ВИСОКА ШКОЛА ЕЛЕКТРОТЕХНИКЕ И РАЧУНАРСТВАСТРУКОВНИХ СТУДИЈА

**ИлићНебојша**

**ВЕБ апликација за евиденциију пројектних активности**

**- завршни рад -**



Београд, новембар 2019.

Кандидат: **Небојша Илић**

Број индекса: **РТ-11/15**

Студијски програм: **Рачунарска техника**

Тема: **ВЕБ апликација за евиденциију пројектних активности**

Основни задаци:

**1. Опис клијентске и серверске стране апликације за евиденцију пројектних активности**

**2.Имплементација ВЕБ апликације за евиденцију пројектних активности**

**3.****Опис корисничког интерфејса**

Ментор:

Београд, новембар2019годинe.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

др Перица Штрбац, проф. ВИШЕР

САДРЖАЈ:

# УВОД

Апликација рађена као завршни рад има за циљ лакшу евиденцију активности унутар тима, одељења или било какве организације у којој је потребна нека врсра организације и евиденције. Апликација пружа табелу са насловима активности и коментарима на исте као и секцију за креиранје и брисанје корисника по потреби.

Апликација је реализована помоћу следећих технологија:

* PHP
* JavaScript / Ajax / jQuery
* CSS
* HTML
* SQL
* Bootstrap

Организација кода је MVC (Model View Controller), о овој организацији ће бити више у подглављу технологије.

# Технологије

Технологије за креиранје ове апликације су PHP, JavaScript / Ajax / jQuery, CSS, HTML, SQL, Bootstrap ове технологије као и организација кода MVC биће описани у овом подглавлју.

## PHP

PHP Hypertext Preprocessor je приграмски јежик оригинално дижајниран за веб дивелопмент. Оригинално је креиран од стране Расмус Лердорфа 1994 године. Жбог своје брзине, флексибилности и прагматичности, PHP покреће већину светских сајтова. Тренутно се користи 7.3 верзија која је избачена у децембру 2018 године.

## JavaScript / Ajax / jQuery

ЈavaScript je jedna od najbitnijih tehnologija World Wide Web-a. Омогућава интерактивне веб странице и самим тим је есенцијални део веб апликације. Већина сајтова као и вечина веб претраживача садрже JavaScript engine како би га извршавали.

Аjax је метода која омогућава моменталне поромене делова странице без поновног учитаванја стране “refresh”.

Док је jQuery библиотеке уграђена у ЈavaScript са већ уграђеним функцијама која олакшава програмеру коцанје кода и скраћује процес изведбе.

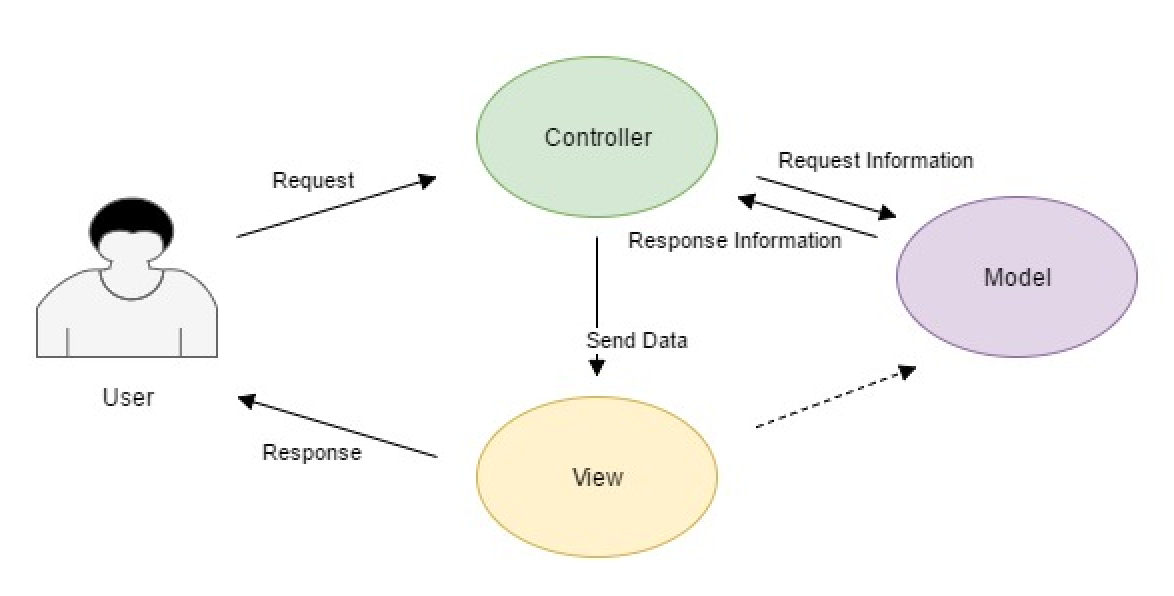
* 1. **HTML & CSS**

HTML (Hyper Text Markup Language) је описни језик специјално намењен за опис веб страница. Помоћу њега се једноставно могу одвојити елементи као што су наслови, параграфи, цитатии слично. Поред тога, у HTML стандард су уграђени елементи који детаљније описују сам документ као што су кратак опис документа, кључне речи, подаци о аутору и слично. Ови подаци су општепознати као мета подаци и јасно су одвојени од садржаја документа.

Док је CSS (Cascading Style Sheets) језик који је намењен да опише HTML или XML. Описује како елементи треба да се понашају на екрану, на папиру, у говору, или другим медијима

**2.4. MVC**

MVC (Model View Controller) се односи на организацију кода унутар пројекта. Као што име каже састоји се од 3 компоненте а то су Model, View и Controller. На (Слика 2.4.1) приказано је како то функционише. View је оно што корисник види кад приступи сајту. На било коју акцију (захтев корисника ) View шаље захтев моделу који комуницира са контролером и обрађује податке које шаље назад у View и корисник добија резултат акције коју је извршио нпр захтев друге странице, login, logout, неке податке из базе података…

Слика 2.4.1 – модел понашанја MVC-a.

### 

### Опис клијентске и серверске стране апликације за евиденцију пројектних активности

Клијентска страна се састоји од следећих страница, hоme (почетна страна), help, login, dashboard, users, create user.

Home страница је почетна (прва која се приказује уласком на сајт) она садржи податке о креатору овог рада као и назив пројекта.

Help страница предвиђена за помоћ око навођења кроз user interface на сајту која у овом случају садржи комплетну докумантацију приказану преко pdf-a.

Login страница садржи форму за логовање и тиме омогућава постојећим корисницима приступ на остале странице у зависности од улоге (права) налога.

Постојеће улоге су: owner, admin и default. Default има права за приступ dashboard страници на којој му је омогућено да дода нови коментар са произвољним насловом и да едитује своје коментаре. Admin има поред ових права да едитује било који коментар као и да га обрише, поред тога има приступ и осталим страницама а то су users и create user. Он је у могућности да направи новог корисника и има евиденцију у остале кориснике. Док оwner има поред свих ових права могућност да брише кориснике, да им промени права из default у admin и обрнуто и могућност да едитује постојеће кориснике тј. да промени име или шифру по потреби. Owner не може да се направи преко сајта, због сигурности та улога је hard-code-ована у базу како би се отежао напад на страницу тако што би нападач променио свом налогу улогу у највишу могућу.

За све странице су коришћени исти header и footer стим што се header разликује у случајевима да ли је корисник улогован или не као и од права које његов налог има.

## Имплементација ВЕБ апликације за евиденцију пројектних активности

Као што је већ поменуто апликација је урађена преко MVC архитектуре. Организацију фајлова у пројекту можете видети на (Слика 4.1). Као што може да се примети поред фолдера са ове три компоненте постоје и дуги фолдери, у којима су фајлови који употпуњују фникционалност и додатно смањују потребу за поновним куцањем кода и олакшавају имплементацију.

Фолдер db садржи копију базе за поновно подиза ње у случају промене сревера или губитка података услед хакерског напада на сајт. Back-up се не ради аутоматски, за то је задужен власник или друго лице коме је задата та улога као и приступ базама које су live.

Фолдер libs као што само име каже библиотеке садржи као и такав фолдер код било ког другог framework-а овоф типа. А то су фајлови који обухватају неку целину функционалности, у овом случају су то на пример модел и контролер. Чија улога је да за одговарајучу страницу пронађу одговарајуће компоненте и споје их у целину како би се добила одговарајућа страница. Овај метод омогућава да ови фајлови само траже руте о обрађују их уместо да се цео тај процес ради појединачно за сваку страницу. Самим тим добијамо смањење количине кода и лакшу рефакторизацију јер је у случају жељене промене потребно променити код у само овим фајловима уместо у фајловима за сваку страницу. Начин и код на који је ово изведено можете видети испод на примеру контролера и модела (Слика 4.2 и Слика 4.3).

Поред ових фајлова можемо видети и фајл View који обавља исту улогу као и предходна 2 само што је његова улога да обради view тј. део странице који је најближи кориснику, пронађе руте и рендерује целину. Поред класичних фајлова који представљају садржај странице постоје универзални header и footer који су учитани у сваку страницу. Овај приступ такође смањује количину кода као и могућност лакше измене тај пример можете видети у header фајлу. Header садржи навигациони бар који у зависности од тога да ли је корисник улогован или не и у зависности од права налога приказује различите елементе у навигационој листи (Слика 4.4)

Ту су такође и фајлови: database, hash и session.

Database служи за комуникацију са базом, што подразумева конекцију са њом као и функције за интеракцију са њом. Конекција је извршена на следећи начин (Слика 4.5). Овај фајл наслеђује PDO о коме ће бити речи даље у тексту. Уз помоћ делова који су дефиниссани у том фајлу овде је довољно да у конструктору проследимо податке о бази који су потребни за успостављање везе.

У овом фајлу су такође дефинисане функције које стандардизују и олакшавају све акције које могу да се врше над базом (select, insert, update, delete)(Слика 4.6). Овим је постигнуто далеко лакше извођење ових акција јер је за било коју од наведених довољна само једна линија кода која дефинише тачно податке који су жељени на том месту. Поред ових основних функција постоје и додатне за селекцију сви података тог типа као и за селектовање само једног реда у табели (један коментар или један корисник) што олакшава имплементацију функција за edit и delete.

Session је задужен за управљање сесојама односно од тренутка кад се корисник улогова до његовог logout-a. Поред овога подаци из кључева сесија се могу користити и у друге сврхе као на пример провера типа налога и самим тим провера права које тај налог садржи. Или порука за добродошлицу коју можете видети на dashboard stranici nakon uloga.

Hash (Слика 4.7) скужи за сигурност. Његова улога је контролосање лозинки. Лоша пракса је складиштити лозинке у базу у формату у ком су унете јер у том случају свако ко има приступ бази има потпун приступ сваком од налога. Због тога постоји hash чија је улога да коришћењем неког алгоритма измени садржај и на тај начин га убаци у базу, наравно поред овога мора да буде у стању да тај унос врати у првобитно стању када корисник покуша да се улогује и унесе лозинку у првобитном стању. Следећи проблем настаје због тога што су сви лагоритми за ове потребе познати и јавни тако да већина алгоритама може веома лако да се пробиоје па чак и преко разних веб апликација које обављају тај посао. Због тога након хешовања добра је пракса да се том герерисаном низу карактера дода део (salt) који је такође један такав низ и после тога уради још један hash пожељно другим алгоритмом. Salt део је дефинисан у другом фајлу о коме ће бити реч даље у тексту. Ево и један пример како изгледа лозинка када је корисник уноси и како изгледа у бази после овеих акција.

И последњи али и не мање битан је bootstrap фајл. Далеко од тога да је није битан јер овај фајл седржи неке од најбитнијих ствари. Овде се врши коначно рендеровање и спајање у једну компоненту. Поред овога задужен је за рутирање као и за генерисање URL-a. Генерисање URL-a је битно за почетак да би с избациле празнине или непожељни карактери. Такође промена имена директоријума тако рећи скривена навигација како не би била присутна дирактна имена фајлова који се користе на тој страници што знатно отежава нападачу да са лакоћом види начин на који сајт ради и то искористи за напад. Поред ових естетских и сигурносних делова обававља редирекцију као на пример ако корисник укуца име странице која не постоји вратиће га на почетну страницу(Слика 4.8). Или ако покуша да приступи страници или страницама за која нема права то јесте users и dashboard без налога или са налогом који нема право за users. Ова библиотека ће га послати на error страницу(Слика 4.9). Делове кода можете видети испод (Слика 4.10).

U фолдеру public се налазе CSS и JavaScript фајлови. Сав дизајн сајта који није урађен преко bootstrap-a и слично је одрађен овде. Дизајн је одвојен на овај начин због лакше претраге и промене у случају промене дизајна. У фикдеру public постоји још једна подела по фолдерима како би било још прегледније а то су офлдери за дизајн и логику.

Што се тиче дизајна већина је одрађена преко bootstrap-a у овом фолдеру су минорне измене како би дизајн довели на одговарајући ниво.

Логика је одрађена преко Ajax-a и JQuery-ja за разлику од дизајна већина логике је ручно направљена за ове потребе. А то су на пример провере да ли су подаци добро унети приликом прављења налога или додавања коментара. Поред тога можете приметити анимацију приликом брисања података и pop-up window приликом измене постојећег коментара. Међутим приликом измене корисника уместо pop-up window-a постоји нова страница. Разлог за то је како би преко овог задатка (ове веб апликације) показао различите начине да се одради једна те иста ствар. Примере кода из овог фајла можете видети испод за брисање података (Слика 4.11), за pop-up, (Слика 4.12), за валидацију (Слика 4.13).

Фолдер utility садржи само један фајл за аутентификацију који је поменут код сесија чија је улога да прати да ли сесија постоји или не и у случају да је сесија истекла врати корисника на login страну јер би у супротом на покушај било које акције био пребачен на error стану (Слика 4.14). Овај фолдер као што и само име каже може да садржи бико који вид портабилног (корисног) кода који надгледа целу апликацију или део ње који такође може бити искоришћен како би допунио неку функцију која ће обављати задатак у целини.

Поред ових фолдера ту су ифајлови .htaccess, config, index.

Index je главна страница на којој су ујључене све компоненте које су потребне како би се било која страница формирала (Слика 4.15). А коментару овог дела кода каже да је боља пракса користити autoloader који би аутоматизовао овај процес и у случају додавања нових фајлова у било ком од већ дефинисаних фолдера аутомацки би био пронађен и додат у ову рутину без потребе да се на овом месту ручно додаје.

Config као што и само име каже садржи конфигурације (Слика 4.16) као што су руте до root-a (корена) апликације, поред њега видимо и руту до фолдера libs. За потребе ове апликације ово је било довољно ако наравно могуће је на овај начин дефинисати све путање што би олакшалп даље рутирање. Поред тога имамо конструктор у коме су прослеђени подаци за повезивање са базом података. И константе у виду кључева за хешовање лозинки као игенерални кључ који тренутно нема улогу али је дефинисан у случају да је потребно хешовати још неки податак. Ово је изведено на овај начин зато што кључеви морају остати исти јер у случају промене систем не би био у могућности да пронађе ни један податак који одговара унетом. Ови кључеви се користе у salt делу о коме је било речи у делу како функционише хешовање лозинки и зашто је то добра пракса.

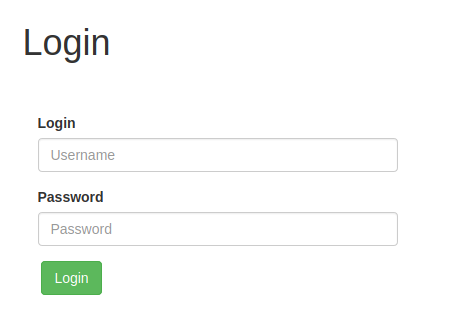
И .htaccess је конфигурацион фајл за потребе веб сервиса који покрећу софтвер Apache Web Server-а. Могу се користити за измену конфигурације у софтверу Apache Web Server-а како би се укључиле или искључиле додатне функције и додаци које Apache Web Server може да понуди (Слика 4.17).

1. **Опис корисничког интерфејса**

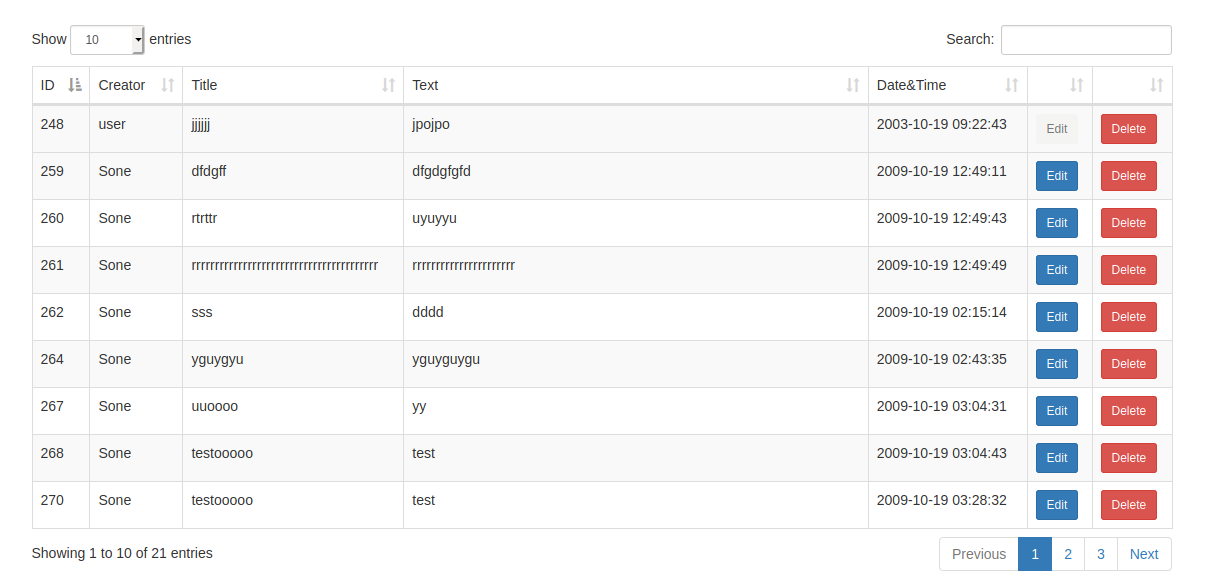
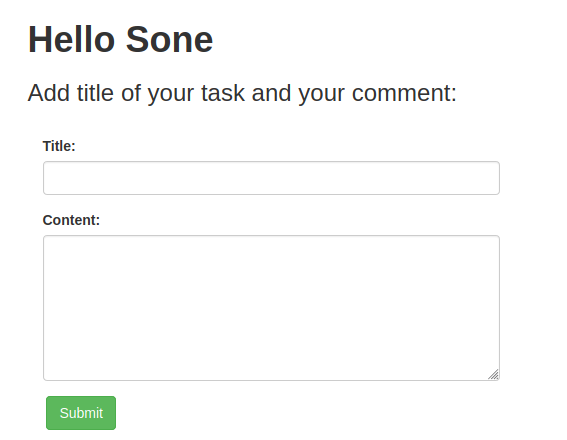
При приступу веб апликацији корисник се среће са почетном страницом (home), на којој се налазе подаци о креатору и пројекту (Слика 5.1).



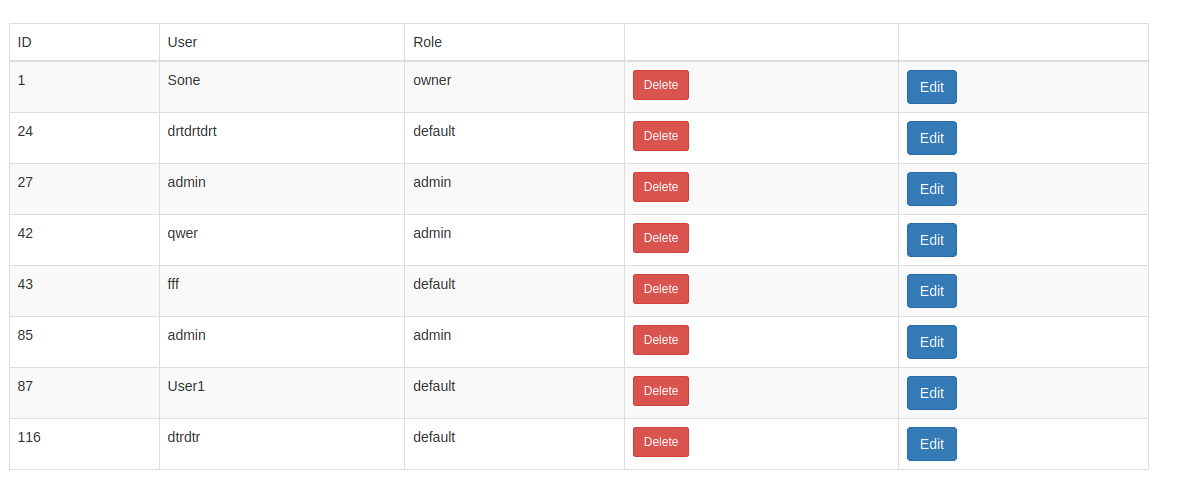
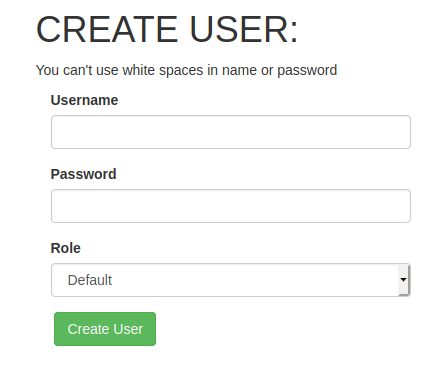
Поред ове странице корисник има приступ help страници која садржи потпуну документацију у PDF формату. Док Login Страница садржи форму за логовање за постојеће кориснике. Корисник није у стању да самостално направи налог, како би дошао до налога потребно је да ту уради неко од админа или власник. Испод овог текста можете видети слику са изгледом login страницe (Слика 5.2). Help страница ће бити финализирана тек са завршетком ове документације која ће бити убачена.



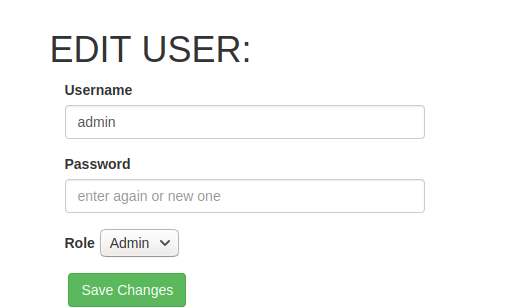
Након успешног логовања као што можете видети (Слика 5.3.1 и 5.3.2) приказана је страница dashboard на којој корисници имају евиденцију у табелу са коментарима и насловима активности које је та особа лично или било која друга особа која је укључена у тај пројекат или било какву организацију за коју је потребна евиденција додала. Такође има опцију да сортира табелу по било којим од датих параметара као и опцију search у којој може укуцати слово, слог или кључну реч по којој жели да претражи табелу. Наравно поред увида у већ постојеће податке може додати нове, уређивати своје старе уносе као и опцију брисања истих.



У зависности од типа налога админ или власник имају приступ страници Users на којој имају опцију за креирање новог налога као и табелу са већ постојећим налозима. Постојећи налози могу бити преуређени по потреби или промењена шифра у колико је корисник захтевао промену или је заборавио стару. Поред ове опције постоји опција за брисање налога. (Слика 5.4.1 И Слика 5.4.2).



За уређивање налога постоји још једна страна (Слика 5.5) по приступу страници поље за корисничко име је већ испуњено старим именом са могућношћу да се промени и поље за шифру као и drop-down мени за промену улоге по потреби.



# ЗАКЉУЧАК

Сваки framework као и овај значајно убрзавају процес израде апликације, смањују понвно куцање и олакшавају имплементацију истог кода на друге пројекте. Захваљујући библиотекама и класама комплетна логика и функционалност могу да се имплементирају на различитим местима уз само једну линију кода. Као и конфигурације за комуникацију и повезибањем са базом података и опцију за писање ручних упита (Api). Који даље поједностављују писање кода.

# ИНДЕКС ПОЈМОВА

**Д**

Doctorine 1, 12, 13, 14, 15

**Ј**

JavaScript 3, 19, 20

**К**

Контролер 1, 2, 5, 8, 9, 10, 16, 17, 18, 30

**М**

MVC 2,

**П**

PHP 1, 3, 4, 5, 7, 8, 9, 12, 13, 14, 16, 17, 32

Пакет менаџер 4

**Р**

Рутирање 1, 5

**С**

Sass 1, 20, 21

Symfony 1, 2, 3, 4, 5, 6, 12, 17

Сервис 1, 10, 16, 17

**Т**

Twig 3, 4, 6, 7, 8, 9, 10, 12, 17, 18, 19, 20

**Ф**

Framework 1, 3, 4, 9

Форма 1, 2, 4

**Х**

HTTP 1, 4, 5, 20

HTML 5, 6, 7, 8, 15, 16, 17

33

# ЛИТЕРАТУРА

[1] Б. Раковић, *Електроника – Линеарна интегрисана кола*, Грађевинска књига, Београд, 1983.

[2]D. Davis, E. Patronis Jr, *Sound system Engineering,*pp. 202-212, Focal Press, 2006.

[3] S.H. Linkwits, "Active Crossover Networks for Noncoincident Drivers", *J. Audio Eng. Soc.*, vol. 24, no. 1, pp. 2-8, 1976.

[4]J.J. Lopez, B. Pueo and M. Cobos, “Conventional and distributed mode loudspeaker arrays for the application of wave-field synthesis to video conference” *124th Conv. Audio Eng. Soc.*,Amsterdam, May 2008.

[5] J. Murray, „A Primer On Ethernet Cabling For Digital Audio” <http://www.prosoundweb.com/article/a_primer_on_ethernet_cabling_for_digital_audio/>, преузето: април 2016.

# Прилози

# изјава о академскoј честитости

|  |  |
| --- | --- |
| **Студент (име, име једног родитеља и презиме):** |  |
| **Број индекса:** |  |

Под пуном моралном, материјалном, дисциплинском и кривичном одговорношћу изјављујем да је завршни рад, под насловом:

1. резултат сопственог истраживачког рада;
2. да овaj рад, ни у целини, нити у деловима, нисам пријављиво/ла на другим високошколским установама;
3. да нисам повредио/ла ауторска права, нити злоупотребио/ла интелектуалну својину других лица;
4. да сам рад и мишљења других аутора које сам користио/ла у овом раду назначио/ла или цитирао/ла у складу са Упутством;
5. да су сви радови и мишљења других аутора наведени у списку литературе/референци који је саставни део овог рада, пописани у складу са Упутством;
6. да сам свестан/свесна да је плагијат коришћење туђих радова у било ком облику (као цитата, прафраза, слика, табела, дијаграма, дизајна, планова, фотографија, филма, музике, формула, вебсајтова, компјутерских програма и сл.) без навођења аутора или представљање туђих ауторских дела као мојих, кажњиво по закону (Закон о ауторском и сродним правима), као и других закона и одговарајућих аката Високе школе електротехнике и рачунарства струковних студија у Београду;
7. да је електронска верзија овог рада идентична штампаном примерку овог рада и да пристајем на његово објављивање под условима прописаним актима Високе школе електротехнике и рачунарства струковних студија у Београду;
8. да сам свестан/свесна последица уколико се докаже да је овај рад плагијат.

У Београду, \_\_. \_\_. 201\_.године

Својеручни потпис студента

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_