

Tesina per il corso di linguaggi per il web (2018-2019)

Sede di Latina.

Cosmo Vellucci 1760067.

July 10, 2020

Contents

| | |
|---|-----------|
| 1 Capitolo 1 - Introduzione e descrizione dei dati | 3 |
| 1.1 Che cosa è, e cosa fa l'applicazione web, a che serve? | 3 |
| 1.2 Quali tipologie di utenti serve? Per ogni tipologia quali funzionalità vengono fornite? | 3 |
| 1.3 Specifiche del sito. | 4 |
| 1.4 Strutture dati utilizzate. | 6 |
| 1.4.1 Base di dati. | 6 |
| 1.4.2 XML. | 6 |
| 1.5 Descrizione delle funzionalità. | 10 |
| 2 Capitolo 2 - Manuale utente. | 11 |
| 2.1 Accedere al sito (pagina principale). | 11 |
| 2.2 Login e registrazione. | 12 |
| 2.3 Pagina delle FAQ. | 13 |
| 2.4 Negozio. | 13 |
| 2.5 Logout. | 13 |
| 2.6 Libreria. | 14 |
| 2.7 Impostazioni. | 14 |
| 2.8 Carrello. | 15 |
| 2.9 Inserimento di un videogioco. | 15 |
| 2.10 Ripristino del database. | 16 |
| 2.11 Gestione utenti. | 16 |
| 2.12 Pagina 404. | 17 |
| 2.13 Pagina di un gioco. | 17 |
| 3 Capitolo 3 - Programmazione. | 18 |
| 3.1 Panoramica della cartella. | 18 |
| 3.2 Scripting di programmazione. | 19 |
| 3.2.1 .htaccess | 19 |
| 3.2.2 toolbar.php | 19 |
| 3.2.3 index.php | 19 |
| 3.2.4 Login.php | 20 |
| 3.2.5 Logout.php | 22 |
| 3.2.6 cancDB.php | 22 |
| 3.2.7 Carrello.php | 23 |
| 3.2.8 Connessione.php | 25 |
| 3.2.9 DatiDB.php | 25 |
| 3.2.10 Install.php | 25 |
| 3.2.11 Negozio.php | 26 |
| 3.2.12 GamePage.php | 27 |
| 3.2.13 AggiuntaCarrello.php | 28 |
| 3.2.14 SezioneCommenti.php | 29 |
| 3.2.15 Libreria.php | 32 |
| 3.2.16 ImpostazioniUtente.php | 32 |
| 3.2.17 InserimentoGioco.php | 34 |

| | |
|---------------------------------|-----------|
| 3.2.18 InserimentoGiocoForm.php | 35 |
| 4 Capitolo 4 - Elenco. | 37 |

1 Capitolo 1 - Introduzione e descrizione dei dati

1.1 Che cosa è, e cosa fa l'applicazione web, a che serve?

Millennium games è una piattaforma online di videogiochi digitali, simile alla già esistente Steam. L'applicazione web ha lo scopo di creare non solo una libreria digitale contentente i videogames acquistati per ogni singolo utente, ma anche di creare una community attraverso commenti ed opinioni sotto la pagina di ogni specifico titolo.

1.2 Quali tipologie di utenti serve? Per ogni tipologia quali funzionalità vengono fornite?

Ci sono cinque tipologie di utenti:

- Utente visitatore: accede al catalogo di videogiochi e i link informativi. Può registrarsi o effettuare il login.
- Utente admin: cambia i ruoli dell'utente e gestisce i fondi (valuta richieste di credito degli utenti).
- Utente gestore: è capace di cancellare commenti e bannare utenti e di inserire i suoi videogiochi nel catalogo.
- Utente bannato: non ha il permesso di commentare nell'apposita sezione.
- Utente standard: ha il permesso di commentare nell'apposita sezione.

Un utente può appartenere contemporaneamente ad una o a più di queste ultime quattro categorie.

Ogni utente in generale e al di fuori delle sue appartenenze a certe categorie, può comprare videogiochi (aggiungendoli così alla propria libreria), commentare nell'apposita sezione commenti, accumulare esperienza tramite acquisti e commenti, caricare credito sul proprio portafoglio dalla propria carta di credito, visualizzare le faq.

Nessuno degli utenti può rimuovere dalla propria libreria un videogioco (in quanto non avrebbe senso).

1.3 Specifiche del sito.

Il sito gestisce un negozio online di videogiochi in formato digitale. Quando si accede al sito, l'utente si troverà davanti alla mainpage. Questa, come tutte le altre pagine del sito (a eccezione della pagina di errore 404), presenta una toolbar nella parte alta della pagina, con la quale si possono raggiungere tutte le altre pagine.

Tuttavia, l'utente dovrà effettuare l'accesso per poter svolgere le sue attività (descritte in seguito).

Sia prima che dopo aver effettuato l'accesso, l'utente può visitare le pagine *mainpage* (cliccando sul logo del sito), *negozi*, *login*, e *F.A.Q.*

Ogni utente può accedere al sito dalla apposita pagina di login (se possiede già un account), e se non possiede già un proprio account, può crearne uno per la prima volta, sempre nella pagina in questione.

La pagina del negozio mostra la lista di giochi che si possono comprare. Cliccando sul titolo si può accedere alla pagina contenente le informazioni del gioco stesso.

La pagina di F.A.Q presenta alcune informazioni del sito (persone che vi lavorano, luogo, email) e le domande più frequenti e specifiche che alcuni utenti possono porsi.

Nel caso in cui venga richiesto dall'utente un oggetto o una pagina mancante o inesistente, si passa a una pagina di errore 404, nella quale si può solo tornare alla pagina precedentemente visitata, attraverso l'apposito bottone (non è infatti presente la toolbar precedentemente nominata).

L'utente può accedere alla propria libreria tramite l'apposita pagina (che appare sulla toolbar dopo aver effettuato l'accesso), dove può vedere la sua collezione (personale) di videogiochi acquistati.

Ogni videogioco può essere acquistato una sola volta, e non può essere cancellato dalla propria libreria.

L'utente può accedere alle proprie impostazioni tramite l'apposita pagina (che appare sulla toolbar dopo aver effettuato l'accesso), dove può modificare l'immagine di profilo, password e rimuovere o aggiungere una e una sola carta di credito.

L'utente può accedere al proprio carrello tramite l'apposita pagina (che appare sulla toolbar dopo aver effettuato l'accesso), che mostra gli acquisti attuali. In questa pagina è possibile rimuovere certi prodotti dal carrello oppure svuotarlo del tutto.

Per quanto riguarda la pagina del negozio, all'interno di essa sono elencati i giochi disponibili alla vendita, e (per ognuno di essi) cliccando sul titolo si accede alla pagina del gioco in questione.

Tale pagina del gioco presenta tutte le informazioni, e presenta anche una sezione commenti (relativa sempre e solo alla pagina del videogioco che si sta visualizzando), dove gli utenti che hanno effettuato l'accesso possono commentare (ad eccezione di coloro che sono stati bannati).

La sezione commenti è quella standard dei forum: ci sono dei post (1 post coincide a livello di logica con 1 commento), ordinati da quelli meno recenti a quelli più recenti, presentano alcune informazioni su chi ha postato (nome, immagine di profilo, livello) e il testo del commento stesso.

Nel caso in cui il commento sia stato marcato come recensione, oltre al testo viene aggiunto un pollice in su o un pollice in giù, per far capire se la recensione sia rispettivamente positiva o negativa.

Nel caso l'utente sia un moderatore, può cancellare il testo del post di un utente e, eventualmente, bannare l'utente stesso.

Si noti che i commenti non vengono letteralmente "cancellati", ma in base all'attributo specificato nel file xml semplicemente li si mostra o meno.

Nel caso l'utente sia anche un produttore di videogiochi, sulla toolbar apparirà un pannello per l'inserimento di un videogioco.

Nella pagina apposita, lo sviluppatore potrà inserire i dati del videogioco in questione (titolo, genere, prezzo, disponibilità), oltre a caricare da una a quattro immagini di screenshot che descrivono il video-

gioco.

Una volta compilato il form apposito, il gioco appare nel negozio del sito.

In ogni libreria associata al nome utente, sono presenti gli ID dei videogiochi acquistati dall'utente. Una libreria di un nuovo utente iscritto al sito può essere anche vuota.

Ogni videogioco ha un suo numero intero identificativo, che deve essere unico. Ha poi un titolo, un genere, da 0 a 4 immagini di screenshot che lo descrivono all'utente che visita la pagina del gioco stesso, un prezzo in euro e una descrizione.

Il livello dell'utente dipende da un valore numerico intero chiamato esperienza, che segue questi parametri:

- Da 0 a 99: livello 0.
- Da 100 a 199: livello 1.
- Da 200 a 299: livello 2.
- Da 300 a 399: livello 3.
- Da 400 a 499: livello 4.
- Da 500 a 599: livello 5.
- Da 600 a 699: livello 6.
- Da 700 a 799: livello 7.
- Da 800 a 899: livello 8.
- Da 900 a 999: livello 9.
- 1000 e oltre: livello 10.

Ogni utente inizialmente registrato sul sito presenta un valore di esperienza pari a 0.

Ogni commento aggiunge 50 di esperienza.

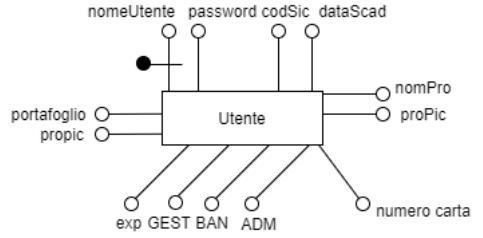
Ogni acquisto aggiunge esperienza pari al prezzo del videogioco per 2,5.

1.4 Strutture dati utilizzate.

1.4.1 Base di dati.

La base di dati utilizzata coincide con l'entità a fianco.

In teoria, la carta di credito andrebbe rappresentata a parte, ma per semplicità di progetto si ha optato per incorporarla in un'unica tabella. Tutto il resto viene rappresentato in xml.



Si noti che per motivi di rappresentazione La chiave primaria è NomeUtente, mentre numeroCarta è un valore che deve essere unico.

In particolare, gli unici valori che possono essere nulli sono quelli della carta di credito (e in questo caso lo sono sempre e solo tutti assieme).

Essi assumono valori nulli solo se il proprietario dell'account rimuove la carta di credito, oppure se lo ha appena creato.

GEST, BAN, ADM indicano l'appartenezza dell'utente a una o più categorie (rispettivamente: gestore, bannato, admin). Possono assumere solo il valore N o Y.

Il portafoglio contiene i crediti dell'utente.

Ogni utente è identificato univocamente dal nomeUtente e accede al sito tramite password. L'exp è quel valore di fiducia di cui si ha parlato prima.

1.4.2 XML.

Le strutture dati rappresentate in .xml sono invece le seguenti:

Per quanto riguarda i **giochi**:

- idGioco (intero che specifica l'id del gioco).
- nomeGioco (titolo del gioco).
- genere (genere del gioco).
- descrizione (descrizione del gioco).
- prezzo (prezzo in euro del gioco).
- da 0 a 4 immagini (screnshoot del videogioco).
- dispo (data di disponibilità del gioco).

Esempio:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <VIDEOGIOCHI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="SchemaGiochi.xsd">
4     <VideoGioco>
5         <idGioco>31</idGioco>
6         <nomeGioco>The Witcher 3</nomeGioco>
7         <genere>GDR</genere>
8         <descrizione>Ripercorri la via del lupo nei panni di Geralt di Rivia, combattendo contro la Caccia Selvaggia ispirata ai miti polacchi.</descrizione>
9         <prezzo>30.50</prezzo>
10        <img>sc1.jpg</img>
11        <img>sc2.jpg</img>
12        <img>sc3.jpg</img>
13        <dispo>2016-10-25</dispo>
14    </VideoGioco>
15
16    <VideoGioco>
17        <idGioco>2</idGioco>
18        <nomeGioco>Spyro Reignited Trilogy</nomeGioco>
19        <genere>Platformer</genere>
20        <descrizione>La trilogia delle avventure del draghetto Spyro ora riunite in un unico remake.</descrizione>
21        <prezzo>20.00</prezzo>
22        <img>sc1.jpg</img>
23        <img>sc2.jpg</img>
24        <img>sc3.jpg</img>
25        <img>sc4.jpg</img>
26        <dispo>2010-07-10</dispo>
27    </VideoGioco>
28
29    <VideoGioco>
30        <idGioco>25</idGioco>
31        <nomeGioco>Medieval City</nomeGioco>
32        <genere>Gestionale</genere>
33        <descrizione>Come si comportava il re di un feudo nel medioevo? In questo gioco puoi provarlo.</descrizione>
34        <prezzo>5.00</prezzo>
35        <dispo>2018-05-07</dispo>
36    </VideoGioco>
37
38    <VideoGioco>
39        <idGioco>77</idGioco>
40        <nomeGioco>Metro Exodus</nomeGioco>
41        <genere>FPS</genere>
42        <descrizione>Basato sugli omonimi libri di Dmitrij Gluchovskij, percorri la strada di Artyom nel trovare una zona sicura dal mondo radioattivo in cui nacque.</descrizione>
43        <prezzo>50.00</prezzo>
44        <img>sc1.jpg</img>
45        <img>sc2.jpg</img>
46        <img>sc3.jpg</img>
47        <img>sc4.jpg</img>
48        <dispo>2033-01-02</dispo>
49    </VideoGioco>
50 </VIDEOGIOCHI>
```

Schema:

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <xsschema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3
4       <xselement name="VIDEOGIOCHI">
5           <xsccomplexType>
6               <xsssequence>
7                   <xselement ref="VideoGioco" maxOccurs="unbounded"/>
8               </xsssequence>
9           </xsccomplexType>
10      </xselement>
11
12      <xselement name="VideoGioco">
13          <xsccomplexType>
14              <xsssequence>
15                  <xselement name="idGioco" type="xs:integer"/>
16                  <xselement name="nomeGioco" type="xs:string"/>
17                  <xselement name="genere" type="xs:string"/>
18                  <xselement name="descrizione" type="xs:string"/>
19                  <xselement name="prezzo" type="xs:decimal"/>
20                  <xselement name="img" type="xs:string" maxOccurs="4" minOccurs="0"/>
21                  <xselement name="dispo" type="xs:date"/>
22              </xsssequence>
23          </xsccomplexType>
24      </xselement>
25
26  </xsschema>
```

Ogni utente possiede una e una sola **libreria**, che contiene una e una sola copia dei giochi acquistati dagli utenti. Quindi, per quanto riguarda le **librerie**:

- id (identificativo del gioco posseduto).

L'associazione utente-libreria viene effettuata tramite attributo XML.

Esempio:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <LIBRERIE xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:noNamespaceSchemaLocation="SchemaLibrerie.xsd">
4      <Libreria utenteAssociato="CosmoVellucci">
5          <id>31</id>
6          <id>2</id>
7      </Libreria>
8
9      <Libreria utenteAssociato="Manigoldo">
10         <id>2</id>
11     </Libreria>
12
13     <Libreria utenteAssociato="Arturo">
14         <id>25</id>
15         <id>77</id>
16     </Libreria>
17
18     <Libreria utenteAssociato="Bramante">
19     </Libreria>
20 </LIBRERIE>
```

Schema:

```
C:\>xampp>htdocs>Sito> SchemaLibrerie.xsd
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3
4      <xs:element name="LIBRERIE">
5          <xs:complexType>
6              <xs:sequence>
7                  <xs:element ref="Libreria" maxOccurs="unbounded"/>
8              </xs:sequence>
9          </xs:complexType>
10     </xs:element>
11
12     <xs:element name="Libreria">
13         <xs:complexType>
14             <xs:sequence>
15                 <xs:element name="id" type="xs:integer" maxOccurs="unbounded" minOccurs="0"/>
16             </xs:sequence>
17             <xs:attribute name="utenteAssociato" type="xs:string"/>
18         </xs:complexType>
19     </xs:element>
20
21 </xs:schema>
```

Per quanto riguarda la sezione commenti di ogni **videogioco**:

- utente: nome dell'utente che ha postato il commento.
- testo: il testo del commento.
- recensione: specifica la presenza di un pollice in su o in giu. La recensione è opzionale.

Una sezione commenti di un certo videogioco potrebbe non ancora avere commenti. Esempio e schema:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <COMMENTI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="SchemaCommenti.xsd">
3
4      <sezioneCommentiGioco idGioco="31">
5          <commento commentoCancellato="NO">
6              <utente>CosmoVellucci</utente>
7              <texto>Questo gioco e' molto bello!</texto>
8              <recensione>P</recensione>
9          </commento>
10         <commento commentoCancellato="NO">
11             <utente>Arturo</utente>
12             <texto>Concordo pienamente con te Cosmo. E' eccezionale.</texto>
13             <recensione></recensione>
14         </commento>
15         <commento commentoCancellato="SI">
16             <utente>Manigoldo</utente>
17             <texto>QUESTO COMMENTO E' CANCELLATO.</texto>
18             <recensione></recensione>
19         </commento>
20         <commento commentoCancellato="NO">
21             <utente>CosmoVellucci</utente>
22             <texto>Ti ringrazio, caro Arturo.</texto>
23             <recensione></recensione>
24         </commento>
25     </sezioneCommentiGioco>
26
27     <sezioneCommentiGioco idGioco="2">
28     </sezioneCommentiGioco>
29
30     <sezioneCommentiGioco idGioco="25">
31     </sezioneCommentiGioco>
32
33     <sezioneCommentiGioco idGioco="77">
34     </sezioneCommentiGioco>
35
36 </COMMENTI>
```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsschema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3
4      <xselement name="COMMENTI">
5          <xsccomplexType>
6              <xsssequence>
7                  <xselement ref="sezioneCommentiGioco" maxOccurs="unbounded"/>
8              </xsssequence>
9          </xsccomplexType>
10     </xselement>
11
12
13     <xselement name="sezioneCommentiGioco">
14         <xsccomplexType>
15             <xsssequence>
16                 <xselement name="commento" maxOccurs="unbounded" minOccurs="0" type="TipoCommento"/>
17             </xsssequence>
18             <xseattribute name="idGioco" type="xs:integer"/>
19         </xsccomplexType>
20     </xselement>
21
22     <xsccomplexType name="TipoCommento">
23         <xsssequence>
24             <xselement name="utente" type="xs:string"/>
25             <xselement name="testo" type="xs:string"/>
26             <xselement name="recensione" type="xs:string" minOccurs="0"/>
27         </xsssequence>
28         <xseattribute name="commentoCancellato" type="xs:string"/>
29     </xsccomplexType>
30
31
32 </xsschema>
```

1.5 Descrizione delle funzionalità.

Esempio 1: commento.

Supponiamo che un utente voglia commentare nell'apposita sezione riguardo un certo videogioco.

Ogni utente non bannato è libero di farlo prima loggandosi al sito e poi andando nella pagina di un certo videogioco, dove troverà un text-box dove potrà inserire tale commento.

Esempio 2: moderazione e ban.

Supponiamo che l'utente *CosmoVellucci* (moderatore) veda un messaggio con una parolaccia postato nella pagina dei commenti, postato dall'utente *Manigoldo*.

In tal caso l'utente CosmoVellucci può decidere non solo di eliminare il messaggio, ma anche di bannare l'utente Manigoldo, impedendogli così di comunicare ulteriormente.

Questa cosa avviene nella sezione commenti stessa ed è ovviamente effettuabile solo dai moderatori.

Esempio 3: pannello impostazioni.

Supponiamo che l'utente *CosmoVellucci* voglia memorizzare una carta di credito oppure modificare l'immagine di profilo, oppure la password.

Potrà farlo attraverso il pannello delle impostazioni, dove troverà dei form appositi (assieme alla possibilità di cambiare la propria immagine di profilo e cambiare password).

Esempio 4: aggiunta al carrello, conferma acquisto.

Supponiamo che l'utente *CosmoVellucci* voglia comprare un gioco.

Per prima cosa, egli deve avere l'importo necessario sul portafoglio. Poi, dalla pagina del negozio selezionare uno o più giochi da aggiungere al carrello, infine confermare l'ordine dal carrello stesso.

Facendo così, CosmoVellucci aggiungerà alla libreria i suoi nuovi giochi.

Ogni gioco può essere aggiunto una sola volta alla libreria dello stesso utente.

Nel caso in cui non l'importo non sia abbastanza, al momento dell'acquisto viene stampato un messaggio di errore che invita l'utente a ricaricare il proprio portafoglio (tramite carta di credito).

Esempio 5: ricarica del portafoglio.

Ogni utente per effettuare degli acquisti deve prima ricaricare il suo portafoglio.

Per farlo, deve selezionare una somma da aggiungere dalla pagina impostazioni, attraverso un form.

Ovviamente, vista la semplicità del progetto, non ci sono controlli sulla carta di credito e qualsiasi valore può essere aggiunto al portafoglio, purchè sia stata associata una carta di credito all'account.

L'operazione avviene a livello back-end.

Esempio 6: logout.

Supponiamo che l'utente *CosmoVellucci* voglia effettuare il logout.

Per farlo, deve solamente cliccare sulla toolbar su **Logout**, da qualsiasi pagina l'utente si trovi, e verrà poi reindirizzato sulla mainpage.

Esempio 7: inserimento gioco.

Supponiamo che un utente sviluppatore voglia inserire il proprio videogioco all'interno del sito.

Per farlo, dalla toolbar l'utente deve accedere all'apposita pagina, dove saranno presenti diversi form per l'inserimento dei dati del videogioco, oltre a 4 pulsanti di upload che servono a caricare le immagini relative ad esso.

È inoltre presente un tasto di upload per l'icona del gioco.

Esempio 8: Login e registrazione.

Supponiamo che un utente voglia effettuare il login o registrarsi. Per farlo, deve cliccare sulla toolbar su **Login**.

Se possiede già l'account, può accedere tramite l'inserimento di username e password.

Se non lo possiede, può crearlo prima cliccando sull'apposito pulsante e poi compilando i form che appariranno.

2 Capitolo 2 - Manuale utente.

2.1 Accedere al sito (pagina principale).

Per accedere al sito, l'utente deve inserire il link

<http://lweb.dis.uniroma1.it/~lweb33/Piattaforma%20video%20games%20online/>

Una volta fatto, si troverà sulla pagina principale del sito, mostrata qui di seguito.



Prima di effettuare l'accesso, l'utente può visitare solo la pagina di accesso, il negozio, la pagina delle FAQ e informazioni. Queste pagine possono essere raggiunte attraverso la toolbar in alto a sinistra.

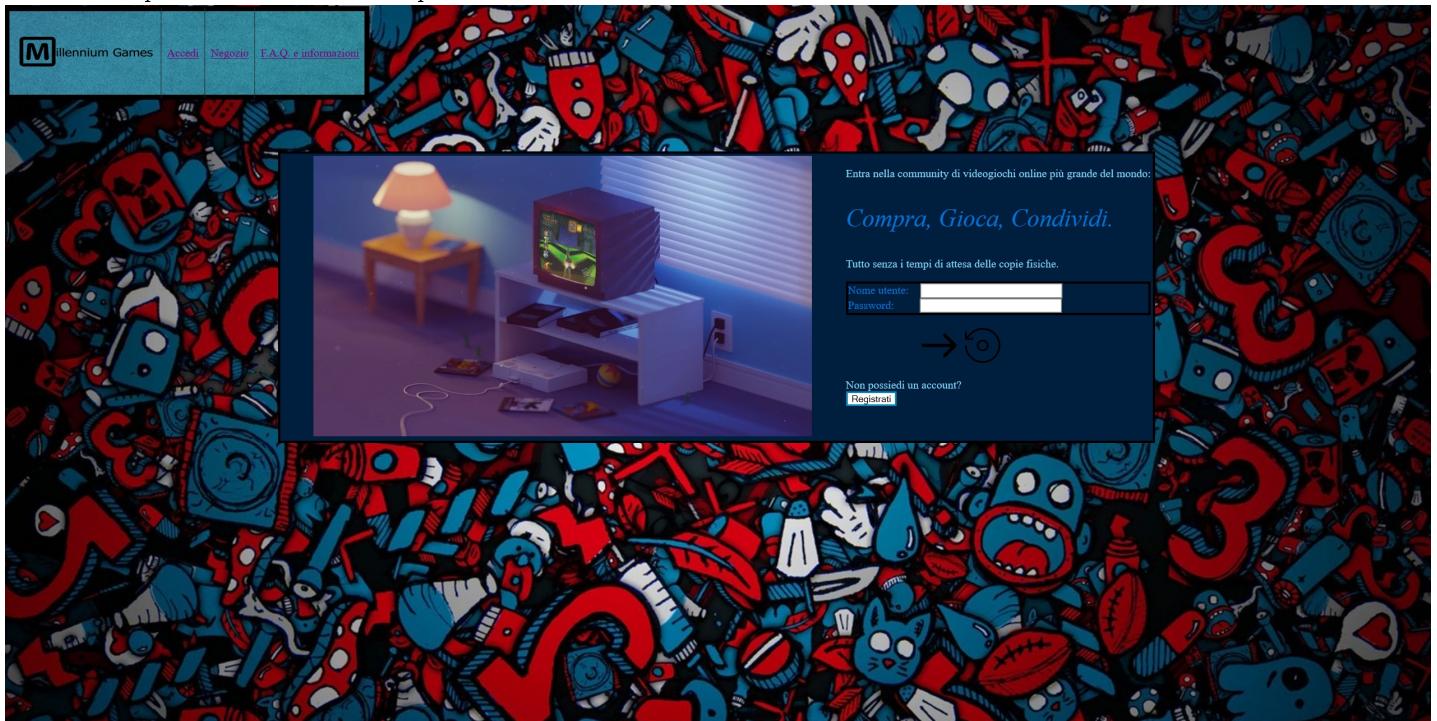
La pagina principale può essere raggiunta tramite il logo "Millennium Games" sulla sinistra della toolbar.

2.2 Login e registrazione.

L'utente può effettuare il login attraverso questa pagina. Per farlo deve inserire il proprio nome utente e la propria password.

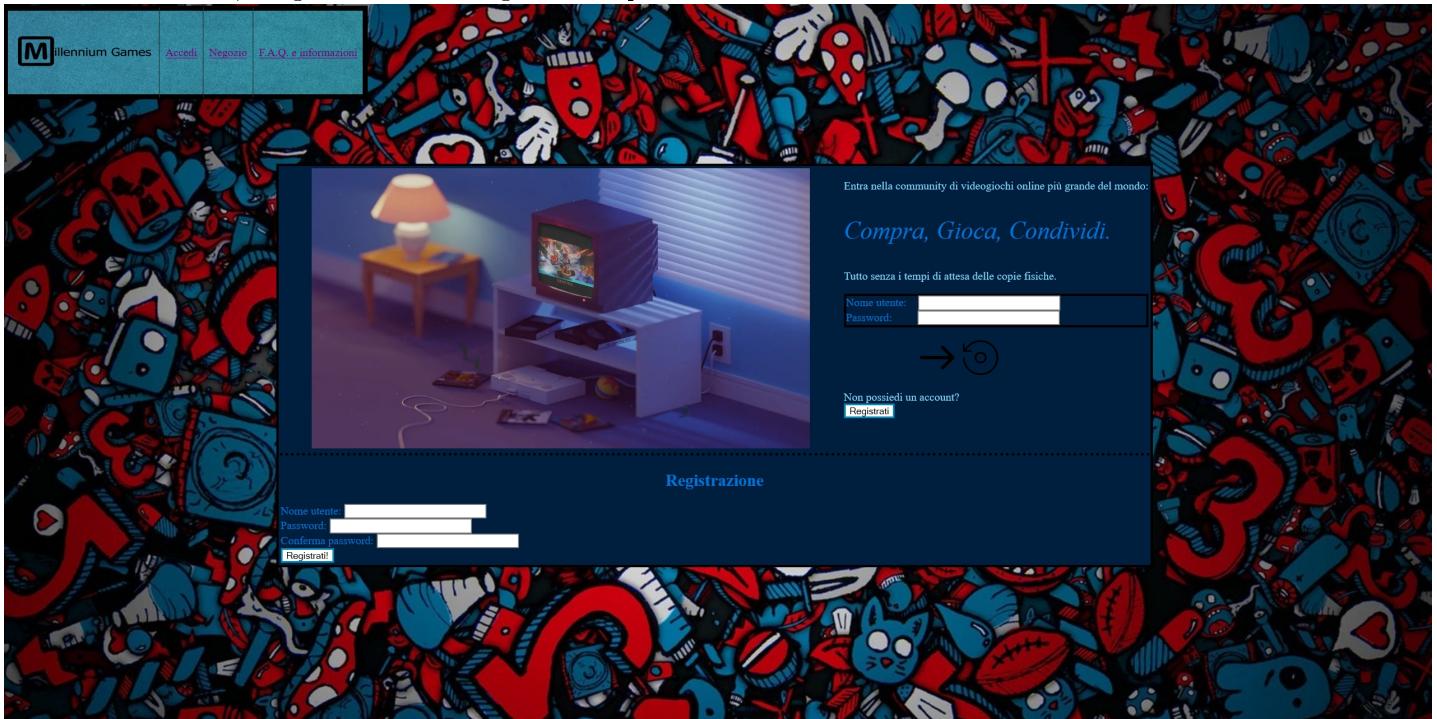
Una volta inseriti, si può procedere al login cliccando sulla freccia con punta a destra nera.

Si possono resettare i campi cliccando sulla freccia circolare nera.



Si può dare il via alla registrazione di un utente premendo sul tasto "Registrati".

In tal caso, vengono mostrati i seguenti campi.



L'utente deve scegliere un nome utente che non sia già esistente nel database e inserire una password, da confermare nell'apposito campo.

Una volta registrato, l'utente effettua l'accesso.

Dopo aver effettuato l'accesso, l'utente viene reindirizzato alla pagina principale.

Inoltre, la toolbar cambia da



A questa nuova toolbar avente altre funzionalità.



2.3 Pagina delle FAQ.

La pagina delle FAQ contiene le informazioni del sito. Queste sono una descrizione, le tre figure principali che lavorano al sito, una mail, una foto di dove si colloca l'ufficio, delle domande e le relative risposte riguardo il sito.

Cos'è questo sito.

Millennium Games è una piattaforma online dove puoi comprare e giocare i tuoi videogames preferiti. Perché aspettare inutilmente i tempi di attesa delle copie fisiche, occupando inoltre spazio ed inquinando ulteriormente attraverso cd e confezioni di plastica? Registrati, accedi, aggiungi alla tua libreria personale, gioca, e se vuoi condividi con gli amici le tue emozioni su una vasta gamma di titoli.

Chi siamo.

Stephen Catwing
Dopo essersi disintossicato dall'erba gatta, Stephen Catwing decise di ricominciare da zero diventando CEO di Millennium Games.

Albert Catstein
Programmatore web, è gestore del sito Millennium Games. Basa il proprio lavoro sul suo emblematico motto "Non so cosa sta facendo, ma funziona quindi va bene."

Linus Catvalds
Responsabile delle risorse umane, è a tutti gli effetti dedicato a abolire ogni abuso su animali, a partire dal gatto nella scatola di Schrödinger.

2.4 Negozio.

Il negozio contiene la lista dei giochi del sito.

Cliccando sul nome di un gioco si viene reindirizzati alla pagina del gioco stesso.

| ID | Titolo | Genere | Prezzo | Disponibilità |
|----|-----------------------|------------|---------|---------------|
| 31 | The Witcher 3 | GDR | 30.50 € | 2016-10-25 |
| 2 | Spyro Re晶ited Trilogy | Platformer | 20.00 € | 2010-07-10 |
| 25 | Medieval City | Gestionale | 5.00 € | 2018-05-07 |
| 77 | Metro Exodus | FPS | 50.00 € | 2033-01-02 |

2.5 Logout.

In qualsiasi momento l'utente può effettuare il logout attraverso la toolbar. Una volta fatto, si viene reindirizzati alla pagina principale.

2.6 Libreria.

Attraverso la toolbar l'utente può accedere alla propria libreria.

In questa pagina vengono mostrati i videogiochi comprati dall'utente.

Viene inoltre simulato un bottone (in realtà finto) che in teoria permetterebbe l'utente di giocare a un certo titolo avviando il gioco.

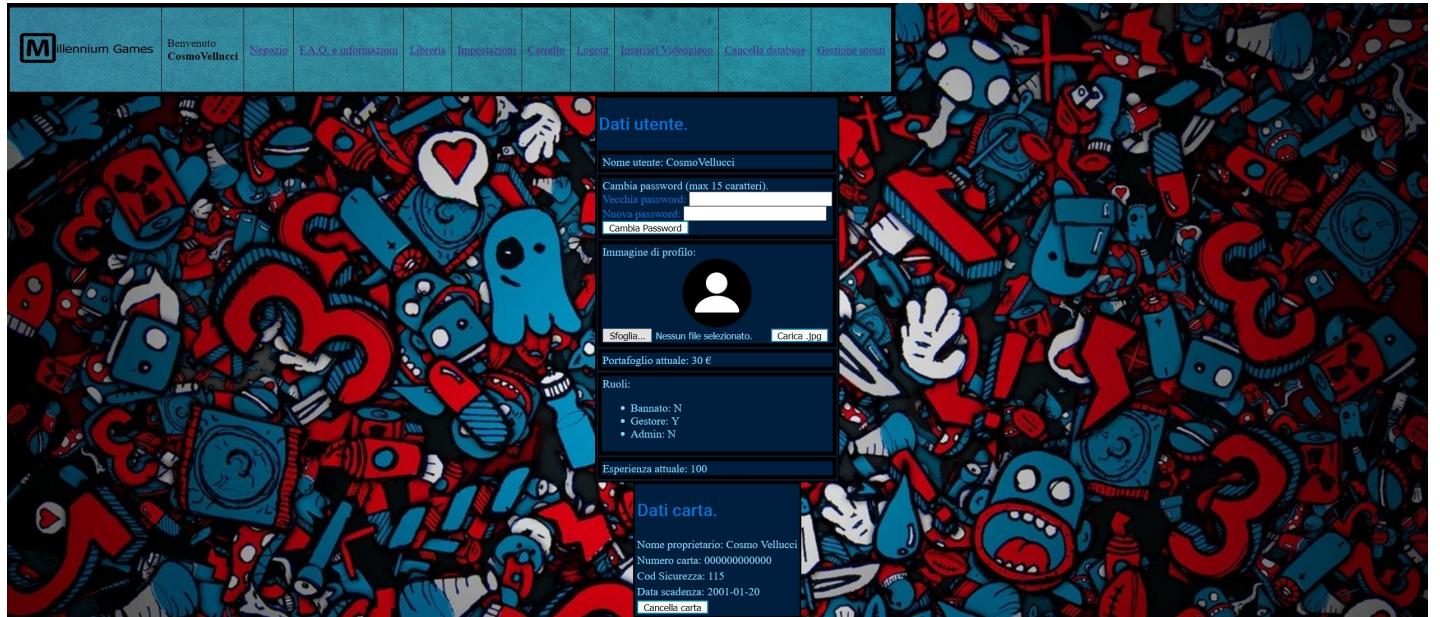


2.7 Impostazioni.

L'utente può cambiare le proprie impostazioni nella pagina "Impostazioni", sempre accessibile dalla toolbar.

In tale pagina l'utente può effettuare le seguenti funzioni...

- Password (inserendo la vecchia e nuova password).
- Immagine di profilo nel caso sia settata quella di default.
- Rimuove l'immagine di profilo nel caso non sia settata quella di default, ma appunto sia stata caricata un'immagine dall'utente.



Sono inoltre visibili le informazioni relative al nome utente, alla quantità di crediti nel portafoglio, all'esperienza, ai ruoli e alla carta di credito.

Nel caso in cui non sia già inserita la carta di credito, appare un form per inserirla (figura seguente).

Nel caso ci sia già una carta di credito, è possibile rimuoverla.

A screenshot of a credit card insertion form titled 'Dati carta.'. The form instructions say 'Nessuna carta di credito inserita. Compila i form per memorizzarne una.' It contains four input fields: 'Nome proprietario (max 300 caratteri):' with a placeholder, 'Numero carta (esattamente 12 caratteri):' with a placeholder, 'Codice di sicurezza (esattamente 3 caratteri):' with a placeholder, and 'Data di scadenza (formato YYYY-MM-DD):' with a placeholder. At the bottom is a 'Memorizza' button.

2.8 Carrello.

In questa pagina è possibile visualizzare il carrello. Se vuoto, appare nel seguente modo:

The screenshot shows the Millennium Games website's navigation bar at the top with links for Benvenuto Arturo, Negozio, F.A.Q. e informazioni, Libreria, Impostazioni, Carrello, Logout, Inserisci Videogioco, Cancella database, and Gestione utenti. Below the navigation bar, a message "Il carrello è attualmente vuoto." (The cart is currently empty) is displayed against a colorful, abstract background.

Altrimenti, il carrello viene visualizzato in questo modo:

The screenshot shows the same website interface with a populated shopping cart. The cart contains two items: "The Witcher 3" and "Medieval City". The total price is 35.50 €. At the bottom of the cart view, there are buttons for "Svuota carrello" (Empty cart), "Rimuovi dal carrello" (Remove from cart), and "Conferma" (Confirm).

Il tal caso è possibile rimuovere uno o più prodotti dal carrello, svuotare il carrello oppure confermare gli acquisti.

2.9 Inserimento di un videogioco.

Se l'account usato dall'utente non possiede il ruolo di gestore, appare un pulsante per tornare indietro alla pagina precedente.

Autorizzazione non consentita. Clicca sul pulsante per tornare indietro.
Val.

Se l'account usato dall'utente possiede il ruolo di gestore, allora può inserire un videogioco tramite il seguente insieme di form.

The form is titled "Compilare tutti i seguenti campi per aggiungere il videogioco." It contains fields for ID, Nome, Genere, Descrizione, Prezzo, and Disponibilità, each with a corresponding input field. Below the fields is a section for uploading images with the instruction "Inserire da 0 a 4 immagini jpg". At the bottom are three buttons: "Sfoglia...", "Nessun file selezionato.", and "Aggiungi Gioco".

2.10 Ripristino del database.

Se l'account usato dall'utente non possiede il ruolo di amministratore, appare un pulsante per tornare indietro alla pagina precedente.

Autorizzazione non consentita. Clicca sul pulsante per tornare indietro.
Val.

Se l'account usato dall'utente possiede il ruolo di amministratore, allora può ripristinare il database cliccando sull'apposito pulsante.

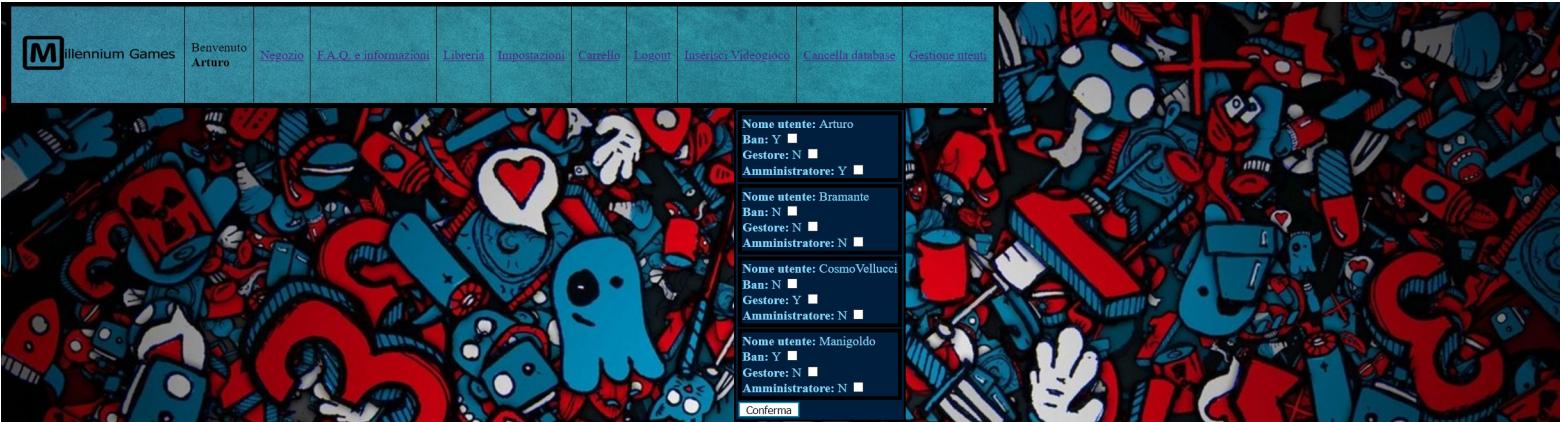


2.11 Gestione utenti.

Se l'account usato dall'utente non possiede il ruolo di amministratore, appare un pulsante per tornare indietro alla pagina precedente.

Autorizzazione non consentita. Clicca sul pulsante per tornare indietro.
Val.

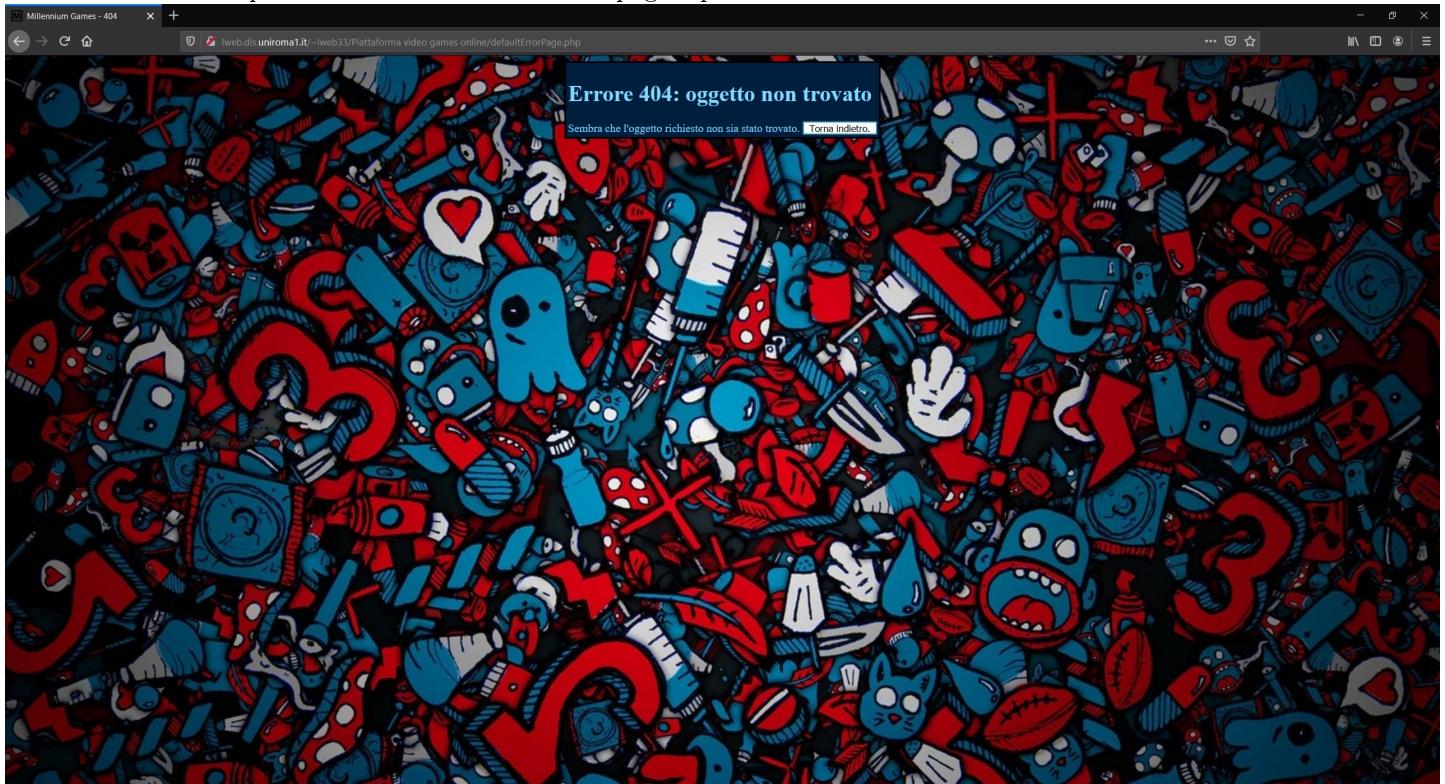
Se l'account usato dall'utente possiede il ruolo di amministratore, allora può gestire i ruoli dell'utente. L'amministratore può cambiare uno o più ruoli di uno o più utenti selezionando gli appositi checkbox e premendo sul pulsante di conferma.



Nel caso sia stato premuto il tasto di conferma ma non sia stato selezionato alcun ruolo da cambiare, appare un messaggio di errore.

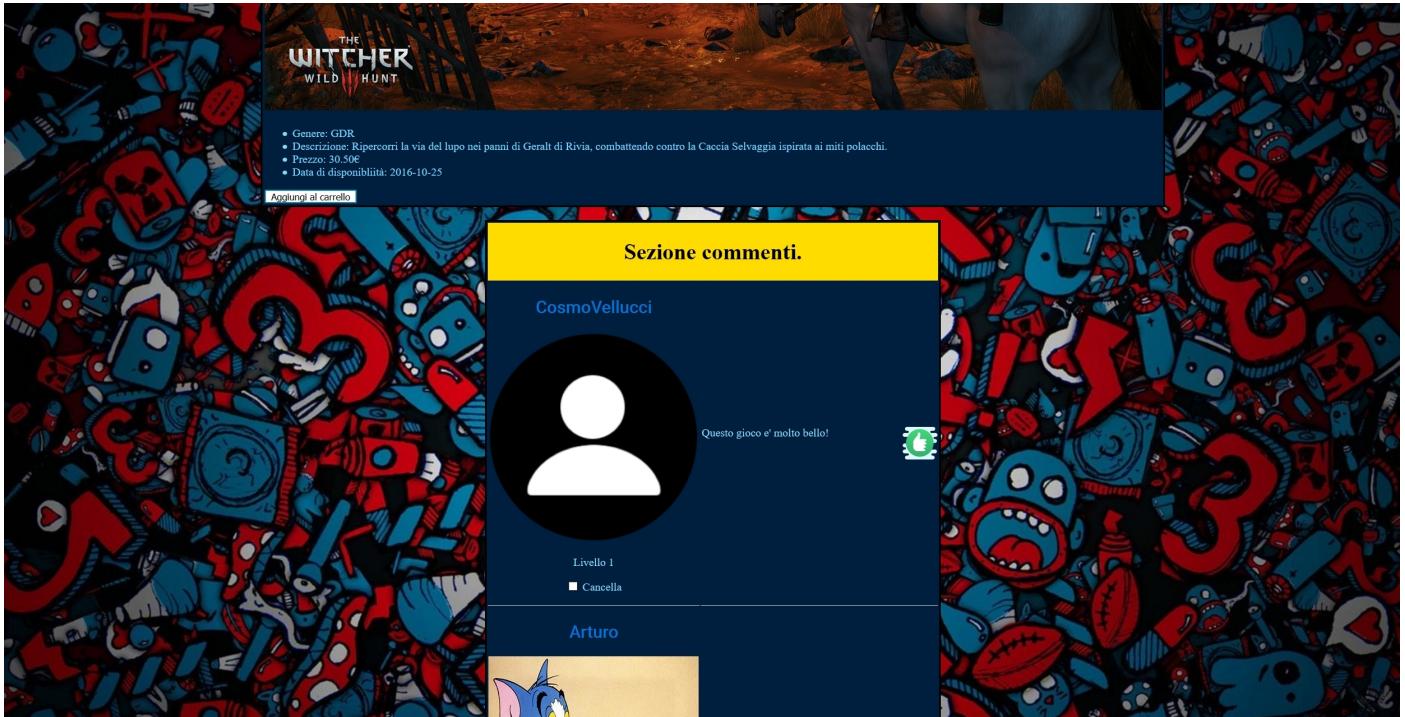
2.12 Pagina 404.

Nel caso in cui si richieda un oggetto non presente sul server, si finisce sulla seguente pagina. L'unica cosa che si può fare è tornare indietro alla pagine precedente.



2.13 Pagina di un gioco.

Ogni gioco ha la sua propria pagina, avente da una a quattro immagini descrittive e le caratteristiche del gioco. L'utente può aggiungere il videogioco al carrello nel caso in cui abbia fatto l'accesso al sito. Ogni utente può vedere i commenti, e se questo ha effettuato l'accesso al sito e non è bannato può commentare.



3 Capitolo 3 - Programmazione.

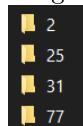
3.1 Panoramica della cartella.

La panoramica della cartella del sito è la seguente:

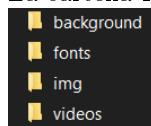
| | | |
|---------------------------|------------------|------------------|
| games | 28/06/2020 18:11 | Cartella di file |
| materials | 24/06/2020 01:29 | Cartella di file |
| profiles | 24/06/2020 01:29 | Cartella di file |
| .htaccess | 11/11/2019 15:59 | File HTACCESS |
| AggiungiRegistrazione.php | 11/11/2019 15:28 | File PHP |
| AggiuntaCarrello.php | 28/06/2020 21:42 | File PHP |
| cancDB.php | 30/05/2020 16:36 | File PHP |
| Carrello.php | 28/06/2020 01:27 | File PHP |
| connessione.php | 12/11/2019 17:14 | File PHP |
| DatiDB.php | 14/05/2020 15:49 | File PHP |
| defaultErrorPage.php | 11/11/2019 21:22 | File PHP |
| DisplayCarta.php | 20/06/2020 14:08 | File PHP |
| Faq.php | 28/06/2020 17:08 | File PHP |
| FormCarta.php | 20/06/2020 17:59 | File PHP |
| formUtenti.php | 27/06/2020 15:44 | File PHP |
| ImpostazioniUtente.php | 29/06/2020 23:20 | File PHP |
| index.php | 01/02/2020 20:39 | File PHP |
| InserimentoGioco.php | 22/06/2020 20:59 | File PHP |
| InserimentoGiocoForm.php | 27/06/2020 15:30 | File PHP |
| install.php | 28/06/2020 14:46 | File PHP |
| Libreria.php | 29/06/2020 02:08 | File PHP |
| Login.php | 23/06/2020 00:05 | File PHP |
| Logout.php | 25/06/2020 11:52 | File PHP |
| main | 03/07/2020 19:09 | Documento CSS |
| Negozio.php | 28/06/2020 18:34 | File PHP |
| QueryUtente.php | 22/05/2020 17:34 | File PHP |
| SchemaCommenti.xsd | 27/05/2020 15:26 | File XSD |
| SchemaGiochi.xsd | 23/05/2020 15:39 | File XSD |
| SchemaLibrerie.xsd | 16/05/2020 02:04 | File XSD |
| SezioneCommenti.php | 03/07/2020 19:18 | File PHP |
| SpecificoDatiCommenti | 27/05/2020 15:30 | Documento XML |
| SpecificoDatiLibrerie | 23/06/2020 19:59 | Documento XML |
| SpecificoDatiVideogiochi | 20/06/2020 23:49 | Documento XML |
| toolbar.php | 30/05/2020 22:55 | File PHP |
| Utenti.php | 30/05/2020 20:46 | File PHP |

Come si può notare, ci sono tre sottocartelle...

- La cartella **games** contiene le sottocartelle dei videogiochi. Ogni sottocartella è nominata con l'id del gioco che rappresenta, e contiene la pagina e gli screenshot del gioco stesso.



- La cartella **materials** contiene i file "estetici" del sito. Le sottocartelle sono autoesplicative.



- La cartella **profiles** contiene le cartelle degli utenti e per ognuno (eventualmente) l'immagine di profilo caricata dall'utente. È inoltre presente l'immagine di default per gli utenti che non hanno caricato un'immagine (cartella vuota).



3.2 Scripting di programmazione.

Qui di seguito sono mostrati gli script di programmazione per ogni pagina.

3.2.1 .htaccess

In questo file è presente la regola per la pagina 404.

```
❶ .htaccess x
C: >xampp >htdocs >Sito >❷ .htaccess
❸   ErrorDocument 404 http://lweb.dis.uniroma1.it/~lweb33/Piattaforma%20video%20games%20online/defaultErrorPage.php
```

DefaultErrorPage.php è una pagina contenente sostanzialmente del codice html che crea un bottone per portare l'utente alla pagina precedentemente visitata.

3.2.2 toolbar.php

La toolbar del gioco presenta uno script che la espande aggiungendo nuove funzionalità nel momento in cui l'utente ha effettuato l'accesso. L'accesso viene verificato attraverso la variabile `$_SESSION['loginEffettuato']`. Se pari a TRUE, rende possibile raggiungere le altre pagine. In rosso sono stati evidenziati i problemi riscontrati durante la progettazione del sito.

```
❶ 3 <table border="1" style="background: url(materials/background/tableBack.jpg); border-style: double; border: 7px solid black; border-collapse: collapse;" >
 4   <tr>
 5     <td style="padding: 8px;"><a href="index.php"></a> </td>
 6     <td style="padding: 8px;">
 7       <?php /*Se l'utente non è ancora connesso, metto "Login".*/
 8       if(isset($_SESSION['loginEffettuato']))==FALSE)
 9         {echo "<a href=\"Login.php\"> Accedi </a>";}
10       else /*Se l'utente è già connesso, metto le impostazioni.*/
11         {echo "Benvenuto <br /> <span style=\"font-weight: bold;\">" . $_SESSION['userName'] . "</span>";}
12     ?>
13   </td>
14   <td style="padding: 8px;"> <a href="Negozio.php" Negozio</a> </td>
15   <td style="padding: 8px;"> <a href="FAQ.php" F.A.Q. e informazioni </a> </td>
16
17   <!--Mostro delle celle aggiuntive con logout, libreria, carrello, impostazioni; Quando l'utente ha effettuato il login.-->
18   <?php
19   /*Come per sopra, if($_SESSION['loginEffettuato'])==TRUE) non va bene perchè la variabile la creo solo dopo il login.*/
20   if(isset($_SESSION['loginEffettuato'])==TRUE) /*isset infatti determina se una variabile è dichiarata e differente da NULL.*/
21   {
22     echo "<td style='padding: 8px;'> <a href=\"Libreria.php\"> Libreria </a></td>";
23     echo "<td style='padding: 8px;'> <a href=\"ImpostazioniUtente.php\"> Impostazioni </a></td>";
24     echo "<td style='padding: 8px;'> <a href=\"Carrello.php\"> Carrello </a></td>";
25     echo "<td style='padding: 8px;'> <a href=\"Logout.php\"> Logout </a></td>";
26     echo "<td style='padding: 8px;'> <a href=\"InserimentoGioco.php\"> Inserisci Videogioco </a> </td>";
27     echo "<td style='padding: 8px;'> <a href=\"cancDB.php\"> Cancella database </a> </td>";
28     echo "<td style='padding: 8px;'> <a href=\"Utenti.php\"> Gestione utenti </a> </td>";
29   }
30
31   ?>
32   </tr> <!--Fine della riga.-->
</table> <!--Fine della tabella.-->
```

Sostanzialmente, la toolbar è costituita da dei collegamenti ipertestuali che reindirizzano l'utente nelle altre pagine del sito.

3.2.3 index.php

La pagina principale del sito è costituita da un semplice script che sceglie in modo casuale l'immagine presente sotto la scritta "coming soon". Per realizzare ciò si utilizza la funzione `rand(1,3)`, che randomizza appunto il numero dell'immagine che deve essere mostrata ogni volta che si accede alla pagina.

```
<!--Script relativo al banner gigante.-->
<?php $n = rand(1, 3); ?>
<br />
<!--Banner "Coming Soon"-->
<div class="bannerMP">
  <div class="bannerMP_CS"> COMING SOON </div>
  
</div>
</body>
</html>
```

Mi serviva uno strumento che scegliesse in modo casuale un'immagine. Non trovandolo in html, ho risolto in php tramite la funzione rand.

3.2.4 Login.php

Attraverso il seguente script l'utente può accedere al sito. Una volta premuto il bottone di submit, lo script verifica prima di tutto tramite le funzioni `empty()` che i campi login e password siano stati compilati (in caso contrario viene stampato un messaggio di errore).

Fatto questo, si procede con una query sull'utente e si verifica che i dati inseriti siano presenti nel database (assieme un controllo sulla query stessa). In caso contrario si stampa un messaggio di errore.

Se i campi sono stati inseriti correttamente, si inizia la sessione e si inizializzano el variabili relative all'utente (carrello, username, la spesa fin ora, una variabile che memorizza l'accesso effettuato dall'utente).

Si viene poi reindirizzati alla pagina principale.

```

47     <!-- Controlli sul form di login -->
48     <?php
49         if (isset($_POST['invio'])) /*Determina se una variabile è stata dichiarata ed è differente da NULL.*/
50         {
51             /*Controllo che nessuno dei due form siano vuoti.*/
52             if (empty($_POST['username']) || empty($_POST['password']))
53                 echo "<br /><span style=\"color: red;\">Si prega di compilare tutti i campi.</span><br />";
54             else /*Procedo al login.*/
55             {
56                 echo "<br /><span style=\"ciano;\">Login in corso...</span><br />";
57                 /*Effettuo la query.*/
58                 $query="SELECT *
59                         FROM $NomeTabellaUtente
60                         WHERE nomeUtente = \"{$_POST['username']}\""
61                         AND password = \"{$_POST['password']}\" ";
62                 /*Effettuo la query sopra.*/
63                 if (!$risultatoQuery = mysqli_query($CONNESSIONE, $query))
64                 {
65                     echo "Errore nell'esecuzione della query per il login. <br />" . mysqli_error($CONNESSIONE);
66                     exit(); /*Termino lo script.*/
67                 }
68
69                 /*Giunti a questo punto, devo controllare i dati inseriti dall'utente con quelli del DB.*/
70                 $riga = mysqli_fetch_array($risultatoQuery);
71                 if ($riga) /*Se la riga ottenuta non è vuota, do inizio alla sessione.*/
72                 {
73                     session_start(); /*Inizio la sessione*/
74                     /*Memorizzo tutti i parametri necessari alla sessione.*/
75                     $_SESSION['carrello'] = array(); /*Memorizza il carrello degli acquisti*/
76                     $_SESSION['username'] = $_POST['username']; /*Memorizzo il nome utente.*/
77                     $_SESSION['spesaFinora']=0;
78                     $_SESSION['loginEffettuato']=TRUE; /*Stabilisce il diritto di accesso dell'utente.*/
79
80                     /*Passo alla pagina principale.*/
81                     header("Location: index.php"); /*Location: reindirizza il browser all'indirizzo a destra.*/
82                     exit(); /*Termino lo script, in quanto ho finito il login.*/
83                 }
84                 else /*Caso di inserimento dati errato.*/
85                 {
86                     echo "<span style=\"color: red;\">Errore: nome utente e/o password errati.</span><br />";
87                 }
88             }?> <!-- Fine controlli form.-->
89     <br /><br />
```

L'utente si può registrare cliccando sull'apposito bottone, che va ad aggiungere il form di registrazione contenuto in `AggiungiRegistrazione.php`

```

90     <!-- "ask" del form di registrazione.-->
91     <span class="ciano"> Non possiedi un account? </span>
92     <form action=<?php $_SERVER['PHP_SELF']?> method="post">
93         <input type="submit" class="bottone" name="Registrazione" value="Registrati"/>
94     </form>
95
96     </td>
97     </tr>
98     <!-- Attivazione del form per la registrazione.-->
99     <?php if (isset($_POST['Registrazione'])) {
100         {echo include 'AggiungiRegistrazione.php';}?>
```

Non sapevo come aggiungere un pannello di registrazione senza dover per forza utilizzare echo. Ho risolto con un include di un file avente contenuto html,

Lo script relativo al form contenuto in AggiungiRegistrazione.php è il seguente:

```
99 <?php if (isset($_POST['Registrazione']))  
10 {echo include 'AggiungiRegistrazione.php';}?>  
11  
12 <?php  
13 if(isset($_POST['ConfermaReg']))/*Check sul submit.*/
14 {  
15     if(empty($_POST['userNameReg']) || empty($_POST['pass1']) || empty($_POST['pass2'])) /*Check sull'inserimento di tutti i dati.*/
16     {echo "<span style=\"color: red;\"> Inserire tutti i campi per la registrazione. </span>";} /*Fine check sull'inserimento dei dati.*/
17     else /*caso di compilazione dei 3 campi.*/
18     {  
19         $queryReg = "SELECT *  
20             FROM $NomeTabellaUtente  
21             WHERE nomeUtente = \"{$_POST['userNameReg']}\" ";
22  
23         /*check sulla query.*/
24         if(!$risultatoQuery = mysqli_query($CONNESSIONE, $queryReg))
25         {  
26             echo "Errore nell'esecuzione della query per la registrazione (1). <br />" . mysqli_error($CONNESSIONE);
27             exit(); /*Termino lo script.*/
28         }
29  
30  
31         $riga = mysqli_fetch_array($risultatoQuery);
32         if($riga) /*Riscontro di un utenname già esistente.*/
33         {echo "<span style=\"color: red;\"> Nome utente già esistente: inserire un'altro. </span>";}
34         elseif (strcmp($_POST['pass1'], $_POST['pass2']) != 0) /*Se password e conferma password sono uguali*/
35         {echo "<span style=\"color: red;\"> Le password non coincidono. </span>";}
36         elseif(strlen($_POST['userNameReg'])>20)
37         {echo "<span style=\"color: red;\"> Il nome utente deve essere al massimo di 20 caratteri. </span>";}
38         elseif(strlen($_POST['pass1'])>15)
39         {echo "<span style=\"color: red;\"> La password deve essere al massimo di 15 caratteri. </span>";}
40         else /*Dopo tutti i controlli, procedo ad inserire l'utente nel DB.*/
41         {  
42             $queryReg = "INSERT INTO $NomeTabellaUtente (nomeUtente, password, proPic, BAN, GEST, ADM, portafoglio, expe, nomePro, numeroCarta, codSic, datacad)  
43                 VALUES (" . $_POST['userNameReg']. ", " . $_POST['pass1']. ", 'profiles/default.png', \N, \N, \N, \"0.00\", \"0\", \"NULL\", \"NULL\", \"NULL\")";
44  
45             if (!$risultatoQuery = mysqli_query($CONNESSIONE, $queryReg))
46             {  
47                 echo "Errore nell'esecuzione della query per la registrazione (2). <br />" . mysqli_error($CONNESSIONE);
48                 exit(); /*Termino lo script.*/
49             }
50  
51             /*Inserimento della nuova libreria per l'utente appena registrato.*/
52             $contenutoXML="";
53             foreach (file("SpecificoDatiLibrerie.xml") as $nodo)
54             {$contenutoXML .= trim($nodo);}
55             $documentoXML = new DOMDocument();
56             $documentoXML->loadXML($contenutoXML);
57             $radice= $documentoXML->getElementsByTagName("LIBRERIE")->item(0);
58             $radice->appendChild($TAG_Libreria);
59             $documentoXML->save("SpecificoDatiLibrerie.xml");
60  
61             /*Creazione della cartella dell'utente.*/
62             $dir = "profiles/" . $_POST['userNameReg'];
63             mkdir($dir, 0700, true);
64  
65             echo "<span style=\"color: green;\"> Registrazione compiuta! Esegui l'accesso. </span>";
66             exit();
67         }
68     }
69 } /*Fine check submit.*/
70 ?>
```

Una volta compilato i campi correttamente e effettuato i controlli sulla lunghezza di password e nome utente si va a inserire la query del nuovo utente all'interno del database e si fa il controllo dell'esecuzione su di essa.

Dopodichè, tramite XML DOM si va a inserire la libreria (vuota) dell'utente nel file **SpecificaDatiLibrerie.xml** e si crea la cartella dell'utente.

Ho trovato difficile il modo per automatizzare la creazione delle cartelle per gli utenti. Basandomi sul fatto che ogni utente possiede un nome unico e quindi una cartella altrettanto unica associata, sono poi riuscito ad automatizzare il processo.

3.2.5 Logout.php

Questo è un semplice script che permette all'utente di effettuare il logout attraverso le funzioni `unset($_SESSION)` e `session_destroy()`

```
3   <?php
4   session_start(); /*Inizio la sessione.*/
5   unset($_SESSION); /*Pulisco le variabili della sessione, come se fosse $_SESSION=Array;*/
6   session_destroy(); /*distruggo i dati registrati alla sessione, come i file di sessione nel file system.*/
7 ?>
8
9
10  <?xml version="1.0" encoding="UTF-8"?>
11  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
12  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
13      <head>
14          <title> Millennium Games </title> <!--Titolo della pagina.-->
15          <link rel="stylesheet" type="text/css" href="main.css" /> <!--Style principale.-->
16          <link rel="icon" href="materials/img/iconaSito.png" /> <!--Icona del sito-->
17      </head>
18      <body>
19          <p align="center">Logout in corso, verrai reindirizzato alla pagina di login...</p>
20          <?php header("Location: index.php"); ?>
21      </body>
22  </html>
```

3.2.6 cancDB.php

Lo script seguente verifica se l'utente abbia il ruolo di amministratore.

In caso di risposta negativa, appare un bottone che se cliccato porta l'utente alla pagina precedente.
In caso di risposta positiva, appare un bottone che se cliccato permette di ripristinare il database.

```
18  <?php
19  if($riga['ADM']=='N')
20  {
21      echo "<table class=\"backgroundTabella\">
22          <tr><td class=\"ciano\">
23              Autorizzazione non consentita. Clicca sul pulsante per tornare indietro.<br/>
24              <button class=\"bottone\" onclick=\"TornaIndietro()\"> Vai. </button>
25          </td></tr>
26      </table>";
27  }
28  else
29  {
30      echo "<table class=\"backgroundTabella\">
31          <tr><td class=\"ciano\"> Premi per ripristinare il database:
32              <form action=\"". $ SERVER['PHP_SELF']."\" method=\"post\">
33                  <input type=\"submit\" class=\"bottone\" name=\"cancellaDB\" value=\"cancella DB\" alt=\"submit\"/>
34              </form>
35          </td></tr>
36      </table>";
37  }
38 ?>
```

Questo script esegue il ripristino del database attraverso `DROP DATABASE`. Vi è inoltre un controllo sulla query.

Dopo aver effettuato il ripristino, l'utente viene reindirizzato alla pagina principale.

```
40  <?php
41  if(isset($_POST['cancellaDB']))
42  {
43      $query = "DROP DATABASE lweb33;";
44      if(mysqli_query($CONNESSIONE, $query)) {}
45      else
46          {echo "Errore nella cancellazione del database: " . mysqli_error($CONNESSIONE); exit();}
47      echo "Database ripristinato con successo!";
48      header('Location: index.php');
49  }
50 ?>
```

3.2.7 Carrello.php

In questa pagina viene mostrato il carrello dell'utente. Se vuoto, viene stampato un messaggio.

Altrimenti, vengono stampati i prodotti aggiunti al carrello, attraverso l'utilizzo di XML DOM si ottengono prezzo, nome e ID del videogioco aggiunto al carrello.

```

17     <?php
18         if (!($_SESSION['carrello'])) /*se il carrello non presenta valori.*/
19             {echo "<tr><td class=\"ciano\">Il carrello &egrave; attualmente vuoto.</td></tr>";}
20         else /*Caso di carrello con almeno un elemento.*/
21         {
22             $contenutoXML=""; /*Inizializzo la stringa che conterrà l'XML.*/
23             foreach (file("specificadativideogiochi.xml") as $nodo)
24                 {$contenutoXML .= trim($nodo); /*Mettiamo il file nel contenuto.*/
25                  /*Creazione del documento da usare.*/
26                  $documentoXML = new DOMDocument();
27                  /*Mettiamo il contenuto nel documento.*/
28                  $documentoXML->loadXML($contenutoXML);
29                  /*creo la lista degli elementi del documento XML.*/
30                  $elementi = $documentoXML->documentElement->childNodes;
31
32                  $_SESSION['spesaFinora']=0;
33                  $listaCarrello=<tr><td class="ciano" colspan="3"><h3 class="faqTitle"> Carrello</h3></td></tr>;
34                  <tr><td class="ciano">ID</td> <td class="ciano">Nome</td> <td class="ciano">Prezzo</td></tr>;
35
36                  foreach($_SESSION['carrello'] as $indice=>$valore)
37                  {
38                      for($i=0; $i<$elementi->length; $i++)
39                      {
40                          $elemento = $elementi->item($i); /*singolo elemento (videogioco) che deve essere lavorato nel for.*/
41                          $objPuntato=$elemento->firstChild;
42                          if($valore==$objPuntato->textContent)
43                          {
44                              $objPuntato=$objPuntato->nextSibling;
45                              $nomegioco=$objPuntato->textContent;
46                              $objPuntato=$objPuntato->nextSibling->nextSibling->nextSibling;
47                              $prezzoGioco=$objPuntato->textContent;
48                              $listaCarrello.= "<tr>
49                               <td class="ciano">
50                               <input type="checkbox" name="daEliminare[] value=\"$indice\" form=idElimina"/> . $valore . </td>
51                               <td class="ciano"> $nomegioco .</td>
52                               <td class="ciano"> $prezzoGioco . &euro;</td>
53                               </tr>";
54                          $SESSION['spesaFinora']+=$prezzoGioco;
55                          break;
56                      }
57                  }
58              }
59
60
61             $listaCarrello.=<tr><td colspan="3" class="ciano" style="text-align:center;">i>Prezzo totale: </i> ". $_SESSION['spesaFinora'] .&euro;</td></tr>;

```

Inizialmente ho trovato difficoltà nel trovare un modo per reperire i dati dei videogiochi contenuti nel carrello. Ho semplicemente risolto attraverso due for nidificati.

Con questo script si creano poi i buttoni per svuotare il carrello, confermare le compere, rimuovere alcuni elementi.

```

59             $listaCarrello.=<tr><td colspan="3" class="ciano" style="text-align:center;">i>Prezzo totale: </i> ". $_SESSION['spesaFinora'] .&euro;</td></tr>;
60
61
62             $listaCarrello.=<tr>;
63             /*Form di azzeramento del carrello.*/
64             $listaCarrello.=<td>
65                 <form action="" . $_SERVER['PHP_SELF'] . "\ method="post">
66                     <input type="submit" name="ClearCarrello" value="svuota carrello" />
67                 </form>
68             </td>;
69             /*Form di rimozione di un elemento dal carrello.*/
70             $listaCarrello.=<td>
71                 <form action="" . $_SERVER['PHP_SELF'] . "\ id="idElimina" method="post">
72                     <input type="submit" name="Rimuovi" value="Rimuovi dal carrello" />
73                 </form>
74             </td>;
75             /*Form di conferma della compra.*/
76             $listaCarrello.=<td>
77                 <form action="" . $_SERVER['PHP_SELF'] . "\ method="post">
78                     <input type="submit" name="confermaCompere" value="Conferma" />
79                 </form>
80             </td>;
81
82             $listaCarrello.=</tr>;
83
84
85             echo $listaCarrello;
86         }
87
88     ?>

```

Se viene premuto il pulsante per svuotare il carrello, si utilizza `unset` per svuotare l'array che rappresenta il carrello.

Se viene premuto il pulsante per eliminare uno o più elementi selezionati, tramite `unset` si tolgoni quegli elementi dall'array.

Se viene premuto il pulsante per confermare le compere e se l'utente ha un portafoglio tale da coprire le spese, i giochi vengono aggiunti alla libreria dell'utente attraverso XML DOM (cercando prima la libreria dell'utente e poi aggiungendo come figli i numeri identificativi dei giochi comprati), i fondi del portafoglio dell'utente vengono decrementati e il carrello viene svuotato.

```

95     if(isset($_POST['ClearCarrello']))
96         | {$_SESSION['carrello']=array(); header("Refresh:0");}
97     if(isset($_POST['Rimuovi']))
98     {
99         if(empty($_POST['daEliminare']))
100            | {echo "<p style='color:red;'>Seleziona qualcosa da cancellare.</p>";}
101        else
102            | {foreach ($_POST['daEliminare'] as $indice=>$indiceDaEliminare)
103                | {unset($_SESSION['carrello'][$indiceDaEliminare]);}
104            header("Refresh:0");
105        }
106    }
107    if(isset($_POST['confermaCompere']))
108    {
109        include 'QueryUtente.php';
110        if($riga['portafoglio']>$_SESSION['spesaFinora'])
111        {
112            $punti=$_SESSION['spesaFinora']*2.5;
113            $query="UPDATE $NomeTabellaUtente
114            SET portafoglio = portafoglio-".$_SESSION['spesaFinora'],
115            | | expe = expe+$punti
116            WHERE nomeUtente = '".$_SESSION['userName']."' ;
117            if (!$risultatoQuery = mysqli_query($CONNESSIONE, $query))
118            | {echo "Errore nell'esecuzione della query per la conferma delle compere" . mysqli_error($CONNESSIONE); exit();}
119
120            /*Inserimento del videogioco nella libreria dell'utente.*/
121            $contenutoXML="";
122            foreach (file("SpecificheDatiLibrerie.xml") as $nodo)
123                | {$contenutoXML .= trim($nodo);}
124            $documentoXML = new DOMDocument();
125            $documentoXML->loadXML($contenutoXML);
126            $elementi = $documentoXML->documentElement->childNodes;
127
128            for($i=0; $i<$elementi->length; $i++)
129            [
130                $objPuntato = $elementi->item($i);
131                if($_SESSION['userName']==$objPuntato->getAttribute("utenteAssociato"))
132                {
133                    foreach($_SESSION['carrello'] as $indice=>$valore)
134                    {
135                        $TAG_id = $documentoXML->createElement("id", $valore);
136                        $objPuntato->appendChild($TAG_id);
137                    }
138                    $i=$elementi->length+1;
139                }
140            ]
141            $documentoXML->save("SpecificheDatiLibrerie.xml");
142            $_SESSION['carrello']=array();
143            header("Refresh:0");
144        }
145        else /*Caso di portafoglio insufficiente.*/
146        | {echo "<p style='color: red;'>Portafoglio insufficiente. Ricaricalo contattando un admin.</p>";}
147    }

```

3.2.8 Connessione.php

Ogni volta che viene chiamato questo file, si avvia una connessione tramite `session_start` e si effettua la connessione al database. Vi è inoltre un controllo sulla connessione.

```
1  <?php
2  session_start(); /*Inizio della sessione.*/
3  include 'DatiDB.php'; /*Carico i dati relativi al DB dal file.*/
4  include 'install.php';/*Carico i dati per popolare le tabelle.*/
5
6  /*Effettuo la connessione al fine di poter svolgere il login.*/
7  $CONNESSIONE = mysqli_connect($nome, $username, $password);
8  /*Effettuo un check sulla connessione.*/
9  if(!$CONNESSIONE) /*Riscontro negativo.*/
10 | {die("Connessione fallita".mysqli_connect_error());} /*Termino la connessione e stampo un messaggio di errore.*/
11 /*Selezione il database*/
12 mysqli_select_db($CONNESSIONE,$NomeDB);
13 ?>
```

In questo script vengono inclusi i file DatiDB.php e install.php

3.2.9 DatiDB.php

Contiene i dati da inserire per accedere al database.

```
4  <?php
5  /*Relative all'accesso.*/
6
7  $nome = "localhost";
8  $username = "lweb33";
9  $password = "lweb33";
10
11
12
13 /*Relative al DB.*/
14 $NomeDB = "lweb33"; /*Nome del database*/
15 $NomeTabellaUtente = "DatiUtentiRegistrati"; /*Nome della tabella contenente i dati degli utenti.*/
16 ?>
```

3.2.10 Install.php

Questo script effettua la creazione del database, assieme a creazione e popolamento dell'unica tabella ivi presente, nel caso entrambi non siano già esistenti. Sono inoltre presenti i controlli per le query.

Dopo aver terminato lo script la connessione del database viene chiusa tramite `mysqli_close()`

```
15 /*Crea il database*/
16 $query = "CREATE DATABASE IF NOT EXISTS $NomeDB"; /*Memorizzo la query per la creazione del DB.*/
17 if(mysqli_query($CONNESSIONE, $query)) {}/*Caso di creazione effettuata con successo*/
18 else /*Caso di creazione effettuata con insuccesso*/
19 {echo "Errore nella creazione del database:" . mysqli_error($CONNESSIONE);}
20
21 /*Selezione il database*/
22 mysqli_select_db($CONNESSIONE,$NomeDB);
23
24 /*Crea la tabella relativa all'utente.*/
25 $query = "CREATE TABLE IF NOT EXISTS $NomeTabellaUtente
26 (
27     nomeUtente VARCHAR(20) NOT NULL PRIMARY KEY,
28     password VARCHAR(15) NOT NULL,
29     propPic VARCHAR(300),
30     BAN CHAR(1) NOT NULL DEFAULT 'N',
31     GEST CHAR(1) NOT NULL DEFAULT 'N',
32     ADM CHAR(1) NOT NULL DEFAULT 'N',
33     portafoglio FLOAT,
34     expe INT,
35     nomePro VARCHAR(300),
36     numeroCarta CHAR(12),
37     codSic CHAR(3),
38     dataScad DATE
39 )";
40 /*Check sulla creazione della tabella.*/
41 if(mysqli_query($CONNESSIONE, $query)) {}/*Caso di creazione effettuata con successo*/
42 else /*Caso di creazione effettuata con insuccesso*/
43 {echo "Errore nella creazione della tabella $NomeTabellaUtente:" . mysqli_error($CONNESSIONE);}
44
45 /*Passo a popolare la tabella*/
46 /*Tabella relativa agli utenti*/
47 $query = "INSERT IGNORE INTO $NomeTabellaUtente
48     (nomeUtente, password, propPic, BAN, GEST, ADM, portafoglio, expe, nomePro, numeroCarta, codSic, dataScad)
49     VALUES
50     ('Cosmo Vellucci', 'cosmoVellucci', 'profiles/default.png', '\n', '\Y', '\N', '30.00', '100', 'Cosmo Vellucci', '0000000000', '115', '01-01-20'),
51     ('Manigoldo', 'Manigoldo', 'profiles/Manigoldo/propic.jpg', '\Y', '\N', '\N', '0.0', '100', 'Manigoldo Manigoldini', '123456789012', '100', '10-10-25'),
52     ('Arturo', 'Arturo', 'profiles/Arturo/propic.jpg', '\Y', '\N', '\N', '700.0', '500', '\NULL', '\NULL', '\NULL', '\NULL'),
53     ('Bramante', 'Bramante', 'profiles/default.png', '\N', '\N', '\N', '0.0', '0', '\NULL', '\NULL', '\NULL', '\NULL');
54 if(!mysqli_query($CONNESSIONE, $query)) /*Check sul popolamento.*/
55 {echo "Errore nell'inserimento dati di " . $NomeTabellaUtente . mysqli_error($CONNESSIONE);}
56
57 /*Chiudo la connessione*/
58 mysqli_close($CONNESSIONE);
59 ?>
```

3.2.11 Negozio.php

Con questo script si costruisce il catalogo del negozio, attraverso l'utilizzo di XML DOM che preleva i dati dei videogiochi e li mette in una tabella. Sulla stessa riga del videogioco viene creato un radio e un bottone.

```

33 <?php
34     $contenutoXML = ""; /*inizializzo la stringa che conterrà l'XML.*/
35     foreach (file("SpecificoDatiVideogiochi.xml") as $nodo)
36     {
37         $contenutoXML .= trim($nodo); /*Mettiamo il file nel contenuto.*/
38     }
39     $documentoXML = new DOMDocument();
40     $documentoXML->loadXML($contenutoXML);
41
42     /*Creo la lista degli elementi del documento XML.*/
43     $elementi = $documentoXML->documentElement->childNodes;
44
45     /*Validazione del documento XML.*/
46     if($documentoXML->schemaValidate("SchemaGiocchi.xsd"))
47     {
48         /*Processo il catalogo.*/
49         $catalogo = "";
50         for($i=0; $i<$elementi->length; $i++)
51         {
52             $elemento = $elementi->item($i); /*singolo elemento (videogioco) che deve essere lavorato nel for.*/
53
54             $objPuntato=$elemento->firstChild;
55             $idGioco=$objPuntato->textContent;
56
57             $objPuntato=$objPuntato->nextSibling;
58             $nomeGioco=$objPuntato->textContent;
59
60             $objPuntato=$objPuntato->nextSibling;
61             $genereGioco=$objPuntato->textContent;
62
63             $objPuntato=$objPuntato->nextSibling->nextSibling;
64             $prezzoGioco=$objPuntato->textContent;
65
66             $objPuntato=$elemento->lastChild;
67             $dispoGioco=$objPuntato->textContent;
68
69             $catalogo.= "<tr>
70                         <td>
71                             <input type='radio' form='passaApagina' id='".$idGioco."' name='identificativoGioco' value='".$idGioco."'/>
72                             <label for='RN'>".$idGioco."</label>
73                         </td>
74                         <td>".$nomeGioco."</td> <td>".$genereGioco."</td> <td>".$prezzoGioco."&euro;</td> <td>".$dispoGioco."</td>
75                     </tr><td colspan='5'>
76                         <form id='passaApagina' action='". $_SERVER['PHP_SELF'] ."'" method='post'>
77                             <input type='submit' form='passaApagina' class='bottone' name='sendCommento' value='Visita pagina' />
78                         </form>
79                     </td></tr>";
80     echo $catalogo; /*Stampa del catalogo.*/
81 }
82 else {echo "Documento SchemaGiocchi.xsd non valido.";}
83 ?>
```

Una volta scelto un titolo e cliccato sul bottone, si crea una variabile `$_SESSION` che contiene l'ID del titolo scelto. Infine l'utente viene reindirizzato alla pagina `GamePage.php`

```

4   <?php
5     if(isset($_POST['identificativoGioco']))
6     {
7         $_SESSION['idDaPassare'] = $_POST['identificativoGioco'];
8         header('Location:games/GamePage.php'); /*Lo metto prima che sennò mi da problemi.*/
9     }
10 ?>
```

3.2.12 GamePage.php

Attraverso il valore passato in `$_SESSION` dalla pagina `GamePage.php` e quindi tramite XML DOM si cerca il gioco scelto dall'utente e si ricavano le informazioni e le immagini (queste se presenti) relative al videogioco, che vengono quindi mostrate nella pagina.

```
11  <?php
12  $contenutoXML="";
13  foreach (file("../SpecificheDatiVideogiochi.xml") as $nodo)
14  {
15      $contenutoXML .= trim($nodo);
16  }
17  $documentoXML = new DOMDocument();
18  $documentoXML->loadXML($contenutoXML);
19  $elementi = $documentoXML->documentElement->childNodes;
20
21  for($i=0; $i<$elementi->length; $i++)
22  {
23      $elemento = $elementi->item($i);
24
25      $objPuntato=$elemento->firstChild;
26      if($_SESSION['idDaPassare'] == $objPuntato->textContent)
27      {
28          $idGioco=$objPuntato->textContent;
29
30          $objPuntato=$objPuntato->nextSibling;
31          $nomeGioco=$objPuntato->textContent;
32
33          $objPuntato=$objPuntato->nextSibling;
34          $genereGioco=$objPuntato->textContent;
35
36          $objPuntato=$objPuntato->nextSibling;
37          $descrizione=$objPuntato->textContent;
38
39          $objPuntato=$objPuntato->nextSibling;
40          $prezzoGioco=$objPuntato->textContent;
41
42          /*Conto i file nella cartella .*/
43          $directory=$idGioco;
44          $files2 = glob( $directory ."/*" );
45          if($files2)
46          {
47              $filecount = count($files2);
48              for($i=0; $i<$filecount; $i++)
49              {
50                  $objPuntato=$objPuntato->nextSibling;
51                  $arrayIMG[$i]=$objPuntato->textContent;
52              }
53          }
54          else $filecount=0;
55
56          $objPuntato=$elemento->lastChild;
57          $dispoGioco=$objPuntato->textContent;
58
59          $i=$elementi->length; /*termino il loop.*/
60      }
61  }
62 ?>
```

Al fine di capire se mostrare o meno le immagini di un gioco, dovevo capire se fossero presenti e quante fossero le immagini stesse. Ho risolto rispettivamente attraverso l'utilizzo di una funzione `glob` e un `for`.

3.2.13 AggiuntaCarrello.php

Nel file GamePage.php viene incluso lo script AggiuntaCarrello.php, mostrato qui di seguito e che si occupa di aggiungere il gioco al carrello.

Si eseguono poi dei controlli tramite degli if()

Prima di tutto si verifica che il gioco sia disponibile all'acquisto confrontando la data di disponibilità del gioco con la data di aggiunta al carrello.

Poi si verifica che l'utente abbia effettuato l'accesso.

Si verifica poi che l'utente non abbia già il gioco e che non sia già stato inserito nel carrello.

Se tutte le condizioni appena elencate sono soddisfatte, si procede ad aggiungere il gioco nel carrello.

```
6  <?php if(isset($_POST['Aggioco'])) {
7      {
8          if($dispoGioco>date("Y-m-d")) /*Caso in cui il gioco non sia ancora uscito.*/
9              {echo "<p style='color:red;'>Il gioco non è disponibile.</p>";}
10         else if(isset($_SESSION['loginEffettuato'])==FALSE) /*Verifica che l'utente sia collegato.*/
11             {echo "<p style='color:red;'>Effettua l'accesso per comprare il gioco.</p>";}
12         else /*Controllo che il gioco non sia presente prima nella libreria e poi nel carrello.*/
13             {/*###Prima nella libreria###/
14                 $contenutoXML=""; /*Inizializzo la stringa che conterrà l'XML.*/
15                 foreach (file("../SpecificheAtLibrerie.xml") as $nodo)
16                     {$contenutoXML .= trim($nodo); } /*Mettiamo il file nel contenuto.*/
17                 $documentoXML = new DOMDocument();
18                 $documentoXML->loadXML($contenutoXML);
19                 if($documentoXML->schemavalidate("../SchemaLibrerie.xsd"))
20                     {
21                         $arrayLibreria=array();
22                         $elementi = $documentoXML->documentElement->childNodes;
23                         for($i=0; $i<$elementi->length; $i++)
24                             {
25                                 $objPuntato = $elementi->item($i);
26                                 if($_SESSION['userName']==$objPuntato->getAttribute("utenteAssociato"))
27                                     {
28                                         $listaGiochi=$objPuntato->getElementsByTagName("id");
29                                         for($j=0; $j<$listagiochi->length; $j++)/*Ricavo gli ID del gioco.*/
30                                             {$arrayLibreria[$j]=$listaGiochi->item($j)->textContent;}
31                                     }
32                                 }/*A questo punto si ha l'array della libreria dell'utente.*/
33                                 $trovato=0;
34                                 foreach ($arrayLibreria as $indice=>$valore)
35                                     {
36                                         if($valore==$idGioco)
37                                             {echo "<p style='color:red;'>Il gioco è già nella tua libreria.</p>"; $trovato=1; break;}
38                                     }
39                                 $trovato2=0; /*###Poi nel carrello###/
40                                 foreach ($_SESSION['carrello'] as $indice=>$valore)
41                                     {
42                                         if($valore==$idGioco)
43                                             {
44                                                 $trovato2=1;
45                                                 break;
46                                             }
47                                         } /*Fine del for.*/
48                                         if($trovato==0 && $trovato2==0)/*Altrimenti lo aggiungo */
49                                         {
50                                             $_SESSION['carrello'][] = $idGioco;
51                                             echo "<p style='color:green;'>Il gioco è stato aggiunto nel carrello.</p>";
52                                         }
53                                         }
54                                         else/*Caso di errore della validazione del documento.*/
55                                         {echo "<p style='color:red;'>Errore nella validazione del documento xml.</p>"; exit();}
56                                         } /*Fine controllo sull'oggetto già presente nella libreria e nel carrello.*/
57                                         } /*Fine if del form di aggiunta al gioco.*/
58 } /*Fine if del form di aggiunta al gioco.*/?>
```

3.2.14 SezioneCommenti.php

In GamePage.php viene inoltre incluso SezioneCommenti.php, mostrato qui di seguito e che si occupa di realizzare la sezione commenti per quella pagina del gioco.

Tramite XML DOM, prima si vede quale sezione dei commenti corrisponde al gioco mostrato in GamePage.php. Poi, per ogni commento non cancellato, si reperiscono le informazioni del commento stesso e dell'utente che ha commentato.

Alcune recensioni sono dotate di recensione, quindi nel caso in cui sia positiva viene messa l'immagine di un pollice in su, altrimenti l'immagine di un pollice in giù.

Se l'utente non ha effettuato l'accesso appare un messaggio che l'invita ad accedere.

Se l'utente è bannato non può commentare e appare un messaggio che lo specifica.

Altrimenti, l'utente può commentare scrivendo del testo nell'apposito textbox e poi inviare il commento con l'apposito pulsante.

In particolare, se l'utente è un gestore, sotto ogni foto profilo dell'utente di ogni commento viene aggiunto un checkbox e un pulsante. Attraverso questi strumenti il gestore può cancellare uno o più commenti. I commenti cancellati non vengono veramente cancellati dal file SpecificaDatiCommenti.xml, ma viene cambiato il tag commentoCancellato in NO che ne impedisce la pubblicazione.

Dopo la cancellazione di uno o più commenti la pagina viene aggiornata tramite `<meta http-equiv="refresh" content="0">`

L'aggiunta del commento viene effettuata memorizzando il commento stesso in coda a tutti gli altri nel file SpecificaDatiCommenti.xml, con attributo commentoCancellato="SI".

Dopo l'aggiunta del commento la pagina viene aggiornata tramite `<meta http-equiv="refresh" content="0">`

Per calcolare il livello dell'utente stavo per creare un sacco di if nidificati, rendendo il codice di orrenda scrittura. Poi ho pensato che lo switch poteva risolvere lo stesso il problema e quindi ho optato per tale soluzione.

```

1  <?php
2      $contenutoXML = ""; /*Inizializzo la stringa che conterrà l'XML*/
3      foreach (file("../specificadatiCommenti.xml") as $nodo)
4          {$contenutoXML .= trim($nodo); /*Mettiamo il file nel contenuto*/}
5      $documentoXML = new DOMDocument();
6      $documentoXML->loadXML($contenutoXML);
7
8      if($documentoXML->schemaValidate("../SchemaCommenti.xsd"))
9      {
10         echo "<tr><td colspan='2' style='background-color:#FFDC00;'><h1>Sezione commenti.</h1></td></tr>";
11         $i=0;
12         $elementi = $documentoXML->documentElement->childNodes;
13         while($i<$elementi->length)
14         {
15             $objPuntato = $elementi->item($i); /*singolo elemento (sezioneCommentiGioco) che deve essere lavorato*/
16             if($_SESSION['idDaPassare']==$objPuntato->getAttribute("idGioco"))
17             {
18                 $utente_e_commento=$objPuntato->getElementsByTagName("commento");/*Ottengo la lista dei commenti.*/
19                 if($utente_e_commento->length==0)/*Caso in cui non ci sono commenti.*/
20                     {echo "<tr><td colspan='2' class='ciano'> Non ci sono ancora commenti.</td></tr>"};
21
22                 for($j=0; $j<$utente_e_commento->length; $j++)/*Caso del commento */
23                 {
24                     if($utente_e_commento->item($j)->getAttribute('commentoCancellato')=='NO')
25                     {
26                         $utente=$utente_e_commento->item($j)->firstChild->textContent;
27                         $Commento=$utente_e_commento->item($j)->firstChild->nextSibling->textContent;
28                         $Recensione=$utente_e_commento->item($j)->lastChild->textContent;
29
30                         /*Effetto la query per ottenere l'immagine.*/
31                         $query="SELECT *
32                             FROM $nomeTabellaUtente
33                             WHERE nomeUtente=\"$utente\"";
34
35                         /*Esegui la query sopra.*/
36                         if (!$_risultatoQuery = mysqli_query($CONNESSIONE, $query))
37                             {echo "Errore nell'esecuzione della query. <br />". mysqli_error($CONNESSIONE); exit(); /*Termino lo script.*}
38                         $riga = mysqli_fetch_array($risultatoQuery);
39
40                         switch(true)
41                         {"Controllo il livello dell'utente."}
42                         case $riga['expe']<100: $plv=0; break;
43                         case $riga['expe']>200: $plv=1; break;
44                         case $riga['expe']>300: $plv=2; break;
45                         case $riga['expe']>400: $plv=3; break;
46                         case $riga['expe']>500: $plv=4; break;
47                         case $riga['expe']>600: $plv=5; break;
48                         case $riga['expe']>700: $plv=6; break;
49                         case $riga['expe']>800: $plv=7; break;
50                         case $riga['expe']>900: $plv=8; break;
51                         case $riga['expe']>1000: $plv=9; break;
52                         case $riga['expe']>10000: $plv=10; break;
53
54                         /*Raccolgo i dati per il commento.*/
55                         $codiceCommento=<tr><td style="border-bottom: 1px solid #ccc;"><h2 class="faqTitle"><b> . $utente . </b></h2>
56                         <img src='./'. $riga['proPic'] ." alt="Profile Picture" style="width:300px; height:300px;"/>
57                         <br> <p class='ciano'>ilivello . $plv .</p>;
58
59                         /*Check sull'utente bannato.*/
60                         if($riga['BAN']=='Y')
61                             {$codiceCommento.= "<p style='color:#FF0000'>Utente bannato</p>"}
62
63                         if(isset($_SESSION['loginEffettuato']))
64                         {
65                             include 'QueryUtente.php';
66                             if($riga['GEST']=='Y' && $_SESSION['loginEffettuato']==TRUE) /*Caso del gestore che vede i commenti.*/
67                             {$codiceCommento.= "<p class='ciano'><input type='checkbox' name='daEliminare[]' value='".$j."' form='idElimina' /> Cancella</p>"}
68
69                         $codiceCommento.= "</td>
70                         <td class='ciano' style='text-align: left; border-bottom: 1px solid #ccc;'> . $Commento;
71
72                         /*Aggiungo l'eventuale recensione.*/
73                         if($Recensione=='P')
74                             {$codiceCommento.= " <img align='right' src='../materials/img/tup.png' alt='Recensione positiva' style='width:15%;' />"}
75                         else if($Recensione=='N')
76                             {$codiceCommento.= " <img align='right' src='../materials/img/tdown.png' alt='Recensione negativa' style='width:15%;' />"}
77
78
79                         /*Termino e stampo il commento.*/
80                         $codiceCommento.= "</td></tr>";
81                         echo $codiceCommento;
82
83                     } /*Fine if del commento non cancellato.*/
84
85                     } /*Fine for dell'utente e del commento.*/
86                     break; /*Termino il for.*/
87
88                 }
89             } /*Fine While*/
90
91         } /*Caso di documento non validato.*/
92         {echo "<tr><td>Errore nella validazione del documento xml.</td></tr>"};

```

```

92     /*Sezione per il commento dell'utente collegato.*/
93     if(isset($_SESSION['loginEffettuato'])==FALSE) /*Caso in cui l'utente non sia collegato.*/
94         echo "<tr><td colspan=\"2\" class=\"ciano\">Accedi per commentare.</td></tr>";
95     else
96     {
97         include 'QueryUtente.php';
98         if($riga['BAN']=='Y')
99             echo "<tr><td colspan=\"2\" style=\"color:red;\">Sei stato bannato e non puoi pi&ugrave; commentare. Convinci un admin a farti sbannare.</td></tr>";
100        else
101        {
102            echo "<tr><td colspan=\"2\" class=\"ciano\">
103                Commenta qui:<br/>
104                <form id=\"idCommento\" action=\"".$_SERVER['PHP_SELF']."' method=\"post\">
105                    <textarea rows=\"3\" cols=\"60\" name=\"boxCommento\"></textarea><br/>
106                    <input type=\"submit\" form=\"idCommento\" class=\"bottone\" name=\"sendCommento\" value=\"Invia commento\"/>
107                    <input type=\"radio\" id=\"RP\" name=\"recensione\" value=\"Recensione positiva\"/>
108                    <label for=\"RP\">Recensione positiva</label>
109                    <input type=\"radio\" id=\"RN\" name=\"recensione\" value=\"Recensione negativa\"/>
110                    <label for=\"RN\">Recensione negativa</label>
111                    <input class=\"bottone\" type=\"reset\" name=\"ricomposta\" value=\"Ricomposta\"/>
112                </form>
113            </td></tr>";
114            if($riga['GEST']=='Y')
115            {
116                echo "<tr><td><form action=\"".$_SERVER['PHP_SELF']."' id=\"idElimina\" method=\"post\">
117                    <input type=\"submit\" name=\"Rimuovi\" value=\"Rimuovi Commenti\"/>
118                </form></td></tr>";
119            }
120
121         /*Caso di invio del commento.*/
122         if(isset($_POST['sendcommento']))
123         {
124             if(empty($_POST['boxCommento']))
125                 echo "<p style=\"color:red;\"> Scrivi un commento prima di inviare.</p>"; exit();
126             $contenutoXML="";
127             foreach (file("../specificabatiCommenti.xml") as $nodo)
128                 ($contenutoXML .= trim($nodo));
129             $documentoXML = new DOMDocument(); $documentoXML->loadXML($contenutoXML);
130             if(!$documentoXML->schemaValidate("../SchemaCommenti.xsd"))
131                 {echo "<p> Errore nella validazione del documento. </p>"; exit();}
132             $elementi = $documentoXML->documentElement->childNodes;
133             for($i=0; $i<$elementi->length; $i++)
134             {
135                 $objPuntato = $elementi->item($i);
136                 if($_SESSION['idaPassare']==$objPuntato->getAttribute("idGioco"))
137                 {
138                     break;
139                     $objPuntato = $objPuntato->lastChild;
140                 }
141
142                 $testoCommento = $_POST['boxCommento'];
143
144                 $TAG_commento = $documentoXML->createElement("commento"); /*Creo il tag del commento.*/
145                 $TAG_commento_Attributo = $documentoXML->createAttribute("commentoCancelled"); /*Aggiungo l'attributo al commento.*/
146                 $TAG_commento_Attributo->value="NO"; /*Setto l'attributo su NO.*/
147                 $TAG_commento->appendChild($TAG_commento_Attributo); /*Inserisco l'attributo al tag commento.*/
148                 $TAG_utente = $documentoXML->createElement("utente", $_SESSION['userName']); /*Creo il tag dell'utente.*/
149                 $TAG_utente->appendChild($TAG_commento); /*Memorizzo il tag del commento.*/
150                 $TAG_utente->appendChild($TAG_utente); // $objPuntato=$objPuntato->firstChild; /*Non va bene!*/
151                 $TAG_utente->appendChild($TAG_utente);
152
153                 if(isset($_POST['recensione'])) ($TAG_recensione = $documentoXML->createElement("recensione", $_POST['recensione']));
154                 else ($TAG_recensione = $documentoXML->createElement("recensione"));
155                 $TAG_recensione->appendChild($TAG_recensione);
156
157                 $documentoXML->save("../specificabatiCommenti.xml");
158
159                 $query="UPDATE $NomeTabellaUtente
160                 SET exp=exp+50
161                 WHERE nomeUtente = '".$_SESSION['userName']."' ";
162                 if(!$risultatoQuery = mysqli_query($CONNESSIONE, $query))
163                     {echo "Errore nell'esecuzione della query per la memorizzazione dei punti exp." . mysqli_error($CONNESSIONE);}
164
165                 echo '<meta http-equiv="refresh" content="0">'; /*Refresh pagina dopo output.*/
166             }
167
168             /*Fine invio commento.*/
169
170             /*Caso di cancellazione del commento*/
171             if(isset($_POST['Rimuovi']))
172             {
173                 if(empty($_POST['daEliminare']))
174                     {echo "<p style=\"color:red;\">Seleziona qualcosa da cancellare (se c'è&rgrave;).</p>";}
175                 else
176                 {
177                     $contenutoXML="";
178                     foreach (file("../specificabatiCommenti.xml") as $nodo) ($contenutoXML .= trim($nodo));
179                     $documentoXML = new DOMDocument(); $documentoXML->loadXML($contenutoXML);
180                     foreach($_POST['daEliminare'] as $indice=>$indiceDaEliminare)
181                     {
182                         /*$i contiene la sezioneCommentiGioco dell'ID del gioco.* / //$indiceDaEliminare-1 contiene l'indice del commento da eliminare.*/
183                         $elementi = $documentoXML->documentElement->childNodes;
184                         $objPuntato = $elementi->item($i); /*Punto alla sezioneCommentiGioco dell'ID del gioco.*/
185                         $objPuntato = $objPuntato->childNodes; /*Raccolgo i commenti.*/
186                         $objPuntato = $objPuntato->item($indiceDaEliminare);
187                         $objPuntato->setAttribute("commentoCancelled", "SI");
188                     }
189                     $documentoXML->save("../specificabatiCommenti.xml");
190                     echo '<meta http-equiv="refresh" content="0">'; /*Refresh pagina dopo output.*/
191                 }
192             }
193         }
194     }?

```

3.2.15 Libreria.php

Questo script costruisce la libreria dell'utente attraverso XML DOOM.

Ogni riga contiene l'identificativo e il nome del videogioco posseduto dall'utente.

Per ogni riga viene inoltre simulato un pulsante fittizio (ovvero che non effettua alcun tipo di funzionalità) che in teoria permetterebbe all'utente di giocare il titolo selezionato.

```

20     $contenutoXML=""; /*Inizializzo la stringa che conterrà l'XML.*/
21     foreach (file("specificadatiLibrerie.xml") as $nodo) {$contenutoXML .= trim($nodo);} /*Mettiamo il file nel contenuto.*/
22     $documentoXML = new DOMDocument(); /*Creazione del documento da usare.*/
23     $documentoXML->loadXML($contenutoXML); /*Mettiamo il contenuto nel documento.*/
24
25     if($documentoXML->schemaValidate("SchemaLibrerie.xsd")) /*Validazione del documento contenente le librerie.*/
26     {
27         $arrayLibreria=array(); $i=0;
28         $elementi = $documentoXML->documentElement->childNodes;
29         while($i<$elementi->length) /*Trovo la libreria giusta.*/
30         {
31             $objPuntato = $elementi->item($i);
32             if($_SESSION['userName']==$objPuntato->getAttribute("utenteAssociato"))
33             {
34                 $listagiochi=$objPuntato->getElementsByTagName("id");
35                 for($j=0; $j<$listagiochi->length; $j++)
36                 {/*Ricavo gli ID del gioco.*/
37                     $arrayLibreria[$j]=$listagiochi->item($j)->textContent;
38
39                     $contenutoXML2=""; /*Inizializzo la stringa che conterrà l'XML.*/
40                     foreach (file("specificadatiVideogiochi.xml") as $nodo2) {$contenutoXML2 .= trim($nodo2);} /*Mettiamo il file nel contenuto.*/
41                     $documentoXML2 = new DOMDocument(); /*Creazione del documento da usare.*/
42                     $documentoXML2->loadXML($contenutoXML2); /*Mettiamo il contenuto nel documento.*/
43
44                     if($documentoXML2->schemaValidate("SchemaGiochi.xsd"))
45                     {
46                         $elementi2 = $documentoXML2->documentElement->childNodes;
47                         for($k=0; ; $k++)
48                         {
49                             $idGioco=$elementi2->item($k)->firstChild->textContent;
50                             if($idGioco==$arrayLibreria[$j])
51                             {
52                                 $nomeGioco = $elementi2->item($k)->firstChild->nextSibling->textContent;
53                                 echo
54                                 "<tr><td style='text-align: left;' class='ciano'>". $arrayLibreria[$j]."-".$nomeGioco . "</td>
55                                 <td><form action=\"". $_SERVER['PHP_SELF']."\" method='post'>
56                                     <input type='submit' class='botone' name='Gioca' value='Gioca' />
57                                 </form> </td></tr>"; /*Stampo la libreria.*/
58                             }
59                         }
60                     }
61                 }
62             }
63             else
64             {echo"<tr><td>Errore nella validazione del documento Videogiochi.</td></tr>";}
65         }
66         break;
67     }
68     $i++;
69 }
70 else
71 {echo "<tr><td>Errore nella validazione del documento libreria.</td></tr>";}
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89

```

3.2.16 ImpostazioniUtente.php

I seguenti script servono a gestire e manipolare le impostazioni dell'utente.

Con questo script l'utente può cambiare la sua password. Dopo aver effettuato un controllo sul corretto inserimento dei dati, si effettua una query SQL al fine di avere un riscontro con la password attuale. Se il riscontro è positivo, si procede con un'altra query SQL che reimposta la password dell'utente.

```

62     <?php
63     if(isset($_POST['CambiaPassword']))
64     {
65         if(empty($_POST['oldpsw']) || empty($_POST['newpsw']) || strlen($_POST['newpsw'])>=15)
66             {echo "<span style='color:red'>Compilare tutti i campi correttamente</span>";}
67         else
68         {
69             $query="SELECT *
70             FROM $NomeTabellaUtente
71             WHERE nomeUtente = '".$_SESSION['userName']."'"
72             AND password = '".$_POST['oldpsw']."' ";
73             if (!$risultatoQuery = mysqli_query($CONNESSIONE, $query))
74                 {echo "Errore nell'esecuzione della query per la verifica della vecchia password." . mysqli_error($CONNESSIONE); exit();}
75             $riga2 = mysqli_fetch_array($risultatoQuery);
76             if($riga2)
77             {
78                 $query="UPDATE $NomeTabellaUtente
79                     SET password = '".$_POST['newpsw']."' "
80                     WHERE nomeUtente = '".$_SESSION['userName']."' ";
81                 if (!$risultatoQuery = mysqli_query($CONNESSIONE, $query))
82                     {echo "Errore nell'esecuzione della query per la memorizzazione della nuova password." . mysqli_error($CONNESSIONE); exit();}
83                 echo "<span style='color:green'>Cambio password effettuato con successo!</span>";
84             }
85             else
86             {echo "<span style='color:red'>Vecchia password non coincidente.</span>"}
87         }
88     }
89     ?>

```

Questo script attraverso un **if** e un **else** costruisce il pulsante per caricare l'immagine di profilo nel caso in cui l'utente abbia l'immagine di default, oppure un pulsante di rimozione nel caso in cui abbia una sua immagine a scelta.

```

93 <td class="ciano" style="border: 6px solid black;">
94     Immagine di profilo: <br/><img src=<?php echo "\$riga['proPic'] .\"";?> alt="Immagine di profilo" style="width:100px; height:100px; margin-left:auto; margin-right:auto; display:block;"/>
95 </td>
96 <?php
97     if($riga['proPic']=="profiles/default.png") /*Caso dell'immagine di default*/
98     {
99         echo "<form action=\"". $_SERVER['PHP_SELF']."\" method=\"post\" enctype=\"multipart/form-data\">
100             <input type=\"file\" name=\"fileToUpload\" id=\"fileToUpload\"/>
101             <input type=\"submit\" class=\"bottone\" name=\"submitUpload\" value=\"Carica .jpg\" alt=\"submit\"/>
102         </form>";
103     }
104     /*Caso dell'immagine caricata dall'utente.*/
105     else
106     {
107         echo "<form action=\"". $_SERVER['PHP_SELF']."\" method=\"post\">
108             <input type=\"submit\" class=\"bottone\" name=\"RimuoviImmagine\" value=\"Rimuovi Immagine\" alt=\"submit\"/>
109         </form>";
110     }
111 </?>
```

Se l'utente non ha ancora caricato un'immagine e quindi possiede l'immagine di default, può caricare un'immagine jpg attraverso il pulsante di upload.

In tal caso viene effettuato un controllo sul formato dell'immagine tramite **pathinfo**, e se non è .jpg viene stampato un messaggio di errore e quindi non si procede all'upload.

Altrimenti, si rinomina il file in **propic.jpg** tramite **move** e si procede all'upload secondo il path **profiles/nameofuser/propic.jpg**

Se l'upload ha successo, viene poi effettuata una query SQL che va a modificare il path di destinazione dell'immagine di profilo dell'utente.

```

112 <?php //Submit aggiunta immagine (caso dell'immagine di default).
113     if(isset($_POST['submitUpload']))
114     {
115         $directory="profiles/".$_SESSION['userName'].'/'; 
116         $file bersaglio = $directory . basename($_FILES['fileToUpload']['name']);
117         $uploadPossible=1;
118
119         /*Check sul formato del file.*/
120         $imageFileType = strtolower(pathinfo($file bersaglio, PATHINFO_EXTENSION));
121         if($imageFileType != "jpg")
122             {echo "Il file deve essere in formato jpg"; $uploadPossible=0;}
123
124
125         $file bersaglio = $directory . "propic.jpg";
126         if(move_uploaded_file($_FILES['fileToUpload']['tmp_name'], $file bersaglio) && $uploadPossible==1)
127         {
128             $query="UPDATE $NomeTabellaUtente
129                 SET propic=\"$directory/propic.jpg\"
130                 WHERE nomeutente = '".$_SESSION['userName']."' ";
131             if (!$risultatoQuery = mysqli_query($CONNESSIONE, $query))
132                 {echo "Errore nell'esecuzione della query per la memorizzazione dell'immagine di profilo." . mysqli_error($CONNESSIONE); exit();}
133             header("Refresh:0");
134         }
135         else
136             {echo "errore upload file";}
137     }
138
139 </?>
```

Se l'utente ha già caricato un'immagine, può rimuoverla attraverso l'apposito bottone.

Una volta cliccato, attraverso **unlink** si cancella l'immagine di profilo dell'utente dalla cartella del server, e tramite una query SQL si setta il path di default per l'immagine di default presente sul server.

Altrimenti vuol dire che l'utente ha una immagine di profilo memorizzata e vengono quindi mostrato un pulsante che permette di rimuoverla.

```

142 <?php //Submit rimozione Immagine (caso dell'immagine caricata dall'utente).
143     if(isset($_POST['RimuoviImmagine']))
144     {
145         unlink($riga['proPic']);
146         $query="UPDATE $NomeTabellaUtente
147             SET propic = 'profiles/default.png'
148             WHERE nomeutente = '".$_SESSION['userName']."' ";
149         if (!$risultatoQuery = mysqli_query($CONNESSIONE, $query))
150             {echo "Errore nell'esecuzione della query per la cancellazione della propic." . mysqli_error($CONNESSIONE); exit();}
151         header("Refresh:0");
152     }
153 </?>
```

All'inizio ho avuto fatica nel trovare un modo per gestire l'immagine di profilo di default. Stavo pensando di caricarla in ogni cartella degli utenti quando questi rimanevano la propria immagine di profilo, ma alla fine caricarne solo una per tutti gli utenti e collegarla con SQL è stata sicuramente una soluzione più facile e meno dispendiosa.

Se l'utente non possiede una carta di credito (fattore identificato dal numero della carta settato su NULL), viene incluso il file `FormCarta.php`, che aggiunge semplicemente del codice HTML che crea il form dati da inserire per memorizzare i dati della carta di credito.

```

171 <?php
172     if($riga['numeroCarta']=='NULL')
173         {include 'FormCarta.php';}
174     else
175     {
176         echo "
177             <tr><td class=\"ciano\">Nome proprietario: ". $riga['nomePro'] . "</td></tr>
178             <tr><td class=\"ciano\">Numero carta: ". $riga['numeroCarta'] . "</td></tr>
179             <tr><td class=\"ciano\">Cod Sicurezza: ". $riga['codSic'] . "</td></tr>
180             <tr><td class=\"ciano\">Data scadenza: ". $riga['datascad'] . "</td></tr>
181             <tr><td>
182                 <form action="" . $_SERVER['PHP_SELF'] . "\" method="post">
183                     <input type="submit" class="bottone" name="CancellaCarta" value="Cancella carta" alt="submit"/>
184                 </form>
185             </td></tr>";
186     }
187 }
188 ?>
```

Se l'utente possiede una carta memorizzata, può cancellarla con l'apposito bottone. In tal caso il seguente script attraverso una query SQL pone a NULL tutti i valori relativi alla carta di credito di quell'utente.

```

15 <?php
16     if(isset($_POST['CancellaCarta']))
17     {
18         $query="UPDATE $NomeTabellaUtente
19             SET nomePro = 'NULL',
20                 numeroCarta = 'NULL',
21                 codSic = 'NULL',
22                 dataScad = 'NULL'
23                 WHERE nomeUtente = '$_SESSION['username']]'";
24         if (!$risultatoQuery = mysqli_query($CONNESSIONE, $query)) /*stampa un messaggio di errore e termino lo script*/
25             {echo "Errore nell'esecuzione della query per la dememorizzazione della carta." . mysqli_error($CONNESSIONE); exit();}
26         echo "</tr><td class='ciano'>Carta cancellata</td></tr>";
27         header("Refresh:0");
28     }?
29 
```

Questo invece è uno script che permette il corretto inserimento dei dati nella form per l'inserimento della carta di credito.

Se i dati sono stati scritti correttamente, si procede al loro inserimento tramite query SQL.

```

28 <?php
29     if(isset($_POST['invio']))
30     {
31         if(empty($_POST['nomeProp']) || empty($_POST['numeroCarta']) || empty($_POST['sicCode']) || empty($_POST['datascad']))
32             {echo "<tr><td class='ciano'> Compilare tutti i campi </td></tr>"}
33         else if(strlen($_POST['nomeProp'])>300 || strlen($_POST['numeroCarta'])!=12 || strlen($_POST['sicCode'])!=3)
34             {echo "</tr><td class='ciano'>Compila i campi correttamente.</td></tr>"}
35         else
36         {
37             $query="UPDATE $NomeTabellaUtente
38                 SET nomePro = '".$_POST['nomeProp']."',
39                     numeroCarta = '".$_POST['numeroCarta']."' ,
40                     codSic = '".$_POST['sicCode']."' ,
41                     dataScad = '".$_POST['datascad']."' ,
42                     WHERE nomeUtente = '$_SESSION['userName']]'";
43             if (!$risultatoQuery = mysqli_query($CONNESSIONE, $query))
44                 {echo "Errore nell'esecuzione della query per la memorizzazione della carta." . mysqli_error($CONNESSIONE); exit();}
45             header("Refresh:0");
46         }
47     }
48 ?>
```

3.2.17 InserimentoGioco.php

Questo script crea un bottone che porta l'utente alla pagina precedente nel caso in cui questo non sia un gestore. Altrimenti include il file `InserimentoGiocoForm.php` che crea il form per l'inserimento dei dati che permettono l'aggiunta del videogioco al negozio.

```

4   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
5   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6   <head>
7       <title> Millennium Games </title> <!--Titolo della pagina.-->
8       <link rel="stylesheet" type="text/css" href="main.css" /> <!--Style principale.-->
9       <link rel="icon" href="materials/img/iconaSito.png" /> <!--Icona del sito-->
10      <script>
11          function vaiindietro() {window.history.back();}
12      </script>
13  </head>
14  <body>
15  <?php include 'toolbar.php'?>
16  <?php include 'QueryUtente.php'; ?>
17  <?php
18      if($riga['GEST']!='N')
19      {
20          echo "<table class='backgroundTabella'>
21              <tr><td class='ciano'>
22                  Autorizzazione non consentita. Clicca sul pulsante per tornare indietro.<br>
23                  <button class='bottone' onclick='vaiindietro()'> Vai. </button>
24              </td></tr>
25          </table>";
26      }
27      else {include 'InserimentoGiocoForm.php';}
28  ?>
29  </body>
30  </html>
```

3.2.18 InserimentoGiocoForm.php

Questo script crea il form da compilare per l'inserimento del videogioco. I campi compilati rimangono compilati dopo l'aver premuto il bottone di submit (anche in caso di errore o campi non compilati del tutto o non correttamente), attraverso l'utilizzo di un `if(isset())` e un `echo` nel `value` di ogni form.

Una volta premuto il tasto `Aggiungi Gioco` si effettua prima di tutto un controllo sul corretto inserimento dei dati.

In caso ci siano dei campi non compilati (verifica effettuata su ognuno di essi tramite `isempty`), si stampa un messaggio di errore in cui si invita il gestore a compilare tali campi.

Passata questa prova, prima di tutto si crea la cartella avente nome del gioco (se non esiste già, altrimenti vuol dire che è già presente un gioco con l'ID inserito e quindi si termina l'esecuzione dello script tramite `exit` senza caricare il gioco nel file XML).

Poi si inseriscono i dati del videogioco tramite XML DOM, attraverso l'utilizzo di `appendChild()`.

In particolare, se sono state caricate delle immagini si fa un controllo sul loro formato (che deve essere `.jpg`) e in caso di riscontro positivo si rinominano e memorizzano le immagini nella cartella di gioco prima creata.

Ho trovato difficoltà nel gestire l'upload di più immagine. Alla fine ho scoperto che si trattava solo di gestire le stesse operazioni per ogni elemento dell'array che memorizza le immagini.

Inoltre non sapevo come "far ricordare" il contenuto del form compilato anche dopo il posto, quindi ho risolto con un if e un echo che rimettessero il valore inserito nel value dei vari form.

```

1 <table style="border:3px solid black; background-color: #00if3f; margin: 0 auto;">
2   <tr><td class="ciano" style="border: 0px solid black;">
3     <p style="color:#0074D9;">Compilare tutti i seguenti campi per aggiungere il videogioco.</p>
4     <form action=<?php $_SERVER['PHP_SELF']?>" method="post" enctype="multipart/form-data">
5       <span>ID </span><input type="text" name="IDgioco" size="30" value=<?php if(isset($_POST['IDgioco'])) {echo htmlentities($_POST['IDgioco']);} ?>/><br />
6       <span>Titolo </span><input type="text" name="NOMEgioco" size="30" value=<?php if(isset($_POST['NOMEgioco'])) {echo htmlentities($_POST['NOMEgioco']);} ?>/><br />
7       <span>Genere </span><input type="text" name="GENEREGioco" size="30" value=<?php if(isset($_POST['GENEREGioco'])) {echo htmlentities($_POST['GENEREGioco']);} ?>/><br />
8       <span>Descrizione </span><br /><textarea rows=3 cols=60 name=DESCRIZIONEgioco ><?php if(isset($_POST['DESCRIZIONEgioco'])) {echo htmlentities($_POST['DESCRIZIONEgioco']);} ?></textarea><br />
9       <span>Prezzo </span><input type="text" name="PREZZOgioco" size="30" value=<?php if(isset($_POST['PREZZOgioco'])) {echo htmlentities($_POST['PREZZOgioco']);} ?>/><br />
10      <span>Disponibilit&agrave; </span><input type="text" name="DISPOgioco" size="30" value=<?php if(isset($_POST['DISPOgioco'])) {echo htmlentities($_POST['DISPOgioco']);} ?>/><br />
11      <p style="color:#0074D9;">Inserire da 0 a 4 immagini .jpg</p>
12      <input type="file" name="fileToUpload" id="fileToUpload" multiple/>
13      <input type="submit" name="aggGioco" alt="submit" value="Aggiungi Gioco"/>
14    </form>
15  </td></tr>
16 </table>
17
18 <?php
19
20
21 if(isset($_POST['aggGioco']))
22 {
23   if(empty($_POST['IDgioco']) && empty($_POST['NOMEgioco']) && empty($_POST['GENEREGioco']) && empty($_POST['DESCRIZIONEgioco']) && empty($_POST['PREZZOgioco']) && empty($_POST['DISPOgioco']))
24   {
25     $dir = "games/".$_POST['IDgioco'];
26     if(file_exists($dir))
27     {
28       echo "<p style='color:red;'> L'id esiste già; sceglierne un altro.</p>";
29       exit();
30     }
31     mkdir($dir, 0700, true);
32
33     $contenutoXML="";
34     foreach (file("SpecificaDatiVideogiochi.xml") as $nodo)
35     {
36       $contenutoXML .= trim($nodo);
37     }
38     $documentoXML = new DOMDocument();
39     $documentoXML->loadXML($contenutoXML);
40
41     $TAG_Videogioco = $documentoXML->createElement("Videogioco");
42     $TAG_IDgioco = $documentoXML->createElement("idgioco", $_POST['IDgioco']);
43     $TAG_NOMEgioco = $documentoXML->createElement("nomegioco", $_POST['NOMEgioco']);
44     $TAG_GENEREGioco = $documentoXML->createElement("genere", $_POST['GENEREGioco']);
45     $TAG_DESCRIZIONEgioco = $documentoXML->createElement("descrizione", $_POST['DESCRIZIONEgioco']);
46     $TAG_PREZZOgioco = $documentoXML->createElement("prezzo", $_POST['PREZZOgioco']);
47     $TAG_DISPOGioco = $documentoXML->createElement("dispo", $_POST['DISPOgioco']);
48
49     $radice = $documentoXML->getElementsByTagName("VIDEOGIOCHI")->item(0);
50     $radice->appendChild($TAG_Videogioco);
51     $TAG_Videogioco->appendChild($TAG_IDgioco);
52     $TAG_Videogioco->appendChild($TAG_NOMEgioco);
53     $TAG_Videogioco->appendChild($TAG_GENEREGioco);
54     $TAG_Videogioco->appendChild($TAG_DESCRIZIONEgioco);
55     $TAG_Videogioco->appendChild($TAG_PREZZOgioco);
56
57     if(is_uploaded_file($_FILES['fileToUpload']['tmp_name'][0])) /*Caso di caricamento dei file */
58     {
59       $nfile = count($_FILES['fileToUpload']['name']);
60       echo $nfile;
61       $uploadPossibile=1;
62       for($i=0; $i<$nfile; $i++)
63       {
64         $file_bersaglio = $dir . "/" . basename($_FILES['fileToUpload']['name'][$i]);
65         $imageFileType = strtolower(pathinfo($file_bersaglio, PATHINFO_EXTENSION));
66         if($imageFileType != "jpg")
67           {echo "Il file deve essere in formato jpg. Caricamento delle immagini non effettuato"; $uploadPossibile=0; $i=$nfile;}
68         $i++; $file_bersaglio = $dir . "/sc". $i ."jpg"; $i--;
69
70         /*Se tutto è andato a buon fine, memorizzo le immagini */
71         if(move_uploaded_file($_FILES['fileToUpload']['tmp_name'][$i], $file_bersaglio) && $uploadPossibile==1)
72         {
73           $i++;
74           $TAG_img =$documentoXML->createElement("img","sc". $i ."jpg");
75           $i--;
76           $TAG_Videogioco->appendChild($TAG_img);
77         }
78         else{echo "errore upload file numero ".$i;}
79       }
80
81       $TAG_Videogioco->appendChild($TAG_DISPOGioco);
82       $documentoXML->save("SpecificadatiVideogiochi.xml");
83       echo "<p style='color:green;'> Gioco inserito con successo!</p>";
84     }
85     else
86     {
87       $error = "";
88       $error .= "<p style='color:red;'>Compilare i seguenti campi:<br />";
89       if(empty($_POST['IDgioco'])) {$error .= "ID del gioco<br />"}
90       if(empty($_POST['NOMEgioco'])) {$error .= "Nome del gioco<br />"}
91       if(empty($_POST['GENEREGioco'])) {$error .= "Genere del gioco<br />"}
92       if(empty($_POST['DESCRIZIONEgioco'])) {$error .= "Descrizione del gioco <br />"}
93       if(empty($_POST['PREZZOgioco'])) {$error .= "Prezzo del gioco <br />"}
94       if(empty($_POST['DISPOgioco'])) {$error .= "Data di disponibilità del gioco <br />"}
95       $error .= "</p>";
96       echo $error;
97     }
98   }
99 }

```

4 Capitolo 4 - Elenco.

Qui di seguito sono elencati gli utenti che popolano il database del sistema, assieme ai loro dati.

| | → | ▼ nomeUtente | password | proPic | BAN | GEST | ADM | portafoglio | expe | nomePro | numeroCarta | codSic |
|----------|-------|--------------|---------------|---------------|-------------------------------|------|-----|-------------|--------|---------|-----------------------|--------------|
| Modifica | Copia | Elimina | Arturo | Arturo | profiles/Arturo/propic.jpg | Y | N | Y | 660.81 | 598 | NULL | NULL |
| Modifica | Copia | Elimina | Bramante | Bramante | profiles/default.png | N | N | N | 0 | 0 | NULL | NULL |
| Modifica | Copia | Elimina | CosmoVellucci | CosmoVellucci | profiles/default.png | N | Y | N | 30 | 100 | Cosmo Vellucci | 000000000000 |
| Modifica | Copia | Elimina | Manigoldo | Manigoldo | profiles/Manigoldo/propic.jpg | Y | N | N | 0 | 100 | Manigoldo Manigoldini | 123456789012 |

Il database è chiamato lweb33, la tabella è chiamata datutentiregistrati.