

DELTA Střední škola informatiky a ekonomie, Základní škola a
Mateřská škola s.r.o. Ke Kamenci 151, PARDUBICE

Canpit

platforma pro online komunikaci určená
jazykovým skupinám

Jméno: Kulhánek Kryštof

Školní rok: 2022/2023

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

Konzultant: Bc. Vladka Janů

Zadání maturitního projektu z infromatických předmětů

Jméno a příjmení: *Kryštof Kulháněk*
Pro školní rok: *2022/2023*
Třída: *3. A*
Obor: *Informační technologie 18-20-M/01*

Téma práce: *Komunikační video platforma pro jazykové skupiny*
Vedoucí práce: *Bc. Vlad'ka Janů*

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:

Cílem práce je vytvořit webovou aplikaci umožňující online video komunikaci s cílem usnadnit plynulost skupinových konverzací v cizích jazycích. Krom samotného video hovoru bude aplikace obsahovat interaktivní aktivity, které mají zapojit účastníky do konverzace a zjednodušit tak průběh. Uživatelé mohou zakládat hovory, sdílet a přidávat ostatní. Spouštět a plánovat společné aktivity z omezeného výběru i v průběhu hovoru.

Většina částí aplikace bude napsána v jazyce JavaScript/TypeScript. Konkrétně bude použit framework React, backend poběží na populárním Node.js. O přenos videa se zvukem mezi uživateli se postará technologie webRTC podporovaná všemi moderními prohlížeči.

Stručný časový harmonogram (s daty a konkretizovanými úkoly):

- ZáříNávrh architektury, schématu databáze a designu aplikace
- Říjen.....Vývoj aplikace a zajištění video komunikace
- polovina Listopadu.....Vývoj a integrování interaktivních aktivit do aplikace
- ProsinecTestování, oprava chyb
- polovina Ledna.....Práce na teoretické části a vyladění aplikace
- začátek Února.....Nasazení aplikace, opravy chyb

Prohlašuji, že jsem maturitní projekt vypracoval samostatně, výhradně s použitím uvedené literatury.

V Pardubicích dne 30. března 2023

Kulhánek Kryštof

Poděkování

Rád bych poděkoval svému konzultantovi Bc. Vladěce Janů, za odborné vedení a konzultace při tvorbě tohoto maturitního projektu.

Abstrakt

Cílem projektu je vytvořit webovou video-komunikační platformu. Platformu, která usnadňuje průběh online komunikace v cizích jazycích, jakož to její primární použití. Vytvořit místo, které dává konverzaci strukturu, zapojí a zabaví všechny zúčastněné. O to se starají přímo integrované aktivity, které jsou jinak těžko proveditelné v rámci online prostoru.

Klíčová slova

Webová aplikace, Online komunikace, Aktivity, LiveKit, Next.js

Abstract

The aim of the project is to create a web-based video-communication platform. A platform smooth-out the flow of online communication in foreign languages, as its primary use case. Create a place that gives structure to the conversation, engages and entertains all involved. This is what directly integrated activities do that are otherwise difficult to implement within the online space.

Keywords

Web application; Online communication, Activities, LiveKit, Next.js

OBSAH

1	Pojmy	2
2	Úvod	3
3	Použité technologie	4
3.1	Technologie pro frontend	6
3.2	Technologie pro backend	7
4	Architektura systému	9
4.1	Aktivity	13
5	Uživatelské rozhraní	14
6	Branding a identita	20
7	Závěr	22

1 POJMY

Peer to Peer

Peer-to-Peer (P2P) síťový druh spojení, ve kterém se počítače propojují napřímo mezi sebou bez centrálního serveru nebo kontrolního uzlu. V P2P síti jsou počítače navzájem rovnocenné a mohou mezi sebou přímo komunikovat a sdílet zdroje.[1]

Open source

Open source je typ licence softwaru s otevřeným zdrojovým kódem. Otevřenost zde znamená jak technickou dostupnost kódu, tak legální. To znamená, že kdokoli může přistoupit k jeho zdrojovému kódu a upravit ho podle svých potřeb nebo ho použít pro své vlastní účely.[13]

Frontend X Backend

Frontend a backend jsou dva základní koncepty webového vývoje, které se liší v tom, na které části webové aplikace se zaměřují. Frontend se zaměřuje na vizuální a interaktivní části, zatímco backend se zaměřuje na uživatelsky neviditelné části aplikace, které jsou nezbytné pro správné fungování aplikace.

2 ÚVOD

Před rokem 2020 si většina lidí nedokázala přestavit komunikaci nebo práci zcela online na dálku. K tomu ale v prvním čtvrtletí byl donucen postupně celý svět. K mému překvapení se i státní instituce a hlavně školy za několik dní přesunuli kompletně na internet a výuka s některými omezeními pokračovala, nakonec několik dlouhých měsíců.

Omezení jsem si všiml i během výuky cizích jazyků. Setkávání skupin pro procvičení konverzace v cizím jazyce je přínosné a samotná konverzace klíčová při učení jazyku. V rámci těchto setkání nebo hodin lze samostatnou konverzaci strukturovat a obohatit o různé aktivity. Tyto aktivity pomáhají rozvíjet jazykové dovednosti, zabavit a zvyšovat sebevědomí účastníků v mluvení.

V online prostoru se ale setkání mohou stát mnohem chaotičtější a náročnější. Účastníci se mohou cítit méně motivováni k aktivní účasti a je tak těžší udržet pozornost.

Řešením se stala webová aplikace, která je veřejně přístupná a spustitelná na široké škále zařízení s webovým prohlížečem. Odbourá zmíněnou bariéru mezi osobním setkáním a online prostorem přímým integrováním aktivit. Vytvoří se tak prostor určený takovým skupinám.

3 POUŽITÉ TECHNOLOGIE

3.0.1 Figma

Figma je webová aplikace pro úpravu grafiky a návrh uživatelského rozhraní (UI). Je vhodná pro návrhy webových stránek, rozhraní nejen mobilních aplikací či prototypování. Figma také umožňuje týmovou spolupráci na projektech v reálném čase.

3.0.2 Typescript

TypeScript je programovací jazyk, který je rozšířením jazyka JavaScript o možnost statického typování, což zlehčuje průběhu vývoje, opravy chyb a zvyšuje bezpečnost kódu. TypeScript kód je pro spuštění nutné následně transpilovat neboli přeložit do standardního JavaScriptu.[4]

3.0.3 LiveKit

LiveKit je nový open source projekt pro vytváření interaktivních video konferencí a streamování v reálném čase. Využívá moderního end-to-end WebRTC stacku, který je podporovaný mnoha prohlížeči.

Jednou z hlavních vlastností LiveKitu je jeho schopnost dynamicky přizpůsobovat kvalitu videa na základě aktuálních podmínek sítě a zařízení účastníka. To znamená, že účastníci s různými internetovými připojeními a zařízeními mohou být připojeni do stejné konference a stále získat optimální zážitek z videokonference[11]

Serverová instance je distribuovaná jako Docker container image[12] pro vyšší kompatibilitu a jednodušší nasazení.

3.0.4 YARN

YARN je nástroj pro správu balíčků (knihoven) v JavaScriptových projektech. Umožňuje efektivní instalaci balíčků a současně provádí jejich kontrolu. Minimalizuje se tím riziko instalace poškozeného balíčku, což by mohlo poškodit kód aplikace. Pomocí paralelního zpracování, to znamená instalaci několika současně, dosahuje vyšších rychlostí. YARN také zajišťuje instalaci stejných verzí balíčků v celém projektu za pomoci konfiguračního souboru. Je kompatibilní s největším registrem komunitních balíčků NPM.[2] [3]

3.0.5 Cloudflare tunnel

Cloudflare Tunnel je služba společnosti Cloudflare, která umožňuje vývojářům připojit své lokální aplikace k Cloudflare bez nutnosti jejich nasazení na veřejný server. Tím umožňuje vývojářům vyvíjet a testovat své aplikace v bezpečném a soukromém prostředí, aniž by museli řešit náročné úkoly spojené s nasazováním na veřejný server. Vytvoří tedy veřejnou https url, přes kterou se lze připojit k localhost adrese na vývojářském prostředí. Pro testování funkčnosti přenosu videa a audia z různých zařízení velice vhodný nástroj.[7]

3.0.6 Git

Git je jedním z nejpobulárnějších systémů pro správu souborových verzí. Používá se pro sledování změn v kódu a koordinaci práce na projektech mezi různými vývojáři. Umožňuje jednoduchou správu změn, práci s větvemi, sloučení změn a řešení konfliktů.[14]

3.1 TECHNOLOGIE PRO FRONTEND

3.1.1 React

React je JavaScriptová knihovna, která se používá na frontendu webových aplikací. React slouží k tvorbě uživatelských rozhraní založených na komponentách, což jsou opakovaně použitelné bloky kódu, které zamezují duplikaci kódu a umožňují snadnou údržbu aplikace. Jednou z hlavních funkcí Reactu je správa stavu (state management) a následné vykreslování DOMu, neboli objektově orientovaná reprezentace webové stránky, což usnadňuje její modifikaci.[17]

3.1.2 Next.js

Next.js je open source JavaScript framework určený pro tvorbu webových aplikací s použitím Reactu. Jeho hlavní předností je, že poskytuje server side rendering (SSR), což znamená, že stránky jsou generovány na serveru a poté přeneseny klientskému prohlížeči. Tento přístup zlepšuje uživatelskou zkušenost, protože uživatelé vidí obsah ihned po načtení stránky, což zvyšuje rychlost a snižuje dobu načítání. SSR také umožňuje zlepšení SEO, protože vyhledávače mohou lépe indexovat obsah stránek. Next.js také obsahuje file system routing, což umožňuje snadnou organizaci struktury aplikace pomocí jednoduchého souborového systému. To zlepšuje přehlednost a usnadňuje tvorbu složitých aplikací.[16]

3.1.3 Styled components

Styled-components je knihovna pro React, která umožňuje vytvářet styly pro komponenty přímo v kódu za pomoci JavaScriptové a CSS syntaxe. Díky tomu můžeme používat proměnné a cykly pro generování různých stylů. Také nabízí CSS preprocessor jako SASS a jednoduchou údržbu všech stylů.

3.1.4 Radix

Radix je knihovna komponent pro React, která poskytuje sadu předdefinovaných, bezstylových a univerzálních UI prvků. Knihovna obsahuje optimalizované a odzkoušené prvky jako například tlačítka, formulářová pole, navigační menu, modální okna, dialogy, atd. Urychluje tak vývoj vlastních komponent.[6]

3.2 TECHNOLOGIE PRO BACKEND

3.2.1 Supabase

Supabase je open-source platforma, která poskytuje backend jako službu pro webové aplikace. Supabase umožňuje vývojářům vytvářet a spravovat databáze, autentizaci a API pomocí webového rozhraní. Také nabízí funkce, jako je autentizaci uživatelů, tvorbu databázových skriptů, zálohování, to vše umožňuje vývojářům vytvářet aplikace rychleji.[9] Další funkcí je přenos dat v reálném čase, který v projektu využívám.

3.2.2 Node.js

Node.js je serverové prostředí, které umožňuje vývojářům psát a spouštět konzolové aplikace v jazyce JavaScript mimo prohlížeč. [10]

3.2.3 Express.js

Express.js je framework pro Node.js, který umožňuje snadněji vytvářet a spravovat webové RESTful API aplikace. Nabízí širokou škálu funkcí, včetně správy routování, middlewaru a integrace s databázemi.[8]

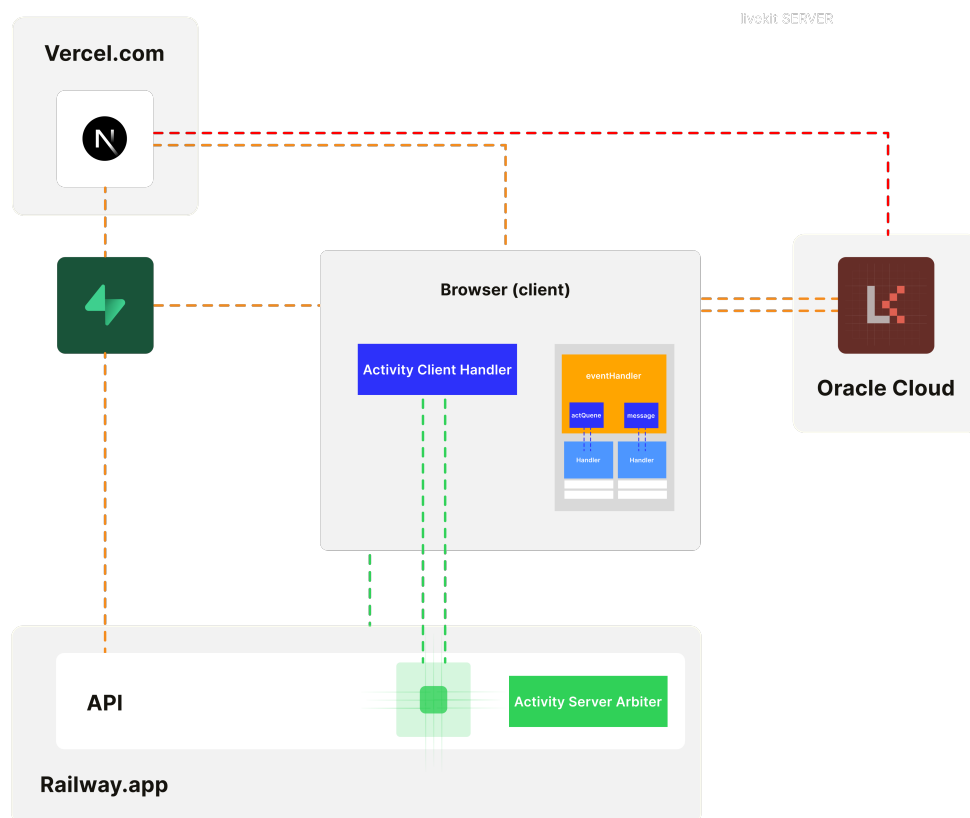
3.2.4 Docker

Docker je open-source platforma, která umožňuje virtualizaci na úrovni operačního systému a běh softwaru v kontejnerech. Kontejnery poskytují izolaci aplikace od zbytku

operačního systému. Umožňují snadné a rychlé nasazení aplikace bez nutnosti instalace všech závislostí na hostitelském operačním systému. Docker podporuje všechny hlavní operační systémy a je kompatibilní s většinou cloudových platforem.

Podporuje automatizované procesy nasazení, testování a škálování aplikace. V kombinaci s nástroji pro správu kontejnerů může Docker výrazně zjednodušit vývoj a nasazení aplikací a snížit náklady na infrastrukturu.[15]

4 ARCHITEKTURA SYSTÉMU



Obrázek 4.1: Vizualizace architektury a propojení služeb

Celá architektura je rozdělena na čtyři části zajišťující určité funkce.

- **Web application** tvořena pomocí knihovny React, přesněji je použit framework Next.js. Obsahuje frontend a některé backendové funkcionality.
- **Supabase** sloužící jako databáze, autorizační poskytovatel a centrální místo pro přenos živých dat.
- **Media server** zajišťuje propojení klientů a přenos videa se zvukem
- **Activity API** obsluhuje a drží logiku aktivit

4.0.1 Autentizace (registrace a přihlášení)

Bezpečnost aplikací je závislá na autentizaci. Autentizace umožňuje uživatelům přihlásit se pouze k účtům, které jim patří nebo mají přístup. Tato funkce je zajištěna registrací a ověřením totožnosti uživatele kombinací emailu a hesla nebo OAuth poskytovatele.

V mém řešení používám autentizační API, která je součástí Supabase. Oproti přihlašování při registraci je potřeba mimo emailu a hesla také zvolit unikátní uživatelské jméno. Po registraci a úspěšném přihlášení se do session prohlížeče přidá záznam s JWT tokenem obsahující potřebné informace pro udržení uživatele přihlášeným v prohlížeči na určitý čas.

Obnova hesla probíhá prostřednictvím emailové adresy, kdy se uživateli odešle speciální link na stránku s možností obnovy.

4.0.2 Úprava databáze a tvorba tabulek

Samotná připravená tabulka uživatelů obsahuje několik hodnot, ale nepočítá s uživatelským jménem. Proto bylo potřeba přidat nový sloupec „username“ typu VARCHAR s povinnou unikátností.

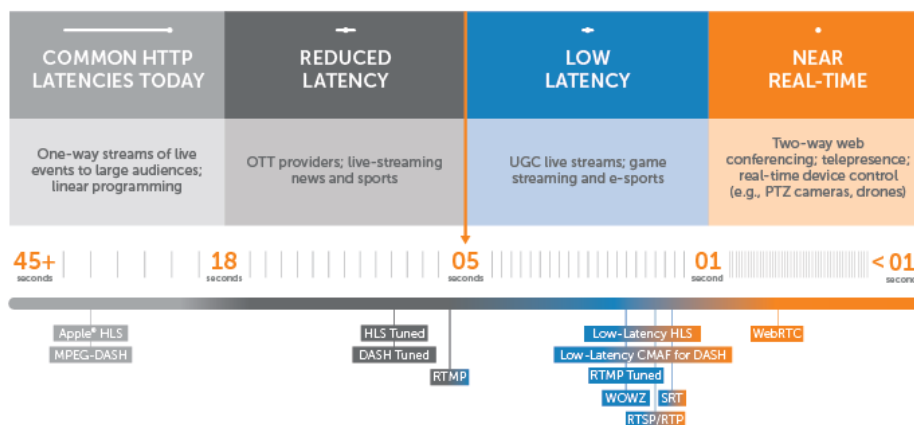
Pro uživatelská nastavení byla vytvořena speciální tabulka „settings“, která má one-to-one relaci se zmíněnou tabulkou ze schématu auth users. Drží celé jméno uživatele a některá nastavení ukládána ve formátu JSON pro jednodušší manipulaci.

Tabulka pro místnosti „rooms“ drží vygenerované unikátní id každé místnosti, čas a id uživatele. I když jsou id místností používány jednorázově, je potřeba vždy zaručit jejich unikátnost. Mohlo by se totiž stát, že přes starý link s daným id půjde připojit do nové místnosti někoho úplně jiného. S touto tabulkou může pracovat pouze server s nikoliv na přímo client.

4.0.3 Přenos videa a zvuku

Pro fungování celého projektu bylo klíčové zajistit přenos videa a zvuku. Už při prvním průzkumů možností vzešla technologie WebRTC. WebRTC (Web Real-Time Communi-

ation) je open-source protokol pro komunikaci v reálném čase mezi webovými prohlížeči. Podporu najdeme u všech moderních webových prohlížečů. Tento protokol se ujal kvůli velice nízké latenci (viz obrázek 4.2), podpoře a adaptaci na síťové podmínky.



Obrázek 4.2: Srovnání latence WebRTC a ostatních protokolů

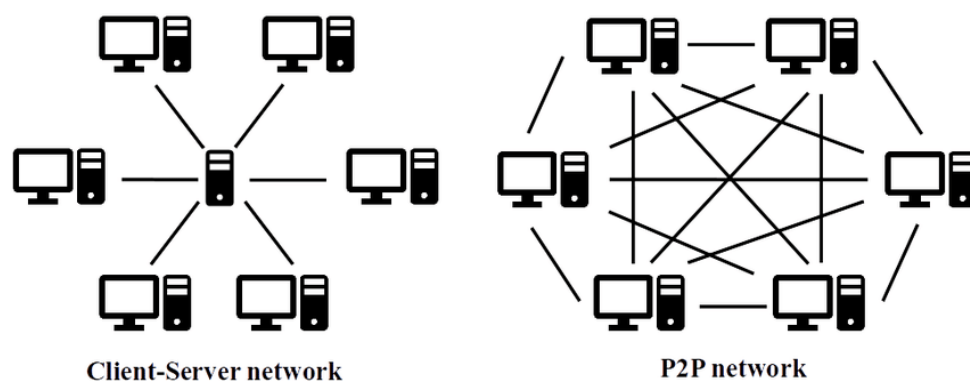
Umožňuje přenos zvuku, videa a dat přímo mezi dvěma prohlížeči bez nutnosti prostředníka, tedy serveru. Toto vedlo k návrhu přenosu dat právě za pomoci peer-to-peer. Toto bylo v rámci malé demo verze otestováno. Výsledek byl funkční a k tomu má tento koncept několik výhod oproti běžné praxi, výčtem.

- Vyšší bezpečnost přenášených dat, jelikož nejdu přes žádného prostředníka.
- Odpadá finanční zátěž a správa centrálních serverů s vysokými nároky nejen na propustnost v síti a blízkost k uživateli.
- Jednodušší vývoj aplikace, jelikož není zapotřebí robustní backend.

Toto řešení má ale své velké limity v podobě nízkého počtu uživatelů v hovoru. Každý klient totiž musí každému posílat výstup z kamery a mikrofону (viz obrázek 4.3). To přenáší výpočetní zátěž na klienta a je zapotřebí vysoká rychlost uploadu při připojení k internetu, která není běžně k dispozici. S každým připojeným se úměrně náročnost připojení zvyšuje.

Proto bylo nakonec zvoleno centralizované řešení, které ve výsledku přináší pro uživatele lepší službu. Zmíněný problém rychlosti uploadu na straně klienta odpadá a místnosti

pojmu až desítky uživatelů.



Obrázek 4.3: Síťové topologie

4.0.4 Inicializace a připojení

Před možným připojením k media serveru pro přenos dat je potřeba obdržet autorizační token. Na základě něho jsou udělena práva a zajišťuje, že se klient připojí do správné místnosti s přístupem.

Ke generaci tokenů slouží POST endpoint `/api/room`. Funkcionalita je rozdělena na dva scénáře.

- Připojení - Pomocí vstupních parametrů id místnosti a uživatele vrací potřebný token pouze v případě, že taková místnost v databázi existuje.
- Vytvoření místnosti - Vyžaduje id uživatele, vytvoří místnost s novým unikátním id v databázi a rovnou vrací uživateli token k připojení.

Povolení přístupů ke kameře a mikrofonu

Přístup k mikrofonu a kameře ve webovém prohlížeči je možné pouze tehdy, pokud uživatel udělí příslušná oprávnění. Tyto oprávnění musí být udělena při prvním použití kamer a mikrofonu jak v daném prohlížeči, tak následně i na jednotlivých webech.

Dnešní moderní prohlížeče vyžadují, aby webové stránky používající kameru a mikrofon běžely na HTTPS protokolu. Toto opatření je důležité pro ochranu soukromí a bezpečnosti uživatelů, a to zejména při přenosu citlivých dat, jako jsou zvukové a obrazové záznamy.

Stopy videa a zvuku

Jednotlivé media data streamy jsou následně za pomoci html elementu video a audio zprostředkované uživateli v prohlížeči.

4.1 AKTIVITY

Aktivitami jsou myšleny dílčí řekněme mini hry různého typu. Příkladem za všechny může být „Hádej kdo jsem“, se kterou se určitě každý někdy setkal.

Z pohledu vývoje jsou aktivity rozděleny na dvě části.

- **Interface** neboli část aktivity na straně klienta, která se stará o vstupy a výstupy v různých podobách.
- **Arbiter** je jádrem každé aktivity, obsahuje logiku a potřebné služby. Tato část je tedy řídicím prvkem a vždy drží pravdivý stav.

Interface

Interface se stará o veškeré UI aktivity na clientské straně. Například manipuluje s video prvky v místnosti a rozložení stránky.

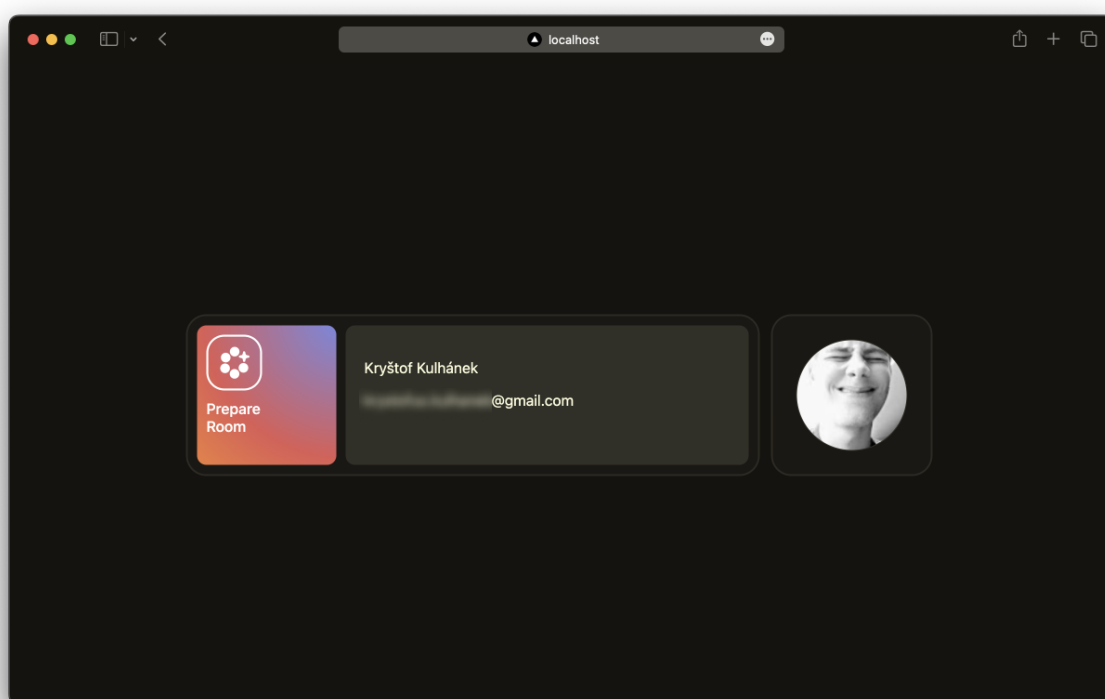
Arbiter

Část nazvaná Arbiter je navenek jednoduchý API endpoint, který přijímá parametry o spuštění aktivity a z jaké místnosti požadavek přichází. Podle toho se Arbiter připojí do stejné místnosti aby mohl poslouchat a odesílat příkazy na Interface. Na základě definované logiky aktivity poté celý průběh řídí.

V případě připojení nového uživatele do místnosti během probíhající aktivity není zapojen, až poté co se aktivita spustí znovu.

5 UŽIVATELSKÉ ROZHRAŇÍ

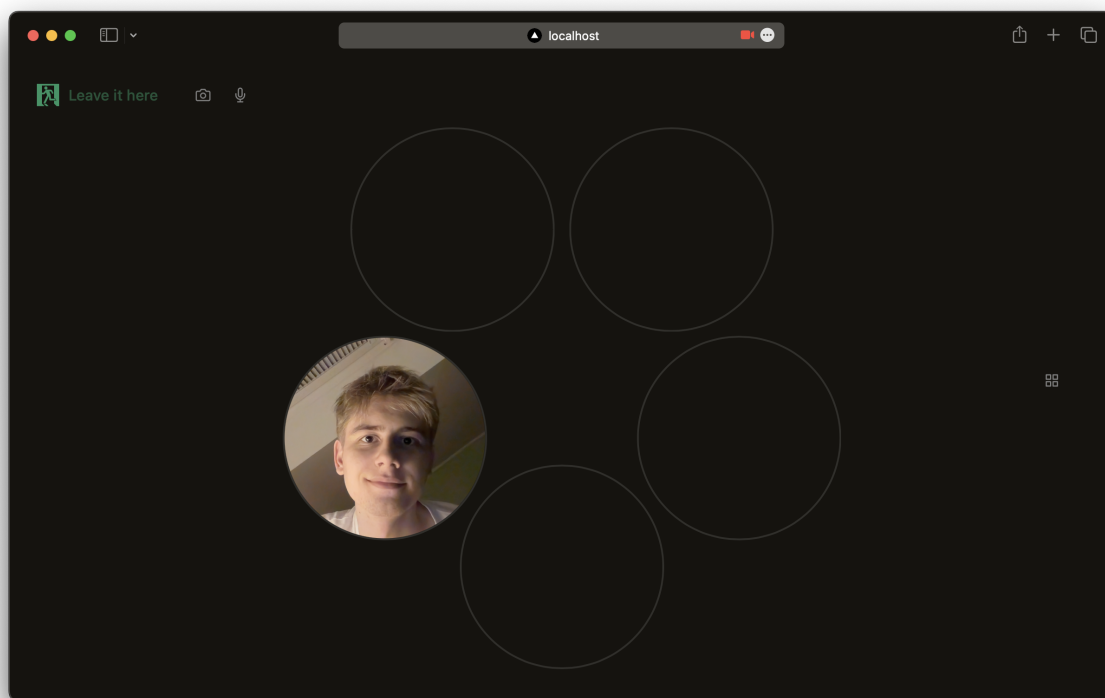
Po registraci a přihlášení se uživatel dostane na domovskou stránku (viz obrázek 5.1). Odtud může měnit nějaká nastavení a v první řadě zahájit neboli vytvořit místnost.



Obrázek 5.1: Domovská stránka uživatele

Po zavolání API endpointu a získání tokenu je uživatel přesměrován na `/room/[id-místnosti]`. Po uživatelově souhlasu použít kameru a mikrofonu se client připojí k media serveru a uživatel může vidět svoji kameru jako na obrázku 5.2. V místnosti je díky talčítčkům

Aby se mohli připojit ostatní uživatelé je zapotřebí sdílet odkaz místnosti. Pokud ho obdrží uživatel, který není přihlášený musí se před připojením do místnosti přihlásit či registrovat. Poté je průběh stejný a po úspěšném připojení bude místnost vypadat jako na obrázku 5.3



Obrázek 5.2: Uživatel v místnosti čekající na připojení ostatních

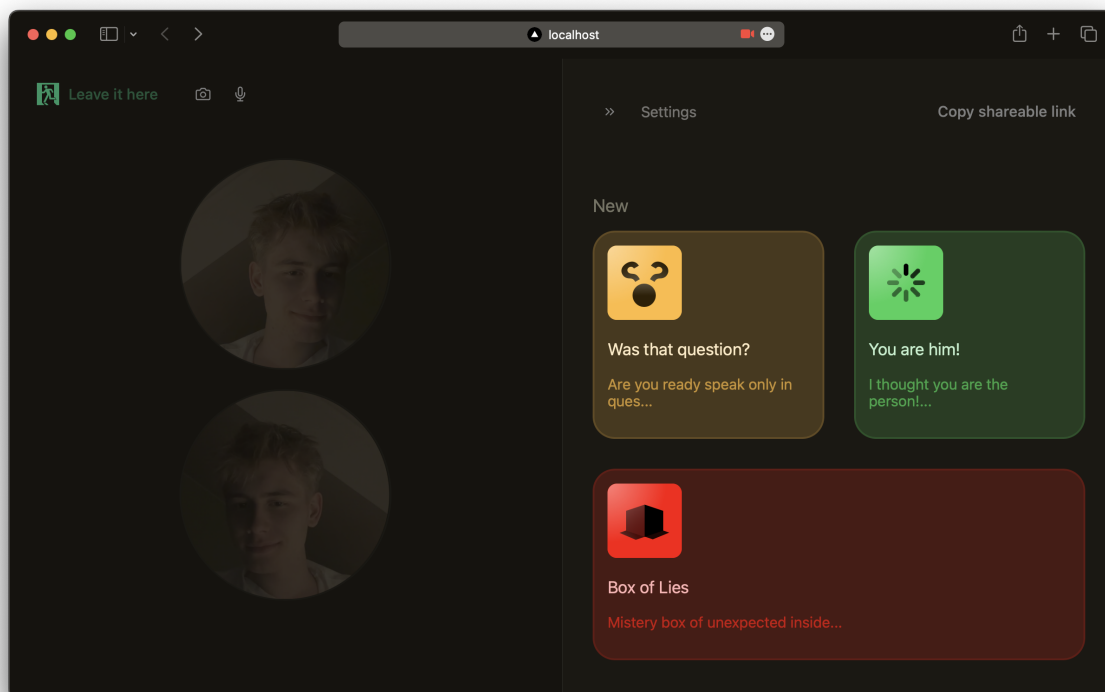
Po kliknutí na ikonku ležící na pravé straně místnosti se otevře knihovna aktivit (obrázek 5.4). Zde se nacházejí veškeré aktivity, které lze spustit.

Aktivitu spustíme kliknutím na ikonku. V tu chvíli se inicializuje u všech uživatelů (obrázek 5.5). Zde má každá aktivita úvodní stranu a v tomto případě jen čeká na potvrzení účasti od všech uživatelů, před jejím zahájením.

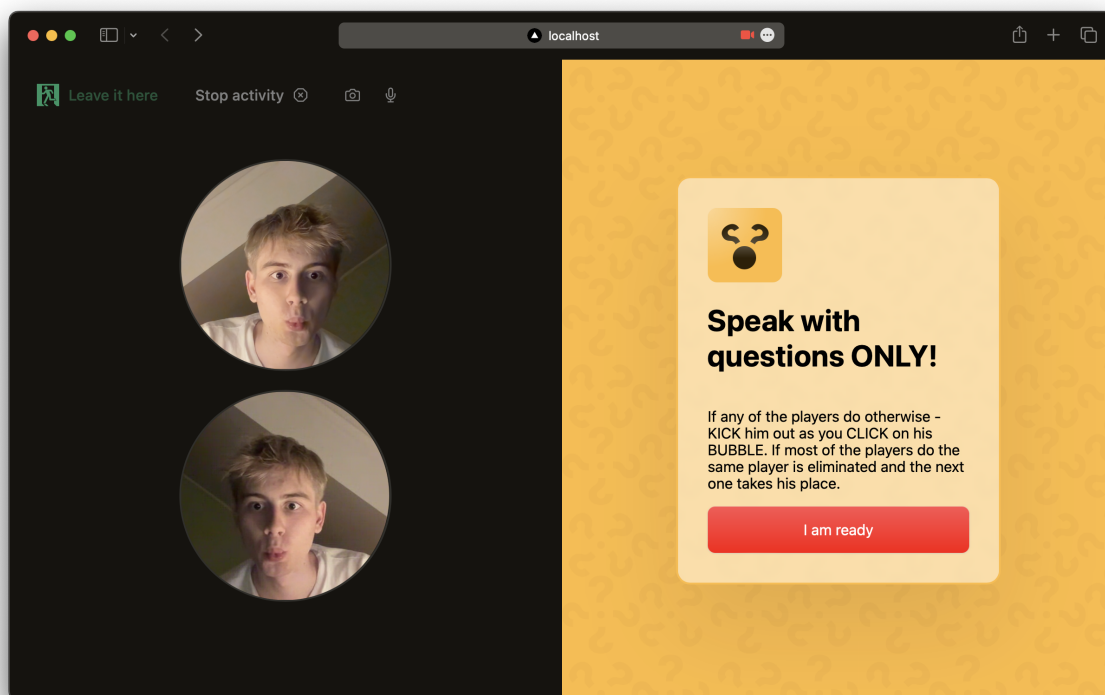
Na obrázku 5.6 je už aktivita spuštěna a zábava může začít.



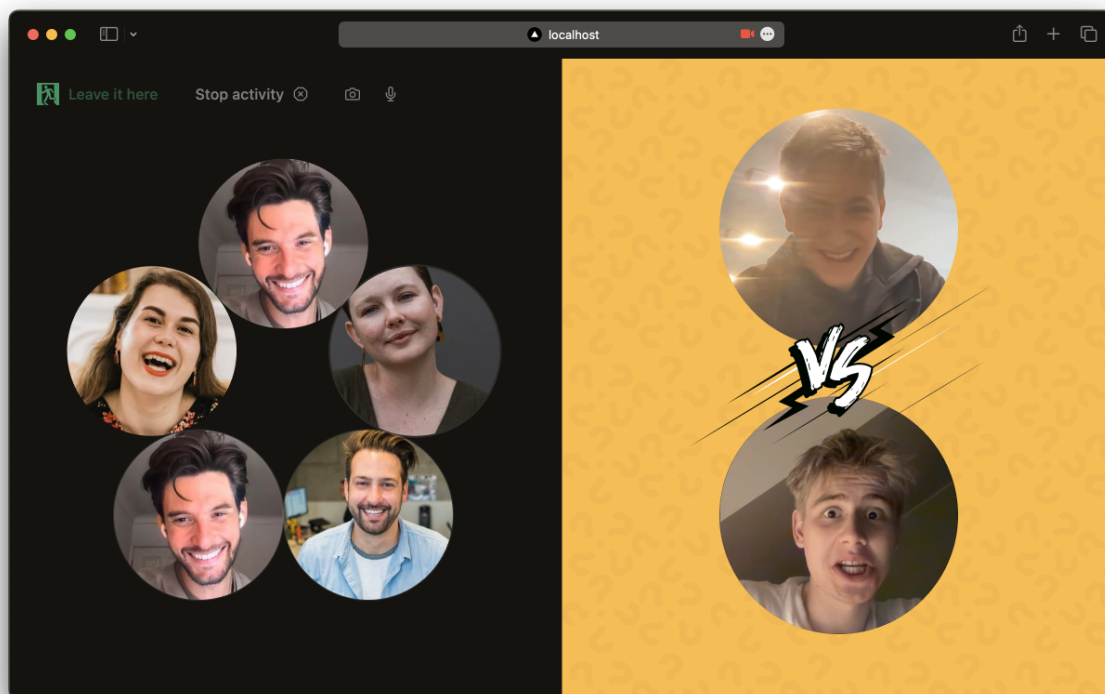
Obrázek 5.3: Místnost, ve které jsou připojení dva uživatelé



Obrázek 5.4: Knihovna aktivit



Obrázek 5.5: Aktivita „Mluv jen otázkou!“ před spuštěním



Obrázek 5.6: Běžící aktivita

5.0.1 Dynamické rozmístění videí do kruhu

Pro rozmístění jednotlivých účastníků hovoru byl zvolen netypický kruh. Má to své nevýhody v podobě nevyužití maximálně celého prostoru. Ale v kruhovém rozmístění si zúčastnění jsou více navzájem rovni a toto rozložení tvoří ikonický bod designu.

Docílení požadovaného layoutu znázorněném na obrázku 5.7 je za pomoci CSS vlastností a matematického vzorce zpracovaném do vlastního React Hook.

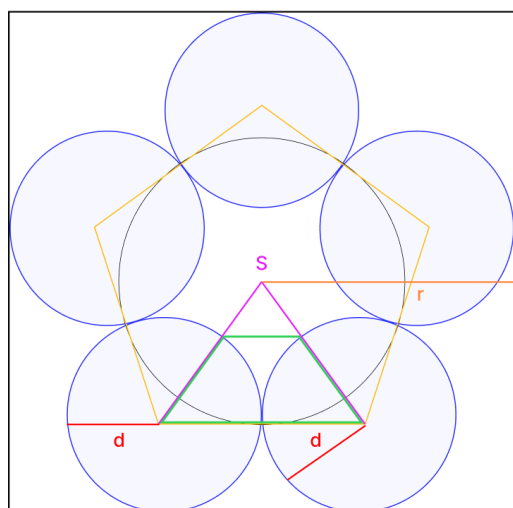
K správnému umístění jsou zapotřebí tři hodnoty. Průměr (d) jednotlivých prvků, úhel (z) od středu (S) mezi každým prvkem a vzdálenost (x) prvku od středu (S). Hodnoty (d) a (x) jsou stejné pro všechny, u každého se mění jen úhel (z).

Požadovaná velikost průměru prvku musí splňovat, aby byli se všechny vešli uvnitř vyhrazeného prostoru (čtverce). Zároveň byli maximálně velké, aby se nepřekrývaly. K tomu slouží následující vzorec kde (n) je počet prvků.

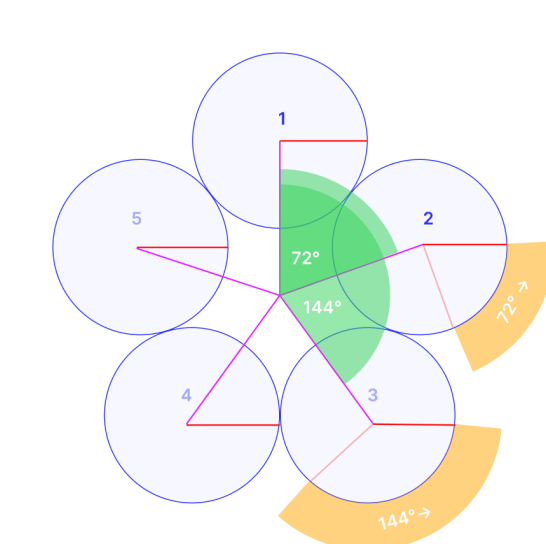
$$d = \frac{r \sin\left(\frac{\pi}{n}\right)}{\sin\left(\frac{\pi}{n}\right) + 1} \quad (5.1)$$

Hodnotu (x) dostaneme rozdílem poloměru (r) a poloměru (d)

Úhel (z) bude vždy násobek pořadí prvku s hodnotou 360 stupňů děleno počtem prvků (n).



(a) Pomocná vizualizace k výpočtu



(b) Znázornění CSS transformace

Obrázek 5.7: Pomocná grafická vizualizace rozmístění videí

Po provedených výpočtech je jednotlivým kulatým prvkům s videem nejprve nastavena stejná výška a šířka. Jelikož jsou kruhové, tak přesněji jejich průměr. Všechny jsou do čtvercového div prvku zarovnány na střed. Poté pomocí CSS vlastnosti transform, který má funkci rotate a translate, umístíme zadáním vypočítaných hodnot prvky na správné místo. U druhého prvku na obrázku 5.7(b) by to mohlo vypadat takto.

```
transform: rotate(72deg) translate(200px) rotate(72deg)
```

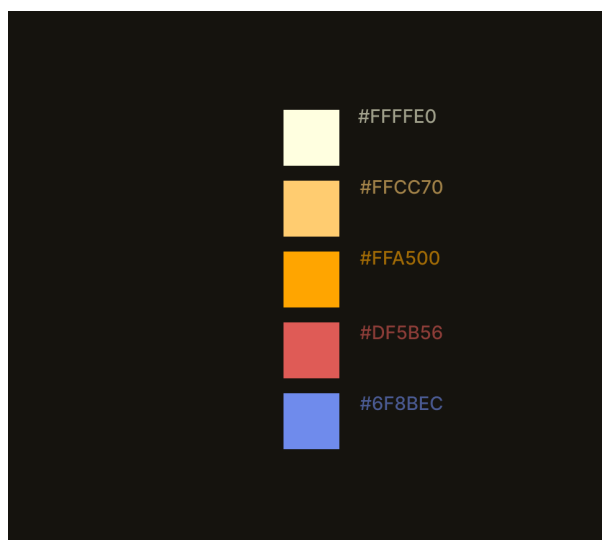
Rozmístění je ještě obohaceno o animace. Při změně velikosti okna a hlavně při změně počtu účastníků hovoru se rozmístění dynamicky mění.

6 BRANDING A IDENTITA

Identita pomáhá produktu či službě stát se snadno rozpoznatelnou, odlišit se a navázat vztah s uživateli. Ve výsledku je to něco, co v saturovaném trhu dává konkurenční výhodu a klíčovou vlastností je nemožnost zkopírování či krádeže.

6.0.1 Filozofie

Jelikož se projekt motá kolem komunikace a hravých aktivit, identita vychází ze známé dětské aktivity. Tím je telefonování za použití špagátu nataženém mezi dvěma plechovkami. To představuje čistou dětskou hravost, komunikaci, propojení a společné přátelství. Další myšlenkou nebo námětem je táborák. Teplé místo do tvaru kruhu, kde se často odehrávají hluboké konverzace a je zábava. Podle toho je laděná paleta barev.



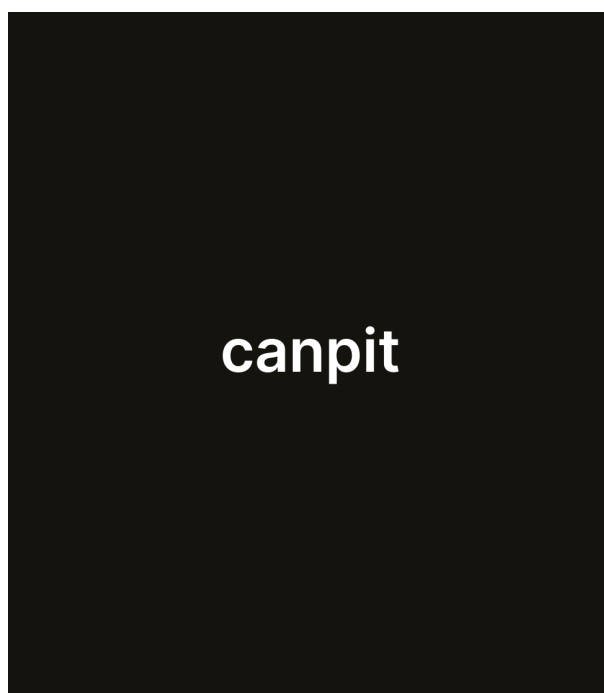
Obrázek 6.1: Nalevo vizualizace a vpravo zvolená paleta barev

6.0.2 Název

Název je nedílnou součástí jakéhokoliv projektu a tvoří jádro identity. Měl by být úderný, dobře znějící a zapamatovatelný. Po nespočtu interakcí se slovy vyhrálo „Canpit“. To představuje spojení anglického slova „Can“ a „pit“. Slovo „can“ je odkazem na zmíněnou aktivitu, v překladu plechovka. Slovo „pit“ v překladu znamená jáma či díra, jakož to táborák, kulaté místo se středem pozornosti.

6.0.3 Logo

Logo vizualizuje propojení plechovek a naznačuje spojení mezi sebou.



Obrázek 6.2: Logo s názvem

7 ZÁVĚR

Projekt se úspěšně povedlo dovést do pilotní funkční verze. Hlavní komponenty jsou funkční. Tedy zajištění přenosu videa a zvuku, je vytvořena architektura nebo mechanismus pro vytváření a spouštění aktivit. Také prostředí, které tyto prvky spojuje do jednoho celku a v uživatelsky známé formě nabízí potenciálním uživatelům. Ti se mohou již registrovat nebo přihlásit, vytvářet místnosti, do kterých se přes vygenerovaný odkaz mohou ostatní připojit. Vybírat a spouštět aktivity.

Možný budoucí problém může nastat ve fázi širšího použití, kdy stávající server nebude takový objem uživatelů zvládat. S tím souvisí zpeněžení, jelikož potřebné zdroje na přenos videa jsou náročnější a s tím samozřejmě přichází vyšší cena za provoz. Tedy bez finančního příjmu nebo dotace z jiných zdrojů není možné platformu provozovat.

Díky architektuře aplikace je možné v budoucnu snadno přidávat další aktivity, což zvyšuje flexibilitu a udržitelnost. V další části projektu plánuji otevřít vývoj aktivit i pro ostatní vývojáře.

Platforma je připravena na plnohodnotné testování s několika desítkami reálných uživatelů v jeden okamžik a ověření celého konceptu. Zpřístupněna je na webové adrese <https://canpit.com/>.

SEZNAM OBRÁZKŮ

4.1	Vizualizace architektury a propojení služeb	9
4.2	Srovnání latence WebRTC a ostatních protokolů	11
4.3	Síťové topologie	12
5.1	Domovská stránka uživatele	14
5.2	Uživatel v místnosti čekající na připojení ostatních	15
5.3	Místnost, ve které jsou připojení dva uživatelé	16
5.4	Knihovna aktivit	16
5.5	Aktivita „Mluv jen otázkou!“ před spuštěním	17
5.6	Běžící aktivita	17
5.7	Pomocná grafická vizualizace rozmístění videí	18
6.1	Nalevo vizualizace a vpravo zvolená paleta barev	20
6.2	Logo s názvem	21

LITERATURA

- [1] *What is Peer-to-Peer Network, and How Does It Work?* [online]. [cit. 2022-03-6]. Dostupné z: <https://www.blockchain-council.org/blockchain/peer-to-peer-network/>
- [2] *Yarn: A new package manager for JavaScript* [online]. [cit. 2022-03-6]. Dostupné z: <https://engineering.fb.com/2016/10/11/web/yarn-a-new-package-manager-for-javascript/>
- [3] *npm vs Yarn - Which Package Manager Should You Use?* [online]. [cit. 2022-03-6]. Dostupné z: <https://www.keycdn.com/blog/npm-vs-yarn>
- [4] *TypeScript for the New Programmer* [online]. [cit. 2022-03-6]. Dostupné z: <https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>
- [5] *What Is WebRTC?* [online]. [cit. 2022-03-27]. Dostupné z: <https://www.wowza.com/blog/what-is-webrtc>
- [6] *Building a design system with Radix* [online]. [cit. 2022-03-27]. Dostupné z: <https://blog.logrocket.com/building-design-system-radix/>
- [7] *Cloudflare Tunnel* [online]. [cit. 2022-03-27]. Dostupné z: <https://www.cloudflare.com/products/tunnel/>
- [8] *Express* [online]. [cit. 2022-03-27]. Dostupné z: <https://expressjs.com/>
- [9] *Supabase* [online]. [cit. 2022-03-27]. Dostupné z: <https://supabase.com/>
- [10] *About Node.js* [online]. [cit. 2022-03-27]. Dostupné z: <https://nodejs.org/en/about>
- [11] *LiveKit Documentation* [online]. [cit. 2022-03-27]. Dostupné z: <https://docs.livekit.io/>

-
- [12] *Use containers to Build, Share and Run your applications* [online]. [cit. 2022-03-27]. Dostupné z: <https://www.docker.com/resources/what-container/>
- [13] *What is open source?* [online]. [cit. 2022-03-27]. Dostupné z: <https://www.redhat.com/en/topics/open-source/what-is-open-source>
- [14] *What is Git* [online]. [cit. 2022-03-27]. Dostupné z: <https://www.atlassian.com/git/tutorials/what-is-git>
- [15] *Docker overview* [online]. [cit. 2022-03-27]. Dostupné z: <https://docs.docker.com/get-started/overview/>
- [16] *Next.js for CTOs. What is Next.js used for and why we love it* [online]. [cit. 2022-03-27]. Dostupné z: <https://tsh.io/blog/what-is-next-js-used-for/>
- [17] *What Is React and How Does It Actually Work?* [online]. [cit. 2022-03-27]. Dostupné z: <https://www.hostinger.com/tutorials/what-is-react>