

# Individual Project – Exam Branch

---

Advanced Web Development (SCS - 3112)

## Assignment Details

**Name:** K. M. K. B. Ranasinghe

**Registration Number:** 2015CS110

**Index Number:** 15001105

**Project:** Exam Registration System for Exam Branch, UCSC

**Technologies:**

- MongoDB
- Express.js
- Angular
- Nodejs

**Tools:**

- Postman
- Selenium IDE – Chrome Extension

**Users of the system:**

- Admin
- Students (Undergraduates/Postgraduates)

**Functionalities:**

Common functions

1. Login
2. Register

Admin:

1. Publish Exams (Update , Delete)
2. Publish Updates (Update, Delete)
3. View exam registrations
4. View Users
5. View repeat forms
6. Edit profile
7. Generate reports

Student:

1. Register Exams (View)
2. View Updates
3. Submit repeat forms
4. View registered exams
5. View repeat submissions
6. Edit profile

## Table of Content

Content	Pages
1. Implemented Functionalities	4 - 15
2. Testing	16 - 19
3. Problems encountered and solutions	19
4. Assumptions	19

## 01) Implemented Functionalities:

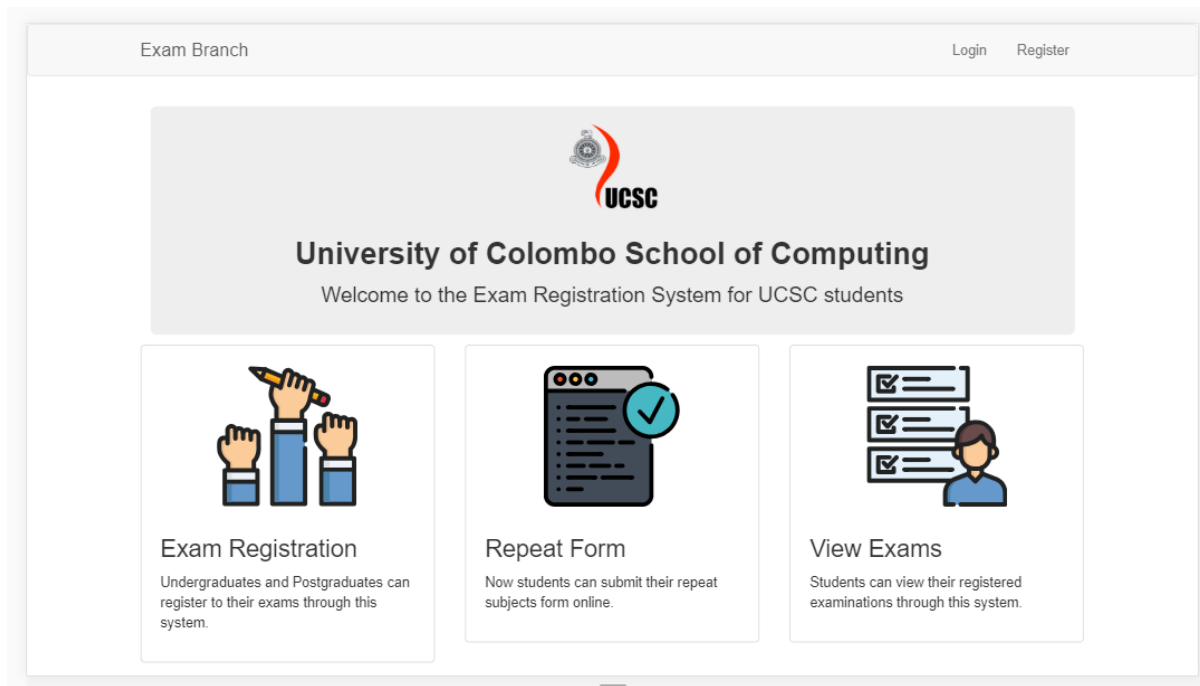


Figure 1: Home page

Common functionalities:

Admin and the students have some common functionalities in this system. They are;

- Login

The screenshot shows the login page of the system. It has a title 'Login' at the top. Below the title, there are two input fields: 'Username' and 'Password'. Each field has a small icon (a person for username, a padlock for password) to its left. Below the password field, there is a checkbox labeled 'Remember me'. At the bottom, there is a blue 'Login' button. Below the button, there is a link 'Not a member? Join now' and another link 'Forgot Password?'.

Figure 2: Login Page

Key	Value	Type
(1) ObjectId("5b30c3130cfdc155187c8d42")	{ 5 fields }	Object
_id	ObjectId("5b30c3130cfdc155187c8d42")	ObjectId
email	admin123@gmail.com	String
username	admin	String
password	\$2a\$10\$muP0hLQkcZZnrNhopglPKOMUGgtEZobK073WiZQiKDzud...	String
_v	0	Int32
(2) ObjectId("5b3336f8133bd23c74a43fc5")	{ 5 fields }	Object
(3) ObjectId("5b33987f133bd23c74a43fc6")	{ 5 fields }	Object
(4) ObjectId("5b3972c69a979b0df01c8eac")	{ 5 fields }	Object
(5) ObjectId("5b3a6e9e632f7e37600fc5c9")	{ 5 fields }	Object

Figure 3: Users collection in the database

“Users” collection has admin details. Let’s take those credentials to login to the system. As you can see all the **passwords are encrypted**.

Login

Success!

Username

admin

Password

.....

☐ Remember me

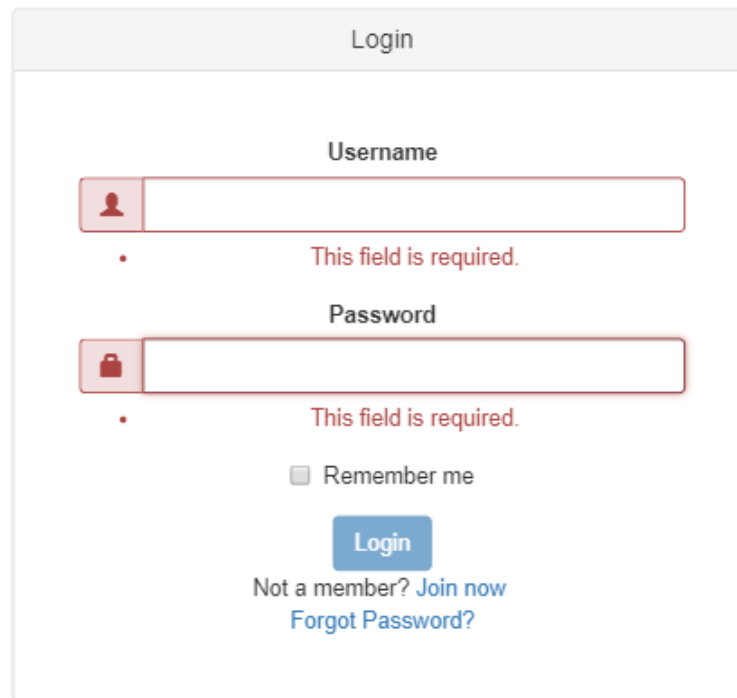
Login

Not a member? [Join now](#)

[Forgot Password?](#)

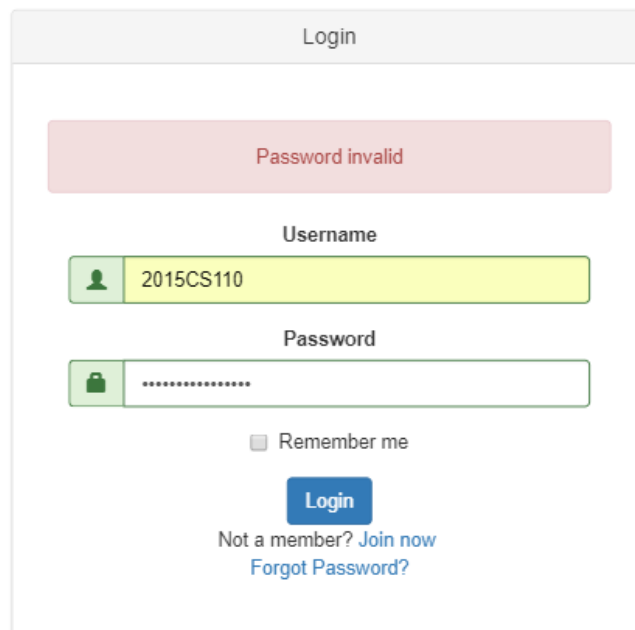
Figure 4: Successful login

When we enter valid user name and password we can successfully logged in.



The image shows a login form titled "Login". It contains two input fields: "Username" and "Password". Both fields are empty and have a red border. Below each field is a red dot and the text "This field is required." Below the password field is a checkbox labeled "Remember me". At the bottom is a blue "Login" button. Below the button are two links: "Not a member? [Join now](#)" and "[Forgot Password?](#)".

User cannot login to the system without providing valid password username. If user puts the fields empty login button moves to the “disabled” state. And the UI will show the error messages.




The image shows the same login form, but with a red error message "Password invalid" at the top. The "Username" field is now filled with "2015CS110" and has a green border. The "Password" field is filled with "\*\*\*\*\*" and has a green border. The "Remember me" checkbox is still present. The "Login" button is still blue. The links "Not a member? [Join now](#)" and "[Forgot Password?](#)" are still at the bottom.

User cannot provide invalid passwords


Login

Username not found.

Username

 admin123

Password

 .....

☐ Remember me

Login

Not a member? [Join now](#)  
[Forgot Password?](#)


If the system cannot find the username that has provided by the user then also the UI will show an error message.

- Register

Register


Account registered!

Username

 2014CS002


• Username is available

Email


 piyumika92@gmail.com

• E-mail is available

Password

 .....

Confirm Password

 .....

Submit

Have an account? [Login](#)

By giving valid username (registration number), email and password user can successfully logged in.


Key	Value	Type
> (1) ObjectId("5b30c3130cfd155187c8d42")	{ 5 fields }	Object
> (2) ObjectId("5b3336f8133bd23c74a43fc5")	{ 5 fields }	Object
> (3) ObjectId("5b33987f133bd23c74a43fc6")	{ 5 fields }	Object
> (4) ObjectId("5b3972c69a979b0df01c8eac")	{ 5 fields }	Object
▼ (5) ObjectId("5b3a6e9e632f7e37600fc5c9")	{ 5 fields }	Object
_id	ObjectId("5b3a6e9e632f7e37600fc5c9")	ObjectId
email	piyumika92@gmail.com	String
username	2014cs002	String
password	\$2a\$10\$2p/r58fxySdrMIRsLevpE.HWH35H2ho0N5HNBuzlbKOhC5h...	String
_v	0	Int32

As in the login, If user provides invalid inputs, system will restrict registration

Register


Username or e-mail already exists

Username

 2014CS002


• Username is available

Email


 kalpanibh@gmail.com

• E-mail is available

Password

 .....

Confirm Password

 .....

Submit

Have an account? [Login](#)

Register

Username or e-mail already exists

Username

 2014CS026

• Username is available

Email

 sadeepmath@gmail.com

• E-mail is already taken

Password

 .....

Confirm Password

 .....


Submit

Have an account? [Login](#)



[Exam Branch](#)
[Home](#)

[Dashboard](#)
[Exams](#)
[Logout](#)



## University of Colombo School of Computing

Welcome to the Exam Registration System for UCSC students

[Repeat Forms](#)
[Students](#)


[New Post](#)

[↻ Reload](#)

**Semester 1 2018 Exam will start on 9th of July**

Semester 1 2018 Exam will start on 9th of July. Please check the time table.

Posted by: admin  
Date: Jul 01, 2018



User level Administrator

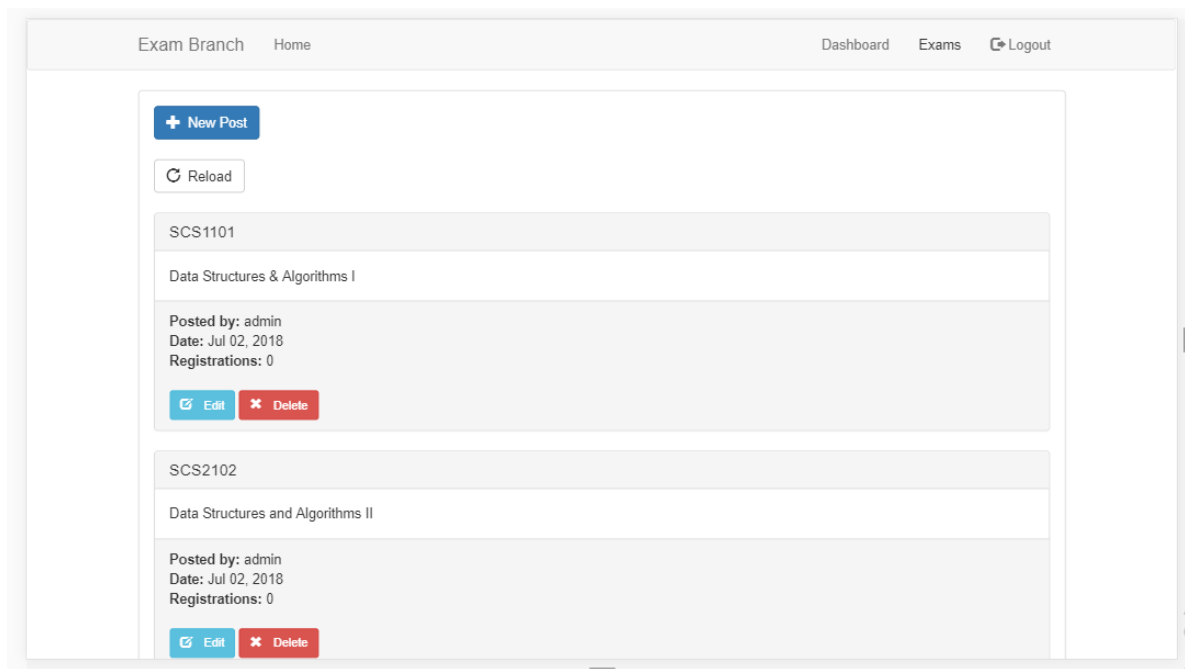
Username admin

Email admin123@gmail.com

[Change Password](#)
[✉](#)

Figure 5: Admin Dashboard

- Publish upcoming exam (publish, edit, delete)



When admin click new post, he will get this form

The screenshot shows the 'New Post' form. The top navigation bar is identical to the previous screenshot. The form contains two input fields:

- Subject Code**  
\*Subject Code
- Subject Name**  
\*Subject Name

At the bottom of the form are two buttons: 'Go Back' and 'Submit'. A footer bar at the bottom of the page contains the text: '© 2018, All rights reserved by University of Colombo School of Computing No. 35, Reid Avenue, Colombo 7, Sri Lanka.'

By entering valid details admin can add new exam to the system successfully.

Exam Branch Home Dashboard Exams Logout

Exam saved!

Subject Code  
SCS2104

Subject Name  
Mathematical Methods II

Go Back Submit

© 2018, All rights reserved by University of Colombo School of Computing No: 35, Reid Avenue, Colombo 7, Sri Lanka.

Soon after adding the exam page will reload and directed back to the exams route which is the admin Exams page. There you can see the newly added exam post.

Exam Branch Home Dashboard Exams Logout

+ New Post

Reload

SCS2104

Mathematical Methods II

Posted by: admin  
Date: Jul 02, 2018  
Registrations: 0

Edit Delete

SCS1111

Programming III

Posted by: admin  
Date: Jul 02, 2018  
Registrations: 0

Edit Delete

In the exams collection also you can see the newly added exam.

Key	Value	Type
> (1) Objectid("5b387aca1e7ba3125c95ccb4")	{ 8 fields }	Object
> (2) Objectid("5b38885440553465082ef430")	{ 8 fields }	Object
> (3) Objectid("5b38888340553465082ef431")	{ 8 fields }	Object
> (4) Objectid("5b388af6fa146f540ca481bf")	{ 8 fields }	Object
> (5) Objectid("5b3891a1a7781e666cd8c5b9")	{ 8 fields }	Object
> (6) Objectid("5b396d14bb96723c4cbfd5de")	{ 6 fields }	Object
> (7) Objectid("5b39726a9a979b0df01c8eab")	{ 8 fields }	Object
> (8) Objectid("5b3a79fa632f7e37600fc5ca")	{ 8 fields }	Object
▼ (9) Objectid("5b3a7a77632f7e37600fc5cc")	{ 8 fields }	Object
_id	Objectid("5b3a7a77632f7e37600fc5cc")	Objectid
subject_code	SCS2104	String
subject_name	Mathematical Methods II	String
createdBy	admin	String
registeredBy	[ 0 elements ]	Array
registrations	0	Int32
createdAt	2018-07-02 18:06:30.258Z	Date
_v	0	Int32

Exam Branch
Home
Dashboard
Exams
Logout

## Edit Exam

Exam Updated!

Subject Code

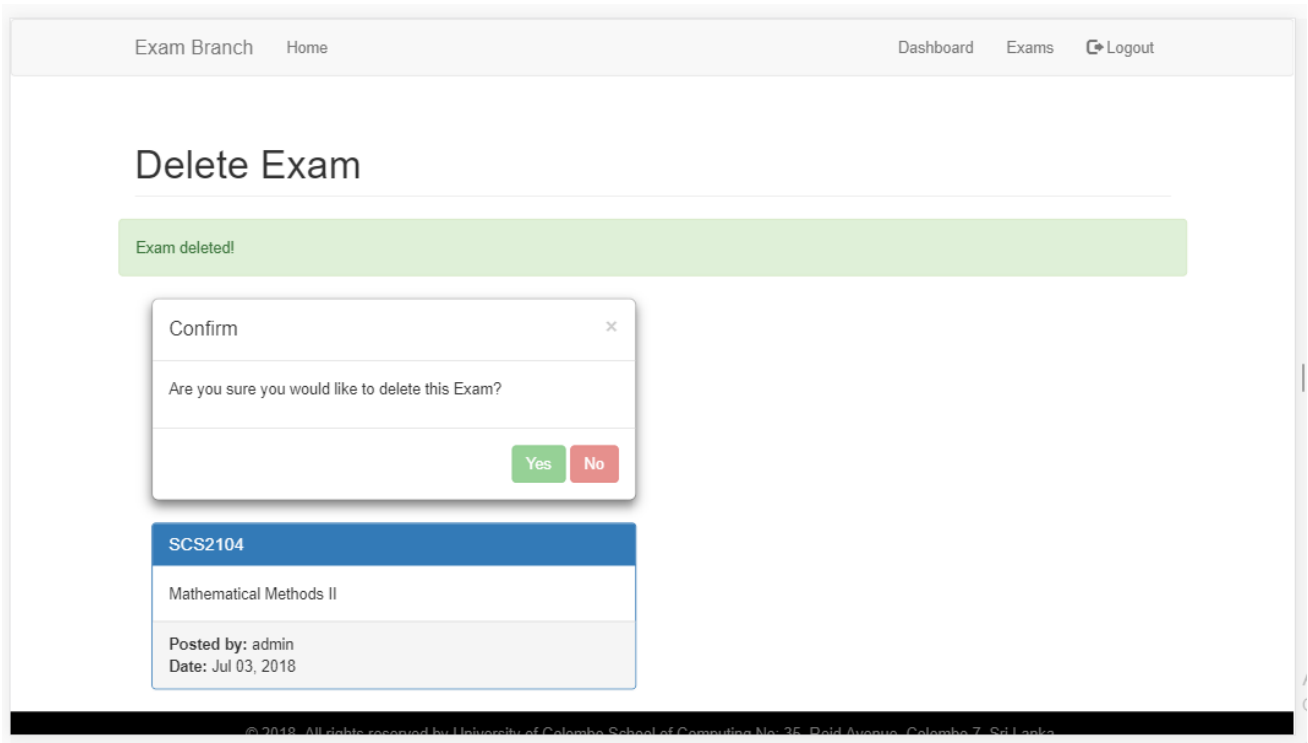
SCS2104

Subject Name

Mathematical Methods II

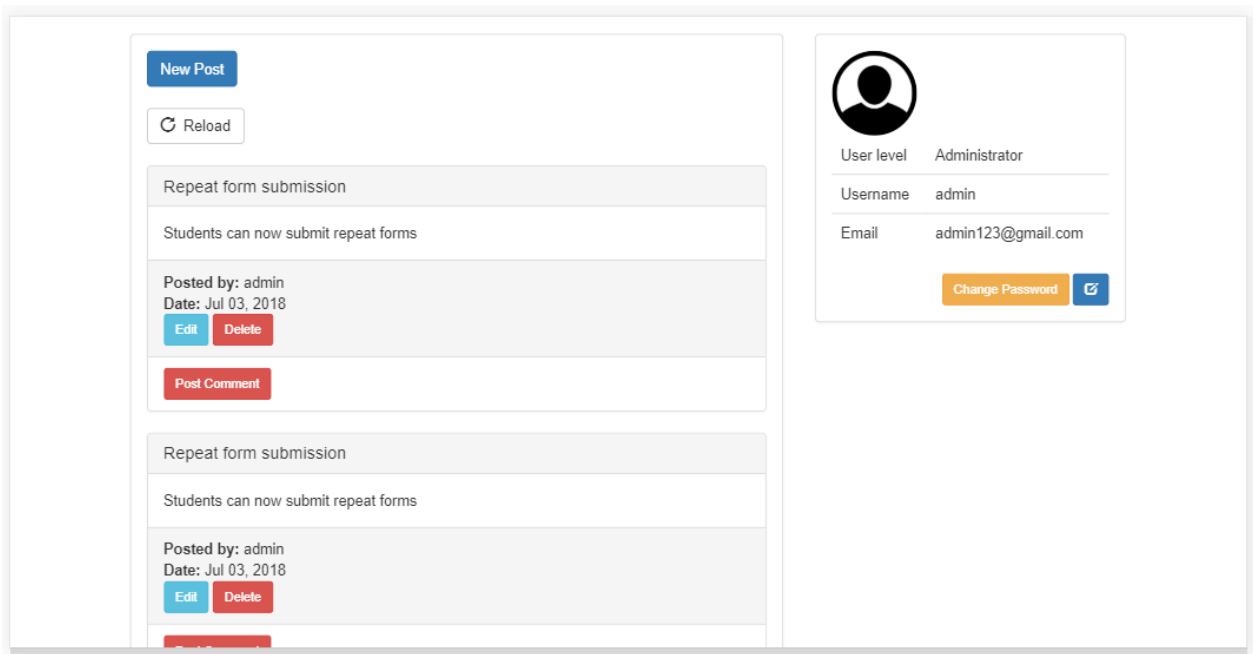
Delete
Go Back
Save Changes

© 2018, All rights reserved by University of Colombo School of Computing No: 35, Reid Avenue, Colombo 7, Sri Lanka.



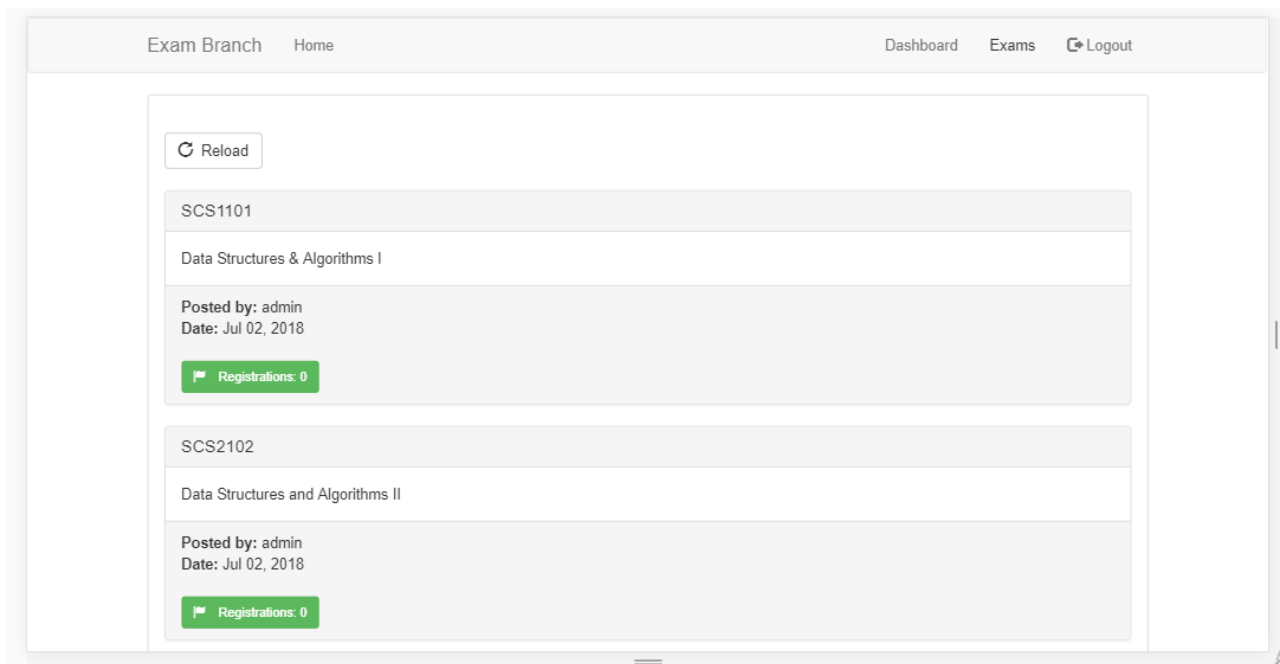
- Updates (add , edit, remove)

Like exams, admin can add, edit and remove updates.



Student:

- View exams

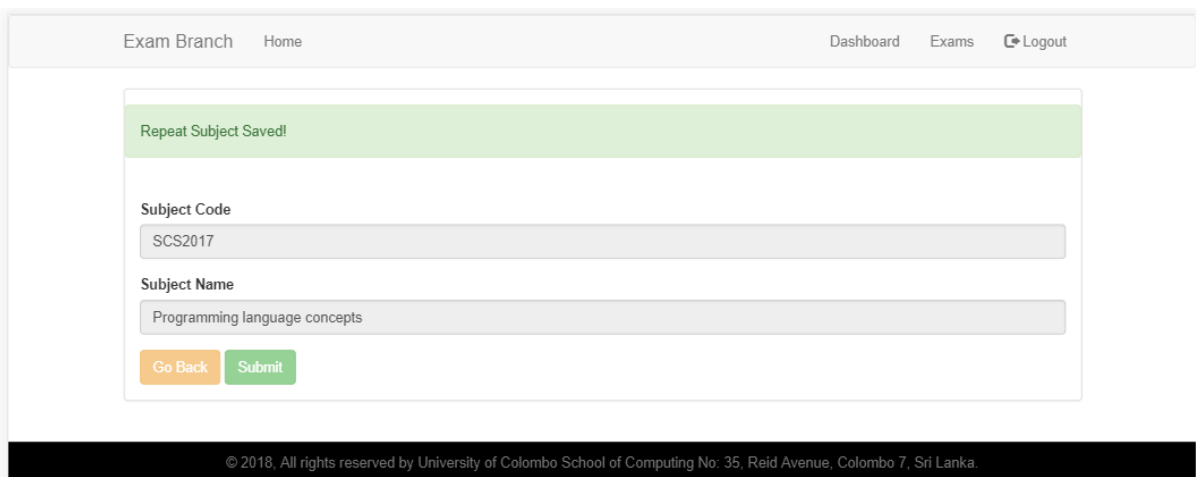


- Register for the exams

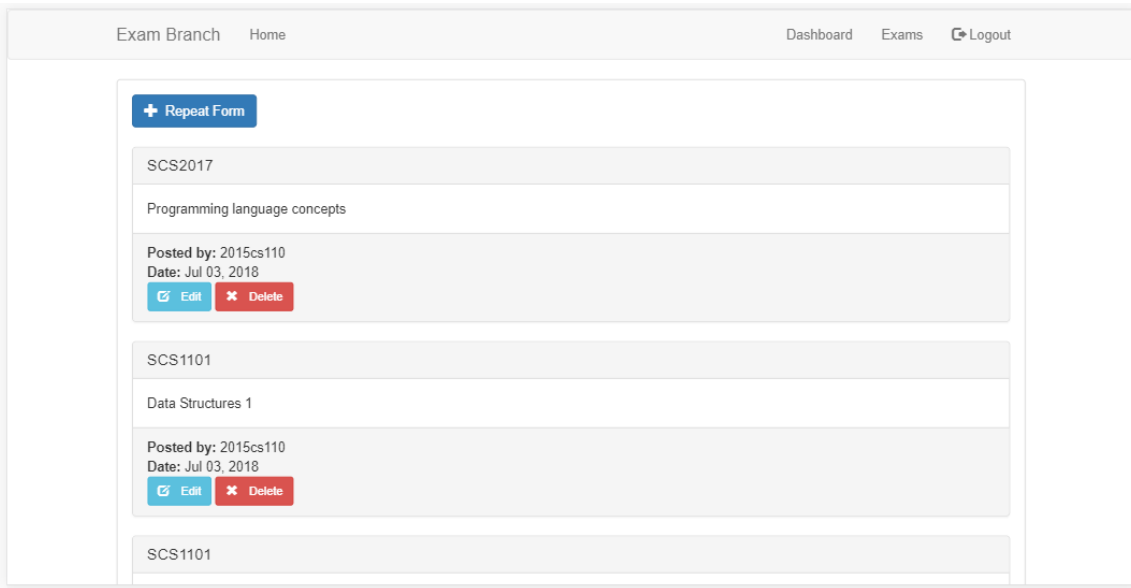
By clicking registration button student can register for a particular exam. And the admin can see the number of registrations and the students who have registered.

- Send repeat forms

By navigating to Repeat Form Submission -> Repeat Form , student can submit the repeat subjects he/she will willing to do.

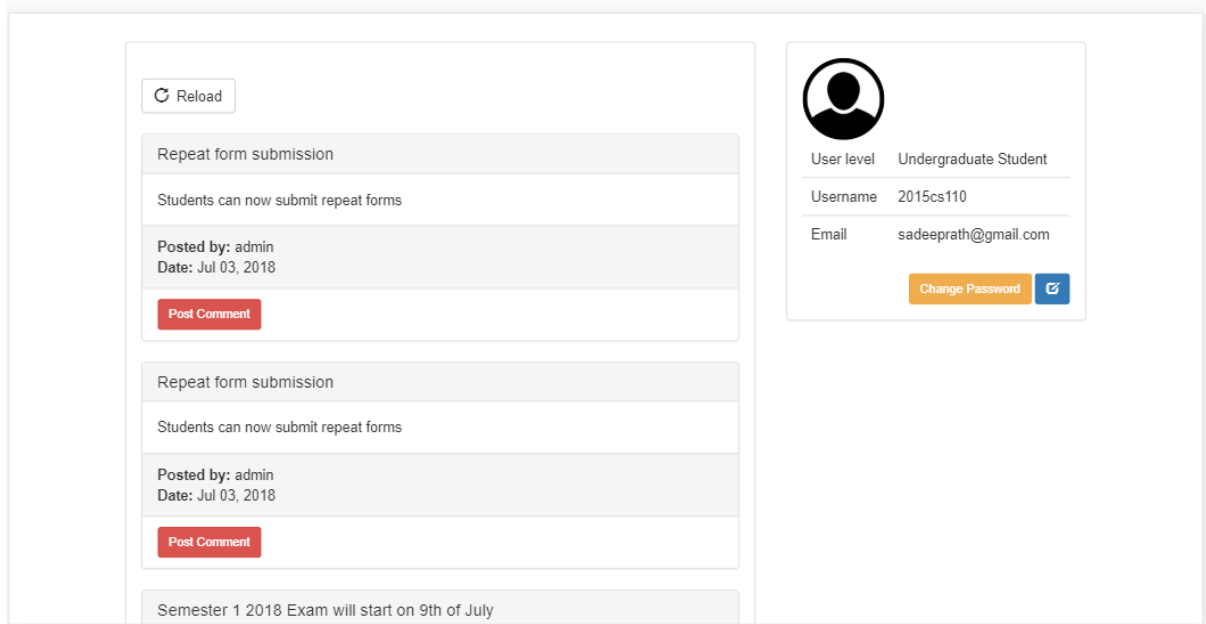


After submitting this for page will again redirect to the repeat\_form route which is the previous page. There you can see the newly added repeat subject. On this page you can see your repeat subject submissions.



- View updates

Student can view updates but cannot edit and delete them.



## 02) Testing Evidence:

Postman

File Edit View Help

New Import Runner +

My Workspace

Filter

History Collections

Clear all

Today

- GET http://localhost:8080/repeats/allRepeats
- GET http://localhost:8080/updates/allUpdates
- GET http://localhost:8080/exams/allExams
- POST http://localhost:8080/exams/newExam
- POST http://localhost:8080/exams/newExam
- POST http://localhost:8080/exams/newExam
- POST http://localhost:8080/authentication/login
- POST http://localhost:8080/authentication/login
- POST http://localhost:8080/authentication/register
- POST http://localhost:8080/authentication/register

GET http://localhost:8080/repeats/allRepeats

Authorization Headers Body Pre-request Script Tests

Key	Value
New key	Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON

```

1 {
2   "success": true,
3   "repeats": [
4     {
5       "id": "5b3b1f92296f4f36703a04ef",
6       "subject_code": "SCS2017",
7       "subject_name": "Programming language concepts",
8       "createdBy": "2015cs110",
9       "v": 0,
10      "createdAt": "2018-07-03T04:56:41.124Z"
11    },
12    {
13      "id": "5b3b0d88296f4f36703a04eb",
14      "subject_code": "SCS1101",
15      "subject_name": "Data Structures 1",
16      "createdBy": "2015cs110",
17      "v": 0,
18      "createdAt": "2018-07-03T04:56:41.124Z"
19    },
20    {
21      "id": "5b3b0d76296f4f36703a04ea",
22      "subject_code": "SCS1101",
  
```

Postman

File Edit View Help

New Import Runner +

My Workspace

Filter

History Collections

Clear all

Today

- GET http://localhost:8080/exams/allExams
- POST http://localhost:8080/exams/newExam
- POST http://localhost:8080/exams/newExam
- POST http://localhost:8080/exams/newExam
- POST http://localhost:8080/authentication/login
- POST http://localhost:8080/authentication/login
- POST http://localhost:8080/authentication/register
- POST http://localhost:8080/authentication/register
- POST http://localhost:8080/authentication/register
- POST http://localhost:8080/authentication/register

GET http://localhost:8080/exams/allExams

Authorization Headers Body Pre-request Script Tests

Key	Value
New key	Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON

```

1 {
2   "success": true,
3   "exams": [
4     {
5       "id": "5b39726a9a979b0df01c8eab",
6       "subject_code": "SCS1101",
7       "subject_name": "Data Structures & Algorithms I",
8       "createdBy": "admin",
9       "v": 0,
10      "registeredBy": [],
11      "registrations": 0,
12      "createdAt": "2018-07-02T00:16:39.926Z"
13    },
14    {
15      "id": "5b396d14bb96723c4cbfd5de",
16      "subject_code": "SCS2102",
17      "subject_name": "Data Structures and Algorithms II",
18      "createdBy": "admin",
19      "v": 0,
20      "registeredBy": [],
21      "registrations": 0,
22      "createdAt": "2018-07-02T00:07:48.700Z"
  
```



Postman

File Edit View Help

New Import Runner My Workspace

Filter

History Collections

Clear all

Today

- GET http://localhost:8080/updates/allUpdates
- GET http://localhost:8080/exams/allExams
- POST http://localhost:8080/exams/newExam
- POST http://localhost:8080/exams/newExam
- POST http://localhost:8080/exams/newExam
- POST http://localhost:8080/authentication/login
- POST http://localhost:8080/authentication/login
- POST http://localhost:8080/authentication/register
- POST http://localhost:8080/authentication/register
- POST http://localhost:8080/authentication/register

http://localhost:8080/updates/allUpdates

GET

Authorization Headers Body Pre-request Script Tests

Key Value

New key Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON

```

1 {
2   "success": true,
3   "updates": [
4     {
5       "id": "5b3b0b296f4f36703a04e7",
6       "title": "Repeat form submission",
7       "body": "Students can now submit repeat forms",
8       "createdBy": "admin",
9       "_v": 0,
10      "comments": [],
11      "createdAt": "2018-07-03T04:56:41.116Z"
12    },
13    {
14      "id": "5b3b0af9296f4f36703a04e6",
15      "title": "Repeat form submission",
16      "body": "Students can now submit repeat forms",
17      "createdBy": "admin",
18      "_v": 0,
19      "comments": [],
20      "createdAt": "2018-07-03T04:56:41.116Z"
21    }
22  ]
23 }

```

## Using selenium IDE

Selenium IDE - Exam Branch\*

Exam Branch\*

Tests +

Search tests...

http://localhost:4200

Command	Target	Value
4. click at	css=div.modal-footer > button[name=button]	34,18
5. mouse over	css=div.modal-footer > button[name=button]	

Command

Target

Value

Comment

Runs: 1 Failures: 0

Log Reference

'Edit exam' completed successfully

Running 'Delete exam'

- open on /exams... OK
- clickAt on css=app-root with value 66,226... OK
- clickAt on css=span.glyphicon.glyphicon-remove with value 9,6... OK
- clickAt on css=div.modal-footer > button[name=button] with value 34,18... OK
- mouseOver on css=div.modal-footer > button[name=button]... OK

'Delete exam' completed successfully

Selenium IDE - Exam Branch\*

Exam Branch\*

Tests +

Search tests...

http://localhost:4200

Command	Target	Value
6. type	name=subject_name	Database III
7. click at	name=save	50,18
8. mouse over	name=save	

Command

Target

Value

Comment

Runs: 1 Failures: 0

Log Reference

'Edit exam' completed successfully

- clickAt on css=a > button[name=button] with value 36,11... OK
- clickAt on name=subject\_code with value 118,21... OK
- type on name=subject\_code with value SCS4122... OK
- clickAt on name=subject\_name with value 108,14... OK
- type on name=subject\_name with value Database III... OK
- clickAt on name=save with value 50,18... OK
- mouseOver on name=save... OK

'Edit exam' completed successfully

Selenium IDE - Exam Branch\*

Exam Branch\*

Tests +

Search tests...

http://localhost:4200

	Command	Target	Value
New Exam*			
New Update	8. click at	xpath=//button[@name='button']][2]	54,17
Register	9. mouse over	xpath=//button[@name='button']][2]	

Command

Target

Value

Comment

Runs: 1 Failures: 0

Log Reference

3. mouseOver on name=button... OK

4. clickAt on name=subject\_code with value 69,13... OK

5. type on name=subject\_code with value SCS4123... OK

6. clickAt on name=subject\_name with value 78,11... OK

7. type on name=subject\_name with value Database III... OK

8. clickAt on xpath=//button[@name='button']][2] with value 54,17... OK

9. mouseOver on xpath=//button[@name='button']][2]... OK

**'New Exam' completed successfully**

Selenium IDE - Exam Branch

Exam Branch

Tests +

Search tests...

http://localhost:4200

	Command	Target	Value
New Update			
Register	8. click at	xpath=//button[@name='button']][2]	53,24
	9. mouse over	xpath=//button[@name='button']][2]	

Command

Target

Value

Comment

Runs: 1 Failures: 0

Log Reference

3. mouseOver on name=button... OK

4. clickAt on name=title with value 63,15... OK

5. type on name=title with value Repeat form submission... OK

6. clickAt on name=body with value 168,22... OK

7. type on name=body with value Students can now submit repeat forms ... OK

8. clickAt on xpath=//button[@name='button']][2] with value 53,24... OK

9. mouseOver on xpath=//button[@name='button']][2]... OK

**'New Update' completed successfully**

Selenium IDE - Exam Branch\*

Exam Branch\*

Tests +

Search tests...

http://localhost:4200

	Command	Target	Value
Register	1. open	/	
	2. click at	//a[contains(text(),'Register')]	31,22
	3. mouse over	//a[contains(text(),'Register')]	
	4. mouse out	css=li.active	

Command

Target

Value

Comment

Runs: 1 Failures: 0

Log Reference

4. mouseOut on css=li.active > a... OK

5. clickAt on name=username with value 226,20... OK

6. type on name=username with value 2014CS112... OK

7. type on name=email with value rangana@gmail.com... OK

8. type on name=password with value Rangana@123... OK

9. type on name=confirm with value Rangana@123... OK

10. clickAt on css=input.btn.btn-primary with value 41,11... OK

**'Register' completed successfully**

Selenium IDE - Exam Branch\*

Exam Branch\*

Tests +

Search tests...

http://localhost:4200

	Command	Target	Value
Delete exam	1. open	/updates	
Edit exam	2. click at	css=div.panel-body > button[name='button']	69,22
New Exam			
New Update			
Register			
Reload*			

Command

Target

Value

Comment

Runs: 1 Failures: 0

Log Reference

3. clickAt on css=span.glyphicon.glyphicon-remove with value 9,6... OK

4. clickAt on css=div.modal-footer > button[name='button'] with value 34,18... OK

5. mouseOver on css=div.modal-footer > button[name='button']... OK

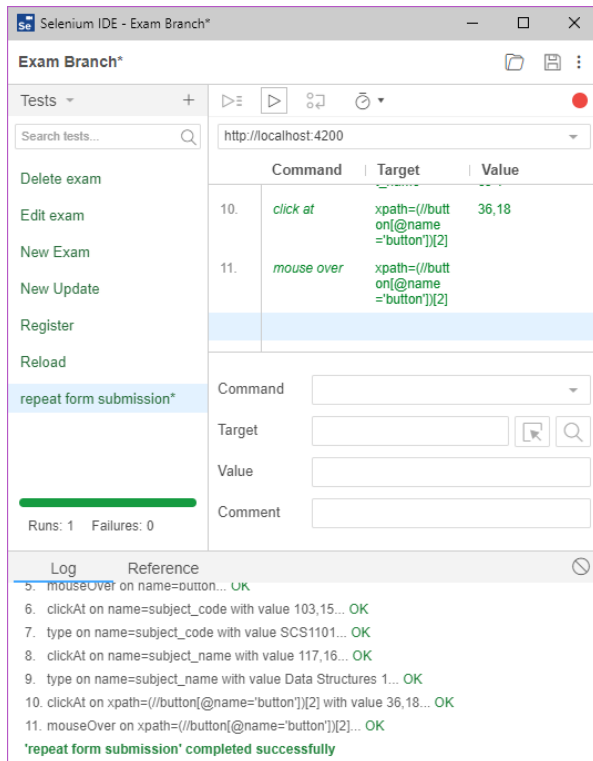
**'Delete exam' completed successfully**

Running 'Reload'

1. open on /updates... OK

2. clickAt on css=div.panel-body > button[name='button'] with value 69,22... OK

**'Reload' completed successfully**



### 03) Problems occurred:

- Registration & Login :

I had to decide between having two login pages and two registration pages for the administrator and the users. Because of the implementation difficulty and the design I thought having one interface for each is most suitable in this scenario.

- Repeat Form Submission

Original Repeat form having a place to attach the payment receipt so when it comes to the system we have to implement it as uploading a receipt.

- Exam Registration

Submitting forms is inefficient so I thought of adding buttons to register.

### 04) Assumptions:

- System has one administrator.
- Student have to register with the system to use it. When registering they have to provide the Registration no as the username and email is a requirement.
- Administrator can view student but cannot alter or delete.
- Administrator can view repeat forms but cannot alter or delete.