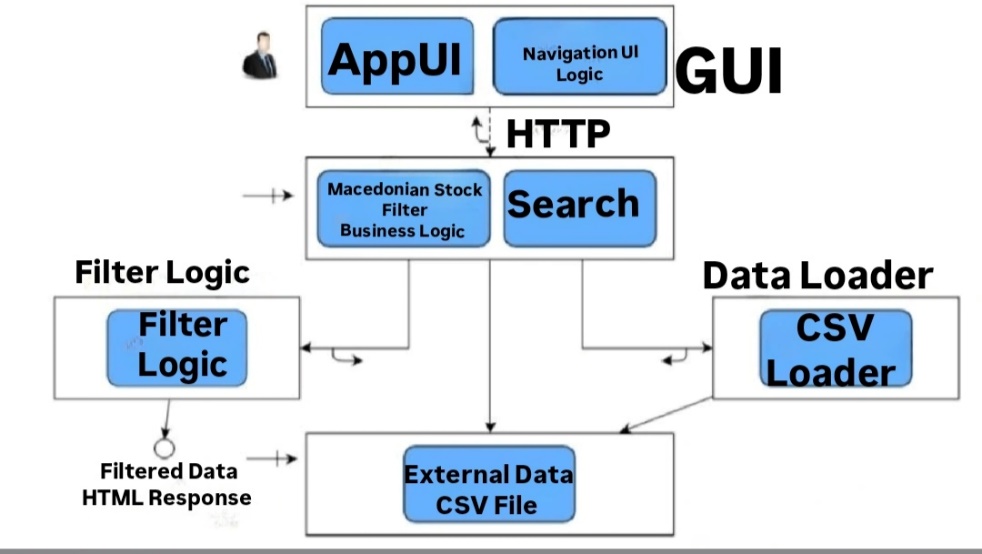
**Архитектурен дизајн**

1. **Концептуална архитектура**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Систем** | **Апстрактни концепти** | **Податоци** | **Засегнати страни** | **Функционалности** |
| **Flask веб апликација** | Корисник, Податоци | Податоци за акции, CSV датотека | Администратор,Корисник | Приказ на податоци, филтрирање на податоци |
| **Кориснички интерфејс (UI)** | HTML форма, Табела | Податоци за акции, филтри | Корисник | Внесување податоци (stockCode, dateFrom, dateTo), Приказ на резултати |
| **Податоци за акции** | CSV file | Податоци од акции (месеци/години) |  | Прочитај CSV, филтрирај податоци |

Ова е погледот на апликацијата на високо ниво и опишува како различните делови работат заедно.

* **Компоненти:**
  + **Кориснички интерфејс (UI):** HTML страници, CSS стилови и JavaScript за интерактивност.
  + **Апликациски слој (Backend):** Flask сервер кој обработува барања, извршува логика за филтрирање и испраќа податоци назад до UI.
  + **Датотечен слој:** CSV фајлот mse\_historical\_data.csv, кој служи како статичен извор на податоци.
* **Подархитектурни компоненти:**
  + **Слоевита архитектура:** UI е одделен од апликациската логика (Flask) и податочниот слој (CSV).
  + **Цевки и филтри:** Податоците поминуваат низ низи за трансформација, како филтрирање по датуми и код.



*Слика 1: Концептуална архитектура*

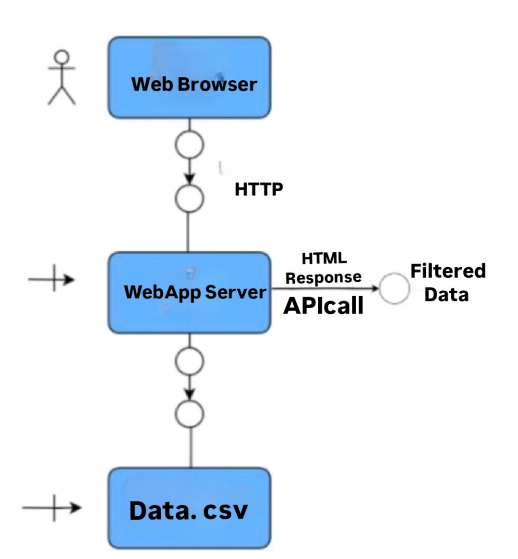
A diagram of a flowchart

Description automatically generated

*Слика 2: Секвенцен дијаграм*

**2. Извршна архитектура**

Ова е погледот кој ги опишува активните компоненти на системот во текот на неговото извршување.

* **Корисничко барање:**
  + Корисникот внесува податоци во веб-формата (код на издавач, датуми).
  + Клиентот (браузерот) испраќа GET или POST барање до Flask серверот.
* **Обработка на барањето:**
  + Flask серверот го прима барањето, чита податоци од CSV фајлот, ги филтрира и ги враќа резултатите како HTML страница.
* **Резултати:**
  + Филтрираните податоци се прикажуваат како табела на веб страната.
* **Дистрибуирана архитектура:**
  + Flask серверот може да се распредели преку контејнери (на пример, со Docker) за да биде лесно достапен.

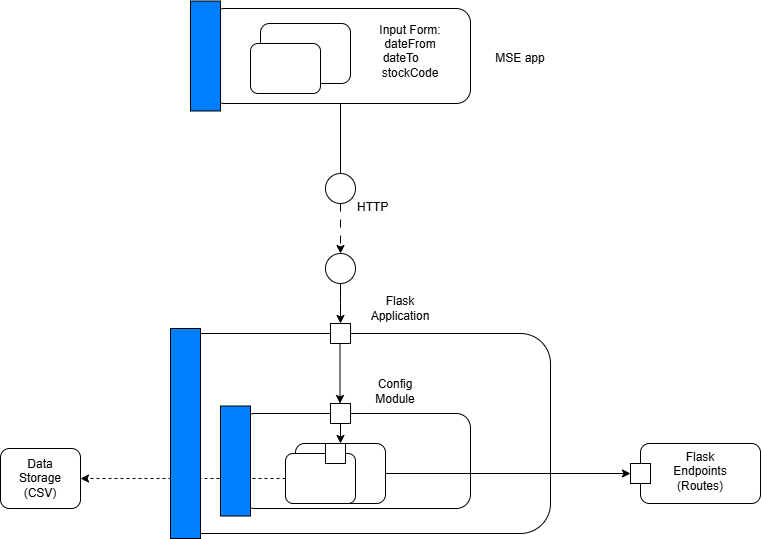
*Слика 3: Извршна архитектура*

**3. Имплементациона архитектура**

Ова е погледот кој ги покажува техничките детали и начините како е имплементирана апликацијата.

**Карактеристики на архитектурниот стил:**

1. **Слоевита веб архитектура:**
   * **UI слој:** HTML и CSS фајлови.
   * **Бизнис логика:** Python код во Flask кој филтрира податоци.
   * **Датотечен слој:** mse\_historical\_data.csv како статичен фајл.
2. **Цевки и филтри:**
   * Податоците од CSV фајлот се обработуваат со **филтрирање** (код на издавач, датум од, датум до) и се прикажуваат како обработен излез.
3. **Дистрибуирана архитектура со микросервиси:**
   * Во иднина, ако нашата апликација треба да се скалира, секоја функционалност (на пр., филтрирање или генерирање на табели) може да се претвори во посебен микросервис.
4. **Контејнеризација:**
   * Користење на Docker за контејнеризирање на апликацијата, што ќе овозможи полесно распоредување и извршување во различни околини.



*Слика 4: Имплементациска архитектура*