Course: COMP1640

Enterprise Web Software Development

Submitted to:
Dr. Ray Stoneham Name
Course leader
University of Greenwich, UK

Submitted by:
Banner ID: 000989717
Submission date: **13th** December 2017

Group name: “Group-7”

Group members:

SL.	Name	Role(s)
1	(Me)	Tester
2	Sazzadul Alam	Programmer
3	Tahera Mohammad Salim Sarker	System Analyst and Database Designer
4	Abdullah Alam Ibnu Raihan Mahmood Ratul	Web Designer

Application Live URL: (http://www.ewsd.algologixbd.com/).
Screen cast URL: (https://www.youtube.com/?)
Group Repository URL: (https://www.dropbox.com/? **)**

Credentials:

SL.	Username	Password	Role
1	ratul	1234	Student
2	shazzad	1234	Student
3	sarwer	1234	EC manager
4	siplu	1234	EC Coordinator
5	mustafiz	1234	EC Coordinator
6	dev	1234	Admin

Table of Contents

Table of Figures:

1. Evaluation of product and process:

Our task was to develop enterprise web software adopting with agile scrum. Since it's a group work so, we played different role in this course work alongside agile scrum methodology. Now I am here to evaluate the product and its process.

System requirements met:

We have been asked build role based web software for a large university student so that they submit claim of Extenuating Circumstances (EC). The system should meet some requirements or criteria; we need to reflect on those criteria during the development of the product:

- EC manager will be able to view the entire claim stored in the system, but he won't be able to process any submitted claim. It should be compatible for all kinds of devices and secure as well as.
- Students can submit EC claims based on different assessment in one designated facility. After that they will be notified shortly via email that their submitted EC. Can view claims as well as.
- EC Coordinator allows claim to be approved according to each faculty, can oversee the all claim and can process. After submitting a EC claim, EC coordinator will be notified via mail. EC Coordinator can't process any claim without any submission, validity of claim will expiry maximum 15 days.
- An admin can do anything within the system, to be honest, admin have the power of manage the whole EC claim software like role base

management, setting up the closure date and final date of EC and can add different assessment to list.

- Statistical analysis of the stored data like graphical presentation of data it can be a chat or something else. It will produce number of reports depending on different type of needs.

(To see all the functionaries and features please go to the my Appendix part OR visit the group repository)

Agile Process:

We have worked together by practicing and adopting the agile scrum methodology. We have recorded the meeting minutes when we did meeting together for our own good. Our group was combining four (4) person based on roles such as information architect, web designer, database designer and tester basically.

We always tried to adopt the agile scrum philosophy during the development of the entire system; the most important things observed from the very beginning from our assignment, these are:

- **Product backlog**: A list of requirement based on the priority was build for making the product more valuable
- **Sprint:** A little piece of work from product backlog was developed.
- **Scrum meeting:** We conducted many meetings up to 1 hour duration, different type of problems are discussed there.

Strength of the system:

- The system has secured role based functionalities so everything is under the control of admin that ensures whole security of the system.
- One of the most important think is responsiveness of the system it can be compatible with all other device and beyond the platform people use.
- It can produce different reports based on different assessment and it has dashboard that displays the entire claim from role's perspective.
- Manageability is the great feature of this system, so that admin can manage whole system and take any changes needed.
- System can take a huge load of pressure when student are intended to claim submission from different place and from large number of student.
- It has great usability and UI design help people to use the system easily. And lots feature to be used.

Weakness of the system:

- There are some minor defect related to data with some browsers
- Need to be hosted in a good qualified server
- There is a compromise with UI design, some parts are designed badly comparatively

Further improvement:

- A Live chat feature would be a good part to be developed in future
- Application performance can be increased in future
- Security should be more strict in further development
- Making more responsive the system
- Making system's UI more useful.

2. Evaluation of team:

Weighting factor for each student (Scale 0 to 10)

Commitment	**Weight**
Fully committed	10
Committed	8
Contributed substantially	6
Contributed partially	4
Minimal contribution	2
No contribution	0

A scoring model of entire team including myself (Scale 0 to 10):

Group members	**Weight score**
Me	8
Sazzadul Alam	8
Tahera Mohammad Salim Sarker	6
Abdullah Alam Ibnu Raihan Mahmood Ratul	8

Collaboration:

Collaboration of the group members is one of the main virtues that made us more helpful to carry out the project along with the agile scrum practices; we have worked collaboratively with other in needs.

Participation:

Without some moment everything was fair and lovely. Everyone has contributed their own efforts best they could in from their own perspectives. They have given their individual effort based on their

Attitude:

Other group member's attitude was awesome from everyone's perspective, they always showed a positive attitude to me and took every single feedback positively that I gave.

Independence:

We worked independently without any barrier of other team members. They always encourage each other to go ahead with the assignment. So that we can do our specific work timely.

Communication:

Communication was our main key aspects to move forward with the development of the software. Everyone has expressed their thinking each other like what they, what want to do, what can be done, how to do any why etc. So that we can keep in touch to development the system within the limited.

Responsiveness

For any reason our group member was very responsive, they reacted positively when asked something about the project. There was no such verbal and non verbal consistency in other team members.

3. Self Evaluation:

This is my honest description of my own contribution to our group to be developed the software. This part reflects based on my performance that performed during the whole process of the coursework:

Effective use of time:

It's a great opportunities to me to learn something like this. Now, I can do something like this timely and with a limited time.

Leadership:

As a tester I very often led the group to do group activities on time, to make the system more dynamic and business values. If there is problem with something I tried to solve with other team members to get rid of something like this.

Contribution to team effort:

I have played a role of system tester, as a tester I have tested major parts of the web software so that as a member of team I can help other team members. My approach was always contribute to team so that I can do better that any other member as my perspective.

Completion of assigned or agreed-upon responsibilities:

I always tried to do my responsibilities that are assigned to me and I have completed majority percent of the own responsibilities time to time. I have tried to complete successfully my responsibilities on my defined duties.

Completion of assigned or agree-upon tasks on time:

This is one of plus point for me to complete every task on time and make the team more dynamic to be developed the system. Sometime, I have not done some tasks for inconvenience due to force of nature.

Participation in team meetings:

I was always present and punctual in team meetings. Any problems that are facing out team, sometimes, I have suggested the way of solution to all the group members.

Quality of written contribution to team effort:

I have written the report and my assigned writing part very precisely as I can do my best. Every section in individual part and group part I written, quality was good enough. Appropriate writing is written in appropriate place where it carries a value.

Lesson learnt:

After completing the group coursework, I have leant how to adjust, communicate, react and collaborate with different types of people in information technology sector, especially in web development sector alongside the agile scrum process and methodology.

4. Appendices:

1. APPENDIX A: Test Plan and Test Log

Test plan is the documentation of planning of what activities need to be tested including scope and approach within specific resource and schedules. It defines different area of intended testing of the web application, such as: items that need to be tested, functions or features, tasks will be tested, the person who will perform test tasks, setup environment of the testing, techniques related to test design and so on. (softwaretestingfundamentals.com, n.d.)

Testing for web apps:

In the web applications, testing is a key and important part of the project. Technically, testing is carried out late time comparatively in the whole project processes and conducted within a small time. So, it is suggested to conduct the testing part so that tasks ensure high quality of the product. It ensures the correctness of the systems that intended to be, like other conventional software testing. (Pressman, 2010)

Objectives:

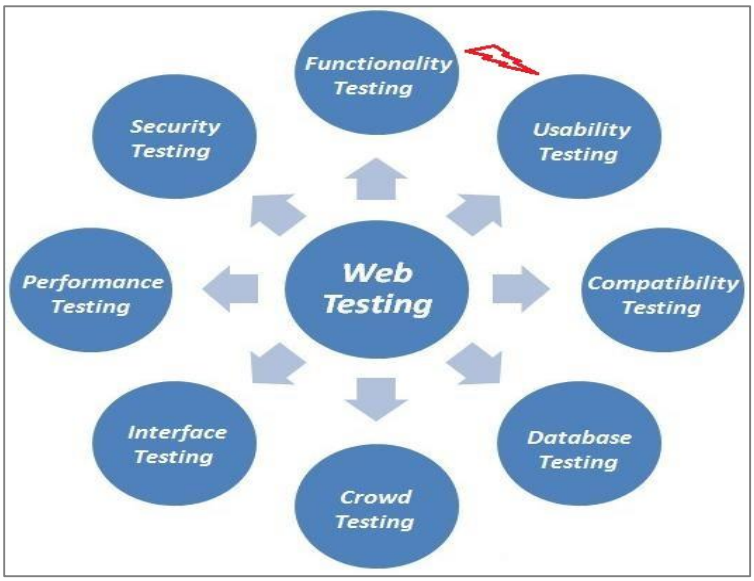
There are some objectives of testing thee web application, the goals and purposes of the behind the testing. Although, it not possible to make sure that the web

application has to be perfect but it's a testing of whether the application met the requirements or not including:

- To check whether the intended functionalities of product met or not, it could be functional and non functional.
- To check the correctness of the functional errors that about to be encountered.
- If users can use the web app easily in terms of usability; how usable the app is.
- Checking the application interfaces that are been designed and developed.
- Any problems or issues of database that are related to operations of program.
- Whether the application compatible or not with deferent platform.
- Overall performance of the system using different way.
- To ensure security of the system as client indented and so on. (Kota, 2016)

Test scope and Testing strategies:

There are many testing criteria and some strategies to carry out the testing process of this web product. Technically, these are the core portion or parts of a web application that need to be tested, these are:



(SoftwareTestingClass.com, 2012)

Figure 1 Web applications testing cycle

A brief description of what testing need to be carried out:

1. **Functional testing**:

   Testing of the functional component of the web application we developed. That included functional and some non functional testing. Technically, it is conducted to make sure that the application satisfied the client expectation. This includes some other testing procedures like different links, web forms, cookies test etc. (SoftwareTestingClass.com, 2012)

   - **User login**
   - Going to the proper page based on the user role

- If error occurs show message
- Checking of allowance to change their password
- **View all the claims**
- EC manager’s page opens
- Shows all the claim made
- Doesn’t let any claim to be processed
- **Manage claims**
- Claim will be approved according to faculty
- Shows all the claims and allows them to be processed
- Receive mail and notification of claim and the time remaining to process
- Doesn’t let claims to be processes without any evidence
- Cannot be able to enter after any claim after 15 days
- **Claim submission**
- Can submit claims for different assessment for one’s designated faculty
- Can upload evidence of claim
- Receive mail about claim progress
- Can view all of their claim details
- **System management**
- Can manage whole role base management of the user
- Set closure and final date for EC claim
- Can add new assessment to the assessment list.

2. **Testing of Usability**:

   This is one of the most important parts of a web app; a tester will carry out the testing of usability of the app. It requires the

   - Navigation
   - Content testing of the web application, what are easy enough to use

3. **Compatibility testing**:

   Checking the compatibility of the webapp, after all it's a non functional testing. It ensures how compatible the web application is to the users, in terms of working environments, OS, browser, hardware configurations, internet speed etc. It involves:

   - Browser compatibility,
   - Different OS,
   - Mobile platform and
   - Printing of web page. (SoftwareTestingClass.com, 2012)

4. **Testing of interfaces**:

   Different sections of interfaces of the application are covered such as: web, application and database server are tested from different point of views. It ensures the consistency of these three server communications and their behaviors. (SoftwareTestingClass.com, 2012)

5. **Database testing**:

   Purpose is to ensure reliability of data related the database, make sure that operational is secure and the way of dealing with the web application. How the application handles the database also is seen in terms of data manipulation.

   - Error of query
   - Data integrity with C~~R~~UD function
   - Checking the time of to execute queries and
   - Any complex query

6. **Performance testing**:

   It is about the performance of the web application, it deals with the performance during different circumstances like how much load it can take in a specific time. These are carried out by:

   - Web stress and
   - Web load testing of the web application.

7. **Security testing**:

   It ensures the security of the web application, as we know as enterprise web software security is the main issue and challenge of this application. How the application handle the critical situation in terms of application's overall security. Different testing criteria need to be tested.

   - Force of direct URL.
   - Try to upload malware
   - Sql injection

- Login authentication
- Brute force attack

8. **Crowd testing**:

   This testing is conducted by tester in associated with a group of non technical person. This ensues how the group tell about the product and their feedback about your software. Also ensure the performance, usability of the system from their perspective if they feel good.

Test Logs and Execute:

Here are the test plans along with execution to see how they behave and show result against their expectations, with each test case. After that we can produce “Test Plan” based on these activities:

Test No.	01
Test type	Functionality testing
Test name	User login
Expected result	Login to proper page based on role beside dash board
How to test	Input username and password of all roles
Actual result	Redirected to the role base page with a dashboard

- User login page:

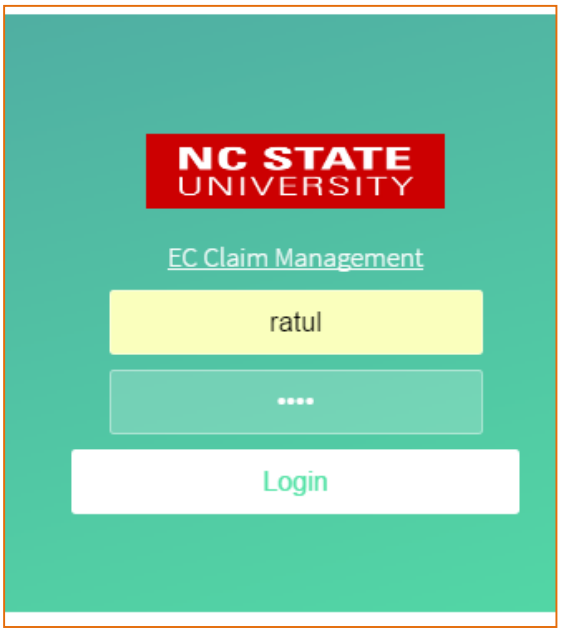


Figure 2 Username and password entered

- After login you will be transferred to intended page based on your role alongside dashboard:

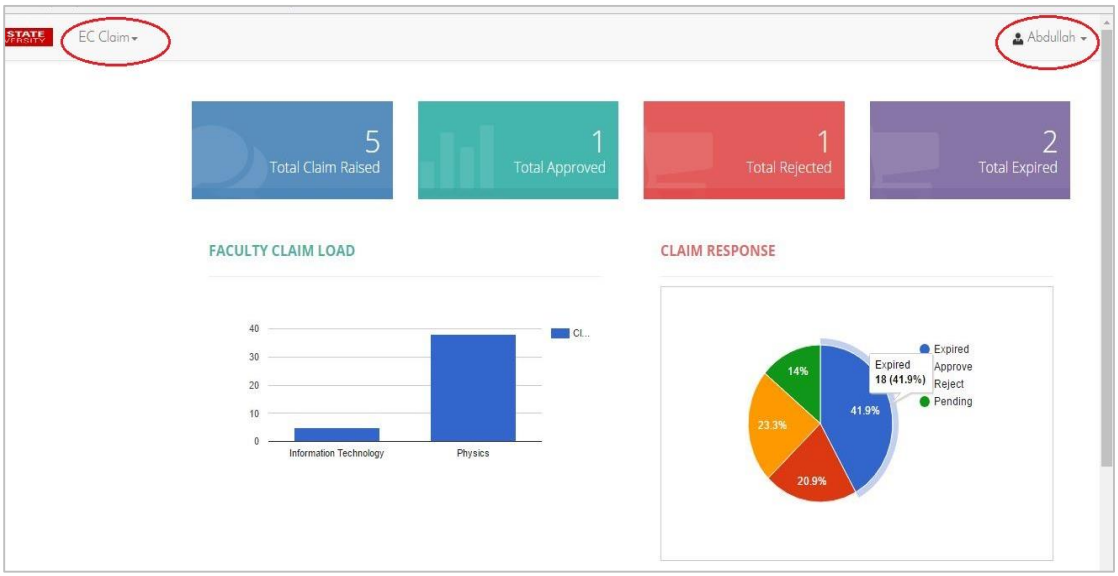


Figure 3 Login successfully to the student page with dashboard

Test No.	02
Test type	Functionality testing
Test name	Submit claim and view own claim
Expected result	Students will be able to submit claim
How to test	By selecting assessment, reason, remarks and evidences
Actual result	Claim submitted successfully, shown claim

- Here is the clam submission part from student perspective:

Figure 4 Claim submission option selected from the menu

- Figure shows claim submission page and information are entered without evidences:

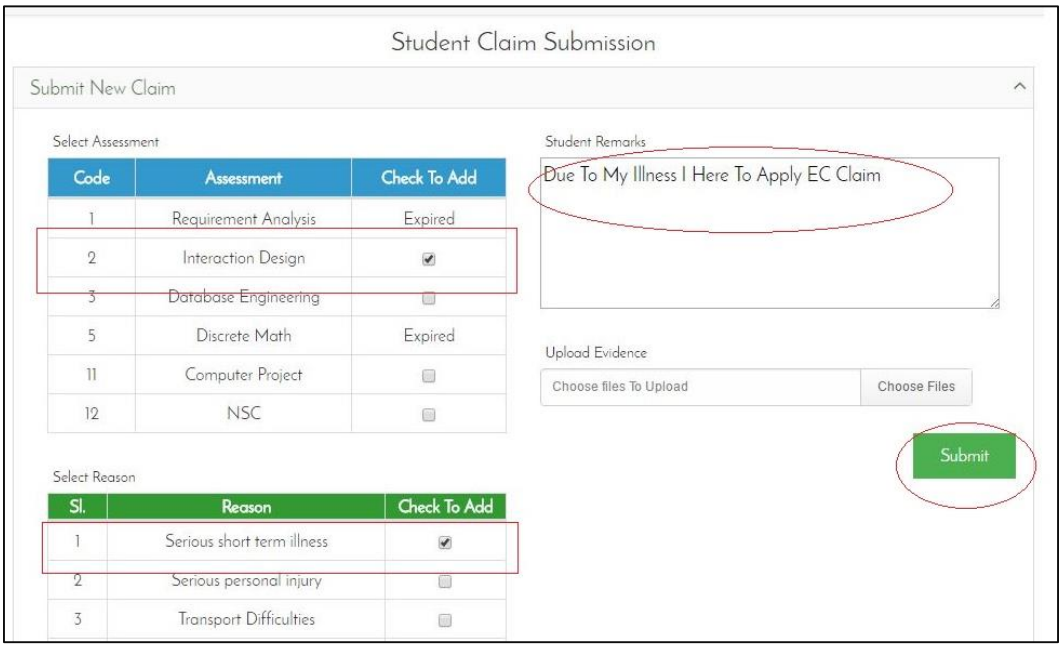


Figure 5 submitting a new claim

- This part show that the situation after pressing the submit button:

Figure 6 Claim submitted successfully

Test No.	03
Test type	Functionality testing
Test name	Re-upload evidences
Expected result	Students will be able to upload evidences
How to test	Open the submitted claim and re-upload evidences
Actual result	Re-upload evidence successfully

- Now I will upload the evidence (re-upload) from submitted claim:

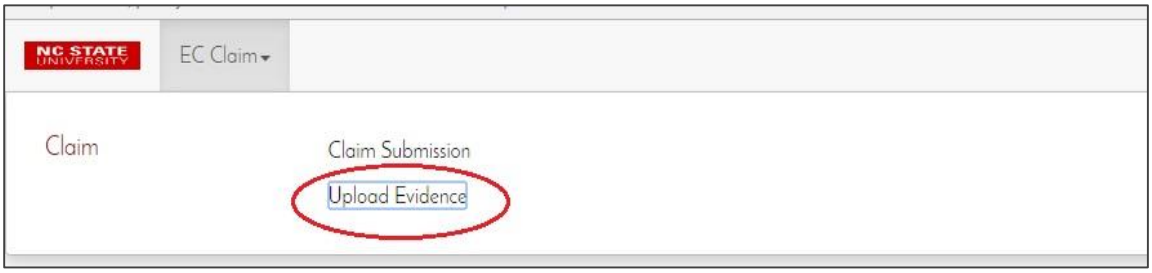


Figure 7 Uploading evidence option selected

- This is the figure of how to view the claim that I submitted before:

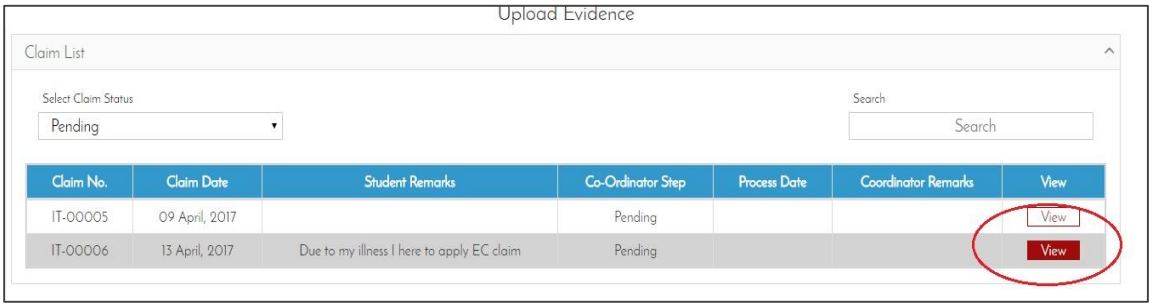


Figure 8 View the submitted claim

- This portion show that Re uploading evidence:

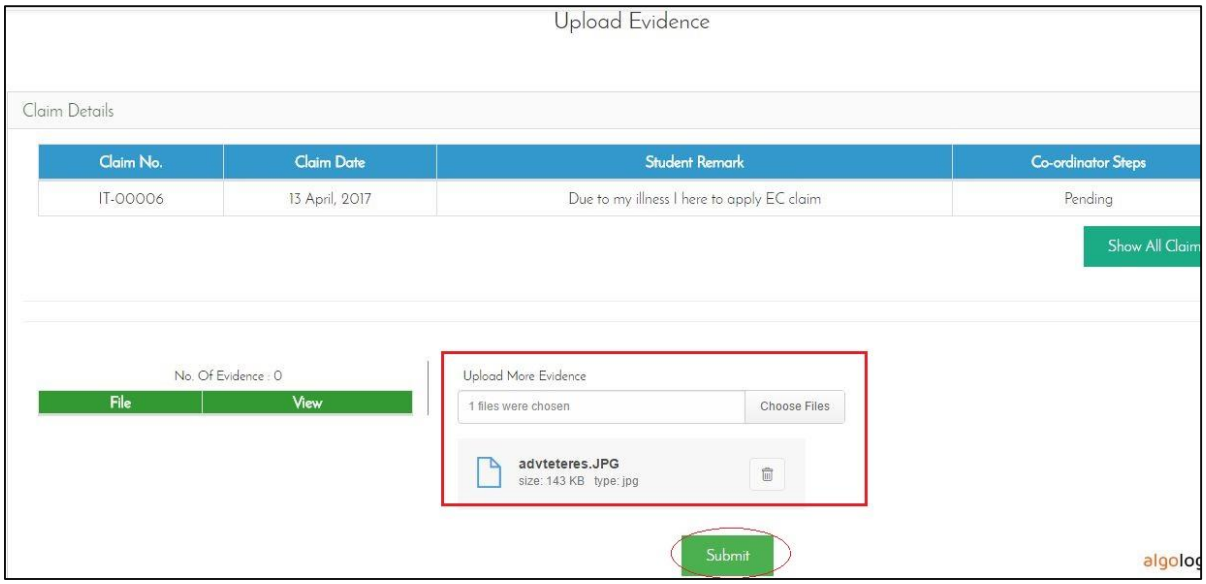


Figure 9 An evidence file were selected and submitted

- This section shows that the evidence upload confirmation:

Figure 10 Evidence uploaded successfully

Test No.	04
Test type	Functionality testing
Test name	Change password
Expected result	System let change the password
How to test	By putting another new passwords along with old password
Actual result	Password changed successfully

- It's time to see that if the password is change or not:

Figure 11 shows the option to change password

- Here is how the "change password" function works:

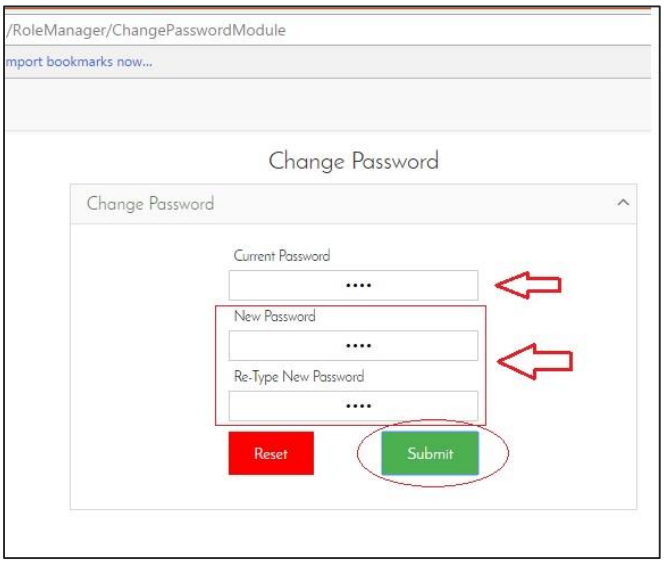


Figure 12 A new password was entered and submitted

- The confirmation that the password has been changed:

Figure 13 Password changed successfully

Test No.	05
Test type	Functionality testing
Test name	View claims (EC manager)
Expected result	EC manager page opens and can see all the claims made, but can't proceed
How to test	Clicking view claim option
Actual result	EC manager can view claim page and all the claims

- This section display the view of claims as EC Manager submitted by students but he can't proceed the claims shown below step by step:

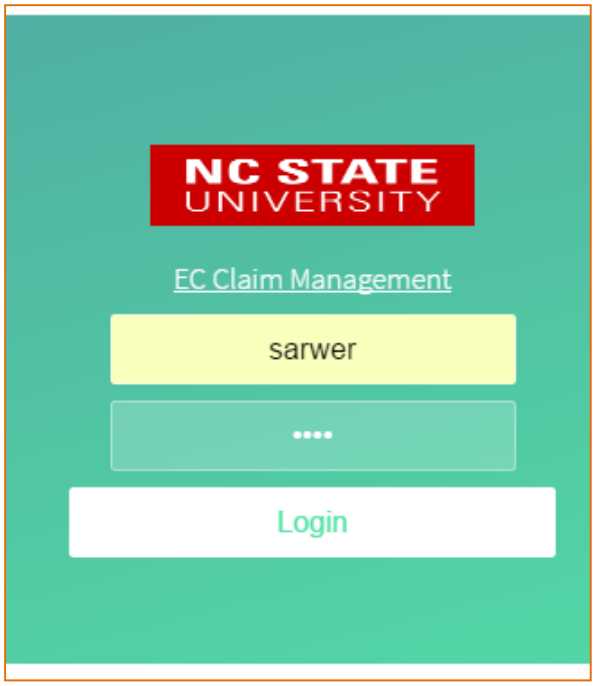


Figure 14 username "sarwer" as EC manager

- The page and dashboard of EC manager:

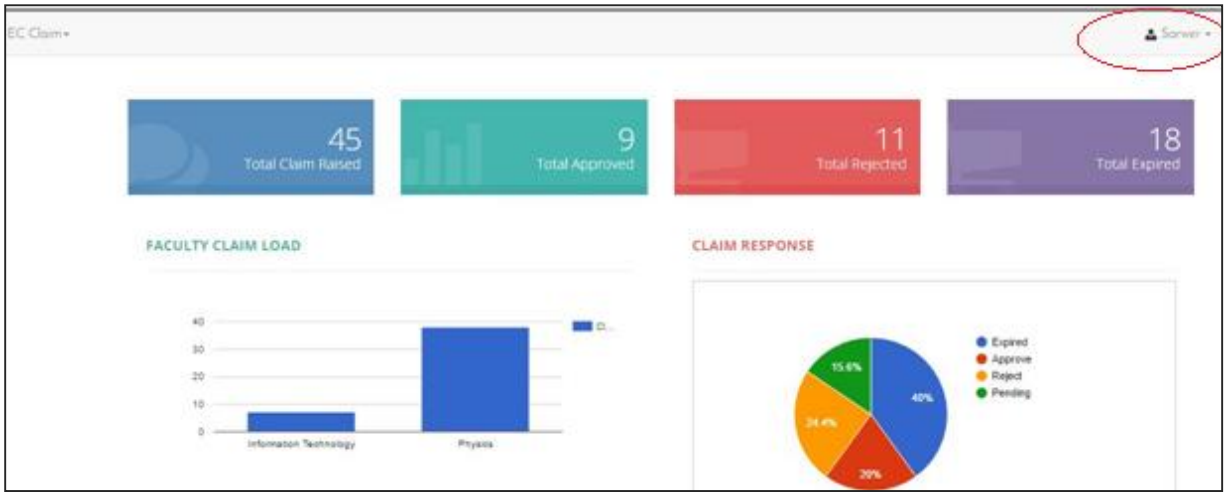


Figure 15 EC manager can see the summary in dashboard and can view claims

- Selecting nav bar in order to view claim:

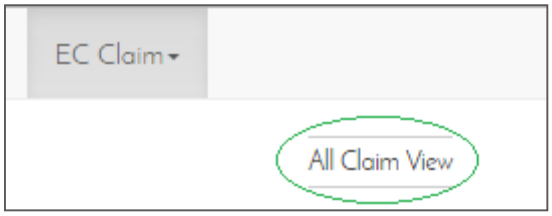


Figure 16 Select sub menu for display claim

- Here are the claims that arisen can view from different option and by different filter or can download as excel file:

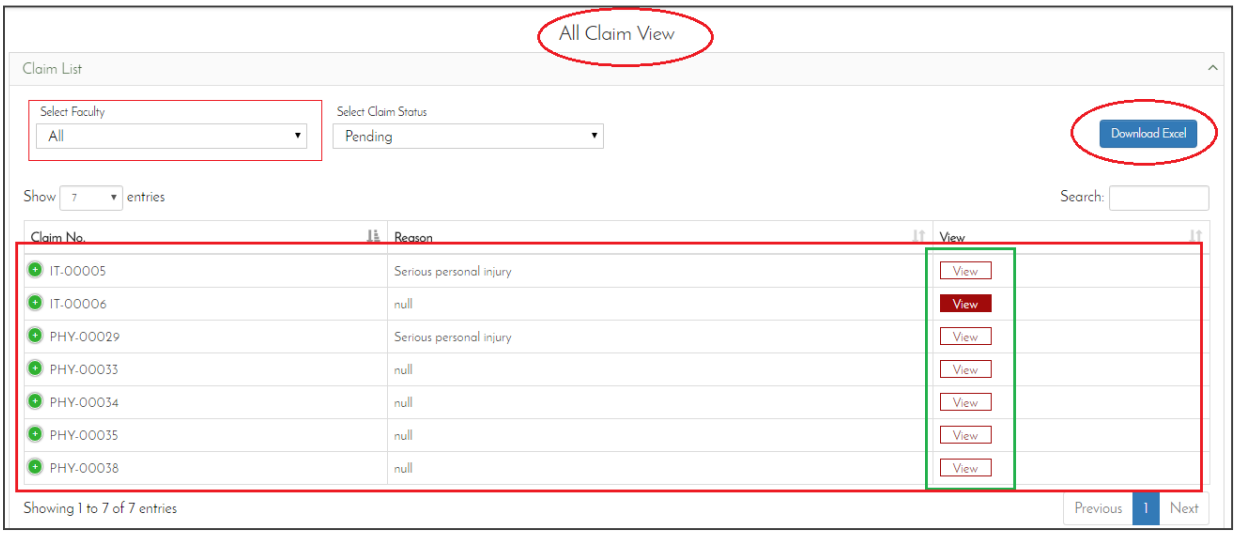


Figure 17 EC manager can oversee all claims

Figure 18 Can oversee details of any particular claim information

Test No.	06
Test type	Functionality testing
Test name	Manage claim by EC coordinator
Expected result	Clams will be shown to EC coordinator
How to test	By clicking Claim Process sub menu
Actual result	EC coordinator can see the claims

- EC coordinator can oversee the claim to be processed:

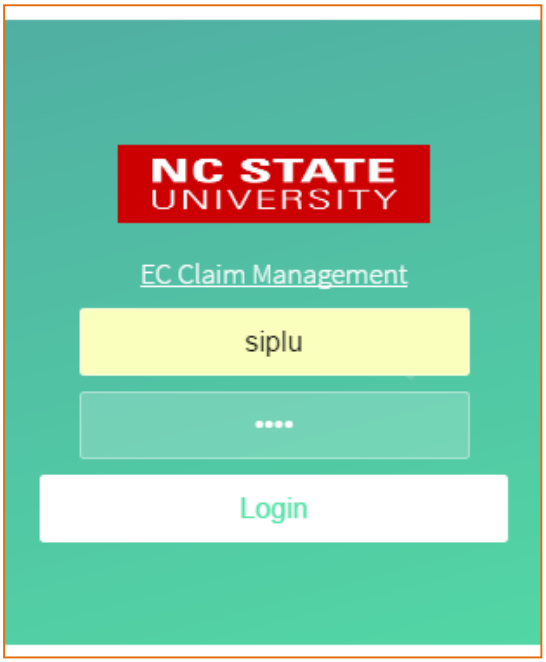


Figure 19 username "siplu" as EC Coordinator login

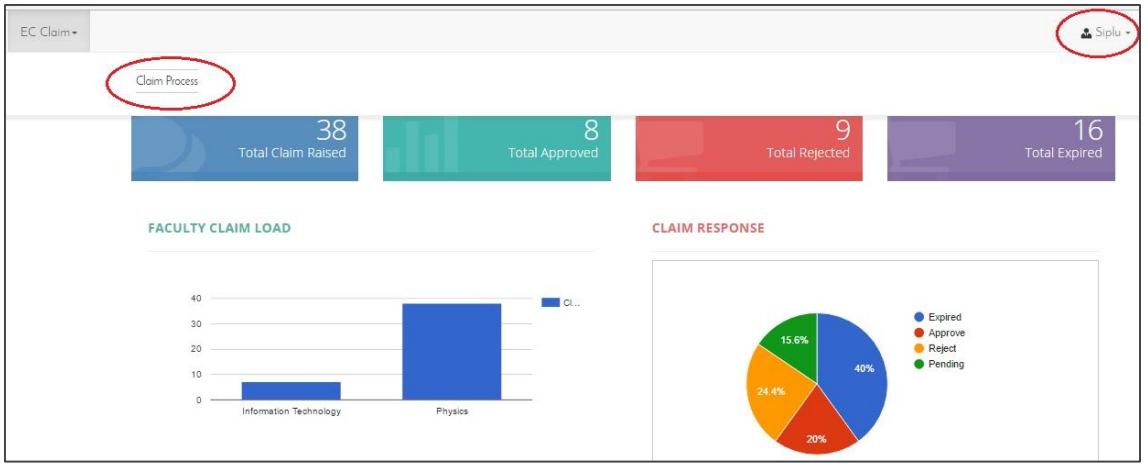


Figure 20 this figure show EC Coordinator homepage and dashboard

Figure 21 EC Coordinator can view all the claims by filtering.

Test No.	07
Test type	Functionality testing
Test name	Manage claim by EC coordinator
Expected result	EC coordinator will be able to process claim
How to test	Submitting approved/reject claim option
Actual result	EC coordinator processed claim successfully

- EC coordinator will be able to process claim:

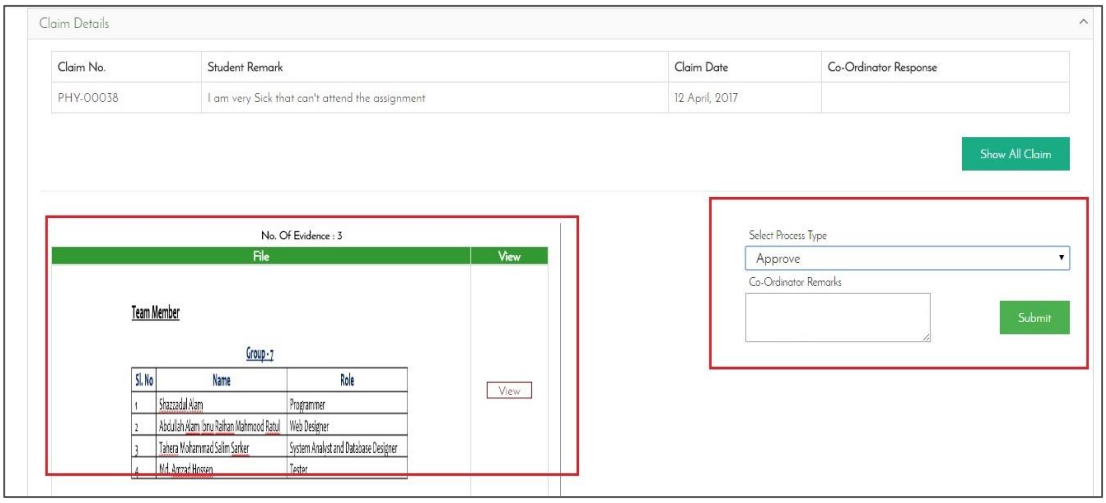


Figure 22 EC Coordinator processes claim

Figure 23 Message shows Claim processed successfully

Figure 24 Proof of Approval

Test No.	08
Test type	Functionality testing
Test name	Receive mail notification
Expected result	EC coordinator will receive an email notification
How to test	By checking email after EC coordinator processed claim
Actual result	EC coordinator have received email of claim processed

- This part show mail from the EC Coordinators:

Figure 25 Email notification from student of claim

Test No.	09
Test type	Functionality testing
Test name	Checking of process without evidence
Expected result	EC coordinator can't process any clam if there is no evidence
How to test	Try to process a claim without evidence
Actual result	Doesn't let claim to be processed without evidence

- System don't alow any claim to be processed withiout evidences:

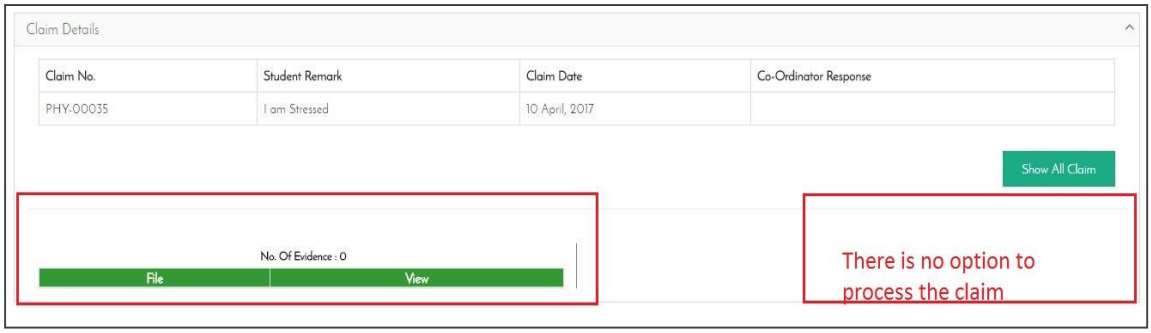


Figure 26 There is no evidence this is why can't process

Test No.	10
Test type	Functionality testing
Test name	Claim expiry date
Expected result	EC coordinator unable to enter claim after 14 days
How to test	Trying to process claim that expire 14 days
Actual result	EC coordinator can't be able to process after 14 days.

- After expiry date (14 days) claim will be legacy:

Test No.	11
Test type	Functionality testing
Test name	System management by admin
Expected result	Admin will be able to manage
How to test	By login to the Admin section
Actual result	Admin able to manage the whole system

- This is how the system admin (here, developer) can manage all funtiona and process of the system, in terms of claim facilities:

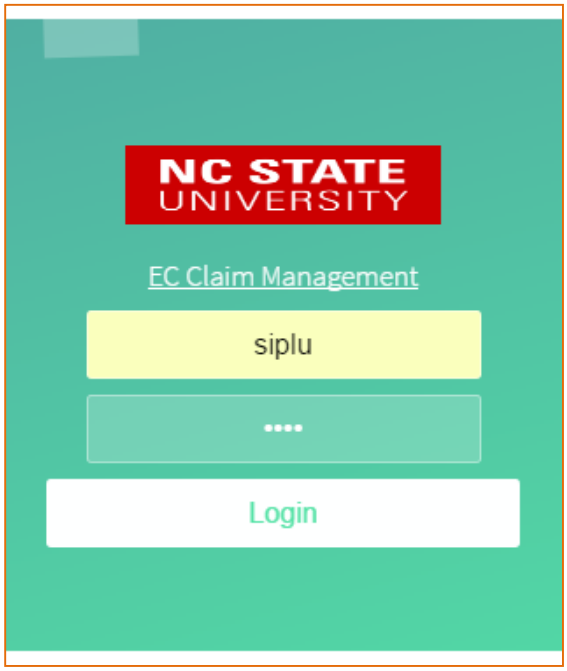


Figure 27 Here is the admin username "dev" login

- Here are the Admin facilities with different navigation and menus:

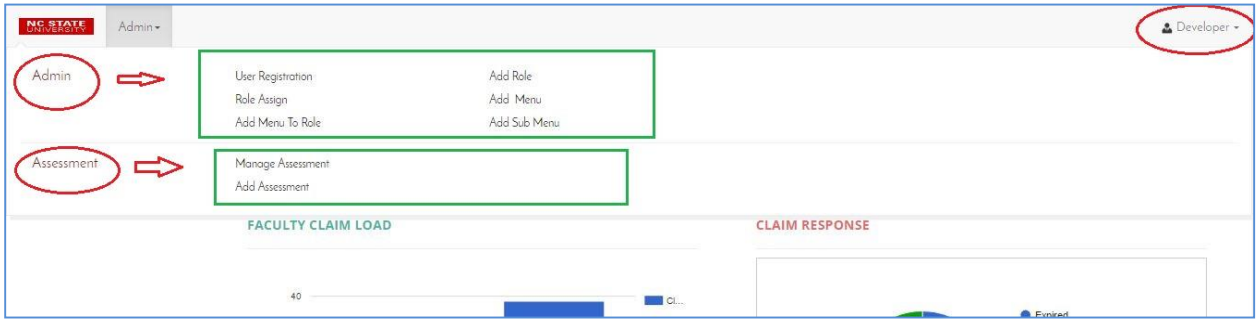


Figure 28 Figure show admin panel facilities

- ADMIN can user registration:

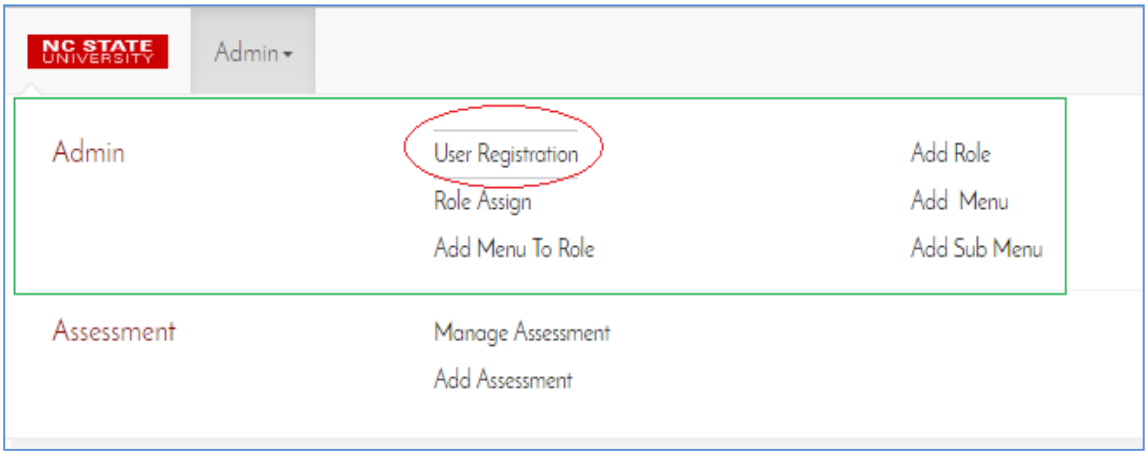


Figure 29 Admin menu User reg. sub menus

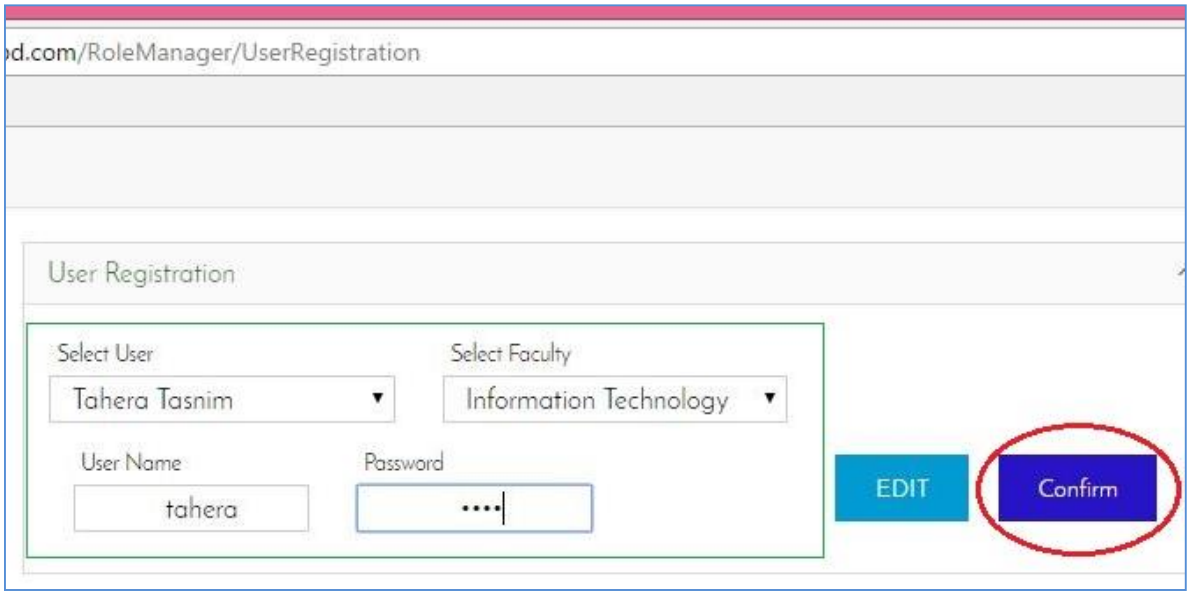


Figure 30 User reg. page and confirmed a user

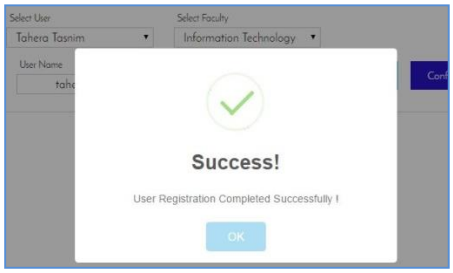


Figure 31 User registration completed successfully

- One can update information about Role/Users that are already assigned:

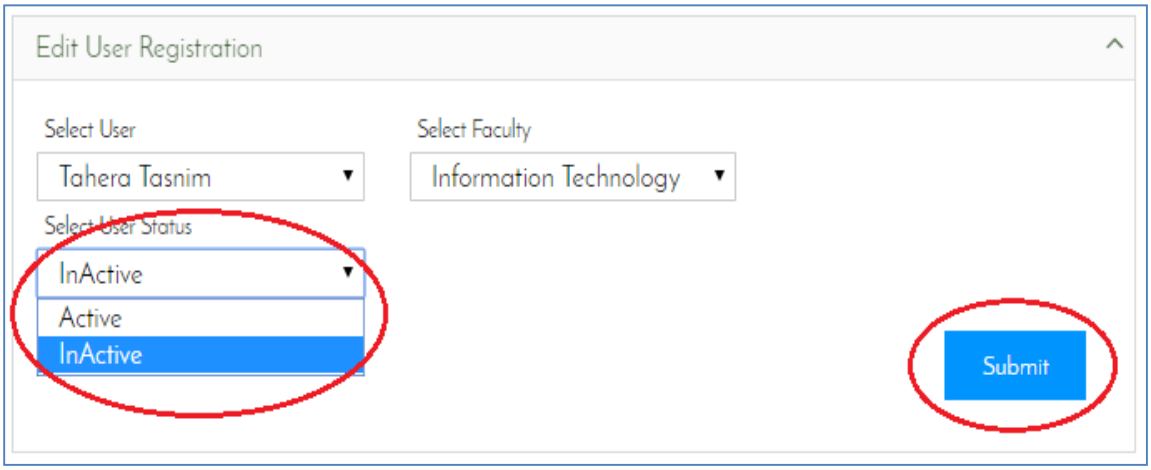


Figure 32 admin can inactive and active a role from the system

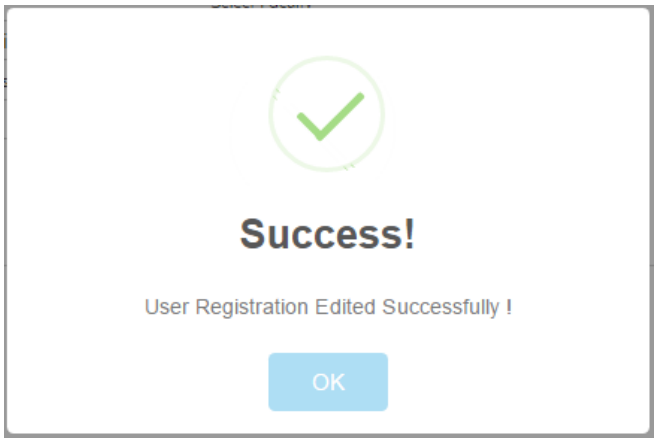


Figure 33 User reg. updated successfully

- Admin can assign a role depend on the user reg.:

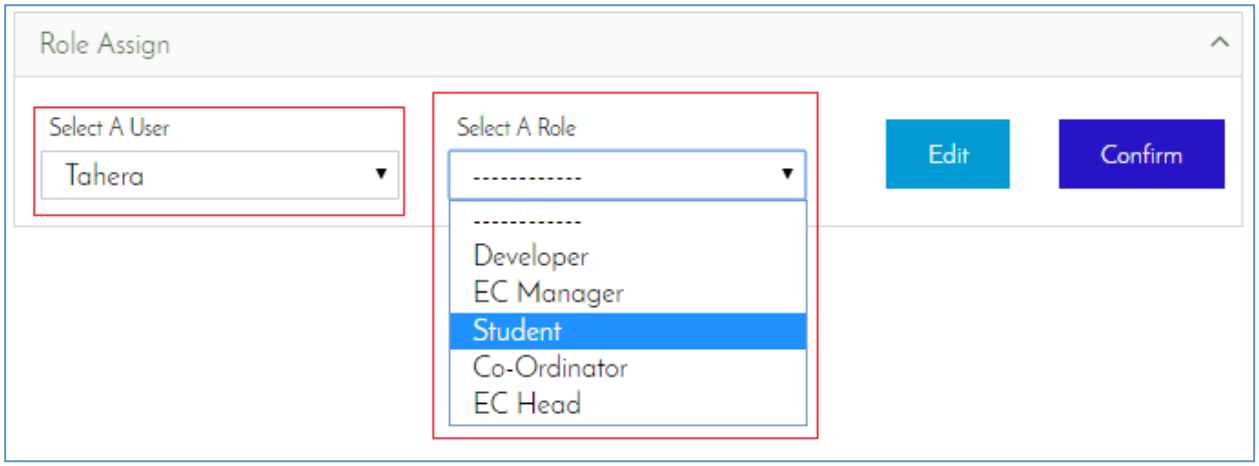


Figure 34 Option for assigning a role to a registered

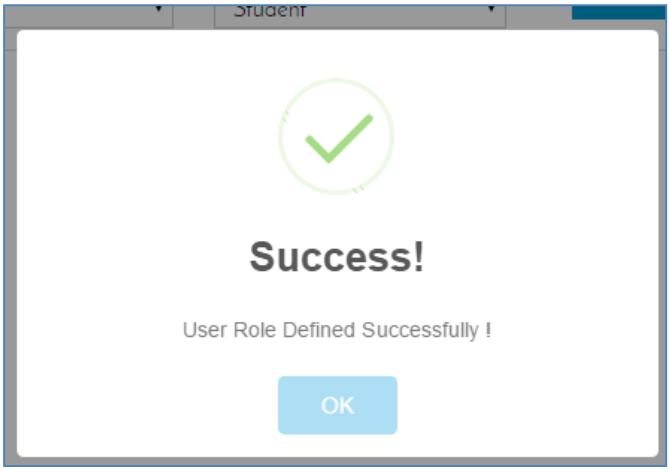


Figure 35 User role defined successfully

- Admin Can add function or Menus of different roles:

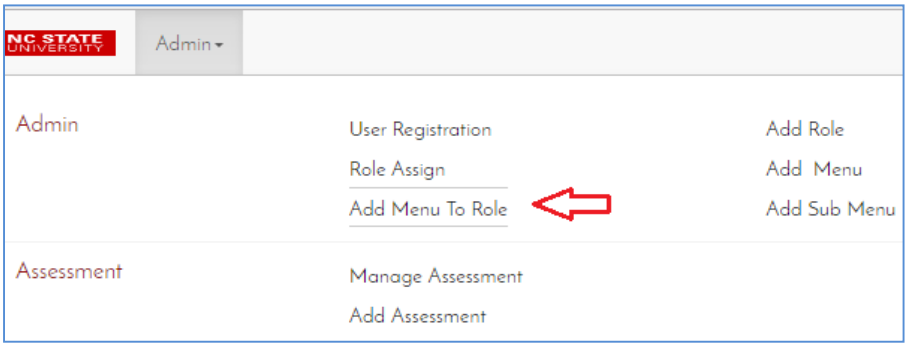


Figure 36 Figure shows how to add menu or sub menu to assigned role

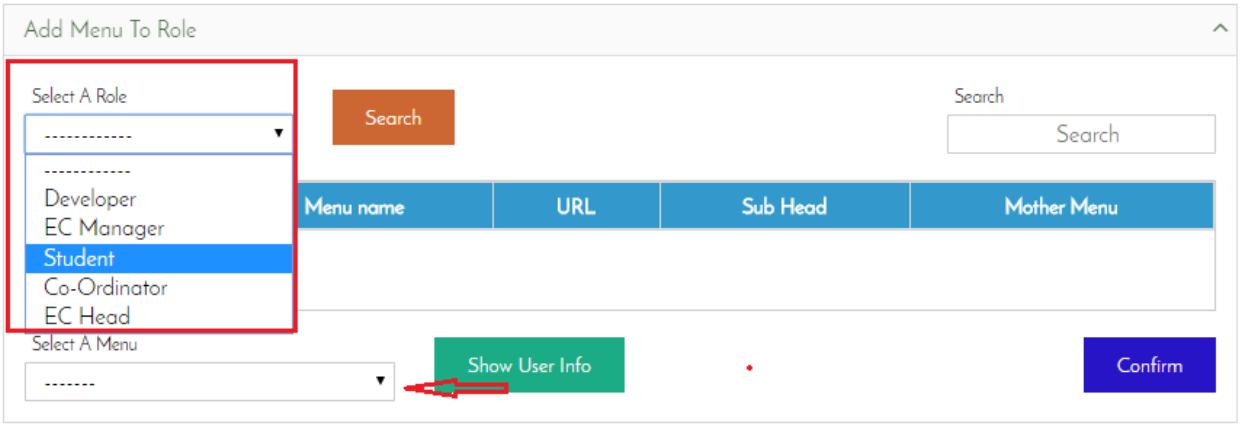


Figure 37 How to add menus to a role

- Admin can create any new role to the System:

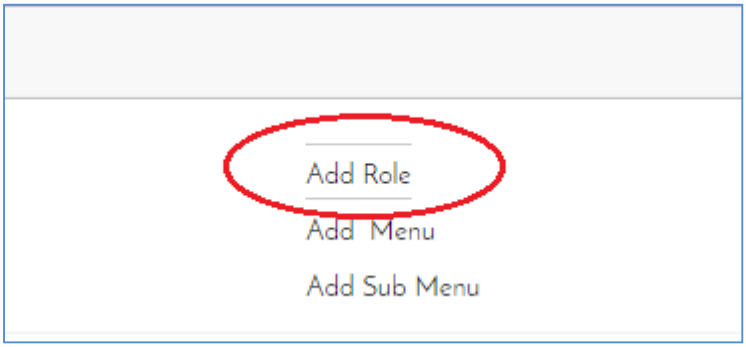


Figure 38 Option to add a role

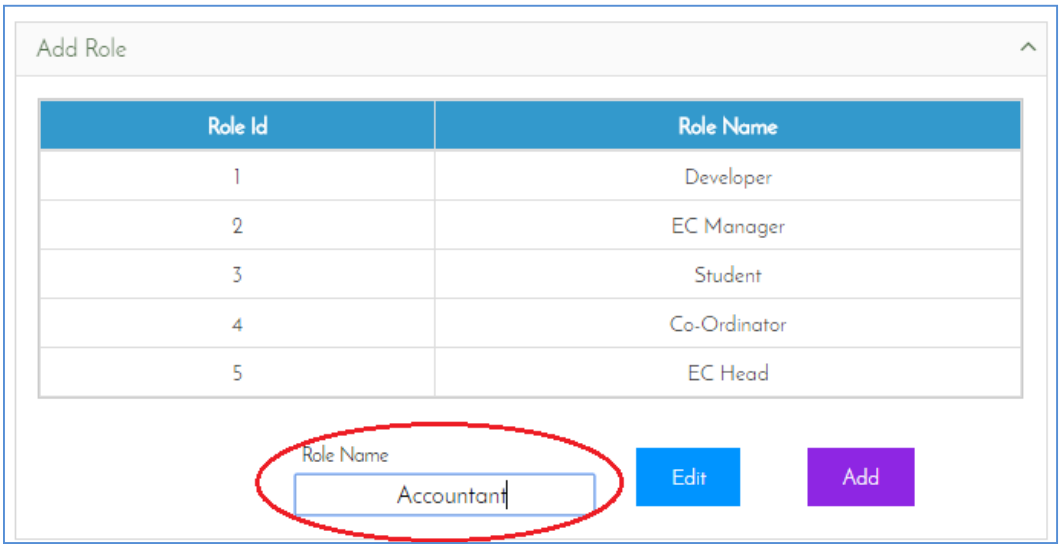


Figure 39 Role adding process

Figure 40 Role added to the list

- One more big is Admin can manage assessments by fixing date:

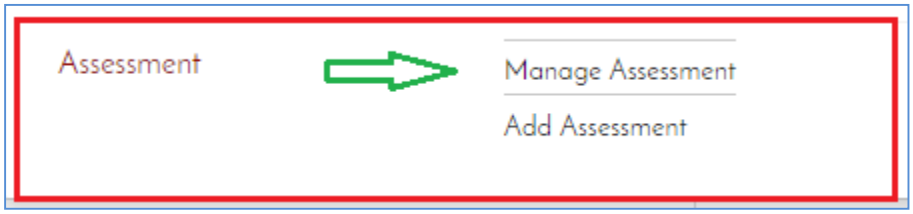


Figure 41 Assessment sub menu

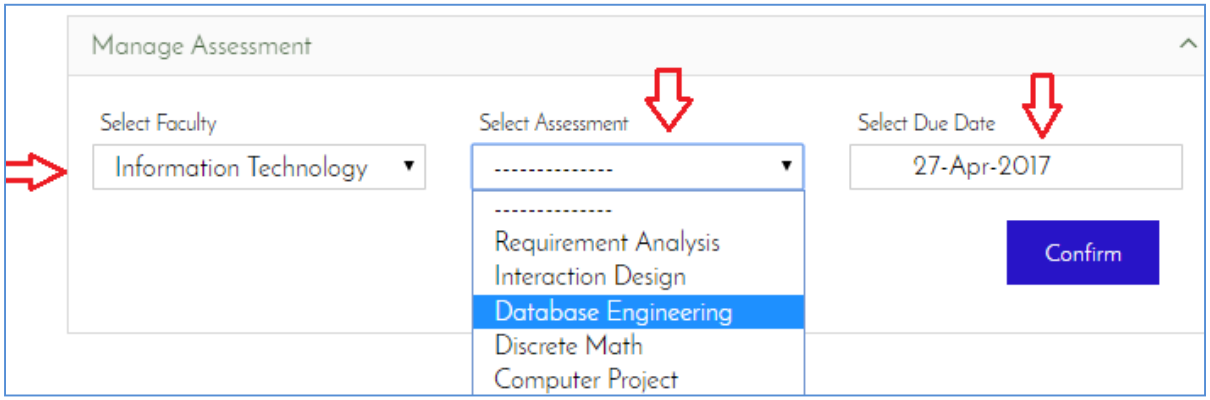


Figure 42 Manage assessment page

- Admin can add assessment by inputting names:

Figure 43 "Add assessment" menu option

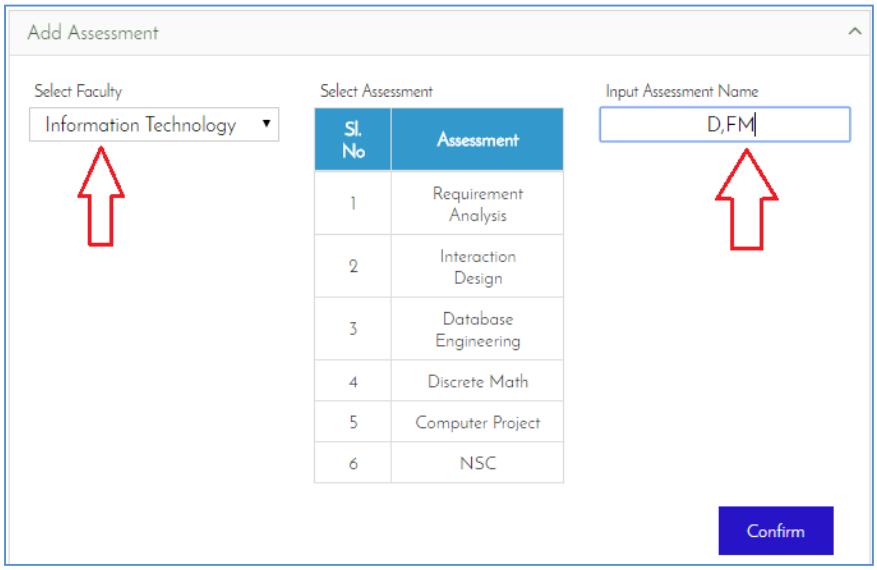


Figure 44 An assessment input

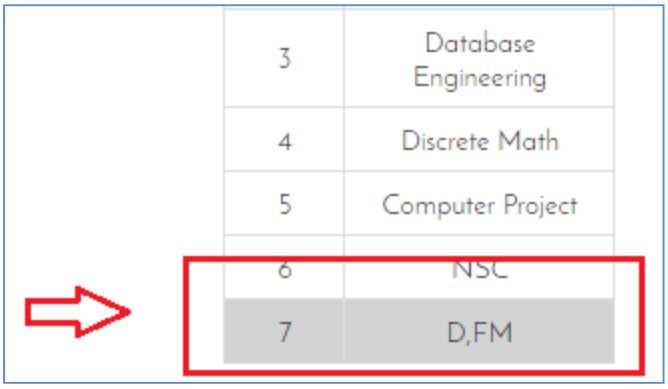


Figure 45 New assessment has been added

Test No.	12
Test type	Usability testing
Test name	Navigation testing of web pages
Expected result	Menu, links, button will be visible and accessible
How to test	Carried out by manually checks and taking screenshots
Actual result	All the navigation are visible and accessible

- This is the proof of Usability of the system:

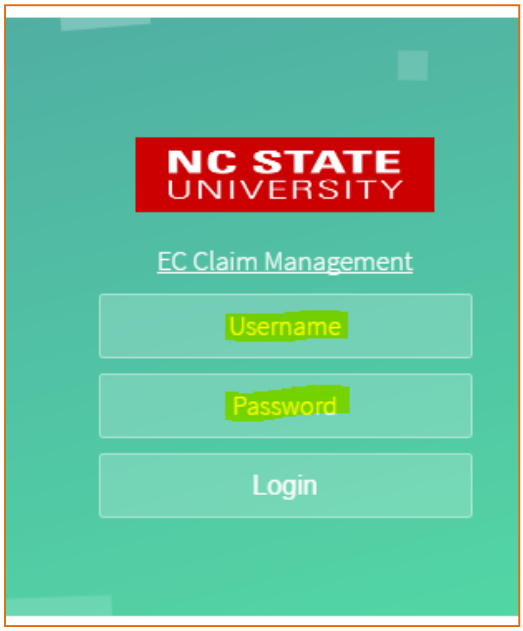


Figure 46 Usable login form

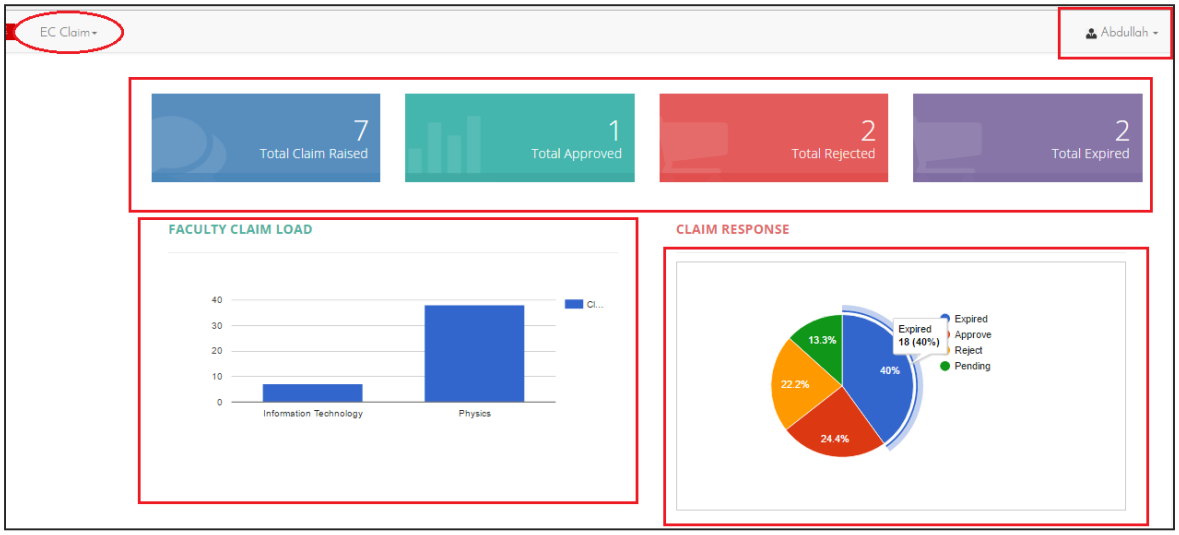


Figure 47 Navigation is are visible, Showing a accessible dashboard etc.

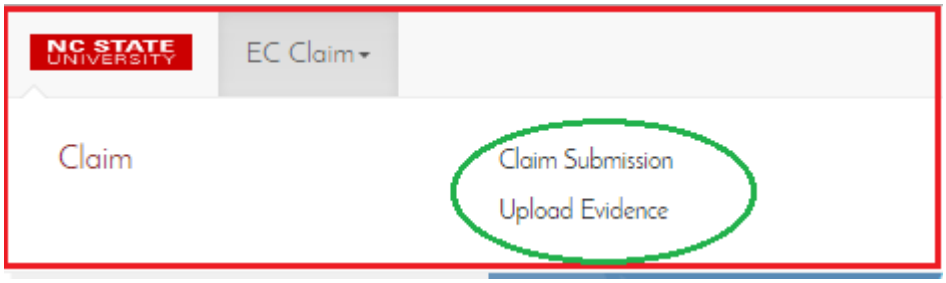


Figure 48 accessible Menus and sub menus

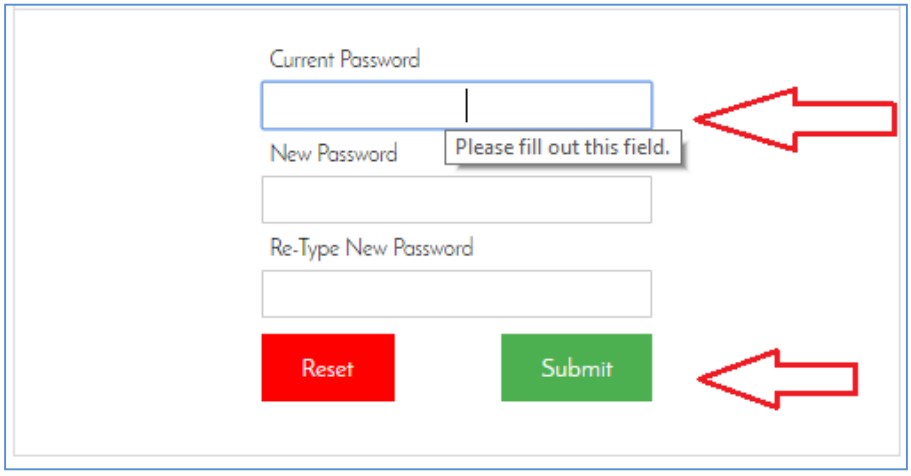


Figure 49 Clearly visible and accessible forms and buttons

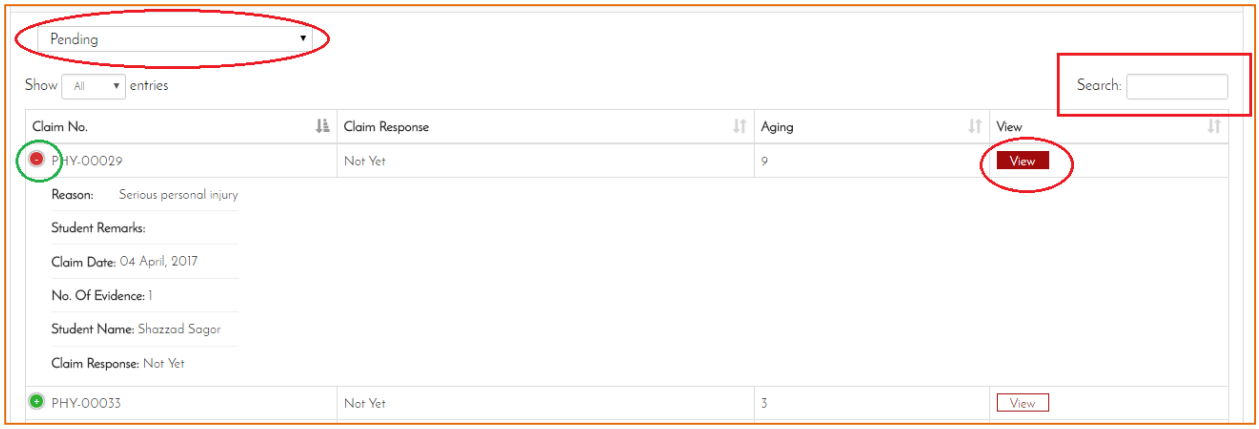


Figure 50 Another type of accessibility

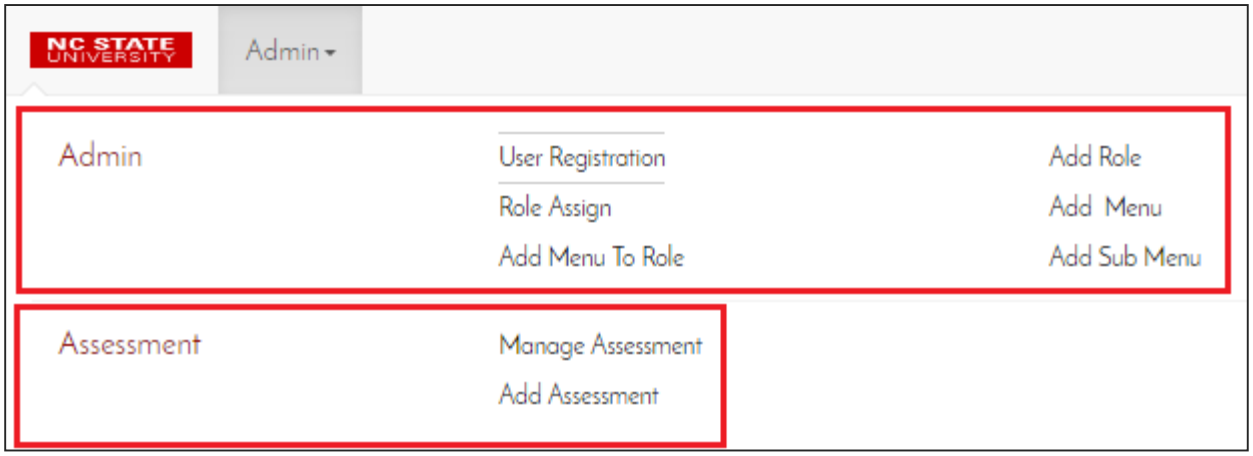


Figure 51 Accessible navigation from user perspectives

Test No.	13
Test type	Usability testing
Test name	Content testing of WebPages
Expected result	Pages color expected to have light color, proper image size and anchor links should be visible
How to test	Carried out by manually checks and taking screenshots
Actual result	Pages colors are light color, proper image size and anchor links are as excepted

- This section shows that the good page color(light), Propoer image size etc:

Figure 52 Login shows a good color matching with the overall interface

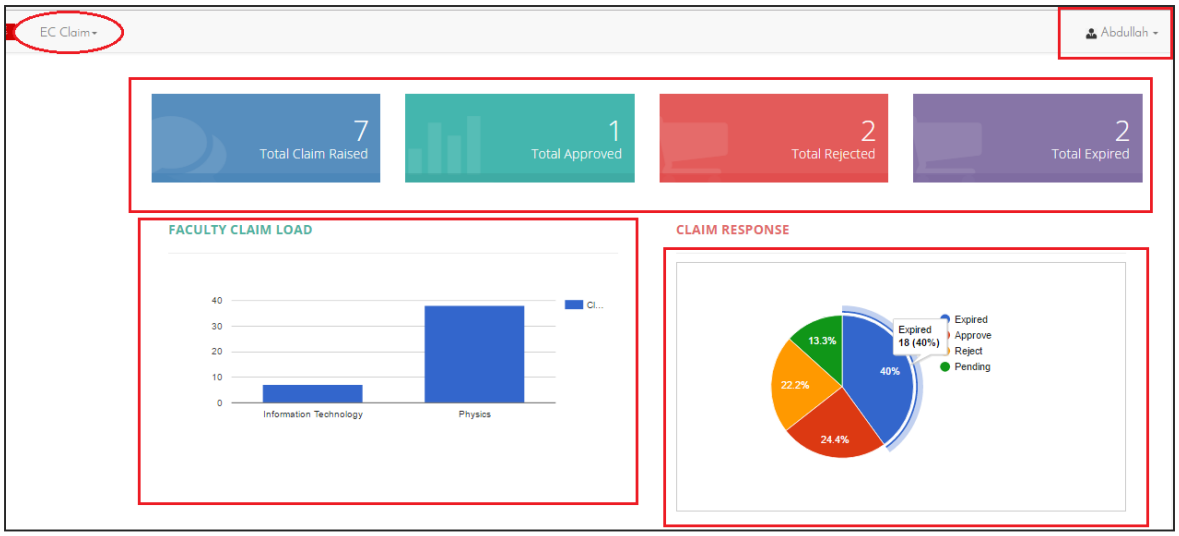


Figure 53 this section show different content color with

Test No.	14
Test type	Compatibility testing
Test name	Browser compatibility
Expected result	Testing will show compatibility in different results in different browsers
How to test	Checking the application in different browser
Actual result	App is compatible with different browser show the different

- Test with Google chrome proof of a home page:

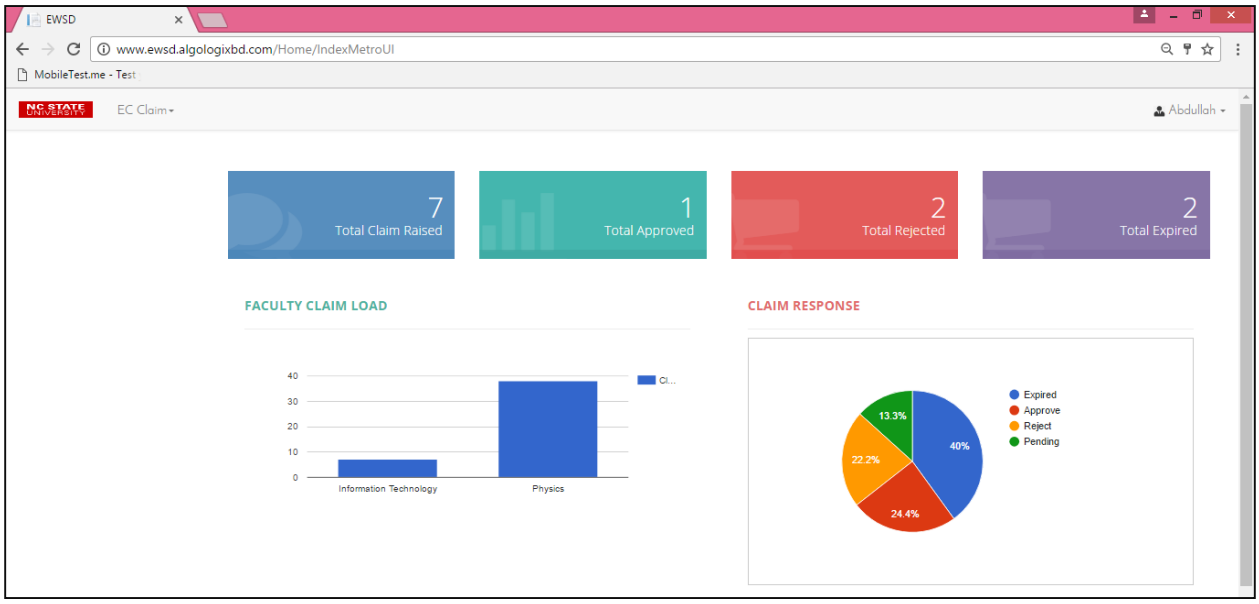


Figure 54 Chrome result of Home page with dashboard

- Testing result with Mizilla firefox:

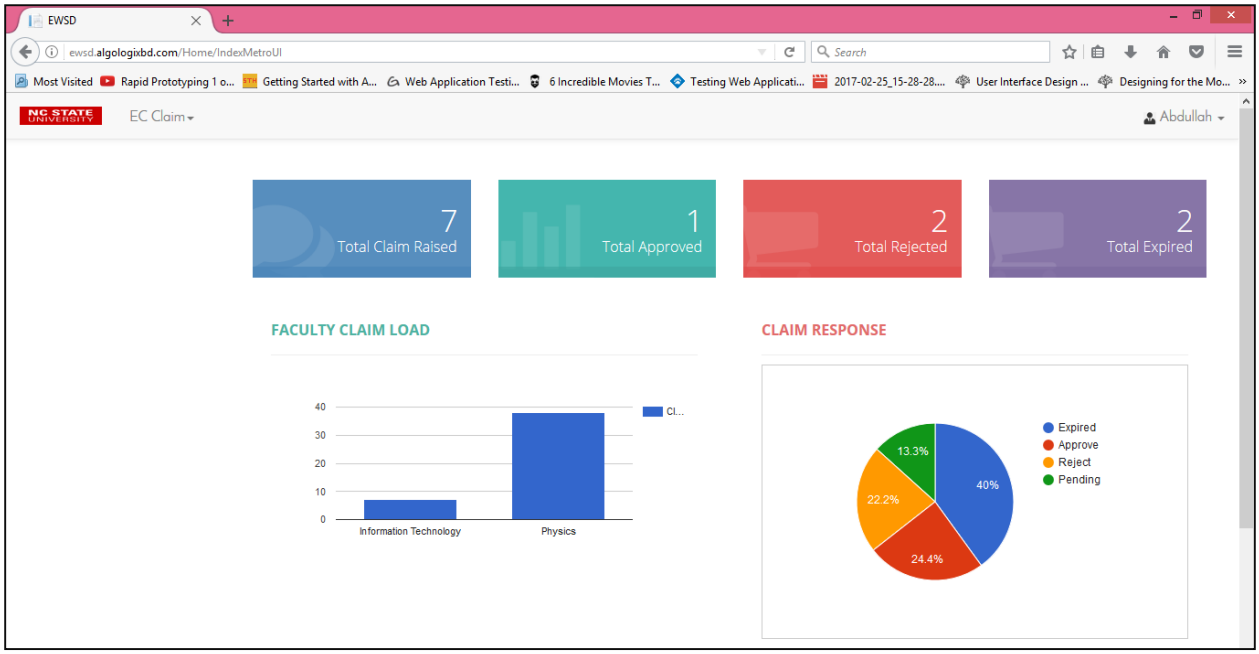


Figure 55 Mozilla Firefox show this result

- Testing with Internet explorer:

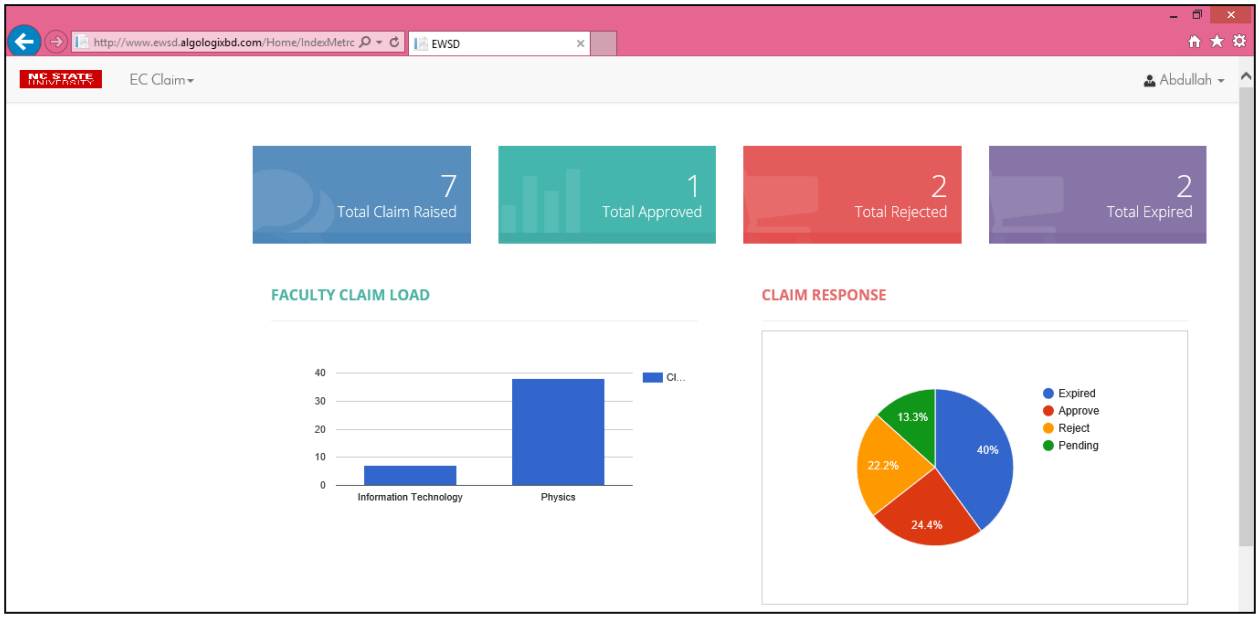


Figure 56 Internet Explorer result of Home page

Test No.	15
Test type	Compatibility testing
Test name	Operating system compatibility
Expected result	Testing will show compatibility in different operating system
How to test	Checking the app different OS
Actual result	App is compatible with different OS browser show the different result

- Show output from different OS and different browser:

Test No.	16
Test type	Compatibility testing
Test name	Mobile compatibility
Expected result	Application is compatible with different mobile browsers
How to test	Checking the app in mobile browser by taking screenshot
Actual result	Found compatibility in mobile platform

- This show different mobile compatibility testing result or output:

Figure 57 compatible with iPhones

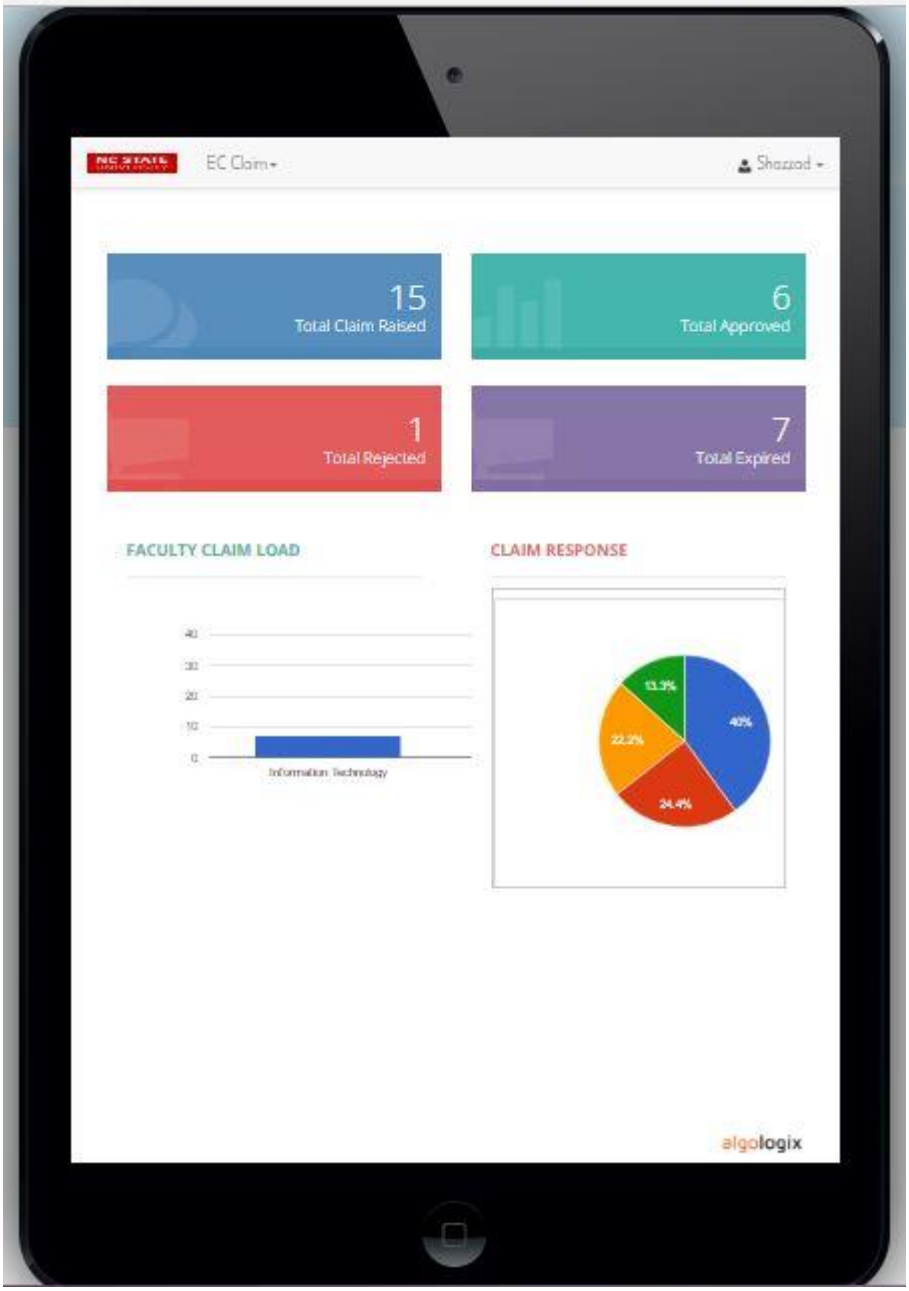


Figure 58 Compatibility test with iPad

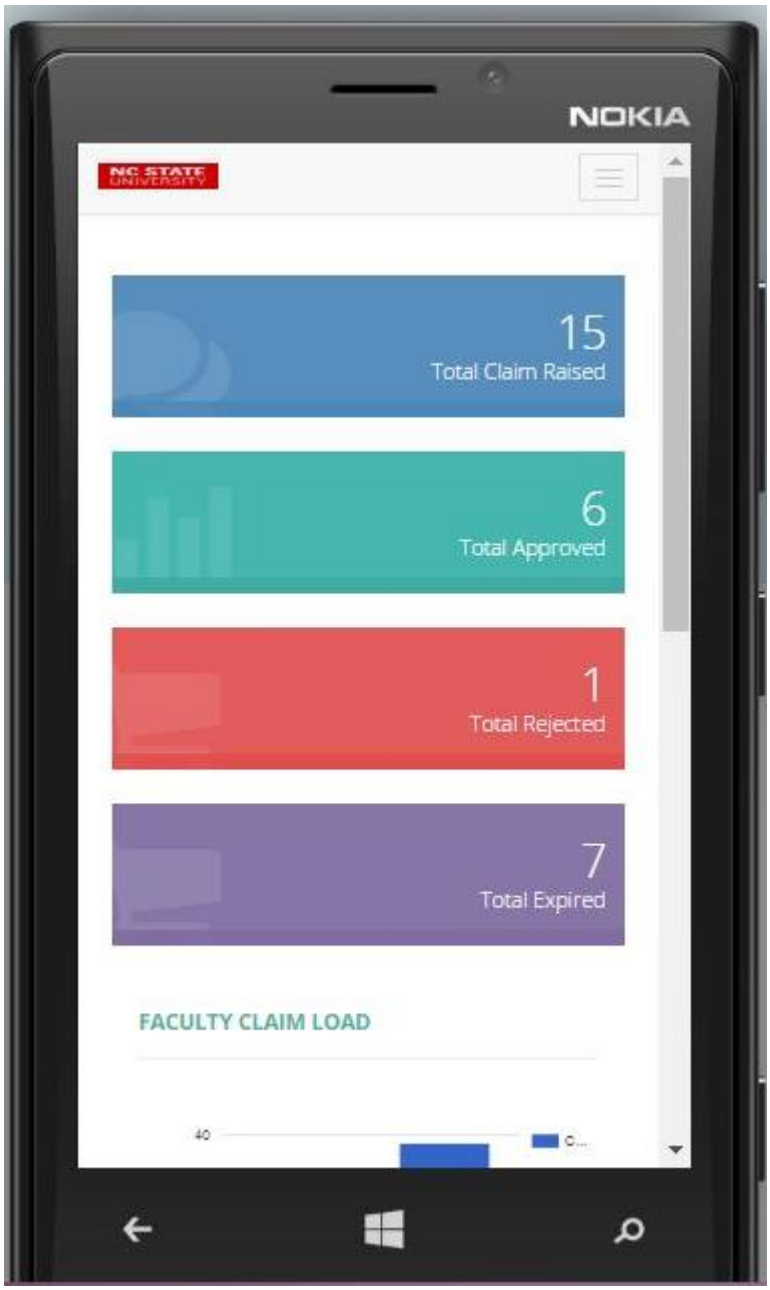


Figure 59 Compatibility testing with Lumia 920

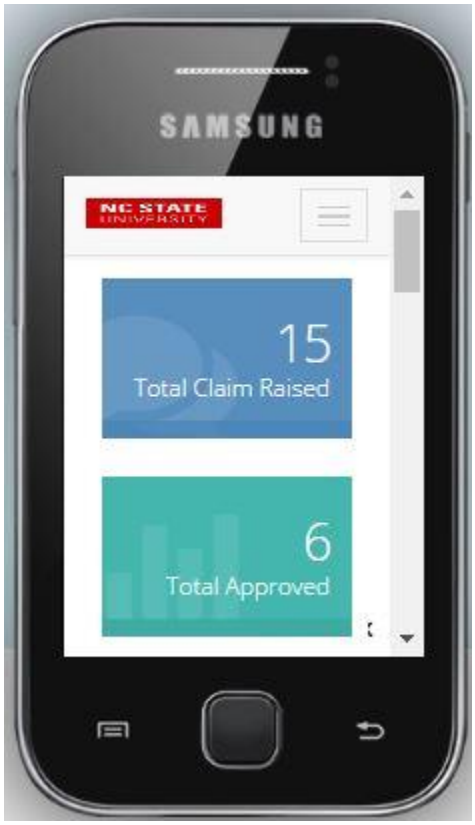


Figure 60 Compatible with a minimum smal screen of Galaxy Y

Test No.	17
Test type	Compatibility testing
Test name	Printing facility
Expected result	Application pages are compatible during printing
How to test	By providing print command and print preview page.
Actual result	Found printing friendly web pages

- Testing of how to see if a page is printed:

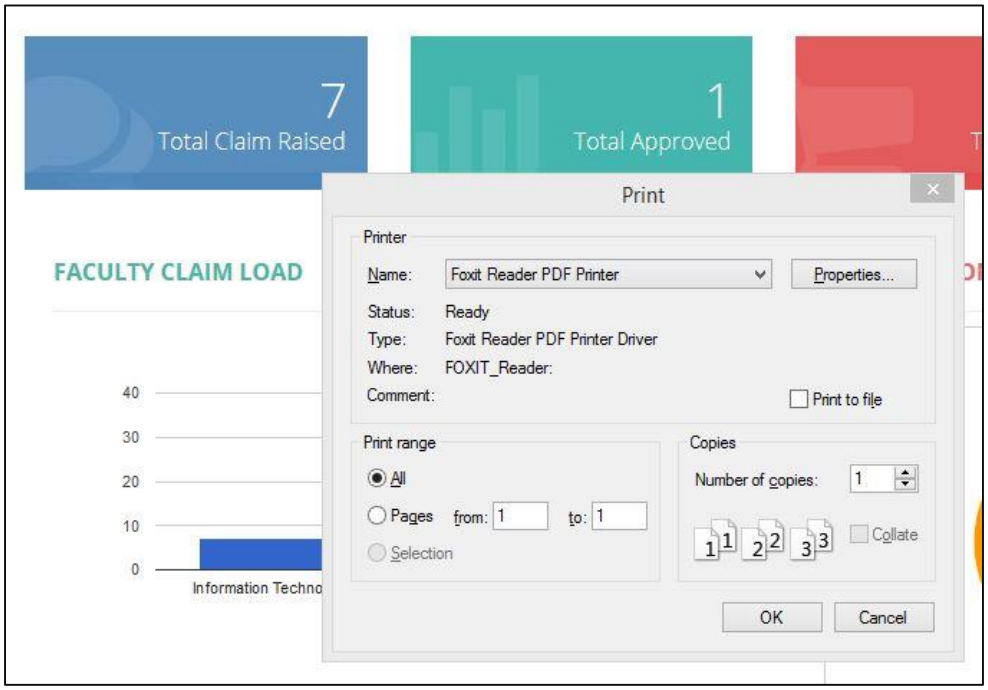


Figure 61 Providing printing command with foxit reader printer

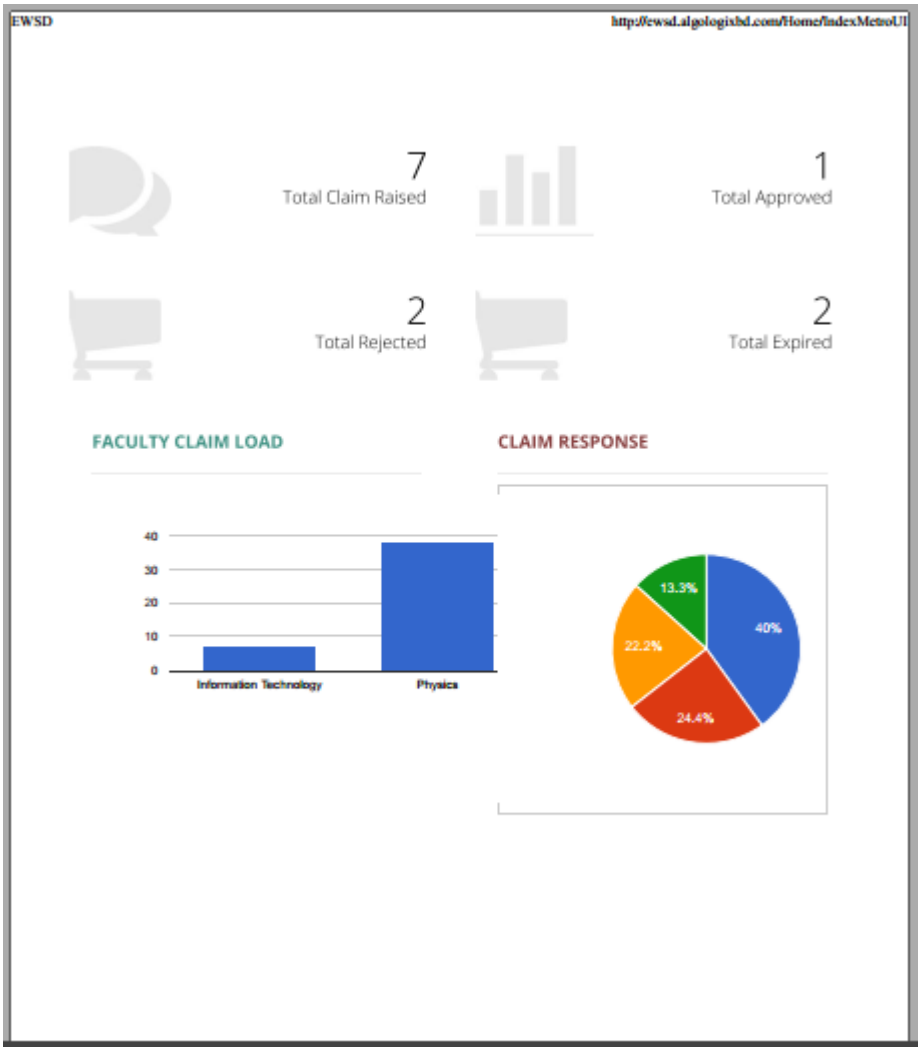


Figure 62 Compatibility testing with print and print preview home page

Test No.	18
Test type	Interface testing
Test name	Web server interface
Expected result	Have to be good
How to test	Manually
Actual result	As expected

Test No.	19
Test type	Interface testing
Test name	Application server interface
Expected result	Good As much as possible
How to test	Doing manually
Actual result	As expected

Test No.	20
Test type	Interface testing
Test name	Database server
Expected result	Behavior is good
How to test	Carried out manually
Actual result	As expected

Test No.	21
Test type	Database testing
Test name	Database connection
Expected result	Connection establishment between app and database
How to test	By checking default data that are connected with database
Actual result	Database connection has been established

- Testing database connection is okay in system:

Figure 63 Figure shows database is connected with system

Test No.	22
Test type	Database testing
Test name	Query runs without error
Expected result	Queries will be able to run without error
How to test	By executing a query from the application
Actual result	Queries executed successfully

- Queries are able to run without any error:

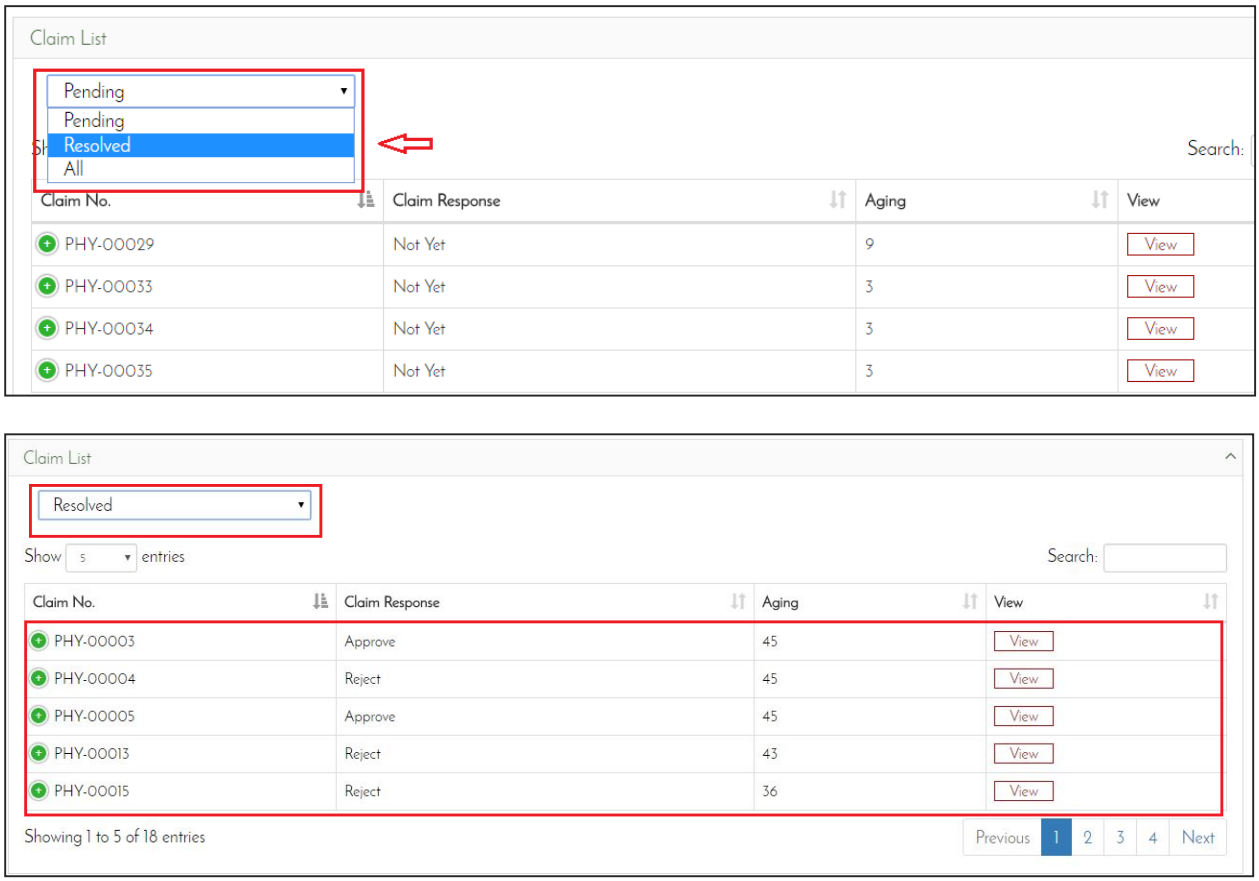


Figure 64 Queries executed successfully

Test No.	23
Test type	Database testing
Test name	Data integrity with CRUD functions
Expected result	Data consistency should be okay from affect of CRUD integrity
How to test	Operate CRUD operation on portion
Actual result	Data are consistence among the tables

Test No.	24
Test type	Database testing
Test name	Time takes to execute query
Expected result	Database is efficient enough to execute queries
How to test	Run a query and observe the duration
Actual result	As expected

Test No.	25
Test type	Performance testing
Test name	Web load testing
Expected result	Load testing performance
How to test	By using
Actual result	As expected

- This part shows web load testing result:

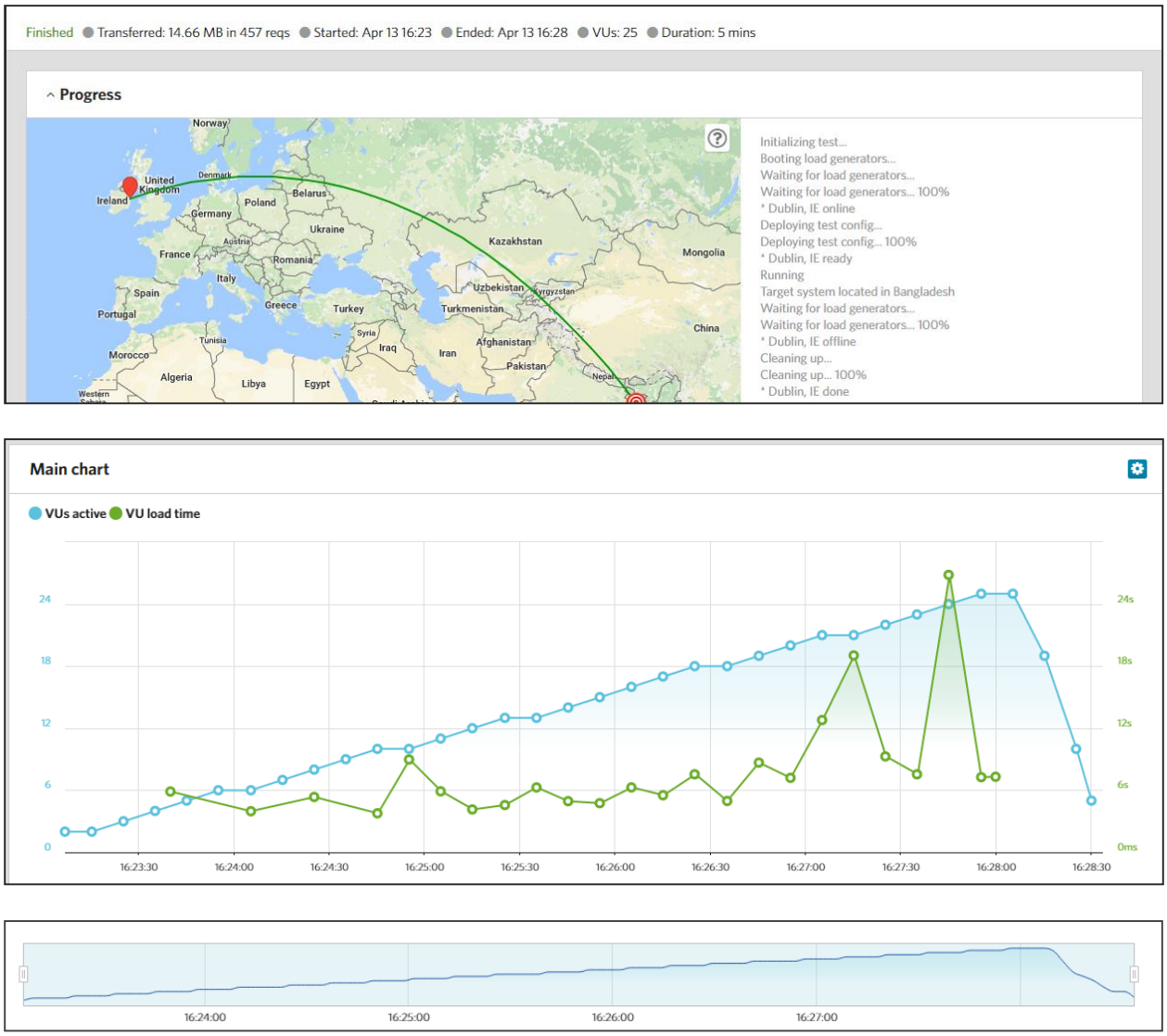


Figure 65 the above figures show performance testing

Test No.	26
Test type	Performance testing
Test name	Web stress testing
Expected result	Stress testing performance
How to test	By using
Actual result	As expected

- This part shows web stress testing result:

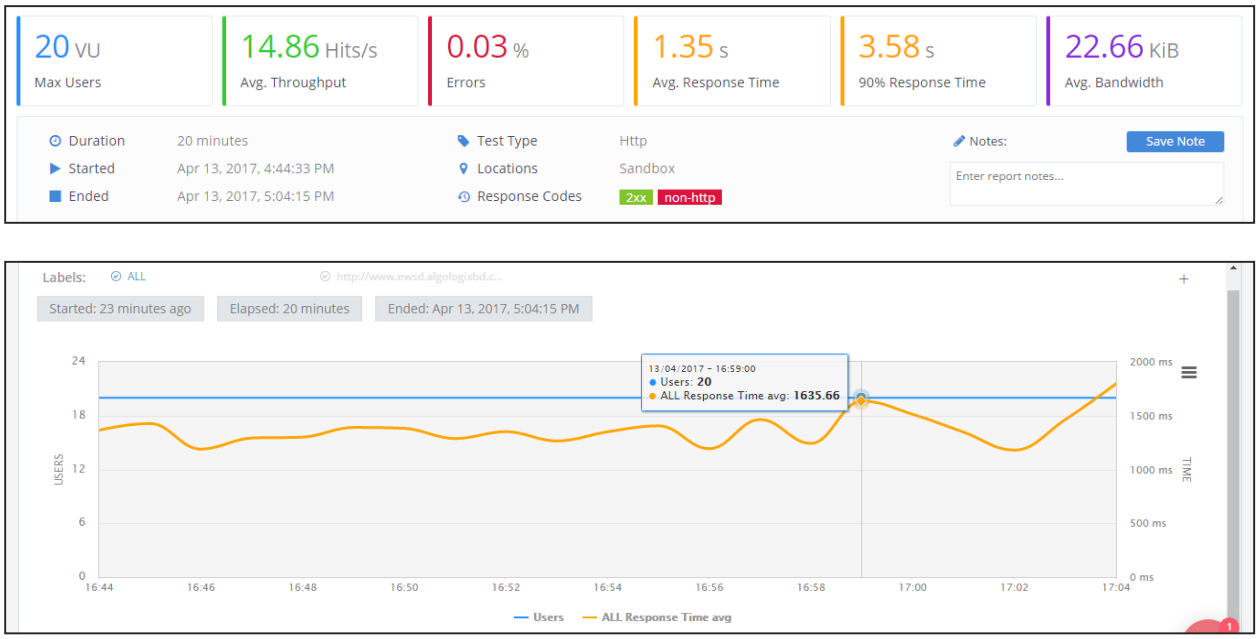


Figure 66 Show report the stress of webApp

Test No.	27
Test type	Security testing
Test name	Force URL testing
Expected result	Doesn't allow anyone to access from direct URL (except logged in mode)
How to test	Copy a URL, log out from system and paste it to address bar and hit enter
Actual result	Showing login page rather showing URL's page

- Shows that system prevent to force URL:

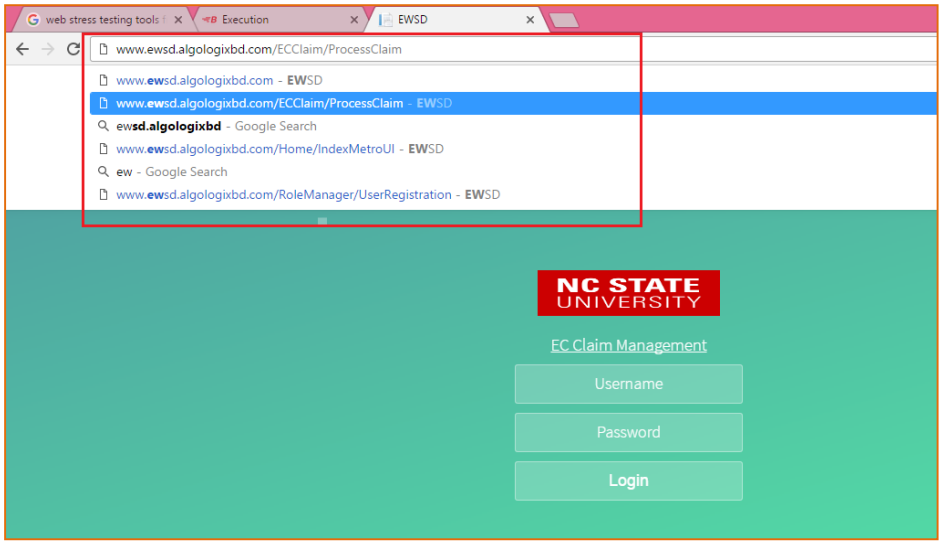


Figure 67 System doesn't allow force URL

Test No.	28
Test type	Security testing
Test name	Checking the passwords are encrypted
Expected result	No one can see the real password
How to test	Go to the database and see the password fields
Actual result	As expected

- The results of checking encrypted password by Hash:

100 %

Results | Messages

	AutoId	UserId	UserName	FacultyCode	Password	PseudoPassword	UserStatus
1	6	1	dev	007	81DC9BDB52D04DC20036DBD8313ED055	NULL	1
2	5	2	ratul	F001	81DC9BDB52D04DC20036DBD8313ED055	819453651	1
3	3	3	shazzad	F002	81DC9BDB52D04DC20036DBD8313ED055	1563592509	1
4	11	4	dalim	F002	81DC9BDB52D04DC20036DBD8313ED055	1102214029	1
5	8	6	siplu	F002	81DC9BDB52D04DC20036DBD8313ED055	NULL	1
6	9	8	mustafiz	F001	81DC9BDB52D04DC20036DBD8313ED055	NULL	1
7	10	9	sarwer	F001	81DC9BDB52D04DC20036DBD8313ED055	1140842348	1

Figure 68 Passwords are encrypted

Test No.	29
Test type	Security testing
Test name	SQL injection testing
Expected result	System will protest sql inject problem
How to test	Testing with a tools
Actual result	As expected

- Our system is fully protected from this attack, but can't check it via tools due to limited time crash possibility.

Test No.	30
Test type	Security testing
Test name	Brute force attack
Expected result	System able to protest brute force attack
How to test	Testing with a tool
Actual result	As expected

- Shows the result of this:

Figure 69 an attempt to attack hacking password

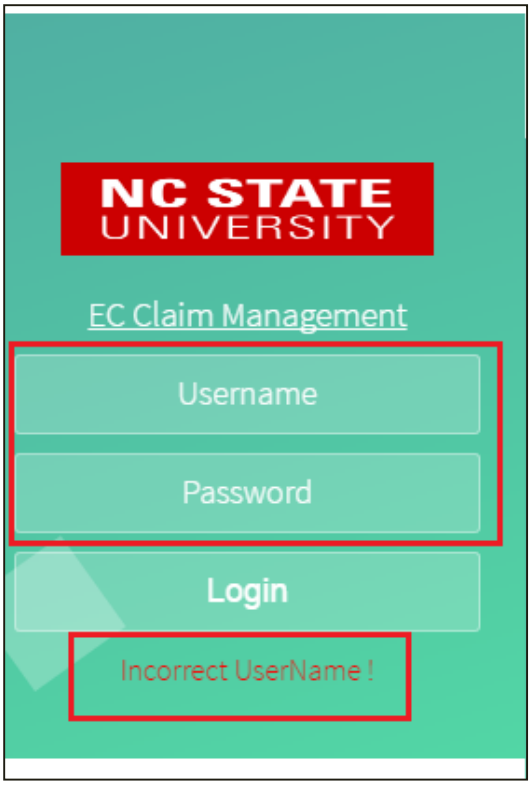


Figure 70 Prevents illegal access.

Test No.	31
Test type	Security testing
Test name	Authentication testing
Expected result	Unauthorized user cant access to the system
How to test	Try to access to the system by putting unauthorized ID and passwords
Actual result	Shows wrong username and password

- Authenticated user login role based by input wrong id password:

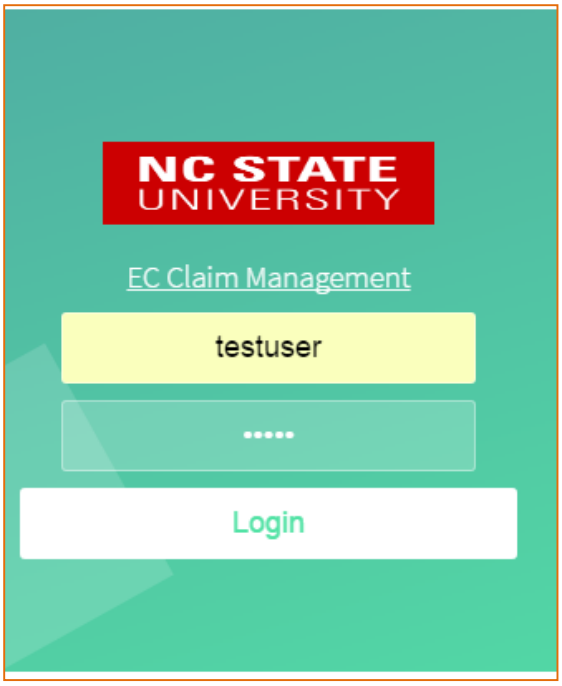


Figure 71 Unauthorized username and password entered

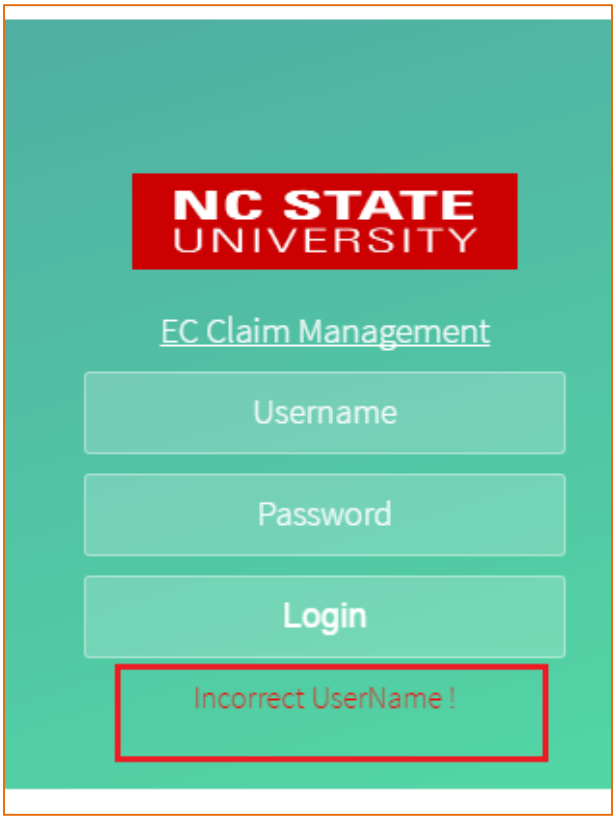


Figure 72 Shows "Incorrect username"

Test No.	28
Test type	Crowd testing
Test name	Carry out by specific group of non technical person (end user)
Expected result	Positive attitude and feed to the system
How to test	Let that three (3) person to use the system for a while
Actual result	As expected

References:

Kota, K. (2016) *AdminiTrack, Inc.*, 10 October, [Online], Available: https://www.adminitrack.com/articles/testingyourwebapps.aspx [2017 March 25].

Pressman, R. (2010) *Software Engineering*, 7th edition, New York: McGraw-Hill Companies, Inc.

SoftwareTestingClass.com (2012) *softwaretestingclass.com*, 12 November, [Online], Available: http://www.softwaretestingclass.com/web-application-testing/ [27 March 2017].

softwaretestingfundamentals.com *http://softwaretestingfundamentals.com*, [Online], Available: http://softwaretestingfundamentals.com/test-plan/ [10 March 2017].