
Raspberry Pi Full Stack

Sveobuhvatan kurs koji će vas naučiti
kako da kreirate modernu IoT aplikaciju



dr Peter Dalmaris

Agencija Eho
www.infoelektronika.net

● Sva prava zadržana. Nijedan deo ove knjige ne sme biti reprodukovan u bilo kom materijalnom obliku, uključujući fotokopiranje ili slučajno ili nenamerno smeštanje na bilo koji elektronski medijum sa ili uz pomoć bilo kog elektronskog sredstva, bez pismenog odobrenja nosioca autorskih prava osim u skladu sa odredbama zakona o autorskim pravima, dizajnu i patentima iz 1988. godine ili pod uslovima izdatim od Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England W1P 9HE. Prijave za pismene dozvole radi štampanja bilo kog dela ove publikacije upućuju se izdavaču ove knjige.

● Izjava: Autor i izdavač su uložili najveće napore da bi se obezbedila tačnost informacija sadržanih u ovoj knjizi. Autor i izdavač ne mogu da pretpostave neprijatnosti i ovom izjavom isključuju bilo kakvu odgovornost za bilo koju stranku koja bi imala gubitke ili štetu uzrokovanu greškama ili propustima u ovoj knjizi, bez obzira da li su greške ili propusti nastali usled nemara, nezgode ili bilo kog drugog razloga.

ISBN 978-86-80134-31-4

Raspberry Pi Full Stack

Naslov originala: Raspberry Pi Full Stack

Autor: dr Peter Dalmaris

Prevod: Biljana Tešić

Izdaje i štampa: Agencija Eho, Niš

e-mail: redakcija@infoelektronika.net

Tiraž: 200

Godina izdanja: 2020

СIP - Каталогизација у публикацији
Библиотеке Матице српске, Нови Сад

004.42.738.5RASPERRYPI(036)

ДЕЛАМАРИС, Петер

Raspberry Pi Full Stack / Peter Dalmaris ; [prevod Biljana Tešić]. - Niš :
Agencija Eho, 2020 (Niš : Agencija Eho). - 425 str. ; 24 cm

Tiraž 200. - Na koricama beleška o autoru i knjizi s autorovom slikom.

ISBN 978-86-80134-31-4

1. Тешић, Биљана, 1978- [prevodilac]

а) Рачунарско програмирање -- Интернет -- Приручници

COBISS.SR-ID 24677129

Sadržaj	3
Da li ste našli grešku?	10
O autoru	10
Kako čitati ovu knjigu	11
Zahtevi	11
Veb stranica knjige	12
Da li ste pronašli grešku?	12
Deo 1 • Početak rada u Raspberry Pi Full Stack-u	13
Poglavlje 1 • O čemu je ova knjiga?	14
Poglavlje 2 • Šetnja kroz projekat Full Stack	16
Poglavlje 3 • Potreban hardver	20
Koji Raspberry Pi?	20
HAT PCB	23
Poglavlje 4 • Kako dobiti pomoć	24
Poglavlje 5 • Spremište koda	26
Deo 2 • Raspberry Pi, Arduino i Raspberry Pi Zero W	27
Poglavlje 6 • Poređenje Raspberry Pi-a i Arduina na visokom nivou	28
Poglavlje 7 • Potreba za efikasnošću: Raspberry Pi Zero W	33
Poglavlje 8 • Potreba za brzinom: Raspberry Pi 4 (i 3)	36
Deo 3 • Kako postaviti operativni sistem	38
Poglavlje 9 • Operativni sistemi za Raspberry Pi	39
Poglavlje 10 • Šta je „headless“ operativni sistem	42
Poglavlje 11 • Kako preuzeti i instalirati Raspbian	45
Poglavlje 12 • Kako postaviti SSH i WiFi u headless režimu	48
Poglavlje 13 • Kako postaviti naziv hosta	50
Poglavlje 14 • Prvo pokretanje Raspbian-a	53
Poglavlje 15 • Kako postaviti fiksnu IP adresu	58
Poglavlje 16 • Osnovna konfiguracija	59
Poglavlje 17 • Raditi kao „root“ korisnik	61
Deo 4 • Kako napraviti rezervnu kopiju i oporaviti SD karticu	64
Poglavlje 18 • Pravljenje rezervne kopije SD kartice - MacOS	65
Poglavlje 19 • Oporavak SD kartice - MacOS	67

Poglavlje 20 • Pravljenje rezervne kopije SD kartice • Windows	69
Poglavlje 21 • Oporavak SD kartice - Windows	71
Deo 5 • Pinovi, GPIO i kako ih kontrolisati pomoću Python-a.....	73
Poglavlje 22 • Raspberry Pi pinovi, uloge i brojevi	74
Poglavlje 23 • Python na Command Line Interpreter-u	76
Poglavlje 24 • Python funkcije	80
Poglavlje 25 • Jednostavan Python program	85
Poglavlje 26 • Ožičavanje jednostavnog kola	91
Poglavlje 27 • Upravljajte LED-om pomoću GPIOZERO-a	94
Poglavlje 28 • Upravljajte LED-om pomoću biblioteke rpi.gpio	97
Poglavlje 29 • Pročitajte dugme pomoću GPIOZERO-a	100
Poglavlje 30 • Pročitajte dugme sa RPi.GPIO.....	102
Poglavlje 31 • Kontrolišite LED pomoću dugmeta.....	104
Poglavlje 32 • Podesite DHT22 senzor pomoću Git-a	107
Poglavlje 33 • Koristite DHT22 senzor	111
Deo 6 – Postavljanje steka veb aplikacije	114
Poglavlje 34 • Stek veb aplikacije	115
Poglavlje 35 • Python virtuelno okruženje.....	119
Poglavlje 36 • Povećanje veličine swap datoteke stranične diska.....	122
Poglavlje 37 • Postavljanje sistemskog Pythona – priprema	124
Poglavlje 38 • Preuzmite, kompajlirajte i instalirajte Python 3	125
Poglavlje 39 • Podesite Python virtuelno okruženje aplikacije.....	129
Poglavlje 40 • Instaliranje Nginx-a	131
Poglavlje 41 • Instaliranje aplikacije Flask	133
Poglavlje 42 • Pregled jednostavne aplikacije Flask.....	136
Poglavlje 43 • UWSGI instalacija	138
Poglavlje 44 • Nginx konfiguracija	140
Poglavlje 45 • UWSGI konfiguracija	143
Poglavlje 46 • UWSGI i Nginx testiranje konfiguracije.....	146
Poglavlje 47 • Konfigurišite systemd za automatsko pokretanje uwsgi-a	148
Deo 7 – Postavljanje baze podataka	151
Poglavlje 48 • Instaliranje baze podataka SQLite3	152

Poglavlje 49 • Rad u SQLite3 CLI-u.....	154
Deo 8 – Stilizovanje pomoću Skeleton-a	158
Poglavlje 50 • Statički elementi i Skeleton CSS šablon	159
Poglavlje 51 • Podesite direktorijum statičkih elemenata	162
Poglavlje 52 • Uvod u Skeleton CSS šablon	164
Poglavlje 53 • Kopiranje datoteka pomoću SFTP-a	168
Poglavlje 54 • Flask šabloni	172
Poglavlje 55 • Debugovanja u Flask aplikaciji	176
Deo 9 – Hvatanje i snimanje podataka senzora	179
Poglavlje 56 • Uvod u Deo 9.....	180
Poglavlje 57 • Instalirajte DHT biblioteku i modul rpi-gpio	181
Poglavlje 58 • Prikažite trenutne vrednosti senzora u pregledaču	184
Poglavlje 59 • Kreirajte bazu podataka za čuvanje podataka senzora	189
Poglavlje 60 • Snimate podatke senzora pomoću Python skripta	191
Poglavlje 61 • Planirajte očitavanja senzora pomoću Cron-a	194
Poglavlje 62 • Ažurirajte datoteku aplikacije i šablona	196
Deo 10 - Implementirajte funkciju za odabir vremenskog opsega	201
Poglavlje 63 • Uvod u Deo 10.....	202
Poglavlje 64 • Prototip opsega datuma i vremena u zapisima u SQLite CLI-u.....	204
Poglavlje 65 • Prototip opsega datuma i vremena u pregledaču.....	206
Poglavlje 66 • Validacija znakovnog niza upita URL-a	210
Poglavlje 67 • Brzo čišćenje	214
Poglavlje 68 • Koristite radio-dugmad za lak odabir opsega datum i vremena.....	218
Poglavlje 69 • Obezbedite Python skript za rad pomoću radio-dugmadi	221
Deo 11 – Google Charts i vidžeti za datum i vreme	224
Poglavlje 70 • Uvod u Deo 11	225
Poglavlje 71 • Implementirajte Google Charts	229
Poglavlje 72 • Testirajte Google Charts	234
Poglavlje 73 • Vidžet birača datuma i vremena.....	236
Poglavlje 74 • Implementirajte vidžet za odabir datuma i vremena	238
Poglavlje 75 • Testirajte vidžet birača datuma i vremena	241
Deo 12 – Upotreba vremenskih zona	243

Poglavlje 76 • Prilagodite datum i vreme lokalnoj vremenskoj zoni na strani klijenta	244
Poglavlje 77 • Uvod u Arrow	248
Poglavlje 78 • Implementacija Arrow funkcije	251
Poglavlje 79 • Otpremanje izmena vremenske zone i testiranje	254
Poglavlje 80 • Povežite dve stranice aplikacije	256
Deo 13 – Crtanje pomoću Plotly-a	259
Poglavlje 81 • Šta je Plotly i kako ga instalirati	260
Poglavlje 82 • Isprobajte Plotly u CLI-u	264
Poglavlje 83 • Implementirajte Plotly podršku na strani klijenta	267
Poglavlje 84 • Dodajte Plotly podršku na strani servera	272
Poglavlje 85 • Kako debugovati Javascript	276
Poglavlje 86 • Primer debugovanja na serveru	281
Deo 14 - Pristupite aplikaciji sa Interneta	284
Poglavlje 87 • Kako pristupiti aplikaciji sa Interneta?	285
Poglavlje 88 • Podesite statičku IP adresu	287
Poglavlje 89 • Izložite aplikaciju Internetu pomoću prosleđivanja porta	290
Poglavlje 90 • Kreirajte samopotpisani sertifikat za prijavljivanje	295
Poglavlje 91 • Izmenite Nginx konfiguraciju da biste koristili SSL	300
Poglavlje 92 • Testirajte SSL u Firefox-u, Safari-ju, Chrome-u	303
Deo 15 - Evidentiranje podataka pomoću Google Sheet-a	307
Poglavlje 93 • Šta je evidentiranje podataka i zašto Google Sheet?	308
Poglavlje 94 • Podesite Google API akreditive	311
Poglavlje 95 • Podesite Python biblioteke i Google Sheet	322
Poglavlje 96 • Implementacija evidentiranja podataka Google Sheet-a	328
Deo 16 - Postavite udaljeni Arduino čvor senzora pomoću modula nRF24	330
Poglavlje 97 • Zašto treba postaviti udaljeni Arduino čvor senzora?	331
Poglavlje 98 • Ožičenje Arduino čvora	333
Poglavlje 99 • Skica Arduino čvora	336
Poglavlje 100 • Ožičenje Raspberry Pi-a i modula nRF24	340
Poglavlje 101 • Skript prijemnika Raspberry Pi nRF24	343
Poglavlje 102 • Kako instalirati Python nRF24 module na Raspberry Pi-a	348
Poglavlje 103 • Testirajte nRF24 komunikaciju	353

Poglavlje 104 • Izmenite frontend aplikacije tako da prikazuje podatke udaljenih čvorova	355
Deo 17 – Upozorenja If This Then That	359
Poglavlje 105 • Uvod u If This Then That.....	360
Poglavlje 106 • Kreirajte IFTTT webhook i aplet.....	362
Poglavlje 107 • Dodavanje IFTTT koda u aplikaciju i testiranje.....	371
Poglavlje 108 • Instalirajte skript osluškivača čvora kao sistemsku uslugu	374
Deo 18 - Završetak.....	376
Poglavlje 109 • Ažurirajte stranicu lab_en v_db svakih 10 minuta	377
Poglavlje 110 • Rezime i šta dalje	378
Deo 19 - Proširenje projekta: Razmena tekstualnih poruka pomoću usluge Twilio	380
Poglavlje 111 • O čemu se radi u ovom proširenju projekta?	381
Poglavlje 112 • Uvod u Twilio.....	383
Poglavlje 113 • Podesite Twilio nalog	385
Poglavlje 114 • Kreirajte korisni bash shell skript.....	391
Poglavlje 115 • Dodajte Twilio podršku na Raspberry Pi-u	394
Poglavlje 116 • Instalirajte Twilio CLI	396
Poglavlje 117 • Kreirajte lokalne i javne DNS nazive hostova	400
Poglavlje 118 • Kreirajte pouzdani SSL/TLS sertifikat.....	405
Poglavlje 119 • Pošaljite tekstualne poruke upozorenja.....	413
Poglavlje 120 • Primajte komande tekstualnih poruka	416

Da li ste našli grešku?

Molimo vas da nas obavestite.

Uz pomoć veb brauzera idite na <http://txplo.re/rpifsp> i ispunite obrazac.

Odmah ćemo popraviti.

O autoru

Dr Peter Delamaris je pedagog, inženjer elektronike i stvaralac. Kreator je “Uradi Sam” video kurseva iz elektronike koji se odvijaju uživo i pisac nekoliko tehničkih knjiga. Peter je skoro izdao svoju knjigu “Stvaranje obrazovne revolucije“, o tome kako stvaralaštvo menja način na koji učimo u 21. Veku.

Kao vođa tehničkih istraživanja od 2013 u Tech Explorations, kompanije koju je osnovao u Sidneju, Australija, Peterova misija je da istražuje tehnologiju i obrazuje svet.

Njegova firma nudi obrazovne kurseve i centre za obuku za elektronske hobiste, studente i nastavnike nauke, tehnologije, inženjeringa i matematike (STEM).

Kao stalni učenik, Peterova osnovna veština leži u objašnjavanju teških pojmova preko videa i pisanja. Sa preko 15 godina iskustva predavanja za treći stepen, Peter je razvio jednostavan ali jasan stil učenja koji cene studenti širom sveta.

Njegova strast za tehnologiju i svet hardvera ‘Uradi Sam’ otvorenog koda je bio glavni razlog koji ga je vodio kroz sopstveni razvoj i rad na tehničkim istraživanjima.

Kako čitati ovu knjigu

Svi primeri, opisi i postupci su testirani na Raspberry Pi Zero W, koji pokreće Raspbian Buster. Takođe sam testirao potpuno isti projekat na Raspberry Pi 4, Raspberry Pi 3 i Raspberry Pi 1.

Ova knjiga ima veb stranicu sa resursima koji su dizajnirani da maksimalno povećaju vrednost koja se isporučuje vama kao čitaocu. Pročitajte informacije o veb stranici knjige, šta ona nudi i kako joj pristupiti u odeljku „Veb stranica knjige“, kasnije u ovom uvodnom segmentu.

Na kraju, možda će vas zanimati verzija ove video datoteke. Ovaj kurs sadrži detaljne demonstracije i objašnjenja Full Stack projekta. Video predavanja sadrže tehnike i postupke koje jednostavno nije moguće primeniti u tekstu.

Na veb stranici knjige potražite novosti o ovom projektu. Obavezno se pretplatite na listu e-pošte kompanije „Tech Explorations“ da biste mogli da dobijate novosti.

Zahtevi

Da biste ovu knjigu iskoristili na najbolji način, biće vam potrebno nekoliko stvari. Verovatno ih već imate.

Lista hardverskih zahteva dostupna je na veb sajtu kompanije „Tech Explorations“:

<https://techexplorations.com/parts/rpifs-parts/>

Molimo vas da nabavite ove stavke pre nego što se upustite u ovaj projekat.

Veb stranica knjige

Kao čitalac ove knjige imate pravo na pristup njenim resursima na Internetu.

Ovim resursima možete pristupiti posetom veb stranice knjige na adresi <http://txplo.re/rpifsp>.

Dva dostupna resursa su:

1. Prostor za raspravu o knjizi u Tech Explorations zajednici. Na ovom mestu možete postavljati pitanja koja se odnosi na knjigu i da razgovarate o svojim projektima. Provediće vreme na forumu jednom nedeljno, odgovaraću na pitanja i učestvovaću u diskusijama.
2. Stranica sa greškama. Dok ispravljam greške, postavljaju informacije o ovim ispravkama na ovoj stranici. Proverite ovu stranicu ako sumnjate da ste pronašli grešku. Ako ste pronašli grešku koja nije navedena na stranici sa greškama, koristite obrazac za prijavu greške na istoj stranici da biste me o tome obavestili.

Povremeno ću objavljivati dodatne resurse i ažuriranja za Raspberry Pi Full Stack na ovoj stranici, zato vas molimo da je redovno proveravate. Ako se preplatite na listu e-pošte kompanije „Tech Explorations“, sigurno ćete dobijati moje redovne ispravke knjiga i vesti. Obrazac za pretplatu nalazi se na stranici knjige.

Da li ste pronašli grešku?

Obavestite nas.

Pomoću pregledača posetite stranicu <http://txplo.re/rpifsp> i pošaljite upit.

Molimo vas pripazite da navedete dovoljno detalja u upitu da bih mogao da pronađem grešku. Ne zaboravite da navedete broj stranice u PDF verziji knjige.

Odmah ću ispraviti greške.

Deo 1 • Početak rada u Raspberry Pi Full Stack-u

Poglavlje 1 • O čemu je ova knjiga?

Zamislite ovu knjigu kao vodič za avanturu odlučnih učenika.

Da, avantura.

Avantura hrabrosti, odlučnosti i saradnje.

Ako odlučite da ostanete na ovom putu, bićete nagrađeni znanjem.

I, sa mojim divljenjem.

Jer, ova avantura neće biti šetnja parkom.

Borićete se.

Nećete uspeti.

Mnogo puta.

Proklinjaćete me :-)

Ali, svaki put ćete ustati, obrisati prašinu i nastaviti putovanje da biste dovršili ovu full stack aplikaciju.

Biću tu da vam pomognem na svakom koraku.

Ja ću vas voditi i pokazati vam put.

Ali, moraćete da prođete naporan hod.

Moraćete da napišete kod, da zalemite strujna kola, da testirate veze i da pretražite bazu podataka.

Na kraju nećete steći samo znanje, već ćete se dobro osećati i znaćete da ste dovoljno snažni da pobedite sledeći izazov.

Avantura o kojoj govorim je projekat u kojem je vaše „oružje“ Raspberry Pi.

Vaš cilj je da kreirate korisnu aplikaciju koja funkcioniše na Raspberry Pi-u i obuhvata Internet, a vaš neprijatelj je vaše lenjo ja.

Iako je većina knjiga ispunjena mikro-projektima, ova knjiga je drugačija. Cela knjiga je jedan VELIKI projekat.

Više programskih jezika, bazu podataka, platforme Internet of Things i još mnogo što-šta.

Naučićete kako da koristite ove tehnologije za kreiranje nečega što stvarno funkcioniše.

Ovo će biti aplikacija sa mnogo pokretnih delova. Njena arhitektura i specifične komponenti koje sadrži su precizna analogija komercijalnih aplikacija Internet of Things u domenima kao što su Smart Homes, Smart Cities i Manufacturing.

Aplikacija koju ćete izraditi je skalabilna na globalnom nivou sa nekoliko ciljanih modifikacija i to je platforma na kojoj možete graditi.

Poglavlje 2 • Šetnja kroz projekat Full Stack

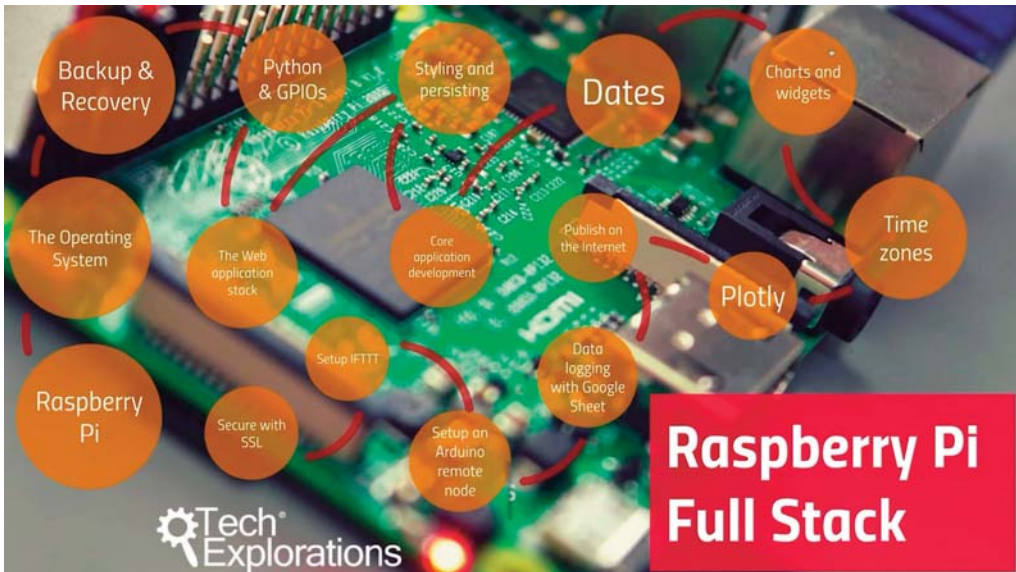
U redu, znate da će ovaj projekat biti sjajan.

Ali, šta ćete tačno učiniti?

Šta ćete tačno naučiti?

Drago mi je da ste pitali. Sada ćemo da „zaronimo“.

Na slici ispod prikazan je put kojim ćete ići kroz ovaj kurs (slika 2.1).



Slika 2.1 – Put kojim ćete ići kroz ovaj kurs

Putovanje započinje pogledom na osnovni hardver - Raspberry Pi. Aplikacija koju ćete na kraju kreirati dobro funkcioniše na svakom Raspberry Pi-u, međutim u ovoj knjizi sam se koncentrisao na Raspberry Pi Zero W (za „Wireless“). Raspberry Pi Zero predstavlja odličan odnos cene u poređenju sa performansama i veličinom. Zapanjen sam koliko možete postići pomoću ovog jeftinog računara. U video verziji ove knjige implementirao sam potpuno istu aplikaciju na Raspberry Pi 4. Implementacija aplikacije ima samo nekoliko malih razlika između Raspberry Pi 4 i Raspberry Pi Zero, koje ću naglasiti kada bude vreme za to.

U prvom odeljku knjige upoznaćete vas sa Raspberry Pi-em i fokusiraću se na Raspberry Pi Zero kao na osnovnu hardversku platformu. Ovaj osnovni hardver predstavlja najniži nivo Full Stack aplikacije.

U drugom delu knjige, tj. u drugom koraku projekta, učićete o operativnom sistemu Raspbian. Ovo je softver koji omogućava programerima aplikacija da implementiraju aplikaciju pomoću alatki visokog nivoa, poput programskog jezika Python i baze podataka SQLite3. Saznaćete da Raspberry Pi, kao pravi višenamenski računar, može da funkcioniše na širokim spektru operativnih sistema, uključujući različite varijante Linuxa, pa čak i na Microsoft Windowsu. U ovom projektu ćete koristiti minimalnu verziju preferiranog operativnog sistema Raspberry Pi, tj. Raspbian.

Pošto je važno samo „kada“, a ne „ako“ se dogodi katastrofa, **u trećem odeljku ove knjige pokazaću vam kako da kreirate rezervne kopije SD kartice na Raspberry Pi-u, a zatim kako da vratite podatke iz rezervne kopije datoteka.** Načiniti grešku tokom implementacije projekta zbog koje je teško ili čak nemoguće nastaviti je nešto što treba da očekujete kada učite nešto novo. Desilo mi se to mnogo puta. Ako ste spremni za takve pojave, uz redovno pravljenje rezervnih kopija, uvek ćete moći da vratite projekat u prethodno radno stanje i da nastavite odatle, umesto da ga ponovo izrađujete od „nule“.

Pravi rad započinje u četvrtom delu. Tamo ćete naučiti o Raspberry Pi pinovima General Purpose Inputs and Outputs (GPIO) i kako da ih koristite u jednostavnim Python programima. Objasniću vam kako da referencirate određeni GPIO i pokazaću vam osnovne Python komande koje su vam neophodne za obavljanje jednostavnih zadataka, poput uključivanja LED-a i očitavanja stanja dugmeta. Sve velike „stvari“ počinju prvim jednostavnim korakom, a za ovaj projekat prvi korak je treptanje LED diode.

U petom odeljku učićete o steku aplikacija i komponentama i servisima koji su prisutni na svakom nivou. Ovde ćete da kreirate radni okvir ovog projekta i naučiti ćete gde su u steku smeštene razne back-end i front-end tehnologije, kao što su Flask mikro-radni okvir za izradu Python veb aplikacija, uWSGI server aplikacija, Nginx veb server visokih performansi, SQLite3 server baze podataka, HTML/CSS/JavaScript/JQuery i, naravno, platforme Internet of Things, kao što su Googleovi različiti Cloud API-ji, IFTTT i Plotly. Da, naučićete kako da koristite sve ove tehnologije, ali ne sve odjednom. Proširićemo učenje na ceo projekat, tako da se možete opustiti.

U šestom delu ćete započeti izradu frontend aplikacije i obezbediti joj postojanost. Postojanost je mogućnost aplikacije da skladišti podatke i da ih kasnije preuzima. Ovde ćete postaviti prvu verziju jednostavne veb aplikacije na Flask Pythonu i dati joj osnovni (ali, elegantan) korisnički interfejs, tako što ćete koristiti HTML i CSS. Za postojanost ćete koristiti SQLite3, koji je jednostavan, ali moćan server baze podataka otvorenog koda. Od sedmog dela, pa nadalje, zaista ulazite u detalje procesa razvoja aplikacija.

U sedmom delu ćete kreirati „jezgro“ svoje aplikacije i razraditi najvažnije funkcije: implementiraćete integraciju senzora tako da možete da dobijete očitavanja i da ih sačuvate u bazi podataka, a zatim da automatizujete merenja da bi bila započeta evidencija podataka.

U osmom delu ćete naučiti kako da koristite datume u bazi podataka, kako da preuzmete podskup zapisa senzora iz baze podataka pomoću znakovnog niza upita URL-a i kako da integrišete grafičke vidžete u frontendu aplikacije da biste krajnjem korisniku olakšali da izabere opsege datuma i vremena.

Sada ste u najisplativijem i najbržem delu projekta u čijim ćete poglavljima implementirati nove funkcije.

U devetom delu ćete naučiti kako da kreirate grafikone pomoću sačuvanih podataka i kako da grafički prikazete više tačaka podataka u opsezima datuma i vremena koji se mogu izabrati.

U desetom delu ponovo ćete pogledati temu o datumu i vremenu koju ste započeli u osmom delu, ali ovog puta u frontendu. Naučićete kako da koristite JavaScript/JQuery i Python za pretvaranje datuma i vremena uskladištenih u bazi podataka vaše aplikacije u tačan datum/vreme za vašu vremensku zonu. Naravno, kada dovršite implementaciju ove funkcije, konverzija će se dogoditi automatski, bez ikakvog korisničkog unosa.

U jedanaestom delu ponovo ćete pogledati grafički prikaz podataka i naučićete kako da kreirate grafikone temperature i vlažnosti pomoću Plotly-a. Ovo je prilika da naučite kako da stupite u interakciju sa moćnim Cloud servisom i kako da povećate vrednost svoje aplikacije pružajući korisniku priliku da obrađuje podatke o svom okruženju pomoću moćnih Plotly alatki za grafičku analizu.

U dvanaestom delu ćete naučiti kako da pristupite svojoj Raspberry Pi Full Stack aplikaciji sa bilo kog mesta na svetu. Na taj način ćete moći da proverite svoju aplikaciju na telefonu bez obzira na to gde se nalazite u svetu. Većina ljudi bi se ovde zaustavila i uzela slobodan dan, ali ne i mi. Imamo još posla.

U trinaestom delu ćete naučiti da beležite podatke sa vaših senzora u Google tabelu pomoću API-ja Google Drive i Sheet. Ovaj posao će vam dati jedinstveni uvid u to kako možete da integrišete sofisticirane Cloud servise, kao što su Google Drive i Google Sheet, uključujući zbunjujuće procese za kreiranje i upotrebu njihovih sertifikata za autentifikaciju i šifrovanje.

U četrnaestom delu ćete naučiti kako da proširite Raspberry Pi Full Stack aplikaciju, tako što ćete (bežično) povezati Arduino čvorove sa sopstvenim senzorom. Zahvaljujući ovom proširenju, vaš Raspberry Pi će snimati podatke iz proizvoljnog broja Arduino čvorova (kao i iz sopstvenog senzora) u lokalnoj bazi podataka i u Google Sheet-u.

Uz novoprosirenu aplikaciju iz četrnaestog dela „uskoćićete“ u petnaesti deo i naučićete kako da koristite još jedan popularni Cloud servis If This Then That (IFTTT). Naučićete kako da primenite podesivu aplikaciju za upozorenje e-poštom koja će vam poslati e-poruku kada temperatura ili vlažnost u nekom od čvorova vaše aplikacije premaši graničnu vrednost. Kada znate kako se to radi, moći ćete da integrišete bilo koju IFTTT mogućnost sa svojom aplikacijom. Neverovatno je šta možete postići u samo dva koraka (korak 1: „ako ovo“ (if this), korak 2: „onda ono“ (if then)).

Da biste zaključili ovaj projekat, u šesnaestom delu ćete naučiti kako da zaštitite svoju aplikaciju pomoću SSL-a. Ovo je posebno važno ako ste odlučili da svoju aplikaciju izložite Internetu, kao što ste naučili u dvanaestom delu. Ako implementirate bezbednosne komponente i konfiguraciju na veb serveru, moći ćete da pristupite stranici aplikacije pomoću bezbednog HTTPS protokola, umesto pomoću nešifrovanog HTTP-a. I, to je to, ovo je projekat u koji ćete „uploviti“. Da li ste spremni?

Poglavlje 3 • Potreban hardver

Ovaj projekat je praktičan. Iako ćete veći deo vremena tokom implementacije provoditi na softverskoj strani, prvo ćete morati da asemblirate hardver. Srećom, hardverski zahtevi su malobrojni i jednostavni.

U nastavku ćete pronaći listu hardverskih komponenata koje će vam biti potrebne. Koristite ovu listu kao brze reference, ali pogledajte listu delova na Internetu da biste bili sigurni da imate najnovije detalje o hardveru.

Listu delova na Internetu možete pronaći ovde: <https://techexplorations.com/parts/rpifs-parts/>

Pitanje koje često dobijam od onih koji uče o Raspberry Pi Full Stack-u je: „Koji Raspberry Pi i koji Arduino treba da koristim?“

Evo mojih smernica:

Koji Raspberry Pi?

Nije stvarno važno. Testirao sam ovaj projekat na originalnom Raspberry Pi 1 modelu A. U stvari, moj Raspberry Pi 1 i dalje koristi originalni kod za ovaj projekat, koji je postao snažniji od 2014. godine.

Takođe sam testirao aplikaciju pomoću Raspberry Pi-a 2, 3 i 4 (modela B), ali i pomoću Raspberry Pi Zero W. U svim slučajevima aplikacija funkcioniše besprekorno.

Vaš izbor Raspberry Pi-a za ovaj projekat svodi se na sledeća razmatranja:

- Da li imate rezervni Raspberry Pi?
- Da li vam smeta da sačekate još malo dok se ne završe kompilacija i instalacija softvera?
- Da li više volite da ne koristite novije i skuplje Raspberry Pi-ove (naročito Raspberry Pi 4 sa 4 GB RAM-a).
- Da li više volite manji otisak?

Kada završite aplikaciju i kada ona bude funkcionisala na vašem Raspberry Pi-u, biće teško „osetiti“ razliku u performansama između različitih Raspberry Pi-ova, sa izuzetkom Raspberry Pi-a 1. Raspberry Pi 1 je sporiji, što će krajnji korisnik i primetiti, pa ga ne bih preporučio.

Dakle, izuzećemo Raspberry Pi 1 iz ostatka ove rasprave.

Aplikacija je trenutno pokrenuta na Raspberry Pi 4, 3 i Zero i teško mogu da osetim razliku.

Zatim uzmite u obzir performanse svakog računara tokom razvoja. Instaliranje i kompajliranje C koda je računski intenzivno. Za kompajliranje Python interpretera i RF24 upravljačkih programa biće potrebno između 30 minuta i jednog sata.

Ovde Raspberry Pi 4 „pobeđuje“ sve druge Raspberry Pi-ove.

Brz je, a to znači da nećete imati vremena za kafu ili čaj tokom kompajliranja. Raspberry Pi Zero je najsporiji od modernih Raspberry Pi-ova zbog male brzine radnog takta i jednog jezgra. Ipak, performanse nisam smatrao nepodesnim. Upravo sam otišao po kafu ili sređivao moju e-poštu dok je Raspberry Pi Zero izvršavao kompilaciju.

Ako ste osoba koja se zalaže za vrhunske sirove performanse, izaberite Raspberry Pi 4. On je toliko snažan da možete lako pokrenuti aplikaciju Full Stack koristeći punu GUI verziju operativnog sistema, ali i druge aplikacije na njemu, poput aplikacije Workbench Automation Computer i druge.

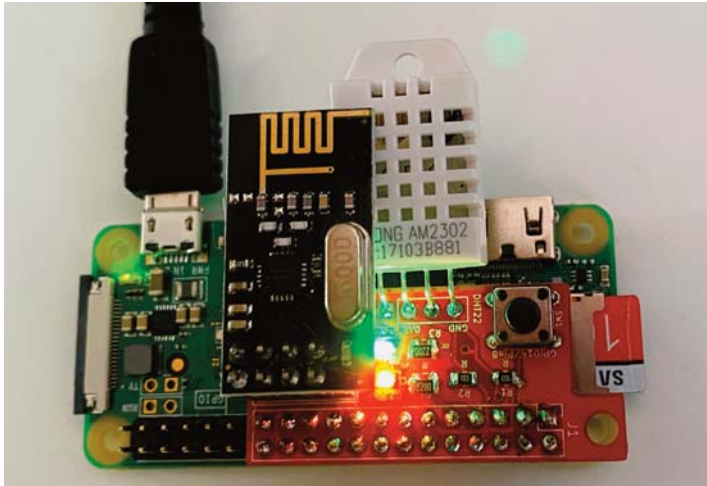
Raspberry Pi 4 je veoma brz, ali se takođe brzo zagreva. Ova toplota obično utiče na temperaturni senzor koji neprekidno pokazuje višu temperaturu na Raspberry Pi 4, iskrivljujući vaše podatke o temperaturi za jedan ili dva stepena Celzijusa.

Ako vas ovo zabrinjava i želite da osigurate preciznija merenja, trebalo bi da izbegnete Raspberry Pi 4 ili da jednostavno zanemarite podatke sa ugrađenog senzora i da umesto njih koristite podatke sa Arduino čvorova.

Ako vas ne oduševljava skup Raspberry Pi 4 i količina toplote koju on proizvodi, razmislite o Raspberry Pi Zero W (Wireless). Nije veoma brz, ali je vrlo jeftin, dovoljno brz tokom razvoja i ima mali otisak. Možete ga koristiti sa redovnim USB napajanjem (umesto sa skupljim i posebno dizajniranim napajanjem za Raspberry Pi 4).

Raspberry Pi Zero W je moja preporuka.

Kao što možete videti na slici 3.2, Raspberry Pi Zero W, sa prilagođenim projektom HAT koji sadrži senzor, dugme, LED-ove i RF24 primopredajnik, ima oko 7 cm dužine, 4 cm visine i 5 cm širine.



Slika 3.2 - Moj Raspberry Pi Zero W sa mojim prilagođenim projektom HAT

Emituje minimalnu toplotu i košta manje od 20 dolara. Ako nemate posebne razloge da koristite drugi model, izaberite Zero.

Koji Arduino?

Za Arduino su smernice mnogo jednostavnije. Samo koristite Arduino Uno ili nešto kompatibilno ili bilo koju Arduino ploču zasnovanu na Atmega328.

Testirao sam strujno kolo i skicu sa pločama Arduino Uno i Arduino Pro Mini, a rad u oba slučaja bio je besprekoran.

Lista hardvera

Potpuniju i ažurniju listu hardverskih komponenti potražite na adresi <https://techexplorations.com/parts/rpifs-parts/>. Sledi pojednostavljena lista koja je usredsređena na Raspberry Pi Zero W.

- Raspberry Pi Zero W
- Arduino Uno ili 100% kompatibilan
- SanDisk Ultra 8 GB (ili veći, ja koristim 16 GB) Class 10 UHS-I MicroSDHC memorijska kartica sa adapterom
- dva 10 K Ω otpornika (pull-up za dugme i pull-up za senzor Data pin)
- dva 330 Ω otpornika (za LED-ove)
- dva LED-a (izaberite željene boje)
- dva DHT22 senzora temperature i vlažnosti
- dva NRF24L01 + 2,4 GHz bežični RF primopredajnik
- trenutno dugme pogodno za prototipsku ploču
- mini prototipska ploča
- kratkospojnici

HAT PCB

Dizajnirao sam mali HAT koji preporučujem da koristite u svom projektu. Fotografiju ovog HAT-a možete videti na slici 3.2.

Otkrio sam da mi je korišćenje ovog Raspberry Pi HAT-a umesto prototipske ploče omogućilo brži rad aplikacije. Nisam morao da brinem o olabavljenim žicama kratkospojnika, a takođe izgleda sjajno. Ovde možete preuzeti kopiju njegovih Gerber datoteka i naručiti prototipske ploče od željenog proizvođača.

Možete i da posetite ovaj URL i da naručite PCB-ove iz mog deljenog projekta:

https://www.pcbway.com/project/shareproject/Raspberry_Pi_Full_Stack_RF24_and_DHT22_HAT.html

Ako odlučite da koristite prilagođeni PCB umesto prototipske ploče, neće biti potrebno da nabavljate prolazne komponente sa liste numerisane brojevima od 4 do 11. Umesto toga, pregledajte listu komponenata u mom zajedničkom projektu za SMD komponente koje će vam biti potrebno za lemljenje na PCB-u.

Poglavlje 4 • Kako dobiti pomoć

U kompaniji „Tech Explorations“ podržavamo studente kroz našu zajednicu Community Space.

Raspberry Pi Full Stack ima svoj prostor kojem možete pristupiti na adresi techexplorations.com/c/raspberry-pi-full-stack.

Ako ste ovu knjigu kupili direktno sa Tech Explorations veb sajta, trebalo bi da ste već dobili poziv da se pridružite našoj zajednici. Potražite ga u prijemnom sandučetu (takođe proverite direktorijum sa neželjenom poštom). U ovoj e-poruci ćete pronaći vezu za prihvatanje mog poziva. Kliknite na link da biste prihvatili pozivnicu i popunite obrazac da biste kreirali svoj besplatni nalog zajednice. Ako ste ovu knjigu kupili od jednog od naših partnera, i dalje se možete pridružiti našoj zajednici, ali moraćete da izvršite sledeće korake:

Pristupite našoj stranici za podršku na adresi <https://txplo.re/support>.

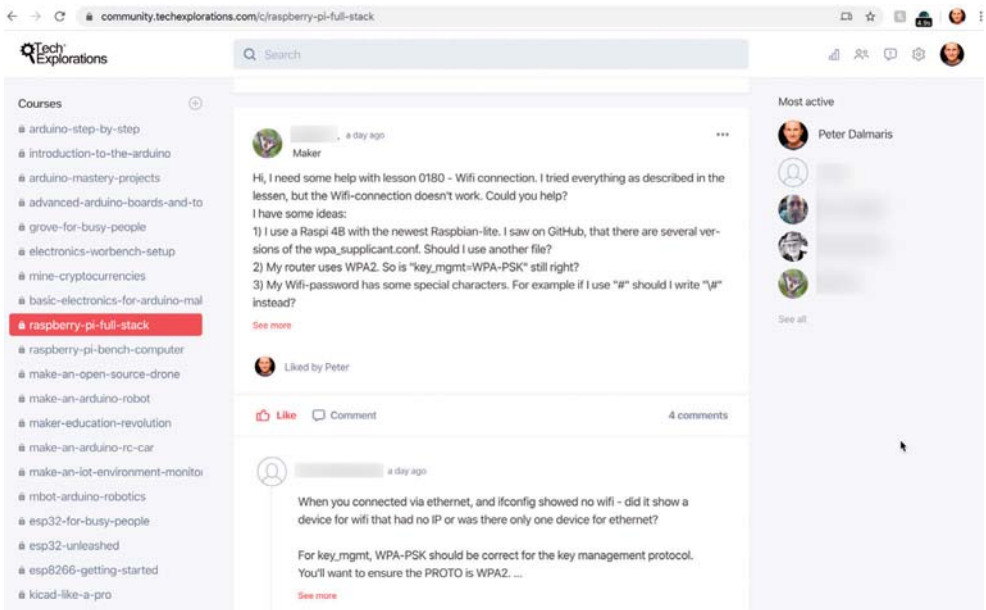
Kliknite na opciju podrške „Join the Makers Club“.

U polju „Where did you purchase your course“ izaberite najprikladniju opciju sa liste.

U tekstualnom polju „Comments“ označite da želite da postanete član zajednice Raspberry Pi Full Stack. Važno: kopirajte detalje o vašoj porudžbini knjige.

U polje „Email configuration to“ unesite adresu e-pošte na koju želite da vam pošaljemo pozivnicu.

To je to. Molimo vas da nam omogućite do 24 sata da obradimo vaš zahtev.



Slika 4.3 – Community Space za projekat Raspberry Pi Full Stack

Poglavlje 5 • Spremište koda

Izvorni kod za ovaj projekat možete pronaći u spremištu koda kursa na Github-u.

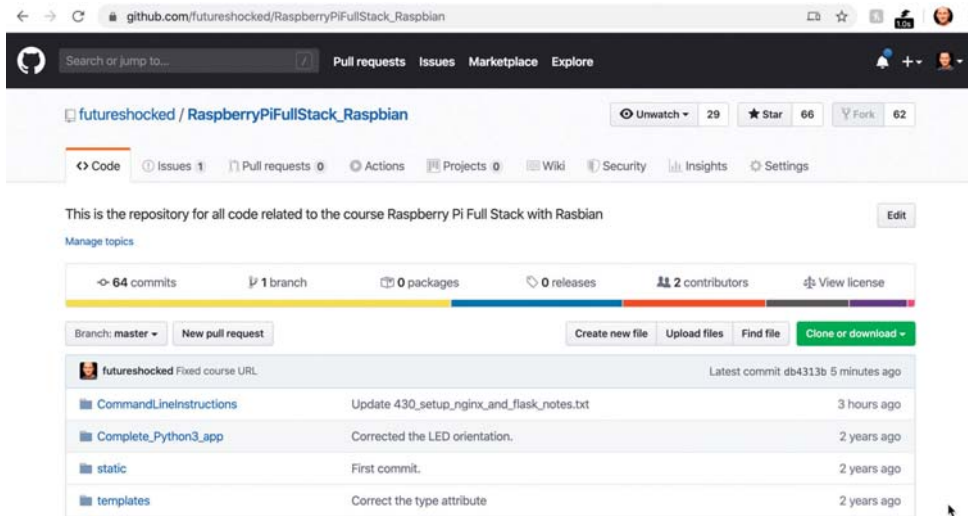
Zabeležite ovaj URL: https://github.com/futureshocked/RaspberryPiFullStack_Raspbian

Spremište sadrži izvorni kod u različitim fazama razvoja. Možete da kopirate ovaj kod u svoj projekat, umesto da ga kucate. Ovo će vam uštedeti vreme i frustracije koje su neizbežne pri kucanju.

Osim Python koda, spremište takođe sadrži moje sesije komandne linije i razne konfiguracione datoteke. Testirao sam svaku od ovih datoteka i uveren sam da funkcionišu.

Kada naiđete na problem, proverite izvorni kod na Github-u da biste se uverili da on funkcioniše i pomoću njega pokušajte da identifikujete problem u kodu.

Imajte na umu da je kod u spremištu „živ“. To znači da ga često ažuriram da bih popravio greške ili poboljšao funkcionalnost. Verovatno ćete pronaći razlike između koda koji sam napisao u ovoj knjizi i „živo“ koda u spremištu.



Slika 5.4 - Spremište koda Raspberry Pi Full Stack projekta na Github-u

Deo 2 • Raspberry Pi, Arduino i Raspberry Pi Zero W

Poglavlje 6 • Poređenje Raspberry Pi-a i Arduina na visokom nivou

Za mnoge od nas Arduino je tehnologija koja nas je uvela u svet programabilne elektronike. Relativno ju je jednostavno naučiti i „prašta“ mnoge uobičajene početničke greške, sa čitavim univerzumom dokumentacije, primera koda, strujnih kola i projekata.

Počinjemo da istražujemo druge tehnologije kada osetimo da smo shvatili bar osnove Arduina. Za mnoge od nas sledeći logičan korak je Raspberry Pi. Raspberry Pi otvara nove mogućnosti, posebno kada je reč o povezivanju projekata sa naprednim onlajn funkcijama ili računski intenzivnim funkcijama.

Pitanje koje redovno dobijam odnosi se na razlike i sličnosti između Raspberry Pi-a i Arduina. Mislim da je to vrlo dobro pitanje, jer kada shvatite razlike i sličnosti, moći ćete da smislite načine na koji oni mogu da funkcionišu zajedno u jednostavnom projektu.

Raspberry Pi i Arduino, ili, tačnije, Arduino Uno, su komplementarne tehnologije. Iako dele neke zajedničke karakteristike, razlikuju se u gotovo svakom aspektu koji je važan. Pa ipak, zbog svojih različitosti savršeno nadopunjuju jedno drugo.

U ovom poglavlju predstavljam poređenje na visokom nivou između Raspberry Pi-a i Arduina, zasnovano na tabeli sa slike 6.5.

Raspberry Pi	Arduino
Microcomputer	Microcontroller
Needs an operating system	Does not need an operating system
Complicated	Simple
Video out, Camera, Ethernet ports, Wifi, Bluetooth, USB, I2C, SPI, UART etc. on board	USB only for power and serial in/out, I2C, SPI, UART
Best for general computer	Best for small tasks that constantly repeat
Capable of performing a huge range of tasks	Optimised for sensing and controlling the world around it
Best for more advanced makers	Best for beginners
Programmed in many languages, including C/C++, Python, Ruby	Programmed in C/C++
Relatively high power consumption	Relatively low power consumption

Slika 6.5 - Poređenje na visokom nivou između Raspberry Pi-a i Arduino Uno-a

Dakle, evo 10 najvažnijih razlika između Raspberry Pi-a i Arduina.

Mikroračunar u poređenju sa mikrokontrolerom

Raspberry Pi je uobičajeni računar, sličan onome na vašem stolu. Isporučuje se u malom pakovanju. Raspberry Pi ima sve uobičajene funkcije i mogućnosti koje očekujete od računara.

Na primer:

- Možete povezati veliki monitor visoke definicije koji može reprodukovati 1080p video.
- Možete da povežete tastaturu, miš i spoljne čvrste diskove.
- Može se povezati na Internet pomoću Ethernet-a i WiFi-a.
- Zahteva operativni sistem i može pokrenuti vaš Linux ili Windows.
- Za mali računar ima respektabilne specifikacije, uključujući 64-bitni četverojezgarni procesor sa hardverskim video ubrzanjem i 4 GB RAM-a.
- Ima više USB konektora za periferni uređaje.

Zbog cene i veličine Raspberry Pi-a ove specifikacije su impresivne.

S druge strane, Arduino je potpuno drugačija mašina. Najvažnija razlika između Arduina i Raspberry Pi-a je u tome što Arduino nije računar. To je mikrokontroler.

Zamislite mikrokontroler kao vrlo mali računar bez ičega što očekujemo od desktop ili prenosivog računara.

Uzmite RAM kao primer i fokusirajte se na Arduino Uno. Arduino Uno ima malu količinu RAM-a, samo 2 kilobajta, i ne može da se poveže na mrežu, kao što je Internet, bez dodatnih komponenti. Uporedite ovo sa 4 gigabajta RAM-a koji su dostupni na Raspberry Pi 4!

Arduino Uno ima samo 32 kilobajta fleš memorije za čuvanje vašeg programa, dok Raspberry Pi može da koristi veoma velike količine prostora za skladištenje na uklonjivim SD karticama (a takođe možete da povežete terabajte spoljnog diska pomoću USB-a).

Druga velika razlika su performanse. Arduino Uno pokreće 8-bitni mikrokontroler od 16 MHz koji sadrži jednojezgarni CPU. Raspberry Pi pokreće 64-bitni četverojezgarni ARM CPU sa radnim taktom od 1,5 GHz. Poređenje je daleko od poštenog. Raspberry Pi može da obrađuje video zapise u realnom vremenu na aplikacijama poput prepoznavanja objekata i veštačke inteligencije, dok bi Arduino Uno bio razvučen do krajnjih granica pokušavajući da identifikuje jaku buku.

Arduino Uno ne koristi operativni sistem. Umesto toga, vaši programi su kompajlirani da bi se pokrenuli direktno na hardveru i zato se njihova kompajlirana verzija naziva „firmware“.

Kao što vidite, ova dva uređaja su potpuno različita. To je zato što je mikrokontroler dizajniran za obavljanje zadataka koji su mali, konstantni i koji se ponavljaju više puta. Mikroračunar je dizajniran kao uređaj opšte namene, sa vrlo velikim repertoarom mogućih primena.

Na primer, ako želite uređaj koji meri temperaturu i vlažnost u sobi, a zatim u zavisnosti od te dve vrednosti da uključite ventilator ili klima uređaj, onda bi nešto poput Arduina ili mikrokontrolera generalno bila najbolja tehnologija na kojoj možete da kreirate takav uređaj.

Ako želite nešto što može prepoznati lice osobe sa video strima, a zatim da pošaljete upozorenje ili da uključite alarm i da zabeležite taj događaj u onlajn bazi podataka, da pustite film na Netflixu i da proverite vašu e-poštu, onda će vam biti potreban Raspberry Pi.

Raspberry Pi je odličan za opšte računarske zadatke, dok je Arduino odličan za stvari koje su male složenosti, male veličine programiranja i neprestano se ponavljaju.

Odličan aspekt Arduina su svi njegovi ulazni i izlazni pinovi i mogućnost povezivanja sa aktuatorima i senzorima, ali i sa „stvarima“, kao što su releji, prekidači i motori, onda je Arduino zaista dobar izbor i optimizovan za detektovanje i kontrolu rada sveta oko njega.

Baš kao i Arduino, Raspberry Pi ima gomilu ulazno-izlaznih pinova opšte namene koje možete povezati sa aktuatorima i senzorima.

Dakle, definitivno treba istaći da Raspberry Pi zapravo može učiniti sve što može i Arduino, pa zato vam Arduino zapravo nije potreban. Ali, postoji i druga strana toga: složenost i robusnost.

Ono što je odlično u vezi sa Arduinom je što je jednostavan i „oprašta“ greške.

Činjenica da ne postoji operativni sistem zbog kojeg biste brinuli znači da ne morate da naučite kako da ga koristite samo da biste uključili relej.

Za početnike uvek kažem da je najbolji način da prvo počnu sa elektronikom i kontrolnim aplikacijama u Arduinu, da steknu razumevanje njegovog jednostavnog hardvera i veštinu izrađivanja koristeći minimalnu infrastrukturu koju mikrokontroler obezbeđuje pre nego što pređu na nešto komplikovanije.

Kada nova osoba uđe u svet elektronike i upravljanja, njeni ciljevi su prilično jednostavni; samo često treptanje LCD-a je odličan početak. Takav jednostavan početak značajno je postignuće koje može dovesti do mnogo većih „stvari“.

Ovakve „pobede“ možete vrlo brzo postići pomoću Arduina.

Programiranje

Još jedna velika razlika između Arduina i Raspberry Pi-a je programiranje.

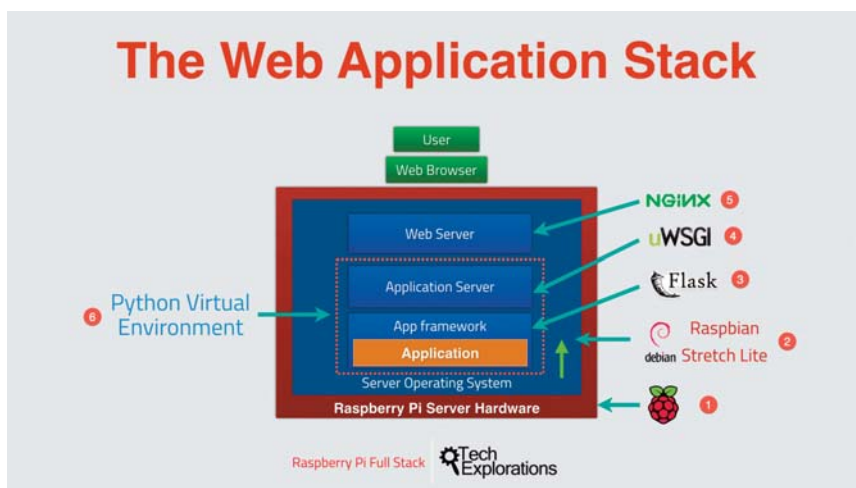
Arduino je programiran na jeziku C++. Uobičajeno, C++ nije jezik koji bih preporučio početnicima. Prilično je složen i mnogi koncepti i tehnike nisu intuitivni. Jezik poput Python-a ili Ruby-a je mnogo pogodniji kao programski jezik za početnike. Međutim, Arduino nije postao toliko popularan uprkos poteškoćama u učenju jezika C++.

Poglavlje 34 • Stek veb aplikacije

Dobrodošli u 6. deo ovog projekta. U poglavljima koja čine ovaj odeljak, pokazaću vam kako da primenite infrastrukturu vaše Full Stack aplikacije.

Ovo je savršeno vreme da naučite šta znači izraz „Full Stack“. Čineći to, razumećete zašto ova veb aplikacija zahteva toliko softverskih komponenti, koja je njihova pojedinačna uloga i kako se odnose jedna prema drugoj.

U ovom poglavlju ću vam prikazati pregled steka veb aplikacije. Počev od sledećeg poglavlja, implementiraćete stek veb aplikacije na vaš Raspberry Pi. U ostatku projekta postepeno ćete dodavati funkcije i mogućnosti, uvek u radnom okviru koji ćete kreirati 6. delu projekta.



Slika 34.51 - Grafički prikaz steka veb aplikacije

Na slici 34.51 možete videti prikaz steka veb aplikacije. U centru, prikazanom u obliku blokova, možete videti glavne komponente steka, kao što su operativni sistem, aplikacija koda i server aplikacije. Izvan blokova možete videti specifični softver koji sam izabrao da koristim i implementaciju za svaku komponentu. Na primer, izabrao sam da koristim uWSGI za implementaciju servera aplikacije, a NGInx za implementaciju veb servera.

Iako je arhitektura steka obično ista (ili veoma slična) među različitim implementacijama, tačne tehnologije korišćene za svaku komponentu mogu se značajno razlikovati. Ono što vam pokazujem u ovom projektu je samo jedan od mogućih izbora visokokvalitetnih tehnologija otvorenog koda. Kada počnete da shvatate kako stek funkcioniše, počecete da razmatrate druge opcije koje odgovaraju vašim specifičnim potrebama. Na primer, možete odlučiti da zamenite bazu podataka SQLite3 koju ćemo koristiti u ovom projektu (dobra je za mala opterećenja i aplikacije za jednog korisnika) bazom podataka koja dozvoljava veća opterećenja klijenta u proizvodnom okruženju.

Sada ćemo pregledati pojedinačne komponente koje čine stek veb aplikacije.

(1) Hardver

Na dnu steka je hardver. Ovde ćete pronaći procesor, memoriju i druge perifernijske uređaje. U ovom projektu, hardver se sastoji od Raspberry Pi Zero W. Mnogo toga što se dešava u hardverskom sloju apstrahuje operativni sistem, različiti upravljački programi i druge komponente steka, tako da mi, kao krajnji korisnici, ne moramo brinuti o detaljima. Moguće je zameniti Raspberry Pi Zero W drugim Raspberry Pi-em i bez promene bilo čega drugog, naša aplikacija će i dalje funkcionisati. Takođe možemo zameniti hardver za alternativu potpuno drugačijom arhitekturom, kao što je Beaglebone Black. Uz samo male izmene u ostatku steka, aplikacija će i dalje funkcionisati.

Iznenaden sam kada pomislim da je hardver, iako odgovoran za izvršavanje stvarnih poslova pokretanja aplikacije, komponenta steka zbog koje moram najmanje da brinem.

(2) Operativni sistem servera

Postoje dve velike vrste operativnih sistema:

- operativni sistemi servera
- desktop operativni sistemi

Desktop operativni sistem je obično onaj koji pokreće softver za produktivnost (ili igre!) koji će koristiti jedan korisnik. Ovo je softver poput veb pregledača, proračunskih tabela i klijenata e-pošte. Ovi operativni sistemi su optimizovani za pokretanje grafičkog korisničkog interfejsa sa brzom reakcijom na korisničke događaje, poput klika na dugme.

Operativni sistem je optimizovan da odgovori na zahteve klijenta koji stižu pomoću mrežnog interfejsa. Pošto ovi zahtevi dolaze iz mrežnog interfejsa (ne sa tastature ili miša), a odgovori se takođe usmeravaju do klijenta pomoću istog mrežnog interfejsa (ne sa ekrana), operativni sistem servera ne koristi grafički korisnički interfejs. Njegovi resursi su posvećeni pokretanju servisa potrebnih za odgovor na zahteve klijenata, kao što su baze podataka i veb serveri.

Aplikacija koju izrađujete u ovom projektu je u osnovi veb aplikacija. Krajnji korisnik će pomoću pregledača uputiti HTTP zahtev serveru aplikacija. Ovaj zahtev će se realizovati pomoću mrežnog interfejsa, a ne pomoću tastature i/ili miša. Ovaj scenario se najbolje uklapa sa slučajem korišćenja za operativni sistem servera. Zbog toga sam izabrao da koristim Raspbian Lite, umesto pune verzije Raspbian-a sa grafičkim komponentama.

Naravno, uzmete u obzir da su moderni operativni sistemi izuzetno svestrani. Možete uzeti Raspbian Lite i instalirati nedostajuće GUI pakete da biste ga pretvorili u puni desktop operativni sistem, a da i dalje koristite svoju veb aplikaciju. Možete da uzmete puni Raspbian i da ga koristite za pokretanje veb aplikacije. Sve je moguće, ali u svakom slučaju postoji optimizacija prema jednom od dva glavna slučaja korišćenja.

(3) Radni okvir aplikacije

Radni okvir aplikacije je kolekcija softvera, šablona i odluka osmišljenih da vam pomognu da brzo kreirate aplikaciju. Postoji mnogo radnih okvira aplikacija koje možete koristiti za izradu gotovo svega: veb aplikacija, igara, mobilnih aplikacija, aplikacija veštačke inteligencije itd.

Kada je reč o radnom okviru za izradu veb aplikacija, možemo da biramo. Postoje radni okviri, kao što su Django, Turbogears i web2py (samo tri od mnogo desetina), koji omogućavaju da brzo kreirate potpuno funkcionalne veb aplikacije.

Postoje mnogi drugi radni okviri veb aplikacija za druge programske jezike, kao što su Ruby-on-Rails za Ruby, Angular za Javascript, ASP.NET za C# i mnogi drugi.

U ovom projektu ćete kreirati jednostavnu veb aplikaciju. Vaša aplikacija neće morati mnogo toga da uradi. Na primer, nema potrebe za autentifikacijom, e-trgovinom ili prilagođenim sistemom za skladištenje. Zbog toga vam neće biti potreban potpuno razvijen radni okvir za razvoj veb aplikacija, koji će, iako može „učiniti bilo šta“, zahtevati mnogo više vremena za učenje.

Zbog toga sam izabrao Flask „mikro-radni okvir“. Zovemo ga „mikro“, jer ima prednosti punog radnog okvira (efikasnost u vremenu razvoja), bez neželjenih softvera koji se isporučuju sa radnim okvirima koji žele da urade sve.

Pomoću Flaska možete da kreirate aplikaciju (poput one na ovom kursu) sa jednom Python datotekom i da naučite zašto to treba da uradite u roku od nekoliko minuta (a ne dana ili nedelja).

Full-stak aplikaciju izradićete u radnom okviru Flask. To je ono što je predstavljeno naranđastim okvirom „Application“ unutar plavog okvira „App framework“ označenim sa „3“ na slici 34.51.

(4) i (5) Server aplikacije i veb server

Možete da kreirate aplikaciju uz pomoć razvojnog radnog okvira, a možete i da pokrenete testnu instancu da biste je testirali. Međutim, da biste pravilno objavili svoju aplikaciju, biće vam potrebne dve dodatne komponente: server aplikacije i veb server (o veb serveru će biti reči kasnije u ovom poglavlju).

Server aplikacija je program koji ima jednu svrhu: upravljanje vašom veb aplikacijom. Iako je aplikacija koju gradite u ovom projektu mala i verovatno nikada neće imati više od jednog ili dva korisnika, ovo je prilika da razmislite o tome kako da kreirate aplikaciju koja potencijalno može imati hiljade korisnika. Server za veb aplikacije odgovoran je za upravljanje aplikacijom, tako da može da se poveća.

Na primer, zamislite da ste na svom Raspberry Pi-u izradili neverovatnu aplikaciju koju koristi mnogo ljudi. Jedan Raspberry Pi nije dovoljan da se nosi sa opterećenjem, pa dodajete još pet u svoj novi klaster. Server aplikacija je tu da vam pomogne u upravljanju ra-

Poglavlje 37 • Postavljanje sistemskog Pythona – priprema

U ovom poglavlju ćete instalirati razne pakete i alatke koji su neophodni kasnije u projektu. Svi ovi paketi potiču iz Debian spremišta u kojem postoji poseban podprojekat za Raspbian. U vreme dok pišem ovo poglavlje Buster je trenutna verzija Raspbian-a. Informacije o svim paketima koji su deo Buster-a možete pronaći ovde: <https://packages.debian.org/buster/>.

Na istom mestu možete pronaći informacije o paketima koje ćete instalirati u ovom poglavlju. Na primer, ako ste radoznali da saznate šta je tačno „build-essential“, pogledajte stranicu <https://packages.debian.org/buster/build-essential>.

Slično tome, ako želite da znate šta se nalazi u paketu „libncurses5-dev“, pogledajte <https://packages.debian.org/buster/libncurses5-dev>.

Za bilo koji od ovih paketa možete otići na Busterov osnovni URL <https://packages.debian.org/buster/> i na kraju dodajte tačan naziv paketa.

Počnite tako što ćete se prijaviti na Raspberry Pi kao korisnik „pi“.

```
$ ssh pi@192.168.111.63
```

Zatim pokrenite ove komande, jednu po jednu:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install build-essential
$ sudo apt-get install libncurses5-dev libncursesw5-dev libreadline6-dev
libffi-dev
$ sudo apt-get install libbz2-dev libexpat1-dev liblzma-dev zlib1g-dev
libsqlite3-dev libgdbm-dev libssl-dev openssl
$ sudo apt-get install libboost-python-dev
$ sudo apt-get install libpulse-dev
$ sudo apt-get install python-dev
$ sudo apt-get install vim
```

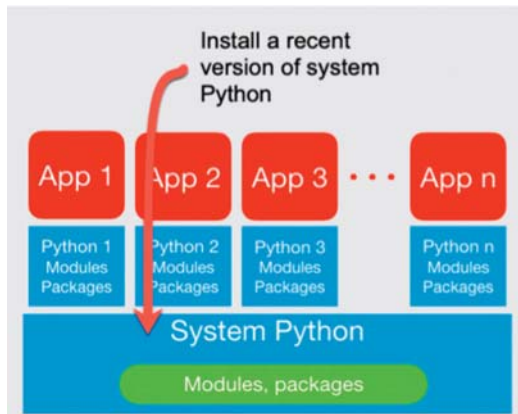
Alatka apt-get može predložiti dodatne radnje, kao što je pokretanje pomoću parametra „—fix-missing“. Obratite pažnju na izlaz u konzoli i sledite sve predložene savete da biste bili sigurni da su svi ovi moduli uspešno instalirani pre nego što nastavite dalje.

To je sve. U sledećem poglavlju instaliraćete najnoviju dostupnu Python verziju na sistemskom nivou. Zatim ćete ovaj Python koristiti za podešavanje virtuelnog Python okruženja za vašu aplikaciju.

>> *Ovo je pravi trenutak da napravite punu rezervnu kopiju SD kartice.*

Poglavlje 38 • Preuzmite, kompajlirajte i instalirajte Python 3

Raspbian se isporučuje sa interpreterom Python 3, međutim on je obično star nekoliko meseci. Poželjno je da koristite novije verzije softvera kad god je to moguće da biste iskoristili poboljšanja performansi i bezbednosti. Zbog tog razloga, savetujem vam da instalirate najnoviju dostupnu Python verziju na Raspberry Pi sistemu, a zatim koristite ovu verziju za podešavanje Python virtuelnog okruženja za vašu aplikaciju (slika 38.53).



Slika 38.53 - U ovom poglavlju ćete instalirati ažuriranu Python verziju na sistemskom nivou

U vreme dok pišem ovo poglavlje najnovija verzija Python-a je 3.9.0. Međutim, detaljno sam testirao ovaj projekat pomoću verzije 3.8.2. Da bi uputstva u ovoj knjizi bila konzistentna, koristiću 3.8.2 kao „zvaničnu“ verziju Python-a za ovaj projekat. Detaljno testiram novije verzije Python-a na otprilike svakih 6 meseci.

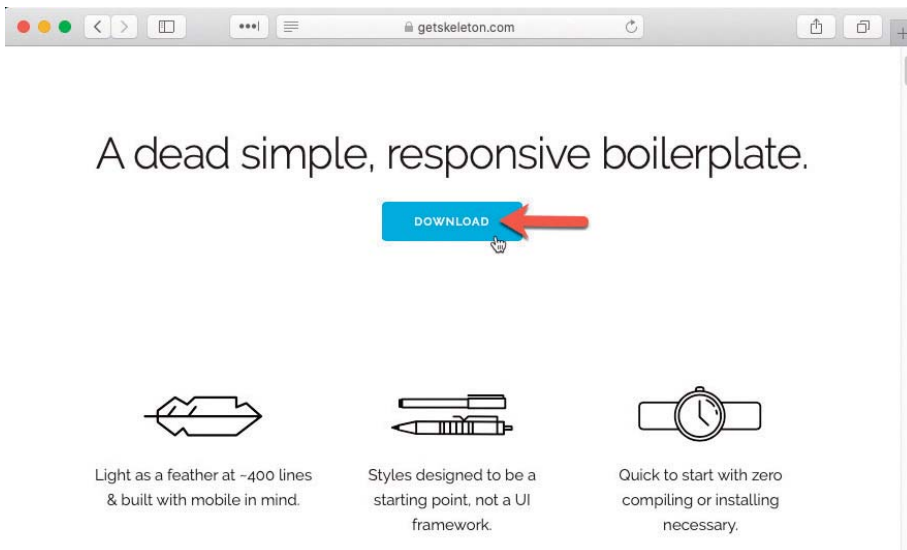
Da biste saznali koje su nove verzije Python-a dostupne posetite stranicu <https://www.python.org/ftp/python/>.

Poglavlje 52 • Uvod u Skeleton CSS šablon

U posljednjem poglavlju ste kreirali jednostavnu HTML datoteku koja se zove „a_static_file.html“. Veb pregledač će ga prikazati pomoću podrazumevanih postavki. Izgleda u redu, ali ne odlično.

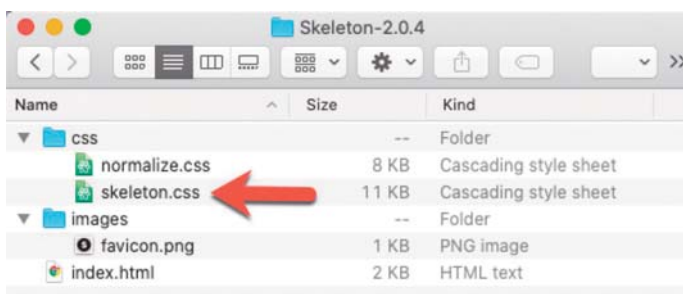
U ovom poglavlju i sledećih nekoliko poglavlja koristićete Skeleton da bi ova stranica izgledala odlično. Pokazaću vam kako da ga instalirate u svoju veb aplikaciju Python Flask i kako da konfigurirate datoteku a_static_file.html da biste mogli da ga koristite.

Prvo posetite stranicu getskeleton.com sa koje možete da preuzmete potrebne datoteke. Na početnoj stranici ćete pronaći dugme Download. Kliknite na njega da biste preuzeli ZIP datoteku šablona (slika 52.64).



Slika 52.64 - Kliknite na Download da biste preuzeli Skeleton datoteke šablona

Preuzimanje ZIP datoteke će trajati nekoliko sekundi. Kada se preuzimanje završi, proširite ga. Proširena arhiva sadrži dva direktorijuma: „css“ i „images“ (slika 52.65).



Slika 52.65 – Sadržaj Skeleton CSS arhive šablona

Unutar „css“ direktorijuma nalaze se dve css datoteke. Ona pod nazivom „skeleton.css“ je datoteka koja sadrži CSS kod koji će stilizovati stranice vašeg veb sajta.

Otvorite ovu datoteku u uređivaču teksta i pogledajte njen kod. Evo malog segmenta:

```
/* Typography
----- */
h1, h2, h3, h4, h5, h6 {
  margin-top: 0;
  margin-bottom: 2rem;
  font-weight: 300; }
h1 { font-size: 4.0rem; line-height: 1.2; letter-spacing: -.1rem;}
h2 { font-size: 3.6rem; line-height: 1.25; letter-spacing: -.1rem; }
h3 { font-size: 3.0rem; line-height: 1.3; letter-spacing: -.1rem; }
h4 { font-size: 2.4rem; line-height: 1.35; letter-spacing: -.08rem; }
h5 { font-size: 1.8rem; line-height: 1.5; letter-spacing: -.05rem; }
h6 { font-size: 1.5rem; line-height: 1.6; letter-spacing: 0; }
```

Ovaj kod stilizuje tagove zaglavlja i kontroliše veličinu fonta i razmak oko svakog zaglavlja. U ovoj datoteci ima mnogo CSS koda koji kontroliše izgled elemenata poput kolona, dugmadi, linkova, obrazaca i tabela.

Možete da naučite kako funkcioniše i izgleda Skeleton CSS ako pogledate primere dostupne na veb sajtu [Get Skeleton](#) i uporedite ih sa stvarnim kodom koji ih generiše. Na primer, na slici 52.65 možete videti kako izgledaju dugmadi u stilu Skeleton-a i CSS kod koji generiše rezultat.

Poglavlje 85 • Kako debugovati Javascript

Moderne veb aplikacije zavise od funkcionalnosti na strani klijenta koje su kodirane pomoću Javascripta. U kombinaciji sa bibliotekama, kao što je jQuery, Javascript kod može brzo postati složeniji. Dok se vaš Javascript kod komplikuje, povećava se rizik od nastanka grešaka.

Srećom, moderni veb pregledači su opremljeni programerskim alatka koje možemo koristiti za debugovanje Javascript koda. U ovom poglavlju ću vam pokazati kako da pomoću programerskih alatki u Chrome-u pronađete i otkrijete jednostavnu grešku u Javascriptu koji smo napisali za implementaciju Plotly funkcije.

Ako više volite da koristite neke druge moderne veb pregledače, kao što su Firefox, Safari i Edge, znajte da se isporučuju sa ekvivalentnim programerskim alatka. Zaključio sam da kada shvatim kako da ispravim Javascript kod u jednom pregledaču, mogu da se snađem koristeći alatke koje se isporučuju sa bilo kojim drugim pregledačem. Ova specifična veština se lako prenosi na platforme i operativne sisteme.

Da bih to demonstrirao, predstavicu grešku u Javascriptu u direktorijumu templates/lab_env_db.html. Za to možete koristiti vim:

```
lab_app) root@raspberrypi-zero:/var/www/lab_app# vim
templates/lab_env_db.html
```

Potražite ovu liniju u kodu:

```
<script src="//code.jquery.com/jquery-3.5.0.min.js"></script>
```

Izvršite sledeću izmenu (napisanu podebljanim tekstem):

```
<script src="//code.jquery.com/jquery
3.5.0.slim.min.js"></script>
```

Umesto da preuzmem kompletnu biblioteku jQuery, preuzimam slim verziju biblioteke. Ova verzija je manja po veličini za preuzimanje, ali joj nedostaje nekoliko funkcija, uključujući funkciju GET koja je našem Javascriptu potrebna za Plotly funkcionalnost.

Sačuvajte svoju „grešku“ verzije datoteke šablona.

Za primer debugovanja koristiću Google Chrome, kao što sam ranije pomenuo, možete slediti odgovarajući metod koristeći ekvivalentne programerske alatke u drugim modernim pregledačima.

Pokrenite Chrome i pristupite svojoj full stack aplikaciji. Za testiranje koristim ovaj URL:

```
http://raspberrypi-zero local/lab_env_db?from=2020-04-
24+00%3A00&to=2020-04-24+22%3A26&timezone=Australia%2FSydney
```

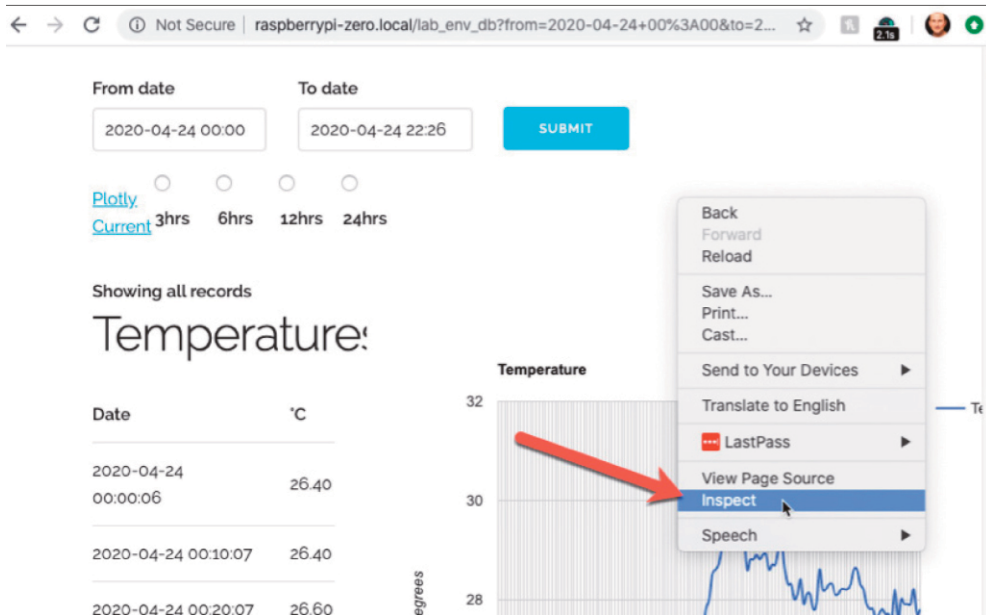
Izgleda u redu. I zaista i jeste.

Kliknite na Plotly link.

Čini se da aplikacija reaguje očekivano. Prikazaće se poruka „Sending data...“. Ali, poruka nestaje i nije zamenjena hiperlinkom URL-a grafikona.

Oh, Bože! Imamo grešku.

Uključimo programerske alatke pregledača. Desnim tasterom miša kliknite bilo gde na stranici i izaberite „Inspect“ (slika 85.105).



Slika 85.105 - Pokretanje programerskih alatki pregledača

Ovo će podeliti prozor pregledača na dva dela. Gornji deo prikazuje vašu veb stranicu. Na dnu se nalaze programerske alatke. Sada imate pristup programerskim alatkama, kao što su:

- Elements, koja vam omogućava da pregledate HTML kod stranice.
- Console, koja vam omogućava interakciju sa stranicom pomoću okruženja komandne linije.
- Sources, koja vam daje informacije o različitim datotekama koje čine vašu veb stranicu, kao što je HTML datoteka, i o različitim Javascript uvozima (kao što je jQuery).

Poglavlje 91 • Izmenite Nginx konfiguraciju da biste koristili SSL

U prethodnom poglavlju ste kreirali ove dve datoteke:

- Privatni samopotpisani ključ u `/etc/ssl/private/nginx-selfsigned.key`.
- Javni samopotpisani sertifikat u `/etc/ssl/certs/nginx-selfsigned.crt`.

U ovom tekstu ćete uređivati Nginx konfiguracionu datoteku aplikacije da biste omogućili veb klijentu da koristi HTTPS (za više detalja - <https://docs.nginx.com/nginx/technical-specs/>). Nginx konfiguracija se može modularizirati pomoću fragmenata datoteka. Ovo olakšava organizaciju različitih direktiva da ne bismo dobili ogromnu konfiguracionu datoteku.

Razdvojio sam proces konfiguracije SSL/HTTPS u četiri koraka. Pobrinite se da sve vreme radite kao superuser.

Korak 1 - Kreirajte isečak koda konfiguracije za datoteke ključeva i sertifikata

Koristite Vim da biste kreirali datoteku pod nazivom „self-signed.conf“, a zatim je sačuvajte u direktorijumu `/etc/nginx/snippets/`:

```
vim /etc/nginx/snippets/self-signed.conf
```

Kopirajte ovaj sadržaj u datoteku `self-signed.conf`:

```
ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;  
ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
```

Tako će Nginx moći da pronade SSL datoteke.

Korak 2 - Kreirajte drugi isečak koda pomoću SSL konfiguracije

Koristite Vim da biste kreirali datoteku pod nazivom „ssl-params.conf“, a zatim je sačuvajte u direktorijumu `/etc/nginx/snippets/`:

```
vim /etc/nginx/snippets/ssl-params.conf
```

Kopirajte ovaj sadržaj u datoteku `ssl-params.conf`:

```
ssl_protocols TLSv1.2;  
ssl_prefer_server_ciphers on;  
ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384;  
ssl_ecdh_curve secp384r1; # Requires nginx >= 1.1.0  
ssl_session_timeout 10m;  
ssl_session_cache shared:SSL:10m;  
ssl_session_tickets off; # Requires nginx >= 1.5.9  
ssl_stapling on; # Requires nginx >= 1.3.7
```

```

ssl_stapling_verify on; # Requires nginx => 1.3.7
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 5s;
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";

```

Ova datoteka postavlja različite SSL parametre, kao što je verzija dozvoljenih protokola (TLS v1.2), i šifre koje će se koristiti.

Korak 3 - Uredite Nginx konfiguracionu datoteku glavne aplikacije da biste koristili SSL

Glavnu Nginx konfiguracionu datoteku za našu aplikaciju možete pronaći ovde: /var/www/lab_app/lab_app_nginx.conf. Morate je urediti da biste omogućili HTTPS port (port 443) i povezali ga sa isečcima koda konfiguracije. Takođe imamo mogućnost blokiranja nešifrovanih zahteva na portu 80 ili jednostavno zadržite port 80 i dodajte mogućnost šifrovanog saobraćaja na portu 443.

Koristite Vim da biste otvorili nginx konfiguracionu datoteku:

```
vim /var/www/lab_app/lab_app_nginx.conf
```

Uredite njen sadržaj na sledeći način (novi sadržaj sam istakao podebljanim tekstom):

```

server {
listen 443 ssl;
listen [::]:443 ssl;
server_name localhost raspberrypi-zero.local 192.168.111.70;
include snippets/self-signed.conf;
include snippets/ssl-params.conf;

charset utf-8;
client_max_body_size 75M;

location /static {
root /var/www/lab_app/;
}

location / { try_files $uri @labapp; }
location @labapp {
include uwsgi_params;
uwsgi_pass unix:/var/www/lab_app/lab_app_uwsgi.sock;
}
}

server {
    listen      80;
    server_name localhost;

```

Poglavlje 112 • Uvod u Twilio

U ranijim poglavljima aplikaciji ste dodavali e-poruke sa upozorenjima pomoću veb usluge „If This Then That“.

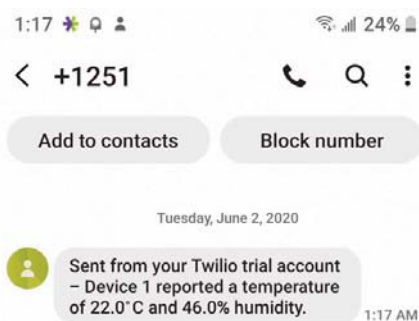
Sada ćemo proširiti aplikaciju tako da uključuje mogućnost podrške razmene tekstualnih poruka pomoću usluge koju obezbeđuje Twilio.

Prvo ćemo podesiti Twilio nalog i dobićemo telefonski broj koji će se koristiti za programske razmene tekstualnih poruka.

Zatim ćemo dodati podršku Twilio na Raspberry Pi-u, uključujući Twilio Command Line Interface (CLI) i potrebne poverljive podatke za komunikaciju sa Twilio serverom.

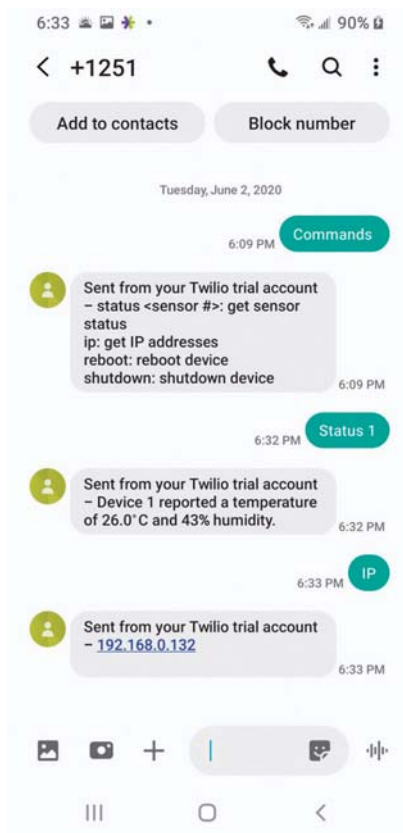
U ovom trenutku poboljšaćemo daljinski pristup Raspberry Pi-u postavljanjem javno dostupnog naziva domena i upotrebom pouzdanog SSL/TLS sertifikata za bezbednu HTTP-S komunikaciju, umesto da se oslanjamo na samopotpisani sertifikat. Ovo je potrebno za podršku dolaznih tekstualnih poruka, jer Twilio ne dozvoljava upotrebu samopotpisanih sertifikata za bezbedan pristup.

Zatim, aplikacija će biti poboljšana za slanje tekstualnih poruka upozorenja na sličan način na koji šalje e-poruke upozorenja (slika 112.173).



Slika 112.173 - Primer tekstualne poruke koju je poslala Full Stack aplikacija

Na kraju, aplikacija će biti dodatno poboljšana tako da pruža mogućnost upravljanja dolaznim tekstualnim porukama koje sadrže komande koje će se izvršavati na Raspberry Pi-u.



Slika 112.174 - Ovaj primer prikazuje razne poruke i operacije koje možete da primenite pomoću razmene tekstualnih poruka

Pomoću ove mogućnosti moći ćemo da zahtevamo trenutni status bilo kog senzora putem tekstualnih poruka. Server će odgovoriti na naš zahtev tekstualnom porukom koja sadrži trenutni status.

Dodatne komande će takođe biti implementirane samo da pokažu mogućnosti korišćenja ove tehnike.