Dizajn poslovnih korisničkih sučelja

Marinović, Matija

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:483788

Rights / Prava: Attribution 3.0 Unported/Imenovanje 3.0

Download date / Datum preuzimanja: **2024-03-29**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Matija Marinović

DIZAJN POSLOVNIH KORISNIČKIH SUČELJA

ZAVRŠNI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Matija Marinović

Matični broj: 0016121065

Studij: Primjena informacijske tehnologije u poslovanju

DIZAJN POSLOVNIH KORISNIČKIH SUČELJA

ZAVRŠNI RAD

Mentor/Mentorica:

Doc. dr. sc. Konecki Mario

Suocjenjivač/ica:

Prof. dr. sc. Lovrenčić Alen

Varaždin, ožujak 2020.

Matija Marinović

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Ovaj će rad opisati poželjne primjere dizajna u današnjem poslovnom sustavu. Suvremeni poslovni sustavi zahtijevaju stabilan rad, lako korištenje i optimizaciju korisničkog sučelja kao nikada prije. Podržava se jednostavnost i grupiranje elemenata korisničkog sučelja. Dobar dizajn, ovisno o svrsi, možemo sagledati sa mnogo aspekata.

Ovisno o publici i klijentima koji koriste određeni sustav, sustav im treba odgovarati. Korištenje najboljih praksa(„best practices") suvremenog dizajna u današnjem web i mobilnom razvoju mogu podići imidž tvrtke ili njenog proizvoda za značajnu vrijednost. Vizualni identitet bitan je dio dobrog i pomno promišljenog dizajna. Može doprinijeti kao opis osobina ili ciljeva kojemu neko poduzeće naginje. Jedinstvenim i primjerenim identitetom (ovisno o subjektu za kojega radimo vizualni identitet), podižemo kvalitetu samog poduzeća, proizvoda ili aplikacije. Možemo i sagledati dijametralno suprotan slučaj, gdje sami poduzetnici olako shvaćaju važnost vizualnog doživljaja njihovog proizvoda, web stranice ili aplikacije. Zbog niza komplikacija i ne logičnosti unutar samog grafičkog sučelja, klijenti ne nalaze potrebu za korištenjem sustava koji im se nudi. Takvi propusti su u današnjem poduzetničkom svijetu ne prihvatljivi. Kako bi korisnik bio zadovoljan, tržište se mora pratiti kao i njegove trendove unutar samoga tržišta.

Cilj ovoga rada je dati pregled dizajnerskih principa i tehnika koje se koriste prilikom izrade poslovnih korisničkih sučelja. U prvome dijelu rada, dotaknuti ću se teorijskog dijela dizajniranja i osmišljavanja samog vizualnog identiteta, dok ću se u drugome dijelu fokusirati na praktični dio. On će sadržavati analizu postojeće aplikacije te dati nove poglede za modernija rješenja. Opisati ću i karakteristike i dizajnerske elemente te ostale specifičnosti važne za ovu vrstu dizajna.

Ključne riječi: web aplikacija, mobilna aplikacija, vizualni identitet, grafički dizajn, dizajn korisničkog sučelja, dizajn korisničkog iskustva, brand, najbolje prakse

Sadržaj

1. Uvod 1
2. Metode i tehnike rada 2
3. Osnove dizajna korisničkog sučelja (UI) 3
4. Osnove dizana korisničkog iskustva 4
5. Proces i arhitektura izrade korisničkog sučelja 6
5.1. Ideja 6
5.2. Wireframe 7
5.3. Analiza i prikaz kretanja po sučelju 8
6. Tehnologije i biblioteke 15
6.1. HTML 15
6.2. React i JavaScript 16
6.3. CSS i SCSS 18
7. Izrada prototipa 21
7.1. Logotip 21
7.2. Boje 22
7.3. Tipografija 23
7.4. Prilagođavanje pozicioniranja elemenata 24
7.5. Animacije 26
8. Sveobuhvatni primjer WWWUI-a 27
9. Zaključak 30
Popis literature 31
Popis slika 33

1. Uvod

U ovome radu govoriti ću najviše o dizajnu web aplikacija. Današnji web je dostigao veliku dozu fleksibilnosti implementirajući razna programska rješenja za programiranje grafičkog korisničkog sučelja. Pomoću raznih modernih razvojih okvira(React, React-Native, Angular) za implementiranje korisničkog sučelja, imamo mogućnost fleksibilnog razvoja aplikacije za različite platforme. Možemo posjetiti web stranicu na mobilnom uređaju, tablet računalu te na mnogo drugih platformi, gdje je cilj da sadržaj same stranice ostane ne promijenjen. Dizajn same stranice može korisnika zbuniti ili odbiti ovisno o implementaciji. Govoreći da je u današnjem razvoju aplikacija općenito veoma bitno zadržati izgled stranice (komponente, navigacije, funkcionalnosti) identično na svim platformama stvara dizajn aplikacije mnogo kompleksnijim.

Aplikacije poslovnoga tipa zahtijevaju veliku dozu stabilnosti i ozbiljnoga pristupa. Pri razvijanju poslovnih korisničkih sučelja treba znati poslovnu domenu klijenta. Korisničko sučelje je kreirano ovisno o analizi potreba i zahtjeva klijenta. Današnji poslovni korisnički sustavi zahtijevaju mobilnost i sposobnost za prilagodbu. Razvijanje aplikacija podrazumijeva konstantna ažuriranja te promjene. Svaka promjena korisničkog sučelja u poslovnome svijetu zahtjeva inicijalni trud korisnika kako bi iznova naučio koristiti aplikaciju.

Današnji dizajn mora biti dinamičan i jasan. Dinamičnost samih aplikacija ovisi uvelike o tehnologiji u kojoj je samo sučelje rađeno. Skriptni jezici dominiraju tržištem kada se priča o razvijanju frontend-a. Razvijanje korisničkog sučelja i njegov client-side razvoj relativno je nepoznata internet disciplina. Mnogo ljudi shvatit će razvoj korisničkog sučelja olako i misliti da samo estetika ima glavnu ulogu (vrlo je važna), no postoji niz tehnologija koje razvojni inženjeri moraju znati (HTML, CSS, Cross-browser,…). O tehnologijama ću više govoriti kasnije.

2. Metode i tehnike rada

Za pisanje ovog završnog rada korištena je stručna literatura dostupna na internetu, koja predstavlja sekundarne izvore podataka tj. podatke koji postoje iz raznih drugih izvora. Tijekom izrade teorijskog i praktičnog dijela rada korištene su razne metode znanstveno-istraživačkog rada. U daljnjem tekstu napisat ćemo koje su se to sve metode koristile tijekom pisanja ovog rada, te ukratko objasniti svaku od korištenih metoda:

- Induktivno – deduktivna metoda je metoda koja uključuje zaključivanje iz općih do konkretnih zaključaka na temelju činjenica.
- Metoda analize i sinteze je metoda koja uključuje raščlanjivanje složenih pojmova, sudova i zaključaka na jednostavnije sastavne dijelove te izučavanje svakog dijela za sebe, a predstavlja i spajanje elemenata u cjelinu.
- Metoda apstrakcije i konkretizacije je metoda koja se koristila kod odvajanja nebitnih i isticanja bitnih pojmova.
- Metoda deskripcije je metoda postupka jednostavnog opisa koja se koristi kod opisivanja pojmova. To je metoda u kojoj istraživač promatra trenutno stanje određene stvari koju opisuje , bez promjene neke od varijabli.
- Metoda komparacije je metoda kojom se proučavaju i uspoređuju sličnost i razlike između dva pojmova, te njihova zajednička obilježja sa ciljem da se dođe do određenih zaključaka.

3. Osnove dizajna korisničkog sučelja (UI)

Moderno doba u kojem je korištenje aplikacija i IT servisa korišteno više nego ikada prije, zahtjevi za dobro modeliranim korisničkim sučeljem svakodnevna su pojava. Dizajn korisničkog sučelja komentira se, izučava i donosi mnoge izazove u praćenju koraka sa konstantnim novim rješenjima. Korisnost sustava popravila se vremenom zbog mnogih testiranja različitih rješenja i veće posvećenosti detaljima. Nepravilnost i složene strukture dizajna korisničkog sučelja ne toleriraju se kao prije i mogu donijeti ogroman poslovni rizik. Primjer dobrog dizajna, kada je primijećen, jasno pokazuje kako dobar dizajn može zadržati klijente ili korisnike sustava uključenima. U dvadesetome stoljeću, velik broj različitih prikaza sučelja i interakcijskih tehnika nazivamo grafičkim korisničkim sučeljem. Izgled stranice, arhitektura grafičkog dizajna i navigacija utječu na korisnikovu prisutnost u više smjerova. Ako su stranice zbunjujuće i ne efikasne, korisnik će teže moći raditi svoj posao i više će griješiti. Loš dizajn može također otjerati korisnike sustava na duže vrijeme ili u potpunosti. Frustracija ili stres sigurno nisu težnje poslovnog sustava pa se korisnički sustav mora unaprijed dobro definirati i testirati. Zbunjujući dizajn kod sustava javne domene (vojni, nuklearni, zdravstveni...) i opće društvene sigurnosti može dovesti do incidenata velikih razmjera. Postoje studije o interakciji čovjeka i računala (engl. Human-computer interaction) u koju su uključene mnoge znanstvene discipline.[1]

Kod izrade grafičkog korisničkog sučelja bitno je razmišljati kao korisnik istoga, te uključiti ra zne profile korisnika u procijenu sučelja u svrhu njegovog poboljšanja. Kao korisnik nemamo pot rebu znati u kojem se programskom jeziku pisao ili koja je tehnologija zaslužna za njegovo funk cijoniranje.Ono što je potrebno korisniku je da razumije kontrole sustava i njegove postavke. Postavke trebaju biti intuitivne i lake za razumjeti. Bitan je cilj korištenja sustava i znati o kakvom profilu korisnika se radi. Korisnici kada nisu zadovoljni grafičkim sučeljem, često znaju govoriti kako je problem „u računalu", što u većini slučajeva nije točno i dovodi do krivih zaključaka. Loša implementacija grafičkog sučelja može uzrokovati probleme i neželjene akcije iako ne mora biti dokaza za takve greške. Takvi sustavi mogu dovesti do velikog rizika za samog poslodavca odnosno vlasnika takvog sustava.[2]

---

[1] Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. (2005). *User interface design and evaluation*. Elsevier. preuzeto 21.09.2020.

[2] Jovanovic J. (2010). User Interfaces in Business Web Application Design Preuzeto 21.09.2020. s https://www.smashingmagazine.com/2010/02/designing-user-interfaces-for-business-web-applications

4. Osnove dizana korisničkog iskustva

Korisničko iskustvo veže svakog posjetioca ili korisnika poslovnog sustava. Dobro izveden UX(engl. *User expirience*) može dovesti do veće potražnje. Što je korisničko iskustvo na sustavu bolje, sustav koji koristimo u velikoj je prednostu u odnosu na konkurentni sustav. Korisnici se vraćaju, ponovno ga koriste i konkurentna rješenja im ne nude jednako dobro korisničko iskustvo. Korisnost dobrog korisničkog iskustva:

- Više kupaca ostaje na platformi, odnosno sustavu koje pružaju bolje korisničko iskustvo. U prosjeku tvtke koje pružaju bolji sadržaj, sveukupno korisničko iskustvo imaju 14,4% više kupaca koji će se vratiti na patformu, odnosno sustav, te će razmotriti drugu kupnju ili povratak na samo web mjesto.
- Više korisnika oduprijet će se poslovanju sa konkurentskim sustavima. U usporedbi s tvrtkama koje nude loše korisničko iskustvo, tvrtke sa dobrim korisničkim iskustvom zadržati će 15,8% klijenata koji će razmotriti poslovanje s konkurentom.
- Preporuke su daleko najbolji i najjeftiniji oblik marketinga. Više kupaca preporičit će tvrtku sa najvišim ocjenama korisničkog iskustva. Čak 16,6% kupaca, odnosno klijenata preporučiti će proizvode ili usluge u odnosu na niže rangirane sustave.[3]

Forrester, tvrtka koja pomaže organizacijama da shvate klijentove želje, stavljaju ga u centar pozornosti. Istraživanjem provedenim na desecima vodećih stručnjaka za dizajn, uporabljivost i etnografsko istraživanje, uključujući akademike i radnike u praksi, otkrili su da četiri najbolje prakse kod stvaranja korisničkog iskustva:

- Postankom korisnika vlastitog sustava dobiva se uvid kako same dizajnirati.
- Prvo se treba pozabaviti dizajnom kako bi se izbjeglo narušavanje korisničkog iskustva, odnosno prepuštanje istoga slučaju.
- Detaljnim testiranjem sustava potvrđujemo da zaista nudimo korisnicima najbolje moguće korisničko iskustvo.
- Dizajn korisničkog iskustva treba biti dio životnog ciklusa razvoja software-a. (engl. *Software Development Life Cycle*)

[3] Gualtieri, M. (2009). Best practices in user experience (UX) design. Design Compelling User Experiences to Wow your Customers, 1-17. preuzeto 24.09.2020

Osim navedenih točaka, Forresterovo korisničko iskustvo u dizajniranju sljedećih praksi na koje bi se stručnjaci za razvoj aplikacija trebali usredotočiti nakon što savladaju osnove:

- Koristite imerzivnu metodologiju razvoja.

Baš kao što su to profesionalci za razvoj aplikacija stručnjaci za programske jezike, baze podataka, okvire i platforme - oni to također moraju biti stručnjaci za dizajn korisničkog iskustva. Imerzivna metodologija razvoja novi je razvoj metodologija potrebnih za udruživanje razvoja aplikacija s dizajnom korisničkog iskustva. Impresivni razvoj stavlja nadarene profesionalce za razvoj aplikacija i korisničko iskustvo u središtu projekta i okružuje ih pratećom ekipom projektnih menadžera, poslovni analitičari i programeri podrške. Ova metodologija odbacuje pojam da je primjena razvoj je poput vođenja automobilske tvornice ili izgradnje trgovačkog centra. Dapače pristupa razvoju aplikacija kao kreativnom procesu koji je sličniji snimanju filma.

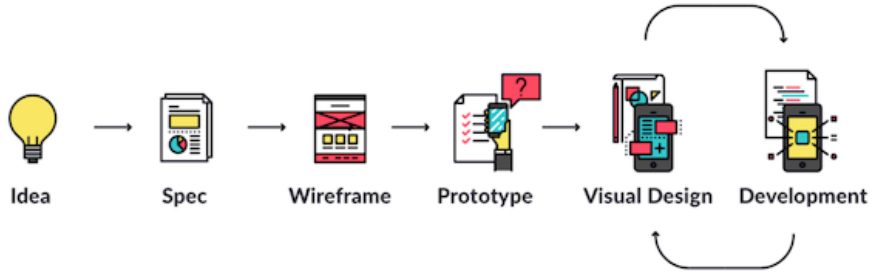
- Osnažite dizajnere, pa čak i krajnje korisnike da vam pomognu u razvoju.

Uloge poslovnih ljudi koji su sposobni za razvoj aplikacija povećavaju se zbog kombinacije tehnološki pametna milenijska generacija koja ulazi u radnu snagu, širenje jednostavnih za upotrebu razvojni alati i rastuća potražnja za aplikacijama. Microsoft nudi novi dizajn alat nazvan SketchFlow koji dizajnerima omogućuje izradu žičanih okvira i prototipova male vjernosti a zatim napredak do prototipova veće rezolucije sve do razvoja. Adobe planira ponuditi sličan alat pod nazivom Catalyst.26 Osim alata, morate navesti i kraj korisnički programeri s određenim skelama za obradu kako bi mogli učinkovito surađivati s njima profesionalni programeri.[4]

[4] Gualtieri, M. (2009). Deputize End-User Developers To Deliver Business Agility And Reduce Development Costs
preuzeto 26.09.2020

5. Proces i arhitektura izrade korisničkog sučelja

Pri izradi poslovnih korisničkih rješenja, bitno je imati ideju. To bi mogla biti vaša ideja ili ideja s kojom vam se klijent obratio. Ideje su sjajne, ali mnogo je faktora koji ideju mogu pretvoriti u radeći sustav, ali i manje uspješnih. Skloni smo stavljati previše zaliha u ideje, jer je pravilno shvatiti ideju daleko manje važno nego što ljudi misle. O ideji kada pričamo, lako možemo subjektivno gledati na istu i pridavati previše pažnje imaginarnim maštanjem o najboljem ishodu. Ideje trebaju biti podvrgnute zdravoj dozi darvinizma prije nego što se progone - preživljavanje najsposobnijih. Kroz praktični rad slijedio sam upute procesa „Smashing Magazina“ i njihovom shvaćanju istoga.



Slika 1. Proces razvoja aplikacija(Izvor: Flaroup Michael, 2016)

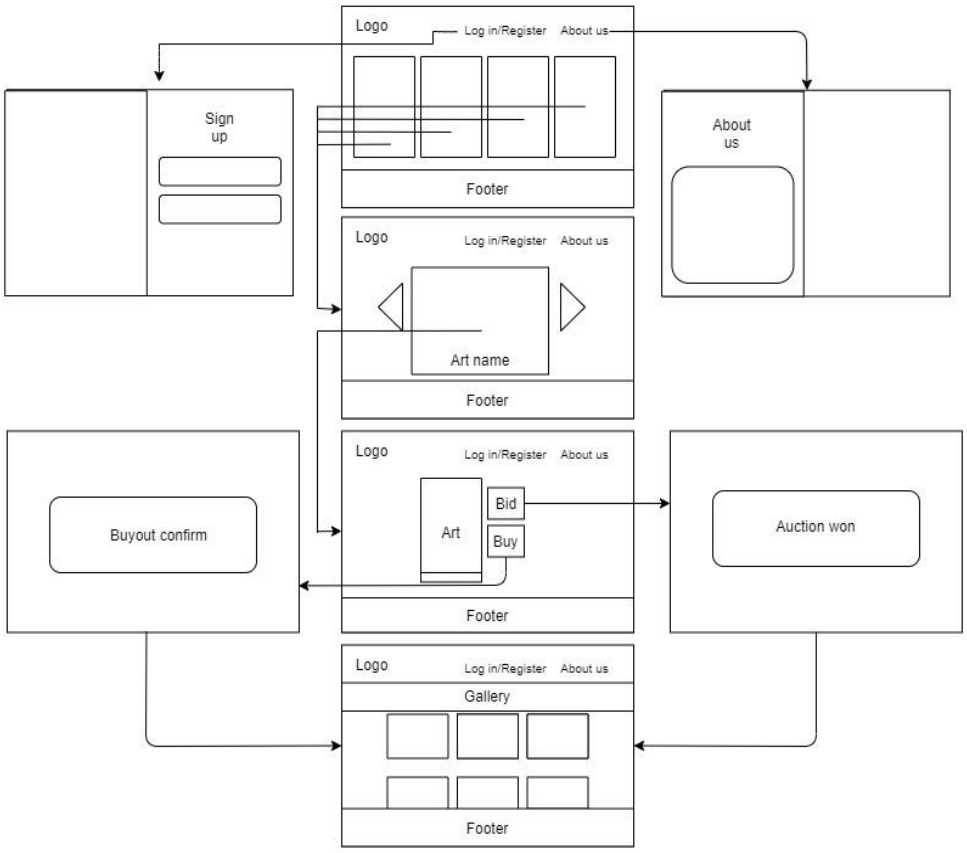
5.1. Ideja

Ideja koju sam razvio je aukcija umjetnina (točnije umjetnina iz primijenjene umjetnosti). Sustav bi služio umjetnicima da izlažu svoje radove, dok bi ljubitelji umjetnina mogli kroz jednostavan korisnički sustav lako kupovati željene umjetnine. Bitno je ideju staviti na papir i znati koji je naš cilj. Znati za kakvu publiku razvijamo sustav i kretanje istim. Aspekt bitan za razmisliti u stadiju „Ideje“ je korisničko iskustvo vezano uz naše web mjesto odnosno poslovno korisničko sučelje. Kod ideje razmišljao sam o publici, kojim kanalima bi privlačili nove klijente, kako i umjetnike koji bi svoju umjetninu izlagali. Svaku ideju vjerujem da se mora istražiti, prije kretanja u praktični dio. Sama konkurencija može imati slično rješenje za pojedini problem ili servis. Sama sličnost izvedbe ne treba biti zastrašujuća. Ako naš sustav, naizgled sličan konkurentnom, ima za ponuditi rješenje više ili samo elegantniju izvedbu istoga, već možemo biti u prednosti. Ideju

treba moći podlagati promjenama, čak i učestalijima no što u prvom planu mislimo. Evolucijom same ideje od zamisli preko zrele ideje spremne za viši stadij razvoja, želimo je dotjerati do najbolje verzije što može postati.[5]

5.2. Wireframe

Wireframe (hrv. Žičani okvir) je vizualni prikaz korisničkog sučelja, lišen bilo kakvog vizualnog dizajna ili elemenata marke. UX dizajneri ga koriste za definiranje hijerarhije stavki na zaslonu i priopćavanje onoga što bi stavke na toj stranici trebale temeljiti na potrebama korisnika. Wireframe okviri tlocrt su vaše web stranice, stvarajući posudu u koju se dizajn i sadržaj mogu ulijevati. O gradnji okvira najlakše je razmišljati u smislu gradnje kuće. Prije nego što primijenite svu boju i namještaj, arhitekt treba nacrtati tlocrt.



Slika 2. Wireframe aplikacije

[5] M. Flarup(2016). What You Should Know About The App Design Process preuzeto 27.09.2020. s https://www.smashingmagazine.com/2016/11/what-everyone-should-know-about-the-process-behind-app-design/

Također biste se trebali potruditi da razmislite kakav će biti sadržaj na stranici. „Lorem Ipsum“ dobro radi za dizajn tiska, ali morate razmisliti informacijama koje korisnik treba vidjeti na stranici, koju vrstu jezika treba koristiti, te gdje bi trebao biti prateći sadržaj.Iako sadržaj zasigurno ne mora biti konačni članak, to bi trebao biti približan približavanje. Uz neke rijetke iznimke, ne želite stavljati slike u svoje žičane okvire. Ako se radi o web mjestu s posebnom robnom markom, možda ćete htjeti pružiti komentar u kojem se navodi kakvu sliku zamišljate tamo.[6]

5.3. Analiza i prikaz kretanja po sučelju

Korisničko sučelje sastojat će se od zaglavlja, sredine(centra) i podnožja. Zaglavlje sadrži logotip, gumb za prijavu i gumb za dobivanje više informacija o sustavu i samoj usluzi.Podnožje sadrži ime web mjesta i informaciju za prava korištena u web sustavu. Dolaskom na web mjesto biti će prikazana podjela pojedinih kategorija primjenjene umjetnosti. Kada kažem „pojedinih” htio bi spomenuti da sam od mnoštva različitih oblika primjenjene umjetnosti odabrao četiri koje smatram dobrima za provedbu prototipa ili ranije verzije sustava.

Kategorije koje sam odabrao su redom:

- Skulptura
- Slikarstvo
- Lončarstvo
- Fotografija

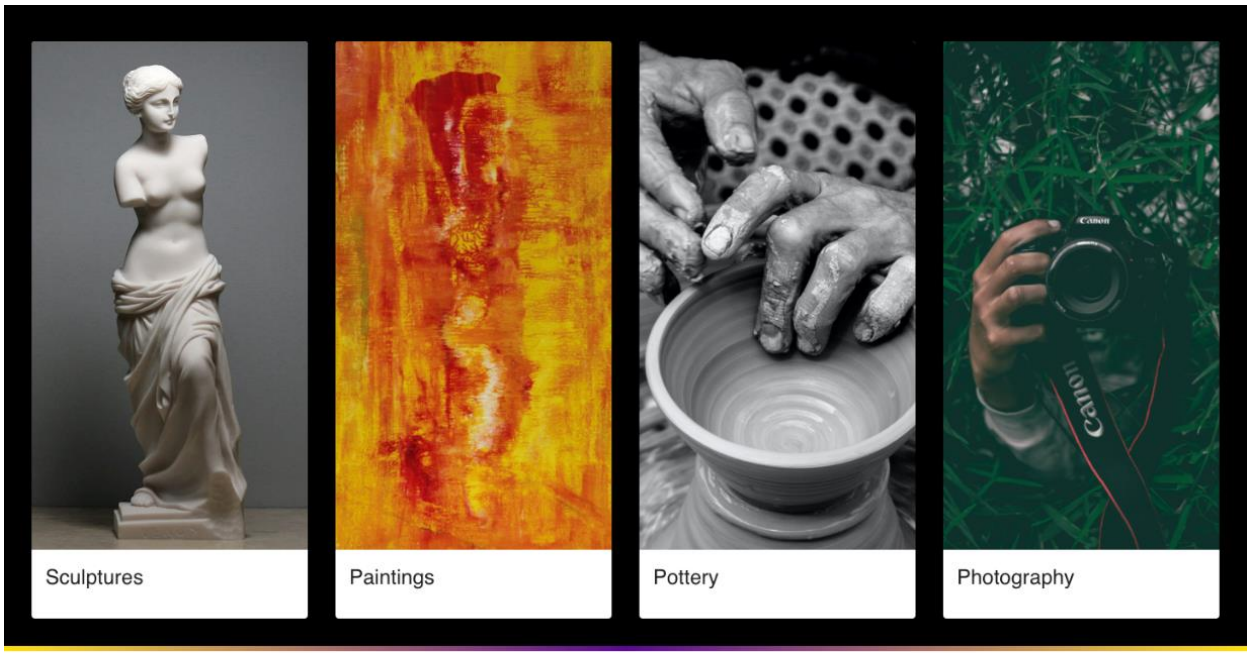
Kroz povijest, posebno u zapadnom svijetu, umjetnost je oduvijek postojala pod svojevrsnom hijerarhijom - povijesno slikarstvo bilo je iznad portreta, portretiranje iznad mrtve prirode, poezija iznad proze, opera iznad popularne pjesme. Primijenjena umjetnost, koliko god je bila potrebna i uvijek prisutna, morala je poslovično sjesti na svoje „važnije" kreativne kolege. Tek krajem 19. stoljeća, usponom pokreta „Umjetnost i obrt i secesija”, primijenjena umjetnost dolazi u prvi plan pozornosti javnosti. U širem smislu riječi, primijenjena umjetnost je ona u kojoj se umjetnički dizajn primjenjuje na utilitarne predmete svakodnevne uporabe (za razliku od likovnih umjetnosti koje nemaju drugu funkciju osim estetskog i intelektualnog stimuliranja gledatelja). Industrijski dizajn, grafički dizajn, modni dizajn, dizajn interijera i dekorativne umjetnosti spadaju u područje primijenjene umjetnosti. Nudeći beskrajne mogućnosti

[6] C. Mears(2013). Wireframes – The Beginner's Guide
preuzeto 27.09.2020. s https://uxmentor.me/wireframes-beginners-guide/

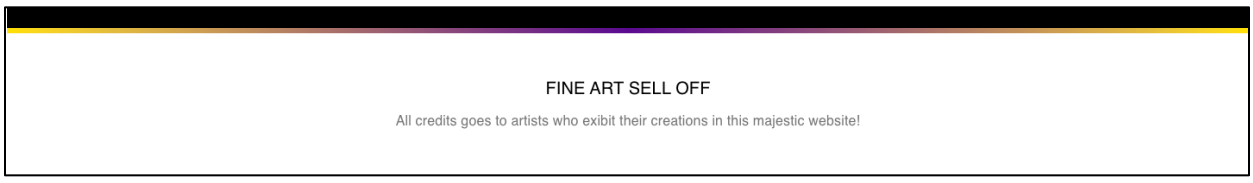
originalnosti, ove umjetnosti svojim doista neizmjernim opsegom utjecaja obogaćuju naš svakodnevni život i čak i najprizemniji pribor u našem domu mogu učiniti zadovoljstvo za oči.[7]



Slika 3. Zaglavlje



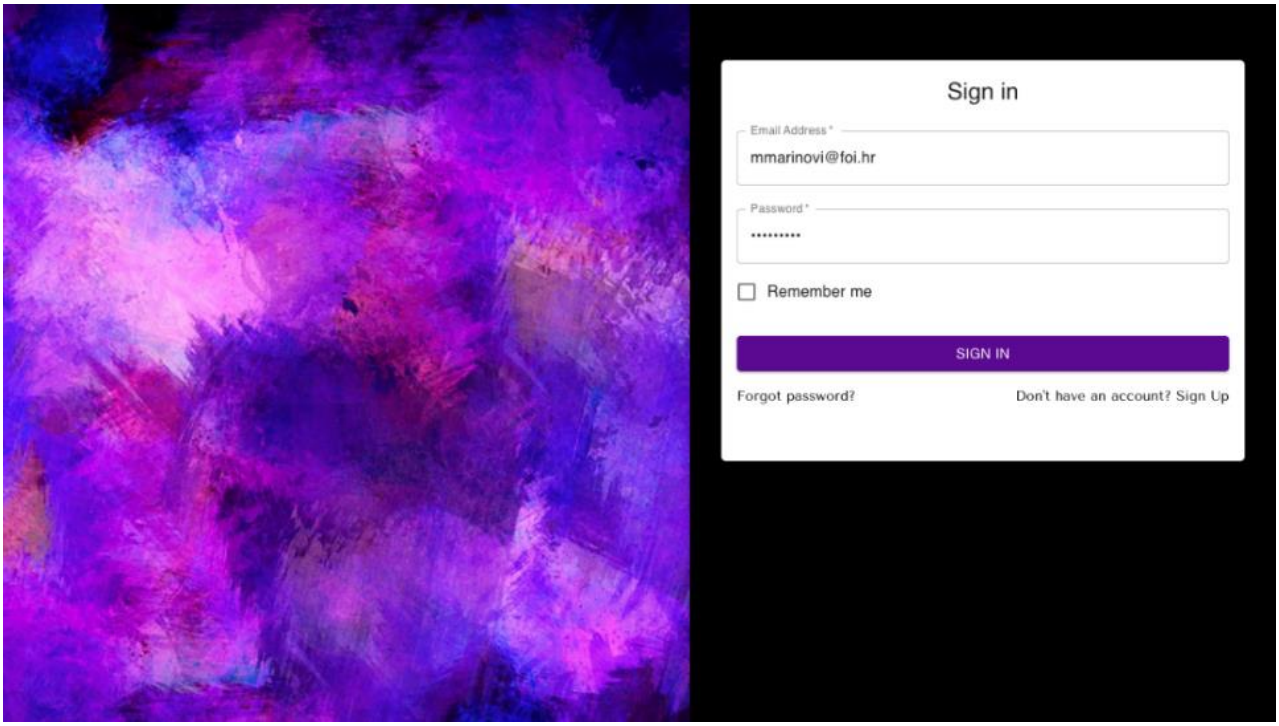
Slika 4. Body stranice



Slika 5. Podnozje

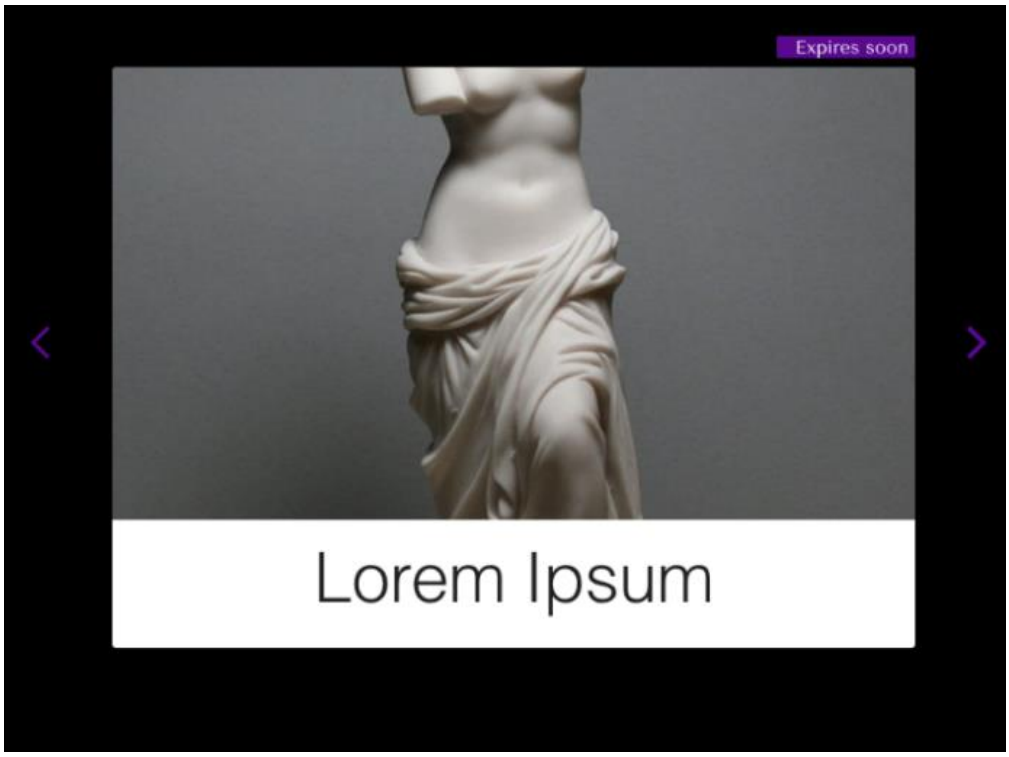
[7] F. Godward(2016.) The Variety of Applied Arts
preuzeto 28.09.2020. s https://www.widewalls.ch/magazine/applied-arts-meaning-kinds

Sustav bi imao mogućnost prijave. Sadržavao bi role i identifikator korisnika kako bi se mogle pratiti galerije kupljenih radova i statusa samog korisnika u postojećoj aukciji. Navigacijski gumb za prijavu je smiješten u desnom dijelu zaglavlja svake stranice. Navigiranjem na stranicu prijave od korisnika se zatražuje da upiše svoju email adresu i lozinku za prijavu u sustav. Korisnik može zatražiti opciju za pamćenje korisničkog imena, resetiranje lozinke ili samu registraciju u sustav.



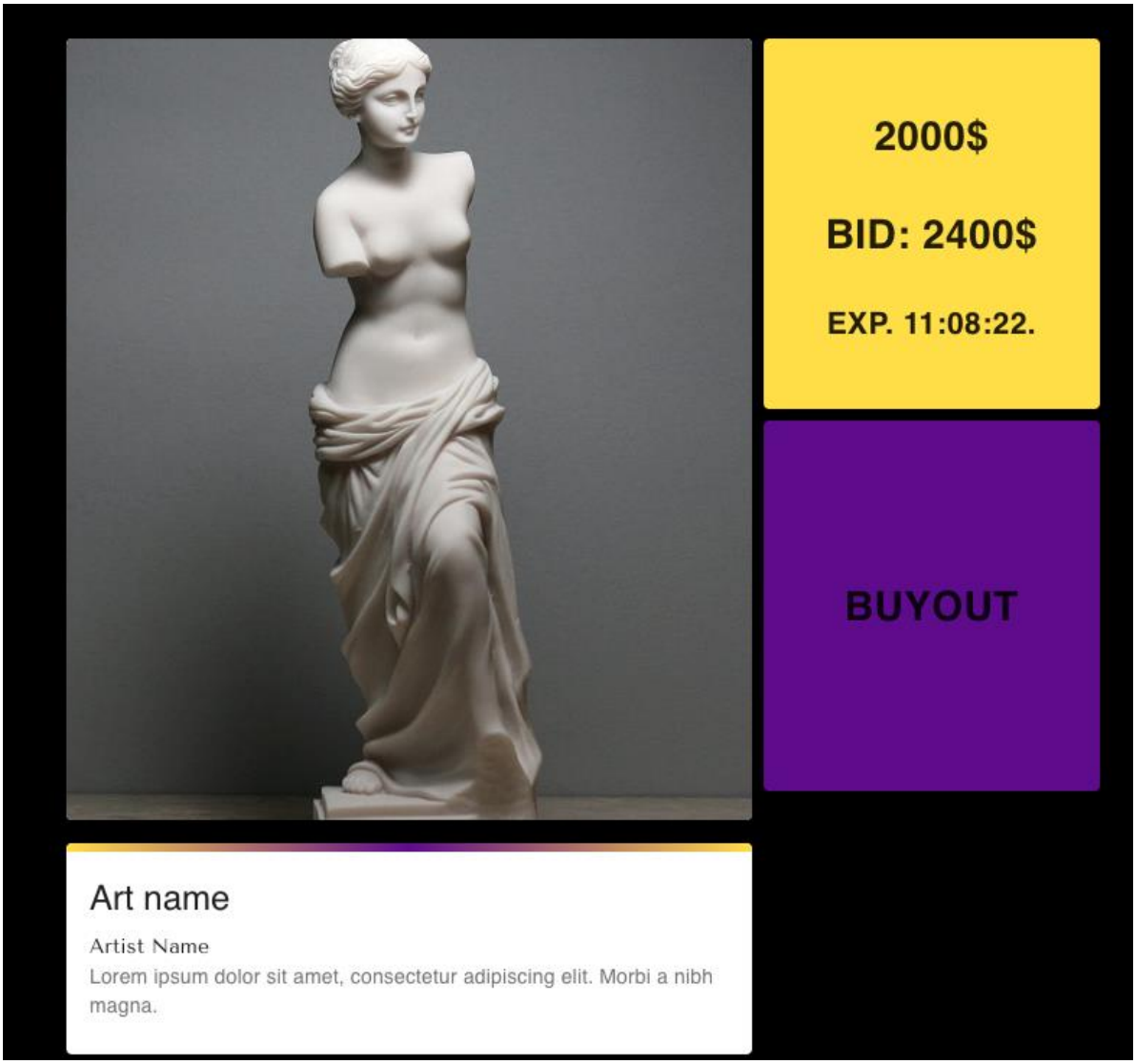
Slika 6. Prijava

Odabirom kategorije dolazimo do niza umjetnina iz istog žanra primjenjene umjetnosti. Iz kolekcije možemo odabrati pritiskom na karticu za željenom umjetninom. Korisnik akcijama “lijevo” i “desno” na gumb prolazi po kolekciji te razgledava radove. Svaki rad sadrži sliku i naziv umjetnine. Namjerno nisam htio prikazati ime umjetnika kako bi se više pažnje obratilo na sam umjetnički rad nego na ime samoga umjetnika. Nije isključivo da se ova opcija u budućnosti može izmjeniti. Umjetnina može također imati oznaku „Exipires soon”. Oznaka se može koristiti kao pomoć korisniku pri odluci o eventualnoj kupnji umjetnine.



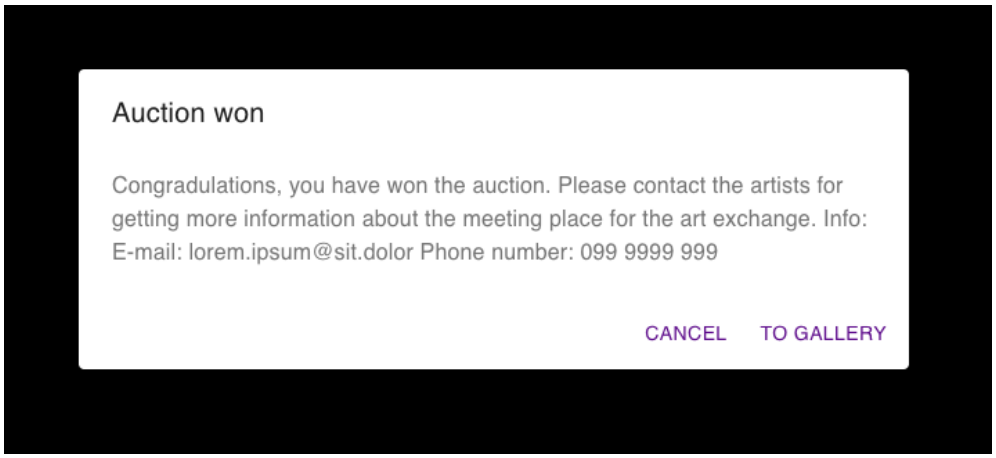
Slika 7. Kolekcija slika

Klikom na karticu željenog entiteta tražene umjetnine, dolazimo na stranicu gdje vidimo više pojedinosti o odabranoj umjetnini. Na centru ekrana primjećujemo odabranu umjetninu sa opisom prikazanim ispod umjetnine. Opis sadrži ime umjetnika koji je izradio umjetninu kao i opis koje je isti napisao za tu umjetninu. Za odabranu umjetninu možemo otkupiti umjetninu ili staviti svoj iznos kojime sudjelujemo u aukciji, odnosno odabrati „buyout“ i „bid“. Svaka će umjetnina imati početnu cijenu. Programski sam napravio da se svaka početna cijena pritiskom na žuti gumb („Bid“) poveća za 10%. Ukoliko vrijeme istekne, a korisnik je zadnji koji je uložio ponudu na određenu umjetninu, smatra se da je pobijedio na aukciji. Tada navigacijom dolazi na međustranicu gdje su mu ispisani podaci o umjetniku čiju je umjetninu kupio. U inicijalnoj verziji programa, napravljeno je da ako korisnik pobijedi na aukciji, ispišu mu se podaci o umjetniku koji prodaje tu umjetninu(telefonski broj i e-mail adresa umjetnika). Slaganje sustava za plaćanje karticom ili nekim drugim servisom naplaćuje se ili je kompleksno za sadašnji stadij aplikacije.Razmatra se implementacija takvog sustava za daljnje razvijanje sustava.

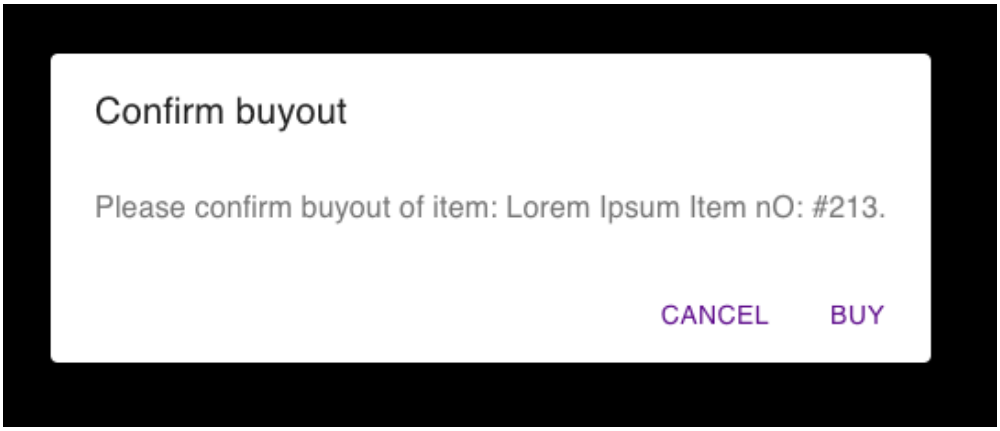


Slika 8. Umjetnina i akcije kupovanja

Pritiskom na ljubičasti gumb, otkupljuje se prikazana umjetnina. Nakon pritiska na gumb, dolazimo na međustranicu gdje se sam korisnik ispituje želi li doista otkupiti umjetninu po punoj njenoj cijeni. Razlog zašto sam implementirao takvu navigaciju je potencijalna greška u kliku gumba koja bi mogla rezultirati ne željenim ponašanjem sustava, tada bi korisnik slučajno kupio umjetninu jer je krivo kliknuo. Takvo ponašanje želimo izbjeći. Štetno je za sami sustav, korisnika koji je slučajno otkupio umjetnički rad i također za samog umjetnika koji svoj rad izlaže na našem web mjestu.



Slika 9. Skočni prozor za dobivenu aukciju

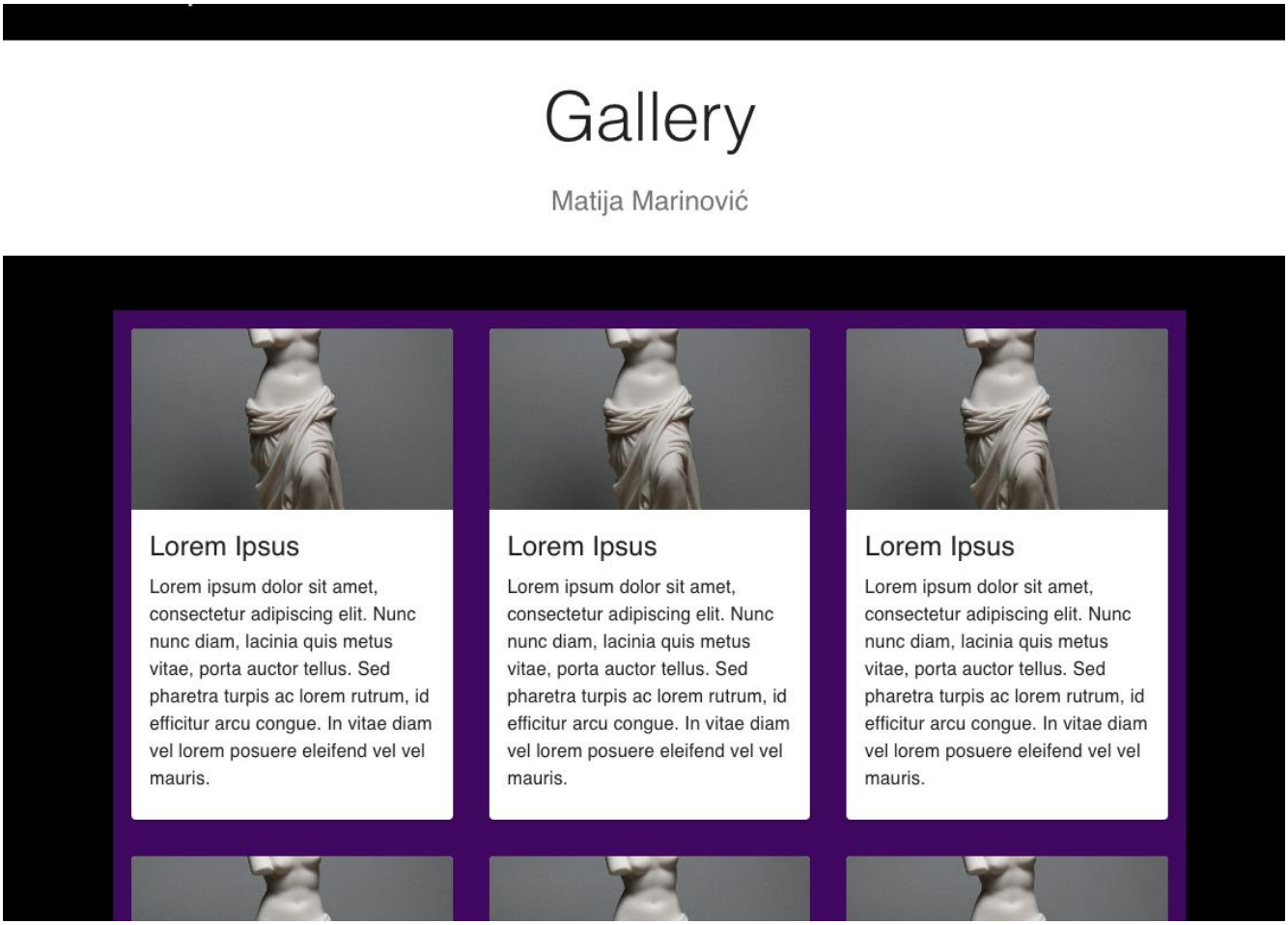


Slika 10. Skočni prozor za dobiven otkup

Modalni dijalog je dijalog koji se pojavljuje na vrhu komponenata glavnog sadržaja(po osi z) i zahtjeva potvrdu akcije od korisnika. Ovaj dijaloški okvir onemogućuje glavni sadržaj dok korisnik izričito ne stupi u interakciju s modalnim dijaloškim okvirom. Modalni dijalozi prvotno su trebali upozoriti korisnike na pogrešku ili na neko drugo stanje sustava koje je zahtijevalo trenutnu korisničku akciju. U tim je slučajevima bilo neophodno da korisnici budu prekidani kako bi ispravili pogrešku. Dakle, postavljanje dijaloškog okvira na sredinu zaslona kao žarišne točke sučelja učinilo ga je vrlo učinkovitim. Velika prednost takvih modalnih dijaloga bila je u tome što su privukli pažnju korisnika i omogućili im da prepoznaju problem i brzo ga isprave.[8]

[8] T. Fassenden(2017). Modal & Nonmodal Dialogs: When (& When Not) to Use Them preuzeto 01.10.2020. s https://www.nngroup.com/articles/modal-nonmodal-dialog/

Galerija slika sastoji se od skupa umjetnina koje je pojedini korisnik kupio odnosno u kojim je aukcijama pobijedio. Ulaskom na galeriju vidimo ime korisnika koji je trenutno prijavljen. Ne prijavljeni korisnik ne može doći navigacijom do web stranice sa tim url-om. Korisnik ne može vidjeti galeriju od drugog korisnika. Galerija umjetnina ujedno je posljednja stranica sustava, te iz galerije korisnik se može navigirati do početne stranice gdje kretnja sustav-om ponovno počinje od početka.



Slika 11. Galerija

6. Tehnologije i biblioteke

Prikaz web stranice ili na engleskom rasprostranjeniji izraz *front-end* je sve što vidite i s čime možete komunicirati pomoću preglednika. Dakle, stvaranje ovog vizualnog dijela naziva se *front-end* razvojem. Mogli biste čak reći da su dizajneri koji izrađuju korisničko sučelje i iskustva u planiranju također *front-end* programeri, jer rade u suradnji na istom dijelu projekta. Da bi stvorili prednji kraj, inženjeri koriste kombinaciju HTML-a (za osnovnu strukturu stranice i sadržaj), CSS-a (za vizualno uređivanje) i JavaScript-a (za izradu interaktivnih web stranica). Isti set alata koristi se za stvaranje progresivnih web aplikacija - mobilnih aplikacija koje izgledaju i osjećaju se kao izvorne, ali stvorene su korištenjem front-end tehnologija.

Tehnologijama kojima sam se služio opisati ću u nastavku a to su:

- HTML
- React
- JavaScript
- CSS
- Sass

6.1. HTML

HTML (ili Hypertext Markup Language) je računalni jezik dizajniran za stvaranje web stranica koje kasnije može istražiti svatko tko pristupi Internetu. HTML se obično koristi za strukturiranje web dokumenta. Definira elemente kao što su naslovi ili odlomci i omogućuje ugradnju slika, videozapisa i drugih medija. HTML se sastoji od niza kratkih kodova koji se nazivaju oznakama, a koje je kreator web mjesta normalizirao u tekstualnu datoteku. Tekst se zatim pohranjuje kao HTML datoteka i promatra kroz preglednik. Preglednik skenira datoteku i interpretira tekst u vidljivi oblik, a u najboljem slučaju prikazuje stranicu onako kako je dizajner planirao.HyperText je način na koji putujemo mrežom klikom na hiperveze - specifični tekstovi koji vas vode na druge stranice. Hyper znači da je nelinearan, što omogućuje premještanje na bilo koje drugo mjesto, jer za to ne postoji unaprijed definirani redoslijed. Označavanje određuje kvalitete koje HTML oznake primjenjuju na tekst u njima. Oznake ga označavaju kao određenu vrstu teksta.Kao jezik, sadrži kodne riječi i sintaksu kao i bilo koji drugi jezik.Od svog početnog izdanja 1991. godine, HTML je doživio mnoga ažuriranja. HTML5 objavljen 2014. Dodao je takve

značajke kao podrška za izvanmrežnu pohranu medija, precizniji elementi sadržaja (tj. zaglavlje, podnožje, navigacija) i podrška za ugrađivanje zvuka i videa.[9]

6.2. React i JavaScript

JavaScript je skriptni ili programski jezik koji vam omogućuje implementaciju složenih značajki na web stranice - svaki put kad web stranica više od pukog sjedenja i prikazivanja statičnih podataka koje možete pogledati - prikazivanje pravodobnih ažuriranja sadržaja, interaktivnih karata, animiranih 2D / 3D grafika, pomicanje video jukeboxa itd. To je treći sloj kolača slojeva standardnih web tehnologija, od kojih smo dva (HTML i CSS)

Osnovni JavaScript jezik na strani klijenta sastoji se od nekih uobičajenih programskih značajki koje vam omogućuju sljedeće:

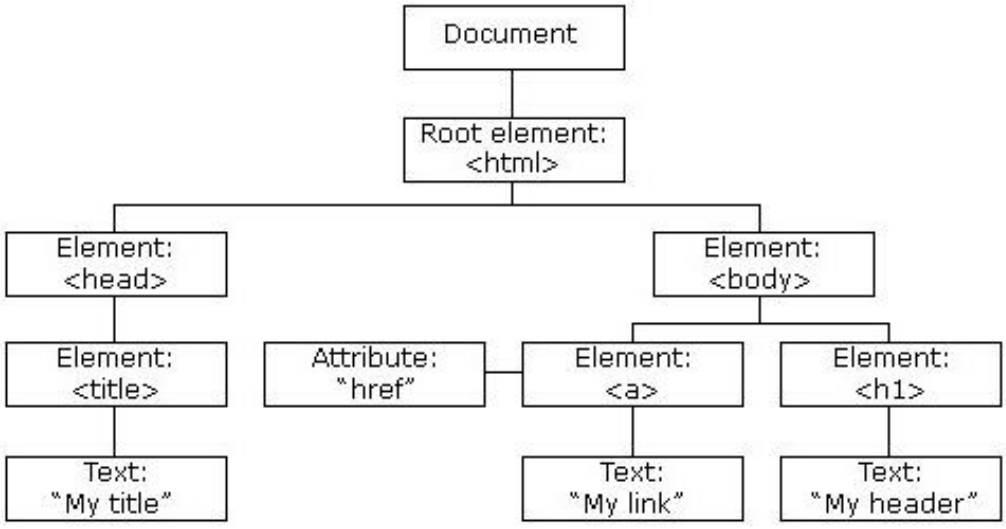
- Pohranite korisne vrijednosti unutar varijabli. Na primjer, u gornjem primjeru tražimo da se unese novo ime, a zatim ga pohranimo u varijablu koja se zove name.
- Operacije nad dijelovima teksta (u programiranju poznate kao "nizovi"). U gornjem primjeru uzimamo niz "Player 1:" i pridružujemo ga varijabli imena kako bismo stvorili cjelovitu tekstualnu oznaku, na pr. " Igrač 1: Chris ".
- Pokretanje koda kao odgovor na određene događaje koji se događaju na web stranici. Koristili smo događaj klika u našem primjeru iznad kako bismo otkrili kada se klikne gumb i zatim pokrenuli kôd koji ažurira tekstualnu oznaku.

Ono što je još uzbudljivije jest funkcionalnost izgrađena na vrhu JavaScript jezika na strani klijenta. Takozvana sučelja za programiranje aplikacija (API-ji) pružaju vam dodatne supersile koje možete koristiti u vašem JavaScript kodu.

API-ji su gotovi skupovi blokova koda koji programeru omogućuju implementaciju programa koje bi inače bilo teško ili nemoguće implementirati. Oni rade isto što i za programiranje kao što to rade gotovi kompleti namještaja za izgradnju kuće - mnogo je lakše uzeti gotove ploče i zamotati ih kako bi napravili policu s knjigama, nego li sami razraditi dizajn, otići i pronaći ispravite drvo, izrežite sve ploče na pravu veličinu i oblik, pronađite vijke ispravne veličine, a zatim ih sastavite da napravite policu za knjige.

[9] Altexsoft(2020). What is Front-End Development: Key Tehnologies and Concepts preuzeto 03.10.2020. s https://www.altexsoft.com/blog/front-end-development-technologies-concepts/

Kada govorimo o React-u , govorimo o novoj tehnologiji za razvoj weba i mobitelih sučelja koje je uveo Facebook. Projekt je započeo Jordan Walke, Facebook softverski inženjer, 2011. Kako bi pojednostavio proces razvoja i potaknuo ugodnije korisničko iskustvo, odlučio je stvoriti knjižnicu koja će omogućiti izgradnju web sučelja s JavaScriptom.[10] Virtualni DOM u Reactu čini korisničko iskustvo boljim, a razvojni programer bržim. DOM (objektni model dokumenta) je logična struktura dokumenata u HTML, XHTML ili XML formatima. Opisujući ga laički rečeno, to je sporazum o gledanju unosa i izlaza podataka koji ima oblik stabla. Web preglednici koriste mehanizme izgleda za transformiranje ili raščlanjivanje HTML-sintakse reprezentacije u objektni model dokumenta, što možemo vidjeti u preglednicima.



Slika 12. DOM stablo objekata (Izvor: W3C Schools)

Glavna briga tradicionalne konstrukcije DOM-a je način na koji obrađuje promjene, tj. Korisničke unose, upite i tako dalje. Poslužitelj neprestano provjerava razliku uzrokovanu tim promjenama kako bi pružio potreban odgovor. Da bi pravilno reagirao, također treba ažurirati DOM stabla cijelog dokumenta, što nije ergonomski valjano, jer su DOM stabla danas prilično velika i sadrže tisuće elemenata.Tim iza Reacta uspio je povećati brzinu ažuriranja pomoću

[10] MDN contributors(2020) What is JavaScript? preuzeto 04.10.2020.s https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

virtualnog DOM-a. Za razliku od ostalih okvira koji rade sa stvarnim DOM-om, ReactJS koristi apstraktnu kopiju - virtualni DOM. Ažurira čak i minimalističke promjene koje je primijenio korisnik, ali ne utječe na ostale dijelove sučelja. To je također moguće zahvaljujući izolaciji komponenata React, do koje ćemo doći za minutu, i posebnoj strukturi podataka u knjižnici.[11]

Značajke React-a:

- Dopuštenje za ponovnu upotrebu React komponenata značajno štedi
- Jednosmjerni tok podataka u ReactJS-u pruža stabilan kod
- Facebook biblioteka otvorenog koda: neprestano se razvija i otvorena je za zajednicu
- Redux: prikladni spremnik stanja
- React Hooks: poboljšano upravljanje stanjem
- Širok skup alata za React i Redux

6.3. CSS i SCSS

CSS je jezik za oblikovanje koji programeri koriste za razvoj web stranica. Koristi se za oblikovanje web stranica i njihovo privlačenje. To je jedan od tri temeljna dijela web razvoja, a druga dva su JavaScript i HTML.CSS je dizajniran da omogući razdvajanje prezentacije i sadržaja, uključujući izgled, boje i fontove. Ovo razdvajanje može poboljšati dostupnost sadržaja, pružiti veću fleksibilnost i kontrolu u specifikaciji karakteristika prezentacije, omogućiti više web stranica za dijeljenje oblikovanja određivanjem relevantnog CSS-a u zasebnoj .css datoteci i smanjiti složenost i ponavljanje u strukturnom kontekstu.Svaka stavka ili element na web stranici dio je dokumenta napisanog na označnom jeziku.

Prednosti CSS-a:

- Dosljednost - CSS pomaže u stvaranju dosljednog okvira koji dizajneri mogu koristiti za izgradnju drugih web stranica. Zbog toga se povećava i učinkovitost web dizajnera.
- Jednostavnost upotrebe - CSS je vrlo lako naučiti i pojednostavljuje razvoj web stranica. Svi su kodovi smješteni na jednoj stranici, što znači da poboljšanje ili uređivanje redaka ne bi uključivalo prolazak kroz nekoliko stranica.

[11] Altexsoft(2020). The Good and the Bad of ReactJS and React Native
preuzeto 05.10.2020. s https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-reactjs-and-react-native/

- Brzina web stranice - Obično kôd koji se koristi za web mjesto može doseći do 2 stranice ili više. Ali sa CSS-om to nije problem. Potrebna su samo 2-3 retka koda, stoga baza podataka web mjesta ostaje neuredna i uklanja probleme s učitavanjem web stranice.

Mane CSS-a:

- Previše verzija - u usporedbi s drugim parametrima poput HTML-a ili Javascripta, CSS ima puno verzija - CSS1, CSS2, CSS2.1, CSS3. Zbog toga CSS postaje vrlo zbunjujući za upotrebu, posebno za početnike.
- Nedostatak sigurnosti - Budući da je CSS sustav zasnovan na otvorenom tekstu, on nema ugrađeni sigurnosni sustav koji će spriječiti njegovo nadjačavanje. Pristupom operacijama čitanja / pisanja,
- Komplikacije - CSS se može zakomplicirati upotrebom softvera treće strane kao što je Microsoft FrontPage.
- Problemi s različitim preglednicima - implementacija početnih CSS promjena na web mjestu jednostavna je za programere. Međutim, nakon što su promjene izvršene, morat ćete potvrditi kompatibilnost ako CSS prikazuje slične učinke promjena na svim preglednicima. To je jednostavno zbog činjenice da CSS različito radi na različitim preglednicima.[12]

CSS pretprocesor je skriptni jezik koji proširuje CSS dopuštajući programerima da napišu kôd na jednom jeziku i zatim ga prevode u CSS. Sass je možda trenutno najpopularniji pretprocesor, ali drugi poznati primjeri uključuju Less i Stylus.Prije nego što nastavimo dalje, na redu je brza najava javne usluge: većina web dizajnera rekla bi da ako ste novi u CSS-u, bolje je izbjegavati Sass (ili bilo koje pretprocesore, proširenja ili okvire) dok ste još uvijek učenje. Iako je istina da nude brojne prednosti u brzini i učinkovitosti, važno je da se zaista upoznate s osnovama CSS-a. Obavezno naučite osnovne koncepte prije nego što počnete istraživati prečace i zbunjivati stvari.Krajnji cilj Sassa je popraviti nedostatke CSS-a. CSS, kao što svi znamo, nije najbolji jezik na svijetu [potreban citat]. Iako je vrlo jednostavan za učenje, brzo može postati prilično neuredan, posebno na velikim projektima.Tu dolazi Sass, kao meta-jezik, kako bi poboljšao sintaksu CSS-a kako bi pružio dodatne značajke i korisne alate. U međuvremenu, Sass želi biti konzervativan u pogledu CSS jezika.[13] Poanta nije pretvoriti CSS u potpuno istaknuti programski jezik; Sass želi

[12] K. Casey, Difference between CSS and SCSS
preuzeto 05.10.2020. s https://codersera.com/blog/difference-between-css-and-scss/
[13] T. Coron(2020) What is Sass? Your guide to this top CSS preprocessor

pomoći samo tamo gdje CSS zakaže. Zbog toga započeti rad sa Sassom nije teže od učenja CSS-a: on jednostavno dodaje par dodatnih značajki.To je rečeno, postoji mnogo načina za korištenje ovih značajki. Neki dobri, neki loši, neki neobični. Ove smjernice trebaju vam pružiti dosljedan i dokumentiran pristup pisanju Sass koda.

Usporedba CSS-a i SCSS:

- SCSS je izražajniji - SCSS koristi manje količine redaka u svom kodu od CSS-a, što ubrzava učitavanje koda.
- Potiče pravilno gniježđenje pravila - standardni CSS ne podržava gniježđenje. Ne možemo napisati razred unutar drugog razreda. Kako projekt postaje veći, to donosi problem s čitljivošću i struktura ne izgleda lijepo.
- Sintaksa - Sintaksa SCSS-a sadrži udubljenja koja nedostaju CSS-u.
- SCSS omogućuje korisniku pisanje bolje umetnute dokumentacije - SASS je fleksibilan s komentarima, ali svaki dobar programer radije će imati ugrađenu dokumentaciju koja je dostupna u SCSS-u. Inline dokumentacija čini retke koda samorazumljivim.
- Bolja funkcionalnost - Uz pomoć SCSS-a kôdu možemo dodati više funkcionalnosti u obliku varijable, selektora i gniježđenja koji nisu prisutni u CSS-u.
- Prilagođavanje Bootstrapa - Poznavanje SCSS-a pomaže u prilagođavanju Bootstrapa 4.
- Matematičke operacije - SCSS nam omogućuje matematiku pomoću operatora. Možemo izvesti jednostavne izračune unutar našeg koda za bolji izlaz.[14]

preuzeto 06.10.2020. s https://www.creativeblog.com/web-design/what-is-sass-111517618

[14] K. Casey, Difference between CSS and SCSS
preuzeto 06.10.2020. s https://codersera.com/blog/difference-between-css-and-scss/

7. Izrada prototipa

Statički rezultati poput žičanih okvira i maketa često pokazuju strukturu i estetiku aplikacije, ali ne ilustriraju njen osjećaj i osjećaje. U nekim slučajevima konačne statičke isporuke uvijek se moraju prilagoditi zbog tehničkih ograničenja ili promjena sadržaja. Drugi nedostaci su to što se mikro interakcije i animacije ponekad zanemaruju tokom čitavog procesa dizajniranja, što čini iskustvo doživljajem ustajalog i neintuitivnog.

7.1. Logotip

Prije nego što uđemo u ono što čini dobre logotipe, vrijedi odvojiti malo vremena da definiramo što je logotip. Logotip je u osnovi simbol koji pomaže u prepoznavanju i promociji preduzeća, proizvoda, organizacije itd. Većina kompanija koje koriste apstraktni ili ilustrativni simbol imaju tendenciju da kombinuju izabrani simbol sa logotipom koji se temelji na fontu, što čini njihov logo mnogo lakšim za pamćenje, posebno u ranim danima prije nego što njihov logo postane sam prepoznatljiv.[15] Logotip moje aplikacije temeljio sam na apstraktnome prikazu. Moderana tipografija u kombinaciji sa jednostavnim ravnim linijama donosi eleganciju i uklapa se u cijeli koncept grafičkog korisničkog sučelja. Dizajn grafičkog sučelja radio sam počevši od logotipa, pa sam sav dizajn nakon temeljio na izgledu logotipa(tipografiju i minimalističke ravne linije).



Slika 13. Logotip

[15] A. Oduye(2018). Creating front-end prototypes for a great user experience preuzeto 07.10.2020. s https://www.heartinternet.uk/blog/creating-front-end-prototypes-for-a-great-user-experience

7.2. Boje

Smjernice za odabir boja:

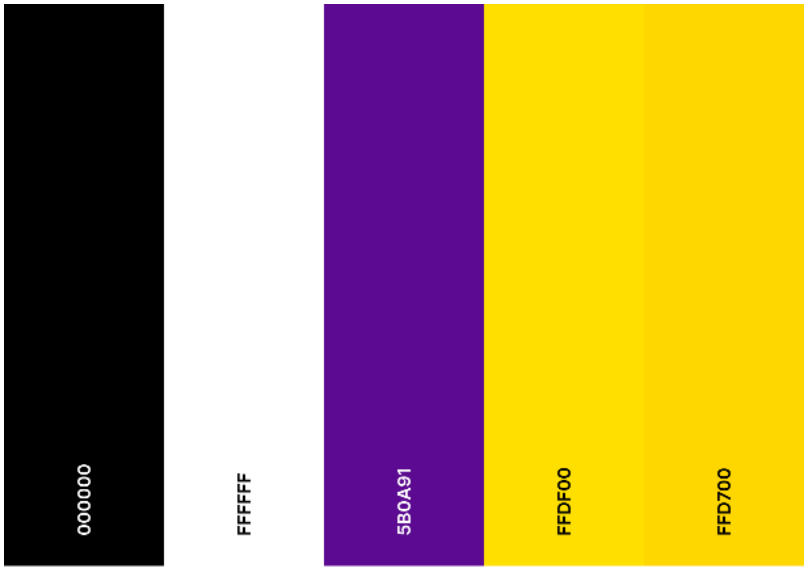
- Koristite ono što imate. Ako klijent ima logotip utvrđene boje, to će obično biti vaša
- Uklonite boje konkurenata. Ako jedan od vaših glavnih konkurenata ima jaku boju marke, nemojte ga kopirati ako mu možete pomoći.
- Razmislite o svojoj ciljnoj publici. Boje web stranice pogrebnog poduzeća vjerojatno bi se vrlo razlikovale od boja dječjeg kluba. Razmislite o tome tko će koristiti web stranicu i kako želite da se osjećaju (uzbuđeni, ozbiljni, zbrinuti itd.).[16]

Za odabir boja u svome radu odabrao sam kao glavne boje:

- Crnu
- Bijelu

Sekundarne boje:

- Metalik ljubičastu
- Zlatnu



Slika 14. Paleta boja

[16] L. Elisabeth(2016). A Simple Web Developer's Color Guide, preuzeto 10.10.2020. s https://www.smashingmagazine.com/2016/04/web-developer-guide-color

Kroz aplikaciju sam pomoću Sass predprocesora i Material UI-Theme definirao boje te sam ih mogao lako koristiti kasnije pisajući kod. Dobra praksa kod takvog pristupa je ta ako želimo promjeniti jednu od korištenih boja, na jednome mjestu je promjenimo i propagira se kroz cijelu aplikaciju.



Slika 15. SCSS rješenje

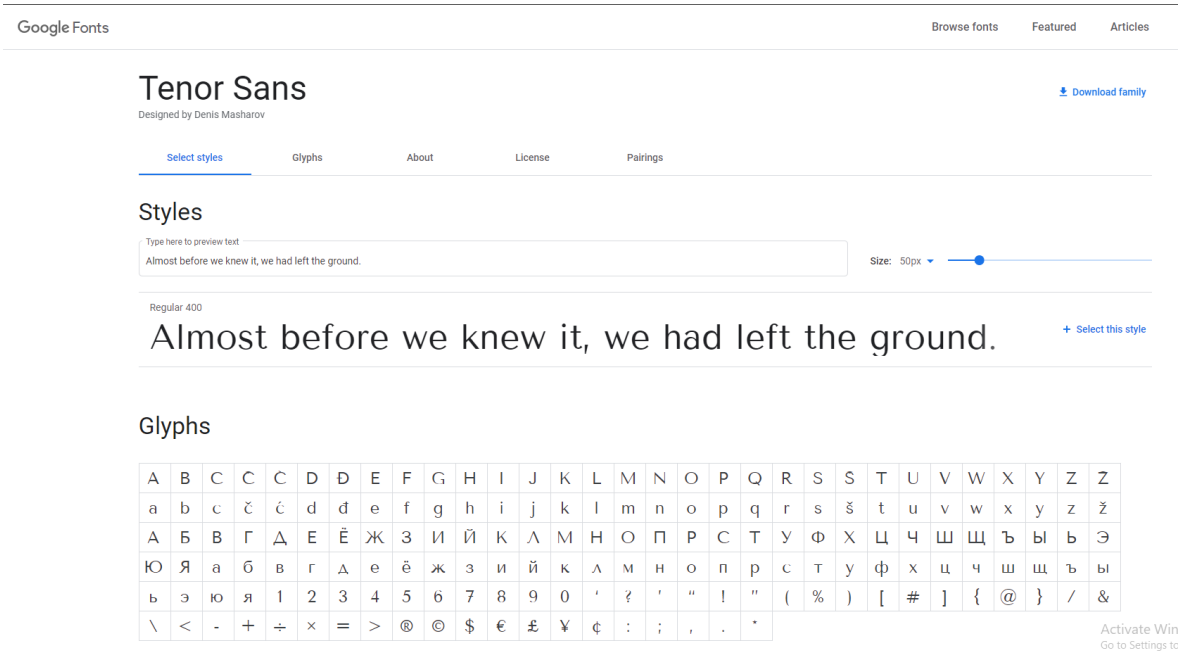
7.3. Tipografija

Kad je riječ o dizajnu tiska, malo je varijabli u načinu na koji će korisnik s njim komunicirati. Ispisana stranica neće se pomicati u obliku, ovisno o tome tko je drži - bojama se može strogo upravljati, a font će ostati statičan. Digitalna tipografija puno je fluidnija. Kada se tipografija prikaže na zaslonu, mnogi su čimbenici koji pridonose tome kako se prikazuje, a neki od njih uključuju:

- Veličina zaslona: Nemoguće je predvidjeti, međutim, responsive design želi prilagoditi bilo koji lagani pomak u dimenzijama. Ali to znači da se slaganje slova ponekad može učiniti uzalud.
- Razlučivost zaslona: Gustoća piksela igra veliku ulogu u čitljivosti teksta. Tehnološki napredak učinio je zaslone visoke razlučivosti dostupnijim masama, ali dizajneri bi trebali imati na umu odstupanja od starijih tehnologija.

- Kalibracija zaslona: Razine svjetline određuju kako će se boje pojaviti na zaslonu. Odabir boja za web dizajn mora prilagoditi svim vrstama postavki zaslona.[17]

Kada sam izradio logo, odlučio sam da će dizajn istoga uvelike utjecati na dizajn aplikacije u cijelosti. Tipografiju sam odabrao nakon izrade loga, te je primijenio na cijeli projekt. Pomoću anotacije import dohvatio sam željenu tipografiju i postavio je u globalnom CSS selektoru.



Slika 16. Tipografija (Izvor: Google fonts)

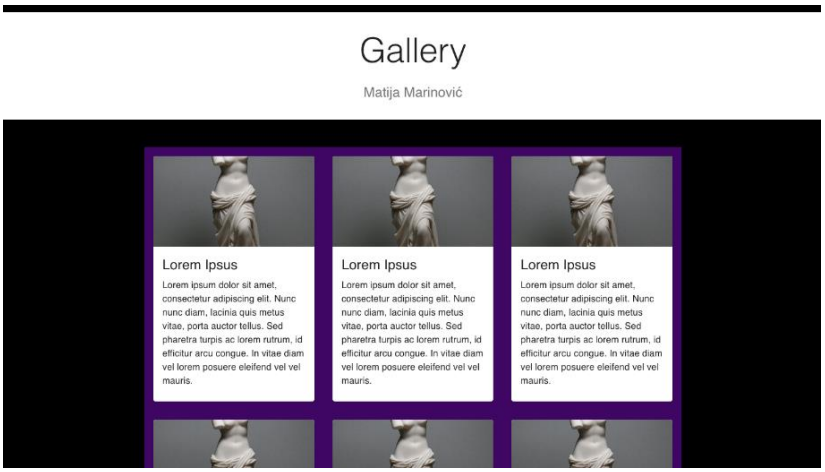
7.4. Prilagođavanje pozicioniranja elemenata

Prošlog desetljeća web tehnologija uvodila je sve više mogućnosti za stvaranje složenih i fleksibilnih izgleda s manje napora. Unatoč tome, unatoč napretku tehničkih mogućnosti i rastućim dizajnerskim timovima, web je i dalje preplavljen dosadnim sučeljima poput alata Bootstrap-a. Zašto je to? Možda zato što većina dizajnera ne razumije u potpunosti principe koji stoje iza reakcije. Prije svega, niti jedan od alata za dizajn ne omogućuje stvaranje istinski odzivnih sučelja (i usput vjerujem da bi tako trebalo i ostati). Stoga je većini timova nejasno tko je vlasnik reakcije

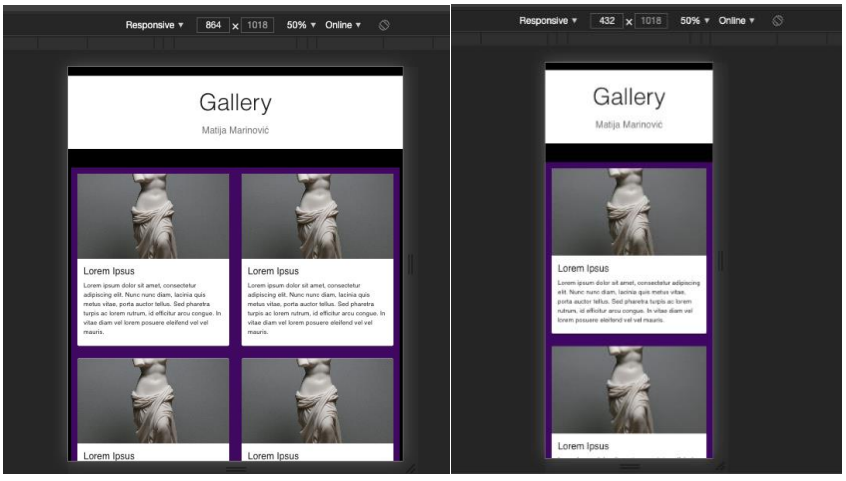
[17] J. Devos(2018) Designing for Readability – A Guide to Web Typography (with Infographic) preuzeto 10.10.2020. s https://www.toptal.com/designers/typography/web-typography-infographic

- dizajneri ne mogu preuzeti potpuno vlasništvo nad njom ako ne uskoče u kod, s druge strane, inženjeri nerado preuzimaju bilo kakvu odgovornost jer se to više osjeća kao tema dizajna.

Web stranica ili aplikacija reagiraju kada izgledaju dobro na svim uređajima: mega-zaslonima, prijenosnim računalima, tabletima i telefonima. Koristi najučinkovitiji izgled za svaku veličinu zaslona, vidljivost i izgled određenih elemenata mogu se razlikovati, veličine fontova i razmak su relativni.[18] U aplikaciji koristio sam *Grid* elemente i elemente *Box* kako bi elementi bili *responzivni* i dobro se skalirali s obzirom na rezoluciju ekrana.



Slika 17. Skalabilnost (cijeli prikaz)



Slika 18. Primjeri responzivnosti za manje uređaje

[18] A. Arteeva(2019). Principles of responsive design
preuzeto 11.10.2020. s https://blog.prototypr.io/principals-of-responsive-design-64c637402e7

7.5. Animacije

Animacija u aplikacijskom okviru ReactJs popularna je značajka i postoji mnogo načina za stvaranje različitih vrsta animacija. Mnogi programeri izrađuju animaciju isključivo pomoću CSS alata i dodavanjem klasa u HTML oznake. Ovo je sjajan način i trebali biste ga koristiti. Ako želite stvoriti složene animacije, možete obratiti pozornost na alat GreenSock. GreenSock je najmoćnija platforma za animaciju. U React-u postoji i puno knjižnica, komponenata za stvaranje animacije.

Primjer implementacije animacije sa React-om:

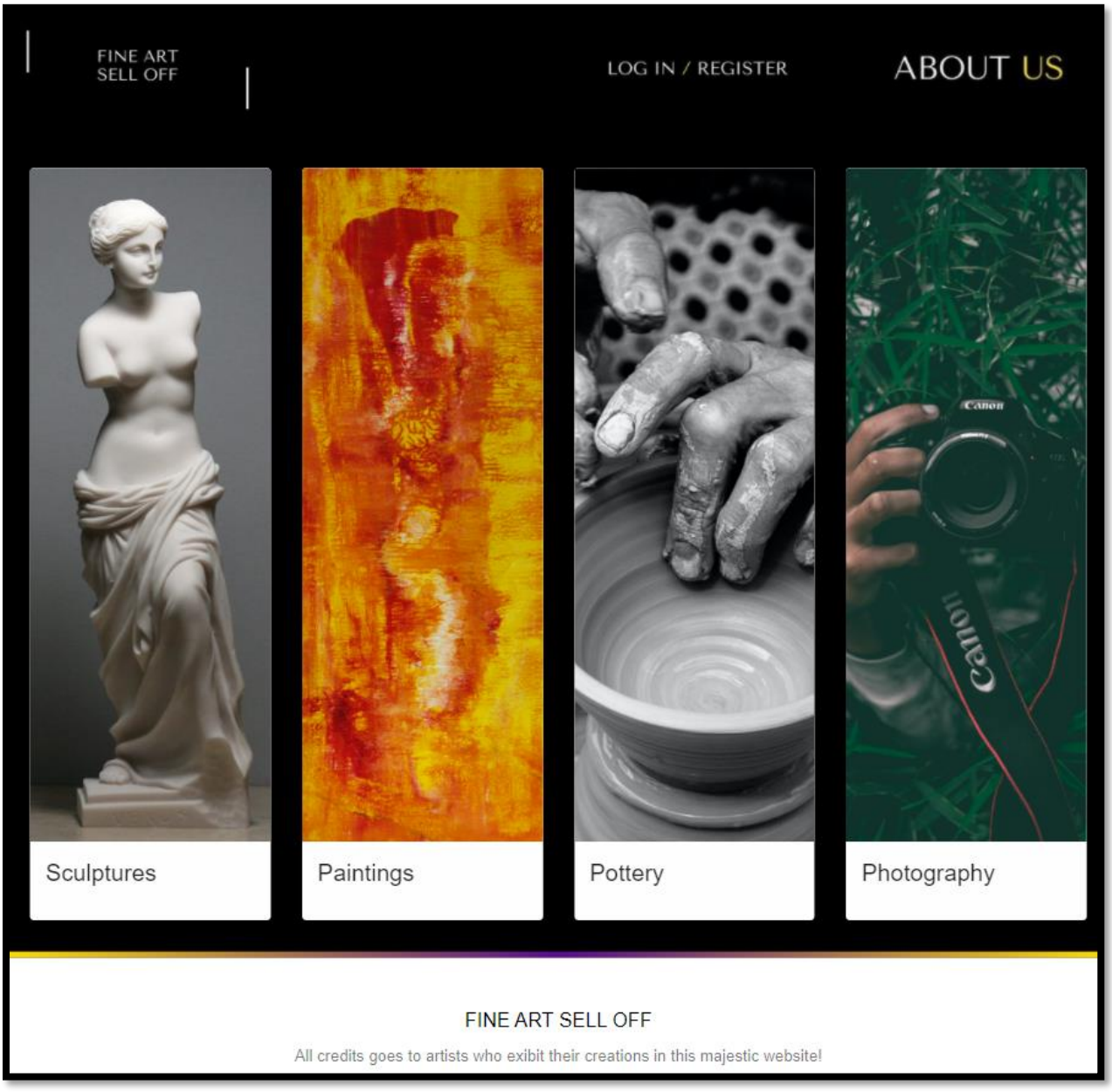
- CSS metoda
- React-transition-group — dodatak komponentama za jednostavniju implementaciju bazičnih CSS animacija i tranzicija
- React-animations — jednostavno korištenje, implementira sve animacije iz animate.css datoteke
- React Reveal — animacijski razvojni okvir za React. [19]

Za animiranje elemenata poslovnog korisničkog sučelja odabrao sam React Reveal. React Reveal je animacijski okvir za React. Ima osnovne animacije poput blijeđenja, okretanja, zumiranja, rotiranja i puno naprednijih animacija. Omogućuje vam kontrolu svih animacija rekvizitima, na primjer: položajem, kašnjenjem, udaljenostom, kaskadom i mnogim drugima. Možete ih vidjeti ovdje. Također možete koristiti prilagođene css efekte. Lista animacija koje možemo postići je slijedeća:

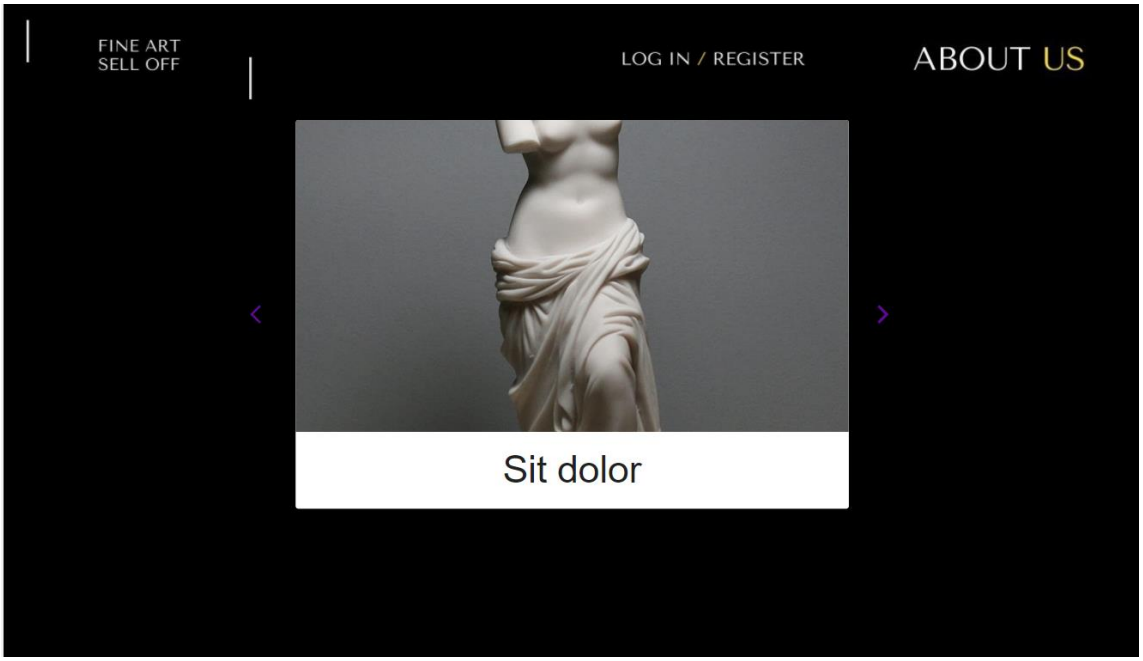
- Fade
- Flip
- Rotate
- Zoom
- Bounce
- Roll

[19] D. Noyhenko(2019) 5 Ways to animate a React app preuzeto 12.10.2020. s https://medium.com/hackernoon/5-ways-to-animate-a-reactjs-app-in-2019-56eb9af6e3bf

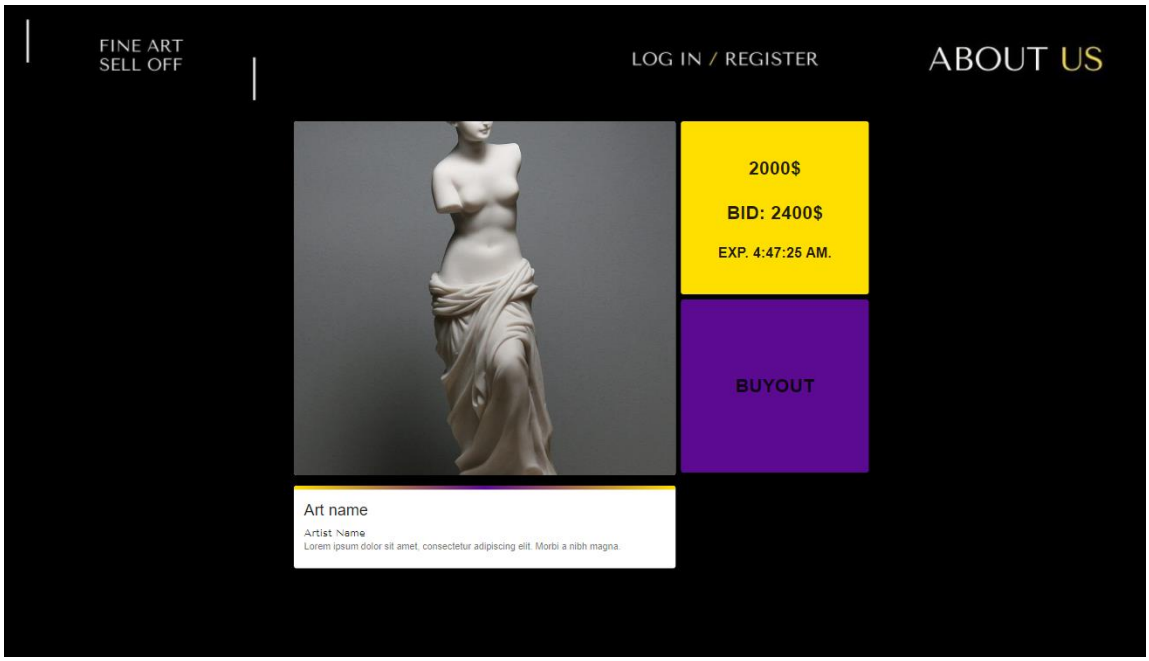
8. Sveobuhvatni primjer WWWUI-a



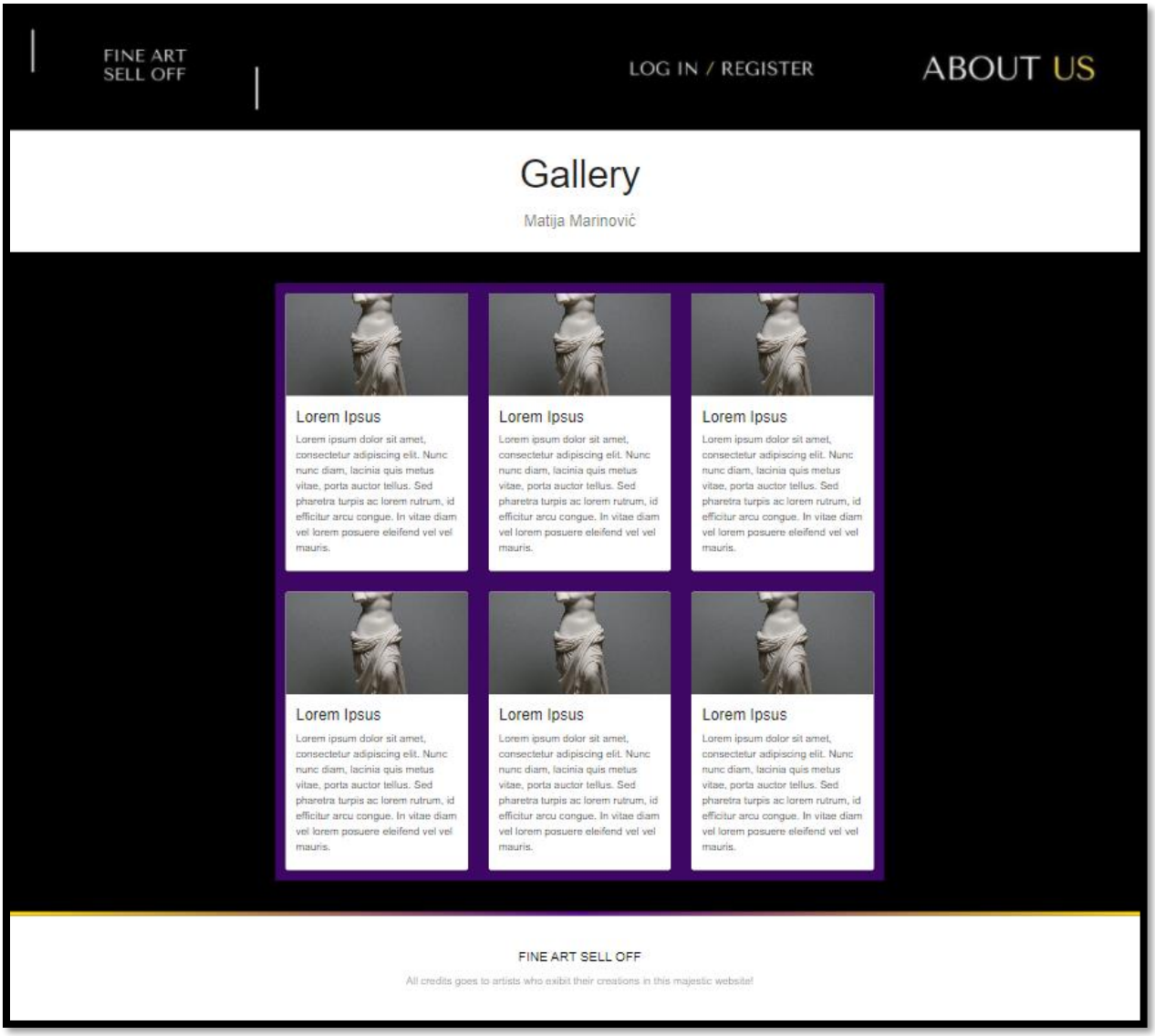
Slika 19. Početna stranica – odabir kategorije



Slika 20. Odabir umjetnine



Slika 21. Pregled cijene i akcija aukcije



Slika 22. Pregled galerije kupljenih umjetnina

9. Zaključak

Izrada poslovnog grafičkog korisničkog sučelja zahtjeva mnogo promišljanja prije kretanja u praktični dio izrade rješenja. Sav proizvod biva “rođen” u ideji, te je ideju kao takvu potrebno razraditi u detalje. Možemo razlikovati otvorene i zatvorene programe. Otvoreni sustavi koji su lako dostupni svima koji otvore račun. Korisnici mogu pristupiti takvim aplikacijama putem weba i mogu otvoriti račun besplatno ili plaćanjem naknade. Zatvorenim sustavima (ili poslovnim aplikacijama) obično nije dostupan izvan tvrtke koja ih koristi, a mogu se smatrati „izvan mrežnim“ aplikacijama. Takvi sustavi obično rade na lokalnoj mreži tvrtke i dostupni su samo zaposlenicima. Uspješno korisničko sučelje fokusira se na korisnike i njihove zadatke. To je ključno, a previše programera nije uspjelo stvoriti dobro korisničko iskustvo. Možete pratiti jedan od bilo kojeg broja procesa u dizajniranju korisničkog sučelja. Možda ga već imate. Međutim, predlažem da razmislite o agilnom pristupu . Za korisnike i klijente korisničko sučelje je proizvod. Sve što žele vidjeti je korisničko sučelje. Agile pristup pomaže kroz svoje ključno načelo: iterativni pristup. Svaka se iteracija sastoji od svih faza definiranih vašim postupkom. Sljedeći korak u procesu je stvaranje prototipova koji će simulirati stvarnu aplikaciju. Prototip može sadržavati jednu ili više značajki (ili sve njih), ali zapravo ne čini ništa. To samo simulira ponašanje stvarne aplikacije, a korisnici će osjetiti da zapravo nešto rade. Prototipi su beskorisni ako ih ne testiramo. Opće je poznato da softverski projekti ograničavaju vrijeme i proračun, pa da biste bili učinkovitiji, testirajte rano i često. Vjerojatno su ključni čimbenici dobrog korisničkog sučelja dosljednost i poznavanje. Korisničko sučelje mora biti dosljedno u svim dijelovima aplikacije, od navigacije do boja, sve do terminologije. To je poznato kao unutarnja konzistentnost. Ali i korisničko sučelje mora biti dosljedno u svom kontekstu, poput operativnog sustava ili drugih aplikacija u svojoj grupi. Dobar pristup dosljednosti je definiranje smjernica za korisničko sučelje za svaki projekt ili za skupinu projekata. Oni bi trebali voditi odlukama koje donosite za sve detalje. To ne samo da će zadržati dosljednost, već će poslužiti i kao dokumentacija koja će članovima tima pomoći da bolje razumiju vaše odluke. Korisnici su učinkoviti kada se usredotoče na određeni zadatak. Jednostavnost je osnovno načelo dizajna korisničkog sučelja. Što je jednostavnije korisničko sučelje, to je jednostavnije za upotrebu. Ali održavanje jednostavnih korisničkih sučelja za poslovne aplikacije izazov je jer aplikacije često imaju puno funkcionalnosti. Ključno je uravnotežiti funkcionalnost i jednostavnost. Suzdržanost je jedan od najučinkovitijih načina postizanja ove ravnoteže: tj. Pronalaženje najjednostavnijeg načina rješavanja problema.Estetika, iako subjektivna i pomalo proizvoljna, igra važnu ulogu u ukupnom zadovoljstvu.

Popis literature

Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. (2005*). User interface design and evaluation*. Elsevier. preuzeto 21.09.2020.

Jovanovic J. (2010). *User Interfaces in Business Web Application Design*
Preuzeto 21.09.2020. s https://www.smashingmagazine.com/2010/02/designing-user-interfaces-for-business-web-applications

Gualtieri, M. (2009). Best practices in user experience (UX) design. *Design Compelling User Experiences to Wow your Customers*, 1-17. preuzeto 24.09.2020

Gualtieri, M. (2009). *Deputize End-User Developers To Deliver Business Agility And Reduce* Development Costs
preuzeto 26.09.2020

M. Flarup(2016). *What You Should Know About The App Design Process*
preuzeto 27.09.2020. s https://www.smashingmagazine.com/2016/11/what-everyone-should-know-about-the-process-behind-app-design/

C. Mears(2013). *Wireframes – The Beginner's Guide*
preuzeto 27.09.2020. s https://uxmentor.me/wireframes-beginners-guide/

F. Godward(2016.) *The Variety of Applied Arts*
preuzeto 28.09.2020. s https://www.widewalls.ch/magazine/applied-arts-meaning-kinds

T. Fassenden(2017). *Modal & Nonmodal Dialogs: When (& When Not) to Use Them*
preuzeto 01.10.2020. s https://www.nngroup.com/articles/modal-nonmodal-dialog/

Altexsoft(2020). *What is Front-End Development: Key Tehnologies and Concepts*
preuzeto 03.10.2020. s https://www.altexsoft.com/blog/front-end-development-technologies-concepts/

MDN contributors(2020) *What is JavaScript?*
preuzeto 04.10.2020. s https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

Altexsoft(2020). *The Good and the Bad of ReactJS and React Native*
preuzeto 05.10.2020. s https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-reactjs-and-react-native/

K. Casey, *Difference between CSS and SCSS*
preuzeto 05.10.2020. s https://codersera.com/blog/difference-between-css-and-scss/

T. Coron(2020*) What is Sass? Your guide to this top CSS preprocessor*
preuzeto 06.10.2020. s https://www.creativebloq.com/web-design/what-is-sass-111517618

A. Oduye(2018). *Creating front-end prototypes for a great user experience*
preuzeto 07.10.2020. s https://www.heartinternet.uk/blog/creating-front-end-prototypes-for-a-great-user-experience

L. Elisabeth(2016). *A Simple Web Developer's Color Guide*,
preuzeto 10.10.2020. s https://www.smashingmagazine.com/2016/04/web-developer-guide-color

J. Devos(2018) *Designing for Readability – A Guide to Web Typography (with Infographic)*
preuzeto 10.10.2020. s https://www.toptal.com/designers/typography/web-typography-infographic

A. Arteeva(2019). *Principles of responsive design*
preuzeto 11.10.2020. s https://blog.prototypr.io/principals-of-responsive-design-64c637402e7

D. Noyhenko(2019) *5 Ways to animate a React app*
preuzeto 12.10.2020. s https://medium.com/hackernoon/5-ways-to-animate-a-reactjs-app-in-2019-56eb9af6e3bf

Popis slika

Slika 1. Proces razvoja aplikacija(Izvor: Flaroup Michael, 2016) 6
Slika 2. Wireframe aplikacije 7
Slika 3. Zaglavlje 9
Slika 4. Body stranice 9
Slika 5. Podnozje 9
Slika 6. Prijava 10
Slika 7. Kolekcija slika 11
Slika 8. Umjetnina i akcije kupovanja 12
Slika 9. Skočni prozor za dobivenu aukciju 13
Slika 10. Skočni prozor za dobiven otkup 13
Slika 11. Galerija 14
Slika 12. DOM stablo objekata (Izvor: W3C Schools) 17
Slika 13. Logotip 21
Slika 14. Paleta boja 22
Slika 15. SCSS rješenje 23
Slika 16. Tipografija (Izvor: Google fonts) 24
Slika 17. Skalabilnost (cijeli prikaz) 25
Slika 18. Primjeri responzivnosti za manje uređaje 25
Slika 19. Početna stranica – odabir kategorije 27
Slika 20. Odabir umjetnine 28
Slika 21. Pregled cijene i akcija aukcije 28
Slika 22. Pregled galerije kupljenih umjetnina 29