Webová aplikace pro komunikaci a správu dat z kvantového generátoru náhodných čísel

Jan Solař

Bakalářská práce
2021

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Jan Solař**
Osobní číslo:	**A17574**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Informační a řídicí technologie**
Forma studia:	**Prezenční**
Téma práce:	**Webová aplikace pro komunikaci a správu dat z kvantového generátoru náhodných čísel**
Téma práce anglicky:	**A Web Application for Communication and Data Control from a Quantum Random Number Generator**

Zásady pro vypracování

1. Nastudujte potřebnou terminologii pro tvorbu webových aplikací.
2. Nastudujte terminologie kvantového generování náhodných čísel.
3. Zvolte potřebné metody a prostředy pro tvorbu aplikace.
4. Navrhněte samotnou aplikaci pro data z kvantového generátoru náhodných čísel.
5. Implementujte a otestujte vytvořenou aplikaci.
6. Vhodně reprezentujte a vyhodnoťte vysledky.

Forma zpracování bakalářské práce: **Tištěná/elektronická**

Seznam doporučené literatury:

1. NIELSEN, Michael A. a Isaac L. CHUANG. Quantum computation and quantum information. 10th Anniversary ed. Cambridge: Cambridge University Press, 2010, xxxi, 676 s. ISBN 9781107002173.
2. MEGLICKI, Zdzislaw. Quantum computing without magic: devices. Cambridge, MA: MIT Press, c2008, 1 on-line zdroj (xx, 422 p.). Scientific and engineering computation series. ISBN 9780262288187. Dostupné také z: http://ieeexplore.ieee.org/xpl/bkabstractplus.jsp?bkn=6267464
3. DAYLEY, Brad. Node.js, MongoDB and AngularJS web development. Upper Saddle River: Addison Wesley, [2014], xiv, 647 s. Developer's library. ISBN 9780321995780.
4. KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015, 496 s. ISBN 9788025143520.
5. BENDAT, Julius S. a Allan G. PIERSOL. Random data: analysis and measurement procedures. 3rd ed. New York: John Wiley, 2000, xvii, 594 s. Wiley series in probability and statistics. ISBN 0471317330.

Vedoucí bakalářské práce: **Ing. Petr Žáček**
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **15. ledna 2021**
Termín odevzdání bakalářské práce: **17. května 2021**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 15. ledna 2021

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Jan Solař, v.r.
podpis studenta

ABSTRAKT

Bakalářská práce se zabývá analýzou webové aplikace pro komunikaci s kvantovým generátorem náhodných čísel a její následnou implementací. Aplikace bude sloužit jako pomůcka při nauce o náhodných číslech. Teoretická část se zabývá zkoumáním metod vývoje moderních webových aplikací a generováním náhodných čísel. V praktické části je popsán návrh aplikace a následná implementace včetně testování.

Klíčová slova: Náhodná čísla, Python, Django framework, MVT, webové aplikace, SQLite

ABSTRACT

This Bachelor thesis is focused on analysis of web application for communication with quantum random numbers generator. Application is meant to be a learning tool for better understanding of random numbers. In the theoretical part the modern methods of web applications development as well as principles of generating random numbers are clarified. The practical part is devoted to a design of new application and implementation with testing are described here.

Keywords: Random numbers, Python, Django framework, MVT, web applications, SQLite

Rád bych tímto poděkoval vedoucímu mé bakalářské práce, panu Ing. Petru Žáčkovi, za poskytnuté náměty pro zlepšení a za veškerou poskytnutou pomoc.

OBSAH

ÚVOD

Člověk by si měl v dnešní době uvědomovat, jak jsou v dnešní době náhodná čísla důležitá. Nyní nemám na mysli hazard, nýbrž kybernetickou bezpečnost, kde náhodná čísla hrají velmi důležitou roli, například při šifrování citlivých dat. Je proto důležité, aby byla co nejmenší pravděpodobnost, aby někdo nechtěně uhádl náhodné klíče a získal tak přístup k cizím datům.

Proto tato práce využívá kvantový generátor náhodných čísel, který funguje na principu kvantových jevů. Tento generátor je pomocí webové aplikace propojen s uživatelem a ten si tak může vyzkoušet, jak vypadají opravdová náhodná data. Sepsaná práce se sestává z teoretické a praktické části.

V teoretické části jsou rozebrány současné principy tvorby webových stránek, zkráceně webů. Jsou zde nastíněny prostředky, jejichž pomocí dochází k realizaci webů, a to jak statické, tak i ty dynamické. U zvoleného frameworku pro vývoj jsou rozepsány bezpečnostní opatření, kterými disponuje. Dále jsou zde popsány rozdíly mezi pseudo-náhodnými a náhodnými čísly a také způsoby, jakými se tyto data získávají.

Praktická část se poté zabývá návrhem aplikace pro komunikaci s kvantovým generátorem náhodných čísel. Jsou zde popsány předpoklady, které by měla aplikace naplňovat. Poté je rozepsaná implementace, je vysvětlena funkčnost jednotlivých částí a výstup z testování. V implementaci je dbáno jak na moderní a vzhledný design, tak i funkčnost a hladký chod aplikace.

Zhotovená aplikace by měla jako pomůcka při výuce. Studentům bude umožněno vyzkoušet si práci s náhodnými daty a prozkoumání či ověření poznatků, které se vztahují k entropii či náhodným číslům.

I. TEORETICKÁ ČÁST

1 TVORBA WEBOVÝCH APLIKACÍ

V této části se text věnuje teorii ohledně tvorby webových aplikací a příklady používaných technologií.

1.1 Webové aplikace

Od aplikací běžící na lokálním zařízení se ty webové liší odpadající nutností instalace používaného hardware a software. Uživatel tak má přístupnou aplikaci, která může komunikovat se zařízeními vysokých částek či výpočetně náročnými bez povinnosti vlastnit drahé zařízení, na kterém je aplikace závislá, nebo disponovat výkonným zařízením. Vše může mít přístupné díky Internetu přes webový prohlížeč.

1.1.1 HTML

HTML (HyperText Markup Language) je značkovací jazyk, jehož pomocí se tvoří webové stránky. Slouží k vyjádření struktury dokumentu, který je dostupný přes internet či z lokálního úložiště. Jeho základními prvky jsou tagy, které dávají textu jeho význam – ať už se jedná o nadpis či odstavec. Další jeho funkcionalitou je vytváření odkazů na další HTML dokumenty. [1]

HTML prošlo velkými změnami od svého založení. Od zobrazení textu, přes formuláře, tabulky, až po stylování, multimédia, skriptování na straně klienta, rozpoznání jazyka nebo zvláštní znaky pro matematické operace. Ruku v ruce s těmito změnami přicházely i změny v tazích. [1]

Nejnovější verzí je HTML5, se kterou dorazily změny jako například jednoduchá deklarace dokumentu, popisnější zdrojový kód (nahrazení *divů* tagy jako jsou *header, footer, section, article, nav, aside* a vlastní datové atributy), úložiště na straně klienta, lepší uživatelské rozhraní (v kombinaci s CSS3), lepší formuláře nebo třeba lepší přístupnost pro lidi s omezením, kteří využívají programy jako třeba čtečky obrazovky, za pomocí nových elementů, které usnadňují zpracování obsahu. Velikou předností nové verze HTML je zpětná kompatibilita, díky které jsou námi vytvořené stránky zobrazitelné na kterémkoliv prohlížeči. [2]

1.1.2 CSS

V nezkrácené verzi Cascading Style Sheets (česky kaskádové styly) je technologie upravující vzhled HTML dokumentu. Pomocí CSS se tedy definuje, jak budou jednotlivé prvky zobrazeny. Je možné definovat výšku, šířku, pozici, barvu textu i pozadí, styl a velikost písma a další.

Spolu s novou verzí HTML vyšla i vylepšená verze CSS3. Díky ní je možné definovat zaoblené okraje, stíny anebo přechody, které přidávají na pocitu animace. Zároveň byly přidány nové selektory pro identifikaci lichých a sudých řádků tabulky, všech vybraných zaškrtávacích políček a třeba i poslední odstavec ve skupině. Dochází tak k redukci zdrojového kódu a k usnadnění stylizace HTML dokumentu. [2][3]

1.1.3 Bootstrap

Ačkoliv nová verze CSS přinesla spoustu zlepšení a zjednodušení, pořád může být design internetových stránek časově velmi náročný. Proto vznikají různé šablony, které mají již definovaný vzhled a programátor se tak pouze stará o přiřazení příslušných atributů prvkům dokumentu.

Jako příklad je zde uveden Bootstrap. Jedná se open-source framework, obsahující šablony pro CSS i JavaScript. Je s ním možné měnit vzhled širokému spektru prvků od formulářů, přes tlačítka až po tabulky. Bootstrap navíc přichází s možností vlastního rozložení dokumentu, a to pomocí systému mřížky, která je rozdělena až na dvanáct sloupců (kde každý jde opět rozdělit na dalších dvanáct a rekurzivně neustále dál) a díky novým vlastnostem CSS3 lze definovat šířku buněk na základě šířky displeje zařízení. Z toho plyne, že díky Bootstrapu je možné vytvořit responsivní design, který je obzvláště v dnešní době plné různých velikostí a rozlišení obrazovek velmi žádaná vlastnost. [4]

1.1.4 JavaScript

Z důvodu zvyšující se poptávky po interaktivně webových stránek na konci minulého století vznikl JavaScript. Jedná se o programovací jazyk běžící na straně klienta, nikoliv serveru. Z důvodu bezpečnosti není pomocí JavaScriptu možné upravovat soubory na lokálním zařízení, tím pádem se nemusí uživatel obávat smazání dat z disku, na druhou stranu nelze čistě pomocí JavaScriptu ani centrálně shromažďovat data a využívat jej tak pro vývoje aplikací, jako např. diskusní fórum nebo počítadlo návštěv stránek. [5]

Uvedená fakta sice omezují okruhy možného použití JavaScriptu, přesto je však možné pomocí něj razantně zvýšit interaktivnost stránek. Je možné sledovat pohyb kurzoru a jeho akce a na základě toho na ně reagovat, dynamicky upravovat atributy HTML elementů, vytváření inteligentních formulářů za pomoci kontroly dat vyplněných uživatelem, práce s okny webového prohlížeče, využívání informací ohledně aktuálního data a času, ale také dynamická komunikace s backendem bez nutnosti obnovování stránek, což bývá označováno jako AJAX. Nejedná se o programovací jazyk ani o framework, ale o jakýsi souhrn technik pro vývoj webu na straně klienta za účelem tvorby asynchronní webové aplikace. [5]

V dnešní době vzniká na základě JavaScriptu spousta frameworků a knihoven, a to za účelem zjednodušení a zefektivnění práce. Jako příklad je dále v textu rozepsána knihovna jQuery, jako příklady frameworků bych uvedl JNode, Vue.js, Angular, React, Blazor a další.

1.1.5 JQuery

Jedná se o rychlou, malou (v komprimované verzi má 30kB) a na funkce bohatou open source knihovnu JavaScriptu. Zjednodušuje práci s HTML dokumenty, zpracování událostí, animace a AJAX. V kombinaci s univerzálností a rozšiřitelností mění způsoby, jakými miliony lidí používají JavaScript. [6][7]

1.1.6 MySQL

Pochází od společnosti Oracle Corporation a je to open source systém řízení databází, což jsou strukturované kolekce dat. Ať už se jedná o informace ohledně nákupního košíku nebo uživatelská data, většina těchto informací by měla být uložena v databázi. [8]

V MySQL jsou databáze relačního typu, to znamená, že jednotlivé tabulky lze propojovat různými vztahy za pomoci primárních a cizích klíčů. Zároveň není potřeba ukládat všechna data do jedné monolitické tabulky, ale pro větší přehled je lze rozdělit do více tabulek, kdy každá tabulka obsahuje pouze nezbytné informace a jednotlivé tabulky jsou poté na základě společných informací propojeny právě přes relace. [8]

Typů relací mezi tabulkami je hned několik – 1:1, 1:N, N:M. Slovně vysvětleno: jeden záznam z tabulky *A* může mít přiřazen pouze nanejvýš jeden záznam z tabulky *B* nebo jeden záznam z tabulky *A* může mít přiřazeno více záznamů z tabulky *B*. Obdobně je to i s posledním typem relací. [8]

1.1.7 SQLite

Je to knihovna implementující samostatný, bez-serverový transakční databázový stroj. Kód pro použití SQLite je volně dostupný pro jakýkoliv účel – osobní či komerční. [9]

Lze tedy říct, že SQLite je vestavěná databáze a na rozdíl od většiny ostatních SQL databází nemá samostatné serverové procesy. Čte a zapisuje přímo z/do souborů na disku. Kompletní databáze se všemi tabulkami, indexy, triggery a procedurami je obsažená v jednom jediném souboru. Formát souborů je kompatibilní napříč různými platformami. Je tedy možné soubor kopírovat mezi 32-bitovými a 62-bitovými systémy nebo mezi little-endian a big-endian architekturami. Díky práci se soubory je možné nahlížet na SQLite ne jako náhradu za jinou databázi, spíše jako na náhradu za knihovny, kterými zapisujeme do souborů a také z nich čteme. Tomuto názoru nahrává i fakt, že při správném nastavení může být tato knihovna rychlejší než systémový I/O. [9]

1.1.8 Frameworky

Prozatím se práce zabývala zobrazováním HTML dokumentu, jeho rozvržením, stylizováním, doplňováním o animace a interakci, případně ukládání dat na straně serveru.

Další část se zkoumá implementaci logiky na straně serveru. K tomu by stačily programovací jazyky, ale vývoj šel ještě dál a tak vznikly frameworky. Jejich úkol je jednoduchý – ulehčit a zefektivnit vývoj webových aplikací. Pokud má zákazník zájem o statické stránky informativního charakteru, není potřeba využívat framework. Dá se bez něj obejít. Ale v dnešní době už je většinou internet dynamický a koluje na něm obrovské kvantum dat. Často lze HTML dokument použít jako šablonu a jediné, co se mění, jsou zobrazovaná data. V takovém případě už je lepší uvažovat o využití logiky na straně serveru, kdy se na základě příchozího požadavku vyberou, zpracují, uloží nebo pošlou příslušná data. Programátor má mnoho možností výběru postupu a je tak čistě na něm, jaký programovací jazyk, respektive framework zvolí.

V další části jsou uvedeny příklady celosvětově používaných programovacích jazyků a z nich vycházejících frameworků.

1.1.8.1 C#

Objektově a komponentně orientovaný jazyk umožňující vytváření mnoha typů bezpečných a robustních aplikací běžících na .NET ekosystému. C# je členem „rodiny C“ díky čemuž je velmi podobný jazykům C, C++, Java a JavaScriptu. Skvělou vlastností tohoto jazyka je jeho neustálý vývoj ze strany Microsoftu. [10]

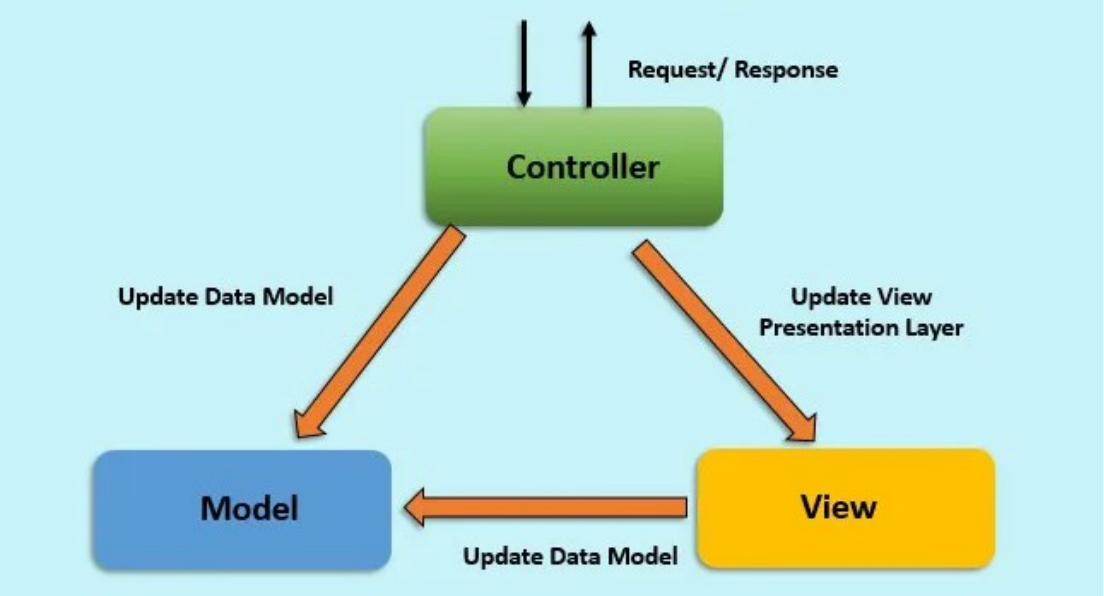
Vlastnosti pomáhající vytvořit robustní aplikace jsou např.:

- Garbage kolektor – automaticky uvolňuje paměť vyhrazenou pro objekty, které jsou nakonec nedosažitelné či nepoužité
- Ošetření nullových typů – zabránění proměnným v odkazování na neexistující objekty
- Zpracování výjimek – strukturovaný a rozšiřitelný přístup k detekci chyb a zotavení

1.1.8.1.1 ASP.NET

Open source framework pro tvorbu webových aplikací a služeb pomocí .NET a C# podporovaný napříč různými platformami. Rozšiřuje platformu .NET nástroji a knihovnami určenými pro tvorbu webových aplikací, API a mikroslužeb. Přidává základní framework pro zpracování požadavků v jazyce C# nebo F#, šablonový systém známý jako Razor pro tvorbu dynamických webových stránek, autentizační systém nebo třeba MVC architekturu. [11][12]

MVC (Model-View-Controller) architektura slouží k lepší organizaci kódu a je využívána v různých jazycích. Hlavní myšlenkou je, že každá část kódu má svůj účel – některý kód slouží k reprezentaci objektů, jiný obsahuje funkcionalitu. Pro utříbení kódu a větší přehlednost při programování je proto lepší rozdělovat na skupiny.



Obrázek 1 Architektura MVC [13]

Model představuje informace o objektech z databáze, jejich vlastnosti. Pro implementaci se používají třídy C#.

View slouží jako uživatelské rozhraní. Předává uživateli data z modelu a umožňuje jejich modifikaci. V ASP.NET je view zastoupený HTML, CSS a Razor syntaxí, která zjednodušuje komunikaci s modely a controllery.

Controller zpracovává požadavky ze strany klienta a vrací příslušnou odpověď. Nachází se zde většina funkcionality a mělo by se tedy jednat o „mozek“ aplikace. Zároveň jsou díky němu navzájem propojeny modely a views.

1.1.8.2 Python

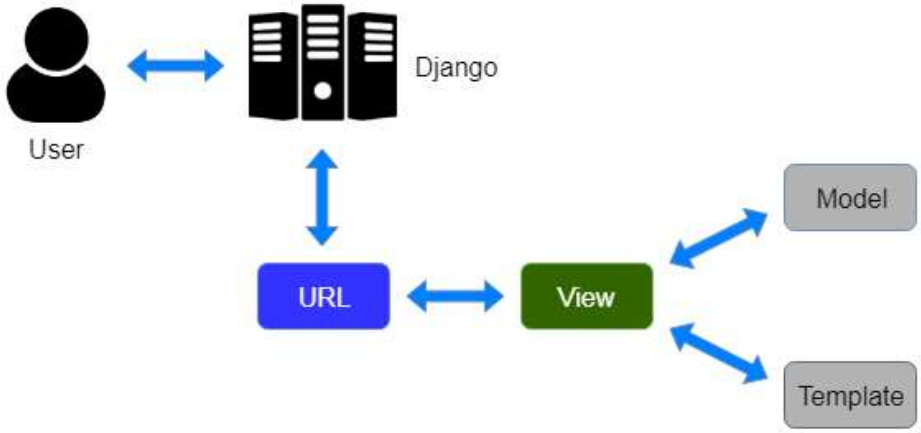
Python je univerzální programovací jazyk, což znamená, že vedle vývoje webových aplikací slouží také k psaní skriptů a desktopových aplikací, používá se také při vědeckých výpočtech a je taky vhodný pro strojové učení. To vše díky knihovnám, které lze doinstalovat. [14]

Na Pythonu jsou postavené různé frameworky pro vývoj webových aplikací. Vedle dále uvedeného Djanga se jedná o Flask, který oproti Djangu nevyžaduje žádné dodatečné nástroje ani knihovny, TurboGears nebo třeba web2py. [15]

2 DJANGO

Jak již bylo řečeno, Django je vysoko úrovňový framework používající Python, který podporuje rychlý vývoj a čistý design. Mottem je výstižný citát „Webový framework pro perfekcionisty (s deadliny)“.

Django využívá podobné architektury, jaká byla uvedena u ASP.NET, pouze s drobnými úpravami v názvech. Na rozdíl od MVC se zde používá MVT (Model-View-Template), kde Model zastává stejnou funkci, View obsahuje většinu logiky odehrávající se na pozadí a zároveň slouží jako spojení mezi Model a Template, který, jak napovídá název, slouží jako šablona a obshauje HTML kód s doplňky jazyka Jinja. Jako doplněk k MVT jsou dále formuláře, které lze definovat samostatně a v Template je pouze vyvolat pomocí Jinja. [16]



Obrázek 2 Architektura MVT [17]

Je zde snaha o dodržování principů DRY (Don’t Repeat Yourself), čemuž významně pomáhá používání šablon a Jinjy. Díky nim je možné například definovat hlavní dokument, jeho sekce a do nich dosazovat data dle výběru. [16]

2.1 Zabezpečení

Django má zabudovanou spoustu bezpečnostních opatření pro zabezpečení webových stránek proti nejznámějším kybernetickým útokům.

2.1.1 Cross site request forgery (CSRF)

Útoky CSRF umožňují útočníkům provádět akce za pomocí uživatelského ověření cizího uživatele bez jeho vědomí či souhlasu. Zabudované zabezpečení proti tomuto typu útoku je tak u Djanga samozřejmostí, pokud je nastaveno a správně použito. [18]

Podobně jako u dalších technik jsou i zde jistá omezení. Například je možné vypnout CSRF modul globálně nebo u vybraných Views a mělo by se tak činit v krajních případech, kdy si je programátor jistý svojí prací. Další limitace může být využití subdomény, která je mimo kontrolu programátora. [18]

CSRF ochrana funguje pomocí cookies obsahující náhodnou hodnotu, která je pro každého uživatele originální a z důvodu bezpečnosti je uživateli změněna při každém novém přihlášení. Tato hodnota je navíc připojena za náhodnou masku, čímž se zabraňuje úniku dat při útoku. [18]

2.1.2 Cross site scripting (XSS)

XSS útoky fungují tak, že útočník vloží skript běžící na straně klienta a ten se následně nahraje do prohlížečů ostatních uživatelů. Většinou je toho dosaženo uložením škodlivého skriptu v databázi a následném nahrání a zobrazení tohoto skriptu ostatním uživatelům nebo donucením uživatele do otevření odkazu, který následovně zaručí spuštění útočníkova JavaScriptového programu v uživatelově prohlížeči. Nicméně útok může být inicializován z jakéhokoliv nedůvěryhodného zdroje dat, jako třeba cookies či webové služby, a to kdykoliv data nejsou dostatečně „očištěna“ před zobrazením uživateli. [18]

Používání Templates uživatele ochraňuje před většinou těchto útoků tak, že překládají určité znaky, které by mohly být nebezpečné v HTML kódu. Avšak i zde jsou omezení. Například při používání dynamických tříd může dojít k vložení škodlivého kódu. Proto by se HTML kód měl do databáze ukládat co nejméně, v nejlepším případě vůbec. [18]

2.1.3 SQL injekce

Jedná se o typ útoku, kdy se útočník pokouší o spuštění SQL kódu, čímž může dojít k úniku dat z databáze či jejich porušení. [18]

Sady dotazů Djanga jsou chráněny proti tomuto typu útoku, jelikož jsou dotazy vytvářeny pomocí parametrizace. SQL kód je definován samostatně a odděleně od parametrů, které mohou být poskytnuty uživatelem a tím pádem nebezpečné, jsou ošetřeny databázovým ovladačem. [18]

2.1.4 Clickjacking

Clickjacking je typ útoku, během něhož stránka může zobrazovat prvky jiné internetové stránky a donutit tak nic netušícího uživatele provést nechtěné akce na druhé stránce. [18]

Ošetření funguje skrze X-Frame-Options middleware, který v podporujícím prohlížeči zabraňuje vyrenderování stránky uvnitř rámce. Je možné tuto ochranu vypnout pro jednotlivé Views. [18]

3 NÁHODNÁ ČÍSLA

Následující kapitola se zabývá konceptem náhodných čísel, jejich rozdělením na pseudo-náhodná a opravdu náhodná čísla, způsoby jejich generování a jejich využití v reálném životě.

3.1 Úvod

Jako náhodná čísla bývají označována taková čísla z matematické množiny, která mají stejnou pravděpodobnost na to být vybrána. Poté je potřeba zajistit, aby program vybíral čísla na základě pravděpodobnosti, tedy náhodně a ne pomocí algoritmu, podle kterého by bylo možné dohledat zvolená čísla.

S náhodnými čísly je potřeba pracovat v mnoha případech, ať už se jedná o kryptografii, hry nebo třeba statistiku. V některých případech není zapotřebí mít náhodné číslo, v jiných je to ale velmi žádané kvůli zabezpečení vůči různým útokům.

3.2 Pseudo-náhodná čísla

Jak napovídá název, nejsou pseudo-náhodná čísla náhodná v pravém slova smyslu. Nejde o náhodnost, jako třeba u hodu kostek nebo mincí. Ve své podstatě jsou generátory pseudo-náhodných čísel algoritmy, které využívají matematické funkce nebo předdefinované tabulky naplněné posloupností čísel, které následně vypadají náhodně. Proběhlo velké množství výzkumu na toto téma a moderní algoritmy pro tento účel jsou tak dobře navrženy, že se výsledná čísla opravdu jeví jako náhodná. [19]

Základní rozdíl mezi pseudo-náhodnými generátory a náhodnými generátory je nejlépe vidět na uvedeném příkladu výběru předchystaných čísel na jedné straně a hodu kostkami na straně druhé. Dalo by se říct, že využití pseudo-náhodného generátoru je jako házet kostkami, zapisovat hozené hodnoty a tyto hodnoty poté pouze přebírat jednu po druhé. Ano, čísla se jeví náhodné, ale ve skutečnosti jsou již předurčené. Náhodný generátor by tedy mohl být přirovnán k házení kostkou a čtení padlého hodu. V praxi se používají fyzikální veličiny, které jsou pro zařízení lehčeji zpracovatelné a které nelze dopředu určit. [19]

Ve prospěch pseudo-náhodných generátorů hraje rychlost produkce náhodných čísel. Naopak jsou deterministické, čímž je myšleno, že při shodných počátečních podmínkách vyjdou pro různé generování stejné výsledky, což může být přínosné při pozdější nutnosti

opakování náhodné sekvence. Typicky tyto generátory bývají i periodické, jinak řečeno daná sekvence čísel se časem bude opakovat. Ačkoliv je tato vlastnost povětšinou nežádoucí, moderní generátory mají nastavené periody tak dlouhé, že ve většině případů se o periodicitu programátor nemusí starat. [19]

Tyto charakteristiky dělají z pseudo-náhodných generátorů vhodné zařízení pro aplikace, kde je potřeba velké množství dat a kde je prospěšné snadné opakování řady náhodných čísel. Jako příklady bych uvedl simulační a modelovací aplikace. Naopak se tyto generátory nehodí pro software s nepředvídatelnými náhodnými hodnotami, například v kryptologii nebo hazardním průmyslu. [19]

3.3 Náhodná čísla

Na rozdíl od předchozího uvedeného typu generátorů se vyznačují náhodné generátory extrakcí náhodnosti z fyzické veličiny a následné interpretace těchto dat počítači. Lehký příklad by byl pohyb myši uživatele nebo perioda mezi stisky kláves na klávesnici. U takovýchto dat je ale potřeba dbát zřetel na způsob zpracování dat a myslet na různé okolnosti. Třeba právě použití klávesnice může být v operačním systému nejdříve ukládáno do bufferu. Jinak řečeno stisky klávesnice bývají nejdříve shromažďovány a posílány najednou do programu, čímž se údaje o periodách vytrácí a zaniká tak zamýšlená náhodnost. [19]

Nicméně existují i jiné způsoby zjišťování náhodných dat. Využití najde v makroskopickém měřítku radioaktivní rozpad, termální, elektrický či atmosférický šum, v mikroskopickém měřítku se pracuje s kvantovými jevy jako třeba fotony a jejich vlastnosti. Pokud ovšem člověk věnuje pozornost všem vlivům na sledovanou veličinu, možnosti jsou neomezené. Za zmínku stojí generátor Lavarand, který pro generování náhodných čísel používal snímky lávové lampy. [19][20]

Náhodné generátory se většinou skládají ze dvou částí – zdroj entropie jako fyzický zdroj náhodných dat, jehož výstupem je digitalizovaný výsledek měření veličiny, a softwarový program pro post-processing pro zpracování dat do takové podoby, aby byly vhodné pro aplikaci využívající tento generátor. Post-processing může být reprezentován jako matematická funkce implementována v softwaru zajišťující korekci nedokonalostí výstupu ze zdroje entropie, například korelace. Náhodné generátory se slabým zdrojem entropie jsou o to více závislé na silném post-processingu, který vedle korekcí může zakrýt i chybu - zra-

nitelnost. Tudíž je potřeba, aby analýza entropie měla přímý přístup k surovým datům namísto těch upravených. [20]

3.3.1 ID Quantique kvantový generátor náhodných čísel

Na základě čipu od ID Quantique vychází najevo fakt, že množství generovaných fotonů ze zdroje světla kolísá kolem určitého průměru. Toto kolísání je čistě kvantového původu, tím pádem jsou zásadně náhodné na základě fyzikálních zákonů. V generátorech je vysílán v krátkých intervalech paprsek světla, kterým je osvícen panel s pixely senzitivními na jednotlivé fotony, přičemž každý zachytí určité množství fotonů. Jednotlivé výstupy jsou poté digitalizovány pomocí AD převodníku, na základě jehož výstupu je poté možné určit množství dopadajících fotonů na každý pixel a jejich fluktuaci. Přechod je tak přímočarý a nepodléhá žádným dalším vlivům, které by mohly zvýšit předvídatelnost výsledků. [20]

3.3.2 Online generátor Random.org

Tyto stránky poskytují náhodná čísla na základě atmosférického šumu. Na základě náhodných čísel potom generují širokou škálu různých typů dat, od celých čísel, přes datum až po hesla. Web vznikl na konci minulého století jako zakázka pro hazardní průmysl. Od té doby se rozrostl do dnešní podoby a poskytuje tak volně náhodná data.[19]

II. PRAKTICKÁ ČÁST

4 NÁVRH

Předem je potřeba uvědomit si, jak by aplikace měla vypadat, co by měla splňovat a co navíc by se dalo připojit. Dále je potřeba vidět aplikaci z pohledu uživatele a snažit se o co nejlepší uživatelské prostředí.

4.1 Registrace

Jelikož se bude jednat o výukový program, je nejdříve zapotřebí rozlišit jednotlivé uživatele, dát jim osobní data, možnost přístupu do výukových skupin a zároveň umožnit přidělení rolí a práv. Je přeci žádané, aby vyučující měl více možností zasahování do systému, než jednotliví studenti. V aplikaci je tedy nutné vytvořit registrační systém, který zařídí vytvoření objektu žáka a také jeho uložení do databáze. Po registraci by měl být žák automaticky přihlášený a měl by mít přiděleny osobní náhodná data.

Dále je potřeba vytvořit samostatný systém pro přidávání uživatelů s rolí kantora. Ten by měl fungovat na principu výběru z existujících uživatelů. K tomuto kroku by měl mít právo pouze další vyučující.

4.2 Přihlášení

Vedle registračního systému musí existovat i ten přihlašovací. Po uživateli bude vyžadováno zadání emailové adresy a hesla. Po přihlášení by měl být uživatel přesměrován na hlavní stranu, kde mu budou přístupna vlastní náhodná data, případně mu bude umožněno přihlášení do skupiny nebo odhlášení.

4.3 Role

Uživatelům je tedy přiřazována role, a to buď student, anebo vyučující. Na základě těchto práv by následovně měly být přístupné různé sekce, jako například přidání/smazání skupiny, smazání neaktivních uživatelů, generování množiny čísel pro skupinu a další.

Vedle dvou zmíněných rolí existuje role administrátora, kterou by měla mít přístupnou jen oprávněná osoba. Jako administrátor bude mít přístup na speciální stránku, která je defaultně generována systémem Django. Na ní lze najít kompletní výpis databáze rozdělený na jednotlivé modely, který lze rozšiřovat nebo mazat dle vůle administrátora. Je zde i možnost přidávání nových objektů k jednotlivým modelům a případně jejich editace či smazání.

4.4 Skupiny

Výběr skupiny by měl mít každý uživatel přístupný. Pro přihlášení do skupiny je zapotřebí vyplnění klíče, který může nastavit pouze uživatel s rolí vyučujícího. Každé skupině by poté měla náležet množina náhodných čísel, která by se měla zobrazovat všem zapsaným uživatelům. V detailu skupiny bude popis skupiny a seznam členů.

Pro vytvoření skupiny bude pro roli vyučujícího přístupné tlačítko, pomocí kterého se dostane do formuláře s poli pro zadání přístupového hesla a popisu skupiny.

Vyučující bude mít navíc možnost jednotlivé studenty ze skupiny vyřadit a průběžně měnit jak klíč, tak i popis skupiny.

4.5 Data z generátoru náhodných čísel

Samotná data budou uložena v databázi. Bude existovat samotný program, který bude databázi průběžně doplňovat. Při vytvoření skupiny nebo uživatele se daným objektům přiřadí jejich množina čísel, které by se pro plynulost programu měly čerpat z databáze naplněné dostatečným množstvím vygenerovaných dat a následně odtud odebrány pro jedinečnost dat pro každý objekt.

5 IMPLEMENTACE

Bylo využito frameworku Django a kombinace databází MySQL pro ukládání dat z generátoru a SQLite pro práci s modely. SQLite byla nastavena jako výchozí databáze pro Dajngo a není použito žádných SQL queries pro manipulaci s touto databází. K práci s ní slouží jednotlivé modely, které budou dále uvedeny a rozepsány. Vedle Djanga je využito HTML, CSS, jQuery a Jinja pro tvorbu front-end části aplikace. Komunikace s generátorem náhodných dat je zprostředkována pomocí rozhraní USB a softwarově je využito příkazů operačního systému.

5.1 Nastavení

Django má vlastní vnitřní strukturu již po založení projektu. Na obrázku jsou ukázané vygenerované složky a soubory. Adresář projektu obsahuje další složku se stejným názvem. V ní se nachází soubory nutné pro chod aplikace. V settings.py se nachází nastavení souboru, informace o databázi, používané aplikace, časové pásmo, jazyk, informace o složkách obsahující statické soubory a media, informace o přihlašovací stránce a další. Podle urls.py Django vyhledává zadanou adresu, která byla předána views z prohlížeče.

Na stejné úrovni jako zmíněnou složku najdeme defaultní soubor databáze SQLite, který slouží k iniciaci tabulek v databázi. Vedle něj je zde i manage.py, pomocí kterého se do projektu přidávají aplikace, spouští se server, dělají migrace a podobně. Jedná se tedy o ovládací prvek projektu, pomocí kterého jej řídíme.

Dále se na této úrovni budou nacházet aplikace, které budou pro tento projekt vytvořeny. Díky tomuto vnitřnímu uspořádání se dají lépe ovládat velké projekty, jelikož vytvářené aplikace by měly být na sobě nezávislé a zastupovat vlastní funkcionality.

V této práci budou vytvořeny tři aplikace:

- Pro přihlášení a registraci do systému
- Pro práci s daty z generátoru
- Pro administraci aplikace

Současně jsou zde uloženy Templates. Ty se v případě rozsáhlejších projektů mohou ukládat přímo do adresářů jednotlivých aplikací nebo, stejně jako v tomto případě, jsou uloženy v samostatné složce v kořenovém adresáři. Poté je složka Templates rozdělena na jednotlivé podsekce dle aplikací pro lepší přehled pro programátora.

5.2 Nastavení databáze

5.2.1 MySQL

Tato databáze obsahuje pouze jednu tabulku. Jedná se o záznamy s náhodnými čísly, které jsou doplňovány z generátoru. Tabulka je naplněna obsáhlým množstvím dat, aby se zabránilo rychlému vyprázdnění a nedostatku při přiřazování čísel jednotlivým objektům. Data jsou doplňována periodicky programem, který každých 15 minut kontroluje množství dat v databázi a v případě poklesu spustí generátor a tabulku doplní.

5.2.2 SQLite

V této databázi se nachází zbytek dat. Pomocí Django models je zprostředkována komunikace pomocí Python kódu a tím pádem odpadá nutnost sepisování SQL dotazů. Models se definují v souboru models.py, který obsahuje každá aplikace.

Aby se mohly propsat změny v modelech do databáze, je nutné používat příslušný příkaz pomocí terminálu a souboru manage.py, o kterém bylo zmíněno v podkapitole 5.1 Nastavení. Pro zobrazení a editaci nových modelů je navíc potřeba je uvést v souboru admin.py, který se nachází ve stejném adresáři jako models.py.

5.3 Komunikace s generátorem

Pro tuto komunikaci je vytvořen samostatný skript, který je spuštěn nezávisle na serveru. Program je velice jednoduchý. Jedná se o zjištění počtu dat v databázi MySQL a v případě nedostatečného naplnění je spuštěno kvantové generování, ze kterého jsou převzatá data uložena ve formě celých čísel, která jsou dále zpracovávána webovou aplikací. Dotazování na databázi probíhá každých patnáct minut.

Podle dokumentace dodané spolu s generátorem by mělo být možné využívat připojené knihovny psané v jazycích Java, C++ nebo C#. Bohužel jsou postupy pro zprovoznění knihoven zastaralé a spousta odkazů na podpůrný software již nebyla aktivní a nikde nebylo uvedeno, čím by se tyto zdroje daly nahradit. Z tohoto důvodu je komunikace zprostředkována pomocí batch souboru, který z připojeného generátoru přesměrovává výstup do souboru na disku. Hlavní program se pak postará o převzetí těchto dat a jejich nahrání do databáze.

5.4 Systém autentizace

Django disponuje vlastním autentizačním systémem, který byl zprvu používán, ale pro lepší ovladatelnost a chuť vlastního návrhu byl tento systém potlačen a nahrazen novým. Implementace je v samostatné aplikaci a jsou definovány modely pro uživatele a skupiny, které se používají napříč celou webovou aplikací.

Registrační panel je definován pomocí formuláře v souboru forms.py. Obsahuje pole pro e-mailovou adresu, jméno, příjmení a heslo, které je nutné zopakovat pro zaručení správnosti. Inputy jsou kontrolovány jak na straně klienta pomocí jQuery, tak na straně serveru. V případě erroru dojde k obnovení registračního formuláře a uživatel je upozorněn pomocí výpisu na chyby, které je potřeba odstranit. Jelikož e-mailová adresa slouží též jako uživatelské jméno, není možné, aby se na jednu adresu registroval uživatel vícekrát. V heslu je vyžadováno uvedení alespoň jednoho malého písmene, jednoho velkého písmene a alespoň jednoho čísla. Po potvrzení registrace je uživatel přihlášen v roli studenta, jsou mu přiděleny osobní náhodná čísla a je přesměrován na hlavní stranu, kde má přístupné osobní náhodná data. Pomocí lišty se následně může zapsat do libovolné skupiny.

Přihlašovací formulář je definován ve stejném souboru jako ten registrační. Obsahuje ale pouze pole pro e-mailovou adresu a heslo. Po odeslání dotazu na serveru dojde ke kontrole správnosti přihlášení a v kladném případě je uživatel přihlášen a přesměrován na hlavní stranu.

Model studenta je složen z jeho osobních náhodných čísel, skupin, do kterých se zapsal, a rolí, které mu náleží. Obojí je realizováno pomocí relace n:m. Vztahem 1:1 je propojen se zabudovaným modelem User. Ten obsahuje informace o uživatelském jméně, heslu, e-mailové adrese a další užitečné informace.

Model skupiny se skládá z náhodných čísel pro skupinu, zapisovacího klíče, popisu skupiny a vztahem 1:1 je svázán se zabudovaným modelem Group.

5.5 Hlavní strana

Hlavní strana je složena z následujících prvků:

- Navigace – zde se nachází odkazy na hlavní stranu, přehled skupin a odhlášení v případě role studenta. S rolí vyučujícího je možné dostat se navíc do výpisu uživatelů.
- Levé menu – slouží k přepínání typu náhodných dat. Je rozděleno na dvě části: první slouží k přepínání mezi typy dat pro skupinu, ve které je uživatel zapsán; druhá část je pro data osobní. Tato část byla přidána z toho důvodu, aby uživatel mohl hned po registraci zkoušet přistupovat k náhodným datům a nebyl tak vázán na skupinu. Uživatel v roli vyučujícího zde navíc má zpřístupněno tlačítko pro vygenerování nových náhodných čísel pro skupinu.
- Pravé menu – je zobrazováno pouze v případě výběru dat pro skupinu, a pokud je uživatel zapsán do více skupin zároveň. Slouží k výběru skupiny, pro kterou chce uživatel data zobrazit. Pro roli vyučujícího je zde navíc tlačítko pro stažení nových čísel pro skupinu.
- Hlavní panel – obsahuje výpis náhodných dat ve formátu, jaký si uživatel zvolí v levém menu

Hlavní strana tak slouží primárně k přístupu k datům z databáze.

5.6 Seznam skupin

Tato sekce je přístupná všem uživatelům bez ohledu na jejich role. Co se ale liší, jsou možnosti, jaké lze provádět s danými skupinami. Student má možnost zobrazit si všechny skupiny, zapsat se do skupin a poté se jim zobrazují veřejná data skupiny, jako je popis a výpis ostatních zapsaných uživatelů. Kantor smí vytvářet nové skupiny, ukládat jejich data, mazat ty nadále nechtěné a vedle zobrazení uživatelů může jednotlivé uživatele ze skupiny odstranit.

5.7 Seznam uživatelů

Seznam uživatelů je přístupný pouze pro vyučující. Mohou si zobrazit jejich informace, jako jsou jméno a email. Hesla zde nejsou přístupná z důvodu bezpečnosti. Dále je možné nadále již neaktivní uživatele odstranit.

5.8 Administrační panel

Jedná se o zabudovanou aplikaci Djanga. Přístup zde by měl mít povolen pouze uživatel v roli administrátora z důvodu zabezpečení aplikace před ztrátou dat. Byla ponechána z důvodu budoucího využití. Jedná se opravdu o silný nástroj, pomocí něhož lze upravovat modely, vytvářet nové nebo odstraňovat nežádoucí. Informace typu hesel nejsou viditelná ani zde – zobrazuje se pouze hash. S tím samozřejmě souvisí i vkládání záznamů nebo jejich úprava.

5.9 Vzhled

K úpravě designu stránek byla převážně používána knihovna Bootstrap, díky které vypadají stránky poněkud moderněji. K tomu napomáhá využívání ikon, které Bootstrap zdarma nabízí. Ne všechno šlo takto navrhnout. Proto se v kořenovém adresáři nachází složka static obsahující CSS soubor, do kterého byly dopisovány vlastní rozšiřující styly.

5.10 Testování

Testování většinou probíhalo lokálně a jednalo se primárně o postupné odstraňování chyb, které vyvstávaly buď svévolně (z důvodu nedostatečné vzdělanosti autora), nebo z důvodu pochybení. Dále bylo potřeba otestovat očekávanou funkčnost a směřovat k vytyčenému cíli. V neposlední řadě se testovaly i nechtěné situace, jako například správné omezení přístupu k jednotlivým modulům pomocí rolí.

ZÁVĚR

Tato bakalářská práce měla za cíl vytvořit webovou aplikaci, pomocí které by se obsluhoval kvantový generátor od společnosti ID Quantique, který byl zapůjčen ústavem fakulty pro vývojové účely. Záměrem této aplikace je sloužit v následujících semestrech jako pomůcka při výuce. Z důvodu očekávaného aktivního používání aplikace bylo při vývoji dbáno na uživatelsky přívětivé rozhraní.

S přihlédnutím k faktu, že se během procesu pracuje s poměrně velkým množstvím dat, byl proto zvolen programovací jazyk Python, který je jako stvořený pro tento úkol. Pomocí frameworku Django, který je postavený právě na Pythonu, byla tato aplikace naprogramována a otestována. Výsledkem jsou webové stránky, které uživateli umožňují stáhnout si vygenerovaná data a dále s nimi pracovat. Uživatel má možnost data reprezentovat v různých datových typech, od celých čísel až po data.

Hlavní strana je rozdělena na dvě sekce – první je určena pro skupinu a druhá pro osobní účely. Také je využíváno rolí uživatele, pomocí kterých se mu mohou povolovat privilegované funkce, do kterých obecně spadá správa veřejných dat.

Jako prostor pro vylepšení a další rozvoj bych uvedl zasílání emailů, a to jak informačních, tak potvrzovacích, kdy by například při zápisu do skupiny mohl vyučující explicitně potvrdit či zamítnout žádost o připojení se ke skupině. Dále by bylo možné přidat další role a uživatelům tak striktněji vytyčit jejich práva. To by ovšem dávalo smysl až při rozsáhlejší aplikaci. Mohlo by se využívat dvoufázového ověření uživatele, pokud by si takovou možnost zvolil.

SEZNAM POUŽITÉ LITERATURY

[1] PÍSEK, Slavoj. *HTML: začínáme programovat*. 3., aktualiz. vyd. [i.e.] 1. vyd. Praha: Grada, 2010. Průvodce (Grada). ISBN 978-80-247-3117-9.

[2] HOGAN, Brian P. *HTML5 a CSS3: výukový kurz webového vývojáře*. Brno: Computer Press, 2011. ISBN 978-80-251-3576-1.

[3] CASTRO, Elizabeth. *HTML, XHTML a CSS: názorný průvodce tvorbou WWW stránek*. Brno: Computer Press, 2007. ISBN 978-80-251-1531-2.

[4] Introduction. *Bootstrap* [online]. 2011 [cit. 2021-5-10]. Dostupné z: https://getbootstrap.com/docs/5.0/getting-started/introduction/

[5] ŠKULTÉTY, Rastislav. *JavaSript: programujeme internetové aplikace*. 2. aktualiz. vyd. Brno: Computer Press, 2004. ISBN 80-251-0144-4.

[6] *JQuery - kuchařka programátora*. Brno: Computer Press, 2010. ISBN 978-80-251-3152-7.

[7] JQuery: write less, do more. *JQuery* [online]. c 2021 [cit. 2021-5-10]. Dostupné z: https://jquery.com/

[8] What is MySQL? *MySQL* [online]. c 2021 [cit. 2021-5-10]. Dostupné z: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

[9] About SQLite. *SQLite* [online]. [cit. 2021-5-10]. Dostupné z: https://sqlite.org/about.html

[10] A tour of the C# language. *Microsoft* [online]. 2021 [cit. 2021-5-10]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/

[11] ASP.NET. *Microsoft* [online]. c 2021 [cit. 2021-5-10]. Dostupné z: https://dotnet.microsoft.com/apps/aspnet

[12] What is ASP.NET? *Microsoft* [online]. c 2021 [cit. 2021-5-10]. Dostupné z: https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet

[13] What is MVC Design Pattern? *Educba* [online]. [cit. 2021-5-10]. Dostupné z: https://www.educba.com/what-is-mvc-design-pattern/

[14] What Is Python Used For? *Skillcrush* [online]. [cit. 2021-5-10]. Dostupné z: https://skillcrush.com/blog/what-is-python-used-for/

[15] The Python Wiki. *Python* [online]. [cit. 2021-5-10]. Dostupné z: https://wiki.python.org/moin/WebFrameworks

[16] Django at a glance. Django [online]. [cit. 2021-5-10]. Dostupné z: https://docs.djangoproject.com/en/3.2/intro/overview/

[17] Django MVT. *JavaTPoint* [online]. [cit. 2021-5-10]. Dostupné z: https://www.javatpoint.com/django-mvt

[18] Security in Django. *Django* [online]. [cit. 2021-5-10]. Dostupné z: https://docs.djangoproject.com/en/3.2/topics/security/

[19] Introduction to Randomness and Random Numbers. *Random.org* [online]. [cit. 2021-5-10]. Dostupné z: https://www.random.org/randomness/

[20] Quantis Random Number Generator: True random number generator exploiting the randomness of quantum physics. *IDQ* [online]. [cit. 2021-5-10]. Dostupné z: https://www.idquantique.com/random-number-generation/products/quantis-random-number-generator/

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
XML	Extensible Markup Language
AJAX	Asynchronní JS a XML
MVC	Model View Controller architektura
MVT	Model View Template architektura
CSRF	Cross Site Request Forgery
XSS	Cross Site Scripting
AD	Analogově digitální

SEZNAM OBRÁZKŮ