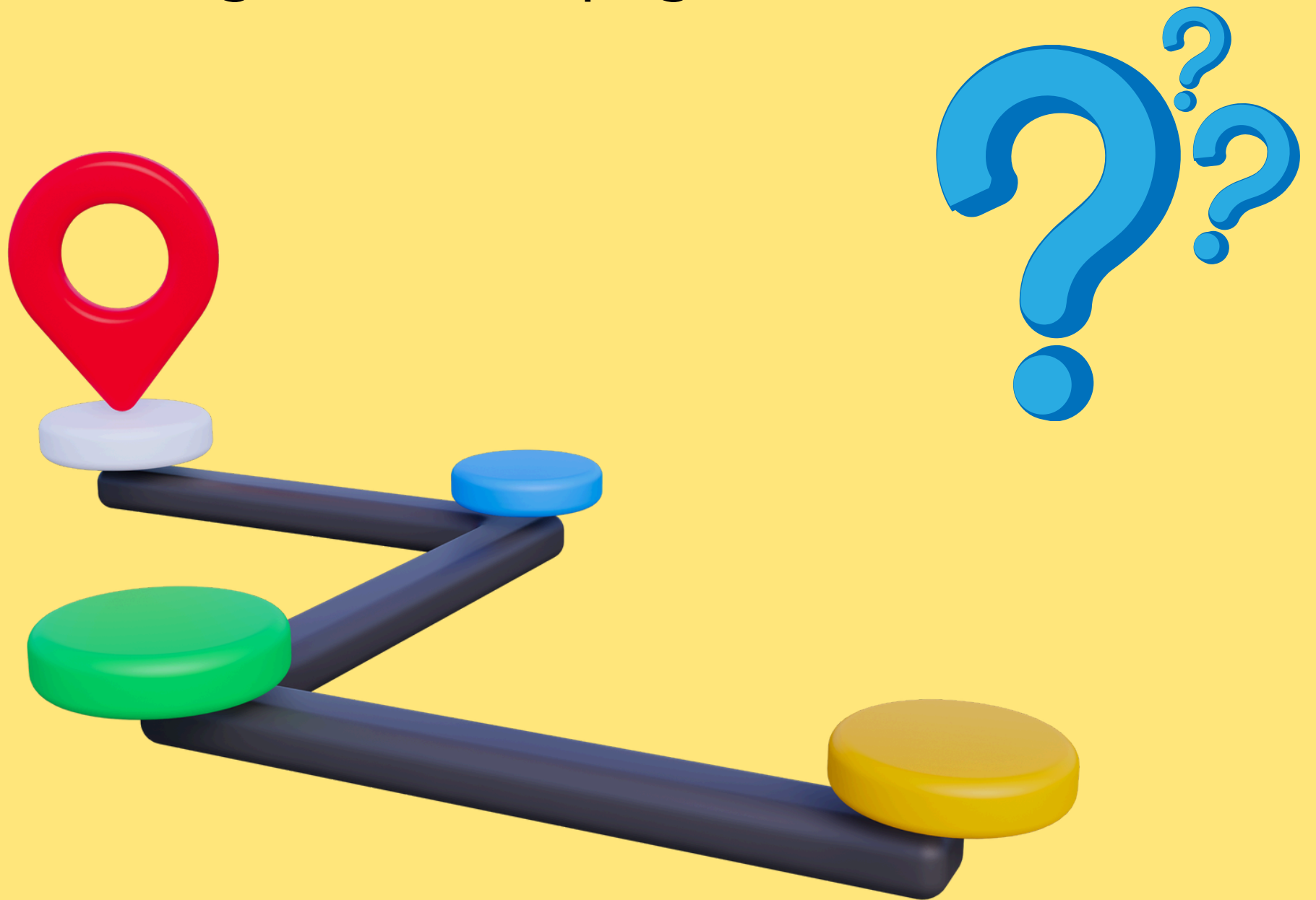# ANGULAR ROUTING

WHY ROUTING?

- Routing allows you to navigate between different views or components within your Angular application

- It helps create a seamless user experience by dynamically loading content without refreshing the entire page
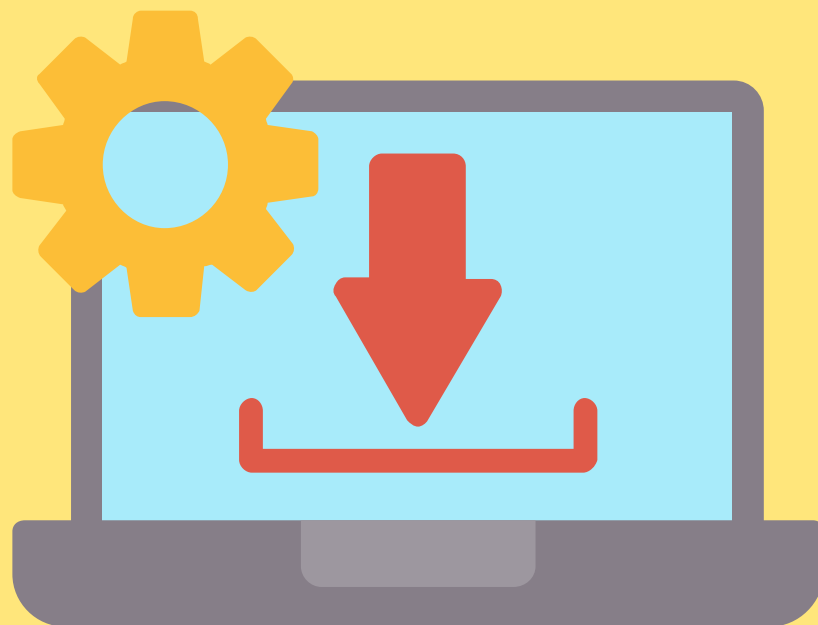
SETTING UP BASIC ROUTING IN ANGULAR

STEP 1: INSTALL ANGULAR CLI AND CREATE A NEW PROJECT

- If you haven't already, install Angular CLI and create a new Angular project

```bash
npm install -g @angular/cli
ng new my-angular-app
cd my-angular-app
```
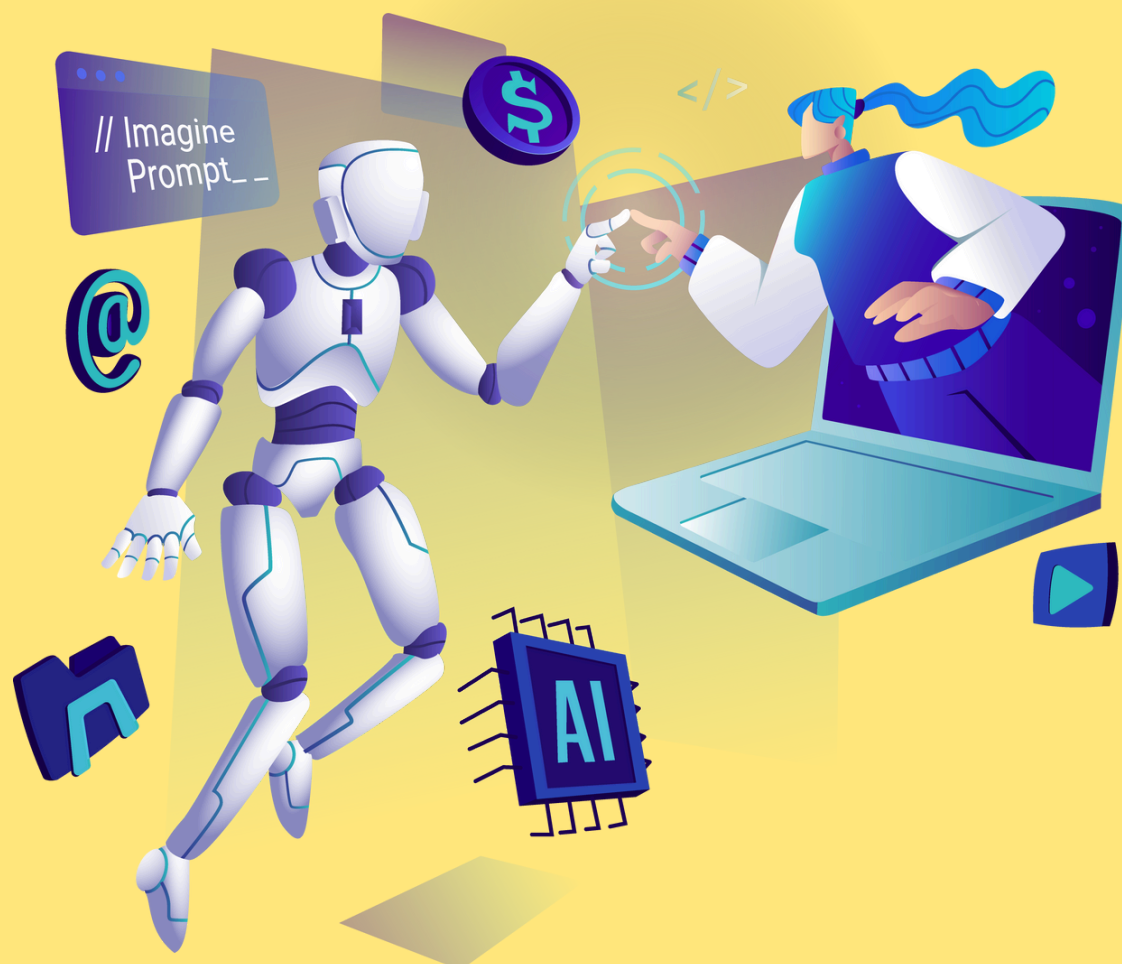
STEP 2: GENERATE COMPONENTS

- Generate a few components to demonstrate routing

- For example, create HomeComponent and AboutComponent:

```
BASH

ng generate component home
ng generate component about
```

STEP 3: DEFINE ROUTES

- Open app-routing.module.ts and define the routes for your application

```typescript
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from './home/home.component';
import { AboutComponent } from './about/about.component';

const routes: Routes = [
  { path: '', redirectTo: '/home', pathMatch: 'full' },
  { path: 'home', component: HomeComponent },
  { path: 'about', component: AboutComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```
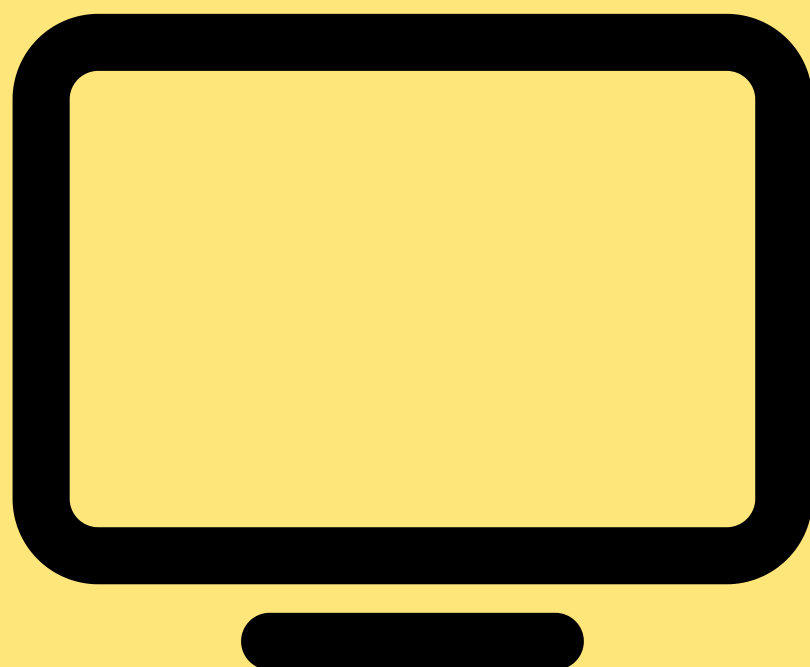
STEP 4: ADD ROUTER OUTLET

- In app.component.html, add the <router-outlet> directive

- This is where the routed components will be displayed

```html
<nav>
  <a routerLink="/home">Home</a>
  <a routerLink="/about">About</a>
</nav>
<router-outlet></router-outlet>
```

STEP 5: UPDATE COMPONENT TEMPLATES

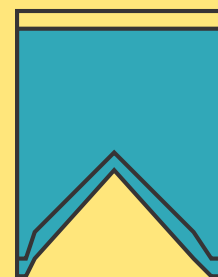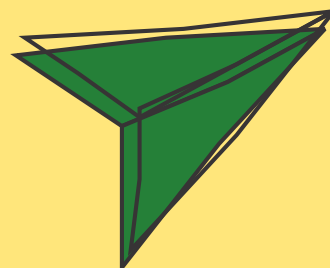- Update the templates for HomeComponent and AboutComponent

home.component.html

```html
<h1>Welcome to the Home Page!</h1>
<p>This is the home page of our Angular app.</p>
```

about.component.html

```html
<h1>About Us</h1>
<p>This is the about page of our Angular app.</p>
```

UPDATE

DID YOU FIND IT HELPFUL ?

Share this with a friend who needs it!

VINCENT RAJA

FULL STACK WEB DEVELOPER