

Projekt iz spletnega programiranja – 1. faza

Ogrodje in organizacija spletišča

Statična stran HTML/CSS

Dinamičnost z JavaScript

Oktober 2016, v1.2

Rok oddaje na učilnici: 27. 11. 2016, možni (in priporočeni) predhodni zagovori.

Predavatelj: dr. Aleš Smrdel

Asistenta: Matevž Jekovec, dr. Gašper Fele-Žorž

Uvod

Pri predmetu Spletno programiranje boste tekom semestra razvijali lastno spletno aplikacijo. Vaše delo bo sestavljeno iz dveh faz – brskalniški del in zaledni del. Pri predmetu zahtevamo in preverjamo sprotno delo!

V prvi fazi boste (I) pripravili razvojno okolje in zasnovali spletišče, (II) dodali vsebino strani v HTML in CSS, ter (III) dodali interaktivnost z uporabo JavaScripta.

Razvojno okolje in zasnova spletišča

Git

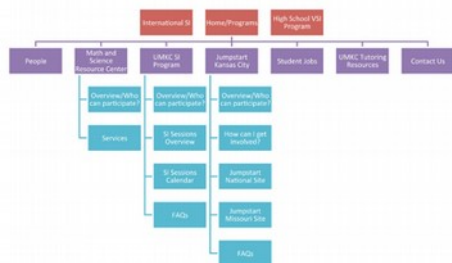
Pri razvoju morate uporabljati porazdeljen sistem za sledenje spremembam, git. Vzpostavite lasten repozitorij ali uporabite storitev, kot je npr. github.com, bitbucket.com in podobno. URL do repozitorija oddajte na učilnici (<https://ucilnica.fri.uni-lj.si/mod/assign/view.php?id=16156>) do **nedelje, 23. oktobra**. Pri predmetu zahtevamo sprotno delo. Izvajalci imamo skripto, ki tedensko preverja vaše spremembe na git. V primeru, da ne bo vidne vsaj ene smiselne spremembe na teden, vam bomo znižali oceno zagovora za 5 pik na vsak neaktiven teden, če za to nimate opravičljivega razloga.

Izbira teme

Za projekt si na učilnici izberite eno od predlaganih tem (<https://ucilnica.fri.uni-lj.si/mod/page/view.php?id=1436>). Temo lahko predlagate tudi sami, vendar vam jo mora v tem primeru odobriti asistent. Predlagamo, da si zamislite aplikacijo za vaše startup podjetje, nekaj, s čimer bi pritegnili investitorje. Pri predmetu ne bomo ocenjevali le tehnologije, ampak tudi prepričljivost in smiselnost vaše ideje. V datoteki README.md vašega projekta podajte opis aplikacije z izbrano temo (kakšen problem rešujete, kdo je ciljna publika, na katerih napravah bo aplikacija podprta itd.).

Zemljevid in struktura spletišča

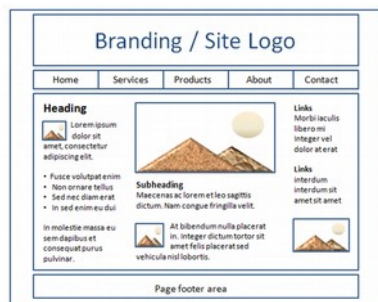
Odločite se za način organizacije spletišča (hierarhična, linearna, naključna). Lahko uporabite tudi kombinacijo treh osnovnih načinov. Spomnite se na lastnosti posamezne organizacije, ki ste jih povedali na predavanjih in obrazložite, zakaj ste izbrali način, ki ga predlagate. Pazite, da ne bodo organizacije kršile osnovnih načel preglednosti spletišč. Narišite skico za navigacijo po straneh. Primer:



Premislite o navigacijskih orodjih spletne strani oz. kje se bodo nahajali meniji in kakšni bodo. Spletna aplikacija mora brez težav delovati tudi na napravah z zasloni na dotik. To pomeni, da ne morete prikazovati in skrivati npr. menijev ob prehodu z miškinim kazalcem, ampak morate najti drugačno rešitev. Načrt spletišča oddajte v datoteko docs/SITEMAP z ustrezno končnico.

Žični okvirji

Za ponazoritev strukture spletne strani uporabite žične okvirje. Primer:



Za tovrsten izris lahko uporabite splošno-namenske programe za vektorsko risanje ali pa orodja, namenjena izključno risanju tovrstnih diagramov. Priporočamo:

- spletno orodje Creately (<http://creately.com/diagram-type/site-map>, za izvoz potrebna brezplačna registracija),
- odprtokodno namizno aplikacijo Evolus Pencil (<http://pencil.evolus.vn>),
- odprtokodni Inkscape,
- Adobe Illustrator,
- Microsoft Visio.

Žične okvirje ustvarite za vse tipske strani vašega spletišča in jih shranite v docs/wireframes/ v zapisu PDF.

Izdelava statičnih strani HTML/CSS

Ko boste imeli izdelan načrt, se lotite izdelave spletnih strani. Če uporabljate slikovni material, ki ni vaš, preverite, da ne kršite avtorskih pravic. Strukturo in vsebino spletišča izdelajte z uporabo HTML. Oblikovanje pa boste izvedli s pomočjo CSS v ločenih datotekah. Za obrobe, prehode, okvirje itd. poskusite uporabiti izključno kombinacijo HTML/CSS brez zunanjih slikovnih datotek. Če določenega efekta na ta način res ni možno doseči, narišite gradnik v programu za vektorsko risanje, ga razrežite in izvozite posamezne dele elementov kot slike. Dodatne točke dobite, če združite vse koščke slik v eno slikovno datoteko (*sprites*) in s tem prihranite število zahtevkov po slikah na strežniku.

HTML 5

Ker ima vsaka izmed tem projektov svojo tematiko, od vas pričakujemo, da izberete in vključite spodaj našteje elemente po vaši presoji o smiselnosti le-tega. Vaša izvirnost in iznajdljivost se bosta upoštevala pri oceni. Pričakujemo, da bo spletišče vključevalo naslednje gradnike:

- Osnovni elementi (HTML 1.0)
 - temeljni elementi (<p>, <h1>,
, ...),
 - povezave (<a>),
 - slike,
 - sezname (, , ; <dl>, <dt>, <dd>), ki naj za znak alinee uporabljajo lastne slike in posebej določeno številčenje;
 - tabele (<table>, <caption>, <tr>, <th>, <td>, <thead>, <tbody>, <tfoot>). Tabele naj imajo ločeno oblikovane regije glave, telesa in noge (<thead>, <tbody>, <tfoot>), uporabite pa tudi atributa colspan in rowspan.
 - elementi za opisovanje regij dokumenta <div> in <section>.
- Obrazci (<form>, <label>, <input>, <textarea>, <select>, <option>)
Definirajte obrazec z vnosnimi polji/kontrolami, ki so primerni za vašo temo: vnosno polje, geslo, potrditveno polje (checkbox), izbirno polje (radio button), polje z besedilom, izbira, gumb potrdi, gumb reset, e-mail ipd.
- Elementi iz HTML5:
 - kontrolni elementi obrazcev (<color>, <date>, <email>, <month>, <number>, <range>, <search>, <tel>, <time>, <url>, ...),
 - multimedijski elementi (video, predvajalnik zvoka, animacijo, ...),
 - elementa <input> in <datalist>,
 - napredni atributi (placeholder ...).

V stran vnesite tipično vsebino, ne oddajajte strani brez vsebine!

Validacija uporabniških vnosov

Poskrbite za validacijo morebitnih vnosov uporabnika. Zanj uporabite regularne izraze in preprečite oddajo obrazca v primeru neuspešne validacije. Nekaj idej: pri registraciji lahko npr. preverite, če sta obe vtiskani gesla pravilni; validirajte podatke o uporabniku, izbor možnosti iz zaprtega nabora možnih vrednosti ipd.

CSS

Spomnite se na principe oblikovanja: ponavljanje, kontrast, bližina, poravnava. Stran oblikujte tako, da zagotovite dostopnost (npr. za slabovidne, za barvno slepe). Izberite barve, ki bodo ustrezale ciljni publiki in ki se bodo med seboj ujemale. Uporabite enotno barvno kompozicijo za vse strani. Premislite o postavitvi slikovnega materiala, o velikosti komponent. Izberite pisavo, ki bo primerna in dovolj razločna za branje. Spomnite se na pravila oblikovanja »easy to read« besedila. Če imate slikovne elemente netipične oblike, jih razrežite in vstavite v ustrezne gradnike na strani.

Držite se dobrih programerskih praks: CSS naj počiva v ločenih datotekah, v spletno stran pa ga vključite z značko <link>; za naslavljanje elementov uporabljajte razrede, naslavljanje prek ID-ja naj bo prisotno le izjemoma; ko je primerno, dedujte attribute posameznih razredov; veliko razredov z veliko atributi nadomestite z manj razredi z manj atributi tako, da elementi HTML uporabijo slog več razredov hkrati.

Pri oblikovanju upoštevajte naslednje zahteve:

- strukturo spletne strani in menije oblikujte s pomočjo CSS,
- uporabite različne načine pozicioniranja (absolutno in relativno),

- besedilo naj bo primerno oblikovano glede na vsebino (barva, pisava, velikost, dekoracija teksta, krepko, ležeče, zamiki, ...), uporabite <div> in značke za urejanje različnih delov besedila,
- v tabelah oblikujte vsak del posebej (glavo, telo in nogo). Ne pozabite oblikovati tako pisave, kot tudi robov,
- uporabite napredne zmožnosti standarda CSS3 (zaobljene robove, gradientna polnila, prehode, sence ipd.),
- elemente lahko naslavljajte po vaši izbiri, uporabite tudi oblikovanje z razredi.

Validacija dokumentov HTML in CSS

S pomočjo validatorjev W3C validirajte pravilnost oblikovanja vseh vaših dokumentov CSS in HTML. Posnetek zaslona (screenshot) uspešne validacije vseh vaših dokumentov vključite v končno oddajo. Slika naj obsega le zaglavje izpisa validatorja (brez morebitnih podrobnih komentarjev na dnu). Lahko uporabite:

- http://validator.w3.org/#validate_by_upload+with_options in
- http://jigsaw.w3.org/css-validator/#validate_by_upload

Preverjanje izgleda

Preverite delovanje strani v vsaj treh brskalnikih, npr. Firefox, Chrome ali Safari, Opera, Microsoft Edge. V poročilu poročajte o morebitnih različnih izgledih strani med njimi. Ne ukvarjajte se z zastarelimi različicami brskalnikov (npr. Internet Explorer 6/7/8), ampak primerjajte zadnje različice med seboj različnih brskalnikov.

Responsive design

Vaša aplikacija mora delovati na fizično različno velikih napravah (telefoni, tablice, namizni računalniki). Poskrbite, da se gradniki v aplikaciji ustrezno prilagodijo velikosti zaslona. Na primer vsebina menija postane skrita in dostopna z dodatnim klikom; elementi, razvrščeni po širini postanejo razvrščeni po višini itd. Tovrstno obnašanje lahko dosežete med gledanjem strani brez uporabe JavaScript.

Netipično oblikovane strani

Sodobni brskalniki podpirajo tehnologije HTML5 Canvas, SVG in WebGL. Če menite, da bo vaša spletna aplikacija bolje sprejeta v ciljni publiki, če uporabite »drznejši« dizajn, se lahko lotite spletišča drugače. Nekatera orodja (npr. Flash Professional, iioengine, animatron, google web designer) že omogočajo samodejen izvoz v omenjene tehnologije. Tovrstna orodja lahko uporabite, vendar ne za celotno spletišče. Npr. lahko naredite del aplikacije ročno (registracija uporabnika, novice, no-javascript del), jedro aplikacije (spletna igra, izobraževalna aplikacija) pa ustvarite v omenjenih orodjih.

Dinamičnost na strani odjemalca z JavaScript

Na strani odjemalca za izboljšanje uporabniške izkušnje uporabite JavaScript. V prvi fazi **ni dovoljena** uporaba dodatnih JavaScript knjižnic (jQuery, Bootstrap itd.).

Skriptno kodo ločite od dokumentov HTML. Držite se dobre programerske prakse, to je smiselno imenovanje spremenljivk in funkcij, zamikanje, kratke funkcije, ponovna uporabljivost, modularnost.

Oživitev uporabniškega vmesnika

Nadgradite množico kontrolnih elementov na spletni strani (gumbob, vnosnih polj itd.) in napišite kodo, ki bo reagirala na dogodke v brskalniku (aktivnosti uporabnika). Glede na elemente poznamo različne dogodke (blur, change, click, focus, select, load itd.). Dodajte in napišite takšne rokovalnike dogodkov, ki bodo vezani tako na kontrolne elemente kot tudi na običajne elemente v strukturi HTML (npr. premik miške preko tabele ali menija).

Z interakcijo skriptne kode z lastnostmi CSS posameznih elementov povečajte dinamičnost vašega spletnega vmesnika: dodajte animacije, dinamično pozicioniranje in prekrivanje elementov, spreminjajte vsebino kode HTML, barve, pisave, ozadje itd.

Glede na interakcijo z uporabnikom dinamično ustvarite del spletne strani z izvedbo metod, ki izvajajo operacije nad DOM stopnje 2 (tipična je uporaba vgrajenih metod `createElement`, `appendChild`, `removeChild`, `insertBefore`, `parentNode`, `previousSibling`, `nextSibling` itd.). Primer: v spletni trgovini lahko dinamično ustvarite del strani, ki obsega artikole, ki jih je uporabnik izbral.

Kako točno boste uporabili v vaši konkretni aplikaciji, je prepuščeno vam. Lahko se osredotočite le na dinamičnost spletne strani glede na specifično uporabnikovo akcijo (npr. skrivanje, prikazovanje elementov), lahko pa greste tako daleč, da implementirate lastne grafične gradnike, kot to sicer počnejo JavaScript knjižnice z gradniki (jQuery UI, Dojo, Bootstrap itd.).

Težave z združljivostjo različnih brskalnikov

Pri izdelavi HTML5 in CSS ste verjetno ugotovili, da pri uporabi različnih pogonov brskalnikov nastopajo težave. Podobno kot pri HTML/CSS poskrbite, da bo JavaScript koda delovala z vsaj tremi različnimi pogoni brskalnikov.

Websocket

Protokol HTTP je t.i. *stateless*. To pomeni, da se vzpostavi povezava do strežnika, izvede se poizvedba, strežnik odgovori in povezavo se zapre. Tehnologija *websocket* vam omogoča, da vaš brskalnik vzporedno vzpostavi dvosmerno sejo s strežnikom, ki ostane aktivna, dokler npr. ne zapustite spletnega mesta. Tako vas ob dogodkih (npr. nekdo vam pošlje sporočilo) lahko pozove strežnik in ni potrebno neprenehno preverjati po novih sporočilih (*polling*) z vaše strani. Tehnologijo npr. uporabljata Facebook in Gmail za hipno sporočanje med uporabniki v brskalniku.

Zajem zvoka, videa, lokacije na strani odjemalca

Sodobni brskalniki omogočajo standardiziran način zajema zvoka, videa in lokacije. Zaželeno je, da poiščete mesto za omenjene tehnologije v vaši aplikaciji. Kot primer lahko preizkusite odprtokodno spletno aplikacijo za konferenčne klice, ki deluje izključno v brskalniku <https://meet.jit.si>.

Podpora okoljem no-javascript

Kvalitetne spletne strani imajo pripravljen tudi scenarij za okolja, ki ne podpirajo skriptnega jezika JavaScript (pajki, varnostno paranoični uporabniki). Poskrbite, da je temeljna funkcionalnost strani (npr. oddaja naročila v spletni trgovini) zagotovljena v okoljih brez JavaScript.

Testiranje

Spletno stran temeljito preverite tako vsebinsko (nedelujoče povezave), oblikovno (napačen razrez, premajhni gradniki za različne napise) in funkcionalno (slabo preverjanje vnosnih polj, nedelovanje rokovalnih funkcij).

Objava vaše spletne aplikacije

Za objavo statičnih strani za potrebe prve faze lahko uporabite kar Github pages. Za dinamično generirano vsebino (druga faza) potrebujete oddaljen strežnik tipa PaaS (Platform as a Service). Lahko izberete enega izmed PaaS ponudnikov ali postavite lasten spletni strežnik. Večina PaaS ponudnikov ponuja eno od vrst brezplačnega gostovanja. Izbirajte tistega, ki razen registracije ne zahteva drugih podatkov od vas (npr. <http://www.heroku.com>). Nekateri ponudniki namreč vežejo registracijo novega uporabnika z vpisom veljavne kreditne kartice, drugi, predvsem manjši, pa to zahtevajo šele potem, ko začnete resno uporabljati njihovo infrastrukturo.

Končni izdelek prve faze

Končni izdelek oddajte v zapisu ZIP, zgeneriranem iz repozitorija git. Ta naj vsebuje:

- Poročilo o izdelavi spletne strani v docs/README.md, ki zajema:
 - opis vaše aplikacije,
 - ciljno publiko in naprave,
 - poročilo o težavah v različnih brskalnikih,
 - 2 zmogljivosti ali gradnika spletne strani, v katere ste vložili poseben trud in ste nanju ponosni,
 - morebitne komentarje o izdelku in morebitne probleme, ki jih je potrebno v vaši implementaciji še rešiti.
- Ostalo dokumentacijo projekta v mapi docs/ :
 - Zemljevid spletišča
 - Žični okvirji tipskih strani
 - Posnetek zaslona validatorjev za vse datoteke HTML in CSS.
- Datoteke spletne strani (HTML, CSS, JavaScript, slikovni material), smiselno organizirane po mapah

Datoteko oddate na učilnico (<https://ucilnica.fri.uni-lj.si/mod/assign/view.php?id=1443>).

Kriteriji za ocenjevanje

V prvi fazi je možno doseči 100 točk in 25 točk bonusa, ki se lahko prenesejo v drugo fazo. Pogoji za uspešno opravljanje vaje je v povprečju zbranih vsaj 50 točk pri obeh fazah (pri posamezni fazi lahko zberete tudi manj kot 50, vendar mora biti zato preostala faza toliko boljše narejena).

Ocenjevali bomo naslednja področja:

Utemeljitev aplikacije, opis ciljne publike in naprav	10
Zemljevid strani	5
Žični okvirji	5
Smiselna strukturiranost kode HTML	10
Smiselna strukturiranost kode CSS	10
Uporaba gradnikov HTML 5 in funkcionalnosti CSS 3	5
Uspešno validiran HTML in CSS	5
JavaScript za izboljšanje uporabniške izkušnje	5
Delovanje in izgled strani v vsaj 3 brskalnikih	5

Dobra programerska praksa in kvaliteta kode	10
Dobro testirana in delujoča spletna stran	5
Delujoč <i>responsive design</i> v vsaj treh velikostih (npr. 5", 9", 14")	10
Lasten grafični gradnik z uporabo HTML+CSS+JavaScript	10
Postavitev lastnega spletnega strežnika, ki gostuje vašo spletišče	5
Uporaba HTML5 Canvas ali SVG za ogrodje strani, WebGL	10
Zajem audio, video, GPS	5
Podpora no-javascript	5
Websockets	5

Viri

Pri izdelavi si pomagajte s predlagano literaturo predmeta, tu pa je še nekaj povezav:

- Knjiga o sistemu Git (<https://git-scm.com/book/sl>)
- HTML5 (http://www.w3schools.com/html/html5_intro.asp)
- HTML5 reference (<http://www.w3schools.com/tags/default.asp>)
- CSS (<http://www.w3schools.com/css/>)
- CSS reference (<http://www.w3schools.com/cssref/default.asp>)
- JavaScript - (<http://www.w3schools.com/js/default.asp>)
- JavaScript - (<http://www.codecademy.com/en/tracks/javascript>)