

Explanation Mvc Pattern

1. HTML (View):

- Create a basic layout to display the app.
- Include input fields for entering the student name and roll number.
- Add a button to trigger the "Add Student" functionality.
- Provide an empty list element () to display the list of students.

2. CSS:

- Style the app for a clean and modern look.
- Add spacing, alignment, and colors for better user experience.

3. JavaScript:

○ **Model (Data):**

- Define a Student class with attributes: name and rollNo.

○ **View (Display):**

- The StudentView class manages all DOM manipulations (displaying students or showing messages).
- Provide methods to update the list of students and alert messages.

○ **Controller (Logic):**

- The StudentController class connects the user interactions with the data and view.
- Define a method to add a new student to the list.
- Bind the "Add Student" button to a function that reads input, creates a Student object, and updates the view.

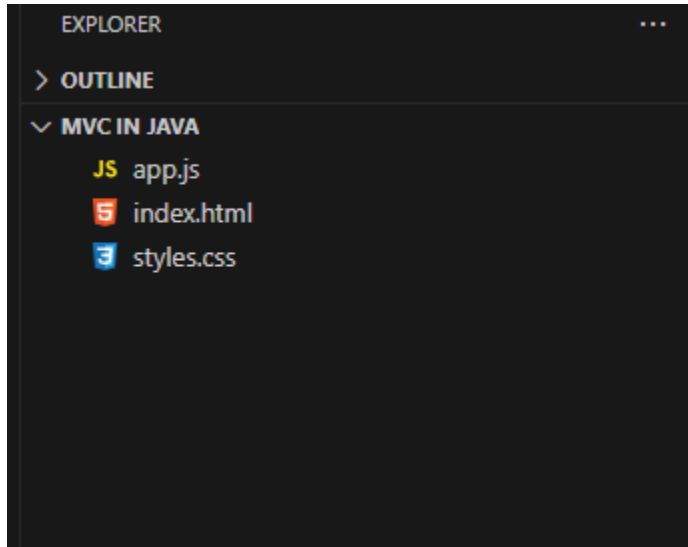
4. Initialization:

- Create instances of StudentView and StudentController to link the MVC components.
 - Automatically bind the controller's methods to the UI elements.
-

How It Works

- The user enters a name and roll number and clicks the "Add Student" button.
- The **Controller** reads the input, creates a new **Student (Model)**, and adds it to the array.
- The **View** updates the displayed list of students and shows an alert message for feedback.

Code:



Model:

```
JS app.js > StudentView > displayMessage
1 // Model
2 class Student {
3     constructor(name, rollNo) {
4         this.name = name;
5         this.rollNo = rollNo;
6     }
7 }
8
```

View:

```
9 // View
10 class StudentView {
11   constructor() {
12     this.studentsList = document.getElementById("studentsList");
13   }
14
15   //all students
16   displayStudents(students) {
17     this.studentsList.innerHTML = "";
18     students.forEach(student => {
19       const li = document.createElement("li");
20       li.textContent = `Name: ${student.name}, Roll No: ${student.rollNo}`;
21       this.studentsList.appendChild(li);
22     });
23   }
24
25   // Display a message
26   displayMessage(message) {
27     alert(message);
28   }
29 }
```

Controller:

```
31 // Controller
32 class StudentController {
33   constructor(view) {
34     this.students = [];
35     this.view = view;
36
37
38     document.getElementById("addStudentBtn").addEventListener("click", () => {
39       const name = document.getElementById("studentName").value;
40       const rollNo = document.getElementById("studentRollNo").value;
41       if (name && rollNo) {
42         this.addStudent(name, rollNo);
43       } else {
44         this.view.displayMessage("Please fill in all fields.");
45       }
46     });
47   }
48 }
```

Initialize MVC:

```
49      // Add a new student
50      addStudent(name, rollNo) {
51          const student = new Student(name, rollNo);
52          this.students.push(student);
53          this.view.displayMessage("Student added successfully!");
54          this.view.displayStudents(this.students);
55      }
56  }
57
58  // Initialize MVC
59  const view = new StudentView();
60  const controller = new StudentController(view);
61
```

Output:

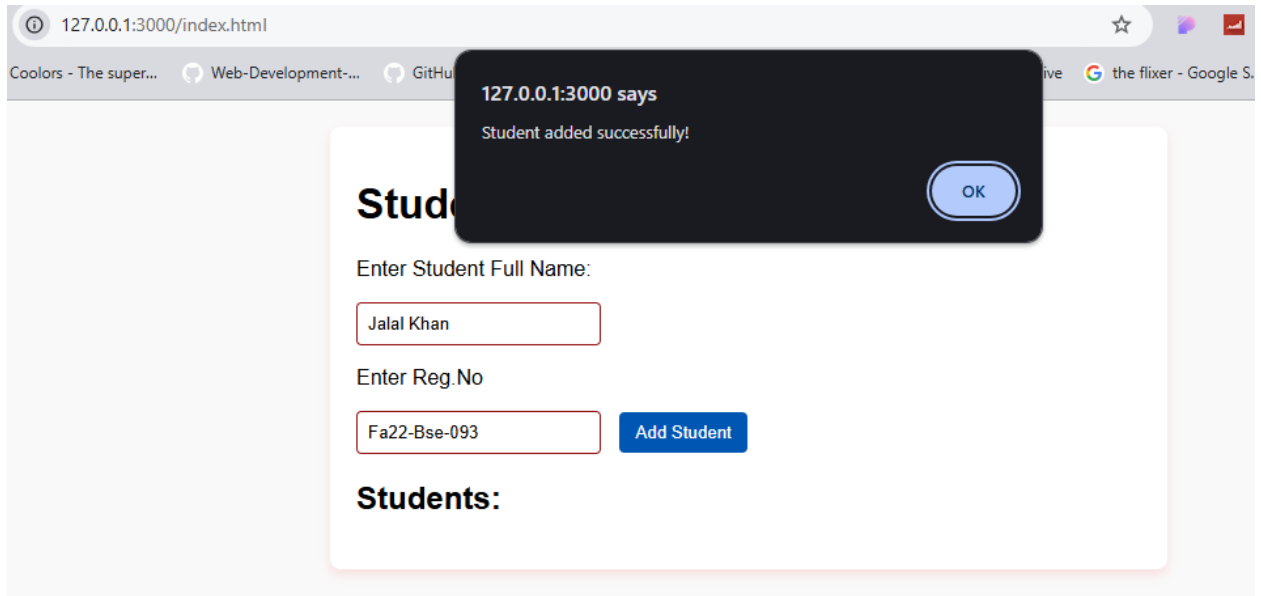
Student Management

Enter Student Full Name:

Enter Reg.No

Students:

Fill:



A screenshot of a web browser window. The address bar shows '127.0.0.1:3000/index.html'. The browser tabs include 'Coolors - The super...', 'Web-Development-...', 'GitHub', and 'the fixer - Google S...'. A dark overlay box in the center of the page displays the message '127.0.0.1:3000 says Student added successfully!' with an 'OK' button. Below the overlay, the form titled 'Student Management' is visible. It contains two input fields: 'Enter Student Full Name:' with the value 'Jalal Khan' and 'Enter Reg.No' with the value 'Fa22-Bse-093'. A blue 'Add Student' button is positioned to the right of the second input field. Below the inputs, the text 'Students:' is displayed.

Final Output:

Student Management

Enter Student Full Name:

Enter Reg.No

Students:

Name: Jalal khan, Roll No: Fa22-bse-093