

Diario di lavoro

Luogo	SAMT
Data	23.01.2019

Lavori svolti

Questa mattina la prima cosa che ho fatto è stata cercare dei template, inizialmente ho cercato quello per il login, essendo ciò che devo implementare oggi, e anche un eventuale template per lo shop.

Ho trovato entrambi, per il login ho deciso di utilizzare un template che permette di switchare nella stessa pagina login e registrazione.

Il template l'ho scaricato da questo sito: <https://codepen.io/Gibbu/pen/ZKYYZW>.

Dopo averlo preso ho controllato per il template dello shop, che poi riadatterò anche per le pagine dell'amministratore del gestore, ed ho optato per uno da shop trovato a questo link: <https://colorlib.com/wp/template/essence/>.

Inizialmente ho avuto alcuni problemi ad adattare il sito al mio template MVC, essendo che i link per i vari file css e JavaScript da caricare non venivano presi, e ho dovuto cambiare i percorsi inserendoli con la struttura MVC, come mostra l'immagine qui sotto.

```
<!-- Core Style CSS -->
<link rel="stylesheet" href="<?php echo URL ?>application/views/shop/css/core-style.css"
<link rel="stylesheet" href="<?php echo URL ?>application/views/shop/style.css">
```

Figura 1 percorsi per css

Quelli mostrati sopra sono 2 esempi di come ho dovuto cambiare i percorsi presenti nelle pagine che ho utilizzato, dovevo prendere la costante on la cartella base e arrivare fino alle cartelle da aprire.

Un'ulteriore applicazione che ho attuato è stata creare dei file "header" e "footer" sia per la pagina principale che per la pagina del login e registrazione, questo perché così è tutto più ordinato e ogni pagina contiene la sua parte e se dovessi utilizzare lo stesso header o footer per più pagine diventa molto più comodo averli in file separati.

Per aprire le pagine con la struttura che utilizzo devo caricare le pagine con delle classi e delle funzioni, ognuna delle quali fa qualcosa.

```
2 class Home
3 {
4     /**
5      * Function to open the buyer's login page
6      */
7     public function index()
8     {
9         require_once 'application/models/connection.php';
10        $connection = new Connection(servername: "localhost", username: "root", password: "", dbName: "ripetizioni");
11        $connection->sqlConnection();
12
13        require 'application/views/_templates/header.php';
14        require 'application/views/shop/index.php';
15        require 'application/views/_templates/footer.php';
16    }
```

Figura 2 apertura della pagina principale

Nell'immagine è mostrata la funzione "index" della classe "Home", che si occuperà di tutti i lavori sulle pagine dello shop, la funzione in questione stabilisce la connessione, l'ho preparata

ora anche se mi servirà più avanti, e dopodiché apre tutte le pagine utili per aprire l'index dello shop, ovvero header, footer e pagina principale.

La stessa cosa è fatta anche per la pagina di login e registrazione, ma questa pagina ho dovuta dividerla di più, perché in un caso deve esserci anche la possibilità di registrarsi altrimenti no.

Quindi ho creato 2 funzioni una che apre tutta la pagina completa e una che crea solo la parte del login, che mostra l'immagine che segue.

```

11 public function index()
12 {
13     require_once 'application/models/connection.php';
14     $connection = new Connection( servername: "localhost", username: "root", password: "", dbName: "ripetizioni");
15     $connection->sqlConnection();
16
17     require 'application/views/login_register/static/header.php';
18     require 'application/views/login_register/switch.php';
19     require 'application/views/login_register/login.php';
20     require 'application/views/login_register/register.php';
21     require 'application/views/login_register/static/footer.php';
22 }
23
24 /*
25  * Function to open the login page
26  */
27 public function openLogin($msn = "")
28 {
29     require_once 'application/models/connection.php';
30     $connection = new Connection( servername: "localhost", username: "root", password: "", dbName: "ripetizioni");
31     $connection->sqlConnection();
32     require 'application/views/login_register/static/header.php';
33     require 'application/views/login_register/login.php';
34     require 'application/views/login_register/static/footer.php';
35 }

```

Figura 3 apertura pagine login/registrazione

Per la pagina di login ho dovuto fare alcune modifiche al css, per renderla più adatta ai colori della pagina principale, quindi l'ho riadattata come mostrato dalle immagini che seguono.

LOGIN

Figura 4 pagina di login

Questa è la pagina di login, ho sostituito la parte verde originale con il colore dei bottoni della pagina principale e quando si va sopra essi si trasforma nel rosso di esse.

Figura 5 pagina registrazione

Quella mostrata sopra invece è la pagina di registrazione in cui ho cambiato i colori anche in questo caso e in più ho modificato il form aggiungendo i campi che mi servono.

Purtroppo la modifica delle informazioni per adattare la pagina di login mi ha preso più tempo del previsto, ma comunque siase dovevo farla da zero mi avrebbe preso forse ancora più tempo dato il fatto che avrei dovuto utilizzare da 0 bootstrap e css e non sono il massimo in questi 2 campi, di conseguenza non sono riuscito a completare i controlli dei vari campi e l'aggiunta del controllo Captcha, ma ho saputo che un mio compagno ha avuto dei problemi di permessi aggiungendolo, quindi attenderò maggiori informazioni da lui prima di inserirlo, per sapere se è effettivamente possibile o si ha bisogno di permessi speciali.

A causa di questo sono anche indietro con l'effettiva registrazione e il login dell'utente al sito, ma per fortuna ho già le funzioni quasi pronte devo solo completarle con i dati dei miei campi e quindi non mi ci vorrà molto a farlo.

Problemi riscontrati e soluzioni adottate

Non ho avuto dei veri e propri problemi ma mi ha impiegato del tempo capire come funzionavano i vari file css e javascript dei template per poterli utilizzare, e ho anche avuto dei leggeri problemi con bootstrap per sapere come funzionano alcune classi nella versione installata del template essendo diversa da quella usata precedentemente.

Punto della situazione rispetto alla pianificazione

Sono leggermente in ritardo rispetto al gantt essendo che non sono riuscito a finire la pagina di login e dovrò finire la prossima lezione.

Programma di massima per la prossima giornata di lavoro

Completare i controlli della pagina login e registrazione e implementare l'effettiva registrazione al database e il login alla pagina, sia quella del cliente che quella del gestore.