Learning Java 9: Databases and Multithreading in Java

User Assignment-1

Problem Statement:

- Create a json file called customers.json, with fields custId, custName, city and pin and populate with some data.
- Write a hibernate program to read the json data from the json file and store it in the Postgre mysql database.

```
☆ Debug  Project Explorer ×

                                                       > 📂 Collectionss
                                                         1 package jsonproject;
                                                         2⊖ import java.io.FileWriter;
> 📂 Complete_java_se8_developer_bootcamp (in complete_ja
                                                         3 import java.io.IOException;
> 📂 employeethread
> # IRISAssignments
                                                         5 import org.json.simple.JSONArray;
> 🔂 JavaTpoint_Programs
                                                         6 import org.json.simple.JSONObject;

        isonproject

  8 public class jsonfile1 {
     jsonproject
                                                        10<sup>©</sup> public static void main(String[] args) {
       v 💹 jsonfile1.java
                                                              JSONObject custs = new JSONObject();

custs.put("custid", 100);

custs.put("custname", "satya prakash das");

custs.put("city", "bhubaneswar");

custs.put("pin" 1001);
                                                        11
          > 🔐 jsonfile1
                                                        12
  > March JRE System Library [JavaSE-17]
                                                       13
   > Neferenced Libraries
                                                       14
    customers.json
                                                        15
                                                              custs.put("pin", 1001);
                                                        16
> # Milestone_1
                                                        17
                                                              JSONObject custsObj = new JSONObject();
> Multithreading
                                                        18
                                                              custsObj.put("Customer1", custs);

> 
    Multithreading_and_Multiprogramming

                                                        19
> 📂 Practise codes for Milestone 1
                                                        20
                                                              JSONObject custs2 = new JSONObject();
> # the_complete_java_developer_course (in JAVA)
                                                       221
                                                              custs2.put("custid", 101);
                                                              custs2.put("custname", "patel");
custs2.put("city", "nyc");
                                                       <u>2</u>22
                                                       23
                                                       124
                                                              custs2.put("pin", 222222);
                                                        25
                                                        26
                                                              JSONObject custsObj2 = new JSONObject();
                                                        27
                                                              custsObj2.put("Customer2", custs2);
                                                        28
                                                        29
                                                              JSONArray custList = new JSONArray();
                                                        30
                                                              custList.add(custsObj);
                                                       31
                                                              custList.add(custsObj2);
                                                        32
                                                        33
                                                              try(FileWriter file = new FileWriter("customers.json")){
                                                        34
                                                               file.write(custList.toJSONString());
                                                        35
                                                               file.flush();
                                                        36
                                                        37
                                                              catch(IOException e) {e.printStackTrace();}
                                                        38
                                                        39
                                                        40 }
```

```
💹 jsonfile1.java
               {} *customers.json ×
 1[{
 2
     "Customer1" : {
 3
       "pin" : 1001,
       "city" : "bhubaneswar",
 4
       "custid" : 100,
 5
 6
       "custname" : "satya prakash das"
 7
     }
 8}, {
 9
     "Customer2" : {
       "pin" : 222222,
10
       "city" : "nyc",
11
       "custid" : 101,
12
       "custname" : "patel"
13
14
15 } ]
```

2.)

```
🛮 assign1.java × 📳 Customer.java
                               customers.json
                                               JSONProcessor.java
  package Product4.Hibernate;
  3 import org.hibernate. Session;
  4 import org.hibernate.SessionFactory;
  5 import org.hibernate.Transaction;
  6 import org.hibernate.cfg.Configuration;
  8 import java.net.URL;
  9 import java.util.List;
 11 public class assign1 {
        private static SessionFactory factory;
 12
 13
 149
        public static void main(String args[]) throws Exception {
 15
            setUp();
 16
 17
            URL file_path = Main.class.getClassLoader().getResource("customers.json");
 18
            JSONProcessor jsonProcessor = new JSONProcessor(file_path.getPath());
 19
            List<Customer> customer = jsonProcessor.parseFile();
 20
 21
            customer.forEach(Main::addCustomer);
 22
        }
 23
 249
        private static void setUp() {
 25
            factory = new Configuration()
 26
                     .addAnnotatedClass(Customer.class)
 27
                     .configure()
 28
                     .buildSessionFactory();
 29
        }
 30
 31⊖
        private static Integer addCustomer(Customer customer) {
 32
            Session session = factory.openSession();
 33
            Transaction tx = session.beginTransaction();
 34
            Integer customerId = (Integer) session.save(customer);
 35
            tx.commit();
 36
 37
            return customerId;
 38
        }
39 }
```

```
🗗 📓 assign1.java
                                                JSONProcessor.java
  1 import javax.persistence.*;
    2
    3 @Entity
    4 @Table(name = "colibri.customer")
    5 public class Car {
          @Column(name = "cusId")
          private String cusID;
    7
    80
          @Column(name = "cusName")
    9
          private String cusName;
          @Column(name = "city")
   10⊝
   11
          private String city;
          @Column(name = "pin")
   12⊖
   13
          private int pin;
   149
          @Id
          @GeneratedValue(strategy = GenerationType.IDENTITY)
   15
   16
          @Column(name = "id")
   17
          private int id;
   18
   19⊝
          public Customer(String cusID, String cusName, String city,int pin) {
   20
              this.cusID = cusID;
   21
              this.cusName = cusName;
   22
              this.city = city;
   23
              this.pin = pin;
   24
          }
   25
   26⊜
          public Customer()
   27
   28
   29⊝
          public int getPin() {
   30
              return pin;
   31
          }
   32
   33⊝
          public String getCity() {
   34
              return city;
   35
          }
   36
   37⊝
          public String getcusName() {
   38
              return cusName;
   39
          }
 40
                                                                A atimata Mindan
```

```
assign1.java
                             Customer.java
                                                      customers.json ×   JSONProcessor.java
         19 {
          2⊝
                    "customers": [
          3⊝
                       {
                          "custId": "1089",
         4
                          "custName": "satya prakash das",
          5
                          "city": "Bubaneswar",
         6
                          "pin": 100205
         7
         8
                      },
         9⊝
                          "custId": "1897",
        10
        11
                          "custName": "Debasishs Patel",
                          "city": "New York",
        12
       13
                         "pin": 753021
        14
                      },
        15⊜
                      {
                          "custId": "1007",
        16
        17
                          "custName": "Arpan",
                          "city": "Kolkata",
        18
        19
                          "pin": 669855;
        20
        21
                    ]
                }
        22
■ assign1.java ■ Customer.java × © customers.json ■ JSONProcessor.java ×
   1 package Product4.Hibernate;
   4⊖ import org.json.simple.JSONArray;
   5 import org.json.simple.JSONObject;
6 import org.json.simple.parser.JSONParser;
    7 import org.json.simple.parser.ParseException;
    9 import java.io.FileReader;
   10 import java.io.IOException;
   11 import java.util.List;
   12 import java.util.stream.Collectors;
   14 public class JSONProcessor {
         private final String targetFilePath;
   16
          JSONProcessor(String targetFilePath) {
   18
             this.targetFilePath = targetFilePath;
   19
          public List<Customer> parseFile() throws IOException, ParseException {
             JSONParser parser = new JSONParser();
JSONObject json = (JSONObject) parser.parse(new FileReader(targetFilePath));
  22
23
24
25
26
27
28
29
             JSONArray customers = (JSONArray) json.get("customers");
             List<JSONObject> customerList = (List<JSONObject>) customers.stream().collect(Collectors.toList());
             return customerList.stream()
.map(x -> new Customer((String) x.get("cusId"), (String) x.get("cusName"), (String) x.get("city"), (Double) x.get("pin")))
                 .collect(Collectors.toList());
   32 }
```