

— prepared by —

- Layal Soud Halwani 2007896
- Afnan Tariq Algogandi 2007926
- Bedoor Ayad Alsulami 2005961

1

DAIRY
MILK / YOGHURT
SMALLGOODS
CEREAL
MUESLI BARS
BREAD

2

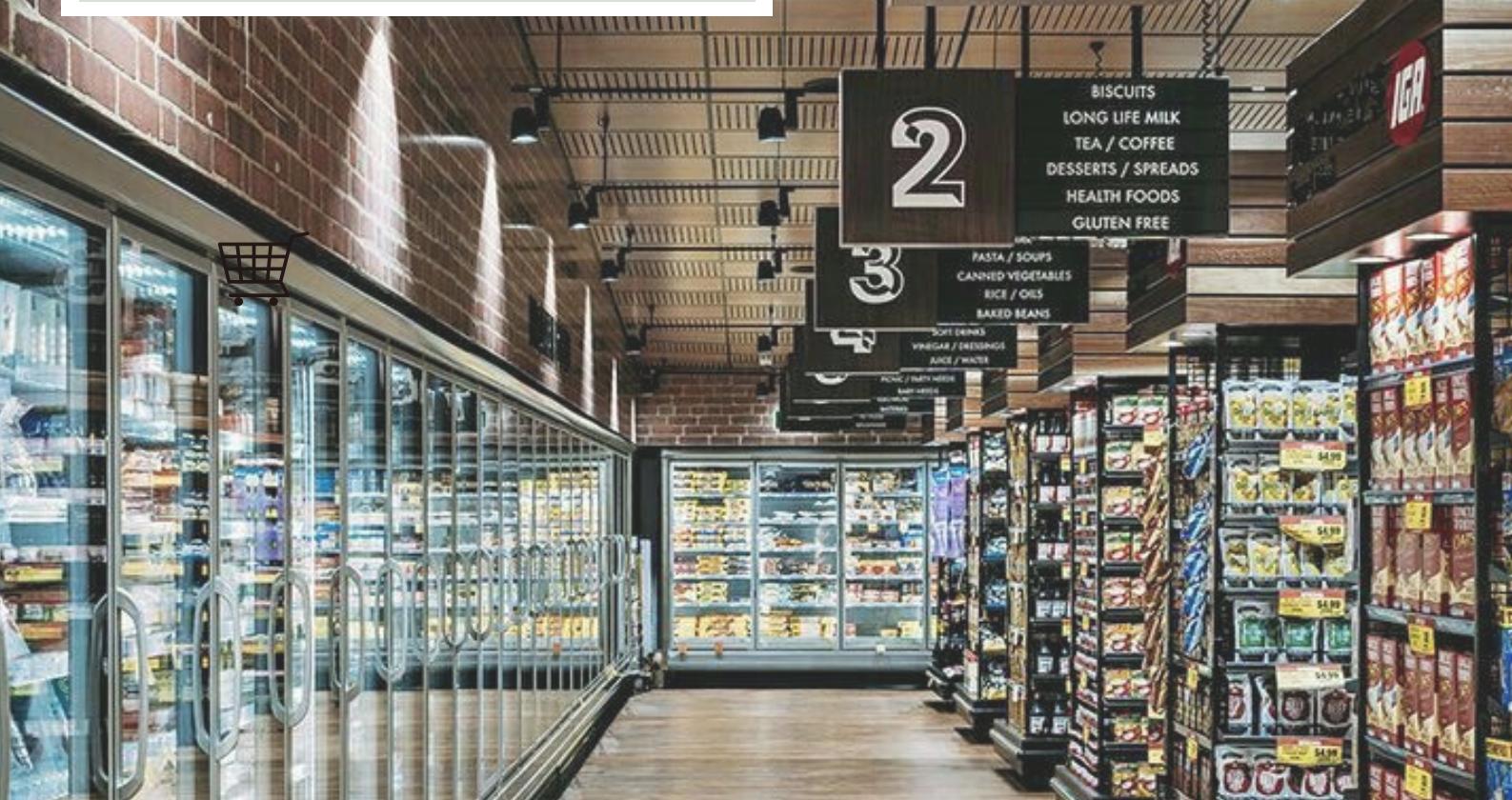
BISCUITS
LONG LIFE MILK
TEA / COFFEE
DESSERTS / SPREADS
HEALTH FOODS
GLUTEN FREE

3

PASTA / SOUPS
CANNED VEGETABLES
RICE / OATS
BAKED BEANS

SODA / DRINKS
VINEGAR / DRESSINGS
JUICE / WATER

PICKLE / PASTES
BABY NEEDS
WOMEN NEEDS



SELF-SERVICE SUPERMARKET

— Fundamentals of AI Project —

Section : A2L

prepared to: Dr.Bushra Alsaadi

TABLE OF

CONTENTS

Introduction	02
Our Solution	03
Programming language	04
AI approach	05
Steps	06
Flowchar	07
User manual	08
Code	13
Output	16
Code testing	18
Conclusion	21
References	22

INTRODUCTION

When you last visited the supermarket, how long did it take you to wait in line?, And how often do you want to ask about an item and whether there is a discount or not? and you cannot find an employee to help you? There are a lot of people facing this.

So our project aims to develop an expert system for self-service in supermarkets by using the prolog language on Switch.



Problem description

The supermarket shopping experience faces some minor problems, such as wasting the customer's time in searching for a product that is not in this supermarket. Or not being aware of the available offers. Or the lack of staff to answer simple customer questions.

Our Solution

So we develop an expert system for self-service in supermarkets to solve this problems and make the customer take a better experience.

This system enables you to:

1. View all supermarkets in the system.
2. Browse all product sections in the supermarket, and search for products that fall under the required section.
3. Review all types of products in supermarkets, and search for products that fall under the required type.
4. Review all the products available in each supermarket, and it enables you to check if the product is available in the required supermarket or not.
5. View the available offers.

Swish

SWISH is an online compiler tool to learn about the prolog language. its source is available on Github and it is under heavy development.

The SWISH application targets primarily a collaborative exploration of data combined with many other tools. and we will use it in this project.

prolog

Prolog is a logic programming language, that deals with artificial intelligence. Used for symbolic and non-numerical computation, Prolog has a built-in intelligent search mechanism and can handle complex problems in compact programs. In prolog, logic is expressed as relations called Facts and Rules which together comprise the knowledge base. Facts and rules use predicates that represent relationships among data objects.

AI APPROACH

Expert System

An expert system is a computer system that simulates the capacity of a human expert to make choices using artificial intelligence.

Expert systems are supposed to handle complicated issues by reasoning about knowledge, which is expressed as IF-THEN rules, rather than employing standard executive code. An expert system's two subsystems are the inference engine and the knowledge base. The knowledge base is a collection of facts and rules. By applying rules to known facts, the inference engine infers new facts. Inference engines can be enhanced with explanation and debugging tools.



STEPS

Initializing supermarket:
supermarkets(supermarketName).

Initializing the section:
section(sectionName).

Initializing the type:
type(typeName).

Initializing the product:
product(productName).

Initializing seasons:
Seasons(SeasonName).

Initializing sale:
Sale(SeasonName,Discount).

connection between section and type:
typeSection(sectionName, typeName).

connection between type and product:
productType(productName,typeName).

connection between supermarket and product:
supermarketProduct(supermarketsName,productName).

Establish connection between product And sales:
offer(productName,SeasonName).

Listing the product according to their supermarket:
supermarketProducts(supermarketName).

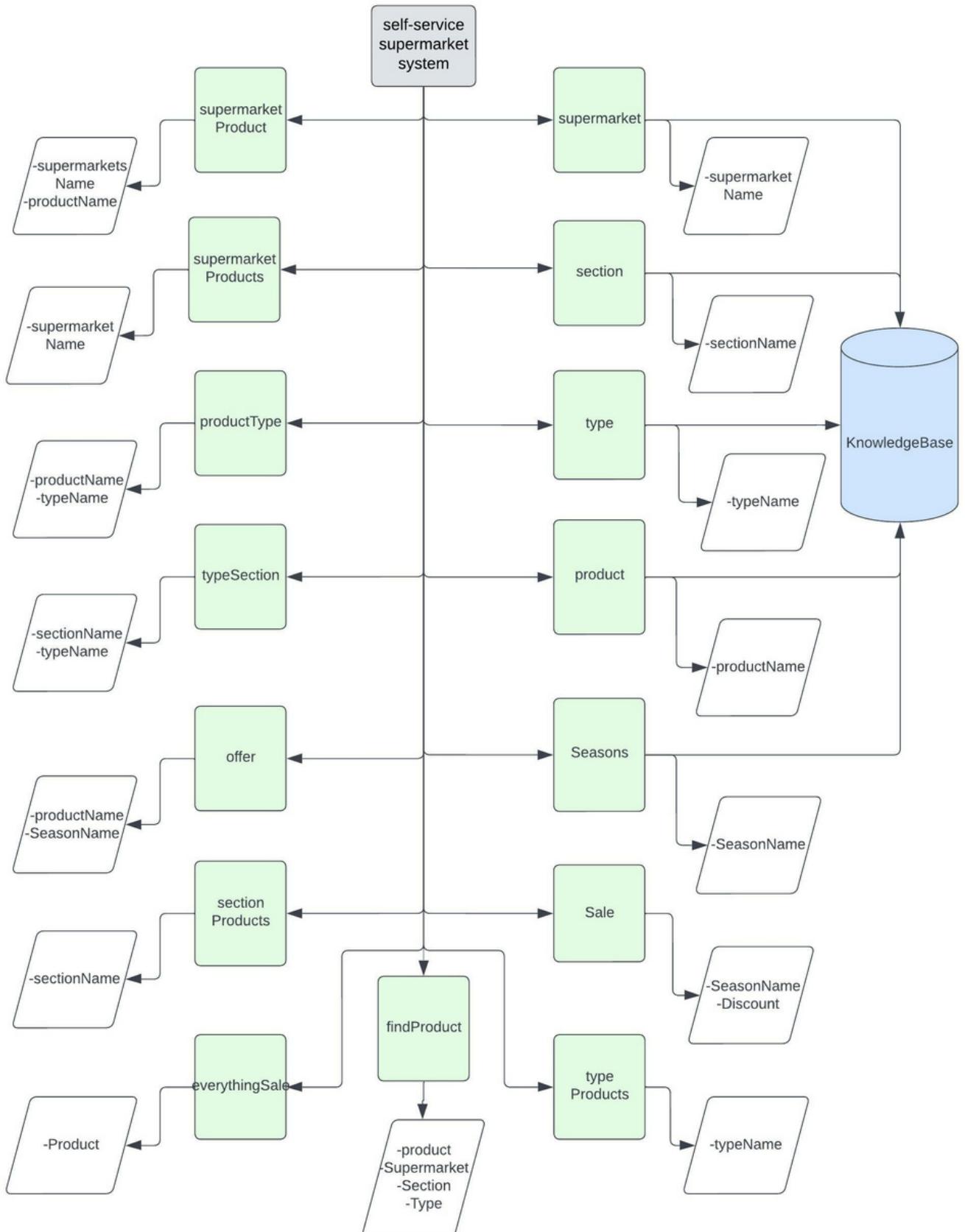
Listing the product accoding to their section:
sectionProducts(sectionName).

Listing the product with their type:
typeProducts(typeName).

Reporting everything according to the Discount :
everythingSale(Product).

find the information about product:
findProduct(product,Supermarket,Section,Type).

FLOWCHAR



DESIGN

USER MANUAL

You may see the prompt "?-" on the screen when a Prolog is executing. The user is asked to enter a query in this box.

A query is a statement that begins with a predicate and ends with its parameters, which may include variables. and the number of arguments in the query must match the number of arguments in the consulting program. The purpose of submitting a query is to find values to substitute into the variables in the query such that the query is satisfied. This is similar to asking "what values will make my statement true?".

```
?- typeSection(fruit,Type).
Breakpoint 283 in 1-st clause of everythingSale/1 at Line 295
Type = apple
Type = banana

?- typeSection(Section,tuna).
Breakpoint 284 in 1-st clause of everythingSale/1 at Line 295
Section = groceries
```

This is a query asking for the type or section. From the program, we can see all types and their section. Therefore the Prolog displays the answers.

```
?- productType(Product,chesse).
Breakpoint 298 in 1-st clause of everythingSale/1 at Line 295
Product = cheddarCheese
Product = fetaCheese

?- productType(chocolateCake,Type).
Breakpoint 299 in 1-st clause of everythingSale/1 at Line 295
Type = cake
```

This is a query asking for the type of products. From the program, we can see all types and their product. Therefore the Prolog displays the answers.

```
?- supermarketProduct(Supermarket,yellowBanana).
Breakpoint 300 in 1-st clause of everythingSale/1 at Line 295
Supermarket = binDawood
Supermarket = panda
Supermarket = alRaya

?- supermarketProduct(panda,Product).
Breakpoint 301 in 1-st clause of everythingSale/1 at Line 295
Product = redApple
Product = greenApple
Product = yellowBanana
Product = bellPapper
Product = greenPapper
Product = bigCarrot
Product = roast
Product = breast
Product = peas
Product = puffPastrySheets
Product = strawberryCake
Product = toast
Product = samoli
Product = fullFatMilk
Product = freshYogurt
Product = tideLaundryPowder
Product = fairyWashingLiquid
Product = basmatiRice
Product = goodyTuna
Product = noMareTuna
Product = cheddarCheese
Product = chickenMortadella
Product = applyJuice
Product = mirinda
Product = pepsi
```

This is a query asking for the supermarkets or products. From the program, we can see all supermarkets and their product. Therefore the Prolog displays the answers.

DESIGN

USER MANUAL

```
?- type(Type).
Breakpoint 274 in 1-st clause of everythingSale/1 at Line 295
Type = apple
Type = banana
Type = pepper
Type = carrot
Type = beef
Type = chicken
Type = frozenVegetable
Type = pastry
Type = cake
Type = bread
Type = milk
Type = yogurt
Type = laundryPowder
Type = washingLiquid
Type = rice
Type = tuna
Type = chesse
Type = mortadella
Type = juice
Type = energyDrink
```

This is a query asking for all the types. From the program, we can see all the types. Therefore the Prolog displays all the types available.

```
?- product(Product).
Breakpoint 277 in 1-st clause of everythingSale/1 at Line 295
Product = redApple
Product = greenApple
Product = yellowBanana
Product = babyBanana
Product = bellPapper
Product = greenPapper
Product = bigcarrot
Product = babycarrot
Product = steak
Product = roast
Product = breast
Product = drumstick
Product = peas
Product = corn
Product = springRollPastry
Product = puffPastrySheets
Product = chocolateCake
Product = strawberryCake
Product = toast
Product = samoli
Product = lowFatMilk
Product = fullFatMilk
Product = greekYogurt
Product = freshYogurt
Product = tideLaundryPowder
Product = arielLaundryPowder
Product = fairyWashingLiquid
Product = persilWashingLiquid
Product = basmatiRice
Product = longGrainRice
Product = goodyTuna
Product = noMareTuna
Product = cheddarCheese
Product = fetaCheese
```

This is a query asking for all the products. From the program, we can see all the products. Therefore the Prolog displays all the products available.

```
?- section(Section).
Breakpoint 273 in 1-st clause of everythingSale/1 at Line 295
Section = fruit
Section = vegetable
Section = meat
Section = frozenFood
Section = bakery
Section = dairy
Section = cleaningProducts
Section = groceries
Section = delicatessen
Section = beverage
```

This is a query asking for all the sections. From the program, we can see all the sections. Therefore the Prolog displays all the sections available.

```
?- season(Season).
Breakpoint 272 in 1-st clause of everythingSale/1 at Line 295
Season = whiteFriday
Season = nationalDay
Season = eidAlAdha
Season = eidAlFitr
Season = normalDay
```

This is a query asking for all the seasons. From the program, we can see all the seasons. Therefore the Prolog displays all the seasons.

DESIGN

USER MANUAL

```
sale(Season,Sale).
Breakpoint 307 in 1-st clause of everythingSale/1 at Line 295
Sale = "90%",
Season = whiteFriday
Sale = "80%",
Season = nationalDay
Sale = "50%",
Season = eidAlAdha
Sale = "70%",
Season = eidAlFitr
Sale = "0%",
Season = normalDay
?- sale(Season,Sale).
```

This is a query asking for the seasons of sales. From the program, we can see the sales for each season. Therefore the Prolog display both sets of results.

```
everythingSale(_).
Breakpoint 261 in 1-st clause of everythingSale/1 at Line 295
Call: supermarketProduct(_5616,_5246)
Supermarket: binDawood, Section: fruit, Type: banana, has yellowBanana Offers 90% In whiteFriday
Supermarket: binDawood, Section: fruit, Type: banana, has yellowBanana Offers 80% In nationalDay
Supermarket: binDawood, Section: fruit, Type: banana, has yellowBanana Offers 50% In eidAlAdha
Supermarket: binDawood, Section: fruit, Type: banana, has yellowBanana Offers 70% In eidAlFitr
Supermarket: binDawood, Section: fruit, Type: banana, has yellowBanana Offers 0% In normalDay
Supermarket: binDawood, Section: fruit, Type: banana, has babyBanana Offers 90% In whiteFriday
Supermarket: binDawood, Section: fruit, Type: banana, has babyBanana Offers 80% In nationalDay
Supermarket: binDawood, Section: fruit, Type: banana, has babyBanana Offers 50% In eidAlAdha
Supermarket: binDawood, Section: fruit, Type: banana, has babyBanana Offers 70% In eidAlFitr
Supermarket: binDawood, Section: vegetable, Type: pepper, has bellPapper Offers 90% In normalDay
Supermarket: binDawood, Section: vegetable, Type: pepper, has bellPapper Offers 80% In nationalDay
Supermarket: binDawood, Section: vegetable, Type: pepper, has bellPapper Offers 50% In eidAlAdha
Supermarket: binDawood, Section: vegetable, Type: pepper, has bellPapper Offers 70% In eidAlFitr
Supermarket: binDawood, Section: vegetable, Type: pepper, has bellPapper Offers 0% In whiteFriday
Supermarket: binDawood, Section: vegetable, Type: carrot, has bigcarrot Offers 90% In whiteFriday
Supermarket: binDawood, Section: vegetable, Type: carrot, has bigcarrot Offers 80% In nationalDay
Supermarket: binDawood, Section: vegetable, Type: carrot, has bigcarrot Offers 50% In eidAlAdha
Supermarket: binDawood, Section: vegetable, Type: carrot, has bigcarrot Offers 70% In eidAlFitr
Supermarket: binDawood, Section: vegetable, Type: carrot, has bigcarrot Offers 0% In normalDay
Supermarket: binDawood, Section: meat, Type: beef, has roast Offers 90% In whiteFriday
Supermarket: binDawood, Section: meat, Type: beef, has roast Offers 80% In nationalDay
Supermarket: binDawood, Section: meat, Type: beef, has roast Offers 50% In eidAlAdha
Supermarket: binDawood, Section: meat, Type: beef, has roast Offers 70% In eidAlFitr
Supermarket: binDawood, Section: meat, Type: beef, has roast Offers 0% In normalDay
Supermarket: binDawood, Section: meat, Type: chicken, has breast Offers 90% In whiteFriday
Supermarket: binDawood, Section: meat, Type: chicken, has breast Offers 80% In nationalDay
Supermarket: binDawood, Section: meat, Type: chicken, has breast Offers 50% In eidAlAdha
Supermarket: binDawood, Section: meat, Type: chicken, has breast Offers 70% In eidAlFitr
Supermarket: binDawood, Section: meat, Type: chicken, has breast Offers 0% In normalDay
Supermarket: binDawood, Section: meat, Type: chicken, has drumstick Offers 90% In whiteFriday
Supermarket: binDawood, Section: meat, Type: chicken, has drumstick Offers 80% In nationalDay
Supermarket: binDawood, Section: meat, Type: chicken, has drumstick Offers 50% In eidAlAdha
?- everythingSale(_).
```

```
supermarket(Supermarket).
Breakpoint 270 in 1-st clause of everythingSale/1 at Line 295
Supermarket = binDawood
Supermarket = panda
Supermarket = alRaya
Supermarket = danube
?- supermarket(Supermarket).
```

This is a query asking for all the supermarkets. From the program, we can see all the supermarkets. Therefore the Prolog display all the supermarkets in the system.

This is a query asking for everything in sales. From the program, we can see all the supermarkets with their section, types, products, offices, and seasons. Therefore the Prolog displays all the set of results.

DESIGN

USER MANUAL

```
?- supermarketProducts(alRaya).
Supermarket: alRaya has: greenApple
Supermarket: alRaya has: yellowBanana
Supermarket: alRaya has: greenPapper
Supermarket: alRaya has: bigcarrot
Supermarket: alRaya has: steak
Supermarket: alRaya has: roast
Supermarket: alRaya has: drumstick
Supermarket: alRaya has: springRollPastry
Supermarket: alRaya has: chocolateCake
Supermarket: alRaya has: strawberryCake
Supermarket: alRaya has: toast
Supermarket: alRaya has: samoli
Supermarket: alRaya has: fullFatMilk
Supermarket: alRaya has: freshYogurt
Supermarket: alRaya has: arielLaundryPowder
Supermarket: alRaya has: persilWashingLiquid
Supermarket: alRaya has: longGrainRice
Supermarket: alRaya has: goodyTuna
Supermarket: alRaya has: cheddarCheese
Supermarket: alRaya has: fetaCheese
Supermarket: alRaya has: applyJuice
Supermarket: alRaya has: mangoJuice
Supermarket: alRaya has: pepsi
false

?- supermarketProducts(alRaya).
```

This is a query asking for products in supermarkets. From the program, we can see all the supermarkets with their products. Therefore the Prolog displays all the products.

```
?- typeProducts(apple).
Type: apple has: redApply
Type: apple has: greenApply
false

?- typeProducts(apple).
```

This is a query asking for finding products from the type. From the program, we can see all the types and their products. Therefore the Prolog displays the answers.

```
?- offer(steak,Season).
Breakpoint 309 in 1-st clause of everythingSale/1 at Line 295
Product: steak
Offers: 90% In whiteFriday
Product: steak
Offers: 80% In nationalDay
Product: steak
Offers: 50% In eidAlAdha
Product: steak
Offers: 70% In eidAlFitr
Product: steak
Offers: 0% In normalDay
false

?- offer(redApple,nationalDay).
Breakpoint 312 in 1-st clause of everythingSale/1 at Line 295
Product: redApple
Offers: 80% In nationalDay
false
```

This is a query to asking for the offer of product in season. From the program, we can see all offers. Therefore the Prolog displays all the answers.

```
?- sectionProducts(vegetable).
Section: vegetable has: bellPapper
Section: vegetable has: greenPapper
Section: vegetable has: bigcarrot
Section: vegetable has: babycarrot
false

?- sectionProducts(vegetable).
```

This is a query asking for finding products from the section. From the program, we can see all the sections and their products. Therefore the Prolog displays the answers.

DESIGN

USER MANUAL

```
?- findProduct(Product,danube,Section,Type).
Breakpoint 265 in 1-st clause of everythingSale/1 at 10 Line 295
Product = babyBanana,
Section = fruit,
Type = banana
Product = greenPapper,
Section = vegetable,
Type = pepper
Product = babycarrot,
Section = vegetable,
Type = carrot
Product = roast,
Section = meat,
Type = beef
Product = peas,
Section = frozenFood,
Type = frozenVegetable
Product = corn,
Section = frozenFood,
Type = frozenVegetable
Product = springRollPastry,
Section = frozenFood,
Type = pastry
Product = chocolateCake,
Section = bakery,
Type = cake
Product = strawberryCake,
Section = bakery,
Type = cake
Product = toast,
Section = bakery,
Type = bread
Product = samoli,
Section = bakery,
Type = bread
Product = lowFatMilk,
?- findProduct(Product,danube,Section,Type).
```

This is a query asking for finding supermarkets. From the program, we can see all the products, sections, and types. Therefore the Prolog displays the set of results.

```
?- findProduct(toast,Supermarket,Section,Type).
Breakpoint 263 in 1-st clause of everythingSale/1 at 10 Line 295
Section = bakery,
Supermarket = binDawood,
Type = bread
Section = bakery,
Supermarket = panda,
Type = bread
Section = bakery,
Supermarket = alRayya,
Type = bread
Section = bakery,
Supermarket = danube,
Type = bread
false
?- findProduct(toast,Supermarket,Section,Type).
```

This is a query asking for finding products. From the program, we can see all the supermarkets with their sections and types. Therefore the Prolog displays the set of results.

```
?- findProduct(Product,Supermarket,Section,rice).
Breakpoint 267 in 1-st clause of everythingSale/1 at 10 Line 295
Product = basmatiRice,
Section = groceries,
Supermarket = binDawood
Product = longGrainRice,
Section = groceries,
Supermarket = binDawood
Product = basmatiRice,
Section = groceries,
Supermarket = panda
Product = longGrainRice,
Section = groceries,
Supermarket = alRayya
Product = basmatiRice,
Section = groceries,
Supermarket = danube
Product = longGrainRice,
Section = groceries,
Supermarket = danube
?- findProduct(Product,Supermarket,Section,rice).
```

This is a query asking for finding types. From the program, we can see all the supermarkets with their products and their sections. Therefore the Prolog displays the set of results.

```
?- findProduct(Product,Supermarket,cleaningProducts,Type).
Breakpoint 266 in 1-st clause of everythingSale/1 at 10 Line 295
Product = tideLaundryPowder,
Supermarket = binDawood,
Type = laundryPowder
Product = ArielLaundryPowder,
Supermarket = binDawood,
Type = laundryPowder
Product = persilWashingLiquid,
Supermarket = binDawood,
Type = washingLiquid
Product = tideLaundryPowder,
Supermarket = panda,
Type = laundryPowder
Product = fairyWashingLiquid,
Supermarket = panda,
Type = washingLiquid
Product = ArielLaundryPowder,
Supermarket = alRayya,
Type = laundryPowder
Product = persilWashingLiquid,
Supermarket = alRayya,
Type = washingLiquid
Product = ArielLaundryPowder,
Supermarket = danube,
Type = laundryPowder
Product = persilWashingLiquid,
Supermarket = danube,
Type = washingLiquid
?- findProduct(Product,Supermarket,cleaningProducts,Type).
```

This is a query asking for finding sections. From the program, we can see all the supermarkets with their products and types. Therefore the Prolog displays the set of results.



DESIGN CODE

```
%Initializing Supermarkets
supermarket(binDawood).
supermarket(panda).
supermarket(alRaya).
supermarket(danube).
```

```
%Initializing Seasons
season(whiteFriday).
season(nationalDay).
season(eidAlAdha).
season(eidAlFitr).
season(normalDay).
```

```
%Initializing Sales to add the Discounts
sale(whiteFriday,"90%").
sale(nationalDay,"80%").
sale(eidAlAdha,"50%").
sale(eidAlFitr,"70%").
sale(normalDay,"0%").
```

```
%Initializing Sections
section(fruit).
section(vegetable).
section(meat).
section(frozenFood).
section(bakery).
section(dairy).
section(cleaningProducts).
section(groceries).
section(delicatessen).
section(beverage).
```

```
%Initializing Types
type(apple).
type(banana).
type(pepper).
type(carrot).
type(beef).
type(chicken).
type(frozenVegetable).
type(pastry).
type(cake).
type(bread).
type(milk).
type(yogurt).
type(laundryPowder).
type(washingLiquid).
type(rice).
type(tuna).
type(chesse).
type(mortadella).
type(juice).
type(energyDrink).
```

```
%Initializing Products
product(redApple).
product(greenApple).
product(yellowBanana).
product(babyBanana).
product(bellPapper).
product(greenPapper).
product(bigcarrot).
product(babycarrot).
product(steak).
product(roast).
product(breast).
product(drumstick).
product(peas).
product(corn).
product(springRollPastry).
product(puffPastrySheets).
product(chocolateCake).
product(strawberryCake).
product(toast).
product(samoli).
product(lowFatMilk).
product(fullFatMilk).
product(greekYogurt).
product(freshYogurt).
product(tideLaundryPowder).
product(rielLaundryPowder).
product(fairyWashingLiquid).
product(persilWashingLiquid).
product(basmatiRice).
product(longGrainRice).
product(goodyTuna).
product(rioMareTuna).
product(cheddarCheese).
product(fetaCheese).
product(beefMortadella).
product(chickenMortadella).
product(applyJuice).
product(mangoJuice).
product(mirinda).
product(pepsi).
```

```
% add types in their section
typeSection(fruit,apple).
typeSection(fruit,banana).
typeSection(vegetable,pepper).
typeSection(vegetable,carrot).
typeSection(meat,beef).
typeSection(meat,chicken).
typeSection(frozenFood,frozenVegetable).
typeSection(frozenFood,pastry).
typeSection(bakery,cake).
typeSection(bakery,bread).
typeSection(dairy,milk).
typeSection(dairy,yogurt).
typeSection(cleaningProducts,laundryPowder).
typeSection(cleaningProducts,washingLiquid).
typeSection(groceries,rice).
typeSection(groceries,tuna).
typeSection(delicatessen,chesse).
typeSection(delicatessen,mortadella).
typeSection(beverage,juice).
typeSection(beverage,energyDrink).
```



SELF-SERVICE SUPERMARKET

% add products in their Type

```
productType(redApple,apple).
productType(greenApple,apple).
productType(yellowBanana,banana).
productType(babyBanana,banana).
productType(bellPapper,pepper).
productType(greenPapper,pepper).
productType(bigcarrot,carrot).
productType(babycarrot,carrot).
productType(steak,beef).
productType(roast,beef).
productType(breast,chicken).
productType(drumstick,chicken).
productType(peas,frozenVegetable).
productType(corn,frozenVegetable).
productType(springRollPastry,pastry).
productType(puffPastrySheets,pastry).
productType(chocolateCake,cake).
productType(strawberryCake,cake).
productType(toast,bread).
productType(samoli,bread).
productType(lowFatMilk,milk).
productType(fullFatMilk,milk).
productType(greekYogurt,yogurt).
productType(freshYogurt,yogurt).
productType(tideLaundryPowder,laundryPowder).
productType(arielLaundryPowder,laundryPowder).
productType(fairyWashingLiquid,washingLiquid).
productType(persilWashingLiquid,washingLiquid).
productType(basmatiRice,rice).
productType(longGrainRice,rice).
productType(goodyTuna,tuna).
productType(rioMareTuna,tuna).
productType(cheddarCheese,chesse).
productType(fetaCheese,chesse).
productType(beefMortadella,mortadella).
productType(chickenMortadella,mortadella).
productType(applyJuice,juice).
productType(mangoJuice,juice).
productType(mirinda,energyDrink).
productType(pepsi,energyDrink).
```

% add products available in supermarkets:

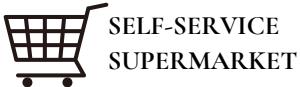
```
% binDawood
supermarketProduct(binDawood,yellowBanana).
supermarketProduct(binDawood,babyBanana).
supermarketProduct(binDawood,bellPapper).
supermarketProduct(binDawood,bigcarrot).
supermarketProduct(binDawood,roast).
supermarketProduct(binDawood,breast).
supermarketProduct(binDawood,drumstick).
supermarketProduct(binDawood,corn).
supermarketProduct(binDawood,puffPastrySheets).
supermarketProduct(binDawood,chocolateCake).
supermarketProduct(binDawood,strawberryCake).
supermarketProduct(binDawood,toast).
supermarketProduct(binDawood,lowFatMilk).
supermarketProduct(binDawood,greekYogurt).
supermarketProduct(binDawood,freshYogurt).
supermarketProduct(binDawood,tideLaundryPowder).
supermarketProduct(binDawood,arielLaundryPowder).
supermarketProduct(binDawood,persilWashingLiquid).
supermarketProduct(binDawood,basmatiRice).
supermarketProduct(binDawood,longGrainRice).
supermarketProduct(binDawood,rioMareTuna).
supermarketProduct(binDawood,cheddarCheese).
supermarketProduct(binDawood,beefMortadella).
supermarketProduct(binDawood,applyJuice).
supermarketProduct(binDawood,mangoJuice).
supermarketProduct(binDawood,mirinda).
```

%panda

```
supermarketProduct(panda,redApple).
supermarketProduct(panda,greenApple).
supermarketProduct(panda,yellowBanana).
supermarketProduct(panda,bellPapper).
supermarketProduct(panda,greenPapper).
supermarketProduct(panda,bigcarrot).
supermarketProduct(panda,roast).
supermarketProduct(panda,breast).
supermarketProduct(panda,peas).
supermarketProduct(panda,puffPastrySheets).
supermarketProduct(panda,strawberryCake).
supermarketProduct(panda,toast).
supermarketProduct(panda,samoli).
supermarketProduct(panda,fullFatMilk).
supermarketProduct(panda,freshYogurt).
supermarketProduct(panda,tideLaundryPowder).
supermarketProduct(panda,fairyWashingLiquid).
supermarketProduct(panda,basmatiRice).
supermarketProduct(panda,goodyTuna).
supermarketProduct(panda,rioMareTuna).
supermarketProduct(panda,cheddarCheese).
supermarketProduct(panda,chickenMortadella).
supermarketProduct(panda,applyJuice).
supermarketProduct(panda,mirinda).
supermarketProduct(panda,pepsi).
%alRaya
supermarketProduct(alRaya,greenApple).
supermarketProduct(alRaya,yellowBanana).
supermarketProduct(alRaya,greenPapper).
supermarketProduct(alRaya,bigcarrot).
supermarketProduct(alRaya,steak).
supermarketProduct(alRaya,roast).
supermarketProduct(alRaya,drumstick).
supermarketProduct(alRaya,springRollPastry).
supermarketProduct(alRaya,chocolateCake).
supermarketProduct(alRaya,strawberryCake).
supermarketProduct(alRaya,toast).
supermarketProduct(alRaya,samoli).
supermarketProduct(alRaya,fullFatMilk).
supermarketProduct(alRaya,freshYogurt).
supermarketProduct(alRaya,arielLaundryPowder).
supermarketProduct(alRaya,persilWashingLiquid).
supermarketProduct(alRaya,longGrainRice).
supermarketProduct(alRaya,goodyTuna).
supermarketProduct(alRaya,cheddarCheese).
supermarketProduct(alRaya,fetaCheese).
supermarketProduct(alRaya,applyJuice).
supermarketProduct(alRaya,mangoJuice).
supermarketProduct(alRaya,pepsi).
```

%danube

```
supermarketProduct(danube,redApple).
supermarketProduct(danube,babyBanana).
supermarketProduct(danube,greenPapper).
supermarketProduct(danube,babycarrot).
supermarketProduct(danube,roast).
supermarketProduct(danube,peas).
supermarketProduct(danube,corn).
supermarketProduct(danube,springRollPastry).
supermarketProduct(danube,chocolateCake).
supermarketProduct(danube,strawberryCake).
supermarketProduct(danube,toast).
supermarketProduct(danube,samoli).
supermarketProduct(danube,lowFatMilk).
supermarketProduct(danube,freshYogurt).
supermarketProduct(danube,arielLaundryPowder).
supermarketProduct(danube,persilWashingLiquid).
supermarketProduct(danube,basmatiRice).
supermarketProduct(danube,longGrainRice).
supermarketProduct(danube,goodyTuna).
supermarketProduct(danube,rioMareTuna).
supermarketProduct(danube,fetaCheese).
supermarketProduct(danube,chickenMortadella).
supermarketProduct(danube,mangoJuice).
supermarketProduct(danube,mirinda).
```



%-----

%print the Product with their Offers

offer(Product,Season):-

```
product(Product),sale(Season,Sale),
write(" Product: "),write(Product), nl,
write(" Offers: "), write(Sale),
write(" In "), write(Season),
nl,fail.
```

%list the Products according to their supermarket

supermarketProducts(Supermarket):-

```
supermarketProduct(Supermarket,Product),
write("Supermarket: "),write(Supermarket),
write(" has: "), write(Product),nl,fail.
```

%list the Products according to their section

sectionProducts(Section):-

```
typeSection(Section,Type),productType(Product,Type),
write("Section: "),write(Section),
write(" has: "), write(Product),nl,fail.
```

%list the Products according to their type

typeProducts(Type):-

```
productType(Product,Type),
write("Type: "),write(Type),
write(" has: "), write(Product),nl,fail.
```

% list everything about products

everythingSale(Product):-

```
supermarketProduct(Supermarket,Product),typeSection(Section,Type)
,productType(Product,Type),sale(Season,Sale),
write("Supermarket: "),write(Supermarket),
write(", Section: "),write(Section),
write(", Type: "),write(Type),
write(", has "),write(Product),
write(" Offers "),write(Sale),
write(" In "),write(Season),nl,fail.
```

% to find information about specific Product

findProduct(Product,Supermarket,Section,Type):-

```
supermarketProduct(Supermarket,Product),typeSection(Section,Type)
,productType(Product,Type).
```

DESIGN OUTPUT

```
typeSection(fruit,Type).
Breakpoint 283 in 1-st clause of everythingSale/1 at Line 295
Type = apple
Type = banana

typeSection(Section,tuna).
Breakpoint 284 in 1-st clause of everythingSale/1 at Line 295
Section = groceries
```

```
productType(Product,chesse).
Breakpoint 298 in 1-st clause of everythingSale/1 at Line 295
Product = cheddarCheese
Product = fetaCheese

productType(chocolateCake,Type).
Breakpoint 299 in 1-st clause of everythingSale/1 at Line 295
Type = cake
```

```
supermarketProduct(Supermarket,yellowBanana).
Breakpoint 300 in 1-st clause of everythingSale/1 at Line 295
Supermarket = binDawood
Supermarket = panda
Supermarket = alRaya

supermarketProduct(panda,Product).
Breakpoint 301 in 1-st clause of everythingSale/1 at Line 295
Product = redApple
Product = greenApple
Product = yellowBanana
Product = bellPapper
Product = greenPapper
Product = bigcarrot
Product = babycarrot
Product = steak
Product = roast
Product = breast
Product = drumstick
Product = peas
Product = corn
Product = springRollPastry
Product = puffPastrySheets
Product = chocolateCake
Product = strawberryCake
Product = toast
Product = samoli
Product = lowFatMilk
Product = fullFatMilk
Product = greekYogurt
Product = freshYogurt
Product = tideLaundryPowder
Product = fairyWashingLiquid
Product = basmatiRice
Product = goodyTuna
Product = noMareTuna
Product = cheddarCheese
Product = chickenMortadella
Product = applyJuice
Product = mirinda
Product = pepsi
```

```
type(Type).
Breakpoint 274 in 1-st clause of everythingSale/1 at Line 295
Type = apple
Type = banana
Type = pepper
Type = carrot
Type = beef
Type = chicken
Type = frozenVegetable
Type = pastry
Type = cake
Type = bread
Type = milk
Type = yogurt
Type = laundryPowder
Type = washingLiquid
Type = rice
Type = tuna
Type = chesse
Type = mortadella
Type = juice
Type = energyDrink

?- type(Type).
```

```
sale(Season,Sale).
Breakpoint 307 in 1-st clause of everythingSale/1 at Line 295
Sale = "90%", Season = whiteFriday
Sale = "80%", Season = nationalDay
Sale = "50%", Season = eidAlAdha
Sale = "70%", Season = eidAlFitr
Sale = "0%", Season = normalDay

?- sale(Season,Sale).
```

```
product(Product).
Breakpoint 277 in 1-st clause of everythingSale/1 at Line 295
Product = redApple
Product = greenApple
Product = yellowBanana
Product = babyBanana
Product = bellPapper
Product = greenPapper
Product = bigcarrot
Product = babycarrot
Product = steak
Product = roast
Product = breast
Product = drumstick
Product = peas
Product = corn
Product = springRollPastry
Product = puffPastrySheets
Product = chocolateCake
Product = strawberryCake
Product = toast
Product = samoli
Product = lowFatMilk
Product = fullFatMilk
Product = greekYogurt
Product = freshYogurt
Product = tideLaundryPowder
Product = arielLaundryPowder
Product = fairyWashingLiquid
Product = persilWashingLiquid
Product = basmatiRice
Product = longGrainRice
Product = goodyTuna
Product = noMareTuna
Product = cheddarCheese
Product = fetaCheese

?- product(Product).
```

```
section(Section).
Breakpoint 273 in 1-st clause of everythingSale/1 at Line 295
Section = fruit
Section = vegetable
Section = meat
Section = frozenFood
Section = bakery
Section = dairy
Section = cleaningProducts
Section = groceries
Section = delicatessen
Section = beverage

?- section(Section).
```

```
supermarket(Supermarket).
Breakpoint 270 in 1-st clause of everythingSale/1 at Line 295
Supermarket = binDawood
Supermarket = panda
Supermarket = alRaya
Supermarket = danube

?- supermarket(Supermarket).
```

DESIGN OUTPUT

```

season(Season).
Breakpoint 272 in 1-st clause of everythingSale/1 at Line 295
Season = whiteFriday
Season = nationalDay
Season = eidAlAdha
Season = eidAlFitr
Season = normalDay

?- season(Season).

offer(steak,Season).
Breakpoint 309 in 1-st clause of everythingSale/1 at Line 295
Product: steak
Offers: 90% in whiteFriday
Product: steak
Offers: 80% in nationalDay
Product: steak
Offers: 50% in eidAlAdha
Product: steak
Offers: 70% in eidAlFitr
Product: steak
Offers: 0% in normalDay
false

offer(redApple,nationalDay).
Breakpoint 312 in 1-st clause of everythingSale/1 at Line 295
Product: redApple
Offers: 80% in nationalDay
false

supermarketProducts(aiRay).
Supermarket: aiRay has: greenApple
Supermarket: aiRay has: yellowBanana
Supermarket: aiRay has: greenPapper
Supermarket: aiRay has: bigcarrot
Supermarket: aiRay has: steak
Supermarket: aiRay has: roast
Supermarket: aiRay has: drumstick
Supermarket: aiRay has: springRollPastry
Supermarket: aiRay has: chocolateCake
Supermarket: aiRay has: strawberryCake
Supermarket: aiRay has: toast
Supermarket: aiRay has: bellPapper

everythingSale(_).
Breakpoint 261 in 1-st clause of everythingSale/1 at Line 295
Call: supermarketProduct(_5616,_5246)
Supermarket: binDawood, Section: fruit, Type: banana, has yellowBanana Offers 90% in whiteFriday
Supermarket: binDawood, Section: fruit, Type: banana, has yellowBanana Offers 80% in nationalDay
Supermarket: binDawood, Section: fruit, Type: banana, has yellowBanana Offers 50% in eidAlAdha
Supermarket: binDawood, Section: fruit, Type: banana, has yellowBanana Offers 70% in eidAlFitr
Supermarket: binDawood, Section: fruit, Type: banana, has yellowBanana Offers 0% in normalDay
Supermarket: binDawood, Section: fruit, Type: banana, has babyBanana Offers 90% in whiteFriday
Supermarket: binDawood, Section: fruit, Type: banana, has babyBanana Offers 80% in nationalDay
Supermarket: binDawood, Section: fruit, Type: banana, has babyBanana Offers 50% in eidAlAdha
Supermarket: binDawood, Section: fruit, Type: banana, has babyBanana Offers 70% in eidAlFitr
Supermarket: binDawood, Section: fruit, Type: banana, has babyBanana Offers 0% in normalDay
Supermarket: binDawood, Section: vegetable, Type: pepper, has bellPapper Offers 90% in whiteFriday

sectionProducts(vegetable).
Section: vegetable has: bellPapper
Section: vegetable has: greenPapper
Section: vegetable has: bigcarrot
Section: vegetable has: babycarrot
false

?- sectionProducts(vegetable).

typeProducts(apple).
Type: apple has: redApply
Type: apple has: greenApply
false

?- typeProducts(apple).

findProduct(Product,danube,Section,Type).
Breakpoint 265 in 1-st clause of everythingSale/1 at Line 295
Product = babyBanana,
Section = fruit,
Type = banana
Product = greenPapper,
Section = vegetable,
Type = pepper
Product = babycarrot,
Section = vegetable,
Type = carrot
Product = roast,
Section = meat,
Type = beef
Product = peas,
Section = frozenFood,
Type = frozenVegetable
Product = corn

findProduct(Product,Supermarket,cleaningProducts,Type).
Breakpoint 266 in 1-st clause of everythingSale/1 at Line 295
Product = tideLaundryPowder,
Supermarket = binDawood,
Type = laundryPowder
Product = arielLaundryPowder,
Supermarket = binDawood,
Type = laundryPowder
Product = persilWashingLiquid,
Supermarket = binDawood,
Type = washingLiquid
Product = tideLaundryPowder,
Supermarket = panda,
Type = laundryPowder
Product = fairyWashingLiquid,
Supermarket = panda,
Type = washingLiquid
Product = arielLaundryPowder,
Supermarket = aiRay,
Type = laundryPowder
Product = persilWashingLiquid,
Supermarket = aiRay,
Type = washingLiquid

findProduct(toast,Supermarket,Section,Type).
Breakpoint 263 in 1-st clause of everythingSale/1 at Line 295
Section = bakery,
Supermarket = binDawood,
Type = bread
Section = bakery,
Supermarket = panda,
Type = bread
Section = bakery,
Supermarket = aiRay,
Type = bread
Section = bakery,
Supermarket = danube,
Type = bread
false

?- findProduct(toast,Supermarket,Section,Type).

findProduct(Product,Supermarket,Section,rice).
Breakpoint 267 in 1-st clause of everythingSale/1 at Line 295
Product = basmatiRice,
Section = groceries,
Supermarket = binDawood
Product = longGrainRice,
Section = groceries,
Supermarket = binDawood
Product = basmatiRice,
Section = groceries,
Supermarket = panda
Product = longGrainRice,
Section = groceries,
Supermarket = aiRay
Product = basmatiRice,
Section = groceries,
Supermarket = danube
Product = longGrainRice,
Section = groceries,
Supermarket = danube
?- findProduct(Product,Supermarket,Section,rice).

```

Implementation

CODE TESTING

```
supermarketProducts(binDawood).
```

Supermarket: binDawood has: yellowBanana
 Supermarket: binDawood has: babyBanana
 Supermarket: binDawood has: bellPapper
 Supermarket: binDawood has: bigcarrot
 Supermarket: binDawood has: roast
 Supermarket: binDawood has: breast
 Supermarket: binDawood has: drumstick
 Supermarket: binDawood has: corn
 Supermarket: binDawood has: puffPastrySheets
 Supermarket: binDawood has: chocolateCake
 Supermarket: binDawood has: strawberryCake
 Supermarket: binDawood has: toast
 Supermarket: binDawood has: lowFatMilk
 Supermarket: binDawood has: greekYogurt
 Supermarket: binDawood has: freshYogurt
 Supermarket: binDawood has: tideLaundryPowder
 Supermarket: binDawood has: arielLaundryPowder
 Supermarket: binDawood has: persilWashingLiquid
 Supermarket: binDawood has: basmatiRice
 Supermarket: binDawood has: longGrainRice
 Supermarket: binDawood has: rioMareTuna
 Supermarket: binDawood has: cheddarCheese
 Supermarket: binDawood has: beefMortadella
 Supermarket: binDawood has: applyJuice
 Supermarket: binDawood has: mangoJuice
 Supermarket: binDawood has: mirinda
false

This is a query searched for all "Bin Dawood" supermarket products.

```
supermarketProducts(noori).
```

false

This is a query searched for "Nori" supermarket products, but no results appear because the system does not support this supermarket

```
?- supermarketProducts(X).
```

Supermarket: binDawood has: greekYogurt
 Supermarket: binDawood has: freshYogurt
 Supermarket: binDawood has: tideLaundryPowder
 Supermarket: binDawood has: arielLaundryPowder
 Supermarket: binDawood has: persilWashingLiquid
 Supermarket: binDawood has: basmatiRice
 Supermarket: binDawood has: longGrainRice
 Supermarket: binDawood has: rioMareTuna
 Supermarket: binDawood has: cheddarCheese
 Supermarket: binDawood has: beefMortadella
 Supermarket: binDawood has: applyJuice
 Supermarket: binDawood has: mangoJuice
 Supermarket: binDawood has: mirinda
 Supermarket: panda has: redApple
 Supermarket: panda has: greenApple
 Supermarket: panda has: yellowBanana
 Supermarket: panda has: bellPapper
 Supermarket: panda has: greenPapper
 Supermarket: panda has: bigcarrot

This is a query searched for all products in all supermarkets.

```
offer(mangoJuice,eidAlAdha)
```

Product: mangoJuice
 Offers: 50% In eidAlAdha
false

This is a query Searched for mango juice offers on Eid Al-Adha

```
offer(mangoJuice,ramadan)
```

false

There are no offers for mango juice in Ramadan

```
offer(mangoJuice,X)
```

Product: mangoJuice
 Offers: 90% In whiteFriday
 Product: mangoJuice
 Offers: 80% In nationalDay
 Product: mangoJuice
 Offers: 50% In eidAlAdha
 Product: mangoJuice
 Offers: 70% In eidAlFitr
 Product: mangoJuice
 Offers: 0% In nurnalDay
false

This is a query Searched for all offers of mango juice

Implementation

CODE TESTING

 `typeProducts(X).`

```
Type: apple has: redApply
Type: apple has: greenApply
Type: banana has: yellowBanana
Type: banana has: babyBanana
Type: pepper has: bellPapper
Type: pepper has: greenPapper
Type: carrot has: bigcarrot
```

This is a query searched for all products in each type

 `sectionProducts(meat).`

```
Section: meat has: steak
Section: meat has: roast
Section: meat has: breast
Section: meat has: drumstick
false
```

This is a query searched for Section meat products

 `typeProducts(orange).`

```
false
```

there are no result for orange type

 `sectionProducts(bakery).`

```
Section: bakery has: chocolateCake
Section: bakery has: strawberryCake
Section: bakery has: toast
Section: bakery has: samoli
false
```

This is a query searched for Section Bakery products

 `typeProducts(apple).`

```
Type: apple has: redApply
Type: apple has: greenApply
false
```

This is a query searched for Type Apple products

 `typeProducts(juice).`

```
Type: juice has: applyJuice
Type: juice has: mangoJuice
false
```

This is a query searched for Type Juice products

 `sectionProducts(X).`

```
Section: fruit has: redApply
Section: fruit has: greenApply
Section: fruit has: yellowBanana
Section: fruit has: babyBanana
Section: vegetable has: bellPapper
Section: vegetable has: greenPapper
Section: vegetable has: bigcarrot
Section: vegetable has: babycarrot
Section: meat has: steak
Section: meat has: roast
Section: meat has: breast
Section: meat has: drumstick
Section: frozenFood has: peas
Section: frozenFood has: corn
Section: frozenFood has: springRollPastry
Section: frozenFood has: puffPastrySheets
Section: bakery has: chocolateCake
Section: bakery has: strawberryCake
Section: bakery has: toast
Section: bakery has: samoli
Section: dairy
```

This is a query searched for all products in each section

Implementation

CODE TESTING

 `everythingSale(chocolateCake).`

```
Supermarket: binDawood, Section: bakery, Type: cake, has chocolateCake Offers 90% In whiteFriday
Supermarket: binDawood, Section: bakery, Type: cake, has chocolateCake Offers 80% In nationalDay
Supermarket: binDawood, Section: bakery, Type: cake, has chocolateCake Offers 50% In eidAlAdha
Supermarket: binDawood, Section: bakery, Type: cake, has chocolateCake Offers 70% In eidAlFitr
Supermarket: binDawood, Section: bakery, Type: cake, has chocolateCake Offers 0% In nurnalDay
Supermarket: alRaya, Section: bakery, Type: cake, has chocolateCake Offers 90% In whiteFriday
Supermarket: alRaya, Section: bakery, Type: cake, has chocolateCake Offers 80% In nationalDay
Supermarket: alRaya, Section: bakery, Type: cake, has chocolateCake Offers 50% In eidAlAdha
Supermarket: alRaya, Section: bakery, Type: cake, has chocolateCake Offers 70% In eidAlFitr
Supermarket: alRaya, Section: bakery, Type: cake, has chocolateCake Offers 0% In nurnalDay
Supermarket: danube, Section: bakery, Type: cake, has chocolateCake Offers 90% In whiteFriday
Supermarket: danube, Section: bakery, Type: cake, has chocolateCake Offers 80% In nationalDay
Supermarket: danube, Section: bakery, Type: cake, has chocolateCake Offers 50% In eidAlAdha
Supermarket: danube, Section: bakery, Type: cake, has chocolateCake Offers 70% In eidAlFitr
Supermarket: danube, Section: bakery, Type: cake, has chocolateCake Offers 0% In nurnalDay
false
```

This is a query searched for every thing about chocolateCake

 `everythingSale(kiwi).`

```
false
```

No results for kiwi, not available in any supermarket

 `findProduct(mirinda,binDawood,Section,Type).`

```
Section = beverage,
Type = energyDrink
```

This is a query Searched for mirinda in "Bin Dawood" supermarket

 `findProduct(mirinda,alRaya,Section,Type).`

```
false
```

There is no Miranda in "Al-Raya" supermarket

 `findProduct(Product,Supermarket,Section,Type).`

```
Product = breast,
Section = meat,
Supermarket = binDawood
Product = drumstick,
Section = meat,
Supermarket = binDawood
Product = breast,
Section = meat,
Supermarket = panda
Product = drumstick,
Section = meat,
Supermarket = alRaya
```

This is a query Find all chicken type information

 `findProduct(Product,Supermarket,bakery,Type).`

```
Product = chocolateCake,
Supermarket = binDawood,
Type = cake
Product = strawberryCake,
Supermarket = binDawood,
Type = cake
Product = toast,
Supermarket = binDawood,
Type = bread
Product = strawberryCake,
Supermarket = panda,
Type = cake
Product = toast,
Supermarket = panda,
Type = bread
Product = samoli,
Supermarket = panda,
Type = bread
Product = chocolateCake,
Supermarket = alRaya,
Type = cake
```

This is a query Find all Bakery Section information

 `findProduct(mirinda,Supermarket,Section,Type).`

```
Section = beverage,
Supermarket = binDawood,
Type = energyDrink
Section = beverage,
Supermarket = panda,
Type = energyDrink
Section = beverage,
Supermarket = danube,
Type = energyDrink
```

This is a query Find all Mirinda locations

 `findProduct(Product,alRaya,Section,Type).`

```
Product = yellowBanana,
Section = fruit,
Type = banana
Product = greenPapper,
Section = vegetable,
Type = pepper
Product = bigcarrot,
Section = vegetable,
Type = carrot
Product = steak,
Section = meat,
Type = beef
Product = roast,
Section = meat,
Type = beef
Product = drumstick,
Section = meat,
Type = chicken
Product = springRollPastry,
Section = frozenFood,
Type = pastry
```

This is a query Find all "Al-Raya" supermarket information

CONCLUSION

In conclusion, our project will make shopping at the supermarket more convenient.

Customers will save time and effort by being able to see all of the supermarkets in the system, explore all of the products in each supermarket, review all types of products in supermarkets, review all of the products available in each supermarket, and examine the available offers.

We hope that our project will be helpful to the customer and give him a better experience.

"Trust yourself.
You can do this."

- @reallygreatsite



TABLE OF

REFERENCES

Pinterest, Wattpad, Embarazo no planeado.	00
Pinterest, Pro-Light Design & Technology Ltd	02
https://swish.swi-prolog.org/ ("about", Definition of swish)	04
AI fundaments-JU-LAB-01: Introduction to Prolog	04
AI fundaments-JU-labII_expert system (Swish)	05
http://www.ablmcc.edu.hk/~scy/prolog/proo2.htm	08
Pinterest, DIE KLEINE BÄCKEREI FLOCK	21
