

Vse spodaj naštete teme so že oddane, tu pa jih puščam kot primer.

Aplikacija za Android in/ali iOS za Slovensko turno kolesarsko pot

[Slovenska turno kolesarska pot](#) je 1800 km dolga pot po Sloveniji, ki teče (kjer je le možno) prek hribov, tako da skupno nabere 50.000 višinskih metrov. Opisi in zemljevidi etap, višinski profili, sezname kontrolnih točk in vsi ostali podatki so na spletni strani, ki pa ni najbolj praktična za uporabo na telefonih, ko smo prešvicani izgubljeni sredi klanca, po možnosti brez signala. Veliko lepše bi bilo imeti priročno aplikacijo, ki bi si podatke s strežnika osvežila, ko bi bil za to primeren čas, poleg tega pa bi bila organizirana bolj praktično kot spletna stran. Bonus točke: deljenje GPXov z drugimi aplikacijami, nameščenimi na telefonu.

Zaradi prihodnjega vzdrževanja aplikacije je zaželeno, da je njena koda lepa in da je njen "uradni" izvajalec (ki ima tudi možnost objavljanja novih različic) FRI. Seveda pa naj se vseeno ve tudi, kdo jo je v resnici razvil.

Zahtevana znanja: dobro znanje programiranja, programiranje mobilnih aplikacij, nekaj preprostejšega dela s podatkovno bazo

Zaledni sistem za program Odisej

[Odisej](#) je program za sestavljanje (skoraj besedilnih) pustolovskih iger, namenjena je predvsem osnovnošolcem za pouk računalništva (za primer glej kar [navodila](#)). Mentor ga je sprogramiral nekoč na dopustu. Teče v brskalniku (se pravi program, ne mentor), a brez komunikacije s strežnikom; napisana je v JavaScriptu ter uporablja React, Bootstrap in Blockly ([koda na githubu](#)). Njena osnovna pomanjkljivost je, da vse podatke shranjuje v brskalniku, ta pa se začne hitro pritoževati, čim vanjo shranimo nekaj slik. Sprogramirati bi bilo potrebno zaledni sistem (back end). Ta bi omogočal, da se uporabnik vpiše, odpira nove projekte (igre), shranjuje trenutno različico igre in pripadajoče datoteke (za zdaj: slike, nekoč tudi zvok). Da bo stvar delovala, bo seveda potrebno poseči tudi v že razviti front-ent, vendar lahko to vsaj delno naredi tudi mentor.

Ker bo sistem po študentovi diplomi in izginotju najbrž vzdrževal mentor, bodo potrebna pogajanja o jeziku in ogrođju. Mentor ima rad Node, Python in Kotlin, nima nič proti Rubyju (čepprav ga ne zna), ni pa fan Java. Med ogrođji so mu vseč lahkotnejša in daje, recimo, prednost Flasku pred Django.

Zahtevana znanja: dobro znanje programiranja, back-end, delo s podatkovno bazo.

Dodatek za Orange: Analiza omrežij

[Orange](#) je program za podatkovno rudarjenje. Ima tudi [dodatek za analizo omrežij](#) (v smislu grafov, ne računalniških omrežij). Ta dodatek kljub mentorjevemu velikemu zanimanju za to področje stagnira, saj je prej omenjeno zanimanje žrtev mentorjevih časovnih omejitev. Lepo bi bilo imeti študenta, ki bi tu dodal kaj novega. Za začetek očitno manjkajo algoritmi za izračun različnih lastnosti omrežij.

Zahtevana znanja: dobro znanje programiranja v Pythonu, poznavanje algoritmov na grafih.