

AllGames

Progetto del corso Tecnologie Web

Anno 2020-21

Informazioni sul gruppo

Componenti	Barilla Gianmarco - 1149259
	Chiarello Federico - 1187598
	Furioso Patrick - 1187271
	Piacere Ivan - 1187524

Informazioni sul sito

Indirizzo sito web: <http://localhost/ipiaceres/php/home.php>

Email referente gruppo: ivan.piacere@studenti.unipd.it

Tabella 1: Utenti

Utente	Login	Password
Admin	admin	admin
Utente	user	user

Indice

1	Analisi	2
1.1	Analisi dell'utenza	2
1.1.1	Ricerca	2
2	Fase di Progettazione	3
2.1	Obiettivi	3
2.2	Classi di utenti:	3
2.3	Struttura del sito	3
2.3.1	Header	4
2.3.2	Breadcrumb	4
2.3.3	Contenuto	4
3	Fase di Realizzazione	5
4	Presentazione	6
4.1	Desktop	6
4.2	Mobile	6
4.3	Print	6
5	Comportamento	7
5.1	JavaScript	7
5.2	BackEnd	8
5.2.1	Php	8
5.2.2	Database	11
6	Fase di Test	12
6.0.1	Colori	12

Abstract

ALLGames è stato sviluppato per fornire agli appassionati di videogame un mezzo che permettesse loro di ricercare qualsiasi informazione utile riguardo ai giochi di loro interesse. Per fare ciò sul sito è presente una pagina che raccoglie al suo interno i vari giochi, visualizzati a partire dal più recente, nella quale si potrà scegliere se navigare liberamente utilizzando i vari filtri disponibili o se fare una ricerca più specifica grazie alla barra di ricerca. Inoltre ogni gioco presente nel sito oltre ad avere una sua scheda dedicata contenente le informazioni generali avrà una pagina apposita al cui interno si potrà trovare una recensione scritta da noi, creatori del sito, con una sezione dei commenti sottostante dove ogni utente potrà lasciare la sua opinione riguardo al gioco. Il sito ha poi un'altra pagina dedicata alle notizie generali sul mondo videoludico in modo da fornire all'utente la possibilità di restare aggiornato sugli ultimi videogame usciti, sugli ultimi eventi videoludichi organizzati e sulle novità riguardanti console e pc.

1 Analisi

1.1 Analisi dell'utenza

Il sito web si rivolge ad un pubblico prevalentemente composto da persone giovani, in genere ragazzi con un'età compresa tra i 12 e i 20 anni, ma include anche persone appartenenti a fasce di età molto più ampie, appassionati di videogiochi e del mondo che ne ruota attorno. Il target è potenzialmente molto ampio data la popolarità crescente dei videogiochi in tutte le fasce di età e di genere. Il pubblico è comunque prevalentemente composto da utenti che conoscono e navigano abitualmente il web. Per poter fornire un'esperienza piacevole anche all'utenza occasionale è necessario adottare un linguaggio semplice e facilmente intuibile e rispettare le convenzioni del web.

1.1.1 Ricerca

Dalla composizione del pubblico individuata, consegue che il sito debba mettere a disposizione degli utenti dei mezzi per poter effettuare una ricerca rapida di un preciso argomento, come ad esempio una barra di ricerca, ma al contempo fornire dei meccanismi che permettano una navigazione più generica.

2 Fase di Progettazione

Il sito si basa sul concetto di Responsive Web Design. Tutte le pagine sono fatte in modo tale da adattarsi ai diversi contesti di utilizzo, mantenendo una coerenza estetica e di leggibilità in ogni occasione. Questo è reso possibile grazie al concetto di media query che rende il layout della pagina fluido tramite punti di rottura e ridimensionamenti dinamici. Abbiamo deciso di utilizzare HTML5, rispettando la sintassi XML, garantendo in questo modo il supporto al passato e predisponendo un'ottimizzazione verso il futuro. In particolare abbiamo utilizzato dei tag input introdotti con HTML5, come ad esempio `data-list`, per supportare le fasi di ricerca degli utenti e di inserimento e modifica dei contenuti degli admin.

2.1 Obiettivi

Nella fase di progettazione e sviluppo del sito sono stati perseguiti i seguenti obiettivi:

- **Separazione tra struttura e presentazione:** la struttura del sito è separata dalla parte di presentazione;
- **Flessibilità:** il sito deve adattarsi alle differenti dimensioni dello schermo, in particolare risulta fondamentale garantire una buona fruizione da mobile, visto il numero crescente di accessi tramite dispositivi mobile;
- **Accessibilità:** il sito deve essere accessibile a tutte le categorie di utenti.

2.2 Classi di utenti:

Le possibili classi di utenti presenti nel sito sono tre:

- **Utente non registrato:** Ha accesso a tutti i contenuti presenti nel sito ma non ha la possibilità di commentare. Può comunque ottenere questa funzionalità registrandosi al sito tramite l'apposito form "Registrati";
- **Utente registrato:** E' in grado di commentare sotto le recensioni dei giochi ed ha accesso alla propria area riservata dove può modificare i propri dati come e-mail, password e immagine del profilo;
- **Admin:** E' l'utente con più privilegi in assoluto. Eredita tutte le funzionalità possedute da un normale utente registrato. Può aggiungere inoltre giochi e notizie al sito così come può anche modificare quelle già esistenti.

2.3 Struttura del sito

Il sito si sviluppa in ampiezza piuttosto che in profondità per migliorare la navigabilità e ridurre il senso di disorientamento. La struttura si basa su due livelli di profondità (fatta eccezione del lato amministratore che ne può avere massimo quattro) per fare in modo che l'utente sia in grado di esplorare il sito in maniera veloce ed intuitiva. Nonostante l'organizzazione gerarchica molto semplice l'utente ha anche a disposizione degli strumenti per velocizzare ulteriormente la ricerca di un'informazione specifica.

2.3.1 Header

All'interno dell'header, sulla sinistra, è presente il logo, il titolo ed il sottotitolo del sito. Sulla destra è presente una sezione che contiene l'immagine del profilo dell'utente (se presente), con due pulsanti login e registrati. Al di sotto di questi elementi è presente una barra orizzontale contenente il menu e la barra di ricerca. Il logo contiene un link che rimanda alla home del sito in quanto convenzione esterna presente in molti siti. L'immagine del profilo è un link con due comportamenti distinti a seconda se l'utente è loggato o meno.

- se l'utente non è loggato allora l'immagine del profilo sarà un'immagine di default e cliccandocisi sopra si accederà alla pagina di login
- se l'utente è loggato allora l'immagine del profilo corrisponderà alla immagine personalizzata dall'utente e cliccandocisi sopra si entrerà nell'area riservata dell'utente stesso.

Il menu disposto orizzontalmente contiene tre link (home, giochi, notizie) che portano alle sezioni principali del sito. La barra di ricerca posta sulla destra dà la possibilità all'utente di cercare un gioco specifico e mostra i suggerimenti in base a quanto l'utente sta digitando. Questa funzionalità fa sì che chi vuole avere più informazioni riguardo un gioco in particolare lo possa fare direttamente senza dover visitare più pagine del sito.

2.3.2 Breadcrumb

Il breadcrumb è presente in tutte le pagine del sito, sia mobile che desktop. E' posto sotto l'header e mostra la posizione dell'utente all'interno del sito. Si struttura come una serie di link partendo dalla pagina con gerarchica più alta scendendo fino alla pagina in cui ci si trova. La pagina corrente non è un link in modo tale da evitare la presenza di link circolari.

2.3.3 Contenuto

Pagina Home: Questa pagina contiene le ultime notizie e alcune funzionalità

3 Fase di Realizzazione

Riportiamo le parti realizzate dai componenti del gruppo:

- **Barilla Gianmarco:**

1. HTML di alcune pagine;
2. CSS di alcune parti del sito;
3. CSS per mobile e stampa;
4. Stesura della relazione.

- **Chiarello Federico:**

1. HTML di alcune pagine;
2. CSS di alcune parti del sito;
3. File JavaScript;
4. Stesura della relazione.

- **Furioso Patrick:**

1. HTML di alcune pagine;
2. CSS di alcune parti del sito;
3. Alcune funzioni JavaScript;
4. Stesura della relazione.

- **Piacere Ivan:**

1. HTML di alcune pagine;
2. Creazione Database e procedure;
3. Creazione classi e pagine PHP;
4. Stesura della relazione.

Inizialmente abbiamo creato un template contenente l'header e il footer, in modo che ogni componente potesse sviluppare le pagine a lui assegnate separatamente. Abbiamo lavorato fin dall'inizio sullo stesso file CSS, definendo inizialmente le regole comuni e aggiungendo in seguito le parti relative a pagine specifiche. In un secondo momento è iniziato lo sviluppo del Database, del PHP e di JavaScript. Le pagine create inizialmente come file HTML con del contenuto provvisorio, per poter definire l'aspetto della pagina, sono state modificate inserendo dei placeholder, utilizzate dagli script PHP per la creazione effettiva delle pagine. Infine abbiamo effettuato le attività di verifica, cercando inoltre di risolvere eventuali conflitti ed aumentare la qualità del codice prodotto. Nello sviluppo sono insorte delle difficoltà legate ai conflitti creati dallo sviluppo contemporaneo di vari file.

4 Presentazione

CSS...

4.1 Desktop

4.2 Mobile

4.3 Print

5 Comportamento

Il comportamento del sito è gestito lato server da PHP e lato client da JavaScript.

5.1 JavaScript

Abbiamo usato un unico file JavaScript nominato `checkForm.js`, contenente principalmente funzioni che svolgono controlli sull'input delle form. Di seguito riportiamo le funzioni con una breve spiegazione:

- **`validateForm()`**: esegue un controllo lato client dell'input della form. Viene chiamata in tutte le pagine che contengono delle form per l'inserimento o la modifica dei dati lato admin e nelle pagine di login e registrazione lato utente. Per facilitare eventuali aggiunte di campi input nel sito, la funzione si appoggia all'array `dettagliForm` contenente come chiavi gli id dei campi input a cui vengono associati array contenenti le espressioni regolari per la verifica dell'input e i corrispondenti messaggi d'errore. La funzione procede chiamando `validateInput(input)` per ogni elemento di `dettagliForm` trovato nella pagina di invocazione, prosegue chiamando `checkChecked()` ed infine ritorna un valore booleano in base all'esito dei controlli effettuati;
- **`validateInput(input)`**: effettua il controllo del campo input. Nel caso sia già presente un messaggio (di errore o di conferma del corretto inserimento), lo rimuove. Gestisce in seguito i casi limite, non gestibili tramite l'array di supporto: campi input di tipo file o data, il campo `repeatedpassword` e i campi opzionali. Effettua quindi il controllo dei restanti campi input sfruttando le espressioni regolari presenti in `dettagliForm` e procede ritornando il valore ricevuto dall'invocazione di `showMessage(input, stato)`;
- **`showMessage(input, stato)`**: crea un messaggio che comunica all'utente se l'input inserito è corretto o meno. Abbiamo deciso di fornire 2 meccanismi di feedback all'utente, presentando sia un messaggio testuale che una colorazione del bordo del campo input, in questo modo possiamo facilitare la compilazione delle form a tutte le classi di utenti.
- **`checkPassword()`**: controlla che gli input in `password` e in `repeatedpassword` siano uguali e invoca opportunamente `showMessage(input, stato)`;
- **`checkChecked()`**: effettua un controllo dei campi checkbox e radio sfruttando l'array di supporto `spuntabili` in modo analogo a `validateForm()`. Crea un array contenente coppie chiave-valore nella forma: id del campo input, valore booleano (che indica se il rispettivo campo input è stato spuntato). Tale array viene poi fornito come parametro a `showMessageCheckbox(sezioni)`;
- **`showMessageCheckbox(sezioni)`**: opera in modo analogo a `showMessage(input, stato)`, creando però dei messaggi che comunicano all'utente se i campi input (di tipo checkbox o radio) sono stati spuntati o meno;

- **handleClick():** fa comparire/scompare il campo 'Gioco trattato' se viene selezionata/deselezionata la categoria Giochi' in Tipologia nella form per l'inserimento di una notizia;
- **removeNoJs():** viene invocata al completamento del caricamento della pagina, rimuove la classe 'no-js' dal tag `<html>`. Tale classe è stata utilizzata per la definizione di alcune regole `css` che gestiscono la casistica in cui JavaScript è disattivato. Ad esempio nel caso del menù principale nella versione mobile, con JS disattivato, viene rimosso il pulsante e viene mostrata la tendina aperta;
- **checkNotEmpty(id):** controlla che il campo `input` con `id = 'id'` non sia vuoto o composto da spazi;
- **responsiveMenu():** gestisce l'apertura e chiusura del menù a tendina nella versione mobile. E' associata ad un evento `onclick`, scelto perchè gestisce anche l'evento `ontouch` per dispositivi mobile. Il pulsante è situato nel top-right della pagina, dove è più facilmente raggiungibile da un utente mobile;
- **preparaFiltri():** viene chiamata al completamento del caricamento della pagina Giochi. Gestisce l'evento `onclick` dei button tramite event delegation nel `div` con classe 'container'. All'evento `onclick` di un bottone dei filtri di ricerca lascia aperta la tendina contenente i filtri, mentre ad un secondo evento `onclick` all'interno del 'container' la richiude;
- **checkAnni():** effettua un controllo sull'intervallo degli anni di uscita dei videogiochi selezionato nella pagina giochi. Inserisce un messaggio d'errore se l'intervallo è inammissibile (es: da 2020 a 2016) e blocca il submit.

5.2 BackEnd

Il backend del server è stato sviluppato con script PHP e database relazionale MariaDB.

5.2.1 Php

Gli script che costituiscono il backend si possono suddividere in 3 macro categorie in base alla loro funzione:

1. modellazione dei dati.

- `comment.php`
- `game.php`
- `image.php`
- `news.php`
- `user.php`
- `review.php`

2. interazione con il database;

- dbConnection.php

3. creazione e presentazione delle pagine html;

- edit_gioco.php
- edit_notizia.php
- edit_profilo.php
- form_gioco.php
- form_notizia.php
- giochi.php
- gioco_notizie.php
- gioco_recensione.php
- gioco_scheda.php
- home.php
- lista_utenti.php
- login.php
- notizia.php
- notizie.php
- profilo.php
- registrati.php

4. funzionalità condivise;

- replacer.php

Modellazione dei dati I dati gestiti dal nostro sito sono per la maggior parte classificabili in 5 tipi, illustrati singolarmente nei paragrafi successivi. Per favorire l'ordine all'interno del codice e nella comunicazione dei dati, ogni tipo di viene associato ad una classe.

Gioco Contiene le informazioni riguardanti un gioco e la recensione che lo riguarda.

Notizia Contiene le informazioni riguardanti una notizia.

Utente Contiene le informazioni riguardanti un utente salvato nel sistema, che può essere di tipo amministratore o utente semplice.

Immagine Contiene il percorso di un'immagine salvata nel server e il testo alternativo che la accompagna.

Commento Contiene le informazioni un commento rilasciato da un utente riguardo ad un gioco.

Recensione Contiene il contenuto di una recensione e una indicazione del gioco a cui è riferita, dell'utente che l'ha scritta e della data e ora dell'ultima modifica.

Interazione con il database Il file `dbConnection.php` presenta la classe `DBAccess` che permette di aprire una connessione verso il database e di effettuare richieste ad esso. Per ognuno dei tipi di dato presentati nel paragrafo precedente sono rese disponibili le seguenti funzionalità:

- ottenimento della lista dei dati;
- aggiunta di un nuovo dato;
- modifica di un dato già presente;
- rimozione di un dato già presente.

In aggiunta sono presenti delle specializzazioni di queste operazioni e alcune ulteriori funzionalità per l'accesso alle tabelle del database non direttamente rappresentate da una classe, ovvero `PrequelSequel`, `GamesConsoles` e `GamesGenres`.

Pulizia input Ogni dato, prima di essere inserito nel database viene validato tramite la funzione `textttmysql_real_escape_string`. Essa fa l'escape di tutti i caratteri speciali presenti nella stringa, disattivando così eventuali attacchi di tipo `sql injection`. La documentazione di questa funzione è fornita alla pagina <https://www.php.net/manual/en/mysqli.real-escape-string.php>;

Creazione e presentazione delle pagine html Gli script inclusi in questa categoria costituiscono la vista del nostro sito, ovvero la parte che interagisce con l'utente. Ognuno degli script segue, nella sua esecuzione, lo schema seguente:

1. apertura connessione col database;
2. costruzione template html specifico della pagina;
3. controllo del tipo di utente connesso (utente non autenticato, utente semplice, amministratore);
4. raccoglimento valori di input;
5. costruzione del contenuto della pagina in base a:
 - input
 - tipo di utente
 - informazioni contenute nel database
6. assemblamento del contenuto col template;
7. inserimento pagina costruita all'interno di un template di più alto livello contenente header e footer;
8. output della pagina all'utente.

Form Anche le pagine contenenti form si basano sulla struttura appena delineata. In particolare in esse il punto 4 è costituito dai seguenti passi:

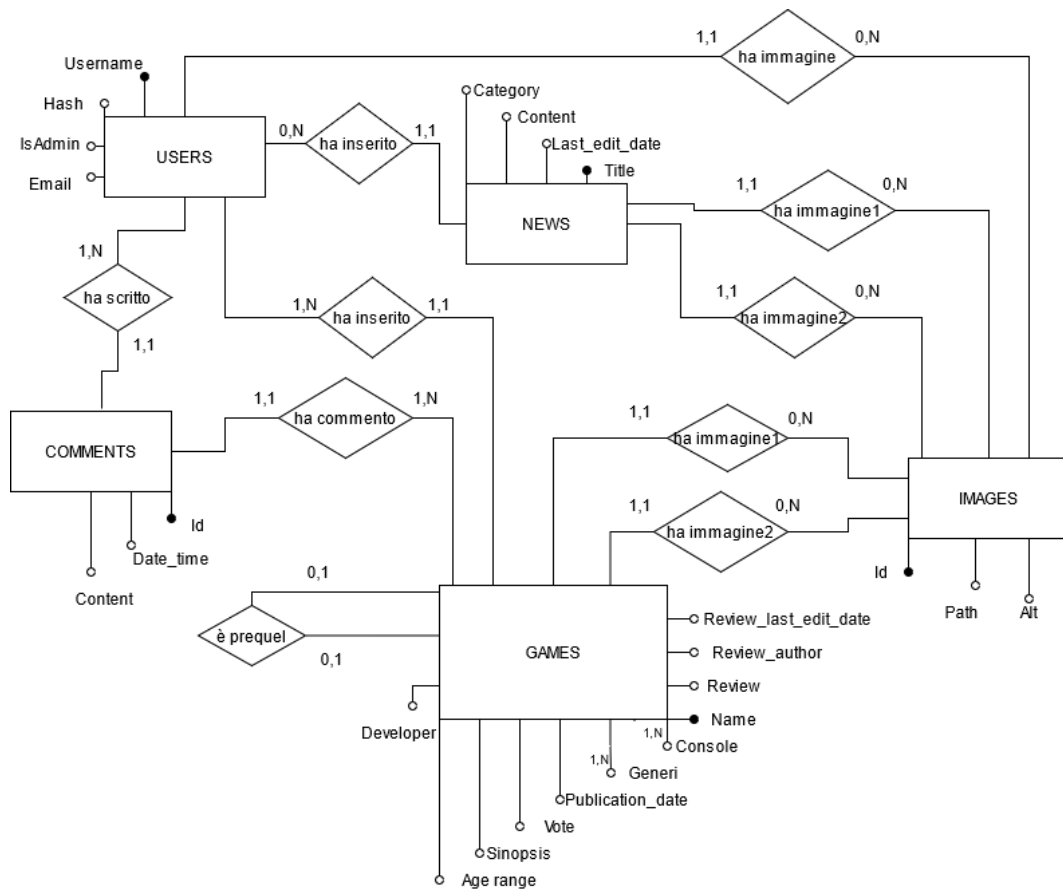
1. raccoglimento valori di input e loro sanificazione
2. validazione di ogni valore inserito e selezione degli eventuali messaggi di errore da mostrare

Pulizia e validazione input Ogni valore di input ricevuto è sottoposto ad una sanificazione prima di essere utilizzato. La funzione adibita a questo compito è `getSafeInput` definita in `replacer.php`.

La validazione è eseguita tramite la funzione `validateValue` e i messaggi di errore sono ottenuti tramite la funzione `getValidationError`. Anche queste due funzioni si trovano in `replacer.php`.

Funzionalità condivise Il file `replacer.php` contiene funzionalità e dati utilizzati da molti script diversi.

5.2.2 Database



6 Fase di Test

Tutte le pagine e il relativo codice sono stati sottoposti a una validazione attraverso i seguenti strumenti:

- **W3C Validator:** per la validazione dei file HTML;
- **W3C CSS Validator:** per la validazione dei file CSS;
- **Contrast Analyzer:** per verificare che i colori utilizzati presentassero un adeguato contrasto, abbiamo usato il sito <https://webaim.org/resources/contrastchecker/>.

Il sito è stato testato sui principali browser (Google Chrome, Mozilla Firefox e Microsoft Edge), sfruttando inoltre l'estensione Developer Toolbar per poter effettuare dei test più approfonditi riguardanti il comportamento del sito in situazioni limite, quali ad esempio la disattivazione di Javascript o il mancato caricamento delle immagini.

6.0.1 Colori

I colori principali del sito sono il blu, il rosso, il nero ed il bianco. Come background, abbiamo usato un'immagine che raggruppa una serie di icone dei videogame. Di seguito viene riportata una tabella contenente le combinazioni di colori usati nel sito e il relativo rapporto tra colore di sfondo e di testo. Tutti i risultati dei rapporti superano il test WCAG AAA, infatti il livello AAA richiede un rapporto di contrasto di almeno 7:1 per il testo normale e 4.5:1 per il testo in grassetto.

Tabella 2: Valutazione contrasti

Colore sfondo	Colore testo	Rapporto
#FFFFFF	#600000	14.02:1
#FFFFFF	#1B365D	12.12:1
#FFFFFF	#000000	21:1
#FFFFFF	#676767	5.65:1
#74C243	#000000	9.54:1
#000000	#3E8E41	5.15:1