

Имена: Добромир Атанасов, Диян Михайлов		фн: [REDACTED]
Начална година: 2017	Програма: бакалавър (КН)	Курс: 4
Тема: "Управление на процес по дипломиране"		имейли: daatanasov98@gmail.com
Дата: 2021-01-22	Предмет: w15prj_KN_final	didom8@abv.bg

преподавател: доц. д-р Милен Петров

## ТЕМА: "Управление на процес по дипломиране"

### 1. Условие

Да се направи система, която управлява процес по дипломиране на студенти. Системата трябва да предоставя възможност за регистрация според ролята – студент или администрация. След като потребителят влезе в системата, му се предлагат различни функции според типа профил. Студентите могат да заявяват присъствие на церемонията по дипломиране и да предоставят друга информация, необходима за организацията на процеса. Администрацията обработва тази информация, генерира статистики за присъствие, както и списъци по различни критерии, които след това може да експортира в CSV формат.

### 2. Въведение

Церемонията по дипломиране е един от най-запомнящите се моменти от висшето образование на всеки човек. Тя предполага солидна предварителна организация, която изисква от студента да направи набор от заявки, които стават наживо. Това отнема от времето на студента, а също така затруднява и работата на администрацията, които трябва да обслужват всеки един студент. Освен това всичко се случва на хартия, а както знаем това неминуемо води до грешки както при попълване, така и при пренасяне на тази информация от едно място на друго.

Нашата концепция е да създадем система, която значително да автоматизира този процес, улеснявайки работата на администрацията и спестявайки време на студента. Всеки студент или служител от администрацията има възможността да си създаде профил според ролята.

След като един студент влезе в профила си, той може да заяви дали иска да присъства на церемонията по дипломиране. В зависимост от отговора му, системата може да поиска информация за броя гости, които смята да покани, размер на тогата, дали иска да си поръча шапка, както и дали иска да държи реч при отличен успех. От своя страна служителите от администрацията могат да въвеждат награди на заслужили студенти в системата, да следят кой е върнал своята тога след церемонията, да изкарват статистики за присъствеността, както и да генерират списъци по най-различни критерии, напр. списък на присъстващите, списък за подпис при получаване на диплома, списъци за речи, тоги, шапки и т.н. Ползата от тази система е, че цялата информация се съхранява на едно място, значително се намаляват рисковете от допускане на грешки, улеснява се работата на администрацията, а студентът лесно може да следи своята информация.

### **3. Теория**

Системата има трислойна архитектура:

- Презентационен слой – изграден е чрез HTML, CSS и JavaScript;
- База данни – MySQL;
- Бизнес логика – имплементирана е чрез езика PHP.

### **4. Използвани технологии**

Технологиите, чрез които е реализиран проектът, са HTML, CSS, JavaScript, PHP и MySQL.

HTML, CSS и JavaScript се използват в т.нар. frontend част, като чрез HTML се създава съдържанието на сайта. CSS се използва за стилизация на страниците с цел по-“user friendly” интерфейс, защото знаем, че много често успехът на един продукт се крие и във външния му вид. Чрез JavaScript е реализирана интерактивността на системата, като чрез него се достъпва функционалността ѝ.

Backend частта е имплементирана на PHP, като чрез него е реализирана цялата функционалност на системата. Също така PHP оперира с базата данни, която е създадена на MySQL.

## 5. Инсталация и настройки

Системата пристига в готов вид, т.е. не изисква никаква инсталация. Трябва просто да се изтегли от платформа, където би могла да бъде качена или да се качи от някакъв носител на информация, като флаш памет и др.

По-особената част е базата данни, която е в експортиран вид и файлът трябва да се импортира в phpmyadmin.

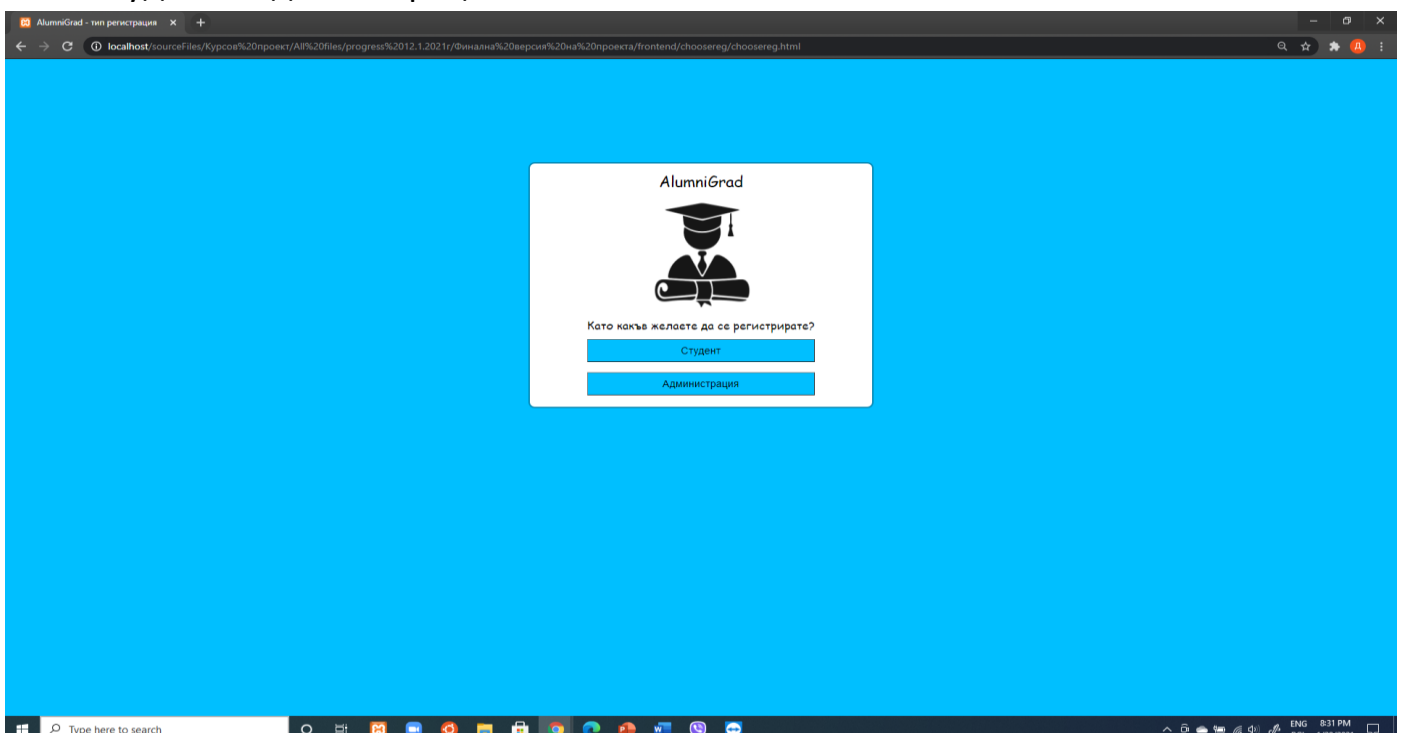
## 6. Кратко ръководство на потребителя

### Стартиране на системата

Очаква се системата да бъде пусната от служител на администрацията на университета преди да се започне подготовката за церемонията по дипломиране. При стартирането на системата, администраторът трябва да има файл с име **“graduating students”** в **CSV** формат с всички завършващи студенти (бакалаври, магистри, докторанти). В този файл за всеки един от завършващите студенти се пази един ред със следната информация: **<факултетен\_номер>,<оценка>** (без ъгловите скоби). Оценката на студент трябва да е във формат **<x>.<x><x>** (без ъгловите скоби) и да принадлежи на интервала [3.00, 6.00].

### 1. Регистрация

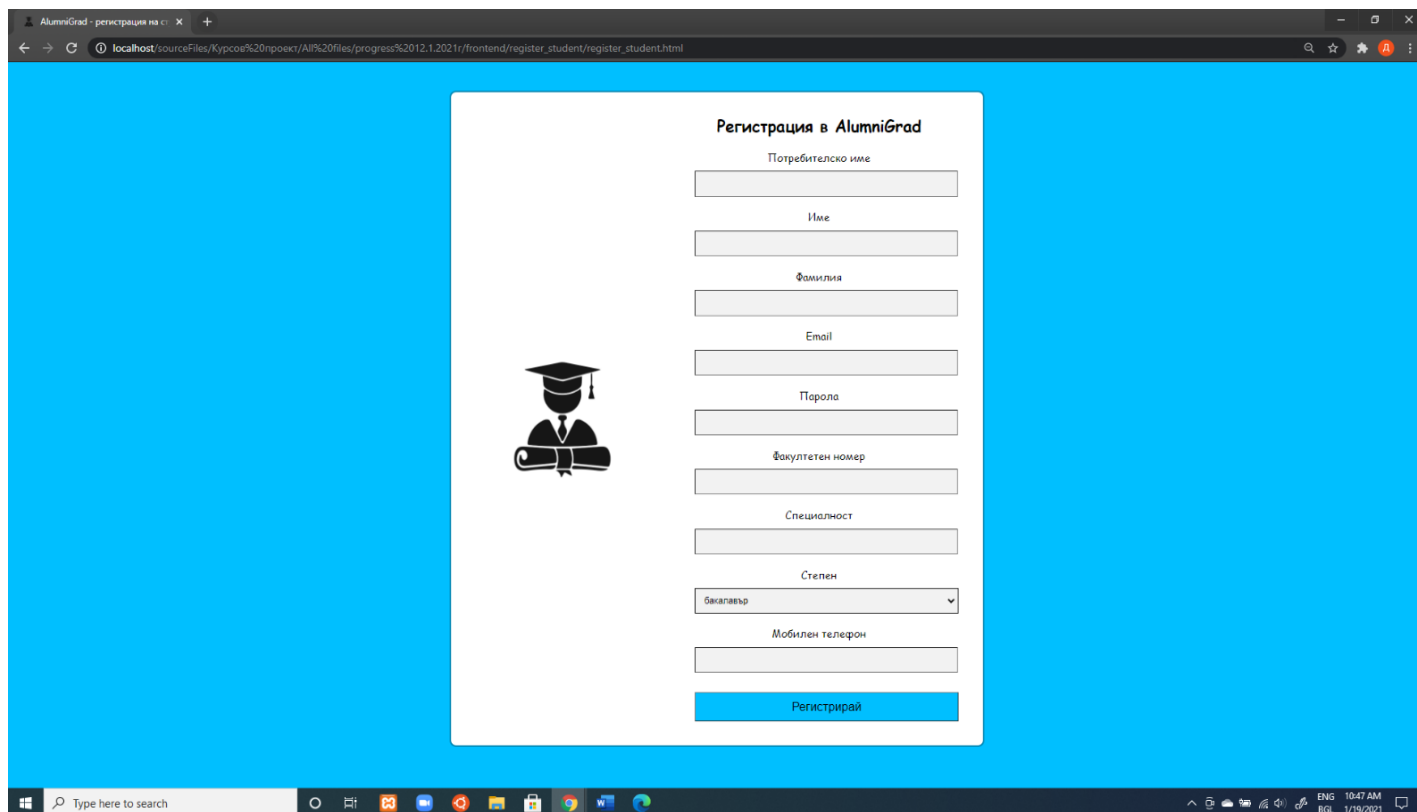
След като системата е стартирана, в нея могат да започнат да се регистрират както завършващи студенти, така и служители от администрацията на университета, т.е. има 2 роли за регистрация – студент и администрация.



След като желаещият да се регистрира избере като какъв иска да се регистрира, системата го пренасочва към различни форми за регистрация според ролята.

### 1.1 Регистрация като студент

Ако избраната роля е студент, то се появява следната страница:



Регистрация в AlumniGrad

Потребителско име

Име

Фамилия

Email

Парола

Факултетен номер

Специалност

Степен

бакалавър

Мобилен телефон

Регистрирай

За успешна регистрация трябва да се попълнят коректно всички полета.

### ВАЖНО !!!

- Системата не позволява да се регистрират студенти, които не фигурират в първоначално импортирания списък със завършващи студенти.
- Системата не позволява да се направи втора регистрация с вече регистриран факултетен номер.

### 1.2 Регистрация като администрация

Ако избраната роля е администрация, то се появява следната страница:

The screenshot shows a web browser window with the URL `localhost/sourceFiles/Курсов%20проект/All%20files/progress%2012.1.2021/frontend/register_administration/register_administration.html`. The page has a bright blue background. In the center is a white rectangular form titled "Регистрация в AlumniGrad". On the left side of the form is a black icon of a graduate wearing a cap and holding a diploma. To the right of the icon are several input fields: "Потребителско име", "Име", "Фамилия", "Email", "Парола", and "Длъжност". Below these fields is a blue button labeled "Регистрирай".

За успешна регистрация трябва да се попълнят коректно всички полета.

## 2. Влизане в системата

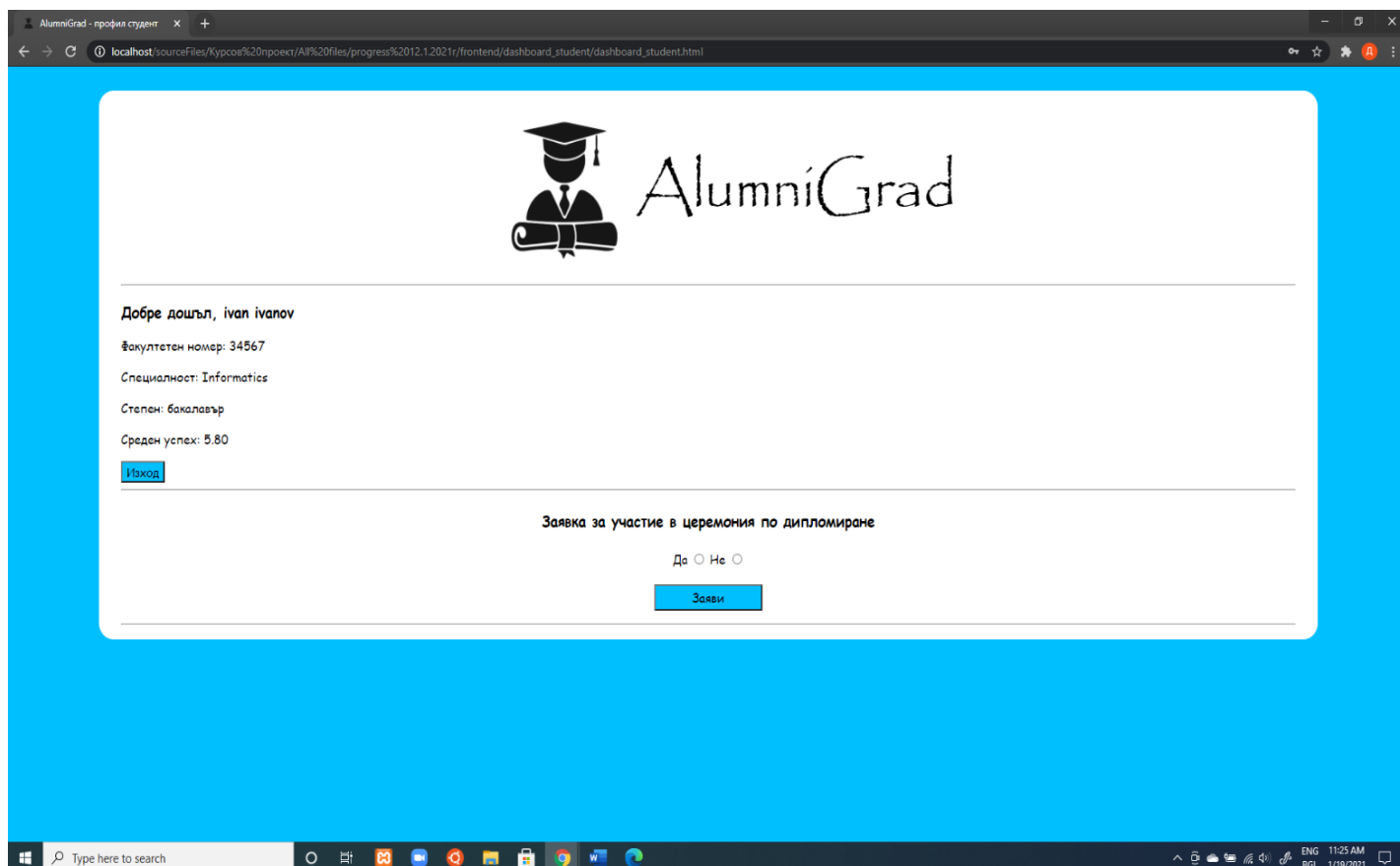
След като се регистрират, потребителите могат да влязат в системата посредством потребителско име и парола:

The screenshot shows a web browser window with the URL `localhost/sourceFiles/Курсов%20проект/All%20files/progress%2012.1.2021/frontend/login/login.html`. The page has a bright blue background. In the center is a white rectangular form titled "Влизане в AlumniGrad". On the left side of the form is a black icon of a graduate wearing a cap and holding a diploma. To the right of the icon are two input fields: "Потребителско име" and "Парола". Below these fields is a blue button labeled "Вход". At the bottom of the form, there is a link that says "Нямате профил? [Регистрация](#)".

### 3. Функционалност

#### 3.1 Роля “Студент”

Когато потребител се е регистрирал като студент и влезе успешно в системата, той бива пренасочен към следната страница:



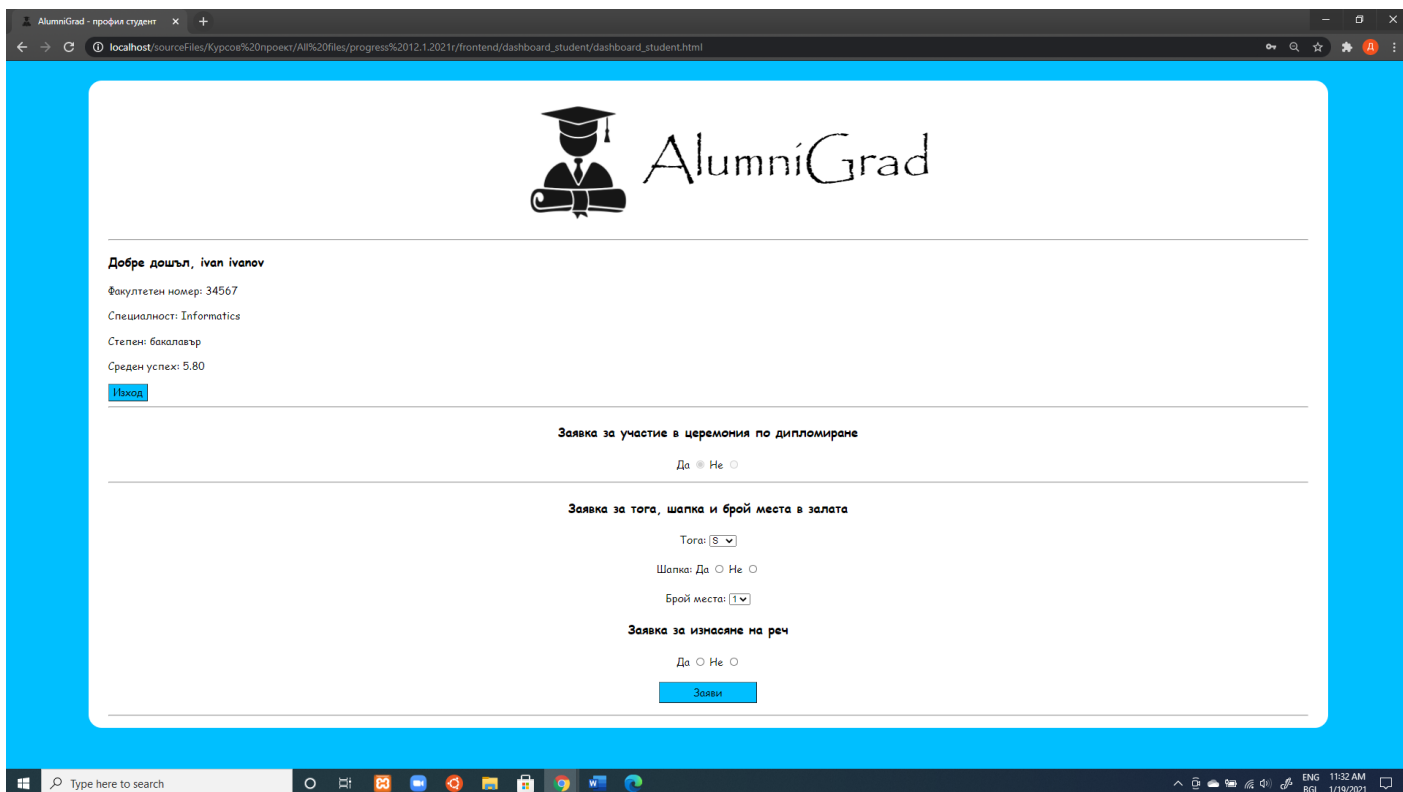
#### **ВАЖНО!!!**

Веднъж въвел някакъв отговор, студентът не може да го промени, т.е. отговорът се записва в системата и съответното поле за отговор става неактивно!

Студентът трябва да посочи дали ще присъства на церемонията по дипломирането.

##### 3.1.1 Студентът ще присъства

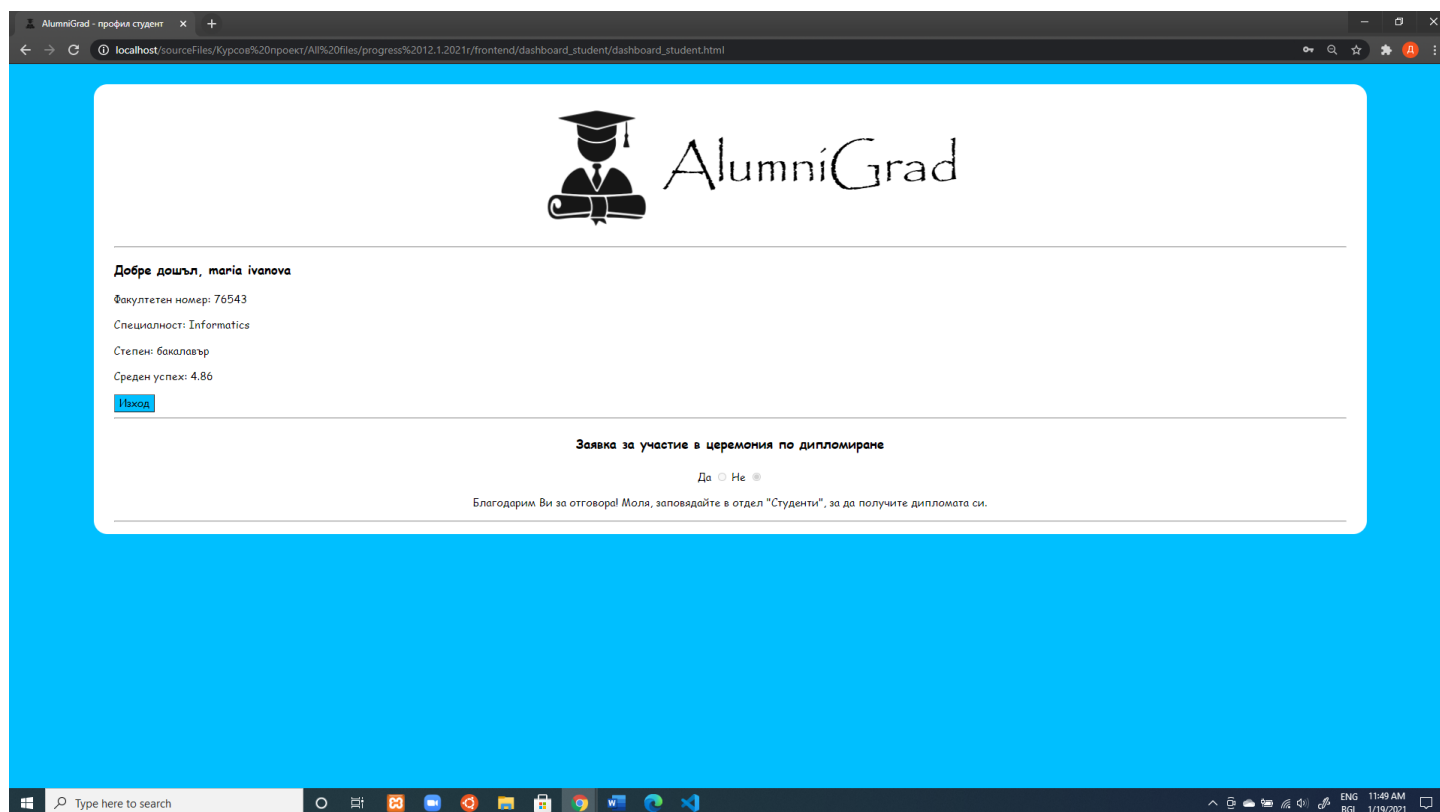
Ако заяви, че ще присъства на церемонията, му се появяват допълнителни полета, които трябва да попълни:



Студентът трябва да си избере размер тога от падащото меню, да посочи дали желае шапка, както и да избере колко гости смята да доведе на церемонията. За отличните студенти (с успех  $\geq 5.50$ ) се появява допълнително поле, в което студентът попълва дали иска да изнесе реч по време на церемонията. След като попълни цялата тази информация, студентът може да излезе от системата чрез бутона “Изход”.

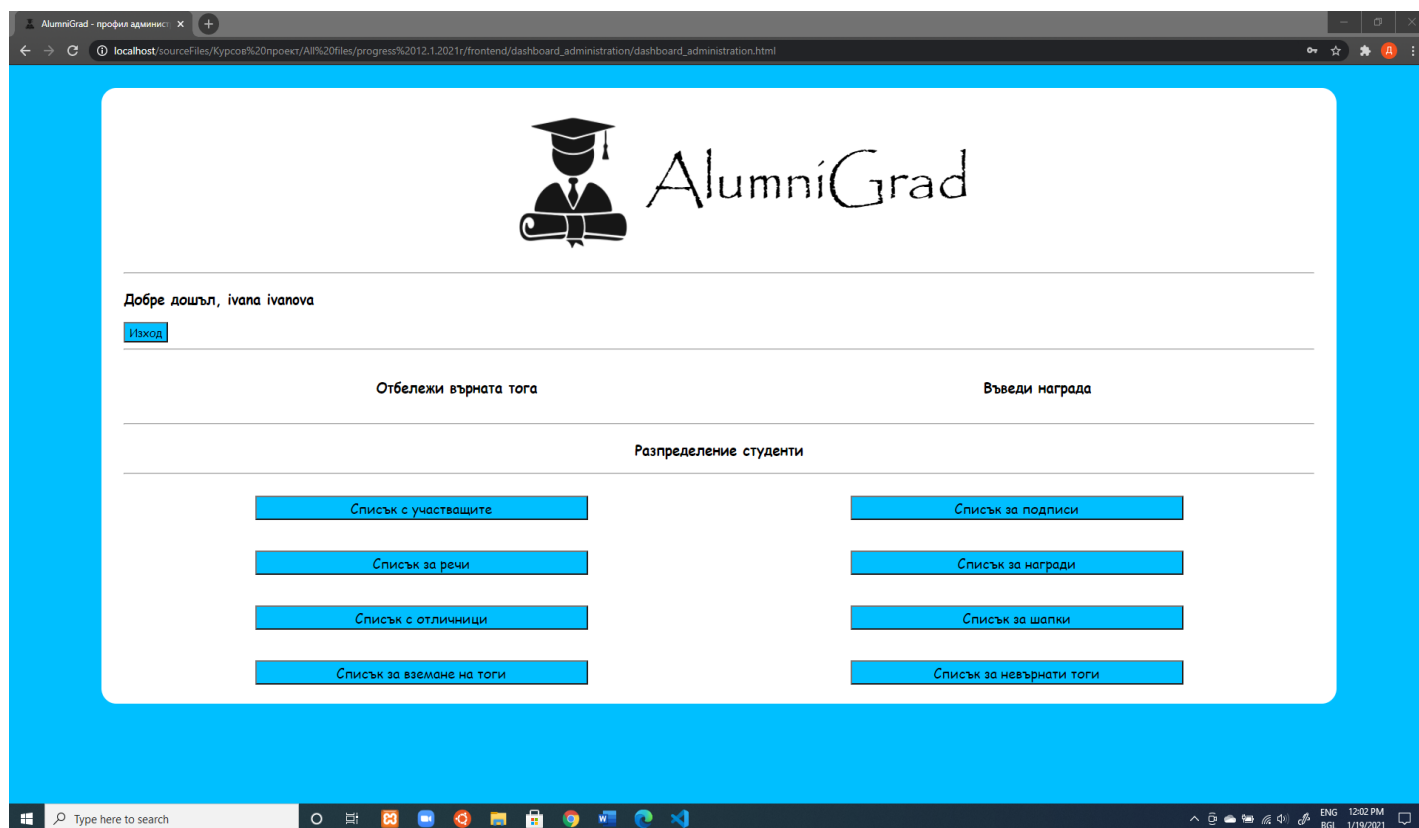
### 3.1.2 Студентът няма да присъства

Ако студентът няма да присъства на церемонията, му се появява следната страница:



### 3.2 Роля “Администрация”

Когато потребител се е регистрирал като администрация и влезе успешно в системата, той бива пренасочен към следната страница:



**Служителите от администрацията могат да попълват по факултетни номера:**

- 1) кои студенти са върнали тогите си;
- 2) награди за студентите, които ще присъстват на церемонията – награди за отличниците, за студентите, представили се отлично на състезания, научни конференции и т.н.

Попълването на тази информация става като се натисне върху “Отбележи върната тога” и “Въведи награда” съответно.

Служителите от администрацията имат достъп и до статистическа информация, когато натиснат върху “Разпределение студенти”. Тази информация е свързана с това колко студенти са се регистрирали в системата, колко от тях ще присъстват на церемонията, колко няма да присъстват, както и ориентиран брой гости, които присъстващите студенти са попълнили, че ще доведат.



След като е минала датата, до която е обявено, че студенти могат да се регистрират и да попълват необходимата информация, служител от администрацията може да влезе в системата и да изтегли различни списъци в **CSV** формат:

- 1) Списък със студентите, които ще присъстват на церемонията, като студентите са подредени по степен и успех.
- 2) Списък за подписи, в който получилите своите дипломи студенти ще се подписват.
- 3) Списък за речи, в който са избрани 3-ма отличници от изразилите желание да изнесат реч по време на церемонията.
- 4) Списък с награди, в който са отбелязани присъстващите на церемонията наградени студенти, както и наградите, които те получават.
- 5) Списък с отлично завършващите студенти.
- 6) Списък за шапки, в който са записани всички студенти, изяви́ли желание да носят шапка на церемонията.
- 7) Списък за вземане на тоги, в който са записани студентите, взели тоги, както и размерите (S, M, L, XL) на взетите тоги.
- 8) Списък за невърнати тоги, в който са записани студентите, които все още не са върнали своите тоги.

AlumniGrad - профил администратор

localhost/sourceFiles/Курсов%20проект/All%20files/progress%2012.1.2021r/frontend/dashboard\_administration/dashboard\_administration.html

Добре дошъл, ivana ivanova

Изход

Отбележи върната тога

Факултетен номер:

Въведи

Въведи награда

Факултетен номер:

Награда:

Въведи

Разпределение студенти

Общо регистрирани: 10

Заявили присъствие: 7

Незаявили присъствие: 3

Брой присъстващи студенти и гости в залата: 24

Списък с участващите

Списък за подписи

Списък за речи

Списък за награди

Списък с отличници

Списък за шапки

Списък за вземане на тоги

Списък за невърнати тоги

Type here to search

12:12 PM 1/19/2021

## 7. Примерни данни

Както става ясно от секция “Ръководство на потребителя”, при стартиране на системата трябва да ѝ се предостави файл в **CSV** формат с факултетните номера и оценките на завършващите студенти. Примерно съдържание на такъв файл:

**80000,5.11**

**70000,4.34**

**60000,5.50**

**50000,5.27**

**98765,5.10**

**12345,4.20**

**54321,6.00**

**51423,5.07**

**34567,5.80**

**76543,4.86**

**Валидни данни за регистрация на студент, чийто факултетен номер е в списъка със завършващи студенти, са следните:**

Потребителско име: **gesh**

Име: **Георги**

Фамилия: **Георгиев**

Email: [gesh@abv.bg](mailto:gesh@abv.bg)

Парола: **123456Qw**

Факултетен номер: **12345**

Специалност: **CS**

Степен: **бакалавър**

Мобилен телефон: **0888765432**

**Валидни данни за регистрация на служител от администрацията са:**

Потребителско име: **reni**

Име: **Ренета**

Фамилия: **Иванова**

Email: [reni@abv.bg](mailto:reni@abv.bg)

Парола: **123456qW**

Длъжност: **инспектор**

## 8. Описание на програмния код

В тази секция ще покажем основни части от програмния код, реализиращи функционалността на приложението.

- Установяване на връзка към базата от данни, като параметрите на връзката се прочитат от отделен конфигурационен файл с име "config-db.ini".

```
32     public function __construct() {
33         $config = parse_ini_file("config-db.ini", true);
34
35         $host = $config['db']['host'];
36         $dbname = $config['db']['name'];
37         $user = $config['db']['user'];
38         $password = $config['db']['password'];
39
40         try {
41             $this->connection = new PDO("mysql:host=$host;dbname=$dbname", $user, $password,
42                 array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8", PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC));
43             $this->connection->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
44             $this->prepareStatements();
45
46             $this->rowCountStatement->execute();
47             $numberOfRows = $this->rowCountStatement->fetchColumn();
48             if ($numberOfRows == 0) {
49                 $this->getStudents();
50             }
51
52         } catch(PDOException $e) {
53             return "Connection to the database failed: " . $e->getMessage();
54         }
55     }
```

- Пример за получаване на данни в JSON формат от frontend частта (JavaScript) на backend частта (PHP), изпълняване на заявка към базата данни и връщане на отговор в JSON формат обратно към JavaScript.

```

1  <?php
2      require_once "../database_files/DB.php";
3      require_once "../database_files/processInput.php";
4
5      session_start();
6      header("Content-type: application/json");
7
8      if ($_POST) {
9
10         $data = json_decode($_POST["data"], true);
11
12         $facnum = isset($data[0]) ? testInput($data[0]) : "";
13         $toga = isset($data[1]) ? testInput($data[1]) : "";
14         $hat = isset($data[2]) ? testInput($data[2]) : "";
15         $places = isset($data[3]) ? testInput($data[3]) : "";
16         $speech = isset($data[4]) ? testInput($data[4]) : "";
17
18         $db = new Database();
19         $errors = [];
20         $response = [];
21
22         $data = [
23             "fn" => $facnum,
24             "toga" => $toga,
25             "hat" => $hat,
26             "places" => $places,
27             "speech" => $speech
28         ];
29
30         if (!$errors) {
31             $res = $db->insertExtraInfo($data);
32
33             if ($res["success"] == false) {
34                 $errors[] = $res["error"];
35             }
36         }
37
38         if ($errors) {
39             $response = ["success" => false, "error" => $errors];
40         }
41         else {
42             $response = ["success" => true, "message" => "Информацията е попълнена!"];
43         }
44
45         echo json_encode($response);
46     }
47

```

- Пример за използване на **fetch**, чрез който изпращаме данни от JavaScript към PHP и получаваме отговор от PHP скрипта, който отговор се преобразува от JSON формат в JavaScript обект. JavaScript използва получените данни, за да променя DOM дървото.

В проекта използваме на много места и **fetch** с **GET** заявка, когато искаме да получаваме данни от backend частта (PHP), но не и да изпращаме данни.

```

82 const insertTogaReturn = event => {
83     event.preventDefault();
84
85     let fn1 = document.getElementById("fn1").value;
86
87     const fields = [
88         fn1
89     ];
90
91     const settingsPost = {
92         method: 'POST',
93         headers: {
94             'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8'
95         },
96         body: `data=${JSON.stringify(fields)}`
97     };
98
99     fetch("../backend/dashboard_administration/dashboard_adm_togaRet.php", settingsPost)
100     .then(response => response.json())
101     .then(response => {
102         if (response.success === true) {
103             document.getElementById("toga-error").innerHTML = response.message;
104             document.getElementById("fn1").value = "";
105             setTimeout(() => {
106                 document.getElementById("toga-error").innerHTML = "";
107             }, 3000);
108         }
109         else {
110             document.getElementById("toga-error").innerHTML = response.error[0];
111             document.getElementById("fn1").value = "";
112             setTimeout(() => {
113                 document.getElementById("toga-error").innerHTML = "";
114             }, 3000);
115         }
116     })
117     .catch(res => console.log(res))
118 }
119
120 document.getElementById("togaForm").addEventListener("submit", insertTogaReturn);

```

- Използваме следния код, за да експортираме различни списъци в **CSV** формат. При натискане на бутон, съответният списък се сваля.

```

230 const exportList = (event, phpFileName, headerRow, downloadFileName, flag) => {
231     event.preventDefault();
232
233     fetch(phpFileName, {method: "GET"})
234     .then(response => response.json())
235     .then(response => {
236         if (response.success == true) {
237             const rows = response.data;
238             let csvContent = "data:text/csv;charset=utf-8,";
239             csvContent += headerRow;
240
241             rows.forEach((elem) => {
242                 const elemString = createStringRow(elem, flag);
243                 csvContent += elemString;
244             })
245
246             const encodedUri = encodeURI(csvContent);
247             let link = document.createElement("a");
248             link.setAttribute("href", encodedUri);
249             link.setAttribute("download", downloadFileName);
250             document.body.appendChild(link);
251
252             link.click();
253         }
254     })
255     .catch(error => console.log(error));
256 }
257
258 document.getElementById("btn-partic").addEventListener('click', (event) => {
259     exportList(event, "../backend/dashboard_administration/exportParticipants.php", "Име,Фамилия,ФН,Специалност,Степен\n", "participantsInCeremony.csv", true);
260 });
261
262 document.getElementById("btn-sign").addEventListener('click', (event) => {
263     exportList(event, "../backend/dashboard_administration/exportSignature.php", "Име,Фамилия,ФН,Подпис\n", "signatureList.csv", false);
264 });
265
266 document.getElementById("btn-speeches").addEventListener('click', (event) => {
267     exportList(event, "../backend/dashboard_administration/exportSpeeches.php", "Име,Фамилия,ФН\n", "speechesList.csv", true);
268 });

```

```

205   const createStringRow = (obj, flag) => {
206       let row = "";
207
208       for(const prop in obj) {
209           row += (obj[prop] + ",");
210       }
211
212       let res = "";
213
214       if (flag == true) res = row.slice(0, -1);
215       else res = row;
216       res += "\n";
217
218       return res;
219   }

```

- Влязъл в системата потребител може да е отворил приложението едновременно в няколко различни таба на брауъра. Системата излиза от профила на потребителя от всички табове, след като той излезе от един от тях. За предаване на информация за настъпване на събитие между табове на брауъра използваме localStorage в JavaScript. Кодът на JavaScript:

Табове следят за излизане от системата. Ако потребителят е излязъл, всички табове се пренасочват към login страницата.

```

9       window.addEventListener("storage", (event) => {
10           if (event.key == "administration-logout") {
11               location.replace("../login/login.html");
12           }
13       })

```

При натискане на бутона за излизане се предава информация към останалите табове, че потребителят е излязъл от системата.

```

37   const logout = (event) => {
38       event.preventDefault();
39
40       fetch("../backend/logout/logout.php", {method: 'GET'})
41           .then(response => response.json())
42           .then(response => {
43               if (response.success === true) {
44                   location.replace("../login/login.html");
45                   localStorage.setItem("administration-logout", "logout" + Math.random());
46               }
47           })
48       .catch(error => console.log(error));
49   }
50
51   document.getElementById("logout-btn").addEventListener('click', logout);

```

## 9. Ограничения и възможности за бъдещо разширение

Системата има потенциал да влезе в реална употреба, като е необходимо да се извършат всички дейности, необходими за пускането ѝ в уеб пространството. Освен това би било много важно да има SSL сертификат, защото обработва чувствителни лични данни, които е редно да се защитят от зловредни прихващания.

Въпреки че системата дава значителен набор от функции и може да се използва за организацията на реален процес по дипломиране, винаги има място за бъдещи доработки. Може да се добави опция за изпращане на съобщения, за да може от администрацията да се свържат със студента директно в рамките на системата при възникнал проблем, както и много други функции, тясно свързани с прилаганите практики и спецификата при организация на процес по дипломиране.

## 10. Какво научих

Разработването на един такъв проект, макар и с учебна цел, със сигурност води до много по-задълбочено разбиране на целия материал. Научава се как различните технологии, изучавани на упражнения, се обединяват в цялостен продукт. В реална работна среда това несъмнено би помогнало за по-лесно справяне с такъв вид уеб проекти. Конкретно от този проект научихме:

- Как да обработваме потребителските събития
- Целия процес от това потребителят да въведе дадена информация, да се валидира с JS, след това да се изпрати към сървърната част, да се получи от нея, да се обработи и накрая да се запише в базата данни
- Особености при създаването на сесия за потребител
- Експортиране на информация в CSV формат

и много други.

## 11. Използвани източници

### HTML:

<https://www.w3schools.com/html/>

<https://developer.mozilla.org/en-US/docs/Web/HTML>

### CSS:

<https://www.w3schools.com/css/>

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

### JavaScript:

<https://www.w3schools.com/js/DEFAULT.asp>

<https://www.tutorialspoint.com/javascript/index.htm>

### PHP:

<https://www.php.net/docs.php>

### MySQL:

<https://dev.mysql.com/doc/refman/8.0/en/>

Danny Pule, "Export JSON to CSV file using Javascript (in the browser)", публикувано на 8 февруари 2017 г.

[<https://medium.com/@danny.pule/export-json-to-csv-file-using-javascript-a0b7bc5b00d2>]

Nosa Obaseki, "localStorage in JavaScript: A complete guide", последна промяна на 29 декември 2020 г. [<https://blog.logrocket.com/localstorage-javascript-complete-guide/>]

Sajal Soni, "How to Use Sessions and Session Variables in PHP", публикувано на 4 октомври 2018 г. [<https://code.tutsplus.com/tutorials/how-to-use-sessions-and-session-variables-in-php--cms-31839>]

iTechOwl, "JavaScript: Automatically Logout from all Open Tabs When User Logout in One of them", публикувано на 19 юли 2019 г.

[<https://itechowl.wordpress.com/2019/07/11/javascript-automatically-logout-from-all-open-tabs-when-user-logout-in-one/>]

snowmonkey (във форум на feecodecamp), "Pass the event to the callback within the event listener that triggered it?", май 2020 г.



[<https://forum.freecodecamp.org/t/pass-the-event-to-the-callback-within-the-event-listener-that-triggered-it/258259/2>]

Предал (подпис): .....

*/фн, имена, спец., група/*

Приел (подпис): .....

*/доц. Милен Петров/*