BSc (Hons) in Information Technology

Specializing in Software Engineering

Year 3 - 2021

SE3040 – Application Frameworks

Group Project

Technical Report Draft

Group Name - **CreativeAF**

Group ID - **2021S1_REG_WE_38**

Student ID	Name
IT19216874	R. Y. Senevirathne
IT19217796	M.A.A.H.Ratnayake
IT19362854	S.L.Abeygunawardana
IT19105826	M.N.D.Meegoda

Repository Link: https://github.com/nilanmeegoda99/AF-project-3rdyr-1st-sem.git

---For Evaluators' use only---

Evaluator's Name:

Comments:

Description

The conference management tool for the International Conference on Application Frameworks (ICAF) is a web application developed for various types of users to submit their research papers, presentations, workshop materials, edit or approve those submissions, edit conference details, manage users or make payments. Likewise, this application consists of a large scale of functionalities for different user types. The conference will be held at SLIIT premises physically and only the participants who have registered to the system can participate in it. At this conference, researchers will present their latest experiments and related stuff on different programming languages and technology stacks.

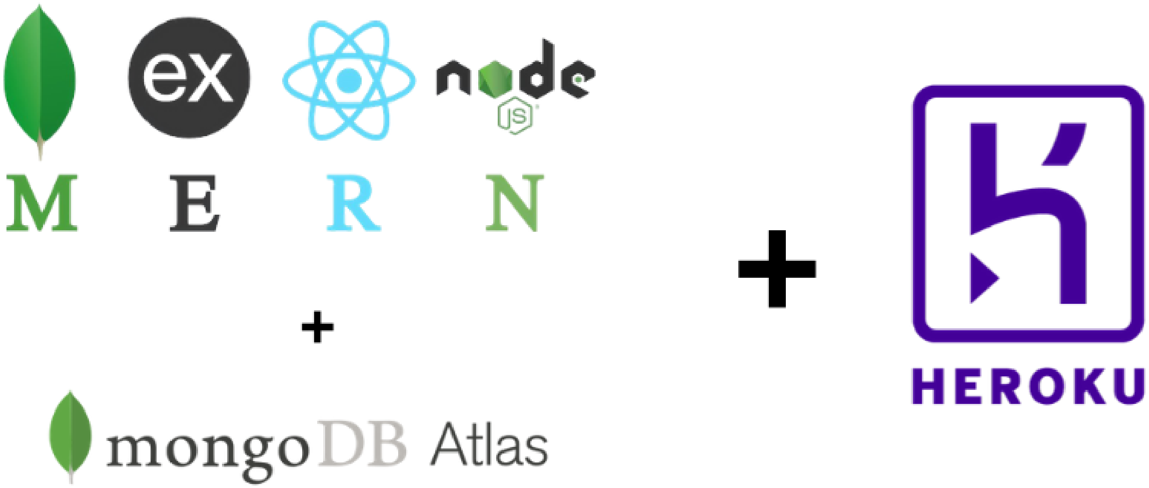
The system mainly consists of four user types as admin, editor, reviewer, and user. When the user registers to the system, he or she has three options to select to register, as a researcher, workshop conductor, or attendee. The attendee must pay a fee to attend the conference. The reviewer can view the submitted research papers and other workshop conducting materials and he can approve or decline those submissions. If the researcher's papers got approved, the relevant researcher must pay a fee for those papers to be presented at the conference. But the workshop conductors don’t need to pay any fee after their submission gets approved by the reviewer.

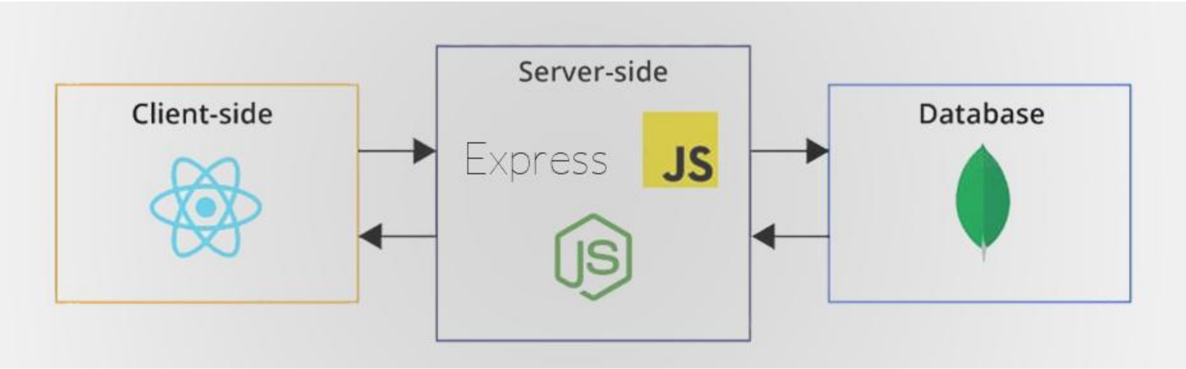
The editor can add or edit conference details such as agenda, speakers, topics, and other information on the conference. and the admin should approve them for those to be previewed on the website. When it comes to admin privileges, the admin can view all these activities through a dashboard. In this dashboard, he can check the payments done by attendees and the researchers. Other submissions such as research papers and workshop conducting materials can also be viewed by the admin. Admin can also add, edit or remove users who have registered to the system.

This web application is a single-page application and we have used the MERN technology stack for the development and the application will be deployed to the Heroku cloud platform. For the payment process, we will be using PayHere payment gateway which is a locally developed gateway. All the technologies will be briefly described in the next section under the technology stack.

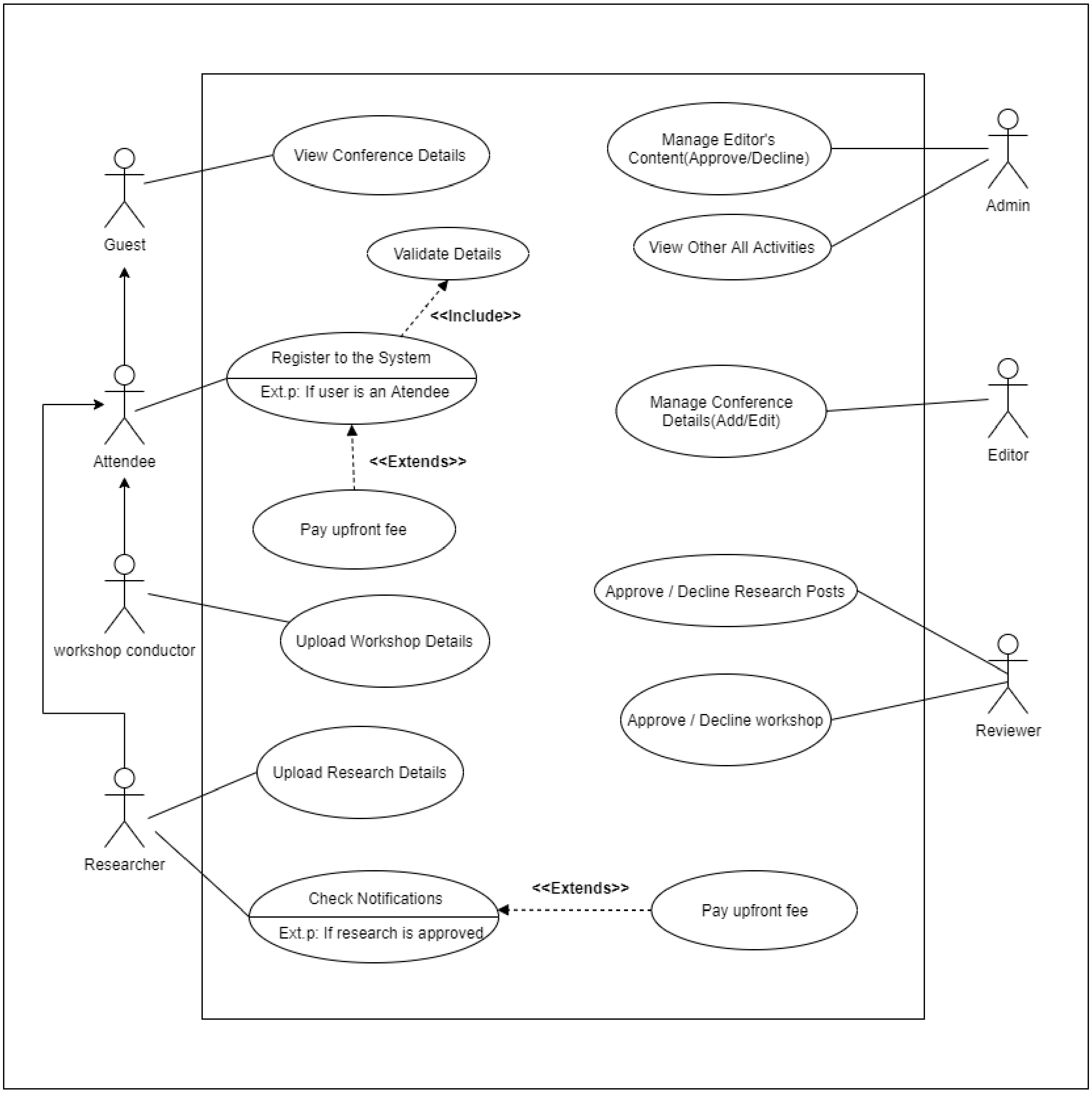
Technology Stack

The application is developed using the MERN technology stack, Under MERN stack we have used MongoDB which is a No-SQL database to store data and it is deployed in a cloud service called Atlas by Mongo company. Likewise for the frontend we have used React js which is a javascript library from the Facebook team and for the backend, we have used express framework and node js server. To ensure that sensitive data like database credentials, cloud credentials and other keys are secured, we have used a npm dependency module named dotenv module. Likewise we have used axios, redux dependencies for the development.For the backend development, we have used representational state transfer software architectural style(REST Api).

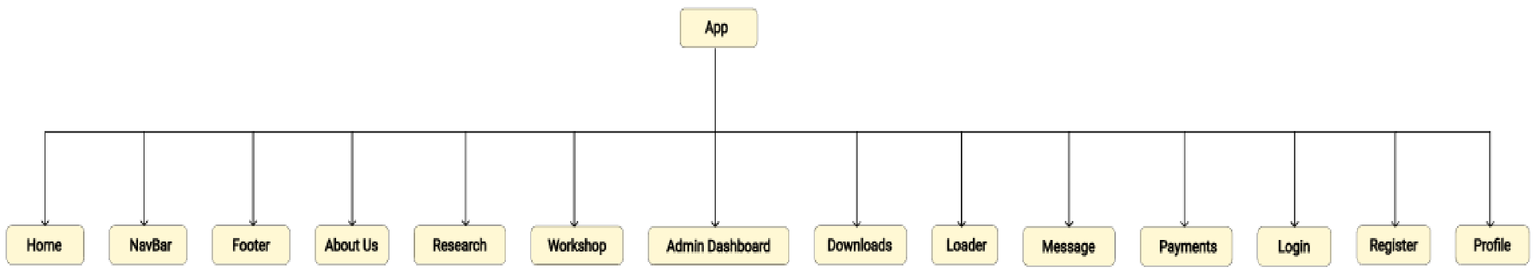




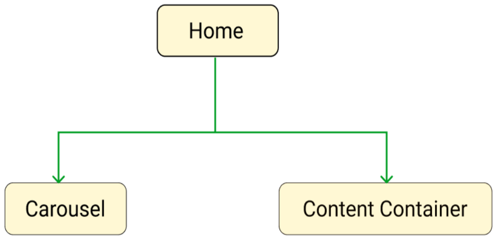
Use Case diagram



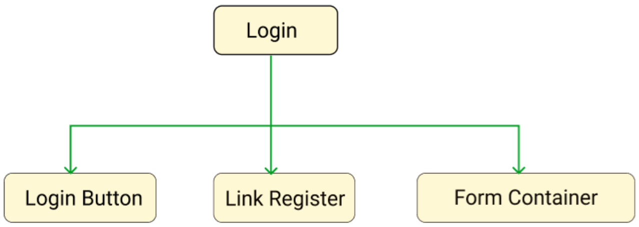
React Component Tree



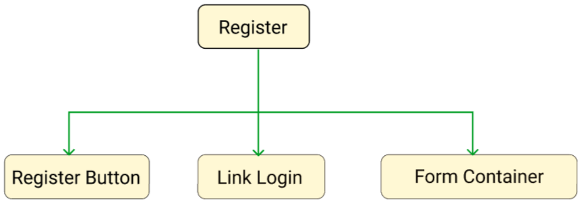
Home Component



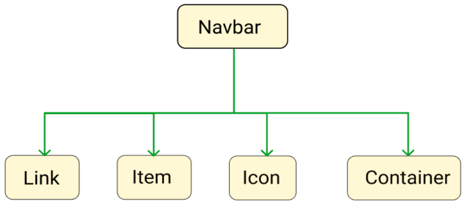
Login Component



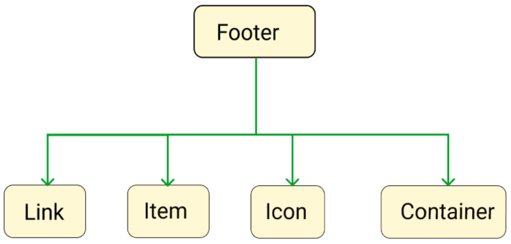
Register Component



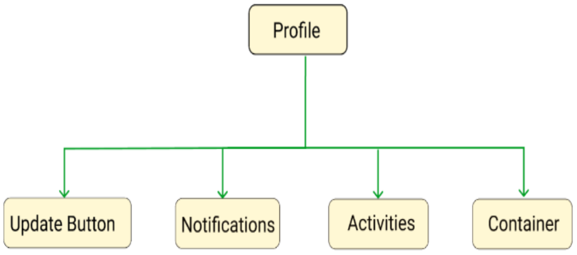
NavBar Component



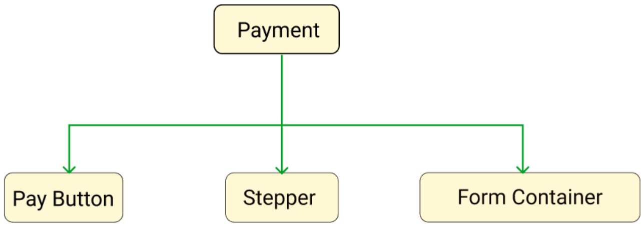
Footer Component



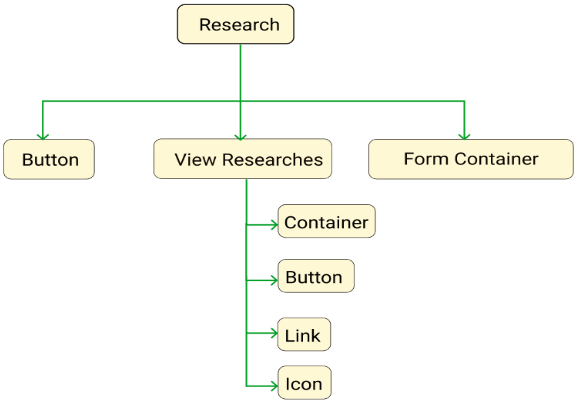
Profile Component



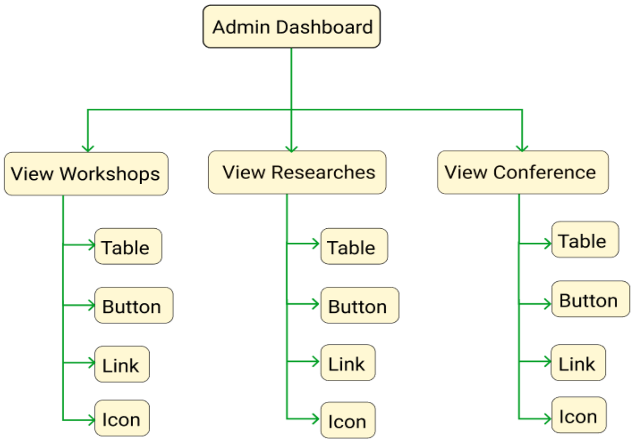
Payment Component



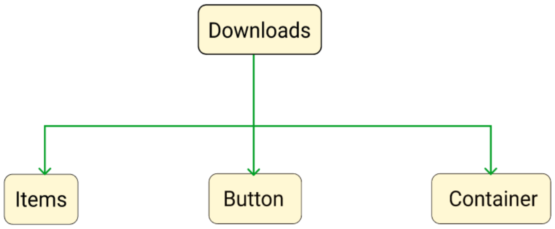
Research Component



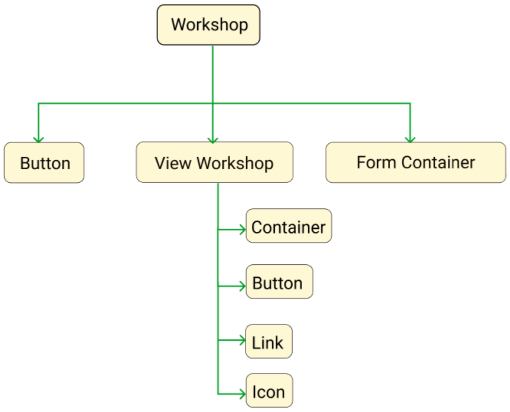
Admin Dashboard



Downloads Component



Workshop Component



Usage of REST API

User related endpoints

Create User - POST (“/api/users”)

- Create new user(attendee/researcher/workshop conductor)

Get User By Id - GET(“/api/users/:id”)

- Get specific user details by user id

Get All Users - GET (“/api/users”)

- Get all users details

Delete User - DELETE (“/api/users/:id”)

- Delete specific user details by id

Update User - PUT(“/api/users/:id”)

- Update specific user details by id

User Login - POST(“/api/users/login”)

- Validate user and login to the system

Conference details related endpoints

Create Conference - POST(“/api/conference”)

- Editor can create a new conference

Update Conference - PUT(“/api/conference/:id”)

- Editor can update conference

Delete Conference - DELETE(“/api/conference/:id”)

- Editor can delete a conference

Get all Conferences - GET(“/api/conference”)

- Every user can view all conferences

Get conference by id - GET(“/api/conference/:id”)

- Every user can view a specific conference

Update Conference approval - PUT(“/api/workshop/:id”)

- Admin can approve conference

Workshops related endpoints

Create Workshop - POST("/api/workshop")

- Workshop Conductor can create a new Workshop

Update workshop - PUT("/api/workshop/:id")

- Workshop Conductor can update Workshop

Delete workshop - DELETE("/api/workshop/:id")

- Workshop Conductor can delete a Workshop

Get all workshop - GET("/api/workshop")

- Workshop Conductor can view all Workshop

Get workshop by id - GET("/api/workshop/:id")

- Workshop Conductor view a specific Workshop

Update workshop approval - PUT("/api/workshop/:id")

- Reviewer can update Workshop

Research paper submission related endpoints

Create Research - POST("/api/research")

- Researcher can submit a new research paper

Update Research - PUT("/api/research/:id")

- Researcher can update research paper

Delete Research - DELETE("/api/research/:id")

- Researcher can delete a research paper

Get All Research - GET("/api/research")

- Researcher can view all conferences

Get Research by id - GET("/api/research/:id")

- Researcher view a specific research paper

Payments related endpoints

Add a payment - POST("/api/payment")

- Researcher and the attendee can add a payment

Get All payments - GET("/api/payment")

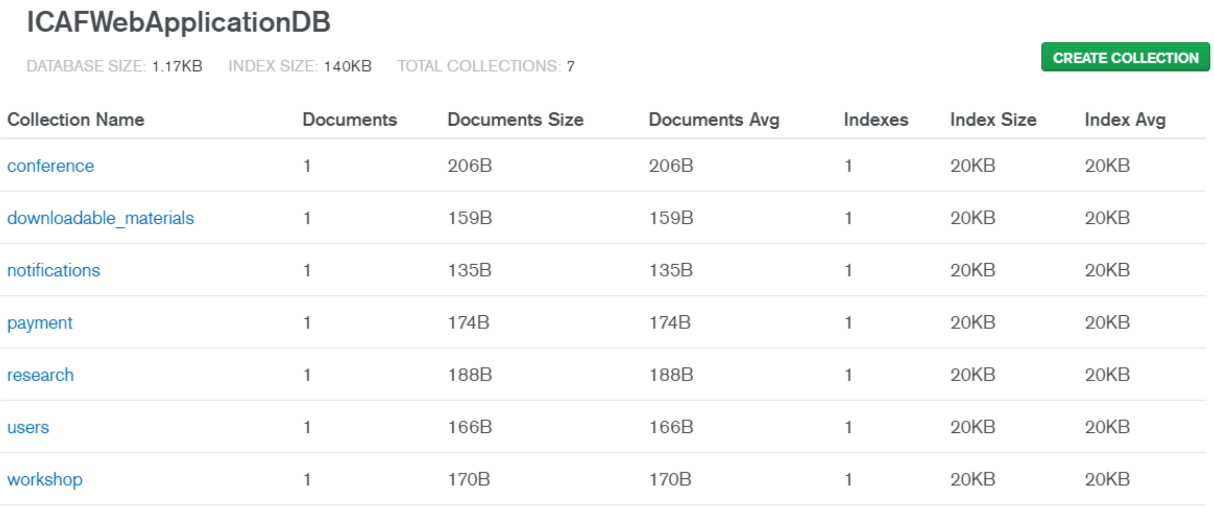
- Admin can view all payments

Get a payment by id - GET("/api/payment/:id")

- Admin can view a specific payment

Above mentioned REST API endpoints might change in future implementations with the development of the application.

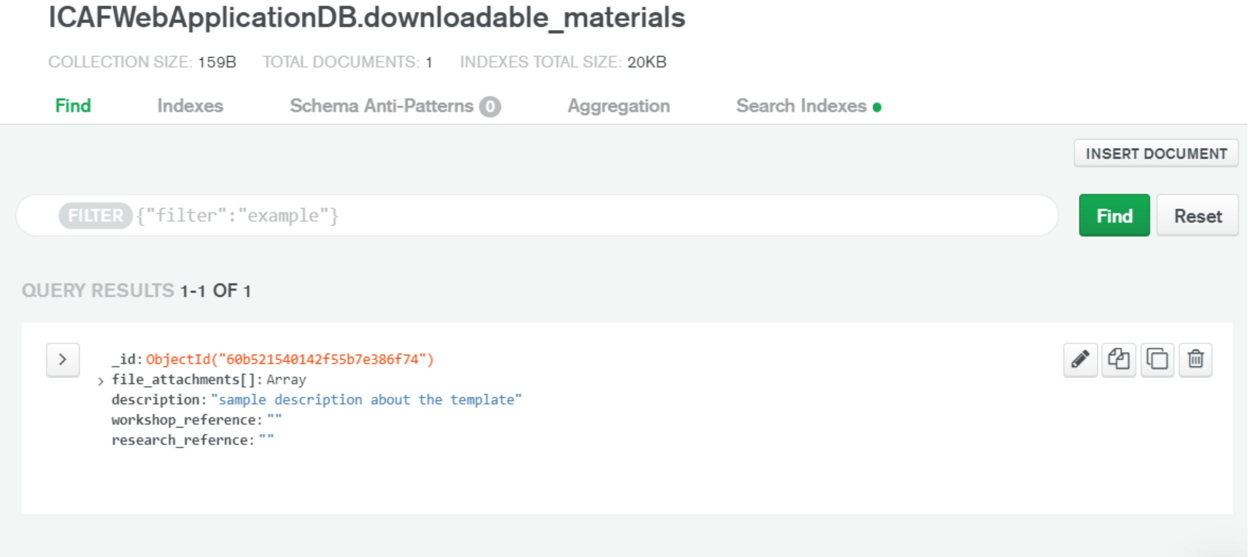
Database



ICAFWebApplicationDB

DATABASE SIZE: 1.17KB INDEX SIZE: 140KB TOTAL COLLECTIONS: 7

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
conference	1	206B	206B	1	20KB	20KB
downloadable_materials	1	159B	159B	1	20KB	20KB
notifications	1	135B	135B	1	20KB	20KB
payment	1	174B	174B	1	20KB	20KB
research	1	188B	188B	1	20KB	20KB
users	1	166B	166B	1	20KB	20KB
workshop	1	170B	170B	1	20KB	20KB

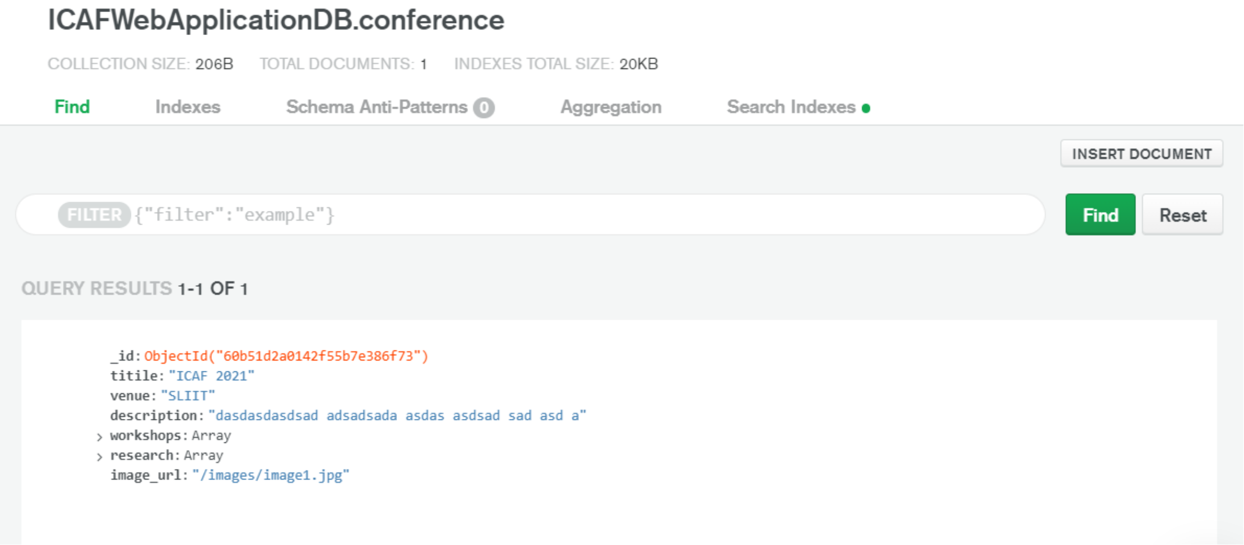


ICAFWebApplicationDB.downloadable_materials

COLLECTION SIZE: 159B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

QUERY RESULTS 1-1 OF 1

```
  _id: ObjectId("60b521540142f55b7e386f74")
> file_attachments[]: Array
  description: "sample description about the template"
  workshop_reference: ""
  research_refernce: ""
```

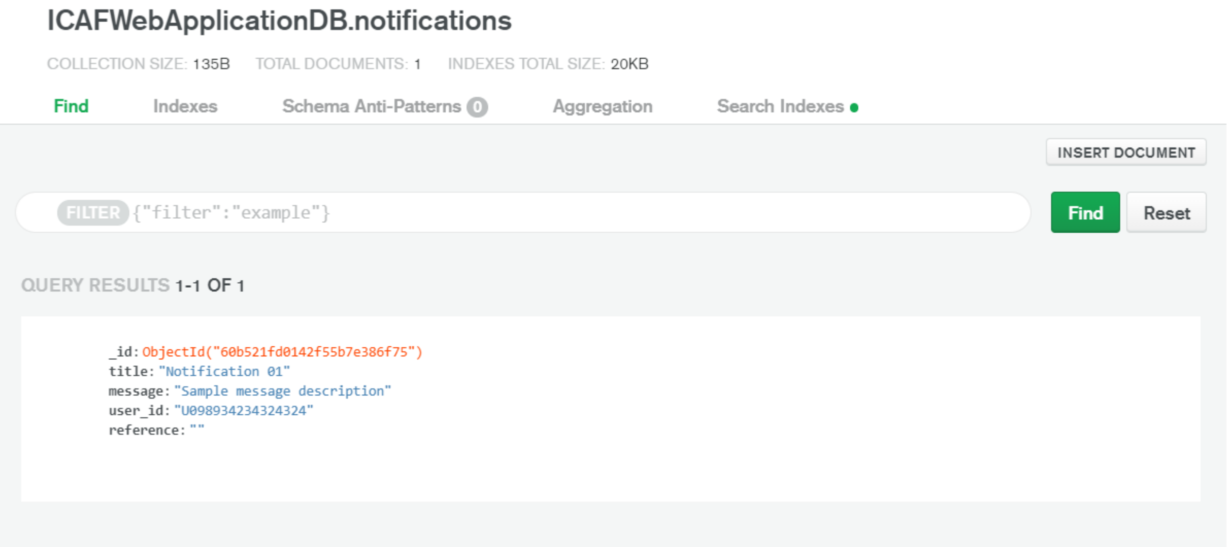


ICAFWebApplicationDB.conference

COLLECTION SIZE: 206B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

QUERY RESULTS 1-1 OF 1

```
  _id: ObjectId("60b51d2a0142f55b7e386f73")
  titile: "ICAF 2021"
  venue: "SLIIT"
  description: "dasdasdasdsad adsadsada asdas asdsad sad asd a"
> workshops: Array
> research: Array
  image_url: "/images/image1.jpg"
```

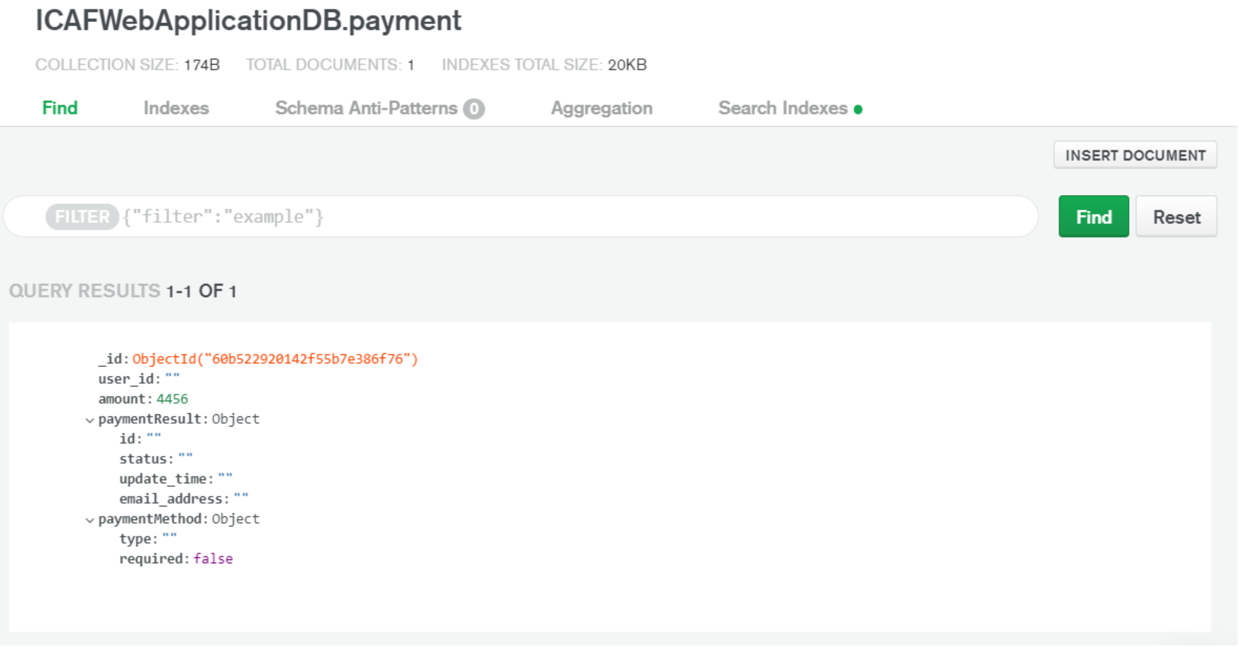


ICAFWebApplicationDB.notifications

COLLECTION SIZE: 135B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("60b521fd0142f55b7e386f75")
title: "Notification 01"
message: "Sample message description"
user_id: "U098934234324324"
reference: ""
```

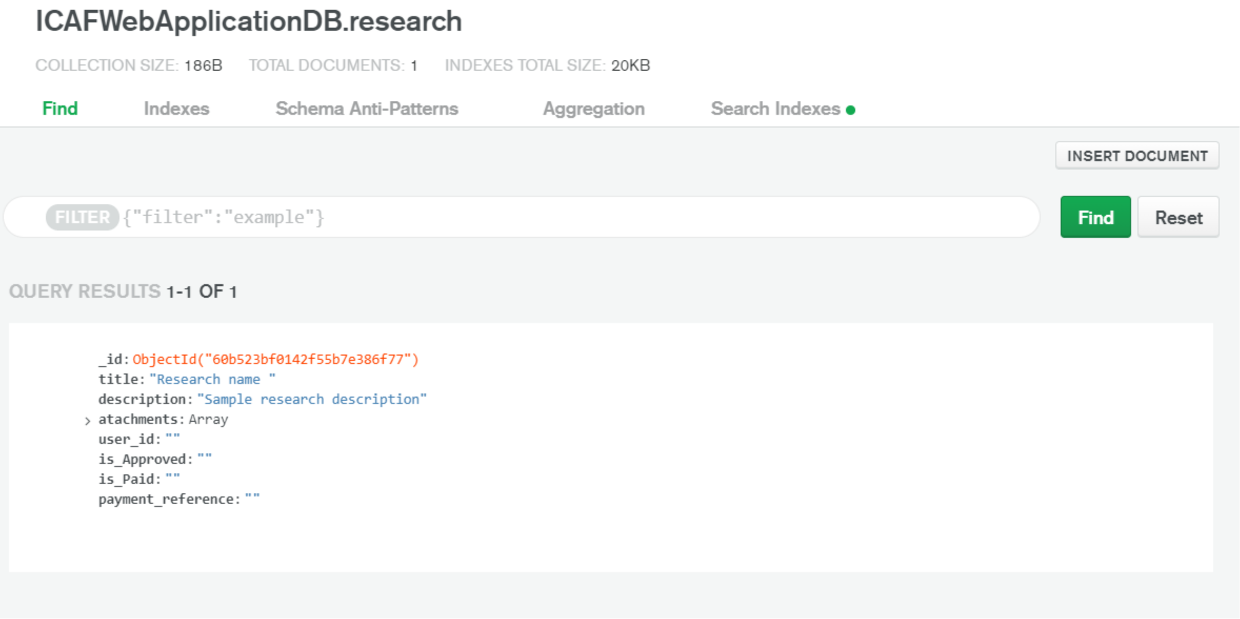


ICAFWebApplicationDB.payment

COLLECTION SIZE: 174B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

QUERY RESULTS 1-1 OF 1

```
  _id: ObjectId("60b522920142f55b7e386f76")
  user_id: ""
  amount: 4456
⌄ paymentResult: Object
    id: ""
    status: ""
    update_time: ""
    email_address: ""
⌄ paymentMethod: Object
    type: ""
    required: false
```



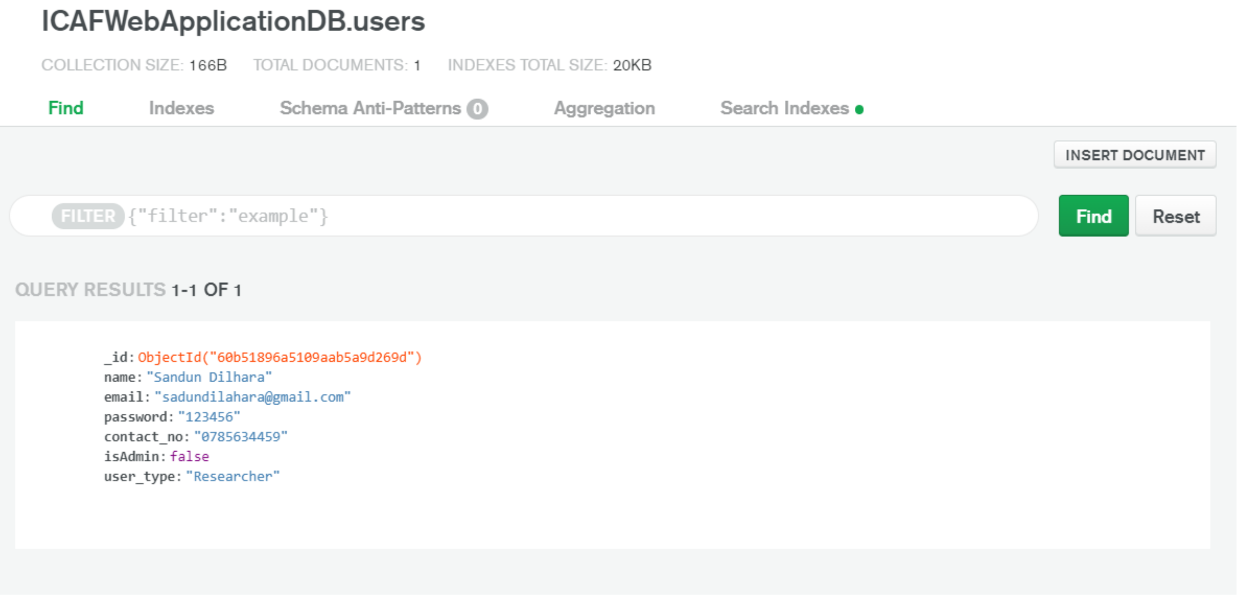
ICAFWebApplicationDB.research

COLLECTION SIZE: 186B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

{"filter":"example"}

QUERY RESULTS 1-1 OF 1

```
  _id: ObjectId("60b523bf0142f55b7e386f77")
  title: "Research name "
  description: "Sample research description"
> atachments: Array
  user_id: ""
  is_Approved: ""
  is_Paid: ""
  payment_reference: ""
```



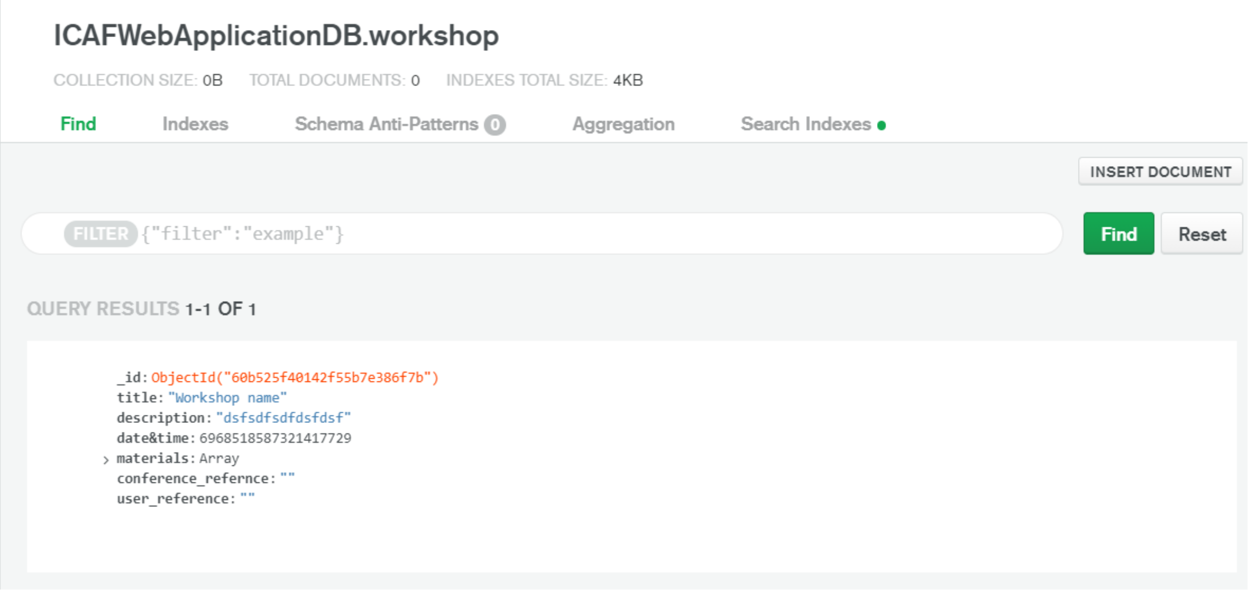
ICAFWebApplicationDB.users

COLLECTION SIZE: 166B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

{"filter":"example"}

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("60b51896a5109aab5a9d269d")
name: "Sandun Dilhara"
email: "sadundilahara@gmail.com"
password: "123456"
contact_no: "0785634459"
isAdmin: false
user_type: "Researcher"
```



ICAFWebApplicationDB.workshop

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

{"filter":"example"}

QUERY RESULTS 1-1 OF 1

```
  _id: ObjectId("60b525f40142f55b7e386f7b")
  title: "Workshop name"
  description: "dsfsdfsdfdsfdsf"
  date&time: 6968518587321417729
> materials: Array
  conference_refernce: ""
  user_reference: ""
```