

Aste on-line

Introduzione

Lo scopo è di realizzare un sito dinamico attraverso il web framework Django che ponga all'utente un ambiente in cui poter scegliere fra diverse aste e, in base alla disponibilità di tempo, di poter puntare su prodotti e acquistarli nel caso di vittoria.

Il sito è stato pensato per essere utilizzato da tre tipologie di utente:

- amministratore dispone di privilegi di staff e può creare e gestire le aste. Inoltre loggandosi tramite il sito può anch'esso partecipare ad aste altrui.
- utente registrato può partecipare alle aste e vedere il proprio storico ma non ha privilegi di staff e per questo non può creare e gestire aste.
- utente non registrato ha la possibilità di visitare il sito ed effettuare ricerche sulle aste; nel momento in cui desidera partecipare ad un'asta deve iscriversi al sito (tramite l'apposita pagina di signup) in modo da poter rialzare il prezzo di un prodotto e controllare il proprio storico.

L'applicazione web dinamica deve consentire una ricerca sulle aste basata sul prezzo e sulla data di inizio. I due tipi di filtri vengono posti in AND fra loro in modo che il risultato della ricerca sia incrociato (es. aste scadute nella fascia fra i 50 e i 200 €). Per la ricerca basata sulla data è stato scelto di utilizzare un widget di tipo checkbox in quanto in questo modo è possibile selezionare più fasce di tempo contemporaneamente. A livello di sviluppo questo significa porre i box selezionati in OR in modo da ritornare quelle aste che rispecchiano tutti i box preferiti (es. sia aste scadute che odierne).

Ogni rialzo da parte di un utente viene effettuato di una quantità standard (0.25 €) e va ad incrementare il prezzo visualizzato da tutti gli utenti. Come in qualsiasi asta, l'ultimo utente che rilancia è colui che si aggiudica l'oggetto desiderato; tuttavia questa unità non è stata trattata nel progetto a causa dei tempi ristretti e del numero degli sviluppatori. I rispettivi rialzi degli utenti vengono salvati nella propria pagina personale in modo che se ne possa tenere traccia e possano essere utilizzati in un futuro in maniera costruttiva.

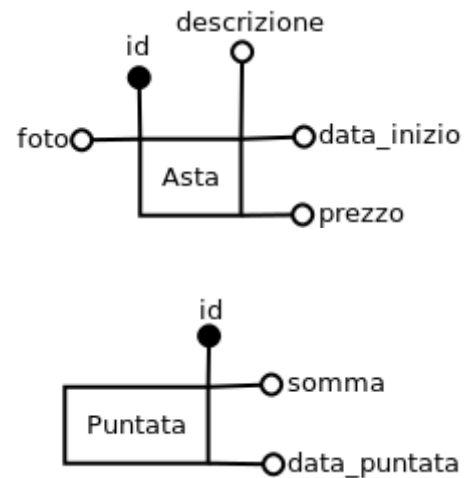
È presente inoltre una funzionalità per la quale un utente registrato, interessato ad un'asta futura, può farsi recapitare una mail di notifica in modo da avere memoria di quando questa comincerà, senza dover tornare frequentemente sul sito per controllare la data di inizio.

Struttura del database

Per la struttura del modello sono state scelte due entità, Asta e Puntata. Come entità Utente è stata utilizzata l'entità User predefinita da django.

L'asta è composta da: una descrizione che serve a definire un breve incipit dell'asta, una data di inizio di tipo Datetime che indica data e ora di inizio, un prezzo che inizialmente sarà quello impostato dal creatore dell'asta mentre quando sarà scaduta sarà quello di vendita e infine una foto che contiene l'indirizzo in memoria della foto da visualizzare del prodotto. Come vedremo dall'ER completo Asta ha inoltre una foreign key added_by che rappresenta l'utente creatore dell'asta.

L'entità Puntata rappresenta una singola puntata effettuata da un utente relativamente ad un'asta. Puntata è composta da una somma che rappresenta la somma puntata e la data in cui viene effettuata. Si trovano inoltre due foreign key: utente che identifica l'utente che effettua la puntata e asta che serve invece a identificare l'asta a cui la puntata si riferisce. Viene inoltre aggiunto un vincolo di unicità sull'asta e la somma in modo che non possano essere inserite delle puntate con la stessa somma sulla medesima asta.



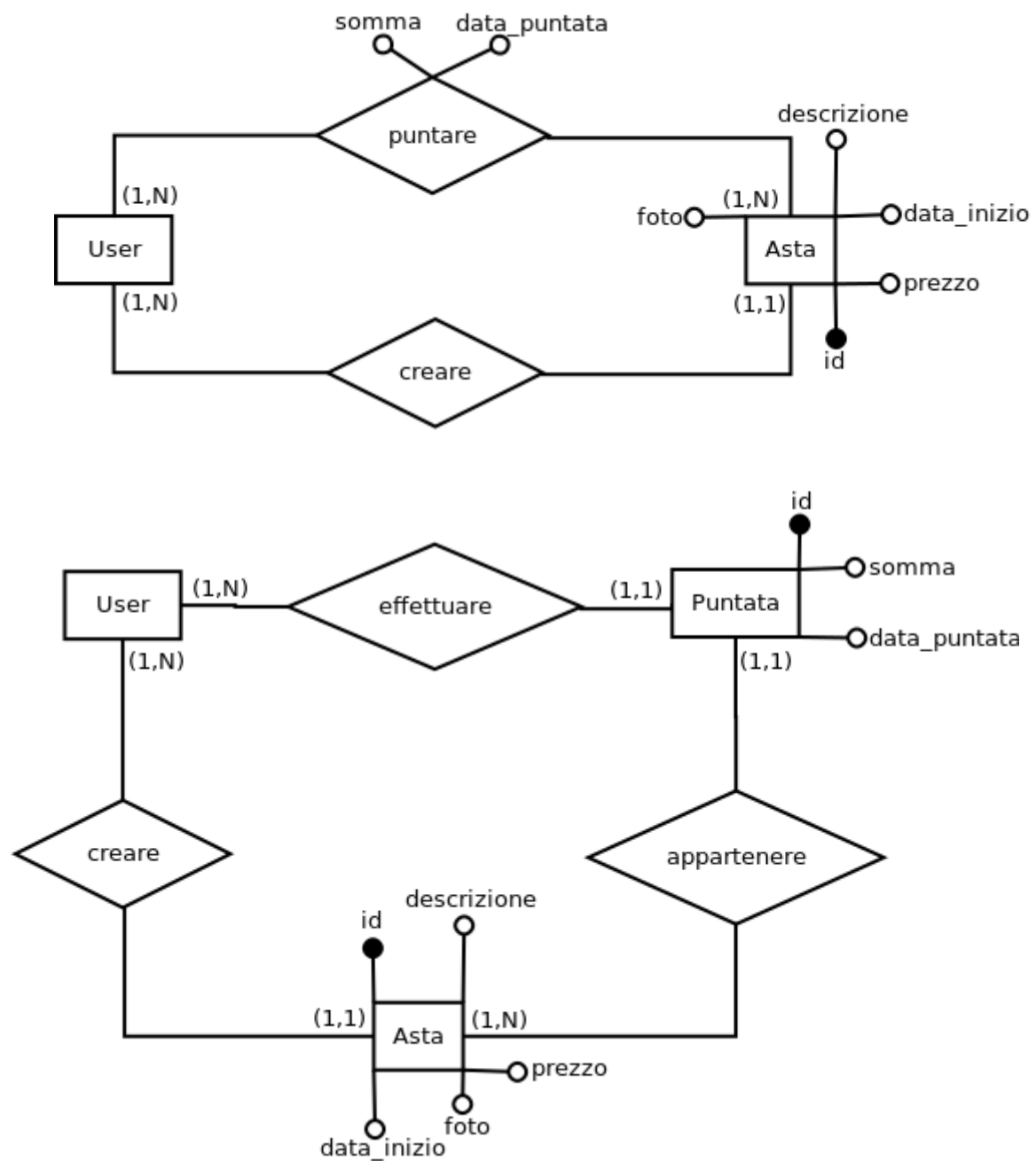
Schema logico

Asta(id, descrizione, data_inizio, prezzo, foto, id_user)
 FK: id_user REFERENCES User

Puntata(id, somma, data_puntata, id_asta, id_user)
 FK: id_asta REFERENCES Asta
 FK: id_user REFERENCES User

ER

Inizialmente l'ER completo comprendeva due entità, Asta e Utente, e due relazioni, Puntare con cardinalità N:N e Creare con cardinalità 1:N. Si è scelto di reificare la relazione Puntare per farla diventare un'entità in modo da ridurre la ridondanza. Di seguito è riportato il passaggio dalla prima alla seconda idea.



Scelte implementative

Per il supporto, la creazione e la gestione del database è stato scelto l'utilizzo del DBMS sqlite3. Questo database manager risulta essere leggero in quanto sfrutta la capacità di memorizzare i dati su un file ed è inoltre supportato da API semplici da utilizzare. Alcuni comandi che possono essere utilizzati dopo aver lanciato "sqlite3" da terminale sono:

- `.open "db.sqlite3"` → per aprire il database del progetto
- `.tables` → una volta aperto il database servirà per mostrare le tabelle
- `.schema "aste_asta"` → per visualizzare la struttura di una tabella
- query in generale: `select, delete, drop ...`

Struttura gerarchica del progetto

Di seguito verrà proposta la struttura del progetto, il tutto secondo le regole predisposte da Django.

Progetto_A è il nome del progetto e, come ogni progetto Django, all'interno di questo è presente l'applicazione progetto_A. Nella root directory sono inoltre presenti il file "db.sqlite3", un file "README.md", il file "manage.py" e le sottocartelle "/doc", "/media", "/aste" e "/templates". Tramite l'applicazione progetto_A sono gestite le funzioni basilari per il funzionamento del sito come: Home, Login, Logout, SingUp, l'accesso all'area admin, ...

Il file README.md contiene una breve descrizione su come utilizzare l'applicativo mentre il file db.sqlite3 contiene un dump del database ed è utilizzabile appunto con SQLite3.

L'applicazione progetto_A contiene un file di nome settings che è utilizzato per gestire tutte le impostazioni di progetto come i path, le impostazioni per l'utilizzo di una mail, le applicazioni esterne da utilizzare, le impostazioni del database, le impostazioni riguardanti la lingua e la zona del fuso orario, ...

Il file urls contiene una lista per la gestione degli urls navigabili all'interno del sito, questo file importa il file views per poter richiare funzioni all'interno di esso in seguito alla richiesta di un url. Tramite il file views vengono eseguite delle funzioni e visualizzate le pagine web attraverso i templates in formato html. Un altro file importante in questa sottocartella è forms che contiene la struttura di un modello da visualizzare all'interno di un file html per diminuire il codice e renderlo facilmente modificabile. Nel caso di questo specifico file forms è utilizzato solamente per la creazione di un nuovo utente, quindi per la funzione di signup.

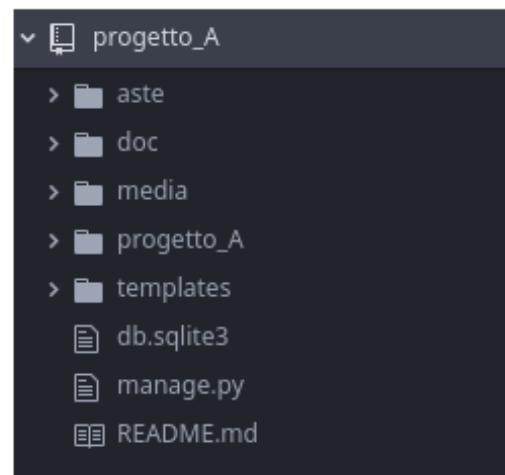
La sottocartella templates contiene tutti i file html del progetto_A ma non della sottoapplicazione aste. È stato creato un template base.html da cui ereditano tutti la struttura in modo da riutilizzare il codice senza doverlo scrivere ogni volta, inoltre è presenta una cartella registration contenente i soli file di login e signup.

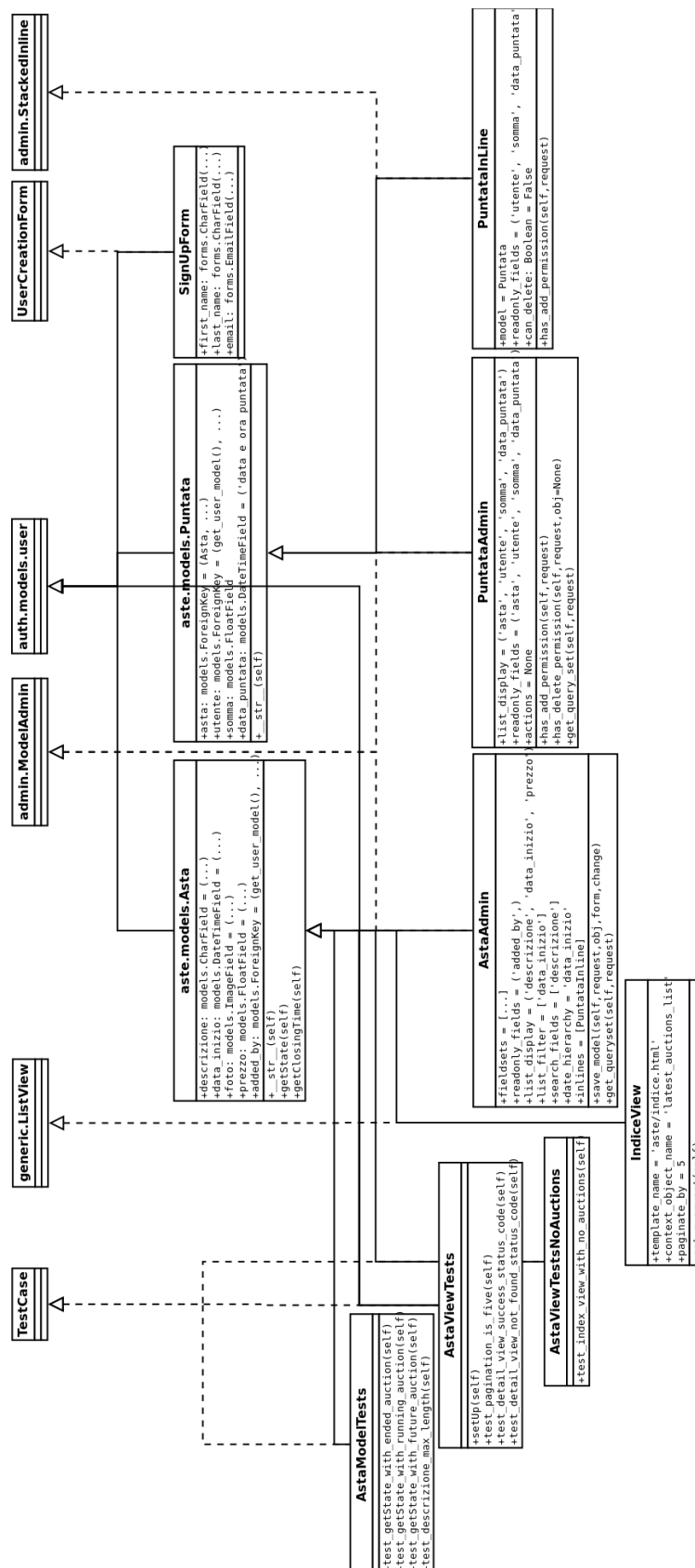
Nella cartella media sono presenti le foto caricate dai membri dello staff relative ai prodotti delle aste.

Nella cartella doc invece sono contenuti i file relativi alla documentazione come la relazione.

La cartella aste è una sottoapplicazione di progetto_A e sviluppa le funzionalità relative alla gestione delle aste e delle puntate effettuate dagli utenti.

Vengono per prima cosa create le due entità Asta e Puntata all'interno del file models il che equivale a creare due nuove tabelle all'interno del database. Intorno a queste due classi funziona tutto l'applicativo, ad esempio il file views contiene i metodi per visualizzare l'elenco delle aste e ogni asta nel singolo dettaglio. Sono presenti inoltre altre funzioni come l'invio di un promemoria o la creazione di un asta (nel caso se ne abbiano i privilegi). Nel file urls sono presenti gli url utilizzabili dal sotto indirizzo /aste/ che facendo uso dei metodi nelle views reindirizzano l'utente ai templates desiderati. Il file tests serve ad effettuare test di controllo sull'applicazione in modo che non si venga mai a trovare in condizioni anomale; in seguito si troverà la spiegazione dettagliata. Infine nel file admin sono contenuti i dettagli grafici della parte amministrativa del sito. Un esempio di modifica apportato a questo file è la visualizzazione delle aste, a cui sono state aggiunte le puntate relative ad ogni asta le quali però non possono essere modificate. Anche nella visualizzazione delle puntate non è possibile modificarle in quanto è stato scelto che una volta effettuata una puntata questa non possa essere rimossa.





Test

La fase di testing django è stata apportata sul modello e sulla view dell'applicazione aste.

- **metodo *getState* Asta:**
in questo test viene controllato se i tre tipi di stato (0, 1, 2) ritornati dalla funzione *getState* del modello Asta sono ritornati adeguatamente in modo che nella pagina di dettaglio venga visualizzato ciò che deve in base al periodo (passato, presente o futuro) in cui si è svolta o si svolgerà l'asta.
- **lunghezza massima del field *descrizione* di Asta:**
viene controllato che la lunghezza massima della descrizione di un'asta sia realmente quella assegnatagli in fase di creazione del modello.
- **paginazione della view *indice*:**
controllo effettuato sulla view in cui viene accertato che la paginazione applicata al template è realmente quella richiesta nella view (nel caso di questo progetto di 5 aste in ogni pagina).
- **codici successo/insuccesso della view *dettaglio*:**
in questo test viene controllato che, in base al parametro passato nell'url, vengano ritornati i codici di stato adeguati (se l'asta esiste 200 mentre se non esiste 404).
- **ritorno senza aste della view *Indice*:**
in questo test viene controllato che nel caso in cui non ci siano aste da visualizzare nella lista all'interno dell'indice, venga presentato un messaggio appropriato per l'utente.

Javascript

Nello sviluppo del progetto era richiesto anche l'utilizzo di script tramite linguaggio javascript e ne è stato pensato il suo utilizzo per lo sviluppo di un timer che possa scandire la durata delle aste.

- **timer**
questa funzione è posta nella pagina di dettaglio di un'asta ed è necessaria a visualizzare in tempo reale quanto durerà ancora l'asta. In questo modo l'utente riesce a gestire meglio le puntate. Il funzionamento è semplice, viene usato il metodo *getClosingTime* della classe asta per recuperare l'ora in cui finirà l'asta e dopodichè ogni secondo viene aggiornato il timer facendo la sottrazione fra l'istante "ora" e la fine dell'asta. Una particolarità dei mesi di javascript è che iniziano da 0, per questo quando viene recuperata l'ora di fine asta bisogna sottrarre uno al mese perché django parte a contare i mesi da 1.

Extra

Con l'aggiunta della paginazione è stato riscontrato un bug nella view *indice*. Questo problema si manifestava nel caso in cui venivano applicati filtri tramite il form alla lista delle aste e vi erano un numero di pagine maggiori o uguali a due. La lista ritornata era corretta per la prima pagina ma andando a premere su pagina successiva o precedente si perdevano i filtri (su data e/o prezzo) e veniva mostrata la lista di aste senza filtri applicati. Per questo motivo è stato realizzato un tag ad hoc da applicare ai link per la navigazione delle pagine del template *indice.html* che va a salvare i filtri dell'url e li mantiene anche nel caso di pressione dei bottoni *successiva* o *precedente*.

```
@register.simple_tag(takes_context=True)
def url_replace(context, **kwargs):
    query = context['request'].GET.dict()
    query.update(kwargs)
    return urlencode(query)
```

Descrizione grafica dell'applicativo

Di seguito verranno presentati gli screenshot dell'applicazione con la relativa spiegazione in modo che sia semplice e intuibile il suo utilizzo.

Home

L'home page contiene i link principali e le istruzioni basilari per poter navigare il sito. La pagina personale è accessibile esclusivamente da utenti registrati e contiene l'elenco delle puntate di un utente.

Sito di Aste on-line

Benvenuto sul sito di aste, di seguito troverai una lista di link con cui poter navigare il sito. Chiunque può navigare il sito, con la differenza che gli utenti non registrati possono solamente guardare le aste disponibili.

- [Login](#)
- [Indice aste disponibili](#)
- [Pagina personale](#)

Simone Cavana - Unimore
© 2018, Django

Il link *indice aste disponibili* porta all'elenco delle aste ed è accessibile anche da utenti anonimi. Con login è possibile accedere se si ha un account e nel caso si effettui l'accesso verrà visualizzato il nome dell'utente e apparirà un nuovo link per poter effettuare il logout. Nel caso ci si loggi come utente amministratore o staff verrà visualizzato inoltre il link per accedere al portale admin.

Login e Signup

La pagina di login è così composta e nel bottom della pagina è inserito il link per poter creare un account di tipo utente base. A seguito della pressione del bottone login si verrà reindirizzati nella home page.

Di seguito viene mostrata la pagina d'iscrizione al sito contenente alcuni campi facoltativi e le istruzioni necessarie all'accettazione del nuovo account. I campi nome e cognome sono facoltativi perché non è presente alcuna funzione che li utilizzi mentre la mail è ritenuta obbligatoria in quanto è presente la funzione di invio delle mail per permettere all'utente di ricordarsi delle aste da lui prescelte. A seguito della pressione del bottone signup si verrà reindirizzati nella home page.

Aste django

Pagina di login

Nome utente:

Password:

login

Non hai un account? [Iscriviti.](#)

Registrati

Nome utente: Obbligatorio. 150 caratteri o meno. Solo lettere, cifre e @/./+/_

First name: Opzionale.

Last name: Opzionale.

Email: Obbligatorio. Inserisci un indirizzo email valido.

Password:

- La tua password non può essere troppo simile alle altre tue informazioni personali.
- La tua password deve contenere almeno 8 caratteri.
- La tua password non può essere una password comunemente usata.
- La tua password non può essere interamente numerica.

Conferma password: Inserisci la stessa password inserita sopra, come verifica.

Pagina personale

Qui invece viene mostrata un esempio di pagina personale in cui nella tabella sono mostrate le puntate effettuate con i relativi dettagli di prezzo e orario. Anche qui è presente un link per tornare alla home e rendere perfettamente navigabile il sito.

Lista delle puntate effettuate:

asta	somma	data
CT Helmet	41,95	Giovedì 21 Giugno 2018 19:22
CT Helmet	42,2	Giovedì 21 Giugno 2018 19:23
CT Helmet	42,45	Giovedì 21 Giugno 2018 19:23
Bose speakers	191,2	Giovedì 21 Giugno 2018 19:32
CT Helmet	42,7	Giovedì 21 Giugno 2018 19:32
Bose speakers	191,45	Giovedì 21 Giugno 2018 19:33
Bose speakers	191,7	Giovedì 21 Giugno 2018 19:33
CT Helmet	42,95	Giovedì 21 Giugno 2018 19:33
CT Helmet	43,2	Venerdì 22 Giugno 2018 10:50
CT Helmet	43,45	Venerdì 22 Giugno 2018 10:50
CT Helmet	44,2	Venerdì 22 Giugno 2018 15:38
Gopro 5 Black	181,15	Martedì 26 Giugno 2018 09:52
Gopro 5 Black	181,4	Martedì 26 Giugno 2018 09:52
Macchinetta caffè	85,37	Mercoledì 27 Giugno 2018 16:35
Macchinetta caffè	85,62	Mercoledì 27 Giugno 2018 16:35

Premere [Home](#) per tornare alla pagina principale del sito.

Indice e Dettaglio

Nella pagina di indice è presente la lista delle aste create in ordine decrescente di data in gruppi di 5 per pagina e il relativo slider delle pagine (precedente e successiva). Si trova inoltre la possibilità di applicare un filtro basato sulla data o sul prezzo delle aste tramite un form. Questo form sfrutta il passaggio di parametri di tipo GET e va a filtrare la lista da visualizzare tramite query dinamiche *Q filter* di django. Tramite la pressione del bottone *filtra* viene applicato il filtro e rivisualizzata la lista. Sono presenti link per tornare alla pagina home e per creare una nuova asta, se si dispone dei privilegi adeguati. Ogni asta della lista è un link che si ricollega alla pagina di dettaglio dell'asta per poter visualizzare: una foto dell'oggetto, il tempo rimasto per effettuare puntate se l'asta è in corso oppure la data in cui è finita o inizierà l'asta, il prezzo corrente o il prezzo di vendita e i link per tornare alle pagine precedenti. Il tutto è facilmente comprensibile tramite le immagine sottostanti. Nella pagina di dettaglio, nel caso in cui l'asta sia in corso, è presente un bottone *rialza* che fa parte di un form e permette di aumentare la quota dell'oggetto per aggiudicarselo e questo bottone va a salvare nel db la nuova puntata (asta, utente, quantità e data) e aggiorna il prezzo dell'asta in modo che il prezzo di rialzo sia sempre crescente.

Aste disponibili:

- [Frigo](#)
- [Equipaggiamento base d'arrampicata](#)
- [Bose speakers](#)
- [Macchinetta caffè](#)
- [Gopro 5 Black](#)

[precedente](#) Pagina 2 di 3. [successiva](#)

Filtra per:

data: ☐ scadute
☐ oggi
☐ prossimamente

prezzo: ☐ qualsiasi
☐ € 0 - € 50
☐ € 50 - € 200
☐ € 200 - € 500
☐ > € 500

Filtra

[Crea una nuova asta](#)
[Home](#)

Gopro 5 Black



Scaduta il: Martedì 26 Giugno 2018 09:52

Prezzo di vendita: 181,4 €

[Home](#)
[Torna alle aste](#)