



STŘEDNÍ ŠKOLA TECHNICKÁ A EKONOMICKÁ, BRNO,
PŘÍSPĚVKOVÁ ORGANIZACE
18–20–M/01 Informační technologie

TEMNÝ PÁN SAURON

PROJEKT WEBOVÉ STRÁNKY

ADAM FOLTÝN

Předmět: Programové vybavení

Třída: IT3B

Vytvořeno v roce 2016 a 2017 v Brně

Obsah

1. Výběr tématu:.....	3
2. Návrh a grafický design:.....	4
3. Popis layoutu:.....	5
4. Screen stránky:	6
5. Dynamická struktura:	7
6. Použité nástroje:	18
7. Závěr	19
8. Zdroje:	20

1. Výběr tématu:

Saurona jsem si vybral, protože knižní svět Hobita a Pána prstenů je velice zajímavý. Ačkoliv je Sauron hlavní zápornou postavou v knižní trilogii Pán prstenů, tak o něm zase není tolik známo, proto jsem se rozhodl zpracovat historii tohoto druhého temného pána Středozemě a dalších duchů, kteří byli s ním nebo se Středozemí byli významně spojeni, včetně stvoření Prstenů moci i jejich osudy.

2. Návrh a grafický design:

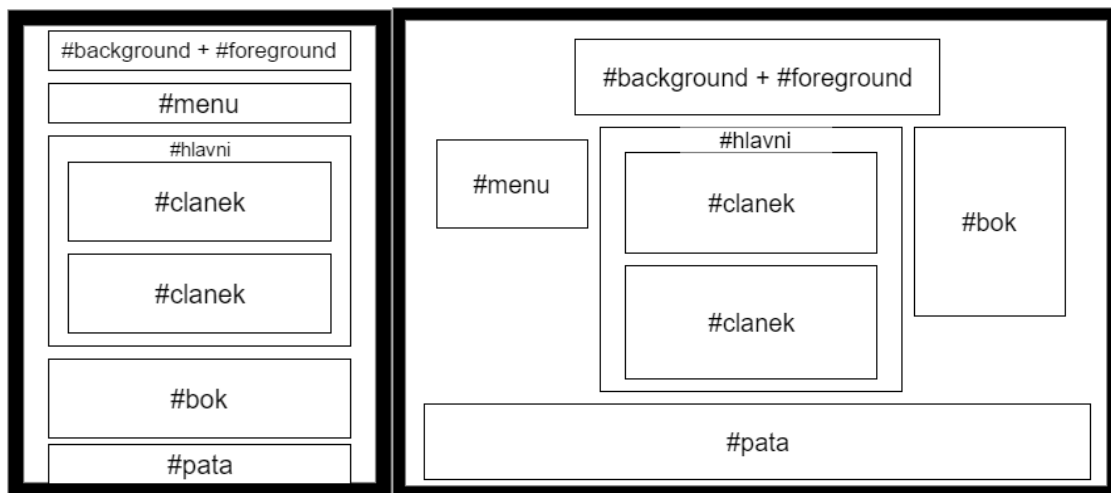


Tento grafický design jsem si vybral, protože mám rád šedou a jí podobné odstíny. Současně si myslím, že se výběr barev hodí k webu o démonické bytosti.

3. Popis layoutu:

Smartphone:

Desktop:



Za hashtagy jsou názvy použitých divů a jejich účel:

#background + #foreground: Logo

#menu: Zde si vybíráte stránku skrze tlačítka

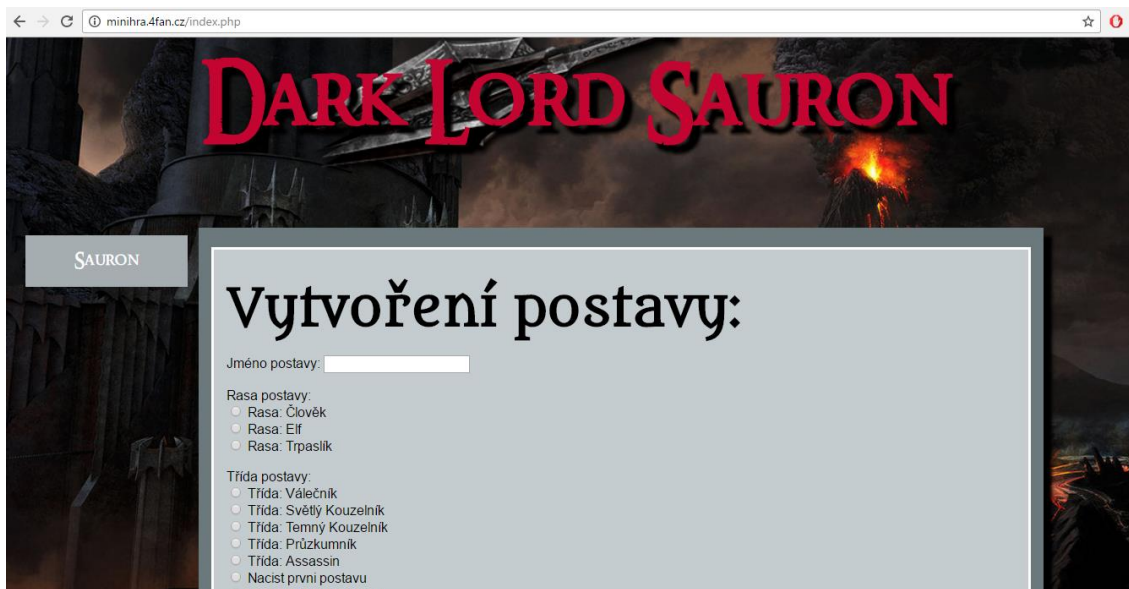
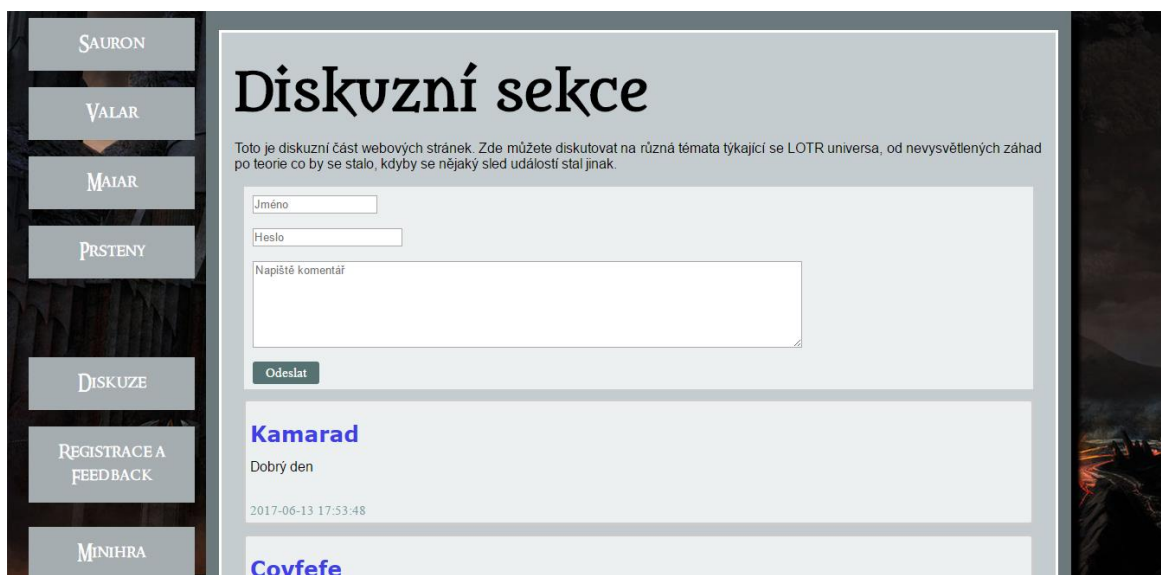
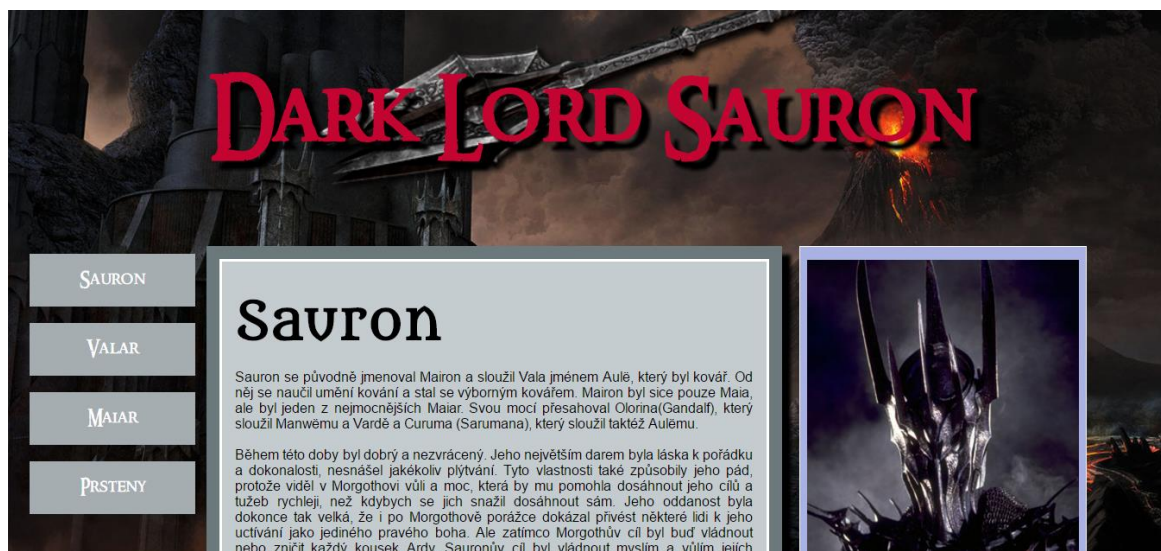
#hlavni: Slouží pro seskupení článků v jedné rovině

#clanek: Je do něj vložen hlavní obsah, každý článek je oddělen

#bok: Panel se základními informacemi o Sauronovi

#pata: Údaje o autorovi, linky na stáhnutí a validaci

4. Screen stránky:



5. Dynamická struktura:

```
<?php

        if (isset($_GET['stranka'])) {

            require($_GET['stranka'] . '.php'); }

else {

    require('sauron' . '.php');    }

?>
```

Výše vypsáný skript se nalézá v souboru index.php a řeší načítání obsahu z dalších souborů jako je například soubor feedback.php. Pokud je splněna podmínka tím, že je zadán cílový soubor kliknutím na tlačítko, skript načte jeho obsah, pokud ale podmínka splněna není, skript načte obsah souboru sauron.php.

```
<?php

$zdroj=$_GET['stranka'];

switch($zdroj){

    case "sauron":

        echo '<p >

        Sauron. In: The Lord of the Rings Wiki: Fandom powered by
Wikia [online]. [cit. 28.11.2016]. Dostupné z: <a
href="http://lotr.wikia.com/wiki/Sauron"
target="_blank">http://lotr.wikia.com/wiki/Sauron</a>

        </p>';

        break;

        case "valar":

            echo '<p >

            Valar. In: The Lord of the Rings Wiki: Fandom powered by
Wikia [online]. [cit. 28.11.2016]. Dostupné z: <a
href="http://lotr.wikia.com/wiki/Valar"
target="_blank">http://lotr.wikia.com/wiki/Valar </a>

            </p>';

            break;

            case "maiar":

                echo '<p >Maiar. In: The Lord of the Rings Wiki: Fandom
powered by Wikia [online]. [cit. 28.11.2016]. Dostupné z: <a
href="http://lotr.wikia.com/wiki/Maiar"
target="_blank">http://lotr.wikia.com/wiki/Maiar</a>

                </p>';

                break;

                case "rings":

                    echo '<p >Rings of Power. In: The Lord of the Rings Wiki:
Fandom powered by Wikia [online]. [cit. 28.11.2016]. Dostupné
z: <a href="http://lotr.wikia.com/wiki/Rings_of_Power"
target="_blank">http://lotr.wikia.com/wiki/Rings_of_Power</a>

                    </p>';
```

```
break;

case "feedback":

    echo '<p> Markle, B (2016) PHPMailer (Verze 5.2.0) [Source
code]. Dostupné z: <a
href="http://www.inmotionhosting.com/support/email/send-email-
from-a-page/using-phpmailer-to-send-mail-through-
php">http://www.inmotionhosting.com/support/email/send-email-
from-a-page/using-phpmailer-to-send-mail-through-php</a>

</p>';

break;

default:

    echo '<p >

    Sauron. In: The Lord of the Rings Wiki: Fandom powered by
Wikia [online]. [cit. 28.11.2016]. Dostupné z: <a
href="http://lotr.wikia.com/wiki/Sauron"
target="_blank">http://lotr.wikia.com/wiki/Sauron</a>

</p>';

}

?>
```

Skript řeší která stránka je momentálně zobrazená, podle toho jí zobrazí citaci v patičce.

```
<?php

// $name a $message jsou data,

// která byla sem přenesena z php stránky obsahující feedback
formulář

$name = $_REQUEST['name'] ;

$message = $_REQUEST['message'] ;

$vzhled = $_REQUEST['vzhled'] ;

$tema = $_REQUEST['tema'] ;

$kvalita = $_REQUEST['kvalita'] ;

$ns = "Zpravu napsal: " . $name;

require('PHPMailer_5.2.0/class.phpmailer.php');

$mail = new PHPMailer();

// nastaví používání protokolu SMTP

$mail->IsSMTP();

$mail->SMTPDebug = 2;

$mail->SMTPAuth = true;      // zapne SMTP autentizaci

$mail->SMTPSecure = 'ssl';

$mail->Host = "smtp.gmail.com";

$mail->Port = 465;

$mail->IsHTML(true);

// Při posílání e-mailu skrze PHPMailer, musíte použít validní
e-mailovou adresu

// Pro tento případ jsem vytvořil tuto e-mailovou adresu:

$mail->Username = "saursend@gmail.com"; // SMTP uživatelská
adresa

$mail->Password = "StopBeingAPleb"; // SMTP heslo
```

```
// Adresa e-mailu kam budeme posílat feedback
$mail->AddAddress("saurget@gmail.com");

// nastaví word wrap (zalomení dlouhého slova) na 50 znaků
$mail->WordWrap = 50;

// nastaví e-mailu formát HTML
$mail->IsHTML(true);

$mail->Subject = "Obdrzel jste feedback z vasi stranky!";

// $message obsahuje uživatelskou zprávu
// obsah $message nastavujeme
// na vrcholu této stránky příkazem:
// $message = $_REQUEST['message'] ;

$mail->Body = $ns . '<br> <br> Hodnoceni: <br> vzhled: '.
    $vzhled. '<br> tema: '. $tema. '<br> kvalita zpracovani
    informaci: ' . $kvalita . '<br> <br> feedback: ' . $message;

$mail->AltBody = $ns . '<br> <br> Hodnoceni: <br> vzhled: '.
    $vzhled. '<br> tema: '. $tema. '<br> kvalita zpracovani
    informaci: ' . $kvalita . '<br> <br> feedback: ' . $message;

if(!$mail->Send())
{
    echo "Zpráva nemohla být odeslána. <p>";
    echo "Mailer Error: " . $mail->ErrorInfo;
    echo ("Návrat: ");
    exit; }

echo ("Zpráva odeslána, návrat: ");

?>
```

Výše vypsaný skript je ze souboru email.php a řeší zasílání feedbacku přes SMTP skrze jeden Gmailový účet na druhý Gmailový účet. Na začátku skriptu se uloží zadaná data ve formuláři do proměnných, nastaví se připojení, například číslo portu a příjemce. Ke konci skriptu se do těla mailu nahraje obsah a e-mail se odešle, zkontroluje se, zda byl e-mail odeslán a v případě neúspěchu se zobrazí příčina chyby.

```
<?php

$con=mysqli_connect("localhost","root","","comment");
//připojuje se k databázi

    // Check connection

    if (mysqli_connect_errno())
    {

        echo "Failed to connect to MySQL: " .
mysqli_connect_error(); //pokud nastala chyba vypise error,
jinak se připoji

    }

?>
```

Slouží k připojení k databázi jménem comment jako root bez hesla na localhostu.

```
<?php
```

```
$name = $_REQUEST['name']; //vytahne si jmeno a obsah komentare
```

```
$comments = $_REQUEST['comments'];
```

```
if (!$name) {$name="anonym";} //pokud je jmeno nevyplnene, pak se nastavi na anonym
```

```
require("db/db.php"); //spoji se s php souborem, který zajistuje připojení k databázi
```

```
mysqli_query($con, "INSERT INTO comments(name, comments)  
VALUES('$name','$comments')"); //vlozi do databaze data
```

```
$result = mysqli_query($con, "SELECT * FROM comments ORDER BY id ASC"); //nacte data z  
databaze
```

```
while($row=mysqli_fetch_array($result)){
```

```
echo "<div class='comments_content'>";
```

```
echo "<h1>" . $row['name'] . "</h1>"; //postupne data vypisuje
```

```
echo "<p>" . $row['comments'] . "</p><br>";
```

```
echo "<h3>" . $row['date_publish'] . "</h3>";
```

```
echo "</div>";
```

```
}
```

```
mysqli_close($con); //ukonci spojeni s databazi
```

```
?>
```

Vloží odeslaný komentář do databáze a následně jej vypíše, v 1. řádku bude jméno, ve 2. bude obsah komentáře a ve 3. bude čas a datum odeslání komentáře. Ke konci spojení s databází ukončí.

```
<?php
```

```
require("db/db.php"); //spoji se s php souborem, který zajistuje připojení k databázi

$result = mysqli_query($con, "SELECT * FROM comments ORDER BY id ASC"); //nacte data z
databáze, bude je vypisovat postupne podle id od nejmensi po nejvetsi

while($row=mysqli_fetch_array($result)){

echo "<div class='comments_content'>"; //radek po radku vypisuje vysledky kazdy do sve
tabulky

echo "<h1>" . $row['name'] . "</h1>";

echo "<p>" . $row['comments'] . "</p><br>";

echo "<h3>" . $row['date_publish'] . "</h3>";

echo "</div>";

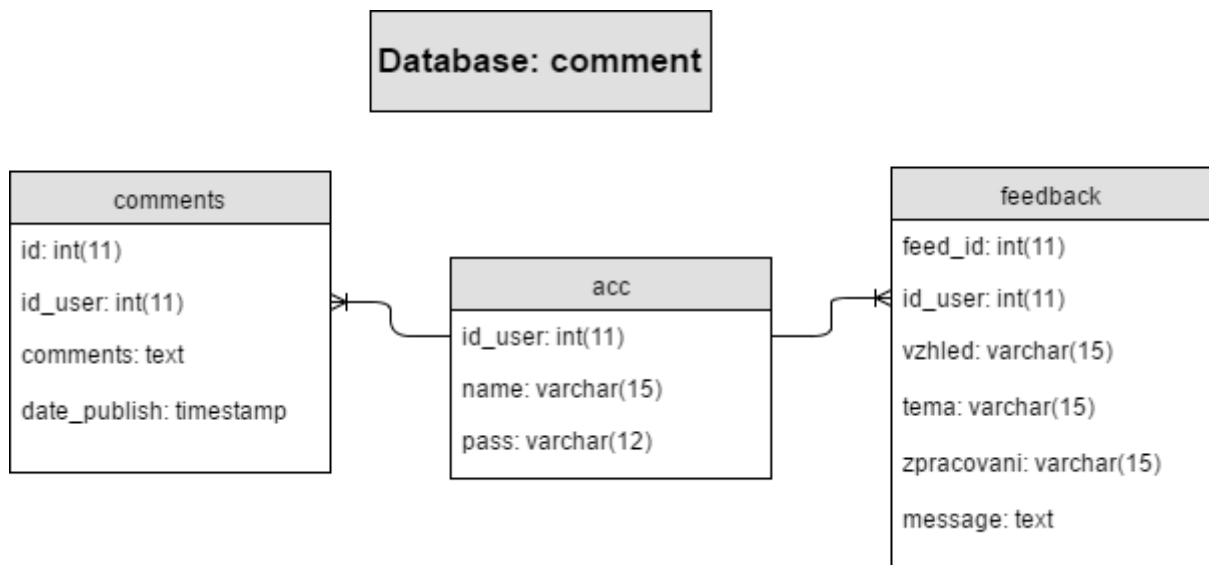
}

mysqli_close($con);

?>
```

Připojí se k databázi a následně vypíše komentáře z databáze. Nejprve vypíše jméno, poté komentář a na posledním řádku datum.

6. Návrh Databáze



```

$result = mysqli_query($con, "SELECT * FROM acc ORDER BY
id_user ASC");

while($row=mysqli_fetch_array($result)){

if (($name==$row['name']) and ($pwd==$row['pass'])){

    $g=1;

    $id=$row['id_user'];  } }

if ($g==1) {

    mysqli_query($con, "INSERT INTO comments(id_user,
comments) VALUES('$id','$comments')");
  
```

Tato část kódu slouží k získání uživatelů z databáze a porovnávání přihlašovacích údajů. Pokud je nalezena shoda, tak je proměnné g přiřazena hodnota 1 a do proměnné id se uloží id uživatele, který přidává obsah. V podmínce, kde se porovnává g, se vkládá obsah komentáře do databáze, jako pojítka s uživatelem, který ho tam vložil slouží id_user.

```
$result = mysqli_query($con, "SELECT * FROM comments INNER  
JOIN acc ON comments.id_user = acc.id_user ORDER BY id ASC");  
while($row=mysqli_fetch_array($result)){  
    echo "<div class='comments_content'>";  
    echo "<h1>" . $row['name'] . "</h1>";  
    echo "<p>" . $row['comments'] . "</p><br>";  
    echo "<h3>" . $row['date_publish'] . "</h3>";  
    echo "</div>";  
  
} } else { echo "Neplatné jméno nebo heslo";}  
mysqli_close($con); //ukonci spojeni s databazi  
?>
```

Tato část kódu se stará o vypisování obsahu z databáze. Pomocí inner join se spojí dvě tabulky tam, kde je shodný id_user a zařídí se tak, aby při vypsání byl komentáři přiřazen správný autor.

7. Generátor

Aplikace generuje dvě postavy, první uloží do proměnných a zároveň do sessionu, druhá postava se při druhém zadávání formuláře uloží pouze do proměnných. Při každém odeslání formuláře proběhne boj. Nejprve se porovná hodnota štěstí, jako první začíná postava s větším štěstím, pokud jsou hodnoty štěstí stejné, tak o pořadí rozhodne náhodně skript pomocí funkce random. Poté započne cyklus while, který probíhá do té doby, dokud má proměnná \$konec hodnotu 0. V cyklu se začnou vypisovat kola, jako první útočí ten s vyšším štěstím, obránci se od bodů života odečte poškození útočníka, pokud obránce přežije první kolo, tak útočí on. Jakmile životy jedné z postav klesnou přinejmenším na nulu, cyklus se ukončí a vypíše se výsledek.

8. Použité nástroje:

PsPad

Editor; pro psaní kódu.

Malování

Grafický editor; pro uložení screenu obrazovky.

Gimp

Grafický editor; pro úpravu obrázků.

Google Chrome

Prohlížeč; pro prohlížení vzhledu stránek.

W3C online validátor

Validační nástroje; ověření validity HTML a CSS.

Endora

Hosting; nahrání a online prezentování vytvořených stránek.

Síla SEO

Test SEO; Kontrola, zda web má potřebné meta tagy, zda je optimalizovaný pro tisk atd.

Draw.io

Online nástroj pro tvorbu diagramů; Pro vytvoření schéma layoutu.

Color Combos

Online nástroj pro generování barev; Identifikace hlavních barev z webu.

Coolors

Online nástroj pro generování barev; Vytvoření barevného schéma.

EasyPHP

Vývojové prostředí lokálního serveru; vývoj a testování php skriptů a databází.

9. Závěr

Po upřesnění mých představ o vzhledu stránek byl můj první problém s obrázky ve funkci „float“, obrázek například vyskočil mimo článek, nebo jej neobtékal text, tento problém byl však vyřešen změnou obrázku. Po vyřešení tohoto problému jsem se zaměřil na chyby v HTML kódu, často způsobenými neukončenými tagy, nevěnoval jsem však pozornost responzivitě webu.

Responzivitě webu jsem opravil až při druhém odevzdávání webu, a současně jsem trochu změnil vzhled. Ze souboru index.php jsem vyjmul obsah a vložil jsem ho do souboru sauron.php, v indexu však zůstal layout a na původní místo obsahu jsem dal PHP skript, který vypíše obsah zvoleného souboru.

Dále jsem vytvořil PHP skript, který odešle informace z formuláře na e-mail prostřednictvím dalšího e-mailu. Tento skript jsem sice stáhnul z internetu, ale i přesto jsem si ho musel upravit podle sebe, aby fungoval správně.

Vytvořil jsem generátor, který vytvoří dvě postavy, ty se však musí vytvářet až po sobě, jedna postava se proto musí uložit do sessionu. Po vytvoření dvou postav nastane cyklus, kde se odečítají životy obránce od poškození útočníka, každé kolo je okomentované a vítěz je vypsán.

Pro přidávání komentářového obsahu a odesílání feedbacku je třeba se zaregistrovat. Registrovaný uživatel má možnost přidávat komentáře, ty se ukládají do databáze a tabulka komentářů je propojena s tabulkou uživatelů skrze společné id_user a inner join. Feedback se stále odesílá na emailovou adresu, ale také se nyní ukládá i do databáze pro případ znepřístupnění adresy.

10. Zdroje:

Sauron. In: *The Lord of the Rings Wiki: Fandom powered by Wikia* [online]. [cit. 28.11.2016]. Dostupné z: <http://lotr.wikia.com/wiki/Sauron>

Valar. In: *The Lord of the Rings Wiki: Fandom powered by Wikia* [online]. [cit. 28.11.2016]. Dostupné z: <http://lotr.wikia.com/wiki/Valar>

Maiar. In: *The Lord of the Rings Wiki: Fandom powered by Wikia* [online]. [cit. 28.11.2016]. Dostupné z: <http://lotr.wikia.com/wiki/Maiar>

Rings of Power. In: *The Lord of the Rings Wiki: Fandom powered by Wikia* [online]. [cit. 28.11.2016]. Dostupné z: http://lotr.wikia.com/wiki/Rings_of_Power

Markle, B (2016) PHPMailer (Verze 5.2.0) [Source code]. Dostupné z: <http://www.inmotionhosting.com/support/email/send-email-from-a-page/using-phpmailer-to-send-mail-through-php>

Parbuaya, G (2014) Chat box (Verze 1.0.0) [Source code]. Dostupné z: <http://www.sourcecodester.com/php/7191/how-create-comment-box-using-php-mysql.html>