Tartalomjegyzék

Dr. Theiss Technikai dokumentum

Bevezetés

Előszó

A drtheiss.hu weboldal kialakításakor tudatosan választottuk a natív PHP 8.x verzióját, mellőzve a megszokott keretrendszereket. Ez az innovatív megközelítés számos előnnyel jár, melyek hatással vannak a teljes fejlesztési folyamatra.

A natív PHP lehetőséget teremt a szabad tervezésre és egyedi, személyre szabott megoldásokra, ugyanakkor elkerüli az előre meghatározott korlátok és szabályok szigorát. Ennek révén a fejlesztők kreatív felszabadultságban tervezhetik és valósíthatják meg projektenként az elképzeléseiket.

Az agilis fejlesztési környezetben kiemelt fontosságú a kód könnyű szerkeszthetősége és hozzáférhetősége. A natív PHP használatával nem kell komplex struktúrákkal vagy elrendezésekkel szembesülni, amelyek esetleg gátolhatnák a fejlesztők munkáját. Ezáltal lehetőség nyílik a gyors és hatékony kódmódosításokra, valamint az egyszerű hibakeresésre.

A PHP 8.x verzió számos újítást és optimalizációt hozott, melyeket a natív PHP teljes mértékben kihasznál. Ez lehetővé teszi a hatékony és gyors kódfejlesztést, amely javítja az alkalmazások teljesítményét és a felhasználói élményt.

A natív PHP rugalmas fejlesztést és a projekt specifikus igények kielégítését teszi lehetővé. A fejlesztők szabadon formálhatják az alkalmazás szerkezetét és működését, anélkül, hogy keretrendszerek szabályaihoz kötve lennének.

Végső soron elmondhatjuk, hogy a natív PHP jól használható rugalmassága, a kreatív kibontakozás lehetősége és a hatékony teljesítmény révén egyedülálló és innovatív projektek hozhatók létre, amelyek tökéletesen tükrözik a fejlesztői elképzeléseket. A weboldal célja a Dr. Theiss termékek rendezett rendszerezése és kezelése, ugyanakkor közvetlen értékesítésre nem használható. Fontos tudni, hogy Magyarországon az internetes gyógyszer értékesítés kizárólag azoknak engedélyezett, akik rendelkeznek patikai engedéllyel, és a tevékenységüket megkezdés előtt be kell jelenteniük az illetékes hatóságnak.

Ez a technikai leírás a lényeges elemeket szemlélteti. Ugyanakkor kiemelten fontos hangsúlyozni, hogy rajtunk kívül álló és engedély nélküli módosítások esetén nem vállalunk felelősséget az oldal vagy annak, tartalmának biztonságáért és integritásáért.

Technikai adatok

PHP

A PHP általános beállításai

PHP Version	8.1.20	
max_execution_time	180	180
max_file_uploads	20	20
max_input_nesting_level	64	64
max_input_time	180	180
max_input_vars	3000	3000
memory_limit	512M	512M
post_max_size	128M	128M
upload_max_filesize	128M	128M
upload_tmp_dir	/var/www/clients/client630/web1405/tmp	/var/www/clients/client630/web1405/tmp
cURL support	enabled	

Működési elv

Működés

Három fő szereplőre oszthatjuk az adatok feldolgozását:

1. drtheiss.hu weboldal (Rackhost Zrt.)
2. PROGEN Mérnöki Fejlesztő és Szolgáltató Kft. (sERPa vállalatirányítási rendszer)
3. AZILE Kft. (sipo.hu web áruház, products.csv)

A rendszer alapvető adathalmazát a sERPa szolgáltatja, mely az elsődleges terméklistát szállítja. Ezt a kényes adatállományt a rendszer minden nap négyszer, az előre meghatározott időpontokban (reggel 8-kor, délben 2-kor, délután 6-kor és este 10-kor) frissíti. E rendszeres frissítés folyamán a sERPa központi rendszere naprakész információkat juttat a rendszerünkbe, amelyek alapján a terméklista mindig aktuális és megbízható.

Azonban, ha a sERPa által továbbított adatok között olyan azonosító található, amely eddig még nem létezett a rendszerünkben, az alkalmazás figyelmesen és gondosan eljár. Ilyen esetekben a rendszer az adatbázisba az INSERT művelettel rögzíti az új adatokat, biztosítva ezzel az átlátható és teljes körű adatkezelést.

A sERPa által szállított adatok tehát az üzleti működés alapját képezik, és a rendszer szerves részét alkotják. A meghatározott frissítési időpontok szabályozott rendje gondoskodik a folyamatos adatáramlásról, és biztosítja, hogy minden üzleti nap kezdetén, délben, délután és este friss és pontos információk álljanak rendelkezésre. Ezen folyamatok összessége révén a rendszerünk az adatok legfrissebb és legmegbízhatóbb verzióját nyújtja, ami kritikus fontosságú a sikeres és zökkenőmentes működéshez, valamint a végfelhasználók elégedettségének megőrzéséhez.

A sERPa által küldött xml fájlok a következők:

Fájl: /expert/*.xml

- Categories.xml
- CustomerGroupData.xml
- OpenOrder.xml
- **Products.xml**
- ProductTree.xml
- SpecialPrice.xml
- StandardProduct.xml
- Stock.xml

Ebből a **Products.xml** fájlt használjuk az új rendszerben.
A képfájlokat az /expert/kepek/*.JPG lokalizációba tölti fel a sERPa.

Feldolgozó fájl: /cron/xml-handler.php

mezőnév	leírás	funkció
code	Termék egyedi kódja	Termékazonosítás
name	Termék sERPa ban szereplő neve	Másodlagos név
barcode	Vonalkód	Jelenleg nincs funkciója
net_weight	Nettó súly	Jelenleg nincs funkciója
gross_weight	Bruttó súly	Jelenleg nincs funkciója
main_image	Termékkép	Elsődlegesen megjelenő
short	Bevezető	Elsődlegesen megjelenő
content	Leírás	Elsődlegesen megjelenő

A Sipo web áruház által küldött adatok:
Fájl: /sipo/products.csv

mezőnév	leírás	funkció
vonalkod	Termék vonalkód	Jelenleg nincs funkciója
cikkszam	Termék egyedi kódja	Termékazonosítás
termeknev	Termék Sipo neve	Jelenleg nincs funkciója
sipo_ar	Kedvezményes ár	Kulcs egyezés alapján kiírt ár
sipo_30	Normál ár	Kulcs egyezés alapján kiírt ár
siplo_link	Sipo web áruház termék adatlap	Elsődlegesen megjelenő

Igen mélyen belelátva a rendszer működésébe, egyértelműen kirajzolódik az adatok beszerzésének két különböző forrása. Azonban a kettő közötti különbség olyan érzékeny jellegű, hogy e rendszer legfontosabb aspektusát, az árképzést érintően különösen érzékennyé válik. Ennek azért van kiemelkedő szerepe, mivel a fájlba történő bármilyen változtatás azonnal kihatással van az aktuális árak állapotára, és ez által a frissített adatok rögtön megtükröződnek.

Amennyiben bármilyen hiba, inkonzisztencia vagy hibás adat jelentkezik a fájlban, ez az árképzésben is váratlan problémákat és zavarokat idézhet elő. Ebből következően az adatintegritás és a fájlkezelés hibátlan működése a rendszer fenntarthatóságának és zavartalan üzemelésének sarokköve, melynek kritikus jelentőségű ellenőrzése és gondozása nélkülözhetetlen a színvonalas és problémamentes működés biztosításához.

Felépítés

SQL tábla

Adatbázis: c43432theiss_db

- brands (Márkáink)
- faq_category (Gyakori kérdések főcímei)
- faq_items (Gyakori kérdések elemei)
- feed_categories (Blog kategóriák)
- feed_items (Blog bejegyzés)
- feed_item_labels (Blog bejegyzések-hez kapcsolt címkék)
- feed_labels (Blog címkék)
- high_products (Kiemelt termékek)
- info (Főoldali folyamatok visszajelzés adattábla)
- lexicon (Lexikon bejegyzések)
- log (Weboldalon monitorozott tevékenység visszajelzése)
- magazines (Magazinok)
- markers (Patika lista, jelenleg nem használt, előzőből áthozott)
- menus (Az oldal termék menüinek táblája)
- pages (Fix oldal lista, jelenleg nem használt, előzőből áthozott)
- products (Termékek)
- product_categories (Termék kategóriák)
- product_documents (Termék dokumentumai, jelenleg nem használt, előzőből áthozott)
- related_feed_products (Blog bejegyzésekhez csatolt termékek)
- related_lexicon_products (Lexikon bejegyzésekhez csatolt termékek)
- related_products (Kapcsolt termékek)
- subscribes (Feliratkozók táblája)
- users (Felhasználók, admin)
- video_category (Videó kategóriák)
- video_item (Videó elemei)

A legfőbb mezőnevek:

- id: elsődleges kulcs
- title: cím
- slug: linkelérés url je

Fájlszerkezet főbb elemei

- activation.php
- admin.php
- **assets**
    - **css**
    - **fontawesome**
    - **fonts**
    - **img**
    - **js**
    - **slim**
- **ckeditor**
    - **adapters**
    - **lang**
    - **plugins**
    - **samples**
    - **skins**
- **ckfinder**
    - **_samples**
    - **_thumbs**
    - ckfinder.php
    - config.php
    - **core**
    - **files**
    - **help**
    - **images**
        - **boldogsagprogram**
        - **cikkek**
        - **diy**
        - **fagyoskalandok**
        - **husvet**
        - **kornyezettudatossag**
        - **lassitas**
        - **megfazas-es-kohoges**
        - **oszi-kalandok**
        - **szepsegtrend**
        - **zoldebb-jovo**
    - **lang**
    - **plugins**
        - **dummy**
        - **fileeditor**
        - **flashupload**
        - **gallery**
        - **imageresize**
        - **watermark**
        - **zip**
    - **skins**
        - **kama**

        - **v1**
    - **userfiles**
        - **_thumbs**
- connect.php
- **cron**
    - xml_handler.php
- csv.php
- **data**
    - **contents**
    - products.php
    - recovery.php
    - representatives.php
- phpinfo.php
- **pic**
- search.php
- **select2**
    - **.github**
    - **dist**
    - **docs**
    - **src**
    - **tests**
- **sipo**
- **stats**
- **uploads**
    - **brands**
    - **documentitem**
    - **experience**
    - **feed_items**
    - **gallery**
    - **gamers**
    - **magazines**
    - **maps**
    - **medicines**
    - **menu**
    - **packages**
    - **popup**
    - **product_catalog**
    - **product_category**
    - **products**
    - **slider**
    - **video**
    - **widget**
    - **xml**

Az oldal tartalmát a kifinomult rétegekben szerveződő struktúra adja. Az "oldalak" mappában megtalálhatóak az alapvető weboldalak, míg az "admin" alkönyvtár a kiváltságos adminisztrációs funkciók otthona. Az "ajax" könyvtár a háttérben futó műveletek irányításáért felelős, míg a "cron" gyökérkönyvtár az időzített automatizált folyamatok otthona.

Kiemelkedő fontosságú a "phpmailer" főkönyvtár jelenléte az "oldalak" mappában, hiszen ez létfontosságú összeköttetést biztosít a rendszer működéséhez, amely kibővíti a kommunikációs lehetőségeket és megteremti a hatékony adatfolyamot.

Egyértelműen elkülönülnek az adminisztrációs funkciók, melyeket az "oldalak/admin" könyvtárban találunk, így lehetővé téve a különbségtételt és könnyítve a rendszerkezelést.

Ez a struktúra olyan szintű organizációt kínál, amely lehetővé teszi az oldal tartalmának és funkcionalitásának hatékony és rugalmas kezelését, valamint a zavartalan és zökkenőmentes működést a Rackhost Zrt. szervereinek szigorú házszabályainak megfelelően.

functions.php

Az alkalmazott függvények a PHP fájlban kifinomult módon szolgálják ki a céljaikat. Fontos megemlíteni, hogy a Rackhost Zrt. gondosan kialakított szerverkörnyezete szigorú házszabályokkal operál. Ennek következtében a fájlkezelési feladatokat kreatív megoldással oldottuk meg, melynek középpontjában a cURL áll. Ezen technológia segítségével teremtettünk hatékony kapcsolatot a különböző erőforrások között, lehetővé téve a szerves adatáramlást és az optimális szolgáltatásnyújtást.

Létezik- e a fájl:

```
function isFileExists($url) {
  $ch = curl_init($url);
  curl_setopt($ch, CURLOPT_NOBODY, true);
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
  curl_exec($ch);
  $responseCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
  curl_close($ch);

  return $responseCode == 200;
}
```

Kép méretének lekérdezése:

```php
function getImageSizeWithCurl($url) {
 $ch = curl_init($url);
 curl_setopt($ch, CURLOPT_NOBODY, true);
 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
 curl_exec($ch);
 $responseCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);

 // Ellenőrizzük, hogy a kép létezik-e (200-as válaszkód)
 if ($responseCode === 200) {
    // Kérjük le a fejlécet, ami tartalmazza a kép méretét
    $contentLength = curl_getinfo($ch, CURLINFO_CONTENT_LENGTH_DOWNLOAD);

    // Ellenőrizzük, hogy a tartalomhossz nem null
    if ($contentLength) {
       // A kép mérete érvényes
       return $contentLength;
    } else {
       // Érvénytelen kép méret
       return false;
    }
 } else {
    // A kép nem található (nem létezik vagy nem elérhető)
    return false;
 }

 curl_close($ch);
}
```

Base64 kép formátumának lekérdezése:

```php
function getImageFormatFromBase64($base64ImageData) {
 $imageData = base64_decode($base64ImageData);
 $finfo = new finfo(FILEINFO_MIME_TYPE);
 $mime = $finfo->buffer($imageData);

 switch ($mime) {
    case 'image/jpeg':
       return 'jpeg';
    case 'image/png':
       return 'png';
    default:
       return 'jpg';
 }
}
```

M a régebbi tartalmak is figyelembe vételre szorultak, kénytelenek voltunk a függvények között olyan megoldásokat beépíteni, amelyek a tartalom formázását hatékonyan valósítják meg. Ezen megoldások egyfajta formázási kényszert hoznak létre a folyamat során, amelynek célja az adatok összehangolt és egységes megjelenítése a felületen.

Az új funkciók integrálása során ugyanis előtérbe került a múltbeli adatok megtartása és megjelenítése, ami a felhasználók számára is fontos és értékes aspektus. Ennek eredményeképpen a függvények struktúrájába beillesztettünk olyan megoldásokat, amelyek gondoskodnak a régi tartalmak formai összhangjáról az új funkciókkal.

Ezen formázási kényszer megoldásokkal való felvértezés lehetővé teszi a platform számára, hogy rugalmasan és hatékonyan kezelje a különböző típusú tartalmakat, és

biztosítsa a konzisztens és vonzó felhasználói élményt, tovább erősítve a webalkalmazás hitelességét és funkcionalitását.

Frontend content alapvető formázásainak felülírása:

```
function modositBetegtajekoztato($szoveg) {
  $keresettSzoveg = '<p>Betegtájékoztató: Információk a felhasználó számára</p>';
  $ujSzoveg = '<p style="text-align: center;"><font class="pil">Betegtájékoztató</font><br>Információk a felhasználó számára</p>';

  // Megkeressük a keresett bekezdést
  $pozicio = strpos($szoveg, $keresettSzoveg);

  if ($pozicio !== false) {
    // A keresett bekezdés előtti részt töröljük
    $szoveg = substr($szoveg, $pozicio);

    // Módosítjuk a bekezdést az új formátumra
    $szoveg = str_replace($keresettSzoveg, $ujSzoveg, $szoveg);
  }

  return $szoveg;
}

function modositBetegtajekoztatoKIEMELES($szoveg) {
  // Ellenőrizzük, hogy van-e sorszám a bekezdésben
  if (preg_match('/<p>(\d+)\./', $szoveg, $talalat)) {
    $sorszam = $talalat[1];

    // Kicseréljük a sorszámot vastag formázásra
    $szoveg = preg_replace('/<p>\d+\./', '<p style="font-weight:bold;">' . $sorszam . '.', $szoveg);
  }

  // Átalakított szöveg visszaadása
  return $szoveg;
}
```

phpmailer

A kiküldött e-mailek rendszere a felhasználó kezeléséhez és a feliratkozásokhoz kapcsolódóan kerül felhasználásra. Jelenleg elsősorban a regisztrációhoz kapcsolódik, de a jövőben további alkalmazási területek is elképzelhetők.
Küldő email cím: noreply@drtheiss.hu

Menürendszer

A weboldalon elérhető főmenük állandóak és változatlanok. Az egyetlen dinamikusan módosítható menü a termékkategóriáké. Ezek a kategóriák generáltan jelennek meg az oldalon, és a tartalmukhoz való hozzáférés a $_GET metódus segítségével valósul meg. Ez a dinamika lehetővé teszi, hogy könnyen kezelhető és rugalmasan alakítható legyen a termékkategóriák megjelenítése és elérése a weboldalon.

Asztali menük és generálásuk:

```
  <ul id="desktop">
    <li class="dropdown">Termékcsoportok
      <div class="dropdown-content">

        <ul>
        <?php
        $zero = 0;
        $one = 1;
        $stmt = $con->prepare("SELECT * FROM menus WHERE parent=? AND active=? ORDER BY menu_order ASC");
```

```
        $stmt->bind_param("ii",$zero,$one);
        $stmt->execute();
        $result = $stmt->get_result();
        $totalCount = $result->num_rows; // Teljes darabszám
        $count = 0;
        while ($row = $result->fetch_assoc()) {
           echo '<li><a href="'.$main_url.'kategoria/'.$row['menu_short'].'">'.$row['menu_name'].'</a></li>';
           $sub_category = $con->prepare("SELECT * FROM menus WHERE parent=? AND active=? ORDER BY menu_order ASC");
           $sub_category->bind_param("ii",$row['id'],$one);
           $sub_category->execute();
           $get_sub = $sub_category->get_result();
           $num_sub = $get_sub->num_rows;
           if($num_sub>0) {
             echo '<div id="sub-menu">';
             while($sub_row = $get_sub->fetch_assoc()) {
              echo '<a href="'.$main_url.'kategoria/'.$sub_row['menu_short'].'">'.$sub_row['menu_name'].'</a><br>';
             }
             echo '</div>';
           }
           $count++;

           if ($count % 6 == 0 && $count != $totalCount) {
              echo '</ul><ul>';
           }
        }
        ?>
        </ul>
        <ul>
        <li>
        <img src="<?=$main_url?>assets/img/menupic.jpg">
        </li>
        </ul>
```

Mobil menük és generálásuk:

```
<div id="myNav" class="overlay">
 <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>
 <div class="overlay-content">
  <ul>
  <?php
        $zero = 0;
        $stmt = $con->prepare("SELECT * FROM menus WHERE parent=? ORDER BY menu_order ASC");
        $stmt->bind_param("i",$zero);
        $stmt->execute();
        $result = $stmt->get_result();

        $totalCount = $result->num_rows; // Teljes darabszám
        $count = 0;

        while ($row = $result->fetch_assoc()) {
           echo '<li><a href="'.$main_url.'kategoria/'.$row['menu_short'].'">'.$row['menu_name'].'</a>';

           echo '</li>';
           $count++;
        }
        ?>
        <li><a href="<?=$main_url?>rolunk">Rólunk</a></li>
        <li><a href="<?=$main_url?>markaink">Márkáink</a></li>
        <li><a href="<?=$main_url?>magazin">Magazin</a></li>
        <li><a href="<?=$main_url?>media">Média</a></li>
        <li><a href="<?=$main_url?>ugyfelszolgalat">Ügyfélszolgálat</a></li>
  </ul>
 </div>
</div>
```

Termék menük $_GET metódus generálása:

```
    $kategoria = $_GET['kategoria'];
    $links = array();
    $links[] = '<a href="'.$main_url.'"><img src="'.$main_url.'assets/img/house.png"></a>';
    $links[] = '<a href="'.$main_url.'kategoria/'.$kategoria.'">Termékcsoportok</a>';
    if(isset($_GET['termek'])) {
      $slug = trim($_GET['termek']);
      $query_data = $con->prepare("SELECT *  FROM products WHERE slug=?");
      $query_data->bind_param("s",$slug);
      $query_data->execute();
      $get_data = $query_data->get_result();
      $num = $get_data->num_rows;
      if($num>0) {
        $fetch = $get_data->fetch_assoc();
        $links[] = '<a href="'.$main_url.'kategoria/'.$kategoria.'/'.$slug.'">'.$fetch['title'].'</a>';
      }
    }
      echo '
      <div id="head-content">
      <div id="back">';
      echo implode("<font class='line'0>|</font>",$links);
      echo '</div>
      <div id="headpic-about-us">
      <a target="_blank" href="https://sipo.hu/aktualis/dr-theiss-naturprodukt-termekek"><div id="webshop-without">
      <div id="webshop-inside">
        <span>webshop</span>
        <img src="'.$main_url.'assets/img/sipo-logo.png">
      </div>
      </div></a>
      <div id="head-logos">
      <div><a target="_blank" href="https://www.facebook.com/dr.theissmagyarorszag"><img src="'.$main_url.'assets/img/head-logos-
blue.png"><br><font>Dr. Theiss</font></a></div>
      <div><a target="_blank" href="https://www.facebook.com/lacalut.hu/"><img src="'.$main_url.'assets/img/head-logos-
red.png"><br><font>Lacalut</font></a></div>
      <div><a target="_blank" href="https://www.facebook.com/Dr.TheissCosmetics"><img src="'.$main_url.'assets/img/head-logos-
orange.png"><br><font>Cosmetics</font></a></div>
      </div>
      </div>
      </div>
      ';
      echo '<div id="content">
      <div id="inside-block" class="products-main-title">
        <h1 style="margin-top: 70px;">Termékeink</h1>
        <p style="margin-bottom: 30px;font-size:18px;">Válasszon termékeink közül vagy szükítse azokat.</p>
      </div>';
      include('pages/products.php');
      echo '</div>';
```

Képek létrehozása és szerkesztése

A képek feldolgozásához kapcsolódóan fontos megjegyezni, hogy a képek base64 formátumú kódját az $_POST['base64ImageData']; mezőben továbbítjuk. Ugyanakkor érdemes kiemelni, hogy ezen a ponton szükségessé vált a base64 kód tömörítése, és ebben rejlik a fő ok. Alapvetően ugyanis a base64 kód használata 33%-kal nagyobb fájlméretet eredményezhet, amely különösen érzékeny terület a webalkalmazások és a szerver-üzenetváltások során.

Ez a fajta adattömörítés lényegesnek bizonyult, hiszen a kódolt string mérete olyan méreteket öltött egyes esetekben, hogy elérte a PHP konfigurációban meghatározott maximális POST méretet, vagyis a max_input_vars paraméter értékét. Ennek eredményeként a korábban problémamentesen működő adatküldési folyamatok váltak instabillá és megszakadtak, ami kihatással lehetett a felhasználói élményre és a webalkalmazás teljesítőképességére.

Ezen adattömörítési mechanizmus mellett az adatokat a frontend-en található űrlapokhoz is hozzárendeljük, amelyek a HTML <form></form> tag-ek között találhatók. Ez a megközelítés lehetővé teszi a könnyű és hatékony adatkezelést, valamint a felhasználói interakció simaságának fenntartását. Ahhoz, hogy az adatok helyesen és problémamentesen kerüljenek feldolgozásra, kulcsfontosságú, hogy a base64 kódolás és annak megfelelő tömörítése megfelelően működjön, és a PHP konfiguráció paraméterei is megfelelően legyenek beállítva.

Képfeltöltés, cropper és tömörítés meghívása:

```
<script src="<?=$main_url?>assets/js/resizer.js"></script>
<script>

function imageData() {
  return {
    showCroppie: false,
    hasImage: <?php echo empty($fetch['image']) ? 'false' : 'true'; ?>,
    originalSrc: "",
    croppie: null, // change from empty object to null

    updatePreview() {
      var reader,
        files = this.$refs.input.files;

      reader = new FileReader();

      reader.onload = async (e) => {
        this.showCroppie = true;
        this.originalSrc = e.target.result;

        // Process the image and check if resizing is needed
        const processedImage = await process_image(files[0], 100); // 300KB as the minimum size
        if (processedImage) {
          this.originalSrc = processedImage;
        }

        this.bindCroppie(this.originalSrc);
      };

      reader.readAsDataURL(files[0]);
    },
    initCroppie() {
      this.croppie = new Croppie(this.$refs.croppie, {
```

```
        viewport: { width: 300, height: 243, type: "square" },
        boundary: { width: 720, height: 583 },
        showZoomer: true,
        enableResize: true
      });
      $('.cr-slider').attr({ 'min': 0.5000, 'max': 1.5000 });
    },
    clearPreview() {
      this.$refs.input.value = null;
      this.showCroppie = false;
    },
    swap() {
      this.$refs.input.value = null;
      this.showCroppie = false;
      this.hasImage = false;
      this.$refs.result.src = "";
      $('#showinput').append('<input type="hidden" name="deleteFile" value="true">');
    },
    edit() {
      this.showCroppie = true;
      this.hasImage = true;
      this.bindCroppie(this.originalSrc);
    },
    saveCroppie() {
      this.croppie
        .result({
          type: "base64",
          size: "original"
        })
        .then((croppedImage) => {
          this.$refs.result.src = croppedImage;
          this.showCroppie = false;
          this.hasImage = true;
          var hiddenInput = document.getElementById("base64ImageData");
          hiddenInput.value = croppedImage;
        });
    },
    bindCroppie(src) {
      setTimeout(() => {
        if (!this.croppie) {
          this.initCroppie();
        }
        this.croppie.bind({
          url: src
        });
      }, 200);
    }
  };
}

</script>
```

resizer.js

```
/**
 * Resize a base 64 Image
 * @param {String} base64Str - The base64 string (must include MIME type)
 * @param {Number} MAX_WIDTH - The width of the image in pixels
 * @param {Number} MAX_HEIGHT - The height of the image in pixels
 */
async function reduce_image_file_size(base64Str, MAX_WIDTH = 450, MAX_HEIGHT = 450) {
  let resized_base64 = await new Promise((resolve) => {
    let img = new Image()
    img.src = base64Str
    img.onload = () => {
      let canvas = document.createElement('canvas')
      let width = img.width
      let height = img.height

      if (width > height) {
```

```
            if (width > MAX_WIDTH) {
                height *= MAX_WIDTH / width
                width = MAX_WIDTH
            }
        } else {
            if (height > MAX_HEIGHT) {
                width *= MAX_HEIGHT / height
                height = MAX_HEIGHT
            }
        }
        canvas.width = width
        canvas.height = height
        let ctx = canvas.getContext('2d')
        ctx.drawImage(img, 0, 0, width, height)
        resolve(canvas.toDataURL()) // this will return base64 image results after resize
    }
  });
  return resized_base64;
}


async function image_to_base64(file) {
  let result_base64 = await new Promise((resolve) => {
    let fileReader = new FileReader();
    fileReader.onload = (e) => resolve(fileReader.result);
    fileReader.onerror = (error) => {
      console.log(error)
      alert('An Error occurred please try again, File might be corrupt');
    };
    fileReader.readAsDataURL(file);
  });
  return result_base64;
}

async function process_image(file, min_image_size = 100) {
  const res = await image_to_base64(file);
  if (res) {
    const old_size = calc_image_size(res);
    if (old_size > min_image_size) {
      const resized = await reduce_image_file_size(res);
      const new_size = calc_image_size(resized)
      console.log('new_size=> ', new_size, 'KB');
      console.log('old_size=> ', old_size, 'KB');
      return resized;
    } else {
      console.log('image already small enough')
      return res;
    }

  } else {
    console.log('return err')
    return null;
  }
}

/*- NOTE: USE THIS JUST TO GET PROCESSED RESULTS -*/
async function preview_image() {
  const file = document.getElementById('file');
  const image = await process_image(file.files[0]);
  // console.log(image)
}

/*- NOTE: USE THIS TO PREVIEW IMAGE IN HTML -*/
// async function preview_image() {
//   const file = document.getElementById('file');
//   const res = await image_to_base64(file.files[0])
//   if (res) {
//     document.getElementById("old").src = res;

//     const olds = calc_image_size(res)
//     console.log('Old ize => ', olds, 'KB')
```

```
//        const resized = await reduce_image_file_size(res);
//        const news = calc_image_size(resized)
//        console.log('new size => ', news, 'KB')
//        document.getElementById("new").src = resized;
//    } else {
//        console.log('return err')
//    }
// }


function calc_image_size(image) {
   let y = 1;
   if (image.endsWith('==')) {
      y = 2
   }
   const x_size = (image.length * (3 / 4)) - y
   return Math.round(x_size / 1024)
}
```

Minden egyes kép küldésével kapcsolatban, amikor szerkesztésről van szó, gondoskodunk arról, hogy ellenőrizzük az eredeti kép jelenlétét. Amennyiben szükséges, a régi képet eltávolítjuk, majd végrehajtjuk az új kép feltöltését és az adatbázis frissítését.

```
        if (!empty($base64ImageData)) {

            $base64Data = substr($base64ImageData, strpos($base64ImageData, ',') + 1);
            $imageData = base64_decode($base64Data);
            $imageSize = strlen($imageData);

            $imageFormat = getImageFormatFromBase64($base64ImageData);
            $validImageFormats = ['jpg', 'jpeg', 'png'];

            if ($imageSize > 6000000) {
                $tablename = 'brands';
                $type = 2;
                $text = 'A kép mérete nem lehet nagyobb, mint 6 MB.';
                $user_id = $user['user_id'];
                $time = time();
                $insert = $con->prepare("INSERT INTO `log` (`text`,`type`,`user_id`,`m_table`,`timestamp`) VALUES (?,?,?,?,?)");
                $insert->bind_param("siisi",$text,$type,$user_id,$tablename,$time);
                $insert->execute();
            } elseif (!in_array($imageFormat, $validImageFormats)) {
                $tablename = 'brands';
                $type = 2;
                $text = 'A kép formátuma csak jpg, jpeg vagy png lehet.';
                $user_id = $user['user_id'];
                $time = time();
                $insert = $con->prepare("INSERT INTO `log` (`text`,`type`,`user_id`,`m_table`,`timestamp`) VALUES (?,?,?,?,?)");
                $insert->bind_param("siisi",$text,$type,$user_id,$tablename,$time);
                $insert->execute();
            } else {

                $filename=WEB_ROOT.'uploads/brands/'.$fetch['image'];
                echo $filename;
                if(file_exists($filename))
                {
                  $errors[] = 'Régi fájl törölve';
                $img=unlink(WEB_ROOT.'uploads/brands/'.$fetch['image']);
                } else {
                  $errors[] = 'Régi fájl nem lett törölve';
                }

                // Folytasd a képfájl feldolgozásával és az adatbázis frissítésével
                $timestamp = time();
```

```
            $newImageName = $slug . '-' . $timestamp . '.' . $imageFormat;
            $newImageUrl = WEB_ROOT.'uploads/brands/' . $newImageName;



            if (file_put_contents($newImageUrl, $imageData)) {
                // 'A fájl sikeresen létrejött.';
            } else {
              $errors[] = 'Hiba történt a fájl létrehozásakor.';
            }

            $active = 0;
            $insert = $con->prepare("INSERT INTO brands (title,slug,short,image,content,seo_title,seo_keywords,seo_desc,created_at,active)
VALUES (?,?,?,?,?,?,?,?,?,?)");
            $insert-
>bind_param("sssssssssi",$title,$slug,$short,$newImageName,$content,$seo_title,$seo_keywords,$seo_desc,$created_at,$active);
            $insert->execute();
            $tablename = 'brands';
            $type = 1;
            $text = $title.' márka feltöltve.';
            $user_id = $user['user_id'];
            $time = time();
            $insert = $con->prepare("INSERT INTO `log` (`text`,`type`,`user_id`,`m_table`,`timestamp`) VALUES (?,?,?,?,?)");
            $insert->bind_param("siisi",$text,$type,$user_id,$tablename,$time);
            $insert->execute();
            header("Location: ".$main_url."admin/brands");
          }
        }
```

Ajax és fájlművelet

Általánosságban elmondható, hogy az adott adminisztrációs fájl tartalmazza az oda vonatkozó AJAX szkriptet. Az egyéb segédszkriptek külső forrásból kerülnek betöltésre. Ezt a megoldást akkor alkalmazzuk, amikor nagyobb méretű fájlokat szeretnénk feltölteni a szerverre, és időt kell hagyni a folyamatnak.

Ez a mechanizmus jelenik meg például a Magazin és a Média menüpontoknál, ahol az egyik esetben PDF fájlok, a másik esetben videó állományok feltöltése történik. Itt mind a felhasználói felületen, mind az adminisztrációs felületen elágazások találhatóak. Ez lehetővé teszi, hogy amennyiben csak a címeket kívánjuk módosítani, a rendszer ne próbálja meg újra megtalálni és feltölteni az akkor éppen "nem létező", még nem kiválasztott fájlt.

Ehhez szükségesek a rejtett inputok:

```
  <form name="edit" id="edit" action="<?=$main_url?>admin/media/new" method="post" enctype="multipart/form-data">
  <input type="hidden" id="id" value="<?=$id?>">
  <input type="hidden" id="modvideo" value="0">
  <input type="hidden" id="modimage" value="0">
  <input type="hidden" id="base64ImageData" value="data:<?php echo $imageFormat; ?>;base64,<?php echo $base64Encoded; ?>"
name="base64ImageData">
  <label>Név*</label>
  <input type="text" name="title" value="<?=$fetch['title']?>" id="title" required>
  <label>Keresőbarát cím*</label>
  <input type="text" name="slug" id="slug" value="<?=$fetch['slug']?>" required readonly>
  <label for="file"><span>MP4, MOV, (max 128MB)</span>   <?php
  echo ' - <font style="color:green;font-weight: bold;">Videófájl észlelve: '.$fetch['mp4'].'. Fájl cseréléséhez tallózzon újat.</font>';
  ?></label>
  <input type="file" name="video" class="video" id="file" required/>
  <script>
    $('.video').on("change", function(){
     $('#modvideo').val(1);
    });
  </script>
```

Elágazás a képfeltöltésben:

```
<script src="<?=$main_url?>assets/js/resizer.js"></script>
<script>

function imageData() {
 return {
   showCroppie: false,
   hasImage: <?php echo empty($fetch['image']) ? 'false' : 'true'; ?>,
   originalSrc: "",
   croppie: null, // change from empty object to null

   updatePreview() {
     var reader,
       files = this.$refs.input.files;

     reader = new FileReader();

     reader.onload = async (e) => {
       this.showCroppie = true;
       this.originalSrc = e.target.result;

       // Process the image and check if resizing is needed
       const processedImage = await process_image(files[0], 100); // 300KB as the minimum size
       if (processedImage) {
         this.originalSrc = processedImage;
       }

       this.bindCroppie(this.originalSrc);
     };

     reader.readAsDataURL(files[0]);
   },
   initCroppie() {
    this.croppie = new Croppie(this.$refs.croppie, {
     viewport: { width: 300, height: 243, type: "square" },
     boundary: { width: 720, height: 583 },
     showZoomer: true,
     enableResize: true
    });
    $('.cr-slider').attr({ 'min': 0.5000, 'max': 1.5000 });
   },
   clearPreview() {
    this.$refs.input.value = null;
    this.showCroppie = false;
   },
   swap() {
    this.$refs.input.value = null;
    this.showCroppie = false;
    this.hasImage = false;
    this.$refs.result.src = "";
    var hiddenInput = document.getElementById("base64ImageData");
       hiddenInput.value = '';
       var modimage = document.getElementById("modimage");
       modimage.value = 1;
   },
   edit() {
    this.showCroppie = true;
    this.hasImage = true;
    this.bindCroppie(this.originalSrc);
   },
   saveCroppie() {
    this.croppie
     .result({
       type: "base64",
       size: "original"
     })
     .then((croppedImage) => {
       this.$refs.result.src = croppedImage;
```

```
        this.showCroppie = false;
        this.hasImage = true;
        var modimage = document.getElementById("modimage");
        modimage.value = 1;
        var hiddenInput = document.getElementById("base64ImageData");
        hiddenInput.value = croppedImage;
      });
    },
    bindCroppie(src) {
      setTimeout(() => {
        if (!this.croppie) {
          this.initCroppie();
        }
        this.croppie.bind({
          url: src
        });
      }, 200);
    }
  };
}

</script>
```

Ajax php példa a videófájl feltöltsére:

```
<?php
session_start();
ob_start();
include('../../connect.php');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    function getImageFormatFromBase64($base64ImageData) {
        $imageData = base64_decode($base64ImageData);
        $finfo = new finfo(FILEINFO_MIME_TYPE);
        $mime = $finfo->buffer($imageData);

        switch ($mime) {
            case 'image/jpeg':
                return 'jpeg';
            case 'image/png':
                return 'png';
            default:
                return 'jpg';
        }
    }
    $response = array();
    $errors = array();
    define("WEB_ROOT", substr(dirname(__FILE__), 0, strlen(dirname(__FILE__)) - 11));
    $title = trim($_POST['title']);
    $slug = trim($_POST['slug']);
    $modvideo = trim($_POST['modvideo']);
    $modimage = trim($_POST['modimage']);
    $id = trim($_POST['id']);
    $base64ImageData = $_POST['base64ImageData'];
    $deleteFile = trim($_POST['deleteFile']);
    $category = 4;
    $error = 0;
    if (empty($title) || empty($slug)) {
        $errors['fields'] = 'A *-al jelzett mezők kitöltése kötelező!';
        $error = 1;
    } else {
    if($modvideo == 1 && $modimage == 1) {
        $allowedExts = array("mp4", "mov", "webm");
        $extension = pathinfo($_FILES['video']['name'], PATHINFO_EXTENSION);
        if ((($_FILES["video"]["type"] == "video/mp4")
            || ($_FILES["video"]["type"] == "video/mov")
            || ($_FILES["video"]["type"] == "video/webm"))
            && ($_FILES["video"]["size"] < 128000000)
            && in_array($extension, $allowedExts))
        {
```

```
        if ($_FILES["video"]["error"] > 0) {
            $errors['video'] = "Return Code: " . $_FILES["video"]["error"];
            $error = 1;
        } else {
            $time = time();
            $targetFileName = $slug ."-". $time . "." . $extension;
            $targetPath = WEB_ROOT . "/uploads/video/4/" . $targetFileName;

            if (file_exists($targetPath)) {
                $errors['video'] = $targetFileName . " létezik.";
                $error = 1;
            }
        }
    } else {
        $errors['video'] = "Érvénytelen fájl. Ellenőrizze a fájl formátumát és méretét.";
        $error = 1;
    }
    if (!empty($base64ImageData)) {
        $base64Data = substr($base64ImageData, strpos($base64ImageData, ',') + 1);
        $imageData = base64_decode($base64Data);
        $imageSize = strlen($imageData);

        $imageFormat = getImageFormatFromBase64($base64ImageData);
        $validImageFormats = ['jpg', 'jpeg', 'png'];

        if ($imageSize > 6000000) {
            $errors['image'] = 'A kép mérete nem lehet nagyobb, mint 6 MB.';
            $error = 1;
        } elseif (!in_array($imageFormat, $validImageFormats)) {
            $errors['image'] = 'A kép formátuma csak jpg, jpeg vagy png lehet.';
            $error = 1;
        }
    } else {
        $errors['image'] = 'Kép megadása kötelező!';
        $error = 1;
    }
}
if($modvideo == 1 && $modimage == 0) {
    $allowedExts = array("mp4", "mov", "webm");
    $extension = pathinfo($_FILES['video']['name'], PATHINFO_EXTENSION);
    if ((($_FILES["video"]["type"] == "video/mp4")
        || ($_FILES["video"]["type"] == "video/mov")
        || ($_FILES["video"]["type"] == "video/webm"))
        && ($_FILES["video"]["size"] < 128000000)
        && in_array($extension, $allowedExts))
    {
        if ($_FILES["video"]["error"] > 0) {
            $errors['video'] = "Return Code: " . $_FILES["video"]["error"];
            $error = 1;
        } else {
            $time = time();
            $targetFileName = $slug ."-". $time . "." . $extension;
            $targetPath = WEB_ROOT . "/uploads/video/4/" . $targetFileName;

            if (file_exists($targetPath)) {
                $errors['video'] = $targetFileName . " létezik.";
                $error = 1;
            }
        }
    } else {
        $errors['video'] = "Érvénytelen fájl. Ellenőrizze a fájl formátumát és méretét.";
        $error = 1;
    }
}
    if($modimage == 1 && $modvideo == 0) {
    if (!empty($base64ImageData)) {
        $base64Data = substr($base64ImageData, strpos($base64ImageData, ',') + 1);
        $imageData = base64_decode($base64Data);
        $imageSize = strlen($imageData);

        $imageFormat = getImageFormatFromBase64($base64ImageData);
        $validImageFormats = ['jpg', 'jpeg', 'png'];
```

```
        if ($imageSize > 6000000) {
            $errors['image'] = 'A kép mérete nem lehet nagyobb, mint 6 MB.';
            $error = 1;
        } elseif (!in_array($imageFormat, $validImageFormats)) {
            $errors['image'] = 'A kép formátuma csak jpg, jpeg vagy png lehet.';
            $error = 1;
        }
    } else {
        $errors['image'] = 'Kép megadása kötelező!';
        $error = 1;
    }
  }
}
  $query = $con->prepare("SELECT * FROM video_item WHERE category=? AND id=?");
  $query->bind_param("ii",$category,$id);
  $query->execute();
  $get = $query->get_result();
  $num = $get->num_rows;
  if($num>0) {
    $fetch = $get->fetch_assoc();
  } else {
    $error = 1;
  }
  if ($error == 0) {
    if($modvideo == 1 && $modimage == 1) {
    $officialvideoPath = WEB_ROOT . "/uploads/video/4/" . $fetch['mp4'];
    if (file_exists($officialvideoPath)) {
        unlink($officialvideoPath);
    }
    if (move_uploaded_file($_FILES["video"]["tmp_name"], $targetPath)) {
        $response['message'] = "Stored in: " . $targetPath;
    } else {
        $errors['video'] = "A fájl nem lett feltöltve.";
        $error = 1;
    }
    $officialimagePath = WEB_ROOT . "/uploads/video/4/" . $fetch['image'];
    if (file_exists($officialimagePath)) {
        unlink($officialimagePath);
    }
    $timestamp = time();
    $newImageName = $slug . '-' . $timestamp . '.' . $imageFormat;
    $newImageUrl = WEB_ROOT.'/uploads/video/4/' . $newImageName;
    if (file_put_contents($newImageUrl, $imageData)) {
        $response['message'] = 'A fájl sikeresen létrejött.';
    } else {
        $errors['image'] = 'Hiba történt a képfájl létrehozásakor.';
        $error = 1;
    }
    $active = 0;
    $update = $con->prepare("UPDATE video_item SET title=?,slug=?,image=?,mp4=?,category=?,active=?,updated_at=? WHERE id=?");
    $update->bind_param("ssssiisi",$title, $slug, $newImageName, $targetFileName, $category, $active, $updated_at, $id);
    $update->execute();
    }
    if($modimage == 1 && $modvideo == 0) {
        $officialimagePath = WEB_ROOT . "/uploads/video/4/" . $fetch['image'];
        if (file_exists($officialimagePath)) {
            unlink($officialimagePath);
        }
    $timestamp = time();
    $newImageName = $slug . '-' . $timestamp . '.' . $imageFormat;
    $newImageUrl = WEB_ROOT.'/uploads/video/4/' . $newImageName;
    if (file_put_contents($newImageUrl, $imageData)) {
        $response['message'] = 'A fájl sikeresen létrejött.';
    } else {
        $errors['image'] = 'Hiba történt a képfájl létrehozásakor.';
        $error = 1;
    }
    $active = 0;
    $update = $con->prepare("UPDATE video_item SET title=?,slug=?,image=?,category=?,active=?,updated_at=? WHERE id=?");
    $update->bind_param("sssiisi",$title, $slug, $newImageName, $category, $active, $updated_at, $id);
    $update->execute();
```

```
    }
    if($modvideo == 1 && $modimage == 0) {
        $officialvideoPath = WEB_ROOT . "/uploads/video/4/" . $fetch['mp4'];
        if (file_exists($officialvideoPath)) {
            unlink($officialvideoPath);
        }
    if (move_uploaded_file($_FILES["video"]["tmp_name"], $targetPath)) {
        $response['message'] = "Stored in: " . $targetPath;
    } else {
        $errors['video'] = "A fájl nem lett feltöltve.";
        $error = 1;
    }
    $active = 0;
    $update = $con->prepare("UPDATE video_item SET title=?,slug=?,mp4=?,category=?,active=?,updated_at=? WHERE id=?");
    $update->bind_param("sssiisi",$title, $slug, $targetFileName, $category, $active, $updated_at, $id);
    $update->execute();
    }
    if($modvideo == 0 && $modimage == 0) {
        $active = 0;
        $update = $con->prepare("UPDATE video_item SET title=?,slug=?,category=?,active=?,updated_at=? WHERE id=?");
        $update->bind_param("ssiisi",$title, $slug, $category, $active, $updated_at, $id);
        $update->execute();
    }
    $tablename = 'video_item';
    $type = 1;
    $text = $title.' média frissítve.';
    $user_id = $user['user_id'];
    $time = time();
    $insert = $con->prepare("INSERT INTO `log` (`text`,`type`,`user_id`,`m_table`,`timestamp`) VALUES (?,?,?,?,?)");
    $insert->bind_param("siisi",$text,$type,$user_id,$tablename,$time);
    $insert->execute();

    $response['success'] = true;
    $response['message'] = 'Sikeres feltöltés!';
  } else {
    $response['success'] = false;
    $response['errors'] = $errors;
  }
  // Kiküldjük a választ JSON formátumban
  header('Content-Type: application/json');
  echo json_encode($response);
}
?>
```

Mint látható az adattömbök json formátumban kerülnek elküldésre, a válaszhoz pedig szükséges ennek visszaalakítása. A fájl útvonalak pontos meghatározása érdekében pedig, egyetlen egy konstans változó került deklarálásra:

```
define("WEB_ROOT", substr(dirname(__FILE__), 0, strlen(dirname(__FILE__)) - 11));
```

Hírlevél

Feliratkozók táblája

Ennek a táblának a feltöltéséről, az index.php fájl footerjében elhelyezett „Hírlevél” form gondoskodik, ajax kéréssel.

Feliratkozók küldése levelező részére

Ezen téma kifejtésre kerül a kódsor integrációja után.

Kiegészítő domain

naturprodukt.hu

Fontos, hogy a "naturprodukt.hu" domain ne kerüljön átmenetileg sem alias státuszba, amíg olyan hirdetések futnak, amelyek a fenti domainra mutatnak. Jelenleg ez a domain önálló oldalként működik, és egy PHP fordító scriptet tartalmaz. Ennek eredményeként, ha a hirdetett termékkategóriák "slug" nevei változnak, azok nem fogják a megfelelő célpontra irányítani. Ha ilyen változásra sor kerül, szükség lehet a PHP scriptben található tömb (array) utólagos konfigurálására.

```php
<?php
if($_GET['kategoria']) {
    $actual_link = (isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] === 'on' ? "https" : "http") .
"://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
    $eredetiLink = $actual_link;
    $ujDomain = "drtheiss.hu";
    // Kategória szűrési tömb
    $kategoriaSzures = array(
        "vas" => "vaspotlas",
        "szemcsepp" => "szem"
        // Itt további egyedi szűrőket adhatsz meg a kívánt link nevekhez
    );
    // Kategória nevének megszerzése az eredeti linkből
    preg_match("/termekek\/([^\/]+)/", $eredetiLink, $talalat);
    if (count($talalat) > 1 && isset($kategoriaSzures[$talalat[1]])) {
        $ujKategoria = 'kategoria/'.$kategoriaSzures[$talalat[1]];
        $modositottLink = str_replace("termekek/$talalat[1]", "$ujKategoria", $eredetiLink);
        $modositottLink = str_replace("naturprodukt.hu", $ujDomain, $modositottLink);
        header('Location:'.$modositottLink.'');
        echo $modositottLink;
    } else {
        // Alapértelmezett működés, ha nincs egyezés a szűrésben
        $ujKategoria = "kategoria";
        $modositottLink = str_replace("naturprodukt.hu", $ujDomain, $eredetiLink);
        $modositottLink = str_replace("termekek", $ujKategoria, $modositottLink);
        header('Location:'.$modositottLink.'');
        echo $modositottLink;
    }
} else {
    header('Location:https://drtheiss.hu');
}
?>
```

drtheisspharma.hu

Kiegészítő domain, fejlesztési státuszra alkalmazva, index.php átirányítás a fő domainre.