

APDS7311 POE

KIAAN MAHARAJ

IIE VARSITY COLLEGE - DURBAN NORTH

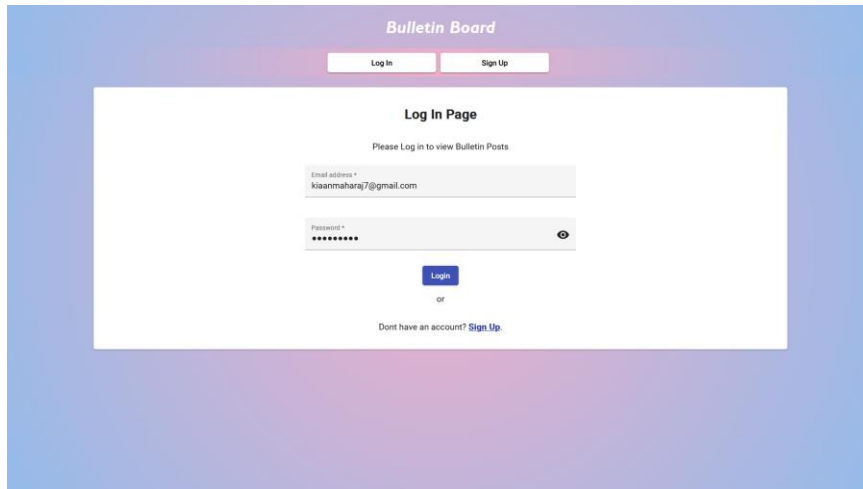
APDS7311

TABLE OF CONTENTS

Login.....	2
Error Message.....	2
Error Dialog	3
Sign Up	4
Console display	4
Error Message	5
List of departments.....	5
Successful Sign Up	6
Bulletin Posts	7
Error Handling	7
Create Post	8
Console Display	8
View Posts / Posts by different users	9
No Posts/ Delete Post	10
Log Out.....	11
Further Development.....	12
Refactor the code.....	12
Clean-up your debugging console.log	12
DOM sanitizer to sanitize (secure) your output	13
Prevent brute force attacks.....	15
Signup	15
Login.....	15
General express security – Helmet	16
Server logging.....	17
References	18

LOGIN

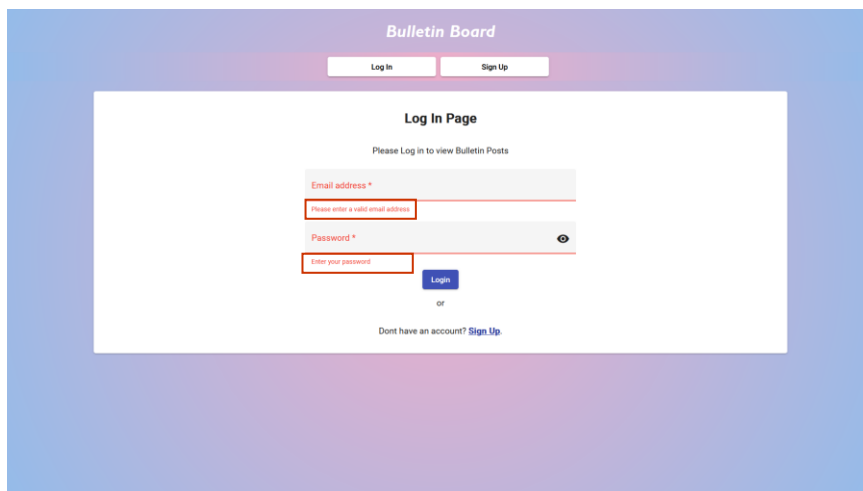
This is the login / start up page that a user will view when the web application has launched.



The screenshot shows the 'Bulletin Board' login page. At the top, there are 'Log In' and 'Sign Up' buttons. Below them is a 'Log In Page' section with the instruction 'Please Log in to view Bulletin Posts'. The form contains two input fields: 'Email address *' with the value 'kiaanmaharaj7@gmail.com' and 'Password *' with masked characters. A 'Login' button is positioned below the password field. Below the button is the text 'or' and a link 'Dont have an account? [Sign Up](#)'.

Error Message

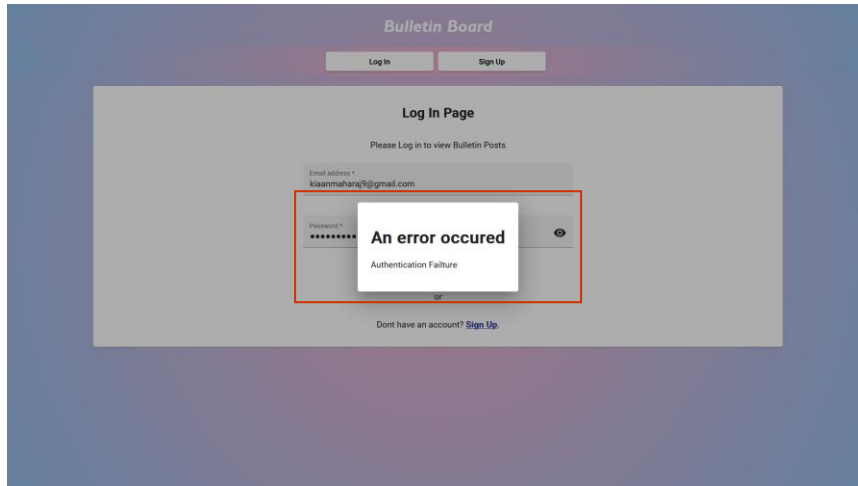
When a user does not enter the required details for the user input to read, there will be customised error messages that are displayed to the user under each of the relevant components for the user to know what needs to be corrected to proceed.



This screenshot shows the same login page as the previous one, but with error messages displayed in red boxes. The 'Email address *' field has an error message 'Please enter a valid email address'. The 'Password *' field has an error message 'Enter your password'. The 'Login' button and the 'or' text are still visible, along with the 'Dont have an account? [Sign Up](#)' link.

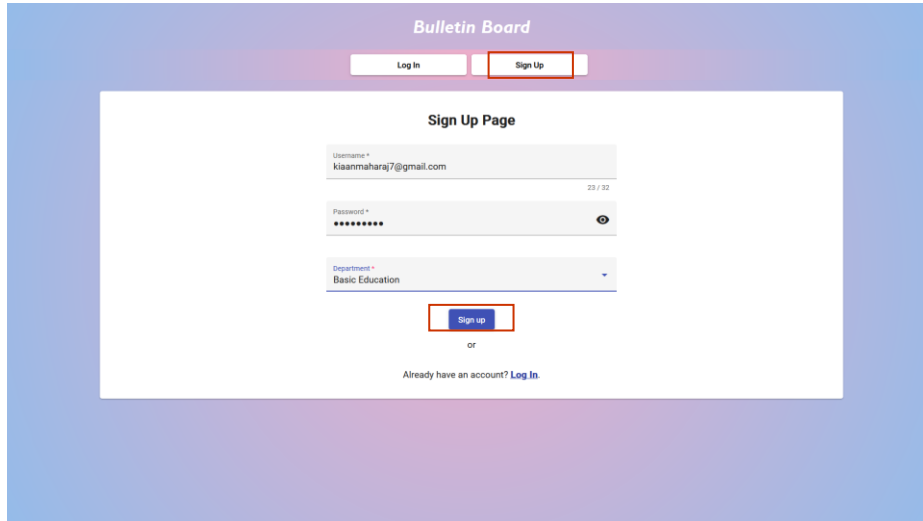
Error Dialog

If the user for example decides to tap the login button without correcting their data input even though there has been customised error messages , the second feature implemented will display an alert box to the user indicating the error.



SIGN UP

If the user does not have an account register with the bulletin board web application, the user may simply select the sign up button either on the top left of the screen or the bottom navigation link which says "sign up/login". This will allow the user to create an account which is stored to the mongo database,

A screenshot of a web application titled "Bulletin Board". At the top, there are two buttons: "Log In" and "Sign Up", with "Sign Up" highlighted by a red box. Below this is a "Sign Up Page" form. The form has three input fields: "Username" with the value "kiaanmaharaj7@gmail.com", "Password" with masked characters, and "Department" with a dropdown menu showing "Basic Education". A "Sign up" button is at the bottom of the form, also highlighted by a red box. Below the button is the text "or" and a link "Already have an account? Log In".

However, If an application already exists and the user tries to create an account again. The alert box will pop up and state that there has been an authentication failure.

```
}
posts recieved
User validation failed: username: Error, expected username to be unique.
posts recieved
Token Verified
```

Console display

This is a display using console.log to test if the user has successfully created an account. Used for testing purposes.

```
user created
{
  username: 'kiaan.maharaj@outlook.com',
  password: '$2b$10$F73G8i2NLrzoRWLLN0UgIOhtT7S.1.S.b5m51vKZDBu4d35cCtWvG',
  department: 'Military Veterans',
  _id: new ObjectId("636502b615b066fa403fcfa8"),
  __v: 0
}
```

Error Message

When a user does not enter the required details for the user input to read, there will be customised error messages that are displayed to the user under each of the relevant components for the user to know what needs to be corrected to proceed.

The screenshot shows the 'Bulletin Board' Sign Up Page. At the top, there are 'Log In' and 'Sign Up' buttons. The 'Sign Up Page' title is centered. Below it, the 'Username *' field has an error message: 'Please enter a valid username'. The 'Password *' field has an error message: 'Please enter a password that contains lowercase, uppercase letters and at least one number'. The 'Department *' dropdown menu has an error message: 'Select a Department'. A 'Sign up' button is visible, along with a link to 'Log In' for users who already have an account.

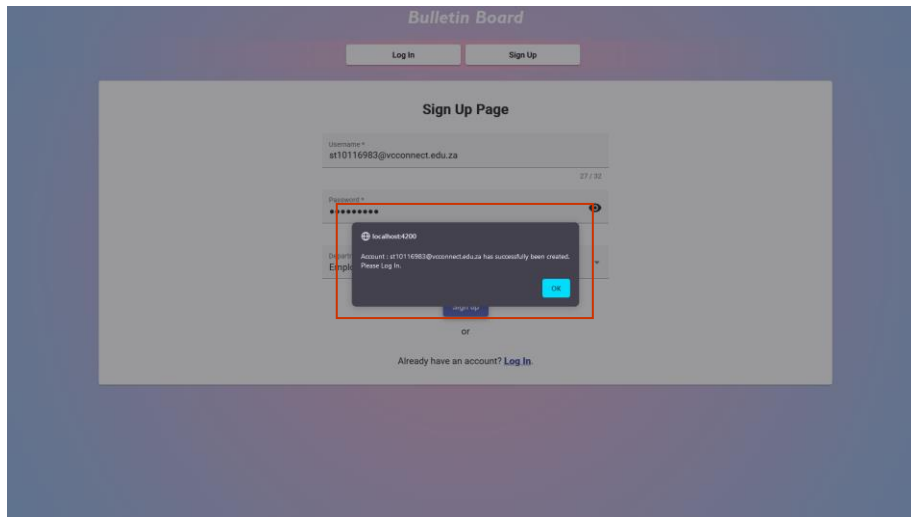
List of departments

Given that this is a South African region , Bulletin Board Web Application, there has been a list of preloaded departments in the South African Government that has been loaded for the user to select. (departments, n.d.)

The screenshot shows the 'Bulletin Board' Sign Up Page with the 'Department *' dropdown menu open. The list of departments includes: Independent Police Investigative Directorate, International Relations and Cooperation, Justice and Constitutional Development, Military Veterans, and Mineral Resources and Energy. The 'Sign up' button and the 'Log In' link are also visible.

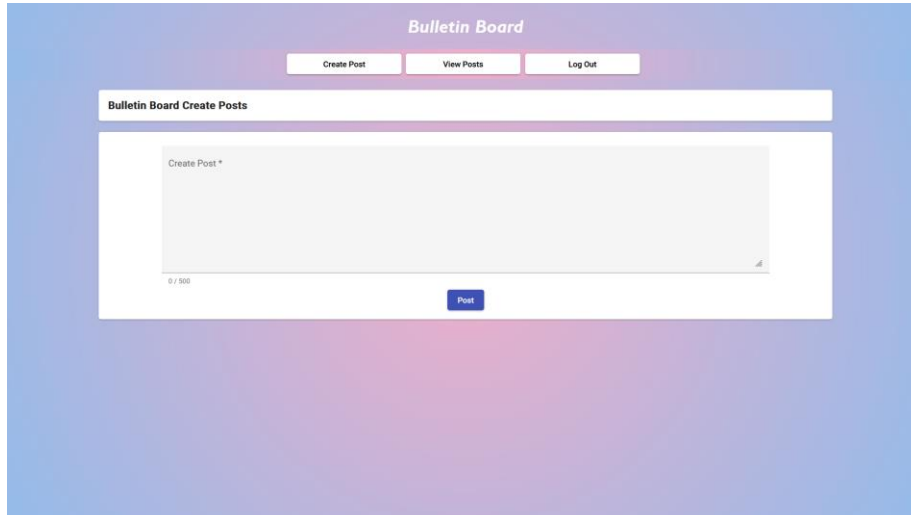
Successful Sign Up

If a user has successfully created an account, a pop up alert dialog will be displayed , confirming the account that has been created and the user will be indicated that he/she will be redirected to the login page.



BULLETIN POSTS

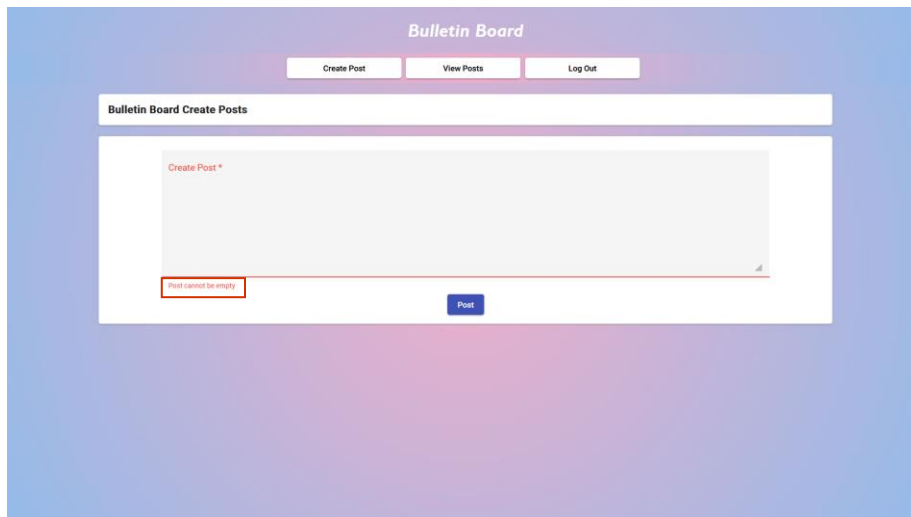
When a user has logged in successfully, the user will then be redirected to the create posts page. This page will allow the user to create bulletin posts which will be stored to the mongo database, of the user has not signed in or if their session expires, then there will be an error thrown.



The screenshot shows a web application titled "Bulletin Board". At the top, there are three buttons: "Create Post", "View Posts", and "Log Out". Below these is a section titled "Bulletin Board Create Posts". Inside this section, there is a large text input area labeled "Create Post *". Below the input area, there is a character count "0 / 500" and a "Post" button.

Error Handling

When a user does not enter the required details for the user input to read, there will be customised error messages that are displayed to the user under each of the relevant components for the user to know what needs to be corrected to proceed.



The screenshot shows the same "Bulletin Board Create Posts" page as before, but with an error message displayed. The error message, "Post cannot be empty", is shown in a red box below the text input area. The "Post" button remains visible.

Create Post

An example of how the user can create a post , and is limited to only 500 characters.

The screenshot shows a web interface for a bulletin board. At the top, there's a header with the title 'Bulletin Board' and three buttons: 'Create Post', 'View Posts', and 'Log Out'. Below this is a section titled 'Bulletin Board Create Posts'. Inside this section is a form with a text area. The text area contains the following text: 'Create Post * Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.' Below the text area, there's a character count '367 / 500' and a 'Post' button.

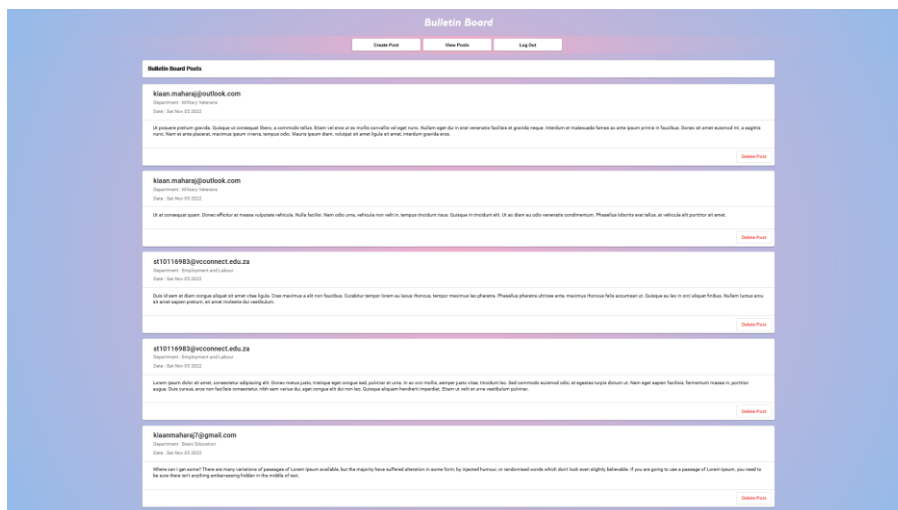
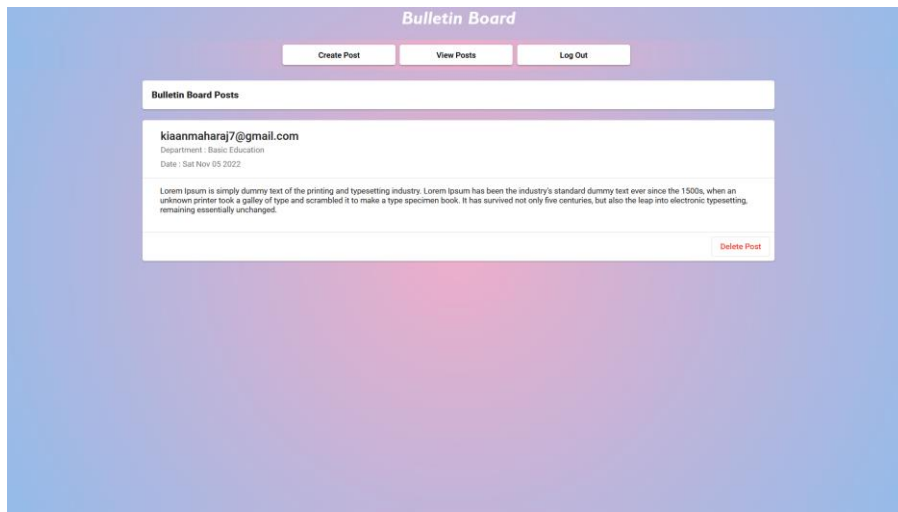
Console Display

This is a display using console.log to test if the user has successfully created a post. Used for testing purposes.

```
Token Verified
{
  username: 'kiaanmaharaj7@gmail.com',
  date: 'Sat Nov 05 2022',
  department: 'Basic Education',
  postContent: 'Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.',
  _id: new ObjectId("63664cf015b066fa403fcfce")
}
creating post ...
{
  username: 'kiaanmaharaj7@gmail.com',
  date: 'Sat Nov 05 2022',
  department: 'Basic Education',
  postContent: 'Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.',
  _id: new ObjectId("63664cf015b066fa403fcfce"),
  __v: 0
}
```

View Posts / Posts by different users

The user may also have the option to view all posts that has been created by other users that have successfully signed up and created posts as well.



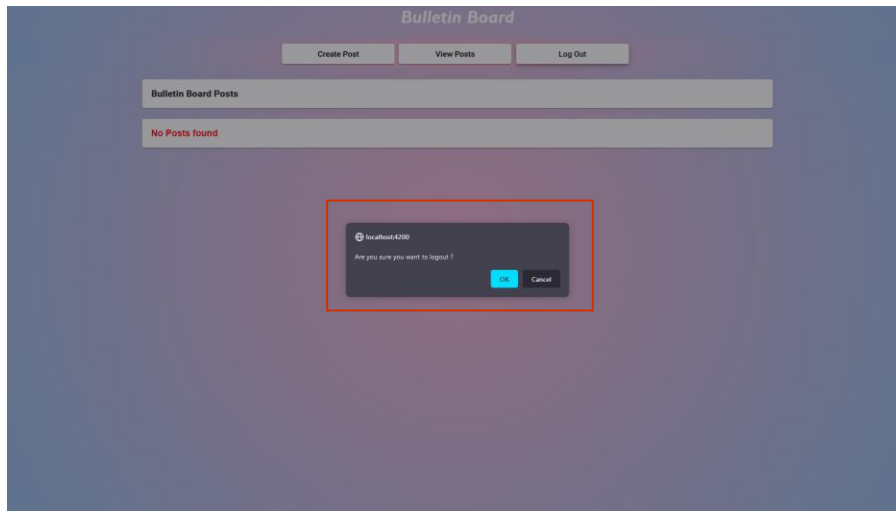
No Posts/ Delete Post

If there are no posts found or if a user deletes all the posts, a message will be displayed to the user.



LOG OUT

If the user wishes to end their session or log out a confirm dialog will be displayed to the user to confirm their action.



FURTHER DEVELOPMENT

Refactor the code.

Please check source code.

Clean-up your debugging console.log

```
posts deleted from the database
Token Verified
{
  username: 'kiaanmaharaj7@gmail.com',
  date: 'Sat Nov 05 2022',
  department: 'Basic Education',
  postContent: 'What is Lorem Ipsum? Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.',
  _id: new ObjectId("63664ef15b066fa403fcfe1")
}
creating post ...
{
  username: 'kiaanmaharaj7@gmail.com',
  date: 'Sat Nov 05 2022',
  department: 'Basic Education',
  postContent: 'What is Lorem Ipsum? Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.',
  _id: new ObjectId("63664ef15b066fa403fcfe1"),
  __v: 0
}
posts recieved
Token Verified
{
  username: 'kiaanmaharaj7@gmail.com',
  date: 'Sat Nov 05 2022',
  department: 'Basic Education',
  postContent: 'Why do we use it? It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English.',
  _id: new ObjectId("63664fd415b066fa403fcfe4")
}
creating post ...
{
  username: 'kiaanmaharaj7@gmail.com',
  date: 'Sat Nov 05 2022',
  department: 'Basic Education',
  postContent: 'Why do we use it? It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English.',
  _id: new ObjectId("63664fd415b066fa403fcfe4"),
  __v: 0
}
posts recieved
Token Verified
{
  username: 'kiaanmaharaj7@gmail.com',
  date: 'Sat Nov 05 2022',
  department: 'Basic Education',
  postContent: 'Where does it come from? Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1',
  _id: new ObjectId("63664fe115b066fa403fcfe7")
}
creating post ...
{
  username: 'kiaanmaharaj7@gmail.com',
  date: 'Sat Nov 05 2022',
  department: 'Basic Education',
  postContent: 'Where does it come from? Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1',
  _id: new ObjectId("63664fe115b066fa403fcfe7"),
  __v: 0
}
```

DOM sanitizer to sanitize (secure) your output

```
You, 2 days ago | 1 author (You)
1 import { Component, OnInit, SecurityContext } from '@angular/core';
2 // add Forms import to use NgForm
3 import { NgForm } from '@angular/forms';
4 // Import service
5 import { PostServiceService } from '../post-service.service';
6
7 import { DomSanitizer } from '@angular/platform-browser';
8
9 import { Subscription } from 'rxjs';
10 import { AuthServiceService } from '../../auth/auth-service.service';
11 You, 2 days ago | 1 author (You)
12 @Component({
13   selector: 'app-post-create',
14   templateUrl: './post-create.component.html',
15   styleUrls: ['./post-create.component.css']
16 })
17 export class PostCreateComponent implements OnInit {
18
19   constructor(public postService: PostServiceService, protected sanitizer: DomSanitizer, public authService: AuthServiceService) {}
20
21   postError: string = 'Post cannot be empty';
22
23   private loginSub: Subscription = new Subscription();
24   isLoggedIn: boolean = false;
25
26   ngOnInit(): void {
27     // You, last week * added components and services
28     this.isLoggedIn = this.authService.checklogin();
29   }
30
31   onAddPost(postForm: NgForm) {
32     if (postForm.invalid) {
33       return;
34     }
35     // http post
36     this.postService.addPostService(new Date().toDatestring(), this.sanitizer.sanitize(SecurityContext.HTML, postForm.value.postContent))
37     postForm.resetForm()
38   }
39 }
```

```
You, 2 days ago | 1 author (You)
1 import { Component, OnInit, SecurityContext } from '@angular/core';
2 import { NgForm } from '@angular/forms';
3 import { ActivatedRoute, Router } from '@angular/router';
4 import { AuthServiceService } from '../../app/auth/auth-service.service';
5 import { DomSanitizer } from '@angular/platform-browser';
6
7 You, 2 days ago | 1 author (You)
8 @Component({
9   selector: 'app-login',
10   templateUrl: './login.component.html',
11   styleUrls: ['./login.component.css']
12 })
13 export class LoginComponent implements OnInit {
14
15   constructor(public authService: AuthServiceService, private router: Router, protected sanitizer: DomSanitizer) {}
16
17   emailError: string = 'Please enter a valid email address';
18   passwordError: string = 'Enter your password';
19
20   public showPassword: boolean = false;
21
22   public togglePasswordVisibility(): void {
23     this.showPassword = !this.showPassword;
24   }
25
26   ngOnInit(): void {
27   }
28
29   onlogin(form: NgForm) {
30     if (form.invalid) {
31       return;
32     } else {
33       this.authService.login(
34         this.sanitizer.sanitize(SecurityContext.HTML, form.value.enterusername),
35         this.sanitizer.sanitize(SecurityContext.HTML, form.value.enterpassword)
36       );
37     }
38   }
39
40   /*
41   Witish Kaushik
42   Create Show / Hide password in Angular with Angular Material
43   https://nitishkaushik.com/show-hide-password-in-angular-with-angular-material/
44   November 13, 2021
45   You, 2 days ago * code attribution
```

```

You, yesterday | 1 author (You)
1 import { Component, OnInit, SecurityContext } from '@angular/core';
2 import { NgForm } from '@angular/forms';
3 import { ActivatedRoute, Router } from '@angular/router';
4 import { AuthServiceService } from '.../app/auth/auth-service.service';
5 import { DomSanitizer } from '@angular/platform-browser';
You, yesterday | 1 author (You)
6 @Component({
7   selector: 'app-signup',
8   templateUrl: './signup.component.html',
9   styleUrls: ['./signup.component.css']
10 })
11 export class SignupComponent implements OnInit {
12
13   usernameError: string = 'Please enter a valid username';
14   passwordError: string = 'Please enter a password that contains lowercase, uppercase letters and at least one number';
15   departmentError: string = 'Please Select a Department';
16
17   constructor(public authService: AuthServiceService, private router: Router, private sanitizer: DomSanitizer) { }
18
19
20   public showPassword: boolean = false;
21
22   public togglePasswordVisibility(): void {
23     this.showPassword = !this.showPassword;
24   }
25
26   ngOnInit(): void {
27   }
28
29   onsignup(signupform: NgForm) {
30     if (signupform.invalid) {
31       return;
32     } else {
33
34       this.authService.signup(
35         this.sanitizer.sanitize(SecurityContext.HTML, signupform.value.enterusername),
36         this.sanitizer.sanitize(SecurityContext.HTML, signupform.value.enterpassword),
37         this.sanitizer.sanitize(SecurityContext.HTML, signupform.value.enteredDepartment)
38       );
39
40       alert('Account : ' + signupform.value.enterusername + ' has successfully been created.\nPlease Log In. ');
41     }
42   }
43 }
44
45 /*
46 Nitish Kaushik
47 Create Show / Hide password in Angular with Angular Material
48 https://nitishkaushik.com/show-hide-password-in-angular-with-angular-material/
49 November 13, 2021
50 */

```

Prevent brute force attacks

SIGNUP

```
14 // brute force protection
15 const ExpressBrute = require("express-brute");
16 const store = new ExpressBrute.MemoryStore();
17 const bruteforce = new ExpressBrute(store);
18
19 // post method used for registering user
20 router.post("/signup", bruteforce.prevent, (req, res) => {
21   // hashing the users password
22   bcrypt.hash(req.body.password, 10).then((hash) => {
23     const user = new User({
24       username: req.body.username,
25       password: hash,
26       department: req.body.department,
27     });
```

LOGIN

```
47 // post method used for user login
48 router.post("/login", bruteforce.prevent, (req, res) => {
49   let fetchedUser;
50
51   // checks the database to see if the username exists
52   User.findOne({ username: req.body.username })
53     .then((user) => {
54       if (!user) {
55         return res.status(401).json({
56           message: "Authentication Failure",
57         });
58       }
```


General express security – Helmet

```
4  
5 //helmet  
6 const helmet = require("helmet");
```

```
38  
39 app.use(helmet());  
40
```

Server logging

```
//Logs requests to a log file
//app.use(morgan("DATE -> :date[clf]\t| METHOD -> :method| URL -> :url\t| STATUS -> :status\t| RESPONSE TIME -> :response-time ms\t|BODY -> :tbody", { stream: accesslogStream}));
//Logs requests
app.use(
  morgan(
    "REQ\t| DATE -> :date[clf]\t| METHOD -> :method| URL -> :url\t| STATUS -> :status\t| RESPONSE TIME -> :response-time ms\t|BODY -> :tbody",
    {
      immediate: true,
      stream: accesslogStream,
    }
  )
);

// Logs responses
app.use(
  morgan(
    "RES\t| DATE -> :date[clf]\t| METHOD -> :method| URL -> :url\t| STATUS -> :status\t| RESPONSE TIME -> :response-time ms",
    {
      stream: accesslogStream,
    }
  )
);
```

[illegible]

REFERENCES

[Brute], N., n.d. [Online]

Available at: <https://www.npmjs.com/package/express-brute>

[Accessed September 2022].

departments, N., n.d. [Online]

Available at: <https://www.gov.za/about-government/government-system/national-departments>

[Accessed 2 November 2022].

Manager, N. P., n.d. [Online]

Available at: <https://docs.npmjs.com>

[Accessed September 2022].

Marketplace, V. S. C., n.d. [Online]

Available at: <https://code.visualstudio.com/docs/editor/extension-marketplace>

[Accessed September 2022].

Mozilla, n.d. [Online]

Available at: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

[Accessed September 2022].

Nodemon, n.d. [Online]

Available at: <https://www.npmjs.com/package/nodemon>

[Accessed September 2022].

Validator], N. [, n.d. [Online]

Available at: <https://www.npmjs.com/package/mongoose-unique-validator>

[Accessed September 2022].

W3Schools, n.d. [Online]

Available at: <https://www.w3schools.com/nodejs/default.asp>

[Accessed September 2022].