Dharmsinh Desai University, Nadiad

Department of Information Technology

Subject Name: Design Patterns and Application Frameworks

Lab Work

B.Tech. IT, Sem: VI

Submitted By:

Roll No.: IT074

Name: Kevla Merja K.

Batch: H4

ID: 19ITUOS100

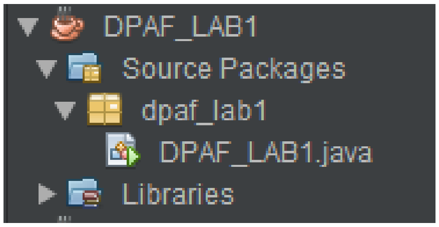
Index

Experiment-1

Aim : Implement programming solution that uses Abstract class, interface, Object composition, and inheritance.

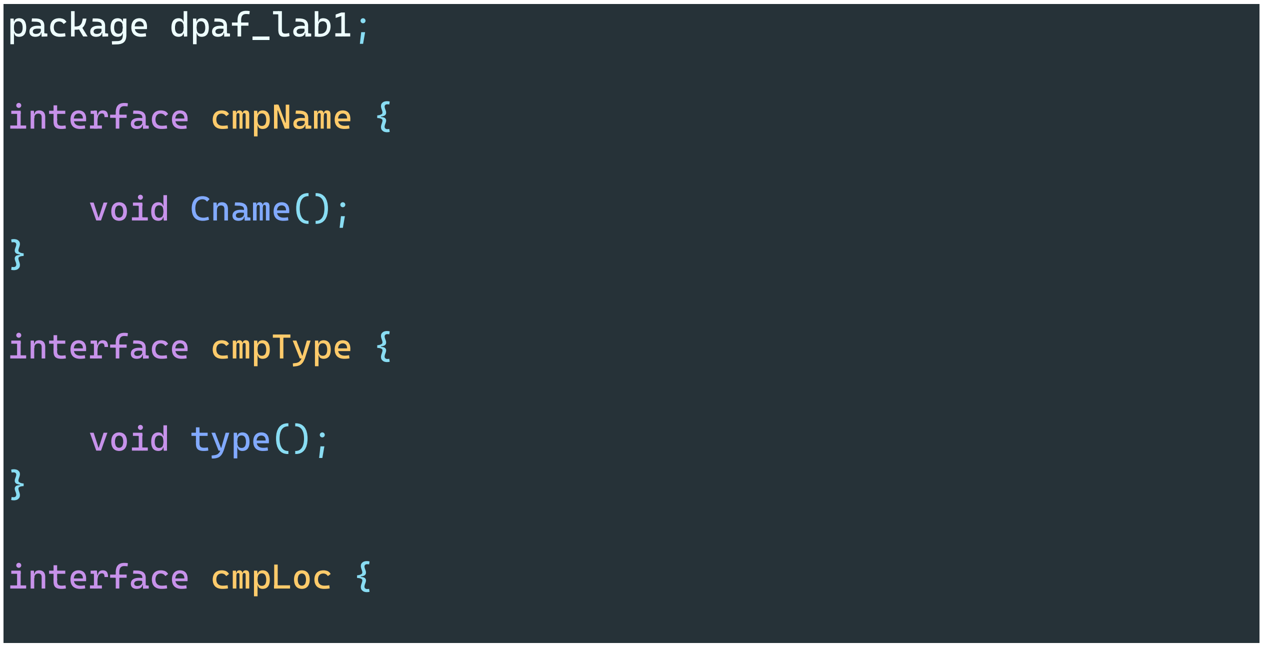
- In this I have used my father Company which is running in surat .

File Structure :-

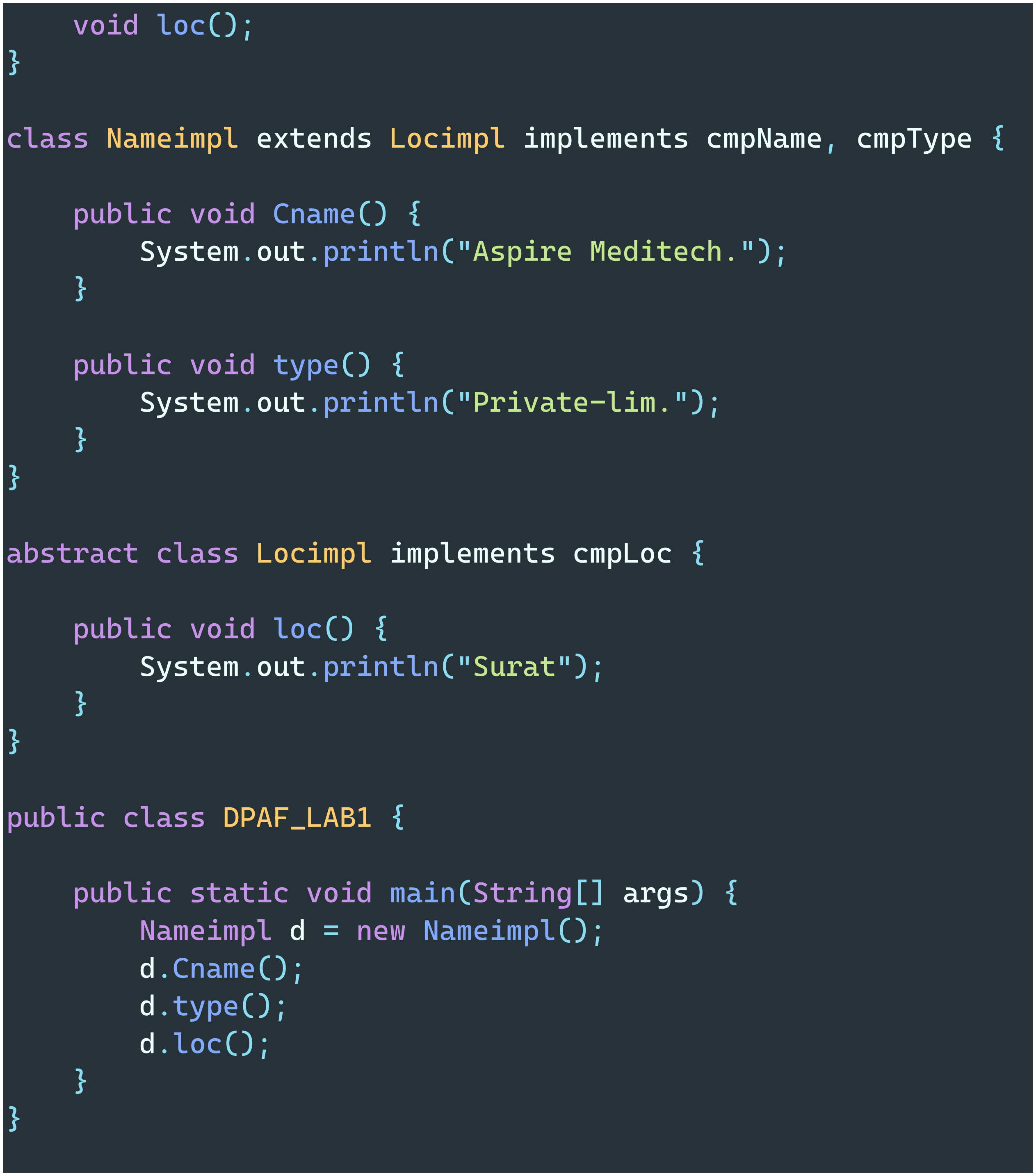


```
▼ DPAF_LAB1
  ▼ Source Packages
    ▼ dpaf_lab1
        DPAF_LAB1.java
  ▶ Libraries
```

Code :-

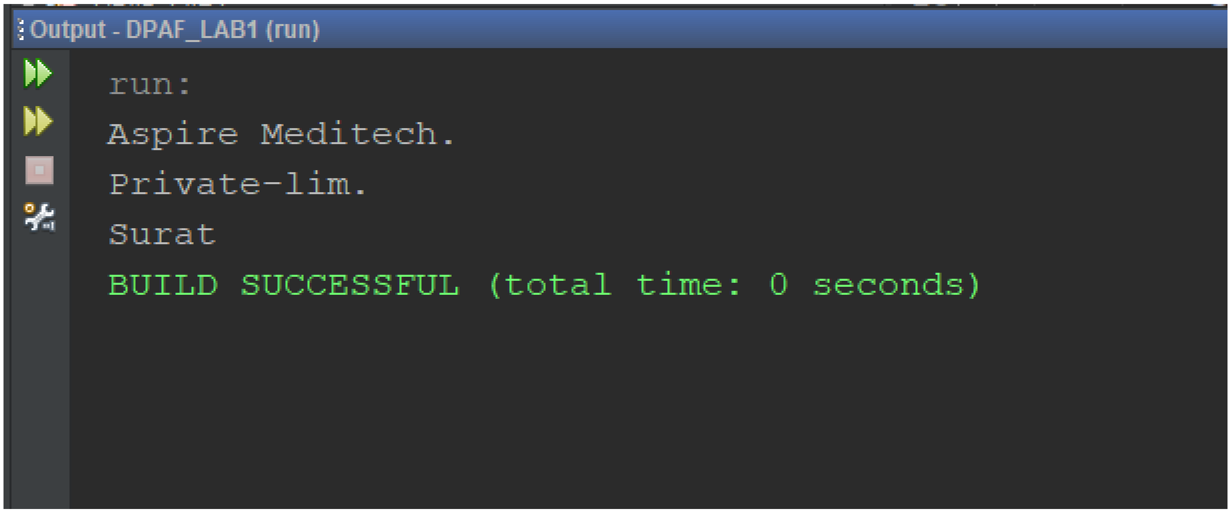


```
package dpaf_lab1;

interface cmpName {

    void Cname();
}

interface cmpType {

    void type();
}

interface cmpLoc {
```



```
    void loc();
}

class Nameimpl extends Locimpl implements cmpName, cmpType {

    public void Cname() {
        System.out.println("Aspire Meditech.");
    }

    public void type() {
        System.out.println("Private-lim.");
    }
}

abstract class Locimpl implements cmpLoc {

    public void loc() {
        System.out.println("Surat");
    }
}

public class DPAF_LAB1 {

    public static void main(String[] args) {
        Nameimpl d = new Nameimpl();
        d.Cname();
        d.type();
        d.loc();
    }
}
```

Output : -



```
run:
Aspire Meditech.
Private-lim.
Surat
BUILD SUCCESSFUL (total time: 0 seconds)
```

Experiment-2

Aim : Implement programming solutions that use following design pattern

a. Singleton

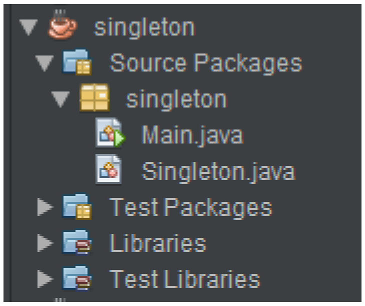
b. AbstractFactory

c. Prototype

a. Singleton :-

- Here I have take very simple and useful example of making database connection one instance at a time .

File Structure :-

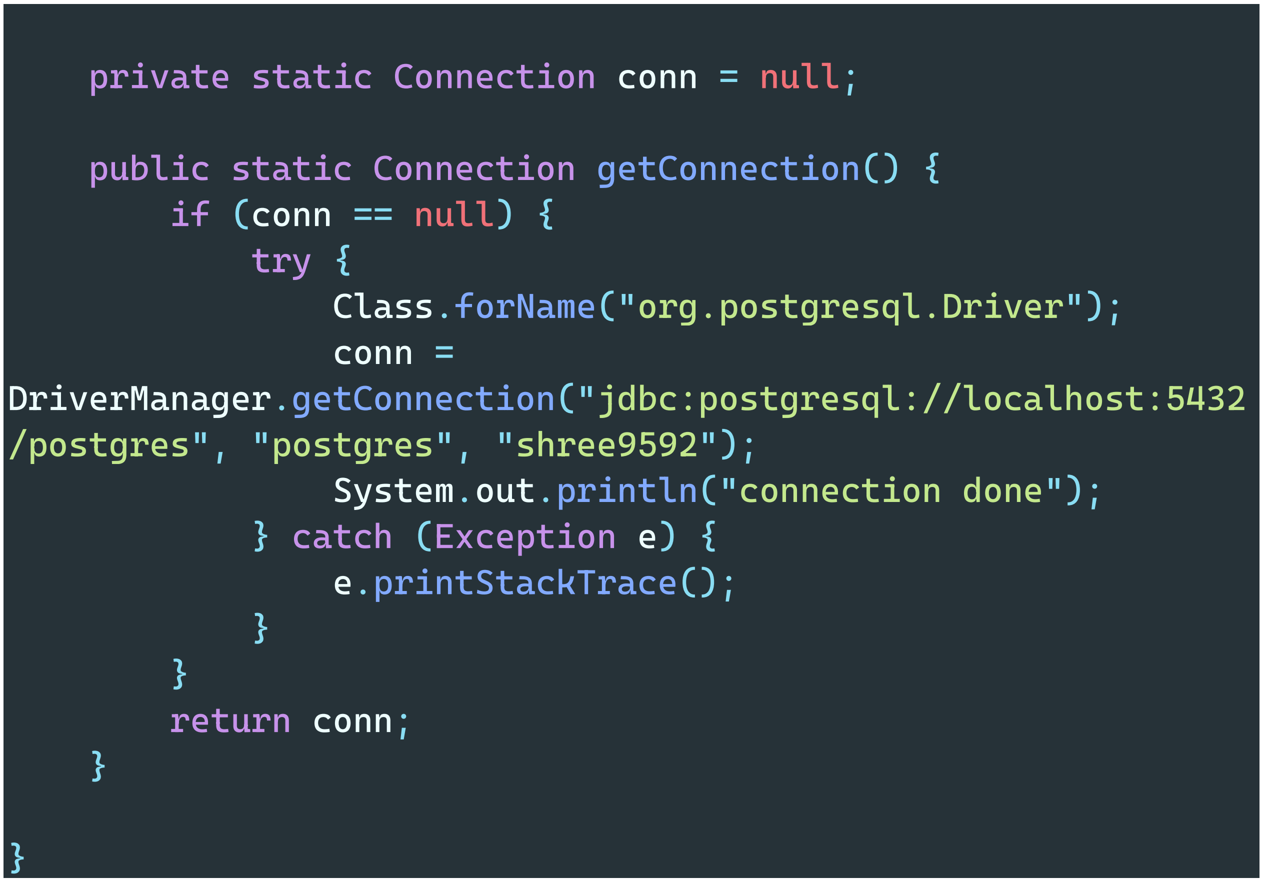


```
▼ singleton
  ▼ Source Packages
    ▼ singleton
        Main.java
        Singleton.java
  ▶ Test Packages
  ▶ Libraries
  ▶ Test Libraries
```

Code :-

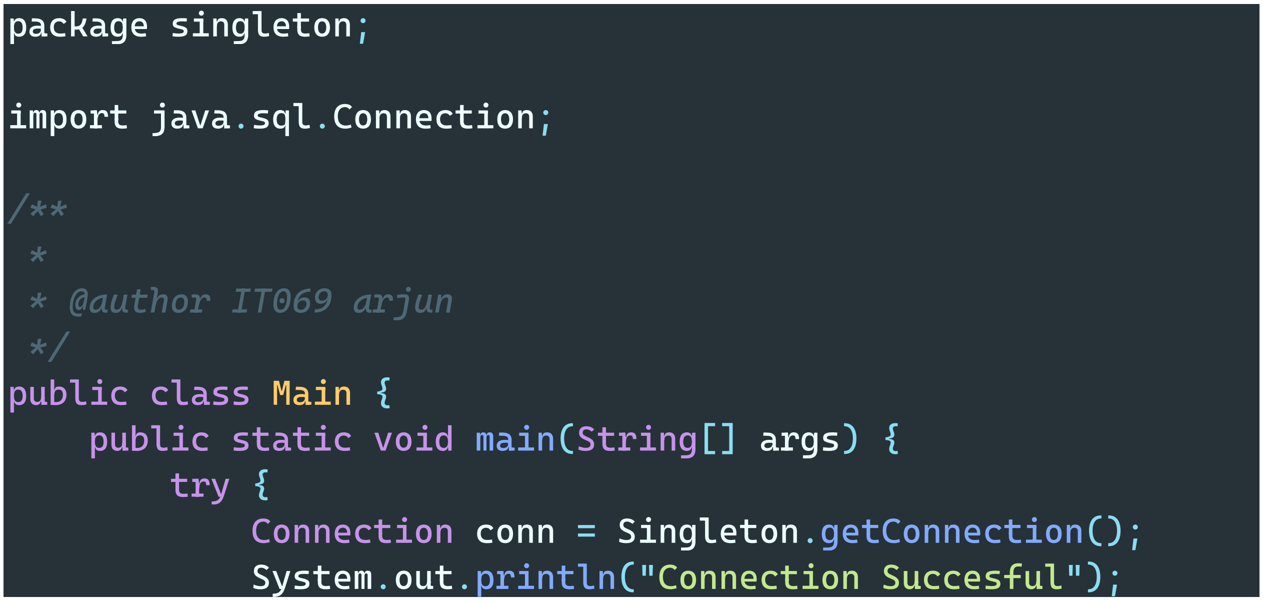
Singleton.java :-

```
package singleton;

import java.sql.Connection;
import java.sql.DriverManager;

public class Singleton {
```

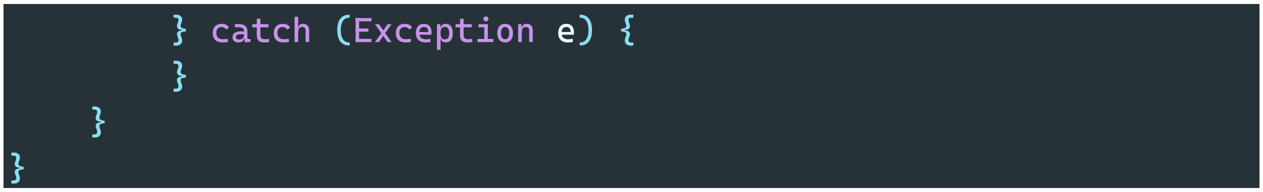


```
    private static Connection conn = null;

    public static Connection getConnection() {
        if (conn == null) {
            try {
                Class.forName("org.postgresql.Driver");
                conn =
DriverManager.getConnection("jdbc:postgresql://localhost:5432
/postgres", "postgres", "shree9592");
                System.out.println("connection done");
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return conn;
    }

}
```

Main :-



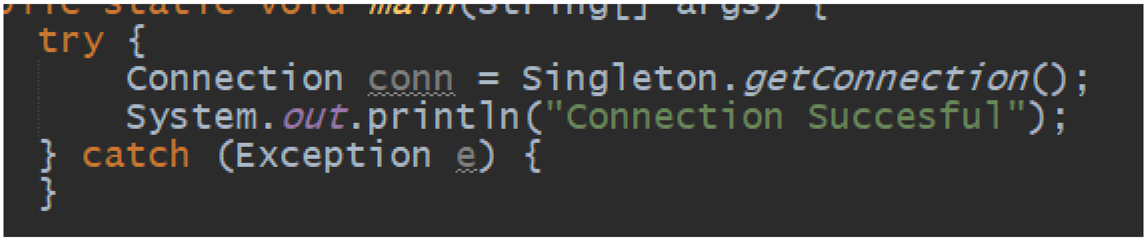
```
package singleton;

import java.sql.Connection;

/**
 *
 * @author IT069 arjun
 */
public class Main {
    public static void main(String[] args) {
        try {
            Connection conn = Singleton.getConnection();
            System.out.println("Connection Succesful");
```



```
        } catch (Exception e) {
        }
    }
}
```

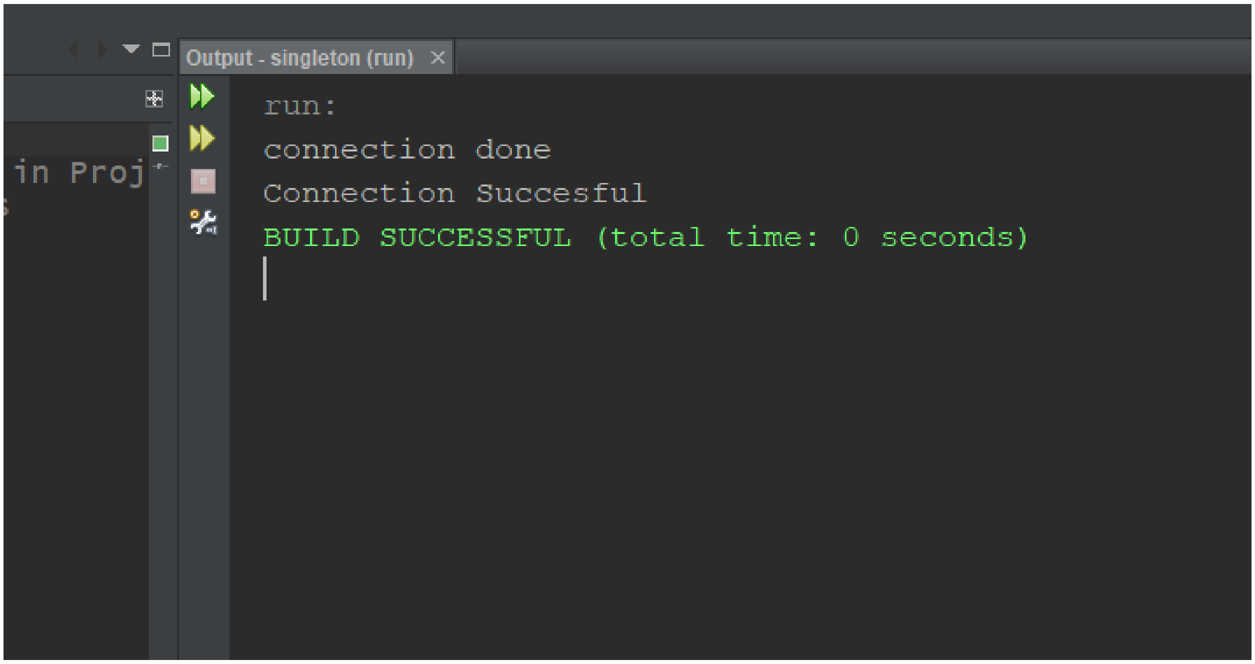
Input/Output :-

Input :-



```
lic static void main(String[] args) {
 try {
     Connection conn = Singleton.getConnection();
     System.out.println("Connection Succesful");
 } catch (Exception e) {
 }
```

Output :-

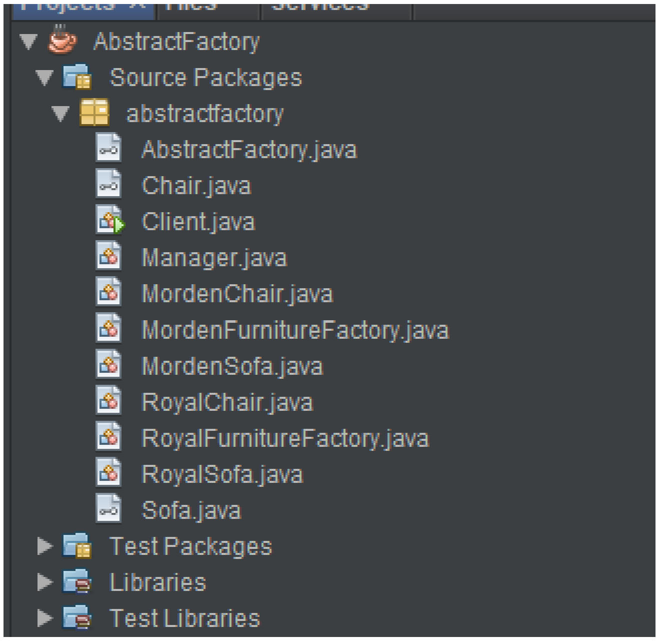


```
run:
connection done
Connection Succesful
BUILD SUCCESSFUL (total time: 0 seconds)
```

b. Abstract Factory :-

- Here I have use furniture shop example .

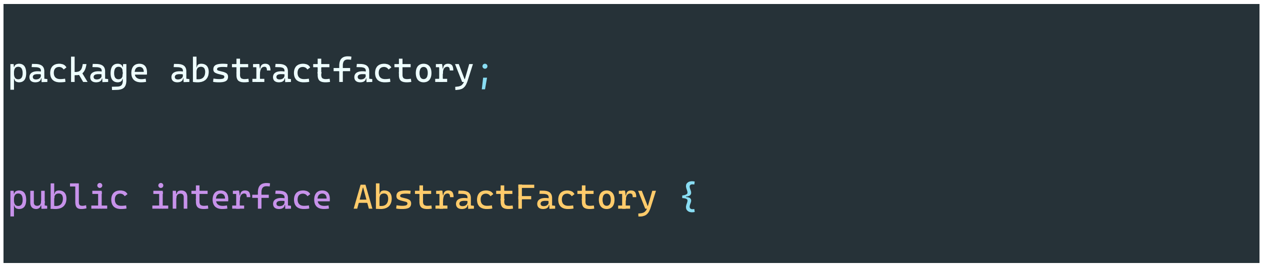
File Structure :-



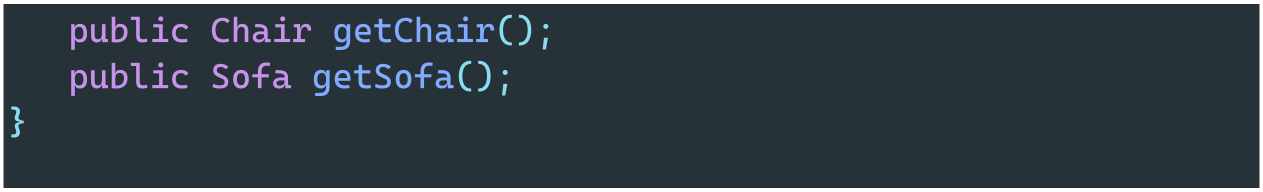
```
▼ AbstractFactory
  ▼ Source Packages
    ▼ abstractfactory
        AbstractFactory.java
        Chair.java
        Client.java
        Manager.java
        MordenChair.java
        MordenFurnitureFactory.java
        MordenSofa.java
        RoyalChair.java
        RoyalFurnitureFactory.java
        RoyalSofa.java
        Sofa.java
  ▶ Test Packages
  ▶ Libraries
  ▶ Test Libraries
```

Code :-

AbstractFactory.java :-

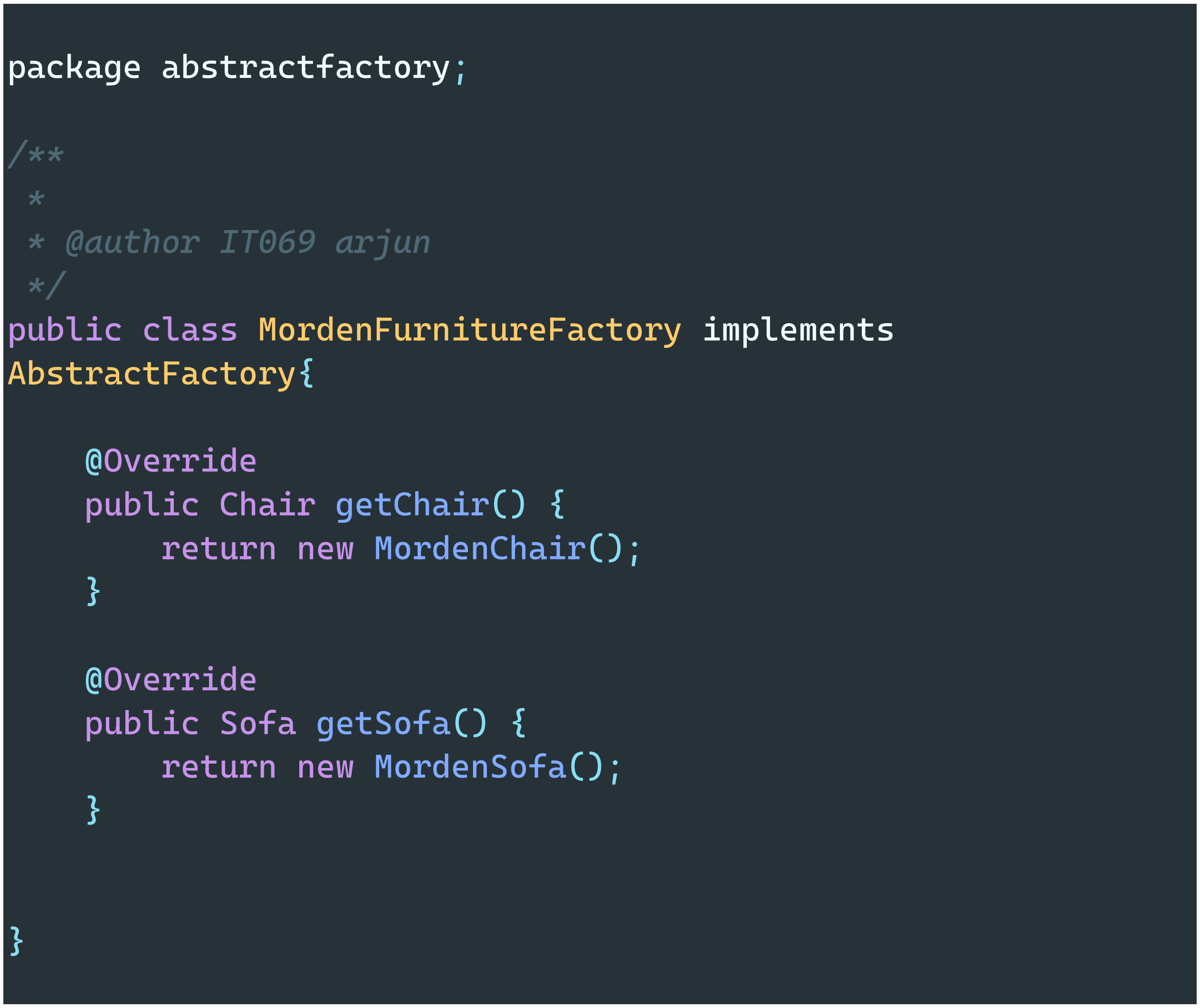


```java
package abstractfactory;

public interface AbstractFactory {
```



```
    public Chair getChair();
    public Sofa getSofa();
}
```

MordenFurnitureFactory.java :-



```
package abstractfactory;

/**
 *
 * @author IT069 arjun
 */
public class MordenFurnitureFactory implements
AbstractFactory{

    @Override
    public Chair getChair() {
        return new MordenChair();
    }

    @Override
    public Sofa getSofa() {
        return new MordenSofa();
    }


}
```

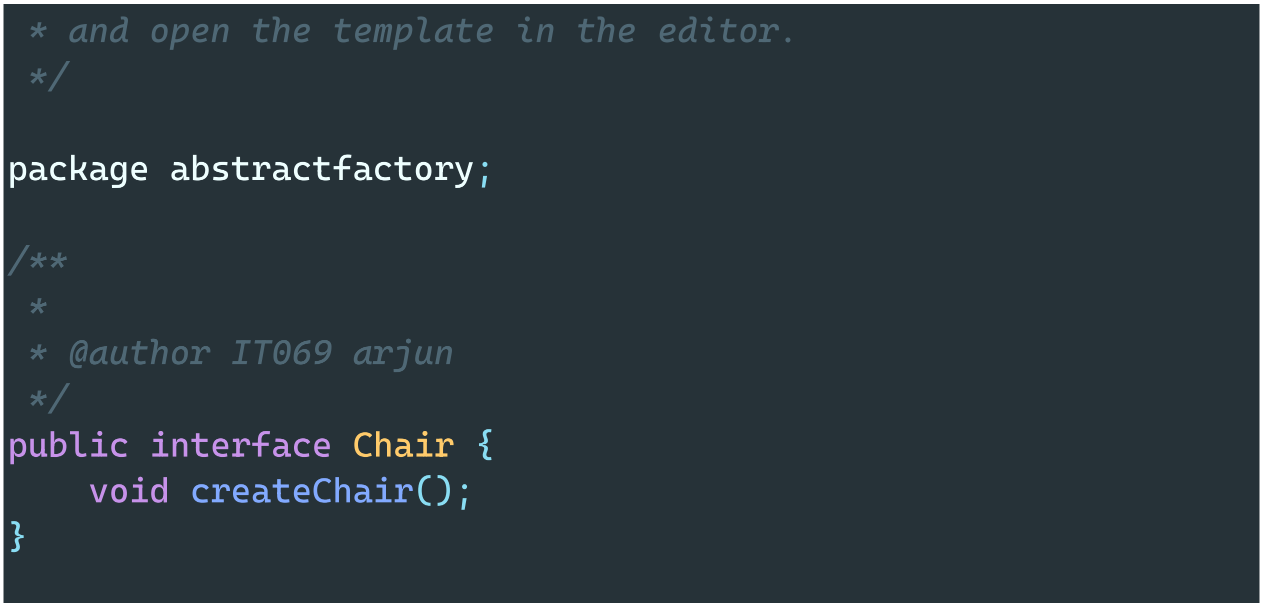
RoyalFurnitureFactory.java :-

```
/*
```

```
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package abstractfactory;

/**
 *
 * @author IT069 arjun
 */
public class RoyalFurnitureFactory implements
AbstractFactory{

    @Override
    public Chair getChair() {
        return new RoyalChair();
    }

    @Override
    public Sofa getSofa() {
        return new RoyalSofa();
    }



}
```

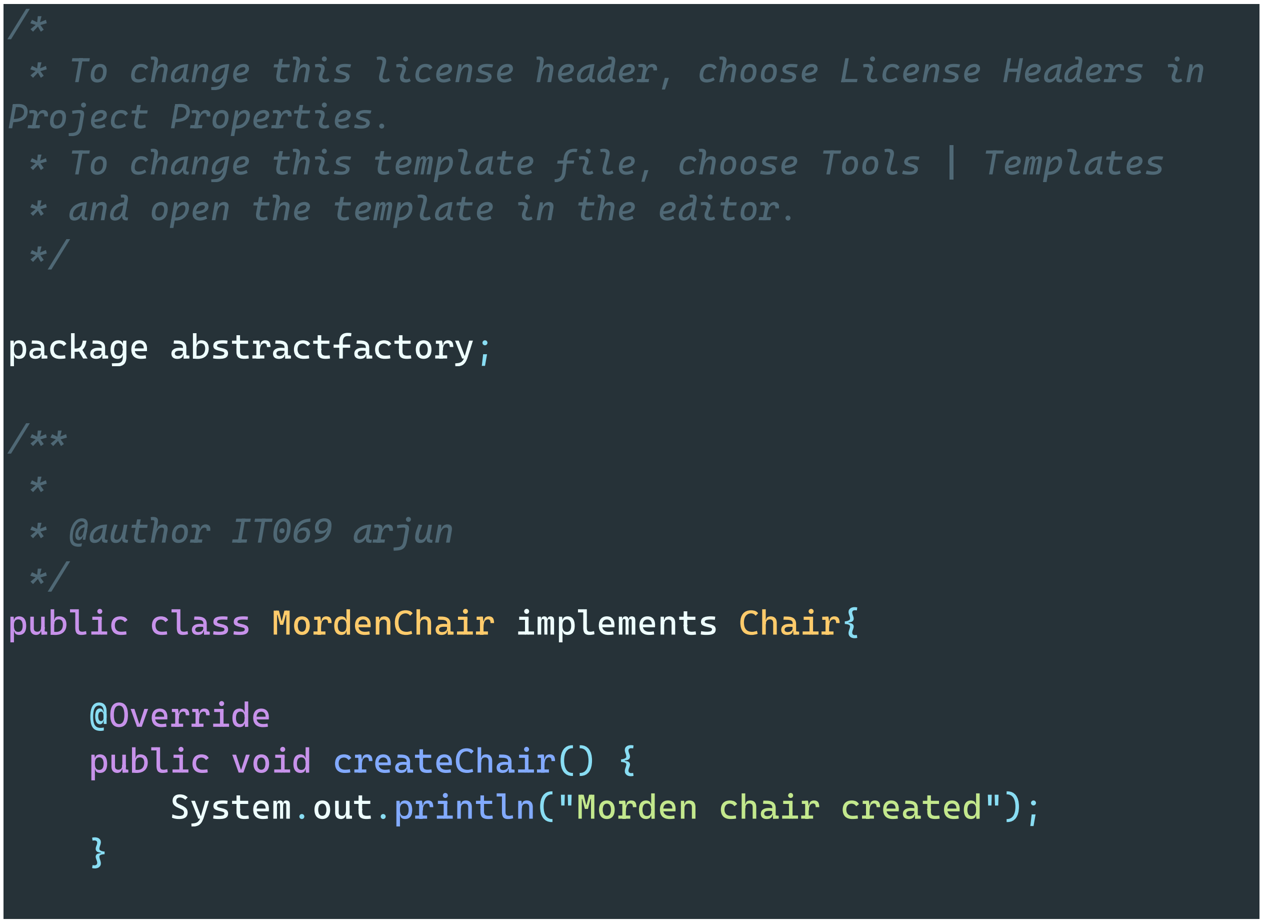
Chair.java :-

```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
```



```
 * and open the template in the editor.
 */

package abstractfactory;

/**
 *
 * @author IT069 arjun
 */
public interface Chair {
    void createChair();
}
```

MordenChair.java :-



```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package abstractfactory;

/**
 *
 * @author IT069 arjun
 */
public class MordenChair implements Chair{

    @Override
    public void createChair() {
        System.out.println("Morden chair created");
    }
}
```

RoyalChair.java :-

```
package abstractfactory;

/**
 *
 * @author IT069 arjun
 */
public class RoyalChair implements Chair{

    @Override
    public void createChair() {
        System.out.println("Royal chair has created");
    }

}
```

Sofa.java :-

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package abstractfactory;

/**
 *
 * @author admin
 */
```

```
public interface Sofa {
    void createSofa();
}
```

MordenSofa.java :-

```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package abstractfactory;

/**
 *
 * @author IT069 arjun
 */
public class MordenSofa implements Sofa{

    @Override
    public void createSofa() {
        System.out.println("Mordern sofa created");
    }

}
```

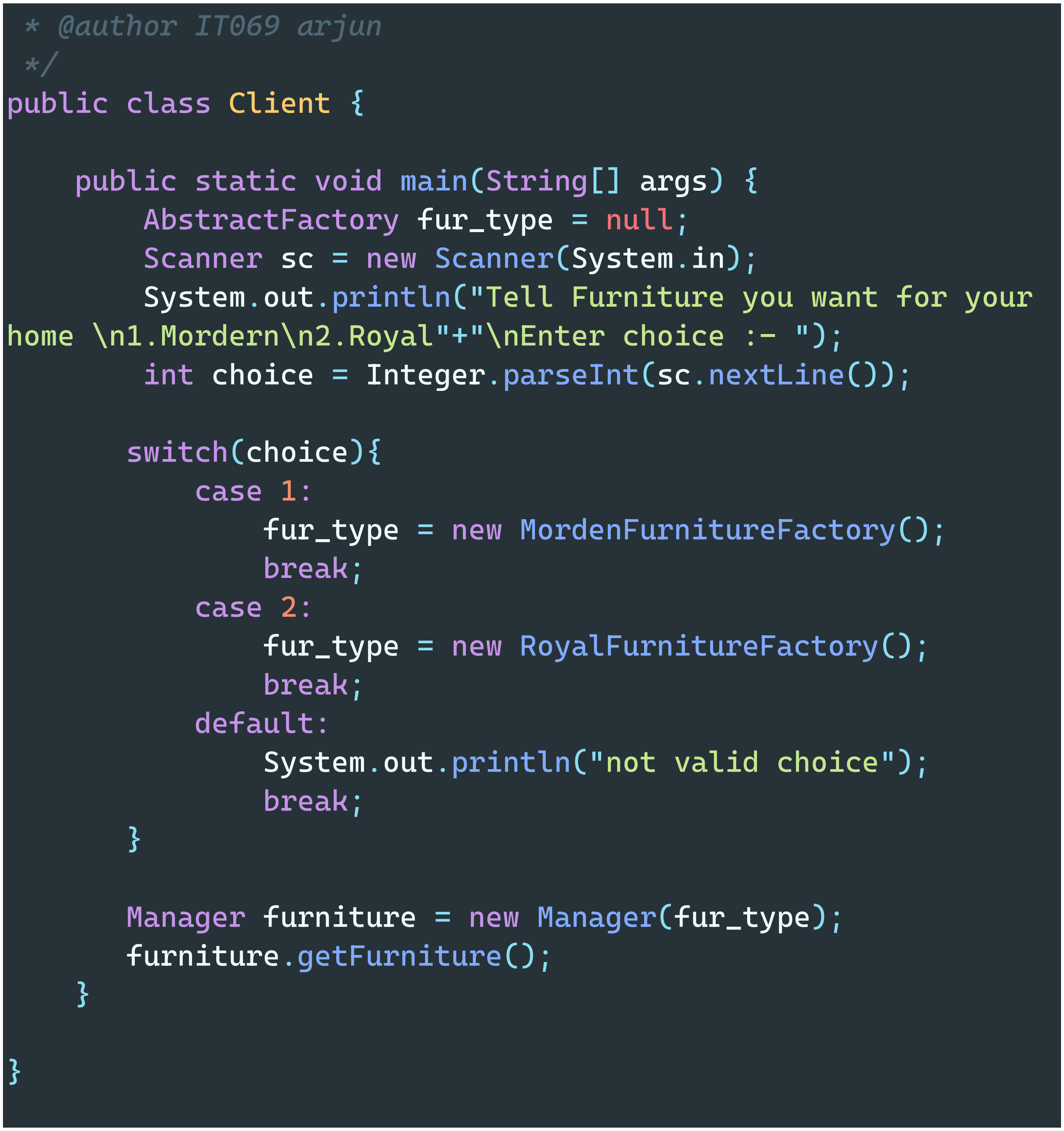
RoyalSofa.java :-

```
/*
```

```
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package abstractfactory;

/**
 *
 * @author IT069 arjun
 */
public class RoyalSofa implements Sofa{

    @Override
    public void createSofa() {
        System.out.println("Royal sofa created ");
    }

}
```

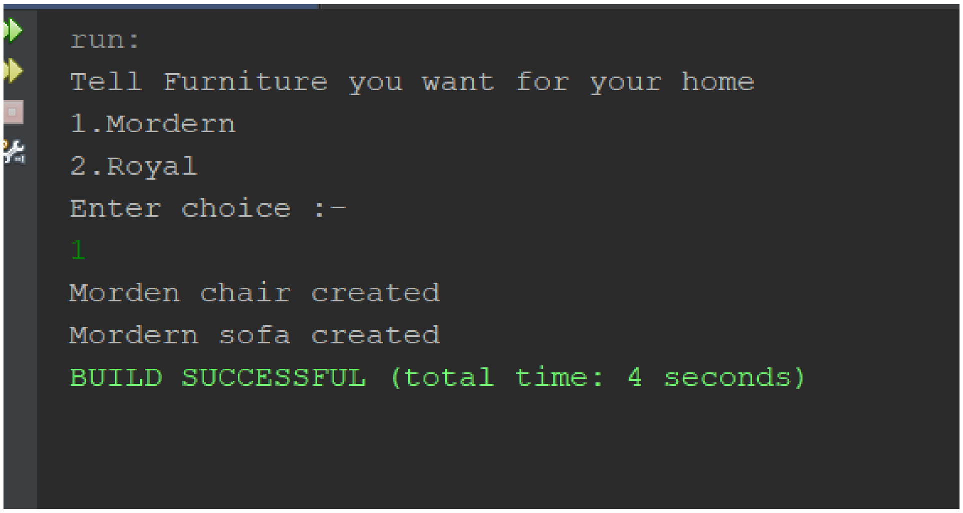
Client.java:-

```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package abstractfactory;

import java.util.Scanner;

/**
 *
```

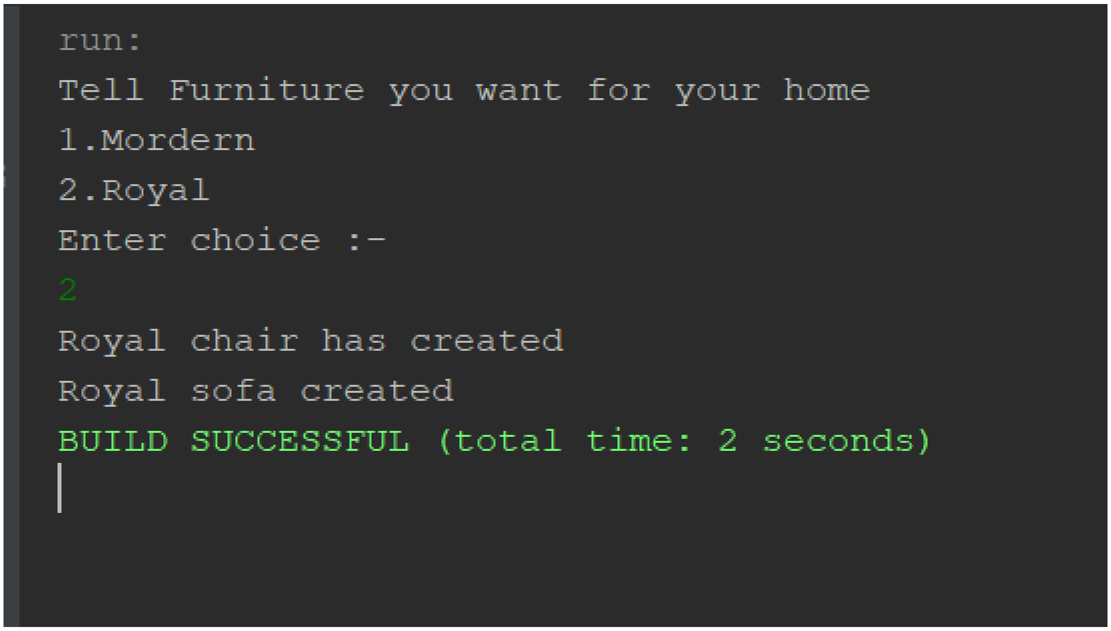


```
 * @author IT069 arjun
 */
public class Client {

    public static void main(String[] args) {
        AbstractFactory fur_type = null;
        Scanner sc = new Scanner(System.in);
        System.out.println("Tell Furniture you want for your
home \n1.Mordern\n2.Royal"+"\nEnter choice :- ");
        int choice = Integer.parseInt(sc.nextLine());

       switch(choice){
           case 1:
               fur_type = new MordenFurnitureFactory();
               break;
           case 2:
               fur_type = new RoyalFurnitureFactory();
               break;
           default:
               System.out.println("not valid choice");
               break;
       }

       Manager furniture = new Manager(fur_type);
       furniture.getFurniture();
    }

}
```

Input/Output :-



```
run:
Tell Furniture you want for your home
1.Mordern
2.Royal
Enter choice :-
1
Morden chair created
Mordern sofa created
BUILD SUCCESSFUL (total time: 4 seconds)
```

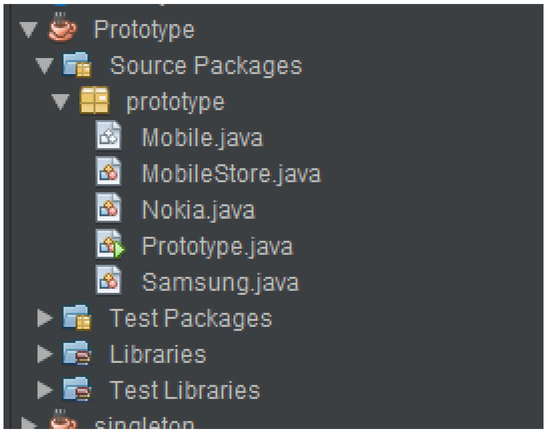


```
run:
Tell Furniture you want for your home
1.Mordern
2.Royal
Enter choice :-
2
Royal chair has created
Royal sofa created
BUILD SUCCESSFUL (total time: 2 seconds)
```

c. Prototype :-

- Here I have use mobile shop example;

File Structure :-

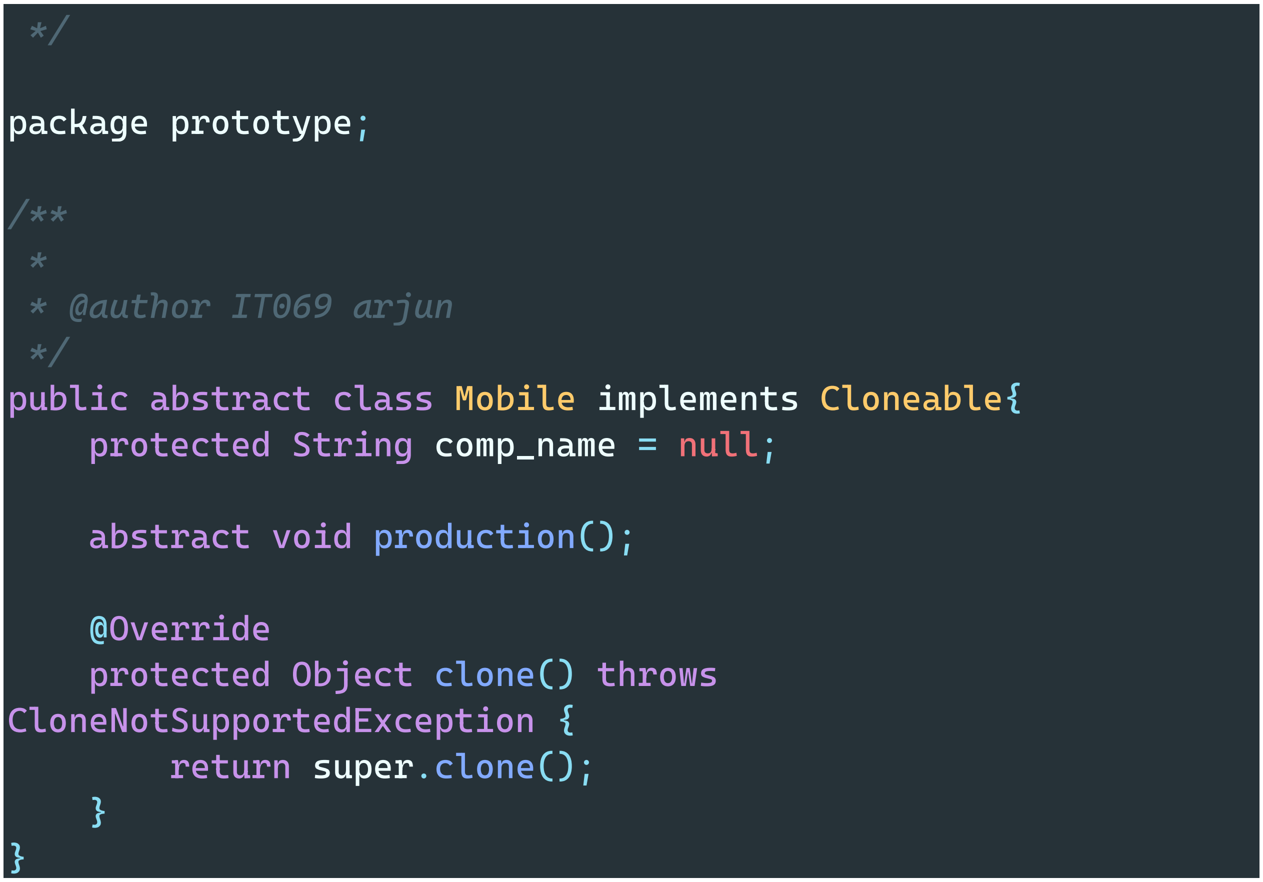


```
▼ Prototype
  ▼ Source Packages
    ▼ prototype
        Mobile.java
        MobileStore.java
        Nokia.java
        Prototype.java
        Samsung.java
  ▶ Test Packages
  ▶ Libraries
  ▶ Test Libraries
▶ singleton
```

Code :-

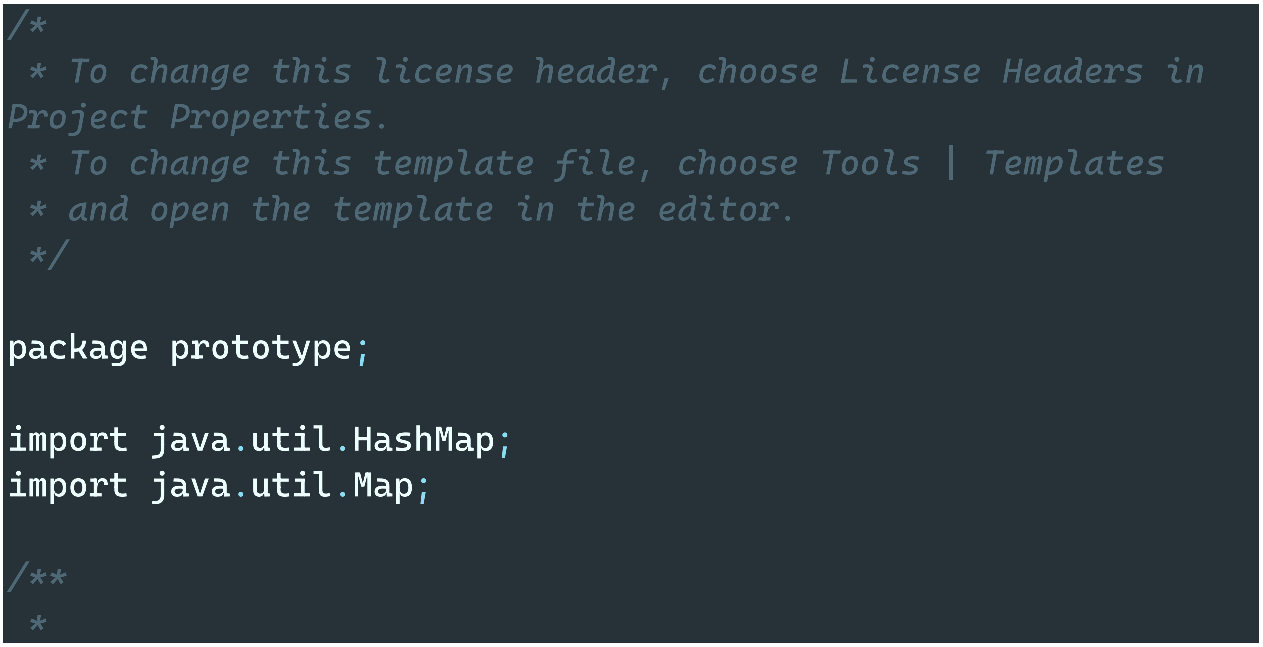
Mobile.java :-

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
```

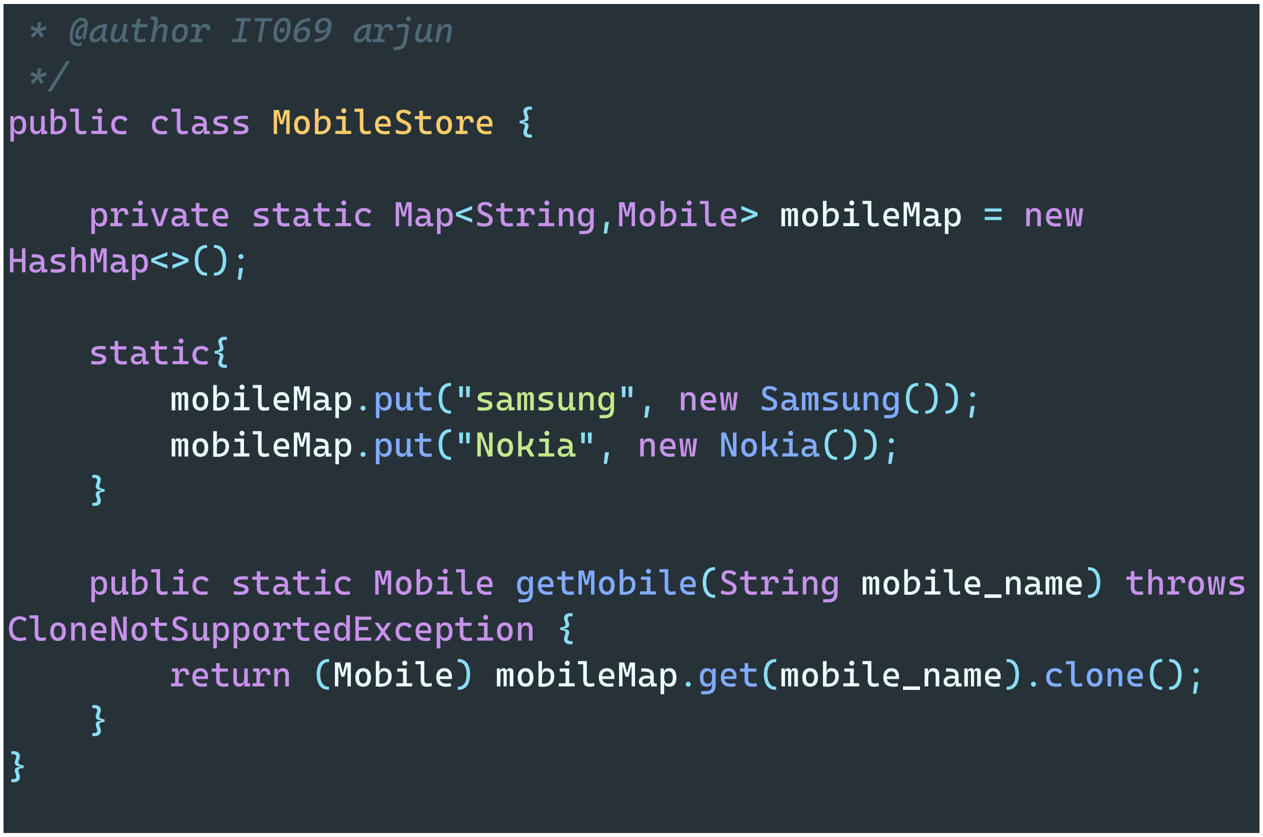


```
 */

package prototype;

/**
 *
 * @author IT069 arjun
 */
public abstract class Mobile implements Cloneable{
    protected String comp_name = null;

    abstract void production();

    @Override
    protected Object clone() throws
CloneNotSupportedException {
        return super.clone();
    }
}
```

MobileStore.java :-

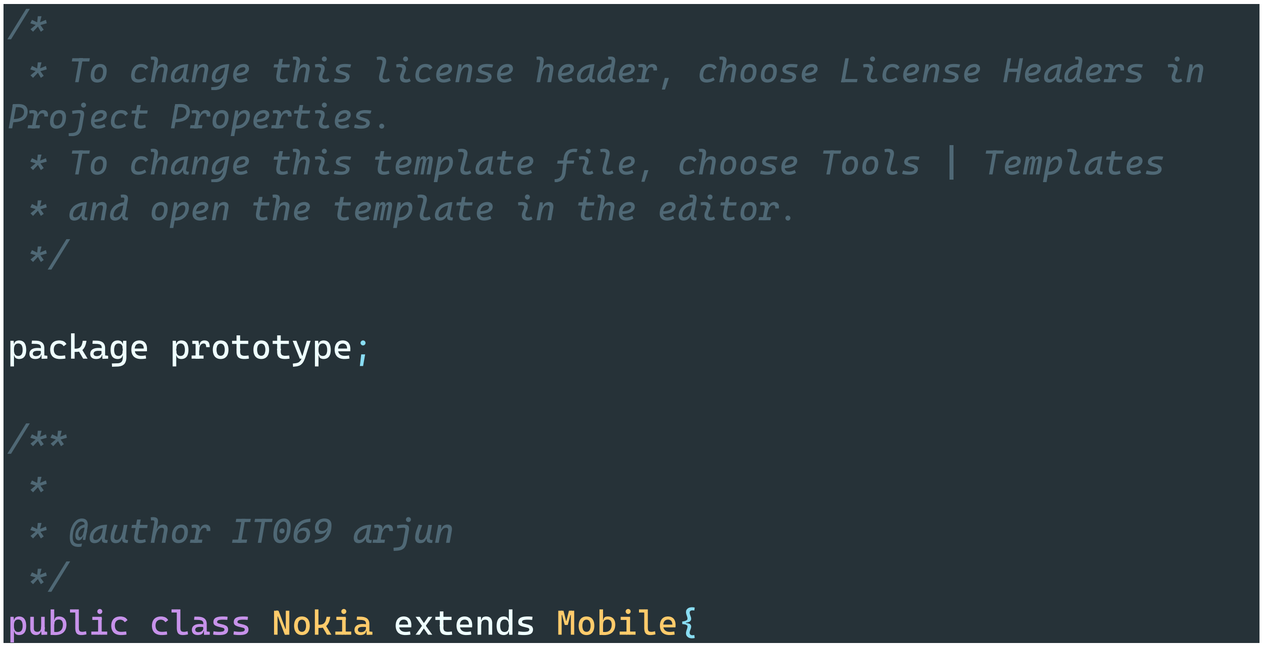


```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package prototype;

import java.util.HashMap;
import java.util.Map;

/**
 *
```

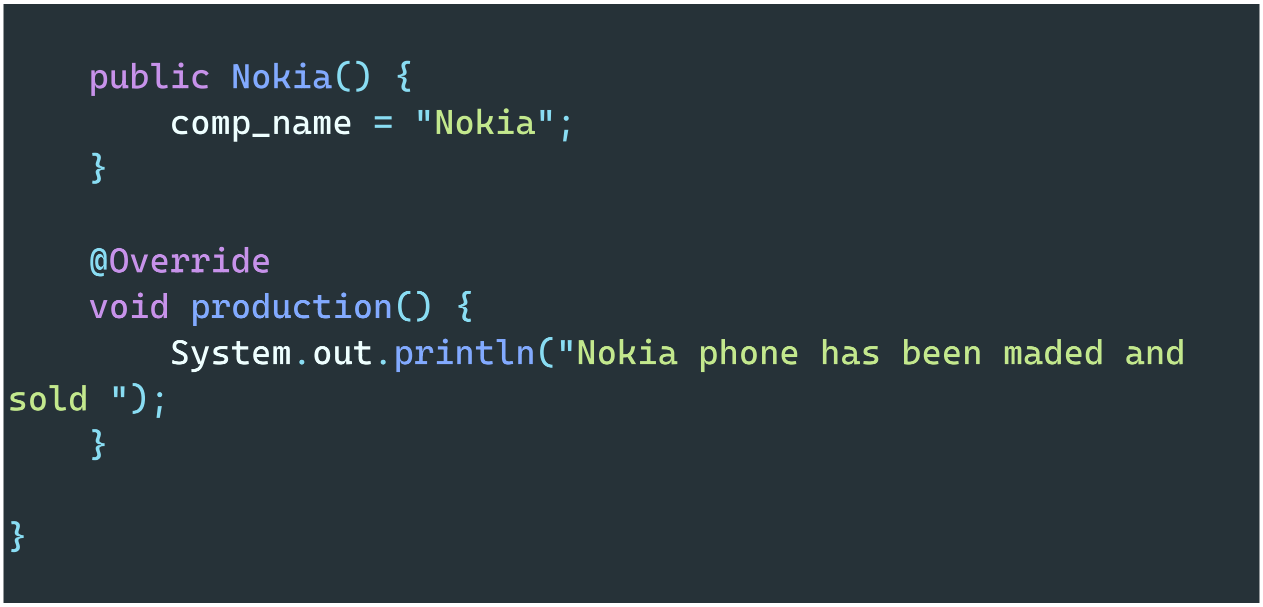


```
 * @author IT069 arjun
 */
public class MobileStore {

    private static Map<String,Mobile> mobileMap = new
HashMap<>();

    static{
        mobileMap.put("samsung", new Samsung());
        mobileMap.put("Nokia", new Nokia());
    }

    public static Mobile getMobile(String mobile_name) throws
CloneNotSupportedException {
        return (Mobile) mobileMap.get(mobile_name).clone();
    }
}
```

Nokia.java :-

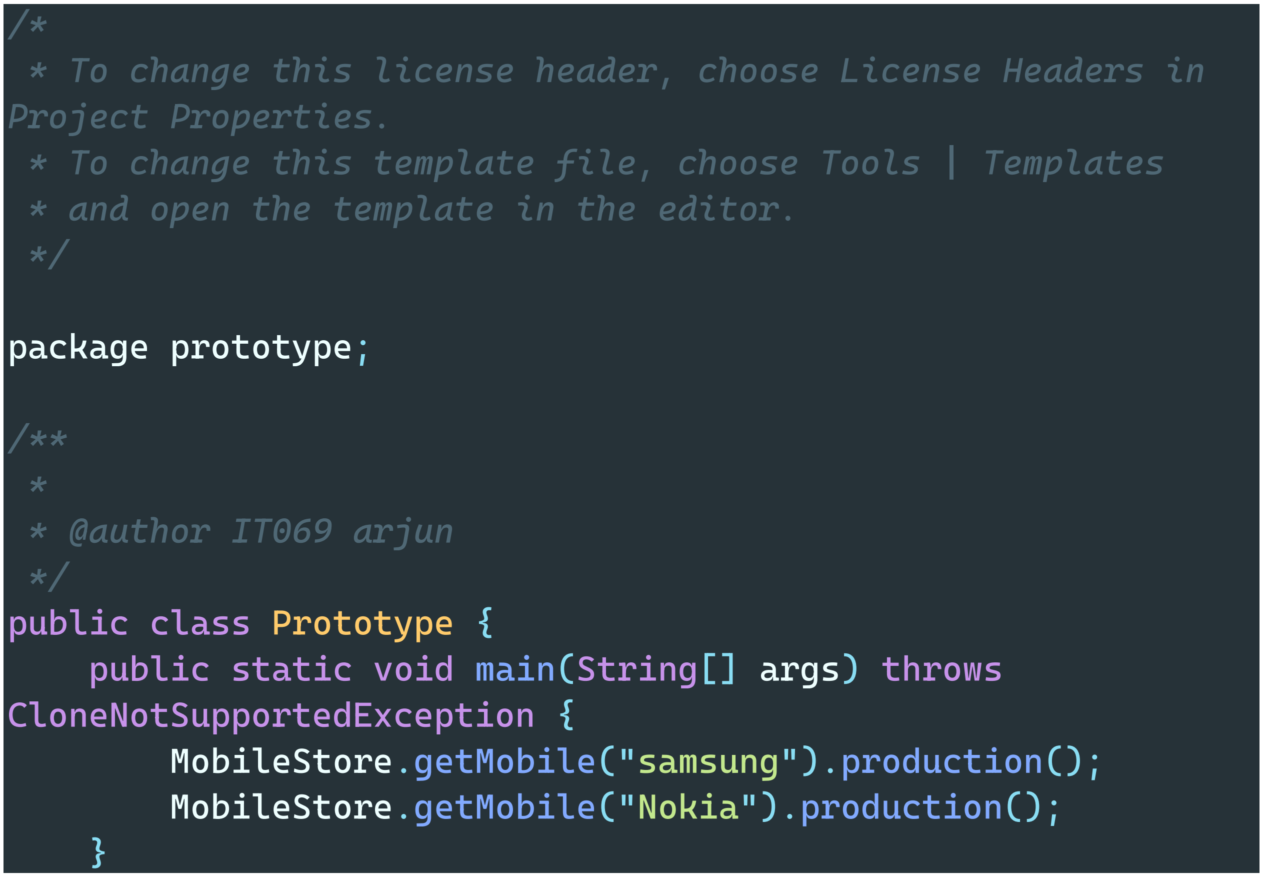


```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package prototype;

/**
 *
 * @author IT069 arjun
 */
public class Nokia extends Mobile{
```



```
    public Nokia() {
        comp_name = "Nokia";
    }

    @Override
    void production() {
        System.out.println("Nokia phone has been maded and 
sold ");
    }

}
```

Prototype.java :-



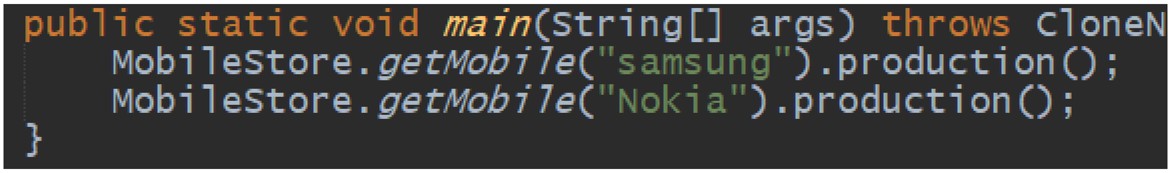
```
/*
 * To change this license header, choose License Headers in 
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package prototype;

/**
 *
 * @author IT069 arjun
 */
public class Prototype {
    public static void main(String[] args) throws 
CloneNotSupportedException {
        MobileStore.getMobile("samsung").production();
        MobileStore.getMobile("Nokia").production();
    }
}
```

Samsung.java :-

```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package prototype;

/**
 *
 * @author IT069 arjun
 */
public class Samsung extends Mobile {

    public Samsung() {
        comp_name = "samsung";
    }

    @Override
    void production() {
        System.out.println("samsung phone has maded and
sold");
    }

}
```

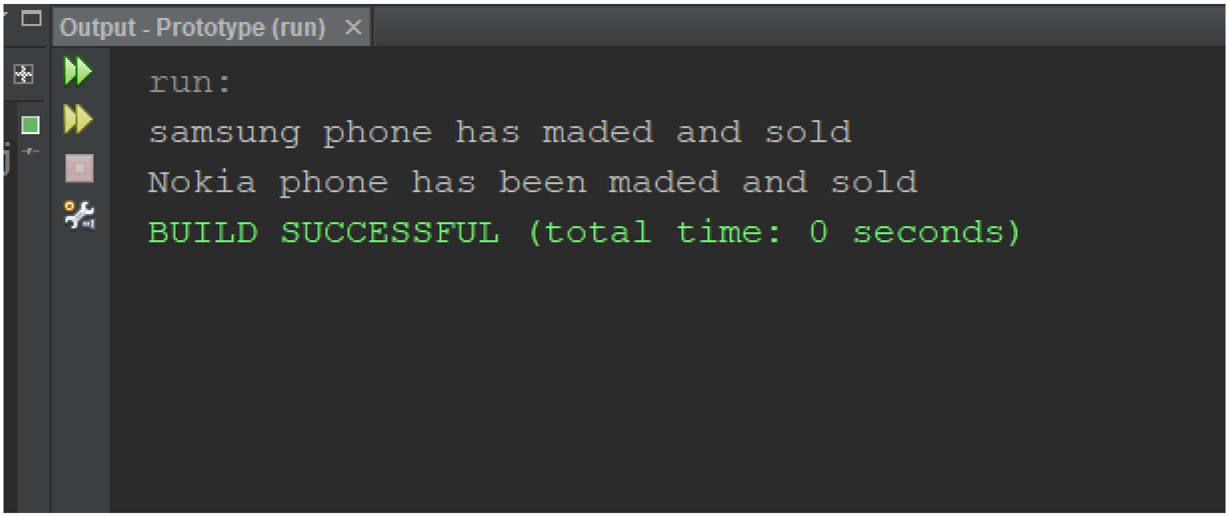
Input / Output :-

Input :-



```
public static void main(String[] args) throws CloneN
    MobileStore.getMobile("samsung").production();
    MobileStore.getMobile("Nokia").production();
}
```

Output :-



```
run:
samsung phone has maded and sold
Nokia phone has been maded and sold
BUILD SUCCESSFUL (total time: 0 seconds)
```

Experiment-3

Aim : Implement programming solutions that use following design pattern

a. Composite

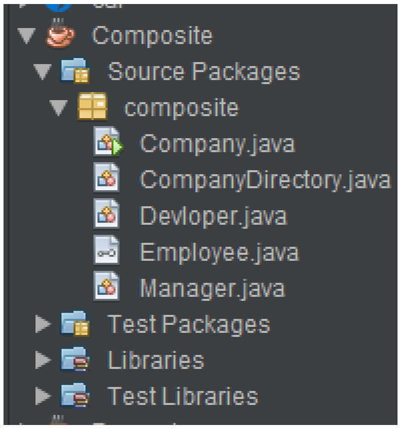
b. Decorator

c. Façade

a. Composite :-

- Here I have take one Company Hierarchal tree ;

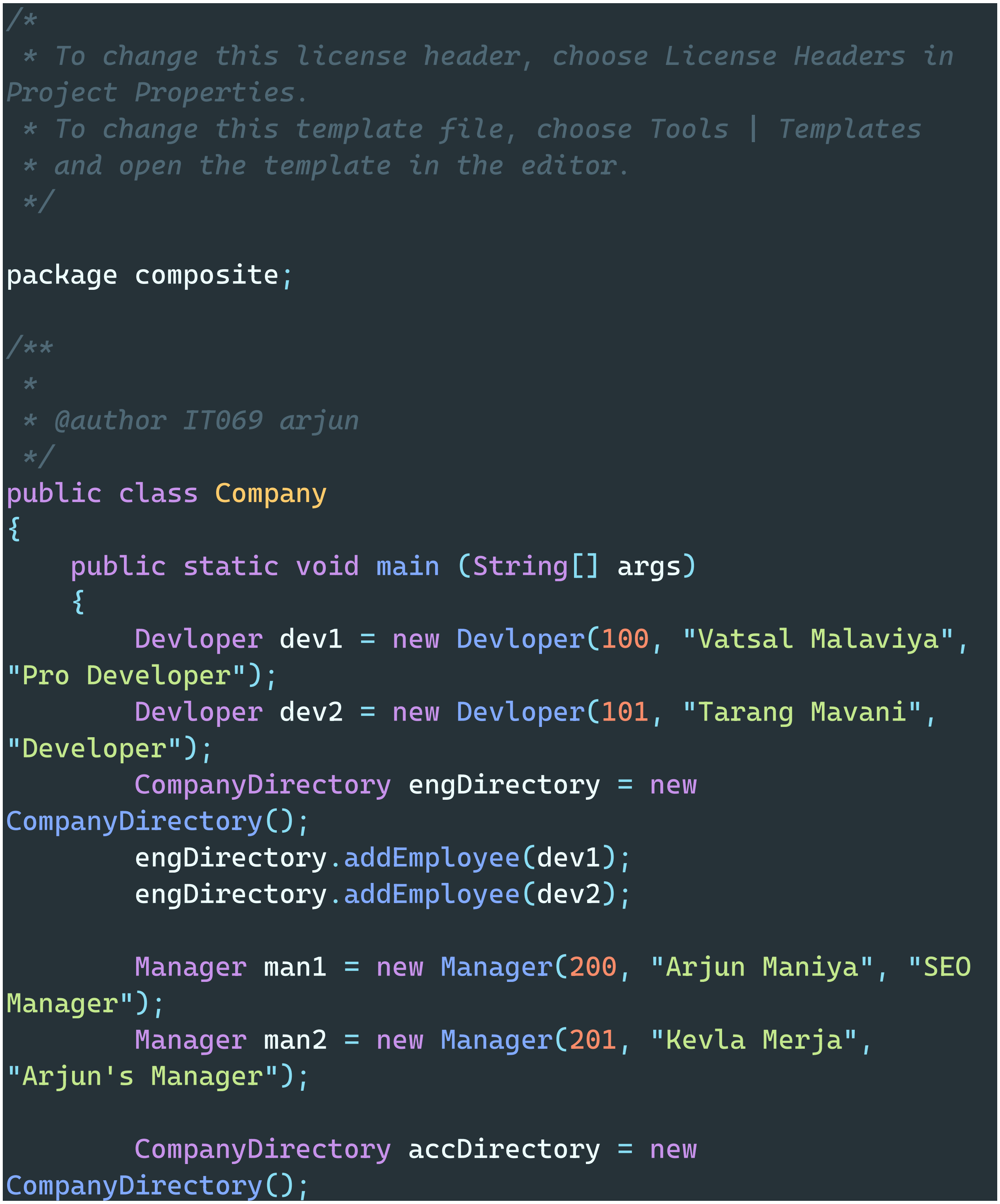
File Structure :-



```
▼ Composite
  ▼ Source Packages
    ▼ composite
        Company.java
        CompanyDirectory.java
        Devloper.java
        Employee.java
        Manager.java
  ▶ Test Packages
  ▶ Libraries
  ▶ Test Libraries
```

Code :-

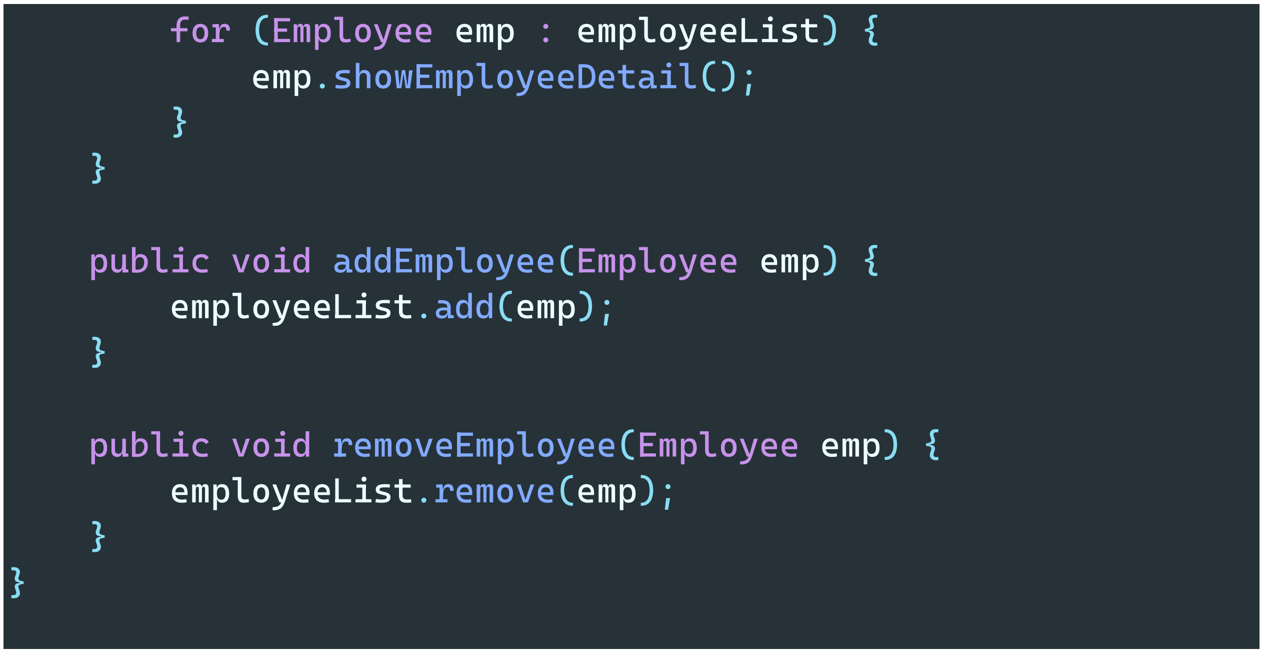
Company.java :-



```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package composite;

/**
 *
 * @author IT069 arjun
 */
public class Company
{
    public static void main (String[] args)
    {
        Devloper dev1 = new Devloper(100, "Vatsal Malaviya", "Pro Developer");
        Devloper dev2 = new Devloper(101, "Tarang Mavani", "Developer");
        CompanyDirectory engDirectory = new CompanyDirectory();
        engDirectory.addEmployee(dev1);
        engDirectory.addEmployee(dev2);

        Manager man1 = new Manager(200, "Arjun Maniya", "SEO Manager");
        Manager man2 = new Manager(201, "Kevla Merja", "Arjun's Manager");

        CompanyDirectory accDirectory = new CompanyDirectory();
        accDirectory.addEmployee(man1);
        accDirectory.addEmployee(man2);

        CompanyDirectory directory = new CompanyDirectory();
        directory.addEmployee(engDirectory);
        directory.addEmployee(accDirectory);
        directory.showEmployeeDetail();
    }
}
```

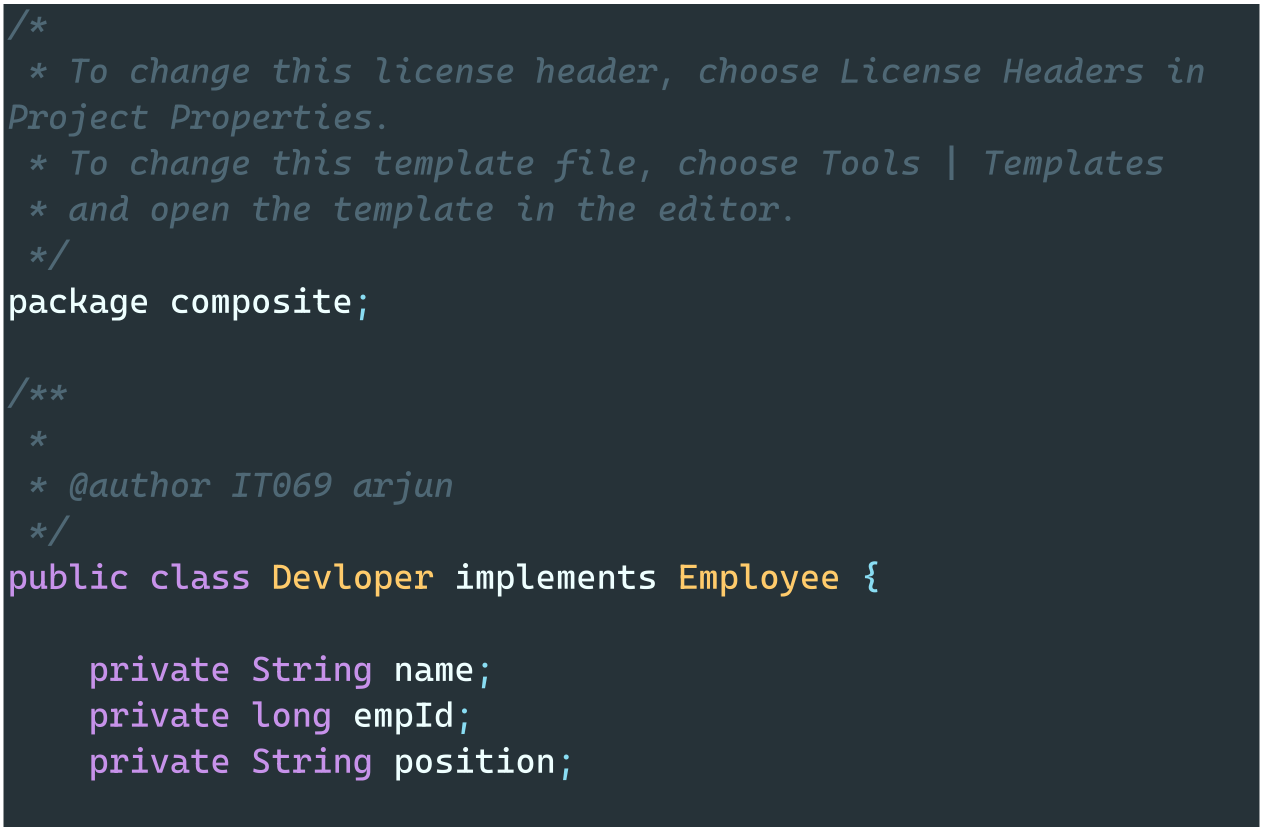
CompanyDirectory.java :

```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package composite;

/**
 *
 * @author IT069 arjun
 */
import java.util.ArrayList;
import java.util.List;

public class CompanyDirectory implements Employee {

    private List<Employee> employeeList = new
ArrayList<Employee>();

    @Override
    public void showEmployeeDetail() {
```

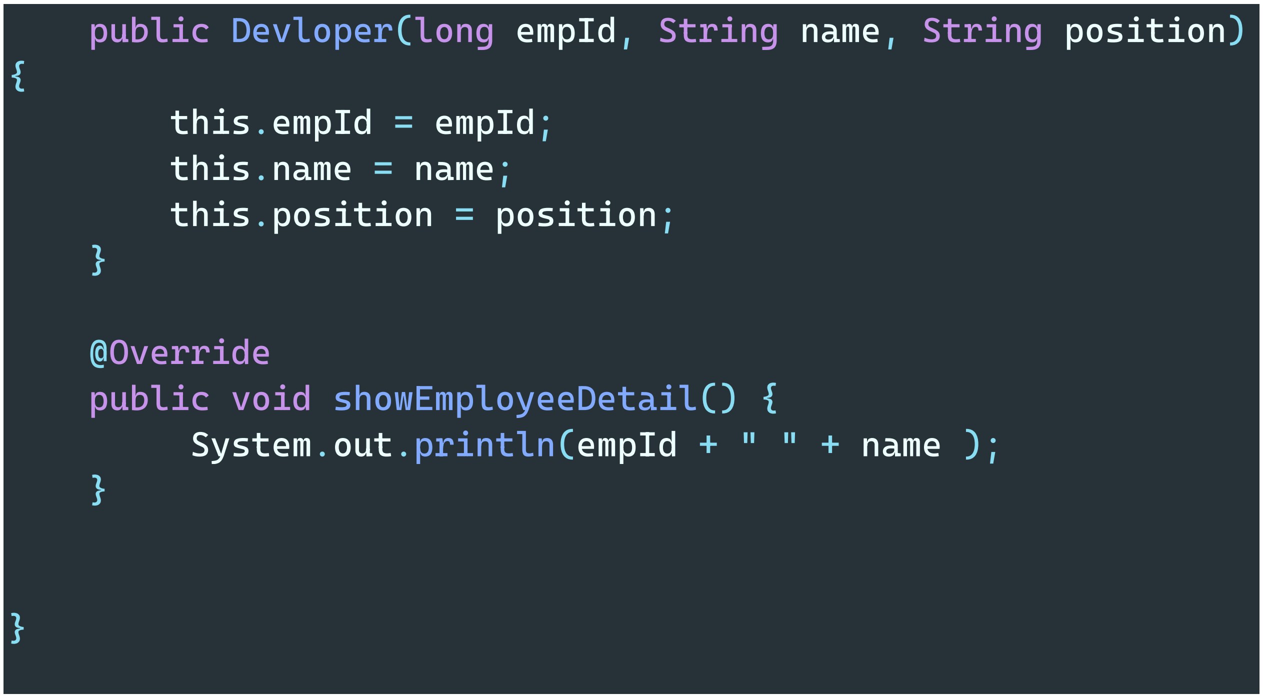


```
        for (Employee emp : employeeList) {
            emp.showEmployeeDetail();
        }
    }

    public void addEmployee(Employee emp) {
        employeeList.add(emp);
    }

    public void removeEmployee(Employee emp) {
        employeeList.remove(emp);
    }
}
```

Devplore.java :-

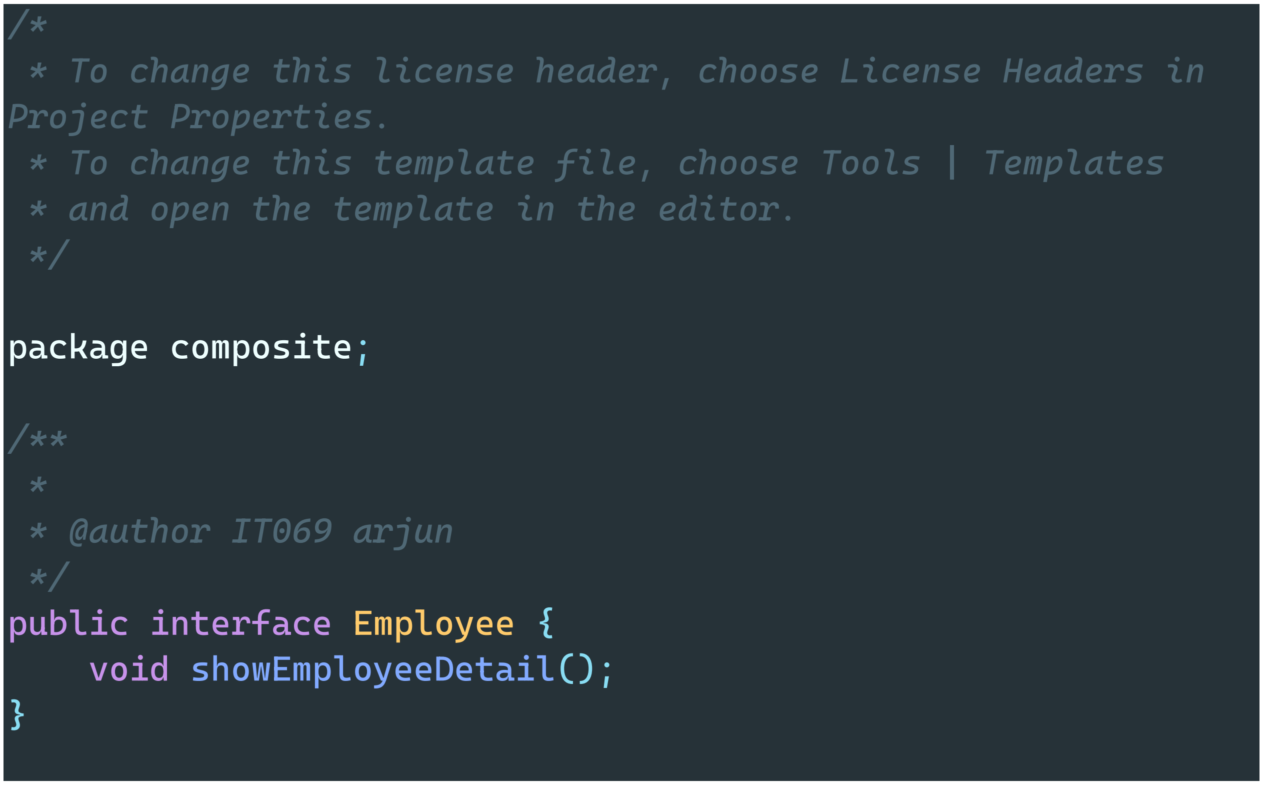


```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package composite;

/**
 *
 * @author IT069 arjun
 */
public class Devloper implements Employee {

    private String name;
    private long empId;
    private String position;
```



```
    public Devloper(long empId, String name, String position)
{
        this.empId = empId;
        this.name = name;
        this.position = position;
    }

    @Override
    public void showEmployeeDetail() {
         System.out.println(empId + " " + name );
    }


}
```

Employee :-

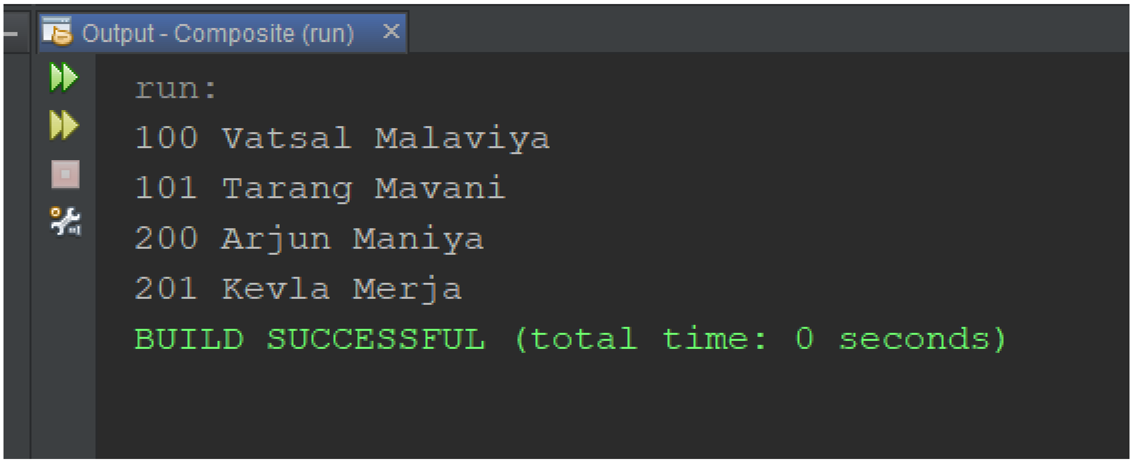


```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package composite;

/**
 *
 * @author IT069 arjun
 */
public interface Employee {
    void showEmployeeDetail();
}
```

Manager.java :-

```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package composite;

/**
 *
 * @author IT069 arjun
 */
public class Manager implements Employee {

    private String name;
    private long empId;
    private String position;

    public Manager(long empId, String name, String position)
{
        this.empId = empId;
        this.name = name;
        this.position = position;
    }

    @Override
    public void showEmployeeDetail() {
        System.out.println(empId + " " + name);
    }
}
```

Input/Output :-

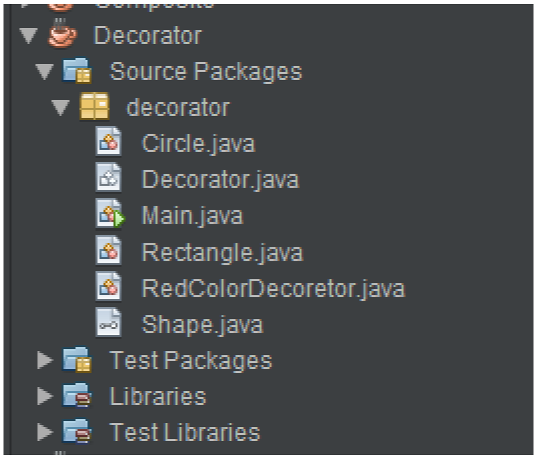


```
run:
100 Vatsal Malaviya
101 Tarang Mavani
200 Arjun Maniya
201 Kevla Merja
BUILD SUCCESSFUL (total time: 0 seconds)
```

b. Decoretor :-

- **Here I have use shape decorator as example :-**

File Structure :-



```
▼ Decorator
  ▼ Source Packages
    ▼ decorator
        Circle.java
        Decorator.java
        Main.java
        Rectangle.java
        RedColorDecoretor.java
        Shape.java
  ▶ Test Packages
  ▶ Libraries
  ▶ Test Libraries
```

Code:-

Circle.java :-

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package decorator;

/**
 *
 * @author admin
 */
public class Circle implements Shape {

    @Override
    public void draw() {
        System.out.print("\ncircle");
    }


}
```

Decoretor.java :-

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
to edit this template
 */
package decorator;

/**
 *
 * @author admin
 */
public abstract class Decorator implements Shape{
    protected Shape decoShape;
    
    public Decorator(Shape decoShape){
        this.decoShape = decoShape;
    }
    
    @Override
    public void draw() {
        decoShape.draw();
    }
    
    
}
```

Main.java :-

```
/*
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to
edit this template
 */
package decorator;
```

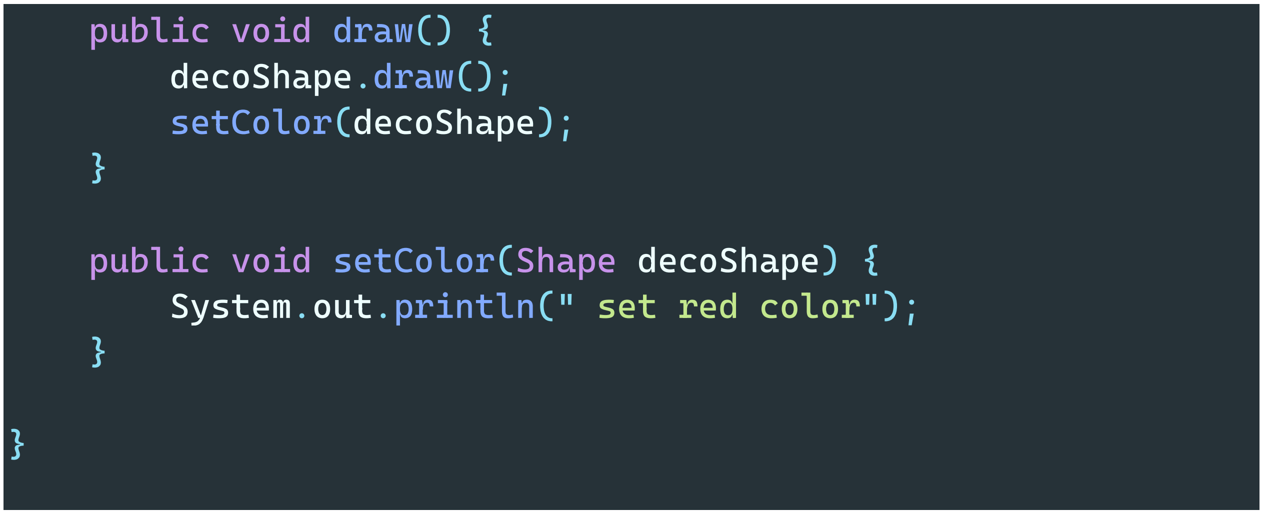
```java
/**
 *
 * @author admin
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
       Shape redCircle = new RedColorDecoretor(new Circle());
       Shape redRec = new RedColorDecoretor(new Rectangle());
       redCircle.draw();
       redRec.draw();

    }

}
```

Rectangle.java :-

```java
/*
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
to edit this template
 */
package decorator;

/**
 *
 * @author admin
 */
public class Rectangle implements Shape {

    @Override
    public void draw() {
            System.out.print("\nRectangle");
    }

}
```

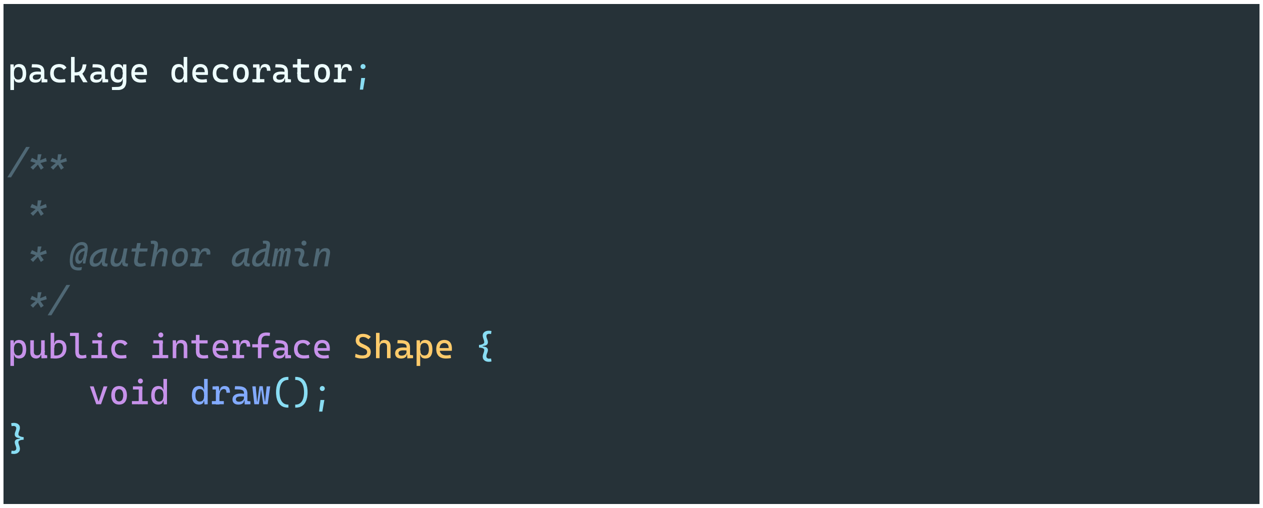
RedColorDecoretor.java :-

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package decorator;


/**
 *
 * @author admin
 */
public class RedColorDecoretor extends Decorator {
    public RedColorDecoretor(Shape decoShape)
    {
        super(decoShape);
    }

    @Override
```



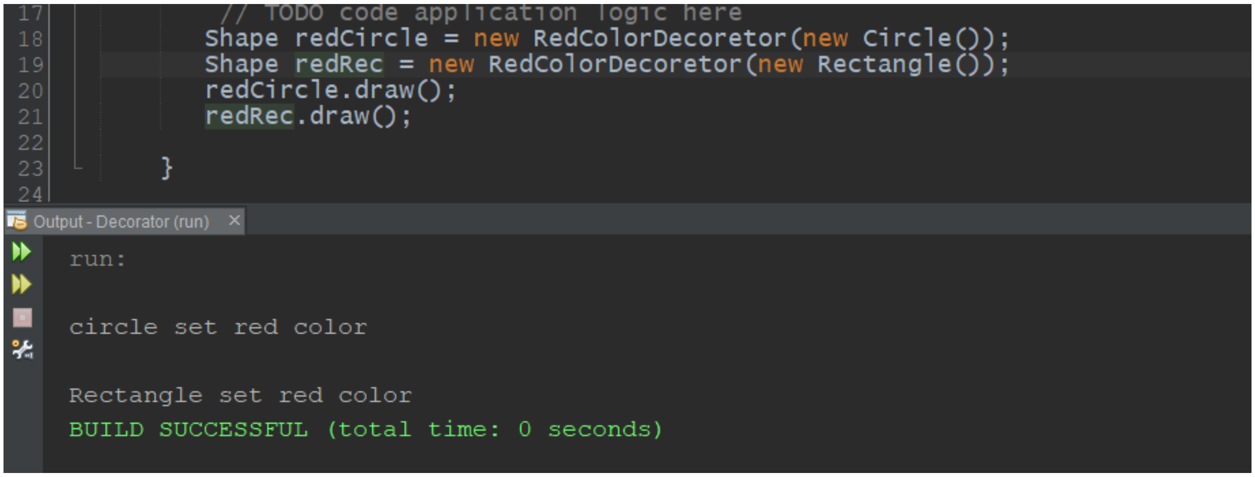
```
    public void draw() {
        decoShape.draw();
        setColor(decoShape);
    }

    public void setColor(Shape decoShape) {
        System.out.println(" set red color");
    }

}
```

Shape.java :-



```
package decorator;

/**
 *
 * @author admin
 */
public interface Shape {
    void draw();
}
```

Input / Output :-



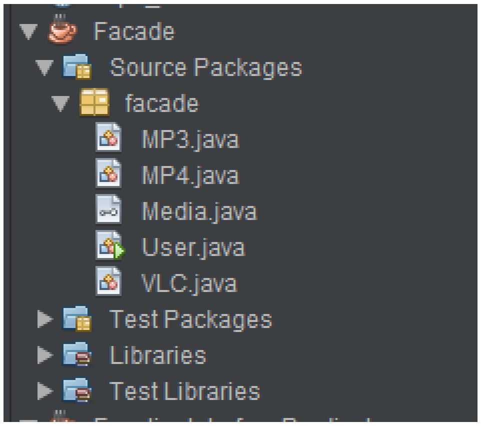
```
        // TODO code application logic here
        Shape redCircle = new RedColorDecoretor(new Circle());
        Shape redRec = new RedColorDecoretor(new Rectangle());
        redCircle.draw();
        redRec.draw();

    }
```

```
run:

circle set red color

Rectangle set red color
BUILD SUCCESSFUL (total time: 0 seconds)
```

c. Façade :-

Here I have take VLC example that can open media file as per user command (wish).

File Structure :-

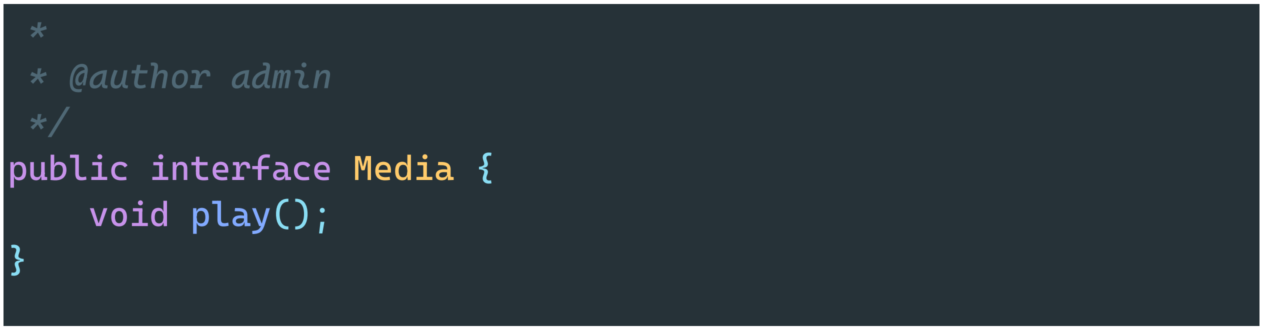


```
▼ Facade
  ▼ Source Packages
    ▼ facade
        MP3.java
        MP4.java
        Media.java
        User.java
        VLC.java
  ▶ Test Packages
  ▶ Libraries
  ▶ Test Libraries
```

Code :-

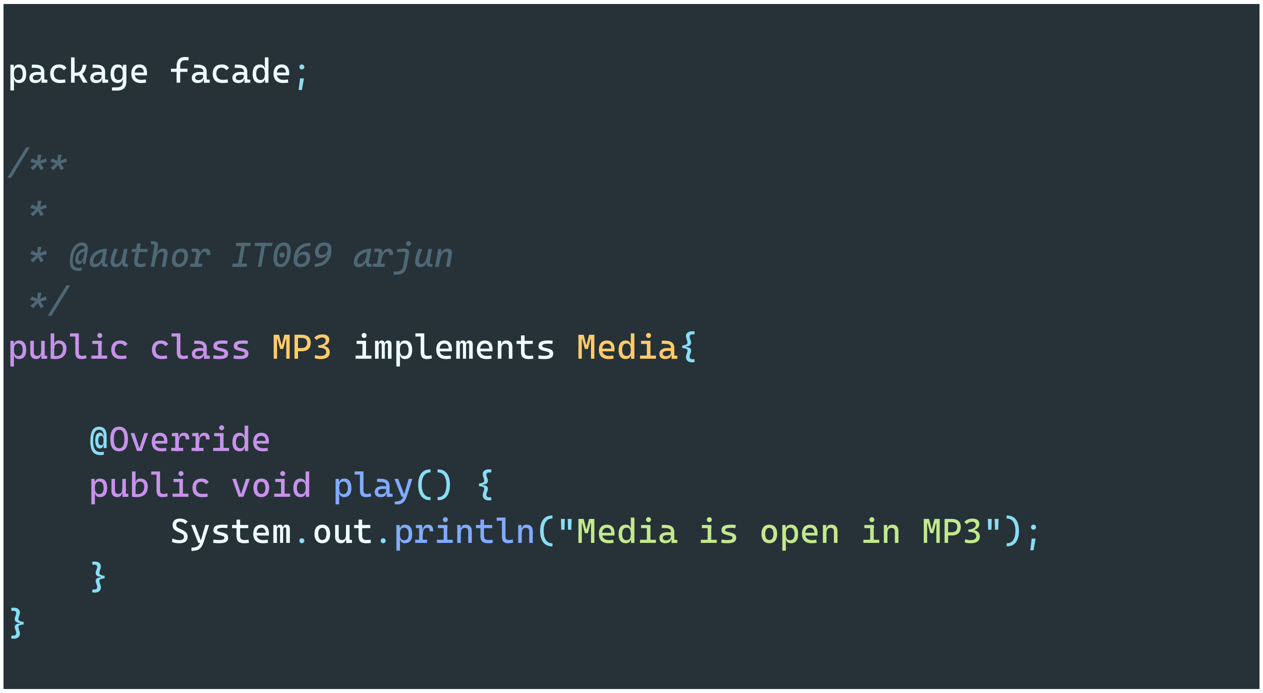
Media.java :-

```
package facade;

/**
```



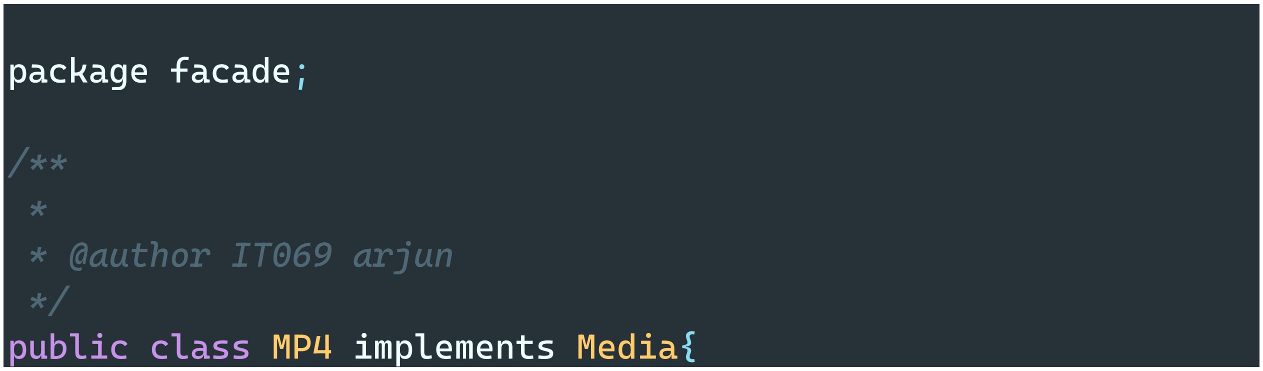
```
 *
 * @author admin
 */
public interface Media {
    void play();
}
```

MP3.java :-

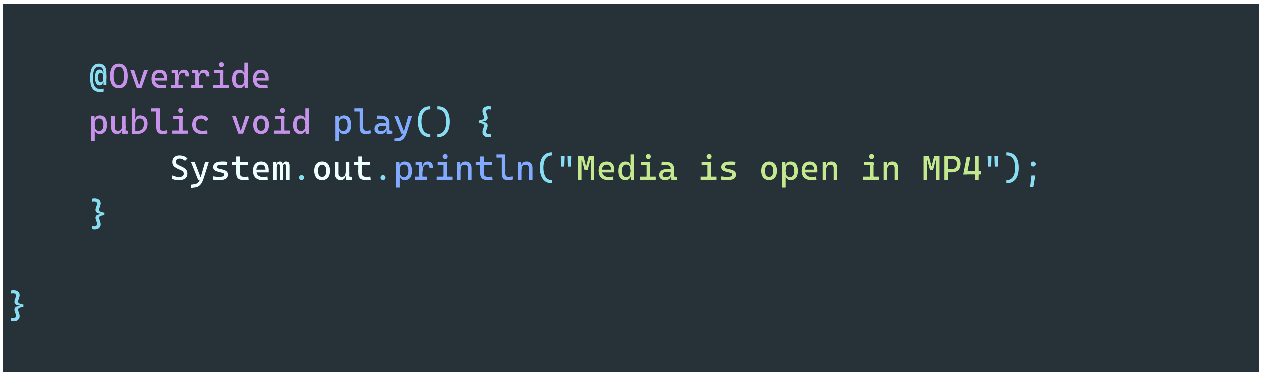


```
package facade;

/**
 *
 * @author IT069 arjun
 */
public class MP3 implements Media{

    @Override
    public void play() {
        System.out.println("Media is open in MP3");
    }
}
```

MP4.java :-

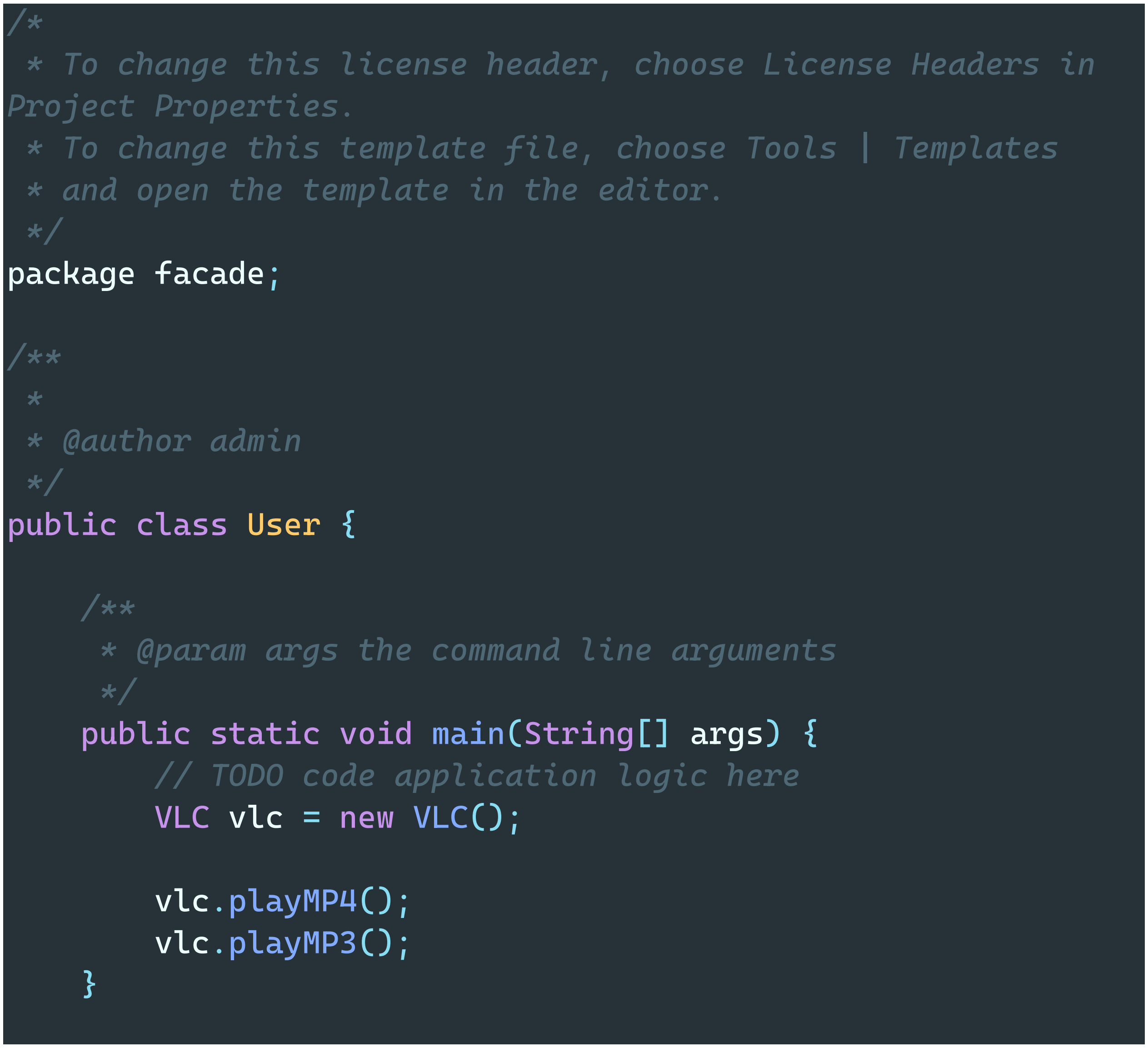


```
package facade;

/**
 *
 * @author IT069 arjun
 */
public class MP4 implements Media{
```



```java
    @Override
    public void play() {
        System.out.println("Media is open in MP4");
    }

}
```

User.java :-

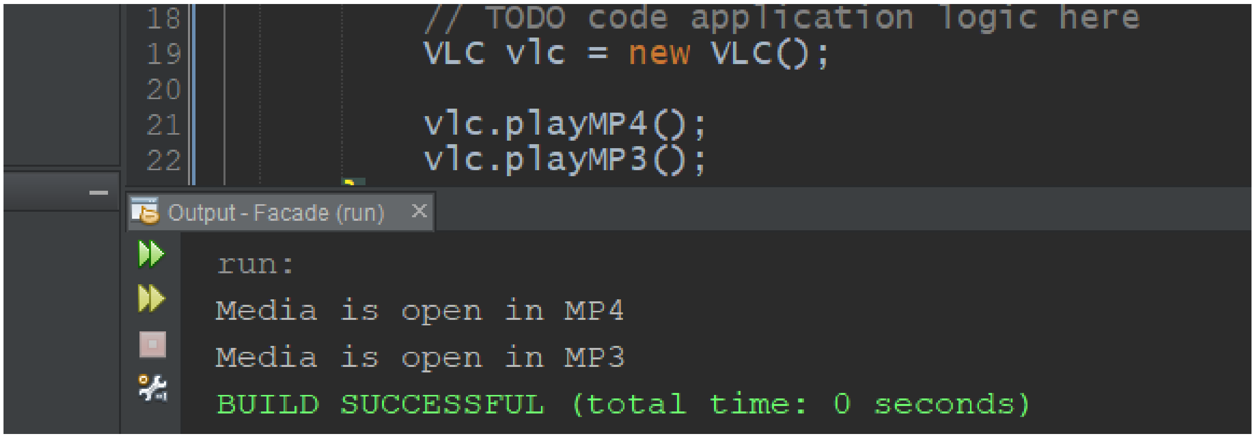


```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package facade;

/**
 *
 * @author admin
 */
public class User {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        VLC vlc = new VLC();

        vlc.playMP4();
        vlc.playMP3();
    }
}
```

VLC.java :-

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package facade;

/**
 *
 * @author IT069 arjun
 */
public class VLC {

    private Media mp4, mp3;

    public VLC() {
        mp4 = new MP4();
        mp3 = new MP3();
    }

    public void playMP3() {
        mp3.play();
    }

    public void playMP4() {
        mp4.play();
    }

}
```

Input/Output :-



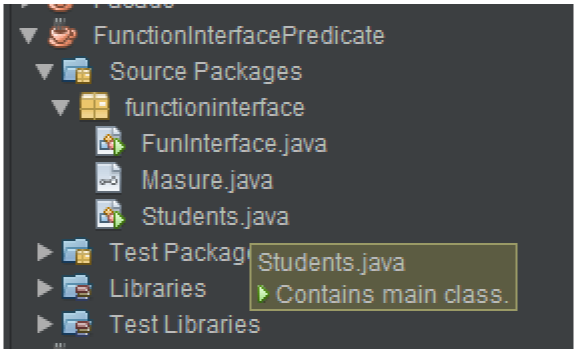
```
        // TODO code application logic here
        VLC vlc = new VLC();

        vlc.playMP4();
        vlc.playMP3();
```

```
run:
Media is open in MP4
Media is open in MP3
BUILD SUCCESSFUL (total time: 0 seconds)
```

Experiment-4

Aim : Solving programming problems using Java-8 functional programming**.**

File Structure :-



```
▼ FunctionInterfacePredicate
  ▼ Source Packages
    ▼ functioninterface
        FunInterface.java
        Masure.java
        Students.java
  ▶ Test Packag
  ▶ Libraries
  ▶ Test Libraries
```

Code :-

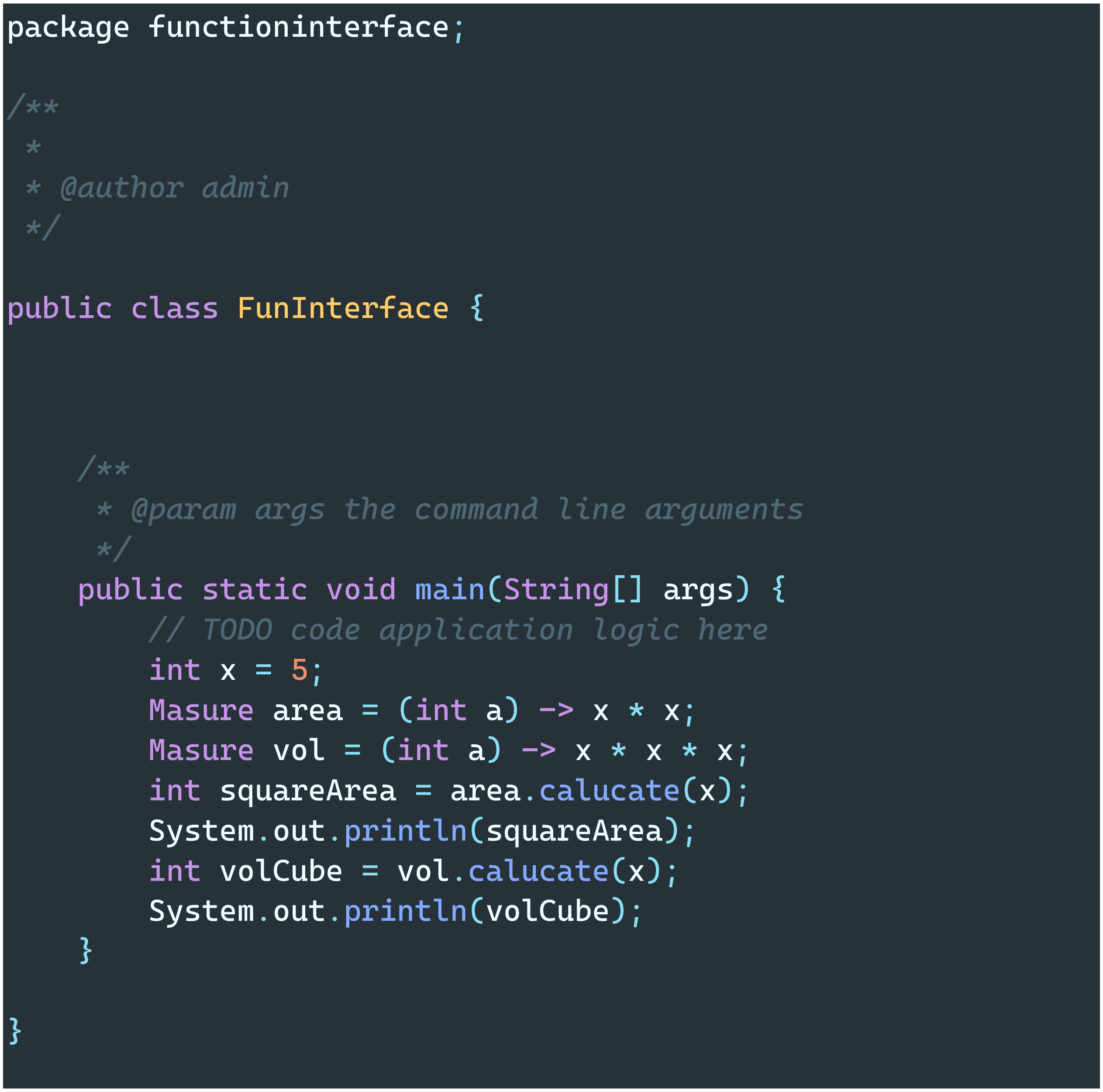
Functional Interface Example :-

Mesure.java:-

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package functioninterface;

/**
 *
 * @author admin
 */
@FunctionalInterface
interface Mesure{
    int calucate(int x);
}
```

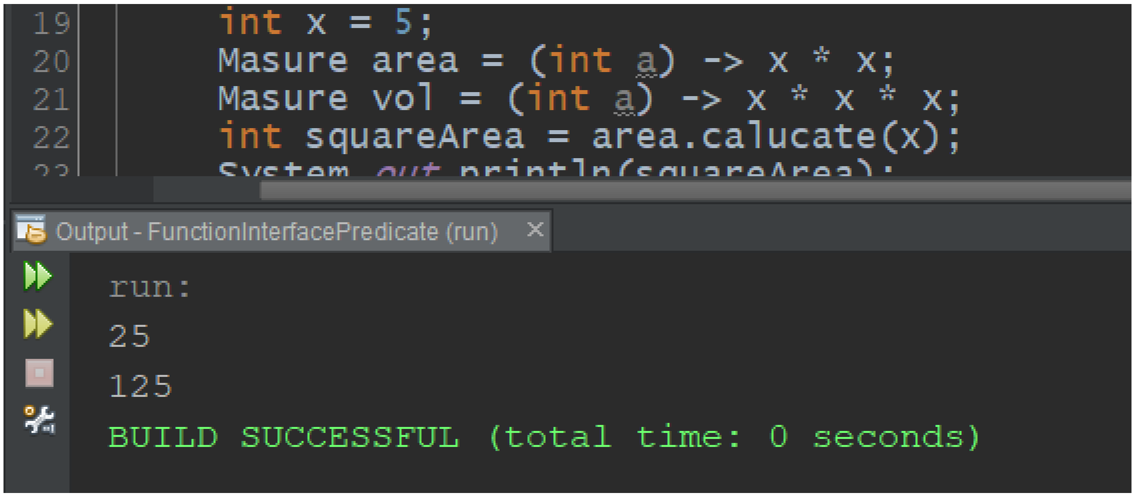
FunInterface.java :-

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
```



```
package functioninterface;

/**
 *
 * @author admin
 */

public class FunInterface {



    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        int x = 5;
        Masure area = (int a) -> x * x;
        Masure vol = (int a) -> x * x * x;
        int squareArea = area.calucate(x);
        System.out.println(squareArea);
        int volCube = vol.calucate(x);
        System.out.println(volCube);
    }

}
```

Input/Output :-



```
        int x = 5;
        Masure area = (int a) -> x * x;
        Masure vol = (int a) -> x * x * x;
        int squareArea = area.calucate(x);
        System.out.println(squareArea);
```

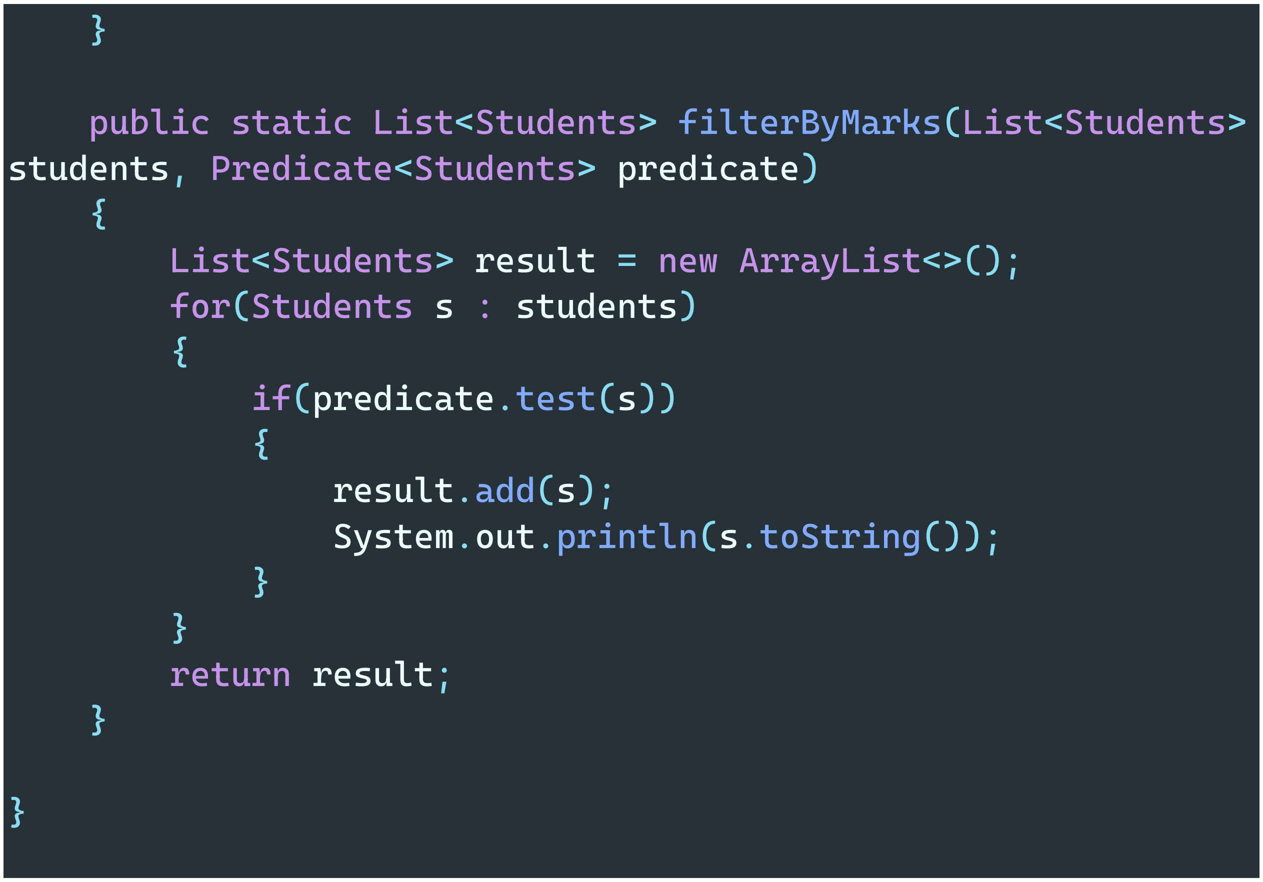
```
run:
25
125
BUILD SUCCESSFUL (total time: 0 seconds)
```

Predicate Example :-

Students.java :-

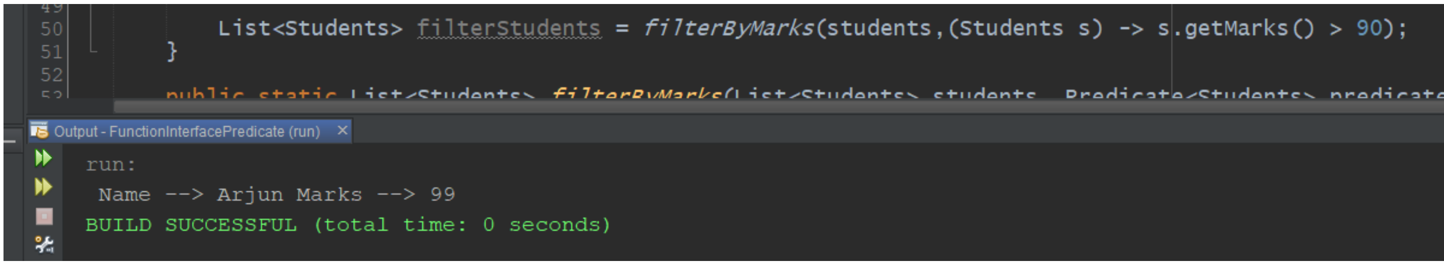
- **Here I have make one filter to sort out student by marks (ex. Name of students that have more than 90 marks)**

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package functioninterface;

import java.util.ArrayList;
import java.util.List;
import java.util.function.Predicate;

/**
 *
 * @author admin
 */
public class Students {
    String name;
    int marks;

    public Students() {
    }

    public Students(String name, int marks) {
        this.name = name;
        this.marks = marks;
    }

    public String getName() {
        return name;
    }

    public int getMarks() {
        return marks;
    }

    @Override
    public String toString() {
        return " Name --> "+getName()+" Marks --> "+
getMarks(); //To change body of generated methods, choose
Tools | Templates.
    }



    public static void main(String[] args) {
        // TODO code application logic here
        List<Students> students = new ArrayList<>();
        students.add(new Students("Arjun", 99));
        students.add(new Students("xyz", 50));

        List<Students> filterStudents =
filterByMarks(students,(Students s) -> s.getMarks() > 90);
```



```
    }

    public static List<Students> filterByMarks(List<Students>
students, Predicate<Students> predicate)
    {
        List<Students> result = new ArrayList<>();
        for(Students s : students)
        {
            if(predicate.test(s))
            {
                result.add(s);
                System.out.println(s.toString());
            }
        }
        return result;
    }

}
```

Input/Output :-



```
        List<Students> filterStudents = filterByMarks(students,(Students s) -> s.getMarks() > 90);
    }

    public static List<Students> filterByMarks(List<Students> students, Predicate<Students> predicate
```

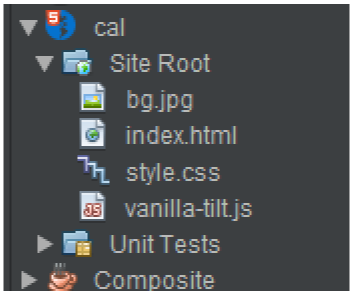
```
run:
 Name --> Arjun Marks --> 99
BUILD SUCCESSFUL (total time: 0 seconds)
```

Experiment-5

Aim :- : Designing and implementing web interface using JavaScript, and CSS

- **Here I have made simple calc using javascript and css .**

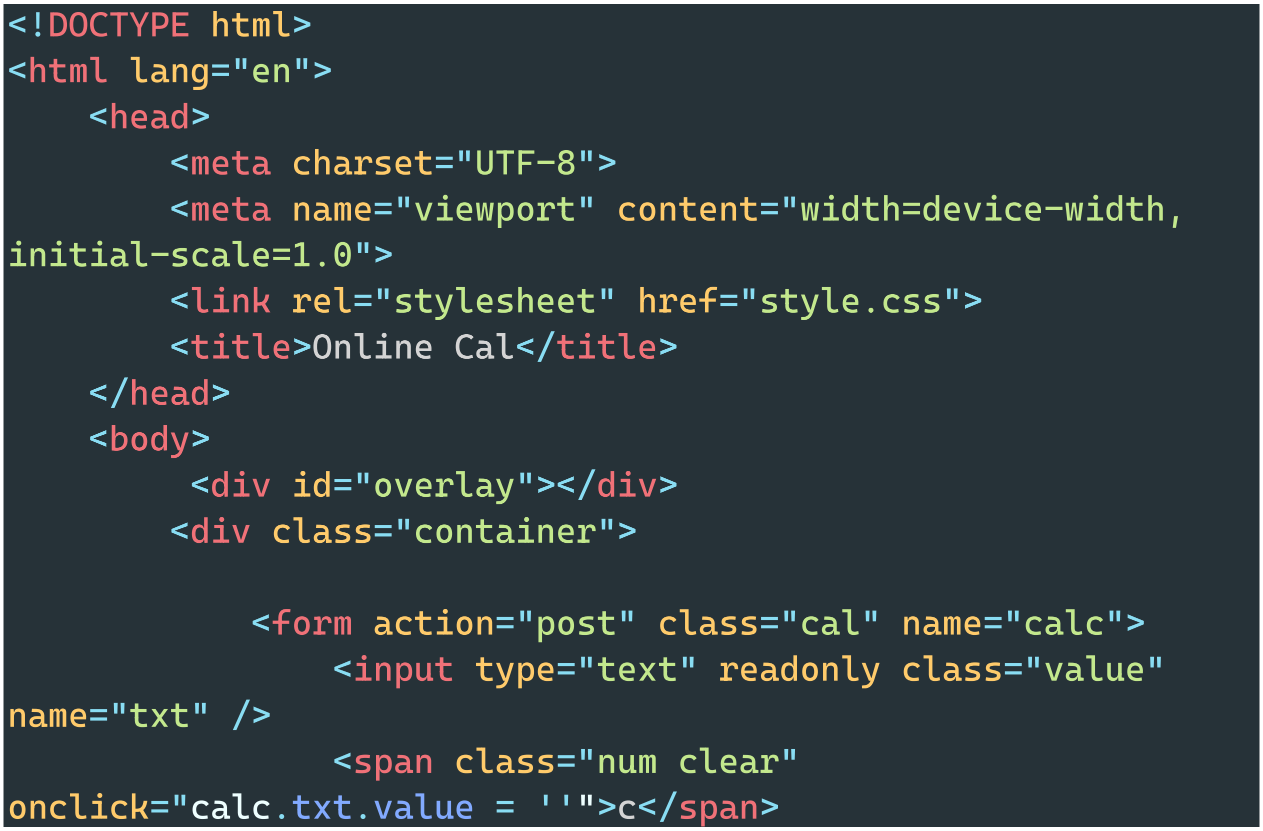
File Structure :-



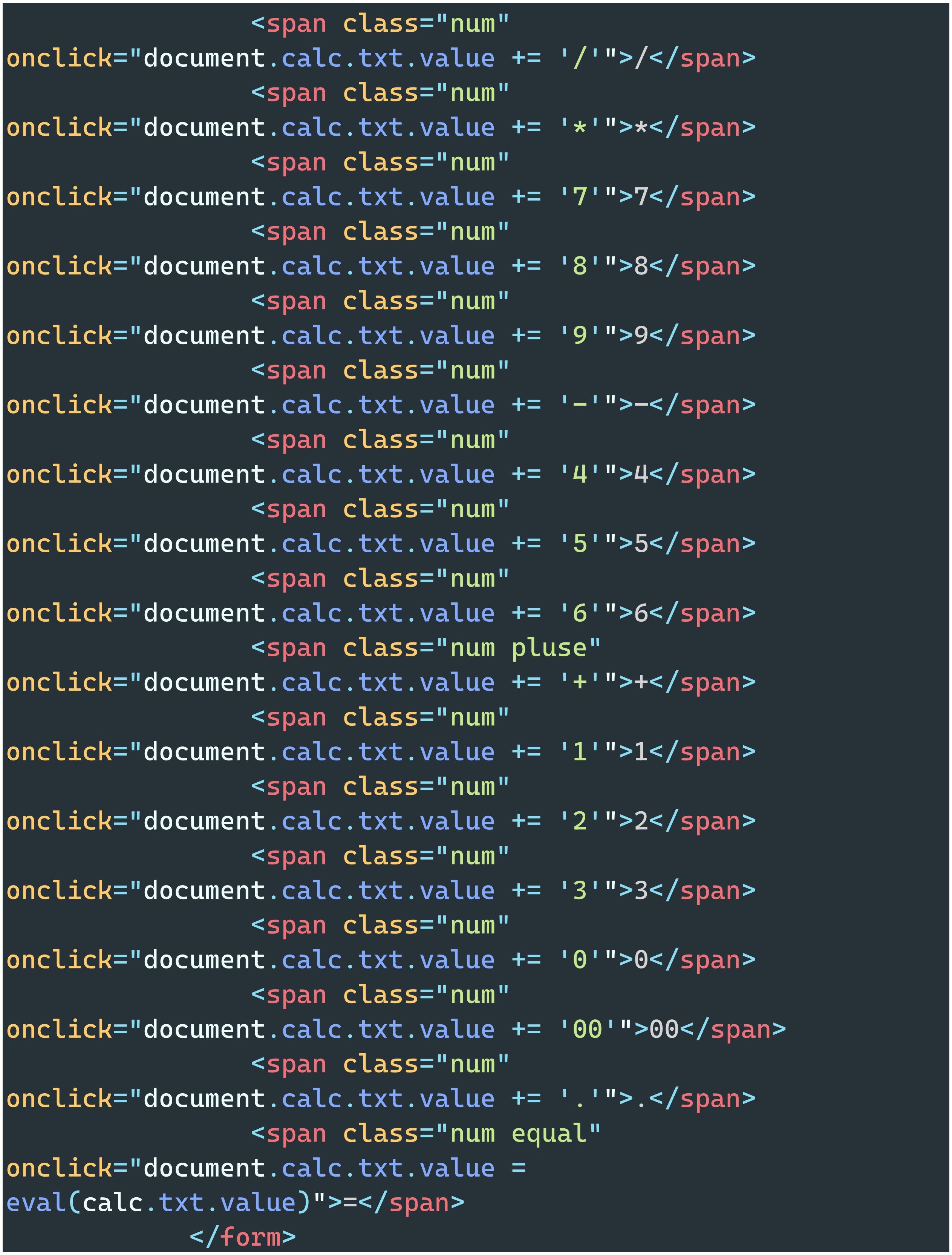
```
cal
  Site Root
    bg.jpg
    index.html
    style.css
    vanilla-tilt.js
  Unit Tests
Composite
```

Code :-

Index.html :-



```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1.0">
        <link rel="stylesheet" href="style.css">
        <title>Online Cal</title>
    </head>
    <body>
         <div id="overlay"></div>
        <div class="container">

            <form action="post" class="cal" name="calc">
                <input type="text" readonly class="value"
name="txt" />
                <span class="num clear"
onclick="calc.txt.value = ''">c</span>
```

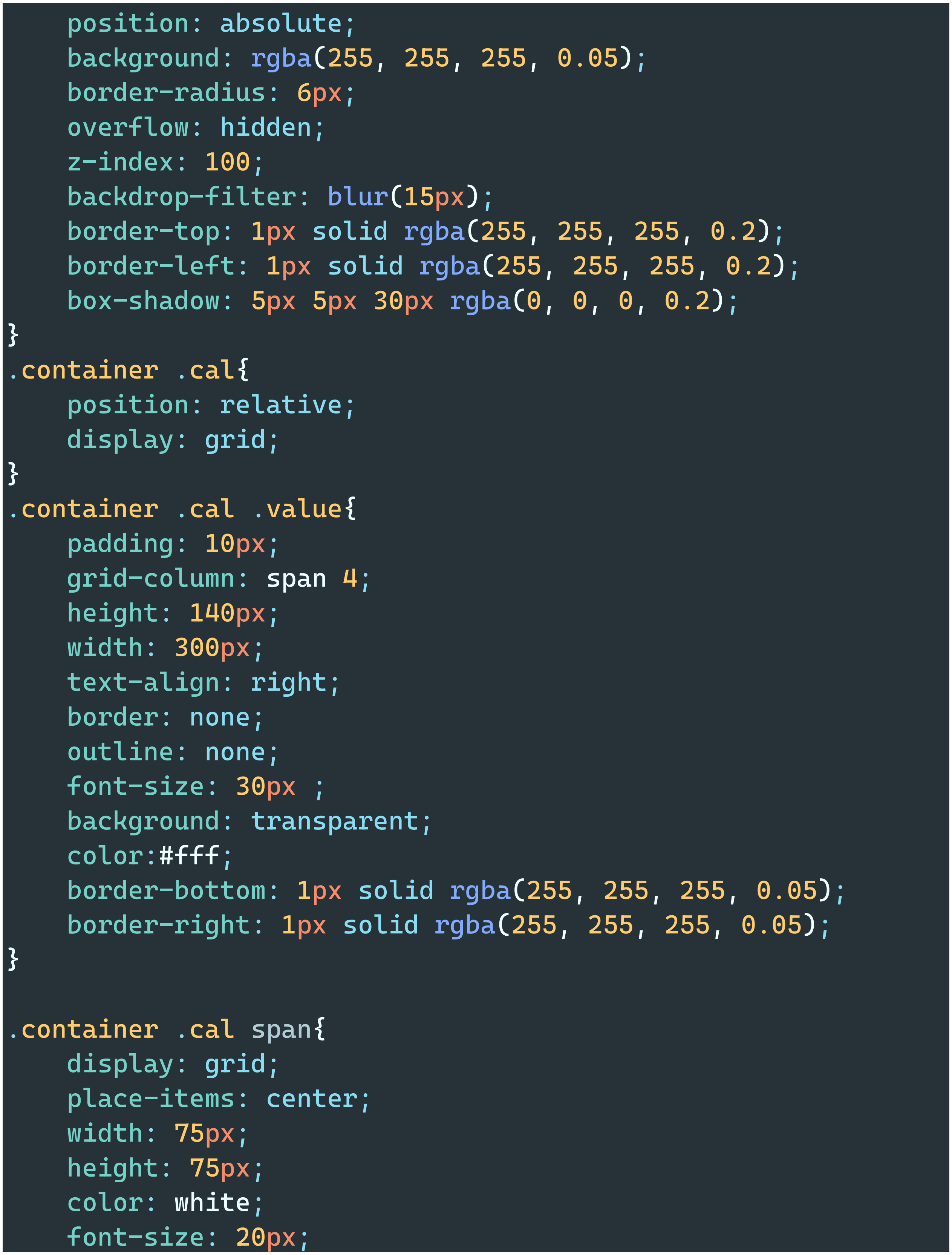


```
                <span class="num"
onclick="document.calc.txt.value += '/'">/</span>
                <span class="num"
onclick="document.calc.txt.value += '*'">*</span>
                <span class="num"
onclick="document.calc.txt.value += '7'">7</span>
                <span class="num"
onclick="document.calc.txt.value += '8'">8</span>
                <span class="num"
onclick="document.calc.txt.value += '9'">9</span>
                <span class="num"
onclick="document.calc.txt.value += '-'">-</span>
                <span class="num"
onclick="document.calc.txt.value += '4'">4</span>
                <span class="num"
onclick="document.calc.txt.value += '5'">5</span>
                <span class="num"
onclick="document.calc.txt.value += '6'">6</span>
                <span class="num pluse"
onclick="document.calc.txt.value += '+'">+</span>
                <span class="num"
onclick="document.calc.txt.value += '1'">1</span>
                <span class="num"
onclick="document.calc.txt.value += '2'">2</span>
                <span class="num"
onclick="document.calc.txt.value += '3'">3</span>
                <span class="num"
onclick="document.calc.txt.value += '0'">0</span>
                <span class="num"
onclick="document.calc.txt.value += '00'">00</span>
                <span class="num"
onclick="document.calc.txt.value += '.'">.</span>
                <span class="num equal"
onclick="document.calc.txt.value =
eval(calc.txt.value)">=</span>
            </form>
```

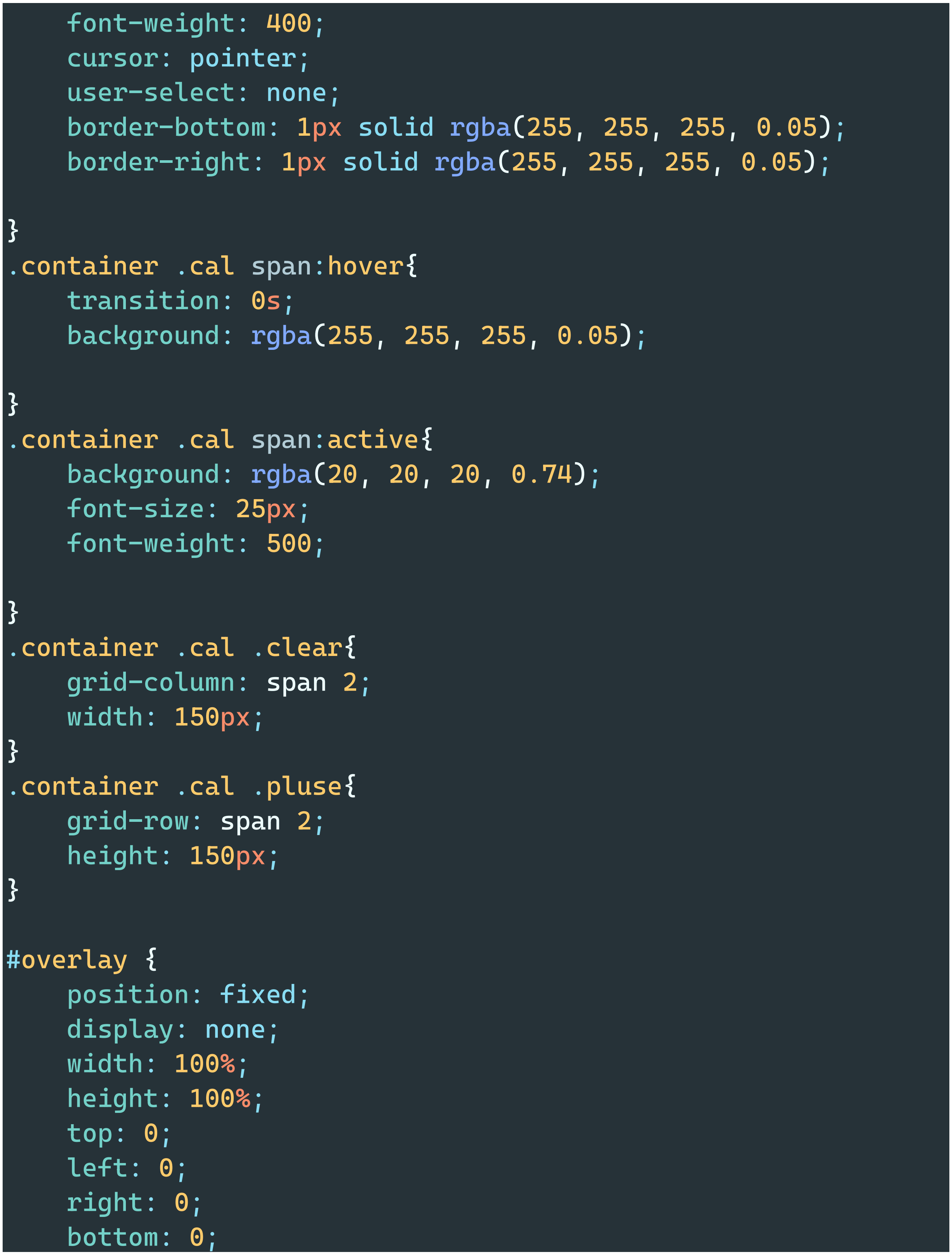
```
        </div>

    </body>
</html>
```

Style.css :-

```
/*
To change this license header, choose License Headers in
Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
*/
/*
    Created on : 28 Mar, 2022, 9:30:13 PM
    Author     : admin
*/

*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;

}
body{

    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background-image:url("bg.jpg");
    background-position: center;
    background-size: cover;

}
.container{
```

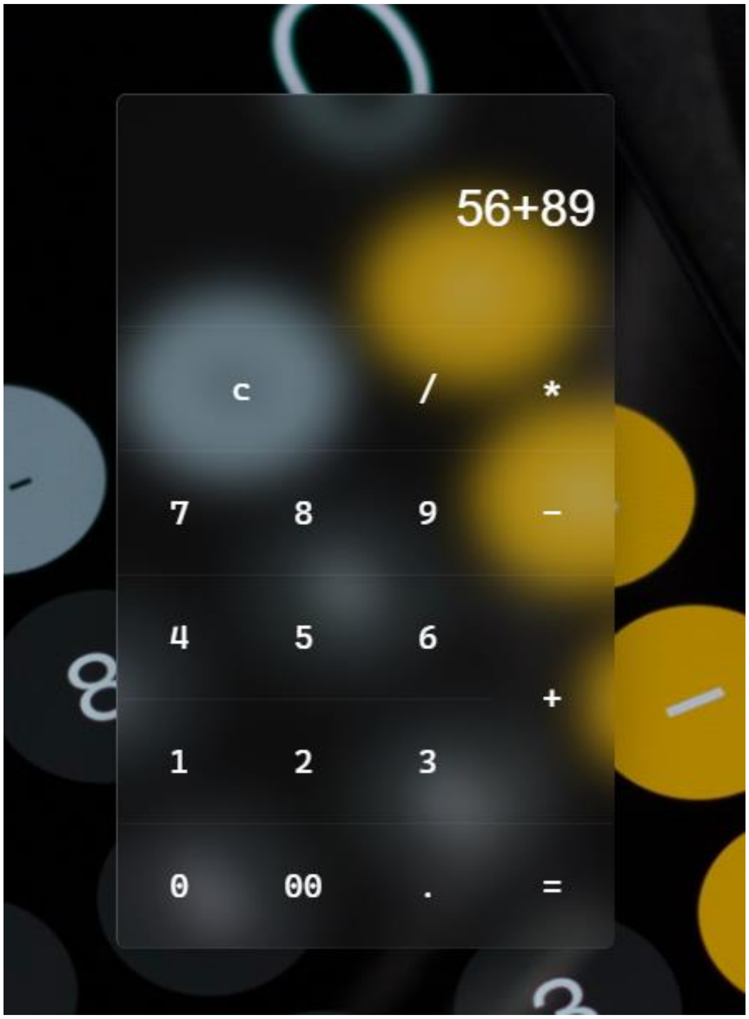


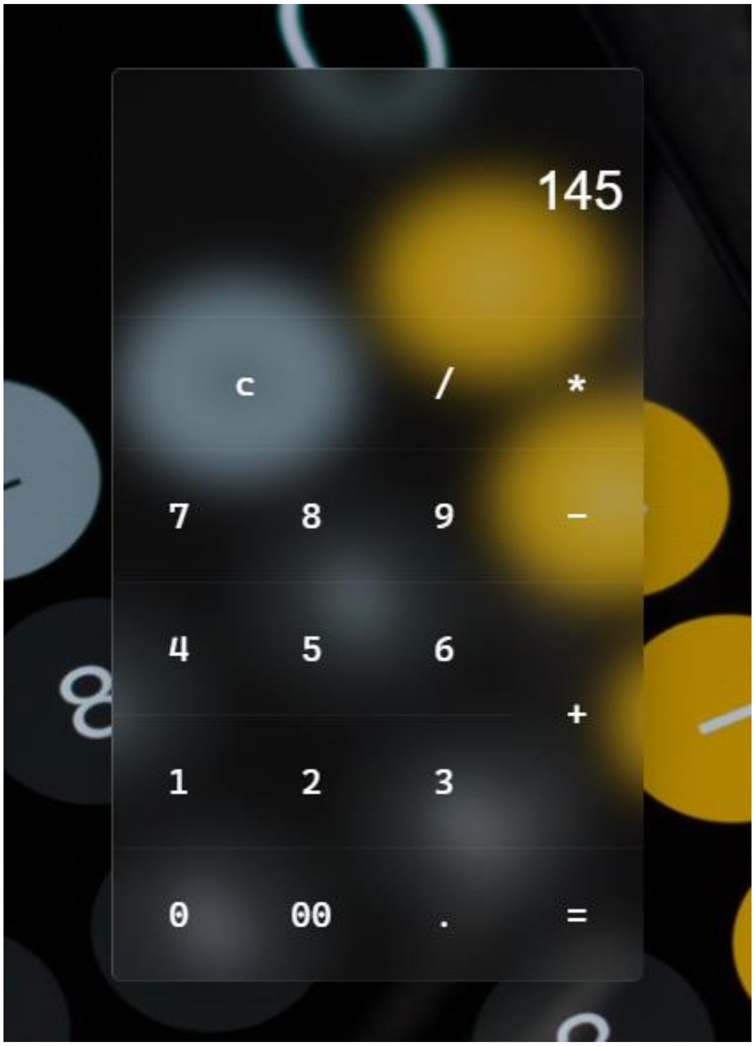
```
    position: absolute;
    background: rgba(255, 255, 255, 0.05);
    border-radius: 6px;
    overflow: hidden;
    z-index: 100;
    backdrop-filter: blur(15px);
    border-top: 1px solid rgba(255, 255, 255, 0.2);
    border-left: 1px solid rgba(255, 255, 255, 0.2);
    box-shadow: 5px 5px 30px rgba(0, 0, 0, 0.2);
}
.container .cal{
    position: relative;
    display: grid;
}
.container .cal .value{
    padding: 10px;
    grid-column: span 4;
    height: 140px;
    width: 300px;
    text-align: right;
    border: none;
    outline: none;
    font-size: 30px ;
    background: transparent;
    color:#fff;
    border-bottom: 1px solid rgba(255, 255, 255, 0.05);
    border-right: 1px solid rgba(255, 255, 255, 0.05);
}

.container .cal span{
    display: grid;
    place-items: center;
    width: 75px;
    height: 75px;
    color: white;
    font-size: 20px;
```



```
    font-weight: 400;
    cursor: pointer;
    user-select: none;
    border-bottom: 1px solid rgba(255, 255, 255, 0.05);
    border-right: 1px solid rgba(255, 255, 255, 0.05);

}
.container .cal span:hover{
    transition: 0s;
    background: rgba(255, 255, 255, 0.05);

}
.container .cal span:active{
    background: rgba(20, 20, 20, 0.74);
    font-size: 25px;
    font-weight: 500;

}
.container .cal .clear{
    grid-column: span 2;
    width: 150px;
}
.container .cal .pluse{
    grid-row: span 2;
    height: 150px;
}

#overlay {
    position: fixed;
    display: none;
    width: 100%;
    height: 100%;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background-color: rgba(0,0,0,0.5);
    z-index: 1;
    cursor: pointer;
}
```

Input/Output :-

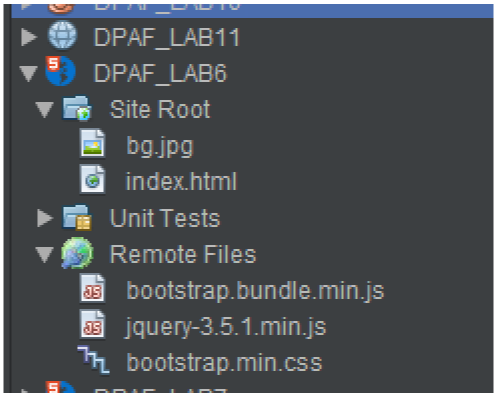



145
c
/
*
7
8
9
–
4
5
6
+
1
2
3
0
00
.
=

Experiment-6

Aim : Designing and implementing web interface using Bootstrap and customizing CSS

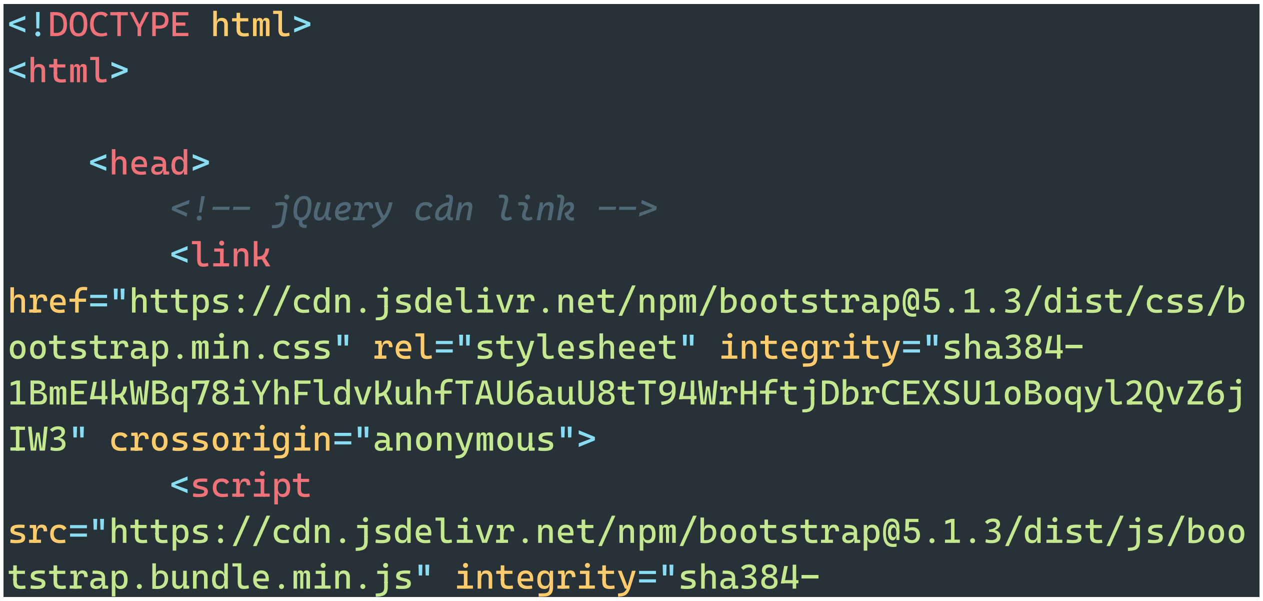
File structure :-



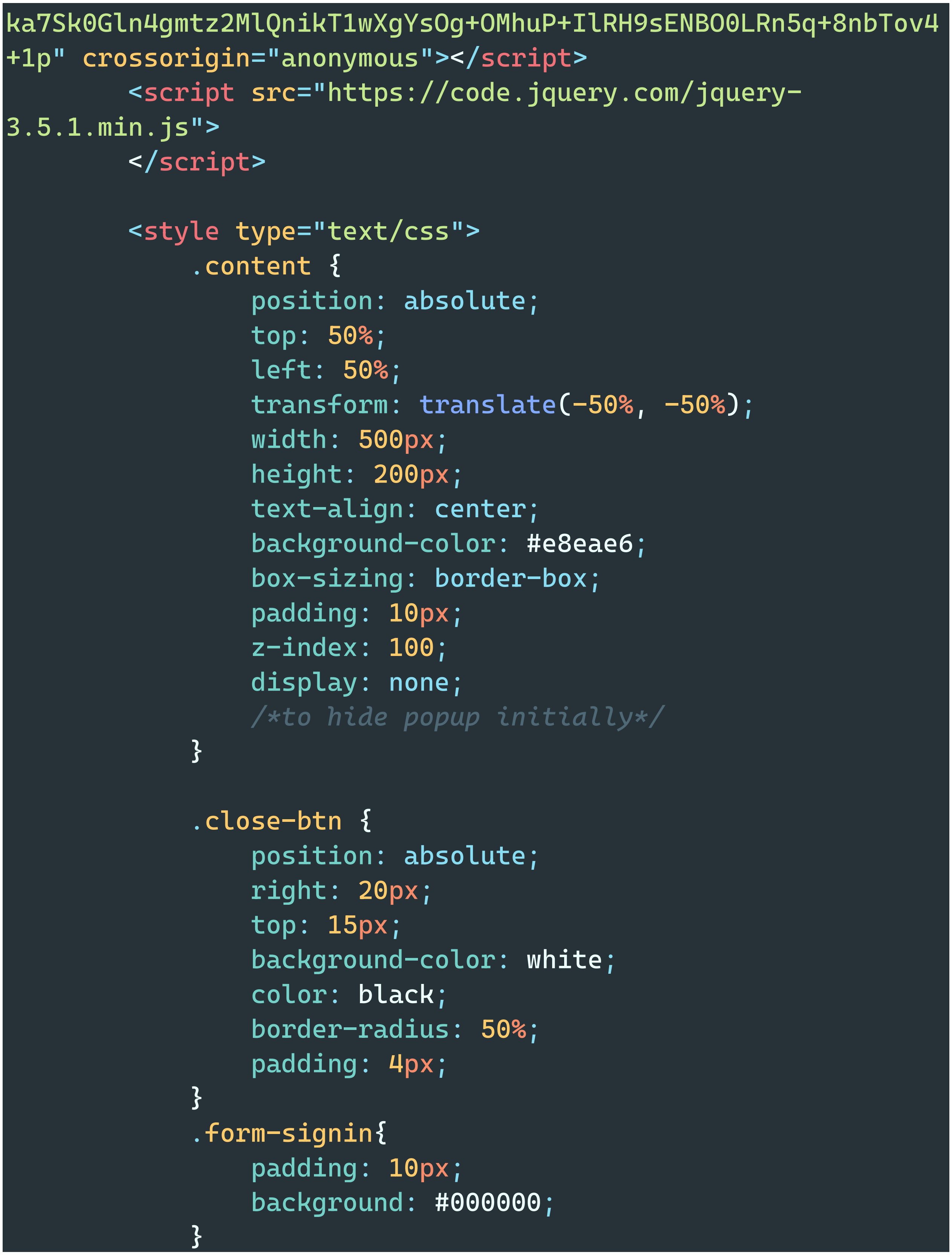
```
▶ DPAF_LAB11
▼ DPAF_LAB6
    ▼ Site Root
        bg.jpg
        index.html
    ▶ Unit Tests
    ▼ Remote Files
        bootstrap.bundle.min.js
        jquery-3.5.1.min.js
        bootstrap.min.css
```

Code :-

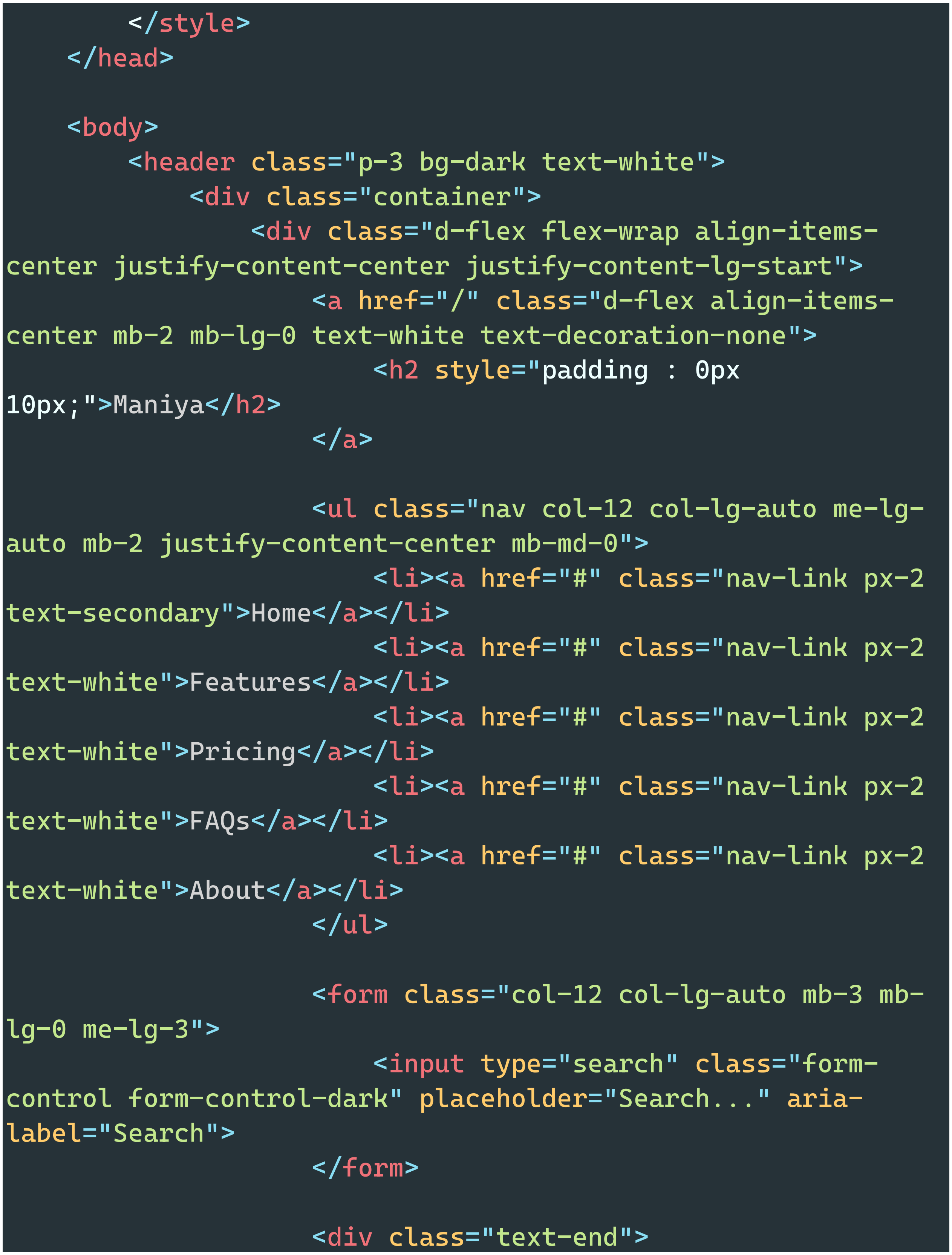
index.html:-



```
<!DOCTYPE html>
<html>

    <head>
        <!-- jQuery cdn link -->
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/b
ootstrap.min.css" rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6j
IW3" crossorigin="anonymous">
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/boo
tstrap.bundle.min.js" integrity="sha384-
```

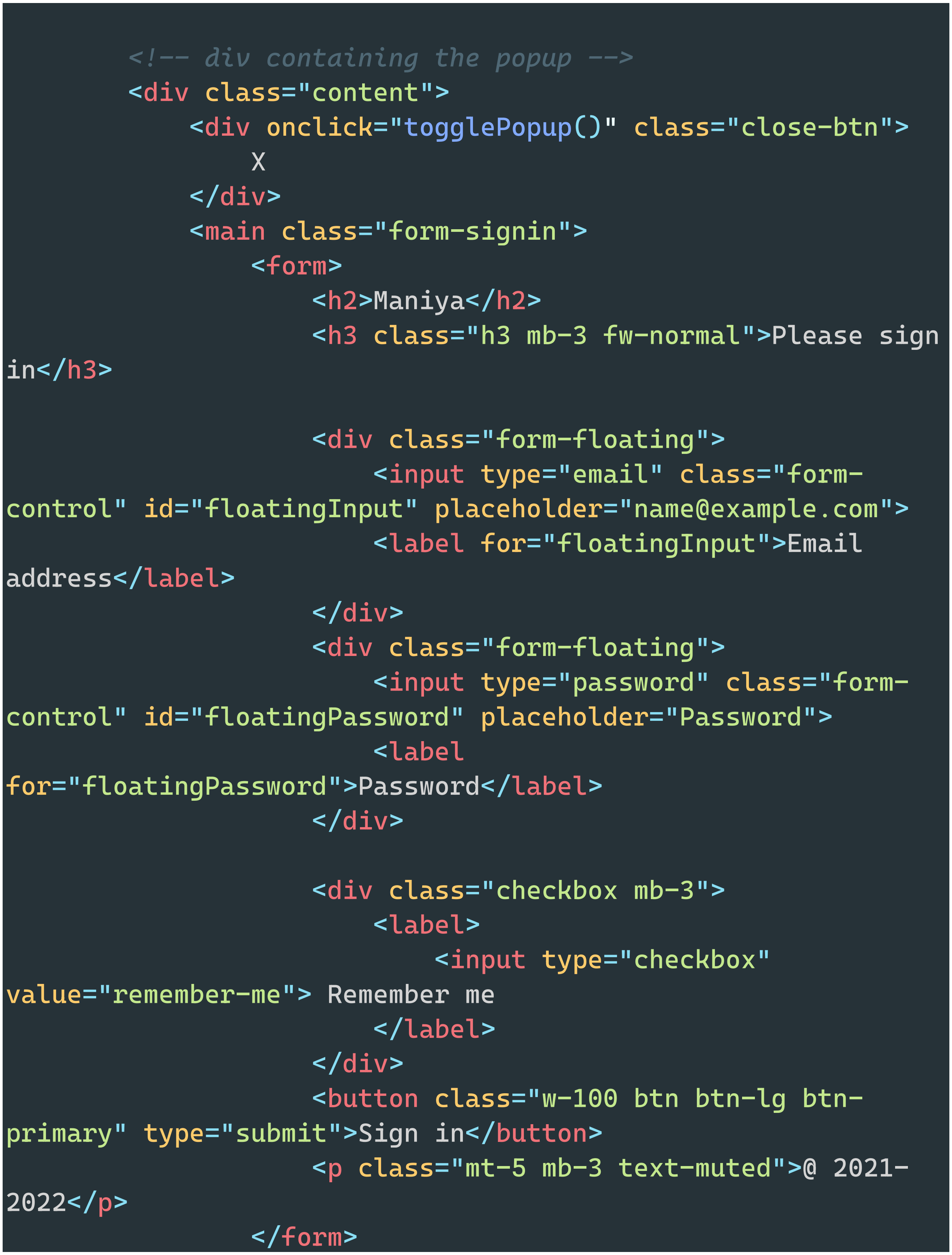


```
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4
+1p" crossorigin="anonymous"></script>
        <script src="https://code.jquery.com/jquery-
3.5.1.min.js">
        </script>

        <style type="text/css">
            .content {
                position: absolute;
                top: 50%;
                left: 50%;
                transform: translate(-50%, -50%);
                width: 500px;
                height: 200px;
                text-align: center;
                background-color: #e8eae6;
                box-sizing: border-box;
                padding: 10px;
                z-index: 100;
                display: none;
                /*to hide popup initially*/
            }

            .close-btn {
                position: absolute;
                right: 20px;
                top: 15px;
                background-color: white;
                color: black;
                border-radius: 50%;
                padding: 4px;
            }
            .form-signin{
                padding: 10px;
                background: #000000;
            }
```



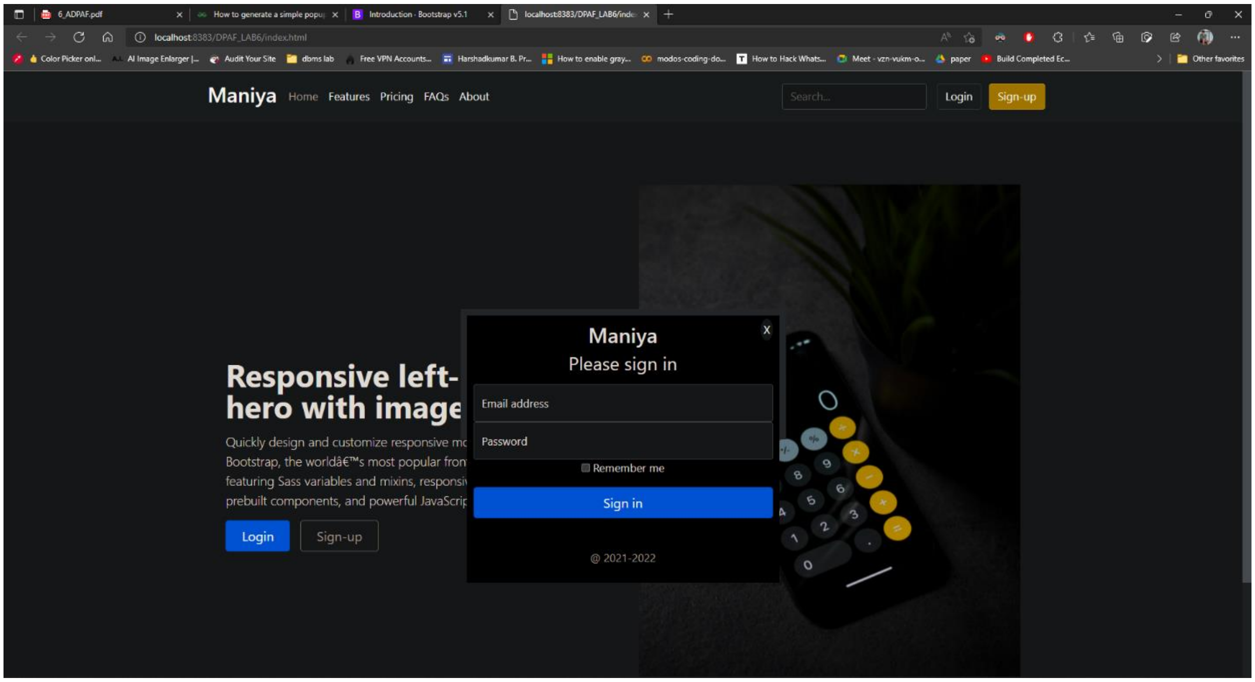
```
        </style>
    </head>

    <body>
        <header class="p-3 bg-dark text-white">
            <div class="container">
                <div class="d-flex flex-wrap align-items-
center justify-content-center justify-content-lg-start">
                    <a href="/" class="d-flex align-items-
center mb-2 mb-lg-0 text-white text-decoration-none">
                        <h2 style="padding : 0px
10px;">Maniya</h2>
                    </a>

                    <ul class="nav col-12 col-lg-auto me-lg-
auto mb-2 justify-content-center mb-md-0">
                        <li><a href="#" class="nav-link px-2
text-secondary">Home</a></li>
                        <li><a href="#" class="nav-link px-2
text-white">Features</a></li>
                        <li><a href="#" class="nav-link px-2
text-white">Pricing</a></li>
                        <li><a href="#" class="nav-link px-2
text-white">FAQs</a></li>
                        <li><a href="#" class="nav-link px-2
text-white">About</a></li>
                    </ul>

                    <form class="col-12 col-lg-auto mb-3 mb-
lg-0 me-lg-3">
                        <input type="search" class="form-
control form-control-dark" placeholder="Search..." aria-
label="Search">
                    </form>

                    <div class="text-end">
```

```
                        <button type="button" class="btn btn-
outline-light me-2" onclick="togglePopup()">Login</button>
                        <button type="button" class="btn btn-
warning" >Sign-up</button>
                    </div>
                </div>
            </div>
        </header>

        <div class="container col-xxl-8 px-4 py-5">
            <div class="row flex-lg-row-reverse align-items-
center g-5 py-5">
                <div class="col-10 col-sm-8 col-lg-6">
                    <img src="bg.jpg" class="d-block mx-lg-
auto img-fluid" alt="Bootstrap Themes" width="700"
height="500" loading="lazy">
                </div>
                <div class="col-lg-6">
                    <h1 class="display-5 fw-bold lh-1 mb-
3">Responsive left-aligned hero with image</h1>
                    <p class="lead">Quickly design and
customize responsive mobile-first sites with Bootstrap, the
world’s most popular front-end open source toolkit, featuring
Sass variables and mixins, responsive grid system, extensive
prebuilt components, and powerful JavaScript plugins.</p>
                    <div class="d-grid gap-2 d-md-flex
justify-content-md-start">
                        <button type="button" class="btn btn-
primary btn-lg px-4 me-md-2"
onclick="togglePopup()">Login</button>
                        <button type="button" class="btn btn-
outline-secondary btn-lg px-4">Sign-up</button>
                    </div>
                </div>
            </div>
        </div>
```



```
        <!-- div containing the popup -->
        <div class="content">
            <div onclick="togglePopup()" class="close-btn">
                X
            </div>
            <main class="form-signin">
                <form>
                    <h2>Maniya</h2>
                    <h3 class="h3 mb-3 fw-normal">Please sign
in</h3>

                    <div class="form-floating">
                        <input type="email" class="form-
control" id="floatingInput" placeholder="name@example.com">
                        <label for="floatingInput">Email
address</label>
                    </div>
                    <div class="form-floating">
                        <input type="password" class="form-
control" id="floatingPassword" placeholder="Password">
                        <label
for="floatingPassword">Password</label>
                    </div>

                    <div class="checkbox mb-3">
                        <label>
                            <input type="checkbox"
value="remember-me"> Remember me
                        </label>
                    </div>
                    <button class="w-100 btn btn-lg btn-
primary" type="submit">Sign in</button>
                    <p class="mt-5 mb-3 text-muted">@ 2021-
2022</p>
                </form>
            </main>
        </div>

        <script type="text/javascript">

            // Function to show and hide the popup
            function togglePopup() {
                $(".content").toggle();
            }
        </script>



</html>
```

Input / output :-



Responsive left-
hero with image

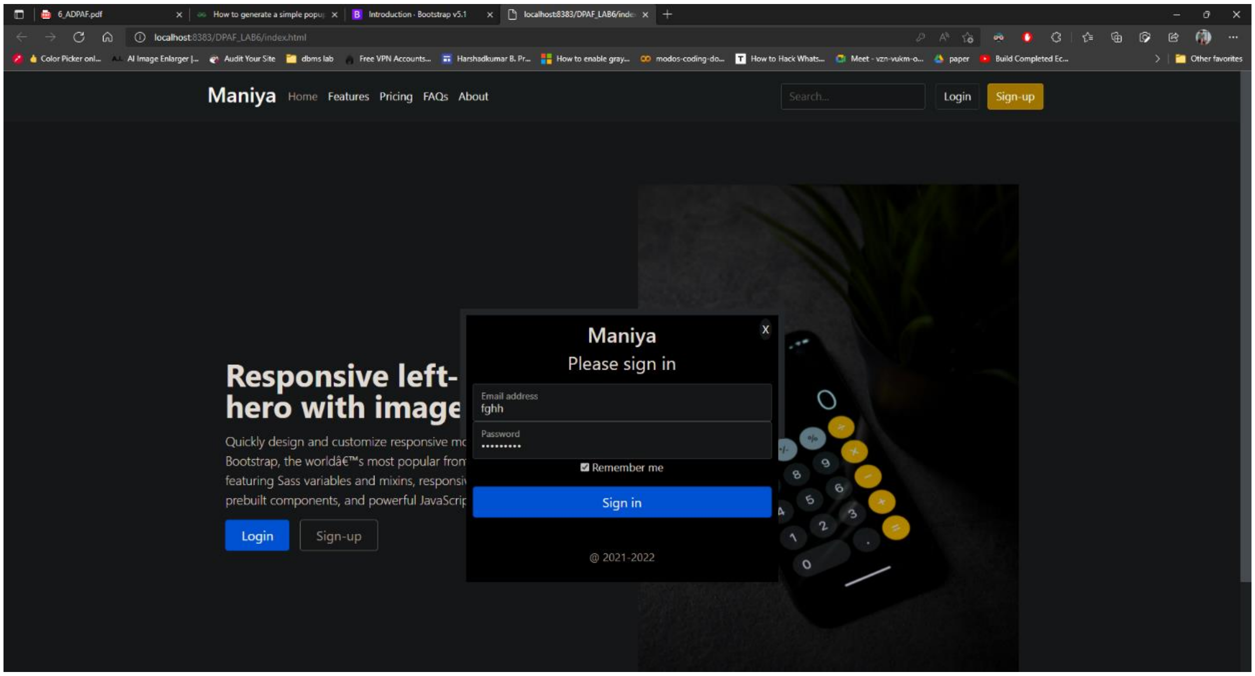
Quickly design and customize responsive mc
Bootstrap, the worldâ€™s most popular fron
featuring Sass variables and mixins, responsi
prebuilt components, and powerful JavaScrip

Maniya

Please sign in

KEVLA MERJA
30/03/22

DPAF LAB



Responsive left-
hero with image

Quickly design and customize responsive mc
Bootstrap, the worldâ€™s most popular fron
featuring Sass variables and mixins, responsi
prebuilt components, and powerful JavaScrip

Login Sign-up

Maniya

Please sign in

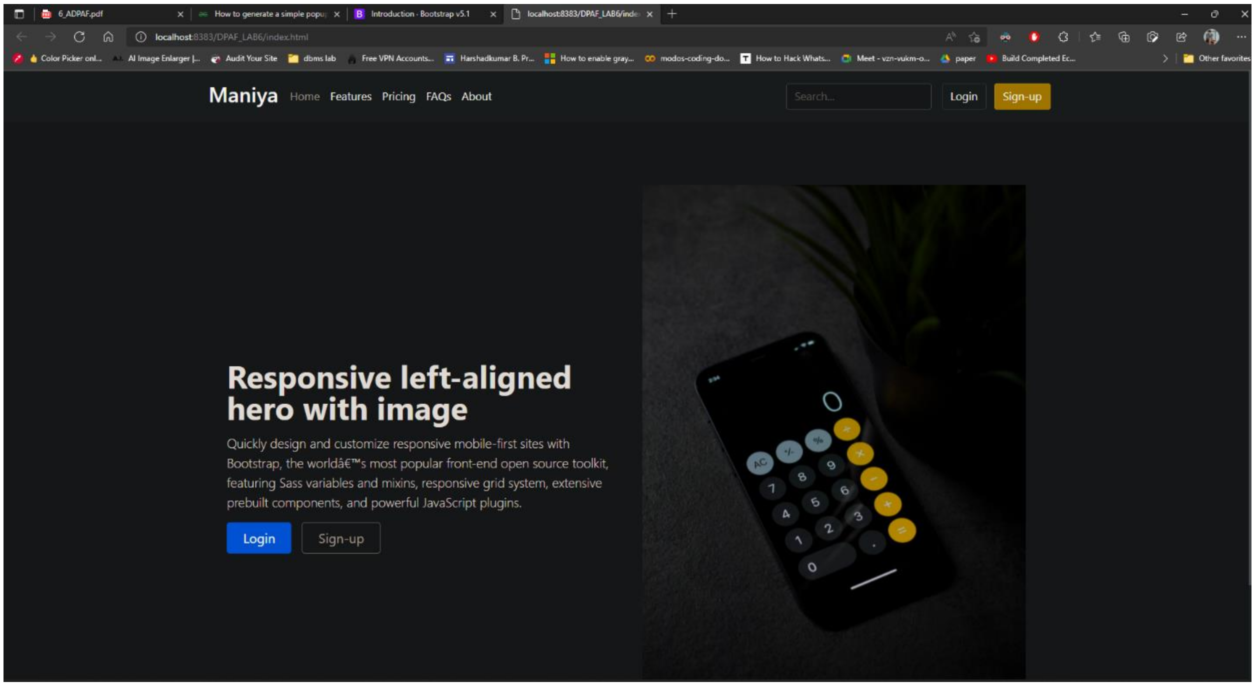
Email address
fghh

Password
•••••••••

☑ Remember me

Sign in

@ 2021-2022



Responsive left-aligned hero with image

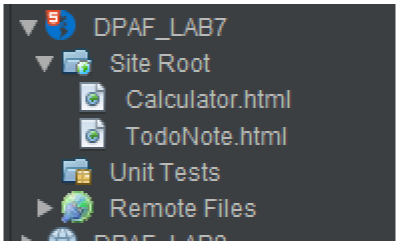
Quickly design and customize responsive mobile-first sites with Bootstrap, the worldâ€™s most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

Login Sign-up

Experiment-7

Aim : Designing and implementing a single page web interface using Angular JS

File Structure :-

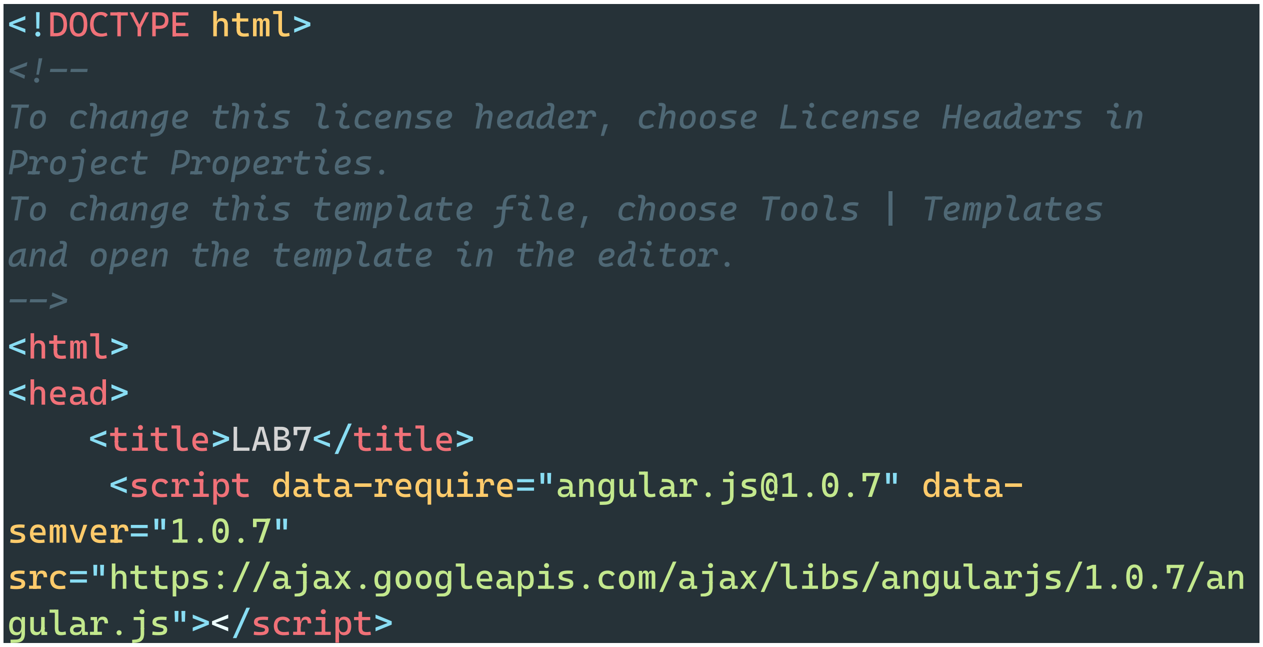


DPAF_LAB7
Site Root
Calculator.html
TodoNote.html
Unit Tests
Remote Files

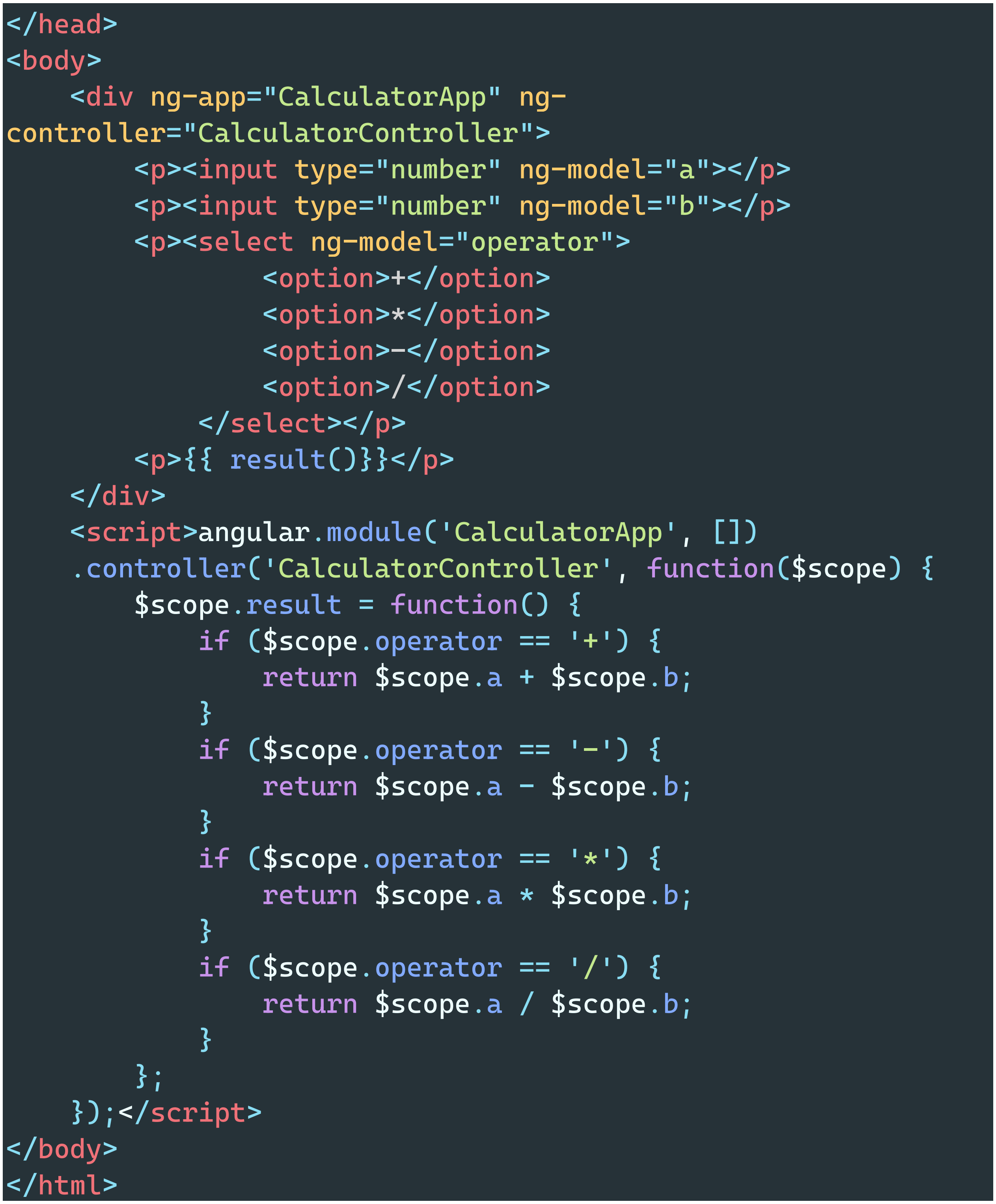
Code :-

Basic Calculator :-

Cal.html :-

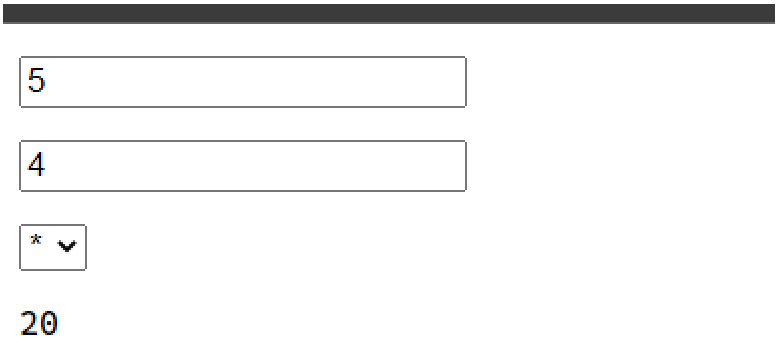


```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in
Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
<head>
    <title>LAB7</title>
     <script data-require="angular.js@1.0.7" data-
semver="1.0.7"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/an
gular.js"></script>
```



```
</head>
<body>
    <div ng-app="CalculatorApp" ng-
controller="CalculatorController">
        <p><input type="number" ng-model="a"></p>
        <p><input type="number" ng-model="b"></p>
        <p><select ng-model="operator">
                <option>+</option>
                <option>*</option>
                <option>-</option>
                <option>/</option>
            </select></p>
        <p>{{ result()}}</p>
    </div>
    <script>angular.module('CalculatorApp', [])
    .controller('CalculatorController', function($scope) {
        $scope.result = function() {
            if ($scope.operator == '+') {
                return $scope.a + $scope.b;
            }
            if ($scope.operator == '-') {
                return $scope.a - $scope.b;
            }
            if ($scope.operator == '*') {
                return $scope.a * $scope.b;
            }
            if ($scope.operator == '/') {
                return $scope.a / $scope.b;
            }
        };
    });</script>
</body>
</html>
```

Input / Output :-



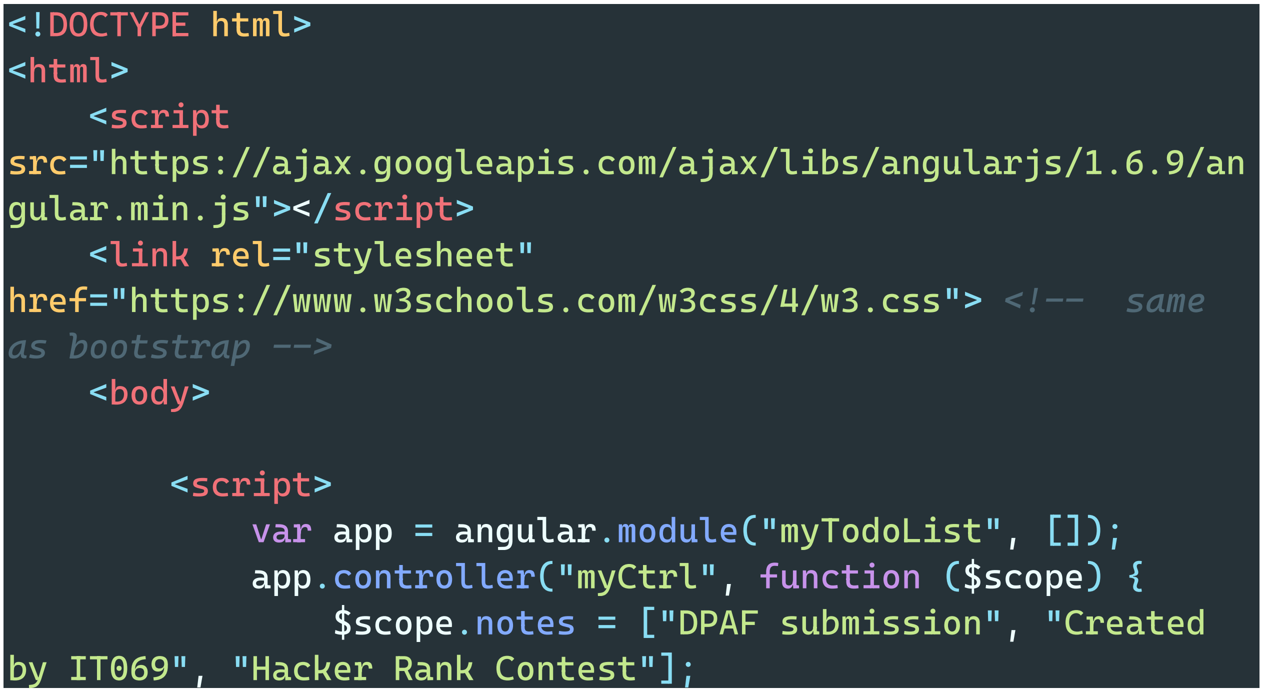
5

4

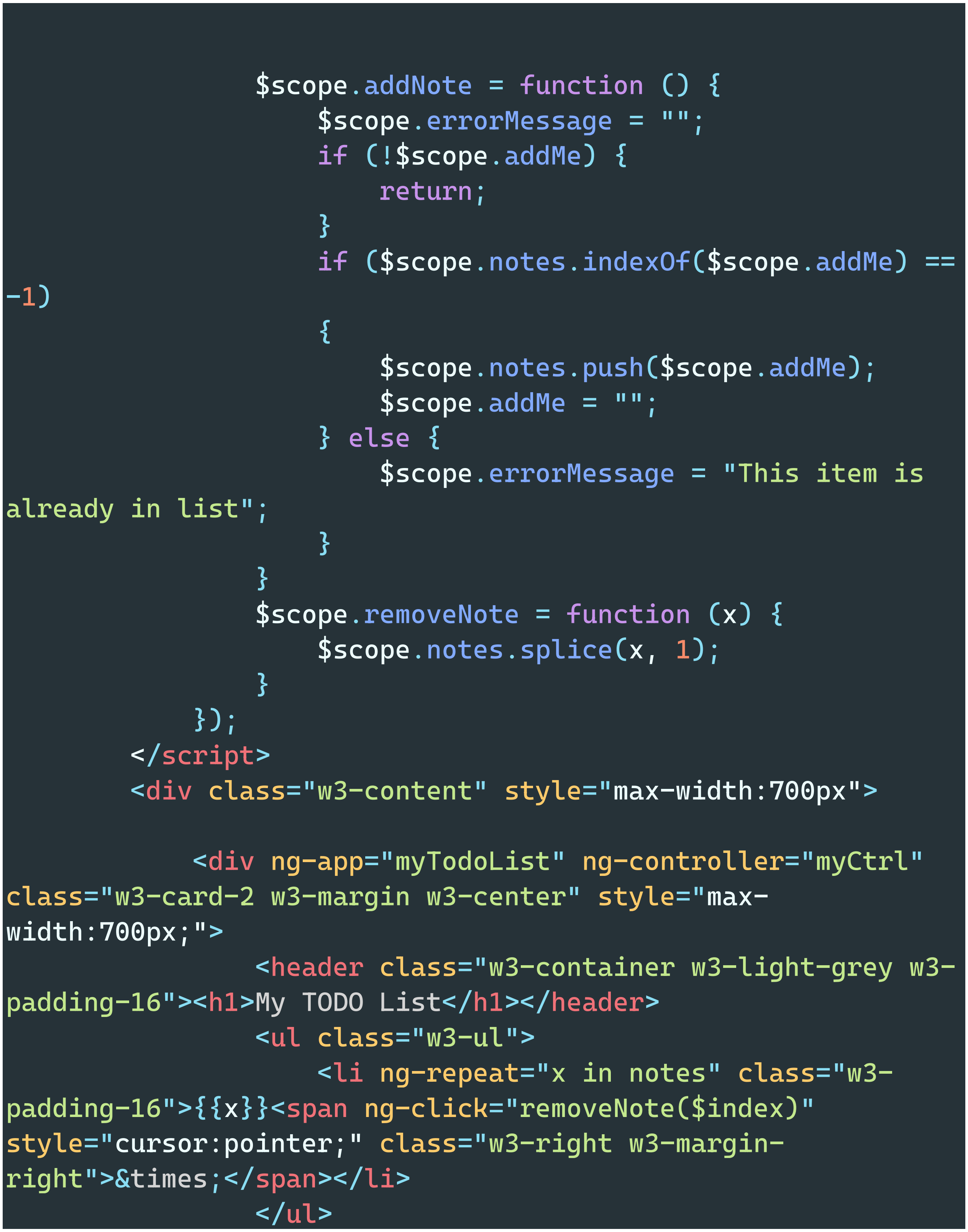
20

TODO Note Example :-

TodoNote.html :-

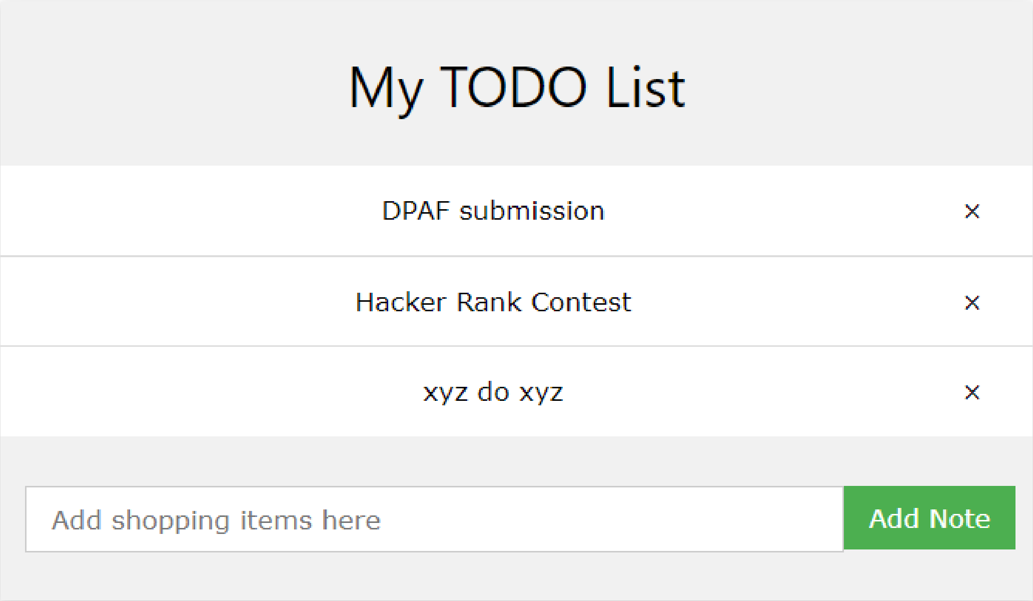


```
<!DOCTYPE html>
<html>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/an
gular.min.js"></script>
    <link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css"> <!--  same
as bootstrap -->
    <body>

        <script>
            var app = angular.module("myTodoList", []);
            app.controller("myCtrl", function ($scope) {
                $scope.notes = ["DPAF submission", "Created
by IT069", "Hacker Rank Contest"];
```



```
                $scope.addNote = function () {
                    $scope.errorMessage = "";
                    if (!$scope.addMe) {
                        return;
                    }
                    if ($scope.notes.indexOf($scope.addMe) ==
-1)
                    {
                        $scope.notes.push($scope.addMe);
                        $scope.addMe = "";
                    } else {
                        $scope.errorMessage = "This item is
already in list";
                    }
                }
                $scope.removeNote = function (x) {
                    $scope.notes.splice(x, 1);
                }
            });
        </script>
        <div class="w3-content" style="max-width:700px">

            <div ng-app="myTodoList" ng-controller="myCtrl"
class="w3-card-2 w3-margin w3-center" style="max-
width:700px;">
                <header class="w3-container w3-light-grey w3-
padding-16"><h1>My TODO List</h1></header>
                <ul class="w3-ul">
                    <li ng-repeat="x in notes" class="w3-
padding-16">{{x}}<span ng-click="removeNote($index)"
style="cursor:pointer;" class="w3-right w3-margin-
right">&times;</span></li>
                </ul>
                <div class="w3-container w3-light-grey w3-
padding-16">
                    <div class="w3-row w3-margin-top">
                        <div class="w3-col s10">
                            <input placeholder="Add shopping
items here" ng-model="addMe" class="w3-input w3-border w3-
padding">
                        </div>
                        <div class="w3-col s2">
                            <button ng-click="addNote()"
class="w3-btn w3-padding w3-green">Add Note</button>
                        </div>
                    </div>
                    <p class="w3-text-
red">{{errorMessage}}</p>
                </div>
            </div>
        </div>


    </body>
</html>
```

Input / Output :-



My TODO List

DPAF submission ×

Hacker Rank Contest ×

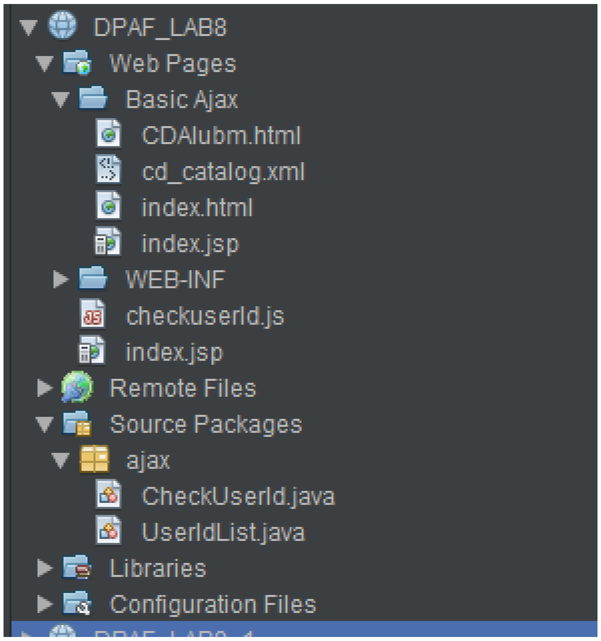
xyz do xyz ×

Add shopping items here

Experiment-8

Aim : Designing and implementing interactive web interface using AJAX

File Structure :-



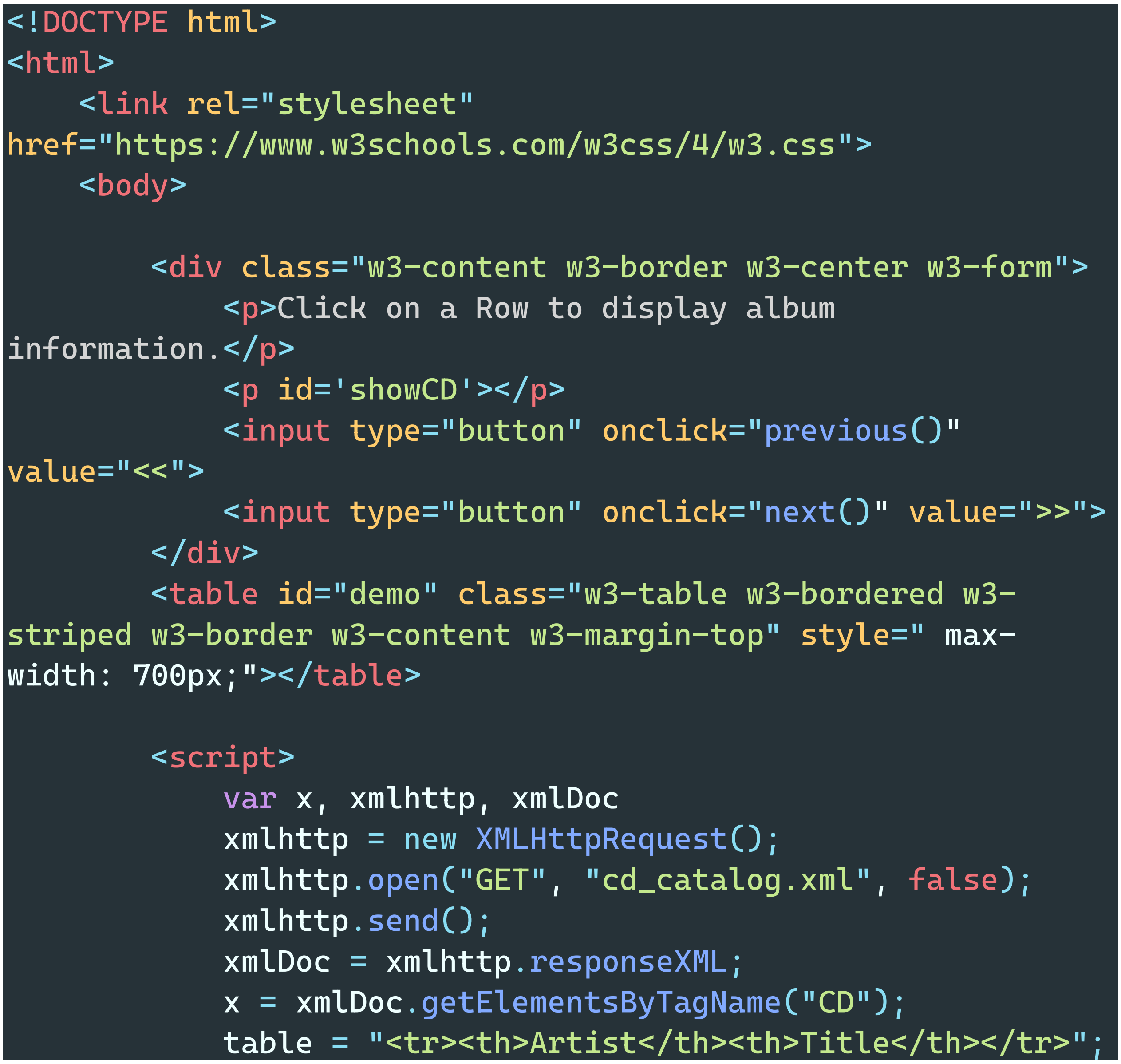
```
▼ DPAF_LAB8
  ▼ Web Pages
    ▼ Basic Ajax
        CDAlubm.html
        cd_catalog.xml
        index.html
        index.jsp
    ▶ WEB-INF
      checkuserId.js
      index.jsp
  ▶ Remote Files
  ▼ Source Packages
    ▼ ajax
        CheckUserId.java
        UserIdList.java
  ▶ Libraries
  ▶ Configuration Files
```

Code :-

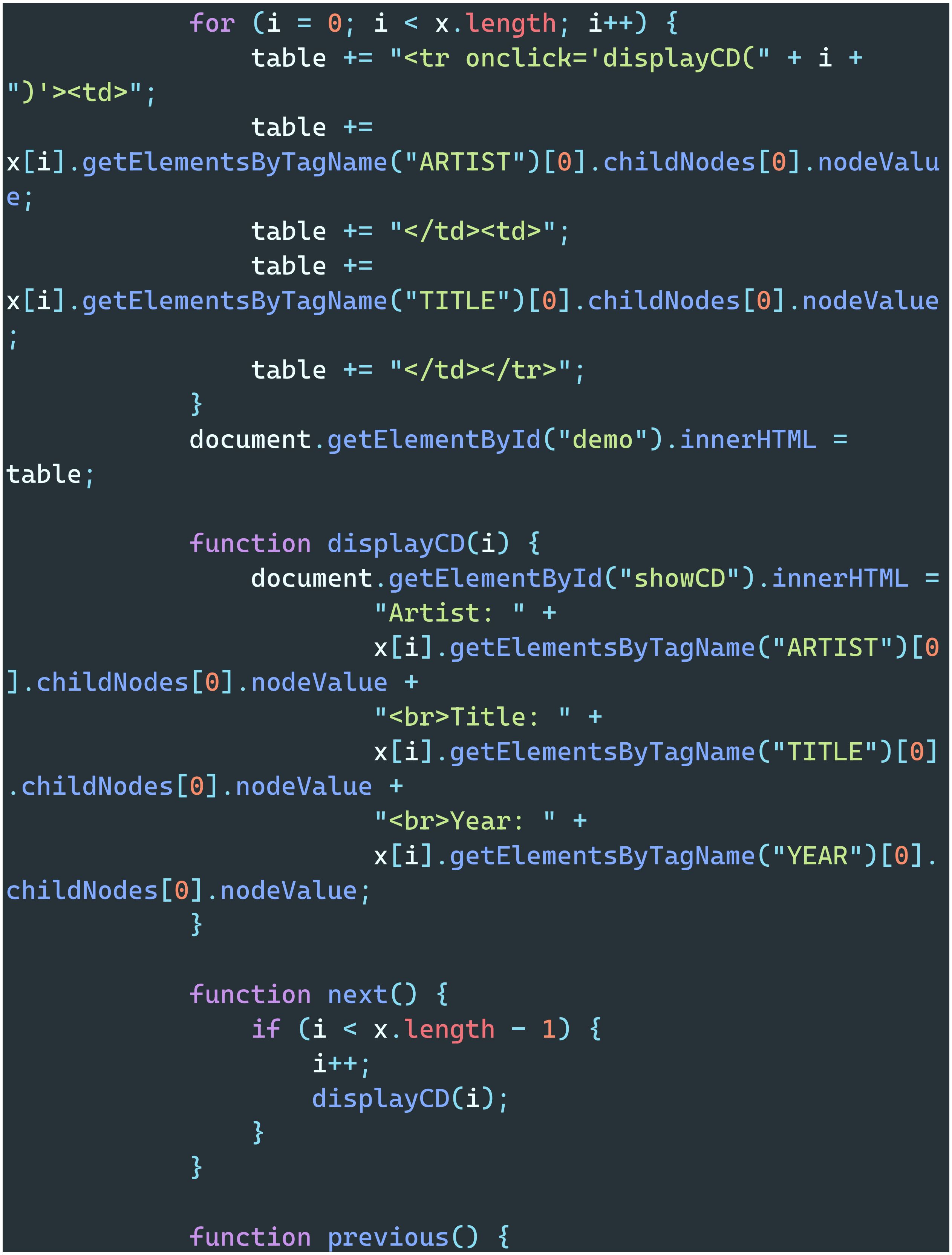
1. **Basic Ajax Examples :-**

CDAlbum.html :-

This will sawing all song albums list in xml files.

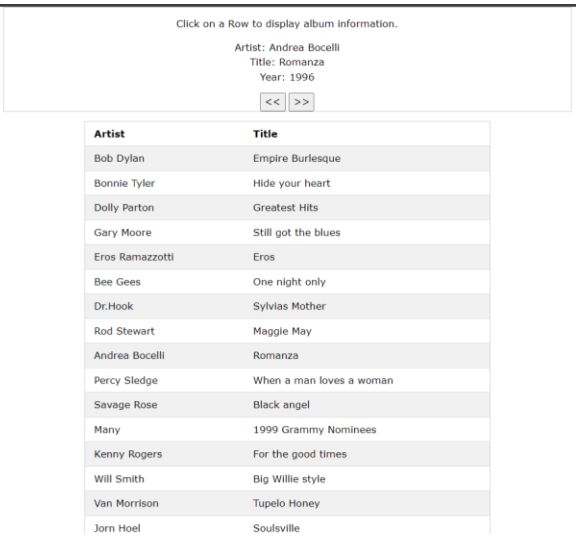


```
<!DOCTYPE html>
<html>
    <link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">
    <body>

        <div class="w3-content w3-border w3-center w3-form">
            <p>Click on a Row to display album
information.</p>
            <p id='showCD'></p>
            <input type="button" onclick="previous()"
value="<<">
            <input type="button" onclick="next()" value=">>">
        </div>
        <table id="demo" class="w3-table w3-bordered w3-
striped w3-border w3-content w3-margin-top" style=" max-
width: 700px;"></table>

        <script>
            var x, xmlhttp, xmlDoc
            xmlhttp = new XMLHttpRequest();
            xmlhttp.open("GET", "cd_catalog.xml", false);
            xmlhttp.send();
            xmlDoc = xmlhttp.responseXML;
            x = xmlDoc.getElementsByTagName("CD");
            table = "<tr><th>Artist</th><th>Title</th></tr>";
```



```
            for (i = 0; i < x.length; i++) {
                table += "<tr onclick='displayCD(" + i +
")'><td>";
                table +=
x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValu
e;
                table += "</td><td>";
                table +=
x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue
;
                table += "</td></tr>";
            }
            document.getElementById("demo").innerHTML =
table;

            function displayCD(i) {
                document.getElementById("showCD").innerHTML =
                        "Artist: " +
                        x[i].getElementsByTagName("ARTIST")[0
].childNodes[0].nodeValue +
                        "<br>Title: " +
                        x[i].getElementsByTagName("TITLE")[0]
.childNodes[0].nodeValue +
                        "<br>Year: " +
                        x[i].getElementsByTagName("YEAR")[0].
childNodes[0].nodeValue;
            }

            function next() {
                if (i < x.length - 1) {
                    i++;
                    displayCD(i);
                }
            }

            function previous() {
                if (i > 0) {
                    i--;
                    displayCD(i);
                }
            }
        </script>

    </body>
</html>
```

Input / Output :-



Click on a Row to display album information.

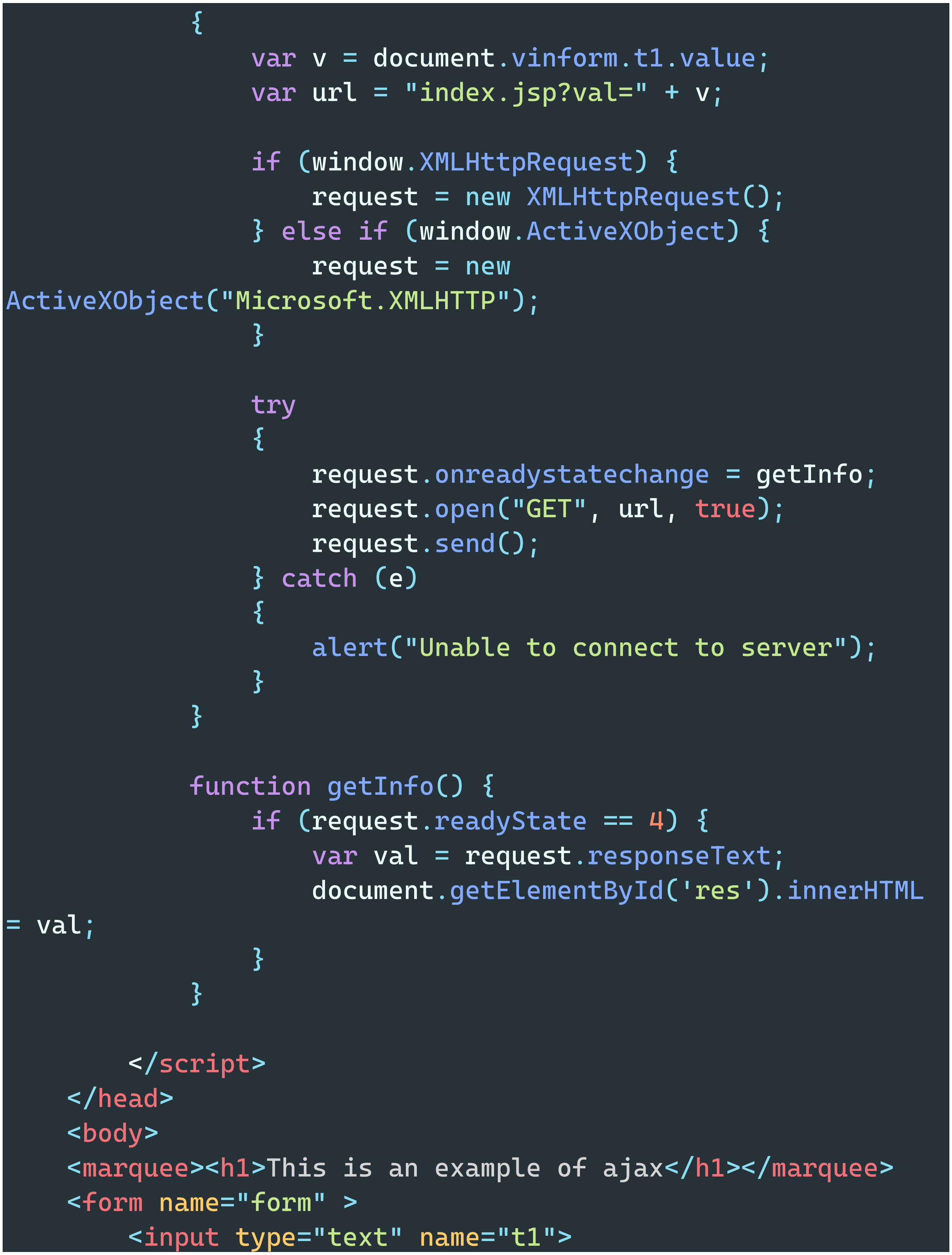
Artist: Andrea Bocelli
Title: Romanza
Year: 1996

Artist	Title
Bob Dylan	Empire Burlesque
Bonnie Tyler	Hide your heart
Dolly Parton	Greatest Hits
Gary Moore	Still got the blues
Eros Ramazzotti	Eros
Bee Gees	One night only
Dr.Hook	Sylvias Mother
Rod Stewart	Maggie May
Andrea Bocelli	Romanza
Percy Sledge	When a man loves a woman
Savage Rose	Black angel
Many	1999 Grammy Nominees
Kenny Rogers	For the good times
Will Smith	Big Willie style
Van Morrison	Tupelo Honey
Jorn Hoel	Soulsville

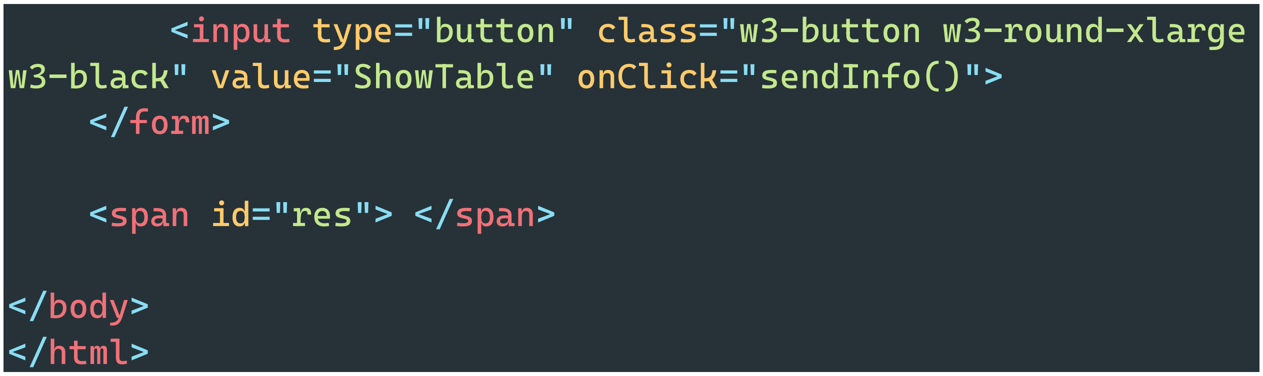
Math table (Dynamic) :-

table.html :-

```
<html>
    <head>
        <link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">
        <script>
            var request;
            function sendInfo()
```

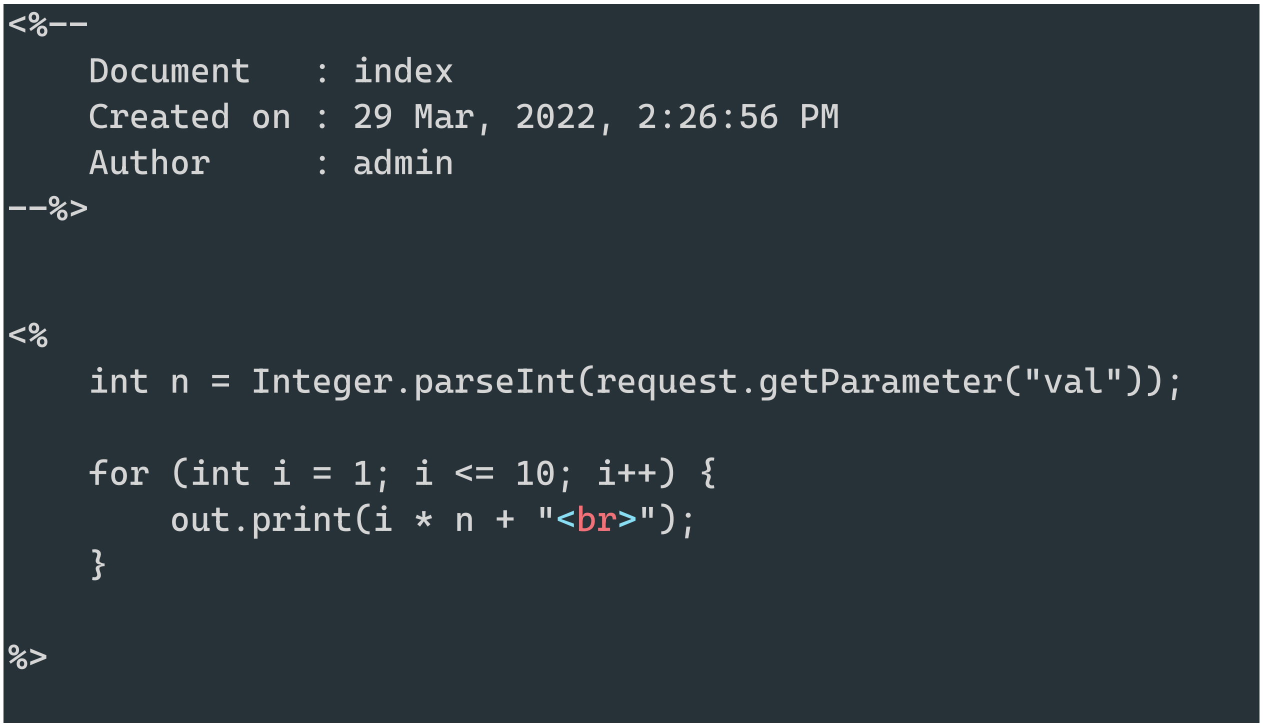


```
            {
                var v = document.vinform.t1.value;
                var url = "index.jsp?val=" + v;

                if (window.XMLHttpRequest) {
                    request = new XMLHttpRequest();
                } else if (window.ActiveXObject) {
                    request = new
ActiveXObject("Microsoft.XMLHTTP");
                }

                try
                {
                    request.onreadystatechange = getInfo;
                    request.open("GET", url, true);
                    request.send();
                } catch (e)
                {
                    alert("Unable to connect to server");
                }
            }

            function getInfo() {
                if (request.readyState == 4) {
                    var val = request.responseText;
                    document.getElementById('res').innerHTML
= val;
                }
            }

        </script>
    </head>
    <body>
    <marquee><h1>This is an example of ajax</h1></marquee>
    <form name="form" >
        <input type="text" name="t1">
```



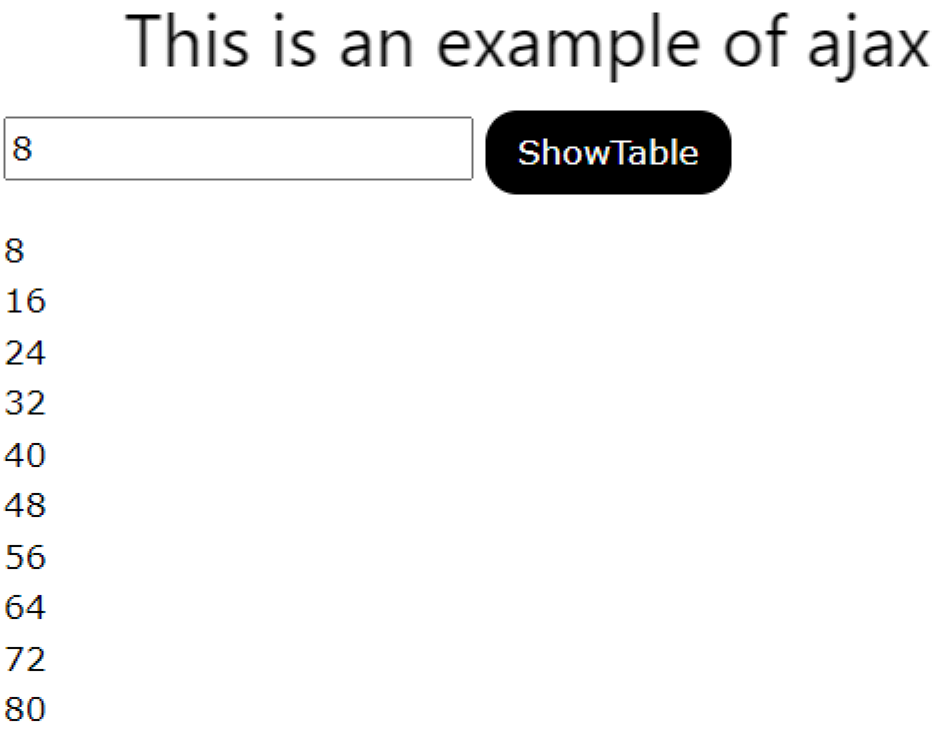
```
        <input type="button" class="w3-button w3-round-xlarge
w3-black" value="ShowTable" onClick="sendInfo()">
    </form>

    <span id="res"> </span>

</body>
</html>
```

Index.jsp :-



```
<%--
    Document   : index
    Created on : 29 Mar, 2022, 2:26:56 PM
    Author     : admin
--%>


<%
    int n = Integer.parseInt(request.getParameter("val"));

    for (int i = 1; i <= 10; i++) {
        out.print(i * n + "<br>");
    }

%>
```

Input / Output :-



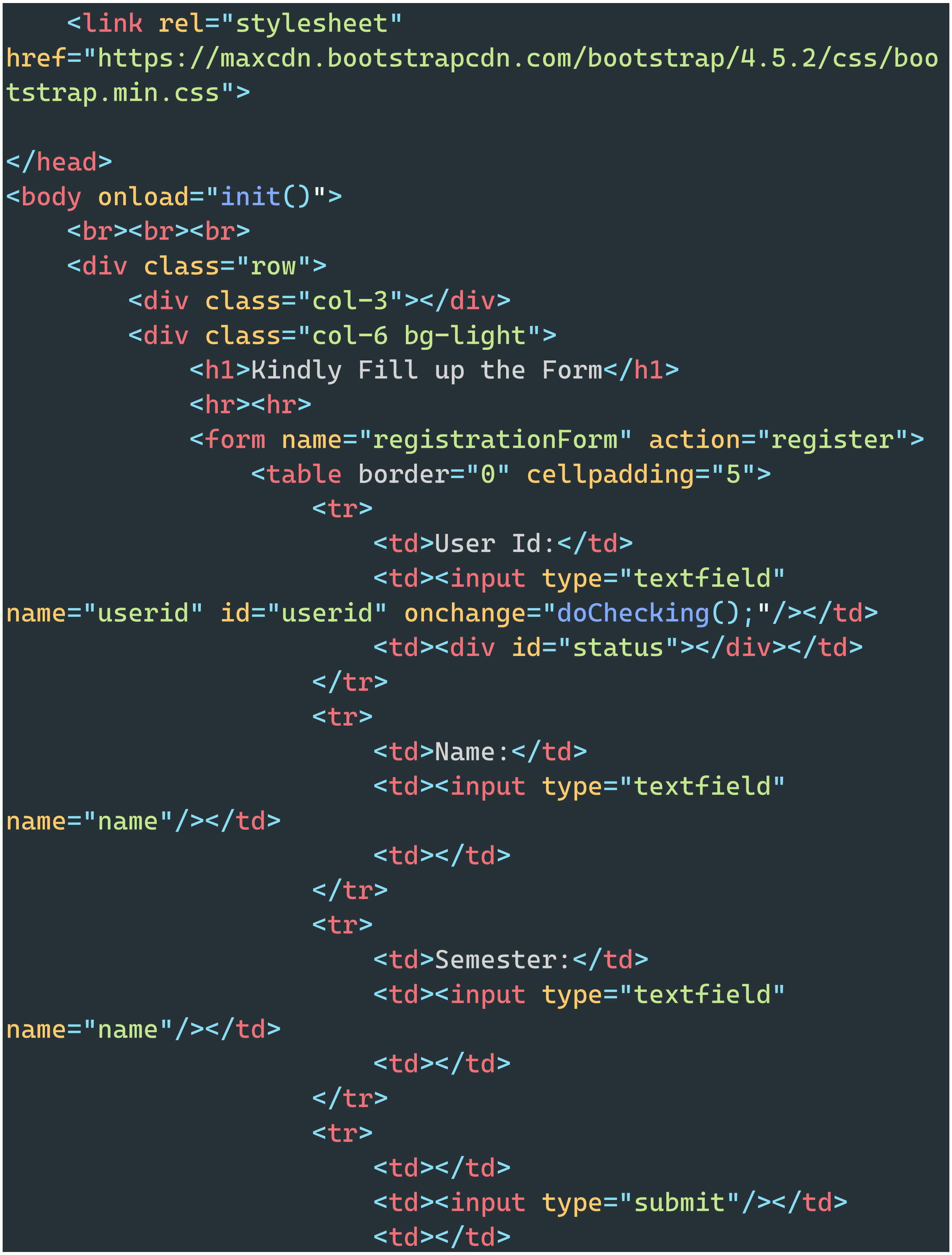
This is an example of ajax

```
8
16
24
32
40
48
56
64
72
80
```

2. **Check userId is unique or not (Ajax + jQuery + javascript + Bootstrap + java + servlet)**

   Index.html :-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<head>
    <script type="text/javascript"
src="checkuserId.js"></script>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <title>UniqueUser</title>
```



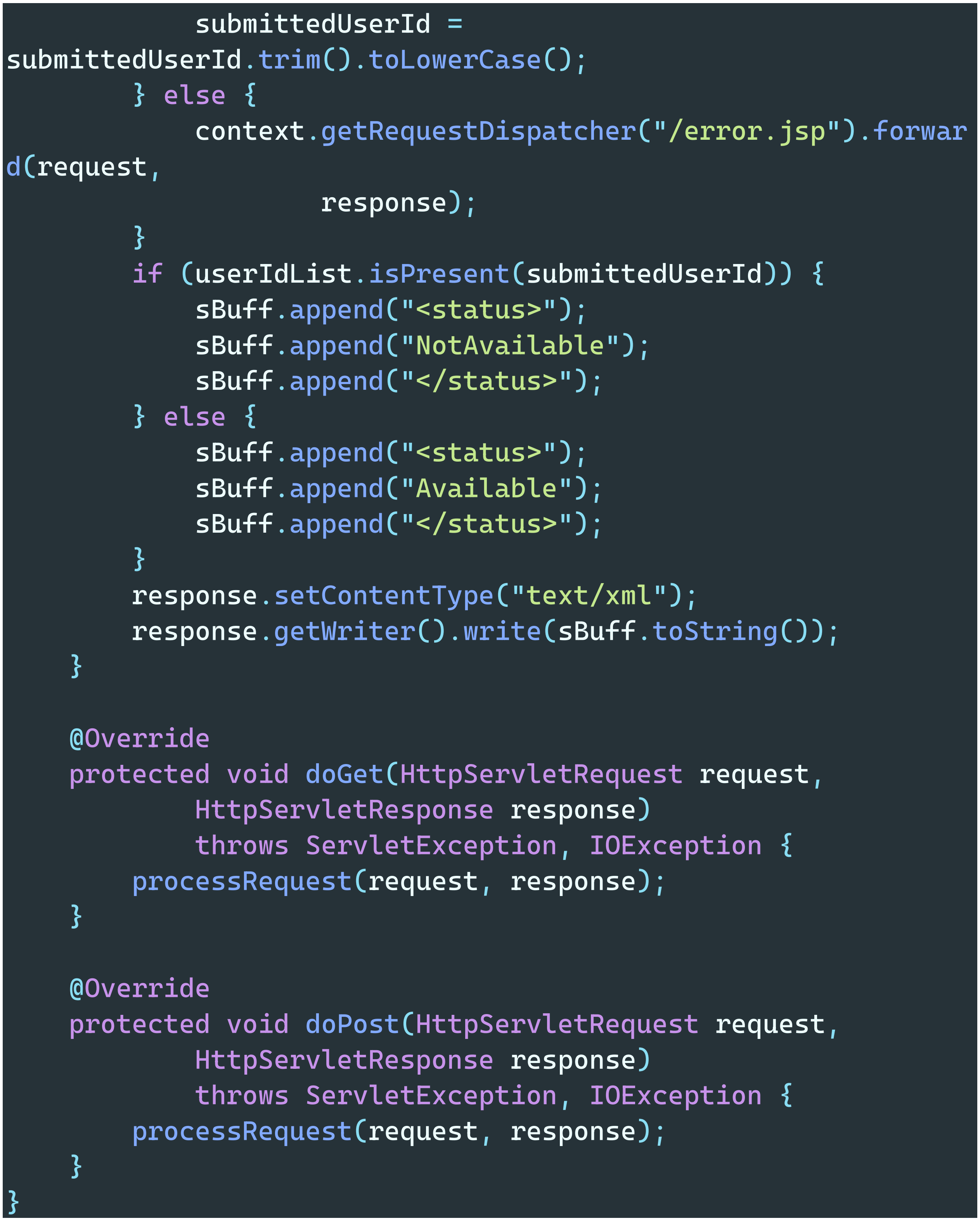
```
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/boo
tstrap.min.css">

</head>
<body onload="init()">
    <br><br><br>
    <div class="row">
        <div class="col-3"></div>
        <div class="col-6 bg-light">
            <h1>Kindly Fill up the Form</h1>
            <hr><hr>
            <form name="registrationForm" action="register">
                <table border="0" cellpadding="5">
                    <tr>
                        <td>User Id:</td>
                        <td><input type="textfield"
name="userid" id="userid" onchange="doChecking();"/></td>
                        <td><div id="status"></div></td>
                    </tr>
                    <tr>
                        <td>Name:</td>
                        <td><input type="textfield"
name="name"/></td>
                        <td></td>
                    </tr>
                    <tr>
                        <td>Semester:</td>
                        <td><input type="textfield"
name="name"/></td>
                        <td></td>
                    </tr>
                    <tr>
                        <td></td>
                        <td><input type="submit"/></td>
                        <td></td>
                    </tr>
                </table>
            </form>
        </div>
        <div class="col-3"></div>
    </div>
</body>
</html>
```

checkedUserId.js :-

```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */


var xhr;
var useridField;
var statusField;
function init() {
    useridField = document.getElementById("userid");
    statusField = document.getElementById("status");
}
function doChecking() {
    var url = "CheckUserId?userid=" +
escape(useridField.value);
    xhr = createXHRObject();
    xhr.open("GET", url, true);
    xhr.onreadystatechange = callback;
    xhr.send(null);
}
function createXHRObject() {
    var xhrObject;
    try {
        // Opera 8.0+, Firefox, Safari
        xhrObject = new XMLHttpRequest();
    } catch (e) {
        // Internet Explorer Browsers
        try {
            xhrObject = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e) {
            try {
                xhrObject = new
ActiveXObject("Microsoft.XMLHTTP");
            } catch (e) {
                // Something went wrong
                alert("Your browser broke!");
                return false;
            }
        }
    }
    return xhrObject;
}
function callback() {
    if (xhr.readyState == 4) {
        if (xhr.status == 200) {
            setMessaage(xhr.responseXML);
        }
    }
}
function setMessaage(responseXML) {
    var statusElement =
responseXML.getElementsByTagName("status")[0];
    var status = statusElement.childNodes[0].nodeValue;
    if (status == "NotAvailable") {
        statusField.innerHTML = "<font color=red>This User Id
is already exist. Please try with different.</font>";
    } else {
        statusField.innerHTML = "<font color=green>This is 
valid UserID.</font>";
    }
}
```

CheckUserId.java : -

```
package ajax;

import java.io.IOException;
import javax.servlet.ServletConfig;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class CheckUserId extends HttpServlet {

    private UserIdList userIdList = new UserIdList();
    private ServletContext context;

    @Override
    public void init(ServletConfig config) throws 
ServletException {
        this.context = config.getServletContext();
    }

    protected void processRequest(HttpServletRequest request, 
HttpServletResponse response) throws
            ServletException, IOException {
        String submittedUserId = 
request.getParameter("userid");
        StringBuffer sBuff = new StringBuffer();
        if (submittedUserId != null) {
```

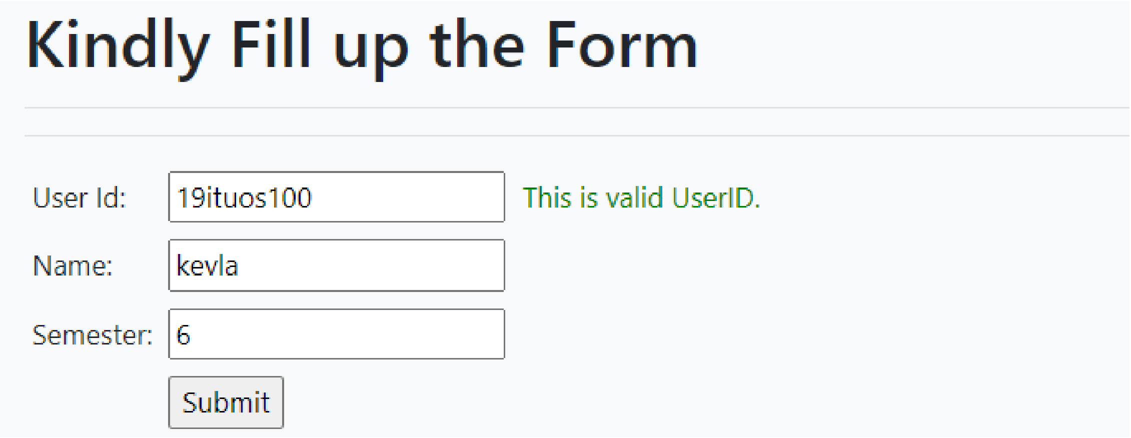


```
            submittedUserId =
submittedUserId.trim().toLowerCase();
        } else {
            context.getRequestDispatcher("/error.jsp").forwar
d(request,
                    response);
        }
        if (userIdList.isPresent(submittedUserId)) {
            sBuff.append("<status>");
            sBuff.append("NotAvailable");
            sBuff.append("</status>");
        } else {
            sBuff.append("<status>");
            sBuff.append("Available");
            sBuff.append("</status>");
        }
        response.setContentType("text/xml");
        response.getWriter().write(sBuff.toString());
    }

    @Override
    protected void doGet(HttpServletRequest request,
            HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request,
            HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
}
```

UserIdList.java :-

```java
package ajax;

import java.util.ArrayList;

public class UserIdList {

    private final ArrayList<String> userIdList = new ArrayList<>();

    public ArrayList<String> getList() {
        return userIdList;
    }

    public UserIdList() {
        userIdList.add("arjun@gmail.com");
        userIdList.add("kevla@gmail.com");
        userIdList.add("tarang");
        userIdList.add("rohit");
    }

    public void addId(String userId) {
        userIdList.add(userId);
    }

    public boolean isPresent(String userId) {
        boolean status = false;
        for (String id : userIdList) {
            if (id.compareTo(userId) == 0) {
                status = true;
                break;
            }
        }
        return status;
    }
}
```

Input / Output :-

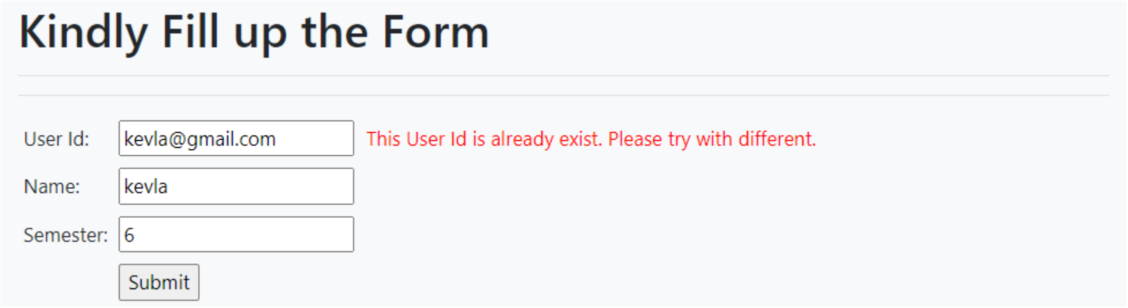


Kindly Fill up the Form

User Id: 19ituos100 This is valid UserID.

Name: kevla

Semester: 6



Kindly Fill up the Form

User Id: kevla@gmail.com This User Id is already exist. Please try with different.

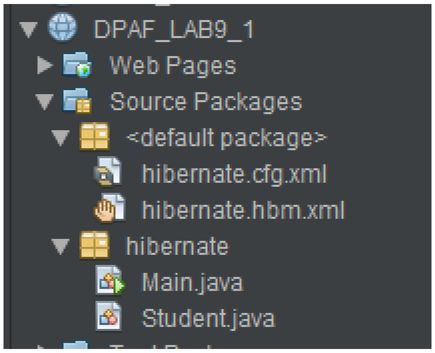
Name: kevla

Semester: 6

Experiment-9

Aim : Implement data access using Hibernate

File Structure :-



```
DPAF_LAB9_1
  Web Pages
  Source Packages
    <default package>
      hibernate.cfg.xml
      hibernate.hbm.xml
    hibernate
      Main.java
      Student.java
```

Code :-

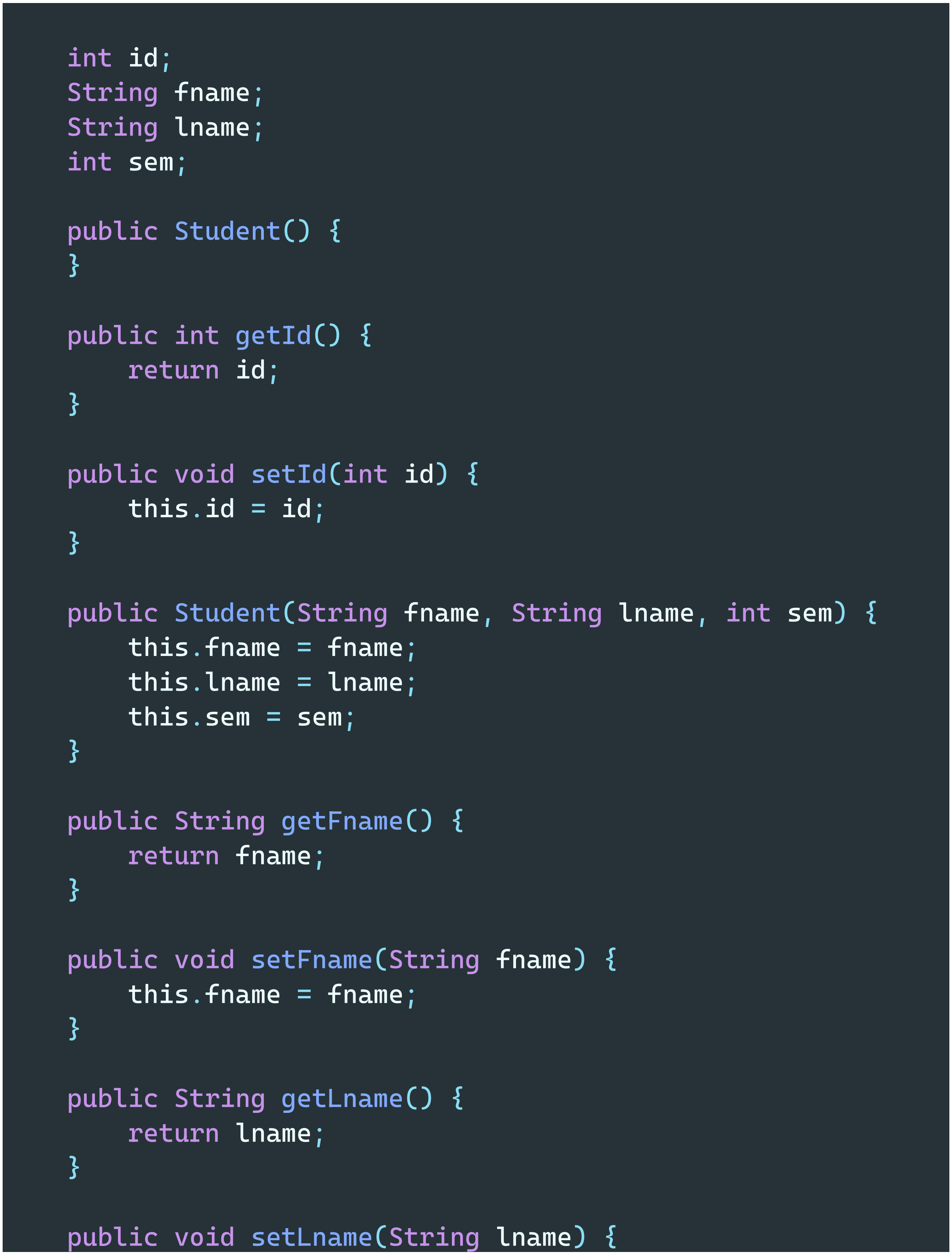
Main.java :-

```
package hibernate;
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author IT069 arjun
 */
```

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

public class Main {

    public static void main(String[] args) {
        Student st = new Student("arjun", "maniya", 6);
        SessionFactory sessionFactory =new
Configuration().configure().buildSessionFactory();
        Session session = sessionFactory.openSession();
        Transaction tx = session.beginTransaction();
        session.save(st);
        tx.commit();
        session.close();
        sessionFactory.close();
    }
}
```

Student.java :-

```
package hibernate;
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author IT069 arjun
 */
public class Student {
```

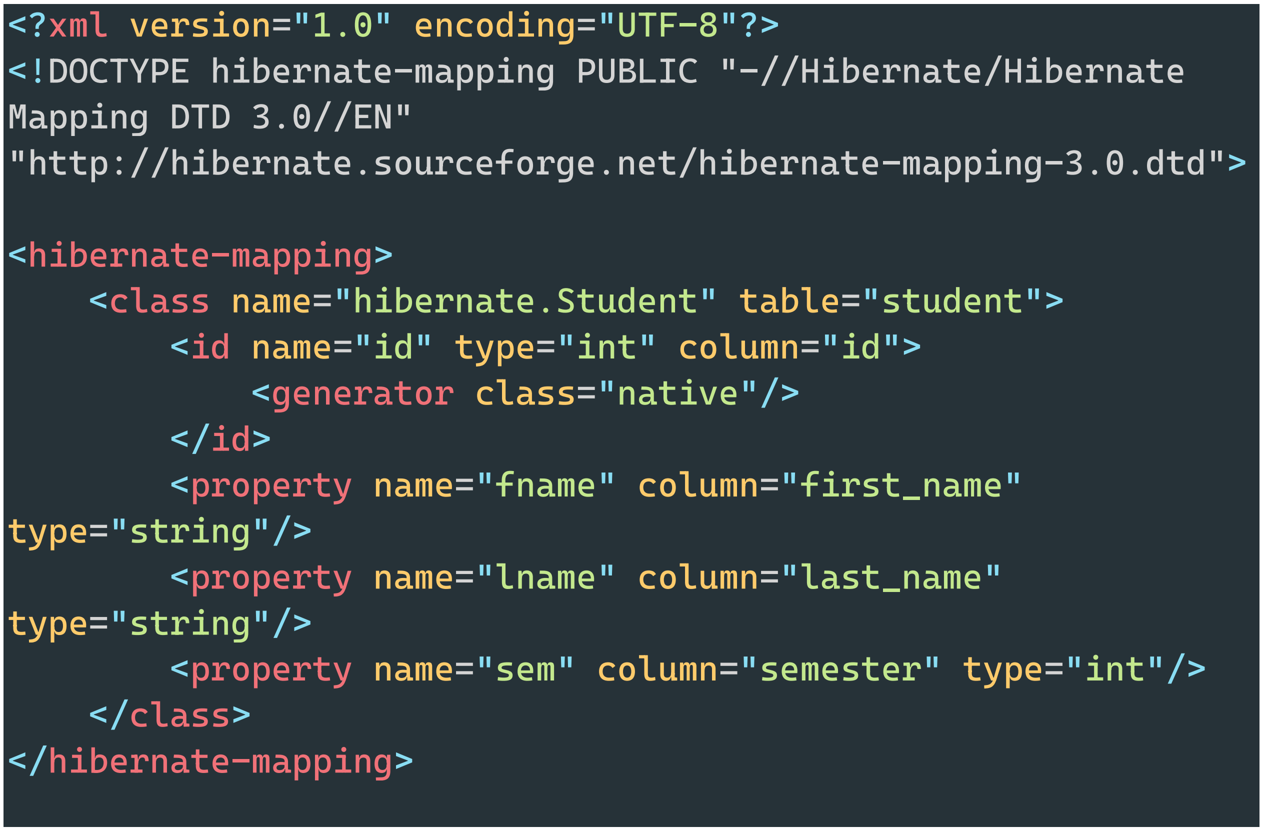


```
    int id;
    String fname;
    String lname;
    int sem;

    public Student() {
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public Student(String fname, String lname, int sem) {
        this.fname = fname;
        this.lname = lname;
        this.sem = sem;
    }

    public String getFname() {
        return fname;
    }

    public void setFname(String fname) {
        this.fname = fname;
    }

    public String getLname() {
        return lname;
    }

    public void setLname(String lname) {
        this.lname = lname;
    }

    public int getSem() {
        return sem;
    }

    public void setSem(int sem) {
        this.sem = sem;
    }

    @Override
    public String toString() {
        return "student{" + "fname=" + fname + ", lname=" + 
lname + ", sem=" + sem + '}';
    }

}
```

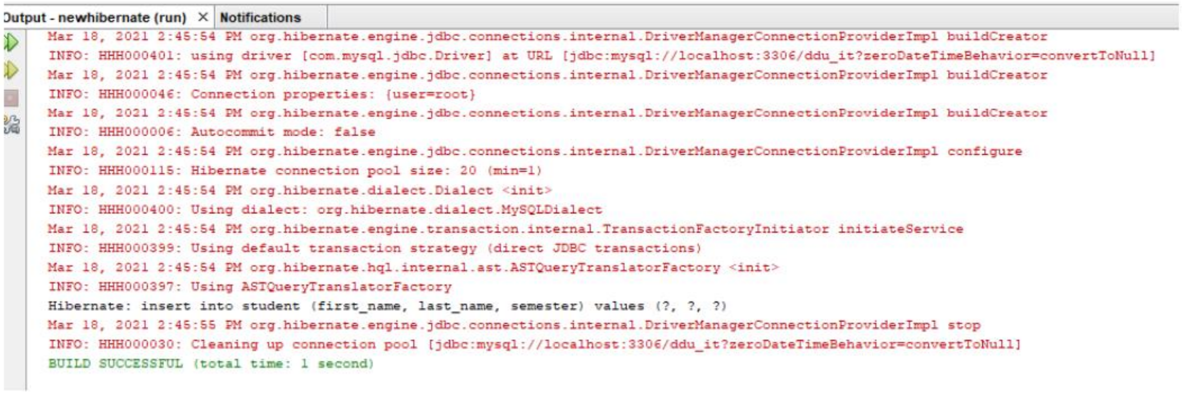
Hibernate.cfg.xml :-

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-
//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-
3.0.dtd">

<hibernate-configuration>
  <session-factory>
    <mapping resource="hibernate.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```

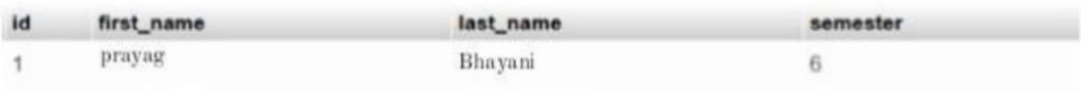
Hibernate.hbm.xml :-



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate
Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
    <class name="hibernate.Student" table="student">
        <id name="id" type="int" column="id">
            <generator class="native"/>
        </id>
        <property name="fname" column="first_name"
type="string"/>
        <property name="lname" column="last_name"
type="string"/>
        <property name="sem" column="semester" type="int"/>
    </class>
</hibernate-mapping>
```



```
Mar 18, 2021 2:45:54 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH000401: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/ddu_it?zeroDateTimeBehavior=convertToNull]
Mar 18, 2021 2:45:54 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH000046: Connection properties: {user=root}
Mar 18, 2021 2:45:54 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH000006: Autocommit mode: false
Mar 18, 2021 2:45:54 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Mar 18, 2021 2:45:54 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Mar 18, 2021 2:45:54 PM org.hibernate.engine.transaction.internal.TransactionFactoryInitiator initiateService
INFO: HHH000399: Using default transaction strategy (direct JDBC transactions)
Mar 18, 2021 2:45:54 PM org.hibernate.hql.internal.ast.ASTQueryTranslatorFactory <init>
INFO: HHH000397: Using ASTQueryTranslatorFactory
Hibernate: insert into student (first_name, last_name, semester) values (?, ?, ?)
Mar 18, 2021 2:45:55 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl stop
INFO: HHH000030: Cleaning up connection pool [jdbc:mysql://localhost:3306/ddu_it?zeroDateTimeBehavior=convertToNull]
BUILD SUCCESSFUL (total time: 1 second)
```

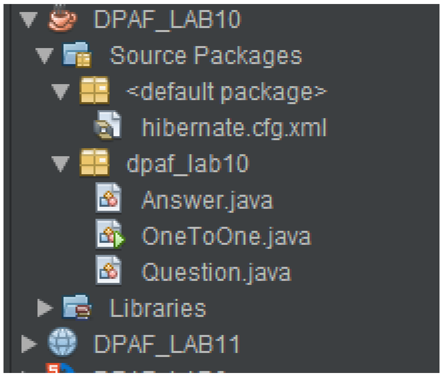


id	first_name	last_name	semester
1	prayag	Bhayani	6

Experiment 10

Aim : : Implement relational data access using Hibernate.

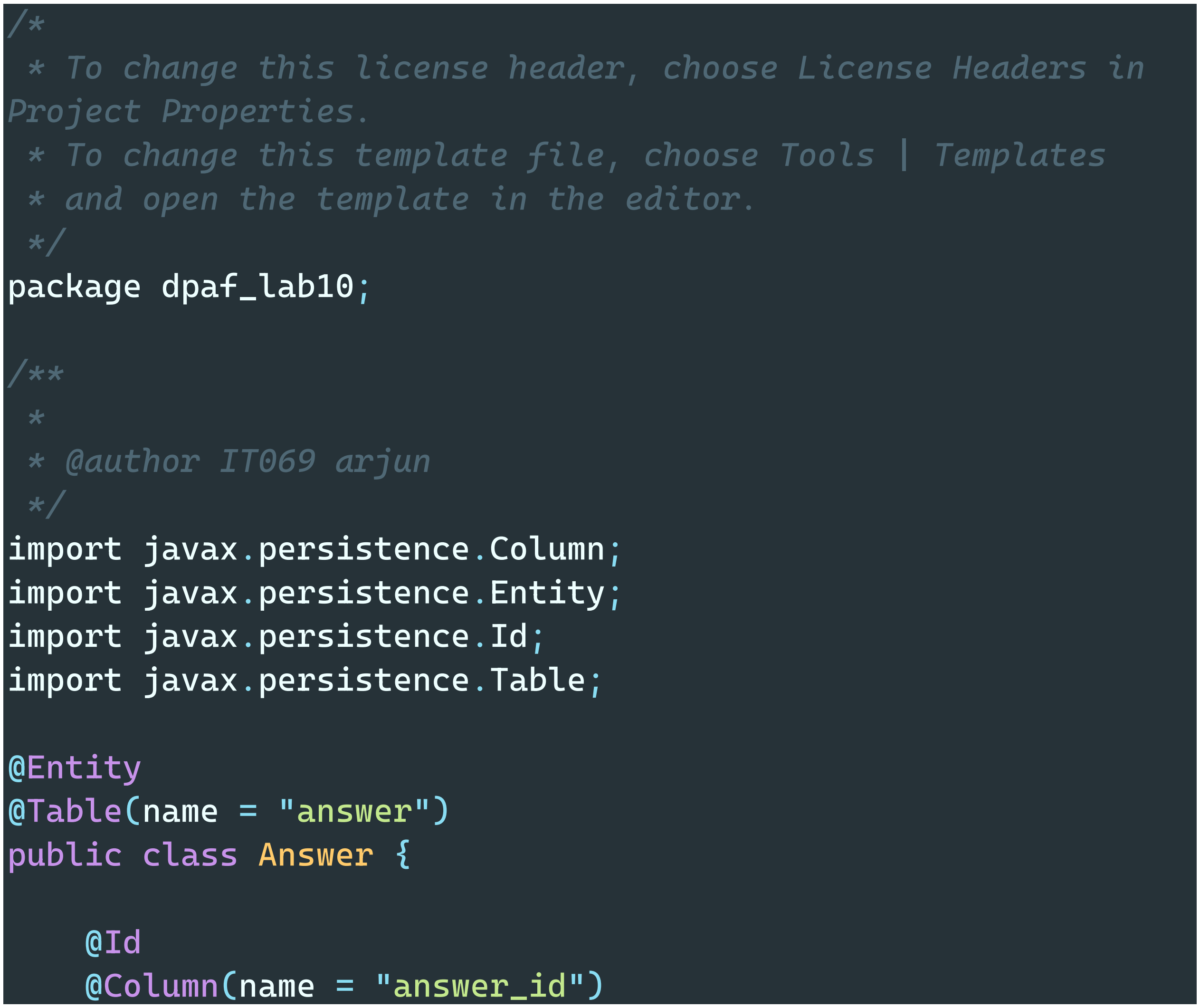
File Structure :-



```
DPAF_LAB10
  Source Packages
    <default package>
      hibernate.cfg.xml
    dpaf_lab10
      Answer.java
      OneToOne.java
      Question.java
  Libraries
DPAF_LAB11
```

Code :-

Answer.java :-



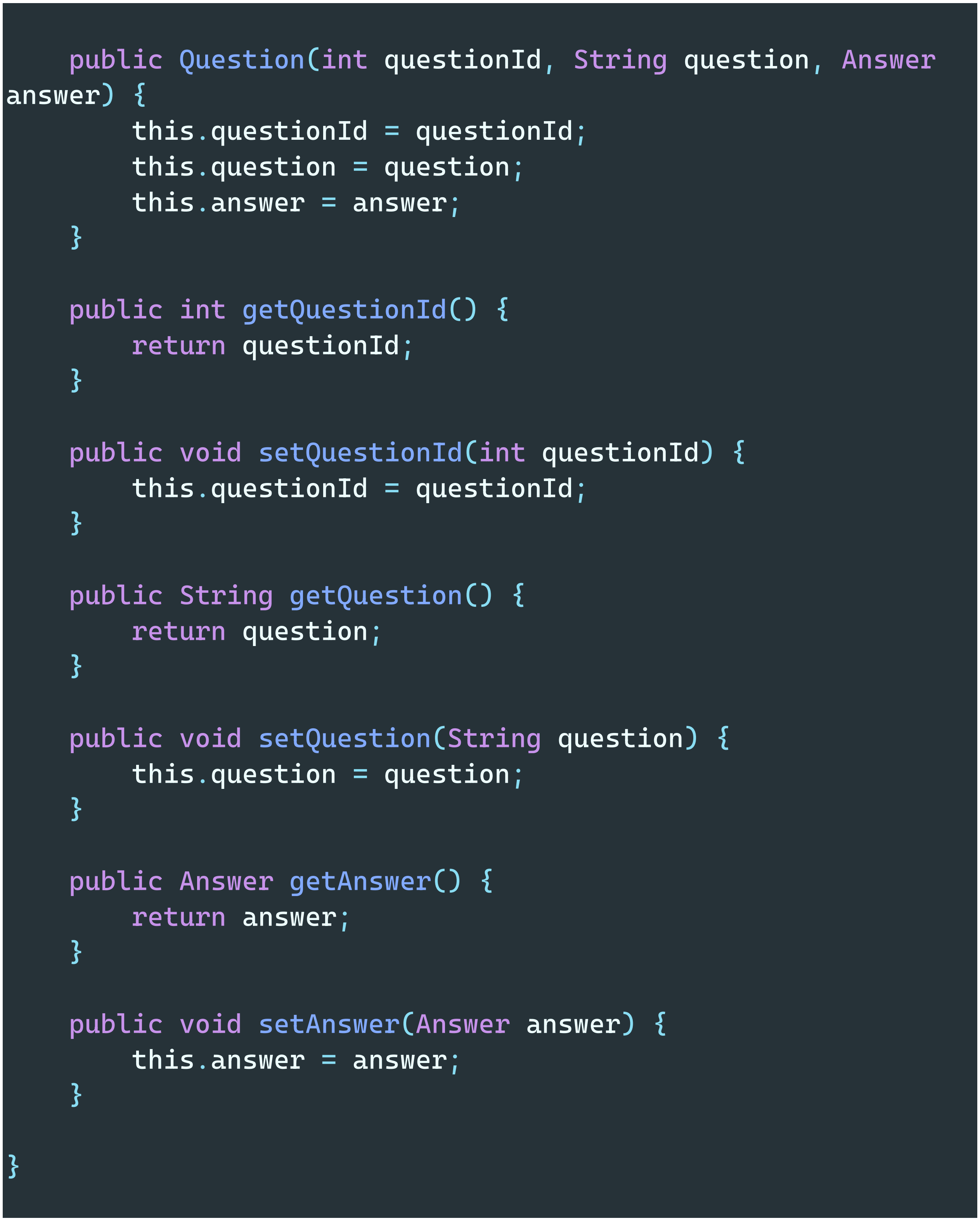
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dpaf_lab10;

/**
 *
 * @author IT069 arjun
 */
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "answer")
public class Answer {

    @Id
    @Column(name = "answer_id")
    private int answerId;
    private String answer;

    public Answer(int answerId, String answer) {
        this.answerId = answerId;
        this.answer = answer;
    }

    public Answer() {
    }

    public int getAnswerId() {
        return answerId;
    }

    public void setAnswerId(int answerId) {
        this.answerId = answerId;
    }

    public String getAnswer() {
        return answer;
    }

    public void setAnswer(String answer) {
        this.answer = answer;
    }
}
```

OneToOne.java :-

```
/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dpaf_lab10;

/**
 *
 * @author IT069 arjun
 */
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

public class OneToOne {

    public static void main(String[] args) {

        Answer answer1 = new Answer(25, "Java is Programming
Language");
        Question q1 = new Question(1, "What is Java?",
answer1);
        Answer answer2 = new Answer(27, "API work with object
in java");
        Question q2 = new Question(2, "What is Collection
Framework?", answer2);

        SessionFactory sessionFactory = new
Configuration().configure().buildSessionFactory();
        Session session = sessionFactory.openSession();
        Transaction tx = session.beginTransaction();
        session.save(q1);
        session.save(q2);
        session.save(answer1);
        session.save(answer2);
        tx.commit();
        session.close();
        sessionFactory.close();
    }

}
```

Qustions.java :-

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dpaf_lab10;

/**
 *
 * @author IT069 arjun
 */
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToOne;

@Entity
public class Question {

    @Id
    @Column(name = "question_id")
    private int questionId;
    private String question;
    @OneToOne
    private Answer answer;

    public Question() {
    }
```

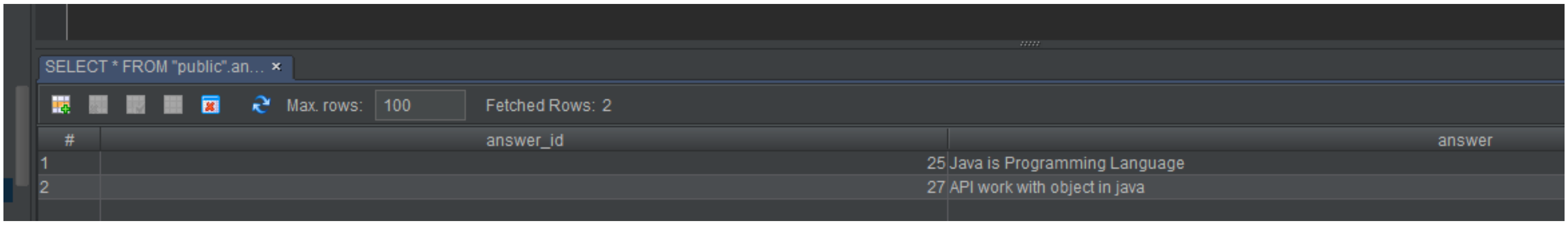


```
    public Question(int questionId, String question, Answer
answer) {
        this.questionId = questionId;
        this.question = question;
        this.answer = answer;
    }

    public int getQuestionId() {
        return questionId;
    }

    public void setQuestionId(int questionId) {
        this.questionId = questionId;
    }

    public String getQuestion() {
        return question;
    }

    public void setQuestion(String question) {
        this.question = question;
    }

    public Answer getAnswer() {
        return answer;
    }

    public void setAnswer(Answer answer) {
        this.answer = answer;
    }

}
```

Hibernet.cfg.xml :-

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
    <session-factory>
        <property name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
        <property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
        <property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/postgres</property>
        <property name="hibernate.connection.username">postgres</property>
        <property name="hibernate.connection.password">shree9592</property>
        <property name="hibernate.hbm2ddl.auto">create</property>
        <property name="hibernate.show_sql">true</property>
        <mapping class="dpaf_lab10.Answer"/>
        <mapping class="dpaf_lab10.Question"/>

    </session-factory>
</hibernate-configuration>
```

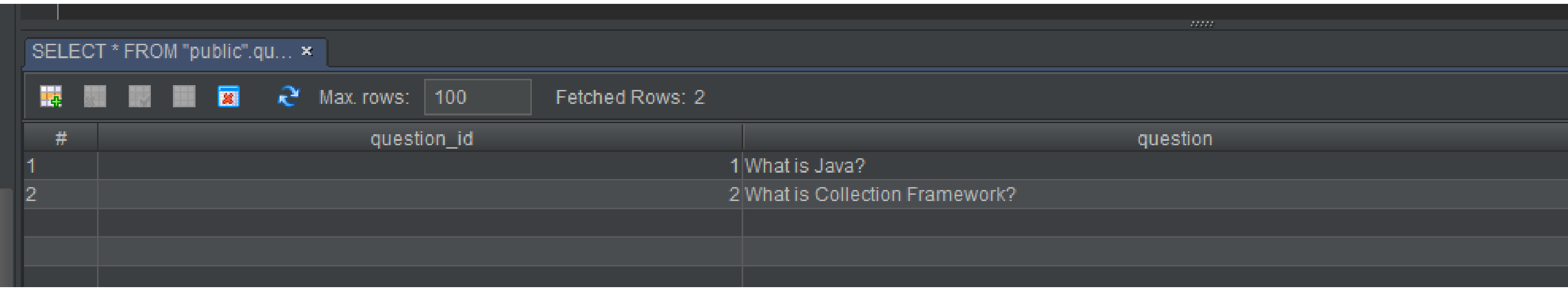
Input / Output :-



SELECT * FROM "public".an... ×

Max. rows: 100 Fetched Rows: 2

#	answer_id	answer
1	25	Java is Programming Language
2	27	API work with object in java



SELECT * FROM "public".qu... ×

Max. rows: 100 Fetched Rows: 2

#	question_id	question
1	1	What is Java?
2	2	What is Collection Framework?

Experiment-11

Aim : Implement form handling with data binding using

Code :-

Student.java

```
public class Student {
 private Integer age;
 private String name;
 private Integer id;
 public void setAge(Integer age) {
 this.age = age;
 }
 public Integer getAge() {
 return age;
 }
 public void setName(String name) {
 this.name = name;
 }
 public String getName() {
 return name;
 }
 public void setId(Integer id) {
 this.id = id;
 }
 public Integer getId() {
 return id;
 }
}
StudentController.java
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.ui.ModelMap;
@Controller
public class StudentController {
 @RequestMapping(value = "/student", method = RequestMethod.GET)
 public ModelAndView student() {
 return new ModelAndView("student", "command", new Student());
 }

 @RequestMapping(value = "/addStudent", method = RequestMethod.POST)
 public String addStudent(@ModelAttribute("SpringWeb")Student student,
 ModelMap model) {
 model.addAttribute("name", student.getName());
 model.addAttribute("age", student.getAge());
 model.addAttribute("id", student.getId());


 return "result";
 }
}
```

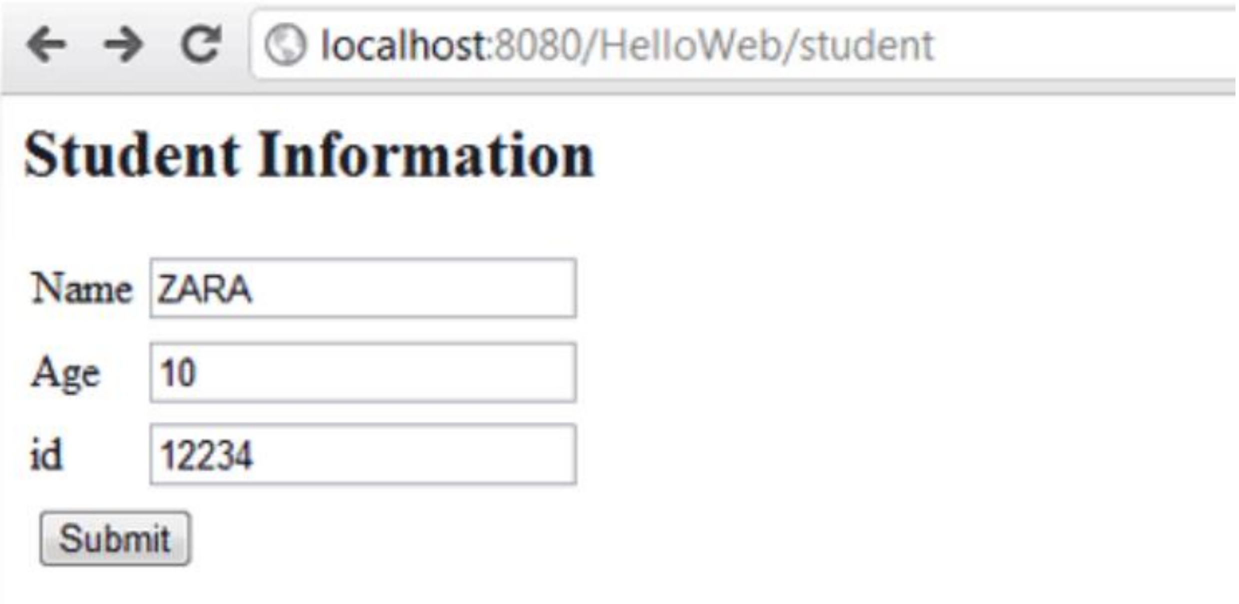
Student.jsp

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
 <title>Spring MVC Form Handling</title>
</head>
<body>
<h2>Student Information</h2>
<form:form method="POST" action="/HelloWeb/addStudent">
 <table>
 <tr>
 <td><form:label path="name">Name</form:label></td>
 <td><form:input path="name" /></td>
 </tr>
 <tr>
 <td><form:label path="age">Age</form:label></td>
 <td><form:input path="age" /></td>
 </tr>
 <tr>
 <td><form:label path="id">id</form:label></td>
 <td><form:input path="id" /></td>
 </tr>
 <tr>
 <td colspan="2">
 <input type="submit" value="Submit"/>
 </td>
 </tr>
</table>
</form:form>
</body>
</html>
```

result.jsp

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
 <title>Spring MVC Form Handling</title>
</head>
<body>
<h2>Submitted Student Information</h2>
 <table>
 <tr>
 <td>Name</td>
 <td>${name}</td>
 </tr>
 <tr>
 <td>Age</td>
 <td>${age}</td>
 </tr>
 <tr>
 <td>ID</td>
 <td>${id}</td>
 </tr>
</table>
</body>
</html>
```

Input / output :-



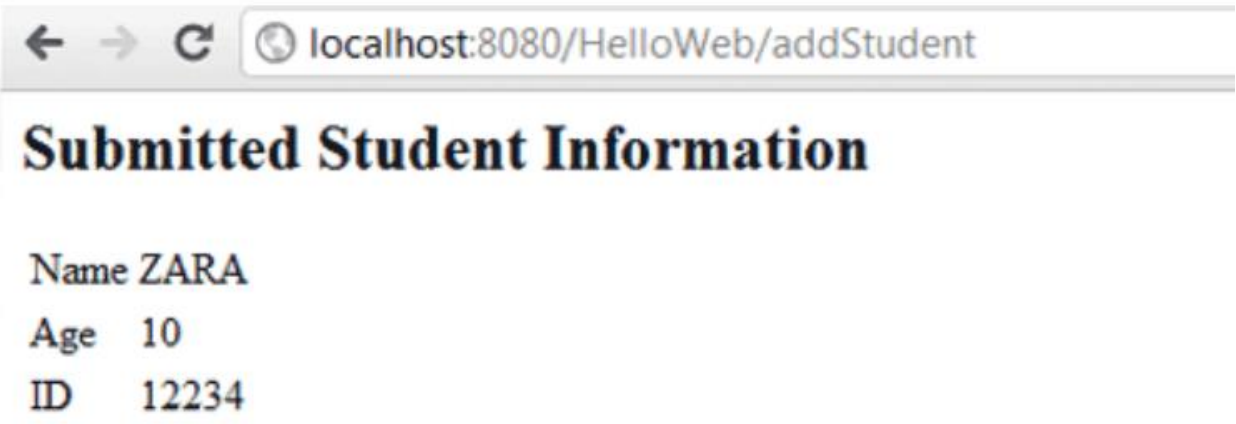
localhost:8080/HelloWeb/student

Student Information

Name ZARA

Age 10

id 12234



localhost:8080/HelloWeb/addStudent

Submitted Student Information

Name ZARA

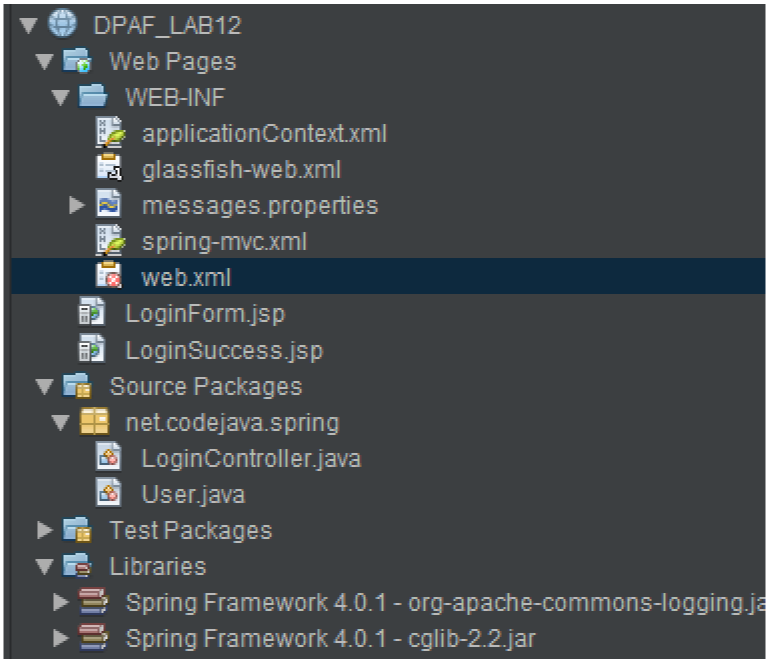
Age 10

ID 12234

Experiment-12

Aim : Implement form handling with form validation using SpringMVC

File Structure :-



```
DPAF_LAB12
  Web Pages
    WEB-INF
      applicationContext.xml
      glassfish-web.xml
      messages.properties
      spring-mvc.xml
      web.xml
    LoginForm.jsp
    LoginSuccess.jsp
  Source Packages
    net.codejava.spring
      LoginController.java
      User.java
  Test Packages
  Libraries
    Spring Framework 4.0.1 - org-apache-commons-logging.ja
    Spring Framework 4.0.1 - cglib-2.2.jar
```

Code :-

LoginController.java :-

```java
package net.codejava.spring;

import java.util.Map;

import javax.validation.Valid;

import org.springframework.stereotype.Controller;
import org.springframework.validation.BindingResult;
import
org.springframework.web.bind.annotation.ModelAttribute;
import
org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

/**
 *
 * @author www.codejava.net
 *
 */
@Controller
public class LoginController {
    @RequestMapping(value = "/login", method =
RequestMethod.GET)
    public String viewLogin(Map<String, Object> model) {
        User user = new User();
        model.put("userForm", user);
        return "LoginForm";
    }

    @RequestMapping(value = "/login", method =
RequestMethod.POST)
    public String doLogin(@Valid @ModelAttribute("userForm")
User userForm,
            BindingResult result, Map<String, Object> model)
{

        if (result.hasErrors()) {
            return "LoginForm";
        }

        return "LoginSuccess";
    }
}
```

User.java :-

```
package net.codejava.spring;

import javax.validation.constraints.Size;

import org.hibernate.validator.constraints.Email;
import org.hibernate.validator.constraints.NotEmpty;

/**
 *
 * @author www.codejava.net
 *
 */
public class User {
    @NotEmpty
    @Email
    private String email;

    @NotEmpty(message = "Please enter your password.")
    @Size(min = 6, max = 15, message = "Your password must
between 6 and 15 characters")
    private String password;

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

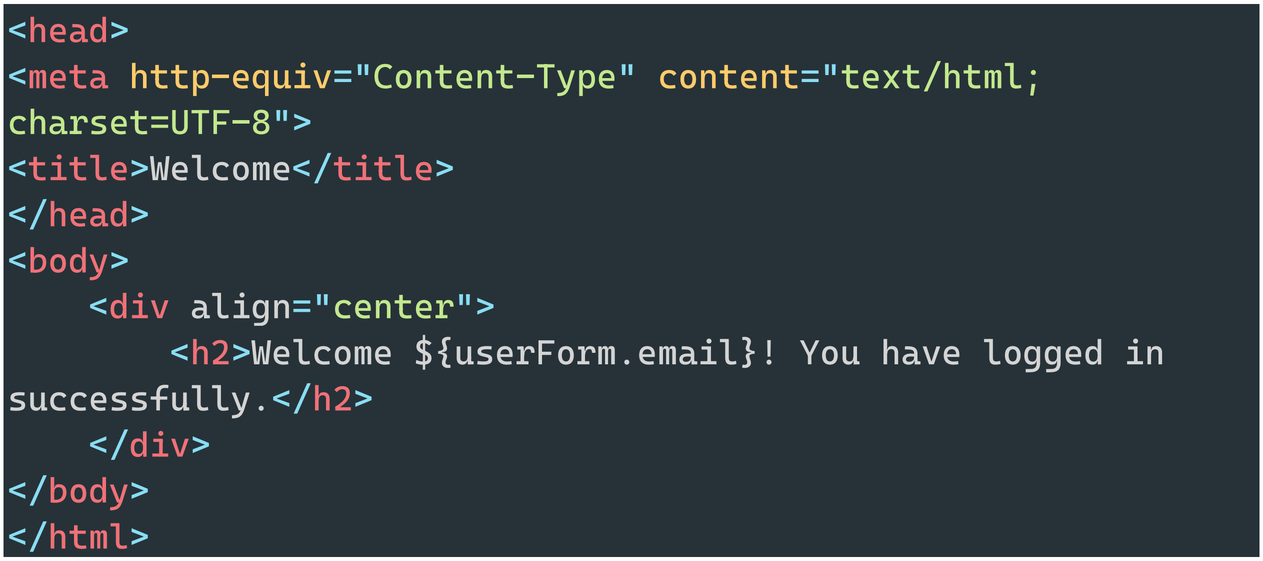
LoginForm.jsp :-

```
<%@ page language="java" contentType="text/html; charset=UTF-
8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="form"
uri="http://www.springframework.org/tags/form"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Login</title>
<style>
    .error {
        color: red; font-weight: bold;
    }
</style>
</head>
<body>
    <div align="center">
        <h2>Spring MVC Form Validation Demo - Login Form</h2>
        <table border="0" width="90%">
        <form:form action="login" commandName="userForm">
```

```
                <tr>
                    <td align="left" width="20%">Email: </td>
                    <td align="left" width="40%"><form:input
path="email" size="30"/></td>
                    <td align="left"><form:errors
path="email" cssClass="error"/></td>
                </tr>
                <tr>
                    <td>Password: </td>
                    <td><form:password path="password"
size="30"/></td>
                    <td><form:errors path="password"
cssClass="error"/></td>
                </tr>
                <tr>
                    <td></td>
                    <td align="center"><input type="submit"
value="Login"/></td>
                    <td></td>
                </tr>
        </form:form>
        </table>
    </div>
</body>
</html>
```

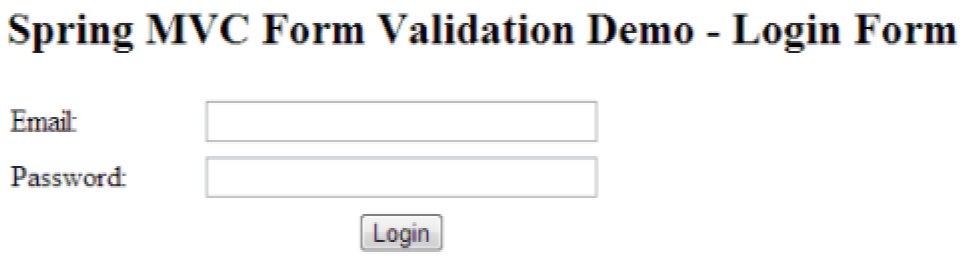
LoginSuccess.jsp :-

```
<%@ page language="java" contentType="text/html; charset=UTF-
8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
```

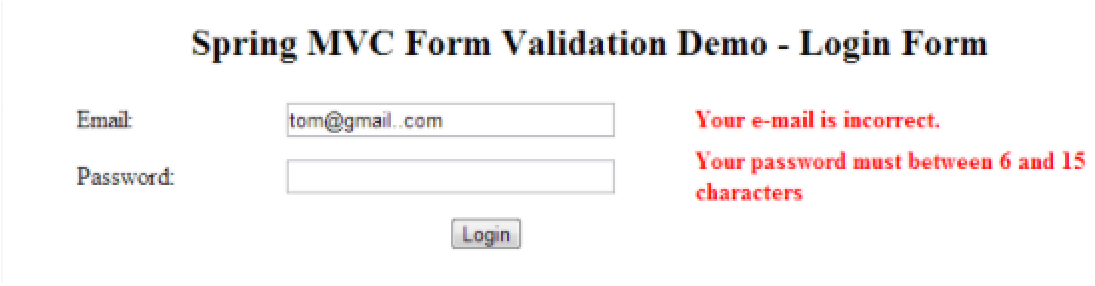


```
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Welcome</title>
</head>
<body>
    <div align="center">
        <h2>Welcome ${userForm.email}! You have logged in
successfully.</h2>
    </div>
</body>
</html>
```

Input/ output :-



Spring MVC Form Validation Demo - Login Form



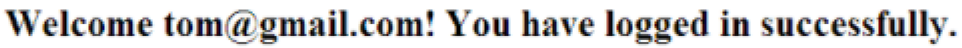
Spring MVC Form Validation Demo - Login Form

Email: tom@gmail..com

Your e-mail is incorrect.

Password:

Your password must between 6 and 15 characters



Welcome tom@gmail.com! You have logged in successfully.