**VELEUČILIŠTE U RIJECI**

Preddiplomski stručni studij Telematika

**IT oprema**

Projektna dokumentacija

MENTOR STUDENT

Vlatka Davidović Petar Vuletić

Rijeka, 2024

1. Uvod

U ovom radu opisujemo implementaciju sustava za automatizirano praćenje statusa opreme. Sustav koristi tri povezane baze podataka i SQL trigere kako bi olakšao dohvaćanje informacija o trenutnom stanju opreme (kvar ili servisirano), povijesti kvarova i servisa te osnovnih podataka o opremi.

1. Struktura baza podataka

Kako bi sustav ispravno funkcionirao, kreirane su tri tablice koje imaju specifične funkcije:

1. IT\_oprema\_projekt\_2025: Tablica za osnovne informacije o opremi.
   * Polja:
     + id: Jedinstveni identifikator opreme.
     + naziv: Naziv opreme (npr. laptop, printer).
     + serijski\_broj: Serijski broj opreme.
     + god\_proizvodnje: Godina proizvodnje.
     + dobavljac: Dobavljač opreme.
2. servis\_opreme\_projekt\_2025: Tablica za praćenje povijesti kvarova i servisa.
   * Polja:
     + id: Jedinstveni identifikator zapisa.
     + oprema\_id: Strani ključ koji povezuje zapis s tablicom IT\_oprema\_projekt\_2025.
     + datum\_kvara: Datum kada je kvar prijavljen.
     + kvar\_opis: Opis prijavljenog kvara.
     + datum\_servisa: Datum kada je servis obavljen.
     + servis\_opis: Opis izvršenog servisa.
3. status\_opreme\_projekt\_2025: Tablica koja prati trenutni status opreme.
   * Polja:
     + id: Jedinstveni identifikator zapisa.
     + oprema\_id: Strani ključ povezan s tablicom IT\_oprema\_projekt\_2025.
     + status: Trenutni status opreme (servisirana ili kvar).
     + datum: Datum kada je status postavljen.
     + opis: Opis trenutnog statusa.
   1. Kreiranje tablice ‘IT-oprema\_projekt\_2025’

CREATE TABLE oprema (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Jedinstveni ID za svaku stavku

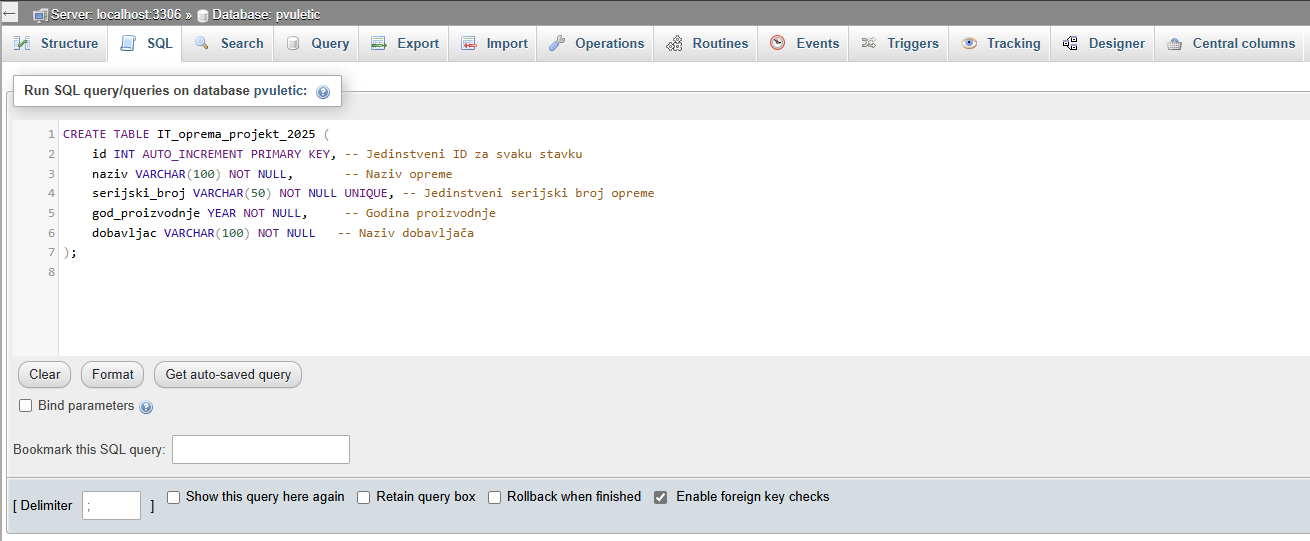
naziv VARCHAR(100) NOT NULL, -- Naziv opreme

serijski\_broj VARCHAR(50) NOT NULL UNIQUE, -- Jedinstveni serijski broj opreme

god\_proizvodnje YEAR NOT NULL, -- Godina proizvodnje

dobavljac VARCHAR(100) NOT NULL -- Naziv dobavljača

);



* 1. Kreiranje tablice ‘servis\_opreme’

CREATE TABLE servis\_opreme\_projekt\_2025 (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Jedinstveni ID zapisa

oprema\_id INT NOT NULL, -- ID opreme (FK na IT\_oprema\_projekt\_2025)

datum\_kvara DATE NOT NULL, -- Datum kada je kvar prijavljen

kvar\_opis TEXT NOT NULL, -- Opis kvara

datum\_servisa DATE, -- Datum kada je servis obavljen

servis\_opis TEXT, -- Opis obavljenog servisa

FOREIGN KEY (oprema\_id) REFERENCES IT\_oprema\_projekt\_2025(id)

ON DELETE RESTRICT -- Ne dopušta brisanje ako postoje povezani zapisi

ON UPDATE CASCADE -- Ažurira strani ključ pri promjeni ID-a u glavnoj tablici

);

A screenshot of a computer

Description automatically generated

* 1. Unos opreme u IT\_oprema\_projekt\_2025

INSERT INTO IT\_oprema\_projekt\_2025 (naziv, serijski\_broj, god\_proizvodnje, dobavljac)

VALUES

('Laptop', 'LAP12345', 2020, 'Dell'),

('Monitor', 'MON67890', 2022, 'LG'),

('Printer', 'PRI54321', 2019, 'HP'),

('Tastatura', 'TAS98765', 2021, 'Logitech'),

('Miš', 'MIS45678', 2023, 'Microsoft'),

('Router', 'ROU11223', 2018, 'TP-Link');

**A screenshot of a computer

Description automatically generated**

* 1. Unos podataka u tablicu servis\_opreme\_projekt\_2025

INSERT INTO servis\_opreme\_projekt\_2025 (oprema\_id, datum\_kvara, kvar\_opis, datum\_servisa, servis\_opis)

VALUES

-- Laptop ima kvar, ali nije servisiran

(1, '2024-12-15', 'Baterija se brzo prazni', NULL, NULL),

-- Monitor je imao kvar i servisiran je

(2, '2024-11-01', 'Problem s prikazom boja', '2024-11-05', 'Zamijenjena LED matrica'),

-- Printer je imao kvar i servisiran je

(3, '2023-10-10', 'Papir se zaglavljuje', '2023-10-15', 'Očišćen mehanizam i zamijenjen valjak'),

-- Tastatura je imala kvar, ali nije servisirana

(4, '2024-05-20', 'Neke tipke ne rade', NULL, NULL),

-- Router je imao kvar i servisiran je

(6, '2024-01-15', 'Prekid mrežne veze', '2024-01-20', 'Ažuriran firmware i zamijenjen adapter');

1. Dohvaćanje svih zapisa iz tablice IT\_oprema\_projekt\_2025

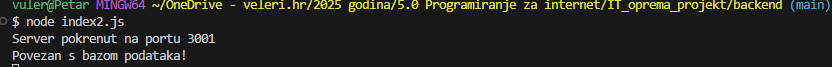
Provjeru radimo na pregledniku Mozila add-on RESTED Client

Priprema prije pokretanja testa moramo u terminal upisati komande

* npm install
* npm install -g nodemon

Nakon toga pokrecemo node index2.js (promjene u kodu zahtijevaju ručno ponovno pokretanje servera. Ctrl+C za prekid) ili nodemon index2.js (Pokreće Node.js program, ali **automatski ponovno pokreće server** kad otkrije promjene u kodu.)

Ako je sve u redu info koji dobijemo u terminal izgleda ovako :



|  |  |  |  |
| --- | --- | --- | --- |
| HTTP metoda | CRUD operacija | Pristupna točka | Objašnjenje |
| GET http://localhost:3001/api/oprema | READ | /api/oprema | Dohvat svih zapisa iz tablice opreme |
| GET http://localhost:3001/api/oprema/1 | READ | /api/oprema/:id | Dohvat pojedinačnog zapisa prema ID-ju |
| POST http://localhost:3001/api/oprema | CREATE | /api/oprema | Unos nove opreme u tablicu |
| PUT http://localhost:3001/api/oprema/1 | UPDATE | /api/oprema/:id | Ažuriranje postojećeg zapisa prema ID-ju |
| DELETE  http://localhost:3001/api/oprema/1 | DELETE | /api/oprema/:id | Brisanje zapisa iz tablice opreme prema ID-ju |

* 1. Dohvaćanje svih zapisa iz tablice IT\_oprema\_projekt\_2025

**URL:** <http://localhost:3001/api/oprema>

**Metoda:** GET

A screenshot of a computer

Description automatically generated

* 1. Dohvaćanje pojedinog zapisa iz tablice IT\_oprema\_projekt\_2025 prema ID-u

**URL**: <http://localhost:3001/api/oprema/1>

**Metoda**: GET

A screenshot of a computer

Description automatically generated

* 1. Dodavanje nove opreme

**URL**: http://localhost:3001/api/oprema

**Metoda**: POST

**Tijelo zahtjeva (JSON)**:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. Ažuriranje opreme

**URL**: http://localhost:3001/api/oprema/1

**Metoda**: PUT

**Tijelo zahtjeva** (JSON):

A screenshot of a computer

Description automatically generated

* 1. Brisanje opreme

**URL**: http://localhost:3001/api/oprema/7

**Metoda**: DELETE

A screenshot of a computer

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| HTTP metoda | CRUD operacija | Pristupna točka | Objašnjenje |
| GET http://localhost:3001/api/povijest | READ | /api/povijest | Dohvaća sve servise povezane s određenom opremom |
| POST  http://localhost:3001/api/servisi | CREATE | /api/servisi | Unos novog servisa za opremu |
| PUT  http://localhost:3001/api/servisi/1 | UPDATE | /api/servisi/:id | Ažuriranje postojećeg servisa prema ID-ju |
| DELETE  http://localhost:3001/api/servisi/1 | DELETE | /api/servisi/:id | Brisanje servisa prema ID-ju |

* 1. Popis svih zapisa iz povijest\_opreme\_projekt\_2025
* **URL**: http://localhost:3001/api/povijest
* **Metoda**: GET

A screenshot of a computer

Description automatically generated

* 1. Dohvaćanje povijesti određene opreme
* **URL**: http://localhost:3001/api/povijest/1
* **Metoda**: GET

A screenshot of a computer

Description automatically generated

* 1. Dodavanje novog zapisa u povjest
* **URL:** [**http://localhost:3001/api/povijest**](http://localhost:3001/api/povijest)
* **Tijelo:** JSON

Vidimo promjenu u SQL-u  
A screenshot of a computer

Description automatically generated

* 1. Ažuriranje servisa
* **URL**: http://localhost:3001/api/servisi/1
* **Metoda**: PUT
* **Tijelo zahtjeva (JSON)**:

A screenshot of a computer

Description automatically generated

* 1. Brisanje servisa
* **URL**: http://localhost:3001/api/servisi/1
* **Metoda**: DELETE

A screenshot of a computer

Description automatically generated

Vidimo promjenu brisanja po id-u servisa

A screenshot of a computer

Description automatically generated

* 1. Kreiranje tablice statusa opreme

CREATE TABLE status\_opreme\_projekt\_2025 (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Jedinstveni ID zapisa

oprema\_id INT NOT NULL, -- ID opreme (FK na IT\_oprema\_projekt\_2025)

status VARCHAR(50) NOT NULL, -- Status opreme: "servisirana" ili "kvar"

datum VARCHAR(50) NOT NULL, -- Datum statusa (servisa ili kvara) kao VARCHAR

opis TEXT, -- Opis statusa (kvara ili servisa)

FOREIGN KEY (oprema\_id) REFERENCES IT\_oprema\_projekt\_2025(id)

ON DELETE CASCADE

ON UPDATE CASCADE

);

INSERT INTO status\_opreme\_projekt\_2025 (oprema\_id, status, datum, opis)

SELECT

oprema\_id,

'servisirana' AS status,

datum\_servisa AS datum,

servis\_opis AS opis

FROM servis\_opreme\_projekt\_2025

WHERE datum\_servisa IS NOT NULL;

INSERT INTO status\_opreme\_projekt\_2025 (oprema\_id, status, datum, opis)

SELECT

oprema\_id,

'kvar' AS status,

datum\_kvara AS datum,

kvar\_opis AS opis

FROM servis\_opreme\_projekt\_2025

WHERE datum\_kvara IS NOT NULL AND datum\_servisa IS NULL;

|  |  |  |  |
| --- | --- | --- | --- |
| HTTP metoda | CRUD operacija | Pristupna točka | Objašnjenje |
| GET http://localhost:3001/api/status\_opreme?status=servisirana | READ | api/status\_opreme?status=servisirana | Dohvaća svu opremu koje su imale servis |
| GET  http://localhost:3001/api/status\_opreme?status=kvar | READ | api/ status\_opreme?status=kvar | Dohvaća svu opremu koje su imale zabilježen kvar |

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. Implementacija trigera

SQL trigeri automatski unose podatke u tablicu status\_opreme\_projekt\_2025 nakon svakog unosa u tablicu povijest\_opreme\_projekt\_2025. Dva glavna trigera su:

1. **Triger za unos servisirane opreme**

DELIMITER $$

CREATE TRIGGER nakon\_unosa\_servisa

AFTER INSERT ON servis\_opreme\_projekt\_2025

FOR EACH ROW

BEGIN

IF NEW.datum\_servisa IS NOT NULL THEN

INSERT INTO status\_opreme\_projekt\_2025 (oprema\_id, status, datum, opis)

VALUES (NEW.oprema\_id, 'servisirana', NEW.datum\_servisa, NEW.servis\_opis);

END IF;

END$$

DELIMITER ;

**Opis:**

* Aktivira se nakon svakog unosa u tablicu servis\_opreme\_projekt\_2025.
* Provjerava je li datum\_servisa popunjen.
* Ako je uvjet ispunjen, unosi novi zapis u tablicu status\_opreme\_projekt\_2025 sa statusom servisirana.

1. **Triger za unos opreme s kvarovima**

DELIMITER $$

CREATE TRIGGER nakon\_unosa\_kvara

AFTER INSERT ON servis\_opreme\_projekt\_2025

FOR EACH ROW

BEGIN

IF NEW.datum\_kvara IS NOT NULL AND NEW.datum\_servisa IS NULL THEN

INSERT INTO status\_opreme\_projekt\_2025 (oprema\_id, status, datum, opis)

VALUES (NEW.oprema\_id, 'kvar', NEW.datum\_kvara, NEW.kvar\_opis);

END IF;

END$$

DELIMITER ;

**Opis:**

* Aktivira se nakon svakog unosa u tablicu servis\_opreme\_projekt\_2025.
* Provjerava je li datum\_kvara popunjen i je li datum\_servisa prazan.
* Ako su uvjeti ispunjeni, unosi novi zapis u tablicu status\_opreme\_projekt\_2025 sa statusom kvar.

**Primjeri unosa i rezultata**

1. **Unos servisirane opreme:**

INSERT INTO servis\_opreme\_projekt\_2025 (oprema\_id, datum\_kvara, kvar\_opis, datum\_servisa, servis\_opis)

VALUES (1, '2024-01-01', 'Baterija ne radi', '2024-01-10', 'Zamjena baterije');

**Rezultat u status\_opreme\_projekt\_2025:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Oprema\_id | Status | Datum | Opis |
| 1 | 1 | Servisirana | 10-01-2024 | Zamjena baterije |

1. **Unos opreme s kvarom:**

INSERT INTO servis\_opreme\_projekt\_2025 (oprema\_id, datum\_kvara, kvar\_opis, datum\_servisa, servis\_opis)

VALUES (2, '2024-02-01', 'Ne uključuje se', NULL, NULL);

**Rezultat u status\_opreme\_projekt\_2025:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Id | Oprema\_id | Status | Datum | Opis | | 2 | 2 | Kvar | 01-02-2024 | Ne uključuje se | |  |  |  |  |
|  |  |  |  |  |

**Što triger radi?**

* Aktivira se nakon svakog **INSERT** u tablicu **servis\_opreme\_projekt\_2025**.
* Provjerava:
  + Je li **datum\_kvara** ispunjen (IS NOT NULL).
  + Je li **datum\_servisa** prazan (IS NULL), što znači da kvar još nije servisiran.
* Ako su uvjeti ispunjeni:
  + Dodaje zapis u **status\_opreme\_projekt\_2025** sa statusom **kvar**.
  + Ubacuje podatke:
    - oprema\_id → ID opreme s kvarom.
    - status → "kvar".
    - datum → Datum prijave kvara (datum\_kvara iz originalne tablice).
    - opis → Opis kvara (kvar\_opis iz originalne tablice).

Ovi trigeri olakšavaju upravljanje podacima između tablica i osiguravaju točnost statusa opreme. Automatizacija pomoću trigera smanjuje potrebu za ručnim unosom i ubrzava dohvaćanje podataka za analizu i izvještavanje. Sustav je fleksibilan i lako se može prilagoditi dodatnim zahtjevima, kao što su složeniji izvještaji ili analiza povijesnih podataka.

1. Quasar App
   1. Priprema u frontend

cd "C:\Users\vuler\OneDrive - veleri.hr\2025 godina\5.0 Programiranje za internet\IT\_oprema\_projekt\frontend\quasar-project"

*quasar dev*

*A screenshot of a computer

Description automatically generated*

<http://localhost:9000/>

* 1. Main Layout

const linksList = [

  {

    title: 'Početna',

    caption: 'Početna stranica',

    icon: 'school',

    link: '/'

  },

  {

    title: 'Pregled opreme',

    caption: 'Pregled svih podataka o opremi',

    icon: 'inventory',

    link: '#/pregled\_opreme/'

  },

  {

    title: 'Unos opreme',

    caption: 'Dodavanje nove opreme',

    icon: 'add\_circle',

    link: '#/unos\_opreme/'

  },

  {

    title: 'Servis opreme',

    caption: 'Prikaz servisirane opreme',

    icon: 'build',

    link: '#/povijest\_opreme/'

  },

  {

    title: 'Status opreme',

    caption: 'Pregled statusa opreme',

    icon: 'info',

    link: '#/status\_opreme/'

  }

]

*A screenshot of a computer

Description automatically generated*

* 1. Rute

const routes = [

  {

    path: '/',

    component: () => import('layouts/MainLayout.vue'),

    children: [

      {

        path: '',

        component: () => import('pages/IndexPage.vue')

      },

      {

        path: 'pregled\_opreme',

        component: () => import('pages/PregledOpremePage.vue')

      },

      {

        path: 'unos\_opreme',

        component: () => import('pages/UnosOpremePage.vue')

      },

      {

        path: 'povijest\_opreme',

        component: () => import('pages/PovijestOpremePage.vue')

      },

      {

        path: 'status\_opreme',

        component: () => import('pages/StatusOpremePage.vue')

      }

    ]

  },