Stock-exchange

PROYECTO



INDICE :

1. – Introducción :
   1. Presentacion
   2. Objetivo
2. – Backend :
   1. Docker Composer
      1. Sail (Laravel con Docker)
      2. Contenedor del API (Laravel)
      3. Contenedor del BBDD (MySQL)
      4. Contenedor del Gestores de bases de datos (phpMyAdmin)
      5. Contenedor del Balanceador (Nginx)
      6. Contenedor del Generador de datos (Node)
   2. Generador de datos (Node)
   3. Servidor de API Web (Laravel)
3. – Frontend:
   1. Estructura de página (HTML)
   2. Lógica principal de página (JS/logic)
   3. Los fetch de los API
      1. Registro/Inicio de sesión (Autorozition)
      2. Saco de datos (fetch-data)
4. – Introducción:
   1. Presentación

"Bienvenidos al proyecto de desarrollo web para la bolsa de valores. Este proyecto tiene como objetivo crear una plataforma en línea que permita a los usuarios acceder a información actualizada sobre el rendimiento de las acciones y la bolsa de valores en general. Con la plataforma, los usuarios podrán realizar seguimiento de los valores de varias bolsas y tomar decisiones informadas en tiempo real. Este proyecto combinará la tecnología más avanzada (Docker, Laravel, Nginx …) con una interfaz fácil de usar para brindar una experiencia única y eficiente para los inversores."



1. – Introducción:
   1. Objetivo

El objetivo principal de este proyecto es desarrollar una plataforma web que permita a los usuarios monitorear en tiempo real las cotizaciones de las 10 empresas más importantes de la bolsa IBEX35, incluyendo INDITEX, BANCO DE SANTANDER, BBVA, NATURGY, CELLNEX, CAIXABANK, TELEFÓNICA, REPSOL, FERROVIAL y IBERDROLA.

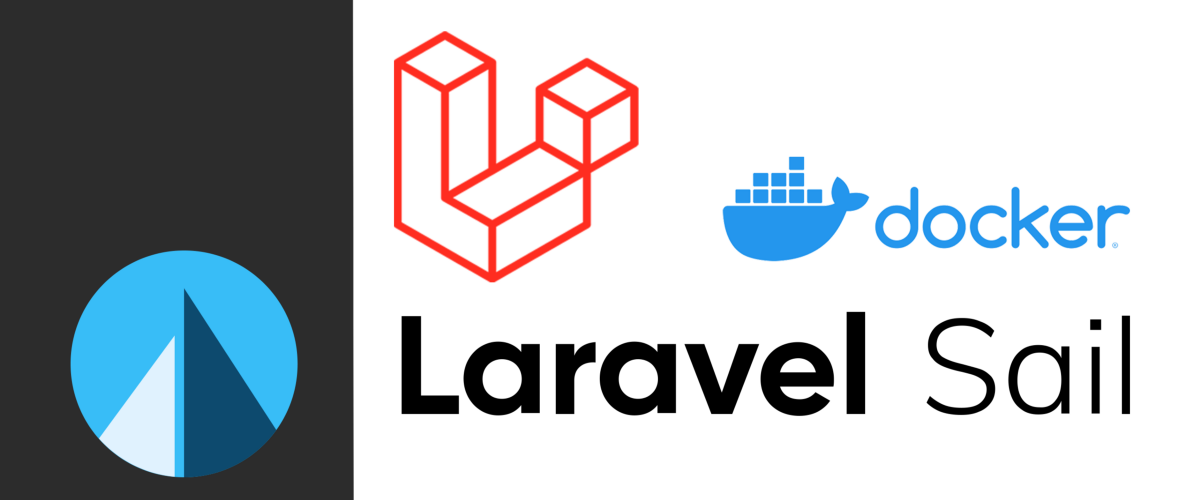
El objetivo adicional de este proyecto es desarrollar un panel de monitoreo de cotizaciones en tiempo real para las empresas seleccionadas por el usuario. La configuración se guardará en Local Storage para su uso futuro y los datos se actualizarán periódicamente a través de fetch y APIs diseñadas. Además, se proporcionará acceso a gráficos de cotizaciones a través de un botón en el panel.

El proyecto constará de un backend con un proceso en segundo plano que generará datos aleatorios y los expondrá a través de una API REST, y un frontend con la funcionalidad descrita anteriormente. Además, la aplicación se contenedorizará y desplegará mediante Docker con un balanceador que haga de Gateway para la API REST y, de ser posible, también actúe como proxy inverso para el frontend. La aplicación estará basada en MySQL y contará con datos de prueba para todo el año para los gráficos.

1. – Backend:
   1. Docker Composer

2.1.1 Sail

Para contenezar todas las tecnologías que nos vamos a usar en el proyecto. Para simplificar el proceso de contenedorización de todas las tecnologías que se utilizarán, se puede utilizar Laravel con Docker (Sail), que es una herramienta de contenedorización y orquestación de aplicaciones basadas en Laravel. Con Sail, se pueden añadir fácilmente todos los servicios requeridos para el proyecto y gestionar su ejecución de forma sencilla y eficiente. Sail proporciona una imagen de Docker que contiene todo lo necesario para ejecutar una aplicación Laravel, incluyendo PHP, Nginx, MySQL, Redis y otros servicios relacionados. Con Sail, se puede fácilmente crear y desplegar aplicaciones Laravel en producción de manera rápida y sencilla.

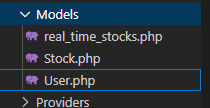


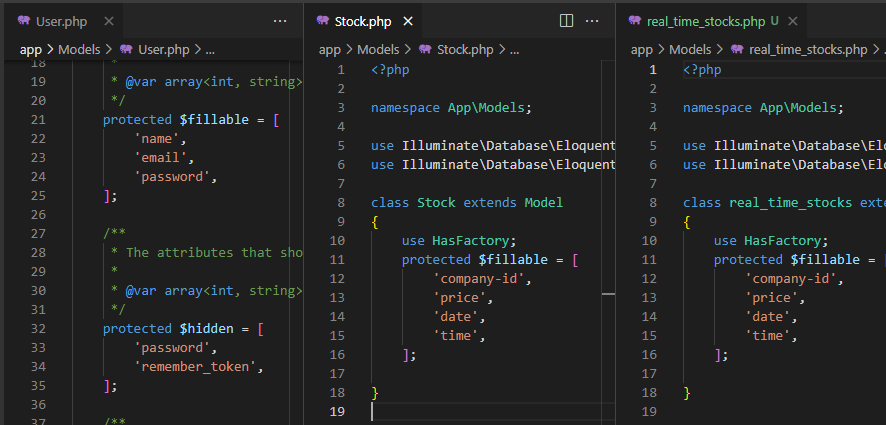
2.0– Backend:

* 1. Docker Composer

2.1.2 API

El API desarrollado con Laravel y conrenezado con sail en un Docker composer:

Los Modelos del Api:



2.0– Backend:

* 1. Docker Composer

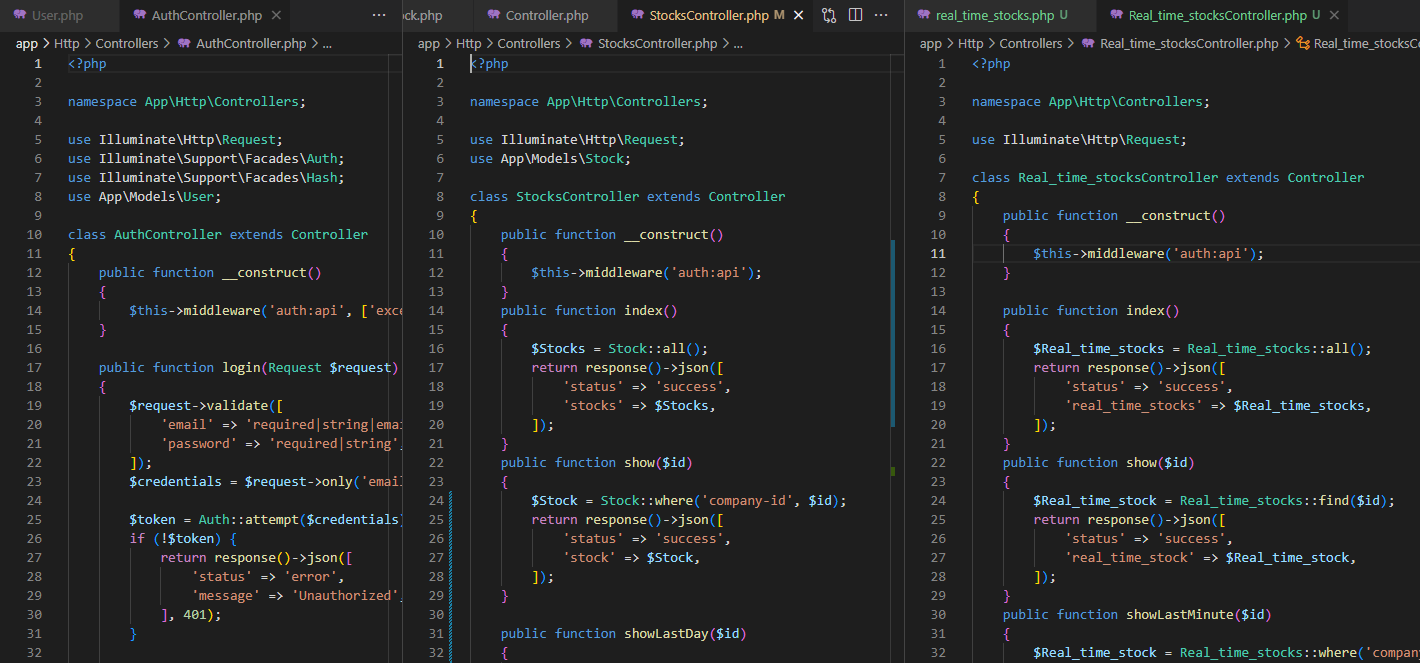
2.1.2 API

En el Api hay tres Controladores principales que controlan todas las Apis

- AuthenticateController

- StocksController

- Real\_time\_StocksController



2.0– Backend:

* 1. Docker Composer

2.1.2 API

Se ha creado varios Apis para todas las necesidades y las pruebas que se puede enfrontar en el proyecto

