**Candidate Name:** Gaurav Lal

**Date of Submit:** 24-Aug-2020

**Case Study:** Quovantis Case Study (ACME Bank)

**Technologies Used:**

**Backend:** Asp.Net Core 3.1, EF.Core, Sqlite, AutoMapper, Postman, JWT Token

**Front-End:** Angular 10, Bootstrap, Alertify

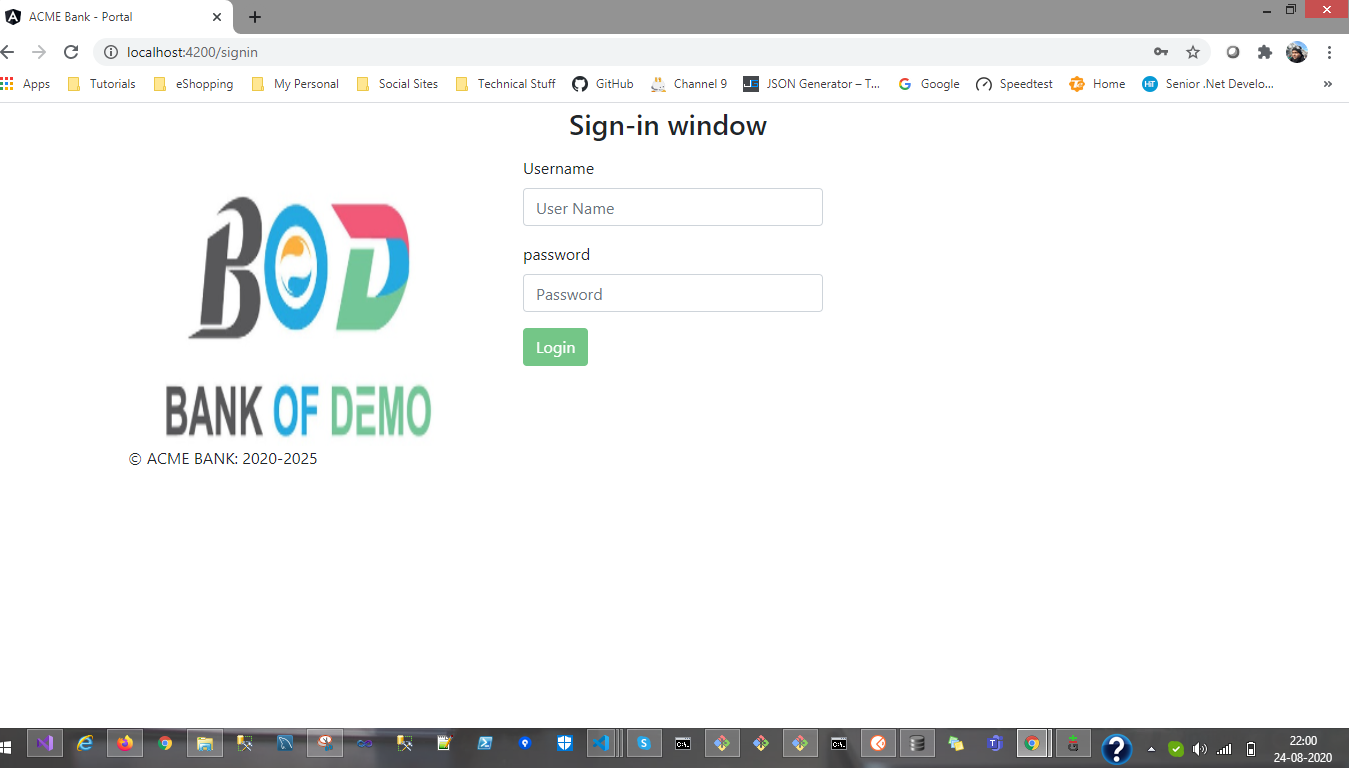
**Backend Service (RESTFUL WEB API)**

**GitHub Repository URL**:

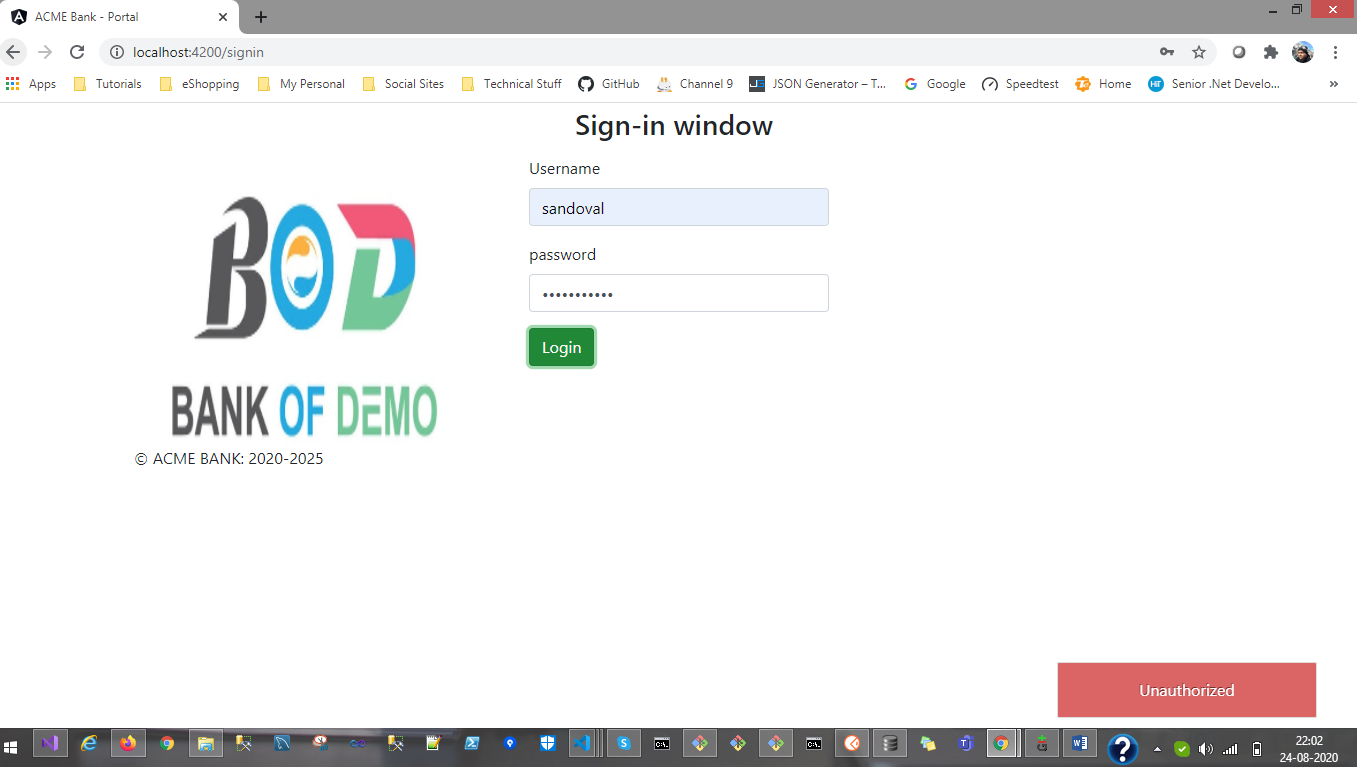
<https://github.com/gaurava16fc/acme-case-study-by-quovantis>

**Front-End Screens**

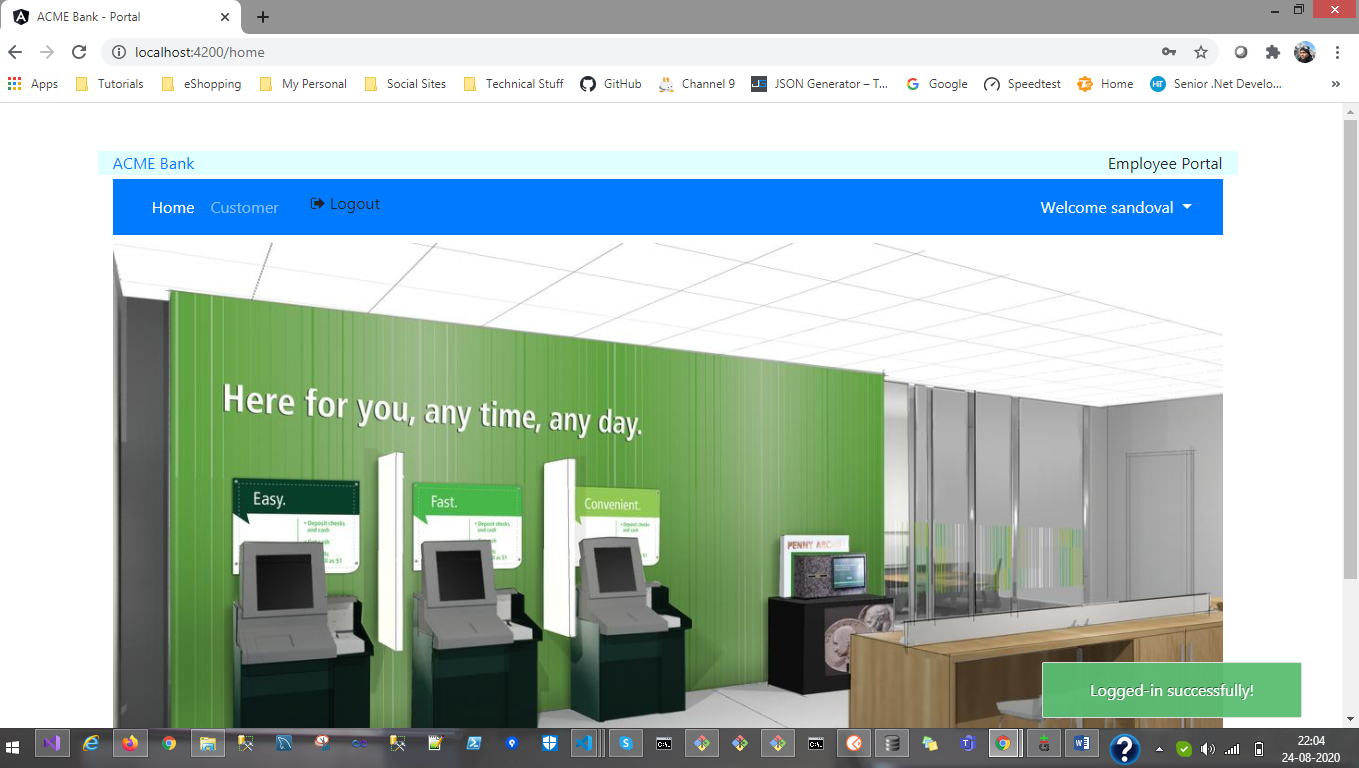
**Login Screen:**



**Invalid Credentials:**

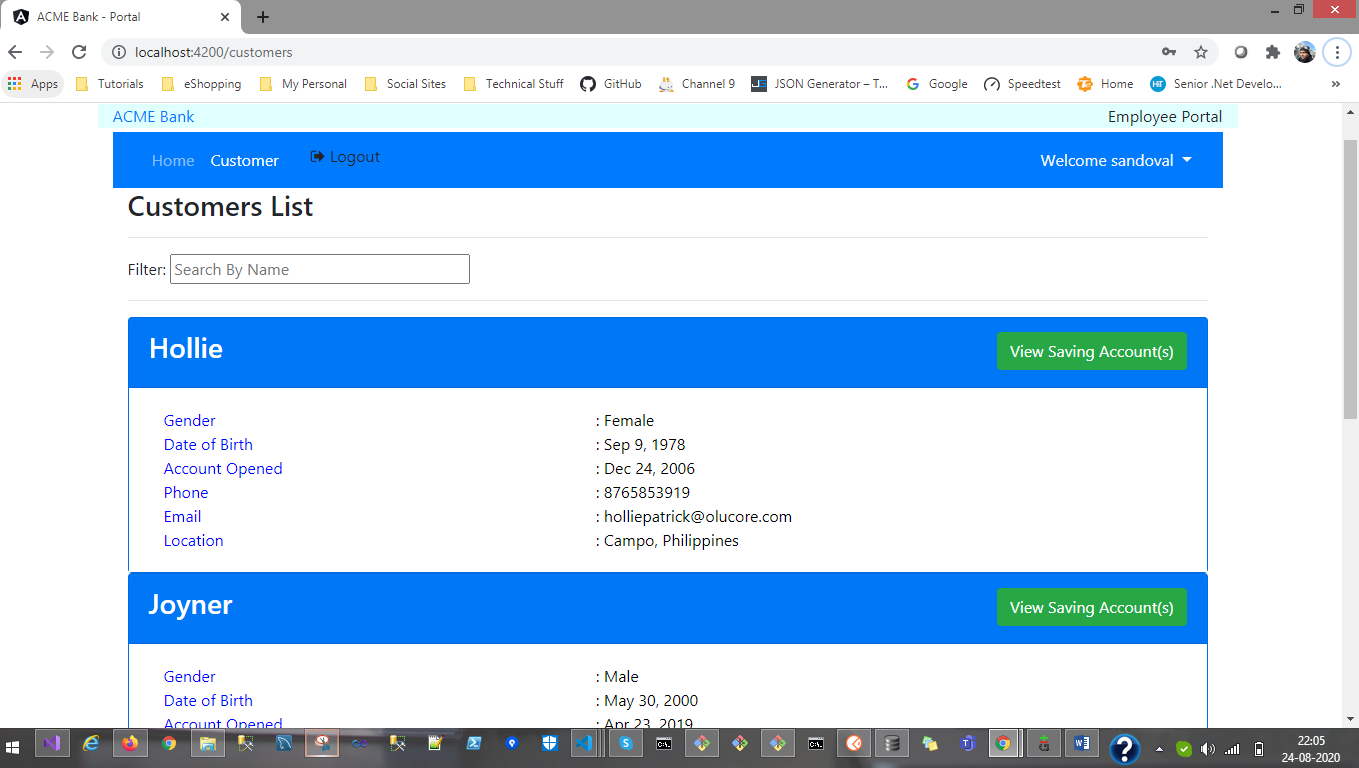


**Home Page (Post Login):**

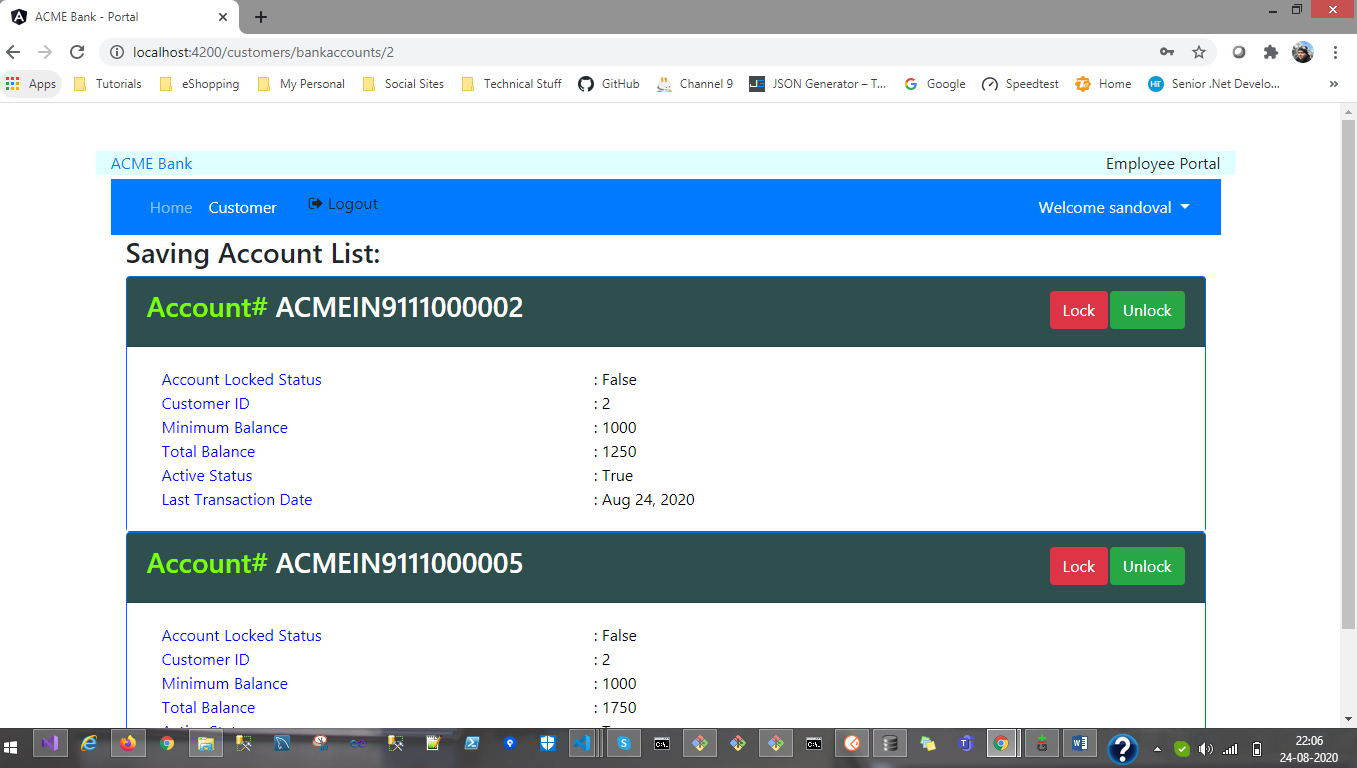


**Employee Login Screens**

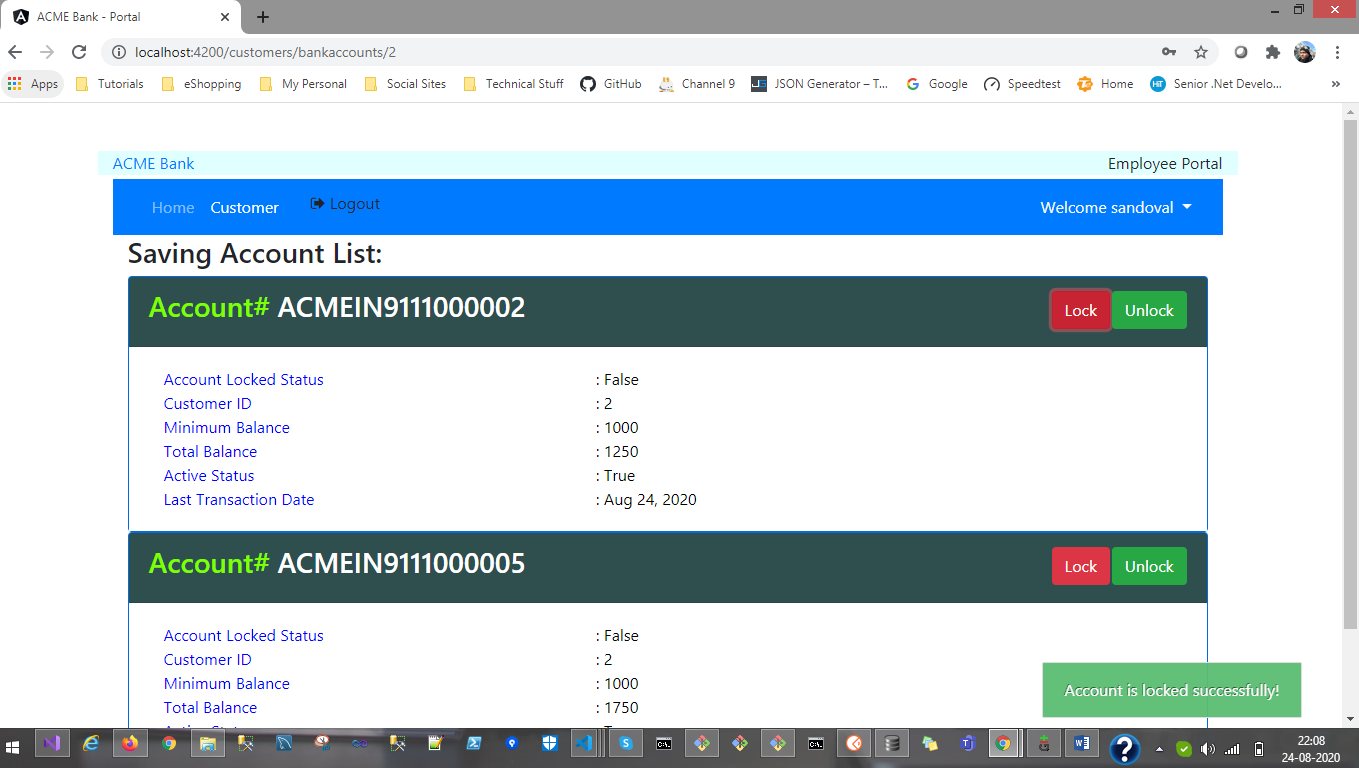
**Customer List:**



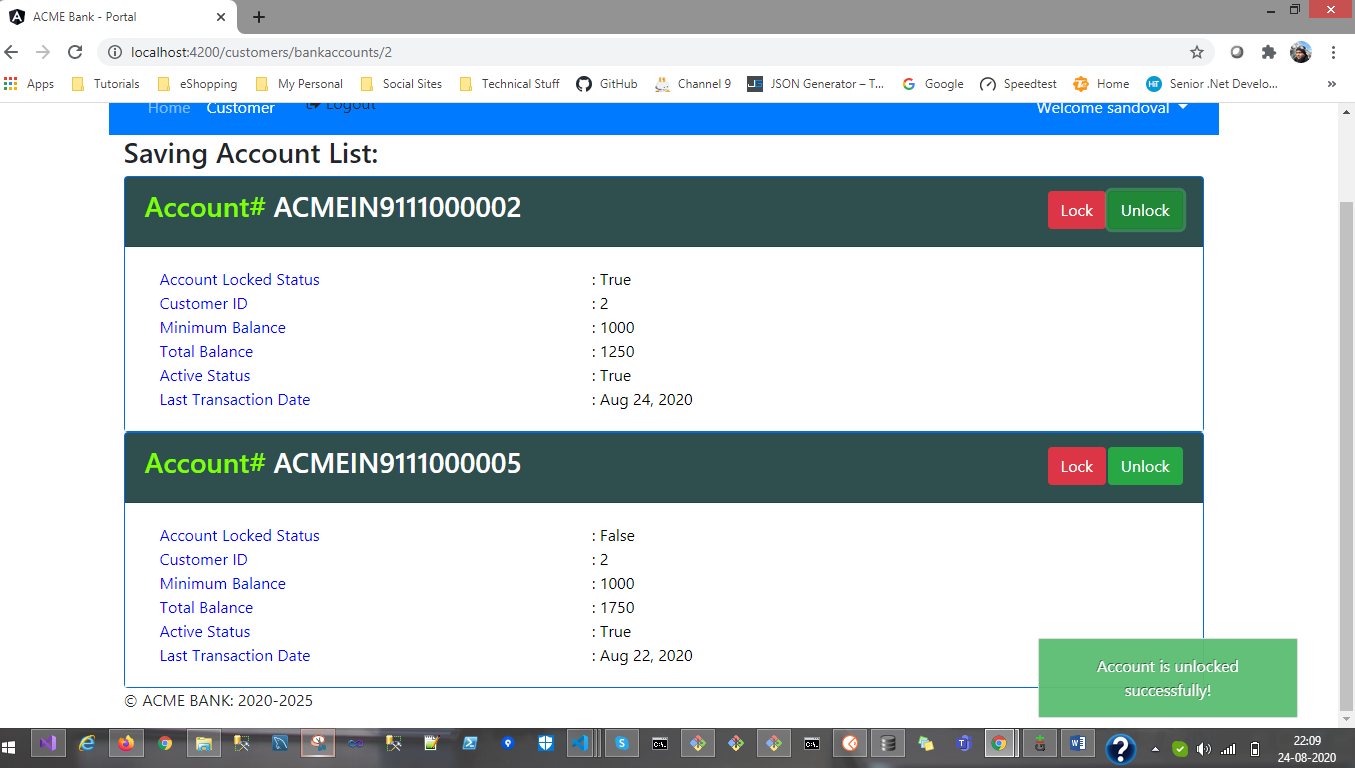
**View Saving Accounts (Customer ID# 2):**



**Lock Account:**



**Unlock Account:**



**CUSTOMER PORTAL**

**Design Approach:**

Similarly, there will be Customer Login and at run time if the login successful, then the system will only load the respective saving account(s) hold be logged-in customer.

Instead of Lock and Unlock button, there will be three buttons/links, as below:

**Deposit Money:** This screen will allow customer to enter the amount to their respective account number and also capture the transaction to Bank Transaction database table.

**Withdrawl Money:** If the logged-in customer account is not in locked state, the customer will be able to withdraw money with subject to limit to their total saving amount in that respective account.

**View Past Transactions:** This screen will be displaying the list of last transactions made to any respective account.

**Note:** I know there are few bugs on this UI and due to time issue and personal urgent work, I was not able to complete this functionality over this weekend.

Thank You,

Gaurav Lal