

Ex 9	Java MVC Framework
-------------	---------------------------

Aim

To create a Java Web application using MVC framework.

Definitions

Web Application

A web application is a software program that runs on a remote server and is accessed via a web browser over the internet, without needing to be installed on a user's device. It is designed for user interaction, allowing users to send and receive data to perform tasks like online shopping, managing documents, or using social media.

Java Framework

A Java framework is a collection of pre-written code, classes, and tools that provides a structure for building Java applications, saving developers from writing common code from scratch. Popular examples include the **Spring framework** for enterprise applications, **Hibernate** for database interaction, and the **Java Collections Framework** for managing data structures.

MVC

A Java framework is a collection of pre-written code, classes, and tools that provides a structure for building Java applications, saving developers from writing common code from scratch. Popular examples include the **Spring framework** for enterprise applications, **Hibernate** for database interaction, and the **Java Collections Framework** for managing data structures.

Procedure

Open NetBeans IDE.

To create a Project go to File Menu → choose New Project → choose Java Web from Categories →choose Web Application from Projects →click next →specify the project name as MVC → Click Next → Click Next → click Finish.

Right Click on Source Packages Folder → new → Java Class → specify the class name as Student → click finish → type the following codes:

Student.java

```
// Student.java  
public class Student {  
    private String rollNo;  
    private String name;  
  
    public Student(String rollNo, String name) {  
        this.rollNo = rollNo;  
        this.name = name;  
    }  
  
    public String getRollNo() {  
        return rollNo;  
    }  
  
    public void setRollNo(String rollNo) {  
        this.rollNo = rollNo;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {
```

```
this.name = name;  
}  
}
```

Right Click on Source Packages Folder → new → Java Class → specify the class name as StudentView → click finish → type the following codes:

StudentView.java

```
// StudentView.java  
public class StudentView {  
    public void printStudentDetails(String studentName, String studentRollNo) {  
        System.out.println("Student Details:");  
        System.out.println("Name: " + studentName);  
        System.out.println("Roll No: " + studentRollNo);  
    }  
}
```

Right Click on Source Packages Folder → new → Java Class → specify the class name as StudentController → click finish → type the following codes:

StudentController.java

```
// StudentController.java  
public class StudentController {  
    private Student model;  
    private StudentView view;  
  
    public StudentController(Student model, StudentView view) {  
        this.model = model;  
        this.view = view;  
    }  
  
    public void setStudentName(String name) {  
        model.setName(name);  
    }
```

```

    }

public String getStudentName() {
    return model.getName();
}

public void setStudentRollNo(String rollNo) {
    model.setRollNo(rollNo);
}

public String getStudentRollNo() {
    return model.getRollNo();
}

public void updateView() {
    view.printStudentDetails(model.getName(), model.getRollNo());
}
}

```

Right Click on Source Packages Folder → new → Java Class → specify the class name as **MVCDemo** → click finish → type the following codes:

MVCDemo.java

```

// MVCDemo.java

public class MVCDemo {

    public static void main(String[] args) {
        // Create a Model
        Student student = new Student("123", "John Doe");

        // Create a View
        StudentView view = new StudentView();
    }
}

```

```
// Create a Controller and link Model and View  
StudentController controller = new StudentController(student, view);  
  
// Display initial student details  
controller.updateView();  
  
// Update student details through the Controller  
controller.setStudentName("Jane Smith");  
controller.setStudentRollNo("456");  
  
// Display updated student details  
controller.updateView();  
}  
}
```

Right click on MVCDemo.java → choose Run File.

Output

run:

Student Details:

Name: John Doe

Roll No: 123

Student Details:

Name: Jane Smith

Roll No: 456

BUILD SUCCESSFUL (total time: 0 seconds)

Result

Thus, a Java Web application has been developed and deployed using MVC framework.