ZADANIE Č.3

I-UPB

Vladimír Hlačina
Fakulta elektrotechniky a informatiky
Slovenská technická univerzita
xhlacina@stuba.sk

Rudolf Bachorík
Fakulta elektrotechniky a informatiky
Slovenská technická univerzita
xbachorik@stuba.sk

Jakub Zigo
Fakulta elektrotechniky a informatiky
Slovenská technická univerzita
xzigo@stuba.sk

Igor Kuzmin
Fakulta elektrotechniky a informatiky
Slovenská technická univerzita
xkuzmini@stuba.sk

20. November 2023

ABSTRAKT

Cieľom zadania je dokončiť webovú aplikáciu podľa špecifikácie uvedenej v AISe. Jadrom zadania je dôraz na bezpečné spracovávanie používateľského vstupu.

1 Rozšírenie databázového modelu podľa zadania

Naším pôvodným zadaním bolo navrhnúť a implementovať webovú aplikáciu na správu dochádzkového sytému. Doterajšie riešenia neumožňovali všetky zadané funkcionality, preto bolo potrebné rozšíriť databázový model pridaním novej tabuľky. Nasledujúca časť kódu z inicializačného sql scriptu popisuje štruktúru novej databázovej tabuľky.

```
1       CREATE TABLE IF NOT EXISTS `mydb`.`time` (
2           `date` VARCHAR(64) NOT NULL,
3           `user_id` VARCHAR(64) NOT NULL,
4           `start` DATETIME,
5           `end` DATETIME,
6           `description` VARCHAR(256),
7           `priority` INT NOT NULL,
8           `month` INT NOT NULL,
9           `year` INT NOT NULL,
10
11          PRIMARY KEY (`date`)
12      )
```

Vyššie spomenutý kód popisuje nastavenie štruktúry novej tabuľky. Pre prehľadnejšie zobrazenie sme uviedli aj nasledovnú tabuľku 1 s jednotlivými stĺpcami.

date - PK	**user_id**	**start**	**end**	**description**	**priority**	**month**	**year**
VARCHAR(64) NOT NULL	VARCHAR(64) NOT NULL	DATETIME	DATETIME	VARCHAR(256)	INT NOT NULL	INT NOT NULL	INT NOT NULL

Tab. 1: Tabuľka štruktúry novej databázovej tabuľky pre zanamenanie časových údajov

Tabuľka obsahuje 8 stĺpcov. Stĺpec `date` slúži na ukladanie dátumu zadania záznamu a taktiež ako primárny kľúč

tabuľky. Stĺpec `user_id` slúži na ukladanie ID používateľa, ktorý daný záznam vytvoril. Stĺpec `start` slúži na ukladanie začiatku záznamu (príchod/odchod alebo dovolenka). Stĺpec `end` hovorí kedy daný časový záznam končí. Stĺpec `description` umožňuje zápis popisu záznamu. Stĺpec `priority` slúži na ukladanie priority záznamu. Stĺpce `month` a `year` definujú na ukladanie mesiaca a roka v ktorom bol záznam vytvorený.

2 Funkcionality aplikácie

Podľa popisu v zadaní sme dospeli k nasledovným funkcionalitám, ktoré by mala naša aplikácia obsahovať.

- Registrácia
- Prihlásenie
- Zaznamenanie dochádzky
- Naslasovanie dovolenky
- Vymazanie záznamu
- Export mesačného prehľadu

Všetky vyššie spomenuté funkcionality sú spomenuté v časti 3, kde je popísané ich použitie s návodom ako sa k nim dostať. Každá funkcionalita je implementovaná na základe platných bezpečnostných protokolov. Komunikácia je neviditeľná pre middlemana, nie je možné ju modifikovať ani prečítať. Túto časť zabezpečujeme tak, že pre telo každej požiadavky najprv vygenerujeme integrity hash zašifrovaný RSA a zašifrujeme body pomocou AES-CFB(256) kľúča. Kľúč z AES následne zašifrujeme pomocou RSA a pošleme to na BE. Backend následne dešifruje RSA kľúčom a dešifruje AES kľúčom. Následne overí integrity hash a po všetkých týchto krokoch overíme či je užívateľ prihlásený, teda či má aktívny session token. Následne backend vykoná akciu z hľadiska funkcionality endpointu a odošle rovnako zašifrované dáta späť s rozdielom, že tentokrát sú zašifrované pomocou verejného kľúča klienta.

2.1 Registrácia

Registrácia je realizovaná pomocou formulára, ktorý obsahuje 3 polia. Prvým poľom je užívateľské meno. V prípade tohoto poľa sme zvolili overenie neprázdnej hodnoty a teda používateľ má možnosť zvoliť si ľubovoľné meno. Ďalším poľom je email, kde sme zvolili overenie pomocou nasledovného štandardného regulárneho výrazu.

```
1           const emailRegex = /^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+[A-Za-z]2,$/;
```

Posledným z krokov je zadanie hodnoty do poľa pre zadanie hesla. Heslo musí spĺňať nasledovné podmienky.

```
1           const errorMessages = {
2               length: 'Heslo musí obsahovať viac ako 12 znakov.',
3               number: 'Heslo musí obsahovať aspoň jedne čislo.',
4               letterCase: 'Heslo musí obsahovať veľké aj malé písmená.',
5               specialCharacter: 'Heslo musí obsahovať aspoň jeden špeciálny znak.'
6           }
```

Výsledkom zadania nesprávnych hodnôt do jednotlivých polí je zobrazenie chybovej hlášky, ktorá hovorí o tom, čo používateľ zadal nesprávne. V prípade, že používateľ zadal všetky hodnoty správne a užívateľ so zadaným emailom ešte neexistuje, je zobrazená hláška o úspešnom vytvorení nového používateľa. Všetky hodnoty sú posielané na BE zašifrované pomocou RSA kde sa pomocou nasledovnej ukážky kódu dešifrujú.

```
1           data = json.loads(str(request.data, 'utf-8'))
2           2 data['email'] = str(decryptRSA(data['email']), 'utf-8')
3           3 data['password'] = str(decryptRSA(data['password']), 'utf-8')
4           4 data['name'] = str(decryptRSA(data['name']), 'utf-8')
```

Proces registrácie obsahuje aj overenie bežne používaných hesiel, ktoré sme bližšie popísali v predošlom zadaní a nie sú obsahom jednotlivých bodov aktuálneho zadania.

2.2 Prihlásenie

Prihlásenie je rovnako realizované pomocou formulára, ktorý obsahuje email a heslo. Po zadaní mena a hesla sa odošlú na server zašifrované pomocou public RSA kľúča. Na úspešné prihásenie musí email spĺňať nasledovné podmienky.

Prvou podmienkou je, overenie brut-force útoku, ktorého podrobnému popisu sme sa venovali v predošlom zadaní. Následne sa email a heslo odšifrujú pomocou privátneho RSA kľúča servera. Ďalej sa overí či užívateľ existuje a skontroluje sa správnosť hesla. Po úspešnom prihlásení je pre používateľa zobrazený admin panel s možnosťami správy dochádzky. Ďalej sme sa rozhodli implementovať tzv. session token pre jednotlivých používateľov. Token pozostáva z 32 náhodných znakov, ktoré sa zašifruje pomocou `AES256`, kde kľúčom je hash hesla a inicializačnný vektor je hash emailu (`SHA256`). Následne sa tento zašifrovaný token pošle späť klientovi, ktorý si ho pomocou svojich prihlasovacích údajov rozšifruje. Token má životnosť 10 minút, po uplynutí tejto doby sa musí používateľ znova prihlásiť ak nevykonal žiadnu požiadavku na sever. Proces prihlásenia obsahuje aj implementáciu anti brute-force opatrení, ktoré sme bližšie popísali v predošlom zadaní a nie sú obsahom jednotlivých bodov aktuálneho zadania.

2.3 Zaznamenanie dochádzky

Táto funkcionalita bola implementovaná na hlavnej stránke aplikácie. Kliknutím na tlačidlo príchod sa na server pošle požiadavka o vytvorenie nového záznamu do tabuľky `time`. Server v prvom rade overí či daný uživateľ uzatvoril svoj posledný príchod tj. nie je možné pridať viac príchodov po sebe. Ak má používateľ príchod uzavretý server vytvorí nový záznam príchodu. Podobným spôsobom funguje aj zaznamenanie odchodu. V prípade, že používateľ zadaný príchod, nie je možné zaznamenať odchod. Týmito dvoma technikami zamedzíme neoprávneným zmenám v dochádzke. Obe tieto akcie sú zaznamenané v prehľadnom kalendári [2] priamo na hlavnej stránke.

2.4 Naslasovanie dovolenky

Nahlasovanie dovolenky funguje na podobnom príncípe ako príchod a odchod 2.3. V tomto prípade si užívateľ zvolí dátum začiatku a konca dovolenky. Server následne overí či sa daný dátum nenachádza v tabuľke `time`. V prípade, že sa daný dátum nenachádza v tabuľke, server vytvorí nový záznam do tabuľky `time` a používateľovi sa v kalendári zobrazí interval dovolenky.

2.5 Vymazanie záznamu

Po kliknutí na záznam v kalendári sa daný záznam vymaže z tabuľky `time` a nebude ho možné znova obnoviť.

2.6 Export mesačného prehľadu

Export mesačného prehľadu je realizovaný pomocou tlačidla, ktoré sa nachádza nad kalendárovou sekciou. Po kliknutí FE následne vytvorí `pdf` dokument, ktorý obsahuje všetky záznamy za daný mesiac. Tento súbor je následne odoslaný späť klientovi, ktorý si ho môže uložiť na svoj počítač. Presunutím tejto požiadavky na používateľskú stranu sme zamedzili zbytočnému zaťaženiu servera. Táto funkcionalita bola implementovaná použitím knižnice `jspdf` [1].

3 Používateľská príručka

V prvom rade chceme dať do pozornosti, že všetky funkcionality prihláseného používateľa sú dostupné po pridaní verejného a privátneho kľúča. Tieto kľúče je možné pridať v sekcii 3.4, ktorá je dostupná z hlavn0ho menu aplikácie. Rovnako je nastavený aj session token, ktorý má platnosť 10 minút od poslednej aktivity. Po uplynutí tohoto času je nutné sa znova prihlásiť a taktiež znova pridať kľúče.

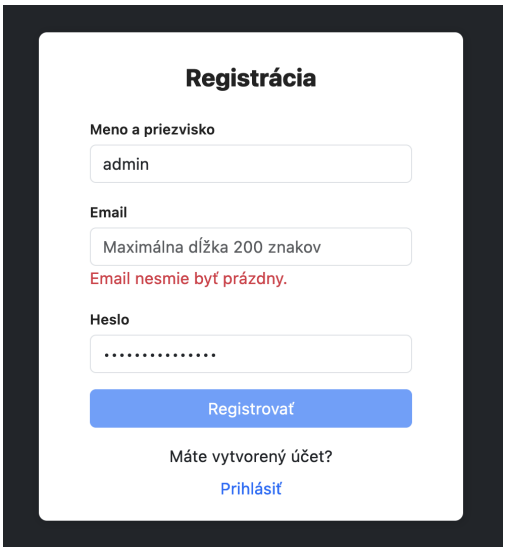
3.1 Prihlásenie

Prvou sekciou aplikácie je prihlásenie. Do tejto sekcie sa vieme dostať priamo pri prvom otvorení aplikácie. Nasledovný obrázok 1 zobrazuje prihlasovací formulár.

Obr. 1: Prihlasovací formulár

3.2 Registrácia

Sekcia registrácia zabezpečuje vytváranie používateľských účtov. Do tejto sekcie sa vieme dostať kliknutím na tlačidlo `Registrovať`, ktoré sa nachádza v prihlasovacom formulári. Nasledovný obrázok 2 zobrazuje registračný formulár.

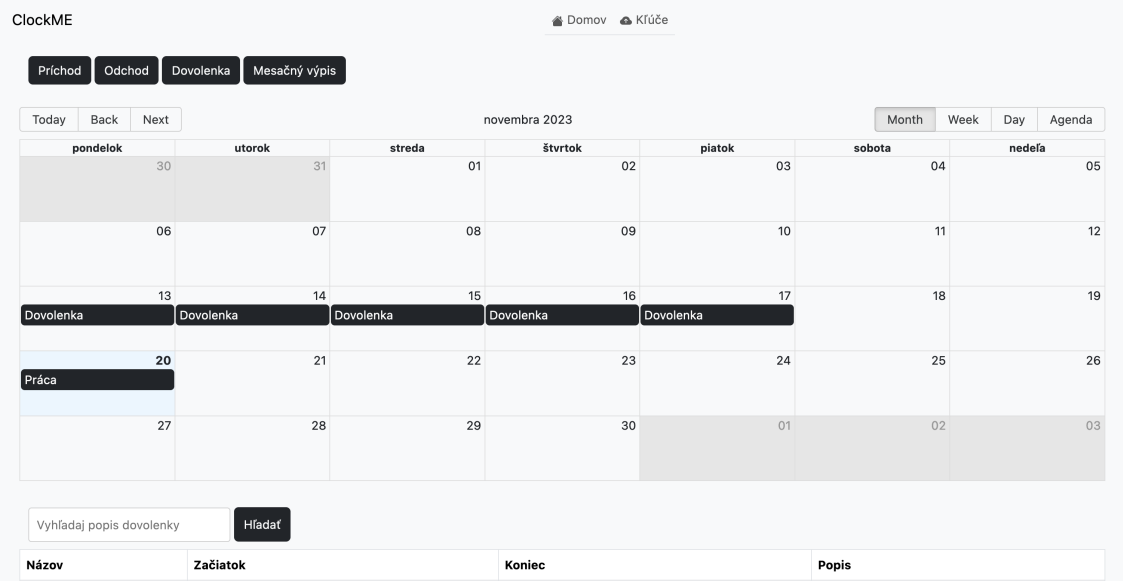


Obr. 2: Registračný formulár

Na predchádzajúcom obrázku môžeme vidieť príklad zobrazenia chybovej hlášky, ktorá hovorí o chýbajúcom zadaní emailu. Po stlačení tlačidla `Registrovať` sa zobrazí hláška o úspešnej registrácii. V prípade že používateľ so zadaným emailom už existuje, je zobrazená chybová hláška, že používateľ so zadaným emailom už existuje.

3.3 Hlavná stránka

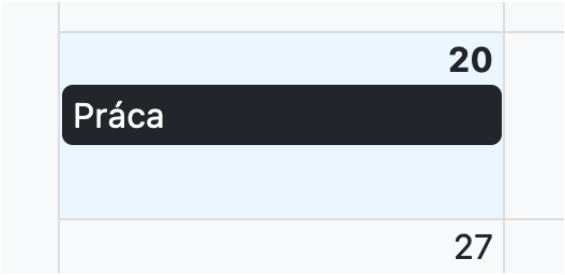
Hlavná stránka obsahuje základné funkcionality aplikácie ako je zaznamenanie príchodu, odchodu a dovolenky. Taktiež obsahuje kalendár, ktorý tieto záznamy prehľadne zobrazuje. Pod kalendárom sa nachádza vyhľadávacie pole ktorým je možné filtrovať tabuľku všetkých záznamov. Vizuál hlavnej obrazovky je zobrazený na nasledujúcom obrázku 3.



Obr. 3: Hlavná obrazovka

3.3.1 Príchod

Príchod používateľa do práce sa zadáva pomocou tlačidla `Príchod`, ktoré sa nachádza v hornej časti aplikácie. Po kliknutí na toto tlačidlo sa automaticky pošle request na pridanie záznamu do tabuľky a nový záznam sa zobrazí v kalendári aj v tabuľke pod ním s podrobným popisom.



Obr. 4: Zobrazenie práce v kalendári

Na obrázku 5 nižšie môžeme vidieť príklad záznamu príchodu, bez zadaného odchodu.

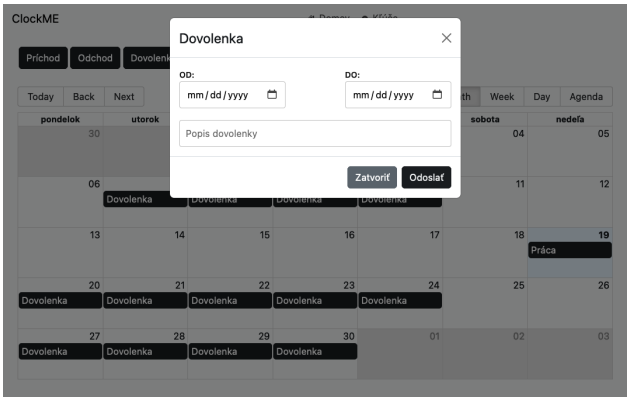
Obr. 5: Zobrazenie práce v tabuľke

3.3.2 Odchod

Odchod používateľa z práce sa zadáva pomocou tlačidla `Odchod` analogicky ako príchod. Po kliknutí na toto tlačidlo sa automaticky pošle request na aktualizovanie záznamu tabuľky a aktualizovaný záznam sa zobrazí v kalendári aj v tabuľke pod ním s podrobným popisom.

3.3.3 Dovolenka

Záznam o dovolenke sa zadáva pomocou tlačidla `Dovolenka`. Po kliknutí sa zobrazí modal, v ktorom je potrebné vyplniť dátum začiatku a konca dovolenky, v inom prípade sa záznam nevytvorí. Taktiež je možné zadať popis dovolenky pomocou textového poľa.



Obr. 6: Zobrazenie zadania dovolenky

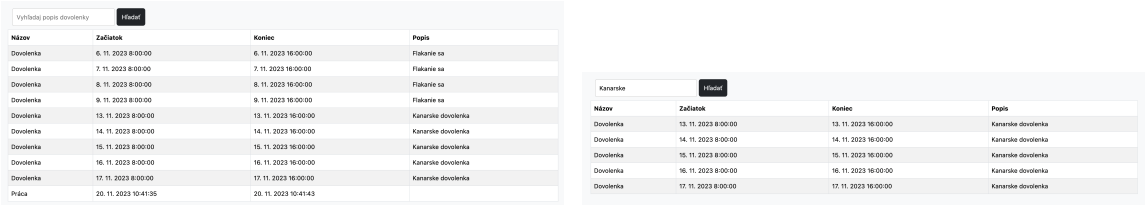
Po odoslaní sa záznam zobrazí v kalendári aj v tabuľke pod ním. Pre zobrazenie popisu dovolenky je možné použiť hover ponad jednotlivý záznam v kalendári, rovnako aj v stĺpci tabuľky. Príklad zobrazenia v tabuľke je na obrázku 7.

Názov	Začiatok	Koniec	Popis
Dovolenka	13. 11. 2023 8:00:00	13. 11. 2023 16:00:00	Kanarske dovolenka
Dovolenka	14. 11. 2023 8:00:00	14. 11. 2023 16:00:00	Kanarske dovolenka
Dovolenka	15. 11. 2023 8:00:00	15. 11. 2023 16:00:00	Kanarske dovolenka
Dovolenka	16. 11. 2023 8:00:00	16. 11. 2023 16:00:00	Kanarske dovolenka
Dovolenka	17. 11. 2023 8:00:00	17. 11. 2023 16:00:00	Kanarske dovolenka

Obr. 7: Zobrazenie dovolenky v tabuľke

3.3.4 Filtrovanie záznamov dovolenky

Implementácia vyhľadávania záznamov dovolenky je realizovaná pomocou jednoduchého vyhľadávacieho poľa pod kalendárom. Vložením popisu dovolenky do daného poľa sa následne v tabuľke pod poľom filtrujú záznami na základe popisu dovolenky. Príklad filtrovania je zobrazený na nasledovnom obrázku 8.



(a) Pred filtrovaním (b) Po filtrovaní

Obr. 8: Funkcionalita vyhľadávacieho poľa

V prípade použitia inej akcie sa záznamy zobrazia v tabuľke bez filtrovania.

3.3.5 Vymazanie záznamu

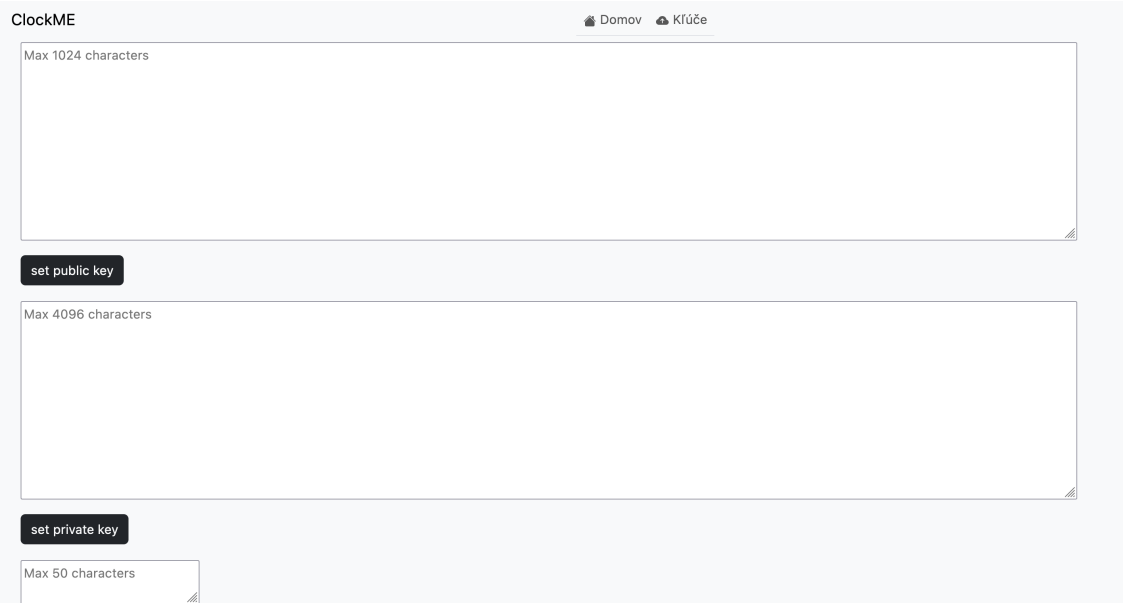
Vymazanie záznamu je implementovaný pomocou kliknutia na daný záznam v kalendári. Po kliknutí je záznam vymazaný z kalendára, tabuľky a rovnako aj z databázovej tabuľky.

3.3.6 Export mesačného prehľadu

Export mesačného prehľadu sme implementovali pomocou tlačidla `Mesačný výpis`, ktoré sa nachádza v hornej časti aplikácie pri ostatných funkcionalitách. Po kliknutí sa automaticky vyexportuje aktuálny stav tabuľky pod kalendárom do `pdf` súboru. V prípade aktuálneho filtrovania sa vyexportuje len filtrovaná tabuľka.

3.4 Zadanie privátneho a verejného kľúča

Ako sme už spomenuli vyššie, aplikácia po úspešnom prihlásení vyžaduje pre fungovanie manuálne zadanie privátneho a verejného kľúča. Sekcia `Kľúče` dostupná z hlavnej obrazovky obsahuje 3 textové polia pre zadanie verejného, privátneho kľúča a odtestovanie ich funkčnosti pomocou odoslania správy na server.



Obr. 9: Sekcia pre zadanie kľúčov

Testovacie kľúče sú uložené v zdrojovom kóde v priečinku `clientKeys` a je možné ich použiť bez potreby generovania nového kľúčového páru. Podrobný popis k funkcionalitám tejto časti sme popísali v predchádzajúcom zadaní.

Literatúra

[1] jspdf: A library to generate pdfs in javascript. `https://github.com/parallax/jsPDF`.

[2] react-big-calendar: An events calendar component built for react. `https://www.npmjs.com/package/react-big-calendar`.