STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

Autor: Jan Poláček

Mladá Boleslav 2022

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

**Autor: Jan Poláček  
Studijní obor: 18-20-M/01 Informační technologie  
Vedoucí práce: Lukáš Mázl**

Mladá Boleslav 2022

# Obsah

[Obsah 3](#_Toc102320087)

[1 Úvod 6](#_Toc102320088)

[2 Obsah práce 7](#_Toc102320089)

[2.1 Využité knihovny 7](#_Toc102320090)

[2.1.1 Node.js 7](#_Toc102320091)

[2.1.2 Express.js 7](#_Toc102320092)

[2.1.3 Socket.io 7](#_Toc102320093)

[2.1.4 Translate 7](#_Toc102320094)

[2.2 Problematika backendu 8](#_Toc102320095)

[2.2.1 Řešení přijetí a odesílání dat 8](#_Toc102320096)

[2.2.2 Řešení překladu dat 8](#_Toc102320097)

[2.2.3 Chyba při překladu 9](#_Toc102320098)

[2.3 Problematika frontendu 9](#_Toc102320099)

[2.3.1 Design/Vzhled stránky 9](#_Toc102320100)

[2.3.2 Řešení odesílání dat 10](#_Toc102320101)

[2.3.3 Řešení přijetí dat 11](#_Toc102320102)

[2.3.4 Vlaječky pro jazyky 11](#_Toc102320103)

[2.3.5 Užitečná tlačítka 11](#_Toc102320104)

[3 Závěr 12](#_Toc102320105)

[4 Přílohy 13](#_Toc102320106)

[4.1 Zdroje 13](#_Toc102320107)

**Prohlášení**

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

# Úvod

Smyslem této práce bylo vytvořit takový překladač jazyků, jako je například na webu Google Překladače. Tento překladač překládá z jednoho jazyka do druhého jazyka a tím přetlumočí uživateli, jak se v překládaném jazyce sděluje text, který chce přeložit. Překladač disponuje samotnou Google Api, která má v sobě zabudovanou databázi překladů a umělou inteligenci, kterou můžeme zdarma využít po omezenou dobu. Projekt by měl zpříjemnit jak uživatelské rozhrání, které by mělo být jednoduché na pochopení, tak i průhlednost a jednoduchost s kódem, který pracuje na pozadí.

Program by měl fungovat na bázi dotazů a odpovědí, přes Express a Socket.io. Když se uživatel připojí na určitou adresu, server mu odpoví a pošle základní webové rozhrání, které je vytvořeno přes HTML a CSS. Na rozhrání uživatel pošle formulář překladu zpátky serveru a server následně argumenty z formuláře přeloží na jiný jazyk, jenž si uživatel sám zvolil. Následně by se měla odeslat přeložená odpověď zpátky uživateli a překlad se mu ukáže na rozhrání, pomocí Javascriptu.

# Obsah práce

## Využité knihovny

### Node.js

Node.js v tomto programu představuje hlavní softwarový systém, ve kterém proudí všechny informace a kódy, které na frontendu nebo backendu naprogramujeme. Dále do Node.js můžeme stahovat další pomocné frameworky, pro lepší a jednodušší práci s daty.

### Express.js

Knihovna Express.js za nás řeší připojení na URL adresu. Společně s odkazy můžeme tuto knihovnu využít na routování mezi soubory webu. Jedná se v podstatě o jeden z hlavních balíčků npm, který potřebujeme na backend pro správnou funkci na webu pro frontend.

### Socket.io

Když uživatel na backend serveru pošle nějaký dotaz, tak potřebuje dostat nějakou odpověď nazpátek. Knihovna Socket.io za nás řeší připojení uživatele na server, a pomocí různých emitů, posílá provedené konstanty zpátky uživateli. Využívá se jak na frontendu, tak i na backendu.

### Translate

Tento framework řeší za nás překlad z jednoho jazyka na druhý jazyk. Translate také potřebuje ke svému běhu vlastní Google Api klíč překladače, který se dá založit na omezenou dobu a s určitým maximálním počtem dotazů.

## Problematika backendu

Program funguje na bázi dotazů a odpovědí, to znamená že se z uživatelské strany musí brát nějaké dotazy, které by měl sám server zpracovat a následně zpátky uživateli odeslat. Program tedy běží na softwarovém systému Node.js a knihovně Socket.io, která nám zaručuje komunikaci mezi backendem a frontendem.

### Řešení přijetí a odesílání dat

Aby se data dala od uživatele nějak převzít, byla zvolena Socket.io knihovna. V našem případě, když se uživatel na server připojí, server čeká, dokud uživatel nepošle dotaz o překlad a následně se odešle uživateli odpověď o přeloženém textu zpátky.

io.on('connection', (socket) => {

socket.on('translate', (args) => {

**if** (args[2] !== '') {

**const** start = **function** () {

**return** 'test';

}

**const** output = start();

console.**log**(socket.id + "[" + args + "]: " + output);

io.to(socket.id).emit('translate', output);

}

});

});

**Zdrojový kód č.1**

### Řešení překladu dat

O překlad se stará framework Translate, který je jeden z balíčků npm. Tento framework ke své produkci potřebuje název a klíč překladače, který můžeme jednoduše nastavit přes funkci engine.

**const** translate = require('translate');

translate.engine = "google";

translate.key = "AIzaSyB-VncHpLDa71H0MysLT0LabfLV6\_NZjqE";

Zdrojový kód č.2

### Chyba při překladu

Při překládání se může vyskytnout chyba v podobě prázdného objektu, který vrací zpátky uživateli při dotazu na frontendu. Tato chyba se dá opravit pomocí synchronizace funkcí, kde se postupně čeká na ukončení předchozí funkce, aby program mohl dále pokračovat.

io.on('connection', (socket) => {

socket.on('translate', async (args) => {

**if** (args[2] !== '') {

**const** start = async **function** () {

**return** await translate(args[2], {from: args[0], to: args[1]});

}

**const** output = await start();

console.**log**(socket.id + "[" + args + "]: " + output);

io.to(socket.id).emit('translate', output);

}

});

});

Zdrojový kód č.3

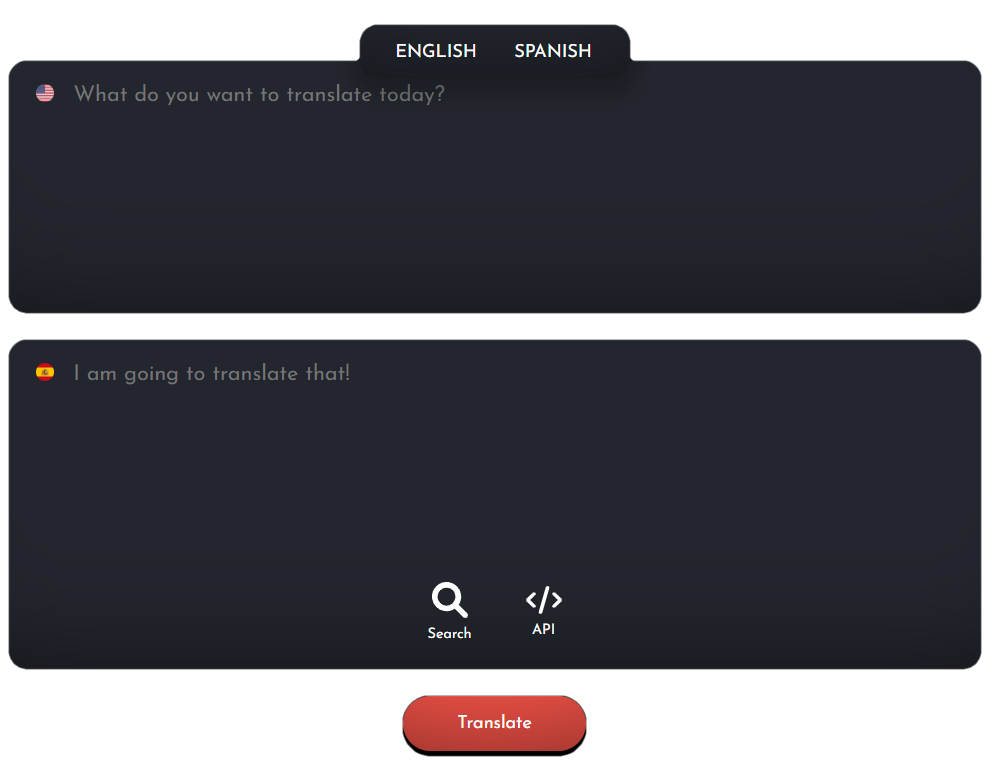
## Problematika frontendu

Při tvorbě stránky pro uživatele se využilo jednoduché HTML společně s CSS pro uživatelské rozhrání. Překladač disponuje funkcí vyhledání v Googlu Překladači a vyhledání použité Api na stránce npm. Aplikace je vytvořená responzivní čili by neměl být žádný problém na jakémkoliv zařízení.

### Design/Vzhled stránky

Design je řešen pomocí jednoduchého stylování CSS, které je responzivní pro všechna zařízení do šířky 260px. Tématika barev je moderně tmavá, jako většina dnešních populárních stránek. Stránka využívá font Josefin Sans z knihovny fontů od Google Fonts a ikony z knihovny ikon od Fontawesome.

Stránka disponuje dvěma textarea, které obě po stranách mají vlaječku, právě užívaného jazyka. První textarea vyžaduje text, který chce uživatel přeložit a druhý představuje přeložený text. Nad těmito textarea se vyskytuje select s překládanými jazyky, a naopak pod textarea se nachází tlačítko pro odeslání formuláře o přeložení (viz. Obrázek 1).



Obrázek 1: Stylování stránky

### Řešení odesílání dat

Aby se dotaz mohl poslat, využívá se již předem nainstalované knihovny Socket.io. Její implementace se skládá pouze ze skriptové reference a samostatné konstanty. Dále přes emity knihovna posílá formulář o překládaném textu a jazycích.

document.getElementById('form').addEventListener('submit', **function**(e) {

e.preventDefault(); // Zakazuje stránce po odeslání stránku aktualizovat

**let** lang1 = document.getElementById('lang1').value;

**let** lang2 = document.getElementById('lang2').value;

**let** input = document.getElementById('input').value;

socket.emit('translate', [lang1, lang2, input]);

});

Zdrojový kód č.4

### Řešení přijetí dat

Poté co server odešle odezvu překladu zpátky uživateli, je za potřebí aby se překlad ukázal u uživatele. To se řeší přes socket.on, který se spustí poté, co klient dostane určitý emit. Na to se u uživatele změní text v textarea překladu.

socket.on('translate', (args) => {

document.getElementById('output').value = args;

});

Zdrojový kód č.5

### Vlaječky pro jazyky

Aby program nějak vypadal, byli vytvořeny vlaječky pro jakýkoliv jazyk dostupný. Vlaječky byli vytvářeny v programu Paint.net, který usnadňuje práci s obrázky. Dále se na frontendu vlaječky upravují podle určité hodnoty v selectu.

### Užitečná tlačítka

Aby stránka nebyla prázdná, u textarea je možnost si vyhledat přeložený text na Google Překladači samotném. Jelikož Google Překladač využívá URL dotazu pro překlad, parametry se dají jednoduše doplnit. Tato funkce se dá využít, kdyby náhodou překlad nebyl na stránce úplný. Druhé tlačítko je odkaz na využitou knihovnu, kterou program využívá.

document.getElementById('button-search').onclick = () => {

window.open('https://translate.google.cz/?hl=en&sl=' +

document.getElementById('lang1').value +

'&tl=' +

document.getElementById('lang2').value +

'&text=' +

document.getElementById('input').value

, '\_blank');

}

Zdrojový kód č.6

# Závěr

Úkolem bylo vytvořit překladač, který by uměl přeložit z jakéhokoliv jazyka, který je na Google Api k dispozici. Aplikace běží na systému Node.js a jsou využity knihovny Express.js a Socket.io, včetně Translate, která běží na svém vlastním Google Api klíči a tím řeší většinu problémů. Aby překlad byl úplný a nedal se jednoduše zneužít, bylo za potřebí využití synchronních funkcí, kdy jedna funkce čeká na druhou, aby nedocházelo k předstihu a navracel se uživateli nedokončený překlad.

Na frontendu u uživatele je design řešen pomocí CSS, který je moderní a jednoduchý na pochopení. Při výrobě se využilo Google Fonts a vlaječky, vytvořené a upravené v editoru fotek Paint.net. HTML obsahuje čistě vytvořené elementy pomocí flex boxů a floutů. Připojení je řešeno přes Socket.io, které je implementováno jedou funkcí a referencí scriptu v HTML. Tato funkce také řídí emity, které uživatel přes formulář posílá serveru. Následně server pošle odpovědi s překladem, které se zpracovávají u uživatele na frontendu. Problém nastal u některých hodnot jazyků. Některé jazyky nefungují přes knihovnu Translate, tak museli být odebrány.

# Přílohy

## Zdroje

Google. *API's & Services* [online]. [cit. 2022-04-25]. Dostupné z: <https://console.cloud.google.com>

Google. *Cloud* [online]. [cit. 2022-05-01]. Dostupné z: https://cloud.google.com/translate/docs/languages

Translate. *Npm* [online]. [cit. 2022-04-25]. Dostupné z: <https://www.npmjs.com/package/translate>

Node.js. [online]. [cit. 2022-04-25]. Dostupné z: <https://nodejs.org/en/>

Fontawesome. *Icons* [online]. [cit. 2022-05-01]. Dostupné z: <https://fontawesome.com/icons>

Google. *Fonts* [online]. [cit. 2022-05-01]. Dostupné z: <https://fonts.google.com/specimen/Josefin+Sans?query=josefin>