

CommentApp

Used Technologies:

UI: HTML,CSS,BootStrap,Jsp,ThemeLeaf

BackEnd: SpringBoot

DB: MySql

Flow of the Application

Sign up:

The new is user ravi@gmail.com is ready to sign up.

Sign Up

Already have an account? [Sign in](#)

E-mail:

Password:

Secret Code:

(give 6 digit numbers)

commentApp by padmesh :)

User registered (Already user is present in DB):

“You are already registered!” is shown in UI when there is already the user is present in DB

Sign Up

Already have an account? [Sign in](#)

E-mail:

Password:

Secret Code:

(give 6 digit numbers)

You are already registered!

[Sign Up](#)

commentApp by padmesh :)

User registered (success page):

After the new user is added this page is shown.

Registration done Successfully..!

[Sign in](#)

commentApp by padmesh :)

Sign in:

The newly added user is getting signed in.

Sign in

Don't have an account? [Sign Up](#)

E-mail:

Password:

Sign in

[forgot password?](#)

commentApp by padmesh :)

Comment App :

Comment App

Sign out

What would like to share with world?

Submit

E-mail	Comments
ram@gmail.com	peace..!
ram@gmail.com	save trees
padmesh@gmail.com	think +ve
padmesh@gmail.com	Good
amar@gmail.com	Bad

Filter

The specific user enters the application and see all the other users comments at first. So the logged in user is now ready to comment.

Comment App

[Sign out](#)

What would like to share with world?

[Submit](#)

E-mail	Comments
amar@gmail.com	Bad
amar@gmail.com	Believe
ravi@gmail.com	Hello Zoho..!
ravi@gmail.com	Hello World..!

[Filter](#)

The above window shows the comments of the logged in user when user clicks Filter button.

DB: (while sign in)

```
1 • SELECT * FROM commentapp_repo.users;
```

id	email	login_status	password	secret_code
1	padmesh@gmail.com	0	±02Äµ~	321900
2	amar@gmail.com	1	¾Ä²¿ÄÆÄ¶¼½	123456
3	ram@gmail.com	0	¾Ä²¿ÄÆÄ¶¼½	123456
4	ash@gmail.com	0	@Ä±³µ·¹	123456
6	ravi@gmail.com	1	¾Ä²¿ÄÆÄ¶¼½	123456
*	NULL	NULL	NULL	NULL

While the user is signing in the login_status attribute in the table is set to true;

DB : (after sign out)

```
1 • SELECT * FROM commentapp_repo.users;
```

id	email	login_status	password	secret_code
1	padmesh@gmail.com	0	±02Äµ~	321900
2	amar@gmail.com	0	¾Ä²¿ÄÆÄ¶¼½	123456
3	ram@gmail.com	0	¾Ä²¿ÄÆÄ¶¼½	123456
4	ash@gmail.com	0	@Ä±³µ·¹	123456
6	ravi@gmail.com	0	¾Ä²¿ÄÆÄ¶¼½	123456
*	NULL	NULL	NULL	NULL

When the user is signing out the login_status attribute in the table is set to false;
So by using this we are fetching the list of emails and comments of logged in users.

Forget password:

Get Password

E-mail:

Secret Code:

Get Password

[Sign In](#)

commentApp by padmesh :)

Getting Password : (success)

Get Password

E-mail:

Secret Code:

Your password is P@dmesh123

Get Password

[Sign In](#)

commentApp by padmesh :)

Getting Password : fail

Get Password

E-mail:

Secret Code:

User does not exist!

Get Password

[Sign In](#)

commentApp by padmesh :)

Encryption and Decryption password in DB:

	id	email	login_status	password	secret_code
▶	1	padmesh@gmail.com	0	±02Âµ~[]	321900

So the password given by the user is encrypted and decrypted properly at the backend.