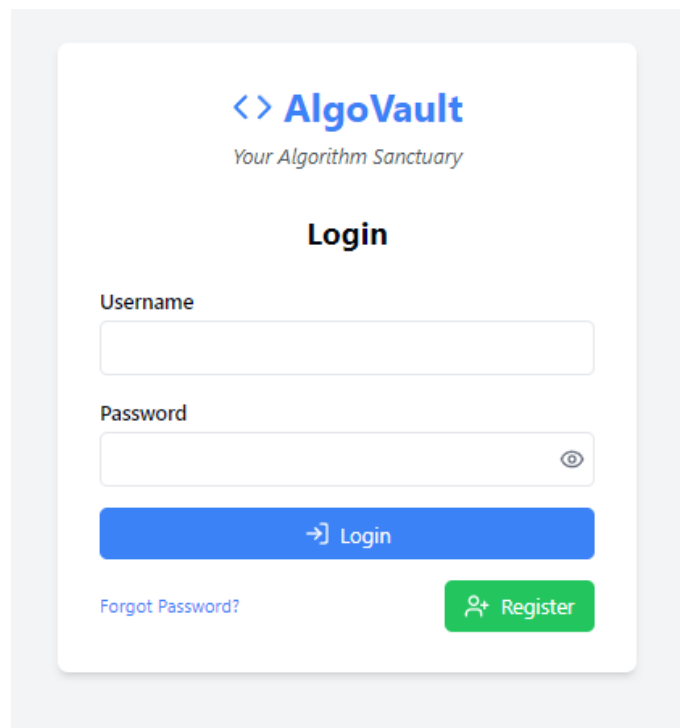


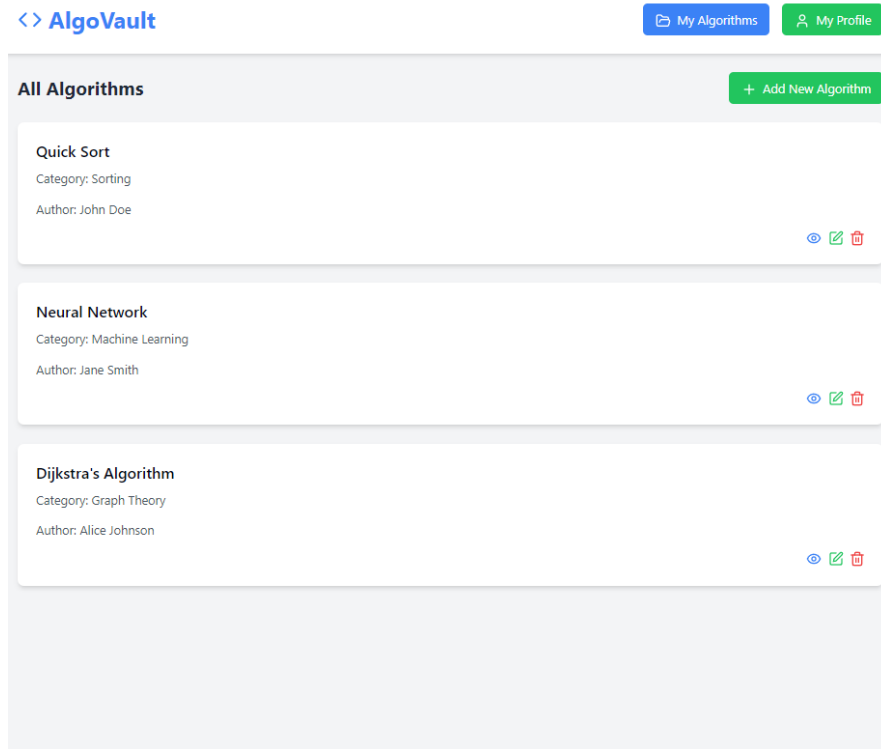
# AlgoVault

A webalkalmazásban algoritmusok és felhasználók tárolására van lehetőség. Minden algoritmusról tároljuk, hogy ki hozta létre, mi a neve, a kategóriája (például gráfelmélet vagy rendezés), rövid leírása, pszeudokódja, időigénye (time complexity), és tárhely igénye (space complexity). Minden felhasználóról tároljuk a felhasználónevét, jelszavát, a telefonszámát, a lakhelyét, és hogy miben specializálódott. Az egyszerűség kedvéért egy felhasználó csak egy kategóriában specializálódik (például gépi tanulás). Az **1. ábrán** látható a bejelentkezési felület, ahol a felhasználónév, és jelszó párossal tudunk belépni.

The image shows a login interface for 'AlgoVault', which is described as 'Your Algorithm Sanctuary'. The interface is clean and modern, with a white background and a light gray border. At the top, the logo consists of a blue '<>' symbol followed by the text 'AlgoVault' in a bold, sans-serif font. Below the logo is the tagline 'Your Algorithm Sanctuary' in a smaller, italicized font. The main heading is 'Login' in a bold, black font. There are two input fields: 'Username' and 'Password'. The 'Password' field has a small eye icon to its right for toggling visibility. Below the input fields is a large blue button with a white right-pointing arrow and the text 'Login'. At the bottom left, there is a link 'Forgot Password?' in a small, blue font. At the bottom right, there is a green button with a white user icon and the text 'Register'.

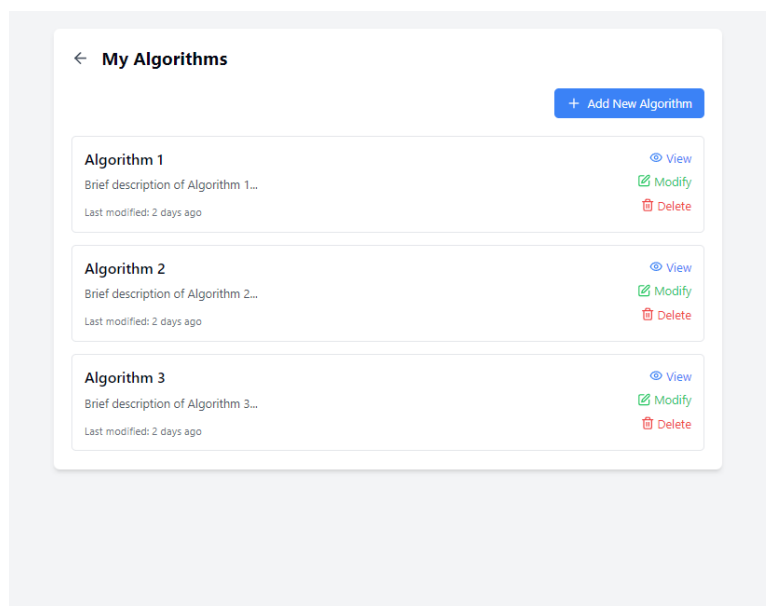
**1. ábra.** Bejelentkezési képernyő

A **2. ábrán** látható képernyő fogad majd minket bejelentkezés után, ahol minden felhasználó algoritmusát láthatjuk. Az algoritmusokat megtudjuk részletesen tekinteni, módosítani, törölni, és újat is létre tudunk hozni.



**2. ábra.** Főképernyő

Ha a **2. ábrán** a „My Algorithms” menüpontra kattintunk akkor a **3. ábrán** lévő felület fogad minket. Ha szintén a **2. ábrán** lévő „My Profile”-ra kattintunk akkor pedig a **4. ábrán** lévő képernyőt láthatjuk.



**3. ábra.** Saját algoritmusok

**← User Profile**

Username:

Password:

Phone Number:

Address:

Specialized Field:

[Edit Profile](#)

[Delete Profile](#)

#### 4. ábra. Profil módosítása

Egy felhasználói fiókot tehát a DB adminisztrátoron kívül maga a felhasználó is tudja módosítani, vagy törölni. Új fiókot vagy a DB adminisztrátor tud létrehozni, vagy a bejelentkező képernyőn a „Register” gombra kattintva a **8. ábrán** lévő nézetben tudunk fiókot létrehozni. Az **5. ábra** egy új algoritmus létrehozásakor a **6. ábra** pedig a megtekintésekor előjövő nézetet mutatja be. A **7. ábrát** egy algoritmus módosításánál láthatjuk majd. A programban szereplő összes legördülő lista esetén egy backend oldalon definiált statikus listát használok majd. A jelenlegi fehér stílus még nem végleges időtől függően remélem összetudok majd rakni valami izgalmasabb színvilágú alkalmazást.

← <> AlgoVault

## Add New Algorithm

Algorithm Name

Category

Select a category

Description


Pseudocode

Time Complexity

e.g.,  $O(n \log n)$

Space Complexity

e.g.,  $O(n)$


 Save Algorithm


### 5. ábra. Új algoritmus létrehozása


← [AlgoVault](#)


## Quick Sort

<> Category: Sorting

 Author: John Doe

 Time Complexity:  $O(n \log n)$  on average,  $O(n^2)$  worst case

 Space Complexity:  $O(\log n)$

 Date Added: 2023-05-15

### Description

QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot.

### Pseudocode

```
quickSort(arr[], low, high)
    if (low < high)
        pi = partition(arr, low, high)
        quickSort(arr, low, pi - 1)
        quickSort(arr, pi + 1, high)

partition(arr[], low, high)
    pivot = arr[high]
    i = (low - 1)
    for j = low to high - 1
        if (arr[j] < pivot)
            i++
            swap arr[i] and arr[j]
    swap arr[i + 1] and arr[high]
    return (i + 1)
```

Edit Algorithm

Delete Algorithm

### 6. ábra. Algoritmus részletek

← <> AlgoVault

### Edit Algorithm: Quick Sort

Algorithm Name

Category

Description

QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot.

Pseudocode

```

quickSort(arr[], low, high)
  if (low < high)
    pi = partition(arr, low, high)
    quickSort(arr, low, pi - 1)
    quickSort(arr, pi + 1, high)

```

Time Complexity

Space Complexity

Delete Algorithm
Save Changes

7. ábra. (felül) és 8. ábra (alul) Algoritmus módosítása & Regisztráció

<> AlgoVault
Your Algorithm Sanctuary

## Register

Username

Email

Password

Confirm Password

Phone Number

Address

Specialized Field

Register
Back to Login