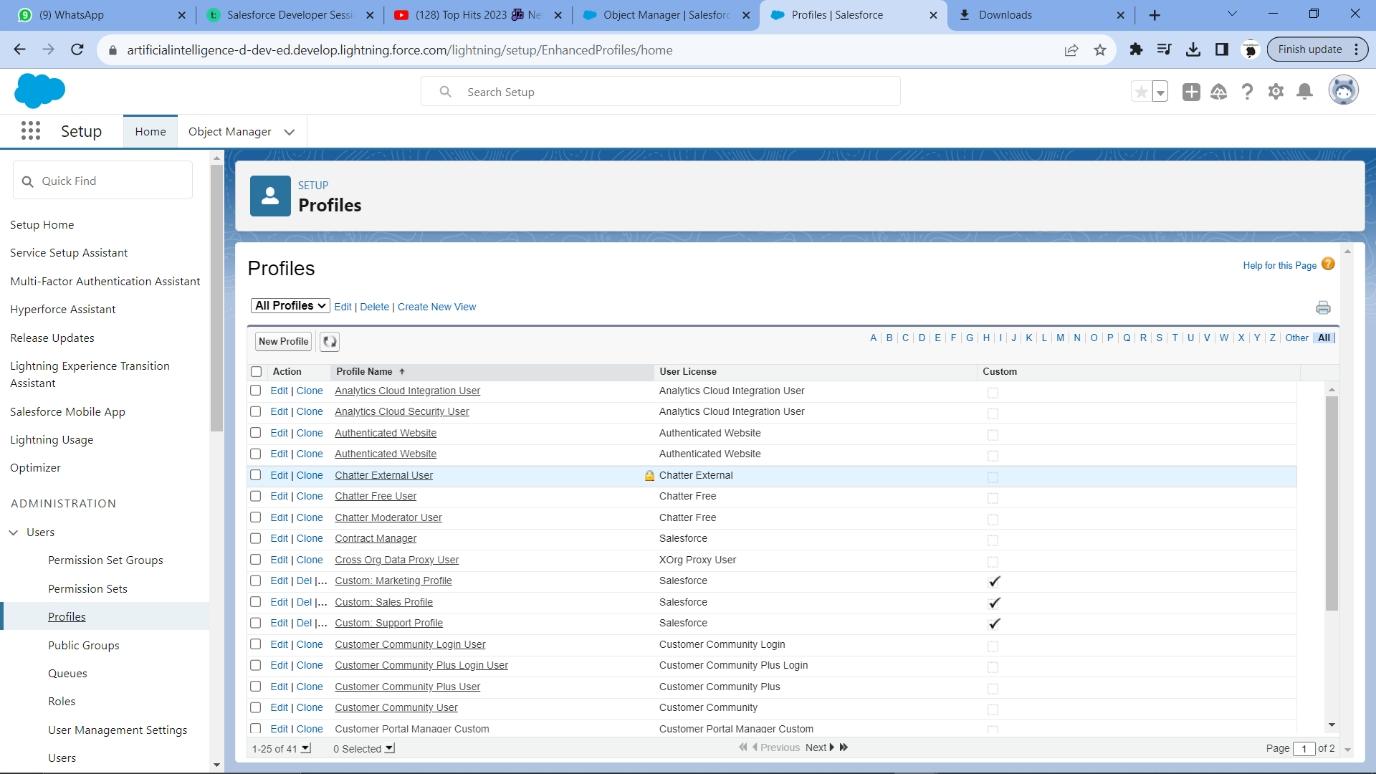
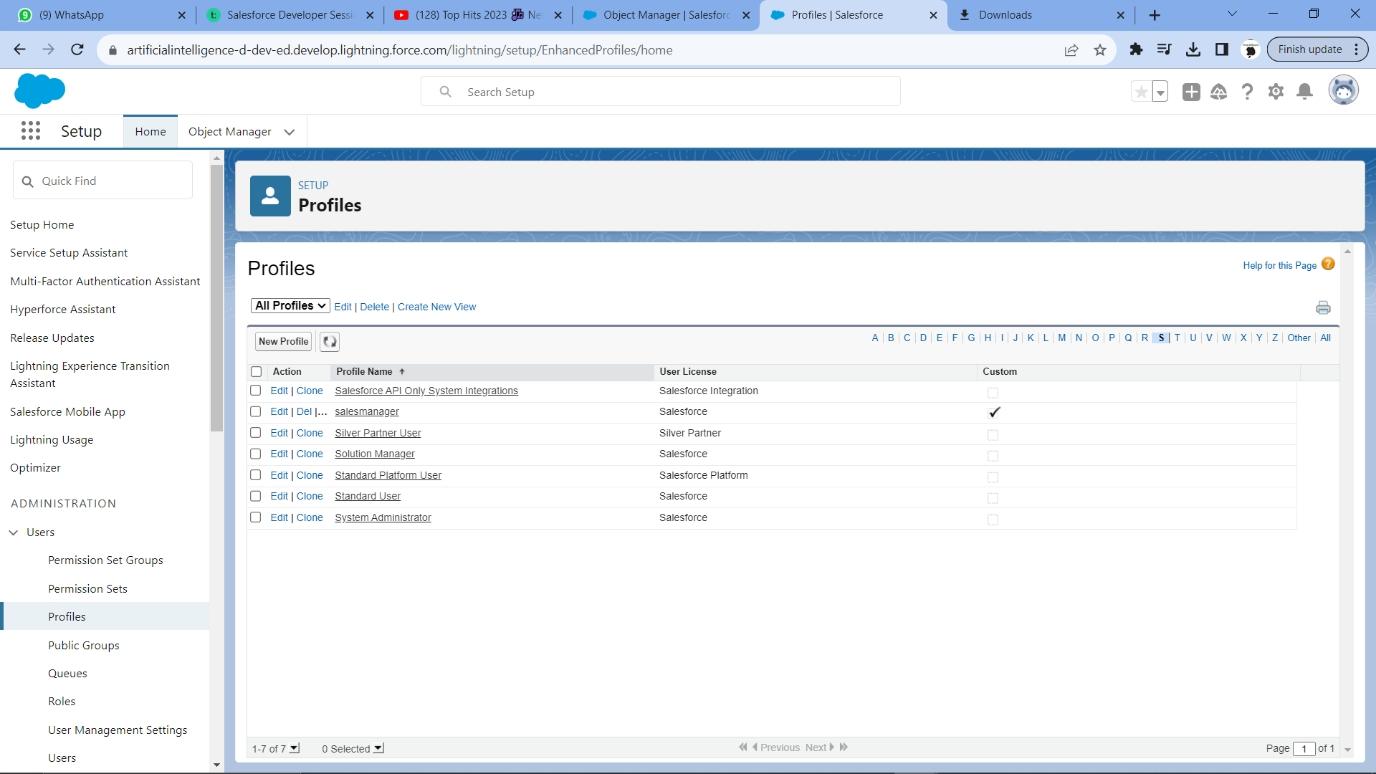
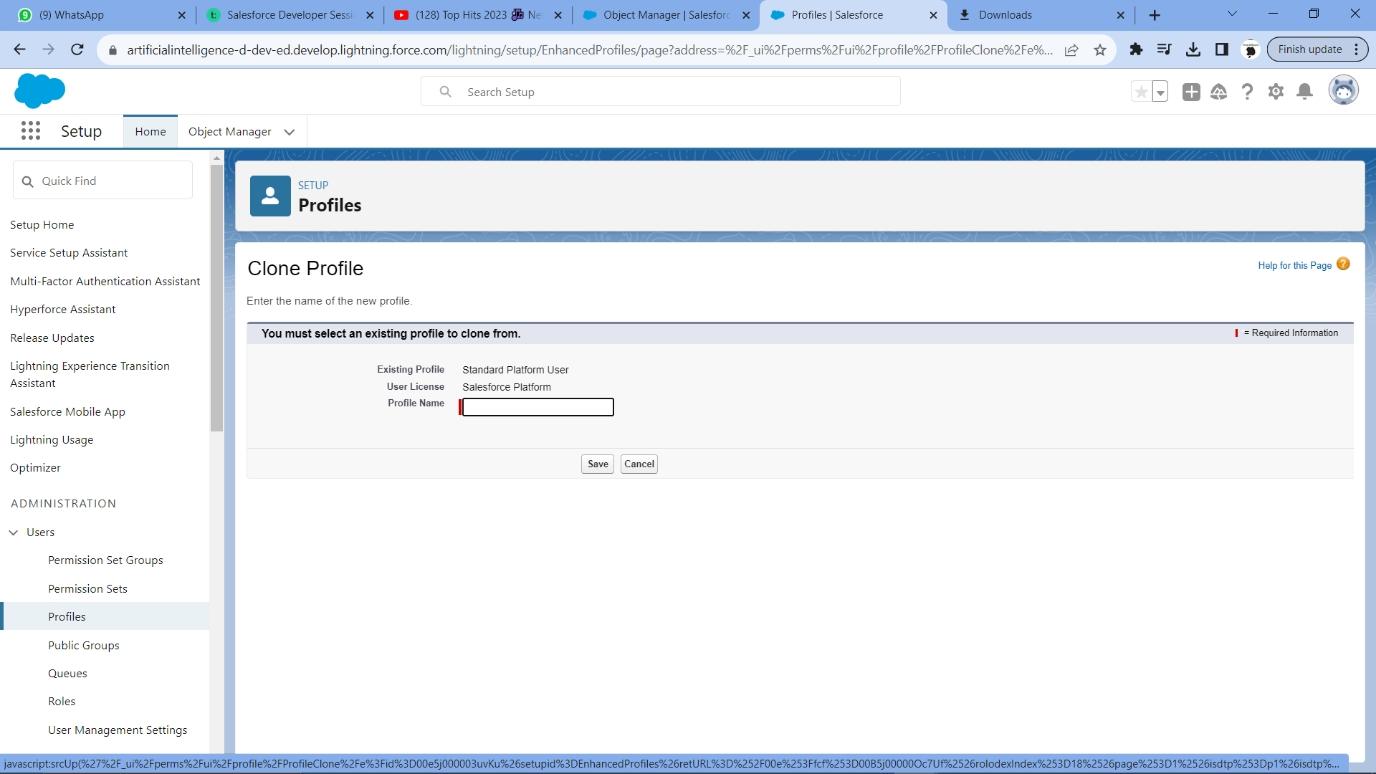
2. If there is 2 user, User A and User B in the organisation and we want in Account object that User A should not see the User B Record and user B should not see User A record then apply the Security for the users.

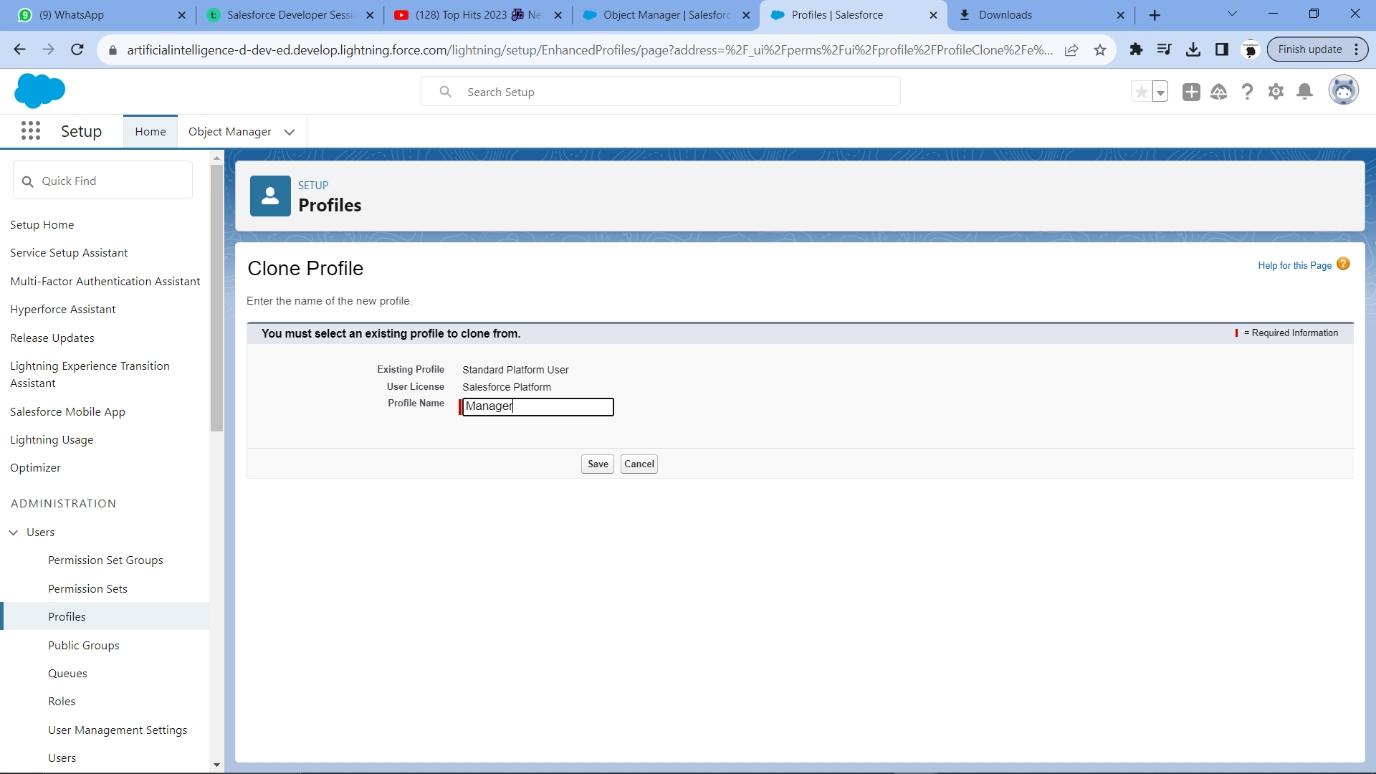
Solution:

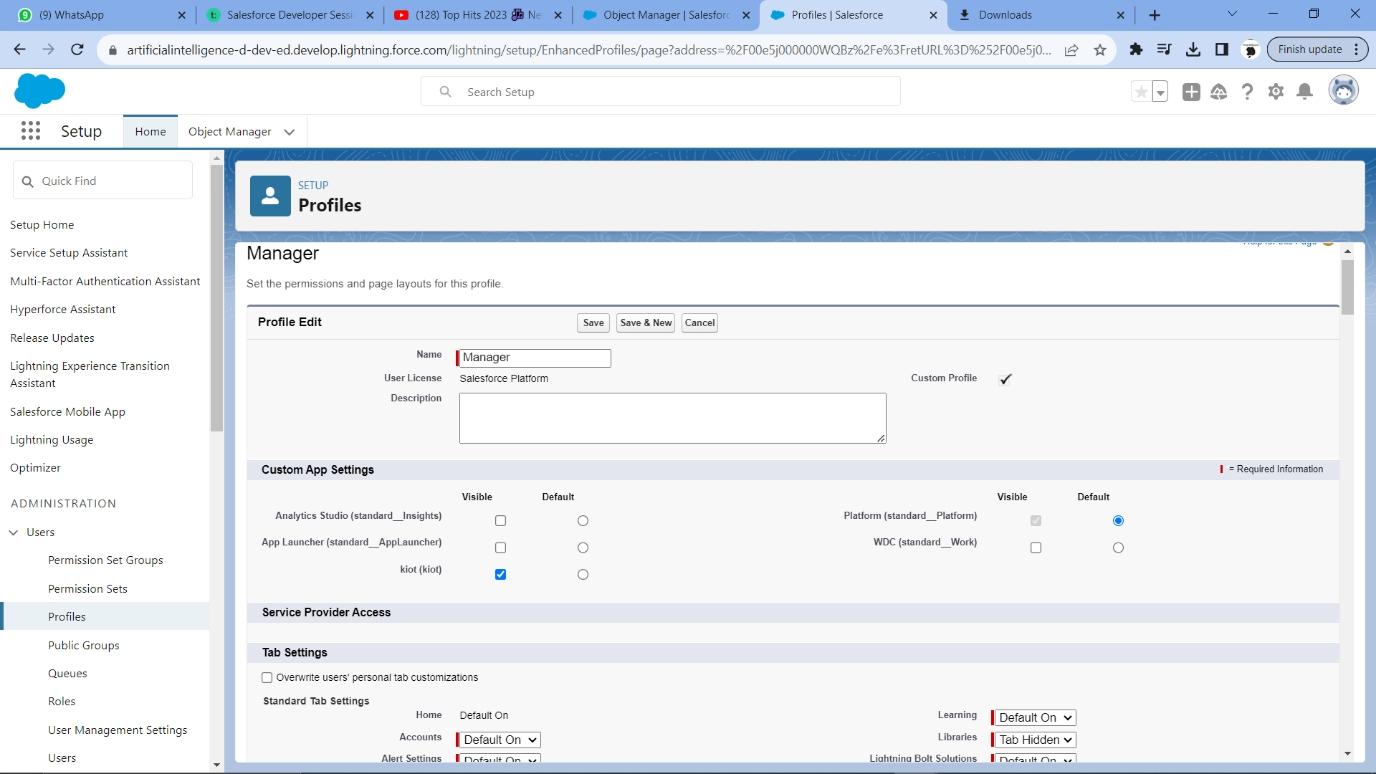
Step 1: Create two separate custom profiles, one for User A and one for User B.

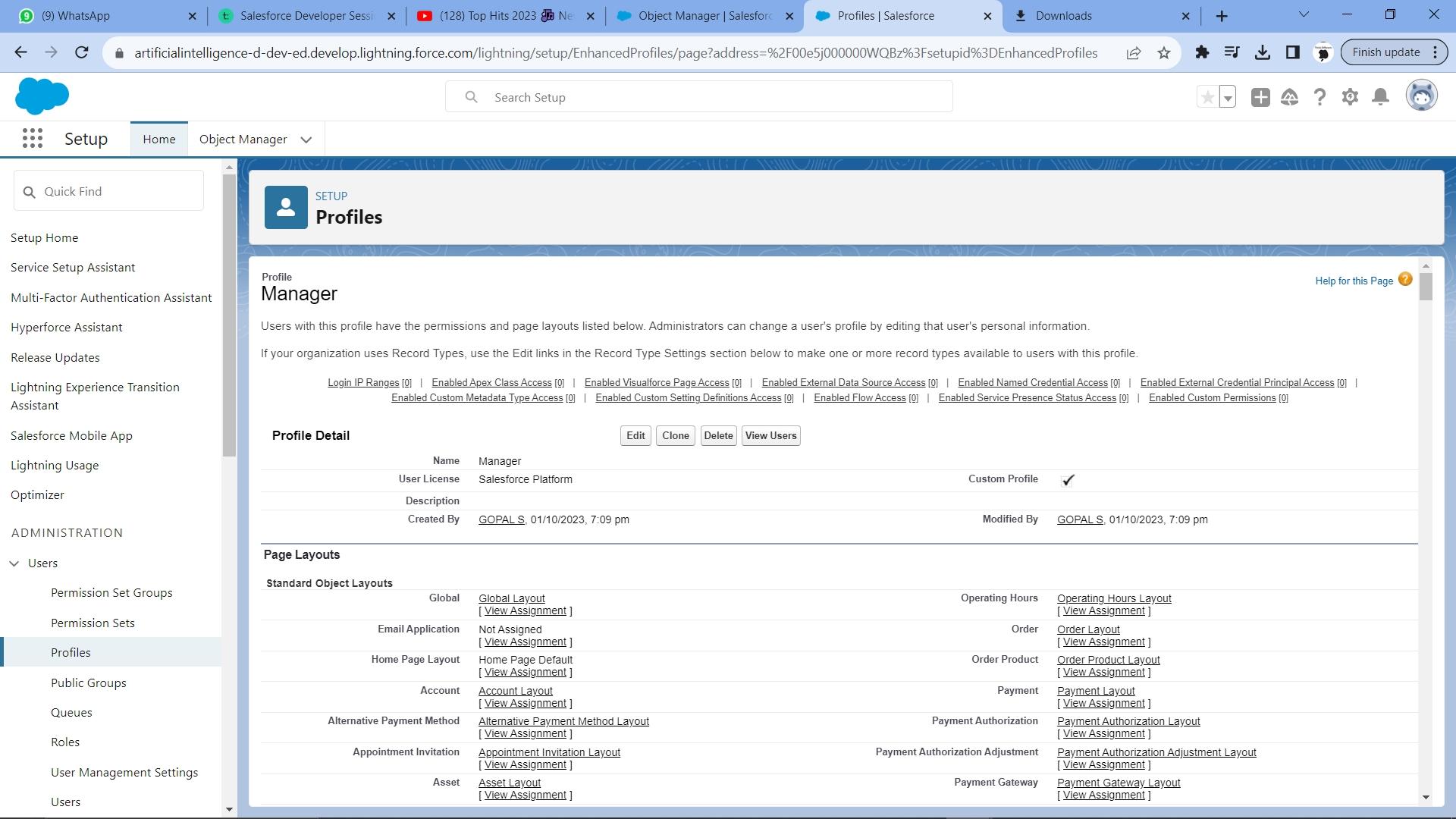


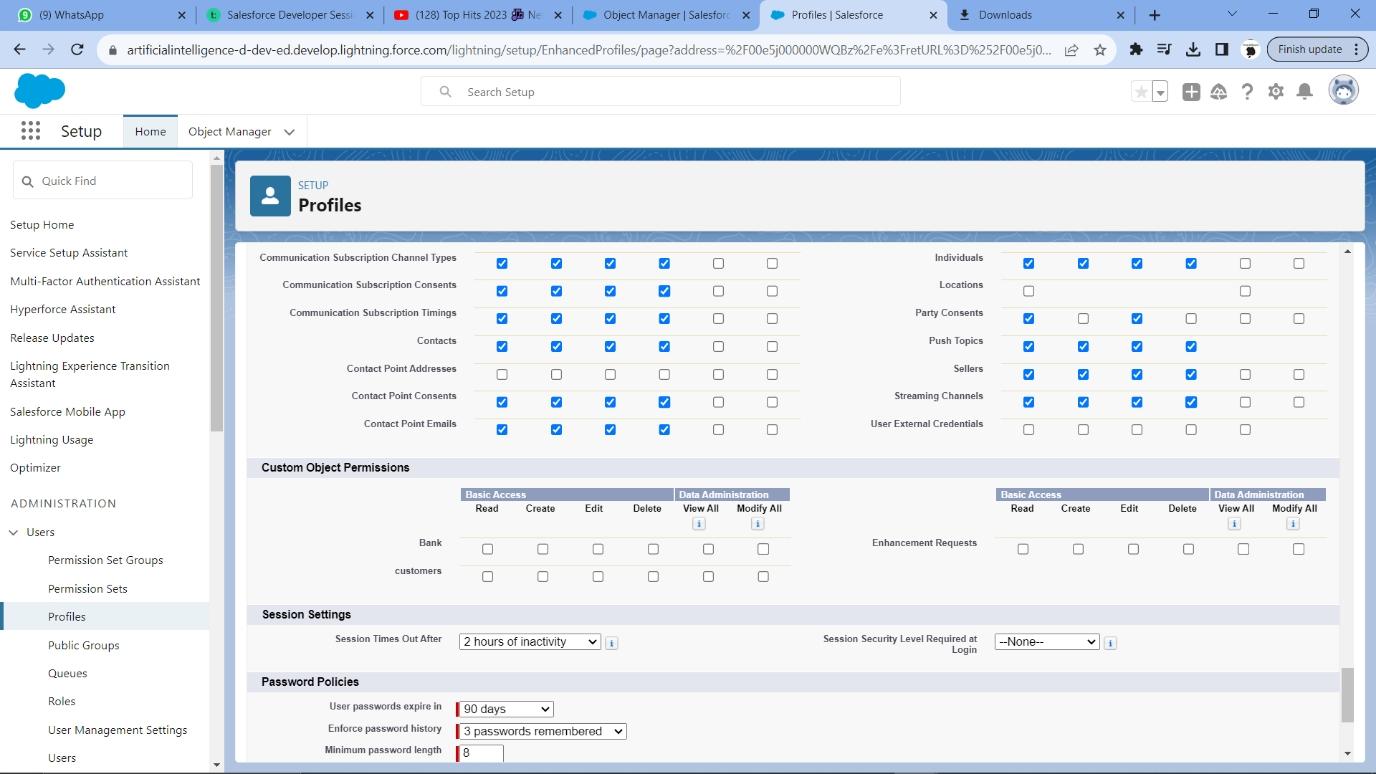


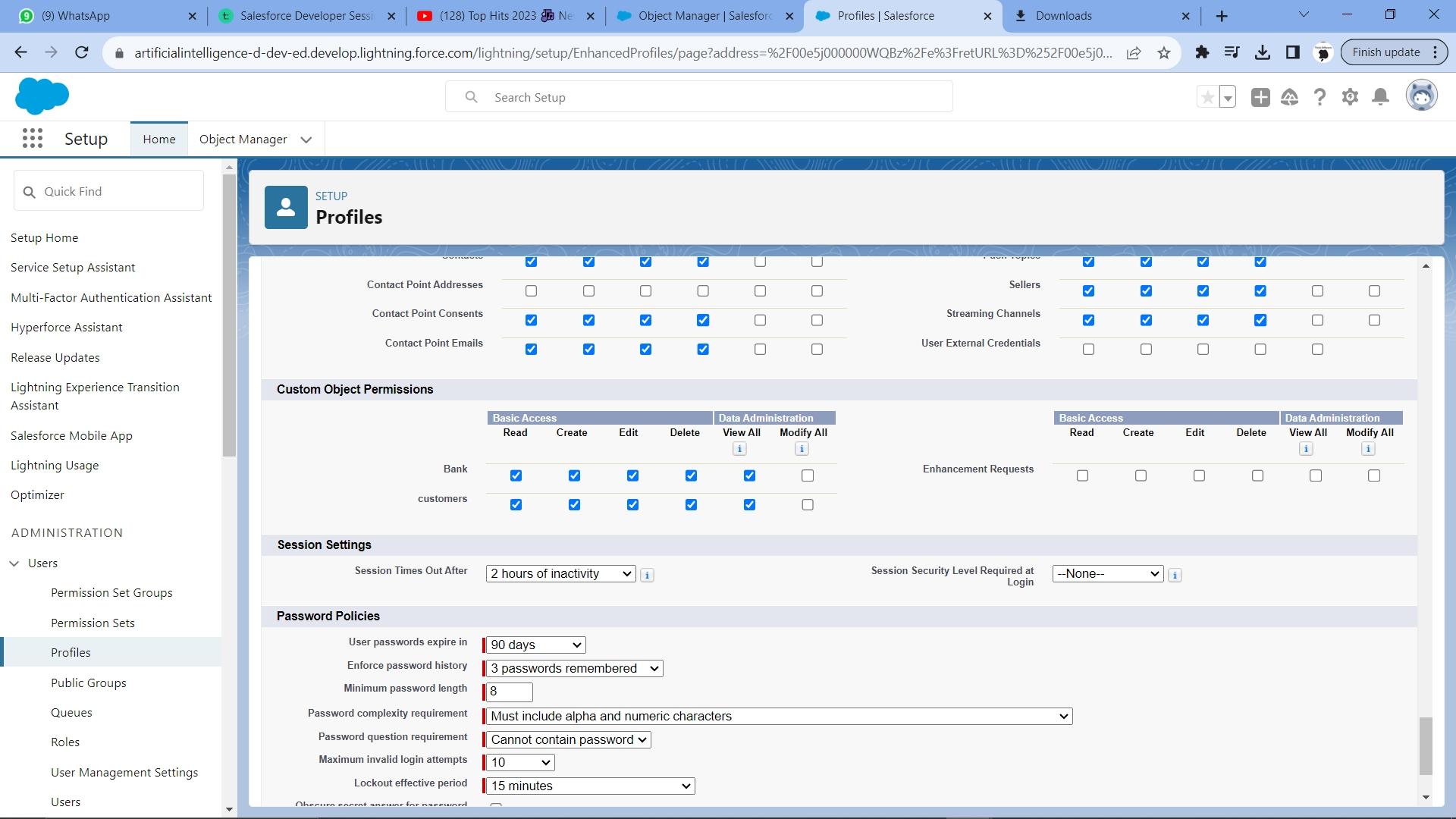


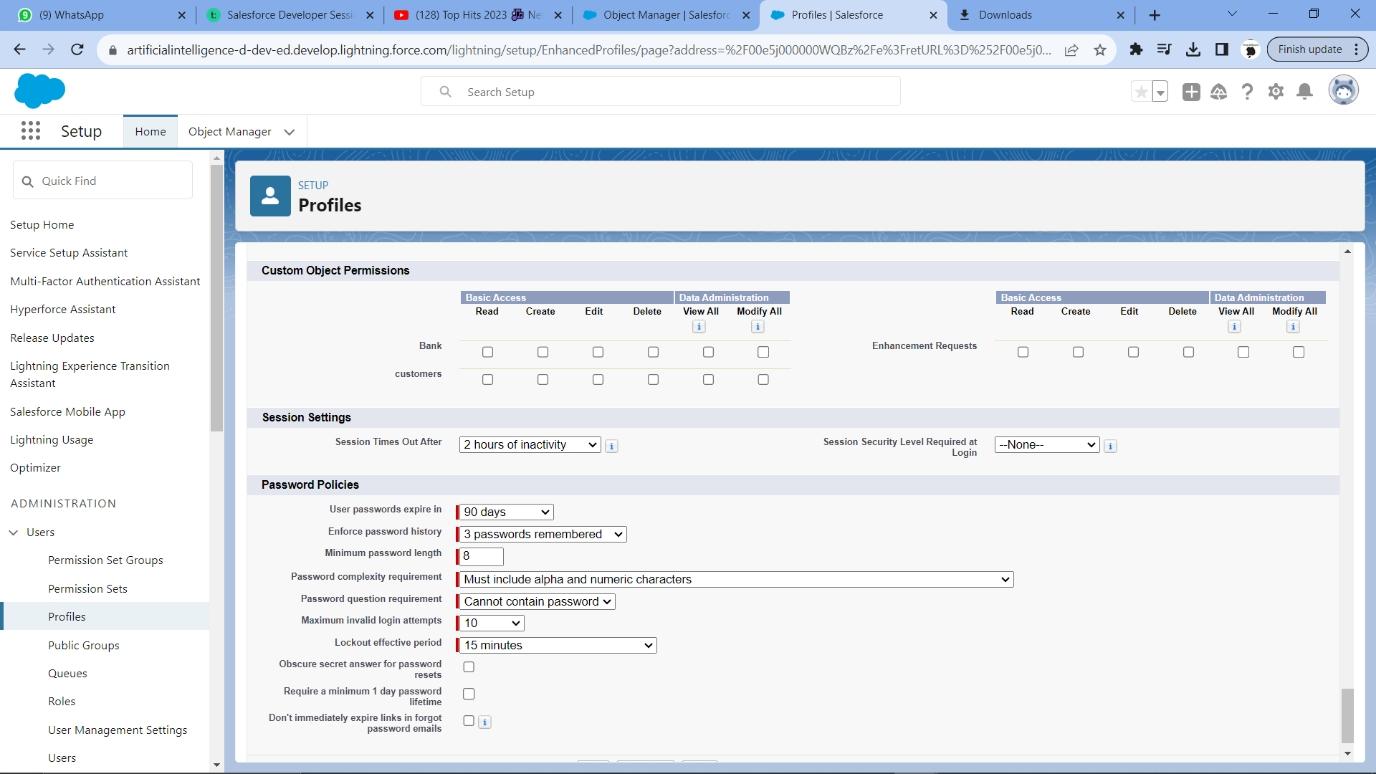
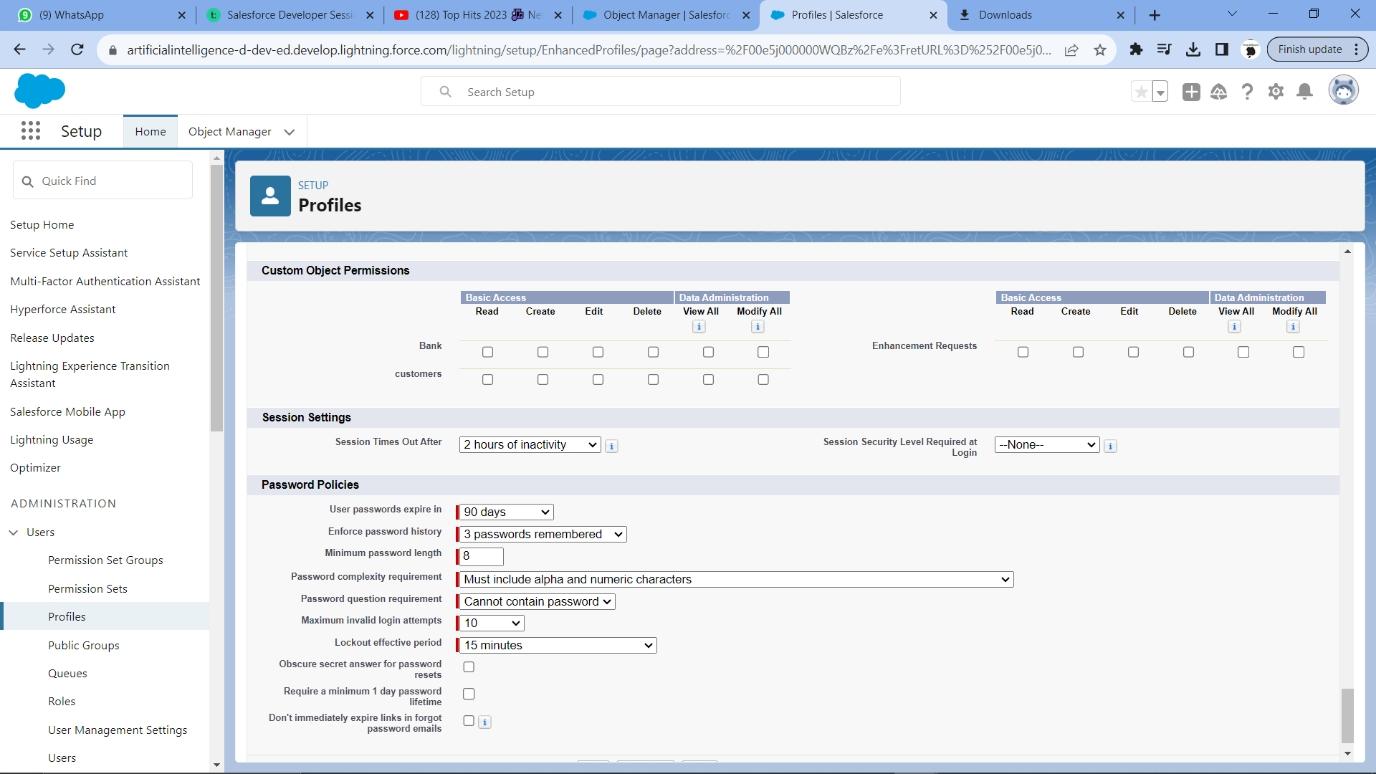


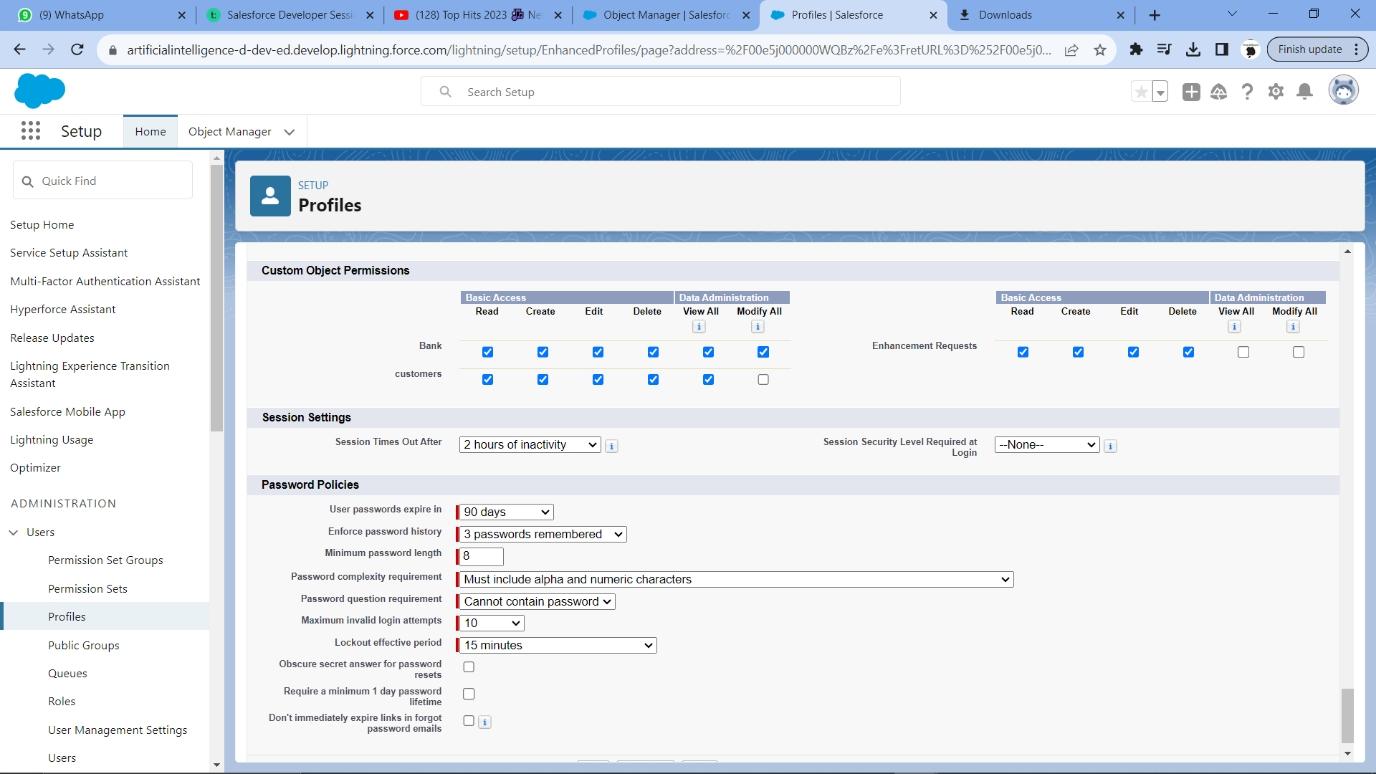


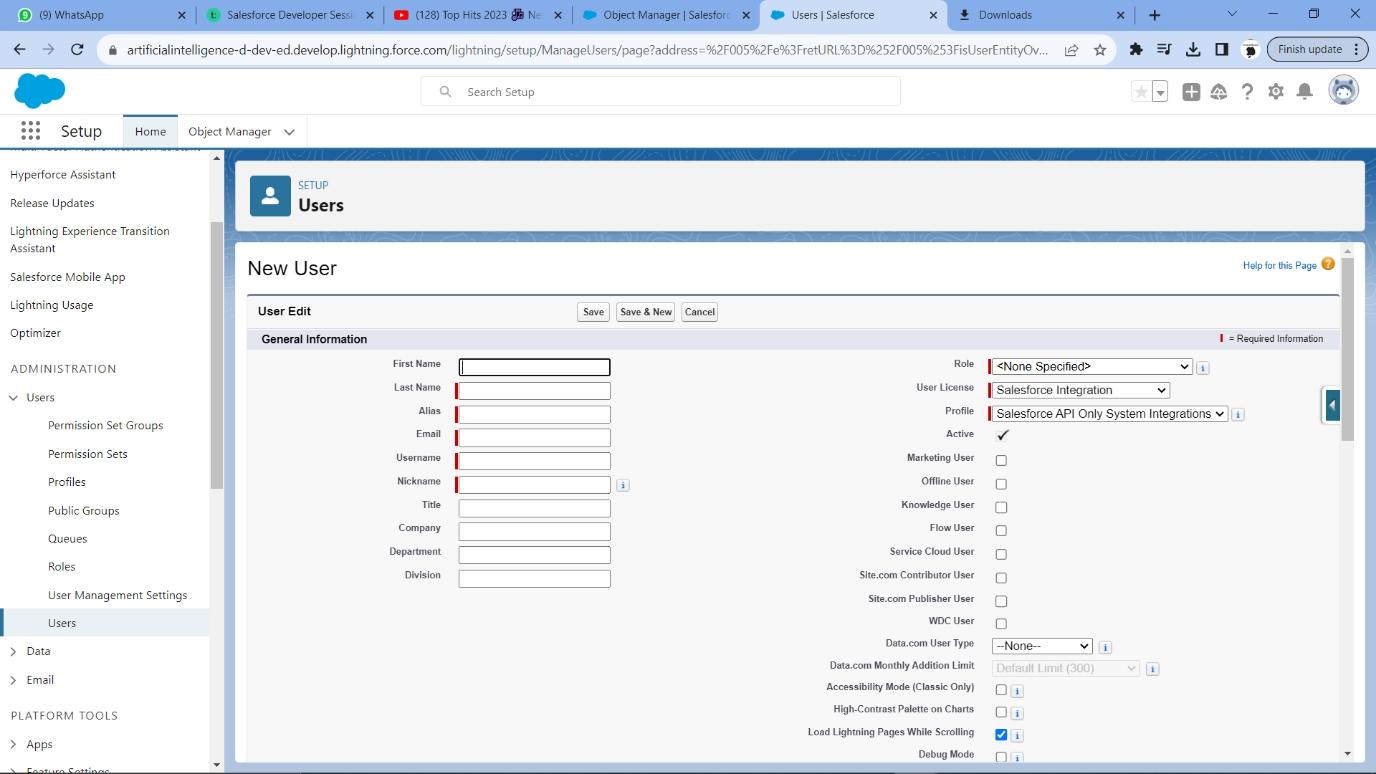


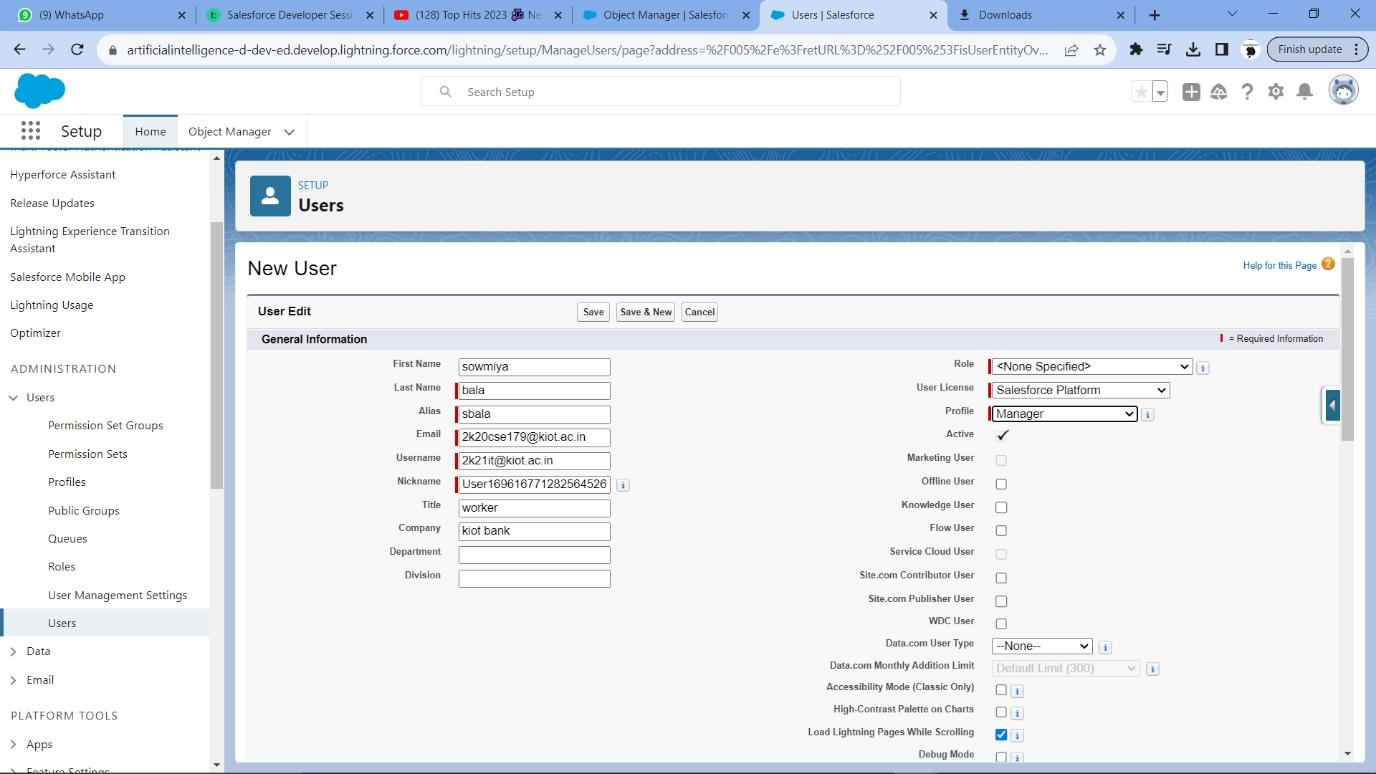




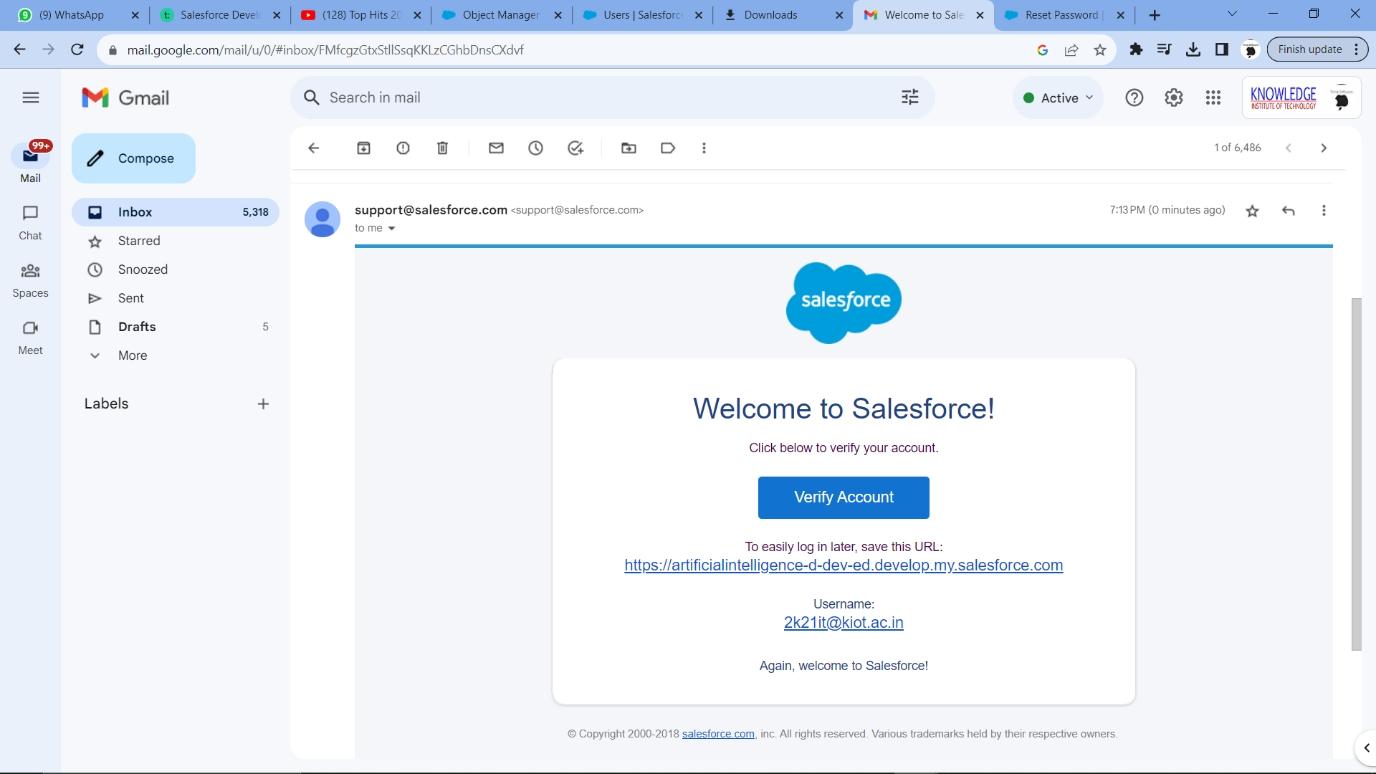


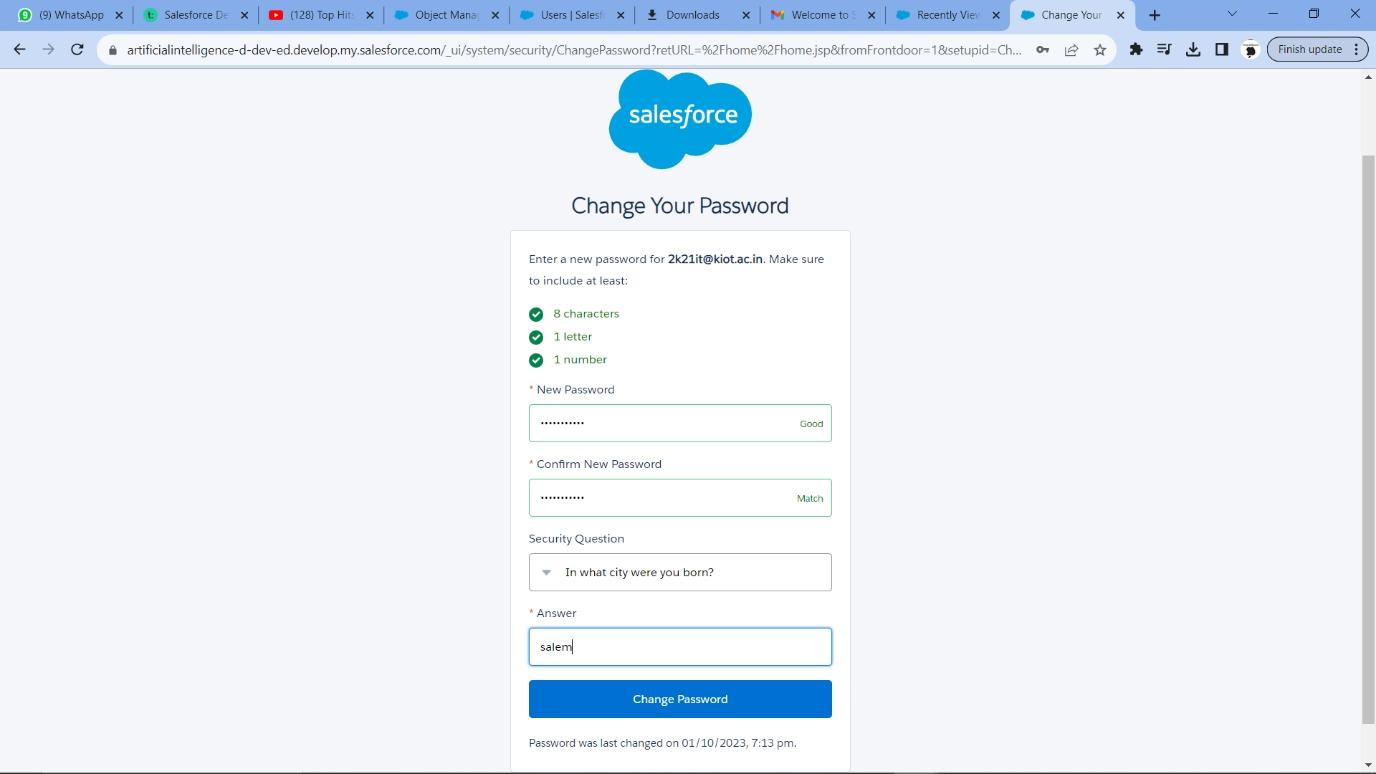


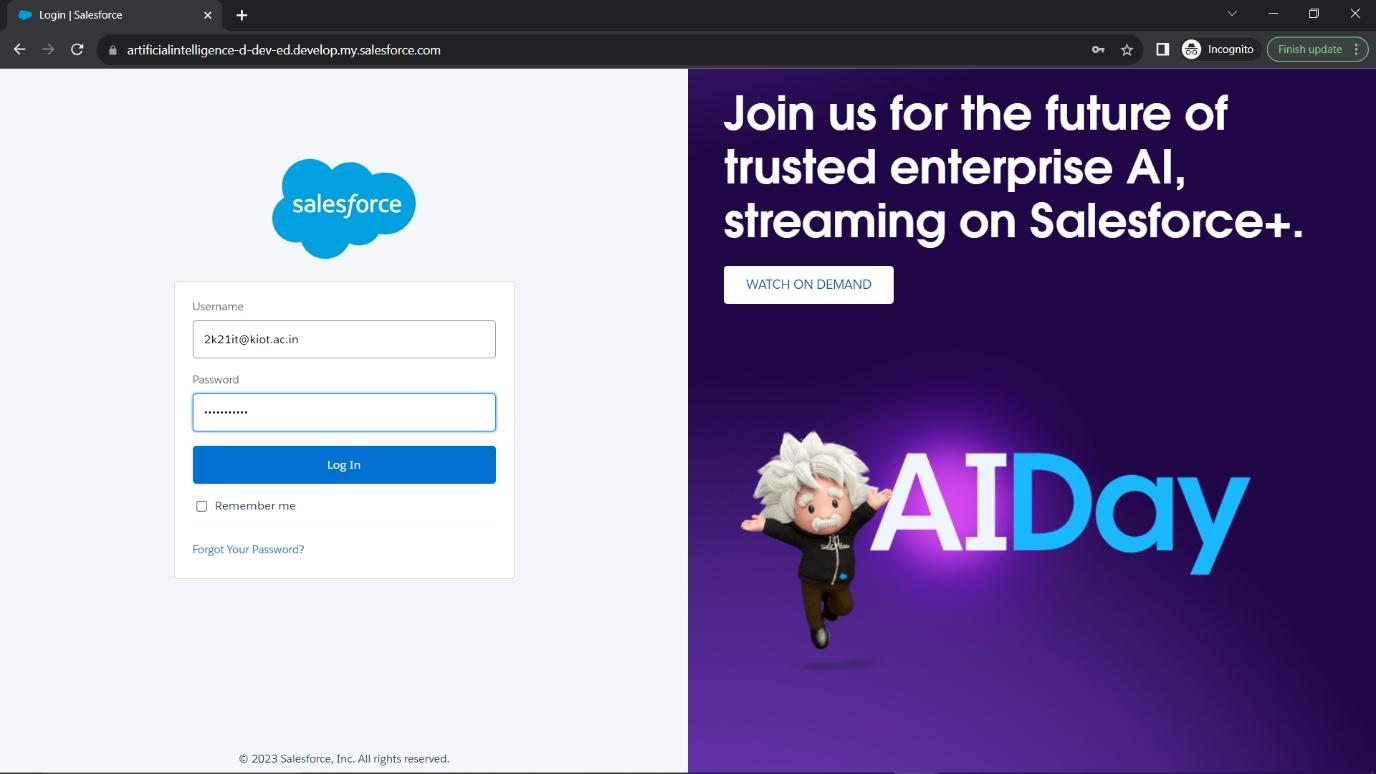


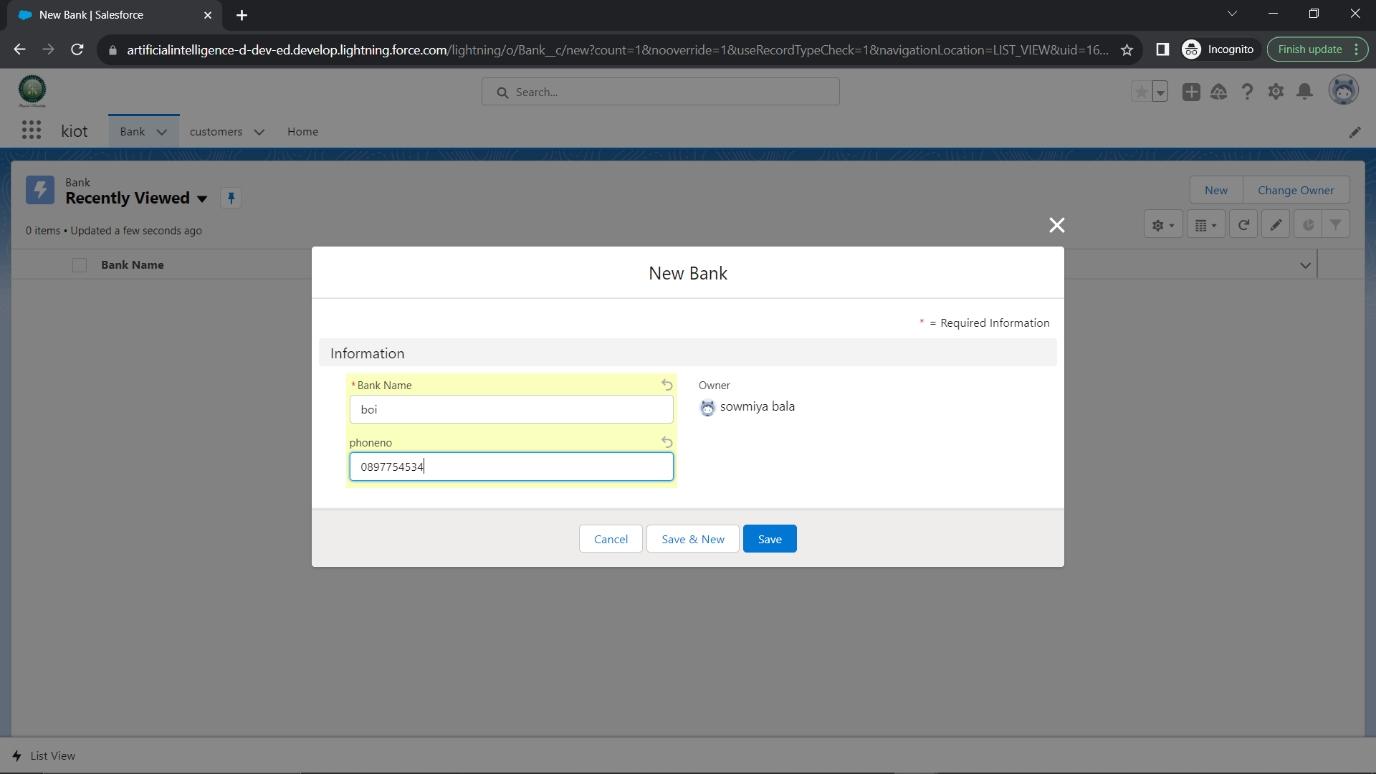
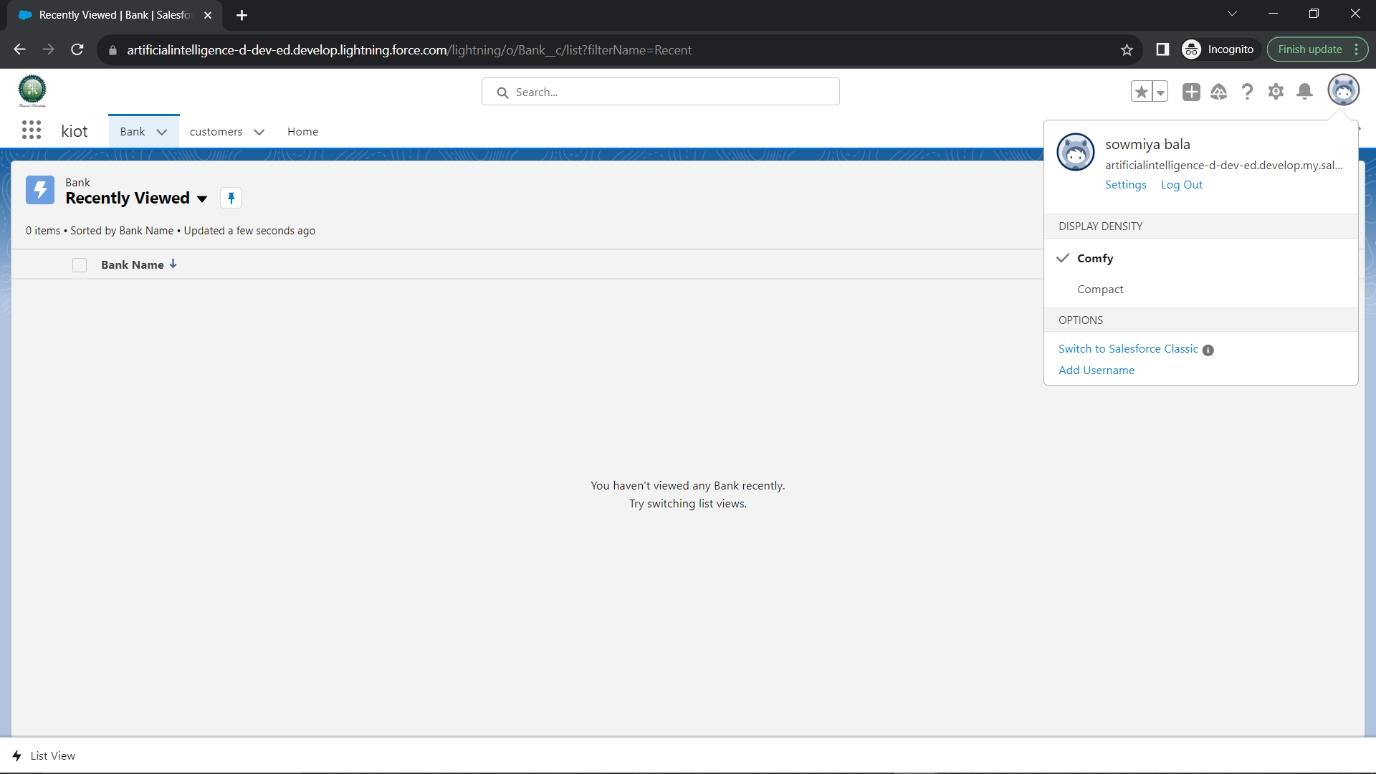


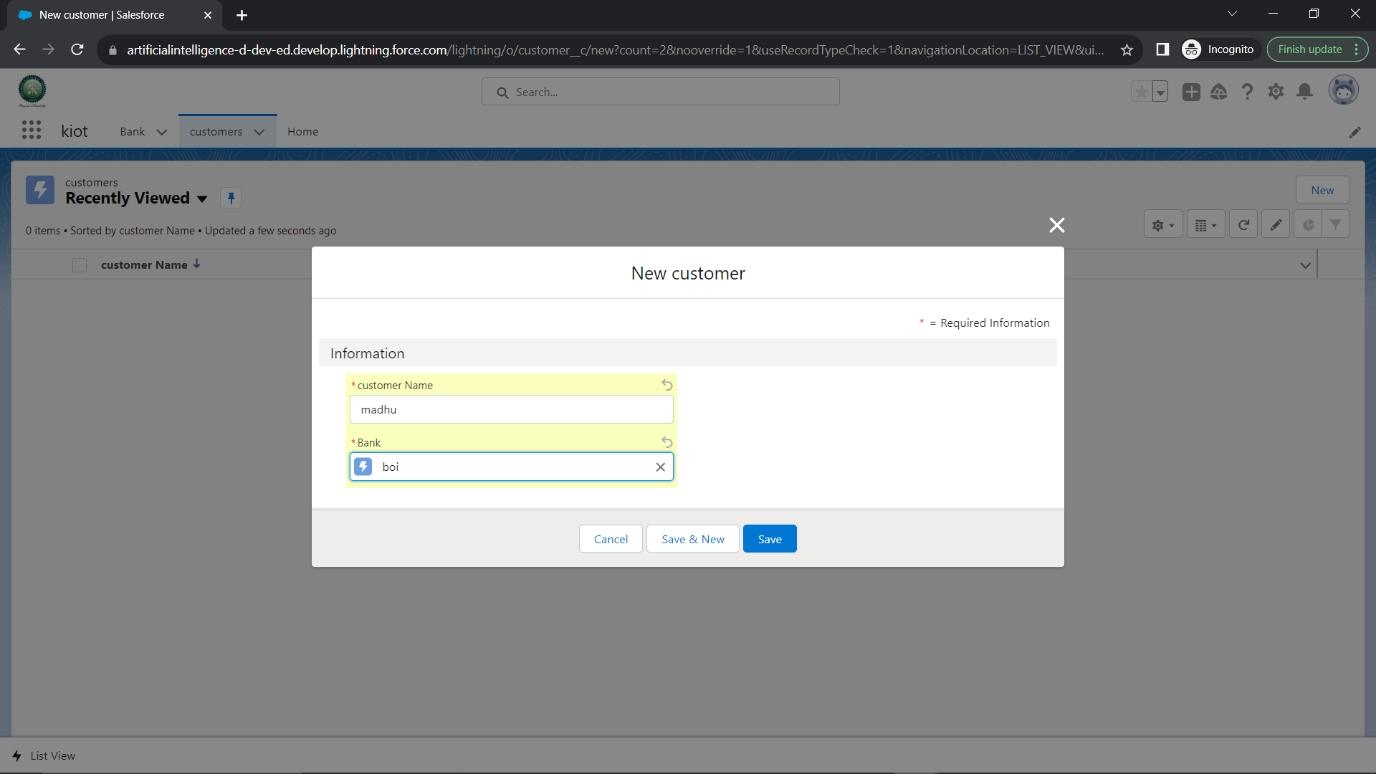


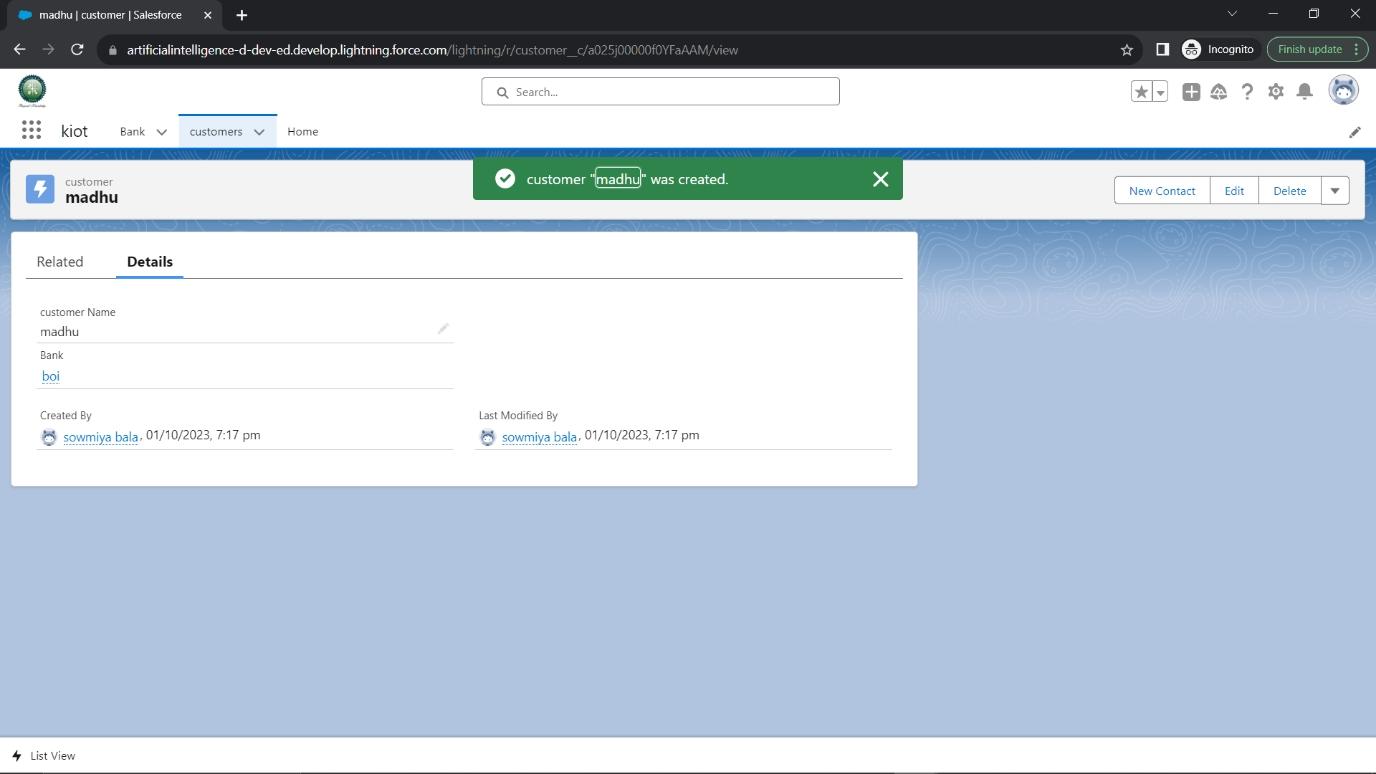
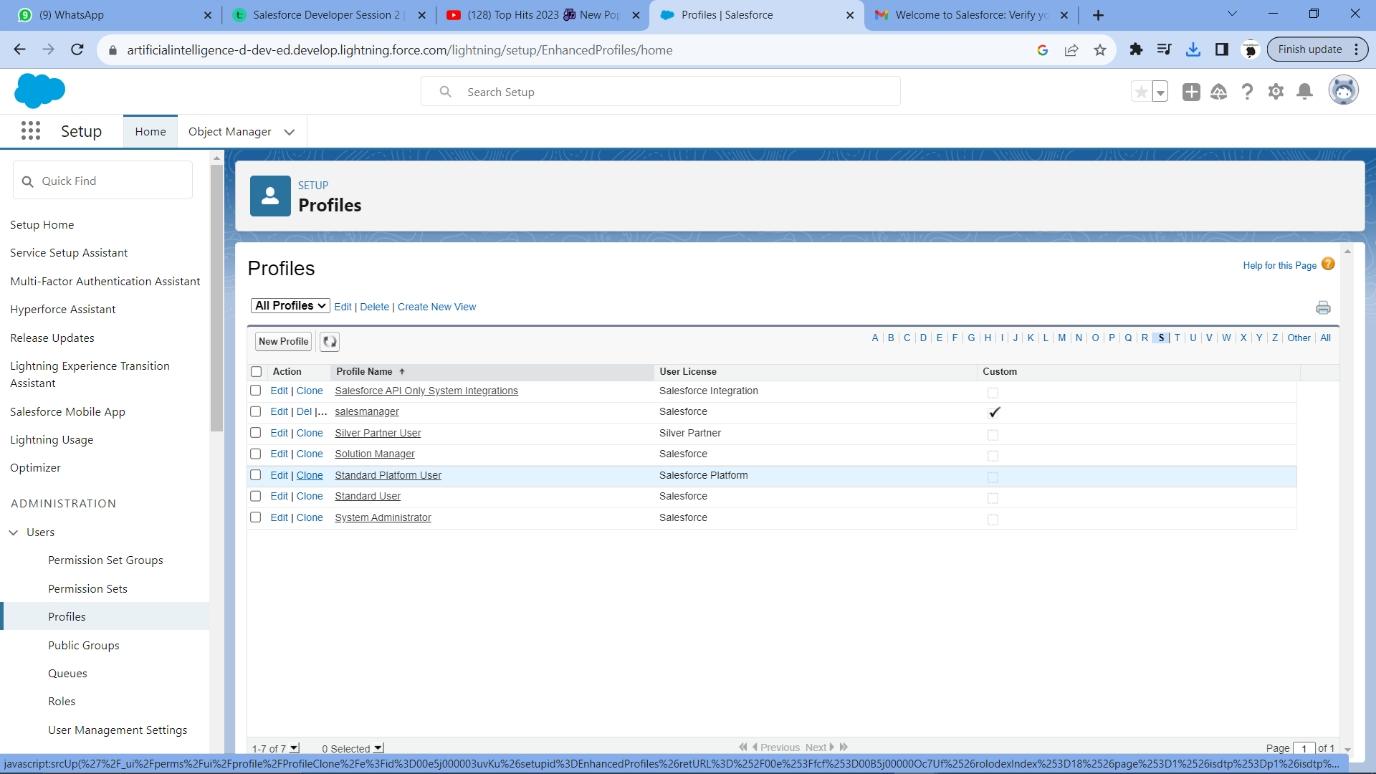


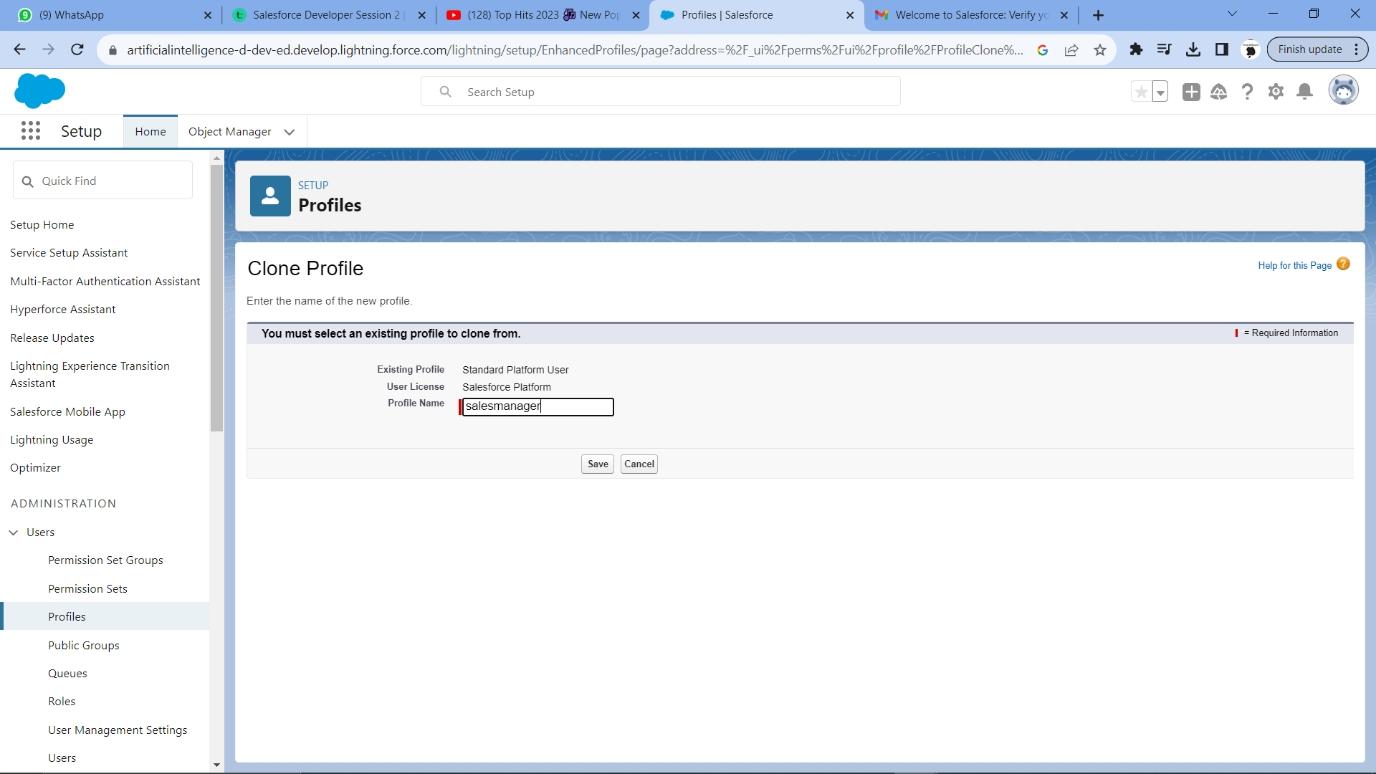


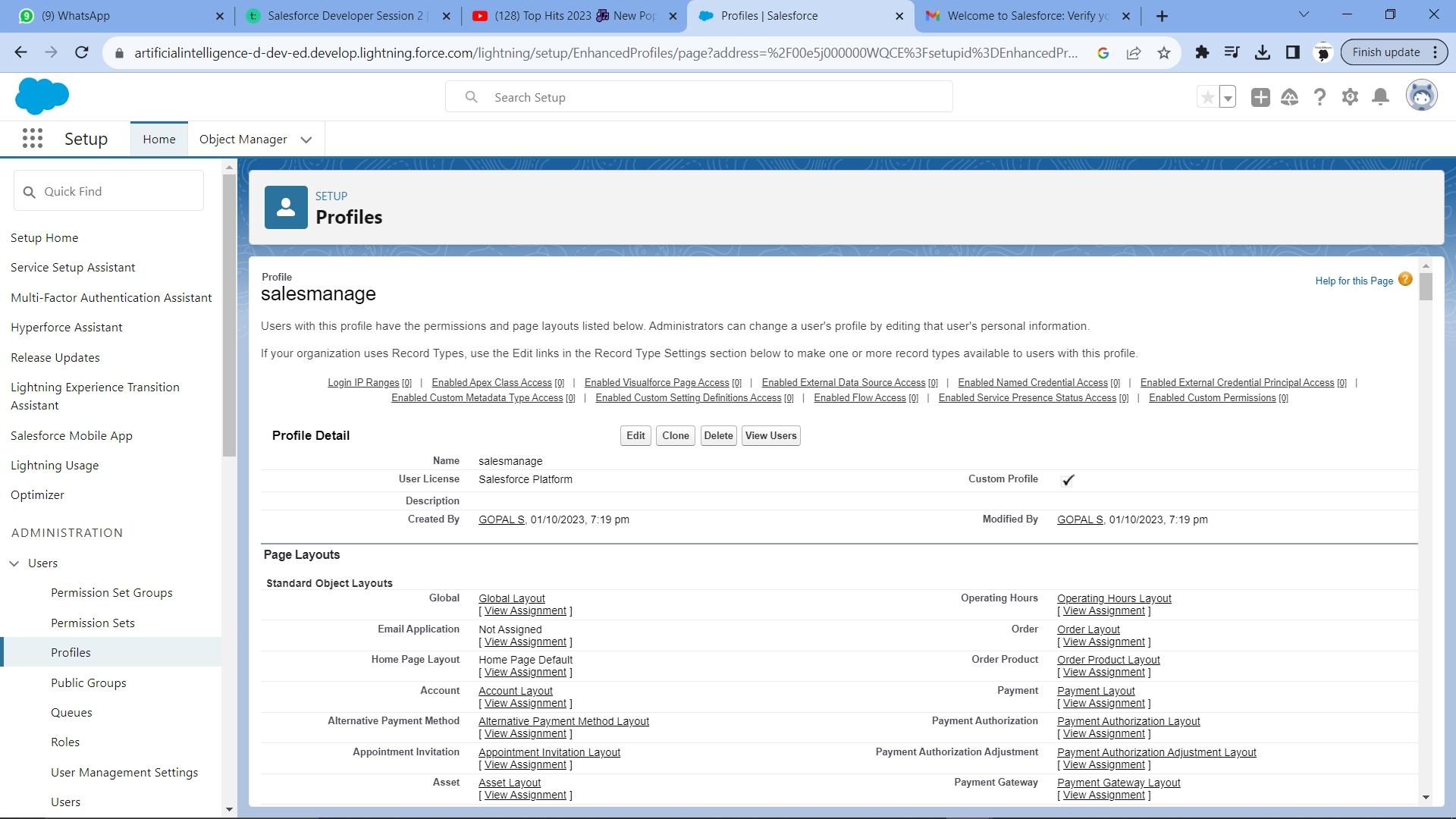


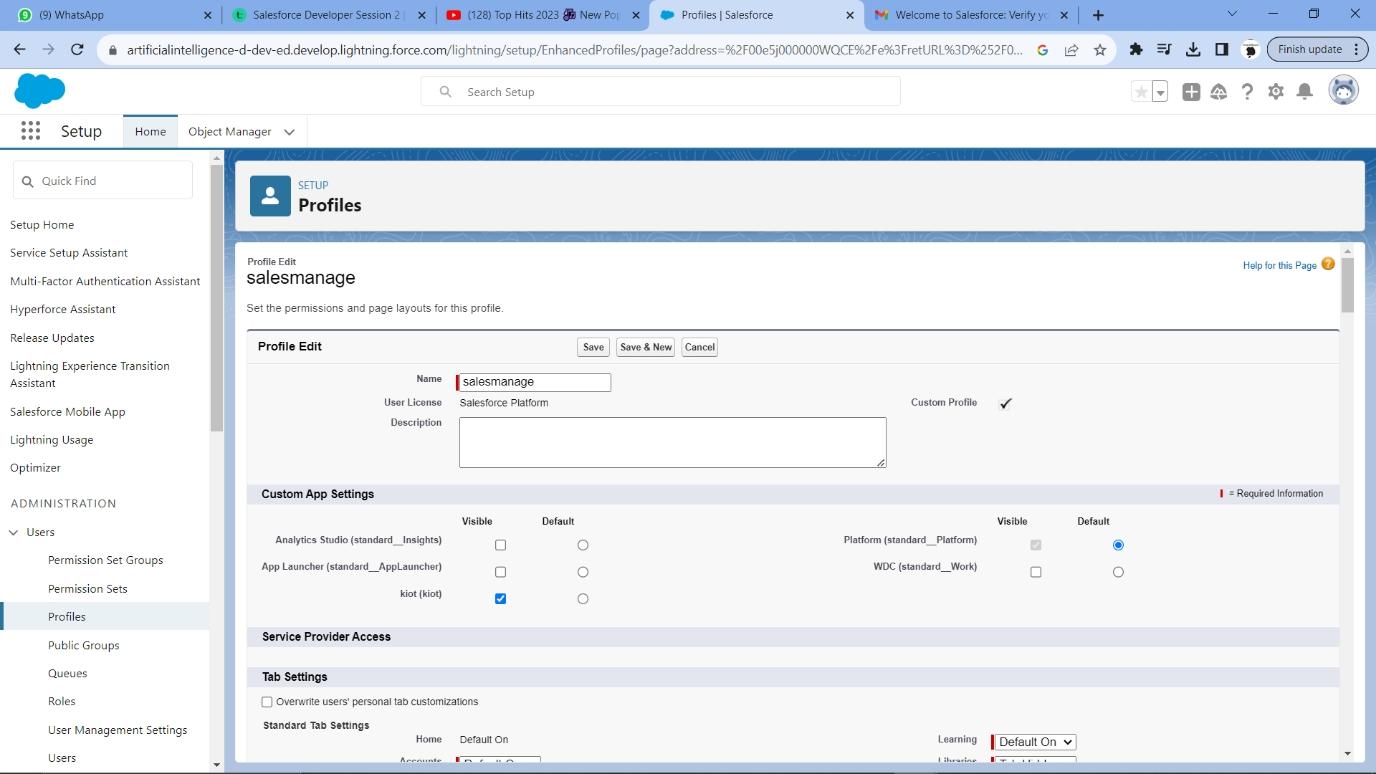


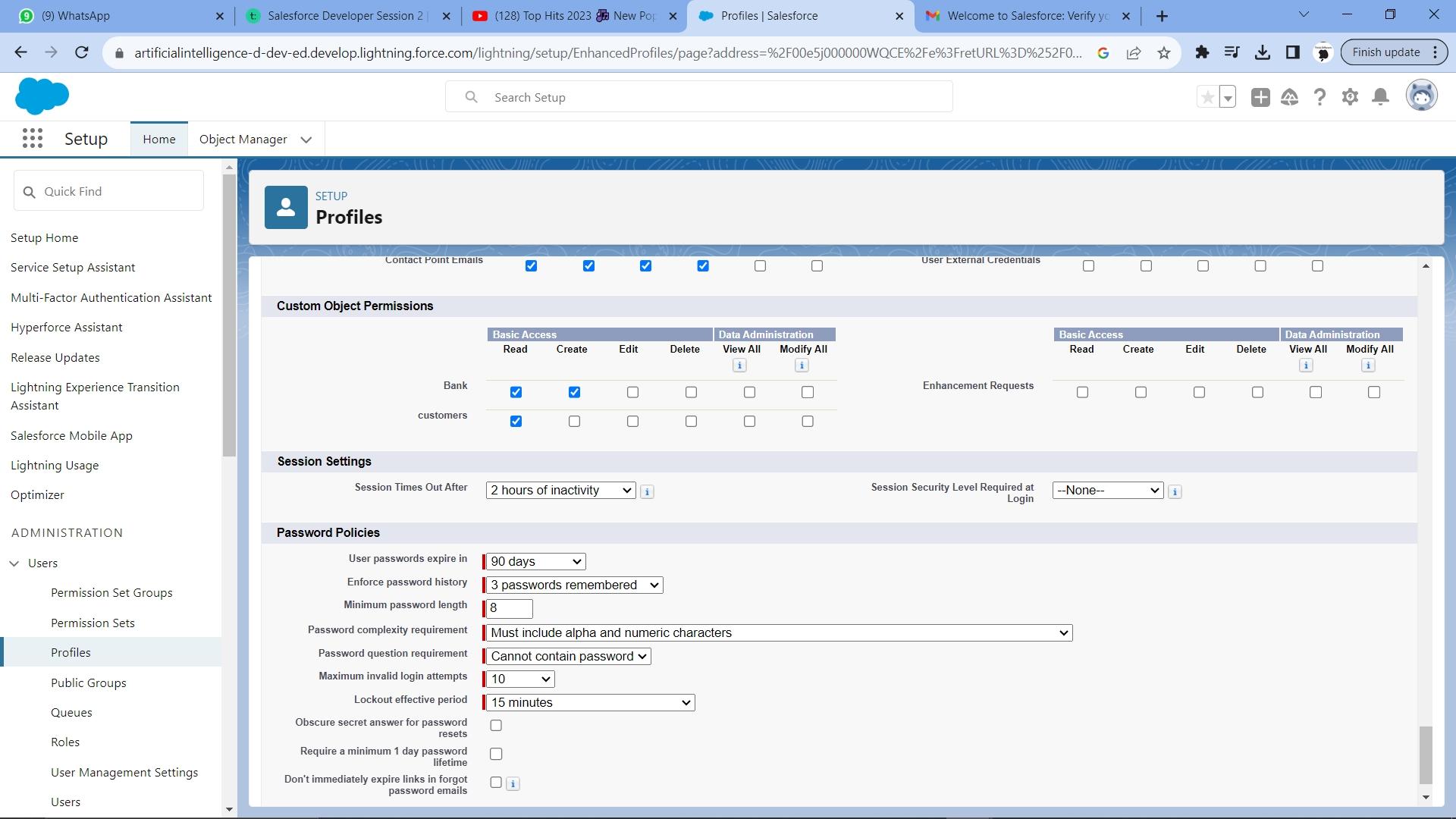


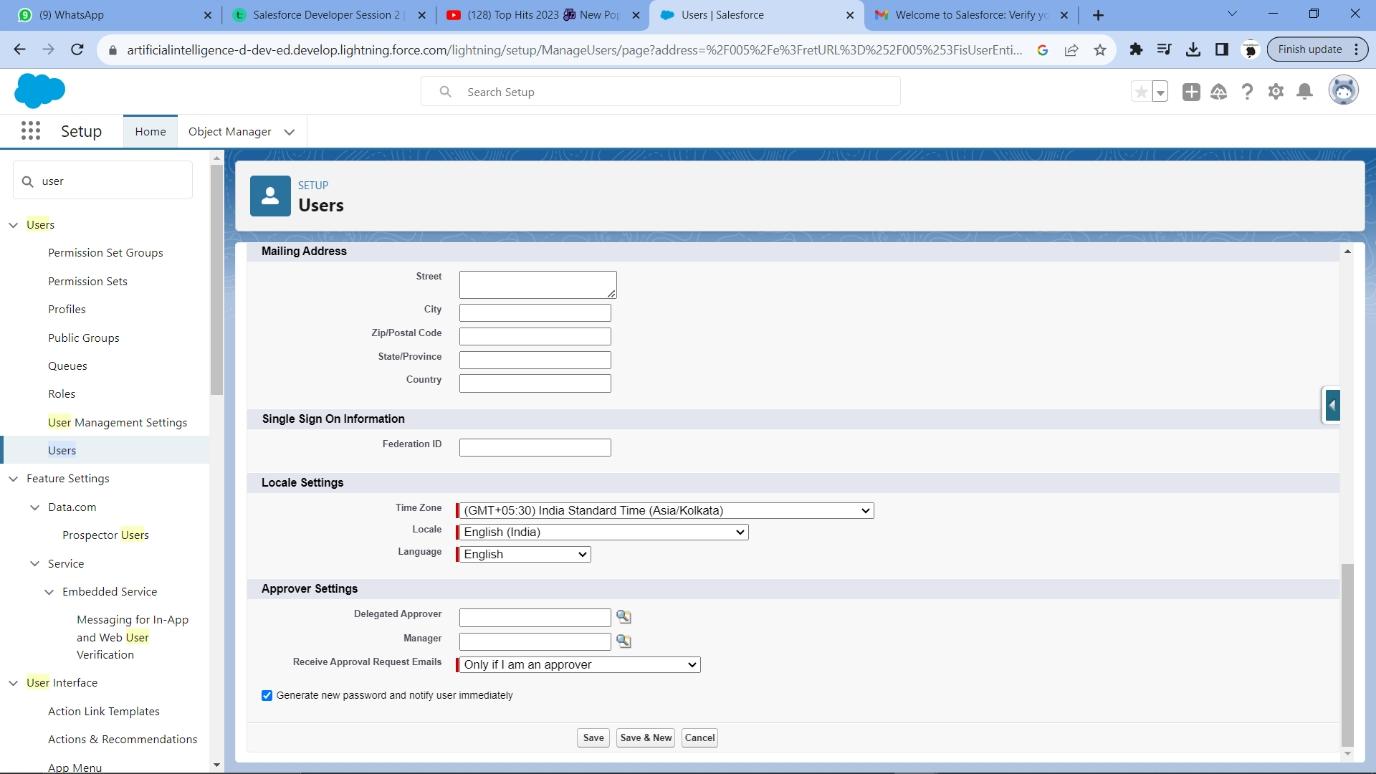
 

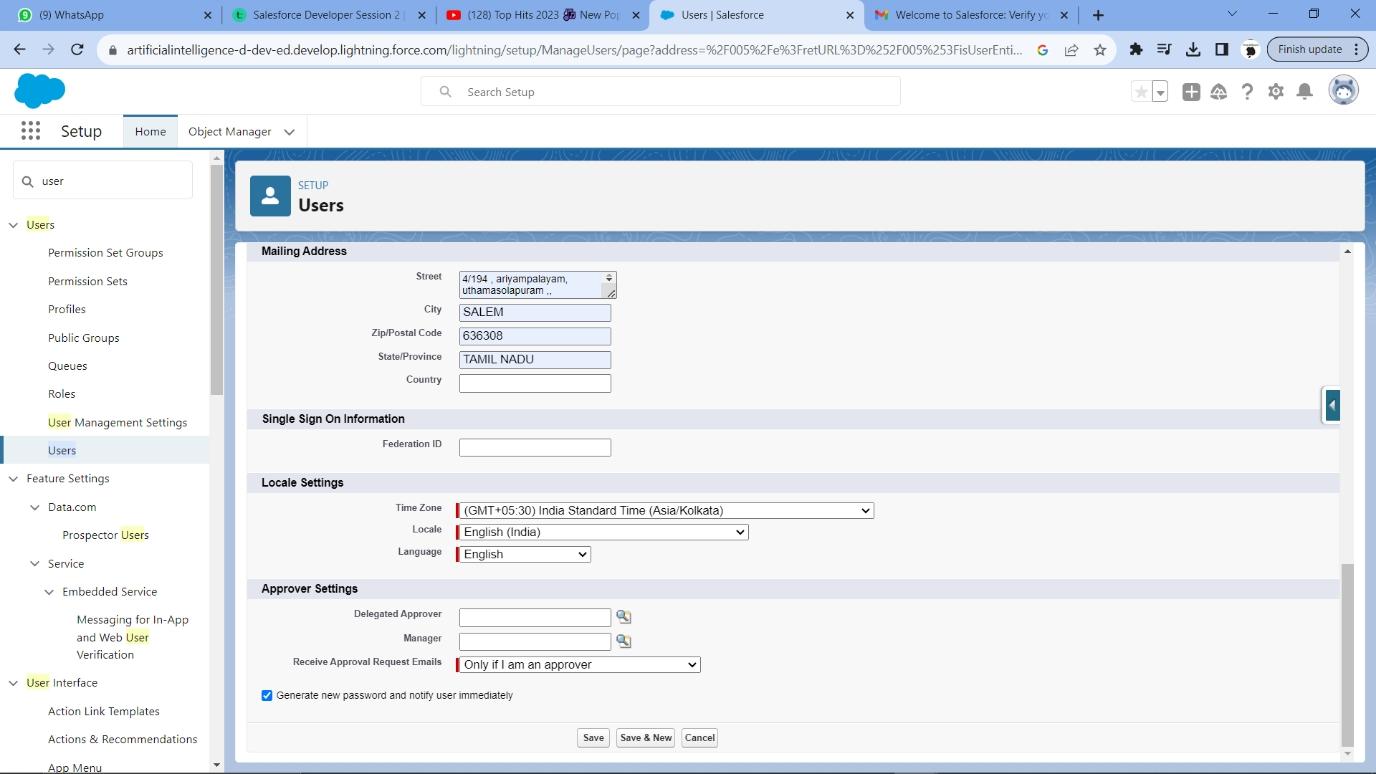


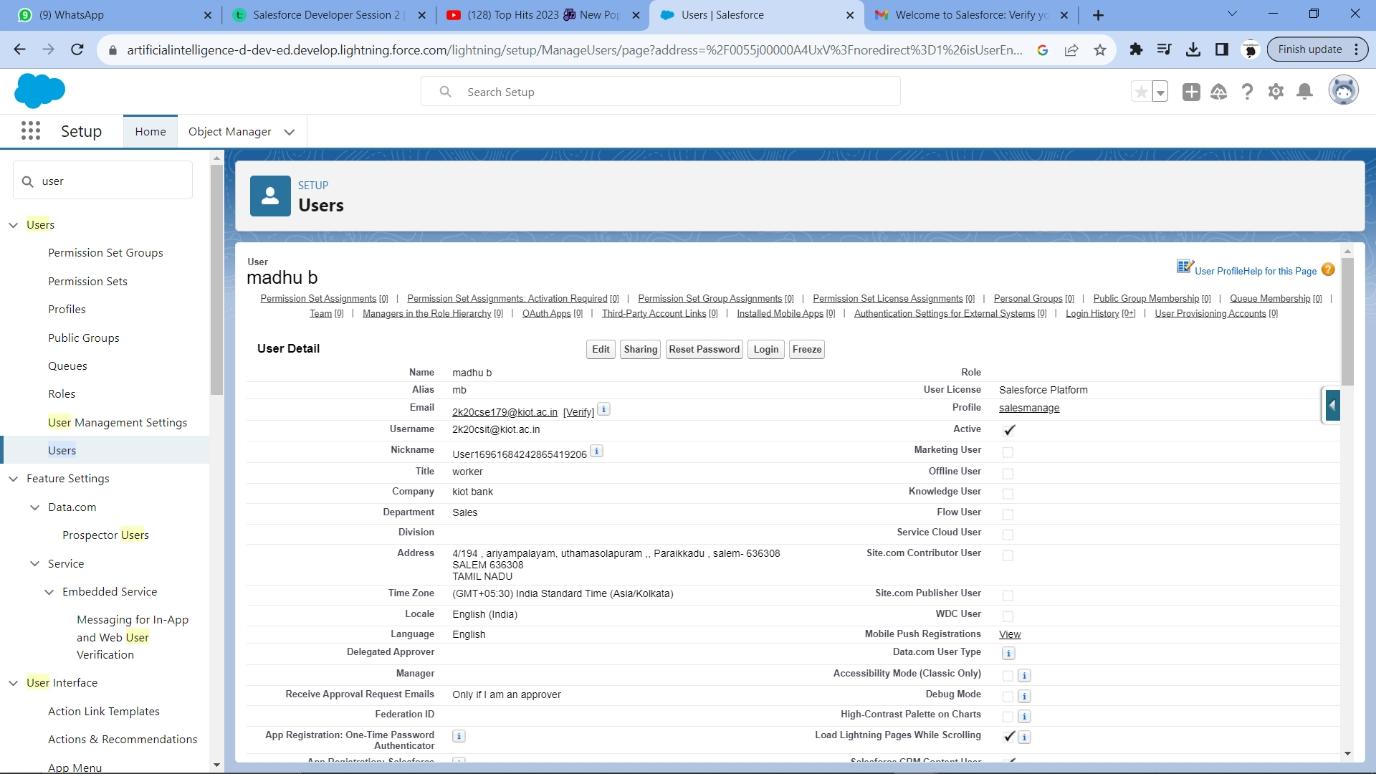


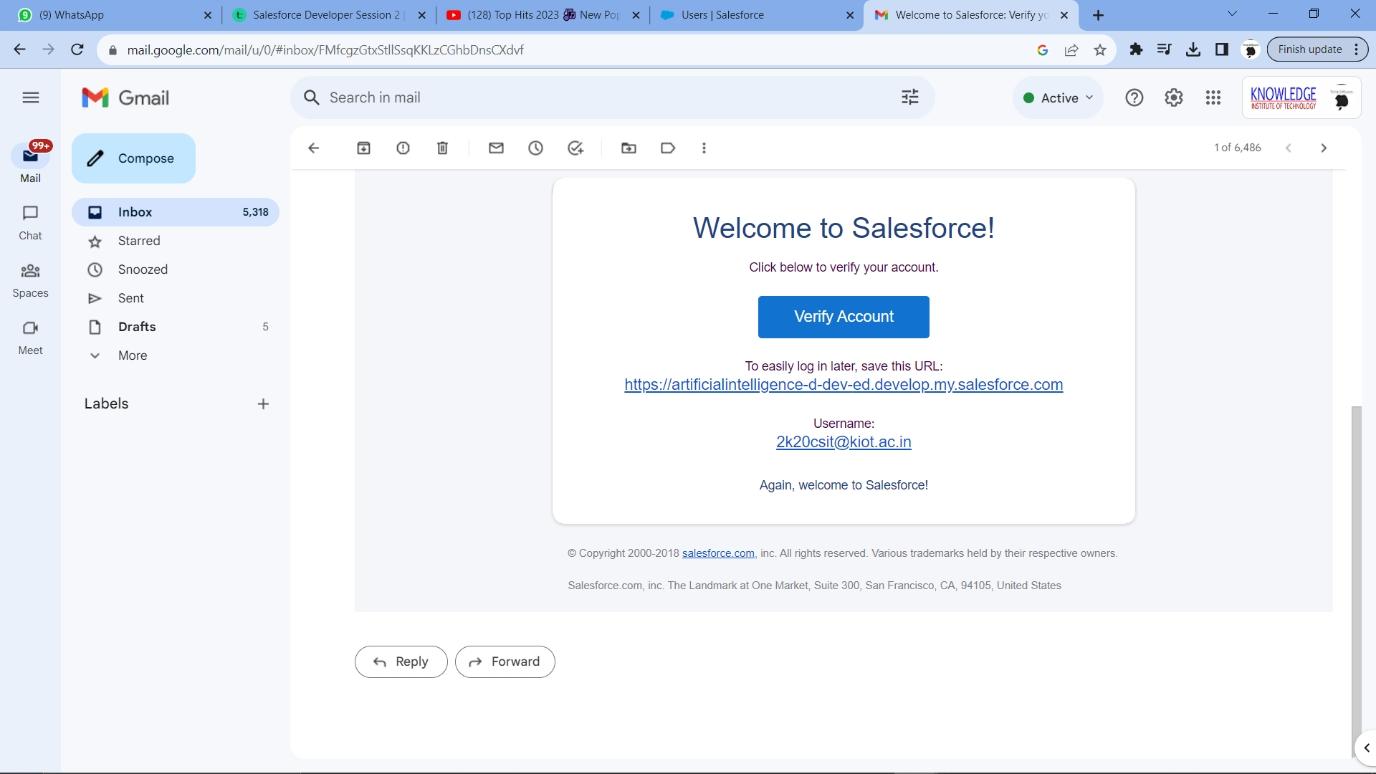


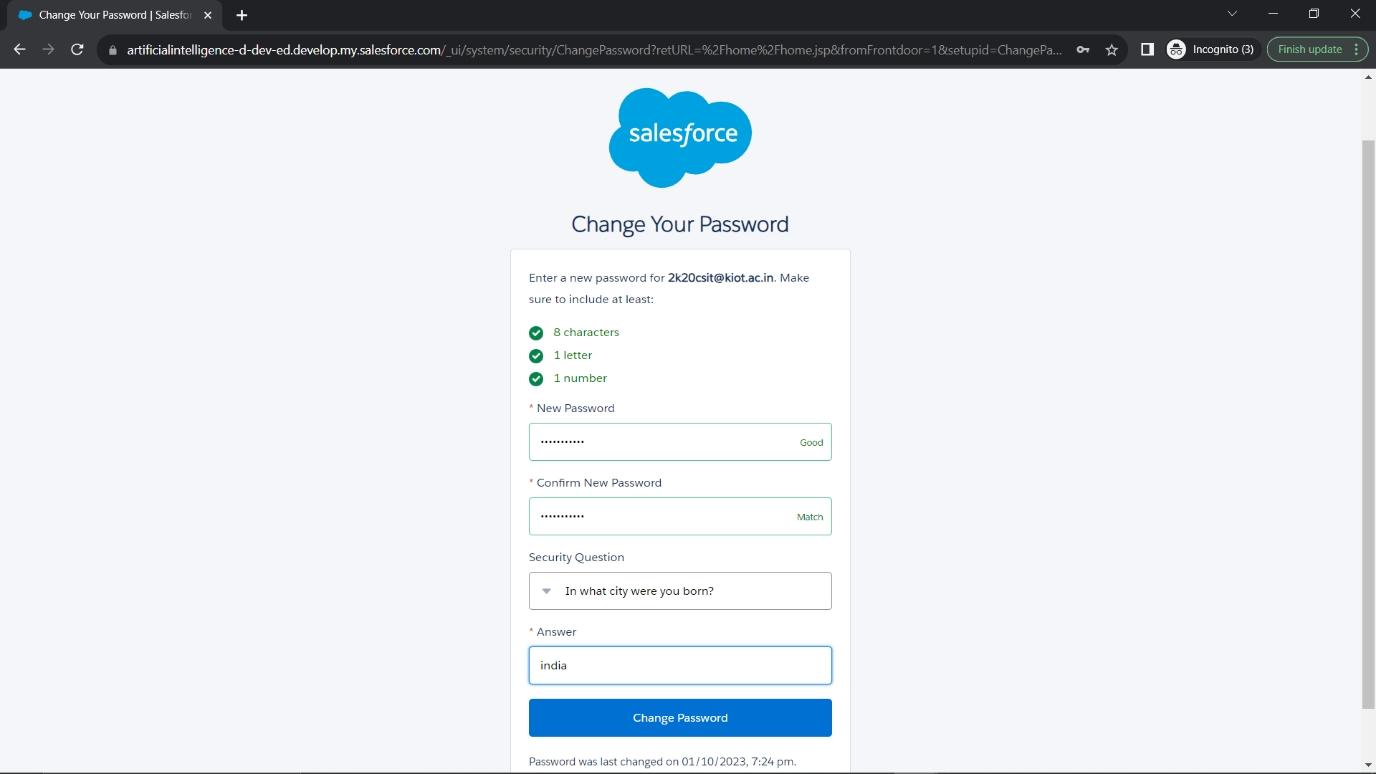


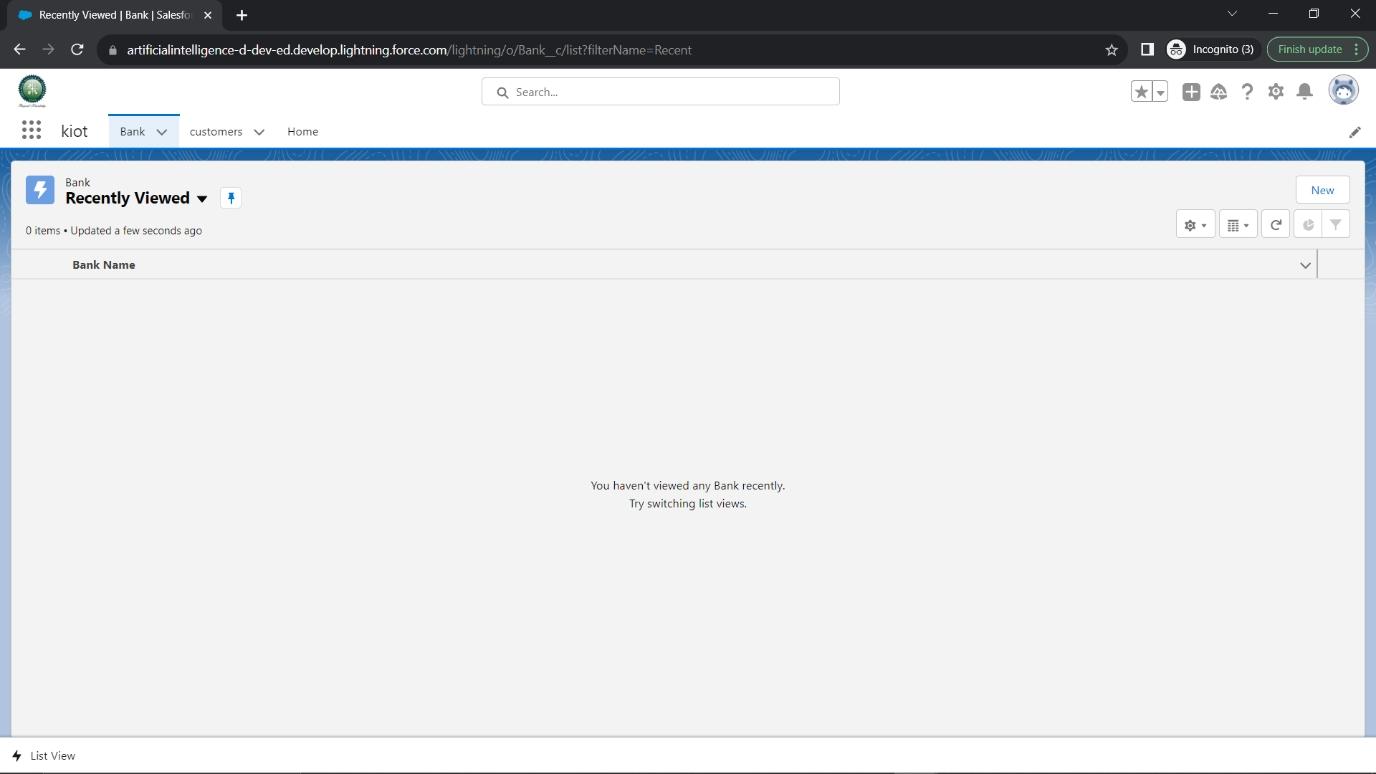


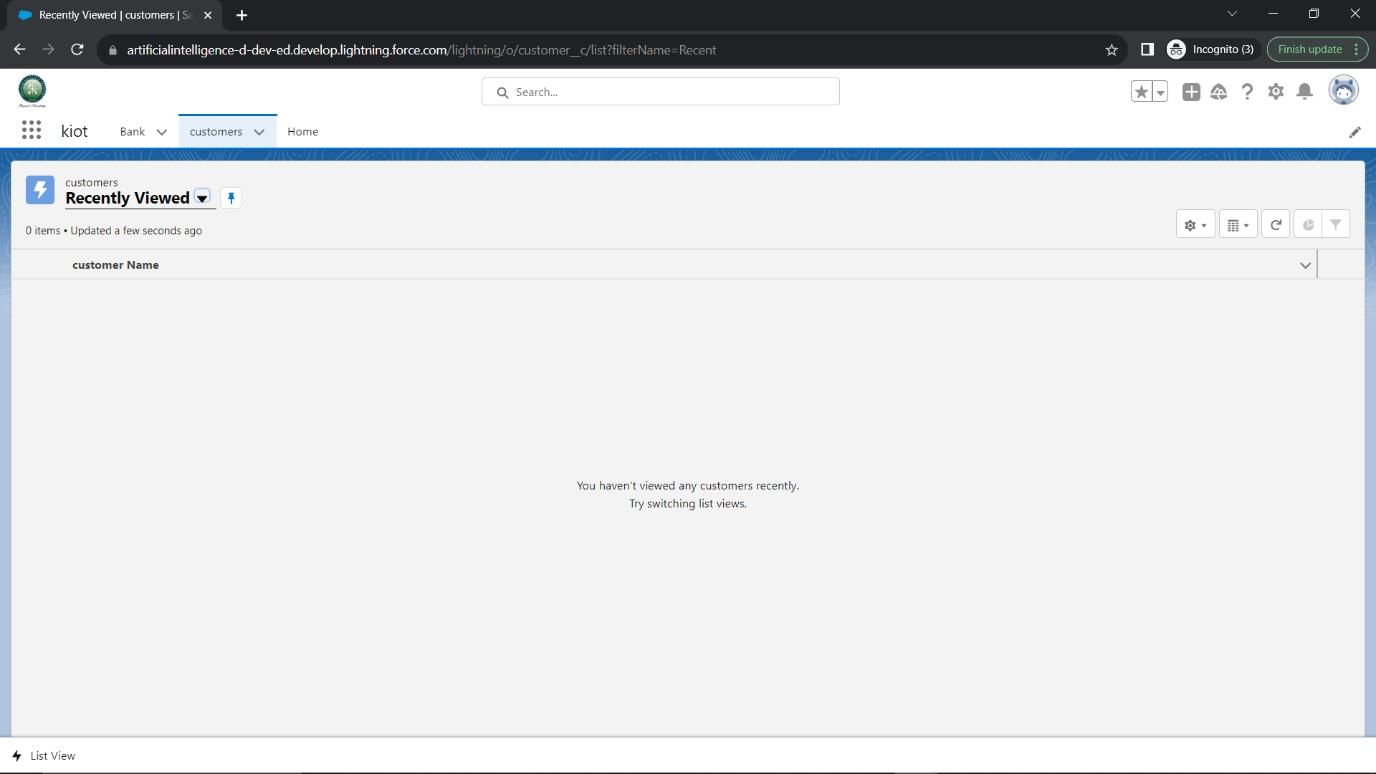












Step 2:

Permission Sets:

• Create two permission sets, one for User A and one for User B.

Object-Level Security:

• In each profile and permission set, set the object-level security for the Account object to “Read” to ensure that both I-Jser A and I-Jser B can view Account records.

Record-Level Security:

• Implement record-level security using Criteria-Based Sharing Rules.

• Create a sharing rule that shares Account records owned by User A with User A and Records owned by IJser B with user B.

• For the sharing rule criteria, specify that records owned by User A are shared with user A, and records owned by User B are shared with User B.

Ownership:

• Ensure that the Account records are owned by the respective users, with User A owning Their records and User B owning their records.

Organization-Wide Defaults:

• Set the organization-wide defaults for the Account object to “Private” to ensure that Records are private by default.

Testing:

• Test the setup by logging in as User A and User B separately to verify that they cannot Access each other’s records.

