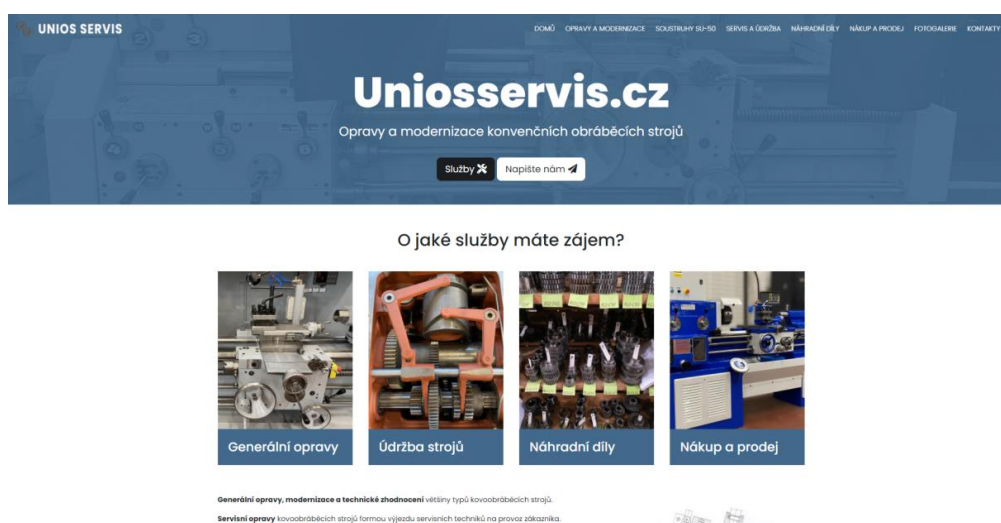


ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

Webová stránka Uniosservis.cz

Lukáš Sukeník



Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování

Třída: IT4

Školní rok: 2020/2021

Poděkování

Rád bych poděkoval především své rodině za to, že mě vždy vedla k informatice. Také bych chtěl poděkovat Danielu Scherzynovi za to, že mě zasvětil do světa informatiky.

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2020

podpis autora práce

ANOTACE

Dokumentace je zaměřena na tvorbu firemních webových stránek. Práce vznikla jako zadání webových stránek od firmy UNIOS SERVIS s.r.o. Mezi hlavní požadavky patří zjednodušení a sjednocení vzhledu, vytvoření plně responzivních stránek s rychlým načítáním a dále také oprava chyb starých stránek. Webové stránky byly vytvořeny v redakčním systému django CMS (Content Management System). Při vytváření stránek bylo využito modulů (plugins), které výrazně usnadní a zefektivní práci na webu.

KLÍČOVÁ SLOVA

Webová stránka, firemní webové stránky, Django, django CMS, CMS, Python

ANNOTATION

Documentation is focused on the creation of corporate websites. The final work was created as an assignment of web page from company UNIOS SERVIS s.r.o. The main requirements includes simplification and unification of site appearance, create fully responsive website with fast page loading and last but not least error correction of old website. Web page was created in the django CMS content management systém. It was used modules (plugins), which will significantly facilitate and increase efficiency of creating websites.

KEYWORDS

Website, corporate websites, Django, django CMS, CMS, Python

OBSAH

ÚVOD	5
1 ZADÁNÍ WEBOVÉ STRÁNKY	6
2 VYUŽITÉ TECHNOLOGIE.....	8
2.1 DJANGO CMS	8
2.2 BOOTSTRAP MODULY	9
2.3 LIGHTGALLERY	11
2.4 ALDRYN FORMS	12
3 ZPŮSOBY ŘEŠENÍ.....	14
3.1 ÚVODNÍ ANALÝZA.....	14
3.2 PŘÍPRAVA	14
3.3 TVORBA STRÁNEK.....	15
3.3.1 Příprava grafických prvků stránek.....	15
3.3.2 Instalace webu a zprovoznění modulů.....	16
3.3.3 Kódování stránek.....	17
3.3.4 Úprava stránek pomocí front-end editoru	18
3.3.5 Programování webu.....	20
3.3.6 Finální úpravy	22
4 VÝSLEDKY ŘEŠENÍ.....	23
4.1 UKÁZKA WEBOVÉ STRÁNKY	23
4.2 ZPROVOZNĚNÍ STRÁNEK	26
ZÁVĚR	27
SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ.....	28

ÚVOD

Tuto problematiku jsem si zvolil, jelikož webové stránky jsem začal tvořit již na základní škole. Vždy mě zajímalo, co vše stojí za tím vytvořit úspěšný web a jak webové stránky vůbec fungují. Z psaní statických webových stránek jsem se po nějaké době dostal na psaní dynamických webových aplikací. Později jsem objevil i CMS a frameworky, které dost zjednodušují práci na webových stránkách. Také jsem začal s tvorbou několika projektů, protože mě tato problematika baví už od samého začátku natolik, že se v ní chci neustále vyvíjet a posouvat dále.

Hlavním cílem práce bylo vytvořit funkční webové stránky pro firmu UNIOS SERVIS s.r.o., jíž jsem byl osloven a zároveň si tento projekt vybral jako svou závěrečnou studijní práci. Webové stránky měly vycházet z jejich starších stránek, ale měly odstranit problémy, které se na nich vyskytovaly. Dále bylo třeba zvýšit přehlednost, vylepšit responzivitě a celkově zlepšit funkčnost stránek. Firma také měla minimální požadavky, které se měly promítnout na webových stránkách.

Nejdříve jsem zvažoval i využití čistého Djanga, anebo systému Wordpress. Nakonec jsem si však pro tyto stránky zvolil systém django CMS, který je rychlý, čistý, přehledný. V neposlední řadě je django CMS díky jeho funkcím opravdu silným nástrojem pro tvorbu stránek, které nevyžadují velkou interakci s uživatelem.

Tato práce popisuje tvorbu webových stránek za pomoci djanga CMS. Poukazuje na kroky, kterými jsem se řídil při tvorbě webu. Zdůvodňuje využití systému django CMS a modulů, kterých jsem při tvorbě stránek využil. Dále práce poukazuje na výhody právě tohoto systému a jak jich bylo využito. V neposlední řadě je v práci nastíněno zprovoznění stránek se systémem django CMS.

1 ZADÁNÍ WEBOVÉ STRÁNKY

Zadáním mého závěrečného projektu bylo vytvořit funkční webové stránky pro firmu **UNIOS SERVIS s.r.o.** Tato firma se zabývá především opravami a modernizací konvenčních obráběcích strojů. Dále také jejich údržbou a prodejem strojů či náhradních dílů.

S předešlými webovými stránkami již v dnešní době firma není moc spokojena, proto se mě také rozhodla oslovit. Důvody jejich nespokojenosti jsou následující:

- vzhled stránek,
- horší responzivita webu na menších rozlišeních,
- rychlost načítání,
- občasné chyby na webových stránkách,
- v případě problémů na stránkách zdlouhavé řešení problémů ze strany tvůrců stránek.



Obrázek 1.1: Ukázka předešlých stránek

Samozřejmě jsem dostal pár bodů, které musely mé webové stránky splnit:

- na stránkách se měly objevit firemní barvy,
- měl jsem za úkol zjednodušit a sjednotit vzhled stránek,
- vytvořit plně responzivní stránky,
- zlepšit rychlost načítání,
- opravit všechny menší chyby z předcházejících stránek,
- vytvořit jednodušší animace s logem firmy,
- vytvořit kontaktní formulář, který by firmě výrazně zjednodušil práci s přesměrováním na různé e-mailové adresy.

2 VYUŽITÉ TECHNOLOGIE

Základem webových stránek je redakční systém django CMS, který vychází z populárního frameworku Django. Ten je napsán v Pythonu. Python má jednoduchou a čistou syntaxi a je dnes ve světě často používaným programovacím jazykem. Dále je využíváno různých rozšíření nebo také modulů (plugins) pro redakční systém.

2.1 Django CMS

Redakční systém **django CMS** je open-source software (software s otevřeným zdrojovým kódem), který jsem zvolil jako základ webových stránek. Ze začátku jsem se rozhodoval také mezi čistým Djangem a redakčním systémem Wordpress.

Čisté Django jsem nezvolil kvůli tomu, že tvorba stránek v django CMS je mnohem rychlejší, jednodušší a v mém případě se jednalo o tvorbu stránek, jenž nevyžadují téměř žádnou interakci s uživatelem. Wordpress jsem si nevybral kvůli jeho pomalejší rychlosti, také kvůli jeho větší popularitě, díky které je snadnějším cílem hackerů.

Mezi důvody, proč jsem zvolil právě django CMS patří:

- jednoduchost a přehlednost systému,
- rychlost,
- snadné úpravy obsahu.

Určitě je dobré dát si pozor na to, jakou verzi zrovna používáte. Před jakoukoli tvorbou stránek se ujistěte, že verze djanga CMS, Djanga i Pythonu, které chcete využívat podporují všechny moduly, jenž chcete později využívat. Ze začátku jsem s tímto měl problém i já, konkrétně když mi nefungovaly Aldryn Forms.

Tento problém jsem chtěl řešit snížením verze djanga CMS tudíž i samotného Djanga. Nakonec jsem ale však zůstal u verze djanga CMS 3.7.4, Djanga 3.0.10 a Pythonu 3.8.5. Učinil jsem tak díky tomu, že jsem zprovoznil všechny potřebné moduly. Dále se to také vyplátilo díky budoucím aktualizacím, tedy tomu, že se potřebná aktualizace na vyšší verzi výrazně oddálí.

Django CMS má však jeden zásadní problém. Tím je jeho výrazně slabší komunita (např. v porovnání s Wordpress). Důsledkem pak je výrazně méně pluginů (rozšíření) a menší podpora jak od samotných tvůrců systému, tak i komunity.



Obrázek 2.1: Logo django CMS

2.2 Bootstrap moduly

Velmi důležitou komponentou, již jsem využil, jsou Bootstrap moduly. Bootstrap je software, který umožňuje tvorbu responzivních webových stránek s využitím návrhářských šablon založených na HTML, CSS a JavaScriptu. Tyto moduly jsou kompatibilní s většinou verzí django CMS. Já využil verze 1.6.0. Umožňují vkládat nebo upravovat elementy stránek přímo na webových stránkách.

Jsou velmi jednoduché na instalaci. Stačí nainstalovat balíček `djangoCMS-bootstrap4` pomocí příkazové řádky. Poté přidat názvy modulů, které chcete využívat, do souboru `settings.py`, konkrétně do listu s názvem `INSTALLED_APPS`. Pak už jen stačí použít poslední migrace.

```
pip install djangoCMS-bootstrap4
```

(instalace balíčku)

```
INSTALLED_APPS = [  
    'djangoCMS_bootstrap4',  
    'djangoCMS_bootstrap4.contrib.bootstrap4_alerts',  
    'djangoCMS_bootstrap4.contrib.bootstrap4_badge',  
    'djangoCMS_bootstrap4.contrib.bootstrap4_card',  
    'djangoCMS_bootstrap4.contrib.bootstrap4_carousel',  
    'djangoCMS_bootstrap4.contrib.bootstrap4_collapse',  
]
```

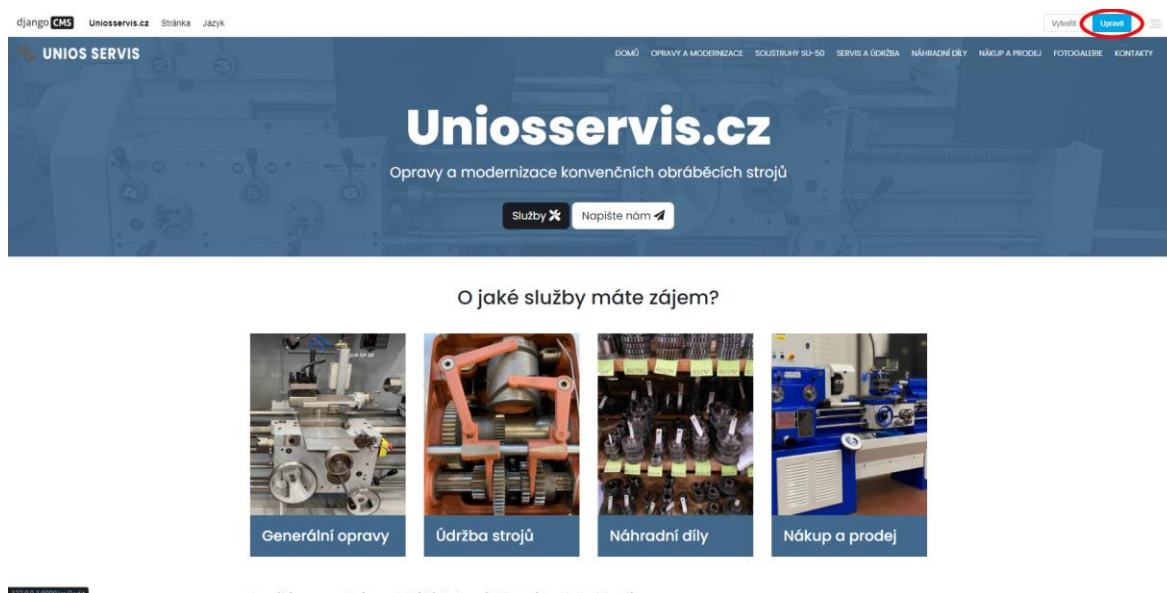
```
'djangocms_bootstrap4.contrib.bootstrap4_content',  
'djangocms_bootstrap4.contrib.bootstrap4_grid',  
'djangocms_bootstrap4.contrib.bootstrap4_jumbotron',  
'djangocms_bootstrap4.contrib.bootstrap4_link',  
'djangocms_bootstrap4.contrib.bootstrap4_listgroup',  
'djangocms_bootstrap4.contrib.bootstrap4_media',  
'djangocms_bootstrap4.contrib.bootstrap4_picture',  
'djangocms_bootstrap4.contrib.bootstrap4_tabs',  
'djangocms_bootstrap4.contrib.bootstrap4_utilities',  
]
```

Settings.py 2.1: Bootstrap moduly

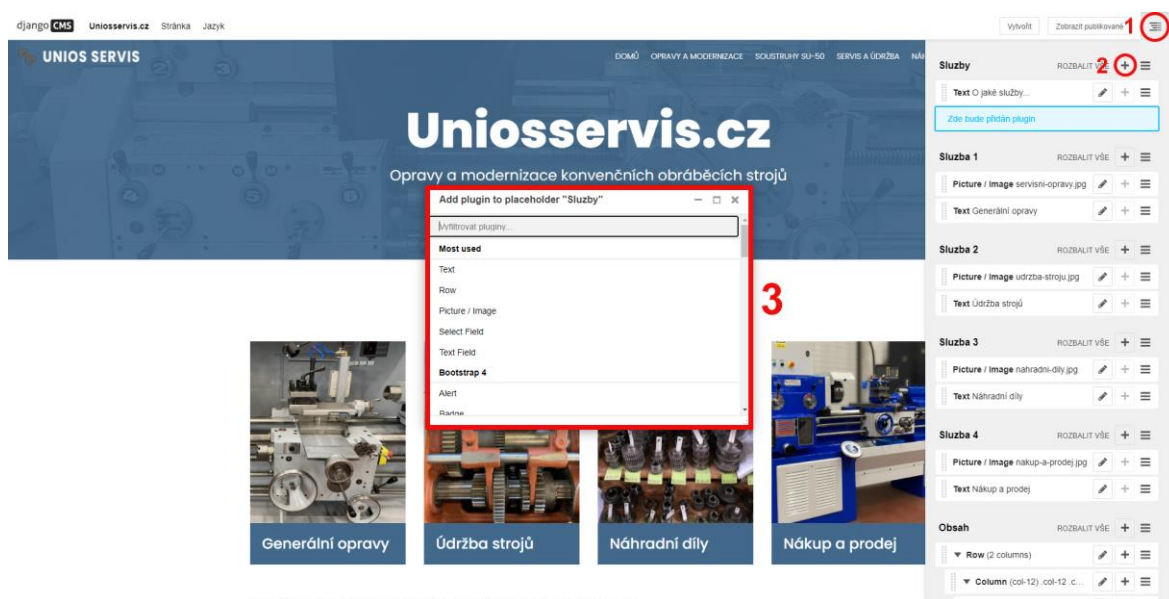
```
python manage.py migrate
```

(spuštění migrací)

Po restartování webového serveru by mělo být možné využít všech Bootstrap modulů. Stačí být přihlášený jako administrátor, v pravém horním rohu kliknout na tlačítko „Upravit“, následně kliknout na tlačítko „Zobrazit strukturu“ a potom tlačítkem „Plus“ vybrat požadovaný modul, který chcete na stránky vložit. Potom se otevře okno, ve kterém si zvolíte požadované parametry a následně už stačí jen potvrdit pomocí tlačítka „Uložit“.



Obrázek 2.2: Úpravy obsahu



Obrázek 2.3: Přidání nového modulu

2.3 LightGallery

LightGallery je modulem pro django CMS, který dokáže vytvořit upravovatelnou, modulární a plně responzivní galerii. Toto rozšíření je inspirováno Lightbox galerií (po kliknutí na obrázek na vás vyskočí vyskakovací okno s obrázky v prohlížeči), která je napsána v JavaScriptu. Tento modul jsem využil na stránce Fotogalerie.

Instalace tohoto modulu je také jednoduchá. Je však důležité dát si pozor na to, aby verze byla kompatibilní s verzí django CMS. Instalace je naprosto totožná s instalací Bootstrap modulů. Nejprve je potřeba nainstalovat balíček djangocms-light-gallery. Poté plugin zaregistrovat v souboru settings.py (do *INSTALLED_APPS* přidat 'light_gallery'). Nakonec stačí spustit migrace.

```
pip install djangocms-light-gallery (instalace balíčku)
```

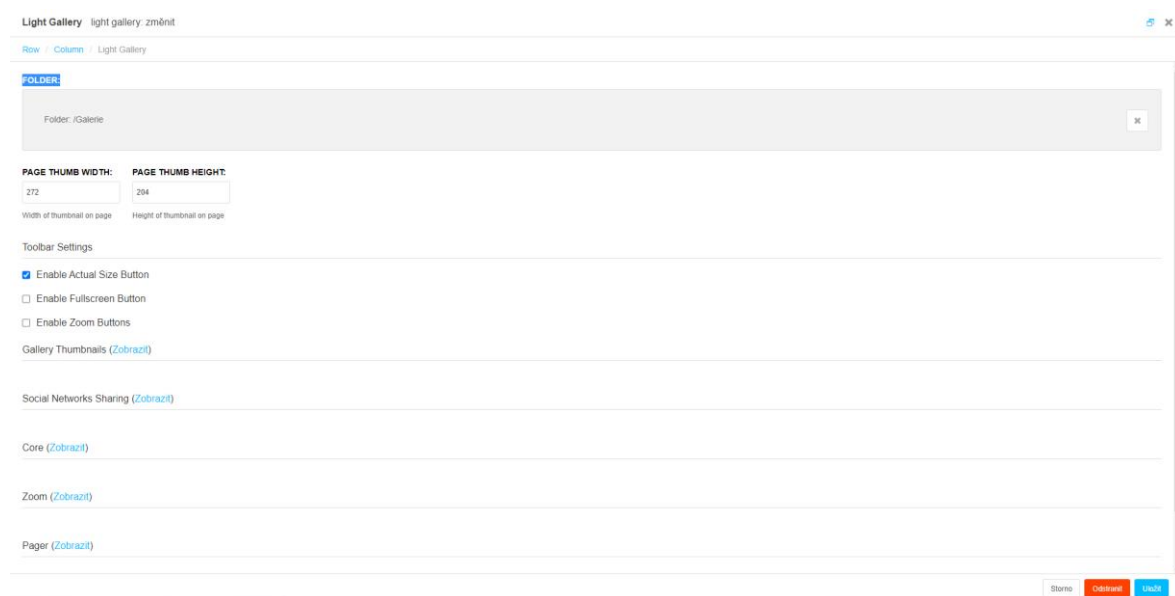
```
INSTALLED_APPS = [
    'light_gallery',
]
```

Settings.py 2.2: LightGallery

python manage.py migrate

(spuštění migrací)

Využití LightGallery je naprosto identické s využitím Bootstrap modulů. Jakmile jste přihlášen jako administrator, v pravém horním rohu kliknete na tlačítko „Upravit“, následně kliknete na tlačítko „Zobrazit strukturu“ a poté tlačítkem „Plus“ zvolíte LightGallery. Teď už je jen potřeba vybrat složku s obrázky v Django fileru (souborovém manažeru), které chcete mít v galerii. Součástí je i jiné pokročilé nastavení.



Obrázek 2.4: Nastavení LightGallery

2.4 Aldryn Forms

Aldryn Forms je rozšíření pro django CMS, které umožňuje vytváření flexibilních HTML formulářů. Dovoluje je integrovat přímo na webových stránkách. Data se následně ukládají do databáze nebo mohou být odesílány do e-mailové schránky.

Na stránku Kontakty jsem potřeboval modul, který by byl schopný vytvořit formulář, jenž dokáže ukládat hodnoty přímo do databáze Djanga. Vyzkoušel jsem asi pět pluginů, ale ani jeden z nich nefungoval. Příčinou bylo to, že všechny tyto pluginy byly kompatibilní pouze s verzemi Djanga 2.2 a nižší. Zkoušel jsem proto různé verze a narazil jsem na Aldryn Forms pre-release verze 6.2.0. I přesto, že se jednalo pouze o pre-release, který zajišťoval

kompatibilitu s Django 3.0 a výš, všechno mi v této verzi fungovalo. Po pár týdnech potom byla vydána oficiální stabilní verze 6.2.1, kterou využívám dodnes.

Pro zprovoznění Aldryn Forms je nutné tento balíček nejprve nainstalovat pomocí příkazové řádky. Poté zaregistrovat všechny balíčky, které jsou potřeba ke správnému chodu pluginů (balíček `absolute`, `aldryn_forms`, `aldryn_forms.contrib.email_notifications`, `emailit` a `filer`). Následně je třeba spustit migrace a nadefinovat e-mailový backend.

```
pip install aldrin-forms
```

(instalace balíčku)

```
INSTALLED_APPS = [  
    'absolute',  
    'aldryn_forms',  
    'aldryn_forms.contrib.email_notifications',  
    'emailit',  
]
```

Settings.py 2.3: Aldryn Forms

```
python manage.py migrate
```

(spuštění migrací)

Využití modulů zůstává pořád stejné. V administraci stačí kliknout na tlačítko „Upravit“, poté „Zobrazit strukturu“, „Přidat“ a následně vybrat „Form“. Poté už do formuláře stačí jen vkládat pole (field) a tlačítka (button) formuláře dle potřeby.

3 ZPŮSOBY ŘEŠENÍ

V této kapitole popisuji základní postupy při vytváření webových stránek. Také to, jak jsem postupoval u tvorby.

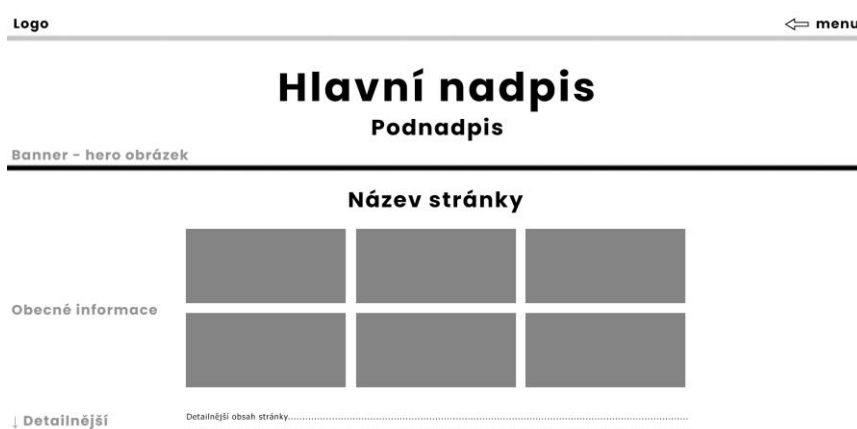
3.1 Úvodní analýza

Ještě před začátkem realizace mého projektu jsem se firmy dotázal na jejich představy a požadavky webových stránek (viz kapitola 1). Probrali jsme nabízené služby, cílovou skupinu zákazníků, rozsah stránek, grafiku, funkčnost služeb. Zkoumal jsem staré webové stránky, dle kterých jsem se občas inspiroval. Také jsem hledal všechny chyby, kterým jsem se následně snažil vyvarovat. Následně jsem zkoumal konkurenci, pro získání různých inovací či nápadů pro pozdější realizaci webových stránek.

3.2 Příprava

V této fázi jsem začal se shromažďováním všech použitelných materiálů. Pokračoval jsem návrhem UX/UI (architektury a grafického) designu stránek v grafickém editoru.

Výsledkem UX (User Experience) designu bylo vytvoření velmi jednoduché orientační struktury stránek, logického uspořádání odkazů, celkového uspořádání informací v rámci webu (od obecným k detailním) a v neposlední řadě vzájemné provázanosti informací. To vše ovlivňuje přehlednost webových stránek, jejich použitelnost a poskytuje základ pro zajištění kvalitní optimalizace pro vyhledávače.



Obrázek 3.1: UX design

Výsledkem UI (User Interface) designu bylo vytvoření minimalistického designu stránek, který vychází z firemních barev. Hlavním prvkem je ozubené kolo, jenž je umístěno v logu firmy kde představuje strojírenství. Responzivnost webu je zajištěna díky Bootstrap prvkům, které jsou na stránkách hojně využity.



Obrázek 3.2: UI design

3.3 Tvorba stránek

V této fázi začalo samotné vytváření webových stránek. V dalších bodech popisuji, jak jsem postupoval při vytváření stránek v redakčním systému django CMS. Kapitoly obsahují přípravu grafických prvků, instalaci webu a zprovoznění modulů, kódování stránek, úpravu obsahu pomocí front-end editoru, programování stránek a finální úpravy.

3.3.1 Příprava grafických prvků stránek

Úplně prvním krokem byla příprava všech grafických prvků stránek, které jsem věděl, že později využiji. Pro firmu jsem již před nějakou dobou vytvořil firemní logo, z kterého jsem dost využíval ozubené kolo. Dále bylo potřeba upravit obrázky, které mi byly zaslány pro následné vystavení na stránkách. Většinou se jednalo o ořez obrázku, občas i sofistikovanější úpravy. V neposlední řadě jsem zpracoval loga firem, které jsou vystaveny v sekci Spolupracujeme.

3.3.2 Instalace webu a zprovoznění modulů

Dalším krokem byla instalace a vytvoření nového virtuálního prostředí (virtual environment). To je nutné využít, protože každý projekt v Pythonu vyžaduje jiné závislosti (dependencies).

<code>python -m pip install --user virtualenv</code>	(instalace virtuálního prostředí)
<code>python -m venv env</code>	(vytvoření virtuálního prostředí)
<code>source env/bin/activate</code>	(aktivace virtuálního prostředí)

*Linux

<code>py -m pip install --user virtualenv</code>	(instalace virtuálního prostředí)
<code>python -m venv env</code>	(vytvoření virtuálního prostředí)
<code>env\Scripts\activate.bat</code>	(aktivace virtuálního prostředí)

*Windows

Pokračoval jsem instalací django CMS, s kterým se nainstalují všechny potřebné závislosti. Po instalaci stačilo jedním příkazem spustit development server a mohl jsem začít tvořit webové stránky.

<code>pip install.djangocms-installer</code>	(instalace django CMS)
<code>djangocms -f -p . testovaci_stranka</code>	(vytvoření nového projektu)
<code>cd testovaci_stranka</code>	(přesunutí do adresáře webu)
<code>python manage.py runserver</code>	(spuštění development serveru)

Velmi důležitým krokem, jak již název napovídá bylo zprovoznění všech modulů, které jsem chtěl na stránkách využít. Jednotlivé instalace jsou rozebrány v kapitole 2.

3.3.3 Kódování stránek

Začal jsem vytvářet první kostru stránek. Logicky jsem začal těmi částmi stránek, které jsou pokaždé stejné (opakují se) a dají se neustále využívat (záhlaví, navigace, spolupráce a zápatí).

Právě proto existují v django CMS tzv. templates (šablony), které výrazně zrychlí a zjednoduší práci. Další výhodou šablon je rychlá úprava HTML kódu pouze na jednom místě. Změny se pak mohou provést na všech stránkách, které této šablony využívají. V šablonách jsem nejvíce využíval bloky (*block* a *endblock*). Ty se dají následně přepsat v šabloně, která závisí na nějaké jiné. To znamená, že takto se dá upravovat pouze část šablony. Kromě šablon existují také tzv. includes, které zajišťují import zdrojového kódu z jiného souboru.

Následujícím užitečným prvkem je *placeholder*, který slouží k pojmenování sekce ve front-end editoru. Další prvek, který je taktéž užitečný, je *page_attribute*. Ten dokáže změnit titulek podle toho, jaký má stránka název ve frond-end editoru.

Výrazně užitečným modulem pak je *django.contrib.staticfiles*, který se stará o všechny statické soubory velmi intuitivně. Místo klasického zadávání celé cesty k souboru stačí pouze odkázat na statickou cestu, případně cestu k souboru v ní a Django si s nalezením požadovaných souborů vždy bez jakýchkoliv problémů poradí.

```
{% load cms_tags menu_tags sekizai_tags cms_static static %}
<!doctype html>
<html lang="{{ LANGUAGE_CODE }}">
  <head>
    <title>Uniosservis.cz | {% page_attribute 'page_title' %}</title>
    <!-- Meta tagy a CSS -->
    <link rel="icon" type="image/png" sizes="16x16" href="{% static 'img/favicon.ico' %}">
    <link rel="stylesheet" href="{% static 'css/styles.css' %}">
    {% render_block "css" %}
  </head>
  <body>
    {% cms_toolbar %}

    {% include "includes/nav.html" %}
```

```
{% block header %}
<div class="hero hero-pages mb-5">
  <div class="banner-nadpis text-white text-center">
    <h1>{% page_attribute 'page_title' %}</h1>
  </div>
</div>
{% endblock header %}

<div class="preloader">
  
</div>

<main class="container">
  {% block content %}{% endblock content %}
</main>

{% include "includes/collaboration.html" %}

<!-- Footer -->

<!-- JavaScript -->
<script src="{% static 'js/main.js' %}"></script>
{% render_block "js" %}
</body>
</html>
```

Citace kódu 3.1: Šablona base.html

3.3.4 Úprava stránek pomocí front-end editoru

Pokračoval jsem vytvořením jednotlivých stránek ve front-end editoru. Po přihlášení jako administrátor stačí kliknout na tlačítko „Stránka“, poté „Vytvořit stránku“ a dále „Nová stránka“ a nakonec vyplnit políčka požadovanými daty.

Upravovat jednotlivé stránky je možné pouze s administrátorskými pravomocemi. Stačí kliknout v pravém horním rohu na tlačítko „Upravit“, následně na tlačítko „Zobrazit strukturu“. Téměř všechn obsah byl na stránkách vytvořen tímto způsobem. Nejvíce jsem využíval už dříve zmiňovaných Bootstrap modulů, díky kterým je stránka responzivní. Používání editoru obsahu je velmi intuitivní.

I přes to, že jsem chtěl, aby celý obsah byl editovatelný díky front-end editoru, našly se i problémy, které jsem touhle metodou nemohl řešit. Například u domovské stránky. Potřeboval jsem zařídit to, aby po kliknutí na jakoukoliv část kartičky byl uživatel přesměrován na další stránku. To jsem musel vyřešit pomocí šablony, ve které jsem využil dříve zmiňovaného placeholderu.

```
{% extends "base.html" %}
{% load cms_tags %}

{% block title %}{% page_attribute "page_title" %}{% endblock title %}

{% block header %}
<div class="hero hero-home mb-5">
  <div class="banner-nadpis text-white text-center">
    <h1>Uniosservis.cz</h1>
    <div id="slogan">Opravy a modernizace obráběcích strojů</div>
    <!-- Tlačítka -->
  </div>
</div>
{% endblock header %}

{% block content %}
{% placeholder "sluzby" %}
<div class="row sekce mb-5">
  <div class="col-xs-12 col-md-3">
    <a href="/opravy-a-modernizace">{% placeholder "sluzba 1" %}</a>
  </div>
  <div class="col-xs-12 col-md-3">
    <a href="/servis-a-udrzba">{% placeholder "sluzba 2" %}</a>
  </div>
  <div class="col-xs-12 col-md-3">
    <a href="/nahradni-dily">{% placeholder "sluzba 3" %}</a>
  </div>
  <div class="col-xs-12 col-md-3">
    <a href="/nakup-a-prodej">{% placeholder "sluzba 4" %}</a>
  </div>
</div>
{% placeholder "content" %}
{% endblock content %}
```

Citace kódu 3.2: Šablona home.html

3.3.5 Programování webu

Funkční prvky stránek jsem řešil pomocí populární javascriptové knihovny jQuery. Prvním problémem, který jsem musel řešit byla aktivace a deaktivace černého pozadí za menu v případě, že uživatel se posune na stránce o kousek dolů. Toho jde celkem snadno dosáhnout tím, že při události *scroll* se zkontroluje pomocí metody *scrollTop()*, zdali se opravdu uživatel posunul na stránce dolů. Pokud ano, metoda *toggleClass()* přidá třídu *scrolled* značce *nav*. Třída *scrolled* je pak v CSS nadefinovaná na tmavě šedé pozadí.

Druhým problémem byla aktivace navigační lišty na zařízeních s nižším rozlišením. To jsem vyřešil následujícím způsobem. Po kliknutí na tlačítko, který jde poznat pomocí *data-trigger*, se tlačítku přidá případně odstraní třída *show* a zároveň u body se přidá či odstraní třída *mobile-nav-active*. Tyto třídy jsou nadefinovány v CSS tak, aby responzivně zobrazily menu pro mobily. Také po kliknutí na třídu *animated-button* se u této třídy přidá nebo odstraní třída *open*, jenž zajišťuje animaci navigační lištičky u mobilů.

Následující problém musel řešit tlačítko, které se zobrazí v pravém dolním rohu v případě, kdy je stránka delší. Toto tlačítko mělo uživatele vrátit na samotných vrchol stránky. Vše jde prakticky vyřešit jednou podmínkou, kdy se porovná např. jestli uživatel se na stránce posunul o více než o výšku okna vydělenou dvěma. Poté stačí ID *scrollToTop* ukázat (metoda *show()*) nebo pomocí větve *else* schovat (metoda *hide()*). Druhá funkce je obdobná, akorát stačí přidat event *scroll* a buďto ID zesílit nebo zmizet pomocí metod *fadeIn()* a *fadeOut()*. V neposlední řadě bylo potřeba po kliknutí na třídu *scrollToTop* (na tlačítko) posout uživatele na samý vrchol stránky. Toho jsem dosáhl tak, že po kliknutí funkce vytvoří animaci pro *html* a *body* s parametrem *scrollTop* nastaveným na nulu – vrchol stránky. Funkce má zpoždění (*delay*) 50 milisekund.

Další bylo vyřešení zmizení preloaderu stránky. To bylo vyřešeno přidáním třídy nadefinované v CSS.

Posledním problémem, který jsem řešil v jQuery bylo přepisování hodnot u jednotlivých *options* ve výběrovém poli v kontaktním formuláři. Bylo nutné zde přepsat hodnoty, které se zobrazují uživateli. Toho jsem dosáhl tak, že jsem si vybral možnosti vždy podle ID *id_presmerovani* a zároveň jednotlivých ID *options* pomocí *option:eq(ID)*. To jsem ručně přepsal pomocí metody *text()*.

```
$(function() {  
    // Vykreslení navigace zafixované na vrch stránky,  
    když uživatel scrolluje dolů  
    $(window).scroll(function(){  
        $('nav').toggleClass('scrolled', $(this).scrollTop() > 0);  
    });  
  
    // Funkce řeší responzivní navigační lištičku na mobilních zařízeních  
    $("[data-trigger]").on("click",function(){  
        var trigger_id=$(this).attr('data-trigger');  
        $(trigger_id).toggleClass("show");  
        $('body').toggleClass("mobile-nav-active");  
    });  
  
    $('.animated-button').on('click',function(){  
        $('.animated-icon').toggleClass('open');  
    });  
  
    // Tlačítko pro vracení se zpět na vrchol stránky  
    if($(window).scrollTop() > ($(window).height() / 2)){  
        $('#scrollToTop').show();  
    }  
    else{  
        $('#scrollToTop').hide();  
    }  
  
    $(window).on('scroll',function(){  
        if($(window).scrollTop() > ($(window).height() / 2)){  
            $('#scrollToTop').fadeIn(100);  
        }  
        else{  
            $('#scrollToTop').fadeOut(100);  
        }  
    });  
  
    $("#scrollToTop").click(function(){  
        $("html, body").animate({  
            scrollTop: 0  
        }, 50);  
    });  
  
    // PreLoader  
    $('.preloader').addClass("preloader-disappeared");  
});
```

```
// Zobrazí ve výběru formuláře oddělení namísto e-mailů
$('#id_presmerovani option:eq(1)').text('Vedení');
$('#id_presmerovani option:eq(2)').text('Servis a generální opravy');
$('#id_presmerovani option:eq(3)').text('Elektro');
$('#id_presmerovani option:eq(4)').text('Nákup a zásobování');
$('#id_presmerovani option:eq(5)').text('Náhradní díly a kooperace (Crho  
nek)');
$('#id_presmerovani option:eq(6)').text('Náhradní díly a kooperace (Podl  
esný)');
});
```

Citace kódu 3.3: Soubor main.js

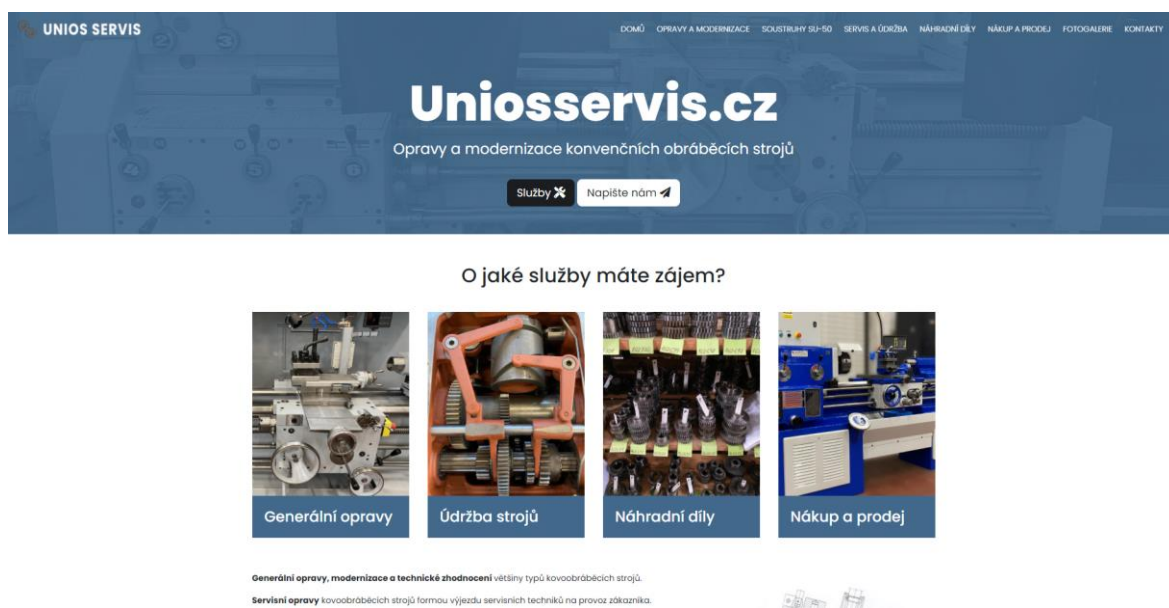
3.3.6 Finální úpravy

Na závěr jsem prováděl typografické úpravy text, velikosti písem a mezer na různých zařízeních. Následovala editace obsahu stránek, který bylo nutné doplnit o texty či obrázky. Dále jsem upravoval části kódu, který se mi zdály zbytečně složité. Kód co nejvíce zjednodušoval, případně přidával poznámky tam, kde byly žádoucí.

4 VÝSLEDKY ŘEŠENÍ

4.1 Ukázka webové stránky

Výsledkem řešení jsou webové stránky minimalistického vzhledu, které přesně odpovídají představám firmy. Obsah byl rozdělen do 8 jednotlivých stránek, ve kterých se lze snadno zorientovat. Na hlavní stránce se nachází přesměrování na stránky jednotlivých služeb. Dále podrobnější informace o poskytovaných službách a něco málo o samotné firmě. Na stránkách jednotlivých služeb jsou k nalezení všechny potřebné informace o firmou nabízených službách. Na stránkách se také nachází samostatná stránka s fotogalerií. V neposlední řadě na stránce s kontakty je k nalezení plně funkční formulář pro kontaktování firmy prostřednictvím přesměrování zprávy na e-mailové adresy dle výběru.



Obrázek 4.1: Hlavní stránka



Na provozě firmy v Lípniku nad Bečvou provádíme **střední a generální opravy strojů** (včetně **nového laku, ovládání, elektroinstalace, revizní zprávy elektro, protokolu přesnosti a prohlášení o shodě**), **modernizace** obráběcích strojů, montáže **digitálního odměřování** polohy, návrhy a stavby **elektrorozvaděčů** dle aktuálních norem.



K opravám v rozsahu GO dodáváme veškerou potřebnou dokumentaci – od nových **elektroschémát** pro nový menší **elektrorozvaděč** umístěný na zadní části rychlostní skříně, po protokol přesnosti dle výrobce, revizní zprávu elektro a prohlášení o shodě.

Rozsah generální opravy (GO) - hrubý popis prací

Mechanická část:

- demontáž stroje na podskupiny, očištění, příprava,
- přebroušení a zaškrabání všech vodících ploch – lože, suporty, vřeteník, koník,
- odstranění starého laku, tmelení a nový lak odstín Ral,
- oprava vodících klínů suportů, přebroušení a zaškrabání přítlačných listů, oprava mazacích drážek,

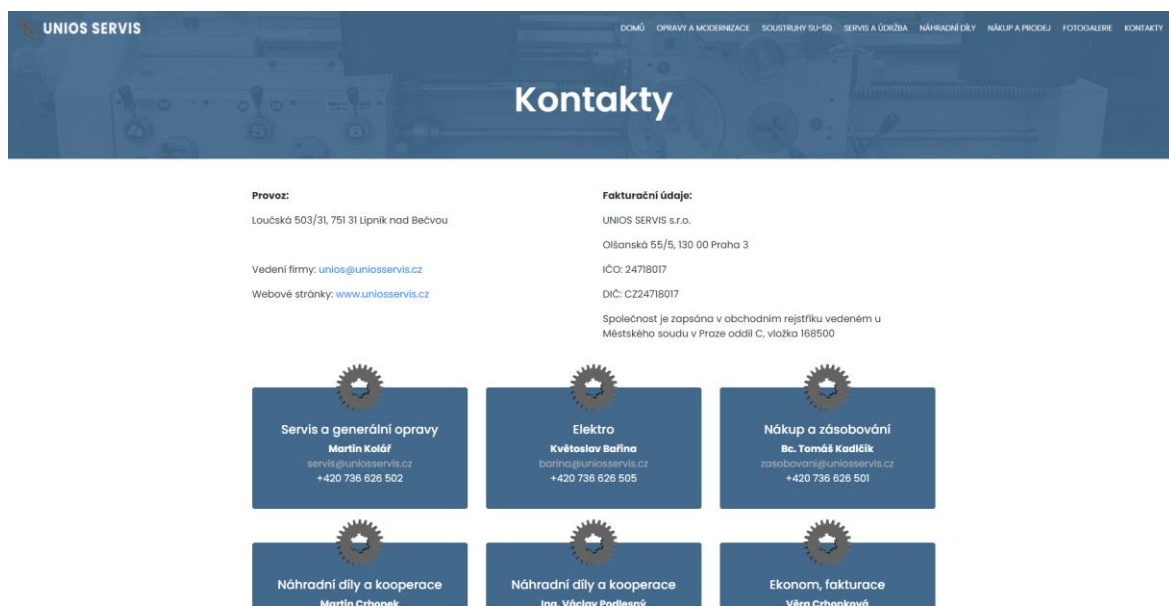
Elektro:

- zpracování projektu **včetně funkce brždění**,
- výroba a instalace nového el. rozvaděče včetně kabeláže a propojení,
- osazení stroje rozvaděčem a novými ovládacími prvky,
- bezpečnostní snímač krytu, osazení stroje bezpečnostními tlačítky Total Stop s funkcí nouzového zastavení,

Obrázek 4.2: Stránky služeb



Obrázek 4.3: Fotogalerie



Obrázek 4.4: Kontakty

UNIOS SERVIS DOMŮ OPRAVY A MODERNIZACE SOUSTAVY SJ-50 SERVIS A ÚDRŽBA NÁHRADNÍ DÍLY NÁKUP A PRODEJ FOTOGALERIE KONTAKTY

Pokud by došlo mezi námi a spotřebitelem ke vzniku spotřebitelského sporu z kupní smlouvy nebo ze smlouvy o poskytování služeb, který se nepodaří vyřešit vzájemnou dohodou, může spotřebitel podat návrh na mimosoudní řešení takového sporu určenému subjektu mimosoudního řešení spotřebitelských sporů, kterým je Česká obchodní inspekce Ústřední inspektorát – oddělení ADR, 120 00 Praha 2, Štěpánská 15. Email: adr@ci.cz Web: adr.ci.cz

Kontaktní formulář

Jméno*

Telefon*

E-mail*

Přesměrování*

Zpráva*

Obrázek 4.5: Kontaktní formulář

4.2 Zprovoznění stránek

Uvedení stránek do reálného provozu je v případě redakčního systému django CMS problematickou částí. Do budoucna však firma počítá s odkoupením webových stránek. To je ovlivněno tím, že firma má podepsanou smlouvu o poskytování webových stránek na dobu určitou. Otázkou však zůstává, jaký způsob bude nejvíce vyhovující? Možností existuje více:

- PaaS (Platform as a Service),
 - s využitím Dockeru,
 - Bez Dockeru,
- VPS (Virtuální Privátní Server).

V budoucnu bych chtěl zprovoznit webové stránky na fyzickém serveru firmy. Server je dost výkonný na to, aby dokázal hostovat tyto webové stránky. Hlavní důvodem, proč bych se chtěl vydat touto cestou je, že server není ani zdaleka plně využitý. Tato možnost je nejspíše nejtěžší na realizaci. Budoucnost stránek má však v rukou samotná firma.

ZÁVĚR

Cílem projektu bylo vytvořit funkční webové stránky pro firmu UNIOS SERVIS s.r.o. Stránky měly vycházet ze stránek starších, ale zároveň měly odstranit problémy, které se na nich vyskytovaly.

Webové stránky splňují všechny požadavky, které byly na začátku zadány. Konkrétně bylo na stránkách využito firemních barev, zjednodušil a sjednotil jsem vzhled stránek, stránky jsou také plně responzivní. Zrychlil jsem načítání stránek a vyvaroval jsem se chyb z předešlých webových stránek. Byly vytvořeny jednoduché animace s logem firmy. V neposlední řadě jsem vytvořil kontaktní formulář, který firmě zjednoduší práci s přesměrováním zpráv na různé e-mailové adresy.

Do budoucna se počítá s uvedením stránek do reálného provozu. Nejlepším řešením by bylo hostování webu na hardwaru firmy, ale existují i jiná řešení, jak by mohly být webové stránky zprovozněny pro širokou veřejnost.

Zdrojový kód je k dispozici na mém Githubu: <https://github.com/lukyncze/final-project>

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- [1] Django CMS. *Enterprise Content Management with Django - django CMS* [online]. [cit. 2021-01-04]. Dostupné z: <https://www.django-cms.org/en/>
- [2] Django. *The Web framework for perfectionists with deadlines* [online]. [cit. 2021-01-04]. Dostupné z: <https://www.djangoproject.com/>
- [3] Stack Overflow. *Where Developers Learn, Share, & Build Careers* [online]. [cit. 2021-01-04]. Dostupné z: <https://stackoverflow.com/>
- [4] Bootstrap. *The most popular HTML, CSS, and JS library in the world.* [online]. [cit. 2021-01-04]. Dostupné z: <https://getbootstrap.com/>
- [5] W3Schools. *W3Schools Online Web Tutorials* [online]. [cit. 2021-01-04]. Dostupné z: <https://www.w3schools.com/>
- [6] YouTube. *Getting Started with django CMS* [online]. 2016 [cit. 2021-01-04]. Dostupné z: <https://www.youtube.com/watch?v=NbsRVfLCE1U>