Angular Content Projection with <ng-content>

Concept

Content projection in Angular allows developers to pass content from a parent component to a child component using the <ng-content> directive. It enables flexible and reusable components by allowing dynamic content injection.

Use Cases

- Reusable UI components (e.g., cards, modals, tabs)
- Dynamic content injection
- Customizable component structures

Implementation

1. Basic Single Slot Content Projection

Example where the child component allows dynamic content injection.

```
<app-card>
  <p>This is projected content inside the card!</p>
</app-card>
```

Child Component:

```
<div class='card'>
  <ng-content></ng-content>
</div>
```

2. Multi-Slot Content Projection

```
<app-panel>
  <div header>Panel Header</div>
  <div content>Main Panel Content</div>
</app-panel>
```

Child Component:

```
<div class='panel'>
  <div class='header'>
    <ng-content select='[header]'></ng-content>
  </div>
  <div class='content'>
    <ng-content select='[content]'></ng-content>
  </div>
</div>
```

Pros and Cons

Pros	Cons
Enhances component reusability	Cannot manipulate projected content directly
Supports flexible UI structures	Limited control over injected content

Best Practices

- Use named slots (`select` attribute) for better control
- Provide default content when necessary
- Avoid excessive nesting
- Utilize `@ContentChild` for content manipulation

Common Errors & Fixes

Error	Solution
Content not appearing	Ensure `<ng-content>` is present
`select` not working	Use correct attribute selectors