

Problem Statement and Application Scope:

1. First, here for this project, there are 2 types of users. One is “admin” who can log in as Admin and can delete or add editor and the other one is “editor” who can log in as Editor and can post only articles. All types of users will land on the ‘login’ page first. There is only one option at the top and that is login. Users need to email, password and role value to log in. Otherwise, it will land on this page if the user does not provide the right information.

[Login](#)

Login Page

<input type="text" value="email"/>	<input type="text" value="enter password"/>	<input type="text" value="Role: admin or editor?"/>	<input type="button" value="Log in"/>	<input type="button" value="Reset"/>
------------------------------------	---	---	---------------------------------------	--------------------------------------

Figure: Login Page View

2. When the admin successfully logs in, the admin will land on the user page and see 2 options at the top. One is ‘User’ and the other is ‘Log out’. And in the bottom, the admin can see the user list. On this ‘user’ page, the admin can add a user and delete a user. Admin can add “editor” as well as “admin” but can not delete the “admin”. By clicking the logout option, the admin will log out and go to the log-in page. Admin can only add one email user. So there is no duplicate email user.

[User](#)
[Log out](#)

Add User:

<input type="text" value="username"/>	<input type="text" value="email"/>	<input type="text" value="password"/>
<input type="text" value="Role: admin or editor?"/>		
<input type="button" value="Add User"/> <input type="button" value="Reset"/>		

User List:

- Username: sas, Email: alam1607022@stud.kuet.ac.bd,
[Delete](#)
- Username: samratalam, Email: samratalamshanto710@gmail.com,
[Delete](#)

Figure: User Page View

3. When the editor logged in, an editor will land on the ‘article’ page. The editor will see 2 options at the top. One is ‘Articles’ and the other one is ‘Log out’. An editor can post and delete articles. And articles will show at the bottom of the page. The article which will add last, will show at the top. By clicking the logout option, an editor will log out and go to the ‘log in’ page

Post Article:

Articles:

- [article3](#)
[Delete](#)
- [article2](#)
[Delete](#)
- [article 1](#)
[Delete](#)

Figure: Article Page View

Schema Diagram:

1. I use mongoDB. In the mongoDB, the user password will store as hased password using bcrypt NPM package.

test.users 2 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Displaying documents 1 - 2 of 2

	_id ObjectId	username String	email String	password String	roles String	
1	ObjectId('629cb5e4f3b3a172f68...')	"samratalam"	"samratalamshanto710@gmail.co..."	"\$2b\$10\$0k9viIRGH.0jqCYICJKeE..."	"admin"	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Duplicate"/>
2	ObjectId('629cb5f1f3b3a172f68...')	"sas"	"alam1607022@stud.kuet.ac.bd"	"\$2b\$10\$5K20wZVCC35.qyFbGwTjO..."	"editor"	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Duplicate"/>

Figure: User Collection

test.articles 3 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Displaying documents 1 - 3 of 3

	_id ObjectId	article String	date Date	__v Int32	
1	ObjectId('629b7d36a692f488aa6...')	"article 1"	2022-06-04T15:41:42.920+00:00	0	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Duplicate"/>
2	ObjectId('629c8feb7dc115c1f17...')	"article2"	2022-06-05T11:13:47.223+00:00	0	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Duplicate"/>
3	ObjectId('629c8ff27dc115c1f17...')	"articles3"	2022-06-05T11:13:54.497+00:00	0	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Duplicate"/>

Figure: Article Collection

2. Database Schema:

User	
PK	<u>user_id int NOT NULL</u>
	customer_name char(50) NOT NULL
	user_email char(50) NOT NULL
	user_password char(100) NOT NULL
	user_roles char(20) NOT NULL

Article	
PK	<u>article id int NOT NULL</u>
	article char(150) NOT NULL
	date date NOT NULL

GitHub Repo

1. Repo Link: [samratalamshanto/Projects \(github.com\)](https://github.com/samratalamshanto/Projects)
2. Deploy Link: [Log In \(newspaper-projects.herokuapp.com\)](https://newspaper-projects.herokuapp.com)

Data stored:

For admin:

Email: asd@gmail.com,

password: 123,

Role: "admin"

For editor:

Email: aaa@gmail.com,

password: 123,

Role: "editor"

(using the admin and editor data, anyone can add or delete user, post or delete article)