

DOKUMENTÁCIÓ

Web technológiák II.

Féléves feladat

Készítette: **Szabó Alexandra Marianna**

Neptunkód: **R65UKG**

Dátum: 2024.06.13

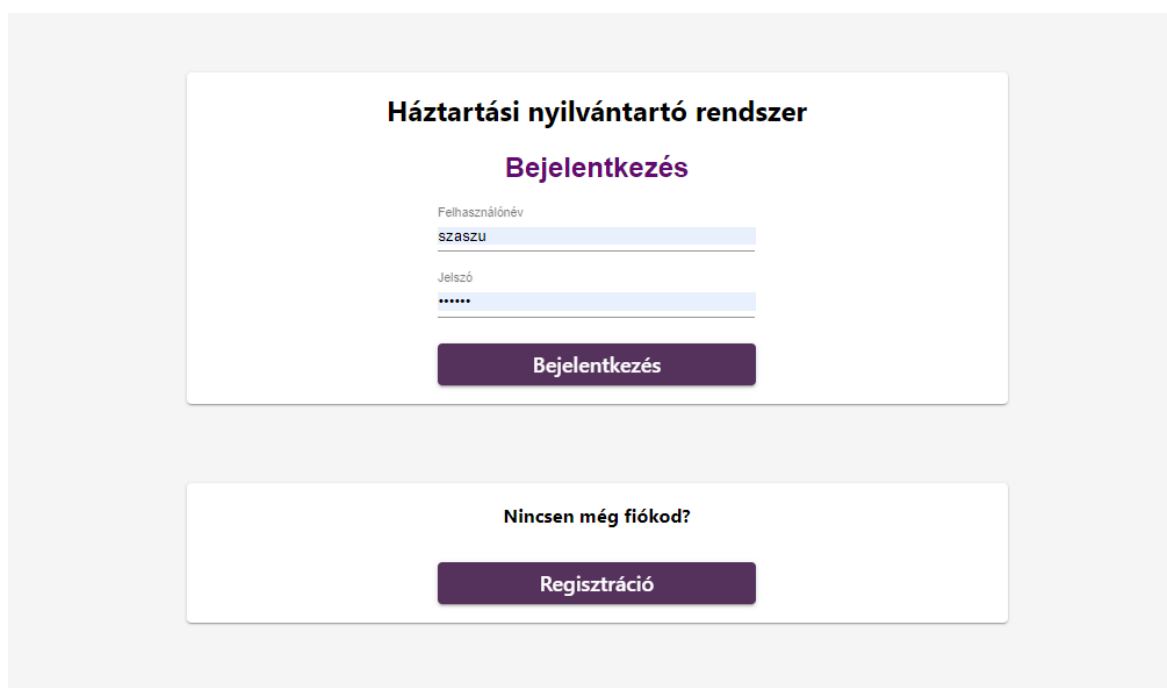
Tartalomjegyzék

Bevezetés	3
Bejelentkezés.....	3
Regisztráció.....	4
Termékek kilistázása.....	6
Termékek hozzáadása	7
Termékek módosítása	9
Felhasználó adatai	11

Bevezetés

A beadandóm témájának egy háztartási élelmiszer és egyéb termékeket nyilvántartó rendszer weboldalának elkészítését választottam. A weboldal színvariációinak főként a lila és a fehér szín árnyalatait használtam fel.

Bejelentkezés



The image shows a web interface for a 'Háztartási nyilvántartó rendszer' (Household Inventory System). It features two main sections: a login form and a registration button. The login form is titled 'Bejelentkezés' (Login) and contains two input fields: 'Felhasználónév' (Username) with the text 'szaszu' and 'Jelszó' (Password) with masked characters '*****'. Below these fields is a purple button labeled 'Bejelentkezés'. Below the login form is a separate box titled 'Nincsen még fiókod?' (Don't have an account yet?) with a purple button labeled 'Regisztráció' (Registration).

A fenti képen a rendszer bejelentkezési felületét láthatjuk. A már regisztrált felhasználók a jelszavuk és a felhasználónevük beírása után a Bejelentkezés gomb segítségével be tudnak lépni a rendszerbe. A még nem regisztrált felhasználók pedig a Regisztráció gomb segítségével regisztrálni tudnak a rendszerbe.

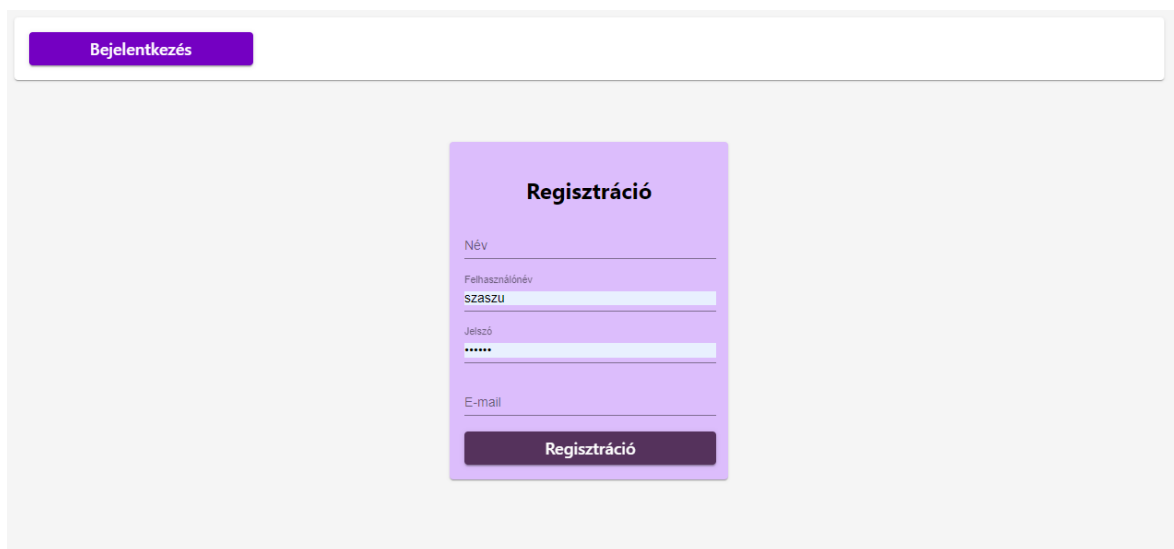
```

<mat-card class="card">
  <h2 class="title">Háztartási nyilvántartó rendszer</h2>
  <h2 class="logintitle">Bejelentkezés</h2>
  <mat-card-content>
    <form class="background" [formGroup]= "loginForm" (ngSubmit)="onSubmit()" class="form" >
      <mat-form-field>
        <input matInput type="text" placeholder="Felhasználónév" formControlName="userName">
      </mat-form-field>
      <br>
      <mat-form-field>
        <input matInput placeholder="Jelszó" type="password" formControlName="password">
      </mat-form-field>
      <br>
      <div id="wrongDatas"></div>
      <button mat-raised-button type="submit" [disabled]="!loginForm.valid" class="submit" >Bejelentkezés</button>
    </form>
  </mat-card-content>
</mat-card>

<mat-card class="card">
  <mat-card-content>
    <h3 class="title">Nincsen még fiókod?</h3>
    <button mat-raised-button routerLinkActive="active" class="reg" routerLink="/registration">Regisztráció</button>
  </mat-card-content>
</mat-card>

```

Regisztráció



The screenshot shows a web interface with a purple header bar containing a 'Bejelentkezés' button. Below the header, there is a light gray background area. In the center of this area is a white registration form with a purple border. The form is titled 'Regisztráció' and contains four input fields: 'Név' (Name), 'Felhasználónév' (Username) with the text 'SZASZU' entered, 'Jelszó' (Password) with masked characters '*****', and 'E-mail'. At the bottom of the form is a purple button labeled 'Regisztráció'.

A fenti képen a Regisztrációs űrlapot tekinthetjük meg. A regisztráció során a felhasználónak meg kell adnia a nevét és az email címét, választania kell egy felhasználónevet és egy jelszót. A névre és a jelszóra formai követelmények is vannak megadva, míg a többinél csak az, hogy tartalmazzon valamit.

```

<mat-card class="menucard">
  <mat-card-content>
    <button mat-raised-button routerLinkActive="active" class="menu" routerLink="/login">Bejelentkezés</button>
  </mat-card-content>
</mat-card>

```

```

<mat-card class="card">
  <h2 class="title">Regisztráció</h2>
  <mat-card-content>
    <form [formGroup]="createForm" (ngSubmit)="onSubmit()" class="form" >

      <mat-form-field >
        <input matInput type="text" placeholder="Név" formControlName="name">
      </mat-form-field>
      <br>

      <mat-form-field >
        <input matInput type="text" placeholder="Felhasználónév" formControlName="userName">
      </mat-form-field>
      <br>

```

```

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

      <mat-form-field >
        <input matInput placeholder="E-mail" type="text" formControlName="email">
      </mat-form-field>

      <br>

      <button mat-raised-button type="submit" class="submit" >Regisztráció</button>

    </form>
  </mat-card-content>
</mat-card>

```

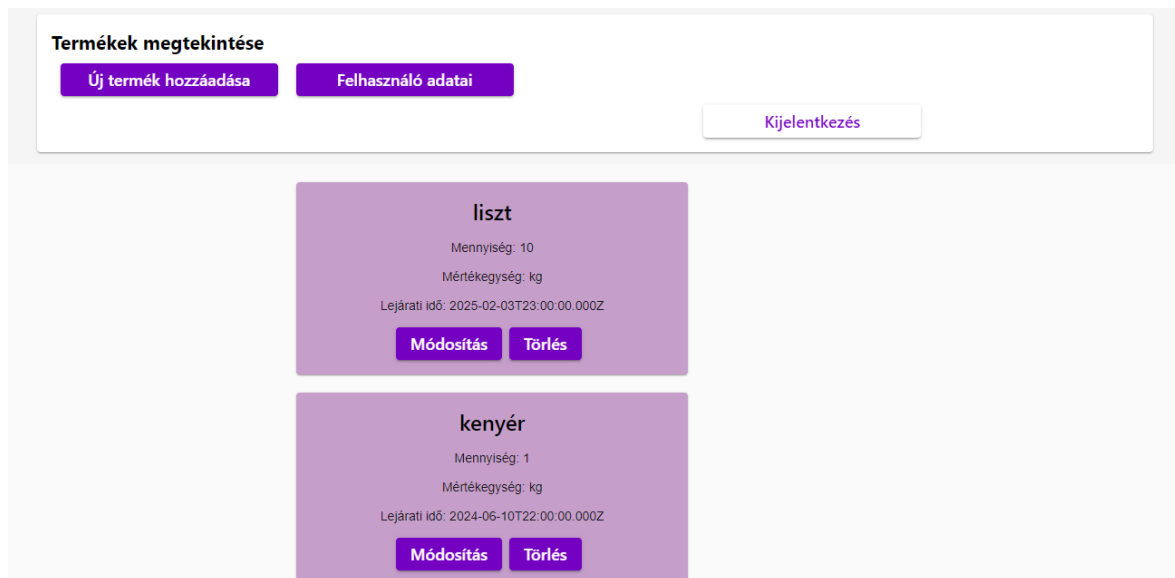
```

mainForm() {
  this.createForm = this.formBuilder.group({
    name: ['', [Validators.required, Validators.pattern('^[A-Z][a-z].+([a-z])+$')]],
    userName: [''],
    password: [''],
    email: ['', [Validators.required, Validators.pattern('^[A-Za-z0-9-\\+](\\.[A-Za-z0-9-\\+])*@[A-Za-z0-9-\\+](\\.[A-Za-z0-9-\\+])$')]],
  });
}

onSubmit() {
  this.submitted = true;
  if (!this.createForm.valid) {
    return false;
  } else {
    this.apiService.createUser(this.createForm.value).subscribe(
      (res) => {
        console.log('Successfully created!');
        this.ngZone.run(() => this.router.navigateByUrl('/login'));
      }, (error) => {
        console.log(error);
      });
  }
}
}

```

Termékek kilistázása



A fenti képen a termékek kilistázását láthatjuk. A Törlés gomb segítségével törölhetjük a kiválasztott terméket. A Módosítás gomb segítségével módosíthatjuk a kívánt terméket és adatait egy űrlap segítségével. A bal felső részben található Új termék hozzáadása gomb segítségével egy új terméket tudunk létrehozni egy űrlap segítségével. A mellette található Felhasználó adatai gomb segítségével megtekinthetjük az általunk a regisztráció során megadott adatokat.

```
<mat-card class="menucard">
  <mat-card-content>
    <h2 class="title">Termékek megtekintése</h2>
    <button mat-raised-button routerLinkActive="active" routerLink="/product-create" class="menubtn">Új termék hozzáadása</button>
    <button mat-raised-button routerLinkActive="active" routerLink="/account-info" class="menubtn">Felhasználó adatai</button>
    <button mat-raised-button routerLinkActive="active" (click)=logout() class="logout">Kijelentkezés</button>
  </mat-card-content>
</mat-card>
```

```
<mat-sidenav-container class="sidenav">
  <div class="list">
    <mat-card class="cardlist" *ngFor="let product of Products; let i= index">
      <mat-card-title class="title">
        {{product.name}}
      </mat-card-title>
      <mat-card-content>
        <br>
        <div> Mennyiség: {{product.amount}} </div>
        <br>
        <div> Mértékegység: {{product.unit}} </div>
        <br>
        <div> Lejáratí idő: {{product.limit}}</div>
      </mat-card-content>
      <button mat-raised-button (click)=edit(i)>Módosítás</button>
      <button mat-raised-button (click)=delete(i)>Törlés</button>
    </mat-card>
  </div>
</mat-sidenav-container>
```

```

readProduct() {
  this.apiService.getProducts().subscribe((data) =>{
    this.Products=data;
  });
}

getUser(){

  if (this.apiService.getCurrentUser().userName== null){
    this.router.navigate(['/login']);
  }

  this.user=this.apiService.getCurrentUser();
  this.name= JSON.stringify(this.user.name)
  this.username= JSON.stringify(this.user.userName)
  this.email= JSON.stringify(this.user.email)
}

logout(){
  this.user= new User()
  this.apiService.setCurrentuser(this.user);
  this.router.navigate(['/login']);
}

edit(index){

  let id=this.Products[index]._id;
  this.router.navigate(['/product-edit/'+ id]);
}

delete(index){

  this.apiService.deleteProduct(this.Products[index]._id ?? "").subscribe(() => {
    console.log("Deleted successfully!");
  });
  this.Products.splice(index,1);
}

```

Termékek hozzáadása

Új termék létrehozása

Termékek megtekintése

Felhasználó adatai

Kijelentkezés

Új termék létrehozása

Name

Mennyiség

Mértékegység

Lejárati idő (yyyy.MM.dd.)

Létrehozás

A fenti képen láthatjuk az új termékek létrehozásához szükséges űrlapot. Az űrlapon megadhatjuk a termék nevét, mennyiségét, ez a mennyiség milyen mértékegységben van megadva és a termék lejárat dátumát. A bal felső részben látható két gomb. Ezek segítségével tudunk visszamenni a termékek kilistázásához, és meg tudjuk innen is tekinteni a saját bejelentkezési adatainkat.

```
Go to component
<mat-card class="menucard">
  <mat-card-content>
    <h2 class="title">Új termék létrehozása</h2>
    <button mat-raised-button routerLinkActive="active" routerLink="/product-list" class="menubtn">Termékek megtekintése</button>
    <button mat-raised-button routerLinkActive="active" routerLink="/account-info" class="menubtn">Felhasználó adatai</button>
    <button mat-raised-button routerLinkActive="active" (click)=logout() class="logout">Kijelentkezés</button>
  </mat-card-content>
</mat-card>

<mat-card class="card">
  <h2 class="title">Új termék létrehozása</h2>
  <mat-card-content>
    <form [formGroup]="createForm" (ngSubmit)="onSubmit()" class="form" >
      <br>
      <mat-form-field >
        <input matInput type="text" placeholder="Name" formControlName="name">
      </mat-form-field>
      <br>

      <mat-form-field >
        <input matInput placeholder="Mennyiség" type="text" formControlName="amount">
      </mat-form-field>
      <br>

      <mat-form-field >
        <input matInput placeholder="Mértékegység" type="text" formControlName="unit">
      </mat-form-field>
      <br>

      <mat-form-field>
        <input matInput [matDatepicker]="picker" placeholder="Lejárat idő (yyyy.MM.dd.)" type="text" formControlName="limit">
        <mat-datepicker-toggle matIconSuffix [for]="picker"></mat-datepicker-toggle>
        <mat-datepicker #picker></mat-datepicker>
      </mat-form-field>

      <br>

      <button color="primary" mat-raised-button type="submit" class="submit" >Létrehozás</button>

    </form>
  </mat-card-content>
</mat-card>
```

```
mainForm() {
  this.createForm = this.formBuilder.group({
    name: [''],
    amount: ['', [Validators.required, Validators.pattern('[0-9]+$')]],
    unit: [''],
    limit: ['']
  });
}

onSubmit() {
  this.submitted = true;
  if (!this.createForm.valid) {
    return false;
  } else {
    this.apiService.createProduct(this.createForm.value).subscribe(
      (res) => {
        console.log('Product successfully created!');
        this.ngZone.run(() => this.router.navigateByUrl('/product-list'));
      }, (error) => {
        console.log(error);
      });
  }
}
```



```

getUser(){
    if (this.apiService.getCurrentuser().userName== null){
        this.router.navigate(['/login']);
    }

    this.user=this.apiService.getCurrentuser();
    this.name= JSON.stringify(this.user.name)
    this.username= JSON.stringify(this.user.userName)
    this.email= JSON.stringify(this.user.email)
}

logout(){
    this.user= new User()
    this.apiService.setCurrentuser(this.user);
    this.router.navigate(['/login']);
}
}

```

Termékek módosítása

Termék adatainak módosítása

Új termék hozzáadása
Termékek megtekintése
Felhasználó adatai

Kijelentkezés

Kiválasztott termék

Név
Mennyiség
Mértékegység
Lejárat idő (yyyy.MM.dd.)

Módosítás

A fenti képen láthatjuk a termék módosításához szükséges űrlapot. Az űrlap segítségével módosíthatjuk a kiválasztott termékünk adatait. Ha mégis meggondolnánk magunkat, és inkább új terméket szeretnénk hozzáadni és megtartani a már meglévőt, azt a bal felső sarokban lévő gomb segítségével megtehetjük. Termékek megtekintése gomb segítségével visszamehetünk a termékek kilistázásához, anélkül, hogy módosítottuk volna a kiválasztott terméket. A Felhasználó adatai gomb segítségével pedig megtekinthetjük a saját adatainkat.

```

Go to component
<mat-card class="menucard">
  <mat-card-content>
    <h2 class="title">Termék adatainak módosítása</h2>
    <button mat-raised-button routerLinkActive="active" routerLink="/product-create" class="menubtn">Új termék hozzáadása</button>
    <button mat-raised-button routerLinkActive="active" routerLink="/product-list" class="menubtn">Termékek megtekintése</button>
    <button mat-raised-button routerLinkActive="active" routerLink="/account-info" class="menubtn">Felhasználó adatai</button>
    <button mat-raised-button routerLinkActive="active" (click)=logout() class="logout">Kijelentkezés</button>
  </mat-card-content>
</mat-card>

```

```

<mat-card class="card">
  <h2 class="title">Kiválasztott termék</h2>
  <mat-card-content>
    <form [formGroup]="createForm" (ngSubmit)="onSubmit()" class="form" >
      <mat-form-field >
        <input matInput type="text" placeholder="Név" formControlName="name">
      </mat-form-field>
      <br>
      <mat-form-field >
        <input matInput placeholder="Mennyiség" type="text" formControlName="amount">
      </mat-form-field>
      <br>
      <mat-form-field >
        <input matInput placeholder="Mértékegység" type="text" formControlName="unit">
      </mat-form-field>
      <br>
      <mat-form-field>
        <input matInput [matDatepicker]="picker" placeholder="Lejáratí idő (yyyy.MM.dd.)" type="text" formControlName="limit">
        <mat-datepicker-toggle matIconSuffix [for]="picker"></mat-datepicker-toggle>
        <mat-datepicker #picker></mat-datepicker>
      </mat-form-field>
      <br>
      <button mat-raised-button type="submit" class="submit" >Módosítás</button>
    </form>
  </mat-card-content>
</mat-card>

```

```

export class ProductEditComponent implements OnInit {
  mainForm() {
    this.createForm = this.formBuilder.group({
      name: [''],
      amount: ['', [Validators.required, Validators.pattern('^[0-9]+$')]],
      unit: [''],
      limit: ['']
    });
  }
}

```

```

onSubmit() {
  this.submitted = true;
  if (!this.createForm.valid) {
    return false;
  } else {
    let id= this.router.url.toString();
    id=id.replace('/product-edit/', '');
    this.apiService.updateProduct(id, this.createForm.value).subscribe(
      (res) => {
        console.log('Product updated successfully!');
        this.ngZone.run(() => this.router.navigateByUrl('/product-list'));
      }, (error) => {
        console.log(error);
      });
  }
}

```

```

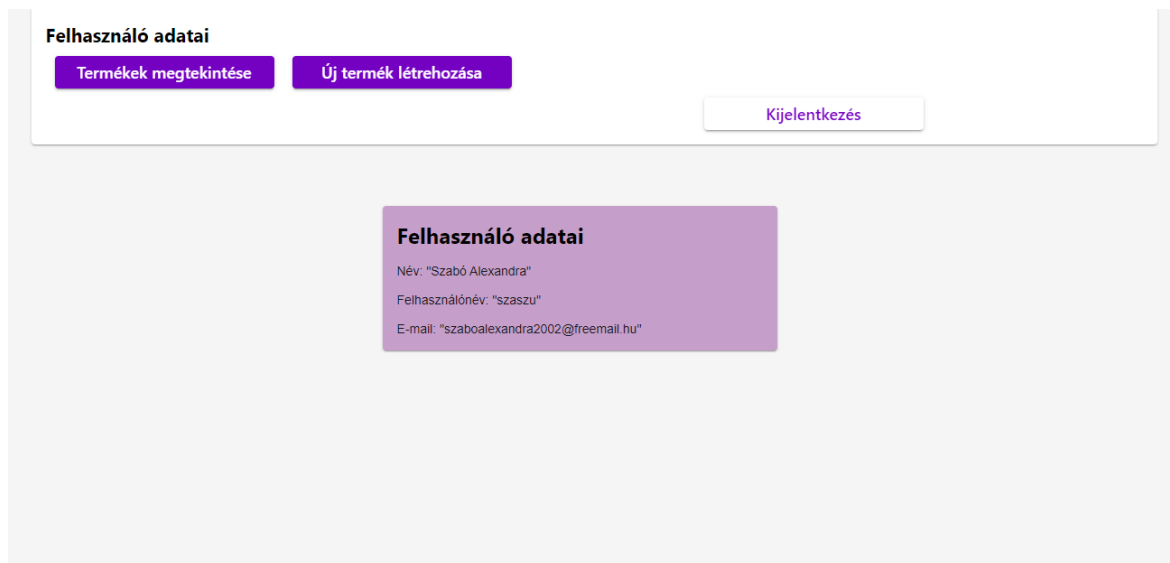
getUser(){
  if (this.apiService.getCurrentUser().userName== null){
    this.router.navigate(['/login']);
  }

  this.user=this.apiService.getCurrentUser();
  this.name= JSON.stringify(this.user.name)
  this.username= JSON.stringify(this.user.userName)
  this.email= JSON.stringify(this.user.email)
}

logout(){
  this.user= new User()
  this.apiService.setCurrentuser(this.user);
  this.router.navigate(['/login']);
}

```

Felhasználó adatai



A fenti képen az adott felhasználó adatait láthatjuk. A Termékek megtekintése gomb segítségével visszamehetünk a termékek kilistázásához. Az Új termék létrehozása gomb segítségével pedig létrehozhatunk egy újabb terméket.

```
<!-- component -->
<mat-card class="menucard">
  <mat-card-content>
    <h2 class="title">Felhasználó adatai</h2>
    <button mat-raised-button routerLinkActive="active" routerLink="/product-list" class="menubtn">Termékek megtekintése</button>
    <button mat-raised-button routerLinkActive="active" routerLink="/product-create" class="menubtn">Új termék létrehozása</button>
    <button mat-raised-button routerLinkActive="active" (click)=logout() class="logout">Kijelentkezés</button>
  </mat-card-content>
</mat-card>

<mat-card class="card">
  <h2 class="title">Felhasználó adatai</h2>
  <mat-card-content>
    <br>
    <div>Név: {{this.name}}</div>
    <br>
    <div>Felhasználónév: {{this.username}}</div>
    <br>
    <div>E-mail: {{this.email}}</div>
  </mat-card-content>
</mat-card>
```

```

@Component({
  selector: 'app-account-info',
  templateUrl: './account-info.component.html',
  styleUrls: ['./account-info.component.css']
})
export class AccountInfoComponent implements OnInit {

  user= new User();
  username: string;
  name: string;
  email: string;

  constructor(
    private router: Router,
    private apiService: ApiService

  ) {
    this.getUser();
  }

  ngOnInit(): void {
  }
}

```

```

  ngOnInit(): void {
  }

  getUser(){

    if (this.apiService.getCurrentUser().userName== null){
      this.router.navigate(['/login']);
    }

    this.user=this.apiService.getCurrentUser();
    this.name= JSON.stringify(this.user.name)
    this.username= JSON.stringify(this.user.userName)
    this.email= JSON.stringify(this.user.email)
  }

  logout(){
    this.user= new User()
    this.apiService.setCurrentuser(this.user);
    this.router.navigate(['/login']);
  }
}

```