|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [permissions\_management.py]** | | |
| **Summary***:*  This file handles all aspects of role and permission management within the Razda Market platform. It provides functions for creating, assigning, revoking, and deleting roles and permissions for users, ensuring granular control over access to various functionalities across the system. This file is crucial for maintaining security and enforcing access control by connecting to a MySQL database to manage and retrieve role-permission associations. It also supports role-based access restrictions by authenticating administrators through environment-stored credentials. | | |
| ***Processes*** | | |
| **Role Management:** | **Functions to create, delete, and list roles in the system, including adding descriptions for each role** | |
| **Permission Management:** | **Functions to add, delete, and list permissions, providing detailed descriptions for each** | |
| **Role-Permission Assignment:** | **Assigns specific permissions to roles and revokes permissions as needed, enforcing role-based access** | |
| **User Role Assignment:** | **Assigns or removes roles from individual users to regulate their access levels** | |
| **Listing and Querying** | **Lists all roles, permissions, roles assigned to specific users, and permissions assigned to roles, providing transparency and administrative oversight** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **Environment Variables** (.env): Credentials and DB info (DB\_HOST, DB\_USER, DB\_PASSWORD, DB\_NAME, ADMIN\_USERNAME, ADMIN\_PASSWORD) | | * **Logging File**: permissions\_management.log to record all actions, warnings, and errors |
| * **Database (MySQL)**: Interacts with roles, permissions, role\_permissions, and user\_roles tables for role and permission data management | | * **Console Output**: Displays feedback messages for admin actions, role and permission listings, and errors |
| **Expected input into file:** | | **Expected output from file:** |
| * **Environment Variables**: For database connection and admin credentials.  |  | | --- | | * ```python | | * DB\_HOST, DB\_USER, DB\_PASSWORD, DB\_NAME, ADMIN\_USERNAME, ADMIN\_PASSWORD * ‘’’  |  | | --- | | * **User Input (CLI)**: CLI arguments to manage roles and permissions, requiring admin authentication. |  |  | | --- | |  | |  |  | | --- | |  | | | * **Logging Output**: Records all major actions (e.g., role creation, permission assignment) for traceability.  |  | | --- | | * ```python * logging.info("Added new role 'admin'") * ‘’’ * **Database Updates**: Executes SQL commands to modify roles, permissions, and relational tables |  |  | | --- | |  | |
| **Things that need to be taking place:** | | |
| |  | | --- | | **● Admin Authentication: Validates admin credentials to restrict role and permission modifications to authorized personnel.** |  |  | | --- | | **● Database Connection Handling: Manages MySQL connection for consistent access and error handling.** |  |  | | --- | | **● Input Sanitization: Ensures inputs are sanitized to prevent SQL injection or other attacks.** |  |  | | --- | | **● Error Logging: Captures and logs all errors and warnings, especially when actions fail due to duplicate entries or missing data.** |  |  | | --- | | **● Role-Permission Consistency Checks: Confirms the existence of roles and permissions before performing assignments or deletions.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| * - Oliver Smith (Razda Admin) Nov 8, 2024: Set up functions for role and permission management, added database and logging configurations. | | |