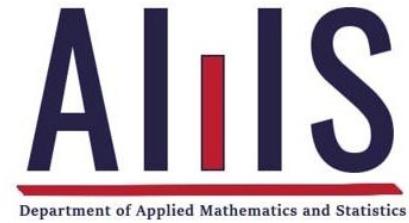




INSTITUTE OF TECHNOLOGY OF CAMBODIA

Department of Applied Mathematics and Statistics



OOP project: Schedule System

Group 1:

Members:

1. CHOUV You Y
2. AN Hengheng
3. KANG Pengkheang
4. HAYAVIN Rongravidwin
5. CHHIN Visal

ID:

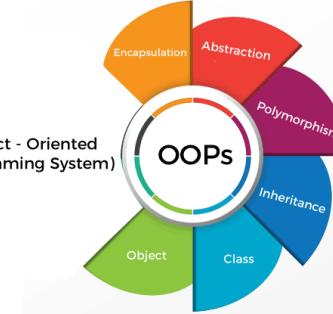
- e20211464
e20211125
e20210527
e20211502
e20210742

HTML

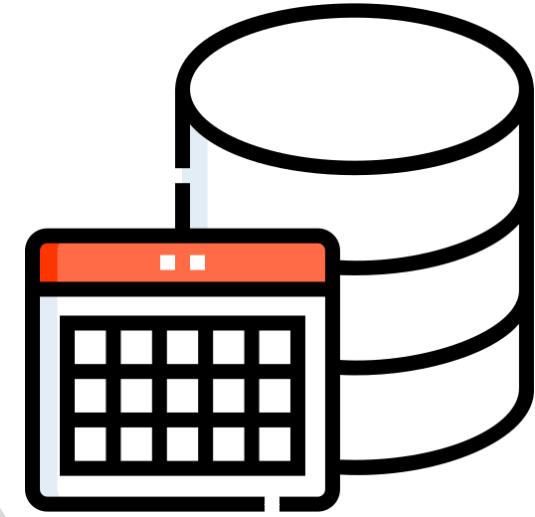


Schedule System

(Object - Oriented Programming System)



Object oriented programming



LECTURE



Na Standa

Ours team member



Chhin Visal
E2020742



CHOUV You Y
E20211464



Hayavin Rongravidwin
E20211502



Kang Pengkheang
E20210527



An Hengheng
E20211125

Table of content

Introduction

Why this project?

List of function

Databases

Code Structure

User Manual

Conclusion

Introduction

Teacher schedule



- We made Teacher schedule Management in order to know about which teacher teaches location and period of teaching.

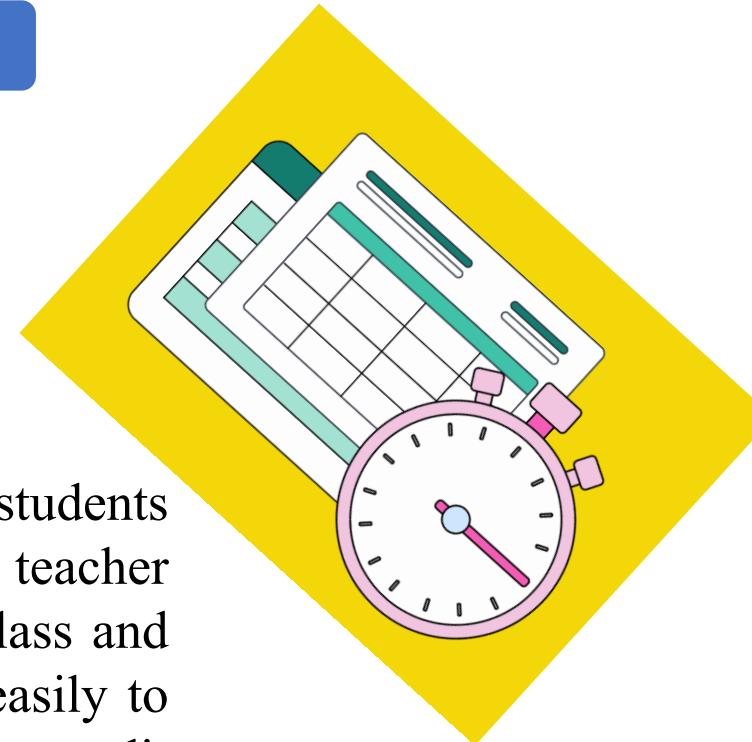
Course information



- We made Course Information Management in order to know about how many credit score, period of learning and the action that involve with those subject.

Why this project?

- We choose this project in order to make both students and teachers manage easily. In case, If one of teacher want to makeup class, we are easy to find the class and what time does it available. Students are more easily to know about the information of all subject such as credit score, period of learning and what action that should do during the class in per semester.



List of function

important packages



Repository

Since we have two tables in our database, there are also 2 controllers classes in our controller package

Model

Model package is where our entity classes are at, Schedule class and subject class



Index

Since we have two tables in our database, there are also 2 index to controller page application in web.

Add

We also have 2 add as well, one for add info of schedule and another for add info of subject in table on databases.

Update

And we have 2 update as well, one for update info of schedule and subject if have something wrong in table on databases

List of function

some info about Controller



Get mapping

We implement Get Mapping functions with different things, such as get mapping to our main page, then to our first and second table respectively



method get all
In here, we use this method in order to display all the information in our tables from database



"/create"
We are using get mapping and post mapping in order to create a new row with new information in our table through our web page



"/update"
In this point, we want to update our old information that already exist in the table, so we use get mapping and post mapping to access our database as well

Databases

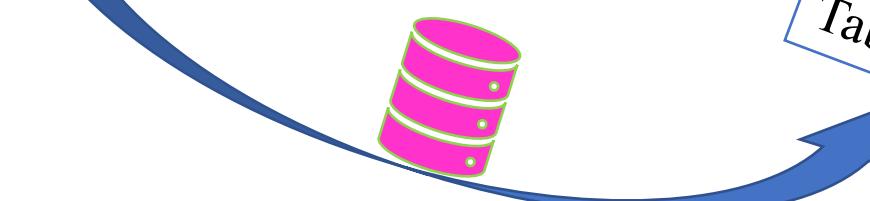
- ❖ We are using MySQL, Our database's name is “teacher” which contains two table, schedule and subject:

In Schedule table, there are 6 columns such as Id, Day Of Week, Time Slot, Teacher’s Name, Subject’s name and Room Number

Table 1

In our second table whose name is subject, there are 5 columns including Id, Subject’s name, credits, Total hours and Total weeks in a semester

Table 2



Code Structure

For Databases:



```
CREATE DATABASE teacher;
```

```
CREATE TABLE schedule(  
id INT PRIMARY KEY AUTO_INCREMENT,  
day_of_week varchar(250),  
time_slot varchar(250),  
teacher_name varchar(250),  
subject_name varchar(250),  
room_number varchar(25)  
);
```

```
INSERT INTO schedule (day_of_week,time_slot,teacher_name,subject_name,room_number)  
VALUES  
('Monday','7-11','Jeny','Programming','109F');
```

mysql> select * from schedule;					
id	day_of_week	time_slot	teacher_name	subject_name	room_number
1	Monday	7-11	Jeny	Programming	109F
6	Monday	9:00-11:00	Chhin Visal	Smos	A201
9	Wednesday	7:00-9:00	Chhin Visal	OOP	A201

Code Structure

For Databases:



```
CREATE DATABASE teacher;
```

```
CREATE TABLE subject (
id INT PRIMARY KEY AUTO_INCREMENT,
subject varchar(250),
credit varchar(250),
hours varchar(250),
weeks varchar(250)
);
```

```
INSERT INTO schedule (subject, credit, hours, weeks)
VALUES
('Linear Algebra','3','24 hours','12 weeks');
```

id	subject	credit	hours	weeks
1	Linear Algebra	3	24 hours	12 weeks
2	Differential Equation	3	24 hours	12 weeks
4	Data Structure and Programming	2	32 hours	16 weeks
5	Calculus III	2.25	24 hours	16 weeks

Code Structure

For Java: 

- ❖ We will briefly talk about our codes' structures as this structure will help with the readability maintenance and collaboration:

1. Import Packages
2. Important libraries
3. We mark the annotation in each package accordingly
 RequestMapping is displayed when needed
4. The handler methods such as GET, POST, DELETE
5. Path variable annotation



Source



Main



Java



Resource



Test



Java



Resources

Code Structure



-Package Controller:

● ● ●

```
1 package com.schedule.itc.ams.scheduleapi.controller;
2 import java.util.ArrayList;
3 import java.util.List;
4 import java.util.Optional;
5
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.stereotype.Controller;
8 import org.springframework.ui.Model;
9 import org.springframework.web.bind.annotation.GetMapping;
10 import org.springframework.web.bind.annotation.ModelAttribute;
11 import org.springframework.web.bind.annotation.PathVariable;
12 import org.springframework.web.bind.annotation.PostMapping;
13 import org.springframework.web.bind.annotation.RequestMapping;
14 import org.springframework.web.bind.annotation.RequestParam;
15 import org.springframework.web.servlet.mvc.support.RedirectAttributes;
16
17 import com.schedule.itc.ams.scheduleapi.data.ScheduleRepository;
18 import com.schedule.itc.ams.scheduleapi.model.Schedule;
19
20
21 @Controller
22 @RequestMapping("/schedule")
23 public class ScheduleController {
24     @Autowired
25     private ScheduleRepository repository;
26
27     @GetMapping("/main")
28     public String showMainPage(Model model) {
29         return "main";
30     }
31
32     @GetMapping
33     public String getAll(Model model, @RequestParam(name = "search", required = false) String search){//, @PathVariable String search
34         List<Schedule> scheduleList = new ArrayList<Schedule>();
35         if(search!=null && !search.isEmpty())
36             scheduleList = repository.findByTeacherNameContainingIgnoreCase(search); //repository.findByTeacherName(search);
37         else
38             scheduleList= repository.findAll();
39         model.addAttribute("scheduleList", scheduleList);
40         return "index";
41     }
42
43     @GetMapping("/create")
44     public String showAddingForm(Model model) {
45         Schedule schedule = new Schedule();
46         model.addAttribute("schedule", schedule);
47         return "new_schedule"; // new
48     }
49
```

```
50     @PostMapping("/create")
51     public String createSchedule(@ModelAttribute("schedule") Schedule schedule) {
52         repository.save(schedule);
53         return "redirect:/schedule";
54     }
55
56     @GetMapping("/update/{id}")
57     public String showUpdatingForm(@PathVariable("id") int id, Model model, RedirectAttributes redirectAttributes) {
58         Optional<Schedule> optionalSchedule = repository.findById(id);
59         if (optionalSchedule.isPresent()) {
60             Schedule schedule = optionalSchedule.get();
61             model.addAttribute("schedule", schedule);
62             return "update_schedule";
63         } else {
64             redirectAttributes.addFlashAttribute("error", "No Schedule found");
65             return "redirect:/schedule";
66         }
67     }
68
69     @PostMapping("/update/{id}")
70     public String updateSchedule(@PathVariable("id") int id, @ModelAttribute("schedule") Schedule schedule,RedirectAttributes redirectAttributes) {
71         Optional<Schedule> optionalSchedule = repository.findById(id);
72         if (optionalSchedule.isPresent()) {
73             Schedule existingSchedule = optionalSchedule.get();
74
75             existingSchedule.setDayOfWeek(schedule.getDayOfWeek());
76             existingSchedule.setTimeSlot(schedule.getTimeSlot());
77             existingSchedule.setTeacherName(schedule.getTeacherName());
78             existingSchedule.setSubjectName(schedule.getSubjectName());
79             existingSchedule.setRoomNumber(schedule.getRoomNumber());
80
81             repository.save(existingSchedule); // Save the updated schedule
82
83             // Add a success flash attribute
84             redirectAttributes.addFlashAttribute("success", "Schedule updated successfully");
85         } else {
86             // Add an error flash attribute if the schedule with the given ID is not found
87             redirectAttributes.addFlashAttribute("error", "Schedule not found");
88         }
89
90         return "redirect:/schedule";
91     }
92     @GetMapping("/delete/{id}")
93     public String delete(@PathVariable int id) {
94         repository.deleteById(id);
95         return "redirect:/schedule";
96     }
97
98 }
```

Code Structure



For Java: -Package Controller:

```
 1 package com.schedule.itc.ams.scheduleapi.controller;
 2
 3 import java.util.ArrayList;
 4 import java.util.List;
 5 import java.util.Optional;
 6
 7 import org.springframework.beans.factory.annotation.Autowired;
 8 import org.springframework.stereotype.Controller;
 9 import org.springframework.ui.Model;
10 import org.springframework.web.bind.annotation.GetMapping;
11 import org.springframework.web.bind.annotation.ModelAttribute;
12 import org.springframework.web.bind.annotation.PathVariable;
13 import org.springframework.web.bind.annotation.PostMapping;
14 import org.springframework.web.bind.annotation.RequestMapping;
15 import org.springframework.web.bind.annotation.RequestParam;
16 import org.springframework.web.servlet.mvc.support.RedirectAttributes;
17
18 import com.schedule.itc.ams.scheduleapi.data.SubjectRepository;
19 import com.schedule.itc.ams.scheduleapi.model.Subject;
20
21 @Controller
22 @RequestMapping("/subject")
23 public class SubjectController {
24     @Autowired
25     private SubjectRepository Repository;
26
27     @GetMapping
28     public String getAll(Model model, @RequestParam(name = "search", required = false) String search){ //, @PathVariable String search
29         List<Subject> subjectList = new ArrayList<Subject>();
30         if(search!=null && !search.isEmpty())
31             subjectList = Repository.findByNameContainingIgnoreCase(search);
32         else
33             subjectList= Repository.findAll();
34         model.addAttribute("subjectList", subjectList);
35         return "index1";
36     }
37
38     @GetMapping("/create")
39     public String showCreateForms(Model model) {
40         model.addAttribute("subject", new Subject());
41         return "add_new_subject";
42     }
43
44     @PostMapping("/create")
45     public String createSubject(@ModelAttribute("subject") Subject subject) {
46         Repository.save(subject);
47         return "redirect:/subject";
48     }
}
```

```
49     @GetMapping("/update/{id}")
50     public String showUpdatingForm(@PathVariable("id") int id, Model model,RedirectAttributes redirectAttributes) {
51         Optional<Subject> optionalSubject = Repository.findById(id);
52         if (optionalSubject.isPresent()) {
53             Subject subject = optionalSubject.get();
54             model.addAttribute("subject", subject);
55             return "update_subject";
56         } else {
57             redirectAttributes.addFlashAttribute("error", "No Subject found");
58             return "redirect:/subject";
59         }
60     }
61
62     @PostMapping("/update/{id}")
63     public String updateSubject(@PathVariable("id") int id, @ModelAttribute("subject") Subject subject,RedirectAttributes redirectAttributes) {
64         Optional<Subject> optionalSubject = Repository.findById(id);
65         if (optionalSubject.isPresent()) {
66             Subject existingSubject = optionalSubject.get();
67
68             // Update the existing subject with the data from the updatedSubject object
69             existingSubject.setSubject(subject.getSubject());
70             existingSubject.setCredit(subject.getCredit());
71             existingSubject.setHours(subject.getHours());
72             existingSubject.setWeeks(subject.getWeeks());
73
74             Repository.save(existingSubject); // Save the updated subject
75
76             // Add a success flash attribute
77             redirectAttributes.addFlashAttribute("success", "Subject updated successfully");
78         } else {
79             // Add an error flash attribute if the subject with the given ID is not found
80             redirectAttributes.addFlashAttribute("error", "Subject not found");
81         }
82
83         // Redirect the user to the subject list page
84         return "redirect:/subject";
85     }
86     @GetMapping("/delete/{id}")
87     public String delete(@PathVariable int id) {
88         Repository.deleteById(id);
89         return "redirect:/subject";
90     }
91
92
93 }
```

Code Structure

For Java: 

-Package Data:

Schedule

```
1 package com.schedule.itc.ams.scheduleapi.data;
2
3 import java.util.List;
4 import java.util.Optional;
5
6 import org.springframework.data.jpa.repository.JpaRepository;
7 import org.springframework.data.jpa.repository.Modifying;
8 import org.springframework.data.jpa.repository.Query;
9 import org.springframework.data.repository.query.Param;
10 import org.springframework.stereotype.Repository;
11
12 import com.schedule.itc.ams.scheduleapi.model.Schedule;
13
14 @Repository
15 public interface ScheduleRepository extends JpaRepository<Schedule, Integer> {
16     Optional<Schedule> findById(int id);
17
18     //@Query("select s from schedule s where lower(s.name) like lower(CONCAT('%', :search, '%')) or city like lower(CONCAT('%', :search, '%'))")
19     //List<Schedule> findByName(@Param("search") String search);
20
21     @Query("select s from Schedule s where lower(s.TeacherName) like lower(CONCAT('%', :search, '%'))")
22     List<Schedule> findByTeacherNameContainingIgnoreCase(@Param("search") String search);
23 //
24 //     @Modifying
25 //     @Query("delete from Schedule s where lower(s.TeacherName) = lower(:teacherName)")
26 //     void removeByTeacherNameIgnoreCase(@Param("teacherName") String teacherName);
27 }
28
29
```

Subject

```
1 package com.schedule.itc.ams.scheduleapi.data;
2
3 import java.util.List;
4 import java.util.Optional;
5
6 import org.springframework.data.jpa.repository.JpaRepository;
7 import org.springframework.data.jpa.repository.Query;
8 import org.springframework.stereotype.Repository;
9
10 import com.schedule.itc.ams.scheduleapi.model.Subject;
11
12 @Repository
13 public interface SubjectRepository extends JpaRepository<Subject, Integer> {
14     Optional<Subject> findById(int id);
15     @Query("select s from Subject s where lower(s.Subject) like lower(CONCAT('%', :subject, '%'))")
16     List<Subject> findBySubjectNameContainingIgnoreCase(String subject);
17
18
19 }
20
```

Code Structure



For Java: -Package Model:

Schedule

```
 1 package com.schedule.itc.ams.scheduleapi.model;
 2
 3 import jakarta.persistence.Column;
 4 import jakarta.persistence.Entity;
 5 import jakarta.persistence.GeneratedValue;
 6 import jakarta.persistence.GenerationType;
 7 import jakarta.persistence.Id;
 8 import jakarta.persistence.Table;
 9
10 @Entity
11 @Table(name = "schedule")
12 public class Schedule {
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15
16     private int id;
17     @Column (name = "day_of_week")
18     private String DayOfWeek;
19     @Column (name = "time_slot")
20     private String TimeSlot;
21     @Column (name = "teacher_name")
22     private String TeacherName;
23     @Column (name = "subject_name")
24     private String SubjectName;
25     @Column (name = "room_number")
26     private String RoomNumber;
27
28     public Schedule (String DayOfWeek, String TimeSlot, String TeacherName, String SubjectName, String RoomNumber) {
29         this.DayOfWeek= DayOfWeek;
30         this.TimeSlot = TimeSlot;
31         this.TeacherName = TeacherName;
32         this.SubjectName = SubjectName;
33         this.RoomNumber = RoomNumber;
34     }
35 }
```

```
35
36     public Schedule() {
37         // TODO Auto-generated constructor stub
38     }
39     public int getId() {
40         return id;
41     }
42     public void setId(int id) {
43         this.id = id;
44     }
45
46     public String getDayOfWeek() {
47         return DayOfWeek;
48     }
49
50     public void setDayOfWeek(String dayOfWeek) {
51         DayOfWeek = dayOfWeek;
52     }
53
54     public String getTimeSlot() {
55         return TimeSlot;
56     }
57
58     public void setTimeSlot(String timeSlot) {
59         TimeSlot = timeSlot;
60     }
61
62     public String getTeacherName() {
63         return TeacherName;
64     }
65
66     public void setTeacherName(String teacherName) {
67         TeacherName = teacherName;
68     }
69
70     public String getSubjectName() {
71         return SubjectName;
72     }
73
74     public void setSubjectName(String subjectName) {
75         SubjectName = subjectName;
76     }
77
78     public String getRoomNumber() {
79         return RoomNumber;
80     }
81
82     public void setRoomNumber(String roomNumber) {
83         RoomNumber = roomNumber;
84     }
85
86
87 }
```

Code Structure

For Java: 

-Package Model:

Subject

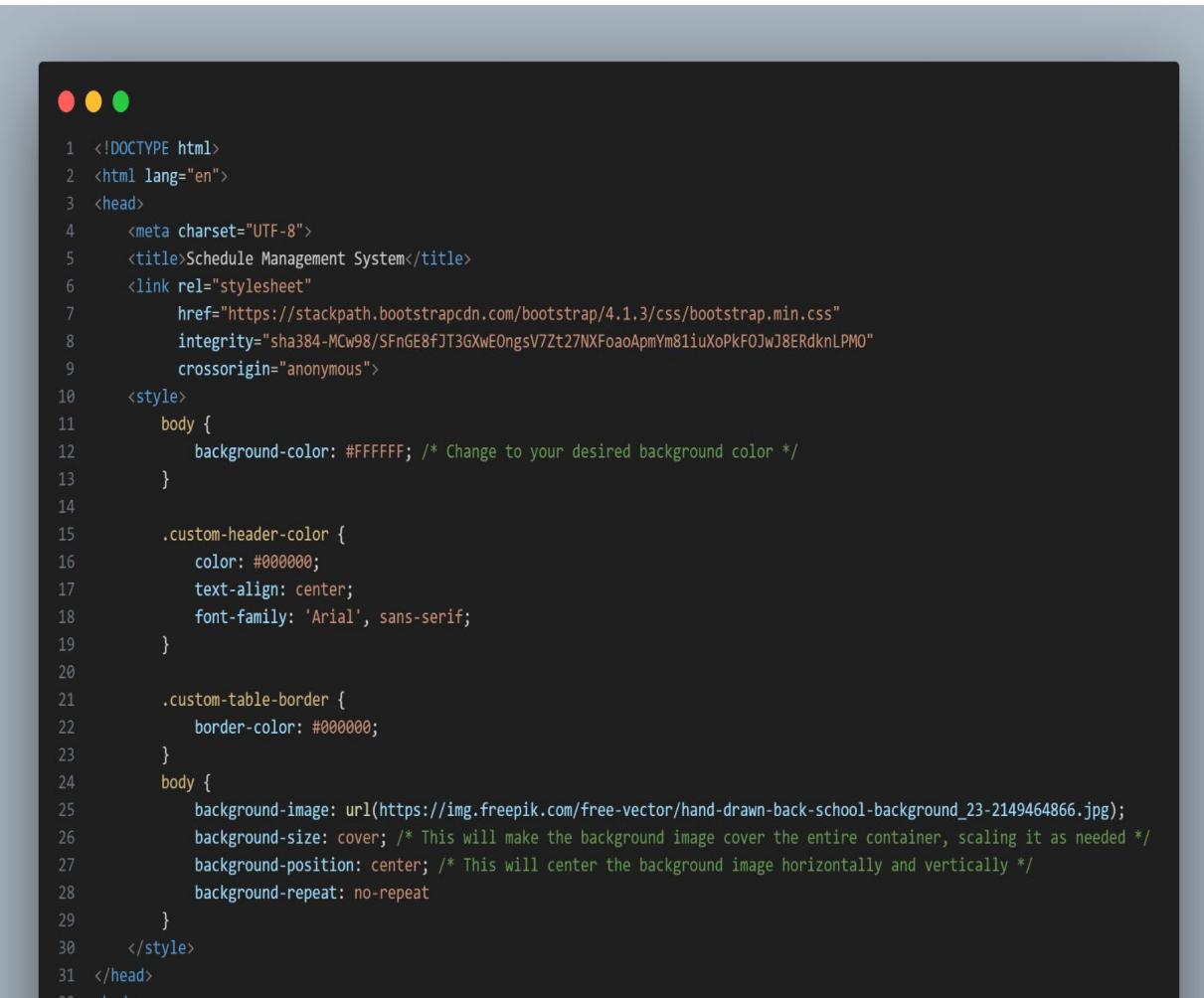
```
● ● ●  
1 package com.schedule.itc.ams.scheduleapi.model;  
2  
3 import jakarta.persistence.Column;  
4 import jakarta.persistence.Entity;  
5 import jakarta.persistence.GeneratedValue;  
6 import jakarta.persistence.GenerationType;  
7 import jakarta.persistence.Id;  
8 import jakarta.persistence.Table;  
9  
10 @Entity  
11 @Table(name = "subject")  
12 public class Subject {  
13     @Id  
14     @GeneratedValue(strategy = GenerationType.IDENTITY)  
15     private int id;  
16     @Column (name = "subject")  
17     private String Subject;  
18     @Column (name = "credit")  
19     private String credit;  
20     @Column (name = "hours")  
21     private String Hours;  
22     @Column (name = "weeks")  
23     private String Weeks;  
24  
25     public Subject(String Subject, String credit, String Hours, String Weeks){  
26         this.Subject = Subject;  
27         this.credit = credit;  
28         this.Hours = Hours;  
29         this.Weeks = Weeks;  
30     }  
31  
32     public Subject() {  
33         // TODO Auto-generated constructor stub  
34     }  
35  
36     public int getId() {  
37         return id;  
38     }  
39  
40     public void setId(int id) {  
41         this.id = id;  
42     }  
43  
44     public String getSubject() {  
45         return Subject;  
46     }  
47  
48     public void setSubject(String subject) {  
49         Subject = subject;  
50     }  
51  
52     public String getCredit() {  
53         return credit;  
54     }  
55  
56     public void setCredit(String credit) {  
57         this.credit = credit;  
58     }  
59  
60     public String getHours() {  
61         return Hours;  
62     }  
63  
64     public void setHours(String hours) {  
65         Hours = hours;  
66     }  
67  
68     public String getWeeks() {  
69         return Weeks;  
70     }  
71  
72     public void setWeeks(String weeks) {  
73         Weeks = weeks;  
74     }  
75  
76     }  
77 }  
78 }
```

```
31  
32     public Subject() {  
33         // TODO Auto-generated constructor stub  
34     }  
35  
36     public int getId() {  
37         return id;  
38     }  
39  
40     public void setId(int id) {  
41         this.id = id;  
42     }  
43  
44     public String getSubject() {  
45         return Subject;  
46     }  
47  
48     public void setSubject(String subject) {  
49         Subject = subject;  
50     }  
51  
52     public String getCredit() {  
53         return credit;  
54     }  
55  
56     public void setCredit(String credit) {  
57         this.credit = credit;  
58     }  
59  
60     public String getHours() {  
61         return Hours;  
62     }  
63  
64     public void setHours(String hours) {  
65         Hours = hours;  
66     }  
67  
68     public String getWeeks() {  
69         return Weeks;  
70     }  
71  
72     public void setWeeks(String weeks) {  
73         Weeks = weeks;  
74     }  
75  
76     }  
77 }  
78 }
```

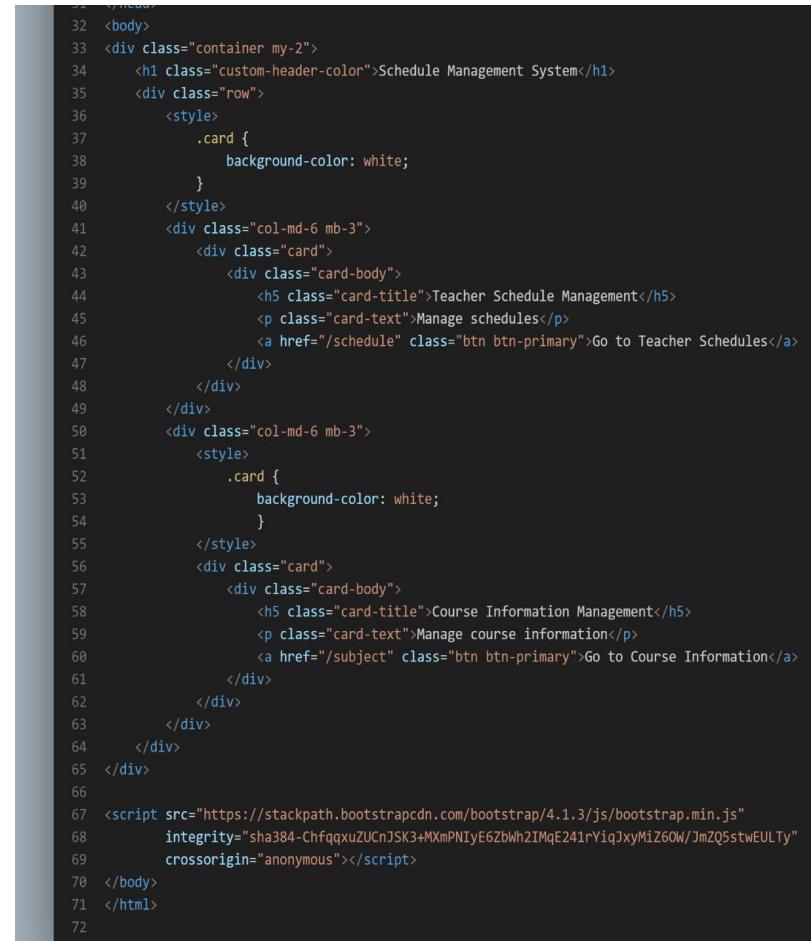
Code Structure

For HTML: 

-Package Main:



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Schedule Management System</title>
6     <link rel="stylesheet"
7         href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
8         integrity="sha384-MCw9SFGE8fJT3GXwEONGsV7Zt27NXFaoApmYm81iuXoPkFOJWJ8ERdknLPMO"
9         crossorigin="anonymous">
10    <style>
11        body {
12            background-color: #FFFFFF; /* Change to your desired background color */
13        }
14
15        .custom-header-color {
16            color: #000000;
17            text-align: center;
18            font-family: 'Arial', sans-serif;
19        }
20
21        .custom-table-border {
22            border-color: #000000;
23        }
24
25        body {
26            background-image: url("https://img.freepik.com/free-vector/hand-drawn-back-school-background_23-2149464866.jpg");
27            background-size: cover; /* This will make the background image cover the entire container, scaling it as needed */
28            background-position: center; /* This will center the background image horizontally and vertically */
29            background-repeat: no-repeat
30        }
31    </style>
32 </head>
33 <body>
```



```
34 <div class="container my-2">
35     <h1 class="custom-header-color">Schedule Management System</h1>
36     <div class="row">
37         <style>
38             .card {
39                 background-color: white;
40             }
41         </style>
42         <div class="col-md-6 mb-3">
43             <div class="card">
44                 <div class="card-body">
45                     <h5 class="card-title">Teacher Schedule Management</h5>
46                     <p class="card-text">Manage schedules</p>
47                     <a href="/schedule" class="btn btn-primary">Go to Teacher Schedules</a>
48                 </div>
49             </div>
50             <div class="col-md-6 mb-3">
51                 <style>
52                     .card {
53                         background-color: white;
54                     }
55                 </style>
56                 <div class="card">
57                     <div class="card-body">
58                         <h5 class="card-title">Course Information Management</h5>
59                         <p class="card-text">Manage course information</p>
60                         <a href="/subject" class="btn btn-primary">Go to Course Information</a>
61                     </div>
62                 </div>
63             </div>
64         </div>
65     </div>
66
67     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
68         integrity="sha384-ChfqxUZCnJSK3+MXmPNIyE6zbWh2IMqE241rYiqJxyMiz60W/JmZQ5stwEULTy"
69         crossorigin="anonymous"></script>
70 </body>
71 </html>
```

Code Structure

For HTML: 

-Package Index:

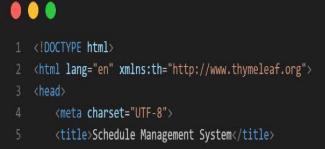
```
● ● ●
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>Teacher Schedule Management System</title>
6
7     <link rel="stylesheet"
8       href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
9       integrity="sha384-MCw8/SFnGE8fJT3GWwEOngvV7Zt27NKFoa0ApM\m8iuXoPKFOJw28ERdknLPM0"
10      crossorigin="anonymous">
11
12     <style>
13       body {
14         background-color: #FFB6C1; /* Change to your desired background color */
15       }
16
17     .custom-header-color {
18       color: #000000;
19       text-align: center;
20       font-family: 'Arial', sans-serif;
21     }
22
23     .custom-table-border {
24       border-color: #000000;
25     }
26     </style>
27     <!--<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4Kbq78iYhF1dvUhFTAU6au8tT94WnifjtDbrCEXSU1oBoqj12o/Z6jIN3" crossorigin="anonymous">-->
28   </head>
29   <body>
30     <div class="container my-2">
31       <h1 class="custom-header-color">Teacher Schedule Management System</h1>
32
33       <a th:href = "@{/schedule/create}" class="btn btn-primary btn-sm mb-3" style="background-color: white; color: black;"> Add Schedule </a>
34
35       <!-- Filter Form -->
36       <div class="row">
37         <div class="col-md-4 mb-3">
38           <form th:action="@{/schedule}" method="GET">
39             <div class="input-group">
40               <input type="text" class="form-control" placeholder="Id or Teacher's Name" id="filterInput" name="search" th:value="${search}" />
41               <button type="submit" class="btn btn-primary" style="background-color: white; color: black;">Filter</button>
42             </div>
43           </form>
44         </div>
45       </div>
46     </div>
```

```
45
46   <table border="1" class = "table table-striped table-responsive-md">
47     <thead>
48       <tr>
49         <th>ID</th>
50         <th>Day Of Week</th>
51         <th>Time Slot</th>
52         <th>Teacher Name</th>
53         <th>Subject Name</th>
54         <th>Room Number</th>
55
56         <th>Actions</th>
57       </tr>
58     </thead>
59     <tbody>
60       <tr th:each="sche : ${scheduleList}">
61         <td th:text="${sche.id}"></td>
62         <td th:text="${sche.DayOfWeek}"></td>
63         <td th:text="${sche.TimeSlot}"></td>
64         <td th:text="${sche.TeacherName}"></td>
65         <td th:text="${sche.SubjectName}"></td>
66         <td th:text="${sche.RoomNumber}"></td>
67         <!-- Update and Delete Actions -->
68         <td>
69           <a th:href="@{/schedule/update/{id}}{id=${sche.id}}" class="btn btn-primary" style="background-color: white; color: black;">Update</a>
70           <a class="btn btn-danger" th:onClick="showConfirmationModal('${sche.id}')" style="background-color: white; color: black;">Delete</a>
71
72           <!--<a th:href="@{/schedule/delete/{id}}{id=${sche.id}}" class="btn btn-danger">Delete</a-->
73         </td>
74       </tr>
75     </tbody>
76   </table>
77   <div class="text-center mt-3">
78     <a href="/schedule/main" class="btn btn-primary" style="background-color: white; color: black;">Go Back to Main Page</a>
79   </div>
80
81 </div>
82
83 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka75k8Gn4gmtz2MlQnikT1w0gysQgOMhuP+IRH9sENB08LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>
84 <!-- Javascript for Confirmation Modal-->
85 <script type="text/javascript">
86   function showConfirmationModal(id) {
87     var confirmation = confirm("Are you sure you want to delete this schedule information?");
88     if (confirmation) {
89       window.location.href = "/schedule/delete/" + id;
90     }
91   }
92 </script>
93 </body>
```

Code Structure

For Java: 

-Package Index1:



```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <meta charset="UTF-8">
5     <title>Schedule Management System</title>
6
7     <link rel="stylesheet"
8         href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
9         integrity="sha384-MCw1iEjdsR3VQoMmRnEAcnGtqZ7NXFaoApmY81uXoPKFOJ78ERdknLPM0"
10        crossorigin="anonymous">
11
12     <style>
13         body {
14             background-color: #E0FFFF; /* Change to your desired background color */
15         }
16
17         .custom-header-color {
18             color: #000000;
19             text-align: center;
20             font-family: 'Arial', sans-serif;
21         }
22
23         .custom-table-border {
24             border-color: #000000;
25         }
26
27     </style>
28 </head>
29 <body>
30     <div class="container my-2">
31         <h1 class="custom-header-color">Courses Information</h1>
32
33         <a th:href = "#@{/subject/create}" class="btn btn-primary btn-sm mb-3" style="background-color: white; color: black;"> Add Subject </a>
34
35         <!-- Filter Form -->
36         <div class="row">
37             <div class="col-md-4 mb-3">
38                 <form th:action="#@{/subject}" method="GET">
39                     <div class="input-group">
40                         <input type="text" class="form-control" placeholder="Subject" id="filterInput" name="search" th:value="${search}" />
41                         <button type="submit" class="btn btn-primary" style="background-color: white; color: black;">Filter</button>
42                     </div>
43                 </form>
44             </div>
45         </div>
46     </div>
```

```
47     <table border="1" class="table table-striped table-responsive-md">
48         <thead>
49             <tr>
50                 <th>ID</th>
51                 <th>Subject</th>
52                 <th>Credit</th>
53                 <th>Hours</th>
54                 <th>Weeks</th>
55                 <th>Action</th>
56             </tr>
57         </thead>
58         <tbody>
59             <tr th:each="subject : ${subjectList}">
60                 <td th:text="${subject.id}"></td>
61                 <td th:text="${subject.Subject}"></td>
62                 <td th:text="${subject.credit}"></td>
63                 <td th:text="${subject.Hours}"></td>
64                 <td th:text="${subject.Weeks}"></td>
65
66                 <td>
67                     <!-- Update and Delete Actions -->
68                     <td>
69                         <a th:href="@{/subject/update/{id}(id=${subject.id})" class="btn btn-primary" style="background-color: white; color: black;">Update</a>
70                         <a class="btn btn-danger" th:onclick="showConfirmationModal('${subject.id}')" style="background-color: white; color: black;">Delete</a>
71                     </td>
72                 </td>
73             </tr>
74         </tbody>
75     </table>
76     <div class="text-center mt-3">
77         <a href="/schedule/main" class="btn btn-primary" style="background-color: white; color: black;">Go Back to Main Page</a>
78     </div>
79
80 </div>
81
82 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-KJqgF1IeTCwzKk+IaQkQjxKqMkPdZSgkEJGwZGZLXyW0OOGN0D8u0Z0I4ZqyJ" crossorigin="anonymous"></script>
83 <!-- JavaScript for Confirmation Modal-->
84 <script type="text/javascript">
85     function showConfirmationModal(id) {
86         var confirmation = confirm("Are you sure you want to delete this subject's detail?");
87         if (confirmation) {
88             window.location.href = "/subject/delete/" + id;
89         }
90     }
91 </script>
92 </body>
93 </html>
```

Code Structure

For Java: 

-Package Update_schedule:

```
1  <!DOCTYPE html>
2  <html lang="en" xmlns:th="http://www.thymeleaf.org">
3  <head>
4  <meta charset="ISO-8859-1">
5  <title>Teacher Schedule Management System</title>
6  <link rel="stylesheet"
7      href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
8      integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
9      crossorigin="anonymous">
10 <!--link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous"-->
11 </head>
12 <body>
13     <div class="container">
14         <h1>Teacher Schedule Management System</h1>
15         <hr>
16         <h2>Update Schedule</h2>
17
18         <form action="#" th:action="@{/schedule/update/{id}(id=${schedule.id})}" th:object="${schedule}" method="POST">
19             <input type="text" th:field="*{DayOfWeek}"
20                 placeholder="Day Of Week" class="form-control mb-4 col-4">
21
22             <input type="text" th:field="*{TimeSlot}"
23                 placeholder="Time Slot (9:00-10:00)" class="form-control mb-4 col-4">
24
25             <input type="text" th:field="*{TeacherName}"
26                 placeholder="Teacher Name" class="form-control mb-4 col-4">
27
28             <input type="text" th:field="*{SubjectName}"
29                 placeholder="Subject Name" class="form-control mb-4 col-4">
30
31             <input type="text" th:field="*{RoomNumber}"
32                 placeholder="Room Number" class="form-control mb-4 col-4">
33
34             <button type="submit" class="btn btn-info col-2"> Update</button>
35         </form>
36
37         <hr>
38
39         <a th:href = "@{/schedule}"> Back to Schedule List</a>
40     </div>
41 </body>
42 </html>
```

Code Structure

For Java: 

-Package Update_subject:

```
● ● ●
1  <!DOCTYPE html>
2  <html lang="en" xmlns:th="http://www.thymeleaf.org">
3  <head>
4  <meta charset="ISO-8859-1">
5  <title>Teacher Schedule Management System</title>
6  <link rel="stylesheet"
7   href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
8   integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
9   crossorigin="anonymous">
10 <!!--<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFlvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">-->
11 </head>
12 <body>
13   <div class="container">
14     <h1>Courses Information</h1>
15     <hr>
16     <h2>Update Subject</h2>
17
18     <form action="#" th:action="@{/subject/update/{id}(id=${subject.id})}" th:object="${subject}"
19       method="POST">
20       <input type="text" th:field="*{Subject}"
21         placeholder="Subject" class="form-control mb-4 col-4">
22
23       <input type="text" th:field="*{credit}"
24         placeholder="Credits" class="form-control mb-4 col-4">
25
26       <input type="text" th:field="*{Hours}"
27         placeholder="Total hours in a semester" class="form-control mb-4 col-4">
28
29       <input type="text" th:field="*{Weeks}"
30         placeholder="Total weeks in a semester" class="form-control mb-4 col-4">
31
32
33       <button type="submit" class="btn btn-info col-2"> Update</button>
34     </form>
35
36     <hr>
37
38     <a th:href = "@{/subject}"> Back to Subject List</a>
39   </div>
40 </body>
41 </html>
```

Code Structure

For Java: 

-Package Add_schedule:

```
1  <!DOCTYPE html>
2  <html lang="en" xmlns:th="http://www.thymeleaf.org">
3  <head>
4  <meta charset="ISO-8859-1">
5  <title>Teacher Schedule Management System</title>
6  <link rel="stylesheet"
7      href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
8      integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm8liuXoPkFOJwJ8ERdknLPM0"
9      crossorigin="anonymous">
10 <!!--<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhF1dvKuhFTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">-->
11 </head>
12 <body>
13     <div class="container">
14         <h1>Teacher Schedule Management System</h1>
15         <hr>
16         <h2>Save All</h2>
17
18         <form action="#" th:action="@{/schedule/create}" th:object="${schedule}"
19             method="POST">
20             <input type="text" th:field="*{DayOfWeek}"
21                 placeholder="Day Of Week" class="form-control mb-4 col-4">
22
23             <input type="text" th:field="*{TimeSlot}"
24                 placeholder="Time Slot (9:00-10:00)" class="form-control mb-4 col-4">
25
26             <input type="text" th:field="*{TeacherName}"
27                 placeholder="Teacher Name" class="form-control mb-4 col-4">
28
29             <input type="text" th:field="*{SubjectName}"
30                 placeholder="Subject Name" class="form-control mb-4 col-4">
31
32             <input type="text" th:field="*{RoomNumber}"
33                 placeholder="Room Number" class="form-control mb-4 col-4">
34
35             <button type="submit" class="btn btn-info col-2"> Save Schedule</button>
36         </form>
37
38         <hr>
39
40         <a th:href = "@{/schedule}"> Back to Schedule List</a>
41     </div>
42 </body>
43 </html>
```

Code Structure

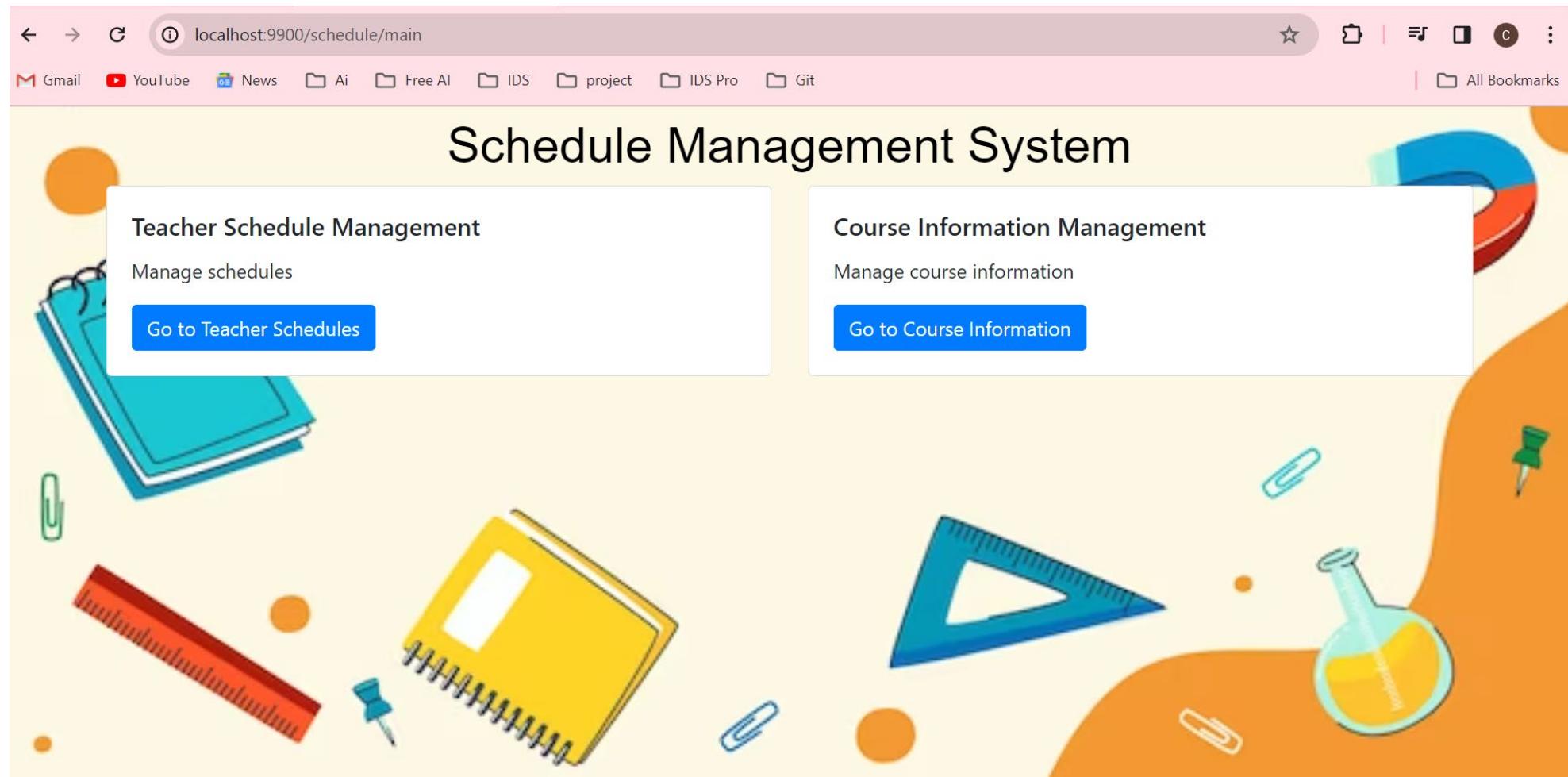
For Java: 

-Package Add_subject:

```
1  <!DOCTYPE html>
2  <html lang="en" xmlns:th="http://www.thymeleaf.org">
3  <head>
4  <meta charset="ISO-8859-1">
5  <title>Teacher Schedule Management System</title>
6  <link rel="stylesheet"
7      href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
8      integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm8iliXoPkFOJwJ8ERdknLPMO"
9      crossorigin="anonymous">
10 <!--<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">-->
11 </head>
12 <body>
13     <div class="container">
14         <h1>Courses Information</h1>
15         <hr>
16         <h2>Save All</h2>
17
18         <form action="#" th:action="@{/subject/create}" th:object="${subject}"
19             method="POST">
20             <input type="text" th:field="*{Subject}"
21                 placeholder="Subject" class="form-control mb-4 col-4">
22
23             <input type="text" th:field="*{credit}"
24                 placeholder="Credits" class="form-control mb-4 col-4">
25
26             <input type="text" th:field="*{Hours}"
27                 placeholder="Total hours in a semester" class="form-control mb-4 col-4">
28
29             <input type="text" th:field="*{Weeks}"
30                 placeholder="Total weeks in a semester" class="form-control mb-4 col-4">
31
32
33             <button type="submit" class="btn btn-info col-2"> Save Schedule</button>
34         </form>
35
36         <hr>
37
38         <a th:href = "@{/subject}"> Back to Subject List</a>
39     </div>
40 </body>
41 </html>
```

User Manual

We use URL: <http://localhost:9900/schedule/main>



Schedule Management System

Teacher Schedule Management

Manage schedules

[Go to Teacher Schedules](#)

Course Information Management

Manage course information

[Go to Course Information](#)

Teacher Schedule Management System

[Add Schedule](#)

Id or Teacher's Name [Filter](#)

ID	Day Of Week	Time Slot	Teacher Name	Subject Name	Room Number	Actions
1	Monday	7-11	Jeny	Programming	109F	Update Delete
6	Monday	9:00-11:00	Na standa	TP OOP	A309	Update Delete
9	Wednesday	7:00-9:00	Phouk Punna	Statistics	E-learning	Update Delete

[Go Back to Main Page](#)

Courses Information

[Add Subject](#)

Subject [Filter](#)

ID	Subject	Credit	Hours	Weeks	Action
1	Mathematical modeling	3	32	16	Update Delete
2	OOP	3	32	16	Update Delete
4	Linear Algebra	3	24 Hours	12 Weeks	Update Delete
7	Optimization	3	32	12	Update Delete

[Go Back to Main Page](#)

User Manual

Save All

Day Of Week

Time Slot (9:00-10:00)

Teacher Name

Subject Name

Room Number

Save Schedule

[Back to Schedule List](#)

Teacher Schedule Management System

Add Schedule

Id or Teacher's Name Filter For search

ID	Day Of Week	Time Slot	Teacher Name	Subject Name	Room Number	Actions
1	Monday	7-11	Jeny	Programming	109F	Update Delete
6	Monday	9:00-11:00	Na standa	TP OOP	A309	Update Delete
9	Wednesday	7:00-9:00	Phouk Punna	Statistics	E-learning	Update Delete

[Go Back to Main Page](#)

Update Schedule

Monday

7-11

Jeny

Programming

109F

Update

[Back to Schedule List](#)

localhost:9900 says

Are you sure you want to delete this schedule information?

OK **Cancel**

User Manual

Save All

Subject

Credits

Total hours in a semester

Total weeks in a semester

Save Schedule

[Back to Subject List](#)

Courses Information

Add Subject

Subject Filter For search

ID	Subject	Credit	Hours	Weeks	Action
1	Mathematical modeling	3	32	16	Update Delete
2	OOP	3	32	16	Update Delete
4	Linear Algebra	3	24 Hours	12 Weeks	Update Delete
7	Optimization	3	32	12	Update Delete

Go Back to Main Page

Update Subject

Mathematical modeling

3

32

16

Update

[Back to Subject List](#)

localhost:9900 says

Are you sure you want to delete this subject's detail?

OK **Cancel**

Conclusion

To sum up, we have implemented API, Databases, and also a little bit of HTML into our project. In the future, we hope to improve and update the necessary functions so that this project will be useful for any kind of work. This is a wonderful and learning experience for our team.



Scan. Pay. Done.

Any Question????



1\$ per question.

Research Presentation End