

Q) Java Coding:

```
*SeasonalActivityOrganizer.java ×
1 package com.example.organizer;
2
3 import java.util.EnumSet;
4
5 import java.util.Set;
6
7 public class SeasonalActivityOrganizer {
8
9     // Enum for Seasons
10    public enum Season {
11        SPRING, SUMMER, AUTUMN, WINTER
12    }
13
14    // Enum for Activities
15    public enum Activity {
16        HIKING, SWIMMING, SKIING, PUMPKIN_CARVING
17    }
18
19    // EnumSets for each season to store activities
20    private EnumSet<Activity> springActivities = EnumSet.noneOf(Activity.class);
21    private EnumSet<Activity> summerActivities = EnumSet.noneOf(Activity.class);
22    private EnumSet<Activity> autumnActivities = EnumSet.noneOf(Activity.class);
23    private EnumSet<Activity> winterActivities = EnumSet.noneOf(Activity.class);
24
25    // Get activities for a specific season
26    public Set<Activity> getActivitiesForSeason(Season season) {
27        switch (season) {
28            case SPRING:
29                return springActivities;
30            case SUMMER:
31                return summerActivities;
32            case AUTUMN:
33                return autumnActivities;
34            case WINTER:
35                return winterActivities;
36            default:
37                throw new IllegalArgumentException("Unknown season: " + season);
38        }
39    }
}
```

```
*SeasonalActivityOrganizer.java ×
39    }
40
41    // Add activity for a specific season
42    public Set<Activity> addActivityForSeason(Activity activity, Season season) {
43        switch (season) {
44            case SPRING:
45                springActivities.add(activity);
46                return springActivities;
47            case SUMMER:
48                summerActivities.add(activity);
49                return summerActivities;
50            case AUTUMN:
51                autumnActivities.add(activity);
52                return autumnActivities;
53            case WINTER:
54                winterActivities.add(activity);
55                return winterActivities;
56            default:
57                throw new IllegalArgumentException("Unknown season: " + season);
58        }
59    }
60
61    // Remove activity from all seasons
62    public void removeActivityFromAllSeasons(Activity activity) {
63        springActivities.remove(activity);
64        summerActivities.remove(activity);
65        autumnActivities.remove(activity);
66        winterActivities.remove(activity);
67    }
68
69    // Get all activities from all seasons
70    public Set<Activity> getAllActivities() {
71        EnumSet<Activity> allActivities = EnumSet.noneOf(Activity.class);
72        allActivities.addAll(springActivities);
73        allActivities.addAll(summerActivities);
74        allActivities.addAll(autumnActivities);
75        allActivities.addAll(winterActivities);
76        return allActivities;
77    }
}
```

```

77     }
78
79     public static void main(String[] args) {
80         SeasonalActivityOrganizer organizer = new SeasonalActivityOrganizer();
81
82         // Adding activities to seasons
83         organizer.addActivityForSeason(Activity.HIKING, Season.SPRING);
84         organizer.addActivityForSeason(Activity.SWIMMING, Season.SUMMER);
85         organizer.addActivityForSeason(Activity.SKIING, Season.WINTER);
86
87         // Getting all activities
88         System.out.println(organizer.getAllActivities());
89
90         // Getting activities for specific seasons
91         System.out.println(organizer.getActivitiesForSeason(Season.SPRING));
92         System.out.println(organizer.getActivitiesForSeason(Season.SUMMER));
93         System.out.println(organizer.getActivitiesForSeason(Season.WINTER));
94
95         // Removing an activity from all seasons
96         organizer.removeActivityFromAllSeasons(Activity.HIKING);
97
98         // Getting activities for Spring after removal
99         System.out.println(organizer.getActivitiesForSeason(Season.SPRING));
100     }
101 }

```

Problems Servers Terminal Data Source Explorer Properties Console ×

```

<terminated> SeasonalActivityOrganizer [Java Application] C:\Users\meghn.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_2
[HIKING, SWIMMING, SKIING]
[HIKING]
[SWIMMING]
[SKIING]
[]

```

Q) Spring Jpa:

```

1 package com.example.curtainapp;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 public class CurtainappApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(CurtainappApplication.class, args);
10    }
11 }
12
13 }

```

```
*SeasonalActivityOrganizer.java  CurtainappApplication.java  CurtainController.java ×
1 package com.example.curtainapp.controller;
2
3 import com.example.curtainapp.entity.Curtain;
4
11
12 @RestController
13 public class CurtainController {
14
15     @Autowired
16     private CurtainRepository curtainRepository;
17
18     @GetMapping("/curtains/price")
19     public List<Curtain> getCurtainsByPriceGreaterThan(@RequestParam int price) {
20         return curtainRepository.getCurtainsByPriceGreaterThan(price);
21     }
22
23     @GetMapping("/curtains/brand")
24     public List<Curtain> getCurtainsByBrand(@RequestParam String brand) {
25         return curtainRepository.getCurtainsByBrand(brand);
26     }
27 }
28
```

```
*SeasonalActivityOrganizer.java  CurtainappApplication.java  CurtainController.java  *Curtain.java ×
1 package com.example.curtainapp.entity;
2
3 import jakarta.persistence.Entity;
4
5
6 import jakarta.persistence.GeneratedValue;
7 import jakarta.persistence.GenerationType;
8 import jakarta.persistence.Id;
9
10 @Entity
11 public class Curtain {
12
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Long id;
16     private String brand;
17     private int price;
18
19     // Getters and setters
20
21     public Long getId() {
22         return id;
23     }
24
25     public void setId(Long id) {
26         this.id = id;
27     }
28
29     public String getBrand() {
30         return brand;
31     }
32
33     public void setBrand(String brand) {
34         this.brand = brand;
35     }
36
```

```

36
37 public int getPrice() {
38     return price;
39 }
40
41 public void setPrice(int price) {
42     this.price = price;
43 }
44
45 @Override
46 public String toString() {
47     return "Curtain{" +
48         "id=" + id +
49         ", brand='" + brand + '\'' +
50         ", price=" + price +
51         '}';
52 }
53 }
54

```

```

SeasonalActivityOrganizer.java  CurtainappApplication.java  CurtainController.java  *Curtain.java  *CurtainRepository.java x
1 package com.example.curtainapp.repository;
2
3 import com.example.curtainapp.entity.Curtain;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 import org.springframework.data.jpa.repository.Query;
7 import org.springframework.data.repository.query.Param;
8
9 import java.util.List;
10
11 public interface CurtainRepository extends JpaRepository<Curtain, Long> {
12
13     @Query("SELECT c FROM Curtain c WHERE c.price > :price")
14     List<Curtain> getCurtainsByPriceGreaterThan(@Param("price") int price);
15
16     @Query("SELECT c FROM Curtain c WHERE c.brand = :brand")
17     List<Curtain> getCurtainsByBrand(@Param("brand") String brand);
18 }
19

```

GET http://localhost:8080/curtains/brand?brand=IKEA

HTTP **http://localhost:8080/curtains/brand?brand=IKEA** Save

GET http://localhost:8080/curtains/brand?brand=IKEA Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	brand	IKEA	
	Key	Value	

Body Cookies Headers (5) Test Results 200 OK 54 ms 236 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "brand": "IKEA",
5     "price": 150
6   },
7   {
8     "id": 3,
9     "brand": "IKEA",
10    "price": 90
11  }
12 ]
```

GET http://localhost:8080/curtains/price?price=100

HTTP **http://localhost:8080/curtains/price?price=100** Save

GET http://localhost:8080/curtains/price?price=150 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	price	150	
	Key	Value	

Body Cookies Headers (5) Test Results 200 OK 11 ms 201 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "brand": "IKEA",
5     "price": 150
6   }
7 ]
```


Q) Spring Boot Full Stack:

```
SeasonalActivityOrgan...  CurtainappApplicationj...  CurtainController.java  Curtain.java  CurtainRepository.java  ShipmentManagementA... ×
1 package com.example.shipment_management;
2
3 import org.springframework.boot.SpringApplication;
4
5
6 @SpringBootApplication
7 public class ShipmentManagementApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(ShipmentManagementApplication.class, args);
11     }
12
13 }
14
```

```
ShipmentManagementApplication.java ×  Shipment.java ×
1 package com.example.shipment_management.model;
2
3 import jakarta.persistence.Entity;
4
5
6 @Entity
7 public class Shipment {
8     @Id
9     @GeneratedValue(strategy = GenerationType.IDENTITY)
10     private Integer shipId;
11     private String trackNo;
12     private String origin;
13     private String destination;
14     private String status;
15
16     // Getters and Setters
17     public Integer getShipId() {
18         return shipId;
19     }
20
21     public void setShipId(Integer shipId) {
22         this.shipId = shipId;
23     }
24
25     public String getTrackNo() {
26         return trackNo;
27     }
28
29     public void setTrackNo(String trackNo) {
30         this.trackNo = trackNo;
31     }
32
33     public String getOrigin() {
34         return origin;
35     }
36
37     public void setOrigin(String origin) {
38         this.origin = origin;
39     }
40
41 }
```

```
42
43- public String getDestination() {
44     return destination;
45 }
46
47- public void setDestination(String destination) {
48     this.destination = destination;
49 }
50
51- public String getStatus() {
52     return status;
53 }
54
55- public void setStatus(String status) {
56     this.status = status;
57 }
58 }
59
```

```
ShipmentManagementApplication.java  Shipment.java  ShipmentRepository.java ×
1 package com.example.shipment_management.repository;
2
3+ import com.example.shipment_management.model.*;
6
7 @Repository
8 public interface ShipmentRepository extends JpaRepository<Shipment, Integer> {
9     Shipment findByTrackNo(String trackNo);
10 }
11
```

```
ShipmentManagementApplication.java Shipment.java ShipmentRepository.java ShipmentController.java ×
1 package com.example.shipment_management.controller;
2
3 import com.example.shipment_management.model.*;
4
5
6
7
8
9
10 @RestController
11 @RequestMapping("/shipment")
12 public class ShipmentController {
13
14     @Autowired
15     private ShipmentService shipmentService;
16
17     @GetMapping("/{trackNo}")
18     public ResponseEntity<Shipment> getShipmentByTrackNo(@PathVariable String trackNo) {
19         Shipment shipment = shipmentService.getShipmentByTrackNo(trackNo);
20         if (shipment != null) {
21             return new ResponseEntity<>(shipment, HttpStatus.OK);
22         } else {
23             return new ResponseEntity<>(HttpStatus.NOT_FOUND);
24         }
25     }
26
27     @PostMapping
28     public ResponseEntity<Shipment> createShipment(@RequestBody Shipment shipment) {
29         Shipment createdShipment = shipmentService.createShipment(shipment);
30         return new ResponseEntity<>(createdShipment, HttpStatus.CREATED);
31     }
32
33     @DeleteMapping("/{shipId}")
34     public ResponseEntity<String> deleteShipmentById(@PathVariable Integer shipId) {
35         String response = shipmentService.deleteShipmentById(shipId);
36         if (response.contains("deleted")) {
37             return new ResponseEntity<>(response, HttpStatus.OK);
38         } else {
39             return new ResponseEntity<>(response, HttpStatus.NOT_FOUND);
40         }
41     }
42 }
43
```



```

ShipmentManagementApplication.java  Shipment.java  ShipmentRepository.java  ShipmentController.java  ShipmentService.java x
1  package com.example.shipment_management.service;
2
3  import com.example.shipment_management.model.*;
4
5
6
7
8
9
10 @Service
11 public class ShipmentService {
12
13     @Autowired
14     private ShipmentRepository shipmentRepository;
15
16     public Shipment getShipmentByTrackNo(String trackNo) {
17         return shipmentRepository.findByTrackNo(trackNo);
18     }
19
20     public String deleteShipmentById(Integer shipId) {
21         Optional<Shipment> shipmentOptional = shipmentRepository.findById(shipId);
22         if (shipmentOptional.isPresent()) {
23             shipmentRepository.deleteBy(shipId);
24             return "The requested shipId-" + shipId + " got deleted";
25         } else {
26             return "Shipment with id " + shipId + " not found";
27         }
28     }
29
30     public Shipment createShipment(Shipment shipment) {
31         return shipmentRepository.save(shipment);
32     }
33 }
34

```

GET http://localhost:8080/curtains/price?price=100

Save

HTTP http://localhost:8080/shipment/TRK002 Send

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (5) Test Results 200 OK 170 ms 254 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "shipId": 3,
3   "trackNo": "TRK002",
4   "origin": "London",
5   "destination": "Paris",
6   "status": "Pending"
7 }

```

GET http://localhost:8080/curtains/price?price=100 **DEL** http://localhost:8080/curtains/price?price=100 + ...

HTTP <http://localhost:8080/curtains/price?price=100> Save

DELETE http://localhost:8080/shipment/3 Send

Params Authorization Headers (8) Body **Pre-request Script** Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (5) Test Results 200 OK 67 ms 198 B Save Response

Pretty Raw Preview Visualize Text

1 The requested shipId-3 got deleted

```
mysql> use shipment;
Database changed
mysql> show tables;
+-----+
| Tables_in_shipment |
+-----+
| shipment            |
+-----+
1 row in set (0.00 sec)

mysql> select *from shipment;
+-----+-----+-----+-----+-----+
| ship_id | destination | origin       | status    | track_no |
+-----+-----+-----+-----+-----+
| 1       | Seattle    | San Francisco | Shipped   | TRK003   |
| 2       | Los Angles | New York      | In Transit | TRK001   |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Q) JSP Full Stack:

```
1 package com.example;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class CreateTask
7  */
8 public class CreateTask extends HttpServlet {
9     private static final long serialVersionUID = 1L;
10
11     /**
12      * @see HttpServlet#HttpServlet()
13      */
14     public CreateTask() {
15         super();
16         // TODO Auto-generated constructor stub
17     }
18
19     /**
20      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
21      */
22     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
23         // TODO Auto-generated method stub
24         response.getWriter().append("Served at: ").append(request.getContextPath());
25     }
26
27     /**
28      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
29      */
30     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
31         String title = request.getParameter("title");
32         String description = request.getParameter("description");
33         String dueDate = request.getParameter("dueDate");
34         String priority = request.getParameter("priority");
35
36         String proxyUri = System.getenv("VSCODE_PROXY_URI");
37
38         System.out.println(proxyUri);
39
40         if (title == null || title.isEmpty() || description == null || description.isEmpty() ||
41             dueDate == null || dueDate.isEmpty() || priority == null || priority.isEmpty()) {
42             String successUrl = proxyUri + "/error.jsp";
43             response.sendRedirect(successUrl);
44         } else {
45             Task task = new Task(title, description, dueDate, priority);
46             request.setAttribute("task", task);
47             String successUrl = proxyUri + "/success.jsp";
48             response.sendRedirect(successUrl);
49         }
50     }
51 }
52 }
```

CreateTask.java

```

1 package com.example;
2
3 public class Task {
4     private String title;
5     private String description;
6     private String dueDate;
7     private String priority;
8
9     public Task() {}
10
11     public Task(String title, String description, String dueDate, String priority) {
12         this.title = title;
13         this.description = description;
14         this.dueDate = dueDate;
15         this.priority = priority;
16     }
17
18     public String getTitle() {
19         return title;
20     }
21
22     public void setTitle(String title) {
23         this.title = title;
24     }
25
26     public String getDescription() {
27         return description;
28     }
29
30     public void setDescription(String description) {
31         this.description = description;
32     }
33
34     public String getDueDate() {
35         return dueDate;
36     }
37

```

```

37
38     public void setDueDate(String dueDate) {
39         this.dueDate = dueDate;
40     }
41
42     public String getPriority() {
43         return priority;
44     }
45
46     public void setPriority(String priority) {
47         this.priority = priority;
48     }
49 }
50

```

Task.java

```
CreateTask.java success.jsp HomeControll... × JspFullStac... ap
1 package com.example;
2
3 import org.springframework.stereotype.Controller;
4
5
6 @Controller
7 public class HomeController {
8
9     @RequestMapping("/")
10    public String Home() {
11        return "index.jsp";
12    }
13    @RequestMapping("/users")
14    public String Tasks() {
15        return "User.jsp";
16    }
17 }
18
```

Home Controller.java

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Error</title>
5     <style>
6         body {
7             font-family: Arial, sans-serif;
8             background-color: #f4f4f4;
9             margin: 0;
10            padding: 0;
11            display: flex;
12            justify-content: center;
13            align-items: center;
14            height: 100vh;
15            color: #333;
16        }
17
18        .container {
19            background: #fff;
20            padding: 40px;
21            border-radius: 5px;
22            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
23            text-align: center;
24        }
25
26        h1 {
27            color: #dc3545;
28            margin: 0;
29            font-size: 24px;
30        }
31
32        p {
33            margin: 20px 0 0;
34            font-size: 16px;
35        }
36
37        a {
38            display: inline-block;
```

```

1
2     p {
3         margin: 20px 0 0;
4         font-size: 16px;
5     }
6
7     a {
8         display: inline-block;
9         margin-top: 20px;
10        padding: 10px 20px;
11        background-color: #007BFF;
12        color: #fff;
13        text-decoration: none;
14        border-radius: 4px;
15        transition: background-color 0.3s;
16    }
17
18    a:hover {
19        background-color: #0056b3;
20    }
21    </style>
22 </head>
23 <body>
24     <div class="container">
25         <h1>Failed to create task. Please fill in all the required fields.</h1>
26         <p>Please go back and ensure all fields are filled out correctly.</p>
27         <a href="javascript:history.back()">Go Back</a>
28     </div>
29 </body>
30 </html>
31

```

Error.jsp

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Create Task</title>
5     <style>
6         body {
7             font-family: Arial, sans-serif;
8             background-color: #f4f4f4;
9             margin: 0;
10            padding: 0;
11            display: flex;
12            justify-content: center;
13            align-items: center;
14            height: 100vh;
15        }
16
17        h1 {
18            color: #333;
19        }
20
21        form {
22            background: #fff;
23            padding: 20px;
24            border-radius: 5px;
25            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
26            width: 300px;
27            box-sizing: border-box;
28        }
29
30        label {
31            display: block;
32            margin-bottom: 8px;
33            color: #333;
34        }
35
36        input[type="text"],
37        input[type="date"],
38        select,

```



```

29
30     label {
31         display: block;
32         margin-bottom: 8px;
33         color: #333;
34     }
35
36     input[type="text"],
37     input[type="date"],
38     select,
39     textarea {
40         width: 100%;
41         padding: 8px;
42         margin-bottom: 20px;
43         border: 1px solid #ccc;
44         border-radius: 4px;
45         box-sizing: border-box;
46     }
47
48     textarea {
49         height: 100px;
50         resize: none;
51     }
52
53     input[type="submit"] {
54         width: 100%;
55         padding: 10px;
56         background-color: #007bff;
57         border: none;
58         border-radius: 4px;
59         color: #fff;
60         font-size: 16px;
61         cursor: pointer;
62     }
63
64     input[type="submit"]:hover {
65         background-color: #0056b3;
66     }
67

```

```

67
68     .form-group {
69         margin-bottom: 15px;
70     }

```

```

70
71     </style>
72 </head>
73 <body>
74     <form action="CreateTask" method="post">
75         <h1>Create Task</h1>
76         <label for="title">Task Title:</label>
77         <input type="text" id="title" name="title"><br><br>
78
79         <label for="description">Task Description:</label>
80         <textarea id="description" name="description" ></textarea><br><br>
81
82         <label for="dueDate">Due Date:</label>
83         <input type="date" id="dueDate" name="dueDate"><br><br>
84
85         <label for="priority">Priority:</label>
86         <select id="priority" name="priority" >
87             <option value="High">High</option>
88             <option value="Medium">Medium</option>
89             <option value="Low">Low</option>
90         </select><br><br>
91
92         <input type="submit" value="Create Task" id="createTask">
93     </form>
94 </body>
95 </html>
96

```

Index.jsp

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Success</title>
5     <style>
6         body {
7             font-family: Arial, sans-serif;
8             background-color: #f4f4f4;
9             margin: 0;
10            padding: 0;
11            display: flex;
12            justify-content: center;
13            align-items: center;
14            height: 100vh;
15            color: #333;
16        }
17
18        .container {
19            background: #fff;
20            padding: 40px;
21            border-radius: 5px;
22            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
23            text-align: center;
24        }
25
26        h1 {
27            color: #28a745;
28            margin: 0;
29            font-size: 24px;
30        }
31
32        p {
33            margin: 20px 0 0;
34            font-size: 16px;
35        }
36

```

```

37         a {
38             display: inline-block;
39             margin-top: 20px;
40             padding: 10px 20px;
41             background-color: #007BFF;
42             color: #fff;
43             text-decoration: none;
44             border-radius: 4px;
45             transition: background-color 0.3s;
46         }
47
48         a:hover {
49             background-color: #0056b3;
50         }
51     </style>
52 </head>
53 <body>
54     <div class="container">
55         <h1>Task Created Successfully</h1>
56         <p>Your task has been created and saved successfully.</p>
57         <a href="/">Return to Home</a>
58     </div>
59 </body>
60 </html>

```

Success.jsp

Ouput

Create Your Task

Enter the Task Title:

Enter the Task Description:

Enter the Due Date:

Priority:

Task Created Successfully

Task Details:

Title: Alarm Setup

Description: It's very good

Due Date: 2024-07-05

Priority: Medium

[Back to Create Task](#)