

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



POLOLETNÍ PROJEKT

TUTORIÁL NA FRAMEWORK NETTE PODLE PROJEKTU TO DO LIST

ANOTACE

Pololetní projekt do předmětů Projekt programování a Dokumentace projektu, který se zabývá vytvořením tutoriálu pro framework Nette podle projektu To do list.

OBSAH

ÚVOD	4
1.1 Zadání	4
1.2 Obsah dokumentace.....	4
1.3 O aplikaci.....	4
1.4 Východiska	4
2 INSTALACE	6
2.1 Proč Nette používat a jak funguje?	6
2.2 Composer.....	6
3 ZPROVOZNĚNÍ.....	7
3.1 Kostra webu	7
4 MVP	8
4.1 Model.....	8
4.1.1 Zprovoznění modelu v projektu	8
4.2 View	10
4.2.1 Šablona	10
4.3 Presenter	12
5 ZPROVOZNĚNÍ DATABÁZE	13
6 ZÁKLADY NETTE – TVORBA PROJEKTU TO DO LIST VIDEOTUTORIÁLY	15
6.1 Co budeme vytvářet.....	15
6.2 Spojení databáze, model + use v presenteru	15
6.3 Formulář + Table records.....	15
6.4 Edit + Delete	15
7 POUŽITÉ SCRIPTY A DOPLŇKY PROJEKTU TO DO LIST.....	16
7.1 Datatables	16
7.2 AJAX.....	16
7.3 .htaccess.....	16
9 POPSANÝ DATOVÝ MODEL APLIKACE VČETNĚ ER MODELU	23
STRÁNKA NA ÚPRAVU ÚKOLU	24
ZÁVĚR.....	25
SEZNAM POUŽITÉ LITERATURY	26
SEZNAM OBRÁZKŮ	27

ÚVOD

1.1 Zadání

Naším cílem je vytvořit tutoriál na framework Nette, ve kterém se naučíme základy tohoto frameworku, podle kterého vypracujeme projekt To do list. Naším cílem je udělat vzorovou aplikaci podle tutoriálu z tohoto frameworku. S tímto frameworkem se také seznamujeme a proto na něj děláme tento tutoriál, jehož výsledkem bude webová aplikace To do list a návod na základy frameworku. Aplikace a dokumentace vznikla na škole OAUH v Uherském Hradišti.

1.2 Obsah dokumentace

Tutoriál je rozdělen do několika částí. První je úvod. Druhá a třetí kapitola se věnuje seznamování s Nette, čili instalace a základní zprovoznění krok za krokem. Čtvrtá kapitola zobrazuje a vysvětluje strukturu frameworku. Pátá se věnuje modelu MVP, logické struktuře webu, kterou se každý framework řídí. Šestá kapitola je zaměřena na práci databázi a práci s programem phpMyAdmin. Zahrnuli jsme také přídatné knihovny k projektu. Poslední kapitola obsahuje video návody, kde je podrobně vysvětlená logika kódu.

1.3 O aplikaci

Projekt byl realizován v Nette ve verzi 3.1 a v PHP 7.4 a vyšší. Webová aplikace To do list ve frameworku Nette bude určena pro zapsání a uchování různých poznámek a úkolů. Uživatel této webové aplikace bude mít na hlavní stránce tabulku s úkoly, činnostmi a poznámkami, se kterými bude moct i nadále pracovat. Uživatel je bude moct editovat, smazat či je označit za splněné. Když si bude chtít uživatel změnit datum nebo znění daného úkolu, tak ho to přesměruje na stránku s editací dané úlohy, kde může změnit jeho datum a znění. Uživatel si může také vyhledat daný úkol či zobrazit různý počet úkolů na hlavní stránce v tabulce.

1.4 Východiska

Tento tutoriál je zaměřen na lidi, kteří se chtějí naučit něco nového. Konkrétně se bude jednat o tutoriál na základy frameworku Nette. Uživatelé budou schopni vytvořit jednoduchou webovou stránku a přitom se naučit jeho základy, kdyby se mu chtěli věnovat a plně se s ním naučit. Bude to základ do tohoto frameworku. Chceme lidi seznámit se základy Nette, jelikož si myslíme, že to pro ně bude přínosné. Tutoriál obsahuje úvod, instalaci, zprovoznění, MVP, databáze a použité skripty a doplňky, které Vám pomohou s pochopením a naučením se s tímto frameworkem. Předem budete

potřebovat nainstalované PHP a to například v podobě programu XAMPP a dále důležitý program Composer, na který je zde dále odkaz a jeho podrobnější vysvětlení. Dále jsme kód psali v programovém prostředí Visual Studio Code.

Srovnání s alternativními produkty

Na framework Nette existuje několik návodů a tutoriálů podobných jako je tento. Můžeme najít tutoriály jenom na základy tohoto frameworku nebo také k vytvoření jiných aplikací a to například kalkulačky či redakčního systému. Výsledkem tohoto tutoriálu je pro naše zájemce stránka na uchovávání a zápis různých úkolů a poznámek. Tím pádem je tento samotný tutoriál originální, protože slouží k vytvoření již zmíněné stránky, kterou jsme udělali pro tento jeden konkrétní tutoriál.

Uživatelské testování

Uživatelské testování vedlo k dílčím úpravám tutoriálu, které ho zlepšily a upřesnily, když se zpětně procházel. Také jsme se snažili, aby byl srozumitelný, dobře čitelný a aby se všechno dalo bez problému naučit, pochopit a následně napsat.

Odkaz na web na hostingu: <http://todolist.cekuj.net/>

2 INSTALACE

2.1 Proč Nette používat a jak funguje?

Z vlastní zkušenosti můžeme říct, že Nette je vlastně velký balík knihoven (kolem 20) pro PHP. Základy v Nette jsou hračka, ale zároveň stále programujete v PHP a tím pádem musíte znát jeho vlastnosti. Framework vám vychází vstříc při programování i debugování za pomoci tzv. laděnky. Framework má obsáhlou komunitu, která ráda poradí s jakýmkoliv problémem a také byl zvolen jako nejpoužívanější framework v české republice. Jeho autorem a vývojářem je David Grudl, který má na starosti spoustu knihoven týkajících se Nette.

Na Nette jede hned několik významných projektů, jako je například GE Money, ČSFD, ESET a další. Tutoriál je určený pro aktuální verzi Nette 3.x a PHP 7.4 nebo novější.

2.2 Composer

Composer je jeden z klíčových nástrojů pro správu balíčků v PHP a umožňuje jednoduše instalovat přídatné knihovny do našeho projektu.

Windows instalátor: <https://getcomposer.org/Composer-Setup.exe>

Vyžaduje již předinstalované PHP. Například si můžete stáhnout PHP prostředí XAMPP.

Linux, MacOS instalátor:

K tomuto vám postačí tyto řádky kódu. Instalátor zároveň zkontroluje soubor php.ini a poté stáhne nejnovější verzi composeru composer.phar. Zadejte níže uvedené 4 očíslované řádky do příkazového řádku (v pořadí jak jdou za sebou).

```
1. php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
2. php -r "if (hash_file('sha384', 'composer-setup.php') ===
    '756890a4488ce9024fc62c56153228907f1545c228516cbf63f885e036d3
    7e9a59d27d63f46af1d4d07ee0f76181c7d3') { echo 'Installer
    verified'; } else { echo 'Installer corrupt';
    unlink('composer-setup.php'); } echo PHP_EOL;"
3. php composer-setup.php
4. php -r "unlink('composer-setup.php');"
```

3 ZPROVOZNĚNÍ

S nainstalovaným composerem lze jednoduše stáhnout a nainstalovat základní kostru webu s knihovnamy, také pod pojmem Web Project, který již obsahuje Nette Framework. K založení projektu si vytvoříme složku v kořenovém adresáři například /var/www (v Linuxu), nebo htdocs/www ve Windowsu.

Pomocí příkazového řádku spustíme příkaz:

```
composer create-project nette/web-project nette-blog
```

3.1 Kostra webu

Tutorial-na-framework-Nette-na-projektu-To-do-list/

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

Adresář www/ je určen pro ukládání kaskádových stylů, obrázků, javascriptů a podobně. Navíc je to veřejně dostupný adresář a jako jediný má povolený .htaccess soubor, který upravuje práva k přístupu. Nejdůležitější složky jsou app/. Zde najdeme soubor Bootstrap.php, ve kterém je třída, která slouží k naštvení celého frameworku, knihoven a nastavení aplikace. Další složkou, která pro nás bude důležitá je config, kde se nachází konfigurační soubory pro databázi local.neon a common.neon k přidavným nastavením knihoven (session apod.).

4 MVP

Podobně jako MVC (Model, View, Controller) má Nette (Model, View, Presenter). MVP nám slouží k oddělení logiky webové stránky, kterou vytváříme. Umožňuje nám zpřehlednit web a oddělit jednotlivé operace.

Více informací zde: <https://youtu.be/FtTvadj4KE>

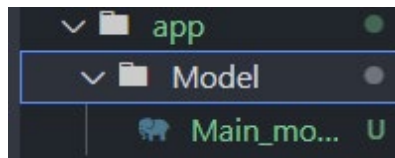
4.1 Model

Jedná se o část MVP, která slouží pro složitější operace s databází, které by nebylo příliš vhodné psát do presenteru, protože by bylo velmi obtížné se v kódu poté vyznat. My se modelem v podstatě v tomto tutoriálu nebudeme příliš zabírat, protože pro základy, které se budeme učit jej budeme využívat jen velmi zřídka. Nicméně i tak nám docela usnadní práci.

4.1.1 Zprovoznění modelu v projektu

1. Musíme ve složce app vytvořit novou složku s názvem **Model** v této složce poté vytvoříme nový soubor s názvem `Main_model.php` přesně, jak můžete vidět na obrázku.

Obrázek 1-Model



2. Musíme říct Nette, že tento soubor chceme využívat jako model. Je tedy nutné jej nadefinovat v soboru `common.neon` ve složce `config`. Výsledek by měl poté vypadat následovně:

```
parameters:
```

```
application:
```

```
    errorPresenter: Error
```

```
    mapping:
```

```
        *: App\*Module\Presenters\*Presenter
```

```
session:
```

```
    expiration: 14 days
```



```
di:
    export:
        parameters: no
        tags: no
```

```
services:
    - App\Router\RouterFactory::createRouter
    - App\Model\Main_model
```

V podstatě pouze v tomto souboru do sevice: vložíme - *App\Model\Main_model* a soubor uložit.

3. Nyní musíme model implementovat do presenteru, abychom jej mohli využívat.

```
<?php
```

```
namespace App\Presenters;
```

```
use Nette;
use App\Model\Main_model;
use Nette\Database\Context;
```

```
final class HomepagePresenter extends Nette\Application\UI\Present
er
{
    private $main_model;

    public function __construct(Main_model $main_model)
    {
        $this->main_model = $main_model;
    }
}
```

4. Posledním krokem je nalinkovat databázi do Main_model, abychom s ní v něm mohli pracovat. Výsledný model tedy bude vypadat nějak takto:

```
<?php
```

```

namespace App\Model;

use Nette;

class Main_model
{

    public function __construct(Nette\Database\Explorer $database)
    {
        $this->database = $database;
    }

```

Nyní jsme připraveni využívat model naplno.

Podrobnější rozbor zde: https://youtu.be/Q8n3V3_LxMA

4.2 View

Pod pojmem view si můžete představit všechno, co chcete na svém webu zobrazit, jedná se o seskupení šablon a html atributů, obsahuje také tagy jiného jazyka.

4.2.1 Šablona

Nette jako takové funguje na tzv. Latte šablonách, tyto šablony jsou v podstatě totéž, jako typický html dokument, nicméně má své specifické funkce a knihovny. Tyto knihovny a funkce můžete najít na oficiálním API na webu Nette. Latte šablony jsou vždy soubory typu: název_šablony.latte. Velmi důležitou operací je možnost zobrazit nějaký prvek presenteru právě v latte šabloně.

Do šablony se většinou odesílá nějaká komponenta presenteru a ve zbylých případech vkládáte kód, který si nadefinujete přímo v šabloně pomocí latte kódu.

Základní šablonou je v Nette @layout.latte jedná se o šablonu, která je nadřazena všem ostatním. Její asi nejpozoruhodnější vlastností je to, že není statická, tudíž se její kód mění, na základě toho, jakou podřazenou šablonu zrovna zobrazuje. Pojďme si to ukázat na příkladu.

Máme `@layout.latte` šablonu:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>To do list</title>
    <title>{ifset title}{include title|stripHtml} | {/ifset}Todo-
List</title>
  </head>

  <body class="body container">
    <div n:foreach="$flashes as $flash" n:class="flash, $flash-
>type">{$flash->message}</div>

    {include content}

    {block scripts}

    {/block}
  </body>
</html>
```

Celá šablona zůstává v průběhu všech operací na stránce stejná, nicméně je zde jedna část, kterou můžeme a chceme ovlivnit jednotlivými dílčími šablonami a sice část `{include content}`. Ve zkratce by se dalo říct, že vše co chcete, aby bylo pro všechny šablony společné dáváte do nadřazeného `@layout.latte` a naopak, pokud máme něco co bychom chtěli mít pouze pro jednu určitou šablonu, většinou se jedná o kód, který běžně píšeme do body v běžném HTML.

`@layout.latte` šablona pracuje tak, že `{include content}` nahradí kódem z konkrétní latte šablony, kterou chceme zobrazit.

Podřazená latte šablona vypadá takto:

```
{block content}
<p>Hello world</p>
{control postForm}

{/block}
```

Všiměte si, že v `@layout.latte` máme `{include content}` a v „podšabloně“ máme `{block content}`. Obsah `{block content}` poté nahrazuje `{include content}`.

4.3 Presenter

Jedná se o jednu z nejdůležitějších částí projektu, slouží jako prostředník mezi view a modelem, kteří o sobě nic neví. Právě proto je spojuje dohromady a vytvoří celek. Presenter je jen přejmenovaný controller v tradičním rozdělení MVC.

Každá šablona by měla mít ideálně pouze jeden presenter. Tento soubor, by měl poté vypadat nějak takto: `NázevPresenter.php`.

Je zde také možnost psát operace, které by byly normálně v presenteru do takzvaného modelu.

V takovém případě se budeme jednak lépe orientovat v presenteru a také v jednotlivých databázových operacích, je pouze nutné daný model implementovat do `config/local.neon` a pak již jen implementovat v presenteru a místo psaní dlouhého kódu, můžeme do presenteru pouze napsat název funkce, která je vytvořena v modelu, právě pro konkrétní databázovou operaci.

5 ZPROVOZNĚNÍ DATABÁZE

zprovoznění databáze v Nette není vůbec nic složitého, jedná se v podstatě pouze o dva jednoduché kroky a poté již můžete s databází směle pracovat.

1. Je potřeba, abyste měli vytvořenou databázi a věděli její název. Tento název je poté nutné zadat do souboru `local.neon` ve složce `config` projektu. Výsledek by měl vypadat následovně...

database:

```
dsn: 'mysql:host=127.0.0.1;dbname=název vaší databáze'
```

```
user: root
```

```
password: *zde vložte heslo k databázi*
```

vysvětlení: dsn: jedná se o propojení se serverem databáze, je zde nutné zadat mysql hosta v drtivé většině případů se bude jednat o 127.0.0.1 neboli localhost. Localhost je loopback adresa používaná koncovým uživatelem čili vámi. Pokud používáte phpMyAdmin ke správě databází, hosta lze zjistit přímo když si otevřeme databázi v horní liště s popisem.

Obrázek 2-phpMyAdmin



Je-li databáze uložena lokálně, mělo by být defaultní nastavení údajů stejné, jako v našem případě. V opačném případě je nutné změnit uživatele (user) a vložit heslo (password), které jsou nastaveny v nastavení přístupu do databáze.

2. Pokud se nám úspěšně povedlo nakonfigurovat `local.neon` potřebujeme ještě zavolat databázi a to buď v presenteru a nebo modelu projektu. My jako začátečníci se nebudeme zabývat modelem, ale budeme vše vykonávat přes presenter. Upravíme tedy náš `homepagePresenter` následovně:

```
<?php

namespace App\Presenters;

use Nette;
use Nette\Database\Context;
final class HomepagePresenter extends Nette\Application\UI\Present
er
{
    public $database;

    public function injectContext(Context $database)
    {
        $this->database = $database;
    }
}
```

Tímto jsme vložili naši databázi do presenteru. Konkrétně jsme ji vložili do proměnné *database*, takže teď již budeme používat jen proměnou, nikoli `$this->database`. \Rightarrow

Těmito dvěma jednoduchými kroky jsme dokázali nastavit Nette tak, aby mělo přístup k databázi a mohli jsme ji dále využívat.

!!!POZOR!!!

Je nutné tento kód zkopírovat vždy na začátek každého nového presenteru, aby bylo možné využívat databáze. Nesmíme ovšem zapomenout na přejmenování třídy na název daného presenteru, který právě využíváme. Pokud tedy mám `PostPresenter.php` tak název třídy bude následovný:

```
final class PostPresenter extends Nette\Application\UI\Presenter
```

6 ZÁKLADY NETTE – TVORBA PROJEKTU TO DO LIST VIDEOTUTORIÁLY

6.1 Co budeme vytvářet

Info zde: <https://youtu.be/FtTvadnj4KE>

6.2 Spojení databáze, model + use v presenteru

Podrobný video návod zde: https://youtu.be/Q8n3V3_LxMA

6.3 Formulář + Table records

Podrobný video návod zde: <https://youtu.be/kNmWInT6HDs>

6.4 Edit + Delete

Podrobný video návod zde: <https://youtu.be/yNUhSuONoYg>

Kód využitý v projektu naleznete zde: <https://github.com/st3panzz/Tutorial-na-framework-Nette-na-projektu-To-do-list>

7 POUŽITÉ SCRIPTY A DOPLŇKY PROJEKTU TO DO LIST

7.1 Datatables

Datatables je velmi jednoduchý nástroj pro snadné nasazení, oživení a zvýšení funkčnosti vaší html tabulky. Mezi jejich největší přednosti patří možnost upravit si jednotlivé atributy, které chceme v samotném souboru Datatables.js.

Informace o instalaci a konfiguraci naleznete zde: <https://datatables.net/>

Konkrétní soubor Datatables.js použitý v projektu naleznete zde:

<https://github.com/st3panzz/Tutorial-na-framework-Nette-na-projektu-To-do-list/blob/main/www/jquery/datatables.js>

7.2 AJAX

AJAX je velmi užitečný, jedná se v podstatě o soubor, který se využívá při práci s databází a webem. My jsme jej využili v projektu za účelem toho, aby při zaškrtnutí check-boxu se do databáze uložila určitá hodnota a podle ní pak AJAX mohl rozeznat to, jestli má být daný check-box zaškrtnutý či nikoli a to i po znovunačtení serveru.

Konkrétní soubor ajax.latte použitý v projektu naleznete zde: <https://github.com/st3panzz/Tutorial-na-framework-Nette-na-projektu-To-do-list/blob/main/app/Presenters/templates/Homepage/ajax.latte>

Více o ajax naleznete zde: https://www.w3schools.com/js/js_ajax_intro.asp

7.3 .htaccess

.htaccess je důležitý soubor frameworku, s jehož pomocí se určuje přístup k souborům a složkám webové aplikace běžným uživatelům. Tento soubor je nezbytné upravit pro správnou funkčnost a zabezpečení webu. Jeho nesprávné nastavení má nepříznivý dopad především na bezpečnost, ukradení či odcizení aplikace a nastavení na samotném hostingu.

Soubor .htaccess naleznete hned v kořenovém adresáři. Po nalezení tohoto souboru jej upravte tak, aby vypadal přesně takto:


```

<IfModule mod_rewrite.c>
RewriteEngine On

RewriteRule ^$ www/ [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_URI} !^/www/
RewriteRule ^(.*)$ www/$1
</IfModule>

```

!!!POZOR!!!

Používejte pouze v případě, že si chcete usnadnit přecházení z jednoho webu do druhého. Každý hosting si to řeší sám. Pokud byste použili tento soubor, způsobilo by to pouze více škody než užitku. Hrozí odcizení dat z webové stránky. Pro nás bude stačit spouštění webu na portu 8000.

Spustíme aplikaci příkazový řádek (cmd.exe) ve Windows. Přejdeme do složky s webem a napíšeme:

```
php -S localhost:8000 -t www
```

Tyto .htaccess soubory jsou ve vašem projektu celkem dva. Jeden je jak bylo již zmiňováno v kořenové složce projektu, a druhý ve složce /www. Druhý zmiňovaný soubor ve složce /www vypadá takto:

```

# Apache configuration file (see
https://httpd.apache.org/docs/current/mod/quickreference.html)
Require all granted

# disable directory listing
<IfModule mod_autoindex.c>
    Options -Indexes
</IfModule>

# enable cool URL
<IfModule mod_rewrite.c>
    RewriteEngine On

```

```

RewriteBase /

# use HTTPS
# RewriteCond %{HTTPS} !on
# RewriteRule .? https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]

# prevents files starting with dot to be viewed by browser
RewriteRule /\.|^\.(!well-known/) - [F]

# front controller
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule
!\.(pdf|js|mjs|ico|gif|jpg|jpeg|png|webp|svg|css|rar|zip|7z|tar|.g
z|map|eot|ttf|otf|woff|woff2)$ index.php [L]
</IfModule>

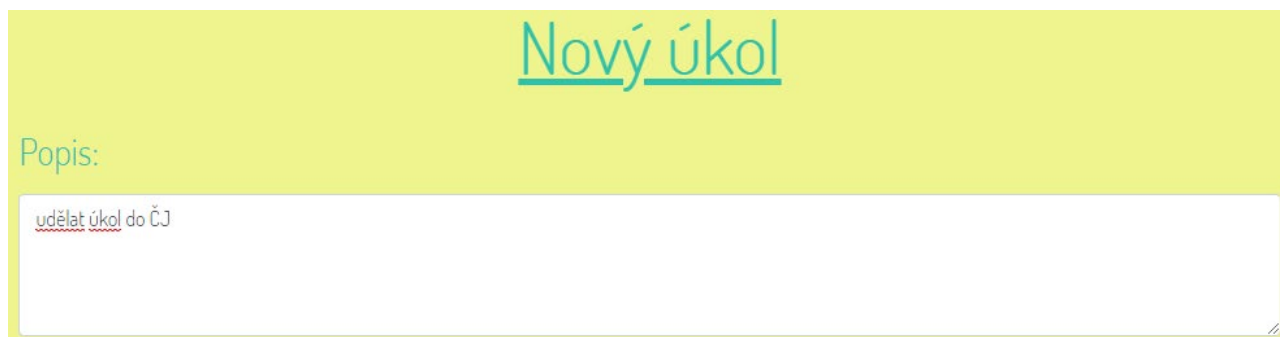
# enable gzip compression
<IfModule mod_deflate.c>
    <IfModule mod_filter.c>
        AddOutputFilterByType DEFLATE text/html text/plain
text/xml text/css application/javascript application/json
application/xml image/svg+xml
    </IfModule>
</IfModule>

```

V případě, že by se ve vaší /www složce soubor .htaccess nenacházel, tak jej prostě a jednoduše vytvoříte, jako jakýkoli jiný soubor, zkopírujete do něj kód výše a uložíte jej.

8 JEDNOTLIVĚ POPSANÉ FUNKCE WEBU

Obrázek 3-Nový úkol



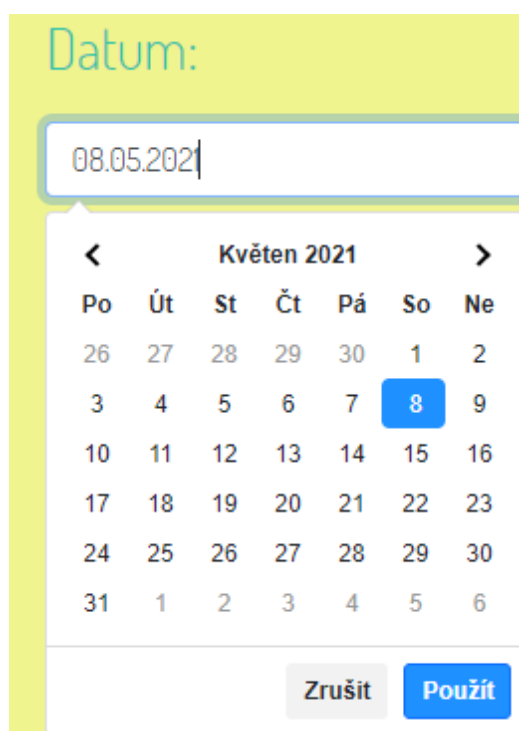
Nový úkol

Popis:

udělat úkol do ČJ

Vytvoření dané poznámky, se kterou budeme nadále pracovat.

Obrázek 4-Datum



Datum:

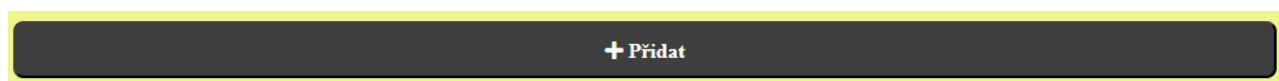
08.05.2021

Květen 2021						
Po	Út	St	Čt	Pá	So	Ne
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Zrušit Použít

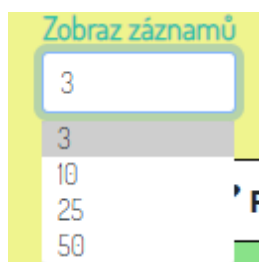
Dále je tu automatické přiřazení data nebo možný výběr jakéhokoliv jiného data z kalendáře.

Obrázek 5-Tlačítko přidat



Tlačítko pro přidání a zapsání úkolu do tabulky.

Obrázek 6-Záznamy



V tabulce se zobrazí určitý počet poznámek neboli záznamů. V základu jsou to 3 zobrazené záznamy, jinak možnost nastavit jiné předem určené, které chceme vidět na jedné stránce.

Obrázek 7-Tabulka a vyhledávání

Zobraz záznamů

3

Hledat:

udělat

🔧 Popis	🕒 Konec	✎ Editovat	🗑 Smazat	✓ Splněno
udělat DPR	2021-01-22	✎	🗑	✓
udělat úkol do ČJ	08.05.2021	✎	🗑	□

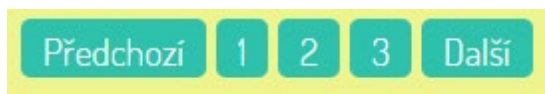
Je zde pole, které slouží pro vyhledávání jednotlivých poznámek podle názvů. Dále se tady nachází tabulka poznámek, kde se zobrazují poznámky, kde uvidíme název, datum konce, možnost úpravy jednotlivé poznámky, její smazání a odškrtnutí splněné poznámky.

Obrázek 8-Počet zobrazených záznamů

Zobrazuji 1 až 2 z celkem 2 záznamů (filtrováno z celkem 9 záznamů)

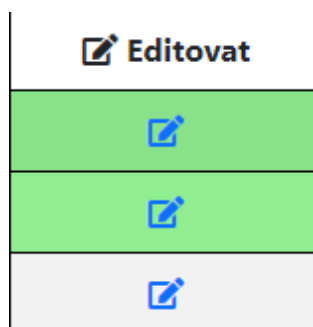
Zde se zobrazuje počet záznamů, které jsou na jedné stránce a také se zde zobrazuje celkový počet záznamů v tabulce. Dále zobrazuje záznamy podle vyhledávání.

Obrázek 9-Tlačítka pro přepínání stran tabulky



Tlačítka pro přepínání stran tabulek záznamů: předchozí, 1,2,3 a další.

Obrázek 10-Úprava poznámky v tabulce



Možnost úpravy poznámky v tabulce po kliknutí na ikonu.

Obrázek 11-Editace úkolu

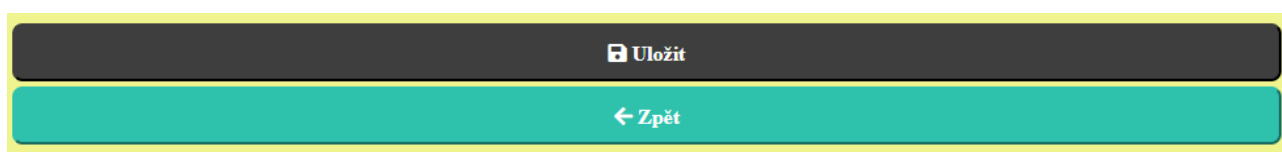
Editace úkolu

Popis:

udělat úkol do ČJ

Upravení znění a názvu poznámky či úkolu.

Obrázek 12-Tlačítka







Tlačítka pro uložení úprav jednotlivých poznámek a vrácení se zpět na hlavní stránku.

Obrázek 13-Tabulka splněno

✓ Splněno
<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>
<input type="checkbox"/>

Možnost odškrtnutí splněné poznámky či úkolu a následné označení celého řádku zelenou barvou.

Obrázek 14-Tabulka smazání

 Smazat




Následuje tabulka s ikonou koše, kde se po kliknutí na ni odstraní celá poznámka či úkol.

9 POPSANÝ DATOVÝ MODEL APLIKACE VČETNĚ ER MODELU

Úvodní stránka

- zadání úkolu
- zadání data
- přidání úkolu
- tabulka s úkoly a jejich zobrazený počet
- vyhledávací pole
- editace úkolů, smazání a odškrtnutí za splnění
- Zobrazuje počet záznamů a překliknutí na další část tabulky úkolů

Obrázek 15-Wireframe1

Nový úkol/poznámka/činnost

Popis:

Sem napište váš úkol:

Datum:

08.02.2021

Přidat

Zobraz záznamů:

3

Hledat:

Popis	Konec	Editovat	Smazat	Splněno
Jet nakoupit	08.02.2021			✓
Udělat úkol do ČJ	10.02.2021			✓
Povysávat	09.02.2021			✓

Zobrazuji 1 až 3 z celkem 5 záznamů

Předchozí 1 2 3 Další

Stránka na úpravu úkolu

- editace daného úkolu
- upravení data
- tlačítko uložení a tlačítko zpět, které nás přesměruje na hlavní stránku

Obrázek 16-Wireframe2

Editace úkolu

Popis:

Datum:

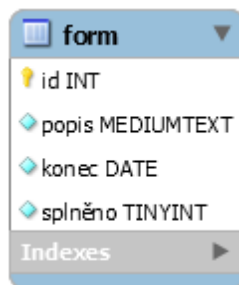
11.02.2021

Uložit

← Zpět

ER-Model

Obrázek 17-ER-Model



Jednoduchý ER-Model tabulky, kde se nachází id INT a dále popis ve formátu text, konec (konečné datum úkolu) ve formátu date a splněno s TINY INT hodnotou.

ZÁVĚR

Naším výsledným produktem je tutoriál na framework Nette, který jsme vytvořili na základě práce na projektu To-do-list, který jsme s pomocí tohoto frameworku vytvořili a také se s ním seznámili a naučili, jak s ním pracovat a využívat ho.

Tento tutoriál jsme připravili pro každého, kdo se s tímto frameworkem chce naučit či seznámit a udělat si podle něj vzorovou aplikaci na zaznamenávání různých úkolů či činností. Doufáme, že Vám tento tutoriál pomůže se seznámením s frameworkem Nette, ať už si budete chtít zkusit jednu konkrétní věc nebo udělat celou vzorovou aplikaci podle tohoto tutoriálu, který vás bude tímto frameworkem provázet a pomáhat. Také bude upozorňovat na případné rizika nebo různé věci, se kterými se můžete při práci s tímto frameworkem setkat.

Myslíme si, že se nám podařilo vytvořit vzorovou aplikaci a potom jsme podle ní začali pracovat na tomto tutoriálu, který vás má seznámit s tímto frameworkem. Je na každém z vás, co se v tomto tutoriálu naučíte nebo zkusíte a co vás zaujme natolik, že si najdete dokumentaci a ponoříte se do světa IT naplno.

Tutoriál je rozdělen do několika kapitol, ze kterých si můžete vybrat tu, která vás nejvíce zaujme nebo si také zkusit projít celý tento tutoriál a napsat stejnou nebo podobnou aplikaci na zaznamenávání poznámek s tabulkou a dalšími různými funkcemi. Tento tutoriál má sloužit pro napsání vzorové aplikace ve frameworku Nette a celkovým popsáním postupu práce na vytvoření tohoto tutoriálu na projektu To do list pro zaznamenání poznámek, úkolů či různých činností.

SEZNAM POUŽITÉ LITERATURY

- (1) Nette: Začínáme s Nette [online]. [cit. 2021-4-30]. Dostupné z:
<https://doc.nette.org/cs/3.1/quickstart/getting-started>
- (2) ITnetwork: Nette sandbox a IDE [online]. [cit. 2021-4-30]. Dostupné z:
<https://www.itnetwork.cz/php/nette/zaklady/tutorial-nette-php-framework-zaciname>

SEZNAM OBRÁZKŮ

Obrázek 1-Model	8
Obrázek 2-phpMyAdmin	12
Obrázek 3-Nový úkol.....	15
Obrázek 4-Datum.....	15
Obrázek 5-Tlačítko přidat	16
Obrázek 6-Záznamy	16
Obrázek 7-Tabulka a vyhledávání.....	16
Obrázek 8-Počet zobrazených záznamů	16
Obrázek 9-Tlačítka pro přepínání stran tabulky	17
Obrázek 10-Úprava poznámky v tabulce.....	17
Obrázek 11-Editace úkolu.....	17
Obrázek 12-Tlačítka.....	17
Obrázek 13-Tabulka splněno	18
Obrázek 14-Tabulka smazání.....	18
Obrázek 15-Wireframe1	21
Obrázek 16-Wireframe2.....	22
Obrázek 17-ER-Model.....	22