ASSET MANAGEMENT COMPLETE EXPLANATION

1. Entity classes corresponding to the schema are created within the package entity with variables declared private, constructors(default and parametrized) and getters,setters methods.

The created Entities are:

* Assets

* Employees

* Asset_allocations

* Maintainence_records

* Reservations

2.An Interface is created with the name **AssetManagementService** and provide contract for all methods.

3.An implementation class is created for Interface with name **AssetManagementServiceImpl** and all methods are implemented.

3.1.Add Asset

Implementation:

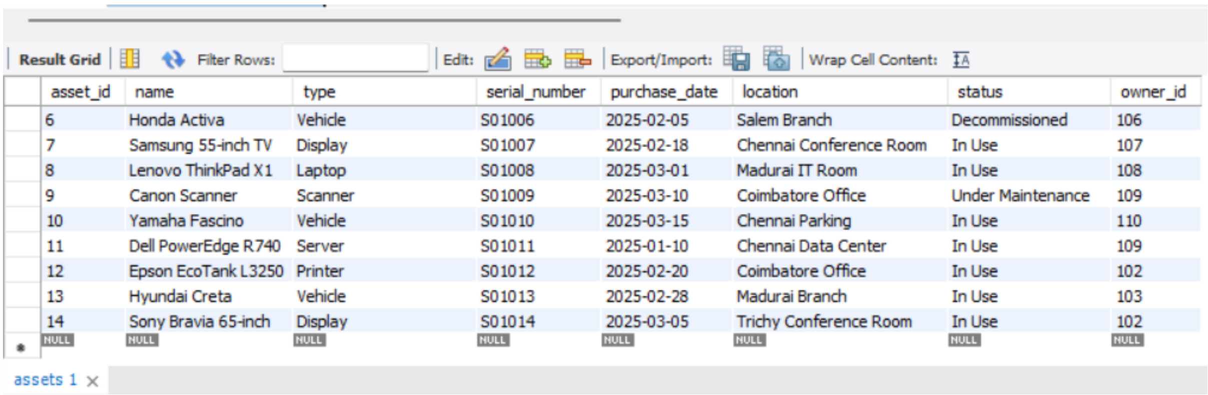
```
@Override
public boolean addAsset(Assets assets) throws SQLException,
ParseException {

    boolean added=false;
    // Insert obtained datas into database
    PreparedStatement pst = connection.prepareStatement("INSERT
INTO assets (asset_id, name, type, serial_number, purchase_date,
location, status) VALUES (?, ?, ?, ?, ?, ?, ?)");

    pst.setInt(1, assets.getAsset_id());
    pst.setString(2, assets.getAsset_name());
    pst.setString(3, assets.getAsset_type());
    pst.setString(4, assets.getSerial_number());
    pst.setDate(5,  assets.getPurchase_date());
    pst.setString(6, assets.getLocation());
    pst.setString(7, assets.getStatus());
    //pst.setInt(8, newOwnerId);
    //Can't able to insert owner_id because it is foreign key

    int rows = pst.executeUpdate();
    if (rows > 0) {
        added=true;
    }

    return added;
}
```

Main

```
if (option == 1) {
    // Get next asset ID
    int newAssetId = service.getNextID("assets", "asset_id");

    // Read input from user
    System.out.print("Enter Asset Name: ");
    String name = sc.nextLine();

    System.out.print("Enter Asset Type: ");
    String type = sc.nextLine();

    // Generate new serial number
    String newSerial = service.generateNextSerialNumber();

    System.out.print("Enter Purchase Date (YYYY-MM-DD): ");
    String purchaseDate = sc.nextLine();

    //TO convert String to Date
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Date utilDate = sdf.parse(purchaseDate);
    java.sql.Date sqlDate = new java.sql.Date(utilDate.getTime());

    System.out.print("Enter Location: ");
    String location = sc.nextLine();

    System.out.print("Enter Status: ");
    String status = sc.nextLine();
    Assets assets = new Assets(newAssetId, name, type, newSerial,
sqlDate, location, status, 0);
    boolean check = service.addAsset(assets);
    if (check) {
        System.out.println("Asset added successfully with ID: " +
assets.getAsset_id());
    } else {
        System.out.println("Asset Cannot be Added Due to Server
Issues");
    }
}
```

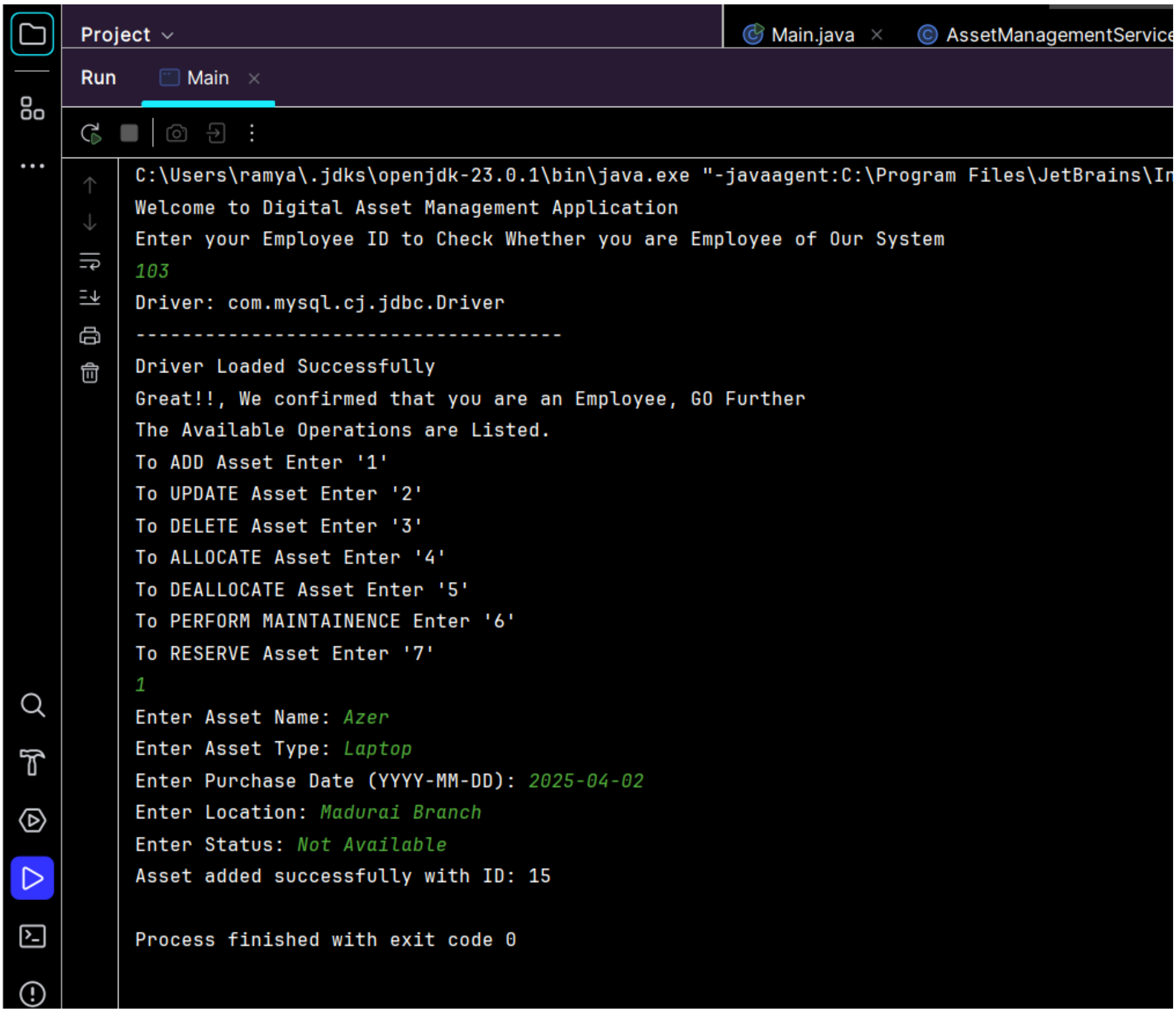
OUTPUT

DATABASE BEFORE

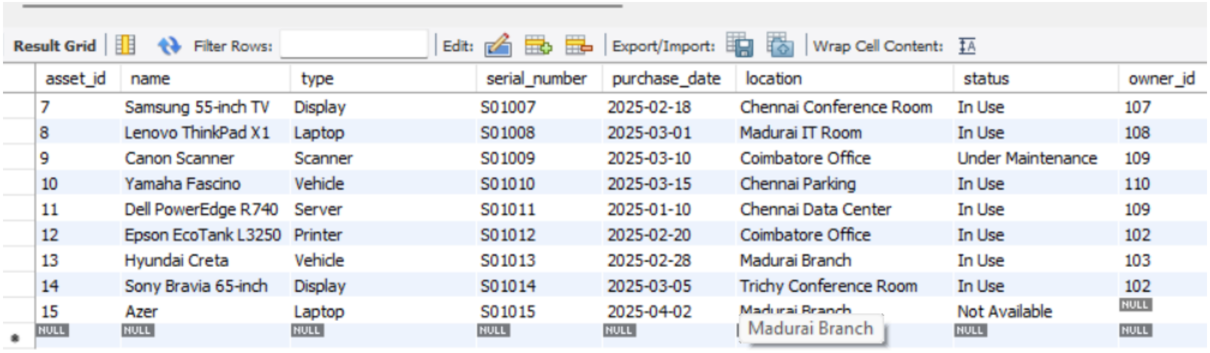


	asset_id	name	type	serial_number	purchase_date	location	status	owner_id
	6	Honda Activa	Vehicle	S01006	2025-02-05	Salem Branch	Decommissioned	106
	7	Samsung 55-inch TV	Display	S01007	2025-02-18	Chennai Conference Room	In Use	107
	8	Lenovo ThinkPad X1	Laptop	S01008	2025-03-01	Madurai IT Room	In Use	108
	9	Canon Scanner	Scanner	S01009	2025-03-10	Coimbatore Office	Under Maintenance	109
	10	Yamaha Fascino	Vehicle	S01010	2025-03-15	Chennai Parking	In Use	110
	11	Dell PowerEdge R740	Server	S01011	2025-01-10	Chennai Data Center	In Use	109
	12	Epson EcoTank L3250	Printer	S01012	2025-02-20	Coimbatore Office	In Use	102
	13	Hyundai Creta	Vehicle	S01013	2025-02-28	Madurai Branch	In Use	103
	14	Sony Bravia 65-inch	Display	S01014	2025-03-05	Trichy Conference Room	In Use	102
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Here, 14 records are present



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\In
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
103
Driver: com.mysql.cj.jdbc.Driver
-------------------------------------
Driver Loaded Successfully
Great!!, We confirmed that you are an Employee, GO Further
The Available Operations are Listed.
To ADD Asset Enter '1'
To UPDATE Asset Enter '2'
To DELETE Asset Enter '3'
To ALLOCATE Asset Enter '4'
To DEALLOCATE Asset Enter '5'
To PERFORM MAINTAINENCE Enter '6'
To RESERVE Asset Enter '7'
1
Enter Asset Name: Azer
Enter Asset Type: Laptop
Enter Purchase Date (YYYY-MM-DD): 2025-04-02
Enter Location: Madurai Branch
Enter Status: Not Available
Asset added successfully with ID: 15

Process finished with exit code 0
```



asset_id	name	type	serial_number	purchase_date	location	status	owner_id
7	Samsung 55-inch TV	Display	S01007	2025-02-18	Chennai Conference Room	In Use	107
8	Lenovo ThinkPad X1	Laptop	S01008	2025-03-01	Madurai IT Room	In Use	108
9	Canon Scanner	Scanner	S01009	2025-03-10	Coimbatore Office	Under Maintenance	109
10	Yamaha Fascino	Vehicle	S01010	2025-03-15	Chennai Parking	In Use	110
11	Dell PowerEdge R740	Server	S01011	2025-01-10	Chennai Data Center	In Use	109
12	Epson EcoTank L3250	Printer	S01012	2025-02-20	Coimbatore Office	In Use	102
13	Hyundai Creta	Vehicle	S01013	2025-02-28	Madurai Branch	In Use	103
14	Sony Bravia 65-inch	Display	S01014	2025-03-05	Trichy Conference Room	In Use	102
15	Azer	Laptop	S01015	2025-04-02	Madurai Branch	Not Available	NULL
NULL	NULL	NULL	NULL	NULL	Madurai Branch	NULL	NULL

DATABASE AFTER

15 records are present

3.2.Update Asset

Implementation:

//To update New Status for Asset
@Override

```
public boolean updateAsset(Assets assets) throws SQLException {

   boolean updated=false;
   PreparedStatement pst = connection.prepareStatement("UPDATE
assets SET status = ? WHERE asset_id = ?");
   pst.setString(1, assets.getStatus());
   pst.setInt(2, assets.getAsset_id());
   int row = pst.executeUpdate();
   if (row > 0) {
      updated=true;
   }
  return updated;
}
```

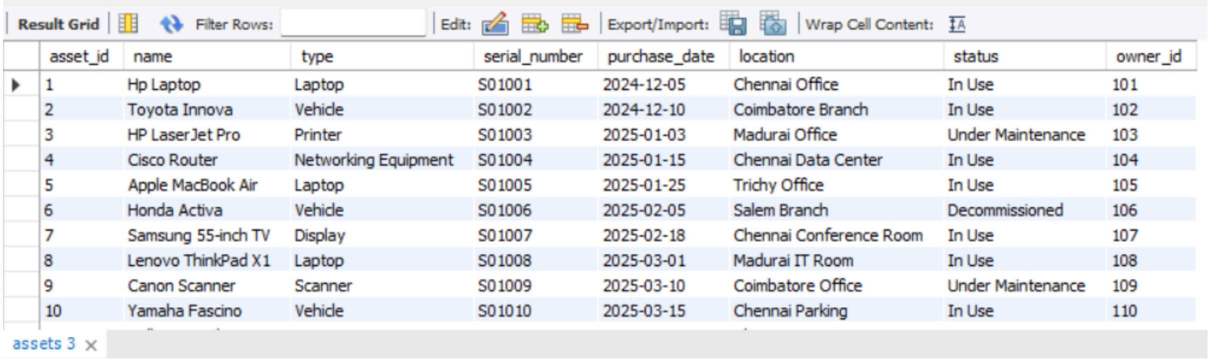
Main

```
else if (option == 2) {
   System.out.println("Enter Asset ID:");
   int aid = sc.nextInt();
   sc.nextLine();
   System.out.println("Enter New Status to be Updated:");
   String newstatus = sc.nextLine();
   Assets assets = new Assets(aid, null, null, null, null, null,
newstatus, 0);
   boolean check = service.updateAsset(assets);
   if (check) {
      System.out.println("Asset Status updated successfully.");
   } else {
      System.out.println("Asset not found.Enter Correct ID");
   }

}
```

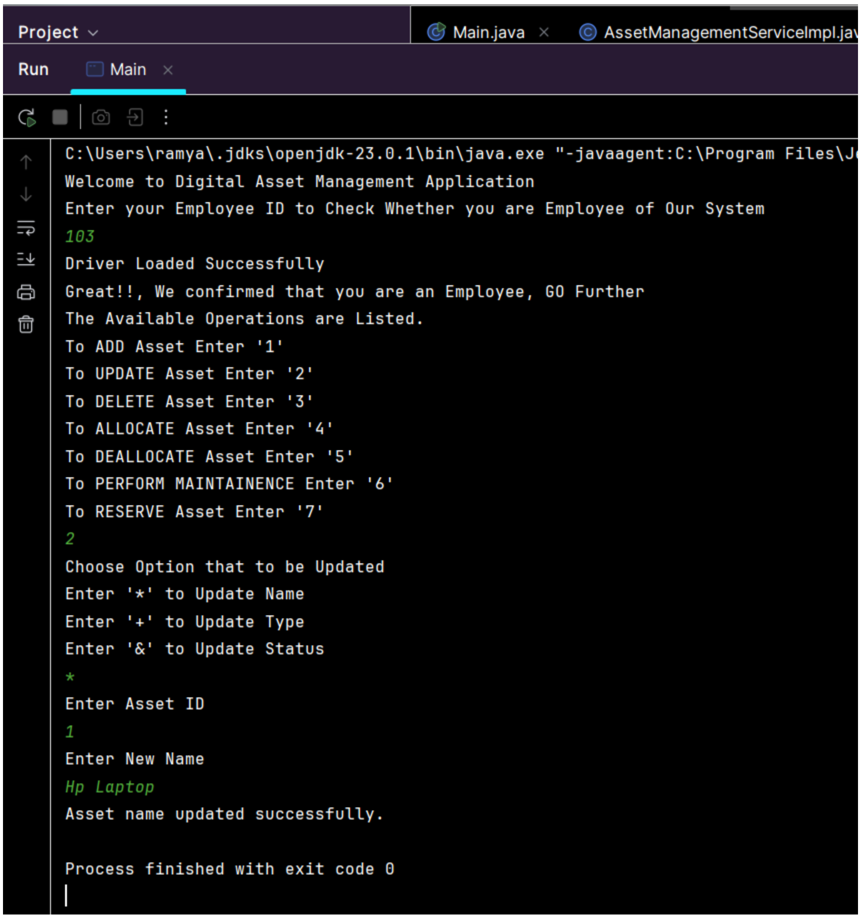
OUTPUT

DATABASE BEFORE

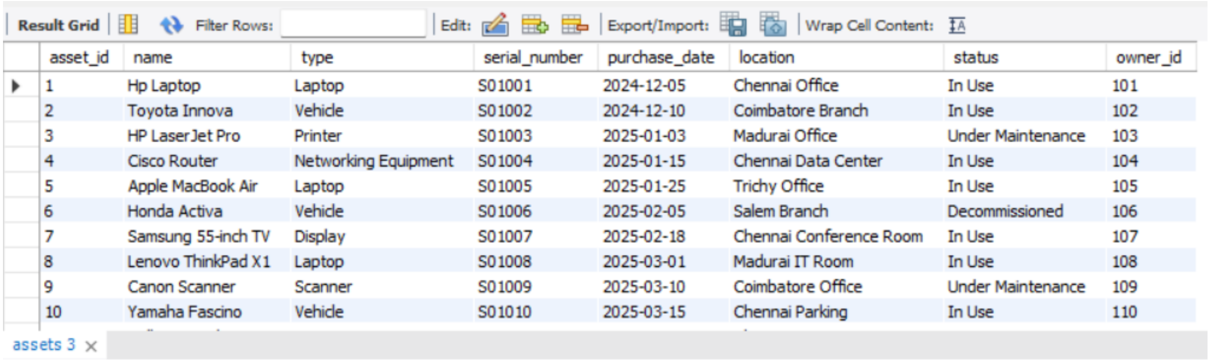
Here Name is updated from dell to hp laptop in id 1



asset_id	name	type	serial_number	purchase_date	location	status	owner_id
1	Hp Laptop	Laptop	S01001	2024-12-05	Chennai Office	In Use	101
2	Toyota Innova	Vehicle	S01002	2024-12-10	Coimbatore Branch	In Use	102
3	HP LaserJet Pro	Printer	S01003	2025-01-03	Madurai Office	Under Maintenance	103
4	Cisco Router	Networking Equipment	S01004	2025-01-15	Chennai Data Center	In Use	104
5	Apple MacBook Air	Laptop	S01005	2025-01-25	Trichy Office	In Use	105
6	Honda Activa	Vehicle	S01006	2025-02-05	Salem Branch	Decommissioned	106
7	Samsung 55-inch TV	Display	S01007	2025-02-18	Chennai Conference Room	In Use	107
8	Lenovo ThinkPad X1	Laptop	S01008	2025-03-01	Madurai IT Room	In Use	108
9	Canon Scanner	Scanner	S01009	2025-03-10	Coimbatore Office	Under Maintenance	109
10	Yamaha Fascino	Vehicle	S01010	2025-03-15	Chennai Parking	In Use	110

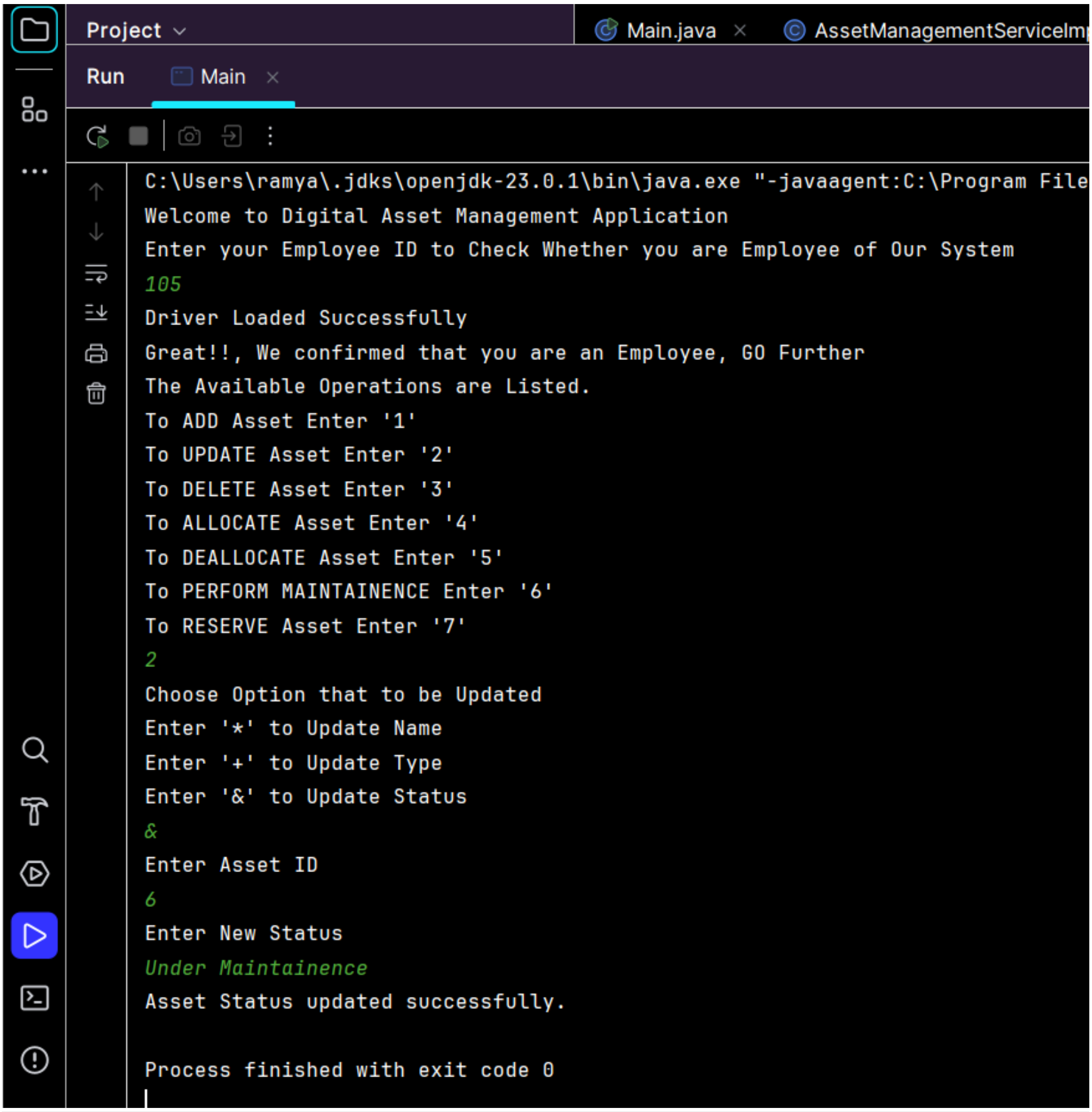


```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\J
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
103
Driver Loaded Successfully
Great!!, We confirmed that you are an Employee, GO Further
The Available Operations are Listed.
To ADD Asset Enter '1'
To UPDATE Asset Enter '2'
To DELETE Asset Enter '3'
To ALLOCATE Asset Enter '4'
To DEALLOCATE Asset Enter '5'
To PERFORM MAINTAINENCE Enter '6'
To RESERVE Asset Enter '7'
2
Choose Option that to be Updated
Enter '*' to Update Name
Enter '+' to Update Type
Enter '&' to Update Status
*
Enter Asset ID
1
Enter New Name
Hp Laptop
Asset name updated successfully.

Process finished with exit code 0
```

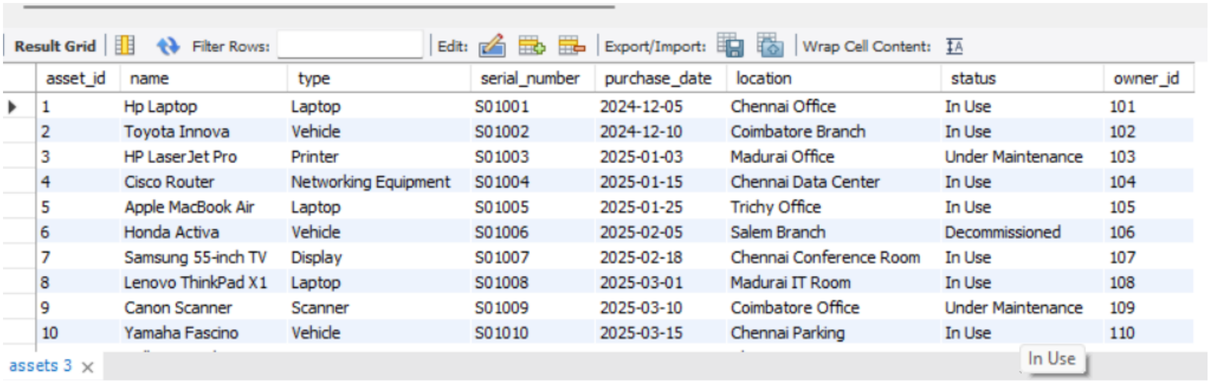


asset_id	name	type	serial_number	purchase_date	location	status	owner_id
1	Hp Laptop	Laptop	S01001	2024-12-05	Chennai Office	In Use	101
2	Toyota Innova	Vehicle	S01002	2024-12-10	Coimbatore Branch	In Use	102
3	HP LaserJet Pro	Printer	S01003	2025-01-03	Madurai Office	Under Maintenance	103
4	Cisco Router	Networking Equipment	S01004	2025-01-15	Chennai Data Center	In Use	104
5	Apple MacBook Air	Laptop	S01005	2025-01-25	Trichy Office	In Use	105
6	Honda Activa	Vehicle	S01006	2025-02-05	Salem Branch	Decommissioned	106
7	Samsung 55-inch TV	Display	S01007	2025-02-18	Chennai Conference Room	In Use	107
8	Lenovo ThinkPad X1	Laptop	S01008	2025-03-01	Madurai IT Room	In Use	108
9	Canon Scanner	Scanner	S01009	2025-03-10	Coimbatore Office	Under Maintenance	109
10	Yamaha Fascino	Vehicle	S01010	2025-03-15	Chennai Parking	In Use	110

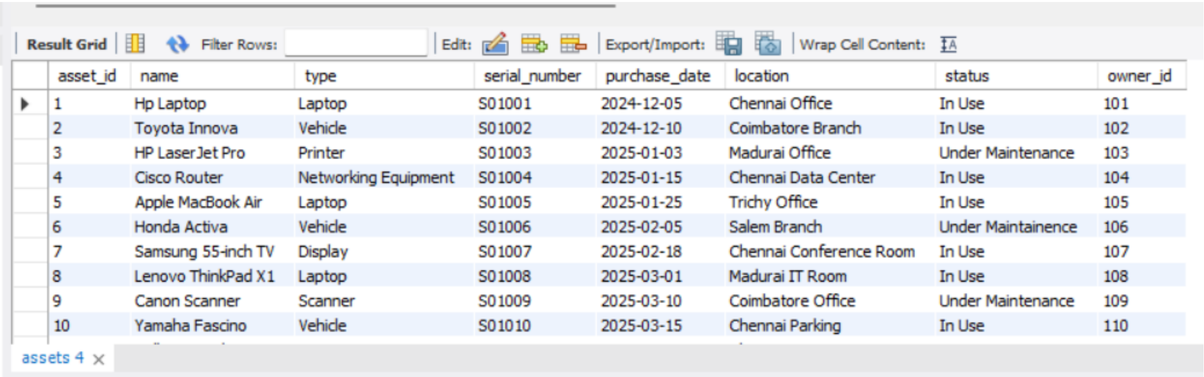
Here Status is updated from decommonished to Under Maintainence



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program File
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
105
Driver Loaded Successfully
Great!!, We confirmed that you are an Employee, GO Further
The Available Operations are Listed.
To ADD Asset Enter '1'
To UPDATE Asset Enter '2'
To DELETE Asset Enter '3'
To ALLOCATE Asset Enter '4'
To DEALLOCATE Asset Enter '5'
To PERFORM MAINTAINENCE Enter '6'
To RESERVE Asset Enter '7'
2
Choose Option that to be Updated
Enter '*' to Update Name
Enter '+' to Update Type
Enter '&' to Update Status
&
Enter Asset ID
6
Enter New Status
Under Maintainence
Asset Status updated successfully.

Process finished with exit code 0
```



asset_id	name	type	serial_number	purchase_date	location	status	owner_id
1	Hp Laptop	Laptop	S01001	2024-12-05	Chennai Office	In Use	101
2	Toyota Innova	Vehicle	S01002	2024-12-10	Coimbatore Branch	In Use	102
3	HP LaserJet Pro	Printer	S01003	2025-01-03	Madurai Office	Under Maintenance	103
4	Cisco Router	Networking Equipment	S01004	2025-01-15	Chennai Data Center	In Use	104
5	Apple MacBook Air	Laptop	S01005	2025-01-25	Trichy Office	In Use	105
6	Honda Activa	Vehicle	S01006	2025-02-05	Salem Branch	Decommissioned	106
7	Samsung 55-inch TV	Display	S01007	2025-02-18	Chennai Conference Room	In Use	107
8	Lenovo ThinkPad X1	Laptop	S01008	2025-03-01	Madurai IT Room	In Use	108
9	Canon Scanner	Scanner	S01009	2025-03-10	Coimbatore Office	Under Maintenance	109
10	Yamaha Fascino	Vehicle	S01010	2025-03-15	Chennai Parking	In Use	110



asset_id	name	type	serial_number	purchase_date	location	status	owner_id
1	Hp Laptop	Laptop	S01001	2024-12-05	Chennai Office	In Use	101
2	Toyota Innova	Vehicle	S01002	2024-12-10	Coimbatore Branch	In Use	102
3	HP LaserJet Pro	Printer	S01003	2025-01-03	Madurai Office	Under Maintenance	103
4	Cisco Router	Networking Equipment	S01004	2025-01-15	Chennai Data Center	In Use	104
5	Apple MacBook Air	Laptop	S01005	2025-01-25	Trichy Office	In Use	105
6	Honda Activa	Vehicle	S01006	2025-02-05	Salem Branch	Under Maintainence	106
7	Samsung 55-inch TV	Display	S01007	2025-02-18	Chennai Conference Room	In Use	107
8	Lenovo ThinkPad X1	Laptop	S01008	2025-03-01	Madurai IT Room	In Use	108
9	Canon Scanner	Scanner	S01009	2025-03-10	Coimbatore Office	Under Maintenance	109
10	Yamaha Fascino	Vehicle	S01010	2025-03-15	Chennai Parking	In Use	110

3.3 Delete Asset

Implementation

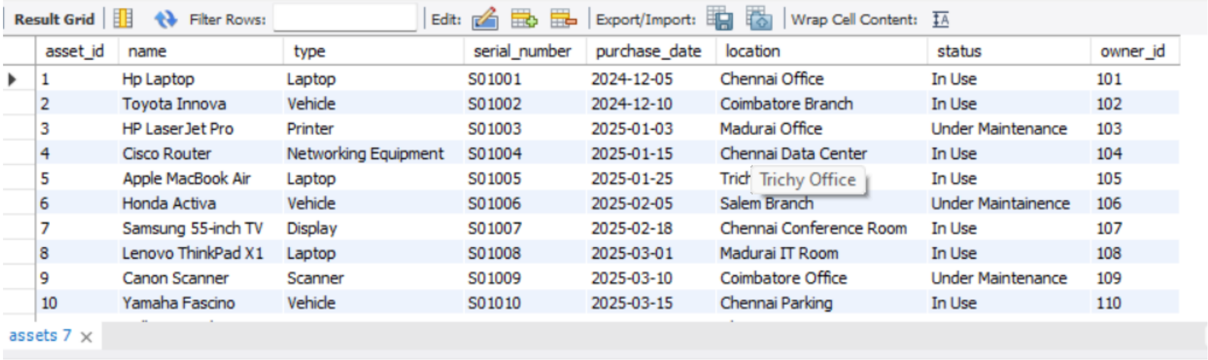
```
//To delete Asset
@Override
public boolean deleteAsset(int assetid) throws SQLException {
    boolean deleted=false;
    PreparedStatement pst = connection.prepareStatement("delete from
assets where asset_id=?");
    pst.setInt(1,assetid);
    int rows= pst.executeUpdate();
    if(rows>0)
    {
        deleted=true;
    }
    return deleted;

}
```

Main

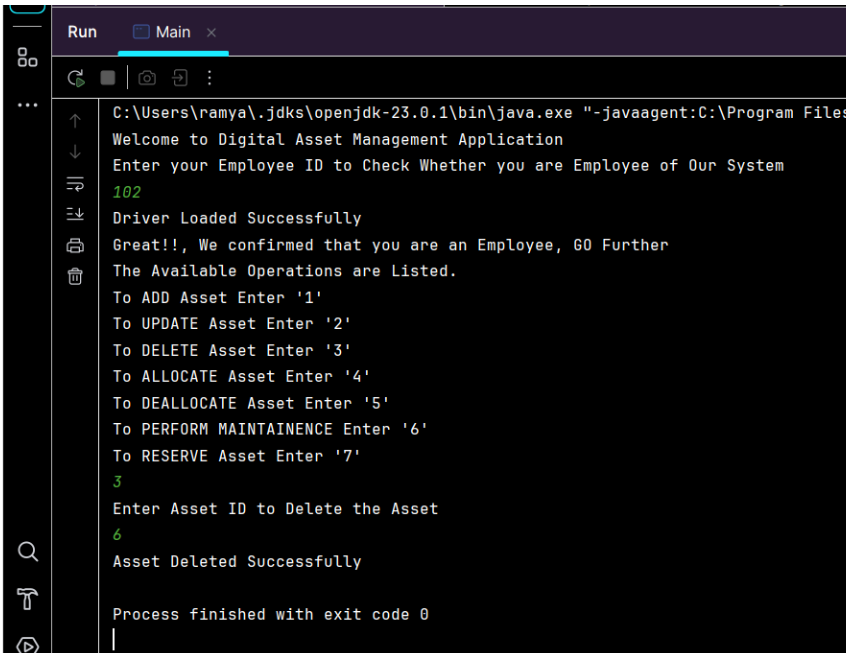
```
else if (option == 3) {
    System.out.println("Enter Asset ID to Delete the Asset");
    int aid = sc.nextInt();
    //Handles AssetNotFoundException
    try {
        service.checkAssetID(aid);
    } catch (AssetNotFoundException e) {
        System.out.println(e.getMessage()); // Prints custom message
    }

    boolean check = service.deleteAsset(aid);
    if (check) {
        System.out.println("Asset Deleted Successfully");
    } else {
        System.out.println("Asset Not Found.Please Check the Asset
ID");
    }

}
```

OUTPUT

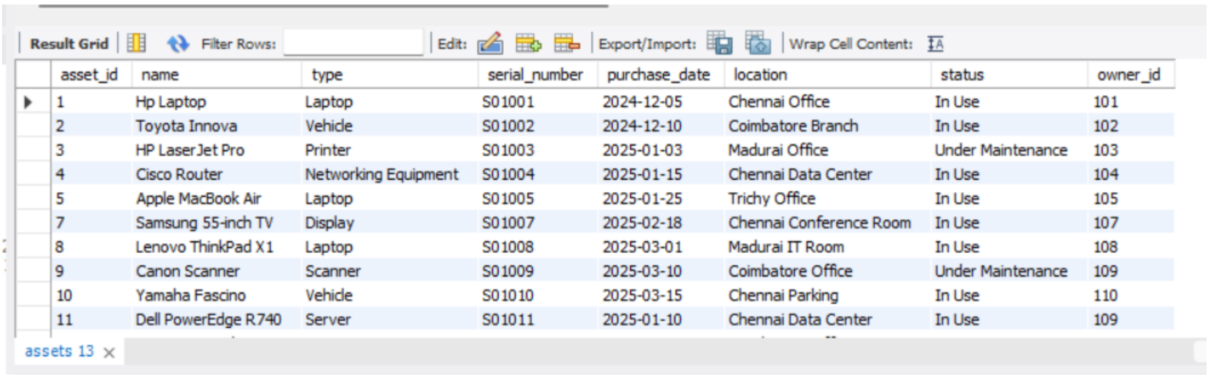
BEFORE DATABASE


Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
assets 7

asset_id	name	type	serial_number	purchase_date	location	status	owner_id
1	Hp Laptop	Laptop	S01001	2024-12-05	Chennai Office	In Use	101
2	Toyota Innova	Vehicle	S01002	2024-12-10	Coimbatore Branch	In Use	102
3	HP LaserJet Pro	Printer	S01003	2025-01-03	Madurai Office	Under Maintenance	103
4	Cisco Router	Networking Equipment	S01004	2025-01-15	Chennai Data Center	In Use	104
5	Apple MacBook Air	Laptop	S01005	2025-01-25	Trichy Office	In Use	105
6	Honda Activa	Vehicle	S01006	2025-02-05	Salem Branch	Under Maintainence	106
7	Samsung 55-inch TV	Display	S01007	2025-02-18	Chennai Conference Room	In Use	107
8	Lenovo ThinkPad X1	Laptop	S01008	2025-03-01	Madurai IT Room	In Use	108
9	Canon Scanner	Scanner	S01009	2025-03-10	Coimbatore Office	Under Maintenance	109
10	Yamaha Fascino	Vehicle	S01010	2025-03-15	Chennai Parking	In Use	110



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
102
Driver Loaded Successfully
Great!!, We confirmed that you are an Employee, GO Further
The Available Operations are Listed.
To ADD Asset Enter '1'
To UPDATE Asset Enter '2'
To DELETE Asset Enter '3'
To ALLOCATE Asset Enter '4'
To DEALLOCATE Asset Enter '5'
To PERFORM MAINTAINENCE Enter '6'
To RESERVE Asset Enter '7'
3
Enter Asset ID to Delete the Asset
6
Asset Deleted Successfully

Process finished with exit code 0
```



asset_id	name	type	serial_number	purchase_date	location	status	owner_id
1	Hp Laptop	Laptop	S01001	2024-12-05	Chennai Office	In Use	101
2	Toyota Innova	Vehicle	S01002	2024-12-10	Coimbatore Branch	In Use	102
3	HP LaserJet Pro	Printer	S01003	2025-01-03	Madurai Office	Under Maintenance	103
4	Cisco Router	Networking Equipment	S01004	2025-01-15	Chennai Data Center	In Use	104
5	Apple MacBook Air	Laptop	S01005	2025-01-25	Trichy Office	In Use	105
7	Samsung 55-inch TV	Display	S01007	2025-02-18	Chennai Conference Room	In Use	107
8	Lenovo ThinkPad X1	Laptop	S01008	2025-03-01	Madurai IT Room	In Use	108
9	Canon Scanner	Scanner	S01009	2025-03-10	Coimbatore Office	Under Maintenance	109
10	Yamaha Fascino	Vehicle	S01010	2025-03-15	Chennai Parking	In Use	110
11	Dell PowerEdge R740	Server	S01011	2025-01-10	Chennai Data Center	In Use	109

Asset 6 is deleted from DB

3.4 Allocate Asset

Implementation

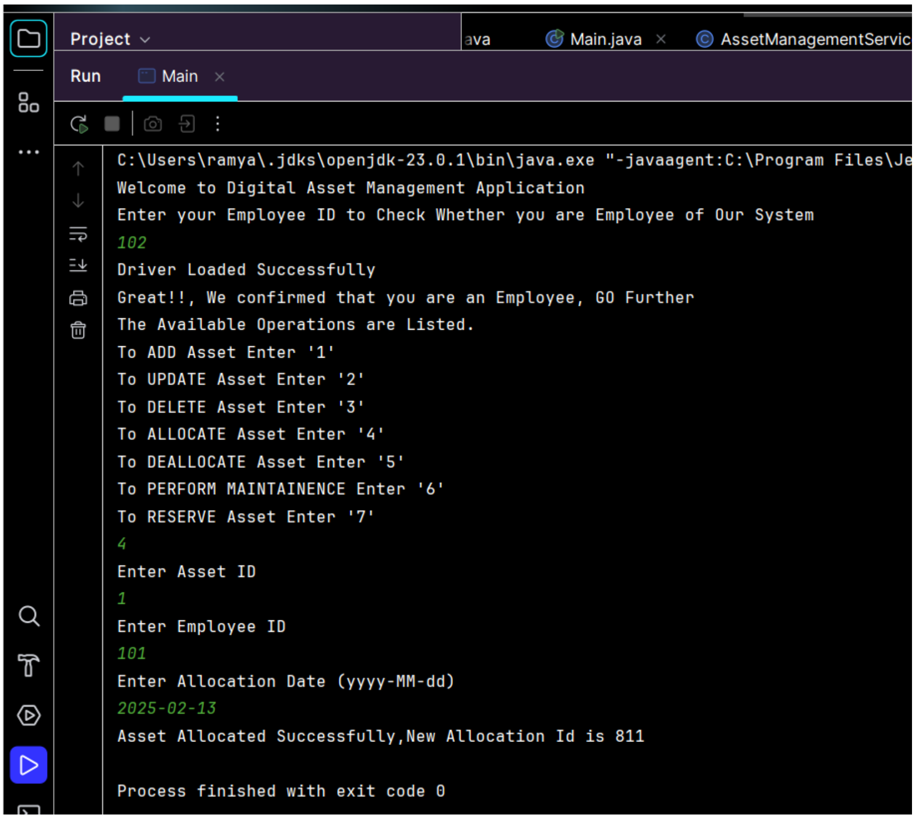
```
//To allocate asset
@Override
public boolean allocateAsset(int assetId, int employeeId, String
allocationDate) throws SQLException {

    // Check if the asset ID exists
    PreparedStatement checkAsset =
connection.prepareStatement("SELECT COUNT(*) FROM assets
WHERE asset_id = ?");
    checkAsset.setInt(1, assetId);
    ResultSet rsAsset = checkAsset.executeQuery();
    if (rsAsset.next() && rsAsset.getInt(1) == 0) {
        System.out.println("Asset ID " + assetId + " does not exist.");
        return false;
    }

    // Check if the employee ID exists
    PreparedStatement checkEmployee =
connection.prepareStatement("SELECT COUNT(*) FROM
employees WHERE employee_id = ?");
    checkEmployee.setInt(1, employeeId);
    ResultSet rsEmp = checkEmployee.executeQuery();
    if (rsEmp.next() && rsEmp.getInt(1) == 0) {
        System.out.println("Employee ID " + employeeId + " does not
exist.");
        return false;
    }
    //To get new allocationid from database
    int
allocation_id=getNewAllocationId("asset_allocations","allocation_id"
);
    DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("yyyy-MM-dd");
    LocalDate allocDate = LocalDate.parse(allocationDate, formatter);

    // Calculate return date
    LocalDate returnDate = allocDate.plusDays(10);
    boolean allocated=false;
    PreparedStatement pst = connection.prepareStatement("Insert into
asset_allocations (asset_id,allocation_id,employee_id,
allocation_date, return_date)values(?,?,?,?,?)");
    pst.setInt(1,assetId);
    pst.setInt(2,allocation_id);
    pst.setInt(3,employeeId);
    pst.setDate(4, Date.valueOf(allocDate));
    pst.setDate(5, Date.valueOf(returnDate));
    int rows = pst.executeUpdate();
    if (rows > 0) {
        allocated=true;
    }

    return allocated;
}
```

Main

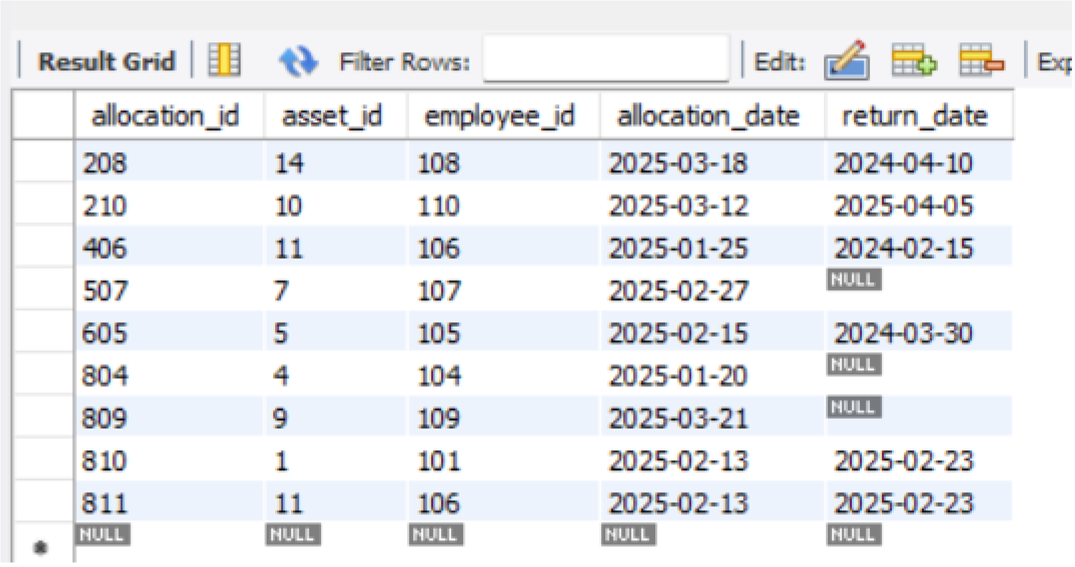
```
else if (option == 4) {
    System.out.println("Enter Asset ID");
    int assetid = sc.nextInt();
```

```
   sc.nextLine();
   System.out.println("Enter Employee ID");
   int employeeid = sc.nextInt();
   sc.nextLine();
   System.out.println("Enter Allocation Date (yyyy-MM-dd)");
   String allocationDate = sc.nextLine();

   //generating allocation id from Database
   boolean allocated = service.allocateAsset(assetid, employeeid,
allocationDate);
   if (allocated) {
      System.out.println("Asset Allocated Successfully,New
Allocation Id is " + service.getNewAllocationId("asset_allocations",
"allocation_id"));
   }

}
```

OUTPUT



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\Je
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
102
Driver Loaded Successfully
Great!!, We confirmed that you are an Employee, GO Further
The Available Operations are Listed.
To ADD Asset Enter '1'
To UPDATE Asset Enter '2'
To DELETE Asset Enter '3'
To ALLOCATE Asset Enter '4'
To DEALLOCATE Asset Enter '5'
To PERFORM MAINTAINENCE Enter '6'
To RESERVE Asset Enter '7'
4
Enter Asset ID
1
Enter Employee ID
101
Enter Allocation Date (yyyy-MM-dd)
2025-02-13
Asset Allocated Successfully,New Allocation Id is 811

Process finished with exit code 0
```



	allocation_id	asset_id	employee_id	allocation_date	return_date
	208	14	108	2025-03-18	2024-04-10
	210	10	110	2025-03-12	2025-04-05
	406	11	106	2025-01-25	2024-02-15
	507	7	107	2025-02-27	NULL
	605	5	105	2025-02-15	2024-03-30
	804	4	104	2025-01-20	NULL
	809	9	109	2025-03-21	NULL
	810	1	101	2025-02-13	2025-02-23
	811	11	106	2025-02-13	2025-02-23
*	NULL	NULL	NULL	NULL	NULL

Asset allocated with allocation ID 811

3.5 Deallocate Asset

Implementation

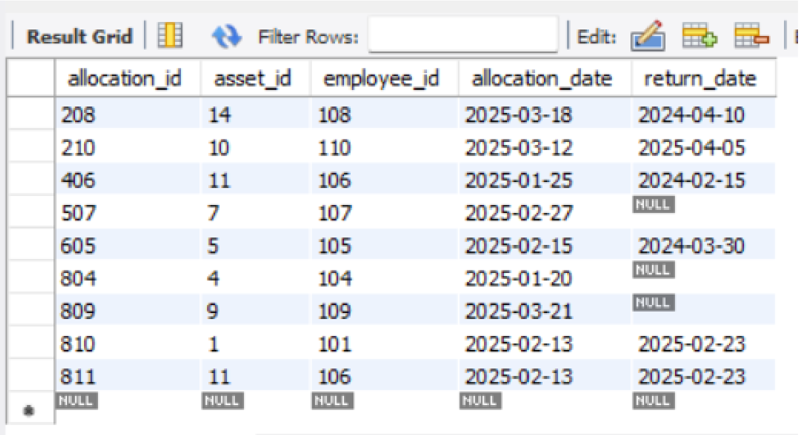
```
//To deallocate asset
@Override
public boolean deallocateAsset(int assetId, int employeeId, String
returnDate) throws SQLException {
    boolean deallocate=false;
    LocalDate retDate = LocalDate.parse(returnDate);

    PreparedStatement pst = connection.prepareStatement(
            "UPDATE asset_allocations SET return_date = ? WHERE
asset_id = ? AND employee_id = ?"
    );
    pst.setDate(1, Date.valueOf(retDate));
    pst.setInt(2, assetId);
    pst.setInt(3, employeeId);

    int rows = pst.executeUpdate();
    if(rows>0)
    {
        deallocate=true;
    }
    return  deallocate;
}
```

Main

```
else if (option == 5) {
    System.out.print("Enter Asset ID: ");
    int assetId = sc.nextInt();
    System.out.print("Enter Employee ID: ");
    int empId = sc.nextInt();
    System.out.print("Enter Return Date (yyyy-MM-dd): ");
    String retDate = sc.next();
    boolean deallocated = service.deallocateAsset(assetId, empId,
retDate);
    if (deallocated) {
        System.out.println("Asset Deallocated Successfully");
    } else {
        System.out.println("Deallocation Failed Due to Invalid Entry");
    }

}
```

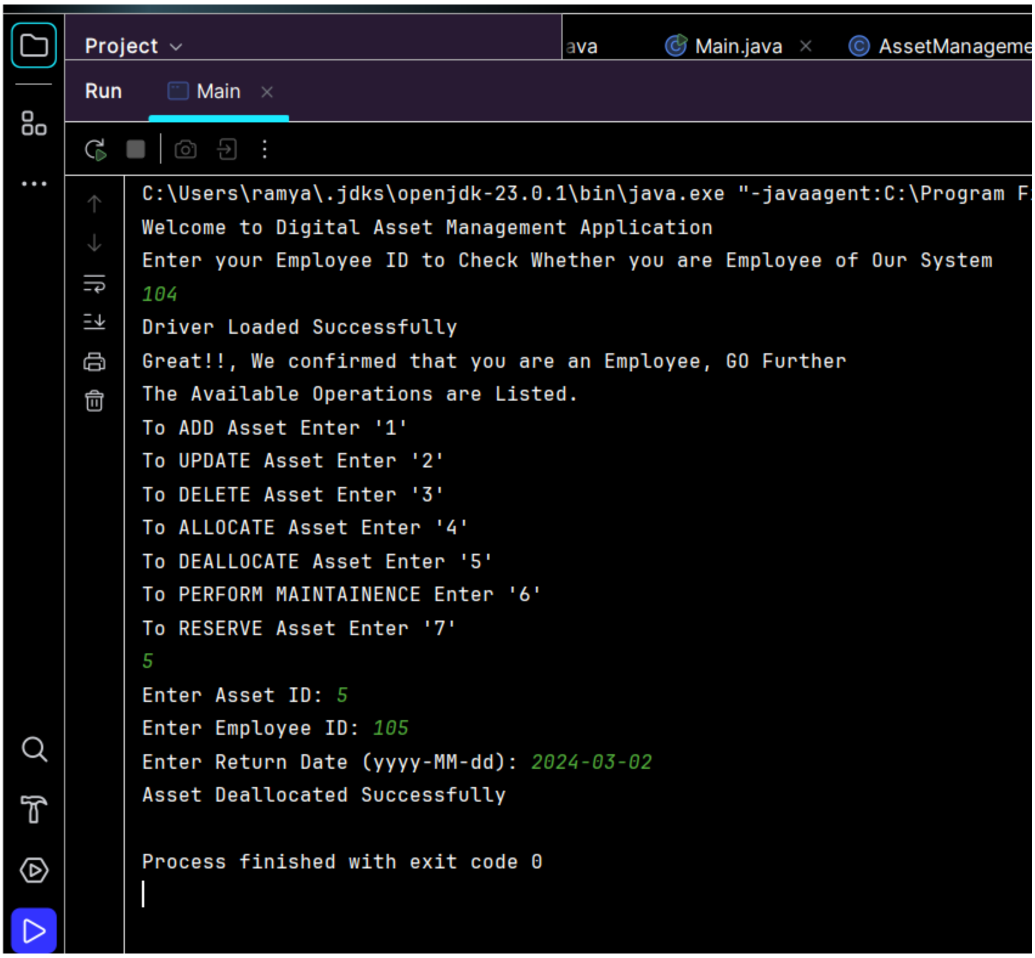
OUTPUT

Here for asset id 5 and employee id 105 the fixed given deallocation return date was 2024-03-30

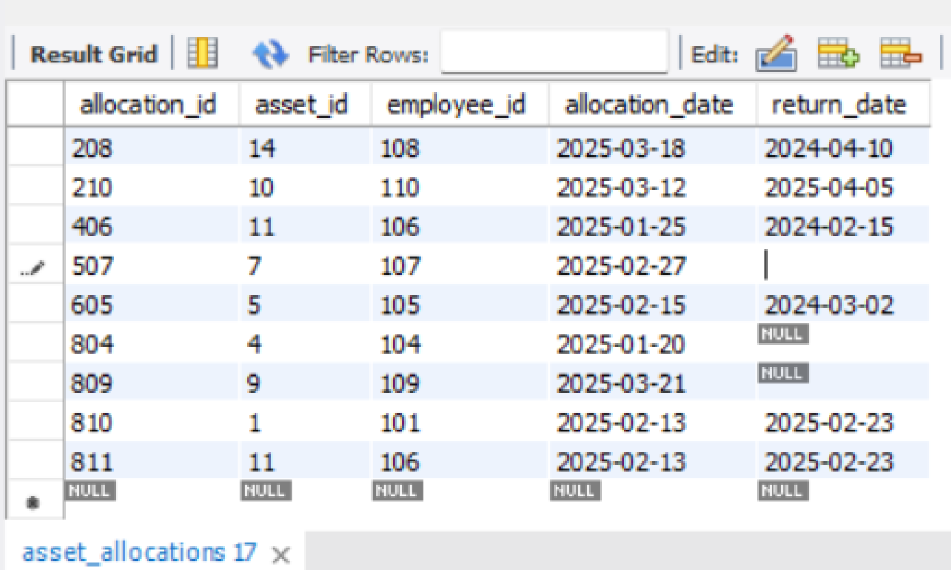


	allocation_id	asset_id	employee_id	allocation_date	return_date
	208	14	108	2025-03-18	2024-04-10
	210	10	110	2025-03-12	2025-04-05
	406	11	106	2025-01-25	2024-02-15
	507	7	107	2025-02-27	NULL
	605	5	105	2025-02-15	2024-03-30
	804	4	104	2025-01-20	NULL
	809	9	109	2025-03-21	NULL
	810	1	101	2025-02-13	2025-02-23
	811	11	106	2025-02-13	2025-02-23
*	NULL	NULL	NULL	NULL	NULL

After Deallocation



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program F
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
104
Driver Loaded Successfully
Great!!, We confirmed that you are an Employee, GO Further
The Available Operations are Listed.
To ADD Asset Enter '1'
To UPDATE Asset Enter '2'
To DELETE Asset Enter '3'
To ALLOCATE Asset Enter '4'
To DEALLOCATE Asset Enter '5'
To PERFORM MAINTAINENCE Enter '6'
To RESERVE Asset Enter '7'
5
Enter Asset ID: 5
Enter Employee ID: 105
Enter Return Date (yyyy-MM-dd): 2024-03-02
Asset Deallocated Successfully

Process finished with exit code 0
```



allocation_id	asset_id	employee_id	allocation_date	return_date
208	14	108	2025-03-18	2024-04-10
210	10	110	2025-03-12	2025-04-05
406	11	106	2025-01-25	2024-02-15
507	7	107	2025-02-27	
605	5	105	2025-02-15	2024-03-02
804	4	104	2025-01-20	NULL
809	9	109	2025-03-21	NULL
810	1	101	2025-02-13	2025-02-23
811	11	106	2025-02-13	2025-02-23
NULL	NULL	NULL	NULL	NULL

For asset id 5 and employee id 105 the return date changed to 2024-03-02.the deallocation done before 28 days.

3.6 Perform Maintainence

Implementation

```
//To perform maintenance
public boolean performMaintenance(int asset_id,String
maintainence_date,String description,double cost) throws
SQLException {
   boolean performed=false;
   int
maintainence_id=getNewMaintainenceId("maintenance_records","ma
intainence_id");
   DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("yyyy-MM-dd");
   LocalDate date = LocalDate.parse(maintainence_date, formatter);
   PreparedStatement pst= connection.prepareStatement("INSERT
INTO maintenance_records (maintenance_id, asset_id,
maintenance_date, description, cost) VALUES (?, ?, ?, ?, ?)");
   pst.setInt(1,maintainence_id);
   pst.setInt(2,asset_id);
   pst.setDate(3, Date.valueOf(maintainence_date));
   pst.setString(4,description);
   pst.setDouble(5,cost);
   int rows= pst.executeUpdate();
   if(rows>0)
   {
      performed=true;
   }
   return performed;
}
```

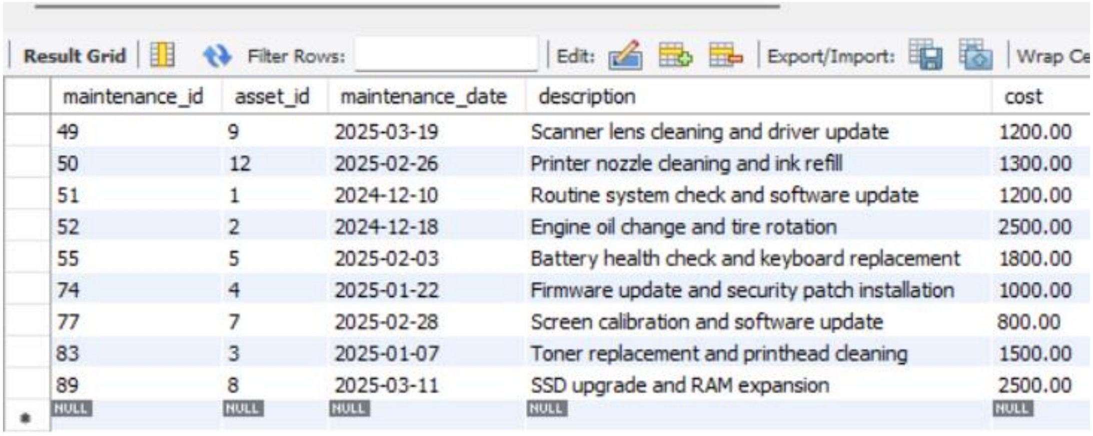
Main

```
else if (option == 6)
{
   System.out.print("Enter Asset ID: ");
   int assetId = sc.nextInt();
```

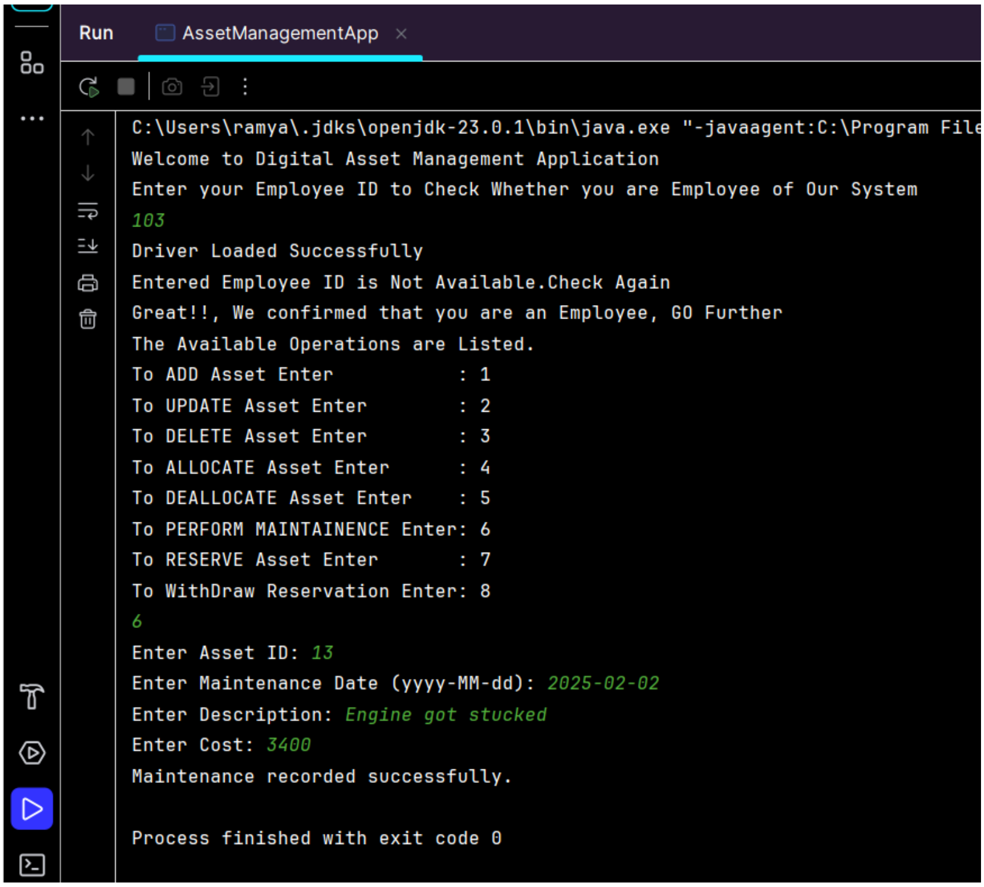
```
    sc.nextLine();
    //handles exception for not found asset
    try {
       service.checkAssetID(assetId);
    } catch (AssetNotFoundException e) {
       System.out.println(e.getMessage()); // Prints custom message
    }
    //handles exception for not sound employee
    try{
       service.checkmaintainenceDate(assetId);
    }catch (AssetNotFoundException e)
    {
       System.out.println(e.getMessage());
    }
    System.out.print("Enter Maintenance Date (yyyy-MM-dd): ");
    String maintenanceDate = sc.nextLine();
    System.out.print("Enter Description: ");
    String description = sc.nextLine();
    System.out.print("Enter Cost: ");
    double cost = sc.nextDouble();
    boolean maintained = service.performMaintenance(assetId,
maintenanceDate, description, cost);
    if (maintained) {
       System.out.println("Maintenance recorded successfully.");
    } else {
       System.out.println("Failed to record maintenance.");
    }
}
```

OUTPUT

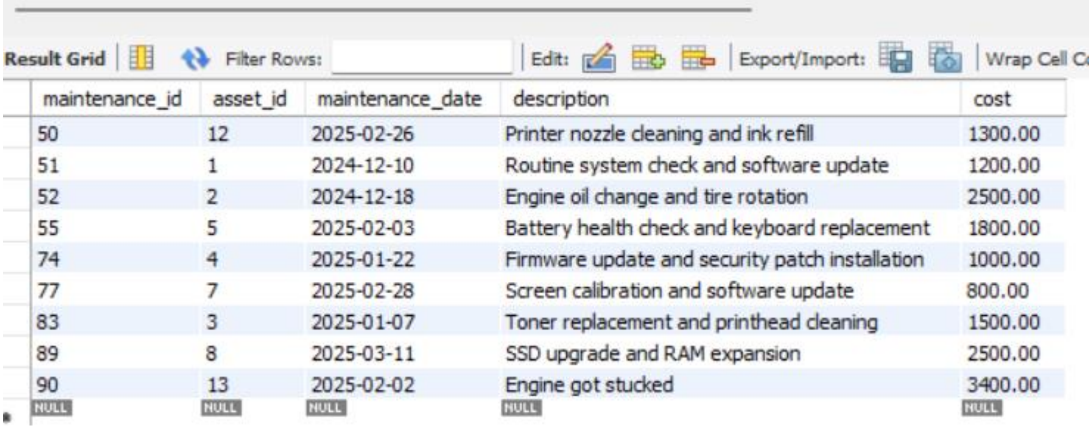
New Maintenance record is added with asset id 13



maintenance_id	asset_id	maintenance_date	description	cost
49	9	2025-03-19	Scanner lens cleaning and driver update	1200.00
50	12	2025-02-26	Printer nozzle cleaning and ink refill	1300.00
51	1	2024-12-10	Routine system check and software update	1200.00
52	2	2024-12-18	Engine oil change and tire rotation	2500.00
55	5	2025-02-03	Battery health check and keyboard replacement	1800.00
74	4	2025-01-22	Firmware update and security patch installation	1000.00
77	7	2025-02-28	Screen calibration and software update	800.00
83	3	2025-01-07	Toner replacement and printhead cleaning	1500.00
89	8	2025-03-11	SSD upgrade and RAM expansion	2500.00
NULL	NULL	NULL	NULL	NULL



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program File
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
103
Driver Loaded Successfully
Entered Employee ID is Not Available.Check Again
Great!!, We confirmed that you are an Employee, GO Further
The Available Operations are Listed.
To ADD Asset Enter           : 1
To UPDATE Asset Enter        : 2
To DELETE Asset Enter        : 3
To ALLOCATE Asset Enter      : 4
To DEALLOCATE Asset Enter    : 5
To PERFORM MAINTAINENCE Enter: 6
To RESERVE Asset Enter       : 7
To WithDraw Reservation Enter: 8
6
Enter Asset ID: 13
Enter Maintenance Date (yyyy-MM-dd): 2025-02-02
Enter Description: Engine got stucked
Enter Cost: 3400
Maintenance recorded successfully.

Process finished with exit code 0
```



maintenance_id	asset_id	maintenance_date	description	cost
50	12	2025-02-26	Printer nozzle cleaning and ink refill	1300.00
51	1	2024-12-10	Routine system check and software update	1200.00
52	2	2024-12-18	Engine oil change and tire rotation	2500.00
55	5	2025-02-03	Battery health check and keyboard replacement	1800.00
74	4	2025-01-22	Firmware update and security patch installation	1000.00
77	7	2025-02-28	Screen calibration and software update	800.00
83	3	2025-01-07	Toner replacement and printhead cleaning	1500.00
89	8	2025-03-11	SSD upgrade and RAM expansion	2500.00
90	13	2025-02-02	Engine got stucked	3400.00
NULL	NULL	NULL	NULL	NULL

New Maintenance record is added with maintenanceid 90

3.7 Reserve Asset

Implementation

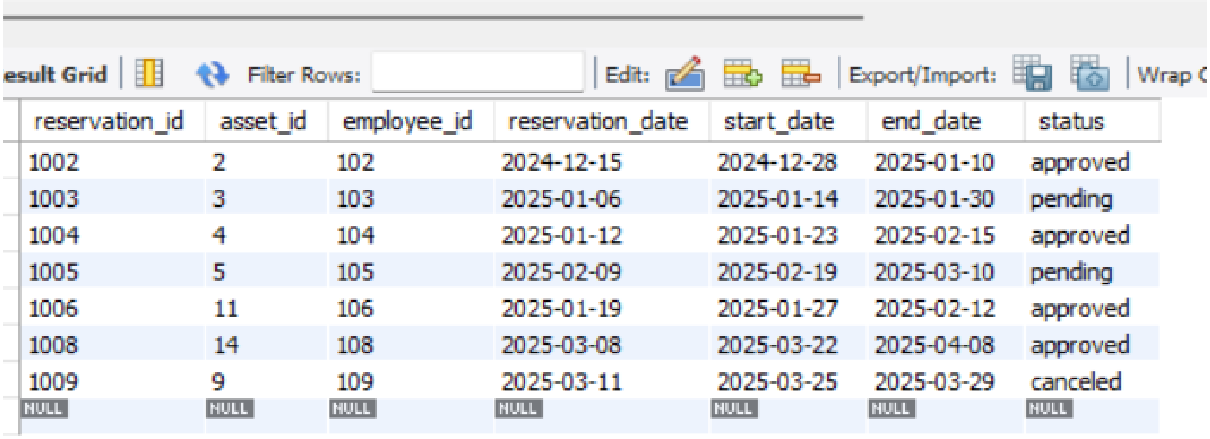
```
//To reserve Asset
@Override
public boolean reserveAsset(int assetId, int employeeId, String
reservationDate, String startDate, String endDate) throws
SQLException {
    boolean reserved = false;

    // Generate new reservation ID
    int reservationId = getNewReservationId("reservations",
"reservation_id");

    DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("yyyy-MM-dd");
    LocalDate resDate = LocalDate.parse(reservationDate, formatter);
    LocalDate sDate = LocalDate.parse(startDate, formatter);
    LocalDate eDate = LocalDate.parse(endDate, formatter);

    PreparedStatement pst = connection.prepareStatement( "INSERT
INTO reservations (reservation_id, asset_id, employee_id,
reservation_date, start_date, end_date, status) " +
            "VALUES (?, ?, ?, ?, ?, ?, ?)");
    pst.setInt(1, reservationId);
    pst.setInt(2, assetId);
    pst.setInt(3, employeeId);
    pst.setDate(4, Date.valueOf(resDate));
    pst.setDate(5, Date.valueOf(sDate));
    pst.setDate(6, Date.valueOf(eDate));
    pst.setString(7, "Reserved");
    int rows = pst.executeUpdate();
    if (rows > 0) {
        reserved = true;
    }
```

```
   return reserved;
}
```

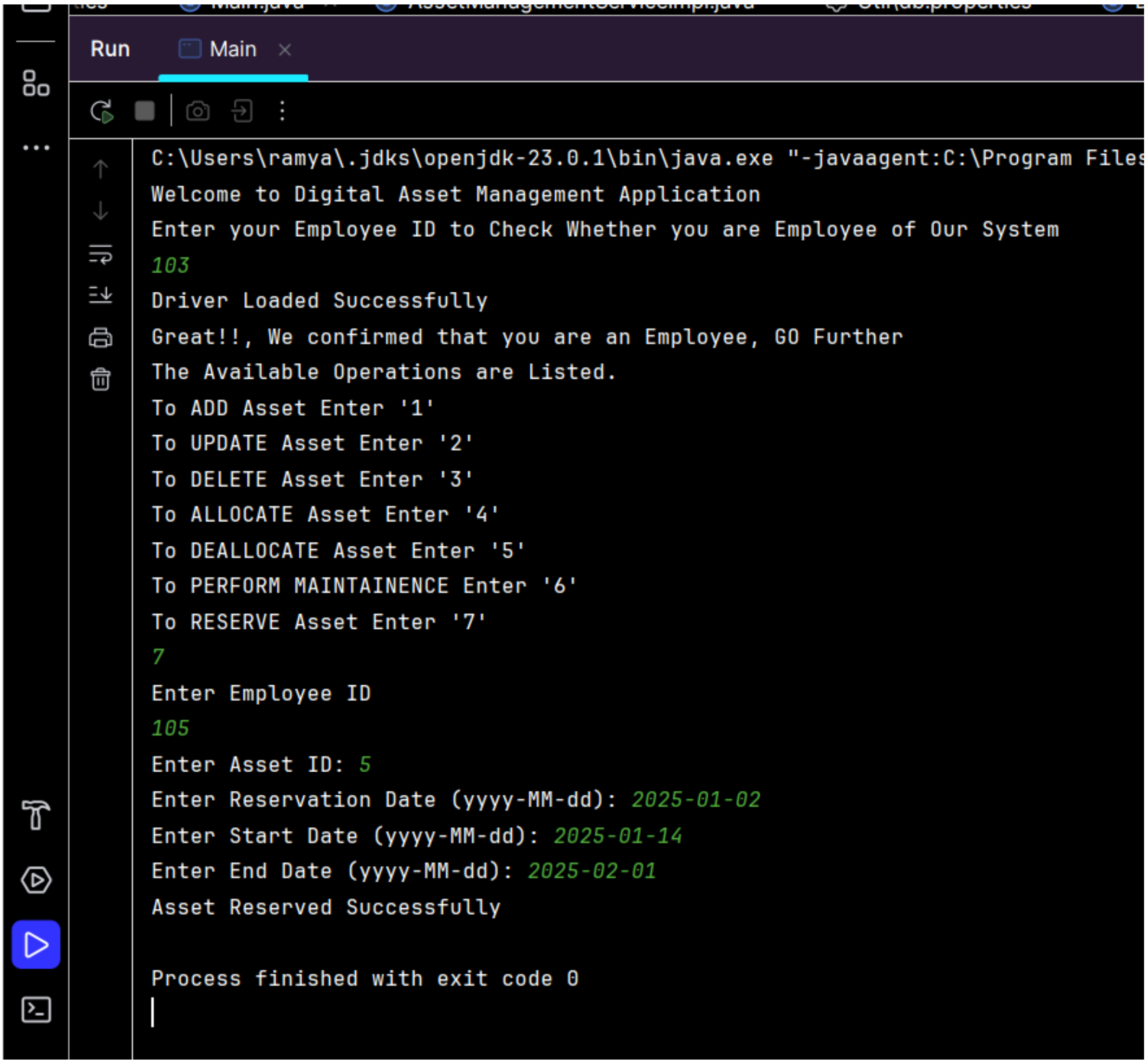
Main

```
else if (option==7)
{
   System.out.println("Enter Employee ID");
   int eid=sc.nextInt();
   try{
      service.checkEmployeeID(eid);
   }catch (EmployeeNotFoundException e) {
      System.out.println(e.getMessage());
   }

      System.out.print("Enter Asset ID: ");
      int assetId = sc.nextInt();
      System.out.print("Enter Reservation Date (yyyy-MM-dd): ");
      String reserDate = sc.next();
      System.out.print("Enter Start Date (yyyy-MM-dd): ");
      String startDate = sc.next();
      System.out.print("Enter End Date (yyyy-MM-dd): ");
      String endDate = sc.next();
      boolean reserved=
service.reserveAsset(assetId,eid,reserDate,startDate,endDate);
      if(reserved)
      {
         System.out.println("Asset Reserved Successfully");
      }
      else {
         System.out.println("Problem in Asset Reservation.Try Later");
      }

}
```

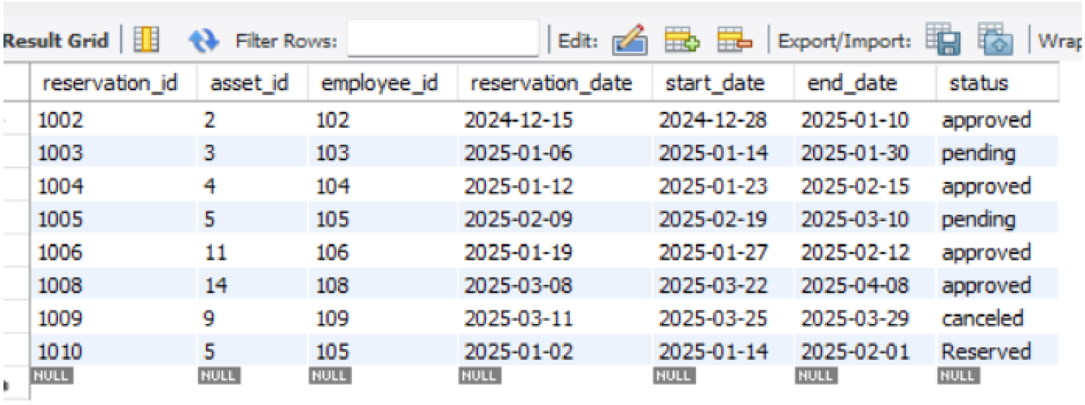
OUTPUT



reservation_id	asset_id	employee_id	reservation_date	start_date	end_date	status
1002	2	102	2024-12-15	2024-12-28	2025-01-10	approved
1003	3	103	2025-01-06	2025-01-14	2025-01-30	pending
1004	4	104	2025-01-12	2025-01-23	2025-02-15	approved
1005	5	105	2025-02-09	2025-02-19	2025-03-10	pending
1006	11	106	2025-01-19	2025-01-27	2025-02-12	approved
1008	14	108	2025-03-08	2025-03-22	2025-04-08	approved
1009	9	109	2025-03-11	2025-03-25	2025-03-29	canceled
NULL	NULL	NULL	NULL	NULL	NULL	NULL



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
103
Driver Loaded Successfully
Great!!, We confirmed that you are an Employee, GO Further
The Available Operations are Listed.
To ADD Asset Enter '1'
To UPDATE Asset Enter '2'
To DELETE Asset Enter '3'
To ALLOCATE Asset Enter '4'
To DEALLOCATE Asset Enter '5'
To PERFORM MAINTAINENCE Enter '6'
To RESERVE Asset Enter '7'
7
Enter Employee ID
105
Enter Asset ID: 5
Enter Reservation Date (yyyy-MM-dd): 2025-01-02
Enter Start Date (yyyy-MM-dd): 2025-01-14
Enter End Date (yyyy-MM-dd): 2025-02-01
Asset Reserved Successfully

Process finished with exit code 0
```



reservation_id	asset_id	employee_id	reservation_date	start_date	end_date	status
1002	2	102	2024-12-15	2024-12-28	2025-01-10	approved
1003	3	103	2025-01-06	2025-01-14	2025-01-30	pending
1004	4	104	2025-01-12	2025-01-23	2025-02-15	approved
1005	5	105	2025-02-09	2025-02-19	2025-03-10	pending
1006	11	106	2025-01-19	2025-01-27	2025-02-12	approved
1008	14	108	2025-03-08	2025-03-22	2025-04-08	approved
1009	9	109	2025-03-11	2025-03-25	2025-03-29	canceled
1010	5	105	2025-01-02	2025-01-14	2025-02-01	Reserved
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Asset with id 5 and EmployeeID 105 is Reserved

3.8 Withdraw Reservation

Implementation

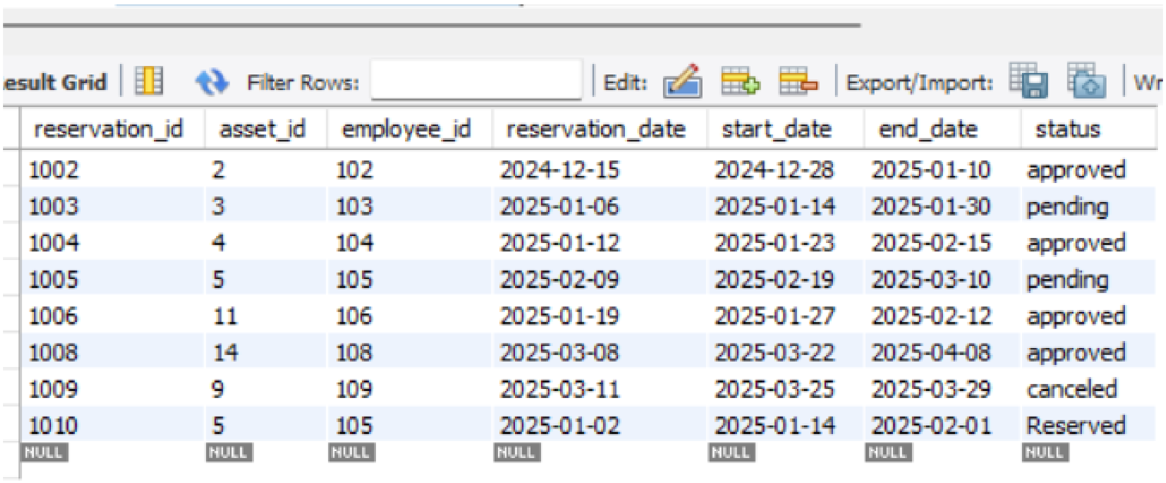
```
//To withdraw reserved asset
@Override
public boolean withdrawReservation(int reservationId) throws
SQLException {
   boolean withdrawn=false;
   PreparedStatement pst = connection.prepareStatement("UPDATE
reservations SET status = 'Cancelled' WHERE reservation_id = ?");
   pst.setInt(1, reservationId);

   int rows = pst.executeUpdate();
   if(rows>0)
   {
      withdrawn=true;
   }
   return withdrawn;
}
```

Main

```
else if (option==8)
{
   System.out.println("To Withdraw Reserved Asset Enter correct
Reservation ID:");
```

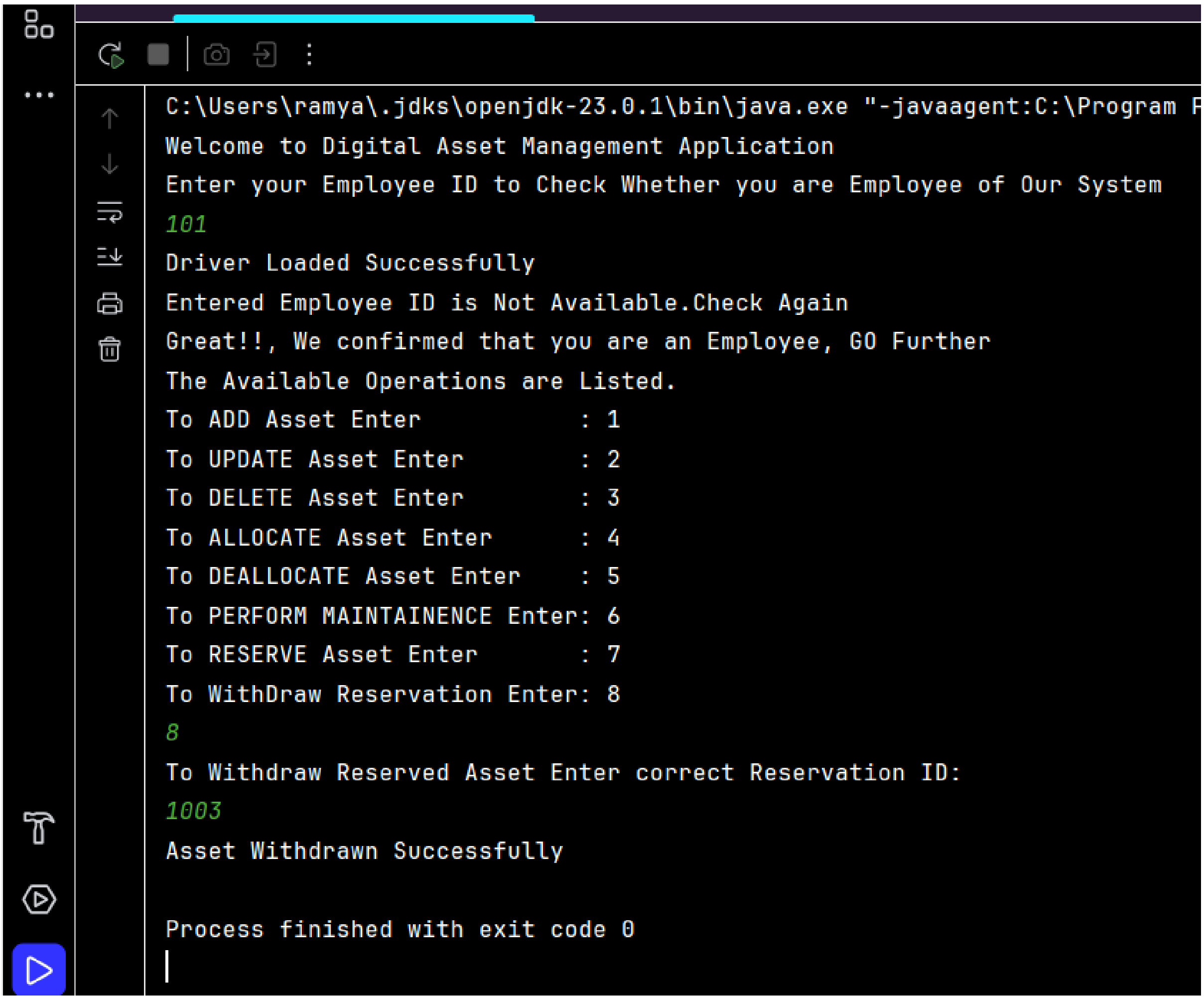
```
    int rid=sc.nextInt();
    boolean ridexist=service.checkReservationID(rid);
    if(ridexist) {
        boolean withdrawn = service.withdrawReservation(rid);
        if (withdrawn) {
            System.out.println("Asset Withdrawn Successfully");
        } else {
            System.out.println("Server Issue,Try Later");
        }
    }
    else
    {
        System.out.println("Reservation ID Not Found");
    }
}
```

OUTPUT

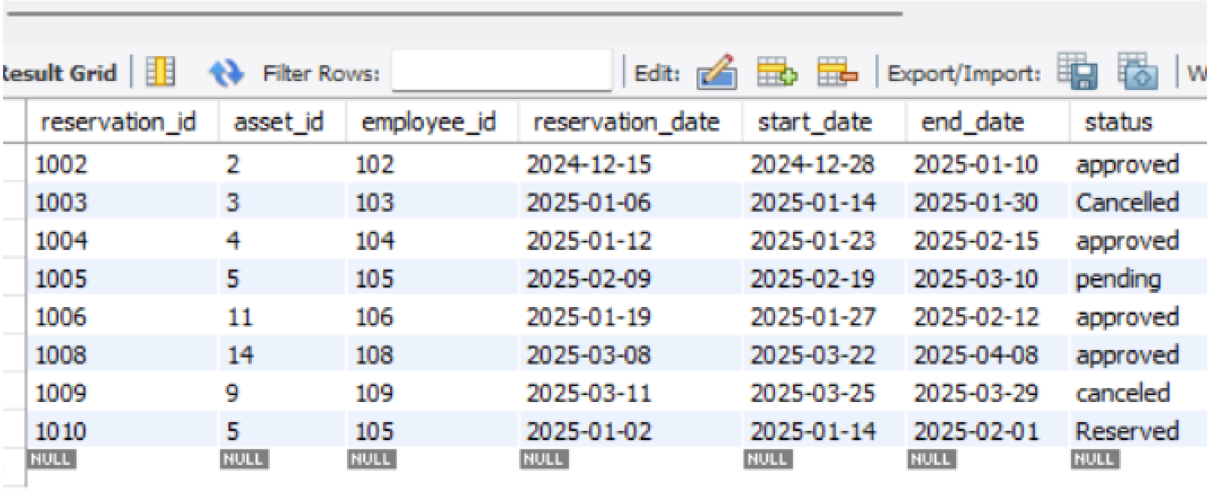


reservation_id	asset_id	employee_id	reservation_date	start_date	end_date	status
1002	2	102	2024-12-15	2024-12-28	2025-01-10	approved
1003	3	103	2025-01-06	2025-01-14	2025-01-30	pending
1004	4	104	2025-01-12	2025-01-23	2025-02-15	approved
1005	5	105	2025-02-09	2025-02-19	2025-03-10	pending
1006	11	106	2025-01-19	2025-01-27	2025-02-12	approved
1008	14	108	2025-03-08	2025-03-22	2025-04-08	approved
1009	9	109	2025-03-11	2025-03-25	2025-03-29	canceled
1010	5	105	2025-01-02	2025-01-14	2025-02-01	Reserved
NULL	NULL	NULL	NULL	NULL	NULL	NULL

To withdraw Reservation where reservationid is 1003 after withdrawn status pending changed to cancelled



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program F
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
101
Driver Loaded Successfully
Entered Employee ID is Not Available.Check Again
Great!!, We confirmed that you are an Employee, GO Further
The Available Operations are Listed.
To ADD Asset Enter          : 1
To UPDATE Asset Enter       : 2
To DELETE Asset Enter       : 3
To ALLOCATE Asset Enter     : 4
To DEALLOCATE Asset Enter   : 5
To PERFORM MAINTAINENCE Enter: 6
To RESERVE Asset Enter      : 7
To WithDraw Reservation Enter: 8
8
To Withdraw Reserved Asset Enter correct Reservation ID:
1003
Asset Withdrawn Successfully

Process finished with exit code 0
```



reservation_id	asset_id	employee_id	reservation_date	start_date	end_date	status
1002	2	102	2024-12-15	2024-12-28	2025-01-10	approved
1003	3	103	2025-01-06	2025-01-14	2025-01-30	Cancelled
1004	4	104	2025-01-12	2025-01-23	2025-02-15	approved
1005	5	105	2025-02-09	2025-02-19	2025-03-10	pending
1006	11	106	2025-01-19	2025-01-27	2025-02-12	approved
1008	14	108	2025-03-08	2025-03-22	2025-04-08	approved
1009	9	109	2025-03-11	2025-03-25	2025-03-29	canceled
1010	5	105	2025-01-02	2025-01-14	2025-02-01	Reserved
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Status ‘pending’ changed to ‘cancelled’

And again status changed to ‘available’ for next allocation

4. Write code to establish a connection to your SQL database.

- Create a utility class **DBConnection** in a package util with a static variable connection of Type Connection and a static method **getConnection()** which returns connection.

- Connection **properties** supplied in the connection string should be read from a property file.

DBConnection

```
package Util;

import java.io.FileInputStream;
import java.io.IOException;
import java.sql.*;
import java.util.Properties;

public class DBConnection {
   public static Connection getConnection() throws IOException,
ClassNotFoundException, SQLException {


        FileInputStream fis = new
FileInputStream("C:\\Users\\ramya\\IdeaProjects\\JDBC\\src\\Util\\db
.properties");
        Properties properties = new Properties();
        properties.load(fis);

        String driver = properties.getProperty("db.driver");
        String url = properties.getProperty("db.url");
        String username = properties.getProperty("db.username");
        String password = properties.getProperty("db.password");


        Class.forName(driver);
        Connection connection = DriverManager.getConnection(url,
username, password);
```

```
    return connection;
   }
}
```

db.properties File:

```
db.url=jdbc:mysql://localhost:3306/asset_management
db.username=root
db.password=******
db.driver=com.mysql.cj.jdbc.Driver
```

5.Create the exceptions in package myexceptions and create the following custom exceptions and throw them in methods whenever needed. Handle all the exceptions in main method,

- AssetNotFoundException: throw this exception when employee enters an invalid asset id which doesn’t exist in db

- AssetNotMaintainException: throw this exception when employee need the asset which is not maintained for 2 years.

1.AssetNotFoundException:

Exception Class

```
package Myexceptions;

public class AssetNotFoundException extends RuntimeException {
   public AssetNotFoundException(String message) {
      super(message);
   }
}
```

Throws exception in Method

```
//To check existence of assetid to proceed further
public void checkAssetID(int assetId) throws SQLException
,AssetNotFoundException{

      boolean check=false;
   PreparedStatement pst = connection.prepareStatement( "SELECT
COUNT(*) FROM assets WHERE asset_id = ?");
   pst.setInt(1, assetId);
   ResultSet rs = pst.executeQuery();
   if (rs.next() && rs.getInt(1) == 0) //Gets the value in the first
column of the current row.
   {
      throw new AssetNotFoundException("Asset with ID " + assetId
+ " does not exist.Kindly Check Again and Proceed");
   }
```

Handling using try-catch block

```
//Handles AssetNotFoundException
try {
   service.checkAssetID(aid);
} catch (AssetNotFoundException e) {
   System.out.println(e.getMessage()); // Prints custom message
}
```

2.AssetNotMaintainException:

Exception Class

```
package Myexceptions;

public class AssetNotMaintainException extends RuntimeException {
   public AssetNotMaintainException(String message) {
      super(message);
   }
}
```

Throws exception in Method

```
@Override
public void checkmaintainenceDate(int assetid) throws SQLException
, AssetNotMaintainException{
   PreparedStatement pst=connection.prepareStatement("select
max(maintenance_date) from maintenance_records where
asset_id=?");
```

```
   pst.setInt(1,assetid);
   ResultSet rs= pst.executeQuery();
   if(rs.next())
   {
      Date maxdate=rs.getDate(1);
      long millisec = 2 * 365L * 24 * 60 * 60 * 1000;
      long diff = System.currentTimeMillis() - maxdate.getTime();

      if (diff >= millisec) {
         throw new AssetNotMaintainException("Asset not Maintained
in the last 2 years.So Performing Maintainence is Not Possible");
      }
   }
```

Handling using try-catch block

```
//handles exception for maintainence date
try{
   service.checkmaintainenceDate(assetId);
}catch (AssetNotFoundException e)
{
   System.out.println(e.getMessage());
}
```

3.EmployeeNotFoundException

Exception Class

```
package Myexceptions;

public class EmployeeNotFoundException extends RuntimeException
{
   public EmployeeNotFoundException(String message) {
      super(message);
   }
}
```

Throws exception in Method

```
/To check existence of employeeid to proceed further
    @Override
    public void checkEmployeeID(int empid) throws SQLException {
        boolean exist=false;
        PreparedStatement pst=connection.prepareStatement("select 1
from employees where employee_id=?");
        pst.setInt(1,empid);
        ResultSet rs = pst.executeQuery();
        if (rs.next()) {
            throw  new EmployeeNotFoundException("Entered Employee
ID is Not Available.Check Again");
        }

    }
```

Handling using try-catch block

```
//handles exception for not found employee id
try{
    service.checkEmployeeID(empid);
}catch (EmployeeNotFoundException e) {
    System.out.println(e.getMessage());
}
```

4. ReservationNotFoundException

Exception Class

```
package Myexceptions;

public class ReservationNotFoundException extends
RuntimeException {
    public ReservationNotFoundException(String message)
    {
        super(message);  }
}
```

Throws exception in Method

```
//To check existence of reservationid to proceed further
@Override
public void checkReservationID(int reservation_id) throws
SQLException {

   PreparedStatement pst=connection.prepareStatement("select
count(*) from reservations where reservation_id=?");
   pst.setInt(1,reservation_id);
   ResultSet rs = pst.executeQuery();
   if (rs.next())
   {
      throw new ReservationNotFoundException("Entered
Reservation ID does not exist.Kindly Check Again and Proceed");
   }
}
```

Handling using try-catch block

```
// Check if the Reservation ID exists
try {
  checkReservationID(reservationId);
}catch (ReservationNotFoundException e) {
   System.out.println(e.getMessage()); // Prints custom message
}
```

6. Create class named **AssetManagementApp** with main method in app Trigger all the methods in service implementation class by user choose operation from the following menu.

Main class provides the menu functionality for the user to choose specific operation. And **conditional statements** are used to navigate the choosen option

Main class

As a first step in main method,the program checks for the existence of employee Id given by the user.

As this digital asset management is only accessed by the employee,this feature is added.

CODE;

```
public class AssetManagementApp {

   public static void main(String[] args) throws SQLException,
ParseException {

       Scanner sc = new Scanner(System.in);
       System.out.println("Welcome to Digital Asset Management
Application");
       System.out.println("Enter your Employee ID to Check Whether
you are Employee of Our System");
       int empid = sc.nextInt();
       IAssetManagementService service = new
AssetManagementServiceImpl();
       //handles exception for not found employee id
       try {
          service.checkEmployeeID(empid);

          System.out.println("Great!!, We confirmed that you are an
Employee, GO Further");
          System.out.println("The Available Operations are Listed.");
          System.out.println("To ADD Asset Enter          : 1\n" +
                "To UPDATE Asset Enter       : 2\n" +
                "To DELETE Asset Enter       : 3\n" +
                "To ALLOCATE Asset Enter     : 4\n" +
                "To DEALLOCATE Asset Enter   : 5\n" +
                "To PERFORM MAINTAINENCE Enter: 6\n" +
                "To RESERVE Asset Enter      : 7\n" +
                "To WITHDRAW Reservation Enter: 8");

          int option = sc.nextInt();
```

After getting the option,it passed to conditional statements.

7.Create **Unit test cases** for Digital Asset Management System are essential to ensure the correctness and reliability of your system. Following questions to guide the creation of Unit test cases:

Here 5 main testcases are created with in **TestCaseDevelopment** Test class

```
package AssetManagementApp;
import Entities.Assets;
import DAO.AssetManagementServiceImpl;
import Myexceptions.AssetNotFoundException;
import Myexceptions.EmployeeNotFoundException;
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import java.sql.SQLException;
import java.text.ParseException;

public class TestCaseDevelopment {

    static AssetManagementServiceImpl ams;
    //create object before all tests
    @BeforeAll
    static void setup()
    {
        ams= new AssetManagementServiceImpl();
    }
    //empty object after all tests
    @AfterAll
    static void destroy()
    {
        ams=null;
    }
```

```
   // test case to test asset created successfully or not.
   @Test
   void TestAssetAdded() throws SQLException, ParseException {

       Assets assets=new Assets(29,"Mac
Book","Electronics","S02985",null,"Velur","Ordered",195);

       boolean test= ams.addAsset(assets);
       Assertions.assertEquals(true,test);
       System.out.println("Test 1 Passed Successfully");
   }
 // test case to test asset is added to maintenance successfully or
not.
   @Test
   void TestAssetMaintained() throws SQLException {
      boolean test=ams.performMaintenance(3,"2025-02-
14","Unexpected Power Off",2460);
      Assertions.assertEquals(true,test);
       System.out.println("Test 2 Passed Successfully");


   }
   // test case to test asset is reserved successfully or not.
   @Test
   void TestAssetReserved() throws SQLException {
       boolean test=ams.reserveAsset(5,108,"2025-01-17","2025-01-
29","2025-02-15");
       Assertions.assertEquals(true,test);
       System.out.println("Test 3 Passed Successfully");
   }
   //test case to test exception is thrown correctly or not asset id
not found in database.
   @Test
   void testAssetNotFoundException() {
       int AssetId = 94;
```

```
        Assertions.assertThrows(AssetNotFoundException.class, () -> {
            ams.checkAssetID(AssetId);
        });
        System.out.println("Test 4 Passed Successfully");
    }
    //test case to test exception is thrown correctly or not asset id
not found in database.
    @Test
    void testEmployeeNotFoundException() {
        int invalidEmpId = 88; // assuming this ID does not exist in DB

        Assertions.assertThrows(EmployeeNotFoundException.class, ()
-> {
            ams.checkEmployeeID(invalidEmpId);
        });
        System.out.println("Test 5 Passed Successfully");
    }
}
```

OUTPUT

- Write test case to test asset created successfully or not

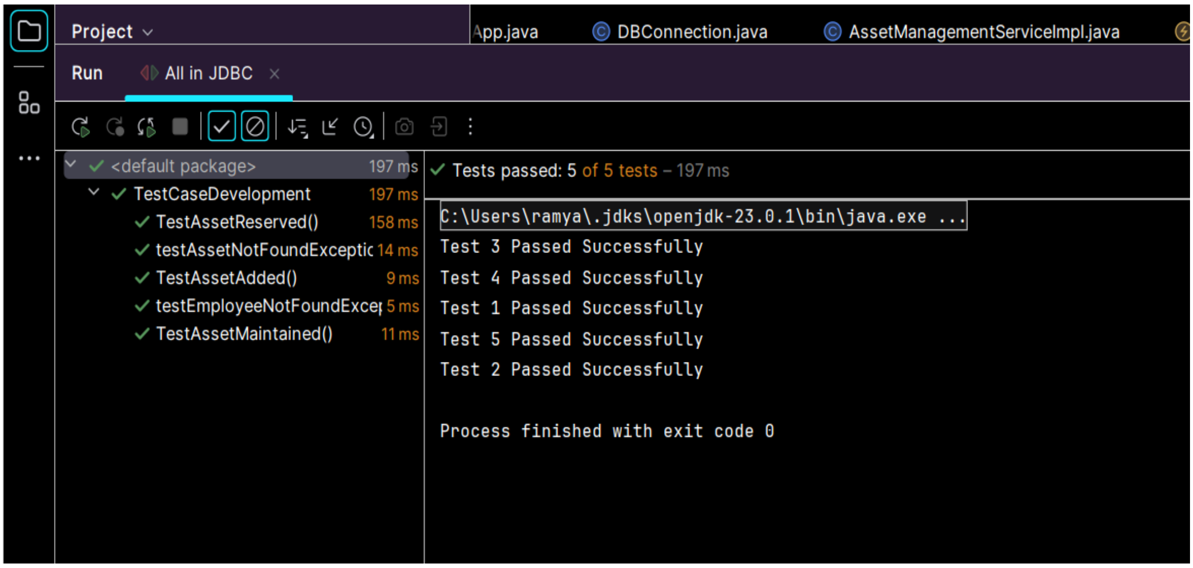
- Write test case to test asset is added to maintenance successfully or not.

- Write test case to test asset is reserved successfully or not.

- write test case to test exception is thrown correctly or not when asset id not found in database.

- write test case to test exception is thrown correctly or not when employee id not found in database.

ALL 5 Passed:



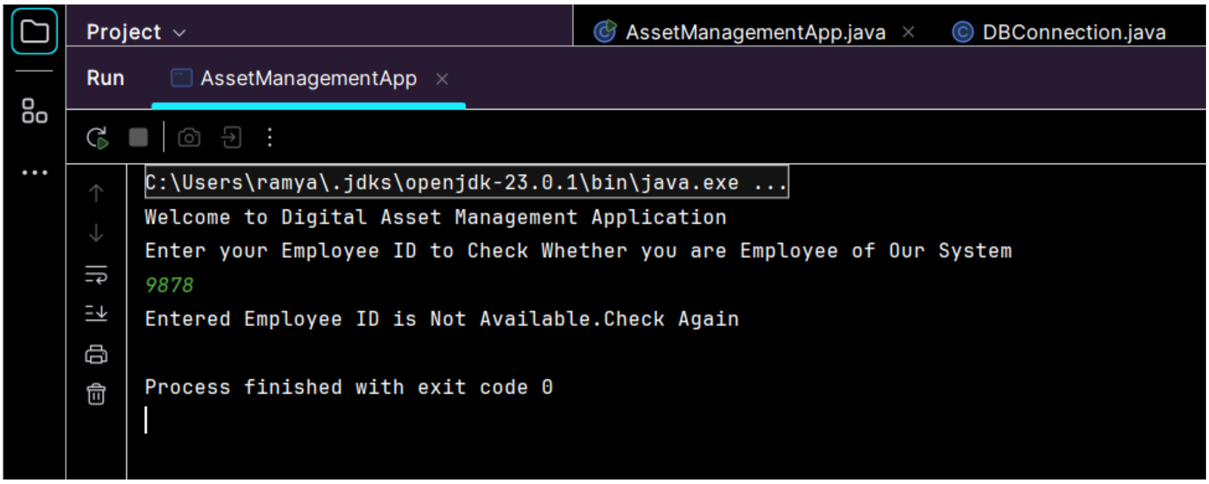
```
<default package>                197 ms
  TestCaseDevelopment            197 ms
    TestAssetReserved()          158 ms
    testAssetNotFoundExceptic    14 ms
    TestAssetAdded()             9 ms
    testEmployeeNotFoundExcep    5 ms
    TestAssetMaintained()        11 ms
```

```
Tests passed: 5 of 5 tests – 197 ms
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe ...
Test 3 Passed Successfully
Test 4 Passed Successfully
Test 1 Passed Successfully
Test 5 Passed Successfully
Test 2 Passed Successfully

Process finished with exit code 0
```

8.A perfect incremental Digital Asset Management Application is Succcessfully developed and sample output is attached.

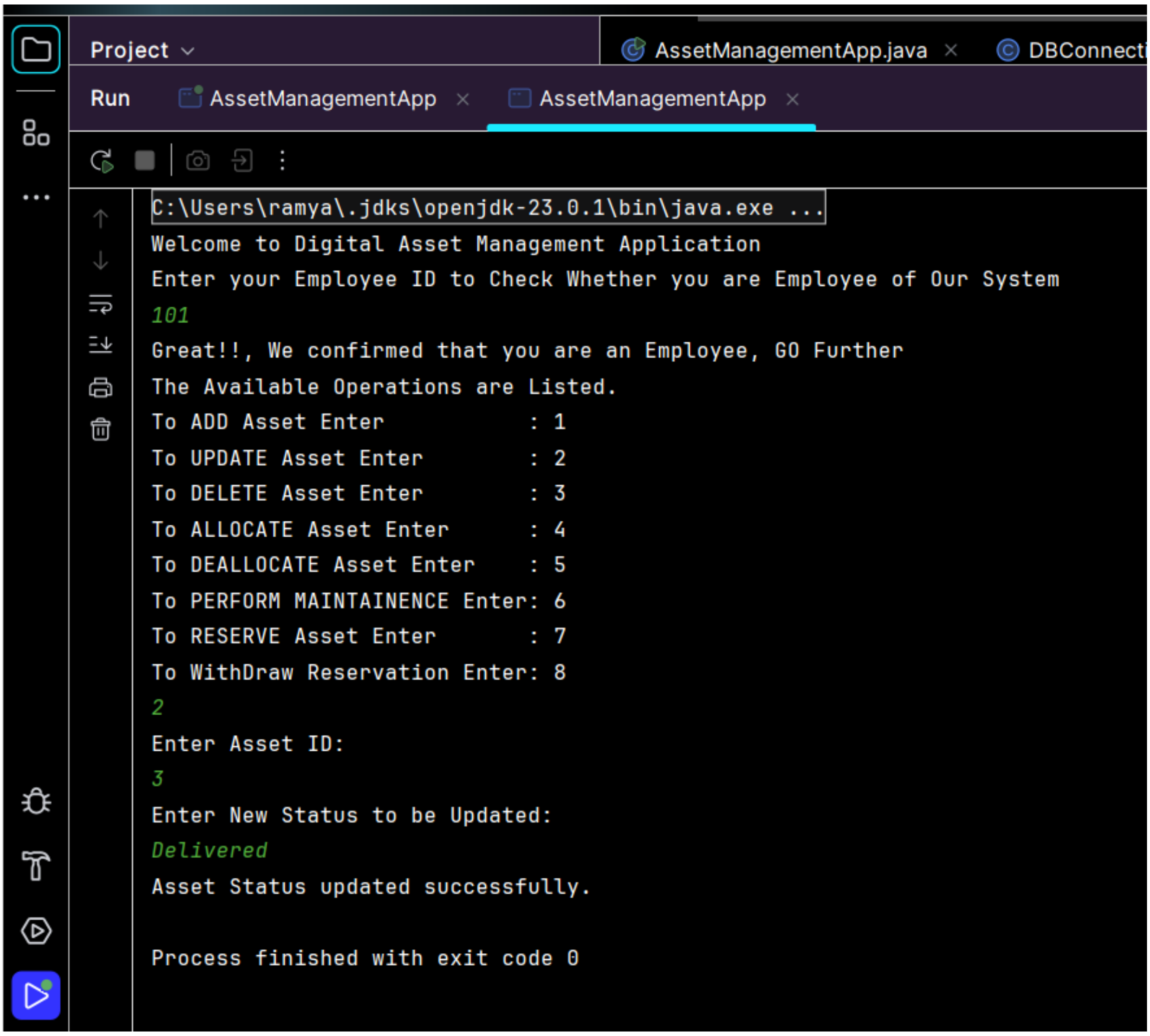
FINAL OUTPUT:

1.Initial check for Emp id(Fails)



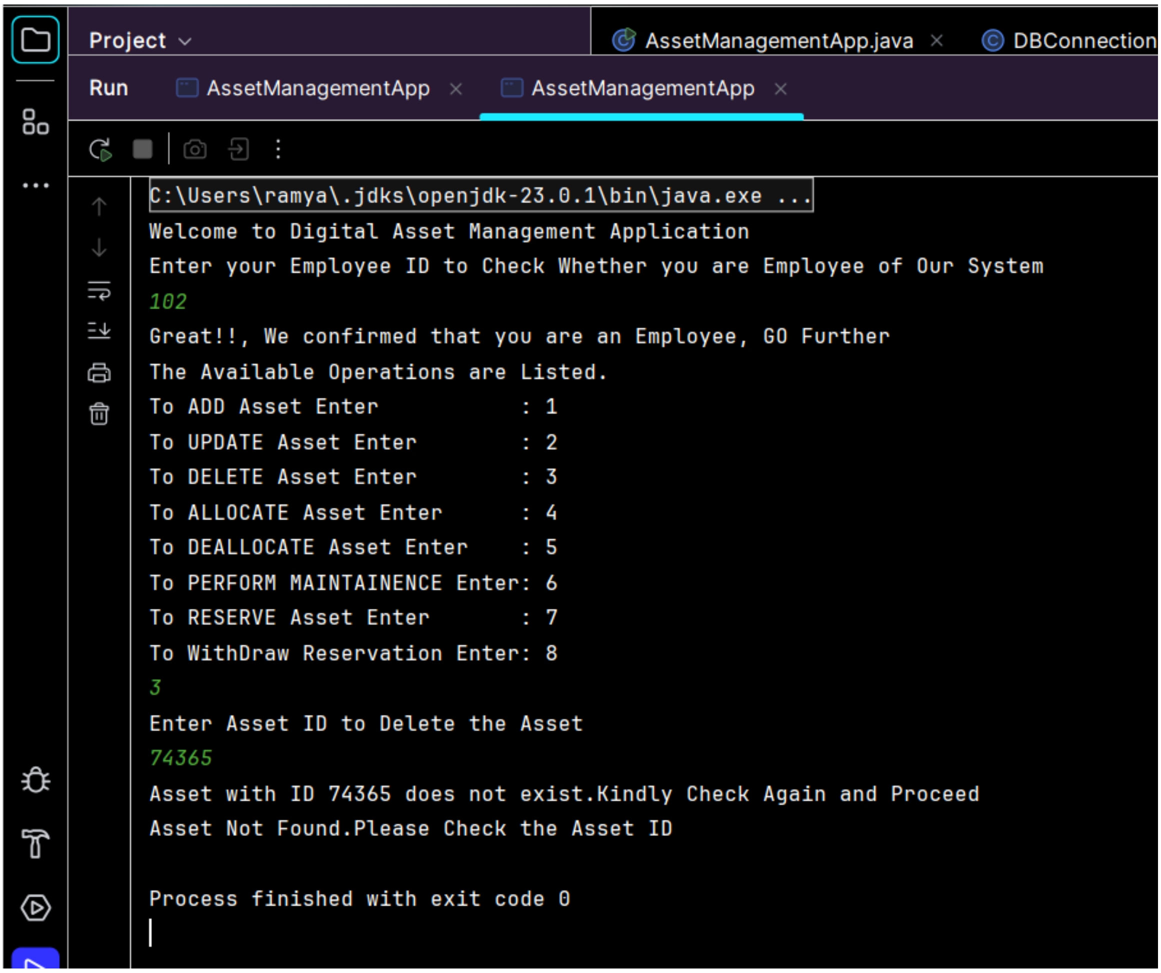
```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe ...
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
9878
Entered Employee ID is Not Available.Check Again

Process finished with exit code 0
```

2.Successful check for Emp id



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe ...
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
101
Great!!, We confirmed that you are an Employee, GO Further
The Available Operations are Listed.
To ADD Asset Enter           : 1
To UPDATE Asset Enter        : 2
To DELETE Asset Enter        : 3
To ALLOCATE Asset Enter      : 4
To DEALLOCATE Asset Enter    : 5
To PERFORM MAINTAINENCE Enter: 6
To RESERVE Asset Enter       : 7
To WithDraw Reservation Enter: 8
2
Enter Asset ID:
3
Enter New Status to be Updated:
Delivered
Asset Status updated successfully.

Process finished with exit code 0
```

3.Exception throwing for Incorrect Asset Id



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe ...
Welcome to Digital Asset Management Application
Enter your Employee ID to Check Whether you are Employee of Our System
102
Great!!, We confirmed that you are an Employee, GO Further
The Available Operations are Listed.
To ADD Asset Enter           : 1
To UPDATE Asset Enter        : 2
To DELETE Asset Enter        : 3
To ALLOCATE Asset Enter      : 4
To DEALLOCATE Asset Enter    : 5
To PERFORM MAINTAINENCE Enter: 6
To RESERVE Asset Enter       : 7
To WithDraw Reservation Enter: 8
3
Enter Asset ID to Delete the Asset
74365
Asset with ID 74365 does not exist.Kindly Check Again and Proceed
Asset Not Found.Please Check the Asset ID

Process finished with exit code 0
```