

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-5384-8766

**VIRTUÁLNE LABORATÓRIUM ZALOŽENÉ NA  
TECHNOLÓGIÍ NODE.JS IMPLEMENTOVANÉ DO  
MOODLE  
DIPLOMOVÁ PRÁCA**

**2016**

**Bc. Jakub Hoblík**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-5384-8766

**VIRTUÁLNE LABORATÓRIUM ZALOŽENÉ NA  
TECHNOLÓGIÍ NODE.JS IMPLEMENTOVANÉ DO  
MOODLE**

**DIPLOMOVÁ PRÁCA**

Študijný program :	Aplikovaná informatika
Číslo študijného odboru:	2511
Názov študijného odboru:	9.2.9 Aplikovaná informatika
Školiace pracovisko:	Ústav informatiky a matematiky
Vedúci záverečnej práce:	Ing. Pavol Bisták PhD.

# SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program :	Aplikovaná informatika
Diplomová práca:	Virtuálne laboratórium založené na technológií Node.JS implementované do Moodle
Autor:	Bc. Jakub Hoblík
Vedúci záverečnej práce:	Ing. Pavol Bisták PhD.
Miesto a rok predloženia práce:	Bratislava 2016

Cieľom tejto diplomovej práce je na základe projektu StarkLab vytvoriť modul pre systém Moodle, ktorý by slúžil ako klientská časť aplikácie. Súčasťou práce je naštudovať si projekt StarkLab, navrhnúť a implementovať vylepšenia a zamerať sa na jeho rozšírenie a rozšíriteľnosť do budúcnosti.

Kľúčové slová: node.js javascript laboratórium moodle php

# ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Applied Informatics
Diploma Thesis:	Virtual laboratory based on Node.JS technology implemented into Moodle
Autor:	Bc. Jakub Hoblík
Supervisor:	Ing Pavol Bisták PhD.
Place and year of submission:	Bratislava 2016

The aim of this thesis is create a module based on project StarkLab which will fit into Moodle as a client part of application. A part of thesis is to get familiar with project StarkLab and suggest some improvements and implement them also the thesis will focus on expandibility of project.

Key words: node.js javascript laboratórium moodle php

# Vyhlásenie autora

Podpísaný Bc. Jakub Hoblík čestne vyhlasujem, že som Diplomovú prácu Virtuálne laboratórium založené na technológií Node.JS implementované do Moodle vypracoval na základe poznatkov získaných počas štúdia a informácií z dostupnej literatúry uvedenej v práci.

Uvedenú prácu som vypracoval pod vedením Ing. Pavol Bisták PhD.

V Bratislave dňa 23.12.2016

.....

podpis autora

# **Pod'akovanie**

Chcel by som sa poďakovať vedúcemu práce Ing. Pavlovi Bistákovi PhD. za zhovievavý prístup a výborné odborné vedenie počas tvorenia tejto práce.

# Úvod

V tejto práci sa budeme venovať virtuálnym laboratóriam. Laboratórium je miesto, kde sa uskutočňujú experimenty. Bežný proces experimentu vyžaduje aby vôbec nejaké fyzické laboratórium existovalo, bolo prístupné a vybavené potrebným zariadením. Ďalej musí byť prítomný laborant aj zaškolený vedúci, ktorý dozerá na priebeh a správnosť experimentu. Takýto rokmi overený prístup k praktickej časti vzdelávania začína byť pomaly zastaraný vzhľadom na to ak rýchlo sa vyvíjajú technológie takmer vo všetkých oblastiach. Nové možnosti hlavne vo webových technológiách nám umožňujú vytvárať veci aké doteraz v tomto priestore neboli možné a tým sa dostávame k pojmu virtuálne laboratórium.

Prínos nových technológií nám umožňuje vytvárať nové veci ale umožňuje nám prerobiť staré veci oveľa lepším, efektívnejším a aj dizajnovo krajším spôsobom preto sa v tejto práci pokúsime využiť čo najnovšie nástroje vhodné pre tvorbu virtuálneho laboratória vo webovom prostredí. Týmto spôsobom sa pokúsime priniesť užívateľovi lepší zážitok a tým zlepšiť kvalitu štúdia.

## Cieľ práce

Cieľom tejto diplomovej práce je oboznámiť sa problematikou virtuálnych laboratórií, popísať ich výhody, nevýhody a preskúmať technológie vhodné pre túto problematiku. Ďalším cieľom je oboznámiť sa s minuloročnou diplomovou prácou Ing. Ericha Starka, ktorou cieľom bolo vytvoriť funkčné virtuálne laboratórium s výhradným použitím javascriptu na strane klienta ale aj na strane servera. V rámci analýzy tejto práce preskúmať možnosti na vylepšenie projektu. Po preskúmaní týchto aspektov nasleduje úloha vytvoriť modul, ktorý bude implementovaný do edukačného prostredia Moodle, kde študenti ale aj učitelia budú mať možnosť preskúmať toto virtuálne laboratórium a tým získať poznatky z oblastí implementovaných experimentov. Po dosiahnutí týchto cieľov sa zameriame na rozšíriteľnosť virtuálneho laboratória o nové experimenty.



# 1 Analýza StarkLabu

Prvým krokom bolo oboznámiť sa s minuloročnou diplomovou prácou Ing. Ericha Starka, ktorá bola zameraná na tvorbu virtuálneho laboratória pomocou javascriptu jak na strane klienta tak aj na strane servera. Toto laboratórium nazval StarkLab. Po prečítaní jeho práce som získal teoretický prehľad o tom ako by jeho práca mala fungovať.

StarkLab sa skladá z troch hlavných častí a to klient, server a matlabovská časť.

Klientská časť aplikácie je napísaná vo frameworku s názvom AngularJS. Táto časť aplikácie umožňuje prihlásenie užívateľov pomocou LDAP. Každý učiteľ či študent, ktorý má platný účet v akademickom informačnom systéme STU je schopný sa takýmto účtom prihlásiť do StarkLabu. Po prihlásení sa užívateľovi zobrazí stránka na ktorej je popísaná problematika experimentu šikmého vrhu keďže StarkLab momentálne vie uskutočniť len tento jeden experiment. Ďalej užívateľ vie zadať počiatočné hodnoty pre uhol a rýchlosť telesa pre ktoré si želá spustiť experiment. Po zadaní týchto hodnôt je užívateľ ďalej presmerovaný na stránku, kde čaká pokiaľ sa experiment začne vykonávať vo výpočtovom prostredí Matlab. Keď Matlab začne posielať dáta v reálnom čase na server, tento server ich hneď preposiela Klientovi, ktorý ich zobrazuje dvoma spôsobmi a to ako hodnoty v grafe a ako animáciu letiaceho telesa. Po ukončení výpočtu a prenosu vie užívateľ prehliadať tieto dáta v grafe a tak isto si vie výsledok experimentu uložiť do databázy pre budúce prehliadanie.

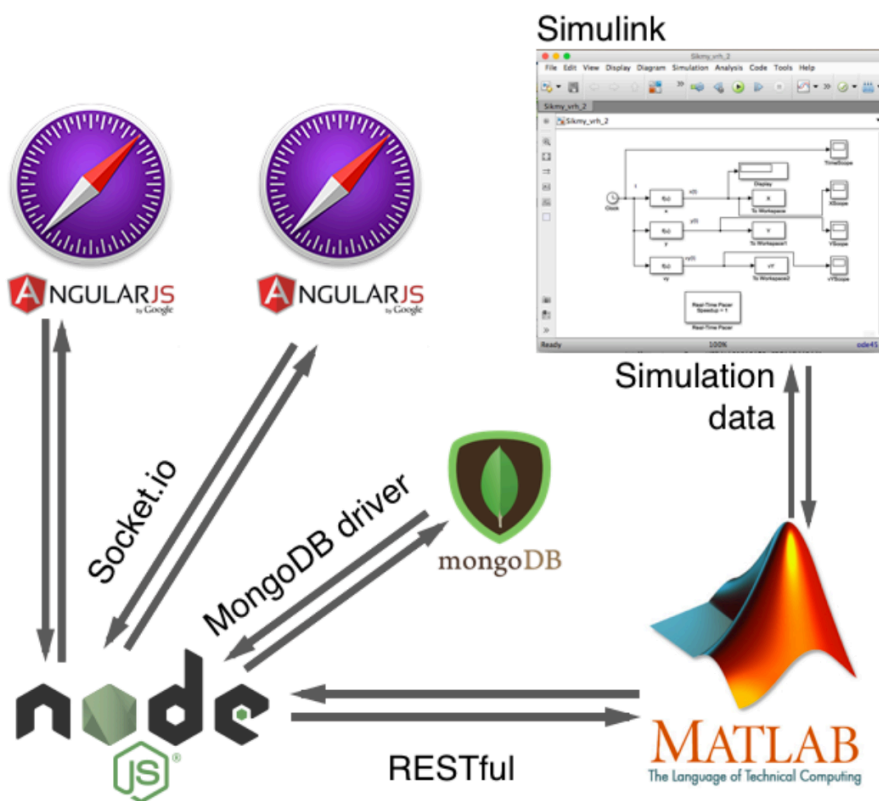
Serverová časť je vytvorená pomocou Express.js čo je opäť javascriptový framework vhodný na tvorbu web serverov, ktoré bežia na Node.js.

Server spája Klienta s Matlabom a to tak, že server obdrží vstupné hodnoty od Klienta ďalej spustí Matlab pomocou príkazového riadku, Matlab začne výpočet experimentu a odovzdáva hodnoty serveru, ktorý ich v reálnom čase preposiela Klientovi priamo na zobrazenie. Ďalej server vie zapisovať výsledky experimentu do databázy. Ako databáza bola zvolená MongoDB databáza. Je to NoSQL databáza, ktorá ukladá dáta vo formáte JSON.

Matlab je v tomto projekte využitý ako výpočtová jednotka, ktorá vie komunikovať REST službami na požadovanú URL čo je veľmi vhodné a užitočné lebo to uľahčuje komunikáciu serveru a Matlabu. Vždy po vypočítaní novej hodnoty pre trajektóriu šikmého vrhu je táto hodnota zapísaná na server matlabovskou funkciou *webwrite*, ktorá vykoná POST request na server tým server obdrží dáta, ktoré ďalej preposiela klientovi. Keby Matlab neposkytoval takúto formu komunikácie bolo by obtiažne z neho dostávať dáta, keďže týmto dostupným spôsobom vie komunikovať s hocijakou URL.

## 1.1 Architektúra Starklabu

Na nižšie uvedenom obrázku je zobrazená architektúra virtuálneho laboratória StarkLab.



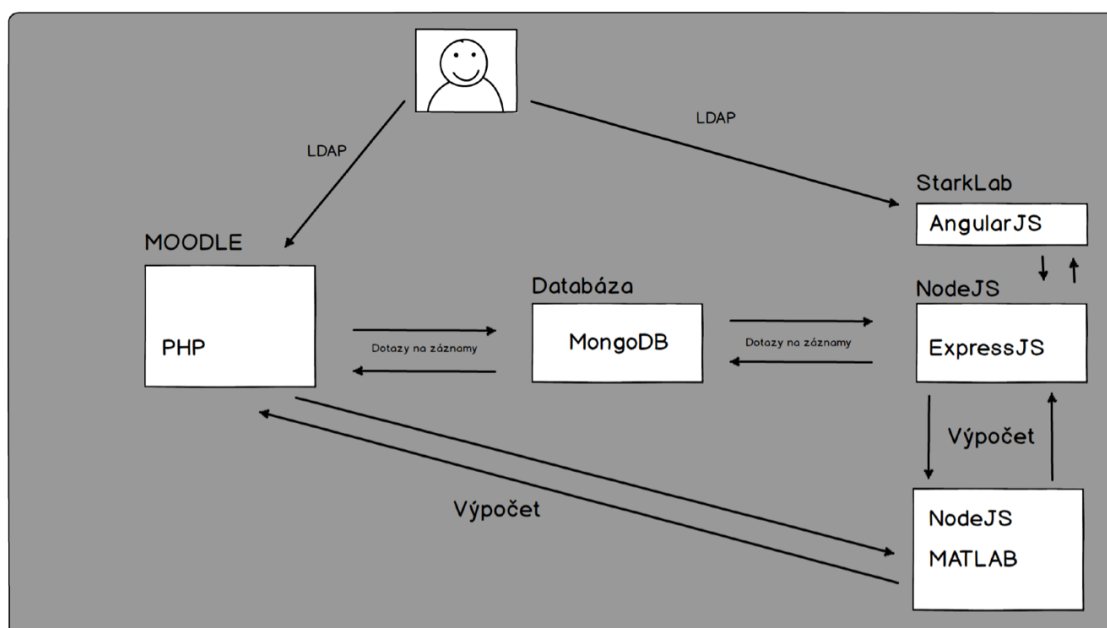
[1]

Obrázok 1 - Architektúra StarkLabu

## 1.2 Nová architektúra

Na nižšie uvedenom obrázku je zobrazená nová architektúra, ktorá vytvorí samostatný server, kde tento server bude obsluhovať Matlab. V predošlej verzii bol Matlab obsluhovaný web serverom, ktorý tak isto obsluhoval AngularJS klienta. Táto zmena prinesie flexibilitu ako aj väčšiu modulárnosť projektu s lepším využitím do budúcnosti pri snahe o rozšírenie.

Druhou závažnou zmenou je vytvorenie PHP modulu, ktorý bude implementovaný v Moodli. Úlohou tohto modulu bude pre prihláseného užívateľa zprostredkovať funkcionality StarkLabu ako prehliadanie experimentov, vytváranie nových simulácií a prehliadanie už nasimulovaných dát. Do Moodlu sa bude užívateľ prihlasovať tak ako do StarkLabu čiže vie pomocou mena a hesla platných pre akademický informačný systém FEI STU.



Obrázok 2- Nová Architektúra

## **2 Laboratória**

V tejto kapitole sa budeme venovať pojmom virtuálne a fyzické laboratórium. Popíšeme si ich význam, použitie, výhody a nevýhody

### **2.1 Fyzické laboratórium**

Fyzické laboratórium je také laboratórium, kde laborant musí byť fyzicky prítomný v laboratóriu a interagovať so sústavou zariadení na ktorej chce daný experiment vykonať. Takýto prístup pri niektorých komplikovaných pokusoch a experimentoch ostáva ako jediná možnosť ale my sa budeme venovať jednoduchším prípadom experimentov, ktoré sa využívajú skôr na edukačné účely v školských laboratóriách. Pre takéto pokusy má fyzické laboratórium viac nevýhod ako výhod.

#### **2.1.1 Výhody fyzického laboratória**

Výhodou fyzického laboratória je, že poskytuje živý zážitok z experimentu a tým vie u človeka zanechať lepší dojem a v niektorých prípadoch lepšie porozumenie problematiky, keďže v takýchto experimentoch je človek nútený si sám všetko správne nastaviť a kontrolovať priebeh experimentu.

#### **2.1.2 Nevýhody fyzického laboratória**

Medzi nevýhody fyzického laboratória môžeme zaradiť napríklad to, že laborant musí byť prítomný v laboratóriu a pracovať so zariadeniami. Z toho vyplýva, že zariadenie je obsadené a nemôže na ňom pracovať súčasne viac laborantov čo vytvára potrebu plánovania rozvrhu pre dané zariadenia. Ďalej tým, že so zariadeniami sa manuálne zaobchádza sa ich stav zhoršuje a podliehajú opotrebeniu a ešte aj zaberajú miesto čiže je potrebné fyzicky vyhradiť miesto.



*Obrázok 3- Fyzické laboratórium*

## **2.2 Virtuálne laboratórium**

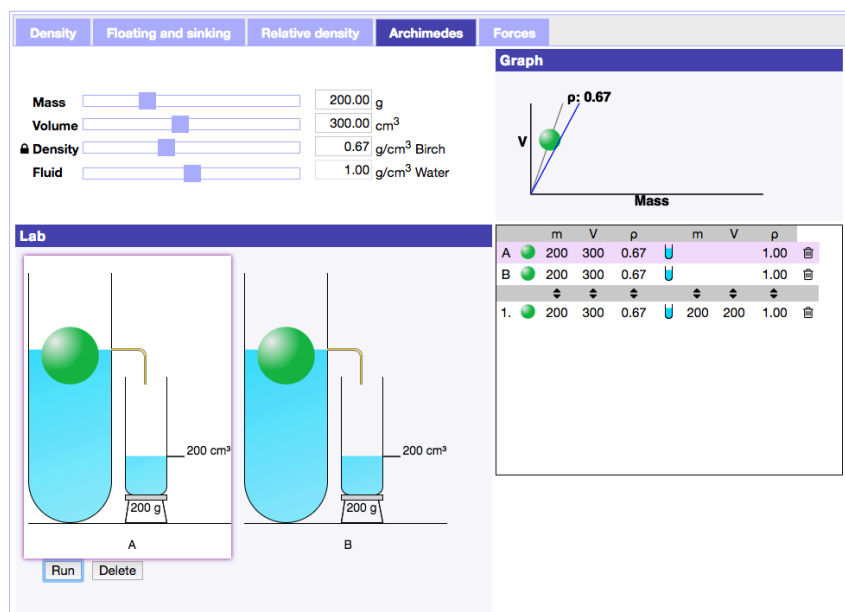
Virtuálne laboratórium je také laboratórium, ktoré nevyžaduje fyzickú prítomnosť laboranta v laboratóriu. Takéto laboratórium je rozdelené spravidla na dve časti a to klientská časť a serverová časť. Klientská časť je interaktívna časť laboratória s ktorou laborant pracuje. Zvyčajne funguje tak, že laborant si zvolí z ponuky experimentov, ďalej si naštuduje teoretickú stránku experimentu, zadá vstupné parametre pre daný experiment a pošle ich na server. Serverová časť aplikácie prijme dáta, vykoná daný výpočet pre vybraný experiment a vráti výsledné hodnoty opäť k lietovi, ktorý ich už patrične reprezentuje užívateľovi v prijateľnej forme ci už animáciou alebo ako údaje v grafoch.

### **2.2.1 Výhody virtuálneho laboratória**

Výhod je hneď niekoľko, nakoľko sa jedná o virtuálne laboratórium tak je možné ho používať z pohodlia domova, stačí mať len prístup ku klientovi. Nie je potrebné vybavovať miestnosti rôznymi zariadeniami, ktoré vedia byť nahradené výpočtovými prostrediami ako Matlab. V takomto prípade je aj finančne výhodnejšie zakúpiť licenciu pre dané výpočtové prostredie ako nákup zariadení. V prípade nejakej poruchy je možné server jednoducho reštartovať. Ďalej vieme vypočítané dáta ľahko ukladať do rôznych typov databáz a využiť ich opätovne neskôr. Medzi najhlavnejšie výhody patrí takmer neobmedzená rozšíriteľnosť

## 2.2.2 Nevýhody virtuálneho laboratória

Predsa len experimenty vykonávané vo virtuálnom prostredí sú simulované a snažia sa odzrkadliť realitu, kde sa vytvára priestor pre chyby. Tieto chyby sa na prvý pohľad dajú prehliadnuť a môžu sa prejaviť neskôr pri spracovaní výsledkov alebo sa neprejavajú vôbec čo môže viesť k chybným záverom.

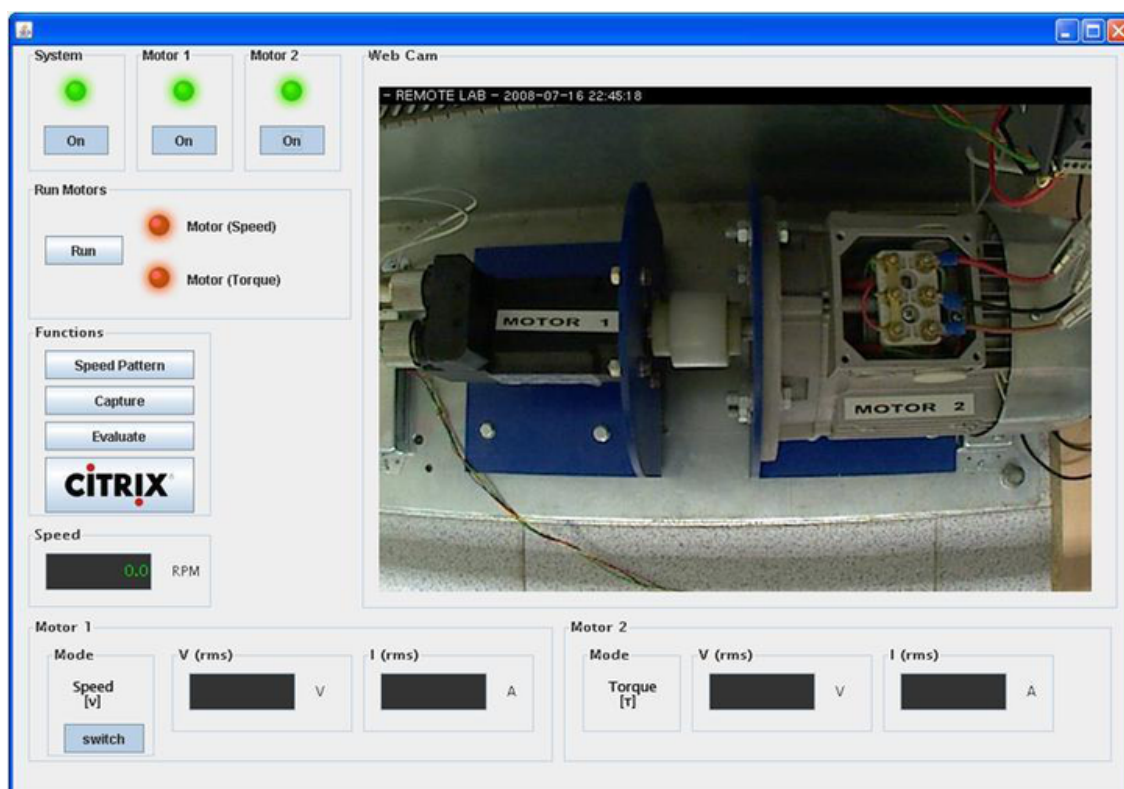


Obrázok 4- Virtuálne Laboratórium

## 2.3 Vzdialené laboratória

Ďalšou z možností je kombinácia virtuálneho a fyzického laboratória, kde je experiment vykonávaný na fyzickom zariadení ale laborant sa fyzicky nenachádza pri zariadení ale spravuje ho vzdialene. Funkčne pripomína virtuálne laboratórium, kde užívateľ zadá vstupné hodnoty do klientskej aplikácie, ktorá opäť komunikuje so serverom a ten predáva hodnoty ďalej do fyzického zariadenia. Pozorovať takýto experiment je možné napríklad pomocou web kamery.

Takýto prístup k fyzickým zariadeniam vie zaručiť, že zariadenia budú vyžité len na také účely na aké boli navrhnuté, keďže užívateľ je limitovaný funkcionalitou klientskej aplikácie.



Obrázok 5- Vzdialené Laboratórium

## 3 Virtuálne laboratórium podrobnejšie

Kedže táto práca je venovaná virtuálnemu laboratóriu v tejto kapitole sa jeho problematike budeme venovať podrobnejšie

### 3.1 Architektúra

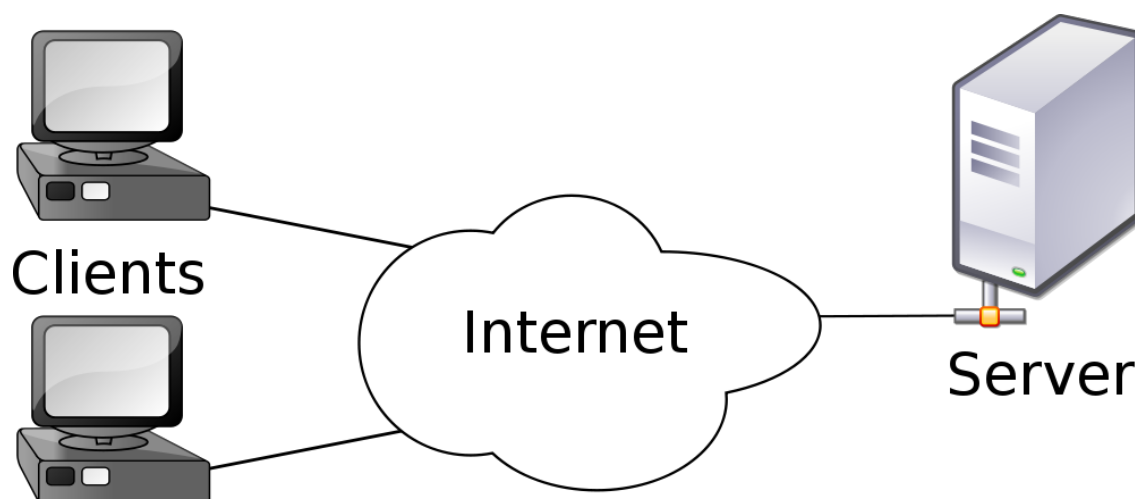
Ako už bolo uvedené vyššie virtuálne laboratórium pozostáva z dvoch častí a to sú server a klient.

Server je časť architektúry, ktorú užívateľ nevidí a ani s ňou neinteraguje. Server prijíma dáta od klientskej časti aplikácie a spracúva ich. Spracúva ich buď on pomocou svojich algoritmov alebo ich posúva ďalej do výpočtových prostredí ako napríklad Matlab, ktorý vie simulovať komplikovanejšie experimenty. Ďalej vie tieto dáta ukladať do databázy pre opätovné pracovanie s už nasimulovanými dátami. Keď server dokončí svoju prácu nad dátami posiela ich späť klientovi, kde už s nimi ďalej pracuje užívateľ.

Klient je vlastne užívateľské rozhranie s ktorým užívateľ prichádza do styku. Vie v ňom nadobudnúť teoretické znalosti týkajúce sa experimentu, ktorý sa práve chystá vykonať. Ďalej vie vkladať vstupné hodnoty a spúšťať experimenty. Po úspešne prenesených dátach na server a späť sa dáta reprezentujú užívateľovi v rozumnej forme a to buď graficky v grafoch alebo v animáciách alebo ako dáta zobrazené v tabuľke.

Úlohou virtuálnych laboratórií je sprístupniť základné poznatky, realizovať experimenty, ktoré by neboli prístupné kvôli či už kapacitným alebo materiálnym nedostatkom. Taktiež sa snažia vytvoriť komplexný študijný materiál pre vzdelávanie v rôznych oblastiach alebo ako doplnok k praktickým predmetom či cvičeniam.





Obrázok 6- Klient Server Architektúra

### 3.2 Virtuálne Laboratória a technológie

Vzhľadom na to, že virtuálne laboratória sú postavené na klient server architektúre tak tieto dve časti mohli byť implementované pomocou rôznych technológií. V minulosti, keď ešte Javascript nebol taký populárny ako je dnes túto pozíciu zastávala Java. Java patrí samozrejme aj dnes medzi obľúbené nástroje vývojárov ale jej popularita klesá hlavne v projektoch ako napríklad virtuálne laboratória, kde nie je potrebná veľmi vysoká ochrana alebo robustnosť ako ponúkajú nástroje Javy. V prípade ak je server napísaný v Jave tak sa môžeme naň pripájať pomocou javovského klienta alebo ak je server formovaný ako web server, tak v tom prípade môžeme použiť web prehliadač.

Ďalšou celosvetovo používanou technológiou je PHP. PHP je využívané na strane serverov čo zapadá do architektúry virtuálnych laboratórií. V prípade vytvorenia servera pre laboratórium pomocou PHP sa jedná o web server. V tomto prípade potom web prehliadač slúži ako klient k tomuto serveru. Čiže dáta sú prezentované ako web stránka.

Medzi najnovšie technológie patrí platforma NodeJS. Jedná sa o javascript na strane servera. Mnohé moderné aplikácie prijali tento trend a využívajú javascript aj na strane server aj na strane klienta. Tento prístup má viacero výhod ale hlavnou je, že pre vytvorenie takejto aplikácie stačí aby programátor ovládal iba jeden skriptovací jazyk a to javascript. Aj keď v dnešnej dobe je takmer nutnosťou mať aké také znalosti z viacerých oblastí.

Dôležitou súčasťou architektúry je komunikácia. Laboratória zväčša majú za úlohu simulovať nejaký netriviálny problém čiže klient nemá dostatočnú výpočtovú silu a preto prenecháva túto úlohu na server. Preposlať vstupné dáta na server nie je problém keďže sa nejedná o veľký počet hodnôt ale zložitejšie je poslať výsledné dáta späť klientovi. Tieto výsledné vypočítané dáta môžeme zaslať naraz po ukončení výpočtu, kde však vzniká veľká časová medzera medzi odoslaním požiadavky a vráteného výsledku. Preto lepší prístup k tomuto problému je preposielanie výsledkov v reálnom čase. Na komunikáciu medzi web serverom a web klientom je najvhodnejšie využitie technológie web socketov, kde sa nadviaže spojenie medzi serverom a klientom na začiatku komunikácie, následne si preposielajú správy, kde obsahom týchto správ sú fragmenty výpočtu serveru v časovej postupnosti. Ďalšou metódou komunikácie je možnosť využiť REST služby ale tento spôsob je menej vhodný, keďže pri každom jednom prenose sa vytvára spojenie medzi serverom a klientom na novo.

Zobrazovanie prijatých dát servera najčastejšie prebieha vo web prehliadači, ktorý využíva technológiu HTML5 a javascript. Javascript poskytuje neobmedzené možnosti pre reprezentáciu dát či už graficky ako animácie alebo ako dáta v grafe. Pre animácie si môžeme vybrať z dvoch technológií a to canvas alebo svg. Canvas ponúka nástroje na rastrovú 2D a 3D grafiku a svg ponúka nástroje pre vektorovú grafiku.

Reprezentácia dát do grafov by bez rôznych knižníc bola pravdepodobne veľmi komplikovaná ale v dnešnej dobe je už veľké množstvo dostupných nástrojov, ktoré tento problém vedia vyriešiť za nás a umožňujú tak reprezentáciu v rôznych typoch grafov.