**Proiect afaceri electronice**

**Marinescu Ruxandra**

https://github.com/rxndrmrnsc/ProiectAE

LOGIN:

A screenshot of a computer

Description automatically generated

Afisare produse + filtrare pe baza pretului:

A screenshot of a computer

Description automatically generated

Filtrare pe baza pretului:

A screenshot of a computer

Description automatically generated

**Structura baze de date:**

const Order = sequelize.define('Order', {

    id: {

        type: DataTypes.INTEGER,

        primaryKey: true,

        autoIncrement: true

    },

    total: {

        type: DataTypes.FLOAT,

    },

    createdAt: {

        type: DataTypes.DATE,

        defaultValue: DataTypes.NOW

    },

    updatedAt: {

        type: DataTypes.DATE,

        defaultValue: DataTypes.NOW

    },

    address: {

        type: DataTypes.STRING,

        validate: {

            len: [3, 100]

        }

    },

    city: {

        type: DataTypes.STRING,

        validate: {

            len: [3, 50]

        }

    },

    name: {

        type: DataTypes.STRING,

        validate: {

            len: [3, 200]

        }

    },

    phone: {

        type: DataTypes.STRING,

    },

    status: {

        type: DataTypes.ENUM,

        allowNull: false,

        values: ['placed', 'sent'],

        defaultValue: 'placed'

    },

    userId: {

        type: DataTypes.INTEGER

    },

})

const Product = sequelize.define('Product', {

    id: {

        type: DataTypes.INTEGER,

        primaryKey: true,

        autoIncrement: true

    },

    price: {

        type: DataTypes.FLOAT,

    },

    name: {

        type: DataTypes.STRING,

    },

    description: {

        type: DataTypes.STRING,

    },

    status: {

        type: DataTypes.ENUM,

        allowNull: false,

        values: ['in stock', 'out of stock'],

        defaultValue: 'in stock'

    },

})

const OrderProduct = sequelize.define('OrderProduct', {

})

const User = sequelize.define('User', {

    id: {

        type: DataTypes.INTEGER,

        primaryKey: true,

        autoIncrement: true

    },

    email: {

        type: DataTypes.STRING,

        unique: true

    },

    password: DataTypes.STRING,

    role: {

        type: DataTypes.ENUM,

        values: ['user', 'tester', 'admin'],

        validate: {

            isIn: [['user', 'tester', 'admin']]

        }

    }

})

**Operatii CRUD pt orders:**

curl --location --request GET 'localhost:8080/orders/' \

--header 'Content-Type: application/json' \

--data '{

"userId": 1

}'

curl --location 'localhost:8080/orders/' \

--header 'Content-Type: application/json' \

--data '{

"userId": 1,

"address": "O adresa adevarata",

"city": "Lazytown",

"name": "Filarmoniu",

"phone": "0732882411"

}'

curl --location --request DELETE 'localhost:8080/orders/2' \

--header 'Content-Type: application/json' \

--data '{

"userId": 1

}'

curl --location --request PUT 'localhost:3000/orders/1' \

--header 'Content-Type: application/json' \

--data '{

"userId": 1,

"name": "Angelina",

"phone": "2315"

}'

**Operatii CRUD pentru Products:**

curl --location --request GET 'localhost:3000/products/' \

--header 'Content-Type: application/json' \

--data '{

"userId": 2

}'

curl --location 'localhost:3000/products/' \

--header 'Content-Type: application/json' \

--data '{

"userId": 1,

"price": 50,

"name": "Product 1",

"description": "Description of product 2"

}'

curl --location --request PUT 'localhost:3000/products/1' \

--header 'Content-Type: application/json' \

--data '{

"userId": 1,

"price": 30,

"name": "Product 1",

"description": "Description of product 1"

}'

curl --location --request DELETE 'localhost:8080/products/1' \

--header 'Content-Type: application/json' \

--data '{

"userId": 2

}'

**Operatii CRUD price range:**

curl --location --request GET 'localhost:3000/products/priceRange' \

--header 'Content-Type: application/json' \

--data '{

"userId": 2

}'

curl --location --request GET 'localhost:3000/products/filter?priceMin=200&priceMax=300' \

--header 'Content-Type: application/json' \

--data '{

"userId": 2,

"priceMin": 20,

"priceMax": 200

}'

**Operatii CRUD auth si user:**

curl --location 'localhost:8080/auth/login' \

--header 'Content-Type: application/json' \

--data-raw '{

"email": "rux@mail.com",

"password": "parola"

}'

curl --location 'localhost:3000/users/' \

--header 'Content-Type: application/json' \

--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNzM0MzY5NjQ3LCJleHAiOjE3MzQzNzMyNDd9.BhbYCjxRs5MFmQeS9t8oUpD95d267Y5-yVFeJh4QIac' \

--data-raw '{

"email": "admin@mail.com",

"password": "admin",

"role": "admin"

}'

curl --location 'localhost:3000/users/' \

--header 'Content-Type: application/json' \

--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNzM0MzY5NjQ3LCJleHAiOjE3MzQzNzMyNDd9.BhbYCjxRs5MFmQeS9t8oUpD95d267Y5-yVFeJh4QIac' \

--data-raw '{

"email": "admin@mail.com",

"password": "admin",

"role": "admin"

}'

**Operatii CRUD OrderProducts:**

curl --location --request GET 'localhost:8080/orders/1/products/' \

--header 'Content-Type: application/json' \

--data '{

"userId": 1

}'

curl --location 'localhost:8080/orders/1/products/1' \

--header 'Content-Type: application/json' \

--data '{

"userId": 1

}'

curl --location --request PUT 'localhost:3000/products/1' \

--header 'Content-Type: application/json' \

--data '{

"userId": 1,

"price": 30,

"name": "Product 1",

"description": "Description of product 1"

}'

curl --location --request PUT 'localhost:3000/products/1' \

--header 'Content-Type: application/json' \

--data '{

"userId": 1,

"price": 30,

"name": "Product 1",

"description": "Description of product 1"

}'