

Дизајн и архитектура на софтвер

Домашна работа 1

Целта на домашните е да се имплементира веб апликација која користи бази на податоци, користејќи различни стилови на архитектура на софтвер, при што треба да се применат концептите од предавањата.

Апликацијата треба да се фокусира на анализа на **Македонската берза** и да вклучува **историски дневни податоци** од веб-страницата на Македонската берза за **сите достапни издавачи** (компаниии или институции). Податоците што ќе се обработуваат треба да ги покриваат најмалку **последните 10 години на дневно ниво**.

Во првата домашна работа ќе се фокусирате на обработка на податоци користејќи ја **Pipe and Filter архитектурата**. Ќе треба да го **автоматизирате процесот** на преземање и трансформирање на податоците од Македонската берза за секој издавач, осигурувајќи дека се задржуваат само потребните информации и дека податоците се правилно форматирани за понатамошна анализа. Потребно е да ги изработите следните задачи:

1. Формирајте тим (**најмногу 3 членови**).
2. Напишете **краток опис на проектот** (долг околу половина страница).
3. Наведете **спецификација на функционалните и нефункционалните барања**. Притоа наведете кои од барањата се функционални, а кои нефункционални. Вклучете **кориснички сценарија, персони и описен наратив** за да ги илустрирате овие барања.
4. Преземете ги податоците што ќе ги користите и пополнете ја вашата база на податоци¹. Притоа треба да ја користите pipe-and-filter архитектурата за проток на податоци. За ова потребно е да дефинирате неколку едноставни, независни функции (филтри) за трансформација на податоците, при што излезот на секој филтер ќе може да се користи за влез на некој друг филтер (филтрите треба да бидат поврзани во цевка). Целта на овие филтри и целата цевка треба да биде да ги трансформираат податоците од необработен формат во кој сте ги презеле, во формат соодветен за внесување во базата на податоци.

Потребните трансформации треба да ги вклучуваат следните чекори:

- (a) **Филтер 1: Автоматски преземете ги сите издавачи наведени на веб-страницата на Македонската берза**
 - Отворете ја страницата на Македонската берза за историски податоци и **автоматски извлечете ја листата на сите издавачи** од паѓачкото мени (со исклучок на обврзниците или сите кодови што содржат броеви).
 - Овој филтер програмски ќе ги собере кодовите за секој наведен издавач, **без рачна интервенција**, осигурувајќи дека се преземени сите валидни издавачи.
- (b) **Филтер 2: Проверете го последниот датум на достапни податоци**
 - За секој код на издавач, преземен во првиот филтер, проверете ја вашата база на податоци (или структурирана датотека) за да видите **до кој датум веќе се земени податоците за тој издавач**.
 - Ако нема постоечки податоци, земете податоци за најмалку последните 10 години.
 - Доколку постојат податоци, идентификувајте го последниот зачуван датум и пренесете ја оваа информација на следниот филтер.

¹ Доколку немате доволни познавања од бази на податоци, за чување на податоците можете да користите и некоја структурирана датотека (csv, Excel, JSON) или друг вид на нерелационски начин за чување на податоци.

(с) **Филтер 3: Пополнете ги податоците што недостасуваат**

- За секој издавач, користете ги **кодот и последниот достапен датум (од вториот филтер)** за да ги преземете сите податоци што недостасуваат до сегашниот датум.
- Осигурајте се дека сите нови податоци се правилно додадени или комбинирани со постоечките податоци во базата на податоци (или структурирана датотека).
- Погрижете се сите датуми да се правилно форматирани во конзистентен формат (Совет: користете ја англиската верзија на податоците за да избегнете недоследности во форматирањето на датумите).
- Цените треба да се форматираат со соодветни разделувачи (запирка за илјадници и точка за децимали, како 21,600.00).

На крајот од овој процес, апликацијата треба да ги складира целосните и ажурирани податоци за берзата во базата на податоци, спојувајќи ги со претходните податоци и осигурувајќи дека целото форматирање е точно.

Можете да креирате повеќе цевки ако користите повеќе од еден извор на податоци. Ако користите повеќе цевки, треба да ги користите истите филтри за иста трансформација во двете цевки наместо да создавате нови.

5. Предизвик за максимални поени

- Имплементирајте тајмер за да измерите колку време е потребно за вашата апликација целосно да ја наполни празната база на податоци со податоците за берзата. Оптимизирајте го овој процес за максимална брзина и ефикасност.
- Резултатите на останатите тимови ќе бидат скалирани врз основа на најдоброто постигнато време. Побрзите и поефикасни решенија ќе добијат повисоки поени.

Слободно можете да го изберете програмскиот јазик, базата на податоци и оперативниот систем што ќе го користите. Python се препорачува за следните домашни задачи поради достапноста на библиотеки погодни за трансформација на податоци и финансиски пресметки. **Проектите се предаваат исклучиво преку GitHub (јавен репозиториум). Проектите предадени преку e-mail или прикачени како .zip на GitHub нема да бидат прифатени.**

Вашиот **јавен GitHub репозиториум** треба да ги содржи документите со описот на проектот и спецификацијата на барањата, кодот за трансформацијата и вчитувањето на податоците и почетните податоци кои ги користите. Ако изберете да користите структурирана датотека за складирање, поставите ги и финалните обработени податоци. Сите датотеки треба да се наоѓаат во папка наречена Домашна 1.

Рокот за изработка на првата домашна работа е до **10 ноември 2024**. Имајте во предвид дека **овој рок е конечен и нема да биде продолжен**.