

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

DYNAMICKÁ WEBOVÁ NAVIGACE

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

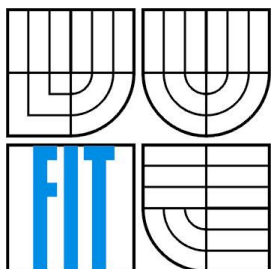
AUTOR PRÁCE  
AUTHOR

JAROMÍR VÝTVAR

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# DYNAMICKÁ WEBOVÁ NAVIGACE

DYNAMIC WEB NAVIGATION

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JAROMÍR VÝTVAR

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. RADEK BURGET, PH.D.

BRNO 2007

## Zadání bakalářské práce

Řešitel: **Výtvar Jaromír**  
Obor: Informační technologie  
Téma: **Dynamická webová navigace**  
Kategorie: Web

### Pokyny:

1. Seznamte se s technologiemi pro tvorbu dynamických webových stránek
2. Prostudujte prostředky pro sledování pohybu uživatelů po stránkách
3. Navrhněte vhodný způsob sledování pohybu uživatele v rámci webu s cílem generování dynamické navigace
4. Implementujte funkce pro generování dynamické navigace na webu na základě sledování pohybu uživatele
5. Implementujte možnost uložení historie navštívených stránek
6. Zhodnoťte dosažené výsledky a navrhněte pokračování projektu

### Literatura:

- Williams, H.E., Lane, D.: PHP a MySQL - Vytváříme webové databázové aplikace. Computer Press. 2002.
- Gilfillan I.: Myslíme v MySQL4.Grada, 2003.
- Welling, L., Thomsonová, L.: PHP a MySQL - rozvoj webových aplikací - 2. vydání, SOFTPRESS, 2003.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Burget Radek, Ing., Ph.D., UIFS FIT VUT**

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
Fakulta informačních technologií  
Ústav informačních systémů  
602 00 Brno, Božetěchova 2



---

doc. Ing. Jaroslav Zendulka, CSc.  
vedoucí ústavu

**LICENČNÍ SMLOUVA**  
**POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

**1. Pan**

Jméno a příjmení: **Jaromír Výtvar**

Id studenta: 84098

Bytem: Spojovací 635, 533 41 Lázně Bohdaneč

Narozen: 01. 05. 1985, Pardubice

(dále jen "autor")

a

**2. Vysoké učení technické v Brně**

Fakulta informačních technologií

se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....

(dále jen "nabyvatel")

**Článek 1**

**Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):  
bakalářská práce

Název VŠKP: Dynamická webová navigace

Vedoucí/školitel VŠKP: Burget Radek, Ing., Ph.D.

Ústav: Ústav informačních systémů

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

tištěné formě                      počet exemplářů: 1

elektronické formě                počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....

Nabyvatel

.....

Autor

## **Abstrakt**

Cílem této práce je obohatit prostředky programátorů webových prezentací v oblasti doplňkových dynamických navigací. Jedná se o drobečkovou dynamickou navigaci a navigaci pomocí oblíbených stránek vycházející z navrženého systému sledování pohybu návštěvníků po stránkách webové prezentace. Z výsledků sledování pohybu návštěvníků prezentací jsou generovány statistické informace. Celá práce je implementována ve skriptovacím jazyce PHP s napojením na databázi MySQL.

## **Klíčová slova**

dynamická navigace, web, PHP, MySQL, drobečková navigace, oblíbené stránky, statistiky

## **Abstract**

The objective of this thesis is to provide additional facilities to web presentation programmers in the area of dynamic navigation. It focuses on the dynamic breadcrumb navigation and the navigation through web page popularity that come from a proposed system of the page visitor monitoring. As an additional result of the monitoring, statistical information about the users is gathered. All the facilities are implemented in the PHP scripting language with the connection to the MySQL database server.

## **Keywords**

dynamic navigation, web, PHP, MySQL, breadcrumbs navigation, favourite pages, statistics

## **Citace**

Jaromír Výtvar: Dynamická webová navigace, bakalářská práce, Brno, FIT VUT v Brně, 2007

# Dynamická webová navigace

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením

Ing. Radka Burgeta, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jaromír Výtvar  
8.5.2007

## Poděkování

Chtěl bych poděkovat Ing. Radku Burgetovi Ph.D. za jeho cenné rady a připomínky k mé práci.

© Jaromír Výtvar, 2007.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Úvod.....	3
1 Vymezení pojmů .....	5
2 Analýza zadání .....	7
3 Návrh řešení .....	8
3.1 Sledování pohybu návštěvníků stránek .....	8
3.1.1 Zpracování sezení.....	8
3.1.2 Přenos SID pomocí cookies .....	8
3.1.3 Přenos SID pomocí URL.....	9
3.2 Dynamická webová navigace .....	9
3.2.1 Návrh drobečkové dynamické navigace.....	10
3.2.2 Zpracování sezení.....	11
3.2.3 Návrh tabulek .....	12
3.2.4 Bezpečnost přenášení SID v URL.....	13
3.3 Oblíbené.....	13
3.3.1 Návrh systému identifikace návštěvníka .....	14
3.3.2 Návrh tabulek .....	14
3.4 Statistiky .....	15
3.4.1 Detekce prohlížeče .....	15
3.4.2 Průměrný čas strávený na stránkách návštěvníky .....	16
3.4.3 Nejčastější vstupní a výstupní stránka webové prezentace .....	17
3.4.4 Celkový počet návštěv .....	17
3.4.5 Nejoblíbenější stránky.....	17
3.4.6 Počet celkově zobrazených stránek všemi návštěvníky .....	17
3.4.7 Počet online návštěvníků.....	17
3.5 Celkový návrh tabulek v databázi.....	18
3.5.1 Životní cyklus položek uložených v tabulkách .....	19
4 Implementace .....	20
4.1 Tabulky v databázi.....	20
4.2 Implementace dynamické navigace.....	21
4.2.1 Detekce možností přenosu SID .....	21
4.2.2 Unikátní SID .....	22
4.2.3 Funkce pro zápis stránky do drobečkové navigace .....	22
4.2.4 Funkce pro výpis drobečkové navigace .....	23
4.3 Implementace oblíbených stránek .....	24



4.3.1	Implementace identifikace návštěvníka .....	24
4.3.2	Funkce pro zápis stránky mezi oblíbené .....	25
4.3.3	Funkce pro výpis oblíbených stránek .....	25
4.3.4	Implementace mazání uložených oblíbených stránek .....	26
4.4	Implementace statistik .....	26
4.4.1	Detekce prohlížeče .....	27
4.4.2	Funkce pro získání statistických údajů .....	28
4.4.3	Likvidace dat pro statistiky .....	30
4.5	Rozdělení do souborů .....	30
4.5.1	Dwn_start.php .....	30
4.5.2	Dwn_function.php .....	31
4.5.3	Dwn_stat_function.php .....	31
4.5.4	Dwn_createtable.php .....	31
4.5.5	Dwn_deletetable.php .....	31
4.5.6	Dwn_config.php .....	31
5	Závěr .....	32

# Úvod

Internet je v současné době jedním z největších zdrojů informací pro člověka. Již skoro každý zde hledá nejrůznější informace přístupné přes nejrozšířenější službu internetu WWW (World Wide Web). Proto se stále více hledí na úspěšnost jednotlivých webových prezentací.

Mezi nejdůležitější vlastnosti úspěšných webových prezentací patří zajisté jejich vlastní obsah, ale také často opomíjená jednoduchá přístupnost hledaných informací. Tato přístupnost je řešena navigací, pomocí které se návštěvníci orientují ve struktuře webové prezentace a hledají požadované informace. Je-li dobrá, návštěvník se rychle dostává k cíli a to i při opakovaně hledaných informacích. Je-li špatná, návštěvník brzy prezentaci opouští a hledá požadované informace jinde. Navigace je základním kamenem každé webové prezentace. Její základní rozdělení je na hlavní, často také označovanou jako základní a doplňkovou.

Hlavní navigace slouží nejvíce k rozdělení struktury webové prezentace do kategorií. Návštěvník ji využívá nejčastěji jako rozcestník při hledání cesty k požadovaným informacím. Zůstává stále na stejném místě a je součástí každé stránky prezentace. Nejčastěji bývá ve formě menu či záložek.

Doplňková navigace doplňuje hlavní a slouží návštěvníkovi k rychlé orientaci ve struktuře webu, pomáhá mu vyhledávat informace, vracet se na oblíbené a již navštívené stránky. Mezi klasické zástupce doplňkové navigace patří drobečková navigace.

Název drobečková vznikl z pohádky o Jeníčkovi a Mařence, kteří cestou lesem použili drobečky chleba k označení cesty zpět domů. Rozlišujeme dva druhy této navigace, a to dynamickou a statickou. Statická navigace slouží jako textová reprezentace pozice stránky v hierarchii webové prezentace. Pozice stránky je nezávislá na cestě, kterou se návštěvník na stránku dostal. Dynamická drobečková navigace ukazuje cestu, jakou se uživatel na stránku dostal. Různí návštěvníci webu můžou mít na stejné stránce různé složení drobečků dynamické navigace podle různého průchodu stránkami prezentace. Tyto drobečky mohou také zastoupit tlačítko zpět a návštěvník se dostane zpět přesně na stránku, kterou vyžaduje.

Dalšími možnostmi doplňkové navigace mohou být například zobrazení oblíbených stránek každého návštěvníka. Zde pak návštěvník rychle nalezne své oblíbené stránky prezentace a nemusí ztrácet čas zdlouhavým procházením struktury webu až k požadované stránce. Často se využívá i mapa webové prezentace.

Tato práce se věnuje právě doplňkovým navigacím, které rozšiřuje, a dává tak možnost programátorům webových aplikací je jednoduše začlenit do svých prezentací. Práce se zabývá dynamickou drobečkovou navigací, která není prozatím příliš rozšířena pro svou implementační náročnost a nedostupnost jednoduchého nástroje pro začlenění této navigace do webové prezentace. Součástí práce je také zobrazování oblíbených stránek jednotlivých návštěvníků. Tato možnost také

bohužel není příliš rozšířená na webových prezentacích, přičemž by mohla značně usnadnit právě často se vracejícím návštěvníkům přístup na své oblíbené stránky.

Nejčastějším měřítkem úspěšnosti webových prezentací se stávají různé statistiky přístupů. Tyto statistiky jsou dnes již nutností každé úspěšné webové prezentace. Odráží se v nich spokojenost návštěvníků, úspěšnost webové prezentace v porovnání s konkurencí a lze z nich vyčíst mnoho údajů o návštěvnících, ze kterých se dají odvodit důležité informace pro vylepšení webových prezentací. Nejčastější statistiky jsou počet návštěv prezentace, počet zobrazených stránek prezentace, dále také informace o návštěvnících, například jejich prohlížeče. Poslední součástí této práce je také zobrazování základních statistických údajů a to i takových, které nejsou obvyklou součástí nabídky specializovaných internetových serverů zabývajících se statistikami přístupů. Tyto statistiky jsou například nejčastější stránka, ze které návštěvníci opouštějí webovou prezentaci, nebo délka pobytu návštěvníka na stránkách.

V následujících kapitolách je popsán návrh jednotlivých částí dynamické webové navigace a přidružených součástí a následně jejich implementace. Nejdůležitější částí pro uživatele dynamické webové navigace je příloha s uživatelským návodem.

# 1 Vymezení pojmů

## **World Wide Web (www, web)**

Celosvětově nejrozšířenější služba internetu vytvářející síť dokumentů neboli webových stránek. Je založena na třech technologiích HTML, HTTP a URL.

## **HTML**

Je zkratka z anglického HyperText Markup Language. Označuje jazyk pro vytváření dokumentů publikovaných pomocí www.

## **HTTP**

Zkratka anglických slov Hyper Text Transfer Protocol. Internetový protokol umožňující přenos dokumentů po síti internet.

## **URL**

URL je zkratka anglického Uniform Resource Locator. Je to řetězec znaků s definovanou strukturou, který slouží k přesné specifikaci umístění zdrojů informací na internetu.

## **PHP**

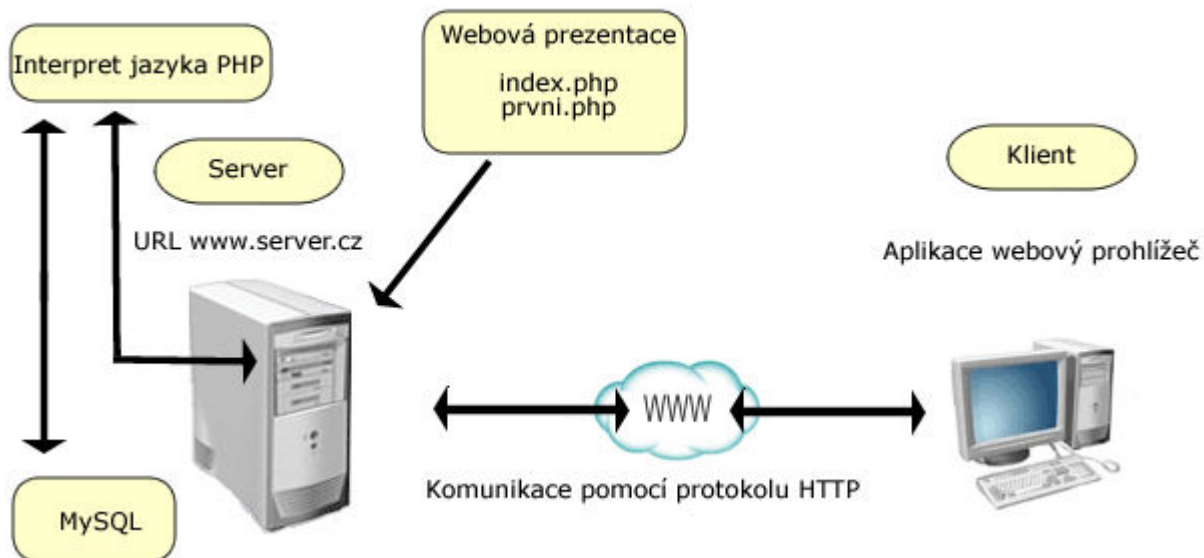
Skriptovací jazyk pro tvorbu dynamických webových stránek. Kód je začleňován do HTML stránek a je vykonáván na straně serveru. Jazyk PHP je nezávislý na platformě a podporuje připojení k databázovým aplikacím.

## **MySQL**

Databázový systém sloužící k uchování dat a k rychlému přístupu k nim. Pro operace s daty se používá jazyk SQL.

## **Webová aplikace v PHP**

Aplikace je založena na modelu klient – server. Na serveru jsou uloženy webové stránky napsané v jazyce HTML s kódem PHP nebo přímo v kódu PHP. Klient požaduje zobrazení stránky a vysílá požadavky na server. K tomu využívá aplikaci webového prohlížeče. Komunikace probíhá pomocí protokolu HTTP a požadovaná stránka je určena pomocí URL. Server požadavek přijme a zpracováním PHP kódu vygeneruje odpověď ve formátu HTML stránky, kterou zasílá zpět klientovi.



Obrázek č.1 – Schématický obrázek komunikace mezi klientem a serverem

### Cookies

Při komunikaci mezi webovým prohlížečem klienta a serverem může dojít k přenosu dat ze serveru na klienta. Tato data jsou ukládána jako cookies u klienta. Při dalším požadavku na komunikaci s tím samým serverem jsou data v cookies odesílána zpět serveru.

## 2 Analýza zadání

Hlavním cílem této práce je navrhnout vhodný způsob sledování pohybu uživatele v rámci webu s cílem generování dynamické drobečkové navigace. Navržený způsob sledování pohybu uživatele musí podporovat povolené i vypnuté ukládání cookies. Současně také musí být navržen způsob ukládání historie navštívených stránek s cílem generování seznamu oblíbených stránek jednotlivých návštěvníků podle počtu návštěv jednotlivých stránek prezentace.

Z informací získaných sledováním pohybu návštěvníků po webové prezentaci generovat statistiky umožňující sledovat oblíbenost webu a chování jeho návštěvníků.

Implementace generování dynamické navigace a oblíbených stránek v podobě funkcí s důrazem na jednoduchost při začleňování do webových prezentací. Celý systém bude implementován pomocí jazyka PHP a databázového systému MySQL.

## 3 Návrh řešení

Prvním úkolem při návrhu řešení je prostudovat možnosti sledování pohybu návštěvníků po webové prezentaci a vybrat nejvýhodnější řešení. Navrhnout systém ukládání informací o jejich pohybu pro generování dynamické navigace a oblíbených stránek do databáze. Zjistit možnosti získání různých statistik z uložených informací a vymyslet životní cyklus položek uložených v tabulkách databáze.

### 3.1 Sledování pohybu návštěvníků stránek

Přenos informací ve webových aplikacích probíhá pomocí protokolu HTTP. Tento protokol je bezstavový, což znamená, že každý požadavek se zpracovává samostatně bez jakékoliv znalosti o předchozích i následujících požadavcích. V době vývoje internetu právě tato vlastnost výrazně přispěla k masivnímu rozšíření protokolu HTTP, bohužel však znemožňuje programátorům jakékoliv sledování pohybu návštěvníků po webových prezentacích a vytváření tak komplexních webových aplikací reagujících na chování uživatele.

Tento problém byl zpočátku řešen pomocí ukládání veškerých informací týkajících se návštěvníka na disk jeho počítače, ale tento postup nebyl příliš vhodný pro omezenou velikost ukládaných dat. Dnes se již tento problém řeší zpracováním sezení.

#### 3.1.1 Zpracování sezení

Zpracování sezení je řešením problému bezstavosti protokolu HTTP. Probíhá tak, že každý nově příchozí návštěvník webové prezentace získá jedinečný identifikační atribut, kterému se říká identifikátor sezení. Anglický výraz pro sezení je session, proto bývá tento identifikátor označován jako session ID zkráceně SID.

Tímto SID je každý návštěvník prezentace jednoznačně identifikován, a proto můžeme k tomuto identifikátoru dát do vzájemného vztahu jakoukoliv hodnotu vztahující se k návštěvníkovi. Tyto svázané proměnné pak lze uložit například do databáze, kde SID je primárním klíčem svazujícím dohromady všechny další atributy daného uživatele. Problém, který zde vzniká, je v přenosu SID. Řešení jsou dvě. Uložením SID do souboru na straně návštěvníka, nebo-li použitím cookies nebo přenosem SID přes URL mezi jednotlivými stránkami webové prezentace.

#### 3.1.2 Přenos SID pomocí cookies

V protokolu HTTP se označuje malé množství dat, které jsou serverem posílány prohlížeči a poté ukládány do počítače návštěvníka webu jako cookie. Jejich velikost je omezena na 4096 bajtů, proto se do nich nevejde velké množství dat. Na uložení SID však toto místo stačí.

Při vstupu návštěvníka na webové stránky se uloží jeho identifikátor do souboru na jeho počítači jako cookie, kde je po celou dobu návštěvy k dispozici. Když je potřeba, dotáže se server, který zpracovává požadavek klienta, na jeho SID uložené v cookie. Tím získá SID pro potřeby stránky, která je právě zpracovávána, a dodají se různé druhy informací vztahující se k získanému SID. Protože cookie může zůstat u klienta i poté, co sezení skončí, může se načíst během následujícího sezení, z čehož vyplývá, že se dají udržet informace i přes dlouhá časová období nečinnosti návštěvníka. Dobu platnosti cookies je možno nastavit. Po jejím uplynutí je soubor obsahující cookie smazán.

Je potřeba mít na paměti, že správa cookies je plně v kompetenci klienta. Může se stát, že klient si ať již úmyslně nebo neúmyslně soubor obsahující cookie smaže, nebo dokonce ukládání cookies na svém prohlížeči zakáže. Jsou také mnohé prohlížeče, které ani ukládání cookies neumožňují. V těchto situacích přenos SID pomocí cookies selhává.

### **3.1.3 Přenos SID pomocí URL**

Druhá metoda přenášení SID spočívá jednoduše v tom, že se SID připojí na konec URL identifikující danou stránku. Tato metoda odstraňuje možnost, že by si návštěvník stránek vypnul podporu cookies a předávání identifikátoru by nemohlo pokračovat, protože přenos SID je na uživatelově prohlížeči i počítači nezávislý.

Tato metoda však také trpí nedostatky. Nedokáže udržovat trvalé informace po delší dobu než jednoho sezení, protože předávání SID pomocí URL nepokračuje po opuštění webové prezentace. Nic nebrání návštěvníkovi, aby zkopíroval adresu a odeslal ji nějakému jinému uživateli. Ten pak pokračuje v sezení jako původní majitel. Stane se tedy to, že na stránkách prezentace se pohybují dva návštěvníci s jedním SID. Toto může v některých situacích způsobit nečekané chování a zmatek. Použití ukládání SID do URL má ale ještě jednu značnou nevýhodu. Pro zajištění funkčnosti takto předávaného SID musíme ke všem lokálním adresám URL tento identifikátor připojit. Stačí jen na jednom místě tento řetězec předávání SID přerušit a identifikátor je ztracen.

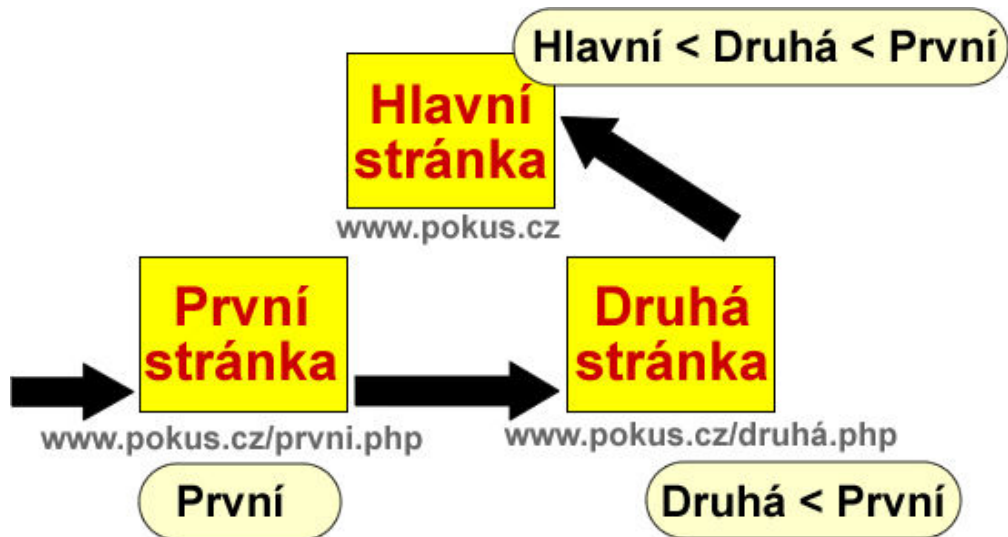
## **3.2 Dynamická webová navigace**

Návrh dynamické webové navigace spočívá v upřesnění a doplnění formy a grafické reprezentace dynamické drobečkové navigace. Součástí je i návrh zpracování sezení, přesný výběr přenosu a uchování SID a návrh tabulek k ukládání záznamů pro navigaci.



### 3.2.1 Návrh drobečkové dynamické navigace

Generování dynamické drobečkové navigace je vizuálně zobrazené na obrázku č.2. Tlustá černá šipka znázorňuje průchod návštěvníka pokusnou webovou prezentací.



Návštěvník dle obrázku vstupuje do webové prezentace přímým zadáním URL do prohlížeče a dostává se na stránku První.php. Proběhne zpracování sezení a návštěvníkovi je přidělen identifikátor SID. Ve výpisu dynamické drobečkové navigace se objeví titulek navštívené stránky První. Návštěvník dále pokračuje přes odkaz na další stránku. Při přístupu na stránku Druhá.php je identifikován podle SID a dynamická drobečková navigace vypíše titulky stránek, které již navštívil, v pořadí zleva doprava od aktuální, kde se nachází návštěvník, po první stránku, ze které přišel. Jeho cesta končí na Hlavní stránce, kde navigace vypíše celou cestu, kudy návštěvník webové prezentace procházel. Jednotlivé drobečky navigace jsou zároveň odkazy na jednotlivé stránky.

Výpis jednotlivých stránek jsem navrhl tak, že aktuální stránka je vždy vypisovaná jako první a poslední směrem vpravo je vypisována první stránka, na kterou návštěvník přistoupil. Tato metoda je více intuitivní než výpis v opačném pořadí, kde aktuální stránka stále mění své místo a posouvá se doprava.

Zajímavou kapitolou u vizuálního zobrazování drobečkové navigace je oddělovač mezi jednotlivými drobečky navigace. Nejčastěji se používají symboly >, :, >>, /, - nebo malý obrázek. Žádný z těchto symbolů však výrazně nepřevyšuje v počtu použití ostatní, proto nechávám na uživateli, aby si rozhodl, který mu bude nejvíce vyhovovat.

Jednou z hlavních vlastností dynamické drobečkové navigace, kterou může návštěvníkovi nabídnout, je návrat o několik stránek zpět jednoduchým kliknutím na požadovanou stránku neboli drobeček navigace. Klasickým chováním návštěvníka bez této možnosti je několikanásobné použití tlačítka zpět. V takové situaci lze nabídnout dvě řešení chování dynamické navigace. Stránku, na

kteřou se návštěvník vrací, vložit na začátek seznamu navigace a jinak pořadí stránek neměnit. Většina uživatelů webových prezentací však očekává odstranění všech stránek z výpisu drobečkové navigace předešlých té, na kterou se chtějí vrátit. Z tohoto důvodu jsem se rozhodl navrhnout drobečkovou navigaci, u které jsou při návratu předešlé drobečky odstraněny.

### 3.2.2 Zpracování sezení

Při návrhu zpracování sezení jsem se rozhodl spojit obě možnosti přenosu SID a to jak pomocí cookies, tak pomocí URL. Přednost však dostává použití cookies z důvodu udržení přehlednosti URL. Toto řešení nabízí funkčnost dynamické drobečkové navigace za všech podmínek. Pro účely dynamické drobečkové navigace je hlavním důvodem zpracování sezení sledování návštěvníka při jedné návštěvě webové prezentace. Po opuštění stránek prezentace již není potřeba návštěvníka dále sledovat. Jeho další návštěva bude zpracována jako nové sezení s novým SID.

Celé zpracování sezení probíhá takto. Návštěvník poprvé přistoupí na jednu ze stránek webové prezentace. Pokud nelze zjistit jeho identifikátor SID ze souboru cookie ani z URL, je mu vygenerováno nové unikátní SID. Dalším krokem je nutné zjistit, zda jsou u návštěvníka povoleny cookies, a tento identifikátor do nich vložit nebo v případě, že povoleny nejsou, identifikátor připojit k URL.

V případě, že u návštěvníka prezentace již lze SID získat z URL při zpracování sezení, dojde opět k pokusu uložit jeho identifikátor do cookie. Nepovede-li se tato operace, pokračuje se dále v přenosu SID pomocí URL. Dojde-li k uložení identifikátoru do cookie například v situaci, kdy návštěvník ukládání cookies povolí, identifikátor z URL je plynule přesunut do cookie a z URL odstraněn. Nedojde tedy k žádné ztrátě údajů o pohybu návštěvníka z doby, kdy se jeho identifikátor přenášel pomocí URL.

Pokud ovšem nastane situace, kdy návštěvník měl povolené ukládání cookies, přistoupil na stránky prezentace a do cookie mu byla nastavena jeho hodnota SID. Poté však ukládání cookies ve svém prohlížeči zakázal. Tímto se stal jeho identifikátor nedostupný a do URL mu byl přiřazen nový. Informace o jeho minulém pohybu je tak ztracena.

Důležitou součástí zpracování sezení je jeho ukončení. V případě použití ukládání SID do URL je situace jednoduchá. Při uzavření prohlížeče nebo přechodu na stránku jiné webové prezentace je SID ztraceno a sezení je ukončeno. V druhém případě je nutno nastavit životnost cookie na stanovenou dobu, po jejímž uplynutí již můžeme s jistotou tvrdit, že návštěvník z webové prezentace odešel. Životnost cookie je potřeba při každém přístupu na stránku prezentace obnovit. Pokud bude doba životnosti cookie příliš krátká, dojde k situaci, při které se návštěvník například začte do zajímavého článku a bude-li chtít dále pokračovat v prohlížení prezentace, dojde ke smazání cookie a tím k ukončení sezení a ztrátě navigace. Bude-li doba příliš dlouhá, bude návštěvník i v případě, že už návštěvu stránek ukončil a za nějakou dobu se opět vrátil, vidět svůj minulý průchod prezentací.

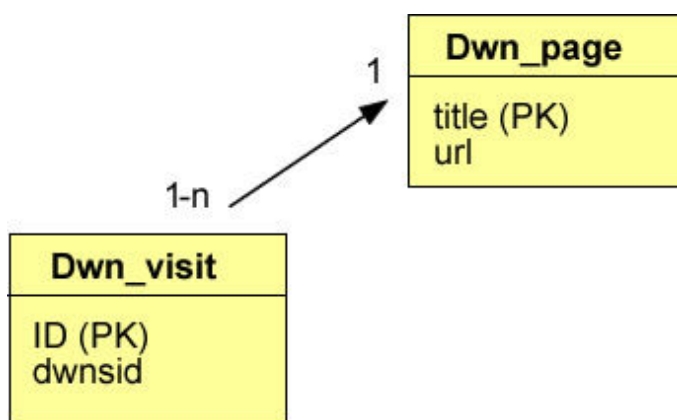
Řešením tohoto problému je nastavení životnosti cookie na dobu spuštěného prohlížeče. Po jeho uzavření dojde ke smazání cookie a tím i k ukončení session.

### 3.2.3 Návrh tabulek

Dalším bodem návrhu dynamické webové navigace je způsob ukládání informací o průchodu návštěvníka jednotlivými stránkami webové prezentace. Jako datové skladiště jsem zvolil relační databázi.

#### 3.2.3.1 Tabulka Dwn\_visit, ukládání a výpis drobečkové navigace

Návrh tabulek pro drobečkovou navigaci je zobrazen na obrázku č.3.



Obrázek č.3 – Tabulka pro uložení informací pro dynamickou navigaci

Při každé návštěvě stránky se do tabulky Dwn\_visit uloží záznam o této návštěvě. Ukládá se identifikátor návštěvníka a odkaz na záznam identifikující stránku v tabulce Dwn\_page. Generování drobečkové navigace je proto velice jednoduché. Probíhá výpisem takto uložených návštěv pro jednoho návštěvníka.

#### 3.2.3.2 Tabulka Dwn\_page a identifikace web stránky prezentace

Tabulka Dwn\_page obsahuje stránky webové prezentace. Pro potřebu dynamické drobečkové navigace bylo potřeba navrhnout jednoznačnou identifikaci stránek webové prezentace. Jako nejvhodnější identifikátor se jeví titulek každé stránky. Je to řetězec, který se vypisuje na stránce mezi tagy <title>titulek</title> v jazyce pro tvorbu webových stránek HTML, a prohlížeče jej zobrazují v horním panelu okna. Tento titulek je také zobrazován jako drobečky dynamické navigace, proto musí být takový, aby z něho návštěvník intuitivně odvodil obsah stránky. Nutnou podmínkou funkčnosti takto identifikovaných stránek je podmínka, aby autor webové prezentace jednotlivé stránky skutečně rozlišoval titulkem.

Další možností identifikace jednotlivých stránek prezentace je pomocí jejich URL. Bohužel z důvodu přenášení různých proměnných jako součástí URL by velice pravděpodobně mohla nastat situace, kdy podle dvou různých URL bude identifikována jedna a ta samá stránka. Další nevýhodou je vypisování celých URL jako text drobečků navigace, což je pro návštěvníka matoucí a nepřehledné. Text URL v drobečkách navigace by mohl být nahrazen také titulkem, ale tím by se mohly vyskytovat v této navigaci dva i více stejně označené odkazy navigace. Každý by pak ukazoval na jinou stránku prezentace, výsledek by opět návštěvníka mátl.

Jednou z možností také může být spojit titulek a URL do jednoho dlouhého identifikátoru stránky. Tato možnost však trpí stejnými problémy jako identifikace pomocí URL.

Tento problém by mohl být vyřešen odstraněním proměnných přenášených pomocí URL. Stránka by tak byla identifikována jednoznačně pomocí URL. Bohužel mnoho webových prezentací je tvořeno jako jeden soubor a stránky jsou rozlišovány právě pomocí proměnných přenášených pomocí URL. Toto řešení je tedy také univerzálně nepoužitelné.

Po úvaze nad všemi těmito možnostmi identifikace stránky webové prezentace jsem zvolil možnost identifikovat stránku pouze pomocí titulku. Je tedy nutné, aby tvůrce webové prezentace, na které má správně fungovat dynamická navigace, rozlišoval jednotlivé stránky pomocí jedinečného a výstižného titulku. Zvolením výstižných pojmenování stránek usnadní návštěvníkovi orientaci v navigaci.

### **3.2.4 Bezpečnost přenášení SID v URL**

Možnosti zcizení sezení a získání přístupu k datům jiného návštěvníka webové prezentace spočívají v získání identifikátoru sezení a jeho následnému použití. Přenášení identifikátoru jednoznačně určujícího návštěvníka je z hlediska bezpečnosti riskantní záležitost. V případě přenášení identifikátoru jako součást URL je tento identifikátor viditelný přímo v adresním řádku prohlížeče, ze kterého jej může kdokoliv zkopírovat a využít pro krádež sezení. Z tohoto důvodu se nedoporučuje tento způsob používat. V tomto případě ale není SID spojeno s žádnými citlivými údaji návštěvníka, proto je jeho přenos pomocí URL použitelný. Dále také po ukončení jeho návštěvy je toto SID již neplatné a jeho další návštěva bude probíhat pod jiným identifikátorem. Jediné čeho může zloděj dosáhnout, je zjištění pohybu návštěvníka po stránkách webové prezentace a to právě z jednoho záznamu jeho návštěvy.

## **3.3 Oblíbené**

Zobrazení oblíbených stránek návštěvníka je jednou z dalších možností doplňkové navigace. Zvláště velký význam tento druh navigace získává při častém návratu návštěvníka na webovou prezentaci. Pojem oblíbená stránka můžeme brát ze dvou úhlů pohledu. Oblíbená stránka může být taková stránka, na které návštěvník strávil nejvíce času, nebo to může být stránka, kterou navštěvuje

nejčastěji. Při návrhu jsem se rozhodl pro druhou možnost, jelikož více vystihuje vžitý pojem oblíbených stránek.

Návrh této části práce spočívá v identifikaci návštěvníka po delší dobu než pouze po dobu jedné návštěvy webové prezentace, jak tomu bylo v případě dynamické drobečkové navigace, kde se při každé nové návštěvě návštěvníkovi generovalo nové SID. Z tohoto důvodu jsou také SID dynamické drobečkové navigace a navigace pomocí oblíbených stránek odlišné a každé bude uchováváno jinde. Součástí návrhu je detailní rozbor možností uložení SID. Dále je nutno navrhnout způsob ukládání oblíbených stránek každého uživatele.

### **3.3.1 Návrh systému identifikace návštěvníka**

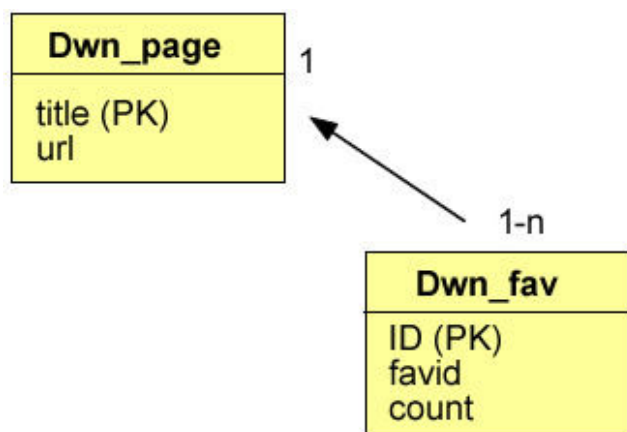
Pro identifikaci návštěvníka na dobu delší než jedné návštěvy webové prezentace již není možné použít přenos SID pomocí URL. Po opuštění stránek je takto přenášený identifikátor ztracen. Jediná možnost jak uchovat identifikátor návštěvníka po delší dobu je uložit ho na straně klienta, tedy na disk jeho počítače.

K tomuto slouží cookies. Při vytváření cookies lze definovat čas jejich životnosti. Pokud tento čas nastavíme na dostatečně vzdálené datum, můžeme návštěvníka identifikovat i po delší době, kdy webovou prezentaci opět navštíví. Toto řešení má však vážné nedostatky. Tím že je identifikátor uložen na straně klienta, je jeho správa v rukou klienta. Ten může cookie smazat či zakázat a tím je jeho identifikace znemožněna. Některé prohlížeče bohužel ani ukládání cookies neumožňují. V takovýchto případech je navigace pomocí oblíbených nefunkční a pokud návštěvník má přání tuto navigaci použít, musí použít prohlížeč s podporou cookies, popřípadě si jejich ukládání povolit.

Jiný přístup k identifikaci uživatele je použitím nějakého identifikátoru, například jména nebo hesla. Tímto způsobem se návštěvník identifikuje. Tento postup však požaduje intervenci ze strany návštěvníka a je tedy pro tyto účely zbytečný.

### **3.3.2 Návrh tabulek**

Při příchodu návštěvníka je nejprve proveden pokus o jeho rozpoznání. Nevlastní-li návštěvník SID, je mu vygenerováno nové a uloženo do cookie. V tabulce `dwn_fav` (obrázek č.4) jsou uchovávány informace o počtu přístupů návštěvníka k jednotlivým stránkám webové prezentace, které jsou v tabulce `dwn_page`.



Obrázek č.4 – Tabulky pro uložení informací o oblíbených stránkách

Výpis oblíbených probíhá získáním všech záznamů pod stejným identifikátorem odpovídajícím SID návštěvníka. U každé stránky je uveden počet přístupů návštěvníka. Tím získáme nejoblíbenější stránku.

## 3.4 Statistiky

Další součástí práce je zpracování statistických údajů o pohybu návštěvníků po webové prezentaci. Na internetu je mnoho komerčních i bezplatných serverů zabývajících se touto tematikou nabízejících různé statistické služby. Jejich výhodou je vykonávání veškeré režie mimo vlastní webové prezentace a tím snížení zátěže serveru, na kterém je prezentace umístěna. Nevýhodou je utajení skutečného principu získávání statistických údajů a často také mnoho statistických údajů bez vysoké vypovídací hodnoty o chování návštěvníků webové prezentace.

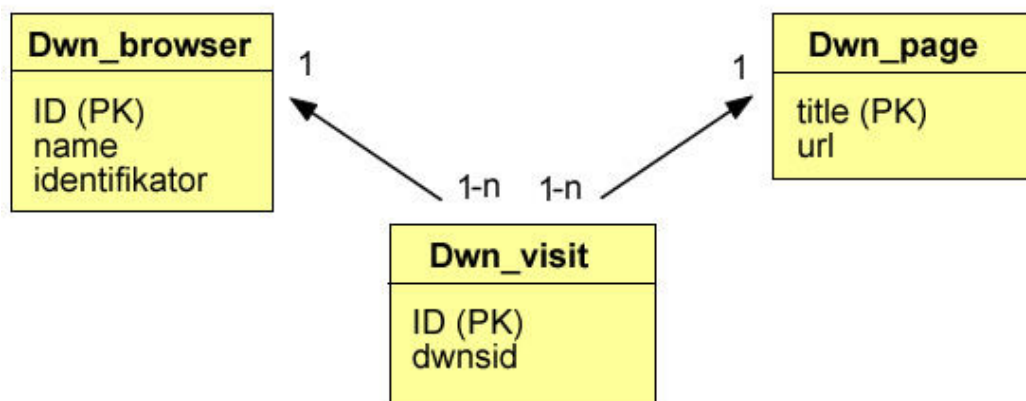
Při návrhu se proto zaměřuji na zajímavé statistiky z pohledu správce a programátora webových prezentací. Vycházím z uložených informací získaných zpracováním pohybu návštěvníků po stránkách za účelem generování drobečkové dynamické navigace. Tabulka s uloženými informacemi o oblíbených stránkách není použitelná. Informace o oblíbených stránkách se ukládají jen návštěvníkům se zapnutou podporou cookies.

Součástí návrhu statistických funkcí je výběr zajímavých statistických veličin a obohacení tabulek databáze pro získání statistických údajů.

### 3.4.1 Detekce prohlížeče

Jedním z užitečných údajů pro programátory webových aplikací jsou typy prohlížečů, které návštěvníci používají. Tento údaj je důležitý, protože každý prohlížeč zobrazuje webové stránky s drobnými odchylkami. Programátoři pak mají možnost nejvíce stránky optimalizovat pro nejpoužívanější prohlížeč návštěvníků.

Každý prohlížeč posílá při každém požadavku pro zobrazení stránky na server svůj identifikátor. Pomocí tohoto identifikátoru je tento prohlížeč identifikovatelný. Proto je tedy nutné mít v databázi připravenou tabulku s identifikačními řetězci prohlížečů, se kterou se zaslaný identifikační řetězec může porovnat. Tím dojde k jeho detekci. Informace o prohlížeči jsou rozšířením tabulky Dwn\_visit.



Obrázek č.5 – Tabulka pro identifikaci prohlížeče

### 3.4.2 Průměrný čas strávený na stránkách návštěvníky

Tato statistika naráží na problém změření časového úseku od vstupu návštěvníka na stránky webové prezentace po jejich opuštění. K uložení času je zapotřebí přidat k tabulce Dwn\_visit položku obsahující čas vzniku každého záznamu. Čas přístupu na první stránku je jednoznačně uložen jako čas vzniku záznamu o návštěvě první stránky návštěvníkem. Problém nastává při pokusu změřit čas opuštění prezentace. Přesný čas odchodu návštěvníka nelze zjistit. Pro účely statistiky se tedy musí použít čas přístupu na poslední navštívenou stránku prezentace. Čas, který návštěvník strávil na této stránce, je bohužel nezjistitelný a zkresluje tuto statistiku.

Z tohoto důvodu jsem se rozhodl počítat jen jednu celkovou průměrnou hodnotu času strávenou návštěvníky na stránkách. Tato hodnota je orientační a zajímavé jsou spíše její změny. V případě, že se čas průměrně strávený návštěvníky na stránkách prezentace v nějakém období zvýšil oproti jinému období, je to známka zvýšeného zájmu o tento web. Pokud se však hodnota snižuje, je to známka, že návštěvníci nenacházejí na stránkách webu nic zajímavého a rychle je opouští.

$$\text{Průměrná doba strávená návštěvníky na stránkách prezentace} = \frac{\sum X}{Z}$$

X – součet všech časů jednotlivých návštěv na stránkách prezentace

Z – počet všech návštěv

### **3.4.3 Nejčastější vstupní a výstupní stránka webové prezentace**

Tyto dvě statistiky jsou velmi důležité z hlediska sledování chování návštěvníků pohybujících se po stránkách prezentace. Nejčastější vstupní stránka nemusí být úvodní stránka prezentace. Velmi často návštěvníci přistupují na stránku, kterou jim nabídne vyhledávač přímo s obsahem, který hledají.

Znalost nejčastější výstupní stránky, jinak řečeno stránky, ze které návštěvník nejčastěji opouští prezentaci, má pro provozovatele webové prezentace vysokou cenu. Ve většině případů to znamená, že se stránkou jsou návštěvníci nespokojení nebo nefunguje.

Pro získání těchto statistik do databáze není potřeba nic přidávat. Informace se berou z tabulky Dwn\_visit, kde jsou již předpřipravené pomocí systému ukládání informací dynamické webové navigace.

### **3.4.4 Celkový počet návštěv**

Tuto statistiku získáme z tabulky Dwn\_visit jednoduchým sečtením všech uložených SID. Statistika ukazuje počet návštěv. Návštěv může být více od jednoho návštěvníka.

### **3.4.5 Nejoblíbenější stránky**

Další zajímavá statistika zobrazuje nejnavštěvovanější stránky prezentace. Lze z ní snadno zjistit, jaké jsou nejoblíbenější stránky návštěvníků. Statistika vzniká sečtením všech zobrazení každé stránky prezentace, která byla alespoň jednou navštívena.

### **3.4.6 Počet celkově zobrazených stránek všemi návštěvníky**

Statistika udává počet celkově zobrazených stránek všemi návštěvníky. Svoji vypovídací hodnotu získává spíše porovnáním s počtem návštěv. Když je počet návštěv shodný s počtem zobrazených stránek, znamená to situaci, kdy návštěvníci přicházejí na stránky prezentace a po shlédnutí jedné stránky ji opouštějí. Vzniká sečtením všech zobrazených stránek webové prezentace.

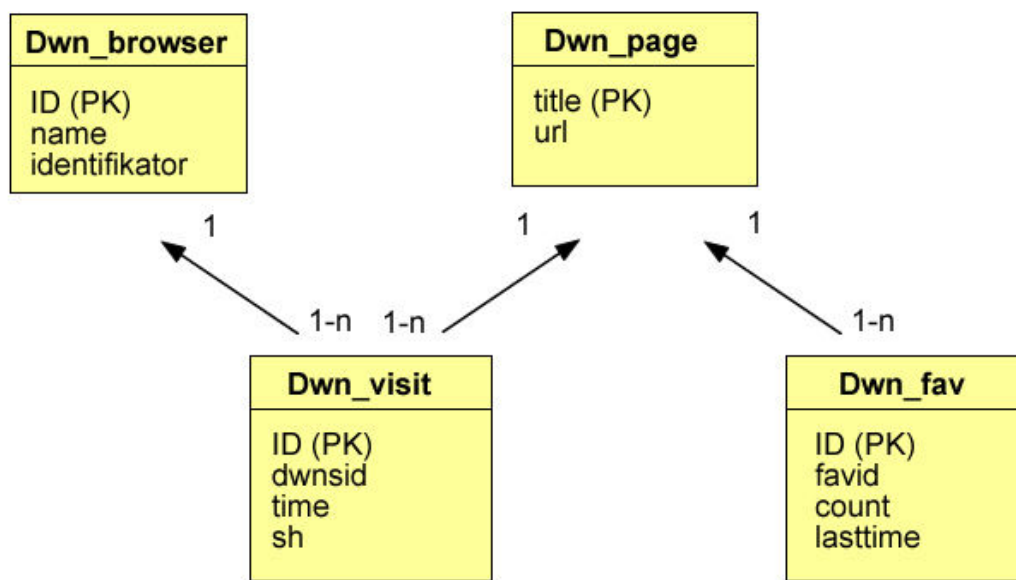
### **3.4.7 Počet online návštěvníků**

Tato statistika zobrazuje aktuální počet návštěvníků pohybujících se po stránkách prezentace. Problém vzniká s rozpoznáním, kdy už návštěvník opustil prezentaci. Tento problém lze vyřešit nastavením doby od poslední aktivity. Je-li tato doba překročena, považuje se, že návštěvník prezentaci opustil. Informace o době poslední aktivity lze zjistit z tabulky Dwn\_visit, kam se ukládá čas poslední aktivity návštěvníka.



## 3.5 Celkový návrh tabulek v databázi

Výsledné schéma tabulek v databázi je složeno z částí, které jsou využívány jednotlivými součástmi celé práce. Jsou to tabulky dynamické drobečkové navigace, tabulky pro generování oblíbených stránek dále doplněné tabulkou a dalšími rozšiřujícími atributy pro generování statistických informací. Předpona Dwn\_ je zkratkou dynamické webové navigace.



Obrázek č.6 – Celkové schéma tabulek v databázi

Při celkovém návrhu tabulek jsem hleděl na jedno zásadní kritérium. Tímto kritériem bylo minimalizovat počet přístupů do databáze při běžném přístupu návštěvníků na stránky prezentace a zefektivnit tyto přístupy pro maximální rychlost běhu celého skriptu, který se bude nejčastěji provádět. Přístup do databáze bývá časově nejnáročnější operací, a proto čím více přístupů, tím větší prodleva při zpracování stránky na serveru.

V této podobě je počet přístupů při vstupu na stránku prezentace obsahující pouze dynamickou navigaci následující. Dojde k detekci prohlížeče. Následuje dotaz na existenci stránky v tabulce Dwn\_page. Pokud stránka neexistuje, je doplněna. Pokud existuje, je rovnou vložen záznam o návštěvě do tabulky Dwn\_visit. Celkem v nepříznivém případě je databáze dotázána čtyřikrát. V příznivém, kdy je stránka již obsažena v tabulce Dwn\_page, jsou dotazy tři.

V případě, že stránka, na kterou návštěvník přistupuje, obsahuje pouze navigaci pomocí oblíbených položek, je počet přístupů následující. Zjištění existence, zda již návštěvník na této stránce byl. Pokud ne, následuje dotaz na existenci stránky v tabulce Dwn\_page. Pokud stránka zaznamenána není, je vložena a vytvořen záznam o návštěvě této stránky návštěvníkem. V případě, že stránka zaznamenána je, dojde přímo ke vložení záznamu o její návštěvě. Nejjednodušší možnost nastává v případě, že již návštěvník stránku navštívil. Dojde tak jen k detekci a zvýšení čítače znázorňujícího

počet návštěv této stránky návštěvníkem. Výsledně tak v nejhorším případě přistupujeme k databázi čtyřikrát, v nejlepší případě pouze dvakrát. V případě prvního přístupu ke stránkám prezentace, které jsou již ale vloženy v tabulce, je databáze dotázána třikrát.

Vzhledem k potřebě mít uloženy jednotlivé stránky v databázi slouží tabulka Dwn\_page jak dynamické navigaci, tak oblíbeným stránkám uživatelů. Její položky se během několika prvních návštěv webové prezentace zaplní všemi stránkami prezentace a eliminují se tak nepříznivé stavy v počtu dotazů na databázi.

### **3.5.1 Životní cyklus položek uložených v tabulkách**

Životním cyklem je v případě položek tabulek v databázi jejich vytvoření a mazání. Důležitou částí je právě mazání těchto položek. Předchází se tím zahlcení databáze a enormnímu nárůstu její velikosti.

Položky v tabulce Dwn\_visit vznikají při používání dynamické drobečkové navigace. Jejich zánik je však mírně komplikován nutností uchovávat informace pro získání statistik. Proto je mazání položek této tabulky plně v kompetenci správce webové prezentace. Ten musí určit, ve kterém časovém období již statistiky nepotřebuje a položky mohou být smazány. V případě, že návštěvník využije vlastnosti dynamické navigace k přechodu na již dříve navštívenou stránku, jsou stránky před požadovanou stránkou, na kterou má být návrat uskutečněn, ze seznamu odstraněny. Jak jsem již vysvětloval, nemohou být však odstraněny z databáze. Proto jsem přidal atribut sh (od slova show) každému záznamu v tabulce. U stránek před požadovanou stránkou k návratu je tato hodnota změněna a stránky se již v dynamické navigaci nezobrazují.

Položky v tabulce Dwn\_fav vznikají ukládáním počtu zobrazení jednotlivých stránek návštěvníky. Jejich mazání probíhá po uplynutí časového kvanta od poslední návštěvy automaticky. K tomu slouží atribut lasttime, ve kterém je uložen čas poslední návštěvy.

Položky v tabulce Dwn\_page obsahují titulky a URL všech stránek prezentace, které byly navštíveny návštěvníky webové prezentace. Jsou na ně odkázány záznamy v tabulkách Dwn\_visit a Dwn\_fav. Smazáním těchto položek by došlo k porušení referenční integrity mezi primárními klíči a cizími klíči. Počet položek tabulky je stejný s počtem titulkem rozlišených stránek prezentace. Pro vyřešení problémů s referenční integritou jsem se rozhodl při mazání této tabulky ke smazání a znovuvytvoření celého systému tabulek. Při tomto kroku jsou smazány všechny záznamy.

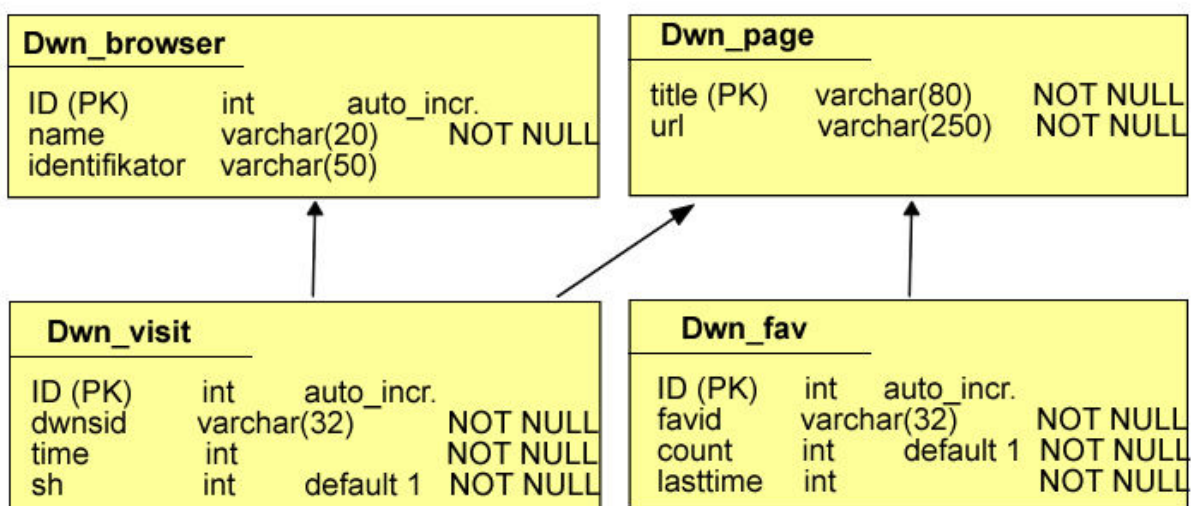
V tabulce Dwn\_browser jsou uloženy identifikační řetězce prohlížečů. Tato tabulka je naplněna při svém vzniku a dále se do ní nic nepřidává. Z tohoto důvodu jí není potřeba samostatně mazat.

## 4 Implementace

Celá práce je implementována ve skriptovacím jazyce PHP a jako úložiště dat je použita relační databáze MySQL. Tyto technologie jsou nejrozšířenějším řešením dynamických webových prezentací. Jednotlivé části řešení jsou implementovány jako soubory obsahující skripty a funkce.

### 4.1 Tabulky v databázi

Prvním krokem implementace celého řešení je vytvoření a zpřístupnění tabulek v databázi. Součástí implementace tabulek je stanovení datových typů atributů tabulek, jejich omezení velikosti a nastavení vazeb mezi tabulkami pomocí primárních a cizích klíčů.



Obrázek č.7 – Implementační schéma tabulek v databázi

Z implementačního schéma tabulek na obrázku č.7 vyplývají omezení pro další implementaci dynamické navigace i pro zobrazování oblíbených stránek návštěvníků. Omezení se týká maximální délky titulku, kterou jsem zvolil na 80 znaků, a maximální délky URL, kterou jsem zvolil 250 znaků. Výhoda použití datového typu varchar je velikost uložených dat. Uložený řetězec zabírá v paměti databáze jen tolik znaků, kolik obsahuje plus jeden.

Pro časové veličiny je zvolen celočíselný datový typ integer. Čas je ukládán jako počet sekund uplynulých od Ledna 1 1970 00:00:00 GMT. Tento čas je také označován jako začátek epochy systému UNIX.

Databáze MySQL podporuje použití cizích klíčů až od verze 5, která stále ještě není natolik rozšířena. Z tohoto důvodu jsem se rozhodl implementovat referenční integritu vlastním způsobem. Cizí klíče jsou standardně vloženy jako další atributy tabulek Dwn\_visit, Dwn\_page a odkazují na

primární klíče tabulek dle vztahu mezi tabulkami. Referenční integrita je hlídána v následujících implementacích jednotlivých navigací.

Názvy tabulek jsou záměrně voleny s předponou Dwn\_, aby byly na první pohled rozlišeny tabulky dynamické webové navigace od ostatních tabulek nacházejících se v databázi.

## 4.2 Implementace dynamické navigace

Jednotlivé součásti implementace drobečkové dynamické navigace jsou detekování možnosti přenosu SID pomocí cookies nebo URL, generování unikátního identifikátoru SID a funkce pro práci s dynamickou navigací.

### 4.2.1 Detekce možnosti přenosu SID

Zjišťování možnosti přenosu SID pomocí cookies probíhá v několika krocích. Skript zajišťující identifikaci návštěvníka a přenos jeho SID musí být součástí každé stránky webové prezentace a vykonává se jako první ještě před odesláním hlavičky stránky prohlížeči. Samotné uložení hodnoty na straně klienta neproběhne okamžitě po zavolání funkce setcookie, ale teprve při odeslání hlavičky webové stránky.

Prvním krokem je zjištění, zda již není SID návštěvníka známé. V takovém případě se nachází buď jako proměnná v poli GET, které obsahuje všechny proměnné přenášené přes URL, nebo je zjistitelné z cookie. Pokud není SID návštěvníka známé, vygeneruje se nová a jeho hodnota je uložena funkcí setcookie a zároveň je jeho hodnota přidána jako proměnná k URL. Hlavička webové stránky je odeslána na klientský prohlížeč pomocí funkce header, která přesměruje aktuální stránku samu na sebe. Tím se uložila hodnota SID do cookie návštěvníka a skript se začne vykonávat znovu. Důležitou poznámkou je ukončení provádění skriptu po volání funkce header funkcí exit. Nedojde-li k tomuto ukončení provádění skriptu, ten je celý dokončen a v důsledku přesměrování na tu samou stránku by byl dokončen dvakrát.

Druhý krok začíná opět zjištěním dostupnosti SID. Pokud je dostupné z cookie, podařilo se uložit cookie v prohlížeči na straně návštěvníka. V tomto případě je zbytečné přenášet SID v URL a opětovným přesměrováním stránky na sebe samu je toto SID z URL odstraněno. Pokud SID nelze zjistit z cookie ale je přeneseno v URL, došlo k situaci, že z nějakého důvodu nelze cookie uložit. V tomto případě pokračuje přenos SID pomocí URL.

Posledním krokem je v případě přenosu SID pomocí URL pokus o uložení stejného SID do URL. Po tomto kroku se již nevolá přesměrování stránky na samu sebe, ale již se nechá zpracovávat stránka dále a požadavek na uložení cookie se vyšle až po odeslání hlavičky ke klientovi. Tímto je docílen plynulý přenos SID z URL do cookie v případě, že návštěvník toto ukládání povolil.

V případě přenosu SID pomocí URL je zapotřebí toto SID přidat ke všem lokálním URL vyskytujícím se na dané stránce. Řešení je implementováno pomocí funkce `dwn_href`, která přebírá celé lokální URL jako vstupní parametr a přidá k němu SID získané z URL aktuální stránky.

## 4.2.2 Unikátní SID

Každému návštěvníkovi webové prezentace je po dobu jedné návštěvy vygenerováno unikátní SID. Požadavky na takto generované SID jsou následující. Jednou vydané SID nebude již nikdy v budoucnu znovu vydané a v jeden čas nebudou vydána dvě stejná SID.

Takový generátor lze v PHP vytvořit spojením třech funkcí. První funkce `uniqid` vygeneruje řetězec o délce 13 znaků. Tento generátor je založený na čase v mikrosekundách a tím splňuje první podmínku. V budoucnosti již nikdy nebude vytvořen stejný identifikátor. V jednom čase však může dojít k vygenerování dvou stejných identifikátorů. Proto je potřeba připojit k tomuto řetězci náhodné pětimístné číslo. Toto číslo získáme pomocí funkce `rand` s upřesněním rozmezí generovaných čísel od 10000 do 99999. Tato funkce již není přímo závislá na aktuálním čase, ale i na stavu hardware počítače, na kterém je spouštěna. Tímto dojde ke splnění i druhé podmínky. Zároveň je takto získaný řetězec těžko předvídatelný. Poslední funkce `md5` tento řetězec převede na jeho hash o délce 32 znaků reprezentujících hexadecimální číslo. Tento výsledný řetězec je dostatečně unikátní, velice těžko předvídatelný a pro použití jako SID dostačující.

Z použití funkce `rand` plyne jedno omezení. Od verze PHP 4.2.0 není již třeba inicializovat generátor zvlášť, proto při provozování dynamické webové navigace na starší verzi PHP není zaručena spolehlivá funkce generování unikátních SID.

Příklad vygenerovaných SID:

```
d4b392db58f294c18c4a1d8b78654ada  
dd3c42c1b0f9f1effc054b767436dc30  
d47f4ff4d9ad0d7acca22f652e8100c5  
ba313d359c3c8ab31bd6e5e57cdd5776
```

Obrázek č.8 – Příklady vygenerovaných unikátních identifikátorů

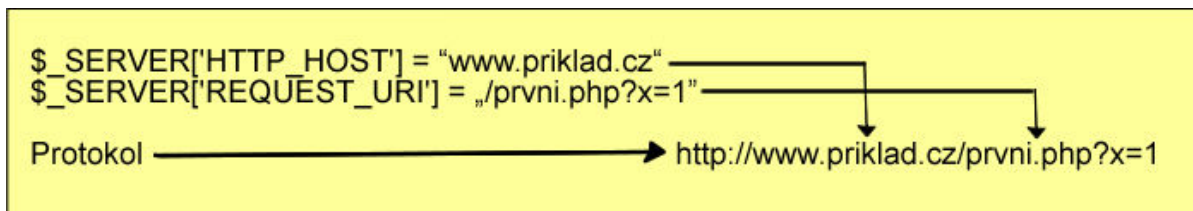
Ve skriptu je SID návštěvníka přenášeno pomocí proměnné `dwnsid`. Tímto je určeno, že SID náleží dynamické navigaci a nemůže dojít k záměně s jiným SID, které může být používáno současně.

## 4.2.3 Funkce pro zápis stránky do drobečkové navigace

Funkce pro zápis navštívené stránky do dynamické drobečkové navigace již není součástí úvodního skriptu, který zajišťuje identifikaci uživatele. Je zde pouze definovaná a volání funkce je tak v rukou programátora webové prezentace neboli uživatele a dává mu tak možnost definovat, které stránky se budou ukládat do dynamické drobečkové navigace.

Při zápisu stránky do drobečkové dynamické navigace je potřeba otevřít připojení k databázi. Tyto údaje zadává uživatel do externího konfiguračního souboru, ze kterého jsou pomocí úvodního skriptu předávány dále jako globální proměnné. Konkrétně informace pro připojení k databázi jsou souhrnně uloženy v globálním poli `dwn_connect`.

Před uložením návštěvy stránky do tabulky `dwn_visit` je provedena kontrola poslední uložené stránky návštěvníka. Pokud se získaný titulek stránky shoduje s naposledy uloženým, návštěvník se odkázal na tu samou stránku a nový zápis do databáze se neprovede. V případě, že jde o jinou stránku, je položen dotaz do databáze, zda je již stránka známá v tabulce `dwn_page`. Není-li tam, je do ní vložena. URL stránky je do databáze ukládáno v celé formě. Pro příklad uvedu toto URL `http://www.priklad.cz/prvni.php?x=1`. URL je zapotřebí poskládat z jednotlivých částí dostupných v poli `$_SERVER`.



Obrázek č.9 – Způsob složení URL

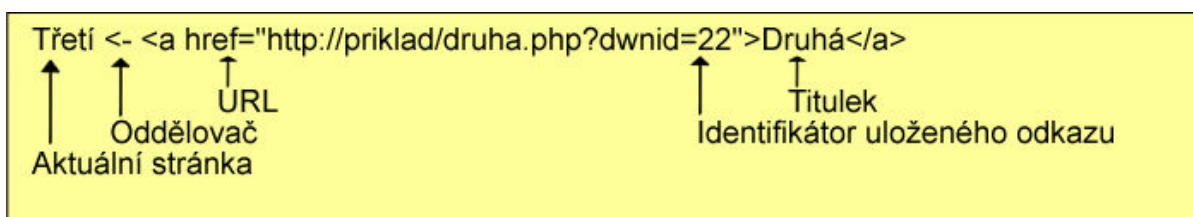
Použitý protokol může být klasický nezabezpečený protokol HTTP nebo zabezpečený HTTPS. Rozlišení protokolu je podle portu, na kterém komunikace probíhá. Nejčastěji zabezpečená komunikace probíhá na portu 443. Tento port však může být z nějakého důvodu změněn, proto je jeho skutečná hodnota brána z konfiguračního souboru, kam ji zadá uživatel. Tato hodnota je opět globální proměnnou `dwn_sslport`. Poslední částí před uložením návštěvy do tabulky je detekce prohlížeče, která je podrobně rozebrána v kapitole 3.4.1. Na konec je do tabulky `dwn_visit` uložena návštěva stránky podle titulku zadaného uživatelem, pod SID zjištěným z cookie nebo URL v aktuálním čase a provedena detekovaným prohlížečem.

## 4.2.4 Funkce pro výpis drobečkové navigace

Funkce pro výpis drobečkové navigace je vkládána do HTML kódu stránky v místě požadovaného výpisu navigace. Implementace se skládá ze dvou částí. Samostatného výběru drobečků navigace z databáze a jejich grafické reprezentace.

Výběr drobečků z databáze probíhá výběrem záznamů v tabulce `dwn_visit` doplněné o URL stránek z tabulky `dwn_page`. Záznamy jsou vybírány jen takové, které mají atribut `sh` nastaven na 1, `dwnsid` totožné s SID návštěvníka a jsou seřazeny vzestupně podle `id` záznamu. Výběr je omezen na tolik drobečků do minulosti, kolik si uživatel vypsát přeje.

Funkce vrací řetězec odkazů na jednotlivé stránky zaznamenané jako drobečky navigace. Jednotlivé drobečky jsou označeny titulky stránek, na které odkazují. Drobeček aktuální stránky, který je vypisován jako první, v seznamu není uveden jako odkaz, jelikož by odkazoval sám na sebe. K URL jednotlivých stránek je na konec vždy přidán identifikátor dwnid označující číslo záznamu v tabulce dwn\_visit. Tento identifikátor je využit při návratu návštěvníka pomocí drobečkové navigace. Součástí úvodního skriptu na začátku každé stránky je část, která detekuje právě tento identifikátor dwnid, odstraní ho z URL a zařídí nastavení atributu sh na hodnotu 0 u všech záznamů s vyšším id než toto dwnid v tabulce dwn\_visit. Tím se již nebudou ve výpisu drobečkové navigace nacházet neboli tak dojde k návratu v drobečkové navigaci. Součástí URL drobečků navigace je v případě přenosu SID pomocí URL i tento identifikátor.



Obrázek č.10 – Rozbor výpisu řetězce odkazů drobečkové navigace

Jednotlivé drobečky navigace jsou odděleny oddělovačem. Tento oddělovač je zadán jako řetězec uživatelem jako parametr funkce. Součástí řetězce může být i obrázek zapsaný v jazyce HTML.

## 4.3 Implementace oblíbených stránek

Části implementace navigace pomocí oblíbených stránek jsou identifikace návštěvníka po opětovném návratu na webovou prezentaci a při pohybu po ní, funkce pro zápis stránky do oblíbených stránek návštěvníka a funkce pro výpis oblíbených stránek. Důležitou součástí je také implementace mazání záznamů již nepoužívaných oblíbených stránek z tabulky dwn\_fav.

### 4.3.1 Implementace identifikace návštěvníka

Řešení je součástí úvodního skriptu, který se vykonává ještě před odesláním stránky prohlížeči návštěvníka. Při příchodu návštěvníka na stránku prezentace je provedena detekce jeho SID. Toto SID je přenášeno v proměnné favid, je nezávislé na SID návštěvníka drobečkové dynamické navigace, které je přenášeno v proměnné dwnsid.

Zjištění SID probíhá detekcí, je-li uloženo cookie s obsaženým identifikátorem návštěvníka. V případě, že návštěvník nebyl rozpoznán, je mu vygenerováno nové SID a uloženo do cookie za předpokladu, že je možné na straně návštěvníka cookies ukládat. Tuto informaci získáme zjištěním,

jakým způsobem je přenášeno SID dynamické drobečkové navigace z předchozí části skriptu týkající se dynamické drobečkové navigace. Doba životnosti cookie je nastavena podle předepsané doby z konfiguračního souboru, která je nastavena uživatelem. Platnost cookie je nastavena podle konfiguračního souboru. Tato nastavení se přenášejí v globálních proměnných `fav_time` a `dwn_location`. K uložení cookie dojde při odeslání hlavičky stránky prohlížeči návštěvníka při přesměrování stránky na sebe samu. Identifikátor SID je generován stejným způsobem jako identifikátor dynamické drobečkové navigace. Přesný postup generování identifikátoru je popsán v kapitole 3.2.2.

V případě, že návštěvník byl rozpoznán, je mu prodloužena doba životnosti cookie opět dle nastaveného času z konfiguračního souboru.

### **4.3.2 Funkce pro zápis stránky mezi oblíbené**

Funkce pracuje s informacemi závislými na programátorovi webové aplikace, které jsou zadány v konfiguračním souboru a přístupné jsou jako globální proměnné. Globální proměnná typu pole `dwn_connect` obsahuje informace pro připojení k databázi.

Po připojení k databázi funkce zjišťuje titulek naposledy přidané návštěvy stránky do tabulky `dwn_fav`. Získaný titulek porovná se zadaným titulkem aktuální stránky. V případě, že jsou titulky shodné, nový záznam se neukládá a funkce nic neprovádí. V opačném případě je položen dotaz do tabulky `dwn_fav`, zda již obsahuje záznam o počtu zobrazení dané stránky návštěvníkem. Pokud záznam obsahuje, je získán počet zobrazení v atributu `count`, zvýšen o jedna a opět uložen do tabulky `dwn_fav`. V případě, že záznam zatím neexistuje, je zjištěno, zda stránka je již zapsána v tabulce `dwn_page`. Není-li tam záznam stránky uložen, je přidán nový obsahující titulek a celou URL stránky. URL stránky je zjišťována pomocí spojení informací uložených v poli `$_SERVER` a pomocí detekce portu, na kterém komunikace probíhá, je zvolen správný protokol HTTP nebo HTTPS. Protokol pro zabezpečenou komunikaci je získán z globální proměnné `dwn_sslport` z konfiguračního souboru. Po uložení stránky do tabulky `dwn_page` je uložen i záznam do tabulky `dwn_fav` s SID návštěvníka a aktuální hodnotou času.

Funkce je implementována jako součást úvodního skriptu, její volání je však v režii programátora webové aplikace a dává mu tak možnost definovat, které stránky se budou ukládat jako oblíbené stránky návštěvníků.

### **4.3.3 Funkce pro výpis oblíbených stránek**

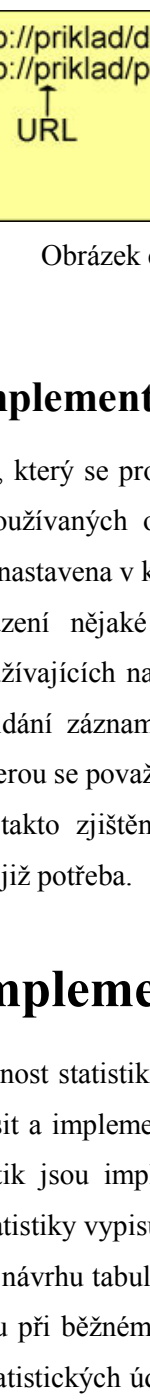
Funkce pro výpis oblíbených stránek návštěvníka se skládá ze dvou částí. Zjištění oblíbených stránek návštěvníka a jejich výpis. Samotnou funkci zavolá programátor webové aplikace v místě HTML kódu stránky, ve kterém si přeje mít vypsané.



Zjištění oblíbených stránek návštěvníka je implementováno dotazem do databáze na propojené tabulky `dwn_fav` a `dwn_page` pro doplnění titulku a URL oblíbených stránek. Jsou vybrány takové záznamy, u kterých odpovídá SID návštěvníka, a seřazeny podle počtu zobrazení. Počet vybraných oblíbených stránek je limitován zadaným parametrem.

Pro výpis je vytvořen řetězec odkazů na jednotlivé stránky seřazený podle oblíbenosti od nejoblíbenějších po nejméně oblíbené. K textu odkazů, který je tvořen titulkem odkazované stránky, je přidána hodnota znázorňující počet návštěv dané stránky.

```
<a href="http://priklad/druha.php">Druhá (25x)</a>  
<a href="http://priklad/prvni.php">Prvni str.(5x)</a>...
```



Obrázek č.11 – Rozbor výpisu řetězce odkazů drobečkové navigace

### 4.3.4 Implementace mazání uložených oblíbených stránek

Součástí skriptu, který se provádí na začátku zobrazení každé stránky, je i část, která implementuje mazání již nepoužívaných oblíbených stránek. Doba, za kterou se považují uložené stránky za nepoužívané, je nastavena v konfiguračním skriptu a je získána z globální proměnné `fav_time`.

Při zobrazení nějaké stránky jakýmkoliv návštěvníkem dojde k vybraní takových SID návštěvníků využívajících navigaci pomocí oblíbených stránek z tabulky `dwn_fav`, u kterých je čas nejnovějšího přidání záznamu o návštěvě nějaké stránky starší než doba od současnosti snižená o hodnotu, za kterou se považuje navigace pomocí oblíbených stránek za nevyužívanou.

U všech takto zjištěných SID dojde ke smazání všech záznamů v tabulce `dwn_fav`. Tyto záznamy nejsou již potřeba.

## 4.4 Implementace statistik

Pro plnou funkčnost statistik je zapotřebí implementovat detekci prohlížeče při ukládání návštěv do tabulky `dwn_visit` a implementovat všechny statistiky jako jednotlivé funkce. Funkce pro výpočet a zobrazení statistik jsou implementovány v soboru `dwn_stat_function`, který je vkládán pouze do stránek, které statistiky vypisují.

Z důvodu návrhu tabulek pro jednoduché ukládání záznamů a co nejmenší zátěže prováděného úvodního skriptu při běžném provozu webové prezentace jsou dotazy na statistické údaje složitější. Pro získávání statistických údajů jsou často použity vnořené dotazy. Tento typ dotazů je podporován

databázi MySQL až od verze 4.1.0. Z tohoto důvodu se stává verze 4.1.0 MySQL minimální podporovanou.

Každá funkce přistupuje k databázi, proto je vždy na začátku provedeno připojení k databázi s využitím předem nastavených informací o připojení získaných z konfiguračního souboru. Tyto informace jsou ve funkcích přístupné z globálního pole `dwn_connect`.

#### 4.4.1 Detekce prohlížeče

Každý prohlížeč, kterým návštěvník přistupuje na stránky webové prezentace, se identifikuje řetězcem. Tento řetězec je v jazyce PHP přístupný v proměnné `$_SERVER['HTTP_USER_AGENT']`. Součástí řetězce jsou informace o názvu prohlížeče, jeho verzi a platformě, na které běží.

Prohlížeč	Identifikační řetězec
IE 6	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; NetCaptor 7.1.0)
Firefox 2.0	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3
Opera 6	Opera/6.0 (Windows 98; U)

Obrázek č.12 – Příklady identifikačních řetězců prohlížečů

Identifikace prohlížeče probíhá porovnáním regulárního výrazu, který identifikuje danou skupinu prohlížečů s identifikačním řetězcem. Tyto regulární výrazy jsou uloženy v tabulce `dwn_browser`. K porovnání dochází přímo databázovým dotazem, jehož součástí je porovnání identifikačního řetězce prohlížeče s regulárním výrazem v databázi.

```
SELECT name FROM dwn_browser WHERE 'Opera/6.0 (Windows 98; U)'  
REGEXP identifier ORDER BY id LIMIT 0,1
```

Obrázek č.13 – Příklad dotazu s regulárním výrazem pro detekci prohlížeče

Při výběru může dojít k vybrání více prohlížečů, proto je z tohoto důvodu výběr omezen na první nalezený, který skutečně odpovídá identifikačnímu řetězci. K vybrání více prohlížečů může dojít, protože je například často součástí řetězce identifikujícího prohlížeč Opera i text obsahující `msie 5.0`. Tento text je tedy rozpoznán i jako prohlížeč Internet Explorer 5.0. Z tohoto důvodu záleží na pořadí ukládaných prohlížečů do databáze. První se ukládají prohlížeče Opera, druhá skupina jsou prohlížeče Microsoft Internet Explorer, třetí skupinou jsou všechny další prohlížeče mimo vyjmenované, čtvrtou skupinou jsou prohlížeče Firefox a poslední skupinou jsou prohlížeče Mozilla. Celý výpis podporovaných prohlížečů i s regulárními výrazy je uveden v příloze.

Prohlížeč	Identifikační regulární výraz
IE 6	msie.6
Firefox 2.0	mozilla/5.*firefox/2\0
Opera 6	opera.6

Obrázek č.14 – Příklady regulárních výrazů identifikujících prohlížeče

## 4.4.2 Funkce pro získání statistických údajů

Každá funkce pro zpracování statistik má dva základní parametry a tím je časové razítko doby, od které požadujeme statistiku, a časové razítko doby, po kterou požadujeme danou statistiku. Časové razítko je počet sekund od doby tzv. Unix Epoch (0:00:00 Leden 1, 1970 GMT). Všechny funkce pracují s tabulkami dwn\_visit a dwn\_page.

### 4.4.2.1 Statistika použitých prohlížečů

Funkce provede výběr počtu všech průchodů stránkami provedených návštěvníky ve vybraném časovém období. Dále vybírá počty průchodů stránkami dle jednotlivých prohlížečů za stejné časové období. Z těchto hodnot se vypočítá procentuální zastoupení prohlížeče na všech návštěvách webové prezentace. Funkce vrací řetězec obsahující skupiny jednotlivých údajů oddělené mezi sebou středníkem a jednotlivé údaje v těchto skupinách oddělené čárkou v pořadí jméno prohlížeče, počet přístupů tímto prohlížečem a procentuální zastoupení ze všech přístupů. Prohlížeče jsou vráceny v řetězci seřazeny od nejčastějšího po nejméně používaný. Počet takto vypisovaných skupin údajů je omezen parametrem funkce udávající počet zobrazovaných prohlížečů.

```
SELECT browser_name,COUNT(browser_name) FROM dwn_visit
WHERE time IN (SELECT time FROM dwn_visit GROUP BY dwnsid,browser_name)
AND time BETWEEN $time_start AND $time_stop GROUP BY browser_name
LIMIT 0,$pocet_stranek
```

Obrázek č.15 – SQL dotaz na jméno prohlížeče a počet jím uskutečněných návštěv webové prezentace

### 4.4.2.2 Průměrný čas strávený na stránkách návštěvníky

Funkce zjišťuje časy přístupů na první stránky a časy přístupů na poslední stránky zaznamenané v průchodu stránkami. Z rozdílů těchto časů je počítána průměrná doba.

```
SELECT MIN(time),MAX(time),dwnsid FROM dwn_visit
WHERE time BETWEEN $time_start AND $time_stop GROUP BY dwnsid
```

Obrázek č.16 – SQL dotaz na čas zobrazení první stránky a čas zobrazení poslední navštívené stránky každého průchodu

#### 4.4.2.3 Nejčastější vstupní a výstupní stránky webové prezentace

Jedná se o dvě samostatné funkce. Jejich princip je však shodný až na vybíranou stránku. Obě zjišťují počet průchodů prezentací, ale jedna vybírá vstupní stránky a druhá poslední navštívené stránky. Funkce vracejí řetězec obsahující skupiny jednotlivých údajů oddělené mezi sebou středníkem a jednotlivé údaje v těchto skupinách oddělené čárkou v pořadí stránky, počet přístupů a procentuální zastoupení ze všech přístupů. Obě funkce jsou také omezeny v počtu vypisovaných stránek vstupním parametrem.

#### 4.4.2.4 Celkový počet návštěv

Funkce počítá počet průchodů webovou prezentací. Jinými slovy řečeno počet návštěv webové prezentace.

```
SELECT COUNT(*) FROM dwn_visit
WHERE time IN (select time from dwn_visit group by dwnsid,browser_name)
AND time BETWEEN $time_start AND $time_stop GROUP BY browser_name
```

Obrázek č.17 – SQL dotaz na počet průchodů webovou prezentací v daném období

#### 4.4.2.5 Nejoblíbenější stránky

Funkce počítá počet zobrazení každé stránky a procentuálně je srovnává s celkovým počtem zobrazených stránek. Funkce vrací řetězec obsahující skupiny jednotlivých údajů oddělené mezi sebou středníkem a jednotlivé údaje v těchto skupinách oddělené čárkou v pořadí titulek stránky, počet přístupů na ni a procentuální zastoupení ze všech přístupů.

```
SELECT dwn_visit.page_title,COUNT(dwn_visit.page_title),dwn_page.url FROM
dwn_visit,dwn_page
WHERE dwn_visit.page_title=dwn_page.title AND time BETWEEN $time_start AND
$time_stop
GROUP BY page_title
ORDER BY COUNT(dwn_visit.page_title) DESC LIMIT 0,$pocet_stranek
```

Obrázek č.18 – SQL dotaz na nejoblíbenější stránky a počty jejich návštěv

#### 4.4.2.6 Počet zobrazených stránek

Funkce počítá celkový počet zobrazených stránek.

```
SELECT COUNT(page_title) FROM dwn_visit  
WHERE time BETWEEN $time_start AND $time_stop  
GROUP browser_name
```

Obrázek č.19 – SQL dotaz na počet zobrazených stránek

#### 4.4.2.7 Počet online návštěvníků

Funkce získává identifikátory návštěvníků, jejichž čas zobrazení poslední stránky nepřesáhl mez stanovenou vstupním parametrem. Počet online návštěvníků je totožný s počtem identifikátorů.

```
SELECT id FROM dwn_visit  
WHERE time > aktualni_cas - max_doba_necinnosti  
GROUP BY dwnsid
```

Obrázek č.20 – SQL dotaz na identifikátory návštěvníků nacházejících se na webové prezentaci

### 4.4.3 Likvidace dat pro statistiky

Pro mazání dat již nepotřebných ke zpracovávání statistik je implementována funkce, která maže všechny záznamy v tabulce dwn\_visit v zadaném období pomocí časových razítek. Tato funkce je také součástí souboru se statistickými funkcemi.

## 4.5 Rozdělení do souborů

Celé řešení je implementováno v šesti souborech. Každý soubor obsahuje skripty nebo funkce významově a v možnostech použití patřící k sobě.

### 4.5.1 Dwn\_start.php

Soubor dwn\_start.php obsahuje úvodní skript vykonávaný jako první při přístupu na jakoukoliv stránku webové prezentace. Úvodní skript je složen z části obstarávající přenos a detekci SID dynamické drobečkové navigace a stará se o přenos SID i navigace pomocí oblíbených stránek, dále obsahuje skript obsluhující návrat v drobečkové navigaci a mazání starých záznamů oblíbených stránek. Jeho součástí je i propojení se souborem dwn\_function. Toto propojení zajišťuje přístupnost funkcí pro práci s drobečkovou navigací i oblíbenými stránkami na všech stránkách prezentace.

## **4.5.2 Dwn\_function.php**

Soubor `dwn_function` obsahuje funkce pro práci s dynamickou drobečkovou navigací i s oblíbenými stránkami. Obsahuje i funkci na doplnění SID k URL a funkci k inicializaci tabulek.

## **4.5.3 Dwn\_stat\_function.php**

Soubor `dwn_stat_function` obsahuje všechny funkce pro práci se statistickými údaji o pohybu návštěvníků po stránkách prezentace. Proto aby bylo možné používat tyto funkce, musí být soubor připojen ke stránce, na které budou funkce volány.

## **4.5.4 Dwn\_createtable.php**

Soubor `dwn_createtable` obsahuje skript pro vytvoření a inicializaci tabulek.

## **4.5.5 Dwn\_deletetable.php**

Soubor `dwn_deletetable` obsahuje skript likvidující všechny záznamy a tabulky z databáze.

## **4.5.6 Dwn\_config.php**

Soubor `dwn_config` obsahuje konfigurační proměnné systému pro identifikaci návštěvníka, dynamické navigace i oblíbených stránek. Ve všech ostatních souborech dynamické webové navigace je vloženo propojení na tento soubor, proto informace z tohoto souboru jsou ve všech skriptech dostupné jako globální proměnné. Pro plnou funkčnost musí být správně vyplněn programátorem webové aplikace. Jediný soubor vyžadující změny.

## 5 Závěr

V posledních dvou letech jsem se začal vážně zabývat tvorbou dynamických webových prezentací budovaných s použitím jazyka PHP a databázového systému MySQL. Při návrhu této práce a její implementaci jsem z menší části využil již získaných zkušeností, nicméně z větší části mi práce byla velkým přínosem v oblasti poznání možností jazyka PHP a databáze MySQL. Při implementaci dynamických navigací a celého systému práce s tabulkami v databázi jsem byl často nucen navrátit se na začátek a znovu přehodnotit celý návrh. Odměnou pak byl jednodušší a méně náročný skript.

Výsledná práce je zcela funkční a připravená k použití a začlenění do webových prezentací. Přináší tak možnost jednoduchého začlenění dynamické drobečkové navigace, navigace pomocí oblíbených stránek a zjišťování zajímavých statistik o pohybu návštěvníků po webových prezentacích založených na technologiích PHP a MySQL. Je tedy velkým přínosem hlavně pro programátory webových prezentací, kteří mají zájem o rozšíření prezentací o tyto možnosti. Výsledná implementace splňuje jak celý návrh, tak i všechny body zadání práce.

Další vývoj této práce vidím v první řadě v doplnění dalších možností doplňkových navigací. Vznikl by tak komplexní nástroj, který by byl jednoduše vložitelný do každého řešení webové prezentace a zajišťoval by veškeré možnosti doplňkových navigací. Výběr použití jednotlivých navigací by byl již na programátorovi webové prezentace. Konkrétně další doplňkovou navigací by mohla být statická drobečková navigace. Bylo by potřeba přebudovat správu a doplnit tabulku obsahující seznam stránek webové prezentace a navrhnout systém rozdělení webových prezentací do úrovní. Další vývoj na této práci by mohl být také zaměřen na ještě lepší a dokonalejší optimalizaci kódu a přístupů k informacím v tabulkách databáze.

Doufám, že tato práce bude přínosem i návštěvníkům webových prezentací, kterým umožní lepší přehled a přístup k požadovaným informacím. Tím také dojde ke zvýšení návštěvnosti webových prezentací s těmito možnostmi navigace a skutečnému zhodnocení výsledků této práce.

# Literatura

- [1] GILFILLAN, I. *Myslíme v MySQL4*. Grada, 2003.
- [2] GILMORE, W. Jason. *Velká kniha PHP a MySQL*. Zoner Press, 2005.
- [3] NARAMORE, E., GERNER, J., SCOUARNEC, Y.L., STOLZ, J. a GLASS, M.K. *PHP5, MySQL, Apache : Vytváříme webové aplikace*. Computer Press, 2006.
- [4] WELLING, L. a THOMSONOVÁ, L. *PHP a MySQL : rozvoj webových aplikací*. 2. vyd. SOFTPRESS, 2003.
- [5] WILLIAMS, H.E. a LANE, D. *PHP a MySQL : Vytváříme webové databázové aplikace*. Computer Press, 2002.
- [6] [cs.wikipedia.org](http://cs.wikipedia.org), 9.5.2007.
- [7] [www.php.net](http://www.php.net), 30.4.2007.
- [8] [www.interval.cz/vyvoj-aplikaci/php/](http://www.interval.cz/vyvoj-aplikaci/php/), 25.4.2007.



# Seznam příloh

Příloha 1. Manuál

Příloha 2. Rozlišované prohlížeče

Příloha 3. CD

# **Přílohy**

# **Příloha 1**

## **Manuál**

# Obsah

Manuál .....	2
1 Požadavky .....	2
2 Soubory .....	2
3 Konfigurační soubor.....	3
4 Inicializace databáze.....	4
5 Drobečková navigace .....	5
6 Oblíbené stránky .....	6
7 Statistiky.....	7
8 Likvidace dynamické navigace .....	9
9 Příklad použití .....	10

# Manuál

Tento manuál slouží jako příručka programátora k začlenění dynamické webové navigace do webových prezentací. Vysvětluje instalaci a užívání jednotlivých funkcí.

## 1 Požadavky

Minimální požadavky na používanou verzi jazyka PHP a databáze MySQL pro provoz dynamické webové navigace.

- Verze PHP 4.2.0 a vyšší.
- Verze MySQL 4.1.0 a vyšší.

Pro správnou funkčnost je také nutné rozlišovat jednotlivé stránky titulkem. Nejlépe již při návrhu webové prezentace mít na zřeteli tento požadavek, nicméně nic nebrání dodatečnému rozlišení všech stránek prezentace pomocí titulku. Titulek by měl jednoznačně a výstižně identifikovat stránku.

## 2 Soubory

Všechny soubory dynamické webové navigace jsou ve složce s názvem dwn. Tuto složku je nutno přesunout ke složkám webové prezentace na webovém serveru. Zde najdete stručný popis a funkci jednotlivých souborů. Jejich použití a začlenění do stránek prezentace bude vysvětleno v dalších kapitolách.

- **dwn\_start.php** - Soubor obsahuje skript, který se provádí na začátku zpracování každé stránky. Musí být připojen na začátek každé stránky webové prezentace. Jeho součástí je i propojení se souborem obsahujícím těla všech funkcí pro práci s drobečkovou navigací a navigací pomocí oblíbených stránek. Tím jsou tyto funkce přístupné na každé stránce prezentace.
- **dwn\_function.php** – Soubor obsahuje funkce pro práci s dynamickou drobečkovou navigací a oblíbenými stránkami.
- **dwn\_stat\_function.php** – Obsahem souboru jsou funkce pro práci a získání statistických údajů o pohybu návštěvníků webových prezentací. Při požadavku na práci

s těmito funkcemi na nějaké stránce prezentace musí být tento soubor připojen k této stránce.

- **dwn\_config.php** – Konfigurační soubor dynamické webové navigace. Jediný soubor, který vyžaduje úpravu. Vypĺňují se v něm důležité informace pro fungování celé dynamické navigace.
- **dwn\_createtable.php** – Soubor obsahuje důležitý skript pro inicializaci tabulek v databázi. Tento skript je potřeba provést jako první před spuštěním webové prezentace s dynamickou webovou navigací.
- **dwn\_deletetable.php** – Soubor obsahuje skript pro likvidaci tabulek dynamické webové navigace v případě odinstalace.

## 3 Konfigurační soubor

Prvním krokem začlenění dynamické webové navigace do webové prezentace je nastavení všech proměnných v konfiguračním skriptu. V následujících řádcích budete seznámeni se všemi proměnnými a doporučeným způsobem jejich nastavení. Všechny proměnné musejí být ukončeny středníkem.

- **\$dwn\_location** – Hodnota této proměnné určuje lokalitu platnosti dynamické webové navigace. Součástí jedné webové prezentace může být více dynamických navigací, každá pro určitou část prezentace. Jednotlivé části jsou rozděleny podle souborů.

Příklad nastavení:

`$dwn_location="/";` - pro nastavení platnosti na celou webovou prezentaci

`$dwn_location="/bp";` - pro část webového prostoru `www.example.cz/bp`

- **\$dwn\_sslport** – Zde nastavujete port, na kterém probíhá zabezpečená komunikace mezi serverem, na kterém poběží prezentace, a klienty. Podle tohoto portu bude rozlišována zabezpečená komunikace od nezabezpečené.

Příklad nastavení:

`$dwn_sslport=443;` - nejčastěji je využíván port 443, ale může být administrátorem serveru změněn

- **\$dwn\_host** – Zde se vyplňuje URL adresa databázového serveru.

Příklad nastavení:

`$dwn_host="mysql.example.cz";`

- **\$dwn\_user** – Vyplňte uživatelské jméno pro připojení k databázi.

Příklad nastavení:

`$dwn_user="root";`

- **\$dwn\_password** – Zde zadejte heslo pro připojení k databázi.  
Příklad nastavení:  
`$dwn_password="heslo";`
- **\$dwn\_database** – Vyplňte název databáze, ke které se připojujete.  
Příklad nastavení:  
`$dwn_database="nazev_db";`
- **\$fav\_time** – Zde je nutné vyplnit dobu, po jejímž uplynutí od poslední návštěvy návštěvníka webové prezentace má dojít ke smazání jeho záznamů oblíbených stránek. Doba se udává v počtu dní.  
Příklad nastavení:  
`$fav_time=30;` - po uplynutí třiceti dní od poslední návštěvy návštěvníka jsou jeho oblíbené stránky smazány

Po řádném vyplnění konfiguračního souboru jej uložte. Do ostatních souborů nezasahujte.

## 4 Inicializace databáze

Druhým krokem nutným pro správnou funkčnost dynamické webové navigace je inicializace tabulek v databázi. Inicializaci provedeme spuštěním skriptu `dwn_createtable.php` přes webový prohlížeč. Při správném nastavení konfiguračního souboru `dwn_config.php` se v prohlížeči objeví „OK“. Jinak se vypíše chybové hlášení. Skript provede vytvoření a inicializaci tabulek v databázi.



Obrázek č.1 – Úspěšné provedení inicializace tabulek

Stejnou práci odvede i funkce `dwn_initdatabase()`, která je přístupná po připojení souboru `dwn_start` ke stránce, na které je volána. V případě, že již databáze nějaké tabulky obsahuje, jsou smazány a znovu vytvořeny. Tuto funkci můžete zakomponovat do webového rozhraní pro správu celé webové prezentace.

## 5 Drobečková navigace

Před použitím funkcí pro práci s drobečkovou dynamickou navigací je zapotřebí ke každé stránce webové prezentace připojit soubor `dwn_start.php`. Tento soubor vkládejte tak, aby se při zpracování stránky provedl jako první ještě před odesláním hlavičky stránky prohlížeči.

Další nutnou podmínkou funkčnosti drobečkové navigace je úprava všech lokálních URL odkazů a odkazů na stránky zpracovávající údaje z formulářů, které se vyskytují na stránkách prezentace. K této úpravě použijte funkci `dwn_href(url)`, jejímž jediným parametrem je řetězec obsahující URL. Příklad použití je na obrázku č.2.

```
<?
include "dwn/dwn_start.php"; //připojení úvodního skriptu
?>

//příklad úpravy lokální URL v odkazu
<a href="<? echo dwn_href("../treti.php?id=123")?>">Třetí</a>

//příklad úpravy lokální URL odkazující na stránku zpracování formuláře
<form id="form1" name="form1" method="post"
      action="<? echo dwn_href("prvni.php"); ?> ">
  <input type="submit" name="Submit" value="Odeslat" />
</form>
```

Obrázek č.2 – Úprava stránek prezentace pro drobečkovou navigaci

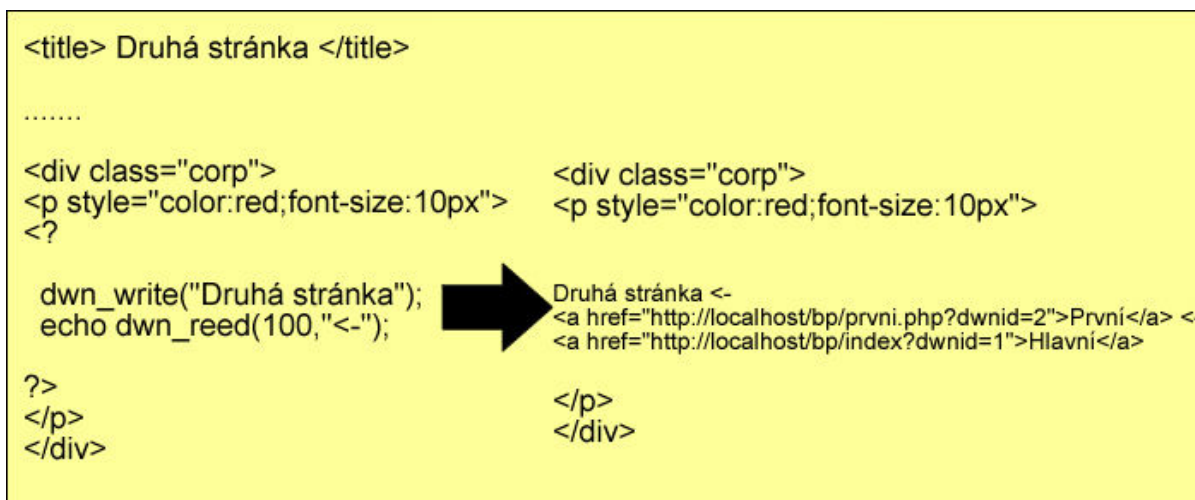
Po těchto úpravách již můžeme přistoupit k funkcím dynamické drobečkové navigace. Funkce jsou dvě `dwn_write` a `dwn_reed`. První funkce zajišťuje zápis do drobečkové navigace a druhá je pro výpis drobečkové navigace.

Funkce `dwn_write(titulek)` zapisuje aktuální stránku při jejím zobrazení do drobečkové navigace návštěvníka. Pokud z jakéhokoliv důvodu nechcete stránku zapisovat do navigace, tuto funkci na stránce nevolejte. Jediným parametrem funkce je řetězec obsahující titulek stránky. Tuto funkci je nutné volat dříve než funkci `dwn_reed`, nebo bude navigace vypsána bez aktuální stránky. Funkce nic nevrací.

Zavoláním funkce `dwn_reed(počet, oddělovač)` získáme řetězec obsahující drobečky navigace ve formě odkazů na jednotlivé stránky. Tento řetězec lze vypsát do stránky například pomocí funkce `echo`. Formátování a grafickou úpravu dynamické drobečkové navigace ve stránce lze řídit pomocí kaskádových stylů v oblasti, do které vkládáme řetězec odkazů. Parametry funkce jsou maximální počet vypisovaných drobečků navigace a oddělovač. Oddělovač je použit jako oddělení mezi drobečky jednotlivých stránek v navigaci. Může být naplněn i řetězcem obsahujícím HTML kód



obrázku, který je pak vkládán jako oddělovač drobečků navigace. Levá část obrázku znázorňuje kód stránky webové prezentace, který se zpracovává na serveru, a pravá část obrázku je kód zaslaný prohlížeči návštěvníka.



Obrázek č.3 – Příklad použití funkcí pro dynamickou drobečkovou navigaci

## 6 Oblíbené stránky

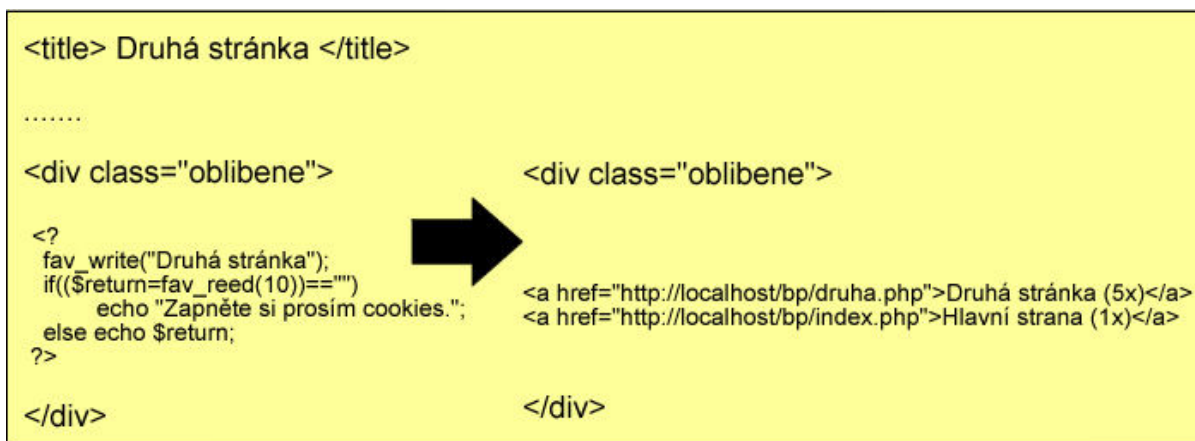
Pokud ještě nemáte ke každé stránce připojený soubor `dwn_start.php` z předešlé kapitoly, připojte jej nyní. Bez úvodního skriptu nebudou oblíbené stránky fungovat ani nebudou jednotlivým stránkám přístupné funkce pro práci s oblíbenými stránkami návštěvníků.

Tyto funkce jsou opět dvě, `fav_wite` a `fav_reed`. První provádí zápis stránky mezi oblíbené a druhá výpis těchto stránek.

Funkce `fav_write(titulek)` musí být volána před funkcí `fav_reed` a jejím parametrem je titulek stránky. Funkci volejte na stránkách, u nichž si přejete jejich zapsání k oblíbeným stránkám návštěvníka. Funkce nic nevrací.

Parametrem funkce `fav_reed(počet)` je pouze počet zobrazovaných oblíbených stránek. Funkce vrací řetězec obsahující odkazy na jednotlivé oblíbené stránky seřazené od nejoblíbenější. Texty odkazů jsou tvořeny titulkem odkazované stránky a počtem přístupů na tuto stránku. Formátování a grafická úprava odkazů je možná pomocí kaskádových stylů. V případě, že není povoleno ukládání cookies na straně návštěvníka funkce, vrací prázdný řetězec, na který je dále možno reagovat například vypsáním žádosti na povolení zapisování do cookies.

Na obrázku č.4 si můžete prohlédnout jednu z možností volání funkcí oblíbených stránek. V levé části obrázku je kód stránky webové prezentace, který se zpracovává na serveru, a pravá část obrázku je kód zaslaný prohlížeči návštěvníka. V externích kaskádových stylech je definována třída „oblíbené“, pomocí které jsou formátovány odkazy.



Obrázek č.4 – Příklad použití funkcí pro oblíbené stránky a vygenerovaného HTML kódu

## 7 Statistiky

Pro použití je nutné ke stránce připojit soubor `dwn_stat_function.php`. Tento soubor obsahuje všechny funkce pro práci a získání statistik. Následující statistické funkce vypisují řetězce, ve kterých jsou řádky odděleny středníkem a jednotlivé informace čárkou. Formátování lze tedy provést různými způsoby nahrazením středníků a čárek za potřebný HTML kód. Vstupními parametry jsou tzv. časová razítka začátku období a konce období, ze kterého chceme zjistit statistiky, udávaná jako počet sekund od doby tzv. Unix Epoch (0:00:00 Leden 1, 1970 GMT) a posledním parametrem je počet zobrazených řádků. Jazyk PHP obsahuje mnoho funkcí pro práci s časovými razítky. Například funkce `mktime(datum)` vrací počet sekund od 1.1. 1970 do zadaného data, tedy časové razítko zadaného data.

- **stat\_browser(čas\_od,čas\_do,počet)** - Vypisuje informace o použitých prohlížečích při přístupu na stránky.  
Formát: název prohlížeče, počet přístupů, procentuální zastoupení ze všech přístupů
- **stat\_startpage(čas\_od,čas\_do,počet)** - Vypisuje nejčastější vstupní stránky na webovou prezentaci.  
Formát: titulek stránky, počet přístupů, procentuální zastoupení ze všech přístupů

- **stat\_endpage(čas\_od,čas\_do,počet)** - Vypisuje nejčastější výstupní stránku návštěvníků prezentace.  
Formát: titulek stránky, počet přístupů, procentuální zastoupení ze všech přístupů
- **stat\_favpage(čas\_od,čas\_do,počet)** - Vypisuje nejoblíbenější stránky návštěvníků prezentace.  
Formát: titulek stránky, počet přístupů, procentuální zastoupení ze všech přístupů

```
//statistika používaných prohlížečů od 1.1.1970 po současnost, max 5 záznamů
stat_browser(0,time(),5);

Vypíše:                               Po úpravě:
Opera 8,5x,50%;IE6,2x,20%;.....       <table>
                                         <tr>
                                         <td>Opera 8</td><td>5x</td><td>50%</td>
                                         </tr>
                                         .....
                                         </table>
```

Obrázek č.5 – Příklad použití funkcí pro oblíbené stránky a vygenerovaného HTML kódu

Následující statistické funkce vracejí číslo. Jejich parametry jsou časová razítka začátku období a konce období, ze kterého chceme zjistit statistiky.

- **stat\_visitortime(čas\_od,čas\_do)** – Funkce vrací průměrný čas strávený návštěvníky na webové prezentaci.
- **stat\_numvisit(čas\_od,čas\_do)** – Funkce vrací celkový počet návštěv prezentace.
- **stat\_numvisitpage(čas\_od,čas\_do)** – Funkce vrací počet navštívených stránek.

Poslední statistická funkce má parametr čas v sekundách, po jehož uplynutí při nečinnosti návštěvníka se považuje za to, že odešel z webové prezentace.

- **stat\_numonline(čas)** – Funkce vrací počet návštěvníků pohybujících se po webové prezentaci v současné době.

K mazání nepotřebných záznamů, ze kterých již nebude potřeba získávat statistiky, slouží následující funkce.

- **stat\_delete(čas\_od,čas\_do)** – Maže záznamy z tabulek v databázi v zadaném časovém období. Po smazání již není možné z tohoto období získat statistické údaje.

## 8 Likvidace dynamické navigace

V případě potřeby odstranění dynamické webové navigace z webové prezentace je nutné odstranit ze všech stránek použité funkce a příložené soubory. Spuštěním skriptu pomocí webového prohlížeče ve složce dynamické navigace dwn s názvem dwn\_deletetable.php odstraníte veškeré záznamy z databáze. Dále smažte složku dwn, která obsahuje soubory dynamické navigace.

## 9 Příklad použití

První obrázek obsahuje příklad částí kódu stránky webové prezentace. Druhý zobrazuje výsledek, který se zobrazí návštěvníkovi při návštěvě stránky. Stránka obsahuje drobečkovou navigaci, hlavní menu, kde je vidět použití funkce `dwn_href`, bez které by navigace nefungovala. Dále obsahuje navigaci pomocí oblíbených stránek a statistiku přístupů podle druhů prohlížečů.

```
<?
include "bp/dwn/dwn_start.php";           //připojení úvodního skriptu
include "bp/dwn/dwn_stat_function.php"; //připojení funkcí pro statistiky
?>
<head>
<title>Treti</title>
.....
</head>
<body>
.....
<!-- ***** dynamická drobečková navigace *****-->

<div class="corp">
  <p style="color:red;font-size:10px">&gt;&gt;
    <?
      dwn_write("Třetí");           //zápis stránky do drobečkové navigace
      echo dwn_reed(100,"<-");     //vypsání drobečkové navigace
    ?>
  </p>
</div>
.....
<!-- ***** hlavní navigace *****-->

<div class="left">
  <p style="display:block;background-color:#99CCCC">Nabídka &gt;&gt;</p>
  <a href="<? echo dwn_href("bp/index.php")?>" >Hlavní strana</a>
  <a href="<? echo dwn_href("bp/prvni.php?jedna=dva&dva=tri") ?>">První</a>
  <a href="<? echo dwn_href("bp/druha.php")?>" >Druhá</a>
  <a href="<? echo dwn_href("treti.php?id=123") ?>" style="color:#FF9966">Třetí</a>
  <a href="<? echo dwn_href("bazar/ctvrta.php") ?>">Čtvrtá</a>
</div>
.....
<!-- ***** oblíbené stránky *****-->

<div class="oblíbene">
  <p style="display:block;background-color:#CCCCFF;color:#000000">Oblíbené &gt;&gt; </p>
  <?
    fav_write("Třetí");           //zápis do oblíbených stránek a výpis
    if(($return=fav_reed(10))== "") echo "Zapněte si prosím cookies.";
    else echo $return;
  ?>
</div>
.....
<!-- ***** statistiky *****-->

echo "<table><tr><td>Název prohlížeče</td><td>Poč. návštěv</td><td>% ze všech návštěv</td>";
echo "<tr><td>".str_replace(',','</td><td>',str_replace(';','</td></tr><tr><td>',stat_browser(0,time(),5)));
echo "</td></tr></table>";
</body>
```

Obrázek č.6 – Příklad kódu stránky webové prezentace s použitou dynamickou navigací

# T ř e t í

Drobečková navigace

>> Třetí <- Druhá!!!!!!!!!!!! <- První

Nabídka >>

Hlavní strana

První

Druhá

Třetí

Čtvrtá

Oblíbené >>

Třetí (3x)

První (2x)

Druhá!!!!!!!!!!!! (2x)

Čtvrtá (1x)

## Statistiky (celkové)

Název prohlížeče	Poč. návštěv	% ze všech návštěv
Firefox 2.0	2x	66.67%
IE 6	1x	33.33%

Obrázek č.7 – Příklad zobrazení stránky webové prezentace s použitou dynamickou navigací

## **Příloha 2**

### **Rozlišované prohlížeče**

# Seznam rozlišovaných prohlížečů

Název prohlížeče nebo skupiny prohlížečů	Identifikační regulární výraz
Opera 2	'opera.2'
Opera 3	'opera.3'
Opera 4	'opera.4'
Opera 5	'opera.5'
Opera 6	'opera.6'
Opera 7	'opera.7'
Opera 8	'opera.8'
Opera 9	'opera.9'
Opera (ostatní)	'opera'
IE 1	'msie.1'
IE 2	'msie.2'
IE 3	'msie.3'
IE 4	'msie.4'
IE 5	'msie.5\0'
IE 5.5	'msie.5\5'
IE 6	'msie.6'
IE 7	'msie.7'
iCab	'icab'
amaya	'amaya'
Lynx	'lynx'
Konqueror	'konqueror'
Lotus	'lotus'
Links	'links .'
Mosaic	'mosaic'
OmniWeb	'omniweb'
Safari	'safari'
AvantGo	'avantgo'
Nautilus	'nautilus'
Galeon	'galeon'
Avant browser	'avant browser'
Název prohlížeče nebo skupiny prohlížečů	Identifikační regulární výraz



<b>Název prohlížeče nebo skupiny prohlížečů</b>	<b>Identifikační regulární výraz</b>
Shiira	'Shiira'
Front page	'ms frontpage*'
ELinks	'elinks'
Nokia'	'nokia'
Phoneix'	'mozilla/5.*phoenix/0.5'
Firebird	'mozilla/5.*firebird/0.*'
Firefox 0	'mozilla/5.*firefox/0.*'
Firefox 1.0	'mozilla/5.*firefox/1\0.'
Firefox 1.4	'mozilla/5.*firefox/1\4.'
Firefox 1.5	'mozilla/5.*firefox/1\5.'
Firefox 2.0	'mozilla/5.*firefox/2\0.'
Firefox 3.0	'mozilla/5.*firefox/3\0.'
Netscape 1	'mozilla/1.*;.(I U)(; \\)'
Netscape 2	'mozilla/2.*;.(I U)(; \\)'
Netscape 3	'mozilla/3.*;.(I U)(; \\)'
Netscape 4	'mozilla/4.*;.(I U)(; \\)'
Netscape 6	'netscape.?6'
Netscape 7	'netscape.?7'
Netscape 8	'netscape.?8'
Mozilla 0.9	'mozilla/5.*rv:0.9'
Mozilla 1.0	'mozilla/5.*rv:1.0'
Mozilla 1.1	'mozilla/5.*rv:1.1'
Mozilla 1.2	'mozilla/5.*rv:1.2'
Mozilla 1.3	'mozilla/5.*rv:1.3'
Mozilla 1.4	'mozilla/5.*rv:1.4'
Mozilla 1.5	'mozilla/5.*rv:1.5'
Mozilla 1.6	'mozilla/5.*rv:1.6'
Mozilla 1.7	'mozilla/5.*rv:1.7*'
Mozilla 1.8	'mozilla/5.*rv:1.8*'
Mozilla 1.9	'mozilla/5.*rv:1.9*'
<b>Název prohlížeče nebo skupiny prohlížečů</b>	<b>Identifikační regulární výraz</b>