

**Obchodní akademie, Vyšší odborná škola a Jazyková škola  
s právem státní jazykové zkoušky Uherské Hradiště**



**WEB - AUTOSERVIS**

# OBSAH

1.1	Popis Software a knihoven .....	4
1.1.1	Framework Nette .....	4
1.1.2	Knihovna Bootstrap .....	4
1.2	Presentery (presenter Opravy) .....	4
1.2.1	Získání jmenných hodnot ID do select menu .....	5
1.2.2	Funkce PostFormSucceeded .....	7
1.2.3	Funkce actionDelete .....	7
1.2.4	Funkce actionEdit .....	8
1.2.5	Routování .....	8
1.2.6	E-R model .....	8
1.3	Uživatelská dokumentace .....	9
1.3.1	Zprovoznění webu na svém webhostingu .....	9
1.3.2	Popis webu .....	10

## ÚVOD

*Práci jsem rozdělil do tří hlavních částí a to na popis software a knihoven, kde jsem vypsal knihovny a framework, který jsem použil. Ve druhé části se věnuji programátorské části webové stránky. Zejména presenteru 'opravy', který obsahuje všechny funkce, které jsou na ostatních stránkách. Nakonec jsem přidal E-R diagram databáze s popisem vztahů mezi tabulkami.*

*Ve třetí části se věnuji zprovoznění webu na webhostingu a vysvětlil základní části včetně propojení s databází.*

# TECHNICKÁ ČÁST

## 1.1 Popis Software a knihoven

### 1.1.1 Framework Nette

Název: Nette

Autor: David Grudl, Nette Foundation

Verze: 3.1

Odkaz ke stažení: <https://doc.nette.org/cs/3.1/composer>

Licence: New BSD, GNU GPL

Další knihovny, které Nette obsahuje jsou na této stránce: <https://nette.org/cs/packages>

### 1.1.2 Knihovna Bootstrap

Název: Bootstrap

Autor: Bootstrap team (<https://getbootstrap.com/docs/5.0/about/team/>)

Verze: v5.0.0-beta3

Odkaz ke stažení: <https://getbootstrap.com/docs/5.0/getting-started/download/>

Licence: MIT

## 1.2 Presentery (presenter Opravy)

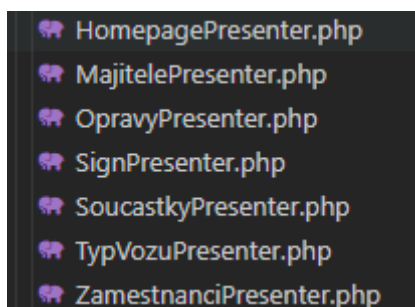
Presentery jsou vlastně controllery. Celkově jsem použil 7 presenterů v rámci přehlednosti.

Na každou stránku jeden presenter. Všimněte si, že všechny presentery jsou ve tvaru

**Název**Presenter. To proto, že v Nette se nemusí řešit routování. Routery, čili jak odkazy vypadají, se upravují v souboru RouterFactory.php.

Presentery jsou si velice podobné, jelikož se mění jenom proměnné. proto detailně rozeberu pouze jeden příklad.

Obrázek 1 – seznam vlastních presenterů



## Presenter Opravy

Tento presenter obsahuje veškerou funkčnost webu.

Obrázek 2 – Část kódu presenteru Opravy

```
protected function createComponentPostForm(): Form
{
    $form = new BootstrapForm;
    $form->setHtmlAttribute('class', 'container-main');
    $mysqli = mysqli_connect("localhost","root","","autoservis");

    $form->addDateTime('datum', 'Začátek opravy (Datum)')
        ->setHtmlAttribute('type', 'text')
        ->setRequired();

    $form->addDateTime('doba_opravy', 'Konec opravy (Doba opravy)')
        ->setHtmlAttribute('type', 'text')
        ->setRequired();

    $sql1 = "SELECT zamestnanec_id, prijmeni FROM opravy JOIN zamestnanci ON opravy.id = zamestnanci.id";
    if ($result1 = mysqli_query($mysqli, $sql1)){
        $list1 = array('Vyberte');
        if($row1 = mysqli_num_rows($result1) > 0) {
            while($row1 = mysqli_fetch_array($result1)){
                $list1[] = $row1['prijmeni'];
            }
        }
    }
    $form->addSelect('zamestnanec_id', 'Zaměstnanec', $list1)
        ->setHtmlAttribute('class', 'form-select')
        ->setRequired();
}
```

Jak můžeme vidět, funkce createComponentPostForm(): Form vytváří objekt knihovny BootstrapForm. Je to knihovna, která slouží k vykreslování formulářů ve stylu bootstrapu. Samozřejmě se do ní dá přidat vlastní styl.

\$form-addDateTime('datum', 'Začátek opravy (Datum)') značí, že se přidá do formuláře datum. Poté setHtmlAttribute('type', 'text') značí, že se přidá html atribut type=text. Samozřejmě by tam mohla být i třída (class) a vedle vypisovat třídy 'třída1 třída2 třída3'.

SetRequired(); znamená, že pole nesmí být prázdné.

Další část tvoří cyklus, pomocí kterého se vybírají identifikační čísla zobrazené jako jména.

### 1.2.1 Získání jmenných hodnot ID do select menu

Obsahuje ho Presenter 'OpravyPresenter.php' a 'SoucastkyPresenter.php'.

Obrázek 3 – kód pro získání informací do select menu

```
$sql1 = "SELECT zamestnanec_id, prijmeni FROM opravy JOIN zamestnanci ON opravy.id = zamestnanci.id";
if ($result1 = mysqli_query($mysqli, $sql1)){
    $list1 = array('Vyberte');
    if($row1 = mysqli_num_rows($result1) > 0) {
        while($row1 = mysqli_fetch_array($result1)){
            $list1[] = $row1['prijmeni'];
        }
    }
}
```

Proměnná 'sql1' obsahuje SQL dotaz, proměnná 'mysqli' obsahuje přihlašovací informace k databázi, tyto proměnné zpracuje funkce 'mysqli\_query', která zkontroluje správnost a připojí se k databázi. Další proměnná je 'list1', do níž je uloženo pole s hodnotou 'Vyberte', aby pole začínalo od jedničky. Další proměnná 'row1' v cyklu 'if' zkontroluje, že počet řádků není 0. Je-li tam více než 1 řádek čili databáze není prázdná, provede následující činnost. Vezme proměnnou 'row1' a z ní 'prijmeni'. To pak uloží do proměnné 'list1', kterou použijeme v metodě addSelect (viz. obr.). Použije se pak do metody addSelect v proměnné 'form'.

Řešení je dosti nepraktické, jelikož SQL zamezuje duplikátům ID. Proto, když se vytvoří hodnota a smaže, tak místo čísla 1 bude mít číslo 2 atd. Bootstrap forms metoda 'addSelect' v Bootstrap forms podporuje pouze pole (array).

Obrázek 4 - funkce addSelect

```
$form->addSelect('zamestnanec_id', 'Zaměstnanec', $list1)
```

Na konci formuláře se zavolá onSuccess pole, do kterého se vše uloží a zavolá se metoda 'PostFormSucceeded'.

Obrázek 5 - PostFormSucceeded

```
$form->onSuccess[] = [$this, 'PostFormSucceeded'];  
  
return $form;
```

### 1.2.2 Funkce PostFormSucceeded

Obrázek 6 - Funkce PostFormSucceeded

```
public function PostFormSucceeded(Form $form, $values): void
{
    if (!$this->getUser()->isLoggedIn()) {
        $this->redirect('Sign:in');
    }
    $formId = $this->getParameter('formId');

    if ($formId) {
        $form = $this->database->table('opravy')->get($formId);
        $form->update($values);
        $this->redirect('opravy');
    } else {

        $forms = $this->database->table('opravy')->insert($values);

        $this->redirect('Soucastky:soucastky');
    }
}
```

Na začátku této funkce je důležitý cyklus, který ověří, zda je uživatel přihlášený, pokud není, tak ho přesměruje na stránku 'in.php' (přihlášení). Další cyklus s proměnnou formID získá informace z tabulky opravy a aktualizuje hodnoty a vloží je do databáze. Poté přesměruje nás přesměruje na stránky 'soucastky' se součástkami.

### 1.2.3 Funkce actionDelete

Obrázek 7 - funkce actionDelete

```
public function actionDelete($id): void
{
    if (!$this->getUser()->isLoggedIn()) {
        $this->redirect('Sign:in');
    }
    $this->database->table('opravy')
        ->where("id", $id)
        ->delete('form');
    $this->redirect('opravy');
}
```

Počáteční cyklus ověří, zda je uživatel přihlášen a poté získá data z tabulky opravy a smaže je podle IDčka. Poté přesměruje na stránku s opravami.

## 1.2.4 Funkce actionEdit

Obrázek 8 - funkce actionEdit

```
public function actionEdit(int $formId): void
{
    if (!$this->getUser()->isLoggedIn()) {
        $this->redirect('Sign:in');
    }
    $form = $this->database->table('opravy')->get(["id" => $formId]);
    if (!$form) {
        $this->error('Příspěvek nebyl nalezen');
    }
    $this['postForm']->setDefaults($form->toArray());
}
```

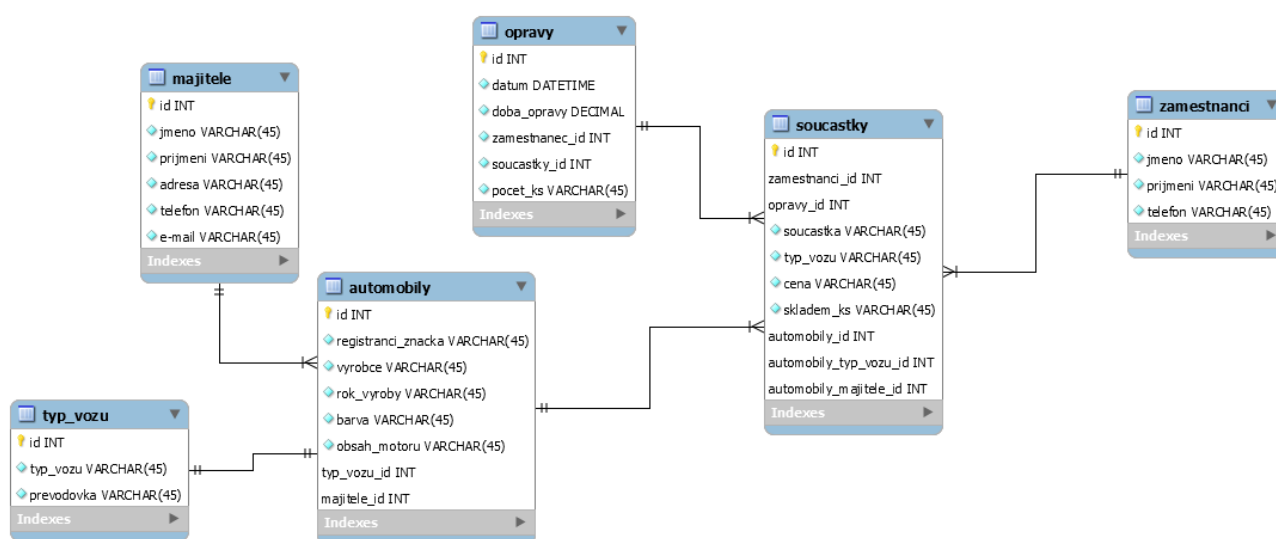
Po ověření přihlášení uživatele vezme z tabulky 'opravy' data podle IDčka. Pokud hodnota neexistuje, vypíše chybovou hlášku „Příspěvek nebyl nalezen.“

## 1.2.5 Routování

Routování jsem zde neřešil, jelikož jsem využíval výchozí nastavení frameworku čili routy se nemusí řešit jak ve frameworku codeIgniter. Například.: Ve složce templates vytvoříme složku s názvem 'Opravy'. Podle této složky uděláme presenter ve tvaru 'OpravyPresenter.php'. Poté se přes šablonu 'layout.latte' spojí všechny tři části webu.

## 1.2.6 E-R model

Databáze obsahuje tabulky majitelé, typ vozu, automobily, opravy, součástky a zaměstnanci.





## 1.3 Uživatelská dokumentace

Odkaz na web: <http://testwebsite.4fan.cz/>

### 1.3.1 Zprovoznění webu na svém webhostingu

První, co je potřeba znát je struktura, která je následující:

Obrázek 9 - struktura aplikace

<b>nette-blog/</b>	
— <b>app/</b>	← adresář s aplikací
— <b>Presenters/</b>	← třídy presenterů
— <b>templates/</b>	← šablony
— <b>Router/</b>	← konfigurace URL adres
— <b>Bootstrap.php</b>	← zaváděcí třída Bootstrap
— <b>bin/</b>	← skripty spouštěné z příkazové řádky
— <b>config/</b>	← konfigurační soubory
— <b>log/</b>	← logování chyb
— <b>temp/</b>	← dočasné soubory, cache, ...
— <b>vendor/</b>	← knihovny instalované Composerem
— <b>autoload.php</b>	← autoloading všech nainstalovaných balíčků
— <b>www/</b>	← veřejný adresář - jediný přístupný z prohlížeče
— <b>index.php</b>	← prvotní soubor, kterým se aplikace spouští

Ve složce 'config' v souboru 'local.neon' jsou lokální údaje k databázi.

Obrázek 10 - local.neon

```

≡ local.neon X
config > ≡ local.neon
1  parameters:
2
3
4  database:
5      dsn: 'mysql:host=127.0.0.1;dbname=autoservis'
6      user: root
7      password:

```

'mysql:host=localhost', jako výchozí nastavení pro hostinky, může být i jinak. '

dbname=autoservis' značí, že jméno databáze. 'user' je přihlašovací jméno do databáze a 'password' je heslo do databáze.

Další údaje se musí změnit v presenterech 'OpravyPresenter' a 'SoucastkyPresenter', kde je nutné změnit přihlašovací údaje k select menu. Ve funkci 'createComponentPostForm() Form' se nachází proměnná 'mysql' kde první údaj localhost je připojení na hosting (necháme localhost),

druhé pole je uživatel (root), třetí je heslo (je prázdné, ale v zájmu bezpečnosti se musí vyplnit), nakonec je jméno databáze (autoservis).

Obrázek 11 - mysqli proměnná připojení k databázi

```
protected function createComponentPostForm(): Form
{
    $form = new BootstrapForm;
    $form->setHtmlAttribute('class', 'container-main');
    $mysqli = mysqli_connect("localhost","root","","autoservis");
```

### 1.3.2 Popis webu

První se přihlásíme, abychom mohli upravovat a přidávat vozidla.

Obrázek 12 - hlavní stránka - přihlásit se

ID	SPZ	Výrobce	Rok výroby	Barva	Obsah motoru	Editovat	Smazat
1	7AM-0365	Ford	1992	stříbrná	1,2	<a href="#">edit</a>	<a href="#">delete</a>
2	4Z7-7130	škoda	2000	modrá	2	<a href="#">edit</a>	<a href="#">delete</a>
3	4Z2-9037	citroen	2008	černá	2,0	<a href="#">edit</a>	<a href="#">delete</a>

Přihlásíme se. Jméno je 'admin' a heslo 'secret'.

Obrázek 13 - přihlášení

Zadáme SPZ, Výrobce, Rok Výroby, barvu a obsah motoru ve tvaru, jak je psáno a klikneme na proved'. Poté se přidá do tabulky níže, kde se může editovat, nebo smazat.

Obrázek 14 - automobily

Autoservis Automobily Majitelé Opravy Součástky Typ vozu Zaměstnanci Odhlásit se

SPZ:  
 Výrobce:  
 Rok výroby:  
 Barva:  
 Obsah motoru:  
 Proved

ID	SPZ	Výrobce	Rok Výroby	Barva	Obsah motoru	Editovat	Smazat
1	7AM-0365	Ford	1992	stříbrná	1,2	<a href="#">edit</a>	<a href="#">delete</a>
2	4Z7-7130	škoda	2000	modrá	2	<a href="#">edit</a>	<a href="#">delete</a>
3	4Z2-9037	citroen	2008	černá	2,0	<a href="#">edit</a>	<a href="#">delete</a>

Poté nás přesměruje automaticky na stránku majitelů a takto pokračujeme dále. Pro vybrání barvy klikněte na černý pruh a po vyjetí z nabídky klikněte bokem. Barva se ukládá v hexadecimálním kódu.

U stránky Opravy a součástky je výběrové menu, kde vybereme zaměstnance a jméno součástky, které zkonvertuje na ID. Začátek a konec opravy napíšeme ve tvaru jak je dáno v políčku.

Obrázek 15 - opravy

Autoservis Automobily Majitelé Opravy Součástky Typ vozu Zaměstnanci Odhlásit se

Začátek opravy (Datum)  
 Konec opravy (Doba opravy)  
 Zaměstnanec  
 Součástky ID  
 Počet/ks  
 Proveďte číslo  
 Proved

id	Datum	Doba opravy	Zaměstnanec ID	Součástky ID	Počet/ks	Editovat	Smazat
1	2021-02-17 00:00:00	2021-02-17 02:00:00	1	2	1	<a href="#">edit</a>	<a href="#">delete</a>
2	2021-02-15 00:00:00	2021-02-15 01:50:00	2	2	1	<a href="#">edit</a>	<a href="#">delete</a>
3	2021-02-27 00:00:00	2021-02-28 03:00:00	3	3	1	<a href="#">edit</a>	<a href="#">delete</a>

*Editace*

Obrázek 16 - editace

Autoservis   Automobily   Majitelé   Opravy   Součástky   Typ vozu   Zaměstnanci   Odhlásit se

Začátek opravy (Datum)

d.m.yyyy h:mm (31.12.1998 23:59)

Konec opravy (Doba opravy)

d.m.yyyy h:mm (31.12.1998 23:59)

Zaměstnanec

Vyberte

Součástky ID

Vyberte

Počet/ks

Zadejte číslo

Proveď

id	Datum	Doba opravy	Zaměstnanec ID	Součástky ID	Počet/ks	Editovat	Smazat
1	2021-02-17 00:00:00	2021-02-17 02:00:00	1	2	1	<a href="#">edit</a>	<a href="#">delete</a>
2	2021-02-15 00:00:00	2021-02-15 01:50:00	2	2	1	<a href="#">edit</a>	<a href="#">delete</a>
3	2021-02-27 00:00:00	2021-02-28 03:00:00	3	3	1	<a href="#">edit</a>	<a href="#">delete</a>

Klikneme na editovat a můžeme začít editovat, poté co máme ve upraveno klikneme na proved' a vrátí vás to zpátky na předchozí stránku. Pokud nechcete editovat, klikněte na tlačítko zpět.

Obrázek 17 - editace stránka

Autoservis   Automobily   Majitelé   Opravy   Součástky   Typ vozu   Zaměstnanci   Odhlásit se

## Editace úkolu

Začátek opravy (Datum)

17.2.2021 0:00

Konec opravy (Doba opravy)

17.2.2021 2:00

Zaměstnanec

Kutil

Součástky ID

těsnění

Počet/ks

1

Proveď

Zpět

Po kliknutí na tlačítko 'Odhlásit se' vás to přesměruje na hlavní stránku.

**Obrázek 18 - Odhlásit se přesměrování na hl. stránku**

Odhlášení bylo úspěšné.

Autoservis [Přihlásit se](#)

SPZ:

Výrobce:

Rok výroby:

Barva:

Obsah motoru:

Proved

ID	SPZ	Výrobce	Rok výroby	Barva	Obsah motoru	Editovat	Smazat
1	7AM-0365	Ford	1992	stříbrná	1.2	<a href="#">edit</a>	<a href="#">delete</a>
2	4Z7-7130	škoda	2000	modrá	2	<a href="#">edit</a>	<a href="#">delete</a>
3	4Z2-9037	citroen	2008	černá	2.0	<a href="#">edit</a>	<a href="#">delete</a>

## ZÁVĚR

Myslím si, že web byl docela náročný na zpracování. Hlavně práce s novým frameworkem dala zabrat, ale šlo to. Nevím, zda by se dal web použít pro autoservis, ale rozhodně jsem se něčemu novému naučil.

Vše se ale nepovedlo tak, jak bych si představoval. Select box, si myslím, že by šel udělat i jinak, jen na framework Nette není tolik návodů jako například na CodeIgniter, ale v nějakých krocích mi značně ulehčil práci. Například jsem se nemusel starat o spojování stránek a nemusel „montovat“ části webu dohromady.

## SEZNAM OBRÁZKŮ

Obrázek 1 – seznam vlastních presenterů .....	4
Obrázek 2 – Část kódu presenteru Opravy .....	5
Obrázek 3 – kód pro získání informací do select menu .....	5
Obrázek 4 - funkce addSelect .....	6
Obrázek 5 - PostFormSucceeded .....	6
Obrázek 6 - Funkce PostFormSucceeded .....	7
Obrázek 7 - funkce actionDelete.....	7
Obrázek 8 - funkce actionEdit .....	8
Obrázek 9 - struktura aplikace .....	9
Obrázek 10 - local.neon .....	9
Obrázek 11 - myqli proměnná připojení k databázi .....	10
Obrázek 12 - hlavní stránka - přihlásit se .....	10
Obrázek 13 - přihlášení .....	10
Obrázek 14 - automobily.....	11
Obrázek 15 - opravy.....	11
Obrázek 16 - editace .....	12
Obrázek 17 - editace stránka .....	12
Obrázek 18 - Odhlásit se přesměrování na hl. stránku .....	13