

# INTERNET TEHNOLOGIJE 2

*Master strukovne studije*

## Projektni zadatak

Potrebno je da se napravi front-end aplikacija po izboru studenta koja će da sadrži sledeće elemente i koncepte obrađene u okviru predmeta:

### Obavezni elementi

#### Komponente

- Više funkcionalnih komponenti organizovanih u logičnu strukturu aplikacije
- Pravilna upotreba selektora, template-a i stilova
- Razdvajanje komponenti po odgovornosti (smart i presentational komponente)
- Korišćenje životnog ciklusa komponente (lifecycle hooks) gde je potrebno
- Koristiti novi Angular control flow template stil (`@if`, `@for`, `@switch`)

#### Komunikacija između komponenti (@Input i @Output)

- Prosleđivanje podataka iz roditeljske u dete komponentu korišćenjem @Input dekoratora
- Emitovanje događaja iz dete komponente ka roditeljskoj korišćenjem @Output i EventEmitter
- Demonstracija dvosmerne komunikacije kroz najmanje jedan praktični primer

#### Servisi i Dependency Injection (DI)

- Implementacija servisa koji enkapsulira poslovnu logiku
- Pravilno registrovanje servisa (providedIn: 'root' ili na nivou komponente / modula)
- Injektovanje servisa u komponente i druge servise putem inject() funkcije
- Razdvajanje odgovornosti između komponenti i servisa

#### HTTP klijent

- Komunikacija sa REST API-jem (može biti javni API ili lokalni mock server, npr. json-server)
- Implementacija GET, POST, PUT i DELETE zahteva
- Obrada odgovora i grešaka (error handling, catchError)
- Korišćenje Observable-a i osnovnih RxJS operatora (map, switchMap, tap, itd.)
- Opciono: implementacija HTTP interceptora (npr. za dodavanje tokena ili logovanje)

#### Rutiranje (Router)

- Konfiguracija ruta sa najmanje nekoliko različitih view-ova
- Navigacija pomoću routerLink direktive i Router servisa
- Korišćenje parametara rute (npr. /proizvod/:id) i query parametara

## **AuthGuard (zaštita ruta)**

- Implementacija barem jednog Guard-a (CanActivate) za zaštitu određenih ruta
- Preusmeravanje neautentifikovanih korisnika na login stranicu
- Demonstracija kroz scenario sa prijavom i odjavom korisnika

## **Forme**

- Implementacija formi korišćenjem Reactive Forms ili Template-driven pristupa
- Validacija polja (obavezna polja, minimalna/maksimalna dužina, pattern, email, itd.)
- Prikaz validacionih poruka korisniku
- Poželjno: korišćenje custom validatora

## **Pipe-ovi**

- Korišćenje ugrađenih pipe-ova (date, currency, uppercase, async, json, itd.)
- Implementacija najmanje jednog custom pipe-a
- Napredno: Lančano (chained) povezivanje pipe-ova u template-u

## **Angular Material**

- Integracija Angular Material biblioteke u projekat
- Korišćenje gotovih komponenti (MatButton, MatInput, MatTable, MatDialog, MatSnackBar, MatToolbar, itd.)
- Primena Material tema i konzistentan vizuelni identitet aplikacije

## **Opcioni elementi (donose dodatne poene)**

- Korišćenje Angular Signals
- Lazy loading i optimizacija performansi
- Custom direktive (atributske ili strukturne)

## **Tehnički zahtevi i predaja**

- Aplikacija mora biti razvijena u najnovijoj stabilnoj verziji Angular framework-a
- Kod treba da bude organizovan po modulima/folderima (komponente, servisi, modeli, guards, pipes, itd.)
- Poštovati osnovne principe čistog koda i konvencije imenovanja u TypeScript
- Projekat predati kao git repozitorijum (link ka repozitorijumu) ili kao .zip arhiva (bez node\_modules foldera)

## **Odbrana projekta**

Student je dužan da prilikom odbrane projekta demonstrira funkcionalnosti aplikacije i odgovori na pitanja vezana za implementaciju. Očekuje se razumevanje korišćenih koncepata, a ne samo njihova mehanička primena.