



# UniLog

Course Registration and  
Tuition Payment  
By Kate Post and Aly  
Westwood



# HIGH LEVEL BACKGROUND INFORMATION

- Students register for Sections
- Students make payments for their Registered Sections
- Admins manage the available Courses and Sections



# OUR SOLUTION

# ADMIN VIEW

- Admins CRUD on courses and sections

## Edit Section

Semester

Spring

Year

2021

Seats Available

0

[Disable Section](#) [Discard Changes](#) [Save Changes](#)

**COUR222**  
Second in the Course Series of Courses

Get a deeper understanding of courses!

Semester: SPRING 2023

Seats Available: 25

Disabled: false

## Add a Section

[New Course](#) [New Section](#)

Course Number (7 characters):

MATH203

Semester:

Fall

Year :

2023

Seats Available:

25

[Add Section](#)

## Sections

[Semester](#)  [Fetch Results](#)

**COUR123**  
A first look at courses  
Seats Available: 35  
Semester: FALL 2022  
\$75.0  
Disabled Status: false

**COUR123**  
A first look at courses  
Seats Available: 35  
Semester: SPRING 2021  
\$75.0  
Disabled Status: false

**COUR222**  
Second in the Course Series of Courses  
Seats Available: 25  
Semester: SPRING 2023  
\$85.0  
Disabled Status: false

**COUR123**  
A first look at courses

Semester: FALL 2022

Students Registered: 4/35

Students registered for this section

Kate Post	Paid
Oli Stud	Unpaid
Aly Westwood	Paid
Vlad Magda	Paid

[Edit Section](#)

# ADMIN VIEW

- Admin user CRUD
- Admin view and delete registrations

## Registrations

							Filter by ▼
Student		Course Code	Semester	Due Date	Payment		
Post, Kate	kpost19@gmail.com	COUR123	FALL 2022	2023-01-01	PAID 2021-12-15	<input type="checkbox"/>	
Post, Kate	kpost19@gmail.com	COUR123	SUMMER 2022	2022-10-01	PAID 2021-12-16	<input type="checkbox"/>	
Stud, Oli*	oli@student.ca	COUR123	FALL 2022	2023-01-01	OUTSTANDING	<input type="checkbox"/>	
Westwood, Aly	aly@student.ca	TEST123	SUMMER 2022	2022-10-01	PAID 2021-12-16	<input type="checkbox"/>	
Westwood, Aly	aly@student.ca	COUR123	SUMMER 2022	2022-10-01	OUTSTANDING	<input type="checkbox"/>	
Post, Kate	kpost19@gmail.com	COUR222	SPRING 2023	2023-07-01	PAID 2021-12-16	<input type="checkbox"/>	
Post, Kate	kpost19@gmail.com	TEST123	SUMMER 2022	2022-10-01	PAID 2021-12-16	<input type="checkbox"/>	
							Delete

## All Users

ID	First Name	Last Name	Email	Address	Date of Birth	Role
1	Kate	Post	kpost19@gmail.com	1234 vancouver st washington	2000-02-01	STUDENT
2	Kate	Post	kathryn.post@johnabbottcollege.net	890 main street virginia delaware	1996-12-11	ADMIN
3	Aly	Westwood	aly@admin.ca	9 rue des Doublons, Blainville QC J7C 5X4	1990-07-02	ADMIN
4	Aly	Westwood	aly@student.ca	9 rue des Doublons, Blainville QC	1990-07-02	STUDENT

# ADMIN VIEW

- Admin user CRUD
- Admin view and delete registrations

Stud, Oli (inactive)

COUR123 - FALL 2022

Student:

Email: oli@student.ca

Payment:

Owing: 75.0

Payment: OUTSTANDING

Due: 2023-01-01

Course:

Title: A first look at courses

Description: A look at courses

Prerequisite: NONE

Close

# ADMIN VIEW

- Admin user CRUD
- Admin view and delete registrations

Email	aly@student.ca	Edit	
First Name	Aly	Edit	
Last Name	Westwood	Edit	
Address	9 rue des Doublons, Blainville QC	Edit	
Date of Birth	02-Jul-1990	Edit	
Role	Student	Edit	
Reset Password	<input type="text" value="New password"/>	<input type="text" value="Confirm new password"/>	Save

Deactivate User

Email	aly@student.ca		
First Name	Aly	Edit	
Last Name	Westwood	Edit	
Address	9 rue des Doublons, Blainville QC	Edit	
Date of Birth	1990-07-02		
Reset Password	<input type="text" value="New password"/>	<input type="text" value="Confirm new password"/>	Save

# STUDENT VIEW

[Home](#)[Student Details](#)[Sign Out](#)

- Students select sections of courses to register for
- Students register through course cart
- Students view registered courses

Logged in as  
aly@student.ca

[Course cart](#)[View all courses](#)[View all sections](#)[Payments](#)[Registered courses](#)

## Available Sections

### COUR123

A first look at courses

Seats Available: 35

Semester: FALL 2022

\$75.0

### COUR123

A first look at courses

Seats Available: 40

Semester: SUMMER 2022

### COUR123

A first look at courses

Semester: SPRING 2021

### COUR123

A first look at courses

A look at courses

Price: \$75.0



# STUDENT VIEW

- Students select sections of courses to register for
- Students register through course cart
- Students view registered courses

[Home](#)[Student Details](#)[Sign Out](#)

Section added  
to course cart  
successfully.



Logged in as  
aly@student.ca

[Course cart](#)[View all courses](#)

## Cart

Course Code	Title	Description	Semester	Price
COUR123	A first look at courses	A look at courses	SPRING 2021	\$75.0
			<b>Subtotal</b>	<b>\$75.0</b>

[Register For Courses](#)

# STUDENT VIEW

- Students select courses to pay for in registered courses
- Students make payments
- Students view past payments

Filter by

Course Code	Title	Description	Semester	Due Date	Owing	
TEST123	Testing101	An in-depth first look at how to test. In this course you will learn how to test tests while testing	SUMMER 2022	2022-10-01	PAID	
COUR123	A first look at courses	A look at courses	SUMMER 2022	2022-10-01	\$75.0	<input checked="" type="checkbox"/>
COUR222	Second in the Course Series of Courses	Get a deeper understanding of courses!	SPRING 2023	2023-07-01	\$85.0	<input type="checkbox"/>
COUR123	A first look at courses	A look at courses	FALL 2022	2023-01-01	PAID	
COUR345	Course3	Course3	SPRING 2022	2022-07-01	\$75.99	<input checked="" type="checkbox"/>

Delete

Pay \$150.99

# STUDENT VIEW

- Students select courses to pay for in registered courses
- Students make payments
- Students view past payments

## Make A Payment

### Summary

Clear Cart

Course Code	Semester	Price
COMP248	SUMMER 2022	\$250.0
Total		\$250.0

### Payment

Enter your payment information

Submit Payment

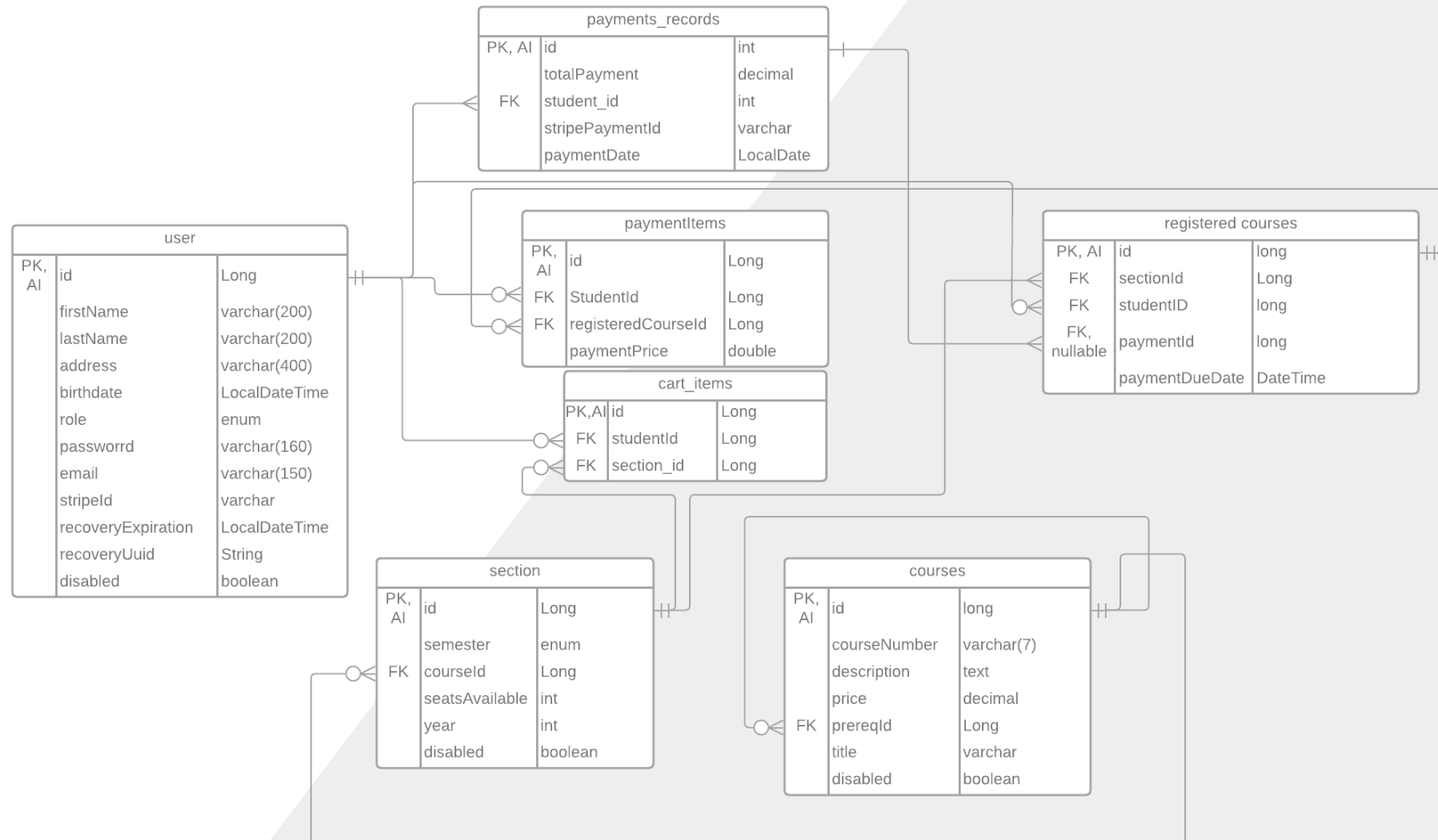
## Payments

Amount	Date Paid	Sections
\$75.0	2021-12-18	Sections Paid ▾
\$250.0	2021-12-19	Sections Paid ▾
COMP248 SUMMER 2022		



# CHALLENGES AND THEIR SOLUTIONS

# DATABASE STRUCTURE



# CHALLENGE : SORTING AND FILTERING

- Sorting using sort class

```
@GetMapping("/userList")
public String viewUsers(Model model, @RequestParam(value = "sortBy", required = false)String sortBy){
    if(sortBy == null){
        sortBy = "id";
    }
    model.addAttribute("attributeName: userList", userService.findAll(Sort.by(sortBy)));
    return "admin/userList";
}
```

ID	First Name	Last Name	Email	Address	Date of Birth	Role
1	Kate	Post	kpost19@gmail.com	1234 vancouver st washington	2000-02-01	STUDENT
2	Kate	Post	kathryn.post@johnabbottcollege.net	890 main street virginia	1996-12-11	ADMIN

# CHALLENGE : SORTING AND FILTERING

- Targetted queries using @Query annotation

```
@Query(value = "SELECT r FROM RegisteredCourse r WHERE r.paymentRecord IS NULL and r.user = ?1")
Optional<List<RegisteredCourse>> findUnpaidByUser(User user);

@Query(value = "SELECT r FROM RegisteredCourse r WHERE r.paymentRecord IS NOT NULL and r.user = ?1")
Optional<List<RegisteredCourse>> findPaidByUser(User user);

@Query(value = "SELECT r FROM RegisteredCourse r WHERE r.dueDate < CURRENT_DATE and r.user = ?1")
Optional<List<RegisteredCourse>> findPastByUser(User user);

@Query(value = "SELECT r FROM RegisteredCourse r WHERE r.dueDate > CURRENT_DATE and r.user = ?1")
Optional<List<RegisteredCourse>> findUpcomingByUser(User user);
```

# CHALLENGE : HEROKU

The screenshot shows the Heroku dashboard for the 'uni-log' application. At the top, there's a header with a user profile icon, a dropdown arrow, a greater-than sign, the app name 'uni-log', a star icon, and two buttons: 'Open app' and 'More' with a dropdown arrow. Below this is a sub-header with the repository 'KatePost/unilog'. A navigation bar contains links for 'Overview', 'Resources' (which is active), 'Deploy', 'Metrics', 'Activity', 'Access', and 'Settings'. The main content area shows 'Free Dynos' with a 'Change Dyno Type' button. Below this is a dyno configuration bar for a 'web' dyno with the command 'java -Dserver.port=\$PORT \$JAVA\_OPTS -jar target/unilog-0.0...', a toggle switch, '\$0.00', and 'Confirm' and 'Cancel' buttons. An 'Add-ons' section has a 'Find more add-ons' button. At the bottom, there's a 'Maintenance Mode' section with a description, a link to 'More info', and a toggle switch labeled 'Maintenance mode is off'.

uni-log

KatePost/unilog

Overview Resources Deploy Metrics Activity Access Settings

Free Dynos [Change Dyno Type](#)

web java -Dserver.port=\$PORT \$JAVA\_OPTS -jar target/unilog-0.0... \$0.00 [Confirm](#) [Cancel](#)

Add-ons [Find more add-ons](#)

**Maintenance Mode**

If you need to take your app offline you can turn on maintenance mode. [More info](#)

Maintenance mode is off



# CHALLENGE : HEROKU

App is called uni-log

- To stop program and turn on maintenance mode

```
$ heroku maintenance:on -a uni-log
```

```
$ heroku ps:scale web=0 -a uni-log
```

- To start the program and turn off maintenance mode

```
$ heroku maintenance:off -a uni-log
```

```
$ heroku ps:scale web=1 -a uni-log
```

# CHALLENGE : AJAX LOAD THYMELEAF FRAGMENTS

- Using AJAX load to dynamically fetch data formatted in Thymeleaf Fragments

## Available Courses

Sort by ▾

Search

Fetch Results

<b>COMP248</b> Intro To Object Oriented Programming Learn the fundamentals of Object Oriented Programming with Java
<b>COUR123</b> A first look at courses A look at courses

```
th:each="course : ${listCourses}" th:attr="onclick=|displayCourse('${course.courseNumber}')|" >
```

## Available Courses

Sort by ▾

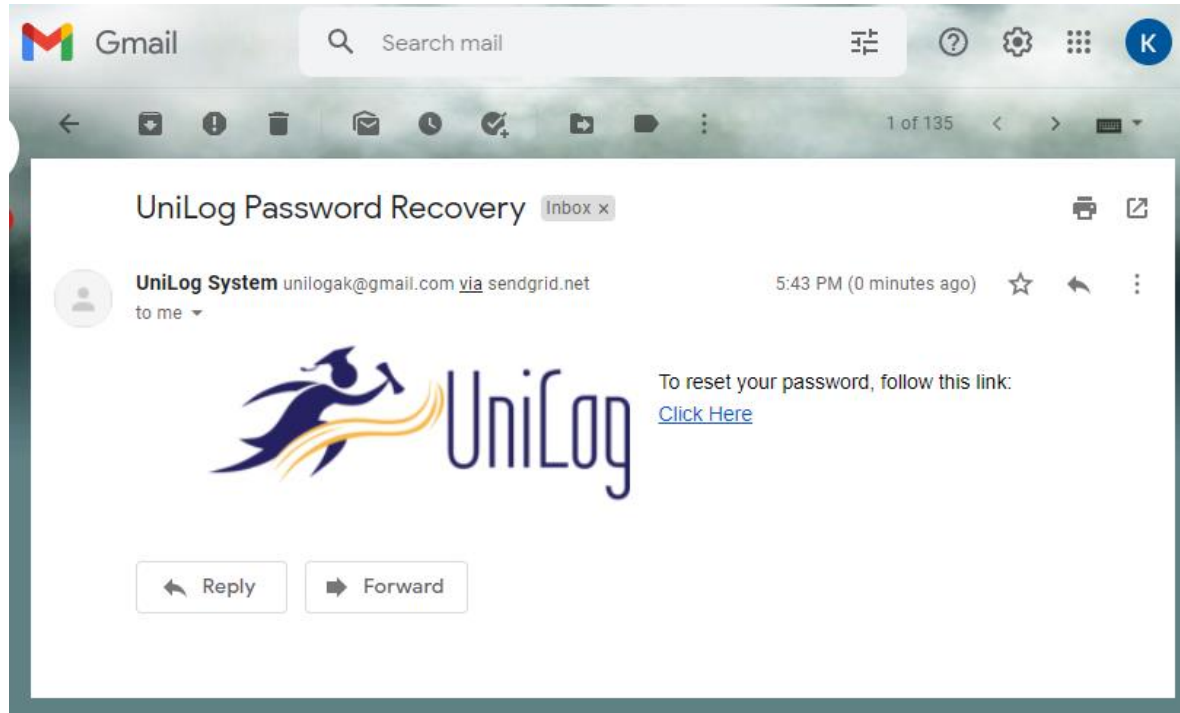
Search

Fetch Results

<b>COMP248</b> Intro To Object Oriented Programming Learn the fundamentals of Object Oriented Programming with Java	<b>COMP248</b> <b>Intro To Object Oriented Programming</b> Learn the fundamentals of Object Oriented Programming with Java Price: \$250.0 <a href="#">View Sections for this course</a>
<b>COUR123</b> A first look at courses A look at courses	

```
function displayCourse(courseNumber) {  
    let url = '/admin/singleCourse/' + courseNumber;  
    $("#singleCourse").load(url);  
}
```

# CHALLENGE : PASSWORD RECOVERY



- Users are sent a link to their unique password recovery page

## Recover Password

Email Address

kpost19@gmail.com

Enter your new  
password

.....

Repeat your new  
password

.....

Recover password

# CHALLENGE : PAYMENT PROCESSING

## Make A Payment

### Summary

Clear Cart

Course Code	Semester	Price
COUR123	SUMMER 2022	\$75.0
Total		\$75.0

### Payment

Enter your payment information

VISA 4242 4242 4242 4242 04 / 22 123 12866

Submit Payment

- The student is shown their cart items and the total, and is asked to input their payment information

Payment of \$75.0 successful X

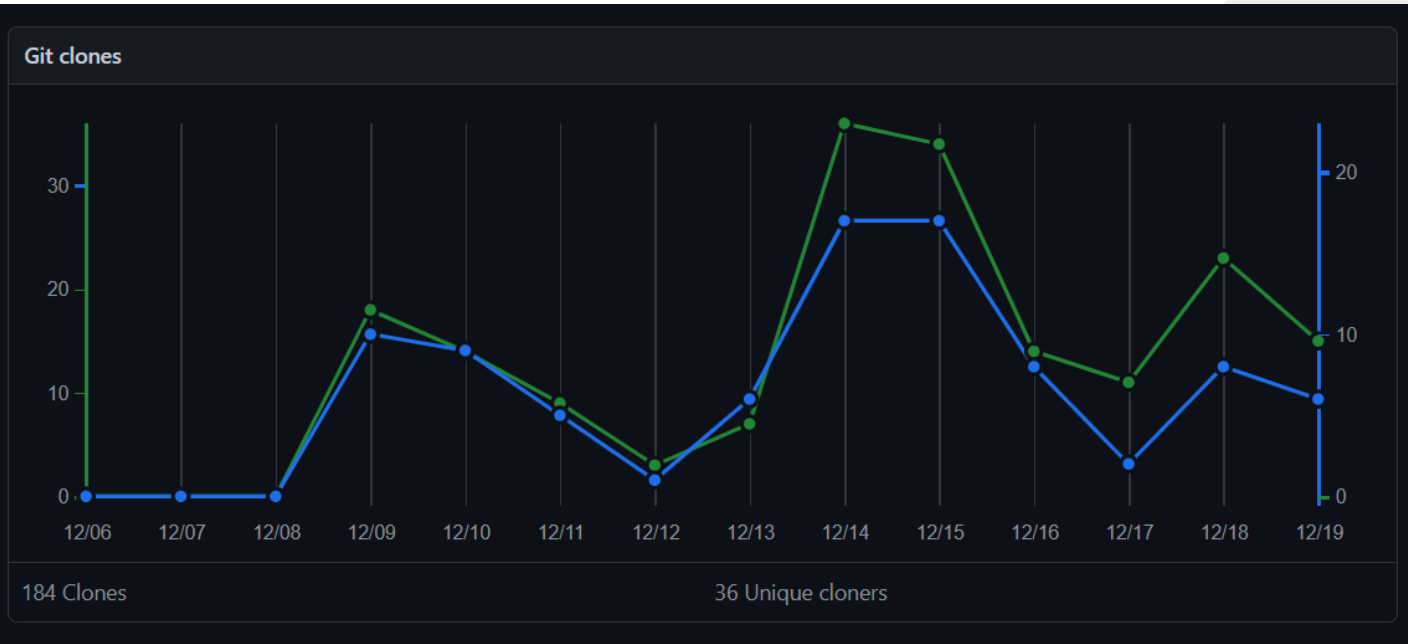
# CHALLENGE : PUBLIC REPOSITORY

- Hiding secrets from GitHub in a public repository

```
private final String STRIPE_SECRET_KEY = System.getenv(name: "STRIPE_SECRET_KEY");
```

User environment variables:

Name	Value
STRIPE_PUBLIC_KEY	pk_test_51K6PnvA17XnbPt50VsqPq...
STRIPE_SECRET_KEY	sk_test_51K6PnvA17XnbPt50bAvU4...
SENDGRID_API_KEY	SG.1UeSrt1cRsOARAM4yrVy4g.kKri...



Config Vars

Hide Config Vars

DATABASE_URL	postgres://tzyjidrmruhxp...		
SENDGRID_API_KEY	SG.1UeSrt1cRsOARAM4yrV...		
STRIPE_PUBLIC_KEY	pk_test_51K6PnvA17XnbP...		
STRIPE_SECRET_KEY	sk_test_51K6PnvA17XnbP...		
KEY	VALUE		



# WHAT WE LEARNED

# THYMELEAF

## Literals

- Before

```
<td><span th:text="${regCourse.section.semester}"></span> <span th:text="${regCourse.section.year}"></span></td>
```

- After

```
<td th:text="|${regCourse.section.semester} ${regCourse.section.year}|"></td>
```

## Conditionals

if `th:if="${otherUser}">`

unless `th:unless="${user.disabled}">`

Boolean statements

```
th:class="${user.disabled}? 'hstack justify-content-center' : 'hstack justify-content-end'"
```

```
th:class="${user.disabled}? 'text-light table-active'";
```

```
<tr th:each="user : ${userList}" th:id="|user${user.id}|" th:onclick="|goToUser(${user.id})|" th:class="${user.disabled}? 'text-light table-active'">
```

# THYMELEAF ATTRIBUTES

th:abbr	th:accept	th:accept-charset	th:name	th:onabort	th:onafterprint	th:onresize	th:onscroll	th:onseeked
th:accesskey	th:action	th:align	th:onbeforeprint	th:onbeforeunload	th:onblur	th:onseeking	th:onselect	th:onshow
th:alt	th:archive	th:audio	th:oncanplay	th:oncanplaythrough	th:onChange	th:onstalled	th:onstorage	th:onsubmit
th:autocomplete	th:axis	th:background	th:onclick	th:oncontextmenu	th:ondbclick	th:onsuspend	th:ontimeupdate	th:onundo
th:bgcolor	th:border	th:cellpadding	th:ondrag	th:ondragend	th:ondragenter	th:onunload	th:onvolumechange	th:onwaiting
th:cellspacing	th:challenge	th:charset	th:ondragleave	th:ondragover	th:ondragstart	th:optimum	th:pattern	th:placeholder
th:cite	th:class	th:classid	th:ondrop	th:ondurationchange	th:onemptied	th:poster	th:preload	th:radiogroup
th:codebase	th:codetype	th:cols	th:onended	th:onerror	th:onfocus	th:rel	th:rev	th:rows
th:colspan	th:compact	th:content	th:onformchange	th:onforminput	th:onhashchange	th:rowspan	th:rules	th:sandbox
th:contenteditable	th:contextmenu	th:data	th:oninput	th:oninvalid	th:onkeydown	th:scheme	th:scope	th:scrolling
th:datetime	th:dir	th:draggable	th:onkeypress	th:onkeyup	th:onload	th:size	th:sizes	th:span
th:dropzone	th:enctype	th:for	th:onloadeddata	th:onloadedmetadata	th:onloadstart	th:spellcheck	th:src	th:srclang
th:form	th:formaction	th:formenctype	th:onmessage	th:onmousedown	th:onmousemove	th:standby	th:start	th:step
th:formmethod	th:formtarget	th:fragment	th:onmouseout	th:onmouseover	th:onmouseup	th:style	th:summary	th:tabindex
th:frame	th:frameborder	th:headers	th:onmousewheel	th:onoffline	th:ononline	th:target	th:title	th:type
th:height	th:high	th:href	th:onpause	th:onplay	th:onplaying	th:usemap	th:value	th:valuetype
th:hreflang	th:hspace	th:http-equiv	th:onpopstate	th:onprogress	th:onratechange	th:vspace	th:width	th:wrap
th:icon	th:id	th:inline	th:onreadystatechange	th:onredo	th:onreset	th:xmlbase	th:xmllang	th:xmlspace
th:keytype	th:kind	th:label						
th:lang	th:list	th:longdesc						
th:low	th:manifest	th:marginheight						
th:marginwidth	th:max	th:maxlength						
th:media	th:method	th:min						



# BOOTSTRAP SCSS

- Download Sass
- Download bootstrap
- Add custom settings to bootstrap.scss
- Run the sass command

```
6  */
7  @import "functions";
8
9  $primary: #009;
10 $secondary: #goldenrod;
11 $warning: #DD0;
12
13 // scss-docs-start import-stack
14 // Configuration
15 @import "variables";
16 @import "mixins";
17 @import "utilities";
18
```

\$ sass [starting-path]/bootstrap.scss:[target-path]/bootstrap.css

# EMAILING PASSWORD RECOVERY

```
public void sendRecoveryMail(User user, String link) {
    Email from = new Email(email: "unilogak@gmail.com", name: "UniLog System");
    String subject = "Password reset";
    Email to = new Email(user.getEmail(), user.getFirstName());
    Content content = new Content(type: "text/html", value: "test");
    Mail mail = new Mail(from, subject, to, content);

    mail.setTemplateId("d-f763e9ea34084659a9c9de3afe1055ea");
    Personalization personalization = new Personalization();
    personalization.addDynamicTemplateData("url", "<a href=" + link + ">Click Here</a>");
    personalization.addTo(to);
    mail.personalization.add(personalization);
    Request request = new Request();
    try {
        request.setMethod(Method.POST);
        request.setEndpoint("mail/send");
        request.setBody(mail.build());
        Response response = this.sendGrid.api(request);
        System.out.println(response.getBody());
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
```

- Set up SendGrid account
- Create Dynamic Template

Subject: UniLog Password Recovery

Preheader:



To reset your password, follow this link:  
{{{url}}}

# EMAILING PASSWORD RECOVERY

```
public void recoverPassword(User user) {  
    final String uuid = UUID.randomUUID().toString().replace(target: "-", replacement: "");  
    user.setRecoveryUuid(uuid);  
    user.setRecoveryExpiration(LocalDate.now().plusMinutes(5));  
    userRepo.save(user);  
  
    String link = "https://uni-log.herokuapp.com/passwordReset/" + user.getRecoveryUuid();  
    sendGridEmailService.sendRecoveryMail(user, link);  
}
```

- Create UUID
- Validate user details

The link you're trying to access is either expired or incorrect.



Password updated successfully



## Recover Password

If an account matching that email is found, it will be emailed the next steps for password recovery



Enter the email  
address for your  
account

Recover password

# PAYMENT PROCESSING

- Using the Stripe Charge Object
- Payment records in the Stripe account

```
Stripe.apiKey = STRIPE_SECRET_KEY;  
  
User student = userService.findByEmail(principal.getName());  
double total = paymentItemService.getTotal(student);  
long totalInCents = (long)(total * 100);
```

## Payments

All   Succeeded   Refunded   Uncaptured

<input type="checkbox"/>	AMOUNT		DESCRIPTION
<input type="checkbox"/>	\$75.00	Succeeded ✓	ch_3K8Zz1A17XnbPt501sRAGgKX

```
Charge charge = Charge.create(params);  
//save chargeId and create paymentRecord  
PaymentRecord paymentRecord = checkoutService.saveCharge(charge, student, total);  
//update registered courses to reflect payment  
paymentItemService.updateKeyForRegisteredCourses(paymentRecord);  
//remove items from cart  
paymentItemService.clearCart(student);  
redirectAttributes.addFlashAttribute(attributeName: "message", attributeValue: "Payment of $" + total + " successful");  
return "redirect:/student/paymentHistory";
```

# FUTURE WORK

Pagination



DRY the code



Realistic Section Dynamics





# SUMMARY

We created a system for students to register and pay for courses and with the ability for admin oversight.



THANK YOU