# **Introduction:**

The User Management System (UMS) script is designed to manage user accounts, including adding, modifying, and removing users. It also allows for file management with ownership tracking. The script supports two types of users: regular users and administrators, with different functionalities available based on the user's role. The script is implemented in Bash and operates using a CSV file to store user data.

# **Tools used:**

* **Bash Scripting:** Bash is a Unix shell and command language used to write the script. It provides various built-in commands and utilities for file manipulation, user input, and control flow.
* **CSV File Handling:** The script uses CSV (Comma-Separated Values) files to store and manage user data. It involves reading from and writing to CSV files, parsing CSV strings, and manipulating CSV data.
* **File Management Commands:** Standard file management commands like touch, rm, mv, and grep are used for creating, deleting, and searching files and their ownership details.
* **Sleep Command:** The sleep command is used to create delays in the script, providing a more interactive and user-friendly experience with functions like sleep\_with\_dots.

# **Functionalities:**

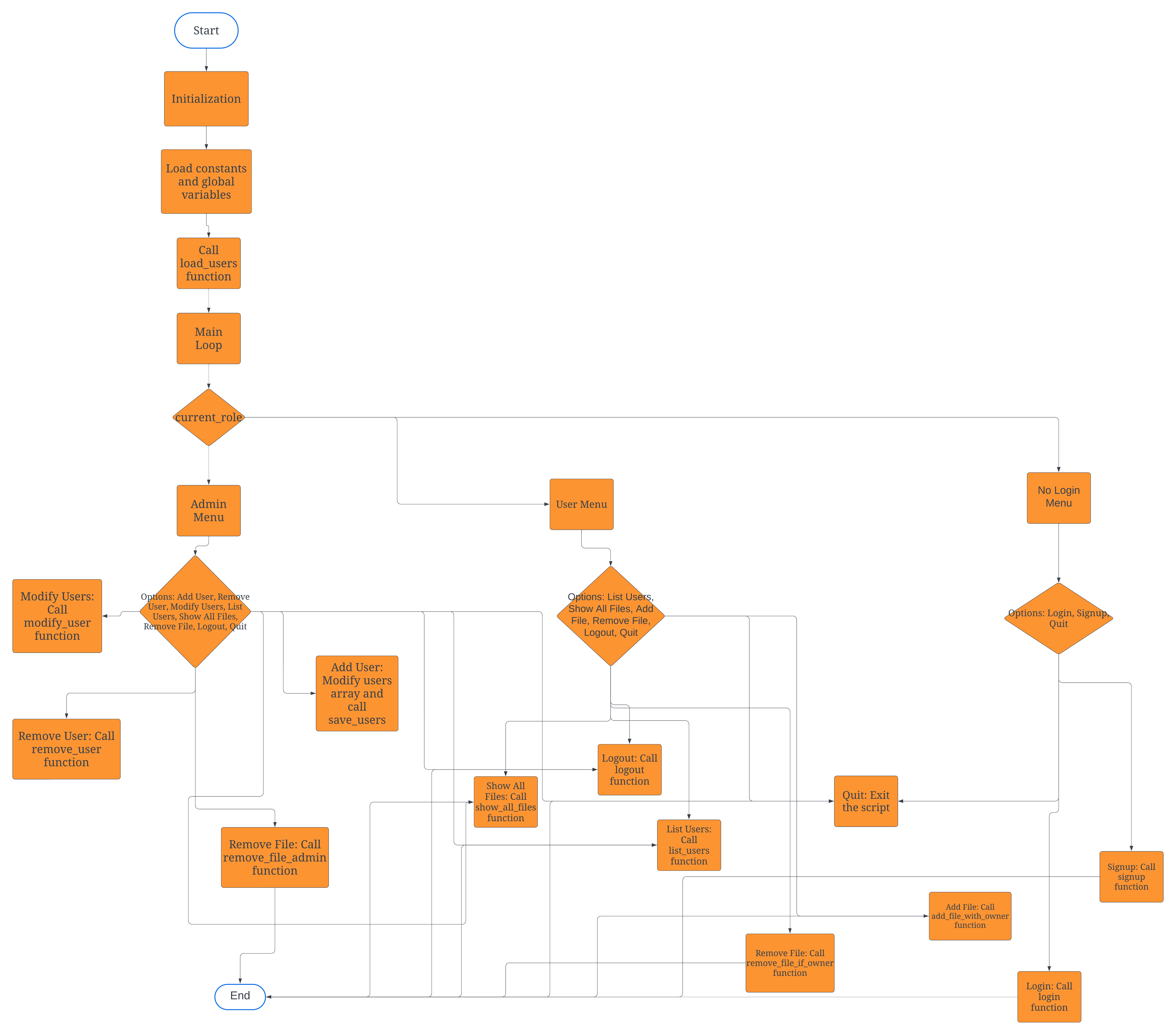
* **User Data Management**
* **Loading Users:** The load\_users function reads user data from the users.csv file into a Bash array, ensuring that the script can persist user data across sessions.
* **Saving Users**: The save\_users function writes the current user data from the Bash array back to the users.csv file.
* **CSV Parsing and Manipulation**
* **csv\_get:** Extracts a specific value from a CSV string.
* **csv\_set:** Sets a specific value in a CSV string and returns the updated CSV string.
* **User Operation**
* **Add User:** Both regular users and admins can add users through the signup function. Admins can add users through the admin menu.
* **Remove User:** Admins can remove users using the remove\_user function.
* **Modify User:** Admins can modify user details such as username, password, email, phone, and role through the modify\_user function.
* **List Users:** Depending on the user role, the list\_users function displays a list of users. Admins see more detailed user information.
* **Authentication**
* **Login:** The login function validates user credentials and sets the current user and role if successful.
* **Logout:** The logout function clears the current user and role.
* **File Management**
* **Add File:** The add\_file\_with\_owner function allows users to add files and assigns ownership to the current user.
* **Remove File:** Regular users can remove their own files using the remove\_file\_if\_owner function, while admins can remove any file using the remove\_file\_admin function.
* **Show All Files:** The show\_all\_files function lists all files along with their owners.
* **Menu System**

The script features a menu-driven interface with three main states:

* **No Login Menu:** Provides options for logging in, signing up, or quitting.
* **User Menu:** Displays user-specific options for file management and user listing.
* **Admin Menu**: Displays admin-specific options for user and file management.
* **Choice Function**

The choice function displays a menu and captures the user's selection, returning the chosen option's index.

# **Flowchart:**



# **Conclusion:**

The User Management System Bash script is a comprehensive tool for managing user accounts and files with ownership tracking. It utilizes the power of Bash scripting to provide a command-line interface for user authentication, user data management, and file operations. The script ensures data persistence through CSV file handling and offers distinct functionalities based on user roles, enhancing security and usability. This script serves as a robust foundation for further enhancements and can be integrated into larger systems requiring user and file management capabilities.