



## **ONLINE SHOPPING**

*A Report for the Evaluation 3 of Project 2*

*Submitted by*

**ADITYA BAGHEL**

**(1713104056)**

Baghelsingh200aditya@gmail.com

*in partial fulfilment for the award of the degree of*

**BACHELOR OF COMPUTER APPLICATIONS**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**Under the Supervision of**

**Mr SANCHIT SAPRA**

**Assistant Professor**

**APRIL / MAY- 2020**

## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
1.	Abstract	1
2.	Introduction	1
3.	Existing System	2
4.	Proposed system	2
5.	Architecture diagrams	3
6.	Code	4
6.	Screenshots	11
7.	Conclusion/Future Enhancement	14
8.	References	14

## **ABSTRACT**

Today the web and its boom have created a replacement economic scenario that not only stresses on the classical concept of the merchandise but also on the fashionable concept of service. It is this level of service that dictates whether a commercial venture will succeed or not in the market. To provide a high accessibility of service we will design the online shopping website, so that potential customers need not go to a physical shop to buy products or services.

## **INTRODUCTION**

It is a web-based project which is made for remote-shopping or shopping through the Internet. As technology is being advanced the way of life is changing accordingly. Now a day's we can place the order for anything from our home. There is no need to go to the shop of the things we want. The order can be placed online through the Internet. The payment, the confirmation of purchasing; we will do everything we would like. Now we will think that how the times are changed with time. People had to face in rows to attend their terms to shop for a specific thing from a well-liked shop. But what's happening now a day's we will extremely surprise that those things are often available on the door-step in few hours.

## **EXISTING SYSTEM**

The present scenario for online shopping in rural areas is that it is still not as developed market chain as that is in urban areas. In urban areas, people get their delivery within a few days. But in rural areas availability of the internet is a bigger problem and because of poor delivery chains in rural areas, people there cannot use these online shopping websites as easily as in the city.

The payment gateway for online shopping relies on COD, net banking, credit card, debit card, and they are in city areas. There is an EMI option on a few selected items in online shopping.

## **PROPOSED SYSTEM**

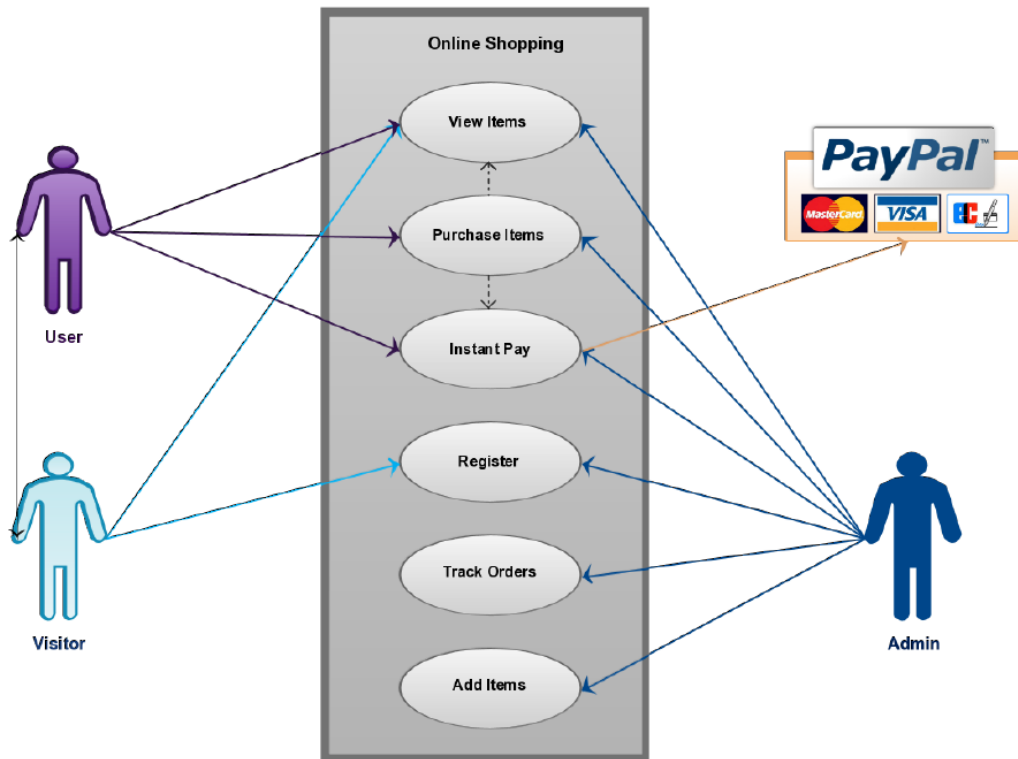
Using our application users can interact with more than one category to shop from.

There are many categories like footwear, clothing, electronic appliances, mobile phones and the feature that make our system different is that customers can use our application to order food from restaurants in their locality. We have an affordable EMI option for every category of products except for the food.

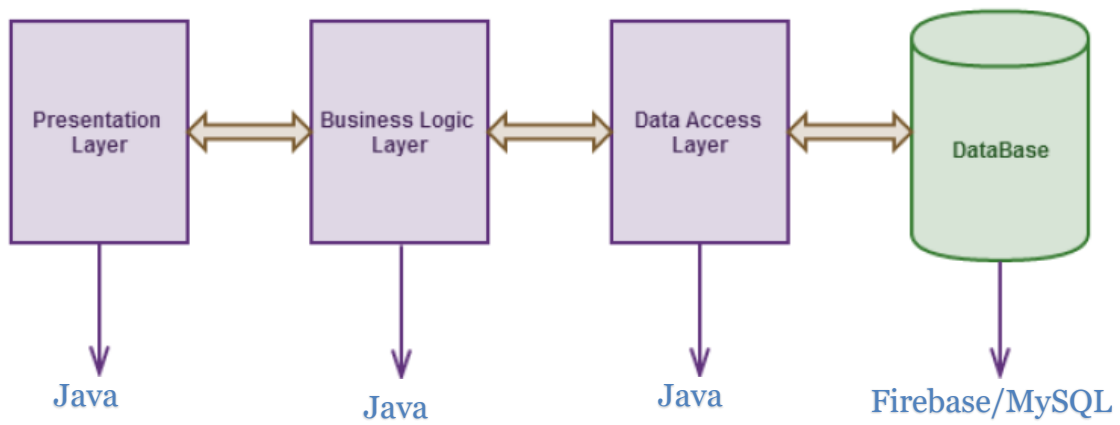
By using our application, the user need not go to different websites to shop and order food. Our application can have multiple administrators like a local shop owner can add their products online and the restaurant owner can also add/delete the menu from the application.

With affordable rates than other websites and affordable prices and easy EMI options made our application more attractive than other applications.

## USE-CASE DIAGRAM



## 3 LAYER ARCHITECTURE



## Project Detail

We have Java programming language to make GUI by which customers, administrator and viewer can interact with.

In the back-end, Firebase and MySQL is used. In case user is not connected with internet still, they can see their previous orders offline but to order anything they have to connect to internet.

Admin : They are those users who can update, view any products from the application.

User : Those people who use our application to purchase products.

Visitors : Those people who only view what is in our application but doesn't purchase anything.

## Implementation / Code

#To check availability of Internet

```
public class CheckInternetConnection {
    Context ctx;
    public CheckInternetConnection(Context context){
        ctx=context;
    }
    public void checkConnection(){

        if(!isInternetConnected()) {

            final FancyAlertDialog.Builder alert = new FancyAlertDialog.Builder(ctx)
                .setBackgroundColor(R.color.accent)
                .setImageResource(R.drawable.internetconnection)
                .setTitle("No Internet")
                .setTextSubTitle("Cannot connect to a servers")
                .setBody(R.string.noconnection)
                .setPositiveButtonText("Connect Now")
                .setPositiveColor(R.color.colorPrimaryDark)
```

```

        .setPositiveButton(new FancyAlertDialog.OnPositiveButtonClicked() {
            @Override
            public void onClick(View view, Dialog dialog) {

                if(isInternetConnected()){
                    dialog.dismiss();
                }else
            {
                Intent dialogIntent = new
Intent(android.provider.Settings.ACTION_SETTINGS);
                dialogIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                ctx.startActivity(dialogIntent);
            }
        })
        .setBodyGravity(FancyAlertDialog.TextGravity.CENTER)
        .setTitleGravity(FancyAlertDialog.TextGravity.CENTER)
        .setSubtitleGravity(FancyAlertDialog.TextGravity.CENTER)
        .setCancelable(false)
        .build();
        alert.show();
    }
}

private boolean isInternetConnected() {
    ConnectivityManager cm = (ConnectivityManager)
ctx.getSystemService(Context.CONNECTIVITY_SERVICE);
    return cm.getActiveNetworkInfo() != null
        && cm.getActiveNetworkInfo().isConnectedOrConnecting();
}
}

```

## #To place order

```

public class PlacedOrderModel {
    private String
orderid,no_of_items,total_amount,delivery_date,payment_mode,deleiveryname,deliveryemail,
deliverymobile_no,deliveryaddress,deliverypincode;
    private String placed_user_name,Placed_user_email,Placed_user_mobile_no;
}

```

```

public PlacedOrderModel (String orderid, String no_of_items, String total_amount, String
delivery_date, String payment_mode, String deleiveryname, String deliveryemail, String
deliverymobile_no, String deliveryaddress, String deliverypincode, String placed_user_name,
String getPlaced_user_email, String getPlaced_user_mobile_no)
{
    this.orderid = orderid;
    this.no_of_items = no_of_items;
    this.total_amount = total_amount;
    this.delivery_date = delivery_date;
    this.payment_mode = payment_mode;
    this.deleiveryname = deleiveryname;
    this.deliveryemail = deliveryemail;
    this.deliverymobile_no = deliverymobile_no;
    this.deliveryaddress = deliveryaddress;
    this.deliverypincode = deliverypincode;
    this.placed_user_name = placed_user_name;
    this.Placed_user_email = getPlaced_user_email;
    this.Placed_user_mobile_no = getPlaced_user_mobile_no;
} public String getOrderid() {
    return orderid;
} public void setOrderid(String orderid) {
    this.orderid = orderid;
} public String getPlaced_user_email() {
    return Placed_user_email;
} public void setPlaced_user_email(String placed_user_email) {
    Placed_user_email = placed_user_email;
}
public String getPlaced_user_mobile_no() {
    return Placed_user_mobile_no;
}
public void setPlaced_user_mobile_no(String placed_user_mobile_no) {
    Placed_user_mobile_no = placed_user_mobile_no;
} public String getNo_of_items() {
    return no_of_items;
} public void setNo_of_items(String no_of_items) {
    this.no_of_items = no_of_items;
} public String getTotal_amount() {
    return total_amount;
} public void setTotal_amount(String total_amount) {
    this.total_amount = total_amount;
} public String getDelivery_date() {
    return delivery_date;
} public void setDelivery_date(String delivery_date) {
    this.delivery_date = delivery_date;
} public String getPayment_mode() {
    return payment_mode;
} public void setPayment_mode(String payment_mode) {
    this.payment_mode = payment_mode;
} public String getDeleiveryname() {
    return deleiveryname;
} public void setDeleiveryname(String deleiveryname) {

```



```

    this.deleiveryname = deleiveryname;
}   public String getDeliveryemail() {
    return deliveryemail;
}   public void setDeliveryemail(String deliveryemail) {
    this.deliveryemail = deliveryemail;
}   public String getDeliverymobile_no() {
    return deliverymobile_no;
}   public void setDeliverymobile_no(String deliverymobile_no) {
    this.deliverymobile_no = deliverymobile_no;
}   public String getDeliveryaddress() {
    return deliveryaddress;
}   public void setDeliveryaddress(String deliveryaddress) {
    this.deliveryaddress = deliveryaddress;
}   public String getDeliverypincode() {
    return deliverypincode;
}   public void setDeliverypincode(String deliverypincode) {
    this.deliverypincode = deliverypincode;
}   public String getPlaced_user_name() {
    return placed_user_name; }

public void setPlaced_user_name(String placed_user_name) {
    this.placed_user_name = placed_user_name;
}   }

```

## #cart

```

public class Cart extends AppCompatActivity {

    //to get user session data
    private UserSession session;
    private HashMap<String,String> user;
    private String name,email,photo,mobile;
    private RecyclerView mRecyclerView;
    private StaggeredGridLayoutManager mLayoutManager;

    //Getting reference to Firebase Database
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference mDatabaseReference = database.getReference();
    private LottieAnimationView tv_no_item;
    private LinearLayout activitycartlist;
    private LottieAnimationView emptycart;

    private ArrayList<SingleProductModel> cartcollect;
    private float totalcost=0;
    private int totalproducts=0;

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

```

```

Toolbar toolbar = findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
toolbar.setTitle("Cart");

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setDisplayShowHomeEnabled(true);

//check Internet Connection
new CheckInternetConnection(this).checkConnection();

//retrieve session values and display on listviews
getValues();

//SharedPreferences for Cart Value
session = new UserSession(getApplicationContext());

//validating session
session.isLoggedIn();

mRecyclerView = findViewById(R.id.recyclerview);
tv_no_item = findViewById(R.id.tv_no_cards);
activitycartlist = findViewById(R.id.activity_cart_list);
emptycart = findViewById(R.id.empty_cart);
cartcollect = new ArrayList<>();

if (mRecyclerView != null) {
    //to enable optimization of recyclerview
    mRecyclerView.setHasFixedSize(true);
}
//using staggered grid pattern in recyclerview
mLayoutManager = new StaggeredGridLayoutManager(1,
StaggeredGridLayoutManager.VERTICAL);
mRecyclerView.setLayoutManager(mLayoutManager);

if(session.getCartValue()>0) {
    populateRecyclerView();
}else if(session.getCartValue() == 0) {
    tv_no_item.setVisibility(View.GONE);
    activitycartlist.setVisibility(View.GONE);
    emptycart.setVisibility(View.VISIBLE);
}
}

private void populateRecyclerView() {

    //Say Hello to our new FirebaseUI android Element, i.e., FirebaseRecyclerViewAdapter
    final FirebaseRecyclerViewAdapter<SingleProductModel,MovieViewHolder> adapter = new
FirebaseRecyclerViewAdapter<SingleProductModel, MovieViewHolder>(
        SingleProductModel.class,

```

```

        R.layout.cart_item_layout,
        MovieViewHolder.class,
        //referencing the node where we want the database to store the data from our Object
        mDatabaseReference.child("cart").child(mobile).getRef()
    ){
        @Override
        protected void populateViewHolder(final MovieViewHolder viewHolder, final
        SingleProductModel model, final int position) {
            if(tv_no_item.getVisibility() != View.VISIBLE){
                tv_no_item.setVisibility(View.GONE);
            }
            viewHolder.cardname.setText(model.getPrname());
            viewHolder.cardprice.setText("₹ "+model.getPrprice());
            viewHolder.cardcount.setText("Quantity : "+model.getNo_of_items());
            Picasso.with(Cart.this).load(model.getPrimage()).into(viewHolder.cardimage);

            totalcost += model.getNo_of_items()*Float.parseFloat(model.getPrprice());
            totalproducts += model.getNo_of_items();
            cartcollect.add(model);

            viewHolder.carddelete.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {

                    Toast.makeText(Cart.this,getItem(position).getPrname(),Toast.LENGTH_SHORT).show();
                    getRef(position).removeValue();
                    session.decreaseCartValue();
                    startActivity(new Intent(Cart.this,Cart.class));
                    finish();
                }
            });
        }
    };
    mRecyclerView.setAdapter(adapter);
}

public void checkout(View view) {
    Intent intent = new Intent(Cart.this,OrderDetails.class);
    intent.putExtra("totalprice",Float.toString(totalcost));
    intent.putExtra("totalproducts",Integer.toString(totalproducts));
    intent.putExtra("cartproducts",cartcollect);
    startActivity(intent);
    finish();
}

//viewHolder for our Firebase UI
public static class MovieViewHolder extends RecyclerView.ViewHolder{

    TextView cardname;
    ImageView cardimage;
    TextView cardprice;

```

```
TextView cardcount;
ImageView carddelete;
```

```
View mView;
public MovieViewHolder(View v) {
    super(v);
    mView = v;
    cardname = v.findViewById(R.id.cart_prtitle);
    cardimage = v.findViewById(R.id.image_cartlist);
    cardprice = v.findViewById(R.id.cart_prprice);
    cardcount = v.findViewById(R.id.cart_prcount);
    carddelete = v.findViewById(R.id.deletecard);
}
}
```

```
private void getValues() {
```

```
    //create new session object by passing application context
    session = new UserSession(getApplicationContext());
```

```
    //validating session
    session.isLoggedIn();
```

```
    //get User details if logged in
    user = session.getUserDetails();
```

```
    name = user.get(UserSession.KEY_NAME);
    email = user.get(UserSession.KEY_EMAIL);
    mobile = user.get(UserSession.KEY_MOBILE);
    photo = user.get(UserSession.KEY_PHOTO);
```

```
}
```

```
@Override
```

```
public boolean onSupportNavigateUp() {
    onBackPressed();
    return true;
}
```

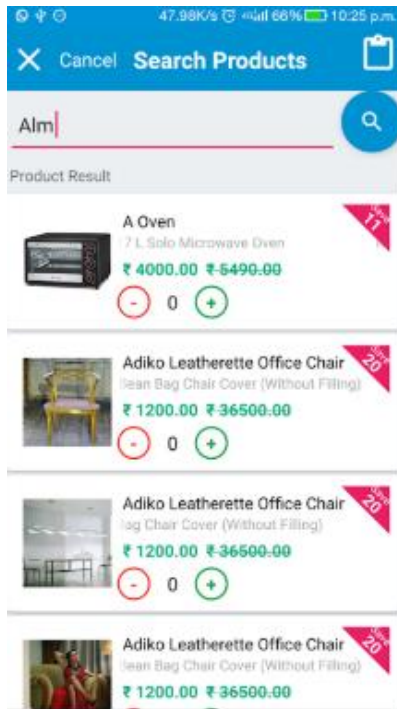
```
public void viewProfile(View view) {
    startActivity(new Intent(Cart.this,Profile.class));
    finish();
}
```

```
@Override
```

```
protected void onResume() {
    super.onResume();
    //check Internet Connection
    new CheckInternetConnection(this).checkConnection();
}
```

```
public void Notifications(View view) {
    startActivity(new Intent(Cart.this,NotificationActivity.class));
    finish(); } }
```

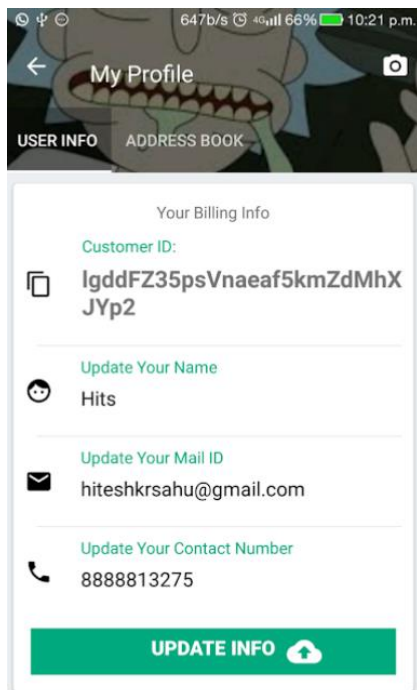
## Screenshots



---

*User can view various products just by searching what they need*

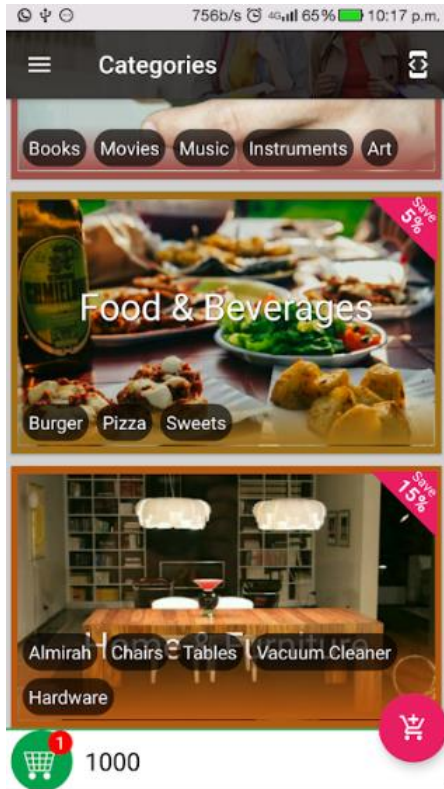
---



---

*Customer can change their contact details if they want to*

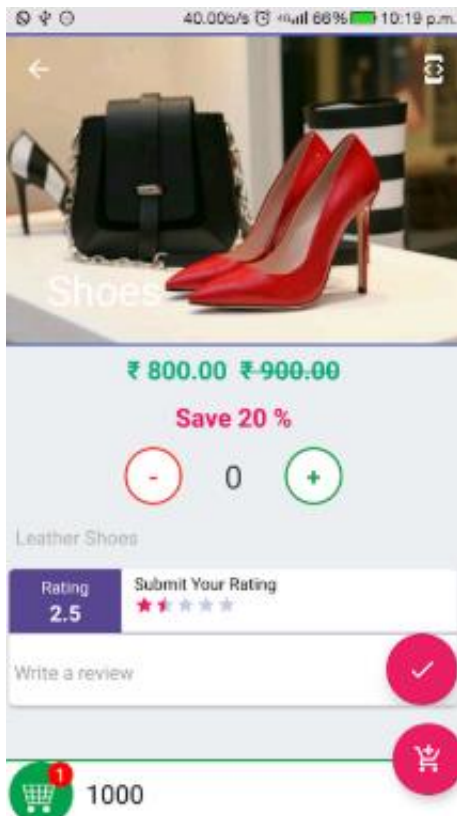
---



---

*Our application has many category to shop/order from. If they want to order food or if they want to shop online for other products This make our application "one stop to shop from"*

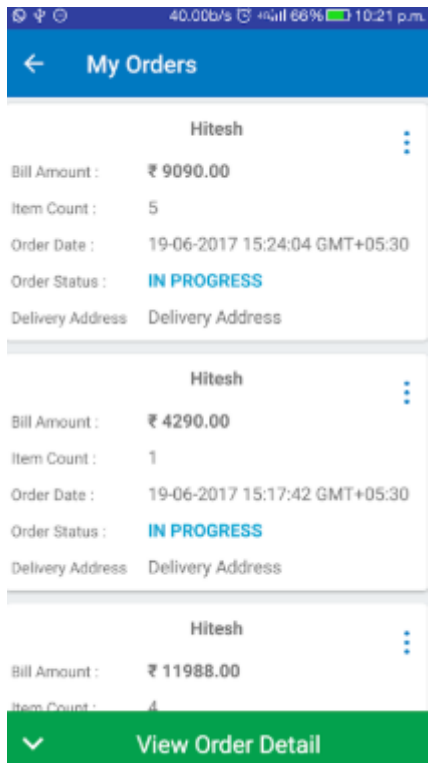
---



---

*Interesting offers make customers buying their products at more affordable price*

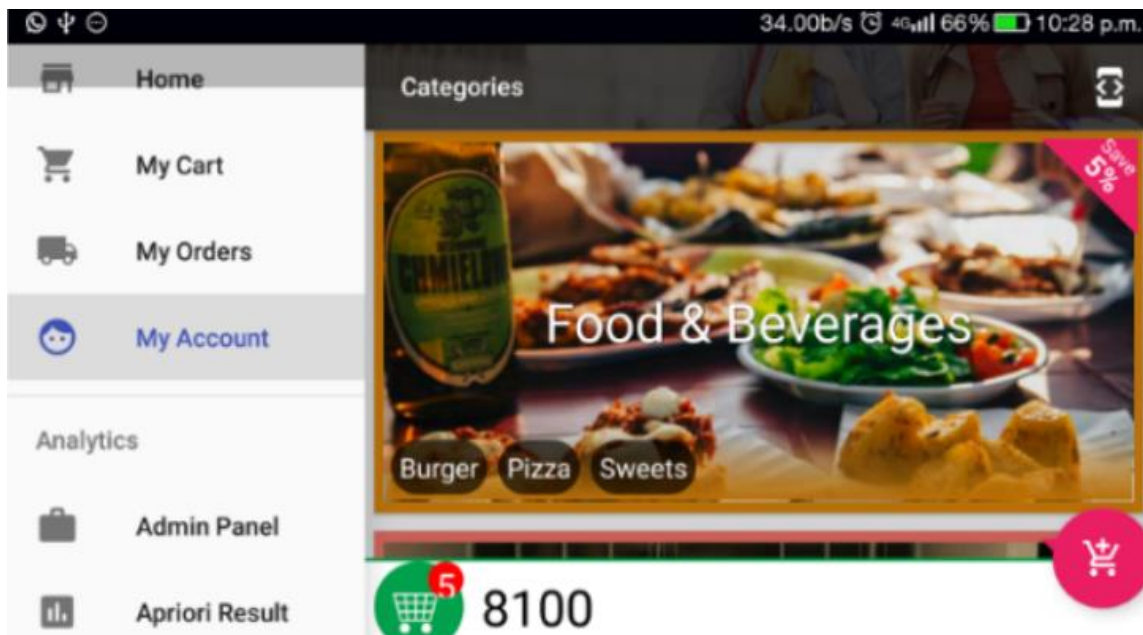
---



---

*User can view their past order details even if they are not connected to the internet. They can track their order.*

---



---

*Our application also works on landscape mode that make our system different from others*

---

## **Conclusion**

Increased Internet penetration, annoy free shopping environment and really high levels of Net savings to ascertain Indians online shopping. But at the same time, companies want to decrease the risks associated with the customer. The objective is not to convert all shoppers to online procuring but to display them it's a choice. In totaling to above, efforts want to be taken to teach the online consumers on the stages that need to be started while creating online buying. Furthermore, the response of an online customer should be taken to classify defects in service transfer. This can be done over online people and blogs that assist as marketing and advertising apparatuses and a basis of response for enterprises. Thus, online selling increases more.

In future, User can shop online just by clicking a photo from camera and shop online. Delivery can be made in several hours. Reachability of delivery will be all over the India even in rural and urban area.

## **References**

[1]. Android Studio Cardview and ListView

<https://www.youtube.com/watch?v=HMjI7cLsyfw>

[2] Saving bitmap to firebase

<https://stackoverflow.com/questions/40885860/how-to-save-bitmap-to-firebase>

[3] Access MySQL from android studio

<https://www.javahelps.com/2018/12/access-mysql-from-android-through.html>