**Recommendation System – Glam Heel Hangout**

The recommendation system in the Glam Heel Hangout app suggests products to users based on their previous purchases and interests (saved favorites).  
If the user has no favorites or purchase history, the system uses a fallback approach by showing the top-rated products based on user reviews.

The goal is to offer a personalized shopping experience using ML.NET and the Matrix Factorization algorithm.

**Implementation Overview**

Key Classes and Components

* ProductService.Recommend(userId)  
  Main method that returns recommended products using a trained model.  
  If the user has no favorites, it falls back to products with the highest average rating.
* ProductEntry  
  A simple class used for mapping user-product interaction:
  + UserId
  + ProductId
  + Label (always 1, since only interested pairs are used)
* ProductScore  
  Used for prediction output. Holds a Score value for each (user, product) pair.
* ML.NET Model  
  Uses MatrixFactorizationTrainer from Microsoft.ML.  
  The model is trained once in memory (singleton pattern) when needed.

**System Functionality**

1. Model Training (in-memory)

* Input: list of favorites (user–product pairs)
* Algorithm: MatrixFactorizationTrainer
* Parameters:
  + NumberOfIterations = 20
  + ApproximationRank = 100

2. Recommendations

* If the user has favorites → system predicts a score for each product.
* If no favorites → system calculates the average product rating from reviews and returns top-rated ones.

Recommended products are shown on the Home screen, under the “Recommended for you” section.

**Database Tables Used**

* Favorites – stores user favorite products
* Products – list of all available products
* Reviews – ratings given by users to products

**Key Benefits**

* Personalized suggestions for each user
* Fallback logic for new users (top-rated products)
* Integrated with other app features (e.g., discount notifications, SignalR alerts)
* Easy to extend and fast to train

**Frontend Integration**

* Component: HomeScreen in the Flutter app
* Section: “Recommended for you”
* API Call: ProductProvider.getRecommendedProducts(userId) → GET /Product/{userId}/recommend
* UI: Horizontal list of product cards (image, name, price, favorite button)

**Code Locations**

* Backend: ProductService.cs → method Recommend(userId)
* Controller: ProductController.cs → [HttpGet("{userId}/recommend")]
* Frontend: HomeScreen.dart → \_recommendedProducts list + horizontal UI scroll

A screen shot of a phone

AI-generated content may be incorrect. A screen shot of a cell phone

AI-generated content may be incorrect.

Button hide/show omogućava korisniku da prilagodi po svojoj želji vizuelni pristup aplikaciji. Hide se koristi da bi sakrio prikaz preporučenih proizvoda, dok show omoćava prikaz istih.  
  
A screen shot of a phone

AI-generated content may be incorrect.