

Kreiranje komponente

Kreiranje klase

ng g class osoba

```
export class Osoba {  
  constructor(public osobaId: number, public ime: string, public prezime: string) {  
  }  
}
```

Kreiranje komponente

```
ng g c listaOsoba
```

```
ng g c listaOsoba --standalone
```

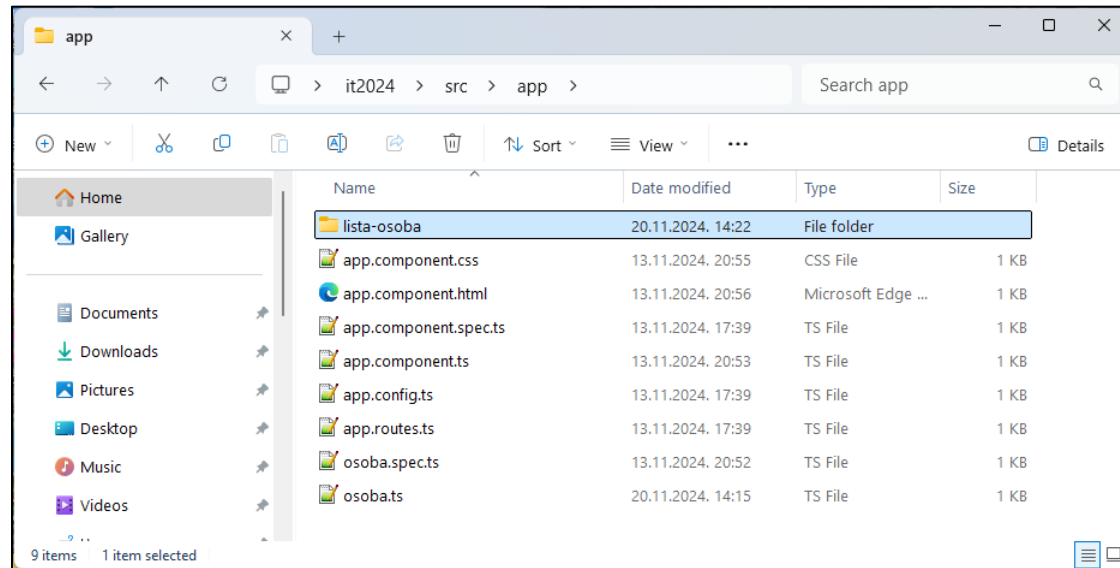
```
import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';

@Component({
  selector: 'lista-osoba',
  standalone: true,
  imports: [CommonModule],
  templateUrl: './lista-osoba.component.html',
  styleUrls: ['./lista-osoba.component.css']
})
export class ListaOsobaComponent {

}
```

```
<p>lista-osoba works!</p>
```

Folder aplikacije

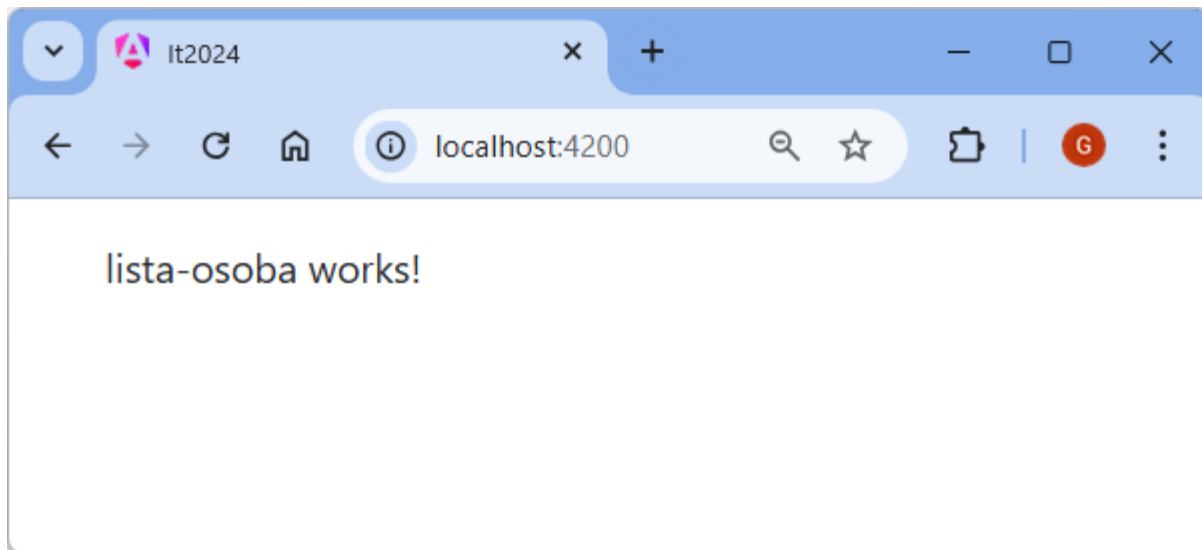


Glavna komponenta

```
import { Component } from '@angular/core';
import { ListaOsobaComponent } from './lista-osoba/lista-osoba.component';
@Component({
  selector: 'app-root',
  standalone: true,
  imports: [ListaOsobaComponent],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
}
```

Glavna komponenta - pogled

```
<div class="container mt-3">
  <div class="row">
    <div class="col-md-6">
      <lista-osoba></lista-osoba>
    </div>
  </div>
</div>
```

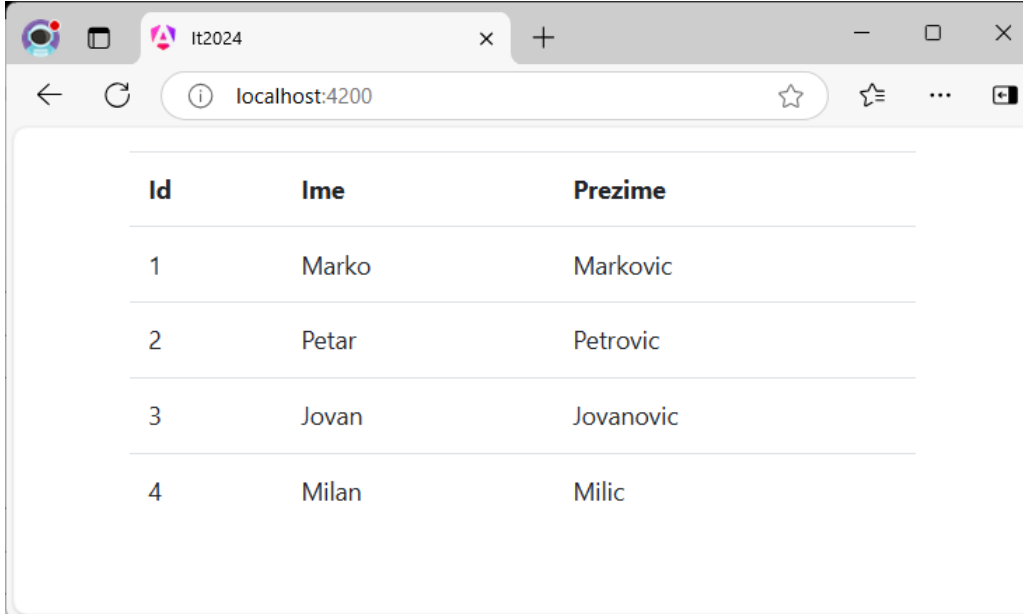


Polje osobe klase ListaOsobaComponent

```
export class ListaOsobaComponent {  
  osobe: Osoba[] = [  
    new Osoba(1, 'Marko', 'Markovic'),  
    new Osoba(2, 'Petar', 'Petrovic'),  
    new Osoba(3, 'Jovan', 'Jovanovic'),  
    new Osoba(4, 'Milan', 'Milic')  
  ];  
}
```

Šablon komponente

```
<table class="table">
  <tr>
    <th>Id</th>
    <th>Ime</th>
    <th>Prezime</th>
  </tr>
  <tr *ngFor="let osoba of osobe">
    <td>{{osoba.osobaId}}</td>
    <td>{{osoba.ime}}</td>
    <td>{{osoba.prezime}}</td>
  </tr>
</table>
```



The screenshot shows a web browser window with the URL localhost:4200. The browser displays a table with the following data:

Id	Ime	Prezime
1	Marko	Markovic
2	Petar	Petrovic
3	Jovan	Jovanovic
4	Milan	Milic

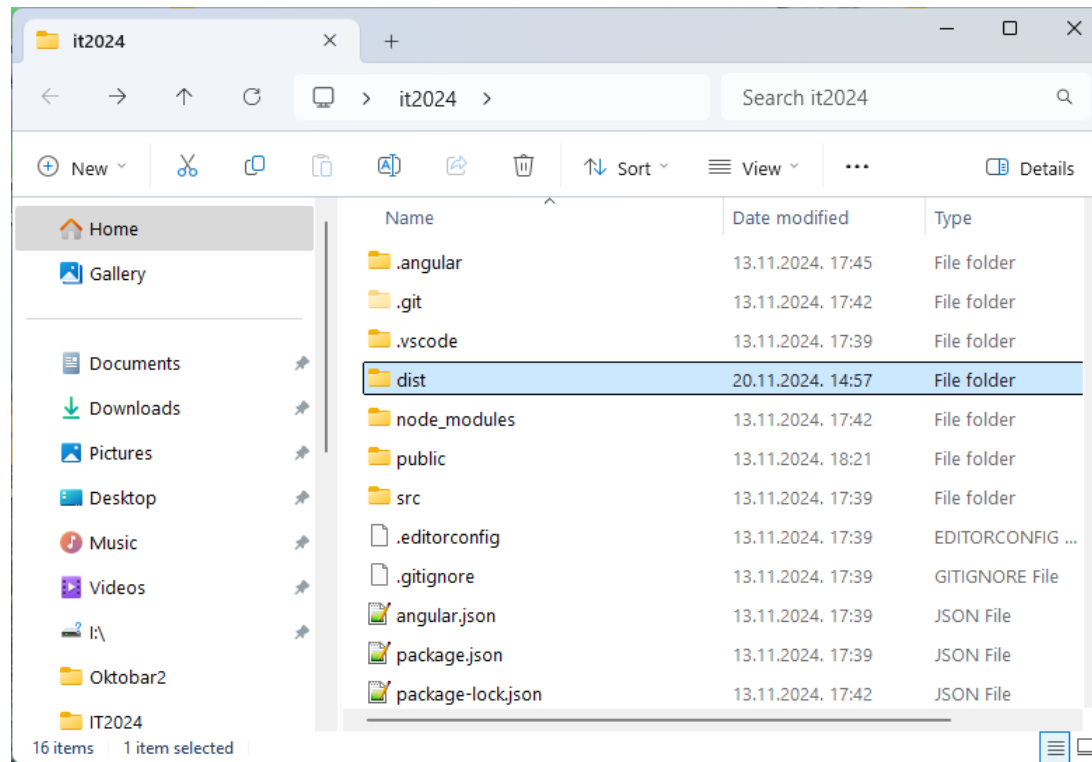
Razmeštanje angular aplikacije

- Unutar fajla index.html (koji je smeštaj u src folderu) u head delu promeniti base direktivu pre razmeštanja na web server:

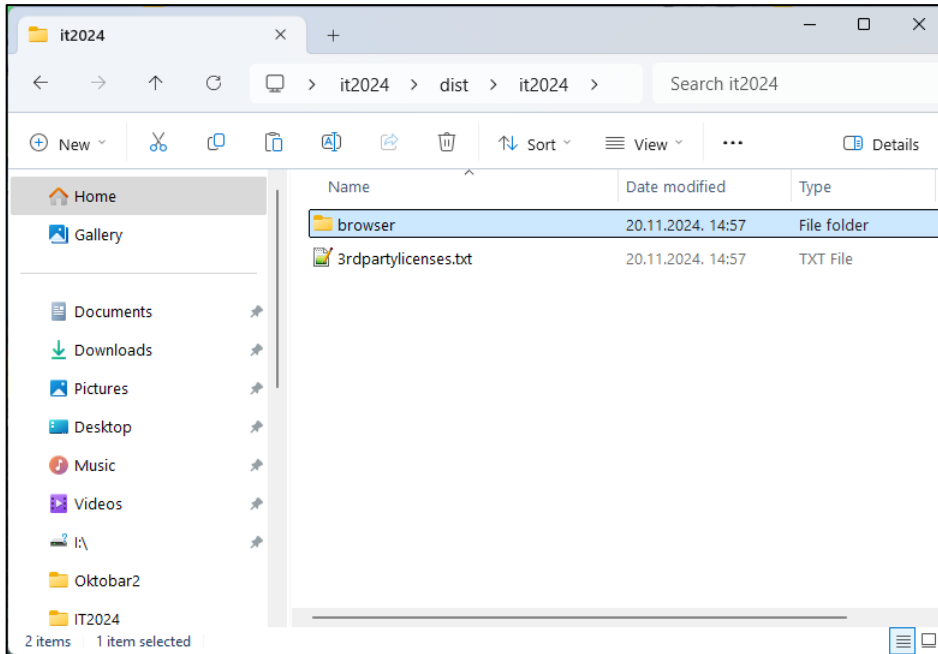
```
<base href="./">
```

- U terminalu kucamo ng build
- Kreiran je dist folder
- Prekopiramo sadržaj ovog foldera na web server

Folder dist – mesto browser aplikacije

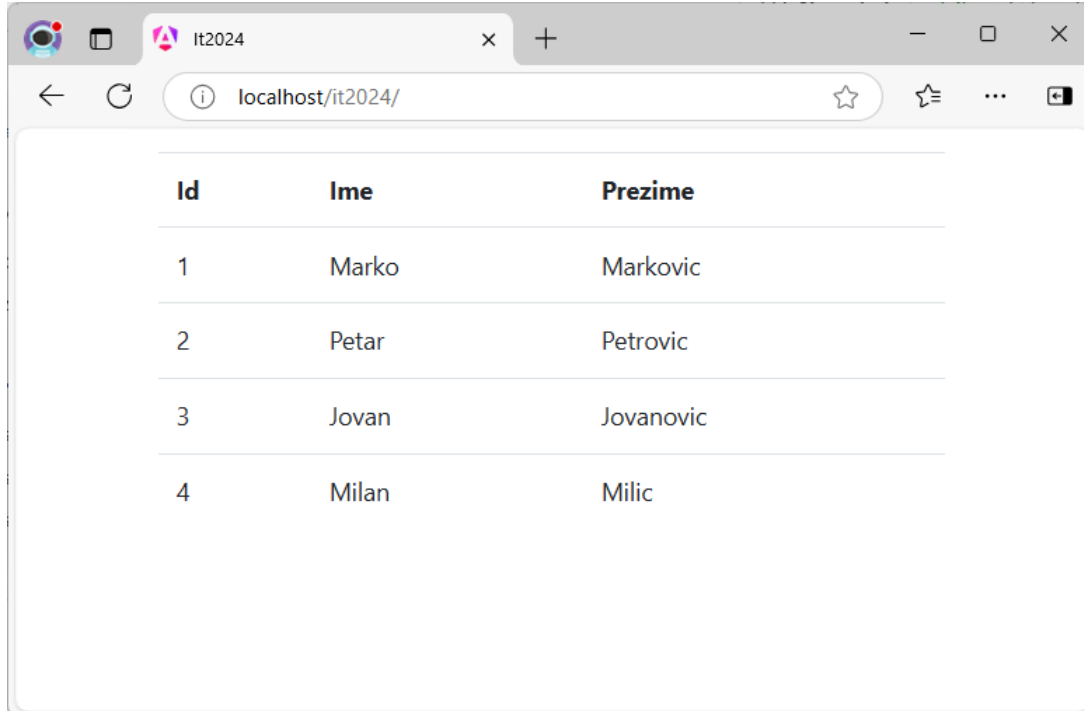


Folder browser



Folder browser se preimenje i kopira na web server

Razmeštanje na lokalni apache web server



A screenshot of a web browser window. The address bar shows 'localhost/it2024/'. The page content is a table with three columns: 'Id', 'Ime', and 'Prezime'. The table contains four rows of data.

Id	Ime	Prezime
1	Marko	Markovic
2	Petar	Petrovic
3	Jovan	Jovanovic
4	Milan	Milic

Komunikacija komponenti

Komunikacija komponenti (parent -> child)

- Roditeljska komponenta komunicira sa child komponentom postavljajući njen properti
- Child komponenta importuje @Input interfejs iz biblioteke @angular/Core
- Properti child komponente koji vrednost dobija od parent komponente označava sa @Input dekoratorom

```
@Input() childValue: number;
```

Kreiranje child komponente

ng g c child

```
import { Component, Input } from '@angular/core';

@Component({
  selector: 'app-child',
  standalone: true,
  imports: [],
  templateUrl: './child.component.html',
  styleUrls: ['./child.component.css']
})

export class ChildComponent {
  @Input() childValue: number = -1;
}
```

Šablon child komponente

```
<hr>  
<h2>Child komponenta</h2>  
Tekuca vrednost brojaca je: {{childValue}}
```


Parent komponenta

```
import { Component } from '@angular/core';
import { ChildComponent } from
'../child/child.component';

@Component({
  selector: 'app-parent',
  standalone: true,
  imports: [ChildComponent],
  templateUrl: './parent.component.html',
  styleUrls: ['./parent.component.css']
})
export class ParentComponent {
  naslov = 'Komunikacija parent-child';
  parentValue = 5;

  uvecaj(): void {
    this.parentValue++;
  }

  umanji(): void {
    this.parentValue--;
  }
}
```


Komunikacija komponenti

Komunikacija parent-child!

Uvecaj

Umanji

Child komponenta

Tekuća vrednost brojača je: 6

Child komponenta šalje podatke parent komponenti

- U child komponenti potrebno je importovati Input, Output interfejsa i EventEmitter klasu iz modula @angular/core
- Da bi generisala događaj child komponenta treba da definiše svojstvo tipa EventEmitter
- Svojstvo tipa EventEmitter se opisuje dekoratorom @Output
- Događaj se generiše pozivom **emit** metode kojoj se prosleđuje vrednost za parent komponentu
- Parent komponenta vrši pretplatu na događaj definisan u child komponenti i definiše handlersku funkciju za taj događaj
- Pretplata na događaj vrši se tehnikom event bindinga

Child komponenta

```
export class Child1Component {  
  
    childValue: number = -1;  
  
    // childValueChange event  
    @Output() childValueChange = new EventEmitter();  
  
    uvecaj() {  
        this.childValue++;  
        this.childValueChange.emit(this.childValue);  
    }  
    umanji() {  
        this.childValue--;  
        this.childValueChange.emit(this.childValue);  
    }  
}
```

Šablon child komponente

```
<hr>  
<h2>Child komponenta</h2>  
  <button (click)="uvecaj()">Uvecaj</button>  
  <button (click)="umanji()">Umanji</button>  
<br>
```

Parent komponenta

```
export class Parent1Component {  
  
  naslov = 'Komunikacija child-parent';  
  parentValue = 5;  
  
  childValueChangeHandler(event: number) {  
    this.parentValue = event;  
  }  
  
}
```

```
<h1>{{ naslov }}!</h1>  
<p>Trenutni broj klikova: {{ parentValue }}</p>  
<app-child1 (childValueChange)="childValueChangeHandler($event)"></app-child1>
```

Parent komponenta pretplaćena na događaj child komponente

Rezultat prosleđivanja podataka

Komunikacija child-parent!

Trenutni broj klikova: 5

Child komponenta

Uvecaj

Umanji

Komunikacija child-parent!

Trenutni broj klikova: 0

Child komponenta

Uvecaj

Umanji

Primer komunikacije komponenti -1

ng g class osoba

```
export class Osoba {  
  constructor(public osobaId: number, public ime: string, public prezime: string){  
  }  
}
```

Primer komunikacije komponenti -2

```
export class OsobeComponent {  
  
  title = 'Primer komunikacije komponenti';  
  osobe: Osoba[] = [  
    new Osoba(1, 'Marko', 'Markovic'),  
    new Osoba(2, 'Petar', 'Petrovic'),  
    new Osoba(3, 'Jovan', 'Jovanovic'),  
    new Osoba(4, 'Milan', 'Mirkovic')  
  ];  
  selektovanaOsoba: Osoba = new Osoba(-1, '',  
  '');  
  
  selektuj(os: Osoba): void {  
    this.selektovanaOsoba = os;  
  }  
}
```

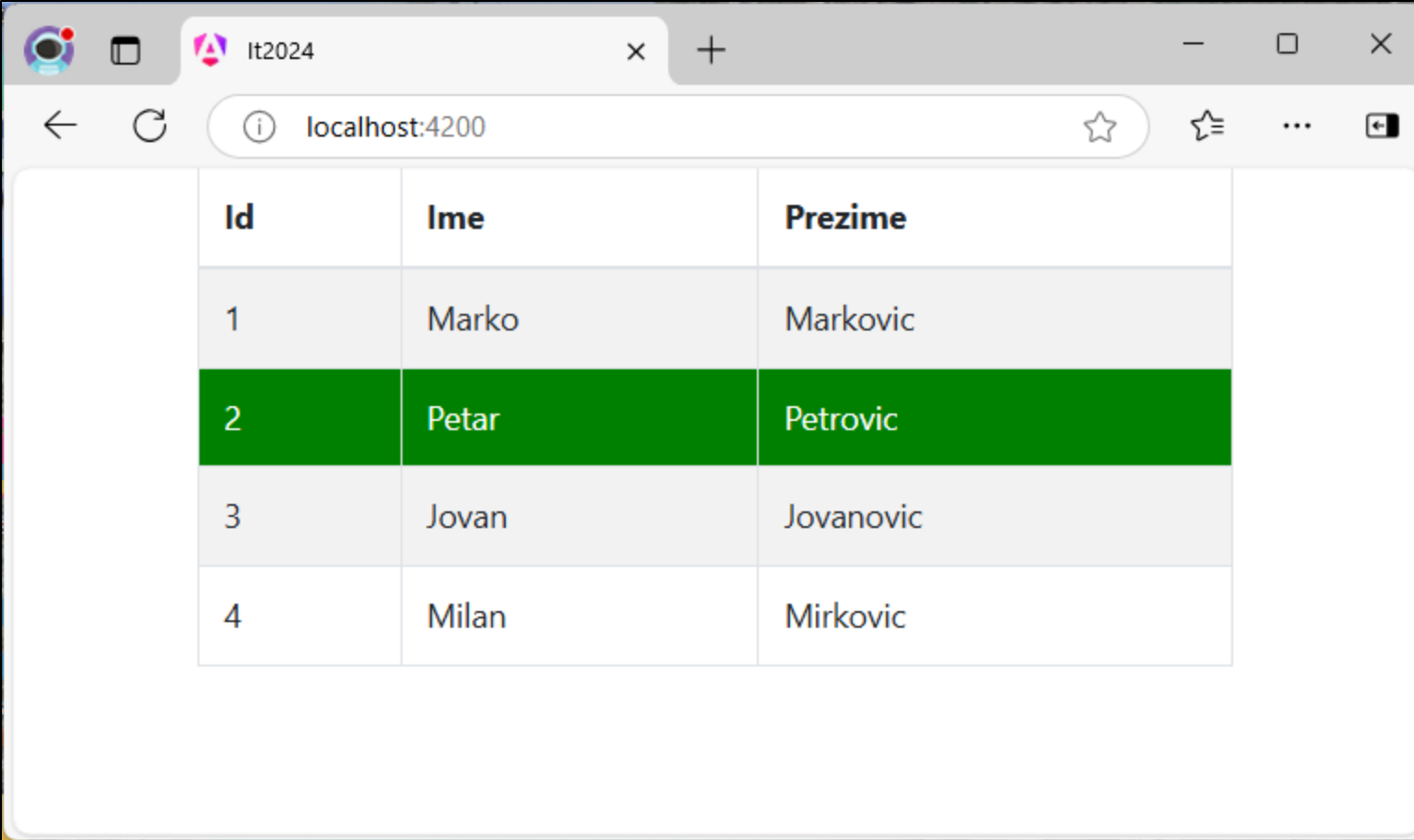
```
.selektujRed {  
  background-color: green !important;  
  color: white;  
}
```

css fajl komponente

Primer komunikacije komponenti -3

```
<div class="row">
  <div class="col-md-7">
    <table class="table table-bordered table-striped">
      <thead>
        <tr>
          <th>Id</th>
          <th>Ime</th>
          <th>Prezime</th>
        </tr>
      </thead>
      <tbody>
        <tr *ngFor="let osoba of osobe" [ngClass]="{ 'selektujRed': osoba === selektovanaOsoba }"
(click)="selektuj(osoba)">
          <td>{{ osoba.osobaId }}</td>
          <td>{{ osoba.ime }}</td>
          <td>{{ osoba.prezime }}</td>
        </tr>
      </tbody>
    </table>
  </div>
</div>
```

Primer komunikacije komponenti -4



The image shows a web browser window with a single tab titled 'It2024'. The address bar displays 'localhost:4200'. The main content area contains a table with the following data:

Id	Ime	Prezime
1	Marko	Markovic
2	Petar	Petrovic
3	Jovan	Jovanovic
4	Milan	Mirkovic

Primer komunikacije komponenti -5

ng g c osobaDetalji

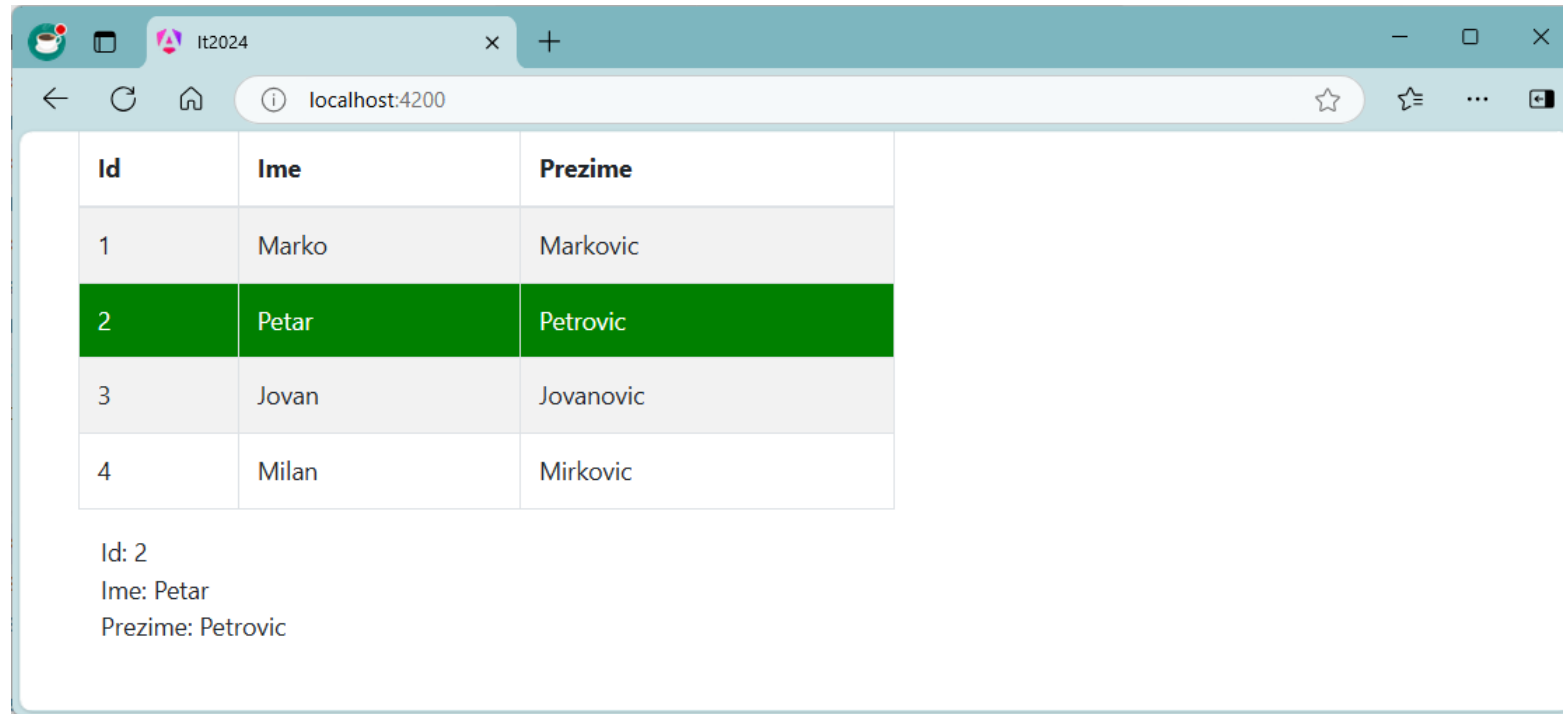
```
export class OsobaDetaljiComponent {  
  
  @Input() osoba: Osoba = new Osoba(-1, '', '');  
  
}
```

```
<div class="col-md-4" *ngIf="osoba.osobaId !=-1">  
  Id: {{ osoba.osobaId }} <br>  
  Ime: {{ osoba.ime }} <br>  
  Prezime: {{ osoba.prezime }}  
</div>
```

Poziv child komponente

```
<div class="row">  
  <div class="col-md-7">  
    <app-osoba-detalji [osoba]=selektovanaOsoba></app-osoba-detalji>  
  </div>  
</div>
```

Komunikacija komponenti



The screenshot shows a web browser window with a single tab titled 'It2024'. The address bar displays 'localhost:4200'. The main content area contains a table with three columns: 'Id', 'Ime', and 'Prezime'. The second row, representing 'Petar Petrovic', is highlighted in green. Below the table, the details for the selected row are displayed: 'Id: 2', 'Ime: Petar', and 'Prezime: Petrovic'.

Id	Ime	Prezime
1	Marko	Markovic
2	Petar	Petrovic
3	Jovan	Jovanovic
4	Milan	Mirkovic

Id: 2
Ime: Petar
Prezime: Petrovic

Pitanje 1

Potrebno je kreirati komponentu OsobeComponent korišćenjem angular CLI alata. Koju komandu treba napisati u terminalu:

- a. `ng g c osobe`
- b. `node g component osobe`
- c. `ng g c OsobeComponent`

Odgovor: a

Pitanje 2

Child komponenta ima selektor **osoba-detalji** i ima ulazni property pod nazivom **osoba** koji je tipa klase **Osoba**. Na koji način parent komponenta poziva child komponentu i prosleđuje joj objekat tipa **Osoba** pod nazivom **selektovanaOsoba**

- a. `<osoba-detalji [child]=selektovanaOsoba></osoba-detalji>`
- b. `<osoba-detalji [osoba]="selektovanaOsoba"></osoba-detalji>`
- c. `<child [emit]="selektovanaOsoba"></child>`

Odgovor: b

Pitanje 3

Child komponenta treba da pošalje podatke parent komponenti generisanjem događaja. Da bi se to postiglo u child komponenti se definiše izlazno svojstvo koje je tipa

- a. EventEmitter
- b. Event
- c. Observable

Odgovor: a

Pitanje 4

Properti child komponente koji vrednost dobija od parent komponente označava sa dekoratorom:

- a. @Input()
- b. @Child()
- c. @Parent()

Odgovor: a

Pitanje 5

Child komponenta treba da pošalje podatke parent komponenti generisanjem događaja. Da bi se to postiglo u child komponenti se definiše svojstvo tipa EventEmitter koga treba opisati dekoratorom

- a. @Output()
- b. @Input()
- c. @Emmit()

Odgovor: a