

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH
TECHNOLÓGIÍ**

Evidenčné číslo: FIIT-16768-116249

**Zber údajov o čistokrvných plemenách a ich
predpríprava na komparatívnu analýzu**

BAKALÁRSKA PRÁCA

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH
TECHNOLÓGIÍ

Evidenčné číslo: FIIT-16768-116249

Zber údajov o čistokrvných plemenách a ich
predpríprava na komparatívnu analýzu
BAKALÁRSKA PRÁCA

Študijný program:	Informatika
Študijný odbor:	Informatika
Školiace pracovisko:	Ústav automobilovej mechatroniky
Vedúci záverečnej práce:	PhDr. Ján Sliacký
Konzultant:	titul, meno a priezvisko

Bratislava 2024

Adam Melikant

Sem sa vkladá naskenovaný originál zadania.

Pod'akovanie (nepovinné)

Na tomto mieste môže byť pod'akovanie napr. vedúcemu práce resp. konzultantom za pripomienky a odborné rady pri vypracovaní práce. Nie je zvykom d'akovať za rutinnú kontrolu, menšiu spoluprácu alebo všeobecné rady. Pozor, pod'akovanie v prípade využitia inej práce sa realizuje formou citácie v Použitej literatúre, odkazy na citáciu sa musia uviesť aj na zodpovedajúcich miestach v texte.

ANOTÁCIA BAKALÁRSKEJ PRÁCE

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČÝCH TECHNOLOGIÍ

Študijný odbor: Informatika

Študijný program: Informatika

Autor: Adam Melikant

Bakalárska práca: Zber údajov o čistokrvných plemenách a ich
predpríprava na komparatívnu analýzu

Vedúci práce: PhDr. Ján Sliacký

Mesiac a rok odovzdania: Máj 2024

Kľúčové slová:

Anotácia obsahuje informáciu o cieľoch práce a jej stručnom obsahu. V závere anotácie sa charakterizuje splnenie cieľa, dosiahnuté výsledky a význam celej práce. Anotácia sa píše súvisle ako jeden odsek (rozsah 100 až 500 slov).

Táto práca sa zaoberá zberom a dokumentáciou údajov o čistokrvných zvieratách, s osobitným zreteľom na mačky, ktoré sú významnou súčasťou spoločnosti. Hlavným cieľom projektu "Clearblooded Animal Species Collection" je vytvoriť komplexnú a spoľahlivú databázu informácií o mačkách. Databáza bude zahŕňať detailné údaje o jednotlivých plemenách, vrátane ich historického pôvodu, fyzických charakteristík, temperamentu a zdravotných požiadaviek. Kolekcia informácií má slúžiť ako nástroj pre majiteľov mačiek, chovateľov a veterinárov, aby mohli lepšie porozumieť jednotlivým plemenám a poskytnúť im optimálnu starostlivosť. Vytvorenie tejto databázy nie je obmedzené na individuálne príspevky, ale otvára sa komunite, aby prispeli svojimi poznatkami a skúsenosťami. Spoločná kolektívna vedomosť môže výrazne obohatiť a rozšíriť informácie o mačkách, čo má dlhodobý prospech pre všetkých, ktorí sa zaujímajú o tieto úžasné zvieratá. Projekt zdôrazňuje význam vzdelávania a informovanosti pre zodpovednú starostlivosť o mačky a ich plemenných spoločníkov. Výsledkom práce bude dockerizovaná aplikácia nasadená na VPS s príručkou pre používateľa a programátorskou príručkou pre implementáciu na čistom VPS. Práca ponúka vhodnú platformu pre študentov zameraných na vývoj webových aplikácií a prácu s dátami. Vytvorená databáza poskytne dostatok údajov pre reálne časové vyhodnotenie a základnú exploratívnu analýzu. Taktiež poskytne základ pre ďalšiu analytickú prácu s rozsiahlymi dátovými množinami a ich spracovanie pomocou pokročilých techník machine learningu v rámci ďalších výskumných projektov.

Táto práca sa zameriava na revolúciu v evidencii a analýze údajov o čistokrvných zvieratách, so zvláštnym dôrazom na mačky. Súčasné dostupné databázy sú zastaralé, nespĺňajú moderné štandardy štruktúry a rozsahu údajov. V dnešnej dobe nám techniky strojového učenia ponúkajú nové možnosti spracovania týchto údajov, ktoré sú oveľa objemnejšie a detailnejšie. Cieľom tejto práce je analyzovať prístupy k zberu, evidencii a spracovaniu údajov o chovných zvieratách v cloude. Navrhne nový prístup, ktorý umožní zber štandardných a expertných údajov s cieľom podporiť informované rozhodnutia chovateľov. Vytvorí aplikáciu implementujúcu tento prístup, ktorá bude nasadená v dockerovom kontajneri na VPS. Súčasťou práce bude aj podrobná používateľská a programátorská príručka. Táto práca otvára dvere pre budúce spracovanie dát pokročilými technikami machine learningu a môže slúžiť ako základ pre ďalšie výskumy v tejto oblasti. Je vhodná pre študentov zaujímajúcich sa o vývoj webových aplikácií a analýzu dát. Výsledkom práce bude moderný nástroj, ktorý podporí rozvoj lepšieho chovu a starostlivosti o našich milovaných spoločníkov – mačky. Práca sa zaoberá analýzou existujúcich databáz o čistokrvných chovných zvieratách s dôrazom na moderné techniky spracovania údajov a ich komparatívnu analýzu. Súčasné databázy výrazne zaostávajú vo forme, štruktúre a obsahu údajov, potrebných pre dnešné požiadavky, ktoré vyžadujú komplexnejšie a detailnejšie štrukturované údaje, vhodné pre aplikácie strojového učenia. Súčasná spoločenská poptávka zameraná na spracovanie a porovnanie údajov, ktoré majú chovatelia zvierat k dispozícii, zdôrazňuje potrebu vytvoriť nový prístup k zberu a spracovaniu týchto údajov v cloude. Cieľom práce je navrhnuť a implementovať inovatívny prístup k zberu a predpríprave údajov o chovných zvieratách pre komparatívnu analýzu. Tento prístup umožní zber štandardných a expertných údajov súčasne. Práca zahŕňa vytvorenie aplikácie, ktorá bude podporovať navrhnutý prístup, s dôrazom na nasadenie v cloude a vizualizáciu údajov. Kritériá používateľského zážitku a výkonnosti sú dôležitými aspektmi implementácie, pričom berie do úvahy diverzifikáciu úrovní hodnotiteľov. Výsledkom práce bude dockerizovaná aplikácia nasadená na VPS s príručkou pre používateľa a programátorskou príručkou pre implementáciu na čistom VPS. Práca ponúka vhodnú platformu pre študentov zameraných na vývoj webových aplikácií a prácu s dátami. Vytvorená databáza poskytne dostatok údajov pre reálne časové vyhodnotenie a základnú exploratívnu analýzu. Taktiež poskytne základ pre ďalšiu analytickú prácu s rozsiahlymi dátovými množinami a ich spracovanie pomocou pokročilých techník machine learningu v rámci ďalších výskumných projektov. Táto práca sa zameriava na analýzu existujúcich databáz obsahujúcich údaje o chovných zvieratách, s dôrazom na čistokrvné (chovné) mačky, ktoré sú momentálne dostupné online. Zistili sme, že tieto databázy sú zastaralé a potrebujú modernizáciu vzhľadom na štruktúru, veľkosť a formu uchovávaných údajov. Súčasné požiadavky na spracovanie údajov vyžadujú komplexnejšie, štrukturované a objemné dáta, aby mohli byť efektívne využité v moderných metódach strojového učenia. Cieľom tejto práce je analyzovať prístupy k zberu, spracovaniu a štandardizácii údajov o chovných zvieratách v cloude. Navrhujeme inovatívny prístup, ktorý umožní zber štandardných a expertných údajov súčasne a predprípravi ich na komparatívnu analýzu. Vytvorená aplikácia bude integrovať tento prístup a poskytne užívateľom

ABSTRACT OF THE BACHELOR THESIS

Slovak University of Technology in Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGY

Study Branch: Informatics

Study Programme: Informatics

Author: Adam Melikant

Bachelor Thesis: názov práce v angličtine

Supervisor: titul pred menom, meno, priezvisko, titul za menom

Submitted: month and year

Keywords:

Text abstraktu vo svetovom jazyku je potrebný pre integráciu do medzinárodných informačných systémov (napr. The Network Digital Library of Theses and Dissertations). Ak nie je možné jazykovú verziu umiestniť na jednej strane so slovenským abstraktom, je potrebné umiestniť ju na samostatnú stranu (cudzojazyčný abstrakt nemožno deliť a uvádzať na dvoch stranách).

Táto práca sa zaoberá projektom "Clearblooded Animal Species Collection," ktorý má za cieľ vytvoriť komplexnú a spoľahlivú databázu údajov o čistokrvných zvieratách, so zvláštnym dôrazom na mačky. Databáza poskytne detailné informácie o jednotlivých plemenách, ich historickom pôvode, fyzických charakteristikách, temperamentu a zdravotných požiadavkách. Cieľom projektu je uľahčiť majiteľom mačiek, chovateľom a veterinárom lepšie porozumieť rôznym plemenám a poskytnúť im optimálnu starostlivosť. Dôležitou súčasťou projektu je možnosť prispievať do databázy a vytvoriť tak komunitu, ktorá sa delí o svoje poznatky a skúsenosti. Spoločná kolektívna vedomosť môže výrazne obohatiť informácie o mačkách, čo má prospech pre všetkých záujemcov o tieto zvieratá. Projekt zdôrazňuje význam vzdelávania a informovanosti pre zodpovednú starostlivosť o mačky a ich plemenných spoločníkov. Výsledkom práce je dockerizovaná aplikácia nasadená na VPS s príručkou pre používateľa a programátorskou príručkou pre implementáciu na čistom VPS. Práca poskytuje platformu pre študentov špecializujúcich sa na vývoj webových aplikácií a prácu s dátami, a otvára cestu k ďalším výskumným projektom v oblasti analýzy a spracovania rozsiahlych dátových množín.

S nasadením moderných technológií a pokročilých metód strojového učenia sme zameraní na vytvorenie inovatívnej platformy na zber, spracovanie a analýzu údajov o čistokrvných zvieratách, so zvláštnym dôrazom na mačky. Existujúce databázové systémy často nedostačujú vo forme, štruktúre a kvalite údajov pre dnešné potreby. Táto práca sa sústreďí na analýzu súčasných prístupov k zberu údajov a navrhuje nový komplexný prístup, ktorý umožňuje získavanie štandardných a expertných údajov a ich následné komparatívne spracovanie.

Cieľom tejto práce je vytvorenie aplikácie, ktorá umožní efektívny zber údajov o mačkách a umožní komplexnú analýzu týchto údajov. Okrem technických aspektov sa zaoberáme aj otázkami použiteľnosti aplikácie a zohľadňujeme faktory, ako je možnosť nasadenia v cloude a výkonnosť vzhľadom na plánovaný objem dát. Táto práca prispeje k zlepšeniu porozumenia potrieb a správania mačiek, čím podporuje zodpovedný chov a starostlivosť o tieto zvieratá. Týmto spôsobom otvára cestu pre budúci výskum v oblasti spracovania údajov o chovných zvieratách a ich aplikáciu v digitálnom svete.

Obsah

Zoznam použitých skratiek a označení.....	14
Úvod.....	15
1 Web framework.....	21
1.1 Python web frameworks.....	21
1.2 Django Framework.....	22
1.3 Prečo používať Django framework.....	22
1.4 HTTP metódy.....	23
1.4.1 GET.....	23
1.4.2 POST.....	23
1.5 Status kódy.....	23
1.5.1 2XX Success.....	23
1.6 Bezpečnosť.....	24
1.6.1 XSS(Cross-Site Scripting).....	24
1.6.2 CSRF token(Cross-Site Request Forgery).....	24
2 Databázy.....	24
2.1 Čo je databáza.....	24
2.2 Relácie dát.....	26
2.3 Schéma databázy.....	27
2.4 Data manipulation language (DML).....	27

2.5 Transakcie v databázach.....	28
3 Docker.....	29
3.1 Dockerfiles.....	30
3.2 The differences between dedicated hosts, virtual machines, and Docker.....	30
4 VPN(Virtual Private Network).....	30
5 Analýza / Návrh riešenia.....	30
5.1 Porovnanie podobných riešení.....	30
5.2 Spôsob riešenia.....	31
6 Opis Riešenia.....	31
6.1 Špecifikácia požiadaviek.....	31
6.2 Návrh.....	31
6.3 Implementácia.....	31
6.4 Overenie riešenia.....	31
7 Zhodnotenie.....	31
8 Technická dokumentácia.....	31
9 Plán Práce.....	32
10 Názov kapitoly.....	33
10.1 Názov podkapitoly.....	33
10.1.1 Názov tretej úrovne.....	33
11 Výsledky práce.....	34
11.1 Obrázky, tabuľky, rovnice, krížové odkazy.....	34
11.1.1 Obrázky.....	34
11.1.2 Tabuľky.....	35
11.1.3 Rovnice, vzorce.....	35
11.1.4 Literatúra.....	36
Prílohy.....	37

Zoznam použitých skratiek a označení

μ	micro, 10^{-6}
SI	Système International
V	volt, základná jednotka napätia v sústave SI

Úvod

Predpisy, ktorými sa riadi vypracovania a odovzdávania záverečných prác na STU sú zverejnené na http://www.stuba.sk/sk/studentov/legislativa/predpisy-suvisiace-s-vypracovanim-a-odovzdanim-zaverecnych-prac.html?page_id=4563. Štruktúru, náležitosti a citovanie bibliografických odkazov upravuje Metodické usmernenie MŠ VVŠ SR č. 56/2011 o náležitostiach záverečných prác ich bibliografickej registrácii, uchovávaní a sprístupňovaní.

Hlavnú textovú časť záverečnej práce tvorí: úvod, jadro, záver, resumé (povinné iba v prípade, ak je práca vypracovaná v inom ako štátnom jazyku), zoznam použitej literatúry.

V úvode autor stručne a výstižne charakterizuje stav poznania alebo praxe v oblasti, ktorá je predmetom záverečnej práce a oboznamuje čitateľa s významom, cieľmi a zámermi záverečnej práce. Autor v úvode zdôrazňuje, prečo je záverečná práca dôležitá a prečo sa rozhodol spracovať danú tému.

V tejto práci sa hlboko zameriavame na projekt zaznamenávania údajov o plemenných mačkách a jeho technickú implementáciu, s dôrazom na význam pre starostlivosť o tieto spoločnícke zvieratá. Práca je systematicky rozdelená do niekoľkých kapitol, pričom každá z nich približuje rôzne aspekty projektu a jeho vplyvu na širšiu verejnosť.

V prvom rade sa venujeme procesu vytvárania databázy, skúmame jej štruktúru a obsah, s cieľom zabezpečiť detailný pohľad na informácie o plemenných mačkách. Následne sa zameriame na nasadenie aplikácie na virtuálny server (VPS), aby bola dostupná online, pričom okrem toho poskytneme príručku pre používateľov, ktorí chcú prispieť do databázy, a programátorskú príručku pre rozvoj projektu.

Cieľom celej práce je vytvoriť komplexnú databázu obsahujúcu historické, fyziologické, behaviorálne a zdravotné údaje o rôznych plemenách mačiek. Táto databáza bude základným zdrojom informácií nielen pre majiteľov mačiek, ale aj pre chovateľov a veterinárov, aby mohli poskytovať odborne informovanú starostlivosť o tieto zvieratá.

Práca zároveň poskytuje platformu pre študentov a vývojárov, ktorí sa zaujímajú o vývoj webových aplikácií a spracovanie dát. Vytvorená databáza bude slúžiť nielen na základnú exploratívnu analýzu, ale aj ako základ pre ďalšiu analytickú prácu s rozsiahlymi dátovými množinami a ich spracovanie pomocou pokročilých techník strojového učenia v rámci budúcich výskumných projektov.

S narastajúcou digitalizáciou spoločnosti a vývojom technológií sa otvárajú nové možnosti v oblasti zberu, spracovania a analýzy dát. Táto práca pristupuje k tejto problematike v kontexte evidencie údajov o čistokrvných mačkách, zdôrazňujúc potrebu moderného prístupu, ktorý zabezpečí rozsiahlejšie a dôkladnejšie štruktúrované údaje pre techniky analýzy dát.

Analyzujeme súčasne verejne dostupné databázy údajov o chovných mačkách a poukazujeme na ich nedostatky v kontexte súčasných požiadaviek na dátové spracovanie.

Naším cieľom je vytvoriť nový prístup k zberu, uchovávaniu a spracovaniu údajov o plemenných mačkách, ktorý zohľadňuje aktuálne potreby a poskytuje komplexný pohľad na ich chov a zdravie.

Táto práca nielenže navrhuje nový prístup k získavaniu údajov o mačkách, ale integruje ho do ucelenej aplikácie. Zároveň sa zameriavame na otázky použiteľnosti aplikácie a berieme do úvahy kritériá výkonnosti s ohľadom na očakávaný objem dát. Cieľom je vytvoriť aplikáciu, ktorá prinesie pridanú hodnotu v oblasti starostlivosti o mačky a poskytne užívateľom efektívne nástroje na zaznamenávanie, uchovávanie a analýzu údajov.

V závere tejto práce hodnotíme dosiahnuté ciele a diskutujeme o význame projektu pre budúcnosť. Otvárame dvere pre ďalšie projekty, ktoré budú vychádzať z vytvorenej databázy a budú môcť prispieť k rozvoju vedy a praxe starostlivosti o mačky a iné spoločenské zvieratá.

1 Webový rámec

Rámec je sada softvéru, ktorá organizuje architektúru aplikácie a zjednodušuje prácu vývojára. Rámec je možné prispôbiť rôznym účelom a poskytuje praktické nástroje na zrýchlenie práce programátora. Niektoré funkcie, ktoré sa pravidelne používajú na webových stránkach, môžu byť automatizované, ako napríklad správa databázy a správa používateľov.

[2]

1.1 Webové rámce v Pythone

V Pythone je k dispozícii veľké množstvo rámcov a knižníc, ktoré vám umožnia vytvárať webové služby založené na formáte JSON; a množstvo dostupných možností môže byť pre vás preplňujúce. Napríklad môžete zvážiť

- Flask,
- Django,
- Web2py and
- CherryPy to name just a few.

Tieto rámce a knižnice ponúkajú rôzne sady funkcií a úrovni sofistikovanosti. Napríklad Django je plnohodnotný webový rámec; teda je zameraný na vývoj nie len webových služieb, ale aj plnohodnotných webových stránok. Pre naše účely je však pravdepodobne nadbytočný a rozhranie Django Rest je len časťou oveľa väčšej infraštruktúry. To však neznamená, že by sme nemohli použiť Django na vytvorenie našich služieb kníh; existujú však jednoduchšie možnosti. Web2py je ďalší plnohodnotný webový rámec, ktorý vylúčujeme z rovnakého dôvodu. Na rozdiel od toho sú Flask a CherryPy považované za neplnohodnotné webové rámce (aj keď ich môžete použiť na vytvorenie plnohodnotnej webovej aplikácie). To znamená, že sú ľahšie a rýchlejšie na začiatok. Pôvodne bol CherryPy zameraný skôr na poskytovanie možnosti vzdialeného volania funkcií, ktoré umožňovali volanie funkcií cez HTTP; avšak toto bolo rozšírené na poskytovanie viac možností podobných REST. V tomto kapitole sa budeme zameriavať na Flask, pretože je jedným z najviac používaných rámcov na ľahké RESTful služby v Pythone.

1.2 Django Framework

Django vznikol v roku 2003 v tlačovej agentúre v Lawrence, Kansas. Je to webový rámec, ktorý používa Python na vytváranie webových stránok. Jeho cieľom je písať veľmi rýchle dynamické webové stránky. V roku 2005 sa agentúra rozhodla zverejniť zdrojový kód Django pod licenciou BSD. V roku 2008 bola vytvorená Django Software Foundation na podporu a rozvoj Django. O pár mesiacov neskôr bola vydaná verzia 1.00 rámcu. [2]

1.3 Výhody používania Django rámca

Existuje mnoho výhod používania Djangového rámca, z ktorých vyberáme niekoľko významných:

- **Licencia BSD:** Django je publikovaný pod licenciou BSD, čo zabezpečuje, že webové aplikácie môžu byť používané a modifikované bez problémov; je tiež zadarmo.
- **Plná prispôsobiteľnosť:** Django je plne prispôsobiteľný. Vývojári ho môžu jednoducho prispôbiť vytváraním modulov alebo prepisovaním metód rámca.
- **Modularita:** Táto modularita prináša ďalšie výhody. Existuje mnoho modulov pre Django, ktoré môžete integrovať. Tým, že často nájdete vysokokvalitné moduly od iných ľudí, získate pomoc s ich prácou.
- **Využitie jazyka Python:** Používanie jazyka Python v tomto rámci vám umožňuje využívať výhody všetkých knižníc Pythonu a zaručuje veľmi dobrú čitateľnosť kódu.
- **Zameranie na dokonalosť:** Django je rámec, ktorého hlavným cieľom je dokonalosť. Bol vytvorený špeciálne pre ľudí, ktorí chcú čistý kód a dobrú architektúru svojich aplikácií. Úplne dodržiava filozofiu "Neprepišuj sa" (DRY), čo znamená udržiavať kód jednoduchý bez nutnosti kopírovania/pálenia rovnakých častí na viacerých miestach.
- **Kvalita:** Čo sa týka kvality, Django integruje množstvo efektívnych spôsobov vykonávania jednotkových testov.
- **Podpora od komunity:** Django je podporovaný silnou komunitou. To je veľmi dôležitý aktívum, pretože vám umožňuje rýchlo riešiť problémy a opravovať chyby. Vďaka komunite môžeme tiež nájsť príklady kódu, ktoré ukazujú najlepšie postupy.

Pozícia Djangového rámca na webe

Django má samozrejme aj nevýhody. Pri začatí používania rámca začína vývojár s fázou učenia. Dĺžka tejto fázy závisí od rámca a od vývojára. Fáza učenia Django je relatívne krátka, ak vývojár pozná jazyk Python a objektovo orientované programovanie.

Taktiež môže nastať situácia, kedy je zverejnená nová verzia rámca, ktorá mení niektorú syntax. Napríklad syntax URL adries v šablónach bola zmenená vo verzii 1.5 Djangového rámca. Napriek tomu dokumentácia poskytuje podrobnosti o každej aktualizácii Djangovej verzie.

Webová adresa: <https://www.djangoproject.com/>

[2]

1.4 HTTP metódy

HTTP podporuje niekoľko rôznych príkazov na žiadosť, nazývaných HTTP metódy. Každá správa HTTP žiadosti má metódu. Metóda hovorí serveru, akú akciu vykonať (získať webovú stránku, spustiť program brány, odstrániť súbor atď.). Tabuľka 1-2 uvádza päť bežných HTTP metód.

V tejto žiadosti je metóda „GET“. V iných diskusiách o REST môžete vidieť, že sa nazýva „HTTP sloveso“ alebo „HTTP akcia“. Názov HTTP metódy je podobný názvu metódy v programovacom jazyku: naznačuje, ako očakáva klient, že server spracuje túto obálku. V tomto prípade klient (môj webový prehliadač) snaží získať niektoré informácie zo servera (www.oreilly.com).

1.4.1 GET

Odoslanie pomenovanej zdrojovej informácie zo servera klientovi.

1.4.2 POST

Uložte údaje z klienta do pomenovanej zdrojovej informácie servera. [4]

1.5 Status kódy

1.5.1 2XX Success

1.5.1.1 200 ("OK")

Význam: Veľmi vysoký.

V väčšine prípadov je to kód, ktorý si klient želá vidieť. Naznačuje, že server úspešne vykonal akciu, ktorú klient požadoval, a že nie je vhodný žiadny konkrétny kód v sérii 2xx. Moja služba záložiek posiela tento kód spolu s reprezentáciou, keď klient požiada o zoznam záložiek. Telo entity: Pre GET požiadavky reprezentácia zdroja, ktorú klient požadoval. Pre iné požiadavky reprezentácia aktuálneho stavu vybraného zdroja alebo popis práve vykonanej akcie.

1.5.1.2 201 ("Created")

Význam: Vysoký.

Server posiela tento stavový kód, keď vytvorí nový zdroj na požiadavku klienta. Moja služba záložiek posiela tento kód ako odpoveď na POST požiadavku, ktorý vytvára nový užívateľský účet alebo záložku. Hlavičky odpovedí: Hlavička Location by mala obsahovať kanonický URI k novému zdroju. [3]

1.6 Bezpečnosť

1.6.1 XSS(Cross-Site Scripting)

1.6.2 CRSF token(Cross-Site Request Forgery)

2 Databázy

There are several different types of database system in common use today including Object databases, NoSQL databases and (probably the most common) Relational Databases. This chapter focusses on Relational Databases as typified by database systems such as Oracle, Microsoft SQL Server and MySQL.

2.1 Čo je databáza

A database is essentially a way to store and retrieve data.

Typically, there is some form of query language used with the database to help select the information to retrieve such as SQL or Structured Query Language.

In most cases there is a structure defined that is used to hold the data (although this is not true of the newer NoSQL or non-relational unstructured databases such as CouchDB or MongoDB).

In a

The diagram illustrates a database table structure. A light blue rectangular box represents the table, labeled 'students' at the bottom right. Inside the box, the first row contains five column headers: 'id', 'name', 'surname', 'subject', and 'email'. Below these headers are four data rows. A yellow vertical highlight is positioned over the 'id' column, with the word 'attribute' written above it. A purple horizontal highlight is positioned over the first data row, with the word 'row' written to its left. The data rows contain the following values: (cs_18, Phoebe, Cooke, Animation, pc@my.com), (cs_21, Gryff, Jones, Games, gj@my.com), (cs_27, Adam, Fosh, Music, af@my.com), and (cs_29, Jasmine, Smith, Games, js@my.com).

id	name	surname	subject	email
cs_18	Phoebe	Cooke	Animation	pc@my.com
cs_21	Gryff	Jones	Games	gj@my.com
cs_27	Adam	Fosh	Music	af@my.com
cs_29	Jasmine	Smith	Games	js@my.com

students

Relational Database the data is held in tables, where the columns define the properties or attributes of the data and each row defines the actual values being held, for example:

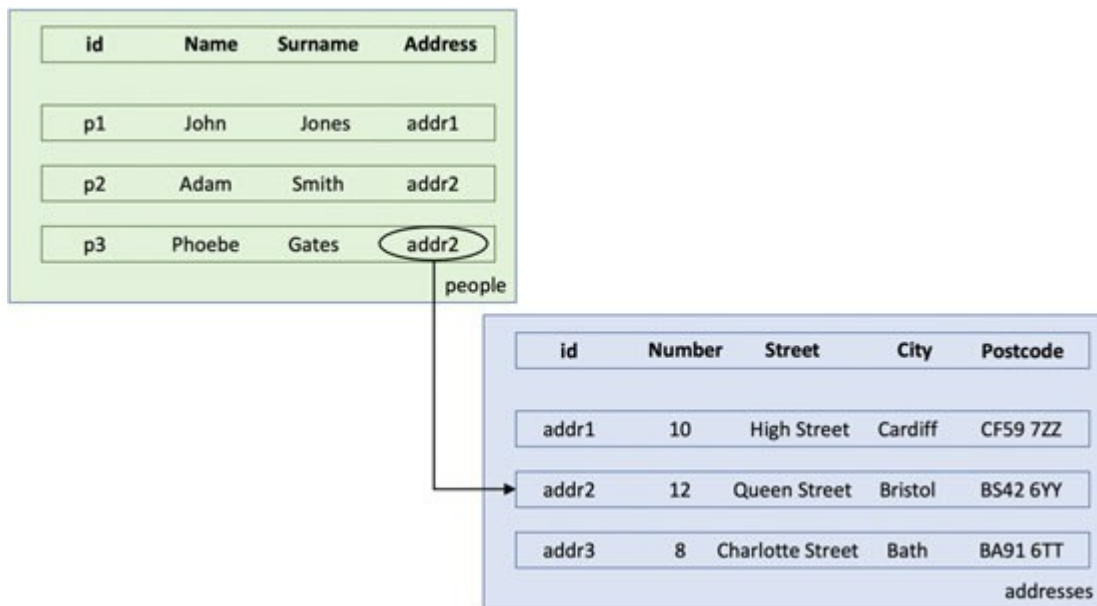
In this diagram there is a table called students; it is being used to hold information about students attending a meeting. The table has 5 attributes (or columns) defined for id, name, surname, subject and email.

In this case, the id is probably what is known as a primary key. The primary key is a property that is used to uniquely identify the student row; it cannot be omitted and must be unique (within the table). Obviously names and subjects may well be duplicated as there may be more than one student studying Animation or Games and students may have the same first name or surname. It is probable that the email column is also unique as students probably don't share an email address but again this may not necessarily be the case.

You might at this point wonder why the data in a Relational Database is called relational and not tables or tabular? The reason is because of a topic known as relational algebra that underpins Relational Database theory. Relational Algebra takes its name from the mathematical concept known as a relation. However, for the purposes of this chapter you don't need to worry about this and just need to remember that data is held in tables.

2.2 Relácie dát

When the data held in one table has a link or relationship to data held in another table then an index or key is used to link the values in one table to another. This is illustrated below for a table of addresses and a table of people who live in that address. This shows for example, that 'Phoebe Gates' lives at address 'addr2' which is 12 Queen Street, Bristol, BS42 6YY.



This is an example of a many to one (often written as many:1) relationship; that is there are many people who can live at one address (in the above Adam Smith also lives at address 'addr2'). In Relational Databases there can be several different types of relationship such as:

- one:one where only one row in one table references one and only one row in another table. An example of a one to one relationship might be from a person to an order for a unique piece of jewellery.
- one:many this is the same as the above address example, however in this case the direction of the relationship is reversed (that is to say that one address in the addresses table can reference multiple persons in the people table).
- many:many This is where many rows in one table may reference many rows in a second table. For example, many students may take a particular class and a student may take many classes. This relationship usually involves an intermediate (join) table to hold the associations between the rows.

2.3 Schéma databázy

The structure of a Relational Database is defined using a Data Definition Language or Data Description Language (a DDL).

Typically, the syntax of such a language is limited to the semantics (meaning) required to define the structure of the tables. This structure is known as the database

schema. Typically, the DDL has commands such as CREATE TABLE, DROP TABLE (to delete a table) and ALTER TABLE (to modify the structure of an existing table).

2.4 Data manipulation language (DML)

Data can also be inserted into a table or existing data in a table can be updated. This is done using the Data Manipulation Language (DML).

For example, to insert data into a table we merely need to write an INSERT SQL statement providing the values to be added and how they map to the columns in the table:

```
INSERT INTO 'students' ('id', 'name', 'surname', 'subject',  
'email') VALUES ('6', 'James', 'Andrews', 'Games',  
'ja@my.com');
```

This would add the row 6 to the table students with the result that the table would now have an additional row.

2.5 Transakcie v databázach

Another important concept within a database is that of a Transaction. A Transaction represents a unit of work performed within a database management system (or similar system) against a database instance, and is independent of any other transaction.

Transactions in a database environment have two main purposes

- To provide a unit of work that allows recovery from failures and keeps a database consistent even in cases of system failure, when execution stops (completely or partially). This is because either all the operations within a transaction are performed or none of them are. Thus, if one operation causes an error then all the changes being made by the transaction thus far are rolled back and none of them will have been made.
- To provide isolation between programs accessing a database concurrently. This means that the work being done by one program will not interact with another programs work.

A database transaction, by definition, must be atomic, consistent, isolated and durable:

- Atomic This indicates that a transaction represents an atomic unit of work; that is either all the operations in the transaction are performed or none of them are performed.
- Consistent Once completed the transaction must leave the data in a consistent state with any data constraints met (such as a row in one table must not reference a non-existent row in another table in a one to many relationship etc.).
- Isolated This relates to the changes being made by concurrent transactions; these changes must be isolated from each other. That is, one transaction cannot see the changes being made by another transaction until the second transaction completes and all changes are permanently saved into the database.
- Durable This means that once a transaction completes then the changes it has made are permanently stored into the database (until some future transaction modifies that data).

Database practitioners often refer to these properties of database transactions using the acronym ACID (for Atomic, Consistent, Isolated, Durable).

Not all databases support transactions although all commercial, production quality databases such as Oracle, Microsoft SQL Server and MySQL, do support transactions. [1]

3 Docker

Docker is Infrastructure as Code.² Docker automates operating system preparation, installing and patching Oracle software, and configuring databases. Users don't need to understand or remember detailed steps. Calling Docker and supplying optional parameters to customize the environment is all that's required. Users who aren't familiar with or comfortable installing a database no longer rely on a DBA to create new working environments.

Docker presents an even broader set of possibilities for database administrators beyond just automating database installations. Consider the effort to set up a primary and standby database in Data Guard. At a high level:

- Provision the hosts
- Install software
- Configure networking
- Create and configure a primary database
- Prepare and recover the standby database
- Set up the broker
- Test and validate

[6]

3.1 Dockerfiles

A Dockerfile is simply a plain text file that contains a set of user-defined instructions. When a Dockerfile is called by the docker image build command, which we will look at next, it is used to assemble a container image.

3.2 The differences between dedicated hosts, virtual machines, and Docker

Docker is a container management system that helps us efficiently manage Linux Containers (LXC) more easily and universally. This lets you create images in virtual environments on your laptop and run commands against them. The actions you perform to the containers, running in these environments locally on your machine, will be the same commands or operations that you run against them when they are running in your production environment.

This helps us in that you don't have to do things differently when you go from a development environment, such as the one on your local machine, to a production environment on your server. Now, let's take a look at the differences between Docker

containers and typical virtual machine environments. [5]

4 VPN(Virtual Private Network)

5 Analýza / Návrh riešenia

5.1 Porovnanie podobných riešení

5.2 Spôsob riešenia

6 Opis Riešenia

6.1 Špecifikácia požiadaviek

6.2 Návrh

TimeStampedModel: This is an abstract base class providing `created_at` and `updated_at` fields to all models that inherit from it. Making it abstract ensures it does not create a table in the database, while still providing common functionality to child models. This is a common practice in Django to avoid code duplication.

1. **Attribute Model:** Represents a generic attribute with a name, type, and a boolean indicating if it's required. It inherits from `TimeStampedModel`, thus including creation and update timestamps.
2. **Form Model:** Represents a form with a name and a boolean to indicate if it should be included in a gallery. Also inherits from `TimeStampedModel`.
3. **Record Model:** Links records to users (`User` model from Django's auth system) and includes a description. It provides methods for upvoting, downvoting, calculating net votes, and retrieving comments, showcasing a feature-rich approach. Inheritance from `TimeStampedModel` adds timestamp functionality.
4. **Vote Model:** Tracks user votes on records with a choice between upvote and downvote. The `unique_together` constraint prevents multiple votes on the same record by the same user, ensuring data integrity.
5. **RecordComment Model:** Allows comments on records, linked to both the `Record` and `User` models. The `approved_by_admin` field suggests a moderation feature.
6. **FormAttribute Model:** Represents a relationship between forms and attributes, with an additional field to indicate if the attribute should be displayed in a gallery. This model illustrates a many-to-many relationship between forms and attributes.
7. **FormAttributeData Model:** Stores values for form attributes in records, demonstrating a way to dynamically store form data.
8. **Gallery Model:** Manages gallery settings for forms, linked to the `Form` model. The `unique_together` constraint in the `Meta` class ensures each gallery is unique for a form at a given creation time.
9. **Use of Django's User and Group Models:** Integrating Django's built-in authentication and authorization models (`User` and `Group`) adds robustness to your application, enabling user management and access control without needing to reinvent these features.

Overall, your database design showcases a comprehensive approach to building a dynamic form and record management system, leveraging Django's capabilities for efficient and maintainable code. The inclusion of abstract models, many-to-many relationships, and the integration of built-in Django models are indicative of a thoughtful and scalable design.

6.2.1 1. One-to-Many Relationships (ForeignKey)

- **Record to User:** Each `Record` is associated with a `User`. This relationship allows tracking of which user created or is associated with a particular record. It's essential for any user-centric application where actions are tied to specific users.
 - **Alternative:** If anonymity is desired, this relationship could be removed, making records user-independent.
- **Vote to User and Record:** Each `Vote` is linked to both a `User` and a `Record`. This setup is crucial for implementing a voting system where each vote is tied to a user and a record, preventing multiple votes by the same user on the same record.
 - **Alternative:** A many-to-many relationship could be used with additional logic to prevent duplicate votes, but the current approach is more straightforward and efficient.
- **RecordComment to User and Record:** Links comments to both the user who made them and the record they are about. This is standard for comment systems, ensuring traceability and context.
 - **Alternative:** If comments need not be user-specific, the link to `User` can be removed, though this would be unusual for most applications.
- **FormAttribute to Form and Attribute:** Each `FormAttribute` instance relates a `Form` to an `Attribute`. This design enables dynamic form structures where each form can have various attributes.
 - **Alternative:** A direct many-to-many relationship between `Form` and `Attribute` could be used, but it would be less flexible in terms of adding additional information specific to the form-attribute pairing.
- **FormAttributeData to Record and FormAttribute:** Captures the value of a specific attribute for a record. This relationship is key for storing form responses.
 - **Alternative:** A more complex single table could be used to store all form data, but it would be less flexible and harder to manage.

2. Many-to-Many Relationships

- **Implicit through FormAttribute:** The relationship between `Form` and `Attribute` is essentially many-to-many, facilitated by the `FormAttribute` model. This allows a form to have multiple attributes and an attribute to be part of multiple forms.
 - **Alternative:** Django's built-in `ManyToManyField` could be used directly, but using an intermediary model (`FormAttribute`) provides more flexibility for additional fields and complex relationships.

3. Abstract Base Class

- **TimeStampedModel:** Used as a base class for other models, it doesn't create a direct relationship but provides a common set of fields (`created_at`,

updated_at) to all inheriting models. This approach promotes DRY (Don't Repeat Yourself) principles.

Design Rationale

- **User-Centric Design:** Linking records, votes, and comments to users is typical in applications where user actions and contributions are central.
- **Flexibility and Scalability:** The use of intermediary models like `FormAttribute` and `FormAttributeData` allows for a highly flexible and scalable design, accommodating various forms and attributes dynamically.
- **Data Integrity and Functionality:** Relationships like the unique constraints in the `Vote` model ensure data integrity and support essential functionalities like voting systems.

Alternatives and Trade-offs

- Alternatives often involve trade-offs between simplicity and flexibility. For instance, using direct many-to-many relationships might simplify the design but at the cost of losing the ability to add additional context or constraints specific to the relationships.
- The choice between alternatives depends on the specific requirements and future scalability needs of your application. The current design is well-suited for a dynamic, user-centric application with a focus on flexibility and data integrity.

6.3 Implementácia

6.4 Overenie riešenia

7 Zhodnotenie

8 Technická dokumentácia

Zoznam použitej literatúry

- [2] Samuel Dauzon,Aidas Bendoraitis,Arun Ravindran, Django: Web Development, 2016, ISBN 978-1-78712-138-6
- [4] David Gourley,Brian Totty, HTTP the definitive guide, 2002, ISBN 978-1-56592-509-0
- [3] Leonard Richardson,Sam Ruby, Restful Web Services, 2007, ISBN 13: 978-0-596-52926-0
- [1] John Hunt, Advanced Guide to Python 3 Programing, 2019, ISBN 978-3-030-25942-6
- [6] Sean Scott, Oracle on Docker, 2023, ISBN 978-1-4842-9032-3
- [5] Russ McKendrick, Mastering Docker, fourth edition, 2015, ISBN 978-1-83921-657-2

9 Plán Práce

10 Názov kapitoly

Záverečná práca má obvykle tieto hlavné časti:

- súčasný stav riešenej problematiky doma a v zahraničí,
- cieľ práce, formulácia problému
- metodika práce a metódy skúmania (alebo materiály a metódy skúmania)
- výsledky práce,
- zhodnotenie výsledkov a záver.

V časti Súčasný stav riešenej problematiky autor uvádza dostupné informácie a poznatky týkajúce sa danej témy. V časti Cieľ práce je charakterizovaný predmet riešenia a jasne sformulovaný riešený problém. Súčasťou sú aj čiastkové ciele, ktoré podmieňujú dosiahnutie cieľa hlavného.

Časť Metodika práce a metódy skúmania spravidla obsahuje:

- charakteristiku objektu skúmania,
- pracovné postupy,
- spôsob získavania údajov a ich zdroje,
- použité metódy vyhodnotenia a interpretácie výsledkov.

Výsledky práce a diskusia sú najvýznamnejšími časťami záverečnej práce. Výsledky (vlastné postoje alebo vlastné riešenie vecných problémov), ku ktorým autor dospel, sa musia logicky usporiadať a pri popisovaní sa musia dostatočne zhodnotiť. Zároveň sa komentujú všetky skutočnosti a poznatky v konfrontácii s výsledkami iných autorov. Ak je to vhodné, výsledky práce a diskusia môžu tvoriť aj jednu samostatnú časť.

Odporúčaná formálna úprava záverečných prác

- Formálna úprava záverečných prác vychádza z technických noriem.
- Záverečná práca sa vypracúva spravidla v štátnom jazyku, v prvej osobe množného čísla.
- Tlačená verzia bakalárskej práce musí byť zhodná s digitálnou verziou odovzdanou do informačného systému vysokej školy (AIS) a zviazaná tak, aby sa jednotlivé listy nedali vyberať.
- Spolu s odovzdaním tlačenej verzie záverečnej práce je potrebné odovzdať aj jej digitálnu verziu na účely jej uchovania v akademickej knižnici. Táto sa odovzdáva na neprepisovateľnom nosiči informácií, najmä na CD v zmysle Metodiky MŠ VVŠ SR. Označenie CD nosiča musí obsahovať všetky náležitosti ako obal záverečnej práce.

10.1 Názov podkapitoly

Podkapitoly práce slúžia na členenie textu práce s cieľom čo najväčšej prehľadnosti.

10.1.1 Názov tretej úrovne

Editujte svoju prácu v kapitolách a podkapitolách. Čísla kapitol a podkapitol (druhej a tretej úrovne) sa citujú v texte práce takto:

... V kapitole 1 sme už uviedli, že ...; ... pozri 1.1.1 ... atď. ...

Odporúčaný rozsah bakalárskej práce je 30 až 50 strán.

Do tohto rozsahu sa počíta len hlavný text, t. j. úvod, kapitoly, záver a zoznam použitej literatúry. Dôležitejšia než rozsah práce je jej kvalita a úroveň jej spracovania. Pri písaní je dôležité dbať na vyváženosť (proporcionálnosť) jednotlivých častí práce:

- úvod má spravidla 1 – 2 strany,
- teoreticko-metodologická časť tvorí spravidla jednu tretinu práce,
- ostatné kapitoly tvoria približne dve tretiny práce,
- záver má zvyčajne 1 – 2 strany.

V časti Súčasný stav riešenej problematiky autor uvádza dostupné informácie a poznatky týkajúce sa danej témy. Zdrojom pre spracovanie sú aktuálne publikované práce domácich a zahraničných autorov. Podiel tejto časti práce má tvoriť približne 30 % práce.

11 Výsledky práce

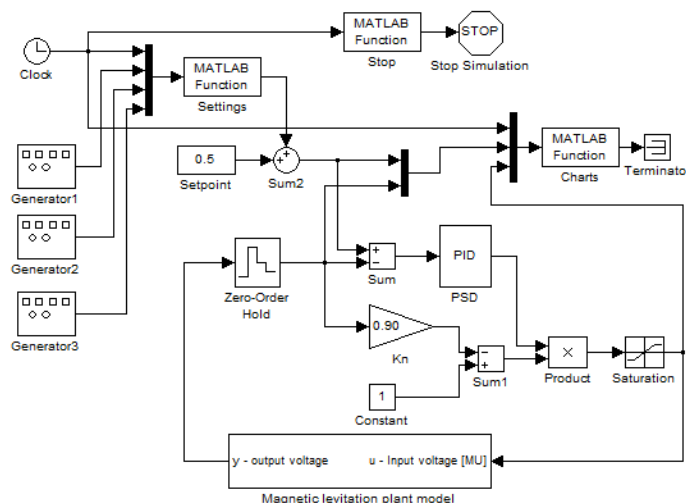
Výsledky práce a diskusia sú najvýznamnejšími časťami záverečnej práce. Výsledky, ku ktorým dospel, autor logicky usporiada a pri popisovaní ich dostatočne zhodnotí. Zároveň autor komentuje všetky skutočnosti a poznatky v konfrontácii s výsledkami iných autorov. Ak je to vhodné, výsledky práce a diskusia môžu tvoriť aj jednu samostatnú časť

11.1 Obrázky, tabuľky, rovnice, krížové odkazy

V práci sa môžu vyskytovať okrem slovného textu aj informácie vyjadrené v obrazovej forme a symbolmi.

11.1.1 Obrázky

Obrázky obsahujú grafy, diagramy, mapy, schémy a pod. Nie je potrebné rozlišovať rozličné typy ilustrácií, stačí, ak sa všetky označia ako „Obrázok“. Obrázky musia byť číslované priebežne v celej práci a každý obrázok musí mať titulok (názov). Text titulku musí byť pochopiteľný aj bez kontextu. Obrázok má byť zaradený za textom, kde sa spomína po prvýkrát (najlepšie na tej istej strane). Obrázok by mal byť podľa možnosti centrováný. Pri odkazovaní na daný obrázok v texte treba použiť odkaz na obrázok (napr. obr. 1).



Obr. 1 Simulačná schéma

11.1.2 Tabuľky

Akýkoľvek tabuľkový materiál, ktorý sa skladá z viac než štyroch alebo piatich riadkov, by mal byť spracovaný do formy tabuľky (Tab. 1). Popis a záhlavie tabuľky má byť zrozumiteľné samostatne bez odkazu na text. Záhlavia majú vyjadrovať druh veličiny a typy jednotiek vo forme „veličina/jednotka“, je potrebné používať rovnaké symboly a skratky ako v texte. Každá tabuľka musí mať poradové číslo a titulok umiestnený nad tabuľkou.

Tab. 1 "Klikni sem a napíš názov tabuľky"

	treg [min]	Σ_{\max} [%]	T_{σ} [min]
metóda umiestnenia pólov	1,31	4,48	11,0
metóda Ziegler-Nichols	1,32	2,84	1,50

11.1.3 Rovnice, vzorce

Rovnice sa uvádzajú odsadené o 0,9 cm zľava, vysvetlivky symbolov na začiatku riadku. Vysvetlivky symbolov sa uvádzajú od začiatku riadka. Čísla vzorcov uvádzame na pravom konci riadka v okrúhlych zátvorkách. Pre písanie fyzikálnych veličín a matematických premenných sa používa kurzíva. Používame sústavu jednotiek SI (ISO 31 a ISO 1001). Pri písaní rovníc používame editor rovníc.

Rovnice vkladáme v tejto šablóne takto:

$$(x+a)^n = \sum_{k=0}^n \binom{n}{k} x^k a^{n-k} \quad (1)$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (2)$$

Napr.:

Začnime rovnicou

$$(x+a)^n = \sum_{k=0}^n \binom{n}{k} x^k a^{n-k} \quad (3)$$

Grafický priebeh riešenia rovnice (2) vidíme na

11.1.4 Literatúra

Zoznam literatúry sa nachádza v sekcii Literatúra. Ide o samostatný číslovaný zoznam, pričom arabská číslica je v hranatých zátvorkách. Pre zoznam je vytvorený štýl *Literatura*. Na položky v zozname sa odkazuje číslom položky v hranatej zátvorke.

Odkaz na knihu

- [1] TIMKO, J. – SIEKEL, P. – TURŇA, J. 2004. *Geneticky modifikované organizmy*. Bratislava : Veda, 2004. 104 s. ISBN 80-224-0834-4.

Odkaz na článok v časopise:

- [2] VODODA, M. et al. An approximated solution to pendulum equation. In *International Journal of Mathematical Education in Science and Technology*. ISSN 0020-739X, 2009, vol. 40, no. 2, p. 206-215.

Odkaz na príspevok v zborníku:

- [3] ZEMAN, D. The effect of semiconductor surface. In *9th International Conference : proceedings*. Vol. 2. Fruit Growing and viticulture. Lednice: Mendel University of Agriculture and Forestry, 2001. ISBN 80-7157-524-0, p. 262-268..

Odkaz na patentové dokumenty:

- [4] Úrad priemyselného vlastníctva Slovenskej republiky: *Spôsob výroby tesnenia valivých ložísk*. Majiteľ a pôvodca patentu: Vladimír Lukáč. Slovenská republika. Patentový spis, 278399. 1997-03-05.

Odkaz na kvalifikačnú prácu:

- [5] BIELIKOVÁ, M. Príspevok k tvorbe konfigurácie softvérového systému s využitím znalostí. Bratislava: FEI STU, 1995. 136 s. Kandidátska dizertačná práca.

Odkazy na zdroje v elektronickej forme:

Pri využívaní informácií z elektronických dokumentov treba dodržiavať tieto **zásady**:

- uprednostňujeme autorizované súbory solídnych služieb a systémov,
- zaznamenáme dostatok informácií o súbore tak, aby ho bolo opäť možné
- vyhľadať,
- urobíme si kópiu použitého prameňa v elektronickej alebo papierovej forme,
- za verifikovateľnosť informácií zodpovedá autor, ktorý sa na ne odvoláva.

Na zápis elektronických dokumentov platia tie isté pravidlá ako na zápis "klasických".

Navyše treba uviesť tieto údaje:

- druh nosiča [online], [CD-Rom], [disketa], [magnetická páska]
- dátum citovania (len pre online dokumenty)
- dostupnosť (len pre online dokumenty)

Poradie prvkov odkazu je nasledovné:

Autor. Názov. In *Názov primárneho zdroja: Podnázov*. [Druh nosiča]. Editor. Vydanie alebo verzia. Miesto vydania: Vydavateľ, dátum vydania. [Dátum citovania]. Poznámky. Dostupnosť. ISBN alebo ISSN.

Zdroje z internetu – online dokumenty:

- [6] TKAČÍKOVÁ, D. Když se řekne digitální knihovna ... In *Ikaros*. [online]. 8/3 1999. [cit: 2000-06-05]. Dostupné na internete: <http://ikaros.ff.cuni.cz/ikaros/1999/c08/usti/usti_tkacikova.htm>. ISSN 1212-5075.
- [7] PASC-L (*Public Access Computer Systems Forum*). [online]. Houston : University of Houston Libraries. June 1989. [cit: 1995-05-07]. Dostupné na: <listserv@uhupvm.uh.edu>
- [8] KATUSCAK [Katuščák], D. *Marc 21*. [online]. Správe pre: diskusná skupina Akademické knižnice. 10. nov. 2001. [cit: 2001-11-11].
- [9] KOLLÁROVÁ, M. *Využívanie databáz v rámci projektu eIFL Direct*. [online]. Správa pre: Jana Krajnová. 08. Apr. 2002; 15:34:03. [cit: 2002-04-12]. <krajnova@utd.elf.stuba.sk>

Prílohy

Táto časť práce obsahuje zoznam všetkých príloh vrátane elektronických nosičov. Názvy príloh v zozname musia byť zhodné s názvami uvedenými na príslušných prílohách. Tlačené prílohy majú na prvej strane identifikačné údaje – informácie zhodné s titulnou stranou práce doplnené o názov príslušnej prílohy (Systémová príručka, Používateľská príručka). Identifikačné údaje sú aj na priložených diskoch alebo disketách. Ak je médií viac, sú označené aj číselne v tvare I/N, kde I je poradové číslo a N je celkový počet daných médií.

Každá príloha začína na novej strane a je označená samostatným písmenom (Príloha A, Príloha B...). Číslovanie strán príloh nadväzuje na číslovanie strán v hlavnom texte.

Príloha A: CD médium – práca v elektronickej podobe, prílohy v elektronickej podobe.

Príloha B: Používateľská príručka

Príloha C: Systémová príručka