**Basic Identity Access Management Code**

class User:

def \_\_init\_\_(self, username, password):

self.username = username

self.password = password

self.roles = []

def add\_role(self, role):

self.roles.append(role)

def \_\_str\_\_(self):

return self.username

class IAMSystem:

def \_\_init\_\_(self):

self.users = {}

def create\_user(self, username, password):

if username not in self.users:

user = User(username, password)

self.users[username] = user

return user

else:

return None

def get\_user(self, username):

return self.users.get(username, None)

def authenticate(self, username, password):

user = self.get\_user(username)

if user and user.password == password:

return user

else:

return None

def grant\_role(self, username, role):

user = self.get\_user(username)

if user:

user.add\_role(role)

def check\_access(self, username, resource, required\_role):

user = self.get\_user(username)

if user and required\_role in user.roles:

print(f"{username} has access to {resource}")

else:

print(f"{username} does not have access to {resource}")

# Usage example:

if \_\_name\_\_ == "\_\_main\_\_":

iam = IAMSystem()

# Create users

user1 = iam.create\_user("user1", "password1")

user2 = iam.create\_user("user2", "password2")

# Authenticate users

authenticated\_user = iam.authenticate("user1", "password1")

if authenticated\_user