1 Seminaras

Justas Mundeikis*

2019 m. vasario 14 d.

1 Programų instaliavimas

Instaliuokite šias programas savo asmeniniame kompiuteryje. Jeigu naudojatės VU kompiuteriais, gali prireikti instaliuoti nešiojamą (portable) versiją Sublime.

1. Instaliuokite R
2. Instaliuokite RStudio
3. Instaliuokite GitBash
4. Instaliuokite Sublime
5. Nepamirškite pagrindinių nustatymų

```
$ git config --global user.name "Vardas ėPavard"
$ git config --global user.email vardas.pavarde@evaf.vu.lt
$ git config --global core.pager cat
```

2 Paskyrų susikūrimas

1. Susikurkite savo paskyrą, naudodami bet kokį pseudonimą @Github, tačiau naudodami savo universitetinį email vardas.pavarde@stud.vu.lt

3 Nuosavo tinklapio sukūrimo projektas

Šioje dalyje naudosimės Git, Sublime ir Github, jog sukurti savo nuosavą internetinį puslapį.
Pastaba: Tarkime, kad mano studento nr yra "S175", Jūsų gali būti kitoks, tada naudojate savo vietoj mano "S175"

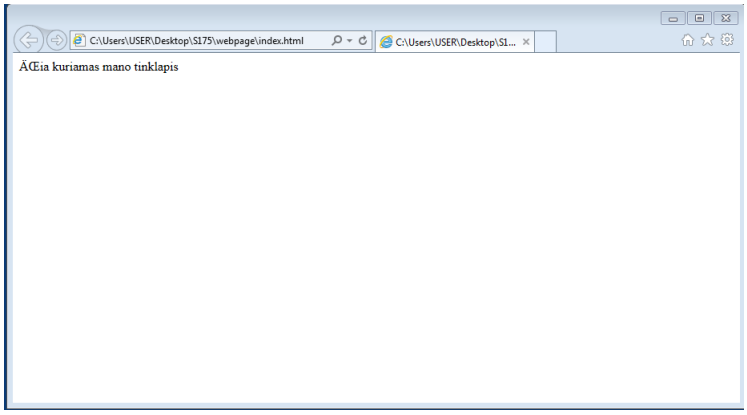
1. Su Git nueiname and Desktopo (arba tiesiog paleidžiama "Git Bash Here" [su dešiniu pelės mygtuku])
2. Sukuriame folderį webpage `mkdir -p S175/webpage`
3. Perėjus į patį webpage folderį, inicijuojame jį su git, taigi prima `cd S175/webpage`, o tada `git init`
4. Pirmas dalykas, kurį mums reiktų sukurti, tai bazinis html dokumentas kurio pavadinimas būtų *index* `touch index.html`
5. Jeigu tiesiog atsidarysite index.html failą su kuria nors naršykle, pamatysite, kad viskas yra absoliučiai balta! Neįdomu :D

*justas.mundeikis@evaf.vu.lt

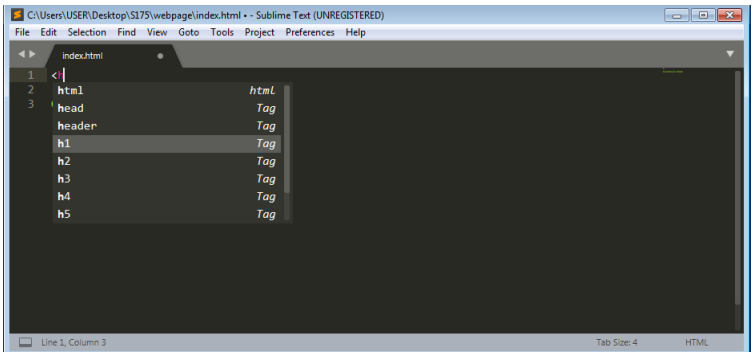
6. Atsidarome *index.html* su *Sublime* editoriumi
7. Editoriuje įrašome tekstą

```
Č
ia kuriamas mano tinklapis
```

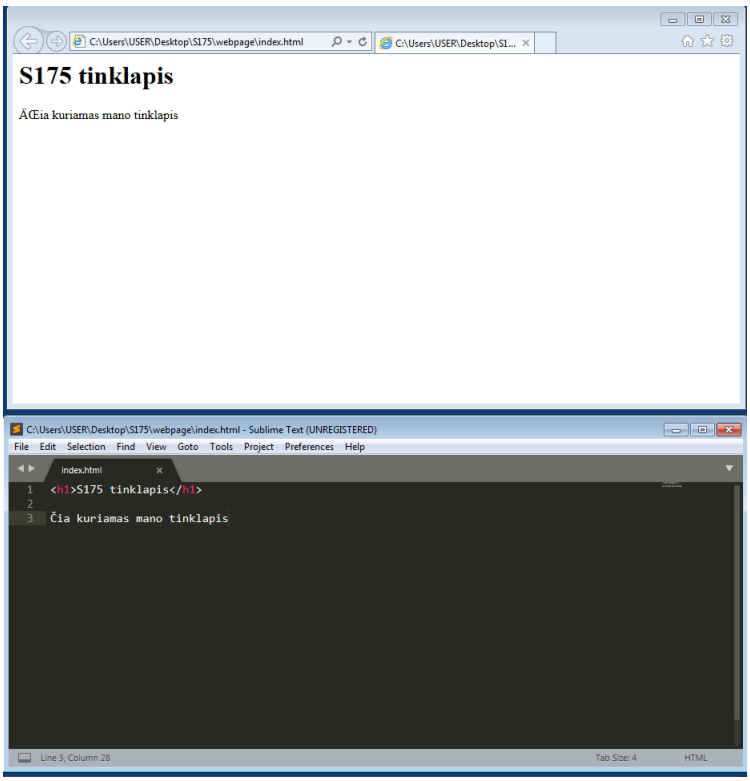
   išaugome (ctrl+s)
8. Atnaujiname interneto naršyklę (F5), rezultatas turėtų atrodyti daugmaž taip



9. Tam kad dokumentas turėtų antraštę, į pirmą eilutę parašom `<h1>S175 tinklapis <h1>`. HTML kode `<h1>` pažymį pimro lygio antraštės *tag*. Tiesa, *Sublime* yra pakankamai protingas ir pradėjus rašyti <h išmeta pasirinkimą, pasirinkus h1 pats supildo viską, bereikia įrašyti tik savo tekstą



10. Išsaugokite pakeitimus editoriuje ir atnaujinkite naršyklę
11. `git add . && git commit -m "Sukuriamas index.html failas"`



12. Dabar prikurkime šiek tiek teksto mūsų tinklapiui pvz., "Šio tinklapio kūrimas yra 1.Seminaro darbas" ir išsaugome editoriaus pakeitimus

```
<h1>S175 tinklapis</h1>Č

ia kuriamas mano tinklapisŠ

io tinklapio ūkrimas yra  1.Seminaro darbas
```

13. Pažiūrime ką rodo `git status`

```
USER@PC MINGW64 ~/Desktop/s175/webpage (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   index.html
```

14. Pažiūrime ką rodo `git diff`, kuris parodo skirtumus tarp failų

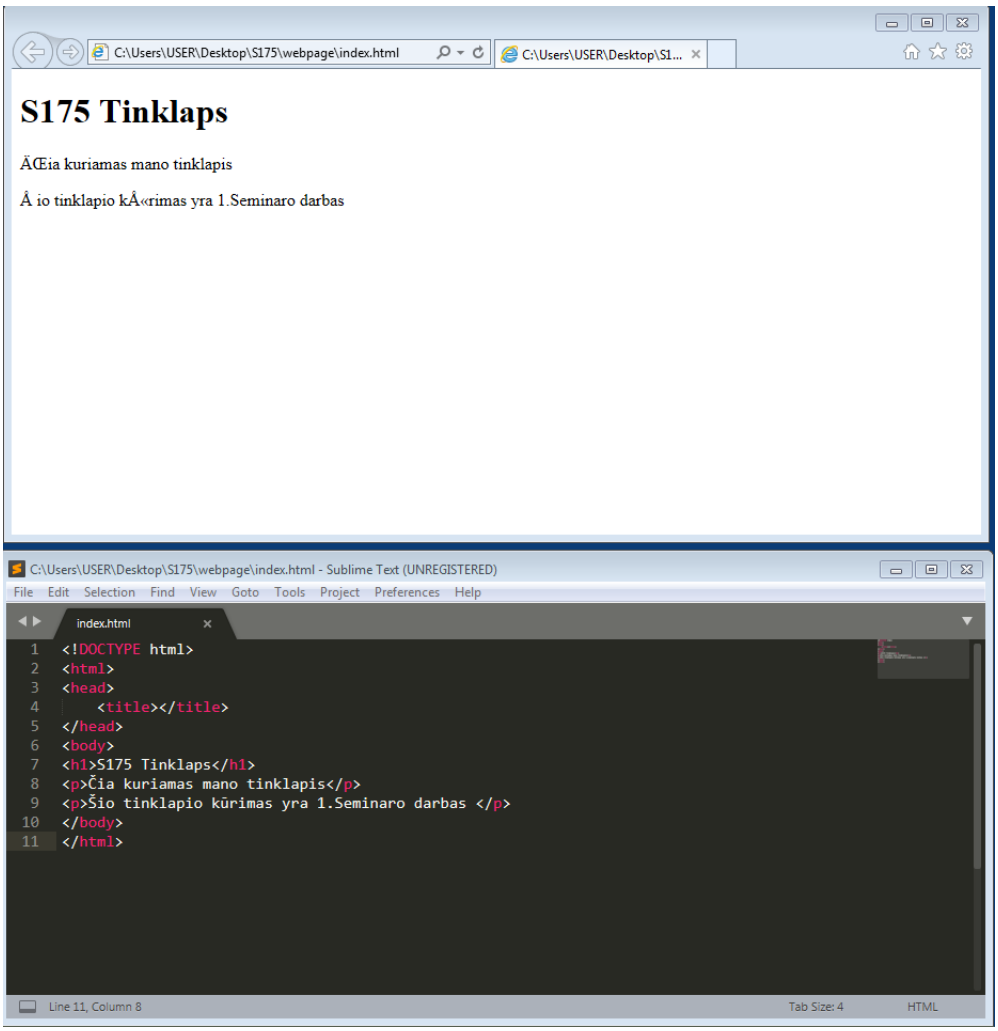
```
USER@PC MINGW64 ~/Desktop/s175/webpage (master)
$ git diff
diff --git a/index.html b/index.html
index 4d8137a..47c1b4b 100644
--- a/index.html
+++ b/index.html
@@ -1,3 +1,5 @@
 <h1>S175 tinklapis</h1>

-Čia kuriamas mano tinklapis
\ No newline at end of file
+Čia kuriamas mano tinklapis
+
+Šio tinklapio kūrimas yra  1.Seminaro darbas
\ No newline at end of file

USER@PC MINGW64 ~/Desktop/s175/webpage (master)
$
```

15. Kadangi pakeitimai teisingi, galima `git commit -am "Sukurta index.html šėantrat."` Kadangi jau trackinam visus failus tai galima naudoti sutrumpintą komandą, kur -am reiškia -all ir -m

16. Na bet į internetinį puslapį toli gražu tai dar nepanašu... Todėl į editorių perrašome šį bloką" (arba pradedame typinti <h ir pasirenkame html ir užpildome trūkstama informacija)

```
<!DOCTYPE html>
<html>
<head>
    <title>Home</title>
</head>
<body>
<h1>S175 Tinklaps</h1>
<p>Čia kuriamas mano tinklapis</p>
<p>Šio tinklapio ūkrimas yra 1.Seminaro darbas </p>
</body>
</html>
```

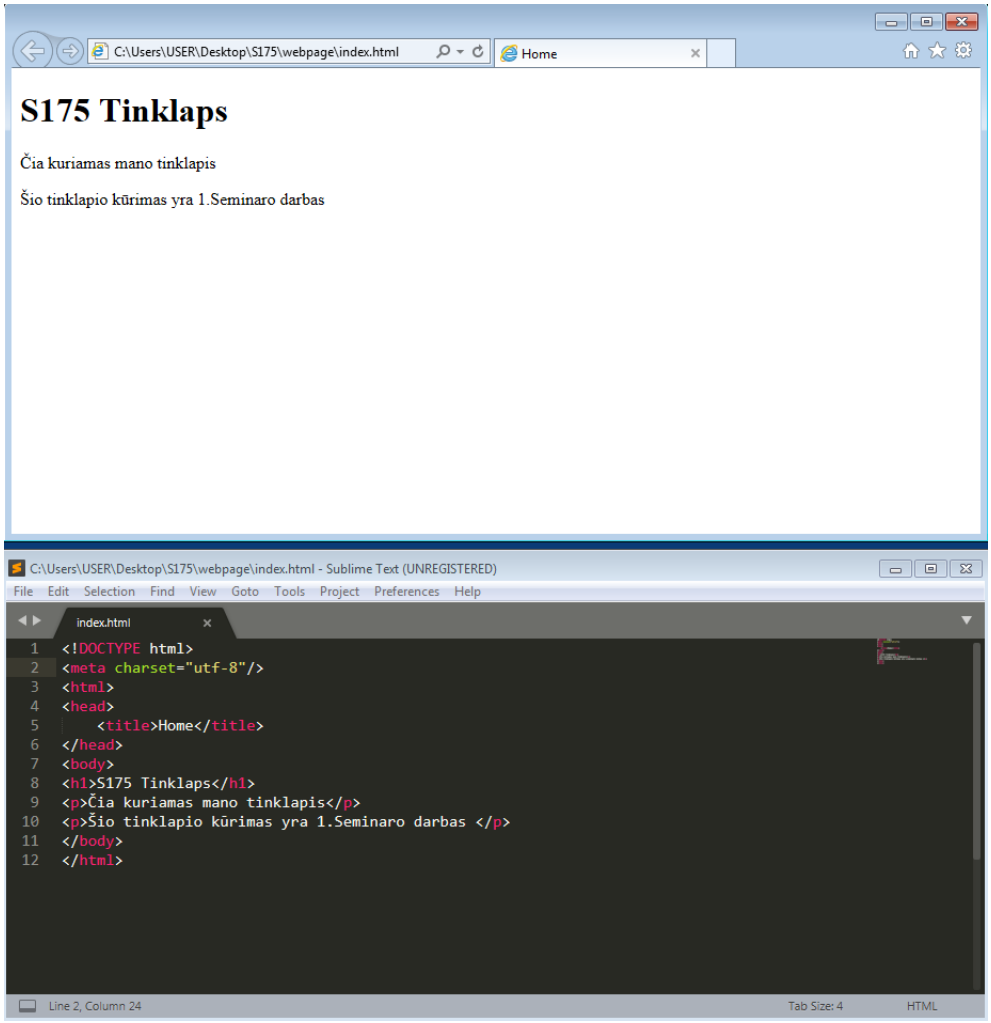
- `<!DOCTYPE html>` nusako koks tai dokumento tipas, šiuo atveju tai html dokumentas
- `<html>` atidaro html tag
- `<head>` atidaro head tag
- `<title></title>` atidaro ir uždaro title tag
- `</head>` uždaro head tag
- `<body>` atidaro body tag (t.y. puslapio turinys)
- `<h1></h1>` atidaro ir uždaro anntraštės tipo tag
- `<p></p>` atidaro ir uždaro paragrafo tag
- `</body>` uždaro body tag
- `</html>` uždaro html tag



17. Matyti, jog naršyklėje esantis Tab rodo ne visai tai, ką norėtume matyti, pvz., "Home", todėl turime editoriaus kodą 4 eilutėje pakeist į `<title>Home</title>`

18. Dar vienas nemalonus dalykas, naršyklė nesupranta lietuviškų simbolių, todėl įterpiame po pirmos eilutės: `<meta charset="utf-8"/>`, taigi priskiriame meta informaciją puslapiui ir pasakome, jog naršyklė skaitydama šį tekstą naudotų utf-8 kodavimą, kuris turi ir lietuviškus simbolius

19. Išsaugome editoriaus pakeitimus, atnaujiname naršyklę ir commitinam pakeitimus `git commit -am "papildytas i`
Dabartinis tinklapis jau daug maž panašus į tinklapį.



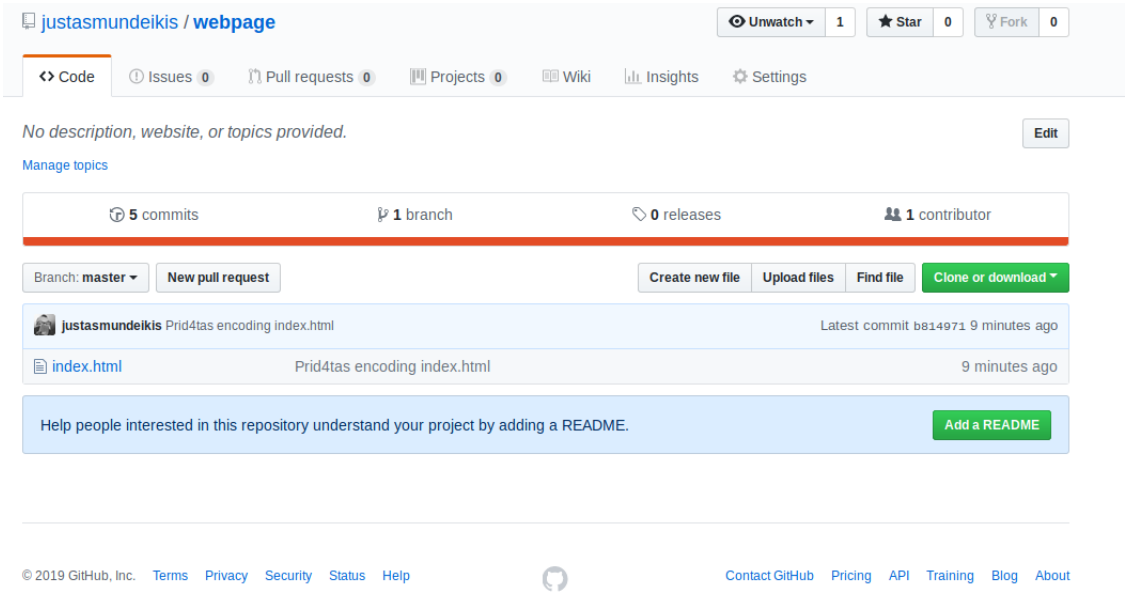
20. Github sukuriame naują repo "webpage", Praleidžiame aprašymą ir paliekame be "README".

21. Keliaujame į Git Bash ir įrašome

    `git remote add origin https://github.com/vartotojovardas/webpage.git` .

    tada shipinam savo darbą į internetą:

    `git push –u origin master"` . -u reiškia, jog vietinė repo išmoksta, kad *Github* yra *upstream repository*, o tai leis ateityje tiesiog parsisiųsti naudojant *pull* komandą. Ataujiname Github psl ir pamatysim



22. Kadangi nėra blogai turėti README failą, tai Git Bash sukuriame ji.

    `touch README.md"`.

    o tada atsidarome failą su *Sublime* ir perrašome šį Markdown formatuotą tekstą:

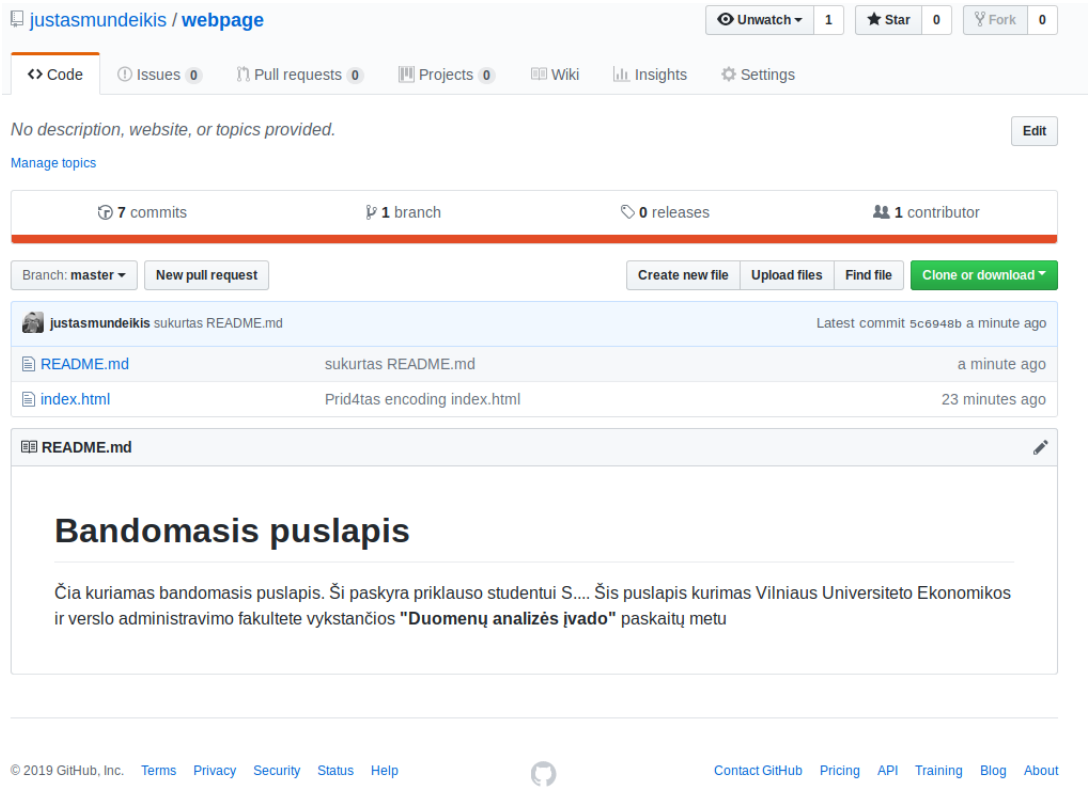
```
1 # Bandomasis puslapisČ
2
3 ia kuriamas bandomasis puslapis. Ši paskyra priklauso studentui S....Š
4 is puslapis kurimas Vilniaus Universiteto Ekonomikos ir verslo administravimo
    ↪ fakultete čvykstanios **"ųDuomen ėanalizs įvado"** ųpaskait metu
```

23. dabar reikia stag'inti ir commit'inti README.md fail, bei ship'inti jį į Github

    `git add .` && `git commit -m "sukurtas README.md"` bei

    `git push`. Kadangi jau pasakėme kas yra *upstream repository*, galime trumpinti komandą.

24. Atnaujinius GitHub puslapį, matome:



25. Čia galima rasti daugiau markdown formatavimo subtilybių: Markdown cheetsheat link

26. Kadangi puslapis yra gana nykus, pabandysim įdėti kokį linksmą paveiksliuką

27. Git Bash sukuriame naują aplanką `mkdir images`

28. Git Bash yra stiprus tuo, jog gali netgi atsiųsti failus iš interneto! Tam naudojame šią komandą:

```
curl -o images/github.png -OL https://upload.wikimedia.org/wikipedia/commons/thumb/b/b3/GitHub.svg/567px-GitHub.svg.png
```

Folderyje images turėjo atsirasti šis paveiksliukas

29. Dabar reikia dar jį įterpti į *index.html* failą. Tai galime padaryti html body įrašę tokią eilutę `<img src="path/to/file"alt="Description">`. Kadangi paveiksliukas yra *images* tai įrašome atitinkamai src="" (source). Na bet kad būtų linksmiau, pridedame dar ir vieną paveikkl-siuk iš interneto

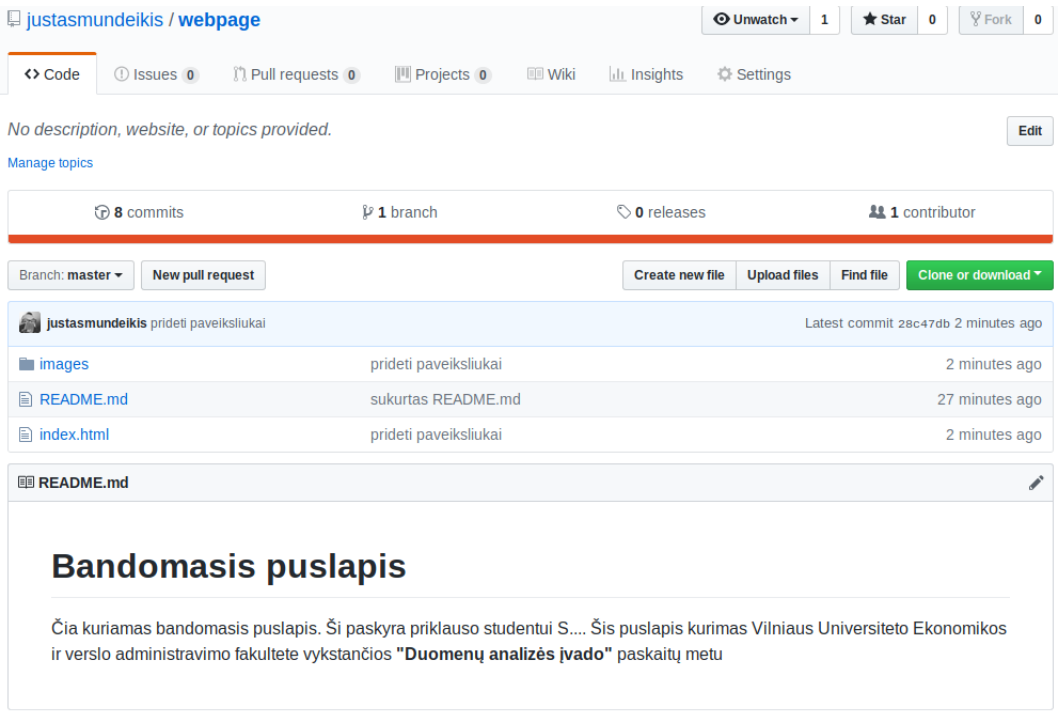
```
<!DOCTYPE html>
<meta charset="utf-8" />
<html>
<head>
    <title>Home</title>
</head>
<body>
<h1>S175 Tinklaps</h1>
<p>Čia kuriamas mano tinklapis</p>
<p>Šio tinklapio ūkrimas yra 1.Seminaro darbas </p>
<p><img src="images/github.png" alt="github paveiksliukas"></p>
<p><img src="https://studyguide.itu.dk/~/media/studyguide/student-life/facilities-at-itu/it-facilities/github/github_logo.png?h=248&w=573&la=en">
</p>
```

```
</body>
</html>
```

`git add . && git commit -m "ėpridti paveiksliukai.md"` bei

`git push`

Ataujinus Github puslapį, matome, jog atsirado folderis "images"



30. GitBash sukuriame naują atšaką ir einame į ją `git checkout -b about-puslapis` [-b sukuria atšaką]. Alternatyviai `git brancha about-puslapis` ir `git checkout about-puslapis`

31. Nukopijuojame index.html į failą pavadinimu about.html naudodami komandą `cp index.html about.html`

32. Atsidarome about.html su Sublime ir pakeičiame turinį, supildant savo duomenis

```
<!DOCTYPE html>
<meta charset="utf-8" />
<html>
<head>
    <title>About</title>
</head>
<body>
<p><strong>Vardas:</strong> Vardas ėPavard</p>
<p><strong>Stud. nr.:</strong> S175</p>
<p><strong>Kursas</strong>: Ekonomika</p>
<p><strong>ėGrup:</strong> xx</p>
</body>
</html>
```

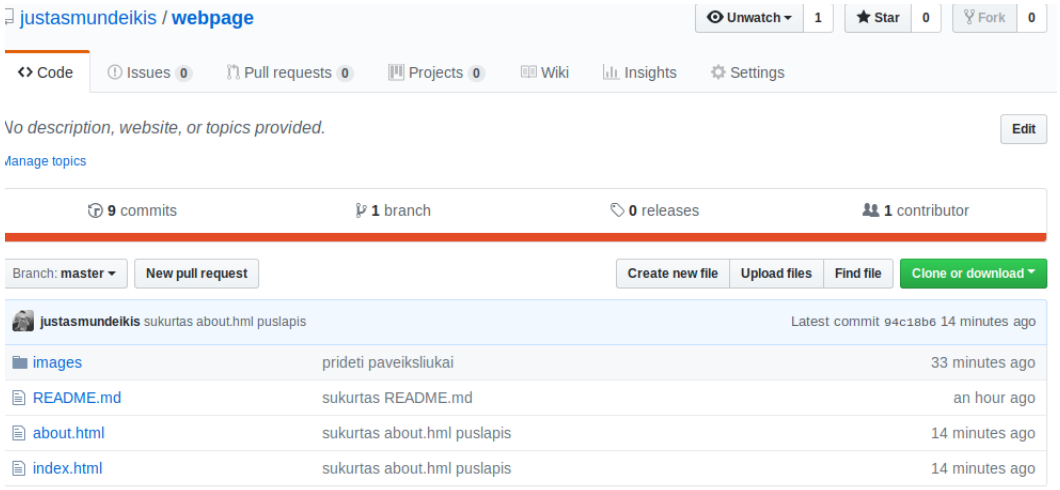
33. Kadangi norime, jog ir pagrindiniame puslapyje būtų nuorodą į puslapi apie mus, turime pakeisti index.html į

```
<!DOCTYPE html>
<meta charset="utf-8" />
<html>
<head>
    <title>Home</title>
</head>
<body>
<h1>S175 Tinklaps</h1>
<p>Čia kuriamas mano tinklapis</p>
```

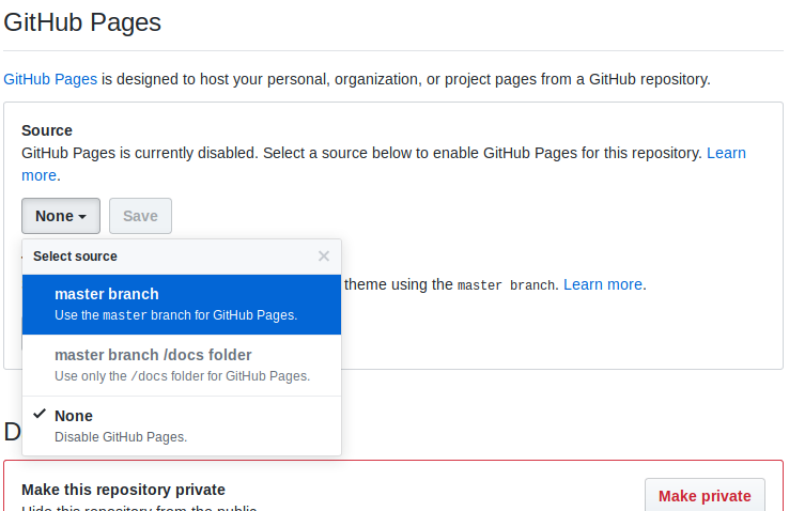
```
<p>Šio tinklapio ūkrimas yra 1.Seminaro darbas </p>
<br>
<a href="about.html">Daugiau informacijos apie mane</a>
<p><img src="images/github.png" alt="github paveiksliukas"></p>
<p><img src="https://studyguide.itu.dk/~/media/studyguide/student-life/facilities-at-itu/it-facilities/github/github_logo.png?h=248&w=573&la=en">
</p>
</body>
</html>
```

taigi įterpiame `<a href="about.html">Daugiau informacijos apie mane</a>`, tagas <a> reiškia "anchor" arba "inkaras" ir yra naudojamas nuorodoms.

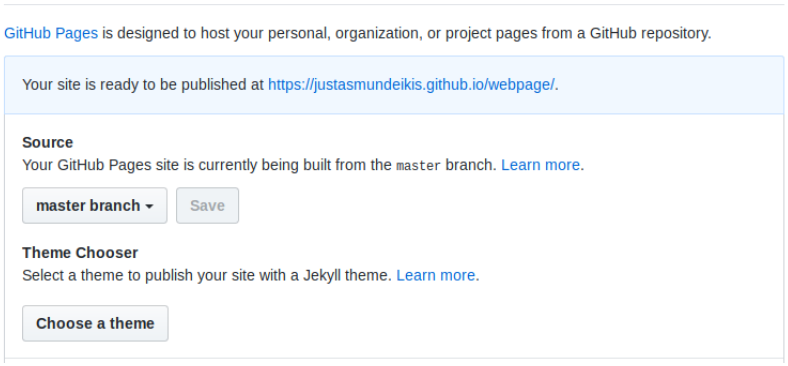
34. `git add . && git commit -m "sukurtas about.html puslapis"`
35. Tačiau esame atšakoje "about-puslpis", todėl grįžtame į "master" su `git checkout master`. Jame suprantama nėra pakeitimų, kurios padarėme atšakoje, todėl nusprendžiame sjingti atšaką su master naudodami komandą `git merge about-puslapis`.
36. Jeigu nėra kokių nors jungimo klaidų, kurias reiktų pataisyti ranka, nuostabu!
37. Jeigu nebereikia about-puslapis atšakos, kuri yra sujungta (merge), galime ją ištrinti: `git branch -d about-puslapis`.
38. Jeigu būtume sukūrę atšaką, bet joje nieko prasmingo nesukūrė,flag -d neleistų ištrinti atšakos, tada reiktų naudoti flag -D, pvz., `git branch -D about-puslapis`.
39. pasitikriname kiek atšakų turime su `git branch`.
40. Dabar galima
41. `git push`. Po sujungimo staginti ir commitinti nebereikia, nes staginimas ir commitinimas įvyko dar about-puslapis atšakoje!
42. dabar reikia nueiti į "Settings"



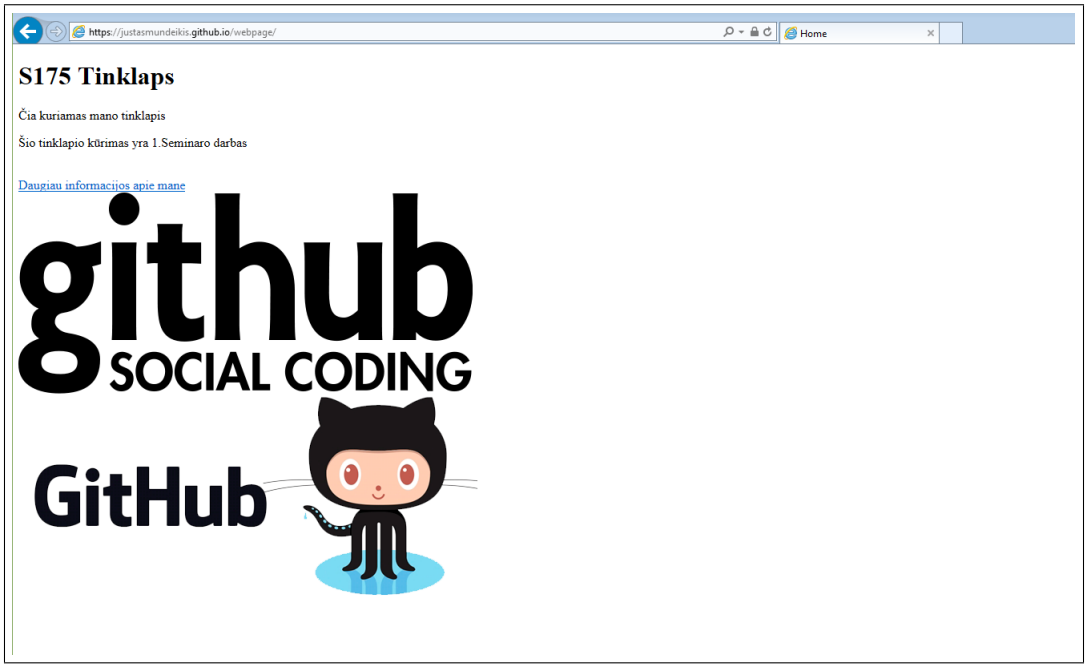
43. Ties "GitHub Pages" iš "None" pasikeisti į "master branch"



44. Paspaudus nuorodą:



45. Atsidaro pilnai veikiantis tinklapis! Sveikinu!

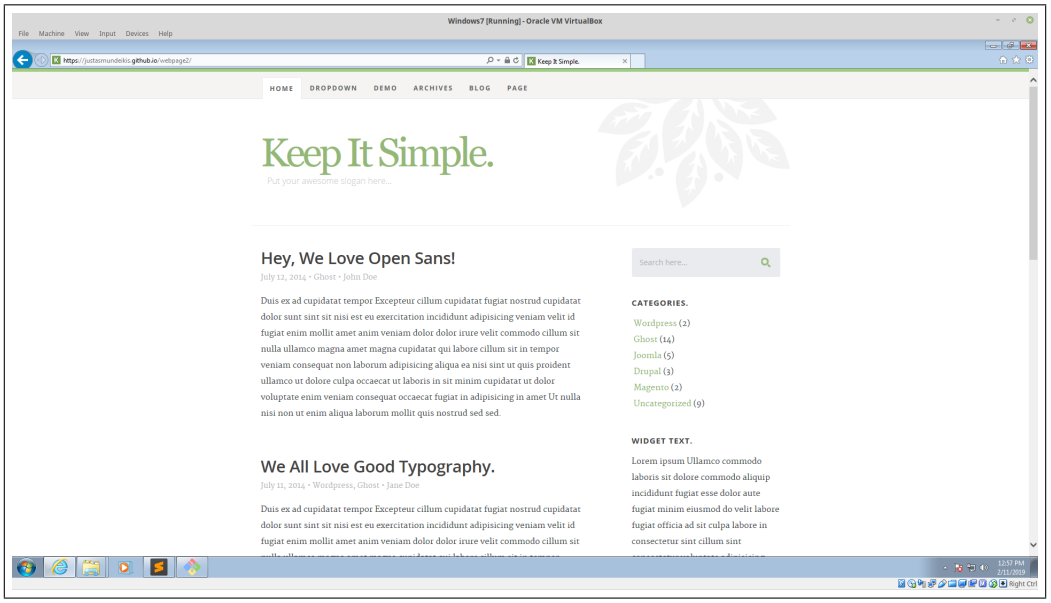


46. Jeigu norite sukurti labiau sofistikuotą puslapį, galima naudotis html editoriaus pagalba, pvz., https://html-online.com/editor/

4 Kaip sukruti nuosavą homepage per 5 min

- Github paskyroje sukuriame naują *repo* pvz., "webpage2"
- Parsisiunčiam iš tinklapio https://designseer.com/free-blog-html-css-templates/ "KeepSimple" tinklapio paketą išsaugant jį S175 folderyje
- Unzipiniam atsisiųstą failų paketą su GitBash `unzip KeepItSimple20.zip`
- Pereinam į išpakuotą folderį, inicializuojame git, pridedame visus failus, ir commitiname naudojant sujungtą komandą:

  `cd KeepSimple20 && git init && git add . && git commit -m "sukelti failai"`
- sujungiame su Github `git remote add origin https://github.com/<username>/webpage2.git"`
- ir pushinam į Github `git push -u origin master"`
- "Settings" esančioje "GitHub pages" dalyje, "Source" pakeičiame į "master"
- ir nuspaudus ant nuorodos turime veikiantį tinklapį



- tiesa, jis kiek sudėtingesnis, turi Java Script, CSS ir t.t., bet pati struktūra yra aiški
- Kuo paprastesnį Template susirasite, tuo papračiau galima turėti savo tinklapį ir jį keisti
- Taip pat puslapį galima susikurti ir tiesiog pačiame Github