

# Спецификација за проект

## Receipts Reading Software

### Проект: Receipts Reading Software | Full Stack Academy

За потребите на проектот **Receipts Reading Software** треба да се изработи веб решение за валидација на фискални сметки кои се користат во наградна игра.

### ОПИС НА ПРОЕКТОТ

#### 1. КОРИСНИЧКИ ПАНЕЛ

Корисничкиот панел претставува страница на која доаѓаат крајните корисници. Може да ја пристапи било кој, не е потребна претходна автентикација. Потребно е да има само едно копче на кое при клик се појавува форма во попап, каде клиентот треба да внесе email адреса и да прикачи слика. Кога ќе се направи submit на оваа форма, се појавува ладер. Во позадина, сликата и email-от се прикачуваат во самиот проект. Потребно е сликата да се зачува со рандом име во фолдер images/ и името на сликата и емаилот на корисникот да се зачуваат во база. После овој чекор се отпочнува процесот опишан подолу во делот **АПИ**. Тој процес може да заврши со 2 можни исходи:

##### 1. прикачената слика е во ред,

- На корисникот му се покажува порака: *Успешно аплициравте за наградната игра. Ви благодариме за учеството, дополнителни информации ќе добиете на email.*

##### 2. прикачената слика НЕ е во ред

- Сликата треба да се избрише од фолдерот images/ исто како и записот во база за конкретната слика и емаил адреса.
- корисникот добива порака да се обиде одново да ја прикачи сликата и дека му остануваат уште 2 обиди. Доколку клиентот не успее да прикачи слика и по три обиди, да му се покаже дека има забрана од 24 часа за повторен обид (во localStorage да се запише инфо дека е блокиран корисникот). Ова ќе значи дека секако тој ќе може од инкогнито таб или од друг пребарувач да продолжи да пробува, ама тој проблем не го решаваме во моментот.

## 2. АПИ ПОВИК

Ова е меѓу-чекор што се случува во тек на барањето претходно опишано. Откога сликата ќе се запише во фолдер `images`, истата ќе биде достапна на саммиот сиситем. Пример, ако проектот се хостира на страна `jager.mk`, тогаш сликата би била достапна на `jager.mk/images/nekoeime.jpg`. За работење локално, ова урл би било `localhost:8000/images/nekoeime.jpg`

Оттука, се прави апи повик до апи кое што ќе биде изработени од студентите на Data Science академијата. До тогаш, ова апи ќе го симулираме на следниот линк: <https://jager.brainster.xyz/> Комплетна документација за апито има на самиот линк.

### Како работи ова апи?

Апи-то очекува задолжителен GET параметар `img`. Ако го има, симулира проверка на тоа дали сликата што е во GET параметарот (сликата што ја праќа корисникот) е фискална сметка или не. Можни одговори од апи-то се:

- **НЕ Е СМЕТКА**

```
{  
    img_status: 0,  
}
```

- **Е СМЕТКА**

```
{  
    img_status: 1,  
    text: 'улица таа и таа, датум тој и тој, сите продукти цени итн,  
    буквално се што препознал алгоритмот на data science како букви'  
}
```

- **МОЖЕБИ Е СМЕТКА**

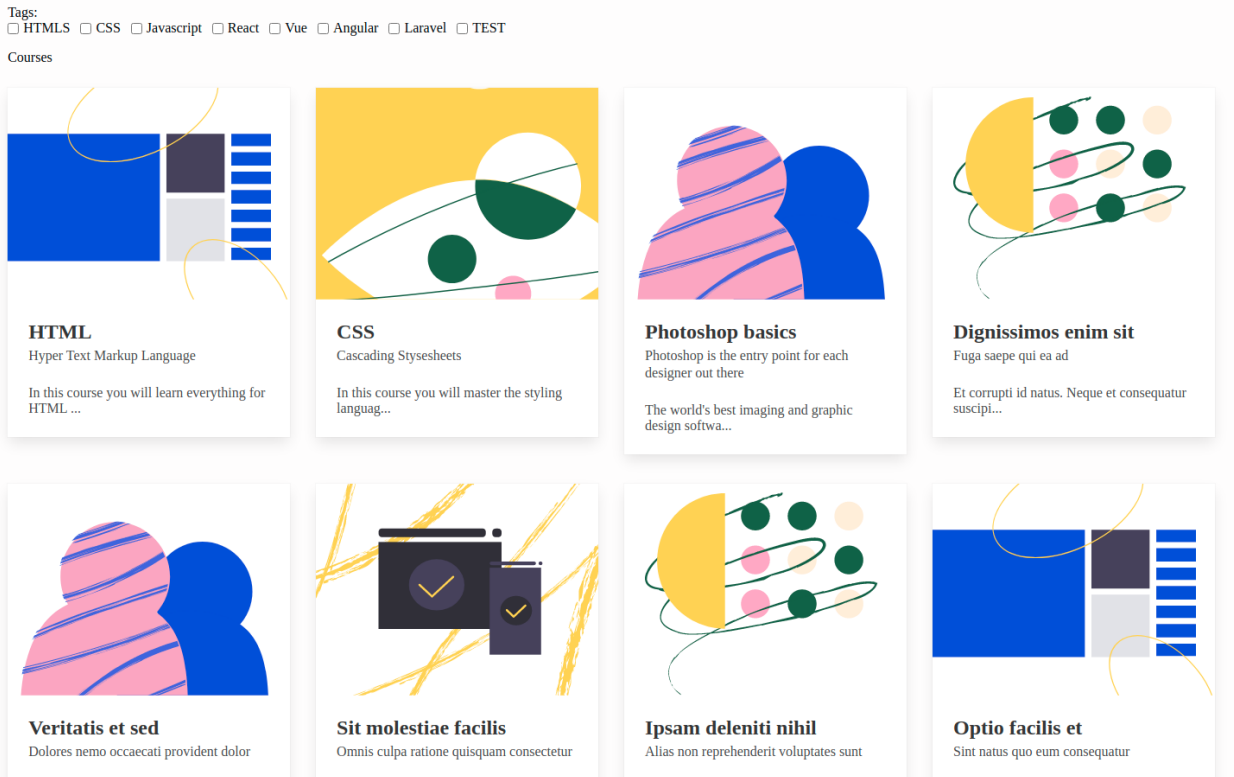
```
{  
    img_status: 2,  
    text: 'улица таа и таа, датум тој и тој, сите продукти цени итн,  
    буквално се што препознал алгоритмот на data science како букви'  
}
```

Ако статусот е 0, не се работи за фискална сметка. Сликата треба да се избрише од проектот, записот од база исто така да се избрише и корисникот да се извести да се обиде повторно.

Статусите 1 и 2 значат дека сликата е во ред. Статус 1 значи дека со сигурност се работи за фискална сметка. Статус 2 значи дека е голема веројатност да е фискална сметка (изгужвани сметки, различен формат итн.) Заедно со еден од овие два статуси, апи-то ќе го врати и целиот текст што успеало да го извади од самата слика (`image to text conversion`). Тој текст, заедно со статусот треба да го запишеме во база кај записот за соодветната слика и емаил адреса.

### 3. АДМИН ПАНЕЛ

Администраторскиот панел е достапен само за автентифицирани администратори. Не треба да може да се регистрираат нови администратори, туку хардкодирани во база да може да се најават. Тука, се листаат сите сметки од база во формат на картички, слично на ова:



Сликата на картичката е самата сметка (сликата што ја прикачил корисникот). Текстот под неа е текстот што го вратило апи-то. Под картичката има 2 копчиња: **Approve / Decline**.

- **Клик на Decline**

Сметката во база се означува како одбиена (declined = 1). Сите одбиени сметки ќе се листаат во ново мени и се тргаат од страната со картички погоре.

- **Клик на Approve**

Се појавува попап од каде што админот ќе може да избере награда. На пример, dropdown листа со производи: капа, маица, наочари итн. После направен избор и submit на формата во попап, на клиентот му се праќа mail со информација дека добил награда, а сликата се маркира како прифатена (approved = 1). Во ново мени админот ќе може да ги види сите прифатени сметки.

Сумарно, админот има пристап до 3 страници: сите одбиени сметки, сите наградени сметки и сите сметки кои чекаат на одлука - pending сметки (скриншотот погоре). *Сите одбиени и сите наградени сметки* може да го имаат истиот дизајн како и сите пендинг сметки, со тоа што не е задолжително. Во менито каде што се листаат сите наградени сметки, до секоја сметка треба да стои информација која награда е доделена.

Кај сите pending сметки, со checkbox-ови треба да може да се филтрираат:

- сите сметки (селектирано по дифолт)  
се листаат сите сметки што чекаат на одлука (без тие што се веќе одобрени или одбиени)
- Сите 'сигурно' сметки  
се листаат само само тие сметки што чекаат на одлука и за кои апи-то вратило дека се со статус 1
- Сите 'можеби' сметки.  
се листаат само само тие сметки што чекаат на одлука и за кои апи-то вратило дека се со статус 2

### **Забелешка**

Во оваа фаза, наместо url-to кое е потребно за моделот од data-science, користим апи симулација кое ќе враќа рандом податоци.

За поедноставно комплетирање на front-end делот препорачано е користење на Bootstrap, но не е задолжително. Можете да понудите сопствено решение или да користите некоја друга библиотека (Tailwind, Bulma, Material UI итн.)

## Систем на бодување

Функционалност	Поени
Функционалности во Кориснички Панел	
На клик се појавува модалот за аплицирање.	5
Front End валидација за секој инпут.	5
Back End валидација за секој инпут.	5
Приказ на ладер.	5
Приказ на преостанати обиди и забрана на корисникот да се пријави во следните 24 часа доколку се направат 3 неуспешни обиди.	5
Функционалности во АПИ повик	
Сликата се зачувува при аплицирање.	5
Доколку АПИ повикот врати статус 1 или 2 се зачувуваат информациите за корисникот.	10
Доколку АПИ повикот врати статус 0 не се зачувуваат информациите за корисникот и се брише сликата.	10
Функционалности во Админ Панел	
Систем за автентикација на админ корисници.	5
Картичките ги содржат сите податоците за клиентите.	5
Приказ на сите pending апликации.	5
Со клик на <b>Approve</b> се наградува клиентот.	5
Можност за одбирање на награда.	5
Испраќање на мејл до клиентот.	5
Со клик на checkbox се филтрираат сметките според статус.	5
Со клик на <b>Decline</b> се запишува картичката како одбиена.	5
Приказ на сите одбиени апликации.	5
Приказ на сите наградени апликации.	5

<b>Вкупно</b>	<b>100 поени</b>
<b>Дополнителни поени</b>	<b>20 поени</b>
Податоците од модалот се праќаат преку AJAX повик. Филтрирањето на картичките се одвива преку AJAX повик. Клиентите се наградуваат преку AJAX повик.	5
Копче во админ панел ( во pending tab-от ) со кое ќе може да се избере сметка по случаен избор за истата да се награди	5
Опција за export на податоците во excel фајл во форма на репорт со клик на копче или преку cron job.	10
<b>Максимален број на освоени поени</b>	<b>120 поени</b>