

Università di Bologna - Campus di Cesena  
Ingegneria e Scienze Informatiche (8615)

## **Relazione per “Tecnologie Web”**

### **VinylsShop**

0001077417 - Alex Mazzoni  
alex.mazzoni3@studio.unibo.it

0001097772 - Samuele Casadei  
samuele.casadei3@studio.unibo.it

# Indice

|          |                                |           |
|----------|--------------------------------|-----------|
| <b>1</b> | <b>Introduzione</b>            | <b>2</b>  |
| 1.1      | E-commerce di vinili . . . . . | 2         |
| <b>2</b> | <b>Design e Mockups</b>        | <b>3</b>  |
| 2.1      | Everyone . . . . .             | 3         |
| 2.1.1    | Home . . . . .                 | 3         |
| 2.1.2    | Search . . . . .               | 4         |
| 2.1.3    | Cart . . . . .                 | 8         |
| 2.1.4    | User Login / Signup . . . . .  | 10        |
| 2.2      | User only . . . . .            | 12        |
| 2.2.1    | User Info . . . . .            | 12        |
| 2.2.2    | Checkout . . . . .             | 14        |
| 2.3      | SuperUser . . . . .            | 16        |
| 2.3.1    | Dashboard . . . . .            | 16        |
| <b>3</b> | <b>Struttura del progetto</b>  | <b>17</b> |
| 3.1      | Config . . . . .               | 17        |
| 3.2      | Core e Utility . . . . .       | 17        |
| 3.3      | Design . . . . .               | 18        |
| 3.3.1    | Controllers . . . . .          | 18        |
| 3.3.2    | Models . . . . .               | 18        |
| 3.3.3    | Views . . . . .                | 18        |
| 3.4      | Resources (Frontend) . . . . . | 19        |
| <b>4</b> | <b>Conclusioni</b>             | <b>20</b> |
| 4.1      | Effetti WOW . . . . .          | 20        |

# 1 Introduzione

## 1.1 E-commerce di vinili

L'obiettivo del progetto di Tecnologie Web è di realizzare un sito di e-commerce relativo ad un bene/servizio a propria discrezione, noi abbiamo scelto i vinili. VinylsShop è un e-commerce che riserva le seguenti funzioni agli utenti (da suddividere tra **utenti ordinari** e **amministratori**):

Utenti ordinari:

- sfogliare il catalogo
- cercare i prodotti
- gestione del carrello
- effettuazione ordini
- monitoraggio dello stato degli ordini
- gestione dei metodi di pagamento
- gestione degli indirizzi di spedizione
- leggere (spuntare/eliminare) le notifiche

Utenti amministratori:

- aggiungere (modificare) prodotti
- gestione utenti
- aggiungere nuovi amministratori
- monitoraggio dello stato degli ordini di un utente

## 2 Design e Mockups

Abbiamo deciso di strutturare il progetto, rendendolo appunto il più trasparente possibile, si può infatti ben osservare che nella struttura MVC da noi utilizzata esiste un controller per attore.

In questo modo ci si può subito focalizzare sui privilegi astratti di autorizzazioni necessarie per ogni controller, astraendo il concetto di autorizzazione nelle tre principali tipologie di attori:

- **Everyone:** nessuna autorizzazione.  
Tutti sono in grado di richiedere e visualizzare il contenuto di queste richieste.
- **User only:** privilegio di utente loggato.  
Solamente un utente loggato è in grado di accedere e visualizzare il contenuto della richiesta.
- **Admin o Super user:** privilegio del venditore.  
È in grado di visualizzare le pagine di gestione del sito (dashboard).

### 2.1 Everyone

Ogni utente, con o senza autorizzazione, può: accedere alla **Home** del sito, utilizzare la **Ricerca**, aggiungere prodotti al suo **Carrello** (rimarranno salvati solamente nella sessione) ed effettuare **Login** o **Iscrizione**.

#### 2.1.1 Home

La Home fornisce, tramite una **Navbar** la possibilità di spostarsi tra le sezioni (Ricerca, Carrello e Utente) e mostra un **carosello** di anteprime di vinili clickabili. La Home è volutamente spoglia di contenuti per via della forte suddivisione che abbiamo applicato sulle sezioni, lasciando che la visita dell'utente sia libera da linee guida.



Figura 2.1: mockup per la Homepage

### 2.1.2 Search

La Ricerca si sovrappone a tutte le altre sezioni, è possibile aprire la barra di ricerca da dovunque per poter consultare il catalogo di prodotti, per poi chiuderla e visualizzare nuovamente il contenuto precedente. La ricerca di contenuti può essere eseguita secondo diversi filtri, è infatti possibile ricercare un vinile per il nome di un brano, il nome dell'artista o il titolo dell'album. La ricerca strutturata in questo modo fa risultare la navigazione molto più veloce e dinamica.



Figura 2.2: mockup per la Ricerca

## Viny

Clickando su uno dei risultati della ricerca si accede alla pagina di dettaglio del vinile scelto. Da qui è possibile consultare sia le informazioni tecniche del vinile (giri, misura e tipo) che le informazioni sui brani contenuti e l'artista. È possibile aggiungere il vinile al carrello sia dai risultati della ricerca, che dalla pagina di dettaglio; inoltre, se il vinile risulta interessante, a pié di pagina è possibile consultare i vinili suggeriti.

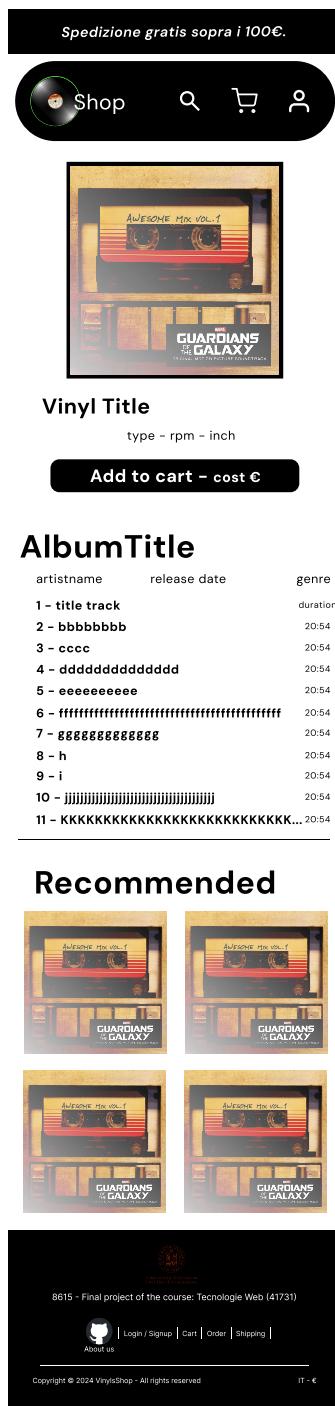


Figura 2.3: mockup per la pagina in dettaglio del vinile

### **2.1.3 Cart**

Nel carrello è possibile consultare i vinili aggiunti tramite il catalogo, aumentarne o diminuirne la quantità a piacimento (nei limiti imposti dalla quantità detenuta del vinile in questione o nei limiti fisici). Gli utenti privi di autorizzazione non possono procedere al checkout del carrello.

Nel carrello è mostrato il prezzo totale degli articoli.

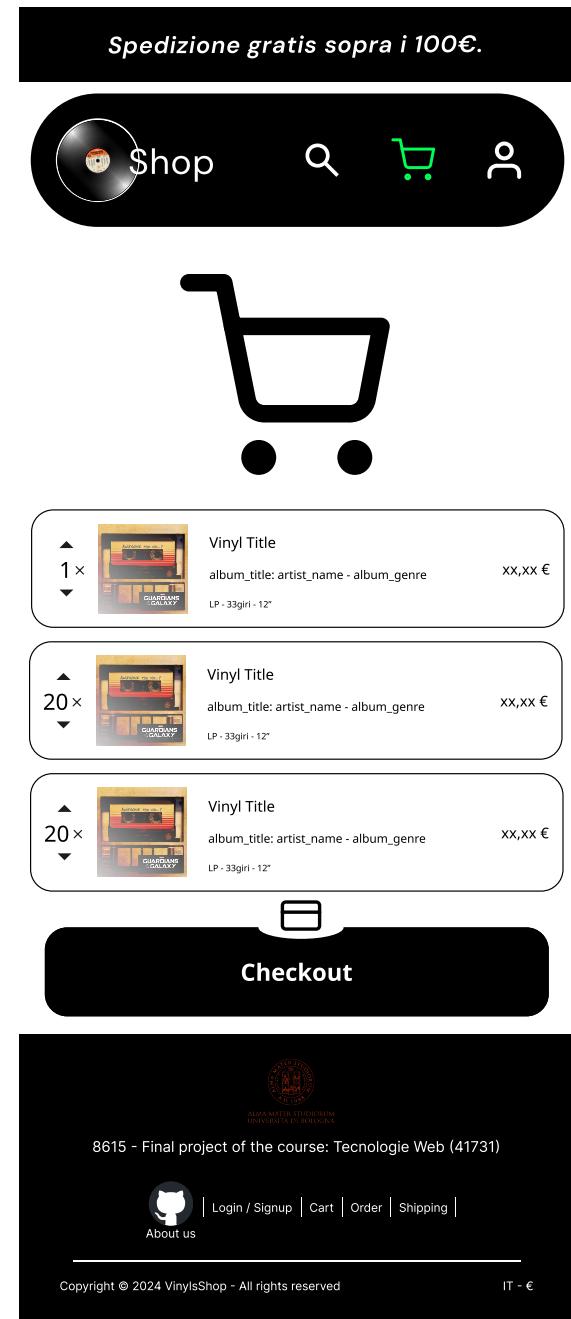


Figura 2.4: mockup per il carrello

## **2.1.4 User Login / Signup**

La pagina di Login/Registrazione è il corrispondente di una sezione utente per un utente senza autorizzazione. È costituita da due semplici form da compilare, uno è per creare un nuovo account, l'altro è per utilizzarne uno già esistente.



## Login / Signup

Mail

Password

Stay signed



Figura 2.5: mockup for login / signup page

## 2.2 User only

Oltre tutti i privilegi disponibili per gli utenti non autorizzati, gli utenti ordinari godono di ulteriori privilegi: effettuare acquisti e gestire i dati del proprio profilo.

### 2.2.1 User Info

Tra le proprie informazioni, l'utente, può gestire le due principali: il **Metodo di pagamento** e l'**Indirizzo di spedizione**.

L'utente può creare nuove informazioni, modificare o eliminare quelle già esistenti e impostare (opzionalmente) un'informazione di ciascuna categoria che verrà quindi applicata come default; ad esempio, può impostare un metodo di pagamento di default che verrà applicato ad ogni acquisto.

Infine l'utente potrà consultare lo stato degli ordini già effettuati, con relativa progressione e informazioni sulla spedizione.

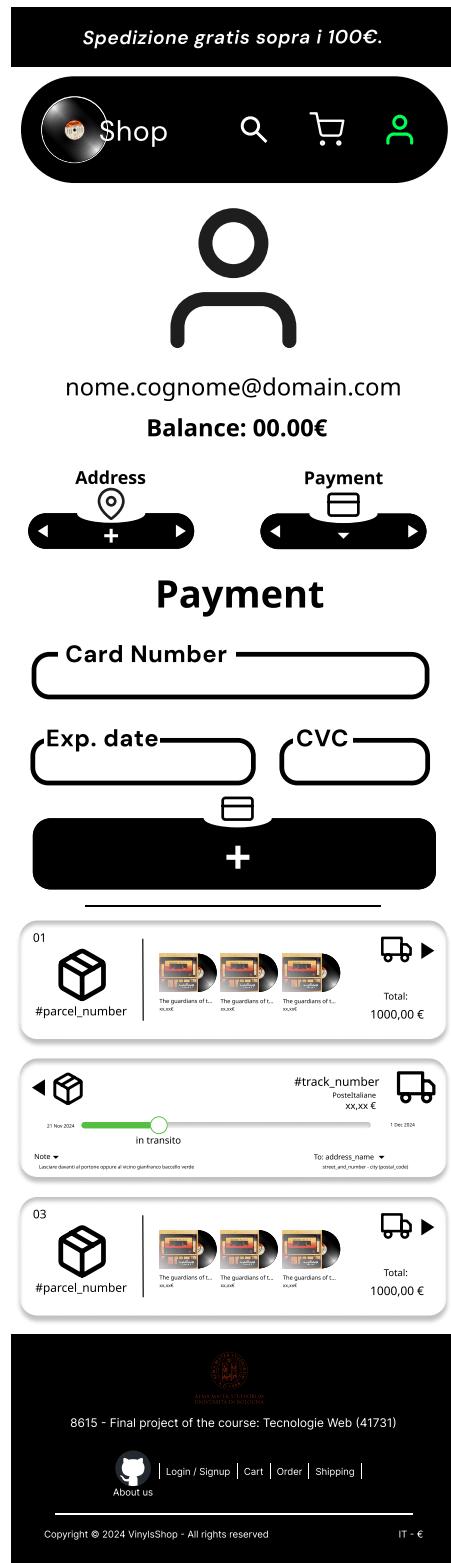


Figura 2.6: mockup per la pagina utente  
15

### **2.2.2 Checkout**

Quando l'utente conferma un ordine, entra nella pagina di checkout, dove visualizza un riepilogo del carrello e l'opzione di acquisto. Una volta effettuato l'acquisto l'utente viene reindirizzato alla Home.



## Checkout

**Address** address\_name

**Payment** Balance: xx €

|      |  |         |
|------|--|---------|
| 1 ×  | Vinyl Title<br>album_title: artist_name - album_genre<br>LP - 33giri - 12" | xx,xx € |
| 20 × | Vinyl Title<br>album_title: artist_name - album_genre<br>LP - 33giri - 12" | xx,xx € |

vinyls\_total: xx,xx € 22% (incl. IVAs)  
shipping\_fee: xx,xx €  
discount\_code: -xx,xx € -20%

---

**Total:** xx,xx €

**on Payment:** Balance: xx,xx €



Figura 2.7: mockup for checkout page

## 2.3 SuperUser

Ogni amministratore ha accesso ad una dashboard divisa in tre sezioni principali: **Vinyls, Users, Shippings**. In Vinyls, l'admin potrà modificare vinili esistenti o aggiungerne di nuovi, in Users potrà designare degli utenti come admin e monitorare gli ordini di altri utenti; in Shippings, l'admin, potrà modificare il costo della spedizione e la soglia per raggiungere la spedizione gratuita.

### 2.3.1 Dashboard

## 3 Struttura del progetto

La struttura del progetto è nata per affrontare il progetto in modo più DRY possibile, in grado di focalizzarci sulle qualità:

- performance
- accessibilità
- quality of life
- installazione

Il progetto ha come obiettivo di fornire all'utente un alta semplicità d'uso e interazione utente-browser in modo di semplificare gli inserimenti e la navigazione, affidandoci agli standard XHTML e WCAG AA, abbiamo utilizzato vari tool per controllarne la qualità, come ad esempio 'lightroom'.

### 3.1 Config

Il progetto parte dall'index.php che include tutti i file di configurazione e utility che il server necessita per il corretto funzionamento. Docker è responsabile della configurazione e l'avvio del progetto in maniera semplice e solida esso infatti è responsabile della qualità di installazione, dato che con una solo riga di comando è possibile avere il progetto completamente funzionante su una qualsiasi macchina. Abbiamo usato le variabili di ambiente semplificandone l'accesso e il funzionamento incorporando ad essa un 'LoadEnvs' in grado di leggere le variabili da ogni file .env, specificato nell'index. Esse infatti sono in grado di garantire un elevato grado di flessibilità anche per la connessione del database, dato che il progetto è stato sviluppato sia in ambiente docker sia in ambiente XAMPP. Grazie a queste viene infatti creato un utente aggiuntivo per interrogare il Database con solo i privilegi di INSERT, SELECT, DELETE, UPDATE.

### 3.2 Core e Utility

Il core è ciò che definiamo l'engine del server, tutto ciò che gli serve per inizializzare le sue fondamenta.

- Router: è responsabile di definire tutti i percorsi e le richieste in grado di soddisfare.

- Controller: È una super classe di controller, in grado di generalizzare la renderizzazione delle pagine dai diversi controller.
- Request e Response: sono responsabili di gestire le richieste e le risposte tra client e server.

Le utility sono quegli script creati come singleton in grado di essere usati da qualsiasi script in qualsiasi momento:

- Database: permette l'interrogazione al database evitando SQL injection e codice DRY.
- Session: definisce tutte le funzioni utili riguardanti le variabili di sessione.
- Routing: definisce e documenta ogni percorso e che i controller possono gestire.
- Render: pagina in comune a tutte definisce gli elementi principali come navbar, footer.

## 3.3 Design

Abbiamo scelto un design MVC (Model-View-Controller):

### 3.3.1 Controllers

Ogni funzione si riferisce ad una specifica richiesta che esso è in grado di soddisfare, occupandosi di parsare i dati inviati nella richiesta e richiamando le varie funzioni definite nei modelli logici, controllando anche le varie autorizzazioni e privilegi che l'attore ha.

### 3.3.2 Models

Ogni model si occupa di interrogare il database fornendo i dati e le funzioni utili per manipolare il suo contenuto.

### 3.3.3 Views

Si riferisce ad ogni componente o pagina che il frontend utilizza per visualizzare tutti i dati in maniera strutturata e accessibile.

## **Pages**

Suddivise per privilegi, sono quelle che il sito fornisce quando deve fornire una pagina.

## **Components**

È codice riusabile utilizzato da sicuramente una o più pagine, in questo modo non dobbiamo ripetere codice simile.

## **3.4 Resources (Frontend)**

Per i motivi sopracitati abbiamo evitato l'uso di bootstrap, non per reinventare la ruota, ma per un discorso di performance e facilità d'uso. Abbiamo fatto uso in JavaScript uso di richieste Ajax, per fornire un interazione diretta e veloce all'utente, dato che siamo consapevoli che ai giorni d'oggi, se non si riceve risposta per più di un secondo l'utente perde interesse.

## 4 Conclusioni

La parte più difficile del progetto è stata sicuramente pensare alle caratteristiche e funzionalità del progetto in fase di creazione dei mockups, cercando anche di rientrare nei tempi previsti.

Una delle fasi che ci ha portato via più tempo è stata la fase di configurazione, per rendere il progetto flessibile sia in ambiente XAMPP sia docker.

### 4.1 Effetti WOW

- DarkMode (con utilizzo di variabili per i colori)
- Notifiche
- Carrello in sessione
- Progressione fittizia della spedizione
- Password e cookie crittografati
- Richieste AJAX