GROCERY APP

GROCERY APP

TITLE : To build an grocery app by using android.
SOFTWARE REQUIRED : android studio.2021.2.1 and some other tools.

INTRODUCTION :

An online grocer is either a brick-and-mortar supermarket or grocery store that allows online ordering, or a standalone e-commerce service that includes grocery items.There is usually a delivery charge for this service. Brick-and-mortar supermarkets that have built internet channels to better service their clients are known as online grocers. Online grocery delivery services are available throughout Europe, Asia and North America, mostly in urban centers. The online ordering is done through e-commerce websites or mobile apps.

The COVID-19 pandemic greatly accelerated the growth of online grocers, and in the first few months of the pandemics online grocery shopping increased by 300%. In addition, first-time online grocery shoppers accounted for 41 percent of online grocery shoppers.The epidemic of COVID-19 has hastened the uptake of online grocery shopping. Pre-COVID-19 food shopping activity accounted for 9 percent of the market, but 63 percent of consumers worldwide purchased more groceries online after the outbreak than they did before they were socially isolated.

Most local online grocers have their own drivers. The most common type of personal delivery involves storing grocery inventory in a warehouse to deliver to customers once orders are placed. Another type of personal delivery which is less common is based on just-in-time business in which there is no warehouse or inventory. In this type of delivery, customers place orders for next-day delivery. The online grocer shops for the groceries on the morning of the delivery day.

Some grocery fulfillment centers are set up as dark stores. Online-only grocers typically have warehouses or distribution centers nearby, to allow local shipping of refrigerated items.

Online grocers with a large regional or national delivery area may ship groceries using courier services. If the order contains cold or frozen items, this involves "flash freezing" the goods and pack them into special shipping containers.

Companies have experimented with automated delivery modes including drone delivery and robots. For instance, in Fall 2016 Washington, D.C. approved a trial run of rolling delivery drones produced by Star ship Technologies. The earthbound robots are similar to wheeled

coolers and carry around 40 pounds of groceries.

Online grocery stores may allow facilitating local food which may reduce the environmental impact of food transport. Small-scale farmers have been embracing digital technologies as a way to sell produce directly, and community-supported agriculture and direct-sell delivery systems are on the rise during the coronavirus pandemic.Furthermore, weekly grocery deliveries can be a better choice than individual trips to a store.

STEPS FOR THE PROJECT:

STEP-1:Create a New Project

To create a new project in Android Studio i refer to How to Create/Start a New Project in Android Studio. Note that select **Java** as the programming language.

STEP-2 : Add dependency of Slider View in build.gradle file

Navigate to the Gradle scripts and then to build.gradle(Module) level. Add below line in build.gradle file in the dependencies section.

CODE:

```
// dependency for slider view
implementation 'com.github.smarteist:autoimageslider:1.3.9'
// dependency for loading image from url
implementation "com.github.bumptech.glide:glide:4.11.0"
```

STEP-3 : Add internet permission in the AndroidManifest.xml file

Navigate to the **app > Manifest** to open the Manifest file and add below two lines.

CODE:

<!–Permission for internet–>

<uses-permission android:name=”android.permission.INTERNET” />

<uses-permission android:name=”android.permission.ACCESS_NETWORK_STATE” />

STEP-4 : Working with the activity_main.xml file

Go to the **activity_main.xml** file and refer to the following code. Below is the code for the **activity_main.xml** file.

CODE FOR XML :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xml:android="http://schemas.android.com/apk/res/android"
    xml:app="http://schemas.android.com/apk/res-auto"
    xml:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!--
    slider animation duration is to set duration for transition between two slides
```

sliderautocycledirection is to set animation between transition of your slides

slider indicator enables is used to display the indicators for slider

slider indicator gravity is to set gravity for indicator gravity

slider indicator margin is to set margin for indicator

slider indicator orientation is used to add orientation for slider

slider indicator padding is use to add padding to indicator

slider indicator selected color is use to specify selected color

and slider indicator unselected color is use to specify the color when the slider is unselected

slider scroll time in sec is used to specify scrolling time in seconds

-->

```
<com.smarteist.autoimageslider.SliderView
        android:id="@+id/slider"
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:layout_centerInParent="true"
        app:sliderAnimationDuration="600"
        app:sliderAutoCycleDirection="back_and_forth"
        app:sliderIndicatorAnimationDuration="600"
        app:sliderIndicatorEnabled="true"
        app:sliderIndicatorGravity="center_horizontal|bottom"
        app:sliderIndicatorMargin="15dp"
        app:sliderIndicatorOrientation="horizontal"
        app:sliderIndicatorPadding="3dp"
        app:sliderIndicatorRadius="2dp"
```

```
        app:sliderIndicatorSelectedColor="#5A5A5A"
        app:sliderIndicatorUnselectedColor="#FFF"
        app:sliderScrollTimeInSec="1" />

</RelativeLayout>
```

STEP-5 : Create a new Modal class for storing data

Navigate to app > java > your app's package name and then right-click on it and New > Java class and name your Model class as **SliderData** and below code inside that Java class. Below is the code for the **SliderData.java** file. Comments are added inside the code to understand the code in more detail.

CODE FOR JAVA :

```
public class SliderData {


    // image url is used to
    public void setImgUrl(String imgUrl) {
        this.imgUrl = imgUrl;
    }
}
```

STEP - 6: Create an XML file for the items of SliderView

Navigate to the app > res > layout > Right-click on it and select New > Layout Resource File and then name your XML file as **slider_layout.xml**. After creating

this file add the below code to it.

CODE FOR XML :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
        xml:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <!--Image we will display is our slider view-->
        <ImageView
                android:id="@+id/my image"
                android:layout_width="400dp"
                android:layout_height="300dp"
                android:layout_centerHorizontal="true"
                android:contentDescription="@string/app_name" />

</RelativeLayout>
```

STEP - 7: Create Adapter Class for setting data to each item of our SliderView

Navigate to app > java > your app's package name and then right-click on it and New > Java class and name your class as **SliderAdapter** and below code inside that Java class. Below is the code for the **SliderAdapter.java** file. Comments are added inside the code to understand the code in more detail.

CODE FOR JAVA :

```
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

import com.bumptech.glide.Glide;
import com.smarteist.autoimageslider.SliderViewAdapter;

import java.util.ArrayList;
import java.util.List;

public class SliderAdapter extends
SliderViewAdapter<SliderAdapter.SliderAdapterViewHolder> {

    // list for storing urls of images.
    private final List<SliderData> mSliderItems;

    // Constructor
    public SliderAdapter(Context context, ArrayList<SliderData>
sliderDataArrayList) {
        this.mSliderItems = sliderDataArrayList;
    }
```

```
        // We are inflating the slider_layout
        // inside on Create View Holder method.
        @Override
        public SliderAdapterViewHolder onCreateViewHolder(ViewGroup parent) {
                View inflate =
LayoutInflater.from(parent.getContext()).inflate(R.layout.slider_layout, null);
                return new SliderAdapterViewHolder(inflate);
        }

        // Inside on bind view holder we will
        // set data to item of Slider View.
        @Override
        public void onBindViewHolder(SliderAdapterViewHolder viewHolder, final
int position) {

                final SliderData sliderItem = mSliderItems.get(position);

                // Glide is use to load image
                // from url in your image view.
                Glide.with(viewHolder.itemView)
                                .load(sliderItem.getImgUrl())
                                .fitCenter()
                                .into(viewHolder.imageViewBackground);
        }

        // this method will return
```

```
        // the count of our list.
        @Override
        public int getCount() {
                return mSliderItems.size();
        }

        static class SliderAdapterViewHolder extends
SliderViewAdapter.ViewHolder {
                // Adapter class for initializing
                // the views of our slider view.
                View itemView;
                ImageView imageViewBackground;

                public SliderAdapterViewHolder(View itemView) {
                        super(itemView);
                        imageViewBackground =
itemView.findViewById(R.id.myimage);
                        this.itemView = itemView;
                }
        }
}
```

STEP-8: Working with the MainActivity.java file

Go to the **MainActivity.java** file and refer to the following code. Below is the code for the **MainActivity.java** file. Comments are added inside the code to

understand the code in more detail.

CODE FOR JAVA :

```
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import com.smarteist.autoimageslider.SliderView;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    // Urls of our images.
    String url1 = "https://www.geeksforgeeks.org/wp-content/uploads/gfg_200X200-1.png";
    String url2 = "https://qphs.fs.quoracdn.net/main-qimg-8e203d34a6a56345f86f1a92570557ba.webp";
    String url3 = "https://bizzbucket.co/wp-content/uploads/2020/08/Life-in-The-Metro-Blog-Title-22.png";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // we are creating array list for storing our image urls.
        ArrayList<SliderData> sliderDataArrayList = new ArrayList<>();
```

```
        // initializing the slider view.
        SliderView sliderView = findViewById(R.id.slider);

        // adding the urls inside array list
        sliderDataArrayList.add(new SliderData(url1));
        sliderDataArrayList.add(new SliderData(url2));
        sliderDataArrayList.add(new SliderData(url3));

        // passing this array list inside our adapter class.
        SliderAdapter adapter = new SliderAdapter(this,
sliderDataArrayList);

        // below method is used to set auto cycle direction in left to
        // right direction you can change according to requirement.

sliderView.setAutoCycleDirection(SliderView.LAYOUT_DIRECTION_LTR);

        // below method is used to
        // set adapter to slider-view.
        sliderView.setSliderAdapter(adapter);

        // below method is use to set
        // scroll time in seconds.
        sliderView.setScrollTimeInSec(3);

        // to set it scrollable automatically
```

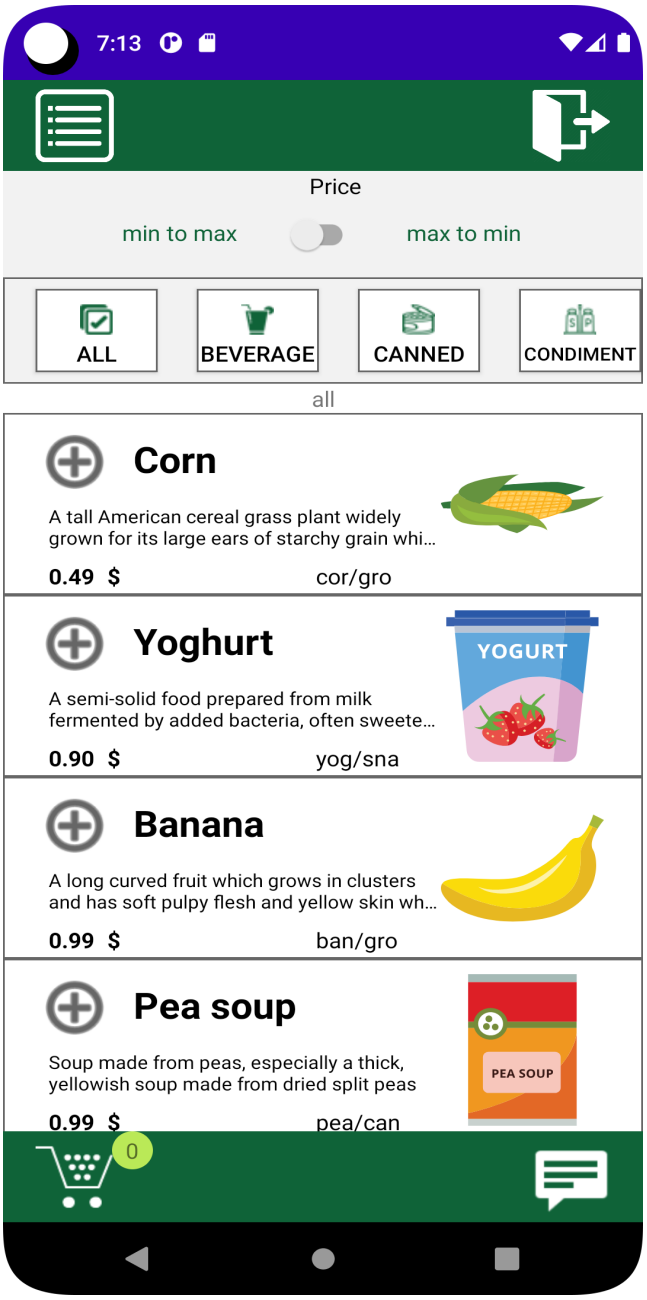
```
            // we use below method.
            sliderView.setAutoCycle(true);

            // to start auto cycle below method is used.
            sliderView.startAutoCycle();
        }
}
```

OUTPUT :

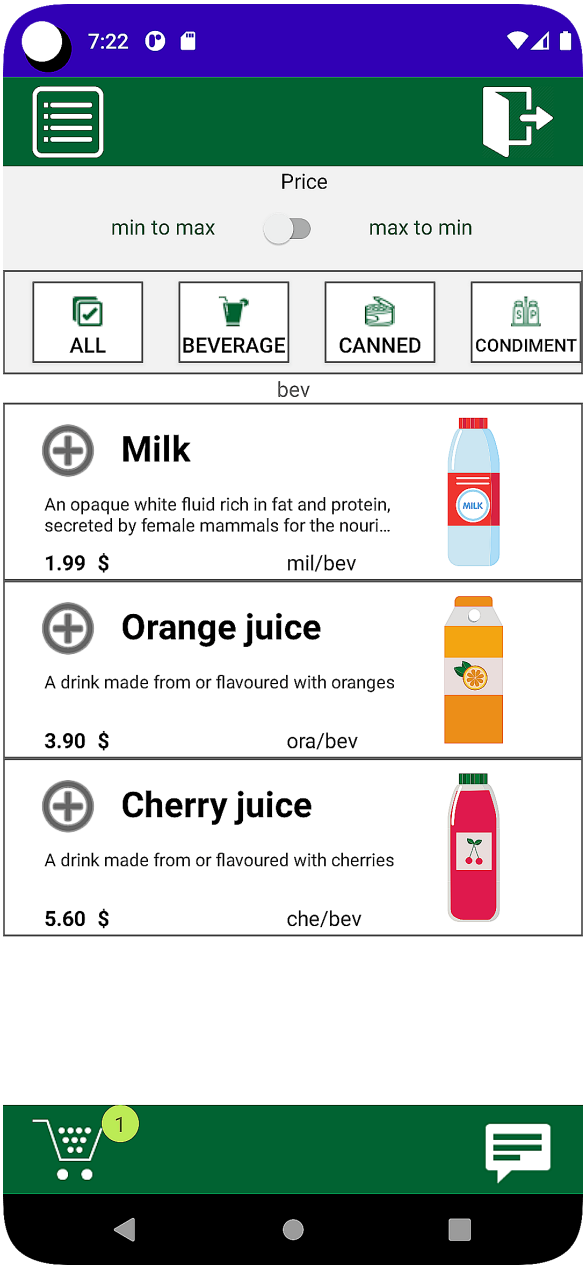
1.By opening app .if you have an account give your mail and password.

if you don't have an account click on sign up.

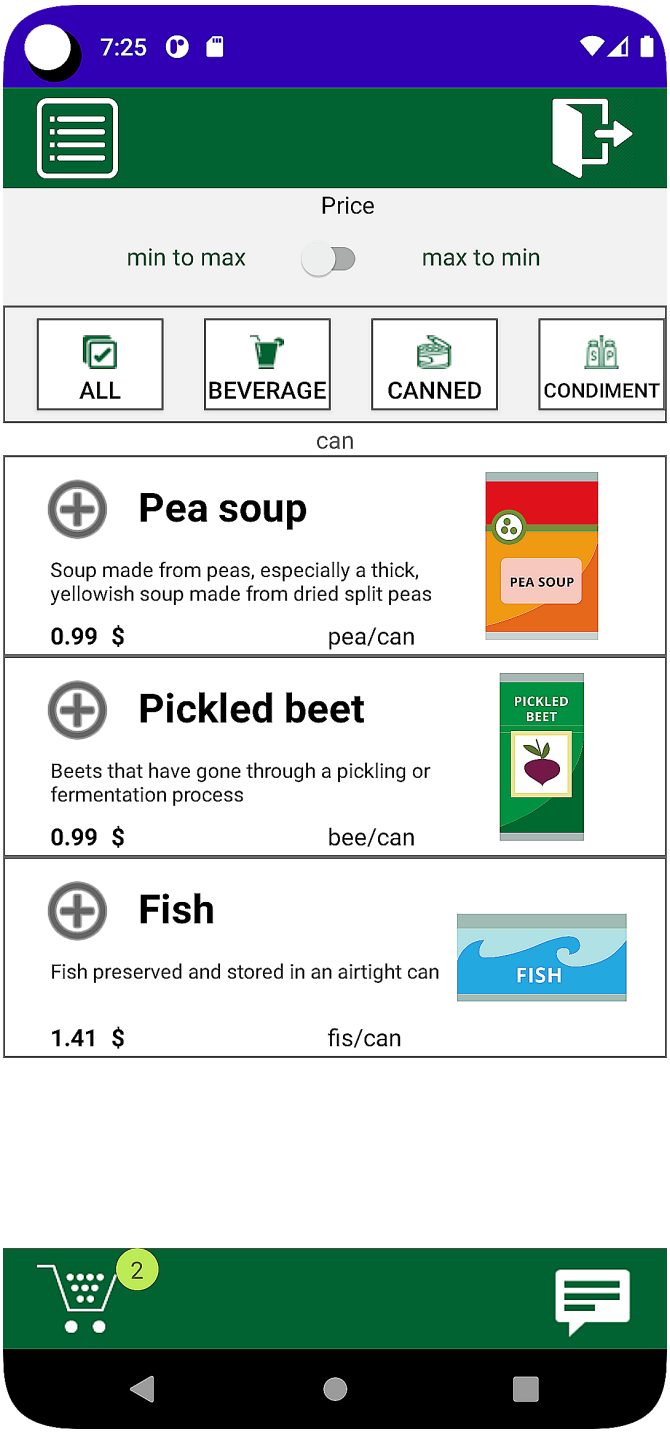
2. Then the app will be



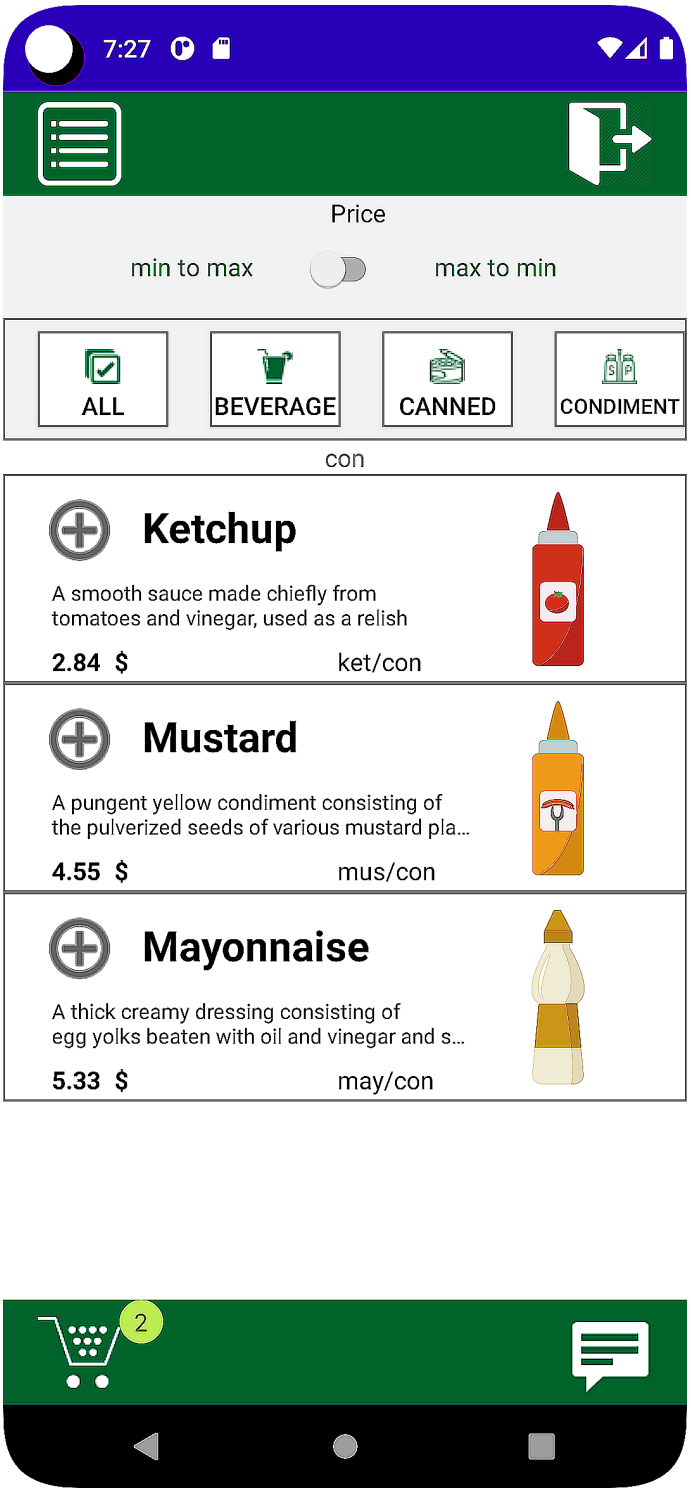
3. if we want Beverage then click on beverage .



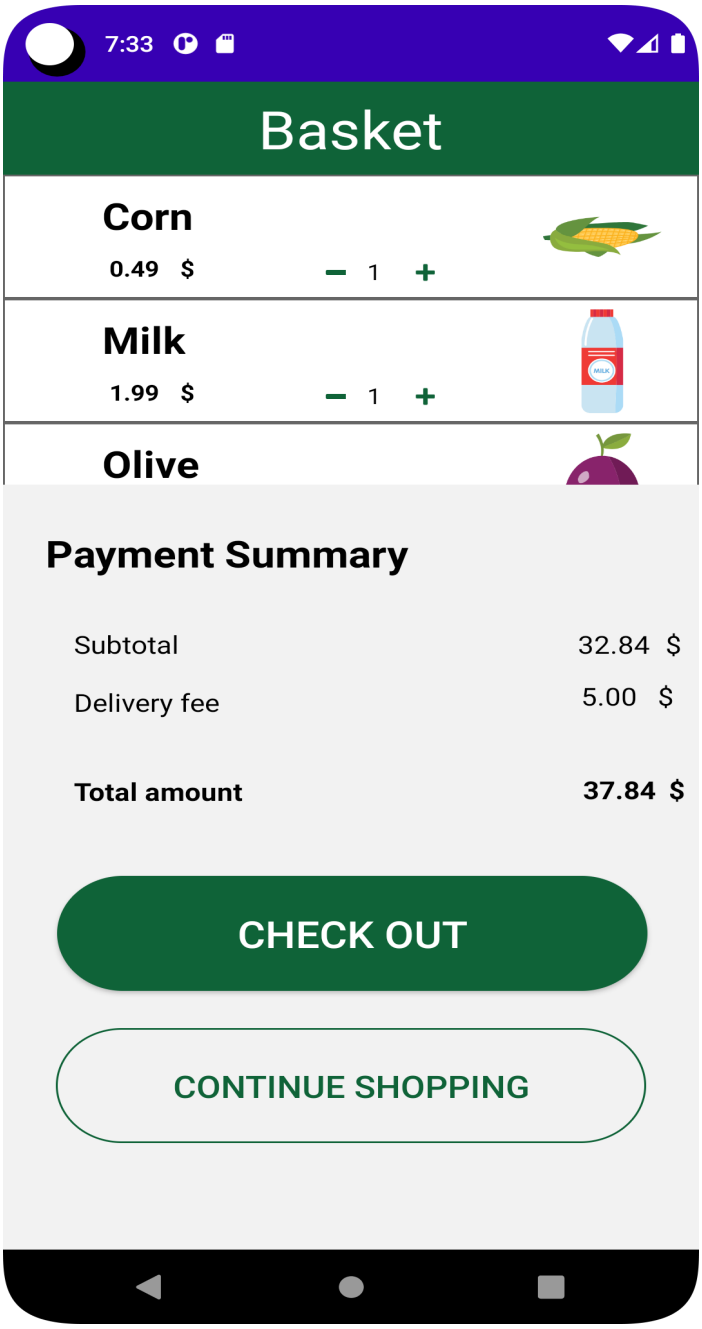
4. Alternatively, if we want canned then click on canned then the app will be



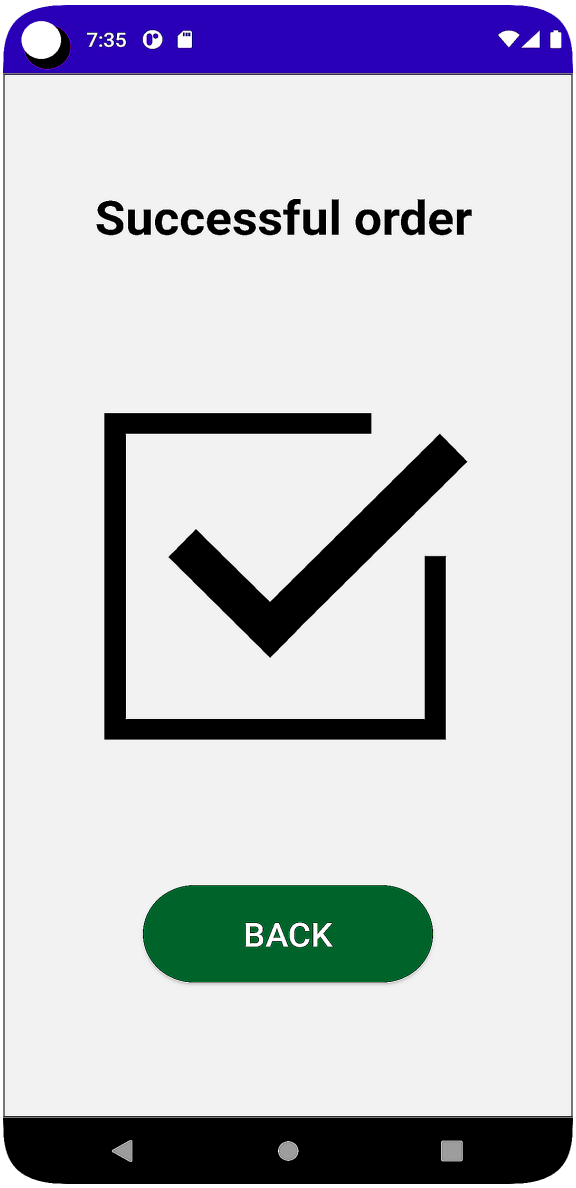
5. Alternatively, if we want condiment then click on condiment



6. select the item we want then the payment will be



7. if we place order then


7:35
Successful order
BACK