|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [product\_recommendations.py]** | | |
| **Summary***:*  Analyzes user purchase history and current product details to generate personalized product recommendations on Razda Market. By leveraging purchase patterns and related products, this module increases cross-selling opportunities and enhances user engagement. | | |
| ***Processes*** | | |
| * **Fetching Purchase History:** | **The get\_user\_purchase\_history() function retrieves all completed purchases for a specified user, providing a foundation for recommendations by identifying previously bought products.** | |
| * **Identifying Related Products:** | **The get\_related\_products() function identifies products within the same category as a given item, generating complementary or similar recommendations.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **Database (razda\_market\_db)**: Fetches user purchase history and product details from the database tables (e.g., orders, order\_items). | * **Logs (product\_recommendations.log)**: Logs each recommendation generated, including any errors encountered for debugging. | |
| * **Environment Config (.env)**: Loads database connection settings securely to ensure proper access. | * **Frontend Templates**: Displays recommendations on user dashboards and product pages to encourage additional purchases. | |
| **Expected input into file:** | | **Expected output from file:** |
| * **User ID**: Required to retrieve purchase history and identify products for recommendations. * **Product ID and Limit**: When fetching related products, limits the number of results to avoid overloading pages. | | * **List of Recommended Products**: Returns a list of products relevant to the user, based on their browsing and purchase history. * **Logging Messages**: Provides feedback on success or failure of recommendation generation for each user. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **- Real-Time Data Analysis: Analyzes recent purchase patterns and user browsing history to offer relevant, timely recommendations.** |  |  | | --- | | **- Category-Based Matching: Finds related products within the same category, enhancing relevance in cross-selling strategies.** |  |  | | --- | | **- Data Privacy: Only leverages purchase and browsing history without exposing sensitive user data, ensuring privacy.** |  |  | | --- | | **- Audit Trail: Records all actions to track recommendation generation, allowing for review and optimization over time.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| | **Possible Enhancements:** | | --- |  |  | | --- | | - **Incorporate Browsing History**: Expand the recommendation algorithm to include recent browsing activity alongside purchase history. |  |  | | --- | | - **Use Machine Learning**: Implement collaborative filtering or content-based filtering for more advanced, predictive recommendations. |  |  | | --- | | - **Dynamic Inventory Check**: Exclude out-of-stock items to ensure recommendations reflect currently available products. |  |  | | --- | | - **Optimize for Similar Products**: Introduce similarity scores based on product attributes (e.g., style, price range) for fine-tuned recommendations. | | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |