Vulnerabilities

1. **SQL Injection**:
Since user input was not validated, this vulnerability was present in the login page. It can be removed by white listing the input fields.

2. **Cross site scripting**:
This vulnerability is present because of the php scripts present. These scripts are used to fetch data from from database or to insert data to the database.
Attacker can use malicious scripts and can steal confidential information.
This vulnerability can be removed using output encoding.

3. **Broken Authentication and Session management**:
If the browser is closed abruptly without logging out of the application, the session variables won't die. Attacker using the same machine can access these session cookies and login into the application.
This vulnerability can be removed by having a timeout for session variables.

4. **Insecure cryptographic storage**:
Since the confidential information provided by the user is stored directly into the database without using any encryption. By using this vulnerability, an attacker can steal, modify such weakly protected data.
This vulnerability can be easily removed by using appropriate hash functions.

5. **Insecure direct object references**:
Since the user id is available in the URL, the attacker can easily change the user id and access another person's account.
This vulnerabilities can be easily removed by avoiding exposing object references in URLs.

Some more vulnerabilities like insufficient transport layer protection exists. This vulnerability is not a part of the web application but can be there because of the data sent over the network and not implementing proper security protocols like SSL.