

Note su un sito di ricette

01/07/2024

Carlo Rosso

Table of Contents

1 Introduzione	1
2 Pagine Web	1
3 Database	2
3.1 Entità	2
4 Backend	2
4.1 Php	2

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 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

NB 1. Chiedere il motivo per cui tutti hanno fatto qualcosa di realistico, nel senso che ci hanno messo un indirizzo "reale".

3 Database

Bisogna progettare il database. Bisogna pensare

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

3.1 Entità

- Utente
- Ricetta
- Ingrediente

4 Backend

4.1 Php

- 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 un file php che gestisce la richiesta, ad esempio:
 - login.php
 - register.php
 - recipe.php
 - ingredient.php
- 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 function query($sql_query, ...$params) {
        // check for sql injection
        // prepare statement
        // bind parameters
        // execute
        // return result
    }
}
```

- ogni richiesta al database viene gestita mediante una funzione apposita per esempio:

```
<?php
function get_recipe($id) {
    $sql = "SELECT * FROM recipes WHERE id = ?";
    $result = Database::getInstance()->query($sql, $id);
    return $result;
}
```

- classe per la gestione delle sessioni/autenticazione/cookies/autorizzazione:

```
<?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'];  
    }  
  
    ...  
}
```

- classe per la generazione delle pagine 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");  
    }  
  
    // ...  
}
```