**Recommendation System – Glam Heel Hangout**

The recommendation system suggests products to users based on their **saved favorites** (user–product pairs).  
If recommendations can’t be generated (e.g., the user has no favorites, there isn’t enough data to train, or model training fails), the system **falls back** to showing **top-rated products** calculated from user reviews.  
The goal is a lightweight, personalized experience using **ML.NET** with the **Matrix Factorization** algorithm.

**Implementation Overview**

Key Classes and Components

* **ProductService.Recommend(userId)**  
  Main method that returns up to **5** recommended products using a trained in-memory model.  
  Excludes products the user already has in **Favorites**.  
  Falls back to top-rated products when needed.
* **ProductEntry**  
  Mapping for user–product interaction:
* UserId
* ProductId
* Label (always 1f, since only positive pairs—favorites—are used)
* **ProductScore**  
  Prediction output with a Score for each *(user, product)* pair.
* **ML.NET Model**  
  Uses MatrixFactorizationTrainer (Microsoft.ML).  
  Trained **once in memory** (static singleton pattern) on first use.

**System Functionality**

1. **Model Training (in-memory)**

* **Input:** all entries from **Favorites** where product isn’t deleted
* **Algorithm:** MatrixFactorizationTrainer
* **Parameters:**
  + - NumberOfIterations = 20
    - ApproximationRank = 100
* **Early fallback conditions:**
  + - Fewer than **2 distinct** *(user, product)* pairs
    - Any exception during Fit

1. **Recommendations**

* If the **user has favorites** and a trained model exists:
  + - Predict a score for **each non-deleted product**
    - Sort by score (desc), **exclude already-favorited products**, take **top 5**
* Otherwise (**no favorites**, **no/failed model**, or **insufficient data**):
  + - Compute average rating from **Reviews** and return **top-rated** products
* For all returned products, if an **active discount** exists, include:
  + - DiscountPercentage and computed DiscountedPrice

**UI placement:** shown on the Home screen under **“Recommended for you.”**

**Database Tables Used**

* Favorites – stores user favorite products
* Products – list of all available products
* Reviews – ratings given by users to products
* Discounts — to enrich recommended results with active discounts

**Key Benefits**

* Personalized suggestions for each user
* Robust fallback for new users or low-data situations (top-rated)
* Fast, simple, in-memory training; easy to extend
* Results enriched with active discout

**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.

The hide/show button allows the user to customize the visual appearance of the application according to their preference. Hide is used to remove the display of recommended products, while Show enables their display.  
  
A screen shot of a phone

AI-generated content may be incorrect.