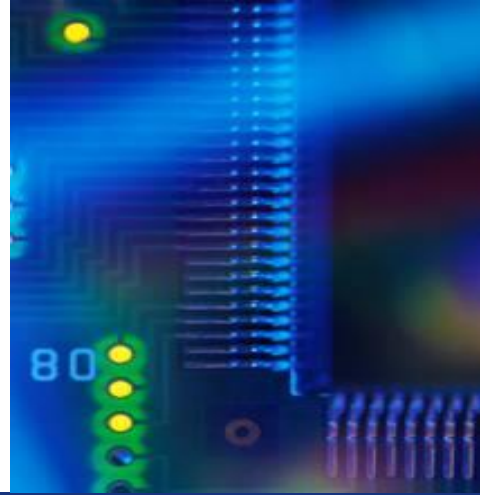
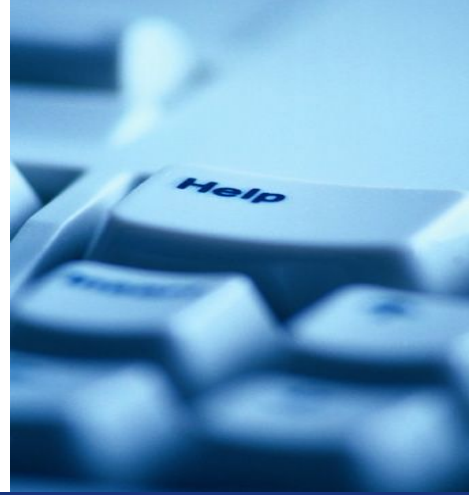




**UNIVERSITAS
BUDI LUHUR**



FAKULTAS TEKNOLOGI INFORMASI

Java Web Programming

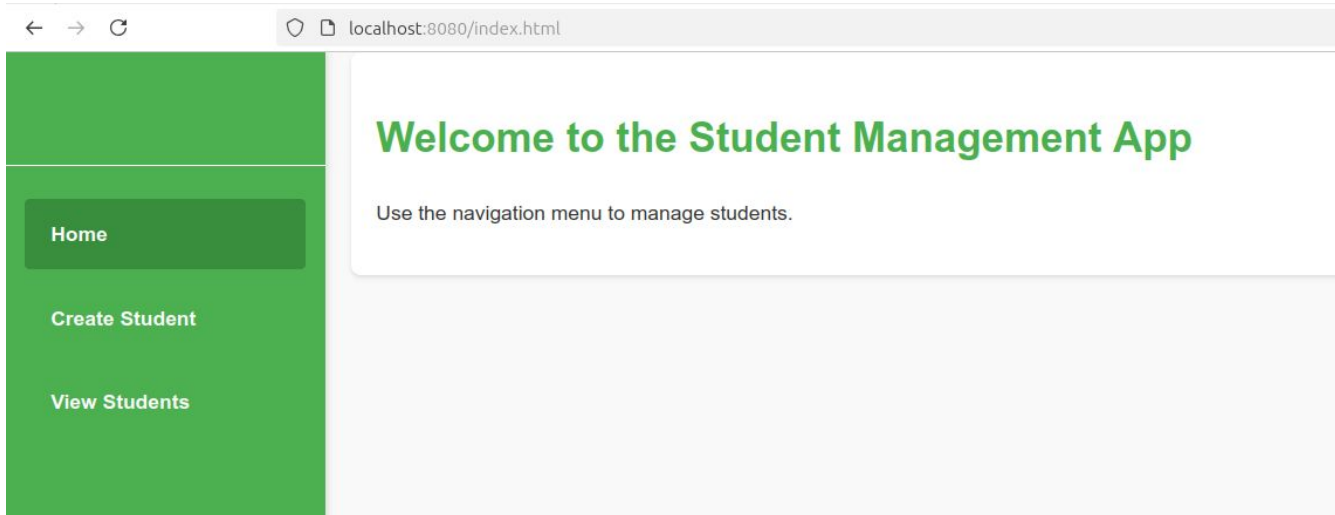
[PG130/ 3 SKS]

Riskiana Wulan, MKom

Pertemuan 15

Spring boot integration with the UI

Tampilan Student Management App



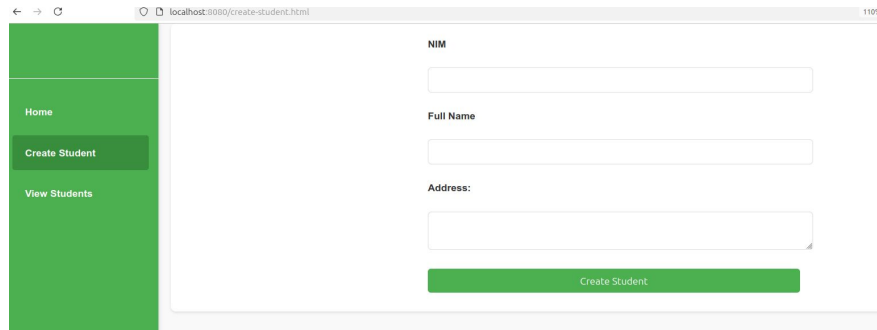
View Students

View Students

NIM	Name	Address	Date of Birth	Actions
22222	Cut Nyak Dien	Banda Aceh	2002-12-03	Edit Delete
33333	Diponegoro	Jakarta	2002-09-13	Edit Delete

[Home](#)
[Create Student](#)
[View Students](#)

Create-student



← → localhost:8080/create-student.html 110%

Home
Create Student
View Students

NIM

Full Name

Address:

Create Student

Edit student



nim

22222

Username:

Cut Nyak Dien

Date of birth

2002-12-03

Address:

Banda Acehhh

Update Student

Cancel

Delete Student

View Students

NIM	Name	Address	Date of Birth	Actions
22222	Cut Nyak Dien	Banda Aceh	2002-12-03	<button>Edit</button> <button>Delete</button>
33333	Diponegoro	Jakarta	2002-09-13	<button>Edit</button> <button>Delete</button>

localhost:8080

Are you sure you want to delete this student?

Cancel

OK

StudentController.java

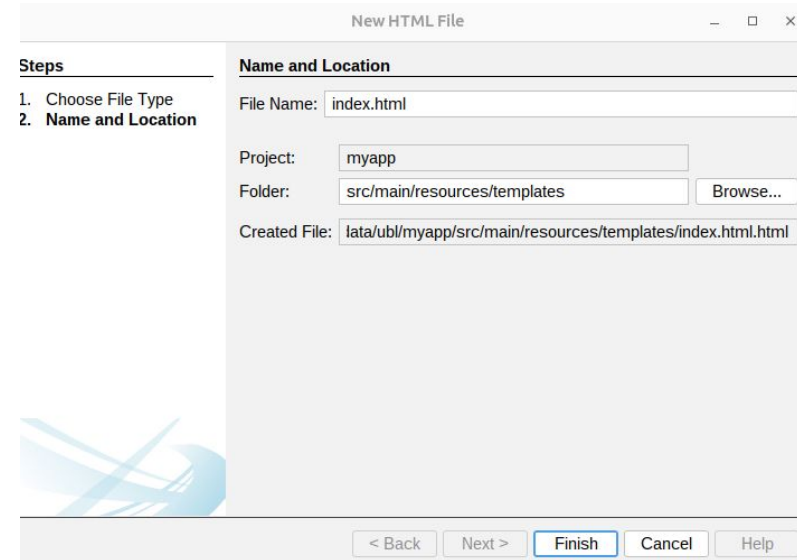
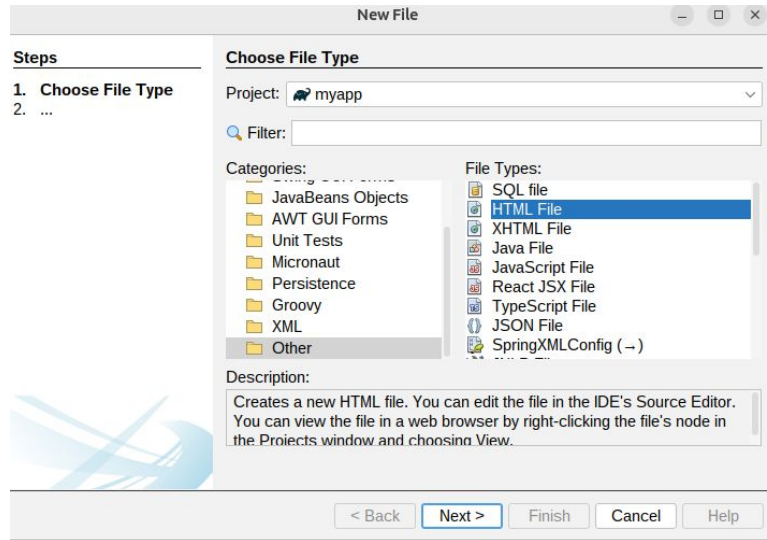
```

23  /
24  @RestController
25  @RequestMapping("/students")
26  public class StudentController {
27
28      private final StudentService studentService;
29
30      public StudentController(StudentService studentService) {
31          this.studentService = studentService;
32      }
33
34      @GetMapping
35      public List<Student> getAllStudents() {
36          return studentService.getAllStudents();
37      }
38
39      @PostMapping
40      public Student createStudent(@RequestBody Student student) {
41          return studentService.saveStudent(student);
42      }
43
44      @GetMapping("/{nim}")
45      public Student getStudentById(@PathVariable String nim) {
46          return studentService.getStudentByNim(nim);
47      }
48
49      @PutMapping("/{nim}")
50      public Student updateStudent(@PathVariable String nim, @RequestBody Student student) {
51          studentService.updateStudent(nim, student);
52          return new Student();
53      }
54
55      @DeleteMapping("/{nim}")
56      public void deleteStudent(@PathVariable String nim) {
57          studentService.deleteStudent(nim);
58      }
59  }
60
61

```

Create HTML Files

Create index.html



Buat template html yang diperlukan

1. Index.html
2. Create-student.html
3. View-students.html
4. Edit-student.html

Siapkan css file yang diperlukan

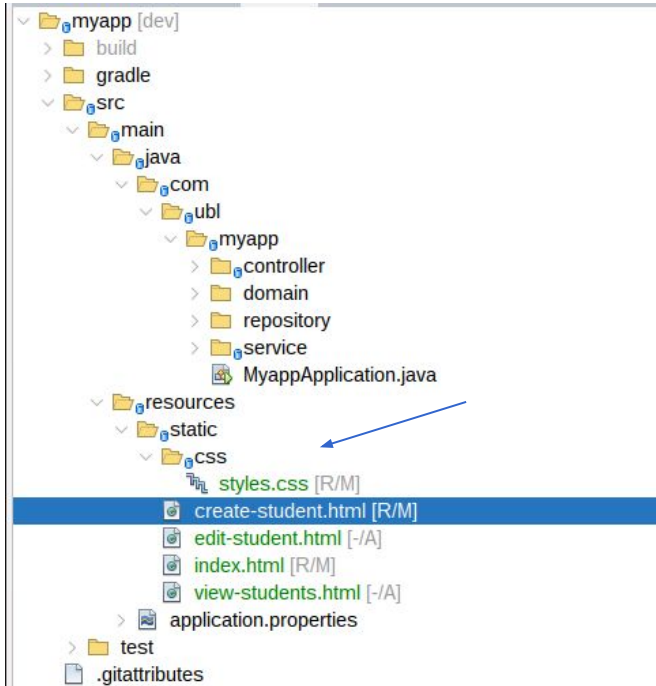
Contoh css file bisa didownload
dsni

<https://drive.google.com/file/d/1ZO8ZkxxmUIO-6LSiQaMbBao1GGzMaWsB/view?usp=sharing>

Lokasi html files ada di dalam static folder

By default, Spring Boot looks for static files in the following locations:

- **src/main/resources/static**
- src/main/resources/public
- src/main/resources/resources
- src/main/resources/META-INF/resources



index.html

```

...es  index.html x  styles.css x  create-student.html x  view-students.html x  StudentController.java x
Source  History  [Icons]
1      <!DOCTYPE html>
2      <html lang="en">
3      <head>
4          <meta charset="UTF-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1.0">
6          <title>Student Management</title>
7          <link rel="stylesheet" href="/css/styles.css">
8      </head>
9      <body>
10         <nav>
11             <h2>Student App</h2>
12             <a href="/index.html" class="active">Home</a>
13             <a href="/create-student.html">Create Student</a>
14             <a href="/view-students.html">View Students</a>
15         </nav>
16         <div class="container">
17             <h1>Welcome to the Student Management App</h1>
18             <p>Use the navigation menu to manage students.</p>
19         </div>
20     </body>
21 </html>
    
```

create-student.html

Memanggil function javascript createStudent

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Create Student</title>
7     <link rel="stylesheet" href="/css/styles.css">
8   </head>
9   <body>
10    <nav>
11      <h2>Student App</h2>
12      <a href="/index.html">Home</a>
13      <a href="/create-student.html" class="active">Create Student</a>
14      <a href="/view-students.html">View Students</a>
15    </nav>
16    <div class="container">
17      <form id="create-student-form" onsubmit="createStudent(event)">
18        <label for="nim">NIM</label>
19        <input type="text" id="nim" name="nim" required>
20        <label for="fullName">Full Name</label>
21        <input type="text" id="fullName" name="fullName" required>
22        <label for="dateOfBirth">Tanggal Lahir</label>
23        <input type="text" id="dateOfBirth" name="dateOfBirth" required>
24        <label for="address">Address</label>
25        <textarea id="address" name="address" required></textarea>
26        <button type="submit">Create Student</button>
27      </form>
28    </div>

```

```

80 <script>
81 async function createStudent(event) {
82   event.preventDefault();
83
84   const student = {
85     nim: document.getElementById("nim").value,
86     dateOfBirth: document.getElementById("dateOfBirth").value,
87     fullName: document.getElementById("fullName").value,
88     address: document.getElementById("address").value,
89   };
90
91   console.log(student);
92
93   const response = await fetch('/students', {
94     method: 'POST',
95     headers: {
96       'Content-Type': 'application/json',
97     },
98     body: JSON.stringify(student),
99   });
100
101   if (response.ok) {
102     alert("Student created successfully!");
103     window.location.href = "/view-students.html";
104   } else {
105     alert("Failed to create student.");
106   }
107 }
108 </script>
109 </body>
110 </html>

```

Function createStudent akan mendefinisikan object student parameter untuk dikirim sebagai Request Body saat memanggil endpoint api create student

Jika sukses, halaman akan berpindah ke view-students.jhtml

view-students.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>View Students</title>
7   <link rel="stylesheet" href="/css/styles.css">
8 </head>
9 <body>
10   <nav>
11     <h2>Student App</h2>
12     <a href="/index.html">Home</a>
13     <a href="/create-student.html">Create Student</a>
14     <a href="/view-students.html" class="active">View Students</a>
15   </nav>
16   <div class="container">
17     <h1>View Students</h1>
18     <table>
19       <thead>
20         <tr>
21           <th>NIM</th>
22           <th>Name</th>
23           <th>Address</th>
24           <th>Date of Birth</th>
25           <th>Actions</th>
26         </tr>
27       </thead>
28       <tbody id="students-table">
29         <!-- Dynamic student rows will be added here -->
30       </tbody>
31     </table>
32   </div>

```

Java script ditambahkan untuk memanggil api endpoint yg diperlukan

Memanggil api endpoint get students

Response list student akan ditampilkan pada row table

```

3  <script>
4  fetch('/students')
5  .then(response => response.json())
6  .then(data => {
7    const table = document.getElementById('students-table');
8    data.forEach(student => {
9      const row = document.createElement('tr');
10     row.innerHTML = `
11       <td>${student.nim}</td>
12       <td>${student.fullName}</td>
13       <td>${student.address}</td>
14       <td>${student.dateOfBirth}</td>
15       <td>
16         <button onclick="editStudent(${student.nim})">Edit</button>
17         <button onclick="deleteStudent(${student.nim})">Delete</button>
18       </td>
19     `;
20     table.appendChild(row);
21   });
22 }
23
24 // Delete student
25 function deleteStudent(nim) {
26   if (confirm('Are you sure you want to delete this student?')) {
27     fetch(`/students/${nim}`, { method: 'DELETE' })
28     .then(response => {
29       if (response.ok) {
30         alert('Student deleted successfully!');
31         location.reload();
32       } else {
33         alert('Failed to delete student.');

```

Memanggil api delete student by nim untuk menghapus student yg dipilih

Page akan berpindah ke edit student form jika edit diklik

Edit student

```

1 <html lang="en">
2 <head>
3   <meta charset="UTF-8">
4   <meta name="viewport" content="width=device-width, initial-scale=1.0">
5   <title>Edit Student</title>
6   <link rel="stylesheet" href="/css/styles.css">
7 </head>
8 <body>
9   <nav>
10    <h2>Student App</h2>
11    <a href="/index.html">Home</a>
12    <a href="/create-student.html">Create Student</a>
13    <a href="/view-students.html">View Students</a>
14  </nav>
15  <div class="container">
16    <form id="edit-student-form">
17      <label for="nim">Nim</label>
18      <input type="text" id="nim" name="nim" required>
19      <label for="fullName">Username</label>
20      <input type="text" id="fullName" name="fullName" required>
21      <br><br>
22      <label for="dateOfBirth">Date of birth</label>
23      <input type="text" id="dateOfBirth" name="dateOfBirth" required>
24      <br><br>
25      <label for="address">Address</label>
26      <textarea id="address" name="address" rows="4" required></textarea>
27      <br><br>
28      <button type="submit">Update Student</button>
29      <button type="button" onclick="window.location.href='/index.html';">Cancel</button>
30    </form>
  
```

```

12 <script>
13 // Extract student ID from the query string
14 const urlParams = new URLSearchParams(window.location.search);
15 const nim = urlParams.get('nim');
16
17 // Fetch student details and prefill the form
18 fetch(`/students/${nim}`)
19   .then(response => response.json())
20   .then(data => {
21     document.getElementById('nim').value = data.nim;
22     document.getElementById('fullName').value = data.fullName;
23     document.getElementById('dateOfBirth').value = data.dateOfBirth;
24     document.getElementById('address').value = data.address;
25   });
26
27 // Handle form submission
28 document.getElementById('edit-student-form').addEventListener('submit', function (event) {
29   event.preventDefault();
30
31   const student = {
32     nim: document.getElementById('nim').value,
33     fullName: document.getElementById('fullName').value,
34     dateOfBirth: document.getElementById('dateOfBirth').value,
35     address: document.getElementById('address').value
36   };
37
38   fetch(`/students/${nim}`, {
39     method: 'PUT',
40     headers: { 'Content-Type': 'application/json' },
41     body: JSON.stringify(student)
42   })
43     .then(response => {
44       if (response.ok) {
45         alert('Student updated successfully!');
46         window.location.href = '/view-students.html';
47       } else {
48         alert('Failed to update student.');

Menganggil endpoint api update student dengan mengirimkan student param sebagai request body


```