

**Vyšší odborná škola  
a Střední průmyslová škola elektrotechnická  
Plzeň, Koterovská 85**

## **DLOUHODOBÁ MATURITNÍ PRÁCE S OBHAJOBOU**

**Téma: Eshop s módou**

**Autor práce: Marek Hejl  
Třída: 4. I  
Vedoucí práce: Mgr. Pavel Anderle  
Dne: 31. 3. 2021**

**Hodnocení:**



Vyšší odborná škola  
a Střední průmyslová škola elektrotechnická  
Plzeň, Koterovská 85

## Zadání dlouhodobé maturitní práce

Žák: Marek Hejl  
Třída: 4.I  
Studijní obor: 18-20-M/01 Informační technologie  
Zaměření: Tvorba webových aplikací  
Školní rok: 2020–2021  
Téma práce: ***Eshop s módou***

### ***Pokyny k obsahu a rozsahu práce:***

Vytvořte e-shop na prodej oblečení a módních doplňků. Uživatel bude moci objednat výrobky z nabízeného sortimentu. Po dokončení objednávky bude uživateli na zadaný e-mail odesláno potvrzení objednávky. Administrátor bude moci přidávat nové výrobky a odebírat prodané. Po přijetí objednávky mu bude doručen e-mail s objednávkou, veškeré objednávky budou také vypsány na samostatné stránce.

### **Určení částí tématu zpracovávaných jednotlivými žáky:**

- příprava DB modelu a realizace fyzické databáze v MySQL
- příprava knihovny objektů pro realizaci backendu v PHP
- realizace kompletního backendu aplikace v PHP
  - seznam produktů
  - kategorie produktů
  - filtrace dle ceny a výrobce,
  - vyhledávání zboží dle názvu
  - detail zboží
- objednávka zboží
- kompletní administrace s autentizací přístupu přes DB

### **Plán konzultací:**

Pravidelné konzultace v rámci výuky předmětu WWA

**Požadavek na počet vyhotovení maturitní práce:** 2 výtisky

Termín odevzdání: **31. března 2021**

Čas obhajoby: **15 minut**

Vedoucí práce: **Mgr. Pavel Anderle**

Projednáno v **katedře** VTT a schváleno ředitelkou školy.

V Plzni dne: 30. října 2020 Ing. Naděžda Mauleová, MBA, v.r.

*ředitelka školy*

## Anotace

Tato práce pojednává o realizaci kompletního backendu eshopu s módou, v rámci dlouhodobé praktické maturitní práce. Pro realizaci byl použit programovací jazyk PHP, pro databáze MySQL.

## Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů. Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Plzni dne 10.2.2019.....

# Obsah

## 1. Uživatelská dokumentace

1. Manuál pro koncového uživatele
2. Manuál pro administrátory

## 2. Technická dokumentace

1. Databázový model
2. Fyzická databáze
3. Filtrace dle kategorie
4. Filtrace dle ceny
5. Detail produktu
6. Objednávací systém
7. Administrace
8. Používaný softwear

## 3. Závěr

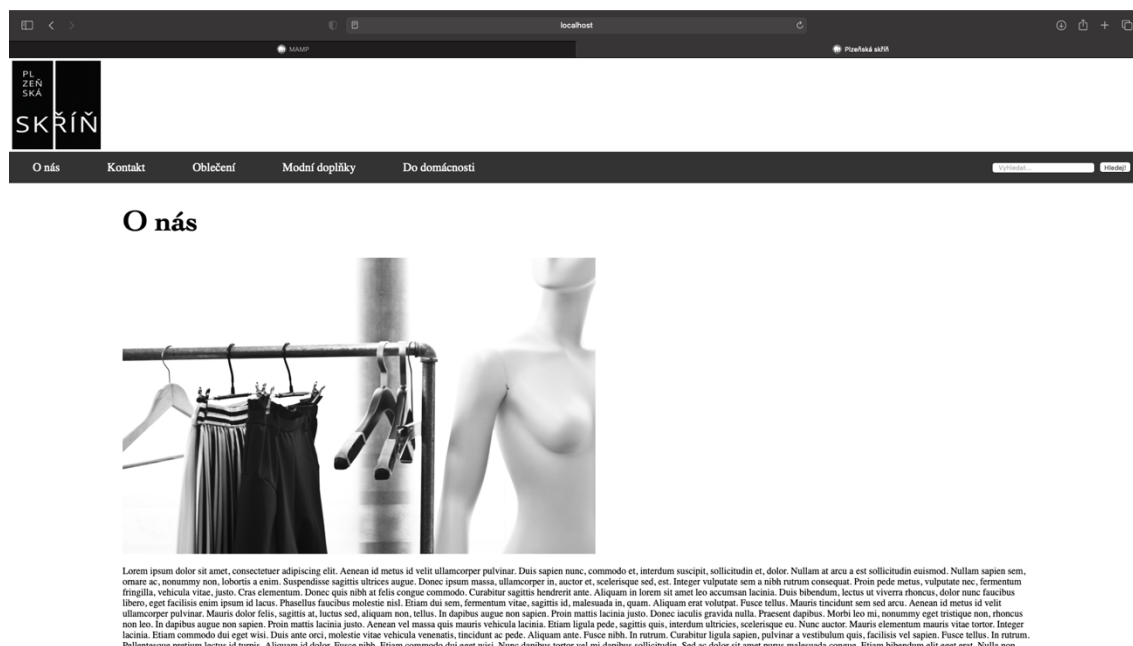
1. Zhodnocení práce

# 1 Uživatelská dokumentace

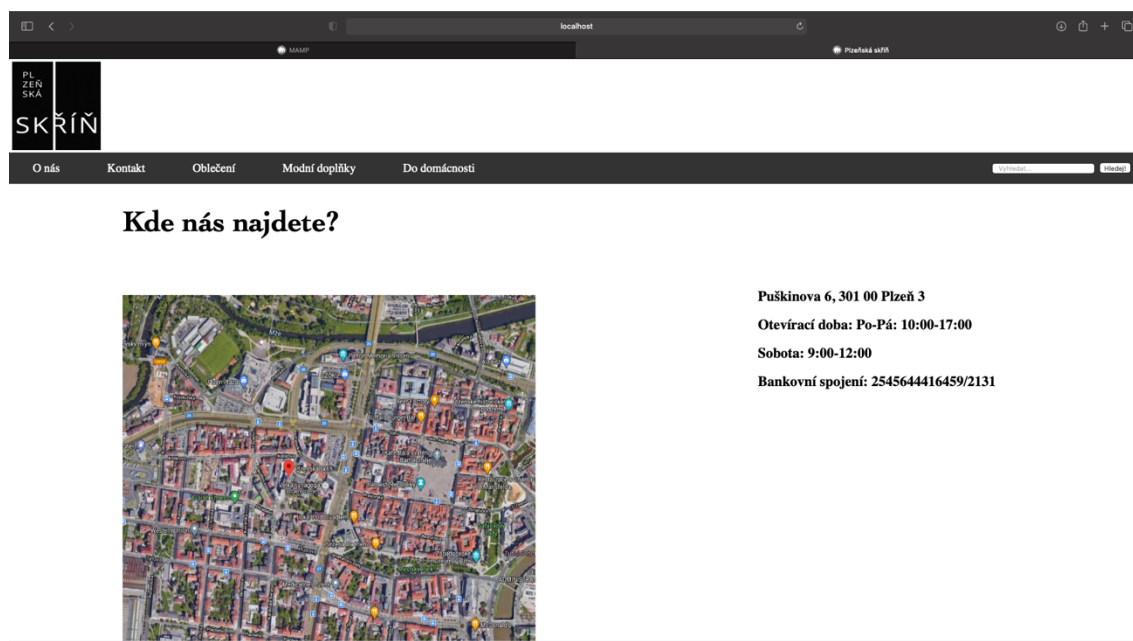
## 1.1 Manuál pro koncového uživatele

Mnou vytvořený eshop s módou je webová aplikace, tudíž na tento eshop můžeme přistupovat pomocí jakéhokoliv webového prohlížeče, po zadání správné URL adresy. Tato webová stránka aktuálně není nikde veřejně přístupná, tudíž lze spustit pouze přes náš vlastní server, tzv. localhost.

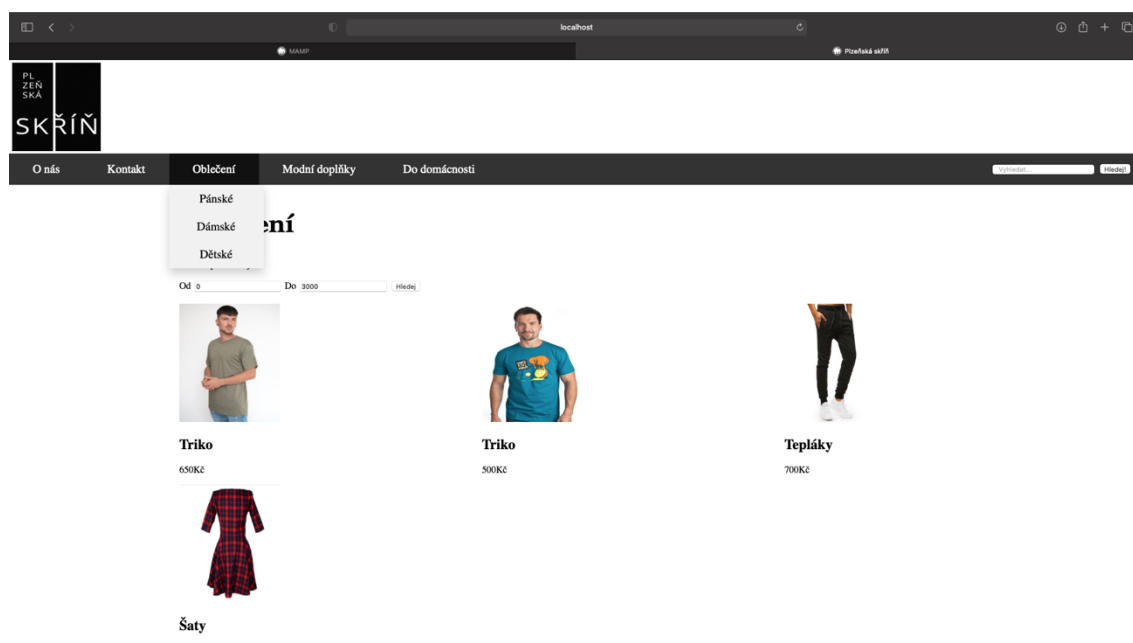
Při prvním otevření stránky se ocitnete na podstraně „O nás“, kde se můžete dozvědět vše, co vám majitel eshopu bude o své firmě chtít říct.



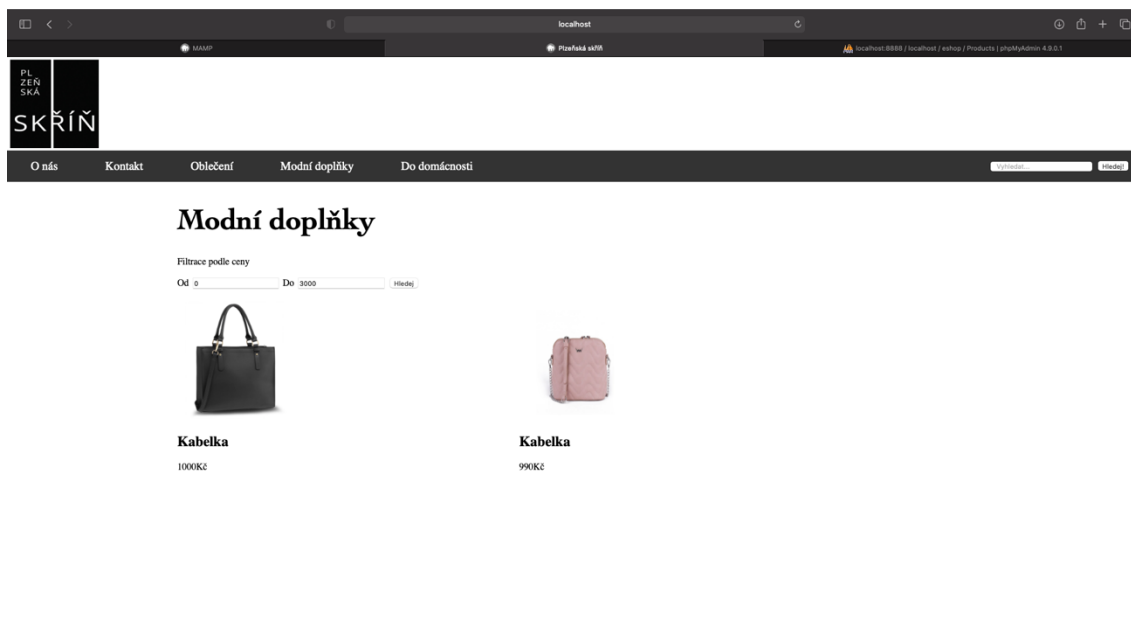
Dále se zde nachází podstrana „Kontakt“, kde se můžete dozvědět kde firma sídlí, na jaké telefonní číslo ji můžete kontaktovat, nebo na jaký email ji napsat.



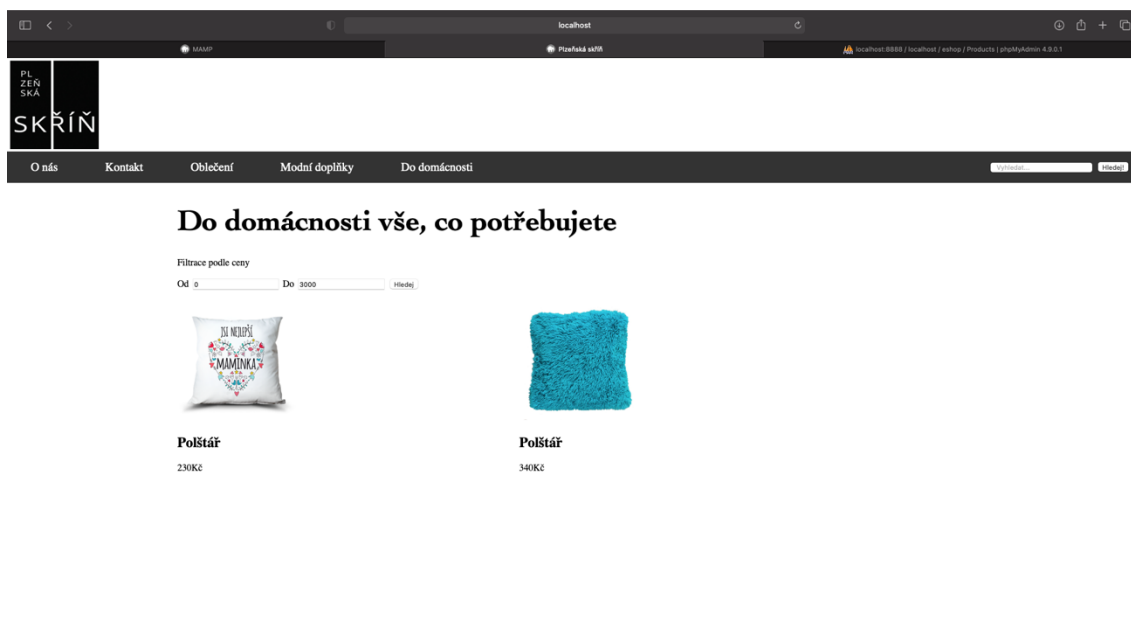
Dále můžete pokračovat na podstranu „Oblečení“, na které již máte na výběr ze všech produktů, které spadají do kategorie oblečení. Tato podstrana se dále rozděluje na pánské, dámské, a dětské oblečení.



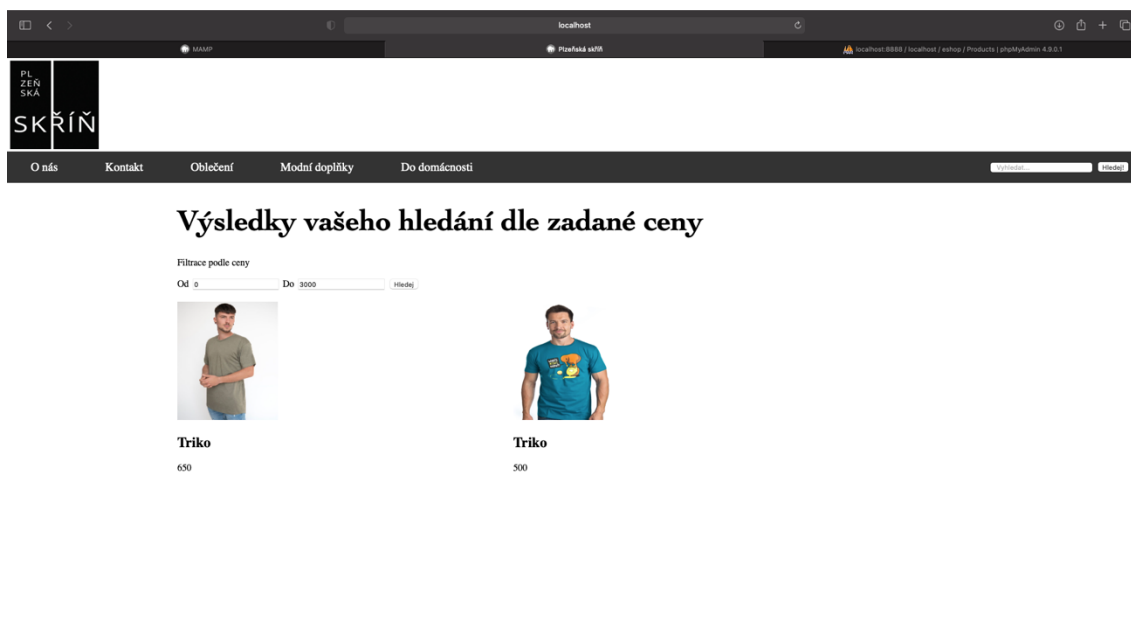
Dále zde je záložka „Módní doplňky“, ve které si můžeme zobrazit všechny produkty z této kategorie.



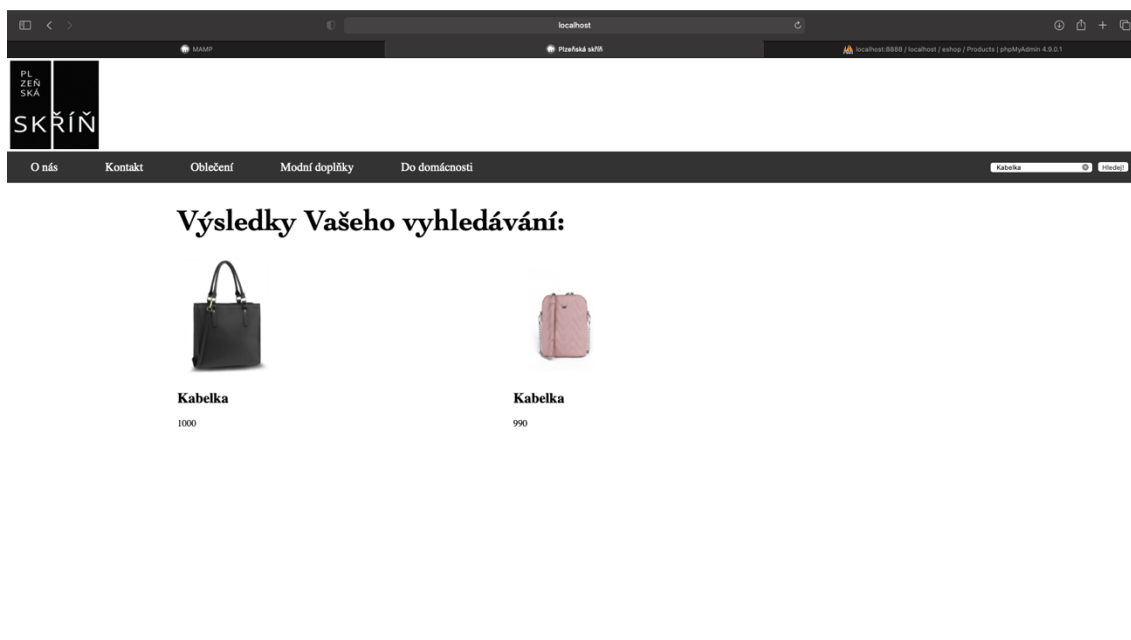
A v neposlední řadě podstrana „Do domácnosti“, ve které si můžete zobrazit produkty z této kategorie.



Na každé z těchto stran také můžete filtrovat produkty dle ceny.




Nebo také můžete vyhledat produkty dle vámi zadaného slova.





Při kliknutí na Vámi vybraný produkt se vám zobrazí detail vybraného produktu. Na této straně také můžete kliknout na tlačítko objednat. Po kliknutí na toto tlačítko se Vám zobrazí formulář, ve kterém vypíšete adresu pro doručení, jméno, příjmení, a další údaje nutné pro doručení objednávky. Po stisknutí tlačítka „odeslat“ se objednávka odešle, v případě nevyplnění nějaké z položek, vyskočí upozornění že objednávku se nepodařilo odeslat. Zároveň s odesláním objednávky Vám přijde email s potvrzením objednávky.

Objednávací formulář

  
650Kč  
Triko

E-mail:

Jméno:

Příjmení:

Adresa:

Telefon:

Vyhberte způsob dopravy:  
Rychlá pošta

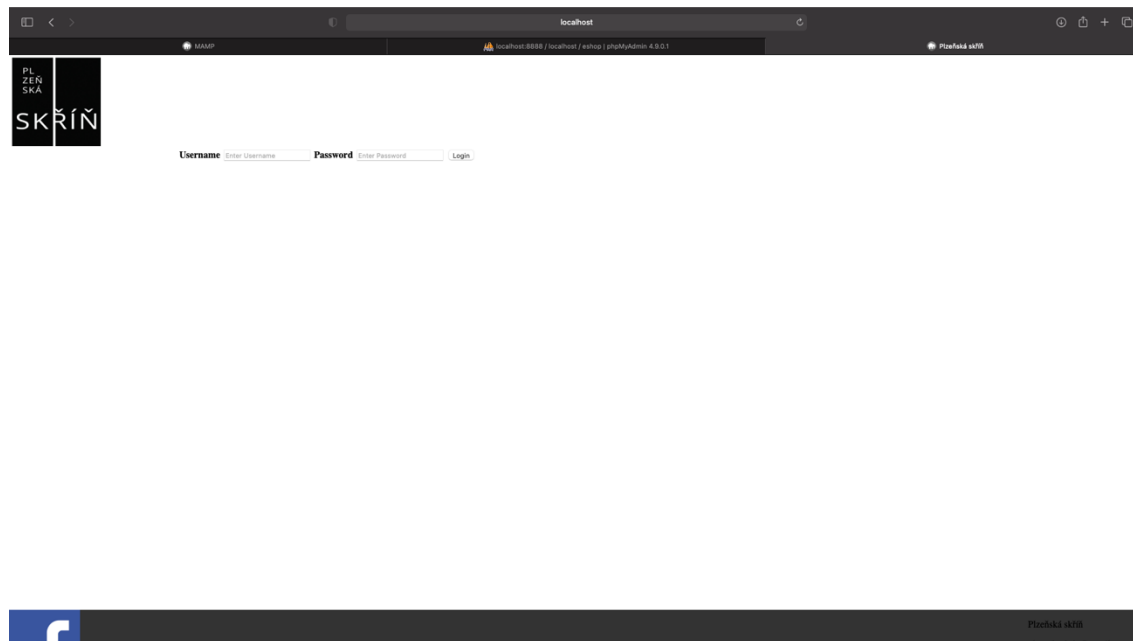
Vyhberte způsob platby:  
Dotazníka

Odeslat

## 1.2 Manuál pro administrátory

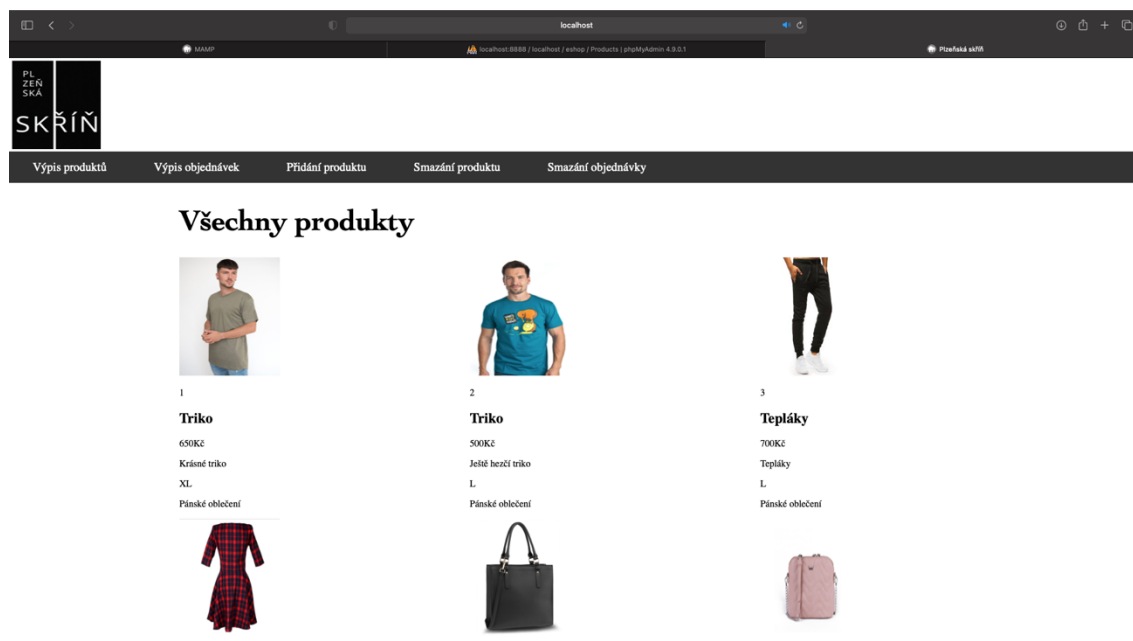
Pro přístup do administrátorské části webu musíme přepsat URL na <http://localhost:8888/Prmat/administrace/loginscreen.latte> . Ve chvíli kdy tato stránka bude mít zakoupenou doménu a webhosting, administrace poběží na samostatné subdoméně.

Při vstupu do administrátorské části se musíme v první řadě přihlásit. Přihlášení provedeme přes stránku k tomu určenou pomocí údajů, které jsou zapsány v databázi jako údaje pro ověření administrátora.



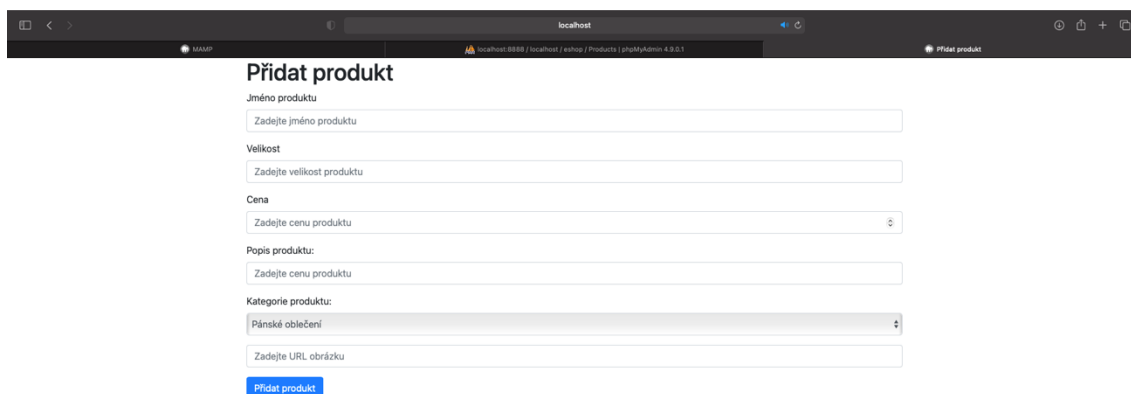
Pokud se Vám povede úspěšně přihlásit, dostanete se na hlavní stránku administrace, ze které se můžete dostat na provedení jednotlivých akcí, které v této administraci lze provést.

První funkce kterou můžete v administraci využít, je výpis všech produktů, které na eshopu máte. U každého produktu vidíte veškeré údaje, které jsou u něj zapsány. Do této záložky se dostaneme kliknutím na tlačítko „Výpis produktů“.



Úplně stejný výpis je i pro objednávky. Dostaneme se do něj kliknutím na tlačítko „Výpis objednávek“.

Další funkcí je záložka „Přidání produktu“, ve které můžete přidávat produkty. Při spuštění této podstrany se vám otevře formulář, do kterého vyplníte veškeré údaje o produktu, který chcete přidat. Po vyplnění formuláře a kliknutí na tlačítko „Přidat produkt“ se vytvoří nový produkt v databázi, a začne se zobrazovat na eshopu.



localhost

### Přidat produkt

Jméno produktu

Velikost

Cena

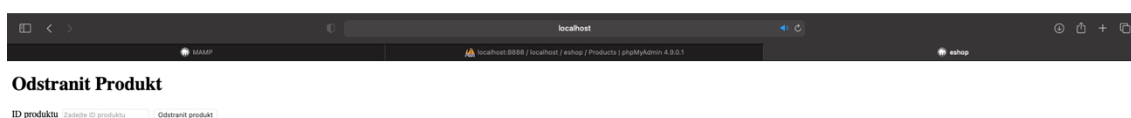
Popis produktu:

Kategorie produktu:

Zadejte URL obrázku

[Přidat produkt](#)

Další funkce které administrace nabízí, je smazání produktu a smazání objednávky. Do formuláře napíšeme ID produktu který chceme smazat, v případě smazání objednávky ID objednávky. Tyto ID si nejjednodušeji najdeme ve výpisu všech produktů / objednávek. Po vyplnění ID a stisknutí tlačítka odstranit se daný produkt/objedávka smaže z databáze, tím pádem se v případě produktu daný produkt přestane zobrazovat na eshopu.



localhost

### Odstranit Produkt

ID produktu  [Odstranit produkt](#)

## 2 Technická dokumentace

### 2.1 Databázový model

Databázový model jsem si sestavil velice jednoduchý, pouze napsaný v poznámkách. Vzhledem k tomu, že jednotlivé tabulky nejsou nijak propojeny, tento jednoduchý návrh mi zcela stačil. Promyslel jsem si, co v jaké tabulce potřebuji, abych tabulky měl finálně připravené, a nemusel je upravovat během samotného vytváření eshopu.

**Products**

1. id
2. name - jméno produktu
3. price - cena
4. size - velikost
5. description - popis produktu
6. image - název souboru s obrázkem
7. category - kategorie do které bude produkt zařazen

**Orders**

1. order\_id - id objednávky
2. product\_id - id objednaného produktu
3. customer\_name - jméno zákazníka
4. customer\_surname - příjmení zákazníka
5. customer\_phonenumber - tel. číslo zákazníka
6. customer\_email - email zákazníka
7. customer\_address - adresa zákazníka
8. shipment\_type - typ dopravy
9. payment\_type - typ platby

**Admins**

1. id - id administrátora
2. uname - přihlašovací jméno administrátora
3. upassword - heslo pro přihlášení administrátora

## 2.2 Fyzická databáze

Pro vytvoření fyzické databáze jsem používal jazyk MySQL, který jsem spouštěl na lokálním serveru pomocí PHPMyAdmin. Databáze byla vytvářena zadáváním jednotlivých SQL dotazů.

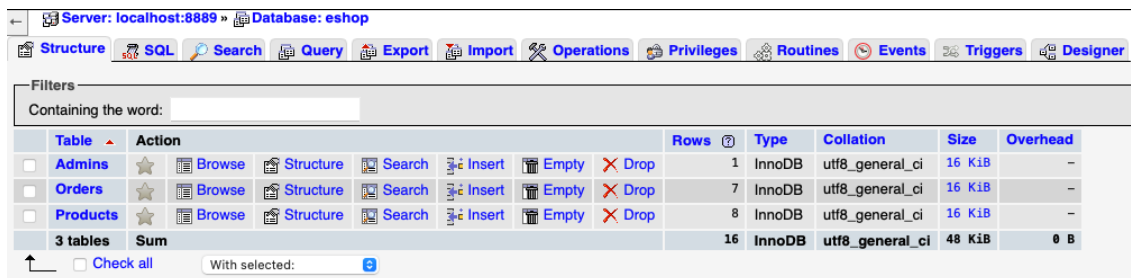


Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> Admins	Browse  Structure  Search  Insert  Empty  Drop	1	InnoDB	utf8_general_ci	16 KiB	-
<input type="checkbox"/> Orders	Browse  Structure  Search  Insert  Empty  Drop	7	InnoDB	utf8_general_ci	16 KiB	-
<input type="checkbox"/> Products	Browse  Structure  Search  Insert  Empty  Drop	8	InnoDB	utf8_general_ci	16 KiB	-
<b>3 tables</b>	<b>Sum</b>	<b>16</b>	<b>InnoDB</b>	<b>utf8_general_ci</b>	<b>48 KiB</b>	<b>0 B</b>

## 2.3 Filtrace dle kategorií

Pro filtrování dle kategorií jsem použil programovací jazyk PHP, pomocí kterého jsem vybíral produkty, které spadají do dané kategorie. Kód se nejprve připojí k samotné databázi, ověří zda připojení proběhlo úspěšně, a pokud ano tak odesílá výběrový SQL dotaz, pro výběr článků z dané kategorie. Vybrané produkty se uloží do pole, které je následně poskytnuto pro použití v šabloně tvořené v jazyku Latte.

```

1 </php
2
3 require "../vendor/autoload.php";
4
5 $latte = new Latte\Engine;
6
7 $latte->setTempDirectory('temp');
8
9 function loadListArticlesByCategory($category){
10     $servername = "localhost";
11     $username = "root";
12     $password = "root";
13     $dbname = "eshop";
14
15
16
17
18     $feedback = true; // počáteční nastavení návratové hodnoty
19     $conn = new mysqli($servername, $username, $password, $dbname); // vytvoření objektu mysqli
20
21     if ($conn->connect_error) {
22         $feedback = false; // nastavení návratové hodnoty na false v případě chyby s připojením k DB serveru
23     }
24
25     $sqlCharset = "SET CHARACTER SET UTF8"; // SQL dotaz nastavující kódovou stránku pro komunikaci s DB serverem
26     $conn->query($sqlCharset); // odeslání SQL dotazu na DB server
27
28     $sql= "SELECT * FROM Products WHERE category='$category'"; // SQL dotaz pro výběr produktů dle kategorie
29     $result = $conn->query($sql);
30     if ($result->num_rows > 0) { // kontrola zdali SQL dotaz SELECT vrátil produkty
31
32         while($row = $result->fetch_assoc()) { // postupně procházení řádek výsledku - fetch_assoc() vrací pole hodnot jednoho řádku
33             $listArticles[] = $row;
34
35         }
36
37         $feedback = $listArticles;
38
39     }
40     else {
41         $feedback = false; // v případě, že DB server nevrátí žádný záznam - výstupní hodnota False
42     }
43     $conn->close();
44     return $feedback;
45 }
46
47 $params = []
48 'listArticles'=> loadListArticlesByCategory("Dámské oblečení"),
49 ];
50
51 $latte->render('../templates/shopoblecení.latte', $params);
52
53
54
55 7>
56

```

## 2.4 Filtrace dle ceny

Na každé straně s produkty také můžeme produkty filtrovat dle ceny. Stejně jako ve všech ostatních bodech jsem použil jazyk PHP. Uživatel vždy zadá minimální částku od které chce vyhledávat produkty, a zároveň maximální částku, nad kterou cena produktu nesmí stoupnout. Znovu se nejdříve připojíme na databázi, ověříme zda připojení proběhlo úspěšně, a poté zadáváme výběrový SQL dotaz. Uživateli zadané hodnoty získáváme pomocí metody GET. Pro každou stranu je vytvořen samostatný PHP kód pro vyhledávání, a to proto aby vždy vyhledávání dle ceny proběhlo vždy jen ve vybrané kategorii. Produkty které projdou výběrem se znovu ukládají do pole, které je poskytnuto pro používání v šabloně jazyka Latte.

```

1  <?php
2
3  require "../vendor/autoload.php";
4
5  $latte = new Latte\Engine;
6
7  $latte->setTempDirectory('temp');
8
9  function loadListArticlesByPrice($category,$high,$low){
10     $servername = "localhost";
11     $username = "root";
12     $password = "root";
13     $dbname = "eshop";
14
15
16
17
18     $feedback = true; // počáteční nastavení návratové hodnoty
19     $conn = new mysqli($servername, $username, $password, $dbname); // vytvoření objektu mysqli
20
21     if ($conn->connect_error) {
22         $feedback = false; // nastavení návratové hodnoty na false v případě chyby s připojením k DB serveru
23     }
24
25     $sqlCharset = "SET CHARACTER SET UTF8"; // SQL dotaz nastavující kódovou stránku pro komunikaci s DB serverem
26     $conn->query($sqlCharset); // odeslání SQL dotazu na DB server
27
28     $sql= "SELECT * FROM Products WHERE category='$category' AND price BETWEEN '$low' AND '$high'"; //výběrový dotaz pro hledání produktů dle ceny
29     $result = $conn->query($sql);
30     if ($result->num_rows > 0) { // kontrola zdali SQL dotaz SELECT vrátil produkty
31
32         while($row = $result->fetch_assoc()) { // postupné procházení řádek výsledku – fetch_assoc() vrací pole hodnot jednoho řádku
33             $listArticles[] = $row;
34
35         }
36
37         $feedback = $listArticles;
38
39     }
40
41     else {
42         $feedback = false; // v případě, že DB server nevrátí žádný záznam – výstupní hodnota False
43     }
44     $conn->close();
45     return $feedback;
46 }
47 $low = $_GET['low']; //získání hodnoty od uživatele
48 $high = $_GET['high']; //získání hodnoty od uživatele
49 $params = [
50     'listArticles'=> loadListArticlesByPrice("Dámské oblečení", $high, $low),
51 ];
52
53 $latte->render("../templates/vyhledani.latte", $params);
54

```

## 2.5 Detail produktu

Při kliknutí na vybraný produkt se vám zobrazí stránka s detailem vybraného produktu. Tento problém je stejně jako ten předchozí řešen pomocí PHP. Tento kód nejdříve provede připojení na databázi, ověří zda připojení proběhlo úspěšně, a poté zadává konkrétní SQL dotaz. Jaký produkt bude v detailu zobrazen je vybrán pomocí jeho ID, které je získáno pomocí metody GET. Vybraný produkt se všemi jeho atributy je uložen do pole, které je poté odesláno pro zobrazení v šabloně v napsané v jazyce latte. Z této stránky se také můžete dostat k samotné objednávce produktu.



```

1 <?php
2
3 require "../vendor/autoload.php";
4
5
6
7
8 $latte = new Latte\Engine;
9
10
11 $latte->setTempDirectory('temp');
12
13
14
15 function loadDataById($id)
16 {
17     $servername = "localhost";
18     $username = "root";
19     $password = "root";
20     $dbname = "eshop";
21
22     $feedBack = true; // počáteční nastavení návratové hodnoty
23     $conn = new mysqli($servername, $username, $password, $dbname); // vytvoření objektu mysqli
24
25     if ($conn->connect_error) {
26         $feedBack = false; // nastavení návratové hodnoty na false v případě chyby s připojením k DB serveru
27     }
28
29     $sql = "SET CHARACTER SET UTF8"; // SQL dotaz nastavující kódovou stránku pro komunikaci s DB serverem
30     $conn->query($sql); // odeslání SQL dotazu na DB server
31
32     $sql = "SELECT * FROM Products WHERE id='$id'"; // SQL dotaz pro výběr produktu dle id z tabulky Products
33
34     $result = $conn->query($sql); // odeslání SQL dotazu na DB server - $result obsahuje výsledek dotazu
35
36     if ($result->num_rows > 0) { // kontrola zdali SQL dotaz SELECT vrátil produkt
37         while($row = $result->fetch_assoc()) { // postupné procházení řádek výsledku - fetch_assoc() vrací pole hodnot jednoho řádku
38             $listArticles[] = $row;
39         }
40         $feedBack = $listArticles;
41     }
42     else {
43         $feedBack = false; // v případě, že DB server nevrátí žádný záznam - výstupní hodnota False
44     }
45     $conn->close();
46     return $feedBack;
47
48 }
49
50 $id = $_GET['id']; // získání id článku
51 $params = [
52     'listArticles' => loadDataById($id),
53 ];
54
55 $latte->render('../templates/Detailscreen.latte', $params);
56

```

## 2.6 Objednávací systém

Po kliknutí na tlačítko objednat se dostanete na formulář, ve kterém vyplníte všechny potřebné údaje pro objednání. Po vyplnění tohoto dotazníku kliknete na tlačítko „Objednat“. Po stisku tohoto tlačítka se objednávka uloží do databáze, a uživateli se odešle email s potvrzením objednávky. Celá tato funkčnost je znovu řešena přes PHP. Prvně se PHP připojí na databázi, dále se ověří zda připojení proběhlo úspěšně. Poté se odesílá SQL dotaz, který zapíše veškeré údaje do databáze, a poté odešle potvrzovací email. Po odeslání je uživatel zpět přesměrován na hlavní stranu eshopu. Veškeré zadané údaje získává PHP kód pomocí metody GET. Pro odeslání emailu je využita PHP metoda mail.

```

1 <?php
2 $id = $_GET['id'];
3 $email = $_GET['email'];
4 $jmeno = $_GET['jmeno'];
5 $prijmeni = $_GET['prijmeni'];
6 $adresa = $_GET['adresa'];
7 $telefon = $_GET['telefon'];
8 $doprava = $_GET['doprava'];
9 $platba = $_GET['platba'];
10 $servername = "localhost";
11 $username = "root";
12 $password = "root";
13 $dbname = "eshop";
14 $conn = new mysqli($servername, $username, $password, $dbname);
15 $sqlCharset = "SET CHARACTER SET UTF8";
16 $conn->query($sqlCharset);
17 $sql = "INSERT INTO Orders (product_id, customer_name, customer_surname, customer_phonenumber, customer_email, customer_address, shipment_type, payment_type)
18 VALUES ('$id', '$email', '$jmeno', '$prijmeni', '$adresa', '$telefon', '$doprava', '$platba')";
19 $conn->query($sql);
20 $conn->close();
21
22 if ($_GET['send']) {
23     $toEmail = $_GET['email'];
24     $emailSubject = "Potvrzení Vaší objednávky";
25     $mailheader = "From: mara@marekhejl.cz\r\n";
26     $mailheader .= "Reply-To: mara@marekhejl.cz\r\n";
27     $mailheader .= "Content-type: text/html; charset=utf-8\r\n";
28     $mailheader .= "X-Mailer: PHP/" . phpversion();
29     $MESSAGE_BODY = "Děkujeme za Vaši objednávku. Vaše objednávka se právě zpracovává. O dalším průběhu Vás budeme informovat na Váš e-mail.";
30
31     if(mail($toEmail, $emailSubject, $MESSAGE_BODY, $mailheader)!=true) {
32
33
34     }
35     <script type='text/javascript'> alert('Objednávku se nepodařilo odeslat!'); </script>
36
37 <?php
38 header("Location: http://localhost:8888/Prmat/templates/shopAboutus2.latte");
39
40 } else{
41     header("Location: http://localhost:8888/Prmat/templates/shopAboutus2.latte");
42 }
43 <script type='text/javascript'>alert('Děkujeme za objednávku!');
44 </script>
45 <?php
46 }
47 }
48
49 ?>

```

## 2.7 Administrace

### 2.7.1 Administrace – Přihlášení

Přihlášení do samotné administrace je řešeno pomocí PHP, které ověřuje zda zadané přihlašovací údaje jsou zadány v databázi jako validní údaje pro přihlášení.

Administrátor vyplní své přihlašovací údaje do formuláře, ze kterého jsou do PHP kódu odeslány pomocí metody GET, a následně jsou tímto kódem ověřeny. Kód se nejprve připojí do databáze, ověří zda připojení proběhlo úspěšně, a dále pokračuje výběrovým dotazem SQL. Pokud výběrový dotaz najde v tabulce administrátorů Vámi zadané údaje, založí session s názvem „admin“, a nastaví její hodnotu na „true“, a přesměruje administrátora na hlavní stranu administrace. Pokud se zadané údaje nepodaří ověřit, zobrazí se hláška „Přihlášení se nezdařilo“, a uživatel je přesměrován zpět na přihlašovací stránku.

```

1  ?php
2
3      $uname = $_GET['name'];
4      $upassword = $_GET['password'];
5
6      $servername = "localhost";
7      $username = "root";
8      $password = "root";
9      $dbname = "eshop";
10
11     $feedBack = true; // počáteční nastavení návratové hodnoty
12
13     $conn = new mysqli($servername, $username, $password, $dbname); // vytvoření objektu mysqli
14
15     if ($conn->connect_error) {
16         $feedBack = false; // nastavení návratové hodnoty na false v případě chyby s připojením k DB serveru
17     }
18
19     $sqlCharSet = "SET CHARACTER SET UTF8"; // SQL dotaz nastavující kódovou stránku pro komunikaci s DB serverem
20     $conn->query($sqlCharSet); // odeslání SQL dotazu na DB server
21
22     $sql= "SELECT * FROM Admins WHERE uname='$uname' AND upassword='$upassword'";
23     $result = $conn->query($sql);
24     if ($result->num_rows > 0) { // kontrola zda-li SQL dotaz SELECT našel uživatele
25         session_start();
26         $_SESSION["admin"] = "true";
27         header("Location: http://localhost:8888/Prmat/administrace/mainstranka.php");
28         exit();
29     }
30
31     else {
32         ?<script type='text/javascript'> alert('Přihlášení se nezdařilo!'); </script>
33         <?php
34         header("Location: http://localhost:8888/Prmat/administrace/loginscreen.latte");
35         exit();
36     }
37     $conn->close();
38
39
40
41
42  ?>

```

## 2.7.2 Administrace – Výpis produktů a objednávek

Výpis produktů a objednávek je řešen pomocí PHP. Kód nejprve ověří, zda je aktivní session „admin“, jinými slovy zda jste přihlášen jako administrátor. Pokud jste, pokračujete dále na tuto stranu, pokud ne, jste přesměrován zpět na přihlašovací stránku. Dále se připojí na databázi, ověří zda se přihlášení zdařilo. Následně vybírám záznamy o produktech/objednávkách z databáze. Výsledky jsou ukládány do pole, které je následně poskytnut stránce v jazyku latte, aby je zobrazilo. Tento kód se pro výpis objednávek a výpis produktů mění pouze SQL dotaz, a to v tom, z jaké tabulky vybírám záznamy.

```

1 <?php
2 session_start();
3
4 if(isset($_SESSION["admin"])) {
5
6 } else {
7     header("Location: http://localhost:8888/Prmat/administrace/loginscreen.latte");
8     exit();
9 }
10 require "../vendor/autoload.php";
11
12 $latte = new Latte\Engine;
13
14 $latte->setTempDirectory('temp');
15
16 function loadListOfOrders(){
17     $servername = "localhost";
18     $username = "root";
19     $password = "root";
20     $dbname = "eshop";
21
22     $feedBack = true; // počáteční nastavení návratové hodnoty
23     $conn = new mysqli($servername, $username, $password, $dbname); // vytvoření objektu mysqli
24
25     if ($conn->connect_error) {
26         $feedBack = false; // nastavení návratové hodnoty na false v případě chyby s připojením k DB serveru
27     }
28
29     $sqlCharSet = "SET CHARACTER SET UTF8"; // SQL dotaz nastavující kódovou stránku pro komunikaci s DB serverem
30     $conn->query($sqlCharSet); // odeslání SQL dotazu na DB server
31
32     $sql= "SELECT * FROM Orders";
33     $result = $conn->query($sql);
34     if ($result->num_rows > 0) { // kontrola zdali SQL dotaz SELECT vrátil záznamy
35
36         while($row = $result->fetch_assoc()) { // postupné procházení řádek výsledku - fetch_assoc() vrací pole hodnot jednoho řádku
37             $listArticles[] = $row;
38         }
39         $feedBack = $listArticles;
40     }
41     else {
42         $feedBack = false; // v případě, že DB server nevrátí žádný záznam - výstupní hodnota False
43     }
44     $conn->close();
45     return $feedBack;
46 }
47
48 $params = [
49     'listArticles'=> loadListOfOrders(),
50 ];
51
52 $latte->render('objednavky.latte', $params);
53
54 ?>
55

```

### 2.7.3 Administrace – Přidání produktu

Přidání produktu je řešeno pomocí PHP. Po vyplnění formuláře všemi potřebnými údaji a stisknutí tlačítka „Přidat“ se spustí PHP kód, který si převezme zadaná data z formuláře pomocí metody GET, připojí se na databázi, ověří zda připojení proběhlo úspěšně, a poté pomocí SQL dotazu запиše data do databáze do tabulky s produkty.

```

<?php
$name=$_GET["name"];
$size=$_GET["size"];
$price=$_GET["price"];
$description=$_GET["description"];
$category=$_GET["category"];
$img=$_GET["img"];

$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "eshop";

$conn = new mysqli($servername, $username, $password, $dbname);

$sql = "SET CHARACTER SET UTF8";
$conn->query($sql);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO Products (name, price, size, description, image, category) VALUES ('$name', '$price', '$size', '$description', '$img', '$category')";

if ($conn->query($sql) === TRUE) {
    header("Location: http://localhost:8888/Prmat/administrace/mainstranka.php");
    exit();
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>

```

## 2.7.4 Administrace – Smazání produktu/objednávky

Pro smazání produktu/objednávky jsem použil PHP. Administrátor zadá ID produktu nebo objednávky kterou chce smazat, a klikne na tlačítko „Odstranit“. PHP kód pomocí metody GET získá administrátorem zadané ID, a podle něj vyhledá produkt/objednávku v tabulce produktů/objednávky, a daný záznam odstraní. Vyhledávání a odstranění produktu/objednávky probíhá pomocí SQL dotazu. Kód pro odstranění objednávky a produktu je rozdílný pouze v tom, z jaké tabulky SQL dotaz záznam odstraňuje.

```

1  <?php
2
3  $id=$_POST["id"];
4  $servername = "localhost";
5      $username = "root";
6      $password = "root";
7      $dbname = "eshop";
8
9      $conn = new mysqli($servername, $username, $password, $dbname);
10
11  $sql = "SET CHARACTER SET UTF8";
12      $conn->query($sql);
13
14
15  if ($conn->connect_error) {
16      die("Connection failed: " . $conn->connect_error);
17  }
18
19
20  $sql = "DELETE FROM Orders WHERE order_id = '$id'";
21
22  if ($conn->query($sql) === TRUE) {
23      header("Location: http://localhost:8888/Prmat/administrace/mainstranka.php");
24  } else {
25      echo "Error: " . $sql . "<br>" . $conn->error;
26  }
27
28  $conn->close();
29
30  ?>

```

## 2.8 Používaný softwear

### Visual Studio Code

Pro psaní PHP kódu jsem využíval program Visual Studio Code, který je zdarma dostupný. Pomocí tohoto programu jsem také práci nahrával na GIT.

## GIT

Verzovací systém, který slouží pro verzování napsaného kódu. Tato služba byla velice vhodná pro spolupráci ve dvojici. Soubory jsme si mohli pomocí této služby jednoduše sdílet. Díky GITU jsme oba ve dvojici stále viděli nejaktuálnější verzi kódu.

## MAMP

Program pomocí kterého jsem si na svém počítači spouštěl lokální server. Zahrnuje také PhpMyAdmin, pomocí kterého jsem vytvářel a upravoval SQL databázi.

## 3 Závěr

### 3.1 Zhodnocení práce

Finální práci hodnotím kladně. Vše se povedlo úspěšně zprovoznit. Myslím si, že jsem splnil všechny body zadání, které jsem splnit měl. Během dělání této práce jsem si vyzkoušel, jaké to je dělat takto velký projekt, utvrdil se v tom, že právě toto je práce, kterou by jsem chtěl v budoucnu dělat, a také zjistil jaké to je, když na práci pracujete ve více lidech. Spolupráci ve dvojici též hodnotím zcela kladně.