EP Practical – 3 **190031920**

Nikhil Reddy Avuthu

Pre-Lab Task:

1)

a. List out different kinds of JDBC drivers.

Ans:

There are 4 types of JDBC drivers:

1. JDBC-ODBC bridge
2. Partial Java Driver
3. Network Protocol Driver
4. Thin driver

b. Write the steps involved in connecting any java application with data base using JDBC.

Steps are:

1.Import packages

2.Loading driver class

3.Establishing a connection

4.Creating a statement

5.Specifying query

6.Executing the Query

7.Process the ResultSet

8.Close the connection

9.Handle exception

c. Write a small description and syntax for the following methods:

i. forName()

forName method return the class object associated with the class or interface with the given string name, using given class loader.

ii. registerDriver()

This method of DriverManager class registers the given driver in the DriverManager's list. If the driver is null, it returns the NullPointerException

iii. getConnection()
This method establishes a connection to database by using the given database url. The appropriate driver from the set of registered JDBC drivers is selected.

iv. createStatement()

This method creates Statement instance for sending SQL statement to database. It is an interface under java.

v. prepareStatement()

It is subinterface of statement. It is used to execute parameterized query.

vi. executeQuery()

This method is used to execute statements that returns tabular data. It also returns the object of ResultSet class.

vii. executeUpdate()

This method is used to execute SQL statement for which we expect to affect the rows of table.

viii. close()

It is a method of reader class which is used to close the stream and release resources which were busy in a stream.

ix. setBinaryStream()

This method is used to set the contents of the given InputStream as a value for the parameter in specified index.

x. setCharacterStream()

It is specified in java, sql by setCharacterStream method.If length of stream is different than the specified one then an exception is thrown.

xi. available()

The method is built-in method of java.io. It tells total number of bytes from the input stream to be read.

2. Write a java program which executes the given SQL queries.

(NOTE: create a table ‘student’ with attributes to store the student’s id(number),name(varchar2),age(number) manually in database)

Add the following records into the ‘student’ table.

Id	name	age
1	Felix	20
2	jack	19
3	mark	21

Print the details of the students whose age is less than 21.

Code:

```
import java.sql.*;

public class App {
   public static void main(String[] args) throws Exception {
      try {
         // step1 load the driver class
         Class.forName("oracle.jdbc.driver.OracleDriver");

         String dbUserName = "system";// your user name goes here

         String dbUserPassword = "nikhil";// your password goes here

         // step2 create the connection object
         Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", dbUse
rName,
               dbUserPassword);

         // step3 create the statement object
         Statement stmt = con.createStatement();

         // Delete any previous values in the table
         stmt.executeUpdate("TRUNCATE TABLE student");

         // Prepare to insert new names in the EMP table
         String query1 = "INSERT INTO student (id, name, age)" + "VALUES ('1', 'Felix', '20')";

         String query2 = "INSERT INTO student (id, name, age)" + "VALUES ('2', 'Jack', '19')";

         String query3 = "INSERT INTO student (id, name, age)" + "VALUES ('3', 'mark', '21')";

         int count = stmt.executeUpdate(query1);
         System.out.println("Number of rows updated in database =  " + count);

         // Executing next SQL INSERT query using executeUpdate() method of Statement
         // object.

         count = stmt.executeUpdate(query2);
         System.out.println("Number of rows updated in database = " + count);
```

```
        // Executing next SQL INSERT query using executeUpdate() method of Statement
        // object.
        count = stmt.executeUpdate(query3);
        System.out.println("Number of rows updated in database = " + count);

        // step4 execute query
        ResultSet rs = stmt.executeQuery("select * from student");
        while (rs.next())
          System.out.println(rs.getInt(1) + "  " + rs.getString(2) + "  " + rs.getInt(3));

        ResultSet rs2 = stmt.executeQuery("select * from student where age<21");
        System.out.println("students whose age is less than 21 are");
        while (rs2.next())
          System.out.println(rs2.getInt(1) + "  " + rs2.getString(2) + "  " + rs2.getInt(3));
        // step5 close the connection object
        con.close();

      } catch (Exception e) {
        System.out.println(e);
      }

    }

}
```

Output:

In Lab Task:

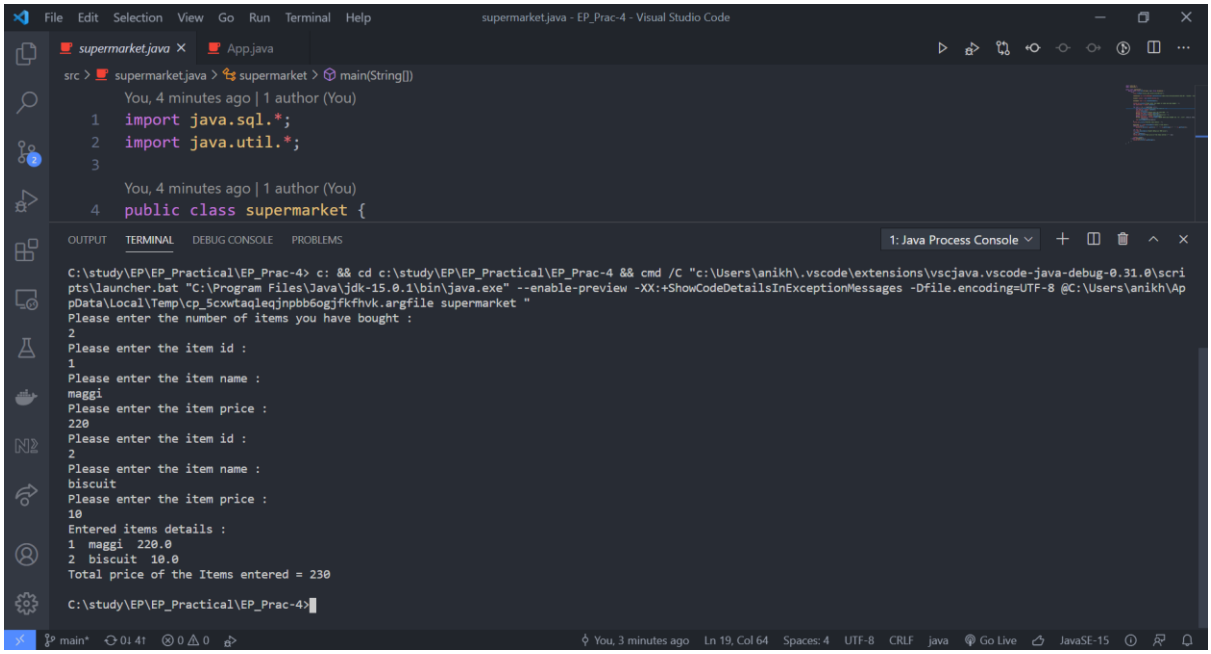
1. Dany went to the kings landing supermarket and bought some groceries. As a vendor you have to ask the numbers of items she bought and store the itemid, itemname, cost of the items and store them in the 'sales' table of database and also execute an SQL query to find the total cost and list out the item she bought.

Code:

```
import java.sql.*;
import java.util.*;

public class supermarket {
   public static void main(String[] args) throws Exception {
      try {
         Class.forName("oracle.jdbc.driver.OracleDriver");

         Connection con = DriverManager.getConnection("jdbc:oracle:thin
:@localhost:1521:XE", "system", "nikhil");

         Scanner scanner = new Scanner(System.in);

         Statement stmt = con.createStatement();

         System.out.println("Please enter the number of items you have b
ought : ");
         int numofitems = scanner.nextInt();

         for (int i = 0; i < numofitems; i++) {
            System.out.println("Please enter the item id : ");
            int item_id = scanner.nextInt();
            scanner.nextLine();
            System.out.println("Please enter the item name : ");
            String item_name = scanner.nextLine();
            System.out.println("Please enter the item price : ");
            double item_price = scanner.nextDouble();
            String insertquery = String.format("INSERT INTO sales VALUES (
%d, '%s', %.2f)", item_id, item_name,
                  item_price);
            stmt.executeUpdate(insertquery);
```

```
            }
            System.out.println("Entered items details : ");

            ResultSet rs = stmt.executeQuery("select * from sales");
            while (rs.next())
               System.out.println(rs.getInt(1) + "  " + rs.getString(2) + "  " + rs.g
        etFloat(3));

            int sum = 0;
            rs = stmt.executeQuery("SELECT SUM(price) FROM sales");
            rs.next();
            sum = rs.getInt(1);
            System.out.println("Total price of the Items entered = " + sum);

            scanner.close();
          } catch (Exception e) {
            System.out.println(e.getMessage());
          }
       }
    }
```

Output:



2. A workshop based on web development is being conducted in the university. Now James wants to know how many members of his class are interested in it and their details.

a. He created a new table namely Student in the Oracle Database with columns: Student_ID, Student_Name, Email, Date_of_Birth.

b. He stored some records into the Student table.

c. He needs to collect ID numbers and contact numbers of the interested students in a new table called"Workshop".

d. He wants to retrieve name and email address from the student table and update the Workshop table by creating 2 new columns and inserting the respective details.

e. If a student, suddenly wants to drop from attending the workshop, James is supposed to delete the student's data from workshop table.

Write the Java programs for the above situations using JDBC API.

Code:

```
import java.sql.*;

import java.util.*;

import java.text.SimpleDateFormat;


public class students {

   public static void main(String[] args) throws Exception {

      try {

         Class.forName("oracle.jdbc.driver.OracleDriver");


         Connection con = DriverManager.getConnection("jdbc:oracle:thin:@loc
alhost:1521:XE", "system", "nikhil");


         Statement stmt = con.createStatement();
```

```
        Scanner scanner = new Scanner(System.in);


        SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");


        // Create the Student table
        String query = "CREATE TABLE students (Student_ID NUMBER(10) PRIM
ARY KEY, Student_Name VARCHAR(255) NOT NULL, Email VARCHAR(255) NOT
NULL, Date_of_Birth NUMBER(6))";
        try {
           stmt.executeUpdate(query);
           System.out.println("Students table created successfully");
        } catch (Exception e) {
           System.out.println("Student Table already exists");
        }


        // Create the workshop table
        query = "CREATE TABLE workshop (Student_ID NUMBER(10) PRIMARY K
EY, contactNumber NUMBER(10) NOT NULL)";
        try {
           stmt.executeUpdate(query);
           System.out.println("workshop table created successfully");
        } catch (Exception e) {
           System.out.println("Workshop Table already exists");
        }


        for (int i = 6; i > 4; i++) {
```

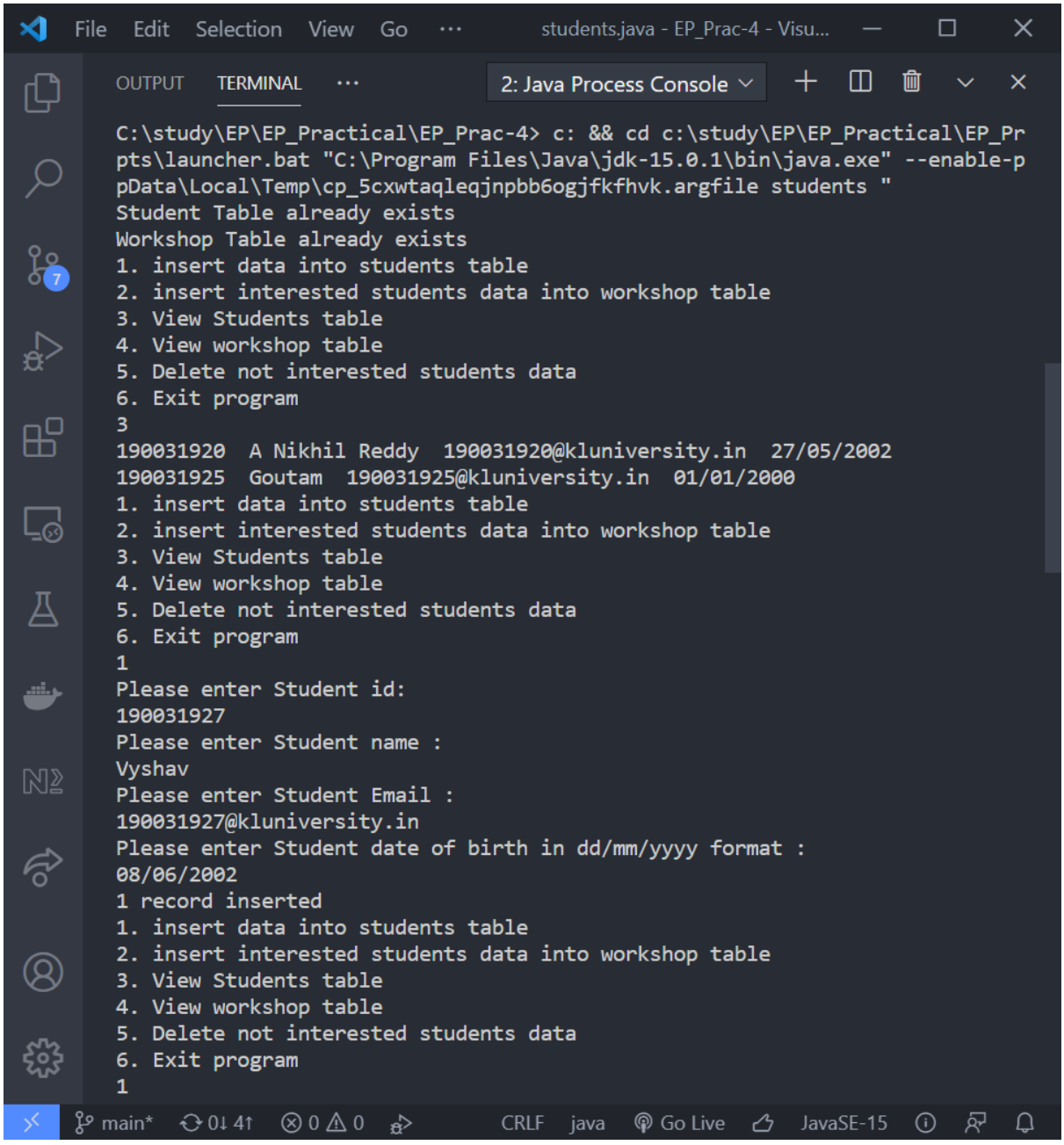
```
            System.out.println("1. insert data into students table");
            System.out.println("2. insert interested students data into workshop t
able");
            System.out.println("3. View Students table");
            System.out.println("4. View workshop table");
            System.out.println("5. Delete not interested students data");
            System.out.println("6. Exit program");
            int choice = scanner.nextInt();
            switch (choice) {
               case 1:
                  PreparedStatement prepstmt = con
                        .prepareStatement("INSERT INTO students VALUES (?, ?, ?,TO
_DATE(?, 'DD/MM/YYYY'))");
                  System.out.println("Please enter Student id: ");
                  int studentId = scanner.nextInt();
                  scanner.nextLine();
                  System.out.println("Please enter Student name : ");
                  String studentName = scanner.nextLine();
                  System.out.println("Please enter Student Email : ");
                  String studentEmail = scanner.nextLine();
                  System.out.println("Please enter Student date of birth in dd/mm/
yyyy format : ");
                  String studentdob = scanner.nextLine();
                  prepstmt.setInt(1, studentId);
                  prepstmt.setString(2, studentName);
                  prepstmt.setString(3, studentEmail);
                  prepstmt.setString(4, studentdob);
```

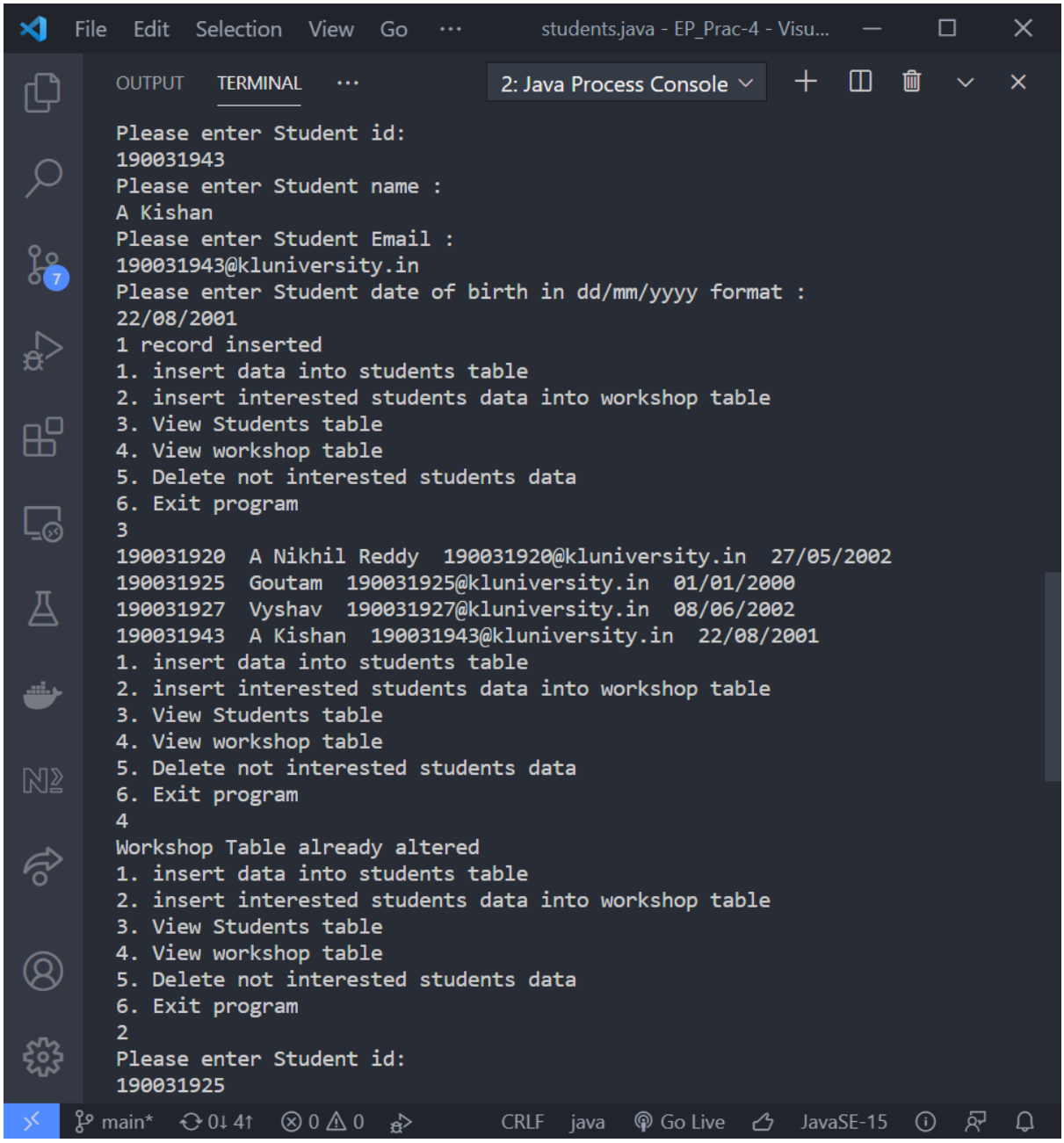
```
                int num = prepstmt.executeUpdate();
                System.out.println(num + " record inserted");
                break;

            case 2:
                PreparedStatement intoWorkshop = con
                        .prepareStatement("INSERT INTO workshop VALUES (?, ?, nul
l, null)");
                System.out.println("Please enter Student id: ");
                int insertedstudentId = scanner.nextInt();
                System.out.println("Please enter Student contact number: ");
                int contactNumber = scanner.nextInt();
                intoWorkshop.setInt(1, insertedstudentId);
                intoWorkshop.setInt(2, contactNumber);
                num = intoWorkshop.executeUpdate();
                System.out.println(num + " record inserted");
                break;

            case 3:
                ResultSet rs = stmt.executeQuery("select * from students");
                while (rs.next())
                   System.out.println(rs.getInt(1) + "  " + rs.getString(2) + "  " + rs.
getString(3) + "  "
                           + formatter.format(rs.getDate(4)));
                break;
            case 4:
                try {
```

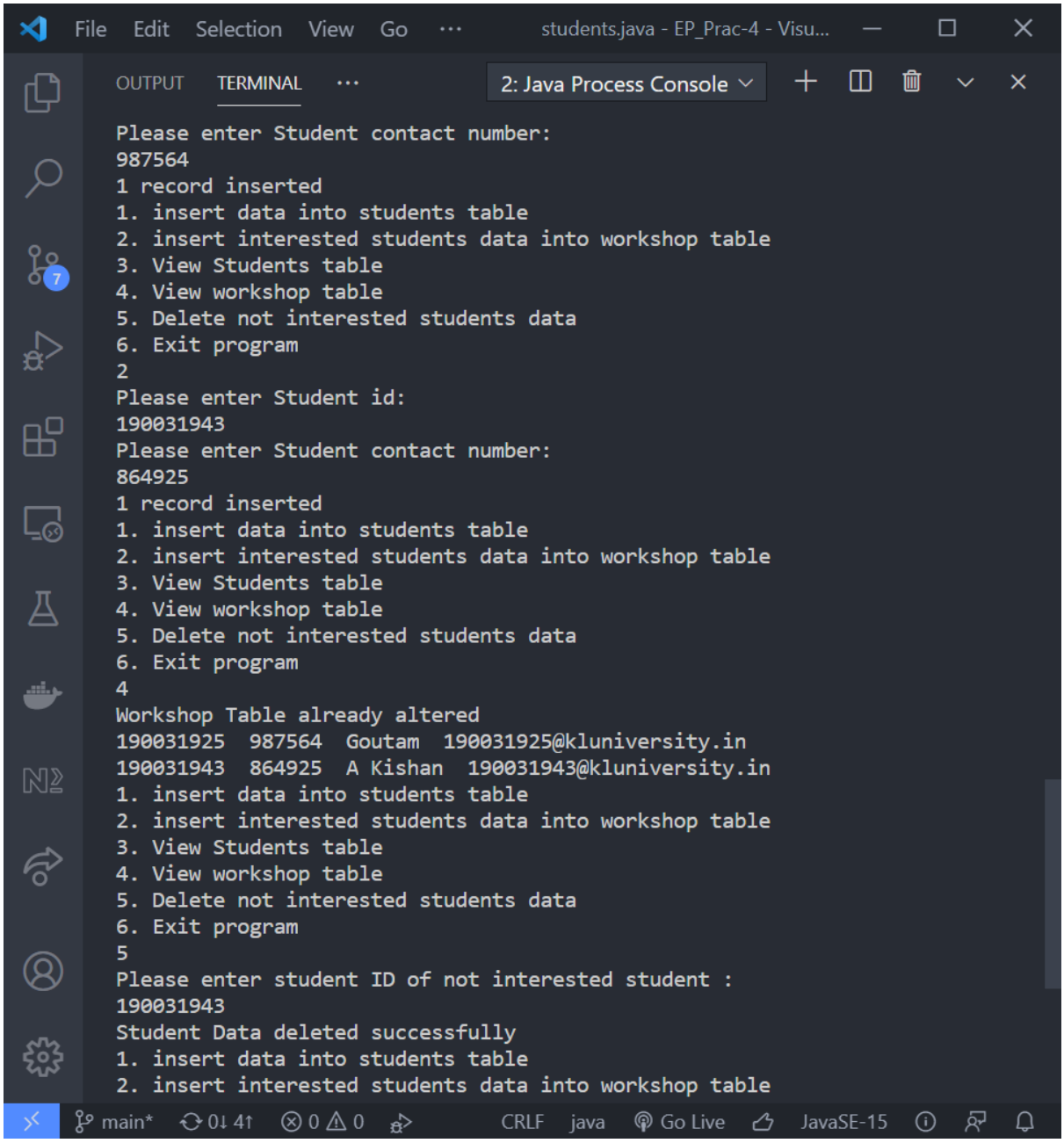
```
                    query = "ALTER TABLE workshop ADD (studentName VARCHAR(
255), studentEmail VARCHAR(255))";
                    stmt.executeUpdate(query);
                    System.out.println("workshop table altered successfully");
                } catch (Exception e) {
                    System.out.println("Workshop Table already altered");
                }
                try {
                    query = "UPDATE workshop t1 SET (StudentName, studentEmai
l) = (SELECT t2.Student_Name, t2.Email FROM students t2 WHERE t1.Student_I
D=t2.Student_ID) WHERE EXISTS (SELECT 1 FROM students t2 WHERE t1.Stude
nt_ID=t2.Student_ID )";
                    stmt.executeUpdate(query);
                } catch (Exception e) {
                    System.out.println();
                }
                rs = stmt.executeQuery("select * from workshop");
                while (rs.next())
                    System.out.println(rs.getInt(1) + "  " + rs.getInt(2) + "  " + rs.get
String(3) + "  "
                            + rs.getString(4));
                break;
            case 5:
                int notInterestStudentId;
                System.out.println("Please enter student ID of not interested stu
dent : ");
                notInterestStudentId = scanner.nextInt();
```

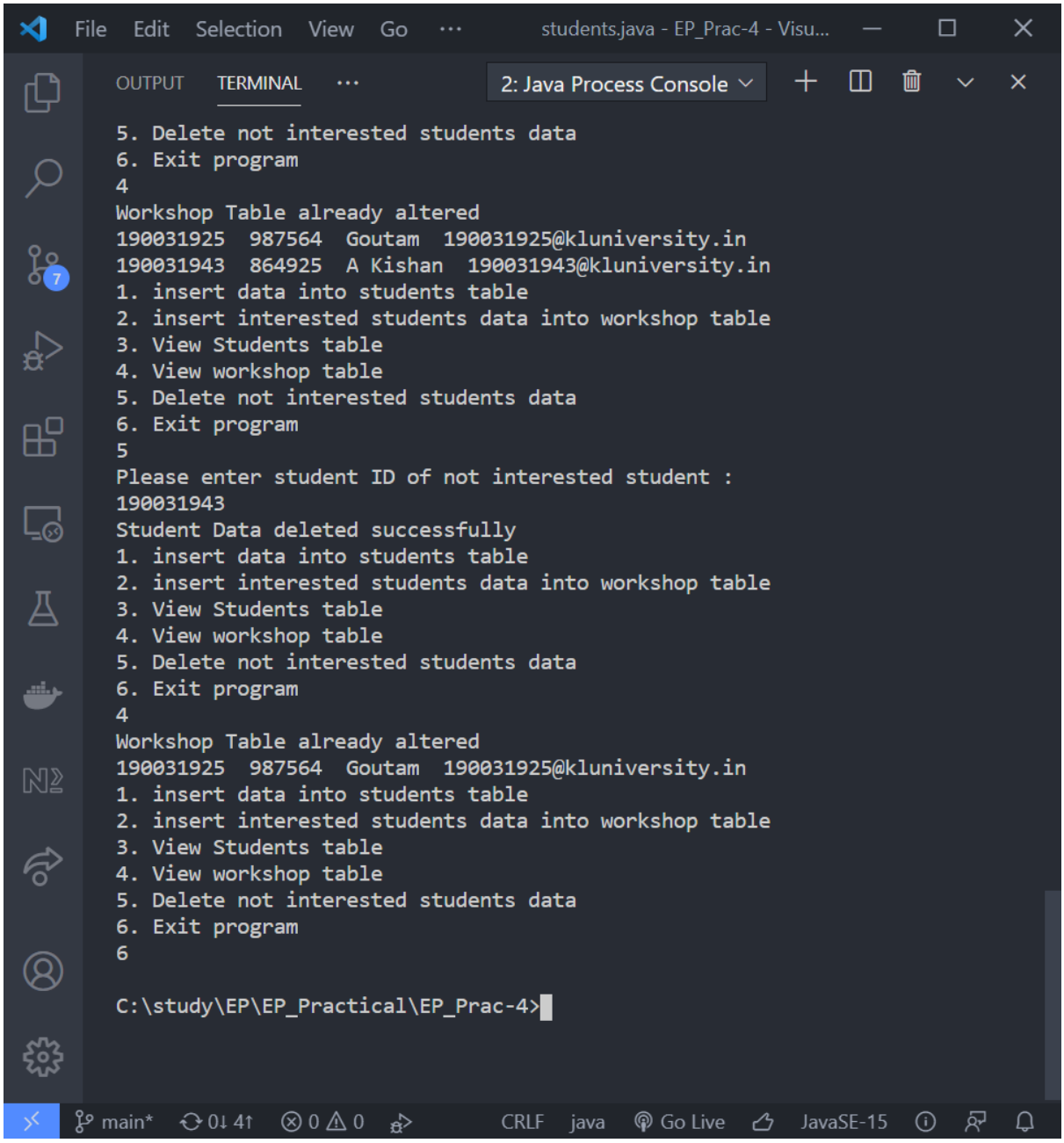
```
                PreparedStatement deleteData = con.prepareStatement("DELETE
FROM workshop WHERE Student_ID=?");
                deleteData.setInt(1, notInterestStudentId);
                try {
                   deleteData.executeUpdate();
                   System.out.println("Student Data deleted successfully");
                } catch (Exception e) {
                   System.out.println("Error: " + e.getMessage());
                   System.out.println("Please try again");
                }
                break;
             case 6:
                i = 2;
                break;

          }
       }

       scanner.close();

       stmt.close();
    } catch (Exception e) {
       System.out.println("Error: " + e.getMessage());
    }
  }
}
```

Output:


File Edit Selection View Go ··· students.java - EP_Prac-4 - Visu...
OUTPUT TERMINAL ··· 2: Java Process Console
C:\study\EP\EP_Practical\EP_Prac-4> c: && cd c:\study\EP\EP_Practical\EP_Pr
pts\launcher.bat "C:\Program Files\Java\jdk-15.0.1\bin\java.exe" --enable-p
pData\Local\Temp\cp_5cxwtaqleqjnpbb6ogjfkfhvk.argfile students "
Student Table already exists
Workshop Table already exists
1. insert data into students table
2. insert interested students data into workshop table
3. View Students table
4. View workshop table
5. Delete not interested students data
6. Exit program
3
190031920 A Nikhil Reddy 190031920@kluniversity.in 27/05/2002
190031925 Goutam 190031925@kluniversity.in 01/01/2000
1. insert data into students table
2. insert interested students data into workshop table
3. View Students table
4. View workshop table
5. Delete not interested students data
6. Exit program
1
Please enter Student id:
190031927
Please enter Student name :
Vyshav
Please enter Student Email :
190031927@kluniversity.in
Please enter Student date of birth in dd/mm/yyyy format :
08/06/2002
1 record inserted
1. insert data into students table
2. insert interested students data into workshop table
3. View Students table
4. View workshop table
5. Delete not interested students data
6. Exit program
1
main* 0↓ 4↑ ⊗ 0 ⚠ 0 CRLF java Go Live JavaSE-15


File Edit Selection View Go ··· students.java - EP_Prac-4 - Visu...
OUTPUT TERMINAL ···
2: Java Process Console
Please enter Student id:
190031943
Please enter Student name :
A Kishan
Please enter Student Email :
190031943@kluniversity.in
Please enter Student date of birth in dd/mm/yyyy format :
22/08/2001
1 record inserted
1. insert data into students table
2. insert interested students data into workshop table
3. View Students table
4. View workshop table
5. Delete not interested students data
6. Exit program
3
190031920 A Nikhil Reddy 190031920@kluniversity.in 27/05/2002
190031925 Goutam 190031925@kluniversity.in 01/01/2000
190031927 Vyshav 190031927@kluniversity.in 08/06/2002
190031943 A Kishan 190031943@kluniversity.in 22/08/2001
1. insert data into students table
2. insert interested students data into workshop table
3. View Students table
4. View workshop table
5. Delete not interested students data
6. Exit program
4
Workshop Table already altered
1. insert data into students table
2. insert interested students data into workshop table
3. View Students table
4. View workshop table
5. Delete not interested students data
6. Exit program
2
Please enter Student id:
190031925
main* 0↓ 4↑ 0 0 CRLF java Go Live JavaSE-15


File Edit Selection View Go ··· students.java - EP_Prac-4 - Visu...
OUTPUT TERMINAL ···
2: Java Process Console
Please enter Student contact number:
987564
1 record inserted
1. insert data into students table
2. insert interested students data into workshop table
3. View Students table
4. View workshop table
5. Delete not interested students data
6. Exit program
2
Please enter Student id:
190031943
Please enter Student contact number:
864925
1 record inserted
1. insert data into students table
2. insert interested students data into workshop table
3. View Students table
4. View workshop table
5. Delete not interested students data
6. Exit program
4
Workshop Table already altered
190031925 987564 Goutam 190031925@kluniversity.in
190031943 864925 A Kishan 190031943@kluniversity.in
1. insert data into students table
2. insert interested students data into workshop table
3. View Students table
4. View workshop table
5. Delete not interested students data
6. Exit program
5
Please enter student ID of not interested student :
190031943
Student Data deleted successfully
1. insert data into students table
2. insert interested students data into workshop table
main* 0↓ 4↑ 0 0 CRLF java Go Live JavaSE-15



Post Lab Task

1. Tony is a wildlife photographer. He went to Amazon rainforest to explore the beauty of the flora and fauna. He captured pictures of different trees and animals. He also described each animal and plant in a separate notepad. Now he wants to save the name, the category (animal / plant), its image and the description file of the species into a database. Help Tony by providing him with an executable java program.

   a. Create a table with columns Name, Category, Image and Description_File before executing the file.

   b. The name, category, path of the image(.jpg), path of the description file (.txt) should be read as input from the console.

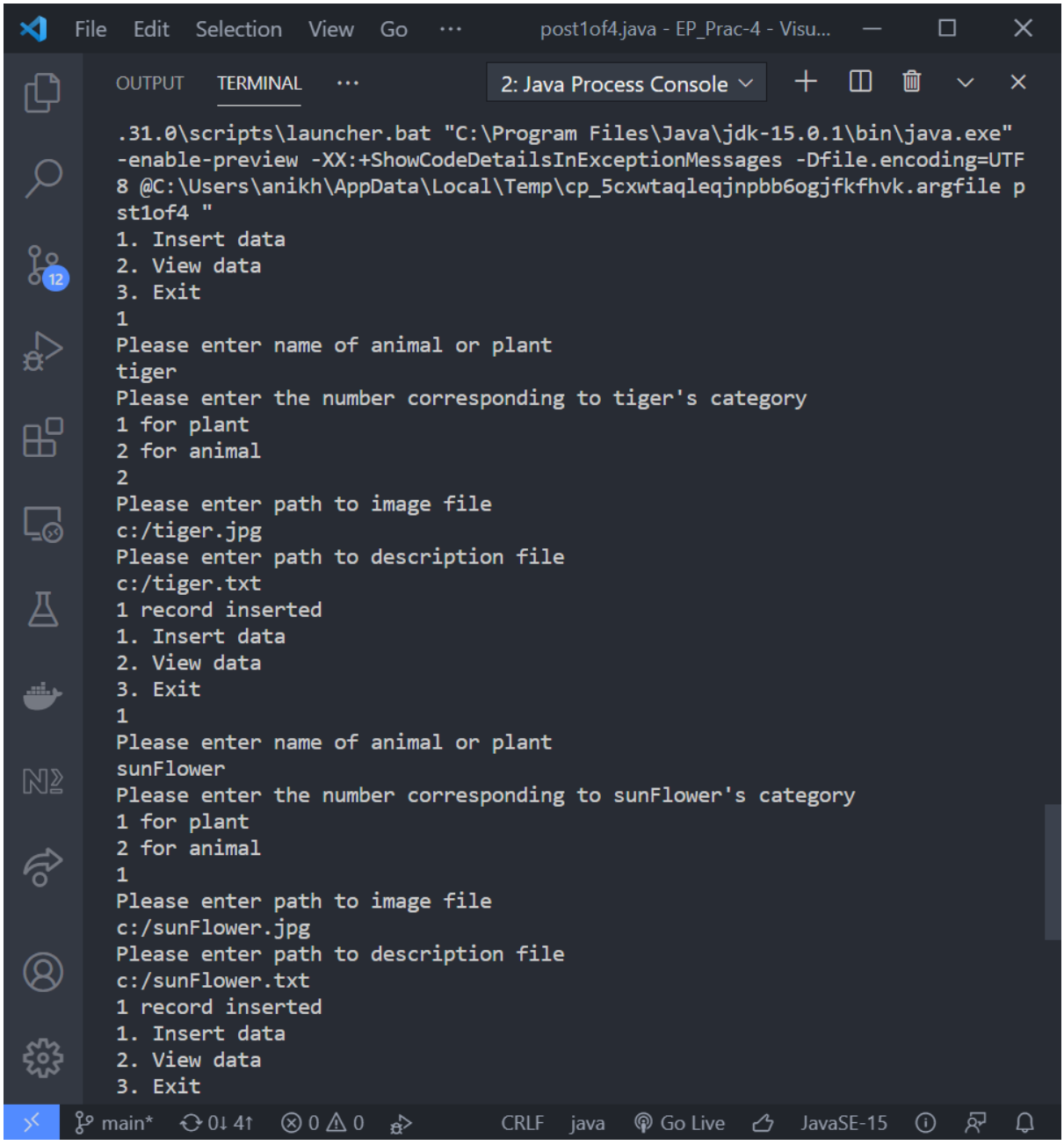
Code:

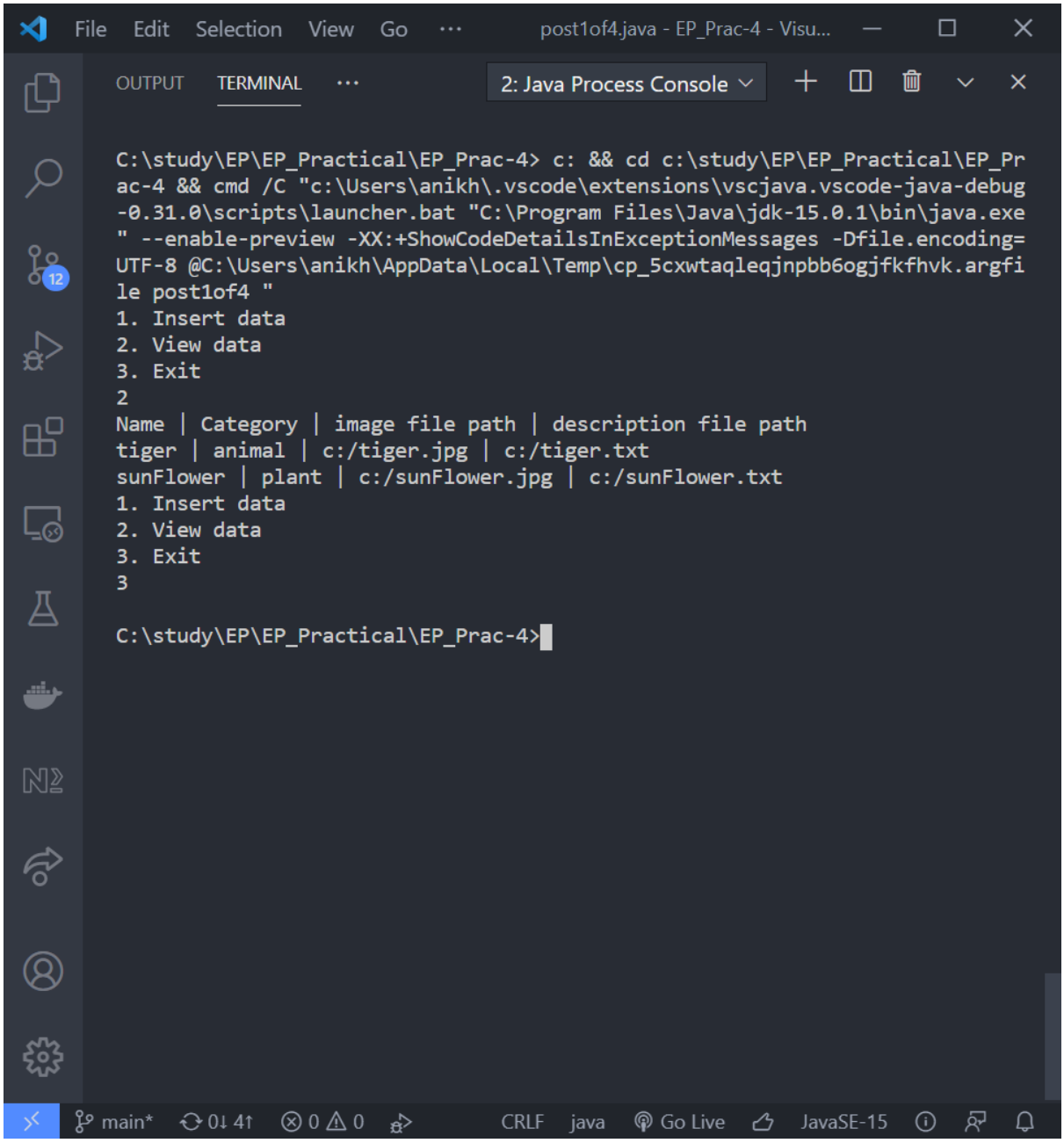
```
import java.sql.*;
import java.util.Scanner;


public class post1of4 {
   public static void main(String[] args) throws Exception {
      try {
         Class.forName("oracle.jdbc.driver.OracleDriver");


         Connection con = DriverManager.getConnection("jdbc:oracle:thin:@loc
alhost:1521:XE", "system", "nikhil");


         Statement stmt = con.createStatement();


         Scanner scanner = new Scanner(System.in);


         // Code used to create wildlife table
         // CREATE TABLE wildlife (
         // name VARCHAR(255) NOT NULL,
         // category NUMBER(1) NOT NULL,
         // image VARCHAR(255) NOT NULL,
         // description VARCHAR(255) NOT NULL
         // );
         for (int i = 6; i > 4; i++) {
            System.out.println("1. Insert data\n2. View data\n3. Exit");
            int choice = scanner.nextInt();
            scanner.nextLine();
```

```
            switch (choice) {
               case 1:
                  try {
                     PreparedStatement insert = con.prepareStatement("INSERT IN
TO wildlife VALUES (?, ?, ?, ?)");

                     System.out.println("Please enter name of animal or plant");
                     String name = scanner.nextLine();
                     insert.setString(1, name);

                     System.out.println("Please enter the number corresponding to
" + name + "'s category");
                     System.out.println("1 for plant \n2 for animal");
                     int category = scanner.nextInt();
                     scanner.nextLine();
                     insert.setInt(2, category);

                     System.out.println("Please enter path to image file");
                     String imgfilepath = scanner.nextLine();
                     insert.setString(3, imgfilepath);

                     System.out.println("Please enter path to description file");
                     String txtfilepath = scanner.nextLine();
                     insert.setString(4, txtfilepath);

                     int num = insert.executeUpdate();
                     System.out.println(num + " record inserted");
```

```
                } catch (Exception e) {
                    System.out.println(e + "\n Please try again");
                }
                break;
            case 2:
                ResultSet rs = stmt.executeQuery("select * from wildlife");
                System.out.println("Name | Category | image file path | descripti
on file path");
                while (rs.next()) {
                    String category = (rs.getInt(2) == 1) ? "plant" : "animal";
                    System.out.println(rs.getString(1) + " | " + category + " | " + rs.g
etString(3) + " | "
                            + rs.getString(4));
                }
                break;
            case 3:
                con.close();
                scanner.close();
                i = 2;
                break;
            }
        }

    } catch (Exception e) {
        System.out.println("Error: " + e);
    }
  }}
```

Output:


File Edit Selection View Go ··· post1of4.java - EP_Prac-4 - Visu...
OUTPUT TERMINAL ···
2: Java Process Console
.31.0\scripts\launcher.bat "C:\Program Files\Java\jdk-15.0.1\bin\java.exe" -enable-preview -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=UTF8 @C:\Users\anikh\AppData\Local\Temp\cp_5cxwtaqleqjnpbb6ogjfkfhvk.argfile post1of4 "
1. Insert data
2. View data
3. Exit
1
Please enter name of animal or plant
tiger
Please enter the number corresponding to tiger's category
1 for plant
2 for animal
2
Please enter path to image file
c:/tiger.jpg
Please enter path to description file
c:/tiger.txt
1 record inserted
1. Insert data
2. View data
3. Exit
1
Please enter name of animal or plant
sunFlower
Please enter the number corresponding to sunFlower's category
1 for plant
2 for animal
1
Please enter path to image file
c:/sunFlower.jpg
Please enter path to description file
c:/sunFlower.txt
1 record inserted
1. Insert data
2. View data
3. Exit
main* 0↓ 4↑ ⊗ 0 ⚠ 0
CRLF java Go Live JavaSE-15



2. After exploring the dense forest, Tony returns to home and wants to publish his work in a magazine. He is supposed to give the information to the publisher in the following way.

   a. Two folders namely Animals and Plants should be created on desktop

   b. The image of every animal should be saved as "animal_name.jpg" and the description file as "animal_name.txt" in the Animal folder

   c. Similarly, the image of every plant should be saved as "plant_name.jpg" and the description file as "plant_name.txt" in the Plant folder

Code:

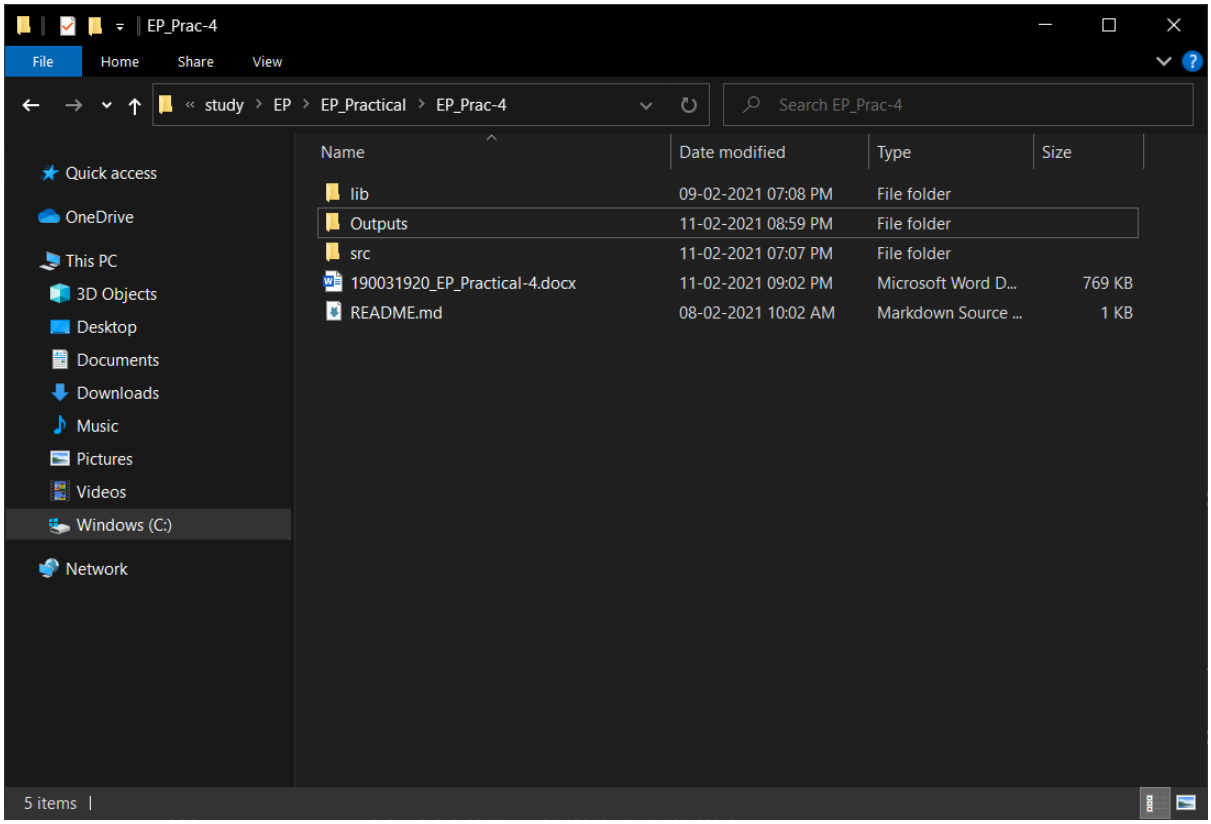
```
import java.io.*;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.sql.*;

public class post2of4 {
   public static void main(String[] args) throws Exception {
      Class.forName("oracle.jdbc.driver.OracleDriver");

      Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localh
ost:1521:XE", "system", "nikhil");

      Statement stmt = con.createStatement();

      String path = "C:/study/EP/EP_Practical/EP_Prac-4/";

      ResultSet rs = stmt.executeQuery("select * from wildlife");

      // Create plants folder
      String plantfolder = path.concat("Plants");
      try {
         Files.createDirectories(Paths.get(plantfolder));
      } catch (Exception e) {
         System.out.println("Error: " + e);
      }
```

```
// Create animals folder
String animalfolder = path.concat("Animals");
try {
   Files.createDirectories(Paths.get(animalfolder));
} catch (Exception e) {
   System.out.println("Error: " + e);
}

String imgFileName;
String txtFileName;
File imgfile = null;
File txtfile = null;

while (rs.next()) {
   if (rs.getInt(2) == 2) {
      imgFileName = "animal_" + rs.getString(1) + ".jpg";
      txtFileName = "animal_" + rs.getString(1) + ".txt";
      imgfile = new File(animalfolder + "/" + imgFileName);
      txtfile = new File(animalfolder + "/" + txtFileName);
   } else if (rs.getInt(2) == 1) {
      imgFileName = "plant_" + rs.getString(1) + ".jpg";
      txtFileName = "plant_" + rs.getString(1) + ".txt";
      imgfile = new File(plantfolder + "/" + imgFileName);
      txtfile = new File(plantfolder + "/" + txtFileName);
   }
   try {
```

```
            FileWriter imgfw = new FileWriter(imgfile.getAbsoluteFile());
            FileWriter txtfw = new FileWriter(txtfile.getAbsoluteFile());
            BufferedWriter imgbw = new BufferedWriter(imgfw);
            BufferedWriter txtbw = new BufferedWriter(txtfw);
            imgbw.close();
            txtbw.close();
        } catch (IOException e) {
            e.printStackTrace();
            System.exit(-1);
        }

    }
  }
}
```

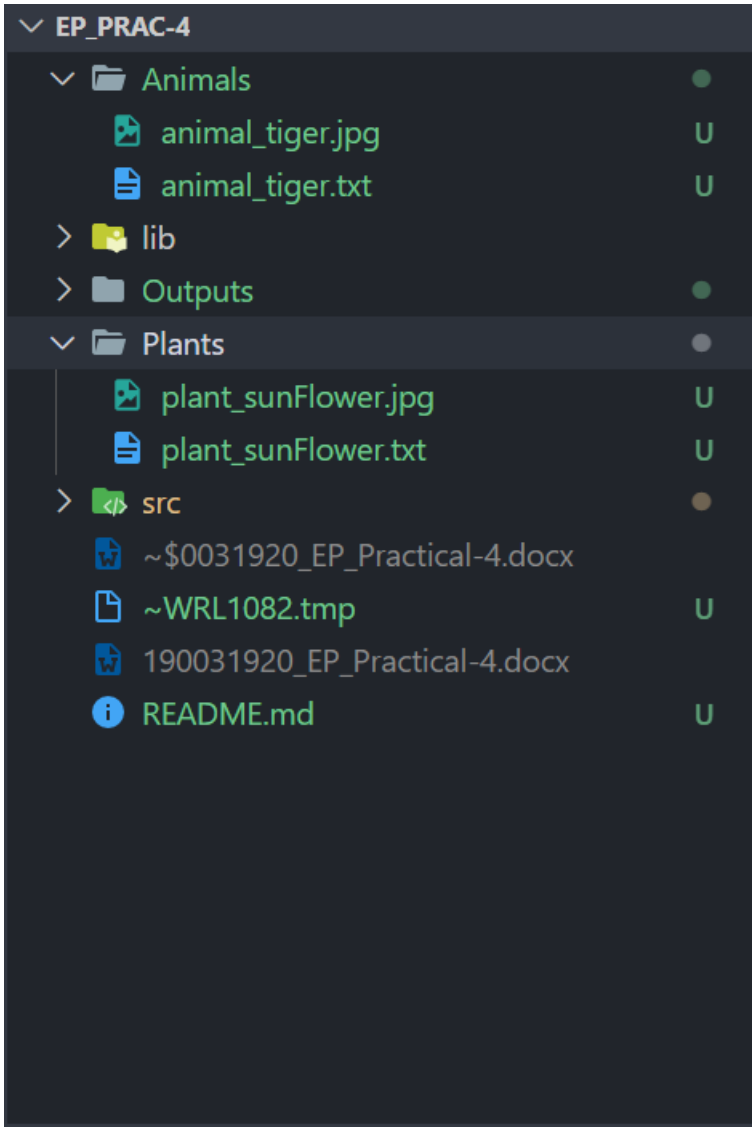
Output:

File structure before running the code:



File structure after running the code:

We can clearly see the new Animals and Plants folders with their respective image and text files


EP_PRAC-4
Animals
animal_tiger.jpg
U
animal_tiger.txt
U
lib
Outputs
Plants
plant_sunFlower.jpg
U
plant_sunFlower.txt
U
src
~$0031920_EP_Practical-4.docx
~WRL1082.tmp
U
190031920_EP_Practical-4.docx
README.md
U