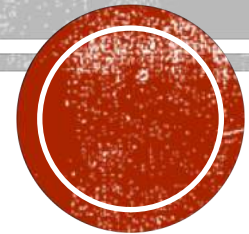


BLOGOSPHERE



- Ajay Goel (001897443)
- Ankit Yadav (001271369)

APPLICATION OBJECTIVE:

- Freedom of thoughts : Our application is based on writing your thoughts in form of posts which other users can read.
- Blogosphere is a post based website where posts are made, edited, organized by its community of users in the form of posts.
- Roles:
 - Admin can login and manage the posts along with the user management system.
 - Users can login, post, delete and edit the post.
- Connectivity: We have used Mongo DB at the backend to store the user data and posts in form of documents/collections.
 - Using Blazor application to create server and client and from server we are making the connection with the database.
 - The client hits the server which then extract data from database.



My Cluster

6 DBS 10 COLLECTIONS

filter

> EmployeeDB

> UserDB

LoginRecord

UserBostRecord

UserRecord

> admin

> adventure_game

> config

> local

localhost:27017 STANDALONE

Collections

CREATE COLLECTION

Collection Name ^	Documents	Avg. Doc
LoginRecord	2	59.0 B
UserBostRecord	2	94.5 B
UserRecord	3	97.0 B

DATABASE : MONGO DB

- Created a database:
 - UserDB
- Created 3 collections:
 - LoginRecord
 - UserBostRecord: All the posts made by the user will be stored in this collection.
 - User Record: All the user data ie., the id, password and role type will be stored in this collection .



SCREENSHOTS: POST

The screenshot shows the MongoDB Compass interface. On the left, the 'My Cluster' sidebar lists databases and collections. The main panel displays the 'UserDB.UserBostRecord' collection. The 'Documents' tab is active, showing a table of two documents. The table has columns: _id (ObjectId), Title (String), Author (String), Posted Date, and Post String. The first document has a title 'Test', author 'admin', and post 'First Blog'. The second document has a title 'Test2', author 'admin', and post 'Test2 Edited'.

MongoDB Compass Community - localhost:27017/UserDB.UserBostRecord

Connect View Collection Help

My Cluster

6 DBS 10 COLLECTIONS

filter

EmployeeDB

UserDB

LoginRecord

UserBostRecord

UserRecord

admin

adventure_game

config

local

localhost:27017 STANDALONE

MongoDB 4.0.4 Community

UserDB.UserBostRecord

DOCUMENTS 2 TOTAL SIZE 189B AVG. SIZE 95B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Explain Plan Indexes

FILTER

OPTIONS FIND RESET

INSERT DOCUMENT VIEW LIST TABLE

Displaying documents 1 - 2 of 2

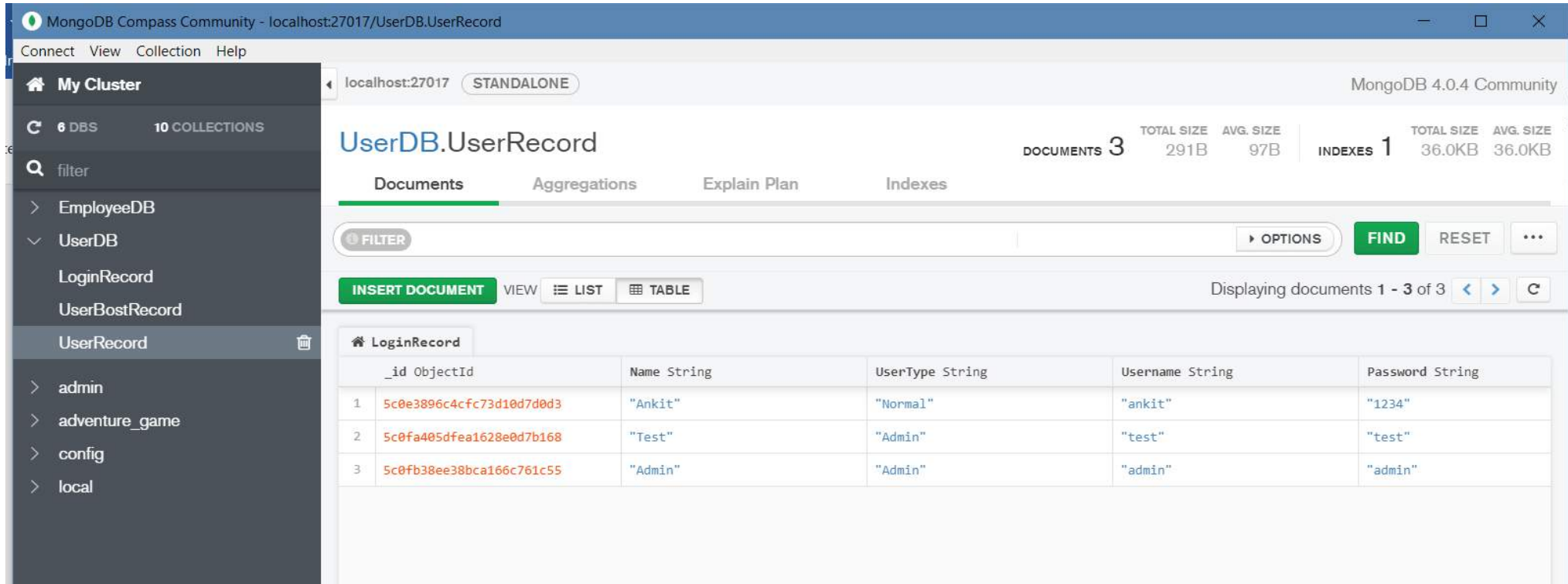
LoginRecord

	_id ObjectId	Title String	Author String	Posted Date	Post String
1	5c10d785681a5f1bc8118a18	"Test"	"admin"	2018-12-12 04:40:21.419	"First Blog"
2	5c10d88f5b1de50d4c434e1f	"Test2"	"admin"	2018-12-12 04:44:47.312	"Test2 Edited"

Collection with the user posts are stored here along with the posted date, string and title.



SCREENSHOTS : USER RECORD

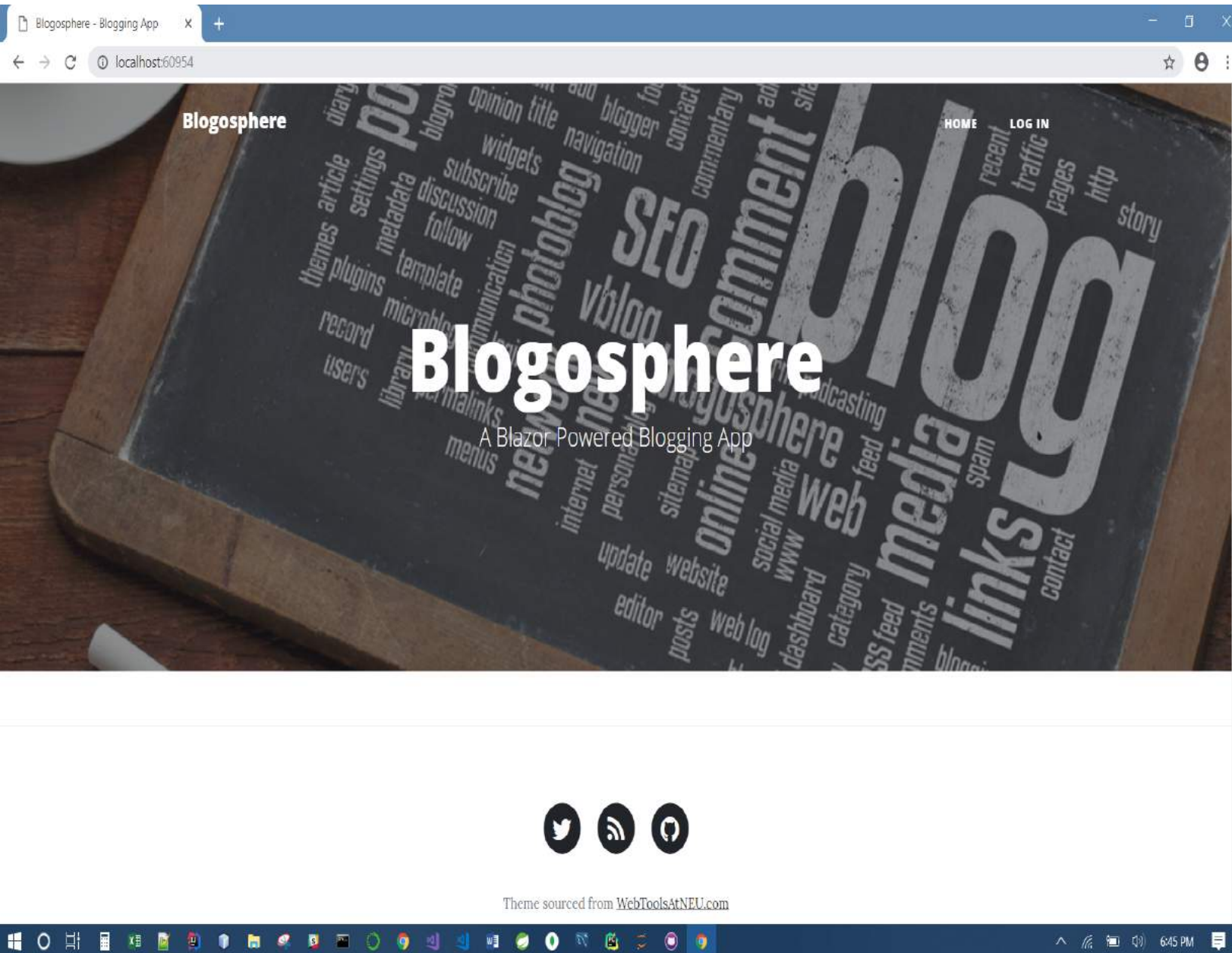


The screenshot shows the MongoDB Compass interface for the 'UserDB.UserRecord' collection. The left sidebar displays the database structure, including 'EmployeeDB', 'UserDB', and 'UserRecord'. The main panel shows the 'Documents' tab with a table of 3 documents. The table has columns: '_id ObjectId', 'Name String', 'UserType String', 'Username String', and 'Password String'. The documents are as follows:

	_id ObjectId	Name String	UserType String	Username String	Password String
1	5c0e3896c4cfc73d10d7d0d3	"Ankit"	"Normal"	"ankit"	"1234"
2	5c0fa405dfea1628e0d7b168	"Test"	"Admin"	"test"	"test"
3	5c0fb38ee38bca166c761c55	"Admin"	"Admin"	"admin"	"admin"

User records data with the name, user type, password etc are stored in this collection.





HOME - BLOGOSPHERE

- Log in button will redirect you to login page.
- Twitter label below will direct you to the twitter home page.
- Github label will redirect you to github profile already coded in the client side.



Blogosphere

Please Enter Your Login Details

admin

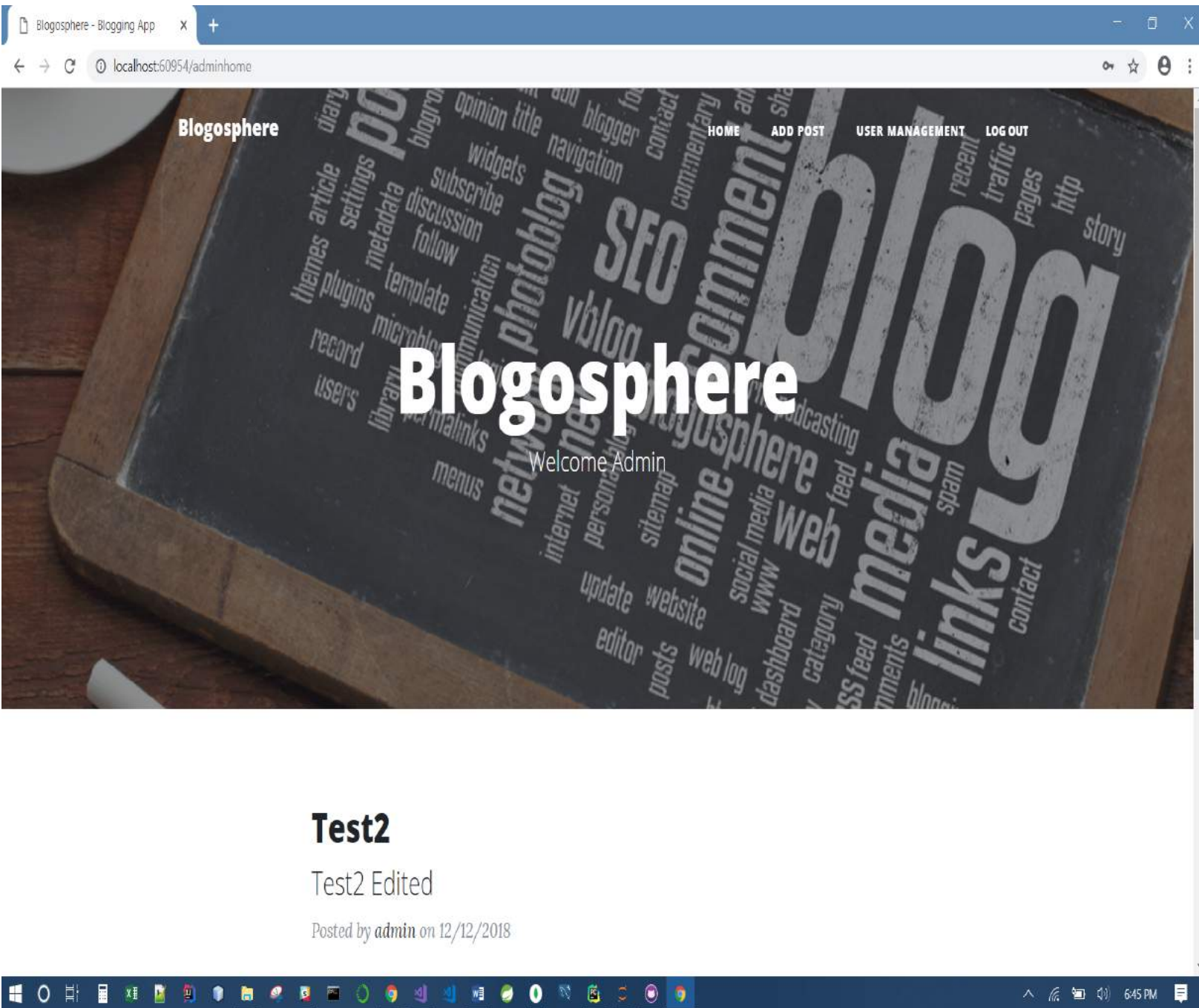
•••••

LOGIN

LOGIN PAGE

- Enter the details as admin in username and password which will take you to the admin home screen.

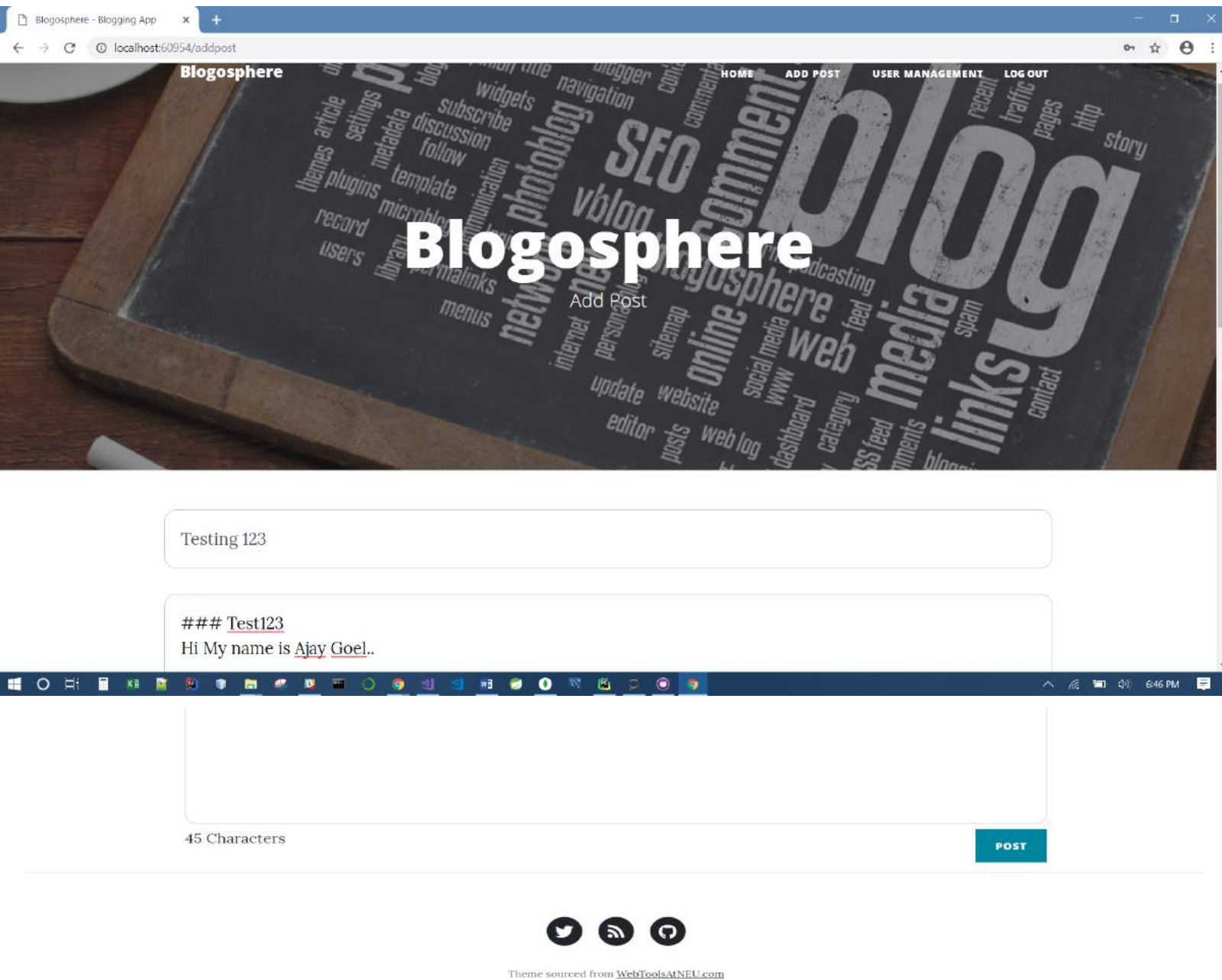




ADMIN HOME SCREEN

- Admin can add the posts by clicking on ADD POST button.
- They can log out clicking on LOG OUT button.
- User Management button will redirect the admin to the page where new users can be added or edited.
- Home: All the posts with the timestamp will be visible here.





ADDING A POST : ADD POST BUTTON

- The first text box will store the heading of the post.
- The Second textbox will store the content of the post.
- It is supported by markdown. So, we can use different sizes in the post.
- Character count is also written to check the length of the post.
- After clicking on post, data will be stored in a database.



Test2

Test2 Edited

Posted by *admin* on 12/12/2018

Test

First Blog

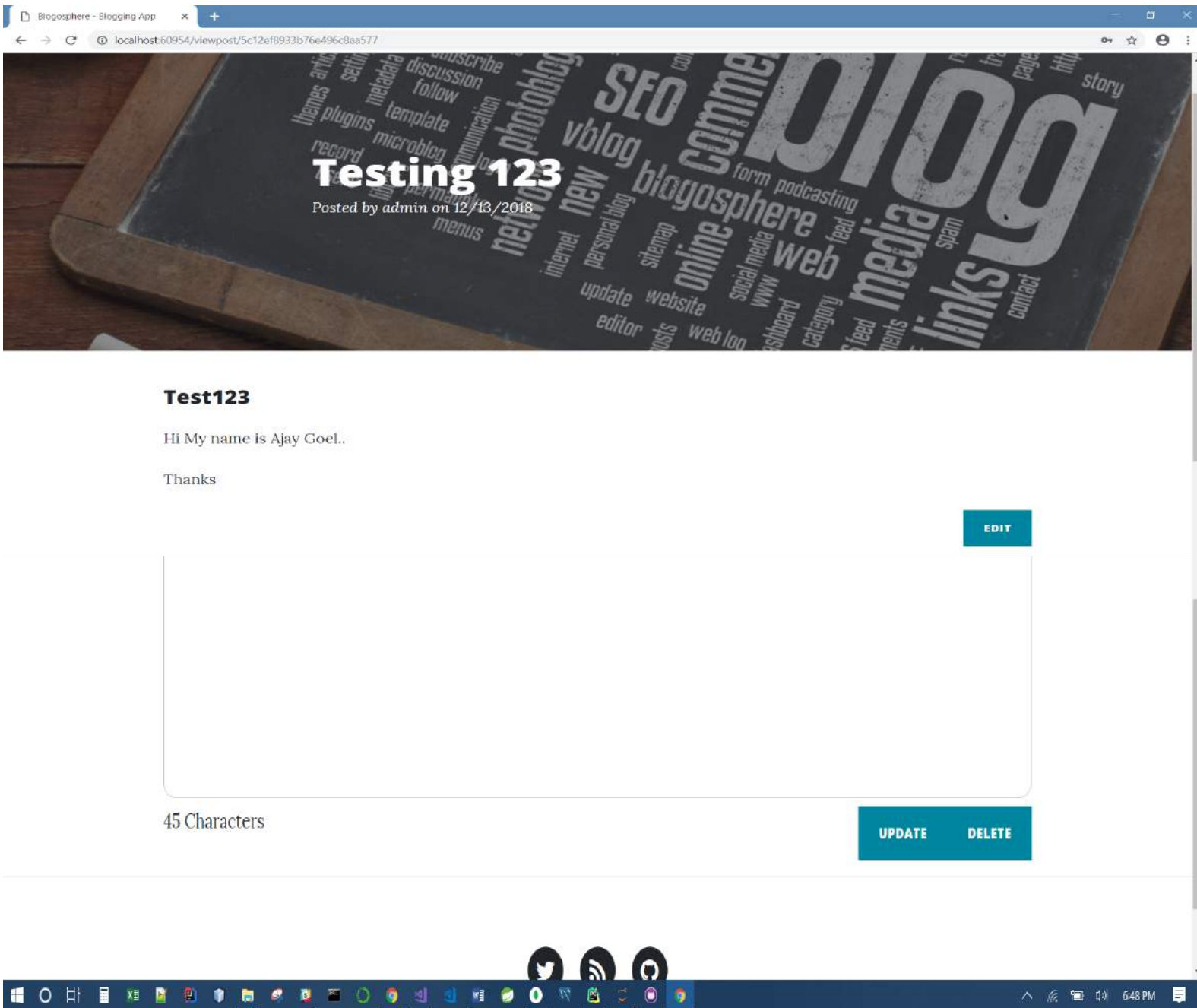
Posted by *admin* on 12/12/2018



HOME — VIEW

- We can view all the posts clicking on home button.





EDIT/DELETE POST

- We can edit the post by clicking on edit button.
- It can be deleted also.




```
_id: ObjectId("5c10d785681a5f1bc8118a18")
Title: "Test"
Author: "admin"
Posted: 2018-12-12 09:40:21.419
Post: "First Blog"
```

```
_id: ObjectId("5c10d88f5b1de50d4c434e1f")
Title: "Test2"
Author: "admin"
Posted: 2018-12-12 09:44:47.312
Post: "Test2 Edited"
```

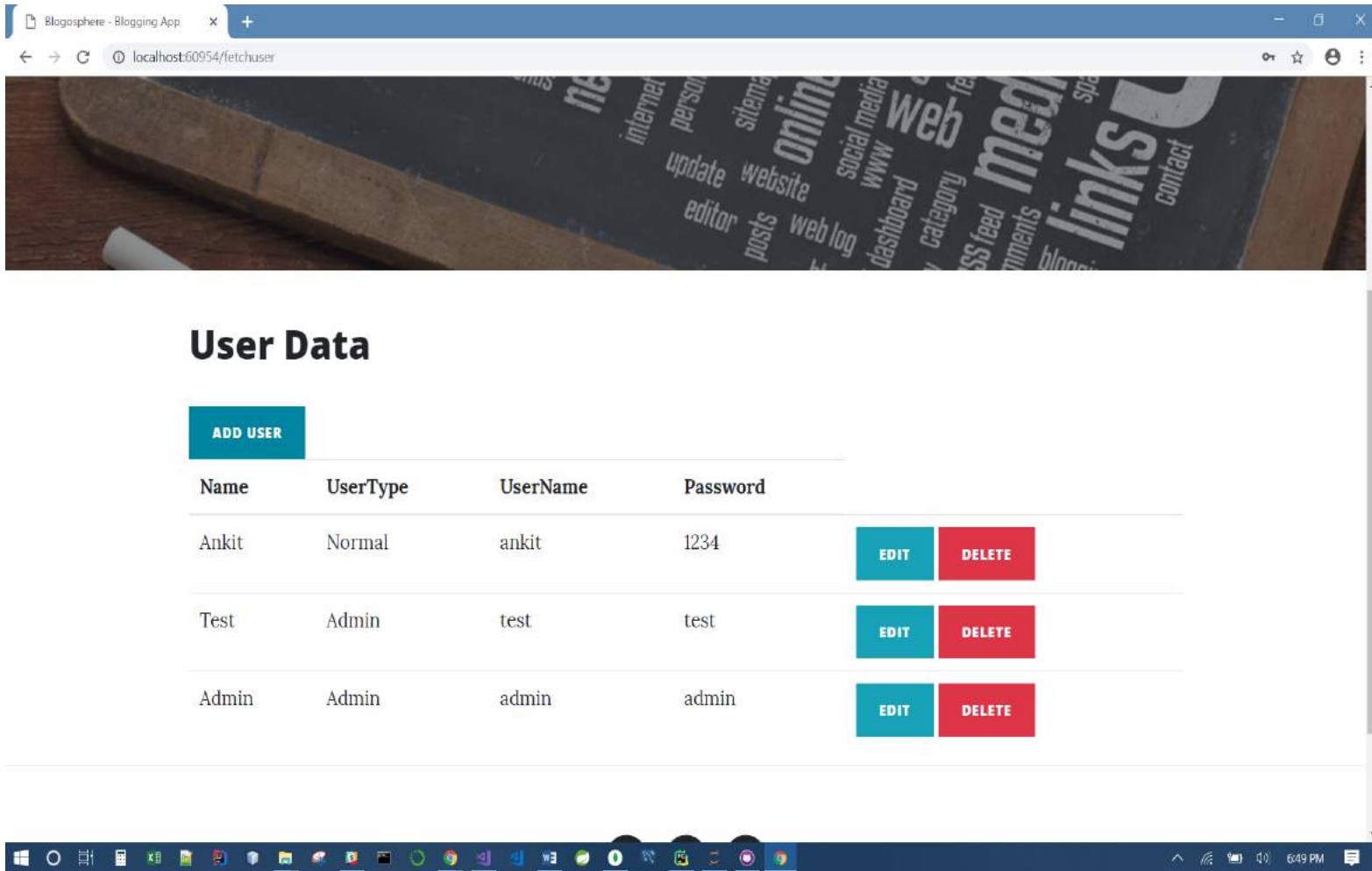
>

```
_id: ObjectId("5c12ef8933b76e496c8aa577")
UserId: null
Title: "Testing 123"
Author: "admin"
Posted: 2018-12-13 23:47:20.018
      "### Test123
Post:  Hi My name is Ajay Goel..
      Thanks"
```

DATABASE COLLECTION VIEW

- We can check if the data is stored in the database or not.





USER MANAGEMENT SYSTEM

- Users can be accessed from User Management Button.
- It can be created, deleted and updated accordingly.



Data

UserType				
Normal	ankit	1234	EDIT	DELETE
Admin	test	test	EDIT	DELETE

Add User

Name

User Type

-- Select User Type --

Username

Password

SAVE

ADD USER

- From this panel, user can be created.
- It will be directly get saved in the database.



DATABASE VIEW

INSERT DOCUMENT

VIEW

LIST

TABLE

Displaying documents 1 - 4 of 4

LoginRecord

	_id ObjectId	Name String	UserType String	Username String	Password String
1	5c0e3896c4cfc73d10d7d0d3	"Ankit"	"Normal"	"ankit"	"1234"
2	5c0fa405dfea1628e0d7b168	"Test"	"Admin"	"test"	"test"
3	5c0fb38ee38bca166c761c55	"Admin"	"Admin"	"admin"	"admin"
4	5c12f0bd33b76e496c8aa579	"robin"	"Normal"	"rob123"	"hillyard"

User Data

ADD USER

Name	UserType
Ankit	Normal
Test	Admin
Admin	Admin
Robin	Normal

EDIT X

Name

Robin

User Type

Normal

Username

RobinHillyard

Password

test

SAVE

EDIT

DELETE

EDIT

DELETE

EDIT

DELETE

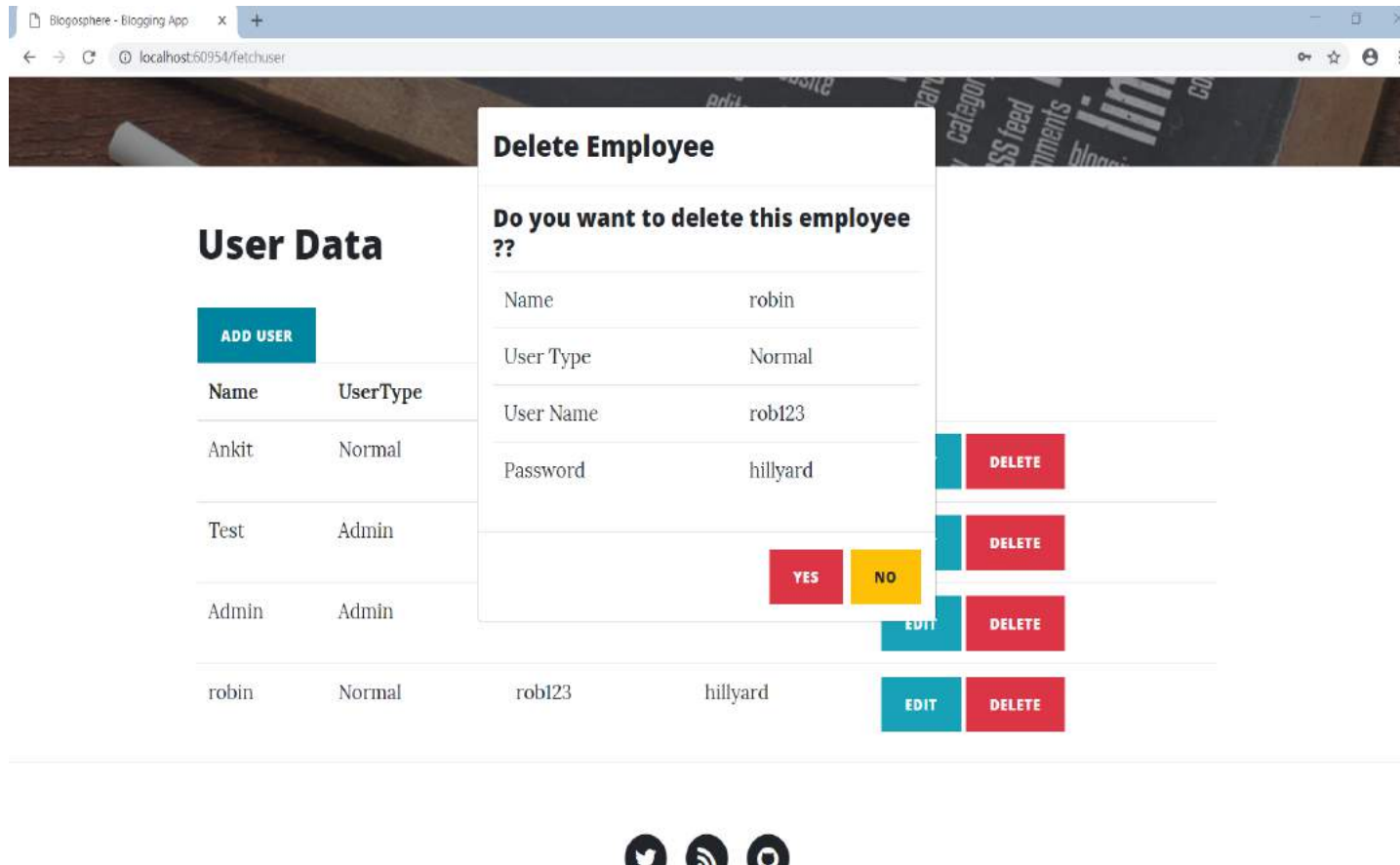
EDIT

DELETE

EDIT THE USER

- We can edit the username and password of the user by clicking on EDIT button.





DELETE THE USER

- User can be deleted from the database also.



REFERENCES:

- <https://blazor.net/>
- <https://blazor.net/docs/>
- <https://ankitsharmablogs.com/asp-net-core-crud-using-blazor-and-entity-framework-core/>



THANK YOU

