEnd)*, která vrací náhodné číslo v uzavřeném intervalu  $< rangeStart; rangeEnd >$  dle parametrů. Další metoda *getRandomBoolean()* je odvozena od předchozí a vrací náhodnou hodnotu typu boolean, tedy true nebo false. Poslední z metod je funkce *shuffleArray(arr)*, která vrací zamíchané vstupní pole.

#### 4.2.2. Geometrické hry

Druhý z dvojice souborů je *geometryGeneration.js*, který obsahuje generaci geometrických úloh.

Zásadní je tvorba objektu pro hru *Geometrie*. První metoda vytvoří objekt s otázkou, tedy zadání, tvar geometrického útvaru, hodnota obsahu a specifikaci trojúhelníku. Druhá metoda vytvoří zadání zaměřené na obsah obdélníku. Třetí

metoda je esenciální, protože vytváří sadu výše uvedených úloh a vrací je. To vše je vytvořeno dle požadovaných parametrů.

## 4.3. Volba správné odpovědi

### 4.3.1. Popis

Hra má následující pravidla. Uživatel si přečte vygenerovanou otázku ve tvaru  $x = n_1 \pm n_2 \pm \dots \pm n_k$  a následně vybere jednu z nabízených odpovědí. Měl by vybrat takovou, aby po dosazení čísla za  $x$  zachoval rovnost.

### 4.3.2. Obtížnosti

Tato hra má 3 obtížnosti. První je určena pro uživatele šesté a sedmé třídy, při níž uživatel sčítá a odečítá čísla od 1 do 10.

Druhá obtížnost je určena pro osmou třídu, při níž uživatel sčítá a odečítá čísla od 1 do 50.

Třetí obtížnost je určena pro uživatele deváté třídy. Při třetí obtížnosti uživatel sčítá a odečítá čísla od 1 do 100.

## 4.4. Hledání dvojic

### 4.4.1. Popis

Tato hra je založena na podobném principu jako pexeso. Uživatel však vidí všechny kartičky, což znamená, že si kartičky nemusí pamatovat, ale soustředit se pouze na matematikou část úlohy.

Mezi kartičkami se nachází dvojice, které se sobě rovnají. Úkolem je právě tyto dvojice najít a označit. Označení správnosti je standardní jako u všech her, nicméně zde je označení rozšířeno ještě o již uhodnuté dvojice a označenou kartičku, ke které hledáme další do páru. Uhodnuté jsou označené šedou barvou a nelze je znovu označit. První označená kartička se značí oranžovou barvou.

### 4.4.2. Obtížnosti

Tato hra má 4 obtížností, první je určena pro žáky šesté a sedmé třídy. Při první obtížnosti uživatel má kartičky s násobením dvou čísel, která jsou menší než 10.

Druhá obtížnost je určena pro žáky šesté a sedmé třídy, při níž má uživatel kartičky se sčítáním dvou čísel, která jsou menší než 40.

Třetí obtížnost je určena pro žáky osmých a devátých tříd. Zde jsou vygenerované kartičky s číslem a odmocninou jeho druhé mocniny. Žáci tím mohou procvičit mocniny a odmocniny například k jednotným přijímacím zkouškám.

Čtvrtá obtížnost je určena pro žáky devátých tříd. Jedná se o nejtěžší hru. Zde si žák může procvičit všechna výše uvedená nastavení, ale se zvětšenou obtížností. Bude to výzva pro žáky deváté třídy i pro studenty prvních ročníků středních škol.

## 4.5. Pravda nebo lež

### 4.5.1. Popis

Pravda nebo lež je hra založená na tom, že je uživateli zobrazena rovnice, kde na jedné straně jsou vygenerovaná čísla s různými operacemi a na druhé je číslo, které má v ideálním případě odpovídat výsledku operací na první straně.

Hra je však nastavena tak, aby rovnice vždy neplatila. Úkolem uživatele je pak rozhodnout, zda rovnice platí nebo ne.

### 4.5.2. Obtížnosti

Obtížnosti této hry jsou stejné jako ve hře *Hledání dvojic*.

## 4.6. Otáčení karet

### 4.6.1. Popis

U této hry je znovu vygenerovaný určitý soubor operací s výslednou hodnotou, který se zobrazí na kartičce. Narozdíl od ostatních her tato nemá aktivní hodnocení. Uživatel tedy nevybírání ze správných odpovědí, nýbrž najetím myši na kartičku se mu zobrazí pouze správná odpověď. Proto tato hra ani neukládá skóre.

### 4.6.2. Obtížnosti

Obtížnosti této hry jsou stejné jako ve hře *Hledání dvojic*.

## 4.7. Řazení kartiček

### 4.7.1. Popis

Řazení kartiček je zřejmě nejtěžší hra v naší aplikaci. Uživateli se zobrazí kartičky, které by měl seřadit tak, aby odpovídaly následujícímu systému.

Pokud bychom prováděli od nuly operace vždy mezi mezi-výsledkem a následujícími čísly, pak bychom měli dostat další mezi-výsledek, až se dostaneme k poslednímu. Tzn. po sečtení prvních dvou čísel dostaneme třetí, poté provedeme operace se dvěma dalšími čísly, dostaneme druhý mezi-výsledek.

Tímto způsobem pokračujeme až do chvíle, kdy dosáhneme posledního mezivýsledku. V tu chvíli jsme hru úspěšně splnili a ověříme výsledek.

### 4.7.2. Obtížnosti

Tato hra má 3 obtížností, z nichž první je určena pro žáky sedmé třídy. Při první obtížnosti dostane uživatel kartičky se sčítáním, odčítáním, násobením nebo dělením čísel, která jsou menší než 20.

Druhá obtížnost je určena pro žáky osmých tříd. Zde se vyskytují čísla ve všech případech v rozmezí 1 až 40.

Třetí obtížnost je určena pro žáky devátých tříd. Zde se vyskytují čísla ve všech případech v rozmezí 1 až 60.

## 4.8. Geometrie

### 4.8.1. Popis

Geometrie je hra, která poskytuje uživateli čtyři věci. Otázku, mřížku, a dvě tlačítka *Odstranit* a *Zkontrolovat*. Otázka je generována náhodným způsobem na základě požadovaných parametrů.

Uživatel dostane náhodně vygenerované otázky, do mřížky potom musí zadat geometrický útvar a nakonec zmáčknout tlačítko *Zkontrolovat*. Kdyby během zadání bodu uživatel omylem zadal špatný útvar, může použít tlačítko *Odstranit*, což odstraní poslední přidaný bod.

Když uživatel zadá správně geometrický útvar, zobrazí se mu následující úloha v pořadí.

### 4.8.2. Obtížnosti

Tato hra má dvě obtížnosti. První je určena pro žáky šestých tříd, druhá je určena pro žáky sedmých a osmých tříd.

## 5. Frontend

### 5.1. Třídy

Mezi hlavní komponenty aplikace patří třídy, ve kterých se celý obsah stránek renderuje. Samotný rendering probíhá ve třídě `MyClass.jsx`, která podle JSON předpisu třídy vygeneruje jednotlivé hry.

Tento předpis je pouze ilustračně poskytnut u šesti třídy. Při rozšíření aplikace je tedy nutné implementovat i předpis ostatních tříd. Obtížnosti jsou však již implementované, takže jsou připraveny k použití. Důvod, proč nebyl implementován, je prostý. Pokud bude někdo aplikaci reálně používat, je jasné, že bude chtít obtížnosti a druhy her kombinovat takovým způsobem, aby odpovídal úrovni konkrétních žáků jednotlivých tříd. Takové úpravy je možné provést v komponentě `useClassContent.jsx`, která funguje jako React hook.

Díky funkcionalitě *lazy loading* v kombinaci s nastavením *ssr: false*, kterou Next.js poskytuje, není první dotaz na server tolik náročný a přesouvá tak částečně výkon na stranu klienta. Při dotazu se totiž klientovi dodá pouze obsah stránky, který aktuálně potřebuje.

### 5.2. Uživatelské portfolio

Uživatelské portfolio má zřejmý účel, zobrazuje statistiky uživatele v jednotlivých hrách, které již hrál a získal v nich určité skóre. Mimo to poskytuje základní funkce jako odhlášení nebo resetování statistik.

Zmíněné statistiky mají dva typy hodnocení, jimiž jsou procentuální ohodnocení a absolutní číselné hodnocení. Aplikace zobrazuje procentuální úspěšnost a neúspěšnost uživatele, kterou získal v jednotlivých hrách vypočtenou na základě počtu špatných a správných odpovědí. Tyto počty se poté zobrazují společně s celkovým počtem záznamů a procenty.

Při resetu statistik se pošle dotaz na server, který pak nenávratně vymaže všechny záznamy tohoto uživatele.

### 5.3. Základní stránky

Mezi základní stránky patří první kořenová stránka v souboru `index.jsx` a domovská stránka v souboru `homepage.jsx`.

Úvodní stránka poskytuje základní popis webové aplikace a tlačítko pro přihlášení nebo aktivaci účtu. Uživatel je přesměrován na stránku se seznamem providerů, skrz které se lze registrovat.

Na domovskou stránku je uživatel přesměrován bezprostředně po přihlášení, z níž se může dostat do jedné ze tříd a to buď skrze horní lištu, nebo kliknutím na jedno z tlačítek v seznamu tříd.

### 5.4. Stavba her

Každá z her má samozřejmě odlišné nároky na svoji stavbu, ale některé principy jsou zobecnitelné a zachovatelné. V této podkapitole jsou tedy uvedeny společné znaky všech her.

Hlavním znakem je bezpochyby to, že fungují samy o sobě jako funkční React komponenty s určitými parametry.

Druhým společným znakem je design. Jsou totiž vytvořené tak, aby se chovali stejně v rámci breakpointů, vizuálu obecně, animací při zvolení odpovědi, apod. S tím souvisí i komponenta *GameNav.jsx*, která se stará o přepínání jednotlivých her do dalších fází (částí).

Třetím společným znakem jsou zmiňované fáze. Protože React funguje na základě takzvaných *state* proměnných, byla vytvořena jedna, která funguje jako pole, do kterého postupně přibývají nové fáze hry načtené z databáze, až do chvíle, kdy dosáhne požadované délky.

## 5.5. UI a UX

Pro design aplikace byl zvolen CSS framework Bootstrap a to konkrétně v jeho verzi pro React reprezentovaný knihovnou *react-bootstrap*. Důvodem pro zvolení takového frameworku je fakt, že aplikace takového rozsahu by byla graficky náročná a tím pádem zbytečně zpomalována.

Zásahům nebo spíše doplněním CSS do projektu se přirozeně nešlo vyhnout ať už, co se barev týče nebo jiným změnám. Změny byly však vždy většinou lokálního charakteru, myšleno v jednotlivých komponentách, díky čemuž byly pro jednotlivé komponenty tvořeny takzvané CSS moduly.

V těchto modulech se například vzhledem ke hře *Hledání dvojic* nacházely použité animace a pokročilejší styly, které bootstrap neobsahuje.

Díky použití výše zmíněné technologie *lazy loading* bylo potřeba zajistit, aby v případě požadování ještě nenačteného obsahu, aplikace dala uživateli vědět, že dochází k načítání. K těmto účelům slouží animace načítacího kolečka utilitní komponenty *LoadingSpinner.jsx*.

## 6. Backend

Následující podkapitoly popisují vše týkající se backendové části aplikace. Celý backend je s frontendem úzce spojen, protože je napsán pomocí technologie Next.js. To se týká zejména funkcionality, která se nazývá *server side rendering*. V tomto projektu je tato funkcionality používána například k autorizaci uživatele, zda má nebo nemá přístup na konkrétní stránky.

### 6.1. Autentifikace

Pro autentifikaci uživatelů jsme zvolili knihovnu s názvem *next-auth*. Tato knihovna poskytuje přívětivé prostředí starající se o takzvanou *session* autentifikaci. Hlavním důvodem pro použití této knihovny byl fakt, že podporuje přihlášení skrze různé providery jako jsou Google, Github, Facebook, Apple, apod. V našem projektu jsme použili první tři z uvedených. K aktivaci je nutné držet se dokumentace knihovny a nastavit podle ní *.env* soubor s proměnnými. Kostra tohoto souboru je k nalezení v kořenové složce projektu.

Knihovna je kompatibilní s mnoha různými databázemi, z nichž jsme vybrali stejnou databázi, kterou využívá i zbytek aplikace (PostgreSQL).

Pro autentifikaci a ověření dotazů na server byla vytvořena metoda *isValidRequest()*, která umí ověřit, zda je uživatel přihlášený, a zároveň ověřuje metodu dotazu. Jednotlivé položky dotazu jsou ověřovány pomocí knihovny *zod*, pomocí které lze vytvořit schémata proměnných na míru každému projektu a následně ověřit, zda určitá položka dotazu odpovídá danému schématu.

### 6.2. Uživatel

Následující podkapitoly obsahují základní informace pro práci s REST API aplikace, které pracuje s uživatelskými daty.

#### 6.2.1. Základní informace [/api/user/info/get]

Tento API endpoint bere z databáze jméno uživatele. Využívá se v sekci *Můj účet*, kde se jméno zobrazuje. Jeho parametrem je e-mail uživatele.

#### 6.2.2. Přidání skóre [/api/user/score/add]

Endpoint slouží pro přidání uživatelského skóre do databáze. Jeho parametry jsou id hry, e-mail uživatele, počet správných odpovědí a počet špatných odpovědí.

#### 6.2.3. Získání skóre [/api/user/score/get]

Endpoint slouží pro získání skóre konkrétního uživatele. Parametrem je pouze e-mail uživatele, na základě kterého vrátí pole polí, v nichž jsou uloženy skóre jednotlivých her, které uživatel již hrál.

#### 6.2.4. Resetování skóre [/api/user/score/reset]

Pomocí tohoto endpointu je možné vynulovat statistiky her, díky čemuž může uživatel zkusit hrát znovu a pokusit se udělat lepší statistiku. Parametrem je e-mail.



## 6.3. Hry

Následující podkapitoly obsahují základní informace pro práci s REST API aplikace, které pracuje s hrami. Získává data her z databáze, přidává hry do databáze, apod.

### 6.3.1. Generace [/api/games/generation]

Tento endpoint generuje data jednotlivých her a následně je ukládá do databáze. Je určený hlavně pro vývoj aplikace a neměl by být využíván uživateli aplikace. Jeho spuštění je náročné, a proto je aplikace navržena tak, aby nebylo nutné hry generovat v runtimu aplikace, tedy když aplikace běží v produkční verzi, ve chvíli, kdy je nějaká z her potřeba.

### 6.3.2. Volba správné odpovědi [/api/games/getChooseCorrectStage]

Při použití hry *Volba správné odpovědi* je volán tento endpoint, který vrátí objekt, na základě kterého se vygeneruje hra na frontendu. Parametry aplikace jsou e-mail, obtížnost a počet členů v mnohočlenu.

### 6.3.3. Stejně dvojice [/api/games/getEqualPairs]

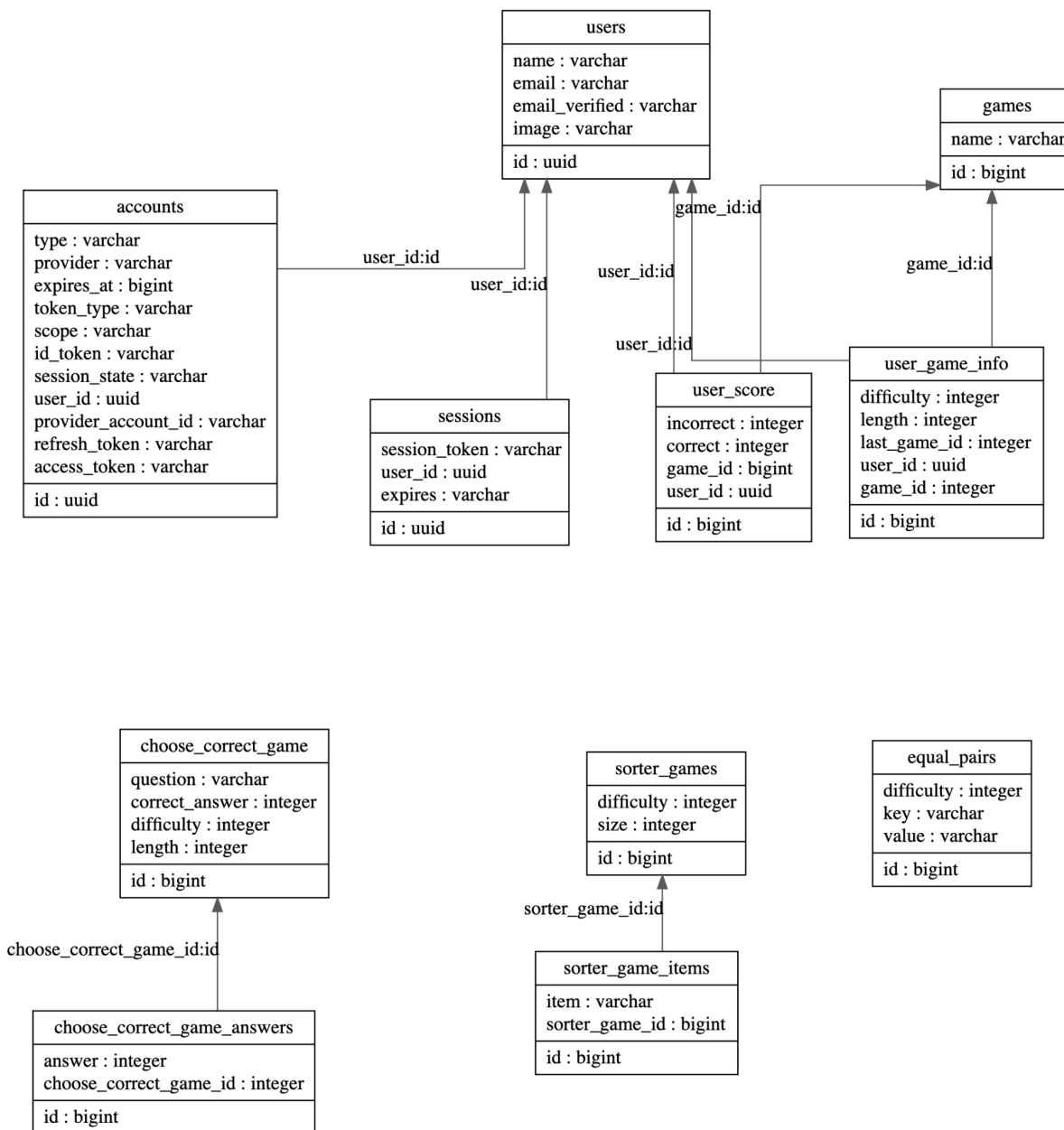
Pro získání sobě se rovnajících dvojic pro vygenerování hry na frontendu se používá tento endpoint. Parametry jsou id hry, e-mail, obtížnost a počet takovýchto dvojic.

### 6.3.4. Řazení kartiček [/api/games/getSorterGame]

Při použití hry *Řazení kartiček* je volán tento endpoint, který vrátí pole, na základě kterého se vygeneruje hra na frontendu. Parametry aplikace jsou e-mail, obtížnost a rozsah požadované hry.

## 7. Databáze

Tato kapitola se věnuje databázi projektu, pro kterou byla zvolena technologie PostgreSQL, a to nejen pro autentifikaci, ale i pro ostatní tabulky. Celkovou databázi popisuje následující schéma vygenerované pomocí programovacího prostředí společnosti JetBrains IntelliJ Idea.



Díky knihovně *next-auth* databáze obsahuje tabulky *accounts*, *sessions* a *users*. Struktura a předpis těchto tabulek lze nalézt v dokumentaci knihovny.

Struktura ostatních tabulek je uložena v souboru */lib/database/*. Zároveň lze v této složce nalézt předepsané SQL funkce pro práci s daty tohoto projektu.

Pro práci s PostgreSQL databází byla použita knihovna *pg*, která poskytuje potřebné metody k vykonávání dotazů do databáze.

## 8. Závěr

Po celkovém zhodnocení projektu, je jasné, že jsme zcela nesplnili zadání a to hlavně v různorodosti úloh. To vše přestože náš projekt podporuje 6 různých her ve všech požadovaných obtížnostech. Projekt byl tedy celkově dost náročný, ať už z hlediska toho, že bylo potřeba se seznámit s prostředím knihovny React.js a jeho backendovou nadstavbou Next.js, ale i kreativity a designu implementovaných her.

Zřejmě největší potíže způsobovala knihovna *next-auth*, která má sice dosti podrobnou dokumentaci, přesto bylo složité ji zprovoznit na našem projektu. Podle dokumentace má totiž oficiálně 2 verze, ale zároveň i další podverze, které se kombinují s předchozími verzemi.

I tak se nám povedlo vytvořit více než použitelnou aplikaci, která je připravena na případná rozšíření, jež by směřovala zejména k obohacení o další hry. Taková rozšíření by nebyla možná, pokud bychom nezajistili dostatečnou obecnost aplikace.

## 9. Bibliografie

- I. Obrázek učebních pomůcek - *Bobyho škola - Zdarma nejen pro prvňáčky* [online]. Copyright © [cit. 02.01.2023]. Dostupné z:  
[https://bobyho-skola.cz/wp-content/uploads/2019/05/school-2596090\\_1280B.jpg](https://bobyho-skola.cz/wp-content/uploads/2019/05/school-2596090_1280B.jpg)
- II. React.js komponenta podporující React Strict Mode - Getting React 18's "Strict mode" to work with react-beautiful-dnd | by William Bernting | Medium. *Medium – Where good ideas find you.* [online]. Dostupné z:  
<https://medium.com/@wbern/getting-react-18s-strict-mode-to-work-with-react-beautiful-dnd-47bc909348e4>
- III. Funkce pro serializaci a deserializaci objektu Map() - Forbidden - Stack Exchange. *Forbidden - Stack Exchange* [online]. Dostupné z:  
<https://stackoverflow.com/questions/29085197/how-do-you-json-stringify-an-es6-map>