```
import org.apache.mahout.cf.taste.impl.model.file.FileDataModel;
import org.apache.mahout.cf.taste.model.DataModel;
import org.apache.mahout.cf.taste.similarity.UserSimilarity;
import org.apache.mahout.cf.taste.impl.similarity.PearsonCorrelationSimilarity;
import org.apache.mahout.cf.taste.neighborhood.UserNeighborhood;
import org.apache.mahout.cf.taste.impl.neighborhood.NearestNUserNeighborhood;
import org.apache.mahout.cf.taste.recommender.Recommender;
import org.apache.mahout.cf.taste.impl.recommender.GenericUserBasedRecommender;
import org.apache.mahout.cf.taste.recommender.RecommendedItem;

import java.io.File;
import java.util.List;
import java.util.Scanner;

public class RecommenderSystem {

    public static void main(String[] args) {
        try {
            System.out.println("=== Product Recommendation System ===");
            System.out.println("Loading data from file: data.csv");

            // Load user-item preference data
            DataModel model = new FileDataModel(new File("data.csv"));

            // Calculate similarity between users
            UserSimilarity similarity = new PearsonCorrelationSimilarity(model);

            // Define user neighborhood (2 nearest users)
            UserNeighborhood neighborhood = new NearestNUserNeighborhood(2, similarity,
model);

            // Create user-based recommender
            Recommender recommender = new GenericUserBasedRecommender(model,
neighborhood, similarity);

            // Input: user ID to generate recommendations
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter user ID to get recommendations: ");
            int userID = sc.nextInt();

            // Generate top 3 recommendations
            List<RecommendedItem> recommendations = recommender.recommend(userID, 3);

            System.out.println("\nRecommended items for user " + userID + ":");
```

```
            if (recommendations.isEmpty()) {
                System.out.println("No recommendations available (possibly already rated all
items).");
            } else {
                for (RecommendedItem item : recommendations) {
                    System.out.printf(" - Item ID: %d, Predicted Rating: %.2f%n",
                            item.getItemID(), item.getValue());
                }
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```