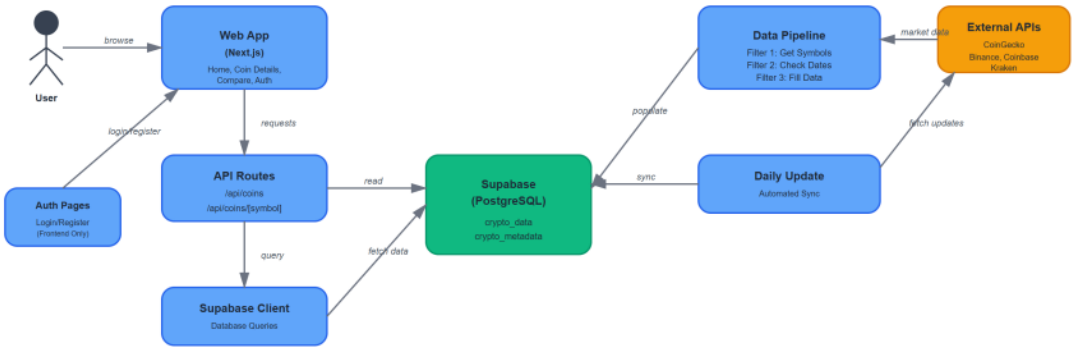
Домашна работа 2

1. Концептуална архитектура



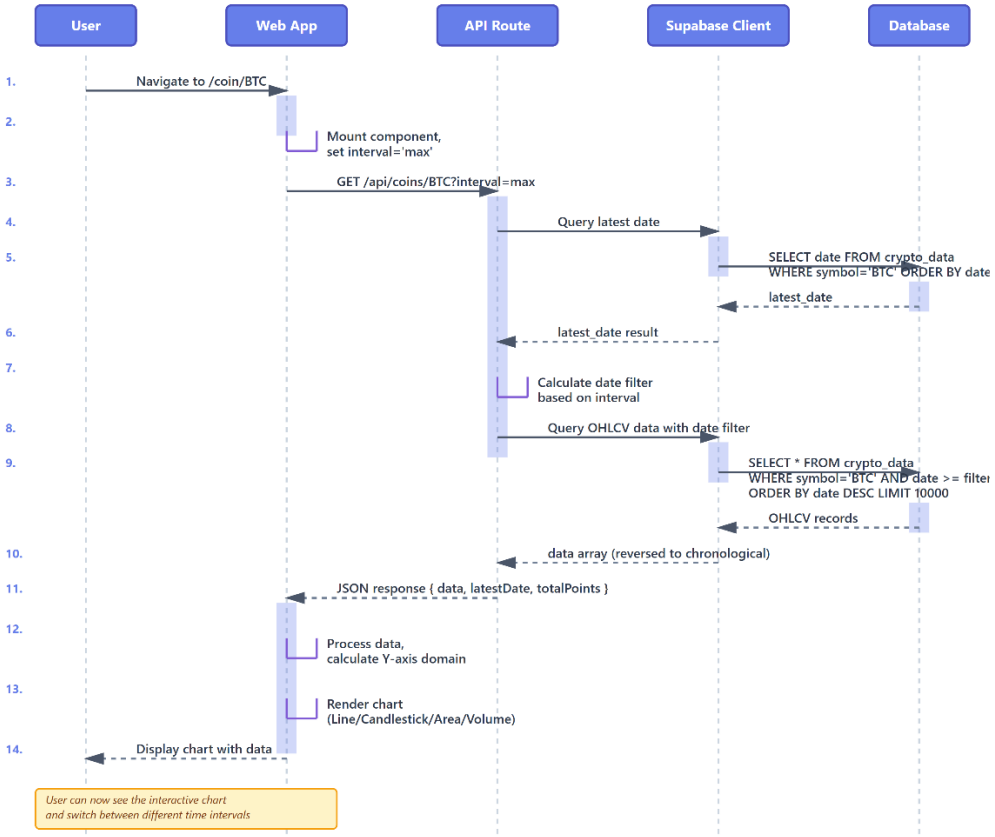
Во овој дијаграм се среќаваат неколку важни концепти од концептуална архитектура, односно начинот на кој се организира системот на ниво на логика и улоги (без конкретна имплементација). Односно некои од тие концпети се: **слоестата архитектура**, **pipeline pattern**, **client-server pattern**, **ETL процесот**, **поделба на обврски**(SoC), **поделба на податоците**(во crypto_data и crypto_metadata)…

Корисник → Web App → API → База

Компонентите во системот се поделени во неколку слоеви:

- **Кориснички слој (frontend):**
  • User – крајниот корисник што го користи системот
  • Web App (Next.js) – страниците за прикажување:
    - Home
    - Coin Details
    - Compare
    - Auth (пријава/регистрација)
  • Auth Pages – Login/Register (Frontend only)
- **API / Backend слој:**
  • API Routes
  • Supabase Client
- **Database слој:**
  • Supabase (PostgreSQL)
- **Automation / Background процеси:**
  • Data Pipeline:
    - Filter 1 → Get symbols
    - Filter 2 → Check dates
    - Filter 3 → Fill data → во база
- **Надворешни извори:**
  • Binance

2. Извршна архитектура



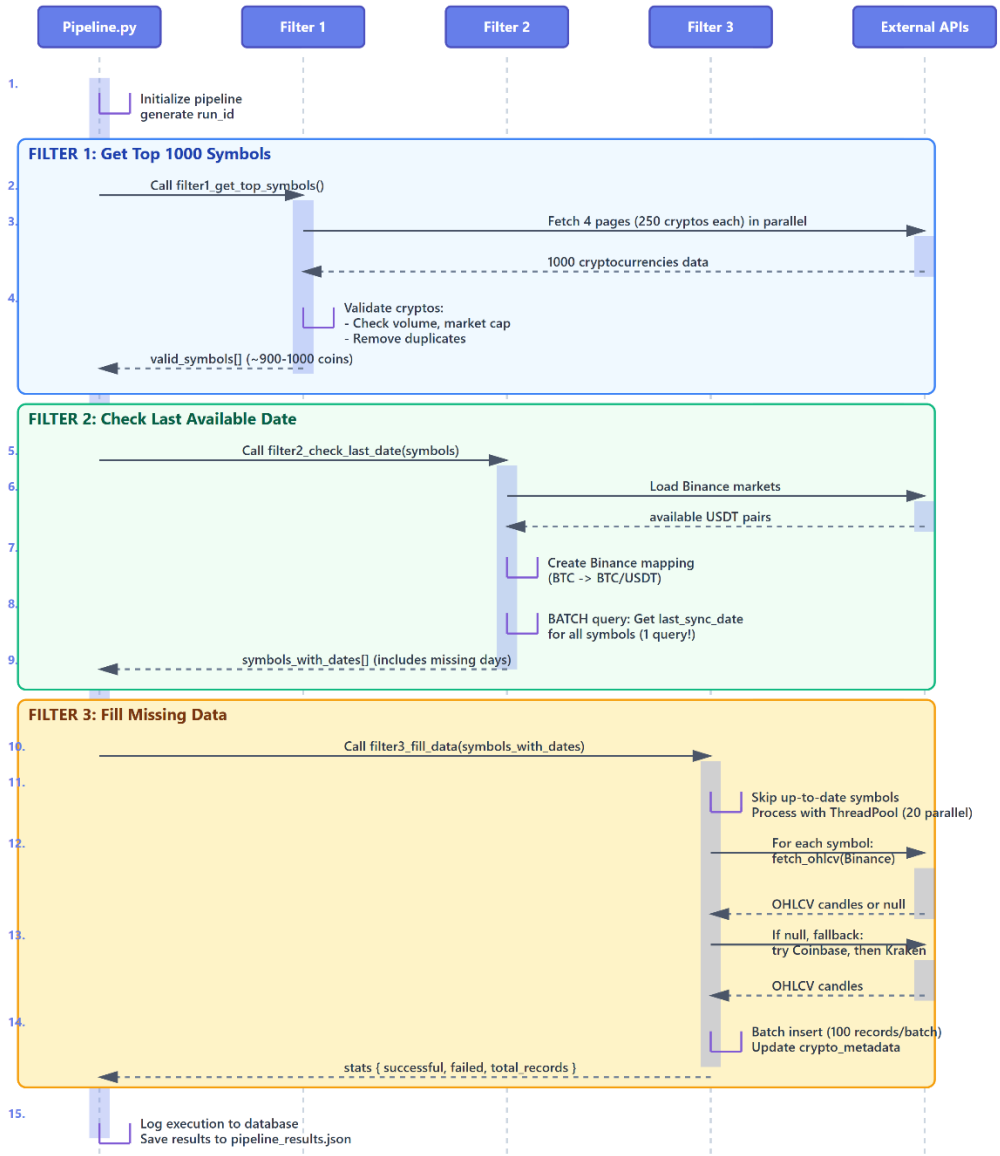
Овој извршен дијаграм прикажува како тече процесот кога корисник сака да види график за криптовалута. Корисникот преку Web App иницира барање, кое се праќа до API Route, каде се пресметува временски филтер и се повикува Supabase Client за да се направат базни query-и. Базата враќа OHLCV податоци, кои API ги форматира во JSON и ги враќа назад до фронтендот. Web App ги процесира податоците и пресметува визуелен опсег, по што се рендерира графикот. На крај, корисникот добива интерактивен приказ и може да менува временски интервали.

Компоненти кои се јавуваат во извршниот слој се:

- Корисник - го иницира процесот
- Web App - го прикажува графикот
- API Route – Backend дел кој процесира барања
- Supabase Client - го врши query-то кон базата
- Database - Чува податоци (crypto_data)

Концепти од извршна архитектура што се имплементирани во дијаграмот се:

- Sequence (редослед на повици) – чекор по чекор се покажува како патува барањето
- Lifeline (вертикални линии) – покажуваат траење на активност на секоја компонента
- Return messages – резултати од база, JSON од backend
- Data Flow – податоци => API => фронтенд => график



Дијаграмот го прикажува извршувањето на Data Pipeline за нашата Crypto Tracker апликација. Користи Pipe & Filter архитектонски модел, каде што секој чекор (филтер) добива податоци од претходниот, ги обработува и ги предава понатаму. Целта е да се обработат топ 1000 криптовалути, да се провери последниот достапен датум со податоци и да се дополни историја со OHLCV податоци.

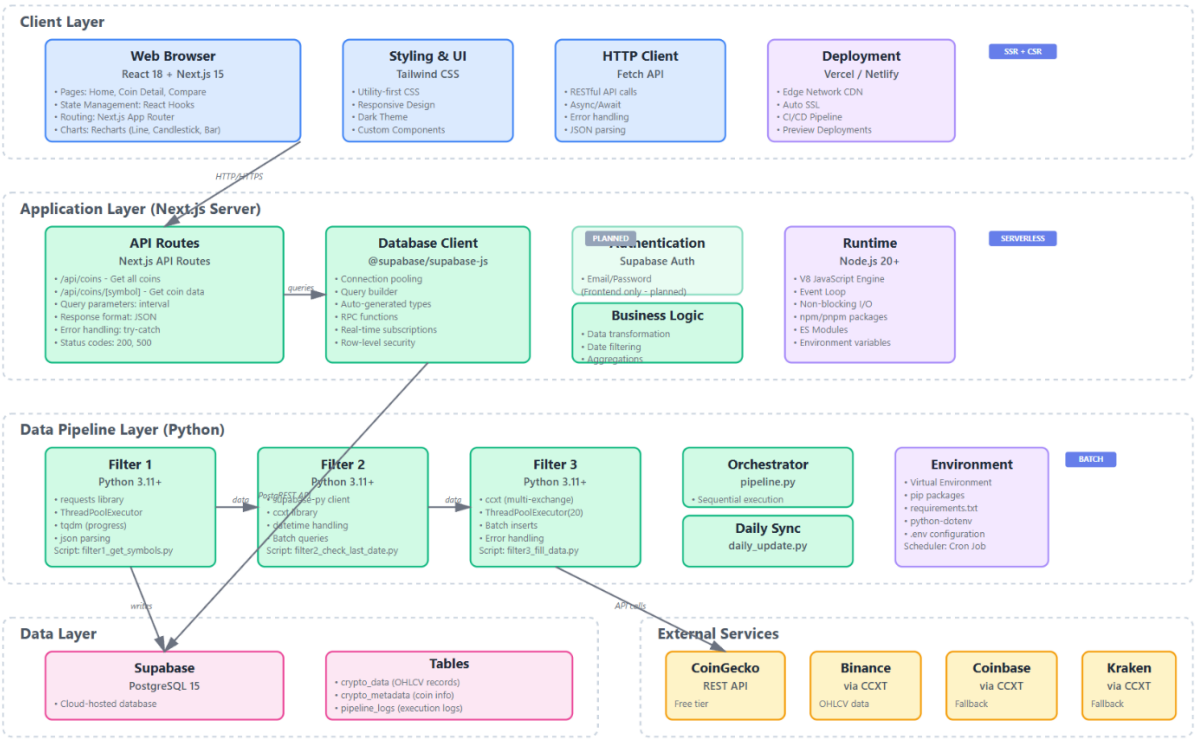
Компоненти во дијаграмот:

- Pipeline.py (иницијализирачка компонента)
- Филтер 1 – Get Top 1000 Symbols
- Филтер 2 – Check Last Available Date
- Филтер 3 – Fill Missing Data

Концепти од извршна архитектура што се присутни во дијаграмот:

- Pipe & Filter Pattern
- Паралелно процесирање
- Batch операции
- Fault-tolerant fallback...

3. Имплементациска архитектура



Дијаграмот покрива целосна имплементација на Crypto Tracker: од фронтенд (кориснички интерфејс), преку Next.js API рути и бизнис-логика, до Python data-pipeline кој извлекува, трансформира и полни историски OHLCV податоци во Supabase. Надворешните извори се користат за добивање на симболи. Vercel за деплој.

Компоненти во дијаграмот:

- Client layer (Web browser / UI)
- Application layer (Next.js API / Serverless)
- Data Pipeline layer (Python scripts)
- Data layer (Supabase / PostgreSQL)
- External services

Концепти од имплементациска архитектура што се присутни во дијаграмот:

- Слоевита архитектура
- Serverless архитектура
- Микросервиси: модуларизација по функции - API рути, pipeline филтри, оркестратор, секој модул има своја независна логика
- Data Pipeline Architecture
- Pipe & Filter Pattern
- Паралелно процесирање
- Batch процесирање: ова е концепт на batch-oriented systems за зголемување перформанси
- External API Integration Pattern: системот интегрира со надворешниот извор – Binance
- Edge Computing & CDN Caching
- Database-centric Architecture…