

Note su un sito di ricette

01/07/2024

Carlo Rosso

Table of Contents

1 Introduzione	1
2 Errori	1
2.1 I nostri errori	1
2.2 ReadMe	2
2.3 Fungo	2
2.4 TechCareGroup	3
3 Pagine Web	3
3.1 Reindirizzamento	5
4 Database	5
4.1 Entità	5
5 Backend	5
5.1 Api	5
5.2 Database	6
5.3 Autenticazione	7
5.4 Pagina	8
5.5 Componenti	8
5.6 Test	9

1 Introduzione

Non avendo passato il progetto di tecnologie web alla prima consegna, dobbiamo ricominciare il progetto da capo. Abbiamo deciso di fare un sito di ricette.

2 Errori

Cominciamo riportando gli errori che si sono verificati nella consegna precedente.

2.1 I nostri errori

2 canvas mancano le descrizioni dei canvas che devono descrivere il grafico inserendo le percentuali/ i valori che sono riportati nel grafico; oppure lo screen reader non deve poter vedere che è presente un canvas

tabella manca la descrizione della tabella

tabella manca l'intestazione della tabella

login il motore di ricerca deve poterlo vedere

utente user/user -> quindi username/password (non email)

login funziona solo tramite "localhost", vuol dire che non è semplice cambiare il dominio del sito -> molto male

404 torna a localhost

h1 > h2 > h3 non si può saltare da h1 a h3

titoli dal generale al particolare

link cliccato/non cliccato

altezza/larghezza si può usare solo %, em, rem. All'interno del css non si

può mai usare px, pt, cm, mm, in, pc, ...

layout a schermo intero definire una dimensione massima per gli elementi

(1024 / 1200px di larghezza al massimo, per gli schermi molto grandi)

print cambiare il font per la stampa, per esempio Serif (con grazie),

controllare che non ci siano parole sopra le altre

mobile torna in cima alla pagina

doppio scroll non deve esserci mai

scroll orizzontale non deve esserci mai

test colori passa, bisogna scriverlo nella relazione

logo di github alt="", perché dopo c'è scritto "github", quindi è inutile

descrivere l'immagine

logo di PennyWise alt="Paperon De Paperoni, logo di Penny Wise", perché la descrizione è importante

disorientamento link cliccabili ma poco distinguibili dal testo e testo

simile ai link

link circolari ovvero che ritornano alla pagina medesima, non devono esserci

nella navbar "About Us" si legge prima di "Registrati/ Accedi", non va bene

test manca la documentazione

js controllo dei campi dati

breadcrumb manca, ovvero il path in cui ci troviamo all'interno del sito

css ci sono dei punti e virgola o non ci sono

2.2 ReadMe

Errori del gruppo ReadMe:

dashboard l'amministratore dovrebbe avere la dashboard con le azioni più frequenti

login bisogna andare in cerca delle cose

login -> home non è possibile fare "pagina del libro" -> "login" -> "pagina del libro"

px non si possono usare

description e keywords devono essere dinamiche per le pagine pubbliche

pagina degli errori con emotional design, bene

catalogo le immagini dei libri hanno gli alt vuoti -> alt="", per indicare che è una scelta

nella pagine del libro copertina del libro con alt="", oppure con descrizione della copertina

link circolare non devono esserci (-1 punto)

test fatti bene

documentazione fatta bene

2.3 Fungo

contrasto tra link visitato e non per esempio si può cambiare invertire lo

sfondo con il testo, sottolineare o non o usare stili diversi (non solo sul colore), toglie 1 punto

funzionalità non funzionante -1 punto

30L hanno messo dei video e degli audio e li hanno resi accessibili

2.4 TechCareGroup

link circolari

bottoni come link vanno bene se si riferiscono a funzionalità

colore del link rispetto allo sfondo fa poco contrasto

css di validazione W3C segna degli errori

meta inesistenti per l'utente, se è una scelta va documentata

keyword manca la "s" quindi il SEO non funziona

201 viene tornato un codice in seguito ad una richiesta, non va bene

pagine di errore non le trova

print il logo occupa una pagina intera

mobile manca il torna in cima

form scroll orizzontale, molto male

img, alternativa testuale se non si vuole si mette alt=""

test con wave e total validator da documentare (cioè l'hanno scritto nella relazione, ma non hanno documentato l'esito)

salta contenuto mancante

controlli js sui form manca per esempio trim, per controllare che non ci sia

una stringa vuota o con soli spazi

link visitati/ non devono essere diversi, il discorso è diverso se fossero bottoni

3 Pagine Web

Di seguito l'elenco delle pagine web del sito:

Home Liste delle ricette, con ricerca per nome

- nome
- immagine
- tempo di preparazione
- difficoltà
- link alla pagina della ricetta

Ricetta Pagina di una ricetta, con:

- ingredienti
- procedimento
- tempo di preparazione
- difficoltà
- copiare giallo zafferano

Aggiungi ricetta Form per aggiungere una ricetta

- nome
- immagine
- ingredienti
- procedimento
- tempo di preparazione
- difficoltà
- link alla pagina della ricetta

Modifica ricetta Form per modificare una ricetta, con i campi come sopra

Pagina di login Form per il login

- username
- password
- link alla pagina di registrazione

Pagina di registrazione Form per la registrazione

- username
- password
- link alla pagina di login

Pagina di profilo Pagina con le informazioni dell'utente

Pagina delle proprie ricette Lista delle ricette dell'utente loggato

Pagine di errore

- 400
- 402
- 404
- ...

About us Pagina con le informazioni sul sito

Contatti Pagina con le informazioni di contatto

Pagina di test accedendo a questa pagina sono eseguiti tutti gli unit e gli integration test e sul sito viene visualizzato il risultato

3.1 Reindirizzamento

- Se l'utente non è loggato e cerca di accedere a una pagina che richiede il login:
 1. Viene salvato in localStorage l'URL a cui voleva accedere, oppure l'URL di partenza
 2. Viene reindirizzato alla pagina di login
 3. Dopo il login viene reindirizzato all'URL salvato
- Idem per la registrazione
- Se il campo nel localStorage è vuoto, viene reindirizzato alla home
- Si può pensare di usare questa tecnica anche per gestire altri reindirizzamenti

4 Database

Bisogna progettare il database. Bisogna pensare

- le entità che ci servono
- le relazioni tra le entità
- le operazioni che dobbiamo fare sul database

4.1 Entità

- Utente
- Ricetta
- Ingrediente
- Categoria (Dolce, ...)
- ...

5 Backend

Il backend è composto dall'insieme del codice php che gestisce le richieste dell'utente e interagisce con il database.

5.1 Api

- ogni API ritorna una pagina HTML, che sia
 - get
 - post
 - errore
- un form errato ritorna una pagina con il form già compilato e un messaggio di errore
- un form corretto reindirizza se è il caso

- per ogni API c'è una funzione o una classe che si occupa di gestire la richiesta e restituisce la pagina html.

5.2 Database

Le connessione al database usa un Singleton del tipo:

```
<?php
class Database {
    private static $instance = null;
    private $conn;
    private $host = 'localhost';
    private $user = 'username';
    private $pass = 'password';

    private function __construct() {
        Database::$conn = new mysqli($this->host, $this->user, $this->pass);
        // or something like that

        if (Database::$conn->connect_error) {
            die("Connection failed: " . Database::$conn->connect_error);
        }
    }

    public function getInstance() {
        if (Database::$instance == null) {
            Database::$instance = new Database();
        }
        return Database::$instance;
    }

    public query($sql_query, ...$params) {
        // check for sql injection
        // prepare statement
        // bind parameters
        // execute
        // return result
    }
}
```


5.3 Autenticazione

Viene usata una classe `Session` per gestire la sessione dell'utente. Questa classe si occupa di gestire:

- la sessione
- l'autenticazione
- i cookies
- l'autorizzazione

E quindi implementa i metodi necessari per fornire queste funzionalità.

```
<?php
class Session {
    public function start() {
        session_start();
    }

    public function set($key, $value) {
        $_SESSION[$key] = $value;
    }

    public function get($key) {
        return $_SESSION[$key];
    }

    public function destroy() {
        session_destroy();
    }

    public function is_logged_in() {
        return isset($_SESSION['user']);
    }

    public function login($user) {
        // check user credentials
        $_SESSION['user'] = $user;
    }

    public function logout() {
        unset($_SESSION['user']);
    }
}
```

```
public function get_user() {  
    return $_SESSION['user'];  
}  
  
...  
}
```

5.4 Pagina

La generazione delle pagine html è gestita da una classe Page che si occupa di:

- leggere i file html
- sostituire i placeholder con i dati
- restituire la pagina html

```
<?php  
class Page {  
    public function get_content($path) {  
        $content = file_get_contents("layout/" . $path);  
        $content = str_replace("{ header }", $this->get_header(), $content);  
    }  
  
    public function get_header() {  
        return file_get_contents("layout/header.html");  
    }  
  
    // ...  
}
```

5.5 Componenti

Da Page si estendono le classi che generano l'html per ogni componente della pagina. Queste classi interagiscono con il db e generano l'html per ogni componente. Di seguito qualche esempio:

- Recipe che implementa i metodi:
 - get_form()
 - get_item()
 - update(...\$params)
- Ingredient nella pagina ingredient.php
 - get_form(): non so bene come gestire questo
 - get_item()
 - update(...\$params)

- User nella pagina `user.php`
 - `get_login_form()`
 - `get_register_form()`
 - `get_update_form()`
 - `get_item()`: just to display user info
 - `update(...$params)`

5.6 Test

Fare una utility per eseguire i test. Questa utility estende `Page` e provvede:

- un metodo per eseguire una funzione di test e tornare il risultato sotto forma di html
- un metodo per contraffare ciascun tipo di dato, per esempio:
 - intero
 - stringa
 - array
 - ...
- un metodo per contraffare una richiesta http → vedere come si può chiamare un'api php da php