



P P SAVANI UNIVERSITY

ASSIGNMENT NO.: 7 ON BLOCKCHAIN TECHNOLOGY(SSCS3021)

TITLE: Car Trading Network using Hyperledger Composer

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY (BSC-IT)

SUBMITTED TO:

Name: MR. KAUSHAL SINGH(KVS)

Designation: ASSISTANT PROFESSOR

P P Savani University

SUBMITTED BY:

Name: RAJ MO FAHIM ZAKIR

Enrollment: 23SS02IT161

BSCIT5B-Batch 2023-26

Faculty Signature: _____

**INSTITUTE OF COMPUTER SCIENCE AND APPLICATIONS
P P SAVANI UNIVERSITY
MANGROL, SURAT- 394125 (GUJARAT)**



Student Name: RAJ MO FAHIM ZAKIR
Enrolment Number: 23SS02IT161
Subject Name: SOFTWARE ENGINEERING
Subject Code: SSCS3010

ASSIGNMENT-7

Date:24/09/2025

Aim: Car Trading Network using Hyperledger Composer

Step 1 — Model File

```
// SPDX-License-Identifier: Apache-2.0
namespace org.example.cartrading
participant Buyer identified by buyerId {
    o String buyerId
    o String name
}
participant Seller identified by sellerId {
    o String sellerId
    o String name
}
asset Car identified by carId {
    o String carId
    o String model
    o Double price
    --> Seller owner
}
transaction BuyCar {
    --> Car car
    --> Buyer newOwner
}
event CarPurchased {
    --> Car car
    --> Buyer newOwner
    --> Seller oldOwner
}
```

Step 2 — Logic File

```
/*
 * BuyCar transaction processor
 * @param {org.example.cartrading.BuyCar} tx
 * @transaction
 */
async function BuyCar(tx) {
    const car = tx.car;
    const oldOwner = car.owner;
    const newOwner = tx.newOwner;

    if (!newOwner) {
```

```

        throw new Error('Buyer not specified');
    }

    // Transfer ownership
    car.owner = newOwner;

    // Update car in registry
    const assetRegistry = await getAssetRegistry('org.example.cartrading.Car');
    await assetRegistry.update(car);

    // Emit CarPurchased event
    const factory = getFactory();
    const event = factory.newEvent('org.example.cartrading', 'CarPurchased');
    event.car = car;
    event.newOwner = newOwner;
    event.oldOwner = oldOwner;
    emit(event);
}

```

Step 3 — Access Control (ACL File)

// SPDX-License-Identifier: Apache-2.0

```

rule Default {
    description: "Allow all participants read access"
    participant: "ANY"
    operation: READ
    resource: "ANY"
    action: ALLOW
}
rule SellerCanCreate {
    description: "Only Seller can add cars"
    participant(p): "org.example.cartrading.Seller"
    operation: CREATE
    resource(r): "org.example.cartrading.Car"
    condition: (r.owner.getIdentifier() === p.getIdentifier())
    action: ALLOW
}
rule BuyerCannotCreateCar {
    description: "Buyers cannot create cars"
    participant(p): "org.example.cartrading.Buyer"
    operation: CREATE
    resource: "org.example.cartrading.Car"
    action: DENY
}

```



Student Name: RAJ MO FAHIM ZAKIR
Enrolment Number: 23SS02IT161
Subject Name: SOFTWARE ENGINEERING
Subject Code: SSCS3010

}

Step 4 — Testing in Composer Playground

Participants → Add

Seller → { "sellerId": "Seller1", "name": "Alice" }

Buyer → { "buyerId": "Buyer1", "name": "Bob" }

Assets → Add Car

```
{  
  "$class": "org.example.cartrading.Car",  
  "carId": "Car1",  
  "model": "Tesla Model S",  
  "price": 75000,  
  "owner": "resource:org.example.cartrading.Seller#Seller1"  
}
```

Transactions → Submit BuyCar

```
{  
  "$class": "org.example.cartrading.BuyCar",  
  "car": "resource:org.example.cartrading.Car#Car1",  
  "newOwner": "resource:org.example.cartrading.Buyer#Buyer1"  
}
```

Expected Results:

Before Transaction:

Car1.owner = Seller1

After Transaction:

Car1.owner = Buyer1

ACL Check:

If a Buyer tries to create a Car → PermissionDenied error.



Student Name: RAJ MO FAHIM ZAKIR
Enrolment Number: 23SS02IT161
Subject Name: SOFTWARE ENGINEERING
Subject Code: SSCS3010

asset-network

admin

Assets

Add asset

ID	description	Current state
car1	Blue Honda	resource:org.example.assets.User#user1
		user1