Marketplace:

Introducció

- Vídeo de presentació del projecte

Marketify és una plataforma de marketplace en línia que connecta artesans i creadors de productes casolans amb consumidors interessats en adquirir articles únics i de qualitat. A través de Marketify, els usuaris poden crear un compte, explorar una àmplia gamma de productes casolans i establir comunicació directa amb els venedors a través d'un sistema de xat integrat. A més, els usuaris també tenen l'oportunitat de crear les seves pròpies botigues en línia dins de la plataforma, exhibint i venent les seves pròpies creacions. Marketify ofereix un sistema de generació automàtica de factures en format PDF per a garantir la transparència en les transaccions i facilitar la gestió administrativa. A través de Marketify, es busca crear una comunitat activa i vibrant, organitzant esdeveniments i fomentant la col·laboració entre els usuaris. L'objectiu principal de Marketify és proporcionar una experiència de compra enriquidora per als consumidors i una plataforma eficaç per als artesans i creadors de productes casolans.

Memoria setmanal

Setmana 12

David

Comprovar l'estat del producte abans de comprar (en la vista del carret, vista de l'ordre i al presionar le botó de comprar) Temps estimat ⇒ 2h Temps Real ⇒ 4h

Controlar errors al crear noves ordres: Temps estimat ⇒ 2h Temps Real ⇒ 1h

Afegir LOGs als següents controladors: Cart, Chat, Order, Product i Shop Temps estimat ⇒ 2h Temps Real ⇒ 2h

Esborrar codi mort, codi de test i comentar funcions Temps estimat ⇒ 1h Temps Real ⇒ 1h

Solucionar error de notificacions i error de paginació Temps estimat ⇒ 2h Temps Real ⇒ 2h

Posibilitat de marcar com fallida una ordre, tornar a posar a la venta els productes Temps estimat ⇒ 3h Temps Real ⇒ 2h

Modificar consulta per modificar el 'display' dels productes Temps estimat ⇒ 1h Temps Real ⇒ 2h

Modificar el vagrant per incluir el projecte laravel a la ip 172.16.50.50 (connexió HTTP) Temps estimat ⇒ 6h Temps Real ⇒ 4h

Permetre Connexió HTTPS i redirecció desde HTTP Temps estimat ⇒ 3h Temps Real ⇒ 5h

Conectar API amb màquina vagrant i fer-la servir amb la màquina vagrant del site Temps estimat

⇒ 6h Temps Real ⇒ 5h

Corregir alguns errors/bugs relacionats amb l'API Temps estimat ⇒ 3h Temps Real ⇒ 4h

Slider fet amb JavaScript per la vista product.show Temps estimat ⇒ 1h Temps Real ⇒ 1h

Fer funcionar LandingPage malgrat que hi hagi un json amb un format erroni Temps estimat ⇒ 1h Temps Real ⇒ 2h

Funcionalitat per canviar contrasenya amb correu i token asociat al compte Temps estimat ⇒ 2h Temps Real ⇒ 2h

Guia de instal·lació Temps estimat ⇒ 1h Temps Real ⇒ 1h

Oscar

Poner los mensajes de Log Info en diferentes páginas Temps estimat ⇒ 1h Temps Real ⇒ 1:30h

Eliminar una imatge en concret Temps estimat ⇒ 2h Temps Real ⇒ 1:30h

Eliminar totes les imatges d'un producte en concret Temps estimat ⇒ 2h Temps Real ⇒ 1:30h

Pujar una imatge en concret Temps estimat ⇒ 2h Temps Real ⇒ 1:30h

Crear el disseny del Historic de Comandes Temps estimat ⇒ 2h Temps Real ⇒

Albert

Problemes amb el shop seeder Temps estimat ⇒ 1h Temps Real ⇒ 1,5h

Crear test de Chat Temps estimat ⇒ 5h Temps Real ⇒ 1,5h → encara continua

Error entorn de desplegament Temps estimat ⇒ 1h Temps Real ⇒ 6,25 → encara continua

Actualitzar test CategoryProduct i crear OrdersItems Temps estimat ⇒ 10h Temps Real ⇒ 5,5h

Modificar ShopFactory Temps estimat ⇒ 1h Temps Real ⇒ 1h

Configuració entorn testin JS Temps estimat ⇒ 1h Temps Real ⇒ 0,75h

Crear vista de management Temps estimat ⇒ 5h Temps Real ⇒ 4,75h

Comentar funcions i afegir logs Temps estimat ⇒ 5h Temps Real ⇒ 1,5h

Update vista d'orders i estils Temps estimat ⇒ 2h Temps Real ⇒ 2,25h

Documentació Temps estimat ⇒ 5h Temps Real ⇒ 6,5h === Setmana 11

David

Ajudar a Albert amb la creació d'orders Temps estimat ⇒ 2h Temps Real ⇒ 4h

Solucionar errors del xat Temps estimat ⇒ 2h Temps Real ⇒ 2h

Xat funcionant jutament amb les commandes Temps estimat ⇒ 3h Temps Real ⇒ 7h

Solucionar errors del carret Temps estimat ⇒ 2h Temps Real ⇒ 3h

Millorar diversos elements de l'estil Temps estimat ⇒ 2h Temps Real ⇒ 2h

Xat funcionant amb diferents enllaços // route(chat.show) Temps estimat ⇒ 2h Temps Real ⇒ 2h

Comprovar l'estat del producte abans de realitzar una comanda Temps estimat ⇒ 6h Temps Real ⇒ 5h

Afegir LOGs a Cart, Chat, Order, Product i Shop Controllers Temps estimat ⇒ 1h Temps Real ⇒ 1h

Oscar

Crear el nou projecte d'API Temps estimat ⇒ 1h Temps Real ⇒ 30min

Crear la taula de Imatges Temps estimat ⇒ 1h Temps Real ⇒ 30min

Obtenir una imatge en concret Temps estimat ⇒ 2h Temps Real ⇒ 3h

Obtenir totes les imatges d'un producte Temps estimat ⇒ 1:30h Temps Real ⇒ 1:30h

Poner los mensajes de Log Info en diferentes páginas Temps estimat ⇒ 1h Temps Real ⇒ 1:30h

Albert

Update factories and seeders Temps estimat ⇒ 3h Temps Real ⇒ 2,5h

Update product status to sold Temps estimat ⇒ 2h Temps Real ⇒ 1,75h

Estandarització de mediaqueries Temps estimat ⇒ 1h Temps Real ⇒ 1,25h

Resoldre problemes amb l'orders Temps estimat ⇒ 1h Temps Real ⇒ 1,75h

Crear new repository Temps estimat ⇒ 1h Temps Real ⇒ 1,5h

Crear orders Temps estimat ⇒ 10h → Total 22,5h Temps Real ⇒ 6,25h → venia de la setmana anterior

Setmana 10

David

Millorar el factory de les categories. Utilitzar categories reals. Temps estimat ⇒ 1h Temps Real ⇒ 1h

Fer URLS personalitzades per diferents shops. Temps estimat ⇒ 3h Temps Real ⇒ 4h

Els usuaris amb tendes no poden afegir al carret els seus propis productes. Temps estimat ⇒ 3h Temps Real ⇒ 3h

Millorar la vista dels formularis de les tendes (crear i modificar tendes i productes). Temps estimat ⇒ 2h Temps Real ⇒ 3h

Validar correctament tots els formularis, utilitzar missatges de validació personalitzats. Temps estimat ⇒ 4h Temps Real ⇒ 5h

Vista de tenda normal i vista de tenda d'administrador millorades. Temps estimat ⇒ 4h Temps Real ⇒ 3h

Night/day mode afegit. Temps estimat ⇒ 5h Temps Real ⇒ 2h

Header/nav responsives, millorat l'estil de les card dels productes (afegir descripció, tags HTML correctes...). Temps estimat ⇒ 2h Temps Real ⇒ 2h

Els seeders ara tenen 3 usuaris: 1 comprador, i 2 vendedors amb les seves respectives tendes. Temps estimat ⇒ 1h Temps Real ⇒ 2h

Per comprar l'usuari ha d'estar logejat. Temps estimat ⇒ 1h Temps Real ⇒ 1h

Alert de javascript per butons amb accions importants. Temps estimat ⇒ 1h Temps Real ⇒ 1h

Solucionar errors de javascript i del carret a servidor. Temps estimat ⇒ 2h Temps Real ⇒ 2h

Funcionalitat del xat, no implementada amb els orders Temps estimat ⇒ 16h Temps Real ⇒ 22h

Oscar

Albert

Crear orders Temps estimat ⇒ 10h Temps Real ⇒ 16,25h → continua encara

Crear logInfo Marketify Temps estimat ⇒ 2h → Total 2,5h Temps Real ⇒ 0,5h → venia de la setmana anterior

Setmana 9

David

Administrador de tenda (crear, modificar, afegir logo, afegir URLs) Temps estimat ⇒ 4h Temps Real ⇒ 5h

Afegir, editar, esborrar i amagar productes des de la vista administrador de la tenda Temps estimat ⇒ 3h Temps Real ⇒ 4h

Venedor no pot afegir al carret els seus productes Temps estimat ⇒ 1h Temps Real ⇒ 2h

Modificar tots els formularis per mostrar el feedback correctament (missatges d'error, color vermell...) Temps estimat ⇒ 3h Temps Real ⇒ 2h

Afegir vista als formularis de la tenda Temps estimat ⇒ 2h Temps Real ⇒ 2h

Afegir llista real de categories Temps estimat ⇒ 1h Temps Real ⇒ 1h

Canviar rutes per noms més apropiats Temps estimat ⇒ 1h Temps Real ⇒ 1h

Test de totes les rutes Temps estimat ⇒ 2h Temps Real ⇒ 3h

Oscar

Acabar els mockups de totes les vistes . Temps estimat ⇒ 1h Temps real ⇒ 30min

Maquetar la página de Landing Page(refactoritzar el codi). Temps estimat ⇒ 1h Temps real ⇒ 2h

Fer que llegeixi un json i mostri productes relacionats amb les categories. Temps estimat ⇒ 2 Temps real ⇒ 3h

Fer que quan facis click al títol de la categoria, surtin tots el productes filtrats per aquesta. Temps estimat ⇒ 2h Temps real ⇒ 2h

Fer el disseny de la Landing Page responsive. Temps estimat ⇒ 2h Temps real ⇒ 2h

Arreglar l'email de confirmació Temps estimat ⇒ 3h Temps real ⇒

Albert

Crear logInfo Marketify Temps estimat ⇒ 2h Temps Real ⇒ 2h → continua encara

Vista detall del producte Temps estimat ⇒ 5h → Total 7,25h Temps Real ⇒ 2,5h → venia de la setmana anterior

Diccionari javascript Temps estimat ⇒ 3h Temps Real ⇒ 2,5h

Realitzar Comanda Temps estimat ⇒ 15h Temps Real ⇒ 1,75 → continua encara

Setmana 8

David

Afegir un seeder de la tenda per l'usuari "venedor". Solucionar error del controlador de la tenda. Temps estimat ⇒ 2h Temps real ⇒ 2h Login ara funciona amb email i user en el mateix camp. Temps estimat ⇒ 1h Temps real ⇒ 1h Canviar formularis: labels, placeholders, names... Temps estimat ⇒ 1h Temps real ⇒ 1h Validar els inputs dels formularis de registre i d'editar usuari. Missatges flash descartats per missatges d'error. Temps estimat ⇒ 2h Temps real ⇒ 3h Carret de servidor completament funcional Temps estimat ⇒ 5h Temps real ⇒ 6h Redireccions 403 i errors 404 Temps estimat ⇒ 2h Temps real ⇒ 2h

Oscar

Crear dos usuaris de prova(venedor,comprador) Temps estimat ⇒ 1h Temps real ⇒ 25min

Fer skectchings de totes les pantalles Temps estimat ⇒ 1h Temps real ⇒ 45min

Fer Wireframes de totes les pantalles Temps estimat ⇒ 1h Temps real ⇒ 1h

Fer mockups de totes les pantalles Temps estimat ⇒ 2h Temps real ⇒ 1:30h

Implementar la pagina de la landing(maquetació) Temps estimat ⇒ 1h Temps real ⇒ 1h

Fer la implementació de la landing Temps estimat ⇒ 2h Temps real ⇒

Albert

Fer test de producte Temps estimat ⇒ 10h → Total 23h Temps Real ⇒ 10,5h → venia de la setmana anterior

Update cards and media queries Temps estimat ⇒ 10h Temps Real ⇒ 6,5h

Vista detall del producte Temps estimat ⇒ 5h Temps Real ⇒ 4,75h → continua oberta

Setmana 7

David

Terminar amb la creació de la tenda (rol de venedor, modificar bdd...) Temps estimat ⇒ 2h Temps real ⇒ 3h Menú amb possibilitat de crear tenda (si l'usuari ja és venedor, es canvia aquest botó per accedir a la seva tenda) Temps estimat ⇒ 2h Temps real ⇒ 3h Separar de manera correcta les pàgines de "edit" i "show". Show funciona amb ids a l'url /user/1 i edit amb /user/edit (ara agafa l'ID amb l'usuari autenticat). També per la tenda. Temps estimat ⇒ 2h Temps real ⇒ 4h Carret es guarda a la base de dades, però no té un ús real. Temps estimat ⇒ 2h Temps real ⇒ 2h

Oscar

Fer funció que permeti cambiar el nom d'usuari a la pagina de perfil d'usuari Temps estimat ⇒ 2h Temps real ⇒ 1h

Fer la funció que permeti canviar la contrasenya al perfil d'usuari Temps estimat ⇒ 1h Temps real ⇒ 1h

Solucionar el problema del canvi de contrasenya Temps estimat ⇒ 2h Temps real ⇒ 2h

Solucionar el problema del login i el register Temps estimat ⇒ 2h Temps real ⇒ 1h

Solucionar el problema de l'enviament d'emails Temps estimat ⇒ 2h Temps real ⇒ 1:30h

Albert

Documentació i planificació Temps estimat ⇒ 1h Temps Real ⇒

Fer test de producte Temps estimat ⇒ 10h Temps Real ⇒ 10,5h → continua oberta

Afegir estil a la vista d'edició de la configuració d'usuari Temps estimat ⇒ 2h Temps Real ⇒ 1,75h

Setmana 6

David

User show page, user edit page: Show user page. Temps estimat ⇒ 1h Temps real ⇒ 1h Edit user page. Temps estimat ⇒ 1h Temps real ⇒ 2h Afegir canvi de contrasenya. Temps estimat ⇒ 2h Temps real ⇒ 3h Afegir avatars (canviar i esborrar) Temps estimat ⇒ 5h Temps real ⇒ 5h Crear tenda (només formulari i vista) Temps estimat ⇒ 2h Temps real ⇒ 3h

Oscar

Albert

Resoldre hotfix en css Temps estimat ⇒ 1h Temps Real ⇒ 0,5h

Resoldre hotfix a les migracions de shop Temps estimat ⇒ 1h Temps Real ⇒ 0,5h

Fer test relació Category-Product Temps estimat ⇒ 2h Temps Real ⇒ 2,5h

Fer test de producte Temps estimat ⇒ 10h Temps Real ⇒ 2h → continua següent setmana

Setmana 5

David

Caràcters especials al factory i seeders Carret funcionant a Javascript (amb localstorage): Petició post finalment descartada. Botons d'afegir i treure al carret al home i a detall de producte. Utilitzar mòduls de javascript. Comprovar si el producte s'ha afegit al carret abans de carregar el botó. Contador de productes a la icona del carret (en el menú). Icona del carret dirigeix amb la stringQuery des de totes les pàgines. Mostra correctament els productes a la pàgina del carret i la suma total dels preus. Temps estimat ⇒ 6h Temps real ⇒ 16h

Oscar

Albert

Investigació de testing Temps estimat ⇒ 5h Temps Real ⇒ 5h

Unificació estil de les targetes i reutilització de codi css Temps estimat ⇒ 5h Temps Real ⇒ 10h

Setmana 4

David

Millorar el ProductControler. Utilitzar sessió per les variables del $Request. Temps estimat ⇒ 2h Temps Real ⇒ 3h

Carret amb productes no únics Temps estimats ⇒ 3h Temps real ⇒ 6h

Passar múltiples variables i repetició de codi en cada Controller a variables de sessió, utilitzant el component 'Header'. Temps estimats ⇒ 2h Temps real ⇒ 2h

Resoldre conflictes Temps real ⇒ 1:30h

Carret amb productes únics Temps estimat ⇒ 1h Temps real ⇒ 1:30h

Mantenir el carret a l'hora de recargar la pàgina i re-utilitzar funcions de javascript (amb mòduls) per poder afegir desde la pàgina de detall del producte. Temps estimat ⇒ 3h Temps real ⇒ 1h

Oscar

Crear el disseny del register. Temps estimat ⇒ 2h Temps real ⇒ 2:30h

Crear el disseny del login. Temps estimat ⇒ 2h Temps real ⇒ 30 min

Crear disseny de la pagina de recuperació del password. Temps estimat ⇒ 2h Temps real ⇒ 2h

Fer la implementació del register. Temps estimat ⇒ 3h Temps real ⇒ 3h

Fer la implementació del login. Temps estimat ⇒ 3h Temps real ⇒ 2:30h

Aconseguir que envii els mails correpsondents. Temps estimat ⇒ 3h Temps real ⇒ 4h

Cambiar la funció selectCategories. Temps estimat ⇒ 1h Temps real ⇒ 15min

Fer la implementació de recordar la contrasenya. Temps estimat ⇒3h Temps real ⇒ 2:45h

Fer que envii els mails a tots els correus. Temps estimat ⇒ 1h Temps real ⇒ 20min ==== Albert

Arreglar problemes d'accesibilitat en els colors. Temps estimat ⇒ 1h Temps real ⇒ 1:45h

Problemes en la versió de node i packaje.json. Temps estimat ⇒ 2h Temps real ⇒ 4:30h

Actualitzar stil global. Temps estimat ⇒ 3h Temps real ⇒ 4h

Millorar i simplificar estil del header. Temps estimat ⇒ 2h Temps real ⇒ 2:30h

Millorar i simplificar estil del footer. Temps estimat ⇒ 1h Temps real ⇒ 1h

Millorar i simplificar estil del navigation. Temps estimat ⇒ 1h Temps real ⇒ 0:30h

Documentació Temps estimat ⇒ 1h Temps real ⇒ 1h

Setmana 3

David

Fer la paginació completament personalitzada. Temps estimat ⇒ 1h Temps real ⇒ 1:30h

Modificar factory per fer tests amb accents i caràcters especials. Temps estimat ⇒ 2h Temps real ⇒ 1:30h

Modificar factory per utilitzar rutes físiques en comptes de URLs. Temps estimat ⇒ 1h Temps real ⇒ 30 min

Solucionar problema relacionat amb els inner joins amb el filtre de les categories. Temps estimat ⇒ 2h Temps real ⇒ 2:30h

Oscar

Albert

Actualitzar la funcionalitat del migrate:rollback per solucionar els errors que hi havia. Temps estimat ⇒ 1h Temps real ⇒ 2h

Investigació sobre com fer test en php Temps estimat ⇒ 5h Temps real ⇒ 10h

Crear test del store amb tdd. Temps estimat ⇒ 1h Temps real ⇒ 1:30h

Crida general als seeders desde el seeder general Temps estimat ⇒ 1h Temps real ⇒ 1h

Documentació Temps estimat ⇒ 1h Temps real ⇒ 1h

Setmana 2

David

Aquesta setmana he redissenyat la home view 3 vegades. Més que res per fer-la completament responsive i ajustar el grid correctament. La primera versió no era responsive. La segona versió tenia 6 media queries, i la tercera i última només té un media query i un grid amb un repeat amb minmax per fer-lo més automàtic. També he buscat i afegit icones, i he fet un logo pel projecte.

A més de la vista del home, he fet un component per al header, i un altre pel navigation. El navigation només es mostra per dispositius mòbils o tablets: he pensat que es una bona idea posar el navigation sota del tot per no saturar massa el header. - Header: logo, search bar, ordenació, icona d'usuari i icona del carret - Navigation: icona de home i icona d'usuari

La barra de resultats de cerca és funcional, he afegit una vista sense detall pel carret i el login i fet la paginació.

He afegit tipografia a la pàgina a "typografy_css", utilitzant l'arquitectura SASS.

He mirat per crear imatges aleatòries amb un factory i faig servir un fakeimg.

Oscar

Aquesta setmana he tingut alguns contratemps amb les migracions i els filtres.

He aconsgeuit resoldre els problemes amb la taula del mig. També he hagut de resideñar algunes migracions, per a que així pugui filtrar bé...

He fet el filtre de categories,encara que m'ha donat bastants problemes.

També he fet el filtre de tags, y que puguis filtrar per catergoria i per tag a la vegada, així com per nom del producte.

Per últim he actualitzat la guía d'estils i la documentació ja que hem redisenyar la home un altra cop.

Albert

Durant la setmana he realitzat les següents tasques:

- Creació de la relació n-m entre les taules.
- La creació de les migracions i seeders a la base de dades. Elaboració de les migracions i dels seeders.

A les migracions s'afegeix l'estructura de taules i relacions que tindrà la base de dades, s'inclou també les connexions per realitzar una relació n-m entre les taules de "productes" i de "categories"

S'afegeixen els seeders que l'ompliran de dades fictícies gràcies a les factories. També s'afeixen les condicions necessàries en el mètode down() dels seeds perquè en cas de fer un migrate:rollback es desvinculin les relacions entre taules i es pugui a continuació esborrar les dades i les taules.

- L'eliminació dels fitxers que van ser inclosos en el repositori remot, i que no ho haurien d'haver estat inclosos.

L'eliminació dels fitxers .env i alguns fitxers de configuració de la màquina virtual de Vagrant per la base de dades.

Aquestes tasques es van allargar més del temps estimat. - Creació i configuració del migrate:rollback ⇒ 2h - Creació de la relació n-m ⇒ 5h Entre documentació, cerca d'informació i realització de les tasques han superat el temps previst i superat el temps de les 18h de classe.

Setmana 1

Hem començat el projecte creant un trello per definir les tasques del projecte i organitzar-nos. Hem creat un repositori al git de l'institut per al projecte, amb les branques principals de treball (main, development)

David

Començar a familiaritzar-se amb git, crear tasques de trello, instal·lar SASS i el layout main de la pàgina.

Oscar

Començar a familiaritzar-se amb git, crear tasques de trello, dissenyar i crear la BBDD, guia d'estil i el layout de la HomePage

Albert

Demostració d'ús de git a l'equip, instal·lació del laravel al projecte, creació del Vagrant per allotjar la BBDD.

Disseny base de dades

[marketifyBBDD] | *media/marketifyBBDD.PNG*

Usuaris

Register

- La funcionalitat de registre d'usuari permet als nous usuaris crear un compte al sistema per accedir a les seves funcionalitats i recursos.
- El nou usuari a de proporcionar al formulari de registre la informació que es demana per tal de finalitzar emb exit el procés de registre.
- Una vegada l'usuari ha proporcionat totes les dades requerides i ha fet clic al botó de registre, el sistema processa la sol·licitud i crea un nou compte d'usuari amb les dades proporcionades. En aquest punt, el sistema realitza diverses comprovacions, com verificar si l'adreça de correu electrònic ja està registrada o si s'han complert determinades polítiques de contrasenya.

Login

- La funció d'inici de sessió permet als usuaris accedir de manera segura al sistema utilitzant les seves credencials úniques.
- Els usuaris proporcionen el seu nom d'usuari o adreça de correu electrònic i la seva contrasenya per autenticar-se en el sistema.
- El sistema verifica l'autenticitat de les credencials i, si són correctes, permet l'accés a les funcions i àrees protegides del sistema. S'implementen mesures de seguretat per protegir les credencials i garantir la privacitat de l'usuari durant el procés d'inici de sessió.

Recordar contrassenya

Aquesta funcionalitat és molt útil per als usuaris que ja estan registrats al sistema i han oblidat la seva contrasenya. Amb aquesta funcionalitat, els usuaris poden recuperar l'accés al sistema creant una nova contrasenya. Aquest procés es realitza mitjançant l'enviament d'un enllaç especial al correu electrònic associat a l'usuari.

Per començar, l'usuari ha d'anar a la pàgina d'inici de sessió i seleccionar l'opció "Recordar contrasenya". A continuació, s'ha de proporcionar la direcció de correu electrònic associada al seu compte. Una vegada l'usuari envia aquesta informació, el sistema processa la sol·licitud i genera un enllaç únic.

Aquest enllaç és enviat a l'adreça de correu electrònic proporcionada per l'usuari. En l'email, s'inclou un missatge informatiu que explica el propòsit de l'enllaç i les instruccions a seguir. L'usuari ha de fer clic a l'enllaç proporcionat, el qual el redirigirà a una pàgina especial per a la creació d'una nova contrasenya.

A la pàgina de creació de contrasenya, l'usuari ha de proporcionar una nova contrasenya i confirmar-la per garantir l'exactitud. També pot ser requerit que l'usuari compleixi certs requisits de seguretat en la creació de la contrasenya, com una longitud mínima o l'ús de caràcters especials.

Un cop l'usuari introdueix la nova contrasenya i la confirma, el sistema verifica i actualitza la contrasenya associada a aquest compte d'usuari. A partir d'aquest moment, l'usuari podrà utilitzar la nova contrasenya per accedir al sistema amb les seves credencials actualitzades.

Aquesta funcionalitat de recordar contrasenya proporciona als usuaris una manera senzilla i segura de recuperar l'accés al sistema en cas d'oblit de contrasenya. És important destacar que s'implementen mesures de seguretat adequades per protegir les dades personals i garantir que només l'usuari legítim pugui canviar la contrasenya.

Productes

Planificació a la base de dades

Per a poder fer la planificació a la base de dades dels productes, vam haver de planificar varies taules.

Taula de productes

- Al nostre grup es va decidir que la taula de productes hi haurien els tags relacionats amb el producte. Perque creiem que era la manera més fácil de gestionar el filtrat per tags, ja que només haviem d'agafar el camp a la taula i filtrar per aquest.

Taula de categories

- Per a la taula de categories vam decidir posar un identificador a la categoria i el propi nom d'aquesta(el qual vam posar com a únic).

Taula intermitja de categories i productes

- Nosaltres vam creure necessaria una Taula intermitja de categories i productes, ja vam pensar que un producte podia tenir més d'una categoria, i una categoria havia de tenir més d'un producte.

Tags

Pel que fa als tags, vam haver de plantejar-nos diferents possibilitats.

- Crear tags i afegir-los als productes.
- Fer subcategories i filtrar també per subcategories.

Finalment ens vam decantar per afegir tags als productes, ja que vam pensar que era la manera més óptima i rápida de fer-ho, sense que ens portès moltes hores. Sobretot, vam creure que la era la manera més fácil i rápida de poder filtrar per productes. === Possibles conflictes

- Quan es carregava el seeder de "prodcuts", apareixia un error de que no podia crear els productes perque les foreign keys de la taula intermitja de categories i productes estava mal formada. Per sort i amb molt de treball, vam trobar la soluciió i vam poder crearla.

Busqueda i filtre

Cerca i filtres

- Prácticament a cada página apareix un botó de desplegament on apareixen totes les categories, allá es pot filtrar per aquestes i apareixen tots el prodcutes que tenen aquesta categoria.
- També apareix una barra de cerca on es pot filtrar, tant per el nom com per el tag/tags que té cada producte.
- Per últim, apareix un altre botó de desplegament on es pot filtrar per l'ordre que l'usuari vulgui que li apareiguin els prodcutes.

Categories

En aquest apartat, només s'han utilitzat funcions amb PHP. On es fa una o varies consultes i es carreguen el productes. De cara al botó de desplegament de les categories, hem utilitzat les següents funcions:

- searchSpecific(). Aquesta funció, fa una consulta a la taula "prodcuts" on s'obté els productes, filtrats per categoria. També es pot filtrar per la barra de cerca i per l'ordre si l'usuari ho dessitja.

Barra de cerca

En aquest altra apartat,també s'han utilitzat funcions amb PHP. Per la barra de cerca, hem utilitzat les següents funcions:

- searchSpecific(). En aquesta funció es fa una consulta a la taula "products" on s'obtenen els productes filtrats per la barra de cerca. També es pot filtrar per categoria i per l'ordre si l'usuari ho dessitja.

Ordre

En aquest últim apartat,també s'han utilitzat funcions amb PHP.

- Per a filtrar per l'ordre en que apareixen els productes, s'utilitzen les dues funcions

mencionades previament, ja que permeten també filtrar per l'ordre.

- L'usuari pot filtrar per nom ascendent i descendent, i per preu ascendent i descendent.

Carret de la Compra

- El carret mostrarà els productes que es trobin a la QueryString de la pròpia vista (/cart?id=).
- Hi ha una suma total dels costs dels productes a la dreta de la vista.
- Pots buidar el carret completament, o bé, pots eliminar un producte de manera selectiva.

Carret en local

Les funcions principals del carret són a JavaScript.

Primer de tot, quan es carrega la pàgina, comprova la variable 'cart' al localstorage per agafar les IDs dels productes corresponents. Després comprova aquestes ID amb les dels productes mostrats a la vista per canviar l'innerText del botó de 'Add to cart' a 'Remove from cart'. Després de comprovar i executar aquesta funcionalitat, hi ha un eventlistener dins d'un forEach esperant que el botó de qualsevol producte sigui clicat.

En cas de mostrar-se l'innerText com 'Remove from cart', l'script encarregat treu la ID específica de l'array allotjat al localstorage.

En cas que l'innerText del botó sigui 'Add to cart', afegeix la respectiva ID a l'array amb un push, s'executa la funcionalitat comptador d'elements de l'array per mostrar-ne un número sobre la icona del carret, i es crea la querystring per mostrar els productes al carret.

Carret en servidor

El carret en servidor utilitza les funcionalitats de JavaScript per funcionar. L'única variació és l'afegiment d'un script per detectar si l'usuari està autenticat a laravel, des d'una ruta feta a web.php, on retorna un boolean.

En cas que no estigués autenticat, s'allotjaria únicament al localstorage.

Si l'usuari s'ha donat d'alta i està logejat, en cada actualització (afegir/treure al carret, navegar a un altra pàgina, etc.) envia una petició POST al servidor on guarda la informació a la base de dades.

Possibles conflictes

- Quan l'usuari logejat tanca sessió, el carret local es buida.
- Quan hi ha informació al localStorage i l'usuari inicia sessió, es sobreescriu la informació local amb la del servidor.
- Quan un usuari no autenticat ha afegit productes al carret i es registra, aquests productes són pujats a la base de dades amb el respectiu usuari.

Shops

Rols d'usuaris

Per determinar si un usuari és propietari d'una tenda o no, hem afegit els rols de 'venedor' i 'comprador', establint aquest últim com el valor per defecte. Depenent del rol de l'usuari, canviaran opcions al menú i icones a les pàgines. En el menú, l'usuari veurà l'opció de 'Crear tenda' o 'Administrar tenda'. En la versió Mobile, la icona del home es canviarà per la icona d'una tenda. L'usuari sempre tindrà a disposició el botó amb el logotip per retornar a la landing page.

Formulari de creació

El formulari de creació de tenda està fet aprofitant un mixin del SASS dedicat als formularis. Aquest formulari demana el nom de la tenda, nom complet i DNI del venedor, logotip, colors de la capçalera i del fons, ...

URLs

Les URL es generen automàticament segons el nom de la tenda. Les lletres passen a ser totes minúscules, els espais són substituïts per guions i els caràcters especials eliminats. En cas d'una combinació estranya amb caràcters especials, els noms de les tendes no poden ser desats si com a resultat dona una URL ja existent.

Administració de shop

La tenda disposa d'una vista per al propietari. El propietari serà l'únic en accedir a l'administració de la seva tenda.

Editar shop

Es pot editar tots els camps de la tenda: · Nom de la tenda

· Nom complet del propietari

· DNI/NIF del propietari

· Logotip de la tenda

· Color de capçalera de la tenda

· Color de fons de la tenda

· Ordre d'aparició dels productes

Es reaprofita el mixin utilitzat pel formulari de creació.

Afegir, editar, amagar i esborrar productes

Des de la vista d'administrador, l'usuari propietari és capaç de crear productes, esborrar-los, editar-los i amagar-los.

Menys l'afegir, aquestes opcions apareixen en cada producte en forma d'icones.

Altres consideracions

- En la creació del seeder de shops, hem tingut problemes amb l'id de botiga, que al estar relacionat amb l'id d'usuari en alguns casos donava errors en la comanda de db:seed.
- Al assignar de forma fixe un user_id a la botiga en cas de repetir l'ordre de db:seed sense havere fet un migrate donava un error per la repetició de la id. Per la qual cosa, després de varies proves. s'ha arribat a la conclusió que es un aspecte a millorar el fet de controlar millor aquest seeder per evitar conflictes.
- Per falta de temps s'accepta que no es pot tornar a llançar les db:seed sense fer un migrate:fresh abans. Ja que en altra manera hi ha duplicitats que fan fallar el sistema.

Realitzar comanda

Control del carret

- La vista order mostra un resum dels articles que conte el carret de l'usuari, amb el nom i el preu dels productes agrupats per venedor
- Es controla que un producte no hagi estat venut abans de pasar al realitzar la comanda
- En cas de que un producte hagi estat venut no es pot realitzar la comanda i s'informa al client dels productes ja venuts i que els ha d'eliminar manualment
- Un cop tots els productes del carret estan disponibles per vendre i realitzaem la comanda. s'activa el botó de realitzar comanda.

Realització de la comanda

- Al presionar el botó de realitzar comanda es genera una order per cada venedor implicat en els productes del carret
- Es crea un registre a la base de dades de orders amb les dades de la comanda. I a la taula de ordersItems es crea un registre per cada producte en aquell order, amb la informacio del order
- Es crea al mateix temps un chat unic entre el client i el venedor relacionat amb aquell id de order

Buidatge del carret

- Quan l'order s'ha realitzat amb exit i s'han creat els registres a la base de dades, aquells productes que han estat venuts en aquelles orders, es marcaran a la taula de products com a que ja estan venuts.

- També es procedeix a borrar les dades del carret tant en local com el carret del servidor asignat a aquell usuari. De manera que quan vulgui tornar a compra el carret estarà buit i es podra tornar a cpmençar el cicle.

Missatgeria Interna

Creació del xat

El xat és creat una vegada que s'ha creat l'ordre. Agafa la ID de l'usuari comprador i la ID de l'usuari venedor per crear-ho. Aquest xat inicia amb un missatge automàtic per part de l'usuari comprador on es confirma i presenta l'ordre.

Missatges automàtics

Els missatges automàtics estan determinats per l'estat de l'ordre, que pot ser un dels quatre següents:

- Pendent: Quan l'ordre encara no s'ha processat completament.
- Pagat: Quan el pagament ha estat confirmat pel venedor.
- Enviat: Quan el venedor ha confirmat l'enviament dels productes.
- Rebut: Quan el comprador ha confirmat la recepció dels productes.
- Failed: Quan el venedor ha rebutjat el pagament.

El canvi d'estat de l'ordre és responsabilitat de l'usuari, ja que és el venedor qui confirma el pagament i l'enviament, mentre que és el comprador qui confirma la recepció dels productes.

Missatges manuals

Els missatges són enviats mitjançant un formulari amb un camp d'entrada de text i un botó de tipus 'submit'. Un cop rebuts, es validen al controlador i s'afegeixen a la base de dades. Posteriorment, la pàgina es refresca per mostrar el nou missatge.

A la base de dades, els missatges s'emmagatzemen amb la següent informació: la ID de l'usuari que ha enviat el missatge, el contingut del missatge i un booleà per indicar si el missatge és automàtic. Aquesta última opció només es considera veritable (true) si es marca així mitjançant la creació d'un missatge dins d'una funció del controlador.

Notificacions

Quan es crea un xat, es generen les notificacions dels usuaris, que inclouen la ID de l'usuari i un booleà que indica si el missatge ha estat llegit o no. Cada xat té dues notificacions, una per a cada participant. En el moment en què l'usuari marca el missatge com a llegit, el booleà es canvia a true en lloc de crear una nova fila.

Funcionament del xat

El xat i les notificacions no són asíncrons; cal actualitzar la pàgina per veure la informació més recent. La pàgina de missatges mostra una llista de tots els xats als quals l'usuari pertany, tant si és un venedor com un comprador, i inclou una secció on es mostren els missatges. Aquesta secció utilitza la funció "show" en el controlador de xats.

Possibles conflictes

- Si l'usuari introdueix manualment una ID a l'URL que no correspon a aquest usuari, el xat no es mostrarà.

Pàgina històric de comandes

- L'històric de comandes mostrará totes las comandes realitzades per un usuari.
- L'usuari podrá veure la comanda detallada si ho desitja.
- El propietari de la botiga podrá consultar les comandes realitzades a la seva botiga.
- L'usuari podrá descarregar un pdf amb la seva comanda.

Historial de comandes

En general, totes les funcions son PHP i están integrades, tant al model (Order) com al controlador.

Només carregar la página, es mostrará una llista amb totes les comandes realitzades (están separades per la botiga a la qual pertany el producte/s) per l'usuari, amb la seva data i el seu id corresponent. Aquesta funció s'executa fent una consulta a la taula "shops" filtrant per l'id del usuari , per a treure l'id de la botiga. Seguidament, es fa una altra consulta a la taula "orders", flitrant per l'id de l'usuari i per l'id de la botiga. Així s'obtenen les comandes.

Detall de la comanda

Quan l'usuari fa click a una comanda, li redirigeix a una página amb els detalls de la comanda. En aquesta página es mostra un llistat de tots el productes amb els seus preus i el preu total. A més, l'usuari tindrá l'opció de descarregar la comanda en PDF.

Al detall de la comanda es mostra el nom de la botiga i els productes que l'usuari ha comprat (d'aquella botiga).

- Per al nom de la botiga, s'utilitza una consulta per agafar l'id de la comanda(filtrant a la taula "order_items" per l'id).
- Després, s'utilitza una altra consulta a la taula "products" filtrant per (per als productes), on retorna el producte/s que ha comprat l'usuari a aquella botiga. Si retornamés d'un producte, retornem un array. En canvi, si la consulta retorna un únic registre, retornem el producte.
- Per últim, l'usuari pot descarregar el PDF de la seva comanda. Per a poder descarregar-ho, s'utilitza la mateixa funció que per a obtenir els prodcutes i el nom de la botiga, però li pasem la

vista creada, generem el contingut del PDF amb "loadHtml()", renderitzem el contingut y per últim descarreguem en PDF la vista.

API d'imatges

API

En aquest apartat s'explica les diferents funcionalitats que té le api i com les utilitzem.

La configuració de la API es troba al Manual d'isntal·lació.

Funcionalitats

- index(). En aquesta funcionalitat s'obtenen totes les imatges principals per mitja d'una consulta. Vam decidir que agafes totes les principals, ja que sino aquesta funció (en principi era agafar totes les imatges amb el mètode all()) no s'utilitzava per res.
- catchImage(). En aquesta funcionalitat es fa una consulta a la taula "images" filtrant per l'id del producte, on s'obtenen totes les imatges d'un producte.
- insertImage(). En aquesta funció es crea la imatge, on pasem com a camps el nom de la imatge, el seu path, un boleeà per si és o no la imatge principal i l'id del producte.
- deleteImage(). En aquesta funció es fa una consulta a la taula "images" filtrant per l'id de la imatge i una vegada obté la imatge, l'esborra.
- deleteAll(). En aquesta imatge es fa una consulta a la taula "images" filtrant per l'id del producte i una vegada s'obtenen les imatges, les esborra.

Realitzar peticions

Per a poder realitzar peticions, es necessari aquest codi $response = $client→get(env('API_IP').'api/images', ['verify' ⇒ false]) envies le url de la api i es fa una petició on et retorna un contingut(en segons quina funció).Per últim, amb aquest codi $data = json_decode($response→getBody(), true) on retorna el contingut de la petició en forma d'array.

Entorns de desplegament

Creació de la maquina virtual vagrant

- Aquest fitxer de configuració de Vagrant estableix la imatge de la màquina virtual de debian.
- Establim les dades requerides del sistema que volem crear i les configuracions per al correcte funcionament de la màquina virtual.
- Amb totes les especificacións del sistema ja podem crear el host vagrant que configurarem en el següent pas.

Configuració de la màquina virtual

- La instal·lació del projecte consta de diversos passos que s'han de seguir per configurar correctament l'entorn de desenvolupament. En primer lloc, és necessari actualitzar els repositoris del sistema operatiu per obtenir les últimes actualitzacions disponibles.
- A continuació, s'instal·la el servidor web Apache, que és l'encarregat de servir les pàgines web i gestionar les sol·licituds entrants.
- Un cop instal·lat Apache, es procedeix a afegir el repositori de PHP 8.2, que és el llenguatge de programació utilitzat en el projecte. Després s'han d'instal·lar les extensions requerides per Laravel, el framework utilitzat en el projecte. Aquestes extensions proporcionen funcionalitats addicionals necessàries per al correcte funcionament de l'aplicació.
- Un cop completada la instal·lació de PHP, s'instal·la Composer, una eina que facilita la gestió de les dependències del projecte. Composer s'utilitza per descarregar i instal·lar les biblioteques i paquets necessaris per al desenvolupament de l'aplicació.
- A continuació, es crea un nou projecte Laravel, que és el punt de partida per al desenvolupament de l'aplicació. Laravel és un framework de codi obert que ofereix una estructura i característiques avançades per al desenvolupament web.
- Després de crear el projecte, es procedeix a configurar Apache perquè pugui servir l'aplicació Laravel. Això implica configurar la ruta d'accés als fitxers del projecte i habilitar la reescriptura d'URL, que és necessària per a l'enrutament de l'aplicació.
- A més de la configuració d'Apache, s'han de realitzar ajustos en els permisos dels fitxers i directoris del projecte per garantir que el servidor web tingui accés adequat a ells.
- Un cop realitzats aquests passos, es reinicia el servidor web Apache per aplicar els canvis realitzats. A continuació, s'instal·la el sistema de gestió de bases de dades MariaDB, que s'utilitzarà per emmagatzemar i gestionar les dades de l'aplicació.
- Es crea la base de dades específica per al projecte i es configura MariaDB per assegurar-se que l'aplicació pugui accedir-hi correctament.
- A més, es crea un usuari amb accés remot a la base de dades, la qual cosa permet que l'aplicació es comuniqui amb la base de dades des de diferents ubicacions.
- Després de configurar MariaDB, es realitza la instal·lació del certificat SSL per habilitar la comunicació segura a través de HTTPS. Això implica generar una clau privada i un certificat autofirmat.
- Finalment, es realitzen modificacions en la configuració d'Apache per habilitar el lloc SSL i redirigir el trànsit HTTP

Pàgines d'error personalitzat

Error de producte

Quan no es troba cap producte amb la ID proporcionada a la ruta 'product.show', es mostra una pàgina amb l'URL '/product-not-found', la qual informa l'usuari que el producte desitjat no existeix.

Error de tenda

Quan no es troba cap tenda amb l'URL proporcionada a la ruta 'shop.show', es mostra una pàgina amb l'URL '/shop-not-found', la qual informa l'usuari que la tenda desitjada no existeix.

Error d'usuari

Quan no es troba cap usuari amb la ID proporcionada a la ruta 'user.show', es mostra una pàgina amb l'URL '/user-not-found', la qual informa a l'usuari que l'usuari desitjat no existeix.

Error de pàgina

Quan no es troba cap pàgina amb l'URL proporcionada, es mostra una pàgina d'error però conservant l'URL, gràcies al fallback establit al fitxer de les rutes.

Mockup d'interficies

Guía d'estils

Enllaços

- Figma (Guia d'estils)

Colors utilitzats

Els colors que hem utilitzat en la nostra aplicació són:

Negre:

- Utilitzem el color negre, específicament el tonal "#0e0e0e", per als menús i el text de la nostra aplicació. Aquest color ens ofereix un bon contrast amb els altres colors de la paleta i permet una fàcil combinació amb ells. El color negre transmet un estil net i elegant, aportant sofisticació i claredat a la interfície. A més, el negre també facilita la llegibilitat del text, ja que proporciona un fons contrastant que permet als usuaris llegir la informació amb comoditat. Amb l'ús d'aquest color, busquem crear una estètica atractiva i professional en el nostre disseny.

Blanc:

- Dins de la nostra paleta de colors, inclou el color blanc. Aquest color és essencial tant per a fons com per a molts altres detalls de la interfície. El color blanc ofereix un contrast perfecte i combina de manera excel·lent amb el color negre. La seva presència permet destacar elements clau, crear espais amb sensació de netedat i transmetre una estètica elegant i minimalista a la nostra aplicació.

Blau:

- Hem seleccionat el color blau "#43a1cc" com a color secundari per al fons dels menús. Aquest color aporta elegància a la web i ofereix un contrast subtil, destacant els detalls importants sense perdre l'harmonia general del menú.
- També hem afegit variacions tonals d'aquest mateix color per a poder crear contrastos en cas que siguin necessaris, com en l'estat "hover" o altres elements on es requereixi destacar visualment. Això ens permet utilitzar diferents nuances de blau, mantenint la coherència i estil de la paleta de colors de la web.

Taronja:

- Hem triat el color vermell "#ff5208" com a color primari per als botons. Aquest color vibrants i enèrgic destaca els botons i crida l'atenció dels usuaris. Amb aquest color, volem transmetre un sentiment de vitalitat i acció, animant els usuaris a interactuar amb els botons i realitzar accions rellevants al sistema.

Verd Lima/Pi:

- Per als botons de compra i generar comanda, farem servir el color verd "#67ce74". Aquest color vibrant i fresc suggereix l'acció positiva de realitzar una compra o generar una comanda, donant una sensació d'èxit i confiança als usuaris. Això els animarà a prendre aquestes accions i reforçarà la seva experiència d'ús en la plataforma.

Vermell:

- El color vermell "#f43e3e" serà utilitzat per als missatges d'error a la nostra aplicació. Aquest color vibrant i intens transmet una sensació d'alerta i destaca clarament els missatges d'error per captar l'atenció dels usuaris. L'ús d'aquest color ens permetrà indicar de manera visual i efectiva quan hi ha problemes o errors en la interacció amb el sistema, ajudant els usuaris a identificar i abordar ràpidament els problemes. Això millorarà la seva experiència d'ús i facilitarà la resolució dels errors de forma eficient.

Variacions de colors

Negre i Blanc:

Dins de la nostra paleta de colors, tenim les següents variacions de negre i blanc:

```
$neutral_darkest: #0e0e0e
$neutral_darker: #1f1f1f
$neutral_dark: #3e3e3e
$neutral_origin: #8f8f8f
$neutral_light: #d8d8d8
$neutral_lighter: #fbfbfd
```

Aquests colors proporcionen una gradació de tonalitats des del negre més fosc fins al blanc més

clar. Són útils per crear contrastos i definir diferents nivells de profunditat en la nostra interfície.

Taronja:

Basant-nos en el color taronja base "#ff5208", tenim les següents variacions:

```
$primary_dark: #7e2a06
$primary_origin: #ff5208
$primary_light: #ff9680
$primary_lighter: #ffd5cc
```

Aquests colors taronja ens permeten jugar amb diverses intensitats i tonalitats, oferint una gamma de colors càlids i enèrgics que es poden utilitzar per destacar elements importants de la interfície.

Blau:

Partint del color blau base "#43a1cc", tenim les següents variacions:

```
$secondary_darker: #175977
$secondary_dark: #3985a9
$secondary_origin: #43a1cc
$secondary_light: #6cc1e9
$secondary_lighter: #abcddc
```

Aquests tons de blau ens proporcionen una paleta diversa per ressaltar elements i crear un aspecte elegant i refrescant a la interfície.

Verd Lima/Pi:

Basant-nos en el color verd base "#67ce74", tenim les següents variacions:

```
$succes_lighter: #e2ebe0
$succes_light: #b3ffbd
$succes_origin: #67ce74
$succes_dark: #638a5c
$succes_darker: #425c3d
```

Aquestes variacions de verd ens permeten destacar elements relacionats amb èxits, aportant una sensació de frescor i confiança a la interfície.

Vermell:

Partint del color vermell base "#f43e3e", tenim les següents variacions:

```
$error_lighter: #fdc3c9
$error_light: #fc9fa8
```

```
$error_origin: #f43e3e
$error_dark: #d90d0d
$error_darker: #780707
```

Aquestes tonalitats de vermell són utilitzades per ressaltar missatges d'error, creant un efecte d'alerta i captant l'atenció dels usuaris.

A través d'aquesta diversitat de colors, podem crear una interfície rica en matisos i contrastos, afegint profunditat i personalitat a la nostra aplicació.

Conflictes i errors

Unresolved directive in <stdin> - include::estils/conflicts.adoc[]

Confecció del manual d'instal·lació/distribució de l'aplicació.

Manual d'Instal·lació: Configuració de l'Aplicació

Aquest manual d'instal·lació descriu els passos necessaris per configurar correctament l'aplicació. Sigueu atents i seguiu les instruccions de manera precisa per assegurar-vos que tot funciona adequadament.

Pas 1: Configuració de la Base de Dades

Els dos projectes Laravel, l'API i el lloc web, requereixen una configuració adequada per connectar-se correctament a la base de dades. Assegureu-vos d'incloure les següents configuracions en els arxius de configuració corresponents:

Copieu els dos arxius de configuracions d'exemples (.env.example) als dos projectes i enganxeu-los al mateix repositori al qual pertanyen. A continuació, elimineu la part '.example' del nom de l'arxiu.

Un cop realitzades aquestes modificacions, el vostre arxiu de configuració .env hauria de tenir l'estructura següent:

```
DB_CONNECTION=mysql
DB_HOST=172.16.50.50
DB_PORT=3306
DB_DATABASE=marketifyBBDD
DB_USERNAME=admin
DB_PASSWORD=1234
```

Configuració específica del lloc web

Confirmeu que l'arxiu .env del lloc web ha de tenir la següent línia de codi per establir la IP de l'API:

```
API_IP=https://172.16.50.60:443/
```

Configuració de l'enviament d'emails

Per a la configuració dels emails, trobeu la següent línia a l'arxiu .env (del site) i substituïu-la amb la vostra configuració de sistema d'enviament d'emails:

```
MAIL_MAILER=smtp
MAIL_HOST=sandbox.smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=155ea25db1b4d7
MAIL_PASSWORD=74e21308539aea
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="${APP_NAME}"
```

Pas 2:

Per instal·lar les dependències dels dos projectes Laravel (site i ApiMarketplace), seguiu les següents instruccions:

- Obriu una nova consola i navegueu fins al directori del projecte "site" utilitzant la comanda següent:

```
cd site
```

- Un cop esteu dins del directori del projecte "site", executeu la comanda següent per instal·lar les dependències utilitzant Composer:

```
composer install
```

- Un cop finalitzat, executeu aquesta comanda per generar una clau de Laravel:

```
php artisan key:generate
```

- Tanqueu aquesta terminal i obriu una altra consola nova i navegueu fins al directori del projecte "ApiMarketplace" utilitzant la comanda següent:

```
cd ApiMarketplace
```

- Un cop esteu dins del directori del projecte "ApiMarketplace", executeu la comanda següent per instal·lar les dependències utilitzant Composer:

```
composer install
```

- Ja podeu tancar aquesta terminal.

Pas 3: Creació de la Màquina Virtual per al Lloc Web

Ara que la connexió està configurada, heu de crear una màquina virtual per al lloc web. Seguiu aquests passos:

1. Obriu una nova terminal.
2. Executeu les comandes següents:

```
cd vagrantSITE
vagrant up
vagrant ssh
```

Generació del Certificat HTTPS

Quan se us demani una contrasenya, introduïu '1234'. A continuació, executeu les següents comandes per generar un certificat autofirmat.

- Generar una nova clau privada encriptada:

```
sudo openssl genpkey -algorithm RSA -out /etc/ssl/private/selfsigned.key -aes128
```

- Generar un certificat autofirmat amb la nova clau privada:

```
sudo openssl req -new -x509 -sha256 -key /etc/ssl/private/selfsigned.key -out
/etc/ssl/certs/selfsigned.crt -days 3650 -subj
"/C=ES/ST=Barelona/L=Terrassa/O=Marketify/OU=NicolauCopernic/CN=Marketify"
```

- Reiniciar Apache

```
sudo systemctl restart apache2
```

- No tanqueu la terminal. Serà útil per a més endavant.

Pas 5: Creació de la Màquina Virtual per a l'API d'Imatges

Ara heu de crear una màquina virtual per a l'API d'imatges. Seguiu aquests passos:

1. Obriu una nova terminal.
2. Executeu les comandes següents:

```
cd vagrantAPI
vagrant up
vagrant ssh
```

Generació del Certificat HTTPS Quan se us demani una contrasenya, introduïu '1234'. A continuació, executeu les comandes següents:

- Generar una nova clau privada encriptada:

```
sudo openssl genpkey -algorithm RSA -out /etc/ssl/private/selfsigned.key -aes128
```

- Generar un certificat autofirmat amb la nova clau privada:

```
sudo openssl req -new -x509 -sha256 -key /etc/ssl/private/selfsigned.key -out
/etc/ssl/certs/selfsigned.crt -days 3650 -subj
"/C=ES/ST=Barelona/L=Terrassa/O=Marketify/OU=NicolauCopernic/CN=Marketify"
```

- Reiniciar Apache

```
sudo systemctl restart apache2
```

- No tanqueu la terminal. Serà útil per a més endavant.

Ja heu generat els dos certificats HTTPS necessaris per al funcionament de l'aplicació.

Pas 4: Execució de les Migracions i Poblament de la Base de Dades

Ara, aneu a la terminal on heu tractat el vagrant del lloc web (vagrantSITE) i executeu les comandes següents:

```
cd /home/marketify/site
php artisan migrate:refresh
```

Torneu a la terminal on heu tractat el vagrant de l'API (vagrantAPI) i executeu les comandes següents:

```
cd /home/marketify/ApiMarketplace
php artisan migrate:refresh
```

Torneu a la terminal vagrantSITE (estant en /home/marketify/site) i executeu:

```
php artisan db:seed
```

Pas 5: Instal·lació del DomPDF per a poder descarregar comandes

Un cop fetes les Migracions i Poblament de la Base de Dades, genereu una nova terminal i executeu le següent comanda:

```
cd site
```

-Instalar el composer del DomPDF

```
composer require barryvdh/laravel-dompdf
```

Ara ja podeu descarregar les comandes en PDF!

Pas 6: Finalització de la Configuració

Felicitats! Heu acabat amb èxit la configuració. Ara, obriu el navegador i introduïu la següent adreça IP per accedir al lloc web:

https://172.16.50.50

Confecció del manual d'usuari.

Línies futures

Línies futures

En aquest apartat volem explicar les diferents funcionalitats que voliem fer però per falta de temps no hem pogut.

- Arreglar el problema de que es pugui enviar el formulari de edició i afegir prodcutes buit.

- Les imatges del producte només s'actualizen en el detall de producte, falta que s'actualitzin a /search i a las botigues.
- La API no guarda imatges a la API sinó que les guarda al site del projecte.
- Es guarda la imatge principal del producte encara que es canvii.
- Totes les imatges que no son de productes(fotos de perfil o de la botiga) funcionen sense la API.
- Falta fer la petició per a borrar una imatge, quan l'usuari vol esborrar un producte.També quan es canvien les imatges d'un producte.

Conclusions. Desviacions en la planificació. Aportacions del projecte als coneixements de l'alumne.

Conclusions

Com a conclusions finals, treiem que nosaltres creiem que hem treballat molt bé, sempre ha hagut un bon ambient i a mesura que avançava el projecte hi havia més encara.

Creiem que ens hem organitzat força bé, ja que a excepció de l'ultima setmana, hem pogut arribar a tot el que ens haviem proposat fer. Si que es veritat, que creiem que a vegadas ens ha faltat comunicació a l'hora d'arribar a totes les tasques proposades, però en línies generals, ens hem organitzat bé.Ens haguès agradat tenir una mica de temps més, per a poder revisar el projecte més profundament i poder arreglar els bugs pertinents, però tot i així estem satisfets amb el treball fet.

Creiem que a mesura que ha anat avançant el projecte, ens hem fet més forts davant les adversitats que ens afrontaven, per molt que no ens surtís alguna cosa ho intentavam fins a aconseguir-ho.

Com a últim dessitg, volem agraïr tant al professorat per ajudar-nos sempre que han pogut, com a cada membre del nostre grup per facilitar un bon ambient de treball i formar un bon equip.

Glossari. Webgrafia. Altres recursos de consulta.

Funció getCookie in cartFunctions.js

https://www.w3schools.com/js/js_cookies.asp

Presentació del projecte.