

Hahaland

Popis implementace

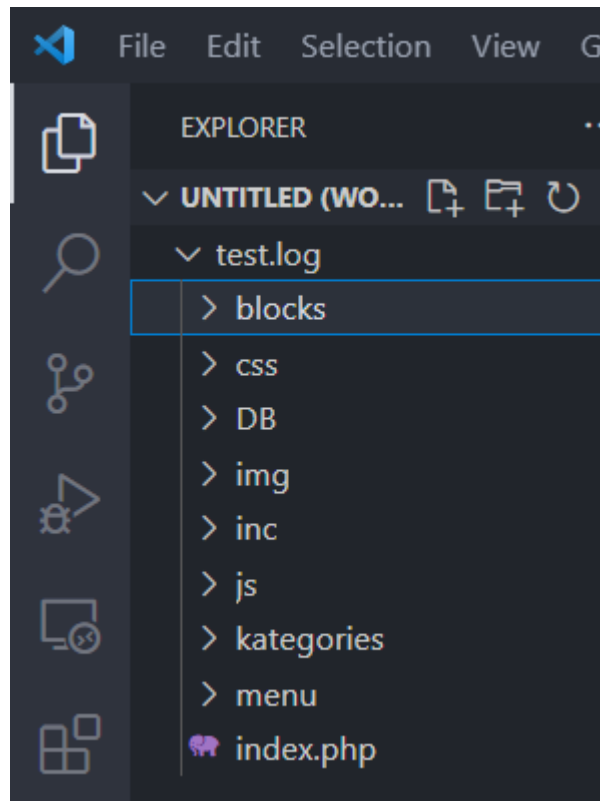
Autor: Chontosh Andrii

OBSAH

Struktura webu.....	3
HTML.....	4
CSS.....	5
JavaScript.....	5
Databáze.....	6
PHP.....	6
Registrace a autorizace.....	7
Přidání vtipů.....	9
Validace.....	10
Hodnocení.....	11
Řazení.....	13

Struktura webu

Celý kód webu byl v závislosti na funkčnosti rozdělen do 8 složek. Složka "bloky" obsahuje části kódu, které se opakují na dalších stránkách. Složka 'css' obsahuje styly webových stránek. Složka 'db' obsahuje databázové soubory. Složka 'img' obsahuje obrázky používané na webu. Složka 'ins' obsahuje soubory, které poskytují funkce na straně serveru. Složka 'js' obsahuje soubory, které poskytují funkce na straně klienta. Složka 'categories' obsahuje stránky kategorií webu. Složka 'menu' obsahuje stránky položek nabídky.



HTML

Všude, kde to bylo nutné, byly použity sémantické značky, jako je *section*, *main*, *nav*, *footer*, *header*. Pole formuláře jsou ověřena pomocí HTML.

```
<div class="div__error">
  <input type="text" name="name" value tabindex="1" id="name" class="form__input surname" pattern="^[a-zA-Z]{2,20}\s*[a-zA-Z]{2,20}$" required>
</div>
</div>
<div class="form-group"><flex>
  <label for="email">Email</label>
  <div class="div__error">
    <input type="email" placeholder="Email slouží jako login" value name="email" id="email" class="form__input email" tabindex="2" pattern="^[a-zA-Z0-9]{2,20}$" required>
  </div>
</div>
<div class="form-group"><flex>
  <label for="psw">Heslo</label>
  <div class="div__error">
    <input type="password" placeholder="Minimálně 8 znaků" value name="psw" id="psw" class="form__input password" pattern="^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#$%^&*])(?=[a-zA-Z0-9!@#$%^&*]{8,})$" required>
  </div>
</div>
</div>
```

CSS

Zajímavým řešením je skrytí registračních a autorizačních formulářů, které bylo implementováno pomocí vlastnosti *transform* a pseudotřídy *:target*. Animace pomocí vlastnosti *transform* byla provedena také v tlačítku hodnocení, řazení do kategorií a také opuštění registrační stránky kde byly použity pseudotřídy *:hover* a *:active* a pseudoprvky *:after* a *:before*.

```
93
94 /*Registration */
95
96 .registration-row{
97     position: fixed;
98     top: 0;
99     left: 0;
100    right: 0;
101
102    transform: translateX(-100%);
103 }
104
105 .registration-row:target{
106     transform: translateX(0);
107     z-index: 1;
108     background-image: url('../img/selmon.jpg');
109     background-size: cover;
110 }
111
112 .registration-row{
113     display: flex;
114     text-align: center;
115     align-items: center;
116     width: 100%;
117     height: 100%;
118 }
119 .registration{
120     color: #00a650;
121     margin: 0 auto 0 auto;
122     border-radius:10px;
123     box-shadow:4px 4px 15px rgba(0, 0, 0, 0.2);
124     background-color: #fff;
125 }
126
```

Pomocí CSS je web přizpůsoben i pro mobilní zařízení, k tomu bylo použito *@media*.

JavaScript

JavaScript na stránce poskytuje ověření formuláře na straně klienta. K tomu byla použita metoda *addEventListener()* s hodnotou prvku *'submit'* a *'input'*. Ve druhém případě jsou hodnoty zadávaných dat porovnány s

hodnotou v atributu **“pattern”** odpovídajícího pole HTML, který je poskytnut pomocí metody **getAttribute()**.

```
167
168 function inputChack(el){
169     const inputValue = el.value;
170     const inputReg = el.getAttribute('pattern');
171     const reg = new RegExp(inputReg);
172     if(!reg.test(inputValue)){
173         if(el.getAttribute('name') == 'name'){
174             showErr(inputName, 'Pouze latinská písmena');
175         }
176         if(el.getAttribute('name') == 'email'){
177             showErr(inputEmail, 'Nesprávný formát e-mailu');
178         }
179         if(el.getAttribute('name') == 'pswHash'){
180             showErr(inputPass, 'Musí být velké malé písmeno a číslo');
181         }
182         if(el.getAttribute('name') == 'phone'){
183             showErr(inputPhone, 'Musí být devět čísel');
184         }
185     }
186
187     if (inputPassre.value != inputPass.value || !reg.test(inputValue)){
188         if(el.getAttribute('name') == 'psw_repeat'){
189             showErr(inputPassre, 'Musí být stejné jako předchozí');
190         }
191     } else {
192         el.classList.remove('error');
193     }
194 }
```

Databáze

Databáze webových stránek je reprezentována dvěma soubory ve formátu **.json**. První soubor **notes.jsone** obsahuje informace o registrovaném uživateli a druhý soubor **anekdots.json** obsahuje anekdoty a informace s nimi související.

PHP

■ Registrace a autorizace

Během registrace jsou všechna pole formuláře očištěna od symbolů, které mohou poškodit web (*registration.php*).

```
1  <?php
2  require '_fileDB.php';
3
4
5  function clear_data($val)
6  {
7      $val = trim($val);
8      $val = stripslashes($val);
9      $val = htmlspecialchars($val);
10     $val = strip_tags($val);
11     return $val;
12 }
13
14 $name = $email = $psw = $psw_repeat = $phone = "";
15
16 if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['addNote'])) {
17     $name = clear_data($_POST["name"]);
18     $email = clear_data($_POST["email"]);
19     $psw = clear_data($_POST["pswHash"]);
20     $psw_repeat = clear_data($_POST["psw_repeat"]);
21     $phone = clear_data($_POST["phone"]);
22 }
```

Poté proběhne validace, o které bude řeč později. Pokud je ověření úspěšné, data se uloží do databáze, kde poběží tzv. solení hesla (*_fileDB.php*).

```

1  <?php
2
3  $filePath_notes = __DIR__ . '/../DB/notes.json';
4
5  $data = json_decode(file_get_contents($filePath_notes), true);
6
7  if ($data == NULL) {
8      $data = array();
9      $encodedData = json_encode($data);
10     file_put_contents(__DIR__ . '/../DB/notes.json', $encodedData);
11 }
12
13
14 function addNote($data, $name, $email, $psw, $psw_repeat, $phone, $volil_a)
15 {
16     global $filePath_notes;
17
18     $newNote = array(
19         'id' => uniqid(),
20         'name' => $name,
21         'email' => $email,
22         'pswHash' => password_hash($psw, PASSWORD_DEFAULT),
23         'psw_repeat' => password_hash($psw_repeat, PASSWORD_DEFAULT),
24         'phone' => $phone,
25         'volil' => $volil_a
26     );
27
28     array_push($data, $newNote);
29     $encodedData = json_encode($data);
30     file_put_contents($filePath_notes, $encodedData);
31 }
32

```

Autorizace (*login.php*):

```

15
16 if (isset($_POST['login'])) {
17     $email = htmlspecialchars($_POST['email']);
18     $psw = htmlspecialchars($_POST['psw']);
19     $user = getUserByEmail($res, $email);
20
21     if ($email && $psw) {
22         if ($user) {
23             if (password_verify($psw, $user['pswHash'])) {
24                 session_start();
25                 $_SESSION['username'] = $user['name'];
26                 $_SESSION['id'] = $user['id'];
27                 setcookie('user', $user['email'], time() + 3600, "/");
28                 header("Location: ../index.php?page=0");
29             } else {
30                 session_start();
31                 $_SESSION['message'] = 'Nesprávné heslo nebo e-mailová adresa.';
32                 header("Location: ../index.php#authorization-row");
33             }
34         } else {
35             session_start();
36             $_SESSION['message'] = 'Nesprávné heslo nebo e-mailová adresa.';
37             header("Location: ../index.php#authorization-row");
38         }
39     } else {
40         session_start();
41         $_SESSION['message'] = 'Nesprávné heslo nebo e-mailová adresa.';
42         header("Location: ../index.php#authorization-row");
43     }
44
45     unset($_SESSION['hello']);
46 }

```


■ Přidání vtipů

Příjem informací z formuláře pokračuje zpracováním nežádoucích znaků pomocí funkce `htmlspecialchars()`. (*addAnekdot.php*):

```
2
3 require 'DBanek.php';
4
5 if (isset($_POST['addAnekdot'])) {
6     session_start();
7     if (!empty($_POST['anekdot'])) {
8         $anekdot = htmlspecialchars($_POST['anekdot']);
9         $kat = $_POST['kategorie_naw'];
10        $hodnota = 0;
11        $point = [];
12        $id = "a" . uniqid();
13        $width = 0;
14        $time = date('Y-m-d H:i:s');
15        $volil = [];
16
17        $_SESSION['anekdot'] = $anekdot;
18
19        addAnekdot($data, $id, $_SESSION['username'], $kat, $anekdot, $point, $hodnota, $width, $time, $volil);
20
21        header('Location: ../index.php#');
22    } else {
23        $_SESSION['anekdot_error'] = 'Pole musí být vyplněno.';
24        header("Location:" . $_SERVER['HTTP_REFERER']);
25    }
26 }
27
28 }
```

a uložení do databáze(*DBanek.php*):

```
1 <?php
2
3 $filePath = __DIR__ . '/../DB/anekdots.json';
4 $data = json_decode(file_get_contents($filePath), true);
5
6 if ($data == NULL) {
7     $data = array();
8     $encodedData = json_encode($data);
9     file_put_contents(__DIR__ . '/../DB/anekdots.json', $encodedData);
10 }
11
12 function addAnekdot($data, $id, $name, $kat, $anekdot, $point, $hodnota, $width, $time, $volil)
13 {
14     global $filePath;
15
16     $newNote = array(
17         'id' => $id,
18         'name' => $name,
19         'kategorie_naw' => $kat,
20         'anekdot' => $anekdot,
21         'stars' => $point,
22         'hodnota' => $hodnota,
23         'width' => $width,
24         'time' => $time,
25         'volil' => $volil
26     );
27
28     array_push($data, $newNote);
29     $encodedData = json_encode($data);
30     file_put_contents($filePath, $encodedData);
31 }
32 }
```

■ Validate

Všechna pole formuláře jsou ověřena na serveru. Při zadání neplatných hodnot se zobrazí upozornění s uvedením chyby. Na registračním formuláři hodnoty se porovnávají pomocí funkce `preg_match()`, zkontroluje se také, zda je pole prázdné, pokud ano, zobrazí se odpovídající chyba.

```
35 $flag = 0;
36 if (isset($_POST['addNote'])) {
37     if ($_SERVER['REQUEST_METHOD'] == 'POST') {
38         if (!preg_match($pattern_name, $name)) {
39             $err['name'] = 'Pouze latinská písmena';
40             $flag = 1;
41         }
42         if (empty($name)) {
43             $err['name'] = 'Toto pole je povinné.';
44             $flag = 1;
45         }
46         if (!preg_match($pattern_email, $email)) {
47             $err['email'] = '<p class="msg">Nesprávný formát e-mailu</p>';
48             $flag = 1;
49         }
50         if (empty($email)) {
51             $err['email'] = '<p class="msg">Toto pole je povinné.</p>';
52             $flag = 1;
53         }
54         if (!preg_match($pattern_pass, $psw)) {
55             $err['pswHash'] = '<p class="msg">Musí být velké malé písmeno a číslo</p>';
56             $flag = 1;
57         }
58         if (empty($psw)) {
59             $err['pswHash'] = '<p class="msg">Toto pole je povinné.</p>';
60             $flag = 1;
61         }
62         if ($_POST['pswHash'] != $_POST['psw_repeat']) {
63             $err['psw_repeat'] = '<p class="msg">Musí být stejné jako předchozí</p>';
64             $flag = 1;
65         }
66         if (empty($_POST['psw_repeat'])) {
67             $err['psw_repeat'] = '<p class="msg">Toto pole je povinné.</p>';
68             $flag = 1;
69         }
70         if (!preg_match($pattern_phone, $phone)) {
```

Autorizační formulář nejprve vyhledá uživatele s platnou e-mailovou adresou a poté zkontroluje heslo.

```

2
3 function getUserByEmail($data, $email)
4 {
5     foreach ((array)$data as $user) {
6         if ($email == $user['email']) {
7             return $user;
8         }
9     }
10    return NULL;
11 }

```

```

11 }
12 if (isset($_POST['login'])) {
13     $email = htmlspecialchars($_POST['email']);
14     $psw = htmlspecialchars($_POST['psw']);
15     $user = getUserByEmail($res, $email);
16
17     if ($email && $psw) {
18         if ($user) {
19             if (password_verify($psw, $user['pswHash'])) {
20                 session_start();
21                 $_SESSION['username'] = $user['name'];
22                 $_SESSION['id'] = $user['id'];
23                 setcookie('user', $user['email'], time() + 3600, "/");
24                 header("Location: ../index.php?page=0");
25             } else {
26                 session_start();
27                 $_SESSION['message'] = 'Nesprávné heslo nebo e-mailová adresa.';
28                 header("Location: ../index.php#authorization-row");
29             }
30         } else {
31             session_start();
32             $_SESSION['message'] = 'Nesprávné heslo nebo e-mailová adresa.';
33             header("Location: ../index.php#authorization-row");
34         }
35     } else {
36         session_start();
37         $_SESSION['message'] = 'Nesprávné heslo nebo e-mailová adresa.';
38         header("Location: ../index.php#authorization-row");
39     }
40
41     unset($_SESSION['hello']);
42 }

```

■ Hodnocení

Pro zajištění procesu hodnocení tak, aby jeden uživatel mohl hodnotit každý vtíp pouze jednou, je k metodě “post” tlačítka odeslání formuláře přidáno “id” vtípu. Tato hodnota je také přidána do databáze uživatelů v poli “volil” a ‘id’ uživatele je rovněž přidána do databáze vtípů. Pokud tedy samostatný vtíp obsahuje v databázi “id” uživatele v poli “volil” a uživatel obsahuje “id” vtípu v poli “volil”, bude tento vtíp uveden jako hodnocení a hodnotící formulář se uživateli skryje, místo toho se mu zobrazí zpráva. Takovou logiku používají registrovaní uživatelé a u neregistrovaných se při vyhodnocování jednoduše vytvoří cookie a následně se kontroluje jeho přítomnost.

```

if (isset($_POST['add_rating'] . $res[$i]['id'])) {
    if (!empty($_POST['stars'])) {
        array_push($res[$i]['stars'], $_POST['stars']);
        $sum = 0;
        $count = 0;
        foreach ($res[$i]['stars'] as $item) {
            $sum = $sum + $item;
            $count++;
        }
        $hodnoceni = round($sum / $count, 2);

        $res[$i]['hodnota'] = $hodnoceni;

        $width = $hodnoceni * 40.2;
        $res[$i]['width'] = $width;
        $timec = 365 * 24 * 60 * 60;
        if (isset($_SESSION['id'])) {
            array_push($res[$i]['volil'], $_SESSION['id']);
        } else {
            array_push($res[$i]['volil'], 0);
        }

        setcookie($res[$i]['id'], ' ', time() + $timec);
        for ($j = 0; $j < count($res2); $j++) {
            if (isset($_SESSION['id'])) {
                if ($_SESSION['id'] == $res2[$j]['id']) {
                    array_push($res2[$j]['volil'], $res[$i]['id']);
                    $newJsonString2 = json_encode($res2);
                    file_put_contents(__DIR__ . '/../DB/notes.json', $newJsonString2);
                }
            } else {
                array_push($res2[$j]['volil'], 0);
            }
        }
    }
    $newJsonString = json_encode($res);
}

```

```

18
19         <?php if (isset($_SESSION['id'])) {
20             if (in_array($_SESSION['id'], $res[$i]['volil'])) {
21                 echo '<form class="anekdoty__rang" method="POST">
22                     <label for="stars"></label>

```

```

35     }
36     } else {
37         if (!isset($_COOKIE[$res[$i]['id']])) {
38             echo '<form class="anekdoty__rang" method="POST">
39                 <label for="stars"></label>
40                 <select name="stars">

```

■ Řazení

K třídění slouží funkce Bublínového řazení, které se volá na začátku stránky. Pro třídění do kategorií se nejprve setřídí celá databáze a poté se vytvoří pole *\$kategorie*, do kterého se přidávají pouze anekdoty vybrané kategorie.

```
3
4 function sortTime($res)
5 {
6     for ($i = 0; $i < count($res); $i++) {
7         for ($j = 0; $j < count($res); $j++) {
8             if ($res[$i]['time'] < $res[$j]['time']) {
9                 $new = $res[$i];
10                $old = $res[$j];
11                $res[$j] = $new;
12                $res[$i] = $old;
13
14                $newJsonString = json_encode($res);
15                file_put_contents(__DIR__ . '/../DB/anekdots.json', $newJsonString);
16            }
17        }
18    }
19
20    return $res;
21 }
```

```
16
17 $kategorie = [];
18
19
20
21 if (isset($_POST['sort'])) {
22     if ($_POST['sort'] == 'Podle času') {
23         sortTimeNew($res);
24     }
25     if ($_POST['sort'] == 'Podle hodnoceni') {
26         sortBest($res);
27     }
28     echo("<meta http-equiv='refresh' content='0'>");
29 }
30
31
32 for ($i = 0; $i < count($res); $i++) {
33     if ($res[$i]['kategorie_naw'] == 'Černý humor')
34         array_push($kategorie, $res[$i]);
35     $_SESSION['kategorie'] = $res[$i]['kategorie_naw'];
36 }
37 $page__count = floor(count($kategorie) / $count);
38
```