```
import random
import time
import json
import os

# Constants
WORD_FILE = "words.json"
LEADERBOARD_FILE = "leaderboard.json"
WORD_CATEGORIES = ["Animals", "Fruits", "Colors"]  # Add more categories as needed

def load_words_from_json(file_path):
    # Load words from a JSON file and return them as a dictionary.
    with open(file_path, "r") as file:
        words_data = json.load(file)
    return words_data

def get_user_input(prompt):
    # Get user input from the terminal.
    return input(prompt)

def typing_test(username, words_data):
    # Implement the typing test.
    category = select_category()
    if category not in words_data:
        print("Invalid category. Please select a valid category.")
        return

    words = words_data[category]
    random.shuffle(words)
```

```
total_words = len(words)

print(f"\nCategory: {category}")
print("Instructions: Type the words exactly as shown. Press 'Ctrl + Q' to quit.\n")

input("Press Enter to start...")

start_time = time.time()
words_typed = 0

for word in words:
    os.system('clear' if os.name == 'posix' else 'cls')  # Clear the terminal
    print("Category:", category)
    print(f"Words Typed: {words_typed}/{total_words}")
    print(f"Current Word: {word}\n")
    user_input = get_user_input("Type the word: ")

    if user_input == "\x11":  # Ctrl + Q to quit
        break

    if user_input == word:
        words_typed += 1

end_time = time.time()
elapsed_time = end_time - start_time

if elapsed_time > 0:
    wpm = int(words_typed / (elapsed_time / 60))
else:
```

```
        wpm = 0

    print("\nTyping Test Results:")
    print(f"Username: {username}")
    print(f"Category: {category}")
    print(f"Words Typed: {words_typed}")
    print(f"Time Taken: {elapsed_time:.2f} seconds")
    print(f"Words Per Minute (WPM): {wpm}\n")

    update_leaderboard(username, wpm)

def select_category():
    # Allow the user to select a category for the typing test.
    while True:
        print("Select a category:")
        for i, category in enumerate(WORD_CATEGORIES, start=1):
            print(f"{i}. {category}")

        choice = get_user_input("Enter the category number: ")
        try:
            choice = int(choice)
            if 1 <= choice <= len(WORD_CATEGORIES):
                return WORD_CATEGORIES[choice - 1]
        except ValueError:
            pass

def update_leaderboard(username, wpm):
    # Update the leaderboard with the user's score.
    leaderboard = load_leaderboard()
```

```
    if username in leaderboard:
        if wpm > leaderboard[username]:
            leaderboard[username] = wpm
    else:
        leaderboard[username] = wpm

    sorted_leaderboard = dict(sorted(leaderboard.items(), key=lambda item: item[1], reverse=True))

    with open(LEADERBOARD_FILE, "w") as file:
        json.dump(sorted_leaderboard, file)

def load_leaderboard():
    # Load the leaderboard from a JSON file.
    if os.path.exists(LEADERBOARD_FILE):
        with open(LEADERBOARD_FILE, "r") as file:
            leaderboard = json.load(file)
        return leaderboard
    else:
        return {}

def show_leaderboard():
    # Display the leaderboard from the JSON file.
    leaderboard = load_leaderboard()
    if not leaderboard:
        print("Leaderboard is empty.")
    else:
        print("\nLeaderboard:")
        print("Username\tWPM")
        for username, wpm in leaderboard.items():
```

```
        print(f"{username}\t\t{wpm}")

def main():
    print("Welcome to the Typing Test Application!")

    username = get_user_input("Enter your username: ")

    while True:
        print("\nOptions:")
        print("1. Start Typing Test")
        print("2. Show Leaderboard")
        print("3. Exit")

        choice = get_user_input("Enter your choice: ")

        if choice == "1":
            typing_test(username, load_words_from_json(WORD_FILE))
        elif choice == "2":
            show_leaderboard()
        elif choice == "3":
            print("Goodbye!")
            break
        else:
            print("Invalid choice. Please select a valid option.")

if __name__ == "__main__":
    main()
```