



**SAPIENZA**  
UNIVERSITÀ DI ROMA

# Realisation of an online interface to connect users of a role-playing game

**Faculty of Information Engineering, Computer Science and Statistics**  
**Department of Computer, Automatic and Management Engineering**  
**Degree Course in Computer and Automatic Engineering**

**Andrea Massignan**  
**Serial no. 1796802**

Rapporteur  
Riccardo Rosati



# Index

<b>Chapter 1 - Introduction</b>	<b>4</b>
1.1 - What is Dohr?	4
1.2 - What does the project consist of?	4
<b>Chapter 2 - Technologies Used</b>	<b>5</b>
2.1 - Development Environment	5
2.2 - Client	5
2.3 - Server	5
<b>Chapter 3 - Clients</b>	<b>6</b>
3.1 - Overview	6
3.2 - Site Sections	7
3.2.1 - Home	7
3.2.2 - Registration	7
3.2.3 - Factions	9
3.2.4 - Forums	10
3.2.5 - Shop	13
3.3 - Features accessible after registration	14
3.3.1 - Profile	14
3.3.2 - Guilds	15
3.3.3 - Group chats	16
3.4 - Changing language	17
<b>Chapter 4 - Servers</b>	<b>18</b>
4.1 - Shop	18
4.2 - Registration and Log in	24
4.3 - Guilds and Factions	29
4.4 - Database	31
4.4.1 - accounts	31
4.4.2 - guilds	31
<b>Chapter 5 - How the project could be expanded</b>	<b>32</b>
5.1 - Improvement of the game shop	32
5.2 - Improving the Registration System	33
5.3 - Mailing list	33

**Bibliography**

**Sitography**

# Chapter 1 - Introduction

## 1.1 - What is Dohr?

The online video game Dohr is characterised by a fantasy context characterised by war between two factions: Drariva and Providentia. Drariva, which is also the name of the Drakorian capital, is the most powerful magic centre in the world. Situated along the coast, it gathers knowledge from all over the world with the help of capacious ships. Famous for its trade, Drariva's culture focuses on the study of ancient dragons, considered to be the most intelligent creatures.

Providentia, on the other hand, is an enormous, highly technological megalopolis. The perfectly circular urban structure rises tens of metres above the ground, supported by pillars; it is divided into eight sectors, below which are the slums that originally made up the city proper; the eight sectors were once separate cities, but were later merged and their names forgotten. The player, as one of the two sides, must influence the fate of the conflict and change its outcome.



Figure 1. Dohr logo

## 1.2 - What does the project consist of?

The project involves the creation of a website that acts as an online interface for the video game, creating a virtual environment in which players can exchange additional information about the game world and manage their inventory. The interface itself is designed to be simple and intuitive, but at the same time aesthetically consistent with the theme of the video game.

The main objective is to ensure that the system encourages communication between users, and its operation will be examined in the following paragraphs.

## Chapter 2 - Technologies Used

### 2.1 - Development Environment

Visual Studio Code with the following extensions was used to write the code:

- HTML CSS Support and HTML Snippets
- JavaScript (ES6) Code Snippets
- PHP IntelliSense

XAMPP software was also used for database management and for the interpretation of dynamic web pages.

### 2.2 - Client

The following technologies were used to realise the client part of the project:

- HTML: well-known markup language used for constructing web pages.
- JavaScript: a programming language useful for creating functions that are invoked by events generated by user interaction with elements on a web page.
- JQuery: JavaScript library used to make script writing faster.
- CSS: language used for creating style sheets for HTML pages
- Bootstrap: CSS framework for creating interfaces
- AJAX (Asynchronous JavaScript and XML): a technique that allows, via JavaScript, the dynamic, background loading of portions of web pages without the need to reload them entirely.

### 2.3 - Server

The following technologies were used for the server:

- PHP (PHP HyperText Preprocessor): an interpreted scripting language that allows web pages to be generated dynamically following a client request.
- MySQL: database service.
- PDO (PHP Data Object): an extension of PHP that offers a unique interface for communication with databases and increased protection against SQL Injection-based threats.

## Chapter 3 - Clients

### 3.1 - Overview

Each page of the site is structured as follows:

1. a bar at the top, within which there is a 'form' for logging in and a button for changing language. If the user has logged in with his own credentials, it will also contain information about his profile;
2. at the top of the page is the site logo;
3. the page includes a Navbar that provides access to the various sections of the site;
4. The side of the page contains information about the server (time, server status, number of subscribers), two banners linking to the game's Instagram and Facebook pages, and a YouTube mini-player allowing you to watch a short trailer;
5. The central part of the page changes according to the section of the site selected via the Navbar (in the case of the homepage, it contains a small welcome message).

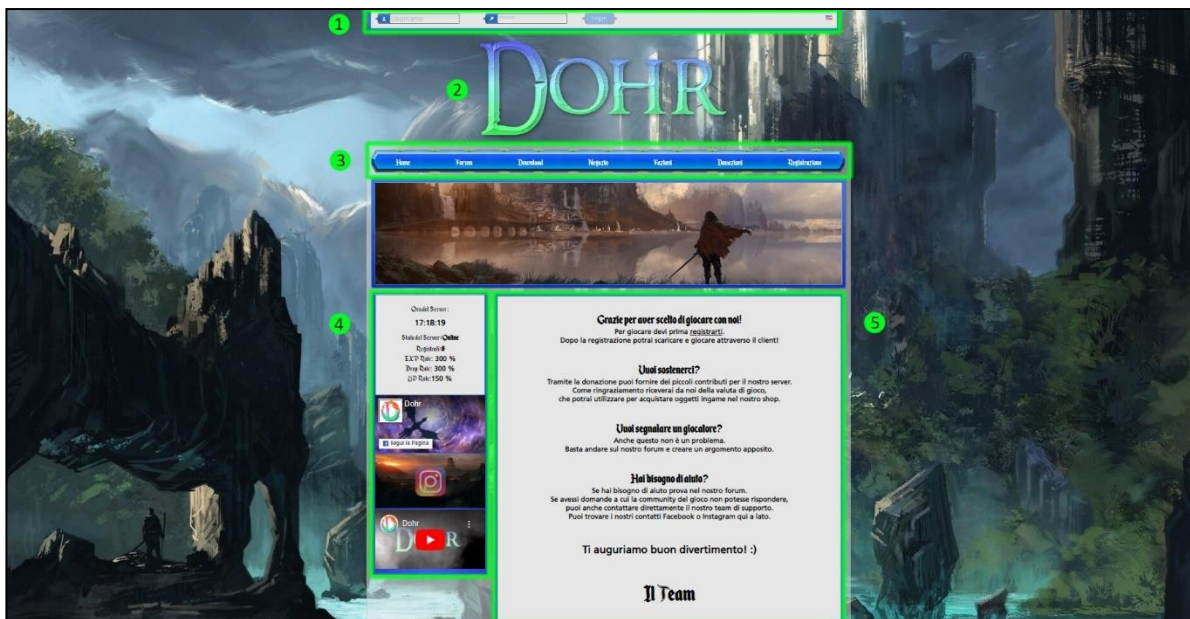


Figure 2. Page structure

## 3.2 - Sections of the site

### 3.2.1 - Home

The home page features a short welcome message.

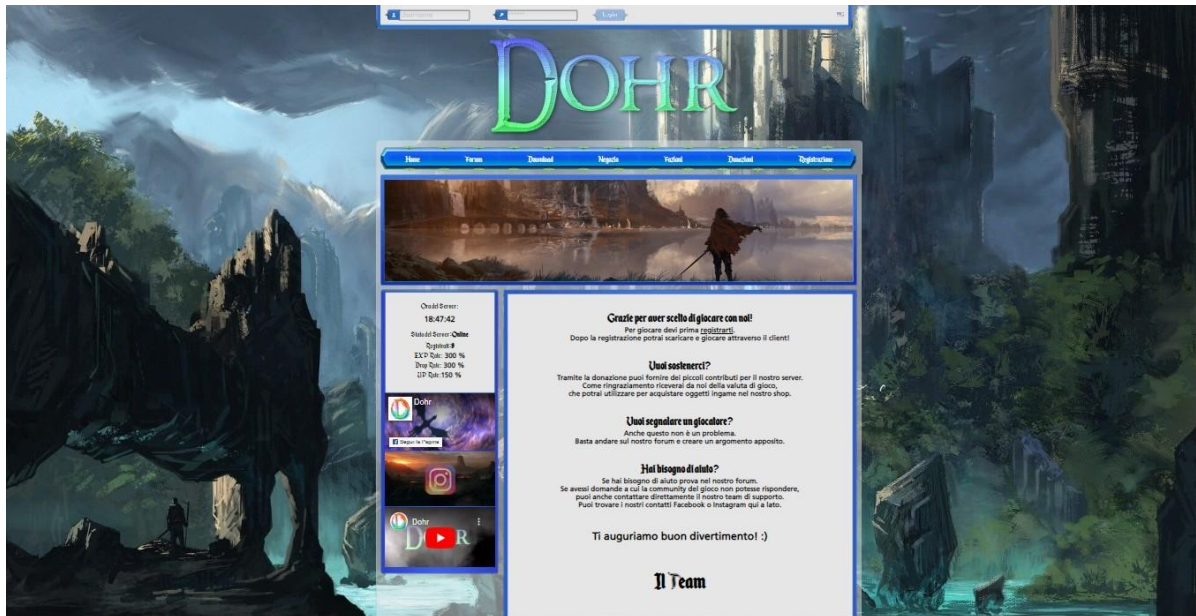


Figure 3. Index

### 3.2.2 - Registration

The registration section allows users to create an account; this is done by filling in a 'form', which requires the entry of a user name, a password and its confirmation. In addition, users are invited to select the logo of the faction they like, so as to distinguish their side within the game world.

Figure 4. Registration form

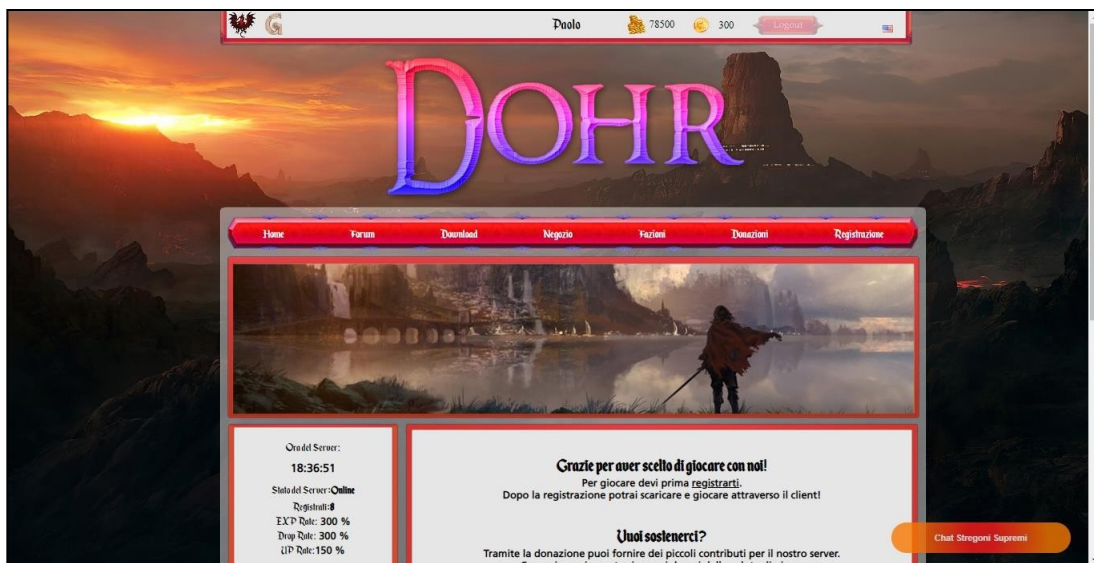


The choice of faction leads to an immediate graphical transformation of the interface. For instance, the choice of Drariva gives the site a vibrant red colour palette, with the intention of immersing users even more in the game's fantasy atmosphere. Conversely, Providentia's choice does not change the default blue tones. At the end of the registration process, the user is automatically directed to the main page of the portal, and his profile is already active. In addition, upon first access, a pop-up explains the additional features exclusive to registered users.



Figure 5. Site in blue version

Figure 6. Site in red version



### 3.2.3 - Factions

On the page of the same name are the logos of the two factions you can join, Drariva and Providentia.



Figure 7. Logos of the factions

Clicking on them will bring up a popup describing the history of the faction.

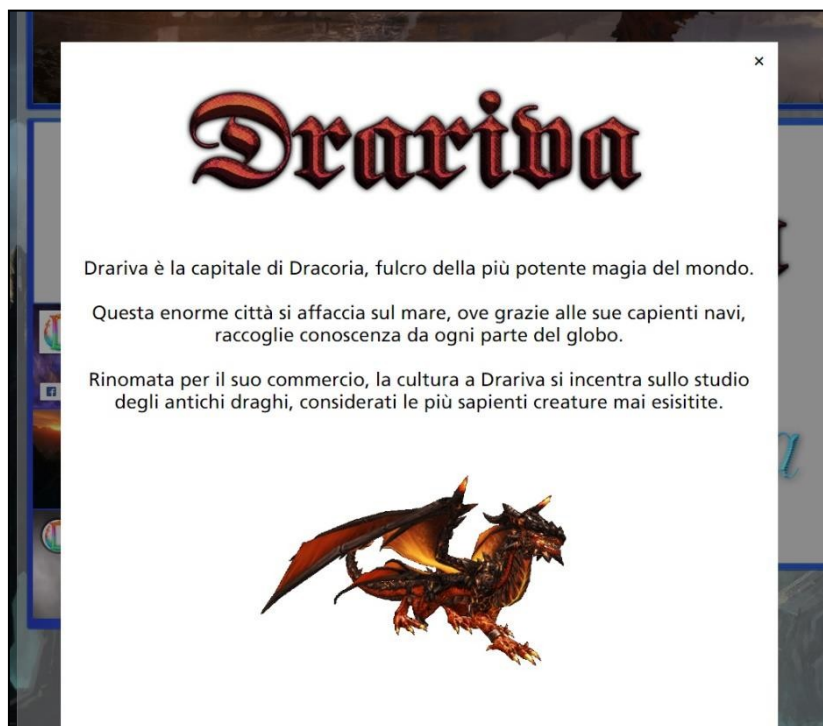


Figure 8. Drariva faction popup

### 3.2.4 - Forums

The Forum section contains a welcome message and the forum title.

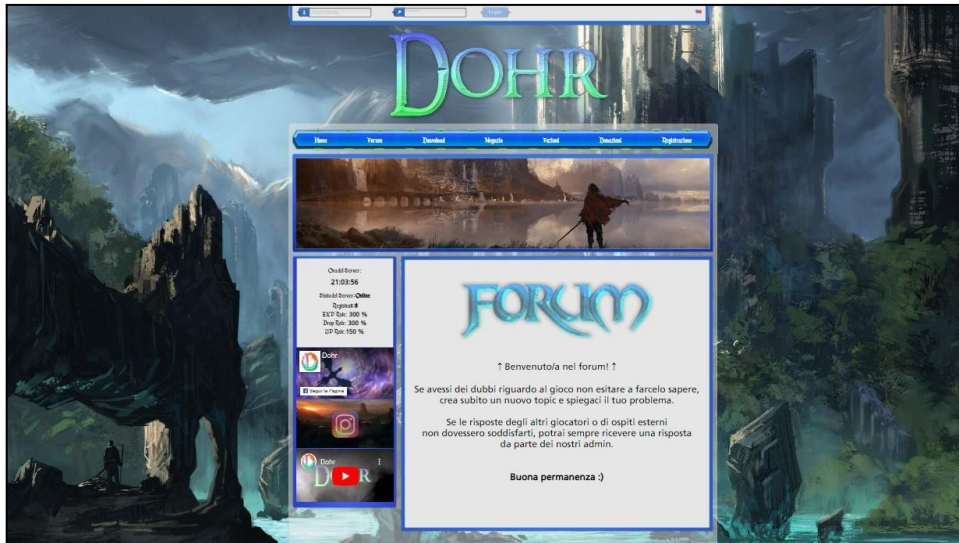


Figure 9. Forum

Clicking on the latter provides access to the list of topics present within it, ordered from least to most recent, and characterised by a title and an author. We preferred to add this double step (click on the Forum button on the Navbar followed by a click on the forum title) and not immediately display the list of topics present, in order to allow the possible future addition of other forums without having to substantially modify the server side.

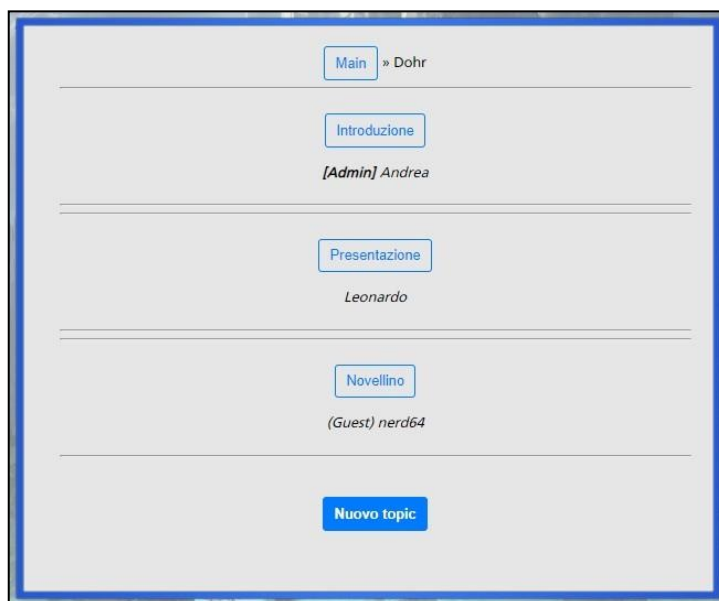


Figure 10. List of topics

Being on the page, the forum presents the path at the top:

(in the format Main >> Forum\_Name >> Topic\_Title).

Followed by the list of topics and the button to create a new discussion. This takes you to a form with three fields: Title, Nickname and Message. The first represents the title of the new post, the second the author, and the third the content of the message opening the thread. All fields are mandatory and if you try to omit one of them you will be warned and the topic will not be created. It is important to note that the Nickname field is disabled if you are logged in, for two reasons. The first is that, if you are logged in, the 'form' will automatically enter the username of the person who is creating the topic as Nickname, and the second is that you have chosen to give unregistered visitors the opportunity to use the forum. By clicking on Create, the topic will be started and you will be redirected to the list of threads, at the bottom of which you will now also find the one you have just added.

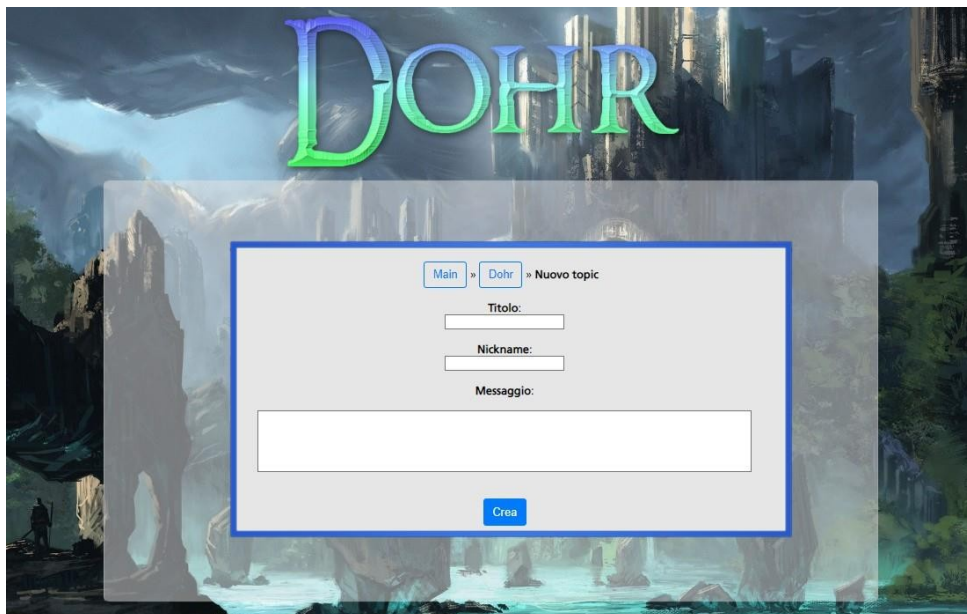
The image shows a web browser window displaying the 'DOHR' forum. The background is a fantasy landscape with a large, colorful 'DOHR' logo at the top. In the center, there is a white form box with a blue border. Inside the form, at the top, are navigation links: 'Main' and 'Dohr' followed by '» Nuovo topic'. Below these are three input fields: 'Titolo:' (Title), 'Nickname:', and 'Messaggio:' (Message). The 'Nickname:' field is disabled. At the bottom right of the form is a blue button labeled 'Crea' (Create).

Figure 11. Form for creating a new topic

Clicking on the title of a topic takes you to the list of posts contained within it, always ordered from least to most recent and characterised by an author and content. Next to the name of the creator it will also be indicated whether or not he is one of the site administrators.



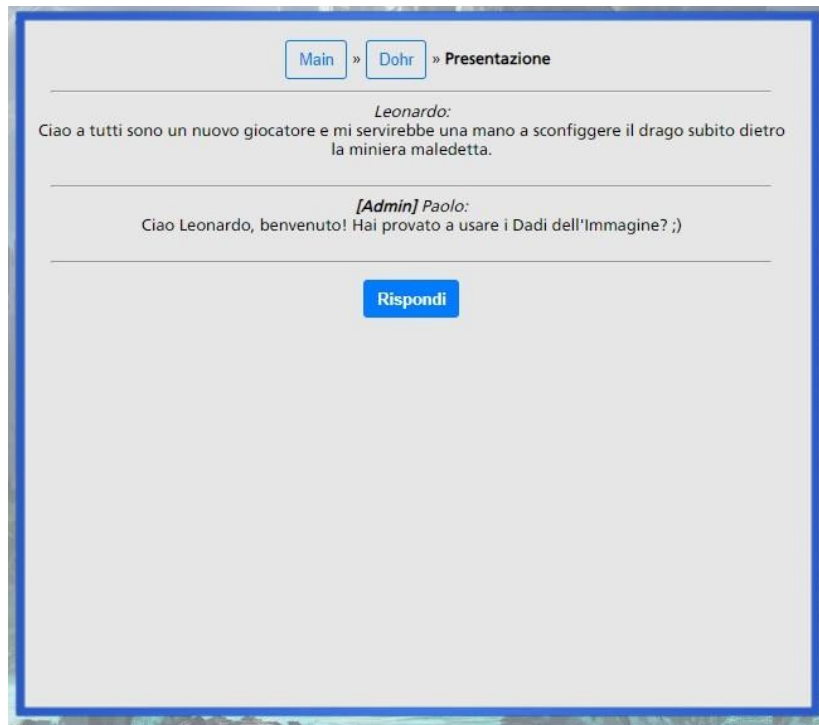


Figure 12. Messages contained in the topic "Presentation"

At the bottom is a 'Reply' button, which will take you to a 'form' that, similarly to before, will allow you to reply to the current discussion.

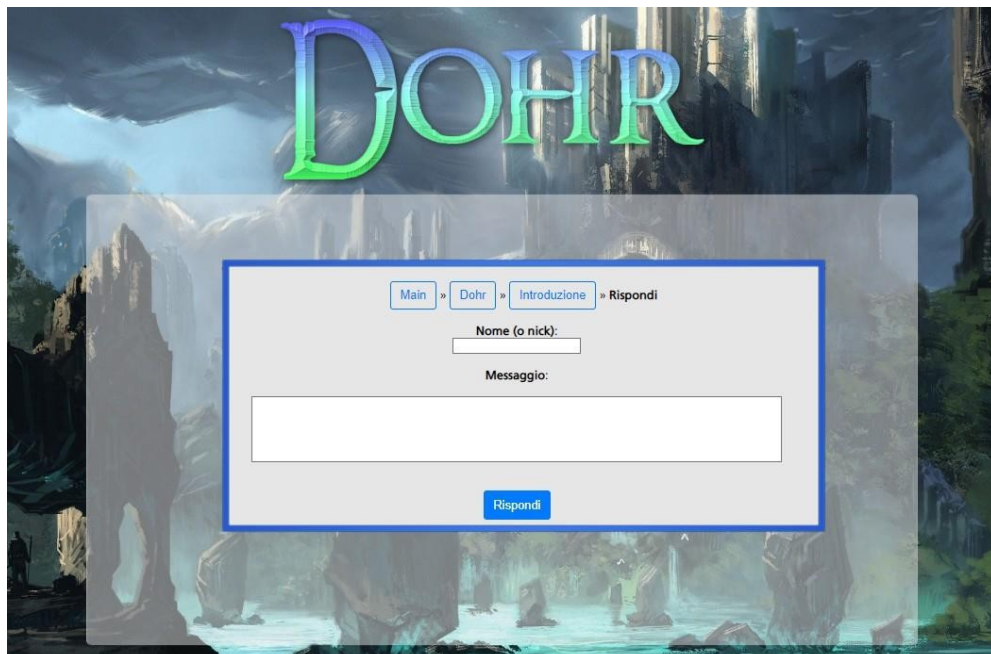


Figure 13. Form for creating a new message

### 3.2.5 - Shop

In the 'Shop' section of the site, visitors have the possibility of accessing a virtual area dedicated to the purchase of equipment and items by spending one of the two available currencies. However, access to the shop is only allowed to users who are logged in. If the user is not logged in, a message will be displayed inviting registration or login to access the virtual shop. Purchased items will be linked to the user's account and can be viewed in the 'Profile' section of the site (see section 3.3.1).



Figure 14. Shop

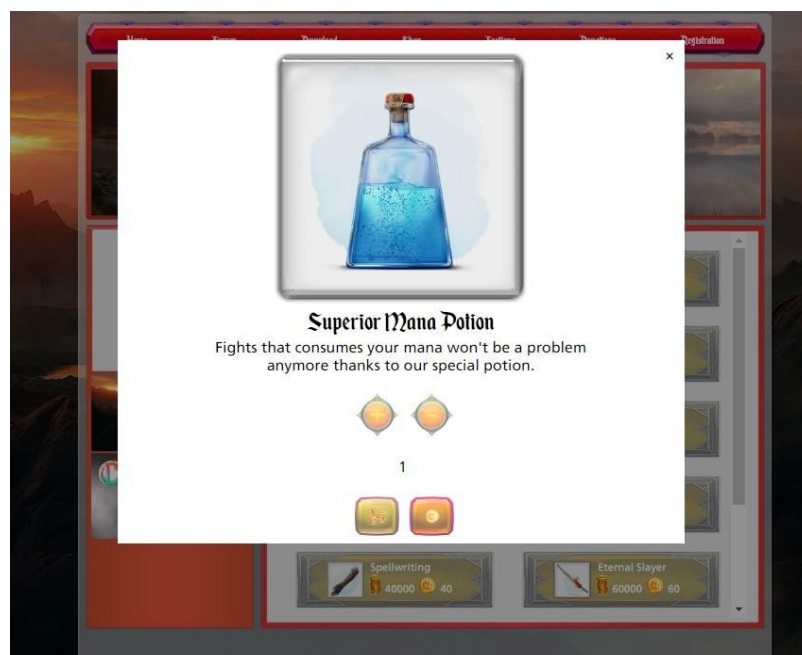


Figure 15. Popup for purchasing an item from the shop

### 3.3 - Features accessible after registration

Once registered, the following functionalities will be unlocked.

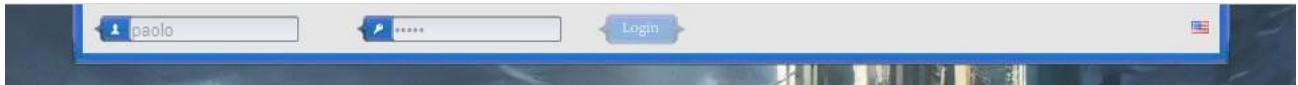


Figure 16. Upper bar before logging in

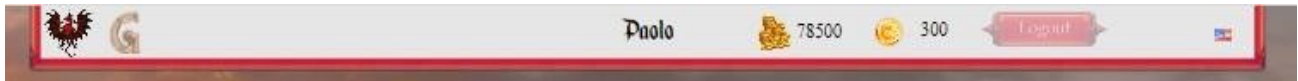


Figure 17. Upper bar following access

#### 3.3.1 - Profile

The top bar of the site displays the user name and available coin balance. Through interaction with the name, a popup appears showing personal profile information. The dedicated section highlights your name, profile picture (which can be customised by uploading an image), your role within your guild and your affiliation (see section 3.3.2). In addition, it is possible to view one's inventory.

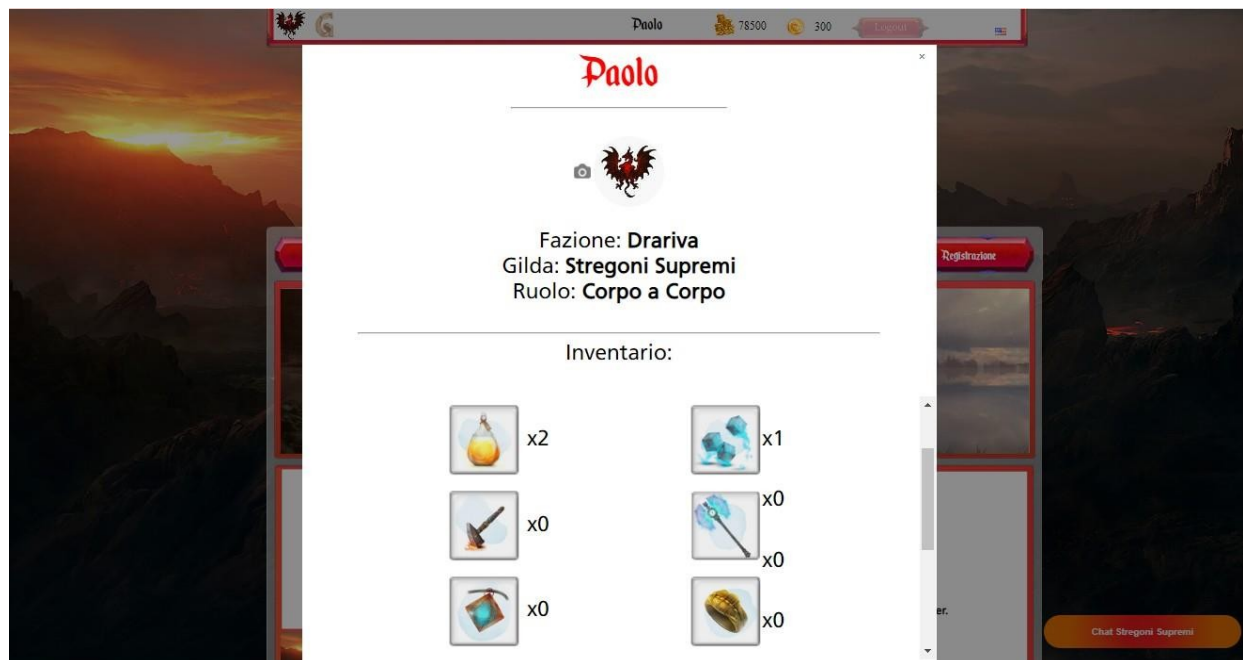


Figure 18. User profile

### 3.3.2 - Guilds

By clicking on the letter 'G' in the top bar, you can access the 'Guilds' section via a pop-up. Here you can see the symbols of the five available guilds: Sorcerers Supreme, Paladins of the Eternal, Ancient Knowers, Guardians of Chaos and Treasure Hunters. By clicking on one of them, you will see the name of the guild, while with the two buttons below you can respectively see the list of members belonging to the selected guild and join it (or leave it if you are already a member). Before joining one of the groups, you can choose the role you wish to play from the four available (Melee, Defender, Magic Offensive and Healer). Once you become a member of a guild, you have access to a group chat with the other members of the guild (see section 3.3.3).

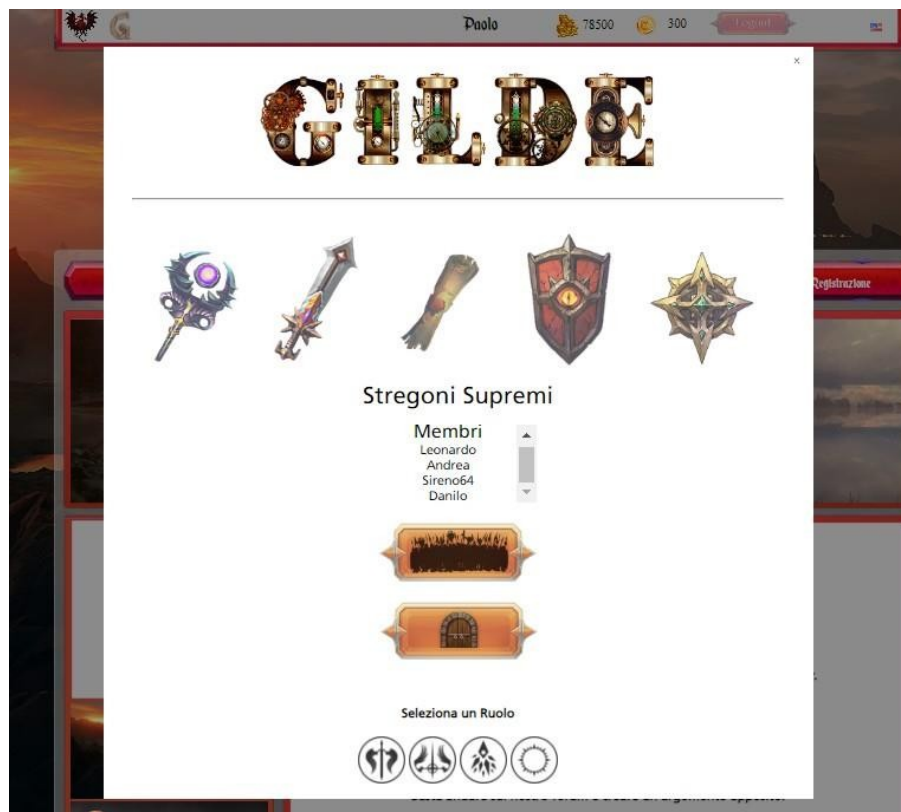


Figure 19. Popup to display guilds



### 3.3.3 - Group chat

Fixed at the bottom right of the screen is the button to access the group chat. This is only available if the user is logged in and belongs to a guild.

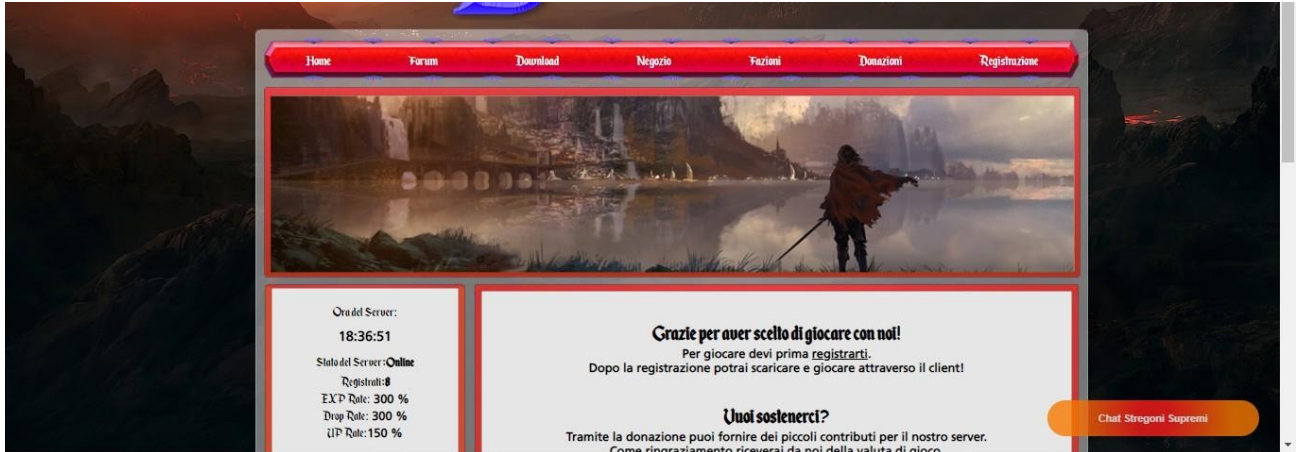


Figure 20. Chat position

Pressing the corresponding button activates the expansion of the button itself, revealing the chat. This, suitably reduced in size to allow simultaneous navigation of the site, is organised in such a way as to display the name of the group with which one is communicating at the top of the window, the messages sent and received in the centre (navigable), a text field in which to write one's messages with the corresponding send button, and the button to fold the chat into an icon. Messages are identified by the sender's name, which is highlighted in blue or red depending on the faction chosen at the time of registration.

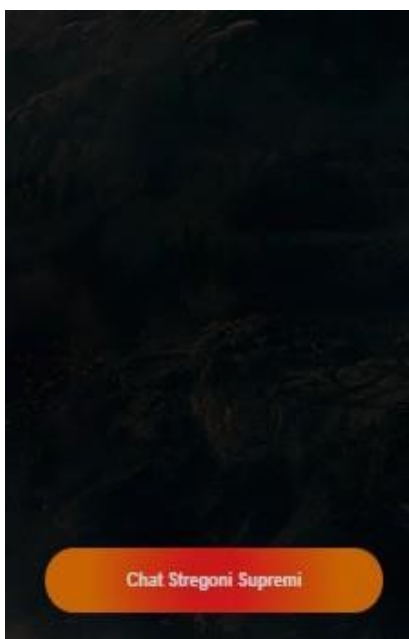


Figure 21. Chat reduced to an icon



Figure 22. Chat open

Chat functionality is provided in real time, being notified of the arrival of a new message through an audible signal as well. In addition, if the chat is minimised to an icon, an additional visual notification is given via a coloured indicator.

### 3.4 - Language change

Fundamental to promoting broad user participation is the possibility of changing the display language of the site. In the top right-hand corner of the navigation bar, there is a flag icon, through which you can select the desired language and automatically translate all site content. Currently, only two languages are available: Italian and English. However, the language change system is designed to be easily expandable.



Figure 23. Button for the Italian language



Figure 24. Button for the English language

## Chapter 4 - Servers

The server-side operation of certain sections of the site is now analysed.

### 4.1 - Shop

Its operation is based on the interaction with the accounts table, here the user's 'gold' and 'units' values are read out, similarly, as the shop contains the purchase requirements for certain items, players can evaluate their purchase based on their possessions. (Please refer to section 4.4 for details on the database structure). On the 'shop.php' page, a check is performed using the variable '\$conduct' in order to verify whether the user is currently authenticated. Depending on the outcome of the condition, the visibility of the content of the 'div' in which the shop representation is displayed is adjusted accordingly. If the check fails, the variable '\$cannotshop' is called and the visitor is shown an invitation to register, as the shop functionalities are only available to registered users. Alternatively, if the check is successful, the variable '\$shop' containing the shop structure is called.

```
<div id="Content">
  <div id="above"<?php echo $style?>"></div>
    <div id="centre"<?php echo $style?>">

      <?php
        //Log in
        if($conduct == 0){
          echo $cannotshop;
        }

        else{
          echo $shop;
        }

      ?>

    </div>
  <div id="under"<?php echo $style?>"></div>
</div>
```

On each page of the site, if the user is authenticated, the following checks are performed in order to read and display their currency availability.

```
<?php if($conduct == 1) {?>
    <?php
        //////////// READING GOLD ////////////
        $stmt = $conn->prepare("Select gold FROM accounts where
szUserID='".$$_SESSION["Username"]."'");
        $stmt->execute();
        while ($checkgold = $stmt->fetch(\PDO::FETCH_ASSOC)){
            $_SESSION['Gold'] = $checkgold['gold'];
        }

        //////////// READING UNITS ////////////
        $stmt = $conn->prepare("Select units FROM accounts where
szUserID='".$$_SESSION["Username"]."'");
        $stmt->execute();

        while ($checkunit = $stmt->fetch(\PDO::FETCH_ASSOC)){
            $_SESSION['Units'] = $checkunit['units'];
        }

        //////////// CHECK IF YOU OVERCOME MONEY LIMIT //////////// if(
$_SESSION["Units"] > 99999){
            $stmt= $conn->prepare("update accounts set units = 99999 where
szUserID='".$$_SESSION["Username"]."'");
            $stmt->execute();
            while($checkunit = $stmt->fetch(\PDO::FETCH_ASSOC)){
                $_SESSION['Units'] = $checkunit['units'];
            }
        }

        if($_SESSION["Gold"] > 999999){
            $stmt= $conn->prepare("update accounts set gold = 999999 where
szUserID='".$$_SESSION["Username"]."'");
            $stmt->execute();
            while ($checkgold = $stmt->fetch(\PDO::FETCH_ASSOC)){
                $_SESSION['Gold'] = $checkgold['gold'];
            }
        }
    }
?>
```

Then the elements that make up the items in the shop are arranged from the variable '\$shop'.

```
<div id="shop" style="font-family: frutiger; font-size: 17px; colour:white;">

<!--HEALTH-->
<div id="back "onclick="document.getElementById('id01').style.display='block'">

    <div id="health"></div>
        <div id="titletext">Superior Care<br>
        <div id="gold"></div>
        <div id="goldprice">500</div>
        <div id="unit"></div>
        <div id="unitprice">1</div>
    </div>
</div>

<!--MANA-->
<div id="back "onclick="document.getElementById('id02').style.display='block'">

    <div id="mana"></div>
        <div id="titletext">Mana Superior<br>
        <div id="gold"></div>
        <div id="goldprice">500</div>
        <div id="unit"></div>
        <div id="unitprice">1</div>
    </div>
</div>

...

</div>
```

The shop elements are arranged in a grid, through which the user can scroll through the various products available. Each product is associated with a modal window that allows the user to select the quantity and type of currency used for payment.

The structure of the modal window is characterised by the use of Javascript in order to allow the user to select the desired quantity of articles, and a 'form' that links to the page dedicated to handling payment.

```
<div id="id01" class="w3-modal">
  <div class="w3-modal-content">
    <div class="w3-container">
      <span onclick="document.getElementById('id01').style.display='none';
        document.getElementById('healthquantity').innerHTML=0;
        document.getElementById('healthquantitytrue').value=0"
        class="w3-button w3-display-topright">&times;
      </span>

      <div id="health"></div><br>
      <font size='8' face='enchant'>
        Superior Cure Potion
      </font>

      This fantastic potion restores an incredible amount of hit points
      to its drinker, making him ready to face the most
      fearsome dangers.

      <input type=button class="plusbutton" onclick="buttonplushealth()"/>
      <input type=button class="minusbutton"
      onclick="buttonminushealth()"/>

      <form action="/languages/itembuy?language=$language" method="POST">
        <span id="healthquantity" class="itemquantity">0</span><br>
        <input id="healthquantitytrue" type="hidden" name="quantity"
        value="0">
        <input type="hidden" name="item" value="healthpotions">
        <input type="hidden" name="gold" value="500">
        <input type="hidden" name="units" value="1">
        <input type="submit" name="action" class="buy_gold" value="gold">
        <input type="submit" name="action" class="buy_units" value="units">
      </form>

    </div>
  </div>
</div>
```

...

Following the decision to pay with one of the two available currencies, the user is redirected to the 'itembuy.php' page, where several checks are performed before proceeding to the actual purchase. In particular, it is first checked that the user is correctly logged in and that the quantities are correct.

```

if($conduct == 0){
    echo "<br><div class='loading' . $style . '></div><br><br>";
    <meta http-equiv='refresh' content='3; URL=../shop?language=" . $language . ">";
}

else{

if($quantity==0){
    echo $noquantity;
    echo "<br><div class='loading' . $style . '></div>";

    <meta http-equiv='refresh' content='3;
    URL=../shop?language=" . $language . ">";
}

...

```

Then a query is performed to determine whether the money available in the user's account is sufficient for the purchase.

```

$gold_control = $conn->prepare("Select gold FROM accounts where
szUserID='" . $_SESSION["Username"] . "'");
$gold_control->execute();

while($row = $gold_control->fetch(PDO::FETCH_ASSOC)){

    $goldcontrolled = $row["gold"];

}

if($goldcontrolled<$gold){ echo

    $notenoughgold;
    echo "<br><div class='loading' . $style . '></div>";
    <meta http-equiv='refresh' content='3;URL=../shop?language=" . $language . ">";

}

```

In the next step, the processing of the purchase is done by editing the database. Depending on the outcome of the check carried out earlier, the system updates the value of the amount held by the user and adds the purchased product to its inventory.

```
$write = $conn->prepare("UPDATE accounts SET gold=gold-".$gold." ,
".$item."=".$item."+".$quantity." where
szUserID='".$_SESSION["Username"]."'");

$write->execute();

if($write){
    echo $buysuccess;
    echo "<br><div class='loading' ".$style."></div>
<meta http-equiv='refresh' content='3; URL=../shop?language=".$language.">";
}

else{
    echo $buyfailure;
    echo "<br><div class='loading' ".$style."></div>
<meta http-equiv='refresh' content='3; URL=../shop?language=".$language.">";
}
```



## 4.2 - Registration and Log In

The registration and, similarly, login procedure constitutes an indispensable step for interaction with the site's various functionalities. To proceed with registration, the user can fill in the 'form' on the 'register.php' page.

```
if (!@$fp = fsockopen("localhost", 3306, $errno, $errstr, 1)){
    echo $registeroffline;
    echo '<div id="noserverdragon"></div><br><br>';
}

else{

    echo '

    <form action="registerclose?language=' . $language . '" method="POST">

        <!--Username Box-->
        <div id="username" . $style . ">
            <div id="login-box-field"><input name="login-box-field"><input name="UsernameS"
            class="form-login" title="Username" placeholder="Username" size="30"
            maxlength="17" required/>
        </div>
        </div>

        <!--Password Box-->
        <div id="Password" . $style . ">
            <div id="login-box-field"><input name="PasswordS" type="password"
            class="form-login" title="Password" placeholder="*****" size="30"
            maxlength="17" required/>
        </div>
        </div>

        <!--Password repetition Box-->
        <div id="PasswordWDH" . $style . ">
            <div id="login-box-field"><input name="PasswordWDHS" type="password"
            class="form-login" title="Password" placeholder="*****" size="30"
            maxlength="17" required/>
        </div>
        </div><br><br>

    <div class="capture" style="margin-top=10;"><br><br>

    ...

```

And as for the choice of faction, the user may choose to join only one of the two available, as they are mutually exclusive due to the type of checkbox used:

```
<div id="checkboxingbox"><input type="radio" value="Providentia" name="faction"
class="providentia" id="checkbox1" required><label
for="checkbox1"></div>

<div id="checkboxingbox"><input type="radio" value="drariva" name="faction"
class="drariva" id="checkbox2" required><label for="checkbox2"></div>

</div>

<!--Send-->
<br><br>
<div id=register>
<input type="submit" name="submit" value="" class="registerbutton" .$.style.">
</div>

...
```

Once you have confirmed the data entered and pressed the registration button, you are redirected to the 'registerclose.php' page, where the main checks are performed, including:

```
//This control takes place on every page of the site
session_start();
if(!isset($_SESSION["Username"]))
{
$conduct = 0;
}
else
{
$conduct = 1;
}

...

// We check that the user is not already logged in and mistakenly redirected to this page
if($conduct == 1){
    //A message and automatic redirection to the home page is displayed
    echo $havetodisconnect;
}
```

If the result is successful, a query will check that the name entered is not already in use by another user.

```
$name_control = $conn->prepare("Select * FROM accounts where
szUserID='".$Username."'");
$name_control-
>execute();
if($name_control-
>rowCount()==1){
    // If the chosen name is already in use, a warning message is printed out
    echo $namealreadyused;
    echo '<br><div class='loading_neutral'></div><br><br>
    <meta http-equiv='refresh' content='3; URL=register?language="'.$language.'">';
}
```

Next, it is checked that the password entered is correct and that the proposed name does not violate the specified parameters, in this case the type of characters used. Then a query is executed that inserts parameters, such as available currency and objects, into the database at the corresponding chosen name.

```
if($PasswordWDHMD5 == $PasswordMD5){
    if (ctype_alnum($Username)){
        $write = $conn->prepare("INSERT INTO accounts Values ('$nr', '$Username',
        '$PasswordMD5', '$faction', 1000, 0,0,0,0,0,0,0,0,0,
        0,'','normal','yes');");

        $write->execute();
    ...
}
```

Once registration has been completed, the user is automatically logged in, then, with each visit to each page of the site, queries are made to read, among other things, the various possessions and general information.

```

////// AUTO-LOGIN ////

////// USERNAME CHECK
$stmt = $conn->prepare("Select * FROM accounts where szUserID='".$Username."'
and szPasswd='".$PasswordMD5.'");
$stmt->execute();

while($row = $stmt->fetch(PDO::FETCH_ASSOC)){
    $Username = $row["szUserID"];
}

$stmt = $conn->prepare("Select * FROM accounts where szUserID='".$Username."'
and szPasswd='".$PasswordMD5.'");
$stmt->execute();

if(($stmt->rowCount()==1)){

    $_SESSION["Username"] = $Username;

    //////////// READING FACTION ////////////
    $stmt = $conn->prepare("Select faction FROM accounts where
szUserID='".$Username."' and szPasswd='".$PasswordMD5.'");
    $stmt->execute();

    while ($checkfaction = $stmt->fetch(PDO::FETCH_ASSOC)){
        $_SESSION['Faction'] = $checkfaction['faction'];
    }

    //////////// READING GOLD ////////////
    $stmt = $conn->prepare("Select gold FROM accounts where
szUserID='".$Username."' and szPasswd='".$PasswordMD5.'");
    $stmt->execute();

    while ($checkgold = $stmt->fetch(PDO::FETCH_ASSOC)){
        $_SESSION['Gold'] = $checkgold['gold'];
    }
    ...

    //////////// READING GUILD ////////////
    $stmt = $conn->prepare("Select guild FROM accounts where
szUserID='".$Username."' and szPasswd='".$PasswordMD5.'");
    $stmt->execute();

    while ($checkguild = $stmt->fetch(PDO::FETCH_ASSOC)){
        $_SESSION["Guild"] = $checkguild['guild'];
    }
    ...
}

```

As far as the login procedure is concerned, this can be performed by the user on any page of the site and is done via a user-friendly 'form'. Subsequently, one is directed to the 'login.php' page, where the accuracy of the data entered is checked and subsequently registered in the active session via PHP. Similarly to the registration phase, an additional check is performed, which leads to the display of a welcome page if the user logs in for the first time.

```
<form action="login?language=<?php echo $language; ?>" method="POST">
  <!--Login Box-->
  <div id="loginbox"<?php echo $style?>">
    <div id="login-box-field">
      <input name="UsernameS" class="form-login"
        title="Username" placeholder="Username" size="30" maxlength="17"
        required/>
    </div>
  </div>
  <!--Password Box-->
  <div id="passwordbox"<?php echo $style?>">
    <div id="login-box-field">
      <input name="PasswordS" type="password"
        class="form-login" title="Password" placeholder="*****" size="30" maxlength="17"
        required/>
    </div>
  </div>
  <!--Send-->
  <?php if (@$fp = fsockopen("localhost", 3306, $errno, $errstr, 1)){?>
    <input type="submit" name="test" value="" class="loginbutton"<?php echo
      $style?>">
  <?php }
  else{?>
    <div class="nlogin"></div>
  <?php } ?>
</form>

...

/// FIRST TIME CHECK ///
$stmt = $conn->prepare("Select first time FROM accounts where
szUserID='".$.$Username.'"");
$stmt->execute();
while($row = $stmt->fetch(PDO::FETCH_ASSOC)){
  $controlfirsttime = $row["firsttime"];
}
if($check once="yes"){
  echo "<script>window.location = 'intro?language='".$.$language.'"</script>";
}
```

## 4.3 - Guilds and Factions

Once the user has completed authentication in their session, they will notice the presence of a button in the top right-hand corner of the site's main bar. This button, with the aid of Javascript code, allows a modal window to emerge which presents a series of options represented by the different guilds available. These are positioned in succession, each being characterised by its own distinctive icon. By a single click on each one, its corresponding name will be made visible.

```
<!-- GUILD SYMBOL -->
<div class="guild"
onclick="document.getElementById('guilds').style.display='block'"></div>

...

<?php echo $guildtitle;?>

<!--ICONE GUILDS-->
<div class="warlock"
onclick="warlock();document.getElementById('exitguildout').style.display='none
';"></div>

<div class="paladin"
onclick="paladin();document.getElementById('exitguildout').style.display='none
';"></div>

...
```

If you are already a member of a guild, a button will be shown at the bottom of the window to leave it, as it will be impossible to try to join another one until you leave your own.

```
<?php if($_SESSION["Guild"] != '') {?>

    <input type="submit" class="exit" id="exitguild" name="exit" value=""
    data-toggle="tooltip" title="Exit your Guild">

<?php } ?>
```

In any case, after selecting a guild via one of the available icons, it is possible to display the list of players belonging to that guild. This functionality is implemented through the use of dynamic loading of AJAX, which allows a specific text field to be updated through the use of an associated button.

```
<div id="members button" class="members"
onclick="loadDocument(this.innerHTML)" data-toggle="tooltip" data-
placement="top" title="Guild Members"></div><br>

<script>

function loadDocument(e1) { var xhttp

= new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("zoneDynamic").innerHTML = this.responseText;
}
};

xhttp.open("GET", e1, true);
xhttp.send();
}

</script>
```

## 4.4 - Database

### 4.4.1 - accounts<sup>1</sup>

dwUserID	szUserID	szPasswd	faction	guild	object
User ID	Username of the user	Password of the user	Selected faction	Guild membership	One of several possessions available for purchase

### 4.4.2 - guild<sup>2</sup>

member	role
Identifier of the guild to which it belongs.	The role chosen during accession.

---

<sup>1</sup> A number of fields in the table have been omitted, as they are redundant for the purposes of merely viewing the structure.

<sup>2</sup> The title of the database relating to the members of a guild was chosen to be represented in a generic manner.



## Chapter 5 - How the project could be expanded

### 5.1 - Improvement of the game shop

New items could be added to the shop. This would include exclusive or time-limited items, which would encourage users to make immediate purchases. Or set up a search system to facilitate purchasing. It would also be useful to divide them into categories or create a search bar to allow users to quickly find what they are looking for.

Other ideas might be:

- Offering discounts and promotions would encourage users to visit the shop and make purchases. This could include special offers for users who make regular purchases or discounts on specific items.
- Providing detailed information on objects, such as their characteristics and statistics, can help users make informed purchasing decisions.
- Collecting user feedback on the shop can help to understand how to improve. It might be useful to offer a rating and review system to help users make better purchasing choices.
- Giving the possibility to customise objects would be an interesting idea. This could include the possibility of choosing the colour or pattern of the objects, or the possibility of adding customised names or messages.

## 5.2 - Improving the registration system

- The inclusion of tips or error messages could help users understand how to solve any problems during registration.
- Offering a welcome bonus to first-time users, such as the gift of an in-game item or additional virtual currency, could encourage users to complete the registration process and start playing.
- Authentication through social media would allow users to register or access the site using their accounts such as Facebook or Twitter. This would make the registration process quicker and easier, eliminating the need to manually enter all the required information.
- It should be important to provide support to new users during the registration process, e.g. through an online chat or support ticket system. In this way, users could get answers to their questions or solve any problems encountered during registration.

## 5.3 - Mailing list

One could turn the main page, which currently displays a simple welcome message, into a newsletter. The implementation would consist of a database with ID, Title, Date and Text fields in which to keep the news items. In the home page a 'form' could be inserted to write the news, only in case the user is also an administrator. Furthermore, via the PHP function 'mail()', a notice of the presence of new content would be sent to each registered user. Finally, it would be necessary to add a special field to the 'accounts' database and modify the registration 'form' to request the users' mail.

## Bibliography

- Jennifer Niederst Robbins, *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*, 5th Edition, O'Reilly Media, 2018
- Ramez A. Elmasri, Shamkant B. Navathe, *Database systems. Fundamentals and complements*, Pearson, 2017

## Sitography

- <https://www.w3.org/>
- <https://www.w3schools.com/html/>
- <https://www.w3schools.com/php/>
- <https://www.w3schools.com/js/>
- <https://www.html.it/guide/guida-ajax/>
- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- <https://api.jquery.com/>
- <https://learn.jquery.com/>
- <https://www.php.net/manual/en/function.mail.php>

*I thank my family and friends.*

*Special thanks to Paolo Bonventre for inspiration for the site's lore.*