

## Day 21 Angular Communication Assignment - README

This project implements the Day 21 Angular Communication assignment using the Standalone Component approach in Angular.

It demonstrates parent-to-child communication using `@Input()` and child-to-parent communication using `@Output()` and `EventEmitter`.

Application Name: dashProd

Components:

1. App (Root Component)
2. Dashboard (Parent Component)
3. ProductDetails (Child Component)

Communication Flow:

- Dashboard displays a list of products.
- When a user selects a product, its details are passed to ProductDetails via `@Input()`.
- ProductDetails allows the user to enter feedback.
- When submitted, feedback is emitted via `@Output()` to the Dashboard, which stores and displays it.

Folder Structure:

src/

??? app/

? ??? app.ts (Root standalone component)

? ??? app.html

? ??? dashboard/

? ? ??? dashboard.ts

? ? ??? dashboard.html

? ? ??? dashboard.css

? ??? product-details/

? ??? product-details.ts

? ??? product-details.html

? ??? product-details.css

## How to Run:

1. Ensure Angular CLI is installed: `npm install -g @angular/cli`
2. Create a new Angular project: `ng new dashProd --standalone`
3. Replace the generated files with the provided assignment files.
4. Run the application: `ng serve`
5. Open `http://localhost:4200` in a browser.

## Technologies Used:

- Angular (Standalone Components)
- HTML, CSS
- TypeScript

## Key Angular Features Demonstrated:

- `@Input()` for parent-to-child communication
- `@Output()` + `EventEmitter` for child-to-parent communication
- `*ngIf` and `*ngFor` structural directives
- `FormsModule` for two-way binding with `[(ngModel)]`

## Main Files:

app.ts:

```
-----  
  
import { Component, signal } from '@angular/core';  
import { RouterOutlet } from '@angular/router';  
import { Dashboard } from '../dashboard/dashboard';  
  
@Component({  
  selector: 'app-root',  
  standalone: true,  
  imports: [RouterOutlet, Dashboard],  
  templateUrl: './app.html',  
  styleUrls: ['./app.css']  
})
```

```
export class App {  
  protected readonly title = signal('dashProd');  
}
```

app.html:

```
-----  
<app-dashboard></app-dashboard>
```

dashboard.ts:

```
-----  
import { CommonModule } from '@angular/common';  
import { Component } from '@angular/core';  
import { ProductDetails } from '../product-details/product-details';
```

```
@Component({  
  selector: 'app-dashboard',  
  standalone: true,  
  imports: [CommonModule, ProductDetails],  
  templateUrl: './dashboard.html',  
  styleUrls: ['./dashboard.css']  
})
```

```
export class Dashboard {  
  products = [  
    { name: 'Laptop', price: 50000 },  
    { name: 'Mobile', price: 40000 },  
    { name: 'Tablet', price: 35000 }  
  ];
```

```
  selectedProduct: any = null;
```

```
  feedbackList: string[] = [];
```

```
  selectProduct(product: any) {  
    this.selectedProduct = product;
```

```

    }

    receiveFeedback(feedback: string) {
        this.feedbackList.push(feedback);
    }
}

```

dashboard.html:

```

-----

<h2>Product List</h2>
<ul>
    <li *ngFor="let product of products">
        {{product.name}} - ${{product.price}}
        <button (click)="selectProduct(product)">View More</button>
    </li>
</ul>

<app-product-details
    *ngIf="selectedProduct"
    [product]="selectedProduct"
    (feedbackEvent)="receiveFeedback($event)">
</app-product-details>

<h3>Feedback Received:</h3>
<ul>
    <li *ngFor="let fb of feedbackList">{{fb}}</li>
</ul>

```

product-details.ts:

```

-----

import { CommonModule } from '@angular/common';
import { Component, EventEmitter, Input, Output } from '@angular/core';
import { FormsModule } from '@angular/forms';

```

```

@Component({
  selector: 'app-product-details',
  standalone: true,
  imports: [CommonModule, FormsModule],
  templateUrl: './product-details.html',
  styleUrls: ['./product-details.css']
})

export class ProductDetails {
  @Input() product: any;
  @Output() feedbackEvent = new EventEmitter<string>();

  feedbackText: string = "";

  sendFeedBack() {
    if (this.feedbackText.trim()) {
      this.feedbackEvent.emit(`${this.product.name}:${this.feedbackText}`);
      this.feedbackText = "";
    }
  }
}

```

product-details.html:

```

-----

<h3>Product Details</h3>
<p><strong>Name:</strong> {{ product.name }}</p>
<p><strong>Price:</strong> {{ product.price }}</p>

<textarea [(ngModel)]="feedbackText" placeholder="Enter your feedback"></textarea>
<br>
<button (click)="sendFeedBack()">Send Feedback</button>

```