Miskolci Egyetem

Gépészmérnöki és Informatikai Kar

Általános Informatikai Intézeti Tanszék

A logo of a university

Description automatically generated

**Informtikai eszköznyilvántartó**

**Szakdolgozat**

**Készítette:**

**Név:** Pázmán András

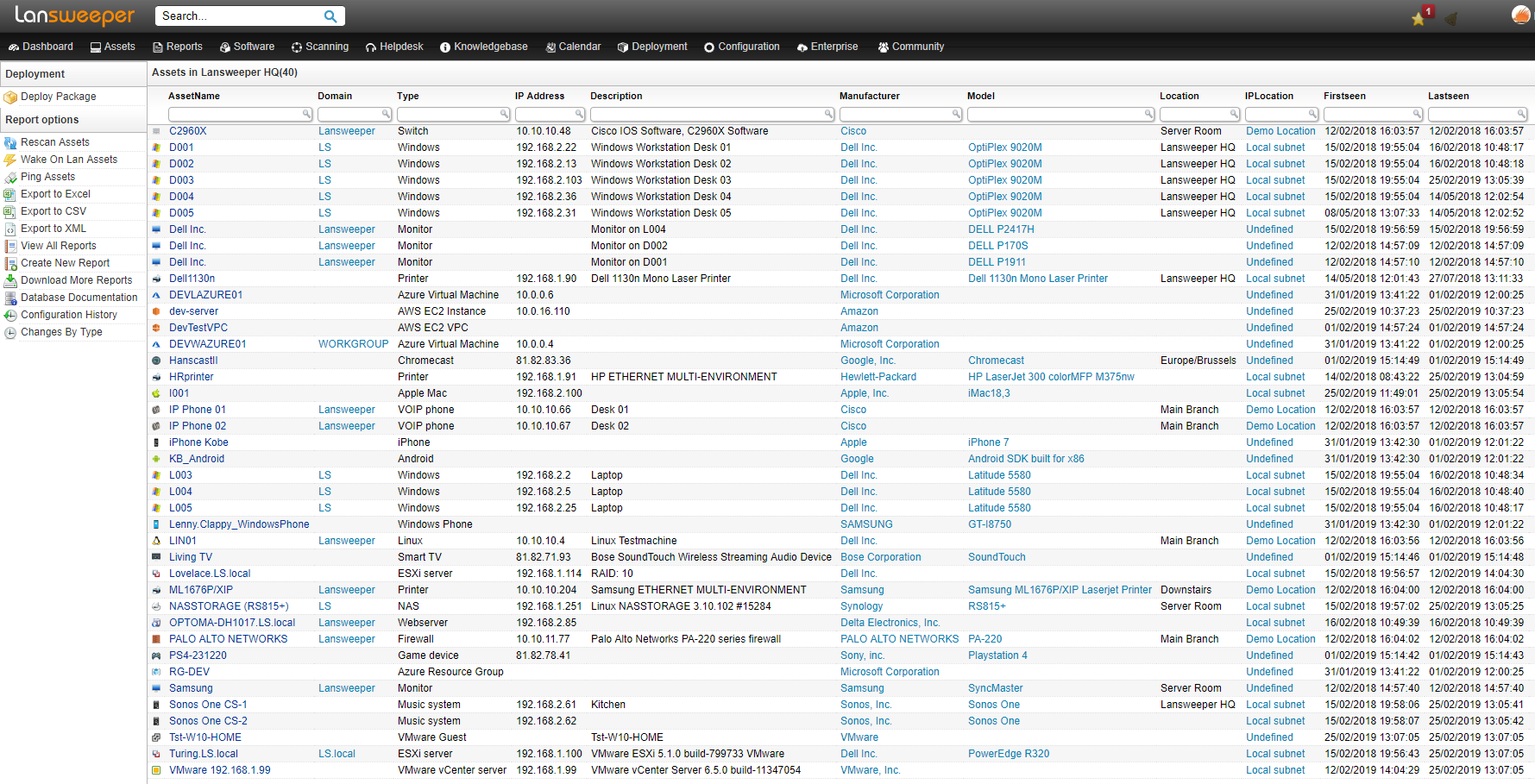
**Neptunkód:** F58KQ8

**Szak:** Mérnök Informatiku BSc

Korszerű web technológiák szakirány

1. **Bevezetés**

A nyílvántartó rendszerek fontossága vitathatatlan napjainkban, mivel a folyamatosan bővülő infrastruktúrákhoz elengedetlen egy adminisztrációs felület amely számon tart akár projekteket, felhasználókat, egyéb eszközöket.   
Ezeknek a rendszereknek az alkalmazása és fenntartása elősegíti az adatok hatékony tárolását és elemzését számos területen, ide értve akár az oktatást, egészségügyet, gyártást.

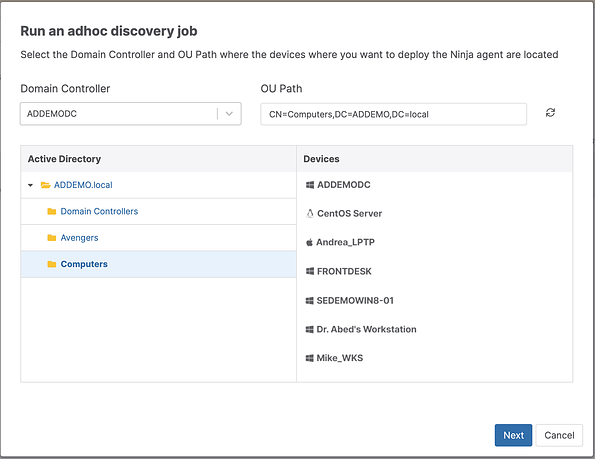
Az általam választott téma egy informatikai eszközöket nyílvántartó alkalmazás elkészítése kis és közép vállakozások számára. A projekt ötlete egy diákmunkámból ered, ahol is a dolgozók által használt eszközök (laptopok, telefonok, perifériák) vezetése még mindig excel táblázatban történt. Léteztek már korábban is hasonló nyílvántartó rendszerek, pl lansweeper ( <https://www.lansweeper.com/> ) vagy ninjaOne ( <https://www.ninjaone.com/>) de a legtöbb IT rendszer inkább az eszközök monitorzásával foglalkozik és nem pedig a használóikat manageli.

[  
<https://en.wikipedia.org/wiki/Lansweeper>

A Lansweeper egy informatikai felderítési és leltári platform, amely betekintést nyújt a felhasználók, eszközök és szoftverek állapotába az informatikai környezetekben. Ez a platform leltározza a csatlakoztatott IT-eszközöket, lehetővé téve a szervezetek számára, hogy központilag kezeljék IT-infrastruktúrájukat. A Lansweeper automatizált folyamatai azonosítják és összeállítják a csatlakoztatott eszközök listáját, beleértve a számítógépeket, útválasztókat, szervereket és nyomtatókat. Eszközspecifikus információkat tartalmaz a telepített szoftverekről, az alkalmazott frissítésekről és javításokról, valamint a felhasználói adatokról. ]

NinjaOne:

<https://www.ninjaone.com/rmm/it-asset-management/>



A NinjaOne leltár jelentései megkönnyítik a készletek, a használat, valamint a hardver, a szoftver és az előfizetések állapotának nyomon követését a NinjaOne vagyonkezelési jelentéseivel.

[ NinjaOne leemzése ]

Továbbá ezek a szoftverek nehezen személyre szabhatóak, tartalmaznak olyan funkciót is ami számunkra nem feltétlen szülséges vagy esetleg hianyoznak olyanok amire szükség lenne . A karbantarthatóság és költséghatékonyság szempontjából pedig előnyösebb hogy saját alkalamzásunkat házon belül kezeljük nem pedig előfizetünk egy harmadik félnek.

A cél tehát egy olyan alkalmazás elkészítése amely a felhasználókhoz rendeli az általuk használt eszközöket. Rögzíti a változásokat ( eszközcsere, tulajdonosváltáltás eszközök között, raktárba vétel ) amelyek az admin felületről követhető. Ezen felül képes riportot készíteni és azt fálj-ba menteni.

3. Fejlesztéshez választott technológiák

3.1. Angular <https://angular.io/docs>

Az Angular egy nyílt forráskódú JavaScript-keretrendszer, amelyet TypeScriptben írtak. A Google karbantartja, és elsődleges célja egyoldalas alkalmazások fejlesztése. Keretrendszerként az Angular egyértelmű előnyökkel rendelkezik, miközben szabványos struktúrát biztosít a fejlesztők számára. Lehetővé teszi a felhasználók számára, hogy nagyméretű alkalmazásokat hozzanak létre karbantartható módon.

A keretrendszerek általában növelik a webfejlesztés hatékonyságát és teljesítményét azáltal, hogy konzisztens struktúrát biztosítanak, így a fejlesztőknek nem kell a semmiből kell újraépíteniük a kódot. A keretrendszerek időt takarítanak meg, és számos extra funkciót kínálnak a fejlesztőknek, amelyek további erőfeszítések nélkül adhatók hozzá a szoftverhez.

Az Angular jellemzői

1. Dokumentumobjektum modell

A DOM (Document Object Model) egy XML vagy HTML dokumentumot fastruktúraként kezel, amelyben minden csomópont a dokumentum egy részét képviseli.

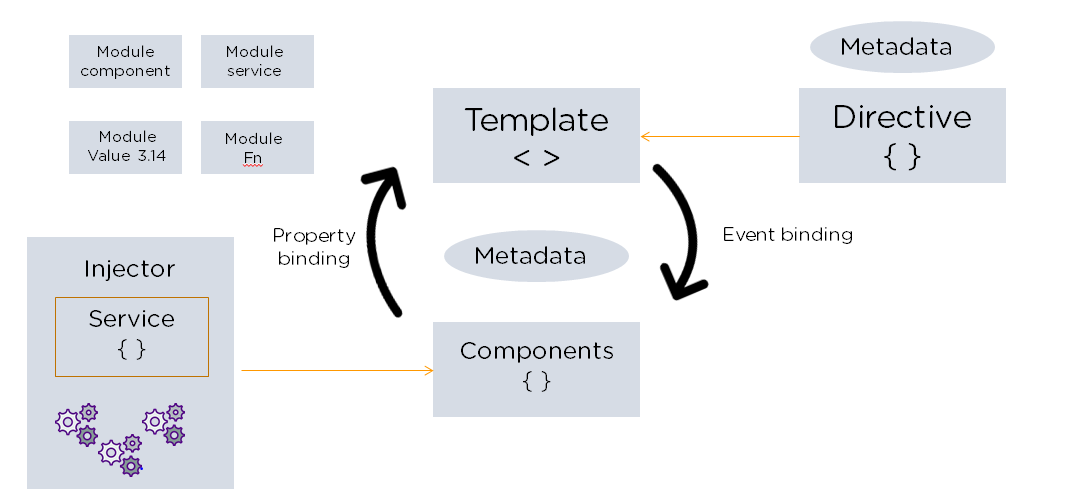
Adatobjektum modell

Az Angular normál DOM-ot használ. Vegyük figyelembe, hogy tíz frissítés történik ugyanazon a HTML-oldalon. Ahelyett, hogy frissítené a már frissítetteket, az Angular frissíti a HTML-címkék teljes fastruktúráját.

**2. TypeScript**

A TypeScript típuskészletet határoz meg a JavaScript számára, amely segít a felhasználóknak könnyebben érthető JavaScript-kód írásában. Az összes TypeScript-kód JavaScripttel fordítódik, és bármilyen platformon zökkenőmentesen futhat. A TypeScript nem kötelező az Angular alkalmazás fejlesztéséhez. Mindazonáltal erősen ajánlott, mivel jobb szintaktikai struktúrát kínál – miközben könnyebben érthetővé és karbantarthatóvá teszi a kódbázis

Az Angular egy teljes értékű modellnézet-vezérlő (MVC) keretrendszer. Világos útmutatást ad az alkalmazás felépítéséhez, és kétirányú adatáramlást kínál, miközben valódi DOM-ot biztosít.



Az angular előnyei:

1. Egyedi komponensek

Az Angular lehetővé teszi a felhasználók számára, hogy saját komponenseiket építsék fel, amelyek a funkcionalitást és a logikát újrafelhasználható darabokká alakíthatják. Webes összetevőkkel is jól játszik.

2. Data binding

Az Angular lehetővé teszi a felhasználók számára, hogy könnyedén mozgassák át az adatokat a JavaScript-kódból a nézetbe, és reagáljanak a felhasználói eseményekre anélkül, hogy kézzel kellene kódot írniuk.

3. Dependency Injection

Az Angular lehetővé teszi a felhasználók számára, hogy moduláris szolgáltatásokat írhassanak, és ott szúrják be őket, ahol szükség van rájuk. Ez javítja ugyanazon szolgáltatások tesztelhetőségét és újrafelhasználhatóságát.

4. Tesztelés

A tesztek első osztályú eszközök, és az Angular az alapoktól kezdve a tesztelhetőség szem előtt tartásával készült. Lehetősége lesz az alkalmazás minden részét tesztelni – ami erősen ajánlott.

5. Átfogó

Az Angular egy teljes értékű keretrendszer, amely kész megoldásokat kínál a szerverkommunikációhoz, az alkalmazáson belüli útválasztáshoz és még sok máshoz.

6. Böngésző kompatibilitás

Az Angular platformfüggetlen, és több böngészővel is kompatibilis. Az Angular alkalmazás jellemzően minden böngészőn (pl.: Chrome, Firefox) és operációs rendszeren futhat, például Windowson, macOS-en és Linuxon.

<https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular>

3.3. HTML és CSS

HTML (HyperText Markup Language) és CSS (Cascading Style Sheets) alapvető építőköveia weboldalaknak. Ezek két különálló technológia, amelyek együttműködve teszik lehetővé a weboldalak kialakítását és formázását.

Az HTML a weboldalak strukturális alapját képezi. Ez a nyelv definiálja azokat az elemeket, amelyeket a böngésző megjelenít a felhasználók számára.

Az HTML egy tag-alapú nyelv, amely különböző elemeket és tartalmat jelöl meg különböző tag-ek segítségével, például fejléc, szöveg, képek, táblázatok stb.

A CSS felelős a weboldalak megjelenéséért és formázásáért. Ez a nyelv különválasztja az adatok tartalmát és megjelenését.

A CSS segítségével a fejlesztők stílusokat és formázási tulajdonságokat adhatnak a HTML elemekhez, például színek, betűtípusok, elrendezés, méretek stb.

A CSS lehetővé teszi a stílusok újrafelhasználását és a konzisztens megjelenés biztosítását az egész weboldalon.

Összességében az HTML és a CSS alapvető fontosságúak a webfejlesztésben, és a modern weboldalak megalkotásának alapjait képezik. Az HTML az információ strukturálására szolgál, míg a CSS a megjelenést és a formázást kezeli. A két technológia együttműködése segít a kreatív és esztétikus weboldalak létrehozásában.

3.4. Bootstrap

A Bootstrap egy ingyenes, nyílt forráskódú front-end fejlesztői keretrendszer webhelyek és webes alkalmazások létrehozásához. Úgy tervezték, hogy lehetővé tegye a mobil-első webhelyek reszponzív fejlesztését, és a Bootstrap szintaxis gyűjteményt biztosít a sablontervekhez.

Keretrendszerként a Bootstrap tartalmazza a reszponzív webfejlesztés alapjait, így a fejlesztőknek csak be kell illeszteni a kódot egy előre meghatározott grid rendszerbe. A Bootstrap keretrendszer a Hypertext Markup Language (HTML), a lépcsőzetes stíluslapokra (CSS) és a JavaScriptre épül. A Bootstrapet használó webfejlesztők sokkal gyorsabban készíthetnek webhelyeket anélkül, hogy az alapvető parancsokkal és funkciókkal kellene foglalkozniuk.

A Bootstrap valósággá teszi a reszponzív webdesignt. Lehetővé teszi, hogy egy weboldal vagy alkalmazás észlelje a látogató képernyőméretét és tájolását, és ennek megfelelően automatikusan igazítsa a megjelenítést. A mobil-első megközelítés azt feltételezi, hogy az okostelefonok, táblagépek és a feladatspecifikus mobilalkalmazások az alkalmazottak elsődleges eszközei a munkavégzéshez. A Bootstrap megfelel ezeknek a technológiáknak a tervezési követelményeinek, és magában foglalja a felhasználói felület összetevőit, elrendezéseit, JavaScript-eszközöket és a megvalósítási keretrendszert. A szoftver előre lefordított formában vagy forráskódként érhető el.

Mark Otto és Jacob Thornton kifejlesztette a Bootstrap programot a Twitteren, hogy javítsa az oldalon használt eszközök konzisztenciáját és csökkentse a karbantartást. A szoftvert korábban Twitter Blueprint néven ismerték, és néha Twitter Bootstrap néven is emlegetik.

3.5 Node Js + Express Js

<https://nodejs.org/en/about>

A Node.js egy szoftverrendszer, melyet skálázható internetes alkalmazások, mégpedig webszerverek készítésére hoztak létre A programok JavaScript-ben írhatók, eseményalapú, aszinkron I/O-val a túlterhelés minimalizálására és a skálázhatóság maximalizálására.

Express:

Az Express egy csomópontos js webalkalmazás-keretrendszer, amely széles körű funkciókat kínál webes és mobilalkalmazások létrehozásához. Egyoldalas, többoldalas és hibrid webalkalmazások készítésére szolgál.

Ez a Node js tetejére épített réteg, amely segít a szerverek és útvonalak kezelésében.  
  
Miért Express?

Az Express-t az API-k és webalkalmazások egyszerű elkészítésére hozták létre,

Sok kódolási időt takarít meg majdnem felére, és továbbra is webes és a mobil alkalmazások hatékonyak.

Az expressz használatának másik oka az, hogy javascriptben van írva, mivel a javascript egyszerű nyelv még akkor is, ha nincs korábbi

bármely nyelv ismerete. Az Express segítségével sok új fejlesztő léphet be a webfejlesztés területére. A node js expressz keretrendszerének létrehozásának oka a következő:

Időtakarékos, Gyors , Gazdaságos, Könnyű megtanulni, Aszinkron

3.6. MongoDB <https://www.mongodb.com/company/what-is-mongodb>

A MongoDB rugalmas, JSON-szerű dokumentumokban tárolja az adatokat, ami azt jelenti, hogy a mezők dokumentumonként változhatnak, és az adatstruktúra idővel változhat

A dokumentummodell az alkalmazáskódban lévő objektumokhoz van leképezve, így az adatok könnyen kezelhetők

Az ad hoc lekérdezések, az indexelés és a valós idejű összesítés hatékony módszereket kínál az adatok elérésére és elemzésére

A MongoDB egy elosztott adatbázis, így a magas rendelkezésre állás, a vízszintes skálázás és a földrajzi eloszlás beépített és könnyen használható

A MongoDB ingyenesen használható. A 2018. október 16. előtt kiadott verziók az AGPL hatálya alá tartoznak.

A MongoDB használata előnyös lehet számos esetben, különösen akkor, ha olyan alkalmazástvagy rendszert fejlesztesz, amelynek az adatkezelési igényei jól illeszkednek a NoSQL adatbázisok által nyújtott lehetőségekhez. Néhány olyan helyzet, amikor előnyös lehet a MongoDB használata:

: A MongoDB JSON formátumot használ az adatok tárolására és lekérdezésére, ami természetes módon integrálódik modern fejlesztői környezetekkel és alkalmazásokkal.

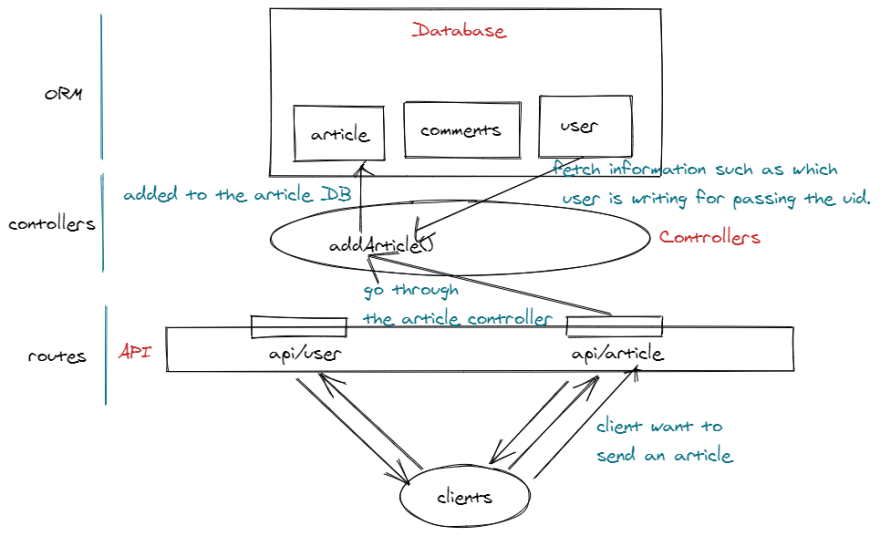
6. TypeORM

<https://www.tutorialspoint.com/typeorm/typeorm_introduction.htm>

A TypeORM keretrendszer egy Object Relational Mapping (ORM) keretrendszer. Általában az objektum rész az alkalmazásban lévő tartományra/modellre vonatkozik, a relációs rész a relációs adatbázis-kezelő rendszer táblái közötti kapcsolatra (pl. Oracle, MySQL, MS-SQL, PostgreSQL stb.), végül a Leképezés rész a a modell és az asztalaink áthidalásának aktusa.

Az ORM egy olyan eszköz, amely az entitásokat adatbázistáblákkal képezi le. Az ORM egyszerűsített fejlesztési folyamatot biztosít az objektum-tábla és tábla-objektum átalakítás automatizálásával. Miután az adatmodellt egy helyre írhatja, könnyebbé válik a kód frissítése, karbantartása és újrafelhasználása.

Mivel a modell gyengén kötődik az alkalmazás többi részéhez, megváltoztathatja azt anélkül, hogy az alkalmazás más részeivel keményen függne, és az alkalmazáson belül bárhol könnyen használható. A TypeORM nagyon rugalmas, elvonatkoztatja az adatbázis-rendszert az alkalmazástól, és lehetővé teszi számunkra, hogy az OOPS-koncepció előnyeit élvezzük.



Visual Studio Code

A fejlesztés során Visual Studio Code-ot fogok használni, mivel számomra ez a legkézenfekvőbb ingyenes fejlesztő környezet. Megtaláható benne a legtöbb bővítmény amely elősegíti a kódolási folyamatokat Léteznek más IDE-k is pl.: WebStrom vagy Eclipse viszont aWebStrom Licenchez kötött az Eclips pedig nem tartalmaz minden bővítményt.