**Stored Procedure Documentation**

**1. ChangeUserPassword.sql**

**Purpose:**  
This procedure is designed to **safely update the password** for a user in the Users table. It takes the user’s ID and the new password as input and updates the record.

**Description:**  
The procedure is helpful in allowing users to change their passwords when needed. Once executed, it confirms the action using a PRINT statement, which could be useful for front-end acknowledgment.

**Output when tested:**  
Upon execution, it displays:

Password updated successfully.

**2. DeleteCategory.sql**

**Purpose:**  
This procedure **deletes a category** from the Categories table based on the provided CategoryID.

**Description:**  
This is typically used in an admin panel where categories need to be managed. It ensures the deletion of unwanted or deprecated categories. It prints a message on successful deletion.

**Output when tested:**

Category deleted successfully.

**3. DeleteSkill.sql**

**Purpose:**  
This procedure **removes a skill** from the Skills table by specifying the SkillID.

**Description:**  
Used when an obsolete or incorrect skill needs to be removed. It's simple, to-the-point, and provides a confirmation message.

**Output when tested:**

Skill deleted successfully.

**4. GetUserByID.sql**

**Purpose:**  
This procedure **retrieves user details** from the Users table based on the given UserID.

**Description:**  
Useful for front-end or admin views where user profiles or account details are to be shown. Instead of returning a message, it returns a row of data from the Users table.

**Output when tested:**  
Returns user record (UserID, Name, Email, etc.)  
No PRINT statement is used; the output is the user data itself.

**5. UpdateCategoryName.sql**

**Purpose:**  
This procedure **updates the name** of a category using the CategoryID.

**Description:**  
When a category name needs correction or rebranding, this procedure allows quick updates. It is minimal and useful for frontend control panels.

**Output when tested:**

Category name updated successfully.

**6. UpdateLastLogin.sql**

**Purpose:**  
This procedure **updates the LastLogin column** in the Users table for a specific user.

**Description:**  
It can be triggered after user login to update the login timestamp, helpful for user activity tracking. The procedure uses GETDATE() to insert the current date and time.

**Output when tested:**

Last login updated successfully.

**7. UpdateSkillDescription.sql**

**Purpose:**  
This procedure **updates the description** of a specific skill in the Skills table.

**Description:**  
It helps in maintaining and refining the description text for skills shown to clients or freelancers. The update is followed by a confirmation message.

**Output when tested:**

Skill description updated successfully.

Let me know when you're ready for the next batch or if you'd like similar documentation for other procedures or triggers.