|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [page\_editor.py]** | | |
| **Summary***:*  Manages roles and permissions for various account types, facilitating secure access to the page editor. Supports functionality to create, edit, and delete pages based on user roles and permissions. Interfaces with permissions management and user setup files for access control. | | |
| ***Processes*** | | |
| * **Page Editor Dashboard** | **Displays the page editor dashboard, showing all pages to users with the appropriate permissions (page\_editor\_home). Uses the PermissionsManager and UserSetup to verify access rights based on user roles.** | |
| * **Create Page** | **Allows authorized users to create a new page by verifying permission through PermissionsManager and saving the page to the database (create\_page). Logs success or failure.** | |
| * **Edit Page** | **Enables users with edit permissions to modify existing pages. Verifies access via PermissionsManager, retrieves page data, and updates the page in the database (edit\_page). Logs each change.** | |
| * **Delete Page** | **Allows authorized users to delete a page after confirming permissions using PermissionsManager (delete\_page). Deletes the page from the database and logs the action.** | |
| * **Permission Check** | **Utilizes PermissionsManager to check if the user’s role has the required permissions for specific actions, ensuring access control for create, edit, and delete functions (check\_permission).** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **permissions\_management.py**: Manages permission checks based on roles, ensuring that users only access permitted functionality (e.g., view, create, edit, delete pages). | * **Database (razda\_market\_db)**: Stores page content and details for each action (create, edit, delete) and provides stored information for existing pages. | |
| * **user\_setup.py**: Retrieves user role information and setup data, verifying the user's identity and account type before providing access to page functions. | * **Logs (page\_editor.log)**: Logs actions taken by users, such as page creation, updates, and deletions, along with access attempts and permission denials. | |
| * **alert\_automation.py**: Sends alerts in cases of permission issues or errors encountered during page management (e.g., failed create, edit, delete operations). |  | |
| **Expected input into file:** | | **Expected output from file:** |
| * **User Session Data**: Includes user\_id and user role, retrieved via user\_setup.py, to identify and verify the user before granting access. * **Page Data**: Input data such as title and content for creating or editing a page, provided via form submission. * **Permission Checks**: Verifies user roles and permissions, provided by PermissionsManager, for each operation (view, create, edit, delete). | | * **Rendered Templates**: Displays templates for the page editor dashboard, create page form, edit page form, and error handling messages. * **Access Alerts**: Notifies users of success or failure in accessing and managing pages, with details on any permission issues or errors. * **Log Entries**: Records each action or access attempt, including success, failure, and errors, ensuring traceable history for auditing. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **- Role-Based Access Control: Ensures that only users with the correct roles (admin, sub-admin, market admin) can access, create, edit, or delete pages.** |  |  | | --- | | **- Data Validation: Validates that the title and content fields are not empty during page creation or editing to maintain data integrity.** |  |  | | --- | | **- Comprehensive Logging: Logs each action taken within the page editor, providing an audit trail for security and transparency.** |  |  | | --- | | **- Secure Alerts and Notifications: Sends alerts to admins or relevant users when unauthorized access attempts occur or errors are encountered.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| | **Possible Enhancements:** | | --- |  |  | | --- | | - **Enhanced Role Customization**: Provides more granular permissions within roles, such as differentiating between basic and advanced admin functions. |  |  | | --- | | - **Preview Mode for Page Edits**: Adds a preview feature, allowing users to view changes before finalizing edits or creation. |  |  | | --- | | - **Detailed Change Logs**: Stores detailed change history for each page to track modifications and revert to previous versions if needed. |  |  | | --- | | - **Automated Error Reporting**: Implements automated error reporting to notify developers or admins of frequent issues in real-time. | | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |