



Università Ca' Foscari - Venezia

Relazione del Progetto di Basi di Dati

Piattaforma di E-commerce

Massimo Costantini - 877336

Filippo Tiozzo - 887683

Samuel

Anno Accademico 2023 - 2024

Indice	2
1 Introduzione	3
1.1 Applicazione	3
1.2 Blueprint	3
1.3 Caricamento grant e trigger	3
2 Funzionalità principali	4
2.1 Esecuzione dell'applicazione	
2.2 Requisiti del progetto	
3 Progettazione concettuale e logica della basi di dati	
3.1 Modello Grafico ad Oggetti	
3.2 Tabelle	
3.3 Relazioni	
3.4 Modello Logico Relazionale	
4 Query principali	
5 Principali scelte progettuali	
5.1 Autorizzazioni	
5.2 Trigger	
6 Ulteriori informazioni	
6.1 Comandi utili	
7 Contributo al progetto	

1 Introduzione

Il progetto scelto riguarda la Piattaforma di E-commerce, una web application sviluppata in linguaggio Python mediante il framework Flask, ed interfacciata con il database relazionale PostgreSQL utilizzando la libreria SQLAlchemy.

In questo documento sono descritte le funzionalità principali dell'applicazione, la progettazione concettuale e logica della base di dati, le query e le scelte progettuali che sono state fatte.

1.1 Applicazione

L'interfaccia web permette di registrarsi e collegarsi come acquirente oppure come venditore fornendo una serie di possibili azioni, diverse a seconda del ruolo, come ad esempio la creazione da parte del venditore di nuovi prodotti da vendere e la scelta da parte dell'acquirente di prodotti da acquistare.

1.2 Blueprint

L'applicazione è suddivisa in Blueprint Flask, un insieme di route, cioè di percorsi URL, che gestiscono le chiamate HTTP.

In questo modo il codice sorgente è suddiviso in moduli (i file acquirente.py, autorizzazioni.py, modelli.py e venditore.py) che facilitano così la gestione del progetto.

1.3 Caricamento grant e trigger

Per il caricamento dei grant e dei trigger necessari al funzionamento dell'applicazione, sono stati inseriti all'interno del progetto due file chiamati grant.sql e trigger.sql.

2 Funzionalità principali

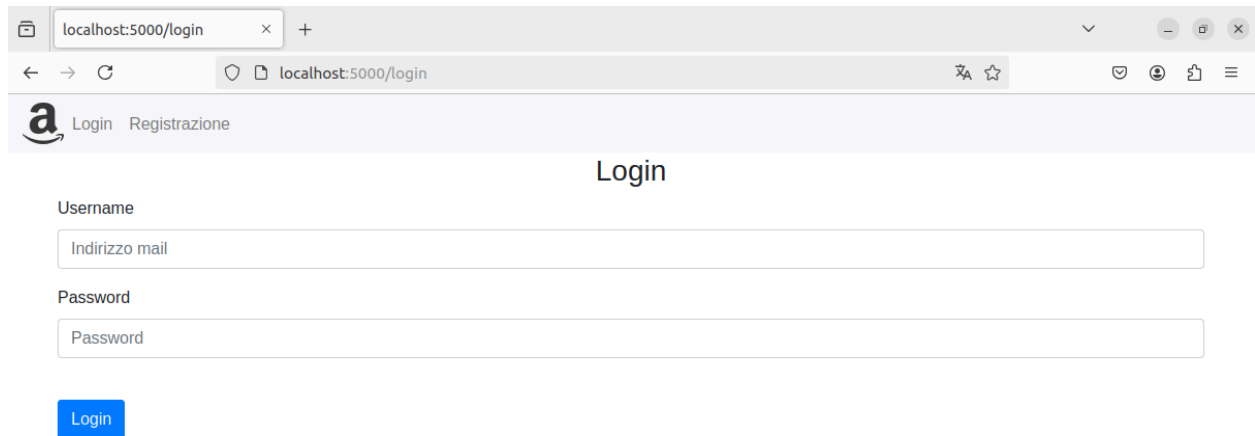
La scelta del progetto Piattaforma di E-commerce ci ha permesso di sviluppare l'applicazione suddividendola opportunamente nelle due parti acquirente e venditore, assegnando un ruolo diverso durante la registrazione dell'utente per poter poi assegnare ad esso le funzionalità corrispondenti.

2.1 Esecuzione dell'applicazione

Di seguito una breve spiegazione di come viene eseguita l'applicazione in base ai requisiti suggeriti.

Si avvia andando direttamente alla pagina di login, dove si ha la possibilità di effettuare il login o la registrazione di un nuovo utente.

In figura 1 la schermata relativa al login di un utente:

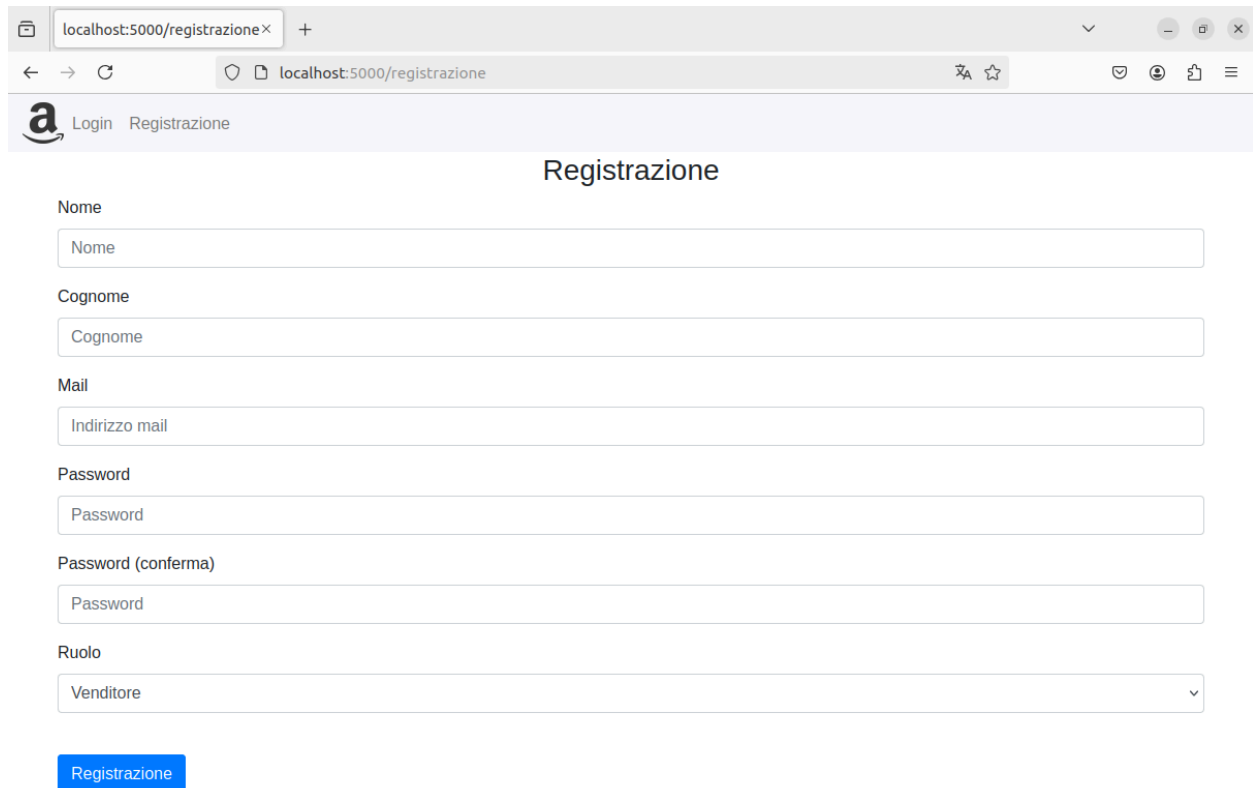


The screenshot shows a web browser window with the address bar displaying 'localhost:5000/login'. The page features a header with an Amazon logo and navigation links for 'Login' and 'Registrazione'. The main section is titled 'Login' and contains two input fields: 'Username' (with placeholder text 'Indirizzo mail') and 'Password' (with placeholder text 'Password'). Below these fields is a blue button labeled 'Login'.

Figura 1: Login utente

Effettuando la registrazione di un nuovo utente si dovrà scegliere il ruolo per il quale ci si registra, acquirente o venditore, andando poi direttamente alla homepage dedicata in base al ruolo scelto.

In figura 2 la schermata relativa alla registrazione di un nuovo utente:



The screenshot shows a web browser window with the address bar displaying 'localhost:5000/registrazione'. The page has a header with a logo and links for 'Login' and 'Registrazione'. The main content area is titled 'Registrazione' and contains a form with the following fields:

- Nome: A text input field with the placeholder 'Nome'.
- Cognome: A text input field with the placeholder 'Cognome'.
- Mail: A text input field with the placeholder 'Indirizzo mail'.
- Password: A text input field with the placeholder 'Password'.
- Password (conferma): A text input field with the placeholder 'Password'.
- Ruolo: A dropdown menu with 'Venditore' selected.

At the bottom of the form is a blue button labeled 'Registrazione'.

Figura 2: Registrazione utente

Gli acquirenti possono quindi scegliere i prodotti da acquistare.

I venditori possono creare nuovi prodotti, modificando o cancellando quelli esistenti.

2.2 Requisiti del progetto

Tutti i requisiti previsti per lo sviluppo del progetto per la Piattaforma di E-commerce sono stati soddisfatti.

3 Progettazione concettuale e logica della basi di dati

In questo paragrafo vengono descritti lo schema grafico ad oggetti, le tabelle, le relazioni ed il modello logico relazionale della base di dati.

3.1 Modello Grafico ad Oggetti

In figura 3 il modello grafico ad oggetti:

Figura 3: Modello grafico ad oggetti

3.2 Tabelle

In questa sezione vengono descritte le tabelle utilizzate per modellare la base di dati.

Le tabelle sono definite come classi ORM (Object Relational Mapping) di SQLAlchemy.

Si utilizzano così funzioni Python per manipolare i dati invece di scrivere direttamente codice SQL.

Definizione classe (tabella) Utenti

```
class Utenti(db.Model, UserMixin):
```

```
    id = db.Column(db.Integer, primary_key=True, index=True)
    username = db.Column(db.String(50), unique=True, nullable=False)
    password = db.Column(db.String(), nullable=False)
    ruolo = db.Column(db.String(10), index=True)
```

Definizione classe (tabella) Acquirente

```
class Acquirente(db.Model):
```

```
    id = db.Column(db.Integer, primary_key=True)
    nome = db.Column(db.String(50), nullable=False)
    cognome = db.Column(db.String(50), nullable=False)
    user_id = db.Column(db.Integer, db.ForeignKey('utenti.id'), index=True)
    user = db.relationship('Utenti', backref='acquirenti')
```

Definizione classe (tabella) Venditore

```
class Venditore(db.Model):
```

```
    id = db.Column(db.Integer, primary_key=True)
```

```
nome = db.Column(db.String(50), nullable=False)
cognome = db.Column(db.String(50), nullable=False)
user_id = db.Column(db.Integer, db.ForeignKey('utenti.id'), index=True)
user = db.relationship('Utenti', backref='venditori')
```

3.3 Relazioni

In questa sezione vengono descritte le relazioni scelte per modellare la base di dati:

3.4 Modello Logico Relazionale

In figura 4 il modello logico relazionale:

Figura 4: Modello logico relazionale

4 Query principali

In questo paragrafo vengono descritte le query principali utilizzate:

5 Principali scelte progettuali

In questo paragrafo vengono descritte le principali scelte progettuali.

Sono stati innanzitutto creati tre utenti con compiti diversi, amministratore, acquirente e venditore:

```
CREATE USER amministratore WITH PASSWORD 'password';
CREATE USER acquirente WITH PASSWORD 'password';
CREATE USER venditore WITH PASSWORD 'password';
```

L'utente amministratore serve per creare gli utenti e fare il login, quindi ha un accesso limitato solo alle tabelle Utente, Acquirente e Venditore.

5.1 Autorizzazioni

In questa sezione vengono descritte le autorizzazioni che devono essere inserite nel database per poter eseguire l'applicazione:

```
GRANT INSERT, SELECT ON Utente, Acquirente, Venditore TO amministratore;
```

Permette all'utente amministratore di eseguire query di inserimento e lettura sulle tabelle Utenti, Acquirente e Venditore.

```
GRANT ALL PRIVILEGES ON SEQUENCE utenti_id_seq TO amministratore;
```

Concede all'utente amministratore tutti i diritti sulla sequenza utenti_id.

```
GRANT ALL PRIVILEGES ON SEQUENCE acquirente_id_seq TO amministratore;
```

Concede all'utente amministratore tutti i diritti sulla sequenza acquirente_id.

```
GRANT ALL PRIVILEGES ON SEQUENCE venditore_id_seq TO amministratore;
```

Concede all'utente amministratore tutti i diritti sulla sequenza venditore_id.

5.2 Trigger

In questa sezione vengono descritti i trigger che sono stati utilizzati:

6 Ulteriori informazioni

Per la parte di autenticazione è stata utilizzata la libreria flask_login.

Per la parte grafica è stato utilizzato il framework Bootstrap.

In tutte le funzioni, per garantire la robustezza del codice, è stato utilizzato il costrutto try-except-finally.

6.1 Comandi utili

Di seguito un riassunto dei comandi necessari all'utilizzo dell'applicazione:

Creazione del database

```
# sudo -u postgres psql  
postgres=# CREATE DATABASE "progetto";  
postgres=# \q
```

Cancellazione del database

```
# sudo -u postgres psql  
postgres=# DROP DATABASE "progetto";  
postgres=# \q
```

Cambio password di default

```
postgres=# ALTER USER postgres PASSWORD 'Venezia123';
```

Elenco dei database ed accesso

```
# sudo -u postgres psql  
postgres=# \l  
postgres=# \c progetto  
progetto=# \q
```

Riavvio PostgreSQL

```
# service postgresql restart
```

Esecuzione applicazione

```
# export FLASK_APP=main.py  
# flask run
```

In alternativa per l'esecuzione dell'applicazione

```
# python3 main.py
```

Cancellazione tabelle

```
# sudo -u postgres psql
```

```
postgres=# \c progetto
```

```
progetto=# DELETE from utenti;
```

```
progetto=# DELETE from acquirente;
```

```
progetto=# DELETE from venditore;
```

```
progetto=# \q
```

7 Contributo al progetto

Massimo Costantini - 877336

Sviluppo codice (autenticazione)

Documentazione (struttura, schemi, screenshot, revisioni)

Filippo Tiozzo - 887683

Sviluppo codice