



STŘEDNÍ PRŮMYSLOVÁ ŠKOLA
OBCHODNÍ AKADEMIE
JAZYKOVÁ ŠKOLA
FRÝDEK-MÍSTEK



FOTOGRAFICKÝ WEB A JEHO ADMINISTRACNÍ SYSTÉM

JIŘÍ VALA

soutěž KYBER STOČ 2022

Třída: **4.IT**

Školní rok: **2021/22**

Vymezení pojmů

Backend [server]: Server, který zpracovává HTTP požadavky a dále s nimi pracuje.

Jedná v podstatě o *mozek* aplikace/webu

HTTP: Hypertext Transfer Protocol, nezabezpečený protokol určený pro komunikaci s WWW servery. Zabezpečená alternativa je **HTTPS**, který k zabezpečení používá SSL certifikáty.

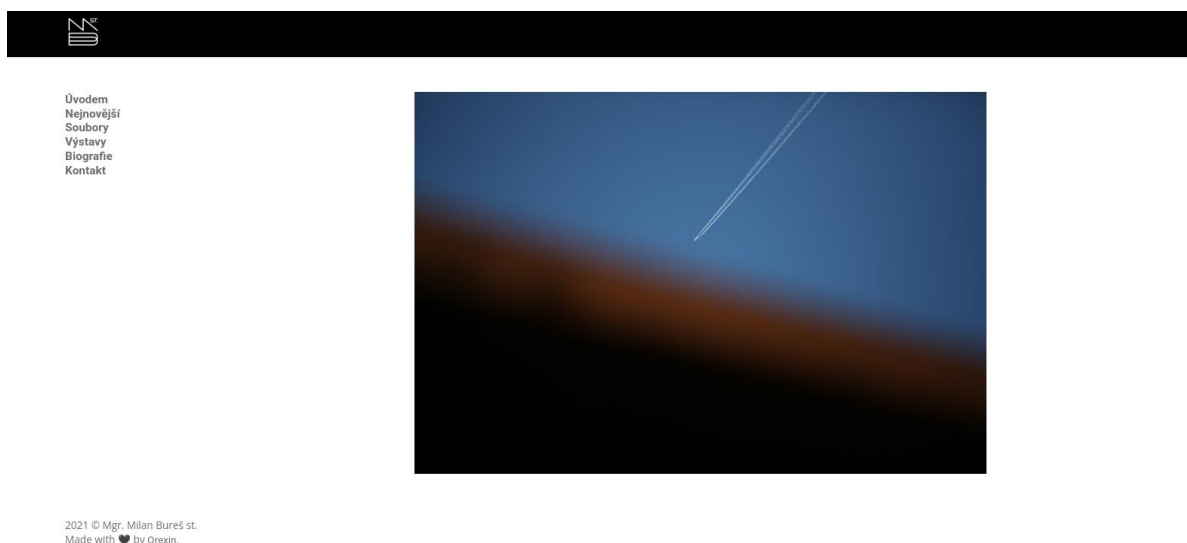
React.js: Javascript framework určený k vytváření uživatelských prostředí

Cíl práce

Pan Milan (dále jen "uživatel", "fotograf", nebo "zákazník") Bureš je amatérský fotograf, jehož fotografie zachycují náhodné okamžiky jeho života, a proto se již před dlouhou dobou rozhodl, že chce, aby se na ně mohli podívat i jiní lidé. Nechal si tedy na zakázku vytvořit web, který by sloužil jako jakási virtuální galerie jeho fotografií. Avšak postupem času začaly být technologie, které byly použity pro vývoj jeho webu, zastaralé a byly nahrazeny modernějšími.

A to přesně je mým úkolem – navrhnout a naprogramovat kompletní web s galerií a administračním systémem, který by umožňoval velmi jednoduché a intuitivní manipulování s fotografiemi a alby, do kterých by uživatel fotografie později rozdělával, a to vše za použití těch nejmodernějších a nejrozšířenějších technologií, které dnes existují. Nakonec jsem měl možnost tento projekt spojit s dlouhodobou maturitní prací, takže *smetu dvě mouchy jednou ranou*.

Webová stránka



Obr.1 - Ukázka webové stránky

Webová stránka obsahuje nejen výše zmíněnou galerii, která krom toho, že zobrazuje název a popis fotky, tak obsahuje i funkci *lajků*, takže pokud se návštěvníkovi daná fotka líbí, může přidat *lajk* a uživatel tak přesně uvidí, která fotka má kolik *lajků* a která z nich je nejoblíbenější.

Data, které web používá k svému fungování nejsou dynamická, jelikož by to z principu fungování statické webové stránky nemělo smysl. Data se stahují z databáze ve chvíli, kdy uživatel přijde na stránku poprvé. To znamená, že pokud uživatel nahraje novou fotografii, změny na webu se projeví až ve chvíli, kdy uživatel stránku znovu načte.

Web obsahuje podstránky jako je *Úvodem*, kde je úvodní slovo a přivítání nových návštěvníků webu. *Biografie* stručně popisuje dosavadní život zákazníka, *Nejnovější* obsahuje seznam fotek, které zákazník označil jako nejnovější, *Soubory*, po rozkliknutí zobrazí seznam všech aktuálních a dostupných alb. Po kliknutí na jedno z nich se objeví galerie s fotografiemi z daného alba. *Výstavy*, jak již název napovídá skrývají seznam všech výstav, na kterých zákazník kdy byl. *Kontakt* obsahuje kompletní kontaktní formulář, který zatím není funkční (nemám backend server, na který bych mohl odesílat požadavky z formuláře a pak je zpracovávat), a odkazy na jeho sociální sítě.

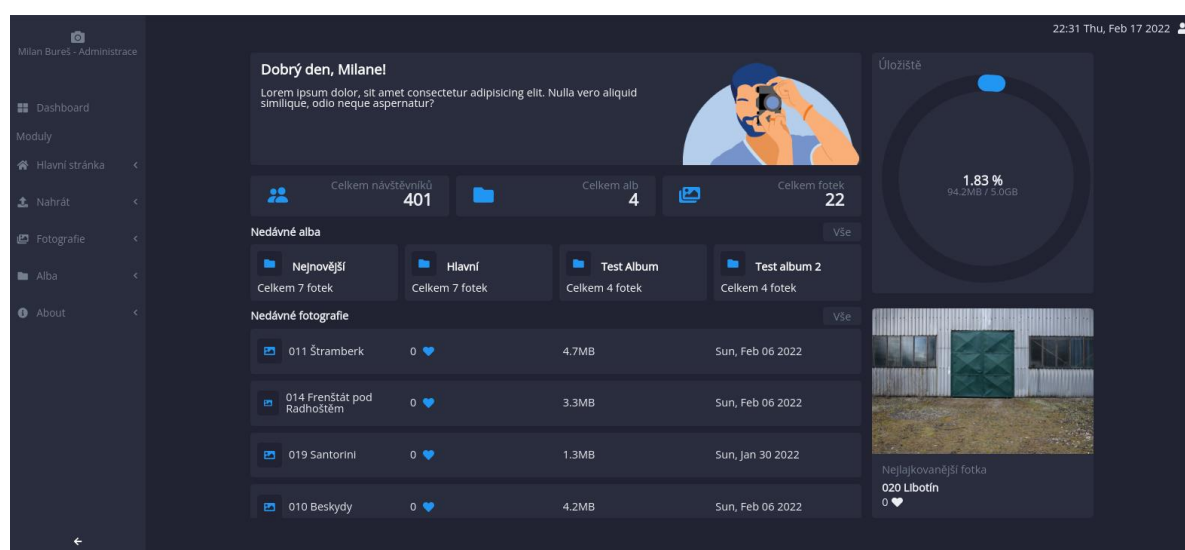
Další vylepšení webu

Webová stránka je skoro kompletně hotová. Obsahuje vše, co ta původní měla, a ještě pár detailů navíc. Jediné, co mi chybí je zoptimalizovat galerii i pro menší zařízení, než je počítač tak, aby se daly fotografie pohodlně prohlížet i například na telefonu.

Dále musím zprovoznit kontaktní formulář - Firebase (Cloud Computing provider) podporuje Cloud Functions (službu, která by mi dovolila zpracovávat HTTP Requests na backend serveru, aby systém automaticky posílal zákazníkovi oznámení/email že mu někdo psal) jen ve chvíli, kdy se za to zaplatí, a proto jsem si to nechal až na konec, kdy budeme přepisovat jeho doménu ke Googlu a podobně, prostě až budeme celý projekt finalizovat.

V budoucnu mám v plánu, jelikož je web napsaný ve vanilla javascriptu, přepsat web pomocí Gatsby.js, což je javascript framework pro generování statických HTML stránek za použití React.js, což bude mít za následek nejen mnohonásobně jednodušší budoucí vývoj, ale také rychlejší načítací časy.

Administrační systém



Obr.1 - Ukázka administračního systému

Administrační systém je systém, který zprostředkovává manipulaci s daty, které se objevují na dané webové stránce. Jednoduše řečeno, díky administračnímu systému je uživatel schopen své fotografie nahrávat, mazat, upravovat jejich název a popis, stejně tak jako uvidí, která z nich je nejvíce oblíbená na základě *lajků* a podobně.

Dále administrační systém obsahuje funkcionalitu pro manipulaci s alby, do kterých bude uživatel své fotografie rozdělovat. Ty samozřejmě může obdobně vytvářet, mazat a upravovat jejich fotografický obsah. V neposlední řadě systém obsahuje i informační nástěnku, kde je velmi dobře vidět například kolik fotografií je v danou chvíli nahraných na webu a v jakých albech, kolik úložného prostoru jeho fotografie zabírají anebo třeba seznam posledních nahraných fotografií, aby je nemusel hledat.

Budoucí plány

Administrační systém je také skoro hotový. Zákazník mě požádal, abych implementoval změnu pořadí fotek v albu přetažením, což pro mě bude poměrně velká výzva, jelikož s tímto nemám vůbec žádnou zkušenost.

Dále budu muset implementovat algoritmus komprese obrázků, jelikož fotografie, které jsou použity jako miniatury (= náhledy) mají původní velikost, která je často přes 30 MB, což je velmi nežádoucí a má to za následek trochu delší prvotní načítací časy, jelikož je prohlížeč ukládá do své cache databáze.

Technické záležitosti

Jak je u webových stránek a aplikací zvykem, pokud chceme, aby se k nim dostali i uživatelé z jiné sítě, je nutné, abychom je někde hostovali.

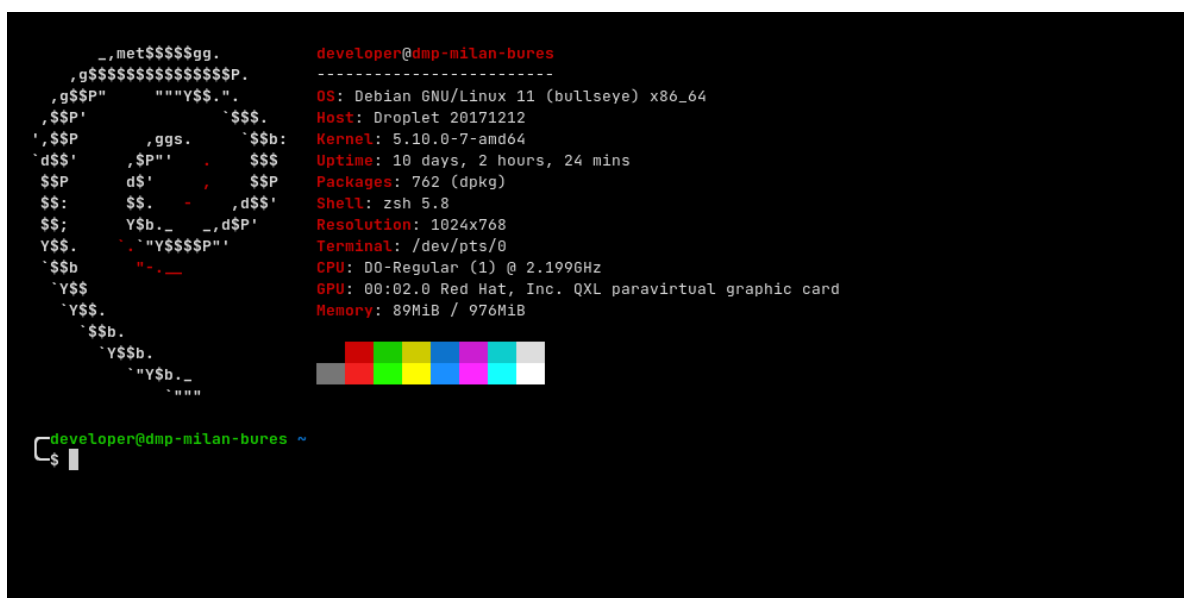
Celou funkční část projektu, která nepatří k mé dlouhodobé maturitní práci a je čistě pro zákazníka) zprostředkovává Firebase, což je Cloud Computing provider od Googlu. Díky Firebase mám zdarma (dokud nevyčerpám resources) hosting až do 10 GB souborů (což by projekt sám o sobě neměl nikdy přesáhnout), NoSQL databázi ve formě Firestore, která slouží jako úložiště záznamů o fotkách, albech, ale také analytických datech, které jsou později zobrazeny na hlavní stránce administračního systému a Storage, který slouží jako úložiště pro všechny fotografie, které uživatel nahraje.

VPS na kterém poběží nekomerční část (maturitní) je poskytnuta od Cloud Computing providera Digital Ocean. Chtěl jsem nejprve využít Googlu, abych měl vše pěkně při sobě, ale Google mi nechtěl verifikovat kartu, proto jsem se rozhodl pro DO.

Můj vlastní VPS hosting v rámci dlouhodobé maturitní práce

Jelikož celý projekt je také mou maturitní prací, je potřeba, aby projekt zasahoval do více předmětů, než je jen programování a databáze. Proto jsem se s vedoucím mé práce dohodl, že abych mu dokázal, že jsem schopný efektivně pracovat v linuxovém prostředí (i když jej používám na denní bázi místo Windows), celý projekt spustím na svém vlastním VPS (Virtual Private Server), který poběží na Linuxu. Samotný virtuální počítač už mám a provedl jsem tam již základní konfiguraci uživatelského a vývojového prostředí, ale projekt na něm zatím neběží.

Zvolená Linux distribuce je Debian 11 a jako webový server je a bude použit nginx, pro jeho jednoduchou konfiguraci a schopnost lépe zvládat nápor requestů (i když těch na tomto "testovacím" serveru moc nebude).



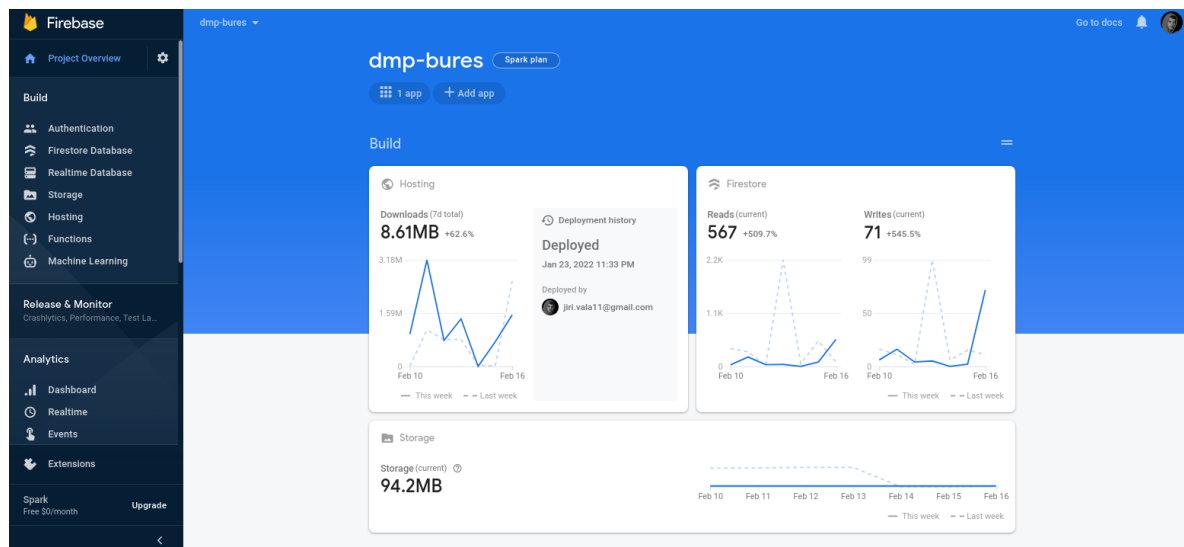
```
developer@dmp-milan-bures
-----
OS: Debian GNU/Linux 11 (bullseye) x86_64
Host: Droplet 20171212
Kernel: 5.10.0-7-amd64
Uptime: 10 days, 2 hours, 24 mins
Packages: 762 (dpkg)
Shell: zsh 5.8
Resolution: 1024x768
Terminal: /dev/pts/0
CPU: D0-Regular (1) @ 2.1996GHz
GPU: 00:02.0 Red Hat, Inc. QXL paravirtual graphic card
Memory: 89MiB / 976MiB

      _ ,met$$$$$gg.
     ,g$$$$$$$$$$$$$$$P.
    ,g$P"         ""Y$$".
   ,$$P'         "Y$$$.
  ,$$P'         ,ggs.   '$$b:
 `d$$'        ,P"      .  $$$
  $$P         d$'      .  $$P
 $$:         $$       ,d$$'
 $$;         Y$b._    ,d$P'
 Y$$         '."Y$$$$P"'
 '$$b        "-._
  Y$$
  Y$$
  Y$b.
  Y$b.
  "Y$b._
   ""
developer@dmp-milan-bures ~
$
```

Obr.3 - Terminál VPS po připojení pomocí SSH

Můj plán je takový, že vzhledem k tomu, že je projekt na Githubu (cloudové úložiště pro ukládání, správu a manipulaci zdrojových kódů většinou open-source projektů), který má funkcionalitu s názvem *Github Actions* tak vytvořím *hook*, který se při každém *commitu* (nahrání nové změny v kódu) spustí, provede request na Node.js API server, který poběží také na VPS a spustí script, který automaticky naklonuje poslední změny k sobě na lokální úložiště, odkud spustí *build* scripty webu a administračního systému (build scripty jsou scripty, které vezmou zdrojový kód, zoptimalizují jej a vytvoří bundle,

který je pak možno poskytovat uživatelům), jejichž output (optimalizované soubory webu a CMS) vezme nginx a bude je *poskytovat* uživatelům, kteří si budou chtít zobrazit web nebo administrační systém.



Obr.4 - Diagram vysvětlující deployment na VPS

Zhodnocení práce

Jsem rád, že jsem měl možnost pracovat na tomto projektu, jelikož během jeho vývoje jsem se naučil hodně věcí z oblasti vývoje webových aplikací, *komunikace se zákazníky*, devops a Linuxu, které pevně věřím, že se mi během mého profesního a kariérního života budou velmi hodit a využiji je.

