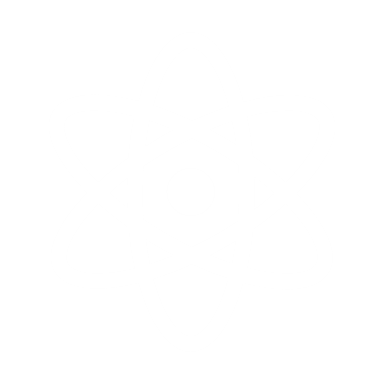
|  |  |
| --- | --- |
| **3-MAVZU** | **“Event-Driven” Voqealarga asoslangan arxitektura asosidagi tizim yaratish.** |

****

**AMALIY  
KO‘NIKMA**

**HAR BIR TALABA O‘ZINI VARIANTIGA MOS MISOLNI ISHLAB VA OLINGAN NATIJALARNI BAHOLASH UCHUN PDF FORMATGA O‘TQAZIB** [**https://elms.tuit.uz**](https://elms.tuit.uz) **SAYTIGA YUKLASHI LOZIM**

**HAR BIR TALABA O’ZI TANLAGAN IXTIYORIY MAVZUDAGI LOYIHASINI USHBU TOPSHIRIQLAR ASOSIDA BAJARISHI TALAB QILINADI.**

**TANLANGAN LOYHA MAVZULARI BIR XIL BO’LMASLIGI KERAK AGAR BIR XIL LOYHADA BIR HIL ISH YUKLANSA BALL QO‘YILMAYDI.**

**Maqsad**: Event-Driven arxitektura (EDA) tizimni voqealar asosida boshqarishni ta’minlaydi. Bu mashg‘ulot davomida siz voqealarga asoslangan tizimni yaratish uchun zarur bo‘lgan amaliy qadamlarni o‘rganasiz. Quyidagi topshiriqlar sizga Event-Driven tizimini yaratishda tajriba orttirishga yordam beradi.

**Topshiriq variantlari**

**1. Topshiriq: Tizim Talablarini Aniqlash**

Maqsad: Tizim uchun kerakli voqealarni va ularning ta’sirini tushunish.

Vazifa:

* Tizimda qanday voqealar yuzaga kelishini aniqlang (masalan, foydalanuvchi ro‘yxatdan o‘tishi, buyurtma berilishi, xabar yuborilishi va h.k.).
* Har bir voqeaga qanday harakatlar amalga oshirilishini belgilash (masalan, buyurtma berilganda, to‘lovni qayta ishlash, tasdiqlash xabarini yuborish).
* Tizimda voqealarni qanday kuzatib borish va saqlashni rejalashtiring.

**2. Topshiriq: Arxitektura Loyihasini Yaratish**

Maqsad: Event-Driven arxitektura tuzilishini loyihalash.

Vazifa:

* Event Producer: Voqealarni yaratadigan komponentlar (masalan, foydalanuvchi yoki tizim tomonidan yuzaga kelgan voqealar).
* Event Bus: Voqealarni tarqatish va tizim komponentlari orasida uzatish uchun vositani tanlang (Kafka, RabbitMQ, yoki o‘zingizning o‘rnatilgan tizimingiz).
* Event Consumer: Voqealarni qabul qilib, ularning qayta ishlanishini ta'minlaydigan komponentlar (masalan, foydalanuvchi ma’lumotlarini yangilash).
* Event Store: Voqealarni saqlash uchun tizim yaratish (masalan, voqealar tarixini saqlash uchun ma'lumotlar bazasi).
* Loyihani chizish va har bir komponentning vazifasini aniqlash.

**3. Topshiriq: Voqealarni Yaratish va Uzatish**

Maqsad: Voqealarni yaratish va ularni boshqa tizimlarga uzatishni o‘rganish.

Vazifa:

* Har bir voqea uchun Event Class yoki Event Message yarating (masalan, UserRegisteredEvent, OrderPlacedEvent).
* Event Producer yaratish va voqealarni Event Bus orqali uzatish.
* Event Consumer yaratib, voqealarni qabul qiling va ularning qayta ishlanishi uchun tegishli ishlarni amalga oshiring (masalan, foydalanuvchi ro‘yxatdan o‘tganida tasdiq xabarini yuborish).

**4. Topshiriq: Event-Driven Tizimni Sinovdan O‘tkazish**

Maqsad: Tizimning ishlashini va voqealarni qayta ishlashini tekshirish.

Vazifa:

* Unit Testlar yozing: Har bir voqeani yaratish va qayta ishlash mexanizmlarini tekshiring.
* Integration Testlar yozing: Event Bus va Event Consumer o‘rtasida integratsiya to‘g‘ri ishlashini tekshiring.
* Load Testlar: Ko‘p miqdordagi voqealarni yuborib, tizimning yuklamasini va ishlash tezligini tekshiring.

**5. Topshiriq: Event-Driven Tizimda Xatoliklarni Boshqarish**

Maqsad: Xatoliklarni aniqlash va tizimni muammosiz ishlashini ta'minlash.

Vazifa:

* Voqealarni qabul qilganda yuzaga kelishi mumkin bo‘lgan xatoliklarni (masalan, tizimga ulanishda xatolik) qanday boshqarish kerakligini rejalashtiring.
* Dead-letter Queue yaratish va xatolik yuzaga kelganda voqealarni bu joyga yo‘naltirish.
* Xatoliklar va muammolarni qayd etish va tizimni tiklash strategiyasini ishlab chiqish.

**6. Topshiriq: Tizimni Kengaytirish va Yangi Voqealar Qo‘shish**

Maqsad: Tizimni yangi voqealar va xizmatlar bilan kengaytirish.

Vazifa:

* Tizimda yangi voqealar qo‘shing (masalan, to‘lovni qayta ishlash, buyurtma tasdiqlash).
* Yangi voqealarni yaratish, uzatish va qayta ishlash jarayonlarini amalga oshiring.
* Kengaytirilgan tizimni sinovdan o‘tkazib, uning samaradorligini tekshiring.

**7. Topshiriq: Event-Driven Tizimni Optimallashtirish**

Maqsad: Tizimni yanada samarali va yuqori samaradorlikka ega qilish.

Vazifa:

* Voqealar uzatish jarayonlarini optimallashtirish (masalan, voqealar hajmini kamaytirish, xabarlarning siqilish texnikalarini qo‘llash).
* Tizimning yuksak ish samaradorligini ta’minlash uchun Event Processing va Event Consumption jarayonlarini yaxshilash.
* Tizimdagi voqealar soni oshgan sari uning ishlash tezligini ta'minlash uchun Event Streaming va Parallel Processing texnologiyalarini qo‘llang.

**8. Topshiriq: Event-Driven Tizimda Monitoring va Logging**

Maqsad: Tizimning ishlashini monitoring qilish va muammolarni aniqlash.

Vazifa:

* Tizimning ishlashini kuzatish uchun monitoring vositalarini o‘rnatish (masalan, Prometheus, Grafana).
* Tizimdagi har bir voqea va xatolikni loglash (masalan, ELK stack: Elasticsearch, Logstash, Kibana).
* Monitoring va loglar yordamida tizimning samaradorligini va xatoliklarni kuzatib borish.

**Xulosa:**

Bu topshiriqlar yordamida siz Event-Driven arxitekturasi asosida tizim yaratish bo‘yicha amaliy tajriba to‘playsiz. Har bir topshiriq tizim yaratishning muhim jihatlarini o‘rganishga yordam beradi va o‘zingizga kerakli ko‘nikmalarni shakllantiradi. Tizimni yaratish va uni sinovdan o‘tkazish jarayonida voqealar asosida tizimni boshqarishning samarali usullarini amaliyotda qo‘llashingiz mumkin.