**Iteration 2**

**1. Insecure Password Storage:**

* **Plaintext passwords:** The most significant vulnerability is storing passwords in plain text. This makes the system highly susceptible to data breaches. If the user\_data.json file is compromised, attackers can gain access to all user passwords.
* **Lack of password hashing:** Password hashing is a cryptographic technique that transforms passwords into a one-way hash, making them irreversible. This prevents attackers from obtaining the original password even if the hash is compromised.

**2. Insecure Data Transmission:**

* **Plaintext transmission:** Passwords are transmitted in plain text over the network. This makes them vulnerable to interception by attackers.

**3. Insufficient Input Validation:**

* **Injection attacks:** The code doesn't validate user input, making it vulnerable to injection attacks like SQL injection or cross-site scripting (XSS).

**4. Lack of Session Management:**

* **Session hijacking:** The code doesn't implement proper session management, which can lead to session hijacking attacks.

**5. Missing Error Handling:**

**Information leakage:** The code doesn't handle errors gracefully, which can potentially leak sensitive information to attackers.