**RTI APPLICATION RECORD KEEPING SYSTEM**A Project Report Submitted to Cotton University in Partial Fulfillment of the  
Requirements for the Degree of  
**Master of Computer Application (MCA)**

In the Department of Computer Science and Information Technology  
By  
**Name of the Student: Mridul Barman**Enrollment Number: MCA 1965028  
Name of Semester: 6th semester (2019-22)



Under the Guidance of:



DEPARTMENT OF COMPUTER SCIENCE AND IT  
COTTON UNIVERSITY, GUWAHATI  
ASSAM-781001, INDIA  
AUGUST 2022



DEPARTMENT OF COMPUTER SCIENCE AND IT  
**COTTON UNIVERSITY, GUWAHATI-781001**Website: http://cottonuniversity.ac.in  
Phone No.: 0361-2608074

**CERTIFICATE**

This is to certify that the project entitled ***“Rti Application Record Keeping System”*** submitted by **Mridul Barman** for the award of the degree of **Master of Computer Application (MCA)** in the Department of Computer Science and Information Technology is the outcome of a bona fide project work under my supervision. This work has not been submitted previously for any other degree of this or any other University. It is further certified that the candidate has complied with all the formalities as per the requirements of Cotton University, Guwahati-01. I recommend that the project report may be accepted in partial fulfillment of the  
requirements for the degree of MCA of this University.





DEPARTMENT OF COMPUTER SCIENCE AND IT  
**COTTON UNIVERSITY, GUWAHATI-781001**Website: http://cottonuniversity.ac.in  
Phone No.: 0361-2608074

**CERTIFICATE**

The Project Report entitled “**RTI APPLICATION RECORD KEEPING SYSTEM**” submitted by **Mridul Barman (MCA1965028)** in partial fulfillment of requirements for the degree of **Master of Computer Application (MCA)** of Cotton University has been examined.



**DECLARATION**

I, **Mridul Barman**, bearing MCA Enrollment No.:MCA1965028 hereby declare  
that the subject matter of the project entitled ***“RTI Application Record Keeping System”*** is the record of work done by me under the guidance of Name of the Guide, Department of Computer Science and Information Technology, Cotton University, Guwahati-01, Assam. I further declare that the contents of this project report did not form the basis for the award of any degree to me or to anybody else to the best of my knowledge. The report has not been submitted to any other University or Institution. This report is being submitted to Cotton University, Guwahati-01 for the degree of MCA in the Department of Computer Science and Information Technology.



Date:

Place:

**Annexure-IV**

**ACKNOWLEDGEMENT**

I am extremely grateful to my supervisor respected **Mr. Prakash Chauhan**,  
Department of Computer Science and Information Technology, Cotton University,  
Guwahati-01 for his valuable advice, encouragement, endless support and guidance  
with patience during the course of my project work and write up this project report.  
I would like to express my sincere gratitude to the Head of the Department, all the  
faculty members, students, non-teaching staff and office staff members of the  
Department of Computer Science and Information Technology, Cotton University,  
Guwahati-01 for their help and moral support in various forms during the course of  
my project work I remember forever my parents for all their motivation and help to reach a greater position in my life. I am also very much thankful to my mother and other family members for their infinite love, motivation, suggestions and supports.

Mridul Barman

**Annexure-I**

**LIST OF ABBREVIATIONS**

| **SL No.** | **ABBREVIATION** | **FULL FORM** |
| --- | --- | --- |
| 1. | SLDC | Software Development Life Cycle |
| 2. | RAM | Random Access Memory |
| 3. | ROM | Read Only Memory |
| 4. | GB | Giga Byte |
| 5. | OS | Operating System |
| 6. | H/W | Hardware |
| 7. | S/W | Software |
| 8. | E-R | Entity-Relationship |
| 9. | DB | Database |
| 10. | APP | Application |
| 11. | UI | User Interface |
| 12. | CPU | Central Processing Unit |
| 13. | HTML | Hypertext Markup Language |
| 14. | CSS | Cascading Style Sheet |
| 15. | TS | Typescript |
| 16. | GraphQL | Graph Query Language |
| 17. | JS | Javascript |
| 18. | MERN | MongoDB Express React Node |

**Annexure-II**

**LIST OF FIGURES**

| **FIGURE NO.** | **FIGURE NAME** | **PAGE NO.** |
| --- | --- | --- |
| 1. | 0-Level DFD | 15 |
| 2. | 1-Level User DFD | 16 |
| 3. | 1-Level Admin DFD | 17 |
| 4. | ER Diagram | 19 |
| 5. | Snapshots of the project | 24-29 |

**Annexure-III**

**ABSTRACT**

Using the rti application record keeping system anyone can apply or manage already applied rti remotely. After applying the rti admins will send the replies to the users. The main objective of this project is to make applying for rti easy and accessible. Admins can filter old rti files very easily with applicants name and date.

React with tailwind css is used to develop its frontend in typescript programming language. In its backend, NodeJs with Express is used as server written in Typescript programming language with GraphQL for serving its API and MongoDb is used as a NoSQL database in its backend.

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

TypeScript is a free and open source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript and adds optional static typing to the language. It is designed for the development of large applications and transpiles to JavaScript

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine and executes JavaScript code outside a web browser, which was designed to build scalable network applications.

Page | 1

**TABLE OF CONTENTS**

| LIST OF ABBREVIATIONS. **. . . . . . . . . . . . . . . . . . . . . . . .** | I |
| --- | --- |
| LIST OF FIGURES. **. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .** | II |
| ABSTRACT. **. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .** | III |

CHAPTER PAGE NO.

1. **INTRODUCTION** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 3-4
   1. About The Project . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
   2. About The Organization . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . 3
   3. Aim And Objective Of The Project . . .. . .. .. . .. . .. . . .. . . .. .. 4
2. **REVIEW OF LITERATURE**- - - - - - - - - - - - - - - - - -- - -- - – 5-6

2.1 Problem . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2.2 Solution. . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2.3 Disadvantages of the Proposed System. . . . . .. . . . . . . . . . . . . 6

2.4 Advantages of the Proposed System. . . . . .. . . . . . . . . . . . . . . 6

1. **SOFTWARE AND HARDWARE REQUIREMENT** - - - - - - 7-12
   1. Softwares Used In the Project . . . . . . . . . . . . . .. . . .. . . . .. . . 7
   2. System Requirements . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . .. 12
2. **SYSTEM DESIGN**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - – 13-20
   1. Introduction . . . . . . .. .. . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . 13
   2. Structured Designed. . .. . . . . . . . . . . . . . .. . . . . . . . . . . . . . . 13
   3. DFD (Data Flow Diagram). . .. . . . . . . . . . . . . . .. . . . . . . . . . 14
   4. ER (Entity Relation) Diagram.. .. . . . . . . . . . . . . . .. . . . . . . . 18
   5. Database Design. . .. . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . 20
3. **IMPLEMENTATION OR DEVELOPMENT**- - - - - - - - - - - 21-22
   1. Introduction. . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
   2. System Implementation .... . . . . . . . . . . . . . . . . . . . . . . . . . . 22
4. **RESULT AND DISCUSSION**- - - - - - - - - - - - - - - - - - - - - - 23-29
   1. Result and discussion. . .. . . . . . . . . . . . . . .. . . . . . . . . . . . . . 23

Page | 2

* 1. Snapshots. . .. . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . 24

1. **CONCLUSION AND FUTURE WORK**- - - - - - - - - - - - - - 30
   1. Conclusion. . .. . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . 30
   2. Future Works. . .. . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . 30

**LIST OF REFERENCES**- - -- - - - - - - - - - - - - - - - - - - - - - - - - - - 31

Page | 3



**INTRODUCTION**

* 1. **About the Project**

Using the rti application record keeping system anyone can apply or manage already applied rti remotely. After applying the rti admins will send the replies to the users. The main objective of this project is to make applying for rti easy and accessible. Admins can filter old rti files very easily with applicants name and date.

The advantage of this project is that it will allow anyone to access or maintain rti applications from anywhere.

**1.2 About the Organization**

Rashtriya Uchchattar Shiksha Abhiyan (RUSA) is a holistic scheme of development for higher education in India initiated in 2013 by the Ministry of Human Resource Development, Government of India. The centrally sponsored scheme aims at providing strategic funding to higher educational institutions throughout the country. Funding is provided by the central ministry through the state governments and union territories (UT), which in coordination with the central Project Appraisal Board will monitor the academic, administrative and financial advancements taken under the scheme. A total of 316 state public universities and 13,024 colleges will be covered under it.

RUSA aims to provide equal development to all higher institutions and rectify weaknesses in the higher education system. Its target achievement is to raise the gross enrolment ratio to 32% by the end of XII Plan in 2017. The major objectives are to: improve the overall quality of existing state institutions by ensuring that all institutions conform to prescribed norms and standards and adopt accreditation as a mandatory quality assurance framework. usher transformative reforms in

Page | 4

the state higher education system by creating a facilitating institutional structure for planning and monitoring at the state level, promoting autonomy in state universities and improving governance in institutions. ensure academic and examination reforms in the higher educational institutions. enable conversion of some of the universities into research universities at par with the best in the world. create opportunities for states to undertake reforms in the affiliation system in order to ensure that the reforms and resource requirements of affiliated colleges are adequately met. ensure adequate availability of quality faculty in all higher educational institutions and ensure capacity building at all levels of employment. create an enabling atmosphere in the higher educational institutions to devote themselves to research and innovations. expand the institutional base by creating additional capacity in existing institutions and establishing new institutions, in order to achieve enrolment targets. correct regional imbalances in access to higher education by facilitating access to high quality institutions in urban and semi-urban areas, creating opportunities for students from rural areas to get access to better quality institutions and setting up institutions in un-served and underserved areas. improve equity in higher education by providing adequate opportunities of higher education to SC/STs and socially and educationally backward classes; promote inclusion of women, minorities, and differently abled persons.

**1.3 Aim and Objective of the Project**

Aim of the project is to make applying and managing rti applications easy and fast. The main objective of this project is to make applying for rti easy and accessible. Admins can filter old rti files very easily with applicants name and date.

Page | 5



**REVIEW OF LITERATURE**

**2.1 Problem**

The main problem that arises is the absence of an efficient system. The pen-paper technique requires more time and as it is recorded manually so there are high chances that either the data may get lost or the information is recorded incorrectly. Since the data is not recorded in a database in case of manual process it leads to data redundancy. Also, manual record maintenance consumes a lot of time and is a much slower process than the digitalized system.

**2.2 Solution**

The present goal is to create a web based system that will help anyone to apply for rti online. In the system admins(SPIO, 1st & 2nd appellate authorities) who can view the applications and send replies to users or transfer the application to third party.

At the same it will become easy for the admin to view and also keep record for all the rti applications in a web based system which will be way better than the traditional pen and paper.

Page | 6

**2.3 Disadvantages of the Proposed System**

* The current system is time consuming.
* It takes a lot of effort to do everything in a manual mode. As it includes traditional file system which is a difficult task.
* It is very difficult to check for data redundancy and managing all the system processes manually.
* Lots of human effort is required.
* Finding old rti applications consumes lots of time.

**2.4 Advantages of the Previous System**

* In comparison to the previous system the proposed system is less time consuming.
* It takes less effort in managing the whole admission process as it will be done in web application.
* It will be easy in comparison to the traditional process as repeating the same process will be easy in web application as compared to the traditional system and it can be updated any time. The chance of missing data meshing information and following a lengthy process can be skipped.
* It is very cost effective.
* A very less amount of human effort is required as compared to the traditional process.
* Filtering and Sorting old rti application is very fast.

Page | 7



**SOFTWARE AND HARDWARE REQUIREMENT**

**3.1 Softwares Used In the Project**

**3.1.1 IDE’s:**

**VISUAL STUDIO CODE**: Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, C++, C, Rust and Fortran. It is based on the Electron framework.

**3.1.2 Server:**

**NODEJS**: Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine (i.e. V8 engine) and executes JavaScript code outside a web browser, which was designed to build scalable network applications. Node.js lets developers use JavaScript to write command line tools and for server-side scripting running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying

Page | 8

web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

**3.1.3 API:**

**GRAPHQL**: GraphQL is an open-source data query and manipulation language for APIs, and a runtime for fulfilling queries with existing data. GraphQL was developed internally by Facebook in 2012 before being publicly released in 2015. On 7 November 2018, the GraphQL project was moved from Facebook to the newly established GraphQL Foundation, hosted by the non-profit Linux Foundation. Since 2012, GraphQL's rise has closely followed the adoption timeline as set out by Lee Byron, GraphQL's creator. Byron's goal is to make GraphQL omnipresent across web platforms.

It provides an approach to developing web APIs and has been compared and contrasted with REST and other web service architectures. It allows clients to define the structure of the data required, and the same structure of the data is returned from the server, therefore preventing excessively large amounts of data from being returned. But this has implications for how effective web caching of query results can be. The flexibility and richness of the query language also adds complexity that may not be worthwhile for simple APIs. Despite the name, GraphQL does not provide the richness of graph operations that one might find in a full-fledged graph database such as Neo4j, or even in dialects of SQL that support transitive closure. For example, a GraphQL interface that reports the parents of an individual cannot return, in a single query, the set of all their ancestors.

Page | 9

GraphQL consists of a type system, query language and execution semantics, static validation, and type introspection. It supports reading, writing (mutating), and subscribing to changes to data (realtime updates – most commonly implemented using Websockets). GraphQL servers are available for multiple languages, including Haskell, JavaScript, Perl, Python, Ruby, Java, C++, C#, Scala, Go, Rust, Elixir, Erlang, PHP, R, D and Clojure. The result of a single query is returned in JSON format.

**3.1.4 Languages & Database:**

**HTML:** The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page

Page | 10

**Css:** Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

**Tailwind CSS:** Tailwind CSS can be used to make websites in the fastest and the easiest way. Tailwind CSS is basically a utility-first CSS framework for rapidly building custom user interfaces. It is a highly customizable, low-level CSS framework that gives all of the building blocks that we need to build bespoke designs without any annoying opinionated styles we have to fight to override.

The beauty of this thing called tailwind is it doesn’t impose design specification or how our site should look like, we simply bring tiny components together to construct a user interface that is unique. What Tailwind simply does is take a ‘raw’ CSS file, processes this CSS file over a configuration file, and produces an output.

**Typescript:** TypeScript is a free and open source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript and adds optional static typing to the language. It is designed for the development of large applications and transpiles to JavaScript. As it is a superset of JavaScript, existing JavaScript programs are also valid TypeScript programs.

TypeScript may be used to develop JavaScript applications for both client-side and server-side execution (as with Node.js or Deno). Multiple options are available for transpilation. The default TypeScript Compiler can be used, or the Babel compiler can be invoked to convert TypeScript to JavaScript.

Page | 11

TypeScript supports definition files that can contain type information of existing JavaScript libraries, much like C++ header files can describe the structure of existing object files. This enables other programs to use the values defined in the files as if they were statically typed TypeScript entities.

**Mongo Db:** MongoDB is an open-source document-oriented database that is designed to store a large scale of data and also allows us to work with that data very efficiently. It is categorized under the NoSQL (Not only SQL) database because the storage and retrieval of data in the MongoDB are not in the form of tables. MongoDB uses JSON-like documents with optional schemas.

The MongoDB database is developed and managed by MongoDB.Inc under SSPL(Server Side Public License) and initially released in February 2009. It also provides official driver support for all the popular languages like C, C++, C#, and .Net, Go, Java, Node.js, Perl, PHP, Python, Motor, Ruby, Scala, Swift, Mongoid. So, that we can create an application using any of these languages. Nowadays there are so many companies that used MongoDB like Facebook, Nokia, eBay, Adobe, Google, etc. to store their large amount of data.

Page | 12

**3.2 System Requirements**

**3.2.1 Operating System**

* + Windows 10 (Intel/AMD 64-bit)
  + Linux (Intel/AMD 64-bit, kernel 3.10.0 or higher, Glibc 2.17 or higher)
  + Mac Os

**3.2.2 Web Browser**

* + Google Chrome
  + Mozilla Firefox
  + Microsoft Edge
  + Opera

**3.2.3 Hardware Requirements**

* + Processor: Intel Processor
  + Ram: 8GB

Page | 13



**SYSTEM DESIGN**

**4.1 Introduction:**

System design is the process of defining the architecture components modules, interfaces and data for a system to satisfy specified requirements. It is a solution to the question of how to approach to the creation of a new important phase composed of several steps. It facilitates the understanding and procedural details for implementation of the system recommended in the feasibility study. A systemic approach is required for a coherent and well- running system. Bottom-Up or Top-Down approach is required to take into account all related variables of the system. A designer uses the modeling languages to express the information and knowledge in a structure of system that is defined by a consistent set of rules and definitions.

**4.2 Structured Design:**

Structured design is a conceptualization of problem into several well organized elements of solution. It is basically concerned with the solution design. Benefit of structured design is, it gives better understanding of how the problem is being solved. Structured design also makes it simpler for designer to concentrate on the problem more accurately. Structured design is mostly based on 'divide and conquer’ strategy where a problem is broken into several small problems and each small problem is individually solved until the whole problem is solved.

Page | 14

**4.3 DFD (Data Flow Diagram):**

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various sub processes the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships. Data flow diagrams visually represent systems and processes that would be hard to describe in a chunk of text.

## DFD symbols are- Process

**The destination system data**



**The Flow of Information**

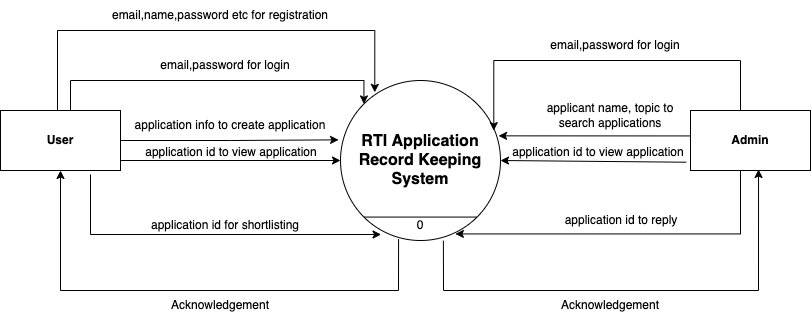
**The table in which information will be stored**



**The result output**

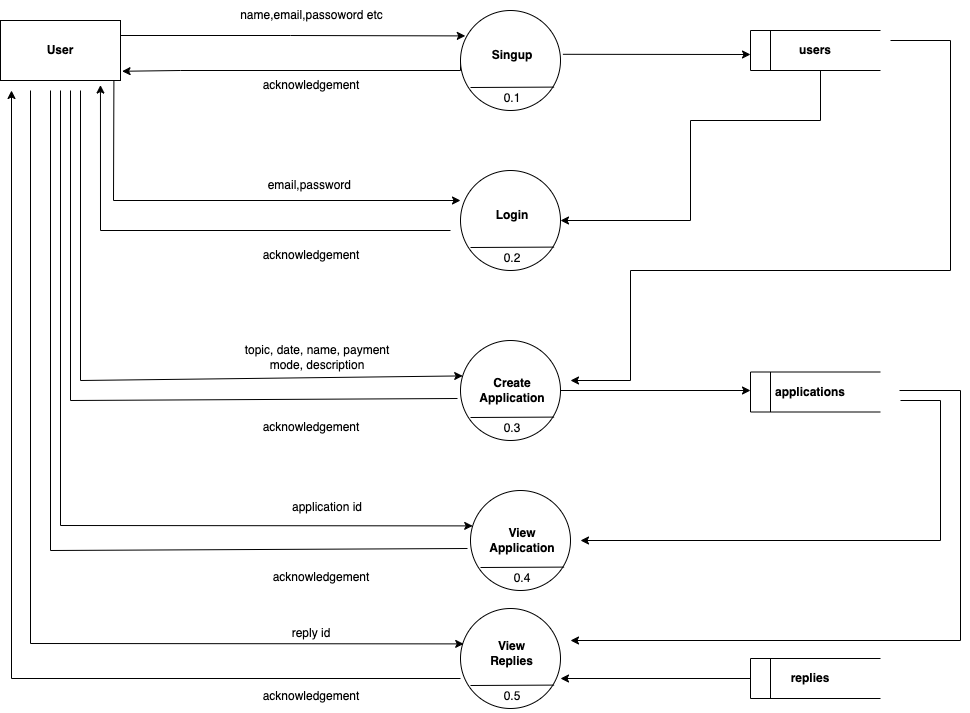
Page | 15

**4.3.1 Zero Level DFD of the Proposed System**

****

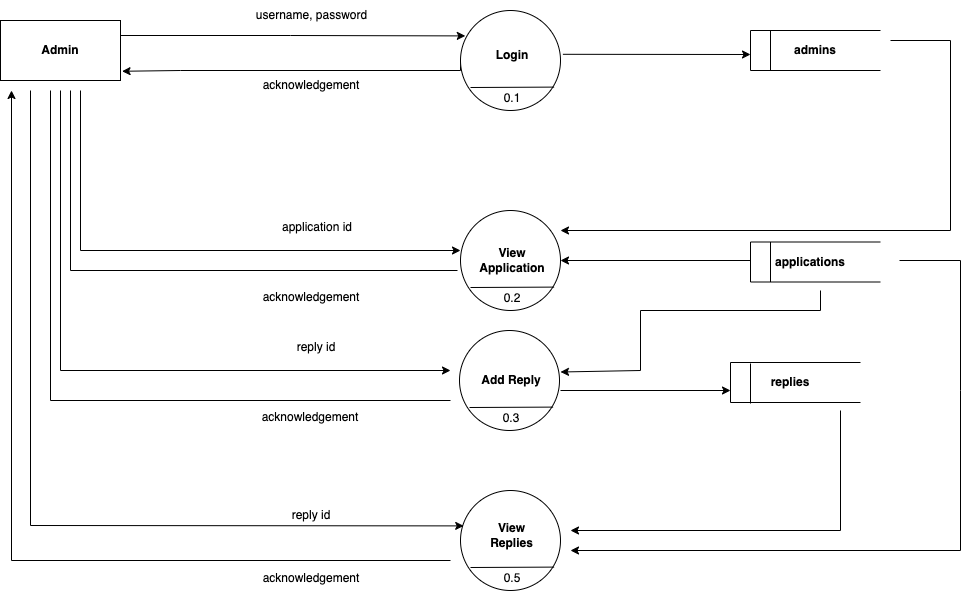
Page | 16

**4.3.2 User DFD of the Proposed System**

****

Page | 17

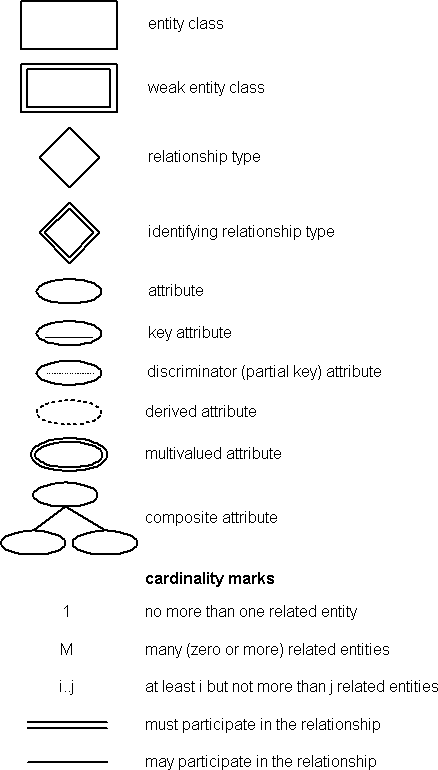
**4.3.3 Admin DFD of the Proposed System**

****

Page | 18

**4.4 Entity Relation(ER) Diagram:**

An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an objector concept about which data is stored.



Page | 19

**4.4.1 ER Diagram of the Proposed System**

****

Page | 20

**4.5 Database Design:**

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database management system manages the data accordingly.

Database design involves classifying data and identifying interrelationships. This theoretical representation of the data is called an ontology. The ontology is the theory behind the database's design.

Page | 21



**IMPLEMENTATION OR DEVELOPMENT**

**5.1 Introduction**

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer.

Two Category of Software Testing:

Black box testing.

White box testing.

System test falls under the black box testing category of software testing.

White box testing is the testing of the internal workings or code of a software application. In contrast, black box or System Testing is the opposite. System test involves the external workings of the software from the user's perspective.

Page | 22

Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called End to End testing scenario.

Verify thorough testing of every input in the application to check for desired outputs.

Testing of the user's experience with the application.

**5.2 System Implementation**

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system is giving confidence on the new system for the users that it will work efficiently and effectively. The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of the current system and its constraints on implementation, designing of methods to achieve the changeover and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system. The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation. The implementation process comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

Page | 23



**RESULT AND DISCUSSION**

**6.1 Result and discussion**

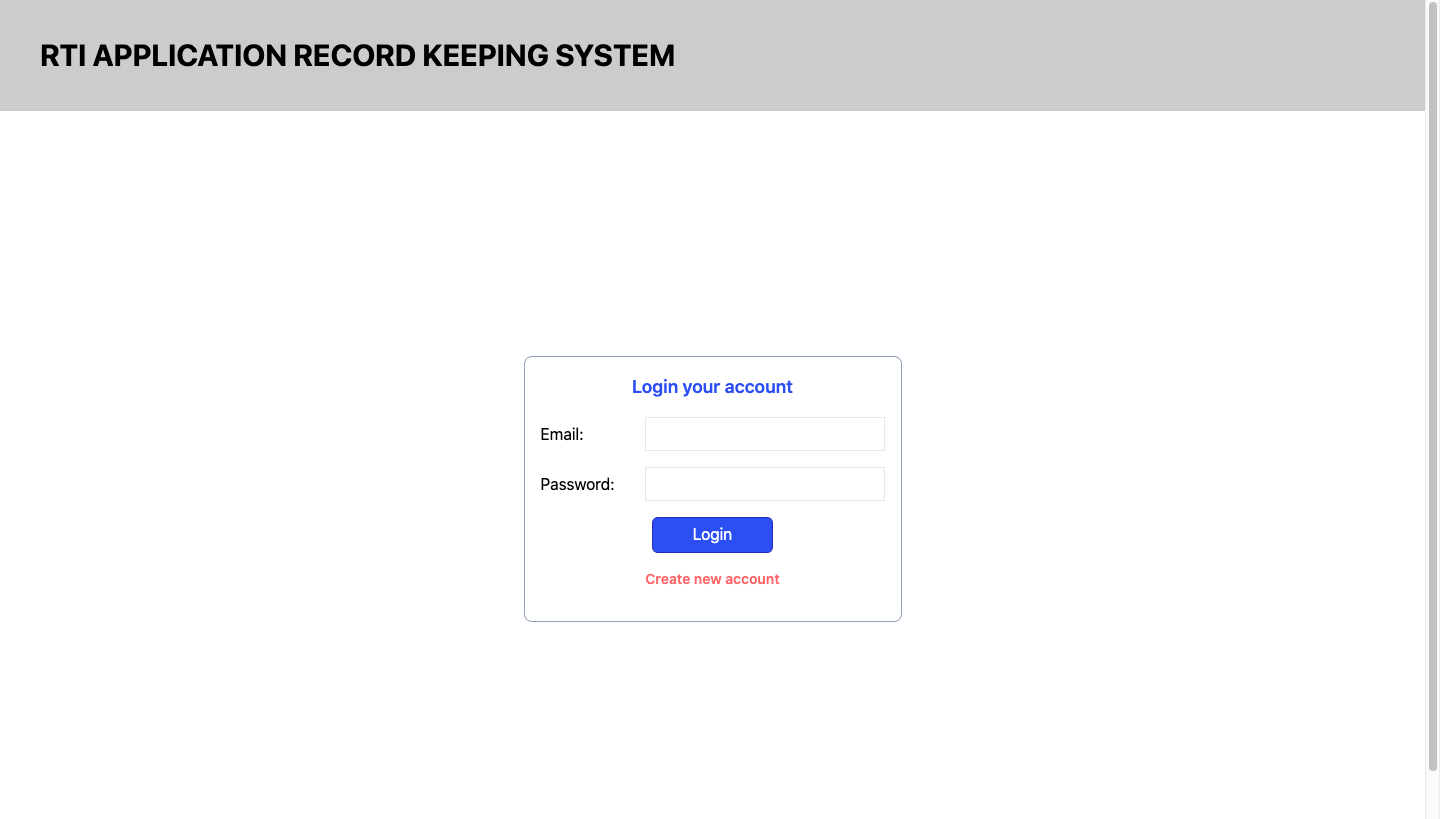
The major goal of this project is to allow anyone to file RTI from anywhere and making managing rti applications simple and fast. In this web application user first signup with their details and then login with email and password. After logging-in user can create new rti application with topic, payment details and description. Admin will view the application and give replies to users or transfer the application to other appellate authorities or to third parties.

Full project is using mongodb as its main database and nodejs as server with react as ui framework. Its using graphql for managing its apis.

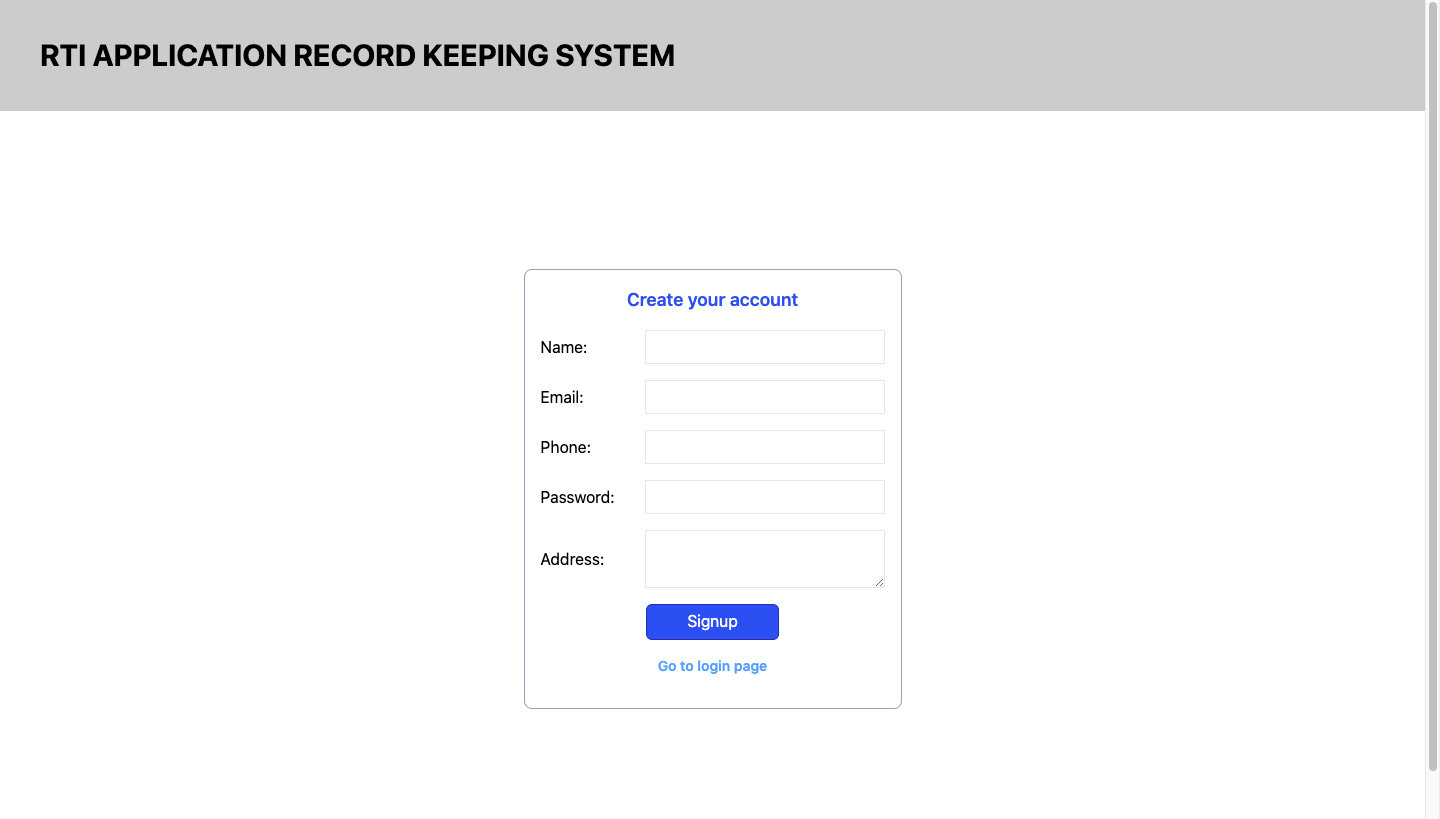
Page | 24

**6.2 Snapshots**

**6.2.1 User Login Page**

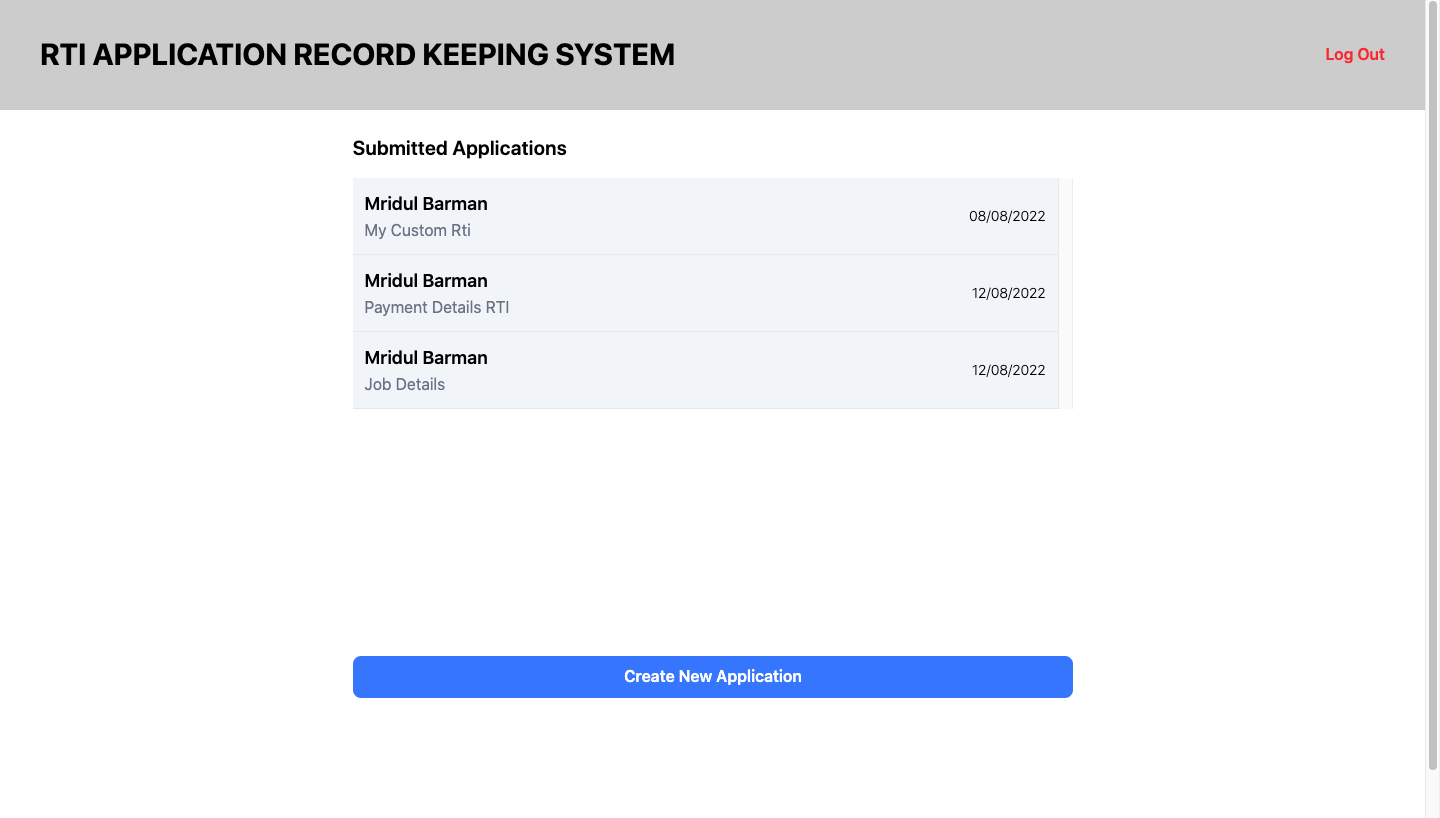
****

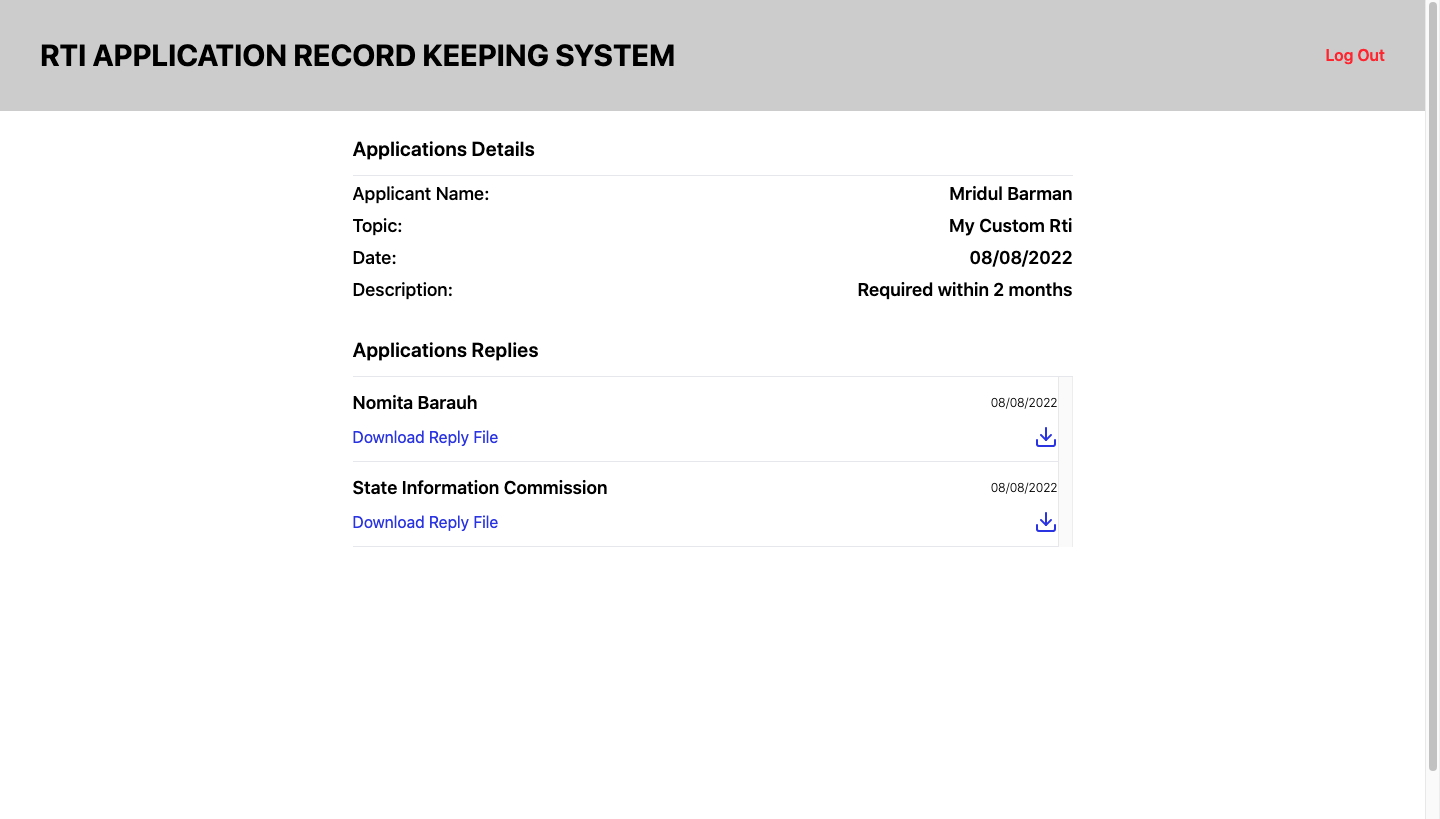
**6.2.2 User Signup Page**

****

Page | 25

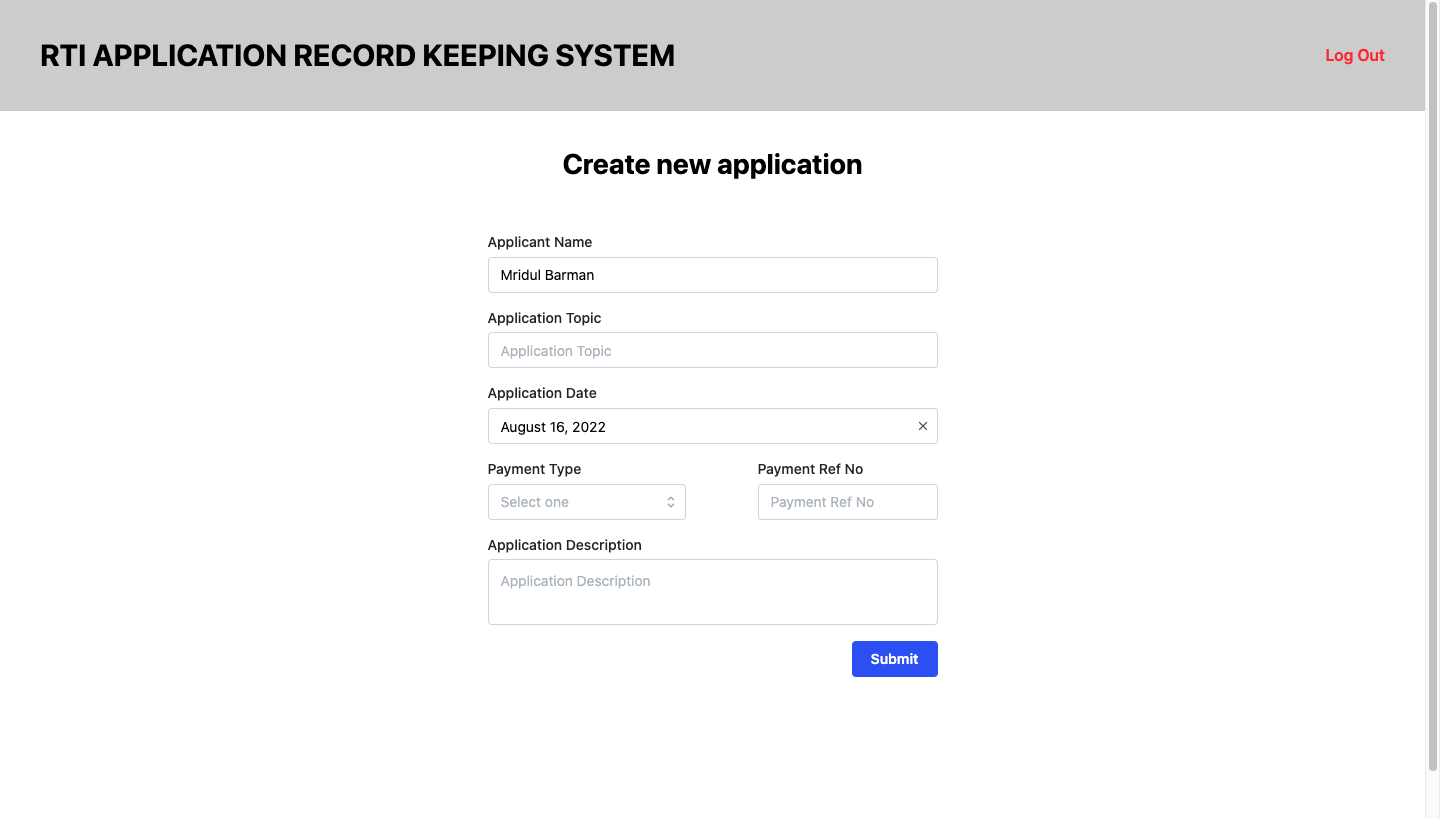
**6.2.3 User Home Page**

****

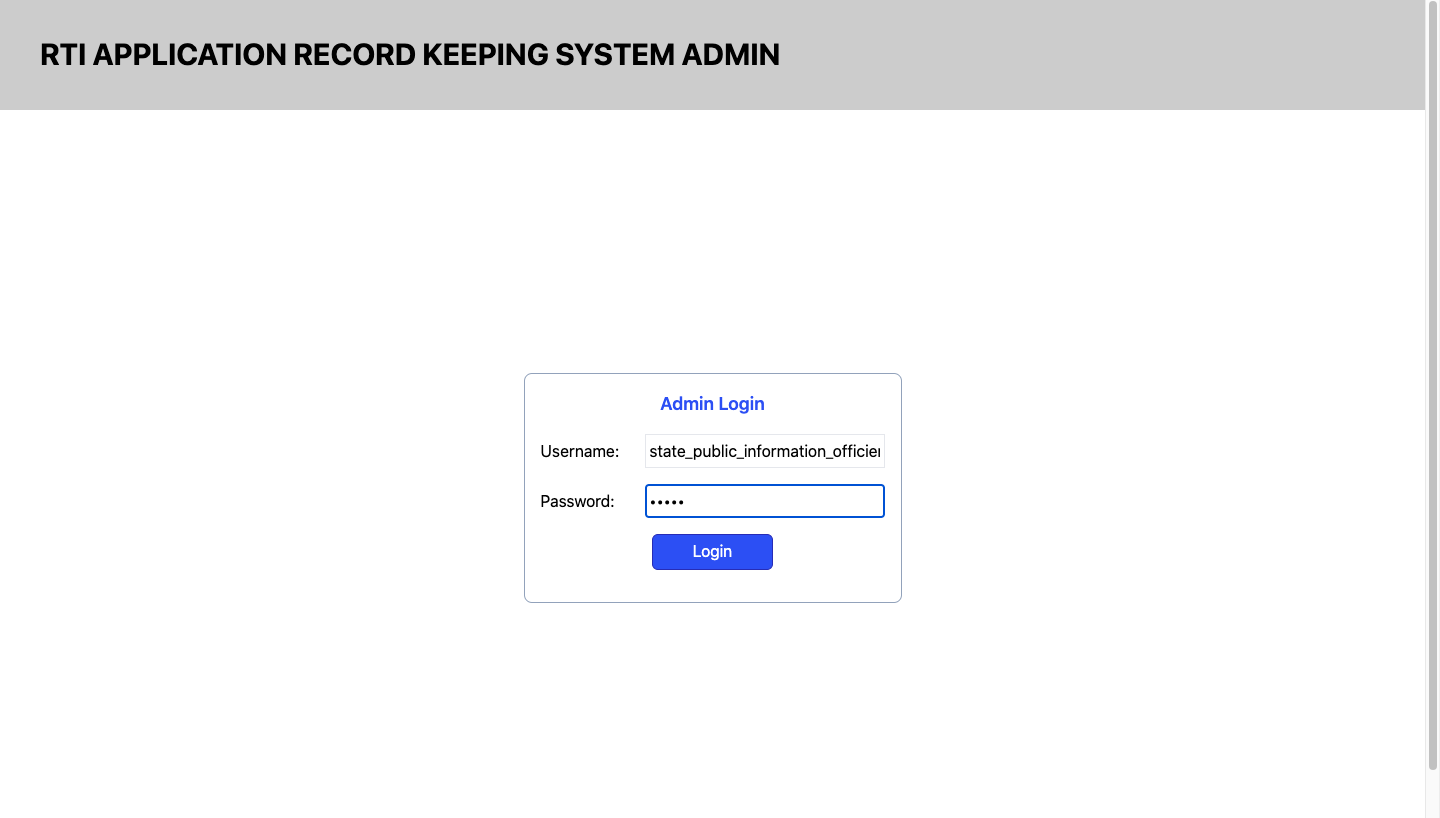
**6.2.4 User View Replies Page**

Page | 26

**6.2.5 User Create Application Page**

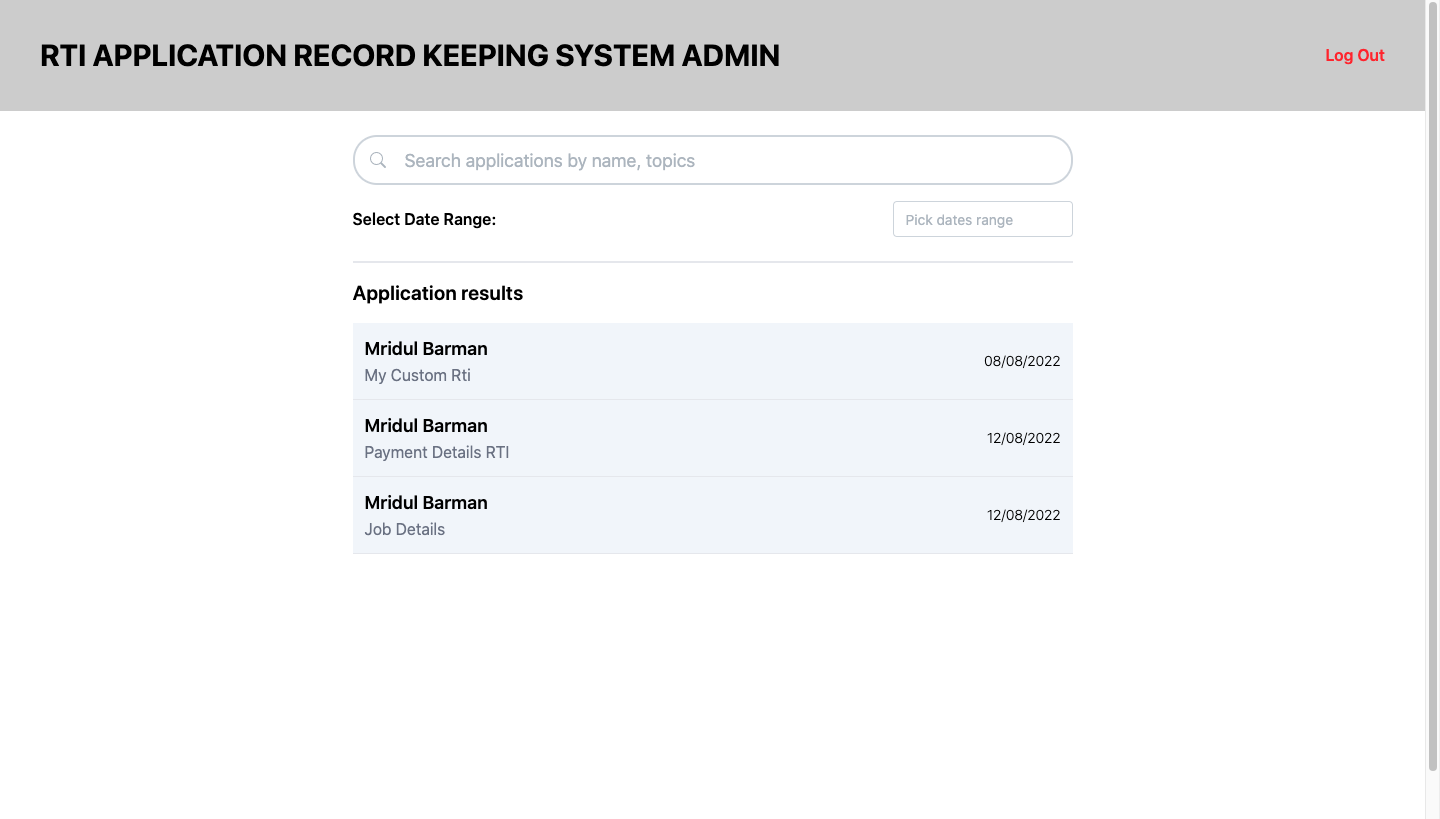
****

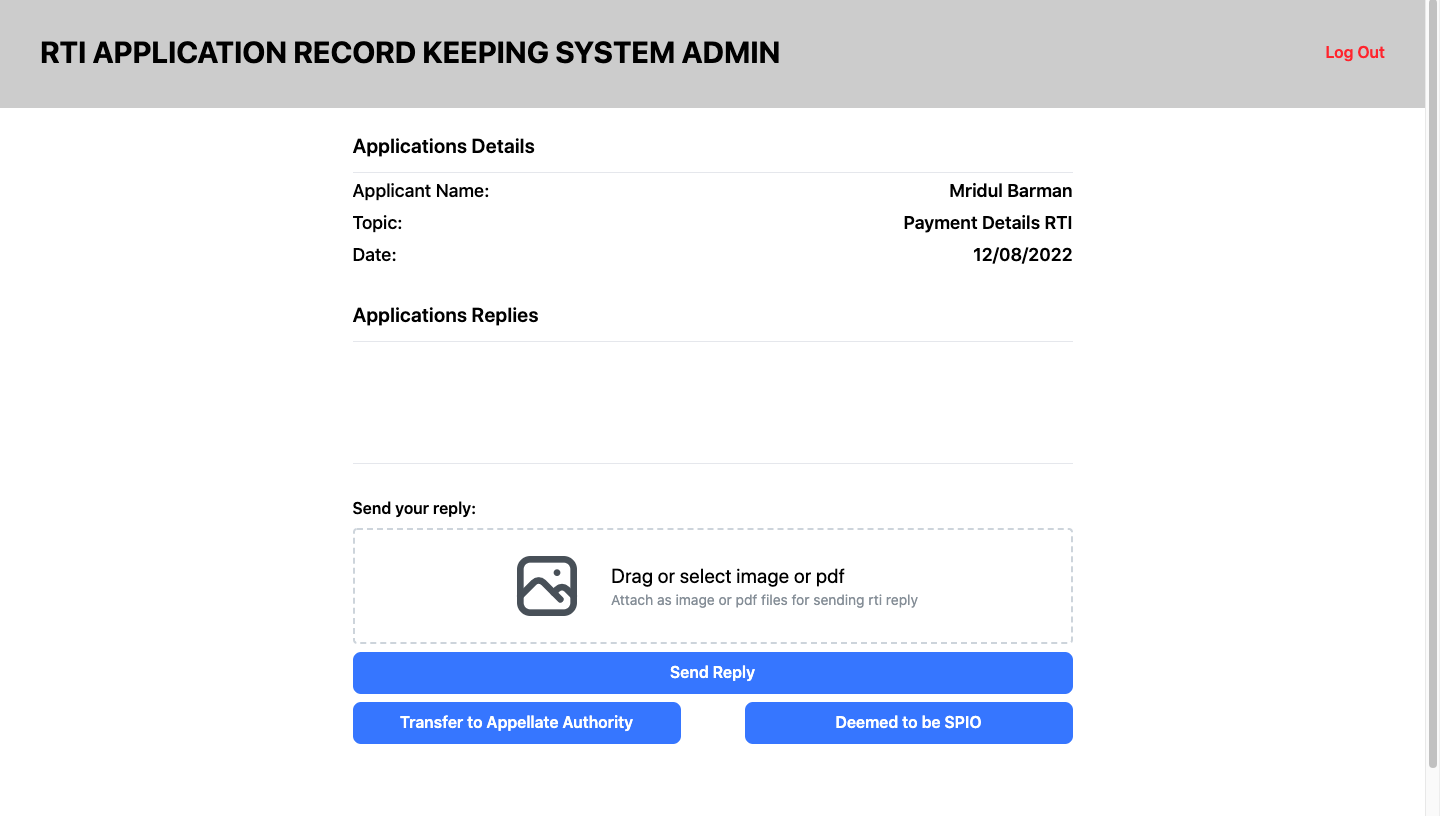
**6.2.6 Admin Login Page**

****

Page | 27

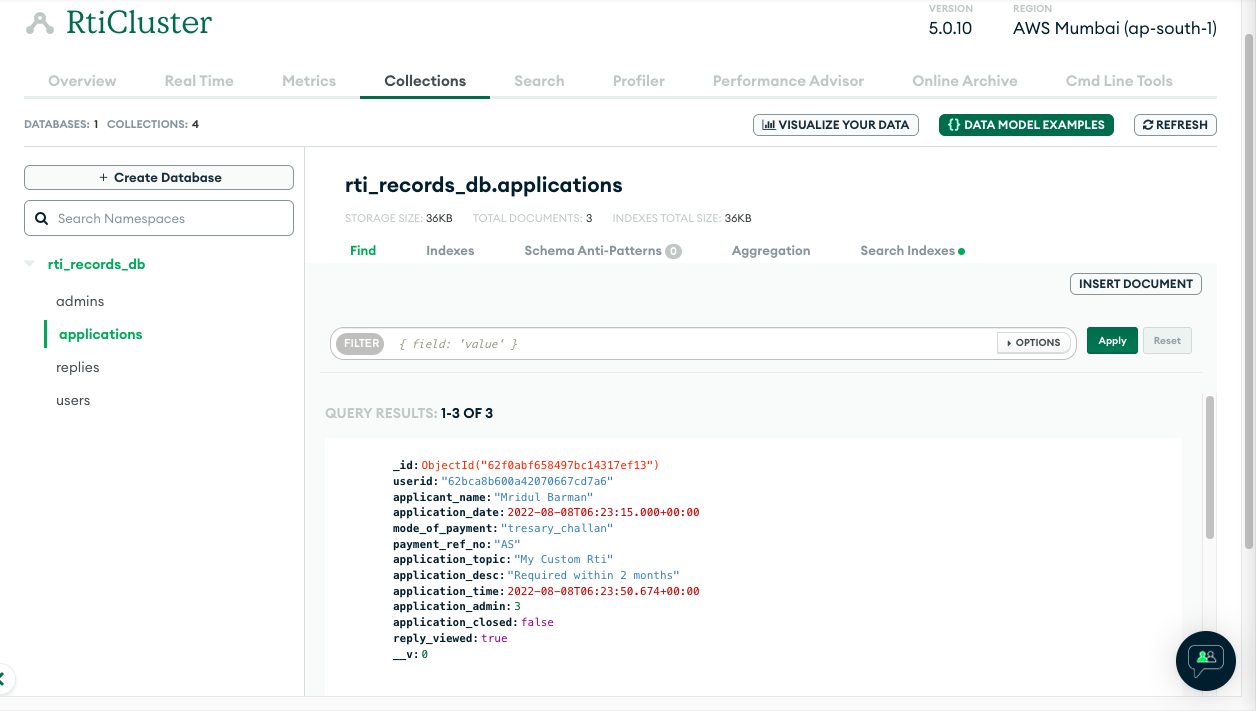
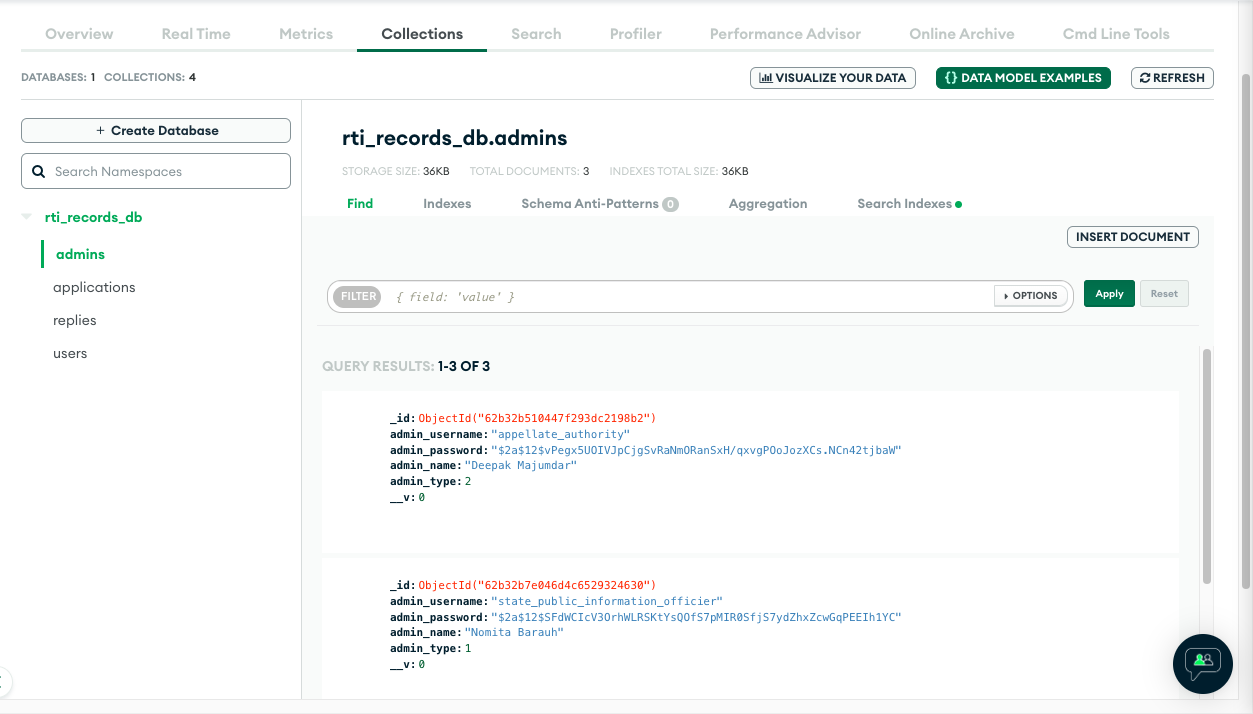
**6.2.7 Admin Home page**

****

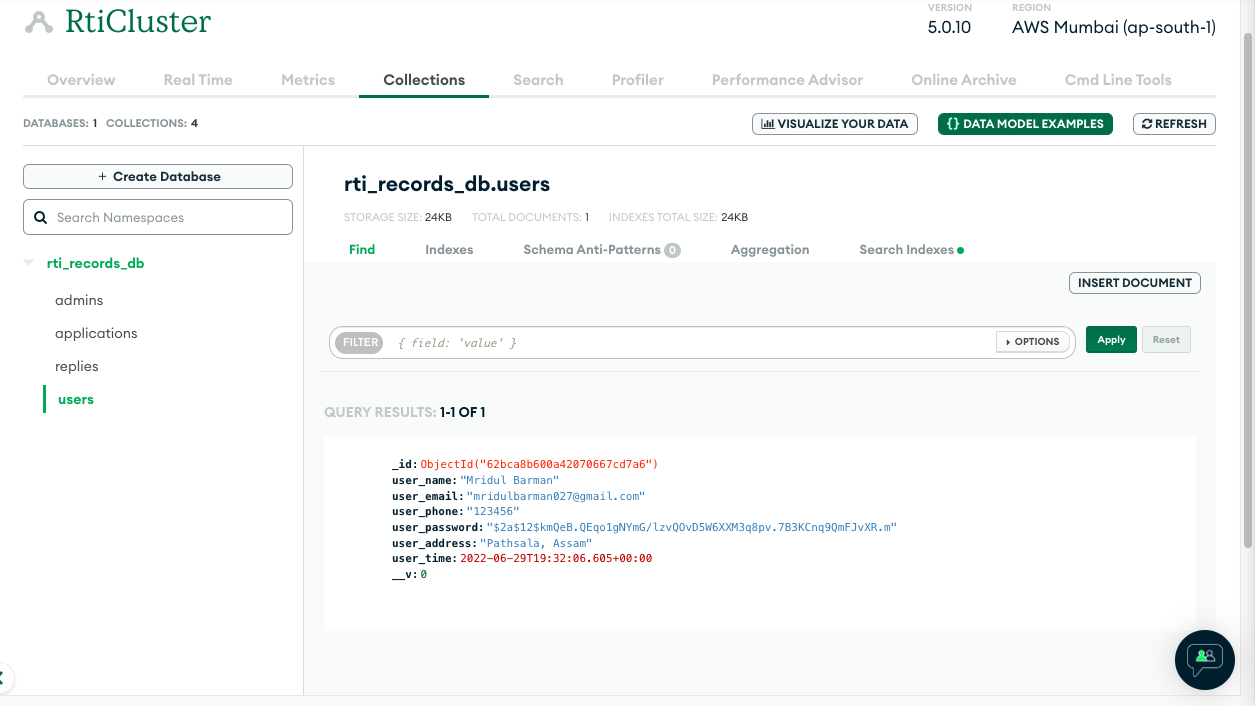
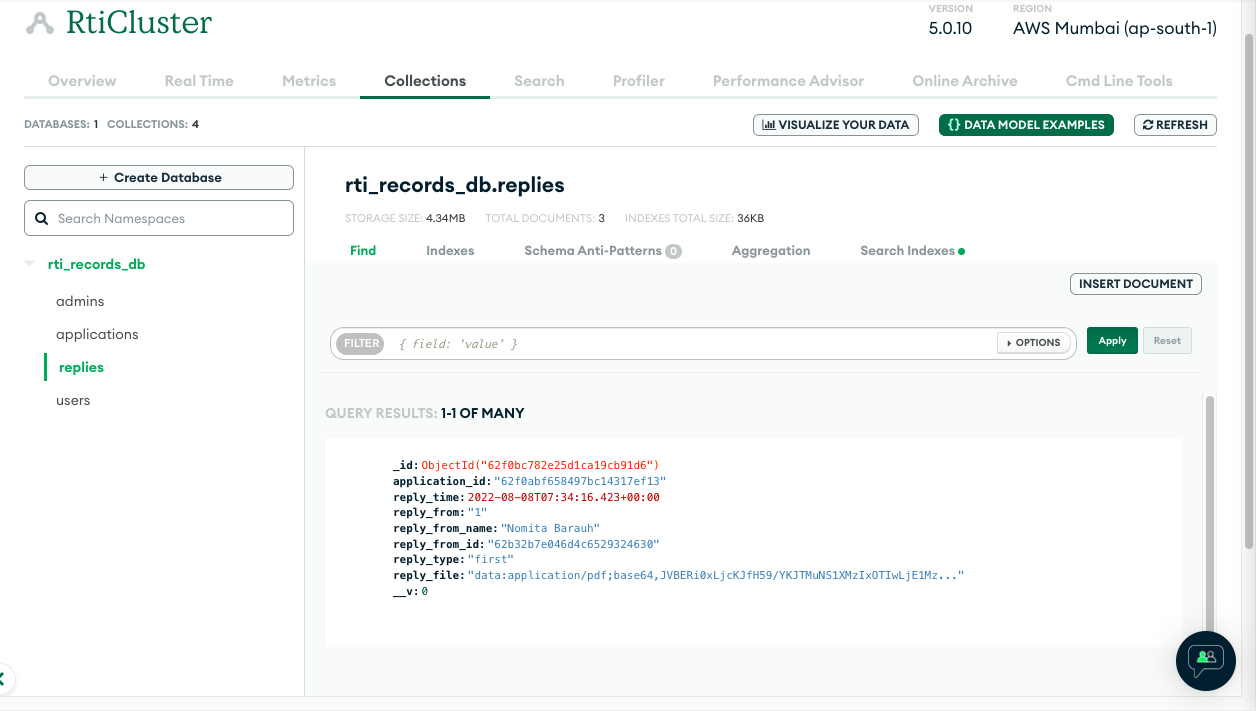
**6.2.8 Admin Send Reply**

Page | 28

**6.2.9 Database Collections**

****

Page | 29

****

Page | 30



**CONCLUSION AND FUTURE WORK**

**7.1 Conclusion**

The main motive of the “Rti Application Record Keeping System” is to computerize the way of keeping all the records of rti applications and developing an easier way to apply for rti and also develop an easy user interface. The system satisfies the needs of the users and is expected to eliminate the problems of the manual system. When things are done manually, it takes a lot of time and also difficult to handle. That is why an automated system of rti application management is much needed to replace the traditional system. The proposed system makes the storing and processing of data simpler and easier.

**7.2 Future Works**

The future scope is it can be implemented in every government department and can add more security and validations like random password generation, adding more modules to the system, hosting it online to make it more accessible to a wide range of users. It can also support manual file upload system so that anyone who doesn’t have proper education can upload rti files directly. It can also support multi regional language support in website so that anyone can use this who doesn’t have proper knowledge in english.

Page | 31

**LIST OF REFERENCES**

1. [https://www.wikipedia.org](https://towardsdatascience.com)
2. [https://www.geeksforgeeks.org](https://towardsdatascience.com)
3. [https://www.mongodb.com](https://towardsdatascience.com)
4. [https://www.typescriptlang.org](https://towardsdatascience.com)
5. [https://graphql.org](https://towardsdatascience.com)
6. [https://tailwindcss.com](https://towardsdatascience.com)
7. [https://nodejs.org/en](https://towardsdatascience.com)
8. [https://rtionline.gov.in](https://towardsdatascience.com)