Login code with CSRF token in Angular 14, Spring Boot and PostgreSQL

To implement a login system with CSRF token using Angular 14, Spring Boot, and PostgreSQL, you can follow the steps outlined below:

1. Create a new Angular project using the Angular CLI:

```arduino
ng new angular-login-app
```

1. Install the `@angular/material`, `@angular/flex-layout`, and `@angular/forms` packages:

```bash
npm install @angular/material @angular/flex-layout @angular/forms
```

1. Create a login component:

```
ng generate component login
```

1. In the login component, create a form with email and password fields:

```html
<form [formGroup]="loginForm" (ngSubmit)="onSubmit()">
  <mat-form-field>
    <input matInput placeholder="Email" formControlName="email">
  </mat-form-field>

  <mat-form-field>
    <input matInput placeholder="Password" type="password" formControlName="
  </mat-form-field>

  <button mat-raised-button color="primary" type="submit">Login</button>
</form>
```

1. In the login component class, create a `FormGroup` for the login form:

```typescript
import { Component, OnInit } from '@angular/core';
import { FormGroup, FormControl, Validators } from '@angular/forms';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
  loginForm: FormGroup;

  ngOnInit() {
    this.loginForm = new FormGroup({
      email: new FormControl('', [Validators.required, Validators.email]),
      password: new FormControl('', Validators.required)
    });
  }

  onSubmit() {
    // handle login form submission
  }
}
```

1. Install the `ngx-cookie-service` package to manage cookies:

```
npm install ngx-cookie-service
```

1. In the login component class, inject the `CookieService`:

```typescript
import { CookieService } from 'ngx-cookie-service';

constructor(private cookieService: CookieService) { }
```

1. When the login form is submitted, use the `HttpClient` to send a POST request to the Spring Boot backend:

```typescript
import { HttpClient } from '@angular/common/http';

constructor(private http: HttpClient) { }

onSubmit() {
  const credentials = {
    email: this.loginForm.get('email').value,
    password: this.loginForm.get('password').value
  };

  this.http.post('/api/login', credentials, { withCredentials: true }).subsc
    // handle successful login
  });
}
```

1. In the Spring Boot backend, create a `UserController` with a `/login` endpoint that generates a CSRF token:

```java
@RestController
public class UserController {
  @Autowired
  private CsrfTokenRepository csrfTokenRepository;

  @PostMapping("/login")
  public ResponseEntity<?> login(@RequestBody Credentials credentials, HttpS
    // authenticate user
    // generate CSRF token
    CsrfToken csrfToken = this.csrfTokenRepository.generateToken(request);
    this.csrfTokenRepository.saveToken(csrfToken, request, response);

    return ResponseEntity.ok().build();
  }
}
```

1. In the Angular app, use the `HttpClient` interceptor to add the CSRF token to all outgoing requests:

```typescript
import { Injectable } from '@angular/core';
import { HttpInterceptor, HttpRequest, HttpHandler, HttpEvent } from '@angul
import { Observable } from 'rxjs';
import { CookieService } from 'ngx-cookie-service';

@Injectable()
export class CsrfInterceptor implements HttpInterceptor {
  constructor(private cookieService: CookieService) { }

  intercept(request: HttpRequest<any>, next: HttpHandler
```