

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему «Розробка системи електронної комерції
за допомогою Next.js (TypeScript) та Axum (Rust)»

Виконав: Маслов Нікіта Євгенович

Куратор: Корнійчук Максим Анатолійович

Задача

Створити сучасний інтернет-магазин для поширення та продажу приладів альтернативної медицини.

Аудиторія — дорослі та літні люди

Інтуїтивний інтерфейс
та допоміжні елементи

Інтерактивність та плавність

SEO

Доступ до всіх замовлень
та повідомлення про нові
замовлення

Інтернаціоналізація



<Кі нець>



Це тільки відкриваючий тег

Технології

**Інтуїтивний
інтерфейс та
допоміжні елементи**
Figma, Photoshop

**Інтерактивність
та плавність**
SPA, React

SEO

Next.js, SSR, SSG,
оптимізація зображень

**Доступ до всіх
замовлень
та повідомлення
про нові замовлення**

MongoDB, MongoDB
Compass, повідомлення
через Telegram-бота

**Інтернаціоналізаці
я**

Інтернаціоналізована
маршрутизація Next.js,
next-i18next

**Підходи та
архітектури**

Mobile First, Гібридна
верстка (адаптивна +
відгукова), MVC, слабке
зв'язування

Надійність

TypeScript, Rust (Axum)

Етапи розробки

1. Створення дизайну застосунку.
2. Визначення моделей даних.
3. Створення REST API інтерфейсу взаємодії клієнта та сервера.
4. Реалізація фронтенд частини.
5. Реалізація бекенд частини.
6. Поєднання фронтенд і бекенд.
7. Тестування та зневадження.

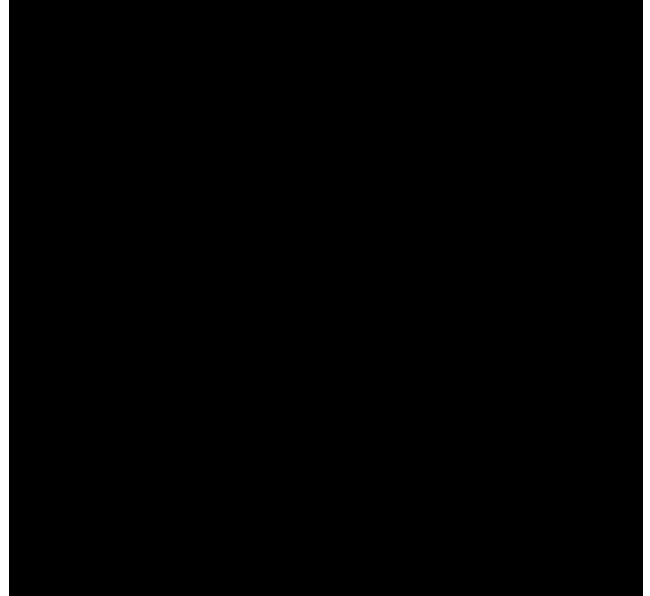
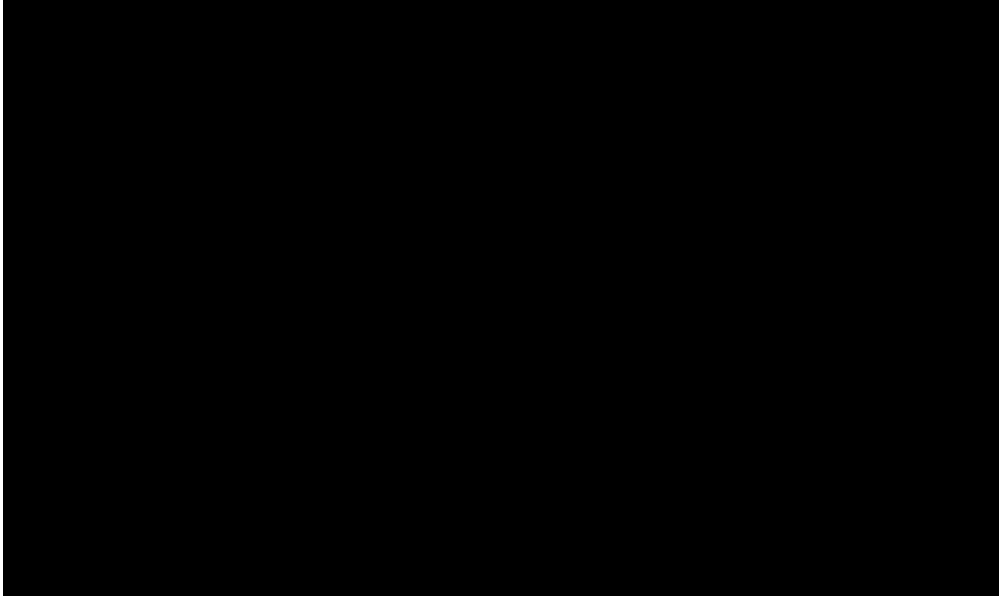
Створення дизайну застосунку

Етап розробки дизайну застосунку включив у себе наступні задачі:

- створення структури сайту,
- підготовка текстових матеріалів,
- фотозйомка приладів,
- робота над фотографіями у Photoshop,
- розробка макетів сторінок у Figma.



UI & UX. Інтуїтивний інтерфейс та допоміжні елементи



Створення моделей даних

У результаті цього етапу було створено моделі даних для двох сутностей:

Product

+

Users?
Order

Авторизація

?



Тільки пришвидшення
оформлення
замовлення



Один прилад на все
життя



~~Авторизація~~

я



User живе в Order

Модель Order

- id
 - creation_time
 - first_name
 - last_name
 - email
 - phone_number
 - ordered_products
 - product_id
 - quantity
 - delivery
 - novaposhta
 - city
 - warehouse
 - courier
 - city
 - address
 - zip
 - worldwide
 - country
 - city
 - address
 - zip
- або
- або

Модель Product

- id
- name
- descr
- img
 - path
 - width
 - height
- price
- discount_info

REST API

Створення замовлення (POST /orders):

```
{
  "firstName": "Hikita",
  "lastName": "Маслов",
  "email": "maslov.n.e@gmail.com",
  "phoneNumber": "+380 (67) 538 47 98",
  "delivery": {
    "kind": "worldwide",
    "data": {
      "country": "Poland",
      "city": "Варшава",
      "address": "варшавская 97",
      "zip": "98789"
    }
  },
  "orderedProducts": [
    {
      "productId": "645f6f60acd5793eddfa8ae9",
      "quantity": 5
    },
    {
      "productId": "645f705facd5793eddfa8aec",
      "quantity": 3
    }
  ]
}
```

Отримання продукту

(GET /products/:product_id):

```
{
  "_id": {
    "$oid": "645f6f60acd5793eddfa8ae9"
  },
  "name": "KGS",
  "descr": {
    "uk": "Середній за вартістю варіант.",
    "en": "An average cost option.",
    "ru": "Средний по стоимости вариант."
  },
  "img": {
    "path": "/images/kgs-kit.jpg",
    "width": 3436,
    "height": 1730
  },
  "price": 4900.0
}
```

REST API

Отримання всіх існуючих продуктів
(GET /products):

```

{
  "products": [
    {
      "_id": {
        "$oid": "645f6f60acd5793eddfa8ae9"
      },
      "name": "KGS",
      "descr": {
        "uk": "Середній за вартістю варіант.",
        "en": "An average cost option.",
        "ru": "Средний по стоимости вариант."
      },
      "img": {
        "path": "/images/kgs-kit.jpg",
        "width": 3436,
        "height": 1730
      },
      "price": 4900.0
    },
    {
      "_id": {
        "$oid": "645f705facd5793eddfa8aec"
      },
      "name": "KGS-MINI",

```

Не знайшов, чим
заповнити цей
простір

Структура фронтенду

- public
 - images
 - locales
- src
 - api
 - components
 - elements
 - modules
 - templates
 - layouts
- src
 - constants
 - context
 - hooks
 - styles
 - types
 - utils

Особливості верстки

Mobile First

Моральна сторона

+

SEO

Гібридна верстка

Адаптивна (adaptive)

+

Відгукова (responsive)

Центрування між текстовими блоками

The image shows a screenshot of a web page with a header and a main content area. The header includes a logo for 'VORTEX' and a shopping cart icon with a '1' notification. The main content area features a blue highlighted text block on the left and a grey text block on the right, connected by a right-pointing arrow. An accessibility inspector is open over the blue text block, displaying its style and accessibility properties.

Header:

- VORTEX logo
- Shopping cart icon with '1' notification
- Menu icon

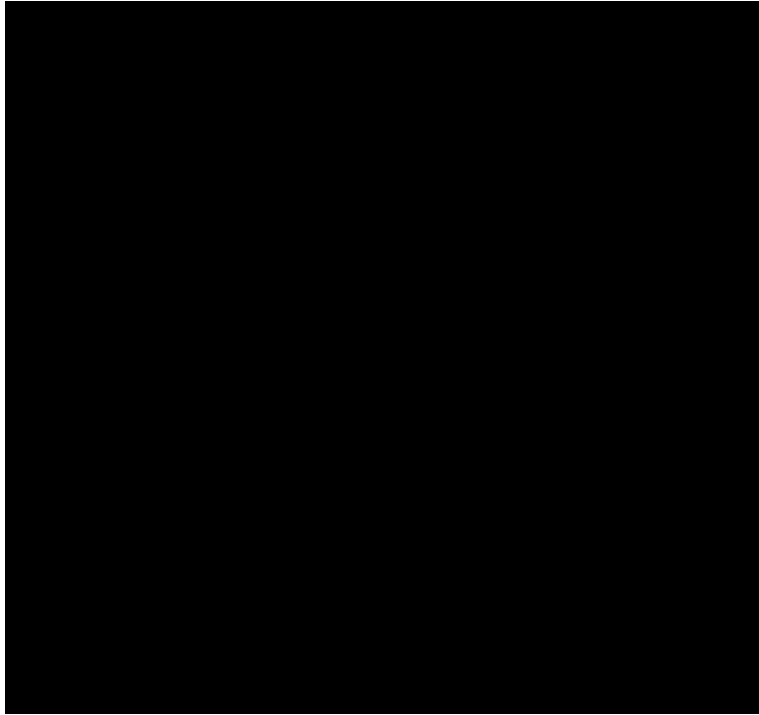
Text Blocks:

- Left block (highlighted in blue): Котушка Мішина розбиває чужорідні структури
- Right block (grey): Організм самостійно виводить залишки від захворювань

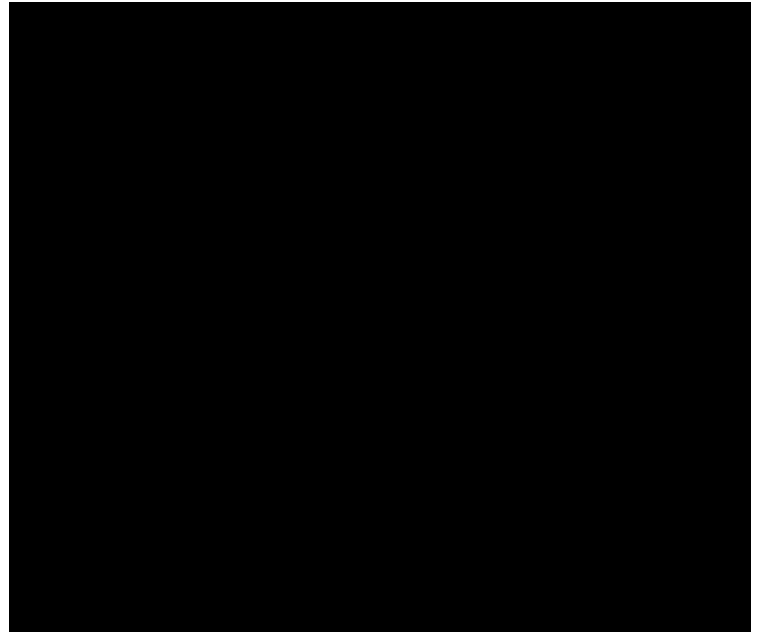
Accessibility Inspector (over the blue text):

- Element: `span` (238.13 x 90)
- Color: #2D2E30
- Font: 20px __Noto_Sans_772f7d, __Noto_Sa...
- ACCESSIBILITY: Contrast 13.59 (checked), Name, Role generic, Keyboard-focusable (unchecked)

Інтеграція бібліотечних компонентів



MUI



Notistack

Оптимізація зображень

```
<div className={styles.stepOne}>
  <Image
    src="/images/plug-into.jpg"
    alt="plug in the sinus generator"
    width={681}
    height={681}
    sizes="100vw,
      (min-width: 700px) 50vw,
      (min-width: 1200px) 417px"
    quality={100}
  />
  <p className={styles.imgCaption}>
    1. Увімкнути генератор синуса в&nbsp;
  </p>
</div>
<div className={styles.stepTwo}>
```

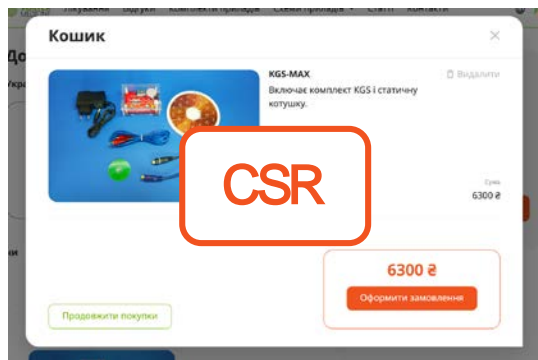
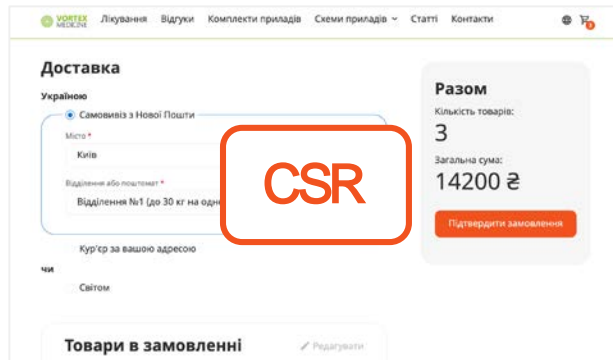
```

```

Компонент Image з Next.js

- Оптимізація розміру
- Сучасні формати, такі як WebP і AVIF
- Запобігання зсуву макета (layout shift)
- Ліниве завантаження (lazy loading)

Види рендерингу (CSR, SSR, SSG)



Рідко з'являються нові продукти => SSG

Сторінка оформлення замовлення не індексується => CSR

Немає глобального `getStaticProps` + вміст кошика не індексується => CSR

Local storage


Додані до кошика товари зберігаються на клієнті в local storage, щоб не втратити дані між браузерними сесіями.

Інтернаціоналізація. Next-i18next

- public
 - locales
 - en
 - common.json
 - home.json
 - uk
 - common.json
 - home.json
 - ru
 - common.json
 - home.json

```
module.exports = {  
  i18n: {  
    defaultLocale: "en",  
    locales: ["uk", "en", "ru"],  
  },  
};
```

next-i18next.config.js

```
no usages  Nikita Maslov  
export default appWithTranslation(App);
```

/pages/_app.tsx

Інтернаціоналізація. Next-i18next

```
{
  "intro": {
    "heading": "Котушка Мішина\u00A0\u2013\u00A0ваш домашній лікар",
    "subheading": "Прилад для\u00A0лікування більшості захворювань у\u00A0домашніх умовах",
    "textBlock1": "Котушка Мішина допомагає позбутися гіпертонії, хвороб нирок, аритмії, ВСД, ангіни та\u00A0",
    "textBlock2": "Технологія має 6 основних переваг: немає побічних ефектів; лікує більшість захворювань;
    "textBlock3": {
      "mainText": "Ми розповімо про\u00A0те, як\u00A0працює котушка Мішина, не\u00A0залазячи глибоко у\u00A0",
      "linkText": "7-годинний вебінар з\u00A0вихрової медицини"
    }
  }
}
```

/uk/home.json

Інтернаціоналізація. Next-i18next

```
import DefaultLayout from "@components/layouts/Default";
import HomePage from "@components/templates/HomePage";
import { serverSideTranslations } from "next-i18next/serverSideTranslations";
```

no usages Nikita Maslov

```
export default function Home(): JSX.Element {
  return (
    <DefaultLayout>
      <HomePage />
    </DefaultLayout>
  );
}
```

no usages Nikita Maslov

```
export async function getStaticProps({ locale }: { locale: string }) {
  return {
    props: {
      ...(await serverSideTranslations(locale, namespacesRequired: ["common", "home"])),
    },
  };
}
```

```
function Intro(): JSX.Element {
  const { t : TFunction<"translation", undef... } = useTranslation();

  return (
    <section className={styles.intro}>
      <IntroImage />
      <Container>
        <h1 className={styles.heading}>{t( key: "home:intro.heading")}</h1>
        <p className={styles.subheading}>{t( key: "home:intro.subheading")}</p>
        <div className={styles.mainContentWrapper}>
          <div className={styles.textBlock}>
            <p>{t( key: "home:intro.textBlock1")}</p>
            <p>{t( key: "home:intro.textBlock2")}</p>
            <p>
              {t( key: "home:intro.textBlock3.mainText")}{" "}
            </p>
          </div>
        </div>
      </Container>
    </section>
  );
}
```

Перший блок на головній сторінці

Компонент головної сторінки

Інтернаціоналізація. Перемикач мов

```
const router : NextRouter = useRouter();

const switchToLocale = useCallback(
  callback: (locale: string) => {
    const path : string = router.asPath;

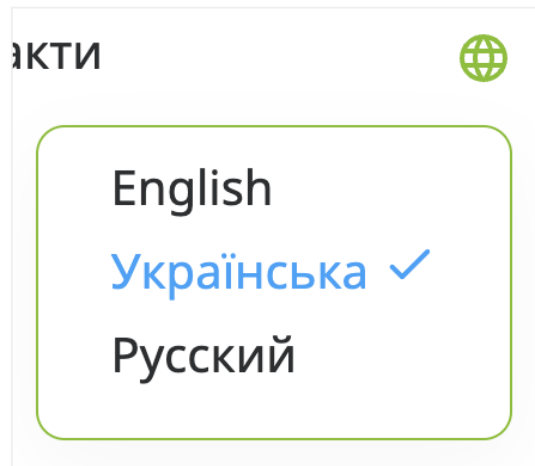
    return router.push(path, path, options: { locale, scroll: false });
  },
  deps: [router]
);

const languageChanged = useCallback(
  callback: async (option: LanguageOption) : Promise<void> => {
    setValue(option);

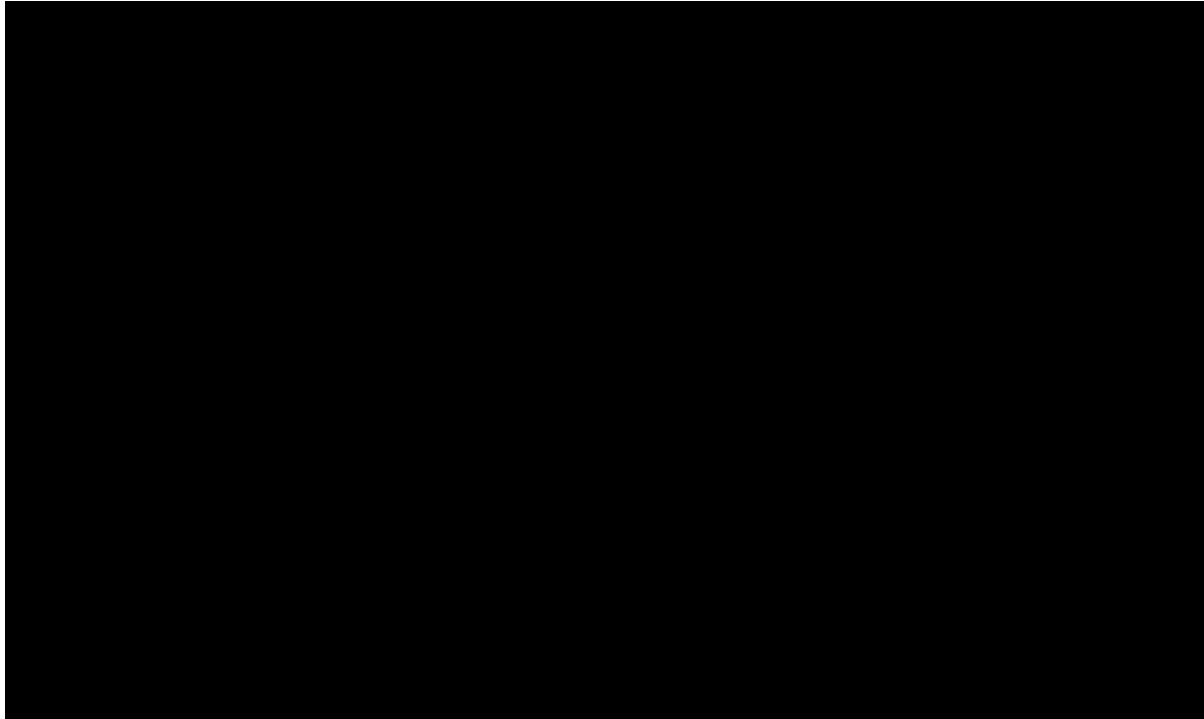
    const locale : string = option.value;

    await switchToLocale(locale);
  },
  deps: [switchToLocale]
);
```

Компонент перемикача мов

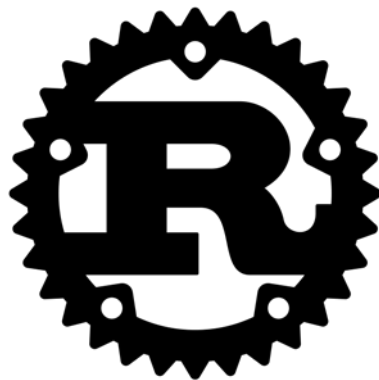


Інтернаціоналізація. Демонстрація



Бек енд

- src
 - controllers
 - models
 - repositories
 - utils
 - constants



Axum



mongoDB®

Слабке зв'язування

```
#[derive(Debug, Clone)]
pub struct MongoRepository {
    orders: Collection<Order>,
    products: Collection<Product>,
}

impl MongoRepository {
    pub async fn init() -> Self {
        let uri: String = dotenv::var(<key> MONGOURI).unwrap();

        let client: Client = Client::with_uri_str(uri).await.unwrap();
        let db: Database = client.database(<name> DB_NAME);
        let orders: Collection<Order> = db.collection:<Order>(<name> ORDERS_COLLECTION);
        let products: Collection<Product> = db.collection:<Product>(<name> PRODUCTS_COLLECTION);

        Self { orders, products }
    }
}

#[async_trait]
impl OrderRepository for MongoRepository {
    type Order = Order;
    type CreateOrderResult = FullOrder;
    async fn create_order(&self, order: Self::Order) -> Result<Self::CreateOrderResult> {...}
}

#[async_trait]
impl ProductRepository for MongoRepository {
    type Product = Product;
    async fn get_all_products(&self) -> Result<Vec<Self::Product>> {...}
    async fn get_product_by_id(&self, product_id: &str) -> Result<Option<Self::Product>> {...}
}
```

repositories/mongo.rs

```
#[async_trait]
pub trait OrderRepository {
    type Order;
    type CreateOrderResult;

    async fn create_order(&self, order: Self::Order) -> Result<Self::CreateOrderResult>;
}
```

repositories/order.rs

```
#[async_trait]
pub trait ProductRepository {
    type Product;

    async fn get_all_products(&self) -> Result<Vec<Self::Product>>;
    async fn get_product_by_id(&self, product_id: &str) -> Result<Option<Self::Product>>;
}
```

repositories/product.rs

```
pub async fn create_order<R, O>(
    State(db): State<Arc<dyn OrderRepository<CreateOrderResult = R, Order = O>>>,
    Json(order): Json<O>,
) -> impl IntoResponse {
```

controllers/order.rs

Слабке зв'язування

```
let db : Arc<MongoRepository> = Arc::new( data: MongoRepository::init().await);

let app : Router = Router::new()
    .route( path: "/orders", method_router: post( handler: create_order)) : Router<Arc<MongoRepository>, Body>
    .route( path: "/products", method_router: get( handler: get_all_products)) : Router<Arc<MongoRepository>, Body>
    .route( path: "/products/:product_id", method_router: get( handler: get_product_by_id)) : Router<Arc<MongoRepository>, Body>
    .layer(
        CorsLayer::new()
            .allow_origin( origin: "http://localhost:3001".parse:::<HeaderValue>().unwrap()) : CorsLayer
            .allow_methods( methods: [Method::GET, Method::POST]) : CorsLayer
            .allow_headers( headers: [http::header::CONTENT_TYPE]),
    ) : Router<Arc<MongoRepository>, Body>
    .with_state( state: db);
```

main.rs

CORS

```
CorsLayer::new()  
  .allow_origin( origin: "http://localhost:3001".parse::<HeaderValue>().unwrap()) : CorsLayer  
  .allow_methods( methods: [Method::GET, Method::POST]) : CorsLayer  
  .allow_headers( headers: [http::header::CONTENT_TYPE]),
```

main.rs

Адмін-панель. MongoDB Compass

The screenshot displays the MongoDB Compass interface for the 'vortex-medicine.products' collection. The left sidebar shows the database structure with 'products' selected. The main area shows a list of documents with a filter bar and 'ADD DATA' and 'EXPORT COLLECTION' buttons. The right-hand pane shows the details of a selected document.

Database: vortex-medicine...
Collection: vortex-medicine.products
Documents: 3
Indexes: 1

Filter: Type a query: { field: 'value' }
Buttons: Reset, Find, More Options

Document 1:

```
{ "_id": ObjectId("645f6f60acd5793eddffa8e9"),  
  "name": "KGS",  
  "descr": Object,  
  "uk": "Середній за вапирце варіант.",  
  "en": "An average cost option.",  
  "ru": "Средний по стоимости вариант.",  
  "img": Object,  
  "path": "/images/kgs-kit.jpg",  
  "width": 3438,  
  "height": 1738,  
  "price": 4900 }
```

Document 2:

```
{ "_id": ObjectId("645f705facd5793eddffa8ec"),  
  "name": "KGS-MINI",  
  "descr": Object,  
  "uk": "Бюджетний варіант, простий у використанні, з мінімальною функціональніс...",  
  "en": "A budget option, easy to use, with minimal functionality.",  
  "ru": "Бюджетный вариант, простой в использовании, с минимальной функциональнос...",  
  "img": Object,  
  "path": "/images/kgs-mini-ktt.jpg",  
  "width": 4032,  
  "height": 2528,  
  "price": 3000 }
```

Document 3:

```
{ "_id": ObjectId("645f70ecacd5793eddffa8ed"),  
  "name": "KGS-MAX",  
  "descr": Object,  
  "uk": "Включач комплект KGS і статичну катушку.",  
  "en": "Includes the KGS kit and the static coil.",  
  "ru": "Включает комплект KGS и статическую катушку.",  
  "img": Object }
```

Selected Document Details:

```
{ "_id": ObjectId("645falb2e9a6174ef79e43ba"),  
  "firstName": "Hikita",  
  "lastName": "Maclov",  
  "email": "maslov.n.e@gmail.com",  
  "phoneNumber": "+380 (67) 538 47 98",  
  "delivery": Object,  
  "orderedProducts": Array }
```

Повідомлення про нове замовлення

```
pub async fn send_telegram_notification(message: &str, token: &str, chat_id: &str)
{
    let payload : Value = json!({
        "text": message,
        "chat_id": chat_id
    });

    let client : Client = request::Client::new();
    client
        .post( url: format!(
            "https://api.telegram.org/bot{token}/sendMessage",
            token = token
        ))
        .json(&payload)
        .send()
        .await?;

    Ok(())
}
```

```
New order: {
  "_id": {
    "$oid": "6460e3eea582fcd1cfe0809e"
  },
  "creationTime": "2023-05-14T13:36:45.790793Z",
  "firstName": "Ярема",
  "lastName": "Мединський",
  "email": "var@gmail.com",
  "phoneNumber": "+380 (67) 538 47 98",
  "delivery": {
    "kind": "novaposhta",
    "data": {
      "city": "Київ",
      "warehouse": "Відділення №145 (до 30 кг): вул. Героїв
Дніпра, 53"
    }
  },
  "orderedProducts": [
    {
      "product": {
        "id": "645f6f60acd5793eddfa8ae9",
        "name": "KGS",
        "price": 4900.0
      },
      "quantity": 3
    },
    {
      "product": {
        "id": "645f705facd5793eddfa8aec",
        "name": "KGS-MINI",
        "price": 3000.0
      },
      "quantity": 2
    }
  ]
}
```

</ Кі нець >