|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [user\_onboarding.py]** | | |
| **Summary***:*  The user\_onboarding.py script provides a guided onboarding process for new users joining Razda Market. It integrates with profile setup and user signup functionalities, progressively guiding users through setting up their profile, introducing key marketplace features, and providing tips for effective usage. This structured onboarding experience helps new users familiarize themselves with the platform, improving engagement and retention | | |
| ***Processes*** | | |
| * **Environment Setup and Imports** | **Loads environment variables from .env for secure access to database credentials. Imports necessary modules for routing (flask), logging, and database management. The dotenv library ensures secure handling of environment variables, while logging provides traceability for actions taken during onboarding.** | |
| * **Blueprint Initialization** | **Sets up the user\_onboarding\_bp Flask blueprint, encapsulating all routes and views specific to the onboarding process. This structure improves modularity, making onboarding functionalities easy to maintain and extend.** | |
| * **Onboarding Step Navigation:** | **Manages each step of the onboarding process with session tracking (session['onboarding\_step']). Steps include profile setup, feature introduction, marketplace tips, and completion. Users are directed to appropriate templates, and actions are logged to track their progress.** | |
| * **Database Connection Function:** | **Contains connect\_db() to establish and return a secure database connection. This function encapsulates the database logic for reuse, enabling error logging and graceful handling if a connection issue occurs.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **.env file**: Contains sensitive database credentials for secure connection setup. Environment variables such as DB\_HOST, DB\_USER, and DB\_PASSWORD are loaded for database configuration. | * **Log file (user\_onboarding.log)**: Logs user progress, warnings, and errors encountered during onboarding. This file records actions (e.g., step completion), making it easier to trace issues and analyze user onboarding behavior. | |
| * **Setup/profile\_setup.py**: Imports initialize\_profile, enabling profile setup for new users. | * **Redirect to Homepage (main.home)**: After onboarding, users are redirected to the main homepage, indicating successful onboarding completion. | |
| * **Setup/user\_signup.py**: Imports register\_user to create new user accounts during onboarding. |  | |
| **Expected input into file:** | | **Expected output from file:** |
| * **Username, Email, Password, Profile Data**: User-provided information during registration and profile setup. **Username**, **Email**, and **Password** are required for account creation; **Profile Data** includes optional fields for preferences and other setup parameters. | | * **User Registration Record**: Successful onboarding creates a new user entry in the database and updates user preferences in user\_profiles. The log file documents onboarding completion and progress through steps. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **1. Environment Variable Security: Confirm that all necessary environment variables are present in .env, including database credentials. Ensure secure storage of the .env file, as it contains sensitive information.** |  |  | | --- | | **2. Logging Verification: Test that logging in user\_onboarding.log correctly captures each action (e.g., step completion) and any errors encountered, such as registration issues or failed database connections.** |  |  | | --- | | **3. Database Connection Pooling: If applicable, confirm that the database configuration aligns with connection pooling, improving resource efficiency and scalability in high-demand situations.** |  |  | | --- | | **4. Session Management: Validate that session['onboarding\_step'] accurately tracks user progress, allowing users to resume from the correct step if interrupted. Verify that session data is cleared upon onboarding completion.** |  |  | | --- | | **5. Error Handling and Flash Messages: Ensure that each route properly handles exceptions (e.g., database issues), displays user-friendly flash messages, and logs details for troubleshooting.** |  |  | | --- | | **6. Template Routing and Rendering: Check that all routes correctly link to templates (onboarding.html, onboarding\_step1.html, etc.) and render them based on session progress. Ensure consistency between step numbers and their assigned templates.** |  |  | | --- | | **7. User Data Security: Securely handle all user data, including form input for registration and profile setup. Prevent session hijacking and verify that users cannot skip steps or modify session data to access steps prematurely.** |  |  | | --- | | **8. Profile Setup and Preference Storage: Confirm that initialize\_profile and preference updates in user\_profiles function correctly, storing each user’s unique preferences. Ensure that the update\_preferences SQL statement executes as expected.** |  |  | | --- | | **9. Final Step Confirmation: Test the final onboarding step, ensuring that it redirects users to the homepage and displays a success message. Confirm that session data is cleared upon completion.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |