

## **Bakgrunn**

**TechSupport AS** er et lite IT-selskap med 15 ansatte som leverer teknisk support til lokale bedrifter. Daglig mottar de mellom 20-30 henvendelser via e-post, telefon og chat - alt fra "jeg har glemt passordet mitt" til "serveren er nede!".

Kundehenvendelsene forsvinner i e-postkasser, viktige saker blir glemt, og ingen vet helt hvem som jobber med hva. Sjefen, Kari, har fått nok av kaoset. I forrige uke gikk en kritisk henvendelse fra kommunen tapt i innboksen - det kostet selskapet 50 000 kr i tapt kontrakt.

## **Oppdraget**

Kari har hørt at dere går på IT-linja og trenger et enkelt ticketsystem. Budsjettet er stramt, så store løsninger som Zendesk eller Jira er uaktuelt. Hun vil ha noe dere kan bygge og tilpasse selv.

- **Sikkerhet** - ansatte må ha egne kontoer med sikker innlogging
- **Oversikt** - hvem jobber med hva? Hva er statusen?
- **Videre utvikling** - systemet må legges på GitHub slik at andre kan videreutvikle og forbedre det over tid

Kari vet at selskapet vokser, og at andre IT-elever eller utviklere kanskje må ta over og legge til nye funksjoner senere. Derfor er det viktig at koden er **oversiktlig** og **tilgjengelig** for videre arbeid.

## **Oppgave**

Dere skal utvikle første versjon av dette ticketsystemet. Det trenger ikke være fancy, men det må **fungere** og være **lett å jobbe videre med**. Koden skal legges på GitHub med kommentarer slik at andre kan klonе, forbedre og utvide systemet når det trengs. **Kravene til oppgaven står beskrevet på neste side.**

### **Grunnleggende oppsett:**

- Flask-applikasjon som kjører
- MariaDB database med 2 tabeller (users, tickets)
- GitHub repository med README

### **Database-struktur:**

- users tabell: id, brukernavn, passord, rolle (bruker/drift)
- tickets tabell: id, bruker\_id, tittel, beskrivelse, status, behandler\_id
- Foreign key fra tickets til users

### **Innlogging:**

- Registreringsside (velg rolle: bruker eller drift-arbeider)
- Innloggingsside
- Utlogging
- Sjekk at bruker er innlogget før tilgang til sider

### **For vanlige brukere:**

- Side for å opprette ny ticket (tittel, beskrivelse)
- Side som viser oversikt over egne tickets
- Kan se status på egne tickets

### **For drift-arbeidere:**

- Side som viser alle tickets i systemet
- Kan "ta" en ticket (tildele seg selv som behandler)
- Kan endre status på tickets (åpen/under arbeid/lukket)

### **Teknisk:**

- Bruk av render\_template for alle sider
- POST for å sende data (registrering, innlogging, nye tickets)
- GET for å vise sider

### **GitHub:**

- README med hvordan man setter opp og kjører prosjektet
- README med tanker rundt hva du har lært

**Levering: GitHub-lenke til repository med prosjekt og README**