DUMITRU Irina-Alexandra
341C3

Fashionstore
Proiect Baze de Date 2

Descriere proiect

Proiectul reprezinta un magazine online de fashion. Se pot regasi 8 categorii de produse. Clientul poate pune orice produs in cosul sau, putand alege, de asemenea, si marimea dorita. Pentru a putea da o comanda, acesta trebuie sa-si fi creat in prealabil un cont de utilizator si sa fie logat.

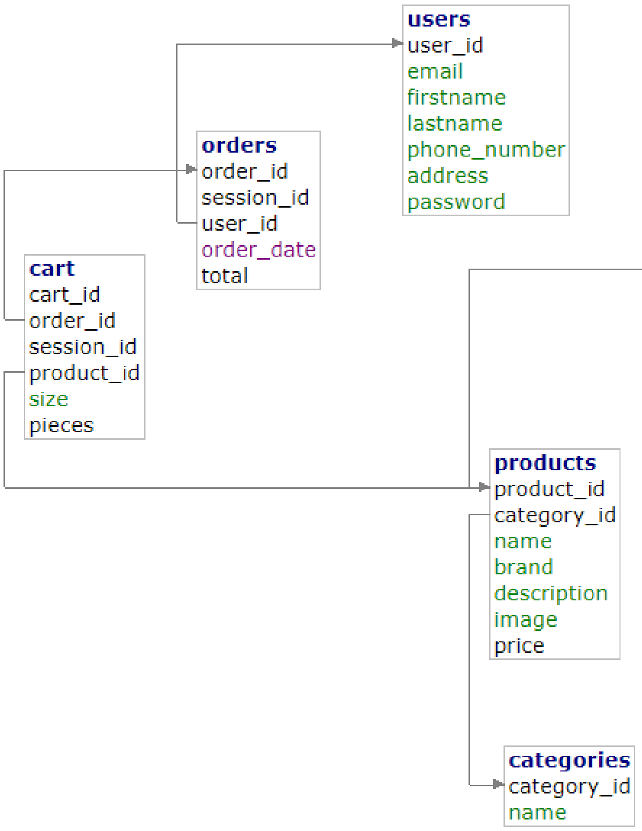
Descrierea bazei de date

Baza de date se numeste FASHIONSTORE si contine 6 tabele. Baza de date este PostgreSQL.

Pentru backend am folosit Python, Flask (server) si docker pentru baza de date (contine si un utlitar pentru a putea vizualiza mai usor datele).

Pentru frontend am folosit HTML, CSS si Javascript.

Diagrama bazei de date



DUMITRU Irina-Alexandra
341C3

Structura tabelelor

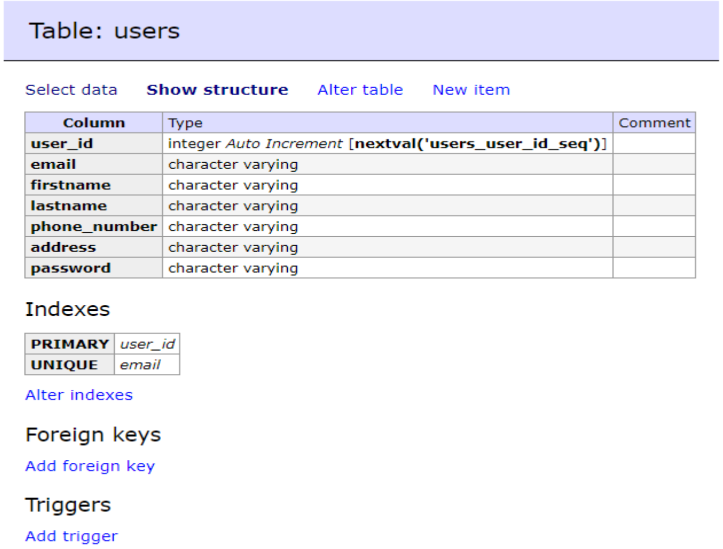


Table: users

Column	Type	Comment
user_id	integer *Auto Increment* [**nextval('users_user_id_seq')**]	
email	character varying	
firstname	character varying	
lastname	character varying	
phone_number	character varying	
address	character varying	
password	character varying	

Indexes

PRIMARY	*user_id*
UNIQUE	*email*

Foreign keys

Triggers

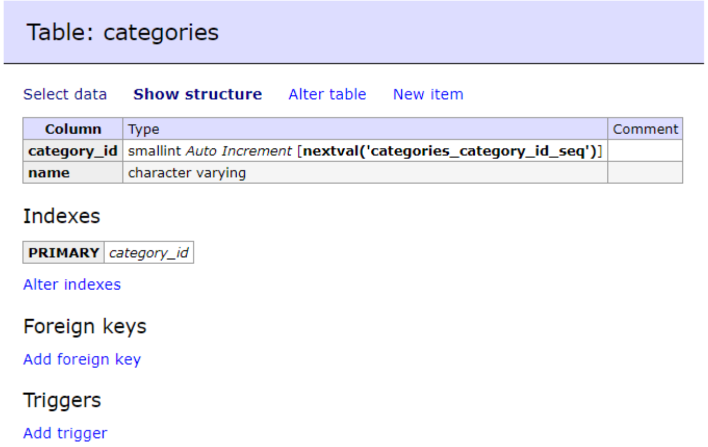


Table: categories

Column	Type	Comment
category_id	smallint *Auto Increment* [**nextval('categories_category_id_seq')**]	
name	character varying	

Indexes

PRIMARY	*category_id*

Foreign keys

Triggers

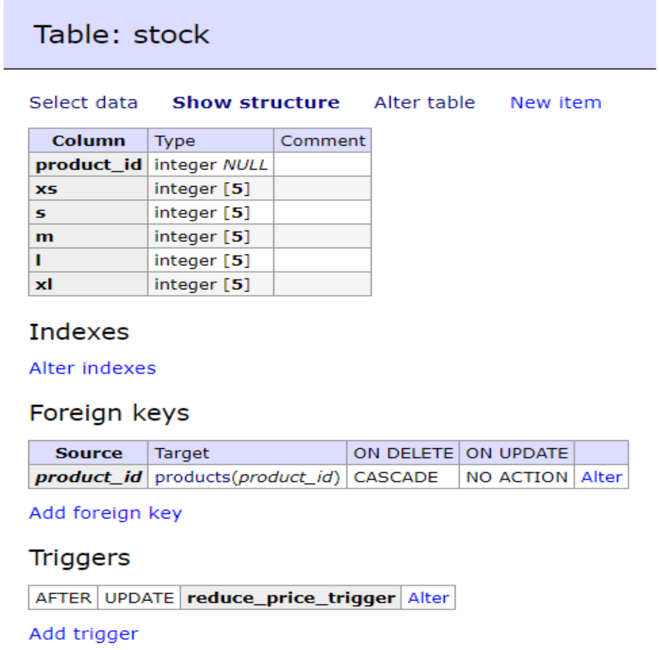


Table: stock

Column	Type	Comment
product_id	integer *NULL*	
xs	integer [**5**]	
s	integer [**5**]	
m	integer [**5**]	
l	integer [**5**]	
xl	integer [**5**]	

Indexes

Foreign keys

Source	Target	ON DELETE	ON UPDATE	
product_id	products(*product_id*)	CASCADE	NO ACTION	Alter

Triggers

AFTER	UPDATE	**reduce_price_trigger**	Alter

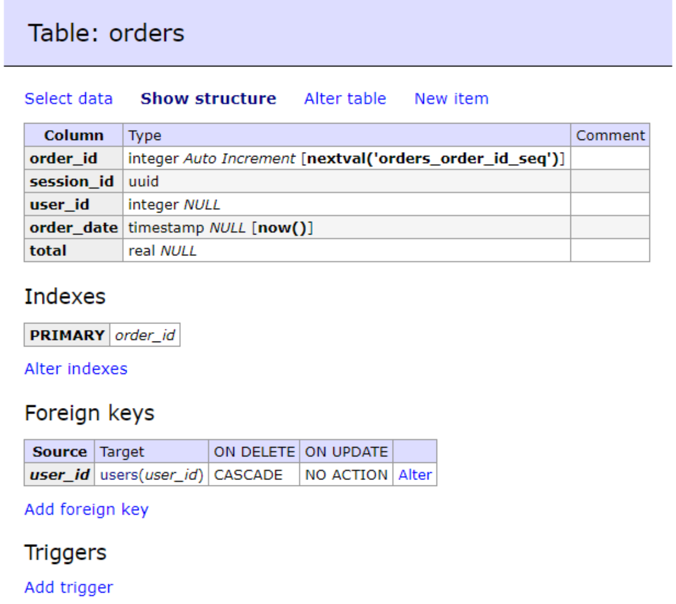


Table: orders

Column	Type	Comment
order_id	integer *Auto Increment* [**nextval('orders_order_id_seq')**]	
session_id	uuid	
user_id	integer *NULL*	
order_date	timestamp *NULL* [**now()**]	
total	real *NULL*	

Indexes

PRIMARY	*order_id*

Foreign keys

Source	Target	ON DELETE	ON UPDATE	
user_id	users(*user_id*)	CASCADE	NO ACTION	Alter

Triggers

DUMITRU Irina-Alexandra
341C3

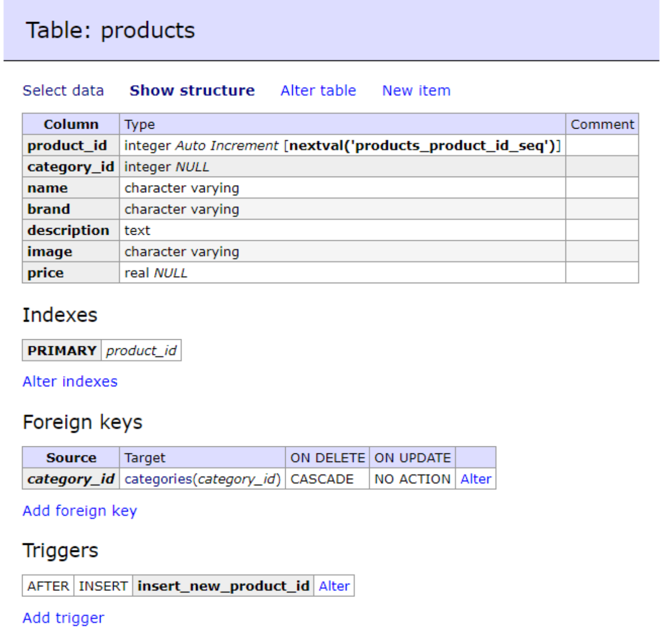


Table: products

Column	Type	Comment
product_id	integer *Auto Increment* [**nextval('products_product_id_seq')**]	
category_id	integer *NULL*	
name	character varying	
brand	character varying	
description	text	
image	character varying	
price	real *NULL*	

Indexes

PRIMARY	*product_id*

Foreign keys

Source	Target	ON DELETE	ON UPDATE	
category_id	categories(*category_id*)	CASCADE	NO ACTION	Alter

Triggers

AFTER	INSERT	**insert_new_product_id**	Alter

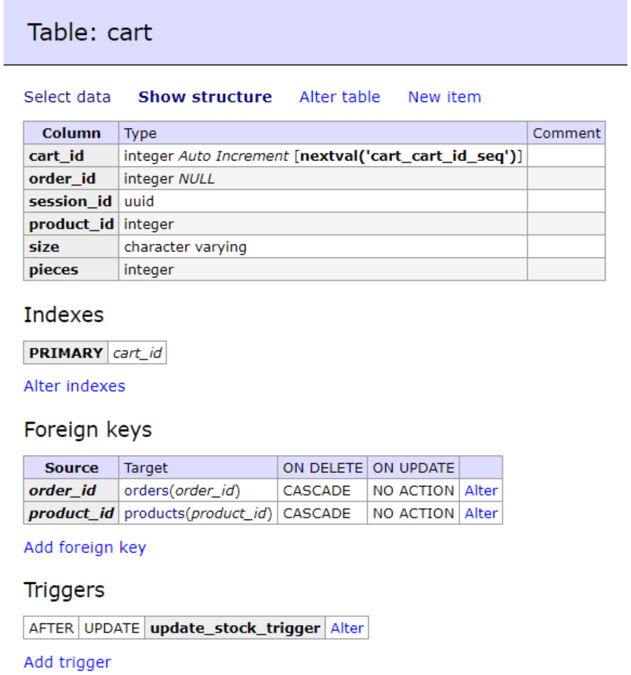


Table: cart

Column	Type	Comment
cart_id	integer *Auto Increment* [**nextval('cart_cart_id_seq')**]	
order_id	integer *NULL*	
session_id	uuid	
product_id	integer	
size	character varying	
pieces	integer	

Indexes

PRIMARY	*cart_id*

Foreign keys

Source	Target	ON DELETE	ON UPDATE	
order_id	orders(*order_id*)	CASCADE	NO ACTION	Alter
product_id	products(*product_id*)	CASCADE	NO ACTION	Alter

Triggers

AFTER	UPDATE	**update_stock_trigger**	Alter

Descrierea constrangerilor de integritate

Tabela USERS are constrangere pe campul email, pentru a nu exista mai multe campuri cu acelasi email si ca si cheie primara are campul user_id, de tipul SERIAL (NOT NULL, AUTO INCREMENT SEQUENCE).

CATEGORIES are cheie primara category_id, de tipul SMALLSERIAL (deoarece range-ul de small int este suficient).

PRODUCTS referentiaza campul category_id din CATEGORIES, iar ca si cheie primara are product_id, SERIAL.

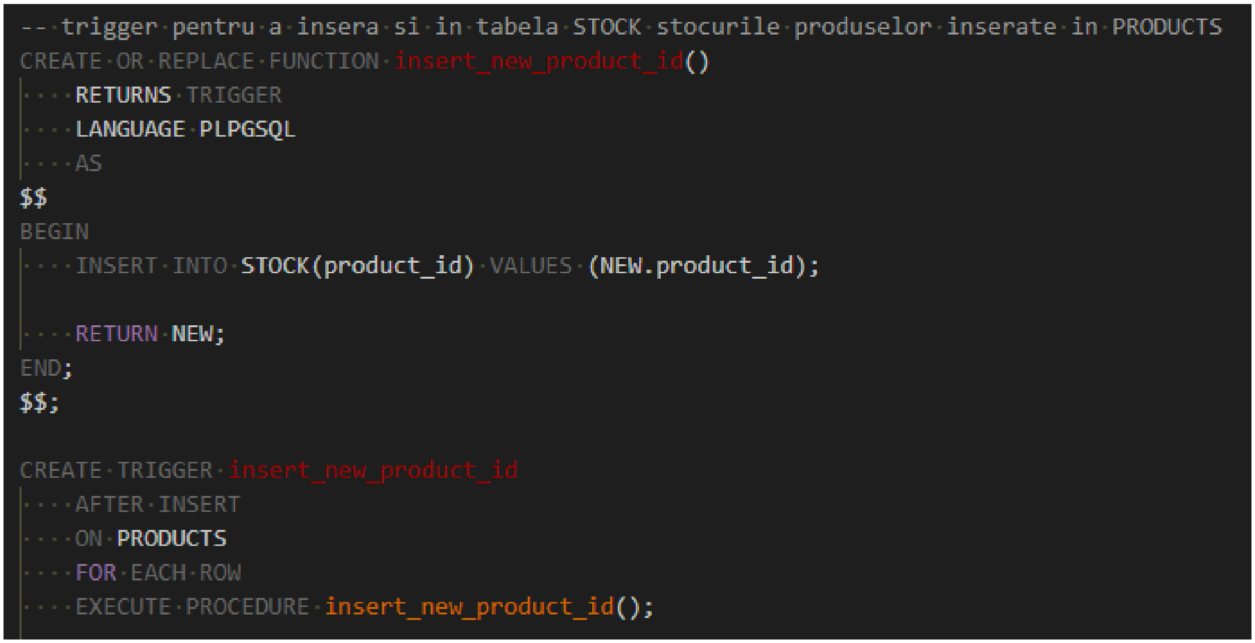
STOCK are FOREIGN KEY product_id din PRODUCTS si contine stocul de produse pentru fiecare marime.

CART contine obiecte adaugate in cos. Va referentia order_id, ce se va introduce la plasarea comenzii.

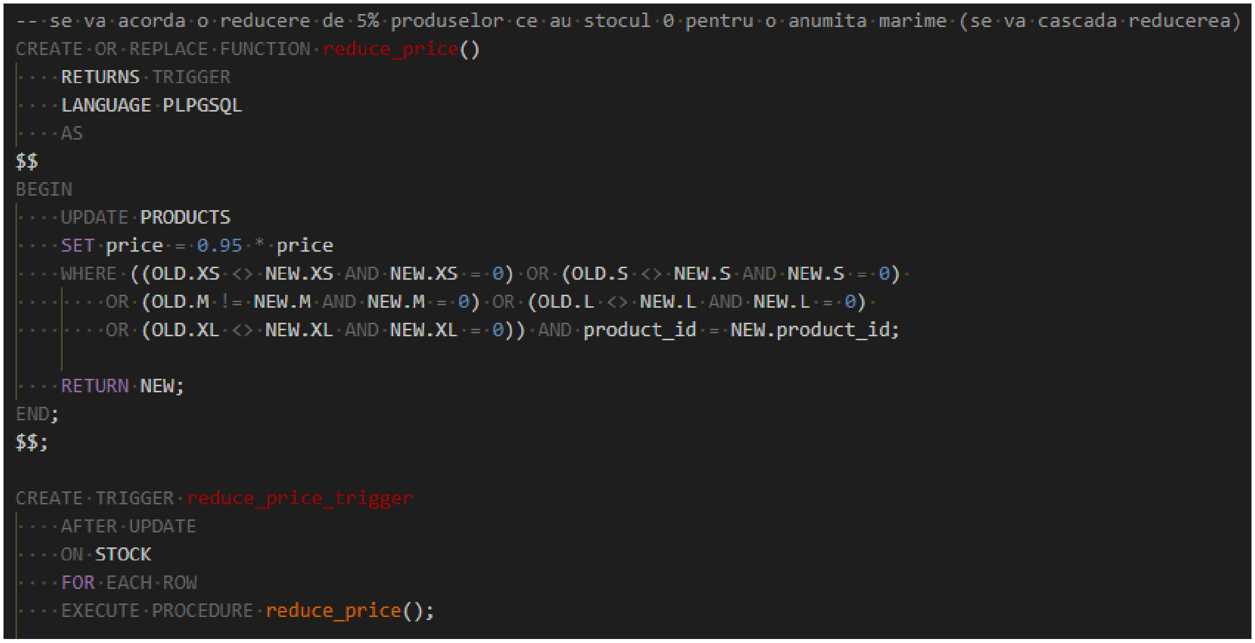
ORDERS va contine subtotalul tuturor comenzilor.

DUMITRU Irina-Alexandra
341C3

Descrierea procedurilor si a functiilor

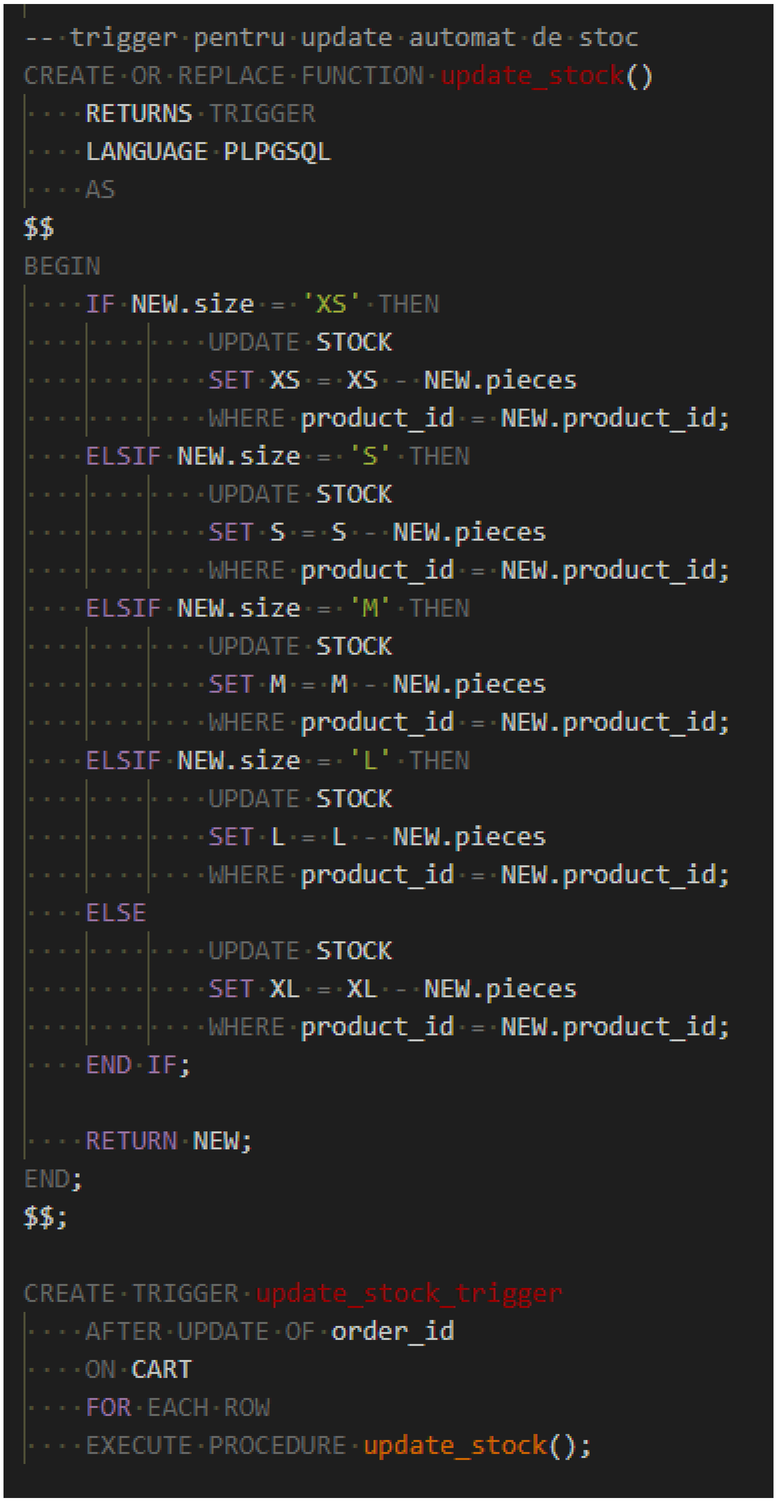


```sql
-- trigger pentru a insera si in tabela STOCK stocurile produselor inserate in PRODUCTS
CREATE OR REPLACE FUNCTION insert_new_product_id()
    RETURNS TRIGGER
    LANGUAGE PLPGSQL
    AS
$$
BEGIN
    INSERT INTO STOCK(product_id) VALUES (NEW.product_id);

    RETURN NEW;
END;
$$;

CREATE TRIGGER insert_new_product_id
    AFTER INSERT
    ON PRODUCTS
    FOR EACH ROW
    EXECUTE PROCEDURE insert_new_product_id();
```



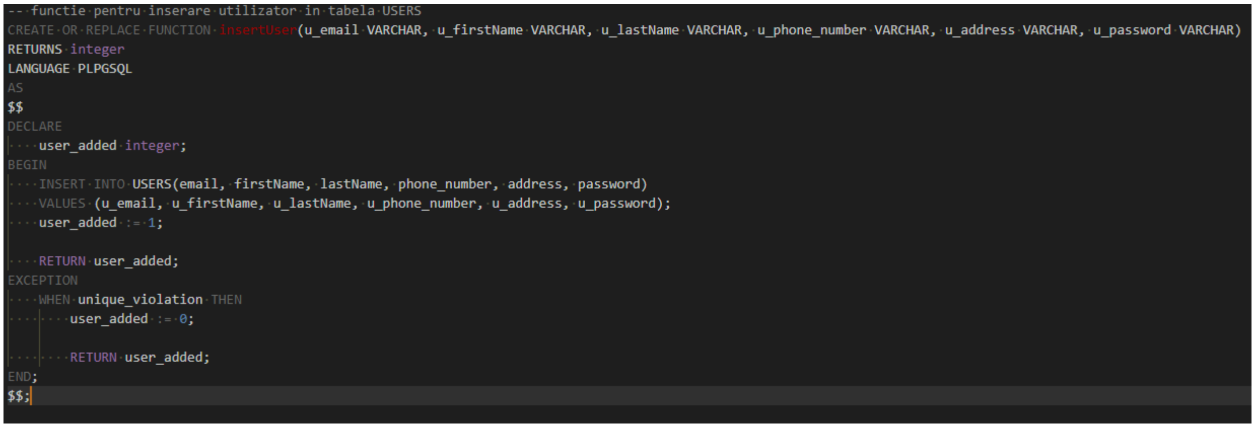
```sql
-- se va acorda o reducere de 5% produselor ce au stocul 0 pentru o anumita marime (se va cascada reducerea)
CREATE OR REPLACE FUNCTION reduce_price()
    RETURNS TRIGGER
    LANGUAGE PLPGSQL
    AS
$$
BEGIN
    UPDATE PRODUCTS
    SET price = 0.95 * price
    WHERE ((OLD.XS <> NEW.XS AND NEW.XS = 0) OR (OLD.S <> NEW.S AND NEW.S = 0)
        OR (OLD.M != NEW.M AND NEW.M = 0) OR (OLD.L <> NEW.L AND NEW.L = 0)
        OR (OLD.XL <> NEW.XL AND NEW.XL = 0)) AND product_id = NEW.product_id;

    RETURN NEW;
END;
$$;

CREATE TRIGGER reduce_price_trigger
    AFTER UPDATE
    ON STOCK
    FOR EACH ROW
    EXECUTE PROCEDURE reduce_price();
```

DUMITRU Irina-Alexandra
341C3

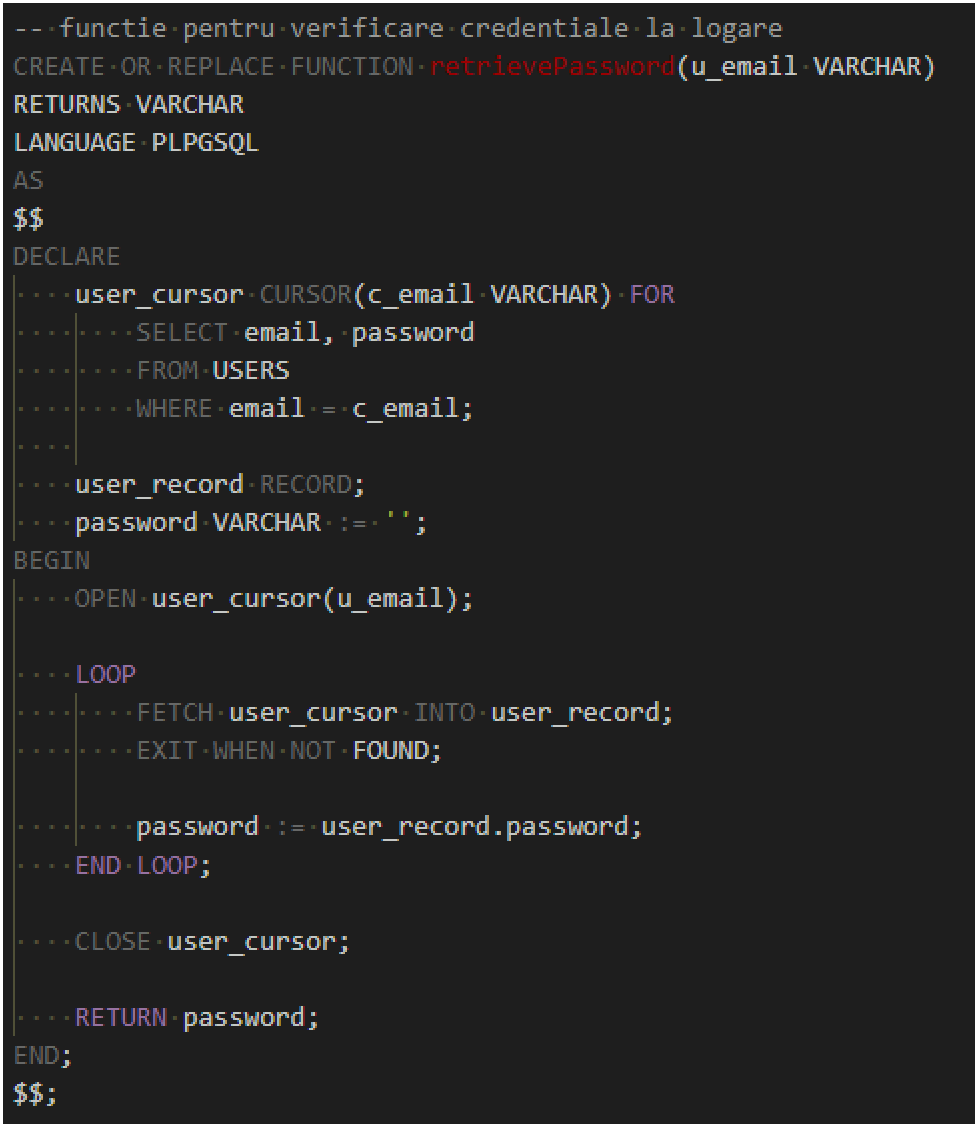


```
-- trigger pentru update automat de stoc
CREATE OR REPLACE FUNCTION update_stock()
    RETURNS TRIGGER
    LANGUAGE PLPGSQL
    AS
$$
BEGIN
    IF NEW.size = 'XS' THEN
            UPDATE STOCK
            SET XS = XS - NEW.pieces
            WHERE product_id = NEW.product_id;
    ELSIF NEW.size = 'S' THEN
            UPDATE STOCK
            SET S = S - NEW.pieces
            WHERE product_id = NEW.product_id;
    ELSIF NEW.size = 'M' THEN
            UPDATE STOCK
            SET M = M - NEW.pieces
            WHERE product_id = NEW.product_id;
    ELSIF NEW.size = 'L' THEN
            UPDATE STOCK
            SET L = L - NEW.pieces
            WHERE product_id = NEW.product_id;
    ELSE
            UPDATE STOCK
            SET XL = XL - NEW.pieces
            WHERE product_id = NEW.product_id;
    END IF;

    RETURN NEW;
END;
$$;

CREATE TRIGGER update_stock_trigger
    AFTER UPDATE OF order_id
    ON CART
    FOR EACH ROW
    EXECUTE PROCEDURE update_stock();
```

DUMITRU Irina-Alexandra
341C3

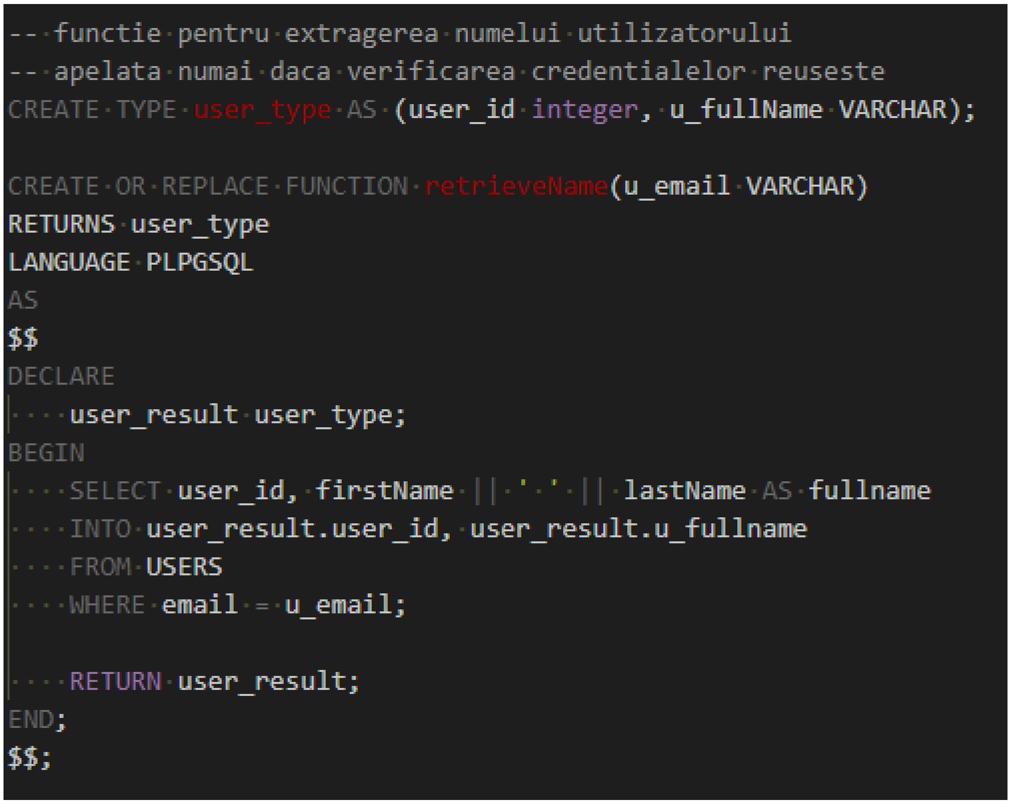


```sql
-- functie pentru inserare utilizator in tabela USERS
CREATE OR REPLACE FUNCTION insertUser(u_email VARCHAR, u_firstName VARCHAR, u_lastName VARCHAR, u_phone_number VARCHAR, u_address VARCHAR, u_password VARCHAR)
RETURNS integer
LANGUAGE PLPGSQL
AS
$$
DECLARE
    user_added integer;
BEGIN
    INSERT INTO USERS(email, firstName, lastName, phone_number, address, password)
    VALUES (u_email, u_firstName, u_lastName, u_phone_number, u_address, u_password);
    user_added := 1;

    RETURN user_added;
EXCEPTION
    WHEN unique_violation THEN
        user_added := 0;

        RETURN user_added;
END;
$$;
```

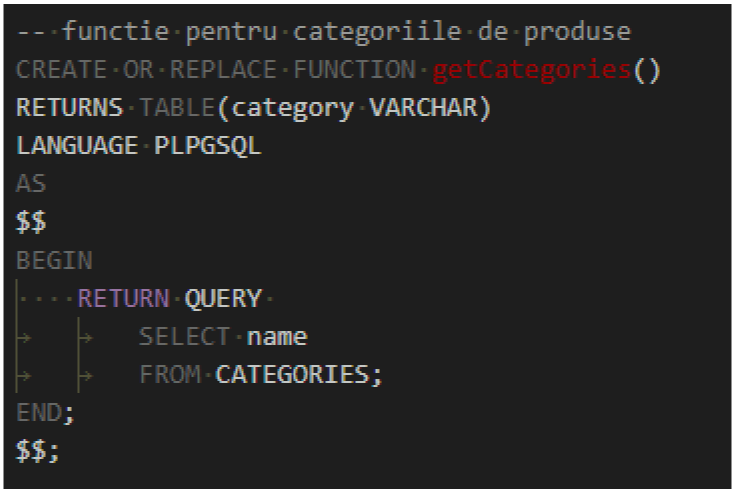


```sql
-- functie pentru verificare credentiale la logare
CREATE OR REPLACE FUNCTION retrievePassword(u_email VARCHAR)
RETURNS VARCHAR
LANGUAGE PLPGSQL
AS
$$
DECLARE
    user_cursor CURSOR(c_email VARCHAR) FOR
        SELECT email, password
        FROM USERS
        WHERE email = c_email;

    user_record RECORD;
    password VARCHAR := '';
BEGIN
    OPEN user_cursor(u_email);

    LOOP
        FETCH user_cursor INTO user_record;
        EXIT WHEN NOT FOUND;

        password := user_record.password;
    END LOOP;

    CLOSE user_cursor;

    RETURN password;
END;
$$;
```

DUMITRU Irina-Alexandra
341C3

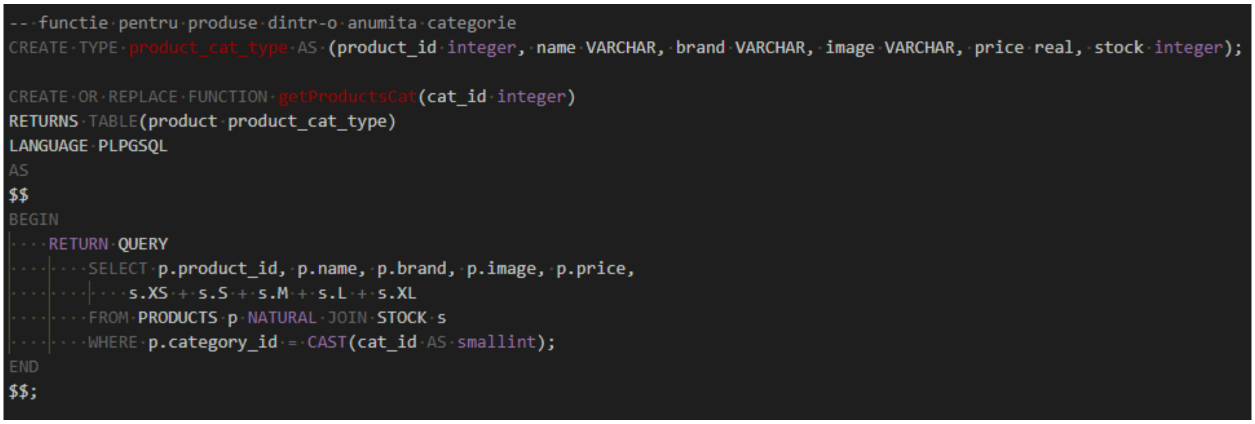


```
-- functie pentru extragerea numelui utilizatorului
-- apelata numai daca verificarea credentialelor reuseste
CREATE TYPE user_type AS (user_id integer, u_fullName VARCHAR);

CREATE OR REPLACE FUNCTION retrieveName(u_email VARCHAR)
RETURNS user_type
LANGUAGE PLPGSQL
AS
$$
DECLARE
    user_result user_type;
BEGIN
    SELECT user_id, firstName || ' ' || lastName AS fullname
    INTO user_result.user_id, user_result.u_fullname
    FROM USERS
    WHERE email = u_email;

    RETURN user_result;
END;
$$;
```

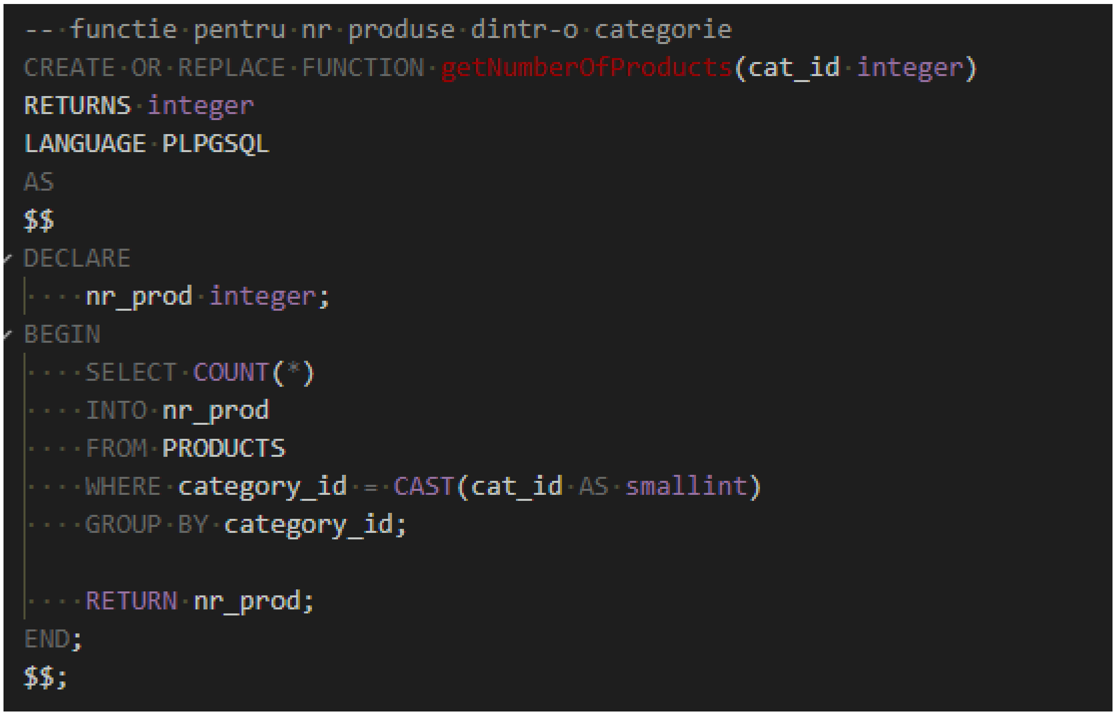


```
-- functie pentru categoriile de produse
CREATE OR REPLACE FUNCTION getCategories()
RETURNS TABLE(category VARCHAR)
LANGUAGE PLPGSQL
AS
$$
BEGIN
    RETURN QUERY 
        SELECT name
        FROM CATEGORIES;
END;
$$;
```

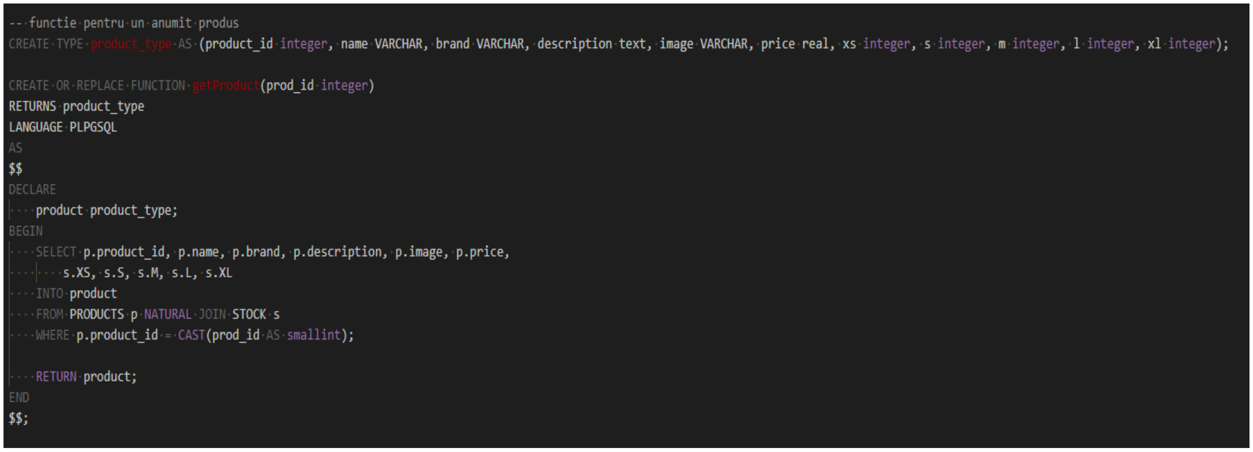
DUMITRU Irina-Alexandra
341C3



```sql
-- functie pentru produse dintr-o anumita categorie
CREATE TYPE product_cat_type AS (product_id integer, name VARCHAR, brand VARCHAR, image VARCHAR, price real, stock integer);

CREATE OR REPLACE FUNCTION getProductsCat(cat_id integer)
RETURNS TABLE(product product_cat_type)
LANGUAGE PLPGSQL
AS
$$
BEGIN
    RETURN QUERY
        SELECT p.product_id, p.name, p.brand, p.image, p.price,
            s.XS + s.S + s.M + s.L + s.XL
        FROM PRODUCTS p NATURAL JOIN STOCK s
        WHERE p.category_id = CAST(cat_id AS smallint);
END
$$;
```

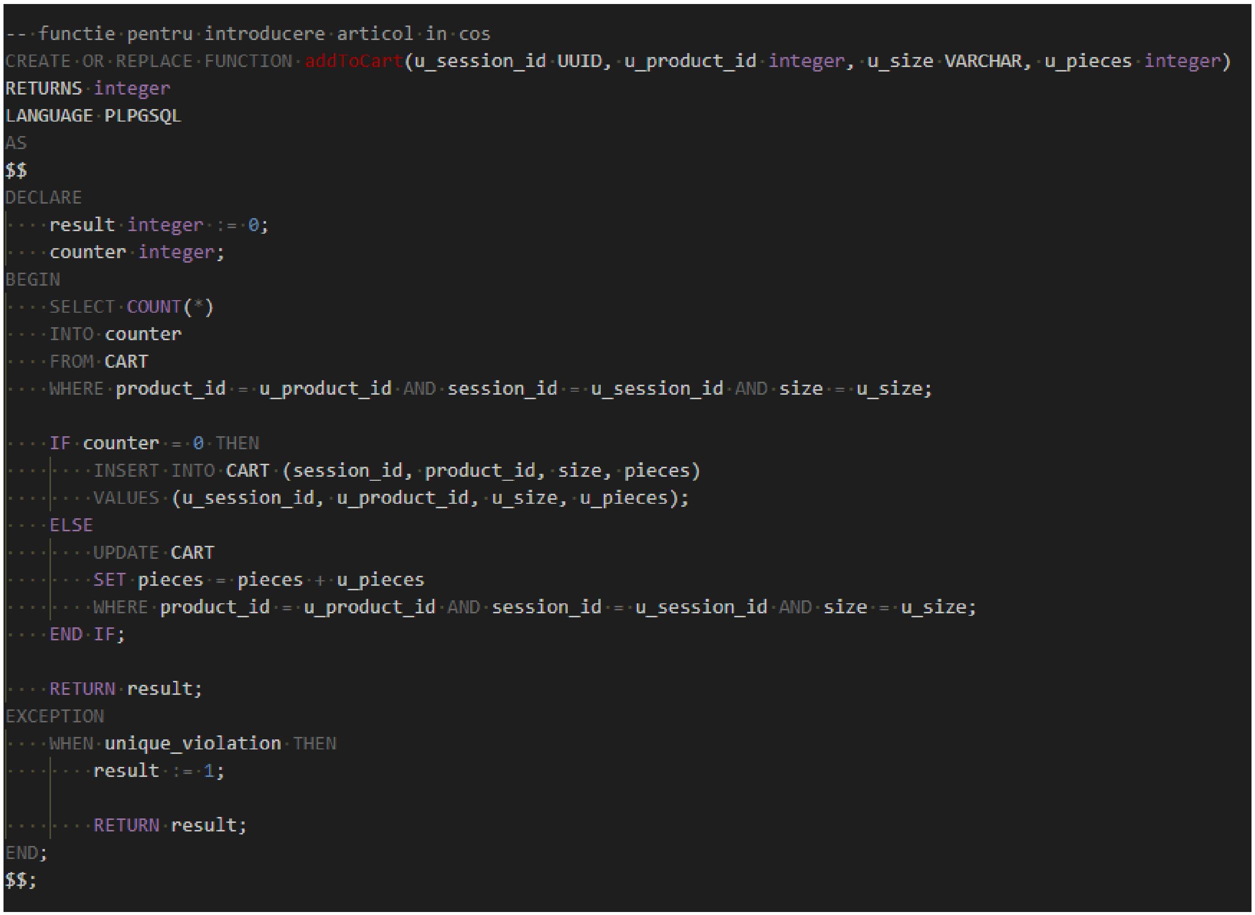


```sql
-- functie pentru nr produse dintr-o categorie
CREATE OR REPLACE FUNCTION getNumberOfProducts(cat_id integer)
RETURNS integer
LANGUAGE PLPGSQL
AS
$$
DECLARE
    nr_prod integer;
BEGIN
    SELECT COUNT(*)
    INTO nr_prod
    FROM PRODUCTS
    WHERE category_id = CAST(cat_id AS smallint)
    GROUP BY category_id;

    RETURN nr_prod;
END;
$$;
```

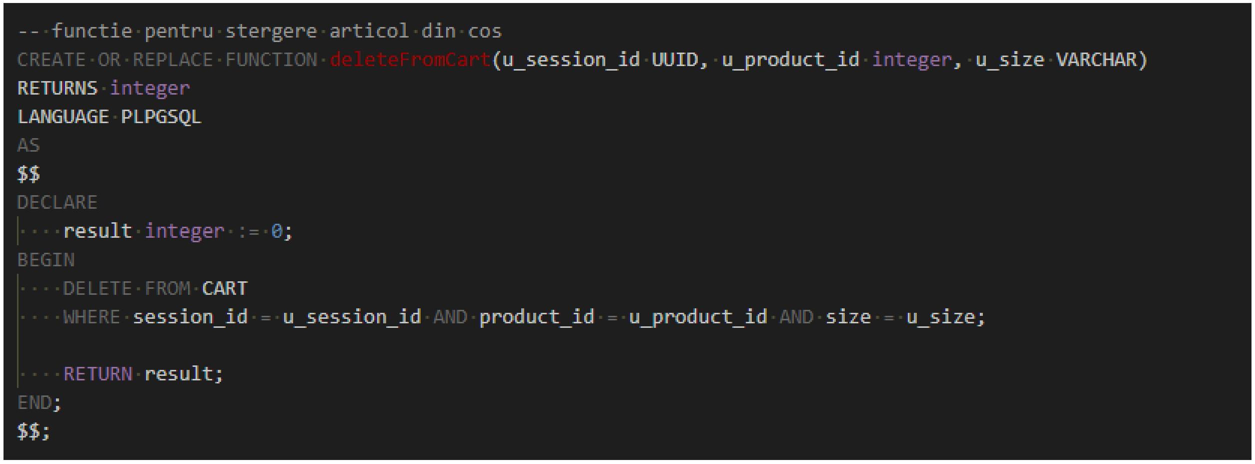


```sql
-- functie pentru un anumit produs
CREATE TYPE product_type AS (product_id integer, name VARCHAR, brand VARCHAR, description text, image VARCHAR, price real, xs integer, s integer, m integer, l integer, xl integer);

CREATE OR REPLACE FUNCTION getProduct(prod_id integer)
RETURNS product_type
LANGUAGE PLPGSQL
AS
$$
DECLARE
    product product_type;
BEGIN
    SELECT p.product_id, p.name, p.brand, p.description, p.image, p.price,
        s.XS, s.S, s.M, s.L, s.XL
    INTO product
    FROM PRODUCTS p NATURAL JOIN STOCK s
    WHERE p.product_id = CAST(prod_id AS smallint);

    RETURN product;
END
$$;
```

DUMITRU Irina-Alexandra
341C3

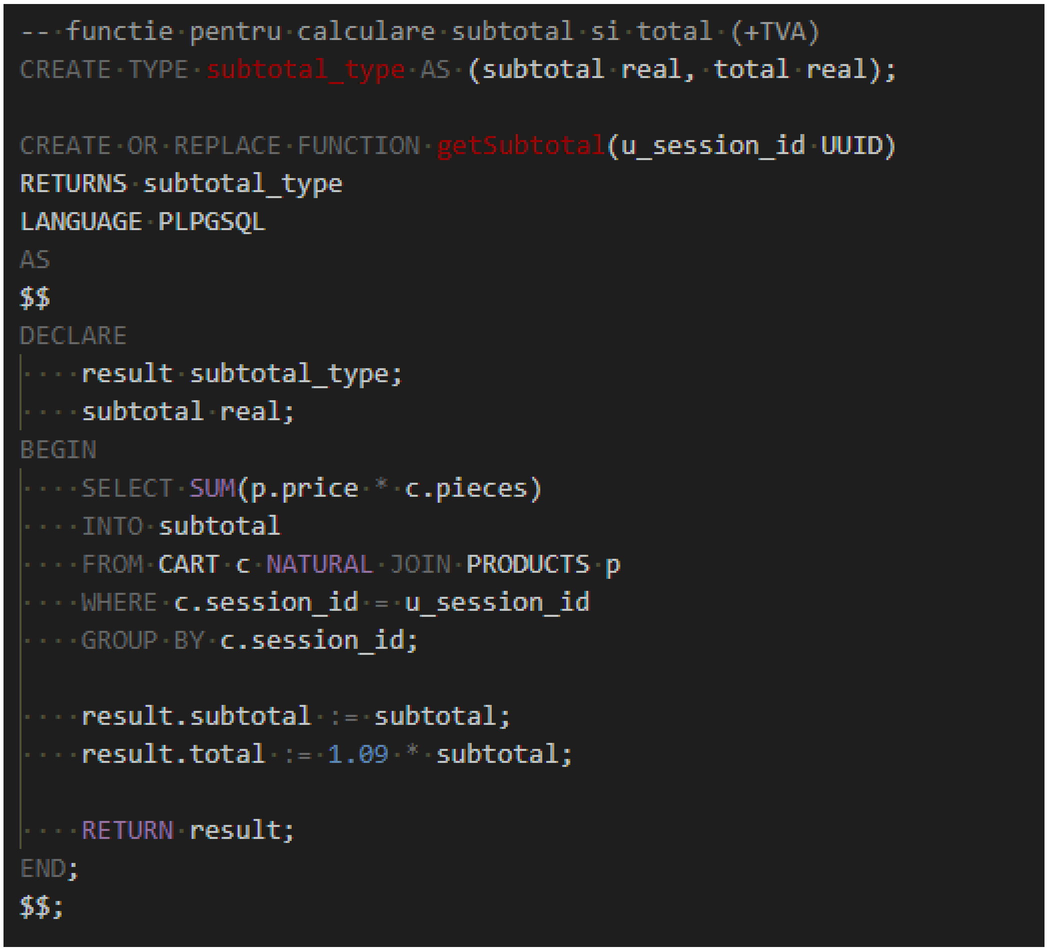


```
-- functie pentru introducere articol in cos
CREATE OR REPLACE FUNCTION addToCart(u_session_id UUID, u_product_id integer, u_size VARCHAR, u_pieces integer)
RETURNS integer
LANGUAGE PLPGSQL
AS
$$
DECLARE
    result integer := 0;
    counter integer;
BEGIN
    SELECT COUNT(*)
    INTO counter
    FROM CART
    WHERE product_id = u_product_id AND session_id = u_session_id AND size = u_size;

    IF counter = 0 THEN
        INSERT INTO CART (session_id, product_id, size, pieces)
        VALUES (u_session_id, u_product_id, u_size, u_pieces);
    ELSE
        UPDATE CART
        SET pieces = pieces + u_pieces
        WHERE product_id = u_product_id AND session_id = u_session_id AND size = u_size;
    END IF;

    RETURN result;
EXCEPTION
    WHEN unique_violation THEN
        result := 1;

        RETURN result;
END;
$$;
```



```
-- functie pentru stergere articol din cos
CREATE OR REPLACE FUNCTION deleteFromCart(u_session_id UUID, u_product_id integer, u_size VARCHAR)
RETURNS integer
LANGUAGE PLPGSQL
AS
$$
DECLARE
    result integer := 0;
BEGIN
    DELETE FROM CART
    WHERE session_id = u_session_id AND product_id = u_product_id AND size = u_size;

    RETURN result;
END;
$$;
```

DUMITRU Irina-Alexandra
341C3

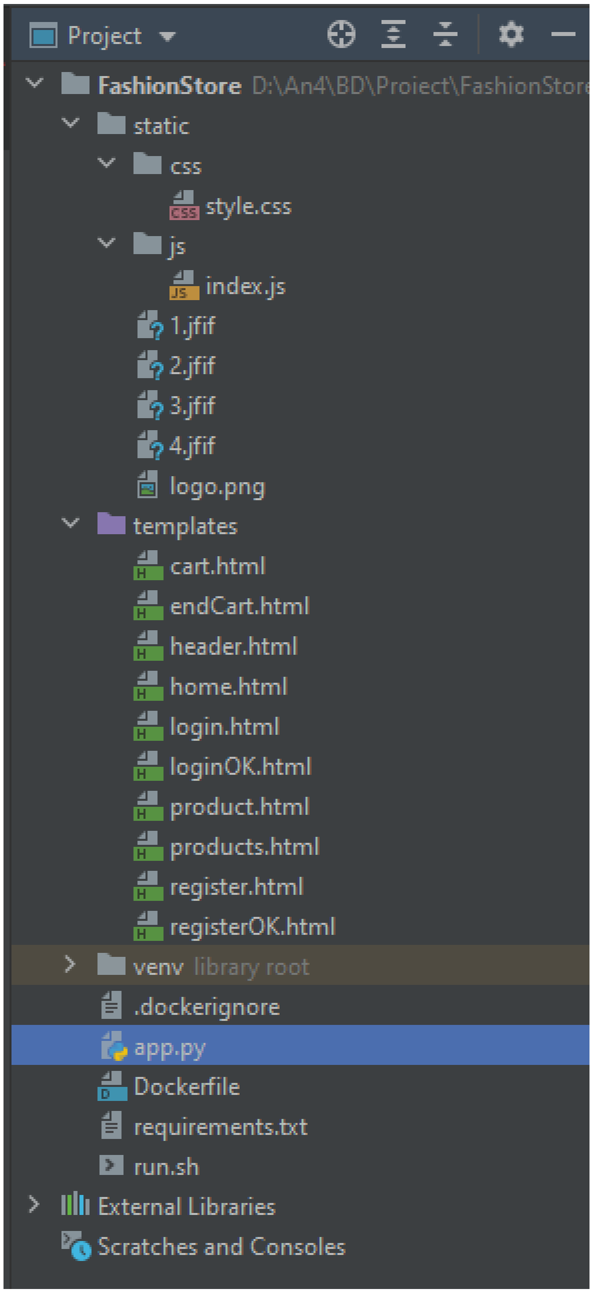


```
-- functie pentru calculare subtotal si total (+TVA)
CREATE TYPE subtotal_type AS (subtotal real, total real);

CREATE OR REPLACE FUNCTION getSubtotal(u_session_id UUID)
RETURNS subtotal_type
LANGUAGE PLPGSQL
AS
$$
DECLARE
    result subtotal_type;
    subtotal real;
BEGIN
    SELECT SUM(p.price * c.pieces)
    INTO subtotal
    FROM CART c NATURAL JOIN PRODUCTS p
    WHERE c.session_id = u_session_id
    GROUP BY c.session_id;

    result.subtotal := subtotal;
    result.total := 1.09 * subtotal;

    RETURN result;
END;
$$;
```

DUMITRU Irina-Alexandra
341C3

Descrierea aplicatiei

Diagrama de clase



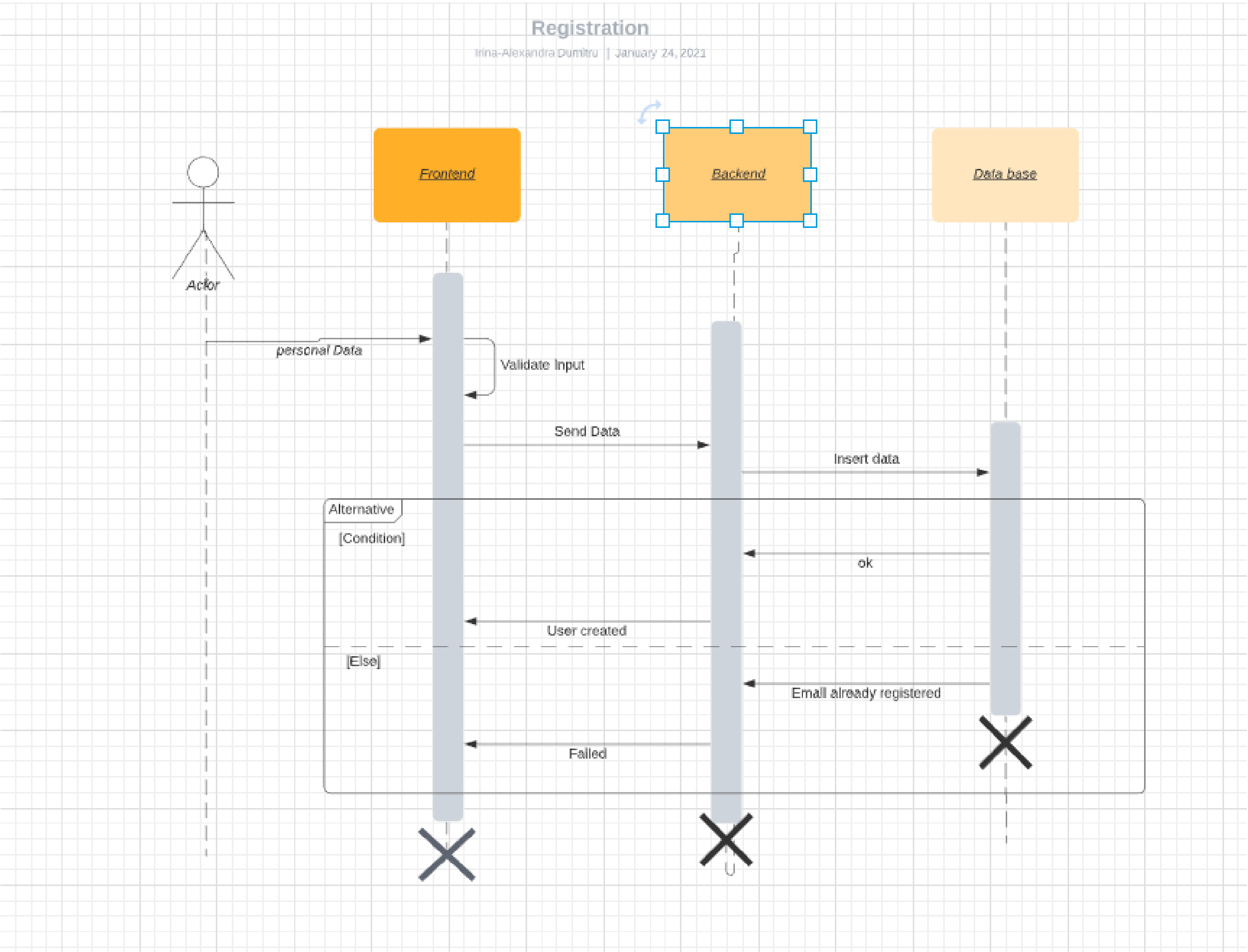
```
Project
FashionStore D:\An4\BD\Proiect\FashionStore
    static
        css
            style.css
        js
            index.js
        1.jfif
        2.jfif
        3.jfif
        4.jfif
        logo.png
    templates
        cart.html
        endCart.html
        header.html
        home.html
        login.html
        loginOK.html
        product.html
        products.html
        register.html
        registerOK.html
    venv library root
    .dockerignore
    app.py
    Dockerfile
    requirements.txt
    run.sh
External Libraries
Scratches and Consoles
```

DUMITRU Irina-Alexandra
341C3

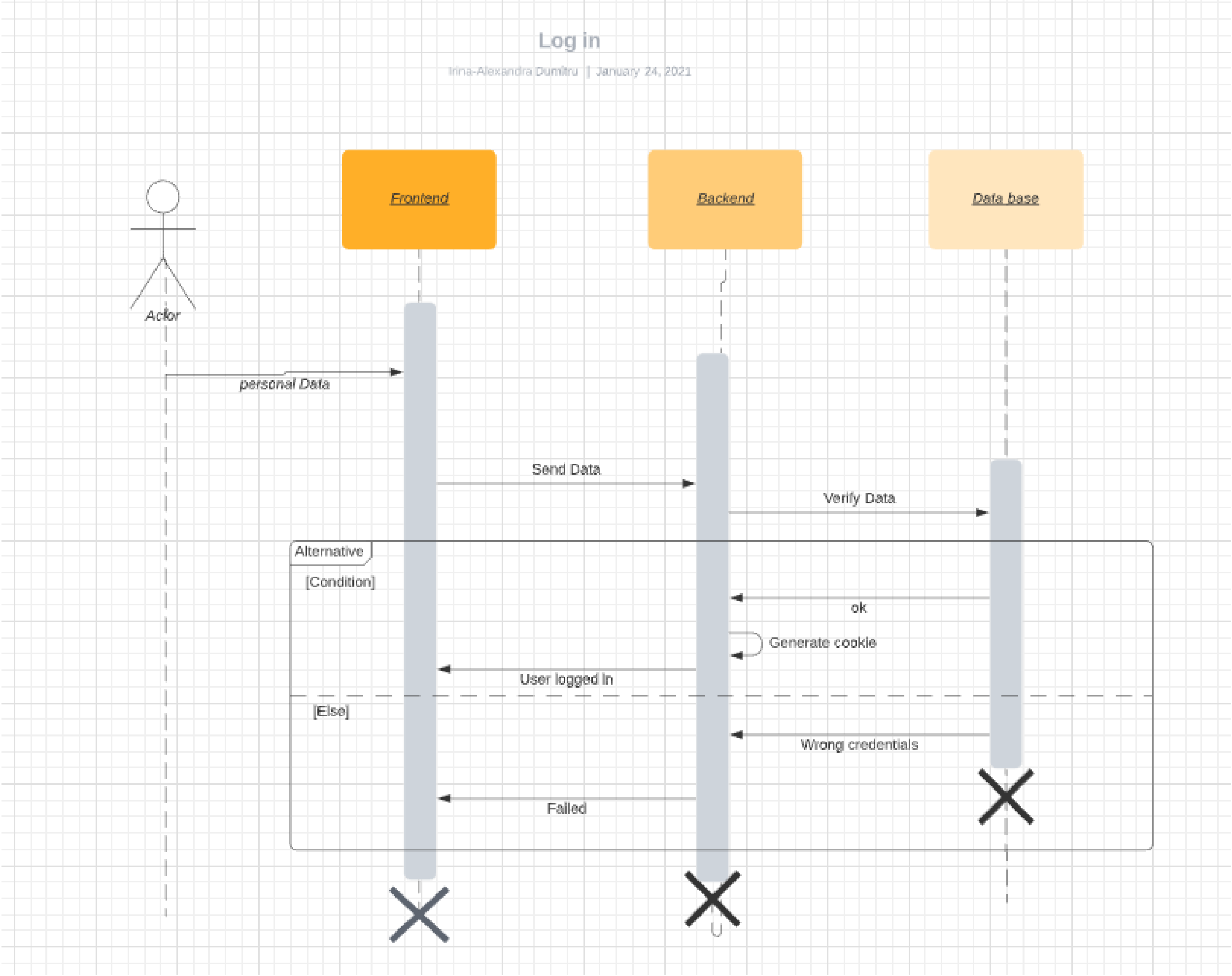
Structura claselor

In app.py se afla server-ul, care se ocupa si de randarea paginilor HTML. In css, se afla stilul obiectelor, iar in templates, template-urile pentru diversele interfete.

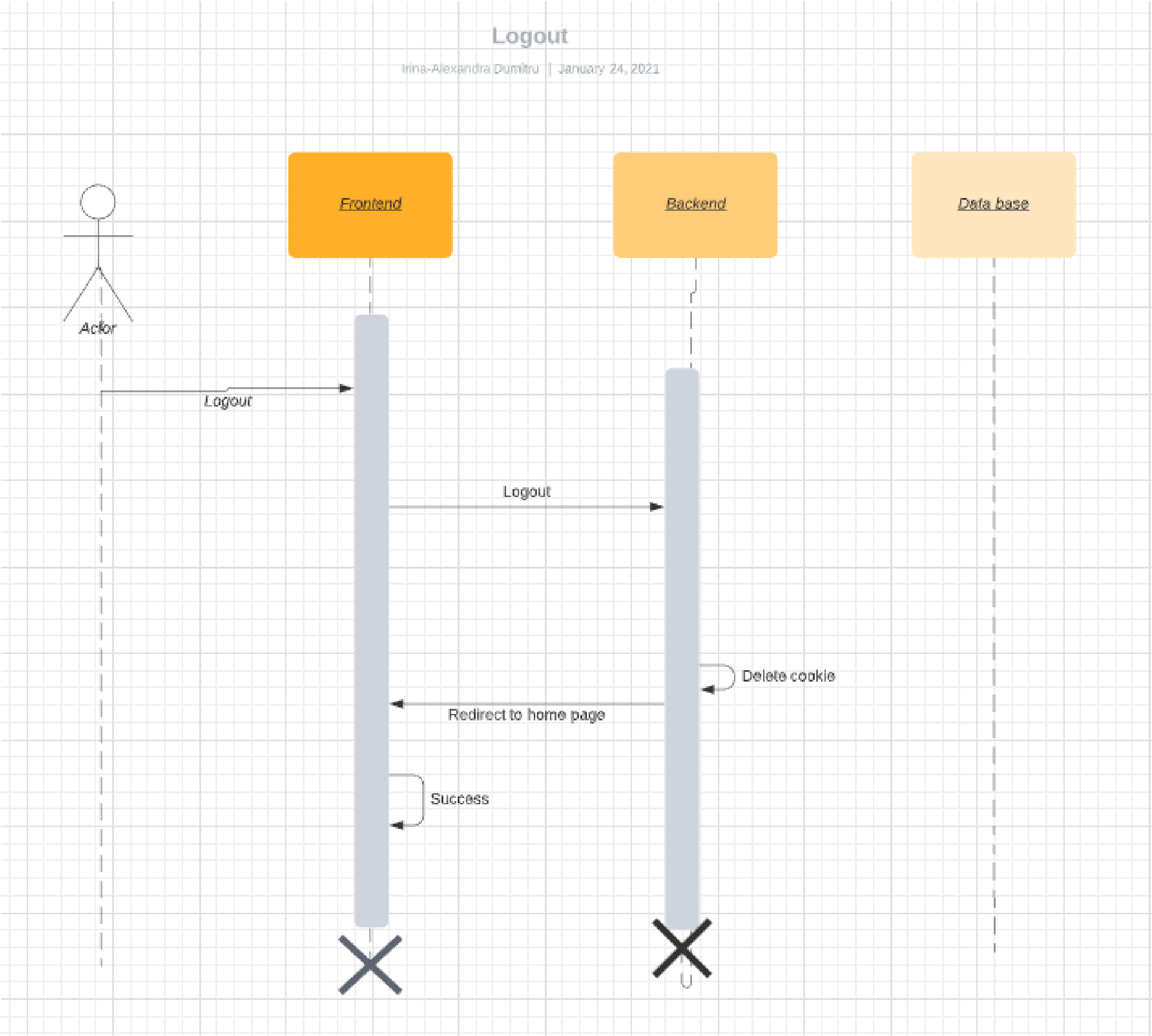
Diagrama de stari



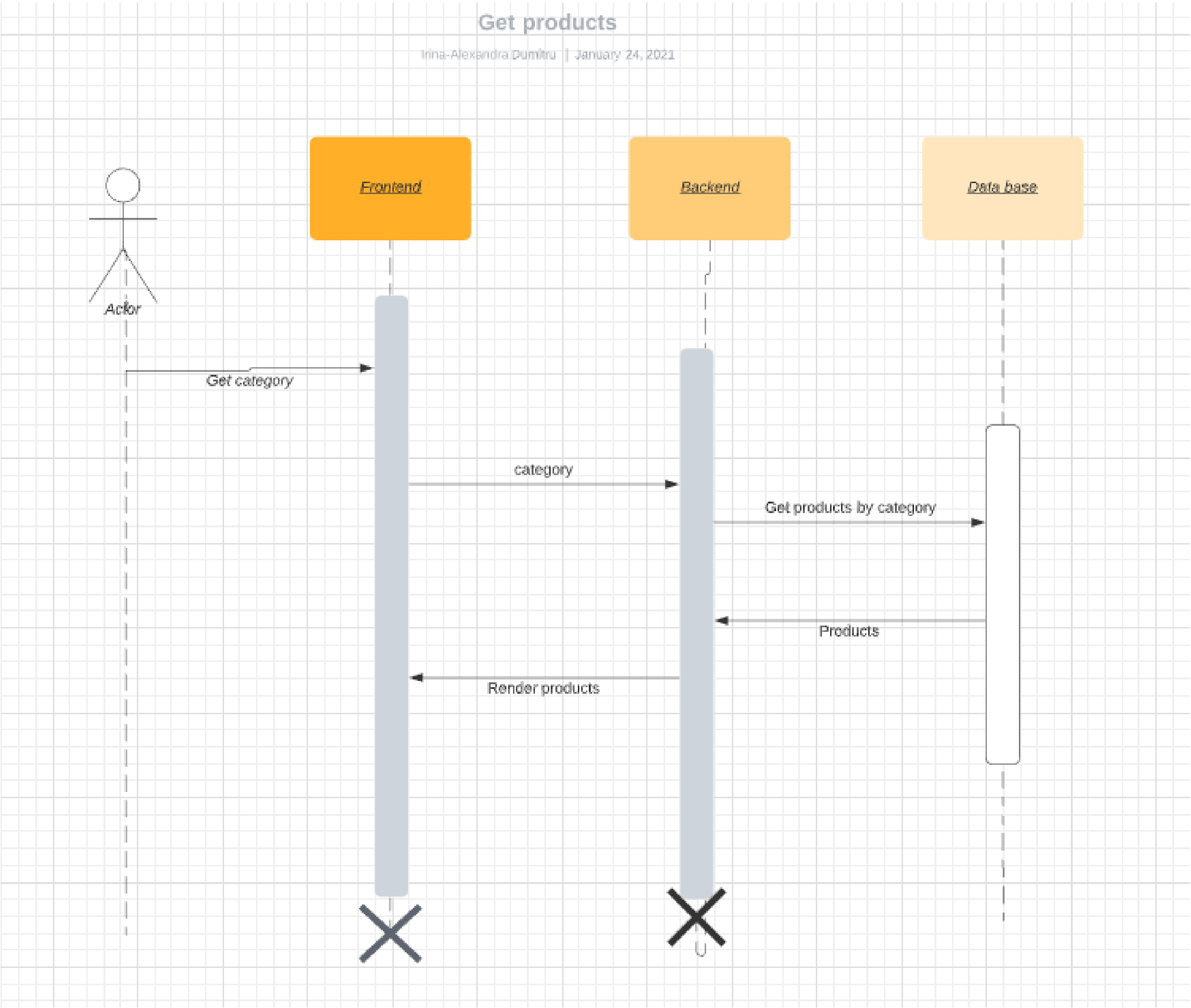
DUMITRU Irina-Alexandra
341C3



DUMITRU Irina-Alexandra
341C3



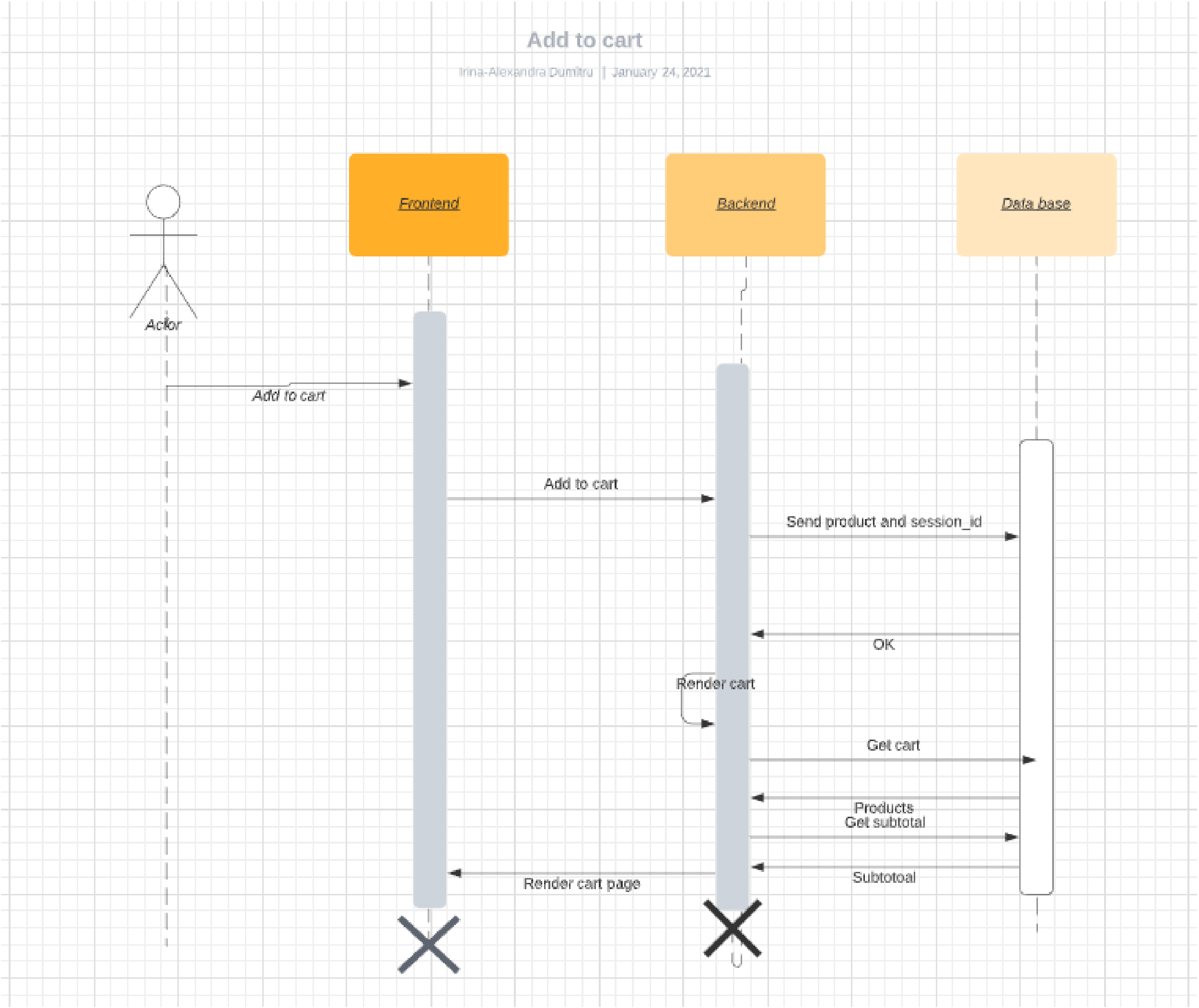
DUMITRU Irina-Alexandra
341C3



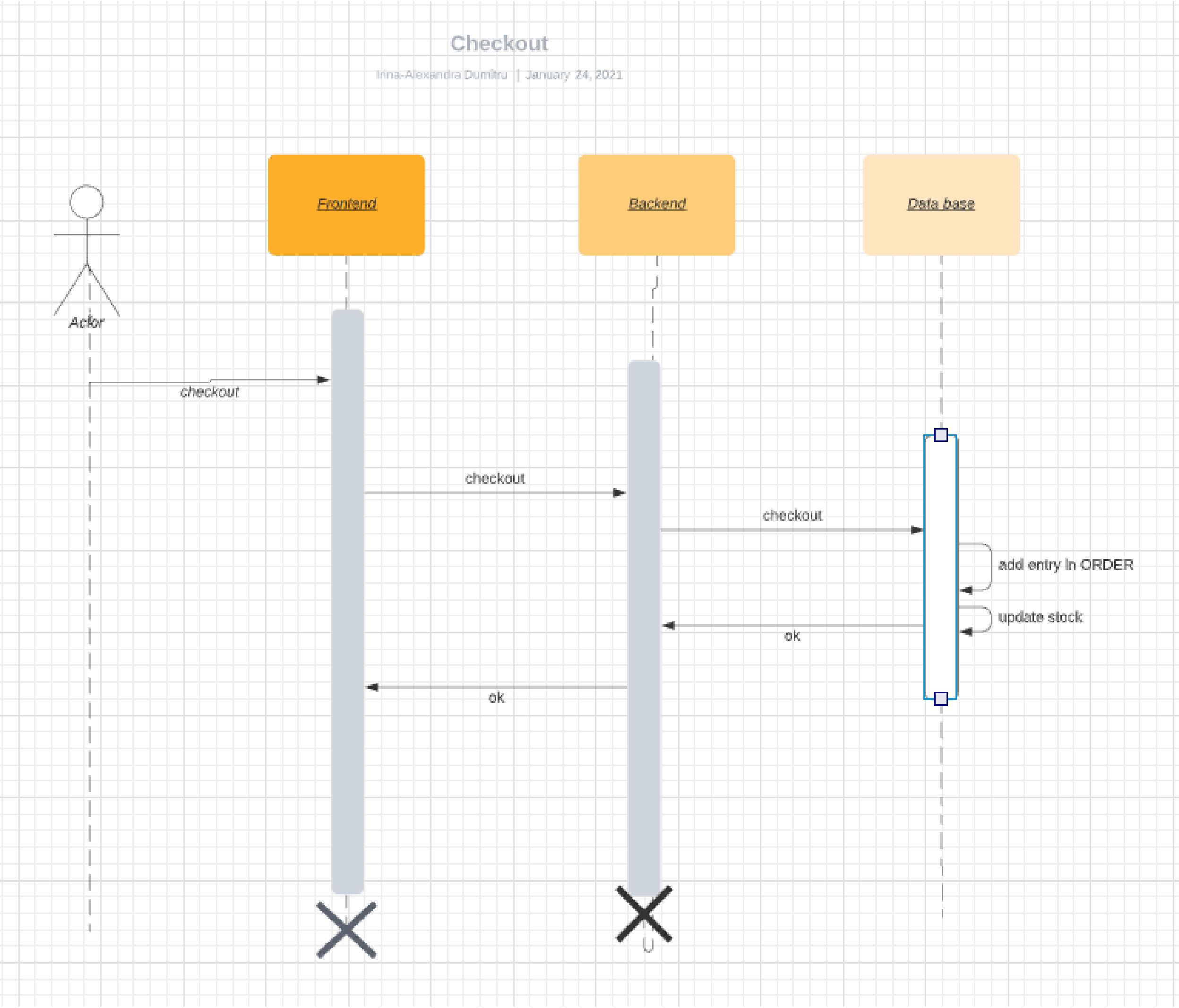
Get products

Irina-Alexandra Dumitru | January 24, 2021

DUMITRU Irina-Alexandra
341C3

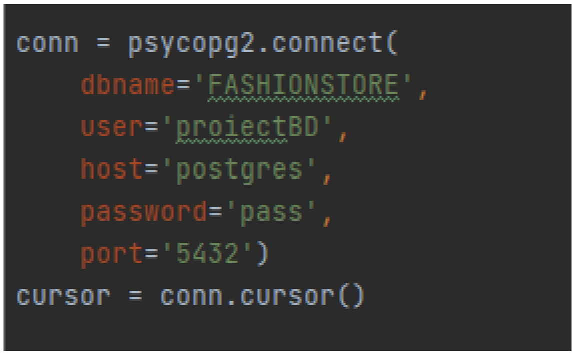


DUMITRU Irina-Alexandra
341C3

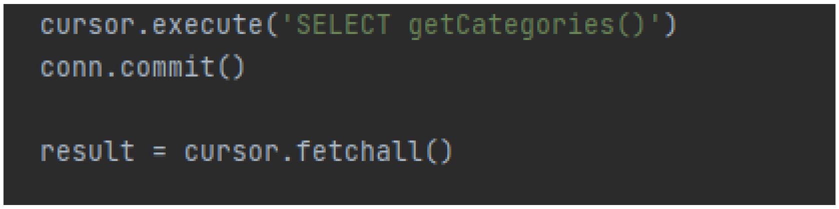


DUMITRU Irina-Alexandra
341C3

Modul in care se face conexiunea la BD



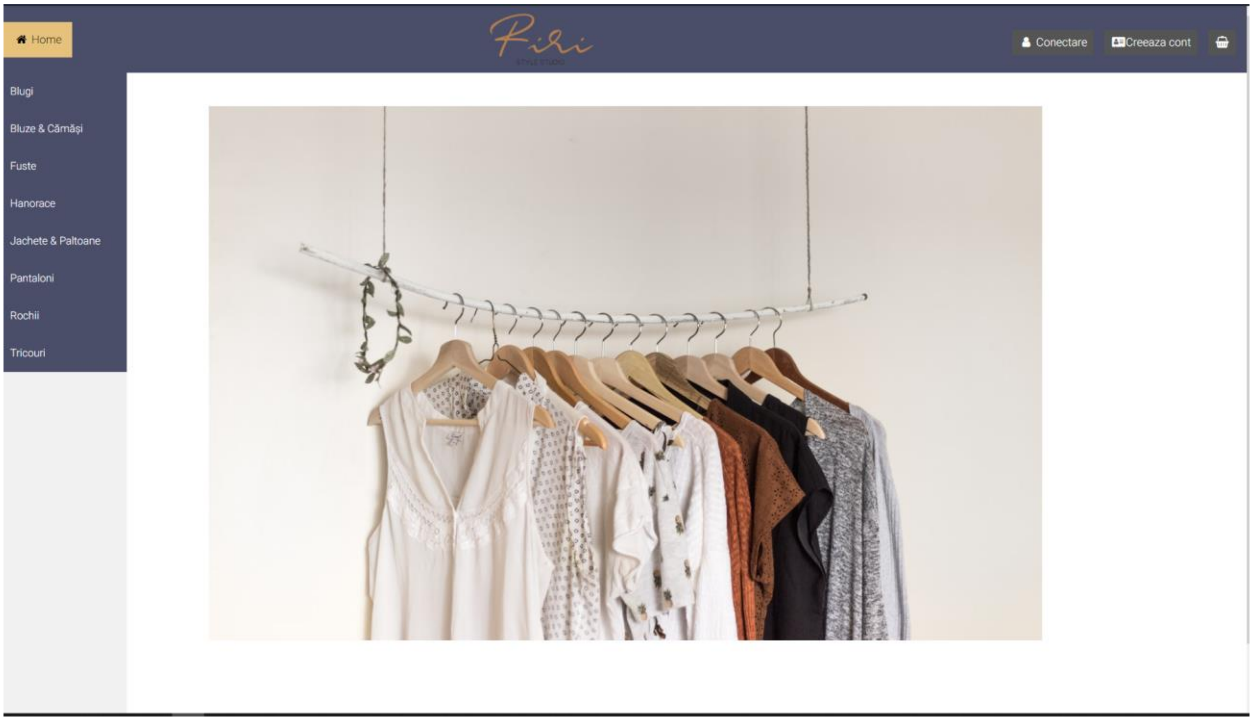
```
conn = psycopg2.connect(
    dbname='FASHIONSTORE',
    user='proiectBD',
    host='postgres',
    password='pass',
    port='5432')
cursor = conn.cursor()
```

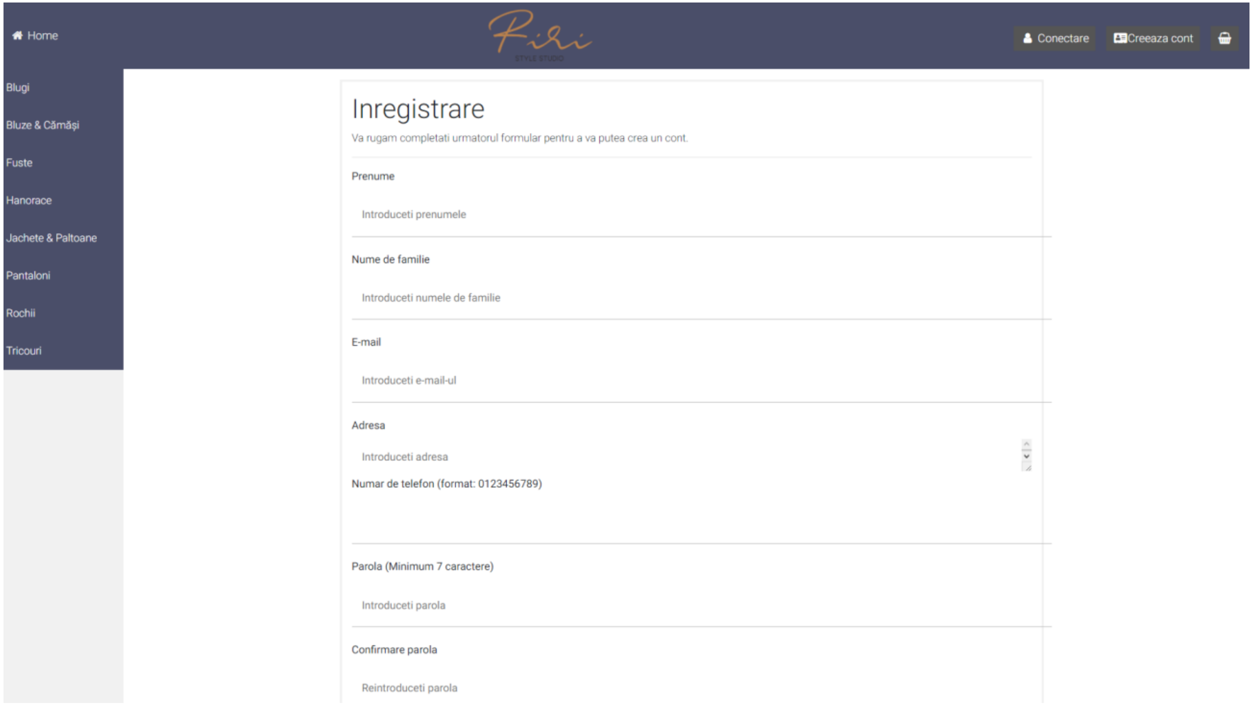


```
cursor.execute('SELECT getCategories()')
conn.commit()

result = cursor.fetchall()
```

DUMITRU Irina-Alexandra
341C3

Interfete





Inregistrare

Va rugam completati urmatorul formular pentru a va putea crea un cont.

Prenume

Introduceti prenumele

Nume de familie

Introduceti numele de familie

E-mail

Introduceti e-mail-ul

Adresa

Introduceti adresa

Numar de telefon (format: 0123456789)

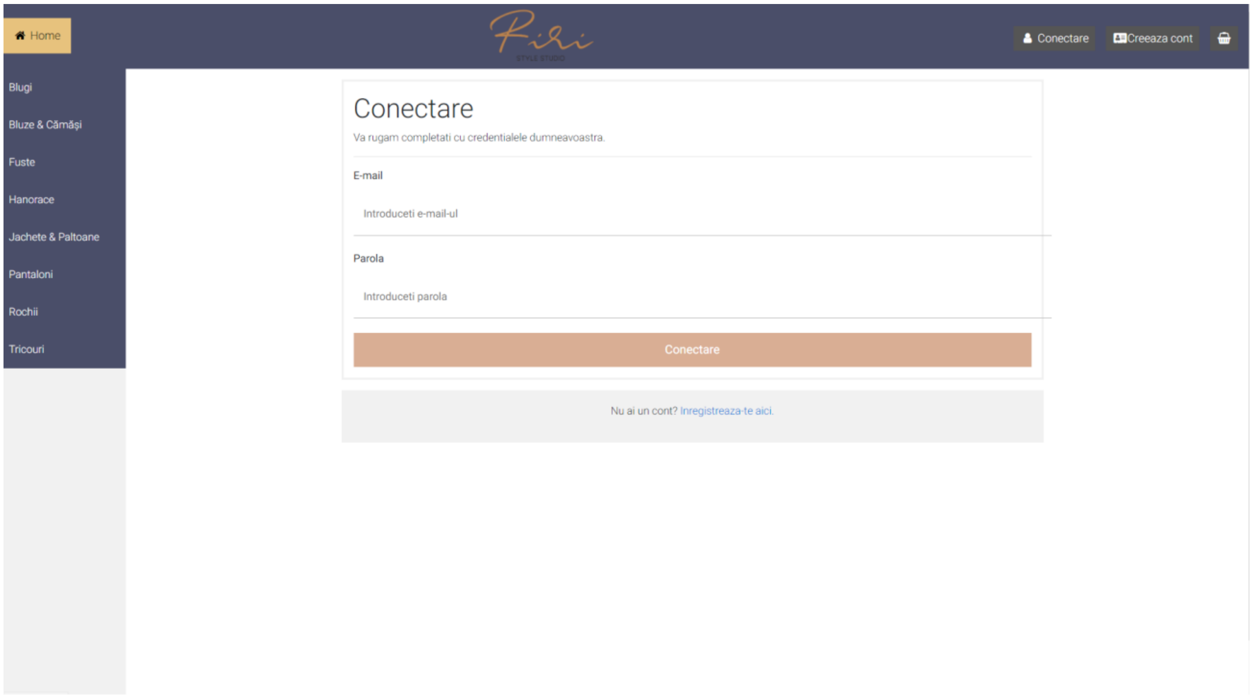
Parola (Minimum 7 caractere)

Introduceti parola

Confirmare parola

Reintroduceti parola

DUMITRU Irina-Alexandra
341C3



Conectare

Va rugam completati cu credentialele dumneavoastra.

E-mail

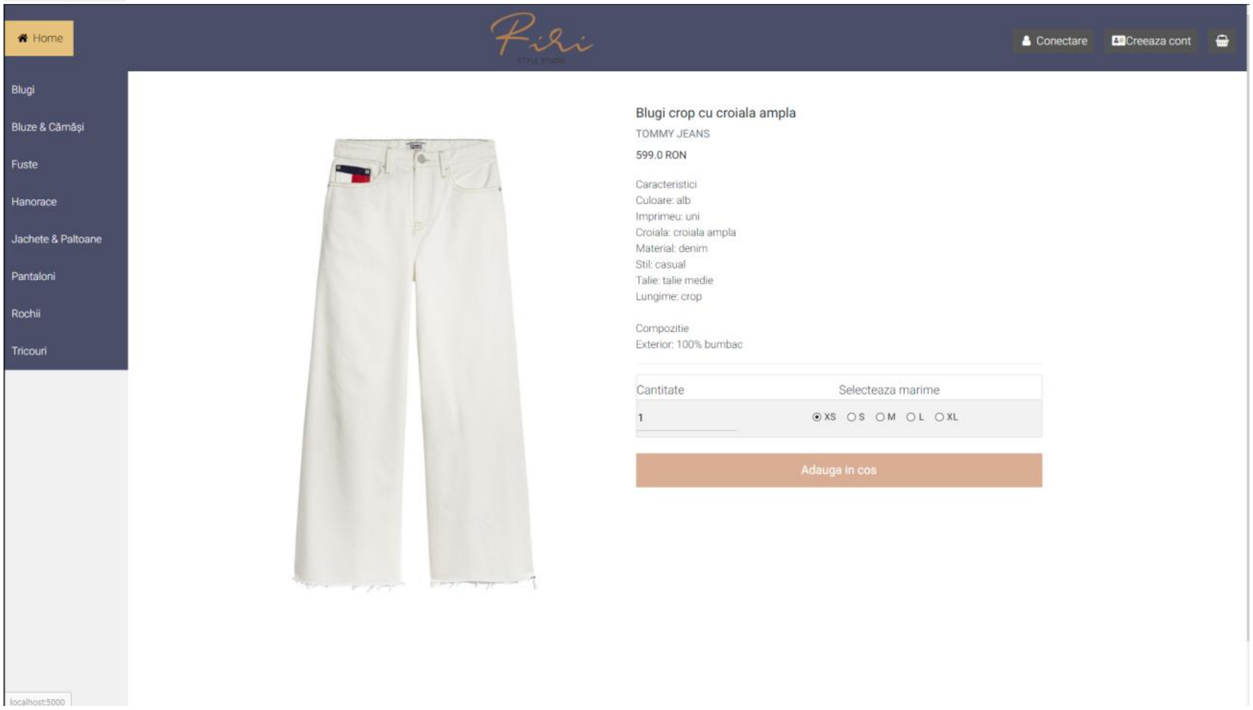
Introduceti e-mail-ul

Parola

Introduceti parola

Conectare

Nu ai un cont? Inregistreaza-te aici.



Blugi crop cu croiala ampla

TOMMY JEANS

599.0 RON

Caracteristici
Culoare: alb
Imprimeu: uni
Croiala: croiala ampla
Material: denim
Stil: casual
Talie: talie medie
Lungime: crop

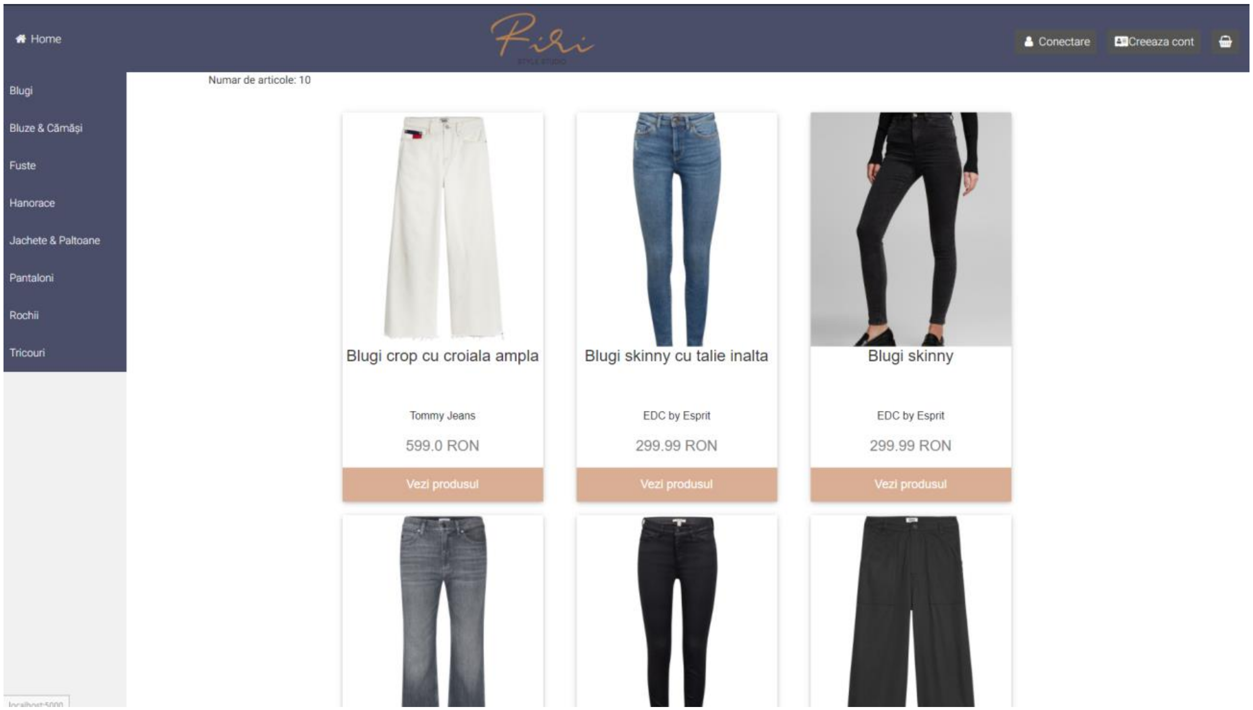
Compozitie
Exterior: 100% bumbac

Cantitate: 1

Selecteaza marime: ● XS ○ S ○ M ○ L ○ XL

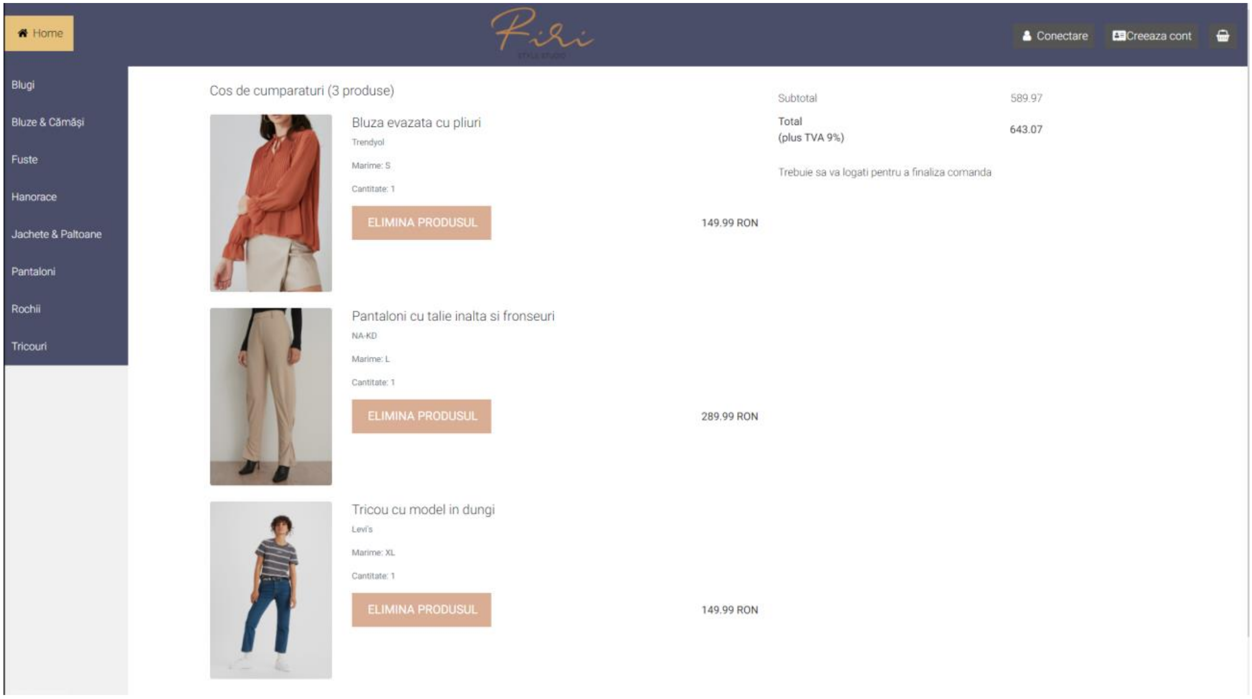
Adauga in cos

DUMITRU Irina-Alexandra
341C3



Numar de articole: 10

Blugi crop cu croiala ampla	Blugi skinny cu talie inalta	Blugi skinny
Tommy Jeans	EDC by Esprit	EDC by Esprit
599.0 RON	299.99 RON	299.99 RON



Cos de cumparaturi (3 produse)

Bluza evazata cu pliuri
Trendyol
Marime: S
Cantitate: 1
149.99 RON

Pantaloni cu talie inalta si fronseuri
NA-KD
Marime: L
Cantitate: 1
289.99 RON

Tricou cu model in dungi
Levi's
Marime: XL
Cantitate: 1
149.99 RON

Subtotal	589.97
Total (plus TVA 9%)	643.07

Trebuie sa va logati pentru a finaliza comanda

Concluzii

A fost un proiect solicitant, dar foarte satisfacator, mai ales ca am putut vedea „frumusetea” lui. Cu siguranta mi-a starnit interesul pentru web development, iar partea de backend a fost mult mai usoara decat cea de frontend.