

UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

WEBOVÁ APLIKÁCIA PRE VÝUČBU  
PROGRAMOVACIEHO JAZYKA POMOCOU INÉHO  
JAZYKA  
BAKALÁRSKA PRÁCA

2021  
MARTINA VESELÁ



UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

WEBOVÁ APLIKÁCIA PRE VÝUČBU  
PROGRAMOVACIEHO JAZYKA POMOCOU INÉHO  
JAZYKA  
BAKALÁRSKA PRÁCA

Študijný program: aplikovaná informatika  
Študijný odbor: informatika  
Školiace pracovisko: FMFI.KAI - Katedra aplikovanej informatiky  
Školiteľ: Ing. František Gyárfáš, PhD.

Bratislava, 2021  
Martina Veselá





Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Martina Veselá  
**Študijný program:** aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)  
**Študijný odbor:** informatika  
**Typ záverečnej práce:** bakalárska  
**Jazyk záverečnej práce:** slovenský  
**Sekundárny jazyk:** anglický

**Názov:** Webová aplikácia pre výuku programovacieho jazyka pomocou iného jazyka  
*Web application for learning of a programming language using another one*

**Anotácia:** Cieľom bakalárskej práce je navrhnúť a vytvoriť interaktívne webové prostredie pre výuku programovania v programovacom jazyku (napr. Java) za pomoci znalostí iného programovacieho jazyka (napr. Python). Prostredie ponúkne študentovi riešenia úloh v jemu známom jazyku a vedie ho k preprogramovaniu týchto úloh do druhého jazyka. Úlohy budú definované formou testmi riadeného programovania (TDD). Aplikácia bude obsahovať editory kódu vo vybraných jazykoch a testovanie na serveri pomocou jednotkových testov. Systém bude realizovaný pomocou technológií/nástrojov: PHP alebo Python (framework), MySQL, HTML5, CSS, JavaScript (jQuery, Bootstrap), knižnice pre testovanie, virtuálny server.

**Vedúci:** Ing. František Gyarfaš, CSc.  
**Katedra:** FMFI.KAI - Katedra aplikovanej informatiky  
**Vedúci katedry:** prof. Ing. Igor Farkaš, Dr.  
**Dátum zadania:** 04.10.2020

**Dátum schválenia:** 04.01.2021

doc. RNDr. Damas Gruska, PhD.  
garant študijného programu

.....  
študent

.....  
vedúci práce





## Abstrakt

Výsledkom tejto práce je webová aplikácia na výučbu jedného programovacieho jazyka pomocou iného programovacieho jazyka. Implementované sú jazyky *Python* a *Java*. Používatelia majú možnosť naučiť sa syntax bez toho, aby ich aplikácia zbytočne nanovo učila základné princípy programovania. Súčasťou výučby sú jednotkové testy.

Definícia lekcií je flexibilná, môžu mať predprogramovaný *Python* kód a zadanie pre *Javu* alebo naopak. Taktiež môže chcieť od používateľa, aby si sám napísal testy alebo postupoval technikou testmi riadeného programovania.

Aplikácia je pohodlná vďaka editorom kódu, zbiehaní kódov na serveri, ukladaním verzií a pomocným ťahákom.



# Abstract

The result of this thesis is a web application for learning a programming language using another programming language. The implemented languages are *Python* and *Java*. Users can learn a syntax of a new language without having to needlessly go through lessons that would try to teach him basic concepts of programming. This application includes unit tests.

The definition of lessons is flexible, they can have predefined code for *Python* and teach *Java* or the other way around. It can also require from the user to write their own unit tests or to program using test driven development.

The application is comfortable for its users thanks to code editors, running the codes on a server, version control and helpful cheat-sheets.



# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Východiská</b>	<b>3</b>
1.1 Existujúce riešenia . . . . .	3
1.1.1 CodeAcademy . . . . .	4
1.1.2 CodeWars . . . . .	5
1.2 Potrebná teória . . . . .	6
1.2.1 Webové prostredie . . . . .	6
1.2.2 Editor kódu . . . . .	6
1.2.3 Client-server . . . . .	6
1.2.4 Jednotkové testy . . . . .	7
1.2.5 Test Driven Development . . . . .	7
1.2.6 Virtual Machine . . . . .	8
1.2.7 Relačná databáza . . . . .	8
1.3 Použité technológie . . . . .	8
1.3.1 Backend . . . . .	8
1.3.2 Frontend . . . . .	10



# Zoznam obrázkov

1.1	Ukážka webového editora CodeAcademy . . . . .	4
1.2	Ukážka webového editora CodeWars . . . . .	5
1.3	Textový editor . . . . .	6
1.4	Client-server model . . . . .	7
1.5	Diagram procesu testmi riadeného programovania . . . . .	7
1.6	Ukážka jednoduchého jednotkového testu v jazyku Python . . . . .	9
1.7	Ukážka jednoduchého jednotkového testu v Jave . . . . .	9
1.8	Princíp funkcionality Redux state . . . . .	11
1.9	Demo ukážka vytvorenia CodeMirror editoru . . . . .	11



# Úvod

Cieľom tejto práce je navrhnúť a vytvoriť interaktívne webové prostredie pre výučbu programovania v jednom programovacom jazyku za pomoci znalostí iného programovacieho jazyka. Prostredie bude poskytovať úlohy spolu s riešením v jednom programovacom jazyku a návod na preprogramovanie daného problému do druhého programovacieho jazyka. Aplikácia je dostatočne flexibilná, aby podporovala úlohy na písanie testov a na programovanie technikou testmi riadeného programovania (TDD - Test Driven Development). Pre túto prácu budú použité programovacie jazyky Python a Java. Aplikácia bude obsahovať editory kódu v daných jazykoch, spolu s testmi pre dané úlohy a výsledok po zbehnutí kódu. Testovanie bude prebiehať na serveri pomocou jednotkových testov.

V dnešnej dobe stále viac a viac ľudí má záujem o online kurzy, takže existuje veľa interaktívnych nástrojov na výučbu programovacích jazykov, ale nenašla som žiaden, ktorý by využíval iný, používateľovi už známy jazyk na výučbu pre neho nového jazyka. Výučba týmto spôsobom má mnohé výhody. Ak už používateľ ovláda základy programovania, väčšina nástrojov na výučbu by sa mu ich snažila nanovo vysvetliť. Táto aplikácia sa sústreďuje na rozdiely v programovacích jazykoch, čiže naučí používateľa danú syntax.

Vďaka interaktívnemu elementu si používateľ môže všetko hneď vyskúšať a naučiť sa danú problematiku praxou. V niektorých zadaniach si môže meniť aj kód v oboch jazykoch a takisto zmeniť testy, ktoré budú zbiehané. Taktiež sa naučí písať testy a zažije si výhody (ale aj nevýhody) testmi riadeného programovania. Okrem úloh sa dá aplikácia využiť ako sandbox, umožňuje slobodu prepisovania kódu a testov v oboch jazykoch.

V tejto práci najprv ukážem podobné existujúce riešenia, vysvetlím potrebnú teóriu na pochopenie práce a popíšem použité technológie. Ďalej definujem ciele aplikácie, vysvetlím jej architektúru, databázový model, vybrané vnútorné procesy a implementáciu. Nasleduje popis používania aplikácie s ukážkami. V predposlednom rade sú poznámky a poznatky z používateľského testovania a na záver výsledky práce a možné vylepšenia a rozšírenia.





# Kapitola 1

## Východiská

V tejto kapitole budú ukážky podobných existujúcich riešení, teória potrebná na pochopenie práce a technológie použité v aplikácii.

Užitočný materiál na naučenie sa Javy pomocou Pythonu je `java4python` [?] napísaný autormi *Brad Miller* a *James Riely*. Inšpirovala som sa ním pri písaní lekcí pre moju aplikáciu.

### 1.1 Existujúce riešenia

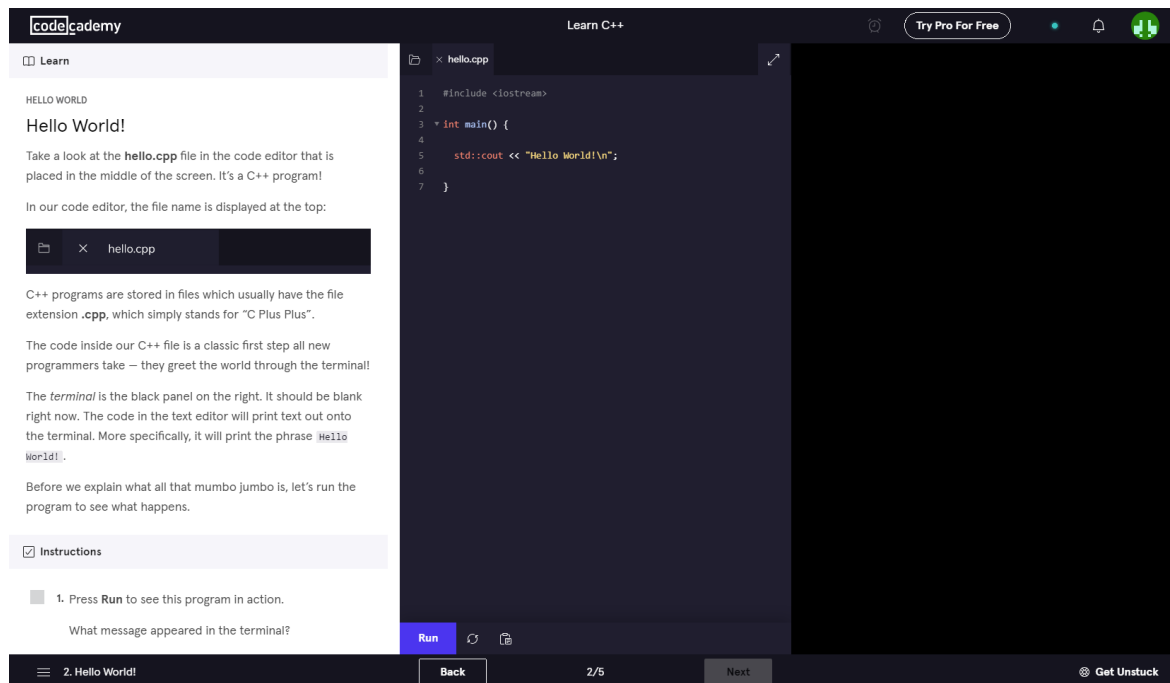
Nepodarilo sa mi nájsť žiadnu existujúcu aplikáciu, ktorá by učila programovací jazyk pomocou iného programovacieho jazyka.

Existuje veľmi veľa online kurzov vyučujúcich nejaký programovací jazyk. Väčšina takýchto stránok sa sústreďuje na výučbu web developmentu. Takmer všetky stránky si vyžadujú registráciu používateľa. Pri niektorých je to veľmi jednoduchý a rýchly proces, iné si pýtajú veľa informácií o používateľovi.

Drvivá väčšina stránok má niektoré kurzy zadarmo (poväčšine začiatocnícke) a ostatné kurzy sa odomknú mesačným poplatkom.

Niektoré programovacie jazyky majú svoju vlastnú stránku na interaktívnu výučbu. Príkladom takejto stránky by bol <https://tour.golang.org> pre programovací jazyk Go.

### 1.1.1 CodeAcademy



Obr. 1.1: Ukážka webového editora CodeAcademy

CodeAcademy je online interaktívna platforma, ktorá ponúka kurzy programovacích jazykov.

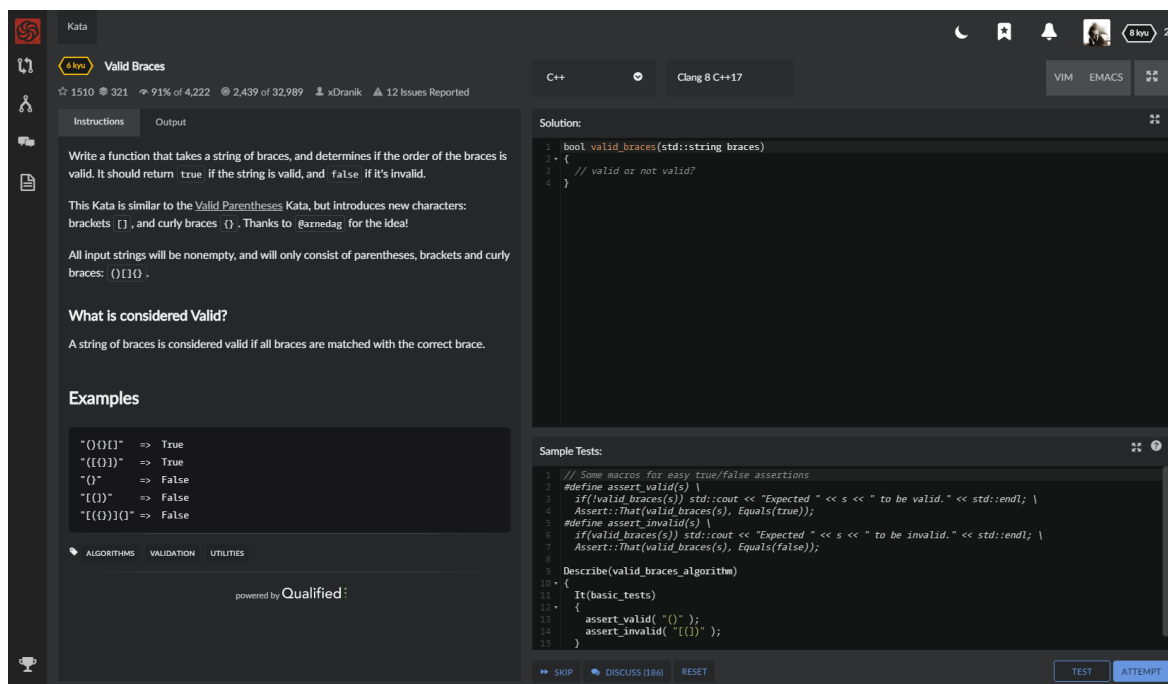
Registrácia je jednoduchá, dá sa pripojiť priamo cez účet na LinkedIn, Google, Facebook, Github alebo Twitter.

Momentálne ponúka kurzy v jazykoch HTML, CSS, Python, JavaScript, Java, SQL, Bash/Shell, Ruby, C++, R, C#, PHP, Go, Swift a Kotlin.

Na prístup ku všetkým kurzom treba platiť mesačný poplatok.

Na rozdiel od mojej aplikácie nevyužíva na výučbu ďalší programovací jazyk. Tak-tiež sa nedá pozrieť si zbierané testy ani ich modifikovať.

## 1.1.2 CodeWars



Obr. 1.2: Ukážka webového editora CodeWars

CodeWars je tiež online interaktívna platforma s kurzmi programovacích jazykov.

Jej hlavnou výhodou je, že, keď splníte zadanie, ukáže Vám riešenia ostatných používateľov, na ktorých sa dá naučiť, čo robiť lepšie alebo koľko je možných spôsobov implementácie.

Registrovať sa dá buď emailom a heslom alebo cez účet na GitHub.

Ponúkané kurzy sú v mnohých jazykoch: Clojure, CoffeeScript, C, Coq, C++, C#, Crystal, Dart, Elixir, F#, Go, Groovy, Haskell, Java, JavaScript, Kotlin, Lean, Lua, NASM, PHP, Python, Racket, Ruby, Rust, Scala, Shell, SQL, Swift a TypeScript.

Ďalej ponúka ďalšie jazyky, ktorých kurzy sú zatiaľ v beta: Agda, BF, CFML, COBOL, CommonLisp, Elm, Erlang, Haxe, Idris, Factor, Forth, Fortran, Julia, Nim, Objective-C, OCaml, Pascal, Perl, PowerShell, Prolog, PureScript, R, Raku, Reason, Solidity a VB.

Orientácia na stránke je trochu zložitá pre nového používateľa. Kurzy sú zadarmo, ale za mesačný poplatok sa odomknú prémiové funkcionality ako vyššie spomenuté porovnávanie riešení, výsledky testov ukázané priebežne, nie až po dobehnutí, beta funkcionality atď.

Síce neumožňuje výučbu programacieho jazyka pomocou iného jazyka, ale má tie isté úlohy vo viacerých jazykoch a testy sú viditeľné a dokonca aj editovateľné.

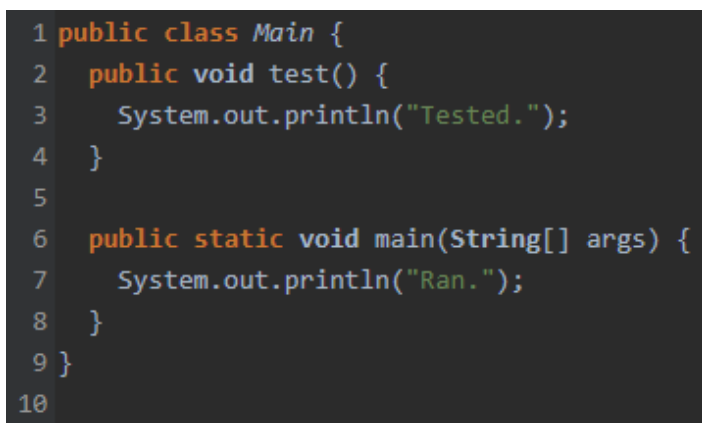
## 1.2 Potrebná teória

### 1.2.1 Webové prostredie

Webové prostredie je také, ku ktorému má používateľ prístup cez webový prehliadač. Vďaka tomu nemusí nič inštalovať a hneď po príchode na stránku má prístup k aplikácii. Z pohľadu developera vyvíjať webovú aplikáciu môže byť jednoduchšie, keďže existuje veľa knižníc a vývojových rámcov. Má to, ale svoje obmedzenia, každé webové prostredie používa HTML, CSS a JavaScript. Bez nich sa nedá zaobísť a nedajú sa nahradiť.

### 1.2.2 Editor kódu

Editor kódu je druh textového editora, ktorý je špecializovaný na písanie zdrojového kódu konkrétneho programovacieho jazyka. Dobrý editor používateľom prácu zjednodušuje pomocou zafarbenia, automatického odsadzovania, dopĺňaním koncových zátvoriek atď. Môže mať vlastné klávesové skratky a tiež môže byť schopný upozorniť používateľa na chybu. Taktiež môže ovládať syntax daného programovacieho jazyka. Väčšina má očíslované riadky pre jednoduchšiu orientáciu.

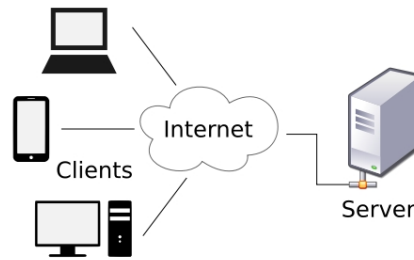


```
1 public class Main {  
2     public void test() {  
3         System.out.println("Tested.");  
4     }  
5  
6     public static void main(String[] args) {  
7         System.out.println("Ran.");  
8     }  
9 }  
10
```

Obr. 1.3: Textový editor - viď. rôznymi farbami zvýraznené typy kľúčových slov. V tejto ukážke je použitá populárna farebná téma Darkula

### 1.2.3 Client-server

Client-server model je model aplikačnej štruktúry, kde jeden alebo viac serverov poskytuje funkcionality, alebo servis jednému, alebo viacerým klientom. Pováčšine klient a server nezdieľajú spoločný hardvér, ale komunikujú medzi sebou na rôznych hardvéroch, a to buď cez domácu sieť alebo internet.



Obr. 1.4: Príklad client-server modelu

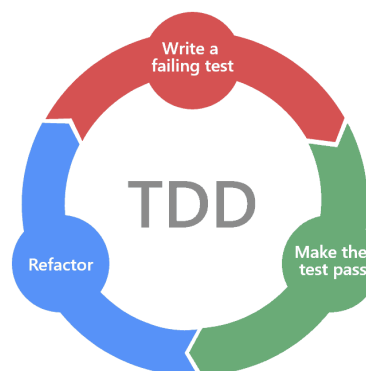
### 1.2.4 Jednotkové testy

[?] Jednotkové testy (unit test), bežne automatizované, kontrolujú funkčnosť kódu, a to po najmenších celkoch. Písanie testov nie je najpríjemnejšia činnosť, ale je veľmi užitočná na uistenie sa, že aplikácia funguje, ako má. Dá sa tak predchádzať chybám v okrajových prípadoch a pri zmenách kódu.

### 1.2.5 Test Driven Development

[?] [?] Testmi riadené programovanie (TDD) je metodológia programovania, pri ktorej sa počas celého vývoja využívajú testy. Postup programovania je nasledujúci: Najprv sa napíše test, ktorý musí zlyhať. Potom sa napíše metóda, ktorá ho splní a nepokazí doterajšie testy. Ak je to potrebné, zrefaktoruje sa. Potom sa proces opakuje, až pokým metóda nespĺňa dané požiadavky.

Táto metodológia je málokedy využívaná v praxi. Na začiatku projektu spomaľuje prácu a triviálne testy sa zdajú byť zbytočné. Z dlhodobého hľadiska sa oplatí, keďže výsledný program by mal mať menej chýb a mal by sa dať ľahšie upravovať.



Obr. 1.5: Diagram procesu testmi riadeného programovania

### 1.2.6 Virtual Machine

Virtuálna mašina je emulátor, ktorý používa softvér namiesto fyzického počítača. Umožňuje zbiehanie programov alebo aj celého operačného systému. Pri zbiehaní operačného systému umožní prístup k inému OS, než je nainštalovaný, napr. pre developerské účely. Môže taktiež simulovať iný hardvér. Veľkou výhodou virtual mašiny je, že malvér alebo škodlivý zásah, ktorý sa dostane do virtual mašiny nemá prístup k danému počítaču. Pre túto prácu využívam virtuálnu mašinu na to, aby používatelia nemali prístup k serverovému počítaču, keďže na ňom môžu zbiehať akýkoľvek kód (v programovacích jazykoch Java a Python).

### 1.2.7 Relačná databáza

Relačná databáza sa dá považovať za sériu tabuliek, na skladovanie dát. Každý riadok reprezentuje jeden záznam v tabuľke.

## 1.3 Použité technológie

### 1.3.1 Backend

#### Python

Aplikácia je programovaná v programovacom jazyku Python [?]. Python je interpretovaný high-level programovací jazyk, ktorý je známy jednoduchosťou čítania kódu. Podľa mňa je to ideálny jazyk na rýchle, menšie projekty, ale pri veľkých dielach môže byť mätúci, keďže premenné nemajú pevne zadaný typ. Backend mojej aplikácie nie je príliš rozsiahly, preto som sa rozhodla na neho použiť Python.

#### Flask

[?] Flask je webový framework pre Python. Slúži na prepojenie medzi backendom a frontendom. Na prenos informácií v mojej aplikácii využívam *http* requesty *GET* a *POST*.

## Python unittest

Vývojový rámec pre jednotkové testy v programovacom jazyku Python. [?]

```
1  import unittest
2
3  class Test(unittest.TestCase):
4
5      def test(self):
6          self.assertEqual(1 + 1, 2)
7
8  if __name__ == '__main__':
9      unittest.main()
10
```

Obr. 1.6: Ukážka jednoduchého jednotkového testu v jazyku Python

## JUnit

JUnit je vývojový rámec pre jednotkové testy v programovacom jazyku Java [?]. Python unittest ním bol inšpirovaný.

```
1  import static org.junit.Assert.*;
2  import org.junit.Test;
3
4  public class Test {
5
6      @Test
7      public void test() {
8          assertEquals(1 + 1, 2);
9      }
10 }
```

Obr. 1.7: Ukážka jednoduchého jednotkového testu v Java

## Docker

V tejto aplikácii sú používané jednotkové testy. Zbiehať cudzí kód nie je bezpečné, keďže by používateľ dostal prístup k počítaču, na ktorom beží server. Preto používam Docker [?], aby testy zbiehali virtuálne. Docker dostane text z editora aplikácie a pošle ho danému testovaču (Python - UnitTest, Java - JUnit). Po zbehnutí testu sa výsledok pošle na stránku, kde používateľ môže vidieť, kde urobil chyby.

Docker je efektívnejší, než bežné virtuálne mašiny, keďže všetky jeho kontajnery zdieľajú servery jedného jadra operačného systému. Kontajnery sú navzájom izolované a obsahujú pre seba potrebný softvér, knižnice a konfiguračné súbory. Môžu medzi sebou komunikovať pomocou kanálov.

## PostgreSQL databáza

[?] Je to relačná databáza v ktorej ukladám dáta potrebné pre aplikáciu. Najdôležitejšími sú lekcie; aplikácia potrebuje odniekiaľ prečítať text lekcie a predpripravené kódy. Ďalej mám tabuľku používateľov, ktorým sa po prihlásení budú ukladať verzie zbiehaného kódu, ak sa k nim budú potrebovať vrátiť.

## Psycopg2

[?] Je to najpopulárnejší adaptér PostgreSQL databázy pre programovací jazyk Python. Používam ho na pripájanie sa k mojej databáze a jej upravovanie.

## 1.3.2 Frontend

### TypeScript

[?] TypeScript je open-source programovací jazyk od Microsoftu, nadstavba JavaScriptu. Pridáva statické typy, vďaka čomu je kód čitateľnejší, dokumentácia je lepšia a jeho funkcionality sa dá ľahšie kontrolovať. TypeScript podporuje ES6 moduly. ES6 - ECMAScript 2015 je JavaScriptový štandard, čo pomáha stránkam fungovať na rôznych webových prehliadačoch a pridáva užitočné funkcionality ako konštanty, šípkové funkcie, for cyklus, predvolené parametre a veľa ďalších.

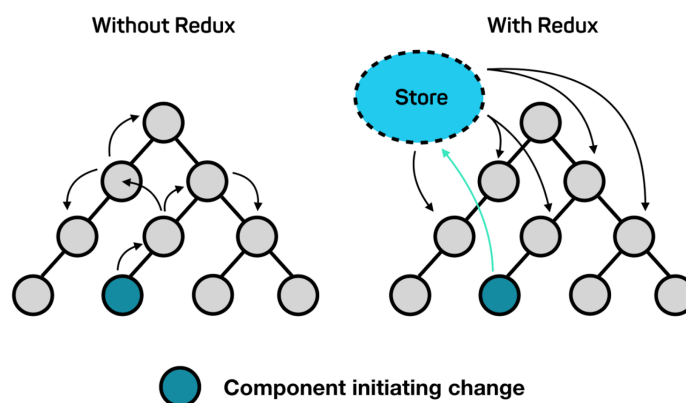
### React

[?] React je open-source JavaScript knižnica pre stavbu používateľských rozhraní. Vie byť použitý ako báza pre jedno-stránkové webové stránky. Pre viac-stránkové stránky je potrebné použiť ďalšiu knižnicu na routing, ako napríklad React Router [?], ktorý aj používam v tomto projekte. Kód reactu je tvorený pomocou komponentov. Keď sa komponent renderuje, dajú sa mu poslať hodnoty tzv. props. Komponenty môžu byť funkcie alebo triedy.

### React Redux

[?] Redux sa používa viac menej na ukladanie globálnych premenných. Vďaka nemu si komponenty nemusia postupne podávať premenné ale môžu ich vyťahovať priamo z Redux *state*.





Obr. 1.8: Princíp funkcionality Redux state

## React Bootstrap

[?] Najpopulárnejší frontend framework prerobený pre React. Rozloženie stránky sa robí veľmi jednoducho za pomoci Kontajnerov. Bootstrap obsahuje užitočné komponenty, ktoré sa dajú ľahko pridať na stránku ako tlačidlá, dropdown menu, navigácia, tabuľky a mnoho ďalších. Netreba veľa úsilia na to, aby stránka vyzerala dobre.

## CodeMirror

[?] Verzatilný textový editor implementovaný v JavaScripte špecializovaný na editovanie kódu. Všetky editory v mojej aplikácii sú vytvorené pomocou neho. Podporuje väčšinu programovacích jazykov. Jeho výzor sa dá prispôbiť a dá sa vybrať z populárnych farebných schém. Vie využívať klávesové skratky, ktoré sa dajú aj meniť. Automaticky odsadzuje riadky a zvýrazňuje zodpovedajúce zátvorky.

Po dokončení tejto práce vyšla nová verzia CodeMirror.

## This is CodeMirror

```

1 <!-- Create a simple CodeMirror instance -->
2 <link rel="stylesheet" href="lib/codemirror.css">
3 <script src="lib/codemirror.js"></script>
4 <script>
5   var editor = CodeMirror.fromTextArea(myTextarea, {
6     lineNumbers: true
7   });
8 </script>

```

Obr. 1.9: Demo ukážka vytvorenia CodeMirror editoru dostupná na ich stránke

