



2024/25/1

Notebook Weboldal

Dokumentáció

Készítették:

Bitó Tamás - I866YG

Mixtaj Flóra - EETLOL

GitHub projekt URL: <https://github.com/bitotamas/Java-alkalmazasok-Gyakorlat-Beadando>

Bevezető.....	2
A Felhasználói felület kialakítása	3
A feladatrészek megvalósítása.....	4
Adatbázis.....	4
Controller	5
HomeController	5
Home oldal: /	5
Admin felület: /admin	5
Bejelentkezés: /login	5
Regisztráció: /register és /register_process.....	5
Kapcsolatfelvételi űrlap: /contact és /contactResult	6
Üzenetek megtekintése: /messages	6
Termékek megjelenítése: /products	6
Szerkezeti felépítés	7
1. Fő projektstruktúra	7
2. Forrásmappa (src/main/java).....	7
A menü Lehetőségei.....	8
regisztráció almenü.....	8
Regisztráció - nézete	8
Regisztráció - Működése	9
Bejelentkezés almenü	9
Bejelentkezés - nézet	9
Bejelentkezés – működése.....	10
Kapcsolat almenü	11
Kapcsolat – nézet	11
Kapcsolat-működése	11
Üzenetek almenü	12
Üzenetek - nézet	12
Üzenetek - működése	13
Termékek almenü	13
Termékek - nézet.....	13
Termékek - működése.....	14
Admin Almenü.....	15
Admin nézet	15
RESTful API	16
cUrl.....	18
GET	18

POST	18
PUT	19
DELETE.....	19
Postman	20
GET	20
POST	21
PUT	21
DELETE.....	22
Feladatmegoldásban használt elemek.....	23
Megjelenítések.....	23
css mappa.....	23
images mappa	23
js mappa.....	23
Egyéb mappák	23
templates mappa	23
HTML fájlok	24

BEVEZETŐ

A program futtatásához létrehoztunk egy **weboldal_indito.bat** indító fájlt, ami elindítja a vele egy mappában lévő **notebook_web.jar** fájlt. Erre azért volt szükség, mivel amennyiben csak önmagában a jar kiterjesztésű fájlt indítjuk el, semmi visszajelzést nem kapunk arról, hogy elindult-e vagy sem. Csak abból tudjuk, ha a localhost:8080-as url nem tér vissza hibával, vagy a feladatkezelőben fut a Java(TM) Platform SE binary folyamat. Ezzel a megoldással nyomon tudjuk követni egy konzolablakban a futási eredményt, illetve azzal be is tudjuk zárni a weboldal szolgáltatását.

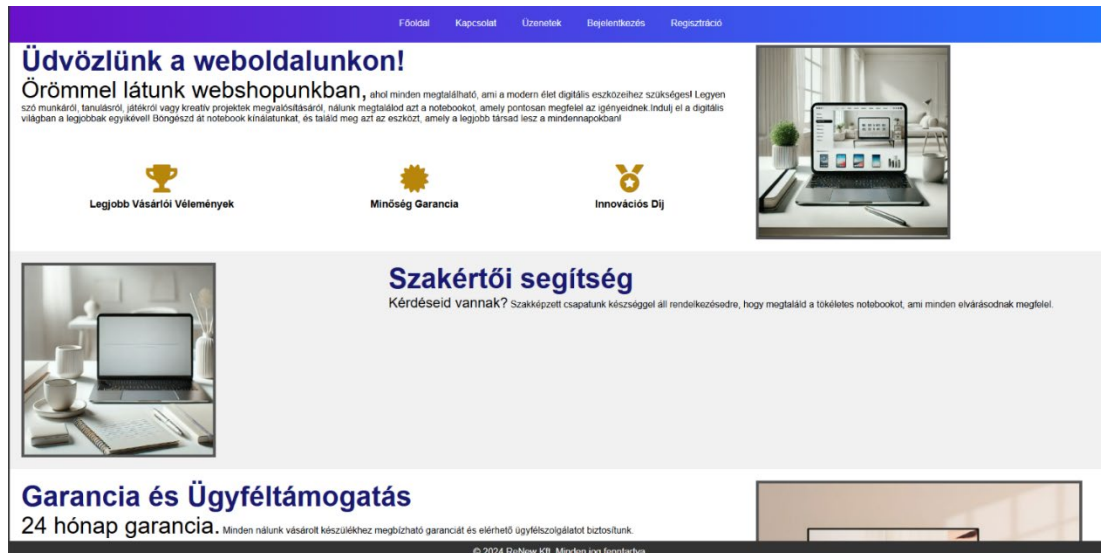
```
echo off
java -jar notebook_web.jar
pause
```

A visszajelzéshez pedig egy külön indító Main osztályt hoztunk létre ami elindítja a fő programunkat

```
public class Main {
    public static void main(String[] args) {
        boolean started=false;
        try{
            NotebookWebApplication.main(args);
            started=true;
        } catch (Exception e) {
            System.out.println("\nHiba! a Weboldal nem indult el! Kérjük indítsa újra a programot!");
        }
        if (started) {
            System.out.println("\nA weboldal sikeresen elindult! Kérjük írja a böngészőjébe a weboldal megnyitásához: localhost:8080");
        }
    }
}
```

A FELHASZNÁLÓI FELÜLET KIALAKÍTÁSA

A kezdőlapra egy üzenet fogadja a látogatókat, amely bemutatja az oldal funkcióit és kiemeli a kínálat fő előnyeit. Az oldal tetején elhelyezett menüsáv (Főoldal, Kapcsolat, Bejelentkezés, Regisztráció) segíti a látogatók gyors eligazodását a webhely különböző funkciói között.



```
<div xmlns:th="http://www.thymeleaf.org"
      xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">
  <div>
    <span sec:authorize="isAnonymous()">
      <a th:href="@{/}">Főoldal</a>
      <a th:href="@{/contact}">Kapcsolat</a>
      <a th:href="@{/messages}">Üzenetek</a>
      <a th:href="@{/login}">Bejelentkezés</a>
      <a th:href="@{/register}">Regisztráció</a>
    </span>

    <span sec:authorize="hasRole('ROLE_USER')">
      <a th:href="@{/}">Főoldal</a>
      <a th:href="@{/contact}">Kapcsolat</a>
      <a th:href="@{/messages}">Üzenetek</a>
      <a th:href="@{/products}">Termékek</a>
      <a sec:authentication="principal.username" class="inactiveLink"></a>
      <a th:href="@{/logout}">Kijelentkezés</a>
    </span>

    <span sec:authorize="hasRole('ROLE_ADMIN')">
      <a th:href="@{/}">Főoldal</a>
      <a th:href="@{/admin}">Admin</a>
      <a sec:authentication="principal.username" class="inactiveLink"></a>
      <a th:href="@{/logout}">Kijelentkezés</a>
    </span>
  </div>
</div>
```

```
<header>
  <div th:insert="~{menu}" class="topnav"></div>
</header>

<main>
```

A FELADATRÉSZEK MEGVALÓSÍTÁSA

ADATBÁZIS

Az adatbázis egy öt táblából áll. A táblák az alábbiak, ahogy a képen is láthatjuk:

contact, gep, oprendszer, processzor és users. Az adatbázist a DB Browser for SQLite segítségével hoztuk létre, feladat.db néven.

A contact tábla, ami eltárolja a beérkező üzeneteket, ezen belül a felhasználó azonosítóját, email-címét, az üzenet küldési idejét.

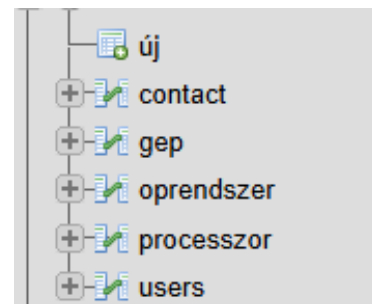
A gep tábla eltárolja a a geo azonosítóját, gyártóját, típusát, kijelző méretét, memóriáját, merevlemezét, videóvezérlőjét, árát, a processzor azonosítóját, az operációsrendszer azonosítóját és a gépek darabszámát.

Az oprendszer tábla eltárolja az operációs rendszer azonosítóját (ez alapján tudjuk a gep táblában melyik operációs rendszer tartozik hozzá), illetve az operációs rendszer nevét.

A processzor tábla pedig a processzor azonosítóját (hasonló eset az operációs rendszeréhez), gyártóját, illetve típusát tárolja el

A user tábla a felhasználók adatait tárolja el. Azonosítót, email-címet és a jelszavakat hashalt állapotban a biztonság miatt.

A processzor tábla pedig a processzor azonosítóját (hasonló eset az operációs rendszeréhez), gyártóját, illetve típusát tárolja el.



password

\$2a\$10\$QEaf3I.eLiZC4F4pDnqmC.sTysFIJ59wgROmw3ATxce...

\$2a\$10\$exVjZOnYQ3oFdNTP7qVHOoL8K2XhKpWXY3r8duw8v9...

\$2a\$10\$mIRM7vP9d7TWf7Cw6DqwA.irQjOkh0AVxiCm6bu7v3A...

\$2a\$10\$0DNmtKwQizGAeR91fleJ/eSCvza9cHqdj6P/LZdTikN...

\$2a\$10\$iQKqGjCRmUTbwIjXu.dtWuVHkAWC6DgSpdtdMqoewqp...

\$2a\$10\$w8zpNM1EvbkUwQWz7ZZUcOgNZ48x/En/2ACBdmANfz9...

\$2a\$10\$wplZ14rnjrAXzwk45gvvzec8bEp6N1INn3IZaa3ceP3...

CONTROLLER

HOMECONTROLLER

A HomeController egy Spring-alapú webalkalmazás vezérlője, amely felhasználók kezelésével, autentikációval, kapcsolatfelvétellel, üzenetek megjelenítésével, valamint termékek véletlenszerű kiválasztásával és megjelenítésével foglalkozik. Az osztály jól elkülöníti az egyes funkciókat, és Spring Boot erőforrások segítségével dolgozik az adatokkal és az autentikációval.

HOME OLDAL: /

- A home() metódus egy egyszerű GET végpont, amely az alkalmazás főoldalára irányítja a felhasználót az index.html nézet betöltésével.

ADMIN FELÜLET: /ADMIN

- **Cél:** Az összes felhasználó megjelenítése adminisztrációs célokra.
- **Működés:**
 - A userRepo.findAll() lekéri az adatbázisban tárolt összes felhasználót.
 - Ezeket a felhasználókat egy userList nevű attribútumként átadja az admin.html nézetnek.

BEJELENTKEZÉS: /LOGIN

- **Cél:** Bejelentkezési oldal megjelenítése és hibaüzenet biztosítása.
- **Működés:**
 - Ha a login végpontra érkezik egy "error" paraméter, akkor a modellhez egy hibaüzenetet (loginError) ad hozzá, amely hibás bejelentkezési adatokat jelöl.

REGISZTRÁCIÓ: /REGISTER ÉS /REGISTER_PROCESS

- **Regisztrációs oldal (GET):**
 - Az új User objektumot előkészíti és a register.html nézetnek átadja.
- **Regisztráció feldolgozása (POST):**
 - Ellenőrzi, hogy az e-mail vagy a név már foglalt-e az adatbázisban.
 - Ha nem foglalt:
 1. A jelszót BCrypt algoritmussal titkosítja (BCryptPasswordEncoder).
 2. A User objektum szerepkörét (ROLE_USER) beállítja.
 3. Mentésre kerül az adatbázisban.

4. A felhasználót a bejelentkezési oldalra irányítja sikerüzenettel.

- Ha az e-mail vagy a név már foglalt, hibaüzenettel visszairányítja a regisztrációs oldalra.

KAPCSOLATFELVÉTELI ŰRLAP: /CONTACT ÉS /CONTACTRESULT

- **Kapcsolatfelvételi oldal (GET):**

- Ha a felhasználó be van jelentkezve, a felhasználó e-mail címét betölti, különben üres e-mail címként jelenik meg.
- Egy új Contact objektumot inicializál.

- **Kapcsolatfelvétel beküldése (POST):**

- Az űrlap adatait a Contact entitásba tölti.
- Ha a felhasználó nem azonosított (pl. vendég), a nevet "Guest"-ként állítja be.
- Az adatokat menti az adatbázisba és visszairányítja a /contact oldalra sikerüzenettel.

ÜZENETEK MEGTEKINTÉSE: /MESSAGES

- **Cél:** Összes kapcsolatfelvételi üzenet megjelenítése az admin számára.

- **Működés:**

- Lekéri az adatbázisból az összes Contact objektumot.
- Ezeket időrendi sorrendben rendezi a created_at attribútum szerint.
- Ha nincs üzenet, egy üres lista kerül a modellhez.

TERMÉKEK MEGJELENÍTÉSE: /PRODUCTS

- **Cél:** Notebook termékek, processzorok és operációs rendszerek megjelenítése.

- **Működés:**

- Lekéri az összes Notebook, CPU és OS entitást az adatbázisból.
- Véletlenszerűen kiválaszt legfeljebb 16 notebookot, amelyek elérhetők (db > 0).
- Az egyes notebookokhoz hozzákapcsolja a hozzájuk tartozó processzort és operációs rendszert.
- A termékeket a products.html nézethez adja át.

SZERKEZETI FELÉPÍTÉS

1. FŐ PROJEKTSTRUKTÚRA

- **notebook_web**: Ez a gyökérmappa, amely tartalmazza a teljes projektet, és az alábbi mappákra van bontva:
 - **.idea, .mvn**: Ezek általában a projekt konfigurációs és Maven fájljait tartalmazzák.
 - **src**: A projekt fő kódmappája, ahol az alkalmazás forrásfájljai találhatók.

2. FORRÁSMAPPA (SRC/MAIN/JAVA)

Ez tartalmazza a Java forráskódot, struktúrált módon, különböző funkcionális egységekre bontva:

- **Controller**:
 - **HomeController.java**: Egy alapértelmezett vezérlő, amely a kezdőoldalhoz vagy az alapvető funkciókhoz kapcsolódik.
- **Model**:

Ez a csomag tartalmazza az alkalmazás modelljeit (az adatokat reprezentáló osztályokat):

 - **Contact.java**: Egy kapcsolat vagy személy adatait modellezi.
 - **CPU.java**: A notebook processzorának adatait reprezentálhatja.
 - **NewContact.java**: Új kapcsolatok adatainak modellezésére szolgálhat.
 - **Notebook.java**: A notebookok adatait tartalmazó fő modell lehet.
 - **OS.java**: Az operációs rendszerek adatait tartalmazza.
 - **User.java**: A felhasználói adatokhoz kapcsolódó modell.

notebook_web

Ebben a csomagban több funkcionális egység is jelen van:

- **Contact.java, User.java, HomeController.java**
- **Repository osztályok**: Adatbázis-interakciókért felelős interfészek, például:
 - **ContactRepository.java, UserRepository.java**: Kapcsolatok és felhasználók adataival dolgozó interfészek.
- **WebSecurityConfig.java**: A biztonsági beállításokat tartalmazó osztály.

Repository csomag

Ez a csomag az összes adattárolási és adatbázis-kezelési logikát tartalmazza:

- **ContactRepository.java, CPURepository.java, NewContactRepository.java, NotebookRepository.java, OSRepository.java, UserRepository.java**: Ezek az egyes modellekhez kapcsolódó adatbázis-lekérdezéseket kezelik.

RestAPI csomag

- **ContactApiController.java**: A kapcsolatokhoz kapcsolódó REST API végpontokat kezelő osztály.

Security csomag

- **WebSecurityConfig.java:** Az alkalmazás biztonsági konfigurációit kezeli (például hitelesítés, hozzáférési szabályok).

Service

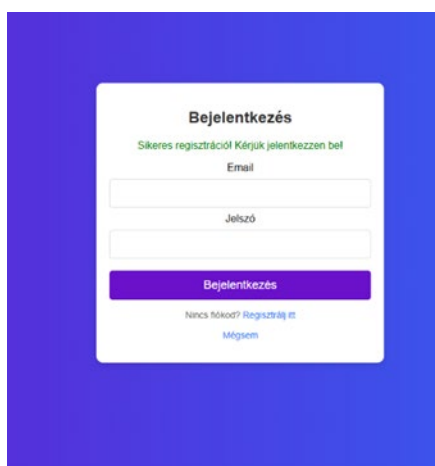
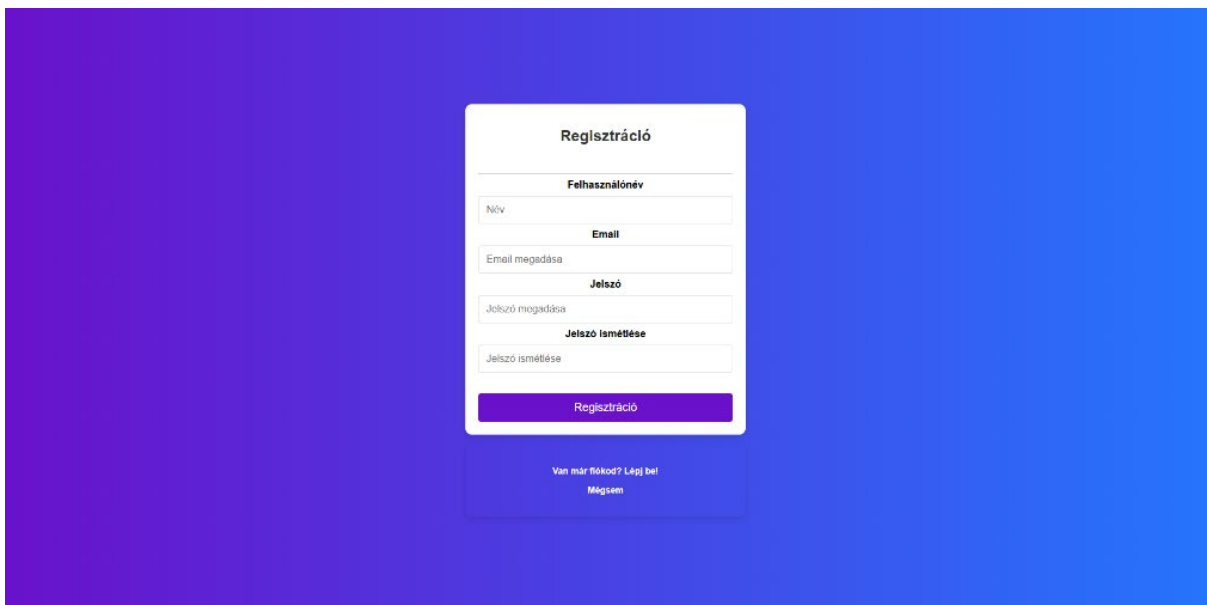
- **NotebookWebApplication.java:** Az alkalmazás indításáért és alapvető beállításaiért felelős osztály.

A MENÜ LEHETŐSÉGEI

REGISZTRÁCIÓ ALMENÜ

REGISZTRÁCIÓ - NÉZETE

A regisztráció menü lenyomásával, megjelenik a regisztrációs felület, ahol az új felhasználók tudnak regisztrálni. Az új adatok a regisztráció elküldése után hozzáadódnak a users táblához. Az alábbi műveletet egy javascript kód támogatja, amely segít végrehajtani a kéréseket.



Sikeres regisztráció esetén van lehetőségünk rögtön bejelentkezni. Erről zöld felirattal kapunk is megerősítést, hogy a regisztráció sikeres volt.

REGISZTRÁCIÓ - MŰKÖDÉSE

Ez a JavaScript kód a következő validálásokat végzi el a regisztrációs űrlap előtti hibák elkerülése érdekében:

1. A név legalább 4 karakter hosszú kell, hogy legyen.
2. Az email cím formátuma helyes kell, hogy legyen.
3. A két jelszó egyeznie kell.

Ha bármelyik feltétel nem teljesül, akkor a hibák megjelennek a felhasználó számára, és az űrlap nem kerül elküldésre, amíg minden helyes adatot nem adnak meg.

isValid: Ez az alapértelmezett változó, amely nyomon követi, hogy az űrlap érvényes-e (ha bármelyik ellenőrzés nem teljesül, a változó false-ra változik).

messages: A hibüzeneteket tárolja, ha valamelyik ellenőrzés nem teljesül

```

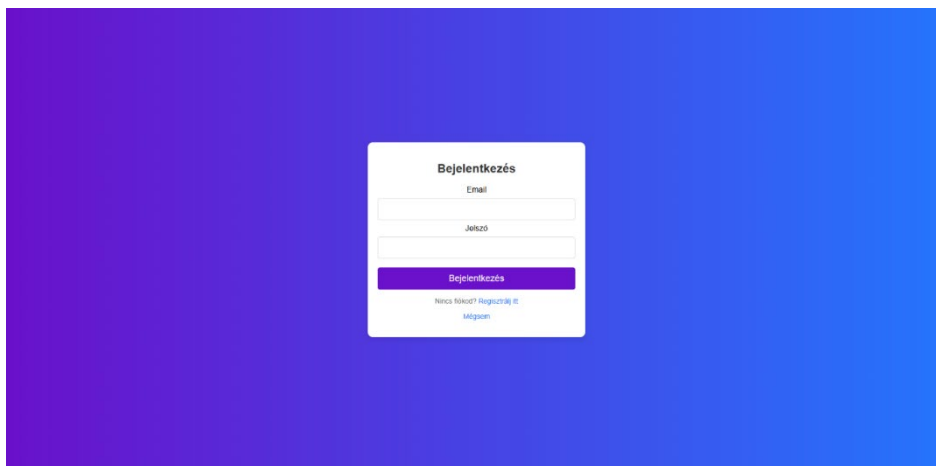
1 document.addEventListener('DOMContentLoaded', function () {
2     const passwordRepeatInput = document.getElementById('password-repeat');
3     const errorMessage = document.getElementById('error-message');
4
5     form.addEventListener('submit', function (event) {
6         let isValid = true;
7         let messages = [];
8
9         // Név ellenőrzése (min. 4 karakter)
10        if (nameInput.value.length < 4) {
11            //alert("A névnek minimum 4 karakterből kell állnia!");
12            messages.push("A névnek minimum 4 karakterből kell állnia!");
13            isValid = false;
14        }
15
16        // Email formátum ellenőrzés
17        const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
18        if (!emailPattern.test(emailInput.value)) {
19            //alert("Kérlek, adj meg egy érvényes email címet!");
20            messages.push("Kérlek, adj meg egy érvényes email címet!");
21            isValid = false;
22        }
23
24        // Jelszavak egyezésének ellenőrzése
25        if (passwordInput.value !== passwordRepeatInput.value) {
26            //alert("A két jelszónak egyeznie kell!");
27            messages.push("A két jelszónak egyeznie kell!");
28            isValid = false;
29        }
30
31        // Ha bármelyik feltétel nem teljesül, megakadályozza az űrlap elküldését
32        if (!isValid) {
33            event.preventDefault();
34            errorMessage.style.display = "block";
35            errorMessage.style.color = "red";
36            errorMessage.innerHTML = messages.join("<br>");
37        }
38    });
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

BEJELENTKEZÉS ALMENÜ

BEJELENTKEZÉS - NÉZET

A bejelentkezés menü a felhasználói hitelesítés folyamatát valósítja meg. A következő lépések biztosítják a biztonságos és hatékony működést. Regisztráció után az ügyfeleknek van lehetőségük bejelentkezni, hogy többműveletet is tudjanak használni.



Ha a successMessage változó értéke nem üres, akkor egy zöld színű üzenetet jelenít meg => Sikeres bejelentkezés. Ha a loginError változó nem megfelelő, akkor egy piros szöveges hibajelzés jelenik meg => Hibás email cím vagy jelszó

BEJELENTEKZÉS – MŰKÖDÉSE

```

<!DOCTYPE html>
<html lang="hu" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bejelentkezés</title>
  <link rel="icon" type="image/x-icon" href="/images/webicon.jpg">
  <link rel="stylesheet" href="/css/login.css">
</head>
<body>

<div class="login-container">
  <h2>Bejelentkezés</h2>
  <div th:if="${successMessage}">
    <p style="color: green;" th:text="${successMessage}"></p>
  </div>
  <div th:if="${loginError}">
    <p style="color: red;" th:text="${loginError}"></p>
  </div>
  <form method="post" action="/login">
    <label for="email">Email</label>
    <input type="email" id="email" name="username" required>

    <label for="password">Jelszó</label>
    <input type="password" id="password" name="password" required>

    <button type="submit">Bejelentkezés</button>
    <div class="container signin">
      <p class="register-link">Nincs fiókod? <a href="/register">Regisztrálj itt</a></p>
      <p class="register-link"><a href="/">Mégsem</a></p>
    </div>
  </form>
</div>

</body>
</html>

```

KAPCSOLAT ALMENÜ

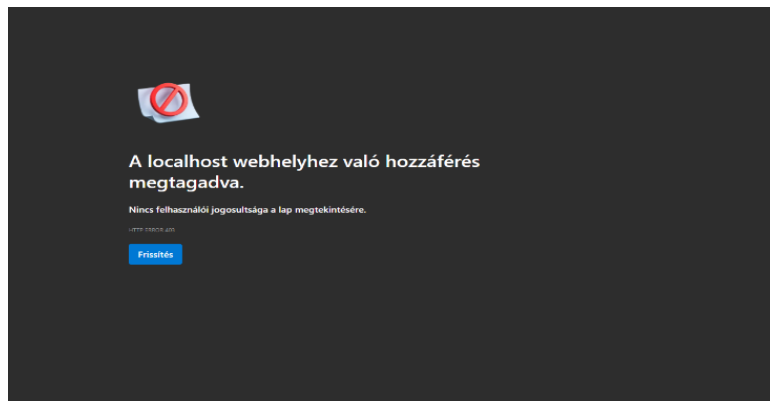
KAPCSOLAT – NÉZET

A kapcsolat menü lehetőséget ad a felhasználóknak arra, hogy kapcsolatba lépjenek a weboldal üzemeltetőjével, adatokat (email, üzenet) küldjenek el, és sikeres üzenetküldés esetén visszajelzést kapjanak. Az űrlap helyes kitöltése esetén a felhasználók egy sikerüzenetet, hibás kitöltés esetén pedig hibát kapnak. Az elküldés gomb után az üzenetek a contact táblában fognak megjelenni. Fontos megjegyezni, hogy üzenetet csak a bejelentkezett felhasználók tudnak küldeni.

The screenshot shows a web application interface with a blue header and a blue gradient background. The header contains navigation links: 'Főoldal', 'Kapcsolat', 'Üzenetek', 'Termékek', 'valaki12@gmail.com', and 'Kijelentkezés'. The main content area features a white box titled 'Kapcsolati űrlap'. Inside this box, there are two input fields. The first is labeled 'Email' and contains the text 'valaki12@gmail.com'. The second is labeled 'Subject' and contains the placeholder text 'Írj nekünk...'. Below these fields is a green button labeled 'Elküldés'. At the bottom left of the white box, there is a link that says 'Vissza a Főoldalra'.

KAPCSOLAT-MŰKÖDÉSE

- Az első mező az email cím megadására szolgál. Az **<label>** az email mezőhöz tartozik, az input mező **th:field="*{email}"** direktívája az adat bindingot végzi, hogy az űrlap kitöltött adatainak megfelelően átvegye a modellből az email értékét.
- A **value=""** mező üresen van hagyva, tehát a felhasználónak kell kitöltenie.
- Az **required** attribútum biztosítja, hogy az email mező kitöltése kötelező.
- A következő mező egy szövegdox (textarea), ahol a felhasználók üzenetet írhatnak.
- A **th:field="*{message}"** segítségével a szöveg tartalma a szerver oldali modellből (message) van bindelve.
- A gomb az űrlap elküldésére szolgál. A felhasználó a "Elküldés" gombra kattintva indítja el az űrlap adatainak beküldését a szerverre.
- A felhasználó, ha nem szeretne kapcsolatot felvenni, a "Vissza a Főoldalra" hivatkozásra kattintva visszatérhet a főoldalra.



ÜZENETEK ALMENÜ

ÜZENETEK - NÉZET

Az üzenetek gomb lenyomásához csak az adminnak van jogosultási , más esetben egy egyszerű felhasználónak nem fogja megtalálni a keresett oldalt.

Tartalom

Szerkeszt

SQL

Keresés

Beszűrés

Exportálás

Importálás

Jogok

Műveletek

Eseményindító

✓ Sorok megjelenítése 0-11 (összesen 12, A lekérdezés 0,0003 másodpercig tartott.)

SELECT * FROM `contact`

☐ Adatgyűjtés

Szerkesztés helyben

Módosítás

Az SQL magyarázata

PHP-kód létrehozása

Frissítés

☐ Összes megjelenítése

Sorok száma: 25

Sorok szűrése: Keresés a táblában

Sort by key: Nincs

Extra options

←

→

▼

id

name

email

message

created_at

☐

Módosítás

Másolás

Törlés

17

Guest

user@gmail.com

asdasdasd

2024-11-14 21:03:09

☐

Módosítás

Másolás

Törlés

18

User

user@gmail.com

asdasdfsasd

2024-11-14 21:14:31

☐

Módosítás

Másolás

Törlés

19

User

user@gmail.com

sziaaa

2024-11-14 21:36:48

☐

Módosítás

Másolás

Törlés

21

User

user@gmail.com

asdsdfgdfghdfhjghjgh

2024-11-15 18:13:49

☐

Módosítás

Másolás

Törlés

22

Guest

teszt@gmail.com

afhghfghdfghjfgjdjhgdjgh

2024-11-15 18:14:18

☐

Módosítás

Másolás

Törlés

23

User

user@gmail.com

ggggggggggg

2024-11-15 18:15:49

☐

Módosítás

Másolás

Törlés

24

User

user@gmail.com

gdfghdfggggggggggggggggg

2024-11-15 18:16:01

☐

Módosítás

Másolás

Törlés

25

Guest

user@gmail.com

ysfgsdghfghfghdfghd

2024-11-15 18:16:20

☐

Módosítás

Másolás

Törlés

26

Guest

teszt@gmail.com

asdsadsdasdasda

2024-11-15 18:16:46

☐

Módosítás

Másolás

Törlés

27

Guest

teszt@gmail.com

ffffffffffffffffffff

2024-11-15 18:17:43

☐

Módosítás

Másolás

Törlés

31

Valaki11

valaki11@gmail.com

fvujdnfvne

2024-11-17 12:42:26

A bejelentkezett felhasználók könnyen küldhetnek üzenetet az adminisztrátoroknak, legyen szó kérdésekről, javaslatokról vagy akár panaszok kezeléséről. Mivel a rendszer csak azokat a felhasználókat engedi üzenetet küldeni, akik be vannak jelentkezve, az adminisztrátorok biztosak lehetnek abban, hogy az üzenetek mögött valódi, azonosítható felhasználók állnak. Ezen felül, a bejelentkezett felhasználók által küldött üzenetek segítenek az adminisztrátoroknak a felhasználói adatkezelés hatékonyabbá tételében is, mivel az üzenet tartalmán kívül a rendszer a feladó azonosítóját is rögzíti.

ÜZENETEK - MŰKÖDÉSE

Ha, mint admin szeretnénk az üzeneteket kezelni akkor ezekkel az adatokkal tudunk belépni :

Email: admin@gmail.com

Jelszó: jelszo1

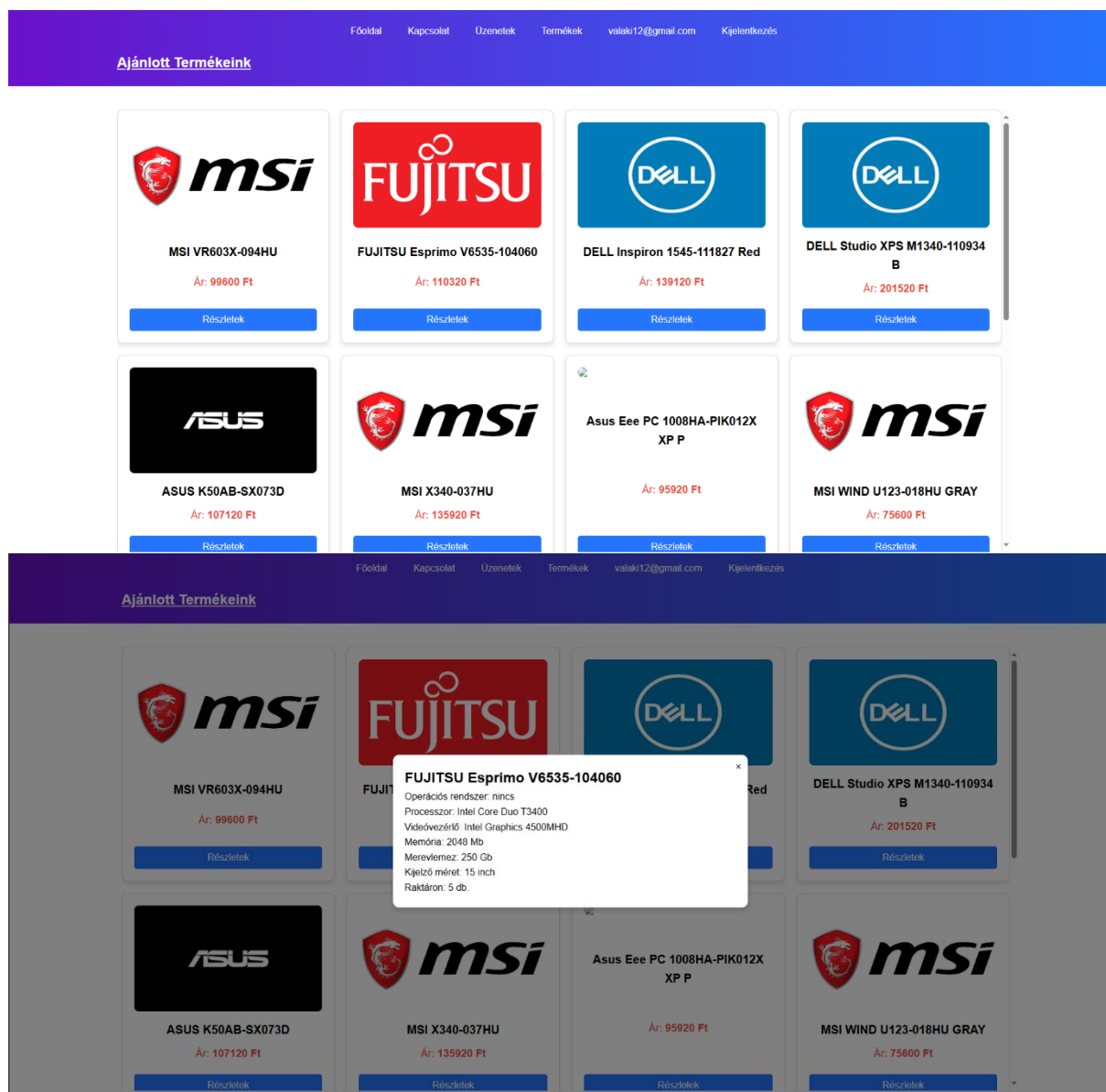
Más esetben hibát fogunk kapni és nem enged tovább minket az oldal.

```
document.addEventListener( type: "DOMContentLoaded", listener: function () :void {  
    const form :HTMLFormElement = document.querySelector( selectors: "form");  
    const emailInput :HTMLInputElement = document.getElementById( elementId: "email");  
    const errorMessage :HTMLInputElement = document.getElementById( elementId: "error-message");  
    const userEmail :string = document.getElementById( elementId: "userEmail").textContent;  
  
    // Kitöltjük az email input mezőt, ha az userEmail nem üres  
    if (userEmail) {  
        document.getElementById( elementId: "email").value = userEmail;  
        document.getElementById( elementId: "email").readOnly = true;  
    }  
    form.addEventListener( type: "submit", listener: function (event :SubmitEvent) :void {  
        let isValid :boolean = true;  
        let messages :any[] = [];  
  
        // Email formátum ellenőrzés  
        const emailPattern :RegExp = /^[^\\s@]+@[^\\s@]+\\.([^\\s@]+)$/;  
        if (!emailPattern.test(emailInput.value)) {  
            //alert("Kérlek, adj meg egy érvényes email címet!");  
            messages.push("Kérlek, adj meg egy érvényes email címet!");  
            isValid = false;  
        }  
  
        if (!isValid) {  
            event.preventDefault();  
            errorMessage.style.display = "block";  
            errorMessage.style.color = "red";  
            errorMessage.innerHTML = messages.join("<br>");  
        }  
    });  
});
```

TERMÉKEK ALMENÜ

TERMÉKEK - NÉZET

Az alábbi termékek fülön csak a bejelentkezett felhasználók számára elérhetőek a termékek, amelyeket szabadon megtekinthetünk. A termékek között való válogatás csak akkor lehetséges, ha a felhasználó be van jelentkezve. Ha rákattintunk a „Részletek” gombra, akkor a termékek adatait tekinthetjük meg. Ezáltal biztosítva van, hogy csak azok férnek hozzá a termékek részletes információihoz, akik regisztráltak és bejelentkeztek az oldalra.



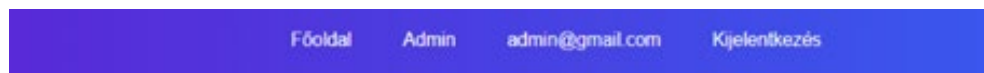
TERMÉKEK - MŰKÖDÉSE

Amennyiben a felhasználó kiválasztott egy számára tetszőleges Notebookot a listából és rányom a Részletek gombra, meghívódik a details.js fájl, ami megjeleníti az adatbázisból a kívánt adatokat. Miután ez megtörtént, kiléphetünk és vissza állítja a nézetet a kiindulási helyzetbe.

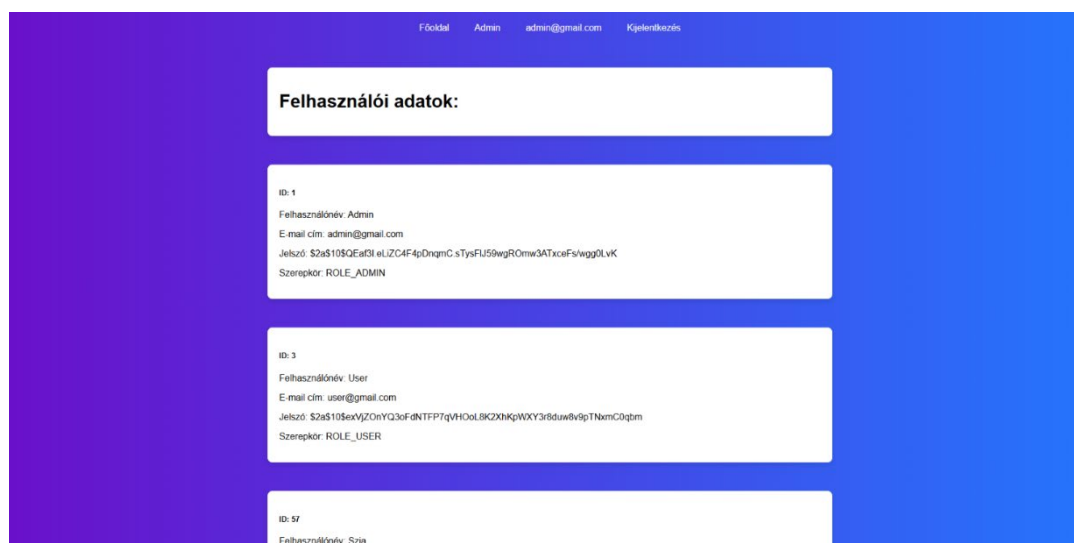
```
1
2
3
4 function reszletek(a) {
5     // A gombbal azonos sorban lévő "hiddenData" kiválasztása
6     const hiddenData = a.closest(".product-card").querySelector("#hiddenData");
7
8     if (hiddenData) {
9         // HiddenData tartalmának másolása a popupba
10        const popupOverlay = document.getElementById("popupOverlay");
11        const popupInnerContent = document.getElementById("popupInnerContent");
12
13        popupInnerContent.innerHTML = hiddenData.innerHTML; // Tartalom másolása
14        popupOverlay.style.display = "block"; // Popup megjelenítése
15    }
16 }
17
18 function closePopup() {
19     // Popup bezárása
20     const popupOverlay = document.getElementById("popupOverlay");
21     popupOverlay.style.display = "none";
22 }
```

ADMIN ALMENÜ

Ezen az oldalon kizárólag az üzemeltető tud művelteket végrehajtani.



ADMIN NÉZET



Az admin felület egyik legfontosabb funkciója az felhasználói adatok kezelése. A felhasználók a kapcsolatfelvételi űrlap segítségével bármilyen kérdésekkel, visszajelzésekkel vagy panaszokkal kereshetik fel az üzemeltetőt. Az admin felületen az üzemeltetők számára biztosított egy egyszerű és átlátható rendszer, ahol minden beérkező üzenetet részletesen megtekinthetnek. Az üzenetek listázása időrendben történik, így az adminisztrátor mindig könnyedén hozzáférhet a legfrissebb üzenetekhez.

A felhasználói adatok kezelése szintén kulcsfontosságú része az adminisztrációs felületnek. Az adminisztrátorok számára lehetőséget biztosít arra, hogy nyomon követhessék a felhasználók regisztrációit, profiljaikat. Az admin felületen részletes adatokat találhatunk a felhasználók személyes adatairól, mint például a nevük, email címük, címük és egyéb profil információk. Ezen adatok kezelésére szigorú adatvédelmi előírások vonatkoznak, így biztosítva, hogy a felhasználók személyes adatai biztonságban legyenek.

RESTFUL API

Az alkalmazásban létre lett hozva egy RESTful API, amit a ContactApiController vezérel. Ebben a controllerben 5 fajta api hívás lett létrehozva, 2 GET, egy POST, egy PUT és egy DELETE.

```
@RestController
class ContactApiController {
    private final ContactRepository repo;
    ContactApiController(ContactRepository repo) { this.repo = repo; }
```

Az első GET, a /contactAPI útvonallal lekéri az adatbázisban eltárolt összes üzenetet az olvasMind módszer segítségével.

```
@GetMapping("/contactAPI")
Iterable<Contact> olvasMind() { return repo.findAll(); }
```

A második GET kérés, egy az url-ben megadott azonosító értéke alapján vesz ki egy adatot az adatbázisból az olvasEgy módszerrel. Itt, ha olyan olyan id értéket adunk át ami nem létezik, hibát dob.

```
@GetMapping("/contactAPI/{id}")
Contact olvasEgy(@PathVariable int id) {
    return repo.findById(id)
        .orElseThrow(() -> new ContactNotFoundException(id));
}
```

A POST kérés küldésénél a megadott adatokkal új üzenetet tudunk felvenni az adatbázisba a contactPost módszer segítségével

```
@PostMapping("/contactAPI")
Contact contactPost(@RequestBody Contact newContact) { return repo.save(newContact); }
```

A PUT kérés küldésével tudjuk módosítani az egyik eltárolt üzenetet a `contactPut` metódus meghívásával. Amennyiben az adott id-vel nem rendelkezik üzenet, úgy felveszi új elemként az adatbázisba az üzenetet

```
@PutMapping("/{id}")
Contact contactPut(@RequestBody Contact contactData, @PathVariable int id) {
    return repo.findById(id)
        .map(a -> {
            a.setName(contactData.getName());
            a.setEmail(contactData.getEmail());
            a.setMessage(contactData.getMessage());
            a.setCreated_at(contactData.getCreated_at());
            return repo.save(a);
        })
        .orElseGet(() -> {
            contactData.setId(id);
            return repo.save(contactData);
        });
}
```

A DELETE kérés küldésénél, amennyiben létező id lett megadva az URL-be, úgy azzal az azonosítóval rendelkező üzenet, törlésre kerül az adatbázisból.

```
@DeleteMapping("/{id}")
void contactDelete(@PathVariable int id) { repo.deleteById(id); }

static class ContactNotFoundException extends RuntimeException {
    1 usage
    ContactNotFoundException(int id) { super("A megadott id-vel nem létezik üzenet: " + id); }
}
```

Az alábbiakban meg a cURL és a Postman segítségével létrehozott kérések képernyőképei következnek

CURL

GET

```
C:\Windows\System32\cmd.exe

C:\Users\Tomi\Desktop\NJE GAMF\7. félév\Java alkalmazások\Gyakorlat\Beadando_Beta\notebook_web>curl localhost:8080/contactAPI
[{"id":17,"name":"Guest","email":"user@gmail.com","message":"asdasdasd","created_at":"2024-11-14T20:03:09.000+00:00"},{"id":18,"name":"User","email":"user@gmail.com","message":"asdasdfsasd","created_at":"2024-11-14T20:14:31.000+00:00"},{"id":19,"name":"User","email":"user@gmail.com","message":"sziaaaa","created_at":"2024-11-14T20:36:48.000+00:00"},{"id":20,"name":"Guest","email":"teszt@gmail.com","message":"Szia Lajos hogy vagy?","created_at":"2024-11-15T17:12:56.000+00:00"},{"id":21,"name":"User","email":"user@gmail.com","message":"asdsdfgdfghdfhjghjgh","created_at":"2024-11-15T17:13:49.000+00:00"},{"id":22,"name":"Guest","email":"teszt@gmail.com","message":"afhghfghdfghjfgjdjhgjdjgh","created_at":"2024-11-15T17:14:18.000+00:00"},{"id":23,"name":"User","email":"user@gmail.com","message":"ggggggggggg","created_at":"2024-11-15T17:15:49.000+00:00"},{"id":24,"name":"User","email":"user@gmail.com","message":"gdfghdfgggggggggggggggg","created_at":"2024-11-15T17:16:01.000+00:00"},{"id":25,"name":"Guest","email":"user@gmail.com","message":"ysfgsdghfghfghdfghd","created_at":"2024-11-15T17:16:20.000+00:00"},{"id":26,"name":"Guest","email":"teszt@gmail.com","message":"asdsadsdasdasda","created_at":"2024-11-15T17:16:46.000+00:00"},{"id":27,"name":"Guest","email":"teszt@gmail.com","message":"ffffffffffffffffffffffffffffffff","created_at":"2024-11-15T17:17:43.000+00:00"}]
C:\Users\Tomi\Desktop\NJE GAMF\7. félév\Java alkalmazások\Gyakorlat\Beadando_Beta\notebook_web>

C:\Windows\System32\cmd.exe

C:\Users\Tomi\Desktop\NJE GAMF\7. félév\Java alkalmazások\Gyakorlat\Beadando_Beta\notebook_web>curl -v localhost:8080/contactAPI
* Host localhost:8080 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
* Trying [::1]:8080...
* Connected to localhost (::1) port 8080
* GET /contactAPI HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/8.9.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 0
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY
< Content-Type: application/json
< Transfer-Encoding: chunked
< Date: Sat, 16 Nov 2024 13:56:31 GMT
<
[{"id":17,"name":"Guest","email":"user@gmail.com","message":"asdasdasd","created_at":"2024-11-14T20:03:09.000+00:00"},{"id":18,"name":"User","email":"user@gmail.com","message":"asdasdfsasd","created_at":"2024-11-14T20:14:31.000+00:00"},{"id":19,"name":"User","email":"user@gmail.com","message":"sziaaaa","created_at":"2024-11-14T20:36:48.000+00:00"},{"id":20,"name":"Guest","email":"teszt@gmail.com","message":"Szia Lajos hogy vagy?","created_at":"2024-11-15T17:12:56.000+00:00"},{"id":21,"name":"User","email":"user@gmail.com","message":"asdsdfgdfghdfhjghjgh","created_at":"2024-11-15T17:13:49.000+00:00"},{"id":22,"name":"Guest","email":"teszt@gmail.com","message":"afhghfghdfghjfgjdjhgjdjgh","created_at":"2024-11-15T17:14:18.000+00:00"},{"id":23,"name":"User","email":"user@gmail.com","message":"ggggggggggg","created_at":"2024-11-15T17:15:49.000+00:00"},{"id":24,"name":"User","email":"user@gmail.com","message":"gdfghdfgggggggggggggggg","created_at":"2024-11-15T17:16:01.000+00:00"},{"id":25,"name":"Guest","email":"user@gmail.com","message":"ysfgsdghfghfghdfghd","created_at":"2024-11-15T17:16:20.000+00:00"},{"id":26,"name":"Guest","email":"teszt@gmail.com","message":"asdsadsdasdasda","created_at":"2024-11-15T17:16:46.000+00:00"},{"id":27,"name":"Guest","email":"teszt@gmail.com","message":"ffffffffffffffffffffffffffffffff","created_at":"2024-11-15T17:17:43.000+00:00"}]* Connection #0 to host localhost left intact

C:\Windows\System32\cmd.exe

C:\Users\Tomi\Desktop\NJE GAMF\7. félév\Java alkalmazások\Gyakorlat\Beadando_Beta\notebook_web>curl localhost:8080/contactAPI/20
{"id":20,"name":"Guest","email":"teszt@gmail.com","message":"Szia Lajos hogy vagy?","created_at":"2024-11-15T17:12:56.000+00:00"}
C:\Users\Tomi\Desktop\NJE GAMF\7. félév\Java alkalmazások\Gyakorlat\Beadando_Beta\notebook_web>

C:\Windows\System32\cmd.exe

C:\Users\Tomi\Desktop\NJE GAMF\7. félév\Java alkalmazások\Gyakorlat\Beadando_Beta\notebook_web>curl localhost:8080/contactAPI/888
com.example.RestAPI.ContactApiController$ContactNotFoundException: A megadott id-vel nem létezik üzenet: 888
```

POST

```
C:\Users\Tomi\Desktop\NJE GAMF\7. félév\Java alkalmazások\Gyakorlat\Beadando_Beta\notebook_web>curl -X POST localhost:8080/contactAPI -H "Content-type:application/json" -d '{"name":"Lajos","email":"lajos@gmail.com","message":"Szia Lajos, kutyaiddat setaltatod?","created_at":"2024-11-16T15:09:09Z"}'
{"id":28,"name":"Lajos","email":"lajos@gmail.com","message":"Szia Lajos, kutyaiddat setaltatod?","created_at":"2024-11-16T15:09:09.000+00:00"}
```

☐ Módosítás Másolás Törlés 28 Lajos lajos@gmail.com Szia Lajos, kutyaiddat setaltatod? 2024-11-16 16:09:09

PUT

```
C:\Windows\System32\cmd.exe
C:\Users\Tomi\Desktop\NJE GAMF\7. f  l  v\Java alkalmaz  sok\Gyakorlat\Beadando_Beta\notebook_web>curl -X PUT localhost:8080/contactAPI/28 -H "Content-type:application/json" -d '{"name":"LajosKAA","email":"lajosKAA@gmail.com","message":"Szia LajosKAA, kutya  dat setaltatod?"}'
{"id":28,"name":"LajosKAA","email":"lajosKAA@gmail.com","message":"Szia LajosKAA, kutya  dat setaltatod?","created_at":"2024-11-16T15:09:09.000+00:00"}
C:\Users\Tomi\Desktop\NJE GAMF\7. f  l  v\Java alkalmaz  sok\Gyakorlat\Beadando_Beta\notebook_web>
```

☐ M  dos  t  s M  sol  s T  rl  s 28 LajosKAA lajosKAA@gmail.com Szia LajosKAA, kutya  dat setaltatod? 2024-11-16 16:09:09

DELETE

```
C:\Users\Tomi\Desktop\NJE GAMF\7. f  l  v\Java alkalmaz  sok\Gyakorlat\Beadando_Beta\notebook_web>curl -X DELETE localhost:8080/contactAPI/28
C:\Users\Tomi\Desktop\NJE GAMF\7. f  l  v\Java alkalmaz  sok\Gyakorlat\Beadando_Beta\notebook_web>
```

SELECT * FROM `contact`

☐ Adatgy  jt  s [Szerkeszt  s helyben] [M  dos  t  s] [Az SQL magy  r  zata] [PHP-k  d l  trehoz  sa] [Friss  t  s]

☐   sszes megjelen  t  se | Sorok s  ma: 25 v | Sorok sz  r  se: Keres  s a t  bl  ban | Sort by key: Nir

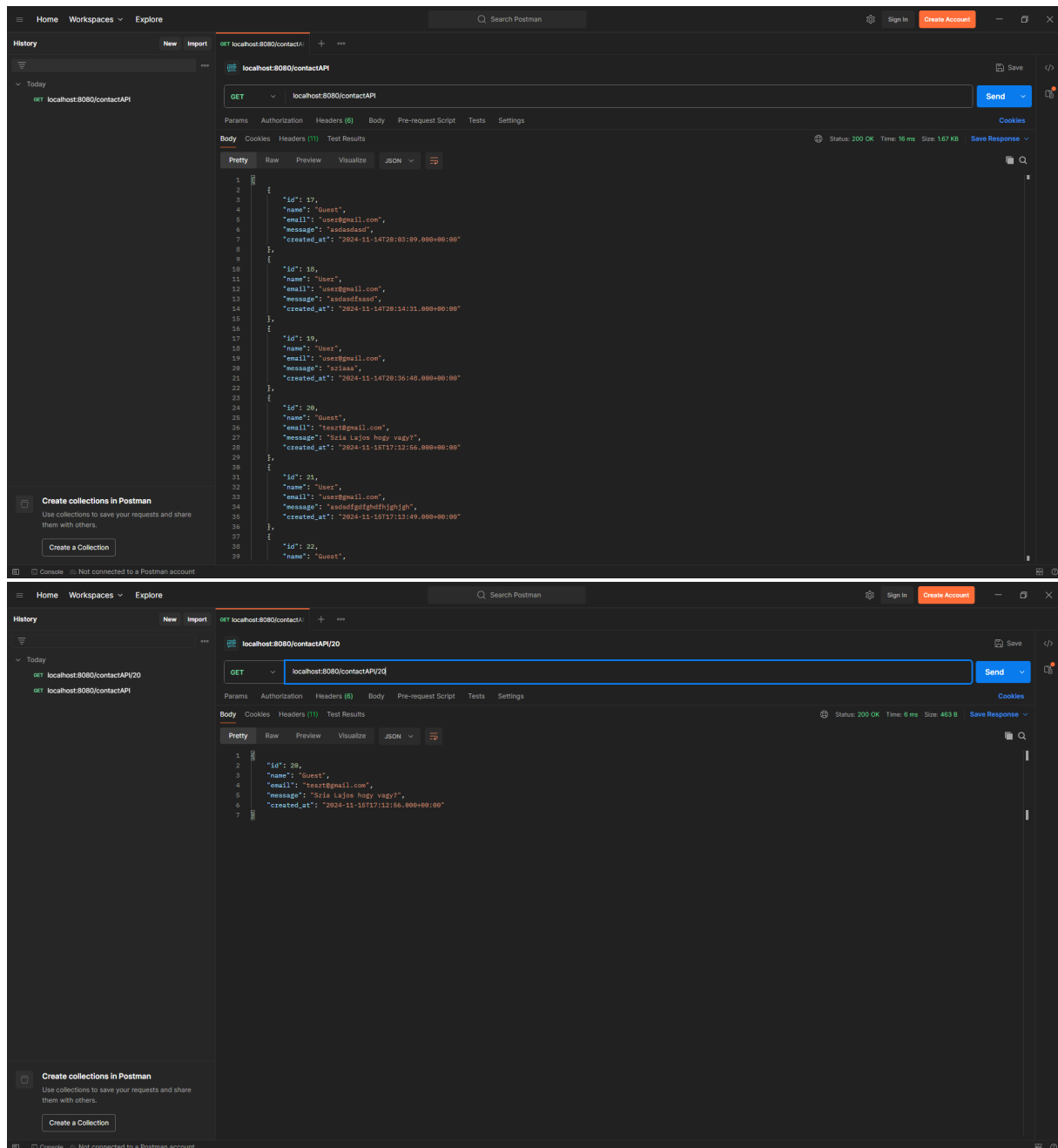
Extra options

				id	name	email	message	created_at			
<input type="checkbox"/>		M��dos��t��s		M��sol��s		T��rl��s	17	Guest	user@gmail.com	asdasdasd	2024-11-14 21:03:09
<input type="checkbox"/>		M��dos��t��s		M��sol��s		T��rl��s	18	User	user@gmail.com	asdasdfsasd	2024-11-14 21:14:31
<input type="checkbox"/>		M��dos��t��s		M��sol��s		T��rl��s	19	User	user@gmail.com	sziaaa	2024-11-14 21:36:48
<input type="checkbox"/>		M��dos��t��s		M��sol��s		T��rl��s	20	Guest	teszt@gmail.com	Szia Lajos hogy vagy?	2024-11-15 18:12:56
<input type="checkbox"/>		M��dos��t��s		M��sol��s		T��rl��s	21	User	user@gmail.com	asdsdfgdfghdfhjghjgh	2024-11-15 18:13:49
<input type="checkbox"/>		M��dos��t��s		M��sol��s		T��rl��s	22	Guest	teszt@gmail.com	afghghghdfghjfgjdjhgdjgh	2024-11-15 18:14:18
<input type="checkbox"/>		M��dos��t��s		M��sol��s		T��rl��s	23	User	user@gmail.com	ggggggggggg	2024-11-15 18:15:49
<input type="checkbox"/>		M��dos��t��s		M��sol��s		T��rl��s	24	User	user@gmail.com	gdfghdfggggggggggggggggg	2024-11-15 18:16:01
<input type="checkbox"/>		M��dos��t��s		M��sol��s		T��rl��s	25	Guest	user@gmail.com	ysfgsdghfghfghdfghd	2024-11-15 18:16:20
<input type="checkbox"/>		M��dos��t��s		M��sol��s		T��rl��s	26	Guest	teszt@gmail.com	asdsadsdasdasda	2024-11-15 18:16:46
<input type="checkbox"/>		M��dos��t��s		M��sol��s		T��rl��s	27	Guest	teszt@gmail.com	ffffffffffffffffffffffff	2024-11-15 18:17:43

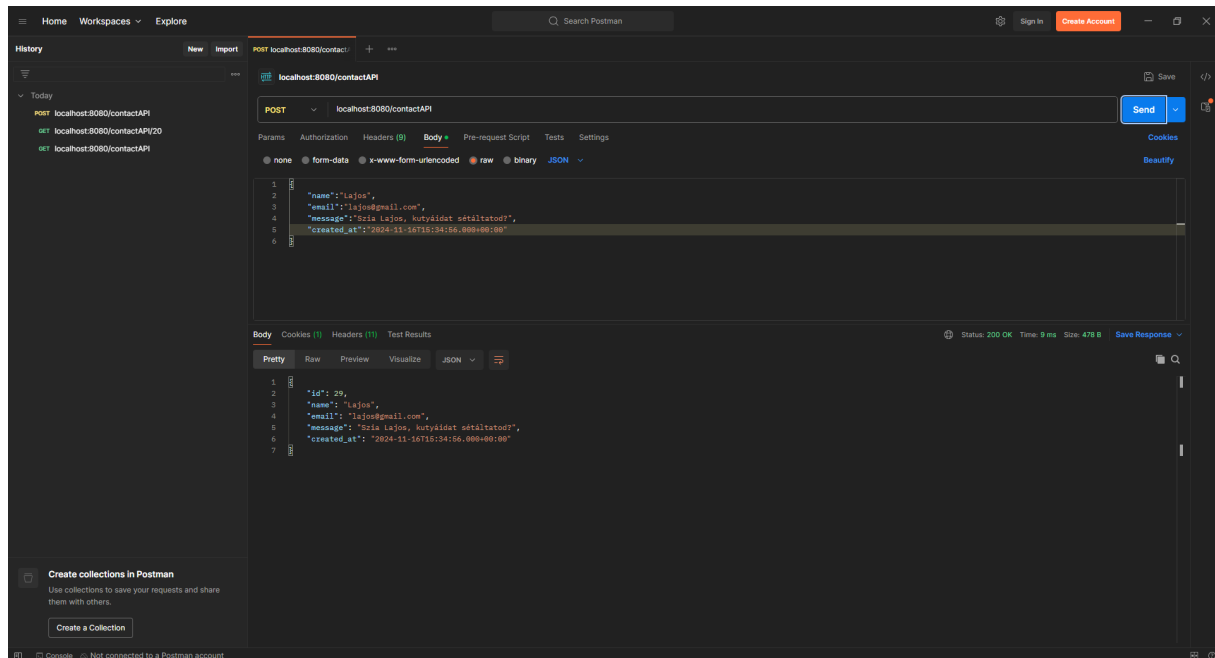
☐   sszes bejel  l  se A kijel  ttekkel v  gzend   m  velet: M  dos  t  s M  sol  s T  rl  s Exp

POSTMAN

GET



POST



Postman interface showing a POST request to `localhost:8080/contactAPI`. The request body is a JSON object with the following fields:

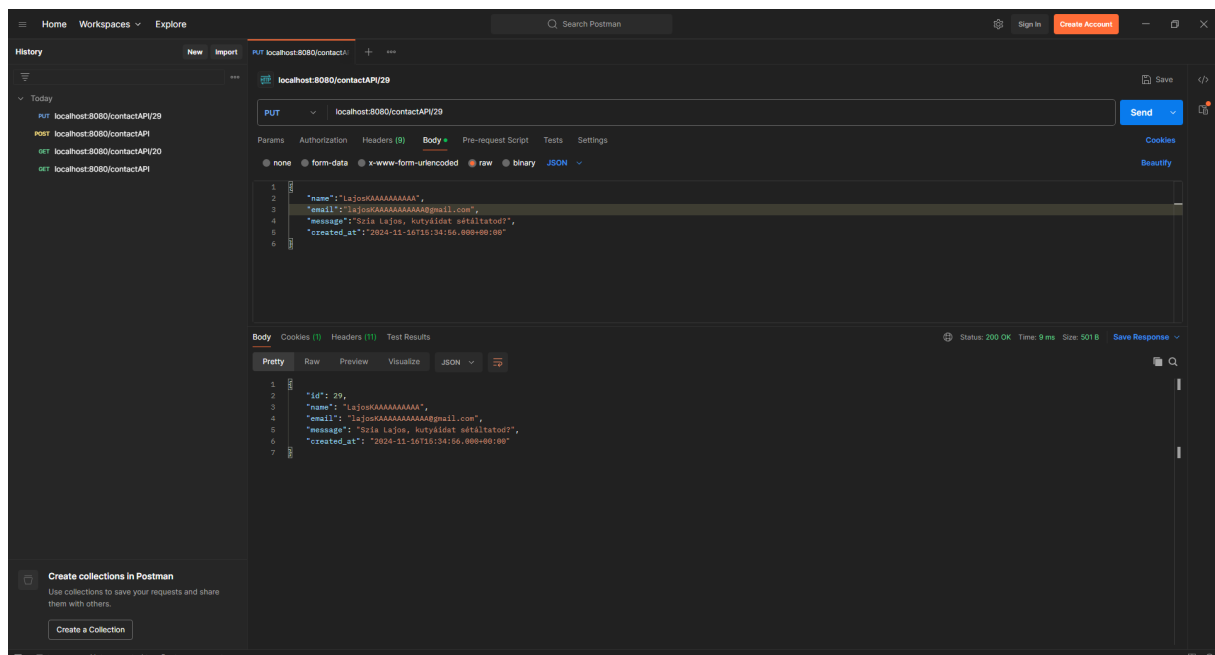
```
{  "name": "Lajos",  "email": "lajos@gmail.com",  "message": "Szia Lajos, kutyáidat sétáltatod?",  "created_at": "2024-11-16T15:34:56.000+00:00"}
```

The response body is a JSON object with the following fields:

```
{  "id": 29,  "name": "Lajos",  "email": "lajos@gmail.com",  "message": "Szia Lajos, kutyáidat sétáltatod?",  "created_at": "2024-11-16T15:34:56.000+00:00"}
```

Below the interface, a toolbar contains the following items: `Módosítás`, `Másolás`, `Törlés`, `29`, `Lajos`, `lajos@gmail.com`, `Szia Lajos, kutyáidat sétáltatod?`, and `2024-11-16 16:34:56`.

PUT



Postman interface showing a PUT request to `localhost:8080/contactAPI/29`. The request body is a JSON object with the following fields:

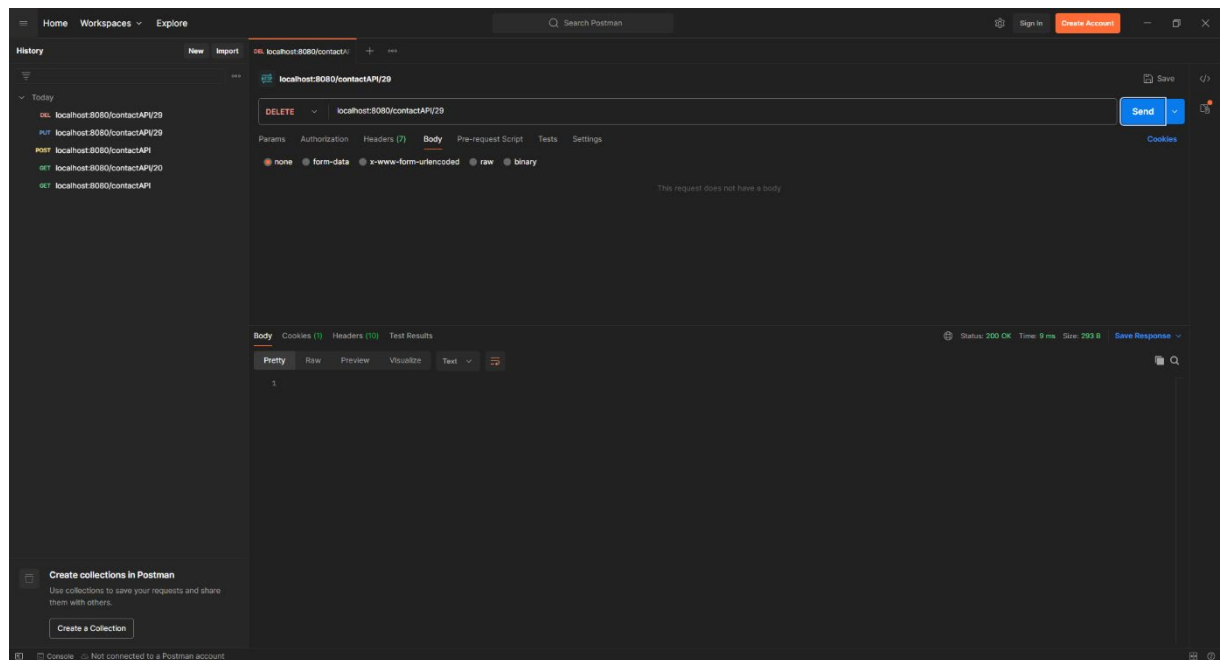
```
{  "name": "LajosKAAAAAAAAA",  "email": "lajosKAAAAAAAAA@gmail.com",  "message": "Szia Lajos, kutyáidat sétáltatod?",  "created_at": "2024-11-16T15:34:56.000+00:00"}
```

The response body is a JSON object with the following fields:

```
{  "id": 29,  "name": "LajosKAAAAAAAAA",  "email": "lajosKAAAAAAAAA@gmail.com",  "message": "Szia Lajos, kutyáidat sétáltatod?",  "created_at": "2024-11-16T15:34:56.000+00:00"}
```

Below the interface, a toolbar contains the following items: `Módosítás`, `Másolás`, `Törlés`, `29`, `LajosKAAAAAAAAA`, `lajosKAAAAAAAAA@gmail.com`, `Szia Lajos, kutyáidat sétáltatod?`, and `2024-11-16 16:34:56`.

DELETE



✓ Sorok megjelenítése 0-10 (összesen 11, A lekérdezés 0,0002 másodpercig tartott.)

`SELECT * FROM `contact``

Adatgyűjtés [Szerkesztés helyben] [Módosítás] [Az SQL magyarázata] [PHP-kód létrehozása] [Frissítés]

☐ Összes megjelenítése | Sorok száma: 25 | Sorok szűrése: Keresés a táblában | Sort by key: Ninc

Extra options

				id	name	email	message	created_at
<input type="checkbox"/>	Módosítás	Másolás	Törlés	17	Guest	user@gmail.com	asdasdasd	2024-11-14 21:03:09
<input type="checkbox"/>	Módosítás	Másolás	Törlés	18	User	user@gmail.com	asdasdfsasd	2024-11-14 21:14:31
<input type="checkbox"/>	Módosítás	Másolás	Törlés	19	User	user@gmail.com	sziAAA	2024-11-14 21:36:48
<input type="checkbox"/>	Módosítás	Másolás	Törlés	20	Guest	teszt@gmail.com	Szia Lajos hogy vagy?	2024-11-15 18:12:56
<input type="checkbox"/>	Módosítás	Másolás	Törlés	21	User	user@gmail.com	asdsdfghdfghjghjgh	2024-11-15 18:13:49
<input type="checkbox"/>	Módosítás	Másolás	Törlés	22	Guest	teszt@gmail.com	afhghfghdfghjfgjdjhgdjgh	2024-11-15 18:14:18
<input type="checkbox"/>	Módosítás	Másolás	Törlés	23	User	user@gmail.com	ggggggggggg	2024-11-15 18:15:49
<input type="checkbox"/>	Módosítás	Másolás	Törlés	24	User	user@gmail.com	gdfghdfggggggggggggggg	2024-11-15 18:16:01
<input type="checkbox"/>	Módosítás	Másolás	Törlés	25	Guest	user@gmail.com	ysfgsdghfghfghdfghd	2024-11-15 18:16:20
<input type="checkbox"/>	Módosítás	Másolás	Törlés	26	Guest	teszt@gmail.com	asdsadsdasdasda	2024-11-15 18:16:46
<input type="checkbox"/>	Módosítás	Másolás	Törlés	27	Guest	teszt@gmail.com	ffffffffffffffffffffffff	2024-11-15 18:17:43

↑ ☐ Összes bejelölése A kijelöltekkel végzendő művelet: ☐ Módosítás ☐ Másolás ☐ Törlés ☐ Export

FELADATMEGOLDÁSBAN HASZNÁLT ELEMEEK

MEGJELENÍTÉSEK

- **static:** Az alkalmazás statikus fájljait (pl. CSS, JavaScript, képek) tartalmazza.

CSS MAPPA

Ez a mappa tartalmazza a stíluslapokat (CSS fájlokat), amelyek az oldal kinézetét és elrendezését határozzák meg:

- **admin.css:** Az adminisztrációs felülethez kapcsolódó stílusokat tartalmazza.
- **contact.css:** A kapcsolati oldal (a kapcsolattartási űrlap) megjelenéséért felel.
- **login.css:** A bejelentkezési oldal stílusai.
- **menu.css:** A menük elrendezését és formázását kezeli.
- **popup.css:** Felugró ablakok (popup) kinézetéért felel.
- **products.css:** A termékek megjelenítéséhez kapcsolódó stílusok.
- **register.css:** A regisztrációs űrlap formázása.
- **style.css:** Egy általános stíluslap lehet az egész alkalmazáshoz.

IMAGES MAPPA

Ez a mappa képeket tárol, amelyeket az alkalmazásban megjelenítenek.

- **brand:** Egy alkönyvtár, amely különböző notebook márkák logóit tartalmazza (pl. **ACER.jpg**, **ASUS.jpg**, **DELL.jpg**).
- Egyéb képek:
 - **lotpic.jpg**, **macpic.jpg**, **scrollpic.jpg**: termékfotók
 - **webicon.jpg**: az oldal ikonja

JS MAPPA

Ez a mappa tartalmazza az alkalmazás frontendi JavaScript fájljait, amelyek az interaktív elemeket kezelik:

- **contact.js:** A kapcsolati oldal JavaScript logikáját kezeli.
- **details.js:** Termékek részletes megjelenítéséhez kapcsolódó funkcionalitást biztosít.
- **register.js:** A regisztrációs űrlapok validációjához/interakcióihoz kapcsolódó kód.

EGYÉB MAPPÁK

- **icon:** Ikont tartalmaz, amely a felhasználói felületen jelenik meg.

TEMPLATES MAPPA

A **templates** mappa a HTML sablonfájlokat tartalmazza, amelyeket a Spring Boot a dinamikus tartalom megjelenítésére használ. Ezek a sablonok a backendből érkező adatokat megjelenítik a felhasználók számára.

HTML FÁJLOK

- **admin.html:** Az adminisztrációs felület sablonja.
- **contact.html:** A kapcsolattartási oldal sablonja, amely egy űrlapot vagy információkat jeleníthet meg.
- **contactList.html:** Egy listaoldal, amely kapcsolatok adatait jeleníti meg.
- **index.html:** Az alkalmazás kezdőlapja.
- **login.html:** A bejelentkezési oldal sablonja.
- **menu.html:** Valószínűleg a navigációs menüt tartalmazza.
- **products.html:** A termékek listájának vagy részletes megjelenítésének sablonja.
- **register.html:** A regisztrációs oldal sablonja.
- **user.html:** A felhasználói profil adatok megjelenítéséhez tartozó sablon.

