

# Asztali alkalmazás fejlesztői dokumentáció:

## Osztályok:

### DbConnect:

#### Osztály leírása

A DbConnect osztály az Isabike alkalmazás adatbázis elérését kezelő osztálya. Az osztály felel az alkalmazás és a szerver közötti kommunikációért, a felhasználó bejelentkezésért, a termékek lekérdezéséért és a szűrésért.

#### Függvények

##### **getData(string uri):**

- Leíró szöveg: Ez a függvény egy megadott végpontról kér le adatokat a backend szerverről Json formátumba.
- Paraméterek:
  - uri: A lekérdezni kívánt végpontból címe.
- Visszatérési érték:
  - A függvény a visszatérő Json adatból egy Jarray objektumot készít siker esetén. Sikertelen lekérdezés esetében **null** értékkel tér vissza.

##### **getUserData(string uri):**

- Leíró szöveg: Ez a függvény a felhasználói adatokat kér le a backend szerver megadott végpontjáról. A függvény a Login.getToken() metódus által visszaadott tokennel hitelesíti a felhasználót.
- Paraméterek:
  - uri: A lekérdezni kívánt végpont címe.
- Visszatérési érték:
  - A függvény a visszatérő Json adatból egy Jarray objektumot készít siker esetén. Sikertelen lekérdezés esetében **null** értékkel tér vissza.

### **getFilteredData(string uri, string terméknev, double suly, string gyarto):**

- Leíró szöveg: Ez a függvény egy megadott végpontról kér le adatokat a backend szerverről Json formátumba, majd szűri az adatot a megadott termék neve, súlya, és gyártója alapján
- Paraméterek:
  - uri: A lekérdezni kívánt URL címe.
  - terméknev: A keresett termék neve.
  - suly: A keresett termék súlya.
  - gyarto: A keresett termék gyártója.
- Visszatérési érték:
  - A függvény a visszatérő Json adatból egy Jarray objektumot készít a szűrt termékek adataival siker esetén, sikertelen lekérés esetén **null** értékkel tér vissza

### **loginToProg(string email, string password, string url):**

- Leíró szöveg: Ez a függvény a felhasználót bejelentkezteti az alkalmazásba a megadott e-mail-cím és jelszó páros alapján.
- Paraméterek:
  - email: A felhasználó email címe.
  - password: A felhasználó jelszava.
  - url: A bejelentkezési API végpont URL címe.
- Visszatérési érték:
  - A függvény true értéket ad vissza sikeres bejelentkezés esetén, egyébként false értéket.

### **logOut(string token, string url):**

- Leíró szöveg: Ez a függvény kijelenti a felhasználót az alkalmazásból a megadott felhasználói tokennel.
- Paraméterek:
  - token: A felhasználó bejelentkezési tokenje.
  - url: A kijelentkezési API végpont URL címe.
- Visszatérési érték:
  - A függvény nem térít vissza értéket (void).

## **Hibakezelés**

A DbConnect osztály minden függvénye try-catch blokkot használ a hibák kezelésére. Sikertelen lekérdezés vagy hitelesítési hiba esetén a függvények a MessageBox osztály segítségével megjelenítenek egy üzenetablakot amelyben részletes információt ad a hibáról. A hibaüzeneteket a backend szolgálja.

## DbOperations

### Osztály leírása

Az Isabike.DbOperations osztály az Isabike alkalmazás adatbázis műveleteit kezelő osztály. Az osztály felel a termékek és felhasználók felvételéhez, törléséhez és frissítéséhez a REST API segítségével.

### Függvények

#### **getKey(string res):**

- **Leíró szöveg:** Ez a függvény egy stringből próbálja kinyerni a token kulcsot. A függvény a megadott válasz (res) stringből eltávolítja a felesleges karaktereket (" ", "{", "}") és kettőspontot (':') használva szétválasztja a kulcs-érték párokat. A token kulcs pozíciója alapján (a második elem az érték) téríti vissza a megtalált token értéket.
- **Paraméterek:**
  - res: A token értékét tartalmazó válasz string.
- **Visszatérési érték:**
  - A függvény a megtalált token értéket adja vissza stringként, vagy üres stringet, ha a művelet sikertelen volt.

#### **addProduct(string jsondata, string url):**

- **Leíró szöveg:** Ez a függvény egy új terméket ad hozzá az adatbázishoz. A termék adatai JSON formátumban kerülnek átadásra a jsondata paraméterben. Az url paraméter pedig az API végpontot tartalmazza, ahova a HTTP POST kérés küldendő.
- **Paraméterek:**
  - jsondata: JSON formátumú string, amely a hozzáadandó termék adatait tartalmazza.
  - url: A termékek hozzáadásához használt API végpont URL címe.
- **Visszatérési érték:**
  - A függvény nem térít vissza értéket (void). A művelet eredménye a MessageBox osztály segítségével jelenik meg.

### **deleteProduct(string jsondata, string url):**

- Leíró szöveg: Ez a függvény töröl egy terméket az adatbázisból. A termék azonosítója a JSON formátumú jsondata paraméterben kerül átadásra. Az url paraméter pedig az API végpontot tartalmazza, ahova a HTTP DELETE kérés küldendő. A függvény a felhasználó megerősítését kéri a törlés előtt.
- Paraméterek:
  - jsondata: JSON formátumú string, amely a törölni kívánt termék azonosítóját tartalmazza.
  - url: A termékek törléséhez használt API végpont URL címe.
- Visszatérési érték:
  - A függvény nem térít vissza értéket (void). A művelet eredménye a MessageBox osztály segítségével jelenik meg.

### **updateProduct(string jsondata, string url):**

- Leíró szöveg: Ez a függvény frissít egy terméket az adatbázisban. A termék módosított adatai JSON formátumban kerülnek átadásra a jsondata paraméterben. Az url paraméter pedig az API végpontot tartalmazza, ahova a HTTP PATCH kérés küldendő.
- Paraméterek:
  - jsondata: JSON formátumú string, amely a módosítani kívánt termék adatait tartalmazza.
  - url: A termékek frissítéséhez használt API végpont URL címe.
- Visszatérési érték:
  - A függvény nem térít vissza értéket (void). A művelet eredménye a MessageBox osztály segítségével jelenik meg.

### **updateUser(string jsonData, string url):**

- Leíró szöveg: Ez a függvény frissít egy felhasználót az adatbázisban. A felhasználó módosított adatai JSON formátumban kerülnek átadásra a jsonData paraméterben. Az url paraméter pedig az API végpontot tartalmazza, ahova a HTTP PUT kérés küldendő.
- Paraméterek:
  - jsonData: JSON formátumú string, amely a módosítani kívánt felhasználó adatait tartalmazza.
  - url: A felhasználók frissítéséhez használt API végpont URL címe.
- Visszatérési érték:
  - A függvény nem térít vissza értéket (void). A művelet eredménye a MessageBox osztály segítségével jelenik meg.

## Hibakezelés

Az DbOperations osztály minden függvénye try-catch blokkot használ a hibák kezelésére. Sikertelen művelet esetén a függvények a MessageBox osztály segítségével jelenítik meg a hibaüzenetet.

## Termékek

### Osztály leírása

Az Isabike.Termékek osztály egy egyszerű adatstruktúra, amely a termékek adatainak tárolására szolgál. Az osztály három tulajdonsággal rendelkezik:

- `termek_nev`: A termék neve.
- `gyarto_neve`: A termék gyártójának neve.
- `tomeg_erteke`: A termék tömege.

### Tulajdonságok

- **`termek_nev`:**
  - Típus: string
  - Leírás: A termék neve.
  - Példa: "Kerékpár"
- **`gyarto_neve`:**
  - Típus: string
  - Leírás: A termék gyártójának neve.
  - Példa: "Trek"
- **`tomeg_erteke`:**
  - Típus: float
  - Leírás: A termék tömege.
  - Példa: 10.5

## Használat

Az Isabike.Termékek osztályt a termékek adatainak tárolására és továbbítására lehet használni. Például egy listában tárolhatjuk a termékeket, és megjeleníthetjük őket egy táblázatban.

## Függőségek

### Newtonsoft.Json

A Newtonsoft.Json csomag függőségének indoklása

A **Newtonsoft.Json** csomag használata az Isabike programban a következő okok miatt indokolt:

### **1. JSON adatcsere:**

A program API-hívásokat hajt végre, melyek JSON formátumban továbbítanak és fogadnak adatokat. A Newtonsoft.Json csomag hatékony eszköz a JSON objektumok szerializálására és deszerializálására, ami egyszerűsíti az API-hívások implementációját.

### **2. Objektum-adatmodell leképezése:**

A program adatmodellje JSON struktúrákon alapul. A Newtonsoft.Json csomag lehetővé teszi a JSON objektumok egyszerű leképezését a program objektummodelljére, minimalizálva a manuális adatkonverzió szükségességét.

### **3. Kompatibilitás:**

A Newtonsoft.Json széles körben elterjedt és számos más .NET-es programmal kompatibilis. Ez megkönnyíti a program integrálását más rendszerekkel, és növeli a hordozhatóságát.

### **4. Teljesítmény:**

A Newtonsoft.Json gyors és hatékony, minimalizálva a JSON adatcserével kapcsolatos teljesítménybeli overhead-et.

### **5. Egyszerű használat:**

A Newtonsoft.Json API-ja könnyen tanulható és használható, egyszerűsítve a fejlesztési folyamatot.

### **Alternatívák:**

Léteznek más JSON-kezelő csomagok is .NET-hez, mint a **System.Text.Json**. Azonban a Newtonsoft.Json továbbra is az egyik legnépszerűbb és legmegbízhatóbb megoldás, és a fenti előnyök miatt ideális választás az Isabike programhoz.

### **Összefoglalva:**

A Newtonsoft.Json csomag használata az Isabike programban számos előnnyel jár, beleértve a hatékony JSON adatcserét, az egyszerű adatmodell-leképezést, a kompatibilitást, a teljesítményt és az egyszerű használatot. A csomag használata megkönnyíti a program fejlesztését és karbantartását, és javítja a program integrálhatóságát és hordozhatóságát.

# Formok

## MainForm

### Osztály leírása

A MainForm osztály az Isabike alkalmazás főablakát reprezentáló osztály. Az osztály felel a felhasználói felület kezeléséért, a termékek és vélemények megjelenítéséért, valamint a szűrésért és a termékekkel kapcsolatos műveletekért (törlés, frissítés).

### Vezérlők

- **manufacturerBox** (ComboBox): A gyártók legördülő listája. A lista a DbConnect.getData függvény segítségével lekéri a gyártókat a `http://127.0.0.1:8000/api/gyartok` URL-ről. A ValueMember property beállítása a gyártó azonosító (gyarto\_id) mezőre történik, a DisplayMember property pedig a gyártó nevének (gyarto\_neve) megjelenítését határozza meg.
- **categoryBox** (ComboBox): A kategóriák legördülő listája. A lista a DbConnect.getData függvény segítségével lekéri a kategóriákat a `http://127.0.0.1:8000/api/kategoria` URL-ről. A ValueMember property beállítás a kategória azonosító (kategoria\_id) mezőre történik, a DisplayMember property pedig a kategória nevének (kategoria\_neve) megjelenítését határozza meg.
- **viewGrid** (DataGridView): A termékek vagy vélemények megjelenítésére szolgáló táblázat. A táblázat oszlopainak fejlécei a lekérdezett adatok mezőneveinek megfelelően alakulnak. A függvény a lekérdezés eredményét a táblázat adatforrásaként állítja be (viewGrid.DataSource).
- **termeknevTextbox** (TextBox): A szűréshez használt terméknév beviteli mezője. A felhasználó által beírt terméknév alapján történik a szűrés.
- **suly\_textbox** (TextBox): A szűréshez használt súly beviteli mezője. A felhasználó által beírt súly alapján történik a szűrés.
- **ReviewsBtn** (Button): A termékek és a vélemények közötti váltásra szolgáló gomb. A gridState változó alapján állapítható meg az aktuális nézet (true: vélemény, false: terméke). A gomb megnyomása során a connectToDB függvény meghívódik a megfelelő adattípus betöltésére.
- **refreshBtn** (Button): Az adatok frissítése gomb. A gomb megnyomása az aktuális nézet (termékek vagy vélemények) frissítését indítja el. A connectToDB függvény újrahívásával lekéri a legfrissebb adatokat a szerverről.
- **SalesBtn** (Button): Az eladások gomb ( jelenleg nincs implementálva). A gomb feladata egy jövőbeli eladások modul megjelenítése lenne.
- **filterBtn** (Button): A szűrés gomb. A gomb megnyomása után a szűrőfeltételek (terméknév, súly, gyártó) alapján történik az adatok lekérdezése. A szűrést a DbConnect.getFilteredData függvény valósítja meg.

- **updateButtonColumn** (DataGridViewButtonColumn): A táblázat egy oszlopa, amely "Módosítás" gombokat tartalmaz minden sorban. A gomb megnyomása a gomb sorában lévő cellákban átírt adatokkal módosítja a sor adatbázisban eltárolt adatait. A műveletet a viewGrid\_CellClick eseménykezelő függvény valósítja meg.
- **deleteButtonColumn** (DataGridViewButtonColumn): A táblázat egy oszlopa, amely "Törlés" gombokat tartalmaz minden sorban. A gomb megnyomása törli a kiválasztott terméket. A felhasználó megerősítése után a DbOperations.deleteProduct függvény segítségével történik a törlés a megadott API végpontra (<http://127.0.0.1:8000/api/deletetermek>) küldött HTTP DELETE kéréssel.

## Eseménykezelők

- **Main\_Load:** Az űrlap betöltésekor meghívódik. A connectToDB függvényt hívja meg a termékek betöltésére. Ellenőrzi, hogy a "Törlés" és "Módosítás" oszlopok léteznek-e a táblázatban, ha nem, akkor beszúrja őket.
- **ReviewsBtn\_Click:** A "Vélemények" gomb megnyomásakor meghívódik. A connectToDB függvényt hívja meg a vélemények betöltésére, és megváltoztatja a gomb háttérszínét.
- **refreshBtn\_Click:** A "Refresh" gomb megnyomásakor meghívódik. A connectToDB függvényt hívja meg az aktuális nézet (termékek vagy vélemények) frissítésére.
- **filterBtn\_Click:** A "Filter" gomb megnyomásakor meghívódik. A DbConnect.getFilteredData függvényt hívja meg a szűrőfeltételek alapján szűrt termékek lekérdezésére.
- **viewGrid\_CellClick:** A táblázat celláinak kattintásakor meghívódik. A kattintás helyétől függően (Törlés/Módosítás oszlop) a DbOperations.deleteProduct illetve a DbOperations.updateProduct függvényeket hívja meg a termék törlésére vagy módosítására.
- **UserBtn\_Click:** A "User Operations" gomb megnyomásakor meghívódik. Megjeleníti a UserForm űrlapot.

## Adatbázis műveletek

- **connectToDB:** Csatlakozik az adatbázishoz és lekéri a termékeket vagy a véleményeket a DbConnect.getData függvény segítségével. Beállítja a táblázat adatforrását.
- **deleteProduct:** A DbOperations.deleteProduct függvényt hívja meg a kiválasztott termék törlésére.
- **updateProduct:** A DbOperations.updateProduct függvényt hívja meg a kiválasztott termék módosítására.



## Login

### Osztály leírása

Az `Isabike.Login` osztály az Isabike alkalmazás bejelentkezési űrlapját reprezentáló osztály. A felhasználó bejelentkezési adatainak megadása és a bejelentkezés kezelése történik az osztályon keresztül. Az osztály statikus `token` attribútumot és metódusokat is tartalmaz a bejelentkezett felhasználó tokenjének kezelésére.

### Tulajdonságok

- **token (static string):**
  - Leírás: A bejelentkezett felhasználó tokenje. A token egy titkos érték, amely a felhasználó azonosítására szolgál a szerver oldali műveleteknél.
  - Hozzáférés: A `getToken` metódus segítségével olvasható, a `setToken` metódus segítségével állítható be.

### Metódusok

- **Login():**
  - Leírás: Az osztály konstruktora. Inicializálja az űrlap elemeit.
- **Login\_Load(object sender, EventArgs e):**
  - Leírás: Az űrlap betöltésekor lefutó eseménykezelő. Az űrlapot az előtérbe helyezi.
- **loginBtn\_Click(object sender, EventArgs e):**
  - Leírás: A "Login" gomb megnyomásakor lefutó eseménykezelő. A megadott e-mail cím és jelszó alapján próbálja megvalósítani a bejelentkezést. Sikeres bejelentkezés esetén a főablak (`MainForm`) példányosodik és megjelenik, a bejelentkezési űrlap pedig eltűnik.
- **closeBtn\_Click(object sender, EventArgs e):**
  - Leírás: A "Cancel" gomb megnyomásakor lefutó eseménykezelő. Beállítja az űrlap bezárását.
- **getToken(): (static)**
  - Leírás: Visszatér a beállított token értékkel.
- **setToken(string userToken): (static)**
  - Leírás: Beállítja a bejelentkezett felhasználó token értékét.

# ProductOperationsForm

## Osztály áttekintése

Az `Isabike.ProductOperationsForm` osztály az Isabike alkalmazás egy termék hozzáadására szolgáló űrlapot reprezentáló osztály. Az űrlap segítségével új termékek adatai hozhatók létre és továbbíthatók az adatbázisba.

## Metódusok

- **populateTomeg(string uri):**
  - Leírás: Beállítja a tömegegység legördülő menü elemeit az `uri` paraméterben megadott API végpontból lekérdezett adatok alapján. A lekérdezés a `DbConnect.GetData` metódus segítségével történik.
  - Paraméterek:
    - `uri`: Az API végpont URL címe, amely a tömegegységekre vonatkozó adatokat tartalmazza.
- **populateGyarto(string uri):**
  - Leírás: Beállítja a gyártó legördülő menü elemeit az `uri` paraméterben megadott API végpontból lekérdezett adatok alapján. A lekérdezés a `DbConnect.GetData` metódus segítségével történik.
  - Paraméterek:
    - `uri`: Az API végpont URL címe, amely a gyártókra vonatkozó adatokat tartalmazza.
- **populateKategoria(string uri):**
  - Leírás: Beállítja a kategória legördülő menü elemeit az `uri` paraméterben megadott API végpontból lekérdezett adatok alapján. A lekérdezés a `DbConnect.GetData` metódus segítségével történik.
  - Paraméterek:
    - `uri`: Az API végpont URL címe, amely a kategóriákra vonatkozó adatokat tartalmazza.
- **ProductOperationsForm():**
  - Leírás: Az osztály konstruktora. Inicializálja az űrlap elemeit, beleértve a termék nevét, gyártóját, kategóriáját, mennyiségét, tömegét, tömegegységét, színét, leírását, árát és elérhetőséget jelző beviteli mezőket, valamint a legördülő menüket és a "Mentés" és "Bezárás" gombokat.
- **closeBtn\_Click(object sender, EventArgs e):**
  - Leírás: A "Bezárás" gomb megnyomásakor lefutó eseménykezelő. Beállítja az űrlap bezárását.

- **okBtn\_Click(object sender, EventArgs e):**
  - Leírás: A "Mentés" gomb megnyomásakor lefutó eseménykezelő. A következő lépéseket hajtja végre:
    - Létrehoz egy JSON objektumot a beírt termékadatokból, beleértve a bejelentkezett felhasználó tokenjét (`Login.GetToken()`), a kategóriát, gyártót, mennyiséget, tömeget, tömegegységet, színét, leírását, árát és elérhetőség értékét.
    - A JSON objektumot string formátumba alakítja (`JsonConvert.SerializeObject`).
    - A `DbOperations.AddProduct` metódust meghívja a termék hozzáadására a megadott API végponton (`http://127.0.0.1:8000/api/addtermek`) a JSON stringgel.
    - Sikeres művelet esetén nem történik külön visszajelzés. Sikertelen művelet esetén a kivételt kiírja egy Message box segítségével.
- **ProductOperationsForm\_Load(object sender, EventArgs e):**
  - Leírás: Az űrlap betöltésekor lefutó eseménykezelő. Inicializálja az űrlap legördülő menüit a gyártók, kategóriák és tömegegységek adataival a megadott API végpontokból.

## UserForm

### Osztály áttekintése

Az `Isabike.UserForm` osztály az Isabike alkalmazás felhasználó kezelő űrlapát reprezentáló osztály. Az űrlap segítségével lehet lekérdezni a felhasználók adatait, és adminisztrátori jogosultságot adni nekik.

### Attribútumok

- **makeadminButtonColumn (DataGridViewButtonColumn):**
  - Leírás: Egy dinamikusan létrehozott DataGridView gomb oszlop, amely a "Felhasználók" táblázat első oszlopában jelenik meg, és a "Makeadmin" feliratú gombot tartalmazza.

### Metódusok

#### **UserForm():**

- Leírás: Az osztály konstruktora. Inicializálja az űrlap elemeit, beleértve a "Felhasználók" táblázatot és a "Frissítés" gombot.

#### **UserForm\_Load(object sender, EventArgs e):**

- Leírás: Az űrlap betöltésekor lefutó eseménykezelő. A következőket hajtja végre:
  - Lekéri a felhasználók adatait a megadott API végpontról (`http://127.0.0.1:8000/api/getfelhasznalok`) a `GetRESTData` metódus segítségével.
  - Létrehozza a "Makeadmin" gombot a táblázat első oszlopában a `adminButtonCreate` metódus segítségével.

#### **GetRESTData(string uri):**

- Leírás: Lekéri a felhasználók adatait az `uri` paraméterben megadott API végpontból a `DbConnect.getUserData` metódus segítségével. A lekérdezett adatokat a "Felhasználók" táblázat adatforrásává állítja be.

#### **adminButtonCreate():**

- Leírás: Ellenőrzi, hogy a "Makeadmin" gomb oszlop létezik-e a táblázatban. Ha nem létezik, létrehozza és hozzáadja az első oszlopként a táblázathoz.

#### **userDataGridView\_CellClick(object sender, DataGridViewCellEventArgs e):**

- Leírás: A "Felhasználók" táblázat celláira kattintáskor lefutó eseménykezelő.
- Funkció: Ha a felhasználó a "Makeadmin" gombra kattint (az első oszlopban lévő gombra), akkor a következőket hajtja végre:
  - Létrehoz egy JSON objektumot a kiválasztott felhasználó adataival, kiegészítve az "adminisztrátori" jogosultsággal (3).
  - A JSON objektumot string formátumba alakítja.
  - A `DbOperations.updateUser` metódust meghívja a felhasználó adatainak frissítésére a megadott API végponton (`http://127.0.0.1:8000/api/updatefelhasznalo`).

**refreshBtn\_Click(object sender, EventArgs e):**

- Leírás: A "Frissítés" gombra kattintáskor lefutó eseménykezelő. Frissíti a "Felhasználók" táblázatot a legfrissebb adatokkal a `GetRESTData` metódus újrahívásával, és szükség esetén újra létrehozza a "Makeadmin" gombot a `adminButtonCreate` metódus segítségével.

# Isabike Program Felhasználói Dokumentáció

## Bevezetés

Ez a dokumentáció az Isabike program felhasználói számára készült. A dokumentáció célja, hogy bemutassa a program különböző űrlapjainak és osztályainak használatát. A dokumentáció a következőket tartalmazza:

- **Áttekintés:** Rövid leírás a program céljáról és funkcióiról.
- **Információ a használt eszközökről:** A programban használt fejlesztői eszközök és technológiák felsorolása (pl. .NET Windows Forms, Newtonsoft.Json).
- **Telepítési útmutatás:** Útmutató az Isabike alkalmazás telepítéséhez
- **Űrlapok dokumentációja:** Részletes leírás minden egyes űrlapról, beleértve:
  - Az űrlap célja és funkciója
  - Az űrlap elemei (gombok, szövegmezők, táblázatok stb.) és azok használata
  - Az űrlapon elérhető műveletek végrehajtásának lépései
- **Osztályok dokumentációja:** Rövid ismertetés minden egyes osztályról, beleértve:
  - Az osztály célja és funkciója
  - Az osztály metódusainak listája, paraméterei és a visszatérési érték leírása
  - Az osztály attribútumainak (ha vannak) leírása

## Használt eszközök:

- .NET Windows Forms: Az asztali alkalmazás grafikus felhasználói felületének létrehozásához használt keretrendszer.
- Newtonsoft.Json: Egy népszerű könyvtár JSON adatcseréhez .NET alkalmazásokban.

## Fontos megjegyzés:

A dokumentáció nem tartalmazza a program belső működésének részletes ismertetését, hanem a felhasználói szempontból fontos információkra koncentrál.

## Áttekintés

Az Isabike egy kerékpár áruház számára fejlesztett program, amely a következő funkciókat biztosítja:

- Felhasználó bejelentkezés
- Termékek kezelése (hozzáadás, módosítás, törlés)
- Termékek listázása
- Vélemények megtekintése
- Felhasználók kezelése (adminisztrátor jogosultság esetén)

## Telepítési útmutató az Isabike programhoz

### Előfeltételek:

- A program futtatásához .NET Framework 4.7.2 vagy újabb verzió szükséges.
- A program telepítéséhez adminisztrátori jogosultsággal kell rendelkeznie.

### Telepítési lépések:

1. Töltse le a telepítőfájlt a link a telepítőfájllhoz(Névszerint: Isabikesetup.msi).

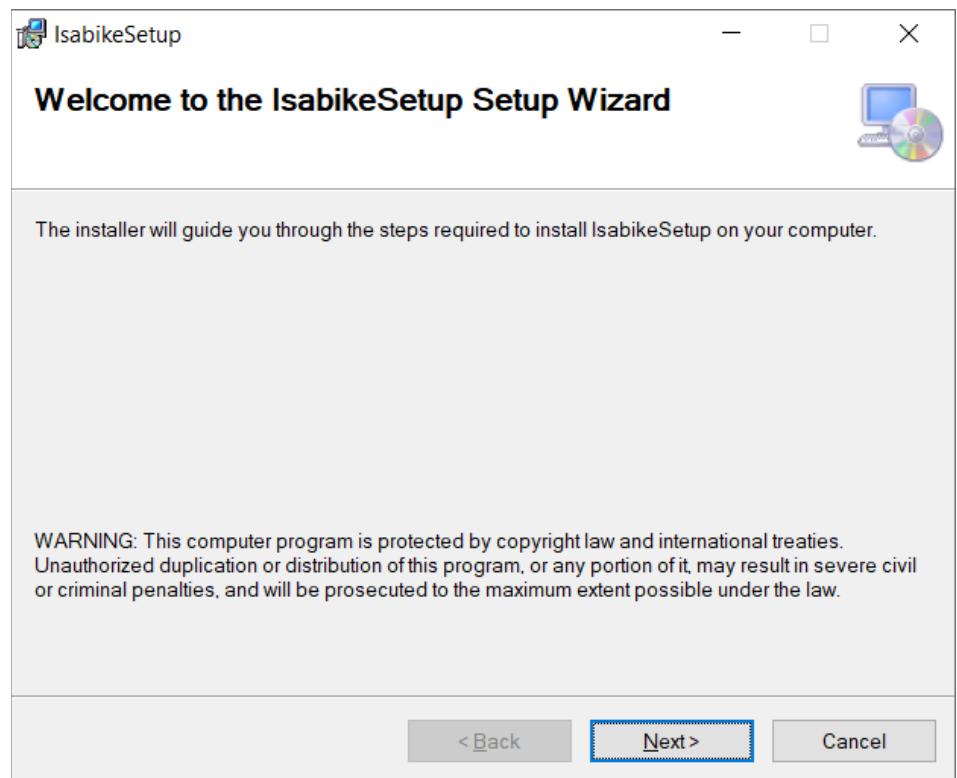
Kép róla:



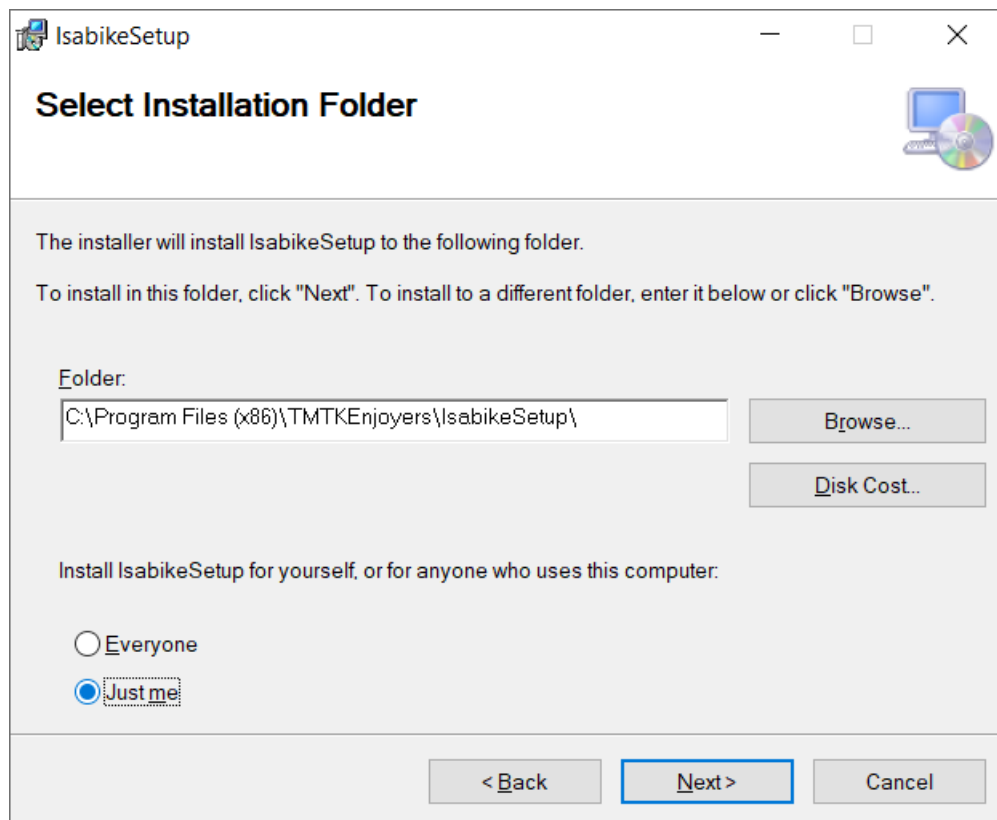
2. Futtassa a telepítőfájlt.
3. A telepítő varázsló lépésein haladva kövesse a képernyőn megjelenő utasításokat.

Képek róla:

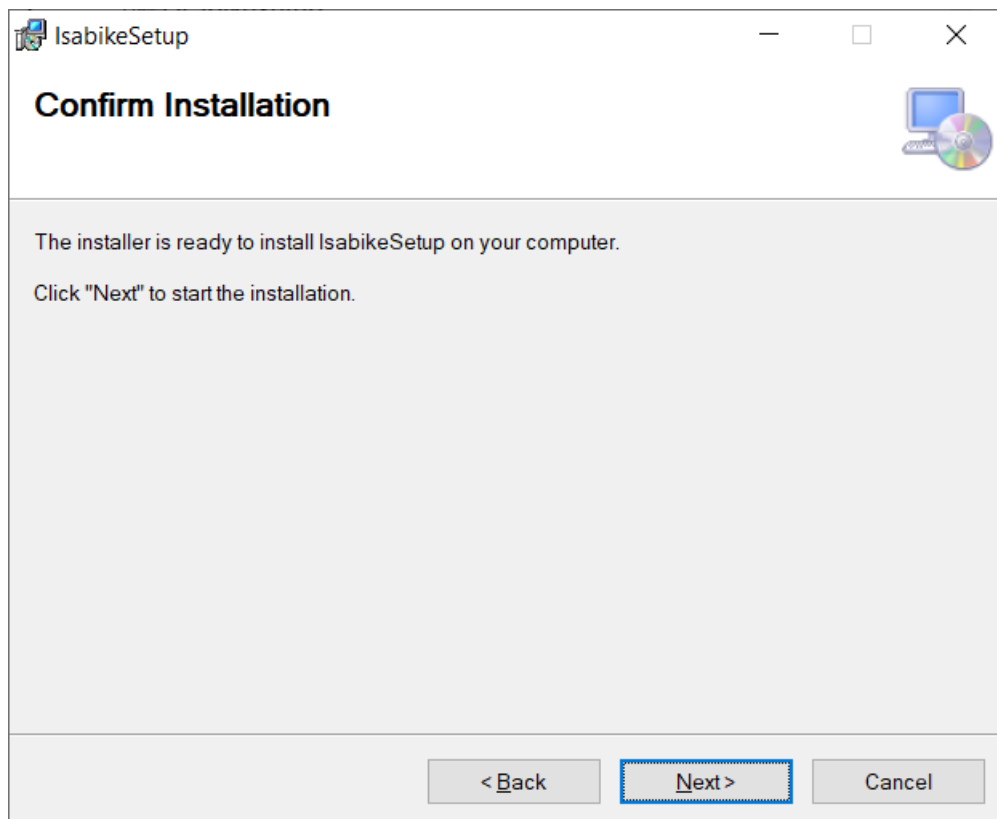
1.



2.



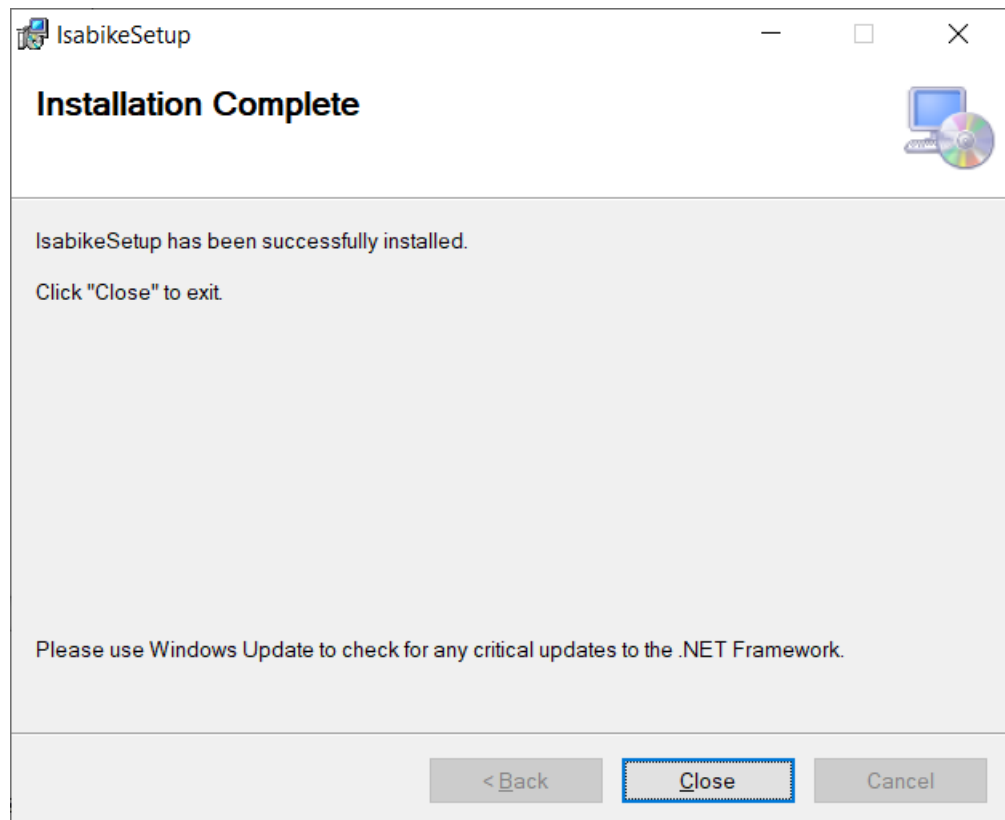
3.



4. A „Next” gomb megnyomása után várja meg míg települ a program



4.



5. A close gomb lenyomása után az asztalon megtalálja a program indító icon-át

**A program használata:**

- A program elindításához kattintson az Isabike ikonra az asztalon.
- A program bejelentkezési űrlappal nyílik meg.
- Adja meg az e-mail címét és jelszavát a bejelentkezéshez.
- A bejelentkezés után a program fő űrlapja jelenik meg.

## Űrlapok dokumentációja

### Login űrlap

- **Cél:** A Login űrlap lehetővé teszi a felhasználók számára a programba való bejelentkezést.
- **Elemek:**
  - Email Textfield: A felhasználó e-mail címének megadása.
  - Password Textfield: A felhasználó jelszavának megadása (jelszómező).
  - Login gomb: A bejelentkezési adatok elküldése a programnak.
  - Close gomb: Az űrlap bezárása.
- **Műveletek:**
  - Írd be az e-mail címed az Email Textfield mezőbe.
  - Írd be a jelszavad a Password Textfield mezőbe.
  - Kattints a Login gombra a bejelentkezéshez.
- **Hibaüzenetek:**
  - A program megjelenít egy hibaüzenetet, ha a bejelentkezési adatok hibásak.
  - A program megjelenít egy hibaüzenetet, ha az adatbázis szerver nem elérhető

## MainForm űrlap

- **Cél:** A MainForm űrlap a program fő kezelőfelülete. Itt lehet megtekinteni a termékeket és a véleményeket, valamint hozzáférni egyéb funkciókhoz.
- **Elemek:**
  - Menüsor:
    - UserBtn gomb: A felhasználó kezelő űrlap megnyitása.
    - Termékek szűréséhez:
      - terméknevTextbox: Terméknév szűrő mező.
      - suly\_textbox: Súly szűrő mező.
      - manufacturerBox: Gyártó szűrő legördülő lista.
      - filterBtn gomb: A szűrés végrehajtása.
  - Termékek táblázata (viewGrid): A termékek adatait megjelenítő táblázat. A táblázat oszlopai a következők lehetnek:
    - Termék azonosító
    - Termék kategória
    - Termék neve
    - Gyártó neve
    - Árkategória
    - Raktárkészlet
    - Tömeg típusa
    - Tömeg értéke
    - Szín
    - Leírás
    - Egységár
    - Elérhető
  - **Művelet gombok:**
    - Delete gomb: A kiválasztott termék törlése (adminisztrátori jogosultság esetén).
    - Update gomb: A kiválasztott termék szerkesztése (adminisztrátori jogosultság esetén).
    - ReviewsBtn gomb: A termékvélemények táblázatának megjelenítése/elrejtése.
    - RefreshBtn gomb: A táblázat frissítése a legfrissebb adatokkal.

- **Műveletek:**

- A termékek szűréséhez:
  - Írd be a keresett termék nevét a terméknevTextbox mezőbe.
  - (Opcionális) Írd be a keresett termék súlyát a suly\_textbox mezőbe.
  - Válaszd ki a keresett termék gyártóját a manufacturerBox legördülő listából.
  - Kattints a filterBtn gombra a szűrés végrehajtásához.
- A termékvélemények megtekintéséhez:
  - Kattints a ReviewsBtn gombra és a táblázatban megjelennek a termékhez kapcsolódó vélemények.
- A táblázat frissítéséhez:
  - Kattints a RefreshBtn gombra.

- **Hibaüzenetek:**

- A program hibaüzenetet jelenít meg, ha a szűrés feltételek nem teljesülnek.
- A program megjelenít egy hibaüzenetet, ha az adatbázis szerver nem elérhető

## ProductOperationsForm űrlap

- **Cél:** A ProductOperationsForm űrlap lehetővé teszi új termékek hozzáadását a programhoz.
- **Elemek:**
  - Termék kategória (productCategoryBox): Legördülő lista a termék kategóriák kiválasztásához.
  - Termék név (productNameText): Szövegmező a termék nevének megadásához.
  - Gyártó (productManufacturerBox): Legördülő lista a termék gyártójának kiválasztásához.
  - Raktárkészlet (productCountText): Szövegmező a termék raktárkészletének megadásához.
  - Tömeg típusa (productWeightclassBox): Legördülő lista a termék tömeg típusának kiválasztásához.
  - Tömeg értéke (productWeightText): Szövegmező a termék tömeg értékének megadásához.
  - Szín (productColorText): Szövegmező a termék színének megadásához.
  - Leírás (productDescText): Szövegmező a termék leírásának megadásához.
  - Ár (productPriceText): Szövegmező a termék árának megadásához.
  - Elérhető (productAvelableBox): Jelölőnégyzet a termék elérhetőségének jelzésére.
  - OK gomb: Az új termék adatainak elküldése a programnak.
  - Close gomb: Az űrlap bezárása.
- **Műveletek:**
  - Válaszd ki a termék kategóriáját a productCategoryBox legördülő listából.
  - Írd be a termék nevét a productNameText mezőbe.
  - Válaszd ki a termék gyártóját a productManufacturerBox legördülő listából.
  - Írd be a termék raktárkészletét a productCountText mezőbe.
  - Válaszd ki a termék tömeg típusát a productWeightclassBox legördülő listából.
  - Írd be a termék tömeg értékét a productWeightText mezőbe.
  - Írd be a termék színét a productColorText mezőbe.
  - Írd be a termék leírását a productDescText mezőbe.
  - Írd be a termék árát a productPriceText mezőbe.
  - Jelöld be a productAvelableBox jelölőnégyzetet, ha a termék elérhető.
  - Kattints az OK gombra az új termék hozzá
  - A program megjelenít egy üzenetet az új termék sikeres hozzáadásáról.

- **Hibaüzenetek:**
  - A program hibaüzenetet jelenít meg, ha az űrlapon nem adtunk meg minden szükséges adatot.
  - A program megjelenít egy hibaüzenetet, ha az adatbázis szerver nem elérhető

## UserForm űrlap

- **Cél:** A UserForm űrlap lehetővé teszi a program felhasználóinak listázását és adminisztrátorrá való kinevezését (adminisztrátori jogosultság esetén).
- **Elemek:**
  - Felhasználók táblázata (userDataGridView): A program felhasználóinak adatait megjelenítő táblázat. A táblázat oszlopai a következők lehetnek:
    - Makeadmin gomb: A kiválasztott felhasználó adminisztrátorrá kinevezése (adminisztrátori jogosultság esetén).
    - Felhasználó azonosító
    - Felhasználónév
    - Vezetéknév
    - Keresztnév
    - Telefonszám
    - E-mail cím
    - Számlázási cím
    - Szállítási cím
    - Jogosultsági szint
  - RefreshBtn gomb: A táblázat frissítése a legfrissebb adatokkal.
- **Műveletek:**
  - A felhasználó adminisztrátorrá való kinevezéséhez:
    - Válaszd ki a kívánt felhasználót a userDataGridView táblázatban.
    - Kattints a Makeadmin gombra.
  - A táblázat frissítéséhez:
    - Kattints a RefreshBtn gombra.
- **Hibaüzenetek:**
  - A program hibaüzenetet jelenít meg, ha a művelet nem hajtható végre.
  - A program megjelenít egy hibaüzenetet, ha az adatbázis szerver nem elérhető

## Osztályok dokumentációja

- **DbConnect:**

- Cél: Adatbázishoz való kapcsolódás és lekérdezések végrehajtása.
- Metódusok:
  - `getData(uri)`: Lekérdezi a JSON formátumú adatokat a megadott URL-ről.
  - `getFilteredData(uri, terméknev, súly, gyártó)`: Lekérdezi a szűrt termékek adatait a megadott URL-ről.
  - `getUserData(uri)`: Lekérdezi a felhasználók adatait a megadott URL-ről.
  - `loginToProg(email, password, uri)`: Bejelentkezés a programba a megadott e-mail címmel és jelszóval.
  - `logout(token, uri)`: Kijelentkezés a programból a megadott tokennel.
  - `addProduct(json, uri)`: Új termék hozzáadása a programban a JSON formátumú adatokkal.
  - `updateProduct(json, uri)`: Termék frissítése a programban a JSON formátumú adatokkal.
  - `deleteProduct(json, uri)`: Termék törlése a programból a JSON formátumú adatokkal.
  - `updateUser(json, uri)`: Felhasználó frissítése a programban a JSON formátumú adatokkal.

- **DbOperations:**

- Cél: Adatbázis műveletek végrehajtása.
- Metódusok:
  - `addProduct(json, uri)`: Új termék hozzáadása a programban a JSON formátumú adatokkal.
  - `updateProduct(json, uri)`: Termék frissítése a programban a JSON formátumú adatokkal.
  - `deleteProduct(json, uri)`: Termék törlése a programból a JSON formátumú adatokkal.
  - `updateUser(json, uri)`: Felhasználó frissítése a programban a JSON formátumú adatokkal.

### További információk:

- A program forráskódja elérhető a [https://github.com/PlesovszkiIstvan/TMTK\\_enjoyer\\_vizsgaremek](https://github.com/PlesovszkiIstvan/TMTK_enjoyer_vizsgaremek) linken.

**Köszönjük, hogy az Isabike programot használja!**