Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Datalog pro aukce

Využití datalogu ve webových aplikacích

2024

Tomáš Karný

Vedoucí práce:
Mgr. Jan Laštovička, Ph.D.

Studijní program: Informatika,
Specializace: Programování a vývoj software

Bibliografické údaje

Autor:	Tomáš Karný
Název práce:	Datalog pro aukce (Využití datalogu ve webových aplikacích)
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2024
Studijní program:	Informatika, Specializace: Programování a vývoj software
Vedoucí práce:	Mgr. Jan Laštovička, Ph.D.
Počet stran:	33
Přílohy:	elektronická data v úložišti katedry informatiky
Jazyk práce:	český

Bibliographic info

Author:	Tomáš Karný
Title:	Datalog for auction (Usage of datalog in web applications)
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2024
Study program:	Computer Science, Specialization: Programming and Software Development
Supervisor:	Mgr. Jan Laštovička, Ph.D.
Page count:	33
Supplements:	electronic data in the storage of department of computer science
Thesis language:	Czech

Anotace

Cílem bakalářské práce je vytvoření webové aplikace pro demonstraci využití jazyku Datalog u online aukce. Umožňuje registrovaným uživatelům dražit předměty a omezeně prohledávat databázi zadáváním datalogových dotazů. Dražbu předmětu vyhraje ten, kdo přihodí nejvíce do uplynutí termínu zadaného prodávajícím. Po konci aukce předá aplikace prodávajícímu a výherci kontaktní údaje a informaci o určené ceně předmětu. Aplikace neposkytuje možnost přímé platby, ale pouze spojuje prodávajícího s kupujícím a stanovuje cenu předmětu.

Synopsis

The aim of the bachelor's thesis is to create a web application for demonstrating the use of the language Datalog at the online auction. Allows registered users to bid on items and perform limited database searches by entering datalog queries. The person who bids the most before the deadline specified by the seller wins the auction of the item. After the end of the auction, the application will give the seller and the winner contact details and information about the determined price of the item.The application does not provide the option of direct payment, but only connects the seller with the buyer and sets the price of the item.

Klíčová slova: webová aplikace; Datalog; aukce

Keywords: web application; Datalog; auction

Děkuji panu Mgr. Jan Laštovička, Ph.D. za odborné vedení bakalářské práce.

Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

Obsah

1 Úvod **7**

2 Obecné pojmy **8**
2.1 Aukce . . . 8
2.2 Webová aplikace . . . 8
2.3 Datalog . . . 9

3 Vlastní podmnožina datalogu **10**
3.1 Syntaxe . . . 10
3.2 Sémantika . . . 11
3.3 Kompilace . . . 13

4 Použité technologie **16**
4.1 PHP . . . 16
4.2 MVC . . . 16
4.3 HTML . . . 17
4.4 CSS . . . 18
4.5 JavaScript . . . 18
4.6 AJAX . . . 19
4.7 MySQL . . . 19

5 Programátorská příručka **20**
5.1 Struktura databáze . . . 20
5.2 Soubor index.php . . . 21
5.3 Adresář images . . . 21
5.4 Adresář login . . . 21
5.5 Uživatelské role . . . 21
5.6 Adresář Controllers . . . 22
5.7 Adresář Views . . . 22
5.8 Adresář Models . . . 22
5.9 Světlý a tmavý režim . . . 23

6 Uživatelská příručka **24**
6.1 Rozhraní aplikace . . . 24
6.2 Úvodní stránka . . . 24
6.3 Mobilní verze . . . 25
6.4 Registrace a přihlášení . . . 25
6.5 Uživatelský prostor . . . 26
6.6 Pravidla aukce . . . 27
6.7 Vyhledávání . . . 28
6.8 Návod . . . 28

7 Testování aplikace **29**

Závěr 30

Conclusions 31

A Obsah elektronických dat 32

Literatura 33

1 Úvod

V dnešní době je stále rostoucí trend online obchodování. Část tohoto trendu se týká i online aukcí, které díky internetu získaly nový rozměr a umožnily účastníkům z celého světa obchodovat s různými produkty a službami.

Cílem mé bakalářské práce je vytvořit webovou aplikaci pro online aukce, která zároveň seznámí uživatele s jazykem datalog a umožní jim vyzkoušet si vyhledávání informací z databáze za pomoci tohoto jazyka. Aplikace je zaměřena na jednoduchost a přehlednost, aby se snadno používala.

Nejdříve se seznámíme s pojmy, co je to aukce a jaké jsou její základní typy. Poté vysvětlíme, co je webová aplikace a řekneme si něco o jazyku datalog. Po seznámení se s těmito pojmy definujeme vytvořenou podmnožinu pro překlad datalogových dotazů, včetně její syntaxe a sémantiky, a popíšeme, jak probíhá překlad. Následně uvedeme, jaké jazyky a technologie jsou použity pro vytvoření práce. Dále se budeme zabývat programátorskou příručkou, která obsahuje strukturu databáze, nejdůležitější adresáře a funkcionality. Následuje uživatelská příručka. Tato příručka slouží uživatelům jako návod. Ukazuje, jakým způsobem probíhá aukce, jak takovou aukci vytvořit a jak vyhledávat pomocí jazyka Datalog. V samostatné sekci testování aplikace se zaměříme na ověření správnosti a výkonu aplikace. Budeme se soustředit na kompatibilitu práce, tedy na to, na kterých prohlížečích byla testována a jak na nich funguje. Na závěr shrneme cíle práce a zhodnotíme její přínos.

2 Obecné pojmy

2.1 Aukce

Aukce[1], známá také jako dražba, je postup prodeje, při kterém je zboží nebo služba nabízena k prodeji a potenciální kupci postupně podávají nabídky.
Je jedním z nejstarších a nejefektivnějších způsobů prodeje a nákupu. Existuje několik typů aukcí, které se liší svými pravidly a mechanismy přihazování, avšak všechny mají společný cíl - dosáhnout co nejvyšší ceny pro prodávajícího nebo co nejvýhodnější ceny pro kupujícího. Mezi nejznámější z nich patří:

Anglická aukce

Anglická aukce je nejběžnější typ aukce. Je charakteristická otevřeným a transparentním procesem přihazování. Každý účastník tedy může vidět aktuální nejvyšší nabídku a přihodit vyšší částku. Aukci vyhrává účastník s nejvyšší nabídkou.

Holandská aukce

Holandské aukce fungující opačně než Anglická aukce. Aukce začíná s vysokou vyvolávací cenou, která se postupně snižuje, dokud některý z účastníků nepřijme aktuální cenu. Tento první účastník se stává vítězem.

Obálková aukce

V uzavřené obálkové aukci podávají účastníci své nabídky tajně, obvykle v zapečetěných obálkách. Každý účastník podá pouze jednu nabídku, aniž by věděl, jaké nabídky podali ostatní. Po uplynutí lhůty se všechny nabídky otevřou najednou a vítězem se stává účastník s nejvyšší nabídkou.

2.2 Webová aplikace

Webová aplikace[2, 3], také nazývaná web app, je typ aplikačního softwaru, který je umístěn na vzdáleném serveru a ke kterému uživatelé přistupují pomocí webového prohlížeče. Na rozdíl od tradičních desktopových aplikací, které se spouštějí přímo na uživatelově zařízení, webové aplikace běží na serveru a komunikují s klienty (uživateli) přes síť, často pomocí protokolu HTTP. Tento model umožňuje snadnější aktualizace aplikace a poskytuje přístup k aplikaci z různých zařízení , aniž by bylo nutné instalovat specifický software na každém zařízení.

V oblasti mobilních počítačů se webové aplikace často porovnávají s nativními aplikacemi, které jsou vyvíjeny specificky pro konkrétní platformy nebo zařízení a instalují se přímo na zařízení. Nativní aplikace využívají hardware specifický pro dané zařízení, jako je například GPS nebo fotoaparát.

Existuje také kategorie aplikací nazývaná hybridní aplikace, které kombinují prvky obou přístupů. Hybridní aplikace fungují podobně jako webové aplikace

, ale instalují se jako nativní aplikace do zařízení. Tyto aplikace mohou využívat interní rozhraní API pro přístup k specifickým funkcím zařízení. Na rozdíl od nativních aplikací však hybridní aplikace obvykle nemohou fungovat offline.

Navigační prvky hybridních aplikací často sdílí podobnosti s webovými aplikacemi, protože jejich základní struktura je často odvozena od webového prostředí. Tento hybridní přístup umožňuje vývojářům kombinovat výhody webových aplikací, jako je snadná aktualizace a široká dostupnost, s možnostmi nativních aplikací využívajících specifický hardware a offline funkcionalitu.

2.3 Datalog

Datalog[4, 5] je deklarativní programovací jazyk používaný pro dotazování a manipulaci s daty v databázích a logických programech. Jeho historie sahá do počátku 70. let 20. století, kdy byl vyvinut v rámci výzkumu v oblasti databázových systémů a umělé inteligence. Datalog je založen na logickém programovacím jazyce Prolog a v průběhu let prošel mnoha vylepšeními a rozšířeními.

Jazyk Datalog se skládá z tří základních prvků a to:

- **Fakta**, což jsou tvrzení, která v daném kontextu jsou považována za pravdivá.
- **Pravidla** definující, jak jsou nová fakta generována z existujících faktů a pravidel.
- **Dotazy** používající se k získání informací z pravidel a faktů.

Datalog stal důležitým nástrojem v oblastech, jako jsou databázové systémy , umělá inteligence a analýza dat.

V databázových systémech se Datalog často používá k definování a získávání relačních dat. Datalog se stal oblíbeným nástrojem pro implementaci různých typů databázových systémů, včetně relačních databází, grafových databází a systémů pro správu znalostí.

V oblasti umělé inteligence se Datalog často používá pro logické programování a reprezentaci znalostí. Jeho schopnost vyjadřovat logické vztahy mezi entitami a vyvozovat závěry na základě definovaných pravidel z něj činí důležitý nástroj pro budování expertních systémů, automatizaci úloh a analýzu dat.

3 Vlastní podmnožina datalogu

V této kapitole se zaměříme na naši specifickou podmnožinu datalogových programů, která umožňuje překlad pravidel a dotazů do jim odpovídajících SQL dotazů. Nejprve si definujeme základní pojmy, které budeme potřebovat k definici složitějších pojmů. Poté se podíváme na syntaxi predikátu a datalogového programu. Následně se zaměříme na sémantiku, tedy na význam a interpretaci predikátů a datalogových programů v kontextu databázového dotazování. Nakonec představíme krok za krokem postup, jakým lze tyto datalogové programy konvertovat do SQL dotazů.

3.1 Syntaxe

Konstanta je pevná hodnota, která se nemění během provádění programu nebo výpočtu. Může se jednat o číslo, řetězec či jiný typ hodnoty. Konstanty se zapisují přímo, bez použití uvozovek či jiných speciálních znaků.

Proměnná je pojmenovaná paměťová jednotka, která může během provádění programu nebo výpočtu měnit svou hodnotu. V naší knihovně platí konvence , že názvy proměnných musí obsahovat otazník jako předponu. Např.

```
?cena, ?velikost, ?id, ...
```

Predikát

Predikát se skládá z názvu a n-tice argumentů. Lze jej napsat jako:

$$\texttt{název(argument}_1\texttt{, argument}_1\texttt{, ..., argument}_N\texttt{)}$$

kde `název` je název predikátu označující konkrétní relaci a $\texttt{argument}_1$, $\texttt{argument}_2$, ..., $\texttt{argument}_N$ může být konkrétní hodnota (konstanta) nebo proměnná. Příkladem predikátu může být:

```
cenik(Auto, ?cena)
```

kde `cenik` je název predikátu obsahující argumenty `Auto` a `cena`, kdy `Auto` je konstanta a `cena` je proměnná.

Datalogový dotaz

Datalogový dotaz je řetězec ve tvaru:

```
hlava :- tělo
```

Pravou část dotazu, zvanou tělo dotazu, tvoří n-tice predikátu oddělených čárkou. Levou část dotazu, zvanou hlava dotazu, tvoří n-tice proměnných oddělených čárkou. Levou a pravou část dotazu od sebe odděluje symbol „:-“. Každý datalogový dotaz musí splňovat podmínku, že každá proměnná nacházející se v hlavě dotazu se musí nacházet i v těle dotazu.

Například u dotazu:

```
?nazev :- kniha(?nazev, ?autor)
```

je hlava tvořena proměnnou `?nazev` a tělo je tvořeno predikátem `kniha(?nazev, ?autor)`.

Datalogové pravidlo

Datalogové pravidlo je řetězec ve tvaru:

```
hlava :- tělo
```

Pravou část pravidla, zvanou tělo pravidla, tvoří n-tice predikátu oddělených čárkou. Levou část dotazu, zvanou hlava pravidla, tvoří právě jeden predikát. Levou a pravou část dotazu od sebe odděluje symbol „:-“. Každé datalogové pravidlo musí splňovat podmínku, že každá proměnná nacházející se v hlavě pravidla se musí nacházet i v těle pravidla. Rozdílem mezi syntaxí datalogového pravidla a dotazu je tedy tvar hlavy. Například u pravidla

```
ceny(?cena) :- cenik(?nazev, ?cena)
```

je hlava tvořena predikátem `ceny(?cena)` a tělo je tvořeno predikátem `cenik(?nazev, ?cena)`.

Syntaktický cukr

Knihovna umožňuje, aby datalogový dotaz bylo možné zapsat i zkráceným zápisem. Je možnost absence hlavy za předpokladu, že chceme získat všechna data splňující splňují podmínky stanovené v jeho těle. Není tedy rozdíl mezi prvním a druhý zápisem, které oba získávají název a cenu předmětu z ceníku.

```
1. :- cenik(?nazev, ?cena)
```

```
2. ?nazev, ?cena :- cenik(?nazev, ?cena)
```

Dále můžeme využít absenci argumentů a zadat místo nich pouze znak „_“ , který bude fungovat jako anonymní proměnná, která se nevyskytuje nikde v těle. Příkladem může být, kdybychom chtěli pouze ceny z ceníku a nezajímal nás název, tak bychom dotaz mohli zapsat takto:

```
?cena :- cenik(_, ?cena)
```

3.2 Sémantika

Sémantika datalogových dotazů určuje způsob, jakým jsou tyto dotazy vyhodnocovány. Datalogové dotazy a pravidla se vyhodnocují pomocí logických pravidel a predikátů.

Predikát

Predikát popisuje vztah mezi svými argumenty. Predikát je pravdivý, pokud je pravdivý pro všechny své argumenty. Každý z argumentů musí být na stejném místě v predikátu, na kterém se nachází v tabulce či pravidle.
Příkladem predikátu může být:

```
ceník(?název, 5)
```

Tento predikát popisuje vztah `ceník` a obsahuje dva argumenty:

- Prvním argumentem je proměnná `?název`, která může zastupovat libovolný název položky v ceníku.
- Druhým argumentem je hodnota (konstanta) `5`, která představuje konkrétní cenu dané položky.

Datalogový dotaz

Datalogový dotaz je otázka nebo požadavek na databázi, jehož cílem je nalézt hodnoty proměnných v hlavě dotazu, které odpovídají relacím v těle dotazu. Tělo dotazu se skládá z predikátů a je pravdivé, pokud jsou všechny predikáty v těle dotazu pravdivé současně. Hlava dotazu tedy specifikuje, které informace , splňující podmínky v těle dotazu, chceme získat. Znak „:-“ zde zastupuje logickou implikaci. Příklad dotazu:

```
?pozice :- zaměstnanec(Radek, Malý, ?pozice)
```

Tento dotaz se ptá, jakou pozici (`?pozice`) zastává osoba jménem Radek Malý.

Pravidlo

Pravidlo je logické tvrzení, které popisuje, jaká relace platí v závislosti na jiných relacích nebo vlastnostech. Pravidlo je vyjádřeno ve formě implikace, kde hlavním predikátem je konkrétní vztah nacházející se v hlavě, který chceme vyjádřit , a podmínky jsou vyjádřeny pomocí jiných predikátů a jejich argumentů v těle pravidla. Pro lepší představu uvedeme příklad pravidla:

```
analytik(?jméno, ?příjmení) :-
zaměstnanec(?jméno, ?příjmení, analytik)
```

kde `analytik` je název pracovní pozice. Toto pravidlo nám příká, že osoba označená proměnnými `?jméno` a `?příjmení` je `analytik`, pokud existuje záznam v predikátu `zaměstnanec`, kde daná osoba má pracovní pozici „analytik“.

3.3 Kompilace

Proces překladu datalogového dotazu a pravidla do SQL dotazu lze rozdělit do několika kroků. Jednotlivé části překladu si ukážeme na následujícím příkladu dotazu:

```
?pozice :- zaměstnanec(Martin, Novák, ?pozice)
```

ptající se na pozici zaměstnance Martina Nováka a pravidle Tento dotaz se ptá na pozici zaměstnance jménem Martin Novák.

```
analytik(?jméno, ?příjmení) :-
zaměstnanec(?jméno, ?příjmení, analytik)
```

definující, za jakých podmínek je zaměstnanec označen, jako analytik.

1. **Identifikace dotazu:** Vstupní řetězec se rozdělí na hlavičku a tělo za pomoci znaku „:-“. Dle tvaru hlavičky se určí o jaký typ dotazu se jedná. Tedy u příkladu dotazu je hlava tvořena proměnnou `?pozice` a tělo predikátem `zaměstnanec(Martin, Novák, ?pozice)`. Hlavička obsahuje proměnnou a program identifikuje vstupní řetězec jako datalogový dotaz. Dále probíhá identifikace tabulek, které budou použity při získávání dat. V obou příkladech je nalezenou tabulkou tabulka `zaměstnanec`. Vytvoří se slovník pro přiřazení aliasů k původním názvům tabulek, což umužňuje provádět spojení v těle dotazu. V dotazu jsou následně pomocí slovníku nahrazeny původní názvy tabulek pro lepší manipulaci a tvorbu dotazu. Slovník bude obsahovat alias `table0` překládající se na název tabulky `zaměstnanec` a řetězec pro dotaz po nahrazení bude vypadat takto:

   ```
   ?pozice :- table0(Martin, Novák, ?pozice)
   ```

2. **Překlad hlavy pravidla:** Hlavu pravidla rozdělíme na název pravidla a pole argumentů pomocí regulárního výrazu. V našem příkladu název pravidla je `analytik` a jeho argumenty `?jméno`,`?příjmení` a `analytik`.

   Pro každý z argumentů se analyzují jednotlivé predikáty v těle dotazu a zjistí se jejich přítomnost v některém z predikátů. Název tabulky pro každý argument se získá z prvního predikátu, který obsahuje daný argument. Poté jsou názvy pravidla a všechny dvojice `název_tabulky.argument` spojeny do první části řetězce, který bude sloužit k vytvoření view v SQL. Tato část řetězce bude u našeho příkladu vypadat následovně:

   ```
   CREATE OR REPLACE VIEW analytik AS SELECT DISTINCT
   zaměstnanec.jméno AS '?jméno', zaměstnanec.příjmení
                        AS '?příjmení'
   ```

3. **Překlad hlavy dotazu:** Překlad hlavy dotazu je podobný, jako u pravidla, ale místo predikátu pracujeme s proměnnými. V našem příkladu hlavu dotazu tvoří proměnná `?pozice`. V případě prázdnosti hlavy jsou vybrány všechny sloupce z těla dotazu. Ke každé proměnné, podobně jako u pravidla, nalezneme název tabulky a vytvoříme dvojice název_tabulky.proměnná a ty jsou spojeny do první části řetězce, který bude sloužit k vytvoření SELECT klauzule v SQL. Ta bude u našeho příkladu vypadat takto:

```
SELECT DISTINCT zaměstnanec.pozice AS '?pozice'
```

4. **Překlad těla:** Překlad těla pravidla i dotazu probíhá stejným způsobem. Tělo dotazu rozdělíme na jednotlivé predikáty a pro každý z nich zjistíme název tabulky pomocí regulárního výrazu, který následně vložíme do pole. U obou příkladů je tělo tvořeno predikátem s názvem `zaměstnanec`. Získané pole názvů tabulek slouží k vytvoření FROM klauzule v SQL. Každý predikát se dále zpracovává, aby se zjistilo, zda obsahuje konstanty , které jsou vloženy do WHERE klauzule. U příkladu dotazu jsou konstanty `Martin` a `Novák` a proměnnou je `analytik`. Predikáty dále mohou také obsahovat proměnné, které se používají ke spojení tabulek mezi sebou , což se zjišťuje výskytem stejných proměnných v různých predikátech těla. Jako výsledek překladu těla dotazu dostáváme řetězec:

```
FROM zaměstnanec AS 'table0'
WHERE table0.jmeno = 'Martin'
AND table0.prijmeni = 'Novák';
```

5. **Spojení výsledků překladu:** Nakonec vezmeme řetězec vytvořený překladem hlavy a řetězec vytvořený překladem těla a spojíme. Spojením těchto dvou řetězců nám vznikne kompletní SQL dotaz. U příkladu pro dotaz tedy:

```
SELECT DISTINCT zaměstnanec.pozice AS '?pozice'
        FROM zaměstnanec AS 'table0'
        WHERE table0.jmeno = 'Martin'
        AND table0.prijmeni = 'Novák';
```

   A u příkladu pro pravidlo:

```
CREATE OR REPLACE VIEW analytik AS SELECT DISTINCT
zaměstnanec.jméno AS '?jméno', zaměstnanec.příjmení
    AS '?příjmení' FROM zaměstnanec AS 'table0'
        WHERE table0.pozice = 'analytik';
```

Výběr typu překladu: Knihovna obsahuje 2 funkce pro překlad dotazů:

- Funkce ‘translateDatalogToSQL‘ umožňující překlad libovolných datalogových dotazů i pravidel do SQL.
- Funkce ‘limitedTranslate‘ mající omezený přístup k tabulkám a může přeložit pouze datalogové dotazy.

U obou funkcí probíhá na začátku překladu základní validace vstupu.

4 Použité technologie

4.1 PHP

PHP (Hypertext Preprocessor)[6] je skriptovací jazyk zaměřený na vývoj webu , který byl vytvořen dánsko-kanadským programátorem Rasmusem Lerdorfem. PHP je od svého počátku navrženo jako jazyk pro tvorbu dynamických webových stránek a webových aplikací. PHP je zpracováváno na serveru a výsledný výstup je zasílán uživateli ve formě HTML.

PHP podporuje širokou škálu databázových systémů, včetně MySQL , PostgreSQL a SQLite, což usnadňuje tvorbu databázových aplikací. PHP lze také využít k vývoji webových služeb, správě cookies a sessions, a práci se soubory.

Syntaxe PHP je podobná syntaxi jazyků C a Perl, což zjednodušuje učení se a používání tohoto jazyka a samotný jazyk PHP je primárně implementován v jazyku C. PHP je case sensitive, což znamená, že rozlišuje velká a malá písmena. Proměnné v PHP začínají znakem $, což je odlišuje od ostatních jazyků.

Daný jazyk je tedy charakteristický tím, že:

- Je vhodný pro vývoj dynamických webových stránek a aplikací.
- Podporuje mnoho databázových systémů.
- Kód je prováděn na serveru, což zvyšuje bezpečnost aplikace.
- Jeho syntaxe je jednoduchá a podobná C a Perl.
- Podporuje objektově orientované programování.
- PHP má automatickou správu paměti díky vestavěnému garbage collectoru , který zajišťuje efektivní uvolňování nepoužívané paměti.
- Podporuje zpracování chyb pomocí výjimek.

4.2 MVC

Model-View-Controller (MVC)[7] je návrhový vzor používaný při vývoji softwaru, který odděluje aplikaci do tří hlavních logických komponent: Model, View a Controller. Tento vzor byl poprvé představen v 70. letech ve společnosti Xerox PARC a od té doby se stal jedním z nejpoužívanějších návrhových vzorů , zejména v oblasti webového vývoje.

MVC je navrženo tak, aby oddělilo interní reprezentaci informací od způsobu, jakým jsou tyto informace prezentovány uživateli, což zvyšuje modularitu a usnadňuje správu a údržbu kódu.

Komponenty MVC:

- **Model**: Představuje data a obchodní logiku aplikace, která určuje, jak jsou data zpracovávána a jaké operace lze na nich provádět. Mezi jeho hlavní úkoly patří komunikace s databází, provádění validace a zpracování dat. Model je nezávislý na uživatelském rozhraní, což znamená, že změny v uživatelském rozhraní neovlivňují data a logiku.
- **View**: View neboli pohled je odpovědný za prezentaci dat uživateli. Generuje uživatelské rozhraní na základě dat poskytovaných modelem. View neobsahuje žádnou obchodní logiku ani zpracování dat a jeho úkolem je pouze zobrazit data uživateli.
- **Controller**: Controller funguje jako rozhraní mezi vrstvami View a Model. Přijímá uživatelské vstupy, zpracovává je a aktualizuje model nebo view podle potřeby. Controller je někdy považován za mozek aplikace , který udržuje vše v pohybu a synchronizaci.

Charakteristika MVC:

- Odděluje obchodní logiku, uživatelské rozhraní a zpracování uživatelských vstupů, což zvyšuje modularitu a usnadňuje údržbu kódu.
- Umožňuje vývojářům pracovat paralelně na různých komponentách aplikace, což zrychluje vývojový proces.
- Usnadňuje testování aplikace díky jasně definovaným rozhraním mezi jednotlivými komponentami.
- Podporuje opakovanou použitelnost kódu díky odděleným komponentám.
- Může být implementován v různých programovacích jazycích, jako jsou Java, C#, PHP, Python a další.

4.3 HTML

HTML (HyperText Markup Language)[8] je základní stavební kámen každé webové stránky. Jeho hlavní účel spočívá v definici struktury a obsahu dokumentu pomocí speciálních značek nebo tagů. Tyto tagy určují, jakým způsobem jsou různé části obsahu organizovány a jak mají být zobrazeny pro uživatele. HTML je statický jazyk, což znamená, že definuje pouze statický obsah stránky, který zůstává neměnný, dokud není změněn na serveru.

Každá webová stránka začíná základní kostrou v HTML, která obvykle zahrnuje hlavičku (head) a tělo (body). V hlavičce jsou umístěny meta informace , odkazy na externí zdroje (jako jsou CSS styly, JavaScriptové soubory, a další) a další metadata, jako je například název stránky. Tělo dokumentu pak obsahuje samotný viditelný obsah stránky, jako jsou texty, obrázky, odkazy a formuláře.

HTML je nezávislý na platformě a zobrazovacím zařízení, což znamená , že stránky napsané v HTML mohou být správně zobrazeny na různých zařízeních od počítačů až po mobilní telefony, pokud je dodržena správná responsivní tvorba. HTML5, nejnovější verze HTML, přinesla řadu pokročilých funkcí pro podporu multimédií, offline aplikací, grafických a matematických efektů a zlepšeného přístupu pro programátory.

4.4 CSS

CSS (Cascading Style Sheets)[9, 10] je jazyk, který se používá k definici vzhledu a formátování webových stránek napsaných v HTML. Zatímco HTML určuje strukturu a obsah stránky, CSS řídí prezentaci těchto prvků, jako jsou barvy , fonty, rozložení a animace. To umožňuje vývojářům a designérům měnit vzhled celého webu pomocí jediného souboru CSS, aniž by měnili samotný obsah HTML.

CSS funguje na principu kaskádového stylu, což znamená, že jednotlivé stylovací pravidla mohou být nadefinovány různými způsoby a aplikovány na stejné HTML elementy. Tím se zajišťuje, že rozdílné styly nebo efekty mohou být aplikovány na jednotlivé části stránky podle potřeby. Například, barva pozadí, velikost písma a zarovnání textu jsou běžně upravovány pomocí CSS.

Existuje mnoho metod, jak aplikovat CSS do HTML dokumentů, včetně vnořeného stylu, vnitřního stylu a externího stylu. Vnořený a vnitřní styl jsou obvykle aplikovány přímo do HTML dokumentu, zatímco externí styl je uložen v samostatném CSS souboru, který je poté odkazován z HTML dokumentu.

CSS3, nejnovější verze CSS, přináší pokročilé možnosti jako jsou gradienty , stíny, transformace, přechody a animace, které umožňují vývojářům vytvářet bohatší a interaktivnější uživatelské rozhraní pro moderní webové aplikace a stránky.

4.5 JavaScript

JavaScript[11] je všestranný skriptovací jazyk používaný pro tvorbu interaktivních a dynamických webových stránek. Vznikl v roce 1995 ve společnosti Netscape a stal se jedním z nejpoužívanějších jazyků na webu. Umožňuje vývojářům přidávat do stránek funkce reagující na uživatelské akce a manipulovat s HTML a CSS.

JavaScript je interpretovaný jazyk, což znamená, že kód je prováděn přímo ve webovém prohlížeči, neboli tedy na straně klienta. Je nezávislý na platformě , takže může běžet na jakémkoli zařízení nebo operačním systému s moderním webovým prohlížečem. Podporuje objektově orientované, imperativní a funkcionální programovací paradigmata, což umožňuje velkou flexibilitu při vývoji aplikací.

Díky schopnosti pracovat s DOM (Document Object Model) může JavaScript dynamicky měnit obsah a strukturu webových stránek. Jeho popularita stoupá díky javascriptovým frameworkům, jako například React.

4.6 AJAX

AJAX[12] (Asynchronous JavaScript and XML) je technologie umožňující dynamickou komunikaci mezi webovým klientem a serverem bez nutnosti úplného obnovení webové stránky. Na rozdíl od tradičních synchronních požadavků, které vyžadují načtení nové stránky po každé interakci, AJAX umožňuje asynchronní přenos dat, což znamená, že pouze specifická část stránky je aktualizována v reálném čase.

Tento přístup výrazně zvyšuje plynulost a efektivitu webových aplikací. AJAX k odesílání HTTP požadavků na server a zpracování odpovědí používá objekt XMLHttpRequest nebo modernější Fetch API.

Data získaná pomocí AJAXu mohou být ve formátu XML, JSON, HTML nebo prostého textu, což poskytuje velkou flexibilitu při práci s různými typy dat. AJAX je často používán v kombinaci s JavaScriptem, HTML a CSS pro tvorbu pokročilých webových aplikací, jako jsou interaktivní formuláře a real-time aktualizace dat.

4.7 MySQL

MySQL[13] je populární relační databázový systém. Byl vyvinutý společností MySQL AB a nyní vlastněný Oracle Corporation. MySQL je známý svou vysokou výkonností, spolehlivostí a snadnou použitelností.

MySQL používá strukturovaný dotazovací jazyk (SQL) pro správu databází , což umožňuje efektivní vytváření, čtení, aktualizaci a mazání dat. Podporuje různé typy dat a nabízí funkce jako transakce, referenční integritu, indexování a uložené procedury.

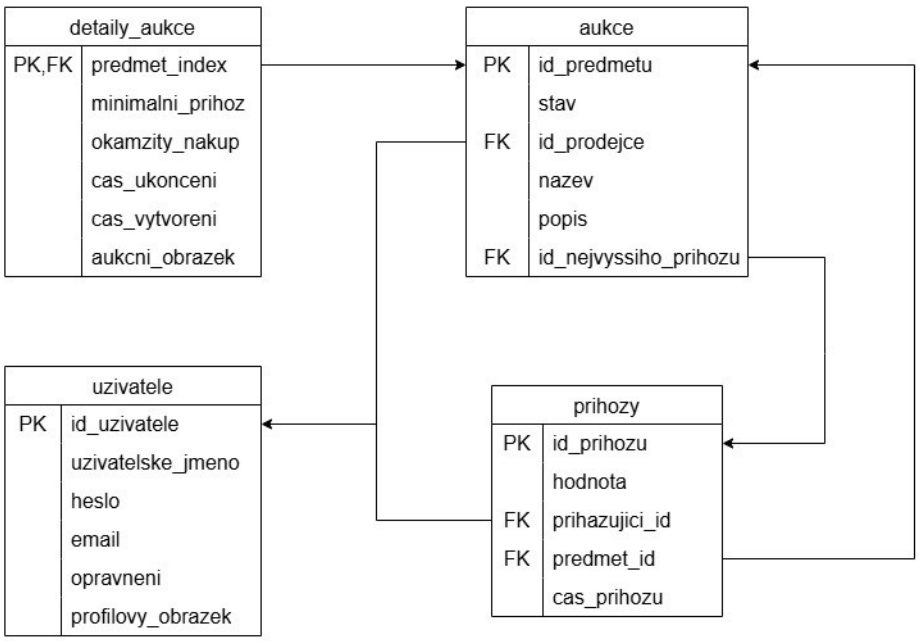
MySQL je multiplatformní, běží na Windows, Linuxu a macOS, a podporuje různé úložné enginy jako InnoDB a MyISAM pro optimalizaci výkonu.

5 Programátorská příručka

V této části se budeme zabývat technickými částmi aplikace. Ukážeme si strukturu databáze a její tabulky, poté strukturu aplikace a hlavní funkcionalitu aplikace.

5.1 Struktura databáze

Databázi tvoří celkem čtyři tabulky. Pro uložení informací o uživateli slouží tabulka `uzivatele`. Další důležitou tabulkou je `aukce`, kde ukládáme základní informace o aukci. Dále tabulka `detaily_aukce` pro uložení více detailů o aukci, jejíž záznamy jsou tvořeny zároveň s záznamy v tabulce aukce. Poslední tabulkou je tabulka `prihozy`, která slouží pro ukládání informací o příhozech.



Obrázek 1: Schéma databáze

5.2 Soubor index.php

Jedná se o hlavní soubor celé aplikace. Prvním jeho úkolem je získat aktuální adresu, na které se uživatel nachází, a následně ji rozdělit na část controller , method a params. Díky těmto částem pak vybere, které soubory mají být vybrány a z nich poskládá výslednou stránku.

5.3 Adresář images

Tato složka slouží k uchovávání všech obrázků používaných v aplikaci. Obsahuje obrázek pro domovskou stránku a 3 podsložky:

- **icons**: Uložení všech lokálních ikonek.
- **aukcni_obrazky**: Obrázky všech předmětů v aukci, jejiž název je změnen na id daného předmětu pro lepší identifikaci.
- **profilove_obrazky**: Profilové obrázky všech uživatelů aplikace, jejiž název je změnen na id daného uživatele.

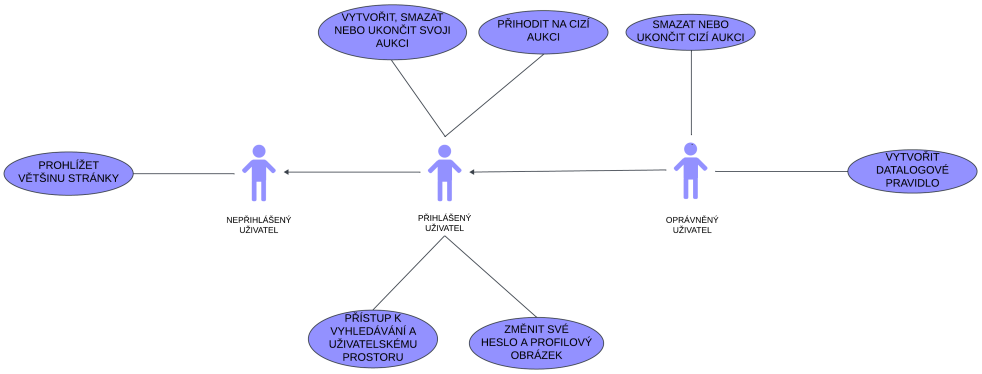
5.4 Adresář login

V této složce se nacházejí PHP skripty potřebné k implementaci přihlášení. Adresář obsahuje metody pro registraci, přihlášení a odhlášení. Kromě toho zahrnuje skript, který ověřuje existenci uživatele na základě zadaných přihlašovacích nebo registračních údajů. Dále zde naleznete soubor, který se stará o vykreslení formulářů pro přihlášení a registraci. Hesla jsou při ukládání do databáze hashována. Informace o přihlášeném uživateli jsou uloženy v proměnné `$_SESSION` , což umožňuje přístup k těmto datům během celé relace. Tato data jsou dostupná na všech stránkách webu, dokud relace neskončí.

5.5 Uživatelské role

Webová aplikace má tři typy uživatelů. Prvním typem je nepřihlášený uživatel , který má možnost si stránku a většinu její částí pouze prohlížet. Druhým typem je přihlášený uživatel, což je běžný uživatel stránky, který se může zapojit do aukcí, vytvářet je a má přístup k vyhledávání. Třetím typem je oprávněný uživatel neboli admin. Systém rozpoznává při přihlášení oprávněného a obyčejného uživatele pomocí hodnoty uložené v tabulce uživatelů ve sloupci `oprávnění`. Kladná hodnota znamená, že daný uživatel je administrátorem. Nastavit tuto hodnotu uživateli může pouze správce webové stránky přímo v databázi.

Diagram případů užití nám pomůže lépe porozumět oprávněním a rozlišení typů uživatelů:



Obrázek 2: Diagram případů užití

5.6 Adresář Controllers

- **homepage**: Výchozí kontroler aplikace.
- **aktivni_aukce**: Stará se o zobrazení všech aukcí i detailů aukce. V případě zobrazení detailů získává id aukce a převádí čas ukončení aukce do uživatelsky přívětivějšího formátu.
- **vyhledavani**: Získává strukturu databáze a zobrazuje vyhledávací stránku.
- **navod**: Zobrazuje návod a získává strukturu databáze.
- **uzivatel**: Zajišťuje zobrazení uživatelského prostoru.

5.7 Adresář Views

Zde jsou uloženy views(pohledy) webové aplikace. Všechny pohledy jsou rozděleny do vlastních složek dle příslušnosti k dané části stránky, kdy složka nese stejný název, jako kontroler. Scripty těchto pohledů se nachází v adresáři scripts.

5.8 Adresář Models

Obsahuje logiku webové aplikace pro zpracování dat. Tvoří ji 3 soubory s metodami, které se používají v různých částech aplikace a složky aktivni_aukce , uzivatel a vyhledávání, pojmenované dle svého kontorleru, sloužící pro zpracování dat v dané části aplikace.

- **Adresář aktivni_aukce**: Stará se o vytáření dražených položek, aktualizaci aukcí dle času jejich ukončení, zisk dat o aukci, podávání nabídek , mazání a ukončení aukcí.
- **Adresář uzivatel**: Zajišťuje aktualizaci profilového obrázku a hesla, zisk dat a tvorbu komponent pro ukončené a moje aukce, vytvoření nové aukce spolu s uložením obrázku předmětu.
- **Adresář vyhledavani**: Stará se o zpracování řetězce předaného z vyhledávání, tedy překlad řetězce pomocí přítomné knihovny a vytvoření komponent pro stránku dle získaných výsledků.
- **Soubor auctionFunctions**: Poskytuje funkce používané v různých částech aplikace. Cílem funkcí je tvorba tabulek na základě předaných dat , tvorba stránkovacích tlačítek apod.
- **Soubor databaseStructure**: Cílem souboru je získat aktuální schéma databáze a s jeho pomocí vytvoří tabulku pro zobrazení schéma uživateli na stránce.
- **Soubor db_con**: Tento soubor zajišťuje připojení k MySQL databázi. Nejprve definuje potřebné parametry připojení, jako jsou hostitel, uživatelské jméno, heslo a název databáze. Poté se pokusí navázat spojení pomocí třídy mysqli. Pokud připojení selže, zobrazí chybovou zprávu a ukončí skript.

5.9 Světlý a tmavý režim

Většina webových aplikací dnes nabízí možnost přepínání mezi světlým a tmavým režimem. Pro správnou funkčnost je nutné v souboru stylů používat CSS proměnné (variables) při nastavování barev prvků. Když uživatel přepne režim , přidá se nebo odebere třída `dark_theme` elementu body, čímž se změní hodnoty těchto proměnných.

6 Uživatelská příručka

Tato kapitola slouží jako návod pro uživatele, kteří s aplikací nemají žádné zkušenosti a projdeme si postupně jednotlivé části webové aplikace.

6.1 Rozhraní aplikace

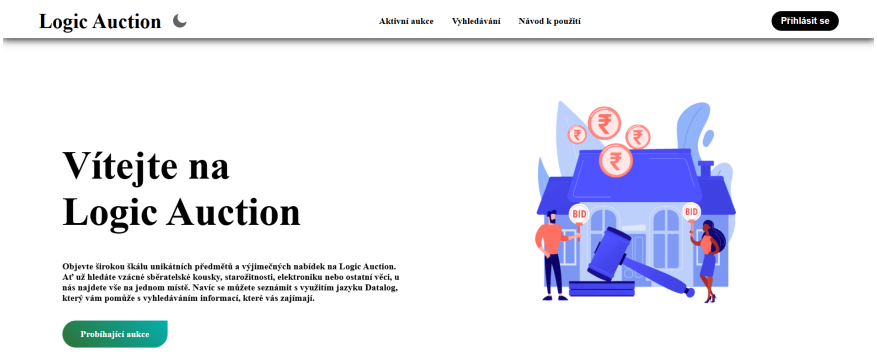
Rozhraní aplikace je navrženo tak, aby bylo jednoduché a přehledné. Skládá se ze dvou částí: horní lišty a hlavního obsahu. První část lišty obsahuje logo aplikace , které zároveň slouží jako odkaz na úvodní stránku, a tlačítko pro přepínání mezi tmavým a světlým režimem. Druhá část lišty obsahuje navigaci, pomocí které může uživatel přecházet mezi aktivními aukcemi, vyhledáváním a návodem. Poslední část lišty se týká uživatele. Pokud není uživatel přihlášen, nachází se zde tlačítko pro přihlášení. V opačném případě je zde ikonka profilového obrázku uživatele. Kliknutím na tuto ikonu se zobrazí možnosti odhlášení nebo přechodu do uživatelské oblasti.

Druhou a nejdůležitější částí rozhraní je samotný obsah. Ten se mění podle toho, na které stránce se uživatel aktuálně nachází.

6.2 Úvodní stránka

Úvodní stránka je rozdělena na poloviny. Na levé polovině uživatel nalezne informaci o tom, na které webové stránce se právě nachází. Pod uvítacím textem je popis, který zve návštěvníky k vyzkoušení této aplikace pro dražbu, která je unikátní díky možnosti vyzkoušet vyhledávání za pomoci jazyku Datalog.
Pod tímto textem se dále nachází tlačítko, pomocí kterého se návštěvník dostane na část stránky obsahující aukce.

Pravou polovinu stránky tvoří ilustrační obrázek, aby bylo návštěvníkovi hned jasné, že se jedná o stránku pro online aukce.



Obrázek 3: Úvodní stránka webové aplikace

6.3 Mobilní verze

V dnešní době je velmi důležité, aby se webová stránka správně zobrazovala i na mobilních zařízeních, protože pro mnoho uživatelů je pohodlnější využívat právě tato zařízení.

Rozdíly mezi desktopovou a mobilní verzí se nacházejí zejména v horní liště.

- Hlavním rozdílem je navigace, která je v mobilní verzi řešena pomocí hamburger menu. Po kliknutí na hamburger menu se zobrazí navigace, a pokud je uživatel nepřihlášený, také tlačítko pro přihlášení.
- Další rozdíl je viditelný po přihlášení uživatele. Pořadí hamburger menu a ikonky profilového obrázku uživatele je zde prohozeno. Toto uspořádání zajišťuje větší přehlednost a lepší vzhled.

Obrázek 4: Webová aplikace na mobilním zařízení

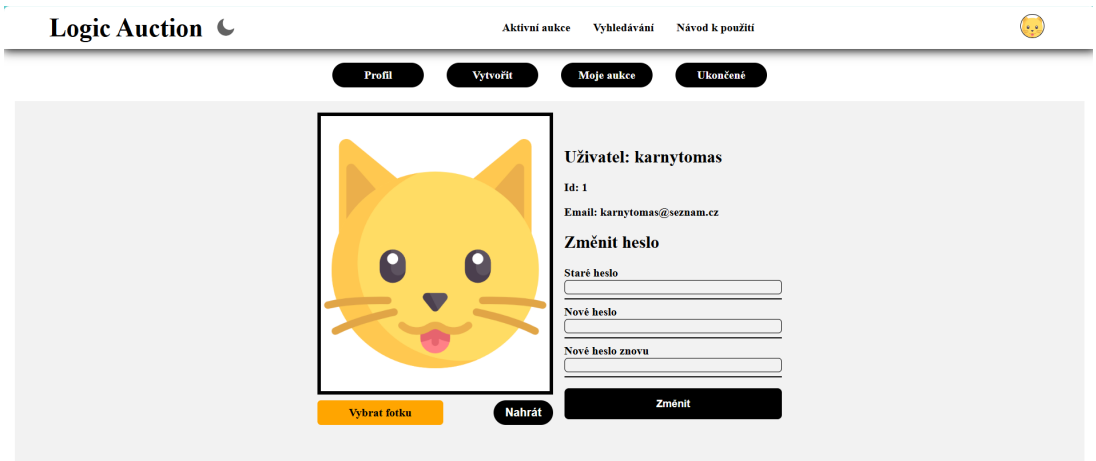
6.4 Registrace a přihlášení

V horní liště stránky se nachází tlačítko „přihlásit se“. Po kliknutí na toto tlačítko se zobrazí formulář s přihlášením. Přihlášení probíhá za pomoci uživatelského jména a hesla. Pokud uživatel není registrován, je ve spodní části formuláře možnost přepnutí na registrační formulář. Při registraci je nutné zadat unikátní uživatelské jméno a email spolu s heslem.

6.5 Uživatelský prostor

Pro přihlášení nebo registraci je uživatel přesměrován do uživatelského prostoru. Uživatelský prostor obsahuje vlastní navigaci a dělí se do čtyř částí:

- **Profil** V této části jsou uvedeny detaily o uživateli. Zobrazuje se zde ID uživatele, jméno, email a profilový obrázek. Je zde také možnost změnit heslo nebo profilový obrázek.
- **Vytvořit** Zde má uživatel možnost vytvořit vlastní aukci. Pro její vytvoření je nutné zadat všechny potřebné údaje o aukci, a to: název a popis předmětu, fotku předmětu, nastavení minimálního příhozu a částky pro okamžitý nákup, a nakonec termín, kdy má být aukce ukončena. U některých z uvedených detailů platí jistá omezení. Název předmětu nesmí být delší, než 30 znaků a popis předmětu nesmí přesáhnout 1000 znaků. Dále částka pro okamžitý nákup musí být vždy vyšší, než minimální příhoz. A nakonec termín ukončení aukce musí být alespon hodinu po zahájení aukce, ale ne pozdější, než rok od zahájení aukce.
- **Moje aukce** Uživatel zde vidí seznam svých právě probíhajících aukcí. Po kliknutí na danou aukci je uživatel přesměrován na stránku konkrétní aukce.
- **Ukončené** V pravé části stránky je zobrazen seznam aukcí, které uživatel vyhrál nebo vytvořil a byly již ukončeny. Pomocí filtru lze přepínat mezi aukcemi, které uživatel vyhrál, a těmi, které vytvořil. Po kliknutí na konkrétní aukci se v levé části stránky zobrazí detaily pro předání předmětu , které zahrnují hlavní informace o předmětu a kontakt na prodávajícího nebo kupujícího.



Obrázek 5: Uživatelský prostor

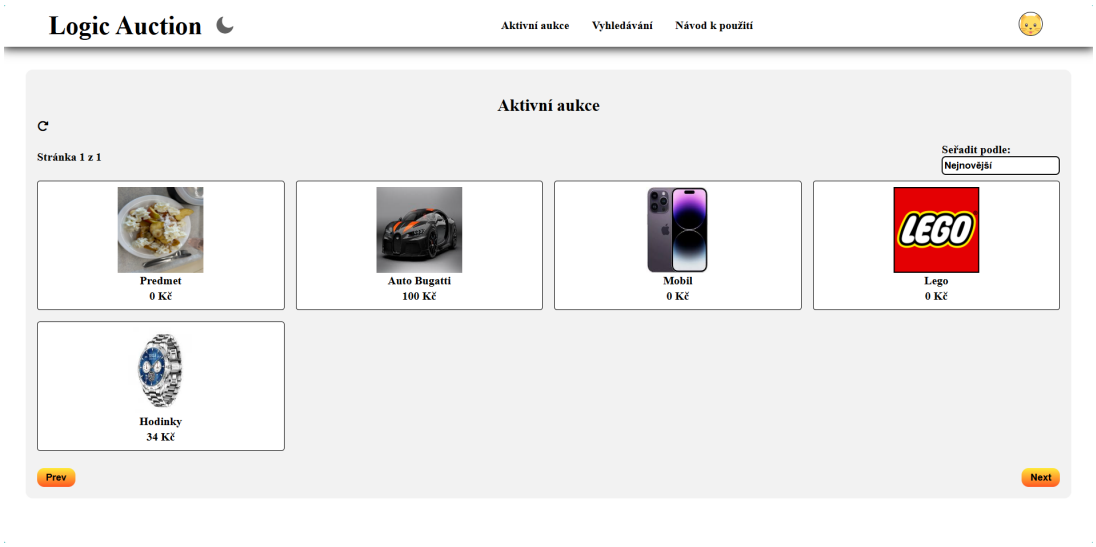
6.6 Pravidla aukce

Nyní se podíváme na to, jak aukce probíhá a jak správně dražit. Aukce začíná zadáním předmětu přihlášeným uživatelem. Od této chvíle mohou všichni přihlášení uživatelé (kromě prodávajícího) na aukci přihazovat. Tento typ aukce se podobá anglické aukci, kde všichni vidí aktuální nejvyšší nabídku. Pro příhoz je nutné zadat částku vyšší, než aktuální nejvyšší nabídka alespoň o hodnotu minimálního příhozu. Výhercem aukce se stává účastník s nejvyšším, tedy posledním , příhozem. Aukce může být ukončena třemi způsoby:

- **Dosažení termínu:** Aukce končí, když uplyne termín zadaný prodávajícím. Dosažení termínů je automaticky kontrolováno každou minutu. Oprávněný uživatel může také zkontrolovat dosažení termínu pomocí tlačítka v levém horním rohu stránky s aukcemi.
- **Okamžitý nákup:** Aukce končí, pokud je dosažena částka pro okamžitý nákup.
- **Ruční ukončení:** Prodávající nebo oprávněný uživatel může aukci okamžitě ukončit pomocí tlačítka u dané aukce.

Aukci lze také zrušit pomocí tlačítka podobného tomu pro ukončení aukce. Tuto akci může provést pouze prodávající nebo oprávněný uživatel. Aukce je pak vymazána a vítězem se nestává nikdo.

Po ukončení aukce se prodávajícímu i výherci přidá nový záznam o aukci na stránku s ukončenými aukcemi. Předání i platba probíhají mimo stránku.



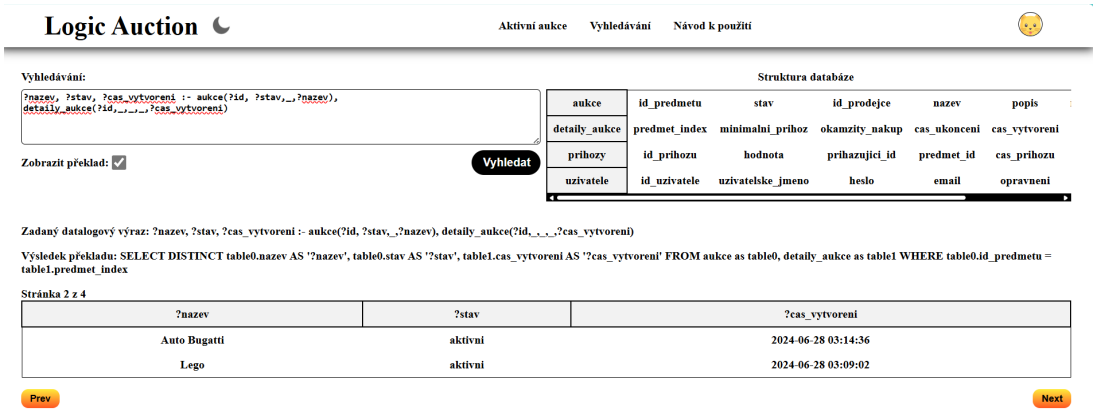
Obrázek 6: Probíhající aukce

6.7 Vyhledávání

Nakonec se podíváme na nejzajímavější část stránky, která ji odlišuje od ostatních. K vyhledávání má přístup pouze přihlášený uživatel. Pro snadnější orientaci je na stránce zobrazena část struktury databáze, která je uživateli přístupná. Do vstupního pole na stránce může uživatel zadat svůj vlastní datalogový dotaz , pomocí kterého může dotazovat na data v databázi. Pokud zadá nesprávný dotaz nebo se pokusí přistoupit k neoprávněné části databáze, bude na tuto skutečnost upozorněn červenou zprávou na obrazovce. Po zaškrtnutí příslušného políčka je také možnost zobrazit SQL dotaz, na který je datalogový dotaz přeložen. Při úspěšném vyhledání se zobrazí tabulka s výsledky hledání.

Oprávněný uživatel má kromě možnosti vyhledávání pomocí datalogového dotazu také možnost využít vstupní pole pro zadání vlastního datalogového pravidla a jeho definování.

Na obrázku 7 lze vidět datalogový dotaz, který se ptá na názvy předmětů , jejich stav a časy vytvoření.



Obrázek 7: Ukázka vyhledávání

6.8 Návod

Návod obsahuje shrnutí toho, jak aukce fungují a jak vytvářet datalogové dotazy. Dělí se na dvě části:

- První část se týká pravidel aukce. Uživatel se zde dozví, jak aukce fungují , jak vytvořit vlastní aukci a jak správně přihazovat na cizí aukce.
- Druhá část se zaměřuje na jazyk Datalog. Uživatel se zde dozví základní informace o jazyku Datalog, co je datalogový dotaz, proč ho potřebuje , z jakých částí se skládá a jak ho vytvořit. Je zde také uvedena struktura databáze, aby uživatel věděl, jaké informace může pomocí tohoto jazyka dohledat.

7 Testování aplikace

Aplikace byla vytvářena a testována především na lokálním webovém serveru Apache aplikace XAMPP. Spolu s ním byla využita lokální databáze MySQL Workbench 8.0 CE. Aplikace byla spouštěna na notebooku ASUS TUF Gaming F15 s procesorem Intel Core i7-12700H, 16GB RAM. V lokálním prostředí běžela aplikace bez problémů.

Aplikace byla rovněž testována na hostingovém webovém serveru logicauction.infinityfreeapp.com, kde je stále přístupná.

Pro správnou funkci webové aplikace je zapotřebí MySQL server a dále webový server s podporou PHP.

Zobrazení aplikace bylo pro Windows testováno na prohlížečích Google Chrome , Microsoft Edge a Mozilla Firefox. Pro android byl využit prohlížeč Samsung Internet a Google Chrome. Aplikace se zobrazovala a fungovala bez problémů.

Závěr

Cílem této bakalářské práce bylo vytvoření webové aplikace pro online aukci předmětů, která umožňuje prohledávání databáze pomocí jazyka Datalog.
Umožňuje přihlášení a registraci uživatelů, přihazovat na aukce a zakládání vlastních aukcí. Uživatelé mohou využívat stránku pro vyhledávání, kde si mohou vyzkoušet provádění dotazů na databázi v jazyku Datalog. V této části mohou oprávnění uživatelé také vytvářet datalogová pravidla. Aplikace je navržena s jednoduchým uživatelským rozhraním a je přizpůsobena pro různá zařízení. Pro zlepšení uživatelského zážitku je k dispozici také tmavý režim.

Aplikace byla vytvořena převážně za pomoci technologie AJAX a jazyku PHP. Motivací pro tuto práci mi byl stále rostoucí trend online obchodování a možnost využití logického jazyka v webové aplikaci.

Během vývoje aplikace jsem se seznámil s novými technologiemi a také získal první zkušenosti s prací na větším projektu. V budoucnu bych aplikaci rád rozšířil o několik dalších funkcionalit a částí. Mohly by se týkat například možnosti komunikace prostřednictvím chatu mezi výhercem aukce a prodávajícím nebo možností přidat více obrázků předmětu.

Vytvořená aplikace nabízí solidní základ pro budoucí vylepšení a přizpůsobení, čímž přispívá k inovaci v oblasti online aukcí a webových technologií.

Conclusions

The goal of this bachelor's thesis was to create a web application for an online auction of items that will allow searching the database using the Datalog language. It allows users to login and register, bid on auctions and create custom auctions. Users can use the search page to practice querying the database in Datalog. In this section, authorized users can also create datalog rules. The application is designed with a simple user interface and is adapted for different devices. A dark mode is also available to improve the user experience.

The application was created mainly with the help of AJAX technology and PHP language. My motivation for this work was the growing trend of online trading and the possibility of using logical language in a web application.

During the development of the application, I got acquainted with new technologies and also gained the first experience of working on a larger project. In the future, I would like to expand the application with several other functionalities and parts. This could include, for example, the ability to communicate via chat between the auction winner and the seller, or the ability to add more images of the item.

The created application offers a solid foundation for future enhancements and customizations, contributing to innovation in online auctions and web technologies.

A Obsah elektronických dat

`text/`
: Adresář s textem práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech (textových) příloh, a všechny soubory potřebné pro bezproblémové vytvoření PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu a příloh, vložené obrázky, apod.

`README.txt`
: Textový soubor s instrukcemi pro zprovoznění webové aplikace, včetně všech požadavků a webové adresy, na které je webová aplikace nasazena.

`src/`
: Zdrojový kód webové aplikace.

Literatura

[1] Anonymous. *Aukce - co to je, definice a koncept* [online]. 2024 [cit. 2024-7-2]. Dostupný z: ⟨https://cs.economy-pedia.com/11032608-auction⟩.

[2] TechTarget. *web application (web app)* [online]. 2024 [cit. 2024-7-2]. Dostupný z: ⟨https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app⟩.

[3] Anonymous. *Web application* [online]. 2024 [cit. 2024-7-3]. Dostupný z: ⟨https://en.wikipedia.org/wiki/Web_application⟩.

[4] Ceri, Stefano; Gottlob, Georg; Tanca, Letizia aj. What you always wanted to know about Datalog(and never dared to ask). *IEEE transactions on knowledge and data engineering*. 1989, roč. 1, č. 1, s. 146–166.

[5] Lark Technologies. *Datalog: Background and Evolution* [online]. [cit. 2024-7-6]. Dostupný z: ⟨https://www.larksuite.com/en_us/topics/ai-glossary/datalog#background-and-evolution-of-datalog⟩.

[6] Anonymous. *PHP* [online]. 2024 [cit. 2024-7-6]. Dostupný z: ⟨https://en.wikipedia.org/wiki/PHP⟩.

[7] Sheldon, Robert. *model-view-controller (MVC)* [online]. 2024 [cit. 2024-7-6]. Dostupný z: ⟨https://www.techtarget.com/whatis/definition/model-view-controller-MVC⟩.

[8] Chris, Kolade. *What is HTML – Definition and Meaning of Hypertext Markup Language* [online]. 2024 [cit. 2024-7-6]. Dostupný z: ⟨https://www.freecodecamp.org/news/what-is-html-definition-and-meaning⟩.

[9] Anonymous. *CSS* [online]. 2024 [cit. 2024-7-6]. Dostupný z: ⟨https://en.wikipedia.org/wiki/CSS⟩.

[10] Anonymous. *What is CSS?* [online]. 2024 [cit. 2024-7-6]. Dostupný z: ⟨https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS⟩.

[11] Anonymous. *JavaScript* [online]. 2024 [cit. 2024-7-7]. Dostupný z: ⟨https://en.wikipedia.org/wiki/JavaScript⟩.

[12] G., Domantas. *What Is AJAX and How Does It Work?* [online]. 2024 [cit. 2024-7-7]. Dostupný z: ⟨https://www.hostinger.com/tutorials/what-is-ajax⟩.

[13] Anonymous. *MySQL* [online]. 2024 [cit. 2024-7-7]. Dostupný z: ⟨https://en.wikipedia.org/wiki/MySQL⟩.