

Домашна работа 2

1. Архитектура

1.1 Вовед

Нашиот тим работи на проектот **InvestMK** кој има за цел да го олесни процесот на изработка на апликацијата и да ја подобри нејзината оптималност. Во текот на развојот, тимот ги следи зададените чекори од курсот, како и интерно дефинираниот редослед за создавање функционална и стабилна веб-апликација. Во првиот дел од архитектурата, се фокусиравме на клучните концепти преку изработка на концептуалната архитектура, во која ги дефиниравме основните компоненти на апликацијата и нивната интеракција. Потоа, создадовме извршна архитектура со акцент на перформансите и употребливоста, каде што го прикажавме текот на податоците и како различните услуги меѓусебно комуницираат. Во следниот чекор, изработивме имплементациски поглед, каде што детално ги опишавме технологиите што ги користиме за реализација на проектот. Притоа направивме табела со предности и слабости на секоја технологија, што ни помогна во изборот на најприкладните алатки.

1.2 Концептуалната архитектура

Со цел да ја дефинираме концептуалната архитектура, ќе извршиме категоризација на клучните концепти според функционалностите што ги наведовме за нашата апликација **InvestMK**. Оваа веб-апликација ќе овозможи:

1. Регистрација преку email адреса.
2. Приказ на дневни цени на акции.
3. Историски трендови за акциите.
4. Филтрирање на податоци според издавач.
5. Филтрирање на податоци според датум.
6. Интерактивна мапа за лоцирање на компаниите.
7. Приказ на детали за компаниите (опис, сектор, податоци за контакт).
8. Приказ на календар со важни финансиски датуми и настани.
9. Можност за сортирање на податоци во табели.
10. Систем за препораки и анализи врз основа на историјата на тргување.
11. Систем за корисничка поддршка (FAQ и можност за контакт).
12. Темна и светла тема за подобро корисничко искуство

Со оваа категоризација, ги истакнуваме функционалностите што ќе бидат централен дел од нашата апликација. Секој од овие концепти ќе биде обработен во архитектурата на извршниот и имплементацискиот дел за да обезбедиме соодветна структура, стабилност и перформанси на апликацијата.

1.3 Извршна архитектура

Оваа архитектура го прикажува текот на извршување на различни компоненти и нивната интеракција.

Апликацијата ќе биде развиена како дистрибуирана архитектура со следните карактеристики:

- Цевки и филтри: Податоците ќе поминуваат низ слоеви за обработка, валидација и трансформација во Backend пред да се прикажат на Frontend.
- Словитата архитектура: Backend е структуриран во три слоеви:
 - Контролери (Controller): Обезбедуваат API и ги обработуваат барањата од Frontend.
 - Сервиси (Service): Бизнес логиката се имплементира овде.
 - Репозиториуми (Repository): Директно комуницираат со базата на податоци.
- Микросервиси: Иако овој прототип е монолитен, во иднината компонентите на Backend можат да се подели како независни микросервиси.
- Контејнеризација: Docker ќе се користи за пакување и распоредување на апликацијата.

1.4 Имплементациска архитектура

Оваа архитектура го дефинира начинот на имплементација и алатките што ќе се Frontend:

- Реализиран со React, користејќи react-router-dom за рутирање помеѓу екрани.
- Користење на fetch() за интеграција со Backend API.

Backend:

- Контролери:
 - Обезбедуваат REST API ендпоинти со Spring Web (/api/all за добивање сите записи, /api/{id} за конкретен запис).
- Сервиси:
 - Имплементирана бизнес логика за обработка на податоци.
- Репозиториуми:
 - Користат Spring Data JPA за интеракција со H2 базата.

Database:

- H2 база конфигурирана со автоматско ажурирање на табелите (spring.jpa.hibernate.ddl-auto=update).
- Сите структури за податоци се мапираат преку JPA ентитети.

Технологии:

- Frontend: React
- Backend: Java 17, Spring Boot 3.3.5
- Database: H2
- Тестирање: JUnit 5 за Backend, React Testing Library за Frontend.

- Контејнеризација: Docker за создавање на одделни контејнери за Backend и Frontend.