

OPŠTI PODACI O PROJEKTNOM TIMU				
Br grupe	Broj indeksa	Ime	Prezime	Email adresa
	PR96/2022	Filip	Velemir	velemirfilip@gmail.com
{{Broj grupe 2}}	PR91/2022	Maja	Bogičević	maja.bogicevic01@gmail.com
Github link				
https://github.com/ficadjole/vp_projekat.git				

OPŠTI PODACI O PROJEKTU	
Naziv projekta:	Analiza i nadzor podataka o radu PMSM motora korišćenjem WCF servisa i manipulacije fajlovima
TEHNIČKI OPIS PROJEKTA	
Sažetak:	<p>Projekat obuhvata implementaciju sistema za analizu i nadzor rada permanent magnet synchronous (PMSM) motora na testnom stendu, zasnovanog na dataset-u prikupljenom od strane LEA odeljenja na Univerzitetu u Paderbornu. Sistem omogućava detekciju anomalija, praćenje ključnih parametara motora i obaveštavanje korisnika o potencijalnim problemima korišćenjem WCF servisa, manipulacije memorijskim tokovima i fajlovima, kao i implementaciju delegata i događaja. Projekat integriše sve obrađene oblasti i povezuje ih u funkcionalnu aplikaciju za nadzor i analizu motora.</p>
Opis projekta:	<p>Projekat „PMSM Motor Monitoring“ zasniva se na objedinjavanju različitih tehnika i koncepata u C#, sa ciljem obrade i organizovanja realnih podataka o motoru. Dataset se nalazi u okviru pripremljene arhive i koristi se kao osnovni materijal za implementaciju.</p> <p>U okviru projekta potrebno je:</p> <ul style="list-style-type: none"> • razviti i konfigurisati WCF servis koji omogućava pristup i osnovne operacije nad podacima, • implementirati Dispose pattern za pravilno upravljanje memorijom i resursima, • omogućiti rad sa fajlovima i direktorijumima, uključujući učitavanje i snimanje podataka iz dataset-a, • realizovati prenos i manipulaciju fajlovima preko mreže korišćenjem različitih tokova, • razviti sistem delegata i događaja za notifikaciju o statusu obrade i ključnim promenama. <p>Na ovaj način se obezbeđuje povezivanje svih ključnih oblasti obrađenih kroz vežbe u koherentnu aplikaciju za nadzor i analizu PMSM motora. Krajnji rezultat je aplikacija koja omogućava pregled, organizaciju i analizu podataka u realnom vremenu, pružajući osnovu za detekciju kvarova, optimizaciju performansi i unapređenje strategija kontrole motora.</p>
Baza podataka:	https://www.kaggle.com/datasets/wkirsngn/electric-motor-temperature
Zadaci:	<p>Potrebno je proveriti da li je struja u d/q komponentama naglo skočila i podići događaj.</p> $\Delta u_d = u_d(t) - u_d(t - \Delta t), \quad \Delta u_q = u_q(t) - u_q(t - \Delta t)$ $ \Delta u_d > U_{d_threshold} \quad \text{ili} \quad \Delta u_q > U_{q_threshold}$ <p>Potrebno je proveriti da li motor brzo odstupa od referentne brzine.</p>

	$\Delta\text{speed} = \text{motor_speed}(t) - \text{motor_speed}(t - \Delta t)$ <p>Napomena:</p> <ul style="list-style-type: none">• Ako je $\Delta\text{speed} > \text{Speed_threshold}$, treba podići događaj.
--	---

Zadaci:

1	Zadatak: Skica sistema i pravila protokola	Broj poena: 2
	Opis: Napraviti skicu arhitekture (klijent ↔ WCF servis ↔ skladište na disku) i jasno definisana pravila slanja : (a) svaka sesija ima meta-zaglavlje {U_q,U_d,Motor_Speed,Profile_Id,Ambient,Torque}; (b) sekvencijalno slanje – klijent prolazi for petljom kroz CSV i šalje po jedan red ; (c) poruke StartSession , PushSample , EndSession ; (d) server vraća ACK/NACK i status IN_PROGRESS/COMPLETED ; (e) pragovi su u konfiguraciji: Ud_threshold , Uq_threshold , Speed_threshold i ±25% odstupanje od tekućeg proseka .	
2	Zadatak: WCF servis, konfiguracija i ugovori	Broj poena: 4
	Opis: Definirati ServiceContract sa operacijama StartSession(meta) , PushSample(sample) , EndSession() . DataContract za MotorSample {U_q,U_d,Motor_Speed,Profile_Id, Ambient,Torque} . Podesiti netTcpBinding (streaming, MaxReceivedMessageSize , timeouts) i hostovanje.	
3	Zadatak: WCF servis, operacije i validacija podataka	Broj poena: 3
	Opis: Implementirati logiku servisa: validacija tipova/jedinica, postojanje obaveznih polja, dozvoljeni opsezi (npr. Motor_Speed > 0). Na greške vraćati standardizovane izuzetke (npr. DataFormatFault , ValidationFault).	
4	Zadatak: Dispose pattern i upravljanje resursima	Broj poena: 5
	Opis: Ispravno implementirati IDisposable u sloju čitanja/pisanja (reader, writer, stream wrapperi) i dokazati zatvaranje resursa kroz test/simulaciju izuzetka (npr. prekid veze usred prenosa).	
5	Zadatak: Rad sa fajlovima, učitavanje CSV na klijentu	Broj poena: 3
	Opis: Odrediti putanju do baze podataka i učitati ceo dataset. Parsirati CSV (invariant culture, tačka kao decimalni separator), učitati prvih 100 redova . Nevalidne/redove viška prijaviti u izdvojeni log.	
6	Zadatak: Snimanje i organizacija fajlova na serveru	Broj poena: 2
	Opis: Pri StartSession kreirati strukturu measurements_session.csv i tokom prijema nadovezati redove (FileStream/StreamWriter). Kreirati i poseban rejects.csv za odbačena merenja.	
7	Zadatak: Mrežni prenos i tokovi, sekvencijalni streaming	Broj poena: 6
	Opis: Klijent šalje po jedan red (sample) u realnom vremenu. Na serveru ispisivati status „prenos u toku...” i „završen prenos”.	
8	Zadatak: Delegati i događaji	Broj poena: 5

	Opis: Implementirati događaje: <code>OnTransferStarted</code> , <code>OnSampleReceived</code> , <code>OnTransferCompleted</code> , <code>OnWarningRaised</code> . Pretplata na događaje radi logovanja i/ili konzolnih obaveštenja. Pragovi dolaze iz konfiguracije (<code>app.config</code>): <code>Speed_threshold</code> , <code>Uq_threshold</code> , <code>Ud_threshold</code> , kao i prag za ±25% odstupanje od tekućeg proseka (računanje srednje vrednosti po sesiji).	
9	Zadatak: Analitika 1, detekcija naglih promena strujnih komponenti (ΔI_q , ΔI_d)	Broj poena: 6
	Opis: Za uzastopne uzorke izračunati $\Delta U_q = U_q[n] - U_q[n-1]$. Ako je $ \Delta U_q > U_q_threshold$ → podići događaj <code>VoltageSpikeQ</code> (sa smerom: „ispod/iznad očekivanog“). Istu proveru uraditi i za <code>Ud</code> strujnu komponentu → podići događaj <code>VoltageSpikeD</code> . <code>Threshold</code> parametri su u konfiguraciji.	
10	Zadatak: Analitika 2, detekcija naglih promene temperature rashladne tečnosti(ΔT)	Broj poena: 6
	Opis: Za uzastopne uzorke izračunati $\Delta Speed = motor_speed[n] - motor_speed[n-1]$. Ako je $ \Delta Speed > Speed_threshold$ → podići događaj <code>SpeedSpike</code> (sa smerom: „ispod/iznad očekivanog“). Paralelno pratiti tekući prosek <code>Speed_mean</code> (running mean) i na svakoj iteraciji proveriti odstupanje ±25% : ako je <code>Speed < 0.75 · Speed_mean</code> ili <code>Speed > 1.25 · Speed_mean</code> → podići <code>OutOfBandWarning</code> sa smerom („ispod/iznad očekivane vrednosti“). <code>Threshold</code> parametar je u konfiguraciji.	