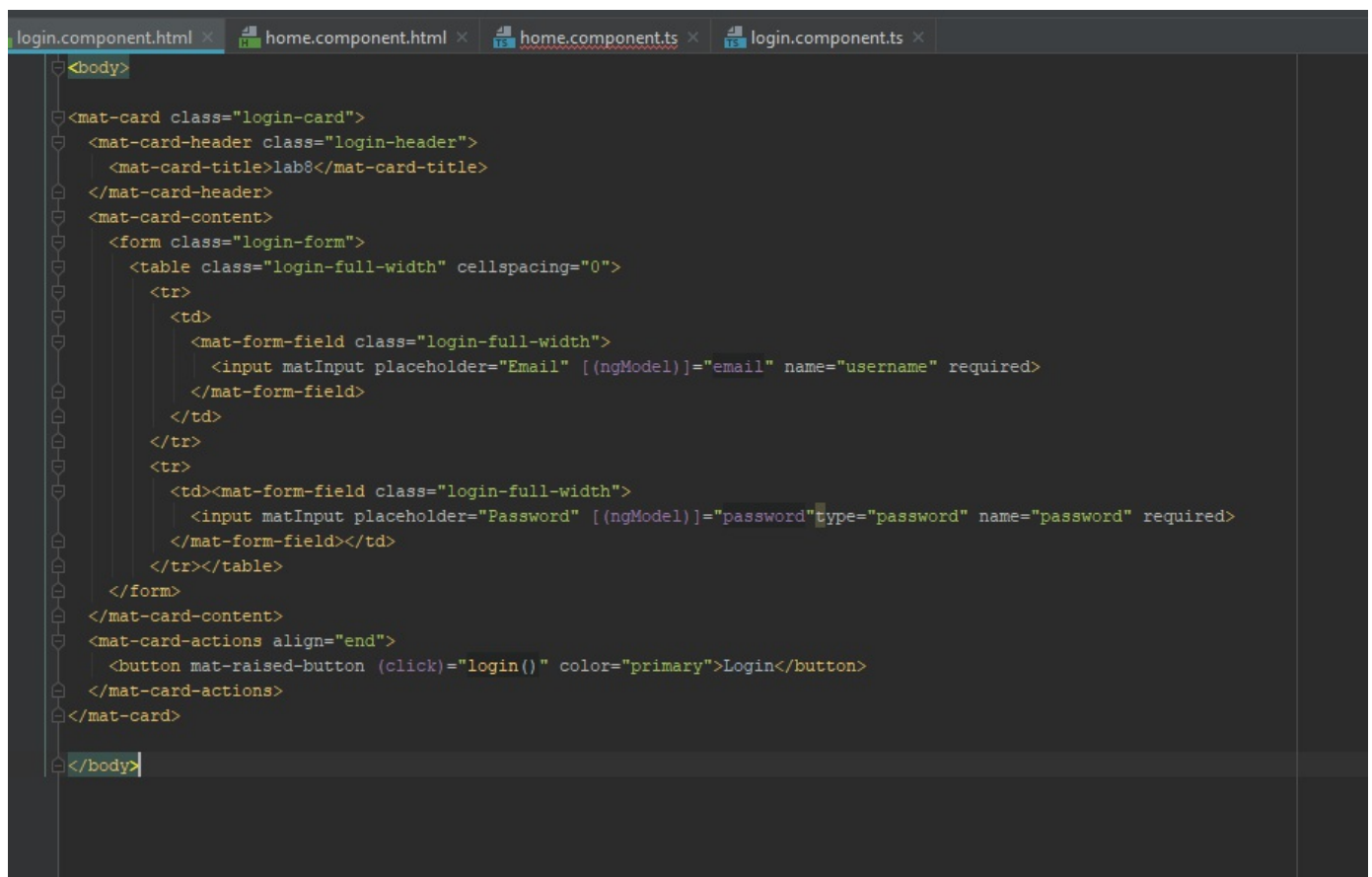


# Objective

Understand the lab-8 source code and write an app to perform user authentication (email, password) using JWT token.

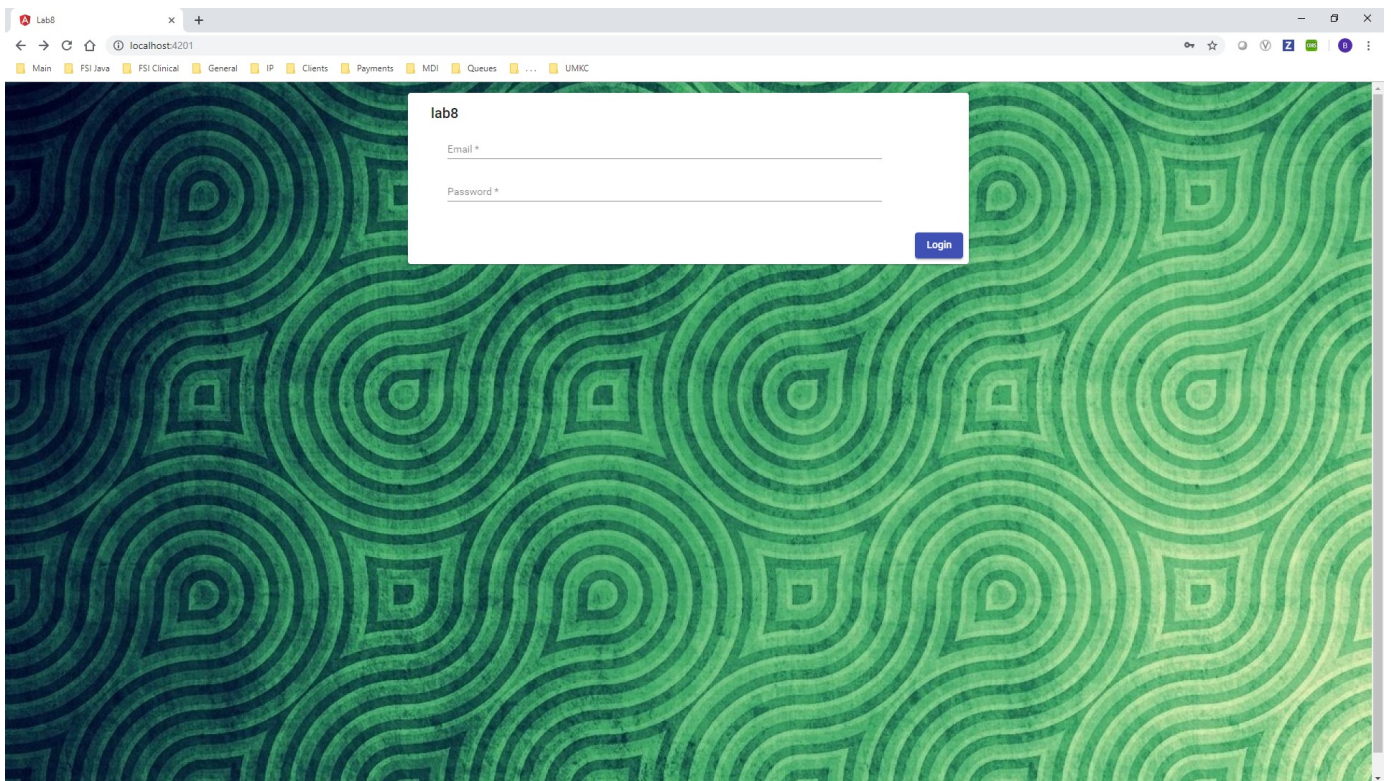
## Login

This page features two input boxes for email address and password. Upon login, the JWT token is stored in local storage using a post request to the URL <http://localhost:3000/api/login>. Once submitted, the app navigates to the home page:



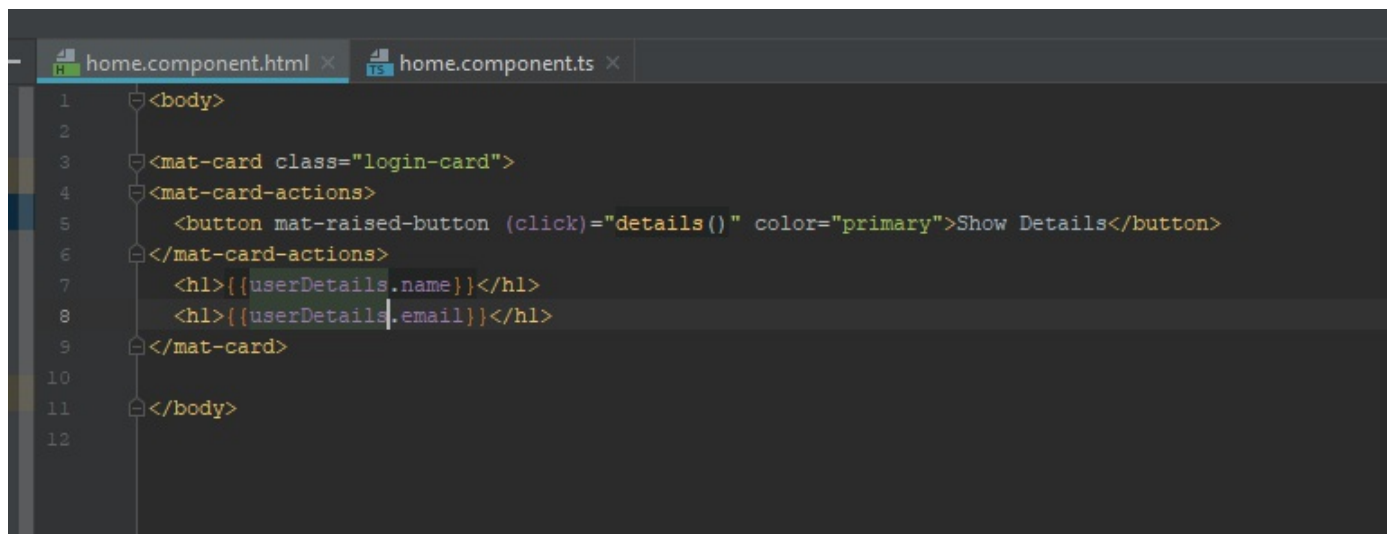
```
<body>
<mat-card class="login-card">
  <mat-card-header class="login-header">
    <mat-card-title>lab8</mat-card-title>
  </mat-card-header>
  <mat-card-content>
    <form class="login-form">
      <table class="login-full-width" cellspacing="0">
        <tr>
          <td>
            <mat-form-field class="login-full-width">
              <input matInput placeholder="Email" [(ngModel)]="email" name="username" required>
            </mat-form-field>
          </td>
        </tr>
        <tr>
          <td><mat-form-field class="login-full-width">
            <input matInput placeholder="Password" [(ngModel)]="password" type="password" name="password" required>
          </mat-form-field></td>
        </tr></table>
      </form>
    </mat-card-content>
    <mat-card-actions align="end">
      <button mat-raised-button (click)="login()" color="primary">Login</button>
    </mat-card-actions>
  </mat-card>
</body>
```

```
home.component.html × home.component.ts × login.component.ts ×
1 import { Component, OnInit } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Router } from '@angular/router';
4
5 @Component({
6   selector: 'app-login',
7   templateUrl: './login.component.html',
8   styleUrls: ['./login.component.css']
9 })
10 export class LoginComponent implements OnInit {
11
12   constructor(private http: HttpClient, private router: Router) { }
13
14   email: string;
15   password: string;
16
17   ngOnInit() {
18   }
19
20   login() {
21     if (this.email !== '' && this.password !== '') {
22       this.http.post( url: 'http://localhost:3000/api/login', body: {name: this.email, email: this.password})
23         .subscribe( next: (data: any) => {
24           localStorage.setItem('auth_token', data.token);
25           this.router.navigate( commands: ['/home']);
26           console.log(data.token);
27         });
28     } else {
29       alert('Invalid credentials');
30     }
31   }
32 }
33
```



# Home

This page features a Show Details button. When clicked, the token is pulled from local storage a get request is sent to URL <http://localhost:3000/api/posts>. Then, the returned details (name, email) are populated on the page:

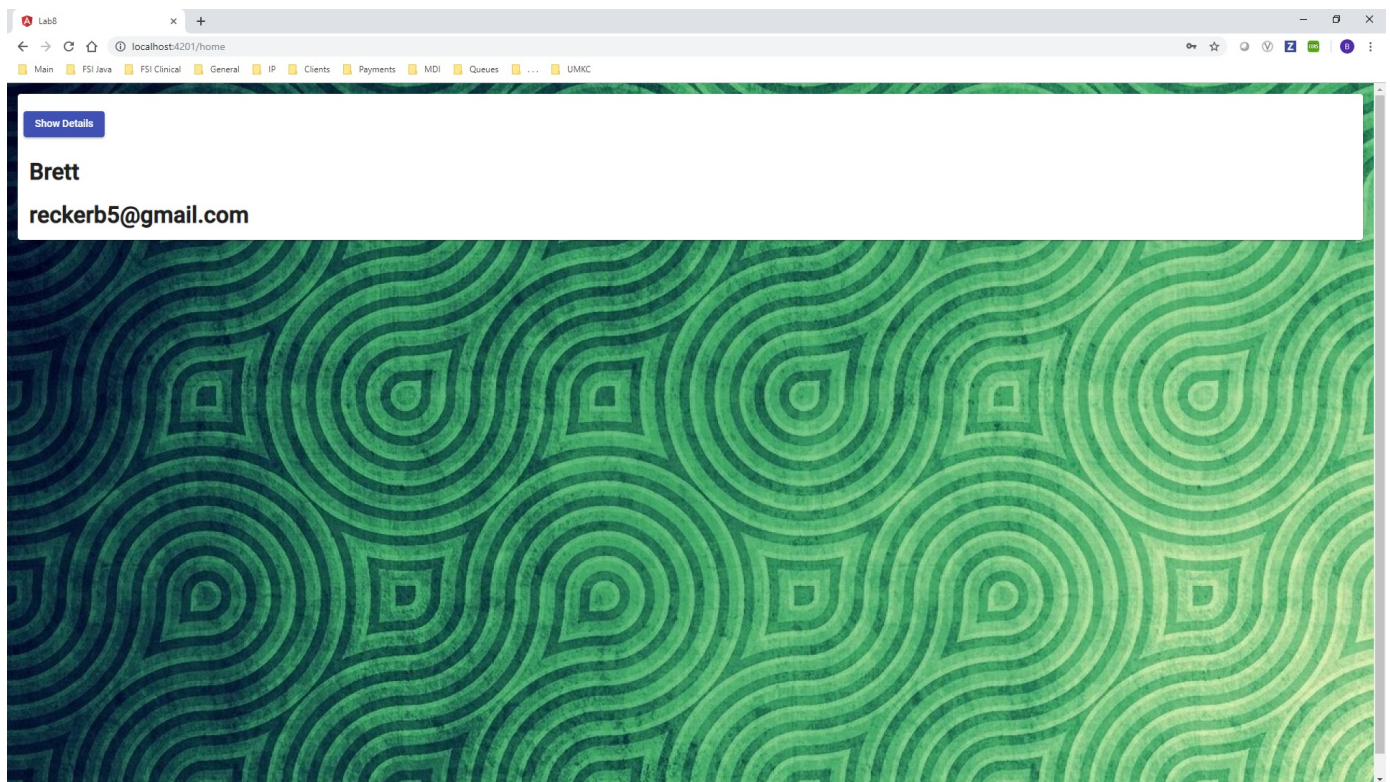
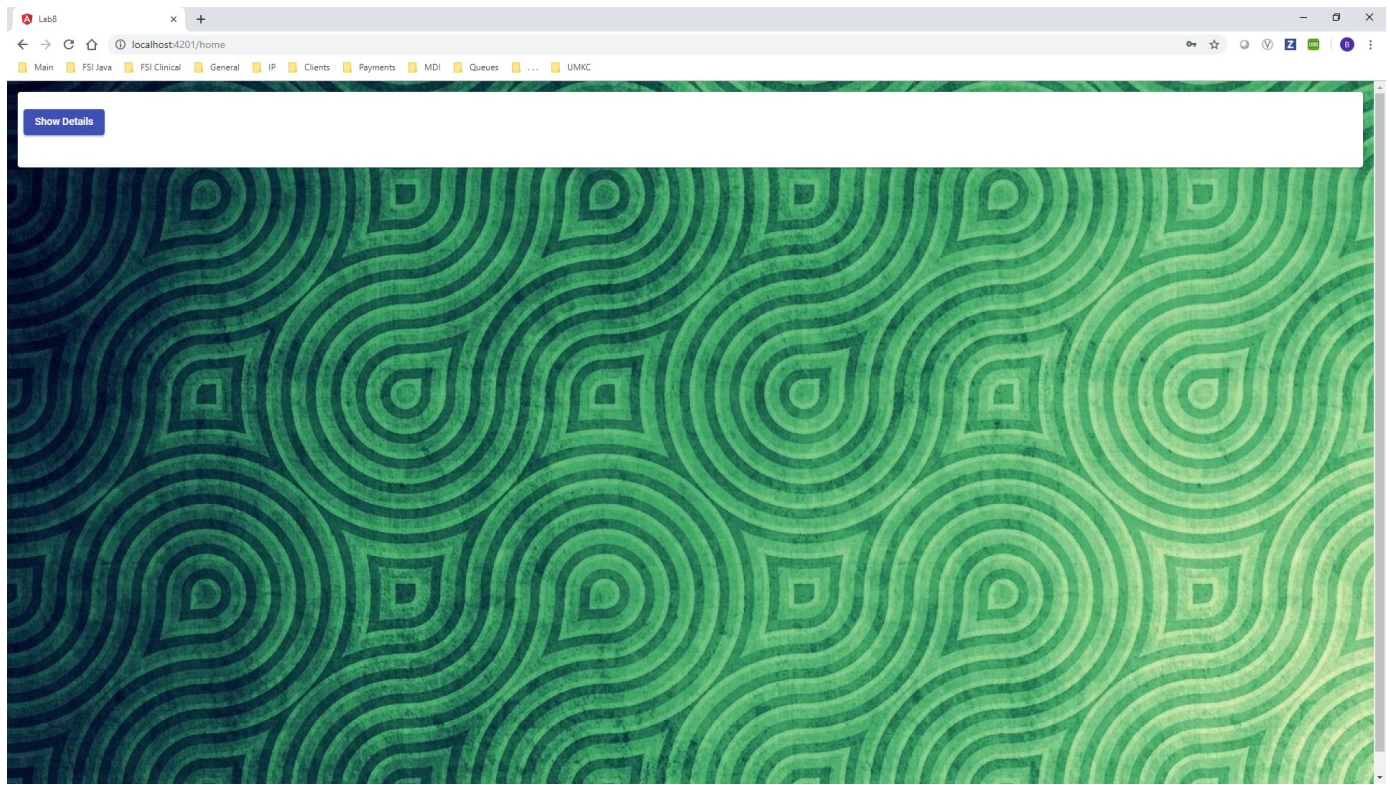


```
1 <body>
2
3 <mat-card class="login-card">
4   <mat-card-actions>
5     <button mat-raised-button (click)="details()" color="primary">Show Details</button>
6   </mat-card-actions>
7   <h1>{{userDetails.name}}</h1>
8   <h1>{{userDetails.email}}</h1>
9 </mat-card>
10
11 </body>
12
```

home.component.ts ×

```
1 import { Component, OnInit } from '@angular/core';
2 import { HttpClient, HttpHeaders } from '@angular/common/http';
3
4 @Component({
5   selector: 'app-home',
6   templateUrl: './home.component.html',
7   styleUrls: ['./home.component.css']
8 })
9 export class HomeComponent implements OnInit {
10
11   userDetails: any;
12
13   constructor(private http: HttpClient) { }
14
15   ngOnInit() {
16   }
17
18   details() {
19     const token = localStorage.getItem( key: 'auth_token' );
20     console.log(token);
21     const headers = new HttpHeaders().set( 'auth_token', token );
22     this.http.get( url: 'http://localhost:3000/api/posts', options: {headers})
23       .subscribe( next: response => {
24         this.userDetails = response.user;
25         console.log(response);
26       });
27   }
28 }
29
```





## Outcome

This was completely new to me and I will need to continue working with tokens to fully understand this workflow before implementing in our

project.