

Criterion C: Development

This program was built through the use of PHP, HTML, CSS, and JavaScript. The following document will outline the list of techniques used to build the product including the system templates, the general files of the product, the library and external services used, and the operating files of the product.

Word Count: 1400

Candidate Code: gqs123

TABLE OF CONTENTS

| | |
|---|------------------|
| List of Techniques Used | 4 |
| Pre-Installed Libraries (CSS, Fonts, jQuery Libraries) | 4 |
| Font | 4 |
| CSS Files | 4 |
| jQuery Libraries | 5 |
| Voice RSS Library | 5 |
| System Templates | 6 |
| inc/header.php | 6 |
| inc/footer.php | 8 |
| inc/footer_scripts | 8 |
| inc/cnx.php | 9 |
| General files of the application | 10 |
| index.php | 10 |
| register.php | 13 |
| reset.php | 17 |
| mail.php | 18 |
| <u>After login</u> | <u>19</u> |
| <i>home.php</i> | 19 |
| <u>Games Files</u> | <u>23</u> |
| complete.php | 24 |
| pair.php | 24 |
| listen.php | 25 |
| translate.php | 25 |
| Operating files of the application | 26 |
| services/check.php | 26 |
| services/loginuser.php | 27 |
| services/logout.php | 28 |
| services/recover.php | 28 |
| services/adduser.php | 29 |
| <u>Services for the games</u> | <u>31</u> |
| services/checkanswer.php → Writing Game | 31 |
| services/checkanswer2.php → Picture Game | 31 |
| services/checkanswer3.php → Complete the Sentence Game | 31 |
| services/checkanswer4.php → Complete the Pair Game | 31 |
| services/checkanswer5.php → Listen Game | 31 |
| services/checkanswer6.php → Translation Game | 31 |

| | |
|---------------------------------|------------------|
| <u>Admin Panel Files</u> | <u>34</u> |
| Explanation of techniques used | 34 |
| Screenshots of display | 34 |
| Bibliography | 37 |

List of Techniques Used

- MySQL Databases
- MySQLi Connect
- Bootstrap Stylesheet Library
- Hyperlinks
- Sessions Variables Technology
- Injections Check Functions (MySQL Real Escape String)
- PHP Mail Function
- PHP Conditions (if then)
- PHP Require Statements
- CURL Methods
- PHP Loops
- Form POST method
- Ajax POST method
- PHP MySQL CRUD Application
- HTML 5 DIV & SPAN elements
- PHP MySQL SELECT Query
- PHP Echo and Print Statements
- PHP isset() Function

***See Appendix C for a short description and definition for the techniques used above*

Pre-Installed Libraries (CSS, Fonts, jQuery Libraries)

Font

Font-awesome.min.css (Awesome fonts)¹

```
<!--Score, back, logout, facebook-->
<link rel="stylesheet" href="css/font-awesome.min.css">
```

CSS Files

Sunset.css + Sunset folder (SUNSET CSS KIT)²

```
<!--sunset ui kit & Welcome Ribbon-->
<link href="css/sunset.css" media="screen" rel="stylesheet">
```

Owl.carousel.css (Owl carousel script)³

```
<!--Login/Register Slider-->
<link rel="stylesheet" href="css/owl.carousel.css">
```

Bootstrap.css (Bootstrap 4 CSS framework)⁴

```
<!-- Adjustments in sizes for all platforms (mobile, computer...)-->
<link rel="stylesheet" href="css/bootstrap.css">
```

Flag-icon.css (Flags icons)⁵

```
<!-- Flags icons -->
<link href="css/flag-icon.css" rel="stylesheet">
```

¹ FontAwesome, Robmadole. "FontAwesome/Font-Awesome." GitHub, July 24, 2019.

<https://github.com/FontAwesome/Font-Awesome>.

² "Arctic Sunset UI Kit." PixelKit Premium UI Design Resources. Themefuse Family. n.d.

<http://pixelkit.com/kits/clean-ui-kit>.

³ Deutsch, David. "Owl Carousel 2." Home | Owl Carousel | 2.3.4. Accessed August 28, 2019.

<https://owlcarousel2.github.io/OwlCarousel2/>.

⁴ Otto, Mark, and Jacob Thornton. "Bootstrap." · The most popular HTML, CSS, and JS library in the world. n.d.

<https://getbootstrap.com/>.

⁵ Lipis. "Lipis/Flag-Icon-Css." GitHub, June 14, 2019. <https://github.com/lipis/flag-icon-css>.

jQuery Libraries

Sunset folder :sunset ui kit ⁶

```
<!-- ui kits js !-->
<script src="js/sunset/general.js"></script>
```

vendor/jquery : jquery framework ⁷

Vendor/bootstrap.min.js : bootstrap framework ⁸

```
<script src="js/vendor/bootstrap.min.js"></script>
```

Pgwslder.min.js : jquery slider ⁹

```
<script src="js/pgwslder.min.js"></script>
```

Waypoints.min.js : jquery waypoints ¹⁰

```
<script src="js/waypoints.min.js"></script>
```

js/progressbar.js ¹¹

```
<script src="js/waypoints.min.js"></script>
```

Voice RSS Library

lib/voicerss_tts.php

This service converts text into an audio format with a \$5 fee. It uses *RESTful API* programming interface.¹²

The call of the API is very simple:

```
<?php
require_once('lib/voicerss_tts.php'); => calling the file from the library

$tts = new VoiceRSS;
$voice = $tts->speech([
    # 'key' => '<API key>', => ## API Key for the recorded audio
    # 'hl' => 'en-us', => ## Language Output
    # 'src' => 'Hello, world!', => ## Text to speech function
    # 'c' => 'mp3' ## File Format
]);
?>
```

⁶ “Arctic Sunset UI Kit.” PixelKit Premium UI Design Resources. Themefuse Family. n.d. <http://pixelkit.com/kits/clean-ui-kit>.

⁷ JS Foundation. “jQuery - Write Less Do More.” jQuery. The jQuery Foundation. n.d. <https://jquery.com/>.

⁸ Otto, Mark, and Jacob Thornton. “Bootstrap.” · The most popular HTML, CSS, and JS library in the world. n.d. <https://getbootstrap.com/>.

⁹ Piat, Jonathan M. “PgWSlider - Responsive Slider for JQuery / Zepto.” PgwJS, 2014. <https://pgwjs.com/pgwslider/>.

¹⁰ Troughton, Caleb. “jQuery Waypoints.” Waypoints. Github, November 2013. <https://github.com/imakewebthings/waypoints>.

¹¹ Brunfeldt, Kimmo. “PROGRESS BAR.JS.” ProgressBar.js - Progress bars with JavaScript, May 28, 2019. <https://kimmobrunfeldt.github.io/progressbar.js/>.

¹² “Voice RRS.” Voice RSS - Free Text-to-speech (TTS) online service, 2018. <http://www.voicerss.org/default.aspx>.

System Templates

The web-app quicklang.net uses system templates, dividing each page into 4 different sections:

inc/header.php

QuickLang

The inc/header.php file displays the logo of the product, but also contains all the CSS scripts for all pages in the website.

```
<!DOCTYPE html>
<html lang="fr" class="no-js">
<head>
<!-- This file is for the design of the header (black logo) and -->

    <!-- Mobile Specific Meta -->
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Favicon-->
    <link rel="shortcut icon" href="img/icon.png">
    <!-- Author Meta -->
    <meta name="author" content="Colorlib">
    <!-- Meta Description -->
    <meta name="description" content="">
    <!-- Meta Keyword -->
    <meta name="keywords" content="">
    <!-- meta character set -->
    <meta charset="UTF-8">
    <!-- Site Title -->
    <title>Quicklang © - Learn English with Simplicity</title>

    <link href="https://fonts.googleapis.com/css?family=Poppins:100,300,500"
rel="stylesheet">

    <!-- CSS ===== -->
    <link rel="stylesheet" href="css/linearicons.css"> <!-- Game logos CSS HOMEPAGE-->
    <link rel="stylesheet" href="css/owl.carousel.css"> <!-- Login/Register Slider-->
    <link rel="stylesheet" href="css/font-awesome.min.css"> <!-- Score, back, logout,
facebook-->
    <link rel="stylesheet" href="css/nice-select.css"> <!-- Multiple Choice in
Listen&Write and Pics&Words-->
    <link rel="stylesheet" href="css/magnific-popup.css"><!-- Congratulations end of
game score-->
    <link rel="stylesheet" href="css/bootstrap.css"> <!-- Adjustments in sizes for all
platforms (mobile, computer...)-->

    <link rel="stylesheet" href="css/main.css"> <!-- Fonts, page structure-->
    <link href="css/flag-icon.css" rel="stylesheet"><!-- Flags-->
    <link href="css/login.css" rel="stylesheet"><!-- Login/Register Style/Design-->
<!-- sunset ui kit !-->
    <?php if ( (($_SERVER['PHP_SELF']) != '/index.php') && (($_SERVER['PHP_SELF']) !=
'/register.php') ) { ?>
    <link href="css/sunset.css" media="screen" rel="stylesheet"> <!-- Welcome Ribbon-->
    <?php } ?>

</head>
<body>
    <!-- <div class="oz-body-wrap"> -->
    <!-- Start Header Area -->
    <div class="header">
```

Furthermore, this header has additional functionalities — the logo **QuickLang** can act as a button resulting in two different outputs through the use of PHP Conditions.

```
<?php

        ### PHP Condition ###

        ## ##### ##

    If the user is in a game session (writing/listen/pair/translate/complete/pics)
    the logo link will redirect to the games menu page, else the logo link will
    redirect to the global main page (first display), hence logging out the user.

        ## ##### ##

if ( (($_SERVER['PHP_SELF']) == '/writing.php') ||      (($_SERVER['PHP_SELF']) ==
'/listen.php') ||      (($_SERVER['PHP_SELF']) == '/pics.php') ||
(($_SERVER['PHP_SELF']) == '/pair.php') ||      (($_SERVER['PHP_SELF']) ==
'/translate.php') || (($_SERVER['PHP_SELF']) == '/complete.php') ) {
    $logo_link='start';
} else {
    $logo_link='https://www.quicklang.net';
}

?>

        ### Displays the QuickLang Logo Image ###

<a href="<?php echo $logo_link; ?>"></a>
    <div class="header-right">

</div>
</div>

    <!-- End Header Area -->
```

inc/footer.php



Copyright©2020
f

This file serves as the bottom footer for all pages in the website. This file includes the logo with a link to my client's business homepage and Facebook page. These functionalities were achieved by using the HyperLink (HREF) attribute and the tag `<a>` between the image and my client's URLs.

```
<!-- Start Footer Area -->
<footer class="section-gap">
<div class="container">
    <div class="footer-bottom d-flex justify-content-between align-items-center flex-wrap">
        <a href="https://www.fondationababou.ma/" target="_blank">
            </a>

        <!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->
        <p class="footer-text m-0"> Copyright<copy><script>document.write(new
Date().getFullYear());</script> <br/><i class="fa fa-facebook" aria-hidden="true"></i>
        <a href="https://www.facebook.com/FondationAbabou/" target="_blank">Fondation
        Ababou</a></p>
    </a></p>
    <!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->

    </div>
    </div>
</footer>
<!-- End Footer Area -->
</div>
<?php include("footer_scripts.php"); ?>
```

inc/footer_scripts

```
#####
This file is an extension to the previous file (inc/footer.php), containing all
of the JavaScripts code called from the game. Through the use of this file, other
algorithms can make a jQuery 3.3.1 library call. jQuery is a javascript framework
designed to simplify html and events calls and handling.
#####

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js"
integrity="sha384-b/U6ypiBEHpOf/4+1nzFpr53nxSS+GLCKfwbDfNTxtclqgenISfwAzpKaMNFNmj4"
crossorigin="anonymous"></script>

<script src="js/vendor/bootstrap.min.js"></script>
<script src="js/jquery.ajaxchimp.min.js"></script>
<script src="js/owl.carousel.min.js"></script>
<script src="js/jquery.nice-select.min.js"></script>
<script src="js/jquery.magnific-popup.min.js"></script>
<script src="js/jquery.counterup.min.js"></script>
<script src="js/waypoints.min.js"></script>
<script src="js/main.js"></script>
<script src="js/login.js"></script>
```



```
<!-- ui kits js !-->
<script src="js/sunset/general.js"></script>
```

inc/cnx.php

This file includes the PHP code to connect to the MySQL database server. I was able to establish a MySQL database connection by using the MySQLi connect technique. In the code below, a new connection is established with the admin credentials given at *\$login* and *\$pass*.

```
<?php
$host="localhost";
$login="quicklan_user";
$pass="taha/123@@";
$bdd="quicklan_database";

$cnx=mysqli_connect($host, $login, $pass);
mysqli_select_db($cnx,$bdd);

    ### Display error for showing error line and debugging in case of errors on code ###
    ###

mysqli_query($cnx,"SET NAMES 'utf8'");

                                ###
    The 1 tells the program to output the following message: 'Please enter a valid email'.
    This can be seen in the following file: js/jquery.ajaxchimp.min.js
                                ###

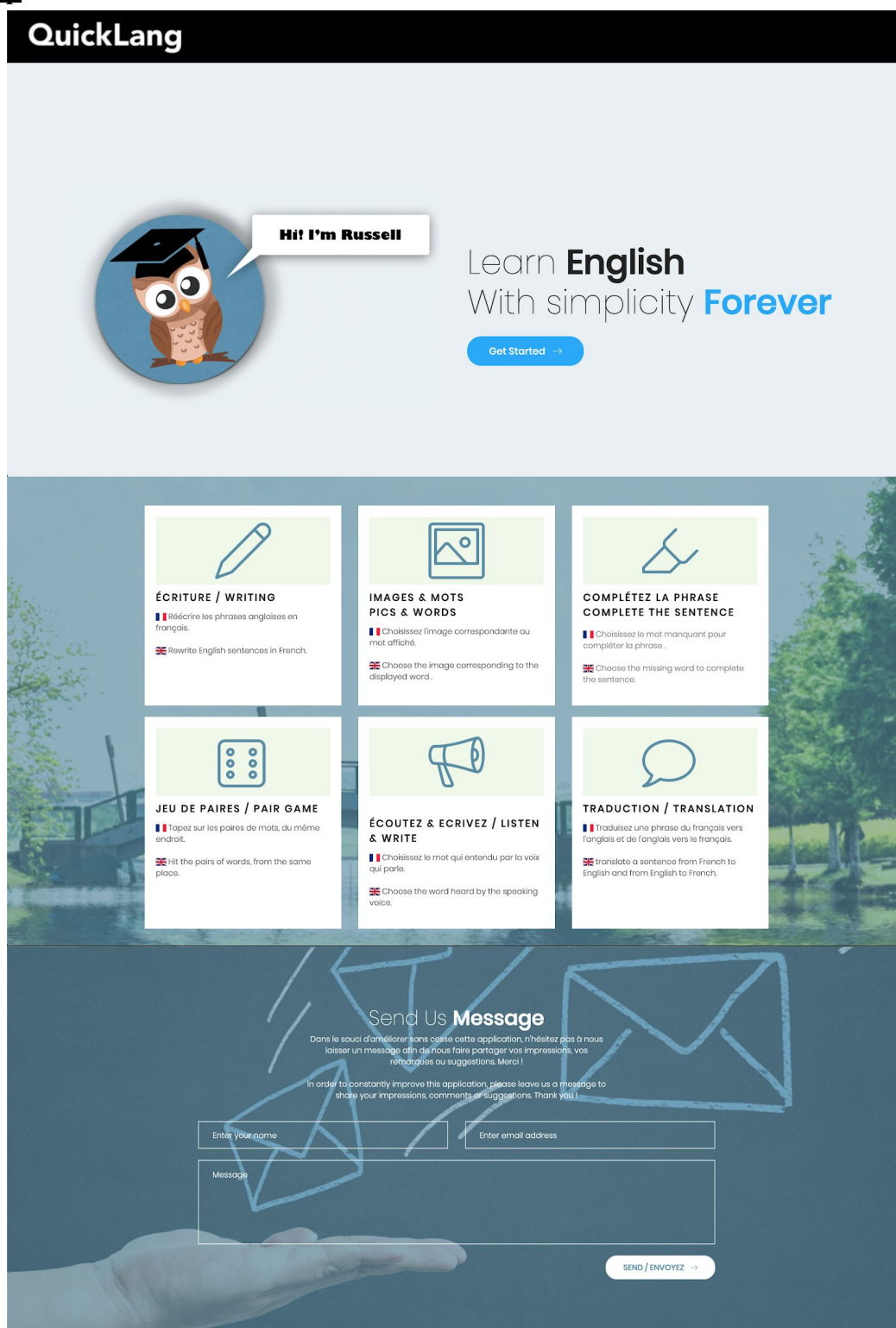
ini_set('display_errors', 1);

?>
```

General files of the application

The web application consists of 11 files that are used as a display to the user. Hence, the reason why they are called the “*General files of the application*”.

index.php



index.php is the homepage and the first display to the user. In the first line of code, the file *inc/header.php* is called through the PHP include() function.

```
<!-- Includes the header.php file (Black Header with Title of Page) -->
<?php include("inc/header.php"); ?>

<?php

#####

The following code was created using HTML 5 DIV tags and SPAN elements, including
External CSS. I used the SPAN element to make the text bold and more appealing to
the user.

#####

#### BODY CODE FOR THE MAIN PAGE - INCLUDES RUSSELL, SLOGAN, AND GET STARTED ####
?>

<!-- Start Banner Area -->
<section class="banner-area relative">
    <div class="container">
        <div class="row fullscreen align-items-center justify-content-center">
            <div class="banner-left col-lg-6">
                
            <!-- Owl Image -->
            </div>
            <div class="col-lg-6">
                <div class="story-content">
                    <h1>Learn <span class="sp-1">English</span><br>
                    With simplicity <span class="sp-2">Forever</span></h1>
                    <a href="register" class="genric-btn primary circle arrow">Get
Started<span class="lnr lnr-arrow-right"></span></a>
                </div>
            </div>
        </div>
    </div>
</section>
```

```
<!--                                Start Feature Area

I used basic Bootstrap and HTML to display the description for each game as
seen in the screenshots above. The following code was repeated with minor
alterations to change the position of the description boxes.

-->

<?php #### ÉCRITURE/WRITING CODE - POSITIONING, EFFECT, DESCRIPTIONS#### ?>
<section class="feature-area pt-100 pb-100 relative">
    <div class="overlay overlay-bg"></div>
    <div class="container">
        <div class="row">
            <div class="col-lg-4 col-sm-6 d-flex align-items-stretch">
                <div class="single-feature">
                    <div class="icon">
                        <span class="lnr lnr-pencil"></span>
                    </div>
                    <div class="desc">
                        <h2 class="text-uppercase">Écriture / Writing</h2>
                        <p>
                            <span class="flag-icon flag-icon-fr"></span> Réécrire les
phrases anglaises en français.<br>
                            <span class="flag-icon flag-icon-gb"></span> Rewrite
English sentences in French.
                        </p>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

```

        </div>

        [...] <!-- Code above repeated for the other game descriptions -->

    </section>

```

The code below was used to create the feedback form at the bottom of the page.

```

<!-- Start Contact Form Area -->
    <a name="contact"></a>
    <section class="contact-area pt-100 pb-100 relative">
        <div class="overlay overlay-bg"></div>
        <div class="container">
            <div class="row justify-content-center text-center">
                <div class="single-contact col-lg-6 col-md-8">
                    <h2 class="text-white">Send Us <span>Message</span></h2>
                    <p class="text-white">
                        Dans le souci d'améliorer sans cesse cette application,
                        n'hésitez pas à nous laisser un message afin de nous faire partager vos impressions, vos
                        remarques ou suggestions. Merci !
                    <br/><br/>
                        In order to constantly improve this application, please leave
                        us a message to share your impressions, comments or suggestions. Thank you !
                    </p>
                </div>
            </div>
        </div>
    </section>
<!-- End Contact Form Area -->

```

The code below tells the program to initiate the PHP script of mail.php file once the user has clicked the *Send* button. The user's information (name, email address, and message) is saved and sent to the mail.php file by using the PHP POST method.

```

<form id="myForm" action="mail.php" method="post" class="contact-form">

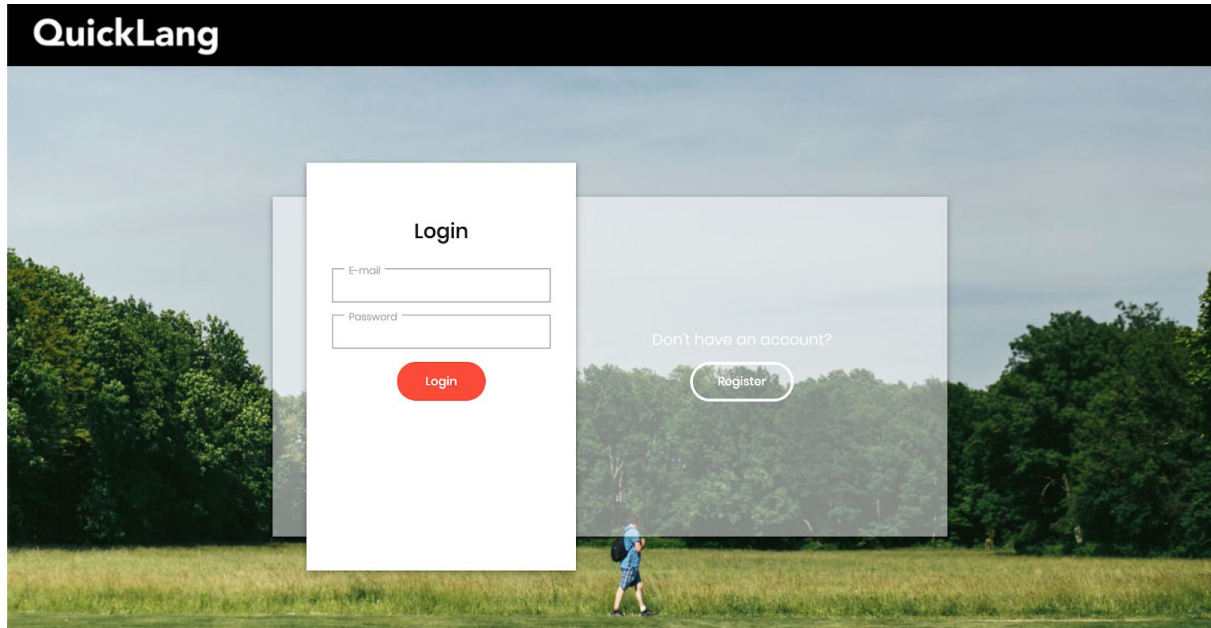
<!-- Once the user inputs all the necessary data below, send the data to mail.php
to proceed action -->

    <div class="row justify-content-center">
        <div class="col-lg-5">
            <input name="fname" placeholder="Enter your name"
onfocus="this.placeholder = '' onblur="this.placeholder = 'Enter your name'"
class="common-input mt-20" required="" type="text">
        </div>
        <div class="col-lg-5">
            <input name="email" placeholder="Enter email address"
pattern="[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{1,63}$" onfocus="this.placeholder =
'' onblur="this.placeholder = 'Enter email address'" class="common-input mt-20"
required="" type="email">
        </div>
        <div class="col-lg-10">
            <textarea class="common-textarea mt-20" name="message"
placeholder="Message" onfocus="this.placeholder = '' onblur="this.placeholder =
'Message'" required=""></textarea>
        </div>
        <div class="col-lg-10 d-flex justify-content-end">
            <button class="primary-btn white-bg d-inline-flex
align-items-center mt-20"><span class="mr-10">Send / Envoyez </span><span class="lnr
lnr-arrow-right"></span></button> <br>
        </div>
        <div class="alert-msg"></div>
    </div>
</form>
</div>
</section>
<!-- End Contact Form Area -->

```

```
<?php include("inc/footer.php"); ##Bottom part of the page?>
<?php include("inc/footer_close.php"); ?>
```

register.php



Copyright©2020
fTaha Ababou

Login

E-mail

Pa

Please fill in this field.

Login

Login

E-mail

gqs123@g.com

Password

Please fill in this field.

By using the PHP POST form method, the program collects the user's data from the client side and sends it to the server side to authenticate the credentials inputted.

```
<?php include("inc/header.php");
session_start();

#####

    This file is divided into two sections: the authentication form and the
    registration form. To access the website's content, the existing user will have
    to go through the authentication form.
    #####

?>

<!-- Start Banner Area -->
<section class="login-area relative">
    <div class="container">
        <div class="row fullscreen align-items-center justify-content-center">
            <div class="col-lg-12 signform">

<div class="veen">
    <div class="login-btn splits">
        <p>Already a user?</p>
        <?php
            ##### initialising variables #####
            $value='';
            $login='login';
            $forgot='';
            $message='';
        <div class="wrapper">
            <form id="login" tabindex="500" action="login" method="POST">
                <h3>Login</h3>
                <?php echo $message; ?>
                <div class="mail">
                    <input type="mail" name="email" <?php echo $value; ?>
<pattern="[a-z0-9._%+-]+[A-Z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$" required>
                    <label>E-mail</label>
                </div>
                <div class="passwd">
                    <input type="password" name="password" pattern=".{6,}"
title="Please enter at least 6 characters" required>
                    <label>Password</label>
                </div>
                <?php echo $forgot; ?>
                <div class="submit">
                    <button class="dark">Login</button>
                </div>
            </form>
```

Below are the actions that the program has to take in the occurrence of an error. The technique used for the following piece of code are PHP Conditional statements and hyperlink attributes.

```
if (isset($_SESSION['error'])) {
    $error=$_SESSION['error'];
    ##### test errors number if error 1 or if error 2 #####
    switch ($error) {
        #####
        Case 1 tells the program that if a new user tries to register a new account with
        an email address that is already registered to an existing user, a message will
        appear with the following output: Email already registered.
        #####

        case 1:
            $message='<i class="error">Email already registered.</i></br></br>';
            $email_error=$_SESSION['error_email'];
```



```
$value='value="'. $email_error. "'';  
$forgot='';  
break;
```

```
#####
```

Case 2 tells the program that if a user attempts to login and inputs the wrong password, a button will appear in the form, allowing the user to reset their password. The button will then redirect the user to the file reset.php.

```
#####
```

```
case 2:  
$message='<i class="error">Wrong password</i></br></br>';  
$forgot='<u class="reset"><a href="reset">Recover  
password</a></u></br></br>';  
break;
```

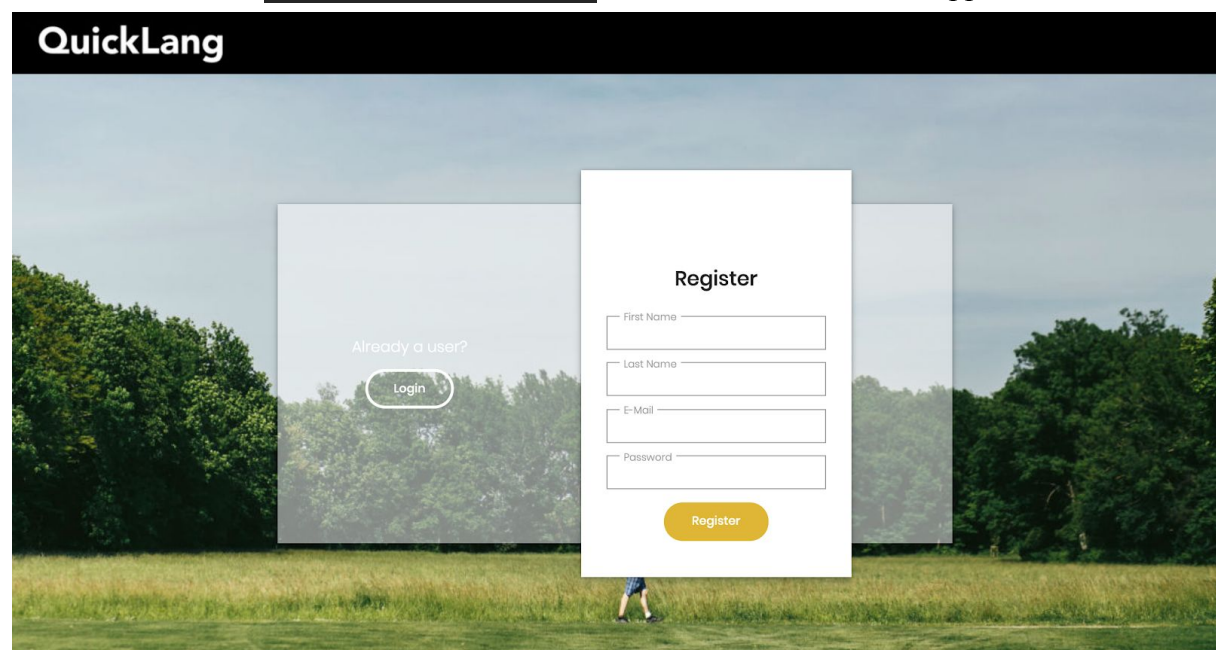
```
#####
```

In Case 3, the program verifies that the user inputted an email address registered in the database. If yes, a message will be displayed confirming that the recovery email has been sent.

```
#####
```

```
case 3:  
$message='<i class="error">The Password recovery email has been sent to  
your on-file email address</i></br></br>';  
$forgot='';  
break;  
}  
##### end errors test #####  
} ?>
```

If the user does not have an account and wishes to register one, they must press the *Register* button telling the program to make a slide translation from the login form to the registration form. The CSS file "`css/owl.carousel.css`" allows this transition to happen.



The code for the registration form is demonstrated in the code below. I provided a registration for interface by using basic HTML DIV tags, assigning each text field as a variable.

```
<!-- REGISTRATION FORM -->
    <button class="active">Login</button>
</div>
<div class="rgstr-btn splits">
    <p>Don't have an account?</p>
    <button>Register</button>
</div>
```

The information inputted by the user is stored in the client server; it is sent to the server side when the user clicks *Register* by using the PHP POST method. The action *adduser* tells the program to send the data from the form to the file: ***services/adduser.php***.

```
<form id="register" action="adduser" tabindex="502" method="POST">
```

The program will proceed the code only if the user inputted the fields under the correct format as seen below. This functionality was achieved through the use of PHP `require()` statements.

```
<h3>Register</h3>
    <div class="name">
        <input type="text" name="firstname"
pattern="[a-zA-Z][a-zA-Z0-9\s]*" required>
        <label>First Name</label>
    </div>
    <div class="name">
        <input type="text" name="lastname"
pattern="[a-zA-Z][a-zA-Z0-9\s]*" required>
        <label>Last Name</label>
    </div>
    <div class="mail">
        <input type="mail" name="email"
pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$.+[A-Z]{2,}$. " required>
        <label>E-Mail</label>
    </div>

    <div class="passwd">
        <input type="password" name="password" title="Please enter at least
6 characters" pattern=".{6,}" required>

</a>

        <label>Password</label>
    </div>
    <div class="submit">
        <button class="dark">Register</button>
    </div>
</form>
</div>
</div>
</div>
</div>
</section>

<!-- Link to the files inc/footer.php and inc/footer_close.php - These files
include the JavaScript code used for all webpages and also displays the bottom
footer where my client's logo and a link to their Facebook page -->

<?php include("inc/footer.php"); ?>
<?php include("inc/footer_close.php"); ?>
```


reset.php

Login

Wrong password

[Recover password](#)

Login

Recover password

Enter the email associated with your account in the text box and then click the "Recover" button.

Recover

Recover password

The email address you entered is not associated with a QuickLang account. Please try again.

Recover

The purpose of this file is to allow an existing user to restore their password in case they cannot remember or simply wish to change it to a new one. This file is a continuation to the *error case 2*.

Technique Used:

- The code is using a PHP Conditional Statements and a MySQL request for checking if the email address inputted exists on the database.

```
<?php include("inc/header.php");
session_start();
?>

<!-- Start Banner Area -->
<section class="login-area relative">
  <div class="container">
    <div class="row fullscreen align-items-center justify-content-center">
      <div class="col-lg-12 signform">

        <div class="veen">

          <div class="wrapper">
            <form id="login" tabindex="500" action="recover" method="POST">
              <h3>Recover password</h3>
              <?php

                ##### test error if email non existent in database show message #####
```

PHP isset() function was used to declare the variable to the database. If the variable is null, the program will output an error message.

```
        if (isset($_SESSION['error'])) {
          $error=$_SESSION['error'];
          if ($error==1) {
            $message='The email address you entered is not associated with a
Quicklang account';
          } else {
            $message='<span class="error">Enter the email associated with your
account in the text box and then click the "Recover" button.</span><br/><br/>';
          }
        }
      </div>
    </div>
  </div>
</section>
?>
```

```
<?php include("inc/footer.php"); ?>
<?php include("inc/footer close.php"); ?>
```



This snippet of code sends an email with the data entered in contact form to a dedicated email address (info@quicklang.net). It uses a simple PHP mail function — a SMTP server is already installed on the hosting server.

```
<?php include("inc/header.php"); ?> <!-- Includes the header.php file (Black Header with Title of Page) -->

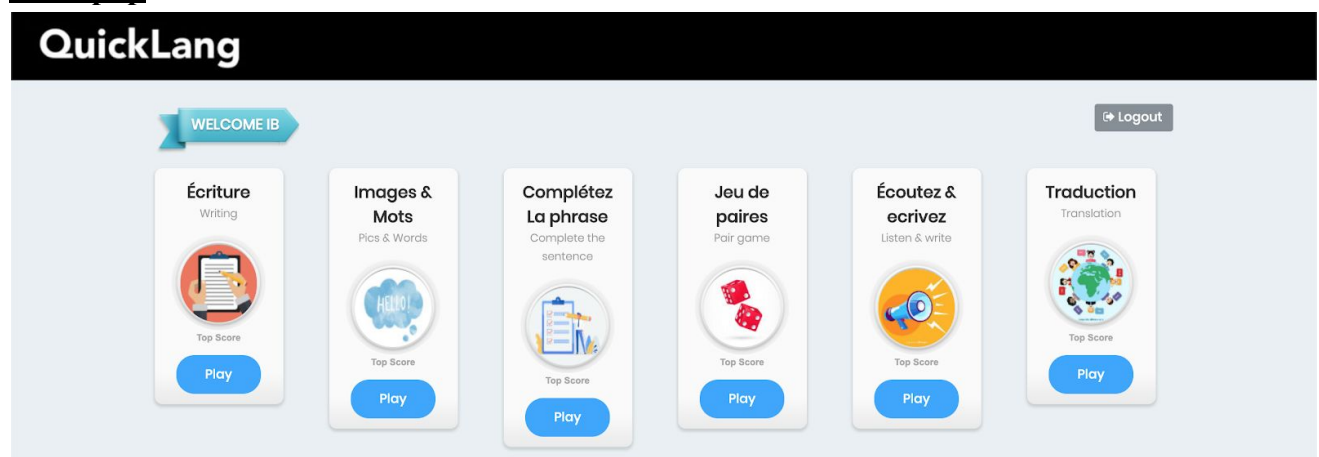
<?php
    $to = 'info@quicklang.net';
    $firstname = $_POST["fname"];
    $email= $_POST["email"];
    $text= $_POST["message"];
    $headers = 'MIME-Version: 1.0' . "\r\n";
    $headers .= "From: " . $email . "\r\n"; // Sender's E-mail
    $headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
    $message = '<table style="width:100%">
        <tr>
            <td>' . $firstname . '</td>
        </tr>
        <tr><td>Email: ' . $email . '</td></tr>
        <tr><td>Email: ' . $text . '</td></tr>
    </table>';

    <!-- PHP POST Method was used to send the user's input (name, email, and their message) to the PHP Mail function -->

    if (@mail($to, $email, $message, $headers))
    {
        echo "<style> body { background-image: url(../img/contact-bg.jpg); } </style>" ?>
        <div class = "container"></div>
        <div class="row justify-content-center text-center"></div>
        <div class="col-lg-3 col-md-6">
            <h2 class="text-white">Your message <span>has been sent!</span></h2>
            <p class="text-white">
        </div>
    }
    else{
        echo 'failed';
    }
?>
```

[After login](#)

[home.php](#)



This file displays the user's home page; this page contains a list of games and a play button for each game. The program acquires the user's data via a MySQL request and stores the data into session variables, which is linked with the file: `"services/check.php"`.

```
<?php

#####

1. This page include session variable with name and best score for logged
   player
2. It uses PHP session technology for getting user data by a MySQL request and
   storing it into session variables (check.php).
3. Second MySQL request is getting the best score of each game based on Player
   ID retrieved previously by MySQL request and stored into the session.
   #####

?>

<?php include("services/check.php"); ?>
<?php include("inc/header.php"); ?>
    <!-- Start Banner Area -->

    <!-- Start Feature Area -->
        <section class="banner-area relative">
            <div class="overlay overlay-bg"></div>
            <div class="container">
                <div class="row">
                    <div class="col-md-4 col-sm-offset-1">
                        <div class="ribbons">
                            <!-- Ribbons -->

                            <?php ##### calling first name variable stored in session after login and displays it
                                in the ribbon ##### ?>
                                <div class="ribbon ribbon-blue"><span>Welcome <?php echo
$_SESSION['firstname']; ?></span></div>
                                <!--/ Ribbons -->
                                </div>
                            </div>

                            <?php ##### logout button ##### ?>
                            <div class="col-md-8 text-right" style="margin-top:25px"><a href="logout"
class="btn btn-secondary btn-sm"><span><i style="color:white" class="fa
fa-sign-out"></i>&nbsp;Logout&nbsp;</span></a></div>

                            </div> <!-- end row div -->

                            <div class="row">
                                <div class="col-md-2">
                                    <!-- Avatar Placeholder -->
                                    <div class="widget-container widget_avatar boxed">
                                        <div class="inner">
                                            <h5>Écriture</h5>
                                            <span class="subtitle">Writing </span>
                                            <div class="avatar"></div>
                                            <div class="followers">
                                                <span>Top Score</span>

                                            <?php ##### showing top_s1 top score variable for game1 from received from
                                                check.php with a sql request (SELECT TOP SCORE from game where id_game=1)
                                                #####?>

                                                <span class="counter"><?php echo $top_s1;?></span>
                                            </div>
                                            <div class="follow">
                                                <a href="writing" class="genric-btn info circle
text-decoration-none"><span>Play</span></a>


```

```

        </div>
    </div>
</div>
<!--/ Avatar Placeholder -->
</div>
<div class="col-md-2">
<!-- Avatar Placeholder -->
<div class="widget-container widget_avatar boxed">
    <div class="inner">
        <h5>Images & Mots</h5>
        <span class="subtitle">Pics & Words</span>
        <div class="avatar"></div>
        <div class="followers">
            <span>Top Score</span>
            <span class="counter"><?php echo $top_s2;?></span>
        </div>
        <div class="follow">
            <a href="pictures" class="genric-btn info circle
text-decoration-none"><span>Play</span></a>
        </div>
    </div>
</div>
<!--/ Avatar Placeholder -->
</div>
<div class="col-md-2">
<!-- Avatar Placeholder -->
<div class="widget-container widget_avatar boxed">
    <div class="inner">
        <h5>Complétez La phrase </h5>
        <span class="subtitle">Complete the sentence</span>
        <div class="avatar"></div>
        <div class="followers">
            <span>Top Score</span>
            <span class="counter"><?php echo $top_s3;?></span>
        </div>
        <div class="follow">
            <a href="complete" class="genric-btn info circle
text-decoration-none"><span>Play</span></a>
        </div>
    </div>
</div>
<!--/ Avatar Placeholder -->
</div>

<div class="col-md-2">
    <!-- Avatar Placeholder -->
    <div class="widget-container widget_avatar boxed">
        <div class="inner">
            <h5>Jeu de paires</h5>
            <span class="subtitle">Pair game</span>
            <div class="avatar"></div>
            <div class="followers">
                <span>Top Score</span>
                <span class="counter"><?php echo $top_s4;?></span>
            </div>
            <div class="follow">
                <a href="pair" class="genric-btn info circle
text-decoration-none"><span>Play</span></a>
            </div>
        </div>
    </div>
    <!--/ Avatar Placeholder -->
    </div>

    <div class="col-md-2">
        <!-- Avatar Placeholder -->
        <div class="widget-container widget_avatar boxed">
            <div class="inner">
                <h5>Écoutez & écrivez</h5>

```

```

        <span class="subtitle">Listen & write</span>
        <div class="avatar"></div>
        <div class="followers">
            <span>Top Score</span>
            <span class="counter"><?php echo $top_s5;?></span>

        </div>
        <div class="follow">
            <a href="listen" class="genric-btn info circle
text-decoration-none"><span>Play</span></a>
        </div>
    </div>
</div>
<!-- Avatar Placeholder -->
</div>

<div class="col-md-2">
    <!-- Avatar Placeholder -->
    <div class="widget-container widget_avatar boxed">
        <div class="inner">
            <h5>Traduction </h5>
            <span class="subtitle">Translation</span>
            <div class="avatar"></div>
            <div class="followers">
                <span>Top Score</span>
                <span class="counter"><?php echo $top_s6;?></span>
            </div>
            <div class="follow">
                <a href="translate" class="genric-btn info circle
text-decoration-none"><span>Play</span></a>
            </div>
        </div>
    </div>
    <!-- Avatar Placeholder -->
</div>
</div>
</section>

<?php include("inc/footer.php"); ?>
<?php include("inc/footer_close.php"); ?>

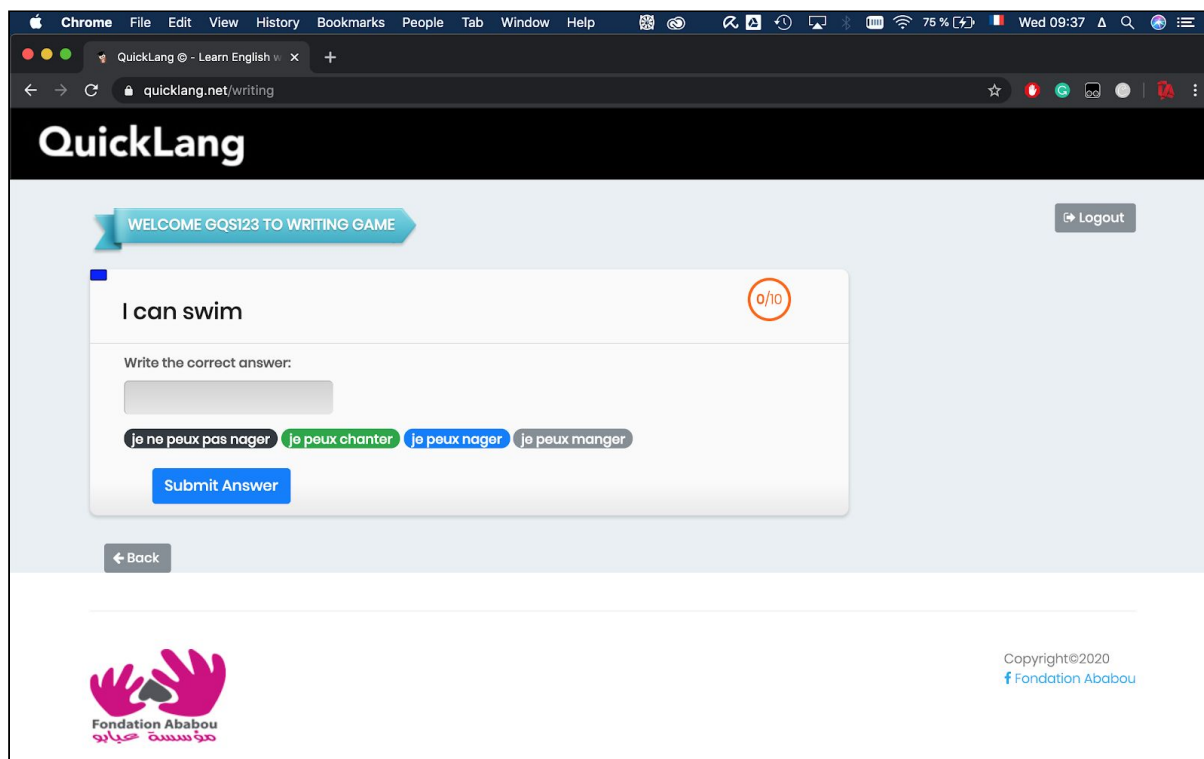
```

Games Files

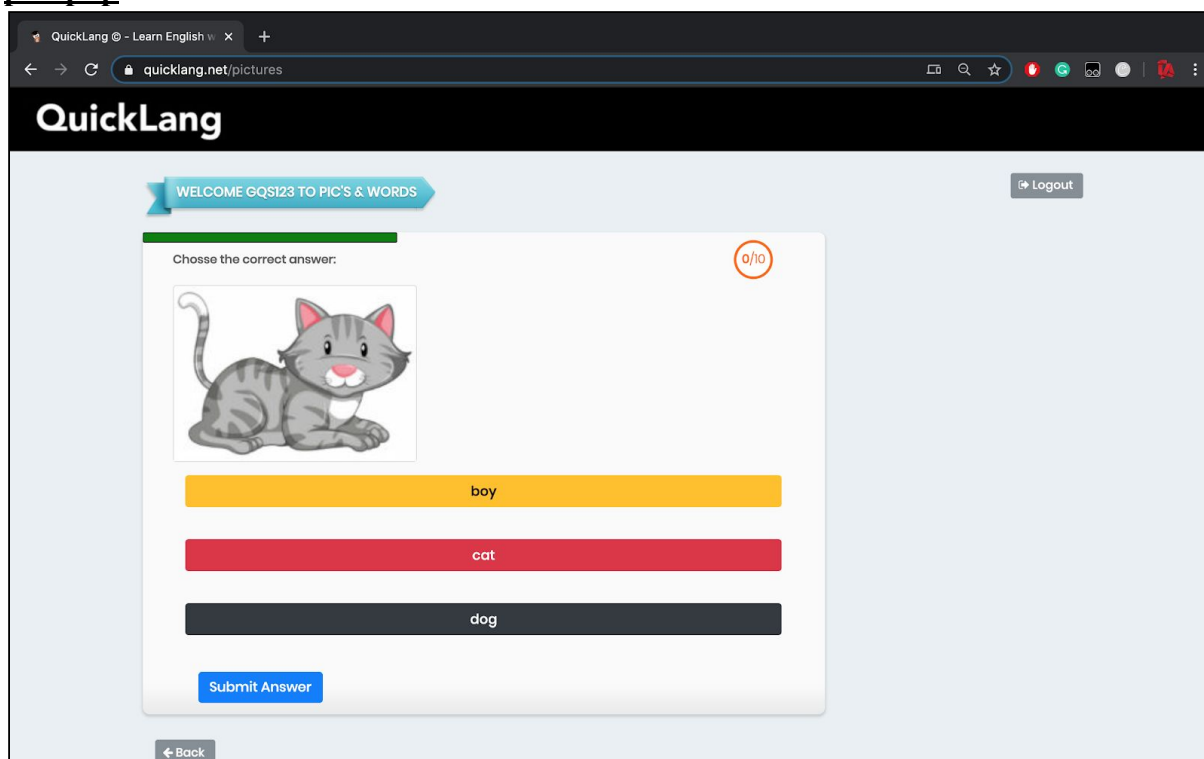
For every game session, 10 questions (with their responses) are chosen at random from the database. This function is enabled by using the PHP MySQL request function.

The operating files for each game are the following:

writing.php



pics.php



complete.php

The screenshot shows a web browser window with the address bar displaying 'quicklang.net/complete'. The page has a dark header with the 'QuickLang' logo. A teal banner at the top reads 'WELCOME GQSI23 TO LISTEN & WRITE GAME'. In the top right corner, there is a 'Logout' button. The main content area features a white box with the text 'I ... working on the computer.' followed by a '0/10' score indicator. Below this, it says 'Choose the missing word:' and lists three options: 'go' (highlighted in green), 'enjoy' (in a dark grey box), and 'run' (in a yellow box). A blue 'Submit Answer' button is at the bottom of the box. A 'Back' button is located below the box. The footer contains a pink logo of two hands holding a heart, the text 'Fondation Ababou', and the copyright notice 'Copyright©2020 Fondation Ababou'.

pair.php

The screenshot shows a web browser window with the address bar displaying 'quicklang.net/pair'. The page has a dark header with the 'QuickLang' logo. A teal banner at the top reads 'WELCOME GQSI23 TO PAIR GAME'. In the top right corner, there is a 'Logout' button. The main content area features a white box with the text 'Choose the pair of related word's' followed by a '0/10' score indicator. Below this, it lists four words, each with a colored square next to it: 'demand' (dark grey), 'see' (teal), 'shoes' (green), and 'supply' (blue). A blue 'Submit Answer' button is at the bottom of the box. A 'Back' button is located below the box. The footer contains a pink logo of two hands holding a heart, the text 'Fondation Ababou', and the copyright notice 'Copyright©2020 Fondation Ababou'.

listen.php

QuickLang @ - Learn English v x +

quicklang.net/listen

QuickLang

WELCOME GQSI23 TO LISTEN & WRITE GAME

Logout

Choose the correct answer: 0/10

0:01 / 0:01

j'aime chanter


j'aime danser

j'aime nager

Submit Answer

Back

Copyright©2020
f Fondation Ababou



translate.php

QuickLang @ - Learn English v x +

quicklang.net/translate

QuickLang

WELCOME GQSI23 TO TRANSLATION GAME

Logout

Il a une voiture 1/10


Write the correct translation:

He has a car He has a bike he buy a car

Submit Answer

Back

Copyright©2020
f Fondation Ababou



Operating files of the application

services/check.php

- This file is present on each game and each action it makes a verification on the database of the login and passwords input through the session variables created before on the file loginuser.php and containing the login and passwords input.
 - This file displays after checking the best score of the player in each game he played via a MySQL request
 - Session variable function
- Non used Cookies (data stored in browser session data is stored on **server side**)

```
<?php
##### This file verifies whether the user has passed through the login form. #####
session_start();
include('inc/cnx.php');

#####defining variable id & email & password #####
$id=$_SESSION['id'];
$email=$_SESSION['email'];
$password=$_SESSION['password'];

#####check if stored variables in session are in database #####
$sql_user="SELECT * from app_users WHERE id='$id' AND email='$email' AND
password='$password'";
$req_user=mysqli_query($cnx,$sql_user);
$total=mysqli_num_rows($req_user);
$affect_user=mysqli_fetch_array($req_user,MYSQLI_ASSOC);

#####if total 0 mean if no record found using this email and password, redirect user
to register page #####
if ($total==0) {
header('location:register');
} else {

##### ELSE get top score user for each game and store it into variable that will be called
in home file #####
$sql_score_game1="SELECT MAX(score) as TOP_GAME1 FROM app_scores where (id_game=1 AND
id_user='".$affect_user['id']."' )";
$sql_score_game2="SELECT MAX(score) as TOP_GAME2 FROM app_scores where (id_game=2 AND
id_user='".$affect_user['id']."' )";
$sql_score_game3="SELECT MAX(score) as TOP_GAME3 FROM app_scores where (id_game=3 AND
id_user='".$affect_user['id']."' )";
$sql_score_game4="SELECT MAX(score) as TOP_GAME4 FROM app_scores where (id_game=4 AND
id_user='".$affect_user['id']."' )";
$sql_score_game5="SELECT MAX(score) as TOP_GAME5 FROM app_scores where (id_game=5 AND
id_user='".$affect_user['id']."' )";
$sql_score_game6="SELECT MAX(score) as TOP_GAME6 FROM app_scores where (id_game=6 AND
id_user='".$affect_user['id']."' )";

$req_score_game1=mysqli_query($cnx,$sql_score_game1);
$req_score_game2=mysqli_query($cnx,$sql_score_game2);
$req_score_game3=mysqli_query($cnx,$sql_score_game3);
$req_score_game4=mysqli_query($cnx,$sql_score_game4);
$req_score_game5=mysqli_query($cnx,$sql_score_game5);
$req_score_game6=mysqli_query($cnx,$sql_score_game6);

$affect_score_game1=mysqli_fetch_array($req_score_game1,MYSQLI_ASSOC);
$top_s1=$affect_score_game1['TOP_GAME1'];

$affect_score_game2=mysqli_fetch_array($req_score_game2,MYSQLI_ASSOC);
$top_s2=$affect_score_game2['TOP_GAME2'];

$affect_score_game3=mysqli_fetch_array($req_score_game3,MYSQLI_ASSOC);
$top_s3=$affect_score_game3['TOP_GAME3'];
```

```

$aaff_score_game4=mysqli_fetch_array($req_score_game4,MYSQLI_ASSOC);
$top_s4=$aaff_score_game4['TOP_GAME4'];

$aaff_score_game5=mysqli_fetch_array($req_score_game5,MYSQLI_ASSOC);
$top_s5=$aaff_score_game5['TOP_GAME5'];

$aaff_score_game6=mysqli_fetch_array($req_score_game6,MYSQLI_ASSOC);
$top_s6=$aaff_score_game6['TOP_GAME6'];

##### store data in sessions to keep user logged in #####
$_SESSION['id']=$aaff_user['id'];
$_SESSION['email']=$aaff_user['email'];
$_SESSION['firstname']=$aaff_user['firstname'];
$_SESSION['lastname']=$aaff_user['lastname'];
$_SESSION['password']=$aaff_user['password'];
}

```

services/loginuser.php

The authentication process is done through a PHP request on the MySQL database to check if the credentials inputted by the user are valid. MySQL guaranteed high data performance and data security.

```

<?php

#####

1)The file to check the login & password entered after insertion in the
   authentication form on register.php
2)If the login & passwords are correct and present in the database (app_users
   table)
3)This file saves the user's data (last name, first name, login, password, email)
   in session variables, which is then used in the services/check.php file

#####

include('../inc/cnx.php');
session_start();

##### check if was sent from login form #####
if ((isset($_POST['email'])) && (isset($_POST['password']))) {

##### clean email & password #####
$email=mysqli_real_escape_string($cnx,$_POST['email']);
$password=mysqli_real_escape_string($cnx,$_POST['password']);

##### check if exist in data base #####
$sql_check="SELECT * FROM `app_users` WHERE email='$email' AND password='$password'";
$req_check=mysqli_query($cnx,$sql_check);
$total=mysqli_num_rows($req_check);
$aaff_check=mysqli_fetch_array($req_check,MYSQLI_ASSOC);

##### get data from database to store int into session #####
$firstname=$aaff_check['firstname'];
$lastname=$aaff_check['lastname'];
$email=$aaff_check['email'];
$id=$aaff_check['id'];
if ($total==0) {

##### user does not exist or wrong password redirect to register
page #####
    header('Location: register');
    $_SESSION['error'] = 2;
} else {
    $_SESSION['firstname'] = $firstname;

```

```

        $_SESSION['lastname'] = $lastname;
        $_SESSION['password'] = $password;
        $_SESSION['email'] = $email;
        $_SESSION['id'] = $id;
        header('Location: start');
    }

} else {
    ##### if no data sent from form #####
    header('Location: home');
}

```

services/logout.php

- File called when user logging out.
- It deletes the session variables and redirects the user to the home page.
- *Technique used: session destroy (empty and delete session variable)*

```

<?php session_start();
session_destroy();
header('location:home');
##### on click on logout button start session destroy session and go to homepage #####

```

services/recover.php

- The file is called during the action/password forgotten allows the program to see if the entered email is present on the table app_users if yes he sends the password by email.
- *Techniques used:*
 - *Session DATA to get user data and MySQL request*
 - *Mail PHP function simple and quick email send function*

```

<?php
include('../inc/cnx.php');
session_start();

/// check if was sent from login form
if ((isset($_POST['email'])) ) {
    //clean email & password//
    $email=mysqli_real_escape_string($cnx,$_POST['email']);

    /// check if exist in database
    $sql_check="SELECT * FROM `app_users` WHERE email='$email'";
    $req_check=mysqli_query($cnx,$sql_check);
    $total=mysqli_num_rows($req_check);
    $aff_check=mysqli_fetch_array($req_check,MYSQLI_ASSOC);
    /// get password from database to send it by email
    $firstname=$aff_check['firstname'];
    $lastname=$aff_check['lastname'];
    $password=$aff_check['password'];
    $email=$aff_check['email'];
    $id=$aff_check['id'];

    if ($total==0) {
        /// email dosent exist in database

        header('Location: reset');
        $_SESSION['error'] = 1;
    } else {
        /// send email script //////////////////////////////////
        include("../emailings/reset.php");
        $message=$reset_message;

        $to = $email;
        $from = 'info@quicklang.net';
        $headers = "From: " . strip_tags($from) . "\r\n";
        $headers .= "Reply-To: " . strip_tags($from) . "\r\n";
        $headers .= "MIME-Version: 1.0\r\n";
    }
}

```

```

$headers .= "Content-Type: text/html; charset=ISO-8859-1\r\n";

$subject="Quicklang Password recovery";
mail($to, $subject, $message, $headers);

//////////
$_SESSION['error'] = 3;
header('Location: register');
}
} else {
header('Location: home');
}
}

```

services/adduser.php

- The program registers a player and add this information on the table app_users (database) in case it is not already registered.
- *Technique used:*
 - *PHP and MySQL Insert Request for adding data to user table*

```

<?php
include('../inc/cnx.php');
session_start();
#####check if there is a data coming from register form #####
if ( (isset($_POST['firstname'])) && (isset($_POST['lastname']))&&
(isset($_POST['email'])) &&(isset($_POST['password'])) ) {

#### using mysql_real_escape_string function to clear data coming from register form ####
$firstname=mysqli_real_escape_string($cnx,$_POST['firstname']);
$lastname=mysqli_real_escape_string($cnx,$_POST['lastname']);
$email=mysqli_real_escape_string($cnx,$_POST['email']);
$password=mysqli_real_escape_string($cnx,$_POST['password']);
##### get user ip #####
$ip = $_SERVER['REMOTE_ADDR'];

##### check if user is not already member #####
$sql_check="SELECT * FROM app_users where email='$email'";
$req_check=mysqli_query($cnx,$sql_check);
$total=mysqli_num_rows($req_check);
if ($total>0) {
    $_SESSION['error'] = 1;
    $_SESSION['error_email'] = $email;
    header('Location: register');
} else {
    ##### if not then registre the user and save data in session #####
    $sql_sign="INSERT INTO `app_users` (`id`, `firstname`, `lastname`, `email`, `password`,
`date`, `ip`) VALUES ('', '$firstname', '$lastname', '$email', '$password', now(),
'$ip')";
    $query_sign=mysqli_query($cnx,$sql_sign);

    $_SESSION['firstname'] = $firstname;
    $_SESSION['lastname'] = $lastname;
    $_SESSION['email'] = $email;
    $_SESSION['password'] = $password;
    #####send welcome email script #####
    include("../emailings/register.php");
    $message=$register_message;

```

```
$to = $email;
    $from = 'no-reply@quicklang.net';
    $headers = "From: " . strip_tags($from) . "\r\n";
    $headers .= "Reply-To: " . strip_tags($from) . "\r\n";
    $headers .= "MIME-Version: 1.0\r\n";
    $headers .= "Content-Type: text/html; charset=ISO-8859-1\r\n";

$subject="Quicklang Account";
    mail($to, $subject, $message, $headers);
### after sending email go to register page to insert login & password newly created ###
        header('Location: register');
    }
} else {
    //header('Location: ../index.php');
}
```

Services for the games

Each game has their own game services to authenticate the user's answer inputs. The files are the following:

- 1) services/checkanswer.php → *Writing Game*
- 2) services/checkanswer2.php → *Picture Game*
- 3) services/checkanswer3.php → *Complete the Sentence Game*
- 4) services/checkanswer4.php → *Complete the Pair Game*
- 5) services/checkanswer5.php → *Listen Game*
- 6) services/checkanswer6.php → *Translation Game*

Due to the word count limit, I will be explaining only of the files above to describe the functionalities used.

services/checkanswer.php

This file allows the verification of the user's answer input / Picks which user is playing from the page **writing.php**

The following data is sent by the page **writing.php** in the format of AJAX via jQuery

1. id_game (Game ID 1 - *Writing Game*)
2. question_list (List of questions pre-generated)
3. answer (Chosen answer for previous question)
4. score (Player's score during the game session)
5. id_question (Question ID)

Technique used:

MySQL Select Request for getting the right answer then insert answer to add score to score table.

```
<?php
#####
This file allows the verification of the user's answer input. If it's a
good answer: add +1 to the score, generate a new question via MySQL request
from the table questions (make sure the question is not already asked to
the user by comparing it with the list of questions already generated by
the request), send a response in JSON format containing the following
elements:
    - New question
    - Proposed answers
    - Score update
If the question asked is the 10th (last) question this file saves the score
in the apps_scores table and displays a message with the final score with
two buttons for "Replay" or "Choose another game."
#####
include('../inc/cnx.php');
session_start();
$id_user=$_SESSION['id'];
```

```

$total_questions=10;
##### getting and cleaning of answer variables from writing
page id_game,question_list,answer,score,id_question (last
question)#####
$id_game=mysqli_real_escape_string($cnx,$_POST['id_game']);
$question_list=mysqli_real_escape_string($cnx,$_POST['question_list']);
$answer=mysqli_real_escape_string($cnx,$_POST['answer']);
$score=mysqli_real_escape_string($cnx,$_POST['score']);
$id_question=mysqli_real_escape_string($cnx,$_POST['qst']);
##### first detecting if correct answer from answers
table#####
$sql_score="SELECT * FROM answers WHERE answer_text='".$answer.'" AND
id_question='".$id_question.'"";
$req_score=mysqli_query($cnx,$sql_score);
$affect_score=mysqli_fetch_array($req_score,MYSQLI_ASSOC);
##### if answer is correct add +1 to score
#####"
$new_score=$affect_score['correct']+$score;
$last_score=" <strong>".$new_score."</strong>/10";
$html_answer='';

##### check how many remaining questions to show based on
question list variable sent by game page#####"
$questionArray = explode(',', $question_list);
$count=count($questionArray);
#####"if remaining question then get new one
not included previous list #####"
$hsab=($count-1);
if($hsab<$total_questions) {
    $sql_next="SELECT * FROM questions WHERE id_game='".$id_game.'" AND id NOT
IN(".$substr($question_list,0,-1).") ORDER BY RAND() LIMIT 1";
    $req_next=mysqli_query($cnx,$sql_next);
    $affect_next=mysqli_fetch_array($req_next,MYSQLI_ASSOC);
    $next_question=$affect_next['question_text'];
    ##### getting new question and store it in
variables that will be sent to game page in html format ( writing) #####

$sql_anw="SELECT * FROM answers WHERE id_question='".$affect_next['id']."' ORDER BY
RAND() LIMIT 4";
$req_anw=mysqli_query($cnx,$sql_anw);
#####applying random bootstrap style for answers options #####"
$i = 1;
$array_cls = ['primary', 'secondary', 'success', 'danger', 'warning', 'info',
'light', 'dark'];
shuffle($array_cls);
while ($affect_anw=mysqli_fetch_array($req_anw,MYSQLI_ASSOC)) {

$html_answer.='<span style="font-size:14px;margin-top:3px" class="badge badge-pill
badge-'. $array_cls[$i].'"> '.$affect_anw['answer_text'].'</span>';
$i++;
}

$new_list=$question_list.$affect_next['id'].',';

```



```

$question_div = "<h3>".$next_question."</h3>";
$send=0;
#####end 0 mean there is other question player not reaching 10 questions
#####
#####send all data question,score, to game page in json format supported by
jqueryy #####
echo json_encode(['end'=>$send, 'score'=>$last_score, 'qst'=>$aff_next['id'],
'score_val'=>$new_score, 'question'=>$question_div,
'question_list'=>$new_list,'html_answer'=>$html_answer]);

} else {
#####if player reaches 10 questions the program sends a message in html
format with end message #####
$send=1;
$html_end='<div class="alert alert-success" role="alert">
  You have completed the quiz and your score is <strong>'.$new_score.'</strong>
</div>
<a href="writing" class="btn btn-primary">Retry</a>
<a href="start" class="btn btn-success">More games</button>

';
#####if reaching 10 questions saving score, with id game, and
id_ser and date now() mean instant date #####
$sql_inset_score="INSERT INTO app_scores
VALUES('','$id_user','$id_game','$new_score',now())";
mysqli_query($cnx,$sql_inset_score);
#####send response to game page in json format (format supported by
jquery)#####
echo json_encode(['end'=>$send, 'score'=>$last_score, 'score_val'=>$new_score,
'html_end'=>$html_end]);
}

```

Admin Panel Files

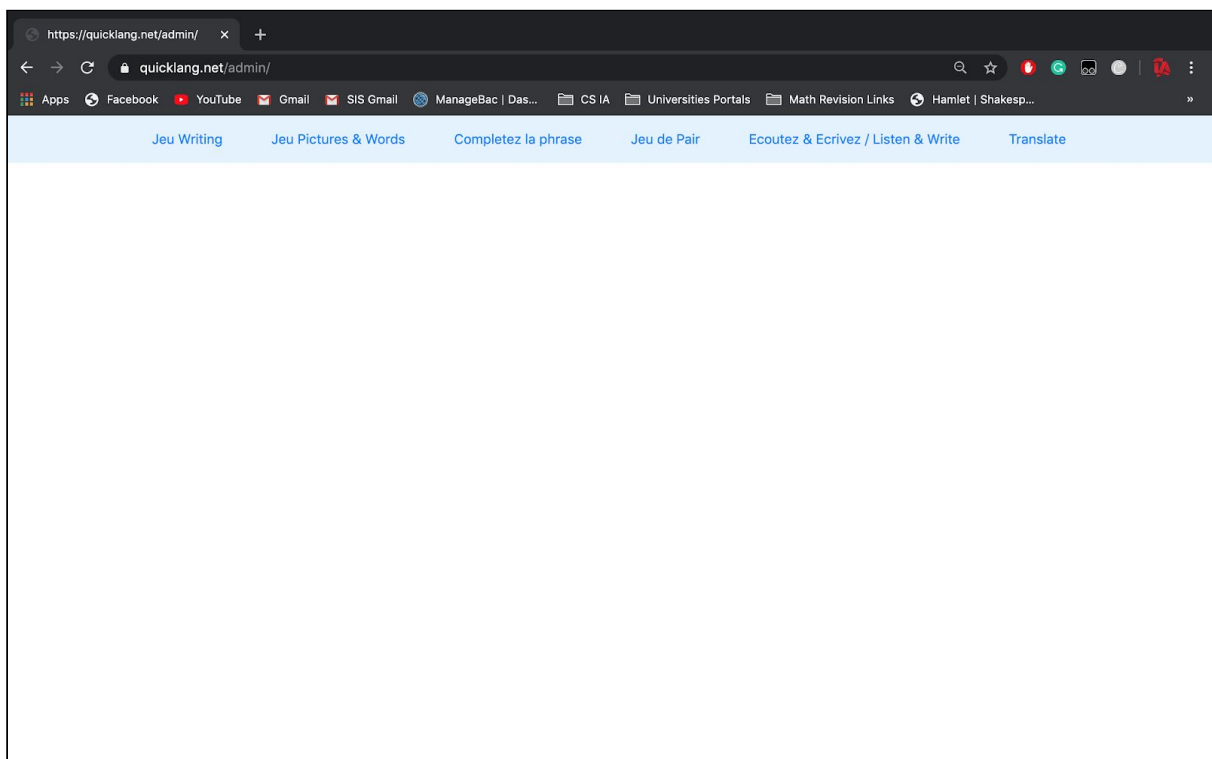
In this section, I will be explaining the code and techniques used that allowed my client to edit the games. These files allow the adding of new questions, answers to the questions, editing the pre-existing questions or answers, and lastly, being able to delete the question for particular games. The files operating the admin panel can be found in *Appendix D*.

Explanation of techniques used

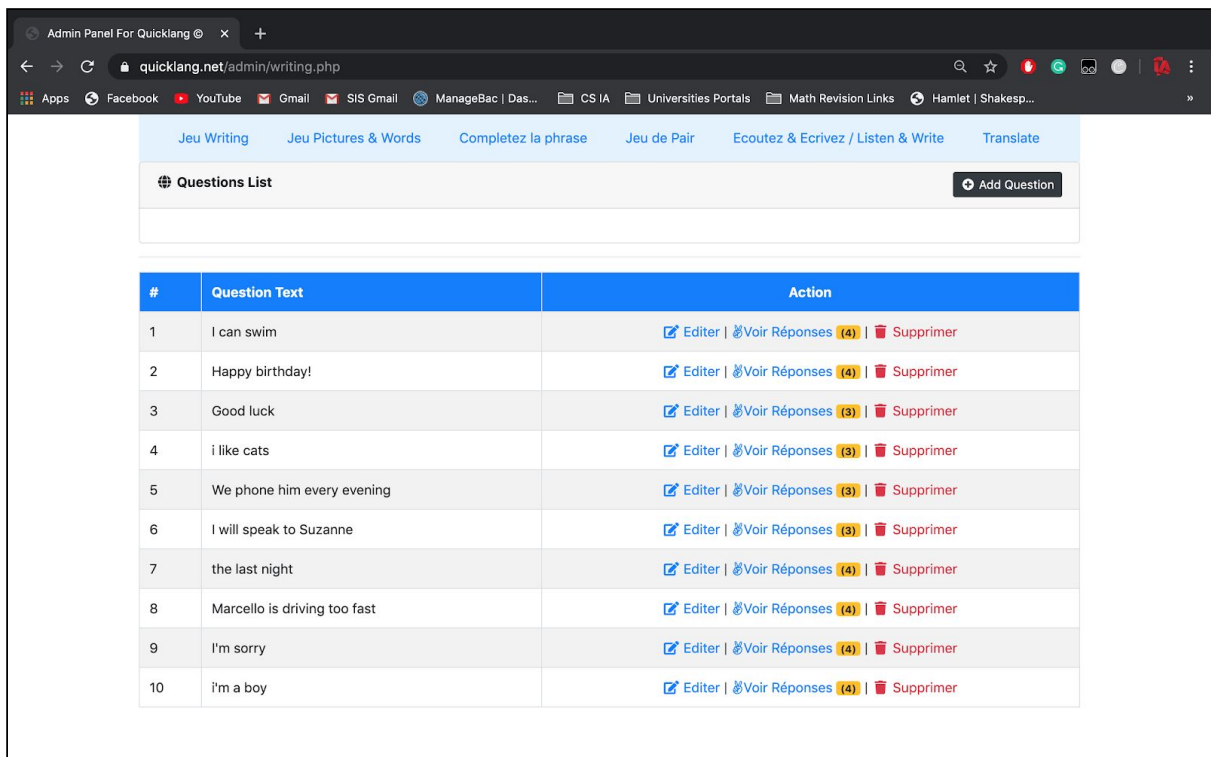
- Admin Backend Used Simple CRUD functions for PHP 5
- CRUD stands for Create-Read-Update-Delete
- Requests are used to add edit or delete questions or answers to the database.
 - *More details included in Appendix D*

Screenshots of display

admin.php (Admin Panel)



admin/writing.php (Menu)



Admin Panel For Quicklang

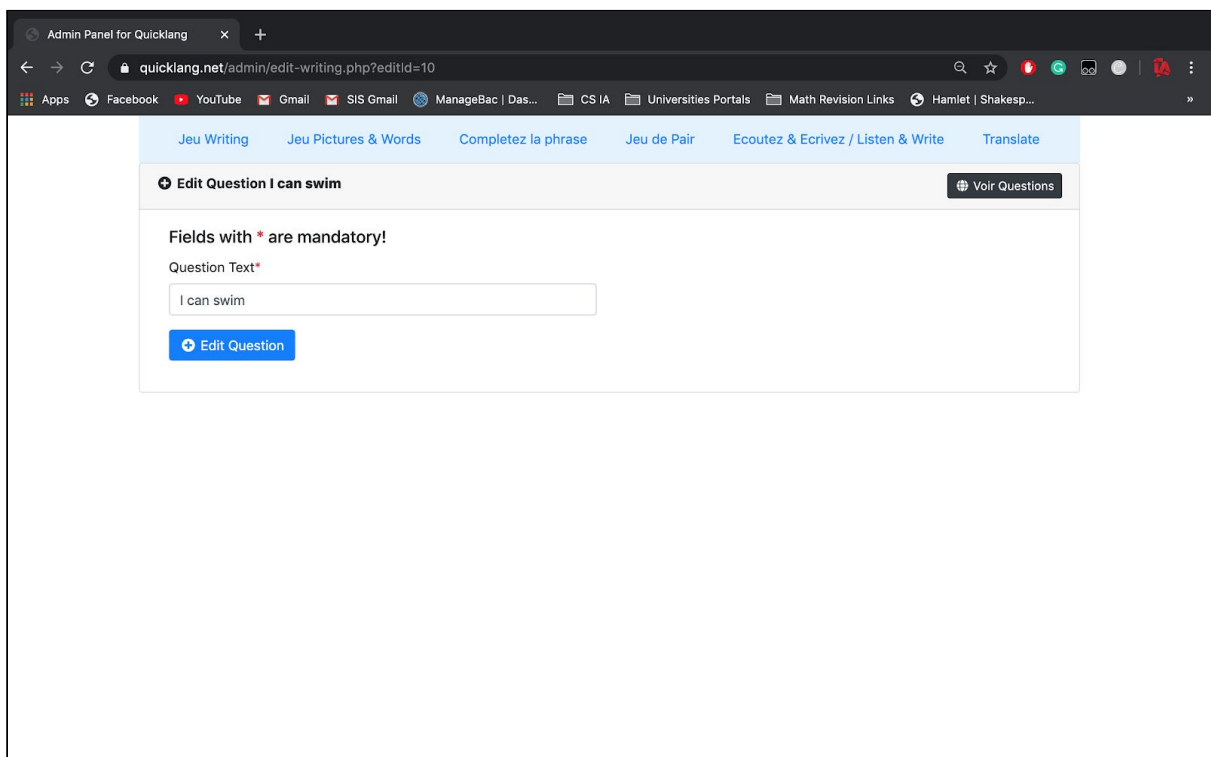
quicklang.net/admin/writing.php

Jeu Writing | Jeu Pictures & Words | Completez la phrase | Jeu de Pair | Ecoutez & Ecrivez / Listen & Write | Translate

Questions List [Add Question](#)

| # | Question Text | Action |
|----|------------------------------|--|
| 1 | I can swim | ✎ Edit 👁 Voir Réponses (4) 🗑 Supprimer |
| 2 | Happy birthday! | ✎ Edit 👁 Voir Réponses (4) 🗑 Supprimer |
| 3 | Good luck | ✎ Edit 👁 Voir Réponses (3) 🗑 Supprimer |
| 4 | i like cats | ✎ Edit 👁 Voir Réponses (3) 🗑 Supprimer |
| 5 | We phone him every evening | ✎ Edit 👁 Voir Réponses (3) 🗑 Supprimer |
| 6 | I will speak to Suzanne | ✎ Edit 👁 Voir Réponses (3) 🗑 Supprimer |
| 7 | the last night | ✎ Edit 👁 Voir Réponses (4) 🗑 Supprimer |
| 8 | Marcello is driving too fast | ✎ Edit 👁 Voir Réponses (4) 🗑 Supprimer |
| 9 | I'm sorry | ✎ Edit 👁 Voir Réponses (4) 🗑 Supprimer |
| 10 | i'm a boy | ✎ Edit 👁 Voir Réponses (4) 🗑 Supprimer |

admin/edit-writing.php (Edit questions)



Admin Panel For Quicklang

quicklang.net/admin/edit-writing.php?editId=10

Jeu Writing | Jeu Pictures & Words | Completez la phrase | Jeu de Pair | Ecoutez & Ecrivez / Listen & Write | Translate

[+ Edit Question I can swim](#) [👁 Voir Questions](#)

Fields with * are mandatory!

Question Text*

[+ Edit Question](#)

admin/answer-writing.php (Edit answers)

Admin Panel For Quicklang

quicklang.net/admin/answer-writing.php?id_question=10

AppsFacebookYouTubeGmailSIS GmailManageBac | Das...CS IAUниверситет ПорталMath Revision LinksHamlet | Shakesp...

Writing Game

[Jeu Writing](#)[Jeu Pictures & Words](#)[Completez la phrase](#)[Jeu de Pair](#)[Ecoutez & Ecrivez / Listen & Write](#)[Translate](#)

Liste Réponse Pour : I can swim

Ajout Reponse

| # | Réponses Texte | Bonne Réponse | Action |
|---|----------------------|---------------|--|
| 1 | je ne peux pas nager | | Editer Supprimer |
| 2 | je peux chanter | | Editer Supprimer |
| 3 | je peux manger | | Editer Supprimer |
| 4 | je peux nager | ✓ | Editer Supprimer |

Bibliography

Piat, Jonathan M. "PgwSlider - Responsive Slider for JQuery / Zepto." PgwJS, 2014.
<https://pgwjs.com/pgwslider/>.

"Arctic Sunset UI Kit." PixelKit Premium UI Design Resources. Themefuse Family.
Accessed September 2, 2019. <http://pixelkit.com/kits/clean-ui-kit>.

Brunfeldt, Kimmo. "PROGRESS BAR.JS." ProgressBar.js - Progress bars with
JavaScript, May 28, 2019. <https://kimmobrunfeldt.github.io/progressbar.js/>.

Deutsch, David. "Owl Carousel 2." Home | Owl Carousel | 2.3.4. Accessed August 28,
2019. <https://owlcarousel2.github.io/OwlCarousel2/>.

FontAwesome, Robmadole. "FontAwesome/Font-Awesome." GitHub, July 24, 2019.
<https://github.com/FontAwesome/Font-Awesome>.

Hussey, Brad. "Free PHP Tutorial - Practical PHP: Master the Basics and Code Dynamic
Websites." Udemy. Udemy. Accessed August 19, 2019.
<https://www.udemy.com/course/code-dynamic-websites/>.

JS Foundation. "jQuery - Write Less Do More." jQuery. The jQuery Foundation.
Accessed July 19, 2019. <https://jquery.com/>.

Lipis. "Lipis/Flag-Icon-Css." GitHub, June 14, 2019.
<https://github.com/lipis/flag-icon-css>.

Otto, Mark, and Jacob Thornton. "Bootstrap." · The most popular HTML, CSS, and JS
library in the world. Accessed August 22, 2019. <https://getbootstrap.com/>.

"PHP - Sessions." Tutorialspoint. Accessed August 25, 2019.
https://www.tutorialspoint.com/php/php_sessions.htm.

“Search Results Web Results PHP Tutorial - An Ultimate Guide for Beginners.”
TutorialRepublic. Accessed July 5, 2019.
<https://www.tutorialrepublic.com/php-tutorial/>.

Troughton, Caleb. “JQuery Waypoints.” Waypoints. Github, November 2013.
<https://github.com/imakewebthings/waypoints>.

“Voice RRS.” Voice RSS - Free Text-to-speech (TTS) online service, 2018.
<http://www.voicerss.org/default.aspx>.