**Algorithm for Camera Rental Project**

1. Initialize the cameraList, users, and wallet data structures.
2. Add camera objects to the cameraList and user objects to the users list with initial data.
3. Display the welcome message and prompt the user to log in.
4. Validate the user's credentials by comparing them with the user objects in the users list.
5. If the credentials are valid, set the logged In User variable to the username.
6. Display the main menu options.
7. Based on the user's choice, perform the corresponding actions:

* Option 1: View and manage the user's cameras.
* Option 2: View available cameras and rent one.
* Option 3: View all cameras in the cameraList.
* Option 4: View the user's wallet balance and deposit more money if desired.
* Option 5: Exit the application.

1. In the "My Camera" section:

* Display the options: Add, Remove, View My Cameras, and Go to Previous Menu.
* Based on the user's choice, perform the corresponding actions:
* Add: Prompt the user to enter the camera details and add it to the user's camera list.
* Remove: Prompt the user to enter the camera ID and remove it from the user's camera list.
* View My Cameras: Display the details of all cameras in the user's camera list.

Go to Previous Menu: Exit the "My Camera" section and return to the main menu.

1. In the "Rent a Camera" section:

* Prompt the user to enter the camera ID they want to rent and the rental period.
* Get the camera from the cameraList based on the ID and check if it exists.
* If the camera is available, update its status to "RENTED," deduct the rent amount from the user's wallet, and add the camera to the user's camera list.
* If the rent amount exceeds the user's wallet balance, display an error message.

10. In the "View All Cameras" section, display the details of all cameras in the cameraList.

11. In the "My Wallet" section, display the user's current wallet balance and provide the option to deposit more money.

12. If the user chooses to deposit more money, prompt them to enter the amount and updatethe user's wallet balance accordingly.

13. Repeat the main menu loop until the user chooses to exit the application.