Predgovor

Kombinacija PHP-a i MySQL-a je najbolji pristup razvoj dinamičkih veb stranica zasnovanih na bazama podataka, koji se i dalje dobro nosi sa izazovima novih integrisanih radnih okvira (eng. *framewors*), koje je teže naučiti. Zbog svojih korena u otvorenom kodu, besplatna je za primenu i stoga je izuzetno popularna opcija za veb razvoj.

Svaki potencijalni programer na Unix/Linux ili čak Windows platformi ima potrebu da ovlada ovim tehnologijama. U kombinaciji sa tehnologijama JavaScript, React, CSS i HTML5 moći ćete da stvarate veb lokacije kalibra industrijskih standarda poput Facebooka, Twittera i Gmaila.

Publika

Ova knjiga je za one koji žele da nauče kako da prave efikasne i dinamičke veb stranice. Ovo može uključivati vebmastere ili grafičke dizajnere koji su već savladali izradu statičkih veb lokacija ili rad sa CMS-om kao što je WordPress, ali žele da svoje veštine podignu na viši nivo, kao i srednjoškolce i studente, nedavno diplomirane studente i samouke pojedince.

U stvari, svako ko je spreman da nauči osnove responzivnog veb dizajna steći će temeljnu osnovu srži tehnologija PHP-a, MySQL-a, JavaScripta, CSS-a i HTML5, a naučićete i osnove biblioteke Reacta.

Pretpostavke koje koristi knjiga

Ova knjiga pretpostavlja da imate osnovno razumevanje HTML-a i da možete sastaviti barem jednostavnu, statičku veb lokaciju, ali ne pretpostavlja da imate predznanje o PHP-u, MySQL-u, JavaScriptu , CSS ili HTML5 – ali ako imate, vaš napredak kroz knjigu će biti još brži.

Organizacija knjige

Poglavlja u ovoj knjizi napisana su određenim redosledom, prvo uvodeći sve osnovne tehnologije koje pokriva, a zatim vas provodi kroz njihovu instalaciju na serveru za veb razvoj, da biste bili spremni da prođete kroz primere.

U prvom odeljku ćete steći utemeljenje u programskom jeziku PHP, pokrivajući osnove sintakse, nizova, funkcija i objektno orijentisanog programiranja.

Sa znanjem PHP-a, preći ćete na uvod u sistem baza podataka MySQL, gde ćete naučiti sve, od načina na koji su MySQL baze podataka strukturisane, do generisanja složenih upita.

Nakon toga ćete naučiti kako možete kombinovati PHP i MySQL da biste započeli pravljenje sopstvenih dinamičkih veb stranica integrisanjem obrazaca i drugih HTML funkcionalnosti. Zatim ćete preći na praktične aspekte razvoja PHP-a i MySQL-a tako što ćete naučiti razne korisne funkcije i kako upravljati kolačićima i sesijama, kao i kako održavati visok nivo bezbednosti.

U narednih nekoliko poglavlja steći ćete temeljnu osnovu o JavaScriptu, od jednostavnih funkcija i upravljanja događajima do pristupa objektnom modelu dokumenta, provere u veb čitaču i rukovanja greškama. Dobićete i opsežan primer korišćenja popularne React biblioteke za JavaScript.

Sa razumevanjem ove tri osnovne tehnologije, naučićete kako da šaljete Ajax pozive iza scene i pretvarate svoje veb lokacije u visoko dinamičko okruženje.

Zatim ćete provesti dva poglavlja učeći o korišćenju CSS-a za oblikovanje i postavljanje svojih veb stranica, pre nego što otkrijete kako vam React biblioteke mogu znatno olakšati razvojni posao. Zatim ćete preći na poslednji odeljak o interaktivnim funkcijama ugrađenim u HTML5, uključujući geolociranje, audio, video i platno. Nakon toga ćete sve što ste naučili spojiti u kompletan skup programa koji zajedno čine potpuno funkcionalnu veb stranicu društvene mreže.

Usput ćete pronaći mnogo saveta o dobroj programerskoj praksi i saveta koji vam mogu pomoći da pronađete i rešite teško uočljive programske greške. Takođe dato je mnogo linkova do veb stranica koje sadrže dodatne detalje o obrađenim temama.

Konvencije korišćene u ovoj knjizi

U ovoj knjizi se koriste sledeće tipografske konvencije:

Ravan tekst
: Označava naslove menija, opcije i dugmad.

Kurziv
: Označava nove termine, URL-ove, adrese e-pošte, imena datoteka, tipove datoteka, putanje, direktorijume i Unix pomoćne programe. Takođe se koristi za imena baza podataka, tabela i kolona.

`Konstantna širina`
: Označava komande i opcije komandne linije, promenljive i druge elemente koda, HTML oznake i sadržaj datoteka.

`Konstantna širina podebljana`
: Prikazuje izlaz, tj. ispis programa i koristi se za označavanje delova koda o kojima se govori u tekstu.

Sadržaj

Predgovor ..... **xxiii**

1. Uvod u dinamički veb sadržaj **1**
HTTP i HTML: Osnove Berners-Leea 2
Procedura zahtev/odgovor 2
Prednosti PHP-a, MySQL-a, JavaScripta, CSS-a i HTML5-a 5
MariaDB: Klon MySQL-a 5
Korišćenje PHP-a 6
Korišćenje MySQL-a 6
Korišćenje JavaScript-a 8
Korišćenje CSS-a 9
I evo ga HTML5 9
Apache veb server 10
Rukovanje mobilnim uređajima 11
O otvorenom kodu 11
Sve to zajedno 11

2. Postavljanje razvojnog servera **15**
Šta je VAMP, MAMP ili LAMP? 15
Instaliranje AMPPS-a na Windows 16
Testiranje instalacije 20
Pristup korenu dokumenta (Windows) 21
Alternativni WAMP-ovi 23
Instaliranje AMPPS-a na macOS 24
Pristupanje korenu dokumenta (macOS) 24
Instaliranje LAMP-a na Linux-u 25
Rad na daljinu 26
Prijavljivanje 26
Korišćenje SFTP ili FTPS 26
Korišćenje editora koda 27
Korišćenje IDE-a 28
Pitanja 30

3. Uvod u PHP 31
Uključivanje PHP-a u HTML 31
Primeri ove knjige 32
Struktura PHP-a 33
Korišćenje komentara 33
Osnovna sintaksa 34
Promenljive 34
Operatori 39
Dodeljivanje promenljive 42
Višeredne komande 44
Tipiziranje promenljivih 46
Konstante 47
Unapred definisane konstante 47
Razlika između naredbi echo i print 48
Funkcije 49
Promenljiv opseg 49
Pitanja 54

4. Izrazi i kontrola toka u PHP-u 57
Izrazi 57
TRUE ili FALSE? 57
Literali i promenljive 58
Operatori 59
Prioritet operatora 60
Asocijativnost 62
Operatori relacije 63
Uslovi 67
Izjava if 67
Naredba else 68
Naredba elseif 70
Naredba switch 71
Trostruki ? operator 73
Petlja 74
Petlje while 75
Petlje do...while 77
Petlje for 77
Izlaz iz petlje 79
Naredba continue 80
Implicitno i eksplicitno pretvaranje 80
PHP dinamičko povezivanje 81
Dinamičko povezivanje na delu 82
Pitanja 83

5. PHP funkcije i objekti 85
PHP funkcije 86
Definisanje funkcije 87
Vraćanje vrednosti 88
Vraćanje niza 89
Prenošenje argumenata po referenci 90
Vraćanje globalnih promenljivih 91
Rezime opsega promenljivih 92
Datoteke iclude i require 92
Naredba include 92
Korišćenje include_once 93
Korišćenje require i require_once 93
Kompatibilnost PHP verzija 93
PHP objekti 94
Terminologija 95
Deklarisanje klase 96
Stvaranje objekta 97
Pristup objektima 97
Kloniranje objekata 98
Konstruktori 99
Destruktori 100
Metode upisivanja 100
Deklarisanje svojstava 101
Deklarisanje konstanti 102
Svojstvo i opseg metoda 102
Statičke metode 103
Statička svojstva 104
Nasleđivanje 105
Pitanja 108

6. PHP nizovi.....109
Osnovni pristup 109
Numerički indeksirani nizovi 109
Asocijativni nizovi 111
Dodela pomoću rezervisane reči array 111
Foreach...as petlja 112
Višedimenzionalni nizovi 114
Korišćenje funkcija niza 117
is_array 117
count 118
sort 118
shuffle 118
explode 118

extract 119
compact 120
reset 121
end 121
Pitanja 122

7. PHP u praksi ... 123

Korišćenje printf 123
Zadavanje preciznosti 124
Nadopunjavanje stringova 125
Korišćenje sprintf 127
Funkcije datuma i vremena 127
Konstante datuma 129
Korišćenje checkdate 129
Rukovanje datotekama 130
Provera da li datoteka postoji 130
Stvaranje datoteke 130
Čitanje iz datoteka 132
Kopiranje datoteka 133
Premeštanje datoteke 133
Brisanje datoteke 133
Ažuriranje datoteka 134
Zaključavanje datoteka za višestruki pristup 135
Čitanje cele datoteke 136
Otpremanje datoteka 137
Sistemski pozivi 142
XHTML ili HTML5? 143
Pitanja 144

8. Uvod u MySQL ... 145

Osnove MySQL-a 145
Rezime termina baze podataka 146
Pristupanje MySQL-u preko komandne linije 146
Pokretanje interfejsa komandne linije 146
Korišćenje interfejsa komandne linije 150
Komande MySQL 151
Tipovi podataka 155
Indeksi 164
Kreiranje indeksa 165
Upit u MySQL bazu podataka 169
Spajanje tabela zajedno 178
Korišćenje logičkih operatora 180

MySQL funkcije 181
Pristupanje MySQL-u preko phpMyAdmina 181
Pitanja 182

9. Ovladavanje MySQL-om 183
Dizajn baze podataka 183
Primarni ključevi: ključevi za relacione baze podataka 184
Normalizacija 185
Prva normalna forma 186
Druga normalna forma 187
Treća normalna forma 190
Kada se ne koristi normalizacija 192
Relacije 192
Jedan-prema-jedan 193
Jedan-prema-više 193
Više-prema-više 194
Baze podataka i anonimnost 195
Transakcije 196
Mehanizam za skladištenje transakcija 196
Korišćenje BEGIN 197
Korišćenje COMMIT 198
Korišćenje ROLLBACK 198
Korišćenje EXPLAIN 199
Pravljenje rezervne kopije i obnavljanje 200
Korišćenje mysqldump 200
Kreiranje rezervne datoteke 201
Vraćanje iz rezervne datoteke 203
Izvoz podataka u CSV formatu 203
Planiranje sigurnosnih kopija 204
Pitanja 205

10. Šta je novo u PHP 8 i MySQL 8 207
O ovom poglavlju 207
PHP 8 208
Imenovani parametri 208
Atributi 208
Svojstva konstruktora 209
Kompilacija u pravo vreme 209
Union Types 210
Null-safe operator 210
Izraz match 210
Nove funkcije 211

MySQL 8 214
Nadogradnje SQL-a 214
JSON (JavaScript Object Notation) 215
Podrška za geografiju 215
Pouzdanost 215
Brzina i performanse 215
Upravljanje 216
Bezbednost 216
Pitanja 217

11. Pristupanje MySQL-u pomoću PHP-a 219
Upit u MySQL bazu podataka sa PHP-om 219
Proces 219
Kreiranje datoteke za prijavljivanje 220
Povezivanje sa MySQL bazom podataka 221
Praktičan primer 225
$_POST niz 228
Brisanje zapisa 229
Prikaz obrasca 229
Upit u bazu podataka 230
Pokretanje programa 231
Praktičan MySQL 232
Kreiranje tabele 232
Opisivanje tabele 233
Uklanjanje tabele 234
Dodavanje podataka 234
Preuzimanje podataka 235
Ažuriranje podataka 236
Brisanje podataka 236
Korišćenje AUTO_INCREMENT 237
Izvršavanje dodatnih upita 238
Sprečavanje pokušaja hakovanja 239
Koraci koje možete preduzeti 240
Korišćenje rezervisanih mesta 242
Sprečavanje ubrizgavanja JavaScript-a u HTML 244
Pitanja 246

12. Rad sa obrascima 247
Izgradnja obrazaca 247
Preuzimanje poslatih podataka 248
Početne vrednosti 250
Tipovi unosa 250
Pročišćavanje unosa 257
Primer programa 258

Poboljšanja HTML5 261
Atribut automatskog dovršavanja 261
Atribut autofokusa 261
Atribut rezervisanog mesta 262
Atribut required 262
Redefinisanje atributa 262
Atributi width i height 262
Atributi min i max 262
Atribut step 263
Atribut form 263
Atribut list 263
Tip unosa color 263
Tipovi unosa number i range 264
Birači datuma i vremena 264
Pitanja 264

13. Kolačići, sesije i autentifikacija 265
Korišćenje kolačića u PHP-u 265
Postavljanje kolačića 266
Pristupanje kolačiću 267
Uništavanje kolačića 267
HTTP autentifikacija 268
Čuvanje korisničkih imena i lozinki 271
Primer programa 272
Korišćenje sesija 276
Pokretanje sesije 277
Završavanje sesije 279
Postavljanje vremenskog ograničenja 280
Sigurnost sesije 281
Pitanja 284

14. Istraživanje JavaScripta 285
JavaScript i HTML tekst 285
Korišćenje skripti unutar glave dokumenta 287
Stariji i nestandardni čitači 287
Uključivanje JavaScript datoteke 288
Otklanjanje JavaScript grešaka 289
Korišćenje komentara 289
Znak tačka sa zarezom 289
Promenljive 290
String promenljive 290
Numeričke promenljive 291
Nizovi 291

Operatori 292
Aritmetički operatori 292
Operatori dodele 292
Operatori poređenja 293
Logički operatori 293
Povećanje, smanjivanje i skraćeno dodeljivanje 294
Nadovezivanje stringova 294
Escape znakovi 294
Tipiziranje promenljivih 295
Funkcije 296
Globalne promenljive 296
Lokalne promenljive 296
Korišćenje let and const 297
Model objekta dokumenta 300
Druga upotreba za simbol $ 301
Korišćenje DOM-a 302
O document.write 303
Korišćenje console.log 303
Upotreba alert 303
Zapisivanje u elemente 303
Korišćenje document.write 303
Pitanja 304

15. Izrazi i kontrola toka u JavaScriptu 305

Izrazi 305
Literali i promenljive 306
Operatori 307
Prioritet operatora 307
Asocijativnost 308
Realcioni operatori 309
Naredba with 311
Korišćenje onerror 312
Korišćenje try...catch 313
Uslovi 314
Naredba if 314
Naredba else 314
Naredba switch 315
Operator ? 317
Petlje 317
Petlje while 317
Petlje do...while 318
Petlje for 319
Izlaz iz petlje 319
Naredba continue 320

Eksplicitno pretvaranje 321
Pitanja 321

16. JavaScript funkcije, objekti i nizovi....323
JavaScript funkcije 323
Definisanje funkcije 323
Vraćanje vrednosti 325
Vraćanje niza 327
JavaScript objekti 327
Deklarisanje klase 327
Kreiranje objekta 329
Pristup objektima 329
Rezervisana reč prototype 330
JavaScript nizovi 332
Numerički nizovi 333
Asocijativni nizovi 334
Višedimenzionalni nizovi 335
Korišćenje metoda niza 336
Pitanja 341

17. Validacija JavaScripta i PHP-a i obrada grešaka....343
Provera korisničkog unosa pomoću JavaScripta 343
Dokument validate.html (1. deo) 343
Dokument validate.html (2. deo) 346
Regularni izrazi 349
Podudaranje sa metaznakovima 349
Podudaranje nejasnih znakova 349
Grupisanje u zagradama 350
Klase znakova 351
Označavanje opsega 351
Negacija 351
Neki složeniji primeri 352
Rezime metaznakova 354
Opšti modifikatori 356
Korišćenje regularnih izraza u JavaScriptu 356
Korišćenje regularnih izraza u PHP-u 356
Ponovni prikaz obrasca nakon PHP validacije 357
Pitanja 363

18. Korišćenje asinhrone komunikacije....365
Šta je asinhrona komunikacija? 366
Korišćenje XMLHttpRequest 366
Vaš prvi asinhroni program 366
Korišćenje GET-a umesto POST-a 370

Slanje XML zahteva 371
Korišćenje radnih okvira za asinhronu komunikaciju 375
Pitanja 376

19. Uvod u CSS 377

Uvoz opisa stilova 378
Uvoz CSS-a iz HTML-a 378
Umetnuti stilovi 379
Korišćenje ID-ova 379
Korišćenje klasa 379
Korišćenje znaka tačka sa zarezom 379
CSS pravila 380
Višestruka dodela 380
Korišćenje komentara 381
Vrste stilova 381
Difolt stilovi 382
Stilovi korisnika 382
Spoljni stilovi 383
Interni stilovi 383
Umetnuti stilovi 383
Selektori CSS-a 383
Selektor tipa 383
Selektor potomak 384
Selektor dete 385
ID selektor 386
Selektor klase 386
Selektor atributa 387
Univerzalni selektor 387
Selektovanje po grupi 388
CSS kaskada 388
Kreatori stilova 388
Metode opisa stila 389
Selektor opisa stilova 389
Razlika između elemenata div i span 391
Merenja 393
Fontovi i tipografija 395
font-family 395
font-style 395
font-size 396
font-weight 396
Upravljanje stilovima teksta 397
Dekoracija 397
Razmak 397
Poravnanje 398

Transformacija 398
Uvlačenje 398
CSS boje 399
Kratki stringovi boje 399
Gradijenti 400
Pozicioniranje elemenata 401
Apsolutno pozicioniranje 401
Relativno pozicioniranje 401
Fiksno pozicioniranje 402
Pseudoklase 404
Skraćena pravila 406
Model i raspored kvadrata 406
Podešavanje margina 407
Primena okvira 408
Podešavanje razmaka 409
Sadržaj objekta 411
Pitanja 411

20. Napredni CSS sa CSS3 413

Selektori atributa 414
Podudaranje delova stringa 414
Svojstvo box-sizing 415
CSS3 pozadine 415
Svojstvo background-clip 416
Svojstvo background-origin 417
Svojstvo background-size 418
Korišćenje vrednosti auto 418
Višestruke pozadine 418
CSS3 okviri 420
Svojstvo border-color 420
Svojstvo border-radius 421
Svojstvo box-shadow 423
Prelivanje elemenata 424
Raspored u više kolona 424
Boje i neprozirnost 426
HSL boje 426
HSLA boje 427
RGB boje 427
RGBA boje 427
Svojstvo opacity 428
Tekstualni efekti 428
Svojstvo text-shadow 428
Svojstvo text-overflow 428
Svojstvo word-wrap 429

Veb fontovi 429
Google veb fontovi 430
Transformacije 431
3D transformacije 433
Tranzicije 433
Svojstva tranzicije 434
Trajanje tranzicije 434
Kašnjenje tranzicije 434
Tempiranje tranzicije 434
Skraćena sintaksa 435
Pitanja 437

21. Pristupanje CSS-u iz JavaScripta 439
Povratak na funkciju getElementById 439
Funkcija O 439
Funkcija S 440
Funkcija C 441
Uključivanje funkcija 441
Pristupanje CSS svojstvima iz JavaScripta 442
Neka uobičajena svojstva 442
Ostala svojstva 443
Umetnut JavaScript 445
Rezervisana reč this 445
Povezivanje događaja objektima u skripti 446
Povezivanje sa drugim događajima 446
Dodavanje novih elemenata 447
Uklanjanje elemenata 449
Alternative dodavanju i uklanjanju elemenata 449
Korišćenje prekida 450
setTimeout 450
Otkazivanje timeouta 451
Korišćenje setInterval 452
Korišćenje prekida za animaciju 453
Pitanja 455

22. Uvod u jQuery 457
Zašto jQuery? 457
Uključivanje jQuerya 458
Odabir prave verzije 458
Preuzimanje 459
Korišćenje mreže za isporuku sadržaja 460
Prilagođavanje jQueryja 460

Sintaksa jQuery 461
Jednostavan primer 461
Izbegavanje sukoba biblioteka 462
Selektori 462
Metoda css 463
Selektor elemenata 463
Selektor ID-ova 463
Selektor klasa 464
Kombinovanje selektora 464
Rukovanje događajima 464
Čekanje dok dokument ne bude spreman 466
Funkcije događaja i svojstva 467
Događaji zamućenja i fokusa 467
Rezervisana reč this 468
Događaji click i dblclick 468
Događaj keypress 470
Obazrivo programiranje 471
Događaj mousemove 471
Ostali događaji sa mišem 474
Alternativne metode miša 475
Događaj submit 475
Specijalni efekti 477
Skrivanje i prikazivanje 478
Metoda toggle 479
fadeIn i fadeOut 479
slideUp i slideDown 480
Animacije 481
Zaustavljanje animacija 484
Manipulacija DOM-om 484
Razlika između tekstualnih i html metoda 485
Metode val i attr 485
Dodavanje i uklanjanje elemenata 486
Dinamička primena klasa 488
Menjanje dimenzija 489
Metode width i height 489
Metode innerWidth i innerHeight 491
Metode outerWidth i outerHeight 491
Kretanje kroz DOM 492
Elementi roditelji 492
Elementi deca 496
Elementi braće i sestara 496
Odabir sledećeg i prethodnog elementa 498
Kretanje kroz jQuery izbore 499
Metoda is 501

Korišćenje jQueryja bez selektora 502
Metoda $.each 502
Metoda $.map 503
Korišćenje asinhrone komunikacije 503
Korišćenje POST metode 503
Korišćenje GET metode 504
Dodaci 505
Korisnički interfejs jQuery 505
Drugi dodaci 505
Pitanja 506

23. Uvod u jQuery Mobile 507
Uključivanje jQuery Mobile 508
Prvi koraci 509
Povezivanje stranica 510
Sinhrono povezivanje 511
Povezivanje u okviru višestraničnog dokumenta 511
Tranzicije stranica 512
Stilizovanje dugmadi 515
Rukovanje spiskovima 517
Spiskovi koji se mogu filtrirati 518
Razdelnici spiskova 520
Šta je sledeće? 523
Pitanja 523

24. Uvod u React 525
U čemu je uopšte suština Reacta? 526
Pristupanje React datotekama 527
Uključuivanje babel.js 528
Naš prvi projekat React 528
Korišćenje funkcije umesto klase 529
Čist i nečist kôd: zlatno pravilo 530
Korišćenje klase i funkcije 531
Osobine i komponente 532
Razlike između korišćenja klase i funkcije 533
Reactova stanja i životni ciklus 533
Korišćenje udica (ako koristite Node.js) 536
Događaji u Reactu 536
Umetnute JSX uslovne naredbe 538
Korišćenje spiska i ključeva 539
Jedinstveni ključevi 540
Rukovanje obrascima 541
Korišćenje unosa teksta 542
Korišćenje textarea 543
Korišćenje select 545

React Native 546
Kreiranje React Native Apps 546
Dalje čitanje 547
Sa Reactom na sledeći nivo 547
Pitanja 548

25. Uvod u HTML5 549
Platno 549
Geolociranje 551
Audio i video 552
Obrasci 553
Lokalno skladište 554
Veb radnici 554
Pitanja 554

26. HTML5 platno 555
Kreiranje i pristup platnu 555
toDataURL funkcija 557
Zadavanje tipa slike 558
Metoda fillRect 559
Metoda clearRect 559
Metoda strokeRect 559
Kombinovanje ovih komandi 559
Metoda createLinearGradient 560
Metoda addColorStop u detaljima 562
Metoda createRadialGradient 563
Korišćenje obrazaca za ispune 565
Pisanje teksta po platnu 566
Metoda strokeText 567
Svojstvo textBaseline 567
Svojstvo font 567
Svojstvo textAlign 568
Metoda fillText 568
Metoda measureText 569
Crtanje linija 569
Svojstvo lineWidth 569
Svojstva lineCap i lineJoin 570
Svojstvo miterLimit 572
Korišćenje putanja 572
Metode moveTo i lineTo 572
Metoda stroke 573
Metoda rect 573
Popunjavanje područja 573
Metoda clip 575

Metoda isPointInPath 578
Rad sa krivama 578
Metoda arc 578
Metoda arcTo 581
Metoda quadraticCurveTo 582
Metoda bezierCurveTo 583
Manipulacija slikama 584
Metoda drawImage 584
Promena veličine slike 585
Izbor oblasti slike 585
Kopiranje sa platna 586
Dodavanje senki 586
Editovanje na nivou piksela 588
Metoda getImageData 588
Metoda putImageData 591
Metoda createImageData 591
Napredni grafički efekti 592
Svojstvo globalCompositeOperation 592
Svojstvo globalAlpha 594
Transformacije 595
Metoda scale 595
Metode save i restore 596
Metoda rotate 596
Metoda translate 597
Metoda transform 598
Metoda setTransform 600
Pitanja 600

27. HTML5 audio i video 601

O kodecima 602
<audio> element 603
<video> element 605
Video kodeci 605
Pitanja 608

28. Ostale HTML5 funkcije 609

Geolociranje i GPS usluga 609
Ostale metode lociranja 610
Geolociranje i HTML5 610
Lokalno skladište 613
Korišćenje lokalnog skladišta 614
Objekat localStorage 614
Veb radnici 616
Prevlačenje i ispuštanje 618

Razmena poruka između dokumenata 620
Ostale HTML5 oznake 624
Pitanja 624

29. Spajanje svega 625
Projektovanje aplikacije za društveno umrežavanje 626
Na veb lokaciji 626
functions.php 626
Funkcije 627
header.php 629
setup.php 632
index.php 633
signup.php 634
Proveravanje raspoloživosti korisničkog imena 635
Prijavljivanje 635
checkuser.php 638
login.php 638
profile.php 640
Dodavanje teksta „O meni” 641
Dodavanje profilne slike 641
Obrada slike 641
Prikaz tekućeg profila 642
members.php 645
Pregled korisničkog profila 645
Dodavanje i poništavanje prijateljstva 645
Spisak svih članova 645
friends.php 648
messages.php 650
logout.php 653
styles.css 654
javascript.js 657
Pitanja 657

A. Odgovori na pitanja u poglavljima 659

Indeks 679

POGLAVLJE 1

Uvod u dinamički veb sadržaj

World wide web je mreža koja se stalno razvija i koja je već daleko napustila svoju koncepciju ranih 1990-ih, kada je stvorena za rešavanje određenog problema. Najsavremeniji eksperimenti u CERN-u (Evropska laboratorija za fiziku čestica, sada najpoznatiji kao operator velikog hadronskog sudarača) proizvodili su neverovatne količine podataka – toliko da su se podaci pokazali nezgrapnima za distribuciju naučnicima koji su učestvovali, a koji su bili rasprostranjeni po celom svetu.

U to vreme, internet je već bio uspostavljen i povezivao je nekoliko stotina hiljada računara, pa je Tim Berners-Lee (član CERN-a) smislio metod navigacije između njih pomoću hiperveza, što je postao poznat kao Hipertext Transfer Protocol ili HTTP. Takođe je stvorio jezik za označavanje pod nazivom Hipertext Markup Language ili HTML. Da bi ih spojio, napisao je prvi veb čitač i veb server.

Danas ove alate uzimamo zdravo za gotovo, ali tada je koncept bio revolucionaran. Korisnici modema su tada imali mogućnosti telefonskog povezivanja od kuće na oglasnu tablu (eng. *bulletin board*) na kojoj su mogli komunicirati i menjati podatke samo s drugim korisnicima te usluge. Shodno tome, morali ste da budete član mnogih sistema oglasnih tabli da biste efikasno elektronski komunicirali sa svojim kolegama i prijateljima.

Ali Berners-Lee je sve to promenio jednim potezom, pa su do sredine 1990-ih postojala tri glavna grafička veb čitača (eng. *browser*) koja su se takmičila za pažnju pet miliona korisnika. Ubrzo je, međutim, postalo očigledno da nešto nedostaje. Da, stranice teksta i grafike sa hipervezama koje vas vode na druge stranice bile su briljantan koncept, ali rezultati nisu odražavali trenutni potencijal računara i interneta da zadovolje posebne potrebe svakog korisnika sa dinamički promenljivim sadržajem. Korišćenje veba je bilo veoma prosto i jednostavno iskustvo, čak iako je postojalo pomeranje teksta i animirani GIF-ovi!

Korpa za kupovinu, pretraživači i društvene mreže jasno su promenile način na koji koristimo veb. U ovom poglavlju ćemo ukratko pogledati različite komponente koje čine veb i softver koji pomaže da njegovo korišćenje postane bogato i dinamično iskustvo.



Potrebno je manje više odmah početi koristiti neke skraćenice. Pokušao sam da ih jasno objasnim pre nego što nastavim, ali ne brinite previše o tome šta predstavljaju ili šta znače ova imena, jer će detalji postati jasni dok čitate.

HTTP i HTML: Osnove Berners-Leea

HTTP je komunikacioni standard koji upravlja zahtevima i odgovorima koji se šalju između čitača koji radi na računaru krajnjeg korisnika i veb servera. Zadatak servera je da prihvati zahtev klijenta i pokuša da na njega odgovori na smislen način, obično posluživanjem tražene veb stranice – zbog čega se koristi izraz *server* (poslužitelj). Prirodni pandan serveru je *klijent* (korisnik), pa se taj izraz primenjuje i na veb čitač i na računar na kojem je čitač.

Između klijenta i servera može postojati nekoliko drugih uređaja, poput rutera, proksija, mrežnih prolaza itd. Oni imaju različite uloge u obezbeđivanju ispravnog prenosa zahteva i odgovora između klijenta i servera. Obično koriste internet za slanje ovih informacija. Neki od ovih uređaja između uređaja mogu takođe pomoći ubrzanju Interneta tako što lokalno skladište stranice ili informacije u pomoćnoj memoriji, takozvanom *kešu* (eng. *cache*), a zatim poslužuju ovaj sadržaj klijentima direktno iz keša, umesto da preuzimaju sve sa izvornog servera.

Veb server obično može da obrađuje više istovremenih veza, a kada ne komunicira sa klijentom, provodi vreme slušajući dolaznu vezu. Kada nešto stigne, server šalje odgovor da potvrdi prijem.

Procedura zahtev/odgovor

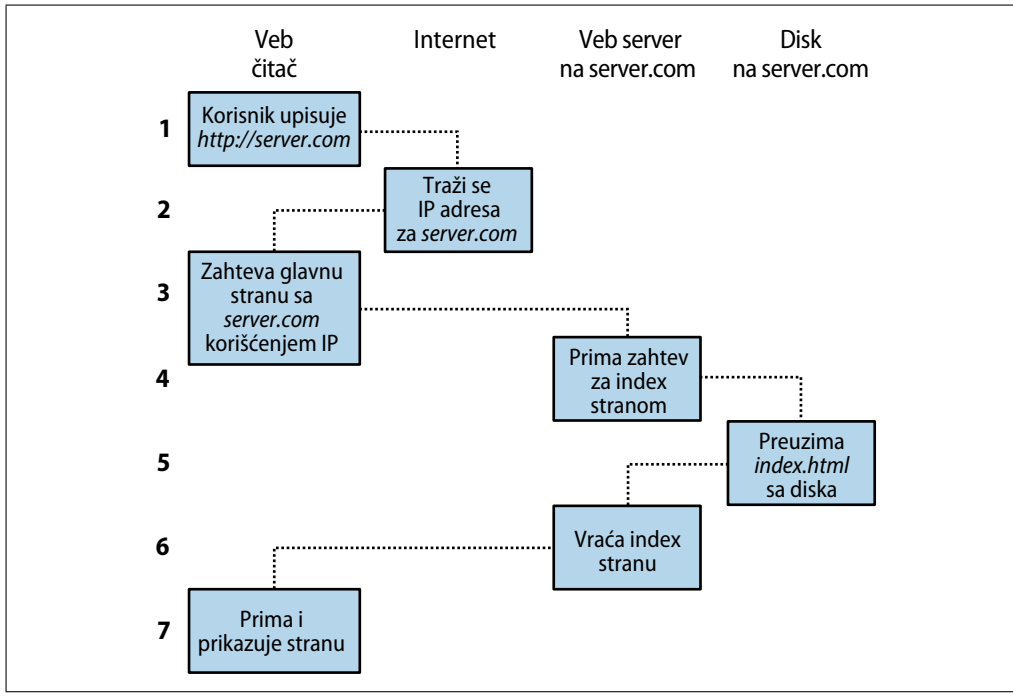
Na svom najosnovnijem nivou, proces zahtev/odgovor (request/response) sastoji se od veb čitača ili drugog klijenta koji traži od veb servera da mu pošalje veb stranicu, a server šalje nazad stranu. Pretraživač se tada brine o prikazivanju ili iscrtavanju stranice (slika 1-1).

Koraci u nizu zahteva i odgovora su sledeći:

1. Unosite *http://server.com* u polje za adresu čitača.
2. Vaš čitača traži adresu Internet protokola (IP) za *server.com*.
3. Vaš čitač šalje zahtev za početnu stranicu na *server.com*.
4. Zahtev prolazi internetom i stiže na *server.com* veb server.
5. Veb server, nakon što je primio zahtev, traži veb stranicu na svom disku.
6. Veb server preuzima stranicu i vraća je čitaču.
7. Vaš čitač prikazuje veb stranicu.

Za prosečnu veb stranicu, ovaj proces se takođe odvija jednom za svaki objekat na stranici: grafiku, ugrađeni video ili Flash datoteku, pa čak i CSS šablon.

U koraku 2, vidimo da čitač traži IP adresu *server.com*. Svaka mašina povezana sa internetom ima IP adresu – i vaš računar – ali generalno pristupamo veb serverima po imenu, kao što je *google.com*. Pretraživač konsultuje dodatnu internetsku uslugu pod nazivom Sistem imena domena (Domain Name System, DNS) kako bi pronašao IP adresu servera, a zatim je koristi za komunikaciju sa računarom.



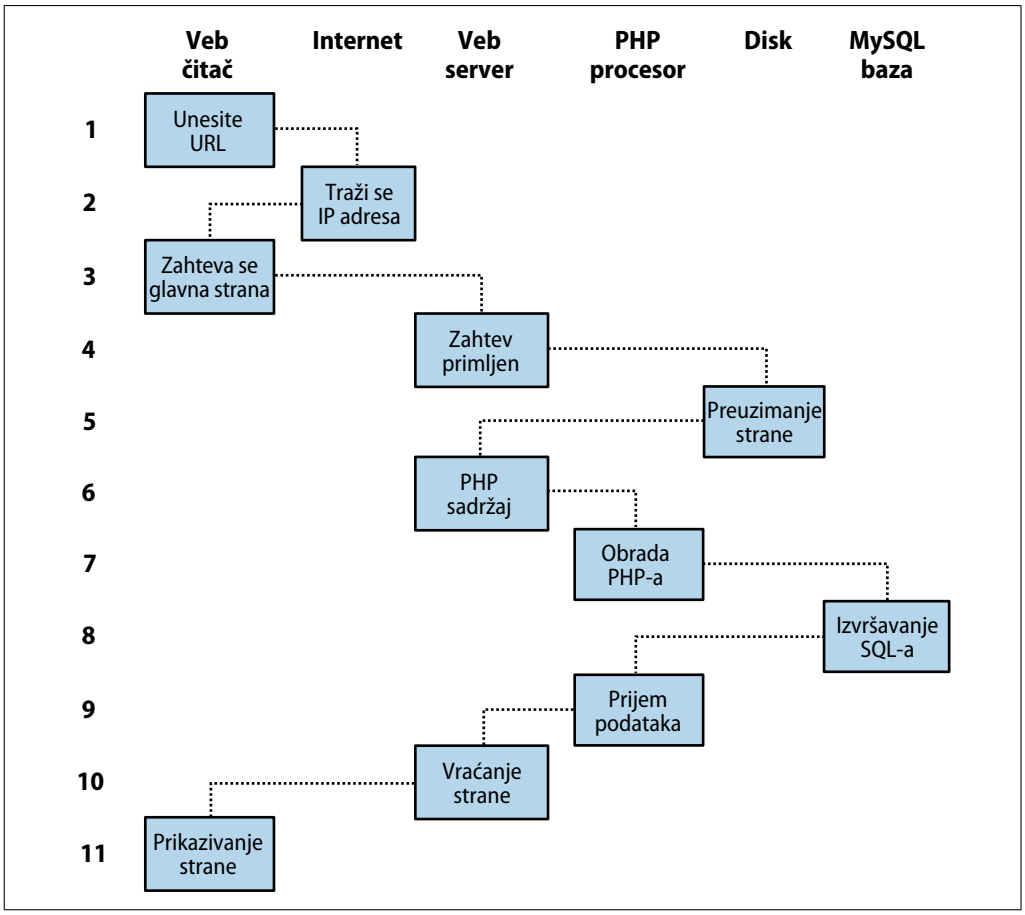
Slika 1-1. Osnovni redosled zahteva/odgovora klijent/servera

Za dinamičke veb stranice, postupak je malo složeniji, jer može uključiti i PHP i MySQL. Na primer, možete kliknuti na sliku kabanice. Zatim će PHP sastaviti zahtev koristeći standardni jezik baze podataka, SQL – čije ćete mnoge komande naučiti u ovoj knjizi – i poslati zahtev na MySQL server. MySQL server će vratiti informacije o kabanici koju ste izabrali, a PHP kôd će sve to umotati u neki HTML koji će server poslati vašem čitaču (slika 1-2).

Koraci su sledeći:

1. Unesite *http://server.com* u polje za adresu čitača.
2. Vaš čitač traži IP adresu za *server.com*.
3. Vaš čitač šalje zahtev na toj adresu za početnu stranicu veb servera.
4. Zahtev prolazi internetom i stiže na *server.com* veb server.
5. Veb server, nakon što je primio zahtev, preuzima početnu stranicu sa svog diska.
6. Pošto je početna stranica sada u memoriji, veb server uočava da je to datoteka koja sadrži PHP kôd i prosleđuje stranicu PHP interpreteru.
7. PHP interpreter izvršava PHP kôd.
8. Neki od PHP-a sadrže SQL izraze, koje PHP interpreteru sada prosleđuje bazi podataka MySQL.
9. MySQL baza podataka vraća rezultate izraza PHP interpreteru.

10. PHP interpreteru vraća rezultate izvršenog PHP koda, zajedno sa rezultatima iz MySQL baze podataka, na veb server.
11. Veb server vraća stranicu klijentu koji je zatražio, a koji je prikazuje.



Slika 1-2. Dinamički redosled zahteva/odgovora klijent/servera

Iako je korisno biti svestan ovog procesa kako biste znali kako tri elementa funkcionišu zajedno, u praksi se ne morate baviti tim detaljima, jer se svi oni dešavaju automatski.

HTML stranice vraćene čitaču u svakom primeru mogu sadržati JavaScript, koji će klijent lokalno interpretirati i koji bi mogao da pokrene drugi zahtev – na isti način kao i ugrađeni objekti, poput slika.

Prednosti PHP-a, MySQL-a, JavaScripta, CSS-a i HTML5-a

Na početku ovog poglavlja predstavio sam svet veba 1.0, ali nije prošlo mnogo vremena pre nego što se krenulo u pravljenje veba 1.1, sa razvoj poboljšanja čitača kao što su Java, JavaScript, JScript (Microsoftova lakša varijanta JavaScripta) i ActiveX. Na strani servera napredovalo se na Common Gateway Interface (CGI) upotrebi skript jezika kao što su Perl (alternativa PHP jeziku) i *skriptama na strani servera*– ubacivanjem sadržaj jedne datoteke (ili izlaza pokretanja lokalnog programa) u drugu dinamički.

Kada se prašina slegla, tri glavne tehnologije stajale su rame uz rame iznad ostalih. Iako je Perl još uvek bio popularan skript jezik sa vernim sledbenicima, PHP-ova jednostavnost i ugrađene veze do programa MySQL baze podataka skupile su više nego dvostruko više korisnika. A JavaScript, koji je postao suštinski deo jednačine za dinamičko manipulisanje kaskadnim tablicama stila (Cascading Style Sheets, CSS) i HTML, preuzeli su još teže zadatke upravljanja klijentskom stranom asinhrone komunikacije (razmena podataka između klijenta i servera nakon što se veb stranica učitala). Koristeći asinhronu komunikaciju, veb stranice obavljaju rukovanje podacima i šalju zahteve veb serverima u pozadini – a da veb korisnik nije svestan da se to dešava.

Nema sumnje da je simbiotska priroda PHP-a i MySQL-a pomogla i jednom i drugom da napreduje, ali šta ih je uopšte privuklo programere njima? Jednostavan odgovor mora biti lakoća s kojom ih možete koristiti za brzo pravljenje dinamičkih elemenata na veb stranicama. MySQL je brz i moćan, ali jednostavan za korišćenje sistema baze podataka koji nudi gotovo sve što bi jednoj veb lokaciji bilo potrebno da pronađe i servira podatke čitačima. Kada se PHP udruži sa MySQL-om radi skladištenja i preuzimanja ovih podataka, imate osnovne delove potrebne za početak Web 2.0.

A kada u mešavinu unesete i JavaScript i CSS, imate recept za izgradnju visoko dinamičnih i interaktivnih veb stranica – pogotovo jer sada postoji širok spektar sofisticiranih radnih okvira (eng. *framework*) JavaScript funkcija koje možete pozvati da biste zaista ubrzali veb razvoj. Ovo uključuje dobro poznati jQuery, koji je do nedavno bio jedan od najčešćih načina na koji programeri pristupaju asinhronim komunikacionim funkcijama i noviju biblioteku React JavaScript, koja postaje sve popularnija. To je sada jedan od najčešće preuzimanih i implementiranih radnih okvira, toliko da od 2020. godine oglasi za posao zaista sadrže dvostruko više potražnje za React nego za jQuery programerima.

MariaDB: Klon MySQL-a

Nakon što je Oracle kupio Sun Microsystems (vlasnika MySQL-a), zajednica je postala zabrinuta da MySQL možda neće ostati u potpunosti otvorenog koda, pa se MariaDB od nje izdvojila kako bi bila besplatna pod GNU GPL licencom. Razvoj MariaDB-a vode neki od originalnih programera MySQL-a tako da zadržava izuzetno blisku kompatibilnost sa MySQL-om. Stoga ćete možda naići na MariaDB na nekim serverima umesto MySQL-a – ali ne brinite, sve u ovoj knjizi radi podjednako dobro i na MySQL-u i na MariaDB-u. Za sve namene, možete zamijeniti jedno s drugim i nećete primetiti nikakvu razliku.

U svakom slučaju, ispostavilo se da se čini da su mnogi početni strahovi ublaženi jer MySQL ostaje otvorenog koda, pri čemu Oracle jednostavno naplaćuje podršku i izdanja koja pružaju dodatne funkcije, poput geo replikacije i automatskog skaliranja. Međutim, za razliku od MariaDB-a, MySQL više ne vodi zajednica, pa činjenica da će MariaDB uvek biti tu, ako ikada zatreba, pruža programerima miran san i verovatno obezbeđuje da će i sam MySQL ostati otvorenog koda.

Korišćenje PHP-a

Uz PHP, jednostavno je ugraditi dinamičke aktivnosti u veb stranice. Kada stranicama date proširenje *.php*, oni imaju trenutni pristup skrip jeziku. Sa tačke gledišta programera, sve što treba da uradite je da napišete kôd, poput sledećeg:

```
<?php
  echo " Danas je " . date("l") . ". ";
?>
Evo najnovije vesti.
```

Početni `<?php` govori veb serveru da dozvoli PHP programu da interpretira sve sledeće kodove do oznake `?>`. Van njih, sve se šalje klijentu kao direktan HTML. Dakle, tekst `Evo najnovije vesti.` se jednostavno šalje u čitač; unutar PHP oznaka, ugrađena funkcija `date` prikazuje tekući dan u nedelji prema sistemskom vremenu servera.

```
Danas je Wednesday. Evo najnovije vesti.
```

PHP je fleksibilan jezik i neki ljudi radije postavljaju PHP graničnike direktno uz PHP kôd, na sledeći način:

```
Danas je <?php echo date("l"); ?>. Evo najnovije vesti.
```

Postoji više načina formatiranja i prikazivanja informacija, koje ću objasniti u poglavljima o PHP-u. Poenta je u tome što sa PHP-om veb programeri imaju skript jezik koji je, iako ne tako brz kao kada pišete vaš kod na C-u ili sličnom jeziku, neverovatno je brz i takođe se besprekorno integriše sa HTML oznakama.



Ako nameravate da unesete primere PHP-a iz ove knjige u vaš editor programa da radite sa mnom, morate zapamtiti da dodate `<?php` ispred `?>` nakon unetog koda kako bi se obezbedilo da ih tumač PHP obradi. Da biste to olakšali, možda ćete želeti da pripremite datoteku pod imenom *example.php* sa tim oznakama.

Koristeći PHP, imate neograničenu kontrolu nad svojim veb serverom. Bilo da treba da menjate HTML u hodu, obrađujete kreditnu karticu, dodajete podatke o korisniku u bazu podataka ili preuzimate informacije sa veb lokacije treće strane, sve to možete učiniti iz istih PHP datoteka u kojima se nalazi i sam HTML.

Korišćenje MySQL-a

Naravno, nema mnogo smisla dinamički menjati HTML izlaz, a da nemate način da pratite informacije koje korisnici pružaju vašoj veb lokaciji dok je koriste. U ranim danima veba,

mnoge veb lokacije su koristile „ravne" tekstualne datoteke za skladištenje podataka kao što su korisnička imena i lozinke. Ali ovaj pristup može da izazove probleme ako datoteka nije pravilno zaključana, jer može doći do njenog oštećenja iz više istovremenih pristupa. Takođe, ravna datoteka može postati toliko velika pre nego što postane nezgrapna za korišćenje – da ne govorimo o teškoćama spajanja datoteka i obavljanja složenih pretraživanja u bilo kom razumnom roku.

Na tom mestu relacione baze podataka sa strukturisanim upitima su postale bitne. MySQL, besplatan za upotrebu i instaliran na velikom broju internet veb servera, odlično je rastao. To je robustan i izuzetno brz sistem za upravljanje bazom podataka koji koristi komande nalik engleskom.

Najviši nivo MySQL strukture je baza podataka, u okviru koje možete imati jednu ili više tabela koje sadrže vaše podatke. Na primer, pretpostavimo da radite na tabeli pod nazivom *korisnici*, u okviru koje ste napravili kolone za *prezime*, *ime* i *e-poštu*, i sada želite da dodate još jednog korisnika. Jedna od naredbi koju možete koristiti za ovo je sledeća:

```
INSERT INTO korisnici VALUES('Smith', 'John', 'jsmith@mysite.com');
```

Ranije ste izdali druge komande za pravljenje baze podataka i tabele i za postavljanje svih ispravnih polja, ali SQL `INSERT` naredba ovde pokazuje koliko jednostavno može biti dodavanje novih podataka u bazu podataka. SQL je jezik dizajniran ranih 1970-ih koji podseća na jedan od najstarijih programskih jezika, COBOL. Međutim, dobro je prilagođen za upite u bazu podataka, zbog čega se i dalje koristi nakon toliko vremena.

Podjednako je lako tražiti podatke. Pretpostavimo da imate adresu e-pošte korisnika i da morate potražiti ime te osobe. Da biste to uradili, mogli biste da postavite MySQL upit, kao što je sledeći:

```
SELECT prezime,ime FROM korisnici WHERE email='jsmith@mysite.com';
```

MySQL će tada vratiti `Smith John` i bilo koji drugi par imena koji može biti povezan sa tom adresom e-pošte u bazi podataka.

Kao što ste očekivali, postoji mnogo više toga što možete da uradite sa MySQL-om od običnog `INSERT` (umetni) i `SELECT` (izbaeri) komande. Na primer, možete kombinovati povezane skupove podataka da biste spojili povezane informacije, tražiti rezultate u raznim narudžbama, napraviti delimična podudaranja kada znate samo deo niza koji tražite, vratiti samo *n* rezultata, i još mnogo toga.

Koristeći PHP, sve ove pozive možete da upućujete direktno u MySQL, a da ne morate sami da pristupate interfejsu komandne linije MySQL. To znači da možete sačuvati rezultate u nizovima (eng. *arrays*) za obradu i izvršiti više pretraživanja, od kojih svako zavisi od rezultata vraćenih iz prethodnih pretraživanja, da biste „iskopali" podatak koja vam je potreban.

Za još veću snagu, kao što ćete videti kasnije, postoje dodatne funkcije ugrađene direktno u MySQL koje možete pozvati da biste efikasno izvršavali uobičajene operacije unutar samog MySQL-a, umesto da ih pravite sa više PHP poziva u MySQL.

Korišćenje JavaScript-a

JavaScript je napravljen da omogući skript pristup svim elementima HTML dokumenta. Drugim rečima, pruža način za dinamičku interakciju korisnika, poput provere valjanosti adrese e-pošte u obrascima za unos i prikazivanja upita poput „Da li ste to zaista mislili?" (iako se na to ne može osloniti zbog bezbednosti, i uvek treba izvoditi na veb serveru).

U kombinaciji sa CSS-om (pogledajte sledeći odeljak), JavaScript je moć koja se nalazi iza dinamičkih veb stranica koje se menjaju pred vašim očima, a ne samo kada server vrati novu stranicu.

Međutim, JavaScript je nekada bio nezgodan za upotrebu, zbog nekih velikih razlika u načinima na koje su različiti dizajneri veb čitača izabrali njegovu implementaciju. Do toga je uglavnom došlo kada su neki proizvođači pokušali da dodaju sopstvene funkcionalnosti svojim čitačima na račun kompatibilnosti sa svojim rivalima.

Na sreću, programeri su većinom došli k sebi i shvatili potrebu za potpunom međusobnom kompatibilnošću, pa je ovh dana manje potrebno optimizovati vaš kôd za različite čitače. Međutim, milioni korisnika i dalje koriste stare čitače, a to će verovatno biti slučaj još mnogo godina. Srećom, postoje rešenja za probleme nekompatibilnosti, a kasnije ćemo u ovoj knjizi pogledati biblioteke i tehnike koje vam omogućavaju da bezbedno zanemarite ove razlike.

Za sada, pogledajmo kako se koristi osnovni JavaScript, prihvaćen od svih čitača:

```
<script type="text/javascript">
  document.write("Danas je " + Date() );
</script>
```

Ovaj deo koda govori veb čitaču da interpretira sve unutar `<script>` oznaka kao JavaScript, ono što čitač radi je ispisivanje teksta `Danas je` u tekući dokument, zajedno sa datumom, koristeći JavaScript funkciju `Date`. Rezultat će izgledati otprilike ovako:

```
Danas je Wed Jan 01 2025 01:23:45
```



Osim ako ne morate da navedete tačnu verziju JavaScripta, možete normalno izostaviti `type="text/javascript"` i koristite samo `<script>` da bi započelo interpretiranje JavaScripta.

Kao što je ranije pomenuto, JavaScript je prvobitno razvijen da nudi dinamičku kontrolu nad različitim elementima u HTML dokumentu, i to je i dalje njegova glavna upotreba. Ali sve više, JavaScript se koristi za *Ajax*, proces pristupa veb serveru u pozadini.

Asinhrona komunikacija je ono što omogućava veb stranicama da počnu da liče na samostalne programe, jer se ne moraju ponovo učitavati u celini da bi se prikazao novi sadržaj. Umesto toga, asinhroni poziv može ažurirati jedan element na veb stranici, poput promene fotografije na društvenoj mreži ili zamene dugmeta na koje kliknete odgovorom na pitanje. Ova tema je u potpunosti obrađena u poglavlju 18.

U poglavlju 22 detaljno obrađujemo jQuery, koji možete koristiti za uštedu ponovnog otkrivanja točka kada vam je potreban brz, kôd za prikaz i manipulaciju vašim veb stranicama. Naravno, dostupni su i drugi paketi, pa smo u poglavlju 24 obradili i React, jedan od najpopularnijih izbora današnjice. Oba su izuzetno pouzdani i glavni alati u kompletu alata mnogih iskusnih veb programera.

Korišćenje CSS-a

CSS je ključni pratilac HTML-a, koji obezbeđuje da su HTML tekst i umetnute slike postavljene dosledno i na način koji odgovara ekranu korisnika. Sa pojavom standarda CSS3 poslednjih godina, CSS sada nudi nivo dinamičke interaktivnosti koji je ranije podržavao samo JavaScript. Na primer, ne samo da možete stilizovati bilo koji HTML element, da promeni njegove dimenzije, boje, okvir, razmake itd, već sada možete na veb stranici dodati i animirane prelaze (eng. *transitions*) i transformacije, koristeći samo nekoliko redova CSS-a.

Korišćenje CSS-a može biti jednostavno kao umetanje nekoliko pravila (komandi) između `<style>` i `</style>` oznaka u zaglavlju veb stranice, na primer:

```
<style>
  p {
    text-align:justify;
    font-family:Helvetica;
  }
</style>
```

Ova pravila menjaju osnovno poravnanje teksta, tako da pasusi koji sadrže `<p>` postaju obostrano poravanati (eng. *justify*) i koriste font Helvetica.

Kao što ćete saznati u poglavlju 19, postoji mnogo različitih načina na koje možete upotrebiti CSS pravila, a takođe ih možete uključiti direktno u HTML oznake ili sačuvati skup pravila u spoljnu datoteku i da ih učitavate. Ova fleksibilnost ne samo da vam omogućava da precizno oblikujete svoj HTML, već može (na primer) da obezbedi ugrađenu funkciju lebdenja (eng. *hover*) za animaciju objekata dok mišem prelazite preko njih. Takođe ćete saznati kako da pristupite svim CSS svojstvima elementa iz JavaScripta, kao i iz HTML-a.

I evo ga HTML5

Koliko god da su bili korisni svi ovi dodaci veb standardima, nisu bili dovoljni za rastući broj ambicioznih programera. Na primer, još uvek nije postojao jednostavan način za manipulisanje grafikom u veb čitaču bez pribegavanja dodacima kao što je Flash. Isto se odnosi i na umetanje audio i video zapisa na veb stranice. Osim toga, nekoliko dosadnih nedoslednosti uvuklo se u HTML tokom njegove evolucije.

Dakle, da bi se sve ovo raščistilo i internet otišao dalje od Weba 2.0 i u njegovu sledeću iteraciju, stvoren je novi standard za HTML koji će otkloniti sve ove nedostatke: *HTML5*. Njegov razvoj započeo je 2004. godine, kada su prvi nacrt sastavile Mozilla Foundation i Opera

Softvare (programeri dva popularna veb čitača), ali je tek 2013. konačni nacrt dostavljen u World Wide Web Consortium (W3C), međunarodno upravljačko telo za veb standarde.

Bilo je potrebno nekoliko godina da se HTML5 razvije, ali sada smo na vrlo čvrstoj i stabilnoj verziji 5.1 (od 2016.). To je, međutim, beskonačan razvojni ciklus, a vremenom će se u njega ugraditi više funkcija, s verzijom 5.2 koja je objavljena kao preporuka W3C 2017., HTML 5.3 (sa predloženim funkcijama, kao što je automatsko pisanje velikih slova) koje se planira još od 2020. itd. Neke od najboljih funkcija u HTML5 za rukovanje i prikazivanje medija uključuju `<audio>`, `<video>` i `<canvas>` elemente koji dodaju zvuk, video i naprednu grafiku. Sve što treba da znate o ovim i svim drugim aspektima HTML5 detaljno je obrađeno počevši od poglavlja 25.



Jedna od sitnica koje mi se sviđa u HTML5 je specifikacija da XHTML sintaksa više nije potrebna za elemente koji se sami zatvaraju. Ranije ste mogli da prikažete prelom redova koristeći `<br>` element. Zatim, kako bi se osigurala buduća kompatibilnost s XHTML-om (planirana zamena za HTML koja se nikada nije dogodila), ovo je promijenjeno u `<br/>`, u kojem je završni `/` znak dodat (pošto se očekivalo da svi elementi sadrže završnu oznaku sa ovim znakom). Ali sada su stvari rešene i možete koristiti bilo koju verziju ovih vrsta elemenata. Dakle, radi sažetosti i manje pritiska na tastere, u ovoj knjizi sam se vratio na prethodni stil `<br>`, `<hr>` itd.

Apache veb server

Pored PHP-a, MySQL-a, JavaScripta, CSS-a i HTML5, postoji i šesti junak na dinamičkom vebu: veb server. U slučaju ove knjige, to znači Apache veb server. Malo smo objasnili šta veb server radi tokom razmene HTTP server/klijent, ali iza scene radi mnogo više.

Na primer, Apache ne poslužuje (servira) samo HTML datoteke – već rukuje širokim spektrom datoteka, od slika i Flash datoteka do MP3 audio datoteka, RSS (Really Simple Syndication) izvorima itd. I ti objekti ne moraju biti statičke datoteke poput GIF slika. Svi oni mogu biti generisani programima poput PHP skripti. Tako je: PHP može čak i da stvara slike i druge datoteke za vas, bilo u hodu ili unapred da bi se kasnije upotrebile.

Da biste to uradili, obično imate module koji su unapred kompajlirani u Apacheu ili PHP-u ili se pozivanju tokom izvršavanja. Jedan takav modul je biblioteka GD (Graphics Draw), koju PHP koristi za izradu i rukovanja grafikom.

Apache takođe podržava veliki broj sopstvenih modula. Pored PHP modula, najvažniji za vaše potrebe veb programera su moduli koji se bave bezbednošću. Drugi primeri su modul Rewrite (Prepisivanje), koji veb serveru omogućava rukovanje nizom URL adresa i njihovo prepisivanje prema sopstvenim internim zahtevima, i Proxy modul, koji možete koristiti za posluživanje često zahtevanih stranica iz keša kako biste smanjili opterećenje na serveru.

Kasnije u knjizi videćete kako da koristite neke od ovih modula za poboljšanje funkcija koje pružaju tri osnovne tehnologije.

Rukovanje mobilnim uređajima

Sada smo čvrsto u svetu međusobno povezanih mobilnih računarskih uređaja, a koncept razvoja veb stranica samo za desktop računare je zastareo. Umesto toga, programeri sada imaju za cilj da razviju responzivne veb lokacije i veb aplikacije koje se prilagođavaju okruženju u kojem se nalaze.

Novo u ovom izdanju je da pokazujem kako možete lako da stvarate ove vrste proizvoda koristeći samo tehnologije opisane u ovoj knjizi, zajedno sa moćnom jQuery Mobile bibliotekom odgovarajućih JavaScript funkcija. Pomoću nje ćete moći da se fokusirate na sadržaj i upotrebljivost vaših veb lokacija i veb aplikacija, znajući da će način na koji se prikazuju biti automatski optimizovan za niz različitih računarskih uređaja – jedna stvar manje o kojoj morate brinuti.

Da bi demonstrirao kako u potpunosti iskoristiti raspoloživu moć, u poslednjem poglavlju ove knjige pravimo, kao primer, jednostavnu veb lokaciju društvene mreže, koristeći jQuery kako bi bila potpuno responzivna (odzivna) i obezbedila da se dobro prikazuje na bilo čemu, od malog ekrana mobilnog telefona do tablet ili desktop računar. Mogli smo podjednako da koristimo React (ili druge JavaScript biblioteke ili radne okvire), ali možda je to vežba koju ćete želeti da sami prođete kada završite izučavanje ove knjige.

O otvorenom kodu

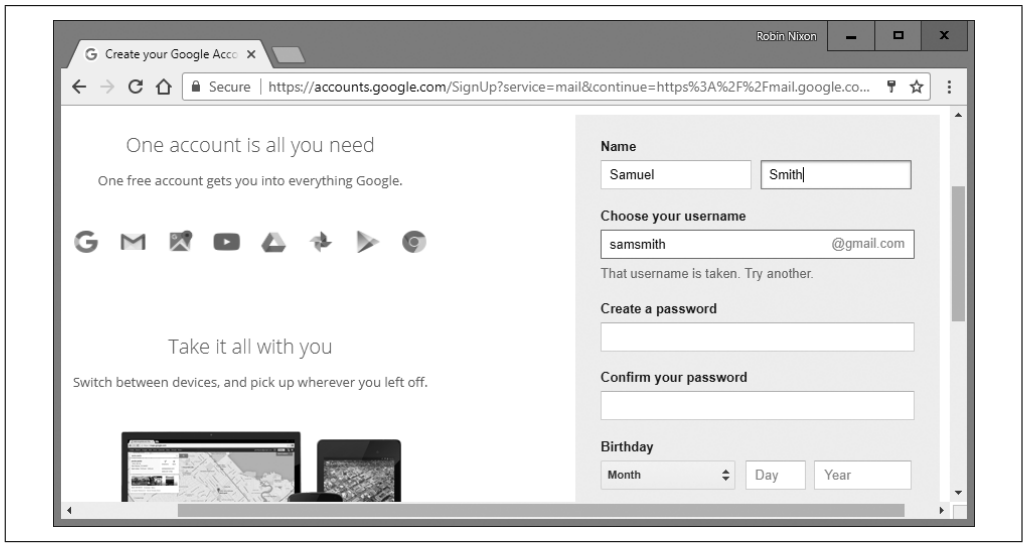
Tehnologije u ovoj knjizi su otvorenog koda: svakome je dozvoljeno da čita i menja kôd. Često se raspravljalo o tome da li je ovaj status razlog zašto su ove tehnologije toliko popularne, ali PHP, MySQL i Apache *jesu* tri najčešće korišćena alata u svojim kategorijama. Ono što se definitivno može reći je da njihov otvoreni kôd znači da ih je razvila zajednica programera koji su napisali funkcije koje su im potrebne, sa originalnim kodom koji je dostupan svima, da svako može da vidi i promeni kôd. Greške se mogu brzo pronaći, a bezbednosni propusti mogu se sprečiti pre nego što se dogode.

Postoji još jedna prednost: svi ovi programi su besplatni za upotrebu. Nema brige oko kupovine dodatnih licenci ako morate da povećate svoju veb lokaciju i dodate više servera, a ne morate da proveravate budžet pre nego što odlučite da li ćete se nadograditi na najnovije verzije ovih proizvoda.

Sve to zajedno

Prava lepota PHP-a, MySQL-a, JavaScripta (ponekad uz pomoć Reacta ili drugih biblioteka), CSS-a i HTML5 je divan način na koji svi zajedno rade na stvaranju dinamičkog veb sadržaja: PHP obavlja sav glavni posao na veb serveru, MySQL upravlja svim podacima, a kombinacija CSS-a i JavaScripta brine o prezentaciji veb stranice. JavaScript takođe može komunicirati sa vašim PHP kodom na veb serveru kad god treba nešto ažurirati (bilo na serveru ili na veb stranici). A pomoću moćnih funkcija u HTML5, kao što su canvas (platno), audio i video i geolokacija, možete učiniti svoje veb stranice visoko dinamičnim, interaktivnim i multimedijskim.

Bez korišćenja programskog koda, rezimirajmo sadržaj ovog poglavlja posmatrajući proces kombinovanja nekih od ovih tehnologija u svakodnevnu funkciju asinhrone komunikacije koju koriste mnoge veb lokacije: provera da li željeno korisničko ime već postoji na veb lokaciji kada se korisnik prijavljuje da otvori novi nalog. Dobar primer ovoga može se videti sa Gmailom (slika 1-3).



Slika 1-3. Gmail koristi asinhronu komunikaciju da proveri da li je slobodno korisničko ime

Koraci koji čine ovaj asinhroni proces biće slični sledećim:

1. Server šalje HTML za pravljenje veb obrasca, a obrazac traži potrebne detalje, kao što su korisničko ime, ime, prezime i adresa e-pošte.
2. Istovremeno, server dodaje JavaScript HTML-u za nadgledanje polja za unos korisničkog imena i proveru dve stvari: da li je unesen tekst u polje i da li je korisnik napustio polje za unos jer je prešao na drugo polje za unos.
3. Nakon što je tekst unet i polje napušteno, JavaScript kôd u pozadini prosleđuje korisničko ime koje je uneto, nazad u PHP skriptu na veb serveru i čeka odgovor.
4. Veb server traži korisničko ime i odgovara JavaScriptu da li je to ime već zauzeto.
5. JavaScript zatim postavlja oznaku pored polja za unos korisničkog imena kako bi pokazao da li je ime dostupno korisniku – možda znak zelene štikle ili crvenog krstića, zajedno sa nekim tekstom.
6. Ako korisničko ime nije dostupno, a korisnik pošalje obrazac, JavaScript prekida slanje i ponovo naglašava (možda sa jačom grafikom i/ili okvirom upozorenja) da korisnik mora izabrati drugo korisničko ime.
7. Opciono, poboljšana verzija ovog procesa bi čak mogla na osnovu korisničkog imena koje je zatražio korisnik, da korisniku predloži alternativna imena koja su trenutno dostupna.

Sve ovo se odvija tiho u pozadini i omogućava ugodno i besprekorno korisničko iskustvo. Bez asinhrone komunikacije, ceo obrazac bi morao da se dostavi serveru, koji bi zatim vratio HTML, kojim bi se potom iscrtala cela stranica sa naznačenim delovima. To bi bilo upotrebljivo rešenje, ali ni izbliza tako uredno i prijatno kao obrada polja u letu (eng. *on-the-fly*).

Asinhrona komunikacija se može koristiti za mnogo više od jednostavne provere i obrade unosa; kasnije u ovoj knjizi ćemo istražiti mnoge dodatne stvari koje možete učiniti sa njom.

U ovom poglavlju ste pročitali solidan uvod u osnovne tehnologije PHP-a, MySQL-a, JavaScripta, CSS-a i HTML5 (kao i Apachea) i naučili kako zajedno rade. U poglavlju 2, pogledaćemo kako možete da instalirate sopstveni server za veb razvoj na kome ćete vežbati sve ono što ćete učiti.

Pitanja

1. Koje četiri komponente (minimum) su potrebne za izradu potpuno dinamičke veb stranice?
2. Šta znači *HTML*?
3. Zašto naziv *MySQL* sadrži slova *SQL*?
4. *PHP* i *JavaScript* su dva programska jezika koja generišu dinamičke rezultate za veb stranice. Koja je njihova glavna razlika i zašto biste koristili oba?
5. Šta znači *CSS*?
6. Navedite tri glavna nova elementa uvedena u HTML5.
7. Ako naiđete na grešku (što je retko) u nekom od alata otvorenog koda, šta mislite kako biste mogli da je rešite?
8. Zašto su radni okviri kao što su jQuery ili React toliko važni za razvoj savremenih veb lokacija i veb aplikacija?

Za odgovore na ova pitanja pogledajte „Poglavlje 1: odgovori” u dodatku A.