FitCourse

Progetto tecnologie web

Valentyn Blaha | Syed Jawad Haider

# Sommario

[Sommario 1](#_Toc143873898)

[Introduzione 2](#_Toc143873899)

[Descrizione del progetto 3](#_Toc143873900)

[Use case diagram 3](#_Toc143873901)

[Diagramma di classi del database 4](#_Toc143873902)

[Activity diagram – sistema del carrello 5](#_Toc143873903)

[Tecnologie usate 5](#_Toc143873904)

[API 5](#_Toc143873905)

[Autenticazione 7](#_Toc143873906)

[Organizzazione logica dell’applicazione 7](#_Toc143873907)

[Recommendation system 8](#_Toc143873908)

[Descrizione dei test effettuati 10](#_Toc143873909)

[Ecommerce Api 10](#_Toc143873910)

[Course Api 11](#_Toc143873911)

[Risultati dell’applicazione 11](#_Toc143873912)

[Pagina di login 11](#_Toc143873913)

[Pagine di registrazione 11](#_Toc143873914)

[Homepage 12](#_Toc143873915)

[Pagina di elenco dei corsi 13](#_Toc143873916)

[Dettaglio del corso 14](#_Toc143873917)

[Pagina di elenco dei prodotti 14](#_Toc143873918)

[Pagina del dettaglio di un prodotto 15](#_Toc143873919)

[Pagina del carrello e del checkout 15](#_Toc143873920)

# Introduzione

FitCourse è una piattaforma e-commerce, adibita anche all’iscrizione ai corsi, che permette a chi si iscrive di fare acquisti di prodotti e iscriversi ai corsi a tema fitness o attività fisica in generale.

Il sito ammette tre tipologie di utenti (oltre agli amministratori):

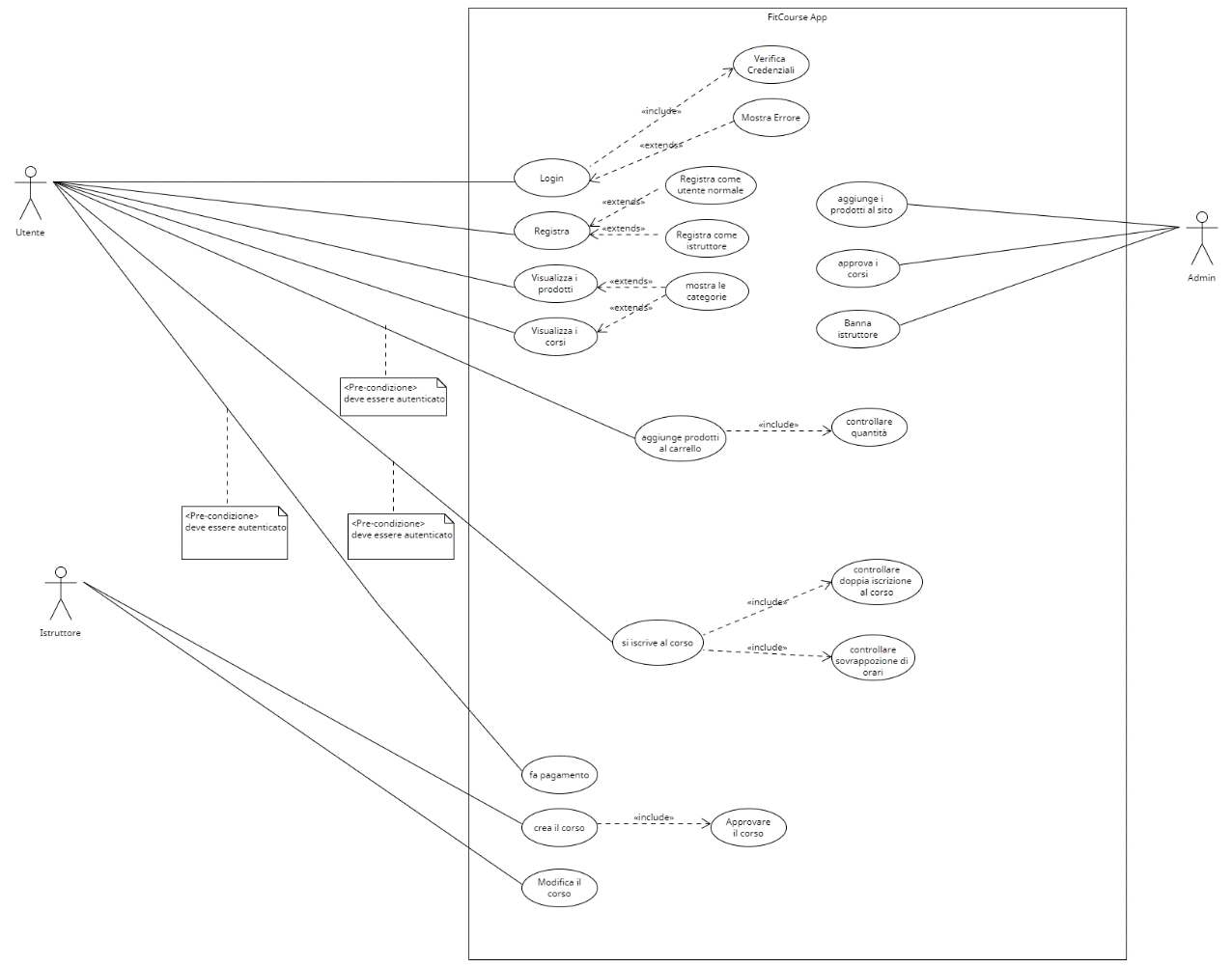
* Anonimi
* Clienti
* Istruttori

Le funzionalità di cui può usufruire ogni categoria sono espresse in quanto segue:

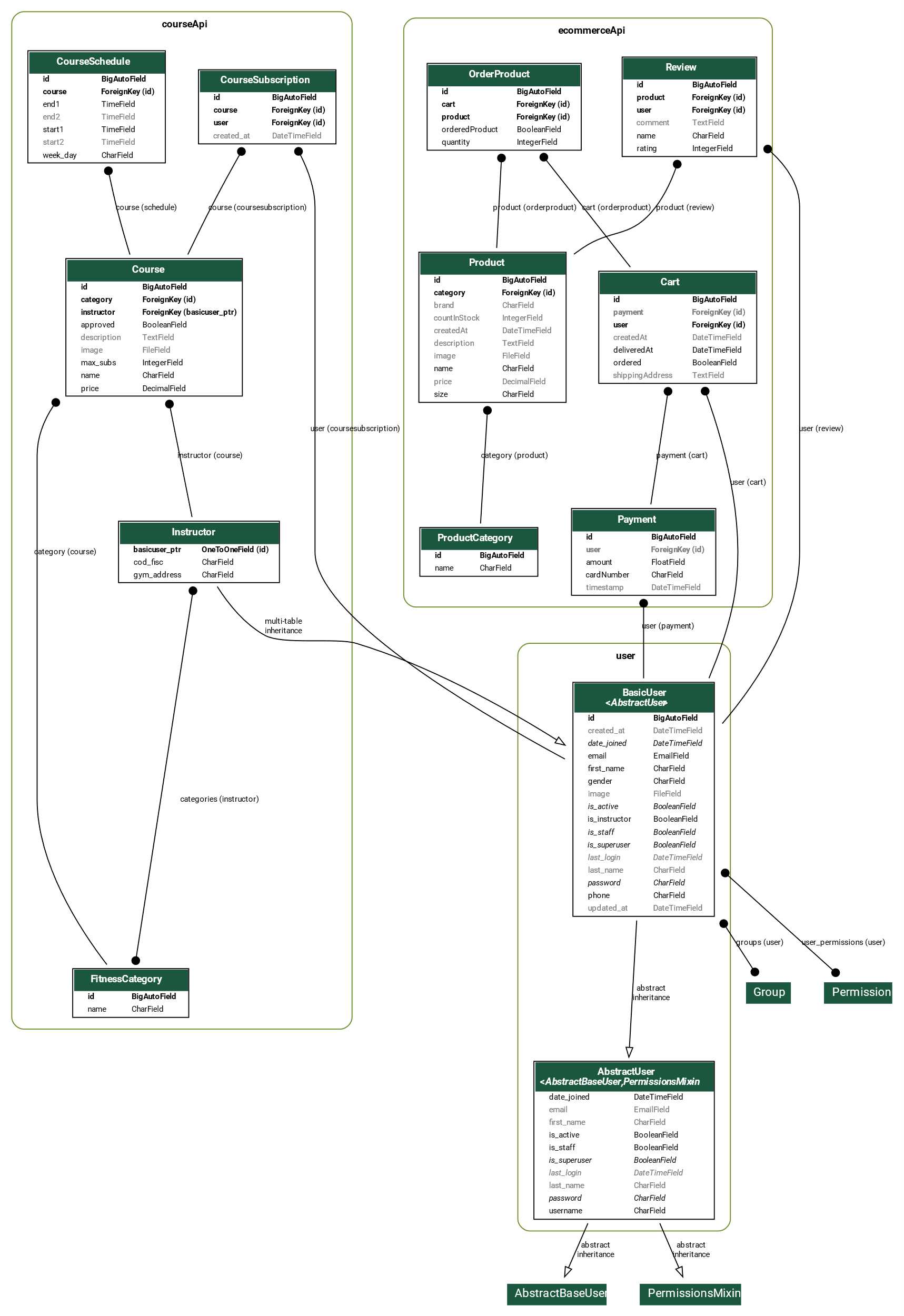
* Gli utenti anonimi possono navigare il sito e visualizzare i corsi e i prodotti disponibili, ma senza la possibilità di acquistarli (cosa per cui è necessaria la registrazione)
* Gli utenti possono registrarsi come clienti o come allenatori
* Sia i clienti che gli allenatori (registrati) possono visualizzare e acquistare corsi o prodotti
* Ogni utente registrato da cliente o da allenatore può iscriversi ad un corso e acquistare prodotti a cui può rilasciare una recensione (con punteggio)
* Ogni utente che acquista i prodotti ha la possibilità di inserirli in un carrello per poter effettuare l’acquisto di più prodotti in una sola volta
* Ogni utente registrato da allenatore deve specificare uno o più campi di competenza e la palestra in cui lavora, inoltre può creare corsi (aderenti ad uno dei sui campi) che dovranno essere approvati dall’amministratore del sito.
* Ogni corso avrà un prezzo mensile, un determinato calendario con giorni e orari settimanali e un numero massimo di iscritti
* Quando un utente si iscrive ad un corso, un sistema avvisa il cliente nel caso in cui gli orari del corso a cui si vuole iscrivere si sovrappongono con gli orari di uno o più corsi a cui è già iscritto, dandogli la possibilità di proseguire con l’iscrizione o annullarla
* L’amministratore del sito, oltre ai massimi poteri, ha anche la possibilità di aggiungere i prodotti che possono essere acquistati dagli utenti registrati
* Il sistema è dotato di un meccanismo di ricerca che permetta di selezionare i prodotti in base a diverse caratteristiche (prezzo, categoria e caratteristiche offerte, ...) e di un recommendation system basato sulle recensioni, in base alla similarità del profilo degli acquisti dell’acquirente

# Descrizione del progetto

## Use case diagram

Qui sotto è riportato il diagramma use case che riassume le funzionalità primarie e secondarie di cui dispongono gli utenti del sito

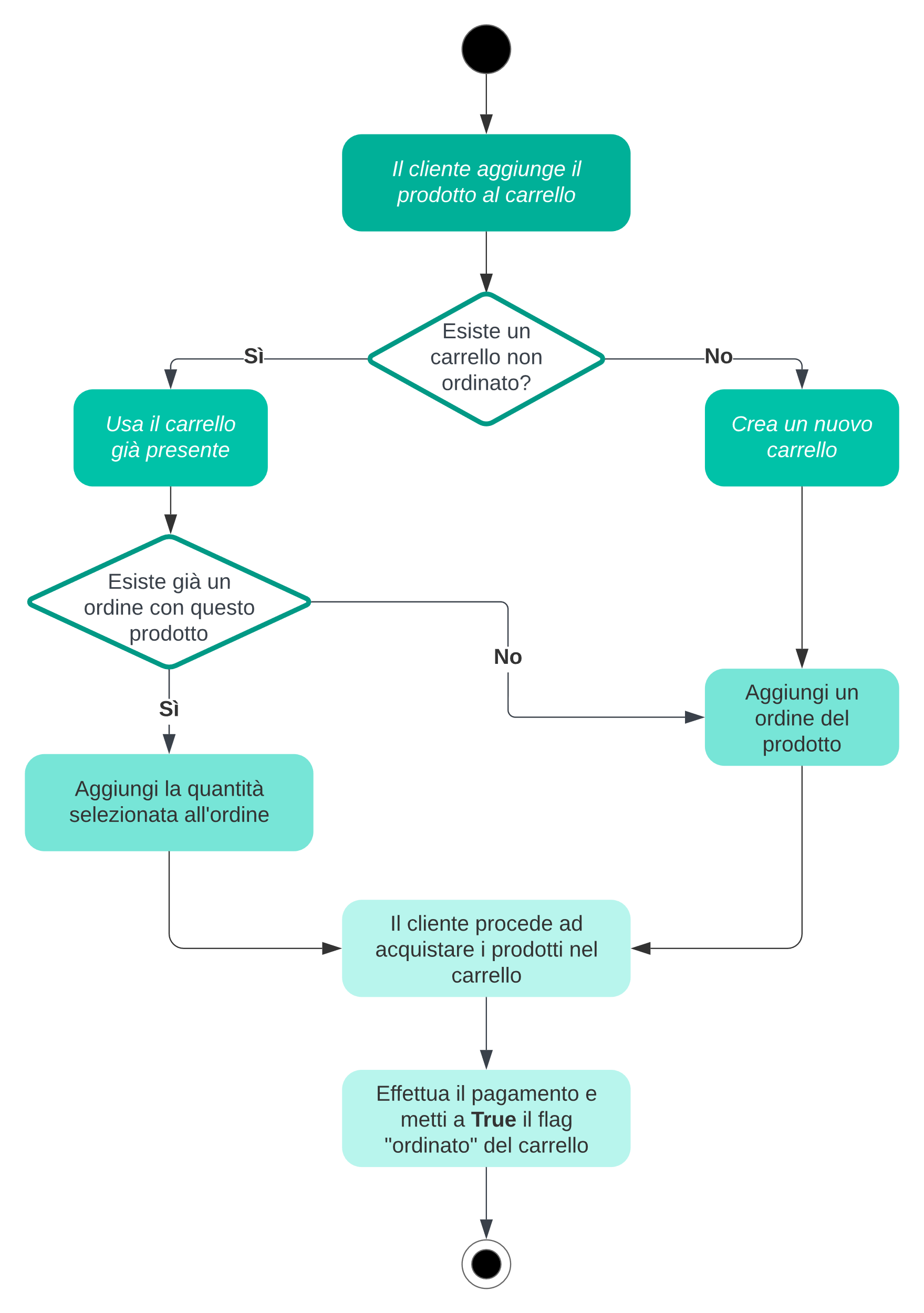
## Diagramma di classi del database

Qui sotto è riportato il diagramma di classi dello schema del database utilizzato dall’applicazione

NOTA: il diagramma non include parti del database istanziate automaticamente da Django che sono necessarie al funzionamento delle parti incluse di default nel framework

## Activity diagram – sistema del carrello

In questo diagramma viene illustrato il funzionamento del sistema del carrello. Nel nostro il termine “carrello” è riferito anche ad un ordine effettuato su più prodotti acquistati in una sola volta (possibilmente anche con diverse quantità) e che sono presenti nello storico degli acquisti dell’utente. Lo storico è necessario per il corretto funzionamento del recommendation system. Lo storico è accessibile all’utente tramite l’API ma non è stata realizzata un’interfaccia client che permetta di visualizzarlo.



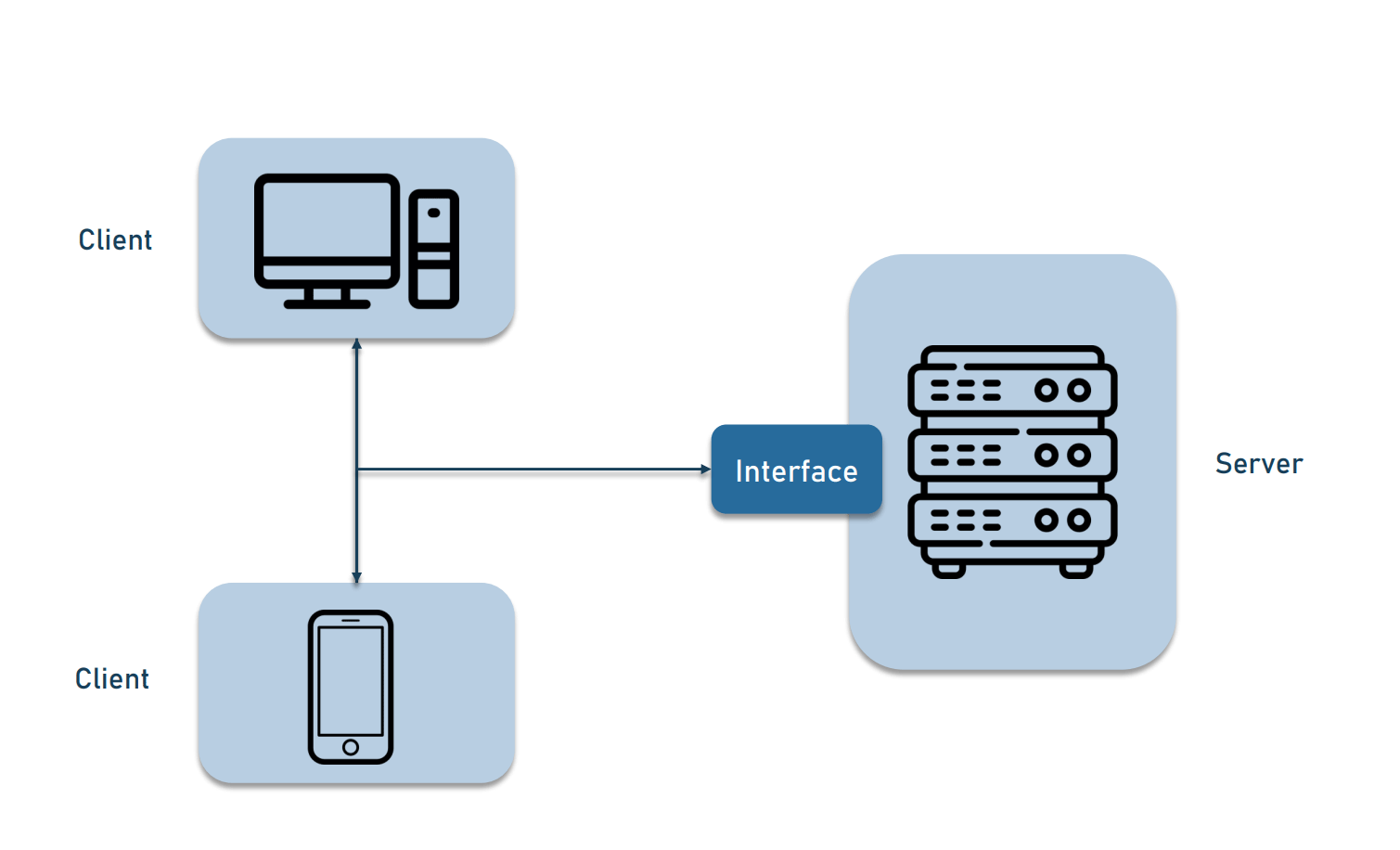
# Tecnologie usate

## API

L’applicazione è divisa in due macrosistemi che sono indipendenti l’uno dall’altro:

* Backend: parte del sistema che agisce sul server ed è controllata utilizzando il framework Django, scritta in linguaggio Python
* Frontend: parte che viene presentata all’utente, ossia l’applicazione web vera e propria con cui egli interagisce, realizzata principalmente tramite la libreria React.js e scritta in Javascript (nella forma JSX)

Questi due sistemi, che rappresentano rispettivamente il server e il client, interagiscono tra di loro tramite chiamate asincrone ai metodi più comuni del protocollo http (GET, POST, PUT, DELETE) da parte del client verso il server, in un modello architetturale chiamato REST (Representational State Transfer) e tramite un’interfaccia definita come API (Application Programming Interface)



Questo tipo di architettura permette di avere un’applicazione che può presentarsi in modo diverso all’utente a seconda del dispositivo utilizzato e di avere anche client completamente diversi dalle tipiche pagine web che possono usufruire delle risorse messe a disposizione dal server, purché concordino sull’interfaccia.

Le applicazioni REST interagiscono tramite metodi del protocollo HTTP tipicamente con questa convenzione (nel nostro caso risorsa e collezione sono equivalenti):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| URL | Metodi | | | |
| GET | PUT | POST | DELETE |
| http://fitcourse/api/risorsa | **Restituisce** una collezione di elementi | **Sostituisce** un’intera collezione con un’altra | **Crea** un nuovo elemento nella collezione | **Elimina** l’intera collezione |
| http://fitcourse/api/risorsa/id | Restituisce l’elemento della collezione tramite identificativo | Sostituisce (o aggiorna) l’elemento indirizzato | Tratta l’elemento come una collezione e crea un elemento | Elimina l’elemento indirizzato dalla collezione |

Nel nostro caso le possibilità con sfondo rosso nella tabella non sono state utilizzate (e tipicamente non vengono utilizzate).

## Autenticazione

Per l’autenticazione abbiamo deciso di usare la tecnologia JWT (Javascript Web Token). Questo è un tipo di autenticazione token-based. A differenza del token di Django un token JWT non viene salvato sul server, ma solo lato client. Il token viene generato dal server al momento del login da parte dell’utente con l’indirizzo e-mail e password corretta. Le informazioni dell’utente vengono cifrate tramite una chiave univoca (contenuta nelle impostazioni del progetto, nel file settings.py) e spedite al client dentro il token. Il token è composto da tre parti:

* Header: contiene il metodo di cifratura e informazioni sul sottotipo di token
* Payload: contiene le informazioni dell’utente (nome, e-mail, ecc..) insieme alla data e orario di emissione e di scadenza del token. Questa parte del token è personalizzabile da parte dello sviluppatore
* Signature: firma di verifica del token (HMAC)

# Organizzazione logica dell’applicazione

Abbiamo deciso di suddividere il sistema in 3 app che gestiscono il sito secondo quanto segue:

Abbiamo deciso di non utilizzare il modello di default fornito da Django ma crearne uno ereditato dalla classe base AbstractUser. Questo ci ha permesso di aggiungere campi personalizzati per il controllo del tipo di utente, caricamento dell’immagine del profilo e impostazione dell’indirizzo email invece del campo “username” come identificativo univoco per l’utente, che effettuerà il login tramite e-mail e password.

# Recommendation system

Abbiamo scelto di adottare recommendation system con user-based collaborative filtering. Il metodo dello user-based collaborative filtering è stato da noi ritenuto più adatto al nostro caso, in quanto il nostro sistema è fondato sulle recensioni dei prodotti da parte degli utenti. Da queste recensioni è possibile ricavare dei punteggi di correlazione tra i prodotti.

Si parte dalle recensioni dei prodotti date dagli utenti:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Utente1 | Utente2 | Utente3 | Utente4 | Utente5 | Utente6 |
| Prodotto1 | - | 3 | - | 7 | 8 | - |
| Prodotto2 | 1 | - | - | 3 | 4 | - |
| Prodotto3 | 9 | - | 10 | - | 7 | 9 |
| Prodotto4 | 5 | 10 | - | 6 | - | - |

Da questa tabella si calcola una matrice di correlazione, usando la SVD (decomposizione a valori singolari):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Prodotto1 | Prodotto2 | Prodotto3 | Prodotto4 |
| Prodotto1 | 1 | 0,80 | 0,16 | 0,33 |
| Prodotto2 | 0,95 | 1 | 0,54 | -0.10 |
| Prodotto3 | -0,45 | 0,54 | 1 | 0,11 |
| Prodotto4 | 0,33 | -0,10 | 0,11 | 1 |

Da questa tabella si cercano i prodotti che sono presenti nello storico degli ordini dell’utente. Supponiamo siano 2 e 3:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Prodotto1 | Prodotto2 | Prodotto3 | Prodotto4 |
| Prodotto2 | 0,95 | 1 | 0,54 | -0.10 |
| Prodotto3 | -0,45 | 0,54 | 1 | 0,11 |

Da questa tabella eliminiamo le colonne con i prodotti scelti. Non vogliamo che vengano ripresentati all’utente siccome li ha già acquistati, ma saranno comunque presi in considerazione per l’ordinamento finale

|  |  |  |
| --- | --- | --- |
|  | Prodotto1 | Prodotto4 |
| Prodotto2 | 0,95 | -0.10 |
| Prodotto3 | -0,45 | 0,11 |

Da qui calcoliamo la mediana dei punteggi per i prodotti 1 e 4, che sarà in questo caso pari alla media:

* Prodotto 1 =
* Prodotto 4 =

Tutti gli altri prodotti avranno punteggio neutro, ossia 0. Da cui possiamo ordinare i prodotti in base al loro calcolato. In caso di punteggio uguale saranno ordinati in base al punteggio del loro rating dato dalle recensioni. Questo criterio di ordinamento permette anche agli utenti nuovi, quelli che non hanno acquistato ancora nessun prodotto e che quindi avranno un punteggio di 0 per tutti i prodotti, di visualizzare per primi i prodotti che hanno ricevuto recensioni più positive.

# Descrizione dei test effettuati

## Ecommerce Api

* test\_total\_bill

Effettua test su tutti i modelli dell’ecommerceApi verificando che la somma totale dei prezzi di tutti i prodotti nel carrello sia giusta. Nella parte di inizializzazione crea istanze arbitrarie dei modelli in questione e infine asserisce che il totale sia effettivamente quello giusto. Questo verifica anche che i modelli siano correlati tra di loro in modo corretto. Effettua anche un test sulla creazione dell’utente.

* test\_checkout\_url\_is\_risolved

Verifica che l’indirizzo URL “checkout” risolva correttamente nella view assegnata.

* test\_products\_list\_GET

Verifica che la chiamata all’API per l’ottenimento della lista dei prodotti restituisca una risposta con stato 200.

* test\_products\_POST

Verifica che la chiamata all’API per la creazione di un prodotto restituisca una risposta con stato 405 (Method Not Allowed), vale a dire che solo l’amministratore può creare un prodotto

## Course Api

* test\_courses\_url\_is\_risolved

Verifica che l’indirizzo URL per l’ottenimento della lista dei corsi disponibili venga risolto correttamente.

* test\_courses\_list\_GET

Verifica che la chiamata all’API per l’ottenimento della lista dei corsi restituisca una risposta con stato 200.

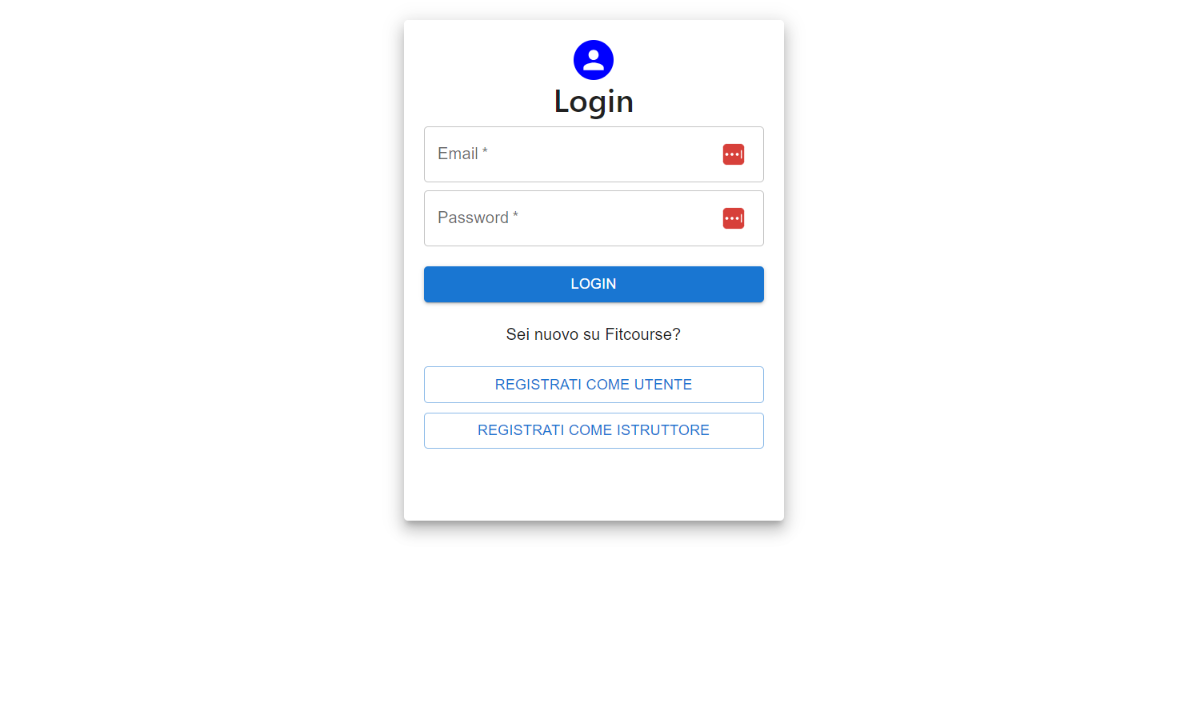
* test\_courses\_POST

Verifica che la chiamata al metodo POST per la creazione di un corso restituisca una risposta con stato 401 (Unauthorized), vale a dire che solo un istruttore può creare un corso

# Risultati dell’applicazione

## Pagina di login

In questa pagina l’utente inserisce l’indirizzo e-mail e la password per autenticarsi nel sito. Viene utilizzata da tutti gli utenti iscritti, sia clienti che istruttori



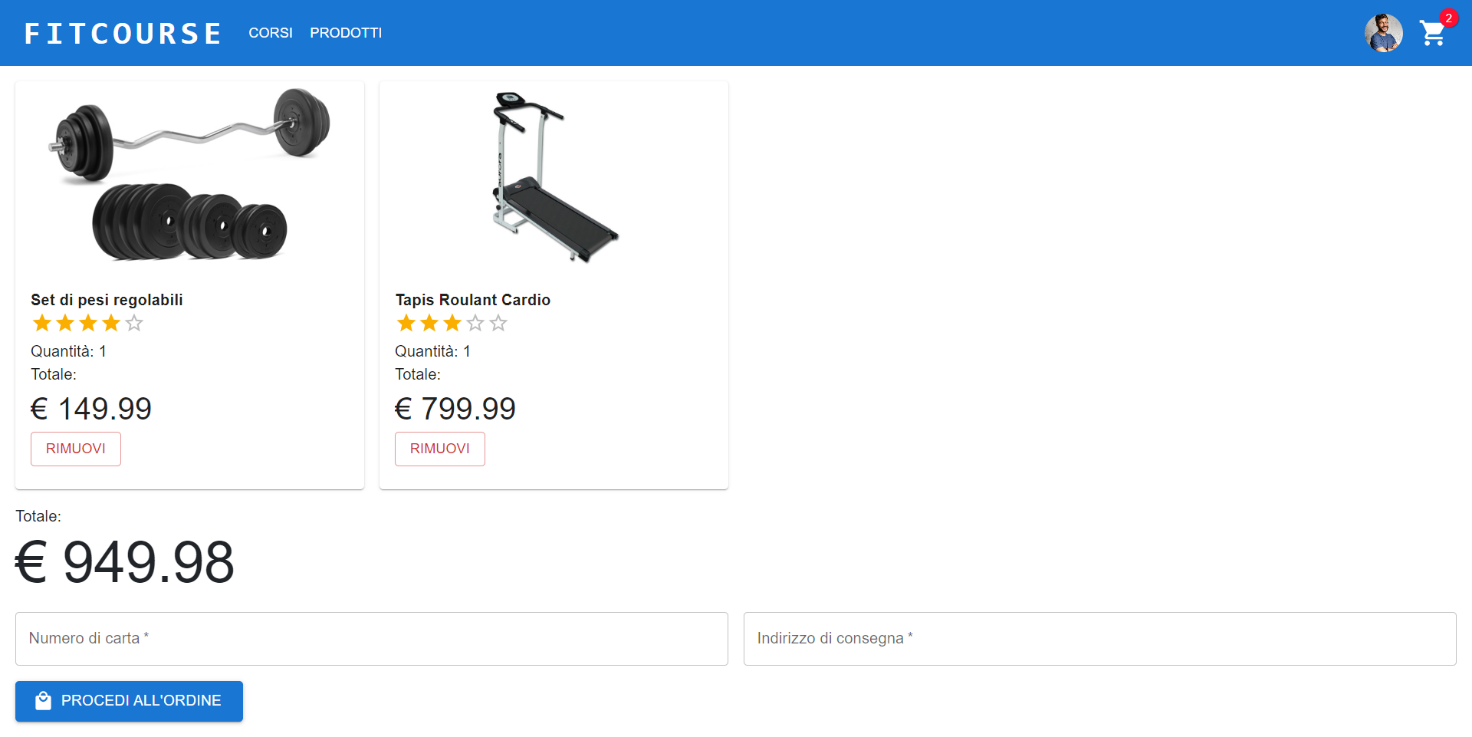
## Pagine di registrazione

Pagine di registrazione rispettivamente dell’utente e dell’istruttore

|  |  |
| --- | --- |
|  |  |
|  |  |

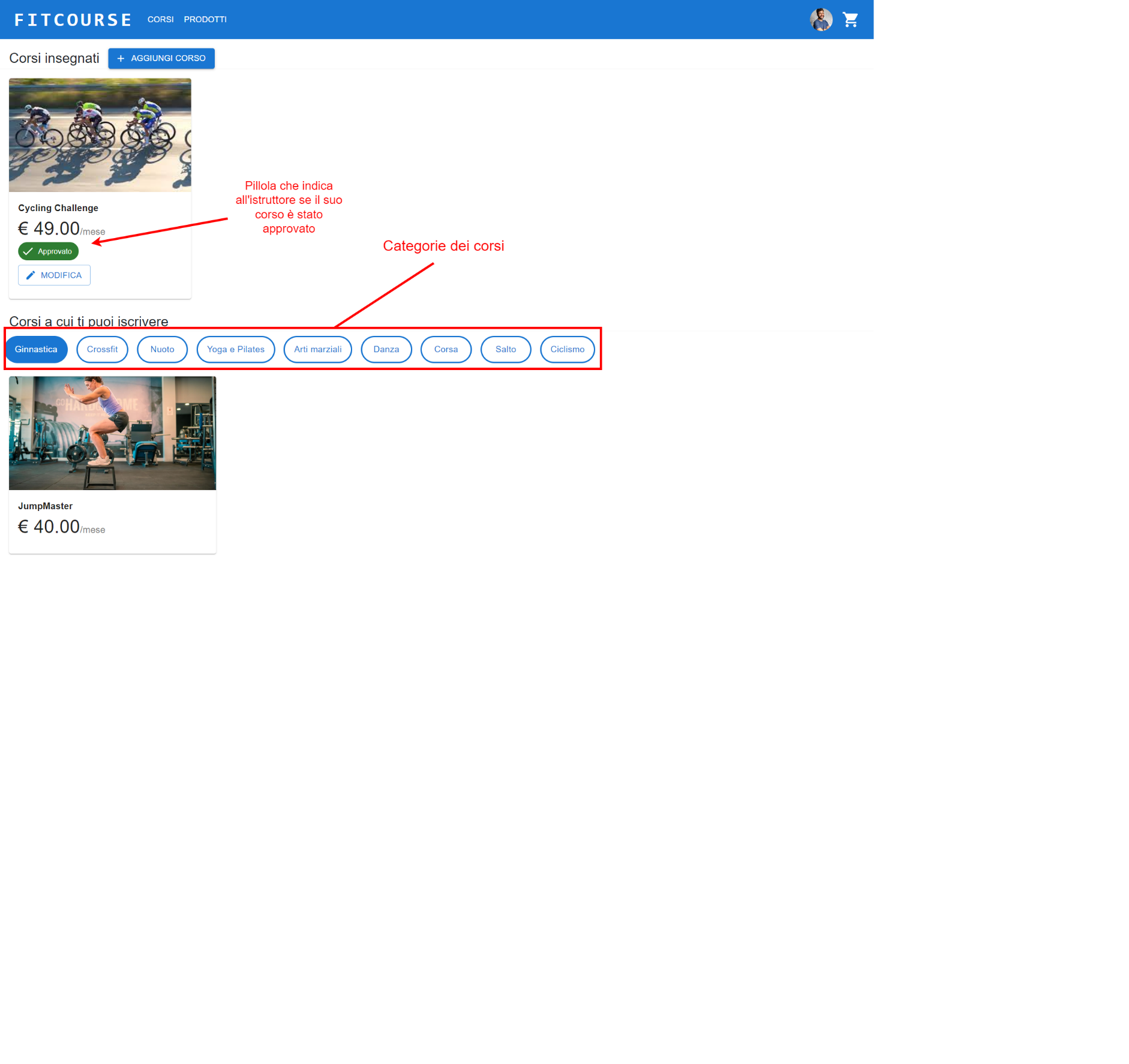
## Homepage

Questa è la homepage, pagina in cui vengono mostrati i prodotti consigliati (primi 4) e i corsi a cui si è iscritti (solo per gli utenti autenticati). I prodotti vengono mostrati in un riquadro uguale a quello della pagina in cui vengono mostrati tutti i prodotti (spiegato successivamente)

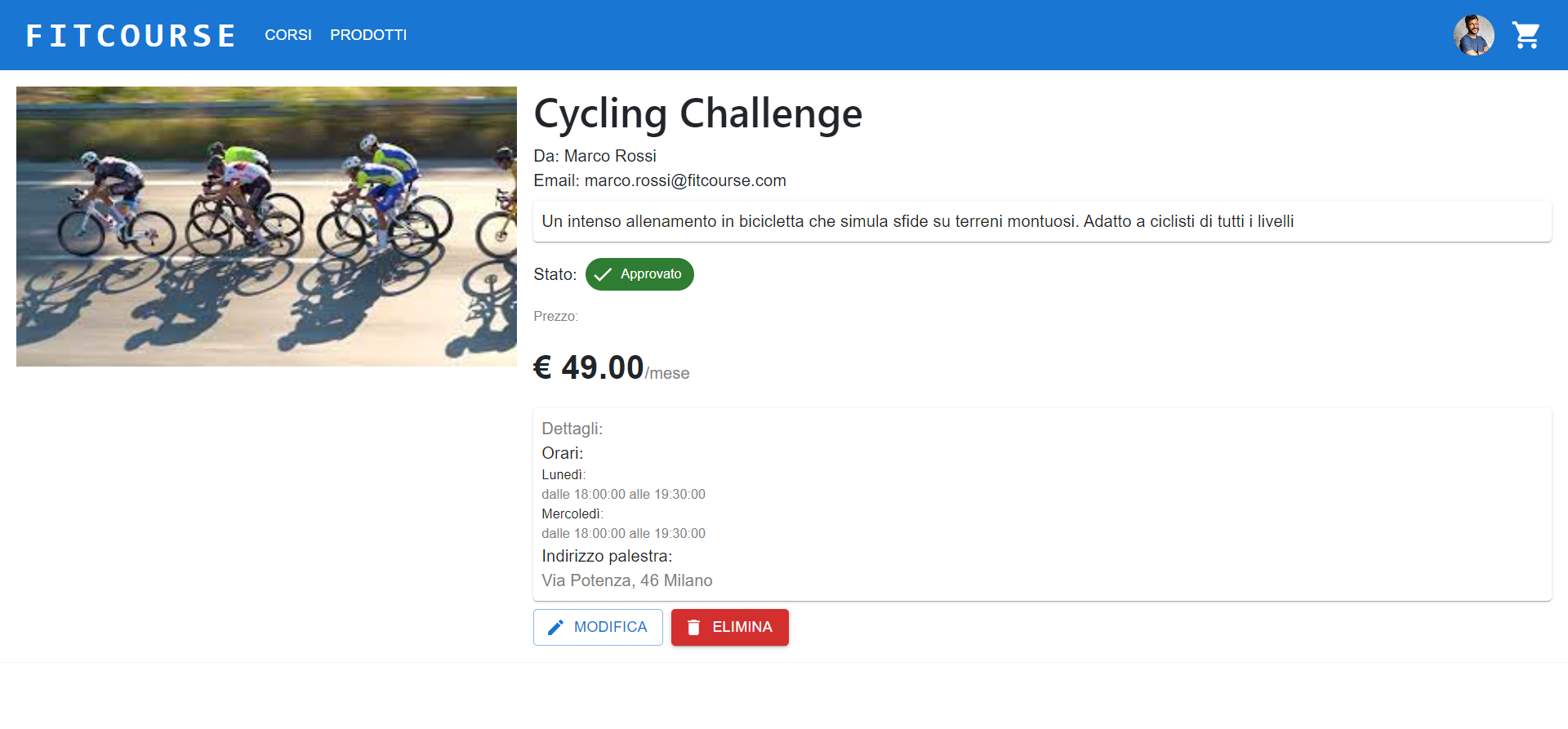


## Pagina di elenco dei corsi

Questa pagina mostra i corsi disponibili (approvati dall’amministratore) a cui un utente si può iscrivere. Per poter visualizzare il dettaglio bisogna cliccare sul singolo riquadro che riassume le principali informazioni sul corso. Gli istruttori hanno una sezione aggiuntiva in cui gli vengono mostrati i corsi che vengono creati da loro.

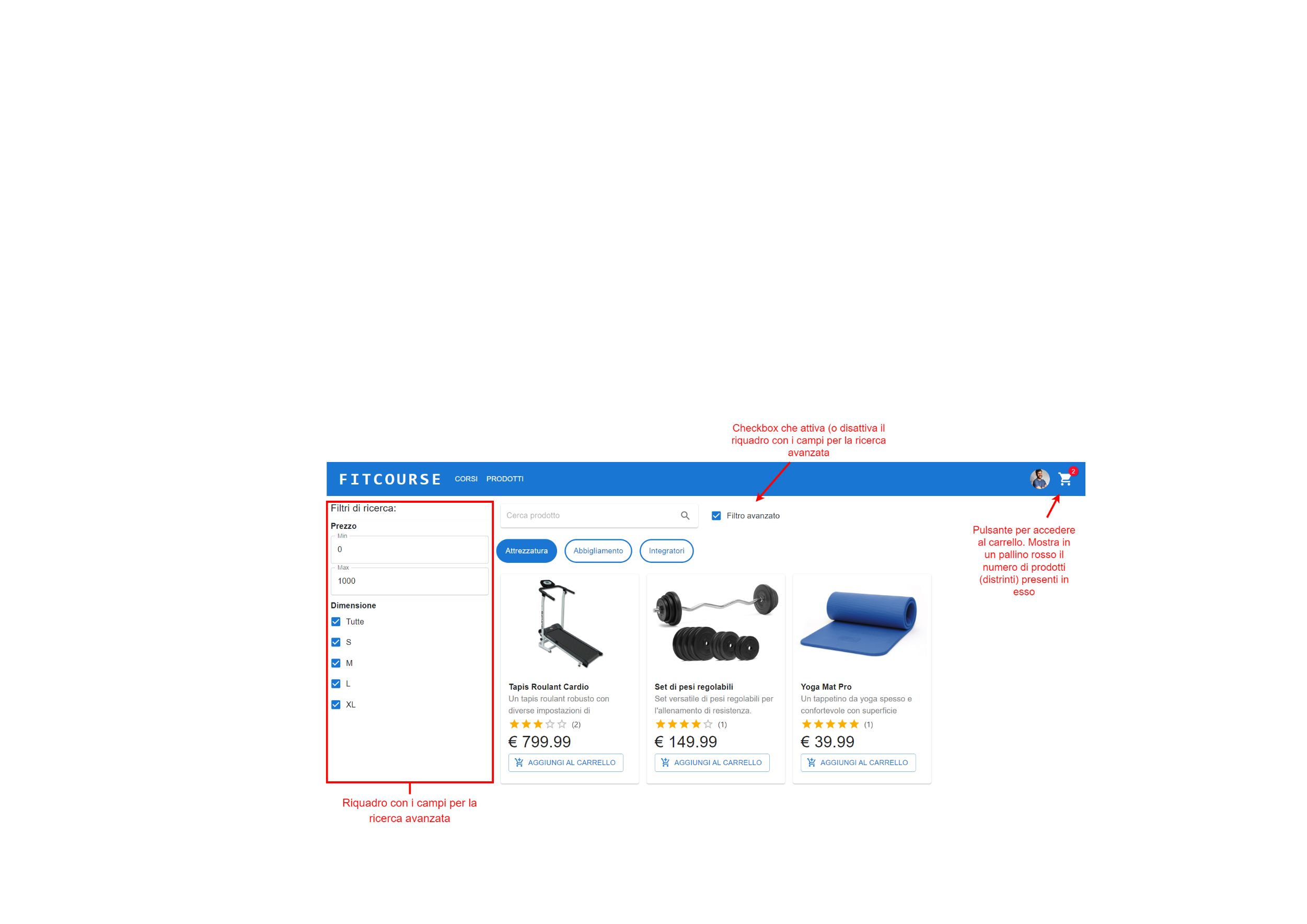


## Dettaglio del corso



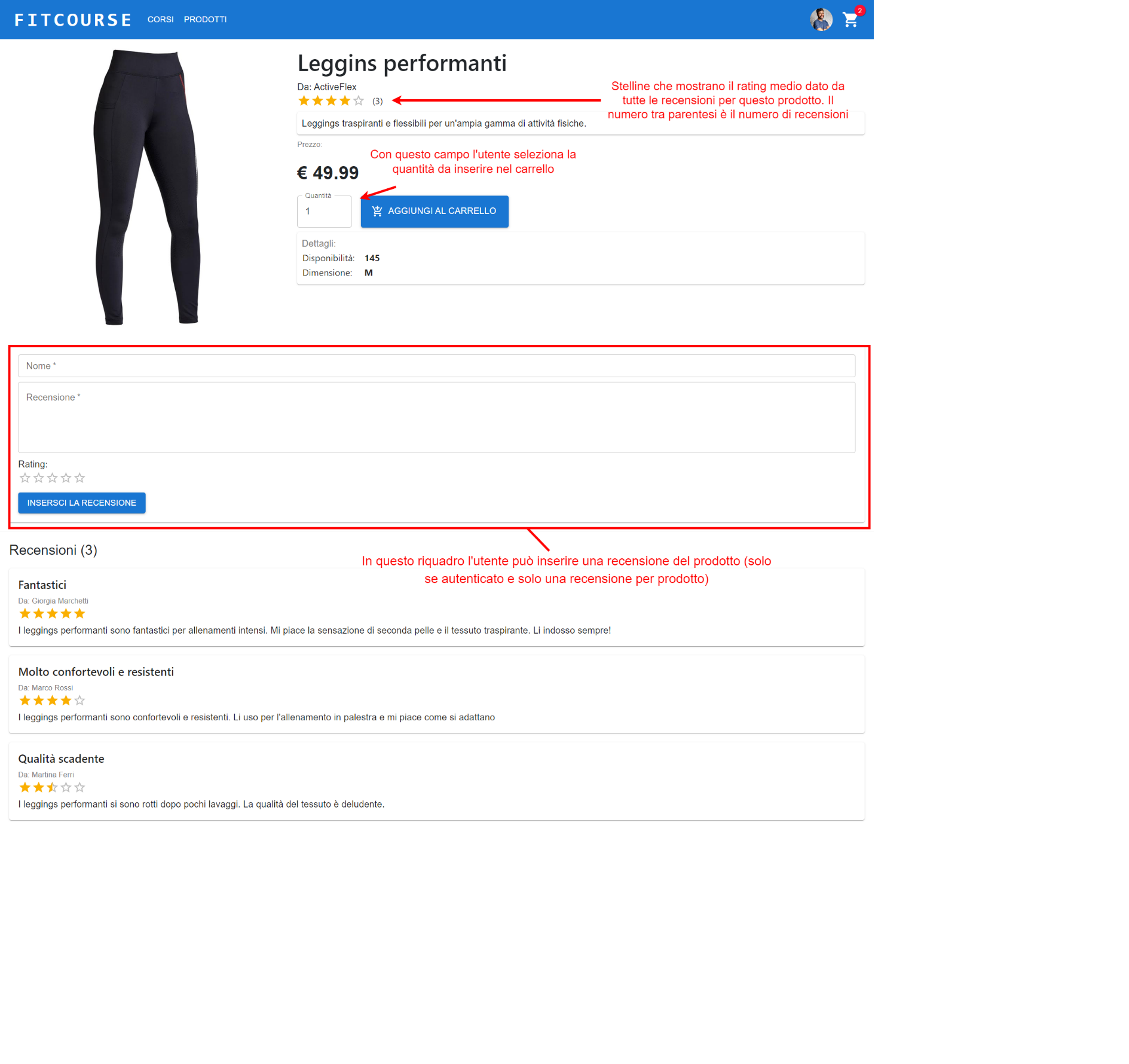
## Pagina di elenco dei prodotti

Questa pagina mostra tutti i prodotti disponibili per l’acquisto sul sito. Come i corsi, anche i prodotti riassumono le informazioni principali in un riquadro con cui è possibile interagire accedendo così al dettaglio del prodotto. Nel riquadro viene mostrato anche un pulsante “Aggiungi al carrello” per permettere all’utente di aggiungere immediatamente il prodotto al carrello (solo una quantità alla volta)



## Pagina del dettaglio di un prodotto

Questa pagina mostra il dettaglio con tutte le informazioni riguardanti un prodotto. Ne visualizza tutte le recensioni e permette ad un utente autenticato di inserirne una propria (solo una per ogni prodotto). Da qui l’utente può anche scegliere la quantità da inserire nel carrello per poter acquistare il prodotto successivamente.



## Pagina del carrello e del checkout

In questa pagina vengono mostrati tutti i prodotti attualmente nel carrello, con la relativa quantità per ogni prodotto. Da qui si procede anche all’ordine dei prodotti, inserendo un numero di carta e un indirizzo di consegna.

