



Univerza v Ljubljani  
Fakulteta *za računalništvo  
in informatiko*

Univerza v Ljubljani  
**Fakulteta za računalništvo in informatiko**  
Šol. l. 2019/2020

# Dodatna seminarska naloga

Pri predmetu spletne tehnologije



## KAZALO

<b>1.</b>	<b>DOMAČA NALOGA SPACE INVADERS .....</b>	<b>0</b>
<b>2.</b>	<b>DOMAČA NALOGA RESTAVRACIJA .....</b>	<b>0</b>
<b>3.</b>	<b>TEHNOLOGIJE IN KNJIŽNICE NA STRANI ODJEMALCA .....</b>	<b>1</b>
3.1.	HTML: HYPERTEXT MARKUP LANGUAGE .....	1
3.2.	CSS: CASCADING STYLE SHEETS .....	2
3.3.	JAVASCRIPT .....	2
3.4.	JQUERY .....	2
3.5.	BOOTSTRAP .....	3
<b>4.</b>	<b>TEHNOLOGIJE IN KNJIŽNICE NA STRANI STREŽNIKA .....</b>	<b>3</b>
4.1.	PHP: HYPERTEXT PREPROCESSOR .....	3
4.2.	LARAVEL .....	3
4.3.	MYSQL .....	3
4.4.	PHPMYADMIN .....	3
<b>5.</b>	<b>PODATKOVNA BAZA .....</b>	<b>4</b>
<b>6.</b>	<b>ZAGOTAVLJANJE VARNOSTI .....</b>	<b>5</b>
6.1.	1. DOMAČA NALOGA .....	5
6.2.	3. DOMAČA NALOGA .....	5
<b>7.</b>	<b>SKLADNOST Z GDPR .....</b>	<b>5</b>
7.1.	1. DOMAČA NALOGA .....	5
7.2.	3. DOMAČA NALOGA .....	5
<b>8.</b>	<b>ENO-STRANSKE APLIKACIJE .....</b>	<b>6</b>
8.1.	SPA TEHNOLOGIJE .....	6
8.2.	PRIMERJAVA MPA IN SPA .....	7
8.3.	SPA OGRODJA .....	7
<b>9.</b>	<b>AJAX IN WEBSOCKET .....</b>	<b>8</b>
9.1.	AJAX .....	8
9.2.	WEBSOCKET .....	9
9.3.	PRIMERJAVA AJAX IN WEBSOCKET .....	9

## KAZALO SLIK

Slika 1: Space invaders .....	0
Slika 2: Restavracija .....	1
Slika 3: Shema podatkovne baze.....	4

## 1. DOMAČA NALOGA SPACE INVADERS

Igra temelji na nekdanji priljubljeni igri Space invaders, po kateri sem se zgledoval in osnovno idejo preoblikoval. Obdržal sem veliko osnovnih zasnov igre in dodal svoje izboljšave. Cilj igre je preigrati vse stopnje, premagati sovražnike in doseči čim višje število točk. Igralec upravlja vesoljsko plovilo, strelja na sovražnike in se izmika njihovim izstrelkom. Na voljo ima tri življenja, ki se ob trku s sovražnikovimi naboji odštevajo. Če igralec izgubi vsa življenja, se igra povrne nazaj na prvo stopnjo. Med igro se na zaslon izpisuje število preostalih igralčevih življenj in nabojev. Aplikacija je izdelana v programskem jeziku JavaScript in uporablja JQuery knjižnico. Namenjena je uporabi v vseh sodobnih spletnih brskalnikih. Deluje na različnih velikostih zaslonov, saj se prilagaja njihovi ločljivosti. Igranje še dodatno poživijo razni zvočni učinki.

Ob zagonu igre je uporabnik pozvan k vnosu imena in priimka. Po vnosu se uporabniku prikažejo navodila za igranje igre. Igro začne s klikom na ustrezen gumb. Uporabnik lahko spremeni svoje podatke in si ogleda lestvico vseh rezultatov.

Uporabljene tehnologije in knjižnice: HTML, CSS, JavaScript, JQuery.



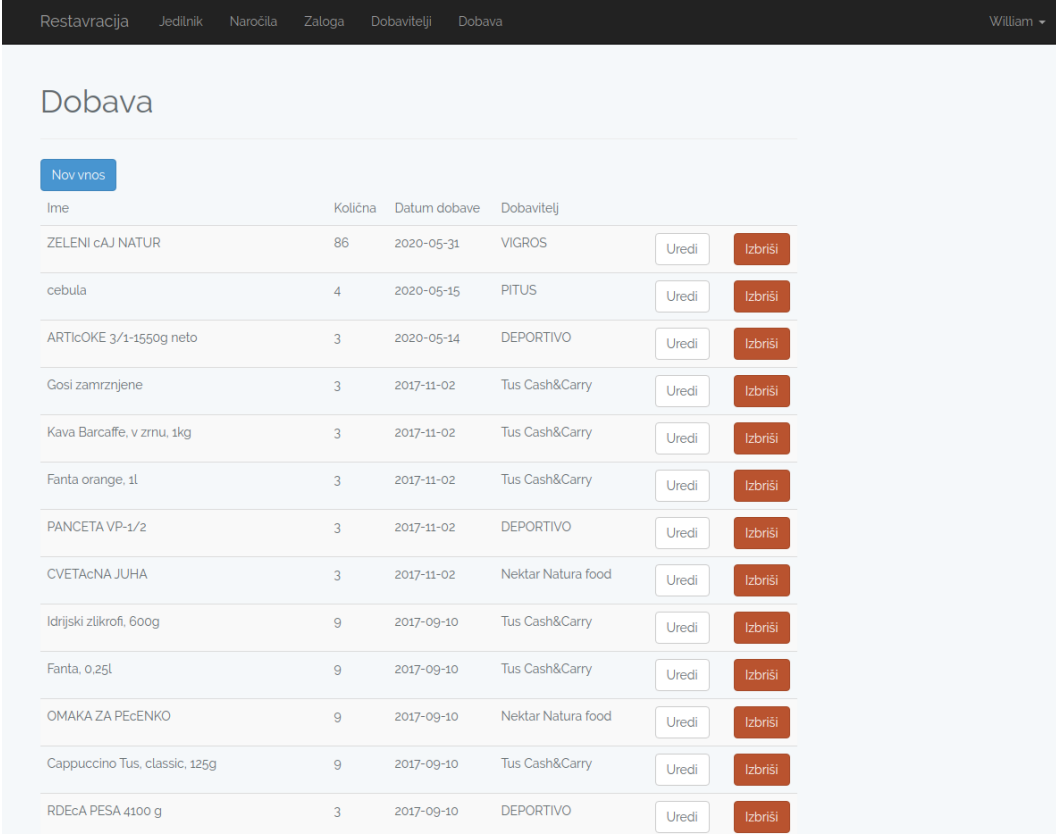
Slika 1: Space invaders

## 2. DOMAČA NALOGA RESTAVRACIJA

Aplikacija omogoča učinkovito upravljanje restavracije. Razvita je z uporabo Laravel ogrodja. Omogoča prijavo in registracijo uporabnikov. Neprijavljenim uporabnikom omogoča ogled jedilnika, naročil, zaloge, dobaviteljev, dobave in zaloge. Ob poskusu izvajanja akcije dodajanja jih preusmeri na prijavno stran, saj nimajo ustreznih pravic. Gumbi za urejanje in brisanje so za njih skriti. Prijavljeni uporabniki lahko tudi dodajajo, spreminjajo in brišejo podatke. Vsak uporabnik lahko spreminja le svoja naročila in ne drugih. Za primerno hranjenje in obdelavo podatkov je poskrbljeno z uporabo Mysql strežnika. Dodatna varnost je zagotovljena z uporabo kriptografske hash funkcije za shranjevanje gesel. Vsi vnosi so preverjeni tako na strani

odjemalca, kot tudi na strani strežnika. Uporabniki so o napačnih vnosih obveščeni. Vmesnik je urejen in uporabniku prijazen. Vsi uporabljeni HTTP zahtevki so standardni.

Uporabljene tehnologije in knjižnice: HTML, CSS, JavaScript, JQuery, Bootstrap, PHP, Laravel, MySql, PhpMyAdmin.



Ime	Količina	Datum dobave	Dobavitelj		
ZELENI cAJ NATUR	86	2020-05-31	VIGROS	<button>Uredi</button>	<button>Izbriši</button>
cebula	4	2020-05-15	PITUS	<button>Uredi</button>	<button>Izbriši</button>
ARTiCKE 3/1-1550g neto	3	2020-05-14	DEPORTIVO	<button>Uredi</button>	<button>Izbriši</button>
Gosi zamrznjene	3	2017-11-02	Tus Cash&Carry	<button>Uredi</button>	<button>Izbriši</button>
Kava Barcaffe, v zrnu, 1kg	3	2017-11-02	Tus Cash&Carry	<button>Uredi</button>	<button>Izbriši</button>
Fanta orange, 1l	3	2017-11-02	Tus Cash&Carry	<button>Uredi</button>	<button>Izbriši</button>
PANCETA VP-1/2	3	2017-11-02	DEPORTIVO	<button>Uredi</button>	<button>Izbriši</button>
CVETACNA JUHA	3	2017-11-02	Nektar Natura food	<button>Uredi</button>	<button>Izbriši</button>
Idrijski zlikrofi, 600g	9	2017-09-10	Tus Cash&Carry	<button>Uredi</button>	<button>Izbriši</button>
Fanta, 0,25l	9	2017-09-10	Tus Cash&Carry	<button>Uredi</button>	<button>Izbriši</button>
OMAKA ZA PEcENKO	9	2017-09-10	Nektar Natura food	<button>Uredi</button>	<button>Izbriši</button>
Cappuccino Tus, classic, 125g	9	2017-09-10	Tus Cash&Carry	<button>Uredi</button>	<button>Izbriši</button>
RDEcA PESA 4100 g	3	2017-09-10	DEPORTIVO	<button>Uredi</button>	<button>Izbriši</button>

Slika 2: Restavracija

### 3. TEHNOLOGIJE IN KNJIŽNICE NA STRANI ODJEMALCA

Uporabljene tehnologije in knjižnice na strani odjemalca.

#### 3.1. HTML: Hypertext markup language

Označevalni jezik za opisovanje strukture in vsebine dokumentov. Ni programski jezik. Definiran je z uporabo SGML. HTML dokument je mešanica vsebine in kontrolnih oznak (zaznamkov in atributov). Prva različica je bila predstavljena leta 1995, zadnja pa leta 2014. HTML5 je namenjen združitvi vseh možnih html sintaks pod eno streho. Nekatere novosti: semantično strukturiranje besedila, zmogljivejši elementi in oblikovanje, integracija multimedijskih vsebin, animacija na slikarski površini in podpora mobilnim napravam/

aplikacijam. HTML omogoča uporabo več različnih elementov, kjer vsak služi svoji nalogi. Delimo jih med vrstične in bločne.

### **3.2. CSS: Cascading style sheets**

Je stilski oziroma slogovni jezik. Omogoča enostavnejše oblikovanje spletnih strani, uniformen izgled spletišča in delitev programiranja in oblikovanja. Je sintaktični mehanizem za specifikacijo informacije o slogih. Predstavlja način kontrole predstavitve HTML dokumentov. Sintaksa je ločena od HTML, vendar lahko vsebovana v HTML dokumentih.

Prva različica je bila predstavljena leta 1996. Trenutno uporabljena različica je 2.1. CSS3 je trenutno v fazi razvoja. Uporabljajo se trije nivoji definiranja sloga: vrstične definicije, na nivoju dokumenta in zunanje slogovne pole.

### **3.3. JavaScript**

Je programski jezik na strani odjemalca. Programska koda je vdelana v dokument HTML in se interpretira na strani odjemalca. Je dinamično tipiziran jezik. Omogoča dostop do elementov dokumenta in dinamično spreminjanje. Je vse bolj popularen tudi na strani strežnika (Node.js).

Delimo ga lahko na tri dele:

- jedro: operatorji, izrazi, stavki, podprogrami,
- stran odjemalca: objekti za podporo nadzora odjemalca in komunikacijo z uporabnikom,
- stran strežnika: objekti uporabni pri spletnem strežniku, npr. podpora komunikaciji s podatkovnimi bazami.

Programski jezik je dinamično tipiziran, kar daje prostor za številne napake. Omogoča enostavno interakcijo z uporabnikom in dinamično interakcijo programerja z dokumentom HTML preko modela DOM. Je občutljiv na velike in male črke. JavaScript kodo lahko vključimo v HTML dokument direktno, ali pa vključimo zunanjo izvirno datoteko (priporočen način). Prvotno je bil namenjen izvajanju na odjemalcu (v brskalnikih). Namera je bila, da bi lahko razvijalec dostopal do posameznih elementov samega dokumenta in jih spreminjal.

### **3.4. JQuery**

Je knjižnica v jeziku JavaScript. Razvita je bila z namenom poenostavitve pisanja skript na strani odjemalca in poenostavitve pogostih opravil (navigacija po dokumentu, izbira in spreminjanje elementov DOM,...). Glavna prednost je, da je relativno majhna in lahka za uporabo. Omogoča premostitev razlik med brskalniki in ločitev kode JavaScript in HTML. Vse funkcije za delo z dogodki so lahko v JavaScriptu. V dokument se jo vključi kot katerokoli drugo skripto.

Pomembno je, da omogoča izvajanje na vseh brskalnikih. Ima uniformno implementirane funkcionalnosti. Vsak JQuery ukaz vrne nazaj objekt JQuery, zato se lahko izvajajo funkcije na vrnjenih objektih.

### **3.5. Bootstrap**

Je odprtokodna knjižnica oz. ogrodje namenjeno dizajniranju spletnih strani in aplikacij. Razvil ga je Twitter in je namenjeno osrednji spletnih aplikacij. Vsebuje dizajnerske predloge za tipografijo, komponente vmesnika in opcijske JavaScript razširitve. Je eden izmed najpopularnejših projektov na GitHubu. Omogoča enostavno doseganje odzivnega dizajna aplikacije, kar omogoča, da se lahko aplikacija prikaže in uporablja na različnih napravah. Je izjemno enostaven za uporabo in sledi pristopu "mobile first".

## **4. TEHNOLOGIJE IN KNJIŽNICE NA STRANI STREŽNIKA**

Uporabljene tehnologije in knjižnice na strani strežnika.

### **4.1. PHP: Hypertext Preprocessor**

Je skriptni jezik, ki se uporablja na strani strežnika. Koda je interpretirana, preden se pošlje odjemalcu. Ustvaril ga je Rasmus Lerdorf leta 1994. Namenjen je procesiranju obrazcev, branju datotek in dostopu do podatkovnih baz. Je dinamično in šibko tipiziran. Ker je interpretiran, je počasnejši kot prevajani jeziki. Rezultat izvajanja je HTML sintaksa.

### **4.2. Laravel**

Je brezplačno odprtokodno spletno PHP ogrodje. Ustvaril ga je Taylor Otwell in temelji na Symfony ogrodju. Namenjeno je razvijanju spletnih aplikacij, ki temeljijo na MVC arhitekturi. Razvijanje bistveno poenostavi. Omogoča uporabo svojega upravljalca paketov in dovoljuje več načinov dostopa do relacijskih podatkovnih baz. Ponuja tudi številna orodja in pripomočke, ki pomagajo pri razvijanju spletnih aplikacij. Ena izmed prednosti je tudi enostavna sintaksa.

### **4.3. MySQL**

Je sistem za upravljanje s podatkovnimi bazami. Je odprtokodna implementacija relacijske podatkovne baze, ki za delo s podatki uporablja jezik SQL. Deluje na principu odjemalec - strežnik, pri čemer lahko strežnik namestimo kot sistem, porazdeljen na več strežnikov. Obstaja veliko število odjemalcev, zbirk ukazov in programskih vmesnikov za dostop do podatkovne baze MySQL. Razvija ga Oracle Corporation. Napisan je v jezikih C in C++ in deluje na več različnih operacijskih sistemih.

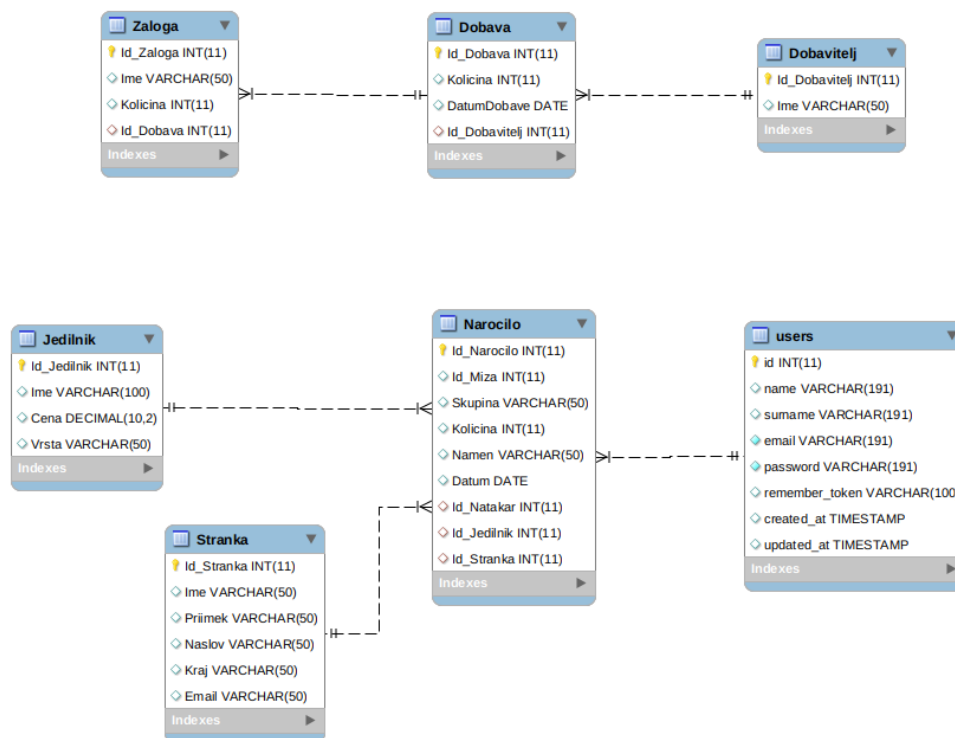
### **4.4. PhpMyAdmin**

Je brezplačno programsko orodje napisano v PHP jeziku. Namenjeno je upravljanju z MySql/MariaDB podatkovno bazo preko spletnega vmesnika. Omogoča uporabo grafičnega vmesnika, kot tudi poganjanje sql ukazov.



## 5. PODATKOVNA BAZA

Podatkovna baza je bila uporabljena pri 3. domači nalogi. Na lokalnem Linux računalniku sem naložil in nastavil MySQL strežnik. Posebno pozornost sem posvetil pravilni konfiguraciji varnostnih nastavitev in uporabnikovih pravic.



Slika 3: Shema podatkovne baze

Za dostopanje do podatkovne baze in upravljanje s podatki sem uporabil Laravel vgrajene funkcionalnosti. To mi je poenostavilo delo in hkrati poskrbelo za večjo varnost in preverjanje pravičnih vnosov.

## **6. ZAGOTAVLJANJE VARNOSTI**

### **6.1. 1. Domača naloga**

Preverjanje vnosov je na strani odjemalca. Uporabljena je HTML validacija z dodatnimi JavaScript preverjanji.

### **6.2. 3. Domača naloga**

Preverjanje vnosov je izvedeno na strani strežnika, kot tudi na strani odjemalca. Na strani odjemalca je uporabljena HTML validacija. Na strani strežnika pa Laravel validacija.

Uporaba Laravel ogrodja zagotovi višji nivo varnosti, brez dodatnih konfiguracij. Aplikacija je varna pred injekcijami SQL, XSS in številnimi drugi napadi.

Varnost zagotavlja tudi avtentikacija in avtorizacija, saj neprijavljeni uporabniki podatkov ne morejo spreminjati. Dodatna varnost je zagotovljena z uporabo kriptografske hash funkcije za shranjevanje gesel.

## **7. SKLADNOST Z GDPR**

### **7.1. 1. Domača naloga**

Za shranjevanje podatkov sta uporabljena "localStorage" in "sessionStorage". Hrani se ime, priimek uporabnika in njegov rezultat. Uporabnik o hrambi podatkov ni obveščen, saj to ni bila zahteva naloge. Aplikacija ni skladna z GDPR.

### **7.2. 3. Domača naloga**

Vsi podatki se hranijo v podatkovni bazi. Ob registraciji se mora uporabnik strinjati z zbiranjem in obdelavo podatkov, sicer registracija ni možna. Ob neprijavljeni uporabi aplikacije se podatki ne shranjujejo. Stikalo, kjer uporabnik potrdi pogoje uporabe spletne strani je na začetku označeno. Spletna stran podatke samo hrani in jih ne pošilja drugim. Podatki se hranijo v varnem okolju. Aplikacija ni skladna z GDPR, saj bi moral zbiranje podatkov uporabniku bolj natančno opisat.

## 8. ENO-STRANSKE APLIKACIJE

Gre za eno novejši tehnologij oz. način pri gradnji spletnih strani. Gre za aplikacije, ki se nanašajo na eno samo stran. Lahko gre tudi za nadgradnjo aplikacij, pri katerih gre za to, da se spletna stran prikazuje progresivno.

Enostranske aplikacije (SPA) so aplikacije ali spletne strani, ki komunicirajo z uporabnikom tako, da dinamično prepisujejo trenutno vsebino namesto nalaganja nove. S tem se izogiba prekinitvi uporabniške izkušnje. Vsa potrebna koda se lahko prenese z enim samim nalaganjem strani. Kontrola se nikoli ne prenese na drugo stran. Interakcija z SPA po navadi vključuje dinamično komunikacijo s strežnikom, ki se izvaja v ozadju.

Enostranske aplikacije imajo krajše nalagalne čase. Spletna stran se ne na novo naloži, samo vsebina se na novo prikaže. Omogoča nemoteno izvajanje aplikacije nekega uporabnika. Ne pride do prekinitve pozornosti. Spletna aplikacija se lahko začne obnašati kot namizna aplikacija. Večina oz. celotna programska koda se lahko prenese samo enkrat. Omogoča uporabo lenega nalaganja (prenašajo se samo potrebne komponente). Zmanjša se količina prenesenih podatkov.

Za večino funkcionalnosti SPA se uporablja JavaScript (uporabniški vmesnik, aplikacijska logika, komunikacija z strežnikom). Strežnik prejme zahteve, formira odgovor in ga posreduje odjemalcu. Odjemalec prejme kodo in poskrbi za ustrezno kopiranje odgovora na sami spletni strani.

Spletne aplikacije so postale popularne šele v zadnjih letih.

### 8.1. SPA tehnologije

Uporabljajo se tri pomembne tehnologije: HTML, CSS in JavaScript.

Za komuniciranje z strežnikom se lahko uporabljajo različne tehnologije:

- JavaScript ogrodja – različna orodja za razvoj SPA
- AJAX – tradicionalni asinhroni zahtevki
- Websockets – dvosmerna komunikacija
- Server-sent dogodki – tehnika za strežniško iniciacijo podatkovnega prenosa
- Brskalniški vtičniki – asinhroni klici na strežnik se lahko inicirajo tudi preko uporabe vtičnikov (zastarelo)

Uporabljajo se tudi različni formati prenosa podatkov: XML, JSON, HTML.

## 8.2. Primerjava MPA in SPA

Enostranske aplikacije so dobre, ker omogočajo dobro uporabniško izkušnjo. Po začetnem nalaganju je hitrejša nalaganje strani pri navigaciji po spletišču. Mogoče je zgraditi naprednejše in bogatejše aplikacije.

Vendar mora uporabnik obvezno vključiti JavaScript in te aplikacije bolj obremenjujejo napravo. Težje je načrtati in vzdrževati enterprise aplikacije, zaradi večjega števila generiranih datotek. Razvijanje teh aplikacij zahteva več znanja in so dražje za razvoj.

Velika večina spletnih aplikacij je večstranskih aplikacij in niso narejene v tehnologiji SPA. MPA so primernejše za monolitne aplikacije s sekcijo z enim stanjem. Začetno nalaganje je hitrejša. So varnejše, saj zahtevke obdeluje strežnik, kjer tečejo preizkušene rešitve. Zanje je lažje dobiti razvijalce, saj je to starejša tehnologija. Obstaja večja podpora za razhroščevanje v velikem številu IDE-jev. Slabost je, da se ne da ponovno uporabiti kode za mobilno verzijo aplikacije in seveda, ponovno nalaganje spletnih strani.

## 8.3. SPA ogrodja

Obstajajo številna ogrodja, ki omogočajo lahek razvoj SPA:

- AngularJS – nekdanj zelo priljubljen, njegova uporaba upada,
- Angular – novejša verzija AngularJS, vzvratno nekompatibilna,
- Aurelia – MVC ogrodje, tisto, kar naj bi bil Angular,
- Ember.js – je MVVM ogrodje, ki mogoča tudi razvoj mobilnih in namiznih aplikacij,
- Meteor.js – je odjemalec-strežnik ogrodje, namenjeno izključno razvoju SPA aplikacij,
- Vue.js – progresivno ogrodje za gradnjo UV,
- React.js – ogrodje za gradnjo UV.

## 9. AJAX IN WEBSOCKET

### 9.1. Ajax

Kratika pomeni Asynchronous JavaScript + XML. To je poimenovanje za način uporabe obstoječih tehnologij.

Na začetku je bila komunikacija med strežnikom in odjemalcem statična, koračna ( zahtevek in odgovor). Odjemalec je moral precej časa čakat, da se sprocesira zahtevek na strežniku. Vedno je bilo potrebno sprocesirat celotno stran, tudi če bi potrebovali le del.

Ajax je nastal zardi potrebe po asinhronih zahtevkih, ki omogočajo spletnim aplikacijam večjo odzivnost. Narasla je potreba po bogatih spletnih aplikacijah, ki imajo bogat uporabniški vmesnik in odzivnost podobno namiznim aplikacijam.

Ajax uporablja asinhrono zahteve. Omogoča prejemanje le delov dokumenta. To bistveno pohitri komunikacijo, ko je potreben izris le dela dokumenta.

Uporabnik pošlje asinhroni zahtevek na strežnik in lahko ševedno uporablja aplikacijo. Odgovor strežnika se doda v spletno stran.

Ajax omogoča:

- izvedbo asinhronih zahtevkov,
- posodabljanje le delov dokumenta namesto prenosa celega dokumenta,
- kombinacija tehnologij na strani odjemalca (JavaScript, XML, HTML, DOM, CSS).

Zahteveke izvedemo z objektom XMLHttpRequest ali raznimi JavaScript orodji.

Odzivnost strani je bistveno boljša, saj uporabnik ni zaklenjen v čakanje. Tudi poraba pasovne širine je bistveno manjša, ker ni potrebe po prenosu celotne strani.

Poznamo štiri faze delovanja:

1. Aplikacija izvede asinhroni zahtevek,
2. Strežnik obdela zahtevek in sporoča o stanju odgovora.
3. Strežnik vrne odgovor in kliče povratno funkcijo,
4. Odjemalec izvede posodobitev strani.

Kompleksne aplikacije zahtevajo več kode. Posledično to zahteva več strežnikov, kar vodi v več ranljivosti.

Možna napada sta injeciranje skript v spletne strani in injeciranje kode, ki spremeni potek izvajanja.

Ker se koda izvaja na odjemalcu, jo lahko kdorkoli vidi.

Obstaja problem nelegalnih asinhronih zahtevkov – asinhroni zahtevki so lahko samo s strežnika, s katerega je koda prišla ( per origin).

## **9.2. Websocket**

Je računalniški komunikacijski protokol, ki omogoča dvosmerno komunikacijo in temelji na TCP. Standardiziran je bil leta 2011. Narejen je bil za uporabo v spletnih brskalnikih in spletnih strežnikih, vendar ga lahko uporablja katerakoli strežniška ali odjemalska aplikacija. Omogoča realno časovno komunikacijo na in od strežnika, vzpostavljanje dolgo-časovnih povezav preko TCP utičnic in takojšne razpošiljanje sporočil z malo redundance. En strežnik lahko komunicira s več odjemalci.

WebSocket podpirajo vsi najpomembnejši brskalniki. Protokol zahteva, da tudi aplikacije na strani strežnika razumejo ta protokol in zagotavljajo podatke. Strežniki imajo lahko odprtih več povezav z različnimi odjemalci v kateremkoli trenutku.

Je enostaven za implementacijo in ne zahteva dodatne infrastrukture.

## **9.3. Primerjava Ajax in WebSocket**

Obe tehnologiji se dopolnjujeta in ne nadomeščata. Ajax se uporablja, ko so potrebne kratke povezave, na primer kratki klici spletnih storitev. Websocket se uporablja, ko je potreba realno-časovna funkcionalnost, saj omogoča nizko-latentno, dvosmerno komunikacijo preko enega samega kanala.