Laravel: Laboratorija

Aldin Hodžić

Sadržaj prezentacije

01

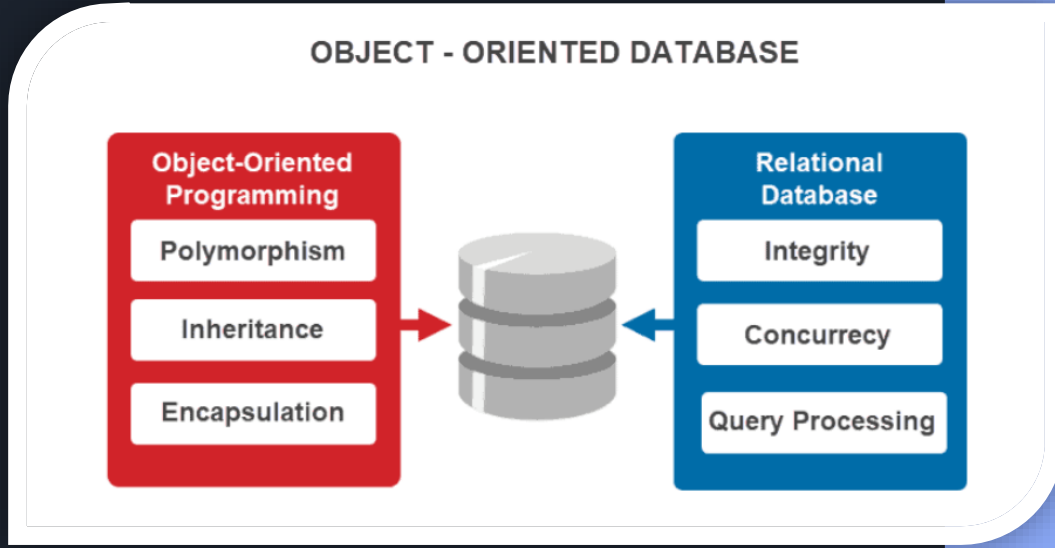
Uvod

Objektno orijentirane baze podataka, Laravel...

O O B P

OOBP predstavljaju tip baza podataka koje podatke čuvaju u obliku objekata, slično kao u OOP.

Za razliku od relacionih baza (tabele, redovi i kolone), OOBP koriste klase, objekte, nasljeđivanje i enkapsulaciju.



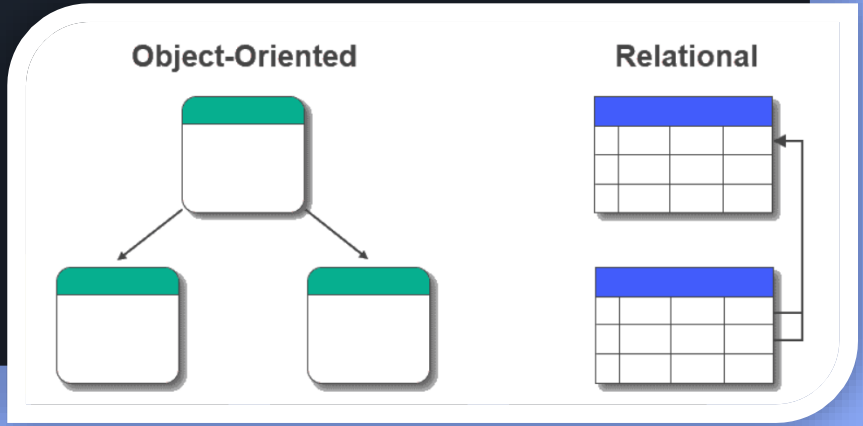
Osnovni elementi:

- Klase – nacrt koji definiše strukturu i ponašanje objekta
- Objekti – instance klase
- Metode – funkcije koje definišu ponašanje objekta
- Pokazivači – povezivanje i pristup objektima

OOBP vs Relaciona BP

- **OOBP**: podaci se čuvaju kao kompletni objekti, složeni tipovi podataka i smanjuje se razlika između baze i programskog jezika.

- **Relacione BP**: podaci organizovani u tabelama, koristi se SQL, fokus na normalizaciji podataka.



Veza sa OOP:

- Polimorfizam - polis + morfi (više + oblika) - „Onaj koji ima mnogo oblika.“
- Nasljeđivanje – hijerarhijski odnos između klasa (roditelj -> dijete)
- Enkapsulacija – kontrola pristupa
- Apstrakcija – fokus na bitne karakteristike, složeni detalji se prikrivaju

Pisanje upita

MySQL	*OODB*
Radi sa tabelama	Radi sa objektima
Koristi JOIN	Koristi reference
SQL	Rad sa klasama
Foreign key	Direktna objektna veza

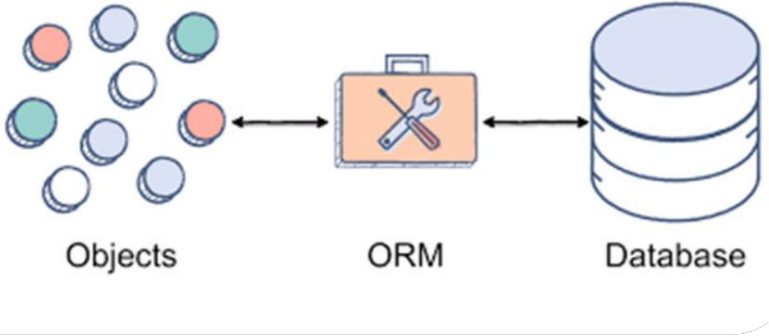
Laravel

PHP web framework za razvoj modernih web aplikacija. Napravljen je da olakša rad programerima tako što nudi gotova rješenja za najčešće funkcionalnosti aplikacije.

Laravel je baziran na PHP jeziku, MVC arhitekturi i ORM sistemu.

Mogućnosti

- Autentifikacija
- Rad sa bazom (MySQL, PostgreSQL)
- ORM
- Slanje mailova
- Routing
- Sigurnost
- API razvoj



ORM

- **M**ehanizam koji omogućava da radiš sa bazom podataka koristeći **objekte i klase**, umjesto da pišeš SQL upite ručno.

SQL `SELECT * FROM patients;`

ORM `$patients = Patient::all();`

Primjer sa vezama

```
SELECT * FROM appointments JOIN patients ON appointments.patient_id = patients.id;
```

```
$appointments = Appointment::with('patient')->get();

//dobiju se svi termini iz baze i uz svaki termin se učita i njegov pacijent - kolekcija objekata
```

02

Laboratorija

Laravel projekat za potrebe laboratorije

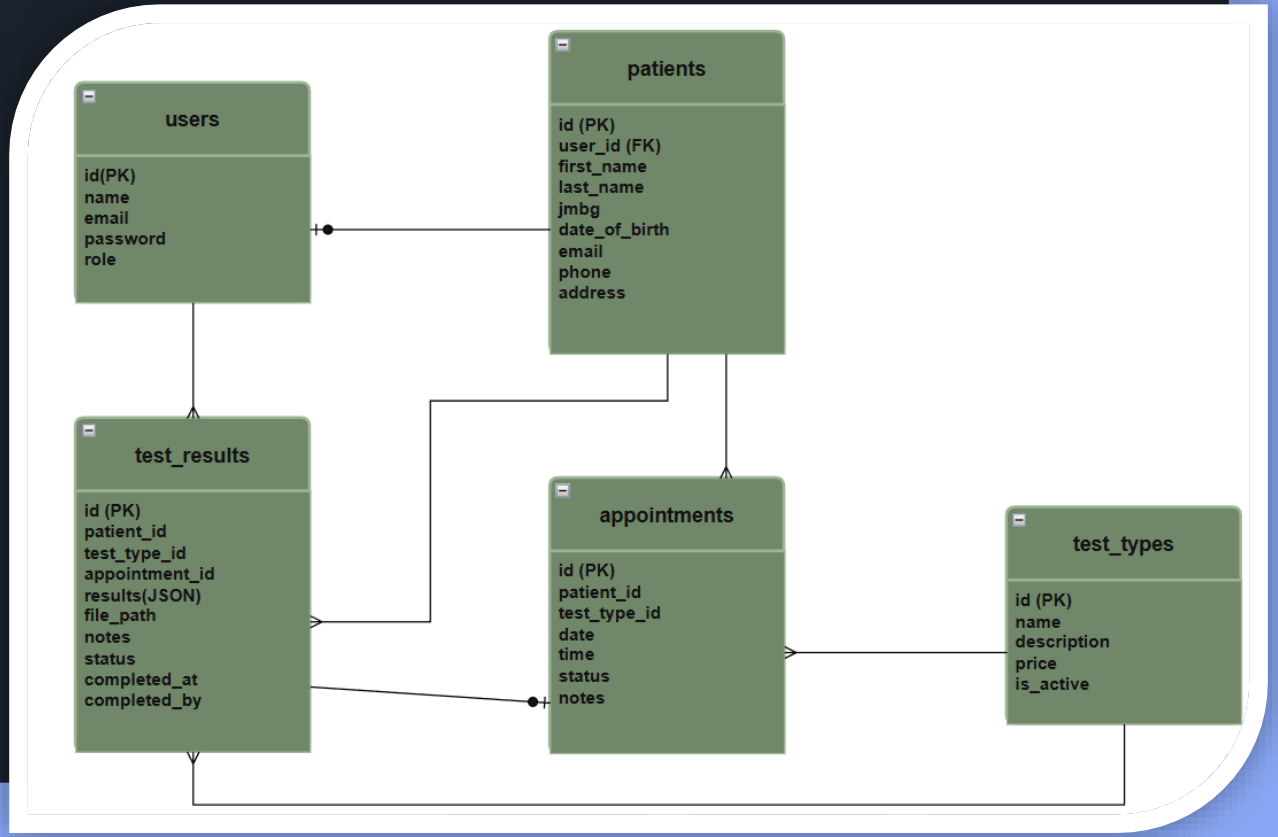
Laboratorija

- Aplikacija programirana u Laravel frameworku.
- Upravljanje pacijentima, zakazivanje termina, unos nalaza...
- Sistem koristi relacionu bazu podataka mySQL, kasnije PostgreSQL.
- Implementira ORM za rad sa podacima.

ER model baze podataka laboratorije

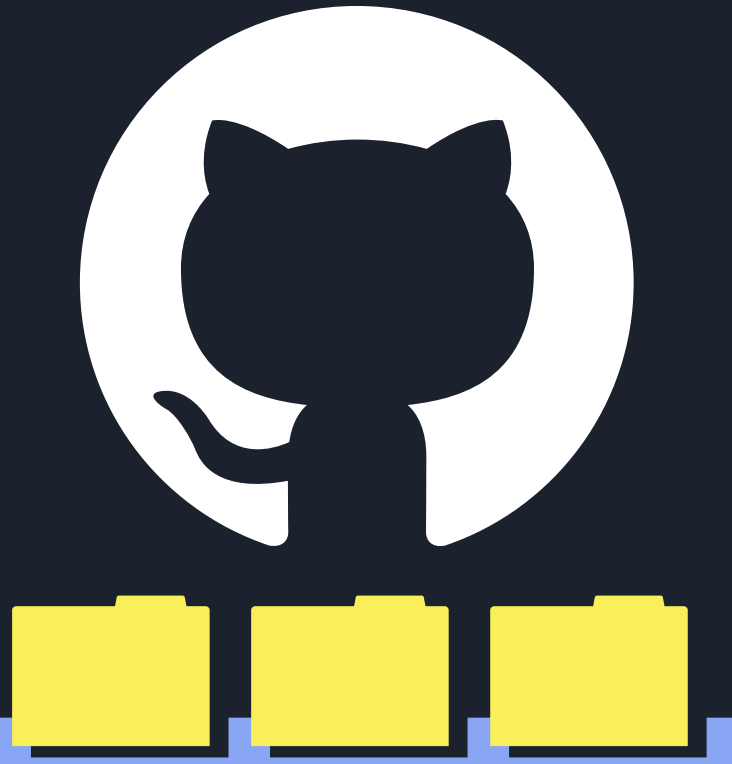
Tabele:

- Users
- Patients
- Test results
- Appointments
- Test types



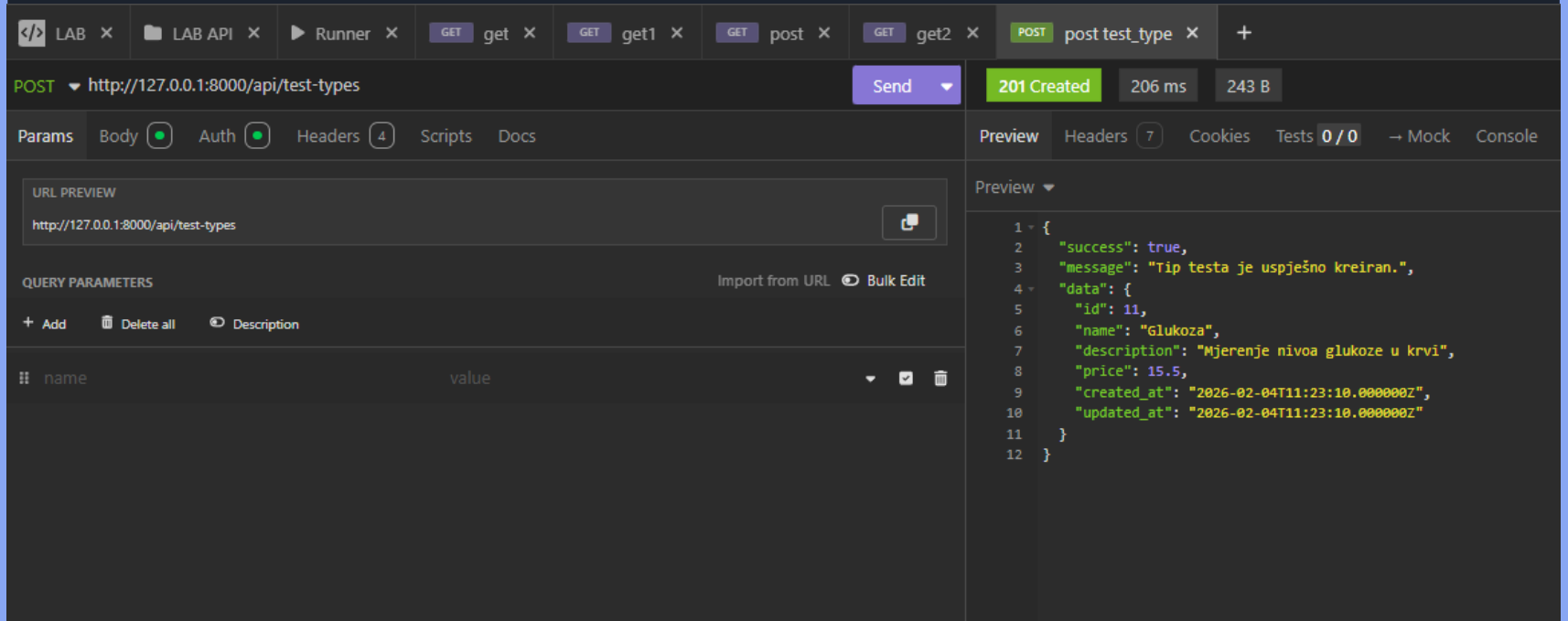
Implementacija

- MVC arhitektura
- Login i register (Jetstream)
- Migracije i relacije više tabela
- Rest API CRUD za pacijente
- Složeni upiti – statistika
- File upload (PDF nalaz)
- Dummy podaci (Seeder/Factory)
- GitHub



Insomnia

Testiranje slanja zahtjeva u Insomniji.



03

PostgreSQL

Osnovni pojmovi,
implementacija

PostgreSQL 

PostgreSQL je napredni sistem za upravljanje bazama podataka koji nudi brojne funkcionalnosti za razvoj aplikacije i upravljanje podacima.

PostgreSQL je **Relational Database Management System** , što znači da:

- Podatke čuva u **tabelama**
- Tabele su povezane putem **relacija (foreign keys)**
- Koristi **SQL (Structured Query Language)**
- Podržava **ACID transakcije** (sigurnost i integritet podataka)

Kada koristiti PostgreSQL?

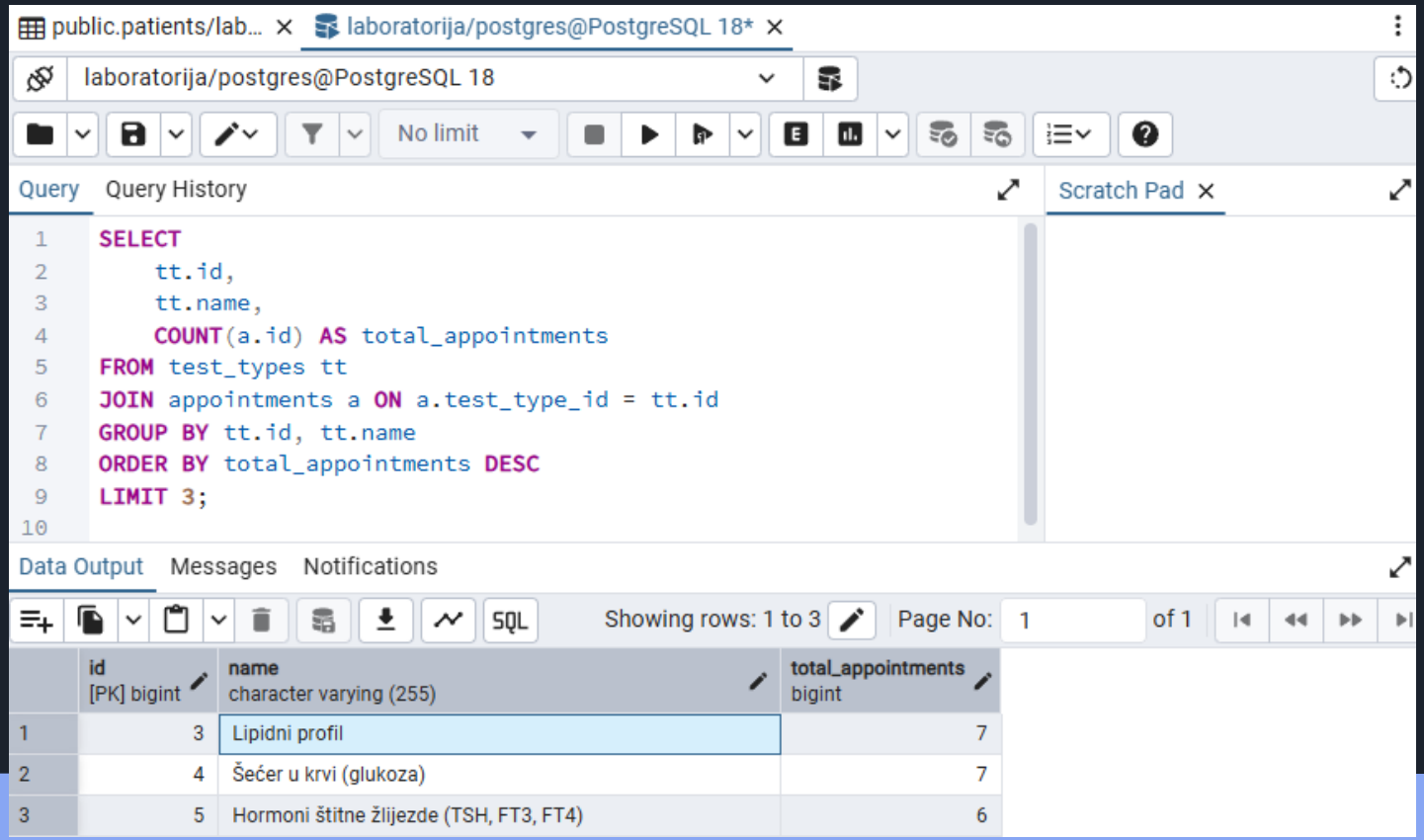
PostgreSQL je pogodno koristiti za:

- Sisteme sa puno relacija
- Kompleksne aplikacije
- Finansijske sisteme
- Analitiku i statistiku
- Projekti koji zahtijevaju visoku sigurnost podataka

PostgreSQL vs MySQL

Kategorija	PostgreSQL	MySQL
Tip baze	ORDBMS	RDBMS
Performanse	Bolji za kompleksne upite i analitiku	Brži za jednostavne SELECT upite
Skalabilnost	Vrlo dobra za velike i kompleksne sisteme	Dobra za web aplikacije srednje veličine
JSON podrška	JSON + JSONB	JSON (osnovna podrška)
Sigurnost	Stroža kontrola tipova, jači integritet podataka	Sigurna (InnoDB), ali fleksibilnija validacija
Potrošnja resursa	Veća (više RAM-a, kompleksniji engine)	Manja, lakši za slabije servere

Primjer upita u PostgreSQL



04

MongoDB

Kako bi izgledala implementacija?

MongoDB

MongoDB je **dokumentna baza podataka** koja podatke čuva u BSON formatu.

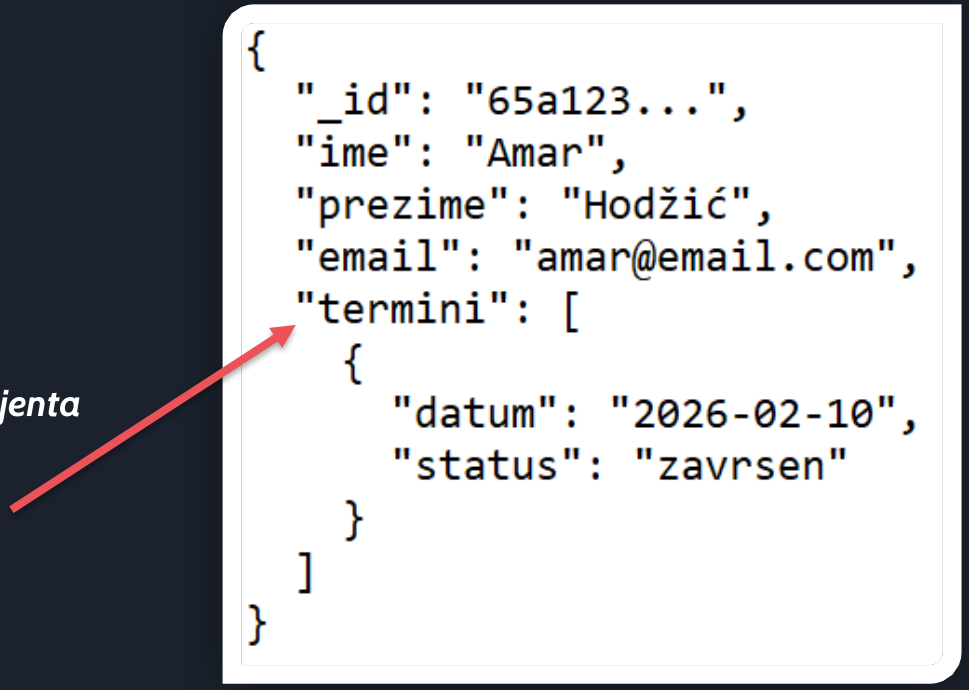
Za razliku od relacionih baza, MongoDB **ne koristi klasične tabele i redove**, već:

- **Collection** (kao tabela)
- **Document** (kao red)
- Polja unutar dokumenta (key–value struktura)

BSON format

Primjetiti:

termini su ugniježđeni unutar ***patcijenta (embedded)***



```
{
  "_id": "65a123...",
  "ime": "Amar",
  "prezime": "Hodžić",
  "email": "amar@email.com",
  "termini": [
    {
      "datum": "2026-02-10",
      "status": "zavrsen"
    }
  ]
}
```

Implementacija

Prebacivanje postojeće Laravel aplikacije na MongoDB znači:

1. promjenu **drivera** (DB konekcije)
2. prilagođavanje **modela i relacija** da rade sa dokumentima umjesto tabela
3. redizajn **upita i statistika** koji sada koriste SQL funkcije

Hvala na pažnji!