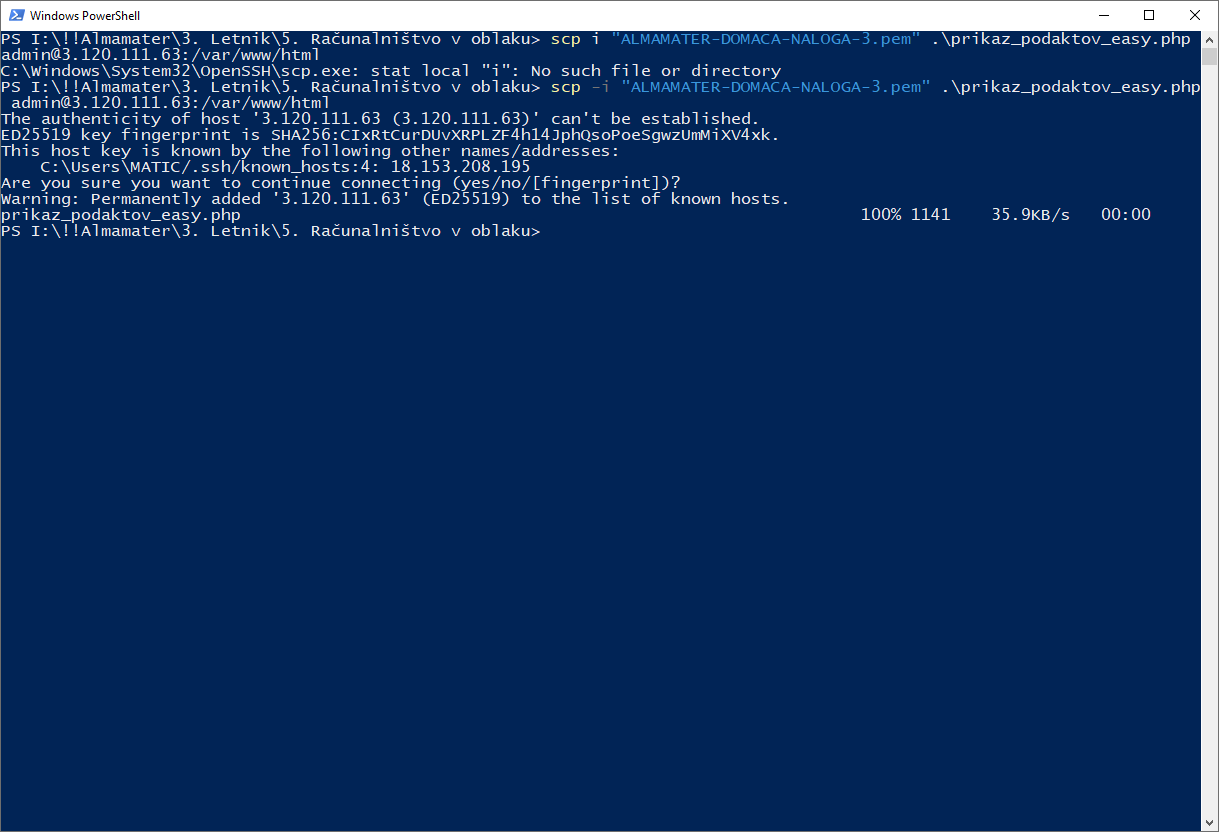
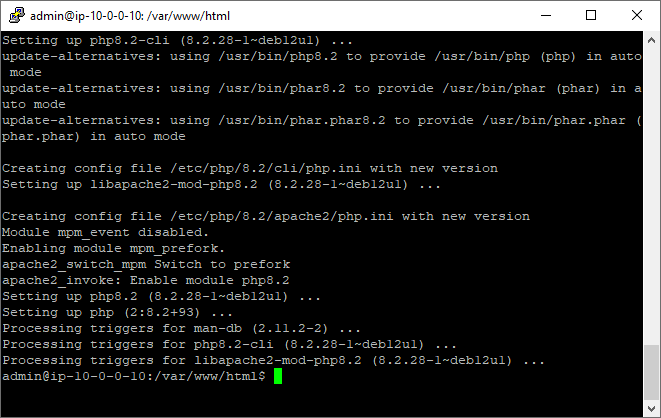
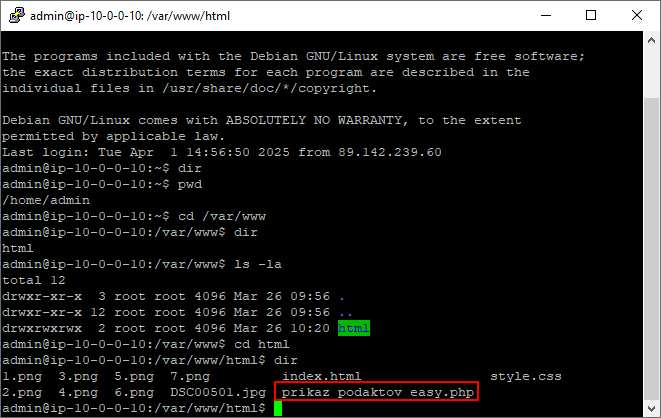
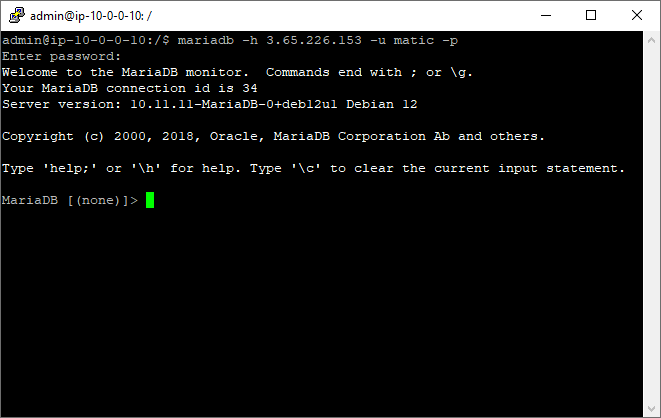
Najprej pogledamo ali ima mapa html pravice za urejanje. Najprej se postavimo znotraj mape www z ukazom cd /var/www. Za pregled pravic napišemo ls -la.

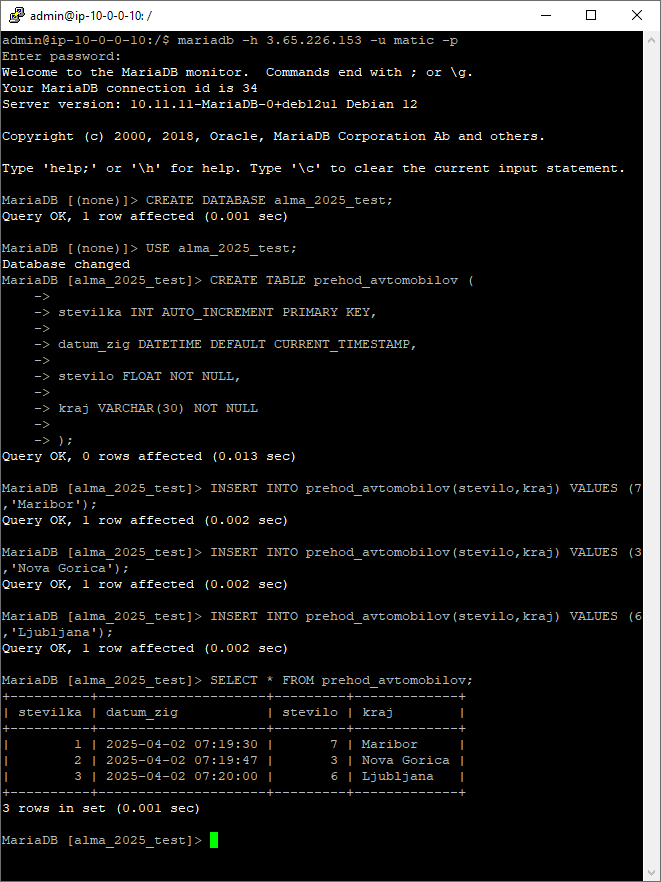
Nato smo z ukazom scp i “naziv ključa.pem” .želena datoteka admin@IPračunalnika:/var/www/html prekopiramo želeno datoteko. V tem primeru smo kopirali php datoteko, katera bo izpisala podatke iz SQL baze.

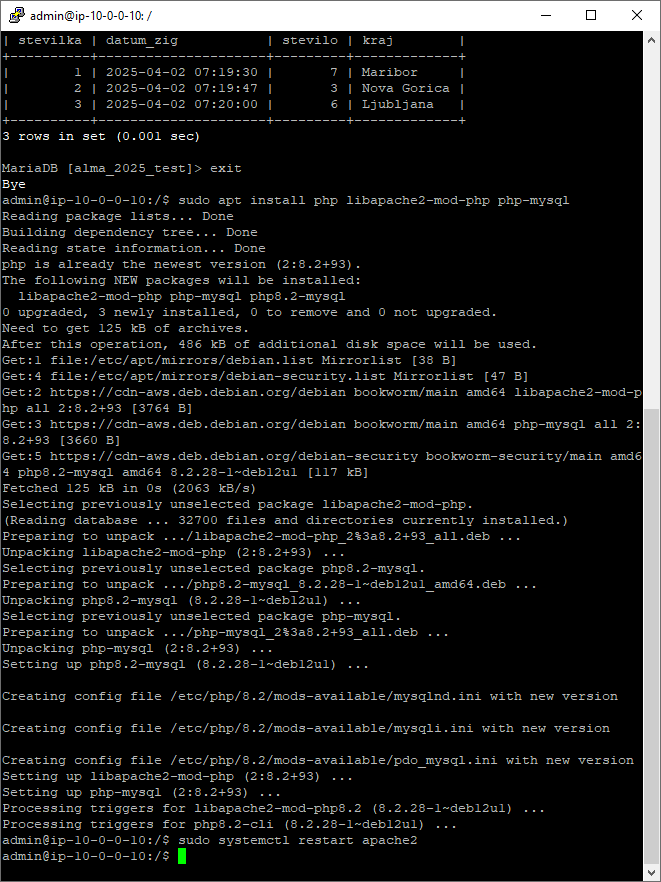
Na virtualnem računalniku se postavimo znotraj html datoteke in z ukazom dir pokaže vse datoteke na strežniku.

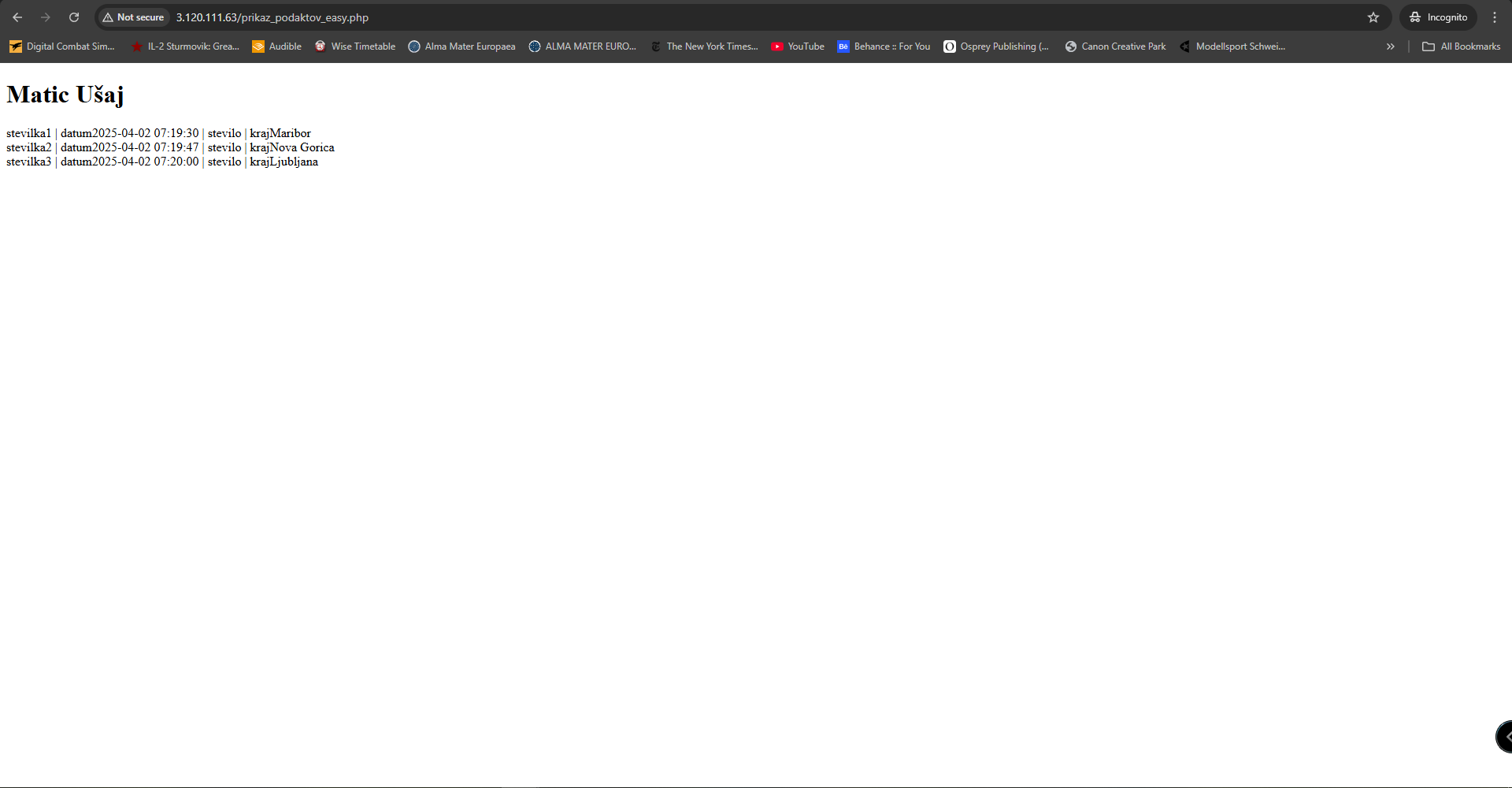
Nato z ukazom sudo aptinstall apache2 namestimo apache.

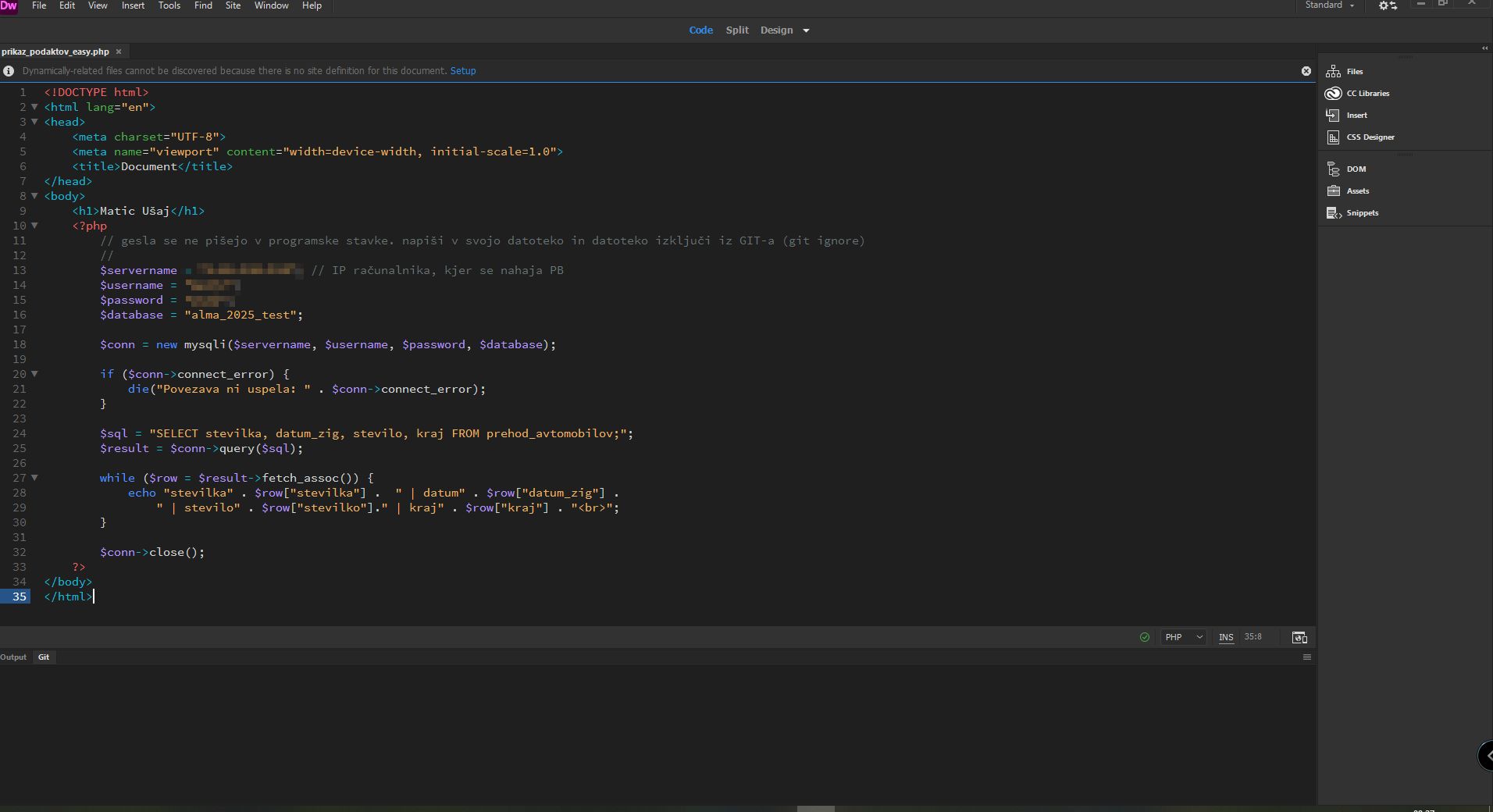
Po namestitvi apache se prijavimo v podatkovno bazo z ukazom mariadb -h [IP] -u matic -p

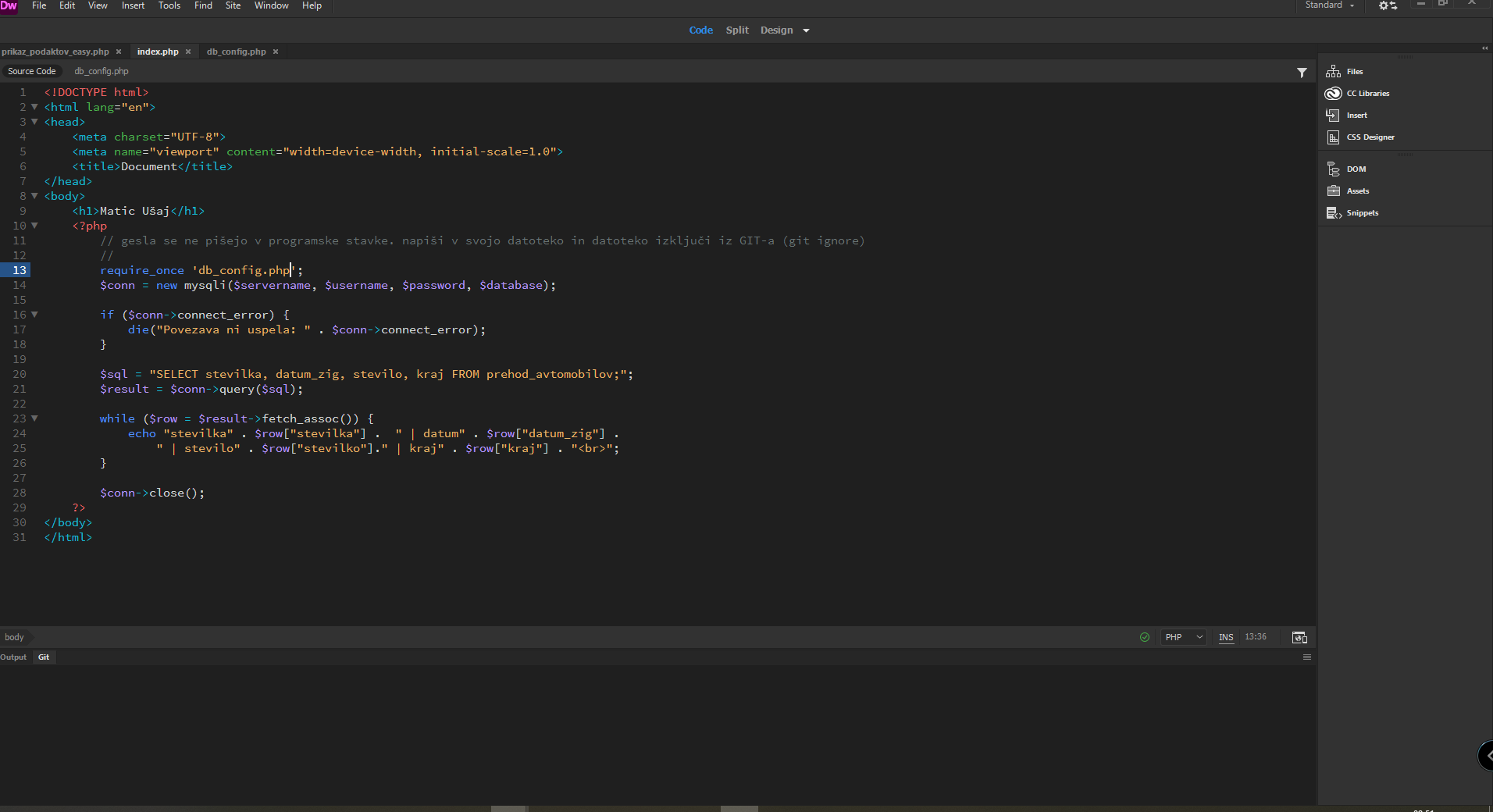
Nato kreiramo podatkovno bazo in jo napolnimo z podatki.

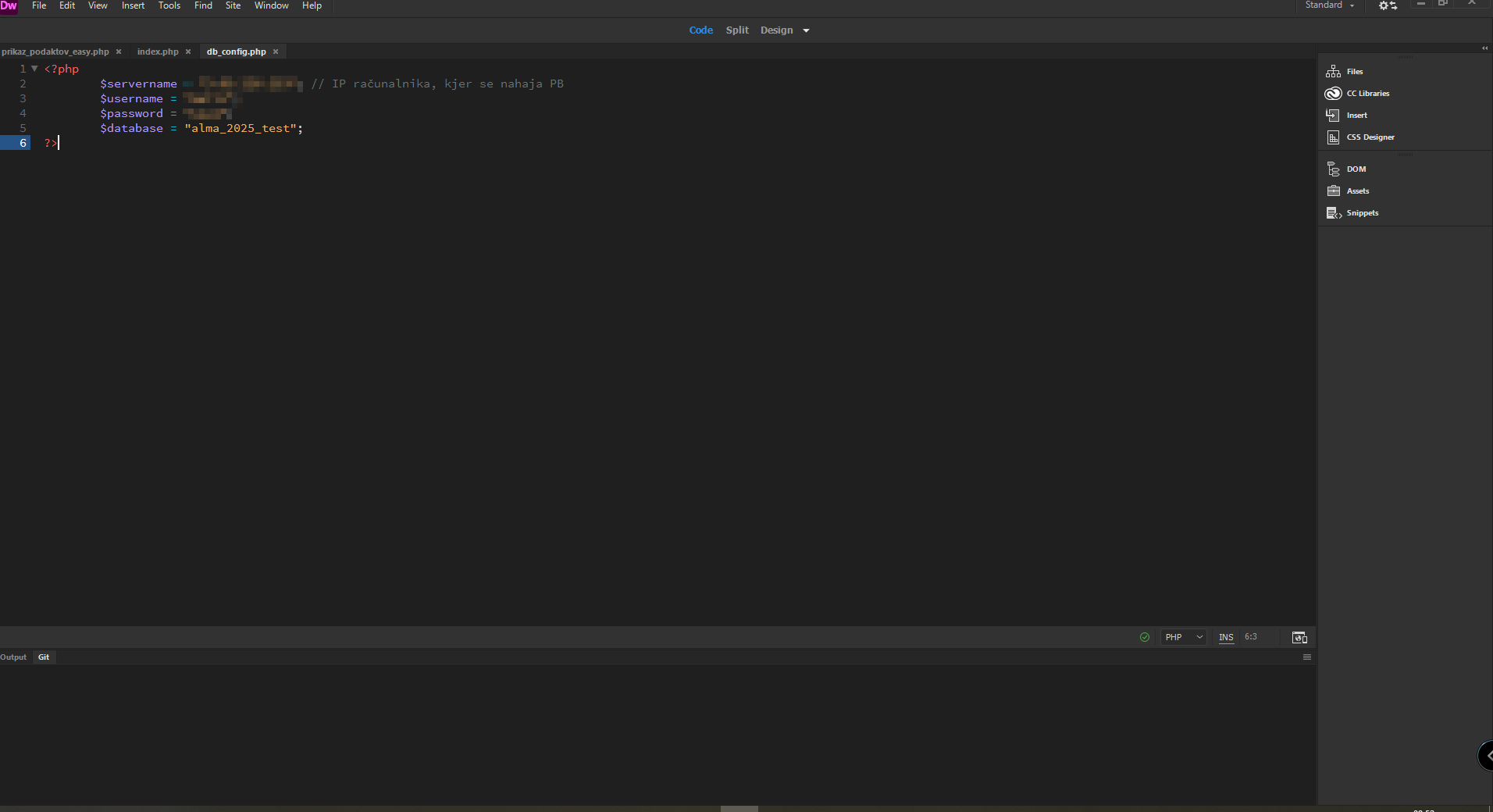


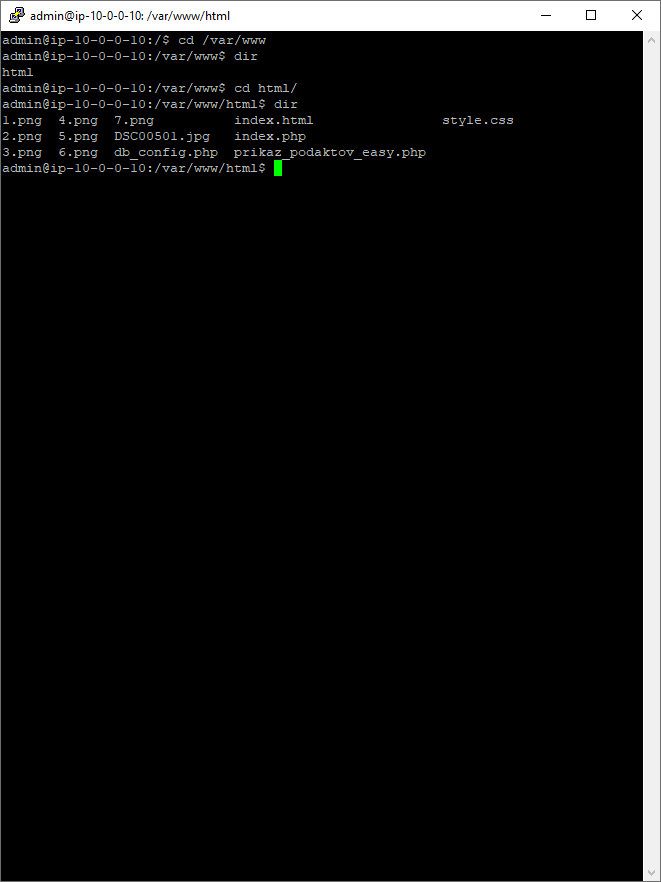
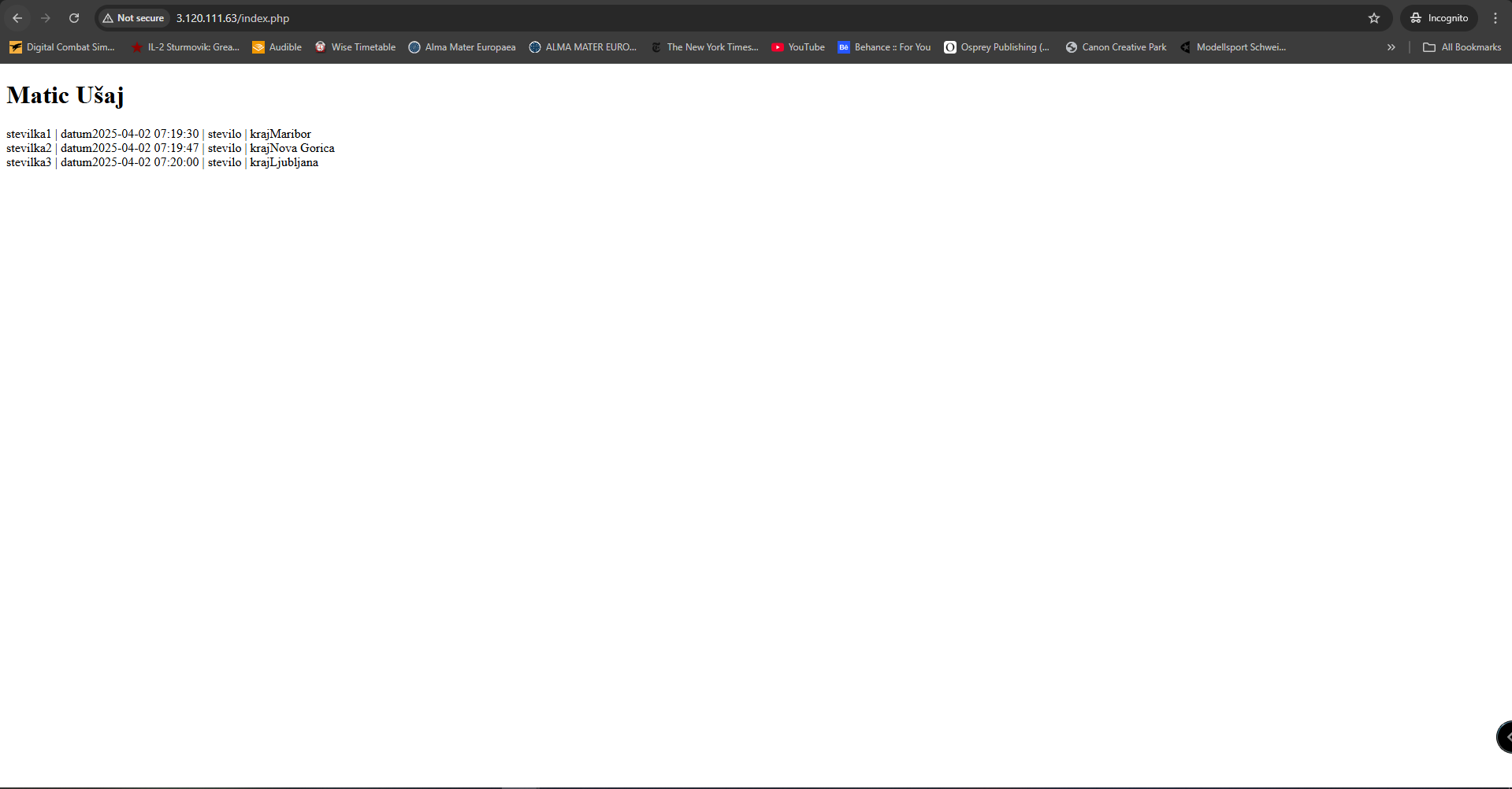
Po kreiranju baze sem moral namestiti dodatek za apache. Sudo apt install php libapache2-mod-php php mysql in resetirati apache z ukazom sudo systemctl restart apache2

Na spletni strani so se prikazali podatki iz podatkovne baze.

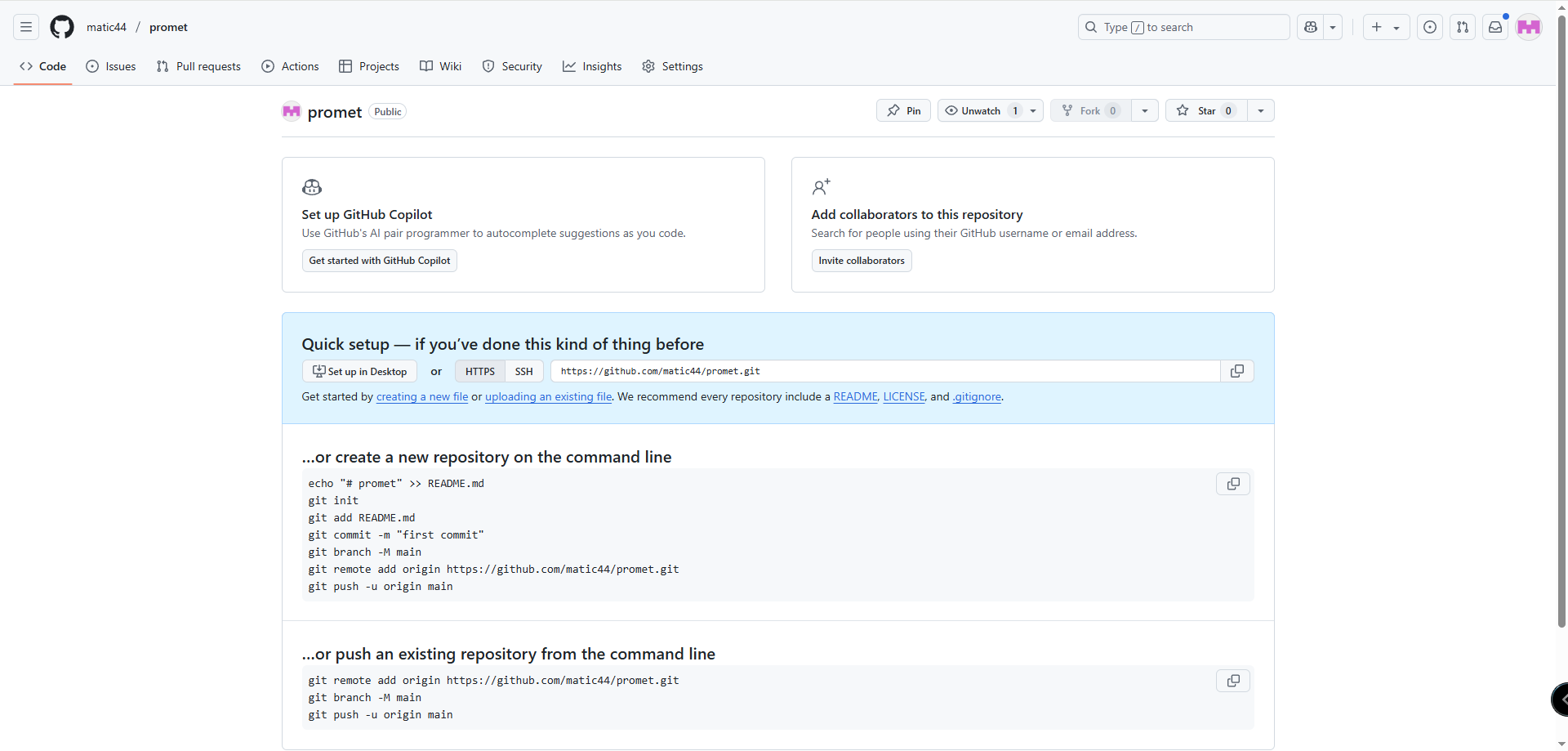
Datoteke prikaz\_podatkov\_easy.php

V naslednjem delu smo preoblikovali php datoteko, tako da smo kreirali novo php datoteko (index.php). Iz nje smo izrezali del kode kateri je namenjen z ukazom require\_once ‘db\_config.php’; Izrezani del smo in prilepili v datoteko db\_config.php.

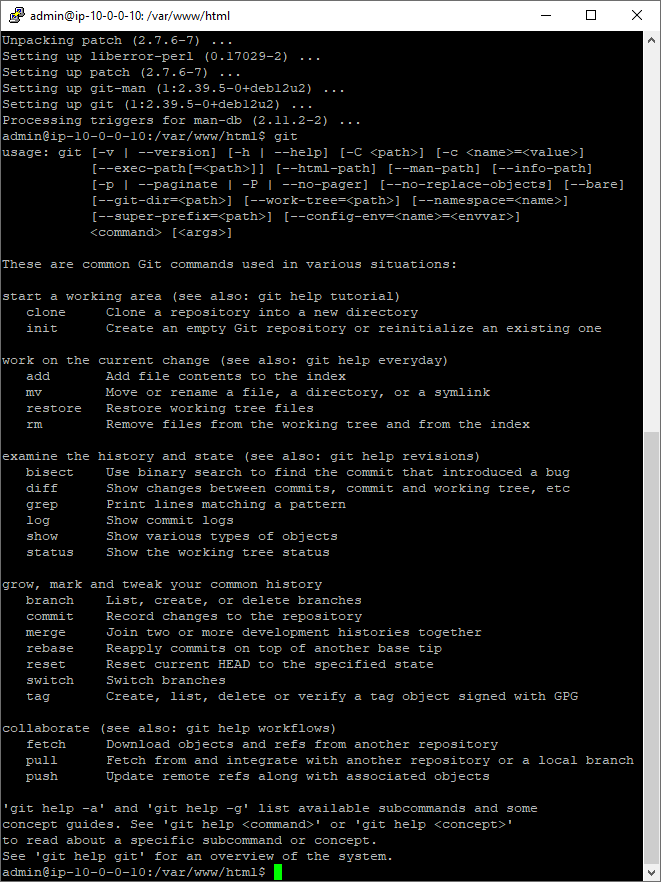


Obe datoteki prenesemo na strežnik in ponovno testiramo, če se v brskalniku pokažejo podatki iz podatkovne baze.

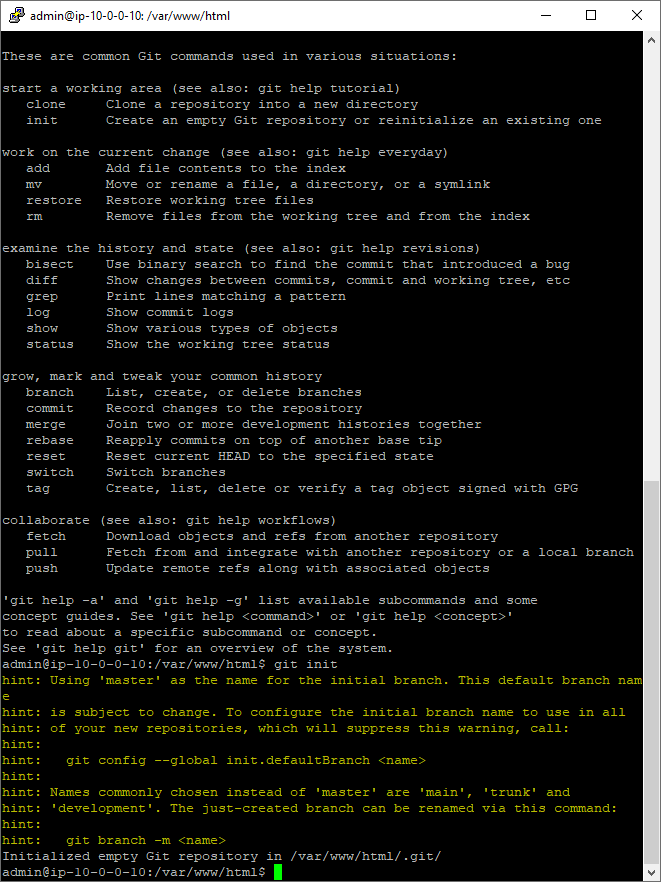
Nato smo kreirali nov repozatori, v katerega bomo avtomatično shranili vse datoteke iz virtualnega računalnika.



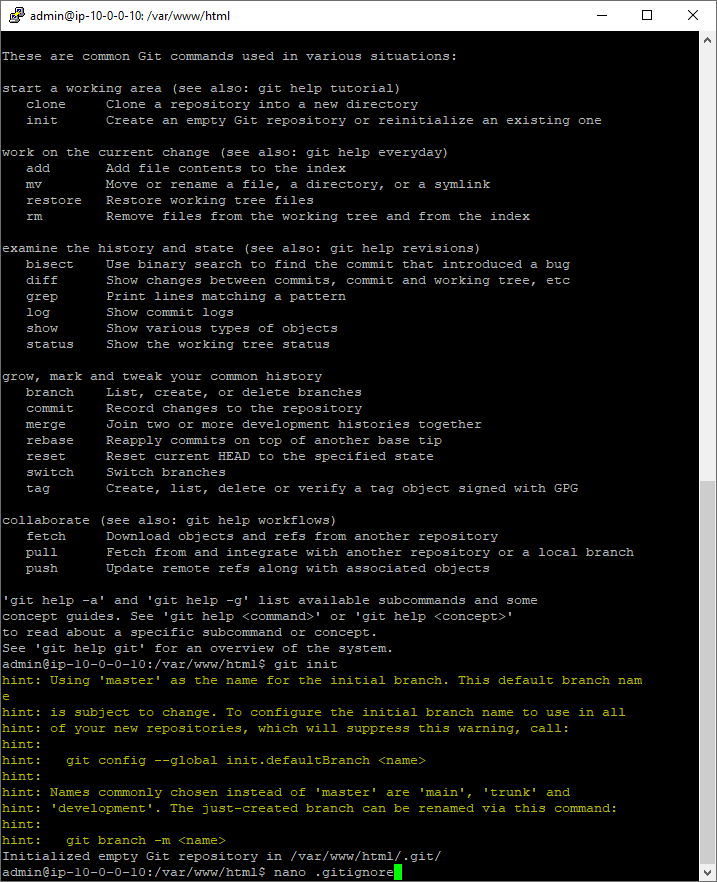
Na virtualni računalnik namestimo Git (sudo apt install git) in preverimo ali je nameščen z ukazom git.



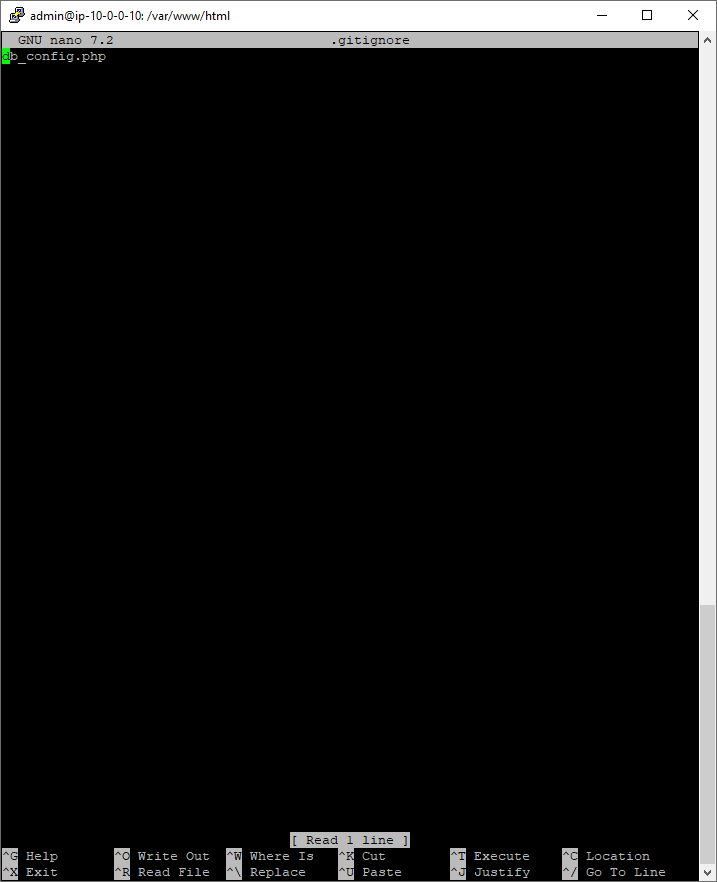
Nato z ukazom git init inicializiramo git.



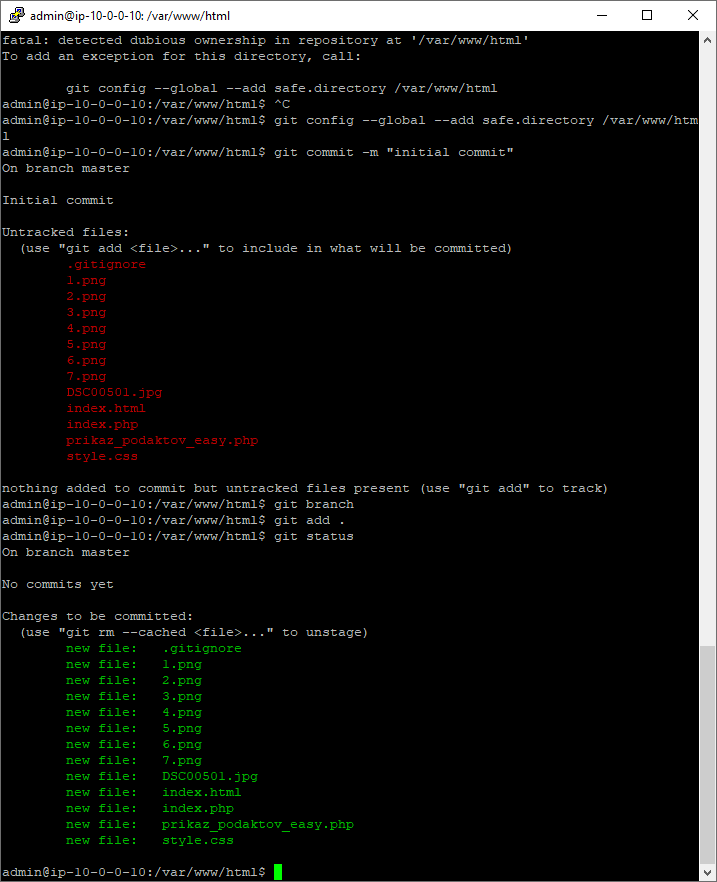
Nato kreiramo datoteko, v katero bomo napisali katere programe naj git spregleda. To naredimo z ukazom nano .gitignore. Nano kreira novo datoteko.

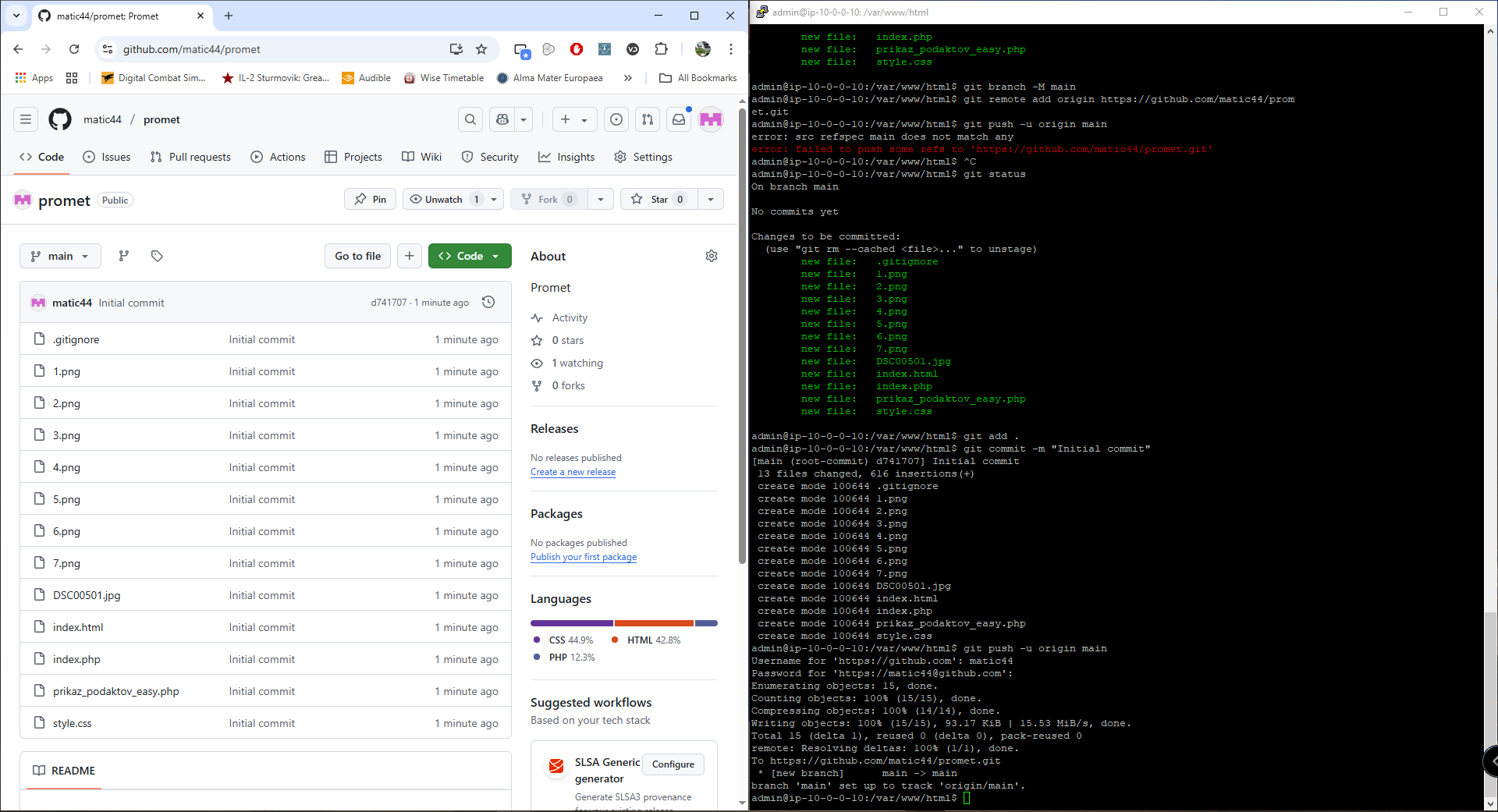


Ignoriral bo datoteko db\_config.php.

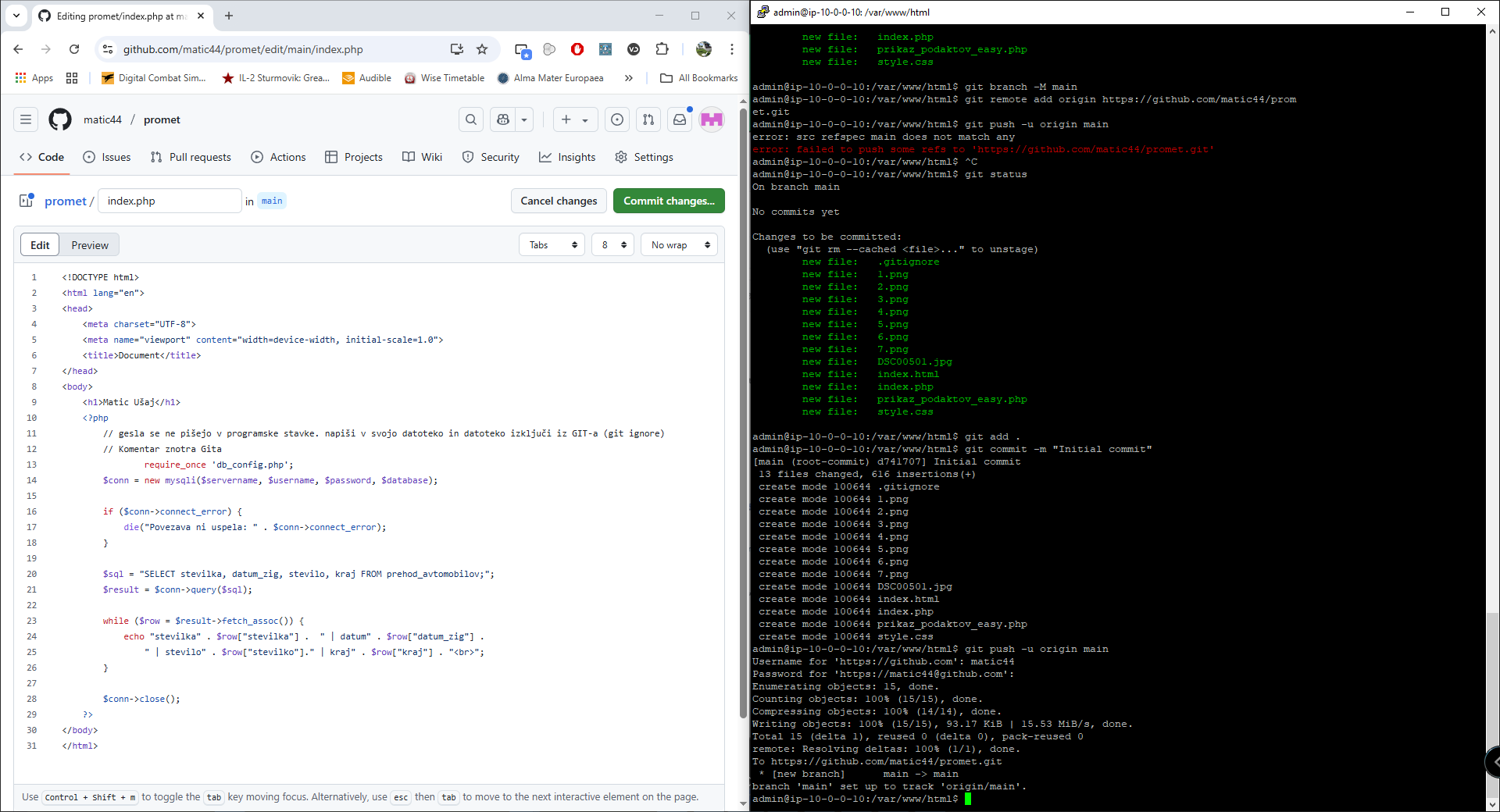


Nato poskusimo objaviti vse datoteke v git.

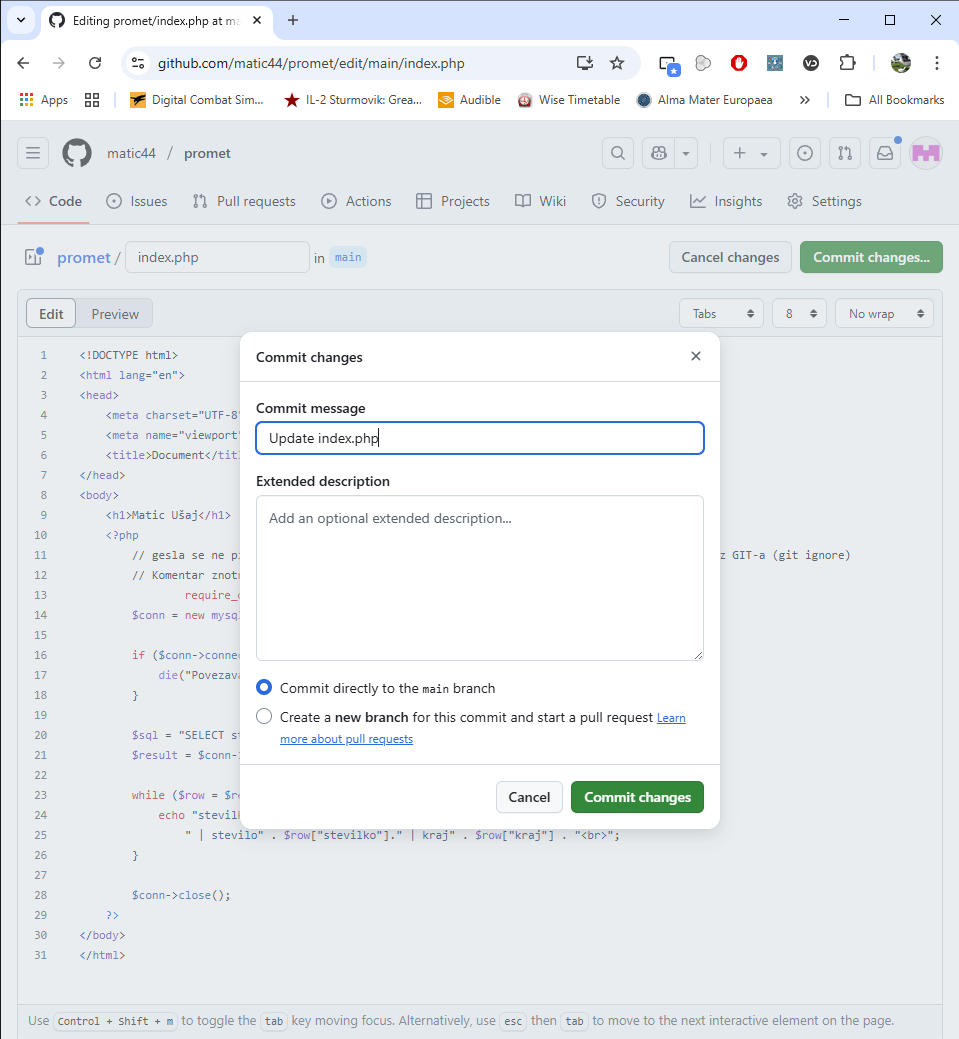
1. Git config –global –add safe.directory /var/www/html
2. Git commit -m “initial commit”
3. Git branch
4. Git add .
5. Git status

Uspešen prenos datotek.

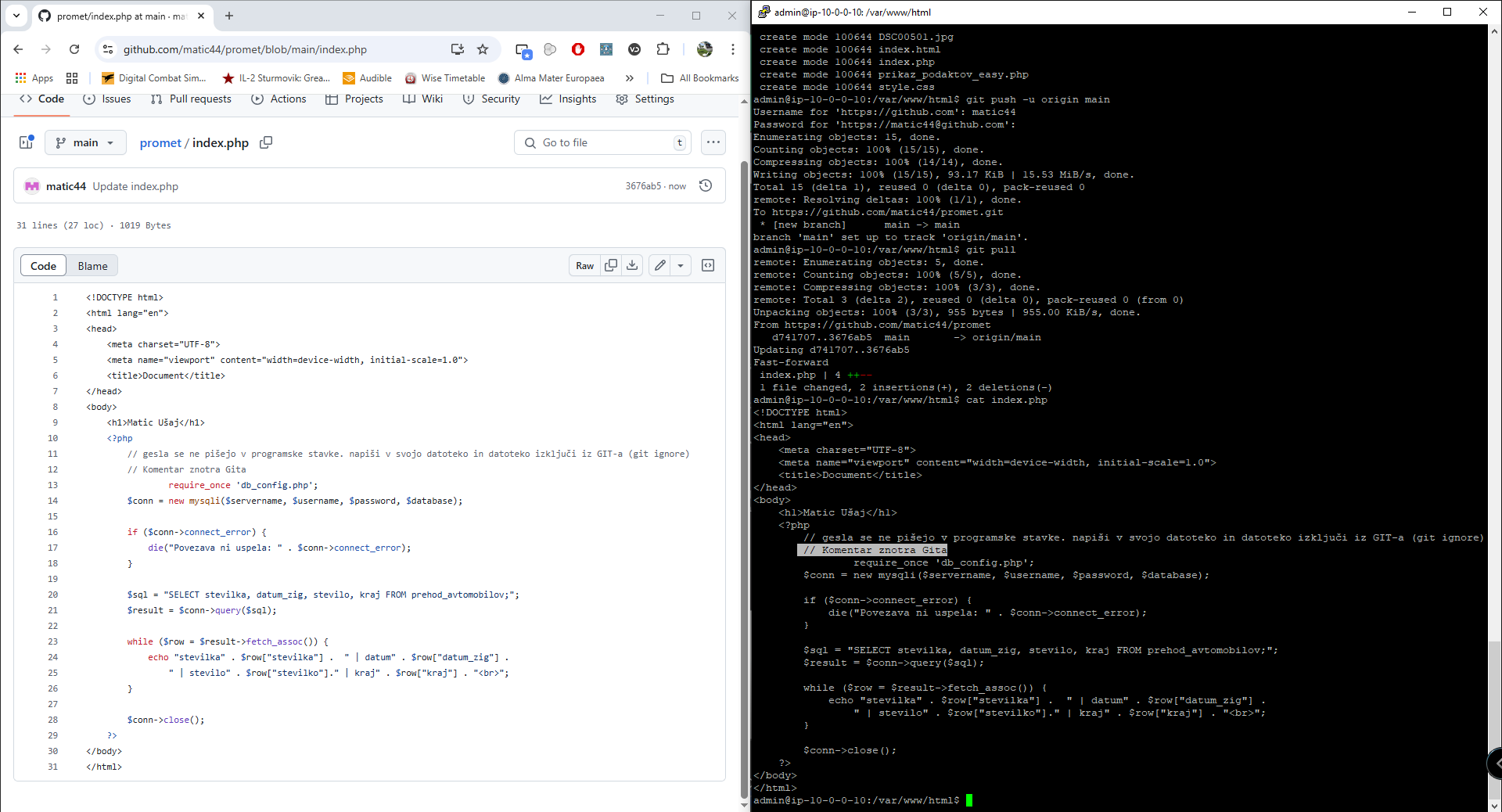
Nato odpremo datoteko in v komentar napišemo komentar.

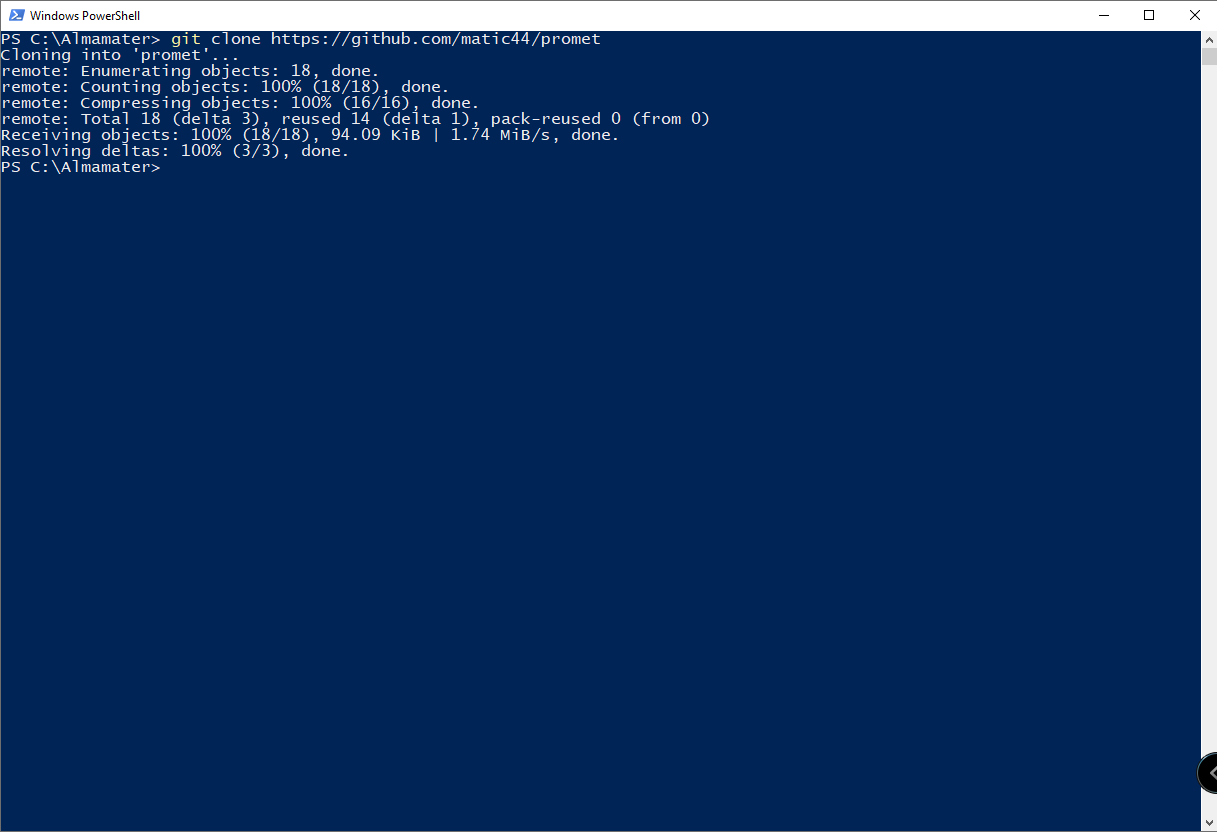


Spremembo objavimo v git u gumbom Cimmit changes.

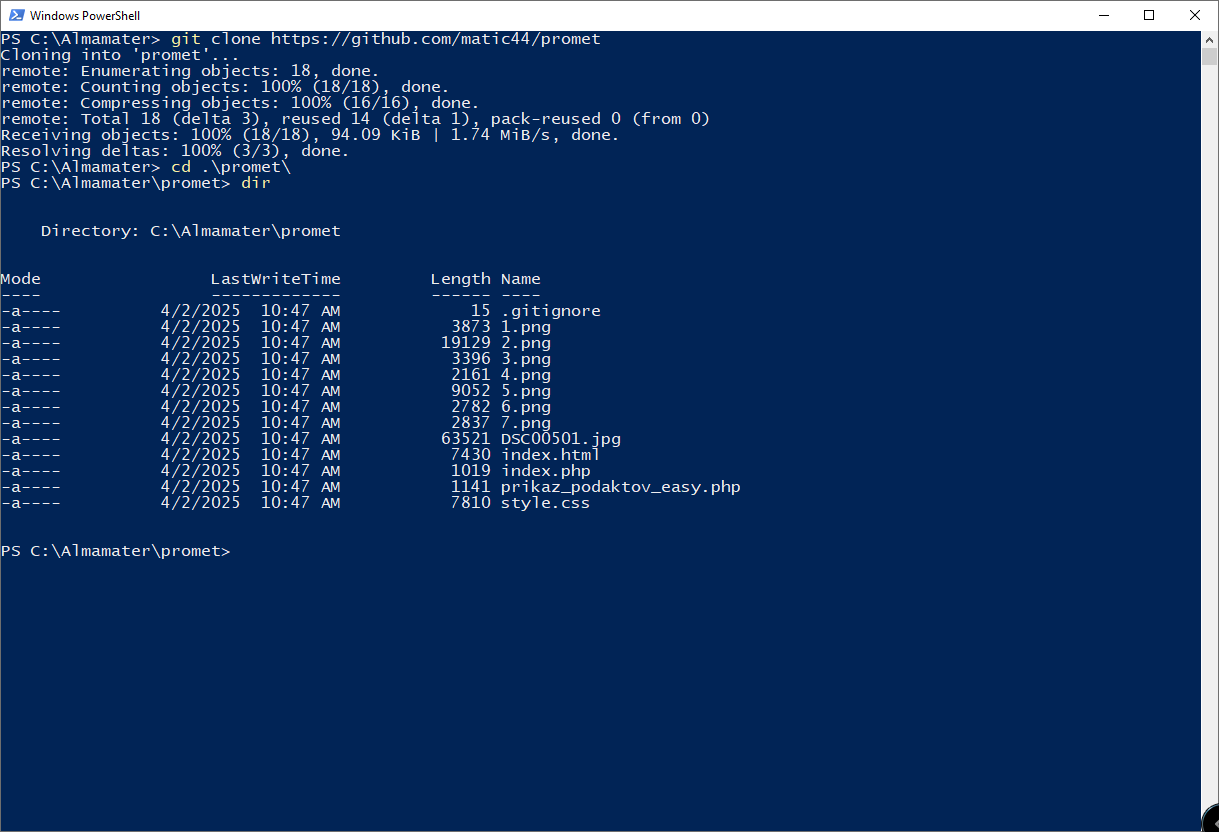


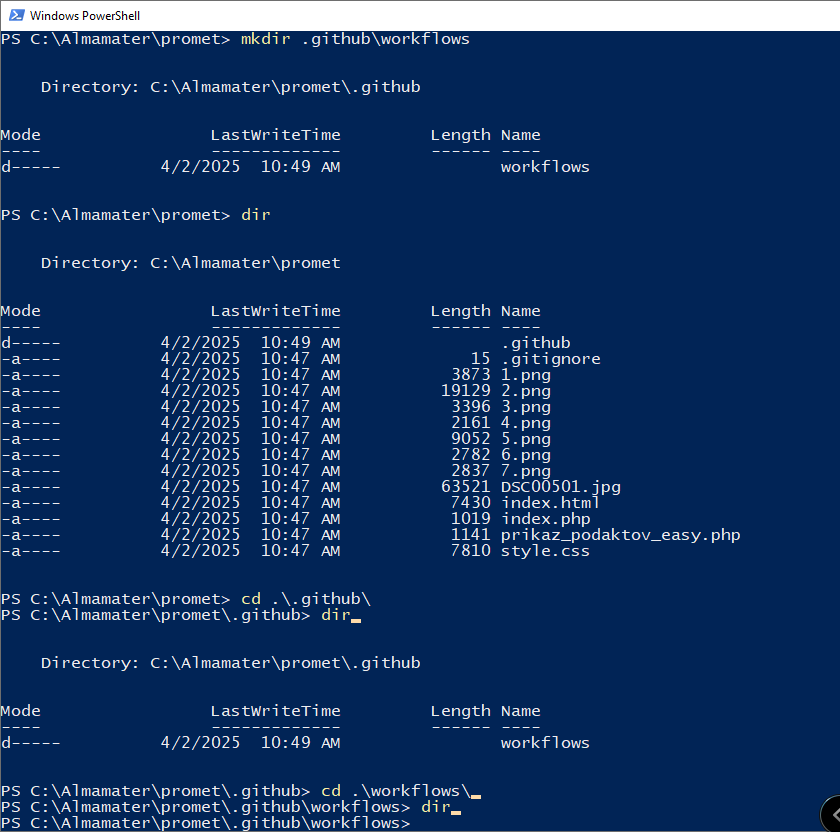
Nato naredimo pull request (git pull) in z ukazom cat index.php odpremo datoteko.



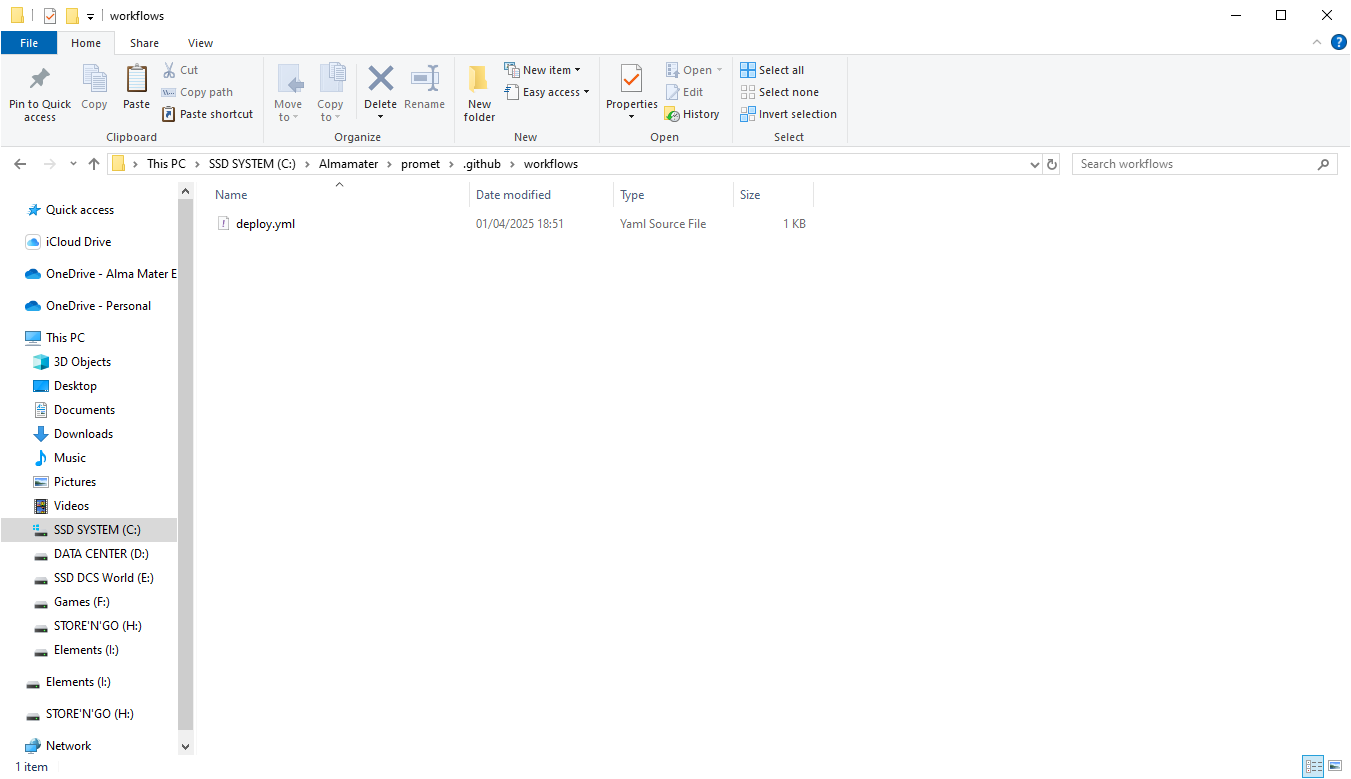
Nato naredimo klon našega git repozitorija na naš lokalni računalnik. git clone [povezava do repozatorija]

Z ukazom dir pregledamo količino elementov, katere so bile prenesene. Z ukazom dir pregledamo vse datoteke.

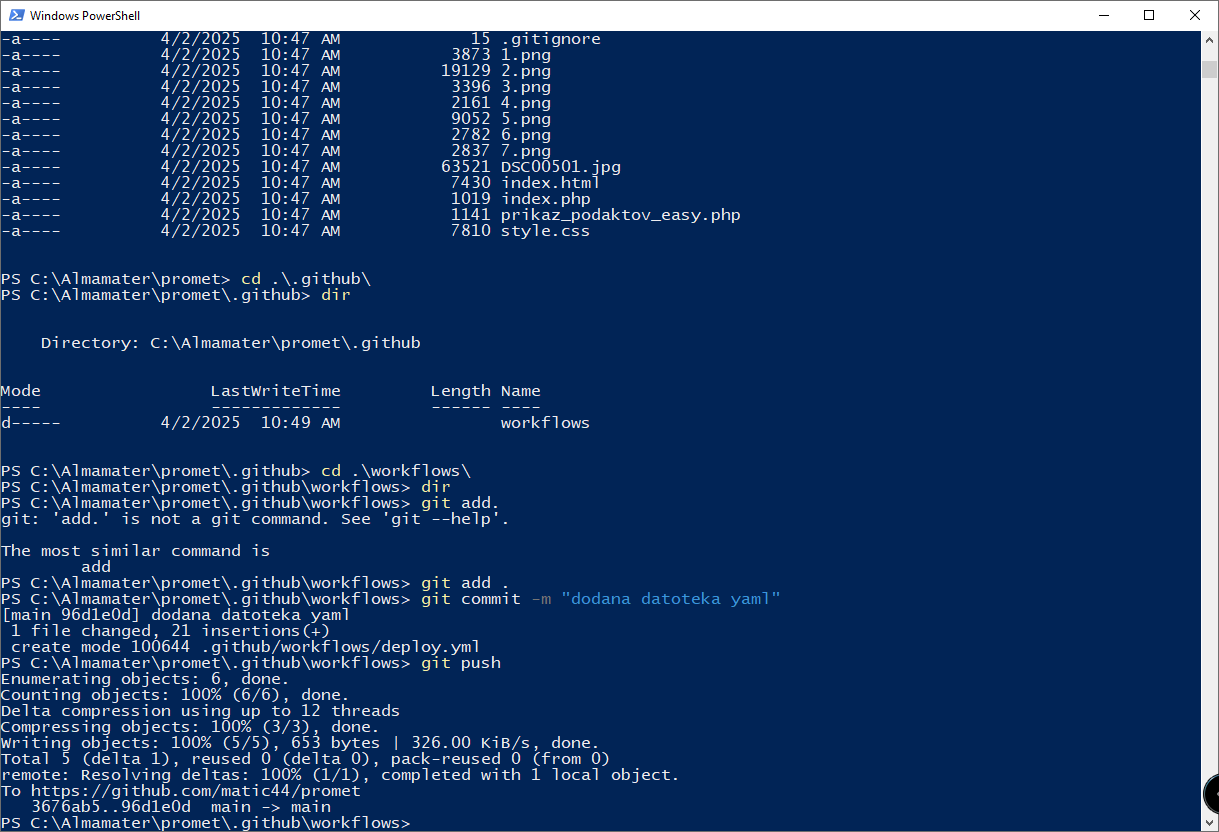


Nato kreiramo novo datoteko z ukazom mkdir .github\workflow in se postavimo znotraj same mape.

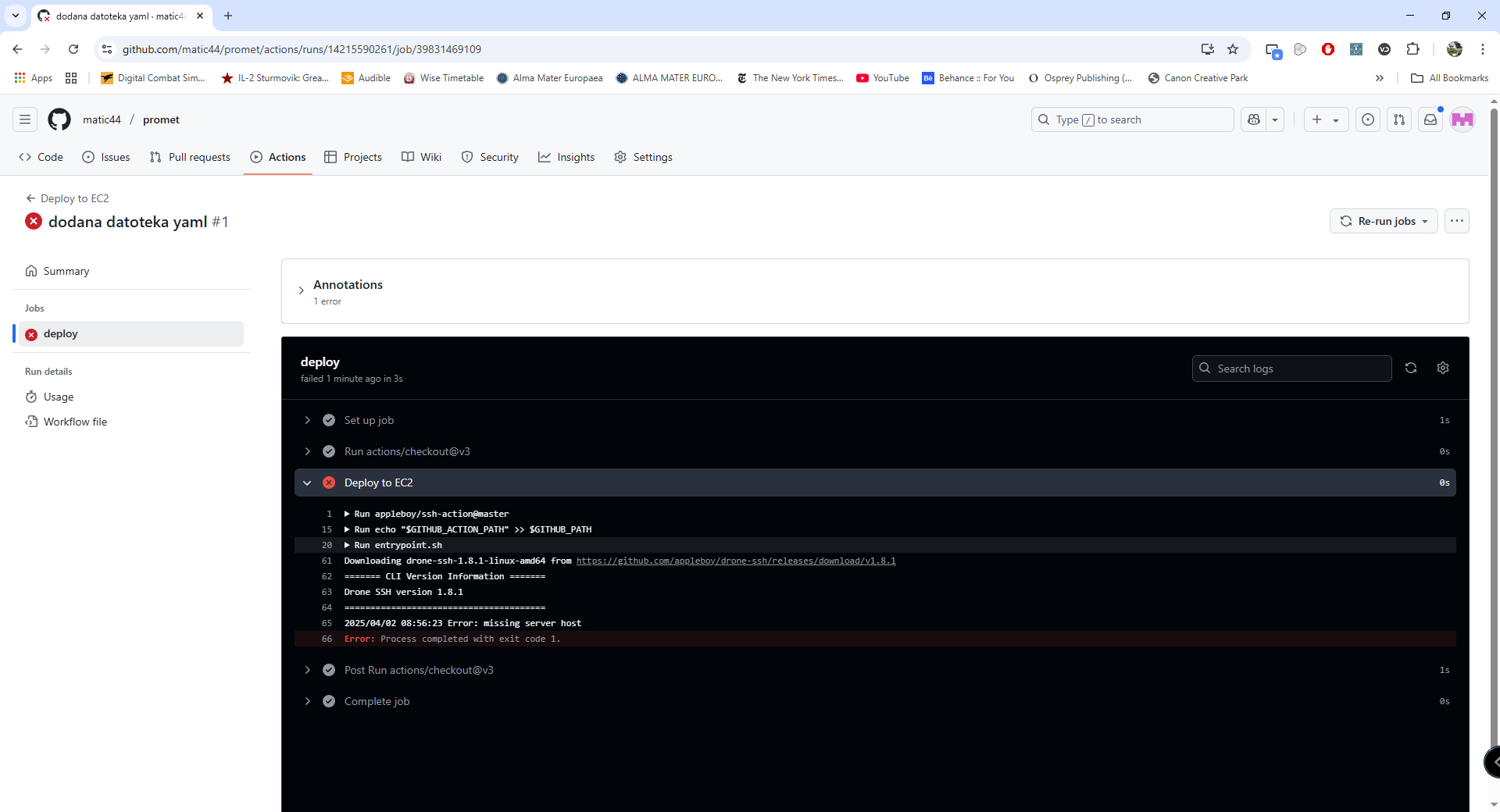
Znotraj te mape prilepimo datoteko deply.yml



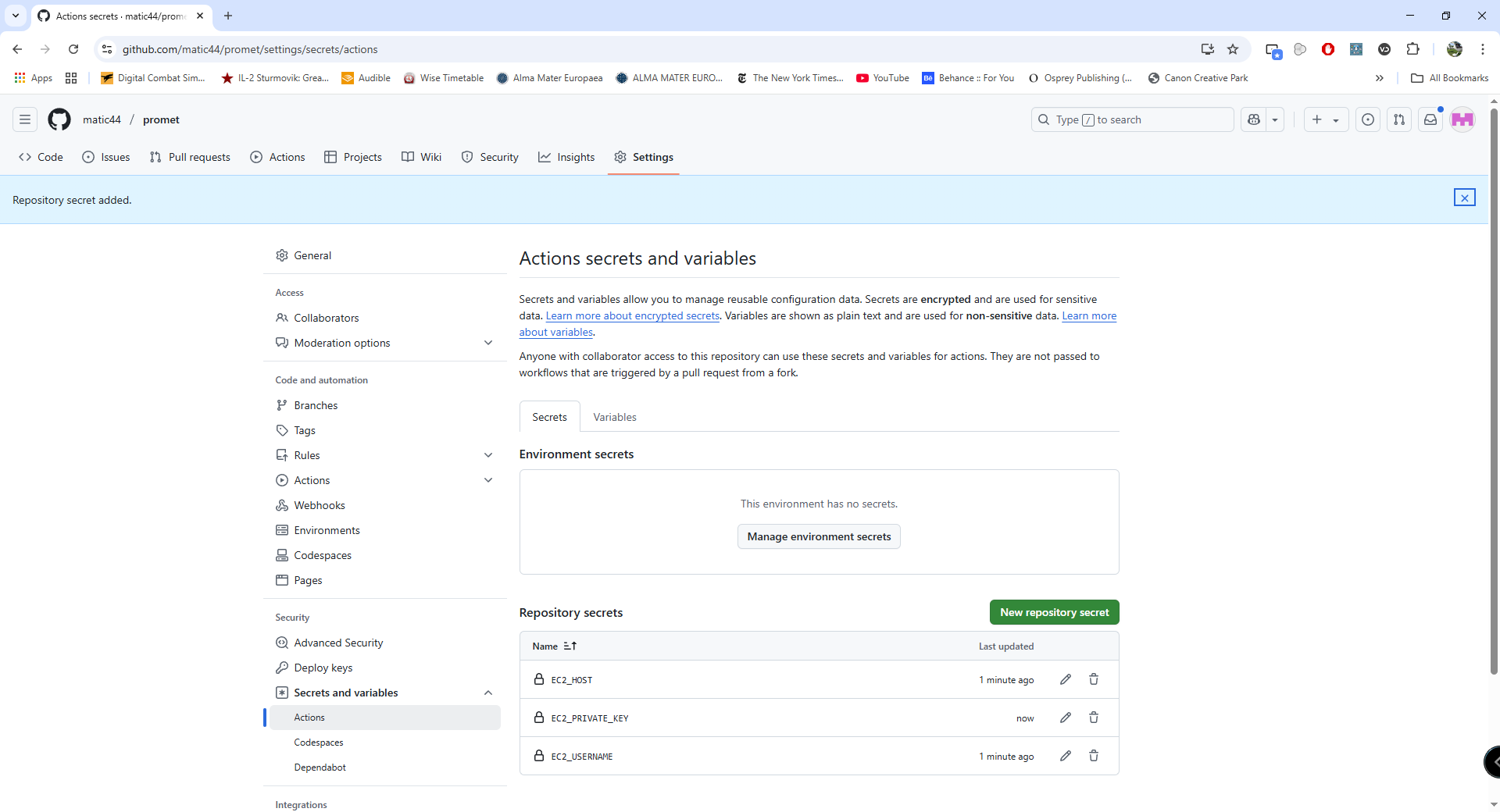
Vse datoteke objavimo na git z ukazom git add, git commit -m “besedilo”, git push



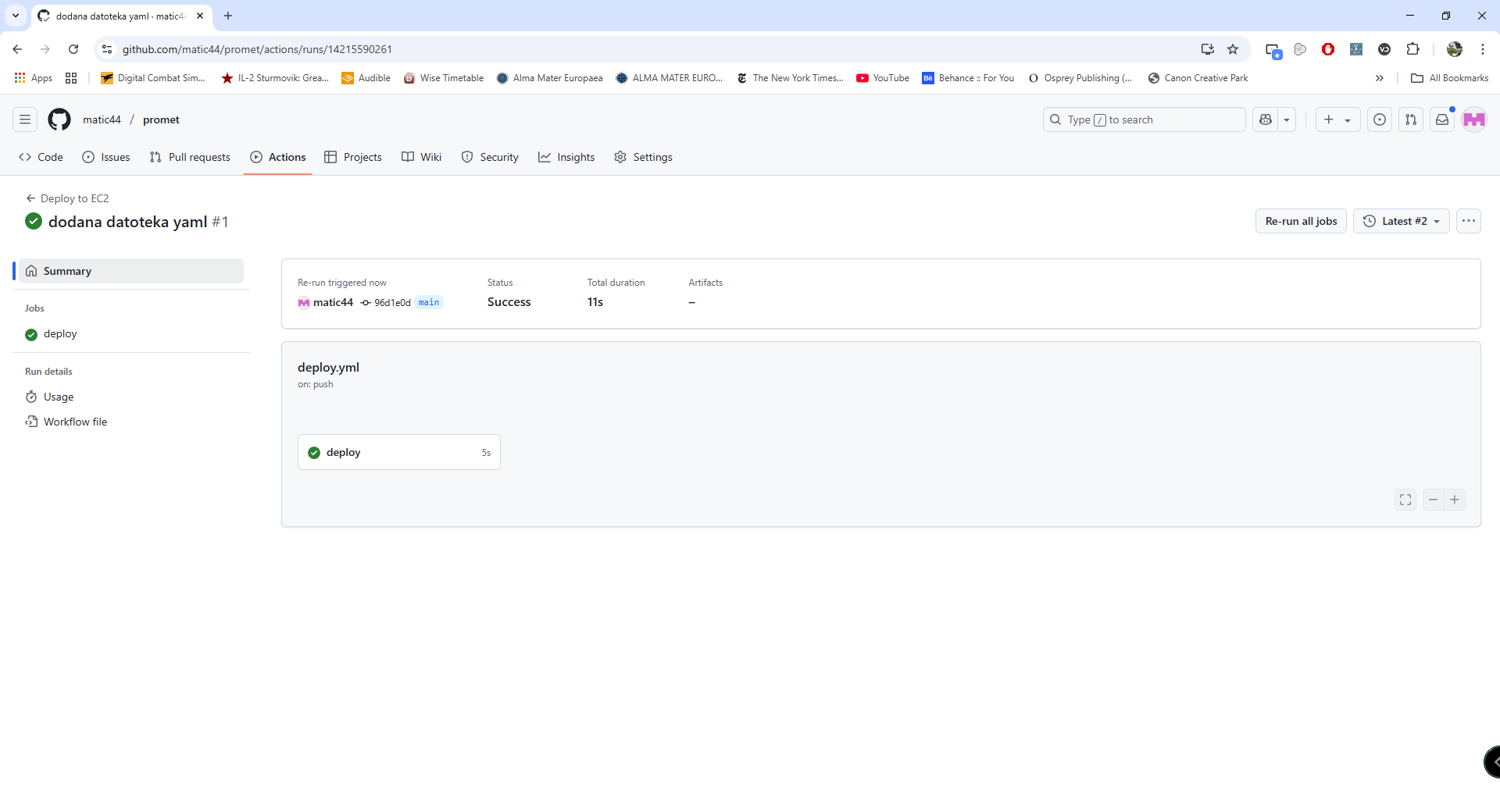
V zavihku “action”, pregled action in odprava napake.

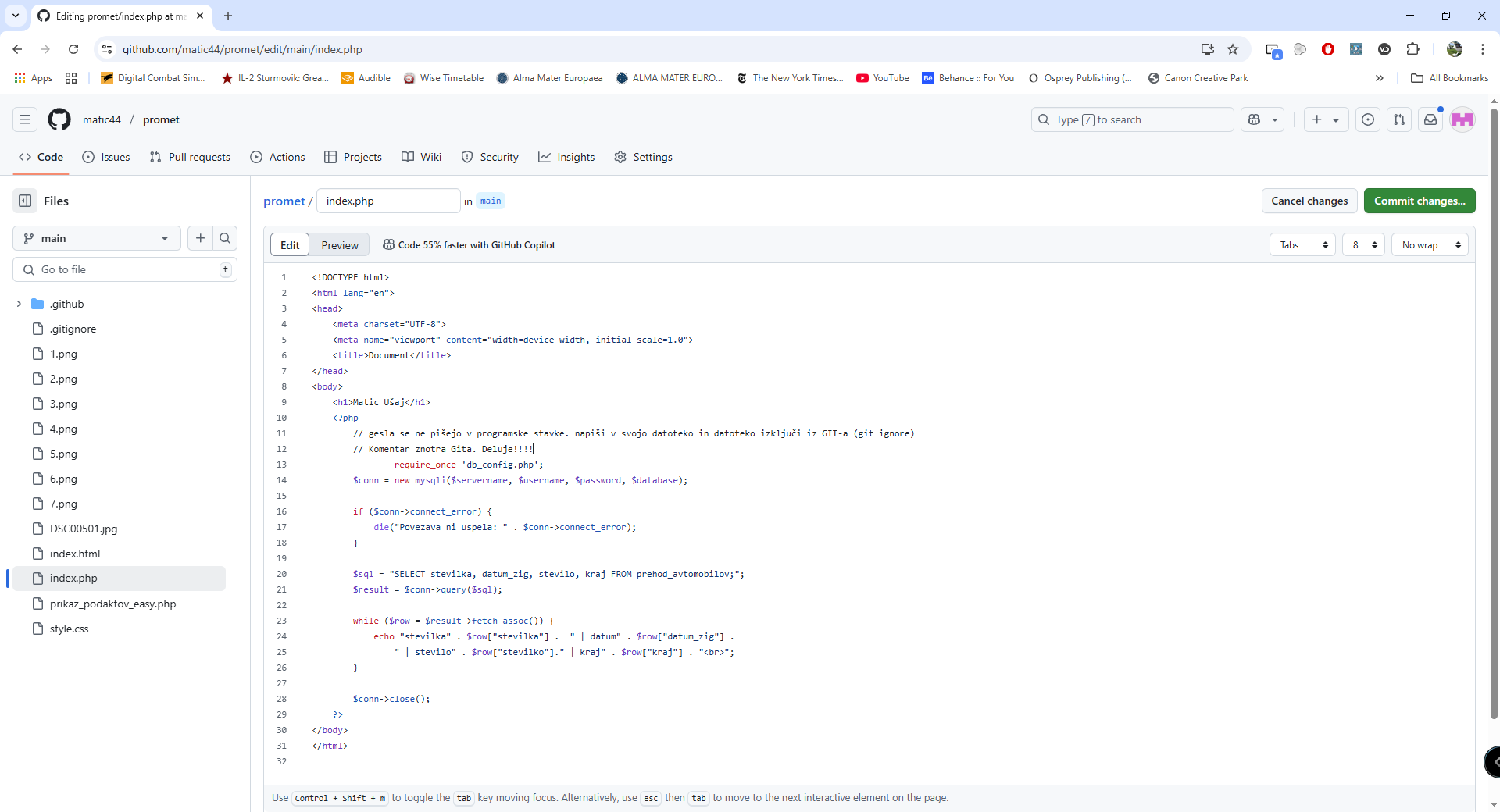


Kreiranje skrivnosti na git repozitoriju za: EC2Host, EC2Private\_key, EC2\_usrename



Ponoven zagon s gumbom “Re-run all jobs”.



Popravimo komentar v kodi.

In zaženemo in preverimo s pul requestom če je skripta res delovala.

