

KLIJENTSKE TEHNOLOGIJE

Vežba — Uvod u GitHub i postavljanje portfolio projekta (GitHub Pages)

Git je sistem za verzionisanje koji pamti svaku promenu fajlova.

GitHub je platforma koja hostuje Git repozitorijume (timovi, kolaboracija, issue-i, PR-ovi).

Zašto nam treba?

- istorija izmena (moguć povratak na staru verziju),
- saradnja i pregled koda,
- besplatan hosting statičkih sajtova (GitHub Pages) — idealno za portfolio.

Pojmovi:

- **Repository (repozitorijum):** “folder” projekta na GitHub-u.
- **Commit:** paket promjena sa porukom “šta sam uradio i zašto”.
- **Branch:** paralelna linija razvoja (npr. *main* je stabilna; *dev* za eksperimente).
- **README.md:** uvodna stranica repoa (opis, ciljevi, uputstva).
- **LICENSE:** dozvola korišćenja tvog koda (npr. MIT).
- **.gitignore:** šta Git treba da ignoriše (npr. privremene fajlove).
- **GitHub Pages:** objavljivanje statičkog sajta direktno iz repoa.

Koja je razlika između “snimanja fajla” i **commita**?

Snimanje fajla čuva promene samo lokalno, dok commit trajno beleži promenu u istoriji projekta sa opisom šta je urađeno.

Zašto je commit poruka važna?

Pomaže da kasnije razumemo zašto je promena napravljena i olakšava praćenje razvoja projekta.


Kreiranje GitHub naloga

1. Otvoriti github.com → **Sign up**.
2. Izabrati **korisničko ime** (npr. ime-prezime-portfolio).
3. Uneti e-mail, lozinku, verifikovati e-mail (klik na link).
4. Po prijavi, otvorite avatar (gore desno) → **Your profile** (videćete praznu listu repozitorijuma).

Kreiranje prvog repozitorijuma (projekta)

1. Klik gore desno + → **New repository**.
2. **Name:** my-portfolio (još bolje ime-prezime-portfolio ili brojindeksa-portfolio)
3. **Description:** “*Personal portfolio website: weekly project for master course*”.
4. **Public** (da bi Pages radio bez prepreka).
5. Uključiti **Add a README file**.
6. (Opcionalno) Dodati **.gitignore** → *None* (za sada, dodaćemo kasnije).
7. (Opcionalno) **License** → *MIT License* (otvoren, jednostavan).
8. Klik **Create repository**.

Uređivanje README.md

1. U repo-u klik na **README.md** →  **Edit**.
2. Dodaćemo neki minimalni sadržaj, a zatim primeni te izmene.
3. Dole: **Commit changes**
 - **Commit message:** *docs: add course context and week 1 goals*
 - (Ostavi “Commit directly to the main branch”) → **Commit changes**.

Konvencija prefiksa u porukama (*feat, fix, docs, style, refactor, chore, test*) pomaže u pretrazi i čitljivosti istorije.

Dodavanje početnog sajta (index.html)

1. **Add file** → **Create new file**
2. File name: index.html
3. Dodaćemo neki osnovni html kod ovde, a zatim primeniti te izmene.
4. **Commit message:** *feat: add minimal portfolio page structure (HTML skeleton)* → **Commit changes**.

Dodavanje stila (style.css)

1. **Add file** → **Create new file**
2. File name: style.css
3. Dodaćemo neki osnovni stil, malo boja, fontova, slika i margina, a zatim primeniti te izmene.
4. **Commit message:** *style: add base typography and layout for sections* → **Commit changes**.

Aktiviranje GitHub Pages

1. U repo-u klik **Settings** → levo **Pages**.
2. **Branch:** izabрати main i / (root) → **Save**.
3. Nakon ~0–3 min pojavi se **Deployment** i URL u formatu:
https://KORISNICKOIME.github.io/my-portfolio/

Provera: Otvorite URL u novom tabu. Ako ne radi, osvežite nakon 1–2 min. Ako i dalje ne radi: proverite da li repo **nije prazan** i da li je izabrana prava grana (main).

Dodavanje licence i .gitignore

- Na glavnoj strani repoa: **Add file** → **Create new file**
- **LICENSE** → upiši MIT licencu (potraži “MIT License template”, ili iz GitHub-ovog “Add a license template” interfejsa).
- **Commit message:** *chore: add MIT license*
- (Opcionalno) **Add file** → **Create new file** → **.gitignore**
U polje unesi tipičan sadržaj, na primer:

```
.DS_Store  node_modules/  *.log  .env
```
- **Commit message:** *chore: add .gitignore to exclude temporary and build files*
- **.gitignore** fajl nije obavezan za jednostavne projekte, ali je koristan za kasnije verzije.

Šta znači MIT licenca?

MIT licenca znači da svako može slobodno da koristi, menja i deli tvoj kod, ali mora da navede tvoje autorstvo. Jednostavna je i pogodna za studente jer ne ograničava upotrebu projekta, čak ni u komercijalne svrhe, dok istovremeno štiti autora od odgovornosti.

Prva planirana izmena (drugi commit)

Cilj: pokazati ritam rada, uraditi malu izmenu → commit → provera na Pages linku.

Primer izmene:

- U index.html promeniti neki tekst ili sliku.
- Dodaj još jednu stavku u **Projects** → “Week 2: Add navigation and about card”.
- **Commit message:** *content: personalize header and extend projects list*
- Otvori URL sajta i osveži (deployment je automatski).

Main task:

- Dopršiti “About me” (3–5 rečenica), dodati profilnu sliku (sliku upload-ovati u repo → images/ folder; u HTML dodati).
- Ažurirati README: dodati “**How to run / How to contribute**” kratko uputstvo (npr. “open index.html in a browser”).
- Podeliti **GitHub Pages URL** na MS Teams-u kroz komentar za pinovanu objavu.

Side task:

- Napraviti **branch dev**, uraditi malu promenu na dev, pa **open Pull Request** ka main i **merge**.
- Dodati **favicon** (<link rel="icon" href="images/favicon.ico">).
- Kreirati projects.html i linkovati ga iz navigacije (Pages i dalje radi; prikazuje svaki HTML iz root-a).

Najčešće greške i kako ih rešiti

- **Pages 404 / ne radi:**
 - Proverite Settings → Pages → Branch = main, folder / (root), repo nije prazan.
 - Sačekajte 1–3 min i osveži.
- **CSS se ne vidi:**
 - Putanja <link rel="stylesheet" href="style.css"> mora biti tačna; fajl u root-u.
- **Specijalni karakteri ne rade:**
 - Dodajte <meta charset="UTF-8"> (već je u šablonu).
- **Slika ne prikazuje:**
 - Putanja (npr. images/me.jpg) i ekstenzija (mala/velika slova) moraju tačno da se poklapaju.
- **Commit bez poruke:**
 - Uvek napišite kratko **šta i zašto** (pomaže i vama i recenzentu).

Zašto je GitHub Pages odličan za portfolio:

- **Besplatan, brz i jednostavan** hosting statičkih sajtova (HTML/CSS/JS).
- Svaka izmena kroz **commit** → “živ” portfolio sa istorijom rasta.
- **Profesionalno:** poslodavci mogu da vide i sajt i izvorni kod, plus commit istoriju.
- **Efikasno:** lako dodate blog, projekat, interaktivne JS komponente, analitiku, formu.

Rad lokalno u VS Code uz Git i povezivanje sa GitHub-om

Ako želite da uređujete fajlove na svom računaru (a ne direktno na GitHub sajtu), možete raditi **lokalno u VS Code** i kasnije **poslati izmene (push)** na GitHub..

◆ 1. Instalacije

Pre svega, potrebno je da instalirate dva alata:

- **Git** – preuzmite sa <https://git-scm.com>
→ tokom instalacije samo klikajte *Next* i ostavite podrazumevana podešavanja.
→ nakon instalacije možete proveriti da li radi otvaranjem **Command Prompt** jednostavnim unosom: `git -version`
- Ako se pojavi broj verzije, npr. `git version 2.44.0`, sve je u redu
- **VS Code (Visual Studio Code)** – preuzmite sa <https://code.visualstudio.com>
→ otvorite i instalirajte ekstenziju **GitHub Pull Requests and Issues**

◆ 2. Kloniranje (preuzimanje) repozitorijuma

1. Otvorite stranicu svog projekta na GitHub-u (npr. my-portfolio).
2. Kliknite zeleno dugme **Code** → **HTTPS** → **Copy link**
(link izgleda npr. `https://github.com/ime/my-portfolio.git`).
3. U VS Code otvorite **Terminal** → **New Terminal** i upišite:
`git clone https://github.com/ime/my-portfolio.git`
4. Nakon nekoliko sekundi, videćete novi folder sa istim imenom (my-portfolio) u kojem se nalaze svi vaši GitHub fajlovi.

◆ 3. Rad lokalno

Sada otvorite projekat u VS Code-u:

- Kliknite **File** → **Open Folder** → **my-portfolio**
- Uređujte fajlove (npr. `index.html`, `style.css`, dodajte slike, nove stranice itd.).
- VS Code automatski prikazuje koje ste fajlove menjali (plavi, crveni ili zeleni markeri u sidebaru).

Sve što menjate ovde je lokalno – dok ne napravite commit, Git ne beleži promenu.

◆ 4. Commit i slanje (push) na GitHub


Kada završite sa izmenama:

1. Otvorite terminal i unesite sledeće komande redom:

```
git add .  
git commit -m "feat: update about section"  
git push origin main
```

- `git add .` → govori Git-u koje fajlove da pripremi za snimanje (tačka znači sve fajlove).
- `git commit -m "poruka"` → pravi zapis o promenama (obavezno napišite kratku i jasnu poruku).
- `git push origin main` → šalje promene nazad na GitHub.

2. Osvežite stranicu svog repozitorijuma, videćete novi commit i ažurirane fajlove.

 *Commit je kao "save" sa opisom, a push je kao "upload" na GitHub.*


◆ 5. GitHub Pages – automatska objava

GitHub Pages će automatski prepoznati promene i obnoviti vaš sajt u roku od 1–2 minuta. Samo osvežite link: <https://korisnickoime.github.io/my-portfolio/>

 *Ako se stranica ne osveži odmah, sačekajte kratko i osvežite browser (Ctrl + F5).*

◆ 6. HTTPS vs SSH (kratko objašnjenje)

- **HTTPS** je jednostavniji za početak, koristi se link sa `https://` i pri svakom push-u GitHub traži prijavu (korisničko ime i token).
- **SSH** je pogodniji za češći rad, jednom se dodaje ključ (key) i više ne unosite lozinku pri svakom slanju. Preporuka je da se SSH koristi tek kada student savlada ove tehnike i redovno commit-uje i push-uje promene.

 *SSH i HTTPS rade isti posao, omogućuju slanje koda na GitHub, ali SSH je praktičniji ako često ažurirate projekat, jer vam pruža brži i sigurniji pristup bez ponovnog unosa lozinke pri svakom push-u.*