

# Commandash



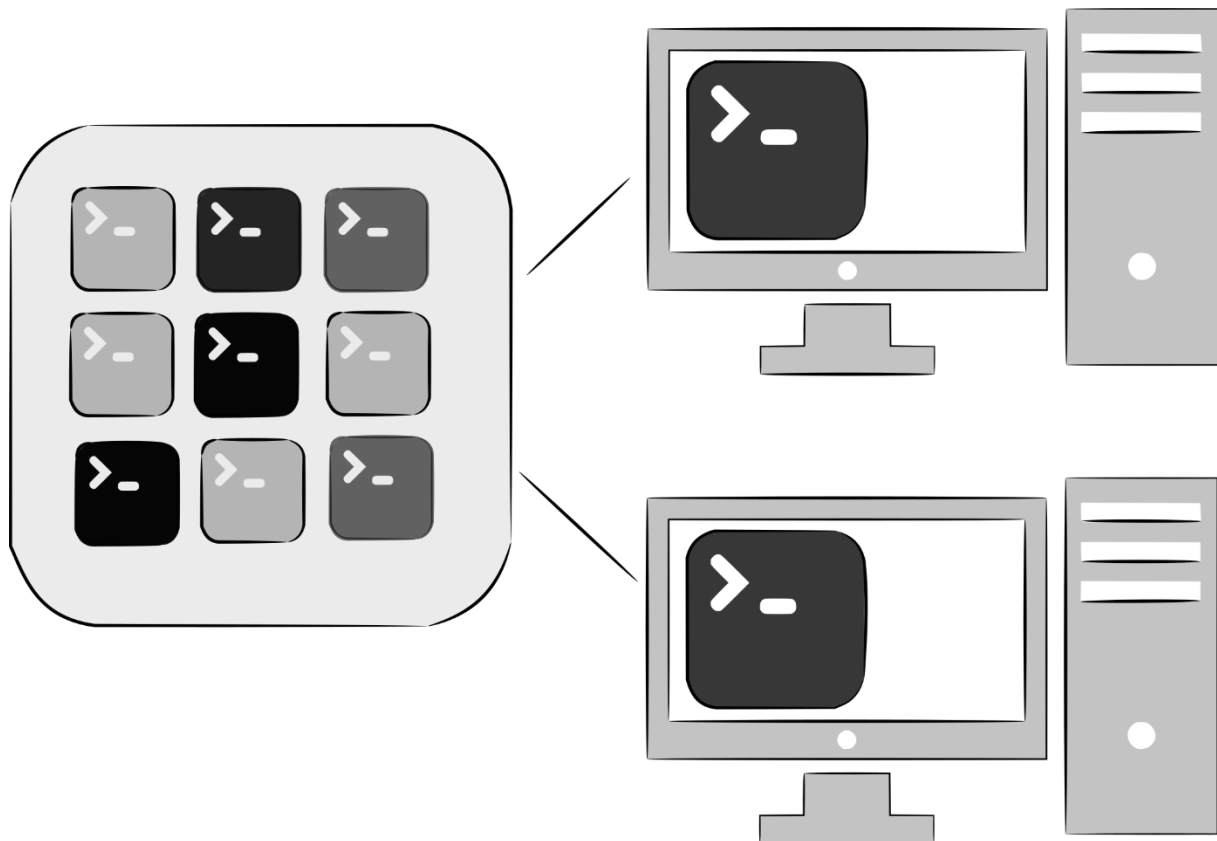
**Informatikai Biztonsági és  
Adatvédelmi Tanácsadó Kft.**

# 1 Tartalomjegyzék

1	Tartalomjegyzék.....	2
2	Célkitűzés.....	3
3	Architektúra.....	4
3.1	Frontend.....	4
3.2	Backend .....	4
3.3	Adatbázis.....	4
4	Funkciók.....	5
4.1	Menü .....	5
4.2	Admin felület .....	5
4.2.1	Jogosultságkezelés .....	5
4.3	Dashboard.....	5
4.3.1	Szerkesztő felület .....	6
5	Dokumentáció .....	7
5.1	Erőforrásterv, munkaidő-nyilvántartás .....	7
5.2	Üzemeltetői dokumentáció .....	7
5.3	Forráskód dokumentáció.....	7
5.4	Felhasználói dokumentáció .....	7
6	A projekt értékelése.....	8
6.1	A feladat értékelésének felhasználó oldali szempontjai.....	8
6.2	A feladat értékelésének technikai szempontjai.....	8
6.3	Projekt megvalósításának piaci jellegű értékelése .....	8
7	Projekt adatlap.....	9

## 2 Célkitűzés

Az alkalmazás célja egy olyan dinamikusan építhető webfelület készítése beépített szerkesztővel, amely lehetővé teszi a távoli számítógépeken való parancsfuttatást.



## 3 Architektúra

### 3.1 Frontend

A frontend feladata, hogy grafikusan megjelenítse a backend funkcionalitását, illetve hogy lehetőséget biztosítson dashboard(ok) készítésére. Technológiai kötöttség nincs, de ajánlott [Angular 10+](#) vagy [Flutter](#) használata, mivel könnyen fejleszthető framework és egyszerűen létrehozható a dinamikus építhető felület is benne.

### 3.2 Backend

A backend feladata a webfelület kiszolgálása adatokkal, illetve az onnan érkező parancsok elküldése a megfelelő szerver felé ssh kapcsolattal, vagy azok lokális gépen való futtatása. Technológiai kötöttség nincs, amíg a választott nyelv képes távoli ssh parancs kiadására, vagy a host-ról indított parancs kimenetének olvasására.

### 3.3 Adatbázis

A feladat megoldásához bármilyen adatbázis alkalmazható. A program által használt összes adat az adatbázisban kerül tárolásra.

## 4 Funkciók

### 4.1 Menü

Az alkalmazás felső menürendszerét használ, ahol a Dashboard, az Admin felület, és a bejelentkezés gomb található meg.

### 4.2 Admin felület

#### 4.2.1 Jogosultságkezelés

Bejelentkezés után Admin szerepkörrel érhető el.

##### 4.2.1.1 Szerepkörök

Szerepkör felvételére alkalmas felületet nem kell létrehozni, a következő szerepkörök felvétele szükséges az adatbázisba:

- Admin (minden funkció elérhető)
- Editor (minden funkció elérhető, a felhasználókezelés kivételével)
- Member (a privát és publikus dashboardokat megtekintheti)
- Guest (nem autentikált felhasználó, csak a publikus dashboardokat tekintheti meg)

##### 4.2.1.2 Felhasználókezelés

Lehetőséget kell biztosítani új felhasználó felvételére, illetve a meglévő szerkesztésére és törlésére.

Egy felhasználó csak egy szerepkörbe tartozhat.

##### 4.2.1.3 Szerver lista

Ezen a felületen lehet az alkalmazáshoz szervereket hozzáadni, szerkeszteni és törölni.

Egy szerverről a következő adatokat kell elmenteni:

- Név (a listában megjelenő név)
- Cím (IP cím vagy FQDN)
- Felhasználónév (opcionális)
- Jelszó (opcionális)

### 4.3 Dashboard

Az alkalmazás kezdőképernyője.

Az alkalmazásban több dashboard készítésére is lehetőséget kell biztosítani. Az elkészült dashboardok közül lehet alapértelmezettet választani, ami a weboldal megnyitásakor betöltődik.

Készíthető publikus és privát dashboard is, mindkét típus ugyanabból a listából érhető el, de a privát csak autentikáció után jelenik meg.

Ha a dashboardhoz tartozó szervernek nincs kitöltve a felhasználóneve, vagy jelszava, akkor a dashboard felületén meg kell jeleníteni beviteli mező(ket) a hiányzó adat(ok) megadására.

#### 4.3.1 Szerkesztő felület

Bejelentkezés után Editor és Admin szerepkörökkel érhető el.

Dashboard hozzáadására, szerkesztésére és törlésére van lehetőség. Minden dashboard első beállítandó paramétere egy szerver, amit egy lenyíló listából lehet választani.

Egy dashboard több parancsot is tartalmazhat, a futtatást indító gomb a parancs előtt helyezkedik el. Egy szerkesztőn keresztül a következő felületi elemek adhatók hozzá egy parancshoz:

- Lenyíló lista
- Szabadszöveges mező
- Checkbox

A lenyíló listák elemeit a lista hozzáadása után közvetlenül a felületen lehet megadni. Minden felületi elem automatikusan kap egy azonosítót, amin keresztül a tartalma beilleszthető a parancsba. Ezek az azonosítók a parancsba beírva, a futtatáskor a backend-en kerülnek behelyettesítésre. A felületi elemek alatt, egy szövegdobozban adható meg a kívánt parancs.

##### 4.3.1.1 Példa a működésre

Parancs:

```
echo „$1”
```

A felületen egyetlen szabadszöveges mező helyezkedik el. A \$1 helyére bekerül a felhasználó által írt szöveg.

Parancs:

```
ls -l -a $1
```

A felületen egy szabadszöveges mező jelenik meg, ahova a mappa útvonalát lehet megadni.

Checkbox-szal választhatóak az „l” és „a” kapcsolók.

Parancs:

```
curl --request $1 -k -s -S --header "Content-Type: application/json" --  
data "{\"path\": \"$2\"}" "http://pelda.url/endpoint"
```

A felületen 2 lenyíló lista jelenik meg, az első lista (\$1) tartalma: POST, PUT, a második lista (\$2) tartalma: TEST1, TEST2, DEV.

## 5 Dokumentáció

### 5.1 Erőforrásterv, munkaidő-nyilvántartás

A specifikáció birtokában a projekt résztvevői készítsenek erőforrástervet. Ez tartalmazza a feladatban részt vevő projektagokat, akik legyenek hozzárendelve a tervezés során azonosított részfeladatokhoz. Minden részfeladat mellé kerüljön egy munkaidő ráfordítási becslés munkaóraban számolva. Ezt a tervet a tényleges fejlesztés előtt le kell adni. A feladat megoldása során az elvégzett munkáról készüljön nyilvántartás részfeladatonként és személyenként a tényleges munkaórák számának megjelölésével. A projekt végén a két dokumentum összehasonlításra, az eltérések elemzésre kerülnek.

### 5.2 Üzemeltetői dokumentáció

Az üzemeltetői dokumentáció célja, hogy a rendszer üzemeltetőinek támogatást adjon a termék üzemeltetésének elsajátításához. Tartalmazza többek között a rendszer architekturális felépítését (alkalmazás stack elemei és azok közti kapcsolat leírása), az alkalmazás fordításához, fejlesztéséhez és futtatásához szükséges követelményeket, technológiákat, továbbá a konfigurációs állományok leírását (ha vannak).

A dokumentációnak a feladat bonyolultságától függő hosszúságúnak kell lennie, maximális terjedelem nincs meghatározva.

### 5.3 Forráskód dokumentáció

A fontosabb függvények és osztályok előtt szerepelnie kell megjegyzéseknek, melyeknek tartalmazniuk kell az azt követő metódus rövid szöveges – akár magyar nyelvű – leírását. A forráskód dokumentációt a munka során folyamatosan kell készíteni.

### 5.4 Felhasználói dokumentáció

Az alkalmazás használatának részletes bemutatása, képernyőképekkel, funkciók pontos leírásával.

## 6 A projekt értékelése

### 6.1 A feladat értékelésének felhasználó oldali szempontjai

A működő alkalmazás tesztelése alapján az alábbiak a legfontosabb jellemzők:

- Kiírást teljes egészében lefedő funkcionalitás
- Kényelmes használat
- Igényes felhasználói felület
- Stabil működés
- Igényes felhasználói dokumentáció

### 6.2 A feladat értékelésének technikai szempontjai

Informatikai szakmai szempontból a megoldás értékelésének alapja:

- Kódkép, a kód tisztasága, kommentelés minősége
- Kódolási konvenciók betartása (Clean Code)
- Dokumentációk színvonala
- Dokumentált tesztelés
- Erőforrás felhasználásának pontos nyilvántartása

### 6.3 Projekt megvalósításának piaci jellegű értékelése

A projekt lezárultával összehasonlításra kerül a kezdeti erőforrásterv, valamint a megvalósítás során dokumentált munka. Ezen dokumentumok elemzéséből levezetésre kerülnek azok a problémák, melyek a piaci környezetben jellemzően megjelennek. Végigtekintjük ezen problémák okait, következményeit, lehetséges elkerülésüknek vagy hatásuk mérséklésének módjait. A jellemző hibák ebből a megközelítésből:

- Határidő csúszása
- Nem megfelelő minőség
- Hiányos, vagy elmaradó tesztelés
- Használhatatlan, pontatlan dokumentáció
- Pontatlan erőforrás becslés
- Aránytalanul magas önköltség
- Az elkészült termék továbbfejlesztésének, karbantartásának nehézségei

A fentiek értékelésén túl fejlesztői szemszögből elemezzük a megvalósítás tapasztalatait, a lehetséges továbbfejlesztés, átalakítás, támogatás kérdéseit és piaci lehetőségeit.



## 7 Projekt adatlap

Projekt neve: Commandash

Feladat rövid ismertetése: Távoli számítógépeken való parancsfuttatást webes felületen keresztül.

Specifikációt összeállította: Selyem Ádám, Szabó Gábor