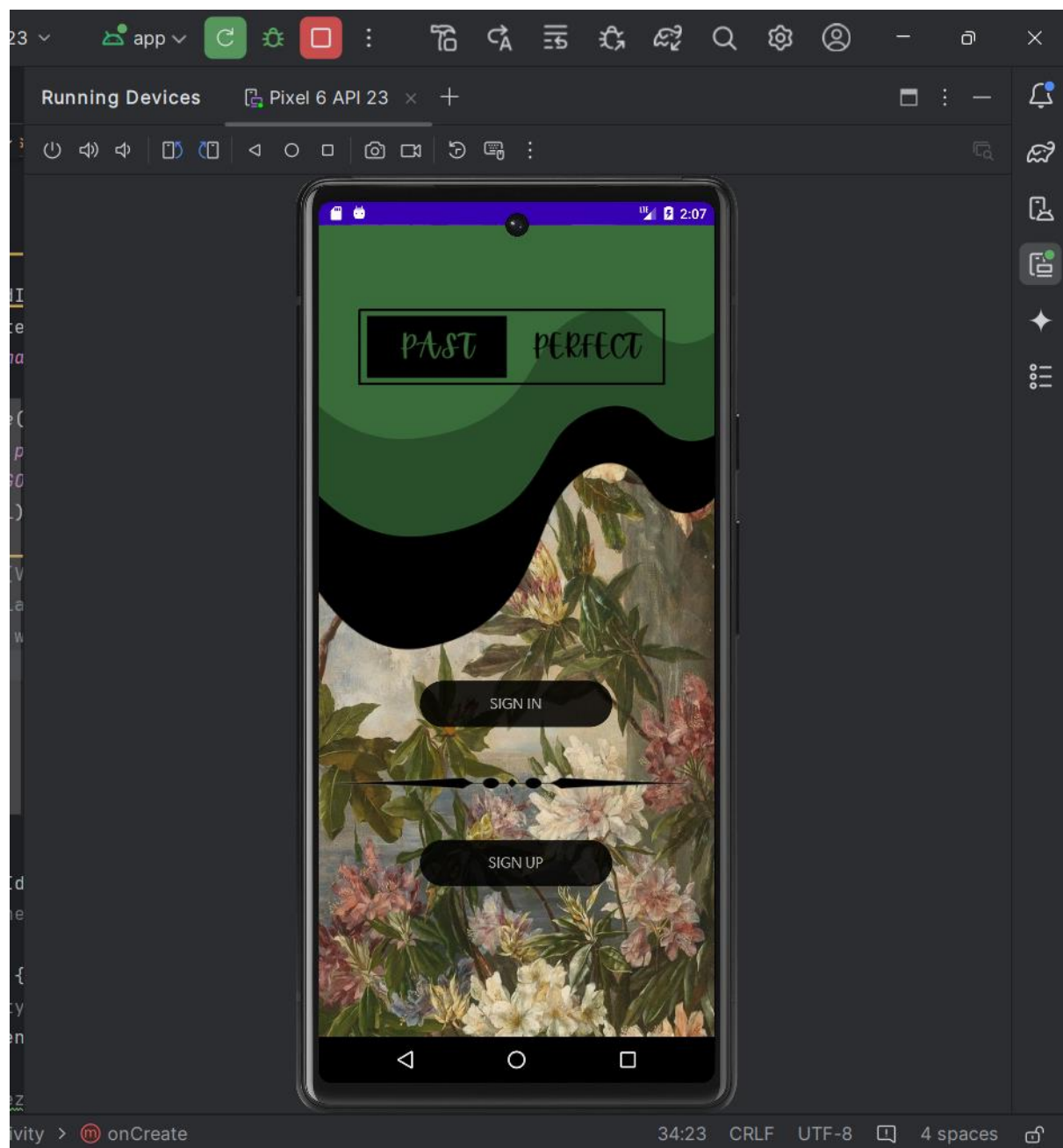


Raport Intermediar
-proiect III-

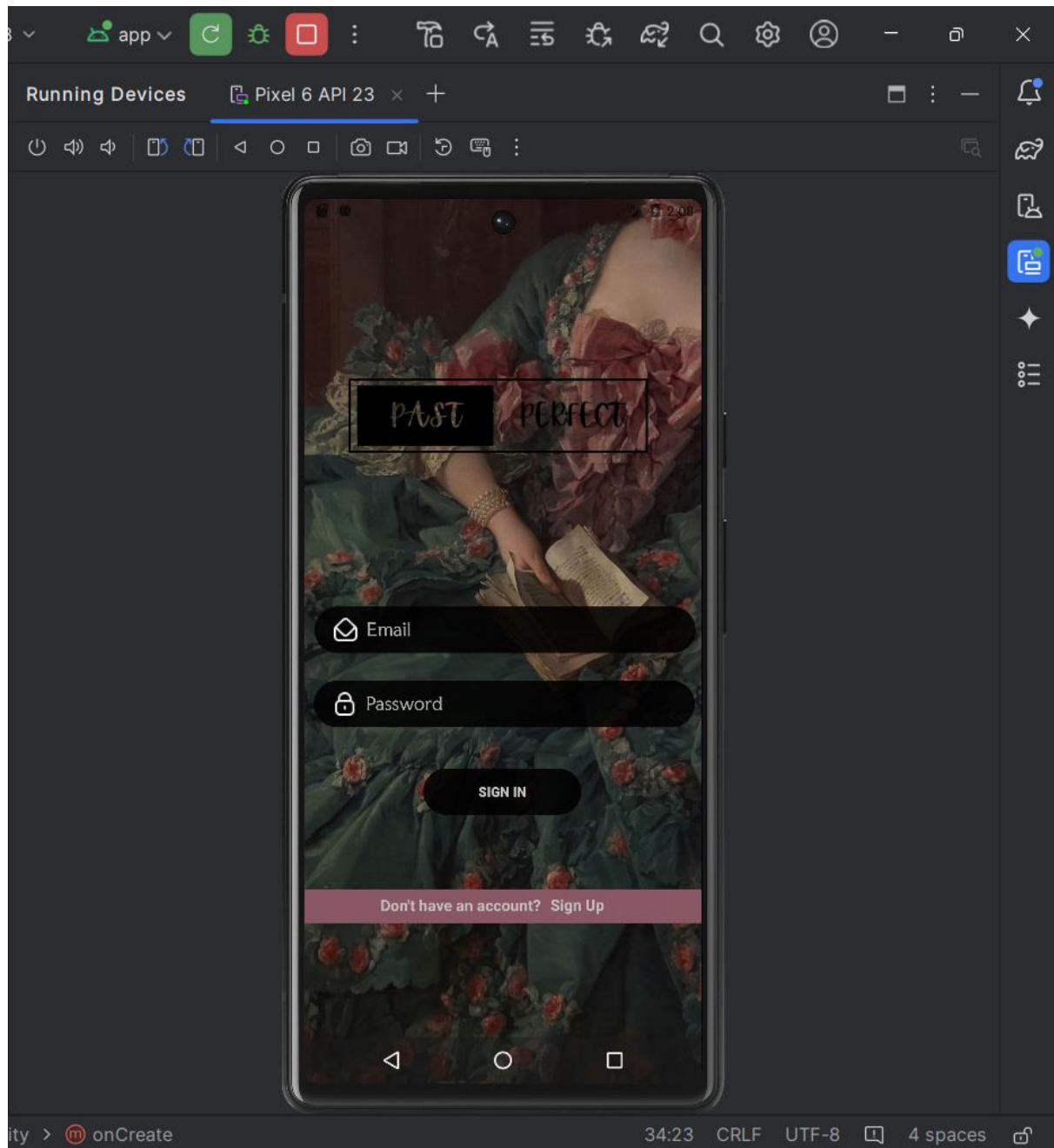
Safta Andreea CR4S2

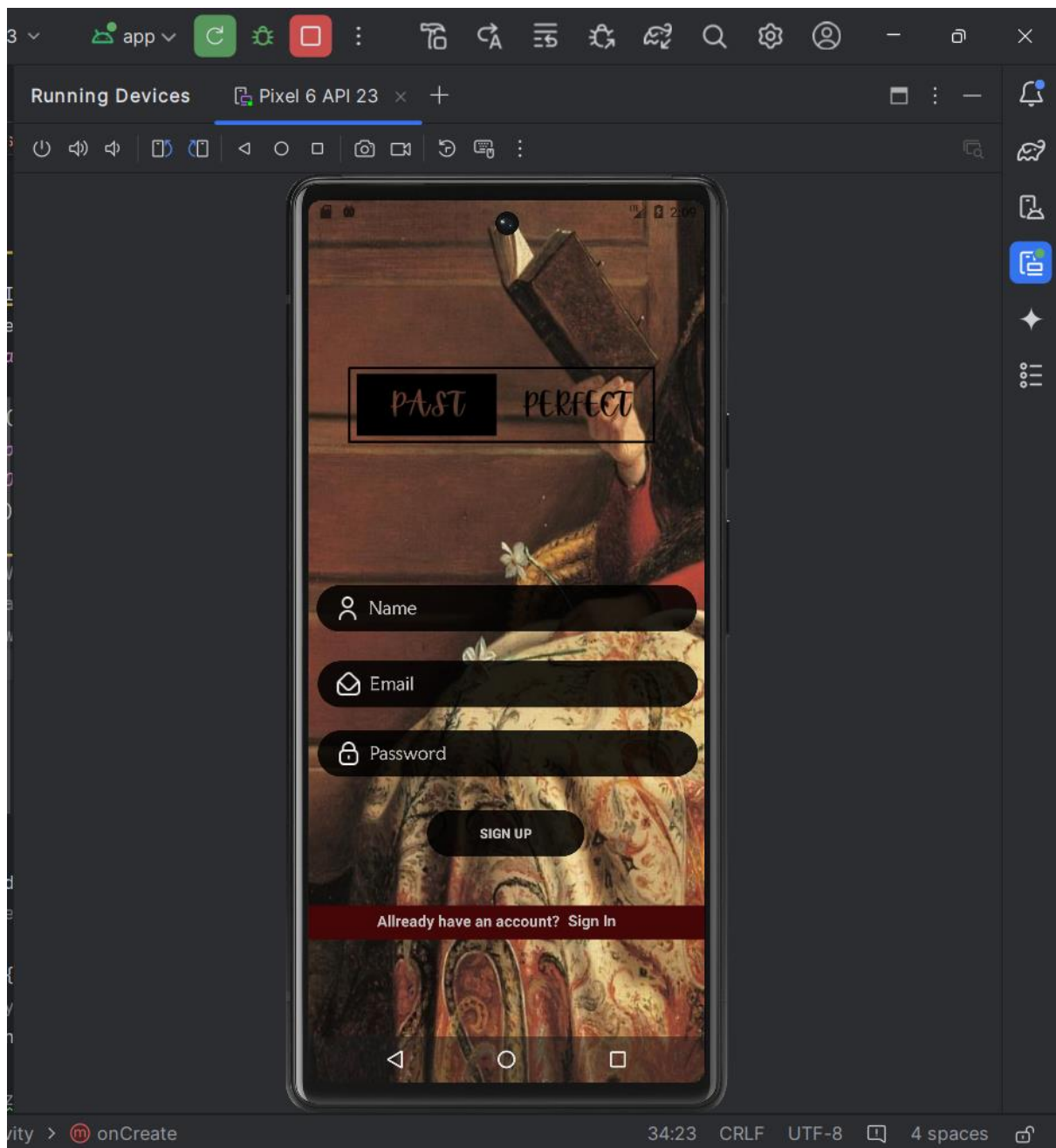
Aplicatia are si ca scop oferirea unei platforme destinate unui magazine vintage .Functionalitatile implementate ale acestei prezentari sunt cele de autentificare ,logare , vizualizare categorii si produse , adaugare si stergere produse cu ajutorul firebase firestore , adaugarea de produse in cos si vizualizarea continutului cosului de cumparaturi .

Pentru realizarea partii de autentificare si inregistrare am folosit firebase authentication pentru a putea stoca si salva datele introduse de catre utilizatori . Inainte ca utilizatorii sa poata vizualiza continutul aplicatiei pagina de pornire va aparea pentru a se putea alege metoda de autentificare sau inregistrare .

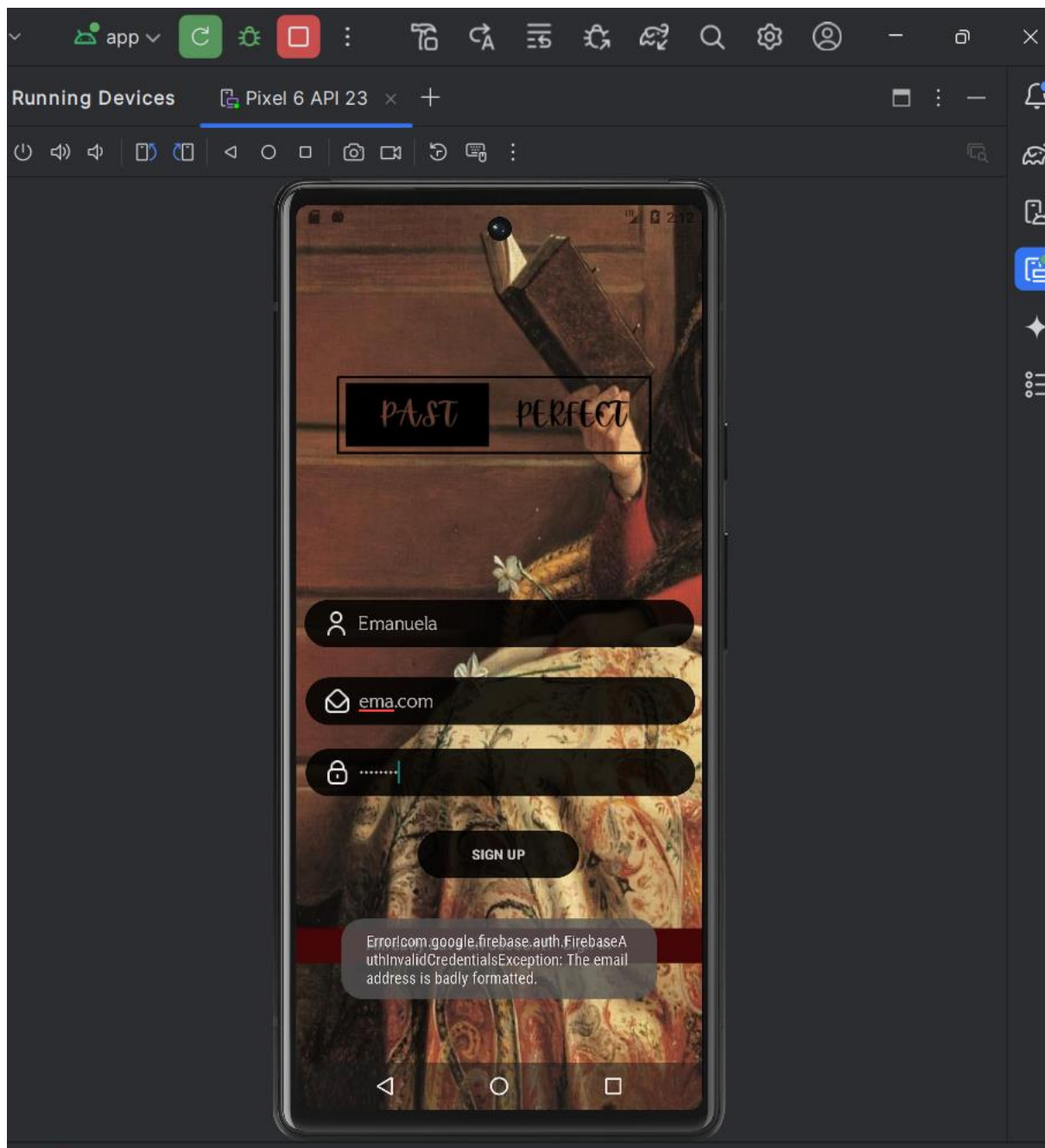


Paginile de Sign In si Sign Up sunt urmatoarele :

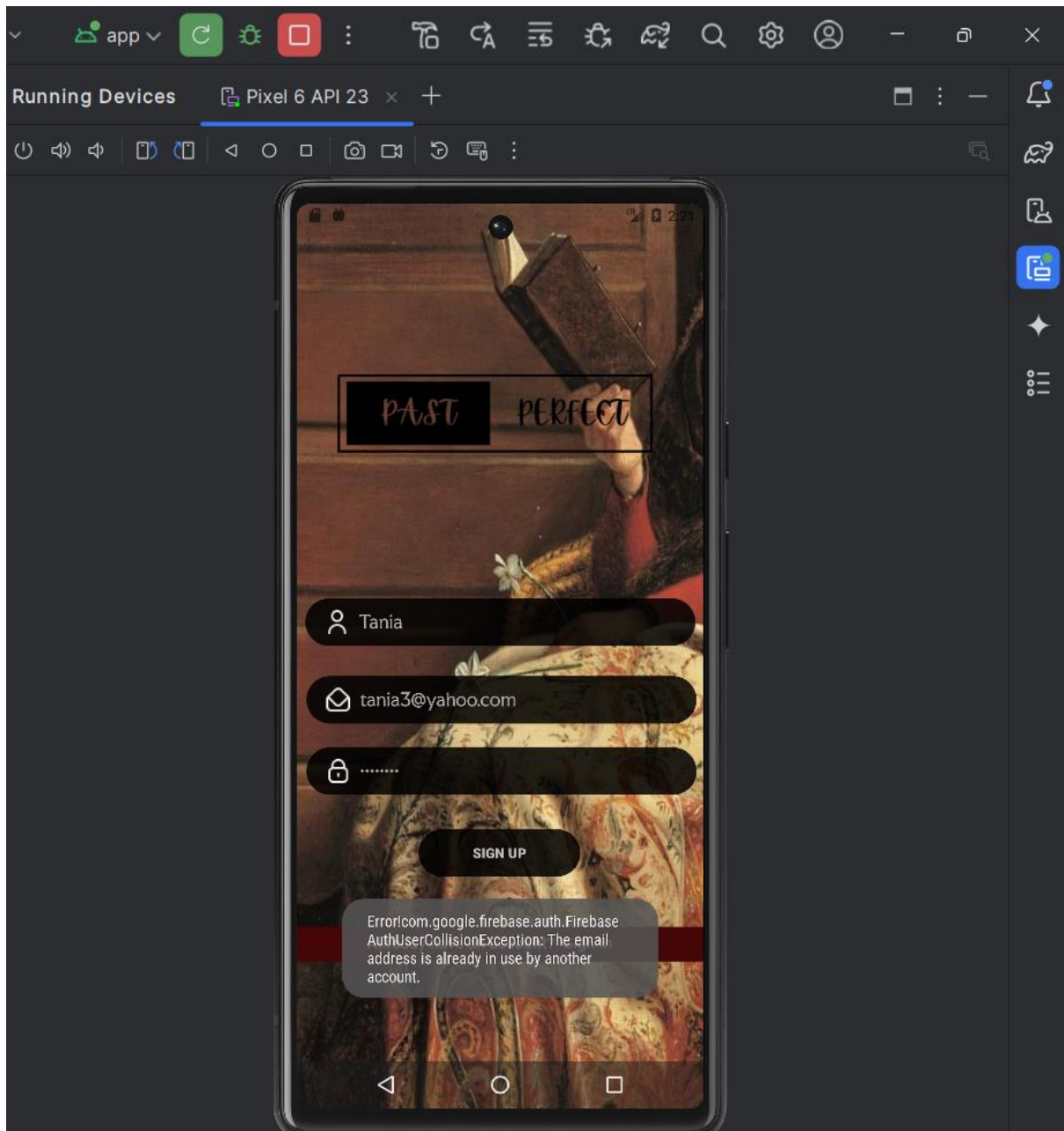




In momentul cand un utilizator doreste sa isi creeze un cont , acesta va introduce datele corespunzatoare . Daca fomatul nu este respectat , de exemplu in cadrul email-ului , daca acesta nu respecta formatul standard de email va aparea un mesaj de eroare :



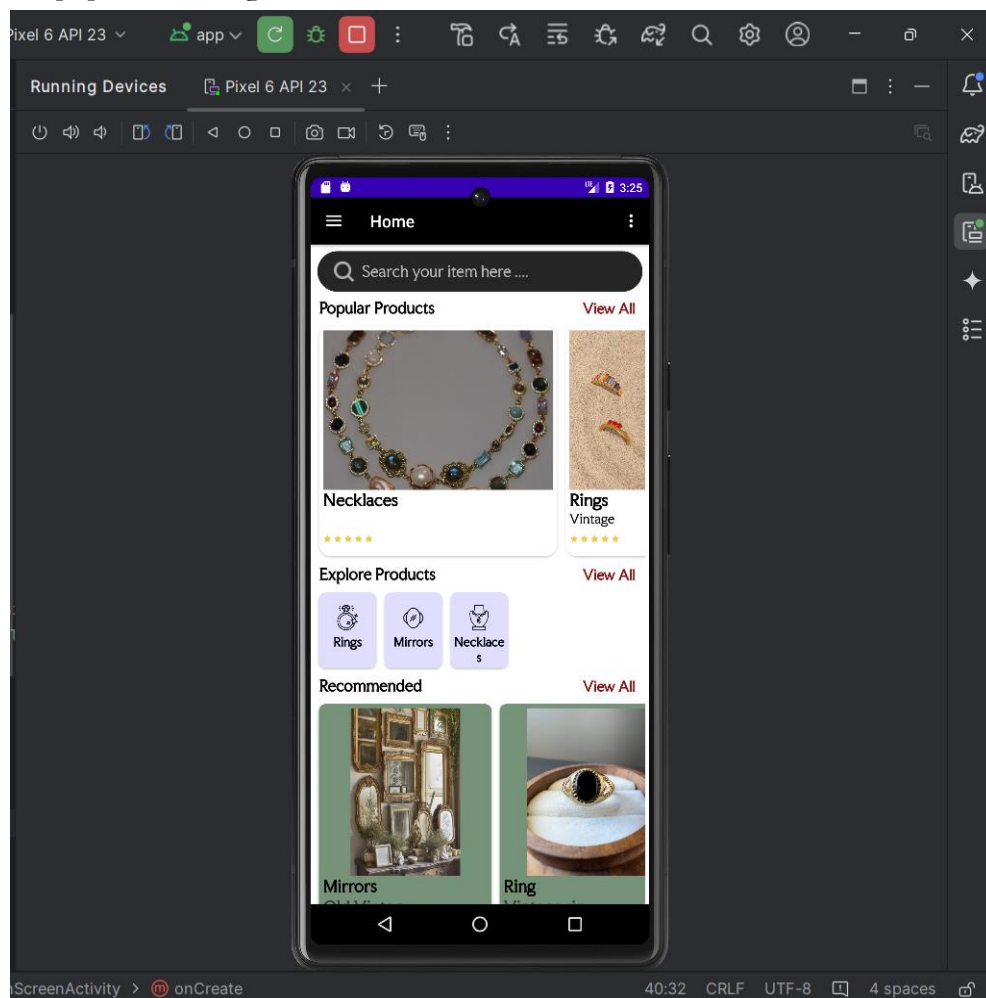
Mesajul informeaza utilizatorul ca adresa de email nu este bine formatata si necesita modificarile necesare incadrarii in tiparele standard . Dupa modificarile necesare va aparea un mesaj de informare "Registration successful!" . Daca cumva un utilizator doreste sa isi faca un alt cont dar cu acelasi mail acest lucru nu se va putea realiza pentru ca in firebase mail-ul respective este deja inregistrat si un mesaj de eroare va aparea .



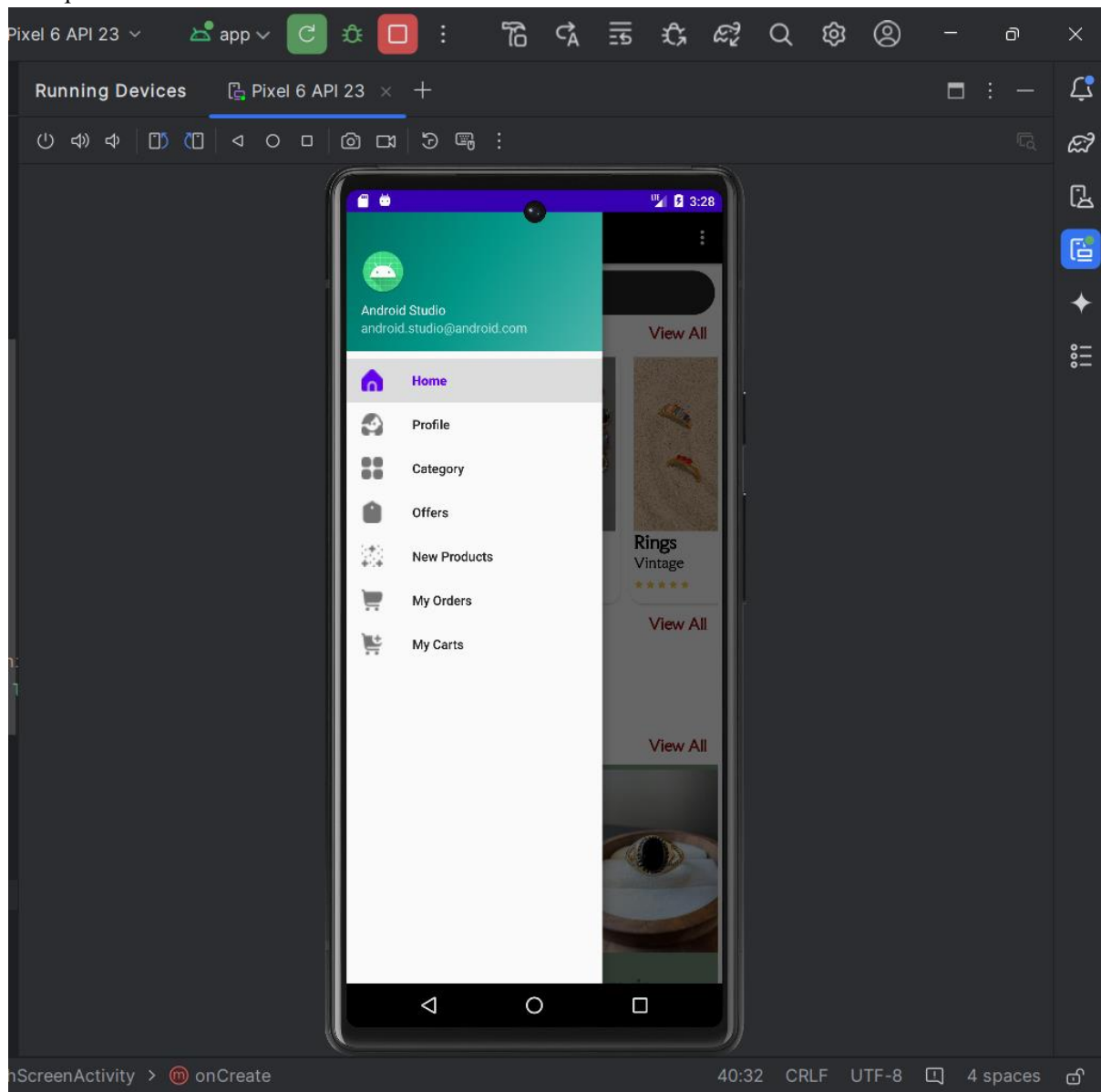
Necompletarea numelui , email-ului si a parolei vor fi atentionate prin mesaje specifice , de asemenea parola are conditii care trebuie indeplinite .


```
VS VintageShop Version control Pixel 6 API 23 app
NavCategoryDetailedAdapter.java SplashScreenActivity.java RegistrationActivity.java
28 public class RegistrationActivity extends AppCompatActivity {
86 private void createUser() {
87
88     String userName = name.getText().toString();
89     String userEmail = email.getText().toString();
90     String userPassword = password.getText().toString();
91
92     if(TextUtils.isEmpty(userName))
93     {
94         Toast.makeText(context, this, text: "Name is empty!", Toast.LENGTH_SHORT).show();
95         return;
96     }
97
98     if(TextUtils.isEmpty(userEmail))
99     {
100         Toast.makeText(context, this, text: "Email is empty!", Toast.LENGTH_SHORT).show();
101         return;
102     }
103
104     if(TextUtils.isEmpty(userPassword))
105     {
106         Toast.makeText(context, this, text: "Password is empty!", Toast.LENGTH_SHORT).show();
107         return;
108     }
109
110     if(userPassword.length() < 6)
111     {
112         Toast.makeText(context, this, text: "Password must be greater than 6 letters!", Toast.LENGTH_SHORT).show();
113         return;
114     }
115
116     //create user
117 }
```

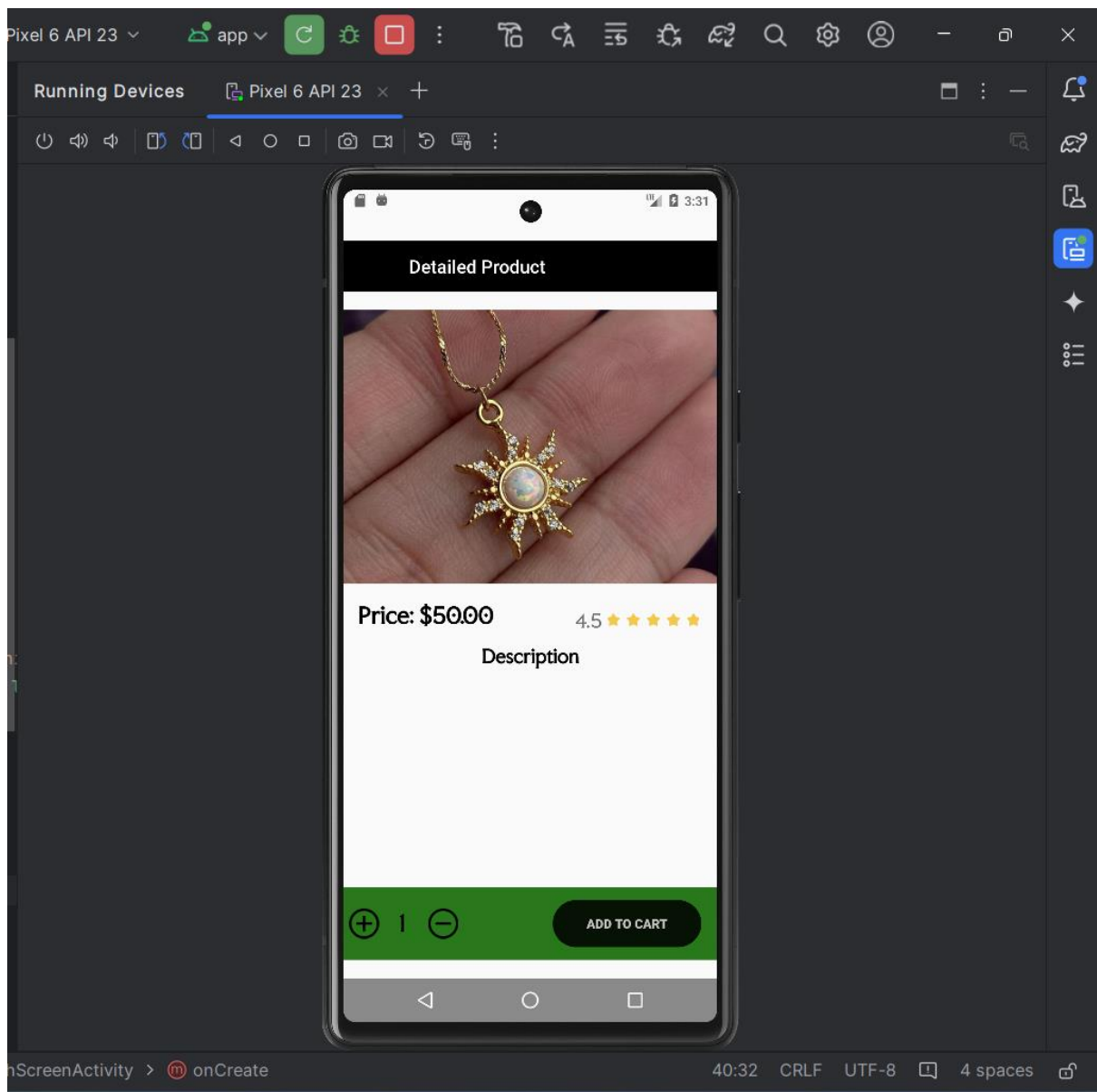
Dupa conectarea cu succes utilizatorul va fi redistribuit catre home page unde se pot vedea Produse populare , categoriile si recomandarile.



În partea din stanga sus avem meniul de unde putem vedea mult mai multe functionalitati ale aplicatiei .

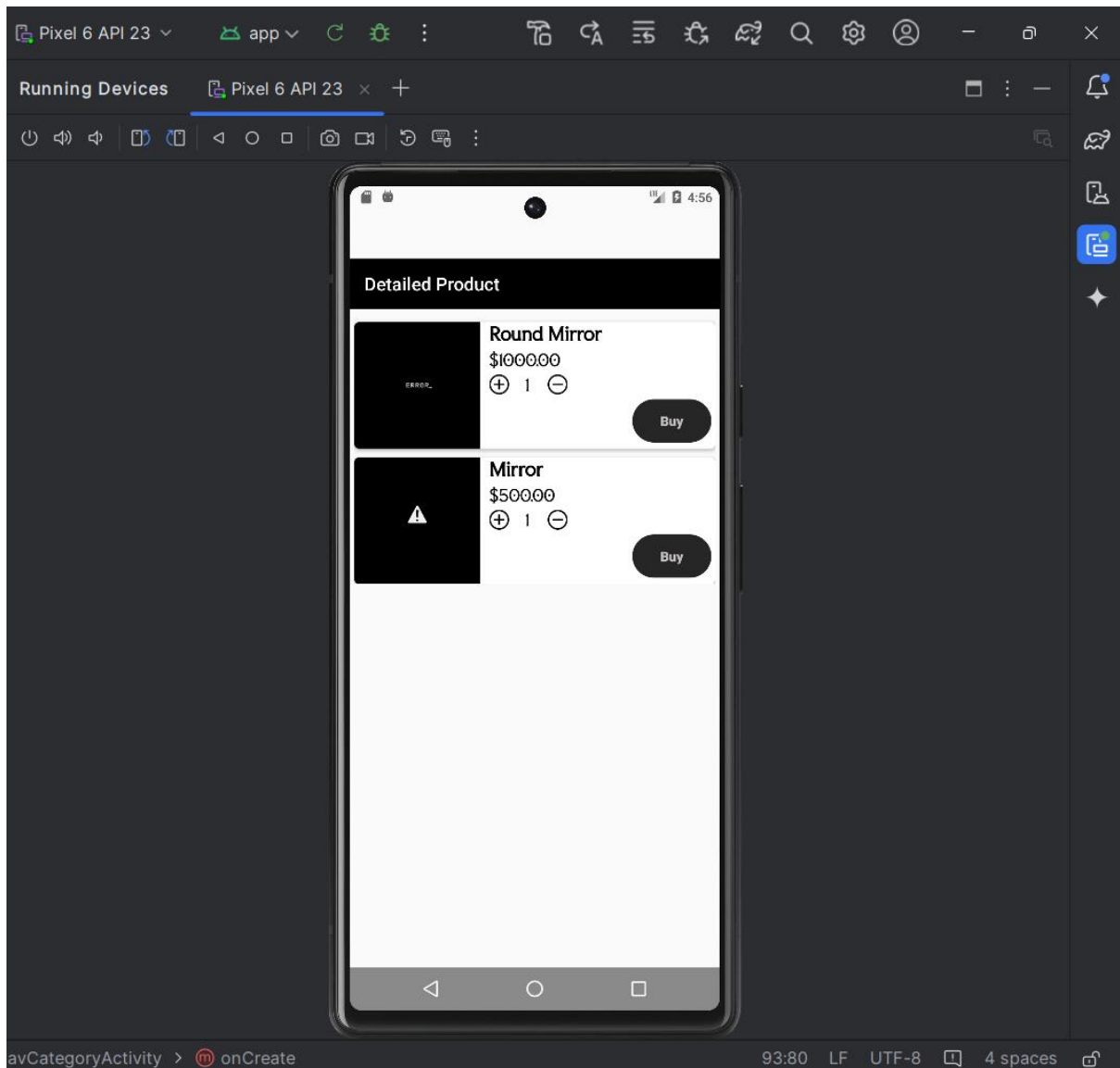


Funcția de adăugare în cos este posibilă, astfel încât dacă se dorește a cumpăra un produs, utilizatorul poate să vizualizeze poza produsului, descrierea acestuia, prețul și rating-ul.

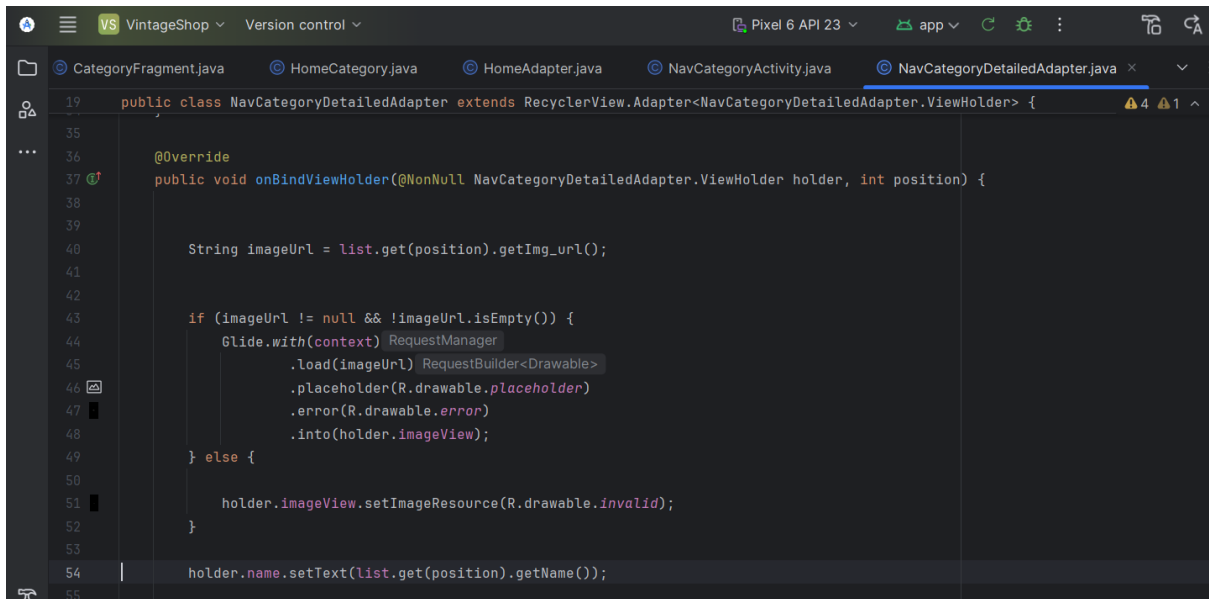


Odată adăugat în cos, un mesaj de confirmare va apărea și dacă se dorește vizualizarea curentă a cosului de cumpărături, utilizatorul poate vedea prețul final, numele produselor, prețul lor, cantitatea, data și ora când acestea s-au adăugat.

In sectiunea de Home -> Ctegrory se pot vedea categoriile de produse pe care aplicatia le ofera . In cazul in care produsele anumitor categorii nu au o poza incarcata o poza de default va fi afisata pentru a putea anunta utilizatorii ca produsul respective nu prezinta o poza specifica . De asemenea daca o poza a unui produs este incarcata gresit o poza de atentionare se va afisa pentru a arata ca s-a produs o eroare .



Codul specific :



```
19 public class NavCategoryDetailedAdapter extends RecyclerView.Adapter<NavCategoryDetailedAdapter.ViewHolder> {
20
21     ...
22
23     @Override
24     public void onBindViewHolder(@NonNull NavCategoryDetailedAdapter.ViewHolder holder, int position) {
25
26         String imageUrl = list.get(position).getImg_url();
27
28         if (imageUrl != null && !imageUrl.isEmpty()) {
29             Glide.with(context).requestManager
30                 .load(imageUrl).requestBuilder<Drawable>
31                 .placeholder(R.drawable.placeholder)
32                 .error(R.drawable.error)
33                 .into(holder.imageView);
34         } else {
35             holder.imageView.setImageResource(R.drawable.invalid);
36         }
37
38         holder.name.setText(list.get(position).getName());
39     }
40 }
```