



# Focus Time! dokumentace

Zpracovala: Ekaterina Gulina

# Obsah

<b>Obsah</b>	<b>2</b>
<b>Zadání</b>	<b>3</b>
<b>Uživatelská příručka</b>	<b>4</b>
Popis funkčnosti webu	4
Registrace	4
Přihlášení	4
Odhlášení	4
Použití časovače	4
Použití To-do seznamu	4
Stránkování seznamu úkolů	5
Ukázky UI webu	6
Header	6
Footer	6
Hlavní stránka – landing page	6
Registrace a přihlášení – login page	7
Personální stránka – personal page	8
Pracovní stránka – work page	9
<b>Popis implementace</b>	<b>10</b>
Popis hlavních rysů programu	10
Architektura	10
Funkčnosti	11
Zabezpečení	12
Zpracování dat	12
Pomocné aplikace	12
Vygenerovaná dokumentace ze zdrojového kódu	12

## Zadání

Udělat webovou aplikaci pro produktivitu a práci s možností použití časovače pro pomodoro techniku a to-do seznam s přidáním a odstraněním úkolů. Webové stránky budou obsahovat popis pomodoro techniky a základní informací o samotné aplikaci. Uživatelé se budou moci zaregistrovat a přihlásit se do aplikace pro ukládání svých seznamů a přístup k nim.

Kromě přidání a odstranění položek v seznamu, bude možné je také označit jako vyplněné, což bude následovně zobrazeno v uložených seznamech na personálních stránkách uživatelů.

Serverová část práce musí být napsána v PHP, klientská – pomocí HTML5 (s CSS styly) a JavaScriptu.

# Uživatelská příručka

## Popis funkčnosti webu

### Registrace

Každý anonymní uživatel má možnost vytvořit účet ve webové aplikaci *Focus time!*. Pro to je třeba přejít na stránku login\_page kliknutím na tlačítko *Sign up* na landing page nebo na tlačítko *Login* v hlavičce webu.

Pro registraci je třeba uvést username, e-mailovou adresu, heslo (a potvrdit ho opětovným zadáním) a datum narození (nepovinné). Pro dokončení registrace je třeba stisknout tlačítko *Create*.

Po úspěšné registraci se pod formulářem objeví nápis *Registration is successful!* a uživatel se bude moci přihlásit.

### Přihlášení

Každý zaregistrovaný uživatel má možnost se přihlásit do svého účtu. Pro to je třeba na stránce login\_page vyplnit své údaje (username a heslo) do formulářů *Welcome back!* a stisknout tlačítko *Log in*. Po úspěšném přihlašování uživatel bude přesměrován do stránky se svými údaji.

### Odhlášení

Každý přihlášený uživatel se může odhlásit pomocí tlačítka *Logout* v hlavičce stránky. Po odhlášení uživatel bude přesměrován na stránku registrace a přihlášení.

### Použití časovače

Na stránce work\_page každý uživatel může použít jednu ze základních funkcionalit webu – časovač.

Pro nastavení času je třeba použít šipky v sekci *Timer*. Šipka nahoru – zvětší čas o minutu, šipka dolů – zmenší čas o minutu. Po kliknutí na tlačítko *Start* timer začne běžet a zastaví se buď po opětovném kliknutí na tlačítko (které se promění na tlačítko *Stop*) nebo až po vypršení času.

### Použití To-do seznamu

Na stránce work\_page každý uživatel může použít druhou základní funkcionalitu webu – seznam úkolů.

Pro přidání nového úkolu je třeba vyplnit políčko *Add a task!* a stisknout tlačítko plus. Úkol se zobrazí na zelené tabuli v sekci *To-do list*.

Po kliknutí na vypsaný úkol – uživatel ho označí jako splněný a úkol bude zaškrtnán.

Po kliknutí na červený symbol *X* vedle vypsaného úkolu – úkol bude smazán ze seznamu.

Přihlášení uživatelé mohou své seznamy ukládat pomocí tlačítka *Save*. Nepřihlášené tuto možnost nemají.

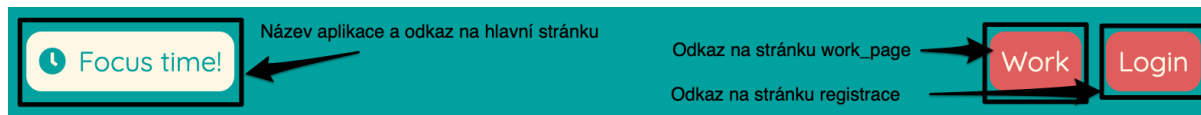
## Stránkování seznamu úkolů

Přihlášení uživatelé mají přístup ke své osobní stránce `personal_page`. Na této stránce uživatelé mohou vidět své uložené To-do seznamy. Pokud seznam obsahuje více než 3 položky, tento seznam se rozdělí na zvláštní stránky, které lze procházet pomocí šipek vpravo a vlevo pod vypsáním seznamem.

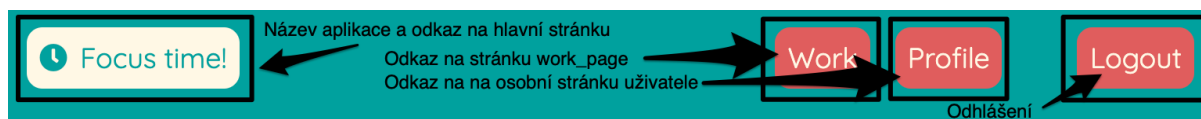
## Ukázky UI webu

Webové stránky používají písmo Quicksand z druhu sans-serif. Hlavními barvami jsou následující 4: #00A19D, #FFF8E5, #FFB344, #E05D5D.

### Header



Header pro nepřihlášené uživatele



Header pro přihlášené uživatele

### Footer

Styl footeru je pro všechny stránky stejný.



Footer

### Hlavní stránka – landing page

Vzhled hlavní stránky se liší podle toho, zda je uživatel přihlášen nebo ne. Hlavním rozlišením je tlačítko [Sign up/Profile](#).

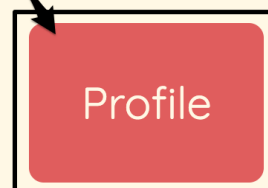


Landing page pro nepřihlášené uživatele

# Sign up to

Odkaz na na osobní stránku uživatele

- Run your own to-do lists
- Use Pomodoro timer
- Save your lists and notes!



*Landing page pro přihlášené uživatele*

## Registrace a přihlášení – login page

Stránka login\_page obsahuje 2 formuláře – registrace a přihlášení.

### Create account!

**Povinné údaje** Field is required!

**Nepovinné údaje** Date of Birth:

Založit účet

*Vytvořit nový účet*

**Welcome back!**

Povinné údaje Field is required!

Username

Password

[Log in](#) Přihlásit se

*Přihlásit se*

## Personální stránka – personal page

**Hello, admin1!**

Jméno uživatele

**Personal info**

E-mailová adresa uživatele

Email: admin1@mail.com

*Hlavní informace o uživateli*

**Your saved to-do!**

~~celebrate New Year~~

~~end semester work~~

be happy

[←](#) Předchozí stránka

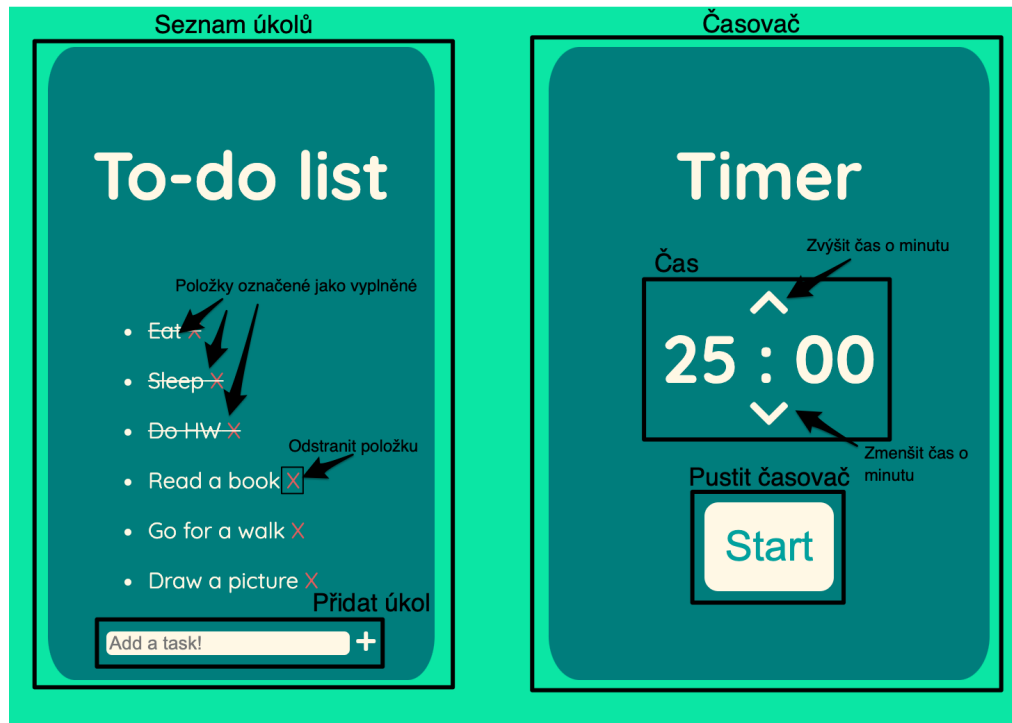
[→](#) Následující stránka

*Uložený seznam úkolů na personální stránce*

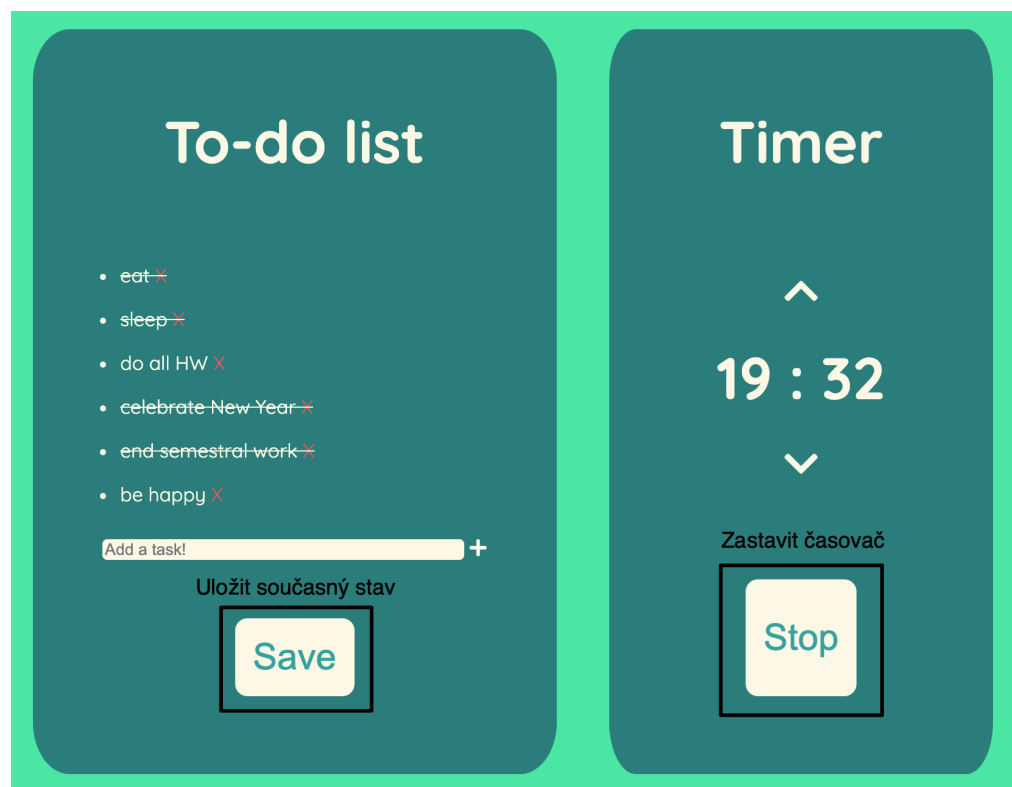


## Pracovní stránka – work page

Pracovní stránka má 2 vzhledy pro přihlášené a nepřihlášené uživatele – liší se tlačítkem **Save**, které umožňuje přihlášeným uživatelem uložit jejich To-do.



Work page pro nepřihlášené uživatele, timer je vypnutý



Work page pro přihlášené uživatele, timer je zapnutý

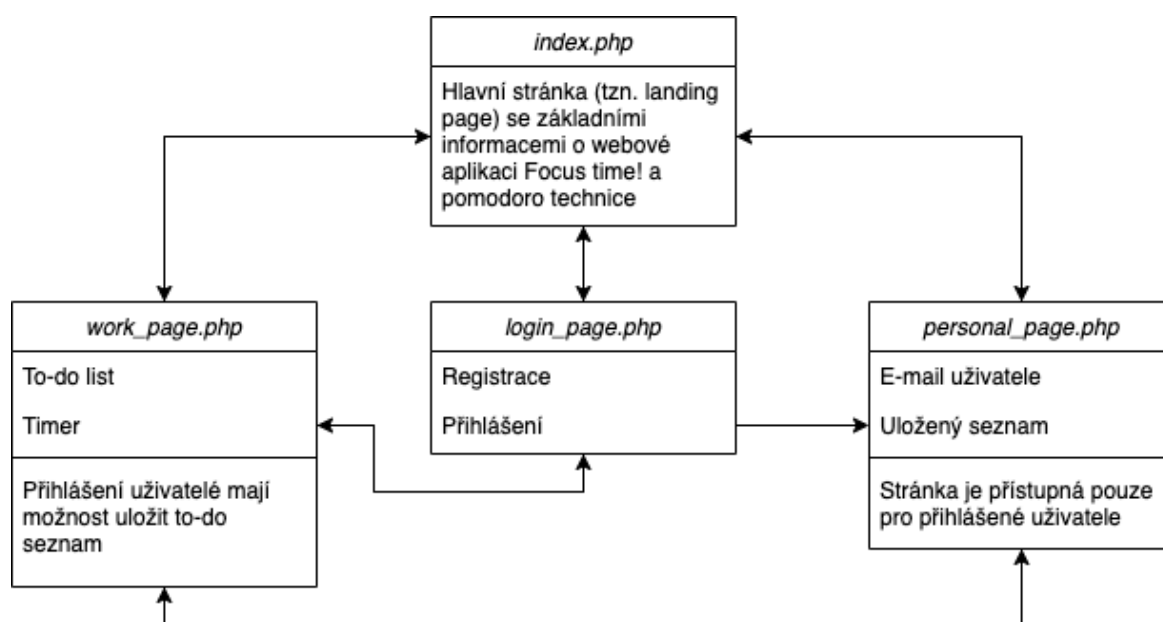
# Popis implementace

## Popis hlavních rysů programu

### Architektura

Serverová část aplikace se skládá z PHP stránek a databáze.

- Stránky: landing page (*index.html*), timer a To-do (*work\_page.php*), registrace a přihlášení (*login\_page.php*), stránka uživatele (*personal\_page.php*), osobní stránka (*personal\_page.php*) a stránka odhlášení (pouze pro přihlášené uživatele a po kliknutí přesměruje uživatele na stránku přihlášení).



UML mapa stránek

- Databáze je realizována pomocí JSON souboru *users.json*, ve kterém jsou data uživatelů (username, email, password). Pak pro každého uživatele je vytvořen vlastní soubor *\*jmeno\_uzivatele\*.json*, který v sebe nese uložený seznam úkolů (tento souboru zpracovává *process\_file.php*).

Frontend se skládá s JS skriptů, CSS stylů pro HTML5 a PHP templates pro opakující se HTML bloky.

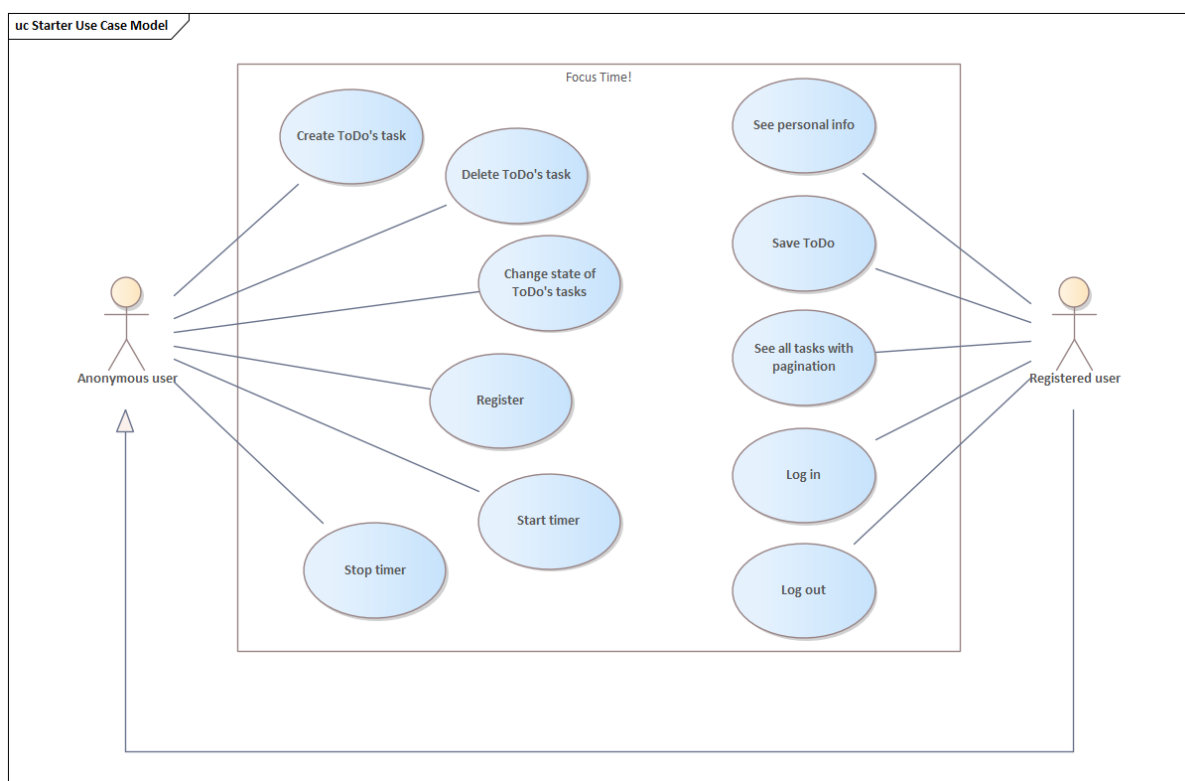
- CSS: *styles\_main.css* – styl pro *index.php*, *personal\_page.php* a *login\_page.php*. *work\_page.php* používá *styles\_work\_green.php*.
- JS: *script.js* soubor určuje chování timeru, to-do seznamu a třídy Todo (pro položky seznamu) spolu s předáním jednotlivých položek do cookie pro následující ukládání. *reg\_input\_control.js* provádí kontrolu uživatelského vstupu při registraci.
- PHP bloky: *footer.php*, *header\_main.php* (určen pro stránky *index.php*, *personal\_page.php* a *login\_page.php*), *header.php* (určen pro *work\_page.php*) a *header\_buttons.php* – nesou opakující se části HTML5 pro všechny stránky aplikace.

## Funkčnosti

Základní funkcionality webové aplikace *Focus time!*:

- Registrace uživatelů – každý uživatel se může zaregistrovat v aplikaci *Focus time!* na stránce login\_page. Username a e-mail mají být unikátní – zadané uživatelem hodnoty jsou kontrolovány jak na straně klienta (pomocí JS skriptů *req\_input\_control.js* a HTML5), tak i na serveru (pomocí PHP).
- Přihlášení – zaregistrované uživatele mohou se přihlásit na stránce login\_page vyplněním povinných údajů username a heslo. Vstupní data jsou kontrolovány na serveru.
- Odhlášení – je realizováno pomocí souboru *logout.php* a funguje po kliknutí na tlačítko *Logout* při přihlášení. Po odhlášení uživatel je přesměrován na stránku registrace.
- To-do list – seznam funguje na základě souboru *script.js* (kde jsou realizované přidání nových položek, výpis všech položek na stránku a předání seznamu pomocí cookie na server) a *work\_page.php* (kde je realizované ukládání dat na server do souboru *\*jmeno\_uzivatele\*.json*).
- Timer – práce timeru je zcela realizována pomocí souboru *script.js*.
- Použití stránkování pro prohlížení uloženého seznamu – je realizováno pomocí *personal\_page.php*.

Následující diagram znázorňuje jaké funkcionality jsou přístupné pro přihlášené a nepřihlášené uživatele (přihlášení uživatele mají funkcionality nepřihlášených a ještě několik výhod navíc):



Starter User Case Model

## Zabezpečení

Pro zabezpečení uživatelských dat a formulářů byly použity následující metody:

- XSS – proti Cross-Site Scripting útokům ke každé proměnné, která byla odeslána na server, byla použita PHP funkce `htmlspecialchars()`, která konvertuje určité symboly v HTML-entity; při předání nové položky do To-do seznamu v JS (`script.js`), zabezpečení uživatelského vstupu provádí funkce `document.createTextNode()`, která vytváří nový textový element.
- PHP CSRF – proti dvojímu odesílání stejného formuláře a Cross-Site Request Forgery byla použita metoda CSRF tokenu pro registraci a přihlašování uživatelů (`login_page.php`).

## Zpracování dat

Všechna políčka vstupů jsou kontrolována jak na straně klienta (např. `reg_input_control.js` – kontrola vstupu při registraci), tak i na straně serveru (`login_page.php` – při registraci a přihlašování). Po vyplnění formulářů registrace a po splnění kontroly se data uživatele zapisují do databáze `users.json`.

Po vyplnění formuláře přihlašování a splnění kontroly se provádí hledání uživatele v databázi.

U To-do seznamu předání nových položek je realizováno pomocí JS (`script.js`). Při kliknutí na tlačítko `Save` data seznamu jsou pomocí cookie předané na server a zapisují se do souboru `*username*.json` (to dělají `work_page.php` a `process_file.json`).

## Pomocné aplikace

U webové aplikace bylo použito následující:

- [Fontawesome](#) – ikonky.
- [Google Fonts](#) – písmo.
- [Favicon](#) – tvorba faviconu.

## Vygenerovaná dokumentace ze zdrojového kódu

[Odkaz na Doxygen](#)