

**STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV  
ROČNÍKOVÁ PRÁCE**

Tadeáš Keller



Mladá Boleslav 2025

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**MERN AUTOMOBILOVÝ KONFIGURÁTOR**

Autor: Tadeáš Keller

Studijní obor: 18-20-M/01 Informační technologie

Vedoucí práce: Jan Till

Mladá Boleslav 2025

## Obsah

1	Úvod.....	6
2	Obsah.....	8
2.1	Obecná struktura kódu a aplikace.....	8
2.2	Design stránek .....	9
2.2.1	Hlavní komponenty.....	9
2.3	Backend .....	10
2.3.1	Připojení backendu k databázi.....	10
2.3.2	Struktura dat v databázi.....	11
2.4	Dotazník.....	12
2.5	Konfigurátor .....	13
2.6	Admin systém .....	14
2.6.1	Formulář pro vytváření modelů.....	14
2.6.2	Formulář pro zobrazení a úpravu modelu .....	15
3	Závěr .....	17
4	Přílohy.....	18
4.1	Seznam obrázků .....	18
4.2	Zdroje.....	18



### **Prohlášení**

Prohlašuji, že jsem svou ročníkovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

Nemám závažný důvod proti zpřístupňování této ročníkové práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Mladé Boleslavi dne ..... podpis: .....



### **Poděkování**

Děkuji mému vedoucímu Janu Tillovi za pomoc a vstřícnost při zpracování této práce.



# 1 Úvod

Cílem mé práce bylo vytvořit automobilový konfigurátor s možností ukládat možné konfigurace do databáze. Součástí aplikace měl být i systém pro správu konfigurací, do kterého se mohou přihlásit správci a upravovat konfigurace přímo v aplikaci.

Automobilový konfigurátor je webová aplikace, která má za úkol ulehčit zákazníkovi výběr modelu a přizpůsobení různých vlastností, jako např. barva, výbava, motorizace, před nákupem nového vozidla. Tento typ aplikace je mezi světovými automobilkami velmi rozšířený, zejména z důvodu uživatelské přívětivosti.

Toto téma jsem si vybral především proto, že celý svůj život žiji v Mladé Boleslavi, kde sídlí největší česká společnost na výrobu osobních automobilů. Celý region je tak na automobilový průmysl velmi silně napojený. Z toho důvodu jsem chtěl vytvořit aplikaci, která bude s tímto odvětvím souviset a tím rozšířit svoje znalosti, které mi mohou být užitečné v budoucím zaměstnání.

Pro vytvoření aplikace jsem využil. MERN stacku, který zahrnuje 4 technologie: React<sup>1</sup>, MongoDB<sup>2</sup>, Express.js<sup>3</sup> a Node.js<sup>4</sup>. Výhodou MERN stacku je, že celý vývoj probíhá v jednom programovacím jazyce – JavaScriptu. Navíc je open-source a má dostupnou dokumentaci.

Celý projekt byl průběžně ukládán do repozitáře na platformě GitHub<sup>5</sup>.

Jako vývojové prostředí jsem použil editor Visual Studio Code<sup>6</sup>, pro jeho přehlednost, rychlost, možnosti přizpůsobení a jeho podporu pro Git<sup>7</sup>.

---

<sup>1</sup> React. Online. 2025. Dostupné z: <https://react.dev/>. [cit. 2025-04-25].

<sup>2</sup> MongoDB. Online. 2025. Dostupné z: <https://www.mongodb.com/>. [cit. 2025-04-25].

<sup>3</sup> Express.js. Online. 2025. Dostupné z: <https://expressjs.com/>. [cit. 2025-04-25].

<sup>4</sup> Node.js. Online. 2025. Dostupné z: <https://nodejs.org/en>. [cit. 2025-04-25].

<sup>5</sup> GitHub. Online. 2025. Dostupné z: <https://github.com/>. [cit. 2025-04-25].



V aplikaci se nachází několik obrázků vygenerovaných pomocí umělé inteligence Microsoft Designer<sup>8</sup>, konkrétně je jeden na domovské stránce jako dekorační prvek a dva jsou u ukázkových modelů pro doplňkovou výbavu.

U ukázkových modelů, které se nacházejí v aplikaci jsem využil umělou inteligenci ChatGPT<sup>9</sup> k vygenerování čtyř obrázků pro konfigurátor.



Obrázek 1: obrázek ukázkového modelu

Pro různé potřeby bylo využíváno grafických editorů Adobe Express<sup>10</sup> (např.: jednoduché logo) a Canva<sup>11</sup>.

---

<sup>6</sup> VisualStudioCode. Online. 2025. Dostupné z: <https://code.visualstudio.com/>. [cit. 2025-04-25].

<sup>7</sup> Git. Online. 2025. Dostupné z: <https://git-scm.com/>. [cit. 2025-04-28].

<sup>8</sup> Microsoft Designer. Online. 2025. Dostupné z: <https://designer.microsoft.com/>. [cit. 2025-04-25].

<sup>9</sup> ChatGPT. Online. 2025. Dostupné z: <https://chatgpt.com/>. [cit. 2025-04-27].

<sup>10</sup> Adobe Express. Online. 2025. Dostupné z: <https://www.adobe.com/express/>. [cit. 2025-04-27].

<sup>11</sup> Canva. Online. 2025. Dostupné z: <https://www.canva.com/>. [cit. 2025-04-27].



## 2 Obsah

### 2.1 Obecná struktura kódu a aplikace

Celá aplikace je rozdělena do dvou základních částí podle své funkce – client a server. Tento způsob rozdělení je dnes běžný u moderních webových aplikací, protože umožňuje oddělit logiku uživatelského rozhraní od serverové logiky a práce s databází. Tato architektura zlepšuje modularitu aplikace, což usnadňuje její údržbu a rozšiřování.

Client představuje tu část aplikace, kterou uživatel přímo vidí a s níž interaguje (frontend). Zajišťuje vykreslování jednotlivých stránek, správu formulářů, navigaci a další interaktivní prvky. Klientská část také obsahuje logiku pro zpracování vstupů od uživatele a odesílání požadavků na server. Tato část je navržena tak, aby poskytovala co nejlepší uživatelský zážitek a rychlost odezvy.

Naopak serverová část slouží k obsluze požadavků, které přicházejí z klientské části (backend). Mezi její hlavní úkoly patří načítání dat z databáze, jejich úprava nebo uložení nových informací. Serverová část zajišťuje komunikaci s databází a poskytuje odpovědi prostřednictvím API endpointů.

Ještě před samotným vývojem aplikace bylo důležité navrhnout její strukturu. Bylo potřeba určit, z jakých podstránek se bude skládat, jaké informace bude každá z nich zobrazovat a jaké funkce bude plnit. Tento návrh zajistil, že vývoj probíhal systematicky a že všechny části aplikace jsou navzájem propojeny a funkční.

Pro běžného uživatele – potenciálního zákazníka – jsou k dispozici podstránky s výpisem všech modelů dostupných ke konfiguraci, samotný konfigurátor, hlavní stránka a další doplňkové sekce. Tyto podstránky jsou navrženy tak, aby byly intuitivní a snadno použitelné.

Pro uživatele, který se přihlásí jako správce aplikace, jsou určeny administrační formuláře pro jednotlivé akce s daty na databázi – přidání modelu, úpravu a





odstranění. K těmto formulářům se správce dostane přes podstránku, která funguje jako rozcestník. Tato podstránka je navržena tak, aby správci poskytovala rychlý a snadný přístup ke všem důležitým funkcím.

## 2.2 Design stránek

K návrhu vzhledu aplikace byl využit online grafický nástroj Figma<sup>12</sup>, pomocí kterého bylo možné ještě před zahájením vývoje vizuálně rozvrhnout komponenty, barevné kombinace a další designové prvky.

Pro tvorbu samotných komponent byl zvolen CSS framework Tailwind<sup>13</sup>, zejména kvůli jeho jednoduchému použití a dostupné dokumentaci. Zároveň je na stránce využíváno jeho předdefinovaných barev, což zajišťuje konzistenci designu po celé aplikaci. V aplikaci převažuje barevná kombinace černé a žluté, doplněná bílou pro dobrou čitelnost.

Nejvýraznější částí aplikace je domovská stránka, která obsahuje upoutávku na konfigurátor. V kombinaci s headerem vytváří parallax efekt, čímž pozadí působí prostorově odděleně od ostatních prvků. Tento efekt přidává stránce dynamiku.

### 2.2.1 Hlavní komponenty

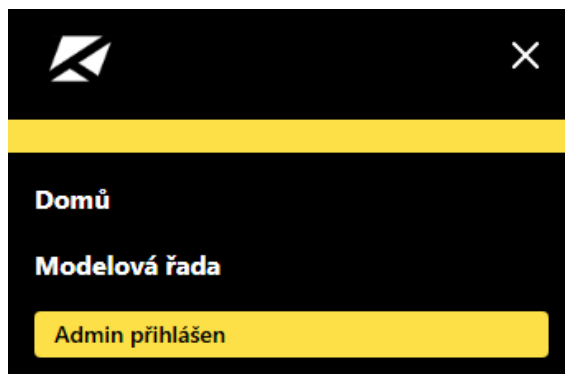
Navigace je zajištěna responzivním headerem, který obsahuje odkazy na klíčové podstránky. Pokud je přihlášen správce, zobrazí se i žlutý ukazatel. V případě mobilního zařízení se odkazy skryjí a objeví se až po kliknutí na ikonu menu z knihovny lucide-react<sup>14</sup>.

---

<sup>12</sup> Figma. Online. 2025. Dostupné z: <https://www.figma.com/>. [cit. 2025-04-27].

<sup>13</sup> Tailwindcss. Online. 2025. Dostupné z: <https://tailwindcss.com/>. [cit. 2025-04-27].

<sup>14</sup> Lucide React. Online. 2025. Dostupné z: <https://lucide.dev/guide/packages/lucide-react>. [cit. 2025-04-27].



Obrázek 2: header na mobilním zařízení

Na spodní části každé stránky se nachází přehledný footer s odkazy na méně využívané sekce, včetně přístupu do administračního systému. Footer je navržen tak, aby byl jednoduchý a neodváděl pozornost od hlavního obsahu stránky, přitom však poskytuje důležité informace a odkazy.

## 2.3 Backend

Backendová (server) část aplikace byla vytvořena pomocí serverového frameworku (Node.js a Express.js). Ten zajišťuje správu jednotlivých API endpointů, které zprostředkovávají komunikaci mezi klientskou částí a databází. Každý endpoint odpovídá určité akci – například získání všech modelů, přidání nového modelu, aktualizace nebo smazání existujícího.

Endpoints jsou rozděleny do samostatných souborů pro přehlednost. Například soubor **routes/models.js** obsahuje všechny operace týkající se správy modelů. Takové rozdělení usnadňuje přehlednost a budoucí rozšiřování kódu.

### 2.3.1 Připojení backendu k databázi

Aplikace je připojena k databázi MongoDB pomocí knihovny Mongoosejs<sup>15</sup>, která práci s databází výrazně usnadňuje. Connection string potřebný pro připojení je

---

<sup>15</sup> Mongoose. Online. 2025. Dostupné z: <https://mongoosejs.com/>. [cit. 2025-04-27].



uložen v souboru **.env**, který je zapsán v **.gitignore**, aby se neuložil na veřejný GitHub. Pro načtení proměnných z **.env** je využita knihovna **dotenv**<sup>16</sup>.

```
require("dotenv").config();
const { MONGO_URL } = process.env;
mongoose.connect(MONGO_URL)
  .then(() => console.log("Database connected"));
```

Zdrojový kód č.1

Mongoose umožňuje kromě samotného připojení i definování datových struktur pomocí schémat.

```
const ColorSchema = new mongoose.Schema({
  name: { type: String, required: true },
  price: { type: Number, required: true },
  rims: { type: [RimSchema], required: true }
});
```

Zdrojový kód č.2

### 2.3.2 Struktura dat v databázi

Data v databázi jsou ukládána ve formě JSON souboru.

*JavaScript Object Notation (JSON) je způsob zápisu dat nezávislý na počítačové platformě, určený pro přenos dat, která mohou být organizována v polích nebo agregována v objektech. Vstupem je libovolná datová struktura (číslo, řetězec, boolean, objekt nebo z nich složené pole), výstupem je vždy řetězec.“*

*JavaScript Object Notation.* Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-2025. Dostupné z: [https://cs.wikipedia.org/wiki/JavaScript\\_Object\\_Notation](https://cs.wikipedia.org/wiki/JavaScript_Object_Notation). [cit. 2025-04-27].

V prvních verzích aplikace bylo možné každému modelu nadefinovat libovolné parametry, což vedlo k problémům při správě obrázků. Proto byla struktura sjednocena, a každý model má nyní pevně dané parametry: název, barvy, kola, motorizace a doplňkovou výbavu. Kvůli jednodušší práci s obrázky je ke každé barvě definována vlastní sada kol a každá kombinace má svůj vlastní obrázek, který se

---

<sup>16</sup> Dotenv. Online. 2025. Dostupné z: <https://www.dotenv.org/>. [cit. 2025-04-27].



zobrazuje přímo v konfigurátoru. Tento přístup usnadňuje práci s daty a zajišťuje, že všechny obrázky jsou správně přiřazeny k jednotlivým variantám.

To je také důvod, proč byla zvolena databáze MongoDB, která umožňuje ukládat komplexní struktury do jednoho dokumentu. Použití relační databáze by vyžadovalo rozdělení dat do více tabulek a použití cizích klíčů pro řešení vzorců relací, což by komplikovalo správu dat a zvýšilo složitost

V databázi jsou také uloženy údaje o správcích – konkrétně jejich ID a hash hesla. Správci musí být přidáni přímo přes MongoDB rozhraní, což zajišťuje, že správu přístupů má pouze oprávněná osoba. Hesla musí být před uložením zahashována a při přihlášení se porovnává zadané heslo s tím, které je uloženo v databázi. Tento systém zajišťuje bezpečnost přístupů do administračního rozhraní a chrání citlivé údaje.

## 2.4 Dotazník

Uživatelé mají možnost zaslat dotaz nebo zpětnou vazbu pomocí speciální podstránky s formulářem. Ten odesílá zprávu na vybraný e-mail prostřednictvím služby EmailJS<sup>17</sup>. Tato služba umožňuje odesílání e-mailů přímo z frontendu (client) bez nutnosti vlastního serveru. Jedná se o praktické řešení, které využívá šablony nastavené v rozhraní EmailJS a propojení s běžnými e-mailovými službami.

Při odeslání formuláře jsou data (například jméno, e-mail a zpráva) odeslána přes zabezpečené rozhraní API do služby EmailJS, která na základě zvolené šablony vygeneruje e-mail a odešle ho na předem definovanou adresu. Výhodou tohoto přístupu je jednoduchá implementace, možnost snadného přizpůsobení vzhledu e-mailu a také to, že zprávy nejsou ukládány v databázi, ale dorazí přímo do e-mailové schránky správce.

V rámci projektu byla vytvořena vlastní šablona e-mailu a propojena s identifikátorem služby (service ID) a šablony (template ID), které poskytuje rozhraní EmailJS. E-mailová adresa příjemce je rovněž nastavena přímo ve webové aplikaci

---

<sup>17</sup> EmailJS. Online. 2025. Dostupné z: <https://www.emailjs.com/>. [cit. 2025-04-27].



EmailJS, takže uživatel nemá možnost ji upravit. Celý systém tak umožňuje snadnou a bezpečnou komunikaci mezi uživatelem a správcem systému.

Při implementaci formuláře byly využity poznatky z výukového videa od uživatele TheIvoryCoder<sup>18</sup> na platformě YouTube, které názorně ukazuje propojení formuláře v Reactu s touto službou.

Odkaz na podstránku s tímto formulářem je k nalezení ve footeru.

## 2.5 Konfigurátor

Nejdůležitější součástí aplikace je konfigurátor. Uživatel se do něj dostane kliknutím na tlačítko „konfigurovat“ v seznamu u konkrétního modelu. V seznamu se vedle modelu zobrazuje vždy indexově první obrázek pro daný model.

```
const image = model.colors?.[0]?.rims?.[0]?.image || "/thumbnails/imgErr.jpg";
```

Obrázek 3: kód pro logiku zobrazení obrázku v seznamu modelů

Při výběru barev a kol se mění náhledový obrázek modelu. Aktuálně zvolený parametr je vždy zvýrazněn žlutou barvou. U doplňkové výbavy se také zobrazuje ilustrační obrázek. Změny v konfiguraci se okamžitě promítají do souhrnné tabulky na konci stránky.

Po dokončení konfigurace automobilu má uživatel možnost si svou volbu exportovat do souboru ve formátu PDF. Tato funkce slouží jako přehledná rekapitulace celé konfigurace a může být využita například při komunikaci s prodejcem nebo pro vlastní archivaci. Export probíhá kliknutím na tlačítko, které spustí generování PDF dokumentu na straně serveru.

Funkcionalita je postavena na knihovnách html2canvas a jsPDF<sup>19</sup>. Nejprve se pomocí html2canvas<sup>20</sup> vytvoří obrazová reprezentace vybraných částí stránky, která je následně převedena na PDF a automaticky uložena pomocí jsPDF.

---

<sup>18</sup> TheIvoryCoder. Online. <https://www.youtube.com/>. 2025. Dostupné z: <https://www.youtube.com/@theivorycoder>. [cit. 2025-04-27].



## 2.6 Admin systém

Správu modelů umožňuje speciální administrační rozhraní. Po zadání přihlašovacích údajů do přihlašovacího formuláře se ověří ID správce, načte se hash jeho hesla z databáze a pomocí zvláštní metody knihovny bcrypt<sup>21</sup> porovná se zadaným stringem. Pokud jsou údaje správné, vygeneruje se token a správce se přihlásí do systému. Token se generuje pomocí nástroje jsonwebtoken<sup>22</sup>.

```
const match = await bcrypt.compare(req.body.password, admin.password);
console.log("Porovnání hesel:", match);
if (!match) {
  return res.status(401).json({ message: "Neplatné přihlašovací údaje" });
}
// Generování jwt
const token = jwt.sign(
  { adminId: admin._id },
  process.env.JWT_SECRET,
  { expiresIn: "1h" }
);
res.status(200).json({
  message: "Přihlášení úspěšné",
  token
});
```

Obrázek 4: kód pro ověření hesla a generování tokenu

### 2.6.1 Formulář pro vytváření modelů

Formulář pro vytvoření modelu umožňuje správci zadat název modelu, základní cenu a typ karoserie. Následně je možné přidávat barvy a k nim přiřazené sady kol. U obou parametrů se zadává název a cena, u kol navíc ještě cesta k obrázku, který je uložen ve složce public v klientské části projektu. Obrázky se musí přidat přímo do aplikace do složky **public** na klientské části.

---

<sup>19</sup> JsPDF. Online. <https://www.youtube.com/>. 2025. Dostupné z: <https://github.com/parallax/jsPDF>. [cit. 2025-04-27].

<sup>20</sup> Html2canvas. Online. 2025. Dostupné z: <https://github.com/niklasvh/html2canvas>. [cit. 2025-04-27].

<sup>21</sup> Bcrypt. Online. 2025. Dostupné z: <https://github.com/kelektiv/node.bcrypt.js>. [cit. 2025-04-28].

<sup>22</sup> Jsonwebtoken. Online. Dostupné z: <https://github.com/auth0/node-jsonwebtoken>. [cit. 2025-04-27].



Po zadání informací o jedné barvě se doplní kola, poté se tato sada potvrdí a je možné pokračovat další barvou. Správce má k dispozici přehledné tabulky pod jednotlivými částmi formuláře, které reflektují aktuálně zadaná data.

Po vyplnění barev a kol se zadávají dostupné motorizace a doplňková výbava, i zde se průběžně generují přehledné tabulky. Vytvoření modelu je dokončeno kliknutím na tlačítko „vytvořit model“.

Obrázek 5: Náhled formuláře pro přidání nového modelu

## 2.6.2 Formulář pro zobrazení a úpravu modelu

Po kliknutí na možnost „spravovat konfigurace“ na admin stránce se správci zobrazí seznam všech uložených modelů. Odtud je možné zobrazit detail vybraného modelu nebo jej upravit.

Detail modelu je vytvořen pomocí přehledných tabulek, které obsahují všechny parametry daného modelu.

Podstránka pro úpravu modelu, je téměř shodná se stránkou pro zobrazení modelu, ale nyní lze obsah tabulek přepisovat. V tabulkách se totiž informace zobrazují jako „inputy“ a aktuální hodnoty jako jejich hodnoty. Po úpravě stačí stisknout tlačítko „uložit změny“ a změny se v databázi přepíší.

Pokud chce správce nějaký model smazat, může tak učinit v seznamu všech modelů, nebo po rozkliknutí některého z nich, stisknutím tlačítka „smazat model“.



Všechny důležité kroky, jako je vytvoření, úprava nebo smazání modelu, jsou doprovázeny potvrzovacími zprávami, které informují správce o úspěšnosti nebo případné chybě při dané operaci.





### 3 Závěr

Ve výsledku se mi tedy povedlo splnit hlavní vytyčené body práce. Aplikace umožňuje konfigurovat vozy a součástí je i admin systém.

Povedlo se mi přidat navíc pár funkcí jako formulář pro odesílání zpětné vazby a dotazů formou emailu.

Projekt jsem měl v plánu ještě více rozšířit, ale to jsem bohužel nestihl. Chtěl jsem například přidat funkci na uložení nakonfigurovaného modelu na databázi, aby mohl systém fungovat i jako takový objednávkový systém. Určitě by se také dal vylepšit vzhled stránky aby působil moderněji a uceleněji.

Můžu říct, že jsem si tímto projektem prohloubil znalosti vytváření dynamických webových stránek. Před tímto projektem jsem s databázemi takřka nepracoval.



## 4 Přílohy

### 4.1 Seznam obrázků

Obrázek 1: obrázek ukázkového modelu .....	7
Obrázek 2: header na mobilním zařízení .....	10
Obrázek 3: kód pro logiku zobrazení obrázku v seznamu modelů .....	13
Obrázek 4: kód pro ověření hesla a generování tokenu .....	14
Obrázek 5: Náhled formuláře pro přidání nového modelu .....	15

### 4.2 Zdroje

React. Online. 2025. Dostupné z: <https://react.dev/>. [cit. 2025-04-25].

MongoDB. Online. 2025. Dostupné z: <https://www.mongodb.com/>. [cit. 2025-04-25].

Express.js. Online. 2025. Dostupné z: <https://expressjs.com/>. [cit. 2025-04-25].

Node.js. Online. 2025. Dostupné z: <https://nodejs.org/en>. [cit. 2025-04-25].

GitHub. Online. 2025. Dostupné z: <https://github.com/>. [cit. 2025-04-25].

Git. Online. 2025. Dostupné z: <https://git-scm.com/>. [cit. 2025-04-28].

VisualStudioCode. Online. 2025. Dostupné z: <https://code.visualstudio.com/>. [cit. 2025-04-25].

Microsoft Designer. Online. 2025. Dostupné z: <https://designer.microsoft.com/>. [cit. 2025-04-25].

ChatGPT. Online. 2025. Dostupné z: <https://chatgpt.com/>. [cit. 2025-04-27].



Adobe Express. Online. 2025. Dostupné z: <https://www.adobe.com/express/>. [cit. 2025-04-27].

Canva. Online. 2025. Dostupné z: <https://www.canva.com/>. [cit. 2025-04-27].

Figma. Online. 2025. Dostupné z: <https://www.figma.com/>. [cit. 2025-04-27].

Tailwindcss. Online. 2025. Dostupné z: <https://tailwindcss.com/>. [cit. 2025-04-27].

Lucide React. Online. 2025. Dostupné z: <https://lucide.dev/guide/packages/lucide-react>. [cit. 2025-04-27].

Mongoose. Online. 2025. Dostupné z: <https://mongoosejs.com/>. [cit. 2025-04-27].

Dotenv. Online. 2025. Dostupné z: <https://www.dotenv.org/>. [cit. 2025-04-27].

JavaScript Object Notation. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-2025. Dostupné z: [https://cs.wikipedia.org/wiki/JavaScript\\_Object\\_Notation](https://cs.wikipedia.org/wiki/JavaScript_Object_Notation). [cit. 2025-04-27].

EmailJS. Online. 2025. Dostupné z: <https://www.emailjs.com/>. [cit. 2025-04-27].

TheIvoryCoder. Online. <https://www.youtube.com/>. 2025. Dostupné z: <https://www.youtube.com/@theivorycoder>. [cit. 2025-04-27].

JsPDF. Online. <https://www.youtube.com/>. 2025. Dostupné z: <https://github.com/parallax/jsPDF>. [cit. 2025-04-27].

Html2canvas. Online. 2025. Dostupné z: <https://github.com/niklasvh/html2canvas>. [cit. 2025-04-27].

Bcrypt. Online. 2025. Dostupné z: <https://github.com/kelektiv/node.bcrypt.js>. [cit. 2025-04-28].



Jsonwebtoken. Online. Dostupné z: <https://github.com/auth0/node-jwt>. [cit. 2025-04-27].