

Szent István Katolikus  
Technikum és Gimnázium

asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd

**Götz patrik, lukács Luca, Farkas Máté**

asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd

**zálogház**

**Szoftverfejlesztés és -tesztelés vizsgaremek**

**5 0613 12 03**

Sátoraljaújhely, 2022

asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asd asdasd asd asd asd asd asd asd asd

Tartalomjegyzék

[BEVEZETÉS 4](#_Toc132556786)

[Rólunk 4](#_Toc132556787)

[Miért választottuk ezt a témát? 4](#_Toc132556788)

[Munkafelosztás 5](#_Toc132556789)

[A programról 6](#_Toc132556790)

[Fejlesztői Dokumentáció 6](#_Toc132556791)

[Főoldal 6](#_Toc132556792)

[Áruház 15](#_Toc132556793)

[Visual Studio Code Leírása 24](#_Toc132556794)

[IntelliSense a Visual Studio Code-ban: 24](#_Toc132556795)

[A kiegészítések típusai 24](#_Toc132556796)

[IntelliSense-funkciók a visual studio code-ban: 24](#_Toc132556797)

[Nyelvkiegészítés: 24](#_Toc132556798)

[Paraméteradatok: 25](#_Toc132556799)

[Gyors Információ: 25](#_Toc132556800)

[Az Stream Analytics Code-ban végzett Visual Studio tesztelésének áttekintése ASA-eszközökkel: 25](#_Toc132556801)

[A helyi futtatás bemeneti szempontjai 25](#_Toc132556802)

[A helyi futtatás kimeneti szempontjai 26](#_Toc132556803)

[Helyi futtatás módjai 26](#_Toc132556804)

[Stream Analytics-lekérdezések helyi tesztelése mintaadatokkal a Visual Studio Code használatával 26](#_Toc132556805)

[Minta adatok előkészítése: 27](#_Toc132556806)

[Lekérdezések helyi futtatása: 30](#_Toc132556807)

[Felhasználói Dokumentáció: 31](#_Toc132556808)

[Rendszerkövetelmények: 32](#_Toc132556809)

[Böngészők 33](#_Toc132556810)

[Weboldalról: 39](#_Toc132556811)

[Források 42](#_Toc132556812)

# BEVEZETÉS

## Rólunk

Farkas Máté: 21 Éves vagyok immáron a második szakmámat végzem el a Szent István Katolikus Technikum és Gimnáziumban. Első szakmámat az érettségire alapuló iskola, rendszerben szereztem meg, a szakmámnak a megnevezése: Rendszerüzemeltető és Karbantartó (OKJ-S képesítés 4+1 év). Azért jelentkeztem a szoftveres részére is, mert szeretem a kihívásokat, és megtudtam, hogy ez technikumra alapuló oktatási rendszer lesz, ami kivált egy felsőfokú érettségi bizonyítványt. Mivel nekem fontos lesz a továbbtanulásomhoz egy felsőfokú érettségi (Etikus Hacker), így gondoltam elkezdem a Szoftverfejlesztő és Tesztelő technikusi osztályt is.

Götz Patrik: 22 éves vagyok, és a második szakmámat végzem. Először a Vay Miklós Református Szakképző Iskola és Diákotthon tanulójaként Épületgépész szakmát szereztem meg. Miután egy kis ideig dolgoztam a szakmában, rájöttem, hogy nem nekem való az épületgépészet, ezért eljöttem a Szent István Katolikus Technikum és Gimnázium Szoftverfejlesztő és - Tesztelő szakra. A tervem az, hogy a szakma elvégzése után munkába álljak ezen a területen.

Lukács Luca: 20 éves vagyok, a második szakmámat végzem itt a Szent István Katolikus Technikum és Gimnáziumban. Első szakmám Közgazdasági Ügyintéző volt, viszont abban nem éreztem magam kényelmesen, ezért váltottam a Szoftverfejlesztő és - Tesztelő irány felé, ami már régóta érdekel engem. A szakma megszerzése után webfejlesztőként szeretnék majd helyt állni, azon belül is a Frontend részén.

## Miért választottuk ezt a témát?

A projektünk témája egy Zálogház nevű ékszerüzlet weboldala. Azért választottuk ezt a témát, mert érdekelt minket a különböző ékszerekkel való kereskedés és úgy gondoltuk, hogy egy érdekes weboldalt tudnánk összehozni ebből a témából. A fejlesztés során igyekeztünk a tudásunk szerinti legkorszerűbb technikákat, eszközöket, alkalmazásokat alkalmazni és a lehető legnagyobb odafigyeléssel eljárni. Munkánk során a HTML, CSS, JavaScript programokat használtuk a Frontend résznél, a MySQL adatbázis kezelőt pedig a Backend fejlesztéshez. Úgy gondoljuk, hogy egy remek weboldalt sikerült létrehoznunk.

## Munkafelosztás

A munkát három felé osztottuk, viszont igyekeztünk minden részt együtt csinálni. Mindenki azt a részét kapta, amelyikhez a legjobb tudással rendelkezik. Az oldal kinézetét Lukács Luca készítette. A projekt kezdetekor eleinte csak Bootstrap segítségét vettem igénybe, de nem rendelkezett elegendő formalitással, és vizuális hibákkal találkoztam. A zökkenőmentes feladat megoldása érdekében a teljes Bootstrap állomány el lett távolítva, és minden egyes elem saját kezűleg lett formázva. A projekt társam a weboldal felépítés szerkezetét javítottam és egészítettem ki. A stíluslaphoz felhasznált képek egy számtalan részét én magam dizájnoltam, és modelleztem.

A HTML szerkezetet és a JavaScript részt Götz Patrik írta meg. A Főoldalon elhelyeztem egy eleinte Bootstrap segítségével elkészült slideshow szerkezetet, viszont az elhelyezés után problémákba ütköztünk, ezért eltávolítottuk végül. Később a helyére egy teljesen saját kezűleg megírt slideshow került a helyére. A weboldal Kosár részének HTML szerkezetét közösen Lukács Luca segítségével készítettem el. A JavaScript kód megírása sok bonyodalommal járt, de végül sikerült megalkotni.

Az oldalnak későbbiekre terveztünk egy regisztráció és bejelentkezés részt, amit Farkas Máté alakított ki. Létrehoztam a server.php nevű állományt, ahol a PhP MySql alkalmazáson, localhoston tudjuk futtatni a weboldalt. Regisztrációnál egy felhasználónevet, e-mail címet és jelszót fog kérni, az utóbbit még egyszer meg kell adni a megerősítéshez. Minden egyes adatot kötelező megadni és egyetlen egy mezőt sem hagyhatunk üresen. Ha a regisztráció gombra kattintunk és sikeres akkor eltárolódik az adatbázisban a felhasználónév email cím és jelszó.

# A programról

## Fejlesztői Dokumentáció

### Főoldal

Főoldalon található egy navbar. Reszponzívra készítettük, ami miatt minden eszközön, minden képernyőn rendesen fog látszódni a felület. Egy „div” tagen belül létrehoztunk „a” tageket, amik arra szolgálnak, hogy egyik oldalról eljuthassunk a másikra. A menüsor legelején egy „img” tag található, amit egy logónak terveztünk. A CSS fájlban megírt reszponzivitás miatt, ha telefonról akarjuk nézni az oldalt, akkor is tökéletesen meg fog jelenni.

HTML kód:

<!-- Navigációs sáv -->

    <div class="topnav" id="myTopnav">

      <a><img id="navbarIkon" src="gyemant.png" alt="Zálogház" title="Zálogház">Zálogház</a>

      <a href="fooldal.html">Főoldal</a>

      <a href="aruhaz.html">Áruház</a>

      <a href="javascript:void(0);" class="icon" onclick="myFunction()">

        <i class="fa fa-bars"></i>

      </a>

    </div>

    <script>

    function myFunction() {

      var x = document.getElementById("myTopnav");

      if (x.className === "topnav") {

        x.className += " responsive";

      } else {

        x.className = "topnav";

      }

    }

    </script>

CSS kód:

body {

    background-color: rgb(223, 232, 241);

    margin: 0;

    font-family: Arial, Helvetica, sans-serif;

    font-weight: bold;

}

#navbarIkon {

  width: 30px;

  height: 30px;

  vertical-align: text-top;

}

.topnav {

  overflow: hidden;

  background-color: rgb(223, 232, 241);

  padding-bottom: 80px;

}

.topnav a {

  float: left;

  display: block;

  color: black;

  text-align: center;

  padding: 14px 16px;

  text-decoration: none;

  font-size: 20px;

}

.topnav a:hover {

  background-color: rgb(192, 210, 228);

  color: black;

}

.topnav .icon {

  display: none;

}

@media screen and (max-width: 600px) {

  .topnav a:not(:first-child) {display: none;}

  .topnav a.icon {

    float: right;

    display: block;

  }

}

@media screen and (max-width: 600px) {

  .topnav.responsive {position: relative;}

  .topnav.responsive .icon {

    position: absolute;

    right: 0;

    top: 0;

  }

  .topnav.responsive a {

    float: none;

    display: block;

    text-align: left;

  }

}

Először a „body” tagen belül a navigációs sáv hátterének színét írtuk meg.

Az oldal egységes megjelenését a CSS fájlban lévő osztályok és azonosítók biztosítják.

A reszponzivitáshoz szükséges beállítások is itt találhatóak, például a „@media screen …” sorok.

Amikor a menüsoron a gombok felé húzzuk a kurzort, akkor a gombok színe megváltozik. Erre szolgál a „:hover” elem.

A Főoldalon továbbá található egy slideshow felület is. A sorban megjelenő képek alján található kis pontokra kattintva lehet váltani a képek között. Ez arra szolgál, hogy a webáruház termékeiből ízelítőt mutassunk be a vásárlóknak.

HTML kód:

<!-- Slideshow -->

      <div id="hcg-slider-1" class="hcg-slider">

        <div class="hcg-slide-container">

        <div class="hcg-slider-body">

          <a class="hcg-slides animated" style="display:flex">

            <img src="https://www.html-code-generator.com/images/slider/1.png" alt="image 1">

            <p class="hcg-slide-text">image 1</p>

          </a>

        </div>

      </div>

      <div class="hcg-slide-dot-control"></div>

      </div>

CSS kód:

#hcg-slider-1 .hcg-slide-container {

    width: 90%;

    height: auto;

}

.hcg-slider {

    text-align: center;

    font-family: Arial, Helvetica, sans-serif;

}

.hcg-slide-container {

    max-width: 100%;

    display: inline-block;

    position: relative;

}

.hcg-slides {

    display: none;

    overflow: hidden;

    justify-content: center;

    align-items: center;

    border-radius: 5px;

    border: solid 1px #a0a0a0;

}

.hcg-slides img {

    width: 100%;

    max-height: 100%;

    display: inline-block;

}

.hcg-slide-text {

    color: #ffffff;

    font-size: 14px;

    padding: 3px 5px;

    position: absolute;

    bottom: 0;

    border-radius: 5px;

    left: 50%;

    text-align: center;

    text-shadow: 0 0 2px #000;

    background-color: rgba(255,255,255,0.30);

    display: inline-block;

    transform: translate(-50%, -5px);

}

.hcg-slide-dot-control {

    margin-top: 10px;

    text-align: center;

}

.hcg-slide-dot {

    cursor: pointer;

    height: 13px;

    width: 13px;

    margin: 0 2px;

    background-color: #bbb;

    border-radius: 50%;

    display: inline-block;

}

.hcg-slide-dot.dot-active {

    background-color: #717171;

}

.animated {

    animation-name: vanishIn;

    animation-duration: 1s;

}

@keyframes vanishIn {

    0% {

        opacity: 0;

        transform-origin: 50% 50%;

        transform: scale(2, 2);

        -webkit-filter: blur(90px);

    }

    100% {

        opacity: 1;

        transform-origin: 50% 50%;

        transform: scale(1, 1);

        -webkit-filter: blur(0px);

    }

}

.vanishIn {

    animation-name: vanishIn;

}

A slideshow a termék kategóriákat mutatja be, ezért próbáltunk egy díszes kinézetet létrehozni neki.

Amikor a képek között váltunk, egy animáció játszódik le, elhalványulás után jelennek meg a képek.

JavaScript kód:

(() => {

        const images\_list = [

    {

        "src": "elso.jpg",

        "alt": "",

        "name": "Nyakláncok, Gyűrűk, Karkötők",

        "link": ""

    },

    {

        "src": "masodik.jpg",

        "alt": "",

        "name": "Arany és Ezüst ékszerek",

        "link": ""

    },

    {

        "src": "harmadik.png",

        "alt": "",

        "name": "Drágaköves ékszerek",

        "link": ""

    }

        ];

        let slider\_id = document.querySelector("#hcg-slider-1");

        let dots\_div = "";

        let images\_div = "";

        for (let i = 0; i < images\_list.length; i++) {

            let href = (images\_list[i].link == "" ? "":' href="'+images\_list[i].link+'"');

            images\_div += '<a'+href+' class="hcg-slides animated"'+(i === 0 ? ' style="display:flex"':'')+'>'+

                            '<img src="'+images\_list[i].src+'" alt="'+images\_list[i].name+'">'+

                            '<span class="hcg-slide-text">'+images\_list[i].name+'</span>'+

                         '</a>';

            dots\_div += '<a href="#" class="hcg-slide-dot'+(i === 0 ? ' dot-active':'')+'" data-id="'+i+'"></a>';

        }

        slider\_id.querySelector(".hcg-slider-body").innerHTML = images\_div;

        slider\_id.querySelector(".hcg-slide-dot-control").innerHTML = dots\_div;

        let slide\_index = 0;

        const images = slider\_id.querySelectorAll(".hcg-slides");

        const dots = slider\_id.querySelectorAll(".hcg-slide-dot");

        const showSlides = () => {

            if (slide\_index > images.length-1) {

                slide\_index = 0;

            }

            if (slide\_index < 0) {

                slide\_index = images.length-1;

            }

            for (let i = 0; i < images.length; i++) {

                images[i].style.display = "none";

                dots[i].classList.remove("dot-active");

                if (i == slide\_index) {

                    images[i].style.display = "flex";

                    dots[i].classList.add("dot-active");

                }

            }

        }

        const dot\_click = event => {

            event.preventDefault();

            slide\_index = event.target.dataset.id;

            showSlides();

        }

        for (let i = 0; i < dots.length; i++) {

            dots[i].addEventListener("click", dot\_click, false);

        }

        setInterval(() => {

            slide\_index++;

            showSlides();

        }, 10000);

    })();

A hozzákapcsolódó JavaScript kód által lesz a képek megjelenése animált.

Az alatta megjelenő pontokra kattintva lehet váltani, hogy melyik kép jelenjen meg.

Az oldal legalján található egy footer, avagy lábléc. Ennek a funkciója az, hogy az egyéb információkat megjelenítse, például az elérhetőség. Továbbá ezen keresztül is el lehet érni a weboldal Áruház részét.

HTML kód:

<!-- Footer -->

      <footer class="site-footer" >

        <div class="container" >

          <div class="row" style="display: flex;">

            <div id="rolunk" class="col-sm-12 col-md-6" >

              <h6 >Rólunk</h6>

              <p>Vásárolj ékszereket ajándékba, vagy magadnak! </p>

            </div>

            <div id="kategoria" class="col-xs-6 col-md-3" >

              <h6>Kategórai</h6>

              <ul class="footer-links">

                <li><a href="aruhaz.html">Arany Ékszerek</a></li><br>

                <li><a href="aruhaz.html">Ezüst Ékszerek</a></li><br>

                <li><a href="aruhaz.html">Drágaköves Ékszerek</a></li>

              </ul>

            </div>

            <div id="elerhetoseg" class="col-xs-6 col-md-3" >

              <h6>Elérhetőségek</h6>

              <ul class="footer-links">

                <li><a href="elerhetoseg.html">E-mail</a></li>

                <li><a href="elerhetoseg.html">Telefonszám</a></li>

                <li><a href="elerhetoseg.html">Facebook</a></li>

                <li><a href="elerhetoseg.html">Ügyfélszolgálat</a></li>

              </ul>

            </div>

          </div>

          <hr>

        </div>

        <div class="container">

          <div class="row">

          </div>

        </div>

    </footer>

CSS kód:

#rolunk{

  width: 50%;

}

#kategoria{

  width: 20%;

}

.site-footer

{

  background-color:#26272b;

  padding:45px 50px 20px;

  font-size:15px;

  line-height:24px;

  color:#737373;

}

.site-footer hr

{

  border-top-color:#bbb;

  opacity:0.5;

}

.site-footer h6

{

  color:#fff;

  font-size:13px;

  text-transform:uppercase;

  margin-top:5px;

  letter-spacing:2px;

  margin-right: 20px;

  margin-left: auto;

}

.container{

  display: inline;

}

.site-footer a

{

  color:#737373;

}

.site-footer a:hover

{

  color:#3366cc;

  text-decoration:none;

}

.footer-links

{

  padding-left:0;

  list-style:none;

}

.footer-links li

{

  display:block

}

.footer-links a

{

  color:#737373;

}

.footer-links a:active,.footer-links a:focus,.footer-links a:hover

{

  color:#3366cc;

  text-decoration:none;

}

.footer-links.inline li

{

  display:inline-block;

}

.site-footer .social-icons

{

  text-align:right;

}

.site-footer .social-icons a

{

  width:40px;

  height:40px;

  line-height:40px;

  margin-left:6px;

  margin-right:0;

  border-radius:100%;

  background-color:#33353d

}

.copyright-text

{

  margin:0

}

@media (max-width:991px)

{

  .site-footer [class^=col-]

  {

    margin-bottom:30px

  }

}

@media (max-width:767px)

{

  .site-footer

  {

    padding-bottom:0

  }

  .site-footer .site-footer

  {

    text-align:center;

    display: inline;

  }

}

A footer részt 3 felé osztottuk: a bal oldalán egy rövidebb üzenet található, közepén a főbb kategóriák, amikkel szintén el lehet jutni az áruház oldalára, a jobb oldalán pedig az oldal elérhetőségeit lehet megtalálni.

### Áruház

Az Áruház az oldal lényegi része, itt lehet kiválasztani a megvásárolandó termékeket. A „Kosárba” feliratú gombra kattintva történik meg a kiválasztás. A termékek kiválogatása után az oldal alján jelenik meg a kiválasztott ékszerek listája. A mennyiséget is itt lehet módosítani, és a vásárlás végösszege is ezen a helyen látható. Ha az adatok megfelelőek, a Vásárlás feliratú gombra kell kattintani a vásárlási folyamat befejezéséhez.

HTML kód:

<!--Áruk-->

  <div class="aruhazTermekek">

    <table>

      <tr class="termekKartya-sor">

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer1.jpg">

          <p class="aruNev">Nyaklánc szett hozzá illő fülbevalóval</p>

          <p class="aruAr">Ár: <span class="aruArSzam">60000 Ft</span></p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer2.jpg">

          <p class="aruNev">Nyaklánc együttes hozzá illő fülbevalóval</p>

          <p class="aruAr">Ár: <span class="aruArSzam">50000 Ft</span> </p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer3.jpg">

          <p class="aruNev">Nyaklánc csomag hozzá illő fülbevalóval</p>

          <p class="aruAr">Ár: <span class="aruArSzam">70000 Ft</span> </p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

      </tr>

      <tr class="termekKartya-sor">

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer4.jpg">

          <p class="aruNev">Arany Gyűrű</p>

          <p class="aruAr">Ár: <span class="aruArSzam">500000 Ft</span> </p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer5.jpg">

          <p class="aruNev">Arany Gyűrű Gyémánttal</p>

          <p class="aruAr">Ár: <span class="aruArSzam">1200000 Ft</span> </p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer6.jpg">

          <p class="aruNev">Ezüst Gyűrű Arany díszesítéssel</p>

          <p class="aruAr">Ár: <span class="aruArSzam">800000 Ft</span> </p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

      </tr>

      <tr class="termekKartya-sor">

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer7.png">

          <p class="aruNev">Arany Fülbevaló</p>

          <p class="aruAr">Ár: <span class="aruArSzam">75000 Ft</span> </p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer8.jpg">

          <p class="aruNev">Arany Fülbevaló Zöld Ékkővel</p>

          <p class="aruAr">Ár: <span class="aruArSzam">100000 Ft</span> </p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer9.jpg">

          <p class="aruNev">Fülbevaló Gyémánttal</p>

          <p class="aruAr">Ár: <span class="aruArSzam">120000 Ft</span></p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

      </tr>

      <tr class="termekKartya-sor">

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer10.png">

          <p class="aruNev">Karkötő</p>

          <p class="aruAr">Ár: <span class="aruArSzam">200000 Ft</span> </p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer11.png">

          <p class="aruNev">Gyémánt Karkötő</p>

          <p class="aruAr">Ár: <span class="aruArSzam">600000 Ft</span> </p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

        <td class="termekKartya">

          <img class="aruKep" src="ekszerek/ekszer12.png">

          <p class="aruNev">Arany Karkötő</p>

          <p class="aruAr">Ár: <span class="aruArSzam">1000000 Ft</span> </p>

          <button class="kosarbaGomb">Kosárba</button>

        </td>

      </tr>

    </table>

  </div>

  <div id="vasarlas" class="container content-section">

    <h2 class="section-header">Kosár</h2>

    <div class="cart-row">

        <span class="cart-item cart-header cart-column">Termék</span>

        <span class="cart-price cart-header cart-column">Ár</span>

        <span class="cart-quantity cart-header cart-column">Mennyiség</span>

    </div>

    <div class="cart-items">

    </div>

    <div class="cart-total">

        <strong class="cart-total-title">Összeg</strong>

        <span class="cart-total-price">0Ft</span>

    </div>

    <button class="btn btn-primary btn-purchase" type="button">Vásárlás</button>

  </div>

CSS kód:

/\* Áruház termékek \*/

html{

    width: 100%;

}

table{

    margin: auto;

    width: 80%;

}

td{

    display: table-cell;

    width: 20%;

    text-align: center;

    overflow: hidden;

    vertical-align: baseline;

    padding: 20px;

}

img{

    width: 100%;

}

.kosarbaGomb{

    display: inline-block;

    padding: 5px 15px;

    font-size: 24px;

    cursor: pointer;

    text-align: center;

    text-decoration: none;

    outline: none;

    color: black;

    background-color: aquamarine;

    border: none;

    border-radius: 15px;

}

.kosarbaGomb:hover{

    background-color: aqua;

}

.kosarbaGomb:active{

    background-color: aqua;

    transform: translateY(4px);

}

.aruKep{

    border-radius: 30%;

}

JavaScript kód:

if (document.readyState == 'loading') {

    document.addEventListener('DOMContentLoaded', kesz)

} else {

    kesz()

}

function kesz() {

    var removeCartItemButtons = document.getElementsByClassName('btn-danger')

    for (var i = 0; i < removeCartItemButtons.length; i++) {

        var gomb = removeCartItemButtons[i]

        gomb.addEventListener('click', removeCartItem)

    }

    var quantityInputs = document.getElementsByClassName('cart-quantity-input')

    for (var i = 0; i < quantityInputs.length; i++) {

        var input = quantityInputs[i]

        input.addEventListener('change', quantityChanged)

    }

    var addToCartButtons = document.getElementsByClassName('kosarbaGomb')

    for (var i = 0; i < addToCartButtons.length; i++) {

        var gomb = addToCartButtons[i]

        gomb.addEventListener('click', addToCartClicked)

    }

    document.getElementsByClassName('btn-purchase')[0].addEventListener('click', purchaseClicked)

}

function purchaseClicked() {

    alert('Köszönjük a vásárlást!')

    var cartItems = document.getElementsByClassName('cart-items')[0]

    while (cartItems.hasChildNodes()) {

        cartItems.removeChild(cartItems.firstChild)

    }

    updateCartTotal()

}

function removeCartItem(event) {

    var buttonClicked = event.target

    buttonClicked.parentElement.parentElement.remove()

    updateCartTotal()

}

function quantityChanged(event) {

    var input = event.target

    if (isNaN(input.value) || input.value <= 0) {

        input.value = 1

    }

    updateCartTotal()

}

function addToCartClicked(event) {

    var button = event.target

    var shopItem = button.parentElement

    var title = shopItem.getElementsByClassName('aruNev')[0].innerText

    var price = shopItem.getElementsByClassName('aruArSzam')[0].innerText

    var imageSrc = shopItem.getElementsByClassName('aruKep')[0].src

    addItemToCart(title, price, imageSrc)

    updateCartTotal()

}

function addItemToCart(title, price, imageSrc) {

    var cartRow = document.createElement('div')

    cartRow.classList.add('cart-row')

    var cartItems = document.getElementsByClassName('cart-items')[0]

    var cartItemNames = cartItems.getElementsByClassName('cart-item-title')

    for (var i = 0; i < cartItemNames.length; i++) {

        if (cartItemNames[i].innerText == title) {

            alert('Ez a termék már hozzá lett adva a kosárhoz!')

            return

        }

    }

    var cartRowContents = `

        <div class="cart-item cart-column">

            <img class="cart-item-image" src="${imageSrc}" width="100" height="100">

            <span class="cart-item-title">${title}</span>

        </div>

        <span class="cart-price cart-column">${price}</span>

        <div class="cart-quantity cart-column">

            <input class="cart-quantity-input" type="number" value="1">

            <button class="btn btn-danger" type="button">Eltávolítás</button>

        </div>`

    cartRow.innerHTML = cartRowContents

    cartItems.append(cartRow)

    cartRow.getElementsByClassName('btn-danger')[0].addEventListener('click', removeCartItem)

    cartRow.getElementsByClassName('cart-quantity-input')[0].addEventListener('change', quantityChanged)

}

function updateCartTotal() {

    var cartItemContainer = document.getElementsByClassName('cart-items')[0]

    var cartRows = cartItemContainer.getElementsByClassName('cart-row')

    var total = 0

    for (var i = 0; i < cartRows.length; i++) {

        var cartRow = cartRows[i]

        var priceElement = cartRow.getElementsByClassName('cart-price')[0]

        var quantityElement = cartRow.getElementsByClassName('cart-quantity-input')[0]

        var price = parseInt(priceElement.innerText.replace('Ft', ''))

        var quantity = quantityElement.value

        total = total + (price \* quantity)

    }

    document.getElementsByClassName('cart-total-price')[0].innerText = total + 'Ft'

}

Ha a vásárlást befejeztük, egy push üzenetet kapunk, ami tájékoztatja a vásárlót arról, hogy a vásárlási adatok rögzítve lettek.

A böngészési élmény és az adatbiztonság biztosítása érdekében az oldal alján egy felugró ablakban jelenik meg a Cookie-k beállításának lehetősége.

HTML kód:

<!-- Cookie-k elfogadása popup ablak -->

    <div id="cookie" class="hide">

      <div id="cookieNotice" class="light display-right" style="display: none;">

        <div id="closeIcon" style="display: none;">

        </div>

      <div class="title-wrap">

        <h4>Cookie Consent</h4>

      </div>

        <div class="content-wrap">

          <div class="msg-wrap">

            <p> Ez a weboldal cookie-kat vagy hasonló technológiákat használ a böngészési élmény fokozása és személyre szabott ajánlások biztosítása

               érdekében. <br>Weboldalunk további használatával Ön elfogadja az  <a style="color:#115cfa;" href="/privacy-policy"> Adatvédelmi szabályzatunkat </a></p>

            <div class="btn-wrap">

              <button class="btn-primary" onclick="acceptCookieConsent();">Elfogad</button>

            </div>

          </div>

        </div>

      </div>

    </div>

CSS kód:

/\* Cookie CSS \*/

#cookie{

    position: fixed;

    bottom: 0;

    left: 0;

    z-index: 2147483645;

    box-sizing: border-box;

    width: 100%;

    color: aliceblue;

    background-color: #26272b;

    text-align: center;

}

#cookie button{

    display: inline-block;

    padding: 5px 15px;

    font-size: 15px;

    cursor: pointer;

    text-align: center;

    text-decoration: none;

    outline: none;

    color: black;

    background-color: grey;

    border: none;

    margin-bottom: 10px;

}

#cookie button:hover{

    background-color: darkgrey;

}

JavaScript kód:

// Sütik létrehozása

function setCookie(cname, cvalue, exdays) {

    const d = new Date();

    d.setTime(d.getTime() + (exdays\*24\*60\*60\*1000));

    let expires = "expires="+ d.toUTCString();

    document.cookie = cname + "=" + cvalue + ";" + expires + ";path=/";

}

// Sütik törlése

function deleteCookie(cname) {

    const d = new Date();

    d.setTime(d.getTime() + (24\*60\*60\*1000));

    let expires = "expires="+ d.toUTCString();

    document.cookie = cname + "=;" + expires + ";path=/";

}

// Sütik be olvasása

function getCookie(cname) {

    let name = cname + "=";

    let decodedCookie = decodeURIComponent(document.cookie);

    let ca = decodedCookie.split(';');

    for(let i = 0; i <ca.length; i++) {

        let c = ca[i];

        while (c.charAt(0) == ' ') {

            c = c.substring(1);

        }

        if (c.indexOf(name) == 0) {

            return c.substring(name.length, c.length);

        }

    }

    return "";

}

// Sütik bele egyesési funkció létrehozása

function acceptCookieConsent(){

    deleteCookie('user\_cookie\_consent');

    setCookie('user\_cookie\_consent', 1, 30);

    document.getElementById("cookieNotice").style.display = "none";

}

// Ellenörzési funkció

// Ha el van fogadva a süti akkor eltűnik a popup ablak ha frissítve van az oldal akkor megint felugrik.

let cookie\_consent = getCookie("user\_cookie\_consent");

if(cookie\_consent != ""){

    document.getElementById("cookieNotice").style.display = "none";

}else{

    document.getElementById("cookieNotice").style.display = "block";

}

## Visual Studio Code Leírása

### IntelliSense a Visual Studio Code-ban:

Az IntelliSense a VS Code-hoz készült Azure Stream Analytics (ASA-) eszközök Stream Analytics-lekérdezési nyelvéhez érhető el.

Az IntelliSense egy kódkiegészítési segéd, amely számos funkciót tartalmaz: Listatagok, Paraméteradatok, Gyors információ és Teljes word.

Az IntelliSense funkcióit néha más nevek is hívják, például "kódkiegészítés", "tartalomsegéd" és "kódmutató".

### A kiegészítések típusai

A VS Code IntelliSense különböző típusú kiegészítéseket biztosít, beleértve a nyelvi kiszolgáló javaslatait, kódrészleteit és egyszerű szóalapú szöveges kiegészítéseket.

|  |  |
| --- | --- |
| **Befejezés** | **Típus** |
| Kulcsszavak | keyword |
| Functions | build-in function, user defined function |
| Adatkészlet neve | input, output, intermediate result set |
| Adatkészlet oszlopának neve | input, intermediate result set |

### IntelliSense-funkciók a visual studio code-ban:

A VS Code-hoz készült ASA Tools bővítmény intelliSense funkcióval rendelkezik, amelyet egy nyelvi szolgáltatás működtet.

A nyelvi szolgáltatás elemzi a forráskódot, és intelligens kódkiegészítéseket biztosít a nyelvi szemantika alapján.

Ha egy nyelvi szolgáltatás ismeri a lehetséges befejezéseket, akkor a kód írása közben megjeleníti az IntelliSense javaslatait.

Ha folytatja a gépelést, a rendszer úgy szűri a tagok listáját, mint a változók és a metódusok, hogy csak a beírt karaktereket tartalmazó tagokat tartalmazzák.

### Nyelvkiegészítés:

A kulcsszó automatikus kiegészítésén kívül az ASA Tools bővítmény képes beolvasni a Stream Analytics-feladat bemeneti és kimeneti nevét, valamint az adatforrások oszlopneveit. A bővítmény megjegyzi ezeket az információkat, hogy olyan névkiegészítési képességeket biztosítson, amelyek hasznosak néhány billentyűleütéssel rendelkező utasítások megadásához:

A kódolás során nem kell elhagynia a szerkesztőt, hogy keresést végezzen a feladat bemeneti neveiben, a kimeneti névben és az oszlopnevekben. Megtarthatja a kontextust, megkeresheti a szükséges információkat, elemeket szúrhat be közvetlenül a kódba, és beállíthatja, hogy az IntelliSense befejezze a gépelést.

Vegye figyelembe, hogy a névkiegészítés használatához konfigurálnia kell a helyi bemenetet vagy az élő bemenetet, majd mentenie kell a konfigurációs fájlt.

### Paraméteradatok:

Az IntelliSense Paraméter adatai lehetőség megnyitja a paraméterlistát, amely információt nyújt a függvény által igényelt paraméterek számáról, nevéről és típusairól. A félkövér betűvel jelölt paraméter a függvény beírásához szükséges következő paramétert jelzi.

A paraméterlista beágyazott függvényekhez is megjelenik. Ha egy függvényt egy másik függvény paramétereként ír be, a paraméterlista megjeleníti a belső függvény paramétereit. Ezután, amikor a belső függvény paraméterlistája befejeződött, a paraméterlista visszaáll a külső függvényparaméterek megjelenítésére.

### Gyors Információ:

A nyelvi szolgáltatás által biztosítottaknak megfelelően a kódban szereplő egyes azonosítók gyors információi láthatók. Néhány példa az azonosítókra: bemenet, kimenet, köztes eredményhalmaz vagy függvény. Amikor az egérmutatót egy azonosító fölé viszi, annak deklarációja megjelenik egy előugró ablakban. A bemenetek tulajdonságai és adatsémái, ha konfigurálva vannak, és a köztes adatkészlet is megjelenik.

### Az Stream Analytics Code-ban végzett Visual Studio tesztelésének áttekintése ASA-eszközökkel:

Az Azure Stream Analytics Tools (ASA **Tools**) for Visual Studio Code használatával helyileg tesztelheti a Stream Analytics feladatokat. A területi beállításoknál három szempontot kell figyelembe venni: a feladat-végrehajtási környezetet (egy helyi gép vagy az Azure-felhőszolgáltatás), a bemeneti forrásokat és a kimeneti fogadókat.

Helyi futtatáskor a lekérdezés a helyi gépen lesz végrehajtva. Bemenetként az adatok helyi fájlokból vagy élő forrásokból is be lesznek ússodva. A kimeneti eredményeket a rendszer fájlokként küldi el egy helyi mappába vagy az élő fogadóknak.

### A helyi futtatás bemeneti szempontjai

A VS Code-ban élő és helyi bemeneteket definiálhat:

* Az élő bemenetek konfigurációs fájlok, amelyek a támogatott bemenetek egy példányára mutatnak (streamelési és referenciaadatok). Emellett JSON-fájlokba való előnézeti és mintaadatokat is kínálnak.
* A helyi bemenetek olyan konfigurációs fájlok, amelyek a támogatott formátumú (JSON/CSV/AVRO) helyi fájlra mutatnak. Ezek a fájlok mintavételt kaphatnak egy élő bemenetből, vagy bármilyen más módon generálhatók.

Helyi bemenet létrehozásakor az egy meglévő élő bemenethez igazítható. Ebben az esetben a helyi bemeneti futtatás során az élő bemenetet fogja utánzatként. A konfigurációs fájl neve az élő bemenet után lesz elnevezve, előtagja: Local\_. A helyi bemenet által használt adatfájlnak nem kell követnie az élő bemenetben meghatározott formátumot és szerializálási formátumot. A formátumaik függetlenek.

### A helyi futtatás kimeneti szempontjai

Amikor helyi kimenetekbe futtat egy feladatot, a kimeneti eredményeket a rendszer a LocalRunOutputs nevű projekt egyik mappájába küldi. Ebben a módban a kimeneteket nem kell definiálni. Az egyetlen korlátozás az INTO, hogy a lekérdezés minden utasítása egyedi kimeneti névre mutat. A helyi kimenetekbe való futtatás után minden egyedi kimeneti névhez létrejön egy JSON-fájl.

### Helyi futtatás módjai

A VS Code ASA-eszközei három módot támogatnak a feladatok helyi futtatásához:

* Helyi futtatás helyi bemenettel és helyi kimenettel: offline fejlesztéshez költség nélkül, egységtesztelés az npm csomaggal...
* Helyi futtatás élő bemenettel és helyi kimenettel: a legjobb választás a bemeneti konfigurációhoz, a de szerializáláshoz és a particionálási hibakereséshez...
* Helyi futtatás élő bemenettel és élő kimenettel: a legjobb választás kimeneti konfigurációhoz, szerializáláshoz és átalakítási hibák hibakereséséhez...

Mindegyik mód különböző bemeneti és kimeneti konfigurációkat támogat:

|  |  |  |  |
| --- | --- | --- | --- |
| **Futtatási** | **Mód** | **Input (Bemenet)** | **Kimenet** |
| VS Code | Helyi bemenet a helyi kimenethez | JSON-/CSV-/AVRO-fájlok | JSON-fájlok (a megfelelő élő kimeneti formátum akkor sem lesz használva, ha létezik) |
| VS Code | Élő bemenet helyi kimenetbe | Minden bemeneti adapter | JSON-fájlok (a megfelelő élő kimeneti formátum akkor sem lesz használva, ha létezik) |
| VS Code | Élő bemenet élő kimenethez | Minden bemeneti adapter | Event Hub, Storage Account, Azure SQL |
| Azure | N/A | Minden bemeneti adapter | Minden kimeneti adapter |

A feladatok helyi futtatásakor nem jár költség a Azure Stream Analytics szolgáltatásból. Az Azure-ban nem Stream Analytics erőforrás létrehozása.

### Stream Analytics-lekérdezések helyi tesztelése mintaadatokkal a Visual Studio Code használatával

A Visual Studio Code-hoz készült Azure Stream Analytics-eszközökkel helyileg tesztelheti a Stream Analytics-feladatokat mintaadatokkal. A kimeneti eredmények JSON-fájlokként találhatók a projekt LocalRunOutputs mappájában.

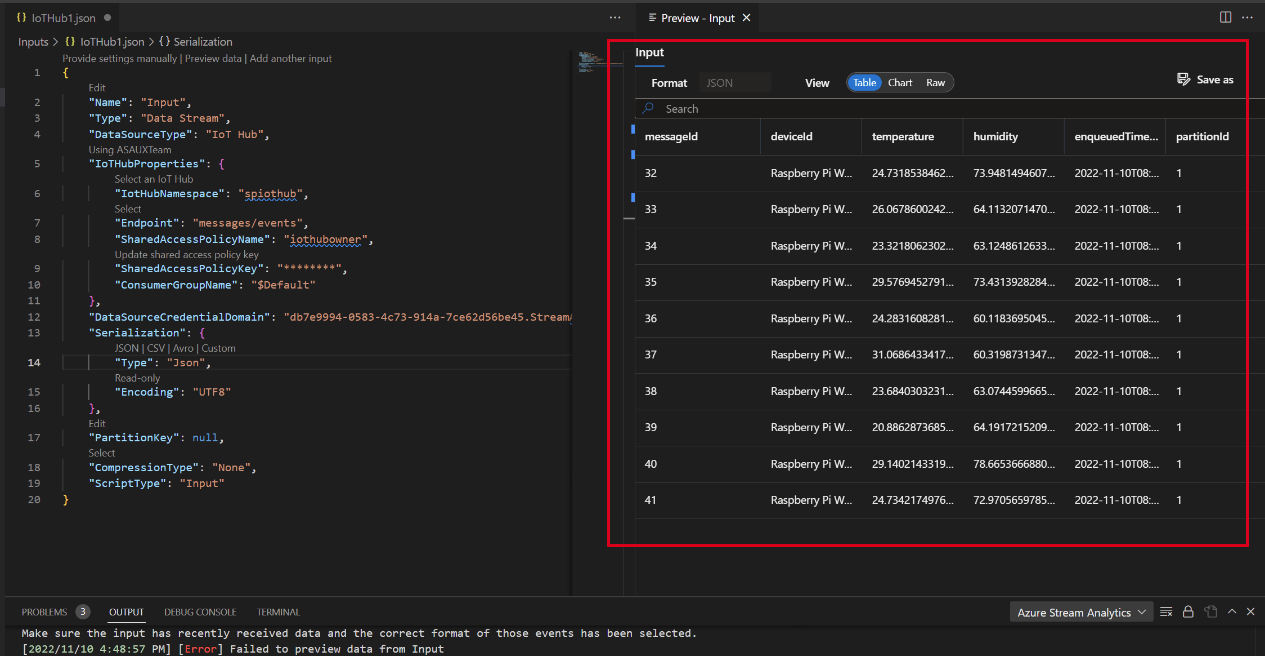
**Előfeltételek**

* Telepítse a .NET Core SDK-t, és indítsa újra a Visual Studio Code-ot.

### Minta adatok előkészítése:

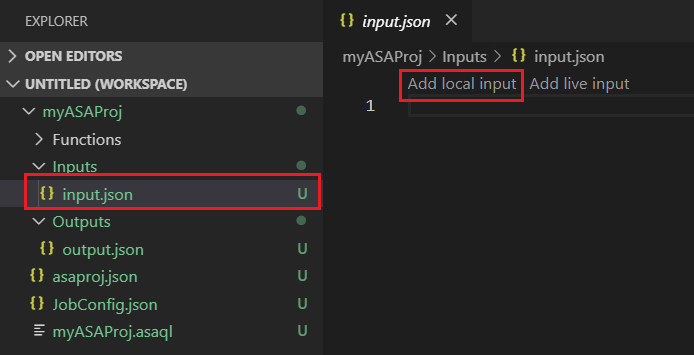
Először elő kell készítenie a bemeneti mintaadatfájlokat. Ha már rendelkezik néhány mintaadatfájllal a gépen, kihagyhatja ezt a lépést, és továbbléphet a következőre.

1. A felső sorból kattintson az **Adatok előnézete** elemre a bemeneti konfigurációs fájlban. A bemeneti adatok egy része IoT Hub lesz beolvasva, és megjelenik az előnézeti ablakban.
2. Az adatok megjelenítése után válassza a **Mentés másként** lehetőséget az adatok helyi fájlba való mentéséhez.

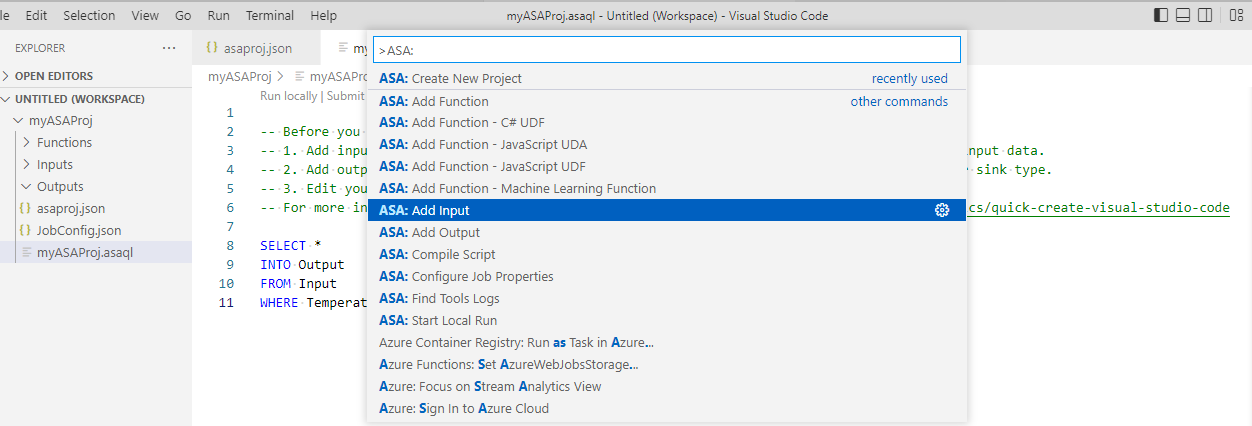


**Helyi bemenet definiálása**

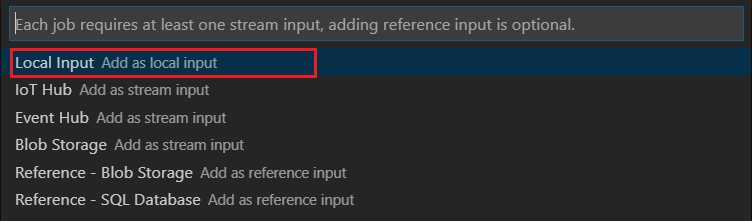
1. Válassza az **input.json** elemet a Stream Analytics-projekt Inputs mappájában. Ezután válassza a **Felső sor Helyi bemenet hozzáadása** elemét.



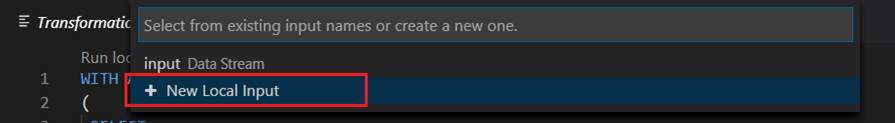
1. A **Ctrl+Shift+P billentyűkombinációval** megnyithatja a parancskatalógust, és beírhatja az **ASA: Bemenet hozzáadása parancsot**.



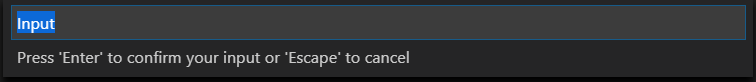
1. Válassza a **Helyi bemenet lehetőséget**.



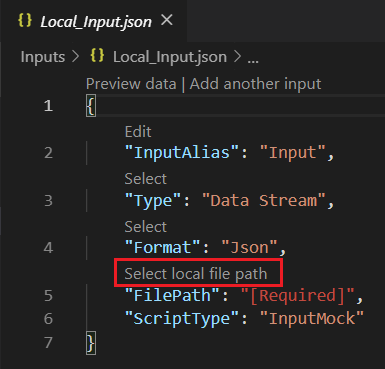
1. Válassza az **Új helyi bemenet lehetőséget**.



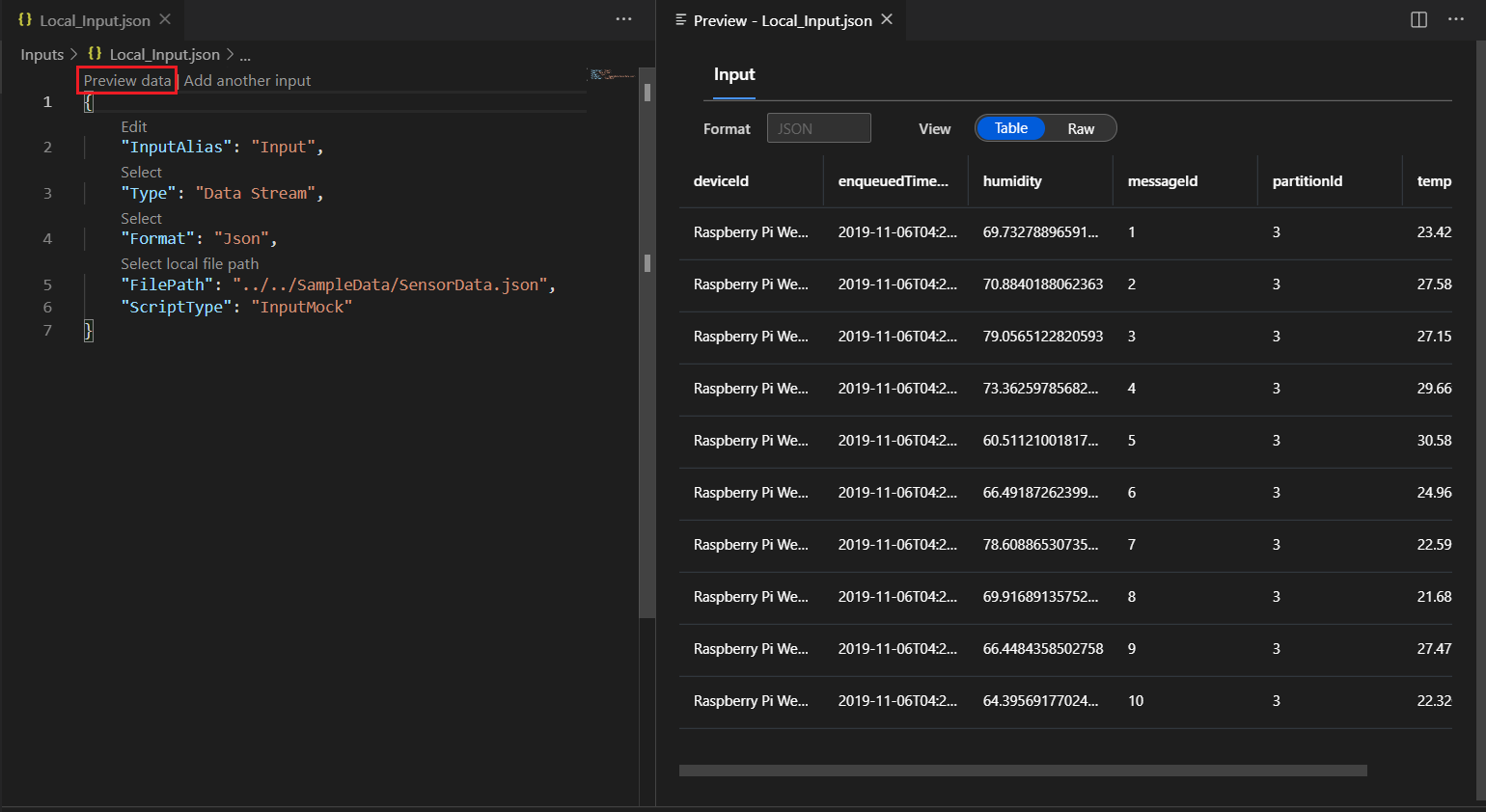
1. Adja meg ugyanazt a bemeneti aliast, amelyet a lekérdezésben használt.



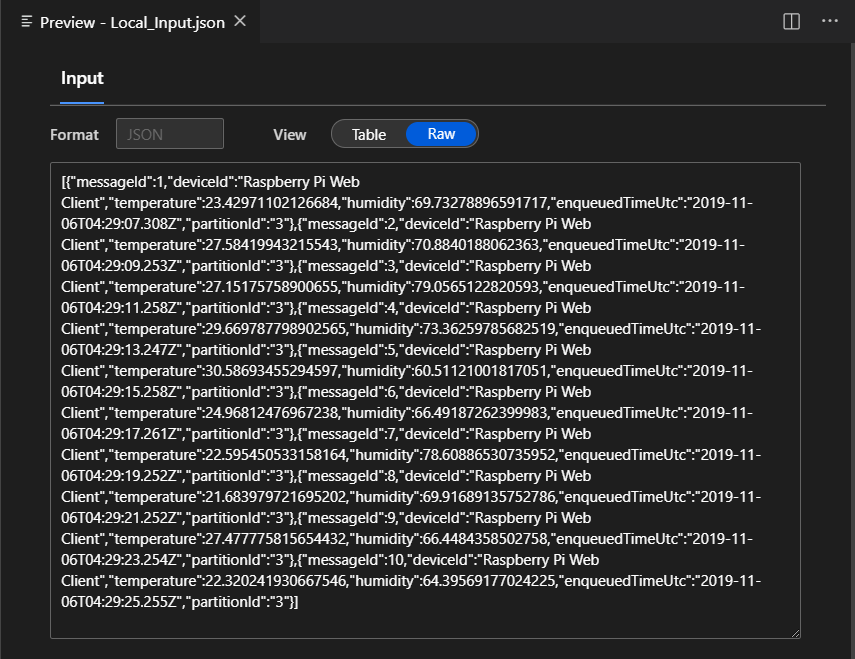
1. Az újonnan létrehozott **LocalInput\_Input.json** fájlban adja meg a helyi adatfájl elérési útját.



1. A bemeneti adatok előnézetének megtekintéséhez válassza az **Adatok előnézete** lehetőséget. A rendszer automatikusan észleli az adatok szerializálási típusát (JSON vagy CSV). A választóval **táblázatos** vagy **nyers** formátumban tekintheti meg az adatokat. Az alábbi táblázat **táblázatformátumú** adatokra mutat be példát:

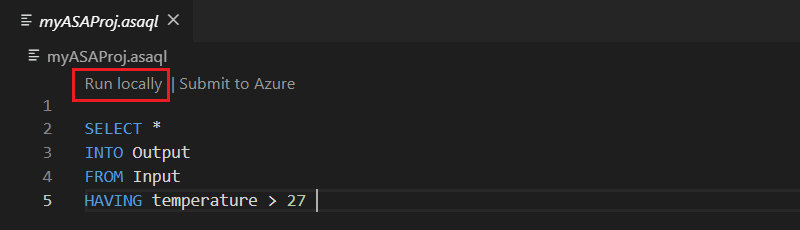


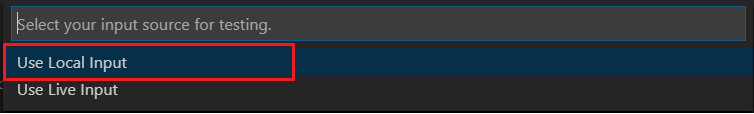
Az alábbi táblázat a **Nyers formátumú** adatokra mutat példát:



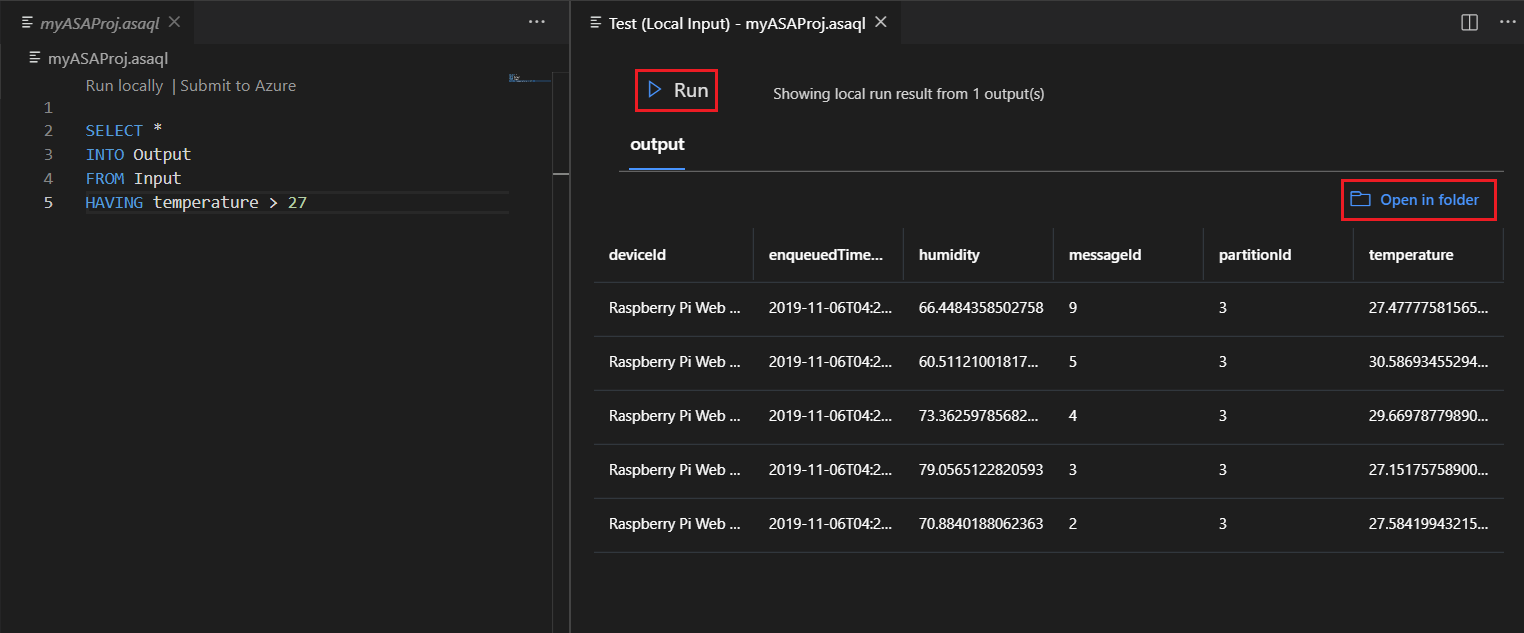
### Lekérdezések helyi futtatása:

Térjen vissza a lekérdezésszerkesztőbe, és válassza a **Helyi futtatás** lehetőséget. Ezután válassza a Helyi **bemenet használata** lehetőséget a legördülő listából.





Az eredmény a jobb oldali ablakban jelenik meg. Az ismételt teszteléshez válassza a **Futtatás** lehetőséget. A **Megnyitás mappában** lehetőség kiválasztásával megtekintheti az eredményfájlokat a fájlkezelőben, és más eszközökkel is megnyithatja őket. Az eredményfájlok csak JSON formátumban érhetők el.



## Felhasználói Dokumentáció:

Először dupla kattintással nyissa meg az index.html fájlt. Ha ezt megtette a weboldal kezdőlapján fogja találni magát. Innen több lehetősége is van. Tovább haladhat a bejelentkező oldalra (kattintson a bejelentkezés feliratra), ahol létrehozhat magának egy felhasználói fiókot. Ha ez elkészült, akkor lehetősége van a vásárláshoz és az eladandó ékszerek feltöltéséhez. Amennyiben már rendelkezik fiókkal, akkor egyszerűen a regisztrációkor megadott email és jelszó segítségével tud bejelentkezni. Ezután tovább haladhat az Áruház oldalára (kattintson az Áruház gombra a menüsor felületén) ahol lehetősége van elkezdeni böngészni az ékszerek között. Amennyiben ékszert szeretne eladásra bocsátani, azt a profiloldalon belül teheti meg. Amennyiben ki szeretne jelentkezni az oldalról, a profiljára kattintva a kijelentkezés gomb használatával teheti ezt meg.

Kezdőlap: A kezdőlapot HTML és Bootstrap segítségével hoztuk létre. A lap tetején egy navigációs sávot találunk, amelyet a Bootstrap navbar elemének segítségével hoztunk létre. A navigációs sávon találunk egy kezdőlap gombot, amely ugyanazon a lapon megnyitja nekünk a kezdőlapot, egy áruház gombot, amely megnyitja a weboldal áruház részét, és egy bejelentkezés gombot, amellyel a profilunkba van lehetőségünk belépni. Továbbá találunk még egy beviteli mezőt és egy gombot, amelyek keresésre adnak lehetőséget az oldalon belül. Az oldal fő részén találunk két oszlopot egy-egy reklám felülettel. Az oldal közepén diavetítés formájában a megvásárolható termékekről láthatunk képeket. A láblécben találunk, egy elérhetőségek részt ahol a cég elérhetőségeit tudjuk megtekinteni.

Áruház oldal: Az áruház oldalon találjuk a megvásárolható termékeket, amelyeket kártyák segítségével jelenítettünk meg. Csak is regisztrált fiókkal rendelkező felhasználóknak van lehetőségük vásárlásra. A termékek között anyag és ár alapján lehet válogatni.

Bejelentkező Fül: A felhasználói fiók létrehozásának lehetőségét biztosítja. Amennyiben rendelkezik a felhasználó felhasználói fiókkal, abban az esetben a bejelentkezés elérhető. A regisztrációhoz felhasználó névre, e-mail címre, jelszóra és jelszó megerősítésre van szükség.

### Rendszerkövetelmények:

Főbb szolgáltatások amik kellenek:

* + Visual Studio Code
    1. CSS
    2. PHP
    3. HTML5
  + Power Point
  + Microsoft World
  + XAMPP
    1. local server futtatására a bejelentkező oldalon megadott adatok tárolására

**Minimum követelmények a megtekintéshez:**

* + 8 GB RAM
  + Intell Core i38140U processzor
  + 250GB SSDIp cím
  + Router(TP-Link 250Mb/s)
  + Billentyűzet
  + Egér
  + Monitor (1280x720P felbontás)
  + Windows 10 operációs rendszer

**Ajánlott követelmények a megtekintéshez a minimumon követelményeken kívül:**

* + 16GB RAM
  + Intell Core i3 9100F processzor
  + 650W tápegység
  + Monitor (1920x1080P felbontás)

### Böngészők

Webböngészőnek vagy böngészőnek (angolul browser) nevezzük azon programokat, amelyekkel az interneten található tartalmakat – legtöbbször weblapokat – lehet megtekinteni, illetve az interneten keresztül elérhető szolgáltatásokat használni.

**Szabványok és protokollok:**

A webböngészők a webszerverekkel HTTP (Hyper Text Transfer Protocol) protokollon keresztül kommunikálnak. A HTTP segítségével a böngészők adatokat küldhetnek a webszervereknek, valamint weblapokat tölthetnek le róluk. A leggyakrabban használt HTTP szabvány a http:// 1.1, melynek teljes leírása az RFC 2616 alatt található meg.

A lapokat a böngésző az URL (más néven webcím) segítségével találja meg, mely a lap címét jelöli. Az URL a címhez tartozó protokollal kezdődik, például a http: a HTTP protokoll jelölése. Sok böngésző több más protokollt is támogat, mint például az ftp: az FTP, a gopher: a Gopher, a https: pedig a HTTPS protokollhoz.

A weblaphoz tartozó fájl formátuma többnyire HTML, ez a HTTP protokollban a MIME tartalomtípus alatt van megadva. A legtöbb böngésző a HTML mellett támogat más formátumokat is, mint amilyen a JPEG, PNG és GIF képtípusok, valamint tovább bővíthető bennük a támogatott fájltípusok listája különböző beépülők használatával. A HTTP tartalom és az URL használatával a webfejlesztők képeket, animációkat, mozgóképeket és hangokat ágyazhatnak be vagy tehetnek elérhetővé a weblapokon.

A HTML a böngészőkkel együtt fejlődött, a „hivatalos” HTML-változatokat a W3C hagyta jóvá illetve készítette el. A böngészők sokfélesége és a cégek saját HTML módosításai kompatibilitási problémákhoz vezettek. A modern webböngészők (mint a Firefox, a Mozilla, az Opera, a Safari, a Google Chrome stb.) már pontosabban támogatják a HTML és XHTML szabványokat (a HTML 4.01-gyel kezdődően), melyek figyelembe vételével a weblapoknak azonosan kell megjelenniük minden ilyen böngészőben. Az Internet Explorer jelenleg még nem támogatja tökéletesen a HTML 4.01 és XHTML 1.x szabványokat. Manapság sok weboldal már valamilyen könnyen kezelhető, azonnali eredményt szolgáltató, úgynevezett ALAKHŰ (WYSIWYG) szerkesztővel készült, mint amilyen a Macromedia Dreamweaver vagy Microsoft Frontpage, bár utóbbi nem szabványos HTML kódot készít, akadályozva a W3C munkáját a szabványok további fejlesztése, különösen az XHTML és a CSS stíluslapok tekintetében.

Egyes böngészők olyan népszerű protokollokat is támogatnak, melyek nem kapcsolódnak szorosan a böngészéshez. Ilyenek például az IRC (Internet Real Chat) csevegő, a Usenet hírolvasó vagy a levelező kliens, melyekhez NNTP (Network News Trensfer Protocol), SMTP (Simple Mail Transfer Protocol), IMAP (Internet Messaging Transfer Protocol) és POP (Post Office Protocol) protokollokat használják.

A modern böngészők kiegészítők révén képesek különböző programozási nyelveken írt kódokat, végrehajtani, ezáltal látványosabb tartalmat jeleníthetnek meg; ilyenek a Flash, Java stb. nyelvek.

A legtöbb böngésző támogat valamilyen szkript nyelvet (JavaScript, VBScript, JScript, stb.). Talán a JavaScript az egyik legelterjedtebb, de ennek is több változata létezik, és a megvalósítása függ a használt operációs rendszertől, böngészőtől, sőt böngészőverziótól.

**Google Chrome:**

Alapértelmezés szerint a Chrome felhasználói felületén megtalálhatók a vissza, előre, frissítés, kezdőoldal, könyvjelző, ugrás, illetve mégse gombok. A kezdőoldal gombja beállítható, hogy az „Új lap” oldalra vigyen vagy más, előre beállított oldalra, de akár el is rejthető.

A böngészőfülek a Chrome felhasználói felületének legfontosabb elemei, ezért az ablak tetejére (címsor, title bar) helyezték, nem a vezérlők alá. Ez a változtatás szembemegy több más böngésző kialakításával. A böngészőfülek (tartalmukkal együtt) könnyen mozgathatók fogd és vidd módszerrel a böngészőablakok között. Minden fülnek saját vezérlői vannak, az Omnibox-ot is beleértve.

Az Omnibox a minden böngészőfül felső részén megtalálható címsáv-féleség, ami a megszokott címsáv és a keresőmező funkcionalitását kombinálja. Ha a felhasználó olyan oldal URL-jét üti be, amin korábban már végzett keresést, a Tab leütésével közvetlenül az Omniboxból kereshet azon az oldalon. Ha gépelni kezd az Omniboxba, a Chrome automatikusan javaslatokat fog kiírni; ezek lehetnek korábban meglátogatott oldalak (az oldal URL-je vagy akár tartalma alapján), népszerű webhelyek (nem feltétlenül olyanok, amiken ténylegesen járt már a felhasználó – a Google Suggest alapján javasol) és népszerű keresések. A Chrome a gyakran látogatott oldalak URL-jeit is automatikusan kiegészíti. Ha a felhasználó több kulcsszót ír az Omniboxba és Entert üt, a Chrome keresést fog végrehajtani az alapértelmezett keresővel.

Amikor a Google Chrome nincs maximalizálva, a „fülsáv” közvetlenül az ablak címsora alatt jelenik meg. Maximalizált állapotban a fülek egybemosódnak a címsáv tetejével. Más böngészőkhöz hasonlóan a Chrome-nak is van teljes képernyős üzemmódja, amikor a felhasználói felület nem létfontosságú részein (a „chrome-on”, csicsán) kívül az operációs rendszer felületét is elrejti.

A Chrome egyik jellemző új képessége az „Új lap” oldal, ami lecserélheti a böngésző kezdőlapját, és új fül (vagy új ablak) létrehozásakor jelenik meg. A korai Chrome-változatokban ez a lap a kilenc leggyakrabban látogatott weboldal kisméretű képét (indexkép) mutatta, valamint a gyakori kereséseket, az újonnan felvett könyvjelzőket, és a frissen bezárt füleket; hasonlóan, mint a Google Toolbarral kibővített Internet Explorer és Firefox esetén, vagy mint az Opera Speed Dial funkciójánál. A Google Chrome 2.0-ban, az „Új lap” oldalon lehetővé tették a nem kívánt weboldalképek eltüntetését. A Chrome 3.0-tól kezdve az „Új lap” oldal a 8 leggyakrabban látogatott weboldalt jeleníti meg. Az indexképek átrendezhetők, rögzíthetők vagy eltüntethetők. Képek helyett szöveges listát is lehet választani. Egy „Nemrégiben bezárt” sáv is megjelent, a közelmúltban bezárt oldalak listájával és egy „Tippek” rész, ahol a böngésző használatával kapcsolatos tippek-trükkök jelennek meg.

2011 márciusában a Google lecserélte a projekt kezdetétől használt, 3D-s logót egy új, egyszerűbb logóra. A Google designere, Steve Rura így indokolta a változtatást: „Mivel a Chrome lényege, hogy a lehető legkönnyebben és zavaró tényezőktől mentesen nyújtsa a webezés élményét, felfrissítettük a Chrome ikonját, hogy jobban kifejezze ezt az érzést. Az egyszerűbb ikon megtestesíti a Chrome szellemiségét – a web könnyebbé, gyorsabbá és elérhetőbbé tételét mindenki számára.”

A Chrome lehetővé teszi a böngészőben webes alkalmazást megnyitó parancsikonok létrehozását az asztalon. Ilyen módon indítva, a Chrome az ablak címsorától eltekintve a szokásos felhasználói felület nélkül nyílik meg, „hogy a felhasználó minél kevésbé legyen akadályozva abban, amit csinálni próbál”. Így a böngésző a többi helyi alkalmazás mellett futhat, hasonlóan a Mozilla Prismhez vagy a Fluidhoz.

A funkció továbbfejlesztése a 9-es stabil verzióval együtt 2011 februárjában debütáló Chrome Web Store, ahonnan webes alkalmazásokat lehet beszerezni és letölteni.

**Mozilla Firefox:**

A Firefox többplatformos alkalmazás, a Microsoft Windows több változatán, OS X-en, Linuxon és számos más Unix-alapú operációs rendszeren is fut. Legutóbbi stabil kiadása, a 26.0-s verzió 2013. december 10-én jelent meg. Forráskódja szabad, a GPL/LGPL/MPL hármas licenc alatt érhető el. A Mozilla Corporation által kiadott, Firefox 3.0.6-os verziónál régebbi telepítőkészletek telepítése előtt el kell fogadni a Mozilla Corporation végfelhasználói licencszerződését (EULA). A Firefox 3.0.6 és az újabb verziók telepítőkészletei az MPL alatt jelennek meg. A Firefox név és logó, egyéb nevek és logók mellett, védjegyoltalmat élvez; ezek felhasználása korlátozott, nem esnek a fenti licencek hatálya alá.

A Firefoxba épített szolgáltatások közé tartozik a füles böngészés, a helyesírás-ellenőrző, az azonnali keresés, az élő könyvjelzők kezelése, a beépített letöltéskezelő és egy keresőrendszer, ami a felhasználó által kiválasztott keresőmotort használja.

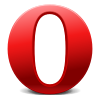
A fejlesztők célja egy olyan böngésző készítése volt, ami „szörfözik a weben” és a „lehető legjobb böngészési élményt nyújtja a lehető legnagyobb embertömeg számára”.

A felhasználók kiegészítőkkel és témákkal alakíthatják át a Firefoxukat saját ízlésük szerint. A Mozilla egy külön tárhelyt üzemeltet ezek tárolására az addons.mozilla.org címen, ahol 2009 júliusában már több mint 5000 kiegészítő volt található. A kiegészítők ingyenesek.

A legnépszerűbb kiegészítők között van az Adblock Plus (a reklámokat tiltja le), a Video DownloadHelper (internetes videók letöltését és konvertálását segítő eszköz), a NoScript (JavaScript-letiltó eszköz), a FlashGot (letöltéskezelő, amely külső programokkal működik együtt), a DownThemAll! (letöltéskezelő), a Tab Mix Plus (amellyel további beállítások végezhetők a füleken), a Cooliris (a támogatott oldalak képeiből 3D-s falat készít), a FoxyTunes (médialejátszó eszköz), a StumbleUpon (a felhasználó érdeklődésének megfelelő oldalakat javasol), a Foxmarks Bookmark Synchronizer (könyvjelző-szinkronizáló) és a Web Developer (webfejlesztői eszköztár).

A Mozilla Firefox számos internetes szabványt támogat, például a HTML-t, az XML-t, az XHTML-t, az SVG 1.1-et (részben), a CSS-t (kiegészítők segítségével), az ECMAScriptet (JavaScript), a DOM-ot, a MathML-t, a DTD-t, az XSLT-t, az XPath-ot és az alfa-csatornával rendelkező (áttetsző) PNG-képeket is. Olyan szabványtervezeteket is támogat, amiket a WHATWG készített, például a klinesoldali adattárolást, és a vásznat.

A Firefox implementálta a Google „safebrowsing” nevezetű kereskedelmi protokollját, ami nem nyílt szabvány. Ez viszont ellentmondhat a Mozilla Kiáltványban (Mozilla Manifesto) leírtaknak.

**Opera:**

Az Opera egy ingyenes, igen kis méretű, több platformon használható internetes tevékenységeket összefogó program: elsősorban egy webböngésző, de ezen kívül tartalmaz beépített levelező klienst, hírolvasó klienst, címjegyzéket, IRC alapú csevegő klienst, RSS és Atom hírolvasót, minialkalmazásokat (widgeteket), valamint hagyományos és BitTorrent alapú letöltésvezérlőt is. Ezek a kiegészítő funkciók mindaddig rejtve maradnak az Operában, amíg a felhasználó nem kezdi el használni valamelyiket.

Az Operát az Opera Software, egy Oslóban jelenleg is működő norvég cég készíti. A böngésző korábban a Presto elnevezésű motorját (rendering engine) használta, később azonban a Google által fejlesztett Blink motorra váltott. Az Opera vezető pozíciót szerzett az okostelefonok és PDA készülékek böngészői között a kis képernyős megjelenítés technológiájának köszönhetően. Az Operát és az Opera Minit C++ nyelven írták.

Az Opera zárt forráskódú, de ingyenesen letölthető és használható program. A jelentősebb böngészőkhöz mérten igen kis méretű – annak ellenére, hogy funkcionalitása sokkal szélesebb körű, magas szinten követi a webes szabványokat. Ezt az irányvonalat jól jellemzi, hogy több alkalmazottja tagja a W3C konzorciumnak. A böngésző biztonság tekintetében is kiemelkedő eredményeket ért el, a többi böngészőhöz képest jóval kevesebb biztonsági hibát fedeztek fel benne, és ezeket is nagyon gyorsan kijavították.

Az Operának léteznek speciális verziói mobiltelefonokra és más eszközökre is. A böngésző mobil változata, az Opera Mini minden Java-képes mobiltelefonon elérhetővé teszi a böngészést, használata ingyenes.

Az Opera feldolgozási sebesség tekintetében a legjobbak között van, több teszt szerint is a leggyorsabb. Az Opera korábban is a leggyorsabb böngészők közé tartozott, azonban a Sunspider, a Celtic Cane és a V8 2009. márciusi eredményei szerint a Google Chrome, a Safari és részben a Mozilla Firefox is megelőzte, csupán az Internet Explorer maradt le mögötte. Az Opera 10.50 az előző verzióhoz képest nyolcszor gyorsabb, ezzel újra az élvonalba került.

Az Opera a böngészés gyorsítása érdekében folyamatosan tárolja a gyorsítótárban a megnyitott weblapokat, így visszalépéskor az új lap azonnal megjelenik, ráadásul az űrlapokba írt információk sem vesznek el. A véletlenül bezárt lapok és ablakok egy kattintással újranyithatók a képernyő jobb felső sarkában található Bezárt fülek gombra kattintva.

Az Opera volt az első böngésző, amely képes volt a zavaró felugró ablakokat letiltani. Jelenleg az alapbeállítás szerint az Opera minden olyan felugró ablakot blokkol, amelyet nem a felhasználó nyitott meg.

Az Androidra és iOS-re fejlesztett Opera böngésző Ethereum-kompatibilis beépített kriptovaluta tárcával rendelkezik. A böngésző-alkalmazással az Ether valután kívül hagyományos ERC-20 tokeneket, és úgynevezett NFT (ERC-721 szabványú) nem helyettesíthető tokeneket is tárolni lehet. Ilyen tokenekre példa a CryptoKitties digitális macskák, illetve más, tulajdonjogot megtestesítő virtuális eszközök.

**Visual Studio Code program részletes bemutatása felhasználói környezetben:**

A Visual Studio Code, más néven VS Code a Microsoft által az Electron keretrendszerrel készített forráskód-szerkesztő program, Windows, Linux és macOS operációs rendszerekhez, amelynek jellemzői közé tartozik a hibakeresés támogatása, a szintaxis kiemelése, az intelligens kódkiegészítés, a snippetek, a kód refaktorálása és a beágyazott Git. A felhasználók megváltoztathatják a témát, a billentyűparancsokat, a beállításokat, és telepíthetnek bővítményeket, amelyek további funkciókat adnak hozzá.

A Stack Overflow 2021-es fejlesztői felmérésében a Visual Studio Code a legnépszerűbb fejlesztői környezetet biztosító eszköznek bizonyult 82 000 válaszadó körében, 70%-uk jelezte, hogy használja.

**Jellemzői a Visual Studio Code-nak:**

A Visual Studio Code egy forráskód-szerkesztő, amely számos programozási nyelvvel használható, többek között a C#, Java, JavaScript, Go, Node.js, Python, C++, C, Rust és Fortran nyelvekkel. Alapja az Electron keretrendszer, amelyet a Blink elrendezési motoron futó Node.js webes alkalmazások fejlesztésére használnak. Ugyanazt a szerkesztő komponenst (kódnevén "Monaco") használja, amelyet az Azure DevOps (korábbi nevén Visual Studio Online és Visual Studio Team Services).

Alapszintű támogatást tartalmaz a legtöbb elterjedt programozási nyelvhez. Ez az alaptámogatás magában foglalja a szintaxis kiemelést, a zárójelek illesztését, a kód összehajtását és a konfigurálható snippeteket. IntelliSense-t is tartalmaz a JavaScript, TypeScript, JSON, CSS és HTML számára, valamint hibakeresési támogatást a Node.js számára. További nyelvek támogatását a VS Code Marketplace-en szabadon elérhető bővítményekkel lehet biztosítani.

A projektrendszer helyett lehetővé teszi a felhasználók számára, hogy egy vagy több könyvtárat nyissanak meg, amelyeket aztán munkaterületekre menthetnek a későbbi újra felhasználás érdekében. Ez lehetővé teszi, hogy nyelv-agnosztikus kódszerkesztőként működjön bármilyen nyelvhez. Számos programozási nyelvet és nyelvenként eltérő funkciókészletet támogat. A nem kívánt fájlok és mappák a beállításokon keresztül kizárhatók a projektfából. Számos funkciója nem a menükön vagy a felhasználói felületen keresztül érhető el, hanem a parancspalettán keresztül.

Bővítményekkel bővíthető a program, amelyek egy központi tárolón keresztül érhetők el. Ez magában foglalja a szerkesztő és a nyelvtámogatás bővítéseit. Figyelemre méltó funkció a bővítmények létrehozásának lehetősége, amelyek új nyelvek, témák, hibakeresők, időutazó hibakeresők támogatását adják hozzá, statikus kódelemzést végeznek, és a Language Server Protocol segítségével kódlintereket adnak hozzá.

A forrásvezérlés a Visual Studio Code beépített funkciója. A menüsoron belül van egy külön fül, ahol a felhasználók elérhetik a verziókezelési beállításokat és megtekinthetik az aktuális projektben végrehajtott változtatásokat. A funkció használatához a telepített programot bármely támogatott verziókezelő rendszerrel (Git, Apache Subversion, Perforce stb.) össze kell kapcsolni. Ez lehetővé teszi a felhasználók számára, hogy tárolókat hozzanak létre, valamint hogy közvetlenül a Visual Studio Code programból push- és pull-kérelmeket tegyenek.

A Visual Studio Code több kiterjesztést is tartalmaz az FTP-hez, így a szoftver ingyenes alternatívaként használható a webes fejlesztéshez. A kód szinkronizálható a szerkesztő és a szerver között, anélkül, hogy bármilyen extra szoftvert kellene letölteni.

Magy a program lehetővé teszi a felhasználók számára, hogy beállítsák a kódoldalt, ahová az aktív dokumentumot mentik, az újsor karaktert és az aktív dokumentum programozási nyelvét. Ez lehetővé teszi, hogy bármilyen platformon, bármilyen nyelvterületen és bármilyen programozási nyelvhez használható legyen.

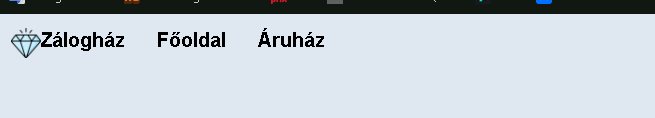
A program összegyűjti a használati adatokat és elküldi azokat a fejlesztő cégének, vagyis a Microsoftnak, bár ez kikapcsolható, az alkalmazás nyílt forráskódú jellegéből adódóan a telemetriakód a nyilvánosság számára is hozzáférhető, akik pontosan láthatják, hogy mit gyűjtenek.

**Felmérés a Visual Studio Code-ról:**

A Stack Overflow 2016-os fejlesztői felmérésében a Visual Studio Code a 13. helyen szerepelt a legnépszerűbb fejlesztői eszközök között, a 47 000 válaszadó mindössze 7%-a használta. Két évvel később azonban a Visual Studio Code az első helyre került, a 75 000 válaszadó 35%-a használta. A 2019-es fejlesztői felmérésben szintén a Visual Studio Code került az első helyre, a 87 000 válaszadó 50%-a használta. A 2021-es fejlesztői felmérésben a Visual Studio Code továbbra is az első helyen állt, a 82 000 válaszadó 70%-a használta, a 2022-es felmérésben pedig a 71 010 válaszadó 74,48%-a. A 2022-es felmérésben a 71 010 válaszadó 74,48%-a használta.

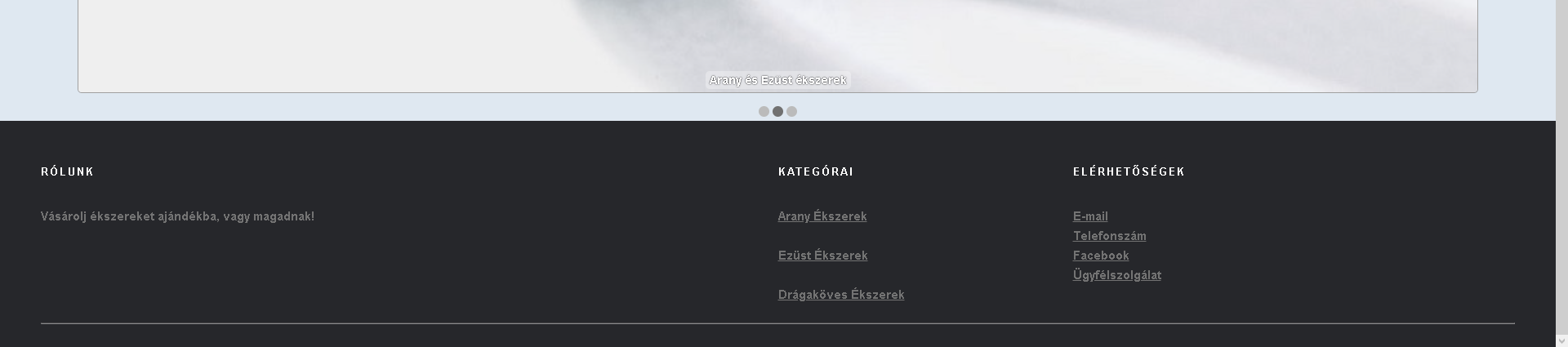
### Weboldalról:

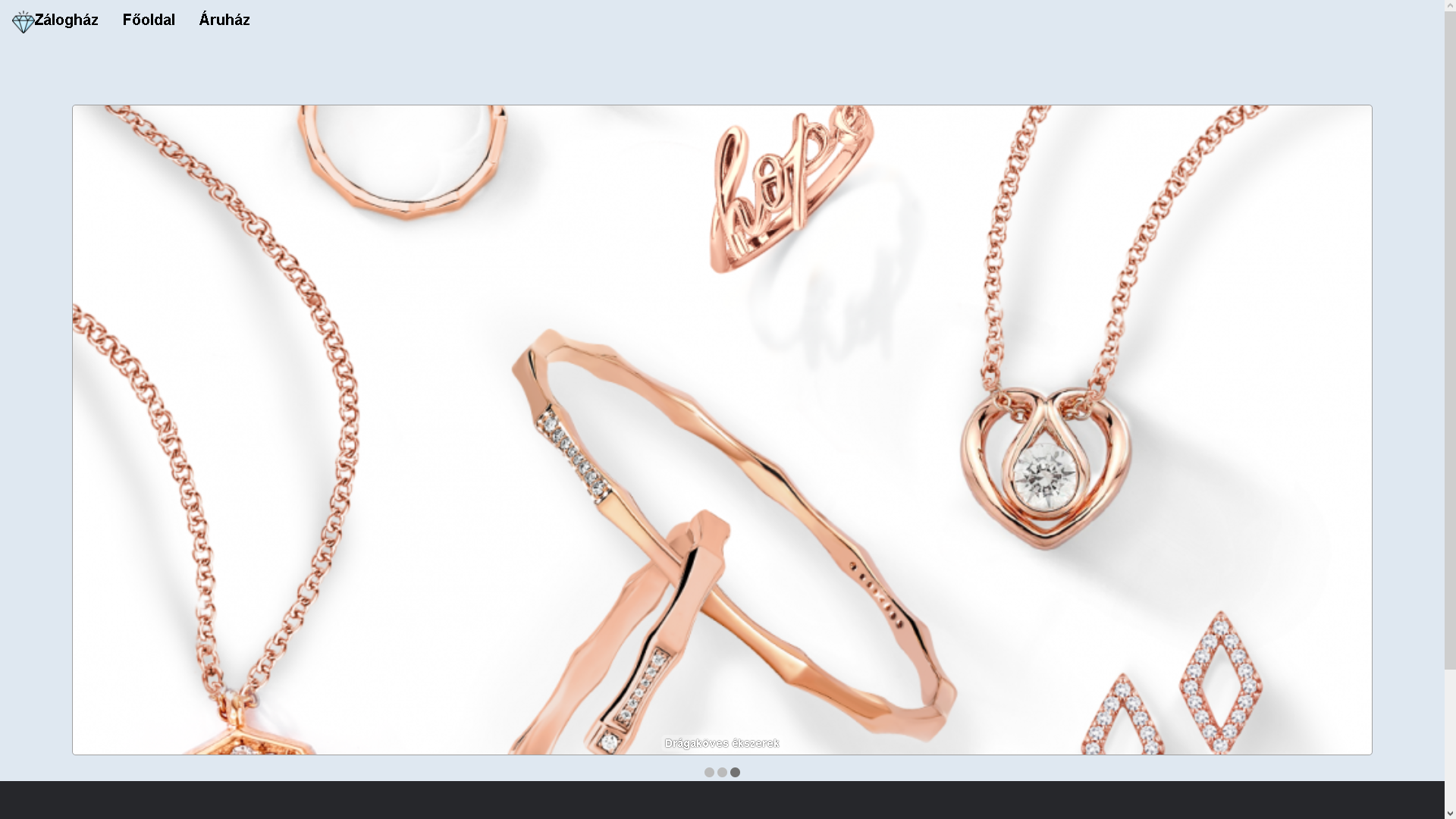
Ha az oldal tetejét megtekintjük, egy menüsort találunk. Ennek a felületén kiválaszthatjuk, hogy a weboldal melyik részére szeretnénk eljutni. Jelenleg kettő lehetőség közül lehet választani: a Főoldal és az Áruház gomb. A jövőben tervezünk hozzá készíteni egy regisztráció és bejelentkező részt, ami majd szintén kiválasztható lesz a menüsoron.



**Főoldal:**

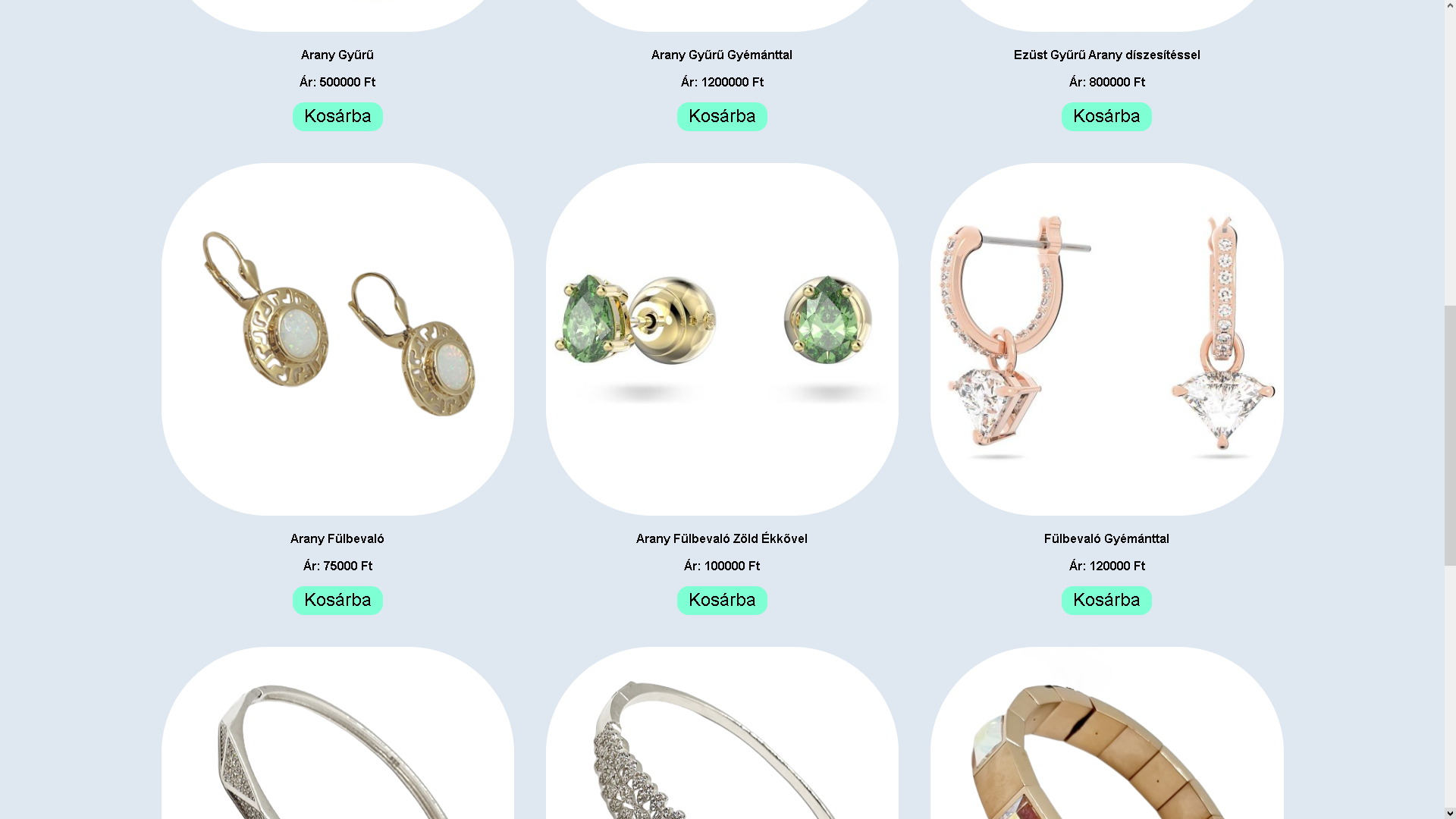
Ha a Főoldal gombra kattintunk, eljuthatunk a weboldal kezdőlapjára. A Főoldalon található egy Slideshow, avagy „diavetítés”. Ezen belül megtekinthető néhány termék, illetve feliratokon olvasható, hogy milyen fajta kategóriákat készítettünk. Ha ennek a jobb vagy bal oldalán lévő nyilakra kattintunk, vagy pedig a kép alján található karikákra, akkor válthatunk a felsorolás részei között. A felület alján továbbá egy „footer”, vagyis egy lábléc található, amin megtekinthetjük az elérhetőségeket, illetve szintén a kategóriákat. Ha ezeken belül bármelyik lehetőségre rákattintunk, akkor az el fog juttatni minket egy másik lapra: Kategóriákon belüli lehetőségek az Áruház oldalra, az Elérhetőségeken belül pedig eljuttat egy olyan helyre, ahol a webáruház elérhetőségeit tekinthetjük meg.





**Áruház:**

Ha az oldal tetején található menüsoron az Áruház gombot kiválasztjuk, akkor eljuthatunk a weboldal áruház részére. Itt jelenleg 12 termék közül lehet választani, hogy melyiket szeretnénk megrendelni. A választható lehetőségek mindegyikén van egy „Kosárba” gomb. Ezzel a lehetőséggel tudja a felhasználó kiválasztani, hogy melyiket szeretné megrendelni. Ha lejjebb tekerünk az oldalon, ott mutatja a megvásárolni kívánt, kosárba helyezett termékeket. Miután már egy terméket kiválasztott a felhasználó, azt még egyszer már nem lehetséges belehelyezni a kosárba. Mindazonáltal, ha mégis többet szeretne vásárolni egy termékből, erre is van lehetőség. A termék kosárba helyezése után a vevő meg tudja változtatni a termék mennyiségét a kosárban lévő termék melletti számláló átírásával, vagy a szám melletti felfelé és lefelé mutató nyilakkal. Miután kiválasztott a felhasználó minden terméket, amit meg szeretne vásárolni, az oldal alján lévő Vásárlás gombra kattintva befejezheti a folyamatot.





## Források

https://hu.wikipedia.org/wiki/Visual\_Studio\_Code

https://hu.wikipedia.org/wiki/Google\_Chrome

https://hu.wikipedia.org/wiki/Mozilla\_Firefox

https://hu.wikipedia.org/wiki/Opera\_(webböngésző)