Univerzitet u Novom Sadu

Ekonomski fakultet u Subotici

**Angular u razvoju poslovnih aplikacija**

DIPLOMSKI RAD

Student:

Marija Matić PIN028/17

Subotica, 2021. godina

Univerzitet u Novom Sadu

Ekonomski fakultet u Subotici

**Angular u razvoju poslovnih aplikacija**

DIPLOMSKI RAD

Mentor: Student:

Sakal dr Marton Marija Matić PIN028/17

Subotica, 2021. godina

Uvod

Razvoj poslovnih aplikacija kakvih danas poznajemo i koristimo prvenstveno se obavljao pomoću HTML-a – jezika za označavanje hiperteksta. HTML je krajem 1991. godine razvio Tim Berners-Li i HTML kao takav sastojao se od samo 18 različitih tagova.

Struktura HTML fajlova definisana je različitim tipovima tagova, na osnovu kojih veb pretraživač formatira i prikazuje sadržaj.

Veb pretraživači konstruisani su tako da primaju HTML fajlove i njihov sadržaj – tekstove, slike i multimediju interpretiraju u vidu veb stranica koje su razumljive i sa kojima mogu da komuniciraju i „obični“ korisnici.

S nastankom potrebe da se razdvoji sadržaj samih veb strana od njenog vizuelnog prikaza, dolazi do nastanka CSS-a, koji se smatra prekretnicom u sferi veb dizajna. CSS omogućava uređivanje izgleda veb strana uz formatiranje izgleda pojedinačnih HTML elemenata – promenu njihovog izgleda, fontova, boja itd.

Separacija formatiranja i strukture doprinosi rastu fleksibilnosti, padu kompleksnosti, ali i prilagođavanju veb strana uređajima različitih dimenzija.

Prvobitne veb stranice bile su potpuno statičke, odnosno, njihov sadržaj nije mogao da se menja nakon prvobitnog učitavanja. Kao posledica želje da se ovo ograničenje otkloni, 1995. godine kompanija Netscape dodaje skriptni jezik u veb pretraživač Navigator i to u cilju ugrađivanja programskog jezika Scheme-a i programskog jezika Jave u tada najkorišćeniji veb pretraživač. Kada su uvideli probleme vezane za implementaciju, dolaze do ideje da kreiraju potpuno nov programski jezik. Iako je inicijalno nosio ime LiveScript, sa prvom objavom beta verzije Navigatora izlazi i novi programski jezik po imenu JavaScript. JavaScript i ostali skritpni jezici zapravo determinišu način na koji će se HTML primljene stranice biti preveden u objektni model dokumenta – DOM[[1]](#footnote-1). Nastankom Internet Eskplorera i njegovog skriptnog jezika JScript-a, dolazi do „rata“ između veb pretraživača. To otežava situaciju i stvara problem programerima kojima je bilo teško da kreiraju veb sajtove koji su mogli da rade na oba veb pretraživača. U godinama koje slede popularnost Microsoft-ovog Internet Explorer-a je rasla, te 2000. godine Navigator gubi trku, i skriptni jezik JSkript postaje standard za razvoj dinamičkih veb stranica.

JavaScript kakav danas poznajemo nastao je 2008. godine s nastankom Microsoft-ovog Google Chrome-a, i to u sklopu V8 – open-source pokretača (biblioteke) koji je služio kao tradicionalni interpreter. Ono što je pravilo razliku u odnosu na ostale veb pretraživače i činilo ga bržim u odnosu na njih, jeste postojanje JIT-a[[2]](#footnote-2) - kompajlera koji nadgleda i optimizuje način izvršavanja kompjuterskog koda.

Danas preko 80% veb sajtova koristi neku od biblioteka JavaScripta[[3]](#footnote-3) ili veb-okvir za podršku dinamičnih veb sajtova, veb aplikacija, veb servisa, razvoju veba uopšte.

Angular

Jedan od najpoznatijih framework-a danas, pored Vue.js-a i React-a jeste Angular. Prva verzija ovog framework-a pod nazivom AngularJS nastala je 2009. godine i održavana je primarno od strane kompanije Google, ali i zajednice pojedinaca i kompanija u cilju razvoja „jednostranične aplikacije[[4]](#footnote-4)“. Ideja je da korisnikovo iskustvo i interakcija sa veb-aplikacijom ili veb-sajtom bude slična interakciji sa aplikacijama na računarima bez internet konekcije. Umesto učitavanja kompletnog seta veb stranica sa veb servera, aplikacija ili veb sajt dinamično učitava ili dopunjava postojeću veb stranicu sa novim podacima sa servera. Na taj dolazi do uštede resursa, što čini samu aplikaciju mnogo efikasnijom.

AngularJs omogućava proširenje HTML vokabulara, dodaje mogućnost korišćenja različitih kontrola, omogućava kreiranje komponenti koje se mogu više puta koristiti, olakšava navigaciju, testiranje, komunikaciju sa backend-om itd. Pored toga, dizajniran je tako da ga je moguće korisiti sa ostalim bibliotekama, a sve predefinisane opcije se mogu prilagoditi pojedinačnim potrebama.

1. Document Object Model [↑](#footnote-ref-1)
2. Just-in-time [↑](#footnote-ref-2)
3. https://en.wikipedia.org/wiki/JavaScript [↑](#footnote-ref-3)
4. Single-page application [↑](#footnote-ref-4)