SLIIT UNI
THE KNOWLEDGE UNIVERSITY

BSc (Hons) in Information Technology

Specializing in Software Engineering

Year 3 - 2021

SE3040 – Application Frameworks

Group Project

Technical Report

Light Waves

2021S1_REG_WD_17

Conference Management System

Student ID	Name
IT19123196	**K.H.T.N Dewangi**
IT19159072	**A.N.S Thenuwara**
IT19129518	**Rathnayaka R.M.D.M**
IT19140308	**D.K.M Dissanayaka**

Repository Link: https://github.com/TharushiNDewangi/AF-ConferenceManagementSystem.git

---For Evaluators' use only---

Evaluator's Name:

Comments:

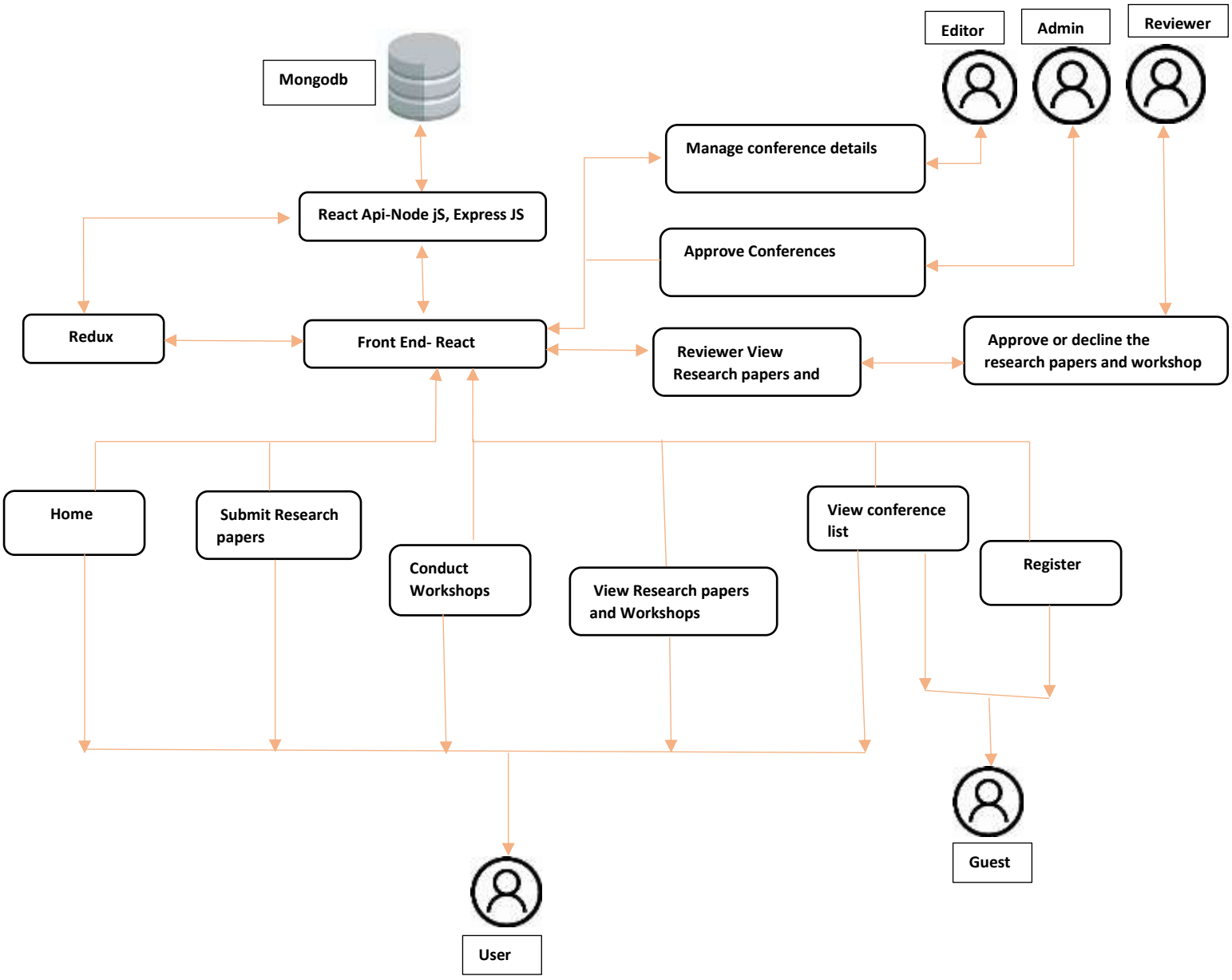
Contents

Module Architecture Diagram 3
Use case Diagram 4
Unregistered User 5
Registered User 5
Admin 5
Reviewer 5
Editor 5
Following technologies were used for the implementation 5
Further more following packages have been used 5
Description of Functionalities 6
- **Home page** 6
- **Register and Login** 6
- **Add research related activities and Workshops** 6
- **Add conference details** 6
- **View Research related activities and Workshops** 6
- **Approve Research related activities and Workshops** 7
- **Approve Conference Details** 7

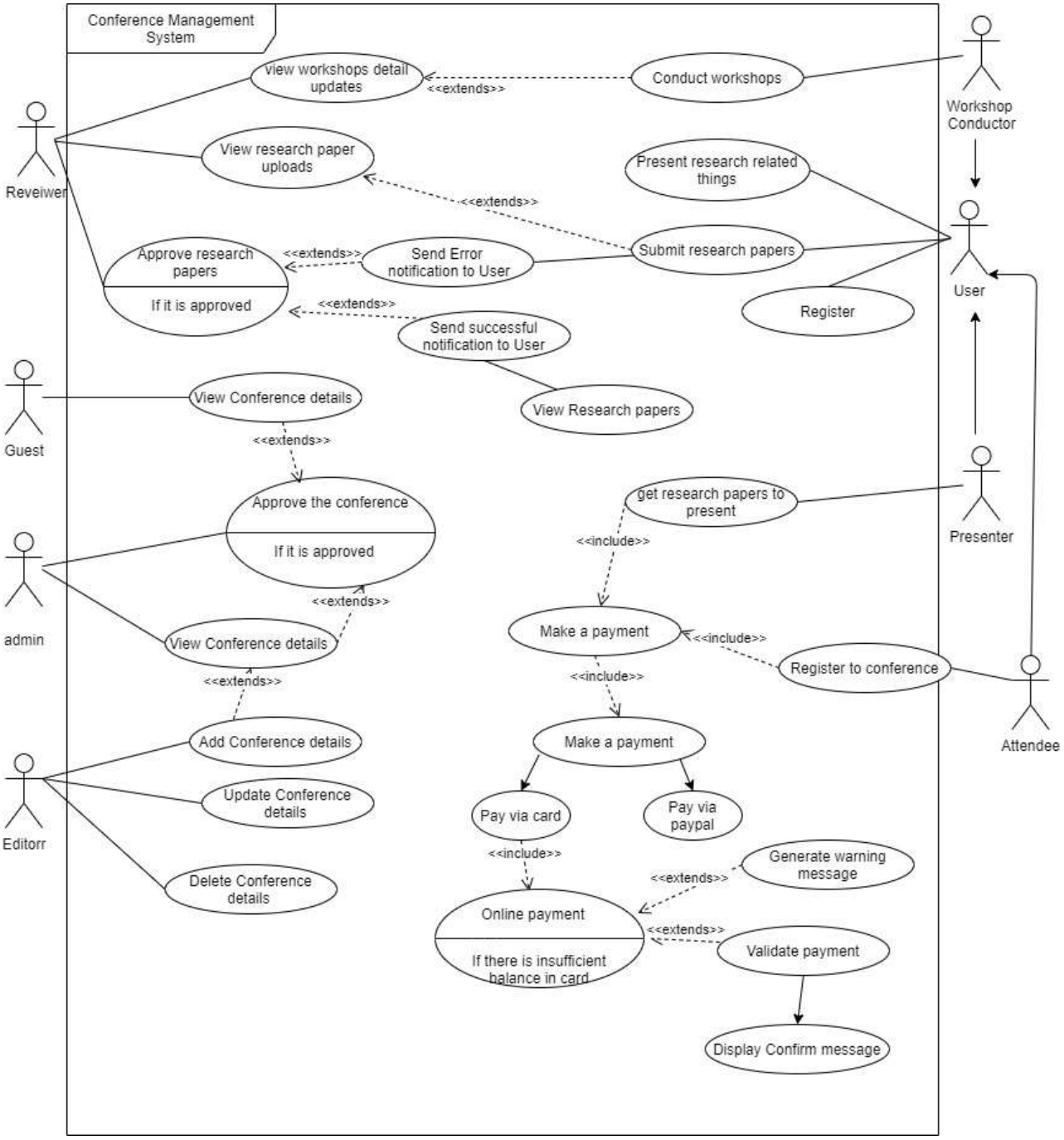
Front end(UI/UX) 7
Back End (REST API) 7
Usage of Rest API 8
Test cases 9
- **Login** 9
- **Registration** 9
- **Download Research paper** 10
- **Register to the conference** 10
- **Approve Conference details** 11

Screenshot of Mongo database collections 12

Module Architecture Diagram



Use case Diagram



Assumption: Attendee ,workshop conductor research paper presenters all are users in system. there are not different user roles in the database. (We use separate it in use case because then we can identify it easily)

This system was developed for a conference management purposes to simply conduct workshops, conferences, Research and related activities. Several technologies have been used for different components to implement the system. System is tightly secured and well authentication mechanisms are used through out the system.

There are five system roles in the system.

Unregistered User

Unregistered users can view conference details and register to the system then they become registered user.User can be registered as a researcher , workshop presenter or attendee.

Registered User

Registered users can add research related activities, conduct workshops and view conference details, research related activities , workshops.

Admin

Admins can approve editor's content before it appears on the website.

Reviewer

Reviewer can see the research paper uploads and workshop detail uploads in separate pages.

And Reviewer can approve or decline the research papers workshop proposals. And sent notification to the relevant user**.**

Editor

Editors can add the conference details and edit.

Following technologies were used for the implementation

- ✓ **HTML/JavaScript**
- ✓ **ReactJs**
- ✓ **MaterialUI**
- ✓ **NodeJs**
- ✓ **ExpressJs**
- ✓ **JSON base web Services**
- ✓ **NoSQL database (MongoDB)**

Further more following packages have been used

- ✓ **React Router to private routing**
- ✓ **React JWT to authenticate**
- ✓ **Redux to store states and variables**

Description of Functionalities

Home page

Home page has sub components header, footer and all the related information including venue and date.HTML has been used with CSS to develop home page .Bootstrap has been used to build a responsive system .React has been used to develop the homepage.

Register and Login

User can register using an email and other details. Using this registration page user and register as a researcher ,workshop presenter. After successfully register user can log in using given credentials and every logged user given token . It is for authentication purposes.

JWT has been used to issue tokens and authenticate. Express JS and NodeJS have been used to develop BACKEND API.

Add research related activities and Workshops

Users can add research papers and conduct workshops. Only users who have access can submit research papers and conduct workshops.

When researcher registered to the system the research paper uploaded alongside the contact information.

When the workshop conductor registered to the system a proposal containing all the necessary details about workshop should be uploaded alongside the contact information.

HTML, React ,Bootstrap and CSS have been used to develop this components .And axios has been used to send http requests, Express JS and Node JS have been used to develop BACKEND API

Add conference details

Editors can add Conference details and edit. When Editor registered to the system the Conference uploaded alongside the contact information.

HTML, React ,Bootstrap and CSS have been used to develop this components .And axios has been used to send http requests, Express JS and Node JS have been used to develop BACKEND API

View Research related activities and Workshops

Reviewers can view research paper uploads and workshop details uploads in separate pages.If user want to download these papers or workshop proposals when user click on link it will navigate to separate download page .

HTML ,React and CSS have been used to develop the frontend. Bootstrap has been used to grid view. Research papers and workshops are retrieved by sending http client using axios and store the data in redux store.

Approve Research related activities and Workshops

Reviewer should approve or decline research related activities and workshop details and send notification using nodemailer to relevant user. Reviewer see these things in separate pages and if it is relevant ,

approve it , then registered users can view these details HTML ,React and CSS have been used to develop the frontend. Bootstrap has been used to grid view. And axios has been used to send http requests, Express JS and Node JS have been used to develop BACKEND API. In database we use separate column for approved or rejected if it is rejected, send notification to user for edit it and again upload .

Approve Conference Details

Admin should see Conference Details and if it is relevant ,approve it , then registered users can view these details HTML ,React and CSS have been used to develop the frontend. Bootstrap has been used to grid view. And axios has been used to send http requests, Express JS and Node JS have been used to develop BACKEND API. In database we use separate column for approved or rejected if it is rejected , send notification to Editor for edit it and again upload

Front end(UI/UX)

React, HTML,CSS used to develop front end. Bootstrap has been used to build a responsive system.

Back End (REST API)

Back end has been developed using node JS and Express JS .Back end is a REST API and front end communicated with server using only available web services .Axios has been used to communicate with backend Axios is light weight http client.

Usage of Rest API

Rest API is developed using React.it runs on port 8065.Rest API has admin, editor, reviewer ,user , Research papers, workshops, conference details , approve details, payment CRUD operations and send email notifications as well .Rest API has access to mongoose database and web client could access mongoose through this Rest API. This is research paper route. we use that way to implements all REST API. we give access permission like that way in route

```
router.post('/research/create',requireSignin,usermiddleware,upload.single('catimg
'),aaddres)
router.delete('/research/deleteres',requireSignin,usermiddleware,delres)
router.get('/research/getres',getres)
router.put('/research/approvecat/:_id',requireSignin,reviewermiddleware,approvere
s)
```

These are some of end points in our system

End point	Description
http://localhost:8065/api/reviewer/signin	Reviewer sign in
http://localhost:8065/api/user/signin	User sign in
http://localhost:8065/api/editor/signin	Editor sign in
http://localhost:8065/api/admin/signin	Admin sign in
http://localhost:8065/api/reviewer/signout	Reviewer sign out
http://localhost:8065/api/user/signout	User sign out
http://localhost:8065/api/editor/signout	editor sign out
http://localhost:8065/api/admin/signout	admin sign out
http://localhost:8065/api/signup	Sign up
http://localhost:8065/api/paper/create	Add research paper
http://localhost:8065/api/getpapers	Get all research papers
http://localhost:8065/api/conference	Add conference details
http://localhost:8065/api/conference/:id	Get conference details by id
http://localhost:8065/api/conference/update/:id	Update conference details by id
http://localhost:8065/api/paper/aprrove/:id	Approve research papere by id
http://localhost:8065/api/conference/approve/:id	Approve conference details by id

Test cases

Login

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
Cl1	Check user login response when valid email and password is entered	1)Enter email address 2)Enter password 3)Click sign in	Email: dewsdrv@gmail.com Password: dewsdrv@98	Login should be successful	Login was successful	pass
Cl2	Check user login response when invalid email and password is entered	1)Enter email address 2)Enter password 3)Click sign in	Email: hell@gmail.com password: hell	User should not login and Get warning "password should be 6 characters"	Get warning "password should be 6 characters"	fail

Registration

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
Rl1	Check user information valid when registering	1)Enter user name 2)Enter email address 3)Enter password 4)Enter user role 5)Click Register	Name: Tharushi Nimasha Role: reviewer Email: dewsdrv@gmail.com Password: dewsdrv@98	Registration should be successful	Registration was successful	pass
Rl2	Check user information Invalid when registering	1)Enter user name 2)Enter email address 3)Enter password 4)Enter user role 5)Click Register	Name: Hell margrat Role: admin Email: hell@gmail.com password: hell	User should not register and Get warning "password should be 6 characters"	Get warning "password should be 6 characters"	fail
Rl3	Check user information Invalid when registering	1)Enter email address 2)Enter password 3)Enter user role	Role: user Email: harry@gmail.com password: harry98	User should not register and Get warning "user name is required"	User should not register and Get warning "user name is required"	fail

		4)Click Register				

Download Research paper

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
Dl1	Check user role is valid when download research papers	1)go to research paper downloading page 2)click download button	Role: user	Download should be successful	Download was successful	pass
Dl2	Check user role is invalid when download research papers	1)go to research paper downloading page 2)click download button	Role: Guest	Download failed and generate warning “you have register”	Download failed and generate warning “you have register”	fail

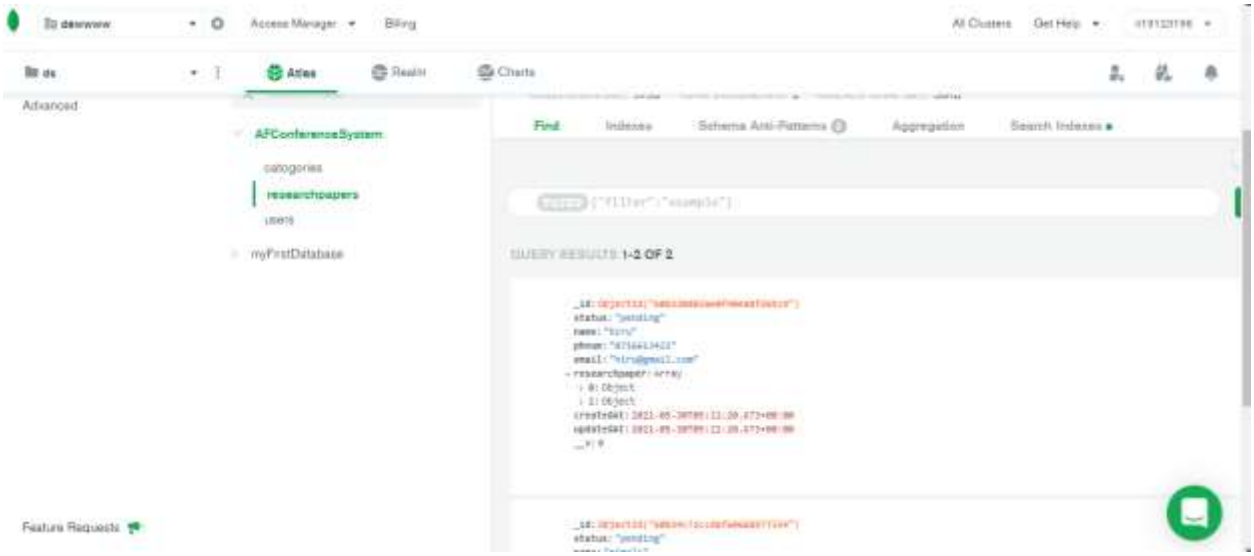
Register to the conference

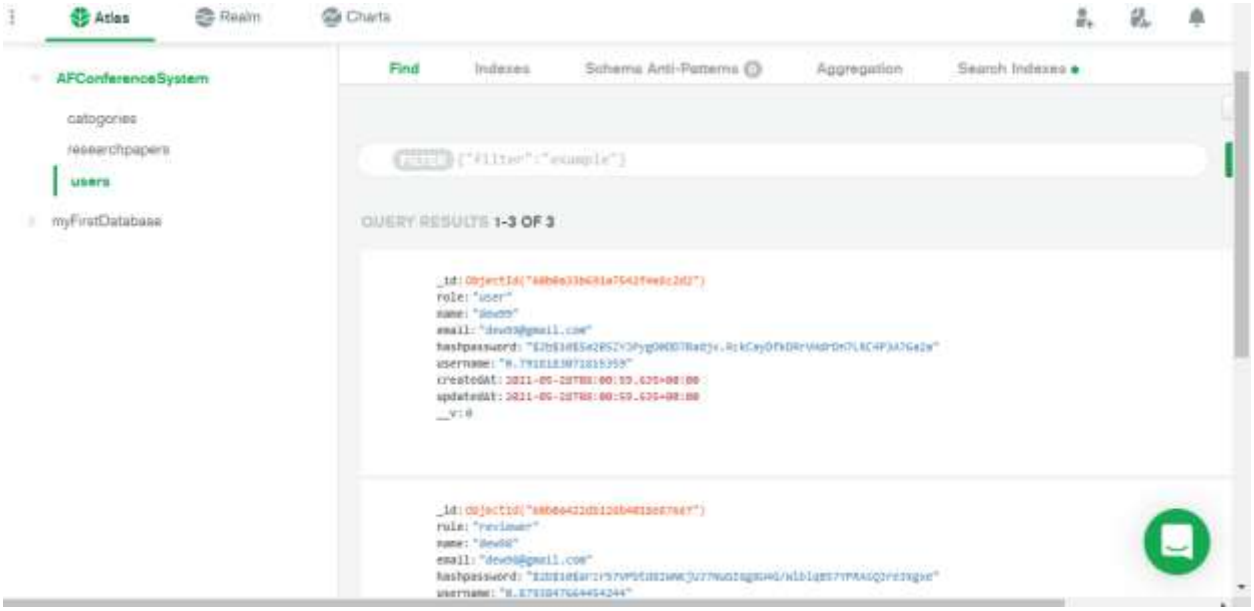
Test Case ID	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
RC1	Check user role and payment is valid when download research papers	1)go to Conference Register page 2)click Registration button	Role: user Method: PayPal	Conference Register should be successful	Conference Register successful	pass
RC2	Check user role is invalid when download research papers	1)go to research paper downloading page 2)click download button	Role: user Method: PayPal	Conference Register failed and generate warning “your account hasn’t enough amount”	Conference Register failed and generate warning “your account hasn’t enough amount”	fail

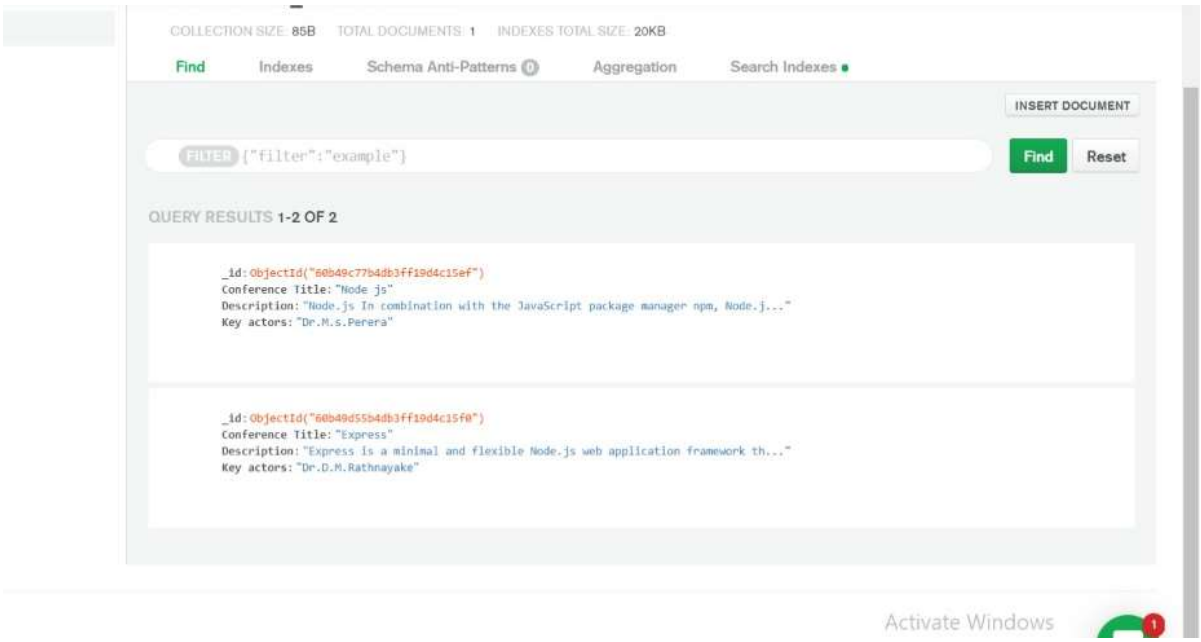
Approve Conference details

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
AC1	Check user role and valid when Approve or Reject Conference details	1)go to Conference Details Approve page 2)click Approve or Reject button	Role: admin	Conference Approve or Reject should be successful	Conference Approve was Reject should be successful	pass
AC2	Check user role and valid when Approve or Reject Conference details	1)go to Conference Details Approve page 2)click Approve or Reject button	Role: user	Generate warning “you don’t have permission to access ”	Generate warning “you don’t have permission to access ”	fail

Screenshot of Mongo database collections







(We are not cover full project yet)