

## **PROJEKAT BR. 1**

### **Dockerizacija višeslojne web aplikacije**

Rok za predaju i prezentaciju: 15.4.2025

Osnovni cilj zadatka jeste da studenti urade kontejnerizaciju web aplikacije koja se sastoji od najmanje tri servisa, koristeći Docker i Docker Compose.

- Opcija 1 – Dati primjer Web aplikacije (moguće osvojiti do 20 bodova)

Potrebno je dockerizovati aplikaciju koja vam je data i ima sljedeću arhitekturu:

1. Frontend: React SPA (Single Page Application)
2. Backend: Node.js REST API (Express)
3. Baza podataka: MongoDB (koristiti image sa Docker Hub-a)

- Opcija 2 – Vlastita aplikacija (do 30 bodova)

Umjesto datog primjera, studenti mogu izabrati drugu web aplikaciju (drugačije tehnologije za frontend, backend, i bazu) sa GitHub-a i dockerizovati je. Takođe moguće da je studenti naprave svoju minimalnu ili koriste već postojeću Web aplikaciju sa različitim tehnologijama i dockerizuju je.

- Uslovi:
  - Aplikacija ne smije imati Dockerfile ili docker-compose.yml
  - Aplikacija mora imati najmanje tri servisa
  - Potrebno je navesti link GitHub repozitorija odakle je preuzet kod
  - Dozvoljeno je koristiti zvanične image-e sa Docker Hub-a (npr. mongo, postgres, nginx, redis...)

Tehnički zahtjevi (vrijede za obje opcije)

- Aplikacija ima najmanje tri servisa (frontend, backend, baza)
- Barem jedan servis mora omogućiti trajnu pohranu podataka
- Barem jedan servis podržava hot-reload (izmjena koda je vidljiva bez rebuildanja image-a)
- Servisi komuniciraju putem Docker mreže
- Jedan od tri servisa mora biti 1 zvanična Docker slika (na Docker Hub-u)
- Konekcija na bazu podataka mora biti konfigurisana putem environment varijabli

Skripte koje trebaju biti dostavljene:

- pripremi\_aplikaciju.sh
  - postavke vezano za pohranu i mrežu
  - Kreiranje/build-anje image-a (ako nisu već gotove)

- pokreni\_aplikaciju.sh
  - Pokreće sve servise aplikacije
  - Omogućava da se kontejneri automatski ponovo pokrenu u slučaju gašenja ili greške
  - Ispisuje poruku s linkom do aplikacije
- zaustavi\_aplikaciju.sh
  - Pauzira aplikaciju (podaci ostaju sačuvani)
- obrisi\_aplikaciju.sh
  - Briše sve: kontejnere, volume, mreže, slike

Potrebno je kreirati:

- Dokument sa kratkim opisom aplikacije i čemu služi
  - Link GitHub repozitorija (ako se koristi vlastiti ili postojeći projekat)
  - Softverske preduslove za pokretanje (Docker, OS...)
  - Arhitekturu aplikacije (nacrti ili opis)
  - Opis servisa, mreža i volume-a
  - Uputstva za pokretanje: pripremi\_aplikaciju.sh, pokreni\_aplikaciju.sh, zaustavi\_aplikaciju.sh, obrisi\_aplikaciju.sh
  - Kako pristupiti aplikaciji (npr. <http://localhost:3000>)

Primjer korištenja

```
./pripremi_aplikaciju.sh
./pokreni_aplikaciju.sh
# Aplikacija dostupna na http://localhost:3000
./zaustavi_aplikaciju.sh
./obrisi_aplikaciju.sh
```

Student je obavezan da usmeno prezentira svoj projekat u terminu predavanja.

Tokom prezentacije, student mora:

- Objasniti arhitekturu i strukturu servisa
- Pokazati rad aplikacije uživo (pokretanje putem skripti)
- Objasniti gdje se koristi perzistentnost
- Na zahtjev pokazati:
  - da se aplikacija može restartovati bez gubitka podataka
  - da koristi zvaničnu Docker sliku (barem za jedan servis)
  - da hot-reload funkcioniše za backend ili frontend
- Prezentacija je obavezna. Projekat neće biti ocjenjen ukoliko student ne prezentuje i odbrani projekat.