

National University of Technology



MOBILE APP DEVELOPEMENT

LAB Submitted to:

Mam Saima Razaq

Submitted By:

Muhammad Abdullah (F22607010)

Muhammad Zohaib Afzaal (F22607027)

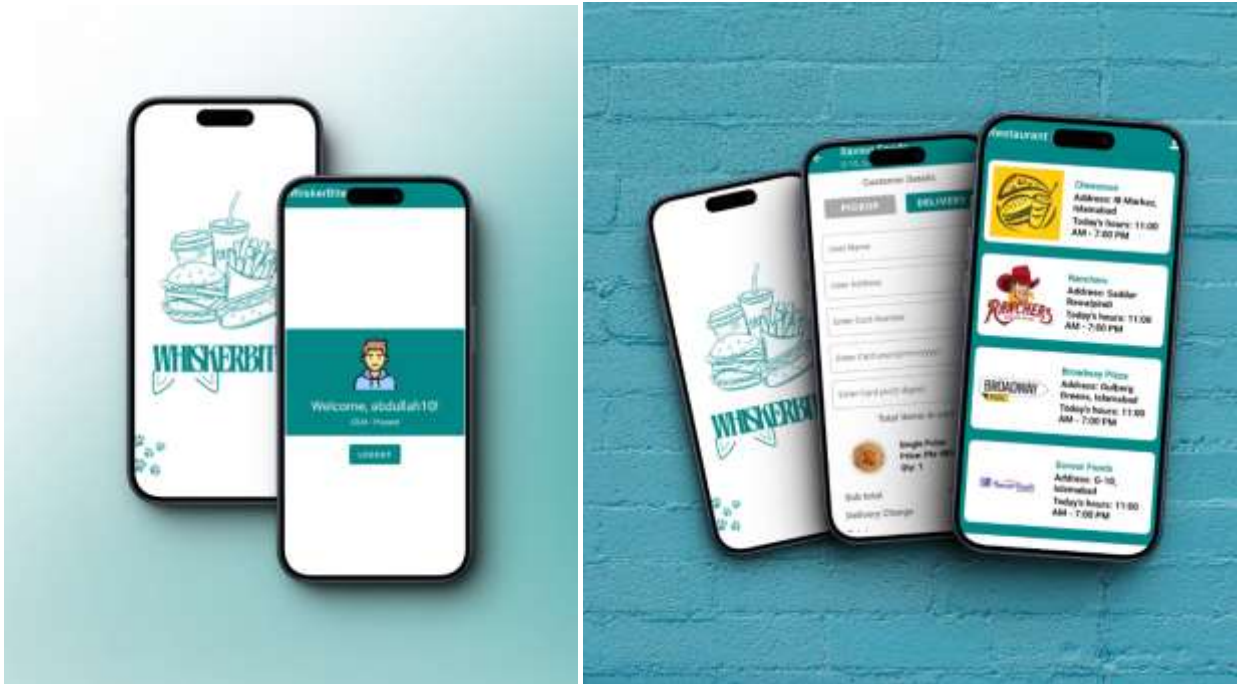
Moez Ahmed(F22607020)

Sultan Mehmood (F22607013)

Batch:

Fall AI 2022

WhiskerBites: Food Ordering App



Introduction

WhiskerBites is an innovative food ordering application designed to provide users with a seamless and efficient way to order their favorite meals from various restaurants. The app incorporates advanced authentication mechanisms, intuitive restaurant and menu listings, comprehensive cart management, and flexible order fulfillment options, including delivery and pickup.




Features

1. User Authentication and Login/Signup:

- **Advanced Authentication Rules:** Ensuring user security through robust password handling, including requirements for

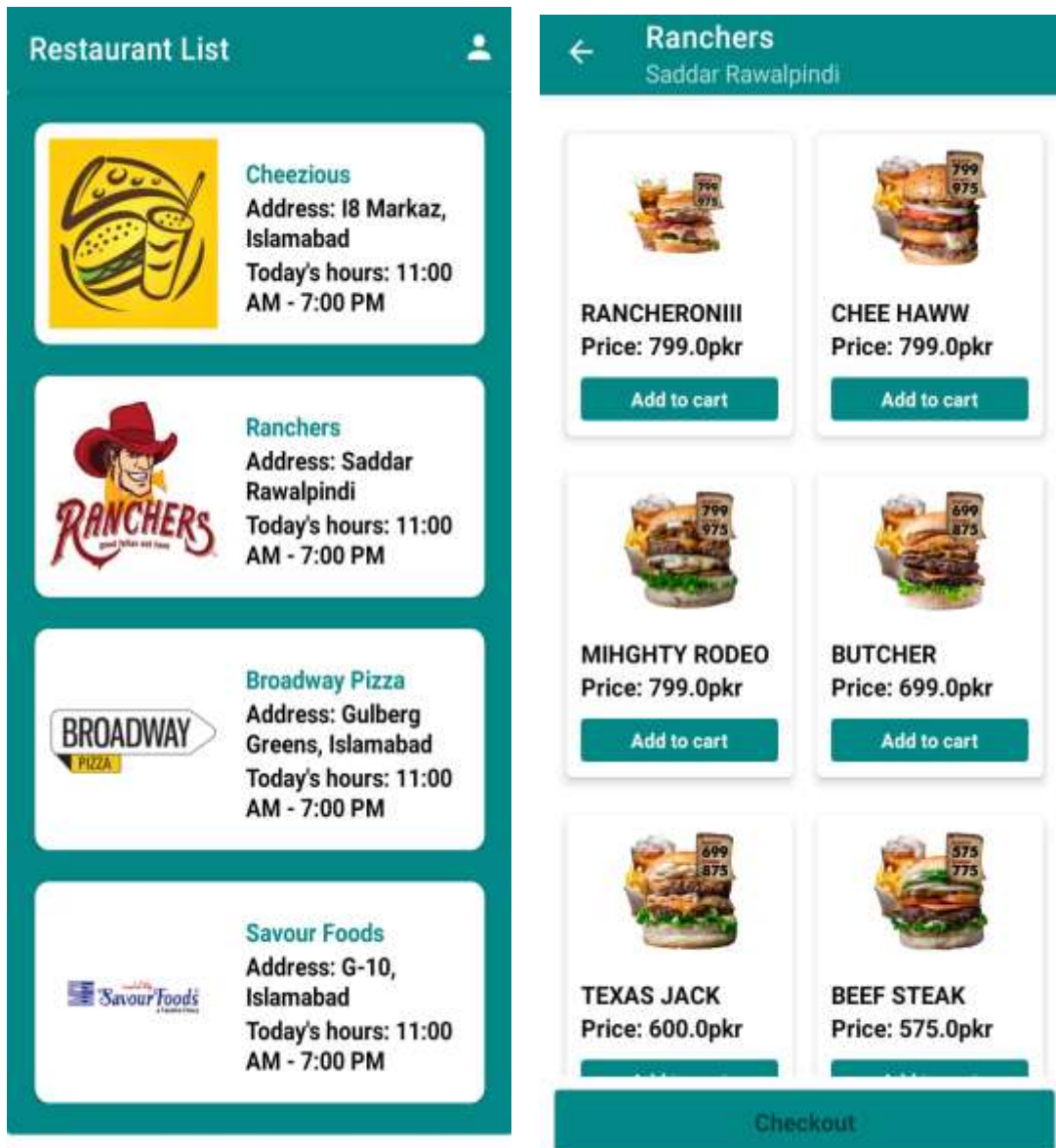
minimum length, special characters, and a mix of upper and lower case letters.

- **Login/Signup with SharedPreferences and Database Integration:** Utilizing SharedPreferences for efficient session management and a secure database for storing user credentials and other sensitive information.

| WhiskerBites | WhiskerBites |
|--|--|
| <h3>Login</h3> <div>User Name</div> <div>Password </div> <div>LOGIN IN</div> <div>Don't Have Account? Sign Up</div> | <h3>Register</h3> <div>User Name</div> <div>Recovery Email</div> <div>Password </div> <div>Confirm Password </div> <div>REGISTER</div> <div>Already have an account? Login</div> |

2. Restaurant and Menu Listings:

- **Comprehensive Listings:** Displaying a list of available restaurants, each containing its respective menu.
- **Dynamic Updates:** Menus are dynamically updated to reflect the latest offerings and prices from the restaurants.



3. Cart Management:


- **Add/Remove Items:** Users can easily add or remove items from their cart, with real-time updates to the total price.
- **Quantity Management:** Users can adjust the quantity of each item directly from the cart.
- **Cart Persistence:** The app saves the cart state, allowing users to resume their order process even after closing the app.

← Ranchers

Saddar Rawalpindi

← Savour Foods


G-10, Islamabad



RANCHERONIII

Price: 799.0pkr


Add to cart



CHEE HAWW

Price: 799.0pkr


− 1 +



MIHGHTY RODEO

Price: 799.0pkr


Add to cart



BUTCHER


Price: 699.0pkr

Add to cart



TEXAS JACK

Price: 600.0pkr



BEEF STEAK

Price: 575.0pkr

Checkout (1) items

Customer Details

PICKUP

DELIVERY

User Name


User Address

Enter Card Number

Enter Card expiry(mm/yyyy)

Enter Card pin(3 digits)

Total items in cart



Single Pulao

Price: Pkr 485.00

Qty: 1

Sub total

Pkr 485.00

Delivery Charge

Pkr 6.00

Total

Pkr 491.00

4.Delivery and Pickup Options:

- **Delivery Service:** Users can choose delivery, with real-time tracking of the order status and delivery progress.

- **Pickup Service:** For users who prefer to collect their orders, the app provides a streamlined pickup option, including estimated preparation times and notifications when the order is ready.

← Savour Foods
G-10, Islamabad

Customer Details

PICKUP DELIVERY

User Name


User Address

Enter Card Number

Enter Card expiry(mm/yyyy)

Enter Card pin(3 digits)

Total items in cart

 Single Pulao
Price: Pkr 485.00
Qty: 1

| | |
|-----------------|------------|
| Sub total | Pkr 485.00 |
| Delivery Charge | Pkr 6.00 |
| Total | Pkr 491.00 |

5.Order Placement:

- **Place Order Feature:** Users can review their cart and place an order with a single click, initiating the payment and order confirmation process.
- **Order Summary:** A detailed summary of the order, including itemized costs, delivery charges, and estimated delivery time, is provided before final confirmation.



Thank You!

Your Order Placed
Successfully

Done

Technical Implementation

Authentication and Login/Signup

- **SharedPreferences for Session Management:**

```
• package com.android.foodorderapp;

import android.content.Context;
import android.content.SharedPreferences;

public class SharedPreferencesHelper {
    private static final String PREFS_NAME = "user_prefs";
    private static final String KEY_IS_LOGGED_IN = "is_logged_in";
    private static final String KEY_USERNAME = "username";

    private SharedPreferences sharedPreferences;

    public SharedPreferencesHelper(Context context) {
        sharedPreferences = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE);
    }

    public void setLoggedIn(boolean loggedIn) {
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putBoolean(KEY_IS_LOGGED_IN, loggedIn);
        editor.apply();
    }

    public boolean isLoggedIn() {
```

```

        return sharedPreferences.getBoolean(KEY_IS_LOGGED_IN, false);
    }

    public void setUsername(String username) {
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString(KEY_USERNAME, username);
        editor.apply();
    }

    public String getUsername() {
        return sharedPreferences.getString(KEY_USERNAME, "User");
    }
}

```

• Database Integration for Storing User Data:

```

• package com.android.foodorderapp;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
public class DBHelper extends SQLiteOpenHelper {
    public static final String DBNAME = "Login.db";
    public DBHelper(Context context) {
        super(context, "Login.db", null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase MyDB) {
        MyDB.execSQL("create Table users(username TEXT primary key, password TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase MyDB, int i, int i1) {
        MyDB.execSQL("drop Table if exists users");
    }

    public Boolean insertData(String username, String password){
        SQLiteDatabase MyDB = this.getWritableDatabase();
        ContentValues contentValues= new ContentValues();
        contentValues.put("username", username);
        contentValues.put("password", password);
        long result = MyDB.insert("users", null, contentValues);
        if(result==-1) return false;
        else
            return true;
    }

    public Boolean checkusername(String username) {
        SQLiteDatabase MyDB = this.getWritableDatabase();
        Cursor cursor = MyDB.rawQuery("Select * from users where username = ?", new String[]{username});
        if (cursor.getCount() > 0)
            return true;
        else
            return false;
    }

    public Boolean checkusernamepassword(String username, String

```



```
password) {
    SQLiteDatabase MyDB = this.getWritableDatabase();
    Cursor cursor = MyDB.rawQuery("Select * from users where username
= ? and password = ?", new String[] {username,password});
    if(cursor.getCount()>0)
        return true;
    else
        return false;
}
}
```

Restaurant and Menu Listings

- **Fetching Data from Database:**

```
• package com.android.foodorderapp.adapters;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.android.foodorderapp.R;
import com.android.foodorderapp.model.RestaurantModel;
import com.bumptech.glide.Glide;

import java.util.List;

public class RestaurantListAdapter extends
RecyclerView.Adapter<RestaurantListAdapter.MyViewHolder> {

    private List<RestaurantModel> restaurantModelList;
    private RestaurantListClickListener clickListener;

    public RestaurantListAdapter(List<RestaurantModel>
restaurantModelList, RestaurantListClickListener clickListener) {
        this.restaurantModelList = restaurantModelList;
        this.clickListener = clickListener;
    }

    public void updateData(List<RestaurantModel> restaurantModelList) {
        this.restaurantModelList = restaurantModelList;
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public RestaurantListAdapter.MyViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.recycler_row,
parent, false);
        return new MyViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull
RestaurantListAdapter.MyViewHolder holder, int position) {
        holder.restaurantName.setText(restaurantModelList.get(position).getName())
```

```

;
        holder.restaurantAddress.setText("Address:
"+restaurantModelList.get(position).getAddress());
        holder.restaurantHours.setText("Today's hours: " +
restaurantModelList.get(position).getHours().getTodayHours());

        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

clickListener.onItemClick(restaurantModelList.get(position));
            }
        });
        Glide.with(holder.thumbImage)
            .load(restaurantModelList.get(position).getImage())
            .into(holder.thumbImage);

    }

    @Override
    public int getItemCount() {
        return restaurantModelList.size();
    }

    static class MyViewHolder extends RecyclerView.ViewHolder {
        TextView restaurantName;
        TextView restaurantAddress;
        TextView restaurantHours;
        ImageView thumbImage;

        public MyViewHolder(View view) {
            super(view);
            restaurantName = view.findViewById(R.id.restaurantName);
            restaurantAddress = view.findViewById(R.id.restaurantAddress);
            restaurantHours = view.findViewById(R.id.restaurantHours);
            thumbImage = view.findViewById(R.id.thumbImage);
        }
    }

    public interface RestaurantClickListener {
        public void onItemClick(RestaurantModel restaurantModel);
    }
}

• package com.android.foodorderapp.adapters;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.android.foodorderapp.R;
import com.android.foodorderapp.model.RestaurantModel;
import com.bumptech.glide.Glide;

import java.util.List;

public class RestaurantListAdapter extends
RecyclerView.Adapter<RestaurantListAdapter.MyViewHolder> {

```

```

        private List<RestaurantModel> restaurantModelList;
        private RestaurantListClickListener clickListener;

        public RestaurantListAdapter(List<RestaurantModel>
restaurantModelList, RestaurantListClickListener clickListener) {
            this.restaurantModelList = restaurantModelList;
            this.clickListener = clickListener;
        }

        public void updateData(List<RestaurantModel> restaurantModelList) {
            this.restaurantModelList = restaurantModelList;
            notifyDataSetChanged();
        }

        @NonNull
        @Override
        public RestaurantListAdapter.MyViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
            View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.recycler_row,
parent, false);
            return new MyViewHolder(view);
        }

        @Override
        public void onBindViewHolder(@NonNull
RestaurantListAdapter.MyViewHolder holder, int position) {

holder.restaurantName.setText(restaurantModelList.get(position).getName());
;

            holder.restaurantAddress.setText("Address:
"+restaurantModelList.get(position).getAddress());
            holder.restaurantHours.setText("Today's hours: " +
restaurantModelList.get(position).getHours().getTodaysHours());

            holder.itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {

clickListener.onItemClick(restaurantModelList.get(position));
                }
            });
            Glide.with(holder.thumbImage)
                .load(restaurantModelList.get(position).getImage())
                .into(holder.thumbImage);
        }

        @Override
        public int getItemCount() {
            return restaurantModelList.size();
        }

        static class MyViewHolder extends RecyclerView.ViewHolder {
            TextView restaurantName;
            TextView restaurantAddress;
            TextView restaurantHours;
            ImageView thumbImage;

            public MyViewHolder(View view) {
                super(view);
                restaurantName = view.findViewById(R.id.restaurantName);
                restaurantAddress = view.findViewById(R.id.restaurantAddress);
                restaurantHours = view.findViewById(R.id.restaurantHours);
            }
        }
    }

```

```

        thumbImage = view.findViewById(R.id.thumbImage);

    }

    }

    public interface RestaurantListClickListener {
        public void onItemClick(RestaurantModel restaurantModel);
    }

}

```

Cart Management

```

private void setDeliveryOption(boolean delivery) {
    isDeliveryOn = delivery;
    if (delivery) {
        btnDelivery.setBackgroundColor(getResources().getColor(R.color.blue));

        btnPickup.setBackgroundColor(getResources().getColor(android.R.color.darker_gray));

        Addresslayout.setVisibility(View.VISIBLE);
        tvDeliveryChargeAmount.setVisibility(View.VISIBLE);
        tvDeliveryCharge.setVisibility(View.VISIBLE);
    } else {
        btnPickup.setBackgroundColor(getResources().getColor(R.color.blue));

        btnDelivery.setBackgroundColor(getResources().getColor(android.R.color.darker_gray));

        Addresslayout.setVisibility(View.GONE);
        tvDeliveryChargeAmount.setVisibility(View.GONE);
        tvDeliveryCharge.setVisibility(View.GONE);
    }
}

private void calculateTotalAmount(RestaurantModel restaurantModel) {
    float subTotalAmount = 0f;

    for(Menu m : restaurantModel.getMenus()) {
        subTotalAmount += m.getPrice() * m.getTotalInCart();
    }

    tvSubtotalAmount.setText("Pkr " + String.format("%.2f", subTotalAmount));
    if (isDeliveryOn) {
        tvDeliveryChargeAmount.setText("Pkr " + String.format("%.2f",
restaurantModel.getDelivery_charge()));
        subTotalAmount += restaurantModel.getDelivery_charge();
    }
    tvTotalAmount.setText("Pkr " + String.format("%.2f", subTotalAmount));
}

```

Order Placement

```

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

    RestaurantModel restaurantModel =
getIntent().getParcelableExtra("RestaurantModel");
    ActionBar actionBar = getSupportActionBar();
    actionBar.setTitle(restaurantModel.getName());
    actionBar.setSubtitle(restaurantModel.getAddress());
    actionBar.setDisplayHomeAsUpEnabled(true);
}

```

```

TextView buttonDone = findViewById(R.id.buttonDone);
buttonDone.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
}

```

Delivery and Pickup

- **Delivery Option:**

- ```

private void setDeliveryOption(boolean delivery) {
 isDeliveryOn = delivery;
 if (delivery) {

 btnDelivery.setBackgroundColor(getResources().getColor(R.color.blue));

 btnPickup.setBackgroundColor(getResources().getColor(android.R.color.darker_gray));
 Addresslayout.setVisibility(View.VISIBLE);
 tvDeliveryChargeAmount.setVisibility(View.VISIBLE);
 tvDeliveryCharge.setVisibility(View.VISIBLE);
 } else {

 btnPickup.setBackgroundColor(getResources().getColor(R.color.blue));

 btnDelivery.setBackgroundColor(getResources().getColor(android.R.color.darker_gray));
 Addresslayout.setVisibility(View.GONE);
 tvDeliveryChargeAmount.setVisibility(View.GONE);
 tvDeliveryCharge.setVisibility(View.GONE);
 }
}

```

## Conclusion

WhiskerBites aims to revolutionize the food ordering experience with its user-friendly interface, secure authentication, and comprehensive features. By integrating advanced technological solutions, the app ensures a reliable and enjoyable experience for all users, whether they prefer delivery or pickup.