**Projektplan för Realtidsövervakning av Trafikflöden**

**Fas 1: Förberedelse och Planering**

**Definiera Projektmål:**

Skapa en dashboard för realtidsövervakning av trafikflöden, väg- och färjerelektioner och störningar i Sverige.

Möjlighet att följa aktuella händelser och analysera historiska data för att identifiera trender.

**Teknikval**

**Datainsamling**:   
Trafikverket API för att hämta realtidsdata om störningar, vägblockeringar, och färjetrafik.

**Databearbetning och Lagring**:

**Apache Kafka** för realtidsdatainsamling och hantering.

**PostgreSQL** eller liknande för lagring av historisk data.

**Visualisering**: Använd **Streamlit** eller **Power BI** för att skapa en interaktiv dashboard.

**Schemaläggning och Automatisering**: **Dagster** eller **Airflow** för att schemalägga datahämtning och uppdatera realtidsflödet.

**Fas 2: Datainsamling**

**Anslut till Trafikverkets API:**

Konfigurera API-nycklar och skapa en pipeline för att hämta data om trafikrestriktioner, störningar och färjetrafik.

Data att hämta inkluderar: vägarbeten, olyckor, färjetrafik, väglag, trafikstörningar, med tidsstämplar och geografiska koordinater.

**Dataöverföring till Lagring:**

Skapa en realtidsdatapipeline med t.ex. **Kafka** eller annan teknik för att skicka realtidsdata till databasen.

Konfigurera lagring av data i **PostgreSQL** eller en liknande databas för både realtids- och historisk data.

**Fas 3: Databearbetning och Strukturering**

**Datamodellering:**

Skapa en **datamodell** för att organisera och lagra information om trafikstörningar, vägnummer, geolokalisering, datum och typ av störning.

Skapa tabeller för **historiska data** och realtidsflöden.

**Datatransformering:**

Använd **dbt** för att skapa modeller som omvandlar rådata från API:et till mer användbara format.

Transformera data för att extrahera och bearbeta nyckelattribut som t.ex. påverkat vägnummer, region och tidsintervall för störningar.

**Lagring av Historisk Data:**

Bygg en rutin för att spara och arkivera data dagligen, veckovis eller månadsvis så att du kan analysera trender över tid.

**Fas 4: Skapa Dashboard**

**Dashboardlayout:**

Använd **Streamlit** för att skapa en webbaserad dashboard eller **Power BI** för en mer avancerad dashboard.

**Funktioner att inkludera:**

**Realtidskarta**: Visa alla pågående störningar och trafikbegränsningar med geolokalisering.

**Tidslinje**: Visualisera störningar över tid och analysera när störningarna inträffar mest frekvent.

**Filtrering**: Låta användaren filtrera data baserat på region, typ av störning (vägarbeten, olyckor etc.), eller tid.

**Realtidsuppdatering**

Implementera en lösning som uppdaterar dashboarden i realtid eller nära realtid, t.ex. med hjälp av **SocketIO** eller schemalagda uppdateringar via **Dagster**.

**Analyser av Historiska Data:**

Lägg till en modul som kan analysera historiska data för att visa långsiktiga trender, t.ex. vilka vägsträckor som ofta drabbas av störningar och när på året störningar är vanligast.

**Fas 5: Testning och Optimering**

**Testa Funktionaliteten**

Testa hela flödet: datainsamling, databearbetning och visualisering i dashboarden.

Kontrollera att data hämtas korrekt från API:et och att realtidsflödet fungerar som förväntat.

**Prestandaoptimering**

Säkerställ att realtidsuppdateringen av dashboarden fungerar utan fördröjningar. Optimera eventuell databearbetning och datalagring.

**Säkerhet och Skalbarhet**

Om systemet skalas upp, överväg säkerhetsaspekter som autentisering för användaråtkomst och dataskydd för API-anrop.

**Fas 6: Implementering och Slutrapport**

**Implementera och Distribuera**

Distribuera dashboarden till användare, antingen som en webbaserad lösning (via Streamlit) eller som en publicerad rapport i Power BI.

**Dokumentation och Rapportering**

Sammanställ en rapport som beskriver den tekniska lösningen, datamodellen, och de insikter som kan dras från realtids- och historiska data.

Analysera användbarheten och möjligheten att förutsäga framtida trafikproblem baserat på datan.

Denna plan ger en strukturerad väg från datainsamling till en färdig dashboard. Det viktigaste är att säkerställa att realtidsuppdateringar fungerar smidigt och att data är lätt att förstå och analysera i din visualisering.