Sri Lanka Institute of Information Technology

B.Sc. Honors Degree in Information Technology

Specialized in Software Engineering

Year 3, Semester 1 (2021)

SE3040 – Application Frameworks

Group Project

Technical Report

Group ID: 2021S1_REG_WD_18

Group Name: Intimidators_WD18

Student ID	Student Name
IT 19123578	Sanduni Jayamali Gamage
IT 19140476	Nandu Gamitha Manawadu
IT 19028774	Athapaththu P.N.P
IT 19143682	Gimhana P.S

Repository Link: https://github.com/SachinduG/ICAF---AF_Project.git

For Evaluations:

Marks Allocation	Leader	Member 2	Member 3	Member 4	
User Interfaces look professional and consistency - 15					
Features are comprehensive and user friendly - 15					
Implementation of the RESTful web services - 15					
Implementing database and Mongo collections - 10					
Implementation of unit tests - 10					
Coding Standards and quality - 5					
Deployed in the cloud - 5					
Maintaining online git repository - 5					
Technical Report (Group Mark) - 5					
User guide (Group Mark) - 5					
Presentation - 10					

Comments:

INTRODCTION

ICAF 2021– International Conference on Application Frameworks 2021 is a conference management tool which developed in order to the organizing of academic conference on application frameworks. This tool will intend to manage all the academic related activities that will be presented in the conference.

The tool was developed using **MERN** (mongo DB, Express, ReactJS, NodeJS) technologies. Frontend was developed using **ReactJS** JavaScript library and backend was developed using **NodeJS** JavaScript runtime environment. **MongoDB** was used as the database. The backend functions relate to **Rest API** which uses **JSON** web services to share data.

The system is mainly divided into four parts according to main roles of the conference. The main roles are **Admin**, **Editor**, **Reviewer**, and **User.** The functions of the tool are also based on these main roles of the system.

Functions

1. Reviewer Function – IT 19123578

Reviewers are responsible for reviewing and approving research papers and workshop details. Once research papers and workshop details are uploaded by users, reviewers can view all these papers and review them until they are approved by the reviewer all the papers are in the pending list. If the reviewer approved a research paper or a workshop the user who uploaded the research or the workshop will get a notification with a approved message. All Approved research papers and workshop documents display in separately. If a research gets approved by researcher must pay a fee to present their research at the conference. There is no need to pay a fee for the workshop conductors.

2. User Management – IT 19140476

There are three types of users in the system Attendee, Researcher, Workshop Conductor. All these users must register to the system to attend to this conference. Any guest user can view the system and conference details but, to use the features of the system user must be a registered user. When registering users can selected their user type (Attendee/Researcher/Presenter (Workshop Conductor)) and register. When an Attendee is registering to the system, they must pay an amount of 10$ as the registration fee. When a Researcher is registering to the system, they must upload their research paper with their contact details and when a Presenter (Workshop Conductor) is registering to the system, they must upload a proposal containing all the necessary details about the workshop with their contact details. Researchers and Presenters can upload multiple research papers and workshop proposals after logged in. After registration users can log into the system by providing their username and password. Both research papers and workshop proposal pages uploaded by each user is shown in their dashboards. The tags are used to show the approve status and payment status of research papers and workshop proposals. The Research papers and Proposals which are approved are displayed to all registered users in the system.

3. Editor Function - IT 19028774

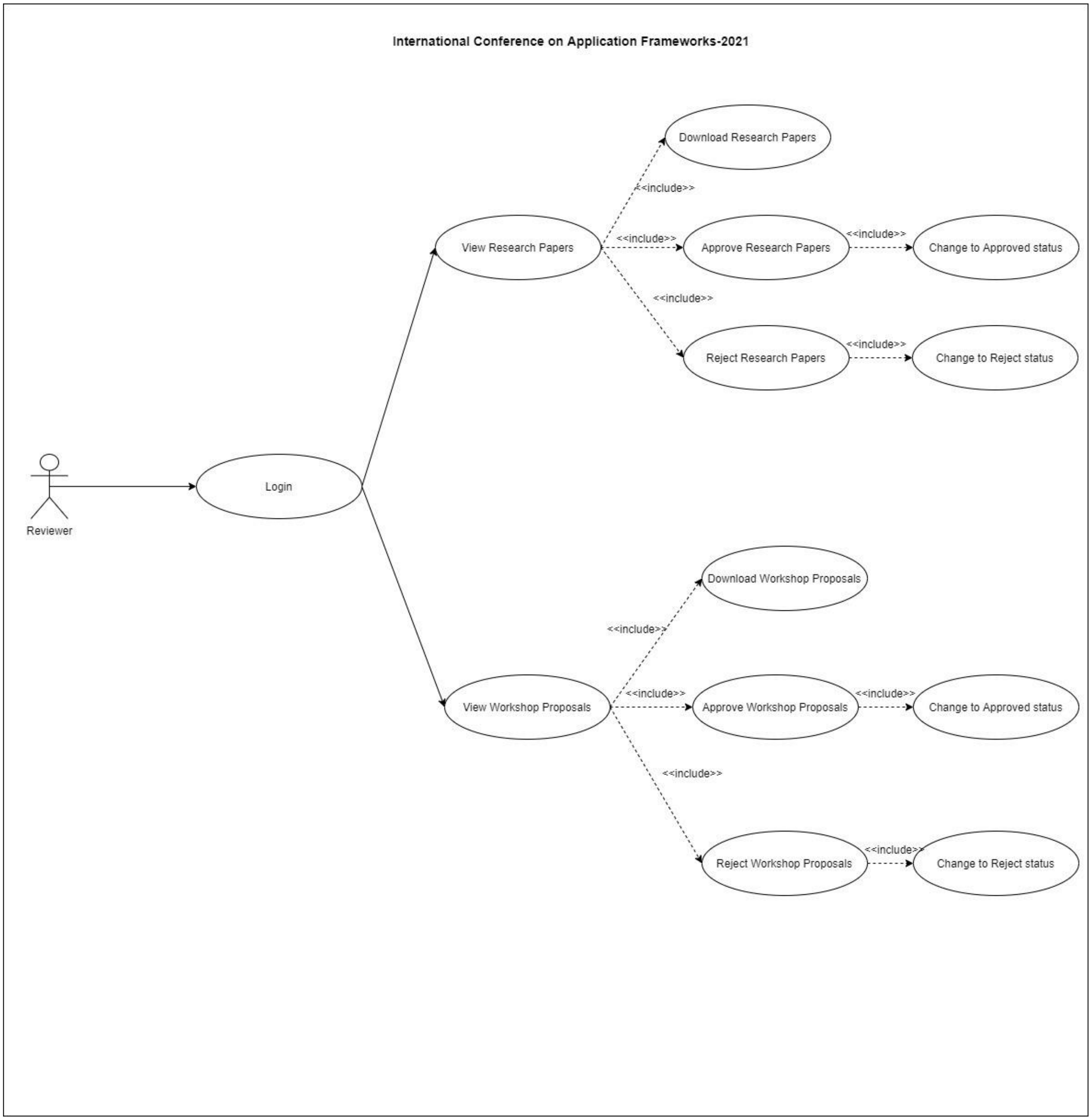
Editor is able to login to editor interface by editor login credentials. Editor uploads conference topic and other conference details like venue, timetables, dates, and events. Editor can submit that information using conference submission forms. After editor submit the form, it will send to admin for approval. If admin approves, the content will be post on the website. If Admin decline the contents, it will send back to editor with error message. Also, Editor can edit conference details posted on the website with the approval of admin. Not only that but also, editor is responsible for upload download templates to the website. So, editor uploads research paper templates and workshop presentation templates using individual, separate interfaces for upload those templates.

4.Admin Management - IT 19143682

All the conference contents can be viewed by the admins, then they can approve or reject those conference contents accordingly. All the approved conference contents will be published only after admins approved. Admins have the privilege to manage all the users, research papers, workshops, and other conference content. Admins have separate login and registration. New admins can register to the admin panel providing necessary information

Use case Diagrams and the Component tree

IT 19123578 - Reviewer Function – Use Case

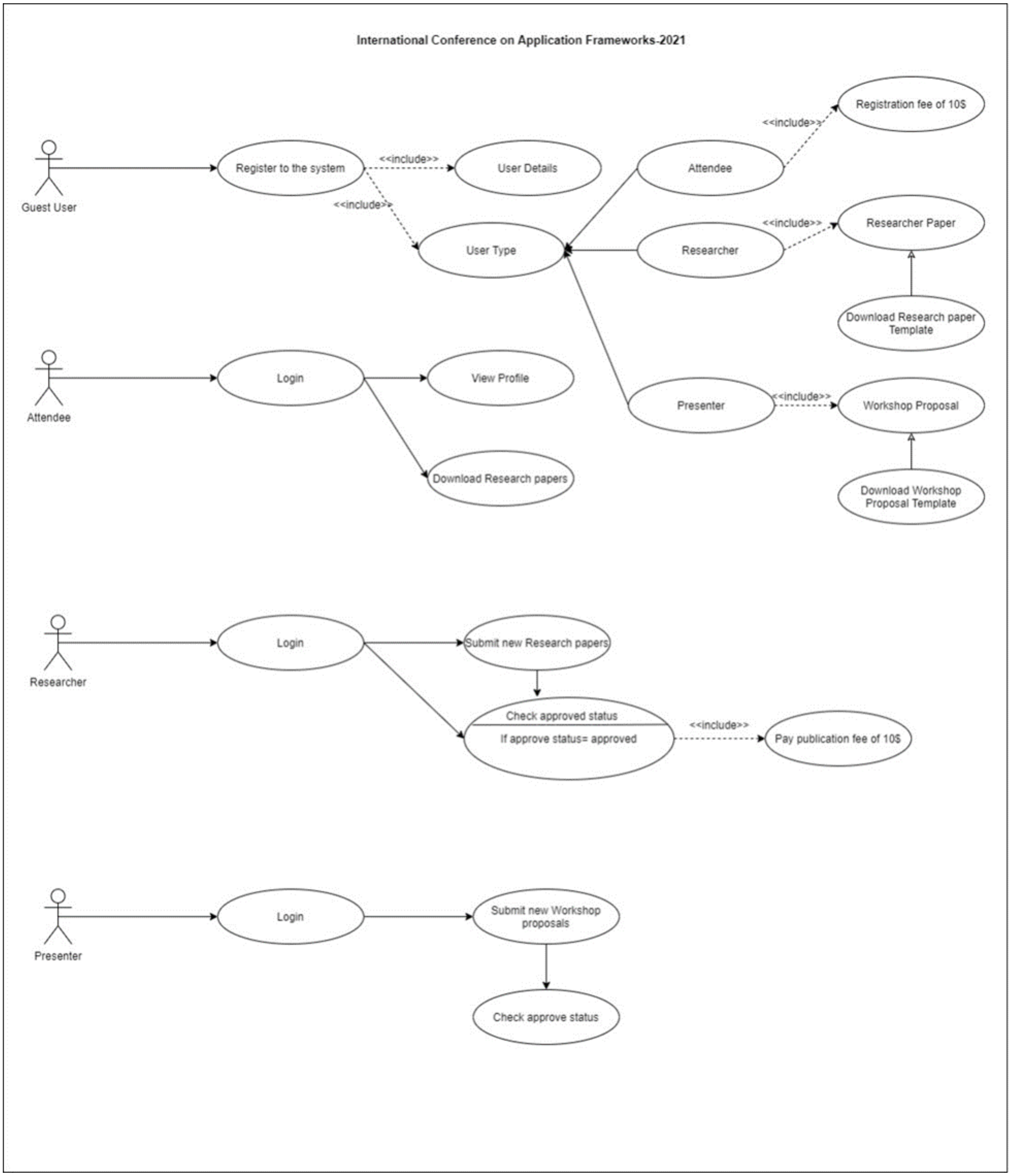


React Component Tree

- App
  - Frontend
    - Common
      - ResearcherCard
      - WorkshopCard
    - Home
      - Title
      - Countdown
    - Download
    - Footer
    - Login
    - Navbar
    - Profile
      - AttendeeProfile
      - PresenterProfile
      - ResearcherProfile
    - Publication
    - Register
      - AttendeeForm
      - ResearcherPresenterForm
    - ResearcherPayment
      - payment
    - ReviewerDashboard
      - Approve
        - ReasearchApproval
        - WorkshopApproval
      - Container
        - ReasearchContainer
        - WorkshopContainer
      - Decline
        - ResearchDecline
        - ResearchDecline
      - New
        - NewResearch
        - NewWorkshop
      - NewWorkshop

    - Workshop

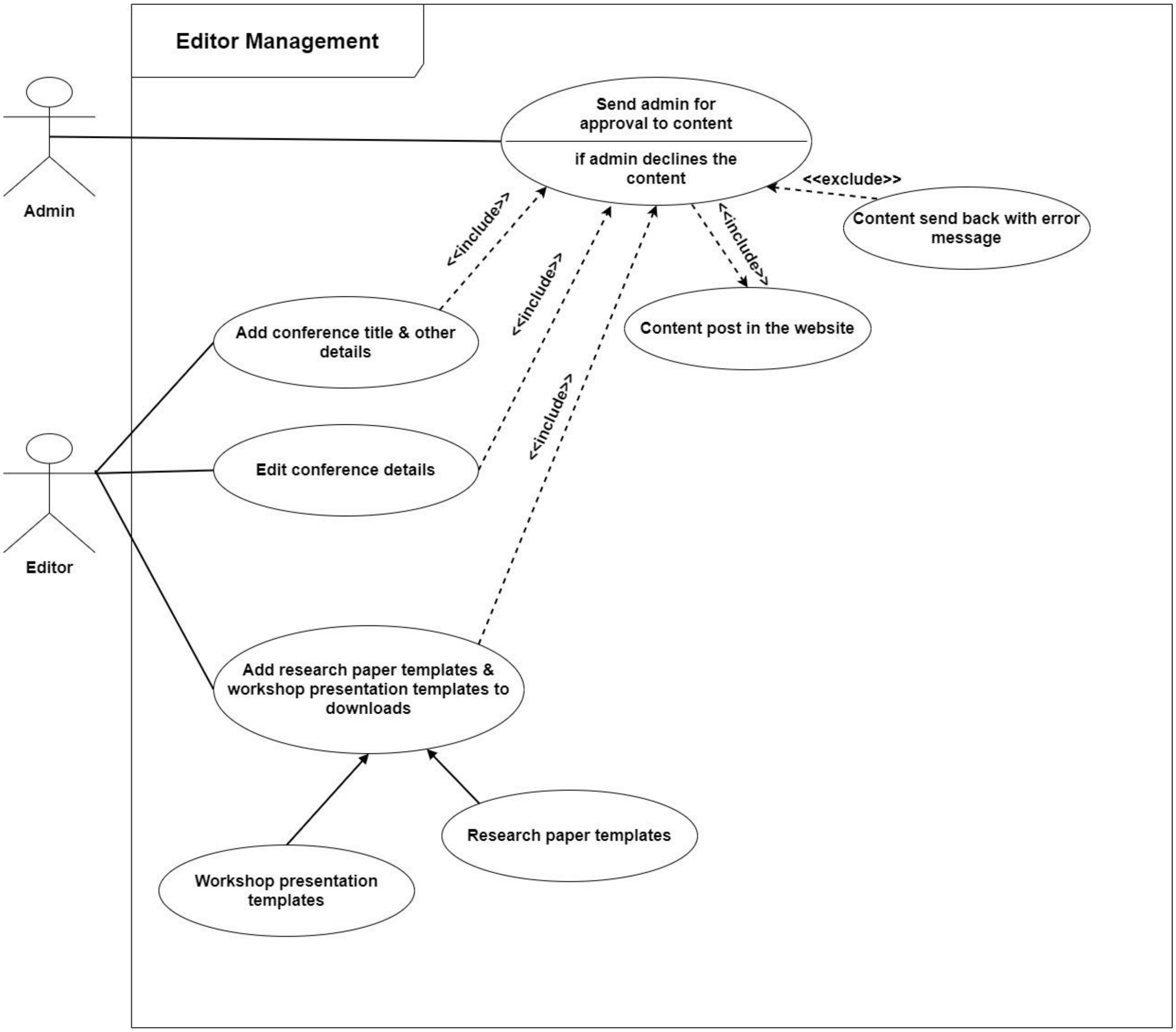
IT 19140476 - User Management – Use Case



React Component Tree

- App
  - Frontend
    - Common
      - ResearcherCard
      - WorkshopCard
    - Home
      - Title
      - Countdown
    - Download
    - Footer
    - Login
    - Navbar
    - Profile
      - AttendeeProfile
      - PresenterProfile
      - ResearcherProfile
    - Publication
    - Register
      - AttendeeForm
      - ResearcherPresenterForm
    - Workshop

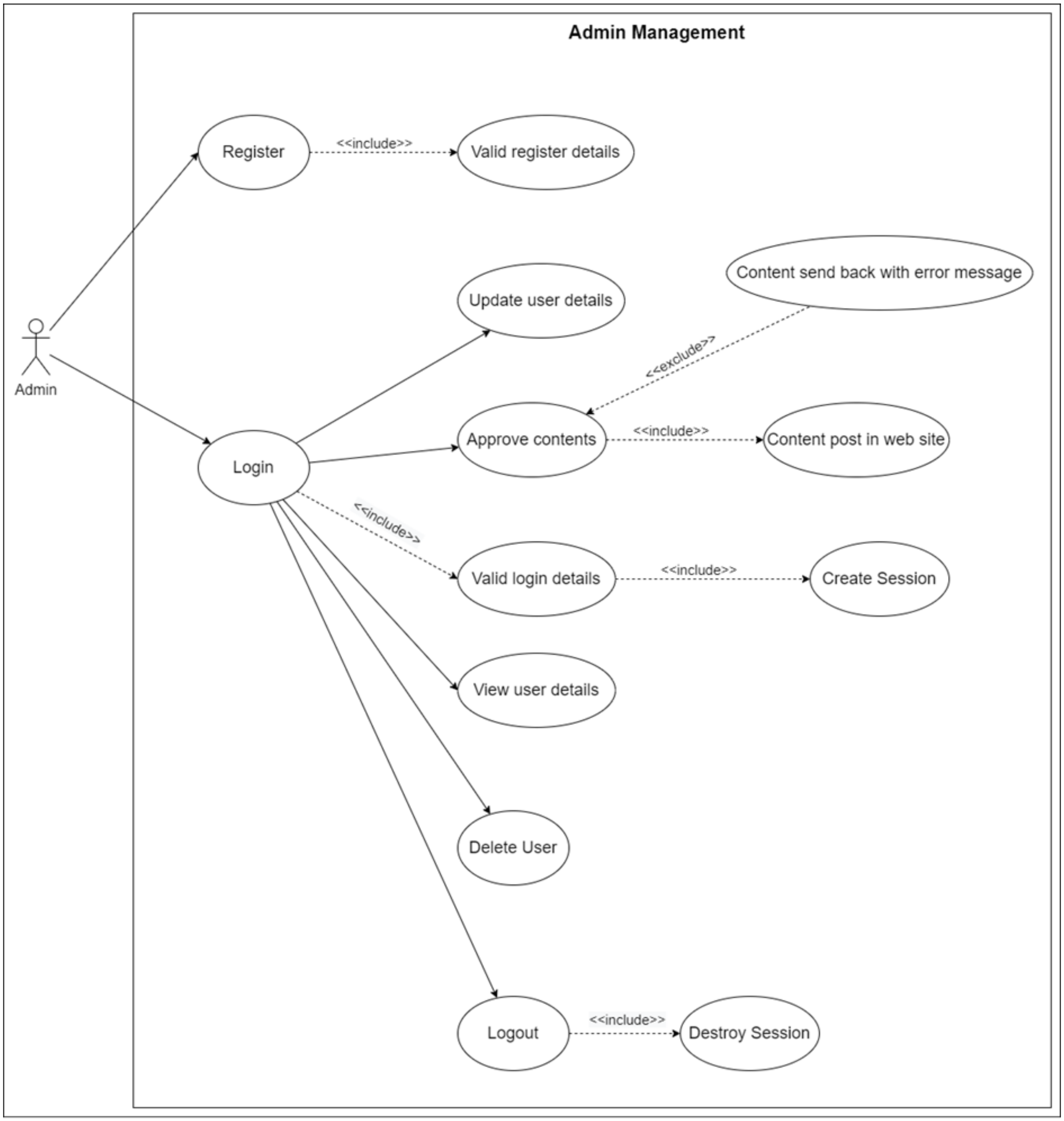
IT 19028774 - Editor Function – Use Case


Editor Management
Admin
Editor
Send admin for approval to content
if admin declines the content
<<exclude>>
Content send back with error message
<<include>>
<<include>>
<<include>>
<<include>>
Content post in the website
Add conference title & other details
Edit conference details
Add research paper templates & workshop presentation templates to downloads
Research paper templates
Workshop presentation templates

React Component Tree

- App
  - Frontend
    - Header
    - Editor Registration
    - Editor Login
    - Editor Dashboard
    - Conferences
      - Insert Conferences
      - View All Conferences
      - Pending Conferences
      - Rejected Conferences
      - Update Conferences
    - Templates
      - Research Paper Templates
        - Upload Research Paper Templates
        - View All Uploaded Research Paper Templates
      - Workshop Presentation Templates
        - Upload Presentation Templates
        - View All Uploaded Presentation Templates
    - Footer

IT 19143682 - Admin Management Use Case



React Component Tree

- App
  - Frontend
    - Header
    - Home
    - Keynotes
    - Submit research papers
    - Submit workshop papers
    - Program
      - Research papers
      - Workshops
    - Pending
      - Workshops
      - Research papers
    - Downloads
      - Research paper templates
      - Workshop PowerPoint Templates
    - Login
    - Registration
    - Dashboard
    - Editor interface
      - Conference Title and Details
    - Footer

Usage of Rest APIs

IT 19123578- Reviewer Function

a) Publication(publication.model)

- Get approved research papers: router.get(“/approved”)
- Get rejected workshops :router.get(“/rejected”)
- Approve research paper :router.patch(“/approve/:id”)
- Reject research paper :router.patch(“/reject/:id”)
- Pending research paper :router.get(“/pending/:id”

b) Workshop(workshop.model)

- Get approved workshops :router.get(“/approved”)
- Get rejected workshops :router.get(“/rejected”)
- Approve workshop :router.patch(“/approve/:id”)
- Reject Workshop :router.patch(“/reject/:id”)
- Pending Workshop :router.get(“/pending/:id”)

c) Researcher (researcher.model)

- Payment: router.post(“/addpayment”)

IT 19140476 -User Function

a) Researcher (researcher.model)

- Save new researcher: router.post("/create")
- Get logged in researcher details: router.get("/my")
- Login : router.post("/login")

b) Attendee(attendee.model)

- Save new attendee: router.post("/create")
- Get logged in attendee details: router.get("/my")
- Login: router.post("/login")
- Payment :router.post("/addpayment")

c) Presenter(presenter.model)

- Save new presenter : router.post("/create")
- Get logged in presenter details: router.get("/my")
- Login:router.post("/login")

d) Workshop(workshop.model)

- Save workshop :router.post("/create")
- My workshops :router.get("/my")
- Pending workshops :router.get("/pending")

e) Publication(publication.model)

- Save research paper :router.post("/create")
- My research papers :router.get("/my")
- Pending research papers :router.get("/pending")

IT 19028774 - Editor Function

a) Conference(conference.model)

- Insert Conference : router.post("/conference/create")
- Get approved conferences : router.get("/conference /approved")
- Get rejected conferences :router.get("/conference /rejected")
- Approve conference :router.patch("/conference/approve/:id")
- Reject conference :router.patch("/conference/reject/:id")
- Pending conference :router.get("/conference/pending/:id")
- Get all conferences :router.get("/conference/")

b) PresentationTemplate(presentationtemplate.model)

- Save presentationtemplate :router.post("/ptemplate/create")
- Get all uploaded presentation templates :router.get("/ptemplate /")

c) ResearchPaperTemplate(researchpapertemplate.model)

- Save researchpapertemplate :router.post("/rpapertemplate /create")
- Get all uploaded presentation templates:router.get("/rpapertemplate/")

IT 19143682 – Admin Management

a) Attendee(attendeeModel)
   - Get all attendees: router.get (“/”)
   - Update attendee: router.put(“/:id”)
   - Delete attendee: router.delete(“/:id”)

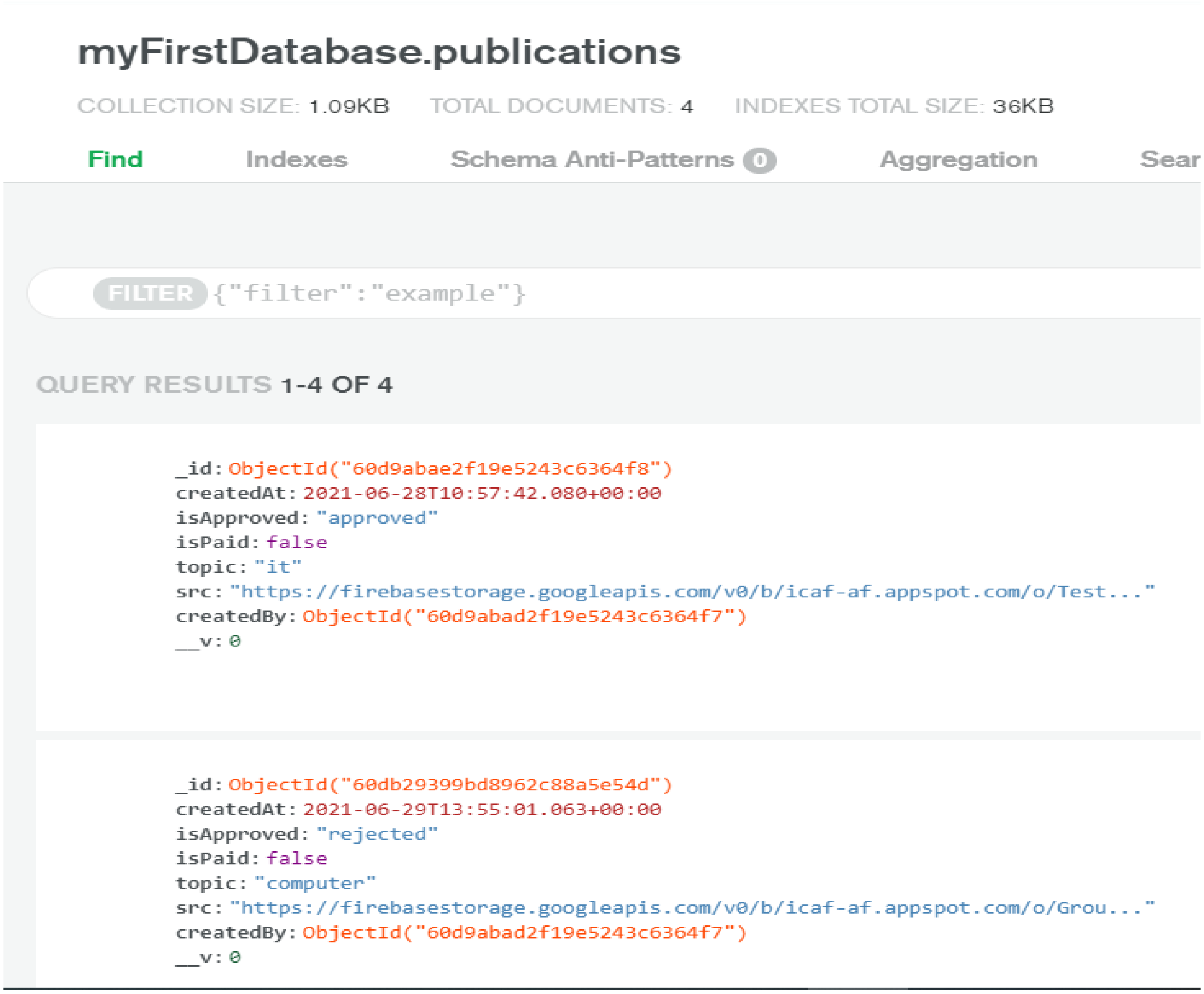
b) Presenter(presenterModel)
   - Get all presenter: router.get (“/”)
   - Update presenter: router.put(“/:id”)
   - Delete presenter: router.delete(“/:id”)

c) Researcher(researcherModel)
   - Get all researcher: router.get (“/”)
   - Update researcher: router.put(“/:id”)
   - Delete researcher: router.delete(“/:id”)

d) Admin(adminModel)
   - Register new admin: router.post (“/register”)
   - Login admin: router.post(“/login”)
   - Logout admin: router.get(“/logout”)
   - Loggedin admin: router.get(“/loggedin”)

Screen Shots of Mongo DB Collections

IT 19123578



myFirstDatabase.publications

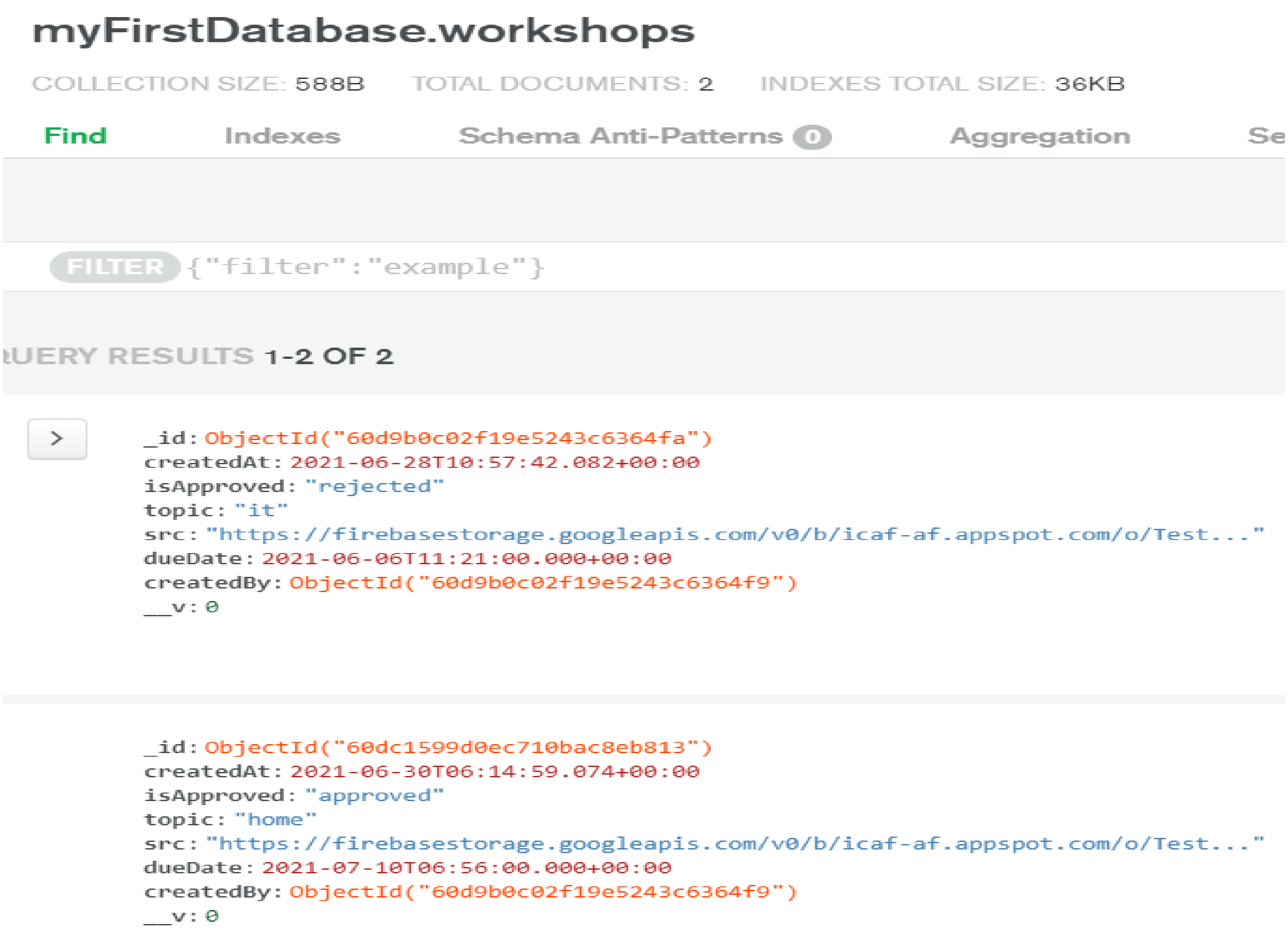
COLLECTION SIZE: 1.09KB   TOTAL DOCUMENTS: 4   INDEXES TOTAL SIZE: 36KB

{"filter":"example"}

QUERY RESULTS 1-4 OF 4

```
_id: ObjectId("60d9abae2f19e5243c6364f8")
createdAt: 2021-06-28T10:57:42.080+00:00
isApproved: "approved"
isPaid: false
topic: "it"
src: "https://firebasestorage.googleapis.com/v0/b/icaf-af.appspot.com/o/Test..."
createdBy: ObjectId("60d9abad2f19e5243c6364f7")
__v: 0
```

```
_id: ObjectId("60db29399bd8962c88a5e54d")
createdAt: 2021-06-29T13:55:01.063+00:00
isApproved: "rejected"
isPaid: false
topic: "computer"
src: "https://firebasestorage.googleapis.com/v0/b/icaf-af.appspot.com/o/Grou..."
createdBy: ObjectId("60d9abad2f19e5243c6364f7")
__v: 0
```



myFirstDatabase.workshops

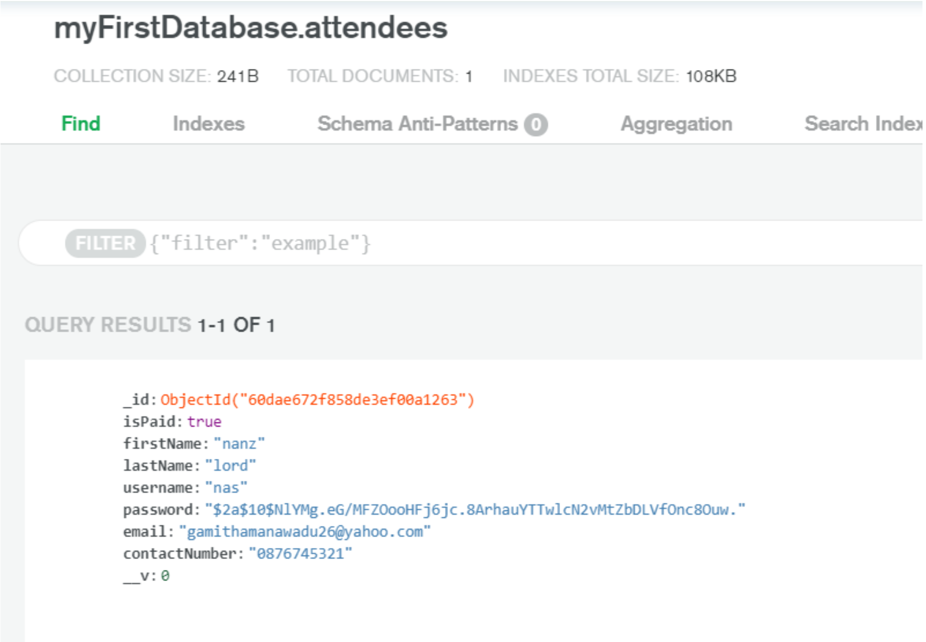
COLLECTION SIZE: 588B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

UERY RESULTS 1-2 OF 2

```
_id: ObjectId("60d9b0c02f19e5243c6364fa")
createdAt: 2021-06-28T10:57:42.082+00:00
isApproved: "rejected"
topic: "it"
src: "https://firebasestorage.googleapis.com/v0/b/icaf-af.appspot.com/o/Test..."
dueDate: 2021-06-06T11:21:00.000+00:00
createdBy: ObjectId("60d9b0c02f19e5243c6364f9")
__v: 0
```

```
_id: ObjectId("60dc1599d0ec710bac8eb813")
createdAt: 2021-06-30T06:14:59.074+00:00
isApproved: "approved"
topic: "home"
src: "https://firebasestorage.googleapis.com/v0/b/icaf-af.appspot.com/o/Test..."
dueDate: 2021-07-10T06:56:00.000+00:00
createdBy: ObjectId("60d9b0c02f19e5243c6364f9")
__v: 0
```

IT 19140476



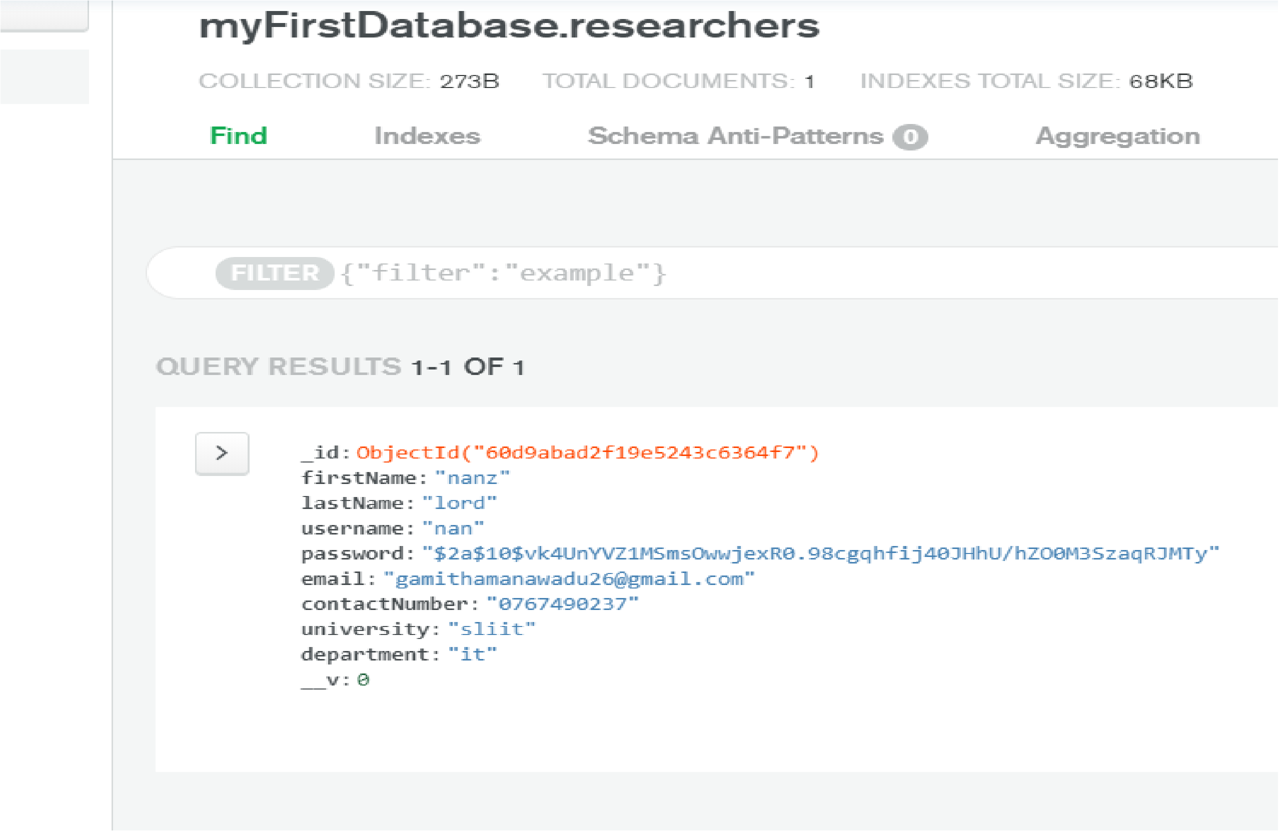
myFirstDatabase.attendees

COLLECTION SIZE: 241B   TOTAL DOCUMENTS: 1   INDEXES TOTAL SIZE: 108KB

{"filter":"example"}

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("60dae672f858de3ef00a1263")
isPaid: true
firstName: "nanz"
lastName: "lord"
username: "nas"
password: "$2a$10$NlYMg.eG/MFZOooHFj6jc.8ArhauYTTwlcN2vMtZbDLVfOnc8Ouw."
email: "gamithamanawadu26@yahoo.com"
contactNumber: "0876745321"
__v: 0
```



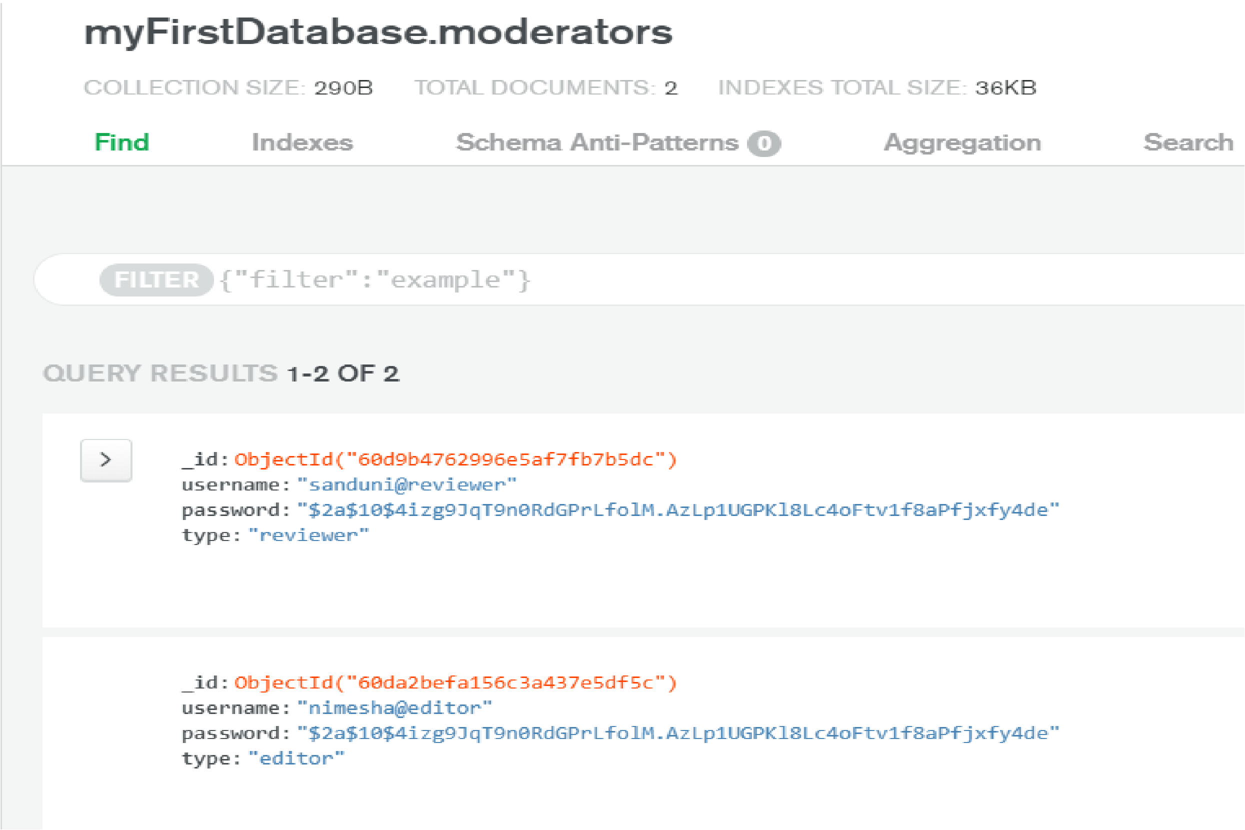
myFirstDatabase.researchers

COLLECTION SIZE: 273B   TOTAL DOCUMENTS: 1   INDEXES TOTAL SIZE: 68KB

{"filter":"example"}

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("60d9abad2f19e5243c6364f7")
firstName: "nanz"
lastName: "lord"
username: "nan"
password: "$2a$10$vk4UnYVZ1MSmsOwwjexR0.98cgqhfij40JHhU/hZO0M3SzaqRJMTy"
email: "gamithamanawadu26@gmail.com"
contactNumber: "0767490237"
university: "sliit"
department: "it"
__v: 0
```



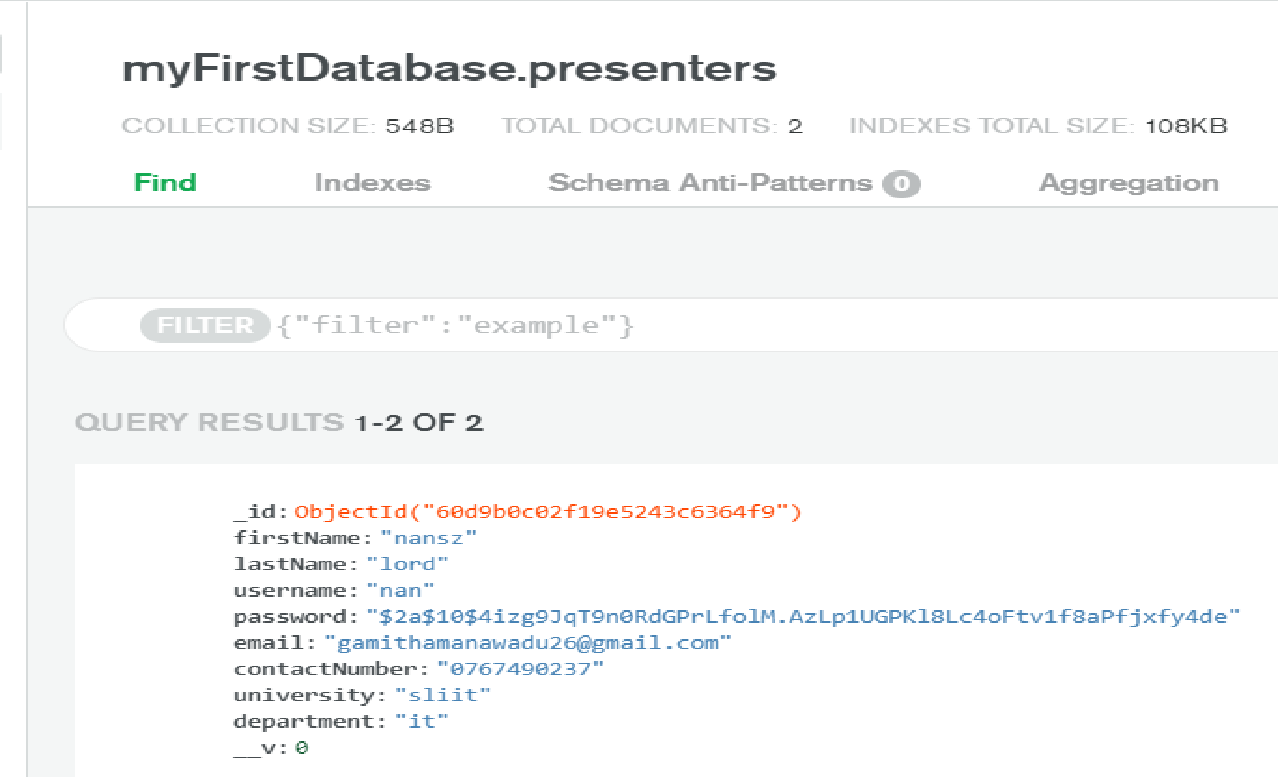
myFirstDatabase.moderators

COLLECTION SIZE: 290B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

QUERY RESULTS 1-2 OF 2

```
_id: ObjectId("60d9b4762996e5af7fb7b5dc")
username: "sanduni@reviewer"
password: "$2a$10$4izg9JqT9n0RdGPrLfolM.AzLp1UGPKl8Lc4oFtv1f8aPfjxfy4de"
type: "reviewer"
```

```
_id: ObjectId("60da2befa156c3a437e5df5c")
username: "nimesha@editor"
password: "$2a$10$4izg9JqT9n0RdGPrLfolM.AzLp1UGPKl8Lc4oFtv1f8aPfjxfy4de"
type: "editor"
```



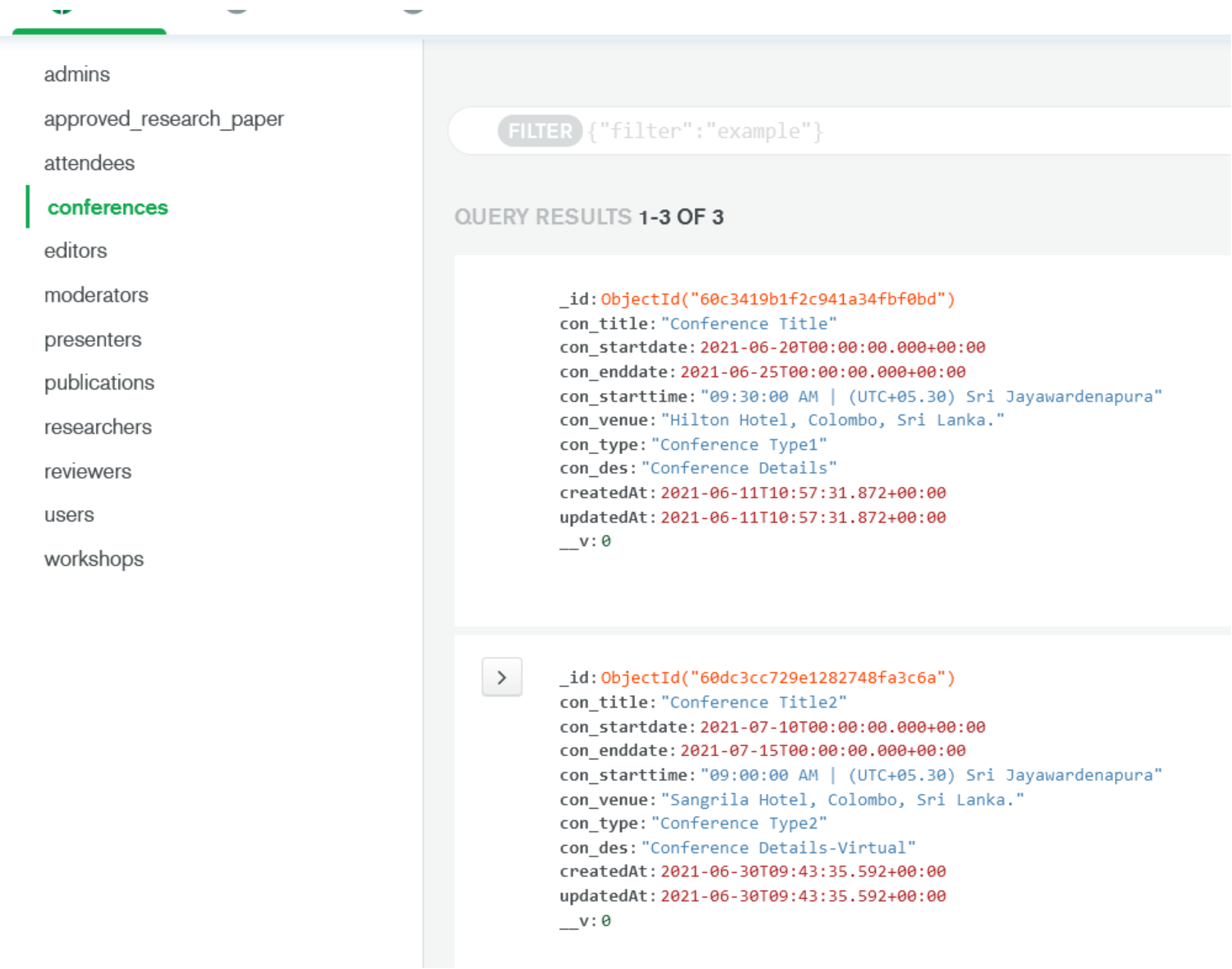
myFirstDatabase.presenters

COLLECTION SIZE: 548B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 108KB

QUERY RESULTS 1-2 OF 2

```
_id: ObjectId("60d9b0c02f19e5243c6364f9")
firstName: "nansz"
lastName: "lord"
username: "nan"
password: "$2a$10$4izg9JqT9n0RdGPrLfolM.AzLp1UGPKl8Lc4oFtv1f8aPfjxfy4de"
email: "gamithamanawadu26@gmail.com"
contactNumber: "0767490237"
university: "sliit"
department: "it"
__v: 0
```

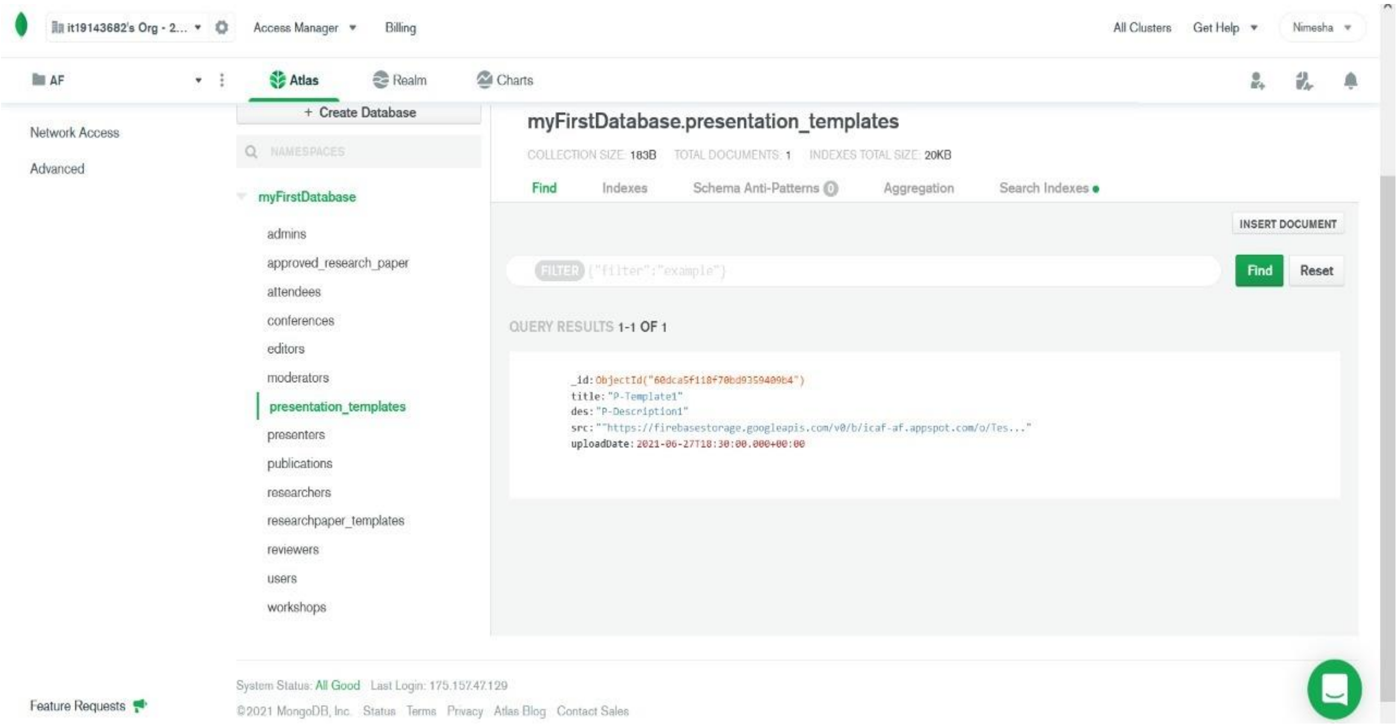
IT 19028774



QUERY RESULTS 1-3 OF 3

```
_id: ObjectId("60c3419b1f2c941a34fbf0bd")
con_title: "Conference Title"
con_startdate: 2021-06-20T00:00:00.000+00:00
con_enddate: 2021-06-25T00:00:00.000+00:00
con_starttime: "09:30:00 AM | (UTC+05.30) Sri Jayawardenapura"
con_venue: "Hilton Hotel, Colombo, Sri Lanka."
con_type: "Conference Type1"
con_des: "Conference Details"
createdAt: 2021-06-11T10:57:31.872+00:00
updatedAt: 2021-06-11T10:57:31.872+00:00
__v: 0
```

```
_id: ObjectId("60dc3cc729e1282748fa3c6a")
con_title: "Conference Title2"
con_startdate: 2021-07-10T00:00:00.000+00:00
con_enddate: 2021-07-15T00:00:00.000+00:00
con_starttime: "09:00:00 AM | (UTC+05.30) Sri Jayawardenapura"
con_venue: "Sangrila Hotel, Colombo, Sri Lanka."
con_type: "Conference Type2"
con_des: "Conference Details-Virtual"
createdAt: 2021-06-30T09:43:35.592+00:00
updatedAt: 2021-06-30T09:43:35.592+00:00
__v: 0
```



myFirstDatabase.presentation_templates

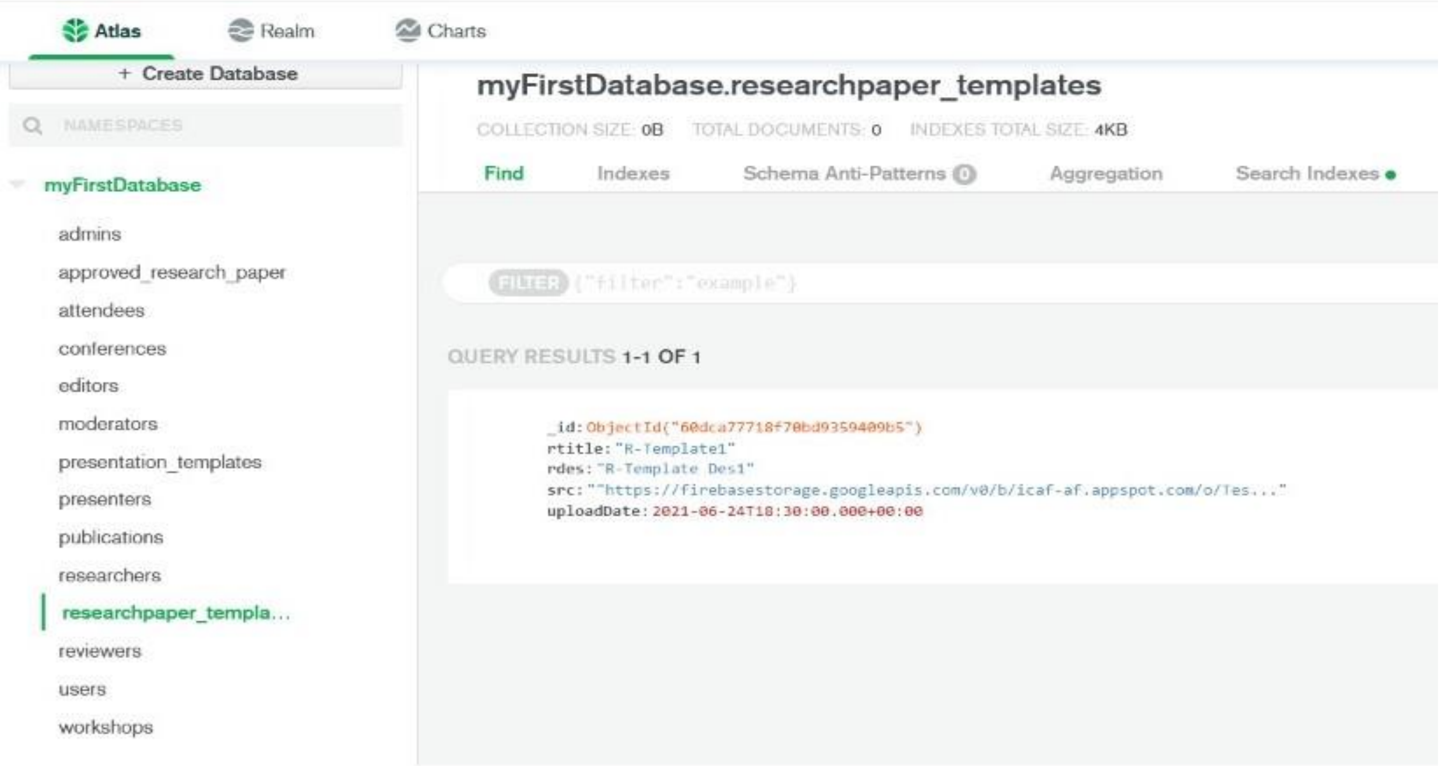
COLLECTION SIZE: 183B   TOTAL DOCUMENTS: 1   INDEXES TOTAL SIZE: 20KB

QUERY RESULTS 1-1 OF 1

```
_id:ObjectId("60dca5f118f70bd9359409b4")
title:"P-Template1"
des:"P-Description1"
src:""https://firebasestorage.googleapis.com/v0/b/icaf-af.appspot.com/o/Tes..."
uploadDate:2021-06-27T18:30:00.000+00:00
```

System Status: All Good   Last Login: 175.157.47.129

©2021 MongoDB, Inc.   Status   Terms   Privacy   Atlas Blog   Contact Sales



myFirstDatabase.researchpaper_templates

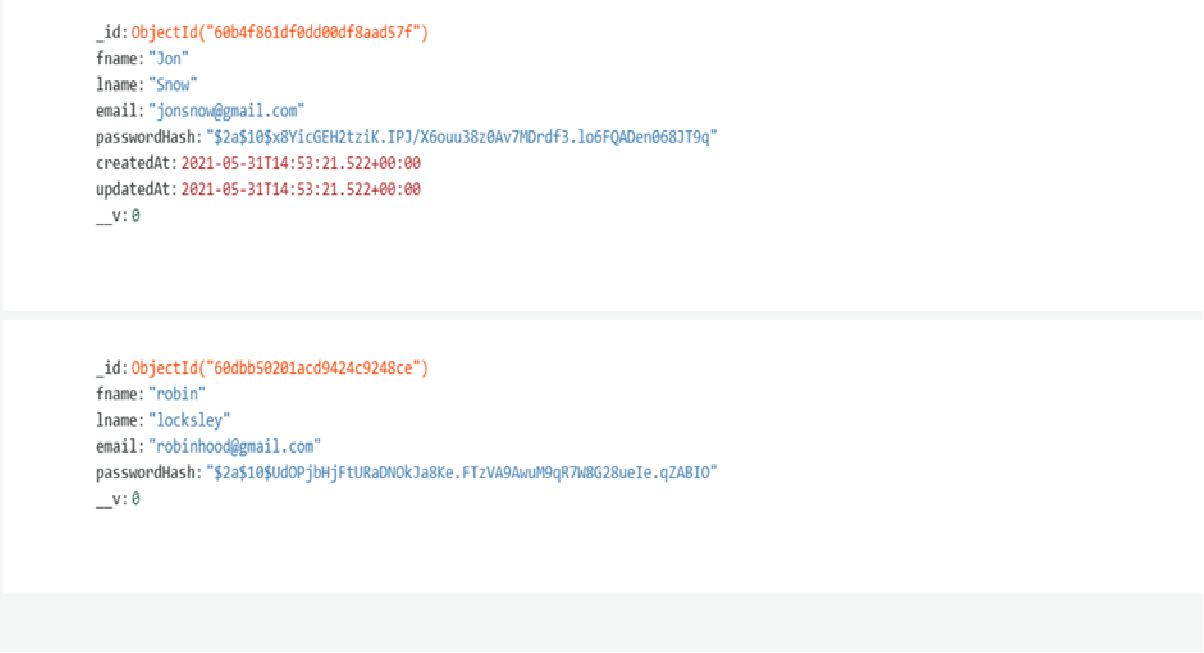
COLLECTION SIZE: 0B   TOTAL DOCUMENTS: 0   INDEXES TOTAL SIZE: 4KB

QUERY RESULTS 1-1 OF 1

```
_id:ObjectId("60dca77718f70bd9359409b5")
rtitle:"R-Template1"
rdes:"R-Template Des1"
src:""https://firebasestorage.googleapis.com/v0/b/icaf-af.appspot.com/o/Tes..."
uploadDate:2021-06-24T18:30:00.000+00:00
```

IT 19143682

Admin Table

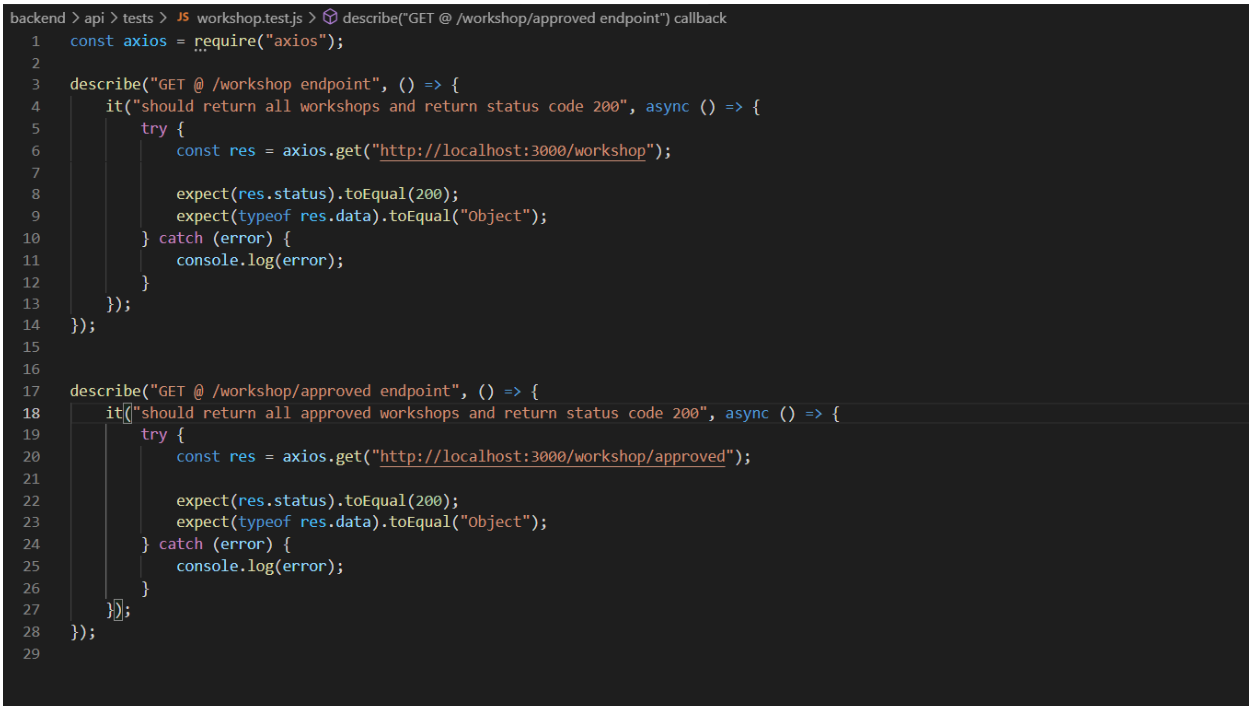


```
_id:ObjectId("60b4f861df0dd00df8aad57f")
fname:"Jon"
lname:"Snow"
email:"jonsnow@gmail.com"
passwordHash:"$2a$10$x8YicGEH2tziK.IPJ/X6ouu38z0Av7MDrdf3.lo6FQADen068JT9q"
createdAt:2021-05-31T14:53:21.522+00:00
updatedAt:2021-05-31T14:53:21.522+00:00
__v:0
```

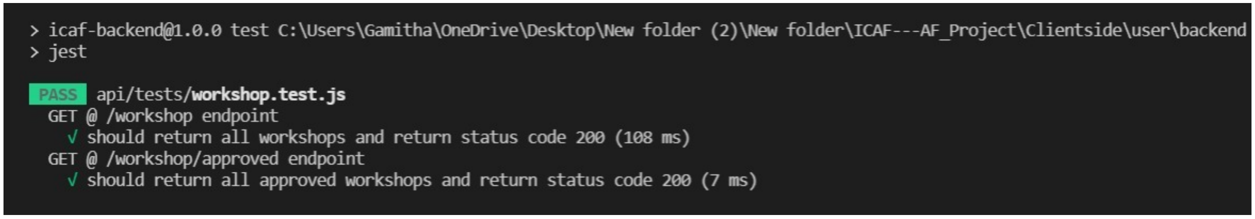
```
_id:ObjectId("60dbb50201acd9424c9248ce")
fname:"robin"
lname:"locksley"
email:"robinhood@gmail.com"
passwordHash:"$2a$10$UdOPjbHjFtURaDNOkJa8Ke.FTzVA9AwuM9qR7W8G28ueIe.qZABIO"
__v:0
```

Test Cases

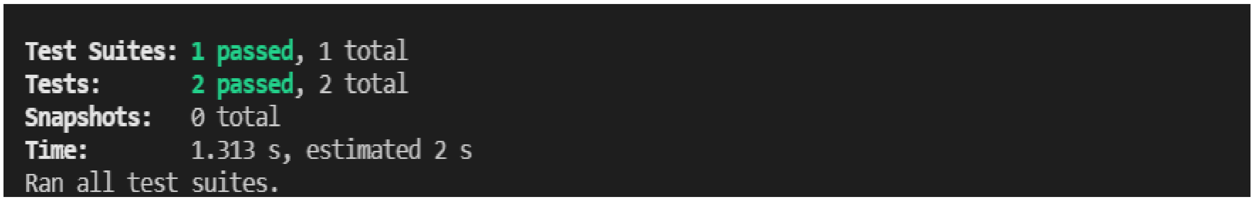
IT 19123578 -Test Cases for Workshop Approval


backend > api > tests > JS workshop.test.js > describe("GET @ /workshop/approved endpoint") callback

```javascript
const axios = require("axios");

describe("GET @ /workshop endpoint", () => {
    it("should return all workshops and return status code 200", async () => {
        try {
            const res = axios.get("http://localhost:3000/workshop");

            expect(res.status).toEqual(200);
            expect(typeof res.data).toEqual("Object");
        } catch (error) {
            console.log(error);
        }
    });
});


describe("GET @ /workshop/approved endpoint", () => {
    it("should return all approved workshops and return status code 200", async () => {
        try {
            const res = axios.get("http://localhost:3000/workshop/approved");

            expect(res.status).toEqual(200);
            expect(typeof res.data).toEqual("Object");
        } catch (error) {
            console.log(error);
        }
    });
});

```

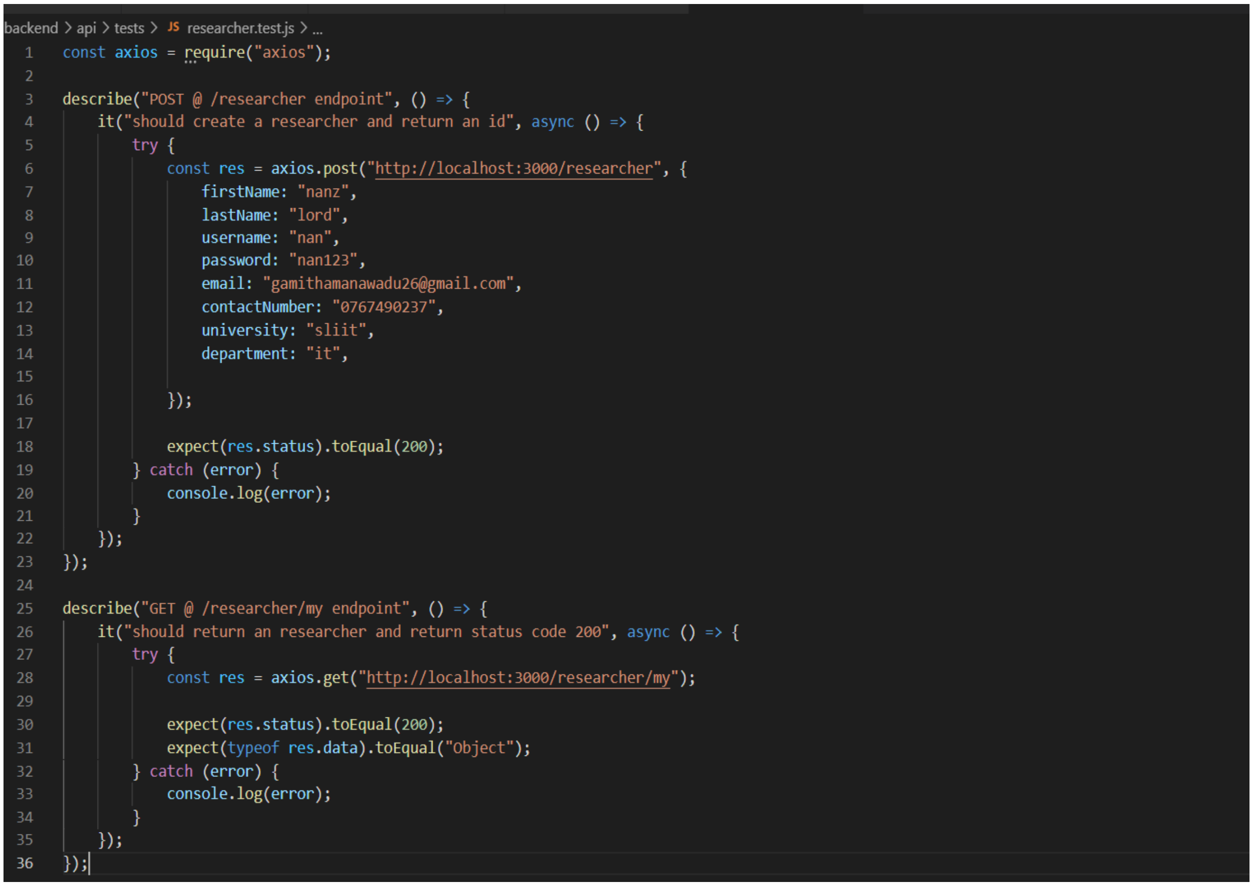


```
> icaf-backend@1.0.0 test C:\Users\Gamitha\OneDrive\Desktop\New folder (2)\New folder\ICAF---AF_Project\Clientside\user\backend
> jest

 PASS  api/tests/workshop.test.js
  GET @ /workshop endpoint
    √ should return all workshops and return status code 200 (108 ms)
  GET @ /workshop/approved endpoint
    √ should return all approved workshops and return status code 200 (7 ms)
```

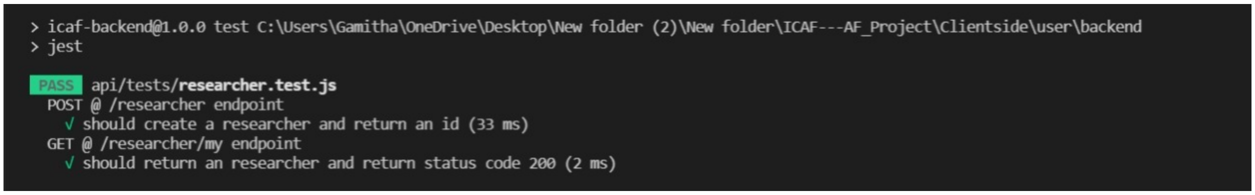


```
Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.313 s, estimated 2 s
Ran all test suites.
```

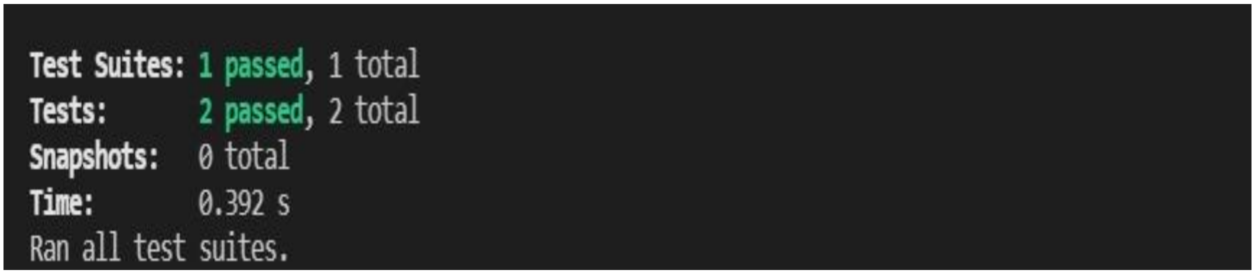
IT 19140476 :-Test Cases for User Login -Researcher



```
const axios = require("axios");

describe("POST @ /researcher endpoint", () => {
    it("should create a researcher and return an id", async () => {
        try {
            const res = axios.post("http://localhost:3000/researcher", {
                firstName: "nanz",
                lastName: "lord",
                username: "nan",
                password: "nan123",
                email: "gamithamanawadu26@gmail.com",
                contactNumber: "0767490237",
                university: "sliit",
                department: "it",

            });

            expect(res.status).toEqual(200);
        } catch (error) {
            console.log(error);
        }
    });
});

describe("GET @ /researcher/my endpoint", () => {
    it("should return an researcher and return status code 200", async () => {
        try {
            const res = axios.get("http://localhost:3000/researcher/my");

            expect(res.status).toEqual(200);
            expect(typeof res.data).toEqual("Object");
        } catch (error) {
            console.log(error);
        }
    });
});
```

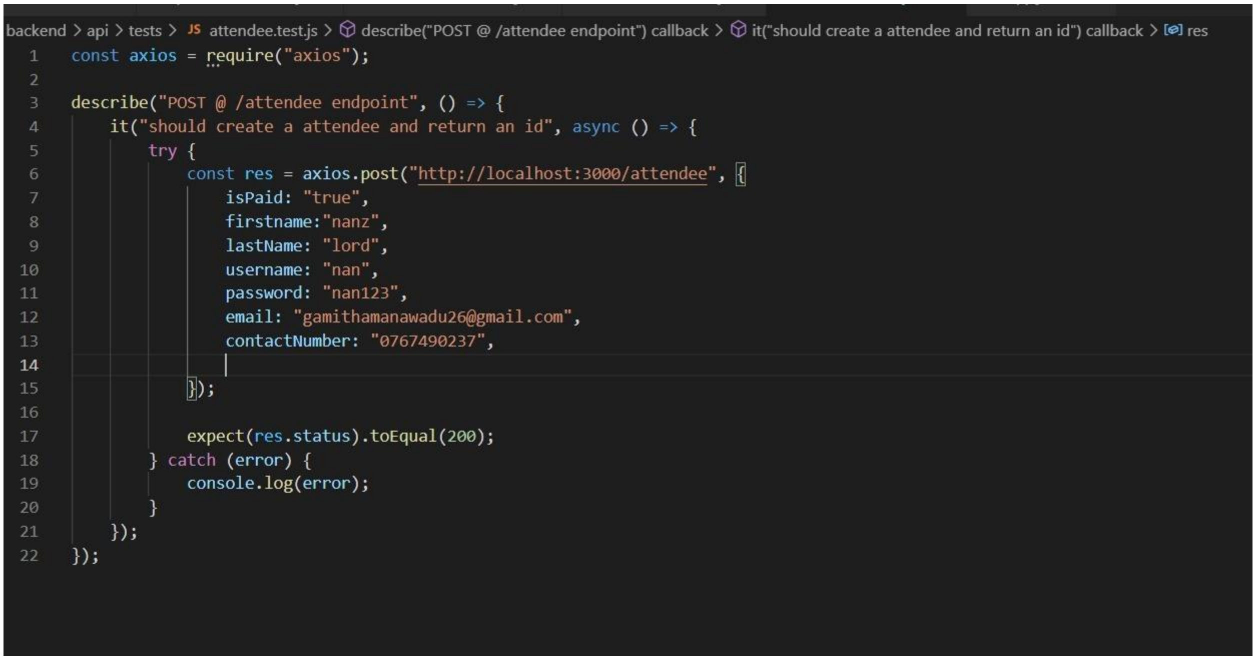


```
> icaf-backend@1.0.0 test C:\Users\Gamitha\OneDrive\Desktop\New folder (2)\New folder\ICAF---AF_Project\Clientside\user\backend
> jest

 PASS  api/tests/researcher.test.js
  POST @ /researcher endpoint
    √ should create a researcher and return an id (33 ms)
  GET @ /researcher/my endpoint
    √ should return an researcher and return status code 200 (2 ms)
```



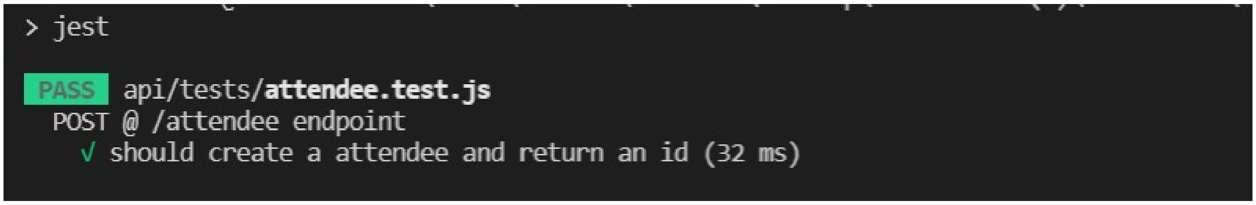
```
Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        0.392 s
Ran all test suites.
```

IT 19028774 :-Test Cases For User Login Attendee

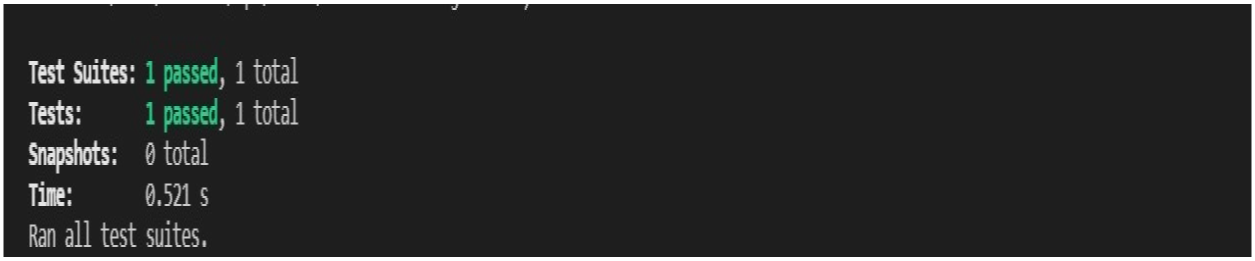


backend > api > tests > JS attendee.test.js > describe("POST @ /attendee endpoint") callback > it("should create a attendee and return an id") callback > res

```
const axios = require("axios");

describe("POST @ /attendee endpoint", () => {
    it("should create a attendee and return an id", async () => {
        try {
            const res = axios.post("http://localhost:3000/attendee", {
                isPaid: "true",
                firstname:"nanz",
                lastName: "lord",
                username: "nan",
                password: "nan123",
                email: "gamithamanawadu26@gmail.com",
                contactNumber: "0767490237",
                
            });

            expect(res.status).toEqual(200);
        } catch (error) {
            console.log(error);
        }
    });
});
```

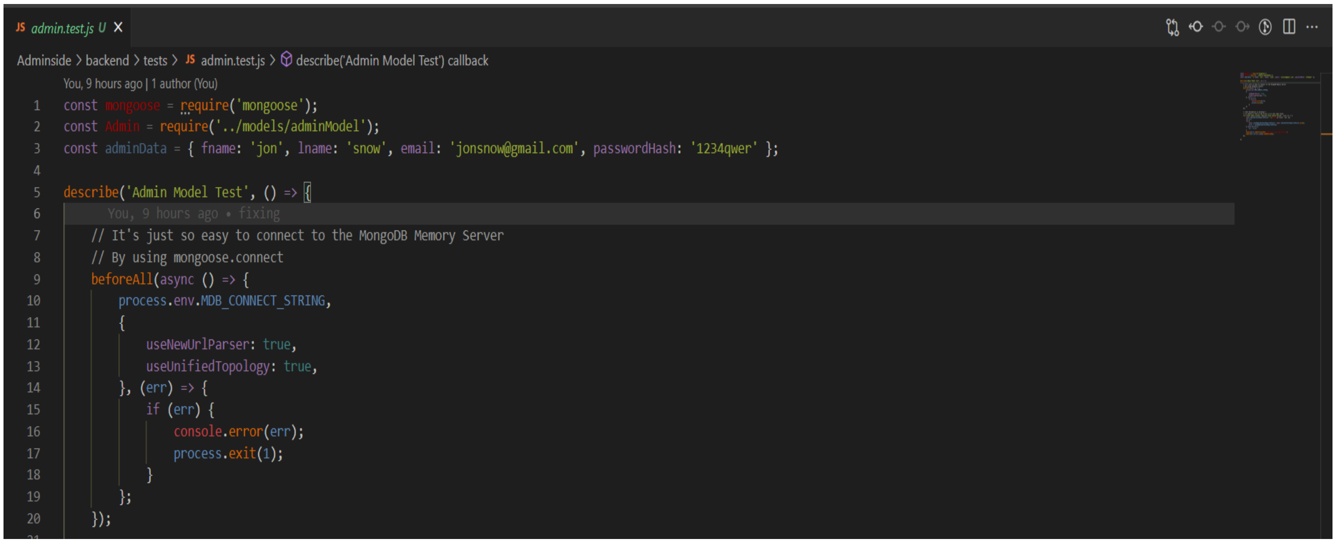


```
> jest

PASS  api/tests/attendee.test.js
  POST @ /attendee endpoint
    √ should create a attendee and return an id (32 ms)
```



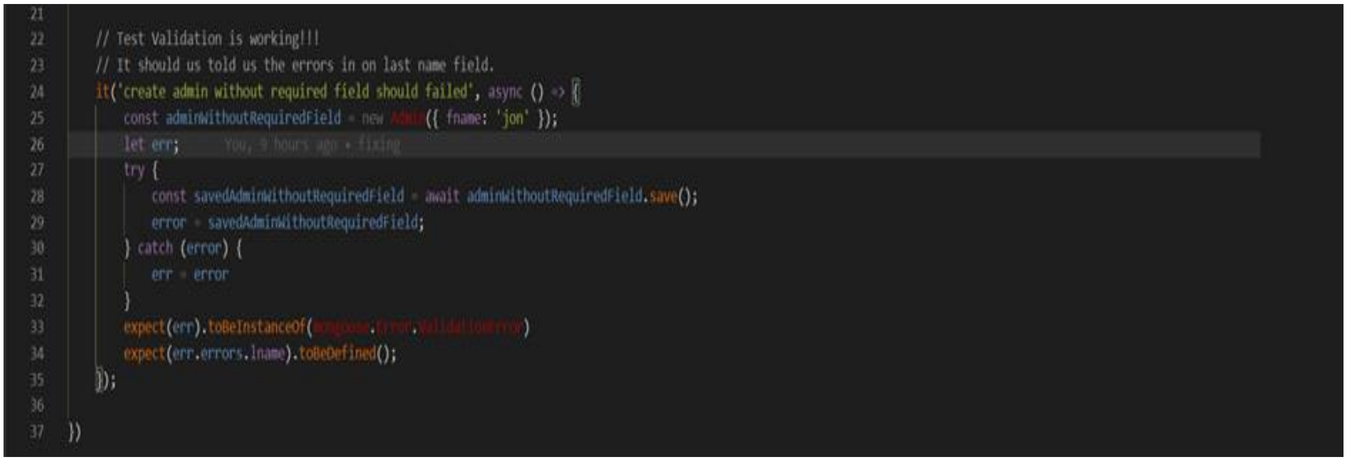
```
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.521 s
Ran all test suites.
```

IT 19143682 – Test Case for Admin Login

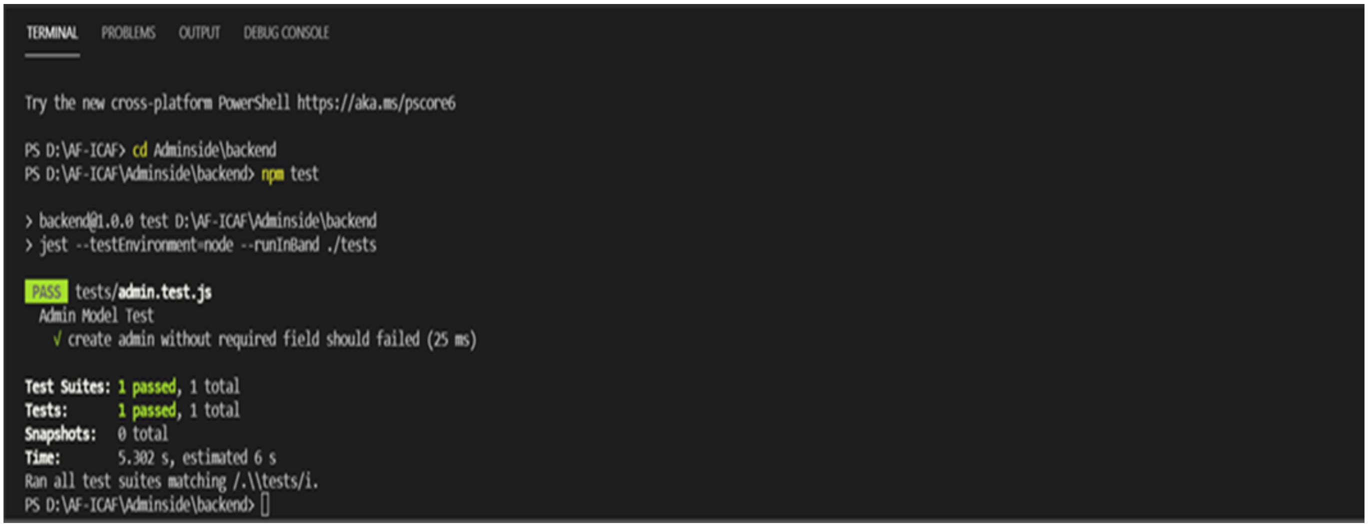


Adminside > backend > tests > JS admin.test.js > describe('Admin Model Test') callback
You, 9 hours ago | 1 author (You)

```
const mongoose = require('mongoose');
const Admin = require('../models/adminModel');
const adminData = { fname: 'jon', lname: 'snow', email: 'jonsnow@gmail.com', passwordHash: '1234qwer' };

describe('Admin Model Test', () => {
      You, 9 hours ago • fixing
    // It's just so easy to connect to the MongoDB Memory Server
    // By using mongoose.connect
    beforeAll(async () => {
        process.env.MDB_CONNECT_STRING,
        {
            useNewUrlParser: true,
            useUnifiedTopology: true,
        }, (err) => {
            if (err) {
                console.error(err);
                process.exit(1);
            }
        };
    });
```



```

    // Test Validation is working!!!
    // It should us told us the errors in on last name field.
    it('create admin without required field should failed', async () => {
        const adminWithoutRequiredField = new Admin({ fname: 'jon' });
        let err;      You, 9 hours ago • fixing
        try {
            const savedAdminWithoutRequiredField = await adminWithoutRequiredField.save();
            error = savedAdminWithoutRequiredField;
        } catch (error) {
            err = error
        }
        expect(err).toBeInstanceOf(mongoose.Error.ValidationError)
        expect(err.errors.lname).toBeDefined();
    });

})
```



```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\AF-ICAF> cd Adminside\backend
PS D:\AF-ICAF\Adminside\backend> npm test

> backend@1.0.0 test D:\AF-ICAF\Adminside\backend
> jest --testEnvironment=node --runInBand ./tests

 PASS  tests/admin.test.js
  Admin Model Test
    √ create admin without required field should failed (25 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        5.302 s, estimated 6 s
Ran all test suites matching /.\\tests/i.
PS D:\AF-ICAF\Adminside\backend> 
```