

Università degli studi UNIMORE
Facoltà di Informatica

Progetto di Tecnologie Web
SmartStore di Lucrezia Sitta (matr. 144884)

Anno Accademico 2022-23

Indice:

1. Traccia dettagliata che illustra le funzionalità richieste	pag. 3
2. Diagrammi	pag. 4
2.1 Diagrammi dei casi d'uso	pag. 4
2.2 Diagramma di utilizzo	pag. 5
2.3 Diagramma delle classi Bike	pag. 6
2.4 Diagramma delle classi –Account	pag. 7
3. Tecnologie utilizzate	pag. 8
3.1 Django	pag. 8
3.2 Javascript	pag. 8
3.3 Bootstrap	pag. 8
3.4 Ajax	pag. 8
4. Descrizione del progetto	pag. 9
4.1 Tipologie di utenti presenti	pag. 9
4.2 Gestione degli ordini	pag. 9
4.3 Vendere un prodotto	pag. 10
4.4 Homepage	pag. 10
5. Recommendation System	pag. 11
6. Test	pag. 12
6.1 Test Login	pag. 12
6.2 Test inserimento articolo	pag. 12
6.3 Test reindirizzamento e login	pag. 13
6.4 Test cambio indirizzo utente	pag. 13
6.5 Test elimino indirizzo utente	pag. 14
7. Screenshot	pag. 15
7.1 Homepage	pag. 15
7.2 Profilo Utente Admin	pag. 15
7.3 Profilo Utente Venditore	pag. 16
7.4 Profilo Utente Compratore	pag. 16
7.5 Visualizzazione del carrello	pag. 17
7.6 Riepilogo degli acquisti utente	pag. 18
7.7 Risultati di ricerca	pag. 18
7.8 Pagina di checkout	pag. 19
7.9 Visualizzazione articolo	pag. 19
7.10 Pagina in base alla categoria selezionata	pag. 20
7.11 Recommendation System	pag. 20
8. Conclusioni	pag.21

1- Traccia dettagliata che illustra le funzionalità richieste

Il progetto consiste nello sviluppo di un e-commerce di biciclette e i vari accessori. Gli utenti che possono visitare il sito sono di tre tipologie, anonimi, registrati e fornitori.

Gli utenti anonimi possono usufruire di diversi servizi quali:

- La navigazione libera nelle pagine del sito
- Ricercare dei prodotti
- Consultare gli articoli presenti nell'e-commerce

Gli utenti registrati invece, oltre ai servizi offerti per gli utenti anonimi, hanno anche la possibilità di:

- Acquistare una bicicletta o un accessorio

Gli utenti fornitori invece, oltre ai servizi offerti per gli utenti anonimi e registrati hanno anche la possibilità di:

- Vendere una bicicletta o un accessorio
- Visualizzare la propria area riservata, nella quale troveranno i propri articoli in vendita, acquisti effettuati e vendite
- Eliminare i propri articoli venduti

I prodotti saranno individuati da un nome (identificazione), dall'immagine, prezzo e da una descrizione.

Il sito deve comunque prevedere altri servizi: sarà possibile cercare i prodotti tramite una barra di ricerca, oppure filtrando le caratteristiche secondo le esigenze dell'utente.

È inoltre presente un Recommendation System che, basandosi sugli acquisti effettuati precedentemente consiglia ad ogni utente eventuali prodotti da acquistare.

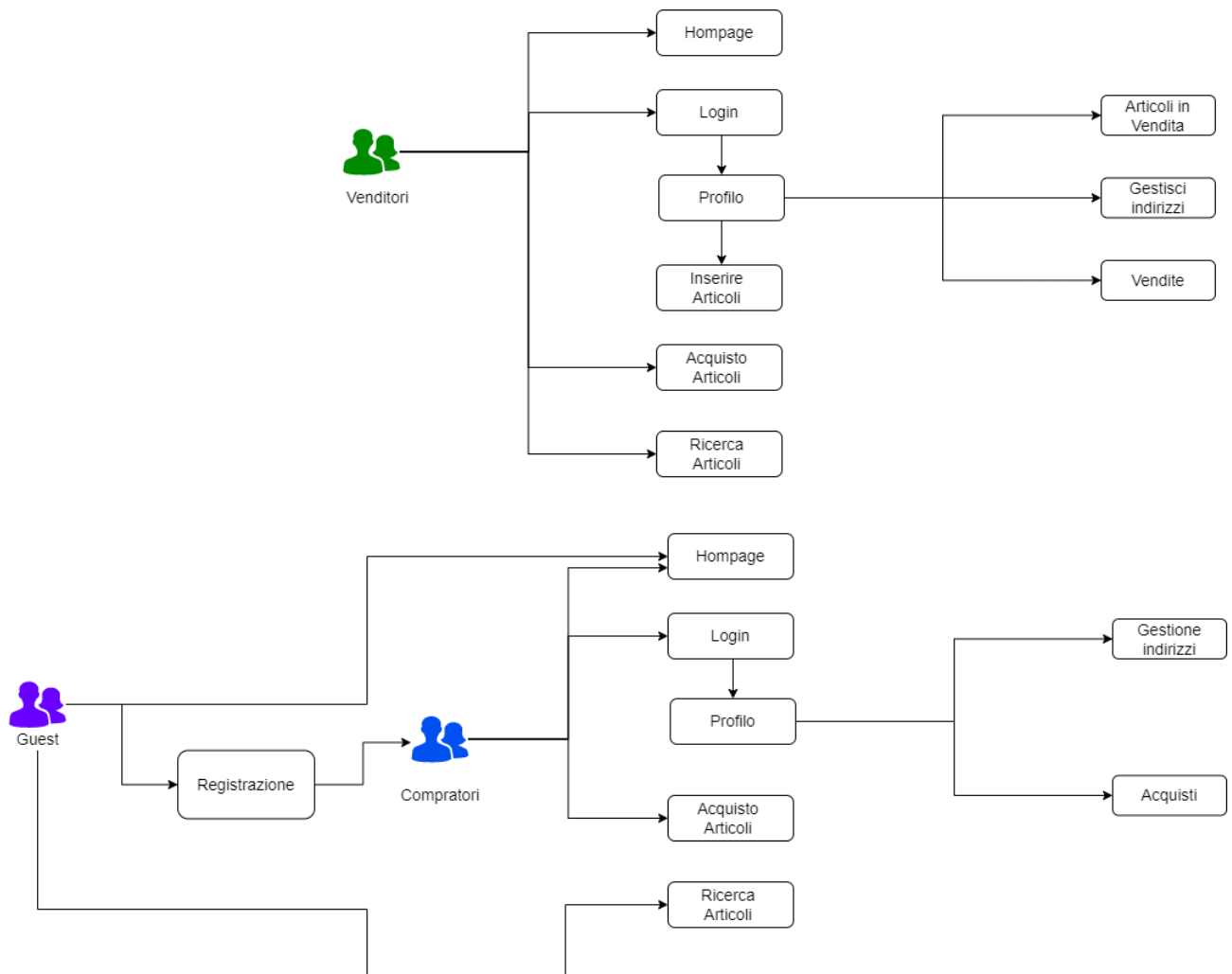
È presente anche una newsletter che permetta, in modo automatico, a tutti gli utenti registrati, di ricevere aggiornamenti su nuovi prodotti inseriti in piattaforma.

Tra i punti facoltativi, che possono implementare la qualità del progetto ho scelto:

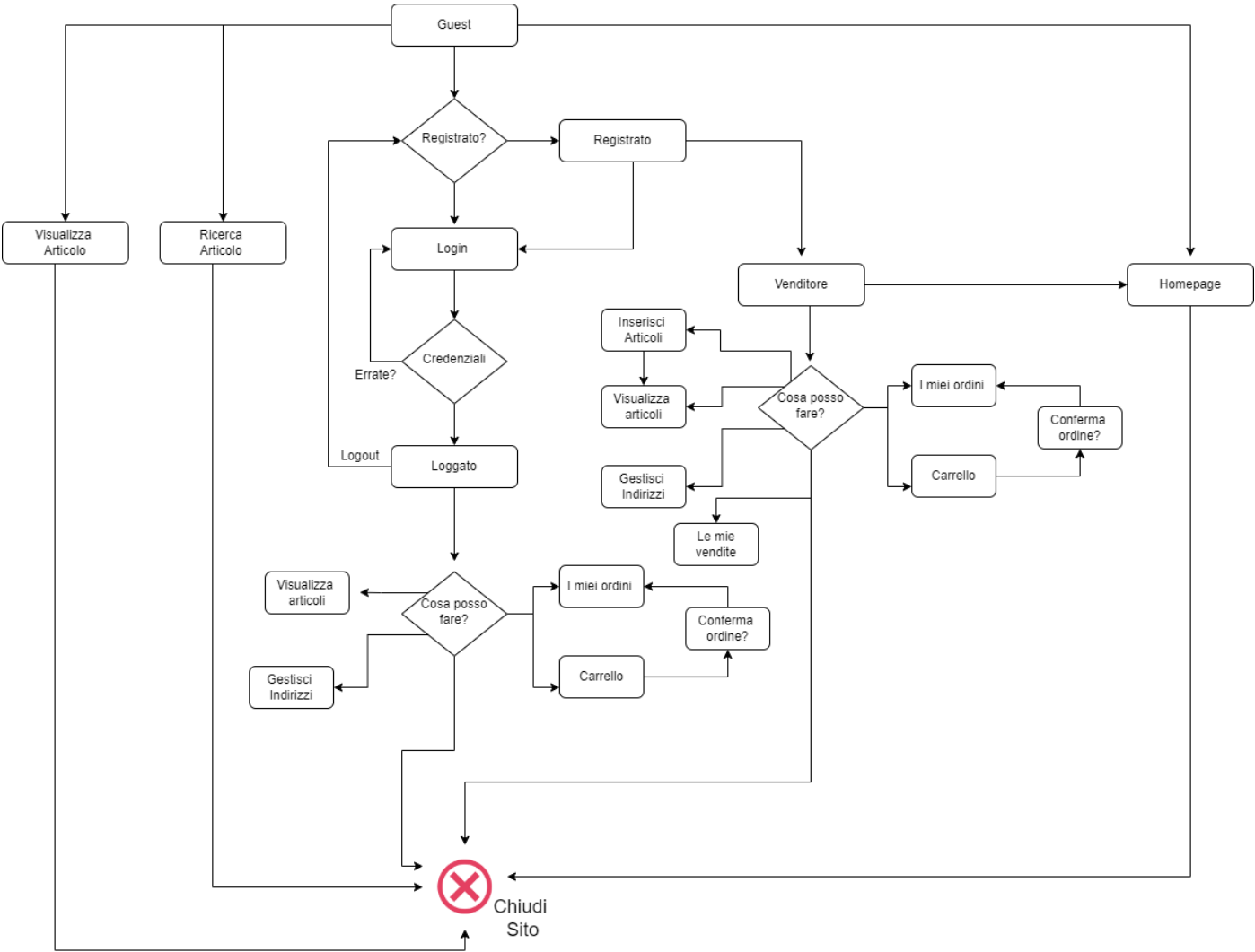
- la funzione carrello, in cui l'utente può inserire i prodotti da acquistare
- il pagamento che può essere effettuato tramite bonifico o alla consegna

2 – Diagrammi

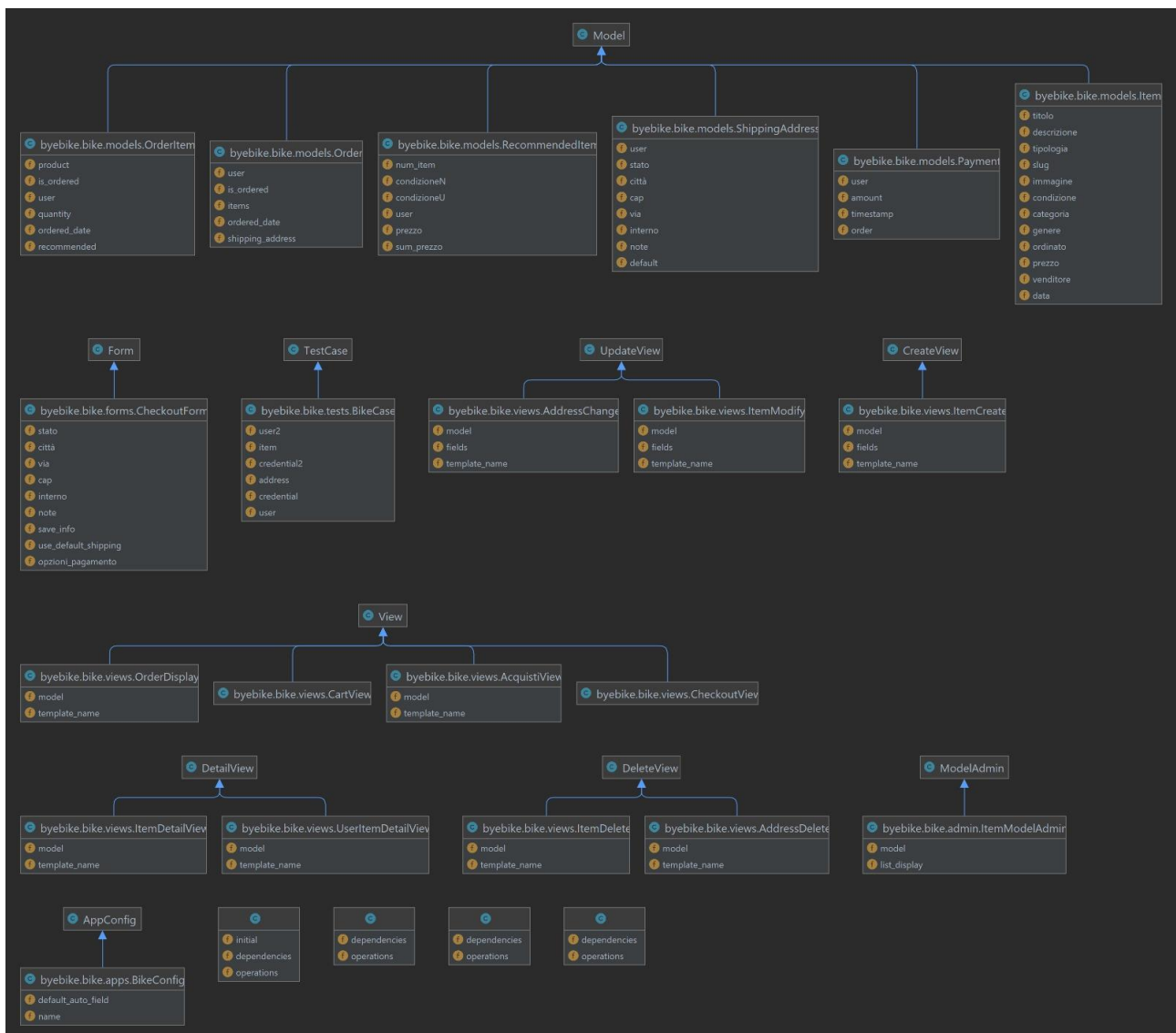
2.1 – Diagramma dei casi d'uso



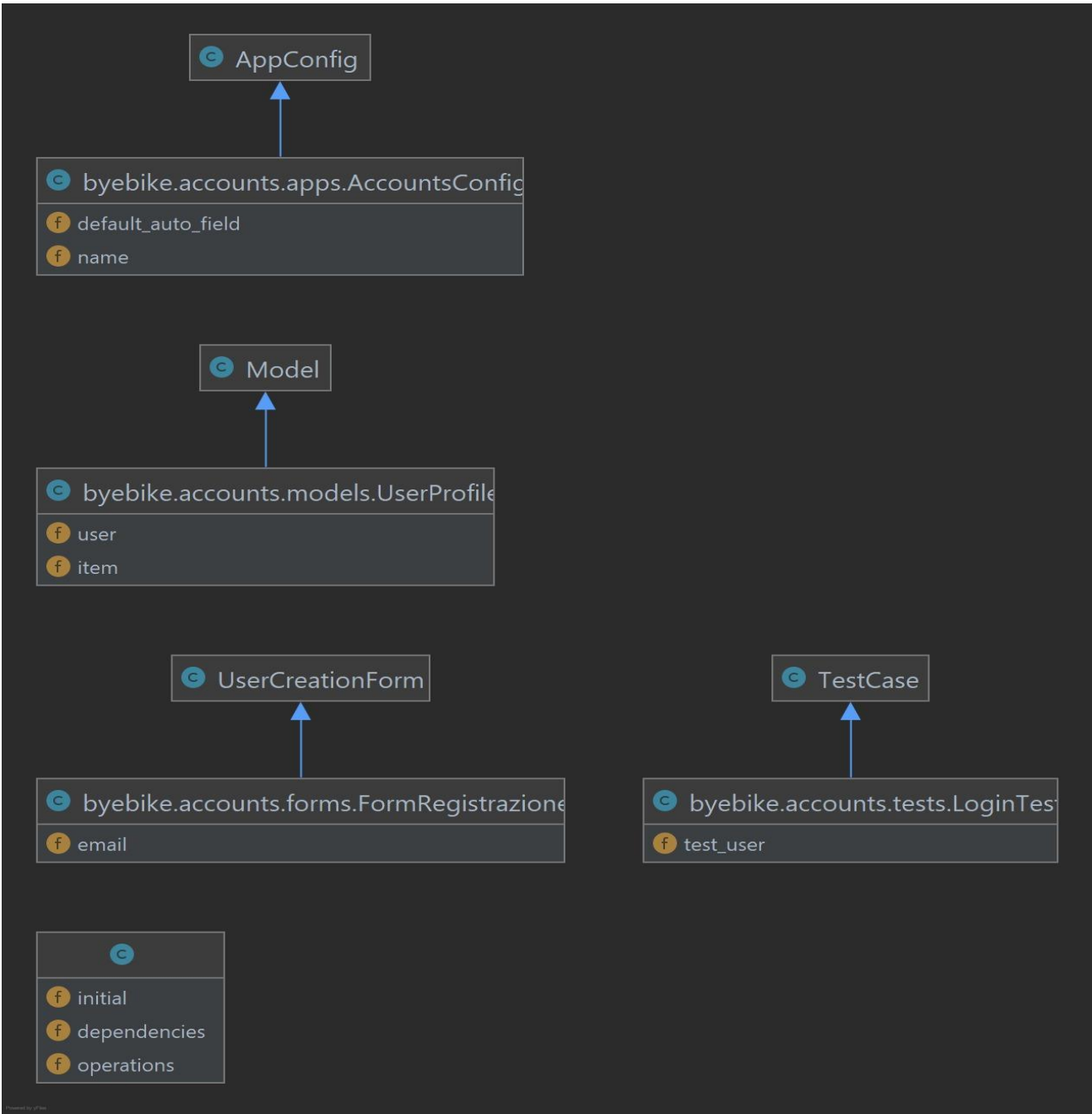
2.2 – Diagramma di utilizzo



2.3 – Diagramma delle classi-Bike



2.4- Diagramma delle classi –Account



3– Tecnologie utilizzate

3.1 – Django

Il progetto è basato su Django, un Framework MVC (Model View Controller) per lo sviluppo Web.

Django di default presenta un'interfaccia di amministrazione che permette di gestire comodamente il database dell'applicazione.

Uno dei punti caratteristici di Django è il fatto che si possono riutilizzare i componenti: cioè permette di dividere il progetto in varie app in modo da facilitare la riusabilità e la leggibilità del codice.

La parte Controller è realizzata tramite i file urls.py, che permettono di mappare le URL sulle opportune richieste dall'utente.

La parte View tratta di funzioni Python che gestiscono il flusso dell'applicazione; in particolar modo in questa parte si definiscono i comportamenti che avranno le pagine in base alle azioni dell'utente.

La parte Model di Django è implementata direttamente come collezione di classi Python che andranno a rappresentare le tabelle del database.

3.2 – Javascript

Javascript è un linguaggio di programmazione interpretato e orientato agli oggetti.

È conosciuto come linguaggi di scripting client-side per pagine web e viene utilizzato principalmente per il design delle funzionalità delle pagine web e per stabilire il comportamento delle pagine web quando viene invocato un evento da parte dell'utente.

3.3 – Bootstrap

Bootstrap è un framework e rappresenta una delle soluzioni più utilizzate per la progettazione dei template per le pagine web, soprattutto per quanto riguarda lo sviluppo *responsive*, ovvero quando sono presenti dei contenuti dinamici.

Permette inoltre di creare progetti in maniera estremamente rapida e veloce.

3.4 – Ajax

Ajax è una tecnica di sviluppo software per la realizzazione di applicazione web interattive.

Lo sviluppo di applicazioni HTML con Ajax si basa su uno scambio asincrono di dati in background fra web browser e server, la quale consente di aggiornare dinamicamente una pagina web senza esplicito ricaricamento da parte dell'utente.

Lo scambio di dati è quindi asincrono perché essi vengono richiesti al server e caricati in background senza interferire con il comportamento della pagina esistente.

4. Descrizione del progetto

Ho progettato un sito e-commerce di vendita di articoli specializzati

L'obiettivo è quello di cercare e naturalmente di vendere dei prodotti in maniera semplice e veloce.

Il progetto è suddiviso nelle seguenti app :

- **accounts**, nella quale viene gestita la parte relativa agli account degli utenti, quali registrazione e gestione degli indirizzi
- **bike**, nella quale viene gestita tutta la parte relativa alle varie pagine del sito, ad esempio la homepage e le vendite degli utenti ecc.
Le maggiori difficoltà sono state riscontrate durante l'implementazione di questa applicazione.
- **E-commerce Bike**, applicazione di base dalla quale parte tutto il progetto

4.1 – Tipologie di utenti presenti

- **Guest** : non è autenticato e ha la possibilità di visitare liberamente il sito, vedere e cercare i prodotti in vendita e i loro dettagli; non potrà invece vedere la sua lista dei consigliati o acquistare prodotti finché non sarà registrato.
- **Utente/Acquirente**: dopo essersi registrato attraverso l'apposito tasto, questo tipo di utente può fare qualsiasi cosa: dall'acquistare i prodotti, alla visualizzazione di una pagina di "articoli consigliati" creata per lui sulla base dei suoi acquisti, sulla media dei prezzi degli articoli acquistati e sulla condizione degli oggetti.
- **Utente/Venditore**: dopo aver contattato l'admin ed essersi accreditato, può vendere direttamente i suoi prodotti sul sito. Un utente/venditore ovviamente, non può acquistare i prodotti da lui messi in vendita.
- **Admin** : L' admin è colui che si occupa della gestione del sito da dietro le quinte; esso potrà eliminare articoli (che non sono più in vendita), autorizzare l'accredito dell'utente/venditore, controllare gli articoli inseriti sul sito che siano inerenti alle richieste, controllare le registrazioni degli utenti/Acquirenti, movimenti relativi ad ordini e pagamenti inseriti sul sito.

4.2 – Gestione degli ordini

Solo da utenti registrati e loggati possono acquistare i prodotti;

Quando l'Utente/acquirente aggiunge prodotti al carrello si creerà un nuovo ordine altrimenti se il carrello è vuoto apparirà il messaggio: "Il carrello è vuoto."

E' stata creata una pagina "carrello" dove vengono evidenziati i prodotti inseriti dall'utente acquirente. Nella pagina apparirà il totale degli oggetti pronti all'acquisto, un pulsante con elimina prodotto (per rimuovere i prodotti inseriti nel carrello) e altri due pulsanti, uno per il checkout e uno continuare con lo shopping (ritorno alla home).

Nel caso in cui l'utente decide di procedere con il checkout, verrà reindirizzato in un'altra pagina nella quale inserirà i dati relativi alla spedizione e alla fatturazione: nome/cognome, indirizzo di spedizione (Via, Città, CAP, Stato, Interno(opzionale) eventuali note aggiuntive, opzionali e il metodo di pagamento.

L'utente dopo aver inoltrato l'ordine, si creerà sul suo profilo un'altra pagina riassuntiva con tutti gli ordini effettuati descritti con numero d'ordine (globale), data dell'ordine e il totale; in più è inoltre presente un pulsante che permette di vedere i dettagli dell'ordine: prodotti acquistati, prezzo e il totale dell'ordine.

4.3 – Vendere un prodotto

La vendita di un prodotto può essere effettuata, come nel caso dell'acquisto, solo da utenti registrati, loggati e autorizzati dall'Admin.

Questa azione può essere eseguita attraverso il pulsante “Vendi un articolo” presente nella pagina del profilo utente.

Il venditore dovrà inserire tutti i dettagli del prodotto quali nome, descrizione, immagine rappresentante il prodotto, condizione dell'oggetto, luogo in cui si trova l'oggetto e CAP; tutti questi campi sono obbligatori.

Una volta che il prodotto viene messo in vendita, sarà visibile da chiunque sul sito.

L'utente potrà quindi vedere il proprio oggetto sia tra gli articoli in vendita all'interno del sito che sul suo profilo, dalla quale ha anche la possibilità di modificarlo o rimuoverlo. Nel momento in cui l'oggetto viene venduto, verrà riportato nella sezione “Vendite” all'interno del profilo utente.

4.4 – Homepage

Nella homepage vengono visualizzati soprattutto gli ultimi articoli inseriti, se l'utente è loggato e una sezione relativa ai “Consigliati per te”.

Nella barra di navigazione troviamo i pulsanti per la registrazione e il login di un utente; qualora l'utente è loggato, apparirà tutta la parte con le informazioni relative al suo profilo quale venditore/acquirente, profilo, cambio password e logout.

5. Recommendation System

Ho incluso nel progetto un sistema di raccomandazione che consiste nel mostrare ad un utente ulteriori articoli da poter acquistare.

Troviamo anche una view “RecommendedItem” che tiene conto degli acquisti precedentemente effettuati, in particolar modo degli articoli acquistati e della media dei totali relativi agli acquisti effettuati.

Questo modello tiene conto del numero di articoli acquistati, del numero di articoli con condizione *Nuova* e *Usata* e della media dei prezzi degli articoli acquistati (salvare i dati degli ordini di un utente)

Si crea una lista di raccomandazioni nella quale vengono aggiunti gli articoli che rispettano determinati canoni di raccomandazione, in questo caso una combinazione tra condizione e prezzo dell’articolo.

Quindi:

- si controlla se il prezzo dell’articolo rientra, ad esempio, nella fascia di media > 100 € oppure < 100 €
- si controlla la condizione dell’articolo e il numero di articoli ordinati per ogni condizione
- se questi requisiti sono soddisfatti, l’articolo viene aggiunto alla lista “Consigliati per te” e vengono mostrati nella homepage dell’utente a partire da quello con il prezzo minore.

```
if (request.user.is_authenticated):
    recommended_items = RecommendedItem.objects.filter(user=request.user)
    for it in object_list:
        for recommended in recommended_items:
            if (it.venditore != request.user):
                if ((it.prezzo >= recommended.prezzo and it.prezzo <= recommended.prezzo + 100) or (
                    it.prezzo <= recommended.prezzo and it.prezzo >= recommended.prezzo - 100)):
                    if ((recommended.condizioneN == recommended.condizioneU) and (
                        recommended.condizioneN >= 1 or recommended.condizioneU >= 1)):
                        if (it not in recommendation_list):
                            recommendation_list.append(it)
                    if ((recommended.condizioneN > recommended.condizioneU) and (
                        recommended.condizioneN >= 1 or recommended.condizioneU >= 1)):
                        if (it.condizione == 'N'):
                            if (it not in recommendation_list):
                                recommendation_list.append(it)
                        else:
                            if ((recommended.condizioneN < recommended.condizioneU) and (recommended.condizioneN >= 1 or recommended.condizioneU >= 1)):
                                if (it.condizione == 'U'):
                                    if (it not in recommendation_list):
                                        recommendation_list.append(it)

for it in object_list:
    if (it.prezzo <= 700 and it.condizione == 'N'):
        newconvenient_list.append(it)
    if (it.prezzo <= 500 and it.condizione == 'U'):
        usedconvenient_list.append(it)
```

6 – Test

6.1 – Test Login (accounts)

Il login di un utente viene controllato attraverso il “Test login” che verifica la correttezza dell’inserimento delle credenziali; provato sia con credenziali corrette che errate.

Naturalmente se le credenziali vengono inserite correttamente, si effettua il login.

```
'''
Test per il login di un utente
Prima viene testato con delle credenziali errate, poi con credenziali corrette
'''
class BikeCase(TestCase):

    def setUp(self):
        self.user = User.objects.create_user(username='test', password='ecommerce', email='test@test.com')
        self.credential = {'username': 'test', 'password': 'ecommerce'}
        self.item = Item.objects.create(titolo="Bicicletta",
                                         descrizione="Bicicletta",
                                         tipologia="B",
                                         immagine="n",
                                         categoria="C",
                                         condizione="N",
                                         genere="U",
                                         prezzo="300",
                                         venditore=self.user)

        self.address = ShippingAddress.objects.create(user=self.user,
                                                       città="Modena",
                                                       stato="O",
                                                       cap="48521",
                                                       via="n",
                                                       interno="3",
                                                       note="ciao")

        self.user2 = User.objects.create_user(username='test2', email='test2@test.com', password='ecommerce')
        self.credential2 = {'username': 'test2', 'password': 'ecommerce'}
```

6.2- Test inserimento articolo (core)

Attraverso il “Test inserimento articolo in vendita” verifica che i campi vengano inseriti correttamente e che i campi siano tutti completi (nessun campo vuoto).

Naturalmente se le credenziali vengono inserite correttamente, si effettua il login.

```
def test_item_create(self):
    '''
    Verifico che nell' inserimento di un articolo i campi obbligatori siano rispettati
    '''
    self.client.login(**self.credential)
    response = self.client.post('/createitem/', {})
    self.assertFormError(response, 'form', 'titolo', 'Questo campo è obbligatorio.')
    self.assertFormError(response, 'form', 'descrizione', 'Questo campo è obbligatorio.')
    self.assertFormError(response, 'form', 'tipologia', 'Questo campo è obbligatorio.')
    self.assertFormError(response, 'form', 'immagine', 'Questo campo è obbligatorio.')
    self.assertFormError(response, 'form', 'categoria', 'Questo campo è obbligatorio.')
    self.assertFormError(response, 'form', 'condizione', 'Questo campo è obbligatorio.')
    self.assertFormError(response, 'form', 'genere', 'Questo campo è obbligatorio.')
    self.assertFormError(response, 'form', 'prezzo', 'Questo campo è obbligatorio.')

    self.assertTemplateUsed(response, 'bike/item_create.html')
    self.assertEqual(response.status_code, 200) # verifica per capire se il template utilizzato è quello corretto

    response = self.client.post('/createitem/',
                                {'titolo': 'Bicicletta Rosa', 'descrizione': 'rosa', 'tipologia': 'U', 'immagine': 's', 'categoria': 'C', 'condizione': 'U', 'genere': 'U', 'prezzo': '1250'})
    self.assertTemplateUsed(response, 'bike/item_create.html')
    self.assertEqual(response.status_code, 200)
```

6.3 – Test reindirizzamento login (core)

Attraverso il “Test re-indirizzamento” verifica che l’utente per mettere in vendita un articolo, deve essere loggato, nel caso contrario viene reindirizzato alla pagina di login. Nel caso sia loggato vi è il controllo della correttezza dell’inserimento dell’articolo.

```
'''
Test per verificare se all' inserimento di un articolo, un utente non loggato
viene reindirizzato prima nella schermata di login.
'''
def test_login_required(self):
    response = self.client.get('/createitem/')
    self.assertRedirects(response, '/accounts/login/?next=/createitem/')
    # 302 --> FOUND: pagina esiste ma non puoi entrarci
    self.assertEqual(response.status_code, 302)
```

6.4 – Test cambio indirizzo utente (core)

Attraverso il “Test cambio indirizzo utente” verifica il corretto cambiamento dei campi dell’indirizzo di un utente e che l’azione possa essere fatta dall’utente che ha creato in precedenza il vecchio indirizzo.

```
'''
Test per il cambio dell' indirizzo di un utente
'''
def test_address_change(self):
    # con utente autenticato
    self.client.login(**self.credential)
    response = self.client.get('/user/address/' + str(self.address.id) + '/modify/')
    self.assertEqual(response.status_code, 200)
    response = self.client.post('/user/address/' + str(self.address.id) + '/modify/', {})
    self.assertFormError(response, 'form', 'città', 'Questo campo è obbligatorio.')
    self.assertFormError(response, 'form', 'via', 'Questo campo è obbligatorio.')
    self.assertFormError(response, 'form', 'stato', 'Questo campo è obbligatorio.')
    self.assertFormError(response, 'form', 'cap', 'Questo campo è obbligatorio.')
    response = self.client.post('/user/address/' + str(self.address.id) + '/modify/',
                                {'città': 'chieti', 'via': 'strada', 'stato': 'it', 'cap': '85710'})
    self.assertRedirects(response, '/user/' + self.user.username + '/address_page/')
    self.client.logout()

    # con utente autenticato ma non creatore
    self.client.login(**self.credential2)
    response = self.client.get('/user/address/' + str(self.address.id) + '/modify/')
    self.assertTemplateUsed(response, 'bike/homepage.html')
    self.assertTemplateNotUsed(response, 'accounts/address_change.html')
```

6.5– Test eliminazione indirizzo utente (core)

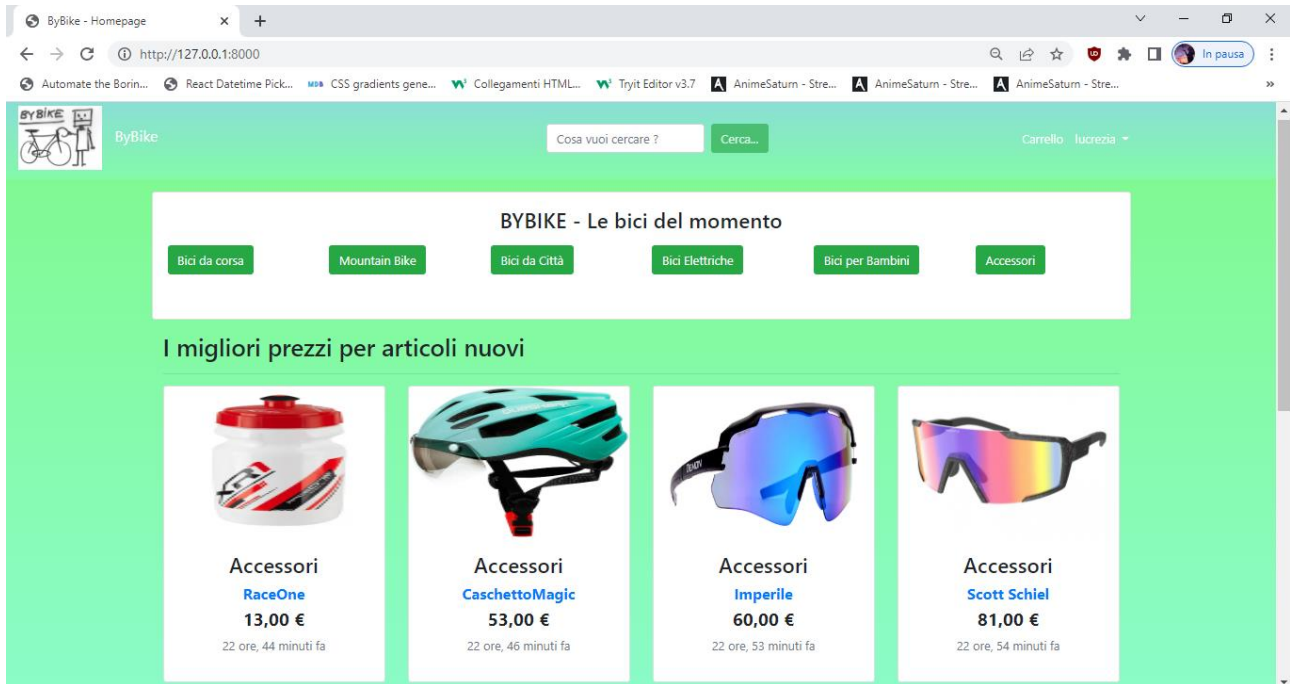
Attraverso il “Test eliminazione indirizzo utente” verifica che l’utente possa eliminare il suo indirizzo di spedizione.

```
'''
Test per l' eliminazione di un indirizzo dell' utente
'''
def test_address_delete(self):
    # con utente autenticato
    self.client.login(**self.credential)
    response = self.client.get('/user/address/' + str(self.address.id) + '/delete/')
    self.assertEqual(response.status_code, 200)
    response = self.client.post('/user/address/' + str(self.address.id) + '/delete/', {})
    self.assertRedirects(response, '/user/' + self.user.username + '/address_page/')
    self.client.logout()

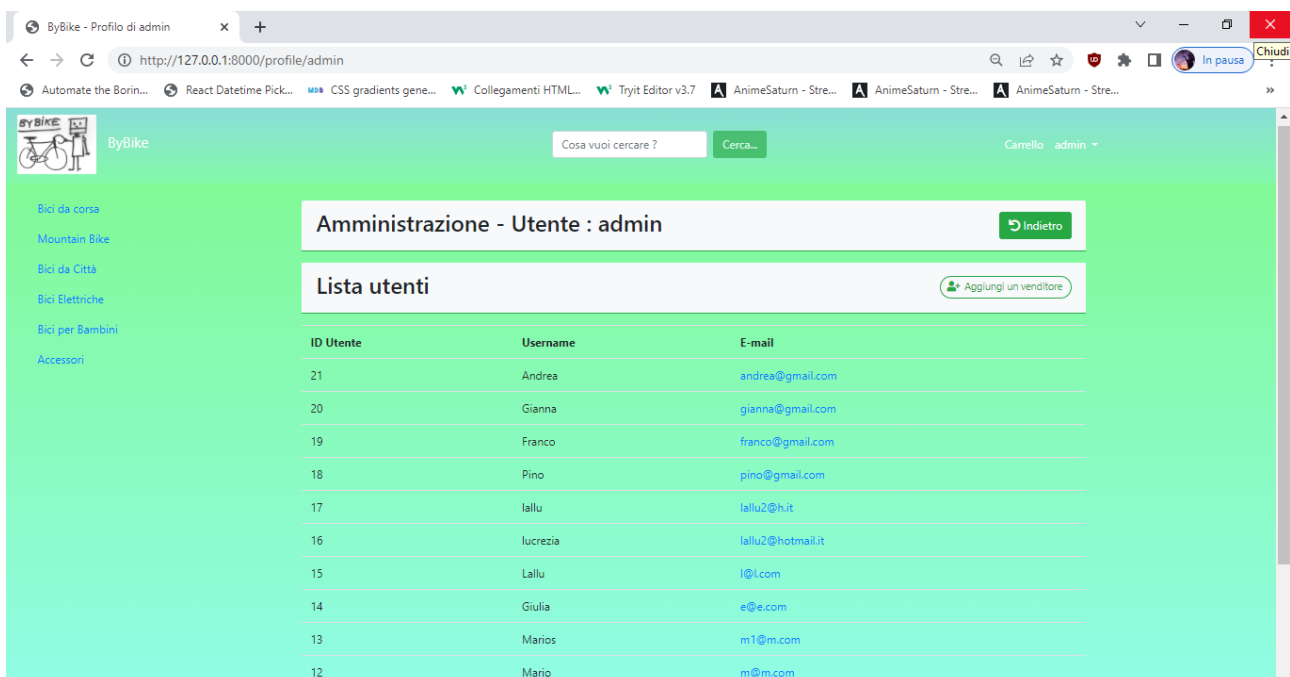
    # con utente autenticato ma non creatore
    self.client.login(**self.credential2)
    response = self.client.get('/user/address/' + str(self.address.id) + '/delete/')
    self.assertTemplateNotUsed(response, 'accounts/address_delete.html')
```

7- Screenshot

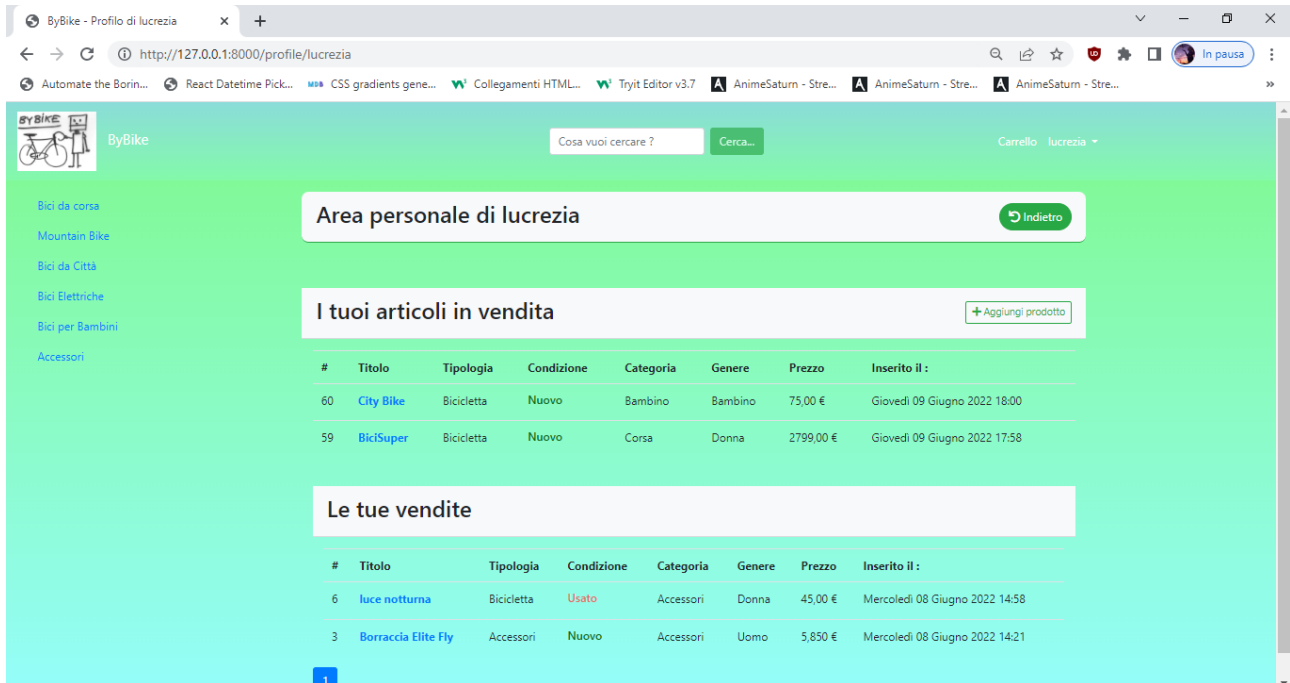
7.1 - Homepage



7.2 - Profilo utente admin



7.3 – Profilo utente venditore con riepilogo delle vendite



ByBike

Cosa vuoi cercare? Cerca...

Carrello lucrezia

Area personale di lucrezia

Indietro

I tuoi articoli in vendita

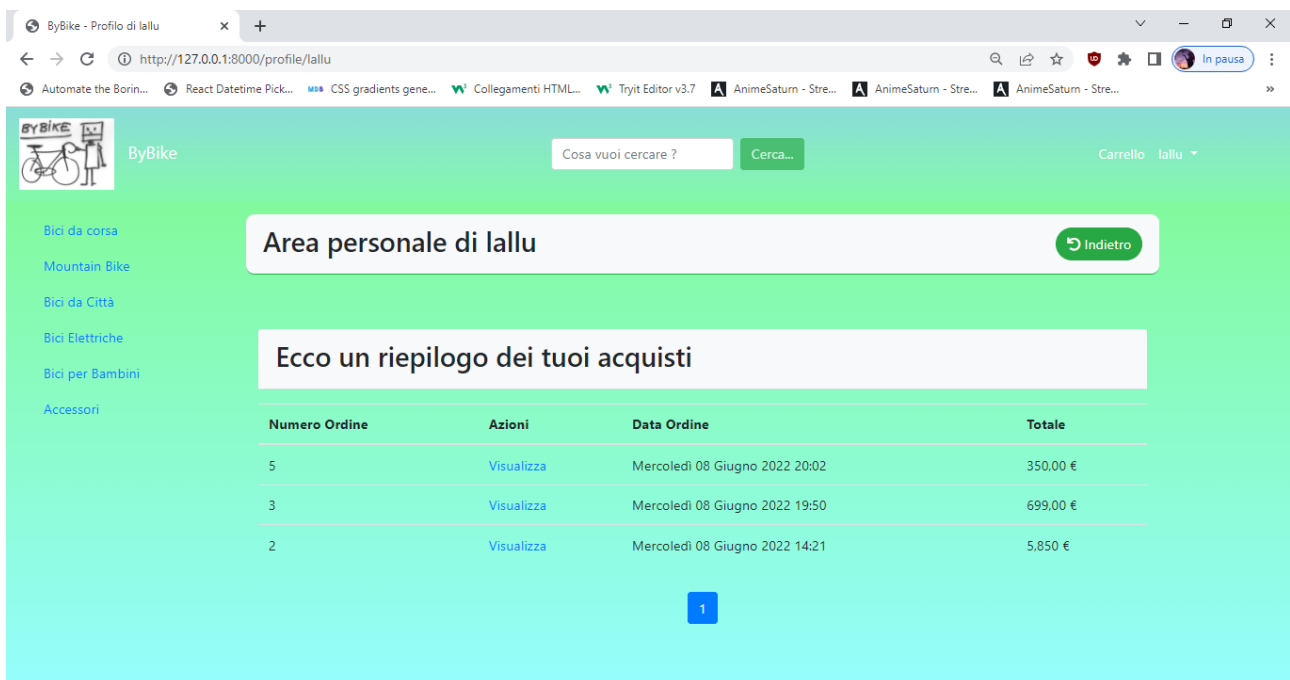
Aggiungi prodotto

#	Titolo	Tipologia	Condizione	Categoria	Genere	Prezzo	Inserito il :
60	City Bike	Bicicletta	Nuovo	Bambino	Bambino	75,00 €	Giovedì 09 Giugno 2022 18:00
59	BiciSuper	Bicicletta	Nuovo	Corsa	Donna	2799,00 €	Giovedì 09 Giugno 2022 17:58

Le tue vendite

#	Titolo	Tipologia	Condizione	Categoria	Genere	Prezzo	Inserito il :
6	luce notturna	Bicicletta	Usato	Accessori	Donna	45,00 €	Mercoledì 08 Giugno 2022 14:58
3	Borraccia Elite Fly	Accessori	Nuovo	Accessori	Uomo	5,850 €	Mercoledì 08 Giugno 2022 14:21

7.4 – Profilo di un utente compratore



ByBike

Cosa vuoi cercare? Cerca...

Carrello lallu

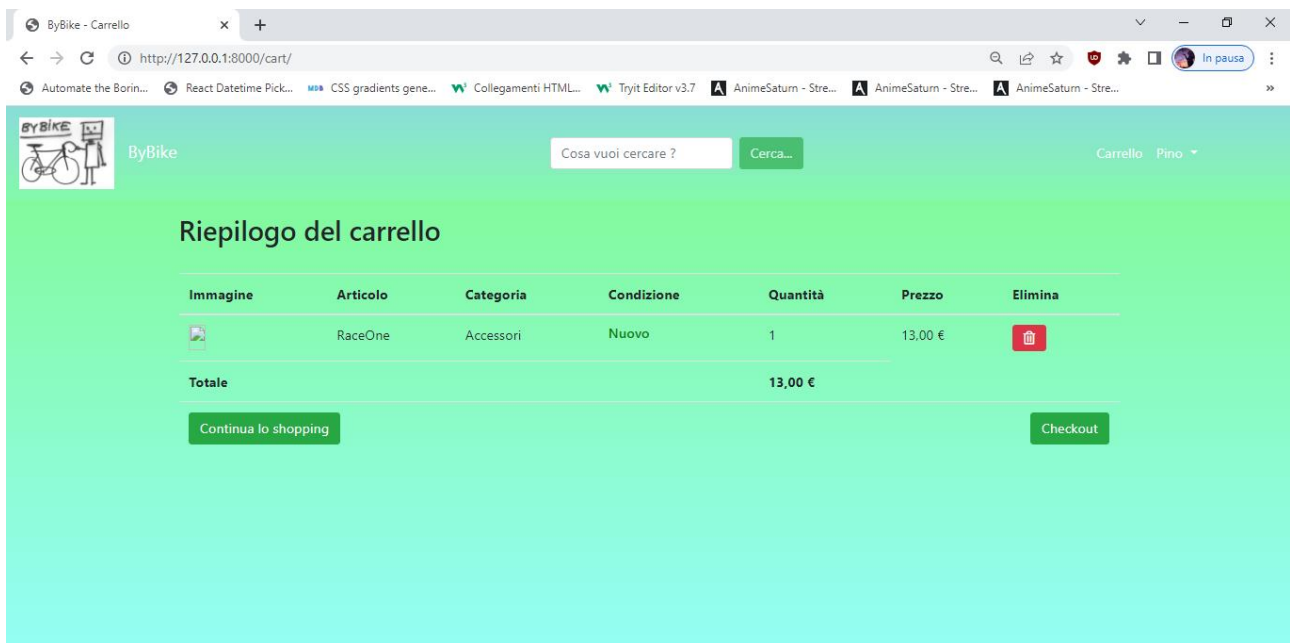
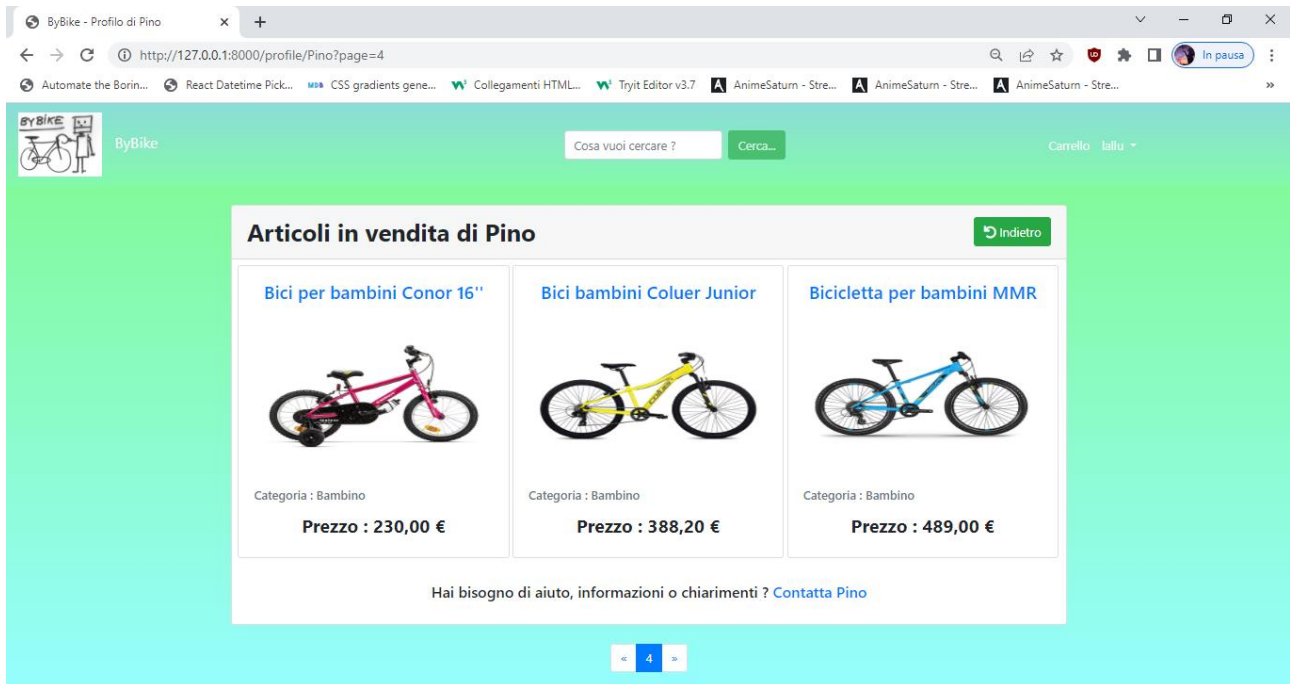
Area personale di lallu

Indietro

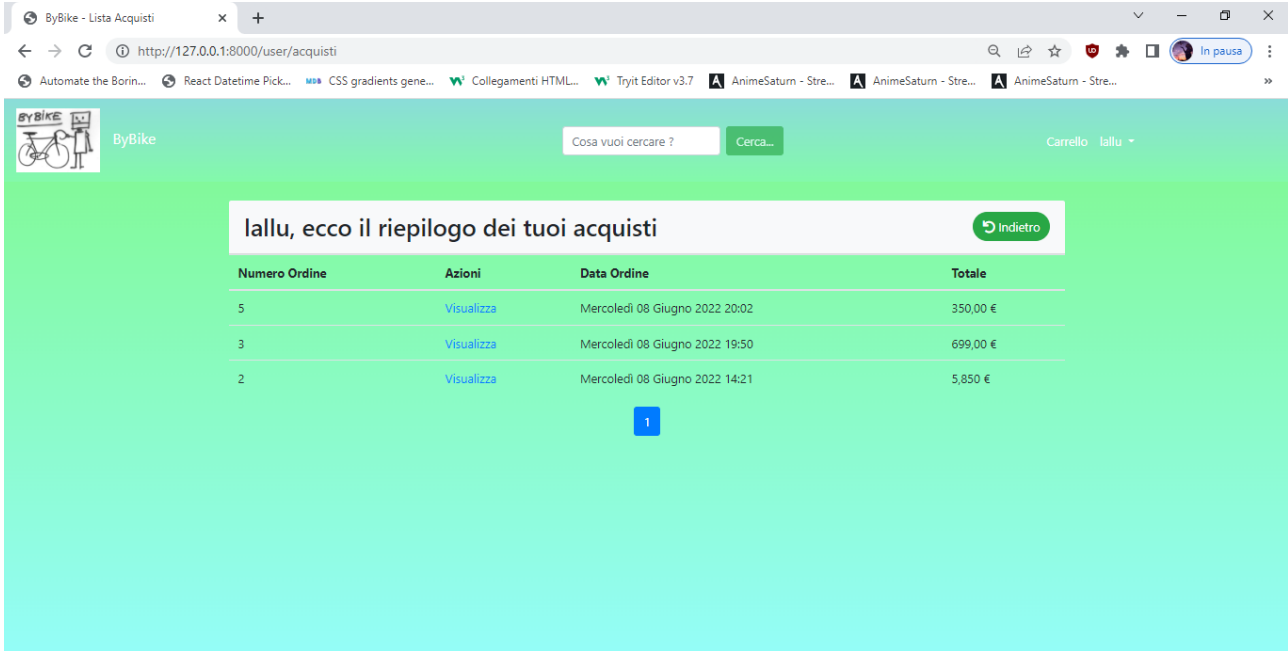
Ecco un riepilogo dei tuoi acquisti

Numero Ordine	Azioni	Data Ordine	Totale
5	Visualizza	Mercoledì 08 Giugno 2022 20:02	350,00 €
3	Visualizza	Mercoledì 08 Giugno 2022 19:50	699,00 €
2	Visualizza	Mercoledì 08 Giugno 2022 14:21	5,850 €

7.5 – Visualizzazione del carrello



7.6 – Riepilogo degli acquisti di un utente



ByBike

Cosa vuoi cercare ? Cerca...

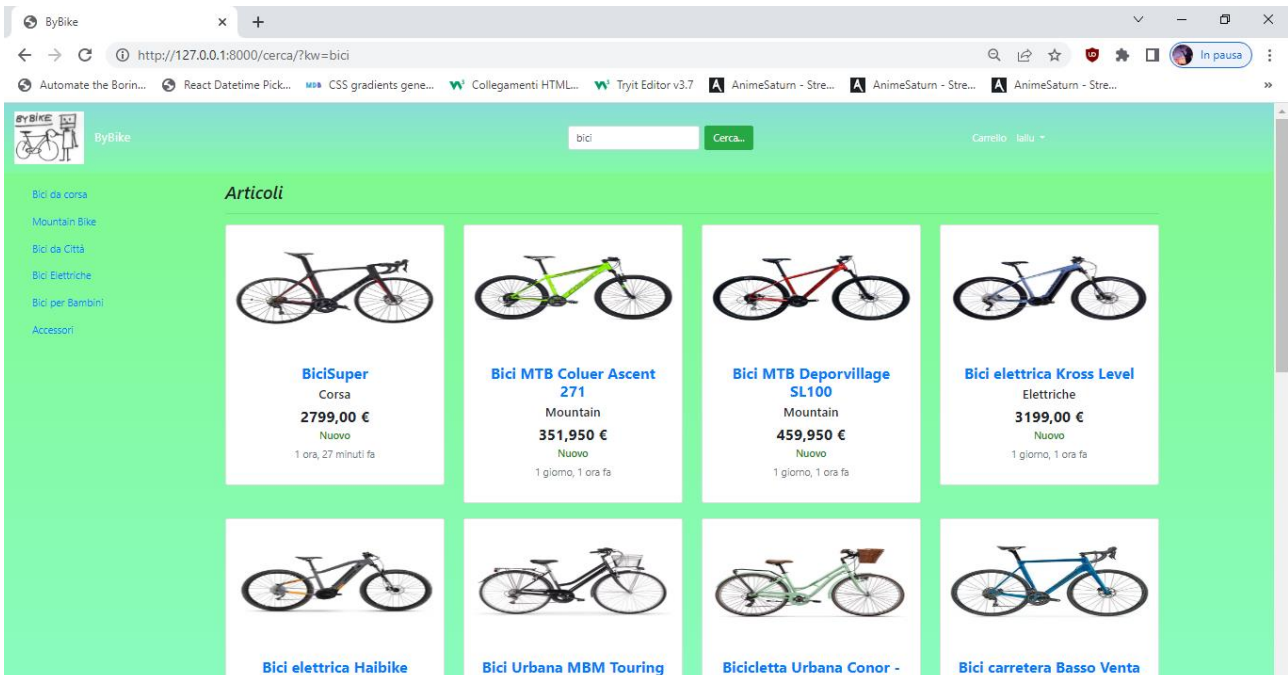
Carrello - Iallu -

Iallu, ecco il riepilogo dei tuoi acquisti [Indietro](#)

Numero Ordine	Azioni	Data Ordine	Totale
5	Visualizza	Mercoledì 08 Giugno 2022 20:02	350,00 €
3	Visualizza	Mercoledì 08 Giugno 2022 19:50	699,00 €
2	Visualizza	Mercoledì 08 Giugno 2022 14:21	5,850 €

1

7.7 – Risultati di una ricerca




ByBike

bici Cerca...


Carrello - Iallu -

Articoli


- [Bici da corsa](#)
- [Mountain Bike](#)
- [Bici da Città](#)
- [Bici Elettriche](#)
- [Bici per Bambini](#)
- [Accessori](#)




BiciSuper
Corsa
2799,00 €
Nuovo
1 ora, 27 minuti fa




Bici MTB Coluer Ascent 271
Mountain
351,950 €
Nuovo
1 giorno, 1 ora fa




Bici MTB Deporvillage SL100
Mountain
459,950 €
Nuovo
1 giorno, 1 ora fa




Bici elettrica Kross Level
Elettriche
3199,00 €
Nuovo
1 giorno, 1 ora fa




Bici elettrica Haibike
Elettriche
2.999,00 €
Nuovo
1 giorno, 1 ora fa



Bici Urbana MBM Touring
Mountain
1.999,00 €
Nuovo
1 giorno, 1 ora fa



Bicicletta Urbana Conor
Mountain
1.999,00 €
Nuovo
1 giorno, 1 ora fa



Bici carretera Basso Venta
Corsa
1.999,00 €
Nuovo
1 giorno, 1 ora fa

7.8 – Pagina di Checkout

ByBike - Checkout

http://127.0.0.1:8000/checkout/

Automate the Borin... React Datetime Pick... CSS gradients gene... Collegamenti HTML... Tryit Editor v3.7 AnimeSaturn - Stre... AnimeSaturn - Stre... AnimeSaturn - Stre...

ByBike

Cosa vuoi cercare ? Cerca...

Carrello: Pino

Indirizzo - Compila tutti i campi di seguito riportati

Stato/Regione

Città/Paese

CAP

Codice postale

Via e numero civico

Strada, P.O. box, ecc.

Interno

Numero appartamento

Appartamento, suite, etc.

Eventuali note

Ulteriori precisazioni

Nome campanello, orari di consegna, etc.

☐ Salva questo indirizzo come default

☐ Utilizza l'indirizzo di default con città

Pagamento - Seleziona un metodo di pagamento

☐ Pagamento alla consegna

☐ Bonifico

[Torna al carrello](#) [Invia l'ordine](#)

7.9 – Visualizzazione articolo

ByBike - Casco Mavic Crossride SL

http://127.0.0.1:8000/product/casco-mavic-crossride-sl

Automate the Borin... React Datetime Pick... CSS gradients gene... Collegamenti HTML... Tryit Editor v3.7 AnimeSaturn - Stre... AnimeSaturn - Stre... AnimeSaturn - Stre...


ByBike

Cosa vuoi cercare ? Cerca...

Carrello: lallu

Casco Mavic Crossride SL

Inserito : Mercoledì 08 Giugno 2022 17:08 | Venditore : [Pino](#)



Categoria : [Accessori](#)

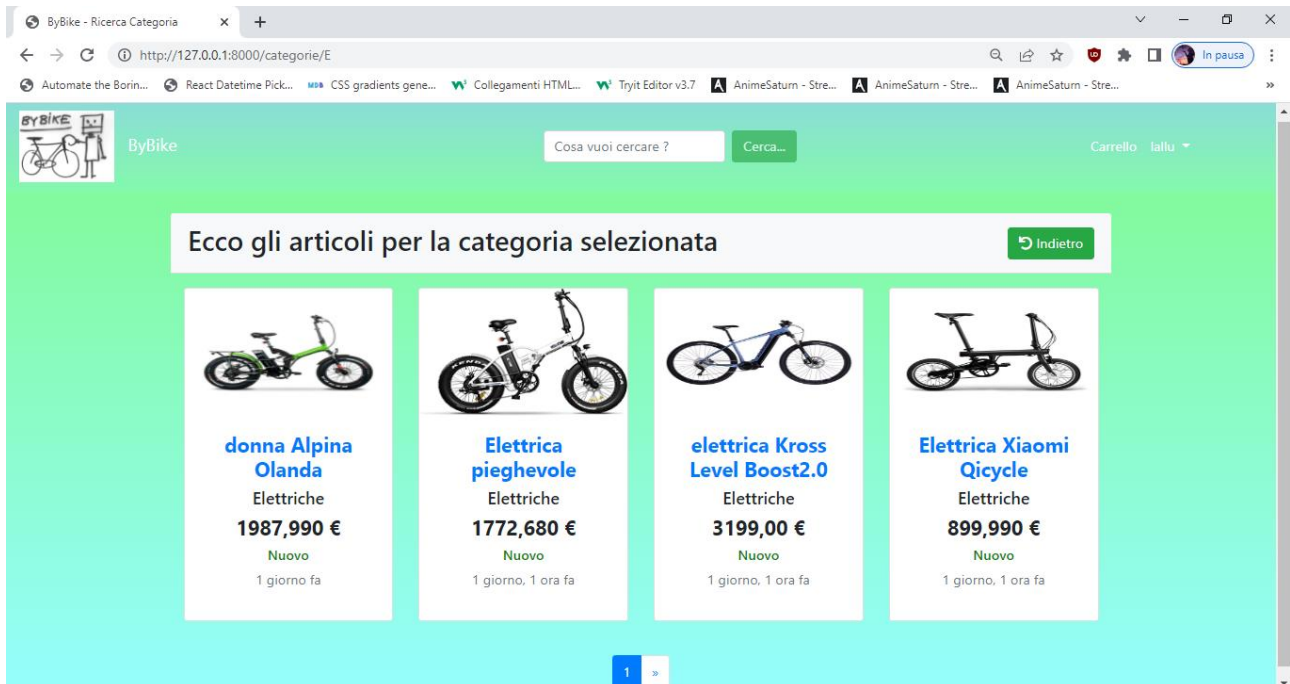
Descrizione : Casco Elite nero per ciclisti professionisti.

Condizione : Nuovo

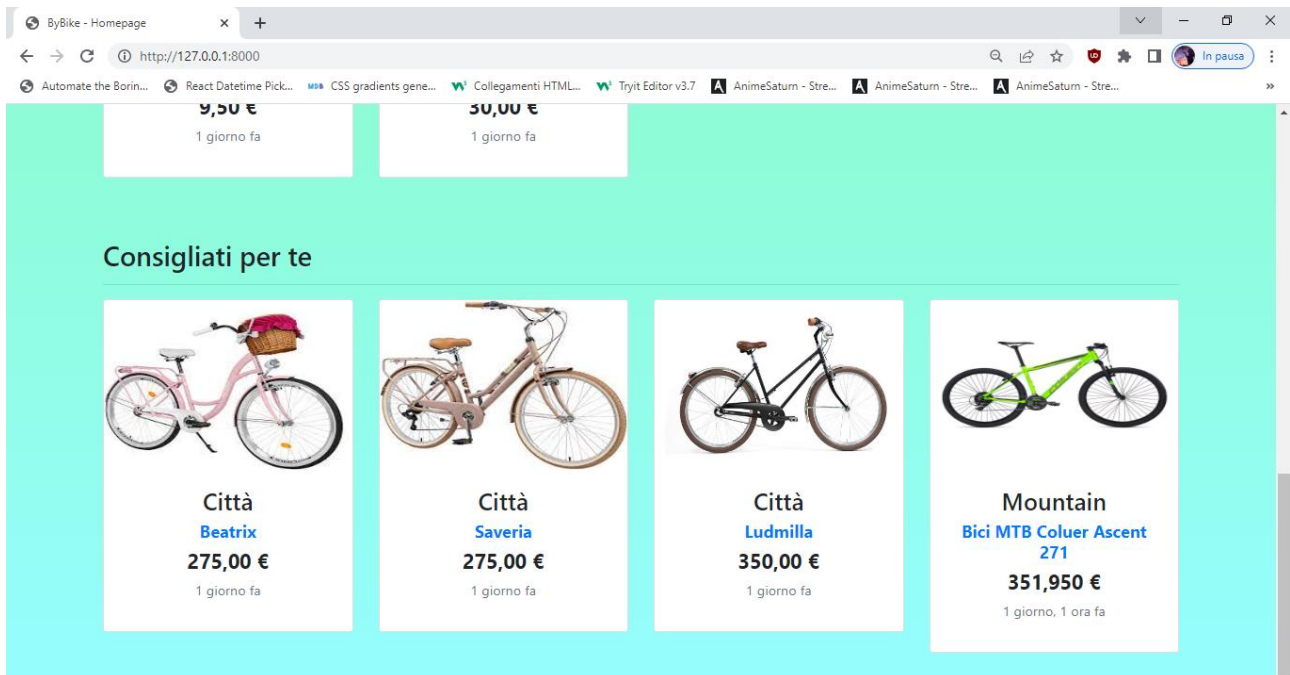
Prezzo : 57,60 €

[Aggiungi al carrello](#) [Indietro](#)

7.10 – Pagina in base alla categoria selezionata



7.11 – Recommendation System



8 – Conclusioni

Al termine di questo percorso posso definirmi molto soddisfatta; è stata la prima volta che ho utilizzato Django e che ho elaborato un progetto di questo tipo da sola ed in autonomia.

È stata una importante occasione di crescita sia a livello di apprendimento di contenuti che di esperienza professionale.

Ho potuto imparare l'utilizzo di metodi di lavoro, di tecniche e dei framework.

Il progetto ha richiesto molto tempo e anche molta pazienza nel "fare, rifare, revisionare", ma anche questi sono elementi di positività e di ricchezza portati "a casa" al termine dell'esperienza.

Sono infine molto soddisfatta di quanto sono riuscita a realizzare e produrre.

Avrei potuto aggiungere altri aspetti e altre soluzioni (ad esempio le procedure di reso e di rimborso).