SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2829

RAZVOJ BIBLIOTEKE WEB KOMPONENTI KORISTEĆI METODOLOGIJU ATOMARNOG OBLIKOVANJA

Antonio Kamber

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2829

RAZVOJ BIBLIOTEKE WEB KOMPONENTI KORISTEĆI METODOLOGIJU ATOMARNOG OBLIKOVANJA

Antonio Kamber

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 11. ožujka 2022.

DIPLOMSKI ZADATAK br. 2829

Pristupnik:	**Antonio Kamber (0036498539)**
Studij:	Računarstvo
Profil:	Programsko inženjerstvo i informacijski sustavi
Mentor:	izv. prof. dr. sc. Alan Jović

Zadatak: **Razvoj biblioteke web komponenti koristeći metodologiju atomarnog oblikovanja**

Opis zadatka:

Web komponente su skup aplikacijskih programskih sučelja koji omogućuje izradu novih, prilagođenih HTML oznaka koje se mogu koristiti na web stranicama i u web aplikacijama. Metodologija atomarnog oblikovanja web komponenti nalaže da se grafičko korisničko sučelje treba oblikovati po uzoru na kemijske elemente. Najmanje jedinice sučelja nazivaju se atomi. Grupiranjem atoma dobivaju se molekule koje sakrivaju komunikaciju i suradnju svojih atoma od korisnika te nude proširene funkcionalnosti. Molekule u suradnji se nazivaju organizmi koji konačno svojim grupiranjem čine predloške (engl. templates). Kako se web komponente prevode u čisti HTML, one su neovisne o tehnologiji programiranja te se mogu koristiti prilikom razvoja u bilo kojoj tehnologiji za razvoj web aplikacija. U okviru ovog diplomskog rada potrebno je razviti biblioteku web komponenti organiziranu i ostvarenu koristeći metodologiju atomarnog oblikovanja. Pritom je potrebno ostvariti projekt u kojemu će se razvijati komponente te projekt koji će služiti za jednostavni pregled razvijenih komponenti. Nakon ostvarenja projekta, potrebno je objaviti biblioteku komponenti na registracijskoj platformi NPM. Konačno, potrebno je izraditi i objaviti statičku web stranicu kako bi se prikazala praktična primjena razvijene biblioteke web komponenti.

Rok za predaju rada: 27. lipnja 2022.

Zahvaljujem mentoru, izv. prof. dr. sc. Alanu Joviću,

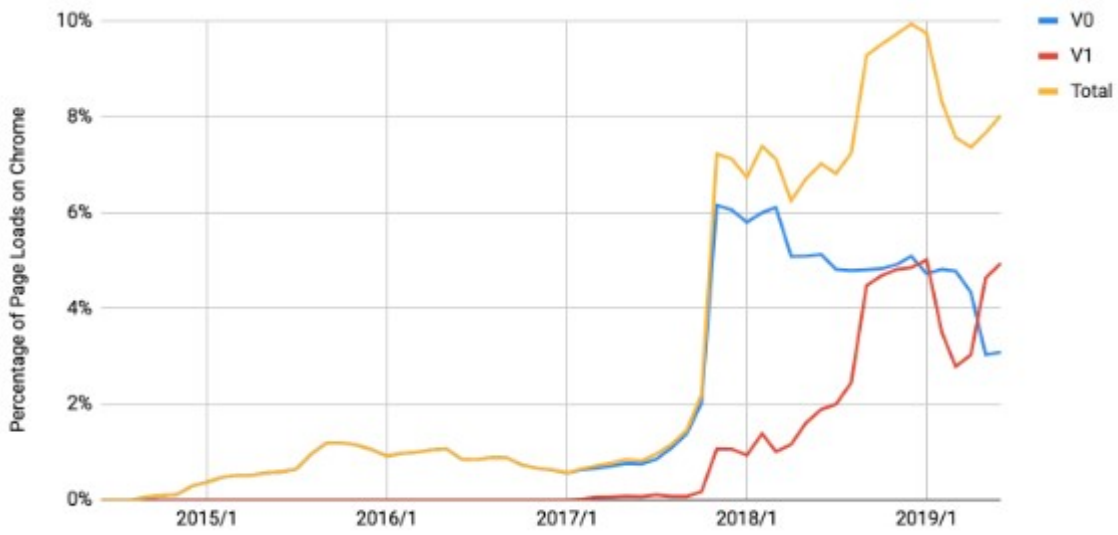
na pomoći, savjetima, strpljenju i razumijevanju.

Sadržaj

Uvod

Web komponente su definirane skupom web programskih biblioteka koje nam omogućuju razvoj prilagođenih HTML-elemenata. Sastoje se od tri zasebne tehnologije: prilagođeni elementi, sjena DOM-a (engl. *Document Object Model*) i HTML-predlošci. Glavni doprinos prilagođenih elemenata je ponovna uporaba i izdvajanje funkcionalnosti u jedinstvenu cjelinu. Svaka komponenta ima definirano grafičko sučelje i kod koji predstavlja implementaciju funkcionalnosti. Svi implementacijski detalji su sakriveni od javnosti, a funkcionalnost je dostupna definiranim sučeljem. Uporabom komponenti se smanjuje opseg funkcionalnosti koje programeri trebaju implementirati u razvijani sustav. Garantira se pouzdani rad bez pogrešaka zbog čega se smanjuje količina testiranja i vrijeme implementacije sustava. Web komponente koje su implementirane prema jasno definiranim valjanim normama garantiraju ispravnost izvođenja u svim modernim preglednicima weba. Od kako su razvijeni mehanizmi kojima se web aplikacije mogu izvoditi na mobilnim uređajima i instalirati na radnu površinu računala, većina programskih rješenja je prilagođena za web. Glavna motivacija za diplomski rad je ostvariti biblioteku web komponenti koje implementiraju općenite i učestalo korištene funkcionalnosti. Biblioteka bi se mogla koristiti neovisno o tehnologiji implementacije sustava. Programer koji je upoznat s bibliotekom može svoje poznavanje prenijeti na bilo koji drugi sustav koji razvija, neovisno o ciljanoj platformi primjene sustava. Poznavanje API-ja biblioteke smanjuje se vrijeme implementacije i testiranja te omogućuje ponovnu uporabu koda.

Zahvaljujući orijentaciji prema web aplikacijama uporaba web komponenti kao mehanizma bilježi značajan porast.



Sl. 1.1 Uporaba web komponenti

Na slici 1.1 je prikazana uporaba web komponenti verzije v0 i v1 u razdoblju 2015. – 2019.. Podatke je kompanija Google javno objavila u travnju 2019. godine s ciljem poticaja uporabe web komponenti. Graf prikazuje postotak web stranica koje koriste web komponente, učitanih unutar web preglednika Google Chrome diljem svijeta. Iako nam nisu dostupni statistički podaci Googla nakon 2019. Godine napretkom web tehnologija se može zaključiti kako je i broj stranica koje koriste web komponente također porastao. Web komponente je prvi predstavio Alex Russell na konferenciji Fronteers 2011. godine. Prvi razvojni okvir Polymer, koji je omogućio implementaciju komponenti razvio je Googleov tim 2011. Iste godine je Firefox dodao 63 razvojna alata kako bi podržao daljnji razvoj. Web komponente su podržane u preglednicima Google Chrome, Mozilla Firefox, Microsoft Edge, Safari i Opera.

Postoji mnoštvo timova koji ulažu trud u razvoj web komponenti. Tim WebComponents.org [1], zadužen za objavljivanje svih promjena i novosti vezanih za tehnologiju web komponenta javnosti, razvio je rješenje koje nam omogućuje pregled postojećih komponenti. Tim Custom Elements Everywhere [2] nudi rješenje za validaciju programske podrške koja implementira web norme za izradu web komponenti u modernim razvojnim okvirima, te prikaz popisa problema i mogućih rješenja.

Kako se broj tehnologija za razvoj web aplikacija sve više povećava smatram da je nužno što prije poticati programere da se okrenu tehnologijama koje su održive i kompatabilne s ostalim

učestalim tehnologijama. Pravi način za ostvariti navedeno je razvoj funkcionalne, smislene i pouzdane biblioteke web komponenti što i je motivacija te cilj ovog diplomskog rada.

1. Tehnologije i načela diplomskog rada

1.1. Tehnologije web komponenti

Kako je spomenuto u prethodnom poglavlju web komponente su zasnovane na tri zasebne tehnologije: prilagođeni elementi, sjena DOM-a i HTML-predlošci.

Prilagođeni elementi su valjani HTML-elementi s vlastitim prilagođenim predlošcima, ponašanjima, funkcionalnostima, sučeljem i oznakama. S gledišta programera koji koristi biblioteku razvijenih komponenti svaki prilagođeni element je jednostavna HTML-oznaka koja je ostvarena skupom funkcionalnosti definiranih JavaScriptovim API-jem. Prilagođeni elementi su definirani u potpunosti HTML-ovom živućom standardnom specifikacijom [3]. Cilj je proširenje funkcionalnosti osnovnih HTML-elemenata i registracija nove oznake kako bi se ona mogla koristiti unutar preglednika. Nije potrebno instalirati dodatne pakete niti podesiti bilo kakve postavke, jer je tehnologija podržana u svim modernim web preglednicima.

Sjena objektnog modela dokumenta omogućuje izolaciju CSS-ovog i JavaScriptovog koda. Tehnologija sprječava konflikte u imenima pravila CSS-a i onemogućava pristup elementima predloška dokumenta. Primjerice, standardni poziv JavaScriptove funkcije `document.querySelector(selector)` neće dohvatiti elemente predloška komponente. Kako bi se oni dohvatili potrebno je nad referencom prilagođenog elementa pozvati funkciju shadowRoot.querySelector(selector). Tehnologija sjene DOM-a omogućuje enkapsulaciju koda.

HTML-predlošci nam omogućuju da umetnemo HTML-ov kod koji neće biti prikazan dokle god se kod ne referencira. Predložak je tijelo HTML-oznake *template*. Oznaku je moguće referencirati proizvoljni broj puta čime će svaki put njezin sadržaj biti prikazan.

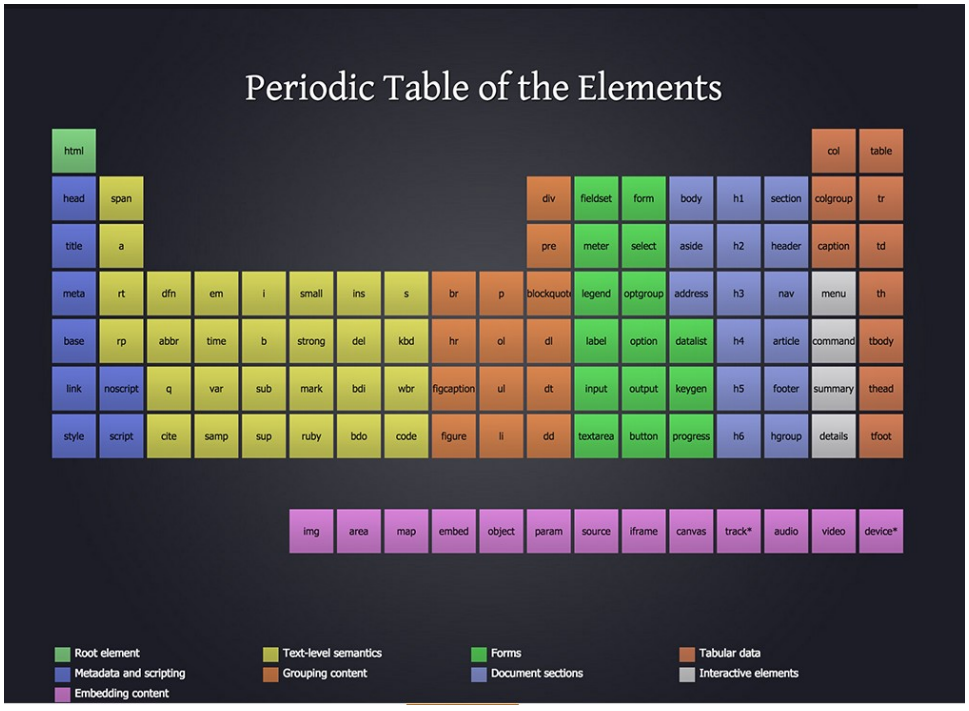
Prilikom implementacija web komponenti programeri najčešće nisu svjesni navedenih tehnologija jer se koriste razvojni okviri čija implementacija ostvaruje sve opisano bez potrebe dodatnog pisanja koda. U radu se koristi tehnologija Angular Elements opisana u sljedećem

poglavlju, a čiji API omogućuje pisanje Angularovih komponenti koje se prilikom prevođenja uporabom API-ja prevode u web komponente dostupne u web pregledniku kao prilagođeni element pod definiranom oznakom.

1.2. Atomarno oblikovanje

Atomarno oblikovanje (engl. *atomic design*) se zasniva na činjenici da sva materija u svemiru može biti razložena u skup atoma [5]. Korisničko sučelje se također može razložiti na najmanje jedinice zvane atomi.

Cilj razlaganja elemenata sučelja je slično objektnom modeliranju u objektno orijentiranim jezicima. Želimo stvoriti apstrakciju koju ćemo potom pojednostavniti razlažući ju na više apstrakcija sve manjih opsega. U kontekstu korisničkog sučelja možemo reći kako je krajnji rezultat skup konačnih manjih elemenata koji se mogu integrirati u veće elemente, međusobno komunicirati i surađivati ostvarujući složenu funkcionalnost skrivajući komunikaciju i nužne prilagodbe pojedinih elemenata. Poznati programer i inženjer Josh Duck smatra kako se sve web stranice i aplikacije mogu sastaviti od istih HTML-elemenata koje je posložio u preglednu tablicu po uzoru na kemijski periodni sustav[5].



Sl 1.1. Periodni sustav HTML elemenata [5]

S obzirom da HTML čini konačni skup elemenata možemo primijeniti isti princip koji uočavamo u prirodnom svijetu kako bismo modelirali i izgradili korisničko sučelje.

Prema atomarnom oblikovanju postoji pet jedinstvenih razina koje surađuju pri izgradnji hijerarhijskog i proširivog sučelja. Pet razina su redom: atomi, molekule, organizmi, predlošci i stranice. Važno je naglasiti da načelo ne predstavlja linearan proces nego konceptualno modeliranje sučelja istovremeno kao jedinstvene cjeline i kolekcije manjih dijelova.

U prirodnom svijetu, atomi su građevne jedinice koje ne mogu biti rastavljeni na manje jedinice. U korisničkom sučelju atomi su temeljni elementi koji svojom funkcionalnošću ne mogu biti rastavljeni na manje jedinice. Svaki atom ima svoja jedinstvena svojstva koja utječu na to kako će se atomi ponašati i izgledati u korisničkom sučelju. Prilikom oblikovanja prvo se razvijaju atomi jer oni čine bazu te se prilikom promjene njihovog sučelja ili funkcionalnosti mijenja ostatak biblioteke. Biblioteka TDT razvijena u ovom diplomskom radu sadrži mnoštvo atoma, primjerice tdt-button, tdt-input, tdt-columns, tdt-scroll-to-top itd.

Molekule u prirodnom svijetu čine grupe atoma povezanih jedinstvenim svojstvima. U korisničkom sučelju molekule su jednostavne grupe atoma koje čine funkcionalnu i smislenu cjelinu. Svaki atom u molekuli ima svoju ulogu, no molekula kao zajednica atoma obavlja funkcionalnost koju pojedini atom sam ne može obaviti. Cilj molekule je da bude primjenjiva i da obavlja samo jednu složenu funkcionalnost. Prilikom razvoja primjenjujemo principe i načela programskog inženjerstva. Princip samo jedne odgovornosti (engl. *Single responsibility princip*le) i načelo ponovne uporabe (engl. *Reusability*). Molekule olakšavaju testiranje korisničkog sučelja i potiču ponovnu uporabu i osiguravaju konzistentnost korisničkog sučelja. Biblioteka TDT sadrži velik broj molekula, primjerice tdt-dropdown, tdt-teaser-list, tdt-sidebar, tdt-text-with-image itd.

U korisničkom sučelju složene cjeline sastavljene od atoma, molekula ili drugih složenih cjelina se prema atomarnom oblikovanju nazivaju organizmi. Organizam čini zasebni odjeljak i smislenu cjelinu korisničkog sučelja iako se sastoji od više manjih dijelova. Ono što mu omogućava da bude zaseban je jedinstveno grafičko sučelje i najčešće više od jedne

funkcionalnosti koju niti jedna od njegovih građevnih jedinica ne bi mogla obaviti samostalno. Sučelje organizma treba biti vrlo općenito i mora omogućiti uporabu u raznim poslovnim domenama web aplikacija i stranica. Biblioteka TDT sadrži organizme tdt-detail-panel i tdt-navbar.

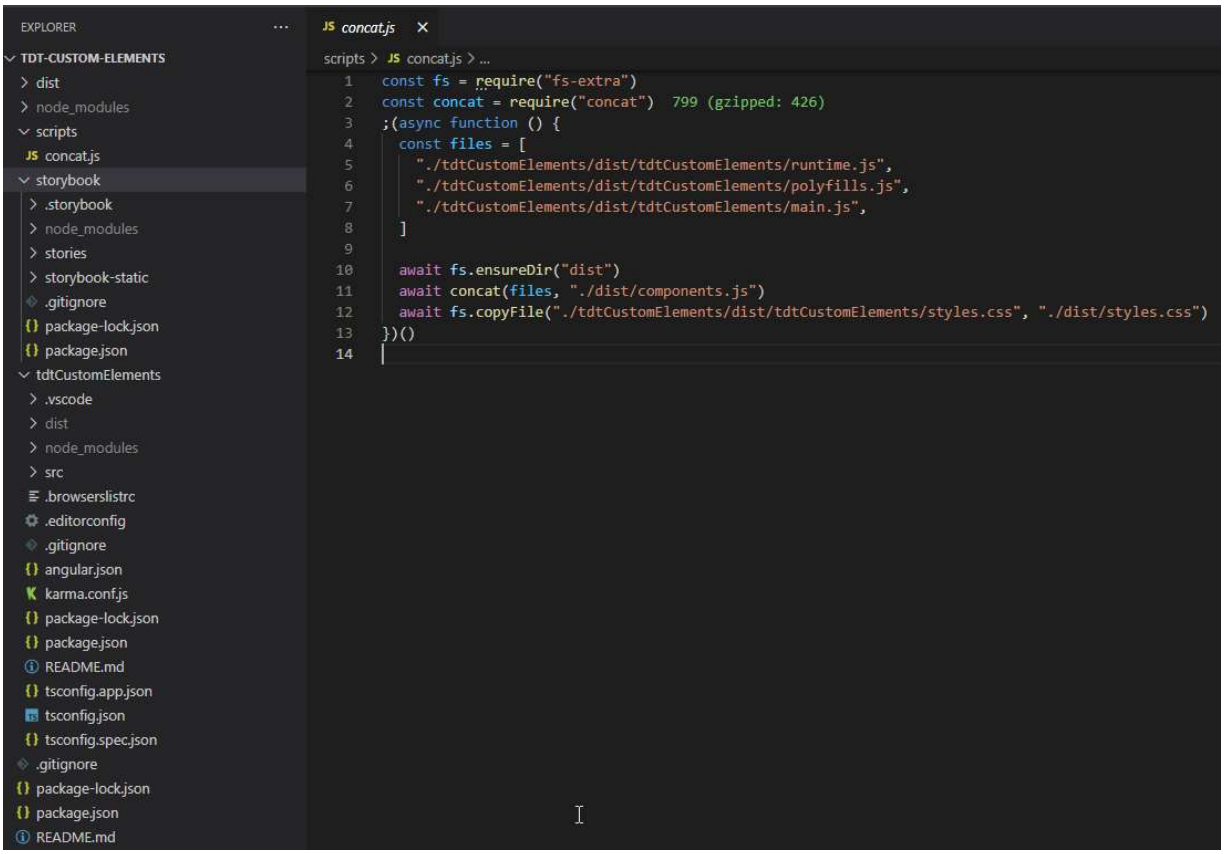
Predlošci su koncept atomarnog oblikovanja najmanje apstrakcije s ciljem da ih tehnički neupućene osobe, poput vlasnika proizvoda, korisnika aplikacije ili projektnih menadžera, mogu jednostavno shvatiti i uočiti u krajnjoj aplikaciji. Predložak je element na razini web stranice koji određuje poredak i raspored organizama i molekula u grafičkom sučelju. Predložak povezuje jednostavnije razine čime ih približava krajnjem korisniku dajući im semantiku. Važna karakteristika predložaka je da predstavljaju elemente stranice, a ne konačne stranice ili konačnu web aplikaciju. Predlošci trebaju osigurati dobro iskustvo korisnika bez obzira na domenu i konkretne podatke koje sučelje prikazuje. Stavlja se fokus na pitanje od čega je građen sadržaj web stranice, a ne što je sadržaj stranice. Konačna implementacija zapravo predstavlja kostur web stranice. Biblioteka TDT sadrži predloške tdt-layout, tdt-overview i tdt-static-template.

Posljednje u hijerarhiji razvoja prema atomarnog oblikovanju su web stranice. One su skup predložaka kojima je predan konkretan kontekst i krajnji podaci koji će biti vidljivi korisniku u sučelju. Sučelje treba biti intuitivno, jasno i razumljivo bez potrebnih pojašnjenja.

2. Arhitektura projekta

2.1. Organizacija projekata

Diplomski rad se sastoji od dva neovisna projekta za razvoj i skripti koje služe za povezivanje razvojnih projekata. Organizacija projekata po direktorijima je prikazana na slici 2.1.



Sl. 2.1. Organizacija projekata

Unutar direktorija tdtCustomElements nalazi se projekt za razvoj Angularovih komponenti koje se prevode u web komponente. Tehnologija i uporaba prevođenja su opisane u sljedećem poglavlju. Jednom kada se implementiraju Angularove komponente i prevede ih se u web komponentu, potrebno je omogućiti pregled i interaktivno testiranje implementacije. Projekt

za testiranje i pregled nalazi se unutar direktorija storybook. Projekt je ostvaren istoimenim alatom opisanim u poglavlju 2.3.

Unutar direktorija scripts nalazi se skripta concat.js prikazana na slici 2.1. Skripta spaja sve JavaScriptove izvršne datoteke projekta tdtCustomElements u jednu datoteku naziva components.js te ju smješta u direktorij dist. Skripta dodatno kopira izvršnu CSS-datoteku projekta tdtCustomElements u direktorij dist.

Svaka korisnička priča projekta storybook referencira datoteke smještene u direktoriju dist u kojemu se, zahvaljujući skripti concat.js, nalazi sav potreban kod implementacije razvijenih web komponenti.

2.2. Projekt tdtCustomElements – Angularova komponenta Element

Angularova komponenta Element je uobičajena Angularova komponenta koja se uporabom funkcije razvojnog okvira može prevesti u prilagođeni element (engl. *Custom elements*). Paket @angular/elements sadrži funkciju `createCustomElement()` koja predstavlja most između sučelja Angularove komponente i funkcionalnosti detekcije promjene u ugrađenom DOM-ovom API-ju [4]. Prvi argument funkcije je referenca na Angularovu komponentu. Drugi argument je konfiguracijski objekt koji nudi podešavanje postavki inicijalizacije razreda. U razvojnom projektu tdtCustomElements, unutar glavnoga modula AppModule, prilikom pokretanja projekta poziva se metoda životnog ciklusa `ngDoBoostrap`, prije bilo koje druge metode životnog ciklusa. Za svaku komponentu projekta pozivamo funkciju `createCustomElement()` i spremamo referencu na vraćeni objekt. Koristimo standardno sučelje Web API-ja `CustomElementRegistry` i pozivamo metodu `define()`. Prvi argument metode je naziv HTML-oznake naše web komponente. Drugi argument je objekt koji je vratila funkcija `createCustomElement()`. Kada se izvedena komponenta dodaje u DOM, preglednik kreira instancu registriranog razreda. Sadržaj koji je implementiran sintaksom Angularovog predloška se transformira u HTML te se podaci umeću u DOM koristeći varijable razreda. Ulazna svojstva razreda se preslikavaju u atribute HTML-elementa.

Ako je primjerice varijabla razreda `MojRazred` nazvana `@Input() mojaVarijabla` s postavljenom numeričkom vrijednosti 5, preslikavanje će generirati HTML-oznaku `<moj-razred />`, te postaviti atribut `<moj-razred moja-varijabla="5" />`. Za okidanje događaja Angular koristi metodu `emit()` razreda EventEmitter. Izlazni događaji razreda `MojRazred` će izazvati okidanje HTML-ovog prilagođenog događaja s istim nazivom. Primjerice, izlazna varijabla razreda `@Output() mojDogađaj = new EventEmitter()` će okinuti prilagođeni događaj s nazivom `mojDogađaj`. Vrijednosti predane metodi `emit()` će biti definirane kao svojstvo `detail` okinutog prilagođenog događaja. Zauzeta memorija se automatski oslobađa u trenutku kada je komponenta uklonjena iz DOM-a.

2.3. Projekt storybook – alat storybook

Storybook je JavaScriptov alat koji omogućava razvojnim inženjerima organiziranje sustava komponenti grafičkog korisničkog sučelja olakšavajući razvojni proces, testiranje i dokumentaciju [6]. Za svaku implementiranu komponentu alat omogućuje stvaranje takozvanih datoteka priče (od engl. *storybook*) kojima se prikazuju različiti slučajevi uporabe komponente. Svaka priča može prikazivati drugačiju verziju komponente. Priča je uokvirena u prozor iFrame koji predstavlja uređaj na kojemu se komponenta prikazuje unutar web preglednika. Moguće je prilagoditi veličinu uređaja, dinamički promijeniti svojstva koja se predaju komponenti putem sučelja alata i pregledati prilagođene događaje komponente. Alat omogućuje organiziranu, preglednu i efikasnu razvojnu okolinu za implementaciju, testiranje, održavanje i dokumentiranje komponenti.

3. Biblioteka web komponenti TDT

TDT je biblioteka koja sadrži mnoštvo web komponenti vrlo velike mogućnosti primjenjivosti s jasno definiranim sučeljima. Svaka komponenta biblioteke ima jedinstveno oblikovano grafičko sučelje te API kojim se definiraju svojstva i događaji koje komponenta okida. Svojstvima se mogu definirati vrijednosti te ih se u bilo kojemu trenutku može čitati kako bi se prikupile potrebne informacije. Ukoliko se programer odluči na uporabu komponenti

biblioteke prilikom razvoja aplikacija garantira se ugodno korisničko iskustvo, konzistentno oblikovanje sučelja te prilagodba raznim uređajima i širinama ekrana. Komponente se jednako ponašaju u bilo kojemu modernom web pregledniku. Može ih se koristiti neovisno o programskom jeziku i razvojnoj okolini. Biblioteka je optimirana i veličina združene (engl. *bundle*) datoteke koja sadrži sav potreban kod je svega 501 kB. S obzirom na tako malu količinu memorije koju je potrebno rezervirati za biblioteku ona je primjenjiva u svim web aplikacijama, jer ne stvara primjetno opterećenje na vrijeme učitavanja i izvršavanja.

Biblioteka sadrži složene komponente čijom se integracijom jednostavno mogu izgraditi složene web stranice. Ukoliko se radi o jednostavnijim stranicama, komponente pružaju implementaciju većine potrebnih funkcionalnosti. Vrijeme implementacije se značajno smanjuje uporabom biblioteke. Kako bi se demonstrirala jednostavnost primjene komponenti, u sklopu diplomskog rada napravljena je jednostavna web stranica knjigovodstvenog servisa.

3.1. Instalacija

Naziv biblioteke je tdt-web-components te se ona može preuzeti putem registracijske platforme NPM. Na lokalnom računalu je potrebno imati datoteku koja sadrži JavaScriptov kod i CSS-ovu datoteku s definiranim stilovima, bojama i animacijama. Preuzimanjem će se automatski preuzeti četiri datoteke; uz već navedene, preuzimaju se i datoteke README.md koja sadrži kratki opis biblioteke te konfiguracijska datoteka package.json. Naredba za preuzimanje paketa putem konzole je

```
> npm install @tdt/tdt-web-components
```

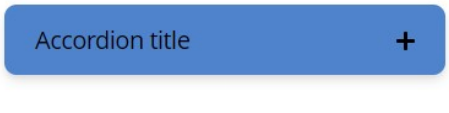
gdje `@tdt` predstavlja korisničko ime mojega profila na platformi, a tdt-web-components je naziv biblioteke. Druga opcija preuzimanja biblioteke je putem mreže za dostavu sadržaja UNPKG. Datoteka se nalazi na adresi *https://unpkg.com/@akbus/tdt-web-components/dist/components.js* . Uobičajeno je preuzimanje uporabom HTML-ove oznake script sintaksom `<script type="module" src="link"></script>`. Sljedeći korak je preuzimanje datoteke sa stilovima preko adrese *https://unpkg.com/@akbus/tdt-web-components/dist/styles.css*". Uobičajeno je koristiti HTML-ovu oznaku link sintaksom `<link`

`rel="stylesheet" href="link" />`. Nakon preuzimanja svega dvije datoteke sve komponente biblioteke se registriraju unutar preglednika kao prilagođeni elementi s vlastitim oznakama. Spremne su i dostupne programeru za uporabu.

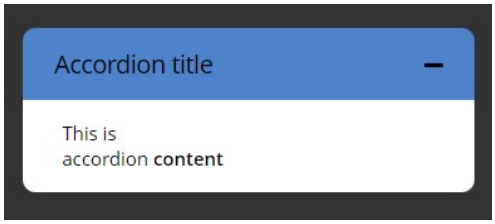
3.2. Atomi

3.2.1.TdtAccordion

TdtAccordion je komponenta čije se sučelje sastoji od naslova i sadržaja. Komponenta može biti u dva stanja: otvorena i zatvorena. Na slici 3.1 je prikazano zatvoreno stanje komponente. Prikazuje se naslov i pored toga ikonu koja ukazuje korisniku da postoji dodatan sadržaj koji je u trenutnom stanju sakriven. Nakon klika na područje naslova ili ikone komponenta prelazi u otvoreno stanje, prikazano na slici 3.2, u kojemu se uz naslov prikazuje i određeni sadržaj. Ikona pored naslova se u otvorenome stanju mijenja i ukazuje korisniku da je moguće sakriti sadržaj komponente.



Sl. 3.1 Zatvoreno stanje



Sl. 3.2 Otvoreno stanje

Prilikom promjene stanja komponenta okida prilagođeni događaj `onOpenChange` u čijem se svojstvu `detail` nalazi istinita vrijednost ukoliko je komponenta upravo prešla u stanje otvoreno, a neistinita ukoliko se dogodio prijelaz u stanje zatvoreno. Sadržaj naslova je definiran svojstvom `title`. Upravljanje i čitanje stanja komponente je ostvareno svojstvom `opened`, tipa boolean. Programer ga može koristiti kako bi izazvao promjenu stanja. Postavljanjem svojstva na vrijednost *true* komponenta prelazi u stanje otvoreno, a postavljanjem svojstva na vrijednost *false* komponenta prelazi u stanje zatvoreno. Na slici 3.3 možemo vidjeti kako osnovno sučelje komponente izgleda po z-osi uzdignuto u odnosu na pozadinu što se postiglo uporabom CSS-svojstva sjene stvarajući dojam kako se pozadina nalazi iza elementa tdtAccordion. Komponenta implementira sučelje IFlat zbog čega sadrži

svojstvo `isFlat`. Definiranjem neistinite vrijednosti se ostvaruje poravnanje komponente po z-osi s pozadinom jer se ne koristi CSS-svojstvo sijena.

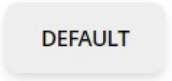
Sl. 3.3 Poravnanje s pozadinom po z-osi

3.2.2. TdtButton

Komponenta TdtButton proširuje funkcionalnost HTML-elementa button stilizirajući njegovo grafičko sučelje. Komponenta je implementirana u tri varijante: primarna, sekundarna i osnovna. Svaka varijanta odabire prikladne boje iz teme biblioteke ovisno o ciljanoj želji isticanja grafičkog sučelja elementa unutar prozora web preglednika. Varijanta je definirana svojstvom `variant`. Primarna varijanta, prikazana na slici 3.4, odabire boje s ciljem isticanja grafičkog sučelja kako bi ljudskom oku sučelje bilo prvo uočljivo. Sekundarna varijanta, prikazana na slici 3.5, odabire boju s ciljem da korisnik prvo ugleda primarne elemente, a potom sve sekundarne. Osnovna varijanta, prikazana na slici 3.6, prikazuje sučelje koje nije posebno istaknuto te ne odvlači pažnju ljudskom oku.

Sl. 3.4 Primarna varijanta

Sl. 3.5 Sekundarna varijanta



Sl. 3.6 Osnovna varijanta

Kada je svojstvo `disabled` istinito korisniku se onemogućuje klik na gumb te se mijenja izgled komponente kako bi to korisniku bilo razumljivo, slika 3.7.

Sl. 3.7 Onemogućeni gumb

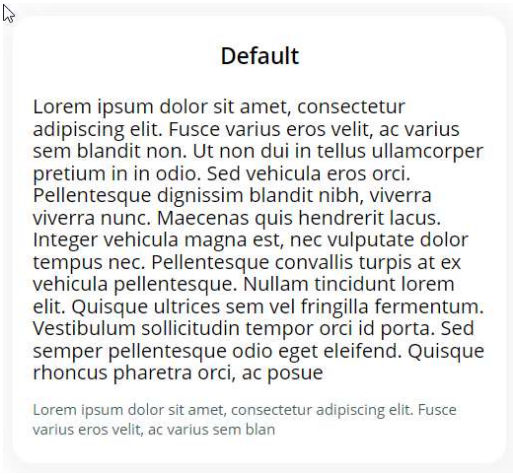
Komponenta implementira sučelje komponente tdtLabel opisano u poglavlju 3.2.8., zbog čega pokriva sve funkcionalnosti opisane u tom poglavlju. Tekst i ikona se u sučelju pozicioniraju unutar gumba, slika 3.8.

Sl. 3.8 Gumb s ikonom

Vrijednost svojstva `type` se propagira HTML-elementu button. Klikom na sučelje komponente okida se prilagođeni događaj `onClick` koji programer može koristiti kako bi reagirao na korisnički klik.

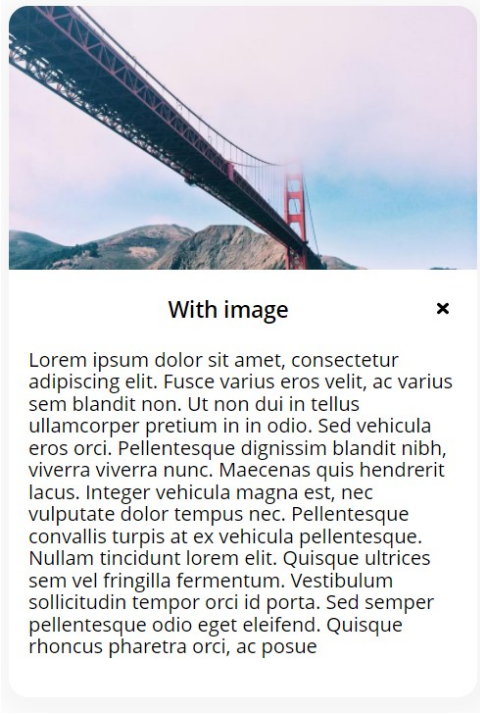
3.2.3.TdtCard

Komponenta TdtCard prikazuje naslov, sadržaj i sliku entiteta bilo koje domene formirajući vizualno u korisničkom sučelju element koji podsjeća na fizičku karticu u stvarnome svijetu. Svojstvo `title` definira naslov kartice. Sadržaj kartice je definiran svojstvom `content`. Uz postojeća svojstva komponenta programeru nudi i mogućnost da samostalno definira odjeljke kartice. Vrijednost svojstva `header` može biti bilo koji HTML-element koji će se pozicionirati u zaglavlja kartice. Ispod zaglavlja se prikazuje vrijednost svojstva `body`, nakon čega dolazi odjeljak definiran preko svojstva `footer`. Na slici 3.9 vidimo kako se u grafičkom sučelju prvo prikazuje tekst `'Default'` što je vrijednost svojstva `header`, potom slijede lorem ipsum teksta koji je vrijednosti svojstava `body`. Na posljetku se prikazuje vrijednost svojstva `footer` smješteno u podnožju komponente s prilagođenom bojom teksta kako se korisniku ne bi odvukla pozornost,



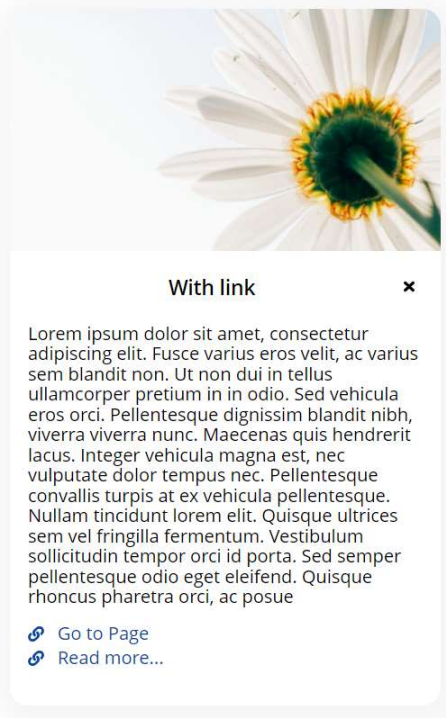
Sl. 3.9 Osnovna varijanta

Kako bi se prikazala slikovita reprezentacija entiteta na koji se kartica odnosi, komponenta nudi svojstvo `image` koje implementira sučelje komponente tdtIcon opisane u poglavlju 3.2.6. Slika je pozicionirana iznad zaglavlja kartice što se vidi na slici 3.10.



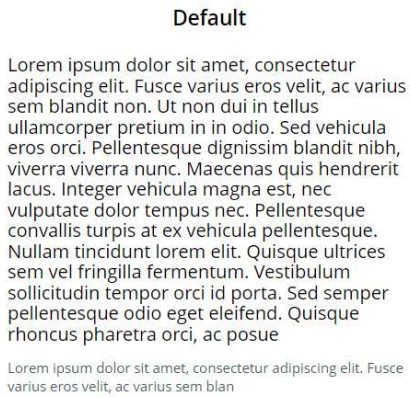
Sl. 3.10 Kartica sa slikom

Ukoliko se ukaže potreba da entitet kartice sadrži poveznicu ne neki URL, komponenta i to omogućuje putem svojstva `links`. U grafičkom sučelju ispod sadržaja kartice prikazuju se definirane poveznice vidljive na slici 3.11.



Sl. 3.11 Kartica s poveznicama

Poput komponente tdtAccordion, kartica je vizualno po z-osi uzdignuta u odnosnu na pozadinu. Kako komponenta implementira sučelje `IFlat` moguće ju je poravnati s pozadinom. Na slici 3.12 vidimo da se poravnavanje postiže na način da se ne koristi oblikovanje rubova korisničkog sučelja komponente, te se ne koristi CSS-svojstvo sjene.

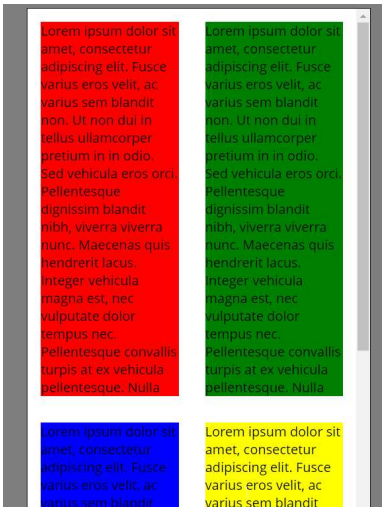


Sl. 3.12. Kartica poravnata po z-osi

Svojstvo `showClose` je tipa boolean. Ukoliko je vrijednost istinita, kartica će pokrenuti animaciju kada korisnik prekrije njenu površinu što korisniku ukazuje na moguću interakciju s komponentom. Klikom na bilo koje područje kartice okida se prilagođeni događaj `onCardClick`. Ukoliko je svojstvo `showClose` istinito klikom korisnika na ikonu za zatvaranje, vidljivu na slici 3.11, okida se prilagođeni događaj `onClose`.

3.2.4.TdtColumns

TdtColumns raspoređuje HTML-elemente djecu po stupcima raspoređujući ih prikladno, ovisno o širini ekrana web preglednika. Svojstvo `columns` je lista brojeva koji predstavljaju broj stupaca na koje će komponenta vizualno raspodijeliti djecu. Prvi element liste predstavlja broj stupaca na uskim ekranima, poput mobitela, slika 3.13.



Sl. 3.13 Mobilni uređaj

Drugi element liste je broj stupaca na ekranima srednje veličine, poput današnjih tableta, slika 3.14, a treći element na svim uređajima veće širine od ekrana srednje veličine, slika 3.15.

Sl. 3.14 Tablet

Sl. 3.15 Desktop

Konkretne vrijednosti za veličine širine ekrana su definirane u datoteci `responsive.scss`, kod 3.1.

```
$mobile-width: 600px;
$tablet-width: 900px;
$desktop-width: 1180px;

@mixin desktop {
  @media screen and (min-width: $desktop-width) {
    @content;
  }
}

@mixin tablet {
```

```
  @media screen and (min-width: $mobile-width) and (max-width:
$desktop-width) {
    @content;
  }
}

@mixin tablet-min {
  @media screen and (max-width: $desktop-width) {
    @content;
  }
}

@mixin mobile {
  @media (max-width: $mobile-width) {
    @content;
  }
}
```

Kod. 3.1 Definicija prilagodljivih širina ekrana i pomoćnih *mixina*

Prema kodu 3.1 može vidjeti da konkretne vrijednosti za veličine mobilnih, tablet i desktop uređaja iznose 600, 900 i 1080 piksela. U datoteci su definirani mixini desktop, tablet, tablet-min i mobile koji prosljeđuju sadržaj uporabom oznake `@content` samo za odgovarajuće širine ekrana definiranjem CSS-selektora `@media`. Mixini u SCSS-sintaksi omogućuju definiranje stilova koji se pozivom oznake `@include ime_mixina` mogu upotrijebiti u bilo kojoj drugoj datoteci sa stilovima, nastavak datoteke je .scss. Ukoliko programer ne definira svojstvo `columns` za svaku širinu ekrana će se prikazati vrijednost svojstva `defaultColumns`. Svojstvo `gaps` je lista stringova koji predstavljaju vizualni razmak između stupaca u sučelju. Analogno svojstvu `columns` se prilagođava različitim širinama ekrana. Ukoliko svojstvo `gaps` nije definirano, razmak na svim širinama ekrana će poprimiti vrijednost svojstva `defaultGap`.

3.2.5. TdtFeedback

Komponenta TdtFeedback služi za prikaz mišljenja fizičke osobe o nekom entitetu te pruža mogućnost ocjenjivanja. Grafičko sučelje prikazuje naslov, komentar, ocjenu, naziv osobe i organizacije koja iznosi komentar. Komponenta nije interaktivna zbog čega ne okida prilagođene događaja. Prikazani naslov je definiran svojstvom `title`, a komentar svojstvom `comment`. Ime fizičke osobe se definira svojstvom `name`, a ukoliko se radi o organizaciji potrebno je definirati svojstvo `company`. Ocjena je predstavljena svojstvom `stars` na temelju čije se vrijednosti u grafičkom sučelju prikazuje odgovarajući broj zvjezdica. Moguće je definirati i decimalnu vrijednost ocijene te će se broj zvjezdica zaokružiti na sljedeću polovičnu vrijednost, primjerice ocjena 4,3 će prikazati četiri pune i jednu polovičnu zvjezdicu u grafičkom sučelju, slika 3.16.

Sl. 3.16 Prikaz ocjene 4.3

3.2.6. TdtIcon

Komponenta TdtIcon prikazuje slikovni sadržaj definiran svojstvom `src` čija je vrijednost poveznica. Komponenta stavlja ograničenja širine na slikovni sadržaj prilagođavajući veličinu raznim širinama ekrana. Ideja je primijeniti komponentu tamo gdje želimo prikazati ikonu unutar nekog drugog HTML-elementa ili web komponente. Ukoliko se želi izbjeći ograničenje širine može se definirati svojstvo `fullWidth`. Širinu i visina se može definirati eksplicitno putem svojstava `width` i `height`. Slikovni sadržaj se animira prelaskom korisničkog kursora nad površinom sadržaja te se ispisuje naslov definiran svojstvom `title`. Kako bi se omogućila interaktivnost potrebno je svojstvo boolean `clickable` postaviti na istinitu vrijednost. Kada je ono istinito prelaskom iznad ikone se pojavljuje kursor što je vidljivo na slici 3.17.

Sl. 3.17 Prekrivanje površine ikone

Ukoliko nije moguće dohvatiti resurs s poveznice zahvaljujući svojstvu `alt` ispisuje se tekstualna poruka. Ukoliko se definira svojstvo `rounded` slikovni sadržaj će biti okrugao što je vidljivo na slici 3.18.

Sl 3.18 Okrugla ikona

3.2.7. TdtInput

Komponenta TdtInput proširuje funkcionalnost i stilizira grafičko sučelje elementa HTML-input. Svojstva `type`, `id` i `name` se direktno prosljeđuju elementu HTML-input. Komponenta prikazuje vrijednost svojstva `value`. Na svaku promjenu vrijednosti od strane korisnika kroz grafičko sučelje komponenta okida prilagođeni događaj `onChange` u čijem se svojstvu `detail` nalazi unesena vrijednost koja je na slici 3.19 '`Input value`'. Iznad prikaza vrijednosti moguće je prikazati i opisni tekst definiran svojstvom `label`.

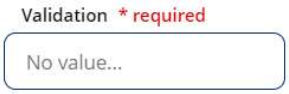
Sl. 3.19 Input s unesenom vrijednosti

Na slici 3.20 vidimo da ukoliko vrijednost nije definirana prikazivat će se vrijednost svojstva `placeholder`, na slici tekst '`No value`'.

Sl 3.20 Input bez definirane vrijednosti

Komponenta nudi pomoćnu funkcionalnost brisanja sadržaja što se u sučelju prikazuje ikonom za brisanje. Ukoliko se funkcionalnost ne želi koristiti potrebno je definirati svojstvo `hideClear`. Dodatna mogućnost koju komponenta pruža je definiranje nužne vrijednosti i

validacije. Na slici 3.21 vidimo da ukoliko je svojstvo `required` istinito prikazivat će se vrijednost svojstva `requiredMessage` ako svojstvo `value` nije definirano.



Sl. 3.21 Validacija nužne vrijednosti

Komponenta se može nalaziti u dva stanja: ispravno i neispravno. Prilikom svake promjene stanja okida se prilagođeni događaj `invalidChange` u čijem će svojstvu `description` biti istinita vrijednost ako je novo stanje ispravno te neistinita vrijednost u slučaju prelaska u neispravno stanje. Kako bi komponenta vršila validaciju potrebno je definirati JavaScriptov validacijski regularni izraz kao vrijednost svojstva `validationRegex`. Prema slici 3.22 vidimo da ako `value` ne zadovoljava definirani regularni izraz vrijednost svojstva `invalid` će biti istinito te će se u sučelju prikazivati vrijednost svojstva `invalidMessage`.

Sl. 3.22 Neispravno stanje

Svojstvo `icon` implementira sučelje komponente tdtIcon opisane u poglavlju 3.2.6. Ovisno o svojstvu `iconPosition`, koje poprima vrijednosti '`left`' ili '`right`', ikona je u odnosu na prikazanu vrijednost svojstva `value` prikazana s lijeve ili desne strane, vidljivo na slikama 3.23 i 3.24.

Sl. 3.23 Ikona s lijeve strane

Sl. 3.24 Ikona s desne strane

3.2.8. TdtLabel

Komponenta TdtLabel prikazuje tekstualni sadržaj i pozicionira opcionalnu ikonu u odnosu na tekst. Prikazani tekst je definiran svojstvom `text`. Klikom na područje komponente se okida prilagođeni događaj `onClick`. Svojstvo `icon` implementira sučelje komponente tdtIcon opisane u poglavlju 3.2.6. Poput komponente tdtInput opisane u prethodnom poglavlju. Komponenta pozicionira ikonu u odnosu na prikazani tekst ovisno o svojstvu `iconPosition` što je vidljivo na slikama 3.25 i 3.26.

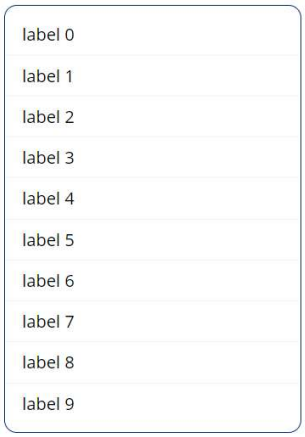
Sl. 3.25 Ikona s lijeve strane Sl. 3.26 Ikona s desne strane

Na slici 3.27 vidimo kako se postavljanjem svojstva `isAnchor` na istinitu vrijednost komponenta stilizira tekst ukazujući korisniku da se radi o poveznici. Klikom na tekst se otvara poveznica koja je definirana kao vrijednost svojstva `href`.

Sl. 3.27 Tekst kao poveznica

3.2.9. TdtList

Komponenta TdtList prikazuje listu tekstualnih stavki. Komponenta implementira sučelje `IListItems`. Sučelje ima svojstvo `items` koje je lista sučelja `IListItem` te svojstvo `onItemClick` koje je po tipu prilagođeni događaj. Svakim klikom na jedan od elemenata liste items se okida događaj `onItemClick` u čijem se svojstvu `detail` nalaze svojstva `IListItem` sučelja kliknutog elementa. Sučelje ima svojstva `id` i `label` gdje je vrijednost svojstva `label` prikazani tekst. Komponenta `TdtList` implementira tri varijante: interaktivna, statička i sadržana. Na slici 3.28 je prikazana statička varijanta koja prikazuje svaku stavku jednu ispod druge.

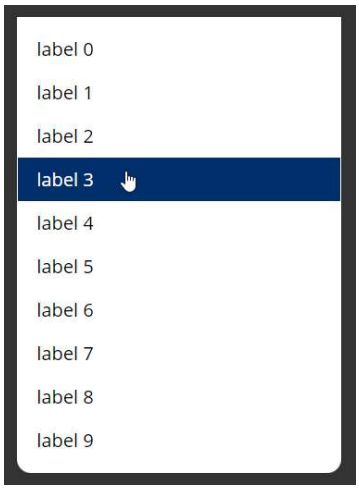


Sl. 3.28 Statička varijanta

Na slici 3.29 je prikazana interaktivna varijanta koja stilizira tekst prilikom prelaska kursora preko stavke ukazujući korisniku na mogućnost interakcije s komponentom.

Sl. 3.29 Interaktivna varijanta

Sadržana varijanta je također interaktivna, no njezino sučelje je drugačije jer se ono koristi u slučaju kada druga komponenta prikazuje listu. Kako bi se prostor koji lista zauzima u grafičkom korisničkom sučelju smanjio na slici 3.30 vidimo da se prostor ispune između stavki smanjio.



Sl. 3.30 Sadržana varijanta

Svojstvo komponente `type` može poprimiti vrijednosti `'bullet'` i `'number'`. Ukoliko je vrijednost `'bullet'` lista će prije teksta umetnuti znak koji ukazuje na nepobrojanu listu stavku, vidljivo na slici 3.31.

Sl. 3.31 Bullet lista

Na slici 3.32 je prikazano sučelje kada je vrijednost svojstva `type 'number'`, komponenta prije teksta umeće odgovarajući redni broj stavke.

Sl. 3.32 Number lista

3.2.10. TdtPagination

Komponenta TdtPagination omogućuje implementaciju prikaza vrlo duge liste entiteta po stranicama. Grafičko sučelje korisniku prikazuje trenutnu stranicu i nudi mogućnost odabira ostalih stranica. Svojstvo `current` je indeks trenutne stranice, `total` je ukupan broj stranica, a `stepsNumber` predstavlja broj susjednih stranica koje će biti numerički označene u sučelju. U grafičkom sučelju se nalaze ikone kojima se može prijeći na posljednju i početnu stranicu. Klikom na ikonu za prelazak na prvu stranicu se okida prilagođeni događaj `onFirst`, a klikom na ikonu za prelazak na posljednju stranicu događaj `onLast`. Na slici 3.33 vidimo kako je indeks trenutne stranice posebno istaknut u odnose ne neaktivne stranice.

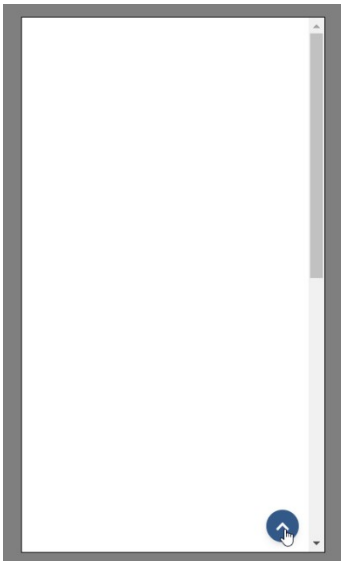
Sl. 3.33 Ikone za posljednju i početnu stranicu

Moguće je prikazati ikone za prelazak na slijedeću i prethodnu stranicu ukoliko se definiraju svojstva `showPrevious` i `showNext`. Klikom na ikone će se okidati prilagođeni događaji `onPrevious` i `onNext`. Na slici 3.34 vidimo grafičko sučelje ikona za prelazak na slijedeću i prethodnu stranicu.

Sl. 3.34 Ikone za prethodnu i slijedeću stranicu

3.2.11. TdtScrollToTop

Komponenta TdtScrollToTop korisničkim klikom uzrokuje klizanje stranice do vrha trenutnog prozora. Prema slici 3.35 vidimo kako grafičko sučelje komponente prikazuje okrugli gumb uvijek pozicioniran u donjem desnom kutu prozora preglednika. Svojstvo `title` prikazuje naslov kada korisnik pređe kursorom preko gumba. Prilikom klika se okida prilagođeni događaj `scrollTo` sa svojstvom `detail` čija je vrijednost jednaka vrijednosti svojstva `title`.



Sl. 3.35 Gumb za klizanje do vrha

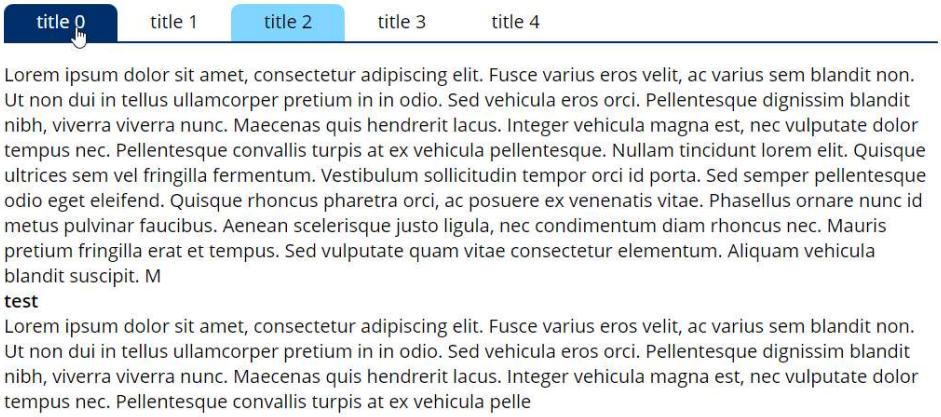
3.2.12. TdtSteps

Komponenta TdtSteps omogućava pregled koraka u višedijelnome procesu. Svojstvo `steps` je lista sučelja `IStep` sa svojstvima `number`, redni broj koraka, i `label`, tekstualni opis koraka. Svojstvo `activeStep` predstavlja indeks trenutnog koraka. Na slici 3.36 vidimo da grafičko sučelje korisniku sugerira koji je korak trenutno aktivan. Prelaskom kursora preko određenog koraka zahvaljujući prilagodbi sučelja se upućuje korisnika na interakciju, odnosno da se može kliknuti na korak. Klikom na korak se okida prilagođeni događaj `onStepClick` u čijem će svojstvu `detail` biti upisan indeks kliknutog koraka.

Sl. 3.36 Pregled koraka

3.2.13. TdtTabs

Komponenta TdtTabs omogućava promjenu trenutnog prikaza i omogućava organiziranje sadržaja prema kategorijama. Svojstvo `tabs` je lista ITabItem sučelja koje ima svojstva `title`, `id` i `content`. U svakome trenutku je aktivna jedna od kartica te se prikazuje njezin sadržaj definiran svojstvom `content` koje je po tipu bilo koji HTML-element. Definiranje aktivne kartice se ostvaruje putem svojstva `activeTabId`. Na slici 3.37 vidimo da korisnik na temelju grafičkog prikaza sučelja može uočiti koja je kartica aktivna. Prelaskom kursora preko neaktivne kartice korisnik razumije da se kartice mogu mijenjati kako bi se odabrao drugi sadržaj za prikaz. Sadržaj ostalih kartica je skriven te se mogu vidjeti samo njihovi naslovi definirani svojstvom `title`.

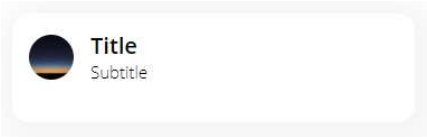


Sl. 3.37 Aktivna kartica i prelazak kursorom preko kartice

Promjenom aktivne kartice komponenta okida prilagođeni događaj `onTabChange` čije svojstvo `detail` sadrži sučelje nove aktive kartice `ITabItem` s odgovarajućim vrijednostima kliknutog elementa iz liste tabs.

3.2.14. TdtTeaser

Komponenta TdtTeaser grupira naslov, podnaslov, prikazuje sliku i tekstualni opis. Namijenjena je da se koristi kako bi se ukratko opisao neki entitet i sažele najvažnije informacije. Implementira sučelje `ITeaser`. Prikazani naslov i podnaslov su definirani svojstvima `title` i `subtitle`. Na slici 3.38 vidimo da okrugla slika pored tekstualnog sadržaja semantički opisuje ono na što se sadržaj odnosi. Definirana je putem svojstva `image` koje implementira sučelje komponente tdtIcon opisane u poglavlju 3.2.6. Klikom na sliku se okida prilagođeni događaj `onImageClick` u čijemu je svojstvu `detail` upisana vrijednost svojstva `image`.



Sl. 3.38 Osnovno sučelje

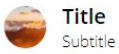
Ukoliko je definirano svojstvo `details`, koje može biti bilo koji HTML-element, komponenta prikazuje sadržaj. Na slici 3.39 je kao vrijednost svojstva definiran kratak isječak lorem ipsum teksta.

Sl. 3.39 Prikaz svojstva details

Komponente sadrži svojstvo `sideContent` koje također može biti bilo koji HTML-element, a u grafičkom sučelju na se prikazuje pored sadržaja. Svojstvo se može koristi primjerice za implementaciju akcija klika na gumb ili kako bi se umetnula poveznica. Na slici 3.39 vrijednost svojstva je HTML-element koji predstavlja gumb vidljiv u desnom rubu grafičkog sučelja komponente. Definiranjem svojstva `clickable` komponenta postaje interaktivna. Na slici 3.40 vidimo kako se postavljanjem svojstva na istinitu vrijednost komponenta prilagođava sučelje kada korisnik kursorom prekrije površinu čime ukazuje na mogućnost klika. Klikom se okida prilagođeni događaj `onTeaserClick` u čijem će svojstvu `detail` biti definirana sva svojstva komponente. Pošto stranice vrlo često prikazuju više primjeraka komponente nužno je moći spoznati na koju se komponentu kliknulo.

Sl.3.40 Interaktivna komponenta

Komponenta je vizualno uzdignuta u odnosu na pozadinu. Navedeno je moguće promijeniti postavljanjem svojstva `isFlat` na istinitu vrijednost čime se grafičko sučelje po z-osi poravnava sa pozadinom, što je vidljivo na slici 3.41. Usporedivši je sa slikom 3.39 vidimo da komponenta nema zakrivljeni rub te se ne koristi CSS-svojstvo sjene za postizanje efekta uzdizanja komponente u odnosu na pozadinu.



Sl. 3.41 Poravnanje po z-osi

3.3. Molekule

3.3.1. TdtAccordions

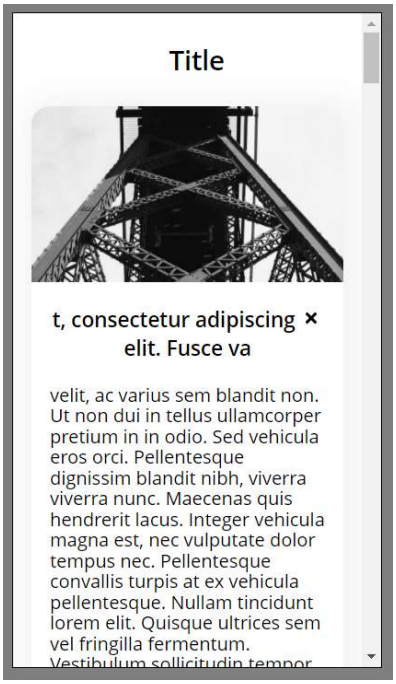
Komponenta TdtAccordions proširuje mogućnosti i u grafičkom sučelje organizira više tdtAccordion komponenti. Svojstvo `title` prikazuje naslov ispod čega se prikazuje lista tdtAccordiona te se omogućuje čuvanje stanja elemenata liste. Istovremeno je moguće da nije otvoren niti jedan element liste ili da je otvoreno proizvoljno mnogo elemenata. Prilikom svakog otvaranja ili zatvaranja jednog od elemenata komponenta generira prilagođeni događaj `onAccordionOpenChange` u čijem se svojstvu `detail` nalaze sva svojstva odgovarajućeg elementa lista. Na slici 3.42 je prikazano grafičko sučelje s otvorenim i zatvorenim elementima liste, te se vidi interakcija komponente s korisnikom.

Sl. 3.42 Interakcija korisnika, prikazani otvoreni i zatvoreni elementi liste

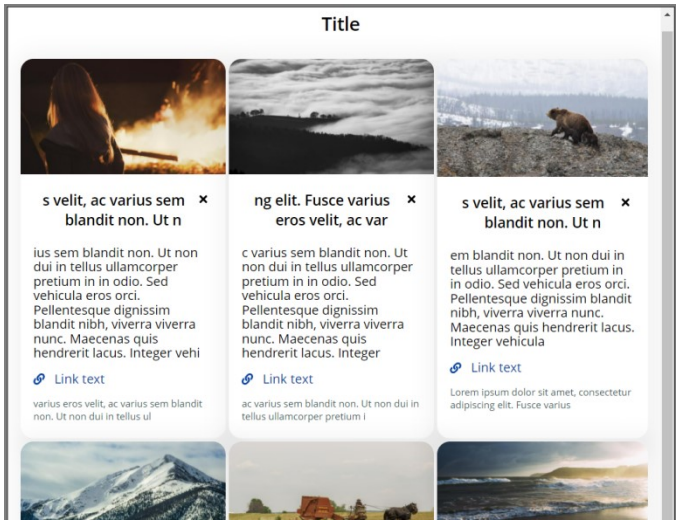
3.3.2. TdtCardsGrid

Komponenta TdtCardsGrid pozicionira elemente liste tdtCard komponenti u grafičkom sučelju prilagođavajući se različitim širinama ekrana. Lista je definirana svojstvom `cards`. Prikazuje se naslov definiran svojstvom `title`. Sučelje komponente omogućuje definiranje mreže stupaca i redaka u koje će elementi liste biti raspodijeljeni. Stupci su ostvareni svojstvom

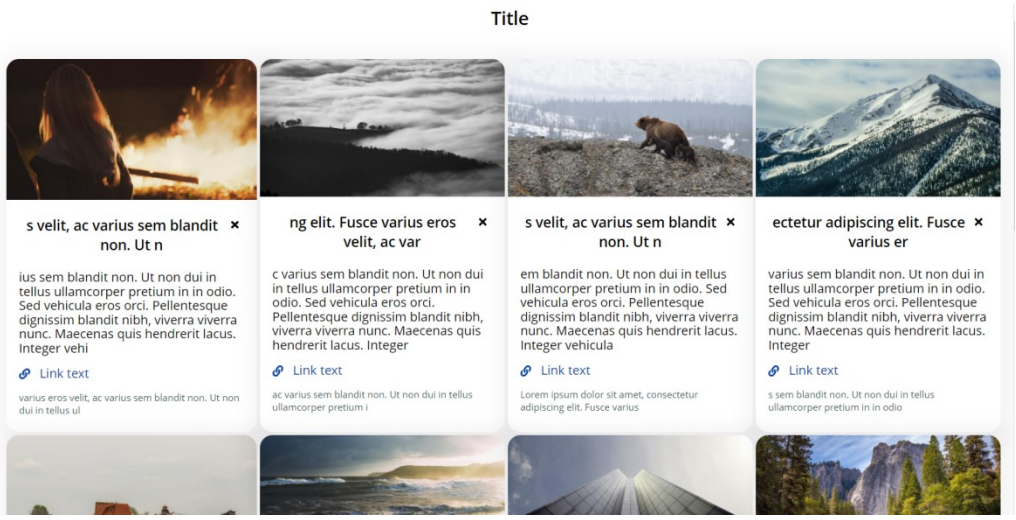
`columns`, a redci svojstvom `rows`. Svojstva su liste čiji prvi element definira vrijednost za mobilne, druga vrijednost za tablet, a treća veličina za desktop širine ekrana. Na slikama 3.43-45 prikazane su različite varijante organizacije elemenata liste `cards` ovisno o širina ekrana.



Sl. 3.43. Mobilni uređaji



Sl. 3.44 Tableti



Sl. 3.45 Desktop uređaji

Komponenta okida prilagođene događaje te u svojstvo `detail` upisuje sva svojstva referentne komponente tdtCard. Događaj `onCardClick` se okida na svaki klik kartice, a događaj `onCardClose` u trenutku zatvaranja kartice.

3.3.3. TdtDropdown

Komponenta TdtDropdown proširuje funkcionalnost i grafičko sučelje komponente tdtList omogućujući korisniku interakciju klikom na element liste. Komponenta koristi sadržanu varijantu tdtList komponente definirajući njezino svojstvo `variant`. Budući da komponenta implementira sučelje `IListItems` za nju vrijedi sva funkcionalnost sučelja opisana u poglavlju 3.2.9. Uz listu, prikazuje se i oznaka definirana svojstvom `label`. Komponenta se u svakom trenutku nalazi u jednom od dva stanja: zatvoreno ili otvoreno. Ukoliko je stanje zatvoreno lista u grafičkom sučelju nije vidljiva te se prikazuje samo oznaka, slika 3.46.

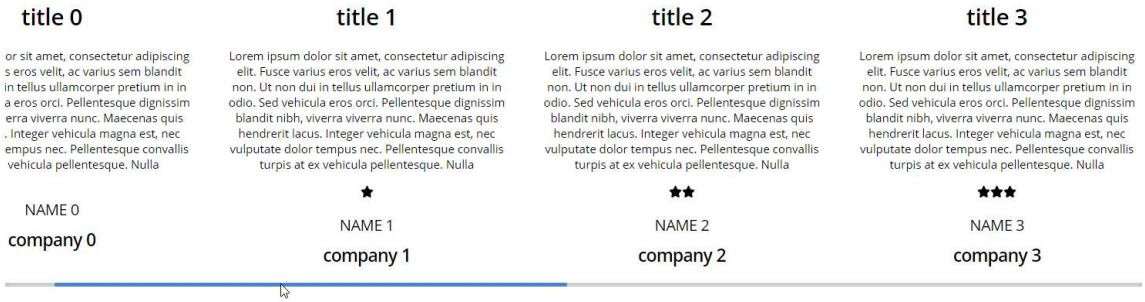
Sl. 3.46 Zatvoreno stanje

Na slici 3.47 vidimo komponentu u stanju otvoreno, ispod oznake je vidljiva lista elemenata definirana svojstvom `items`.

Sl. 3.47 Otvoreno stanje

3.3.4. TdtFeedbacks

Komponenta TdtFeedbacks raspoređuje tdtFeedback komponente u grafičkom sučelju. Prikazuje naslov definiran svojstvom `title` ispod čega slijedi klizeća lista komponenti tdtFeedback definirana svojstvom `feedbacks`. Komponenta nije interaktivna s obzirom da ne okida događaje jer je njezina primarna funkcionalnost vizualni prikaz vidljiv na slici 3.48.



Sl. 3.48 Komponenta TdtFeedbacks

3.3.5. TdtFooter

Komponenta TdtFooter služi za prikaz kratkih i najvažnijih informacija o web stranici. Definirani sadržaj se raspoređuje u stupce te se uobičajeno pozicionira u podnožju stranice. Odjeljci sadržaja, raspoređeni po stupcima, su predstavljeni svojstvom `sections` koje je lista sučelja IFooterSection. Sučelje ima svojstvo `title` koje prikazuje naslov stupca te svojstvo `items` koje je lista tdtLabel komponenti opisanih u poglavlju 3.2.8. zbog čega svaki element liste nasljeđuje sve funkcionalnosti komponente. Na slici 3.49 vidimo kako se svaki element prikazuje unutar trenutnog stupca slijedno jedan ispod drugog. Klikom na element se okida prilagođeni događaj `onItemClick` koji je po svojstvima jednak događaju `onClick` komponente tdtLabel. Komponenta svojstvom `variant` implementira tri varijante: osnovno, primarno i sekundarno. Osnovna varijanta ne sadrži stiliziranu pozadinu.

Sl. 3.49 Osnovna varijanta

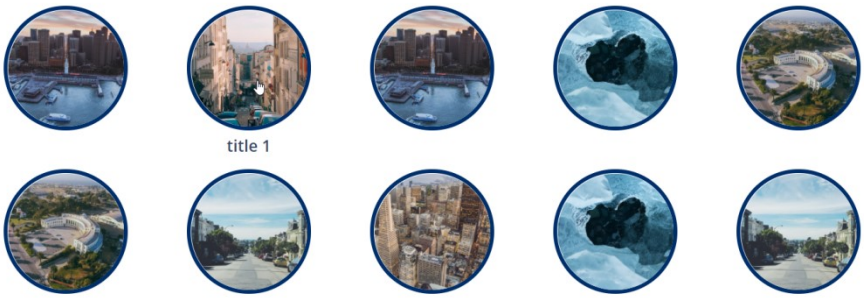
Na slikama 3.50 i 3.51 su prikazane primarna i sekundarna varijanta koje nasljeđuju prikladne boje iz teme.

Sl. 3.50 Primarna varijanta

Sl. 3.51 Sekundarna varijanta

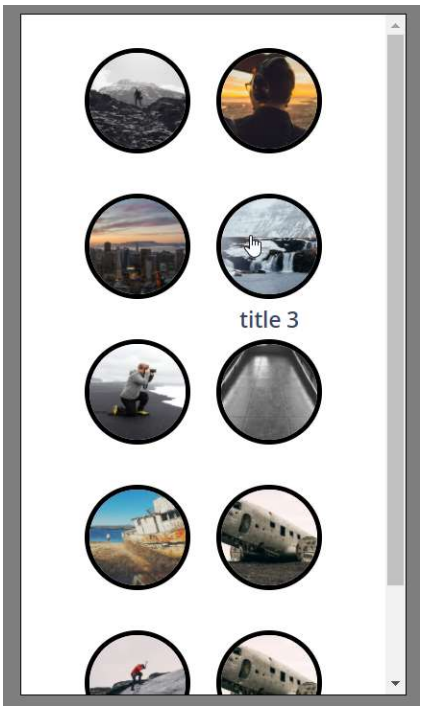
3.3.6. TdtLogoCloud

Komponenta TdtLogoCloud prikazuje stiliziranu listu slika opisanih kratkim naslovom. Lista je definirana svojstvom `logos` koje je lista sučelja HTML-elementa `img`. Slike se raspoređuju u prostoru te se njihov raspored prikladno prilagođava različitim širinama ekrana. Kako je komponenta interaktivna, prelaskom kursora preko slike se pojavljuje odgovarajući naslov ispod slike definiran svojstvom `title` HTML-elementa img. Na slici se kursor nalazi iznad drugog elementa te sučelje korisnik ukazuje da je moguće kliknuti na sliku. Klikom se okida prilagođeni događaj `onLogoClick` u čije se svojstvo `detail` upisuju sva svojstva odgovarajućeg elementa `img`.



Sl. 3.52 Desktop uređaj

Sl. 3.53 Tablet



Sl. 3.54 Mobilni uređaj

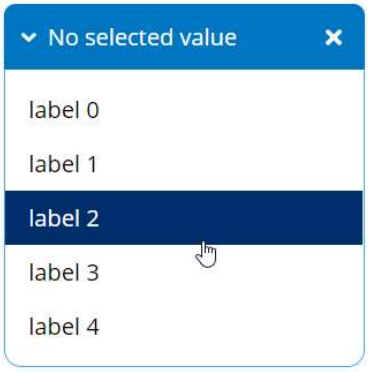
3.3.7. TdtMenu

Komponenta TdtMenu omogućuje dinamički prikaz liste. Lista se otvara klikom na element koji je dijete komponente. Komponenta implementira sučelje `IListItems` stoga vrijede sve funkcionalnosti opisane u poglavlju 3.2.9. Svojstvo `menuPosition` poprima vrijednosti `'left'` i `'right'` i pozicionira dinamičku prikazanu listu u odnosu na element dijete lijevo ili desno. Komponenta može biti u stanju otvoreno ili zatvoreno. Svakom promjenom stanja se okida prilagođeni događaj koji u svojemu svojstvu `detail` sadrži istinu boolean vrijednost ako je novo stanje otvoreno, a neistinitu ako je novo stanje zatvoreno. Na slici 3.55 je vidljivo otvoreno stanje.

Sl. 3.55 Otvoreno stanje

3.3.8. TdtSelect

Komponenta TdtSelect omogućuje odabir vrijednosti iz padajuće liste koja je ostvarena komponentom tdtList. Komponenta implementira sučelje `IListItems` stoga vrijede sve funkcionalnosti opisane u poglavlju 3.2.9. Na slici 3.56 je prikazano otvoreno stanje komponente.



Sl. 3.56 Otvoreno stanje

Svojstvo `currentItemId` određuje trenutno odabrani element liste te po vrijednosti odgovara svojstvu `id` odabranog elementa liste svojstva `items`. Prema slici 3.57 vidimo da ukoliko nije odabran niti jedan element liste komponenta prikazuje tekstualni sadržaj koji je vrijednost svojstva `placeholder`.

Sl. 3.57 Prikaz *placeholder* vrijednosti

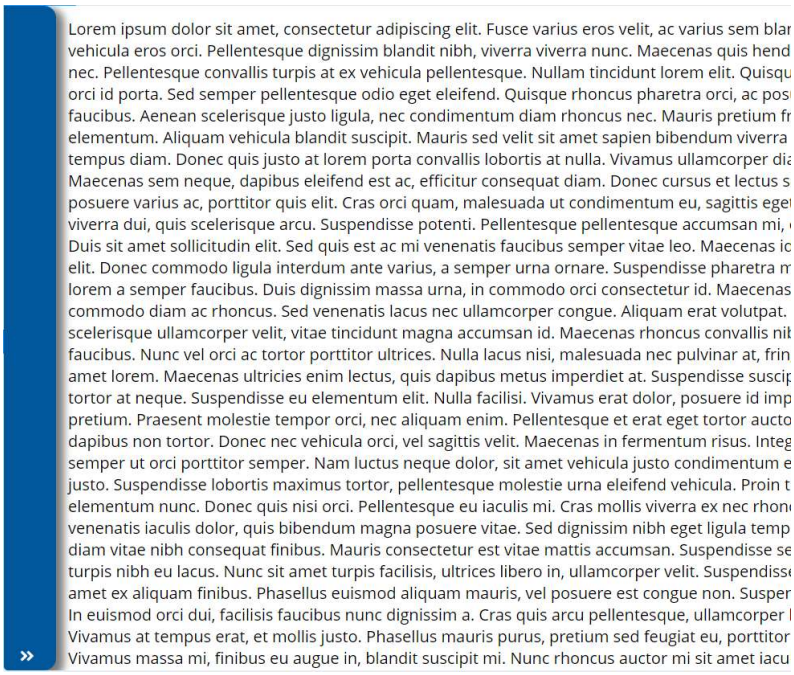
Svojstvo `selected` u svakome trenutku sadrži odabrani element liste. Prilikom promjene odabranog elementa komponenta okida prilagođeni događaj `onValueChange` u čijem je svojstvu `detail` upisana vrijednost odabranog elementa liste. Na slici 3.58 je prikazano sučelje komponente ukoliko postoji odabrani element.

Sl. 3.58 Odabrani element

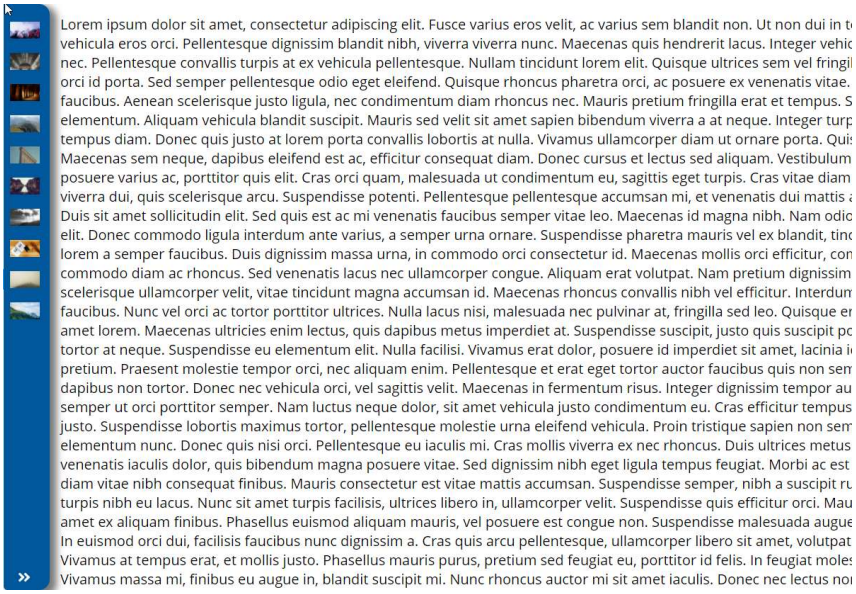
Na slici 3.58 je vidljivo kako komponenta u krajnjem desnom rubu grafičkog sučelja prikazuje ikonu čijim se klikom uklanja odabrani element, nakon čega se prikazuje vrijednost svojstva `placeholder`. Okida se događaj `onValueChange` u čijem je svojstvu `detail` upisana vrijednost *null*.

3.3.9. TdtSidebar

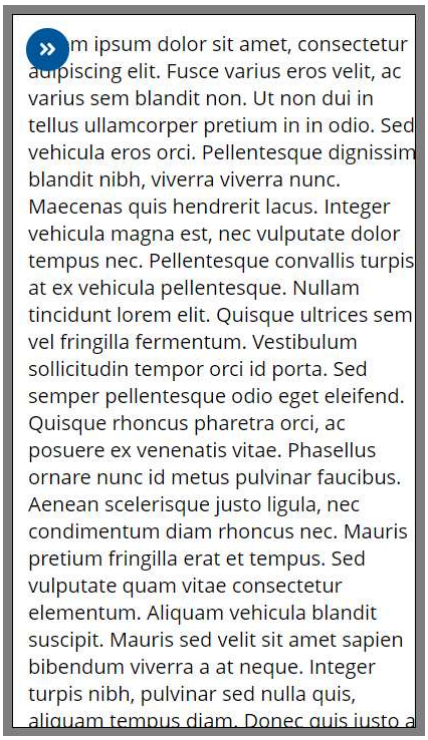
Komponenta TdtSidebar prikazuje sadržaj i klizni izbornik s lijeve strane. Grafičko sučelje se prilagođava različitim širinama ekrana (slike 3.59 – 3.61) kako bi osiguralo optimalno korisničko iskustvo.



Sl. 3.59 Zatvoreno stanje bez definiranih ikona, desktop uređaj

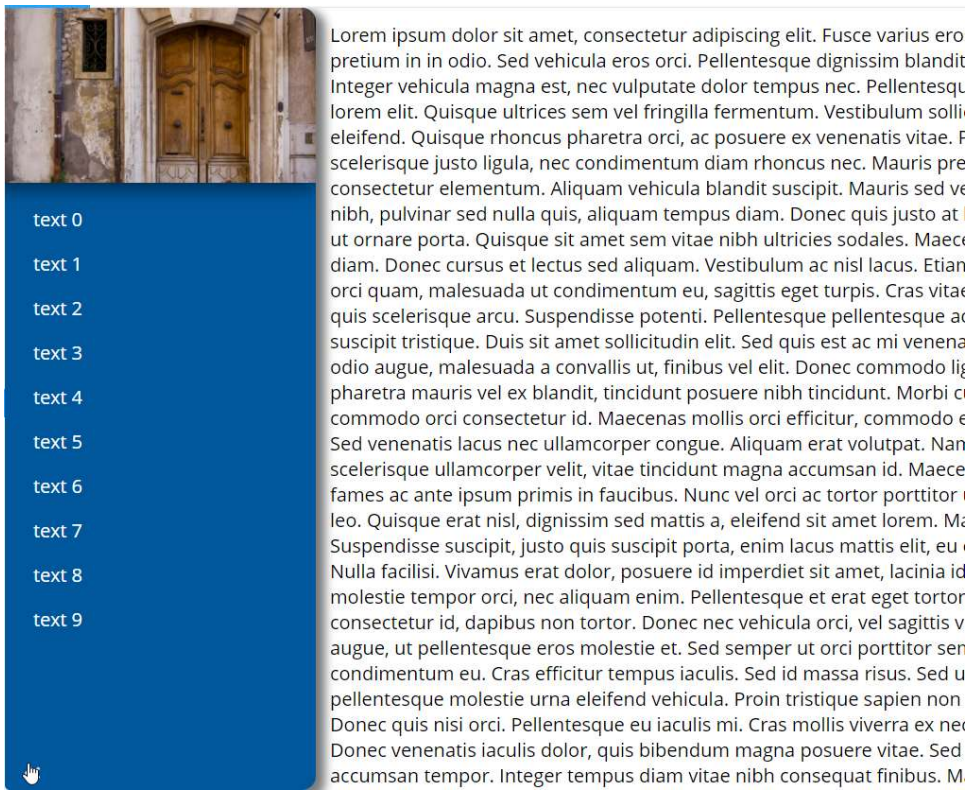


Sl. 3.60 Zatvoreno stanje s definiranim ikonama, desktop uređaj

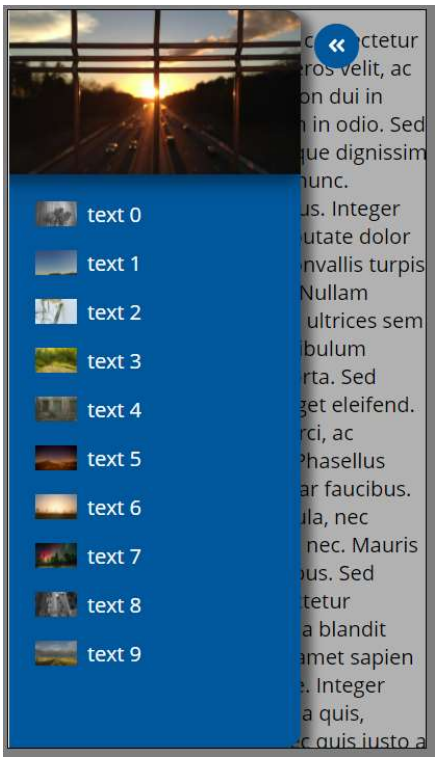


Sl. 3.61 Zatvoreno stanje, mobilni uređaj

Prikazani sadržaj komponente čine HTML-elementi djeca komponente. Komponenta može biti u stanju otvoreno ili zatvoreno. Na slikama 3.59 i 3.60 možemo vidjeti da se u stanju zatvoreno prikazuje samo uska površina izbornika te ikone ukoliko su definirane. Prilagođavajući se mobilnim uređaji prema slici 3.61 možemo vidjeti kako se umjesto površine izbornika prikazuje ikona za promjenu stanja. Klizni izbornik se otvara klikom na odgovarajuću ikonu nakon čega su vidljive stavke izbornika što je vidljivo na slikama 3.62 i 3.63. Klikom na stavku se okida prilagođeni događaj `onLabelClick` u čijem je svojstvu `detail` definirana odgovarajuća stavka. Pri vrhu izbornika je moguće postaviti logo putem svojstva `logo` koje implementira sučelje `IIcon`. Klikom na ikonu se okida prilagođeni događaj `onLogoClick`.

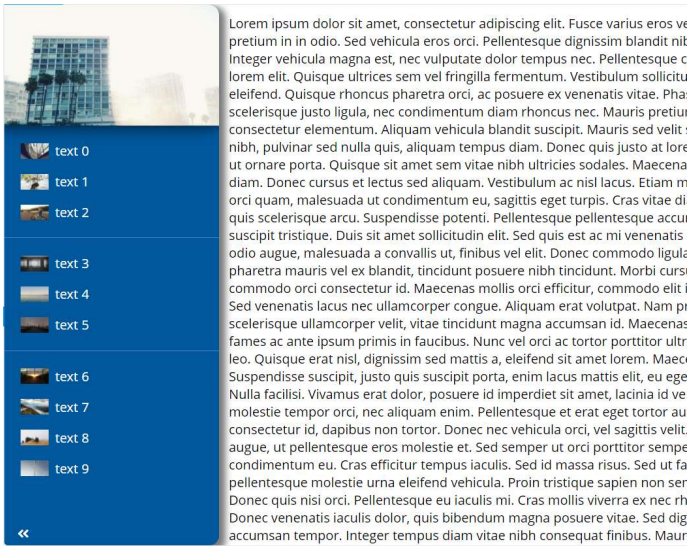


Sl. 3.62 Otvoreno stanje bez definiranih ikona, desktop uređaj



Sl. 3.63 Otvoreno stanje s definiranim ikonama, mobilni uređaj

Svojstvo `collapsed` će u svakome trenutku odgovarati stanju izbornika. Ono će biti istinito kada je izbornik otvoren, a lažno kada je zatvoren. Promjenom stanja se okida prilagođeni događaj `onCollapseChange` u čijem se svojstvu `detail` nalazi istinita vrijednost ako je novo stanje otvoreno, a neistinita vrijednost inače. Stavke izbornika su definirane svojstvom `sections`. Kako izbornik podržava više grupa stavki svojstvo je lista listi sučelja `ILabel`. Prema slici 3.64 vidimo kako su grupe odvojene horizontalnom linijom.



Sl. 3.64 Grupe stavki

3.3.10. TdtSlides

Komponenta TdtSlides korisniku omogućuje listanje slika. Slike koje se mogu listati su definirane svojstvom `slides` koje je lista HTML-elemenata img. S trenutne slike se može prijeći na slijedeću ili prethodnu klikom na odgovarajući ikonu. Naravno, ukoliko ne postoji prethodna ili slijedeća slika onemogućen je klik, što je vidljivo na slici 3.65.

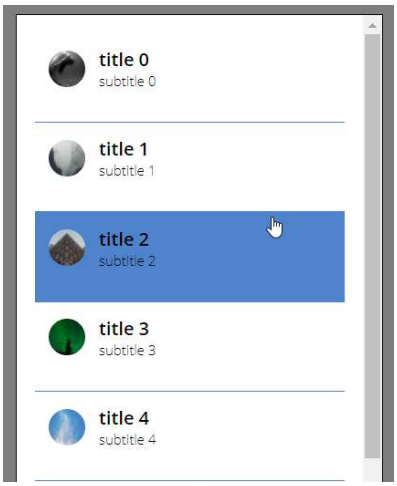
Sl. 3.65 Početna slika

Promjenom sadržaja se okida prilagođeni element onSlideChange u čijem se svojstvu `detail` nalazi odgovarajući element liste slides. Svojstvo `current` odgovara indeksu trenutno

projicirane slike. Programer može definirati visinu slika svojstvom `height`, komponenta uvijek zauzima punu širinu elementa roditelja.

3.3.11. TdtTeaserList

Komponenta TdtTeaserList prikazuje listu komponenti tdtTeaser raspoređujući ih prikladno u grafičkom sučelju. Elementi liste, definirani svojstvom `teasers`, se prikazuju slijedno te su poravni sa pozadinom po z-osi. Prema slici 3.66 vidimo kako se prekrivanjem kursora preko površine elementa liste njegovo grafičko sučelje prilagođava ukazujući korisniku da je moguća interakcija s komponentom.



Sl. 3.66 Interakcija korisnika

Klikom na jedan od elemenata liste se okida prilagođeni događa `onTeaserClick` u čijem su svojstvu `detail` definirana sva svojstva odgovarajuće tdtTeaser komponente.

3.3.12. TdtTextWithImage

Komponenta TdtTextWithImage omogućuje različite varijante grafičkog sučelja prikazujući tekstualni sadržaj i sliku. Slika je definirana svojstvom `image` koje je po tipu HTML-element

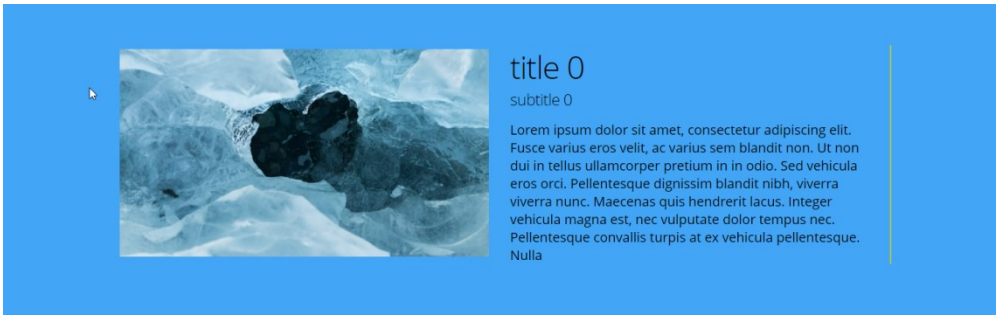
`img`. Prikazani naslov, podnaslov i tekst su definirani svojstvima `title`, `subtitle` i `text`. Svojstvo variant može poprimiti vrijednosti '`primary`' i '`secondary`', pri čemu se boje za tekstualni sadržaj i pozadinu uzimaju iz definirane teme biblioteke. Ukoliko se ne žele koristiti varijante moguće je putem svojstva `backgroundColor` definirati boju pozadine.
Komponenta uz tekstualni sadržaj i sliku omogućuje definiranje akcijskih gumba svojstvom `actionButtons` koje je lista sučelja `IButton` zbog čega svaki od gumba podržava sve funkcionalnosti opisane u poglavlju 3.2.2. osim što se umjesto događaja `onClick` okinuti događaj naziva `onButtonClick`. Najvažnije svojstvo komponente je `imagePosition` koje može poprimiti vrijednosti pobrojane enumeracijom `PositionsEnum` prikazane u kodu 3.2.

```
export enum PositionsEnum {
  LEFT = 'left',
  RIGHT = 'right',
  TOP = 'top',
  BOTTOM = 'bottom',
  BACKGROUND = 'background',
}
```

Kod 3.2 Enumeracija pozicija

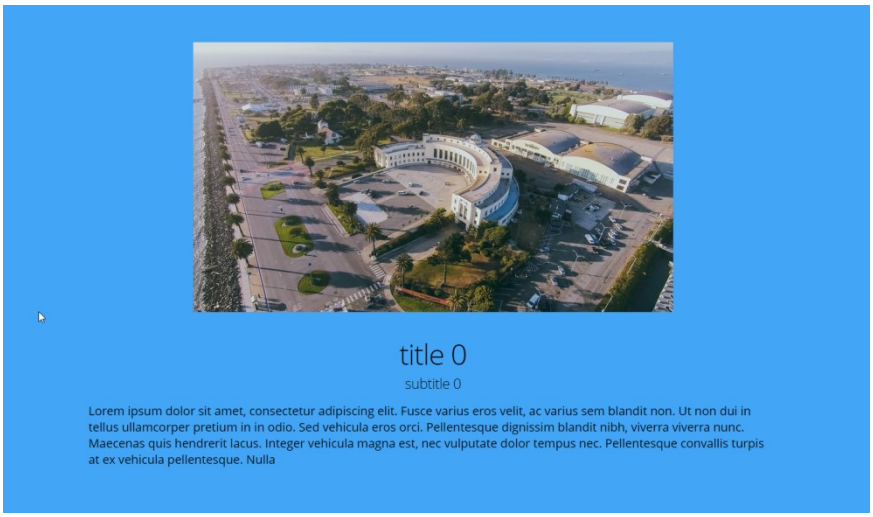
Na sljedećim slikama (3.67 – 3.71) su prikazani svi mogući položaji slika i varijante grafičkog sučelja.

Sl. 3.67 Sekundarna varijanta, slika s desna

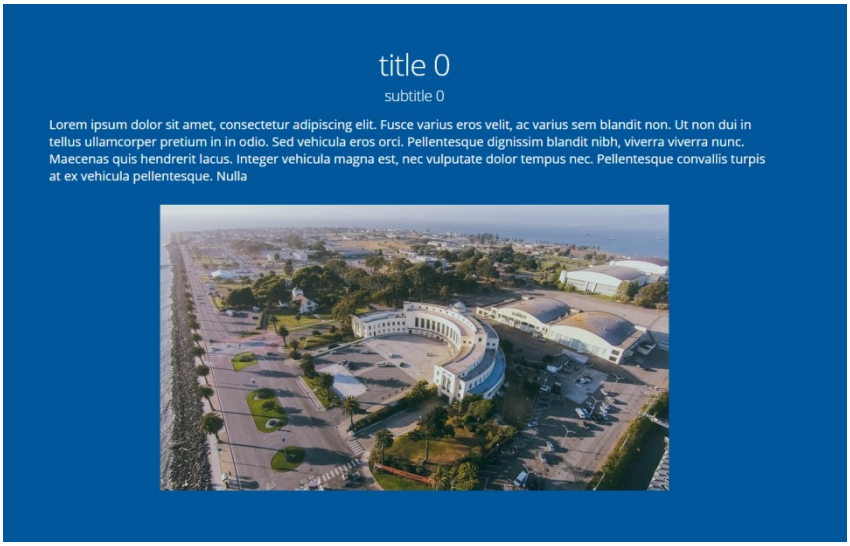


Sl. 3.68 Primarna varijanta, slika s lijeva

Sl. 3.69 Slika kao pozadina



Sl. 3.70 Primarna varijanta, slika na vrhu

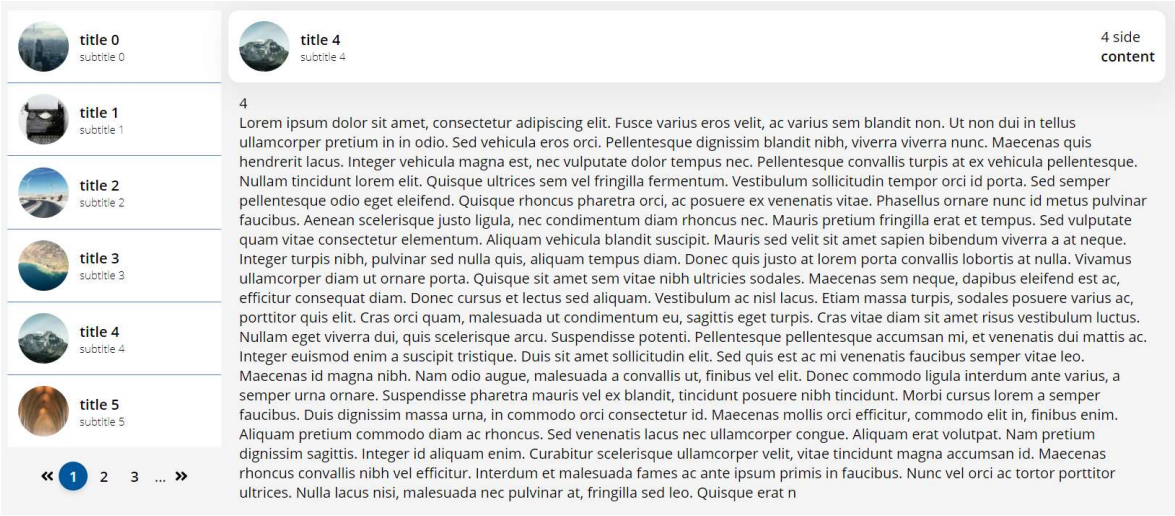


Sl. 3.71. Sekundarna varijanta, slika pri dnu

3.4. Organizmi

3.4.1. TdtDetailPanel

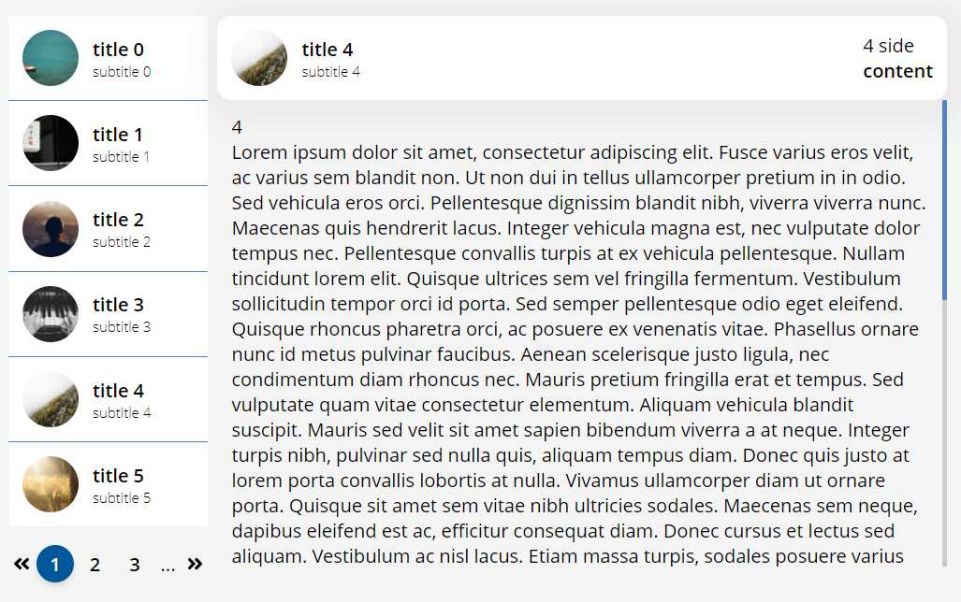
Komponenta TdtDetailPanel omogućuje sažeti prikaz mnoštva entiteta i detaljan prikaz jednog istovremeno. U grafičkom sučelju raspoređuje komponente tdtTeaser, tdtPagination i tdtTeaserList. Upravlja njihovom komunikacijom i suradnjom kako bi proširila funkcionalnosti. Na slici 3.72 vidimo grafičko sučelje komponente koje omogućuje sažeti pregled liste entiteta po stranicama pozicioniranih u lijevom rubu sučelja. Klikom na element liste se prikazuju detalji definirani za odabrani element. Komponenta je primjenjiva u raznim domenama, jer se vrlo često ukazuje potreba za sažetim i detaljnim pregledom entiteta.



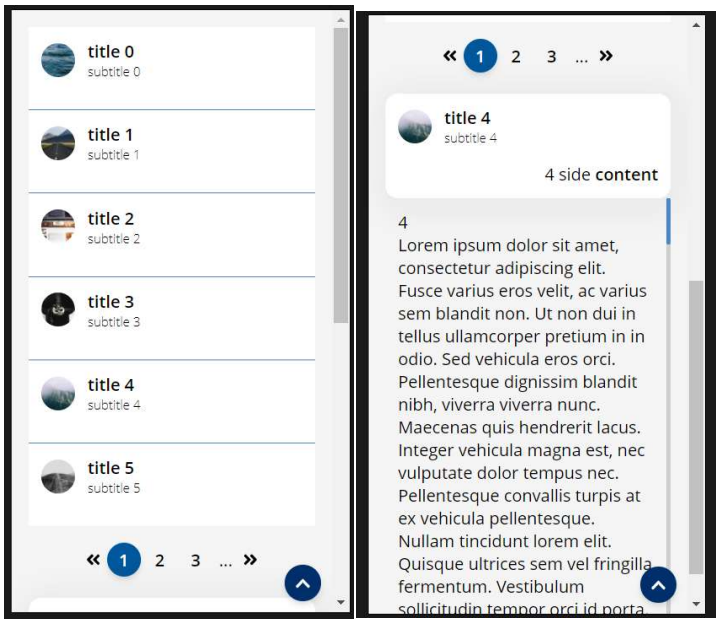
Sl. 3.72 Pregled detalja i sažetaka

Lista elemenata se definira svojstvom `teasers` koje je lista sučelja `ITeaser` opisanih u poglavlju 3.2.14. zbog čega za svaki element liste vrijede opisane funkcionalnosti. U svakome trenutku je dostupna informacija o rednom broju elementa liste čiji se detalji prikazuju putem svojstva `current`. Programer može postaviti svojstvo dinamički što će uzrokovati promjenu sučelja prikazujući detalje odgovarajućeg elementa. Ukoliko nije definirano svojstvo `current` prikazivati će se element liste na poziciji u listi čiji redni broj odgovara vrijednosti svojstva `defaultIndex`. Broj elemenata liste koji će se prikazivati po svakoj stranici je definiran svojstvom `teasersPerPage`. Kako bi se omogućila fleksibilnost programer može definirati vlastitu komponentu za prikaz i rukovanje stranica putem svojstva `pagination` koja je po

tipu HTML-element. Važno je spomenuti da se tada neće prikazivati sučelje komponente tdtPagination, niti će se koristiti njezina funkcionalnost zbog čega programer mora upravljati prikazanim detaljima i elementima liste samostalno postavljanjem svojstava `current` i `teasers` prikladno. Svakim klikom na element liste okida se prilagođeni događaj `onTeaserClick` što je jednako događaju opisanome u poglavlju 3.2.14. Prema slikama 3.73 i 3.74 vidimo da se komponenta prilagođava različitim širinama ekrana. Po potrebi se smanjuju dimenzije elemenata sučelja te se mijenja njihov raspored kako bi se osiguralo najbolje korisničko iskustvo.



Sl. 3.73 Prilagodba dimenzija za tablet



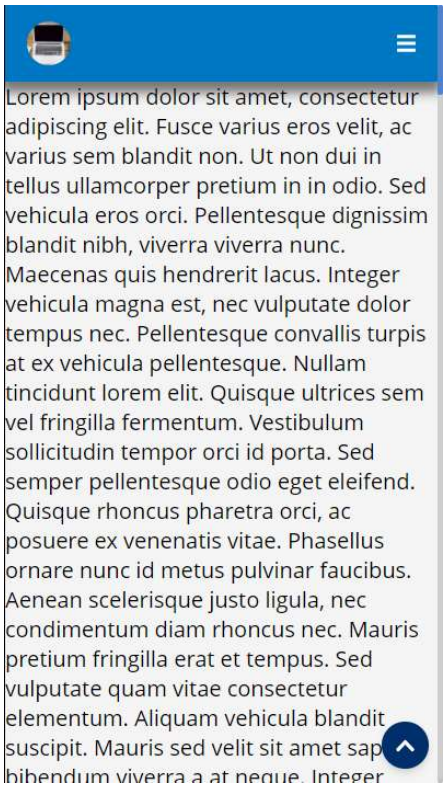
Sl. 3.74 Prilagodba rasporeda za mobilne uređaje

3.4.2. TdtNavbar

Komponenta TdtNavbar omogućuje navigaciju kroz aplikaciju, prikaz loga i liste postavki. Grafičko sučelje komponente je pozicionirano u zaglavlju. Moguće je prikazati logo aplikacije koji će uvijek biti vidljiv u gornjem lijevom rubu zaglavlja. Definira ga svojstvo `logo` koje implementira sučelje komponente tdtIcon opisane u poglavlju 3.2.6. zbog čega vrijede sve spomenute funkcionalnosti, osim što se umjesto prilagođenog događaja `onClick` okida događaj naziva `onLogoClick` jednakog sučelja. Oznake navigacije su ostvarene svojstvom `navItems` koje je lista sučelja `IListItem` opisanog u poglavlju 3.2.9. Klikom na oznaku se okida prilagođeni događaj `onNavItemClick` u čijem se svojstvu `detail` nalazi odgovarajući element.

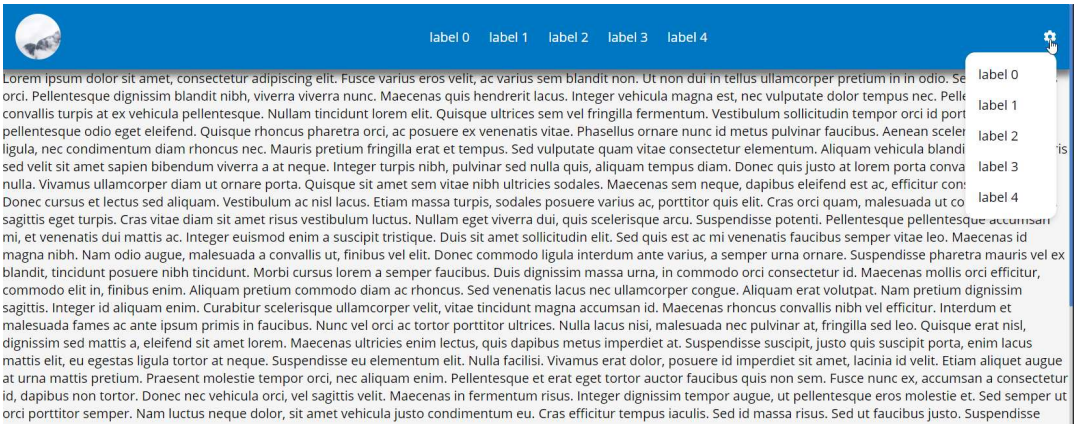
Sl. 3.75 Interakcija s oznakom navigacije

Na ekranima manjih veličina se umjesto oznaka navigacije prikazuje ikona (tri horizontalne linije u obliku hamburgera) vidljiva na slici 3.76 koja korisniku ukazuje kako je moguće klikom otvoriti listu s navigacijskim oznakama.



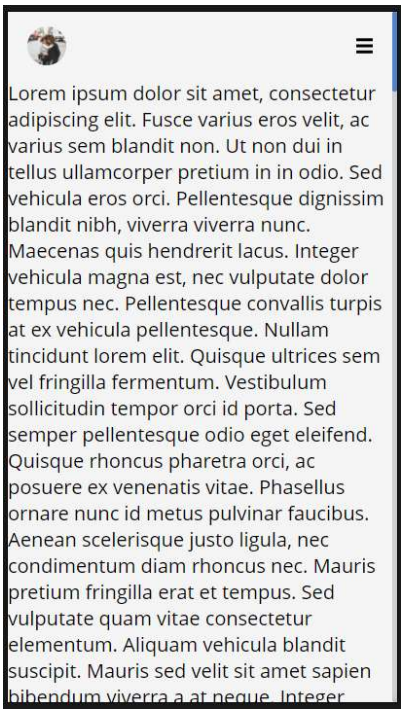
Sl. 3.76 Prilagodba mobilnim uređajima

Komponenta podržava listu postavki putem svojstva `menuItems` koje je po tipu jednako svojstvu `navItems`. U desnom rubu sučelja komponente je uvijek vidljiva ikona postavki čijim se klikom otvara lista oznaka postavki. Klikom na oznaku se okida događaj `onMenuItemClick` u čijem se svojstvu `detail` nalaze svojstva odgovarajućeg elementa. Funkcionalnost postavki je ostvarena komponentom tdtMenu opisane u poglavlju 3.3.7. kojoj je kao element dijete predana ikona postavki. Prikaz otvorenih postavki je vidljiv na slici 3.77.



Sl. 3.77 Prikaz postavki

Osim opisanog zaglavlja komponenta prikazuje i sučelje komponente tdtScrollToTop, što je vidljivo na slici 3.75, te koristi sve njezine funkcionalnosti opisane u poglavlju 3.2.11. Ukoliko se žele koristiti funkcionalnosti komponente potrebno je definirati istinitu vrijednost svojstva `useScrollToTop`. Grafičko sučelje za boju pozadine i navigacijskih oznaka uzima primarne vrijednosti iz definirane teme biblioteke. Postavljanjem svojstva `isFlat` na istinitu vrijednost pozadina postaje prozirna i zaglavlje se po z-osi poravnava sa razinom prozora web preglednika što je prikazano na slici 3.78.

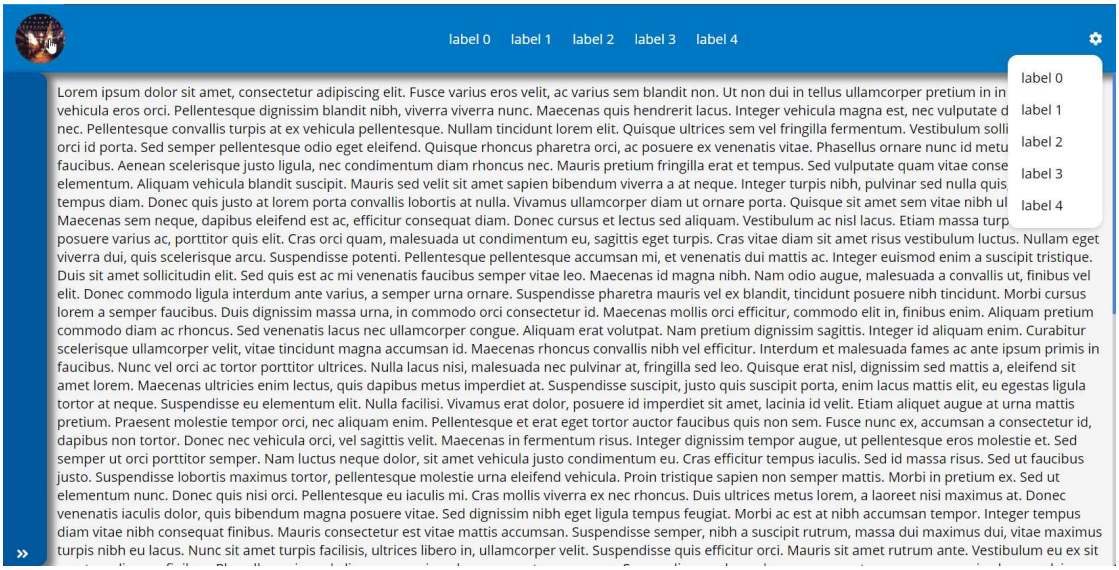


Sl. 3.78 Poravnanje s pozadinom po z-osi

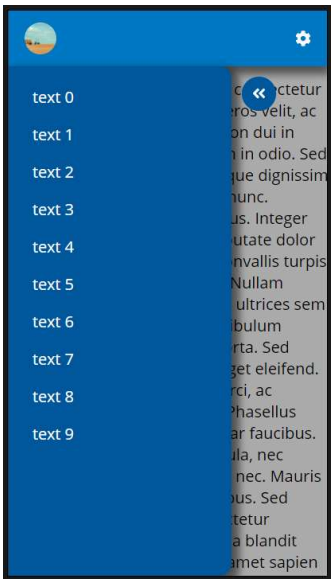
3.5. Predlošci

3.5.1. TdtLayout

Komponenta TdtLayout implementira raspored stranice. Grafičko sučelje se sastoji od navigacije u zaglavlju, kliznog izbornika i glavnog sadržaja. Komponenta je primjenjiva u mnogim aplikacijama raznih domena, jer je funkcionalnost izbornika i navigacije učestalo poželjna. Implementacija koristi komponente tdtNavbar, tdtSidebar i tdtScrollToTop. Skriva složenost suradnje i komunikacije nudeći jedinstveno sučelje te prilagođava grafičko sučelje ekranima raznih širina što je vidljivo na slikama 3.79 i 3.80.



Sl. 3.79 Desktop uređaj – otvorene postavke



Sl. 3.80 Mobilni uređaj – otvoren izbornik

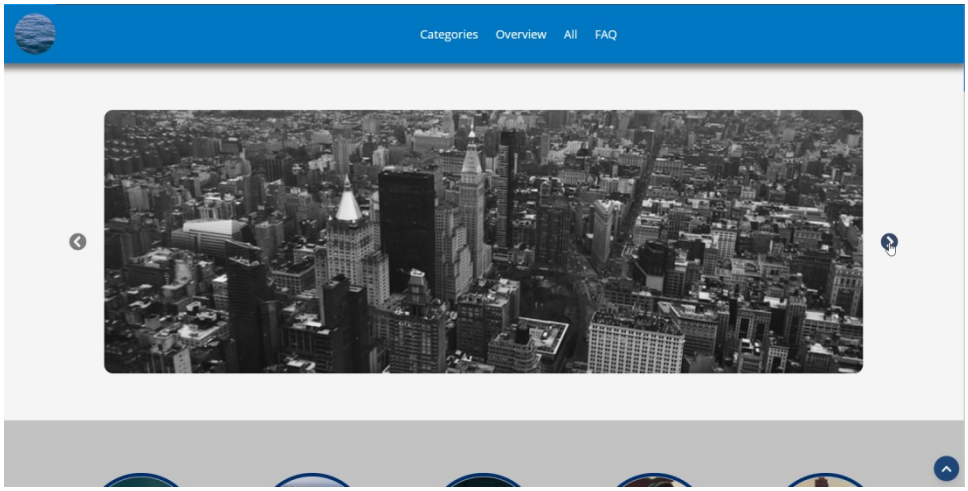
Navigacija se definira svojstvom `navbar` koje je po tipu jednako sučelju komponente tdtNavbar opisane u poglavlju 3.4.2., stoga komponenta nasljeđuje svu opisanu funkcionalnost. Klizni izbornik je definiran svojstvom `sidebar` koje je po tipu jednako sučelju komponente tdtSidebar opisane u poglavlju 3.3.9. zbog čega se također nasljeđuje sva opisana funkcionalnost. Glavni sadržaj rasporeda stranice prikazanog u grafičkom sučelju čine elementi djeca komponente.

2.5.2. TdtOverview

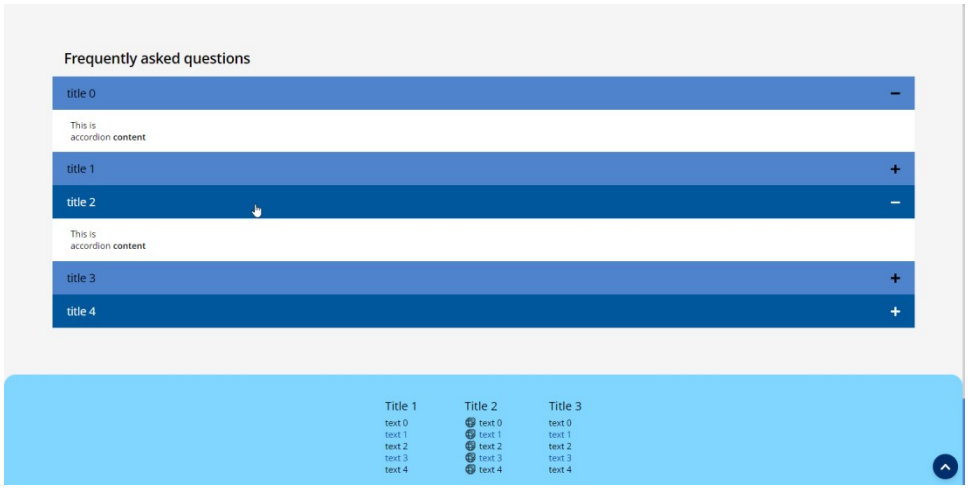
Komponenta TdtOverview implementira vrlo složenu funkcionalnost koja je učestalo korištena u raznim poslovnim domenama. Omogućuje prikaz sadržaja po kategorijama. Za svaku kategoriju se mogu prikazati dodatni sažetci ostvareni rešetkastim prikazom karticama sa slikom, naslovom i sadržajem podijeljenih prema aktivnim karticama. Aktivna kategorija prikazuje sažetak svih svojih stavki te odabir jedne od stavki za detaljniji prikaz. Moguće je definirati podnožje u kojemu se mogu stupčano prikazati kratke informacije. Komponenta podržava i prikaz liste naslova od kojih svaki može imati opis. Za ostvarivanje jedinstvenog sučelje koriste se komponente tdtTabs, tdtCardsGrid, tdtFooter, tdtDetailPanel, tdtNavbar, tdtLogoCloud, tdtAccordions i tdtSlides. Sva logika komunikacije i suradnje komponente je skrivena i odvija se bez potrebnih prilagodbi programera koji koristi komponentu. Grafičko

sučelje također nije potrebno prilagođavati jer komponenta raspoređuje sve korištene komponente prikladno prilagođavajući se raznim širinama ekrana kako bi se garantiralo optimalno korisničko iskustvo. S obzirom na svoju bogatu funkcionalnost komponenta je primjenjiva u mnoštvu domena koje prikazuju kategorizirani sadržaj. Primjerice, moguće je ostvariti prikaz županija Republike Hrvatske. Svaka županija predstavlja kategoriju. Korisnik klikom na županiju može pregledati općeniti sažeti sadržaj raspoređen po aktivnim karticama. Kartice bi mogle biti gradovi županije, demografski podaci o županiji i povijest županije. Za svaku karticu, kada je ona aktivna, korisnik vidi niz kartica podijeljenih po stupcima koje prikazuju informativnu sliku, naslov i sadržaj. Ako je primjerice aktivna kartica gradovi županije mogu se prikazivati samo najvažniji ili najveći gradovi u karticama sa smislenom slikom i kratkim opisom u kojemu pišu najvažnije informacije o gradu. Svaka aktivna kategorija, u našem primjeru županija, omogućuje prikaz velike liste elemenata koja bi mogla biti popis svih gradova određene županije. Korisnik vidi popis svih gradova koji je podijeljen po stranicama, klikom na grad se otvaraju detalji grada, slika 3.81. Lista naslova s opisima može predstavljati listu učestalo postavljenih pitanja i odgovora vezanih za Republiku Hrvatsku. U podnožju je moguće definirati kontakt podatke, poveznice na ostale stranice i administrativne informacije. Upravo smo opisali smislenu stranicu koju je moguće ostvariti samo uporabom jedne web komponente bez potrebe za ikakvom dodatnom prilagodbom ili interakcijom. Jedini je zahtjev definirati svojstva komponente čije se grafičko sučelje i funkcionalnost žele koristiti. S obzirom da je opisani idejni primjer vrlo općeniti te je format navedenih podatka primjenjiv nad velikim skupom domena iz svakodnevnog života možemo vidjeti kako je komponenta izuzetno primjenjiva jer se prilagođava potrebama raznih aktualnih domena. Svojstvom `siteLogo` se definira logo vidljiv u gornjem lijevom rubu prozora preglednika, vrijednost svojstva se definira kao vrijednost svojstva `logo` komponente tdtNavbar. Navigacijske oznake zaglavlja se definiraju svojstvom `sectionsLabels` koje implementira sučelje `IScrollTo`. Sučelju je potrebno definirati svojstvo `title` koje je tekstualna vrijednost prikazane oznake te svojstvo `tagName` koje odgovara oznaci komponente na koju se želi pozicionirati prozor preglednika. Primjerice, `tagName` vrijednost tdt-detail-panel će klikom na oznaku u zaglavlju kliznuti prozor preglednika do odjeljka u kojemu se prikazuje popis entiteta s prikazanim detaljima jednoga (slika 3.82), jer je upravo komponenta tdtDetailPanel zadužena za tu funkcionalnost, a njezina je oznaka (engl. *Tag*

name) tdt-detail-panel. Svojstva `slides`, `accordions` i `footer` odgovaraju sučeljima komponenti tdtSlides, tdtAccordions i tdtFooter opisanih u poglavljima 3.3.10., 3.3.1. i 3.3.5. zbog čega komponente ostvaruje sve funkcionalnosti opisane u navedenim poglavljima.

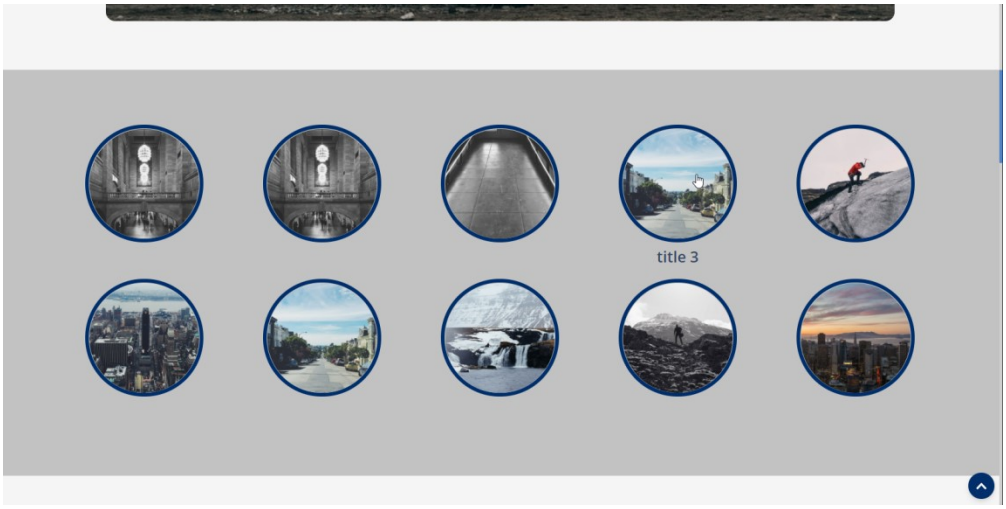


Sl. 3.81 Odjeljak listanja slika



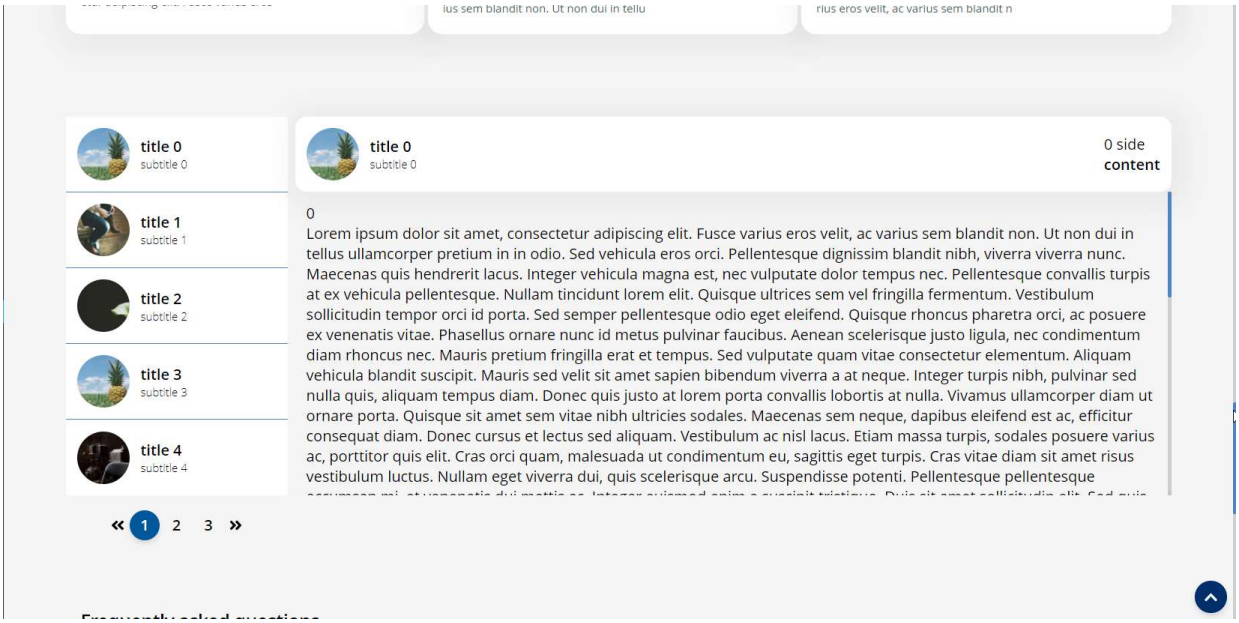
Sl. 3.82 Odjeljci liste naslova sa sadržajem i odjeljak podnožja

Najvažnije svojstvo komponente je svojstvo `categories` jer je uporabom definirane vrijednosti ostvarena sva komunikacija komponenti, funkcionalnost izmjene sadržaja ovisno o aktivnoj kategoriji i prikaz grafičkog sučelja s ispravnim vrijednostima. Svojstvo je lista sučelja `IOverviewCategory`. Sučelje sadrži svojstvo `logo` koje odgovara jednom elementu liste svojstva `logos` komponente tdtLogoCloud opisane u poglavlju 3.3.6.



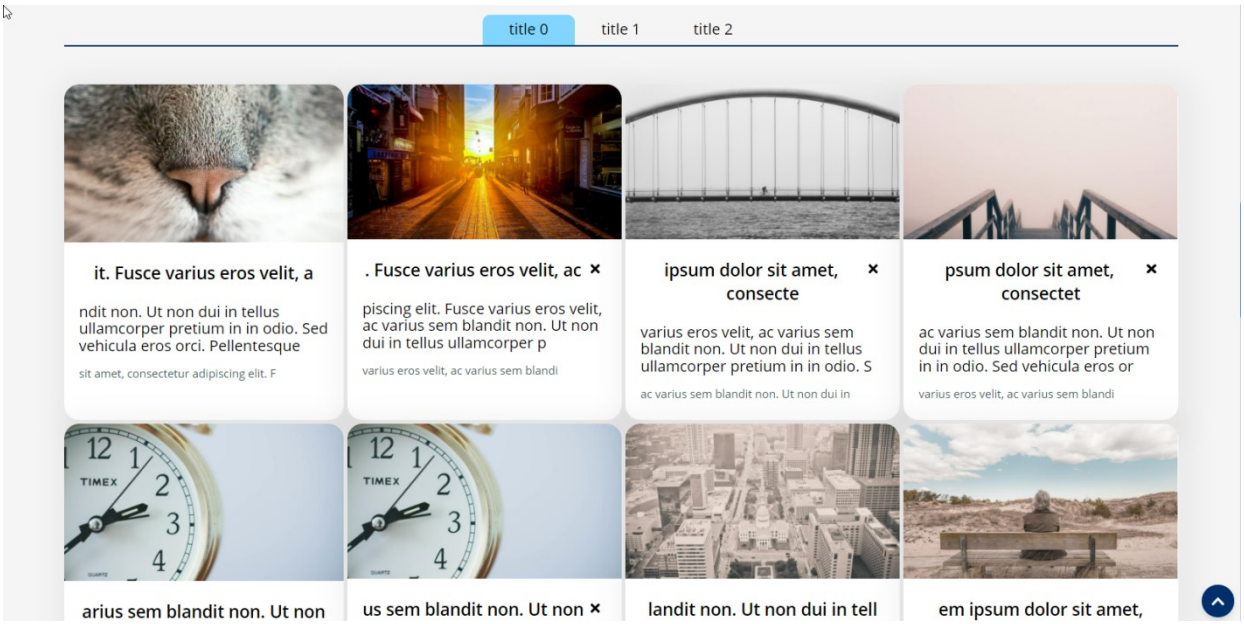
Sl. 3.83 Odjeljak kategorija

Klikom na jedan od loga sa slike 3.83, koji predstavljaju kategorije, ažurira se odjeljak liste svih stavki vidljiv na slici 3.84. Lista je definirana svojstvom `teasers` koje odgovara istoimenome svojstvu komponente tdtDetailPanel opisane u poglavlju 3.4.1 zahvaljujući kojoj se primjenjuje sva opisana funkcionalnost.



Sl. 3.84 Odjeljak popisa entiteta s prikazanim detaljima

Klikom na logo kategorije se ažuriraju i prikazane aktivne kartice sa stupčanim prikazom kartica sažetaka što je ostvareno svojstvom `tabs` koje implementira sučelje `IOverviewTab`. Ono omogućuje podjelu sadržaja po aktivnim karticama i rešetkasti prikaz kartica sažetaka za svaku aktivnu karticu. Svojstvo `tab` odgovara jednom elementu svojstva `tabs` komponente tdtTabs opisane u poglavlju 3.2.13. Komponente se kao element dijete definira komponenta tdtCardsGrid opisana u poglavlju 3.3.2. Prosljeđuje joj se vrijednost svojstva `cards` koje je zaduženo za prikaz listi kartica sažetaka koje su automatski raspoređene po stupcima, slika 3.85.



Sl. 3.85 Odjeljak aktivnih kartica sa rešetkastim prikazom kartica

3.5.3. TdtStaticTemplate

Komponenta TdtStaticTemplate je komponenta čije je grafičko sučelje vrlo konfigurabilnno, prilagodljivo i primjenjivo u mnoštvu domena. Komponenta prikazuje definirane komponente prilagođavajući njihov raspored raznim širinama ekrana. Namijenjena je generiranju statičkih stranica koje služe za pregled sadržaja. Grafičko sučelje garantira ugodno korisničko iskustvo smisleno prikazujući bogati tekstualni i slikoviti sadržaj. Najvažnije svojstvo komponente je svojstvo `rows` koje je po tipu lista. Pojedini elementi mogu biti različitih sučelja, no svaki mora implementirati sučelje jedne od komponenti tdtTextWithImage, tdtFeedbacks, tdtLogoCloud, tdtCardsGrid ili tdtAccordions. Za svaki element je moguće definirati svojstvo `class` koje se koristi kao referenca prilikom definiranja elemenata svojstva `sectionsLabels`. Ovisno o implementiranome sučelje komponenta koristi odgovarajuću komponentu prenoseći joj vrijednosti svih definiranih svojstava. Komponente se u grafičkom sučelje prikazuju u retcima čiji poredak odgovara poretku elemenata svojstva `rows`. Svojstva `siteLogo`, `footer` i `sectionsLabels` su jednaka svojstvima komponente tdtOverview opisane u prethodnom poglavlju. Jedina je razlika u tome što elementi liste definirane svojstvom `sectionsLabels` umjesto svojstva `tagName` trebaju imati definirano svojstvo

`class` koje ima jednaku vrijednost svojstvu `class` jednom od elemenata svojstva `rows`. Klikom na grafičko sučelje jednog od elementa liste `sectionsLabels` će prozor web preglednika kliznuti do grafičkog sučelja odgovarajućeg elemenata liste `rows` čije svojstvo `class` odgovara istoimenom svojstvu kliknutog elementa. Komponenta se može upotrijebiti primjerice za brzu izradu sadržaja web stranice poput oglašavanja usluga obrta bilo koje struke. U slijedećem poglavlju je opisano kako se komponenta može primijeniti za izradu web stranice knjigovodstvenog obrta te su prikazane slike grafičkog sučelja.

3.6. Stranice

3.6.1. Knjigovodstveni obrt

Uporabom komponente tdtStaticTemplate ostvarena je web stranica Knjigovodstvenog obrta Kamber. Omogućuje pregled usluga, iskustva klijenata servisa, učestalo postavljenih pitanja i svih potrebnih informacija usluga kako bi osoba mogla stupiti u kontakt s djelatnicima s ciljem poslovne suradnje. Projekt web stranice je implementiran pomoću jednostavnih tradicionalnih web tehnologija. Poslužitelj koji pokreće web stranicu je implementiran programskim jezikom JavaScript u tehnologiji nodeJS. Koristi se funkcija listen(port, callback) paketa express koja prema vrijednosti prvog argumenta pokreće aplikaciju unutar web preglednika na definiranom portu. Prilikom inicijalnog zahtjeva aplikacija se preusmjeruje na rutu `/home`. Odgovor je statička HTML-stranica izuzetno jednostavnoga sadržaja, slika 3.86.

```
<html lang="en">
▶<head>…</head>
▼<body data-new-gr-c-s-check-loaded="14.1047.0" data-gr-ext-installed> grid
  ▶<tdt-static-template _nghost-xln-c44 ng-version="12.2.4">…</tdt-static-template> == $0
    <script src="scripts/home.js" type="module"></script>
  </body>
</html>
```

Sl. 3.86 HTML-predložak web stranice knjigovodstvenog servisa

Osim nužnih HTML-oznaka `html`, `head` i `body`, stranica sadrži oznaku `tdt-static-template` koja predstavlja komponentu opisanu u prethodnom poglavlju. Biblioteka web komponenti TDT instalirana je prema poglavlju 3.1. uporabom poslužitelja UNPKG. Sadržaj korisničkog sučelja stranice je statički te je trivijalno ostvaren uporabom jednostavne

JavaScriptove skripte `home.js` čija je referenca vidljiva na slici 3.86. Skripta dohvaća referencu na komponentu TdtStaticTemplate i definira joj potrebna svojstva, kod 3.3.

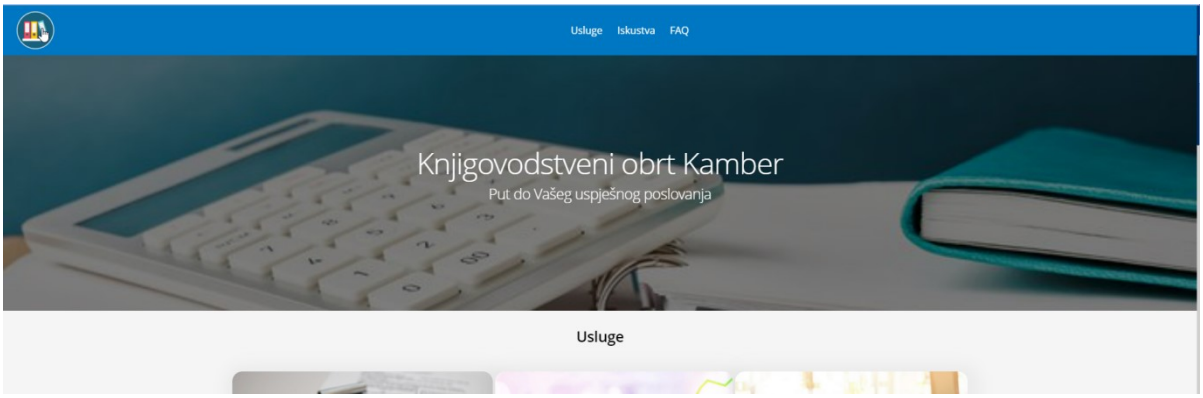
```
// Populate component properties.
document.addEventListener("DOMContentLoaded", function () {
  const template = document.querySelector("tdt-static-template")

  template.siteLogo = accountsData.siteLogo
  template.sectionsLabels = accountsData.sectionsLabels
  template.footer = accountsData.footer
  template.rows = [
    accountsData.imageWithBackground,
    accountsData.servicesRow,
    accountsData.imageWithTextRight,
    accountsData.feedbacksRow,
    accountsData.imageWithTextLeft,
    accountsData.imageWithTextBottom,
    accountsData.faqRow,
  ]
})
```

Kod 3.3. Definiranje svojstava komponente TdtStaticTemplate

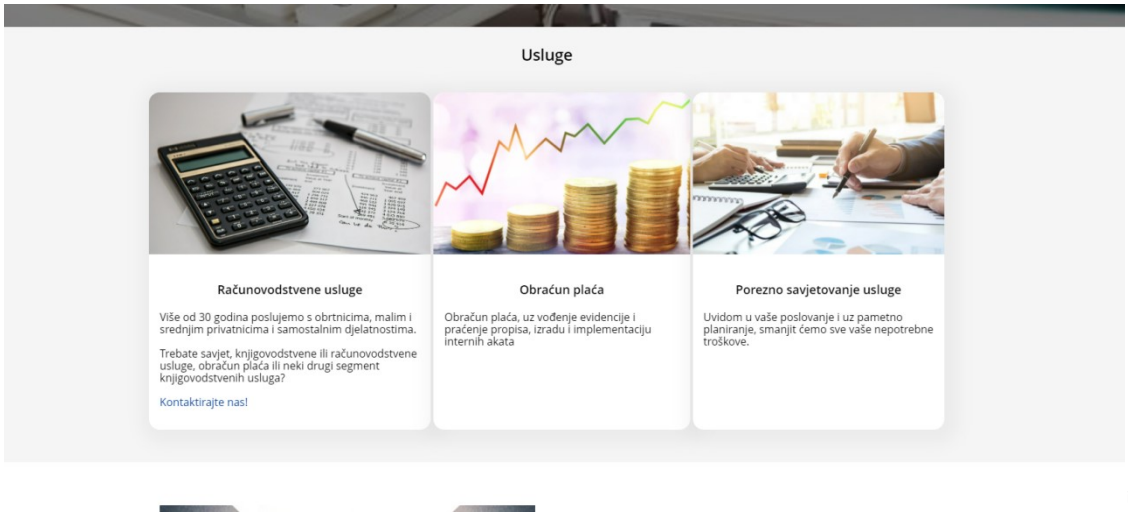
Svojstvom `siteLogo` definira se logo stranice, `sectionsLabels` su oznake u zaglavlju stranice, a `footer` predstavlja podnožje stranice s definiranim kontakt-informacijama. Kako je opisano u prethodnom poglavlju svojstvom `rows` se definirao sav sadržaj stranice koji se prema vizualnom sučelju može podijeliti u retke.

Na slici 3.87 vidimo prvi redak, komponentu tdtTextWithImage kojoj je vrijednost svojstva `variant` postavljena na '`background`'. Prikazuje informativni naslov i podnaslov s zatamnjenom slikom knjigovodstvenog obrta u pozadini. Na slici je prikazano i zaglavlje s oznakama koje klikom vode do odgovarajućeg odjeljka stranice.



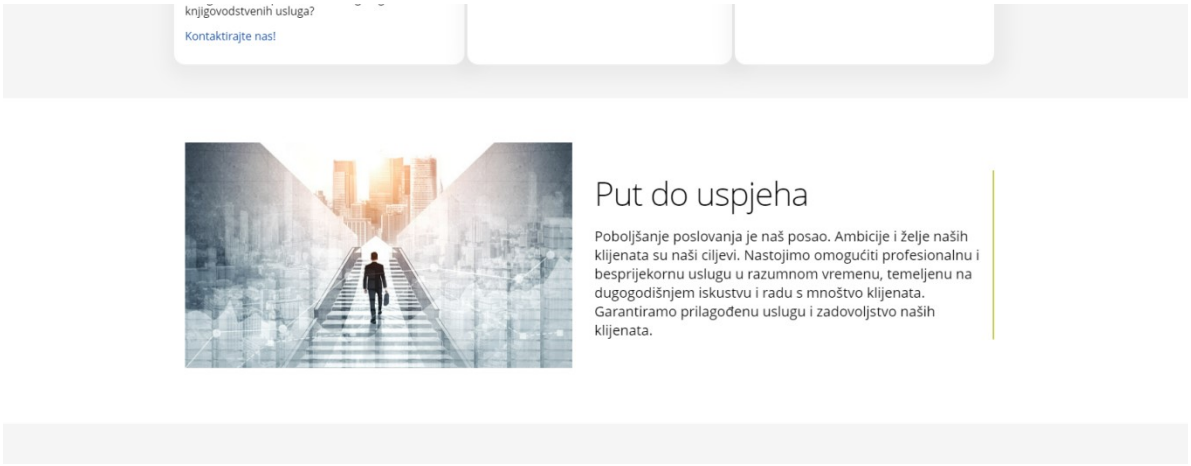
Sl. 3.87 Komponenta tdtTextWithImage - background varijanta

Sljedeći redak je komponenta tdtCardsGrid pomoću koje se prikazuju usluge knjigovodstvenog servisa, slika 3.88.



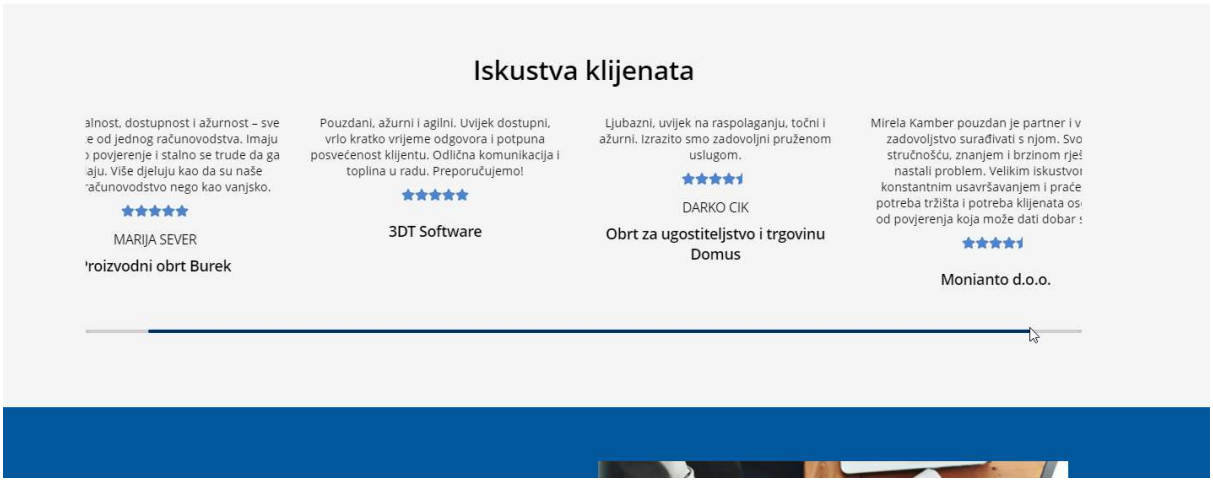
Sl. 3.88 Komponenta tdtCardsGrid – usluge servisa

Nakon retka usluga dolazi tekstualni sadržaj, naslov i opis, uz koji s lijeve strane stoji slika. Redak je komponenta tdtTextWithImage kojoj je vrijednost svojstva `variant` postavljena na '`left`', slika 3.89.



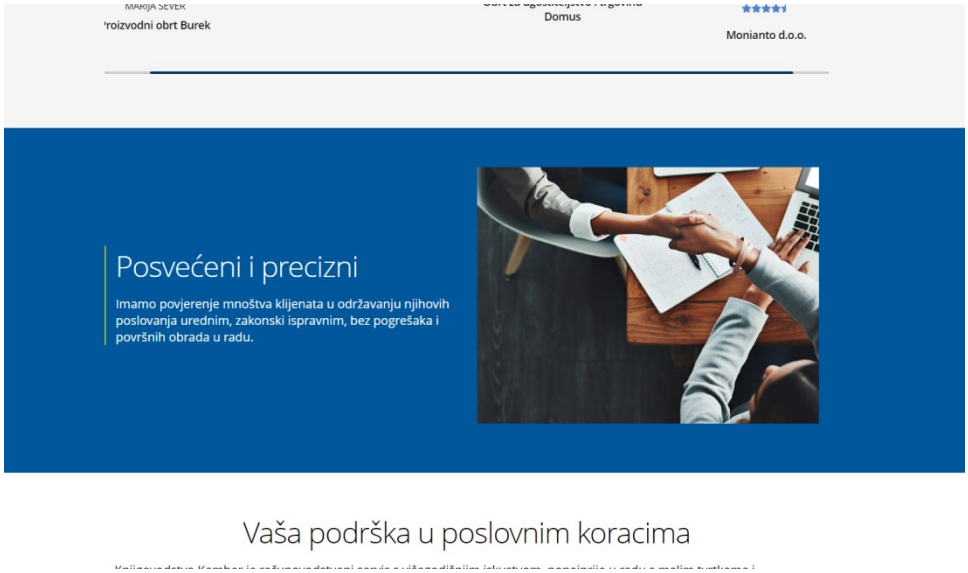
Sl. 3.89 Komponenta tdtTextWithImage –varijanta `left`

Slijedi redak koji prikazuje iskustva dosadašnjih klijenata knjigovodstvenog servisa, ostvaren komponentom tdtFeedbacks. Redak prikazuje naziv kompanije klijenta, opcionalno ime i prezime osobe, komentar i ocjenu, slika 3.90.



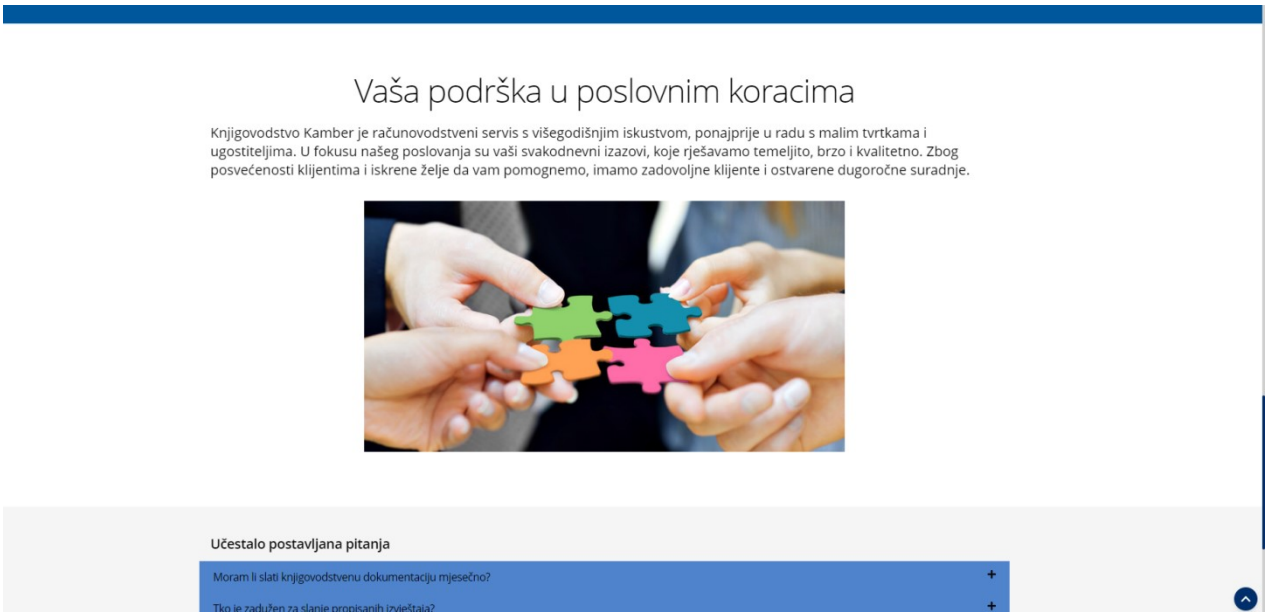
Sl. 3.90 Komponenta tdtFeedbacks – iskustva klijenata

Nakon iskustva klijenata slijedi redak koji ponovno koristi komponentu tdtTextWithImage kako bi se prikazao naslov, opis i slika. Komponenti je vrijednost svojstva `variant` postavljena na '`right`', slika 3.91.



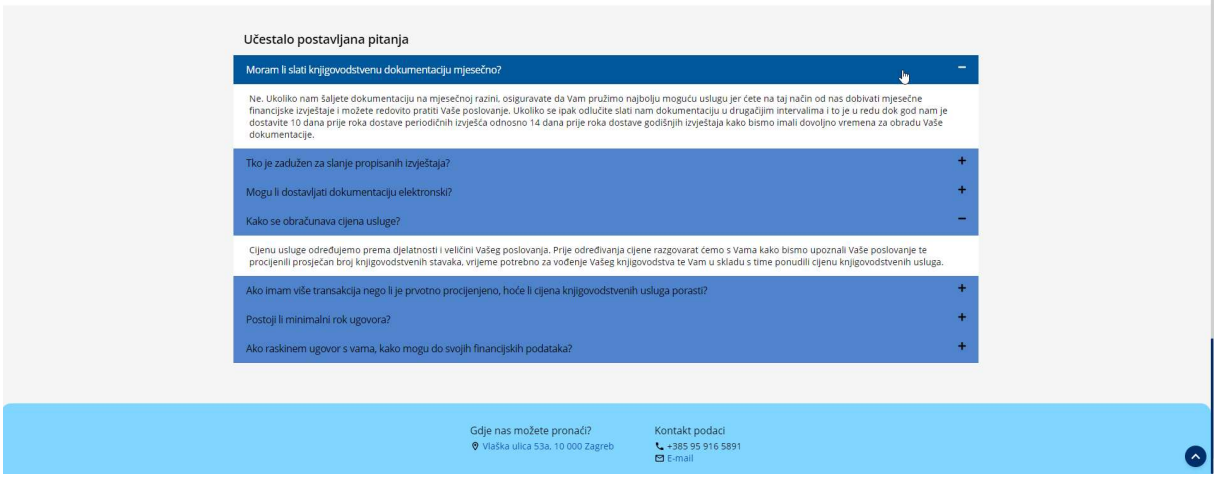
Sl. 3.91 Komponenta tdtTextWithImage –varijanta `right`

Sljedeći redak je ostvaren varijantom '`bottom`' komponente tdtTextWithImage, slika 3.92.



Sl. 3.92 Komponenta tdtTextWithImage –varijanta `bottom`

Posljednji redak predstavlja odjeljak za učestalo postavljena pitanja kako bi se budućim klijentima usluge pomoglo bez da nužno kontaktiraju djelatnike. Odjeljak je ostvaren komponentom tdtAccordions. Na slici 3.93 uz učestalo postavljena pitanja vidimo i podnožje stranice.



Sl. 3.93 Komponenta tdtAccordions – učestalo postavljena pitanja i podnožje

4. Mogućnost budućeg rada i vrijednost biblioteke

Biblioteka web komponenti koja je izrađena u sklopu diplomskog rada pruža skup funkcionalnih komponenti koje se mogu koristiti u bilo kojoj aplikaciji čije tehnologije podržavaju HTML i JavaScript. Kako bi se prikazala jednostavnost, smislenost, grafičko sučelje i ispravnost funkcionalnosti komponenti izrađena je web stranica knjigovodstvenog obrta uporabom svega jedne komponente kojoj je predan statički sadržaj. Kako je biblioteka javno dostupna i besplatna za preuzimanje smatram da je diplomskih radom ostvarena vrijednost za zajednicu otvorenoga koda.

Iako biblioteka pokriva širok skup funkcionalnosti moguća su brojna dodatna proširenja. Za razvoj biblioteke komponenti je u praksi potreban dugogodišnji rad tima razvojnih inženjera. Iza najpopularnijih biblioteka komponenti specifičnih tehnologija stoje kompanije sa snažnom reputacijom poput Googlea i Facebooka. Neke od popularnijih biblioteka web komponenti su također razvijene od strane većih organizacija poput HTML Smart Elements [6] i Polymera [7].

Sljedeći korak u proširenju bi mogao biti dorada postojećih komponenti. Proširenje funkcionalisti dodavanjem novih svojstava sučeljima kako bi komponente postale fleksibilnije i primjenjivije u više slučajeva uporabe.

Biblioteka bi bila potpunija kada bi sadržavala više atoma opće uporabe. Prvo veliko proširenje bi bilo dodati atome poput liste s automatskim nadopunjavanjem, komponentu za oblikovanje teksta i fiksirane obavijesti. Mogle bi se dodati molekule poput formi, prozora dijaloga i grupe gumbi. Naravno, kako bi se vrijeme rada na projektu povećavalo biblioteka bi mogla pokrivati sve više funkcionalnosti što bi zasigurno doprinijelo broju tjednih preuzimanja putem registracijskog tijela NPM. Vrlo bi korisno bilo podržati više varijanti boja koje se koriste u biblioteci te dodati mogućnost konfiguriranja kako bi se promijenile osnovne boje. Smatram kako bi kvaliteti biblioteke značajno doprinijelo unajmljivanje stručnih dizajnera koji bi oblikovali sučelja komponenti prema optimalnim i trenutno popularnim principima grafičkog i digitalnog oblikovanja s ciljem poboljšanja korisničkog iskustva.

5. Zaključak

Nagli porast web tehnologija za razvoj programske podrške zahtijeva konstantno učenje i praćenje aktualnosti. Postoji mnoštvo tehnologija za razvoj prezentacijskog sloja od kojih svaka pojedina ima svoje vrline i mane koje treba uzeti u obzir prilikom odabira tehnologije za razvoj. Najveća mana je činjenica da tehnologije nisu međusobno kompatabilne. Prilikom implementacije gotovo bilo kojeg sustava postoji niz komponenti koje je potrebno uvijek ponovno implementirati ukoliko programer promijeni tehnologiju razvoja. Diplomskim radom sam htio implementirati komponente koje obavljaju učestale funkcionalnosti, imaju uređeno grafičko sučelje i mogu se koristiti neovisno o tehnologiji implementacije. Cilj je ostvaren razvojem biblioteke web komponenti. Web komponente su tehnologija koja rješava mnogo problema. Kompatabilne su s ostatkom tehnologija za razvoj web programske podrške. Enkapsuliraju učestale funkcionalnosti, smanjuju spregu programskog koda jer imaju jasno definirano sučelje. Komponente koje su molekule i organizmi prema atomarnom oblikovanju povećavaju koheziju jer same za sebe predstavljaju modul koji obavlja niz smisleno povezanih funkcionalnosti. Najveća korist web komponenti je ponovna uporaba čime raste njihova vrijednost, jer jednom naučen API biblioteke se može primijeniti čak i ukoliko se programer odluči za razvoj u drugoj tehnologiji, što smanjuje vrijeme prilagodbe novom razvojnom okruženju.

Atomarno oblikovanje se pokazalo kao vrlo korisno i smisleno načelo koje garantira primjenu preporučenih načela koje su vrlo jasno definirane u objektnom oblikovanju, no nisu toliko jasne u razvoju prezentacijskog sloja. Atomi osiguravaju apstrakciju složenih koncepata. Molekule implementiraju preporuku da programski kod treba obavljati samo jednu funkcionalnost. Organizmi implementiraju jedinstveno sučelje, omogućuju ponovnu uporabu, enkapsulaciju, malu spregu i veliku koheziju. Predlošci smanjuju apstrakciju i razumljiviji su krajnjim korisnicima, jer raspoređuju niže razine smisleno i prikladno u grafičkom sučelju. Stranice daju primjer uporabe nižih razina sa semantički smislenim podacima.

Tehnologija Angular elements je odličan izbor za razvoj web komponenti jer obavlja prevođenje Angularovih komponenti u web komponente bez mnogo konfiguracije. Skriva složenost i novitete tehnologije web komponenti iza sličnih koncepata u Angularovoj

tehnologiji. Programer se može fokusirati na smislenost, implementaciju i domenu razvoja komponenti ne trošeći resurse na izazove tehnologije web komponenti.

Diplomskim radom sam pokazao kako je odabir tehnologije Angular elements odličan izbor za razvoj biblioteke web komponenti. Atomarno oblikovanje garantira da je biblioteka jednostavna za održavanje i proširivanje. S obzirom na brzi porast raznih web tehnologija i paradigmi programiranja smatram da bi se više razvojnih timova trebalo orijentirati na budućnost, održivost, ponovnu uporabu i jednostavnost testiranja programskog koda prezentacijskog sloja programske potpore. Također smatram kako će u bliskoj budućnosti porasti interes za web komponentama te će se razvoj sve više orijentirati na implementaciju biblioteka komponenti.

Sažetak

Naslov: Razvoj biblioteke web komponenti koristeći metodu atomarnog oblikovanja

Zadatak diplomskog rada bio je razviti biblioteku web komponenti. Razvoj i oblikovanje biblioteke su se temeljili na atomarnom oblikovanju. Korisnik biblioteke preuzimanjem putem NPM platforme može koristiti mnoštvo jednostavnih i složenih komponenti široke primjenjivosti. Moguće je koristiti organizme biblioteke koji implementiraju učestalo korištene funkcionalnosti i time znatno olakšavaju razvoj web aplikacija. U sklopu diplomskog rada se razvila web stranica knjigovodstvenog servisa uporabom svega jedne web komponente biblioteke kako bi se pokazala primjenjivost, jednostavnost i efikasnost razvijene biblioteke.

Ključne riječi: web komponente, biblioteka komponenti, web aplikacija.

Summary

Title: Development of a web component library using the atomic design method

The task of the master's thesis was to develop a library of web components. The development and design of the library were based on atomic design. By downloading the library via the NPM platform, the library user can use many simple and complex components of broad applicability. Furthermore, it is possible to use library organisms that implement commonly used functionalities and thus significantly facilitate the development of web applications. As part of the master's thesis, an accounting website was developed using a single web component of the library to demonstrate the developed library's applicability, simplicity, and efficiency.

Keywords: web components, component library, web application.

Literatura

[1] Webcomponents.org, Web components. Poveznica: https://www.webcomponents.org/; pristupljeno: 06.03.2022.

[2] Rob Dodson, Custom Elements Everywhere. Poveznica: https://custom-elements-everywhere.com/; pristupljeno: 06.03.2022.

[3] MDN, Custom elements. Poveznica: https://html.spec.whatwg.org/multipage/custom-elements.html#custom-elements, pristupljeno: 06.03.2022.

[4] Angular, Angular elements overview. Poveznica: https://angular.io/guide/elements; pristupljeno: 06.03.2022.

[5] Brad Frost, Atomic design methodology. Poveznica: https://atomicdesign.bradfrost.com/chapter-2/; pristupljeno: 06.03.2022.

[6] Storybook, Introduction to Storybook for React. Poveznica: https://storybook.js.org/docs/react/get-started/introduction; pristupljeno: 06.03.2022.

Skraćenice

- HTML Hypertext Markup Language
- DOM Document Object Model
- API Application Programming Interface
- NPM Node Package Manager
- CDN Content Delivery Networks