

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practice in the
Company

Zadání bakalářské práce

Student:

Vojtěch Holaň

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Poski.com s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Dr. Ing. Eduard Sojka**

Konzultant bakalářské práce: Ing. Tomáš Frydrych

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

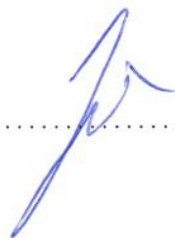




prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 1. dubna 2018



.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 2.4.2018

 **Poski.com s.r.o.**
Chytré systémy na míru & webdesign
28. října 1684/291, 709 00 Ostrava
tel.: 597 317 067 tel.: 733 534 704
IČ: 27791505 DIČ: CZ27791505
www.poski.com info@poski.com

(1)



Poděkování

Chtěl bych poděkovat firmě Poski.com s.r.o. za možnost absolvovat odbornou praxi. Dále bych chtěl poděkovat vedoucímu Ing. Tomáši Frydrychovi za vedení naší odborné praxe a Ing. Jakubovi Mackovi Ph.D za odbornou pomoc při jejím vypracování.

Abstrakt

Bakalářská práce popisuje absolvování odborné praxe ve společnosti Poski.com s.r.o. na pozici front-end developer. Praxe byla zaměřena na vývoj prvku pro CMS systém firmy. V této bakalářské práci jsou uvedeny základní informace o firmě, popis použitých prostředků, řešení a problematika zadaných úkolů, které byly řešeny v rámci odborné praxe. Dále jsou popsány získané dovednosti a zkušenosti nabyté při odborné praxi.

Klíčová slova: Poski.com s.r.o., Odborná praxe , Konfigurator, Žaluzie, JavaScript

Abstract

The Bachelor's thesis describes a professional practice at the Poski.com s.r.o. on position front-end developer. Practise was mainly focused on development items for CMS system of the company. Document said basic information about company, describe used technology, solve and problematics in task which was solved on my professional practice and experience and skills which I reached.

Key Words: Poski.com s.r.o., Professional practice, Configurator, Venetian blind, JavaScript

Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	9
Seznam výpisů zdrojového kódu	10
1 Úvod	11
2 Představení firmy	12
2.1 O firmě	12
2.2 Popis pozice a zaměření	12
3 Použité technologie	13
3.1 HTML a CSS	13
3.2 JS	13
3.3 SASS	13
3.4 jQuery	14
3.5 Vue.JS	14
3.6 Git	15
3.7 Visual Studio Code	16
3.8 FontAwesome	16
3.9 WYSIWYG Editor	16
3.10 SVG	16
4 Zadané úkoly	17
4.1 Nasazení webových stránek www.dekubity.cz	17
4.2 Nasazení webových stránek www.rovelcz.cz	19
4.3 Vytvoření pluginu pro WYSIWYG editor	20
4.4 Implementace komfortního konfiguratoru žaluzií	22
5 Závěr	41
5.1 Znalosti získané v průběhu studia uplatněné v průběhu odborné praxe	41
5.2 Znalosti scházející v průběhu odborné praxe	41
5.3 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení	41
Literatura	42
Přílohy	43

Seznam použitých zkratek a symbolů

HTML	–	Hyper Text Markup Language
CSS	–	Cascading Style Sheets
JS	–	JavaScript
SASS	–	Syntactically Awesome Style Sheets
UI	–	User Interface
CMS	–	Content Management System
SEO	–	Search Engine Optimization
AJAX	–	Asynchronous JavaScript and XML
DOM	–	Document Object Model
SVG	–	Scalable Vector Graphics
WYSIWYG	–	What You See Is What You Get
XML	–	eXtensible Markup Language

Seznam obrázků

1	Ukázka webu dekubity.cz	17
2	Ukázka řešení šestiúhelníku	18
3	Ukázka webu rovelcz.cz	19
4	Ukázka indikátoru pohybu na webu rovelcz.cz	20
5	Ukázka pluginu pro WYSIWYG	21
6	Ukázka vstupu čísel 1. [12]	23
7	Ukázka vstupu čísel 2. [13]	23
8	Ukázka vstupu čísel 3. [14]	24
9	Ukázka vstupu čísel 4. [15]	25
10	Ukázka vstupu čísel 5. [16]	25
11	Ukázka vstupu předdefinovaných hodnot 1.	26
12	Ukázka vstupu předdefinovaných hodnot 2.	26
13	Ukázka vstupu předdefinovaných hodnot 3. [14] [15]	27
14	Ukázka vstupu předdefinovaných hodnot 4.	28
15	Ukázka vstupu předdefinovaných hodnot 5. [14]	28
16	Ukázka zobrazení konfigurátoru 1.[19] [18]	29
17	Ukázka zobrazení konfigurátoru 2. [20][14]	30
18	Ukázka rozložení prvků konfigurátoru 1. [14]	31
19	Ukázka rozložení prvků konfigurátoru 2. [20]	31
20	Ukázka řešení vstupu čísel 1.	33
21	Ukázka řešení vstupu čísel 2.	33
22	Ukázka řešení vstupu čísel 3.	34
23	Ukázka řešení výběru z předdefinovaných hodnot 1.	34
24	Ukázka řešení výběru z předdefinovaných hodnot 2.	35
25	Ukázka řešení výběru z předdefinovaných hodnot 3.	35
26	Ukázka řešení vkládání textu 1.	35
27	Ukázka rozložení konfigurátoru 1.	36
28	Ukázka rozložení konfigurátoru 2.	36
29	Ukázka rozložení konfigurátoru 3.	38
30	Ukázka rozložení konfigurátoru 4.	38
31	Ukázka struktury souboru konfigurátoru 1.	40

Seznam výpisů zdrojového kódu

1	Porovnání SCSS a CSS	13
2	Ukázka Vue.js komponenty	14
3	Ukázka HTML data atributu	21
4	Ukázka měřicího kódu	21
5	Ukázka JSONu pro vkládání čísel	33
6	Ukázka JSONu produktu	39
7	Ukázka JS kódu pro spuštění konfigurátoru	39

1 Úvod

Absolvování odborné praxe jsem si zvolil z důvodů nabytí zkušeností a získání praxe ve firmě. Zároveň jsem si mohl vyzkoušet práci v kolektivu, jak to ve firmě vyvíjející CMS systémy funguje včetně možnosti práce na reálných projektech. Odbornou praxi jsem vykonával ve firmě Poski.com s.r.o. na pozici front-end developera. Tuto firmu jsem si vybral, jelikož vyvíjí webové CMS systémy a v budoucnu bych chtěl pracovat v tomto odvětví. Ať už na pozici front-end developera nebo programátora.

V druhé kapitole představuji firmu Poski.com s.r.o., díky které jsem mohl vykonat odbornou praxi a také pracovní pozici, na které jsem v této firmě pracoval.

Třetí kapitola obsahuje informace o užívaných technologiích v rámci odborné praxe, které jsem použil nebo se s nimi seznámil.

Ve čtvrté kapitole popisuji zadání úkolů na odborné praxi. Problémy, které byly řešeny v průběhu každého z nich, a nakonec postup vypracování úkolů.

Závěrečná kapitola popisuje zkušenosti před odbornou praxi a po jejím absolvování, chybějící znalosti a dosažené výsledky.

2 Představení firmy

2.1 O firmě

Poski.com s.r.o. je firma sídlící v Ostravě, která je na trhu již 20 let a zabývá se vývojem CMS systémů pro webové stránky s různým zaměřením. Firma je rozdělena do třech týmů. Tým PoskiQ se zabývá vývojem CMS pro klasické webové stránky a e-shopy. Další tým PoskiREAL vyvíjí systém pro realitní kanceláře, který pomáhá při práci realitním makléřům. Ve firmě působí i tým marketing / obchodní zastoupení, který se zabývá měřicími kódy na webových stránkách, obsahem webových stránek, SEO a analýzami webu včetně reklam.[1]

2.2 Popis pozice a zaměření

Do firmy jsem nastoupil na pozici front-end developer. Pracovní náplní na této zmíněné pozici je údržba a vývoj CMS systému firmy. Jde převážně o interaktivitu, vzhled a vizuální efekty. Dále nasazení nových webů na tyto systémy dle grafických podkladů. Používané jazyky na této pozici jsou HTML, CSS, JS a okrajově i PHP. Měl jsem šanci si vyzkoušet působení v obou týmech, jak PoskiQ ,tak PoskiREAL, seznámit se a osvojit si praxi s různými systémy.

3 Použité technologie

3.1 HTML a CSS

HyperText Markup Language je název pro značkovací jazyk používaný pro tvorbu webových stránek, který popisuje strukturu webu a rozděluje ho do logických sekcí. Tento jazyk je charakterizován za pomoci značek a jejich vlastností, které jsou strukturovány do DOM. DOM neboli Document Object Model uspořádává značky do stromové struktury. [2] Cascading Style Sheets je jazyk díky kterému můžeme HTML prvkům určovat vzhled. Od barev prvků přes fonty až po nejrůznější odsazení.[3] Dále můžeme řešit vzhled pro nejrůznější velikosti zařízení tak zvaný responsivní design. Díky těmto jazykům můžeme tvořit statické stránky.

3.2 JS

JavaScript je objektově orientovaný skriptovací jazyk používaný pro webové stránky, který pomáhá dynamicky ovlivňovat chování prvků na webu.[4] Můžeme měnit vzhled, vytvářet animace, přidávat třídy (nejedná se o třídy ve smyslu OOP, ale o přidání vlastnosti dané značce) nebo stáhnout data bez nutnosti obnovení stránky. Tento jazyk funguje na straně klienta.

3.3 SASS

Syntactically Awesome Style Sheets je CSS preprocesor, který dovoluje psát CSS mnohem jednodušeji. Kratším zápisem, dovolí vytvářet proměnné, funkce, mixin nebo automaticky přidávat prefixy pro různé prohlížeče.[5] Po uložení SASS souboru se kód překompiluje do CSS souboru, který je poté načítán prohlížečem. Dále pomáhá komprimovat CSS soubory, což zaručí rychlejší načítání webovým prohlížečem.

```
// SASS
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }
  li {
    display: inline-block;
  }
  a {
    display: none;
    padding: 5px 10px;
    text-decoration: none;
  }
}
```

```

    }
}

// CSS
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}
nav li {
  display: inline-block;
}
nav a {
  display: none;
  padding: 5px 10px;
  text-decoration: none;
}

```

Výpis 1: Porovnání SCSS a CSS

3.4 jQuery

Rychlá, malá a funkcemi bohatější JS knihovna.[6] Umožňuje lepší procházení v DOMu webové stránky, zachytávání událostí, animace, lehčí syntaxe AJAX požadavku.

3.5 Vue.JS

JS framework, který slouží pro vytváření uživatelských rozhraní, ale také single-page application. Velikou výhodou Vue.js je vytváření dynamických komponent, díky kterým tvoříme strukturu stránky jednodušším způsobem. Navíc si dokáže každá komponenta udržovat vlastní stav a metody. Lze vytvářet šablony komponenty nebo vytvářet podmínky přímo v HTML kódu. Tím se kód stává mnohem přehlednější a jednodušší. Existují i různá rozšíření do prohlížečů[7], kde lze dynamicky kontrolovat stav jednotlivých komponent a jejich proměnných.[8]

```

//Šablona komponenty
<template>
  <div class="choose-check" :class="checked === true ? 'checked' : ''">
    <div class="img">
      
    </div>
  </div>

```

```

    <p class="description" v-if="value.description && value.description != ''
      ">
      <i class="fa fa-info-circle" aria-hidden="true"></i>
      <span class="text">{{ value.description }}</span>
    </p>
    <input type="checkbox" @click="changeChecked()">
    <label :for="'typeNumber' + value.gId">{{value.name}}</label>
  </div>
</template>

<script>
  export default {
    props: ['part', 'value'],
    //Promenne komponenty
    data() {
      return {
        checked: false,
      }
    },
    //Metody komponenty
    methods: {
      changeChecked() {
        this.checked = !this.checked;
        this.$emit('changeCheckedCheck', this.checked, this.value.name);
        Bus.$emit('change-img', this.value.imageSample, this.value.name);
      }
    }
  }
</script>

```

Výpis 2: Ukázka Vue.js komponenty

3.6 Git

Distribuovaný systém správy verzí, kterým je možné si zálohovat a verzovat soubory na externím uložišti. Tento systém dovoluje organizovat práci více lidí na jednom projektu.

3.7 Visual Studio Code

Open source editor, který zjednodušuje práci s kódem. Podobný například jako Atom nebo Sublime Text 2. Pro tento editor je dostupných mnoho balíčků, které pomáhají vývojáři zjednodušit práci nebo zpřehlednit kód. Moje nejpoužívanější balíčky jsou HTML Snippet pro rychlejší psaní HTML kódu, IntelliSense for CSS class names in HTML, který napovídá třídy z CSS v HTML kódu, Material Icon Theme poskytuje dobře vypadající ikony pro různé typy souborů a Vue 2 Snippets for Visual Studio Code pro rychlejší psaní Vue.js kódu.

3.8 FontAwesome

Ikony ve formě fontu. Na webových stránkách načítají rychleji než jednotlivé soubory. V nových verzích poskytuje i SVG ikony. Existuje jak v placené, tak volně dostupné verzi.

3.9 WYSIWYG Editor

Textový procesor, který se nejčastěji používá k ovlivňování obsahu webových stránek. Kdy neznalí uživatelé vidí editor, který je vzhledově a funkcionalitou velice podobný jako Microsoft Word nebo OpenOffice. Uživatel poté může jednoduše změnit obsah bez znalosti HTML nebo CSS.

3.10 SVG

Formát vektorových obrázků. Stále více se prosazující formát, který není závislý na rozlišení. V praxi to znamená, že pokud budeme obrázek zvětšovat tak neztratí na kvalitě. Je dynamicky upravitelný pomocí CSS a JS. Grafika je definována pomocí jazyka XML.[9]

4 Zadané úkoly

4.1 Nasazení webových stránek www.dekubity.cz

4.1.1 Zadání

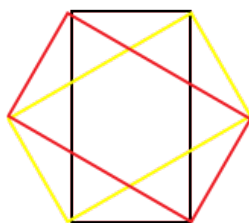
Prvním úkolem, který byl zadán, bylo nasazení grafiky na systém Clevero. Nasazení grafiky je proces, kdy na už vytvořený systém potřebujeme přenést vzhled dle grafického návrhu a finální funkcionalitu dle potřeb zákazníka. První projekt se jmenoval Dekubity, stránka nabízející zdravotnické potřeby a sociální služby. Web se ve firmě řadil do skupiny e-shop simple. Jedná se o zjednodušenou formu elektronického obchodu, která zahrnuje katalog produktů, ale produkty z katalogu lze pouze popsat, nikoliv koupit. Než byla webová stránka předána klientovi bylo nutné provést kompletní revizi webu. Taková kontrola obsahuje hned několik bodů jako například validace HTML nebo kontrola rychlosti načítání stránky.



Obrázek 1: Ukázka webu dekubity.cz

4.1.2 Problémy při vypracování

Při tomto úkolu jsem narazil na několik problémů. První problém byl nastylovat šestiúhelníkové tvary, které jsou hned v úvodu webu. Řešení bylo více, ale nakonec jsem se rozhodl mezi dvěma. První řešení bylo použít SVG obrázky. Druhé bylo použít čistě HTML a CSS. Zvolil jsem druhou variantu, protože v ní lze lépe vyřešit zalamování textu. Klient si vytváří kategorie produktů sám, mohlo by se tedy stát, že bude text přetékat přes šestiúhelník. Řešení pomocí CSS funguje tak, že byly vytvořeny 3 obdélníky, které jsou natočeny pod určitým úhlem, aby tvořil šestiúhelník.



Obrázek 2: Ukázka řešení šestiúhelníku

Plnění webu obsahem ze starých stránek bylo potřeba konzultovat s klientem, a to z důvodu požadavku na úpravu. Některé stránky z původního webu zanikly a bylo vytvořeno několik nových. Při úpravě textu z tohoto původního webu bylo nutné dbát na jeho přehlednost. Textu bylo mnoho a neměli vypovídající obsah, což způsobuje dezorientaci návštěvníka. Každý web by měl návštěvníka upoutat a zároveň mu poskytnout informaci, kterou na webu hledal. Po analýze marketingového oddělení bylo zjištěno, že pokud má stránka velké množství textu stísněného k sobě, neupoutáme návštěvníkovu pozornost a v takovém případě stránku opouští.

4.1.3 Vypracování

Nasazení prvního projektu bylo složité, poněvadž má znalost systému byla téměř nulová. Na startu úkolu byl takzvaný kick-off k projektu, kde se probraly žádosti a požadavky klienta a problémy při výrobě webu. Poté jsem obdržel PSD soubor s grafikou daného webu, který jsem si prohlédl a řešil problematiku nasazení grafiky a jeho řešení včetně responsivního designu. Ten bývá často stěžením, ale pokud si kodér dobře připraví web v základním rozlišení poté responsivní design je mnohem jednodušší vypracovat. Z PSD souboru jsem si vyřízl obrázky a web začal krok po kroku nasazovat. Prvním krokem bylo vypracování hlavní stránky, protože ta tvoří základní pilíř pro ostatní podstránky. Po nasazení jsem pokračoval s responsivním designem a podstránkami.

Dalším krokem bylo naplnění stránky daty ze starého webu a jejich úprava. Následovala výstupní kontrola, která obsahuje mnoho bodů.

První bod byla kontrola validace HTML kódu, kde je velice jednoduché vytvořit chybu, pokud je HTML generováno dynamicky za pomoci PHP.

V druhém bodu bylo potřeba nechat stránku projít nástrojem od Googlu s názvem PageSpeed Insights. Tento nástroj analyzuje načítání webu a upozorní na problémy jako jsou například příliš velké obrázky, pomalá odezva od serveru nebo zda máme komprimovány CSS, HTML a JS a další problémy s načítáním.

V třetím bodu jsem použil další nástroj s názvem Seo Servis, který kontroluje META tagy, robot.txt a sitemap.xml, jenž slouží také pro vyhledávače. Poté reviduje, zda má stránka optimální počet znaků a strukturu nadpisu apod.

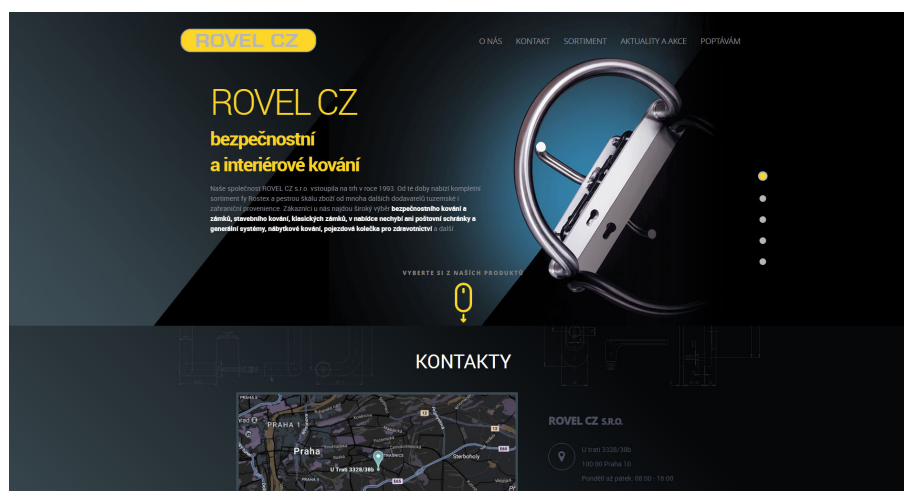
Čtvrtý použitý nástroj byl WebPageTest, který zjišťuje jak dlouho web načítá při prvním spuštění a jeho opakovaných navštíveních. Jako poslední bod bylo nutno testovat web v různých prohlížečích a na různých zařízeních. Tato kontrola se provádí, jelikož má každý prohlížeč jiné vykreslovací jádro a proto stránku může vykreslit rozpadlou. V případě, že nalezneme špatně zobrazenou stránku musíme optimalizovat náš HTML a CSS kód. Kontrolu bylo nutno provést v prohlížečích Google Chrome, Firefox, Internet Explorer a Safari. A na tabletech se systémem Android a iOS v různých prohlížečích. Poté byla stránka předána klientovi, který zaslal připomínky s finálními úpravami.

Čas realizace: 7 dní

4.2 Nasazení webových stránek www.rovelcz.cz

4.2.1 Zadání

Druhým úkolem bylo nasadit takzvaný one-page web propagující bezpečnostní a interiérové kování. Tento web je založen na sekcích, kde každá sekce popisuje určitý produkt a tvoří kostru webu. Klient měl speciální požadavky. Chtěl, aby se stránka animovala a obsahovala indikátor pohybu na stránce. Tento indikátor ukazuje, v jaké sekci se právě ve stránce nacházíme.



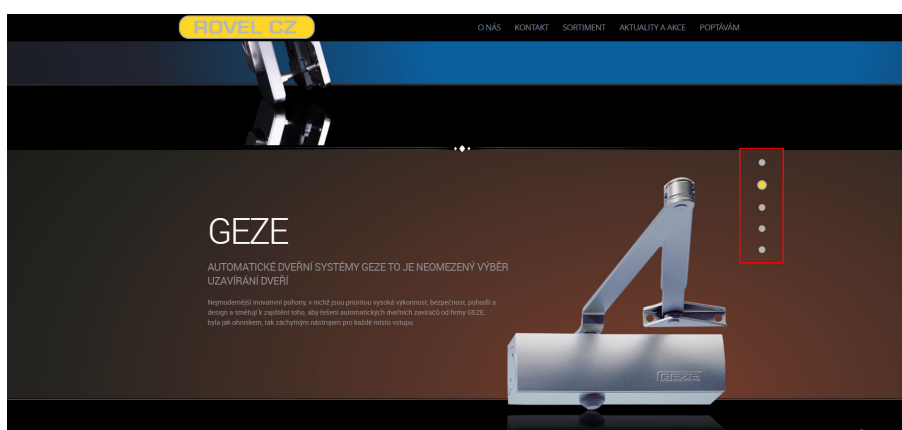
Obrázek 3: Ukázka webu rovelcz.cz

4.2.2 Problémy při vypracování

Krucialním problémem byl požadavek klienta na animace po načtení webu. Musel jsem proto vytvořit JS kód, který během posouvání na stránce provede animaci té sekce, na kterou se právě nacházíme. Základním kamenem je vždy obalovací HTML element s danou třídou. Potomkům tohoto prvku je přidávána v časovém intervalu třída, která daný prvek animuje. Časový interval

je použit z důvodu postupného animování prvku. Pokud se animuje například seznam několika článků budou se animovat postupně.

Další problém byl indikátor pohybu na stránce. Zhotovil jsem ho za pomoci jQuery. Základem indikátoru jsou tečky, kde každá z nich představuje jednu sekci. Při posouvání na webu mění tečky indikátoru barvu vždy podle toho, na jakou sekci jsme zrovna najeli. Tečky indikátoru fungují jako odkazy, což znamená, že pokud se na jednu z nich klikneme, tak se stránka bude animovat na vybranou sekci. Pomocí indikátoru má zákazník přehled, kde se právě na stránce nachází. Počet teček v indikátoru se dynamicky mění podle HTML kódu. Pokud uživatel přidá novou sekci, tak se vytvoří nová tečka s odkazem na ní.



Obrázek 4: Ukázka indikátoru pohybu na webu rovelcz.cz

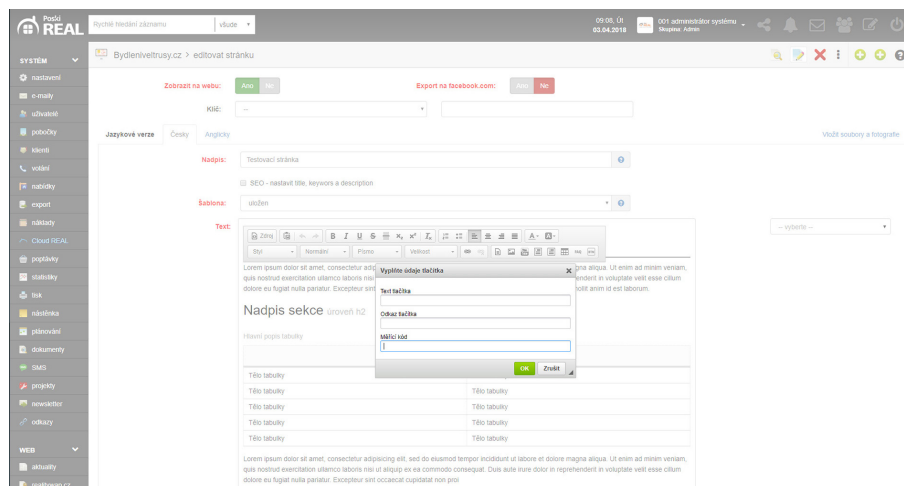
4.2.3 Vypracování

Provedení probíhalo podobně jako úkol 4.1. Jediný rozdíl byl ve specifických požadavcích klienta.

Čas realizace: 5 dní

4.3 Vytvoření pluginu pro WYSIWYG editor

K dalším úkolům patřilo vytvořit plugin do WYSIWYG editoru. Tento úkol se dělil do dvou částí. První část se skládala z rozepsaného pluginu pro vytvoření měniče obrázku na webu, který jsem měl dopracovat. Druhá část byla vytvoření nového pluginu, jenž měl vytvořit odkaz vypadající jako tlačítka. Při vkládání tlačítka do editoru se musí do dialogového okna vložit název tlačítka a odkaz, kam má směřovat. Podle požadavku marketingového oddělení bylo potřeba vložit měřicí kód, který bude pomocí analytických nástrojů měřit počet kliknutí na tento odkaz.



Obrázek 5: Ukázka pluginu pro WYSIWYG

4.3.1 Problémy při vypracování

Překážka se objevila hned v první části. Bylo potřeba ve scriptu vytvářejícího měnič obrázku přechíst atributy zadané do dialogového okna WYSIWYG editoru. Tento problém lze vyřešit pomocí data atributu v HTML. Při tomto řešení jsou data z editoru ukládány do data atributu HTML kódu.

```
<div class="slider" data-auto-move="true">
```

Výpis 3: Ukázka HTML data atributu

Poté ve scriptu už nebyl problém daná data při inicializování měniče obrázku vytáhnout z HTML.

Vkládání měřicího kódu po kliknutí na tlačítko bylo potřeba konzultovat s marketingovým oddělením. Marketingové oddělení poskytlo informaci, kde vložit kód pro správné měření. Vkládal se do atributu "onclick".

```
onclick="ga('send', 'event', 'Formular - HP (prodat)', 'Odeslani formulare', 'HP')"
```

Výpis 4: Ukázka měřicího kódu

4.3.2 Vypracování

Při plnění úkolu jsem postupoval následovně, a to studiem dokumentace[10] o tvoření pluginu do WYSIWYG editoru a teprve poté začala práce na první části. Jelikož lze kód napsat různými způsoby, snažil jsem se zorientovat v rozepsaném kódu abych jej mohl dopsat. Orientace v kódu umožnila pokračovat v jeho psaní, kdy bylo nutné jej napojit na plugin který se nazývá Owl Carousel[11], který vytvoří měnič obrázku na webu. Současně bylo také nutné přidat několik

nastavení, kterými může zákazník ovlivnit chování měniče obrázku, například zapnutí automatického posouvání, nebo po jakých časových intervalech se má posouvat. Rozpracovaný plugin momentálně nedisponuje ořezáváním obrázků. Tato funkcionalita je plánována v budoucnu. Prozatímním řešením bylo vycentrovat obrázky a ořezat je za pomoci CSS. Vzhled musel být upraven jak přímo na stránce, tak i ve WYSIWYG editoru, aby klient plnicí stránku věděl, o jaký prvek se jedná.

Následně začala práce na druhé části úkolu. Prvním krokem bylo přidání položky vložení tlačítka do menu. V dalším kroku jsem musel dohledat způsob vytváření a problematiku dialogového okna. Do dialogového okna zadáme potřebné hodnoty, díky nimž se vytvoří HTML strukturu. Posledním krokem bylo vyřešit způsob vložení měřicího kódu do HTML, aby mohl měřit počet kliknutí.

Čas realizace: 10 dní

4.4 Implementace komfortního konfigurátoru žaluzií

4.4.1 Zadání

Posledním úkolem, který časově zabral více jak polovinu odborné praxe, bylo vytvořit e-shopový konfigurátor pro žaluzie. Firma má několik zákazníků prodávajících žaluzie, proto už disponuje konfigurátorem pro žaluzie. Mým zadáním bylo vytvořit nový komfortnější konfigurátor, který bude vzhledově elegantnější s moderními efekty, bude se validovat po každém kroku a bude spustitelný v rámci libovolné stránky (tedy například z výpisu produktů, detailu produktů či vyhledávání produktů)

Jakou JS technologii použít určeno nebylo, a proto jsem si mohl vybrat takovou, která bude pro řešení nejefektivnější na použití.

První částí úkolu bylo zanalyzovat existující konfigurátory žaluzií vyrobené Poski.com i jinými firmami. Zjistit, z kolika kroků se průměrně konfigurátory žaluzií skládají a jaká data mohou obsahovat. Součástí tohoto úkolu byl průzkum konkurence a konfigurátoru jiných produktu než žaluzií, jejich komponenty a rozložení komponent konfigurátoru.

Druhá část zahrnovala výběr vhodného frameworku pro vypracování konfigurátoru.

Třetí část bylo samotné programování konfigurátoru včetně stylování.

4.4.2 Analýza použitých parametrů a vhodných prvků

U konfigurátorů žaluzií můžeme nalézt mnoho druhů nastavitelných parametrů. Základ ve většině případů tvoří výběr barev jednotlivých částí (lamela, lišta, krycí plech), základní rozměry, strana ovládání, příslušenství. Dále najdete individuální parametry u různých druhů žaluzií a rolet např. výsuv, motor, držák žaluzie, držák vodící lišty nebo zda chcete zahrnout montáž. Je nutno tedy pracovat s různými daty, kde si stanovíme základní komponenty, které musí být pro zákazníka příjemně použitelné, pochopitelné a hlavně přehledné.

4.4.2.1 Vstup pro vkládání čísel

1. HTML element typu input, do kterého se zadávají čísla. Může být použit na rozměry žaluzií, velikosti výsuvu nebo na jakýkoliv parametr obsahující číslo.



Obrázek 6: Ukázka vstupu čísel 1. [12]

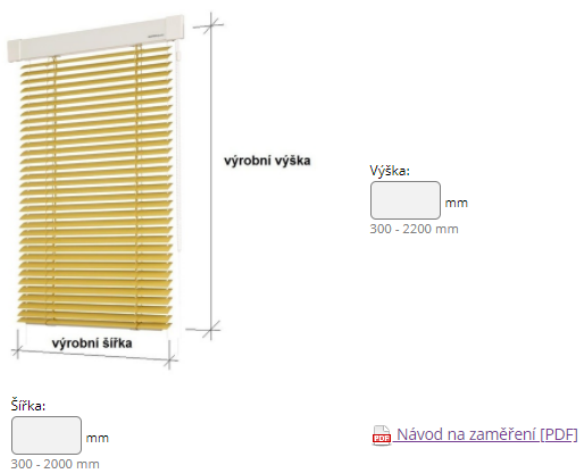
Výhody

- Zákazník si může určit vlastní velikost přesně na počet milimetrů

Nevýhody

- Nutná validace dat (nastavení minimální a maximální hodnoty)
- Minimální interaktivita se zákazníkem
- Nelze zvolit jinou jednotku než milimetry

2. HTML element typu input, do kterého se zadávají čísla. Rozdíl od předchozího je umístění inputu tak, aby bylo zákazníkovi zřetelnější o jaký rozměr se jedná.



Obrázek 7: Ukázka vstupu čísel 2. [13]

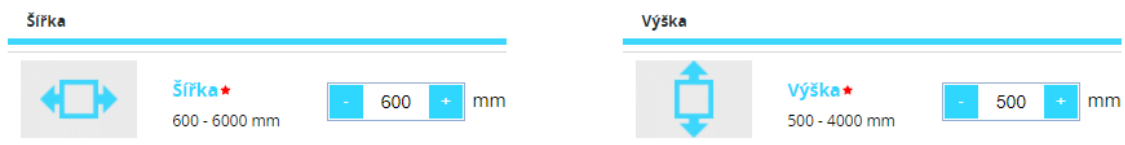
Výhody

- Zákazník si může určit vlastní velikost přesně na počet milimetrů
- Umístění inputu vzhledem k obrázku je pro zákazníka přehlednější

Nevýhody

- Nutná validace dat (nastavení minimální a maximální hodnoty)
- Minimální interaktivita se zákazníkem
- Nelze zvolit jinou jednotku než milimetry

3. HTML element typu input, do kterého se zadávají čísla. Přednosti u tohoto typu jsou dvě tlačítka pro inkrementaci a dekrementaci čísla, což může být příjemné pro zákazníka na telefonních zařízeních.



Obrázek 8: Ukázka vstupu čísel 3. [14]

Výhody

- Zákazník si může určit vlastní velikost přesně na počet milimetrů
- Přidání counteru (tlačítka pro změnu hodnoty), interaktivita
- Vzhledově elegantnější

Nevýhody

- Nutná validace dat (nastavení minimální a maximální hodnoty)
- Nelze zvolit jinou jednotku než milimetry

4. HTML element typu select ve kterém vybíráme čísla. Tato verze má dva selecty určující jednu velikost. První z nich obsahuje délku a druhý jednotku.

WINDOW DIMENSIONS
Enter your window dimensions to get started

Width (inches)

24" 0/8"

Height (inches)

36" 0/8"

Obrázek 9: Ukázka vstupu čísel 4. [15]

Výhody

- Předdefinované hodnoty s výběrem jednotky ulehčují zakazníkovi výběr
- Není nutná validace dat (nastavení minimální a maximální hodnoty)

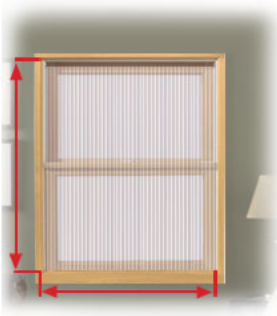
Nevýhody

- Zákazník si může zvolit data pouze z vybraných hodnot

5. HTML elementy typu select a input. Toto řešení dovoluje zadat rozměry ručně a poté si zvolit jednotku ve které jsou zadávány.

Measurements:

Width: 1/8" ▼



Height: 1/8" ▼

Invalid Width for entered Height

*Please enter measurements in inches

Obrázek 10: Ukázka vstupu čísel 5. [16]

Výhody

- Zákazník si může určit vlastní velikost s vybranou jednotkou
- Umístění inputu vzhledem k obrázku je pro zákazníka jasně přehlednější

Nevýhody

- Nutná validace dat (nastavení minimální a maximální hodnoty)
- Při validaci je nutné přepočítávání hodnot z důvodu výběru rozdílných jednotek

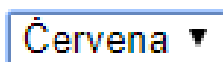
Problémy s komponentou pro vkládání čísel

V této fázi řešení mohou nastat další problémy, protože určité žaluzie se vyrábí v předem daných rozměrech. Prvek, v němž si zadáme vlastní velikost, je v těchto podmínkách nelogický. Naopak, pokud je žaluzie sestavena na míru s rozměry zadanými zákazníkem, byl by seznam definovaných hodnot v konfigurátoru redundantní.

4.4.2.2 Vstup pro výběr předdefinovaných dat s výběrem pouze jedné položky

HTML elementy typu input nebo select, díky kterým si zákazník vybere nastavení z předdefinovaných hodnot. Může být použit pro barvy, stranu ovládaní, druh montáže nebo rozměry s předdefinovanými hodnotami. Komponenta nedovoluje zobrazení obrázků, a to může být problém například u barvy nebo typu montáže, kde ilustrační obrázek pomáhá zákazníkovi s výběrem. Tato komponenta se na různých stránkách moc neliší.

1. HTML element typu select.



Obrázek 11: Ukázka vstupu předdefinovaných hodnot 1.

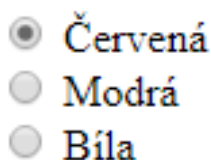
Výhody

- Není nutná validace dat (vždy má zákazník defaultně vybranou jednu barvu)

Nevýhody

- Pod červenou barvou si zákazník může představit jiný odstín

2. HTML element typu input.



Obrázek 12: Ukázka vstupu předdefinovaných hodnot 2.

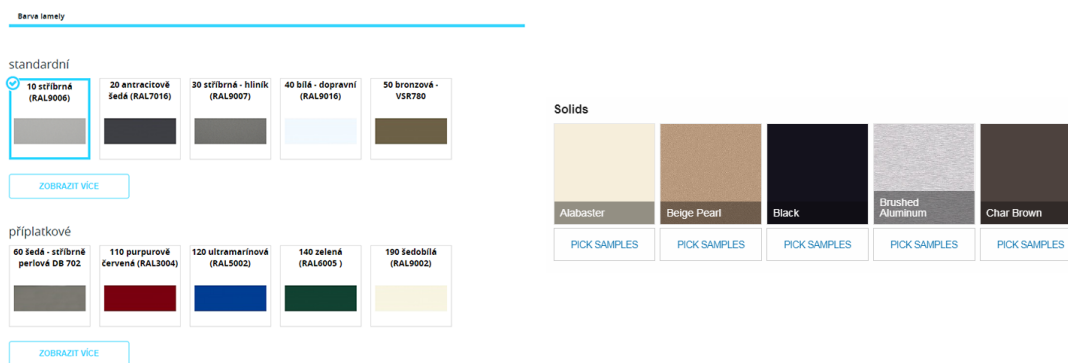
Výhody

- Není nutná validace dat (vždy má zákazník defaultně vybranou jednu barvu)

Nevýhody

- Pod červenou barvou si zákazník může představit jiný odstín

Tento typ umožní po přidání několika HTML elementů a CSS docílit mnohem lepší vizuální formy, u niž lze zobrazit i požadovanou barvu, která je pro zákazníka při výběru velice důležitá.



Obrázek 13: Ukázka vstupu předdefinovaných hodnot 3. [14] [15]

Na obrázku lze taky vidět jedna z možností, jak se vypořádat s velkým počtem barev. Zde máme tlačítko "Zobrazit více". Zobrazíme oblíbené barvy, další se objeví až po kliku na tlačítko.

Výhody

- Není nutná validace dat (vždy máme defaultně vybranou jednu barvu)
- Zákazník vidí ukázkou barvy na obrázku

Problémy s komponentou pro předdefinovaná data s výběrem pouze jedné položky

Při tomto výběru dat je nutné pouze ošetřit, aby vždy byla vybraná jedna hodnota. Toho dosáhneme, pokud vždy zvolíme jednu jako výchozí.

4.4.2.3 Vstup pro výběr předdefinovaných dat s výběrem více položek

HTML element typu input, díky kterého vybereme nějaké nastavení z předdefinovaných hodnot. U tohoto vstupu má zákazník možnost vybrat více než jednu hodnotu. Může být použit pro příslušenství, doplňkové služby nebo možnosti montáže. U této komponenty je opět účelné mít obrázek. Zákazník potom může vidět, jak příslušenství vypadá, o jaký typ nebo části montáže se jedná.

1. HTML element typu input.

- Kolébkový spínač
- Nina
- 5-kanálový dálkový ovladač

Obrázek 14: Ukázka vstupu předdefinovaných hodnot 4.

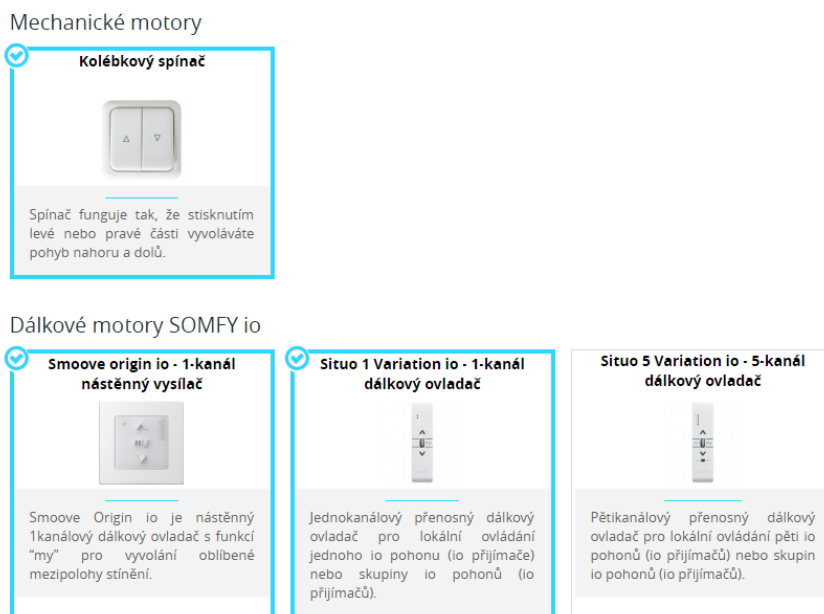
Výhody

- Není nutná validace dat (pokud není omezen počet vybraných hodnot)

Nevýhody

- Pro zákazníka data nejsou hmatatelná bez obrázku (nemusí vědět o co se jedná)
- Zákazníkovi sdělí málo informací

Tento typ umožní po přidání několika HTML elementů a CSS dosáhnout mnohem lepší vizuální formy, kdy můžeme zobrazit obrázek příslušenství s nějakou informační hláškou.



Obrázek 15: Ukázka vstupu předdefinovaných hodnot 5. [14]

Výhody

- Není nutná validace dat (pokud není omezen počet vybraných hodnot)
- Zákazník může vidět jak příslušenství vypadá
- Zákazník se v textu dozví základní informace

Problémy s komponentou pro předdefinovaná data s výběrem více položek

Při tomto výběru dat je validace nutná pouze, pokud chce prodejce omezit počet nakoupených položek. Pokud jako příklad použijeme nákup příslušenství, tak si zákazník může koupit pouze to co potřebuje, anebo si také nemusí koupit vůbec nic. Zde není určen počet příslušenství, které si musí koupit. Jako druhý příklad můžeme použít dárky k produktu, kde si k žaluzii můžeme vybrat například dva čisticí prostředky zdarma. U tohoto typu je nutné ošetřit, aby zákazník vybral právě dvě možnosti.

4.4.2.4 Vstup pro text

HTML element typu textarea do které lze zadávat text. Zde se validuje pouze zda musí být hodnota vyplněna. Komponenta, vhodná pro důležité poznámky k některým hodnotám, které upřesnění informací pro vlastníka e-shopu.

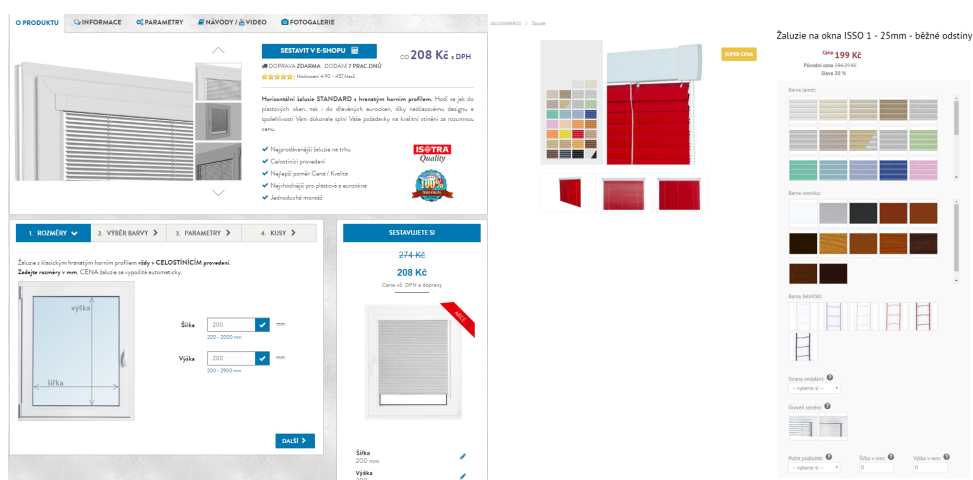
4.4.3 Analýza rozložení prvků konfigurátoru a jeho zobrazení

Webové prvky nabízejí různé možnosti jejich uspořádání. Prvky lze různě zobrazovat nebo skrývat. V dnešní době je důležité vnímat potřebu, že jakékoliv stránky by měly být přizpůsobitelné různým rozlišením. V říjnu roku 2016 dokonce telefony a tablety předčily počítače s počtem návštěv na webech. Telefony a tablety dosáhly procentuálně na 51.3 % návštěv. Z tohoto důvodu je velice důležité myslet na různá rozlišení.[17]

Při prohlížení různých konfigurátorů jsem našel mnoho možností na uspořádání. Srovnal jsem proto několik typů a vybral vhodná pro můj konfigurátor.

4.4.3.1 Analýza dle zobrazení konfigurátoru

1. Konfigurace na stejné podstránce jako produkt



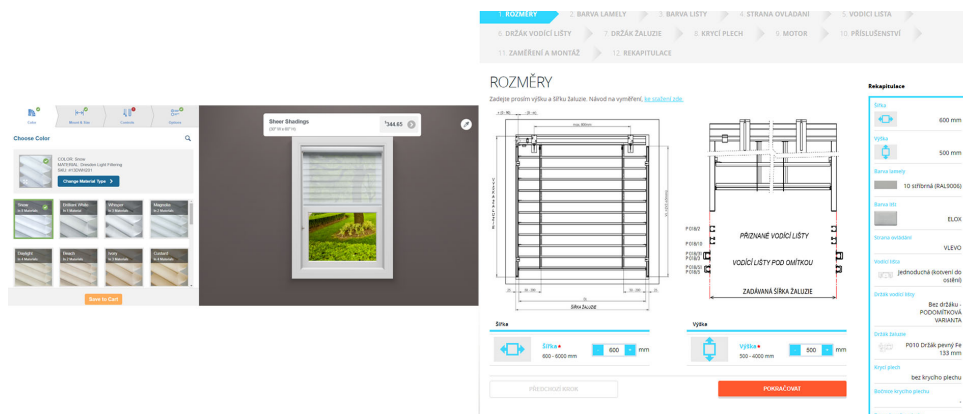
Obrázek 16: Ukázka zobrazení konfigurátoru 1.[19] [18]

Výhody

- Zákazník ihned nakonfiguruje produkt a nemusí být přesměrován na další podstránku

Nevýhody

- Na telefonech a tabletech bude stránka příliš dlouhá
 - Nepřehlednost
2. Konfigurace na nové podstránce. Toto řešení poskytuje daleko více prostoru pro konfigurátor. Díky menšímu obsahu dat bude konfigurátor na malých zařízeních mnohem přehlednější.



Obrázek 17: Ukázka zobrazení konfigurátoru 2. [20][14]

Výhody

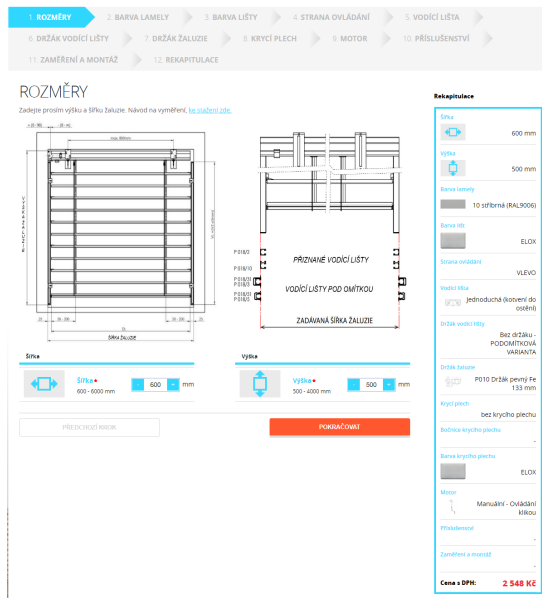
- Přehlednost
- Možnost lepšího řešení pro menší zařízení

Nevýhody

- Nutnost přesměrování na novou stránku z konfigurátorem

4.4.3.2 Analýza rozložení prvků konfigurátoru

1. Uspořádání pod sebou se souhrnem na pravé straně



Obrázek 18: Ukázka rozložení prvků konfigurátoru 1. [14]

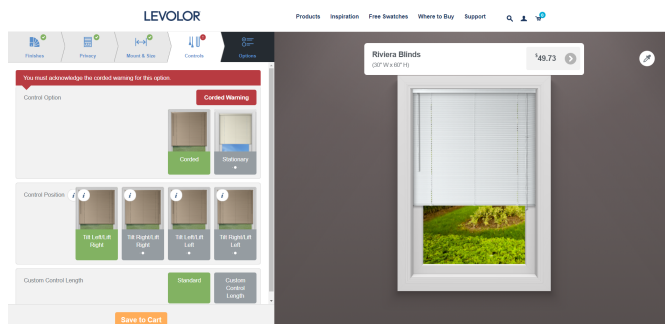
Výhody

- Přehlednost
- Přizpůsobitelnost pro menší zařízení

Nevýhody

- Nutnost přeměrování na konfigurátor

2. Uspořádání vedle sebe bez shrnutí objednávky



Obrázek 19: Ukázka rozložení prvků konfigurátoru 2. [20]

Výhody

- Přehlednost

Nevýhody

- Nutnost přesměrování na konfigurátor
- Vhodné pro menší počet kroků
- Není prostor pro souhrn kroků

4.4.4 Analýza technologií

4.4.4.1 Adobe Flash

Adobe Flash je v dnešní době umírající technologie, a to hned z několika důvodů. Nekompatibilita se zařízeními od Applu a na některých telefonních zařízeních. Časově náročná údržba. Snižuje se jeho podpora, dokonce má většina prohlížečů ve výchozím stavu Adobe Flash zakázaný. Vysoká náročnost na systémové prostředky a bezpečnostní chyby jsou jedny z dalších jeho problémů. Dnešní době však máme jiné a lepší možnosti, jak jej nahradit. Jako například HTML5, CSS3, Canvas nebo SVG.[21][22][23]

4.4.4.2 Vue.js

Nabízí se použití více JS frameworku, jako jsou například Angular nebo React. Nakonec jsem zvolil Vue.js. Důvodů bylo hned několik. Vue.js je v porovnání s výše uvedenými frameworky velikostně nejmenší. Jeho popularita rychle roste, je vhodný i pro menší aplikace a má kvalitní dokumentaci.[24] Malá velikost tohoto frameworku je v tomto případě vhodná, protože framework používám pouze na konfigurátor a ne na celý web. Růst popularity Vue.js zaručuje jeho dlouhou životnost a dobře napsaná dokumentace ulehčuje práci.

4.4.4.3 Pluginy pro vzhled komponent

Existuje hned několik JS pluginů, které se zabývají různými UI prvky, ale pro mé řešení v konfigurátoru jsou nevhodné. Nevýhodou je nutnost importovat celou knihovnu, což je zbytečně obsahově a datově náročné. V tomto případě jsem preferoval vytvoření jednotlivých prvků samostatným řešením.

4.4.5 Komponenty konfigurátoru

4.4.5.1 Vstup pro vkládání čísel

V rámci řešení zadaného úkolu jsem se rozhodl vytvořit dvě komponenty pro vkládání čísel.

První komponenta, v níž zadáváme číslo ručně je použitelná pro žaluzie na míru. Pro větší interaktivitu byla využita tlačítka na změnu hodnoty. Výběr jednotky nebyl použit do řešení, protože je to zbytečně složité pro zákazníky. Validace této komponenty probíhá po napsání

každého znaku a je omezena minimem a maximem. Lze psát pouze čísla a tečku jako desetinnou čárku. Jiná hodnota nelze napsat.



Obrázek 20: Ukázka řešení vstupu čísel 1.



Obrázek 21: Ukázka řešení vstupu čísel 2.

Z obrázku 21 je patrné, že pokud zákazník zadá špatné hodnoty, tak je chyba zřetelně viditelná a pochopí, co musí opravit. Zároveň, pokud nastane chyba a zákazník se snaží jít do dalšího kroku, tak vyskočí takzvané pop-up okno s upozorněním, že tento krok musí být upraven a poté může pokračovat.

```
{
  "name": "Vyska",
  "valueId": 1,
  "gId": 3,
  "default": 200,
  "step": 10,
  "min": 200,
  "max": 2000,
  "rate": "mm",
  "image": "/images/configurator/steps/arrows-alt-h.svg",
  "description": "Zadejte sirku zaluzie v milimetrech",
  "required": true,
  "errorFlash": "Vyplnte te vysku (200mm - 2000 mm)"
}
```

Výpis 5: Ukázka JSONu pro vkládání čísel

Z výpisu 5 je vidět nastavení výchozí hodnoty, minima a maxima, krok o jaký se má číslo zmenšit nebo zvětšit po klikání na tlačítka, případně zda je nutné daný input přímo vyplnit. Pokud není položka "description" vyplněná nebo má hodnotu "null", tak se nezobrazí informační

ikona vedle inputu. Pokud však je hodnota "description" vyplněná, tak bude ikona zobrazená a po najetí na ní se zobrazí informativní hláška. Tento způsob nastavení funguje u všech komponent.

Obrázek 22: Ukázka řešení vstupu čísel 3.

Druhá varianta je pro žaluzie, které nejsou na míru a zákazník musí zvolit jednu z možností. Je to select pro výběr hodnot. Validace zde není nutná, protože máme předem nastavené hodnoty a pouze se z těchto hodnot vybírá. JSON pro tuto komponentu je velice podobný jako u výpisu 5 a najdeme v něm pouze malé rozdíly, jako například položku "options", kde jsou určeny velikosti, ze kterých může zákazník volit.

4.4.5.2 Vstup pro výběr předdefinovaných dat s výběrem pouze jedné položky

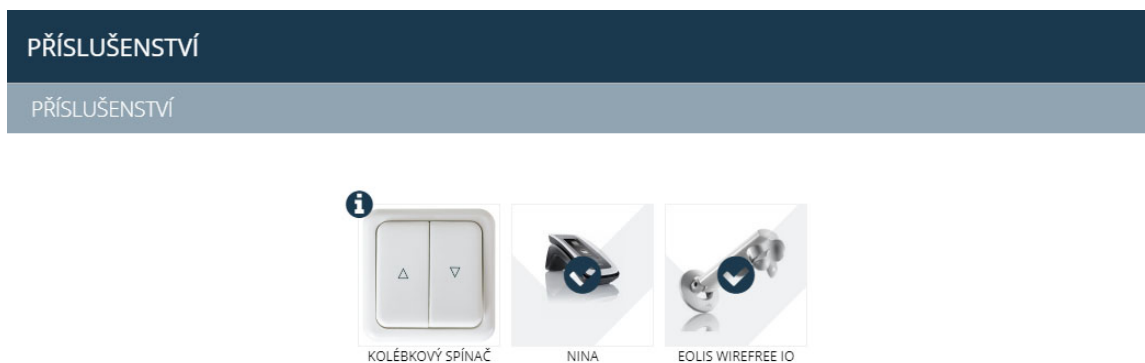
Tato komponenta je vytvářena pomocí input type="radio", kde pomocí stylu dosáhneme dobře vypadajícího vizuálu. Tato komponenta nepotřebuje validaci. Při příliš velkém počtu barev se zbytek barev schová a zobrazí se tlačítko "zobrazit více". V JSONu komponenty nalezneme pro tuto možnost nastavení, kde je položka "firstShowItem", která obsahuje číslo definující, kolik barev je po načtení zobrazeno. Dále byla opět přidána nápověda pomocí ikon.

Obrázek 23: Ukázka řešení výběru z předdefinovaných hodnot 1.

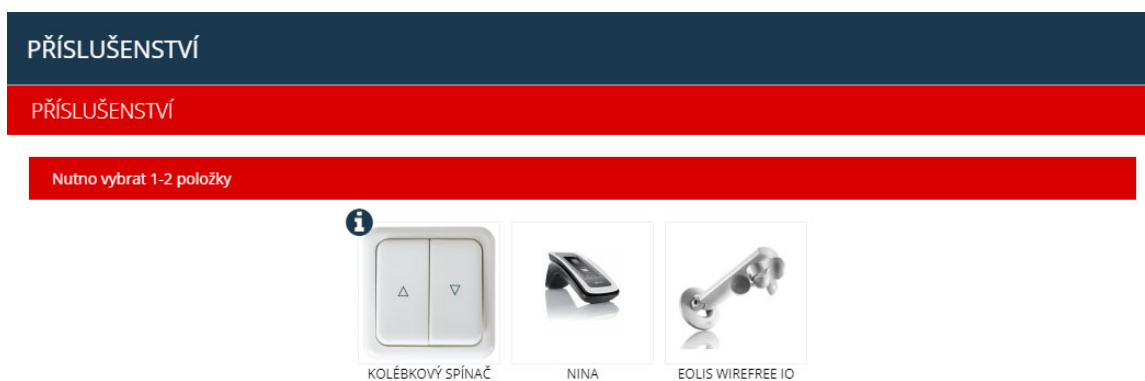
4.4.5.3 Vstup pro výběr předdefinovaných dat s výběrem více položek

Tato komponenta je vytvořena s input type="checkbox", díky čemuž můžeme zaškrtnávat více

polí. Pokud JSON obsahuje položky "min" a "max" komponenta se poté validuje. Pokud mají hodnotu "null" tak může zákazník zvolit libovolný počet položek.



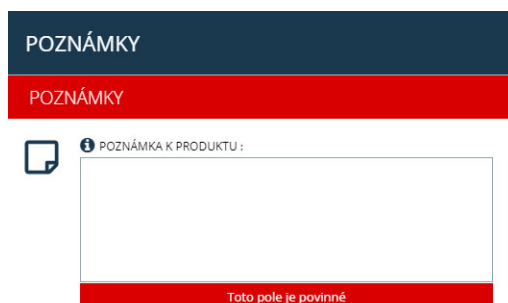
Obrázek 24: Ukázka řešení výběru z předdefinovaných hodnot 2.



Obrázek 25: Ukázka řešení výběru z předdefinovaných hodnot 3.

4.4.5.4 Vstup pro text

Element typu textarea s validací. U této komponenta se validuje pouze zda musí být hodnota vyplněna.



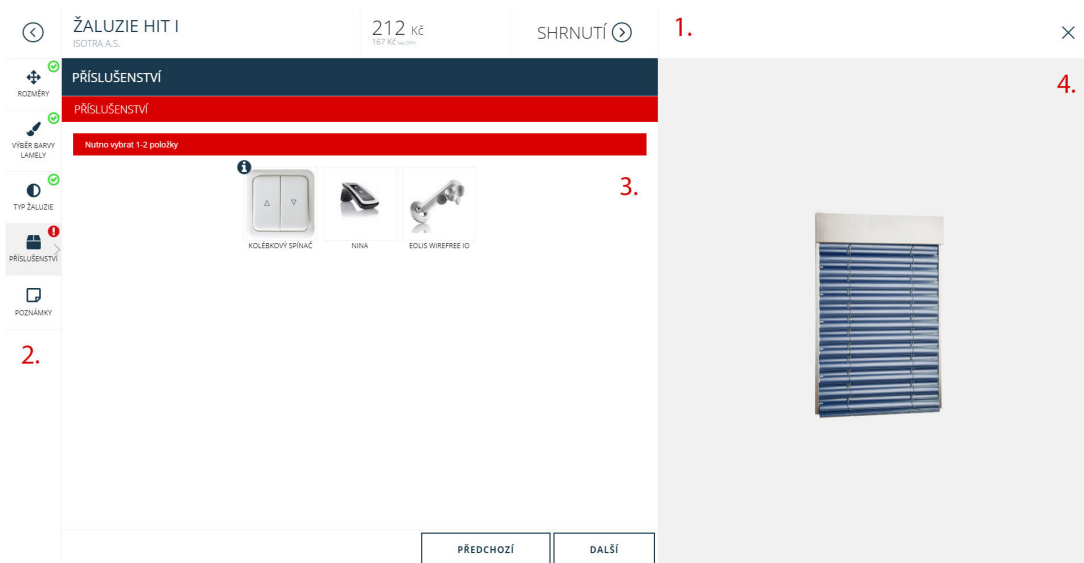
Obrázek 26: Ukázka řešení vkládání textu 1.

4.4.6 Rozložení prvků konfigurátoru a druhy prvků

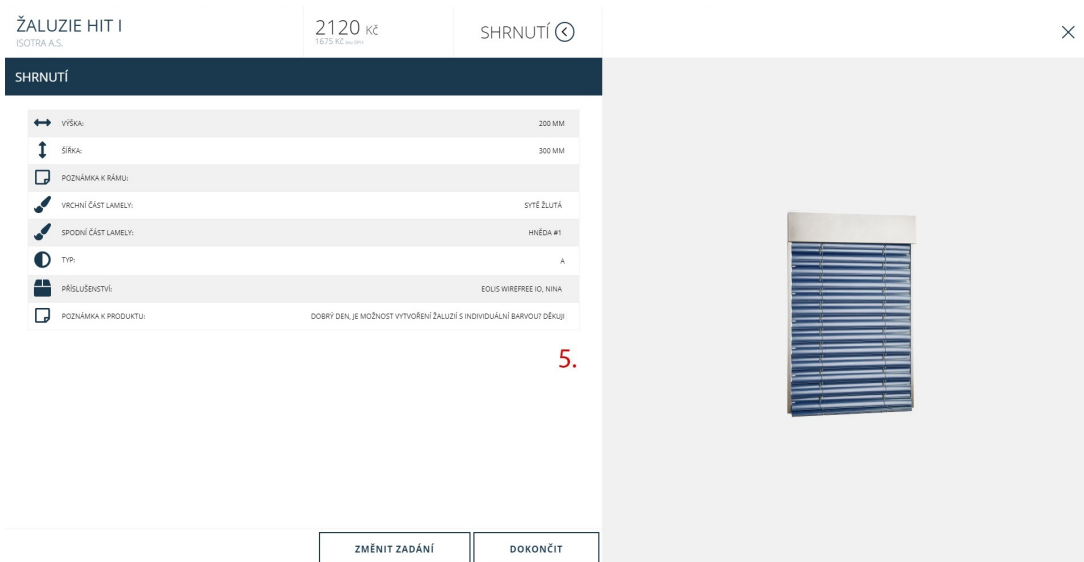
4.4.6.1 Pro počítače

Konfigurátor se otevírá do takzvaného lightboxu, čímž bylo zaručeno, že není nutné přesměrování na novou stránku. Zároveň tím bylo vytvořeno místo pro konfigurátor, jelikož není zobrazen produkt.

Komponent je hned několik: lišta s kroky, obrázek s danou žaluzií, okno s komponentami pro měnění hodnot v konfigurátoru a shrnutí výběru zákazníka.



Obrázek 27: Ukázka rozložení konfigurátoru 1.



Obrázek 28: Ukázka rozložení konfigurátoru 2.

1. Horní informační lišta obsahuje zleva tlačítko pro schování lišty kroků pro desktop. Dále informaci o žaluzii a cenu. Cena se dynamicky mění dle zvolené konfigurace. Následuje tlačítko shrnutí, které slouží k jeho zobrazení a křížek, který vyvolá pop-up okna s potvrzením zavření konfigurátoru.
2. Lišta s kroky slouží pro lepší orientaci zákazníka, aby si uvědomil kolik kroků mu chybí nakonfigurovat a kolik už má za sebou. Dále slouží pro rychlý přechod mezi kroky, pokud chce opravit své zadání. Rozhodl jsem se pro vertikální lištu z důvodu responsivu, kde je na menších zařízeních schována a zobrazuje se po kliknutí na položku v menu pro telefony. Kroky obsahují ikony a název. Kroky nelze přeskakovat, ale pokud jsme daný krok prošli můžeme se k němu vrátit. To že jsme prošli krok značí zelený háček v pravém horním rohu. Pokud nastala chyba v kroku, validace nás nepustí do jiného, dokud neopravíme ten stávající a zobrazí se vykřičník v liště kroku namísto zeleného háčku.
3. Okno obsahující komponenty, které je posouvateľné, pokud je obsah větší než výška obrazovky. Obsahuje vždy aktuální krok a v něm může být až několik částí. Kde každá část reprezentuje nějakou komponentu. Například v kroku rozměry mohou být dvě části. První část bude výška a šířka a druhá bude vstup pro text pro poznámky k rozměrům.
4. Náhled žaluzie. Tento prvek má tři možnosti zobrazování. První možnost je nastavení výchozího obrázku žaluzie. Tento obrázek se mění v průběhu průchodu jednotlivými kroky a to tak, že zobrazí ikonu aktuálního kroku. Pokud není ikona u jednotlivých kroků nastavena, zobrazí se výchozí obrázek žaluzie. Tuto možnost je vhodné zvolit, pokud prodejce má jeden obrázek k žaluzii.

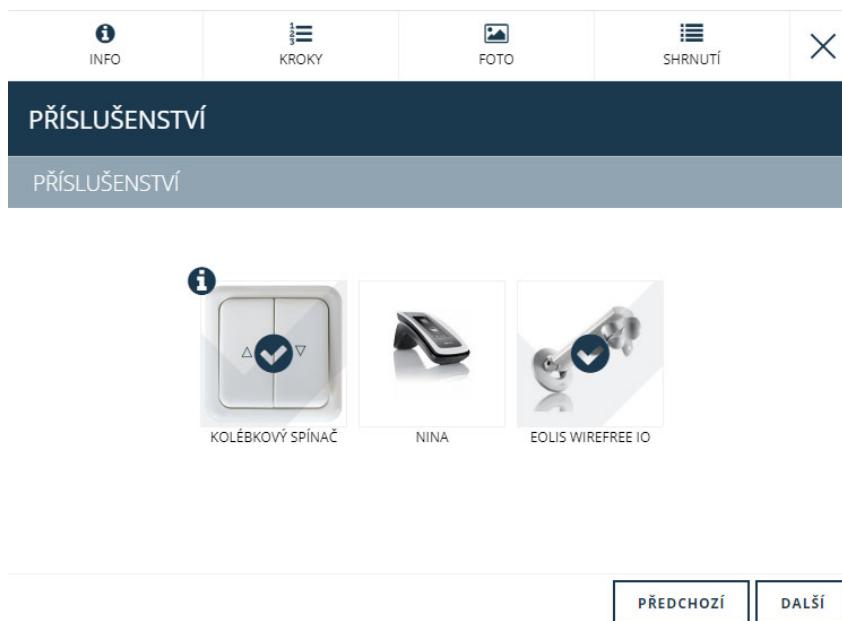
Druhá možnost je zobrazování podle aktuální změněné položky. Pokud budeme vybírat mezi barvami tak po změně výběru barvy se změní obrázek žaluzie dle této barvy nebo po výběru příslušenství se zobrazí obrázek ve větším rozlišení.

Poslední možnost je kombinace vzhledu žaluzie podle zákazníkem zvolených parametrů. Několik obrázků je poskládaných na sebe a jsou průhledné. Každý obrázek definuje jeden konfigurovatelný parametr neboli jednu část žaluzie. Při změně hodnot se mění pouze jeden obrázek to znamená změnu barvy pouze jedné části. Pokud například změníme barvu rámu, tak se změní pouze tato hodnota, ostatní hodnoty zůstanou zachovány. Dle mého názoru je toto nejlepší metoda, protože zákazník vídí, jak vypadá vzhled výsledného produktu. Tato možnost je velice pracná pro prodejce, jelikož potřebuje mít hodně obrázků žaluzií a musí vyřezávat každou část žaluzie, což je časově náročné.
5. Shrnutí je možno zapínat nebo vypínat pomocí tlačítka. Pokud změníme jakoukoliv hodnotu v krocích, tak se hodnoty shrnutí dynamicky mění. Pokud projdeme všemi kroky konfigurátoru, tak se automaticky zobrazí shrnutí z důvodu finální kontroly produktu. Pokud je shrnutí v posledním kroku, tak navíc obsahuje dvě tlačítka. Jedno pro změnění

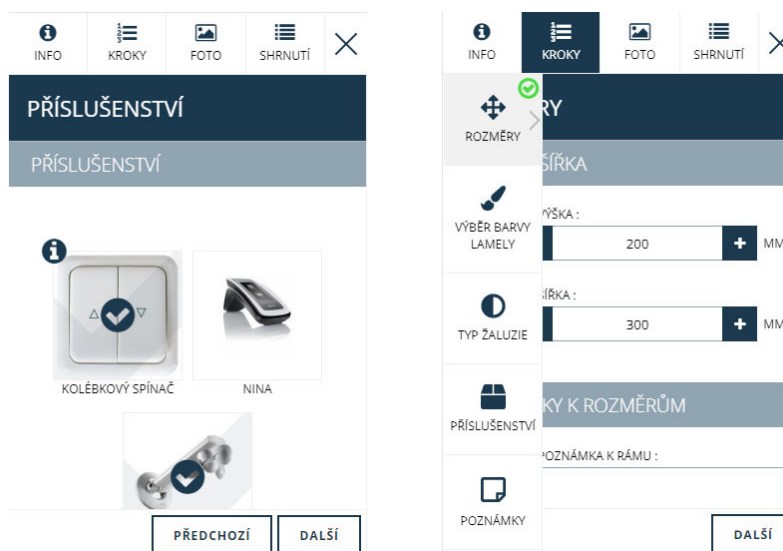
zadání a druhé pro dokončení objednávky. Pokud se konfigurace dokončí, odešle se finální JSON na server.

4.4.6.2 Responsive

Jak již bylo několikrát zmíněno, responsive je jednou z velice důležitých částí webových stránek. V dnešní době existuje mnoho odlišných rozměrů zařízení a je tedy nutné zajistit, aby se stránka zobrazovala správně na všech zařízeních.



Obrázek 29: Ukázka rozložení konfigurátoru 3.



Obrázek 30: Ukázka rozložení konfigurátoru 4.

Příliš mnoho komponent, které se nemohly zobrazit na displeji, jsem schoval a zobrazil menu pro telefony a tablety. Menu obsahuje 4 položky a při kliku na ně budou vyjždět nebo zajíždět zvolené prvky. Tímto řešením došlo k úspoře místa, konfigurátor se stal ineraktivní a dostupnější i na menších zařízeních.

4.4.7 Vypracování

Při analýze konfigurátorů jsem navštívil mnoho českých i zahraničních webových stránek konkurence a začal sepisovat podněty, které mě zaujaly a snažil vybrat z každého konfigurátoru to, co bylo uživatelsky nejpřívětivější. Prvně bylo nutno provést výběr komponent použitých pro žaluzie. Dále se zaměřit na rozvržení konfigurátoru, kde je důležité věnovat důraz na responsivní design. Poté analyzovat, který JS framework je vhodný pro práci.

Dále mohla začít práce na programování konfigurátoru. Na samém začátku programování jsem si vytvořil JSON s produkty. Produkty vypisují za pomoci PHP.

```
{
  "id": 1,
  "name": "Horizontalni zaluzie",
  "variants": 5,
  "prize": 1000,
  "image": "/images/products/horizontalni-zaluzie.jpg",
  "isConfigurateable": true,
  "configurationId": 5
},
```

Výpis 6: Ukázka JSONu produktu

Ve struktuře JSONu produktů je určeno, který konfigurátor produkt obsahuje. Po kliknutí na produkt se stáhne daný konfigurátor podle "configurationId". Vyvolání konfigurátoru je za pomoci JavaScriptové funkce:

```
var startConfigurator = document.getElementsByClassName('JS-start-configurator')
);
var startConfiguratorArr = Array.prototype.slice.call(startConfigurator)

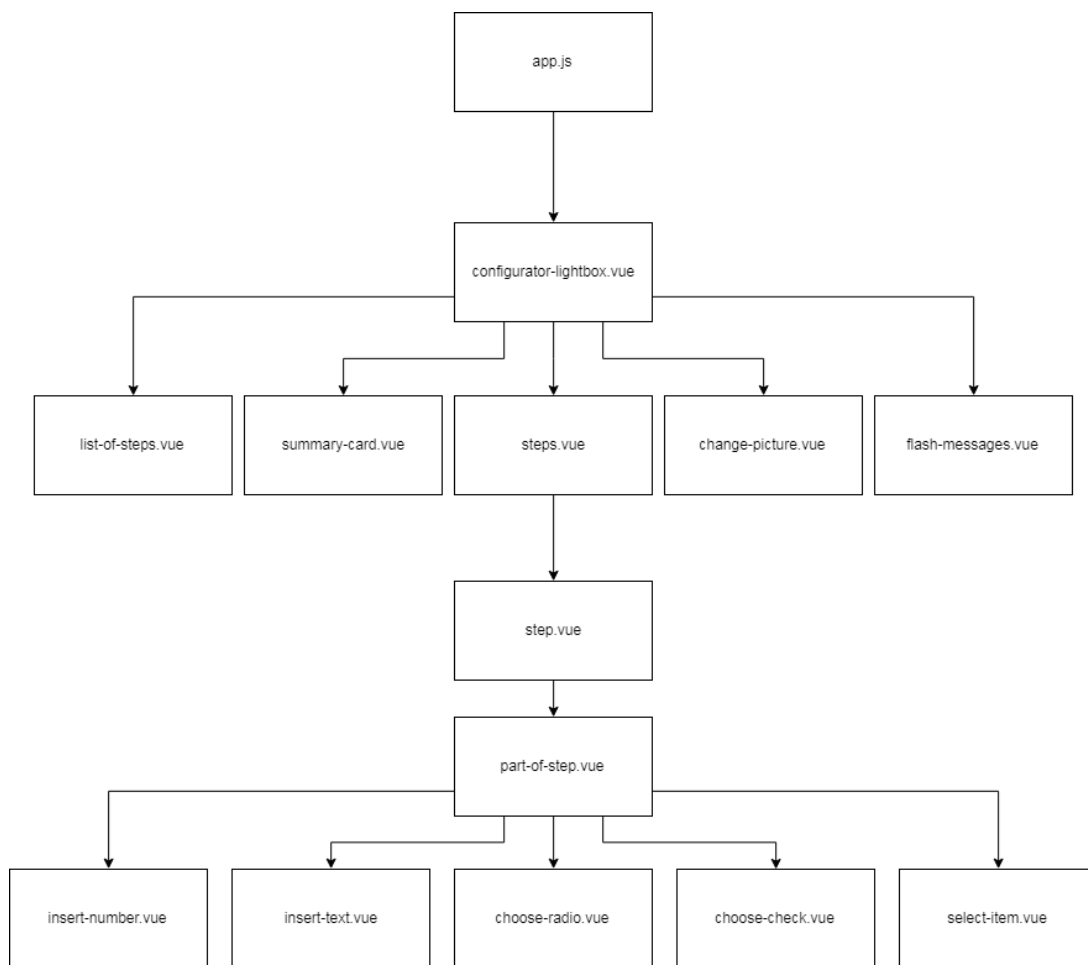
startConfiguratorArr.forEach(function (item, idx) {
  item.addEventListener('click', function () {
    Bus.$emit('change-configurator', item.dataset.confId);
  });
});
```

Výpis 7: Ukázka JS kódu pro spuštění konfigurátoru

Na začátku bylo potřeba se vypořádat s jednotlivými komponentami pro vstup čísel, vstup textu a zvolení z předdefinovaných hodnot, kterých je několik druhů. V průběhu programování docházelo k formování finálního JSONu pro konfigurátor, podle dodaných funkcí.

Bylo na zvážení, jak budou kroky konfigurátoru fungovat a vytvořit list s kroky. Pokračovat listou s informacemi a shrnutím, kde bylo důležité zařídit, aby shrnutí reagovalo na jakoukoliv změnu v kterékoliv komponentě konfigurátoru. Následně jsem pracoval na řešení náhledu žaluzie a potřebu jeho měnitelnosti. V úplném závěru přišlo na rozložení, finální stylování konfigurátoru, výběr vhodných animací pro konfigurátor a testování.

Čas realizace: 28 dní



Obrázek 31: Ukázka struktury souboru konfigurátoru 1.

5 Závěr

5.1 Znalosti získané v průběhu studia uplatněné v průběhu odborné praxe

V průběhu praxe mi také pomohly znalosti ze studia na VŠB. Nejvíce poznatků jsem využil z předmětu Vývoj internetových aplikací a Tvorba webových aplikací pro mobilní zařízení I., kde jsem si mohl vyzkoušet tvorbu webových stránek a aplikací za pomoci HTML, CSS a JS. Další důležité předměty, které mi při praxi pomohly, byly Programování I. a Programování II., kde jsem se naučil jeho základům.

5.2 Znalosti scházející v průběhu odborné praxe

Díky této odborné praxi jsem získal zkušenosti a praktické znalosti. Naučil jsem se analyzovat problém a dívat se na něj z více úhlů a hledat nová řešení. Na rozdíl od studia vyžadovala odborná praxe podrobné aktivní osvojení práce s JS frameworky. V rámci odborné praxe jsem si musel od základů osvojit znalost Vue.js. Tvorba pluginu pro WYSIWYG bylo pro mne také novinkou, stejně jako nasazování webu na systém. Prozatím jsem vždy dělal pouze statické stránky. Dalšími výhodami působení ve firmě byla možnost komunikace se zákazníky při řešení problémů dle jejich zadání a potřeb nebo týmová práce s ostatními programátory.

5.3 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Během působení ve firmě Poski.com jsem plnil více úkolů, které jsou popsány v předešlých odstavcích. Zároveň jsem měl možnost nasazení několika webu na systém, práci na systému Clevero a PoskiREAL. Mým hlavním úkolem na praxi byla implementace komfortního konfigurátoru žaluzií, který zabral více jak polovinu času mé odborné praxe. Tento konfigurátor zatím nebyl použit na žádný e-shop, ale je momentálně připravován na nasazení do jádra systému Clevero.

V rámci svého působení ve firmě jsem získal spoustu zkušeností, které mi v budoucnu mohou pomoci k získání zaměstnání, což potvrzuje i fakt, že mi bylo nabídnuto prodloužení spolupráce u této společnosti.

Literatura

- [1] Historie Poski.com [online]. Ostrava: Poski, c2018 [cit. 2018-04-01]. Dostupné z: <https://www.poski.com/poski/historie>
- [2] Introduction to HTML [online]. Kalifornie: Mozilla Corporation, c2005-2018 [cit. 2018-04-01]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML
- [3] Introduction to CSS [online]. Kalifornie: Mozilla Corporation, c2005-2018 [cit. 2018-04-01]. Dostupné z: http://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS
- [4] About JavaScript [online]. Kalifornie: Mozilla Corporation, c2005-2018 [cit. 2018-04-01]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript
- [5] Sass (Syntactically Awesome StyleSheets) [online]. Kalifornie: Hampton Catlin, Natalie Weizenbaum, Chris Eppstei, 2018 [cit. 2018-04-01]. Dostupné z: https://sass-lang.com/documentation/file.SASS_REFERENCE.html
- [6] JQuery [online]. Silver Spring: jQuery Foundation, c2018 [cit. 2018-04-01]. Dostupné z: <http://jquery.com>
- [7] Vue.js devtools [online]. Chrome web store, 2018 [cit. 2018-04-01]. Dostupné z: <https://chrome.google.com/webstore/detail/vuejs-devtools/nhdogjmejiglipccpnnnanhbledajbpd>
- [8] Introduction [online]. New Jersey: Vue.js, c2014-2018 [cit. 2018-04-01]. Dostupné z: <https://vuejs.org/v2/guide/>
- [9] SVG: Scalable Vector Graphics [online]. Kalifornie: Mozilla Corporation, c2005-2018 [cit. 2018-04-01]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/SVG>
- [10] CKEditor 4 Documentation [online]. Warszawa: CKSource, c2003-2018 [cit. 2018-04-01]. Dostupné z: <https://docs.ckeditor.com/ckeditor4/latest/index.html>
- [11] OWL Carousel [online]. Irsko: Bartosz Wojciechowski, 2013 [cit. 2018-04-01]. Dostupné z: <http://www.landmarkmlp.com/js-plugin/owl.carousel/>
- [12] AUTOR NEUVEDEN. Žaluzie24 [online]. [cit. 2018-04-01]. Dostupné z: <https://www.zaluzie24.eu/zaluzie-hit-i-1>
- [13] AUTOR NEUVEDEN. Žaluzie-online [online]. [cit. 2018-04-01]. Dostupné z: <https://www.zaluzie-online.cz/konfigurator/horizontalni-zaluzie/isotra-horizontalni-zaluzie-hit-i/>

- [14] AUTOR NEUVEDEN. Žaluzie na klíč [online]. [cit. 2018-04-01]. Dostupné z: <https://www.zaluzienaklic.cz/konfigurator/venkovni-zaluzie-zetta-90-50>
- [15] AUTOR NEUVEDEN. Lowe's [online]. [cit. 2018-04-01]. Dostupné z: <https://blinds.lowes.com/product/detail.action?sku=Bali-Northern-Heights-Wood-Blinds&groupName=Wood-Blinds&xdata-width=24&xdata-height=36>
- [16] AUTOR NEUVEDEN. Menards [online]. [cit. 2018-04-01]. Dostupné z: <https://www.menards.com/main/wtmeasuremententry.html?cid=9247>
- [17] Mobile web usage overtakes desktop for first time [online]. Londýn: James Titcomb, 2016 [cit. 2018-04-01]. Dostupné z: <https://www.telegraph.co.uk/technology/2016/11/01/mobile-web-usage-overtakes-desktop-for-first-time/>
- [18] AUTOR NEUVEDEN. ŽaluzieWeb [online]. [cit. 2018-04-01]. Dostupné z: <https://zaluzieweb.cz/zaluzie-na-okna-isso-1-25mm-bezne-odstiny/p-959015c21141>
- [19] AUTOR NEUVEDEN. Žaluzie na míru [online]. [cit. 2018-04-01]. Dostupné z: <https://www.zaluzienamiru.cz/zaluzie-standard>
- [20] AUTOR NEUVEDEN. Levolor [online]. [cit. 2018-04-01]. Dostupné z: <https://www.levolor.com/configurator?prebuiltId=881>
- [21] 5 Reasons Why You Shouldn't be Using Flash on Your Website [online]. Indianapolis: Stephanie Fisher, c2005-2018 [cit. 2018-04-01]. Dostupné z: <https://blog.spinweb.net/10-reasons-why-you-shouldnt-be-using-flash-on-your-website>
- [22] Flash pomalu mizí z internetu. Vadilo zabezpečení i náročnost [online]. Praha: Technet, 2015 [cit. 2018-04-01]. Dostupné z: https://technet.idnes.cz/adobe-flash-nahrazuje-html5-ddi-/sw_internet.aspx?c=A150818_100832_sw_internet_vse
- [23] Máte na webu flash? Zbavte se ho co nejrychleji. [online]. Králův Dvůr: Tomáš Krause, 2017 [cit. 2018-04-01]. Dostupné z: <https://tomaskrause.cz/mate-webu-flash-zbavte-se-nejrychleji/>
- [24] Angular vs. React vs. Vue: A 2017 comparison [online]. Berlin: Jens Neuhaus, 2017 [cit. 2018-04-01]. Dostupné z: <https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176>