Structura Lucrare Licența

1. Introducere
   1. Motivare
   2. Obiective
2. Arhitecturi soft
   1. Monolit
   2. Aplicatii disribuite?
   3. Microservicii
   4. 1. Ce sunt micro-serviciile? Avantajele micro serviciilor in comparație cu arhitectura monolitică.
      2. Scurta istorie a micro serviciilor. Primele persoane care au abordat arhitectura cu micro servicii.
      3. Aplicații din lumea reala care folosesc micro-serviciile.
      4. Datele in micro-servicii. Probleme cu partajarea datelor intre micro-servicii.
      5. Tipuri de comunicare a datelor intre micro-servicii. Comunicare asincrona versus sincrona.
      6. Arhitectura bazata pe micro-servicii. Exemple.
3. Tehnologiipentru dezvoltarea unei aplicații cu micro-servicii.
   1. Application Programming Interface (API)
      1. Ce este un API?
      2. Modern API. REST
      3. Rolul API-ului in aplicația cu arhitectura cu micro-servicii.
      4. Separarea serverului de client.
   2. Docker
      1. De ce se folosește Docker in arhitectura cu micro-servicii? Avantajele folosirii containerelor Docker.
      2. Ce este Docker? Cum rulează Docker pe computer?
      3. Ce este un container? Cum se rulează un container?
      4. Ce este o imagine? Cum se creează o imagine in Docker?
      5. Legătura intre micro-servicii si imagini. Cum se creează o imagine o unui serviciu ?
   3. Kubernetes
      1. Ce este Kubernetes? Fundamentele orchestrării unei colecții de micro-servicii folosind Kubernetes.
      2. Terminologii de Kubernetes importante.
      3. Obiecte create de către Kubernetes pentru a rula micro-serviciile. Pods. Deployments.
      4. Servicii pentru networking.
         1. Comunicarea intre micro-servicii. ClusterIP
         2. Comunicarea micro-serviciilor cu un client extern. NodePort. LoadBalancer.
      5. Instrument de gestiune a obiectelor si serviciilor Kubernetes. Skaffold
4. Studiu de caz . Aplicație bazata pe micro-servicii.
5. Descrierea generală a proiectului
   1. Analiza
   2. Proiectare
   3. Dezvoltare/Implementare
   4. Abordarea monolitica si abordarea cu micro-servicii . Comparații.
      1. Tehnologii alese pentru implementare. Metode de implementare.
   5. Backend
      1. Express.js.
         1. Ce este Express.js? Idei fundamentale.
         2. Cereri POST,PATCH,DELETE,GET.
         3. Typescript. Idei fundamentale.
      2. Managementul bazei de date. MongoDB
         1. Ce este MongoDB ? Crearea bazei de date.
         2. Conectarea la baza de date.
      3. O implementare a unui sistem de mesaje. NATS.
         1. Conectarea la NATS in Express.js.
         2. Managementul sistemului de mesaje.
      4. Metode principale de implementare
         1. Strategii de autentificare a utilizatorului.
         2. Partajarea si reutilizarea codului intre micro-servicii
         3. Create-Read-Update-Delete
   6. Frontend
      1. React
         1. Ce este React? Idei fundamentale
         2. Implementare server-side
6. Testare. Testarea izolata a micro-serviciilor