|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [database\_initialization.py]** | | |
| **Summary***:*  This script is responsible for initializing the Razda Market database. It creates necessary tables and columns, ensuring all essential components exist. The module performs error handling, logs operations, and dynamically manages column additions to avoid schema conflicts. | | |
| ***Processes*** | | |
| * **Database & Table Creation** | * **Creates all essential tables if not already present. The initialize\_database() function calls create\_database() if the specified database doesn't exist and iterates through TABLES to create required tables with detailed fields.** | |
| * **Schema Evolution with Column Updates** | * **Uses add\_missing\_columns() to check for and add missing columns in existing tables. This function ensures schema consistency across all necessary tables by defining required columns in required\_columns.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **- Environment Variables: Loads from .env to get database credentials (DB\_HOST, DB\_USER, DB\_PASSWORD).** | |  | | --- | | * **- Logs (database\_initialization.log): Records success and failure messages during table creation, connection status, and column updates.** |  |  | | --- | |  | | |
| |  | | --- | | * **- MySQL Database: Executes SQL statements to initialize or modify the database schema.** | | - **Console Output**: Prints status messages (e.g., table creation success/failure) for immediate feedback. | |
| **Expected input into file:** | | **Expected output from file:** |
| |  | | --- | | * - **Environment Variables**: DB\_HOST, DB\_USER, DB\_PASSWORD, DB\_NAME. * **TABLES Dictionary**: Contains SQL statements for creating each table. |  |  | | --- | |  | | | |  | | --- | | * - **Database Initialization**: Fully created or updated database schema with all required tables and columns. * **Logs & Console Messages**: Success and error messages for each table and column creation. |  |  | | --- | |  | |
| **Things that need to be taking place:** | | |
| |  | | --- | | **- TABLES Dictionary: Defines SQL statements for all required tables, ensuring a consistent schema across the Razda Market platform.** |  |  | | --- | | **- initialize\_database(): Connects to the database, creates tables if they don’t exist, and calls add\_missing\_columns() to verify schema completeness.** |  |  | | --- | | **- add\_missing\_columns(): Checks for any missing columns in existing tables and adds them, dynamically adapting the schema as needed.** |  |  | | --- | | **- create\_database(cursor): Ensures the specified database exists or creates it if missing.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| | **Tables and Purpose:** | | --- | | - **users**: Manages user data, including login information, profile details, and status. | | - **roles, permissions, role\_permissions, user\_roles**: Define user roles and permissions, establishing a Role-Based Access Control (RBAC) system. | | - **product\_categories, products**: Organize products within categories and manage product details, prices, and inventory. | | - **orders, order\_items**: Manage order processing, including items, quantities, and order statuses. | | - **terms, user\_terms**: Handle terms of service and user acceptance records. | | - **password\_resets**: Stores password reset requests with expiration and usage status for secure recovery. | | - **site\_settings**: Allows customization of platform settings (e.g., themes, policies). | | - **Optional Tables**: **user\_activity\_logs**, **sales\_reports**, **product\_reviews**, **wishlist**, **carts** track user activities, sales, reviews, and wishlist data for improved user engagement and platform functionality. |  | **Possible Enhancements:** | | --- | | - **Schema Versioning**: Implement a versioning system to track schema updates, making it easier to manage database migrations. | | - **Automated Backup & Rollback**: Add functionality to backup data before schema changes and automate rollbacks in case of errors. | | - **Dynamic Data Validation**: Implement checks to ensure data integrity for fields like email, phone number, and date. | | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |