Instruções de instalação

Primeiramente realize o clone do projeto através do git clone.

Após fazer o clone do projeto abra a para raiz no vscode ou sua IDE de preferência.

Dentro da pasta raiz utilize cd para acessar a pasta backend
“cd backend/.”
Dentro da pasta backend utilize o
“composer install”.
Após instalar as dependências do laravel utilize o
“npm run dev”
para instalar as dependências dos módulos.
Com as dependências instaladas instale a chave key através do comando
“php artisan key:generate”.

Com as configurações base pronta vá até o arquivo **.env** e coloque a estrutura do banco de dados.

após configurar o banco de dados pelo arquivo **.env** suba as migrations para criar as tabelas no banco com o comando
“php artisan migrate”

Utilize o comando
“php artisan storage:link”
para criar a rota do storage e armazenar as imagens

(OPCIONAL) Com as tabelas criadas eu implementei dois seeders para popular o banco de dados e facilitar no teste para popular as tabelas através dos siders utilize o comando
“php artisan db:seed”.
Após configurar tudo utilize o comando
“php artisan serve” .

- Após configurar o backend vamos configurar o frontend.

Na raiz do projeto vá a pasta frontend com o comando
“cd frontend/”

dentro da pasta frontend utilize o comando
“npm install”

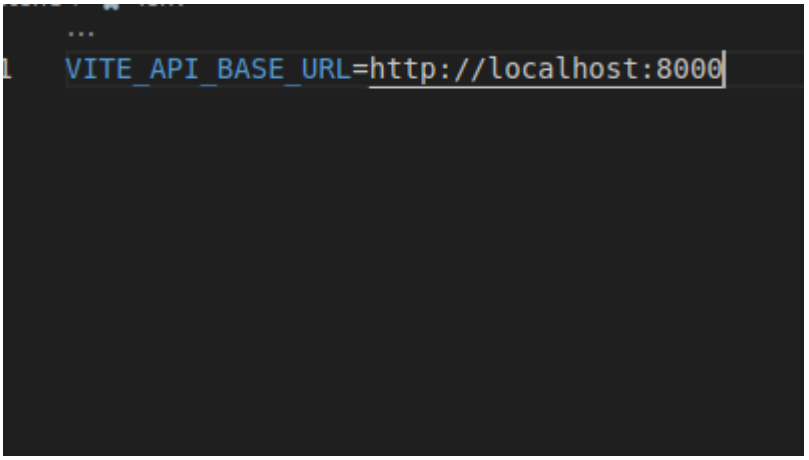
após instalar as dependências do vue aplique o comando
“npm run dev”

(OBS: os comandos **“php artisan serve”** e **“npm run dev”** devem ser feitos em terminais diferentes, então sugiro que caso utilize o vscode utilize 3 terminais 2 para iniciar o servidor e uma para instalar as dependências do projeto.

Após iniciar os dois servidores confira os arquivos **.env** da pasta frontentd

ATENÇÃO!!! Só modifique as estampas seguintes se sua porta (após o localhost) for diferente de 8000

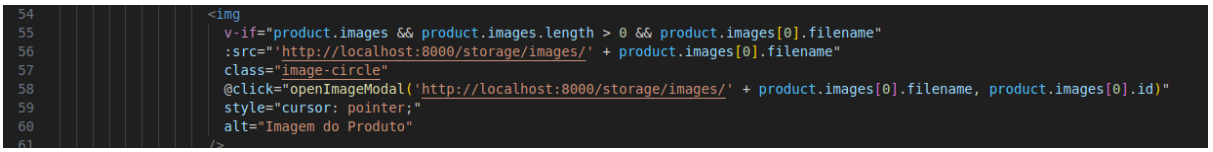
está dentro da pasta **frontend/.env**
a modifique a url base:



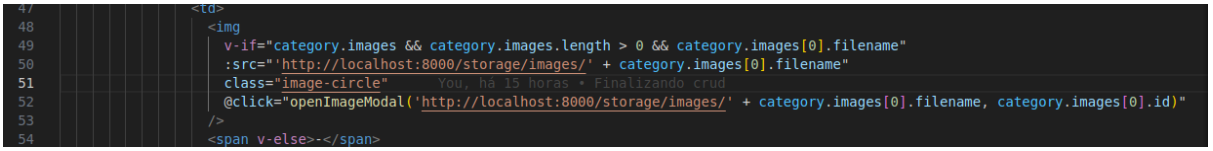
a url base a criar no php artisan serve, geralmente é
http://localhost:8000
depois que modificar vá para pasta:
src/axios.js

lá também deve configurar a rota base para o sistema

após configurar essas duas rotas base configure mais duas que são as últimas (obs nao deu tempo de criar o arquivo base da url da imagem)
uma fica na pasta src/components/Producs/ProductsList.vue



outra fica na pasta src/Categories/CategoriesList.vue



pronto apos toda essa configuração de funcionar normalmente.

ROTA PARA TESTE
criei uma rota para testar as api do servidor com banco de dados e etc:
http://localhost:5173/aBcDeFgHiJkLmNoPqRsTuVwXyZ

Descrição das funcionalidades implementadas.
Funcionalidades, a api faz um CRUD Padrão inserção atualização, visualização e delete dos dados. Na página “/” vai aparecer um Dashboard com estatísticas de produtos e categorias onde o cliente pode realizar o download em pdf.

- Na tela de produtos, vai aparecer o modal onde o cliente pode criar um novo produto ou atualizar. Vai haver uma lista com todos os produtos e imagens. O cliente pode fazer o download da imagem caso queira: para fazer o download clique na imagem e vai abrir um modal no modal clique em download.

- Na tela de categorias as funcionalidades são as mesmas para produtos.

- Foi aplicado Middlewares para verificar se não está havendo ataque em massa, caso aja ele bloqueia o IP

- Também foi adicionado uma tratativa que nao esta em uso mas pode ativar descomentando a parte do codigo que é caso o cliente erre a senha mais de 5 vezes o sistema bloqueia o acesso do ip do dispotisito.

Possíveis melhorias futura

No futuro pode-se reaproveitar o código para criar um controle de estoque dinâmico, mostrando a quantidade de produtos que saíram e entraram, saldo dos produtos, quantidade x valores totais dos produtos em estoque, quantos saíram x quantos entraram e etc…
-> Funcionalidades: utilizar cookie para aumentar a segurança do projeto, aplicar middlewares de segurança como a desativação do servidor de forma dinâmica e avisando o gesto através de uma api do

Telegram, Aplicar testes automatizados em massa para simular ataques hackers e afins.

Tecnologias utilizadas e justificativas das escolhas técnicas.

Foi utilizado o Laravel versão 9 junto com o vue 3

-> Laravel 9 é considerado a versão mais estável do framework com os melhores componentes para se trabalhar.

-> Vue 3 escolhendo a versão padrão e mais atualizada para trabalhar no frontend

-> Foi utilizado um tradutor para facilitar as respostas de erro e sucesso do JSON vindo do laravel como email já existente, senhas não confere e etc… referencia:
https://github.com/lucascudo/laravel-pt-BR-localization

-> Para o css foi utilizado o bootstrap pois é o mais fácil de manipular.