**Code review**

RateDriverActivity.java`

Structure Improvement

הוספת מתודות initializeUI and setUpButtons כדי להפריד בין הלוגיקה של תצוגת UI -הצגת רכיבים בממשק משתמש לבין לוגיקת הפעולה -כלומר מה קורה כאשר משתמש לוחץ על כפתור או מבצע פעולה.ובנוסף שמירה על עיקרון DRY

הוספת מתודה showToast אשר מנהלת את הודעות Toast- הודעות אשר מופיעות על המסך למשך מספר שניות במקום כתיבה חוזרת של makeTest- שמירה על עיקרון DRY

Bug Detection and Efficiency

הוספת בדיקה שיש מזהה משתמש driverID כלומר בעת ביצוע פעולות משתמש נראצה לבדוק שיש מזהה משמש תקין ואם לא נציג הודעת Toast במקרה של חוסר

הוספנו הודעת שגיאה במקרה של כישלון בקריאת הנתונים מ-Firestore.

**המלצות לשיפור הקוד**

1. **הוספת טיפול בזמינות אינטרנט**:  
   לפני ביצוע קריאת הנתונים מ-Firestore, יש לבדוק אם המכשיר מחובר לאינטרנט.  
   ניתן להשתמש במחלקה **ConnectivityManager** כדי למנוע קריסות במקרה של חוסר חיבור.
2. **שימוש ב-ViewBinding לניהול רכיבי הממשק**:  
   במקום שימוש ב-findViewById(), כדאי לעבור ל-ViewBinding.  
   השימוש ב-ViewBinding הופך את הקוד לקריא יותר, מפחית שגיאות של NullPointerException ומפשט את ניהול הממשק.
3. **שיפור תצוגת השגיאות באמצעות Snackbar במקום Toast:**  
   Snackbar מספק חוויית משתמש טובה יותר מכיוון שהוא מאפשר אינטראקציה, למשל, עם כפתור Undo.  
   בנוסף, הודעות השגיאה נשארות זמן ממושך יותר ונראות בצורה ברורה בתחתית המסך.

package com.example.projectcampusride;

import android.content.Intent;

import android.os.Bundle;

import android.widget.Button;

import android.widget.ImageButton;

import android.widget.RatingBar;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.firestore.FirebaseFirestore;

public class RateDriverActivity extends AppCompatActivity {

    private RatingBar ratingBar;

    private Button btnSubmitRating;

    private FirebaseFirestore firestore;

    private String userId;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.rate\_driver\_activity);

        //הפרדת  initializeUI() כדי להפריד את הui לפונקציה נפרדת

        //הפרדת  setUpButtons() כדי להפריד את הגדרת הכפתורים לפונקציה נפרדת

        //בדיקה האם driverID קיים

        userId = getIntent().getStringExtra("driverId");

        if (userId == null || userId.isEmpty()) {

            // הצגת הודעת שגיאה במקרה של חוסר `driverId`

            showToast("User ID is missing!");

            finish();

            return;

        }

    }

    private void initializeUI() {

        // הגדרת רכיבי הממשק

        ratingBar = findViewById(R.id.ratingBar);

        btnSubmitRating = findViewById(R.id.btn\_submit\_rating);

        firestore = FirebaseFirestore.getInstance();

    }

    private void setUpButtons() {

        // הגדרת כפתור הגדרות

        findViewById(R.id.settings\_button).setOnClickListener(v ->

                startActivity(new Intent(this, SettingsActivity.class)));

        // הגדרת כפתור התראות

        findViewById(R.id.notification\_button).setOnClickListener(v ->

                startActivity(new Intent(this, NotificationsActivity.class)));

        // הגדרת כפתור חזרה

        findViewById(R.id.back\_button).setOnClickListener(v -> onBackPressed());

        // הגדרת כפתור שליחת דירוג

        btnSubmitRating.setOnClickListener(v -> {

            float newRating = ratingBar.getRating();

            if (newRating == 0) {

                // הודעת שגיאה במקרה של דירוג ריק

                showToast("Please select a rating!");

            } else {

                submitRating(newRating);

            }

        });

    }

    private void submitRating(float newRating) {

        // שליחת דירוג ל-Firestore

        firestore.collection("users").document(userId)

                .get()

                .addOnCompleteListener(task -> {

                    if (task.isSuccessful() && task.getResult() != null) {

                        // חישוב ממוצע דירוג חדש

                        Double currentRating = task.getResult().getDouble("rating");

                        Integer currentCount = task.getResult().getLong("ratingCount").intValue();

                        double totalRating = currentRating \* currentCount + newRating;

                        int newCount = currentCount + 1;

                        double updatedRating = totalRating / newCount;

                        // עדכון הדירוג ב-Firestore

                        firestore.collection("users").document(userId)

                                .update("rating", updatedRating, "ratingCount", newCount)

                                .addOnSuccessListener(aVoid -> showToast("Rating submitted successfully!"))

                                // הודעת שגיאה במקרה של כישלון בעדכון הדירוג

                                .addOnFailureListener(e -> showToast("Failed to submit rating."));

                    } else {

                        // הודעת שגיאה במקרה של כישלון בקריאת נתוני המשתמש

                        showToast("Error retrieving user data.");

                    }

                });

    }

    private void showToast(String message) {

        // פונקציה להצגת הודעות Toast באופן מרוכז

        Toast.makeText(this, message, Toast.LENGTH\_SHORT).show();

    }

}