Záródolgozat

**Milyen területen írtam a záródolgozatomat?**

Programozással már hamarabb megismerkedtem, már közel 5 éve. Sok programot írtam már, sajnos az esetek jelentős részében nem a saját akaratom és érdeklősédem miatt, hanem beadandó iskolai feladat miatt, de arról nincs kétség, hogy ne élveztem volna egyiket sem. Mindig volt 1 – 2 feladatrész aminél hosszabb ideig kellett gondolkodnom milyen jó megoldások léteznek a problémára, és ez az, ami megfogott az egészben, a problémák megoldása és azok továbbfejlesztése. Ezzel szemben a webfejlesztést csupán alig 2 éve ismerem, és ezen belül is kevesebb mint 1 éve kezdtem el komolyabban foglalkozni vele. Elsőre a webfejlesztés nem érdekelt, mert az elején csak html és css volt, amivel, habár szép és érdekes dolgokat tudtam már akkor létrehozni, nem éreztem azt, amit a programozás tud nyújtani. Ezután ismerkedtem meg a weboldalak újabb, szinte legfontosabb elemét alkotó részével, a javascript-tel. Viszont sokáig ez sem keltette fel az érdeklősédemet. Már tudtam, hogy a weboldalról bevitt adatokkal tudok dolgozni, tudok a felhasználóval kommunikálni, de itt is csak a felszínt kapargattam, ami nem kötött le. De miután jobban el kezdtem beleásni magamat a témába, rájöttem mennyi potencia és lehetőség van egy webes projekt elkészítésében. Ezek után a webes téma választása eléggé meggyőzőnek és ígéretesnek tűnt, de ez még nem volt elég ahhoz, hogy meghozzam a végső döntésemet.

**Web és Mi?**

Ahogy már mondtam, egy webes dolgozat választása jó döntésnek tűnt, de önmagában nem tudta felülmúlni a játék iránti érdeklésemet. Biztos sokszor hallottuk már a ’És te mégis miért érdeklődsz annyira az informatika vagy esetleg a programozás felé?’ kérdésre azt a választ, hogy ’Sokat játszok játékokkal, így jó öltetnek tűnt.’ vagy hogy ’Szeretnék videójátékokat csinálni, és esetleg ebből jelentős pénzre szert tenni majd a jövőben.’. Nos az én gyerekkori nézeteim egy részét a játékkészítés fedte le amikor valamit ezt kérdést tette fel nekem. De szerintem ez majdnem minden leendő programozó, szoftverfejlesztő álma volt, hogy egyszer játék készítő/fejlesztő legyen, néhánynak biztosan sikerült is. Sokáig ez volt a fejembe, hogyha megszerzem ezt a szakmát, akkor én valamilyen játék projektben szeretnék részt venni. Ez a vágy azonban a jelenlegi projekt kiválasztásában nem segített. Mivel az egyik téma a régi, a másik pedig az új érdeklődési körömet fedte le, és emellett azt hitem a kettő terület nagyon messze fekszik egymástól, sokat vacilláltam mégis melyiket válasszam. Nem sok idő elmultával feltettem magamnak a kérdést, mégis miért kéne választanom a kettő közül csak az egyiket? Egy csomó játék található meg az interneten, s már akkor is volt ezekből amikor még el sem elkezdtem érdeklődést mutatni a számítástechnika és az informatika iránt. Manapság pedig már olyan játékokat tudnak az emberek létrehozni az interneten, amik többjátékosak, mind online és offline, emellett meglepően részletes grafikával rendelkeznek. Mért ne lehetne egy olyan weboldalt készíteni, ami önmagában egy játék? Így jutottam arra a döntésre, hogy nem választok, mindkettő ki fog venni valamennyi részt a záródolgozatból, s készítek egy olyan projektet, ami nem csak egy weboldal, de egy játék is egyben, így mindkettő területből tudok tovább fejlődni.

**Milyen programokat, keretrendszereket használtam?**

**Visual Studio Code**

A Visual Studio Code egy világszerte ismert és használt program, amit a Microsoft fejlesztett és adott ki 2015-ben. A szoftver ingyenesen megszerezhető a weboldalukon, emellett több rendszer is támogatja, így akár Windowsra, macOS-re, vagy Linuxra is telepíthető. A program nagyon hamar megnyerte a szoftverfejlesztők tetszését, hisz a program meglehetősen sok személyre szabható elemmel rendelkezik. Alapvetően az alkalmazás támogatja a JS, HTML, CSS és PHP kiterjesztésű fájlok olvasását és formázását, de néhány kiegészítő letöltésével több 10 nyelvhez kaphatunk még támogatást, esetleg kódjavaslatokat is. Az ilyen bővítményekkel a program képes kiértékelni a kódunkat még mielőtt lefuttatnánk, a hibáinkat piros vagy egyéb színnel aláhúzva jelzi a hiba forrását, még képes javaslatot is tenni hogyan lehetne a problémát orvosolni. Az ilyen kiegészítők sokat tudnak segíteni, hogy a kód olvasható és átlátható legyen, ezen felül segíti a még tanulók fejlődését a kód megértésében és a hibák kijavításában is. Emellett az alkalmazás alap funkciója a verziókezelés lehetősége, ami mind kisebb mind nagyobb alkalmazások esetében jól tud jönni. Miután összekötöttük az adott mappát egy verziókezelést támogató szolgáltatással, a program képes felismerni, ha ez megtörtént és bármely változás, amely az adott mappán belül történik felkínálásra kerül az alkalmazásban a felhasználó számára. Alapesetben a program csak a GIT verzió kezelőt ismeri, de ez itt is bővítményekkel kiegészíthető. Maga az alkalmazás kinézete is számos részben megváltoztatható. Lehetőség van a színek változtatására, az alkalmazáson belüli elemek áthelyezésére, még néhány ikon is megváltoztatható. Összességében az alkalmazás nagyon sok személyre szabási lehetőséget az a felhasználó számára, de változtatások nélkül is egy jól felépített, gyors, hatékony és szép alkalmazást kapunk, így nem csoda, hogy a program ilyen sok embernek nyerte el tetszését.

**Node.JS**

A Node.js egy többplatformos, nyílt forráskódú programozási környezet, amely Windows, Linux, Unix, macOS és egyéb rendszereken is futtatható. Az alkalmazás egy háttérben futó JavaScript futtatókörnyezet és keretrendszer, és az ide beérkező JavaScript-kódot webböngészőn kívül hajtja végre. A Node.js lehetővé teszi a fejlesztők számára a JavaScript használatát parancssori eszközök írásához és szerveroldali szkriptek készítéséhez. A JavaScript szerveren való futtatásának képességét gyakran használják dinamikus weboldaltartalom generálására, mielőtt az oldalt elküldené a felhasználó webböngészőjének. A program képes egyetlen programozási nyelv köré gyűjteni a webalkalmazások fejlesztését, szemben azzal, hogy a szerver- és kliensoldali programozáshoz különböző nyelveket használ.

A Node.js lehetővé teszi webszerverek és hálózati eszközök létrehozását JavaScript és a különféle alapvető funkciókat kezelő „modulok” használatával. Modulokat biztosítanak a fájlrendszer kezelésére, hálózatkezeléshez (ilyen a DNS vagy a HTTP/HTTPS), bináris adatokhoz, kriptográfiai funkciókhoz, adatfolyamok és egyéb alapvető funkciókat. A Node.js moduljai olyan API-t használnak, amely csökkenti a kiszolgálóalkalmazások írásának bonyolultságát, így akár kezdőként is hamar meg lehet érteni a különböző folyamatok működését.

A Node.js-t elsősorban hálózati programok, például webszerverek készítésére használják. A legjelentősebb különbség a Node.js és a PHP között az, hogy a PHP-ban a legtöbb függvény a befejezésig blokkol (a parancsok csak az előző parancs befejezése után fut le), míg a Node.js függvények nem blokkolnak (a parancsok párhuzamosan képesek futni, ezzel felgyorsítva a szerveroldali folyamatokat).

Ahogy már említettem, a Node.js rendelkezik úgynevezett modulokkal, melyek előre megírt függvények/osztályok. Hasonlóan, mint egy JavaScript könyvtárhoz, az adott modul letöltése után be tudjuk emelni a projektünkbe és ott szabadon használhatjuk. Ebbe a projektbe is használatra került néhány ilyen modul, melyek megkönnyítették a feladatom egy részét.

**Node.js: Socket.io**