Project Documentation: Linux Test System Script

Introduction

This script provides a simple, interactive test system for users to sign up, sign in, take tests, and view results. It logs all activities, handles timeouts, and stores answers in CSV format. The script utilizes a question bank and allows users to view their past test results.

Features

- **Sign Up & Sign In**: Users can create accounts with alphanumeric user IDs and strong passwords, or log in using their credentials.
- **Test System**: Users can take a test with randomly selected multiple-choice questions, answer them within a set time, and view their results afterward.
- **Logging**: All activities are logged with timestamps in the `test_activity.log` file.
- **Timeout Handling**: Each question has a 10-second time limit, and users can face a timeout if they take too long to answer or provide no input.
- **Question Bank**: The script allows for a customizable question bank, defaulting to `question_bank.txt`.

Requirements

- **Linux OS** with Bash
- **User Credentials File**: Stores user credentials (`.user_credentials.csv`).
- **Question Bank File**: A file containing test questions (`question_bank.txt`).
- **Log File**: Logs all activities (`test_activity.log`).
- **Answer File**: Stores answers to questions (`TestData/answer_file.csv`).

File Structure

- **USER_CREDENTIALS_FILE**: `.user_credentials.csv`
- **LOG_FILE**: `test_activity.log`
- **ANSWER_FILE**: `TestData/answer_file.csv`
- **QUESTION_BANK_FILE**: `question_bank.txt`

Script Breakdown

Variables

- `USER_CREDENTIALS_FILE`: Path to the file that stores registered user credentials.

- `LOG_FILE`: Path where activities are logged.
- `ANSWER_FILE`: Stores answers to the test taken.
- `QUESTION_BANK_FILE`: The file containing the questions for the test.
- `TIMEOUT`: The duration of the timeout (10 seconds).
- `USER_DIR`: Path to the user's home directory.
- `TEST_DATA_DIR`: Path to the `.TestData` directory in the user's home directory.

Functions

1. log()

- **Purpose**: Logs activities with a timestamp into the `test_activity.log`.
- **Usage**: Logs activities like user sign-in, taking tests, viewing results, etc.
- **Parameters**: `activity` - The activity to be logged.

```
log "User signed in: $sign_in_id"
```

2. answer_file_creation()

- **Purpose**: Ensures the creation of the answer file and test data directory if they don't exist. Backs up the previous answer file if necessary.
- **Usage**: Runs during the initialization phase to prepare the testing environment.

```
answer_file_creation
```

3. menu_header()

- **Purpose**: Displays the main menu with options for signing in, signing up, or exiting.
- **Usage**: Called at the beginning of the script to display the welcome menu.

```
menu_header
```

4. view_test_screen()

- **Purpose**: Displays the user's past test results (answers and time taken).
- **Usage**: Runs when the user selects the option to view their test results.

```
view_test_screen
```

5. test_screen()

- **Purpose**: Handles the test-taking process. It displays a question from the question bank, accepts user input, and records the answer along with the time taken.

- **Usage**: The function runs when the user chooses to take a test.

```
test_screen
```

6. test_menu()

- **Purpose**: Displays the test-related menu options: take a test, view results, or go back to the main menu.
- **Usage**: Activated after successful sign-in.

```
test_menu
```

7. sign_in()

- **Purpose**: Prompts the user to enter their credentials for signing in.
- **Usage**: Allows already registered users to log in using their user ID and password.

```
sign_in
```

8. sign_up()

- **Purpose**: Handles the user registration process. Prompts the user for a valid ID, password (with constraints), and validates the user input.
- **Usage**: Runs when the user selects the sign-up option.

```
sign_up
```

9. main Script Loop

- The script continuously presents the main menu, allowing the user to sign up, sign in, or exit the program.
- **Usage**: This loop runs until the user exits the program.

```
while true; do
    menu_header
    read user_choice
    case $user_choice in
        1) sign_in ;;
        2) sign_up ;;
```

```
        3) exit 0 ;;
        *) echo "Invalid choice. Please select a valid option!" ;;
    esac
done
```

Timeout Mechanism

- Every question has a 10-second timer.
- If the user doesn’t answer within the time limit, the script registers the answer as "No Answer" and moves on to the next question.

Log File

- All activities (such as signing in, taking tests, and viewing results) are logged with timestamps in `test_activity.log`.

Security Considerations

- **Password Handling**: Passwords are read securely using `read -s` to hide the input.
- **Credential Matching**: User passwords are matched in plain text. For enhanced security, it is recommended to use a hashing mechanism (e.g., SHA-256) to store and verify passwords.

Error Handling

- The script checks for valid input in various places:
- Ensures the user ID only contains alphanumeric characters.
- Validates the password to meet specified criteria (at least 8 characters, containing numbers and symbols).
- Ensures the user re-enters the password correctly during sign-up.

Backup Mechanism

- If the answer file already exists, it is backed up before creating a new one.

Usage Example

Step 1: Running the Script

Execute the script from the terminal:

```
bash test_script.sh
```

Step 2: Sign Up or Sign In

- **Sign Up**: Choose option `2` to create a new account.
- Enter a user ID (alphanumeric).
- Enter a strong password (min 8 characters, with at least one number and one symbol).
- Confirm the password.
- **Sign In**: Choose option `1` to log into an existing account.
- Enter the user ID and password.

Step 3: Taking the Test

After successful sign-in, you can:

- **Take a Test**: Select option `1` to start the test. Answer questions within 10 seconds.
- **View Results**: Select option `2` to view previously answered questions and times.

Step 4: Exiting

- To exit the program, select option `3` from the main menu.

Conclusion

This shell script provides an interactive and user-friendly system for managing user credentials, taking timed tests, and viewing results. It logs all user activities, ensures security with password validation, and handles file backups.