

Projekt – 1. faza:

Ogrodje in organizacija spletišča

Statična stran HTML/CSS

Dinamičnost z JavaScript

Spletno programiranje

Oktober 2015

Rok oddaje na učilnici: 22.11.2015, možni (in priporočeni) predhodni zagovori.

Predavatelj: Aleš Smrdel

Asistenta: Matevž Jekovec, Žiga Emeršič

Uvod

Pri predmetu Spletno programiranje boste skozi celoten semester razvijali lastno spletno aplikacijo, katere razvoj bo sestavljen iz dveh faz. V prvi fazi boste razvijali statično spletno stran, ki zaenkrat ne bo imela posebnih funkcionalnosti, predstavljala pa bo osnutek za nadaljnji razvoj spletne aplikacije. Osredotočili se boste predvsem na oblikovanje spletne strani in izbor potrebne vsebine, da bo nato spletna stran v naslednji fazi projekta lahko služila svojemu namenu.

Prva faza bo sestavljena iz treh delov, in sicer iz (I) načrtovanja spletne strani, (II) same izdelave strani s tehnologijami HTML in CSS, in (III) dodajanja interaktivnosti obstoječemu spletišču z uporabo JavaScript-a na strani odjemalca.

Da boste lahko začeli z delom na projektu, si morate najprej na spletni učilnici (<https://ucilnica.fri.uni-lj.si/course/view.php?id=74>) izbrati temo projekta. Če vam nobena izmed tem ni posebej všeč ali če ste bolj kreativni, sami predlagajte temo (za ta namen uporabite možnost na učilnici in počakajte na potrditev, ali je tema sprejeta). Podajte opis, ki bo predstavil namen in delovanje spletne strani.

Načrtovanje spletne strani

Preden boste začeli z izdelavo spletne strani, je potrebno izdelati njen načrt. Najprej potrebujete problem, nato idejo, kako ta problem rešiti, potem pa poiščite način, kako rešitev pretvoriti v aplikacijo. Priporočamo, da razmišljate v smeri, da bo po eni strani vaša aplikacija temeljni vir prihodka vašega startup podjetja in po drugi, da imate vizijo za nadaljnji razvoj, s katero boste pritegnili investitorje. Pri predmetu ne bomo ocenjevali le tehnologije, ampak tudi prepričljivost in smiselnost vaše ideje. Ko bo ta dozorela, se lotite izdelave načrta spletne strani. Zapišite ga v obsegu največ 2 A4 strani, ki bo del vašega izdelka, ta pa naj vsebuje točke, zapisane v nadaljevanju:

Vaši podatki

Ime, priimek, vpisno številko, študijsko leto, ime predmeta.

Ime in namen spletne strani

Določite ime spletne strani (aplikacije) in zapišite njen kratek opis. Odgovorite na vprašanja, ki se osredotočajo na to, kaj je bistvo spletne aplikacije, kakšno uporabniško izkušnjo želi zagotoviti, kakšen problem aplikacija rešuje.

Opis ciljne publike in naprav

Premislite, kdo je ciljna publika vaše spletne strani, kakšne so njihove potrebe in želje, ter na kakšnih napravah bodo vašo aplikacijo uporabljali. Poiščite vsaj dve spletni strani, ki ciljata na podobno publiko in potencialno predstavljata vašo konkurenco. Na podlagi ugotovljenega zapišite seznam elementov izvedbe, ki jim bo potrebno posvetiti posebno pozornost ter podajte rešitev, kako jo boste implementirali v vaši aplikaciji (primer: denimo, da stran namenite otroški populaciji: temu ustrezno boste zapisali, kakšen bo vaš nabor izbranih barv, večpredstavnost, ločljivost ipd.)

Zemljevid in struktura spletišča

Odločite se za način organizacije spletišča (hierarhična, linearna, naključna), ki je lahko tudi kombinacija 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. 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. Za ponazoritev strukture spletne strani uporabite žične okvirje in skicirajte vsaj dve tipični strani spletišča.

Primer žičnega okvirja:



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.

Oblikovanje

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. Oblikujte tipski strani, za kateri ste v prejšnjem koraku zasnovali žična okvirja. Uporabite splošno-namenska orodja za vektorsko risanje.

Izdelava statičnih strani HTML/CSS

Ko boste imeli izdelan podroben načrt, se lotite izdelave spletnih strani. Če uporabljate slikovni material, ki ni vaš, preverite, da ne kršite avtorskih pravic. Spletno stran izdelajte z uporabo označevalnega jezika HTML. Oblikovanje spletne strani pa boste izvedli s pomočjo jezika CSS v ločeno datoteko. Poskusite izdelati strani, ki se čim bolj ujemata z vašo grafično podobo iz prejšnjega poglavja. Za obrobe, prehode, okvirje itd. poskusite uporabiti izključno kombinacijo HTML/CSS. Če določenega efekta na ta način ni možno doseči in menite, da je nujen, naredite razrez v programu za vektorsko risanje in izvozite posamezne dele elementov kot slike. Dodatne točke dobite, če združite vse koščke slik v eno slikovno datoteko (ang. *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 boste vključili:

- Osnovne elemente (HTML 1.0)
 - temeljne elemente (<p>, <h1>,
, ...),
 - povezave (<a>),
 - slike,
 - sezname (, , ; <dl>, <dt>, <dd>), ki naj za znak alineje 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.
- elemente za opisovanje regij dokumenta <div> in <section>.
- Obrazce (<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.
- Elemente iz HTML5:
 - kontrolne elemente obrazcev (<color>, <date>, <email>, <month>, <number>, <range>, <search>, <tel>, <time>, <url>, ...),
 - multimedijски element (video, predvajalnik zvoka, animacijo, ...),
 - elementa <input> in <datalist>,
 - napredne attribute (placeholder ...).

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

CSS

V načrtu spletne strani ste opisali tudi, kako boste oblikovali spletno stran. Celotno spletno stran (vse elemente, ki jih naslovite bodisi preko razredov, identifikatorjev ali drugače) oblikujte s pomočjo CSS.

Držite se dobrih programerskih praks: CSS naj počiva v ločeni datoteki, 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 naslavljanje 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 IE in 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.

Objava spletne aplikacije v oblaku

Aplikacijo boste naložili tudi v oblak tipa PaaS (Platform as a Service). Večina ponudnikov PaaS ponuja eno od vrst brezplačnega gostovanja za vaše majhne (ali niti ne tako majhne) projekte. Izbirajte ponudnike, ki razen registracije ne zahtevajo 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 tega ne zahtevajo za zastojno uporabo ampak šele potem, ko začnete resno uporabljati njihovo infrastrukturo.

Na spletni učilnici (<https://ucilnica.fri.uni-lj.si/mod/glossary/view.php?id=1439>) je zbranih nekaj navodil, kako spraviti projekt v določene oblake tipa PaaS iz prejšnjih let. Seveda pa lahko svoje zapiske (navodila), kako ste svoj projekt spravili v oblak objavite tudi vi. Poleg neskončne hvaležnosti in dviga statusa med kolegi se aktivno sodelovanje pri predmetu nagraduje z dodatnimi točkami na koncu semestra za najbolj prizadevne.

Objava v oblaku je običajno narejena posredno prek sistema za vodenje verzij GIT ali SVN. Glede na to, da ste dobri v programiranju, se pričakuje, da ste s tovrstnimi orodji seznanjeni in jih tudi sicer redno uporabljate.

Programska koda na strani odjemalca

Uporabite programsko kodo na strani odjemalca, da oživite vaše spletne strani. Kako boste to naredili za vašo konkretno aplikacijo, je prepuščeno vam.

Brskalniki imajo vgrajeno podporo za programski jezik JavaScript. Priporočamo uporabo ene izmed knjižnic jQuery (jQueryUI), Dojo (DojoX) in drugih knjižnic, ki olajšajo vrsto opravil in prinašajo dodatne grafične gradnike.

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. Uporabite lahko tudi efekte, ki vam jih ponujajo knjižnice, kot je jQuery ali Dojo.

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 artikla, ki jih je uporabnik izbral.

Težave z združljivostjo različnih brskalnikov

Pri izdelavi HTML5 in CSS ste verjetno ugotovili, da pri uporabi različnih pogonov brskalnikov nastopajo težave. Nezdržljive elemente in skriptno kodo običajno nadomestimo z delujočimi ekvivalenti iz knjižnic, omenjenih v prejšnjem poglavju. Uporabite lahko npr. knjižnico Modernizr (<https://modernizr.com>), ki samodejno zazna podprte značilnosti brskalnika. Podobno kot pri HTML/CSS poskrbite, da bo JavaScript

koda delovala z vsaj tremi različnimi pogoni brskalnikov. Ugotovitve o delovanju v različnih brskalnikih zapišite v vaše končno poročilo.

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 vtipkani gesli pravilni; validirajte podatke o uporabniku, izbor možnosti iz zaprtega nabora možnih vrednosti ipd.

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).

Dodatne zahteve (pogoj za oceno 9 in 10)

Netipično oblikovane strani

Želimo nadomestiti tehnologijo Flash. 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.

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>.

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.

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. Če uporabljate enega izmed novjših Javascript ogrodij (Angular, Bootstrap, NodeJS), boste morali zagotoviti ustrezno zamenjavo (*fallback*).

Websocket

Protokol HTTP je t.i. *stateless*. To pomeni, da vzpostavite povezavo s strežnikom, ga povprašate po vsebini, ta vam odgovori in povezavo 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.

Končni izdelek prve faze

Kot končni izdelek oddajte v arhivski datoteki ZIP. Ta naj vsebuje:

- Poročilo o izdelavi spletne strani v zapisu PDF v mapi doc/:
 - vaše podatke,
 - ime in opis aplikacije,
 - povezavo na vaš projekt v oblaku,
 - ciljna publika in naprave,
 - poročilo o težavah v različnih brskalnikih,
 - 2 elementa spletne strani, v katera 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 doc/ :
 - Zemljevid spletišča v zapisu PDF
 - Žični okvirji vsaj dveh tipskih strani v zapisu PDF
 - Grafična podoba vsaj dveh tipskih strani v zapisu PDF
 - Posnetek zaslona validatorjev za vse datoteke HTML in CSS.
- Datoteke spletne strani (HTML, CSS, JavaScript, slikovni material), smiselno organizirane po mapah

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 opravljene vaje je vsaj 50 točk pri vsaki od faz.

Obvezne zahteve za uspešno opravljeno prvo fazo so naslednje:

Utemeljitev aplikacije, opis ciljne publike in naprav	10
Zemljevid strani	4
Žični okvirji vsaj dveh tipskih strani	4
Grafična podoba vsaj dveh tipskih strani	4
Smiselna strukturiranost kode HTML	6
Smiselna strukturiranost kode CSS	6
Uporaba gradnikov HTML 5, funkcionalnosti CSS 3 in integracija nestandardnih grafičnih gradnikov (npr. zemljevidi)	4
Uspešno validiran HTML 5, CSS	4
JavaScript za izboljšanje uporabniške izkušnje	5
JavaScript pri validaciji obrazcev	5
Delovanje in izgled strani v vsaj 3 brskalnikih	5
Dobra programerska praksa in kvaliteta kode	10
Redna uporaba GIT ali SVN pri razvoju projekta	4
Objava strani v oblaku	4

Dodatne zmožnosti aplikacije, zahtevane za nadpovprečno opravljene vaje (ocena 9 in 10) ter bonus točke:

Uporaba HTML5 Canvas ali SVG za ogrodje strani, WebGL	10
Zajem audio, video, GPS	10
Delujoč <i>responsive design</i> v vsaj treh velikostih (npr. 5", 9", 14")	10
Podpora no-javascript	10
Smiselna uporaba websockets	10

Viri

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

- 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>
- jQuery - <http://www.w3schools.com/jquery/default.asp>