1) My project's paths and structure
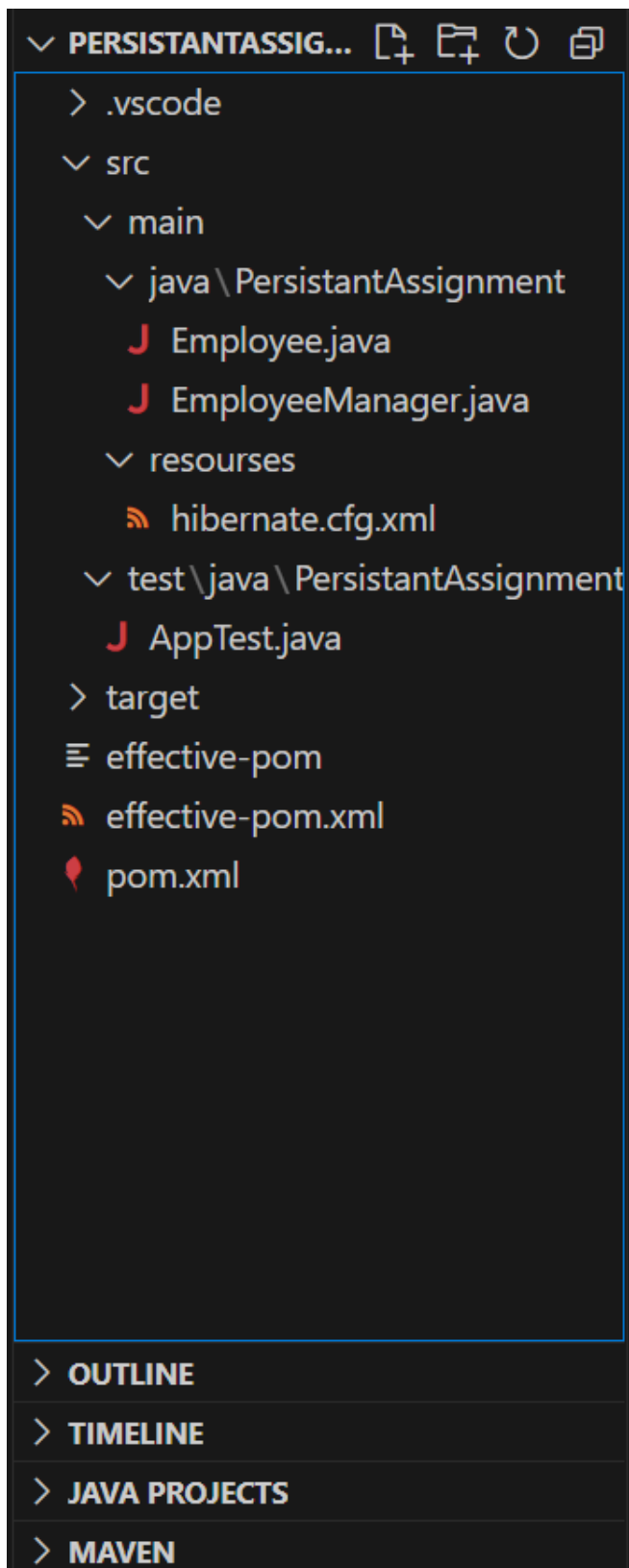
2)Employe.java

```java
1   package PersistantAssignment;
2
3   import javax.persistence.Entity;
4   import javax.persistence.Id;
5   import javax.persistence.GeneratedValue;
6   import javax.persistence.GenerationType;
7
8   @Entity
9   public class Employee {
10      @Id
11      @GeneratedValue(strategy = GenerationType.IDENTITY)
12      private int id;
13      private String name;
14      private String department;
15      private double salary;
16
17      // Getters and Setters
18      public int getId() {
19          return id;
20      }
21      public void setId(int id) {
22          this.id = id;
23      }
24      public String getName() {
25          return name;
26      }
27      public void setName(String name) {
28          this.name = name;
29      }
30      public String getDepartment() {
31          return department;
32      }
33      public void setDepartment(String department) {
34          this.department = department;
35      }
36      public double getSalary() {
37          return salary;
38      }
39      public void setSalary(double salary) {
40          this.salary = salary;
41      }
42  }
43
```

3) EmployeeManager.java

```java
package PersistantAssignment;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

public class EmployeeManager {
    private static SessionFactory factory;

    public static void main(String[] args) {
        try {
            factory = new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            System.err.println("Failed to create sessionFactory object." + ex);
            throw new ExceptionInInitializerError(ex);
        }

        EmployeeManager manager = new EmployeeManager();

        // Create a new employee
        Integer empID1 = manager.addEmployee(name:"John Doe", department:"HR", salary:50000);

        // Retrieve the employee
        manager.getEmployee(empID1);
    }

    public Integer addEmployee(String name, String department, double salary) {
        Session session = factory.openSession();
        Transaction tx = null;
        Integer employeeID = null;

        try {
            tx = session.beginTransaction();
            Employee employee = new Employee();
            employee.setName(name);
            employee.setDepartment(department);
            employee.setSalary(salary);
            employeeID = (Integer) session.save(employee);
            tx.commit();
        }
```

```java
42     }    catch (Exception e) {
43              if (tx != null) tx.rollback();
44              e.printStackTrace();
45          } finally {
46              session.close();
47          }
48          return employeeID;
49      }
50
51      public void getEmployee(Integer EmployeeID) {
52          Session session = factory.openSession();
53          try {
54              Employee employee = session.get(Employee.class, EmployeeID);
55              if (employee != null) {
56                  System.out.println("Employee Details: " + employee.getName() + ", " + employee.ge
57              }
58          } catch (Exception e) {
59              e.printStackTrace();
60          } finally {
61              session.close();
62          }
63      }
```

4) pom.xml

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>PersistantAssignment</groupId>
    <artifactId>PersistantAssignment</artifactId>
    <version>1.0-SNAPSHOT</version>

  <build>
    <plugins>
        <plugin>
            <groupId>org.codehaus.mojo</groupId>
            <artifactId>exec-maven-plugin</artifactId>
            <version>3.3.0</version>
            <configuration>
                <mainClass>PersistantAssignment.EmployeeManager</mainClass>
            </configuration>
        </plugin>
    </plugins>
```

```xml
    </build>

    <dependencies>
        <!-- Hibernate -->
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>5.4.33.Final</version>
        </dependency>
        <dependency>
            <groupId>javax.persistence</groupId>
            <artifactId>javax.persistence-api</artifactId>
            <version>2.2</version>
        </dependency>
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-c3p0</artifactId>
            <version>5.4.33.Final</version>
        </dependency>

        <!-- MySQL connector -->
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.25</version>
        </dependency>

        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```

5) hibernate.cfg.xml

```xml
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate
Configuration DTD 3.0//EN"
        "http://hibernate.sourceforge.net/hibernate-configuration-
3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <!-- JDBC Database connection settings -->
        <property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</pro
perty>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/yourDataba
seName</property>
        <property
name="hibernate.connection.username">AkashBabu</property>
        <property
name="hibernate.connection.password">akashbabu@123</property>

        <!-- JDBC connection pool settings -->
        <property name="hibernate.c3p0.min_size">5</property>
        <property name="hibernate.c3p0.max_size">20</property>
        <property name="hibernate.c3p0.timeout">300</property>
        <property name="hibernate.c3p0.max_statements">50</property>
        <property
name="hibernate.c3p0.idle_test_period">3000</property>

        <!-- SQL dialect -->
        <property
name="hibernate.dialect">org.hibernate.dalect.MySQL5Dialect</property>

        <!-- Enable Hibernate's automatic session context management -
->
        <property
name="hibernate.current_session_context_class">thread</property>

        <!-- Echo all executed SQL to stdout -->
        <property name="hibernate.show_sql">true</property>
```

```xml
        <!-- Drop and re-create the database schema on startup -->
        <property name="hibernate.hbm2ddl.auto">update</property>

        <!-- Mapping files -->
        <mapping class="PersistantAssignment.Employee"/>
    </session-factory>
</hibernate-configuration>
```