|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [user\_signup.py]** | | |
| **Summary***:*  The user\_signup.py script provides functionality for user registration within Razda Market. It validates user input, checks for duplicate entries, and securely stores new user data in the database. The script leverages environment variables, email validation, password hashing, logging, and rate limiting for robust security and reliability in the user signup process. | | |
| ***Processes*** | | |
| * **Environment Setup and Imports** | **Loads essential environment variables from a .env file using dotenv. Imports necessary modules for database management (mysql.connector), email validation (email\_validator), password hashing (bcrypt), and logging (logging).** | |
| * **Database Connection Pooling:** | **Configures a MySQL connection pool to optimize database connectivity, allowing efficient reuse of connections, reducing latency, and supporting concurrent user registration requests. Includes error handling to log issues and exit if the pool cannot be created.** | |
| * **Input Validation:** | **Contains functions to validate usernames and passwords. Includes regular expressions to enforce username format and password complexity, providing basic sanitization to mitigate injection risks. Logs any warnings for non-compliant input.** | |
| * **User Registration:** | **Registers a new user in the system. It validates input, checks for duplicates, hashes passwords, and inserts user data into the database. Logs actions and errors, including validation failures and database errors, to provide feedback for troubleshooting.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **.env file**: Loads environment variables (e.g., database credentials) for secure configuration without hardcoding sensitive information. | * **Log file (app.log)**: Logs system actions, warnings, and errors for auditing and debugging. Logs include timestamps, log levels, and messages describing each event or issue. | |
| **Expected input into file:** | | **Expected output from file:** |
| * **Username, Email, Password, First and Last Name**: Input fields for user registration. **Username** and **Email** are mandatory fields checked for format and uniqueness. **Password** must meet complexity requirements. **First Name** and **Last Name** are optional fields sanitized and stored in the database if provided. | | * **Database Record for New User**: Upon successful registration, a new user entry is added to the users table with fields including username, email, hashed password, first\_name, and last\_name. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **1. Environment Variable Loading: Confirm that the .env file contains necessary credentials (e.g., DB\_HOST, DB\_USER, DB\_PASSWORD, DB\_NAME, LOG\_FILE). Ensure variables are securely loaded to avoid runtime errors.** |  |  | | --- | | **2. Logging and Monitoring: Verify that logging captures key actions, warnings, and errors in app.log. Regularly review logs for anomalies, such as excessive invalid login attempts, which may indicate abuse.** |  |  | | --- | | **3. Database Connection Pooling: Test that the connection pool (razda\_pool) functions as expected, efficiently handling multiple concurrent requests without exhausting resources or causing delays.** |  |  | | --- | | **4. Input Validation Accuracy: Check that username and password validation meet project security standards. Validate sanitize\_input to ensure it removes potentially harmful characters without over-sanitizing input.** |  |  | | --- | | **5. Secure Password Handling: Ensure passwords are hashed with bcrypt before database insertion, and confirm that stored hashes can withstand common cracking methods.** |  |  | | --- | | **6. Rate Limiting Functionality: Test the RateLimiter for various scenarios (e.g., normal usage, excessive attempts) to confirm it accurately limits attempts without hindering legitimate users.** |  |  | | --- | | **7. Error Handling and User Feedback: Ensure that errors (e.g., invalid email format, database issues) are handled gracefully with informative messages, logging details in app.log for troubleshooting.** |  |  | | --- | | **8. Testing of Registration Process: Validate the end-to-end registration process, confirming that new user records are accurately created in the database, duplications are prevented, and security checks are enforced.** |  |  | | --- | | **9. Secure File Handling: Ensure the .env file is protected and not exposed to unauthorized users, as it contains sensitive information. Validate that app.log is stored in a secure location with controlled access.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |