

## Едноставен прикажувач на производи и скенер на баркодови

### Преглед:

Замислете сценарио каде курир за испорака треба да собере производи од различни продавници. Вашата задача е да напишете модул за апликација што ќе прикажува информации за производите, генерира баркодови и наоѓа најблиска продавница за секој производ. Генерирањето на баркод се базира врз конвертирањето на идентификациониот број на производот во бинарен облик и визуелно прикажување со црни и бели вертикални ленти.

### Барања:

#### 1. Добивање на информации за производ и продавница:

- Информациите за сите производи се зачувани во датотеката **scans.json**.
- Информациите за сите продавници се зачувани во датотеката **shops.json**.

#### 2. Информации за производ:

- Прикажување на детали за секој производ, вклучувајќи име, идентификационен број, цена, количина и големина.
- Прикажувањето на производите е во обратен редослед, т.е. прв се прикажува последниот производ од низата со производи, па претпоследниот итн..

#### 3. Генерирање на баркод:

- Конвертирање на идентификациониот број на производ во бинарен облик.
- Генерирање на баркод претстава каде што секој бинарен знак се прикажува како вертикална лента (7px широка).
- Користење на црна за '0' и бела за '1'. Висината на лентите не е специфицирана.

#### 4. Процес на скиенирање:

- Симулирајте процесот на скенирање со позиционирање со покажувачот над баркодот.
- Процесот на скенирање трае 1.3 секунди.
- Ако покажувачот остане над баркодот во текот на целиот период, скенирањето е успешно. По успешно скенирање, во периодот додека се чека вчитувањето на следниот производ, се оневозможува скенирање на веќе успешно скенираниот баркод.
- Ефектот на скенирање се визуелизира со намалување на непроѕирноста (opacity) во текот на 1.3 секунди.

- Ако покажувачот излезе од областа на баркодот, скенирањето се откажува.

#### 5. Достапност во продавница:

- Проверка на достапноста на предметот во различни продавници.
- Предметот е достапен ако продавницата го поседува и има количина поголема или еднаква на потребната количина.
- Ако повеќе продавници го имаат истиот производ, прикажете ги информациите за најблиската продавница според растојанието.
- **Забелешка: Проверката на достапноста во продавниците се извршува САМО по успешно скенирање.**

#### 6. Слика на предмети:

- Секој производ има придружена слика во папката **images**.
- Името на сликата се конструира на следниот начин:  
"{име на производ}- {последните 5 цифри од ид-то}.jpg".
- Пример:  
Производ: Retro Red Dress, со ID: 123456789.  
Името на сликата е: dress-56789.jpg

#### 7. Пораки:

- По успешно скенирање, прикажете информации за најблиската продавница каде што предметот е достапен.
  - Форматот на пораката треба да биде: "Име на продавница, Улица на продавница, Број на улица, Skorje".
- Ако предметот не е достапен во ниту една продавница, прикажете порака:  
"The item is NOT available anywhere"

#### 8. Автоматско прикажување на следен производ:

- По успешно скенирање и прикажување на информации за продавницата, почекајте 3 секунди.
- Автоматски преминете на следниот производ и повторете го процесот.

#### 9. Циклично изминување на производи:

- Доколку сите производи од **scans.json** датотеката се процесираат, процесот започнува од почеток, т.е. циклична имплементација.

#### Насоки за имплементација:

- Во прилог ви е даден почетниот HTML и CSS. Потребно е да ја напишете само JavaScript имплементацијата.

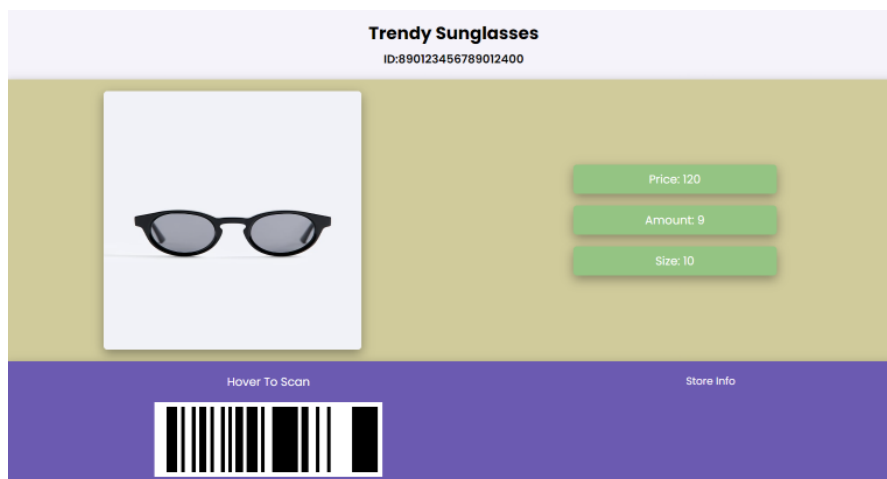
#### Примерен Сценарио:

1. Добивање на информации за производи и продавници од **scans.json** и **shops.json**.

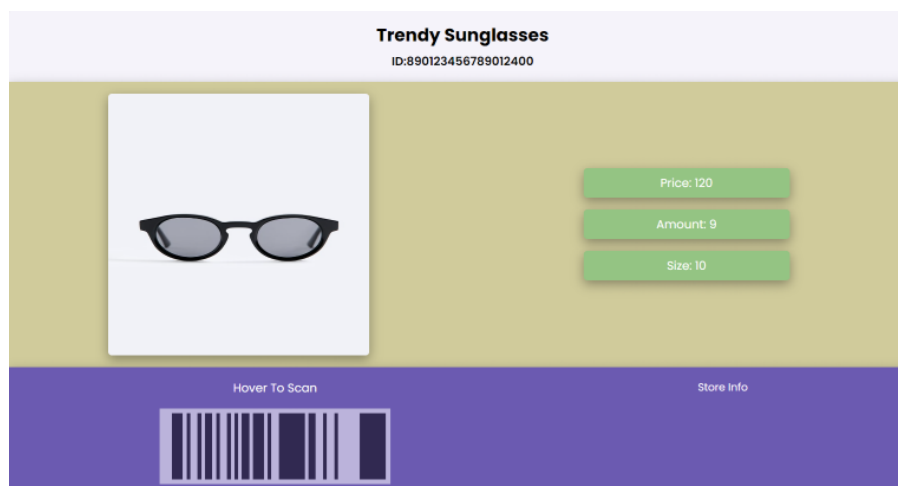
2. Прикажување на детали за Производ 1, генерирање и прикажување на баркод, и прикажување на слика на предметот.
3. Позиционирање над баркодот за симулирање на скенирање (1.3 секунди).
4. Ако скенирањето е успешно, наоѓање на најблиската продавница со производот, прикажување на информации за продавницата.
  - Пример: "Продавница ABC, Главна Улица, 123, Скопје."
5. После 3 секунди, автоматски преминување на следниот производ и повторување на процесот.

Имплементирајте ја апликацијата според вашето разбирање и стил, обезбедувајќи дека податоците се добиваат од JSON датотеките.

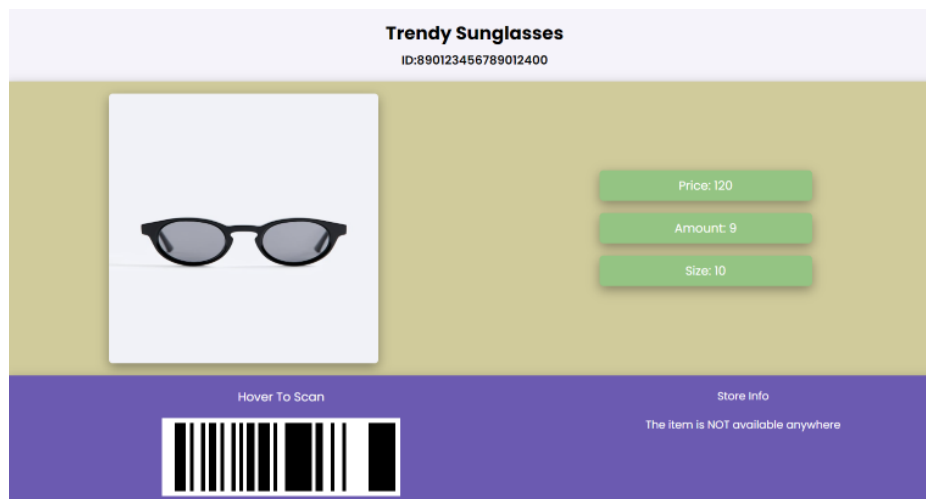
Пример слики:



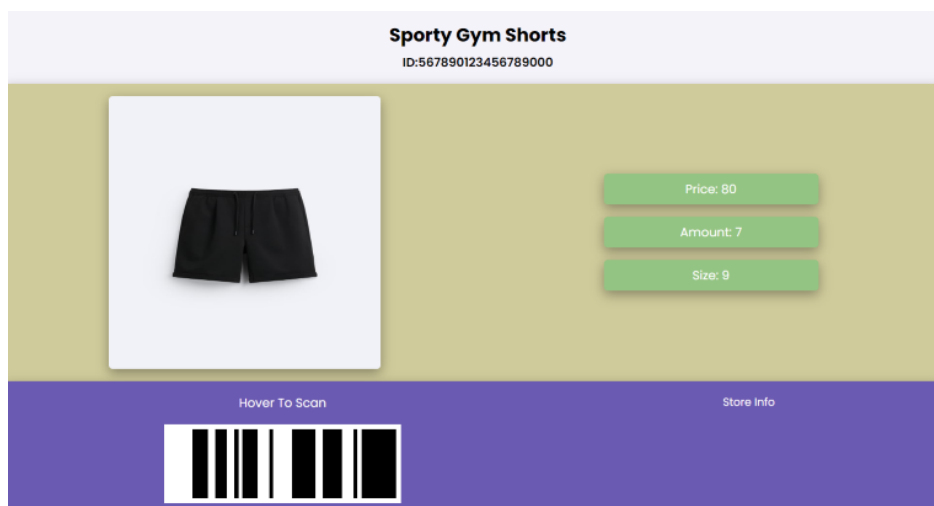
Слика 1. Изглед на успешно прикажани информации за производ и успешно изгенериран баркод.



Слика 2. Избледување на баркодот при негово скенирање.



Слика 3. Успешно скниран баркод и прикажување на информации за неговата достапност.



Слика 4. После 3 секунди, успешно вчитан следен производ.