Systém pro detekci zranitelností HTTPS portálů

Bc. Roman Vyčánek

Diplomová práce
2020

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Roman Vyčánek**
Osobní číslo:	**A18255**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Forma studia:	**Prezenční**
Téma práce:	**Systém pro detekci zranitelností HTTPS portálů**
Téma práce anglicky:	**Vulnerability Detection of HTTPS Portals System**

Zásady pro vypracování

1. Specifikujte funkční požadavky vyvíjeného systému.
2. Zpracujte návrh aplikace s důrazem na její zabezpečení.
3. Proveďte implementaci v testovacím prostředí.
4. Ověřte funkčnost implementace řešení dle požadovaných funkcí (bod 1.).
5. Zpracujte implementační manuál a průvodce pro uživatele.

Rozsah diplomové práce:
Rozsah příloh:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. BINNIE, Chris. *Linux Server security: hack and defend*. Indianapolis, Indiana: Wiley, [2016].
2. RANKIN, Kyle. *Linux? hardening in hostile networks: server security from TLS to TOR*. Boston: Addison-Wesley, [2018]. Pearson open source software development series. ISBN 0134173260.
3. SORIANO, Miguel. *Information and network security*. Prague: Czech Technical University, [2013]. ISBN 978-80-01-05297-6.
4. KIM, Peter. *Hacking: praktický průvodce penetračním testováním*. Přeložil Jan POKORNÝ. Brno: Zoner Press, 2015. Encyklopedie Zoner Press. ISBN 978-80-7413-313-8.
5. OCCUPYTHEWEB. *Linux basics for hackers: getting started with networking, scripting, and security in Kali*. San Francisco: No Starch Press, [2018].

Vedoucí diplomové práce: **Ing. David Malaník, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **28. listopadu 2019**
Termín odevzdání diplomové práce: **15. května 2020**

L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Ve Zlíně dne 9. prosince 2019

Jméno, příjmení: Roman Vyčánek
Název diplomové práce: Systém pro detekci zranitelností HTTPS portálů
Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Roman Vyčánek v. r.
podpis diplomanta

ABSTRAKT

Cílem diplomové práce je vytvořit portál, který umožní uživatelům testovat zranitelnost jiných webových portálů. Tento portál přiřazuje uživatelům tři role: administrátor, uživatel, a neschválený uživatel. Administrátor má možnost spravovat ostatní uživatelské účty, spravovat všechny plánované testy, prohlížet historii přihlašování všech uživatelů a prohlížet výsledky všech testů. Uživatel má možnost si naplánovat testovací úlohu anebo jednorázově otestovat portál. Nový uživatel čeká na schválení administrátorem portálu a nemá přístup k žádnému testování. Všechny výsledky jsou ukládány do databáze pro zpětný přístup. V teoretické části je popsán význam jednotlivých výsledků penetračních testů a v praktické části je návod k instalaci tohoto portálu, návod k obsluze, přehled databázových tabulek i výtažky z kódu, který tyto testy provádí.

Klíčová slova: Penetrační testy, Laravel, PHP, Debian, GNU GPLv3

ABSTRACT

The aim of this thesis is to design web portal that will allow users to test the vulnerability of other web portals. There are three roles for users: administrator, user, new user. Administrator can manage other users accounts, manage all scheduler tests, view the login history of all users, and view results of all tests. The user account has the option to schedule a test task or test the portal once. The new user is waiting for approval by the portal administrator and does not have access to any testing. All results are stored in database for reverse access. The theoretical part describes the meaning of individual results of penetration tests and in the practical part there is an instruction how to install this portal, an operating manual and an overview of database tables as well as extracts from the code that performs these tests.

Keywords: Penetration tests, Laravel, PHP, Debian, GNU GPLv3

Tímto bych rád poděkoval svému vedoucímu panu Ing. Davidu Malaníkovi, Ph.D. za vedení diplomové práce a čas strávený při konzultacích. Dále bych chtěl poděkovat své rodině za podporu a motivaci při tvorbě této práce. Nakonec bych chtěl poděkovat i své přítelkyni za trpělivost a zvládnutí období přehlížení.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD

Tato diplomová práce se věnuje návrhu, vývoji a nasazení testovacího portálu. Testovací portál má sloužit ke zjednodušení práce penetrační laboratoře. V tomto portálu bude mít uživatel možnost jednorázově otestovat zranitelnost jak portálů pouze na vnitřní síti, tak i portálů připojených k internetu. Díky tomu bude mít uživatel rychlý přehled zranitelností se zvýrazněnými závažnými chybami. Následně tyto testy budou ukládány do databáze pro případ zpětné kontroly nebo nutnosti znovu zobrazení výsledků. Dále každý uživatel bude mít možnost testy plánovat a spouštět automaticky, výsledky těchto testů se budou také ukládat do databáze, a navíc jejich výsledek přijde i na emailovou adresu uživatele. Pro zajištění bezpečnosti a zamezení spouštění testů komukoli je nutné uživatele nejprve potvrdit. Tuto akci má povolenou pouze administrátor systému. Administrátor má také možnost upravovat plánované úlohy všech uživatelů a další akce.

Teoretická část popisuje základní požadavky na vytvářený systém. Dále jsou zde rozebrány jednotlivé zranitelnosti a popis testů a jejich výsledků. A v závěru teoretické části jsou popsány jednotlivé technologie, které byly použity při tvorbě této aplikace.

Praktická část začíná přehledem zdrojových kódů jednotlivých stránek a jejich controllerů. Dále následuje podrobný implementační manuál s nastavením Debianu tak, aby zde mohl být tento portál spuštěn. Praktická část končí uživatelským manuálem, kde jsou popsány akce, které může administrátor nebo uživatel dělat.

Cílem práce je návrh a implementace portálu pro testování zranitelností HTTPS portálů a jeho následné nasazení na vnitřní síti fakulty.

I. TEORETICKÁ ČÁST

1 POŽADAVKY NA SYSTÉM

Výstupem této diplomové práce je vytvoření systému, který bude schopen detekovat sadou testů zranitelnosti HTTPS portálů. Systém bude schopen plánování testů a správy přístupu k testování na základě přidělených rolí uživatelům.

1.1 Ověření certifikátu HTTPS

Systém bude schopen pro zvolený portál ověřit platnost HTTPS a přehledně zobrazit informace získané o daném certifikátu.

1.1.1 Zobrazované informace

Systém bude schopen získat a následně zobrazit subjekt, pro který byl daný certifikát vystaven, používané jméno daného portálu a jeho zástupné jméno. Sériové číslo certifikátu, a datum kdy byl certifikát vystaven a datum splatnosti. Dále systém bude zobrazovat jméno certifikační autority, která vydala certifikát, šifrovací algoritmus použitý pro podpis a zjištění DNS CAA.

1.2 Zjištění konfigurace portálu

Systém bude schopen zjistit, jaké komunikační protokoly daný portál podporuje. Bude testovat, pomocí jakého šifrování daný portál umožňuje komunikaci a jakou technologii využívá.

1.2.1 TSL a SSL protokoly

Systém bude schopen zkusit komunikaci s portálem pomocí SSL 2 a SSL 3, a pomocí protokolu TSL 1.0, TSL 1.1, TSL 1.2 a TSL 1.3

1.2.2 Povolené šifry

Systém bude zkoušet spojení s daným portálem pomocí různých šifer. Bude schopen zobrazit, pomocí kterých šifer bylo možno komunikovat s daným portálem a barevně odliší slabé šifry od šifer považovaných za bezpečné.

1.3 Testování zranitelností

Systém bude schopen zobrazit podrobný protokol o možných zranitelnostech jako jsou například BEAST attack, POODLE, Heartbleed, SWEET32, a dalších.

1.4 Kontrola hlaviček odpovědi

Systém bude schopen na daném portálu analyzovat HTTP hlavičky odpovědí na dotazy. Tyto hlavičky jsou zodpovědné za bezpečnost navázaného spojení mezi portálem a klientem. Systém bude kontrolovat přítomnost hlaviček X-Frame-Options, Strict-Transport-Security, X-Content-Type-Options, Content-Security-Policy, Referrer-Policy a Feature-Policy.

1.5 Omezení přístupu podle rolí

Systém bude mít implementované brány, které omezí přístup uživatelům ke stránkám s managementem aplikace. A omezí možnosti spouštění testů nepotvrzených uživatelů administrátorem.

1.6 Ukládání výsledků

Systém bude mít možnost uložit každý provedený test do databáze a bude mít možnost se k těmto testům zpětně vrátit.

1.7 Plánování testů

Systém bude mít možnost vytvoření a automatického spouštění plánovaných úloh. Uživatel bude informován emailovou zprávou o jejich výsledcích.

1.8 Editace barev

Systém bude mít možnost editace barev administrátorem systému. Tyto barvy budou dále používány při testech z důvodu zpřehlednění výsledků a rychlejší orientaci.

1.9 Historie přihlašování

Systém bude ukládat každé přihlášení uživatele do databáze z důvodu bezpečnosti. Administrátor bude mít možnost historii všech uživatelů procházet.

1.10 Aktualizace testovacích nástrojů

Systém bude mít možnost aktualizace externích testovacích nástrojů z důvodu zachování aktuálnosti penetračních testů.

2 TESTY

V této kapitole se nachází přehled jednotlivých testů, které následně bude portál implementovat.

2.1 Test zranitelností

V této kapitole je podrobný popis různých zranitelností portálů, které budou dále v praktické části implementovány.

2.1.1 CCS Injection

OpenSSL neomezuje řádné zpracovávání zpráv ChangeCipherSpec, což umožňuje útočníkům typu MITM aktivovat nulovou délku master klíče v specifických OpenSSL-to-OpenSSL komunikacích a následně ukradnout relace, anebo získat citlivé informace. [22]

2.1.2 Heartbleed

Zranitelnost Heartbleed je součástí knihovny OpenSSL. Tato slabina umožňuje ukradení za normálních podmínek chráněných informací zabezpečením pomocí SSL nebo TSL protokolů. Systémy zabezpečené zranitelnou verzí OpenSSL poskytují možnost čtení paměti komukoliv připojenému na internetu. Takto se útočníci mohou dostat k tajným klíčům, které se používají pro ověření poskytovatelů služeb, mohou dále dešifrovat komunikaci se zranitelným serverem nebo získat hesla a přihlašovací jména uživatelů. [23]

2.1.3 Ticketbleed

Ticketbleed je softwarová zranitelnost TSL a SSL balíku v F5 BIG-IP zařízeních, umožňující vzdálenému útočníkovi získat 31 bajtů neinicializované paměti. Tato paměť může obsahovat klíčová data, nebo osobní data dalších připojení. Útočník poskytne serveru Session ID a Session ticket a server má následně vrátit Session ID jako signál pro přijmutí ticketu. Pokud však Session ID má délku 1 bajt, server odpoví vždy blokem o velikosti 32 bajtů z nichž 31 je neinicializovaná paměť. [24]

2.1.4 ROBOT

Zranitelnost Robot je návrat 19let staré chyby zabezpečení, která umožňuje provádět dešifrování a podepisování RSA pomocí soukromého klíče serveru TLS. Útok spočívá v tom, že SSL server odesílá chybové hlášky šifrované podle standardu PKCS # 1 v1.5 a jejich padding umožňuje adaptive-chosen ciphertext attack (CCA2) útok. Tento útok plně narušuje důvěrnost TLS při použití se šifrováním RSA. [25]

2.1.5 Secure Renegotiation

Secure Renegotiation je zranitelnost protokolů SSL a TSL při vytváření spojení s klientem. Útočník vytvoří spojení s cíleným serverem do tohoto spojení poté spojí nové spojení TLS od klienta. Server považuje počáteční TLS handshake klienta za nové vyjednávání, a proto věří, že počáteční data přenesená útočníkem jsou od stejné entity jako následující klientská data. Útočník díky tomu vidí počáteční TLS handshake klienta v čitelné podobě. [26]

2.1.6 Secure Client-Initiated Renegotiation

Proces vytvoření šifrovaného SSL spojení používá na serveru podstatně více zdrojů než na klientovi. Útočník má díky této vlastnosti možnost zamezit dalším uživatelům v připojování k tomuto serveru útokem Denial of Service. [27]

2.1.7 Crime, TLS

Útok Crime je útok založený na nedostatečném zmatení délky nešifrovaných dat v TSL 1.2 a starších. Tento protokol se používá v Mozilla Firefox, Google Chrome, QT a dalších produktech. Tato zranitelnost umožňuje útokem typu MITM získat prostý text HTTP hlaviček. Útok spočívá v pozorování rozdílů v délce během řady odhadů, ve kterých se požadavek HTTP potenciálně shoduje s neznámým řetězcem v hlavičce HTTP. [28]

2.1.8 BREACH

Útok Breach je útok založen na nedostatečném zmatení délky nešifrovaných dat při komunikaci HTTPS. Tato zranitelnost umožňuje útokem typu MITM získat čitelná data. Útok spočívá v odesílání mnoha HTTP požadavků a porovnáváním, zda se jejich řetězec s URL neshoduje s řetězcem v neznámém těle HTTP odpovědi. [29]

2.1.9 **POODLE, SSL**

SSL ve verzi 3 umožňuje využitím paddingu zjištění textu v daném bloku. Hlavním požadavkem pro úspěšný útok je, že útočník musí být schopen měnit pakety v komunikaci klienta se serverem, což znamená MITM útok. Klient musí spustit útočníkův javascript, který bude generovat HTTPS dotazy. Následně útočník vždy vezme jeden blok a zkopíruje ho na poslední pozici v paddingu. Pokud server spojení neukončí, může útočník XORem s předchozím blokem zjistit dešifrovanou hodnotu tohoto bloku. [30]

2.1.10 **TSL_FALLBACK_SCSV**

Tato zranitelnost souvisí s nastavením klientů připojujících se k serverům. Z důvodu umožnění komunikace i se staršími servery se po neúspěšném připojení pomocí nejnovějšího protokolu pokusí připojit pomocí protokolu staršího. Tito klienti mohou tímto způsobem následně navázat spojení pomocí TLS 1.0 nebo předchůdcem SSL 3.0. Takto snížené zabezpečení komunikace je pro útočníka následně jednodušší k dešifrování. [31]

2.1.11 **SWEET32**

Šifry DES a triple DES, jak jsou využívány v protokolech TSL a SSH a dalších mají narozeninovou hranici přibližně čtyři miliardy bloků, což zjednodušuje vzdáleným útočníkům získání čitelného textu skrze provedení narozeninového útoku na dlouhodobě šifrované relace. Jak bylo dokázáno útokem na HTTPS relaci využívající triple DES v CBC módu, tomuto útoku se říká Sweet32 útok. [32]

2.1.12 **FREAK**

Funkce ssl3_get_key_exchange v souboru s3_clnt.c v OpenSSL umožňuje vzdálenému SSL serveru řídit RSA-to-EXPORT_RSA downgrade útok a využitím brute-force dešifrování použitelného na slabé EXPORT_RSA získat čitelná data. [33]

2.1.13 **DROWN**

Protokol SSLv2 ve stavu, jak se využívá v OpenSSL a dalších produktech požaduje po serveru odeslání ServerVerify zprávy před rozhodnutím, zda má klient doručená RSA data, což umožňuje útočníkovi dešifrovat šifrovaný text využitím nástroje pro Bleichenbacher RSA padding oracle útok. [34]

2.1.14 **LOGJAM**

Zranitelnost Logjam se vztahuje k protokolům TSL 1.2 a mladším. Útok lze provést, pokud server má povolen DHE_EXPORT, ale klient ne. Tato zranitelnost umožňuje pomocí útoku typu MITM snížit šifrování komunikace mezi serverem a klientem. Tento útok souvisí s přepsáním ClientHello s DHE za DHE_EXPORT a přepsáním ServerHello s DHE_EXPORT za DHE. [35]

2.1.15 **BEAST**

TLS 1.0 a protokoly předcházející obsahují zranitelnost, která spočívá vtom, že inicializační vektory, které se používají k maskování dat, se následně šifrují blokovou šifrou, kterou lze odhalit aktivním MITM útokem. Útočníkovi stačí dostatek času a je schopný zjistit, jak odesílaná data vypadala. Protože útočník tipuje inicializační klíč, většinou se mu podaří získat jen malý objem dat, ale to stačí k získání session cookies a autentizačních údajů. [36]

2.1.16 **LUCKY13**

Protokol TSL 1.1 a 1.2 a protokoly DTLS 1.0 a 1.2, jak jsou využívány v OpenSSL, OpenJDK, a dalších produktech, nezohledňují časované útoky bočními kanály pomocí kontrol MAC, což umožnuje vzdáleným útočníkům provádění různých útoků a obnovu čitelného textu skrze statistickou analýzu vytvářených paketů. [37]

2.1.17 **RC4**

Algoritmus RC4, tak jak je využíván v TSL a SSL protokolech má příliš mnoho jednobajtových biasů, což zjednodušuje vzdáleným útočníkům provádění rekonstrukce čitelného textu statistickou analýzou šifrovaného textu pomocí velkého množství relací, které používají stejný čitelný text. [38]

2.2 Test hlaviček odpovědí

V této kapitole bude popsán význam jednotlivých hlaviček, které používají portály k nastavení komunikace mezi uživatelem a portálem.

2.2.1 X-Frame-Options

Hlavička XFO chrání uživatele proti clickjacking útokům. Útočník může přes svou stránku načíst stránku jinou, kterou ale zprůhlední. Pokud poté návštěvník takto upravených stránek klikne na nějaký link, kliká ve skutečnosti na úplně jiné elementy. Tím se útočník může dostat k právům, které mají jen přihlášení uživatelé, a díky tomu využívat zvýšených oprávnění k dalším účelům. Validní hodnoty této hlavičky jsou "DENY", což znamená zakázaní framování, "SAMEORIGIN" - tato hlavička povoluje framování své vlastní stránky a "ALLOW-FROM", díky které lze nastavit stránky, které mohou framovat původní stránku. [61]

2.2.2 Strict-Transport-Security

Tato hlavička HSTS chrání uživatele před útokem typu MITM, kdy útočník odposlouchává spojení oběti a portálu, s kterým oběť komunikuje. Aby útočník viděl komunikaci nezabezpečenou, změní protokol komunikace z HTTPS na nezabezpečený protokol HTTP. Tato hlavička obsahuje čas, po který má být spojení zabezpečeno "max-age=31536000" a může mít další direktivy, jako je třeba "includeSubDomains", kterým se nastavuje, že toto nastavení platí i pro subdomény daného portálu. [64]

2.2.3 X-Content-Type-Options

Tato hlavička zakazuje prohlížečům procházení odkazových souborů a vykonávání nalezených scriptů na stránce. Brání tomu, aby prohlížeče nespustily útočníkem přidaný Java script na daném portálu. Existuje pouze jeden parametr, který se používá, a to je "nosniff". [65]

2.2.4 Content-Security-Policy

Tato hlavička určuje, z jakých zdrojů je možno stahovat soubory. Na webových stránkách se používají různé doplňkové assety, které se stahují spolu s obsahem. Jedná se např. o Google analytics, kaskádové styly a další. Tak lze bránit útočníkovi v Cross Site Scripting útocích. Nastavení v této hlavičce může být třeba "script-src 'self'" čímž se povolí zdroje z vlastní domény. [60]

2.2.5 **Referrer-Policy**

Tato hlavička určuje, kolik informací bude uživatel odesílat při přechodu na další portál. Příchodem na jakoukoliv stránku se odesílá serveru informace o zdroji příchodu, ale touto hlavičkou se tento přenos informací dá zakázat nebo omezit. Portály si díky tomu mohou dělat statistiky, z které webové adresy se k nim uživatelé dostávají. Nastavení se provádí za pomocí klíčových slov, “no-referrer” - nebude s žádostí žádná informace poslána, “no-referrer-when-downgrade" - informace se neposílá na portály, pokud se sníží zabezpečení komunikace z HTTPS na HTTP. Lze použít omezení, kdy se neodesílá celá cesta, ale jen origin. Při použití klíčového slova “origin” se nerozlišuje mezi protokoly komunikace. Při použití klíčového slova “strict-origin” se havička neodešle, když se sníží úroveň zabezpečení. Dále lze nastavit, kolik informací se bude odesílat v závislosti na tom, jestli uživatel zůstává na stejné doméně, nebo z ní odchází. “Same-origin” nastaví, že informace o předchozí stránce se budou odesílat pouze na stejné doméně a na jiné domény se nebude odesílat nic. Dále se používá “origin-when-cross-origin" - tato volba znamená, že se na stejné doméně odesílají celé cesty a pokud uživatel přechází na jinou doménu, je s tím požadavkem posláno jen doménové jméno původního portálu. Dále se používá “strict-origin-when-cross-origin" - jde o možnost, že se neposílají informace v hlavičce, pokud se sníží bezpečnost komunikace z HTTPS na HTTP. Existuje i nedoporučovaná možnost “unsafe-url”, kdy se vždy odesílá celá adresa URL na libovolnou žádost. [63]

2.2.6 **Feature-Policy**

Tato hlavička povoluje portálu zapnout nebo naopak vypnout určité funkce prohlížeče a API pro zvýšení bezpečnosti a soukromí. Tato hlavička není standardizována, je experimentální a stále se vyvíjí. Syntaxe této hlavičky je “Feature-Policy: <feature> <allow list origin(s)>”, kdy se takto nastavuje omezení pro prvek, třeba akcelerometr, a pro zdroje, které mají k němu přístup. [62]

Tyto hlavičky se dají poslat jako jednotlivé záznamy tímto způsobem.

```
Feature-Policy: unsized-media 'none'
Feature-Policy: geolocation 'self' https://example.com
Feature-Policy: camera *;
```

Hlavičky lze poslat i jako součást jednoho záznamu:

```
Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com;
camera *;
```

Tento způsob zaslání zakáže používání unsized-media na webových stránkách. Zakázání geolokace platí pro všechny stránky kromě stránek s původem self a povolí používání kamery v celém kontextu stránek.

Dále se dá vytvořit `<iframe allow="camera 'none'; microphone 'none'">`, kterým se zablokuje používání kamery a mikrofonu.

2.3 Test komunikačních protokolů

Tato část popisuje jednotlivé komunikační protokoly, což jsou množiny pravidel, určující syntaxi a význam jednotlivých zpráv při síťové komunikaci.

2.3.1 SSL

Secure Sockets Layer (SSL) je standardní bezpečnostní technologie pro vytvoření šifrovaného spojení mezi serverem a klientem, což obvykle znamená webovým serverem a prohlížečem, nebo poštovním serverem a poštovním klientem.

SSL umožňuje bezpečný přenos citlivých informací, jako jsou čísla kreditních karet a přihlašovací údaje. Normálně by byla data odesílána mezi prohlížeči a webovými servery v prostém textu což by velmi zjednodušilo útok MITM. [39]

2.3.2 TLS

TLS je kryptografický protokol, který poskytuje zabezpečení komunikace end-to-end při procházení internetu anebo při provádění online transakcí. Je založen na standardu IETF, který má zabránit odposlechu, manipulaci a padělání zpráv. Protokol TSL je efektivnější a bezpečnější než jeho předchůdce SSL, protože má silnější autentizaci zpráv, podporuje předem sdílené klíče, klíče generované pomocí eliptických křivek a protokol Kerberos. [40]

2.3.3 NPN/SPDY

SPDY je experimentální protokol vyvinutý společností Google, jehož cílem je snížit latenci webových stránek. Řeší omezení http 1.1 a odstraňuje existující slabá místa jako je blokování linek a neefektivní používání TCP. Pro vytvoření připojení uživatele k serveru se vymění prvotní rámce pomocí SSL šifrovaného tunelu.

NPN je rozšíření protokolu SSL, které umožňuje klientovi a serveru vyjednávání o aplikačním protokolu, jako součást SSL handshake. [41]

2.3.4 ALPN/HTTP2

ALPN je protokol rozšiřující protokol TLS, který rozšiřuje dotaz „hello“ i vyjednávání o komunikačním protokolu. Tento protokol je schopný vyjednat, který protokol by měl být využit pro komunikaci při zabezpečeném připojení způsobem, který je efektivní a který se vyhýbá dalším zbytečným dotazům mezi klientem a serverem. [42]

HTTP2 má za cíl nahrazení HTTP 1.1. Mezi hlavní výhody patří rychlost, jednoduchost a robustnost. Poskytuje možnost řešení různých problémů na transportní vrstvě. Jako primární cíle nového protokolu je snížení latence, umožnění úplné multiplexace požadavků a odpovědí, minimální režie samotného protokolu pomocí komprese polí hlavičky http a podpora prioritizace požadavků. [43]

2.4 Test podporovaných šifer

Každý z komunikačních protokolů podporuje různé druhy šifrování komunikace. Tyto šifry se starají o zabezpečení komunikace mezi klientem a serverem proti útoku MITM. Mezi základní používané šifry patří AES a triple DES. S bitovou délkou klíče mezi 128 až 256 bity. Výměna klíče probíhá pomocí RSA anebo s využitím ECDH.

2.5 Test server hello

Server hello je zpráva, kterou odpovídá server na klientův prvotní dotaz Client hello, kterým se začíná navazovat spojení mezi serverem a klientem.

2.5.1 TLS extensions

Protokol TSL obsahuje mechanizmus pro přidávání dalších funkčních doplňků. Těmito rozšířeními je možné rozšířit možnosti TSL bez nutnosti měnit vlastní TSL protokol. Možné doplňky jsou server_name, cert_type, signed_certificate_timestamp, pre_shared_key a další. [44]

2.5.2 Session Ticket RFC 5077

Session tickety, specifikované v RFC 5077, jsou možností, jak obnovit TLS spojení ukládáním klíčových dat zašifrovaných na straně klienta. Ve verzi TLS 1.2 zrychlily handshake ze dvou cyklů na jeden. [45]

2.5.3 SSL Session ID support

Je to podpora SSL Session ID doplňku, který umožňuje uživateli si uložit Session ID, díky kterému se k serveru připojí bez nutnosti vytváření celého SSL spojení od začátku. [46]

2.5.4 Session Resumption

Session Resumption je možnost obnovení spojení mezi klientem a serverem za pomoci Session ID anebo Session Ticketů, čímž se zkrátí čas pro navázání spojení po přerušení.

2.5.5 TLS clock skew

TLS clock skew je časový rozdíl mezi časem klienta a serveru. V základním nastavení TLS nevyžaduje, aby delta t = 0, ale některé doplňky TLS to mohou vyžadovat. [47]

2.5.6 Signature Algorithm

Signature algorithm definuje algoritmus, který se používá pro podepisování obsahu. Tento podpis následně slouží pro ověření, zda obsah nebyl po cestě mezi klientem a serverem upraven.

2.5.7 Server key size

Server key size je délka klíče v SSL certifikátu. S větší délkou je klíč bezpečnější, ale také se déle přenáší, což znamená pomalejší handshake na začátku každého nového spojení. Používaná délka je 2048 bitů. [48]

2.5.8 Server key usage

Server key usage definuje hlavní způsoby využití serverového klíče, jako je vytváření digitálně podepsaných dat při předávání klíčů a při handshake.[49]

2.5.9 Server extended key usage

Server extended key usage definuje další způsoby využití serverového klíče, jako je ověření serveru a ověření klienta. [49]

2.5.10 Serial / Fingerprints

Zde je uveden hashovací algoritmus, kterým byl certifikát hashován, což slouží pro ověření pravosti certifikátu. [49]

2.5.11 **Common Name**

Toto pole obsahuje název portálu, pro který byl certifikát vystaven. [49]

2.5.12 **subjectAltName**

Toto pole obsahuje další portály, které jsou podepsané stejným certifikátem. [49]

2.5.13 **Issuer**

V poli Issuer je uvedena autorita, která vydala a podepsala tento certifikát. [49]

2.5.14 **Trust (hostname)**

Toto pole je výsledek kontroly, zda zadaná URI je mezi portály, pro který byl certifikát vystaven.

2.5.15 **Chain of trust**

Chain of trust zkouší, zda je certifikát spojen s důvěryhodnou certifikační autoritou. [50]

2.5.16 **EV cert**

Certifikát EV je certifikace nejvyšší možné důvěryhodnosti na internetu. [51]

2.5.17 **Certificate validity**

Certificate validity je pole, které určuje období platnosti certifikátu. [49]

2.5.18 **Certificate Revocation list**

Certificate Revocation list je seznam zneplatněných certifikátů.

2.5.19 **OCSP URI**

OCSP URI je adresa webového serveru, poskytující informace o stavu certifikátu. [52]

2.5.20 **OCSP stapling**

OCSP Stapling je technika získávání informací o stavu certifikátu již od počátečního HTTPS připojení. Server odpověď OCSP připojí ve své odpovědi automaticky, takže není nutné vytvářet zvlášť dotaz pro OCSP. [53]

2.5.21 **OCSP must stample extension**

Pro ochranu uživatele se používá rozšíření, které říká prohlížeči, že vždy když získá certifikát tak je k němu vždy přiložena i odpověď OCSP. Pokud tato odpověď přiložena není prohlížeč stránku zablokuje. [53]

2.5.22 **DNS CAA RR**

Certification Authority Authorization (CAA) DNS Resource Record umožňuje vlastníkovi DNS doménového jména specifikovat jednu nebo více certifikačních autorit, autorizovaných vydávat certifikáty pro tuto doménu. Záznamy CAA umožňují veřejné certifikační autoritě implementovat další kontroly, které slouží ke snížení rizika nesprávného vydání certifikátu. [54]

2.5.23 **Certificate Transparency**

Projekt Certificate Transparency řeší několik strukturálních nedostatků v systému certifikátů SSL. Tyto nedostatky oslabují spolehlivost a účinnost šifrovaného internetového připojení a mohou ohrozit mechanizmy TLS anebo SSL. Tento projekt umožňuje detekovat SSL certifikáty, které byly omylem vydány certifikační autoritou, nebo nelegálně získány. Umožňuje také identifikovat certifikační autority, které vydávaly certifikáty pro škodlivé účely. [55]

3 VÝBĚR A POPIS POUŽITÝCH TECHNOLOGIÍ

Tato část se věnuje technologiím, použitým při vývoji testovacího portálu. Je zde popsán operační systém, na kterém portál běží, použité programovací jazyky, vývojové prostředí, v kterém je kód portálu napsán i frameworky, které byly využity při vývoji.

3.1 Debian GNU/Linux

Debian je jednou z nejstarších Linuxových distribucí, který je vyvíjen komunitou z celého světa. Operační systém obsahuje základní programy, které umožňují provoz počítače. Všechny nástroje v tomto systému jsou svobodné. Další funkce se do tohoto operačního systému přidávají v podobě balíčků. Součástí operačního systému je i správce balíčků (APT), díky kterému se tyto balíčky do systému stahují a instalují. Hlavní výhodou tohoto operačního systému je stabilita, jsou známy servery, které na Debianu běží déle než rok bez nutnosti restartu. [20]

3.2 Výběr programovacího jazyka

Hlavním kritériem pro volbu programovacího jazyka je bezpečnost, která zabrání zkopírování nebo modifikaci testů. Tyto testy potřebují vyšší výpočetní výkon, a proto je nutné je provádět na straně serveru. Z tohoto důvodu bude využit programovací jazyk PHP, který se vykonává na straně serveru. Dalšími jazyky, které budou sloužit k vytváření vzhledu webového portálu jsou HTML a CSS.

3.2.1 **PHP**

PHP je skriptovací jazyk, který je vhodný pro vývoj webových stránek. Zkratka PHP znamená PHP: Hypertext Preprocessor. PHP kód je zpracován pomocí PHP interpretu, který je implementován jako modul. Výsledek se poté uživateli posílá jako HTML. PHP v této aplikaci byl využit z důvodu použití webového frameworku Laravel. [17]

3.2.2 HTML

HTML je značkovací jazyk, který se používá pro vývoj webových stránek. Zkratka HTML znamená Hypertext Markup Language. HTML kód překládají internetové prohlížeče, což znamená, že je prováděn na straně uživatele. Spolu s tímto programovacím jazykem se využívají taky kaskádové styly CSS a skriptovací jazyky, jako je JavaScript. Prvky HTML tvoří základní bloky webových stránek. Prvky se vytvářejí pomocí značek, kterým se říká tagy. Prohlížeče tyto tagy využívají pro interpretaci obsahu stránky. [18]

3.2.3 JavaScript

JavaScript je objektově orientovaný skriptovací jazyk a společně s HTML a CSS je nejpoužívanější jazyk při vývoji webových technologií. Interpretaci provádí webový prohlížeč uživatele, což znamená, že se spouští na straně klienta a umožňuje do webových stránek přidat prvky pracující v reálném čase. Tento programovací jazyk se využívá v různých frameworcích, jako je například Vue.js. JavaScript byl využit pro dynamické překreslování výsledků testů. [19]

3.3 Výběr vývojového prostředí

K vývoji aplikace je potřeba vývojové prostředí. V tomto projektu se pracuje s více programovacími jazyky i s různými typy souborů, tedy je potřeba použít aplikaci, která není zaměřena pouze pro jeden programovací jazyk. Z tohoto důvodu byl vybrán nástroj Visual Studio Code. Projekt je uložen na serverech GitHubu.

3.3.1 Visual Studio Code

Visual Studio Code je editor zdrojového kódu, vyvinutý společností Microsoft pro Windows, Linux i macOS, který lze použít s velkým množstvím programovacích jazyků, včetně PHP a Node.js. Podporuje ladění a možnost zálohování kódu na GitHub, zvýrazňování syntaxe inteligentní dokončování kódu. Je vysoce přizpůsobivý a umožňuje uživatelům měnit téma vzhledu, modifikovat klávesové zkratky a instalovat rozšíření, která přidávají další funkce. Zdrojový kód je open-source a je licencován pod licencí MIT. [16]

3.4 Použité frameworky

Tato část se zabývá používanými frameworky při vývoji webových aplikací. Frameworky jsou předpřipravené základní softwarové řešení, které může vývojář používat při vývoji pro specifickou platformu. Díky tomuto přístupu vývojáři nemusí vždy začínat úplně od začátku, ale mají předdefinované základní třídy, metody a další možné prvky. [15]

3.4.1 **Laravel**

Laravel je webový framework, založený na PHP pro vývoj webových aplikací, které využívají jeho jednoduché syntaxe. Jeho součástí je rozsáhlá sada nástrojů pro aplikační architekturu. Laravel v sobě zahrnuje i různé vlastnosti jiných technologií, jako je například ASP-NET MVC, Ruby a další. Tento framework je open-source, takže šetří vývojáři čas, protože nemusí celý web vyvíjet od nuly, a navíc má k dispozici pro hledání informací širokou základnu uživatelů. [13]

Laravel v sobě zahrnuje i bezpečnostní opatření, díky čemuž je vývoj rychlejší, takže práce s tímto webovým frameworkem je velmi efektivní. [14]

3.4.2 **Blade**

Blade je jednoduchý webový framework, který se snaží o zjednodušení vývoje JavaWeb, současně zvyšuje jeho výkonnost a flexibilitu. Tento framework je dodáván s Laravelem a na rozdíl od jiných PHP modulů neomezuje vývojáře v používání prostého PHP v pohledech. Všechny pohledy Blade jsou kompilovány do prostého PHP a ukládány do mezipaměti, dokud nejsou upraveny, což znamená, že Blade do aplikace přináší téměř nulovou režii. Soubory zobrazení Blade používají příponu souboru .blade.php a obvykle jsou uloženy v adresáři resources / views. [12] [1]

3.4.3 **Vue.js**

Vue.js je progresivní JavaScript framework pro vytváření uživatelských rozhraní. Je navržen tak, aby bylo možné postupné přidávání tohoto frameworku do již existujících řešení v závislosti na různých případech použití. Skládá se ze základní knihovny, která se zaměřuje na vrstvu viditelnou a širokého ekosystému podporujících knihoven, který pomáhá řešit komplikovanější problémy v aplikaci. [11]

3.4.4 Bootstrap

Bootstrap je bezplatný front-end framework pro rychlejší a snadnější vývoj webu.

Bootstrap zahrnuje návrhové šablony založené na HTML a CSS pro formuláře, tlačítka, tabulky, navigační prvky a mnoho dalších elementů. Bootstrap dává také možnost snadno vytvářet responzivní návrhy. Obsahuje mnoho pluginů, postavených na jQuery. [10]

3.5 Apache HTTP Server

Apache HTTP Server, taky nazývaný Apache, je bezplatný, open-source software pro více platforem webového serveru, vydaný na základě podmínek Apache License 2.0. Apache vyvíjí a udržuje otevřená komunita vývojářů pod záštitou Apache Software Foundation.

Převážná většina instancí HTTP serveru Apache běží na distribuci Linuxu, aktuální verze také podporuje Microsoft Windows. [9]

3.6 MySQL

MySQL je open-source systém pro správu relačních databází. V souladu s podmínkami GNU je k dispozici pod řadou patentovaných licencí. MySQL byl vlastněn a sponzorován švédskou společností MySQL AB, kterou koupila společnost Sun Microsystems (nyní Oracle Corporation) [3]. MySQL je součástí aplikací, jako je LAMP, což je zkratka pro Linux, Apache, MySQL, Perl / PHP / Python. MySQL používá mnoho databázových webových aplikací, včetně Facebooku [5], Flickeru [6], Twitteru [7] a Youtube [8].

3.7 Shrnutí

Aplikace vytvářena v této diplomové práci je vyvíjena pro operační systém Debian. Na tomto severu je nainstalován MySQL pro správu databází a Apache pro webový server. Kvůli bezpečnosti byl zvolen jazyk PHP a pro zjednodušení práce s ním byl zvolen Laravel jako Framework, ve kterém je aplikace vyvíjena. Další jazyky, které jsou při vývoji využity jsou JavaScript a HTML. Další frameworky pro zjednodušení práce s webovým portálem jsou Vue.js, Blade a Bootstrap. Vyvíjená aplikace je tvořena v editoru Visual Studio Code.

4 NÁVRH SYSTÉMU A UŽIVATELSKÉHO ROZHRANÍ

Hlavním požadavkem na tuto aplikaci je zabezpečení, proto byl zvolen framework Laravel, který se o celkovou strukturu a zabezpečení stará. Díky využití tohoto frameworku je vzhled aplikace velmi přehledný a pro nového uživatele intuitivní. Tento framework slouží jako vizuální nástavba pro testy, které probíhají v terminálovém okně.

4.1 Návrh aplikace

Tato část popisuje základní strukturu aplikace a její use casy.

4.1.1 Licence GNU GPLv3

GNU General Public License je licence pro svobodný software. Díla pod touto licencí vyžadují v případě, že se využijí v jiných aplikacích, aby tyto aplikace byly licencovány pod stejnou licencí. Tato licence požaduje zveřejnění lidsky čitelného kódu. Omezuje přidání dalších omezení vůči uživatelům.

4.1.2 Use case diagram

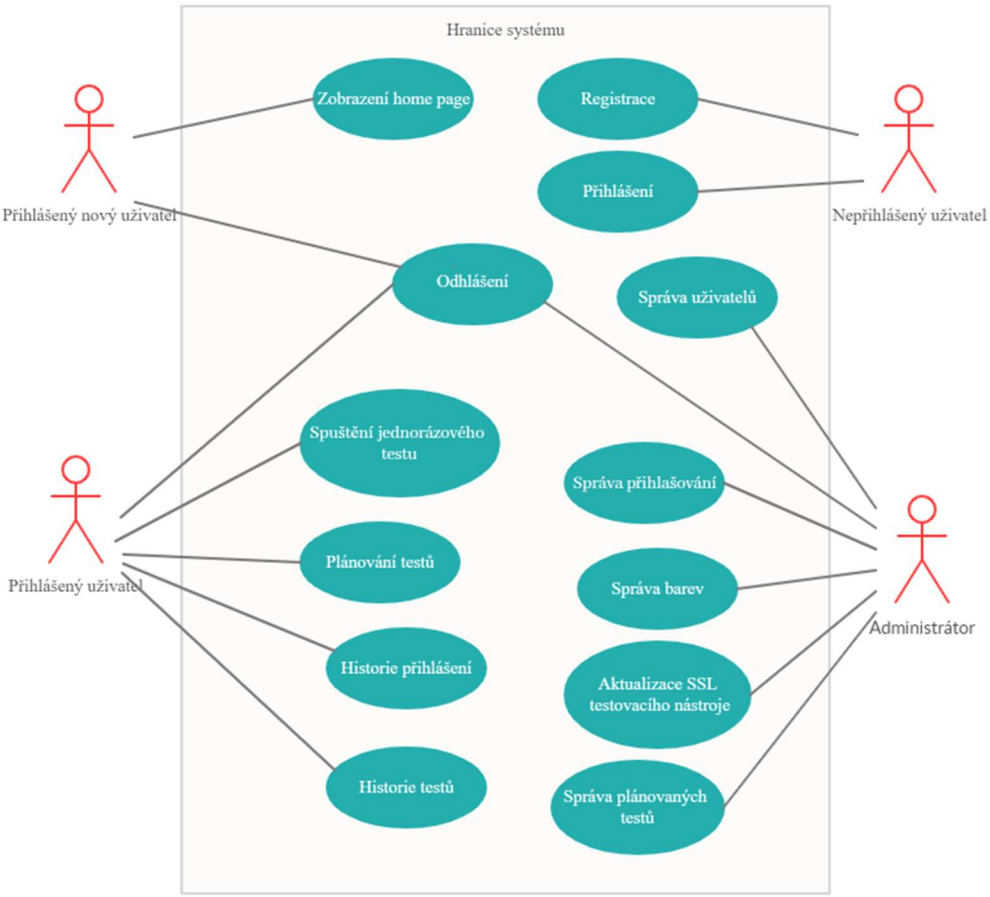
Tato aplikace bude mít implementované role, každá role bude mít jiná práva.

Přihlášený nový uživatel bude mít možnost zobrazení domovské stránky, která ho bude informovat o možnosti používání aplikace až po potvrzení nového uživatele administrátorem systému.

Po schválení nového uživatele administrátorem, dostane tento účet roli uživatele. Uživatel má možnost spouštět jednorázové testy, plánovat testy a zobrazit si historii testů a historii přihlášení.

Administrátor má přístup do správy uživatelů, kde může aktivovat účet, smazat účet a vytvořit nový administrátorský účet. Dále má přistup do správy barev, kde může změnit barvy, kterými jsou zvýrazněny různě závažné hrozby. Administrátor má také možnost aktualizovat Git repositář testovacího nástroje, čímž udržuje testy aktuální. Rovněž má možnost spravovat plánované testy, odebírat je a měnit.

Uml diagram byl vytvořen pomocí online nástroje creately. [21]

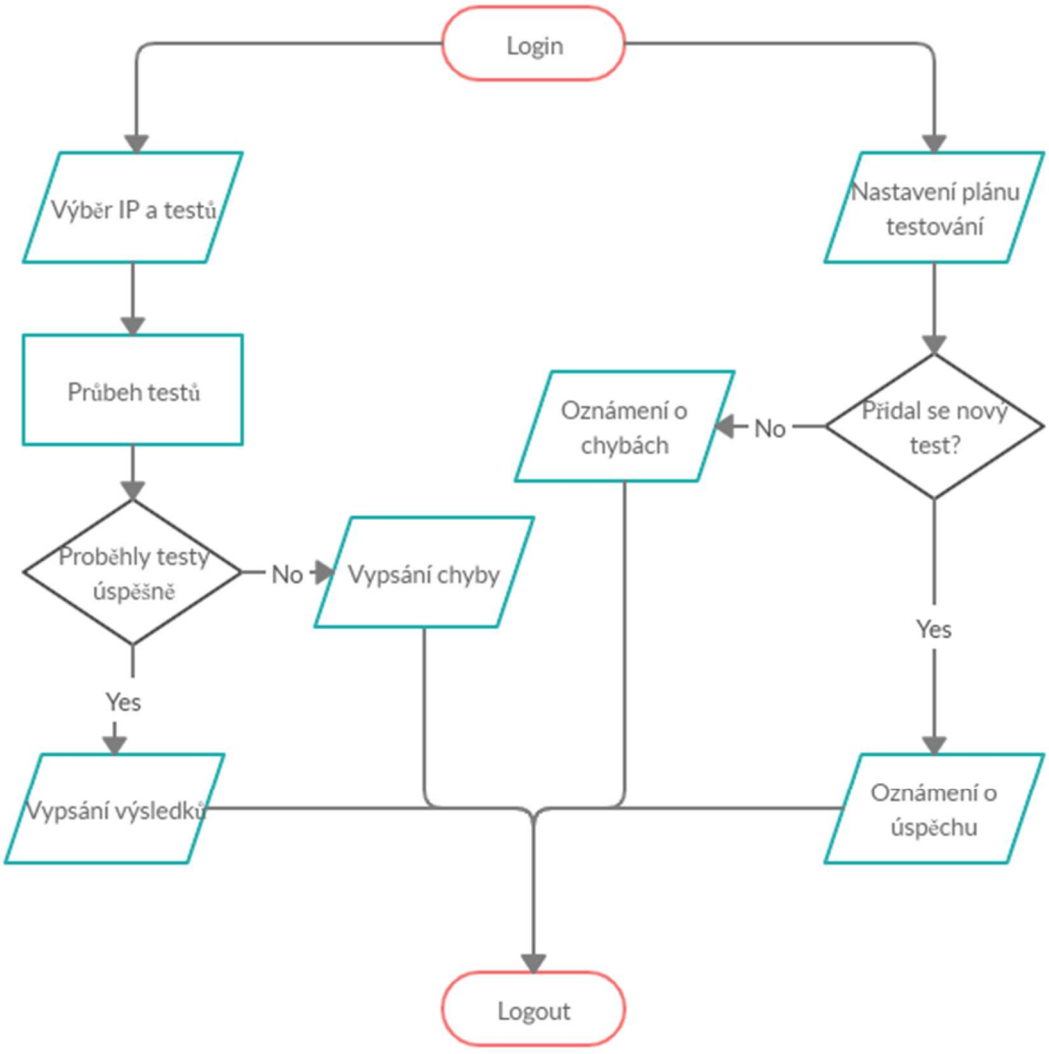


Obrázek 1: UML use case diagram

4.1.3 Diagram aktivit

Diagram aktivit popisuje, jak aktivity přihlášeného uživatele budou probíhat. Jedná se o aktivitu spuštění jednorázového testu, kdy uživatel po zadání požadované IP adresy k testování čeká na výsledky jednotlivých testů. Uživatel si může nastavit, s jakou frekvencí se budou jednotlivé testy spouštět a na jaký email mu budou zasílány výsledky.

Diagram aktivit byl vytvořen pomocí online nástroje creately. [21]



Obrázek 2: Diagram aktivit

II. PRAKTICKÁ ČÁST

5 TVORBA APLIKACE

Tato část se věnuje samotné aplikaci, způsobu jejího návrhu a implementace. Jsou zde uvedeny i části kódů z aplikace k jednotlivým částem.

5.1 Tvorba UI

Prvky z UI jsou vytvořeny pomocí dvou frameworků. Pomocí frameworku Blade [1] a frameworku Vue.js [2]. Pomocí Vue.js je vytvořena stránka pro jednorázové testování, kdy se výsledky postupně přidávají na stránku bez toho, aby bylo potřeba celou stránku překreslovat. Blade byl použit pro vytváření stránek s náhledem plánovaných testů, editací jednotlivých testů a dalších. Každý z těchto frameworků je založen na vytváření jednotlivých komponent nebo šablon pro přehlednější vícenásobné použití.

5.1.1 Šablony

Mezi základní šablony patří přihlašovací formulář, registrační formulář, hlavička, emailová odpověď, oznámení o chybách a úspěšných akcích, formuláře pro vytváření nové plánované úlohy, formuláře pro editaci úloh a indexová obrazovka. Všechny tyto šablony se následně vkládají do jednotlivých stran portálu.

```
<div id="app">
  @include('inc.navbar')
  <main class="py-4">
    <div class="container">
      @include('inc.messages')
      @yield('content')
    </div>
  </main>
</div>
```

Tato část kódu je základní strukturou každé stránky. Zde se vkládá horní lišta a komponenta hlášek, která zobrazuje error a success zprávy. Tělo jednotlivých stránek se uvádí za `@yeld('content')`.

5.1.2 Komponenty

Vue.js framework využívá komponenty, které si uživatel může sám vytvořit jako nové tagy v HTML. V diplomové práci se jedná o tagy `<tests>` které Vue.js nahrazuje za komponentu funkčních testů, jako je odesílání post požadavků, uložení odpovědí do paměti a následné předání těchto dat do dalších komponent, které se starají o vykreslení obsahu, jako jsou komponenty `<presentSH>`, `<missingSH>` a `<testSSL>`. Tyto komponenty data přijmou a překreslí jimi spravovanou oblast.

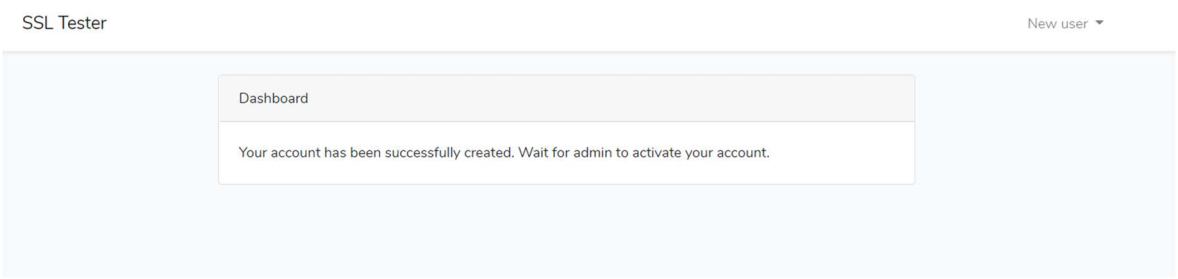
5.1.3 Router

Router přiděluje jednotlivým požadavkům na různé endpointy jejich obsluhu v podobě controlleru. Poté, co prohlížeč odešle serveru nějaký dotaz, controller přiřadí danému dotazu metodu. Tato metoda může obsahovat dotazy do databází nebo kontrolu přístupových práv. Nakonec je odeslána prohlížeči odpověď, touto odpovědí může být přesměrování na jinou stránku, HTML, anebo JSON data.

```
Route::get('/', 'PagesController@index');
```

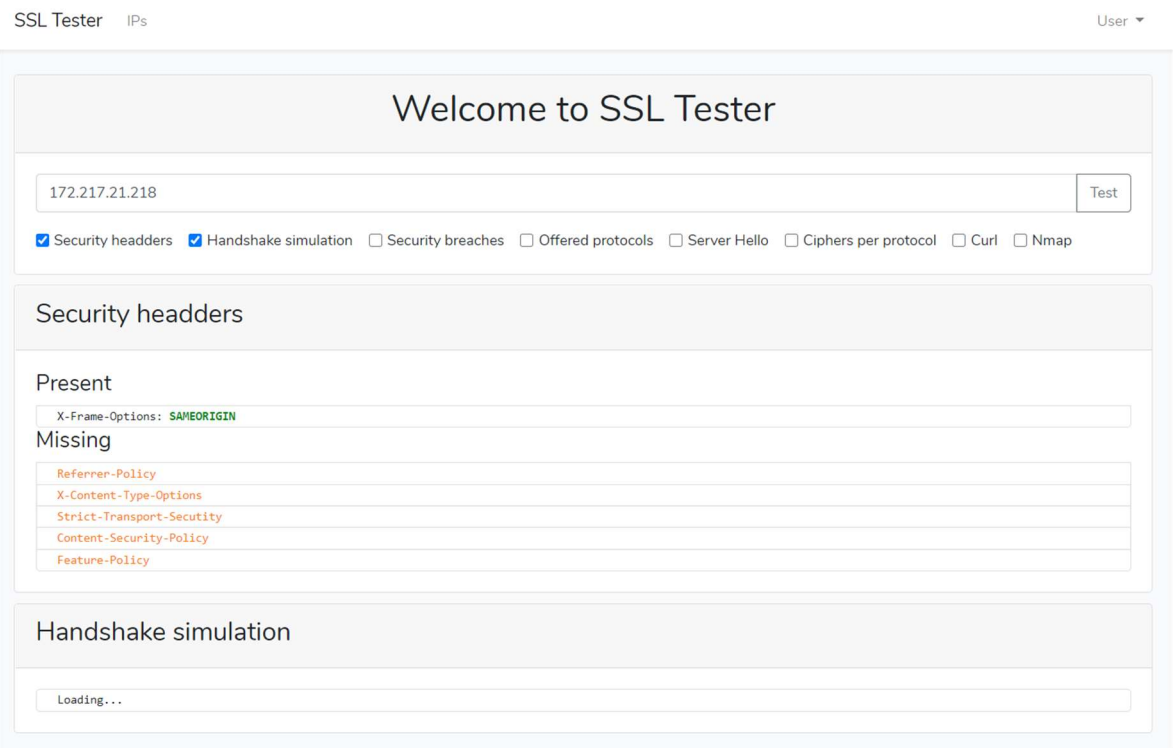
5.2 Úvodní stránka

Úvodní stránka má dvě podoby v závislosti, zda má uživatel práva přístupu či nikoliv. Pokud uživatel nemá přístupová práva úvodní stránka uživatele přesměruje na stránku informující o nutnosti akce admina. Pokud uživatel není přihlášen vůbec, je přesměrován na stránku s přihlášením.



Obrázek 3: Úvodní stránka – nový uživatel

Pokud je uživatel přihlášen a potvrzen administrátorem, má možnost na této stránce spouštět jednorázové testy. Tyto testy se spouštějí nezávisle na sobě. Jak se testy dokončují, stránka postupně zobrazuje výsledky jednotlivých testů.



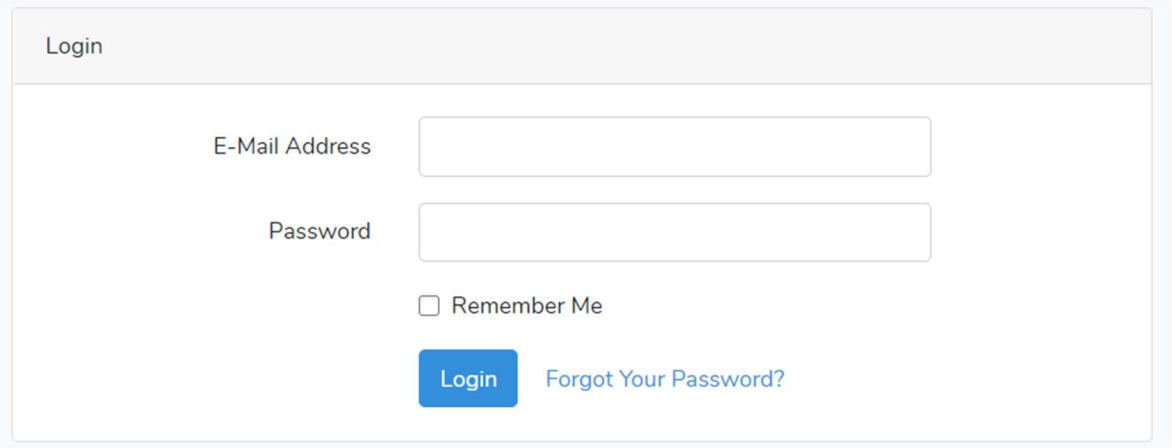
Obrázek 4: Úvodní obrazovka – Přihlášený uživatel

```
@extends('layouts.app')
@section('content')
  @guest
    <h1 align="center">{{$title}}</h1>
  @else
    <div id="tests">
      <tests></tests>
    </div >
  @endguest
@endsection
```

Toto je část kódu, který se stará o zobrazení rozdílných pohledů uživateli v závislosti na tom, zda je uživatel přihlášený či nikoli. Lze vidět i využití nového tagu `<tests>` pro framework Vue.js.

5.3 Stránka přihlášení uživatele

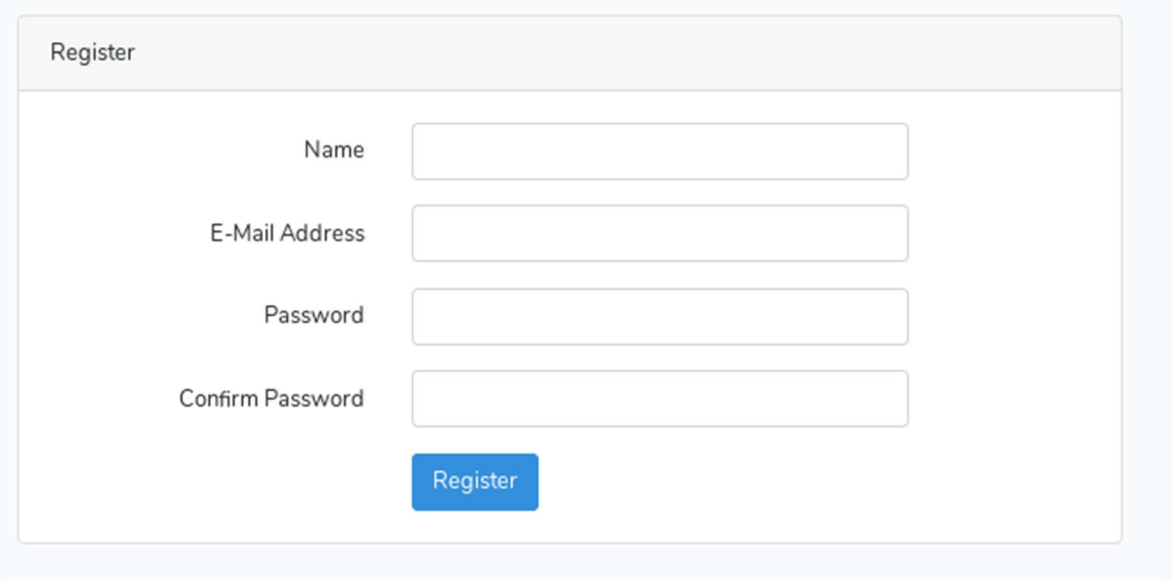
Toto je stránka, která se zobrazí uživateli, pokud se chce přihlásit. Stránka obsahuje validaci vstupů. Pokud uživatel zapomene své přihlašovací údaje má zde možnost zaslat si email pro reset hesla.



Obrázek 5: Stránka přihlášení uživatele

5.4 Stránka registrace uživatele

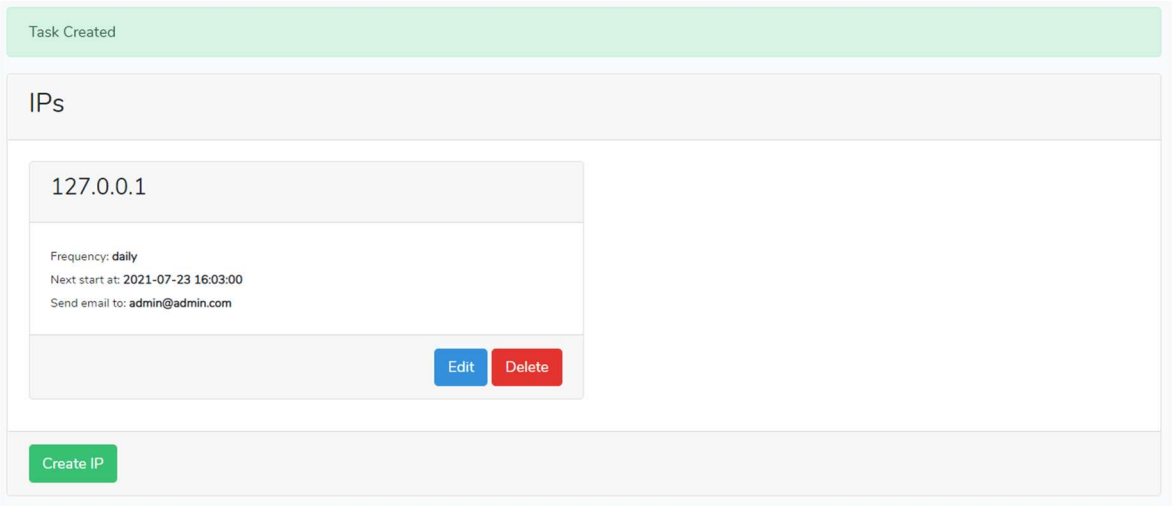
Toto je stránka, která slouží novým uživatelům k vytvoření účtu. Součástí kontroly formuláře je taktéž validace pro emailovou adresu a délku hesla.



Obrázek 6: Stránka registrace uživatele

5.5 Stránka pro plánování testů

Tato stránka slouží přihlášeným uživatelům k vytváření plánovaných úloh. Uživatel zde může přidávat nové plánované úlohy, editovat je či smazat.



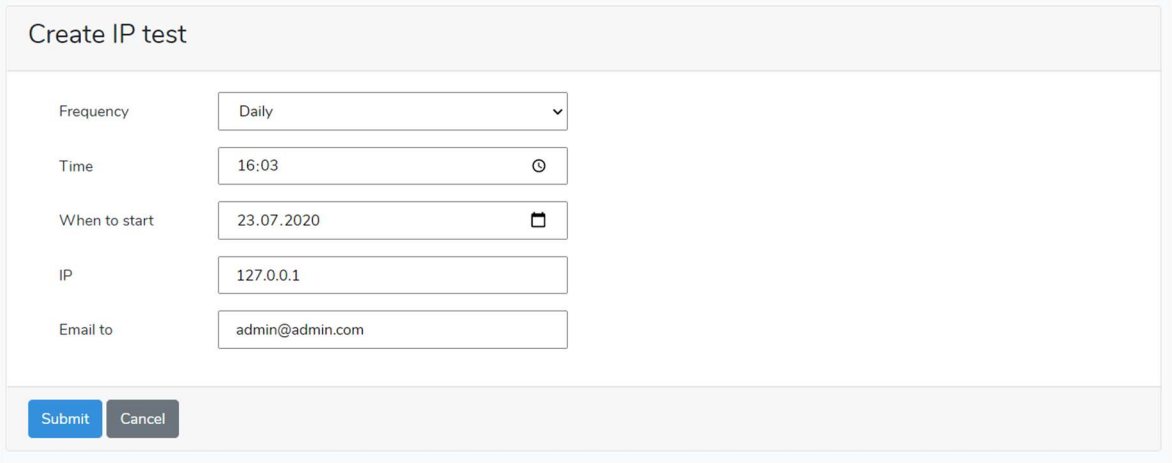
Obrázek 7: Stránka pro plánování testů

```
@extends('layouts.app')
  @section ('content')
  <div class="card card-default">
    <div class="card-header">
      <H3>IPs</H3>
    </div>
    <div class="card-body">
      @if(count($ips)>0)
        @foreach($ips as $ip)
          <div class="card card-default w-50" style="margin-bottom: 10px;">
            <div class="card-header">
              <H4>{{$ip->ip}}</H4>
            </div>
            <div class="card-body">
              <div><small>Frequency: <b>{{$ip->frequency}} </b></small></div>
              <div><small>Next start at: <b>{{$ip->when}} </b></small></div>
              <div><small>Send email to: <b>{{$ip->email}} </b></small></div>
            </div>
            <div class="card-footer d-flex justify-content-end flex-row">
              <a type="button" href="/IPs/{{$ip->id}}/edit" class="btn btn-
primary btn-xs mr-1">Edit</a>
              <form action="{{route('IPs.destroy', $ip->id)}}" method="POST"
class="pull-right">
                @csrf
                {{method_field('DELETE')}}
                <button type="submit" onclick="return confirm('Are you
sure?')"
                class="btn btn-xs btn-danger">Delete</button>
              </form>
            </div>
          </div>
        @endforeach
      @else
        <div class="card card-default w-50" style="margin-bottom: 10px;">
          <div class="card-header"></div>
            <div class="card-body">
              <H5 align="center">No IPs found</H5>
            </div>
```

```
            <div class="card-footer d-flex justify-content-end flex-row">
            </div>
          </div>
        @endif
      </div>
      <div class="card-footer ">
        <a type="button" class="btn btn-success float-left mr-2"
href="/IPs/create">Create IP</a>
          {{$ips->links()}}
    </div>
  </div>
@endsection
```

5.5.1 Vytvoření nového testu

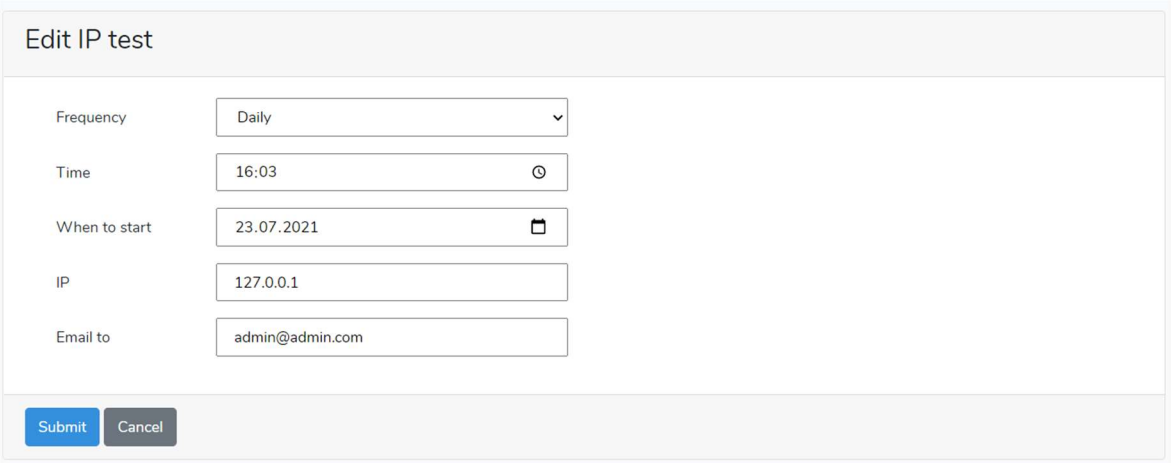
Pro vytváření nových testů slouží tato stránka, na které si uživatel nastavuje pravidelnost testování, čas, kdy se bude plánovaná úloha spouštět a datum, od kterého se začne testovat. Také zde specifikuje IP adresu pro testování a emailovou adresu, na kterou budou chodit automaticky odpovědi s výsledky testů. Formulář je předvyplněn aktuálním datumem a časem, taktéž emailovou adresou přihlášeného uživatele.



Obrázek 8: Stránka pro vytváření testu

5.5.2 Editace nového testu

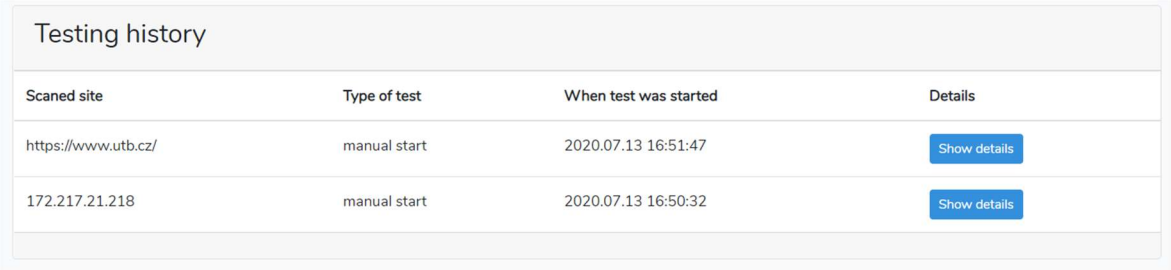
Stránka editace se velmi podobá stránce pro vytváření nového testu. Hlavní rozdíl je v tom, že pole jsou vyplněna podle zadaných hodnot k danému testu.



Obrázek 9: Stránka pro editaci testu

5.6 Stránka pro přehled provedených testů

Tato stránka slouží přihlášeným uživatelům k přehledu všech provedených testů, uživatel zde může prohlížet výsledky testů jak jednorázových, tak i plánovaných.



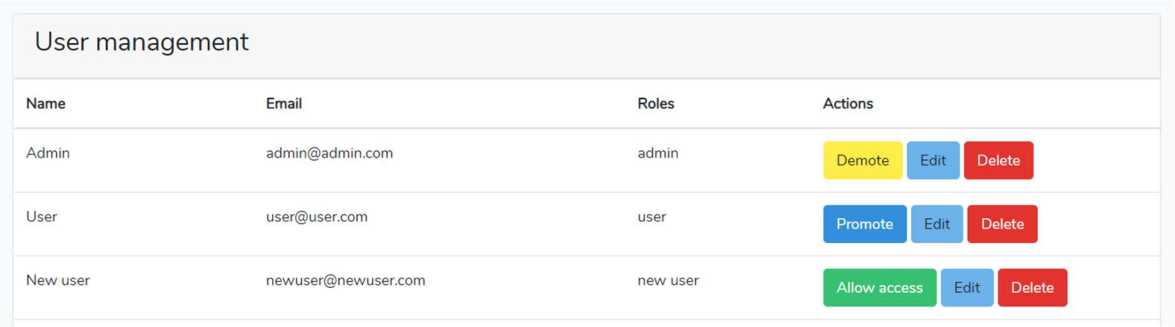
Obrázek 10: Stránka pro přehled provedených testů.

```
@extends('layouts.app')
@section ('content')
  <div class="card card-default">
    <div class="card-header">
      <H3>Testing history</H3>
    </div>
    <table class="table mb-0 table-hover">
      <thead>
        <tr>
          <th scope="col">Scaned site</th>
          <th scope="col">Type of test</th>
          <th scope="col">When test was started</th>
          <th scope="col">Details</th>
        </tr>
      </thead>
      <tbody>
```

```
        @if(count($tests)>0)
          @foreach($tests as $test)
            <tr>
              <td>{{$test->subject}}</td>
              <td>{{$test->type}}</td>
              <td>{{$test->created_at->format('Y.m.d H:i:s')}}</td>
              <td><a href="/tests/{{$test->id}}" type="button" class="btn btn-
primary btn-sm">Show details</a></td>
            </tr>
          @endforeach
        @else
        @endif
      </tbody>
    </table>
    <div class="card-footer">
      {{$tests->links()}}
    </div>
  </div>
@endsection
```

5.7 Stránka pro správu uživatelů

Tato stránka slouží administrátorům k editaci rolí uživatelů systému. Administrátor zde může smazat uživatele, aktivovat účet, přidat administrátorská práva a editovat je.



Obrázek 11: Stránka pro přehled uživatelů

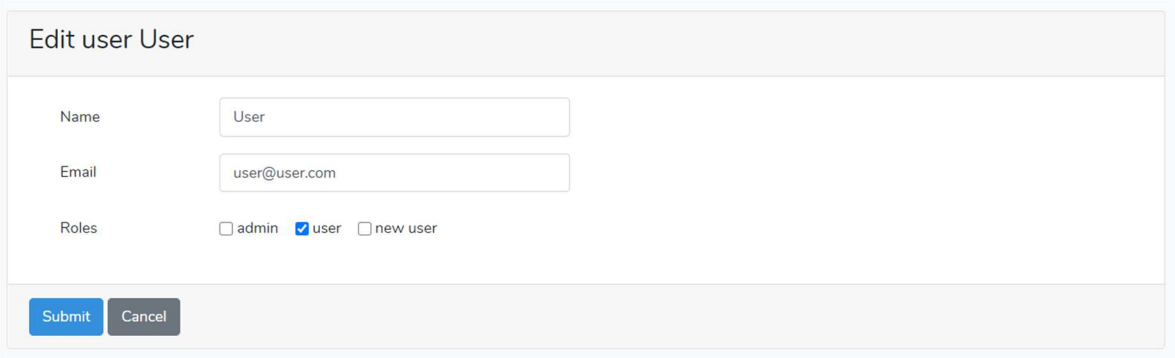
```
@extends('layouts.app')
@section('content')
  <div class="card card-default">
    <div class="card-header">
      <H3>User management</H3>
    </div>
    <table class="table table-hover mb-0">
      <thead>
        <tr>
          <th scope="col">Name</th>
          <th scope="col">Email</th>
          <th scope="col">Roles</th>
          <th scope="col">Actions</th>
        </tr>
      </thead>
      <tbody>
        @foreach ($users as $user)
          <tr>
            <td>{{$user->name }}</td>
            <td>{{$user->email }}</td>
            <td>{{implode(', ', $user->roles()->pluck('name')->toArray()
)}}</td>
```

```
            <td>
              @if ($user->hasRole('new user'))
                <form action="{{route('admin.users.update',$user)}}"
method="POST">
                  @csrf
                  {{method_field('PUT')}}
                  <input id="name" type="hidden" class="form-control
@error('name') is-invalid @enderror" name="name" value="{{ $user->name}}"
required>
                  <input id="email" type="hidden" class="form-control
@error('email') is-invalid @enderror" name="email" value="{{ $user->email}}"
required autocomplete="email">
                  @foreach ($roles as $role)
                    <div class="float-left pr-3">
                      <input style="display: none" class="align-middle"
type="checkbox" name="roles[]" value="{{$role->id}}" @if($role->id==2) checked
@endif>
                      <label style="display: none" class="col-form-label text-
md-right">{{$role->name}}</label>
                    </div>
                  @endforeach
                  <button type="submit" class="float-left mr-1 btn-xs btn btn-
success">Allow access</button>
                </form>
              @else
                @if ($user->hasRole('admin'))
                  <form action="{{route('admin.users.update',$user)}}"
method="POST">
                    @csrf
                    {{method_field('PUT')}}
                    <input id="name" type="hidden" class="form-control
@error('name') is-invalid @enderror" name="name" value="{{ $user->name}}"
required>
                    <input id="email" type="hidden" class="form-control
@error('email') is-invalid @enderror" name="email" value="{{ $user->email}}"
required autocomplete="email">
                    @foreach ($roles as $role)
                      <div class="float-left pr-3">
                        <input style="display: none" class="align-middle"
type="checkbox" name="roles[]" value="{{$role->id}}" @if($role->id==2) checked
@endif>
                        <label style="display: none" class="col-form-label
text-md-right">{{$role->name}}</label>
                      </div>
                    @endforeach
                    <button type="submit" class="float-left mr-1 btn btn-xs
btn-warning">Demote</button>
                  </form>
                  @else
                    <form action="{{route('admin.users.update',$user)}}"
method="POST">
                      @csrf
                      {{method_field('PUT')}}
                      <input id="name" type="hidden" class="form-control
@error('name') is-invalid @enderror" name="name" value="{{ $user->name}}"
required>
                      <input id="email" type="hidden" class="form-control
@error('email') is-invalid @enderror" name="email" value="{{ $user->email}}"
required autocomplete="email">
                      @foreach ($roles as $role)
                        <div class="float-left pr-3">
                          <input style="display: none" class="align-middle"
type="checkbox" name="roles[]" value="{{$role->id}}" @if($role->id!=3) checked
@endif>
                          <label style="display: none" class="col-form-label
text-md-right">{{$role->name}}</label>
                        </div>
```

```
                        @endforeach
                        <button type="submit" class="float-left mr-1 btn btn-xs
btn-primary">Promote</button>
                      </form>
                    @endif
                  @endif
                  @can('edit-users')
                    <a href="{{route('admin.users.edit',$user->id )}}"> <button
type="button" class="float-left mr-1 btn btn-info">Edit</button></a>
                  @endcan
                  @can('delete-users')
                    <form action="{{route('admin.users.destroy',$user->id)}}"
method="POST" class="float-left">
                      @csrf
                      {{method_field('DELETE')}}
                      <button type="submit" onclick="return confirm('Are you
sure?')"
                      class="btn btn-xs btn-danger">Delete</button>
                    </form>
                  @endcan
                </td>
              </tr>
        @endforeach
      </tbody>
    </table>
    <div class="card-footer ">
      {{$users->links()}}
    </div>
  </div>
@endsection
```

5.7.1 Stránka pro editaci uživatele

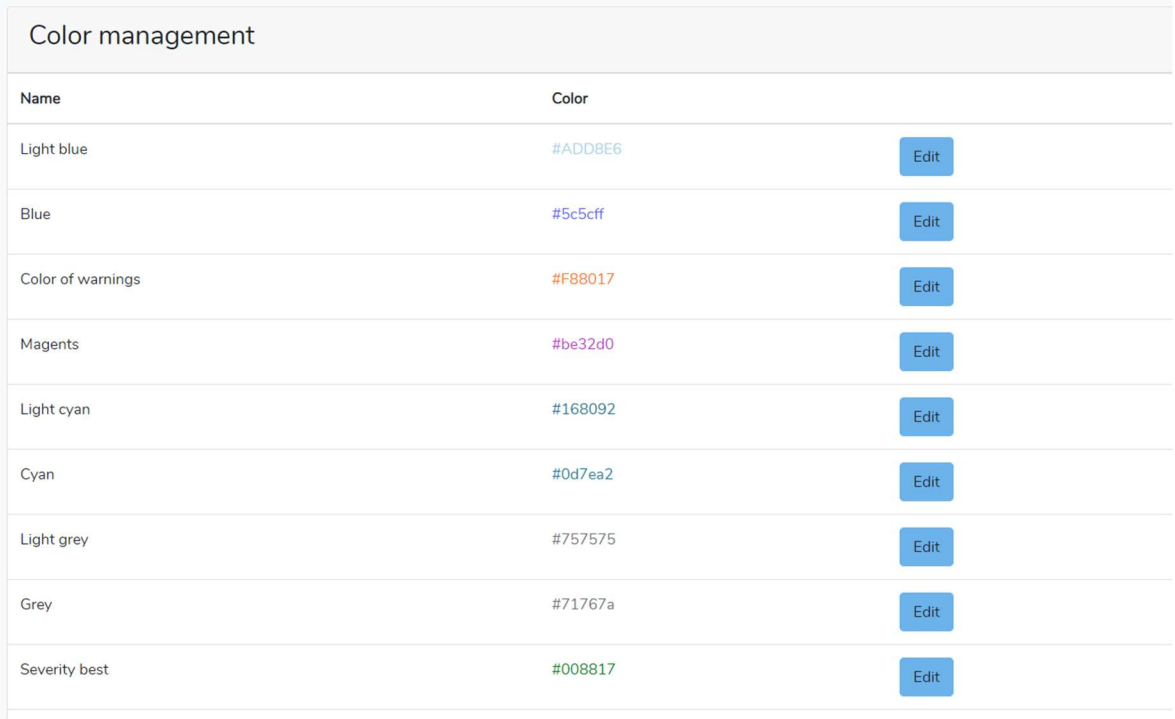
Tato stránka slouží administrátorovi k podrobnější editaci uživatele. Administrátor má možnost změnit jméno, emailovou adresu anebo manuálně přiřadit role.



Obrázek 12: Stránka pro editaci uživatele

5.8 Stránka pro editaci barev

Tato stránka slouží administrátorům systému k editaci barev ve výsledcích testů. Jsou zde definovány barvy, které mají terminálové aplikace ve svých kódech jako možné typy výstupů.



Obrázek 13: Stránka pro editaci barev

```
@extends('layouts.app')
@section('content')
  <div class="card card-default">
    <div class="card-header">
      <H3>Color management</H3>
    </div>
    <table class="table table-hover mb-0">
      <thead>
        <tr>
          <th>Name</th>
          <th>Color</th>
          <th> </th>
        </tr>
      </thead>
      <tbody>
        @if (count($colors) > 0)
          @foreach ($colors as $color)
            <tr data-entry-id="{{ $color->id }}">
              <td field-key='name'>{{ $color->description }}</td>
              <td field-key='color' style="color:{{ $color->color }}">{{
$color->color }}</td>
              <td>
                <a href="{{ route('admin.colors.edit',[$color->id]) }}"
class="btn btn-xs btn-info">Edit</a>
              </td>
            </tr>
          @endforeach
```

```
        @else
          <tr>
            <td colspan="3">No entries in table</td>
          </tr>
        @endif
      </tbody>
    </table>
    <div class="card-footer ">
    </div>
  </div>
@endsection
```

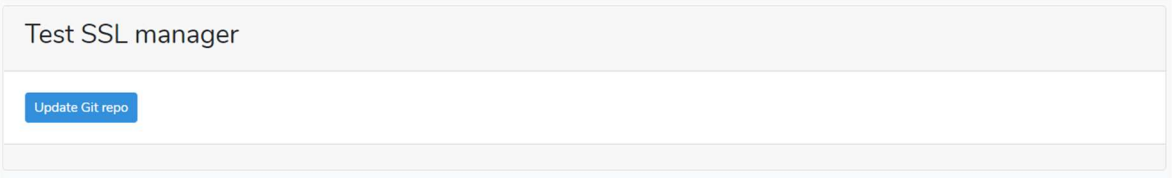
5.8.1 Stránka pro editaci barvy

Na této stránce administrátor může měnit hexadecimální hodnotu pro vybranou barvu, tato nově zvolená barva bude dále používána k vykreslování výsledků testů.

Obrázek 14: Stránka pro editaci barvy

5.9 Stránka pro aktualizaci Git repositáře

Na této stránce má administrátor možnost jedním kliknutím stáhnout novou verzi testovacího nástroje, čímž se udržuje aktuálnost výsledků testů.



Obrázek 15: Stránka pro aktualizaci Git repositáře

5.10 Databáze

Spuštěním migrací se vytvoří několik databázových tabulek v databázi. Jedná se o tabulky pro neúspěšné úkoly, tabulku pro plánované testy, tabulku pro frontu, tabulku migrací, tabulku pro umožnění obnovení hesla, tabulku obsahující uživatele, tabulky pro ukládání výsledků testů, tabulku barev, tabulky pro uživatelské role a tabulku pro historii přihlašování.

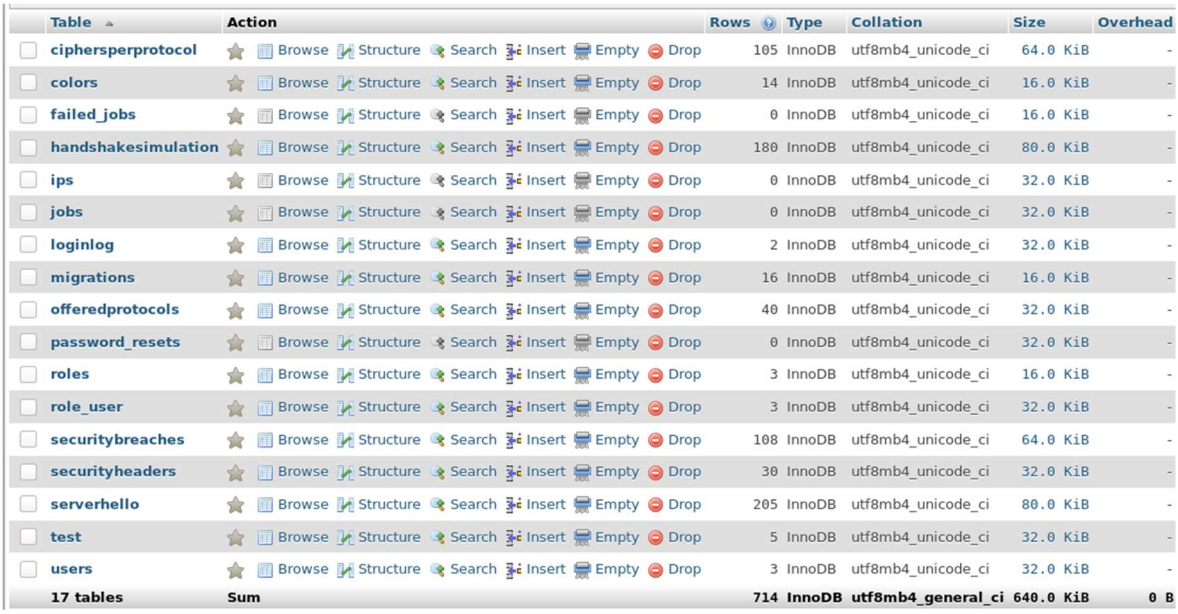


Table	Action	Rows	Type	Collation	Size	Overhead
ciphersperprotocol	Browse Structure Search Insert Empty Drop	105	InnoDB	utf8mb4_unicode_ci	64.0 KiB	-
colors	Browse Structure Search Insert Empty Drop	14	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
handshakesimulation	Browse Structure Search Insert Empty Drop	180	InnoDB	utf8mb4_unicode_ci	80.0 KiB	-
ips	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
loginlog	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
migrations	Browse Structure Search Insert Empty Drop	16	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
offeredprotocols	Browse Structure Search Insert Empty Drop	40	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
password_resets	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
roles	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
role_user	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
securitybreaches	Browse Structure Search Insert Empty Drop	108	InnoDB	utf8mb4_unicode_ci	64.0 KiB	-
securityheaders	Browse Structure Search Insert Empty Drop	30	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
serverhello	Browse Structure Search Insert Empty Drop	205	InnoDB	utf8mb4_unicode_ci	80.0 KiB	-
test	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
users	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
17 tables	Sum	714	InnoDB	utf8mb4_general_ci	640.0 KiB	0 B

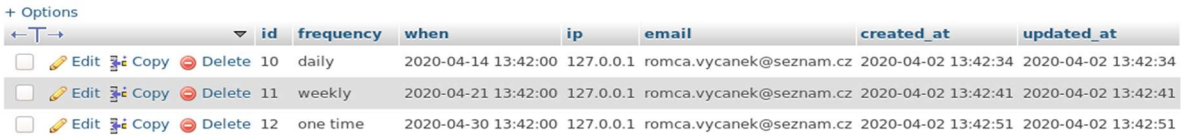
Obrázek 16: Tabulky databáze

5.10.1 Tabulky failed_jobs a jobs

V těchto tabulkách jsou seřazeny jednotlivé úlohy pro asynchronně vykonávané plánované testy. Tyto záznamy zde přiřazuje `TestListener` a následně worker `php artisan queue:listen` spouští jednotlivé testy. Pokud se při obsluze vyskytne nějaký problém, worker o tomto testu vytvoří záznam v tabulce `failed_jobs`.

5.10.2 Tabulka ips

V této tabulce jsou uloženy všechny plánované testy. Pro každý test se ukládá frekvence, s kterou se má test pouštět, čas, kdy se má test pouštět, testovací IP adresa, email, na který se budou posílat výsledky testování a časové značky, kdy byl test naposledy modifikován a kdy byl vytvořen.



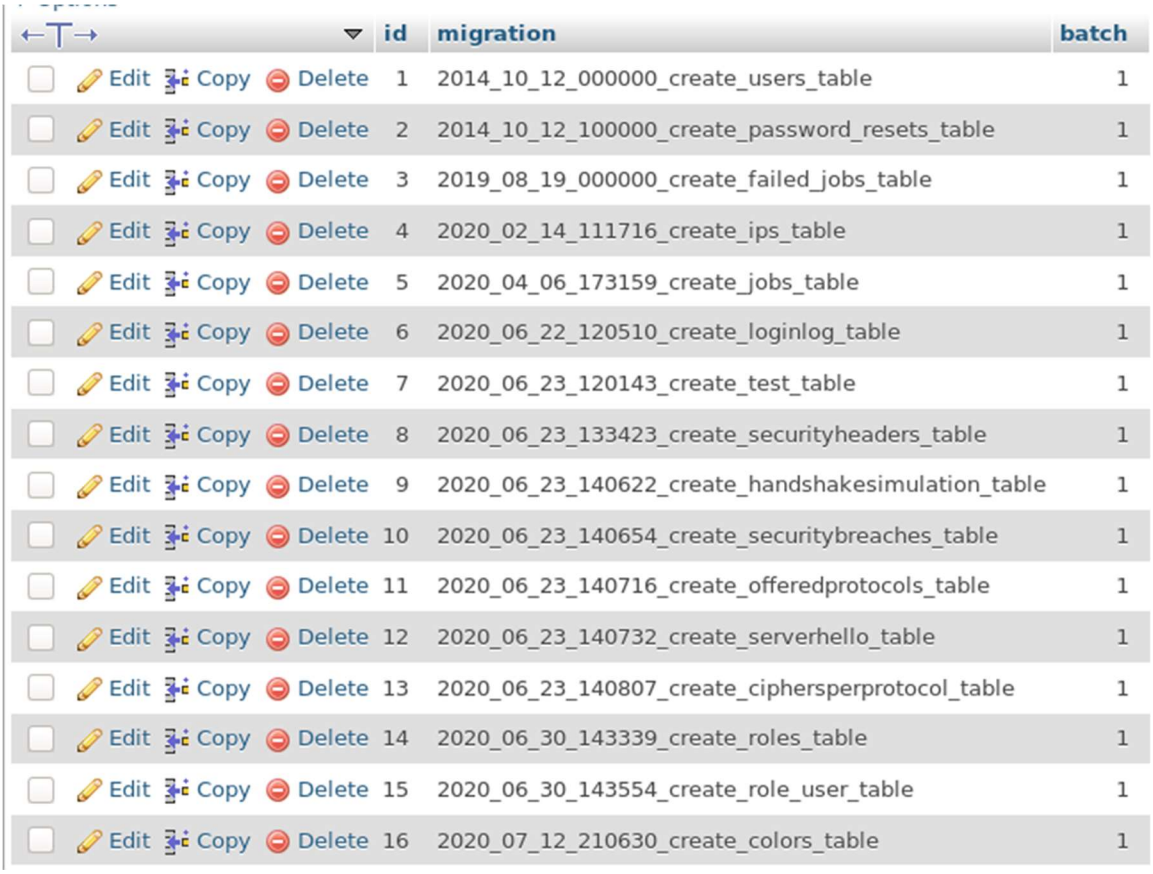
+ Options

	id	frequency	when	ip	email	created_at	updated_at
Edit Copy Delete	10	daily	2020-04-14 13:42:00	127.0.0.1	romca.vycanek@seznam.cz	2020-04-02 13:42:34	2020-04-02 13:42:34
Edit Copy Delete	11	weekly	2020-04-21 13:42:00	127.0.0.1	romca.vycanek@seznam.cz	2020-04-02 13:42:41	2020-04-02 13:42:41
Edit Copy Delete	12	one time	2020-04-30 13:42:00	127.0.0.1	romca.vycanek@seznam.cz	2020-04-02 13:42:51	2020-04-02 13:42:51

Obrázek 17: Tabulka ips

5.10.3 Tabulka migrations

Tuto tabulku používá Laravel pro přehled, které migrace již provedl nad databází, aby ji nepřepisoval, pokud byla daná migrace již jednou provedena.



	id	migration	batch
Edit Copy Delete	1	2014_10_12_000000_create_users_table	1
Edit Copy Delete	2	2014_10_12_100000_create_password_resets_table	1
Edit Copy Delete	3	2019_08_19_000000_create_failed_jobs_table	1
Edit Copy Delete	4	2020_02_14_111716_create_ips_table	1
Edit Copy Delete	5	2020_04_06_173159_create_jobs_table	1
Edit Copy Delete	6	2020_06_22_120510_create_loginlog_table	1
Edit Copy Delete	7	2020_06_23_120143_create_test_table	1
Edit Copy Delete	8	2020_06_23_133423_create_securityheaders_table	1
Edit Copy Delete	9	2020_06_23_140622_create_handshakesimulation_table	1
Edit Copy Delete	10	2020_06_23_140654_create_securitybreaches_table	1
Edit Copy Delete	11	2020_06_23_140716_create_offeredprotocols_table	1
Edit Copy Delete	12	2020_06_23_140732_create_serverhello_table	1
Edit Copy Delete	13	2020_06_23_140807_create_ciphersperprotocol_table	1
Edit Copy Delete	14	2020_06_30_143339_create_roles_table	1
Edit Copy Delete	15	2020_06_30_143554_create_role_user_table	1
Edit Copy Delete	16	2020_07_12_210630_create_colors_table	1

Obrázek 18: Tabulka migrací

5.10.4 Tabulka password_resets

Tato tabulka slouží pro možnost obnovy zapomenutého hesla. Obsahuje jedinečný token, emailovou adresu a časovou známku vytvoření požadavku pro změnu hesla.

+ Options

email	token	created_at
romca.vycanek@seznam.cz	$2y$10$UXuUERe3K.varWJRmovCq.KFxIH.kIoxfks78rjBB38...	2020-04-02 12:36:02

Obrázek 19: Tabulka pro obnovu zapomenutého hesla

5.10.5 Tabulka users

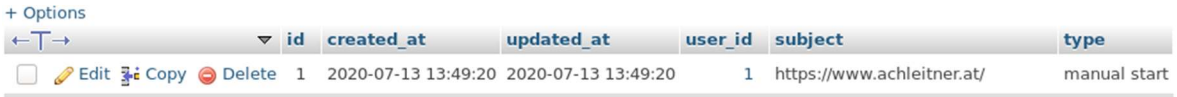
Tato tabulka slouží pro ukládání uživatelských účtů, obsahuje jméno uživatele, jeho email, heslo v hashované podobě, časovou známku, kdy byly uživatelské údaje změněny a kdy byl daný uživatel vytvořen. Další položky jsou vygenerované Laravelem, kdyby bylo potřeba někdy implementovat do aplikace ověření emailové adresy uživatele.

	id	name	email	email_verified_at	password	remember_token	created_at	updated_at
Delete	1	roman	pokus@pokus.pokus	NULL	$2y$10$ZATgdzH5hGZCAv/Gi1Whs.kHXj8Z.LoBZyY.nMQWxB....	NULL	2020-03-19 17:50:54	2020-03-19 17:50:54
Delete	2	pokusnik	pokusnik@pokusnik.com	NULL	$2y$10$CnM1XGqLIpBFJAfPju9g7OTbYuEHNtHvg6mOW3MPBY7...	NULL	2020-03-20 13:42:56	2020-03-20 13:42:56
Delete	3	Roman	romca.vycanek@seznam.cz	NULL	$2y$10$QffB8b8ZzIrG4AQJnFxdGuS0QqfVIqf6bJmiw7N5FYk...	NULL	2020-04-02 12:35:49	2020-04-02 12:35:49

Obrázek 20: Tabulka registrovaných uživatelů

5.10.6 Tabulky pro ukládání historie výsledů testů.

Tabulky `ciphersperprotocol`, `handshakesimulation`, `offeredprotocols`, `securitybreaches`, `securityheaders`, `serverhello` a tabulka `test` slouží pro ukládání výsledků testů. Tabulka `test` slouží jako spojovací tabulka, ve které je uloženo, kdy byl test spuštěn a o jaký typ testu se jednalo, kdo tento test spouštěl a který portál byl testován. V ostatních tabulkách jsou ukládány řádky z každého testu.



+ Options

	id	created_at	updated_at	user_id	subject	type
Edit Copy Delete	1	2020-07-13 13:49:20	2020-07-13 13:49:20	1	https://www.achleitner.at/	manual start

Obrázek 21: Tabulka testů

5.10.7 Tabulka roles a role_user

Tabulka `roles` definuje role v systému. Základními rolemi je admin, user a new user, tyto role jsou následně přiřazovány vztahem many-to-many v tabulce `role_user`.

+ Options

	id	name	created_at	updated_at
Edit Copy Delete	1	admin	2020-07-13 13:44:46	2020-07-13 13:44:46
Edit Copy Delete	2	user	2020-07-13 13:44:46	2020-07-13 13:44:46
Edit Copy Delete	3	new user	2020-07-13 13:44:46	2020-07-13 13:44:46

Obrázek 22: Tabulka rolí

5.10.8 Tabulka colors

Tato tabulka slouží pro doplnění barev testům. Tyto barvy mohou následně uživatelé s rolí admin upravovat. V tabulce je uložen vždy název barvy a poté hexadecimální tvar dané barvy a popis k jednotlivým barvám.

	id	name	color	description	created_at	updated_at
Edit Copy Delete	1	LiteBlue	#ADD8E6	Light blue	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	2	Blue	#5c5cff	Blue	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	3	Warning	#F88017	Color of warnings	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	4	Magenta	#be32d0	Magents	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	5	LiteCyan	#168092	Light cyan	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	6	Cyan	#0d7ea2	Cyan	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	7	LiteGrey	#757575	Light grey	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	8	Grey	#71767a	Grey	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	9	SvrtyBest	#008817	Severity best	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	10	SvrtyGood	#008817	Severity good	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	11	SvrtyLow	#a86437	Severity low	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	12	SvrtyMedium	#F88017	Severity medium	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	13	SvrtyHigh	#FF0000	Severity high	2020-07-22 15:53:46	2020-07-22 15:53:46
Edit Copy Delete	14	SvrtyCritical	#FF0000	Severity critical	2020-07-22 15:53:46	2020-07-22 15:53:46

Obrázek 23: Tabulka barev

5.11 Plánovač úloh

Plánované úlohy se v Laravelu plánují v souboru `app/Console/kernel.php`. Zde se nastavuje, jak často se má daný script spouštět. Spouštění tohoto scriptu se provádí každou minutu.

```
protected function schedule(Schedule $schedule)
{
  $schedule->command('command:runPlannedTests')->everyMinute();
}
```

Následně je vytvořený script, který obsluhuje toto volání. Tento script otestuje, zda je v databázi nějaký plánovaný test starší, než je aktuální čas, a pokud ano, tak tomuto testu upraví datum podle toho, jak často se má test opakovat a vyvolá event `runTestEvent`, na který čeká `TestListener`.

```
public function handle()
{
  $IPs= IP::where('when','<', Carbon::now())->get();
  foreach ($IPs as $IP) {
    if(!strcmp($IP->frequency,'daily')){
      while($IP->when < Carbon::now()){
        $IP->when=Carbon::parse($IP->when)->addDays(1);
      }
      $IP->save();
      event(new runTestsEvent($IP->email,$IP->ip,$IP->user_id));
    }
    if(!strcmp($IP->frequency,'weekly')){
      while($IP->when < Carbon::now()){
        $IP->when=Carbon::parse($IP->when)->addDays(7);
      }
      $IP->save();
      event(new runTestsEvent($IP->email,$IP->ip,$IP->user_id));
    }
    if(!strcmp($IP->frequency,'one time')){
      event(new runTestsEvent($IP->email,$IP->ip,$IP->user_id));
      $IP->delete();
    }
  }
}
```

5.12 Asynchronní spouštění testů

Protože plánované testy jsou spouštěny pomocí systémového nástroje Cron, který volá `php artisan schedule:run`, je potřeba, aby obsluha tohoto volání byla velmi krátká. Proto plánovač nečeká na dokončení testů a jen vyvolá event `runTestEvent`. `TestListener`, který čeká na tento event, rozšiřuje třídu `ShouldQueue`, která tomuto eventu přidává možnost zařadit se do fronty a čekat na obsloužení jiným procesem. I kdyby bylo více testů, které se mají spustit ve stejný okamžik, postupně se zařadí do fronty ke zpracování a budou obslouženy procesem `php artisan queue:listen`. S tímto příkazem se musí použít `--timeout=300`, kterým je potřeba nastavit delší timeout, protože průběh testování jednotlivých

portálů je časově náročný a bez tohoto flagu by se testy přidávaly pouze jako záznamy v tabulce `failed_jobs`. Výsledky testů jsou uloženy v databázi a jsou taktéž odeslány jako html emailem uživateli.

5.13 Testování

Pro testování byly použity různé penetrační nástroje. Pro testování bezpečnostních hlaviček byl použit nástroj Security Header Check [56], který po modifikaci vypisuje výsledky s barvami v terminálovém okně. Tyto výsledky jsou přímo ukládány do databáze. Pro další testy byl využit nástroj testssl.sh [57], který má různé možnosti výstupu, ale v jiných výstupech, než je terminálové okno, neobsahuje všechny informace, které získal. Proto se jako výsledky tohoto nástroje také považuje terminálový výstup, který se následně zpracovává z důvodu zachování barev v terminálu a ukládá do databáze. Dále se k testování používá nástroj Curl [58] a Nmap [59]. Výsledky těchto testů jsou zobrazeny s minimálním formátováním. Hlavní formátování spočívalo v určení maximální délky textu na řádek a rozdělení dlouhých řádků na kratší.

5.13.1 Test bezpečnostních hlaviček

Metoda pro test bezpečnostních hlaviček obsahuje spuštění testovacího scriptu, následné zpracování, uložení do databáze a odeslání formátovaných dat webové stránce.

```
public function runTest(Int $testId)
{
  $test=Test::find($testId);
  $process = new Process(['./shcheck.py', '-g','-d', $test->withHttps()],$cwd =
base_path() . '/app/Http/Controllers');
  $process->setTimeout(0);
  $process->run();
  $terminalResults = explode("\n", $process->getOutput());
  $arrayNoHeadders=array();
  $arrayWithHeadders=array();
  for ($i = 0; $i < count($terminalResults); $i++) {
    if(strlen($terminalResults[$i])>2){
      if (strpos($terminalResults[$i], ':') !== false){
        array_push($arrayWithHeadders, $terminalResults[$i]);
      }else{
        array_push($arrayNoHeadders, $terminalResults[$i]);
      }
    }
  }
  for ($i = 0; $i < count($arrayWithHeadders); $i++) {
    $securityHeaders=new SecurityHeaders;
    $securityHeaders->test_id=$testId;
    $securityHeaders->data=$arrayWithHeadders[$i];
    $securityHeaders->type = true;
    $securityHeaders->save();
  }
  for ($i = 0; $i < count($arrayNoHeadders); $i++) {
    $securityHeaders=new SecurityHeaders;
    $securityHeaders->test_id=$testId;
    $securityHeaders->data=$arrayNoHeadders[$i];
    $securityHeaders->type = false;
    $securityHeaders->save();
  }
  $Styling =new Styling();
  return [ $Styling->TagsToHtml($arrayWithHeadders), $Styling-
>TagsToHtml($arrayNoHeadders)];
}
```

5.13.2 Test možnosti připojení různých platforem

Metoda pro test možnosti připojení různých platforem obsahuje spuštění testovacího scriptu, následné zpracování, uložení do databáze a odeslání formátovaných dat webové stránce.

```
public function runTest(Int $testId)
{
  $test=Test::find($testId);
  $process = new Process(['./testssl.sh', '--client-simulation', '--
quiet'$test->withHttps()],$cwd = base_path() .
'/app/Http/Controllers/testssl.sh');
  $process->setTimeout(0);
  $process->run();
  $Styling =new Styling();
  $terminalResults = explode("\n", $Styling->cmdToTags($process->getOutput()));
  $databaseForm = array();
  for ($i = 0; $i < count($terminalResults); $i++) {
    if(str_contains($terminalResults[$i], 'Testing all')){
      array_push($databaseForm, rtrim($terminalResults[$i]));
      $save= 1;
      for ($l = $i; $l < count($terminalResults); $l++) {
        if(str_contains($terminalResults[$l], 'Start')){
          array_push($databaseForm, rtrim($terminalResults[$l]));
        }
        if(str_contains($terminalResults[$l], 'Running client')){
          $l=$l+1;
          $save= 2;
        }
        if(str_contains($terminalResults[$l], 'Done')){
          $save= 1;
        }
        if($save>1){
          if(strlen(trim($terminalResults[$l]))>8){
            array_push($databaseForm, rtrim($terminalResults[$l]));
          }
        }
      }
      $i = $l;
    }
    else{
      if(str_contains($terminalResults[$i], 'Running client')){
        for ($j = $i+1; $j < count($terminalResults); $j++) {
          if(strlen($terminalResults[$j])>8){
            array_push($databaseForm, rtrim($terminalResults[$j]));
          }
        }
        $i = $j;
      }
      array_pop($databaseForm);
    }
  }
  foreach ($databaseForm as $line) {
    $securityHeaders=new Handshakesimulation;
    $securityHeaders->test_id=$testId;
    $securityHeaders->data=$line;
    $securityHeaders->save();
  }
  return $Styling->TagsToHtml($databaseForm);
}
```

5.13.3 Test zranitelností portálu

Metoda pro test zranitelností portálu obsahuje spuštění testovacího scriptu, následné zpracování, uložení do databáze a odeslání formátovaných dat webové stránce.

```
public function runTest(String $adress, Int $testId)
{
  $test=Test::find($testId);
  $process = new Process(['./testssl.sh', '--vulnerable', '--quiet'$test-
>withHttps()],$cwd = base_path() . '/app/Http/Controllers/testssl.sh');
  $process->setTimeout(0);
  $process->run();
  $Styling =new Styling();
  $terminalResults = explode("\n", $Styling->cmdToTags($process->getOutput()));
  $databaseForm = array();
  for ($i = 0; $i < count($terminalResults); $i++) {
    if(str_contains($terminalResults[$i], 'Testing all')){
      array_push($databaseForm, rtrim($terminalResults[$i]));
      $save= 1;
      for ($l = $i; $l < count($terminalResults); $l++) {
        if(str_contains($terminalResults[$l], 'Start')){
          array_push($databaseForm, rtrim($terminalResults[$l]));
        }
        if(str_contains($terminalResults[$l], 'Testing vulnerabilities')){
          $l=$l+1;
          $save= 2;
        }
        if(str_contains($terminalResults[$l], 'Done')){
          $save= 1;
        }
        if($save>1){
          if(strlen(trim($terminalResults[$l]))>8){
            array_push($databaseForm, rtrim($terminalResults[$l]));
          }
        }
      }
      $i = $l;
    }
    else{
      if(str_contains($terminalResults[$i], 'Testing vulnerabilities')){
        for ($j = $i+1; $j < count($terminalResults); $j++) {
          if(strlen($terminalResults[$j])>8){
            array_push($databaseForm, rtrim($terminalResults[$j]));
          }
        }
        $i = $j;
      }
      array_pop($databaseForm);
    }
  }
  foreach ($databaseForm as $line) {
    $securitybreaches=new Securitybreaches;
    $securitybreaches->test_id=$testId;
    $securitybreaches->data=$line;
    $securitybreaches->save();
  }
  return $Styling->TagsToHtml($databaseForm);
}
```

5.13.4 Test aktivních protokolů

Metoda pro test aktivních protokolů obsahuje spuštění testovacího scriptu, následné zpracování, uložení do databáze a odeslání formátovaných dat webové stránce.

```
public function runTest(String $adress, Int $testId)
{
  $test=Test::find($testId);
  $process = new Process(['./testssl.sh', '--protocols', '--quiet', $test-
>withHttps()],$cwd = base_path() . '/app/Http/Controllers/testssl.sh');
  $process->setTimeout(0);
  $process->run();
  $Styling =new Styling();
  $terminalResults = explode("\n", $Styling->cmdToTags($process->getOutput()));
   $started= 0;
  $databaseForm = array();
  for ($i = 0; $i < count($terminalResults); $i++) {
    if(str_contains($terminalResults[$i], 'Testing all')){
      array_push($databaseForm, rtrim($terminalResults[$i]));
      $save= 1;
      for ($l = $i; $l < count($terminalResults); $l++) {
        if(str_contains($terminalResults[$l], 'Start')){
          array_push($databaseForm, rtrim($terminalResults[$l]));
        }
        if(str_contains($terminalResults[$l], 'Testing protocols')){
          $l=$l+1;
          $save= 2;
        }
        if(str_contains($terminalResults[$l], 'Done')){
          $save= 1;
        }
        if($save>1){
          if(strlen(trim($terminalResults[$l]))>8){
            if(!str_contains($terminalResults[$l], '------------------')){
              array_push($databaseForm, rtrim($terminalResults[$l]));
            }
          }
        }
      }
      $i = $l;
    }
    else{
      if(str_contains($terminalResults[$i], 'Testing')){
        for ($j = $i+1; $j < count($terminalResults); $j++) {
          if(strlen($terminalResults[$j])>8){
            if(!str_contains($terminalResults[$j], '------------------')){
              array_push($databaseForm, rtrim($terminalResults[$j]));
            }
          }
        }
        $i = $j;
      }
      array_pop($databaseForm);
    }
  }
  foreach ($databaseForm as $line) {
    $offeredprotocols=new Offeredprotocols;
    $offeredprotocols->test_id=$testId;
    $offeredprotocols->data=$line;
    $offeredprotocols->save();
  }
  return $Styling->TagsToHtml($databaseForm);
}
```

5.13.5 Test odpovědi serveru a ověření certifikátu pro HTTPS

Metoda pro test odpovědi serveru a ověření certifikátu pro HTTPS pro různé protokoly obsahuje spuštění testovacího scriptu, následné zpracování, uložení do databáze a odeslání formátovaných dat webové stránce.

```
public function runTest(String $adress, Int $testId)
{
  $test=Test::find($testId);
  $process = new Process(['./testssl.sh', '--server-defaults', '--quiet',
$test->withHttps()],$cwd = base_path() . '/app/Http/Controllers/testssl.sh');
  $process->setTimeout(0);
  $process->run();
  $Styling =new Styling();
  $terminalResults = explode("\n", $Styling->cmdToTags($process->getOutput()));
  $databaseForm = array();
  for ($i = 0; $i < count($terminalResults); $i++) {
    if(str_contains($terminalResults[$i], 'Testing all')){
      array_push($databaseForm, rtrim($terminalResults[$i]));
      $save= 1;
      for ($l = $i; $l < count($terminalResults); $l++) {
        if(str_contains($terminalResults[$l], 'Start')){
          array_push($databaseForm, rtrim($terminalResults[$l]));
        }
        if(str_contains($terminalResults[$l], 'Testing server')){
          $l=$l+1;
          $save= 2;
        }
        if(str_contains($terminalResults[$l], 'Done')){
          $save= 1;
        }
        if($save>1){
          if(strlen(trim($terminalResults[$l]))>8){
            array_push($databaseForm, rtrim($terminalResults[$l]));
          }
        }
      }
      $i = $l;
    }
    else{
      if(str_contains($terminalResults[$i], 'Testing server')){
        for ($j = $i+1; $j < count($terminalResults); $j++) {
          if(strlen($terminalResults[$j])>8){
            array_push($databaseForm, rtrim($terminalResults[$j]));
          }
        }
        $i = $j;
      }
      array_pop($databaseForm);
    }
  }
  foreach ($databaseForm as $line) {
    $serverhello=new Serverhello;
    $serverhello->test_id=$testId;
    $serverhello->data=$line;
    $serverhello->save();
  }
  return $Styling->TagsToHtml($databaseForm);
}
```

5.13.6 Test šifrovaných spojení pro různé protokoly

Metoda pro test šifrovaných spojení pro různé protokoly obsahuje spuštění testovacího scriptu, následné zpracování, uložení do databáze a odeslání formátovaných dat webové stránce.

```
public function runTest(String $adress, Int $testId)
{
  $test=Test::find($testId);
  $process = new Process(['./testssl.sh', '--cipher-per-proto', '--quiet','--
color=3',$test->withHttps()],$cwd = base_path() .
'/app/Http/Controllers/testssl.sh');
  $process->setTimeout(0);
  $process->run();
  $Styling =new Styling();
  $terminalResults = explode("\n", $Styling->cmdToTags($process->getOutput()));
  $databaseForm = array();
  for ($i = 0; $i < count($terminalResults); $i++) {
    if(str_contains($terminalResults[$i], 'Testing all')){
      array_push($databaseForm, rtrim($terminalResults[$i]));
      $save= 1;
      for ($l = $i; $l < count($terminalResults); $l++) {
        if(str_contains($terminalResults[$l], 'Start')){
          array_push($databaseForm, rtrim($terminalResults[$l]));
        }
        if(str_contains($terminalResults[$l], 'Testing ciphers')){
          $l=$l+1;
          $save= 2;
        }
        if(str_contains($terminalResults[$l], 'Done')){
          $save= 1;
        }
        if($save>1){
          if(strlen(trim($terminalResults[$l]))>8 ){
            if(!str_contains($terminalResults[$l], '----------------------------
-------')){
              array_push($databaseForm, rtrim($terminalResults[$l]));
            }
          }        }
      }
      $i = $l;
    }
    else{
      if(str_contains($terminalResults[$i], 'Testing')){
        for ($j = $i+1; $j < count($terminalResults); $j++) {
          if(strlen($terminalResults[$j])>8){
            if(!str_contains($terminalResults[$j], '----------------------------
-------')){
              array_push($databaseForm, rtrim($terminalResults[$j]));
            }
          }
        }
        $i = $j;
      }
      array_pop($databaseForm);
    }  }
  foreach ($databaseForm as $line) {
    $ciphersperprotocol=new Ciphersperprotocol;
    $ciphersperprotocol->test_id=$testId;
    $ciphersperprotocol->data=$line;
    $ciphersperprotocol->save();
  }
  return $Styling->TagsToHtml($databaseForm);
}
```

6 IMPLEMENTAČNÍ MANUÁL

V této kapitole je popsán postup, jak nastavit prostředí v systému Debian, tak aby v něm bylo možné spustit testovací portál.

6.1 Instalace MySQL

Pro ukládání dat potřebuje Laravel být spojen s databází. Jednou z nejpoužívanějších databází je MySQL. Pro stažení instalátoru se využije nástroj WGET:

```
wget https://dev.mysql.com/get/mysql-apt-config_0.8.10-1_all.deb
```

Po stažení souboru se tento nástroj nainstaluje pomocí nástroje dpkg příkazem:

```
sudo dpkg -i mysql-apt-config_0.8.13-1_all.deb
```

Při instalaci se zobrazí možnost změny verze databáze spolu s dalšími možnostmi pro instalaci.

Následně se zkontrolují a nainstalují možné další aktualizace konfiguračního programu příkazy:

```
sudo apt update
sudo apt upgrade
```

Nyní je na systém možno naistalovat MySQL server, což se provede příkazem:

```
sudo apt install mysql-server
```

Během instalace uživatel nastaví heslo pro root uživatele.

Nyní je možné vyzkoušet, zda se provedla instalace správně příkazem:

```
sudo systemctl status mysql
```

Tento příkaz ukáže uživateli, pokud vše proběhlo správně:

```
Active: active (running)
```

Nyní, když je MySQL staženo a spuštěno, je potřeba jeho zabezpečení. Tyto bezpečnostní opatření se implementují příkazem:

```
mysql_secure_installation
```

Uživatel má nyní možnost si zvolit, zda databáze bude vyžadovat určitou složitost pro hesla uživatelů, zda chce odstranit z databáze anonymní uživatele a zda chce odstranit ukázkové tabulky.

Nyní se uživatel přihlásí do databáze příkazem a následným zadáním hesla:

```
mysql -u root -p
```

Poté, co je uživatel přihlášen, je potřeba vytvořit databázi, se kterou bude Laravelový projekt pracovat. Tato databáze se vytvoří příkazem:

```
mysql> CREATE DATABASE sslTester;
```

6.2 Instalace PHP

Laravel je postavený na programovacím jazyce PHP, proto je nutné podporu tohoto programovacího jazyku přidat do systému. První se provede instalace příkazy:

```
sudo apt-get install php
sudo apt-get install php7.3-zip
sudo apt-get install php7.3-mbstring
sudo apt-get install php7.3-xml
sudo apt-get install php7.3-mysql
```

Dále je potřeba přidat balíček do php.ini souboru, který se nachází ve složce apache2:

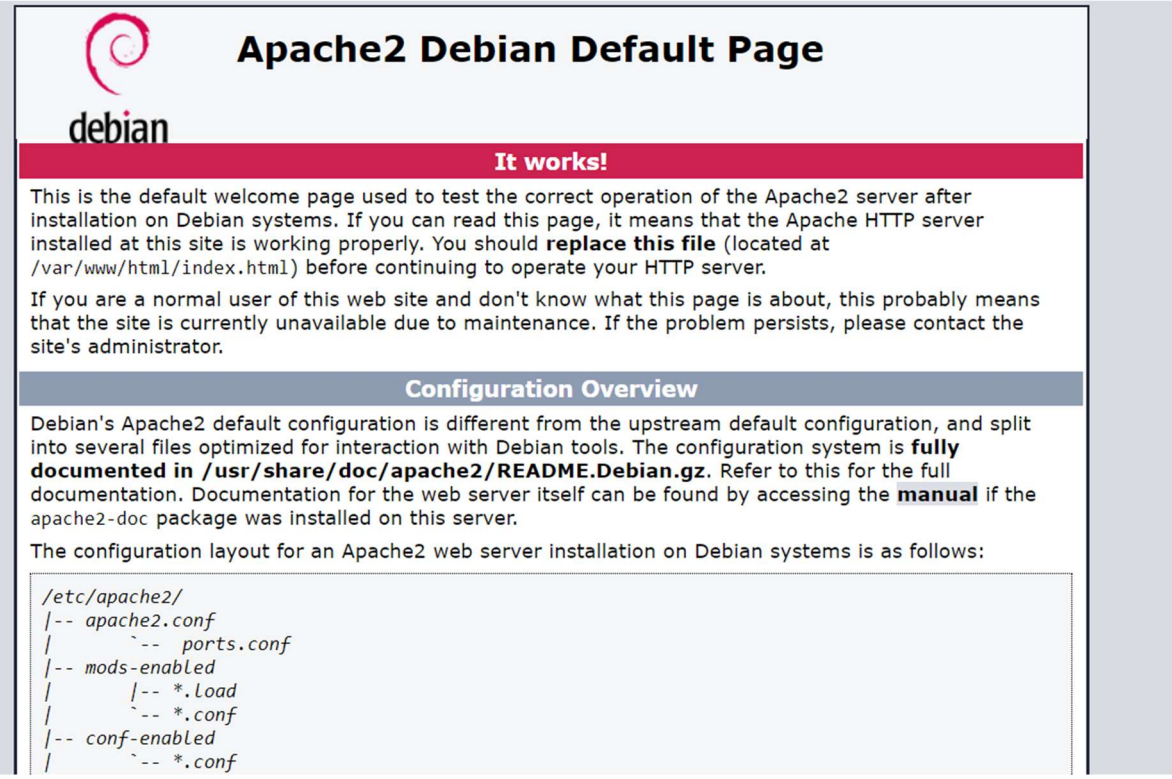
```
/etc/php/7.3/apache2/php.ini
```

A do tohoto souboru přidat záznam:

```
extension=pdo_mysql.so
```

6.3 Nastavení Apache

Po úspěšné instalaci PHP se nainstaluje i Apache, tento nástroj se využívá pro vytvoření webového serveru. Pro ověření správné funkčnosti je možné na adrese `localhost` zobrazit úvodní stránku.



Obrázek 24: Výchozí stránka Apache

Následně je potřeba tento nástroj upravit, aby zobrazoval jinou webovou stránku místo své výchozí. K tomu je potřeba editovat několik souborů.

Soubor s cestou `/etc/apache2/sites-available/000-default.conf` obsahuje nastavení běžících webových stránek. Zde je potřeba přidat záznam s novou webovou stránkou, nastavit jméno nového portálu, a cestu k veřejné složce.

```
<VirtualHost *:80>
        ServerName ssltester.com
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/PortalTester/public
        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
```

```
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Další soubor, který je potřeba modifikovat, je soubor s cestou `etc/apache2/apache2.conf`. Zde je potřeba povolit Override, čímž se zpřístupní odkazy na portálu:

```
<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
</Directory>

<Directory /var/www/PortalTester/public>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
</Directory>
```

Override není defaultně zapnutá volba pro Apache, tato funkce se spustí příkazem:

```
sudo a2enmod rewrite
```

Po těchto úpravách je potřeba Apache server restartovat, aby se změny projevily. To se udělá příkazem:

```
service apache2 restart
```

6.4 Instalace Composer

Composer je nástroj pro správu závislostí v PHP, díky kterému je možné deklarovat knihovny, na kterých projekt závisí a tento nástroj je bude spravovat (instalovat / aktualizovat). Stažení instalátoru se provede příkazem:

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
```

Tento instalátor lze ověřit poskytnutým hashem z GitHubu tímto příkazem:

```
 php     -r     "if     (hash_file('sha384',     'composer-setup.php')     ===
'e0012edf3e80b6978849f5eff0d4b4e4c79ff1609dd1e613307e16318854d24ae64f26d17af3ef
0bf7cfb710ca74755a') { echo 'Installer verified'; } else { echo 'Installer
corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
```

Dále se spustí instalace Composeru příkazem:

```
php composer-setup.php
```

Stažený soubor, kterým se instalace prováděla se může již odstranit, a to se provede tímto příkazem:

```
php -r "unlink('composer-setup.php');"
```

Následně je potřeba kvůli možnosti globálního volání přesunout soubor `composer.phar`. K tomu slouží příkaz:

```
mv composer.phar /usr/local/bin/composer
```

Nyní lze vyzkoušet správné nastavení aplikace Composer příkazem:

```
composer -V
```

6.5 Instalace Node.js

Node je systém navržený pro psaní internetových aplikací, především webových portálů, a proto je taky nutný pro správné nainstalování aplikace. Před samotnou instalací je potřeba stáhnout několik nástrojů, které bude Node využívat. Ty se stáhnou pomocí příkazu:

```
sudo apt-get install git-core curl build-essential openssl libssl-dev python
```

Následná instalace node spočívá v stažení aplikace z GitHubu pomocí příkazu:

```
git clone https://github.com/nodejs/node.git
cd node
```

Poté má uživatel možnost si vybrat, jakou verzi chce nainstalovat. Přehled možných verzí se zobrazí příkazem:

```
git tag
git checkout v13.10.1
```

Následná kompilace a instalace se provede těmito příkazy:

```
./configure
make
sudo make install
```

Pro ověření, zda byla instalace úspěšná, je možno vyzkoušet příkaz:

```
node -v
```

6.6 Instalace NPM

Pro stažení balíčků používá Laravel i Node Package Manager, což je nástroj, který má ve svých repositářích různé doplňky. Stažení aplikace se provede příkazem:

```
curl https://www.npmjs.com/install.sh | sudo sh
```

Pro vyzkoušení funkčnosti npm slouží příkaz:

```
npm -v
```

6.7 Spuštění aplikace Laravel

V této části je popsáno, jak stáhnout aplikaci z Git repositáře, její instalace a následné spuštění.

6.7.1 Stažení aplikace z Gitu

Aplikace je uložená na webovém portálu GitHub. Pro stažení repositáře se využívá aplikace Git, která se nainstaluje příkazem:

```
sudo apt-get install git
```

Dále je nutné vybrat umístění aplikace, ta se následně stáhne do složky příkazem:

```
sudo git clone https://github.com/rovycanek/PortalTester.git
```

Nyní je aplikace stažena.

6.7.2 Instalace aplikace

V adresáři, kde je stažená aplikace z Gitu, je potřeba spustit instalaci pomocí několika příkazů:

```
composer.phar global require laravel/installer
composer install
npm install
npm run production
```

6.7.3 Nastavení proměnných

Po instalaci projektu je nutné nastavit aplikaci spojení s databází a poštovním klientem. Tyto proměnné se nastavují v souboru `.env`, tento soubor se vytvoří zkopírováním již existujícího souboru `.env.example`, a následným přejmenováním.

```
cp .env.example .env
```

Nastavení databáze se provádí v proměnných, začínajících DB_, kdy je důležité jméno databáze DB_DATABASE, přihlašovací jméno DB_USERNAME a heslo DB_PASSWORD. Tyto hodnoty byly vytvořeny při instalaci MySQL:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=SSLTester
DB_USERNAME=root
DB_PASSWORD=
```

Zde se také nastavuje připojení pro poštovního klienta. Toto nastavení se vypisuje pro proměnné začínající MAIL_:

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.utb.cz
MAIL_PORT=25
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=portaltest@ptlab.utb.cz
MAIL_FROM_NAME="${APP_NAME}" MAIL_FROM_NAME="${APP_NAME}"
```

Laravel také šifruje svá přenesená data klíčem, který vytvoříme příkazem:

```
php artisan key:generate
```

6.7.4 Migrace databáze

Když je aplikace správně nainstalována, je potřeba vytvořit jednotlivé tabulky, ve kterých se ukládají záznamy v databázi. Tomuto procesu se v Laravelu říká migrace a provede se příkazem:

```
php artisan migrate
```

6.7.5 Seed databáze

Aplikace má vytvořené tabulky, ale k přihlášení je potřeba potvrzení nového uživatele akcí admina. První admin účet se proto vytvoří tímto příkazem:

```
php artisan db:seed
```

Pro nastavení jiného, než defaultního hesla slouží soubor `/database/seeds/UsersTableSeeder.php`, ve kterém je možno přidat nebo změnit vytvářené uživatele.

```
public function run()
{
  Schema::disableForeignKeyConstraints();
  User::truncate();
  DB::table('role_user')->truncate();
  DB::table('ciphersperprotocol')->truncate();
  DB::table('failed_jobs')->truncate();
  DB::table('handshakesimulation')->truncate();
  DB::table('ips')->truncate();
  DB::table('jobs')->truncate();
  DB::table('loginlog')->truncate();
  DB::table('offeredprotocols')->truncate();
  DB::table('securitybreaches')->truncate();
  DB::table('serverhello')->truncate();
  DB::table('test')->truncate();
  DB::table('securityheaders')->truncate();
  Schema::enableForeignKeyConstraints();
  $adminRole = Role::where('name','admin')->first();
  $userRole = Role::where('name','user')->first();
```

```
    $newuserRole = Role::where('name','new user')->first();
    $admin = User::create([
      'name' => 'Admin',
      'email' => 'admin@admin.com',
      'password' => Hash::make('password'),
    ]);
    $user = User::create([
      'name' => 'User',
      'email' => 'user@user.com',
      'password' => Hash::make('password'),
    ]);
    $newuser = User::create([
      'name' => 'New user',
      'email' => 'newuser@newuser.com',
      'password' => Hash::make('password'),
    ]);
    $admin->roles()->attach($adminRole);
    $user->roles()->attach($userRole);
    $newuser->roles()->attach($newuserRole);
}
```

6.7.6 Zpřístupnění aplikace v internetovém prostředí

V tomto kroku je potřeba přepsat několik konfiguračních souborů Apache v Debianu, z tohoto důvodu je potřeba otevřít textový editor pod právy root příkazem:

```
sudo gedit
```

Následně je nutné přidání do souboru `/etc/hosts` záznamů:

```
127.0.0.1    localhost
127.0.0.1    SSLtester.com
```

V souboru `/opt/lampp/etc/httpd.conf` se includuje soubor vhosts:

```
# Virtual hosts
Include etc/extra/httpd-vhosts.conf
```

A v souboru `/opt/lampp/etc/extra/httpd-vhosts.conf` se přidají virtuální portály:

```
<VirtualHost *:80>
      DocumentRoot „/opt/lampp/htdocs/“
      ServerName localhost
</VirtualHost>

<VirtualHost *:80>
      DocumentRoot „/opt/lampp/htdocs/SSLTester/public“
      ServerName ssltester.com
</VirtualHost>
```

Následně je nutné restartovat apache příkazem:

```
sudo /opt/lampp/lampp restart
```

Pro umožnění fungování linků v aplikaci je potřeba vytvořit soubor `.htaccess` ve kterém se definuje tzv. rewrite adres. V tomto souboru je potřeba nastavit správnou cestu k public adresáři.

```
<IfModule mod_rewrite.c>
    <IfModule mod_negotiation.c>
        Options -MultiViews
    </IfModule>

    RewriteEngine On
    RewriteBase /var/www/SSLTester/public/
    # change above to your site i.e.,  RewriteBase /whatever/public/

    # Redirect Trailing Slashes...
    RewriteRule ^(.*)/$ /$1 [L,R=301]

    # Handle Front Controller...
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteRule ^ index.php [L]
</IfModule>
```

Nyní je aplikace online.

6.7.7 **Automatické spouštění testovaní**

Laravel umožňuje plánovat úlohy přímo v projektu, ale aby mohly být spouštěny, musí se přidat Cron úloha do Linuxového prostředí. Editor Cron úloh se spustí příkazem:

```
crontab -e
```

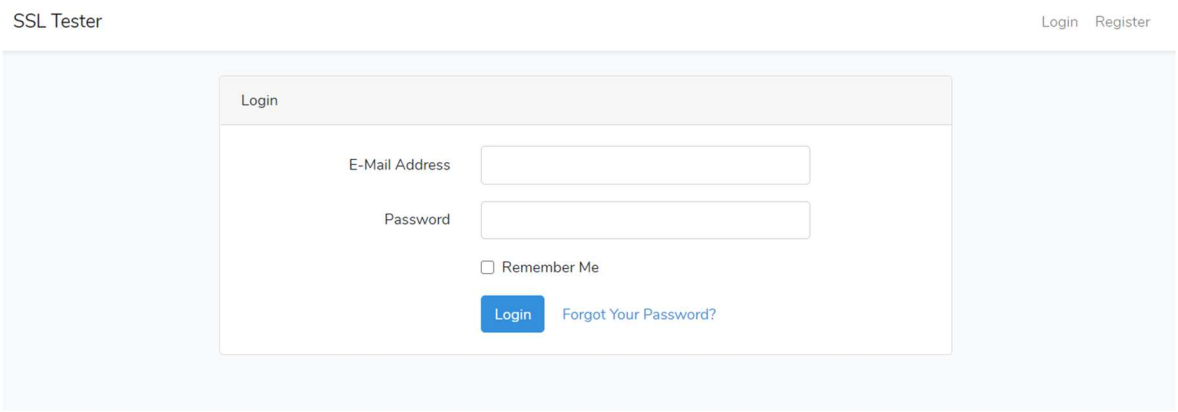
Následně je potřeba přidat záznam, který bude volat artisan plánovač v projektu každou minutu:

```
* * * * * cd /cesta-k-projektu && php artisan schedule:run >> /dev/null 2>&1
```

Aplikace přidává jednotlivé testy automaticky k asynchronnímu zpracování, ale pro samotné spuštění testů je potřeba ještě zapnout démona, který bude jednotlivé testy provádět. Protože testování bude probíhat déle než jednu minutu, je potřeba přidat tomuto démonu i flag, který slouží pro nastavení delšího timeoutu. Ten se spustí příkazem:

```
php artisan queue:listen --timeout=300 &
```

Nyní je aplikace spuštěna a plně funkční.



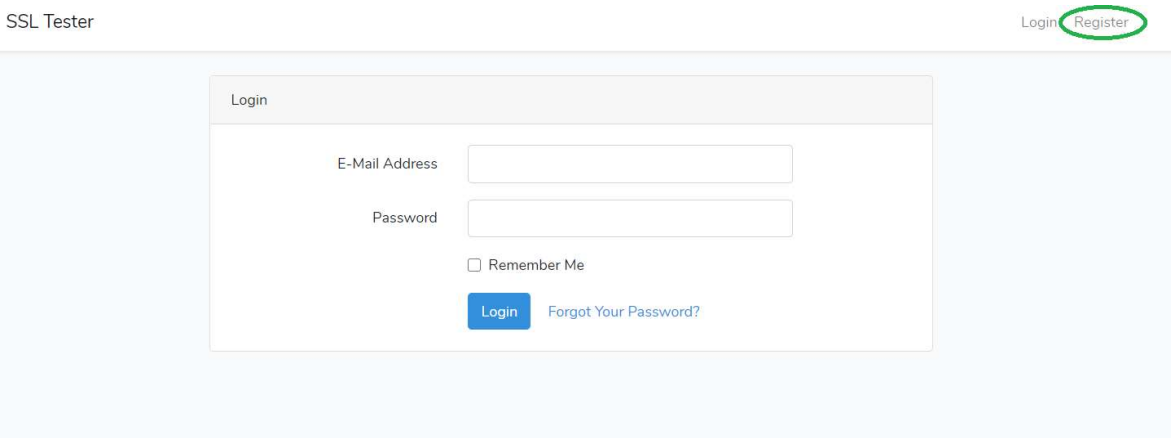
Obrázek 25: Spuštěná aplikace

7 NÁVOD K OBSLUZE

V této části je popsáno, jak ovládat aplikaci SSL Tester. Od vytvoření účtu přes spuštění testů až po plánování jednotlivých testů.

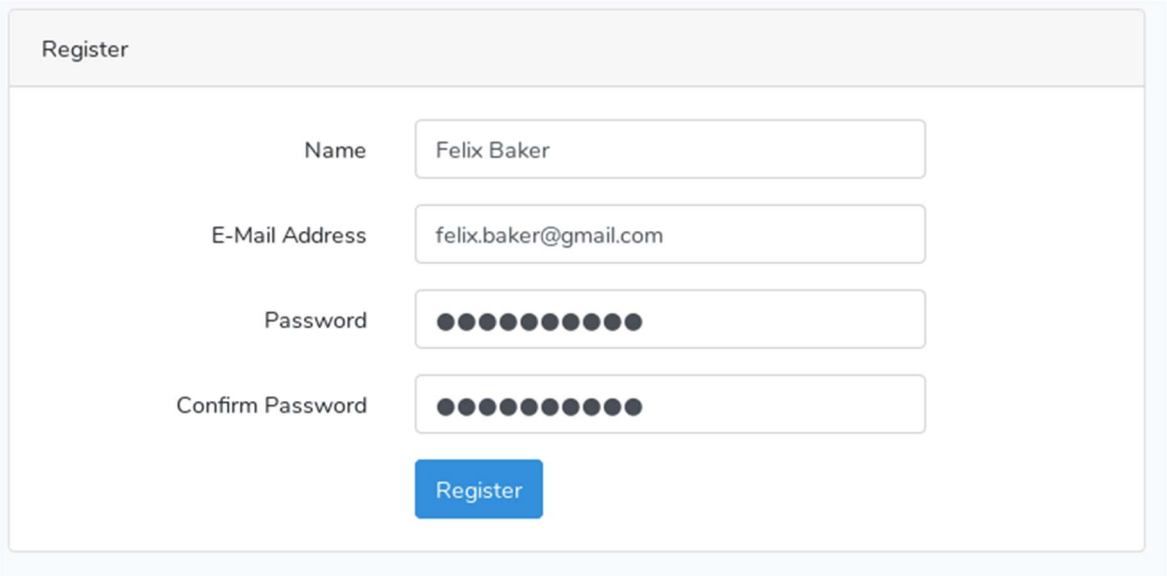
7.1 Registrace uživatele

Pro vytvoření nového uživatele je nutné na úvodní straně vybrat v hlavičce možnost Register.



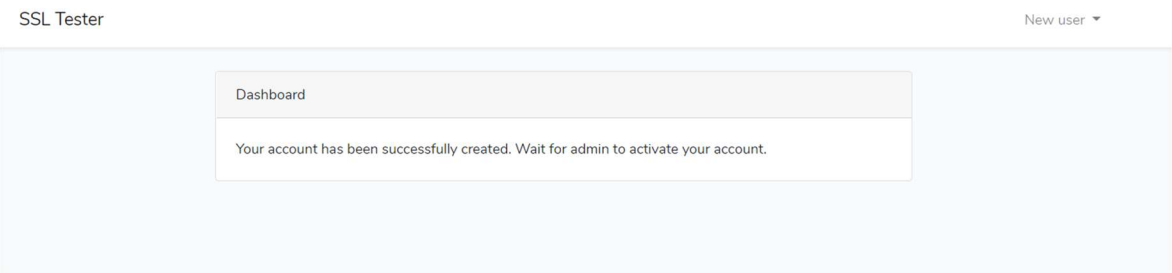
Obrázek 26: Tlačítko registrace

Následně se otevře registrační formulář. Po vyplnění tohoto formuláře se formulář odešle tlačítkem Register.



Obrázek 27: Vyplněný registrační formulář

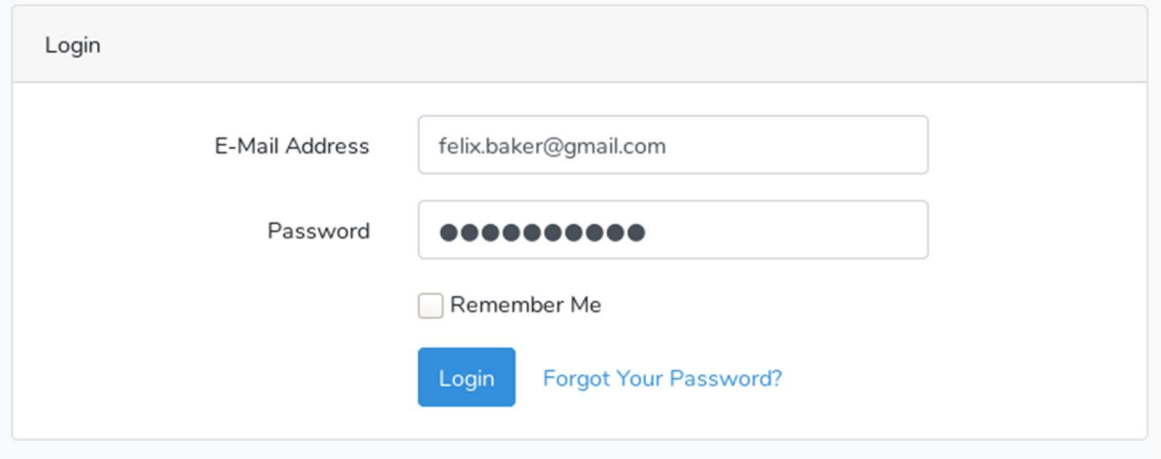
Po úspěšné registraci je nově vytvořený uživatel přesměrován na stránku, která ho informuje o úspěšné registraci a nutnosti potvrzení nového účtu administrátorem.



Obrázek 28: Přihlášení po registraci

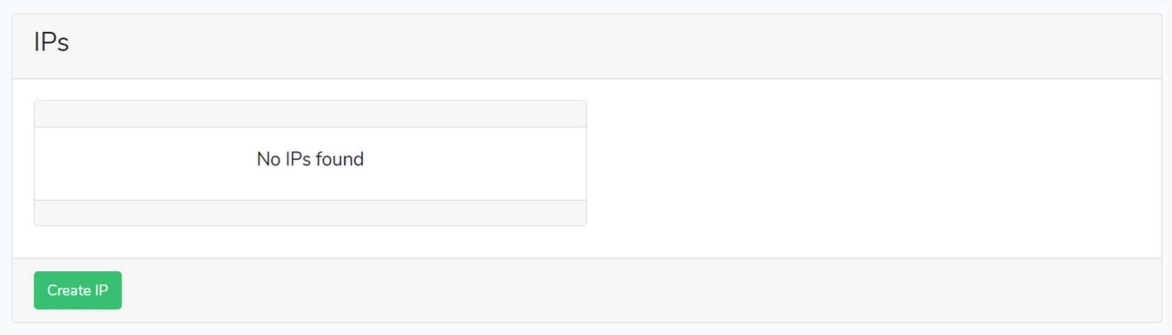
7.2 Přihlášení uživatele

Pro přihlášení již vytvořeného uživatele je každý uživatel přesměrován na formulář s přihlašováním. Po vyplnění se formulář odešle tlačítkem Login.



Obrázek 29: Přihlašovací formulář

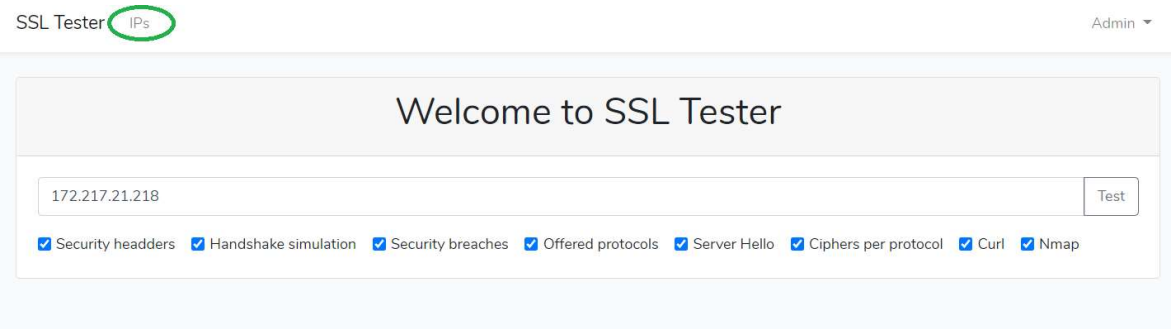
Po zadání validních přihlašovacích údajů je uživatel úspěšně přihlášen do aplikace.



Obrázek 30: Po úspěšném přihlášení

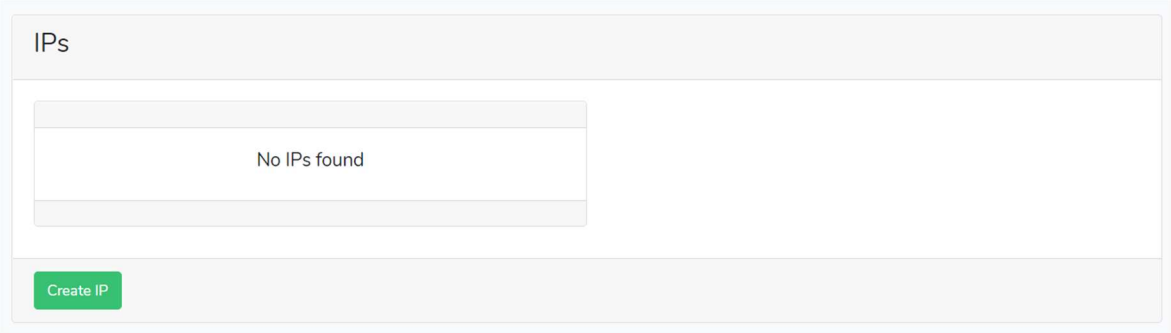
7.3 Plánování testů

Pro možnost plánovat testy je potřeba, aby byl uživatel přihlášený a ověřený administrátorem. Do přehledu naplánovaných testů se uživatel dostane tlačítkem IPs.



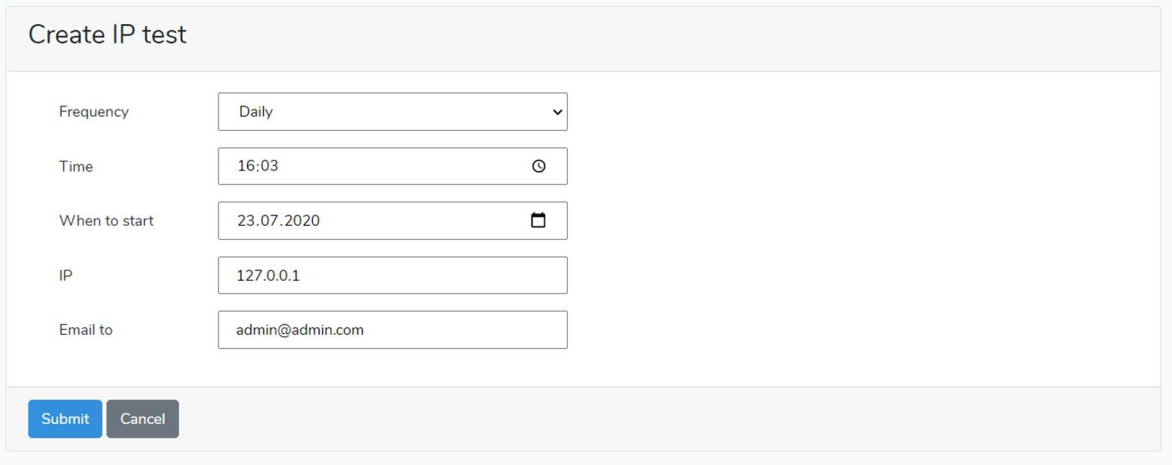
Obrázek 31: Tlačítko pro přehled naplánovaných testů

Následně se zobrazí přehled plánovaných testů. Zde pro vytvoření nového testu slouží tlačítko Create IP.



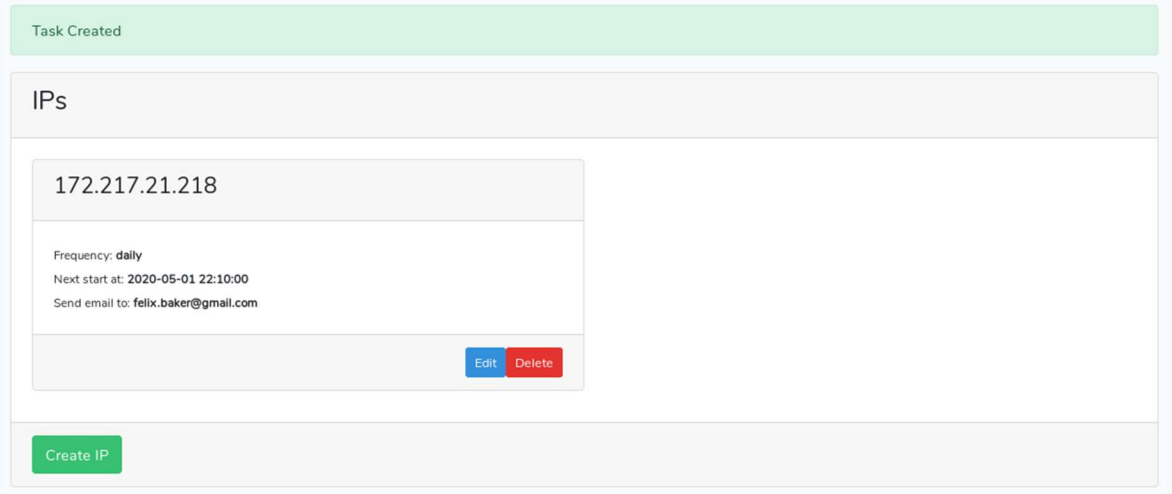
Obrázek 32: Přehled plánovaných testů

Po kliknutí na tlačítko Create IP se zobrazí formulář, ve kterém si uživatel zvolí, jak často chce test provádět. Na výběr je Daily, to znamená každý den, Weekly, to znamená jednou za týden a One time, tato volba znamená, že test bude spuštěn pouze jednou. Poté si uživatel vybere čas, kdy se daný test bude spouštět. V dalším poli je datum prvního testu. Poté si uživatel zvolí IP pro provedení testů a emailovou adresu, na kterou mu budou chodit emaily s výsledky naplánovaného testu. Následným vybráním tlačítka Submit přidá novou plánovanou úlohu. Tlačítko Cancel slouží pro návrat k přehledu plánovaných testů bez uložení nového testu.



Obrázek 33: Vytvoření nové plánované úlohy

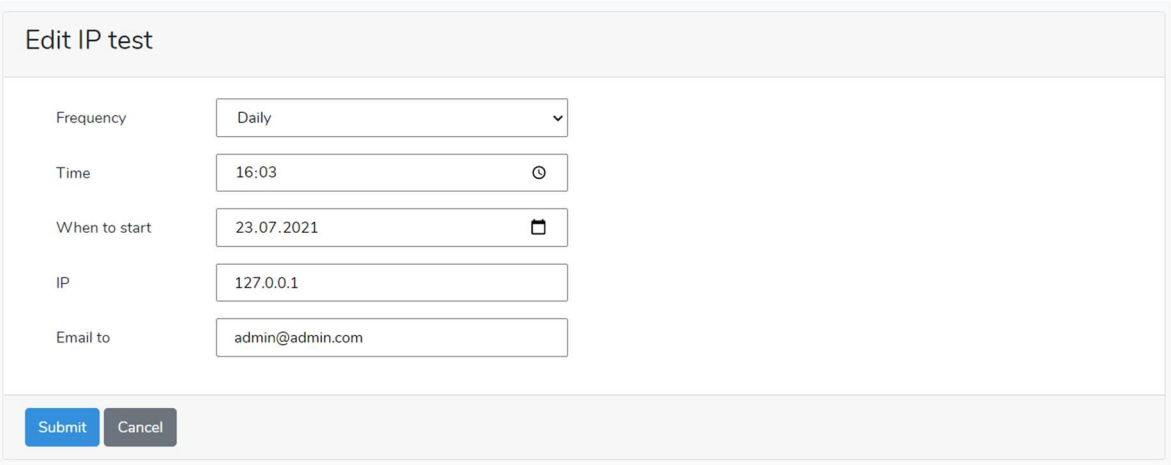
Po úspěšném vytvoření nové plánované úlohy je uživatel přesměrován na přehled plánovaných testů, kde přibyla nová úloha.



Obrázek 34: Přidání nového plánovaného testu

7.4 Editace testů

Z přehledu naplánovaných testů lze měnit nastavené hodnoty testů. Pro editaci nastavených hodnot pro jednotlivé testy slouží tlačítko Edit, které přesměrovává na stránku editace testu.

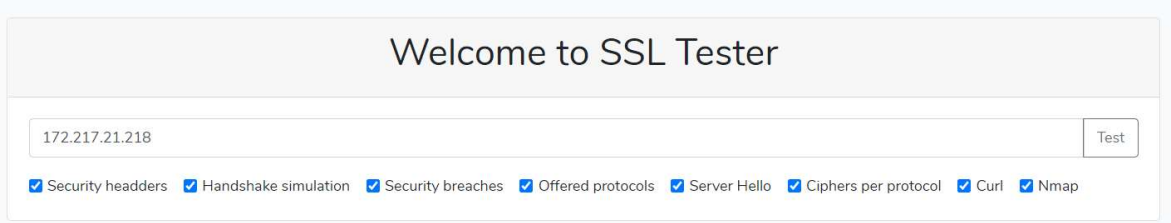


Obrázek 35: Editace plánované úlohy

Zde má uživatel možnost změnit předchozí hodnoty na libovolné nové a změnu potvrdit tlačítkem Submit.

7.5 Spuštění jednorázových testů

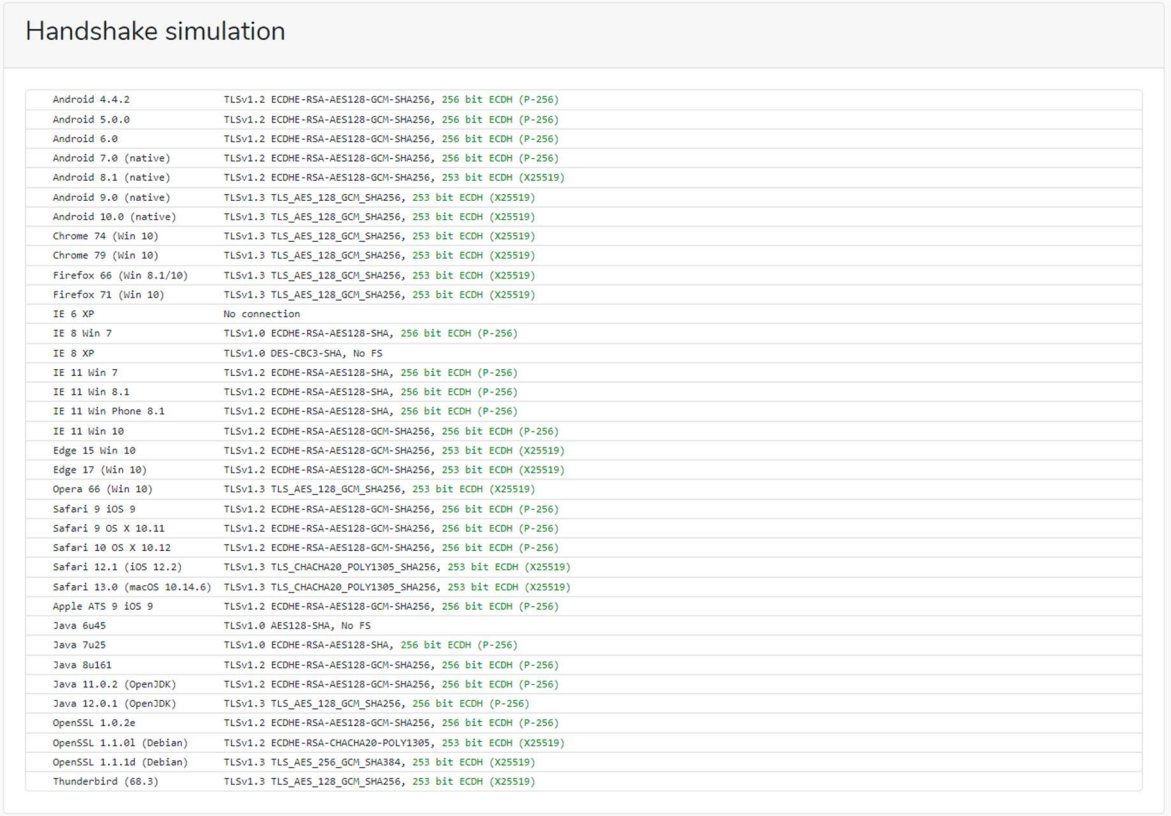
Po přihlášení je uživatel přesměrován na domovskou stránku, na které může spouštět jednorázové testy pro libovolný internetový portál. Má možnost si jednotlivé testy vybrat pomocí checkboxů pod vyhledávacím polem:



Obrázek 36: Jednorázové testování

7.5.1 Handshake simulation

Tento test spočívá v tom, že aplikace simuluje připojení různých uživatelů s různými typy prohlížečů, jako je Chrome, Firefox, Internet Explorer, Safari a další.



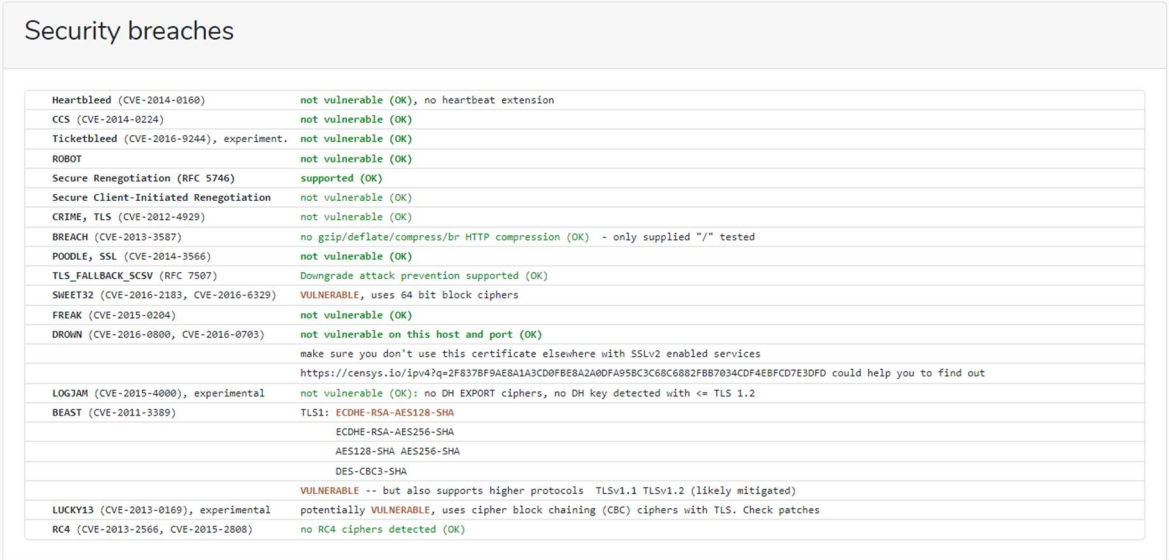
Handshake simulation

Android 4.4.2	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Android 5.0.0	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Android 6.0	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Android 7.0 (native)	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Android 8.1 (native)	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 253 bit ECDH (X25519)
Android 9.0 (native)	TLSv1.3 TLS_AES_128_GCM_SHA256, 253 bit ECDH (X25519)
Android 10.0 (native)	TLSv1.3 TLS_AES_128_GCM_SHA256, 253 bit ECDH (X25519)
Chrome 74 (Win 10)	TLSv1.3 TLS_AES_128_GCM_SHA256, 253 bit ECDH (X25519)
Chrome 79 (Win 10)	TLSv1.3 TLS_AES_128_GCM_SHA256, 253 bit ECDH (X25519)
Firefox 66 (Win 8.1/10)	TLSv1.3 TLS_AES_128_GCM_SHA256, 253 bit ECDH (X25519)
Firefox 71 (Win 10)	TLSv1.3 TLS_AES_128_GCM_SHA256, 253 bit ECDH (X25519)
IE 6 XP	No connection
IE 8 Win 7	TLSv1.0 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
IE 8 XP	TLSv1.0 DES-CBC3-SHA, No FS
IE 11 Win 7	TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
IE 11 Win 8.1	TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
IE 11 Win Phone 8.1	TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
IE 11 Win 10	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Edge 15 Win 10	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 253 bit ECDH (X25519)
Edge 17 (Win 10)	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 253 bit ECDH (X25519)
Opera 66 (Win 10)	TLSv1.3 TLS_AES_128_GCM_SHA256, 253 bit ECDH (X25519)
Safari 9 iOS 9	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Safari 9 OS X 10.11	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Safari 10 OS X 10.12	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Safari 12.1 (iOS 12.2)	TLSv1.3 TLS_CHACHA20_POLY1305_SHA256, 253 bit ECDH (X25519)
Safari 13.0 (macOS 10.14.6)	TLSv1.3 TLS_CHACHA20_POLY1305_SHA256, 253 bit ECDH (X25519)
Apple ATS 9 iOS 9	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Java 6u45	TLSv1.0 AES128-SHA, No FS
Java 7u25	TLSv1.0 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
Java 8u161	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Java 11.0.2 (OpenJDK)	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Java 12.0.1 (OpenJDK)	TLSv1.3 TLS_AES_128_GCM_SHA256, 256 bit ECDH (P-256)
OpenSSL 1.0.2e	TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
OpenSSL 1.1.0l (Debian)	TLSv1.2 ECDHE-RSA-CHACHA20-POLY1305, 253 bit ECDH (X25519)
OpenSSL 1.1.1d (Debian)	TLSv1.3 TLS_AES_256_GCM_SHA384, 253 bit ECDH (X25519)
Thunderbird (68.3)	TLSv1.3 TLS_AES_128_GCM_SHA256, 253 bit ECDH (X25519)

Obrázek 37: Handshake simulation

7.5.2 Security breaches

V tomto testu se aplikace pokouší simulovat známé útoky na portál a posoudí, zda je daný portál proti těmto útokům zabezpečen či nikoli. Ze známých útoků se testuje třeba Heartbleed, POODLE a BEAST.



Obrázek 38: Security breaches

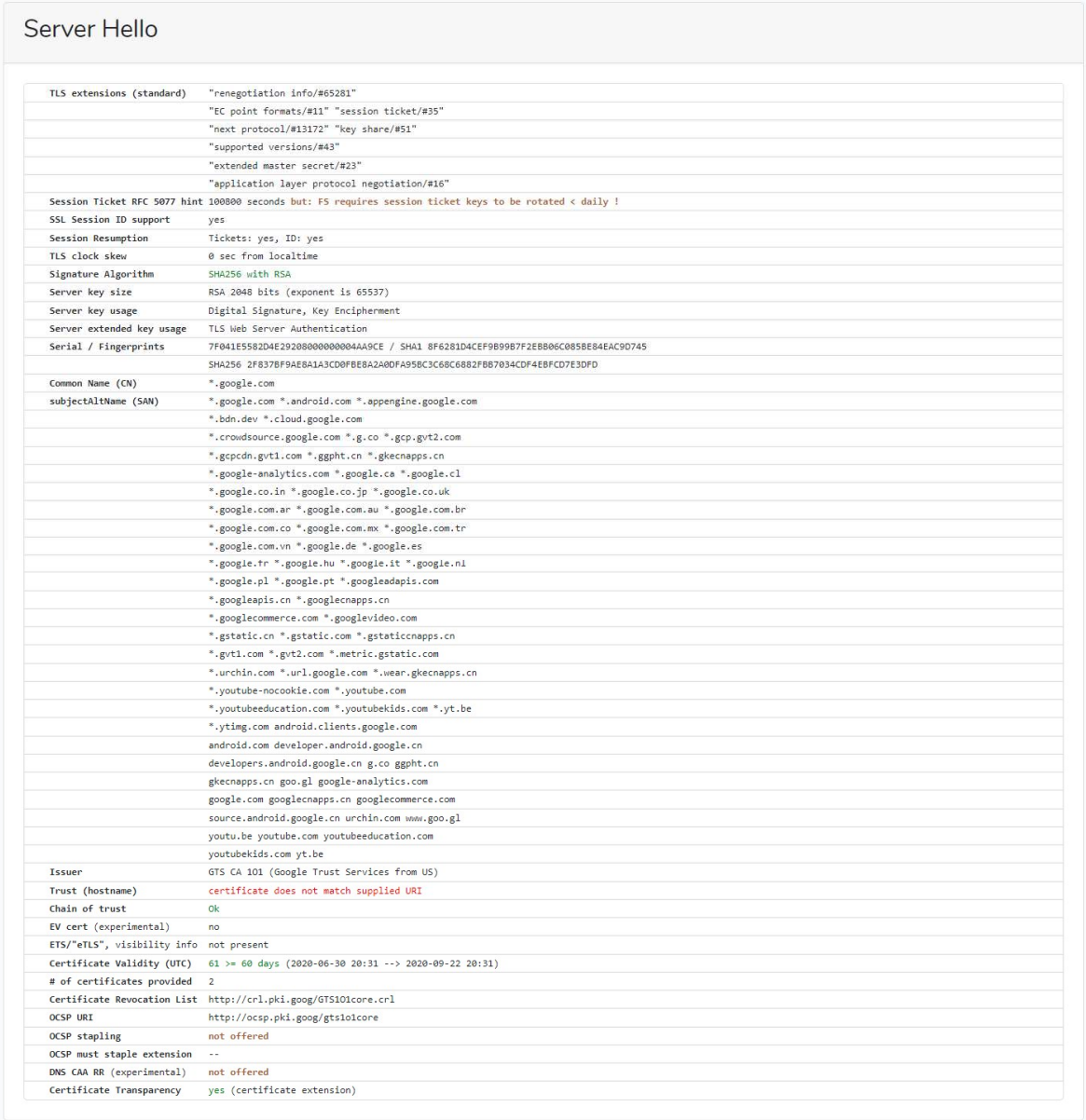
7.5.3 Test komunikačních protokolů

Tento test zkouší komunikovat se serverem pomocí různých protokolů, jako jsou SSL a TSL

Obrázek 39: Komunikační protokoly

7.5.4 Identifikace portálu a ověření platnosti certifikátu

Tento test provede ověření platnosti certifikátu včetně zjištění základních informací o daném portálu.



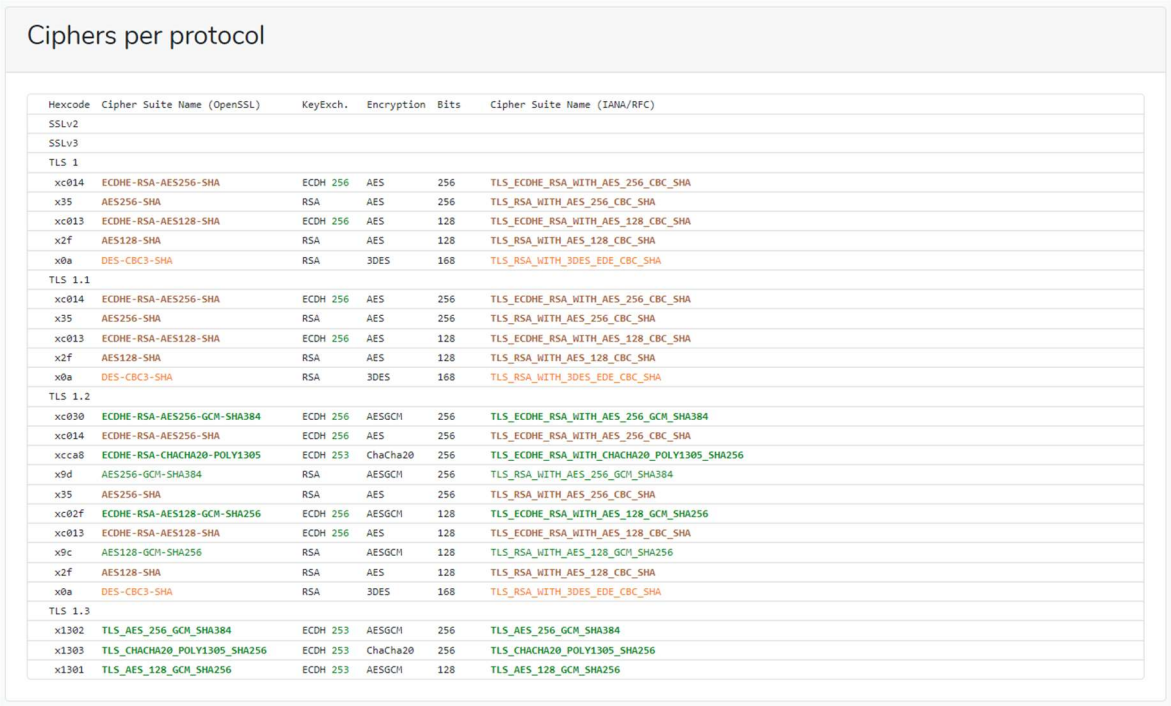
Server Hello

TLS extensions (standard)	"renegotiation info/#65281"
	"EC point formats/#11" "session ticket/#35"
	"next protocol/#13172" "key share/#51"
	"supported versions/#43"
	"extended master secret/#23"
	"application layer protocol negotiation/#16"
Session Ticket RFC 5077 hint	100800 seconds but: FS requires session ticket keys to be rotated < daily !
SSL Session ID support	yes
Session Resumption	Tickets: yes, ID: yes
TLS clock skew	0 sec from localtime
Signature Algorithm	SHA256 with RSA
Server key size	RSA 2048 bits (exponent is 65537)
Server key usage	Digital Signature, Key Encipherment
Server extended key usage	TLS Web Server Authentication
Serial / Fingerprints	7F041E5582D4E292080000000004AA9CE / SHA1 8F6281D4CEF9B99B7F2EBB06C085BE84EAC9D745
	SHA256 2F837BF9AE8A1A3CD0FBE8A2A0DFA95BC3C68C6882FBB7034CDF4EBFCD7E3DFD
Common Name (CN)	*.google.com
subjectAltName (SAN)	*.google.com *.android.com *.appengine.google.com
	*.bdn.dev *.cloud.google.com
	*.crowdsource.google.com *.g.co *.gcp.gvt2.com
	*.gcpcdn.gvt1.com *.ggpht.cn *.gkecnapps.cn
	*.google-analytics.com *.google.ca *.google.cl
	*.google.co.in *.google.co.jp *.google.co.uk
	*.google.com.ar *.google.com.au *.google.com.br
	*.google.com.co *.google.com.mx *.google.com.tr
	*.google.com.vn *.google.de *.google.es
	*.google.fr *.google.hu *.google.it *.google.nl
	*.google.pl *.google.pt *.googleadapis.com
	*.googleapis.cn *.googlecnapps.cn
	*.googlecommerce.com *.googlevideo.com
	*.gstatic.cn *.gstatic.com *.gstaticcnapps.cn
	*.gvt1.com *.gvt2.com *.metric.gstatic.com
	*.urchin.com *.url.google.com *.wear.gkecnapps.cn
	*.youtube-nocookie.com *.youtube.com
	*.youtubeeducation.com *.youtubekids.com *.yt.be
	*.ytimg.com android.clients.google.com
	android.com developer.android.google.cn
	developers.android.google.cn g.co ggpht.cn
	gkecnapps.cn goo.gl google-analytics.com
	google.com googlecnapps.cn googlecommerce.com
	source.android.google.cn urchin.com www.goo.gl
	youtu.be youtube.com youtubeeducation.com
	youtubekids.com yt.be
Issuer	GTS CA 1O1 (Google Trust Services from US)
Trust (hostname)	certificate does not match supplied URI
Chain of trust	Ok
EV cert (experimental)	no
ETS/"eTLS", visibility info	not present
Certificate Validity (UTC)	61 >= 60 days (2020-06-30 20:31 --> 2020-09-22 20:31)
# of certificates provided	2
Certificate Revocation List	http://crl.pki.goog/GTS1O1core.crl
OCSP URI	http://ocsp.pki.goog/gts1o1core
OCSP stapling	not offered
OCSP must staple extension	--
DNS CAA RR (experimental)	not offered
Certificate Transparency	yes (certificate extension)

Obrázek 40: Ověření certifikátu

7.5.5 Test šifer podle protokolů

V tomto testu se aplikace pokouší spojit s testovaným portálem pomocí různých protokolů a různým šifrováním.



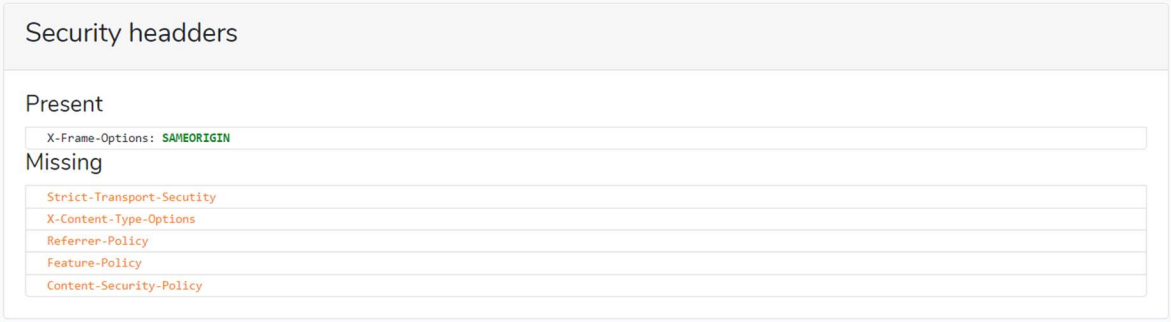
Ciphers per protocol

Hexcode	Cipher Suite Name (OpenSSL)	KeyExch.	Encryption	Bits	Cipher Suite Name (IANA/RFC)
SSLv2					
SSLv3					
TLS 1					
xc014	ECDHE-RSA-AES256-SHA	ECDH 256	AES	256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
x35	AES256-SHA	RSA	AES	256	TLS_RSA_WITH_AES_256_CBC_SHA
xc013	ECDHE-RSA-AES128-SHA	ECDH 256	AES	128	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
x2f	AES128-SHA	RSA	AES	128	TLS_RSA_WITH_AES_128_CBC_SHA
x0a	DES-CBC3-SHA	RSA	3DES	168	TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLS 1.1					
xc014	ECDHE-RSA-AES256-SHA	ECDH 256	AES	256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
x35	AES256-SHA	RSA	AES	256	TLS_RSA_WITH_AES_256_CBC_SHA
xc013	ECDHE-RSA-AES128-SHA	ECDH 256	AES	128	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
x2f	AES128-SHA	RSA	AES	128	TLS_RSA_WITH_AES_128_CBC_SHA
x0a	DES-CBC3-SHA	RSA	3DES	168	TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLS 1.2					
xc030	ECDHE-RSA-AES256-GCM-SHA384	ECDH 256	AESGCM	256	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
xc014	ECDHE-RSA-AES256-SHA	ECDH 256	AES	256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
xcca8	ECDHE-RSA-CHACHA20-POLY1305	ECDH 253	ChaCha20	256	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
x9d	AES256-GCM-SHA384	RSA	AESGCM	256	TLS_RSA_WITH_AES_256_GCM_SHA384
x35	AES256-SHA	RSA	AES	256	TLS_RSA_WITH_AES_256_CBC_SHA
xc02f	ECDHE-RSA-AES128-GCM-SHA256	ECDH 256	AESGCM	128	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
xc013	ECDHE-RSA-AES128-SHA	ECDH 256	AES	128	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
x9c	AES128-GCM-SHA256	RSA	AESGCM	128	TLS_RSA_WITH_AES_128_GCM_SHA256
x2f	AES128-SHA	RSA	AES	128	TLS_RSA_WITH_AES_128_CBC_SHA
x0a	DES-CBC3-SHA	RSA	3DES	168	TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLS 1.3					
x1302	TLS_AES_256_GCM_SHA384	ECDH 253	AESGCM	256	TLS_AES_256_GCM_SHA384
x1303	TLS_CHACHA20_POLY1305_SHA256	ECDH 253	ChaCha20	256	TLS_CHACHA20_POLY1305_SHA256
x1301	TLS_AES_128_GCM_SHA256	ECDH 253	AESGCM	128	TLS_AES_128_GCM_SHA256

Obrázek 41: Možné šifry pro jednotlivé protokoly

7.5.6 Test bezpečnostních hlaviček

Při tomto testu se porovnává přítomnost bezpečnostních hlaviček na daném portálu s očekávaným seznamem.



Security headders

Present

X-Frame-Options: SAMEORIGIN

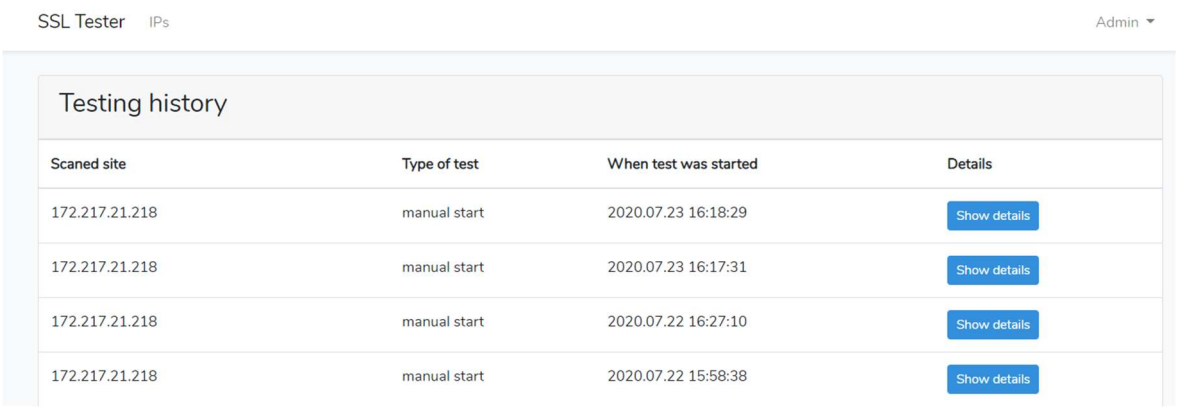
Missing

Strict-Transport-Secutity
X-Content-Type-Options
Referrer-Policy
Feature-Policy
Content-Security-Policy

Obrázek 42: Bezpečnostní hlavičky

7.6 Přehled proběhlých testů

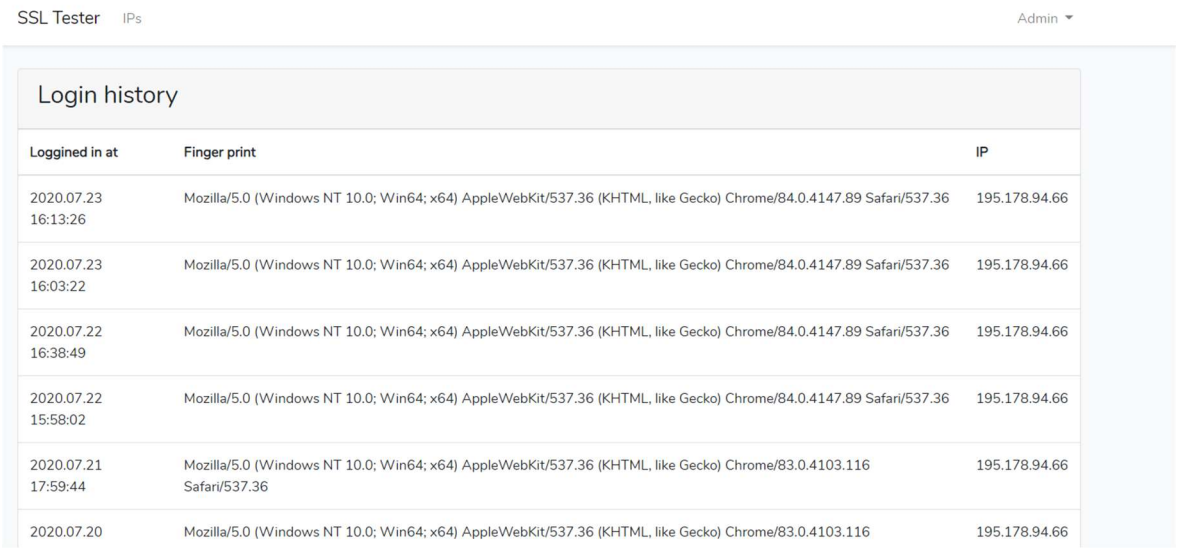
Do této části se uživatel dostane kliknutím na rozbalovací nabídku u svého jména a vybráním možnosti Testing history. Zde uživatel vidí každý svůj test, který spustil a má možnost se vrátit k jeho výsledkům.



Obrázek 43: Přehled proběhnutých testů

7.7 Přehled historie přihlašování

Do této části se uživatel dostane kliknutím na rozbalovací nabídku u svého jména a vybráním možnosti Login history. Zde uživatel vidí přehled o tom, kdy se přihlásil na tento portál, z jaké IP adresy a jaký měl fingerprint jeho prohlížeč.



Obrázek 44: Historie přihlášení

7.8 Administrátorské úkony

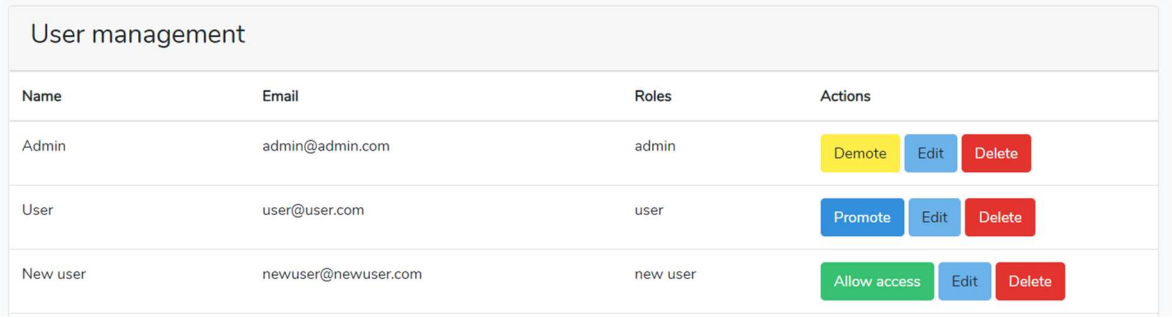
Tato část bude popisovat aktivity administrátorů.

7.8.1 Správa uživatelů

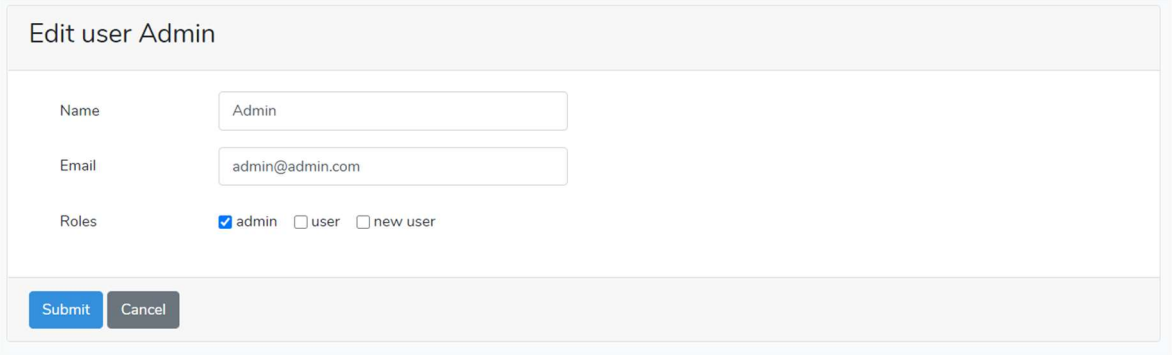
Po přihlášení je administrátor přesměrován na stránku s přehledem všech uživatelů. Zde má možnost aktivovat účet novému uživateli. Tuto akci provede kliknutím na tlačítko Allow access. Dále má možnost uživateli přidat administrátorská práva, tuto akci provede kliknutím na tlačítko Promote. Pokud administrátor potřebuje někoho zbavit administrátorských práv, provede to tlačítkem Demote.

Administrátor má možnost i smazat účet. Tuto akci provede tlačítkem Delete a následným potvrzením dialogového okna, že si je administrátor jist svým rozhodnutím.

Administrátor má možnost i přímé úpravy jména a emailu uživatele. K tomu slouží tlačítko Edit. Po kliknutí je administrátor přesměrován k vyplnění formuláře a po odeslání formuláře je uživatel upraven.



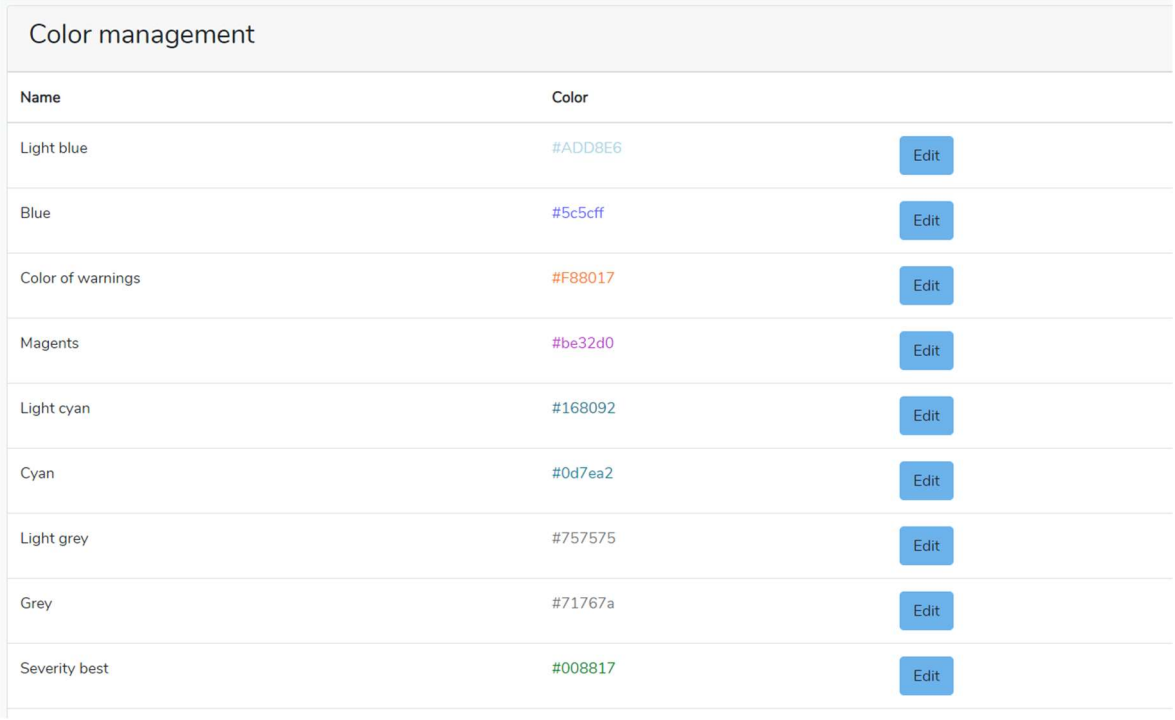
Obrázek 45: Správa uživatelů



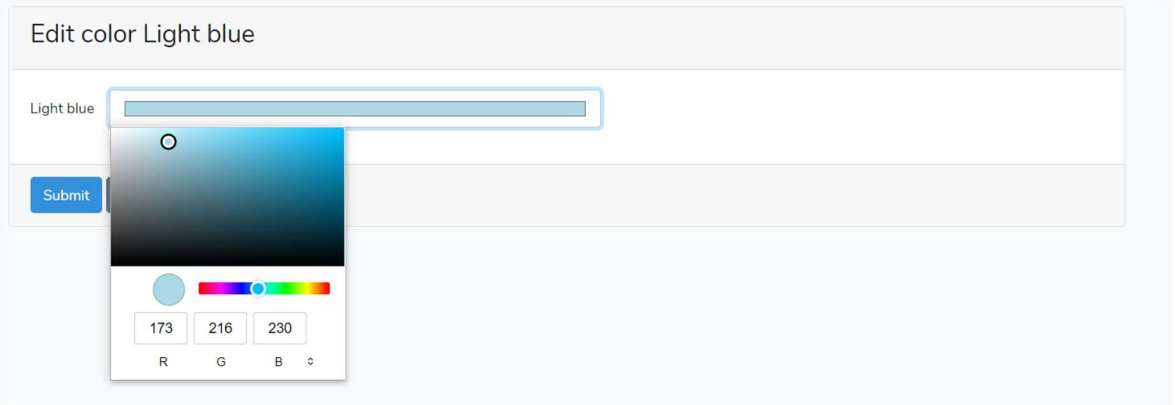
Obrázek 46: Editace uživatele

7.8.2 Změna barvy zvýraznění v textu

Do této části se Administrátor dostane kliknutím na rozbalovací nabídku u svého jména a vybráním možnosti Color management. Zde vidí aktuálně nastavené barvy a tlačítkem Edit je přesměrován na formulář k dané barvě. Zde může změnit barvu podle svého uvážení. Po odeslání formuláře je tato nově zvolená barva používána v jednotlivých testech.



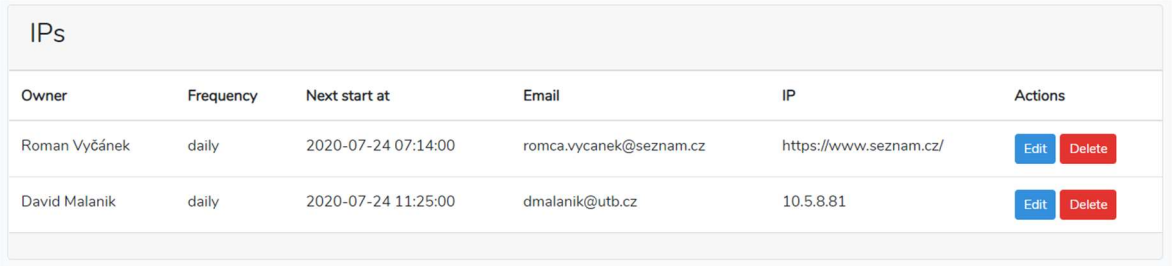
Obrázek 47: Správa barev



Obrázek 48: Editace barvy

7.8.3 Editace plánovaných testů

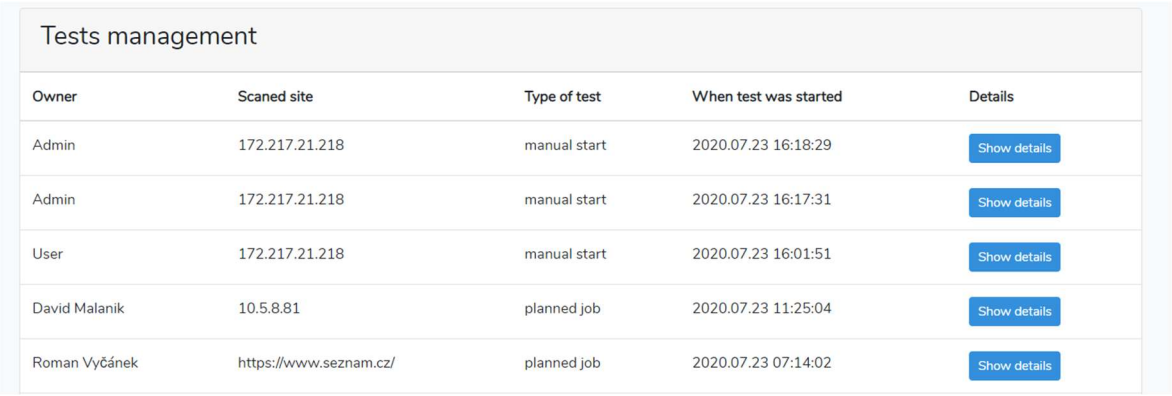
Pokud administrátor chce zkontrolovat, jaké všechny plánované testy existují, může tak učinit v části, do které se dostane kliknutím na rozbalovací nabídku u svého jména a vybráním možnosti IPs management. Zde má následně administrátor možnost plánovanou úlohu smazat tlačítkem Delete, nebo ji změnit tlačítkem Edit. Editace testu je popsána v kapitole 11.4.



Obrázek 49: Editace plánovaných testů

7.8.4 Prohlížení všech proběhlých testů

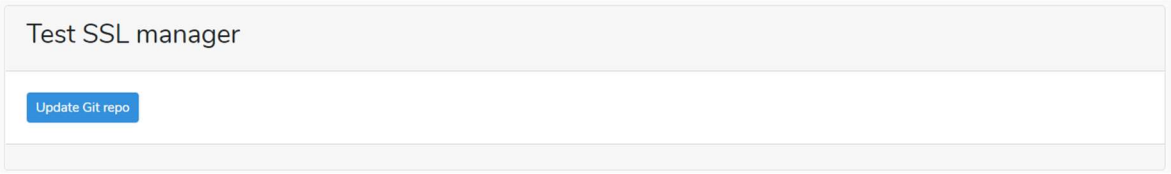
Do této části se Administrátor dostane kliknutím na rozbalovací nabídku u svého jména a vybráním možnosti Tests management. Tady poté administrátor vidí veškeré testy všech uživatelů a má možnost se podívat na jejich výsledky kliknutím na tlačítko Show details.



Obrázek 50: Prohlížení všech proběhlých testů

7.8.5 Aktualizace testovacího balíčku

Do této části se Administrátor dostane kliknutím na rozbalovací nabídku u svého jména a vybráním možnosti SSL test tool Management. Zde se jedním kliknutím na tlačítko Update Git repo provede sada příkazů, které stáhnou poslední verzi testovacího software.



Obrázek 51: Aktualizace Git balíčku

8 MOŽNOSTI ROZŠÍŘENÍ APLIKACE

Aplikace poskytuje velkou škálu možností pro další rozšíření její funkcionality, jak po stránce množství podporovaných testů, tak pro rozšíření vyhodnocování celkových testů. Další možností rozšíření aplikace je ověření emailu, přidání změny hesla a další funkce, které zjednoduší užívání této aplikace uživatelům. Přínosnou funkcí pro usnadnění kontrolní práce administrátorů je přidání soft delete do databáze.

Další požadavky na rozšíření databáze vyplynou z testovacího provozu.

ZÁVĚR

Cílem diplomové práce bylo vytvořit portál pro testování zranitelnosti HTTPS portálů. V teoretické části jsou popsány základní požadavky na systém, které byly v praktické části naplněny. Teoretická část dále popisuje druhy testů a jejich interpretaci. Nakonec se teoretická část zabývá technologiemi, které lze využít pro tvorbu testovacího portálu. Na základě poznatků, uvedených v teoretické části, byl následně navržen testovací portál.

V praktické části byly popsány jednotlivé pohledy, včetně kódů s jejich implementací. Dále byla navržena databáze pro potřeby uchování všech dat o uživatelích a testech. Samotný portál byl vyvíjen ve vývojovém prostředí Visual Studio Code. Jako programovací jazyky byly využity hlavně PHP a JavaScript, doplněné o HTML, CSS a Python. Mezi frameworky použité k vývoji patří Laravej, Bootstrap, Blade a Vue.js. Celý vývojový projekt byl zálohován na GitHub a z tohoto zdroje se dál může šířit. Další oblast praktické části se věnuje samotnému nasazení na čistý systém Debianu. Úplný závěr tvoří uživatelský manuál jak pro uživatelské účty, tak pro účet administrátorský.

Vytvořený portál pro testování zranitelnosti HTTPS portálů je užitečným nástrojem vývojáře pro zvýšení bezpečnosti na internetu.

SEZNAM POUŽITÉ LITERATURY

[1] Blade. Github.com [online]. [cit. 2020-04-16]. Dostupné z: https://github.com/lets-blade/blade#what-is-blade

[2] Vue.js. The Progressive JavaScript Framework [online]. [cit. 2020-04-16]. Dostupné z: https://vuejs.org/

[3] Oracle and Sun Microsystems. Oracle [online]. [cit. 2020-04-20]. Dostupné z: https://www.oracle.com/sun/

[4] LAMP (Linux, Apache, MySQL, PHP). WhatIs.com [online]. [cit. 2020-04-20]. Dostupné z: https://whatis.techtarget.com/definition/LAMP-Linux-Apache-MySQL-PHP

[5] Using, Abusing and Scaling MySQL at Flickr. code.flickr.com [online]. [cit. 2020-04-20]. Dostupné z: https://engineering.fb.com/core-data/myrocks-a-space-and-write-optimized-mysql-database/

[6] MyRocks: A space- and write-optimized MySQL database. code.flickr.com [online]. [cit. 2020-04-20]. Dostupné z: https://code.flickr.net/2010/02/08/using-abusing-and-scaling-mysql-at-flickr/

[7] The Infrastructure Behind Twitter: Scale. engineering.fb.com [online]. [cit. 2020-04-20]. Dostupné z: https://blog.twitter.com/engineering/en_us/topics/infrastructure/2017/the-infrastructure-behind-twitter-scale.html

[8] MySQL Customer: YouTube. mysql.com [online]. [cit. 2020-04-20]. Dostupné z: https://www.mysql.com/customers/view/?id=750

[9] Apache http server project. apache.org [online]. [cit. 2020-04-20]. Dostupné z: https://httpd.apache.org/

[10] Bootstrap. getbootstrap.com [online]. [cit. 2020-04-20]. Dostupné z: https://getbootstrap.com/

[11] Vue.js. github.com [online]. [cit. 2020-04-20]. Dostupné z: https://github.com/vuejs/vue

[12] Laravel documentation. laravel.com/docs [online]. [cit. 2020-04-20]. Dostupné z: https://laravel.com/docs/7.x/blade

[13] Laravel. github.com [online]. [cit. 2020-04-20]. Dostupné z: https://github.com/laravel/framework

[14] Laravel intro. w3schools.in [online]. [cit. 2020-04-20]. Dostupné z: https://www.w3schools.in/laravel-tutorial/intro/

[15] Framework. techterms.com [online]. [cit. 2020-04-20]. Dostupné z: https://techterms.com/definition/framework

[16] Visual Studio Code. github.com [online]. [cit. 2020-04-21]. Dostupné z: https://github.com/microsoft/vscode

[17] What is PHP. php.net [online]. [cit. 2020-04-21]. Dostupné z: https://www.php.net/manual/en/intro-whatis.php

[18] HTML Introduction. w3schools.com [online]. [cit. 2020-04-21]. Dostupné z: https://www.w3schools.com/html/html_intro.asp

[19] An Introduction to JavaScript. javascript.info [online]. [cit. 2020-04-21]. Dostupné z: https://javascript.info/intro

[20] Reasons to Choose Debian. debian.org [online]. [cit. 2020-04-21]. Dostupné z: https://www.debian.org/intro/why_debian

[21] The Visual Workspace For Team Collaboration. creately.com [online]. [cit. 2020-04-21]. Dostupné z: https://creately.com/

[22] NATIONAL VULNERABILITY DATABASE. nvd.nist.gov [online]. [cit. 2020-06-6]. Dostupné z: https://nvd.nist.gov/vuln/detail/CVE-2014-0224

[23] The Heartbleed Bug. heartbleed.com [online]. [cit. 2020-06-6]. Dostupné z: https://heartbleed.com/

[24] Ticketbleed (CVE-2016-9244). filippo.io [online]. [cit. 2020-06-6]. Dostupné z: https://filippo.io/ticketbleed/

[25] The ROBOT Attack. robotattack.org [online]. [cit. 2020-06-6]. Dostupné z: https://robotattack.org/

[26] Transport Layer Security (TLS) Renegotiation Indication Extension. tools.ietf.org [online]. [cit. 2020-06-6]. Dostupné z: https://tools.ietf.org/html/rfc5746

[27] Secure Client-Initiated SSL Renegotiation. crashtest-security.com [online]. [cit. 2020-06-6]. Dostupné z: https://wiki.crashtest-security.com/secure-client-initiated-ssl-renegotiation

[28] NATIONAL VULNERABILITY DATABASE. nvd.nist.gov [online]. [cit. 2020-06-6]. Dostupné z: https://nvd.nist.gov/vuln/detail/CVE-2012-4929

[29] NATIONAL VULNERABILITY DATABASE. nvd.nist.gov [online]. [cit. 2020-06-6]. Dostupné z: https://nvd.nist.gov/vuln/detail/CVE-2013-3587

[30] NATIONAL VULNERABILITY DATABASE. nvd.nist.gov [online]. [cit. 2020-06-6]. Dostupné z: https://nvd.nist.gov/vuln/detail/CVE-2014-3566

[31] TLS Fallback Signaling Cipher Suite Value (SCSV) for Preventing Protocol Downgrade Attacks. tools.ietf.org [online]. [cit. 2020-06-6]. Dostupné z: https://tools.ietf.org/html/rfc7507

[32] NATIONAL VULNERABILITY DATABASE. nvd.nist.gov [online]. [cit. 2020-06-6]. Dostupné z: https://nvd.nist.gov/vuln/detail/CVE-2016-2183

[33] NATIONAL VULNERABILITY DATABASE. nvd.nist.gov [online]. [cit. 2020-06-6]. Dostupné z: https://nvd.nist.gov/vuln/detail/CVE-2015-0204

[34] NATIONAL VULNERABILITY DATABASE. nvd.nist.gov [online]. [cit. 2020-06-6]. Dostupné z: https://nvd.nist.gov/vuln/detail/CVE-2016-0800

[35] NATIONAL VULNERABILITY DATABASE. nvd.nist.gov [online]. [cit. 2020-06-6]. Dostupné z: https://nvd.nist.gov/vuln/detail/CVE-2015-4000

[36] NATIONAL VULNERABILITY DATABASE. nvd.nist.gov [online]. [cit. 2020-06-6]. Dostupné z: https://nvd.nist.gov/vuln/detail/CVE-2011-3389

[37] NATIONAL VULNERABILITY DATABASE. nvd.nist.gov [online]. [cit. 2020-06-6]. Dostupné z: https://nvd.nist.gov/vuln/detail/CVE-2013-0169

[38] NATIONAL VULNERABILITY DATABASE. nvd.nist.gov [online]. [cit. 2020-06-6]. Dostupné z: https://nvd.nist.gov/vuln/detail/CVE-2013-2566

[39] What is Secure Sockets Layer (SSL)?. digicert.com [online]. [cit. 2020-06-6]. Dostupné z: https://www.digicert.com/ssl/

[40] What is Transport Layer Security (TLS)?. networkworld.com [online]. [cit. 2020-06-6]. Dostupné z: https://www.networkworld.com/article/2303073/lan-wan-what-is-transport-layer-security-protocol.html

[41] Simple SPDY and NPN Negotiation with HAProxy. igvita.com [online]. [cit. 2020-06-6]. Dostupné z: https://www.igvita.com/2012/10/31/simple-spdy-and-npn-negotiation-with-haproxy/

[42] ALPN Explained. keycdn.com [online]. [cit. 2020-06-6]. Dostupné z: https://www.keycdn.com/support/alpn

[43] Introduction to HTTP/2. developers.google.com [online]. [cit. 2020-06-6]. Dostupné z: https://developers.google.com/web/fundamentals/performance/http2

[44] Transport Layer Security (TLS) Extensions: Extension Definitions. ietf.org [online]. [cit. 2020-06-6]. Dostupné z: https://tools.ietf.org/html/rfc6066

[45] WE NEED TO TALK ABOUT SESSION TICKETS. filippo.io [online]. [cit. 2020-06-6]. Dostupné z: https://blog.filippo.io/we-need-to-talk-about-session-tickets/

[46] SSL session ID persistence. citrix.com [online]. [cit. 2020-06-6]. Dostupné z: https://docs.citrix.com/en-us/netscaler/12/load-balancing/load-balancing-persistence/session-id-persistence.html

[47] The Transport Layer Security (TLS) Protocol Version 1.2. tools.ietf.org [online]. [cit. 2020-06-6]. Dostupné z: https://tools.ietf.org/html/rfc5246

[48] Why you probably don't want a 4096 bit RSA cert. expeditedsecurity.com [online]. [cit. 2020-06-11]. Dostupné z: https://expeditedsecurity.com/blog/measuring-ssl-rsa-keys/

[49] What extensions and details are included in a SSL certificate?. digicert.com [online]. [cit. 2020-06-11]. Dostupné z: https://knowledge.digicert.com/solution/SO18140.html

[50] Explaining the Chain of Trust. thesslstore.com [online]. [cit. 2020-06-11]. Dostupné z: https://www.thesslstore.com/knowledgebase/ssl-support/explaining-the-chain-of-trust/

[51] Extended Validation certifikáty. slmarket.cz [online]. [cit. 2020-06-11]. Dostupné z: https://www.sslmarket.cz/ssl/ev-certifikaty/

[52] Online Certificate Status Protocol. jamielinux.com [online]. [cit. 2020-06-11]. Dostupné z: https://jamielinux.com/docs/openssl-certificate-authority/online-certificate-status-protocol.html

[53] High-reliability OCSP stapling and why it matters. cloudflare.com [online]. [cit. 2020-06-11]. Dostupné z: https://blog.cloudflare.com/high-reliability-ocsp-stapling/

[54] DNS Certification Authority Authorization (CAA) Resource Record. ietf.org [online]. [cit. 2020-06-11]. Dostupné z: https://tools.ietf.org/html/rfc6844

[55] Certificate Transparency. certificate-transparency.org [online]. [cit. 2020-06-11]. Dostupné z: https://www.certificate-transparency.org/

[55] Introduction to Feature Policy. developers.google.com [online]. [cit. 2020-06-11]. Dostupné z: https://developers.google.com/web/updates/2018/06/feature-policy

[56] meliot/shcheck. github.com [online]. [cit. 2020-07-11]. Dostupné z: https://github.com/meliot/shcheck¨

[57] Testing TLS/SSL encryption. github.com [online]. [cit. 2020-07-11]. Dostupné z: https://github.com/drwetter/testssl.sh

[58] CURL. curl.haxx.se [online]. [cit. 2020-07-11]. Dostupné z: https://curl.haxx.se/

[59] Nmap. nmap.org [online]. [cit. 2020-07-11]. Dostupné z: https://nmap.org/

[60] Content Security Policy. scotthelme.co.uk [online]. [cit. 2020-07-11]. Dostupné z: https://scotthelme.co.uk/content-security-policy-an-introduction/

[61] X-Frame-Options. scotthelme.co.uk [online]. [cit. 2020-07-11]. Dostupné z: https://scotthelme.co.uk/hardening-your-http-response-headers/#x-frame-options

[62] Feature Policy. scotthelme.co.uk [online]. [cit. 2020-07-11]. Dostupné z: https://scotthelme.co.uk/a-new-security-header-feature-policy/

[63] Referrer Policy. scotthelme.co.uk [online]. [cit. 2020-07-11]. Dostupné z: https://scotthelme.co.uk/a-new-security-header-referrer-policy/

[64] HTTP Strict Transport Security. scotthelme.co.uk [online]. [cit. 2020-07-11]. Dostupné z: https://scotthelme.co.uk/hsts-the-missing-link-in-tls/

[65] X-Content-Type-Options. scotthelme.co.uk [online]. [cit. 2020-07-11]. Dostupné z: https://scotthelme.co.uk/hardening-your-http-response-headers/#x-content-type-options

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PHP	PHP: Hypertext Preprocessor
GNU	GNU's Not Unix
GNU GPLv3	GNU General Public License version 3
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
DNS	Domain Name System
DNS CAA	DNS Certification Authority Authorization
DNS CAA RR	DNS CAA Resource Record
SSL	Secure Sockets Layer
CCS	ChangeCipherSpec
ID	Intra Derma
RSA	Rivest Shamir Adleman
PKCS	Public Key Cryptographic Standards
CCA2	Adaptive-Chosen Ciphertext Attack
MITM	Man-In-The-Middle
DES	Data Encryption Standard
CBC	Cipher block chaining
DHE	Ephemeral Diffie-Hellman
MAC	Media Access Control
RC4	Rivest Cipher 4
XFO	X-Frame-Options
CSP	Content-Security-Policy
URL	Uniform Resource Locator
IETF	Internet Engineering Task Force
NPN	Next Protocol Negotiation

SPDY	Speedy
ALPN	Application-Layer Protocol Negotiation
ECDH	Elliptic-Curve Diffie–Hellman
RFC	Request For Comments
OCSP URI	Online Certificate Status Protocol Uniform Resource Identifier
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
MIT	Massachusetts Institute of Technology
MVC	Model-View-Controller
ASP-NET	Active Server Pages NET
MySQL	My Structured Query Language

SEZNAM OBRÁZKŮ

SEZNAM PŘÍLOH

Příloha č. 1: CD s textem diplomové práce a se zdrojovými kódy aplikace