**Имена:**  Росен Андреев Колев, Ралица Атанасова Симова, Георги Николаев Атанасов **ФН:** *0MI0800065, 1MI0800135, 3MI0800092***Начална година:** *2021* **Програма:** бакалавър, (КН) **Курс: 4**   
**Тема: 4.1  
Дата: 26.01.2025 Предмет: w23prj\_KN \_final   
Имейл:** [**rosenandreevkolev1@gmail.com**](mailto:rosenandreevkolev1@gmail.com)**,** [**ralica.a.simova@gmail.com**](mailto:ralica.a.simova@gmail.com)**,** [**nigosto@gmail.com**](mailto:nigosto@gmail.com)

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

# **ТЕМА: w23/4.1 - Alumni Hub - Управление на процес по дипломиране**

## **1. Условие**

Проектът представлява система за управление на процеса по дипломиране. Основната му цел е предоставянето на възможност за контролиране на церемониите за дипломиране и предоставянето на съпътстващата ги информация по удобен начин. Предназначението му е да се превърне в единен портал, където администрацията да предоставя необходимата информация за предстоящи събития и студентите да получават лесно тази информация и да реагират, където това е необходимо. Системата е подходяща за използване от различни университети в България.

**2. Въведение – извличане на изисквания**

В системата са реализирани три различни роли за потребителите - студенти, администрация и админ. Админът има най-големите правомощия в системата - одобрява заявки за регистрация на студенти и администрация и следи за нередности. Администрацията организира процеса по дипломирането и изпраща покани до студентите, а студентите следят за новини и отговарят на поканите от администрацията.

Основните функционалности, реализирани с този проект са:

* Регистрация на потребител. Възможностите за тип профил са студентски или администраторски. Регистрацията трябва да става с потребителско име, имейл и парола, като името и имейла трябва да са уникални в рамката на системата. За студентските профили е необходим и факултетен номер. След изпращане на заявката за регистрация, потребителя трябва да изчака получаване на одобрение от админа.
* Влизане във вече съществуващ профил. За вход са необходими потребителско име и парола.
* Добавяне на нов факултетен номер към студентския профил. Студентът трябва да има възможност да влиза в системата с различни факултетни номера и да вижда различна информация спрямо факултетния номер, който е избрал.
* Информация за профила. Потребителят трябва да може да види данните за неговия профил като потребителско име и имейл, а за студентските профили е налична и информация за следването на студента - средна оценка, дата на завършване и факултетен номер, както и информация за предстоящи церемонии за дипломирането.
* Импорт на студенти. Администрацията трябва да може да импортира данните за студентите, които ще се дипломират през следващата година, като импорта става от файл в CSV формат. След като информацията е импортирана, данните за всички студенти са налични за преглед в таблица.
* Експорт на таблици със студенти и церемонии в CSV формат.
* Създаване на церемония - трябва да се предостави дата, година на завършване и да се поканят студентите. Тези покани може да са обикновена покана за церемонията, покана за изнасяне на речи, покана за отговорник за тоги, дипломи или подписи. Церемониите трябва да могат да се редактират. Всяка промяна в церемонията трябва да стане видима за студента след обновяване на страницата.
* Получаване на информация от студентите за предстояща церемония и даване на отговор на поканите. Студентите трябва да могат да отговорят с ДА/ НЕ на поканата за церемонията и ако приемат да присъстват и имат други покани - за речи и за отговорник тоги, дипломи или подписи, да могат да отговорят на тях също.
* Възможност за повторно изпращане на поканите за речи и за отговорник за тоги, дипломи или подписи от администрацията, ако някой от поканените е отказал.
* Студентите трябва да могат да изберат размер за тогата си, като изборът става само между останалите налични размери.
* Възможност за преглед на всички церемонии и на подробна информация за някоя церемония от администрацията.
* Админът трябва да има възможност да одобрява заявките за създаване на профили.

Нефункционалните изисквания, които трябва да се реализират със системата са:

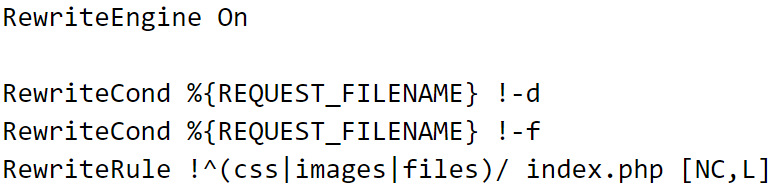
* Сигурност чрез контрол на достъпа спрямо ролите. Преди да успеят да влязат в приложението, потребителите трябва да минават през процес на удостоверяване и овластяване. Контрол на достъпа се осъществява и чрез одобряването на заявките за създаване на профил. Постигане на сигурност при съхранението на паролите чрез криптиране.
* Надеждност се постига чрез валидиране на входните данни и обработка на грешките.
* Използваемост се постига чрез интуитивен дизайн и показване на подходящи грешки на потребителя.
* Възможност за лесна промяна се постига чрез преизползването на общи компоненти и дизайна на архитектурата чрез стила модел-изглед-контролер (MVC)
* Конфигурируемост се постига чрез наличието на конфигурационен файл с променливи на средата.

Ползите от реализацията на тази система са постигането на унифициран и систематизиран процес за дипломирането, събирането на необходимата информация на едно място, достъпно за всички участници в процеса и доставянето на лесен и удобен канал за информация.

**3. Теория – анализ и проектиране на решението**

Реализирана е архитектурата на модел-изглед-контролер (MVC). Моделният компонент управлява поведението и данните на приложението, изпраща информация за неговото състояние и отговаря на инструкции за промяна на състоянието. Изгледът има задължението да управлява представянето на информация пред потребителите. Контролерът управлява взаимодействието с потребителя, валидира входните данни и информира модела или изгледа, за да предприемат подходящи действия.

Като допълнителен архитектурен стил е реализиран Front Controller.Той се състои в използването на централизирана точка за обработка на всички входящи заявки към приложението.



фиг. 3.1 Логиката за пренасочване на заявките към единствената входна точка.

**4. Използвани технологии**

За реализацията на проекта е използвана средата за разработка XAMPP, версия 8.12.2-0, както под операционната система Windows 10 и Windows 11, така и под Ubuntu 22.04. Използваната версия на XAMPP включва:

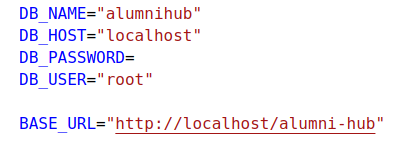
* интерпретатор на PHP, версия 8.12.2;
* сървър Apache, версия 2.4.58;
* дистрибуция на MariaDB, версия 10.4.32.

По време на разработката е използвана системата за контрол на версиите Git, както и отдалеченото хранилище Github.

## **5. Инсталация, настройки и DevOps**

За инсталацията на приложението е необходимо да има инсталирана средата XAMPP, версия 8.12.2-0 или по-висока. Изходният код на приложението трябва да бъде поставен в директорията htdocs (или в някоя нейна поддиректория), която се намира в директорията на инсталация на XAMPP. Например под операционната система Windows, стандартния път до директорията е: C:\xampp\htdocs, а под Linux е /opt/lampp/htdocs. След това трябва през контролния панел на XAMPP да бъдат стартирани Apache Web Server и MySQL Database.

За правилната работа на приложението, трябва да бъде създаден .env файл, който да съдържа променливи на средата. Файлът .env.example съдържа примерната структура на един .env файл, но без конкретни стойности на променливите. Необходимите променливи са име, хост, парола и потребител на базата от данни, както и базовият URL на приложението. Този URL трябва да съдържа като подпътища съответните поддиректории на htdocs, в които е разположен изходния код. Ако кодът на приложението е разположен в директорията C:\xampp\htdocs\alumni-hub, не са правени промени по конфигурацията на MariaDB и името на базата данни е alumnihub, един примерен .env файл би изглеждал по следния начин:



фиг. 5.1 Примерен .env файл

След това, приложението ще бъде достъпно на адреса, сочен от BASE\_URL променливата. Преди обаче да бъде готово за употреба, трябва да се изпълнят миграциите върху базата от данни. Това става, като се изпълни следната команда:

*php scripts/run\_migrations.php -m -a*

Ако е необходимо, могат да се приложат единични миграции, като се изпълни горната команда без аргумента *-a* и вместо него се окаже името на миграция - директорията, в която се намира. Например за изпълнението на миграцията за създаване на началните таблици може да се използва командата:

*php scripts/run\_migrations.php -m m1\_create\_tables*

Ако има нужда да се отменят миграциите може да се използват горните две команди, но вместо аргумента *-m* се използва *-r*. Например за отмяна на всички миграции може да се използва командата:

*php scripts/run\_migrations.php -r -a*

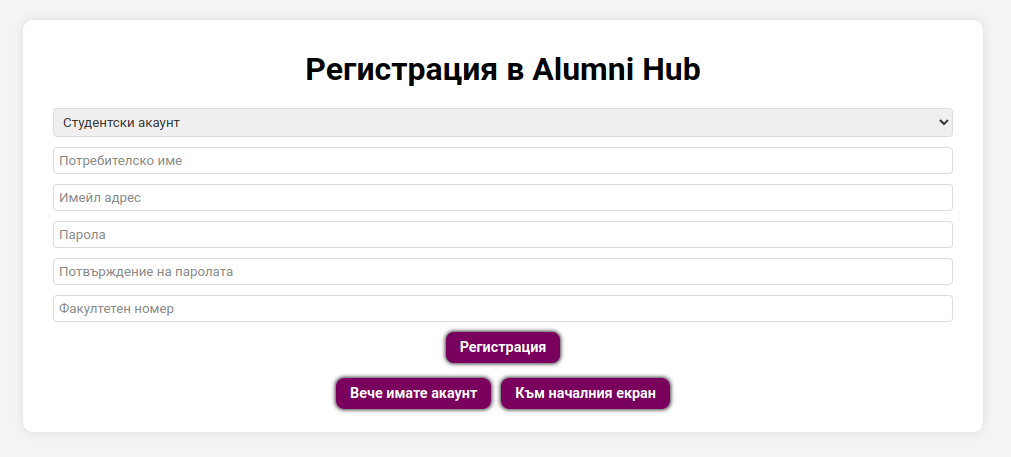
Преди да се започне работа с приложението трябва да се изпълни скрипт, който да зареди началните данни. Това става, като се изпълни следната команда:

*php scripts/seed\_database.php seed*

След изпълнението на този скрипт, в базата от данни се зарежда потребител с роля админ, както и примерни потребители, студенти и дрехи. Без този скрипт няма как да се добави потребител с роля админ и съответно няма как да бъдат одобрени останалите потребители.

## **6. Кратко ръководство на потребителя**

Потребителят е посрещнат от начална страница, но за да достъпи функционалностите, трябва да се регистрира. При регистрация, потребителят може да избере роля на администратор или на студент, като за студентската регистрация трябва да добави и факултетен номер.



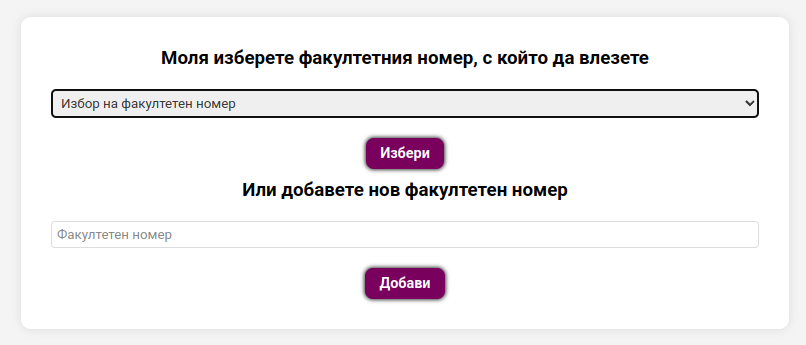
фиг. 6.1 Форма за регистрация с избрана роля за студент

При успешна регистрация, потребителят трябва да изчака одобрение на профила от админа, преди да може да достъпи функционалностите за избраната роля.



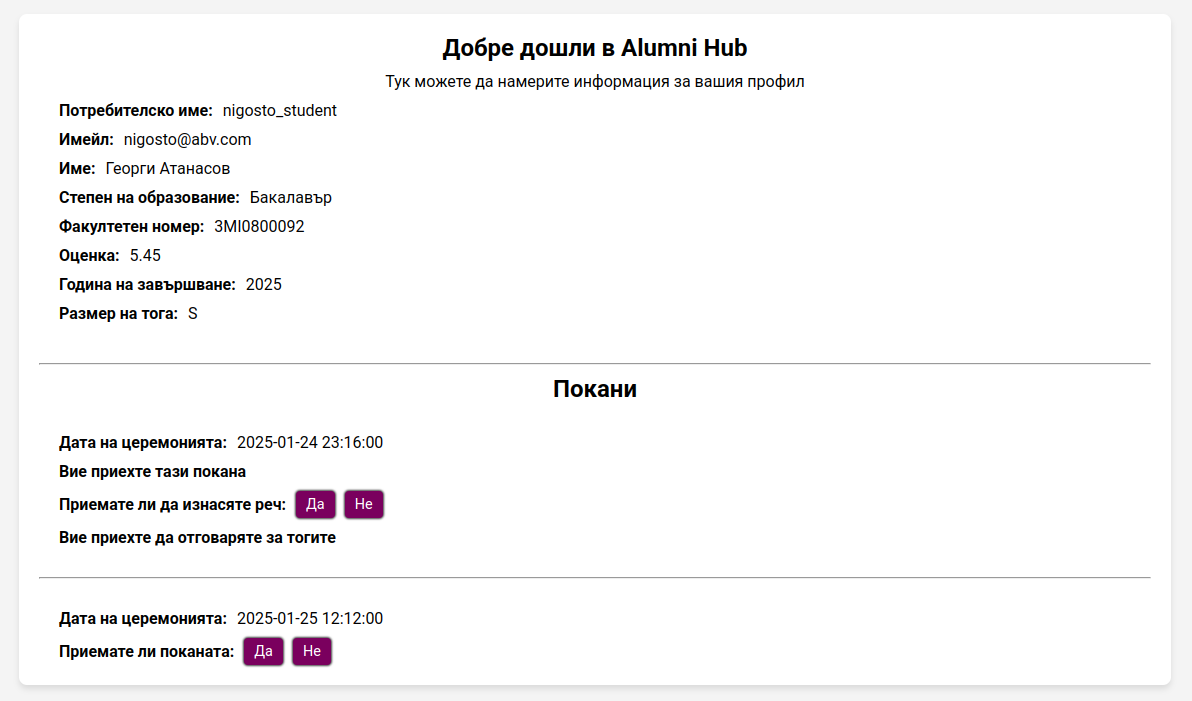
фиг. 6.2 Списък на заявките за одобряване на администратори

След одобряване, администраторът има пълен достъп до всички функционалности на администрацията, но студентът трябва да избере, с кой от одобрените му факултетни номера да влезе, понеже може да има профил с един факултетен номер, докато е бил бакалавър и друг, след като е станал магистър.



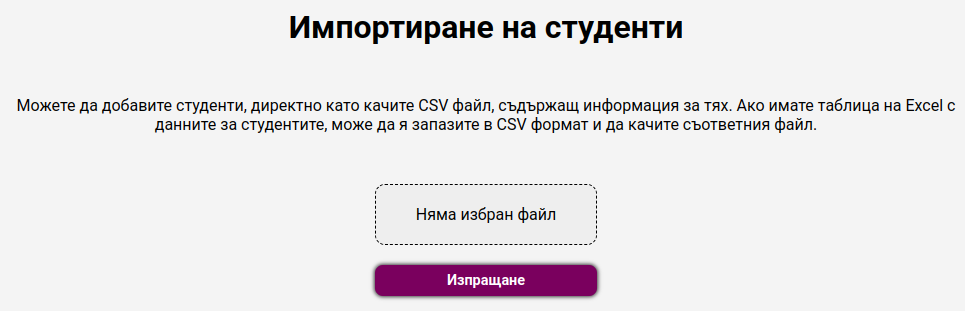
фиг. 6.3 Форма за избор или добавяне на нов факултетен номер

След избиране на факултетен номер, студентът ще бъде пренасочен към профилната му страница, където може да види информацията за себе си, както и поканите за участие в церемониите.



фиг. 6.4 Информация за профила на студент

Администраторът от друга страна, може да добавя студенти, да преглежда списъка със студенти и да го експортира.



фиг. 6.5 Поле за качване на файл, съдържащ информация за студенти

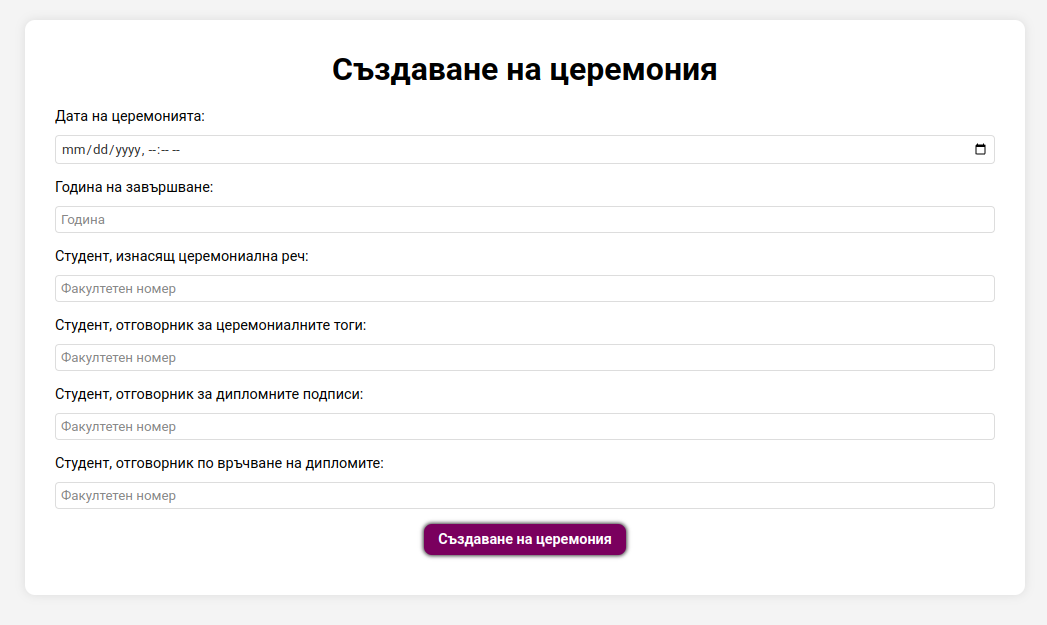


фиг. 6.6 Списък с всички студенти в системата

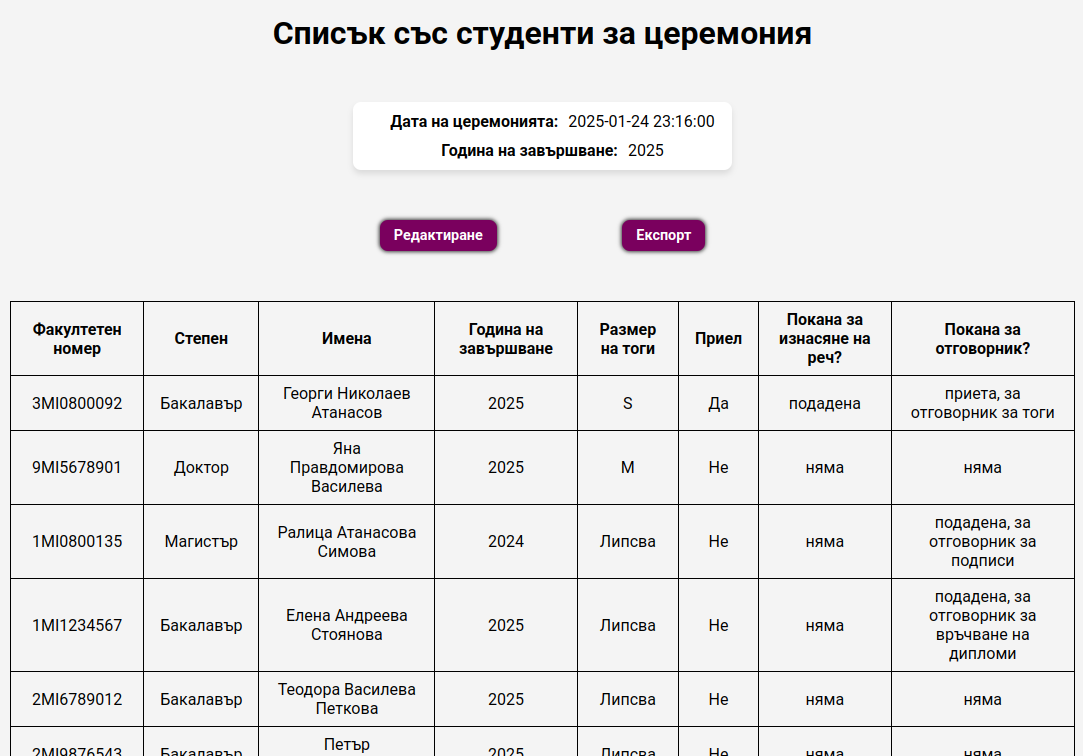
Освен това, администраторът може да управлява церемониите - да създава церемонии, да кани студенти, да редактира церемонии и да преглежда информацията за церемониите. Това включва както преглед на всички церемонии, така и преглед на подробна информация за конкретна церемония.



фиг. 6.7 Списък с всички церемонии в системата



фиг. 6.8 Форма за създаване на церемония



фиг. 6.9 Списък с информация за студентите, поканени на конкретна церемония

## **7. Примерни данни**

## Примерните данни за базата се намират в поддиректория на проекта „*database/seed/seed.php*“.

Това е файл, който генерира примерни данни и ги вмъква в базата. Примерните данни се състоят от потребителски профили, студентски профили и дрехи.

Полезни примерни данни, генерирани от „*database/seed/seed.php*“ :

1. За потребител тип „админ“:
   1. Име: IvanPetrov
   2. Парола: parola123
2. За потребител тип „администратор“, одобрен от „админ“:
   1. Име: MariaIvanova
   2. Парола: qwerty123
3. За потребител тип „администратор“, неодобрен от „админ“:
   1. Име: GeorgiGeorgiv
   2. Парола: pass456
4. За потребител тип „студент“, одобрен от „админ“:
   1. Име: PetarKolev
   2. Парола: abc12345
   3. ФН: 2MI9876543

За изпълнението му е необходимо да се изпълни командата:

*php scripts/seed\_database.php seed*

Това е скриптов файл, чиято цел е да изпълнява файла за генериране на примерни данни в базата. Аргументът *seed* на последно място обозначава поддиректория на папката *database*, в която се намира файла за генериране на данните, който трябва да се казва *seed.php*. Предназначението е по този начин да могат да се създават много файлове за генериране на данни и да се избира между тях в зависимост от ситуацията.

В допълнение към тези данни, има и предварително подготвен файл:

*examples/imports/students\_import.csv*

Той може да се използва за демонстрация на администраторската функционалност за вмъкване на студенти в системата посредством файлове в CSV формат.

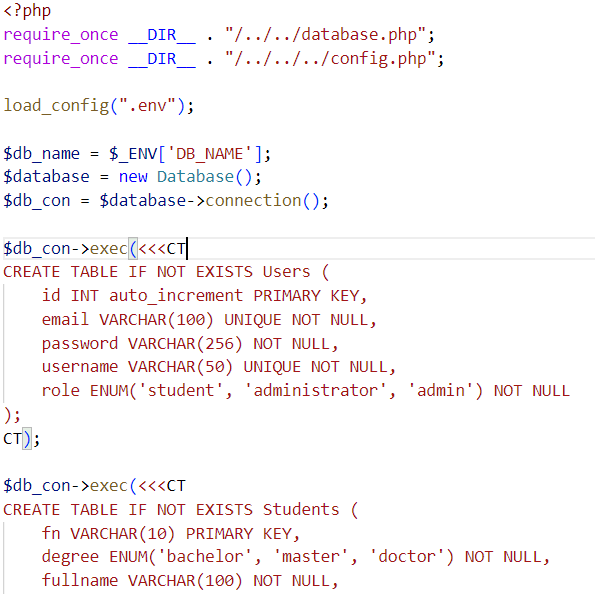
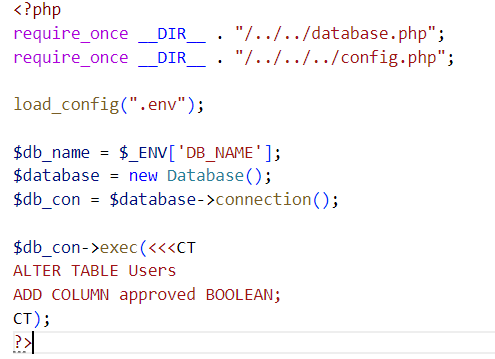
Полезни примерни данни за студенти, генерирани от *examples/imports/students\_import.csv*:

1. Студент 1:
   1. Имена: Георги Николаев Атанасов
   2. ФН: 3MI0800092
   3. Година на завършване: 2025
   4. Степен: Бакалавър
2. Студент 2:
   1. Имена: Петър Станимирев Колев
   2. ФН: 2MI9876543
   3. Година на завършване: 2025
   4. Степен: Бакалавър
3. Студент 3:
   1. Имена: Петър Станимирев Колев
   2. ФН: 9MI9876549
   3. Година на завършване: 2027
   4. Степен: Магистър
4. Студент 4:
   1. Имена: Ралица Атанасова Симова
   2. ФН: 1MI0800135
   3. Година на завършване: 2024
   4. Степен: Магистър

## **8. Описание на програмния код**

Приложението е изградено върху основите на *MVC* - *model-view-controller* (*модел-изглед-контролер*). За по-ясно разяснение на софтуерната архитектура на проекта, ще разгледаме файловата структура, разделена на отделни папки и файлове:

1. *database* - Съдържа скриптови файлове за миграциите на базата данни в процеса на разработка на приложението, както и скриптови файлове за генериране на примерни данни за базата с цел тестване и демонстриране на приложението.

  
фиг. 8.1 Код за първата миграция на базата данни.  
  
  
  
фиг. 8.2 Код за миграция, добавяща нова колона към таблица *Users*   
  
  
фиг. 8.3 Код за генериране на примерни данни и добавянето им в таблица *Users*

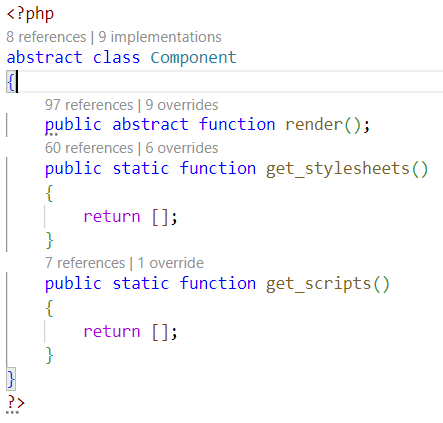
1. *models* - Съдържа класове, моделиращи данните от базата и представлява една част от „*model*“ на *MVC* архитектурата. Съдържа интерфейс за базов модел и конкретни класове наследници.

  
фиг. 8.4 Код за модела *Ceremony*, базиран на таблицата *Ceremony* от базата данни

1. *services -* Съдържа класове с бизнес логика, които най-често боравят с базата за извличане и попълване на данните. Съдържа клас *DataService* с базови функции за комуникация с базата и конкретни класове наследници, комуникиращи с базата относно определени нейни таблици и базирани върху тях модели.

  
фиг. 8.5 Код на класа *DataService* за базова комуникация с базата.

1. *components* - Файлове с преизползваеми графични елементи, които могат да се вмъкват в различните страници. Съдържа абстрактен клас за базов компонент и конкретни класове наследници.

  
фиг. 8.6 Абстрактен клас за базов компонент  
  
  
фиг. 8.7 Компонент, моделиращ заглавната секция на сайта на приложението.

1. *pages* - Съдържа изгледите, моделиращи уеб страниците на приложението. Представлява *view* частта на *MVC* архитектурата.

  
фиг. 8.8 Отрязък от код за страницата за вписване на потребители.

1. *controllers -* Съдържа класове, играещи ролята на контролери от *MVC* архитектурата, свързващи изгледите и моделите посредством бизнес логиката.

  
фиг. 8.9 Отрязък от кода на класа *StudentsController* с функции за показване на страница (изглед), за импорт и експорт и за събиране на данни чрез сървис.

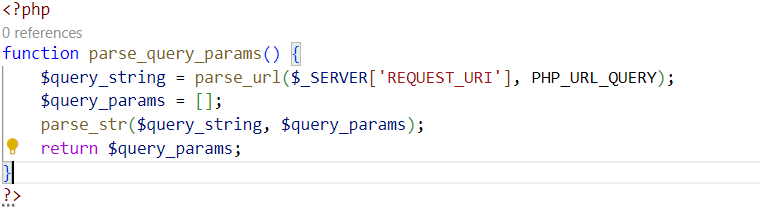
1. *middleware* - Съдържа допълнителна междинна бизнес логика, действаща като посредник между маршрутизацията и контролерите.

  
фиг. 8.10 Междинна логика по удостоверяване на потребителите на приложението.

1. *scripts -* Съдържа код с полезни скриптове, имащи разнообразни цели.

  
  
фиг. 8.11 Отрязък от код на скрипт за изпълнение на миграции на базата от *database*.

1. *utils* - Съдържа файлове с общополезни функции

  
фиг. 8.12 Полезна функция за обработка на параметри на заявката.

1. *router.php* - Клас с обща функционалност, отговаряща за регистрирането на пътищата в уеб приложението.

  
фиг. 8.13 Класа *Router*

1. *index.php* - Позовавайки се на контролерите от *controllers* и на класа *Router*, дефинира реалните пътища в уеб приложението със всичките им видове заявки (*GET, POST, PUT, PATCH, DELETE*).

  
фиг. 8.14 Дефиниране на пътищата за редактиране на церемония.

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

Ралица Симова, 1MI0800135:

Частите от приложението, които разработих са: регистрация на нов потребител, вход във вече направен профил, добавяне и избор на факултетен номер, информация за потребителя, избор на размер за тоги, валидации и показване на грешки на потребителите.

Георги Атанасов, 3MI0800092:

Частите от приложението, които разработих са: панела за админа, импортиране на студенти, преглед и експортиране на студенти, приемане и отказване на покани за церемонии, сигурността покрай овластяването и одобряването на профилите.

Росен Колев, 0MI0800065:

Частите от приложението, които разработих са: създаване на церемонии, изглед със списък на церемониите, изглед със списък на студенти за церемония, експортиране на студенти за церемония, създаване на базата, мигриране на базата и попълване на примерни данни в нея.

Смятаме, че постигнахме относително завършен вид на проекта и че сме покрили основните функционалности за безпроблемна работа с приложението. Като възможност за бъдещо развитие бихме определили добавянето на още функционалности като:

* Добавянето на датата за церемонията в гугъл календара на потребителя
* Потвърждаване на профила чрез изпращане на имейл
* Възможност за смяна на паролата

## **10. Какво научих**

## Ралица Симова, 1MI0800135:

За първи път разработвам уеб проект с PHP и това ме принуди да разуча нови технологии и принципи. Научих архитектурния стил MVC

Георги Атанасов, 3MI0800092:

Научих повече за разработката на сървърни приложения, които могат да сервират уеб страници, както и за архитектурния стил MVC.

Росен Колев, 0MI0800065:

Научих повече за разработката на сървърни приложения посредством *MVC,* за създаване и мигриране на бази от данни, за цялостното интегриране на бизнес логика и изгледи в цялостно приложение чрез PHP и за създаване на междинна логика за удостоверяване и овластяване.

**11. Използвани източници**

1. <https://www.php.net/docs.php>, последно посетен на 26.01.2025
2. [https://webdevetc.com/blog/the-front-controller-design-pattern-in-php](https://webdevetc.com/blog/the-front-controller-design-pattern-in-php/), последно посетен на 26.01.2025
3. <https://developer.mozilla.org/en-US/docs/Glossary/MVC>, последно посетен на 26.01.2025
4. [https://mariadb.com/kb/en](https://mariadb.com/kb/en/), последно посетен на 26.01.2025

Предал (подпис): ………………………….

*0MI0800065, Росен Андреев Колев, КН, група 3*

Предал (подпис): ………………………….

*1MI0800135, Ралица Атанасова Симова, КН, група 3*

Предал (подпис): ………………………….

*3MI0800092, Георги Николаев Атанасов, КН, група 1*

Приел (подпис): ………………………….

/проф. д-р *Милен Петров*/