**Имена:**  Кристина Димова **фн:**  *61886* **Имена:**  Невена Гаджева **фн:**  *61938* **Имена:**  Радослав Гайдаров **фн:**  *61892* **Имена:**  Румен Чолаков **фн:**  *61921*

**Начална година:** *2015* **Програма:** бакалавър, (СИ) **Курс:** 3   
**Тема:** Система за управление на избираеми дисциплини  
**Дата:** 2018-07-01 **Предмет:** wwwTech2017\_18\_10ed\_SI\_summer **имейл:** [krisi29dimova@gmail.com](mailto:krisi29dimova@gmail.com) [v3n17oy@abv.bg](mailto:v3n17oy@abv.bg) [radgajdarov@gmail.com](mailto:radgajdarov@gmail.com) [kelanved96@gmail.com](mailto:kelanved96@gmail.com)

**преподавател:** доц. д-р Милен Петров

# ТЕМА: Система за управление на избираеми дисциплини

## 1. Условие

Да се създаде уеб базирана система, която предоставя следните функционалности:

* генериране на различни изгледи на списък с избираеми дисциплини(по семестър - зимен, летен, по име на дисциплината, по име на лектор, по кредити, по курс, по категория, по специалност, по рейтинг на дисциплината)
* преглед на пълно описание на дадена избираема дисциплина
* предлагане на нови избираеми дисциплини
* записване на студенти за избираеми дисциплини (от самия студент или от лектора, който води дисциплината)
* отписване на студенти от избираеми дисциплини (от самия студент или от лектора, който води дисциплината)
* нанасяне на оценки на студенти, които са записали дадена избираема дисциплина
* комуникация между лектори и студенти, изпращайки си съобщения в системата
* преглед на получени и изпратени съобщения
* генериране на справки за студенти -  предоставят информация кои дисциплини е записал съответният студент, кога ги е записал, кредитите, които му носят, и оценката, която е изкарал по съответната избираема
* генериране на справки за лектори - разпределение на студенти по курсове, специалности, които са записали избираема дисциплина, която се води от даден лектор
* предоставяне на разписание за провеждане на избираемите дисциплини (ден, час и място на провеждане на занятията)
* предоставяне на разписание за провеждане на изпитите по съответните избираеми дисциплини
* редактиране на избираема дисциплина(описание, теми, препоръчителна литература, кредити, лектор, категория, курс, за който е предназначена)
* редактиране на профил - потребителите могат да променят личните си данни, които се пазят в системата
* регистриране на потребител
* изтриване на потребители от системата (администратор)
* добавяне на потребители в системата (администратор)

## 2. Въведение

Разработената система е уеб базирана. Тя е предназначена за използване в университети от лектори и студенти, за   управление на всички аспекти, свързани с избираеми дисциплини. Има за цел да подпомогне процеса в следните насоки: записване, отписване, нанасяне на оценки, предложения за нови дисциплини, комуникация между потребителите, справки за избираемите, график на провеждането им и изпитите, които трябва да се състоят. Системата позволява и генериране на различни изгледи на списък с избираеми дисциплини по различни признаци.

Проектът е публикуван на <https://hci2018.000webhostapp.com/index.html>.

## 3. Теория

Основната част от системата се базира на **ООП** модела и е с **MVC** архитектура, а модулите на отделните типове потребители (администратор, лектор и студент) са реализирани на принципа **SPA** (Single Page Application). **ООП** моделът и **MVC** архитектурата позволяват бързо и лесно подобряване, редактиране, разширяване и преизползване на функционалностите и на части от кода. Този тип архитектура е използвана за реализиране на модулите на отделните типове потребители и за управление на избираемите дисциплини. Всеки един модул разполага с т.нар. модел, който си комуникира с базата данни, контролер, който осъществява връзка между модела и потребителя, и изглед, чрез който потребителят взаимодейства със системата. Използвайки подхода **SPA**, взаимодействието с потребителя се осъществява чрез динамично пренаписване на текущата страница, вместо да се зареждат цели нови страници от сървъра.По този начин се предотвратява прекъсването при преходите между последователните страници, което прави приложението по-скоро като десктоп приложение. Нужните ресурси се извличат от базата данни и се зареждат динамично и се добавят към страницата, ако е необходимо, обикновено в отговор на действията на потребителя. За целта се използват технологията **AJAX** и форматът **JSON**. С тяхна помощ се осъществява връзка със сървъра, без това да е видимо за потребителя. За обработването на заявките от страна на сървъра се използват т.нар. суперглобални променливи. Това са променливи, които могат да бъдат достъпени от всяка функция, всеки клас или файл от страна на сървъра, без да се прави нищо специално.  Суперглобалните променливи, които са използвани основно са: **$\_POST**, **$\_GET** и  **$\_SESSION**, като първите две са използвани за обработка на заявките към сървъра, а последната - за управление на **сесиите**, с помощта на които се съхраняват временни данни, когато потребителят взаимодейства със системата. За тази цел се използват и т.нар. **cookies** (**бисквитки**). Разликата е, че в случая **сесиите** се използват за съхранение на ‘чувствителни’ данни за потребителя, които не е добре да бъдат достъпвани от него, а **бисквитките** - за данни, които не засягат директно данни, които са важни за потребителя и биха довели до злоупотреби като потребителско име, парола, email и т.н. За защита на данните на потребителите всички въведени от тях данни се валидират, а паролите се хешират с алгоритъма **sha256**.

## 4. Използвани технологии

За реализирането на проекта са използвани следните технологии:

* **XAMPP (Apache 2.4.33, MariaDB 10.1.32, PHP 7.2.5, phpMyAdmin 4.8.0.1)**
* **JavaScript**
* **AJAX**
* **HTML5**
* **CSS3**

Причини за избор на технологиите:

* **HTML** и **CSS** са широко разпространени и основа на почти всяко уеб приложение.
* **XAMPP** предоставя всички инструменти, от които имаме нужда за създаване на локален сървър и поддръжка на уеб приложението
* **XAMPP** е cross-platform и лесен за инсталиране
* **JavaScript** ни предоставя възможност за лесно създаване на динамично съдържание
* **AJAX** ни позволява да изпращаме до и получаваме от сървъра данни асинхронно

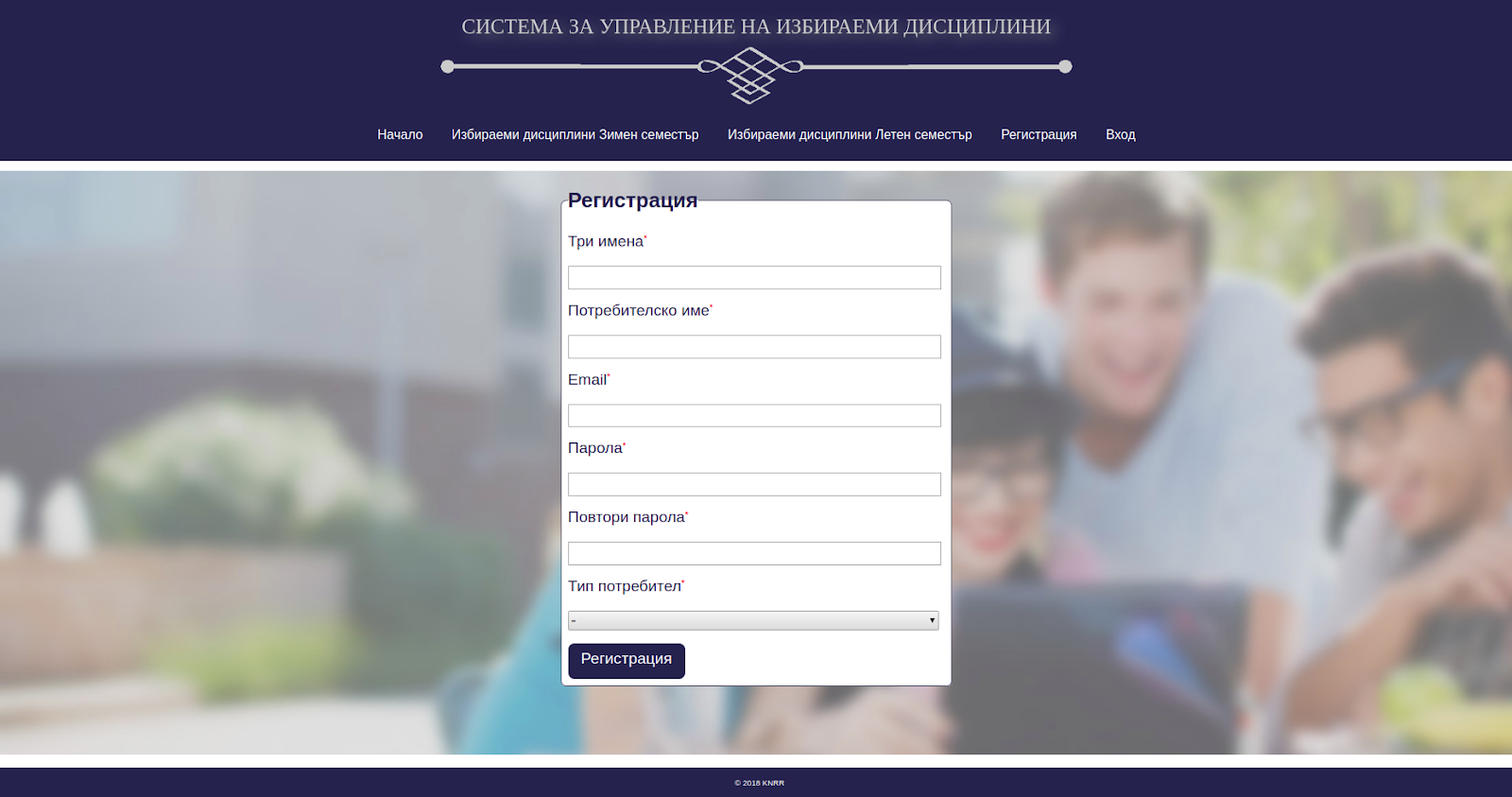
## 5. Инсталация и настройки

1. Предварителни условия
   1. Изтегляне на проекта
   2. Инсталиран **XAMPP** или аналогичен **web server solution stack package**
2. Инсталация
   1. Главната папка, съдържаща проекта, се добавя в **htdocs** директорията на **XAMPP**.
   2. Скриптът **db.sql**, който се намира в директорията config, съдържа заявките, нужни за създаване на базата от данни, а **db\_data.sql** съдържа примерни данни. Използвайки заявките от тези два файла, през интерфейса на **PHPMyAdmin** или през терминала може да бъде създадена базата от данни, нужна за работата на приложението.
3. Настройки
   1. Информацията за връзка с базата се конфигурира, като във файла **config/config.ini** бъдат зададени желаните стойности на отделните параметри.

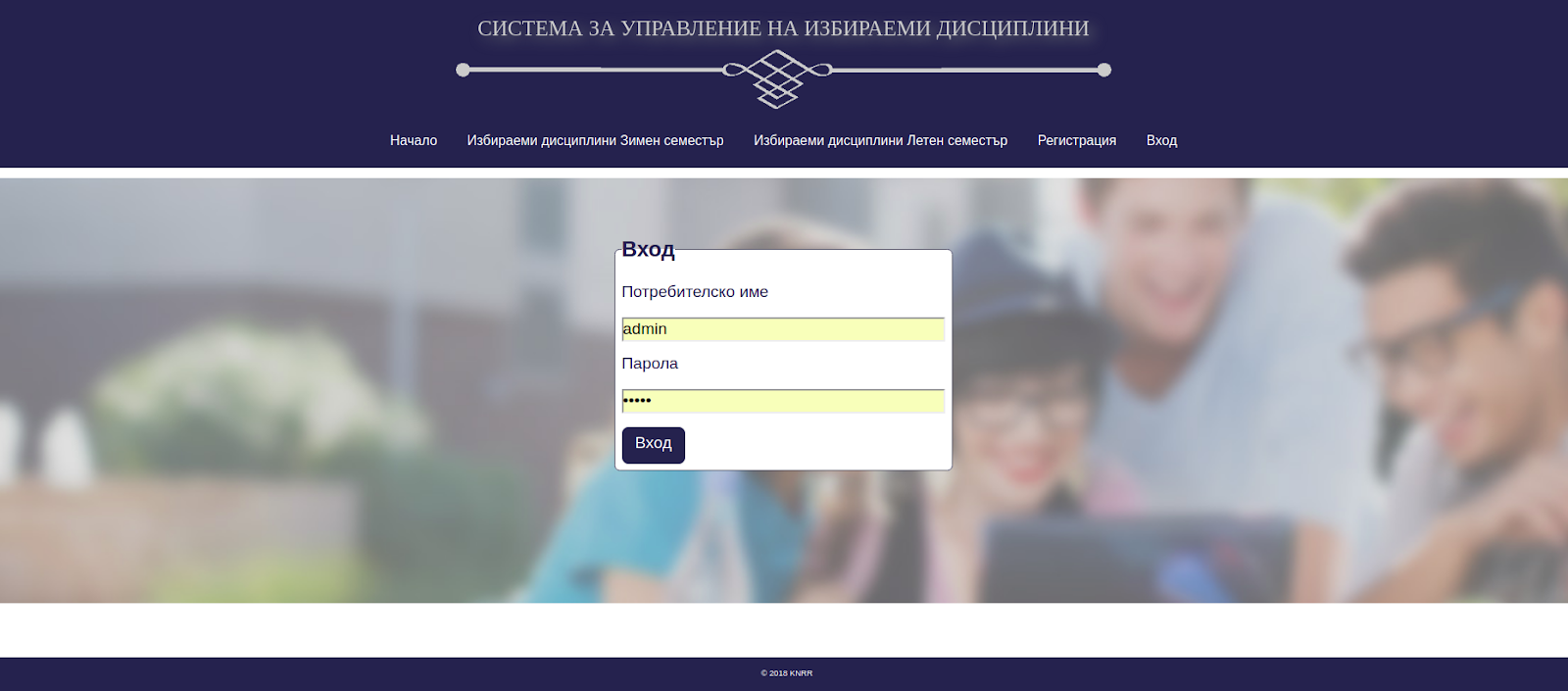
## 6. Кратко ръководство на потребителя

1. Начален екран

Началният екран е първото нещо, което вижда всеки потребител. За да се възползва от функционалностите на системата, потребител трябва да създаде регистрация или да се впише в нея.



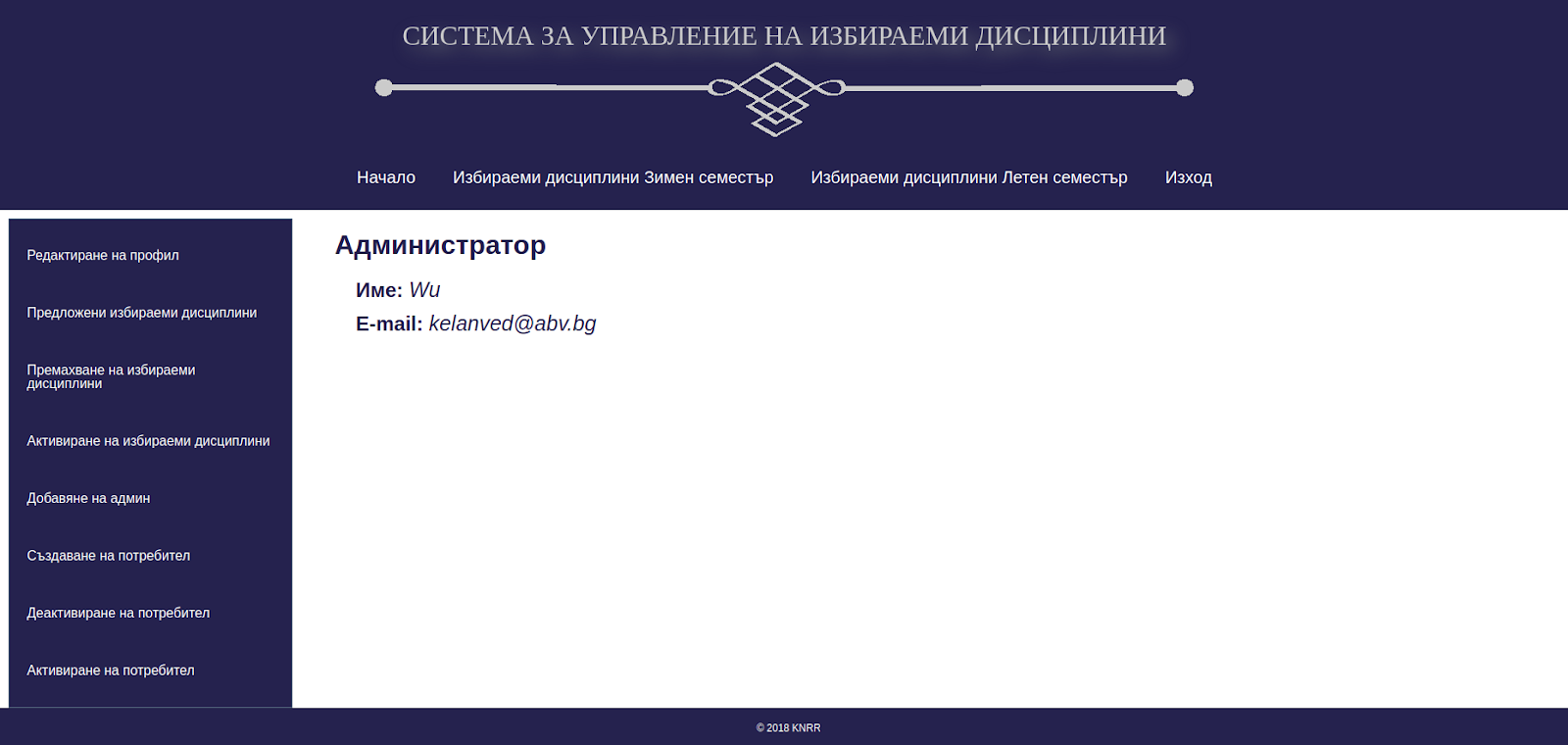
За създаване на нова регистрация потребител трябва да кликне върху линка “Регистрация”, който се намира в горния десен ъгъл(предпоследният от дясно наляво на хоризонталния navigation bar). След това ще се появи форма, в която потребителят трябва да попълни своите входни данни, които варират в зависимост от избрания тип потребител(Студент, Преподавател).



    В случай че потребителят вече притежава регистрация в системата, той може да се впише в нея чрез

Линка “Вход”(последният от дясно наляво на хоризонталния navigation bar). Потребителят трябва да въведе своите “Потребителско име” и “Парола” във формата. В случай на неуспех(въведени невалидни данни) на потребителя ще бъде изведено съобщение.

1. Работен плот



    След вписване в системата потребителят ще бъде въведен в работния плот. Той е съставен от основна част и страничен navigation bar, разположен в лявата част. В side bar-ът е разположен списък с функционалностите, които са на разположение на потребителя, като те са различни за различните типове потребители(Администратор, Студент и Преподавател). В основната част се изпълняват и изобразяват избраните от потребителя функционалности. При клик върху линк “Начало”(Горе най-ляво върху navigation bar-а) потребителя се отвежда в начално положение, като върху основната част от работния плот се изписва информация за него като име и e-mail.

1. Избираеми дисциплини



        При преглед на избираема дисциплина информацията за нея се отваря в нов прозорец. В лявата част на прозореца е разположен navigation bar, в който се съдържат линкове към основните части на страницата. Освен преглед на информацията за дадената избираема, потребител може да гласува за нея(с иконите “Палец нагоре” и “Палец надолу”), както и да остави коментар за въпросната дисциплина. Коментари за дисциплината са налични при кликване върху бутона “Коментар”.

## 7. Примерни данни

## Примерни данни за вход в системата, както и различните типове потребители, могат да бъдат намерени във файла config/loginData.txt, а примерни данни за потребители и избираеми дисциплини има във файла config/db\_data.sq.

## l8. Описание на програмния код

* **HTML** - Файловете разположени в папката **html**. Те съдържат основната структура на интерфейса на приложението.
* **CSS** - Файловете, разположени в папката **css**, те съдържат кода, отговарящ за стилизирането на интерфейса на приложението
* **JavaScript** - **JavaScript** файловете са разположени в папката **js**. В проекта **JavaScript** се използва за генериране на динамично съдържание като създаване на подходящи колони и форми за попълване на данни.
* **PHP**
  + Връзка с база данни - една от функциите, която **PHP** използва в проекта е връзка с базата данни. Чрез **PHP** извличаме и записваме информация в базата данни.
  + Връзка между **front-end** и **back-end** - в проекта **PHP** изпълнява и роля на controller, като приема и обработва информация пристигаща от интерфейсът и я препраща към основната логика на приложението, която също се обработва използвайки **PHP**
  + Запазване на информация за потребител - в проекта **PHP** се ползва за съхранение на информацията за текущия потребител, която се използва от главната логика на приложението
* **SQL** - **SQL** файловете са разположени в папката **config**. Скриптовете в тях се използват за създаване на базата от данни нужна за работата на приложението и за въвеждане на примерни данни.

## 9. Приноси на студента, ограничения и възможности за бъдещо разширение

Като цяло работата по един цялостен проект в група беше доста предизвикателна, трудна, но и интересна задача. Работата по проекта бе разделена по следния начин:

* **Кристина**
  + Основната работа върхуфункционалностите,  предназначени за потребителите, които са лектори
    - Редактиране на личните данни
    - Отписване на студенти от избираема дисциплина
    - Записване на студенти за избираема дисциплина
    - Нанасяне на оценки на студенти, които са записали дадена избираема дисциплина
    - Изпращане/получаване на съобщения до/от студенти
    - Справки за броя записани студенти за дадена избираема дисциплина
    - Добавяне на предложения за нови избираеми дисциплини
    - Предоставяне на график за провеждане на изпитите по избираеми дисциплини, водени от даден лектор
    - Предоставяне на разписание на часовете и залите за провеждане на избираемите дисциплини, водени от даден лектор
    - Редактиране на дадена избираема дисциплина
* **Невена:**
  + цялостно изграждане на front-end частта на проекта
  + основна работа върху функционалностите, предназначени за потребителите, които са студенти
    - редактиране на данните на студента (email и парола)
    - записване на избираема дисциплина при наличие на активна кампания
    - отписване от избираема дисциплина при наличие на активна кампания
    - предлагане на нова избираема дисциплина
    - преглед на предложенията за нови избираеми дисциплини
    - коментиране и гласуване за предложение за нова избираема дисциплина
    - предоставяне на разписание на часовете и залите за провеждане на записаните от студента през текущия семестър избираеми дисциплини
    - предоставяне на график за провеждане на изпитите по записаните от студента през текущия семестър избираеми дисциплини
    - предоставяне на справки за записани избираеми дисциплини, получени оценки и кредити и общ брой събрани кредити от записаните избираеми дисциплини
    - изпращане на съобщение до лектор
    - предоставяне на списък с изпратени и получени съобщения, както и преглед на тяхното съдържание
  + регистрация, вход и изход
  + списък с избираеми дисциплини за зимен и летен семестър
  + страница с пълно описание на дадена избираема дисциплина, включваща всички функционалности (гласуване, коментари, връщане назад)
  + част от структурата на базата данни
  + въвеждане на примерните данни в базата от данни
  + разработване на класа Database, който осъществява връзка с базата от данни

В изброените по-горе части от проекта е използван MVC модел на архитектурата с цел по-лесно обновяване и надграждане на функционалностите, както и ООП и SPA подход на разработка. Единственото по-съществено ограничение е, че потребителският интерфейс на системата не е responsive. Това обаче би могло да се осъществи в бъдеще.

* **Радослав**
  + Работа по функционалностите на администратора
    - Преглед на личен профил
    - Одобряване на предложени избираеми дисциплини
    - Премахване на избираеми дисциплини от списъка с активни дисциплини
    - Добавяне на избираеми дисциплини от списъка с активни дисциплини
    - Създаване на нови администраторски профили
    - Създаване на нови потребителски (студенти и лектори) профили
    - Деактивиране на потребителски профили
    - Активиране на потребителски профили
* **Румен** 
  + Основната работа по създаването на базите от данни
  + Подредба на файловата структура и пътищата на документите във финалната версия на проекта
  + Работа по функционалностите на администратора
    - Преглед на личен профил
    - Одобряване на предложени избираеми дисциплини
    - Премахване на избираеми дисциплини от списъка с активни дисциплини
    - Добавяне на избираеми дисциплини от списъка с активни дисциплини
    - Създаване на нови администраторски профили
    - Създаване на нови потребителск (студенти и лектори) профили
    - Деактивиране на потребителски профили
    - Активиране на потребителски профили

## 10. Какво научих

* **Кристина -** При разработване на този курсов проект се запознах по-отблизо с целият процес на планиране и разработване на един уеб сайт. В процеса на работа се запознах с доста нови неща (PHP, JSON, AJAX) и използвах знания получени в различни курсове - бази от данни, front-end, back-end, и работа в екип, която ми позволи да получа по-реална представа как протичат подобни процеси в реална работна обстановка.
* **Невена -** С разработването на този курсов проект научих много за това как протича цялостната разработка на един уеб сайт – от клиентската част до връзката с базата. Нямах опит с PHP, но с достатъчно разучаване на използваните технологии, успях да се справя задоволително за мен самата. В този проект имах възможността да приложа всичко научено от различни курсове, специализирани в отделните части – бази от данни, back-end, front-end, както и да работя в екип, което, от своя страна, ми беше доста полезно.
* **Радослав -** По време на разработката на проекта успях да усвоя нови знания за уеб програмирането и разработката на уеб приложения. Придобих знания за конкретни технологии и езици като: PHP, XAMPP, JavaScript, AJAX и JSON. Освен това придобих опит и наблюдения върху работата в екип и взаимодействието между членовете на екипа.
* **Румен -** При работата върху проекта научих как протича разработването на уеб сайт от екип от няколко разработчика. Научих за как да използвам  PHP, JavaScrip, AJAX, XAMPP, JSON за създаването на цялостно приложение. Също така имах възможност да упражня знанията си за Бази от данни.

## 11. Използвани източници

[1] Session Handling, PHP Manual (<http://php.net/manual/en/book.session.php>)

    [2] COOKIES, PHP Manual (<http://php.net/manual/en/features.cookies.php>)

    [3] JavaScript Cookies (<https://www.w3schools.com/js/js_cookies.asp>)

    [4] PHP - AJAX and PHP (<https://www.w3schools.com/php/php_ajax_php.asp>)

    [5] JavaScript Object Notation, PHP Manual (<http://php.net/manual/en/book.json.php>)

    [6] JavaScript Object Notation, MDN web docs (<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON>)

[7] hash, PHP Manual (<http://php.net/manual/bg/function.hash.php>)

[8] The MVC pattern and PHP Part 1, Callum Hopkins, March 04, 2013 (<https://www.sitepoint.com/the-mvc-pattern-and-php-1/>)

[9] How Single-Page Applications Work, [Paul Sherman](https://medium.com/@pshrmn?source=post_header_lockup), Apr 11, 2018

(https://medium.com/@pshrmn/demystifying-single-page-applications-3068d0555d46)

Предал (подпис): ………………………….

/*фн, имена, спец., група*/

Приел (подпис): ………………………….

/доц. *Милен Петров*/