

StudentMate

1)Home

HOME.COMPONENT.HTML

The screenshot shows the Visual Studio Code interface with the file 'home.component.html' open. The code displays four mat-card components, each containing an image and some descriptive text. The images are labeled img1.jpg, img2.jpg, img3.jpg, and img4.jpg. The text within the cards is as follows:

```
1 <div class="container heading">
2   <h1>FUN LEARNING WITH FLASHCARDS</h1>
3 </div>
4
5   <div class="container">
6     <mat-card class="card">
7       
9     </mat-card>
10
11    <mat-card class="card">
12      
14    </mat-card>
15
16    <mat-card class="card">
17      
19    </mat-card>
20
21    <mat-card class="card">
22      
24      <div class="sh">
25        <mat-card-content>
26          <strong>
27            <br>
28            For students in classes Ukg-5</strong>
29          </mat-card-content>
30        </div>

```

The status bar at the bottom indicates 'Ln 1, Col 1' and other standard VS Code settings.

HOME.COMPONENT.HTML

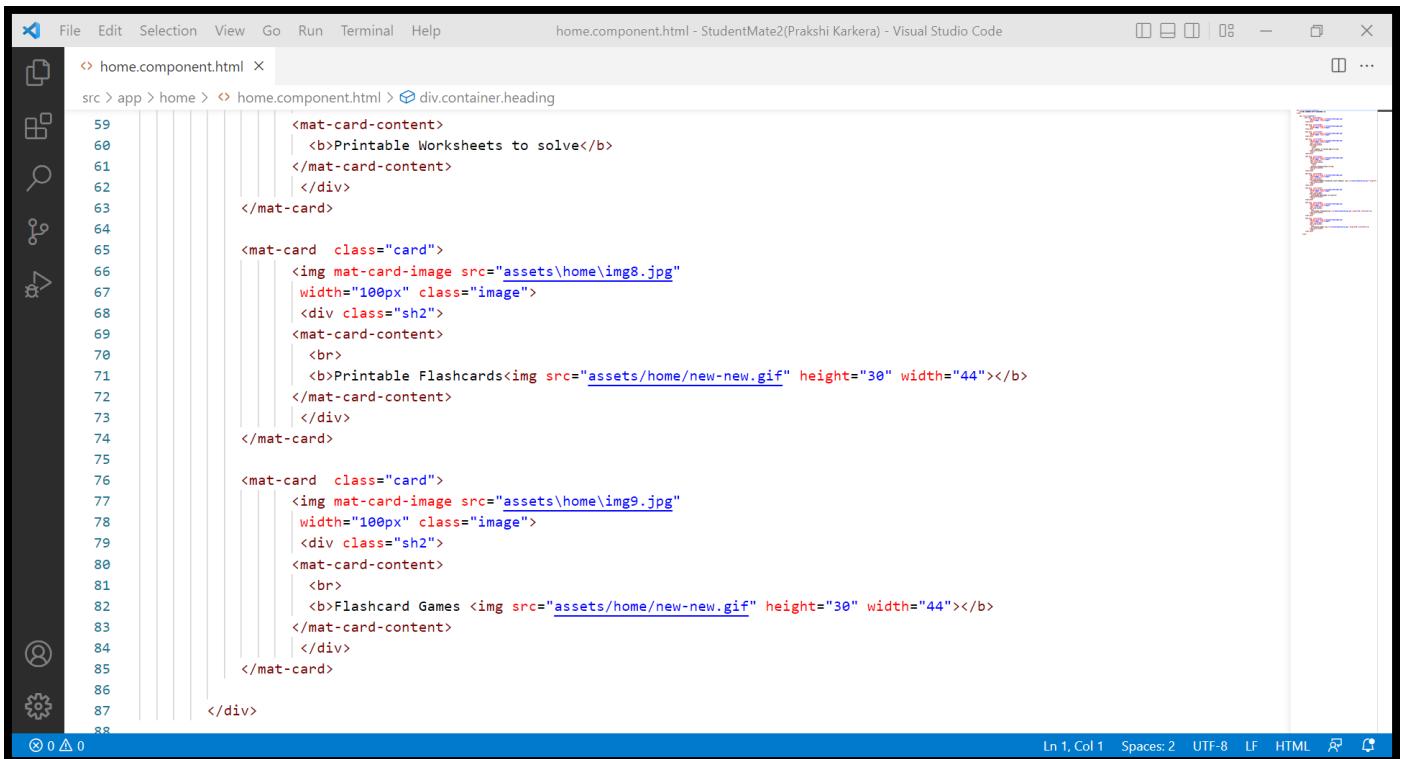
The screenshot shows the Visual Studio Code interface with the file 'home.component.html' open. The code has been updated to include three new mat-card components, each containing an image and descriptive text. The images are labeled img5.jpeg, img6.jpg, and img7.png. The text within the cards is as follows:

```
33   <mat-card class="card">
34     
36     <div class="sh">
37       <mat-card-content>
38         <strong>
39           <br>
40           Activity Flashcard Sets</strong>
41         </mat-card-content>
42       </div>
43     </mat-card>
44
45   <mat-card class="card">
46     
48     <div class="sh">
49       <mat-card-content>
50         <strong>All-Subject Flashcards (with freebies) </strong>
51       </mat-card-content>
52     </div>
53   </mat-card>
54
55   <mat-card class="card">
56     
58     <div class="sh2">
59       <mat-card-content>
60         <b>Printable Worksheets to solve</b>
61       </mat-card-content>

```

The status bar at the bottom indicates 'Ln 1, Col 1' and other standard VS Code settings.

HOME.COMPONENT.HTML



```
File Edit Selection View Go Run Terminal Help home.component.html - StudentMate2(Prakshi Karkera) - Visual Studio Code

src > app > home > home.component.html > div.container.heading

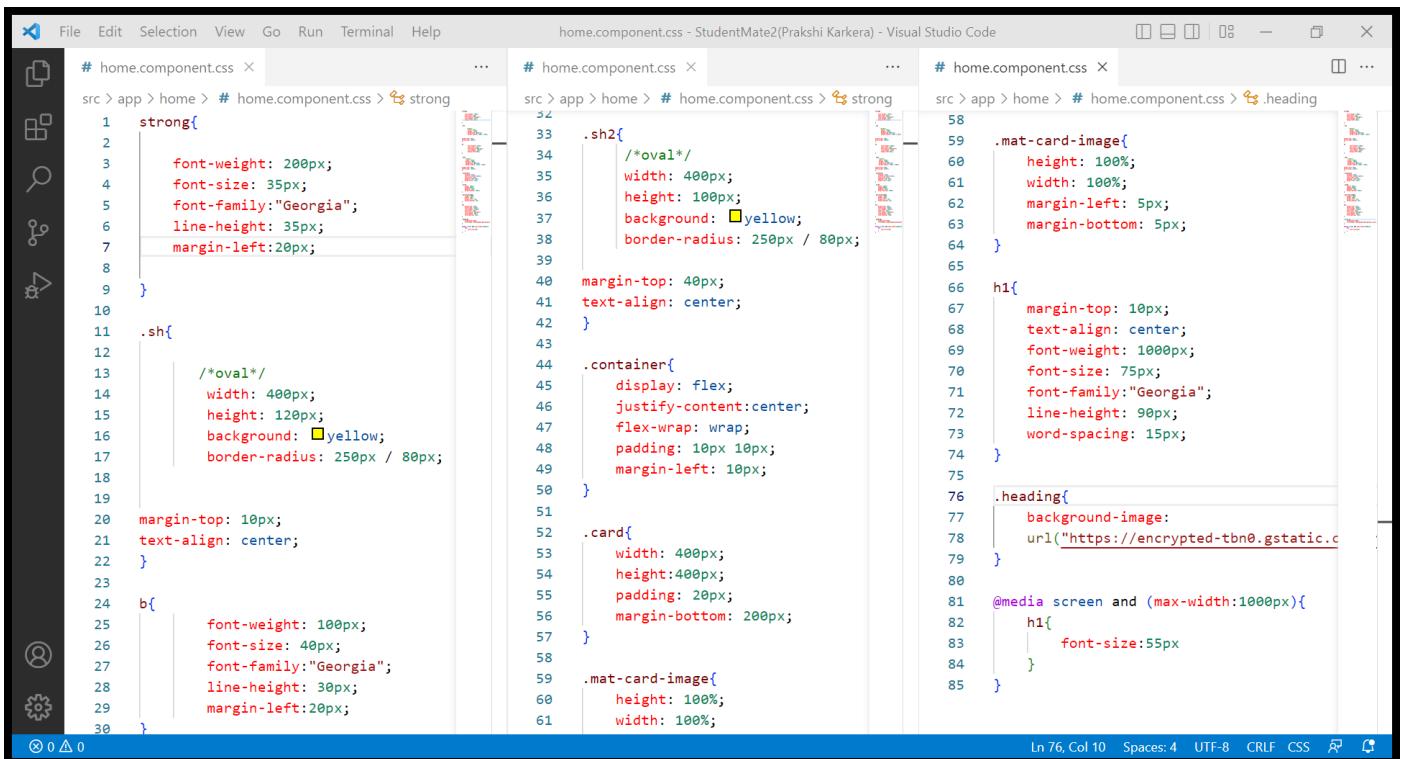
59     <mat-card-content>
60         <b>Printable Worksheets to solve</b>
61     </mat-card-content>
62     </div>
63 </mat-card>

64
65     <mat-card class="card">
66         
68         <div class="sh2">
69             <mat-card-content>
70                 <br>
71                 <b>Printable Flashcards</b>
72             </mat-card-content>
73         </div>
74     </mat-card>

75
76     <mat-card class="card">
77         
79         <div class="sh2">
80             <mat-card-content>
81                 <br>
82                 <b>Flashcard Games </b>
83             </mat-card-content>
84         </div>
85     </mat-card>
86
87 </div>

Ln 1, Col 1 Spaces: 2 UTF-8 LF HTML ⚡ 🌐
```

HOME.COMPONENT.CSS



```
File Edit Selection View Go Run Terminal Help home.component.css - StudentMate2(Prakshi Karkera) - Visual Studio Code

# home.component.css > ...
src > app > home > # home.component.css > strong
1   strong{
2       font-weight: 200px;
3       font-size: 35px;
4       font-family: "Georgia";
5       line-height: 35px;
6       margin-left: 20px;
7   }
8
9
10 .sh{
11     /*oval*/
12     width: 400px;
13     height: 128px;
14     background: yellow;
15     border-radius: 250px / 80px;
16
17     margin-top: 10px;
18     text-align: center;
19 }
20
21
22
23
24 b{
25     font-weight: 100px;
26     font-size: 40px;
27     font-family: "Georgia";
28     line-height: 30px;
29     margin-left: 20px;
30 }

# home.component.css > ...
src > app > home > # home.component.css > strong
31 .sh2{
32     /*oval*/
33     width: 400px;
34     height: 100px;
35     background: yellow;
36     border-radius: 250px / 80px;
37
38     margin-top: 40px;
39     text-align: center;
40 }
41
42
43
44 .container{
45     display: flex;
46     justify-content: center;
47     flex-wrap: wrap;
48     padding: 10px 10px;
49     margin-left: 10px;
50 }
51
52 .card{
53     width: 400px;
54     height: 400px;
55     padding: 20px;
56     margin-bottom: 200px;
57 }
58
59 .mat-card-image{
60     height: 100%;
61     width: 100%;
62     margin-left: 5px;
63     margin-bottom: 5px;
64 }
65
66 h1{
67     margin-top: 10px;
68     text-align: center;
69     font-weight: 1000px;
70     font-size: 75px;
71     font-family: "Georgia";
72     line-height: 90px;
73     word-spacing: 15px;
74 }
75
76 .heading{
77     background-image:
78         url("https://encrypted-tbn0.gstatic.com");
79 }
80
81 @media screen and (max-width:1000px){
82     h1{
83         font-size: 55px
84     }
85 }

Ln 76, Col 10 Spaces: 4 UTF-8 CRLF CSS ⚡ 🌐
```

Home Page

StudentMate JSON Server localhost:4200

FUN LEARNING WITH FLASHCARDS

EARLY CHILDHOOD DEVELOPMENT
Perfect Tool For Teaching
Easy Travel
Stay Organized
Colors in Flash

QUICK AND EASY
FLASHCARDS ACTIVITIES FOR KIDS
FUN GAMES TO
PLAY
www.sharingourexperiences.com

ALPHABET Flashcards
NUMBER COUNTING Flashcards
COLORS & SHAPES Flashcards
FIRST WORDS Flashcards
ANIMALS Flashcards
VEGETABLES Flashcards
FISH Flashcards
CLOCK Flashcards
PENGUINS Flashcards
Fruit Flashcards
apple
Shapes
Color
hexagon
Skip Count by 5's
5 + 5 +
10
15
20
25

StudentMate JSON Server localhost:4200/home

144 CARDS
UKG

100+ ALPHABET ACTIVITY FLASH CARDS FOR KIDS
Great FOR HANDS ON FUN and Learning Alphabets
Click Here for Download

FREE Subjects Flashcards
Math English Arts and Crafts Science

For students in classes Ukg-5

Activity Flashcard Sets

All-Subject Flashcards (with freebies)

StudentMate JSON Server localhost:4200/home

Word Practice **good**
Read the Word. Color the Word.
good **good**
Trace the Word. Write the Word.
good
Correct the Letters. Hand the Word.
g o o d m e k g
d o o d o o d d
Write the Words in Sentences.
Fluffy is a **cat**.

ABC FRUITS Flashcards
Colorable Edition
A is for Apple B is for Banana C is for Cranberry D is for Dragon fruit E is for Elderberry F is for Fig

FUN FLASHCARD GAMES FOR KIDS
Colorful Edition
A girl pointing at the camera.

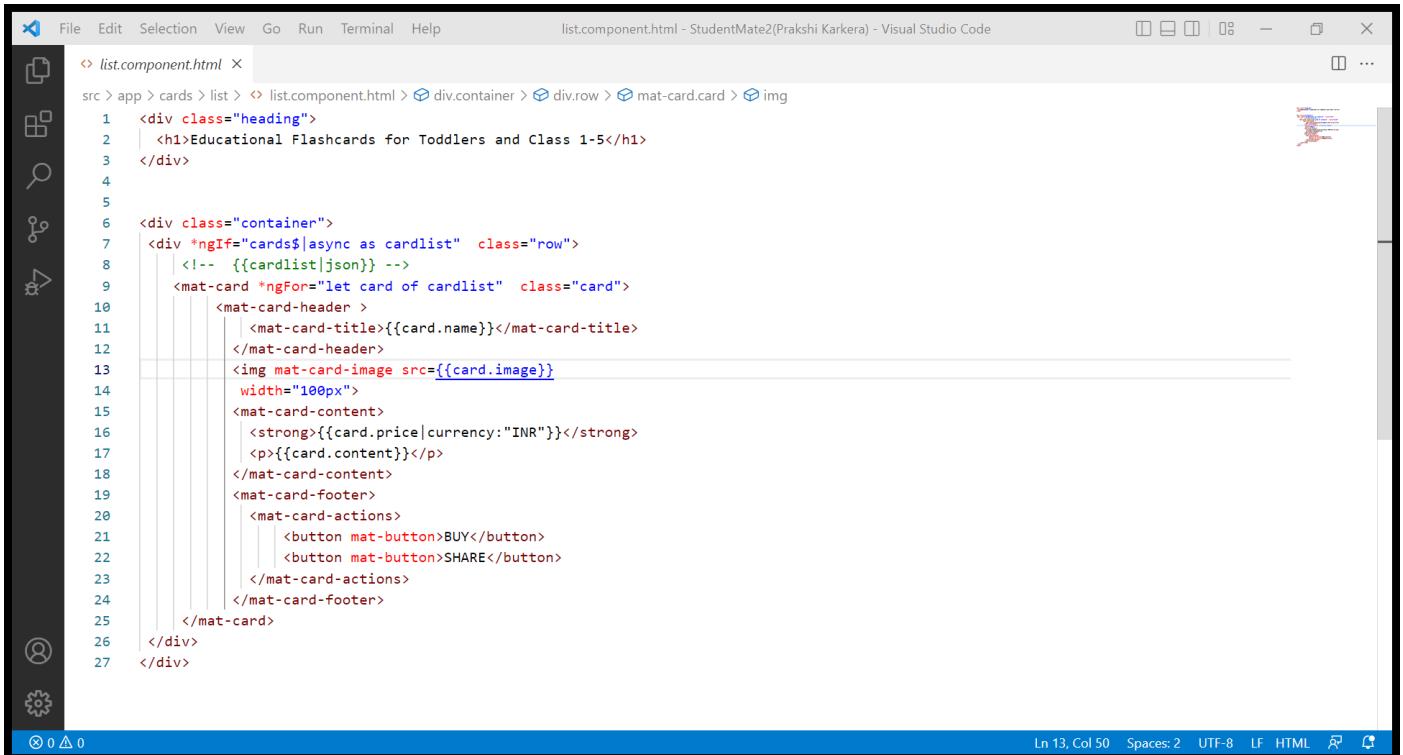
Printable Worksheets to solve

Printable Flashcards

Flashcard Games

2) Flashcards

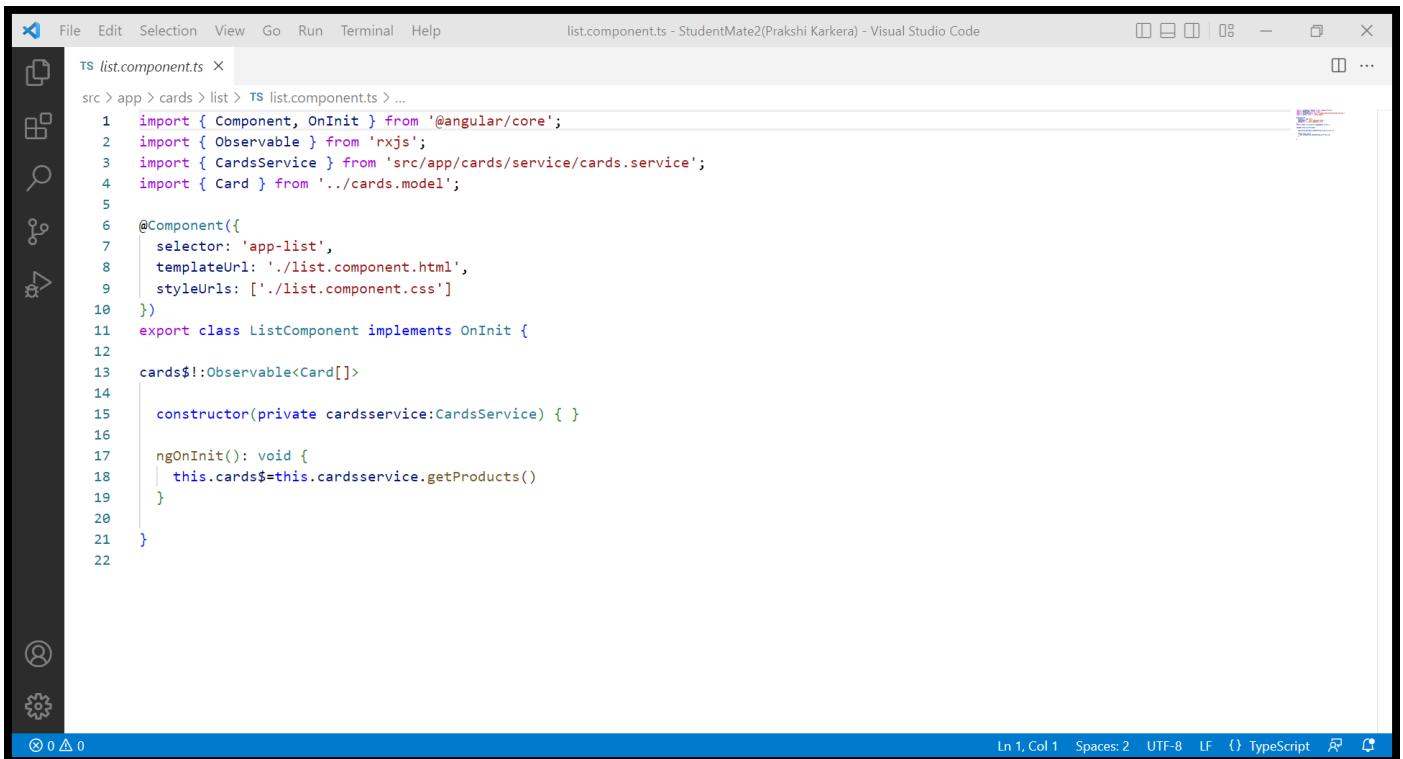
LIST.COMPONENT.HTML



```
<div class="heading">
  <h1>Educational Flashcards for Toddlers and Class 1-5</h1>
</div>

<div class="container">
  <div *ngIf="cards$|async as cardlist" class="row">
    <!-- {{cardlist|json}} -->
    <mat-card *ngFor="let card of cardlist" class="card">
      <mat-card-header>
        <mat-card-title>{{card.name}}</mat-card-title>
      </mat-card-header>
      
      <mat-card-content>
        <strong>{{card.price|currency:"INR"}}</strong>
        <p>{{card.content}}</p>
      </mat-card-content>
      <mat-card-footer>
        <mat-card-actions>
          <button mat-button>BUY</button>
          <button mat-button>SHARE</button>
        </mat-card-actions>
      </mat-card-footer>
    </mat-card>
  </div>
</div>
```

LIST.COMPONENT.TS



```
import { Component, OnInit } from '@angular/core';
import { Observable } from 'rxjs';
import { CardsService } from 'src/app/cards/service/cards.service';
import { Card } from '../cards.model';

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

  cards$: Observable<Card[]>

  constructor(private cardsservice:CardsService) { }

  ngOnInit(): void {
    this.cards$=this.cardsservice.getProducts()
  }
}
```

LIST.COMPONENT.CSS

The screenshot shows three tabs of the same CSS file, list.component.css, displayed side-by-side in Visual Studio Code. The tabs are labeled '# list.component.css X' and are located at the top of each column. The code is identical across all three tabs, demonstrating a multi-column layout for better readability.

```
1 .container{  
2   display: flex;  
3   justify-content:center;  
4   flex-wrap: wrap;  
5   padding: 10px 10px;  
6   margin-left: 10px;  
7   margin-top: 90px;  
8 }  
9 .card{  
10  width: 500px;  
11  padding: 10px;  
12  margin-bottom: 100px;  
13  background-image: url("https://en...");  
14 }  
15 strong {  
16  font-size: 20px;  
17 }  
18 p,button{  
19  font-size: 20px;  
20  margin-left: 10px;  
21 }  
22 }  
23 mat-card-title{  
24  font-size: 35px;  
25 }  
26 .mat-card-image{  
27  height: 100%;  
28  width: 100%;  
29  margin-left: 2px;  
30  margin-right: 2px;  
31 }  
32 mat-card-actions{  
33 {  
34  margin-left:5px  
35 }  
36 h1{  
37  margin-top: 10px;  
38  text-align: center;  
39  font-weight: 1000px;  
40  font-size: 75px;  
41  font-family:"Georgia";  
42  line-height: 90px;  
43  word-spacing: 15px;  
44 }  
45 .heading{  
46  background-image: url("https://en...");  
47 }  
48 }  
49 @media screen and (min-width:100px){  
50  .card{  
51  width: 500px;  
52  padding: 5px;  
53  margin-bottom: 70px;  
54  background-image: url("https://...");  
55 }  
56 h1{  
57  font-size:75px  
58 }  
59 }  
60 @media screen and (max-width:1000px){  
61  .card{  
62  width: 250px;  
63  padding: 5px;  
64  margin-bottom: 100px;  
65  background-image: url("https://...");  
66 }  
67 h1{  
68  font-size: 50px;  
69 }  
70 }  
71 }  
72 }
```

Ln 25, Col 2 Spaces: 4 UTF-8 CRLF CSS ⚙️ 🔍

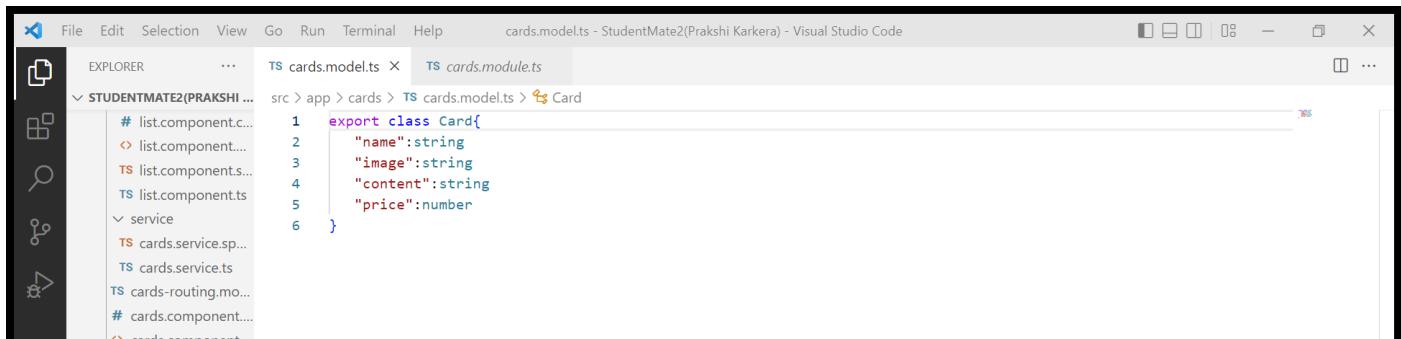
CARDS.SERVICES.TS

The screenshot shows the cards.service.ts file in Visual Studio Code. The tab bar at the top indicates it is the active file. The code defines a service for managing cards, utilizing Angular's HttpClient and Injectable annotations.

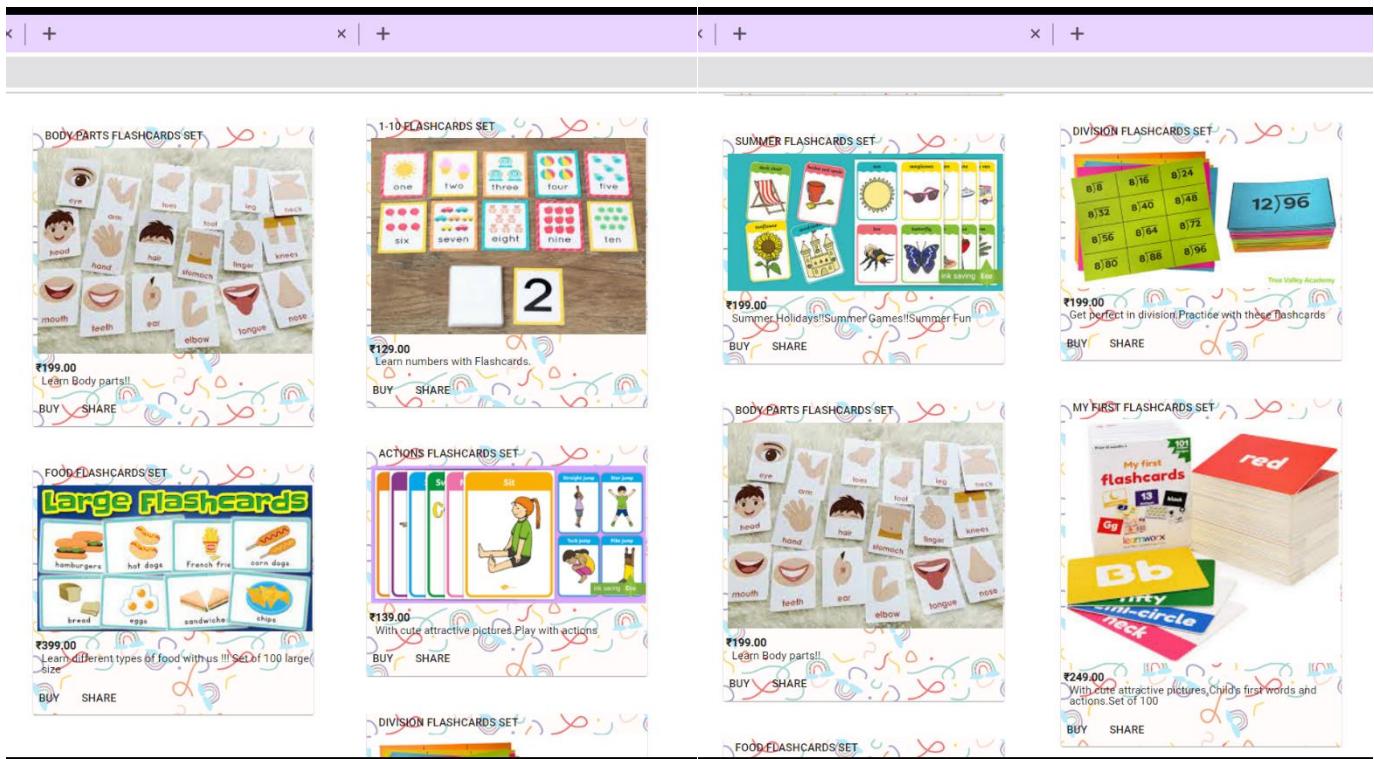
```
1 import { HttpClient } from '@angular/common/http';  
2 import { Injectable } from '@angular/core';  
3 import { Card } from '../cards.model';  
4  
5 @Injectable({  
6  providedIn: 'root'  
7 })  
8 export class CardsService {  
9  
10  constructor(private http:HttpClient) { }  
11  
12  getProducts()  
13  {  
14  return this.http.get<Card[]>("http://localhost:3000/cards/")  
15  }  
16 }  
17  
18  
19 }  
20
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF () TypeScript ⚙️ 🔍

CARD.MODEL.TS

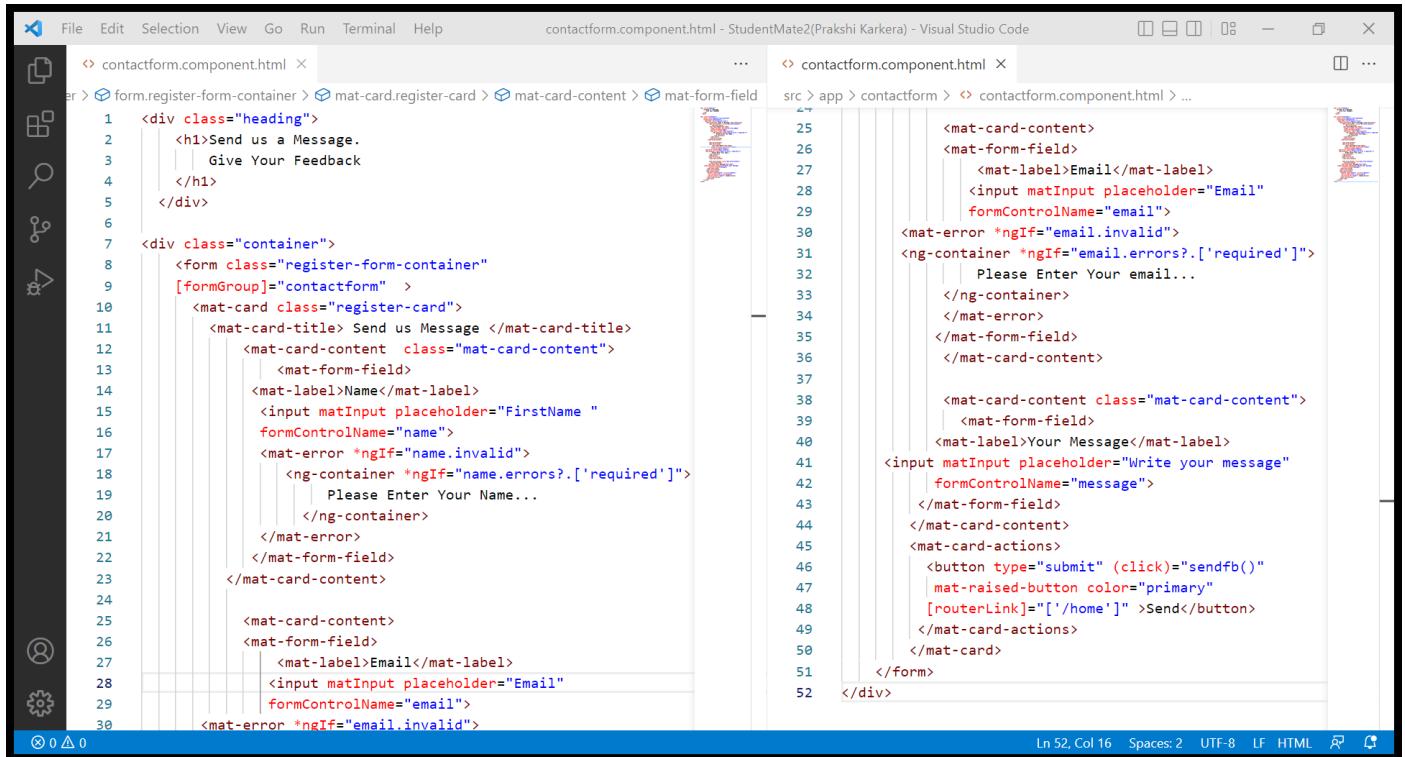


```
File Edit Selection View Go Run Terminal Help cards.model.ts - StudentMate2(Prakshi Karkera) - Visual Studio Code
EXPLORER ... TS cards.model.ts x TS cards.module.ts
src > app > cards > TS cards.model.ts > Card
1 export class Card{
2   "name":string
3   "image":string
4   "content":string
5   "price":number
6 }
```



3) Contact-Us

CONTACTFORM.COMPONENT.HTML



```
<div class="heading">
  <h1>Send us a Message.
  || Give Your Feedback
</h1>
</div>

<div class="container">
  <form class="register-form-container" [formGroup]="contactform" >
    <mat-card class="register-card">
      <mat-card-title> Send us Message </mat-card-title>
      <mat-card-content class="mat-card-content">
        <mat-form-field>
          <mat-label>Name</mat-label>
          <input matInput placeholder="FirstName" formControlName="name">
          <mat-error *ngIf="name.invalid">
            <ng-container *ngIf="name.errors?.['required']">
              Please Enter Your Name...
            </ng-container>
          </mat-error>
        </mat-form-field>
      </mat-card-content>

      <mat-card-content>
        <mat-form-field>
          <mat-label>Email</mat-label>
          <input matInput placeholder="Email" formControlName="email">
          <mat-error *ngIf="email.invalid">

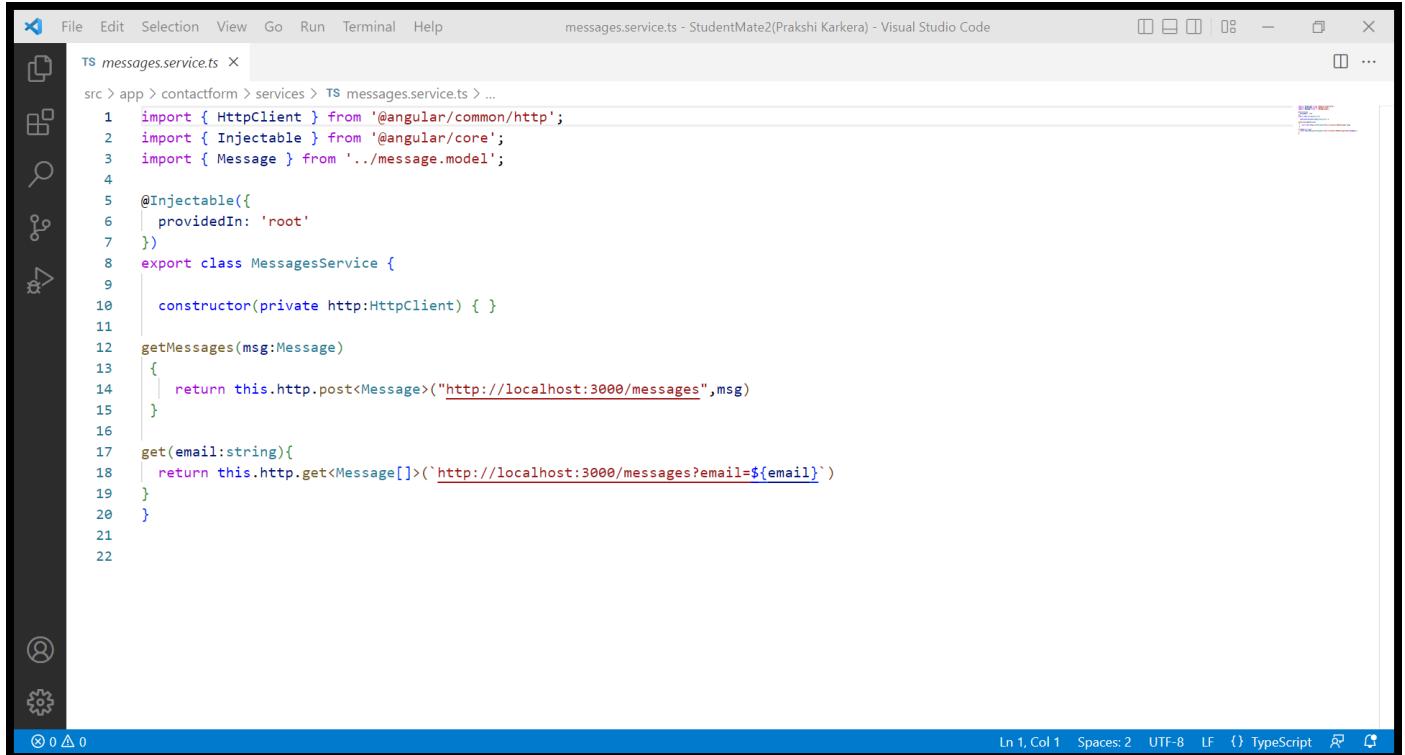
```

```
<mat-card-content>
<mat-form-field>
  <mat-label>Email</mat-label>
  <input matInput placeholder="Email" formControlName="email">
  <mat-error *ngIf="email.invalid">
    <ng-container *ngIf="email.errors?.['required']">
      Please Enter Your email...
    </ng-container>
  </mat-error>
</mat-form-field>
</mat-card-content>

<mat-card-content class="mat-card-content">
  <mat-form-field>
    <mat-label>Your Message</mat-label>
    <input matInput placeholder="Write your message" formControlName="message">
  </mat-form-field>
</mat-card-content>
<mat-card-actions>
  <button type="submit" (click)="sendfb()" mat-raised-button color="primary" [routerLink]="/home">Send</button>
</mat-card-actions>
</mat-card>
</div>
```

Ln 52, Col 16 Spaces: 2 UTF-8 LF HTML

MESSAGES.SERVICES.TS



```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Message } from '../message.model';

@Injectable({
  providedIn: 'root'
})
export class MessagesService {
  constructor(private http:HttpClient) { }

  getMessages(msg:Message)
  {
    return this.http.post<Message>("http://localhost:3000/messages",msg)
  }

  get(email:string){
    return this.http.get<Message[]>(`http://localhost:3000/messages?email=${email}`)
  }
}
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF {} TypeScript

MESSAGE.MODEL.TS

A screenshot of Visual Studio Code showing the file `message.model.ts`. The code defines a class `Message` with properties: `id`, `name`, `email`, and `message`. The file path is `src > app > contactform > message.model.ts`.

```
1 export class Message{
2     "id":number=0
3     "name":string=""
4     "email":string=""
5     "message":string=""
6 }
7
```

CONTACTFORM.COMPONENT.CSS

A screenshot of Visual Studio Code showing two files side-by-side: `contactform.component.css` and `# contactform.component.css`. The left file contains styles for the main component, and the right file contains media queries for different screen widths. Both files include styles for the `.register-form-container` and `.register-card` components.

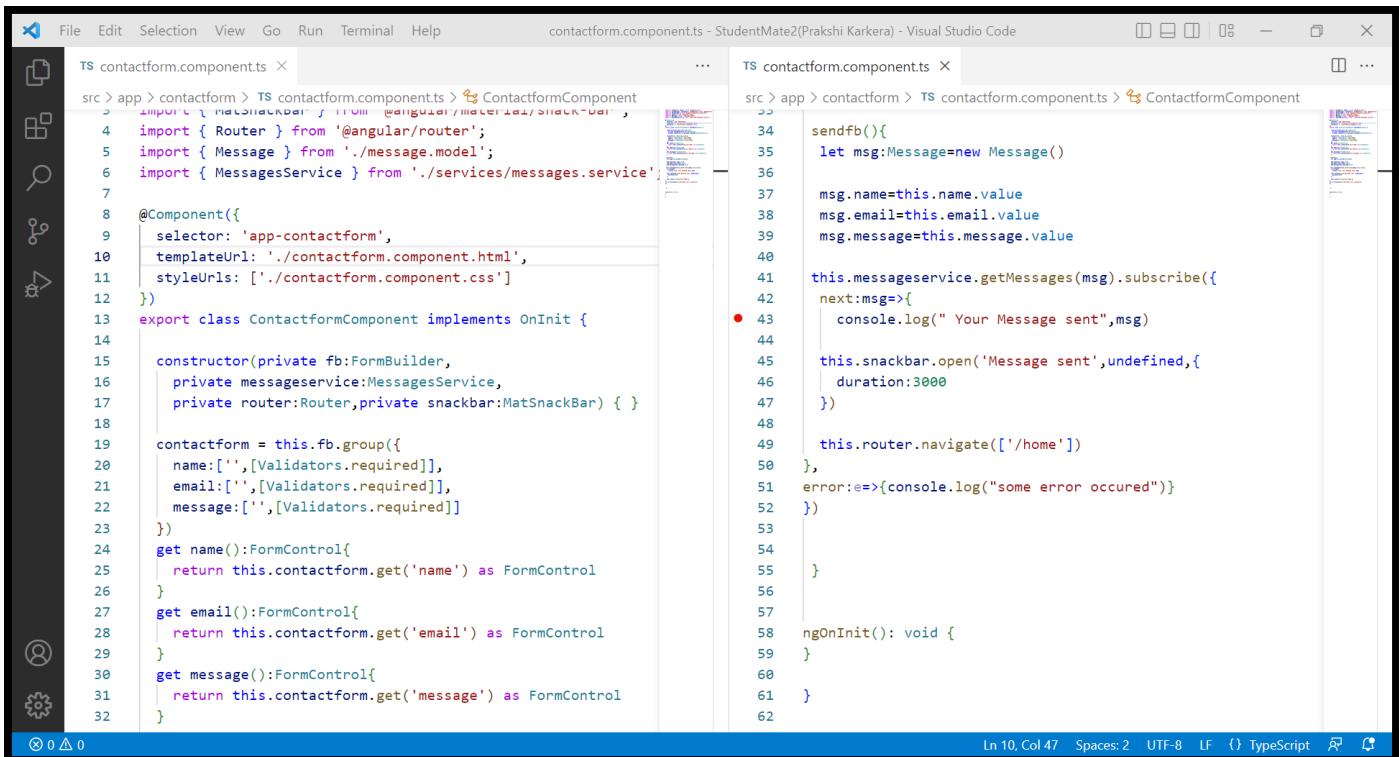
`contactform.component.css` content:

```
1 .register-form-container{
2     display:flex;
3     flex-direction:column;
4     align-items:center;
5     justify-content: center;
6     height: 100%;
7     margin-top: 100px;
8     background-image:
9         url("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9c...")
```

`# contactform.component.css` content:

```
1 # contactform.component.css
2
3 @media(min-width: 100px){
4     .register-form-container{
5         margin-top:70px;
6     }
7 }
8
9 @media screen and (max-width: 1000px){
10    h1{
11        margin-top: 10px;
12        text-align: center;
13        font-weight: 1000px;
14        font-size: 75px;
15        font-family:"Georgia";
16        line-height: 90px;
17        word-spacing: 15px;
18    }
19
20    .register-card{
21        width:30rem;
22    }
23
24    .mat-card-content{
25        display: flex;
26        flex-direction: column;
27        margin-top: 10px;
28    }
29
30 }
```

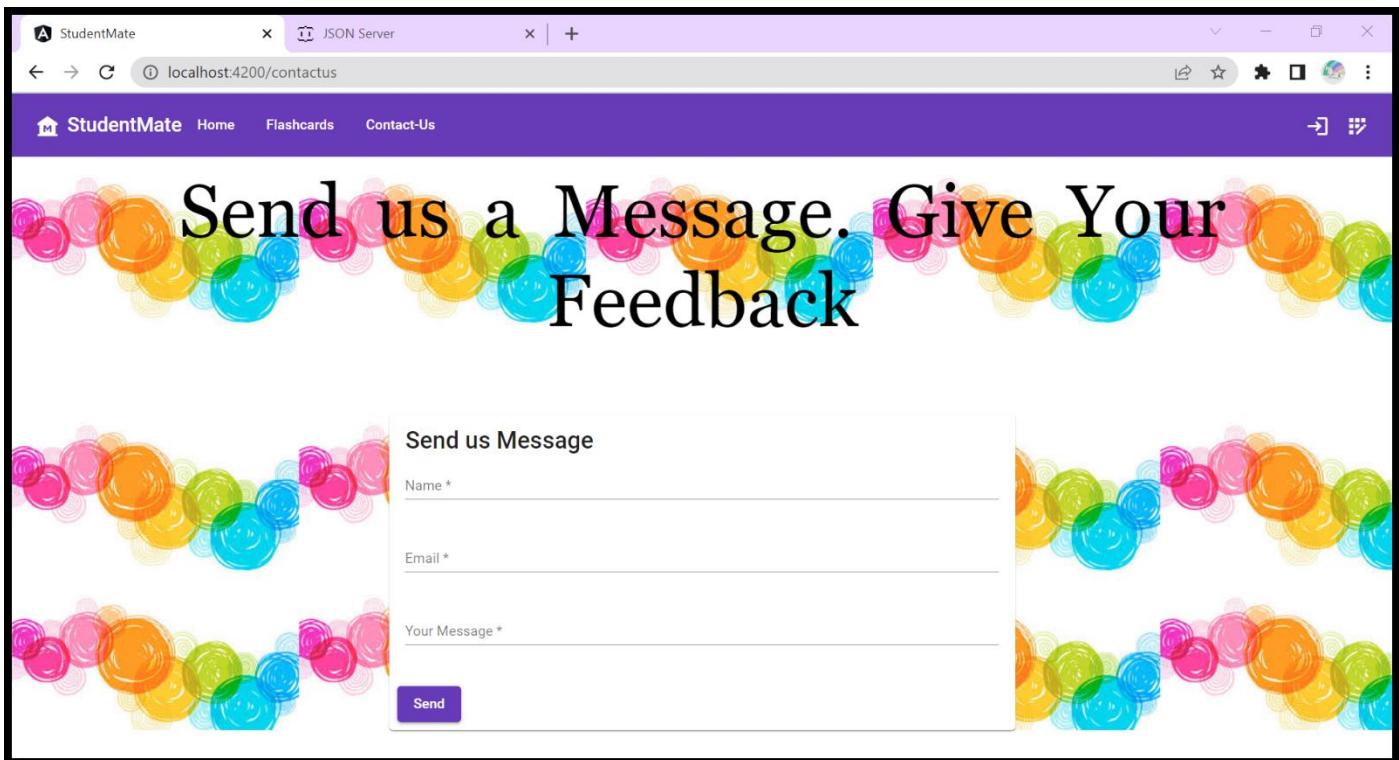
CONTACTFORM.COMPONENT.TS



```
src > app > contactform > TS contactform.component.ts - StudentMate2(Prakshi Karkera) - Visual Studio Code
TS contactform.component.ts
...
src > app > contactform > TS contactform.component.ts - StudentMate2(Prakshi Karkera) - Visual Studio Code
TS contactform.component.ts
...
34  sendfb(){
35    let msg:Message=new Message()
36
37    msg.name=this.name.value
38    msg.email=this.email.value
39    msg.message=this.message.value
40
41    this.messageservice.getMessages(msg).subscribe({
42      next:msg=>{
43        console.log(" Your Message sent",msg)
44
45        this.snackbar.open('Message sent',undefined,{
46          duration:3000
47        })
48
49        this.router.navigate(['/home'])
50      },
51      error:e=>{console.log("some error occurred")}
52    })
53
54  }
55
56
57
58  ngOnInit(): void {
59  }
60
61  }
62
```

Ln 10, Col 47 Spaces: 2 UTF-8 LF (TypeScript

CONTACTFORM



4) Login

LOGIN.COMPONENT.HTML

```
File Edit Selection View Go Run Terminal Help login.component.html - StudentMate2(Prakshi Karkera) - Visual Studio Code ... login.component.html x ... login.component.html x ...
```

```
1 <div class="container">
2   <form class="login-form-container"
3     [formGroup]="loginform" (ngSubmit)="dologin()">
4     <mat-card class="login-card">
5       <mat-card-title> Login </mat-card-title>
6       <mat-card-content class="mat-card-content">
7         <mat-form-field>
8           <mat-label>Email</mat-label>
9           <input matInput placeholder="Email"
10            formControlName="email">
11           <mat-error *ngIf="email.invalid">
12             Please Enter Your Email...
13           </ng-container>
14           </mat-error>
15         </mat-form-field>
16       </mat-card-content>
17
18       <mat-card-content class="mat-card-content">
19         <mat-form-field>
20           <mat-label>Password</mat-label>
21           <input matInput placeholder="password"
22            formControlName="password">
23           <mat-error *ngIf="password.invalid">
24             Please Enter Your password...
25           </ng-container>
26           </mat-error>
27         </mat-form-field>
28       </mat-card-content>
29
30     </mat-card>
31   </form>
32 </div>
```

```
13 Please Enter Your Email...
14 | | | </ng-container>
15 </mat-error>
16 </mat-form-field>
17 </mat-card-content>
18
19 <mat-card-content class="mat-card-content">
20   <mat-form-field>
21     <mat-label>Password</mat-label>
22     <input matInput placeholder="password"
23      formControlName="password">
24     <mat-error *ngIf="password.invalid">
25       Please Enter Your password...
26     </ng-container>
27     </mat-error>
28   </mat-form-field>
29 </mat-card-content>
30
31 <mat-card-actions>
32   <button mat-raised-button color="primary">Login</button>
33   New member? Please
34   <a [routerLink]="/user,'register'"> Click Here </a>
35 </mat-card-actions>
36 </mat-card>
37 </div>
38 </div>
39 </div>
```

Ln 33, Col 4 Spaces: 2 UTF-8 LF HTML

LOGIN.COMPONENT.TS

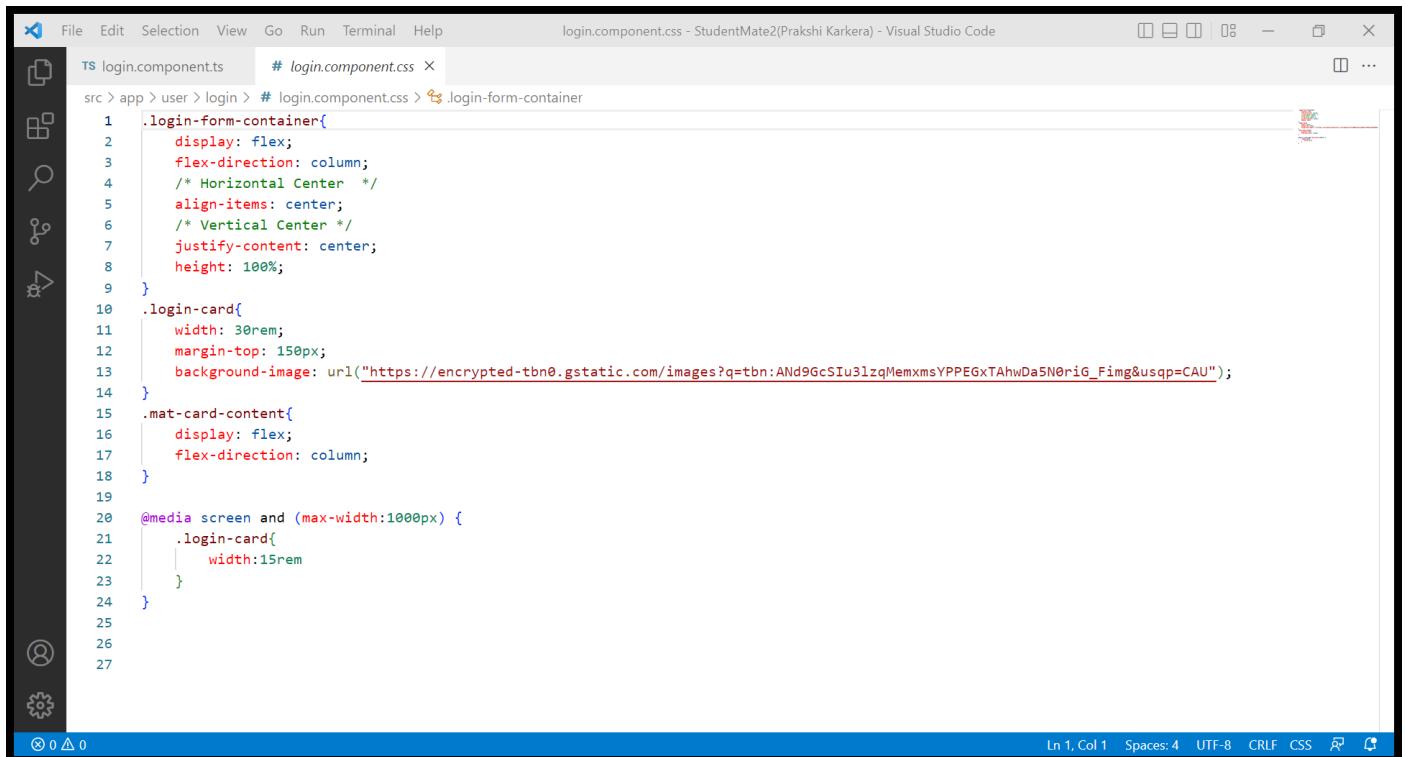
```
File Edit Selection View Go Run Terminal Help login.component.ts - StudentMate2(Prakshi Karkera) - Visual Studio Code ... login.component.ts x ...
```

```
src > app > user > login > TS login.component.ts > LoginComponent > constructor
1 import { Component, OnInit } from '@angular/core';
2 import { FormBuilder, FormControl, Validators } from '@angular/forms';
3 import { MatSnackBar } from '@angular/material/snack-bar';
4 import { Router } from '@angular/router';
5 import { UserService } from '../services/user.service';
6
7 @Component({
8   selector: 'app-login',
9   templateUrl: './login.component.html',
10  styleUrls: ['./login.component.css']
11 })
12 export class LoginComponent implements OnInit {
13
14   constructor(private fb:FormBuilder,
15    private userservice:UserService,
16    private router:Router,private snackbar:MatSnackBar) { }
17
18   loginform = this.fb.group({
19     email:['',[Validators.required]],
20     password:['',[Validators.required]],
21   })
22   get email():FormControl{
23     return this.loginform.get('email') as FormControl
24   }
25   get password():FormControl{
26     return this.loginform.get('password') as FormControl
27   }
28
29   dologin(){
30     let email=this.email.value
```

```
src > app > user > login > TS login.component.ts > LoginComponent > dologin > next
28
29   let email=this.email.value
30   let pass=this.password.value
31
32   this.userservice.login(email,pass).subscribe({
33
34     next:user=>{
35       if(user.length>0)
36       {
37         this.snackbar.open('Login successfull..',undefined,
38         {duration:5000})
39         this.router.navigate(['./user'])
40       }
41       else
42       {
43         this.snackbar.open('Login unsuccessfull....',undefined,
44         {duration:5000})
45       }
46     }
47   })
48
49
50
51
52
53
54
55   ngOnInit(): void {
56 }
```

Ln 15, Col 5 Spaces: 2 UTF-8 LF {} TypeScript

LOGIN.COMPONENT.CSS

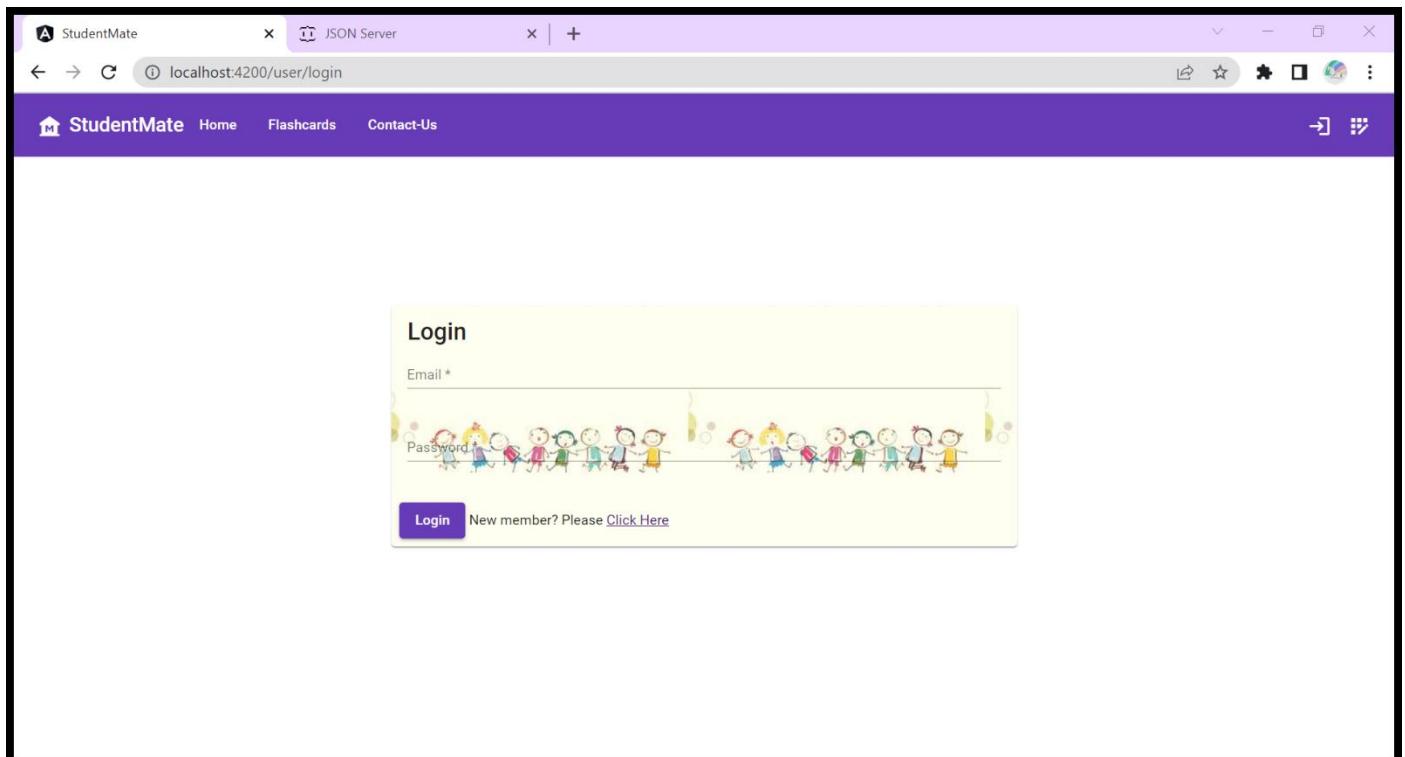


A screenshot of Visual Studio Code showing the `login.component.css` file. The code uses Flexbox and CSS Grid for layout, and includes a media query for smaller screens. A background image URL is present in the code.

```
1 .login-form-container{  
2     display: flex;  
3     flex-direction: column;  
4     /* Horizontal Center */  
5     align-items: center;  
6     /* Vertical Center */  
7     justify-content: center;  
8     height: 100%;  
9 }  
10 .login-card{  
11     width: 30rem;  
12     margin-top: 150px;  
13     background-image: url("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9Gc5Iu3lzqMemxmsYPPEGxTAhwDa5N0riG_Fimg&usqp=CAU");  
14 }  
15 .mat-card-content{  
16     display: flex;  
17     flex-direction: column;  
18 }  
19  
20 @media screen and (max-width:1000px) {  
21     .login-card{  
22         width:15rem  
23     }  
24 }  
25  
26  
27
```

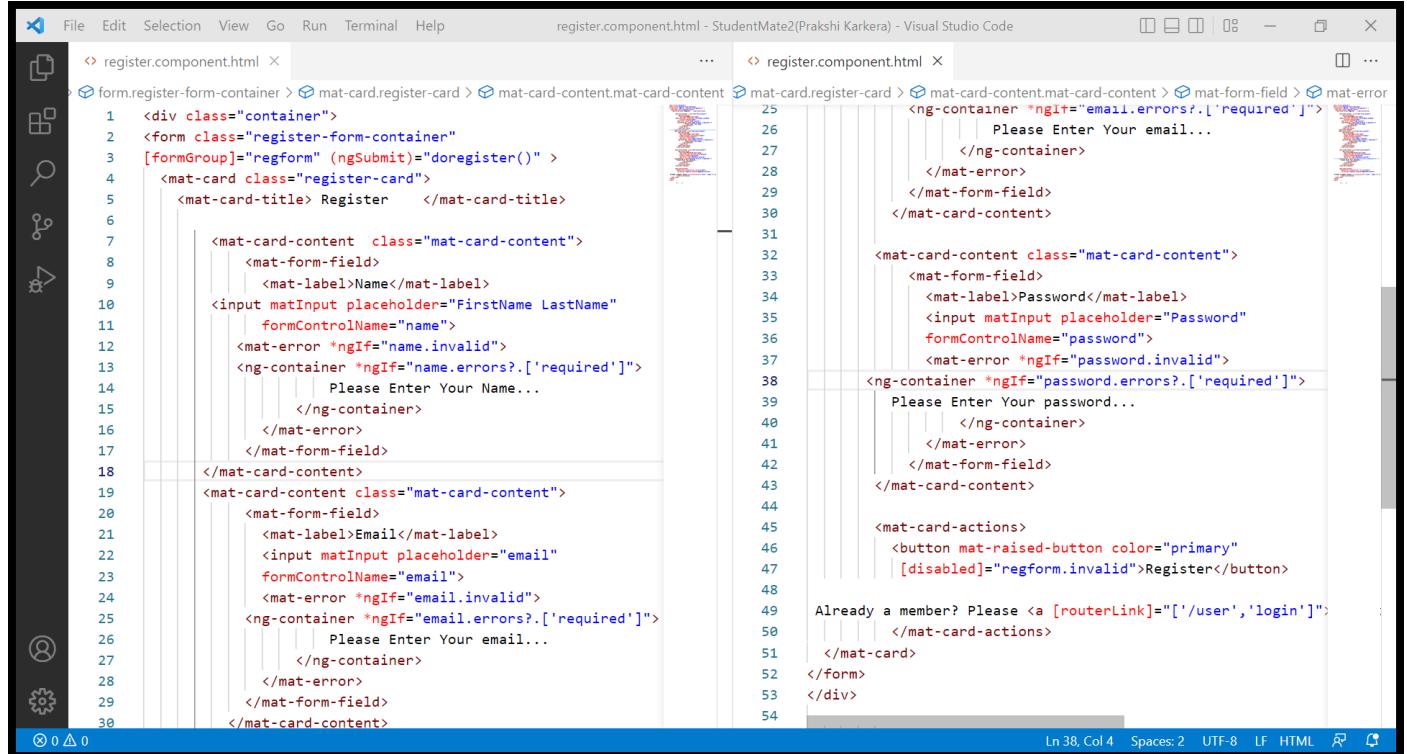
Ln 1, Col 1 | Spaces: 4 | UTF-8 | CRLF | CSS |  

LOGIN PAGE



5) Register

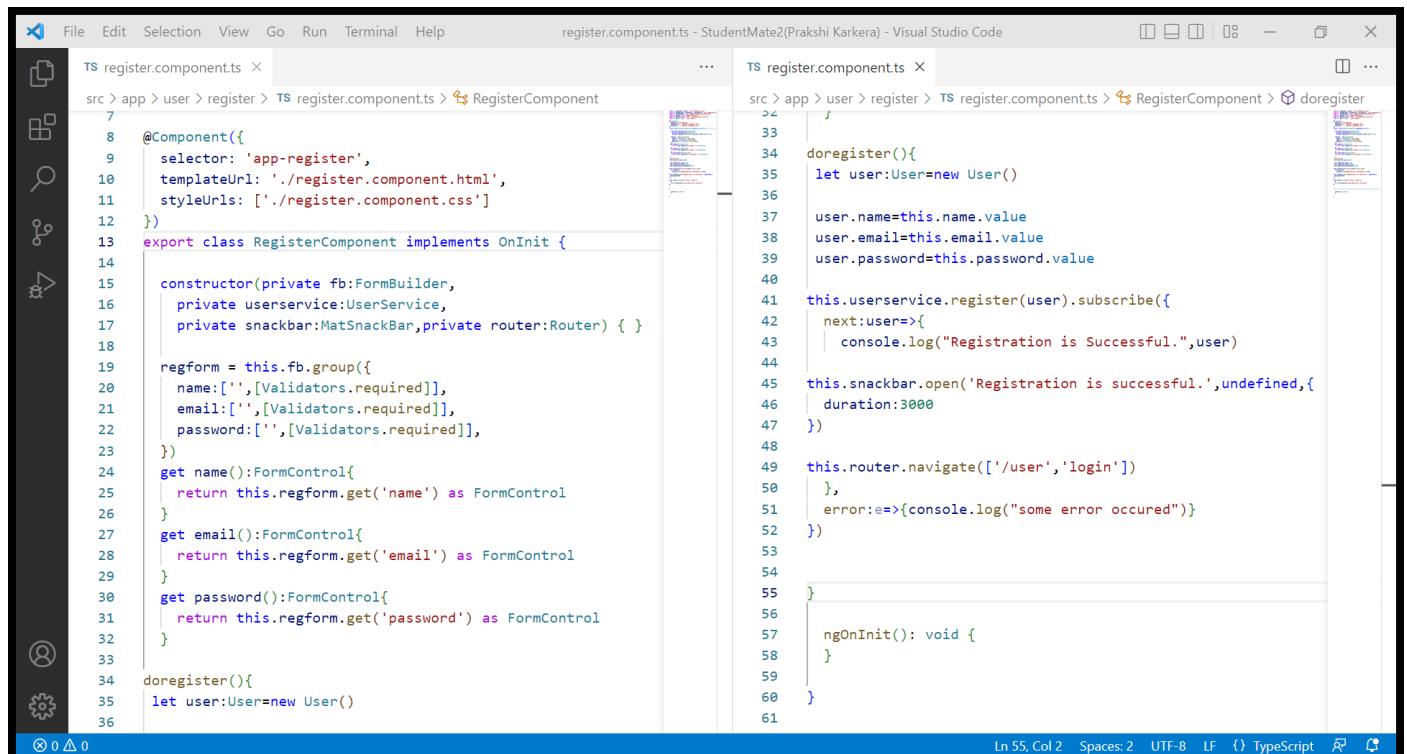
REGISTER.COMPONENT.HTML



```
register.component.html
<div class="container">
<form class="register-form-container" [formGroup]="regform" (ngSubmit)="doregister()">
  <mat-card class="register-card">
    <mat-card-title> Register </mat-card-title>
    <mat-card-content class="mat-card-content">
      <mat-form-field>
        <mat-label>Name</mat-label>
        <input matInput placeholder="FirstName LastName" formControlName="name">
        <mat-error *ngIf="name.invalid">
          Please Enter Your Name...
        </mat-error>
      </mat-form-field>
    </mat-card-content>
    <mat-card-content class="mat-card-content">
      <mat-form-field>
        <mat-label>Email</mat-label>
        <input matInput placeholder="email" formControlName="email">
        <mat-error *ngIf="email.invalid">
          Please Enter Your email...
        </mat-error>
      </mat-form-field>
    </mat-card-content>
  </mat-card>
</form>
```

```
register.component.html
<mat-card.register-card>
  <mat-card-content mat-card-content>
    <mat-form-field>
      <ng-container *ngIf="email.errors?.['required']">
        Please Enter Your email...
      </ng-container>
    </mat-form-field>
  </mat-card-content>
  <mat-card-content mat-card-content>
    <mat-form-field>
      <mat-label>Password</mat-label>
      <input matInput placeholder="Password" formControlName="password">
      <mat-error *ngIf="password.invalid">
        Please Enter Your password...
      </mat-error>
    </mat-form-field>
  </mat-card-content>
  <mat-card-actions>
    <button mat-raised-button color="primary" [disabled]="regform.invalid">Register</button>
  </mat-card-actions>
</mat-card>
```

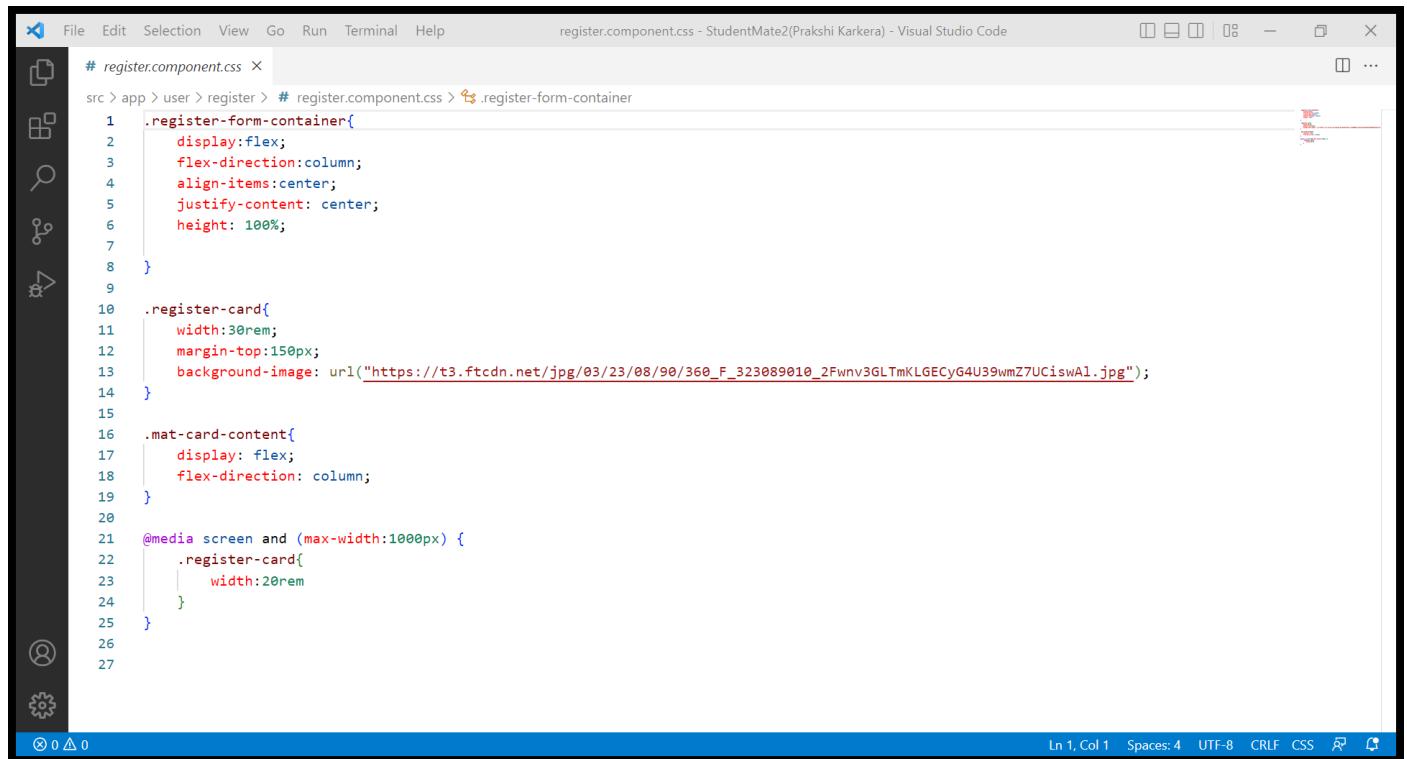
REGISTER.COMPONENT.TS



```
register.component.ts
@Component({
  selector: 'app-register',
  templateUrl: './register.component.html',
  styleUrls: ['./register.component.css']
})
export class RegisterComponent implements OnInit {
  regform = this.fb.group({
    name:[ '',[Validators.required]],
    email:[ '',[Validators.required]],
    password:[ '',[Validators.required]],
  })
  get name():FormControl{
    return this.regform.get('name') as FormControl
  }
  get email():FormControl{
    return this.regform.get('email') as FormControl
  }
  get password():FormControl{
    return this.regform.get('password') as FormControl
  }
  doregister(){
    let user:User=new User()
    user.name=this.name.value
    user.email=this.email.value
    user.password=this.password.value
    this.userservice.register(user).subscribe({
      next:user=>{
        console.log("Registration is Successful.",user)
        this.snackbar.open('Registration is successful.',undefined,{duration:3000})
        this.router.navigate(['/user','login'])
      },
      error:e=>{console.log("some error occurred")}
    })
  }
  ngOnInit(): void {
  }
}
```

```
register.component.ts
doregister(){
  let user:User=new User()
  user.name=this.name.value
  user.email=this.email.value
  user.password=this.password.value
  this.userservice.register(user).subscribe({
    next:user=>{
      console.log("Registration is Successful.",user)
      this.snackbar.open('Registration is successful.',undefined,{duration:3000})
      this.router.navigate(['/user','login'])
    },
    error:e=>{console.log("some error occurred")}
  })
}
```

REGISTER.COMPONENT.CSS

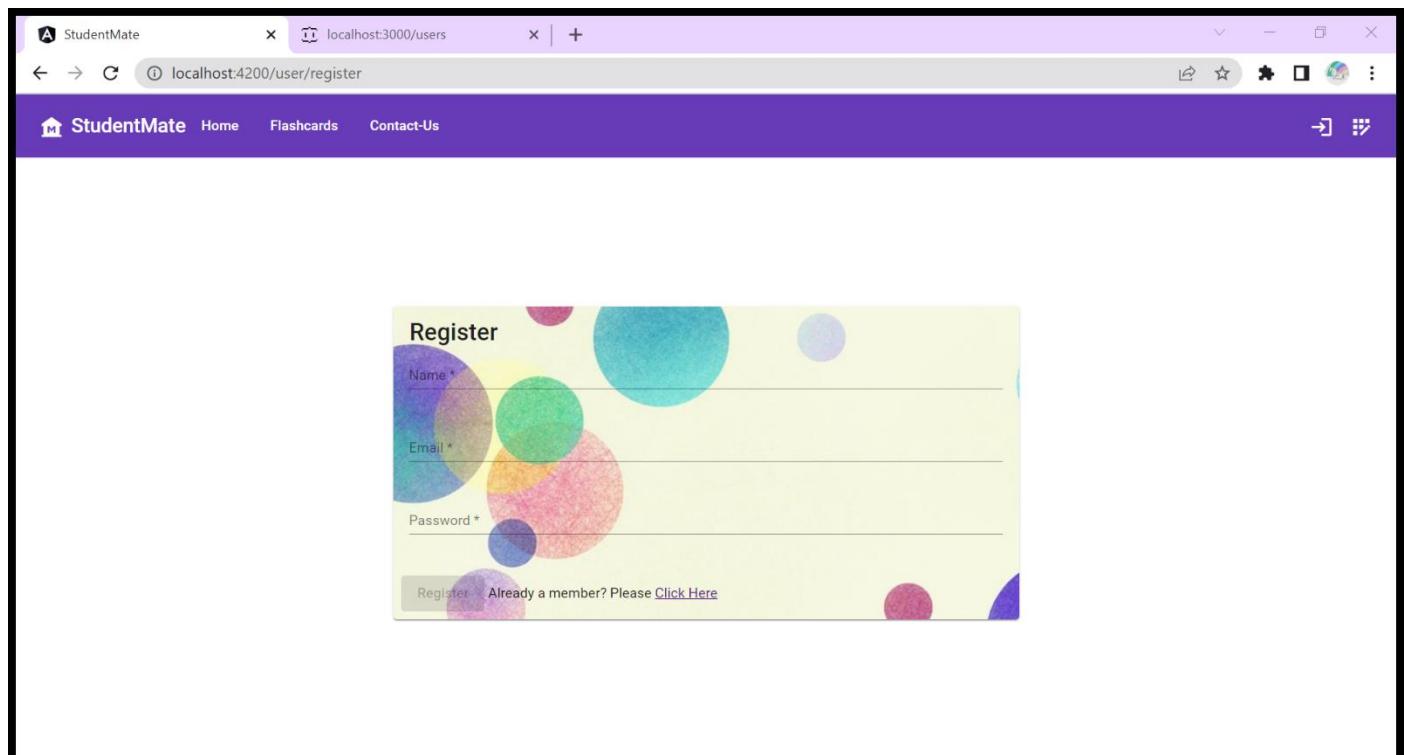


A screenshot of Visual Studio Code showing the `register.component.css` file. The code defines styles for a registration form. It includes a main container with flex properties, a card with a background image, and a media query for smaller screens. The background image URL is a placeholder.

```
# register.component.css
src > app > user > register > # register.component.css > .register-form-container
1 .register-form-container{
2   display:flex;
3   flex-direction:column;
4   align-items:center;
5   justify-content: center;
6   height: 100%;
7 }
8
9
10 .register-card{
11   width:30rem;
12   margin-top:150px;
13   background-image: url("https://t3.ftcdn.net/jpg/03/23/08/90/360_F_323089010_2Fwnv3GLTmLGECyG4U39wmZ7UCiswA1.jpg");
14 }
15
16 .mat-card-content{
17   display: flex;
18   flex-direction: column;
19 }
20
21 @media screen and (max-width:1000px) {
22   .register-card{
23     width:20rem
24   }
25 }
26
27
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF CSS

REGISTER PAGE



USER.MODEL.TS

The screenshot shows the Visual Studio Code interface with the file `user.model.ts` open. The code defines a class `User` with properties: `id:number=0`, `name:string=""`, `email:string=""`, and `password:string=""`. The code editor has syntax highlighting for TypeScript. The Explorer sidebar on the left shows the project structure under `STUDENTMATE2(PRAKSHI ...)`, including `user`, `login`, `register`, `services`, and `app-routing.module.ts`. The bottom status bar indicates `Ln 1, Col 1`, `Spaces: 4`, `UTF-8`, `CRLF`, `TypeScript`, and other icons.

```
1 export class User{  
2     id:number=0  
3     name:string=""  
4     email:string=""  
5     password:string=""  
6 }
```

NAVBAR.COMPONENT.HTML

The screenshot shows the Visual Studio Code interface with the file `navbar.component.html` open. The code defines a `<mat-toolbar color="primary">` element containing several `<button mat-icon-button>` elements. These buttons have `aria-label` attributes like "Example icon-button with menu icon", "Example icon-button with heart icon", and "Example icon-button with share icon". They also have `[routerLink]` attributes pointing to routes such as `['/home']`, `['/user/', 'login']`, and `['/user/', 'register']`. The code editor has syntax highlighting for HTML and CSS classes like `example-icon`, `example-spacer`, and `favorite-icon`. The Explorer sidebar on the left shows the project structure under `STUDENTMATE2(PRAKSHI ...)`, including `app`, `navbar`, and `app-routing.module.ts`. The bottom status bar indicates `Ln 1, Col 1`, `Spaces: 2`, `UTF-8`, `LF`, `HTML`, and other icons.

```
1 <mat-toolbar color="primary">  
2     <button mat-icon-button class="example-icon" aria-label="Example icon-button with menu icon" [routerLink]=["'/home']">  
3         <mat-icon>museum</mat-icon>  
4     </button>  
5     <span>StudentMate</span>  
6     <a mat-button [routerLink]=["'/home']>Home</a>  
7     <a mat-button [routerLink]=["'/cards','list']>Flashcards</a>  
8     <a mat-button [routerLink]=["'/contactus']>Contact-Us</a>  
9  
10    <span class="example-spacer"></span>  
11  
12    <button mat-icon-button class="example-icon favorite-icon" aria-label="Example icon-button with heart icon"  
13        [routerLink]=["'/user/','login']" >  
14        <mat-icon>login</mat-icon>  
15    </button>  
16    <button mat-icon-button class="example-icon" aria-label="Example icon-button with share icon"  
17        [routerLink]=["'/user/','register']">  
18        <mat-icon>app_registration</mat-icon>  
19    </button>  
20 </mat-toolbar>
```

MYCARDS.JSON

```
mycards.json - StudentMate2(Prakshi Karkera) - Visual Studio Code
```

```
mycards.json
mycards.json > [ ] users > {} 2
1 {
2   "users": [
3     {
4       "id": 1,
5       "name": "ewe",
6       "email": "adf@ghjk.com",
7       "password": "456789"
8     },
9     {
10       "id": 2,
11       "name": "Happy",
12       "email": "pkkarkera23@gmail.com",
13       "password": "123456"
14     },
15     {
16       "id": 3,
17       "name": "name",
18       "email": "isha@gmail.com",
19       "password": "123"
20     },
21     {
22       "id": 4,
23       "name": "Asmi",
24       "email": "asm@gmail.com",
25       "password": "12345"
26     },
27     {
28       "id": 5,
29       "name": "Krish Kotian",
30       "email": "krish@gmail.com",
31     }
32   ]
33 }
```

```
mycards.json
mycards.json > [ ] users > {} 2
34   ],
35   "cards": [
36     {
37       "name": "A-Z FLASHCARDS SET",
38       "image": "assets/a-z.jpg",
39       "price": 129,
40       "content": "With cute attractive"
41     },
42     {
43       "name": "1-10 FLASHCARDS SET",
44       "image": "assets/numbers.jpg",
45       "price": 129,
46       "content": "Learn numbers with 1"
47     },
48     {
49       "name": "ACTIONS FLASHCARDS SET",
50       "image": "assets/actions.png",
51       "price": 139,
52       "content": "With cute attractive"
53     },
54     {
55       "name": "DIVISION FLASHCARDS SET",
56       "image": "assets/division.jpg",
57       "price": 199,
58       "content": "Get perfect in division"
59     },
60     {
61       "name": "MY FIRST FLASHCARDS SET",
62       "image": "assets/My-Firstcards.j
63       "price": 249,
64     }
65   ]
66 }
```

```
mycards.json
mycards.json > [ ] users > {} 2
67   ],
68   "messages": [
69     {
70       "id": 1,
71       "name": "isha",
72       "email": "isha@gmail.com",
73       "message": "I love it !!"
74     },
75     {
76       "id": 2,
77       "name": "Asmi",
78       "email": "asm@gmail.com",
79       "message": "We play lot of games"
80     }
81   ]
82 }
```

Ln 18, Col 33 Spaces: 2 UTF-8 LF () JSON



Congrats!

You're successfully running JSON Server
♪(*'▽')*

Resources

[/users](#) 6x
[/cards](#) 8x
[/messages](#) 2x

To access and modify resources, you can use any HTTP method:

[GET](#) [POST](#) [PUT](#) [PATCH](#) [DELETE](#) [OPTIONS](#)

undefined

Thank you