

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta elektrotechniky a informatiky

Evidenčné číslo: FEI-16605-120403

**Program pre vyhodnotenie zásahov pri
športovej streľbe**

Bakalárska práca

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta elektrotechniky a informatiky

Evidenčné číslo: FEI-16605-120403

Program pre vyhodnotenie zásahov pri športovej strelbe

Bakalárska práca

Študijný program:	Aplikovaná informatika
Študijný odbor:	Informatika
Školiace pracovisko:	Ústav elektroniky a fotoniky
Školiteľ:	doc. Ing. Miroslav Hagara, PhD.

2024

Nicolas Droppa



ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Meno Priezvisko**
ID študenta: 012345
Študijný program: jadrové a fyzikálne inžinierstvo
Študijný odbor: elektrotechnika
Vedúci práce: tituly, Meno Priezvisko, tituly
Vedúci pracoviska: tituly Meno Priezvisko, tituly
Miesto vypracovania: Ústav jadrového a fyzikálneho inžinierstva

Názov práce: **Rozšírená šablóna záverečnej práce na FEI STU v Bratislave
v systéme LaTeX**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

- Vytvorte šablónu záverečnej práce pre študentov bakalárskeho a inžinierskeho štúdia na FEI STU v Bratislave.
- Napište návod na písanie záverečnej práce s použitím vytvorenej šablóny.
- Vytvorte prehľad najpoužívanejších typov citovaných zdrojov a zdokumentujte ich tak, aby boli v súlade s normou ISO 690.

Zoznam odbornej literatúry:

- Zákon č. 131/2002 Z. z. z 21. februára 2002 o vysokých školách a o zmene a doplnení niektorých zákonov [online]. [cit. 2024-09-01]. Dostupné na https://www.slov-lex.sk/static/pdf/2002/131/ZZ_2002_131_20240901.pdf
- Vyhláška č. 233/2011 Z. z. Ministerstva školstva, vedy, výskumu a športu Slovenskej republiky z 1. júla 2011, ktorou sa vykonávajú niektoré ustanovenia zákona č. 131/2002 Z. z. o vysokých školách a o zmene a doplnení niektorých zákonov v znení neskorších predpisov [online]. [cit. 2024-09-01]. Dostupné na https://www.slov-lex.sk/static/pdf/2011/233/ZZ_2011_233_20201015.pdf

3. Metodické usmernenie Ministerstva školstva, vedy, výskumu a športu SR č. 56/2011 o náležitostiach záverečných prác, ich bibliografickej registrácii, kontrole originality, uchovávaní a sprístupňovaní č. 2011-11513/31015:4-071 [online]. [cit. 2024-09-24]. Dostupné na <http://www.minedu.sk/metodicke-usmernenie-c-562011-o-nalezitostiach-zaverecných-prac-ich-bibliografickej-registrácii-uchovavani-a-sprístupnovani/>.
4. STN ISO 214: 1998. Dokumentácia. Abstrakty (referáty) pre publikácie a dokumentáciu. Bratislava: Slovenský ústav technickej normalizácie. Bratislava. Úrad pre metrológiu a skúšobníctvo Slovenskej republiky.
5. STN ISO 690: 2022. Dokumentácia – Bibliografické odkazy – Obsah, forma a štruktúra. Bratislava. Úrad pre metrológiu a skúšobníctvo Slovenskej republiky.
6. STN 01 6910: 2023. Pravidlá písania a úpravy písomností. Bratislava. Úrad pre metrológiu a skúšobníctvo Slovenskej republiky.
7. STN ISO 2145: 1997. Dokumentácia. Číslovanie oddielov a pododdielov písaných dokumentov. Bratislava. Úrad pre metrológiu a skúšobníctvo Slovenskej republiky.
8. STN ISO EN 80 000-1: 2022. Veličiny a jednotky. Bratislava. Úrad pre metrológiu a skúšobníctvo Slovenskej republiky.
9. LICHNEROVÁ, L. a B. BELLÉROVÁ. Nové pravidlá citovania podľa ISO 690 z roku 2021. *ITlib. Informačné technológie a knižnice*, 3–4/2023, 10 – 22. [cit. 2024-09-06]. Dostupné na <http://doi.org/10.52036/1335793X.2023.3-4.10-22>.
10. HOFTICH, M. ISO 690 biblatex style [online]. [cit. 2024-09-30]. Dostupné na <https://mirrors.ibiblio.org/CTAN/macros/latex/contrib/biblatex-contrib/biblatex-iso690/biblatex-iso690.pdf>.

Termín odovzdania bakalárskej práce: 31. 05. 2025

Dátum schválenia zadania bakalárskej práce: 28. 02. 2025

Zadanie bakalárskej práce schválil: tituly Meno Priezvisko, tituly – garant študijného programu

Podakovanie

Touto cestou by som sa chcel poďakovať vedúcemu mojej bakalárskej práce doc. Ing. Miroslavovi Hagarovi, PhD. za výpomoc, cenné rady a konštruktívnu kritiku, ktoré mi pomohly pri vypracovaní tejto práce a posunutí sa v odbornej sfére.

Abstrakt

Cieľom bakalárskej práce je... Práca pozostáva z X kapitol. Prvá kapitola sa zameriava na... Druhá kapitola sa venuje... Posledná kapitola obsahuje...

Kľúčové slová

Detekcia, Perspektíva, Strelba

Abstract

The manual for students of the Faculty of Electrical Engineering and Information Technology at Slovak University of Technology in Bratislava provides advice and guidelines on how to approach the formal aspects of writing a final thesis for university studies. The document can also serve as a template for the thesis in the L^AT_EX typesetting system. It covers in detail the rules for typesetting mathematical equations, numbering floating objects, and referencing them. It also thoroughly addresses the method of citing external literary sources.

Keywords

Detection, Perspective, Shooting

Obsah

Úvod	11
1 Športová strelba na terč	12
1.1 Moderná strelba na terč	12
2 Programovací jazyk JavaScript	13
2.1 Výhody jazyka JavaScript	13
2.2 Nevýhody jazyka JavaScript	14
2.3 JavaScript na serveri	14
3 PWA	15
3.1 Silné stránky PWA	15
3.2 Neviditeľná vrstva	16
4 Programovací jazyk PHP	17
5 Databáza	18
6 OpenCV	19
6.1 Čítanie obrázkov	19
6.2 Vlastnosti obrázka	19
6.3 Vyobrazenie geometrických útvarov	19
Záver	21
Literatúra	22
Použitie nástrojov umelej inteligencie	22
A Algoritmus	24
B Výpis dlhého kódu	25
C Slovníček pojmov	26

Zoznam značiek a skratiek

Zoznam algoritmov

Zoznam výpisov kódov

Úvod

Športová strelba na terč je disciplína, ktorá vyžaduje vysokú presnosť, sústredenie a stabilnú techniku. S rozvojom moderných technológií sa nie len zvyšujú potreby zefektívniť proces vyhodnocovania samotných striel na terč, ale aj spätnej väzby pre strelcov aby mohli na základe štatistiky posunúť svoje výkony ešte ďalej.

Táto bakalárska práca sa zaoberá vývojom programu na vyhodnocovanie zásahov v športovej strelbe, ktorý následne analyzuje vyznačené zásahy. Cieľom je vytvoriť systém, ktorý by umožnil používateľom pomocou jednoduchého a prehľadného používateľského rozhrania vyznačovať jednotlivé zásahy a následne efektívne poskytovať dáta pre zlepšovanie výkonu športovcov. Program by mal umožniť trénerom a športovcom lepšie porozumieť ich výkonu a identifikovať slabé miesta.

Na základe týchto cieľov bola v práci použitá metóda spracovania obrazu. ...

1 Športová strelba na terč

Strelba na terč má bohatú históriu, ktorá sa vyvinula z životne dôležitej schopnosti prežitia až po moderný súťažný šport. Tento vývoj bol ovplyvnený technologickým pokrokom, zmenami v spoločenských postojoch k strelným zbraňam a lukostrelbe a vývojom rôznych typov terčov, ako sú ocelové, papierové, skákacie a siluetové terče [1].

Strelba na terč bola zaradená do programu olympijských hier už v Paríži v roku 1896, kde sa prvýkrát objavila ako súťažný šport. V tomto období sa objavili rôzne formy streleckých súťaží, vrátane strelby na živé holuby. Po tomto ročníku bolo zavedené strieľanie na terče z umelej hmoty [2].

1.1 Moderná strelba na terč

Existujúce systémy na vyhodnocovanie zásahov v športovej strelbe kombinujú rôzne technológie na meranie a analýzu zásahov. Tieto riešenia zvyčajne zahŕňajú použitie digitálnych terčov, kamier, senzorov alebo iných optických zariadení, ktoré umožňujú automatické zistenie polohy zásahov na terči. Mnohé komerčné systémy používajú kamery na analýzu pohybu a zásahov v reálnom čase. Tieto systémy sú schopné poskytnúť okamžité výsledky, no sú často veľmi nákladné a vyžadujú zložité zariadenie, čo môže byť obmedzujúce pre amatérske alebo menšie strelecké kluby.

Niektoré z týchto riešení využívajú pokročilé metódy spracovania obrazu, ako je detekcia hrán alebo identifikácia štruktúr na terči, aby presne vyhodnotili zásahy. Napríklad, pri analýze obrazu sa často využíva technika, ktorá sleduje zmeny v pixeloch obrazu medzi jednotlivými snímkami, aby sa určilo miesto zásahu. V niektorých prípadoch sa využívajú algoritmy strojového učenia, ktoré sa učia rozpoznať vzory na terči a určiť bod zásahu s vysokou presnosťou.

Existujú aj systémy, ktoré sú zamerané na analýzu štatistík strelca počas viacerých kôl, čím umožňujú trénerom a športovcom lepšie pochopiť výkon a identifikovať oblasti na zlepšenie.

2 Programovací jazyk JavaScript

JavaScript je objektovo orientovaný, vysokoúrovňový programovací jazyk. Narozdiel od kompilovaných jazykov kde sa najprv kód preloží do strojového kódu a následne vytvorí spustiteľný súbor ako napríklad jazyk C, C++ alebo Rust, JavaScript je jazyk interpretovaný - teda kód sa vykonáva riadok po riadku pomocou interpreteru. O vznik tohoto programacieho jazyka sa zaslúžil Brendan Eich. JavaScript bol vydaný v roku 1995 v máji a to iba za 10 dní. Jeho prvým názvom bolo Mocha [3]. Tento programovací jazyk umožňuje implementovať komplexné funkcie na webových stránkach. Zakaždým, keď webová stránka robí viac, než len sedí a zobrazuje statické dáta - zobrazuje včasné aktualizácie obsahu, interaktívne mapy, animovanú 2D/3D grafiku, rolovacie video jukeboxy alebo iné interaktívne prvky, ide pravdepodobne o zakomponovanie JavaScriptu [4].

JavaScript patrí medzi najpoužívanějšíe programovacie jazyky najmä vďaka svojej jednoduchej a flexibilnej syntaxe, ktorá umožňuje rýchly štart a intuitívnu prácu s webovými technológiami.

Okrem svojej úlohy pri základnej interaktivite sa JavaScript v modernej webovej architektúre presunul z doplnkového jazyka fungujúceho popri html a css na samotne plnohodnotný základ pre vývoj komplexných aplikácií. Vďaka vývoju frameworkov ako React, Vue či Angular sa JavaScript stal nástrojom nielen pre manipuláciu s DOM - Document Object Model - štruktúrovaná reprezentácia HTML, ale aj pre budovanie komponentovo orientovaných systémov, spracovanie stavov aplikácií, routing či prácu s API na strane klienta. Tento posun umožnil načítavanie nových dát a informácií bez opätovného načítania celej stránky. Výsledkom je rýchlejšia, plynulejšia a používateľsky prívetivejšia skúsenosť, porovnateľná s natívnymi aplikáciami.

2.1 Výhody jazyka JavaScript

Medzi veľkú výhodu jazyka patrí "Client-Side Scripting", Teda JavaScript beží v používateľovom prehliadači, to znamená, že má kratší čas odpovedí, pretože nie je nutné komunikovať so serverom. Veľmi kladnou záležitosťou tohoto jazyka je určite jeho všestrannosť, JavaScript je možné použiť pre rôžnu škálu úloh, od jednoduchých výpočtov až po zložité aplikácie na strane servera. Dôvodom, prečo je jazyk tak populárny pri vývoji webových aplikácií je, že umožňuje reagovať na klikanie, stlačenie kláves, zmeny okna a rozhrania, atď. v reálnom čase. Javascript funguje asynchrónne - teda zvláda úlohy ako načítanie dát zo servera bez toho aby zamrzlo používateľské rozhranie. Vďaka obľúbenosti tohoto jazyka medzi programátormi po celom svete, tento jazyk disponuje množstvom dokumentácie a tutoriálov [5].

2.2 Nevýhody jazyka JavaScript

JavaScript je interpretovaný jazyk, to znamená, že je pomalší ako kompilované jazyky ako sú C, C++ alebo Pascal. Keďže JavaScript beží na strane klienta, je náchylný na útoky ako Cross-Site Scripting (XSS) a Cross-Site Request Forgery (CSRF). JavaScript je dynamicky typovaný jazyk, čo znamená, že typy premenných nie sú striktne definované. To môže viesť k chybám, ktoré sa objavia až pri spustení programu, a sťažuje to údržbu väčších kódových báz. JavaScript sa vo veľkej miere spolieha na asynchrónne programovanie, čo môže byť pre vývojárov náročné na pochopenie a správne implementovanie. Aj keď moderné konštrukcie ako Promises a async/await pomáhajú, stále existuje riziko vzniku tzv. "callback hell-[6].

2.3 JavaScript na serveri

Hoci bol JavaScript pôvodne vytvorený ako skriptovací jazyk pre webové prehliadače, vďaka Node.js sa rozšíril tento jazyk aj na stranu serveru - teda back-end. Node.js umožňuje vývojárom písať serverový kód v rovnakom jazyku, aký používajú aj pre klientskú časť, čo zjednodušuje vývoj tzv. fullstack aplikácií.

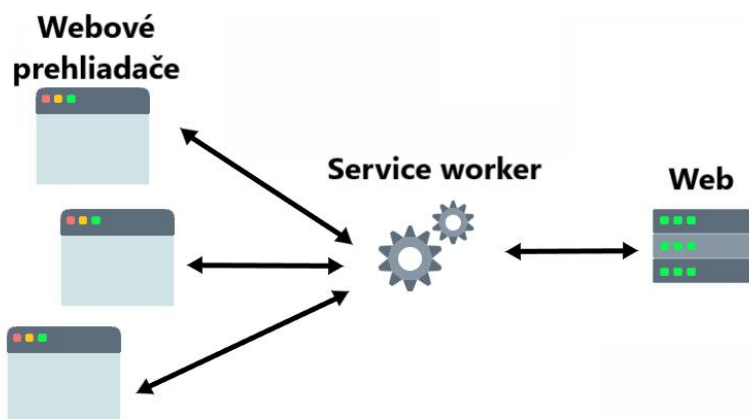
Medzi najbežnejšie využitia JavaScriptu na serveri patrí obsluha API požiadaviek, spracovanie dát, práca s databázami a správa autentifikácie. Jednoduchý a výkonný framework ako Express.js výrazne urýchľuje tvorbu webových serverov. Týmto spôsobom sa z JavaScriptu stal univerzálny jazyk schopný pokrývať celý vývojový cyklus webovej aplikácie [7].

3 PWA

PWA je skratkou pre progressive web app - teda progresívna webová aplikácia vytvorená pomocou štandardných webových technológií ako HTML, CSS a JavaScript, ktorá využíva moderné webové schopnosti na poskytnutie používateľského zážitku podobného natívnym aplikáciám. Tieto aplikácie môžu byť inštalované na zariadenia a fungovať offline, čo zvyšuje ich dostupnosť a použiteľnosť.

3.1 Silné stránky PWA

Dostupnosť bez ohľadu na internetové pripojenie - Progresívne webové aplikácie nie sú závislými od pripojenia používateľa k internetu tak ako to býva u zvyčajných webstránok. Pri používateľovej návšteve PWA bude zaregistrovaný service worker, ktorý deteguje a reaguje na zmeny používateľovho pripojenia. Práve pre túto prácu teda bude kľúčovou funkcionalitou to, že dokáže poskytnúť plnohodnotnú funkcionalitu aj pre takých používateľov ktorý nie sú v daný moment pripojení k internetu. Podrobnosti budú v podkapitole ??.



Obr. 1: Názov obrázka [8]

Krátke doby načítavania - Pri používaní service workerov je možné vytvárať stránky, ktoré budú zobrazené hneď po otvorení bez ohľadu na to ako silné a dobré pripojenie používateľ má. Stránky sa dokážu načítať v priebehu milisekúnd za použitia metódy offline first-[8].

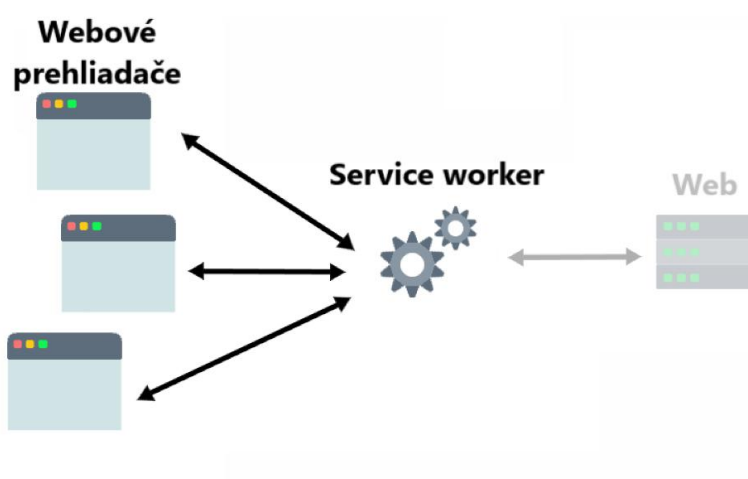
Notifikácie - Progresívne webové aplikácie môžu posilať používateľom takzvané "Push notifications" niekoľko dní po návšteve stránky. Tieto notifikácie sú skvelým nástrojom ako si môže stránka udržať viditeľnosť a svojich navštevovateľov. [8].

Natívny výzor - PWA môžu mať kompletne nerozoznatelný vzhľad od klasických desktopových aplikácií. Je možné aby sa zapínali bez prehliadačového user interface-u, mali animáciu alebo nejaké intro pri spúšťaní, dokonca sa môžu aj uzamknúť na predvolenú orientáciu, na šírku alebo výšku.

3.2 Neviditeľná vrstva

Tak ako sme už načrtli, základom každej progresívnej webovej aplikácie je service worker. Pred tým ako sa vôbec service workeri objavili, kód bežal buď na strane serveru alebo na strane klienta, spolu so service workermi bola zavedená nová vrstva. Service worker je umiestnený za webovými prehliadačmi, tak ako je zobrazené na obrázku 1. Service worker dokáže takto obsluhovať rôzne podnety zo všetkých stránok pod jeho kontrolou.

Práve vďaka tejto vrstve dokážeme odpovedať na podnety bez ohľadu na pripojenie používateľa - teda aj keď je offline, práve to čo je vhodné pre našu aplikáciu. Táto vrstva taktiež dokáže detegovať pomalé odozvy od servera a posilať obsah v cache.



Obr. 2: Názov obrázka [8]

Analogicky touto logikou je možná aj funkcia spomínaných push notifikácií po tom ako používateľ opustil webovú stránku. Teda service worker ostáva stále pracovať a môže s webom naďalej komunikovať.

4 Programovací jazyk PHP

5 Databáza

6 OpenCV

OpenCV je skratka pre Open Source Computer Vision a je to knižnica funkcií, ktoré sú užitočné pri programovaní aplikácií počítačového videnia v reálnom čase. Termín počítačové videnie sa používa pre subjekt, ktorý vykonáva analýzu digitálnych obrázkov a videí pomocou počítačového programu. Počítačové videnie je dôležitou súčasťou moderných disciplín, akými sú umelá inteligencia a strojové učenie [9]. V práci spomínam funkcie mnou považované za kľúčové pre túto prácu, samotná knižnica disponuje niekoľkonásobne viac funkciami, pre viac informácií je dostupná dokumentácia knižnice [10].

6.1 Čítanie obrázkov

Medzi jednu z najdôležitejších funkcií knižnice patrí funkcia na čítanie obrázkov. Knižnica OpenCV poskytuje funkciu s názvom `imread()`, ktorá slúži na načítanie obrázku do pamäte zo zvoleného súboru pomocou prvého argumentu funkcie.

Funkcia prečíta obrázok a uloží ho vo formáte matice, kde tvar matice zodpovedá rozmerom obrázka. Druhým argumentom funkcie je parameter menom `flags`. Tento parameter rozhoduje o tom ako je obrázok načítaný, teda môže byť načítaný ako farebný (predvolené), šedý (jeden kanál) alebo nezmenený, teda čítaný taký aký je, ale zachovávajúci všetky kanály transparentnosti.

6.2 Vlastnosti obrázka

Medzi funkcie pracujúce s vlastnosťami obrázkov radíme funkcie `shape()` a `resize()`

Funkcia `shape()` sa používa na získanie rozmerov obrázka alebo matrice. Táto funkcia vráti veľkosť obrázka alebo matice ako `n-ticu`, čo môže byť užitočné na pochopenie štruktúry údajov. Často sa používa na kontrolu vlastností obrazu, ako sú šírka, výška a farebné kanály.

Funkcia `resize()` sa používa na zmenu veľkosti obrázka na zadanú veľkosť alebo podľa danej mierky. Táto funkcia je kľúčová pri mnohých úlohách spracovania obrazu, kde je potrebné upraviť veľkosť obrázkov pred vykonaním operácií, ako je detekcia alebo klasifikácia objektov.

6.3 Vyobrazenie geometrických útvarov

Kreslenie tvarov a textu na obrázky je užitočné na vizualizáciu údajov alebo anotácií v aplikáciách počítačového videnia. OpenCV poskytuje rôzne funkcie na kreslenie

základných tvarov ako sú čiary(`cv2.line`), obdĺžniky(`cv2.rectangle`) a kruhy(`cv2.circle`), taktiež umožňuje pridávanie textu(`cv2.putText`) do obrázkov.

Záver

Literatúra

1. LEGION TARGETS. *The Evolution of Target Shooting: From Practice to Competition*. 2024. Dostupné tiež z: <https://www.legiontargets.com/blogs/legion/the-evolution-of-target-shooting-from-practice-to-competition>.
2. THE SCOTSMAN. *Final curtain: The last live pigeon shooting event at the Olympic Games, 1900*. 2008. Dostupné tiež z: <https://www.scotsman.com/sport/final-curtain-the-last-live-pigeon-shooting-event-at-the-olympic-games-1900-2474241>.
3. GEEKSFORGEEEKS. *History of JavaScript*. 2024. Dostupné tiež z: <https://www.geeksforgeeks.org/history-of-javascript/>.
4. MDN WEB DOCS. *What is JavaScript?* 2024. Dostupné tiež z: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/What_is_JavaScript.
5. GEEKSFORGEEEKS. *Introduction to JavaScript*. 2024. Dostupné tiež z: <https://www.geeksforgeeks.org/introduction-to-javascript/>.
6. HOOGENRAAD, W. *6 moderných programovacích jazykov a ich nevýhody*. 2023. Dostupné tiež z: <https://www.itpedia.nl/sk/2023/07/13/6-moderne-programmeertalen-en-hun-downsides/>.
7. GEEKSFORGEEEKS. *Introduction to Node.js*. 2023. Dostupné tiež z: <https://www.geeksforgeeks.org/node-js-introduction/>.
8. ATER, T. *Building Progressive Web Apps: Bringing the Power of Native to the Browser*. O'Reilly Media, 2017. ISBN 9781491961650. Dostupné tiež z: https://github.com/febycloud/PROG8110/blob/master/Tal-Ater-Building-Progressive-Web-Apps_-Bringing-the-Power-of-Native-to-the-Browser-OReilly-Media-2017.pdf.
9. TUTORIALSPPOINT. *OpenCV Python Tutorial*. [N.d.] Dostupné tiež z: https://www.tutorialspoint.com/opencv_python/index.htm.
10. OPENCV. *OpenCV Documentation*. [N.d.] Dostupné tiež z: <https://docs.opencv.org/4.x/index.html>.

Použitie nástrojov umelej inteligencie

OpenAI (2025), ChatGPT 4o, časť ??, generovanie vzorca záznamu použitia AI.

OpenAI (2024), ChatGPT 3, Záver, generovanie textu.

OpenAI (2025), ChatGPT 4o, dodatok B, generovanie kódu programu.

OpenAI (2024), ChatGPT 3, dodatok C, generovanie zoznamu.

Dodatok A: Algoritmus

Dodatok B: Výpis dlhého kódu

Dodatok C: Slovníček pojmů