

**Имена:** Стефанка Манахова, Момчил Младенов, Румяна Иванова  
**ФН:** 1MI0800189 5MI0800232 8MI0800231  
**Начална година:** 2022 **Програма:** бакалавър КН **Курс:** 4  
**Тема:** Система за записване и управление на присъствие по време на презентации, упражнения или лекции  
**Дата:** 28.01.2026 **Предмет:** w25proj\_KN\_final  
**имейл:** rumiana.ivanova24@gmail.com  
**преподавател:** проф. д-р Милен Петров

**Предаване:** Задачата се предава в архив с попълнен настоящия документ, проекта/проектите с кодовете, README.txt файл, който описва съдържанието на архива; папка с допълнителни компоненти и използвани ресурси, архива да се казва 1MI0800189\_5MI0800232\_8MI0800231\_final.zip.



## ТЕМА: w25/33.1

### Система за записване и управление на присъствие по време на презентации, упражнения или лекции

#### 1. Условие

Настоящият проект има за цел разработването на учеб приложение за управление на учебни събития и проследяване на присъствието на студенти в тях. Приложението е предназначено да подпомогне учебния процес чрез автоматизиране на дейности като регистриране на присъствия, предоставяне на учебни ресурси, комуникация между студенти и преподаватели и визуализация на статистическа информация.

Системата следва да поддържа различни типове потребители, като достъпът до функционалностите се определя от ролята на логнатия потребител. Основният фокус е върху сигурното управление на данните, лесната навигация и възможността за динамично обновяване на информацията без презареждане на страниците.

Проектът се реализира като клиент–сървър учеб приложение, използващо HTML, CSS и JavaScript за клиентската част, PHP за сървърната логика и MySQL за съхранение на данните.

## **2. Въведение – извличане на изисквания**

### **2.1 Потребителски роли**

В системата са дефинирани две основни потребителски роли:

#### **2.1.1 Студент**

Студентът има достъп единствено до информация, свързана с неговото участие в учебните занятия.

Той може:

- да влиза в системата със собствен профил;
- да преглежда личната си информация и статистика за присъствията;
- да вижда списък със събитията, в които е присъствал;
- да достъпва ресурси от събитията, в които е участвал;
- да добавя коментари към събития, в които е участвал;
- да сменя собствената си парола;
- да излиза от системата.

#### **2.1.2 Преподавател**

Преподавателят разполага с разширени права за управление на учебния процес.

Той може:

- да създава и управлява учебни събития;
- да импортира списъци със студенти;
- да добавя и премахва присъствия;
- да управлява учебни ресурси;
- да разглежда информация за всички студенти;
- да преглежда статистики за присъствията;
- да комуникира със студентите чрез коментари;
- да сменя собствената си парола;
- да излиза от системата.

## **2.2 Функционални изисквания**

Основните функционални изисквания към системата са:

- вход и изход от системата чрез сесии;
- управление на потребителски роли;
- визуализиране на учебни събития;
- страница с информация за събитие;
- добавяне и премахване на присъствия;
- импорт на списъци със студенти от файл;
- добавяне и визуализация на учебни ресурси;
- добавяне и визуализация на коментари;
- визуализация на статистика за присъствия;
- смяна на парола от потребителите.

## **2.3 Нефункционални изисквания**

- сигурност на данните чрез PHP сесии;
- защита срещу SQL injection чрез prepared statements;
- ясна разделеност между frontend и backend логиката;
- използване на JSON за комуникация между клиента и сървъра;
- поддръжка на UTF-8 кодировка;
- лесна поддръжка и разширяемост на кода.

## **2.4 Ползи от реализацията**

Реализацията на приложението предоставя:

- автоматизация на процеса по отчитане на присъствия;
- по-гладко проследване на учебния процес;
- улеснена комуникация между преподаватели и студенти;
- възможност за анализ на учебната активност;
- практическо приложение на изучаваните уеб технологии.

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

### **3.1 Архитектура на приложението**

Приложението е реализирано по класическия клиент–сървър модел, при който:

- клиентската част отговаря за визуализацията и взаимодействието с потребителя;
- сървърната част обработва заявките, извършва проверки за достъп и работи с базата данни;
- базата данни съхранява всички основни данни на системата.

### **3.2 Декомпозиция на приложението**

Приложението е разделено на няколко основни логически модула:

#### **Модул „Потребители и автентикация“**

- вход и изход от системата;
- управление на сесии;
- смяна на парола;
- контрол на достъпа според роля.

#### **Модул „Събития“**

- създаване на нови събития;
- визуализиране на списък със събития;
- детайлна информация за конкретно събитие.

#### **Модул „Присъствия“**

- записване на присъствие;
- динамична промяна на присъствието;
- изчисляване на броя и процента на присъствали.

#### **Модул „Ресурси“**

- добавяне на учебни материали;
- контрол на видимостта на ресурсите;
- визуализация по събития.

#### Модул „Коментари“

- добавяне на коментари;
- визуализация на дискусии;
- комуникация между студенти и преподаватели.

#### Модул „Статистика“

- индивидуална статистика за студент;
- обобщена статистика за преподавател;
- визуализация чрез диаграми.

### 3.3 Комуникация между компонентите

Клиентската част използва JavaScript и Fetch API за изпращане на HTTP заявки към PHP endpoints.

Сървърът обработва заявките, извършва необходимите проверки и връща отговори във формат JSON, които се използват за динамично обновяване на интерфейса.

## 4. Използвани технологии

### 4.1 Основни технологии

- HTML5 – структуриране на съдържанието;
- CSS – оформление и стил;
- JavaScript – клиентска логика;
- PHP – сървърна логика;
- MySQL – релационна база данни.

### 4.2 Среда за разработка

- Apache (локален сървър или Docker контейнер);
- phpMyAdmin – управление на базата данни;
- Visual Studio Code – среда за разработка;

### 4.3 Операционна система

Приложението може да бъде разработвано и стартирано под Windows.

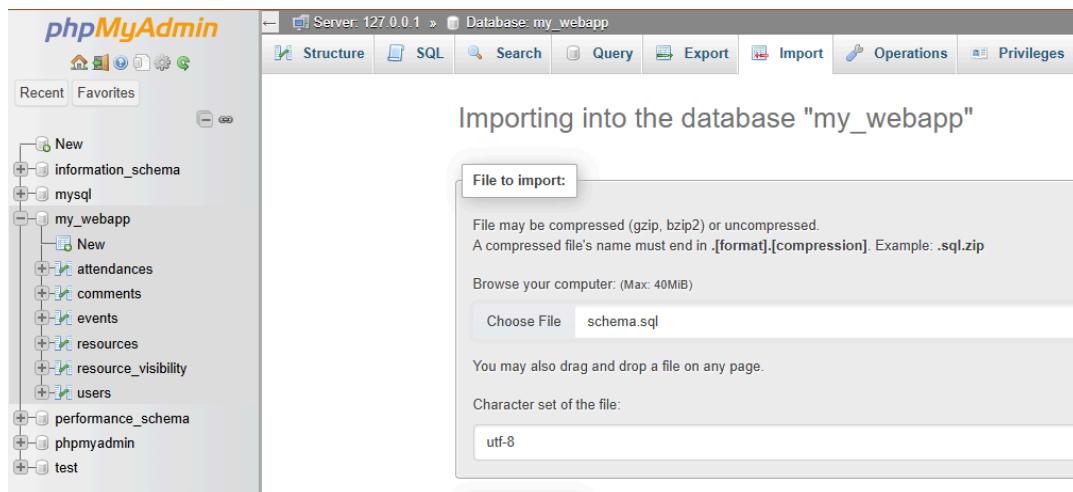
### 4.4 Използвани библиотеки и модули

- PDO (PHP Data Objects);
- password\_hash / password\_verify;
- Fetch API;
- Chart.js (визуализация на статистики).

## 5. Инсталация, настройки и DevOps

Инсталацията на приложението става по следния начин:

- Разархивиране на zip файла в xampp\htdocs<sup>1</sup>. Приемам че е разархивирано в папка 1MI0800189\_5MI0800232\_8MI0800231\_final на указаното място.
- След стартиране на Apache и MySQL в XAMPP, трябва да се отиде на localhost/phpmyadmin, да се създаде нова база данни с името "my\_webapp" и да се вмъкне (import) schema.sql от папката на проекта app\db.



(Какъв трябва да е крайния резултат след вмъкването)

- Вече може да се отиде на началната страница на [localhost/1MI0800189\\_5MI0800232\\_8MI0800231\\_final/app/pages/login.php](http://localhost/1MI0800189_5MI0800232_8MI0800231_final/app/pages/login.php).

Примерни данни за вход за ролята:

- студент:

**име:** maria.nikolova@gmail.com  
**парола:** 0MI0000000

- учител:

**име:** ivan.ivanov@uni.bg  
**парола:** test123

И вече приложението може да използва по предназначение от съответно студентски или учителски профил.

<sup>1</sup> За по-чист вариант на използваните URL на приложението е възможно и разархивиране на място по предпочтение на четящия. След това трябва да се редактира Apache(httpd.conf) от Config бутона на Apache в XAMPP. Там да се промени DocumentRoot и <Directory ...> да сочат към app папката на приложението. Така вече не е нужно да се пише 1MI0800189\_5MI0800232\_8MI0800231\_final/app в пътя за достигане на всички страници.

## 6. Кратко ръководство на потребителя

### За студент:

- ❖ **Влизане в профил** – В системата се влиза с предварително създаден от администратор профил. За студентите предварително зададената парола е факултетният им номер, като имат възможност да я сменят (и това силно се препоръчва).

Логотип на УИТ

Влизане в профил

Потребителско име (e-mail)

Парола

Вход

Студент

Мария Николова

ФИ: 1MI0800185  
e-mail: maria николова@gmail.com

Присъствие на учебни занятия в %

0	10	20	30	40	50	60	70	80	90	100
Yellow bar representing attendance percentage.										

- ❖ **Страница „за мен“** - След успешно влизане в системата се вижда страницата с лична информация за студента – име, факултетен номер, имейл, както и статистика за общото присъствие на учебни занятия в %. В ляво е менюто, през което може да се обходят останалите страници.

Логотип на УИТ

Промяна на парола

Стара парола

Нова парола

Запази промяна

Студент

Профил

+ Добави

Упражнение 14.01	14.01.2026 г.	09:00
Упражнение 11.01	11.01.2026 г.	15:00
Лекция 07.01	7.01.2026 г.	10:00
Лекция 05.01	5.01.2026 г.	15:00
Лекция 04.01	4.01.2026 г.	10:00
Лекция 02.01	2.01.2026 г.	10:00

- ❖ **Мои събития** – Студентите имат възможност да преглеждат списък от събитията, на които са присъствали. При клик върху всяко събитие се отваря неговата страница. Ако студентът е бил в списъка с присъстващи за дадено събитие, но впоследствие преподавателят го е премахнал, това събитие вече не се показва в „Мои събития“.

- ❖ **Страница на събитие** – Страницата на дадено събитие съдържа информация за темата му, датата, преподавателя, броя присъствали. При клик върху бутона „Присъствали: X“ се отваря списък на

присъствалите студенти. По-долу има раздел „Ресурси“, в който както преподавателят, така и студентите могат да качват учебни материали (има две отделни секции). Има и раздел „Коментари“, в който студентите могат да споделят впечатленията си и да задават въпроси.

The screenshot shows the student profile interface. On the left is a sidebar with a logo and buttons for 'За мен' (About me), 'Мои събития' (My events), 'Смяна на парола' (Change password), and 'Изход' (Logout). The main area is titled 'Студент' (Student) and 'Профил' (Profile). It displays information about 'Лекция 05.01' taught by 'Иван Петров' with 'Основи на PHP' information. A button 'Приসъствали: 5' (Present: 5) is shown. Below is a list of students present: Георги Димитров (1MI0800186), Иван Георгиев (1MI0800184), and Мария Николова (1MI0800185).

The screenshot shows the student profile interface. On the left is a sidebar with a logo and buttons for 'За мен' (About me), 'Мои събития' (My events), 'Смяна на парола' (Change password), and 'Изход' (Logout). The main area is titled 'Студент' (Student) and 'Профил' (Profile). It displays the 'Ресурси' (Resources) section with links to 'Основи на PHP – лекция', 'Запис от лекция: Основи на PHP', and 'Финални оценки'. Below is the 'Ресурси от студенти' (Students' resources) section, which includes a form for adding new resources: 'Добавяне на ресурс' (Add resource), 'Заглавие:' (Title:), 'URL:', 'Тип на ресурса:' (Type of resource: pdf), and a 'Добави' (Add) button.

The screenshot shows the student profile interface. On the left is a sidebar with a logo and buttons for 'За мен' (About me), 'Мои събития' (My events), 'Смяна на парола' (Change password), and 'Изход' (Logout). The main area is titled 'Студент' (Student) and 'Профил' (Profile). It displays the 'Коментари' (Comments) section with comments from Ana Петрова, Борис Велинов, and Мария Николова, each with a timestamp. Below is a comment input field 'Добави коментар...' (Add comment...) and a 'Изпрати' (Send) button.

❖ **Излизане от профил – от опция „Изход“ в менюто**

**За преподавател**

- ❖ **Влизане в профил** – В системата се влиза с предварително създаден от администратор профил. Преподавателите имат еднаква предварително зададената парола, като имат възможност да я сменят (и това силно се препоръчва).
- ❖ **Страница „за мен“** - След успешно влизане в системата се вижда страницата с лична информация за преподавателя – име, катедра, имейл, както и статистика за средното присъствие на студентите на неговите учебни занятия в %. В ляво е менюто, през което може да се обходят останалите страници.

The screenshot shows a teacher's profile page. At the top, there are tabs for "Преподавател" (Teacher) and "Профил" (Profile). On the left, a sidebar menu includes "За мен" (About me), "Мои събития" (My events), "Студенти" (Students), "Статистика" (Statistics), "Смяна на парола" (Change password), and "Изход" (Logout). The main content area displays a placeholder user icon, the name "Иван Иванов", and the text "Катедра: Информатика" and "e-mail: ivan.ivanov@uni.bg". Below this is a chart titled "Средно присъствие на мои учебни занятия в %" (Average attendance at my lectures in %) with a value of approximately 12%.

- ❖ **Смяна на парола** – Преподавателите имат опцията да сменят паролата си от менюто – „Смяна на парола“. Това става чрез потвърждаване на старата парола и въвеждане на новата.
- ❖ **Мои събития** – Преподавателите имат възможност да преглеждат списък от събитията, които са създали. При клик върху всяко събитие се отваря неговата страница. При клик върху бутона „Добави“ се отваря страница за създаване на ново събитие.

The screenshot shows the "My events" page. At the top, there are tabs for "Преподавател" (Teacher) and "Профил" (Profile). On the left, a sidebar menu includes "За мен" (About me), "Мои събития" (My events), "Студенти" (Students), "Статистика" (Statistics), "Смяна на парола" (Change password), and "Изход" (Logout). The main content area features a button "+ Добави" (Add) and a table listing four events:

Събитие	Дата	Време
Лекция 04.01 (част 2)	4.01.2026 г.	12:00
Лекция 04.01	4.01.2026 г.	10:00
Лекция 03.01	3.01.2026 г.	10:00
Лекция 02.01	2.01.2026 г.	10:00

- ❖ **Създаване на ново събитие** – При създаването на ново събитие, преподавателят трябва да попълни заглавието му, кратко описание за него, началния и крайния му час. Има възможност да качи файл с присъствали студенти, който е импортиран от Big Blue Button – така лесно и удобно се прехвърля информацията за присъствието на онлайн учебни занятия.

Преподавател

Заглавие  
Кратко описание  
Започва в  
Приключи в  
Избор на файл [Нама избран файл]  
Списък със студенти  
Импорт

Профил

- ❖ **Списък със студенти** - Преподавателите могат да видят списък от всички студенти в системата от менюто – „Студенти“. Те имат възможност също да добавят нов студент чрез бутона „Добави“.

Преподавател

+Добави	
Александър Михов	6MI8877665
Ана Петрова	1MI0800181
Борис Велинов	1MI0800182
Весела Иванова	1MI0800183
Виктория Павлова	0MI1100223
Георги Димитров	465826
Георги Димитров	1MI0800186

Профил

- ❖ **Добавяне на нов студент в системата** – Преподавателят може да добави единичен студент, като попълни полета за имената, имейла и факултетния му номер, както и да качи файл в .csv формат, съдържащ тази информация за множество студенти.

Преподавател

Първо име  
Фамилия  
Имейл  
Факултетен номер  
Импорт

Профил

### Вмъкване на единичен студент

Преподавател

Имейл  
Факултетен номер  
Импорт

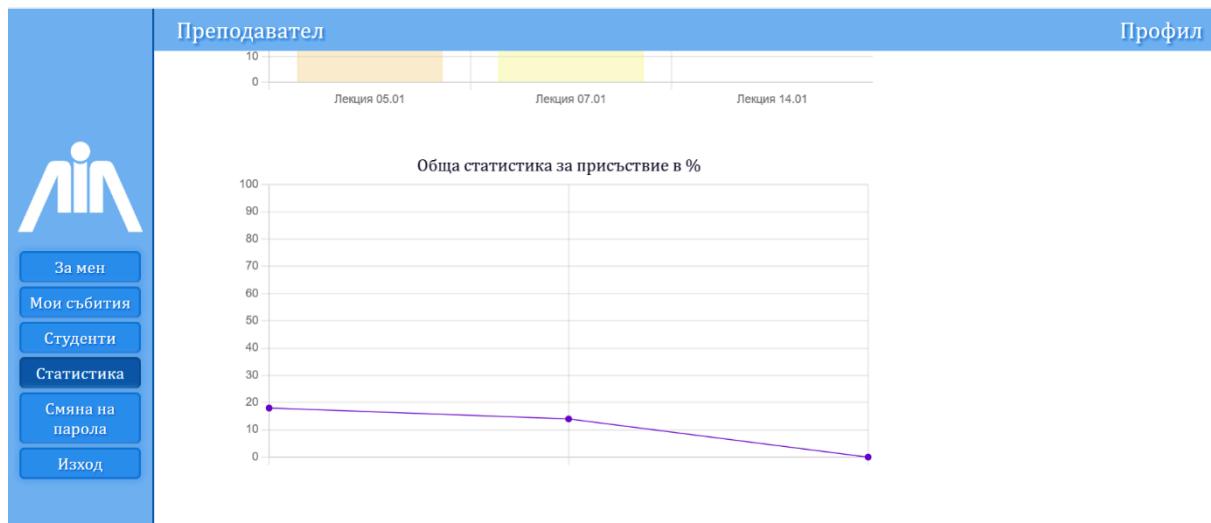
Профил

### Вмъкване на мнозина студенти

Избор на файл [Нама избран файл]  
Списък със студенти  
Импорт

- ❖ **Страница на събитие** – Страницата на дадено събитие съдържа информация за темата му, датата, преподавателя, броя присъствали. При клик върху бутона „Присъствали: X“ се отваря списък на присъствалите студенти. Преподавателят може да премахне студент от този списък, като кликне върху него. При повторно кликване студентът отново се отбелязва за присъствал. Освен това може да се добави ръчно нов студент към списъка от формата „Добавяне на студент“. По-долу има раздел „Ресурси“, в който както преподавателят, така и студентите могат да качват учебни материали (има две отделни секции). Има и раздел „Коментари“, в който преподавателят може да отговаря на въпроси, да сподели допълнителна информация и да види мнението на студентите.

- ❖ **Статистика** – В меню – „Статистика“ преподавателят може да види още статистики за неговите учебни занятия. Първата графика изобразява присъствието в % на последните до 10 негови събития, а втората показва изменението на присъствието за всички негови събития.



- ❖ **Излизане от профил** – от опция „Изход“ в менюто

## **7. Примерни данни**

За целите на тестването и демонстрацията на функционалностите на системата са подгответи примерни данни, които позволяват пълноценно използване на приложението без необходимост от допълнителна конфигурация. В базата данни са налични примерни потребителски акаунти за двете основни роли – студент и преподавател. За студентските акаунти началната парола е факултетният номер на съответния студент, а за преподавателските – предварително зададена обща парола, която може да бъде променена след първото влизане. Това позволява лесно тестване на ролевия достъп и основните функционалности на системата.

Освен потребителски акаунти, системата съдържа и примерни учебни събития с предварително попълнени присъствия, ресурси и коментари. SQL скриптовете за създаване и инициализиране на базата данни се намират в директория db/, като файлът schema.sql съдържа структурата на таблиците, а допълнителни скриптове с тестови записи могат да бъдат използвани за бързо зареждане на примерни данни. Примерни файлове за импорт на студенти и присъствия (в .csv и текстов формат) са разположени в отделна директория и служат за демонстрация на функционалностите за масово добавяне на данни и интеграция с външни системи.

## **8. Описание на програмния код**

Архитектурата е разделена логически на клиентска и сървърна част, като комуникацията между тях се осъществява чрез HTTP заявки и JSON отговори.

Структурата на проекта е организирана модулно, като всеки тип функционалност е отделен в самостоятелна директория.

### **8.1 Структура на директориите**

```
app/
  └── pages/      - PHP страници (UI)
  └── php/        - Backend логика и API endpoints
  └── js/         - JavaScript функционалност (frontend)
  └── css/        - CSS стилове
  └── db/          - SQL схема на базата данни
  └── images/     - Изображения и икони
```

### **8.2 Директория pages/ – потребителски интерфейс**

Тази директория съдържа PHP файлове, които визуализират страниците на приложението и зареждат съответните JS и CSS ресурси.

#### **Основни файлове:**

1. login\_page.php – страница за вход в системата
2. event\_list\_page.php – списък със събития
3. event\_details\_page.php – детайли за конкретно събитие
4. event\_creation\_page.php – създаване на ново събитие
5. teacher\_statistics\_page.php – статистики за преподавател
6. user\_page.php – потребителски профил

Тези файлове не съдържат бизнес логика, а служат като контейнер за визуализация.

### **8.3 Директория `php/` – сървърна логика (backend)**

Тази директория съдържа всички PHP скриптове, които обработват заявки от клиента и работят с базата данни.

#### **Основни модули**

##### **1. Автентикация и потребители**

- `login.php` – вход в системата
- `log_out.php` – изход
- `user.php` – информация за потребител
- `change_password.php` – смяна на парола

##### **2. Събития**

- `event.php` – информация за събитие
- `event_list.php` – списък със събития
- `event_create.php` – създаване на събитие
- `event_add_users.php` – добавяне на студенти към събитие
- `toggle_attendance.php` – промяна на присъствие

##### **3. Ресурси**

- `add_resource.php` – добавяне на ресурс
- `event_resources.php` – извличане на ресурси

##### **4. Коментари**

- `add_comment.php` – добавяне на коментар
- `event_comments.php` – извличане на коментари

##### **5. Статистики**

- `stats_student.php` – статистика за студент
- `stats_teacher.php` – статистика за преподавател
- `view_student_stats.php` – статистика за студент, когато го преглежда преподавател

#### **Примерен фрагмент от програмен код**

##### **Зареждане на събитие (`event.php`):**

```
$eventStmt = $pdo->prepare("SELECT  
    e.id,  
    e.title,  
    e.description AS info,  
    e.start_at,  
    e.end_at,  
    CONCAT(t.first_name, ' ', t.last_name) AS teacher_name  
FROM events e  
JOIN users t ON t.id = e.created_by  
WHERE e.id = ?  
LIMIT 1");
```

##### **Описание:**

Този фрагмент извлича детайлна информация за дадено събитие, като използва prepared statement за защита от SQL injection. Данните се използват за визуализация на страницата за детайли на събитието.

## 8.4 Директория js/ – клиентска логика

JavaScript файловете отговарят за динамичното поведение на страниците и комуникацията със сървъра.

### Основни модули:

1. event-details.js – зареждане на детайли за събитие
2. attendance-list.js – визуализация и управление на присъствия
3. add-resource.js – добавяне на ресурси
4. add-comment.js – добавяне на коментари
5. user-stats.js – статистики за студент
6. teacher-stats.js – статистики за преподавател

Последните два модула използват за визуализация на статистическите данни библиотеката **chart.js**.

### Примерен JS фрагмент

#### Fetch заявка към backend:

```
const res = await fetch(`./php/event.php?eventid=${eventId}`);
const data = await res.json();
```

#### Описание:

Клиентът изпраща GET заявка към PHP endpoint, който връща JSON отговор. Данните се използват за динамично обновяване на HTML съдържанието без презареждане на страницата.

## 8.5 Директория css/ – визуален стил

Всеки основен еcran има отделен CSS файл, което улеснява поддръжката и разширяемостта на проекта.

### Примери:

- event\_details\_styles.css
- login\_styles.css
- stats\_teacher\_styles.css

## 8.6 База данни

Файлът db/schema.sql съдържа структурата на базата данни, включваща таблици за:

- потребители
- събития
- присъствия
- ресурси
- коментари

Връзката с базата данни се осъществява чрез **PDO**, дефиниран в php/db.php.

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

### **9.1 Ограничения на приложението**

- Приложението работи в рамките на локална среда и не използва външни API услуги.
- Ограничена защита срещу злонамерени заявки.
- Данните не се кешират, което може да доведе до излишни заявки към базата.
- Приложението не поддържа многоезичност.
- Няма защита от злонамерени атаки чрез линкове за ресурси.
- Няма разделение по курсове – всички преподаватели в системата споделят студентите.

### **9.2 Възможности за бъдещо развитие**

- Разделение по курсове
- Подобряване на сигурността
- Добавяне на система за известия (email, in-app нотификации)
- Реализация на детайлни потребителски профили.
- Въвеждане на търсene и филтриране на събития и студенти
- Поддръжка на REST API за външни клиенти

## **10. Използване на AI – как и защо**

В процеса на разработка на проекта е използван изкуствен интелект (AI) като помощен инструмент с ограничена и консултативна роля. Основната цел на използването му беше помош при откриване и анализ на грешки в програмния код, както и генериране на примерни тестови данни за базата от данни. Беше използван и за оძнакяване на поясняващи коментари в кода.

### **10.1 Откриване и анализ на грешки**

**AI е използван за:**

- анализ на грешки при комуникацията между frontend и backend;
- разясняване на HTTP статус кодове и невалидни JSON отговори;
- подпомагане при откриване на синтактични и логически грешки в PHP и JavaScript код;
- диагностика на проблеми, свързани със SQL заявки и работа с базата данни.

### **10.2 Генериране на примерни данни**

**Изкуственият интелект е използван и за:**

- създаване на примерни тестови записи за базата от данни;
- генериране на реалистични демонстрационни данни за потребители, събития, ресурси и присъствия;
- улесняване на тестването на функционалностите на системата.

### **10.3 Контрол върху използването на AI**

Всички резултати, предоставени от AI, бяха използвани като помощни предложения и бяха прегледани и адаптирани от нас. AI не е използван за автоматично създаване на основната логика или архитектурата на приложението.

## **11. Какво научих**

**Стефанка Манахова, 1MI0800189:**

За мен най-важното, което научих по време на разработването на проекта, е как да работя ефективно в екип, как става разпределението на ролите и колко е важна комуникацията. Също така научих много за уеб програмирането, за особеностите на Javascript и PHP и колко е трудно да кажеш „готов съм“, когато винаги има какво още да се добави.

**Румяна Иванова, 8MI0800231:**

По време на разработването на проекта научих как се изгражда цялостно уеб приложение с разделение между клиентска и сървърна част. Развих уменията си за работа с PHP, JavaScript и MySQL, както и научих как да управлявам комуникация между frontend и backend чрез HTTP заявки и JSON. Разбрах колко важни са сесиите, контрола на достъпа и защитените заявки към базата данни. Работата по проекта ми помогна също да осъзная колко кючова е добрата организация на кода, планирането и тестването, особено при паралелна работа с останалите хора от екипа.

**Момчил Младенов, 5MI0800232:**

Най-поучителното за мен е работенето по чуждите стандарти, за да моите добавки са възможно най-разбираеми за колегите ми. Docker отне известно време, докато си го припомня и да направя най-удачният избор на изображение за нужните контейнери, които да са близки до XAMPP.

## **12. Dev(sec)Ops – подкарване на проекта - особенности**

**12.1 – Линк към проекта в наш гитхъб за поддържане на различните версии на приложението:**

<https://github.com/ruby-fencerin/Web-project>

**12.2. Инструкции и коментари при подкарването на проекта във ФМИ – настройки на докер/примерни данни – име на sql файл (който може да е различен от sql файл-а за локално подкарване).**

С цел по-лесно използване на проекта и избягване на любимата изказване на всеки програмист “Работи на моята машина” е направен и начин да се пусне чрез Docker и Docker-compose. За целта е необходимо за Windows инсталацията на Docker Desktop и след това от основната папка да се пуснат командите:

```
docker compose build  
docker compose up -d
```

Така се стартира приложението, като разликата спрямо XAMPP имплементацията е, че базовото приложение е се намира на порт **8080**, а **phpmyadmin - 8081**. Един начин да се влезне в phpmyadmin е чрез акаунта user и парола user\_password. Използването става аналогично на бележката.

За да се прекратят Docker контейнерите е необходимо да се използва само командалата от същата локация като указаната по-рано:

```
docker compose down
```

Предал (подпис): .....

Предал (подпис): .....

/1MI0800189 Стефанка Манахова КН 2 гр./

/5MI0800232 Момчил Младенов КН 3 гр./

Предал (подпис): .....

Приел (подпис): .....

/8MI0800231 Румяна Иванова КН 2 гр./

/проф. д-р Милен Петров/

## 13. Използвани източници

### PHP и MySQL

1. **PHP Manual – Official Documentation**  
Автор: The PHP Documentation Group  
Заглавие: PHP Manual  
Линк: <https://www.php.net/manual/en/>  
Описание: Официалната документация на PHP, използвана за работа със сесии, PDO, prepared statements и JSON отговори.
2. **PHP: PDO (PHP Data Objects)**  
Автор: The PHP Documentation Group  
Заглавие: PDO – PHP Data Objects  
Линк: <https://www.php.net/manual/en/book pdo.php>  
Описание: Източник за сигурна комуникация с MySQL база данни чрез prepared statements.
3. **MySQL 8.0 Reference Manual**  
Автор: Oracle Corporation  
Заглавие: MySQL 8.0 Reference Manual  
Линк: <https://dev.mysql.com/doc/refman/8.0/en/>  
Описание: Използван при проектиране на структурата на базата данни и SQL заявки.

### JavaScript и AJAX комуникация

4. **MDN Web Docs – JavaScript**  
Автор: Mozilla Contributors  
Заглавие: JavaScript Guide  
Линк: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>  
Описание: Основен източник за работа с JavaScript, масиви от обекти, събития и DOM манипулация.
5. **MDN Web Docs – Fetch API**  
Автор: Mozilla Contributors  
Заглавие: Using the Fetch API  
Линк: [https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch)  
Описание: Използван при реализиране на комуникацията между frontend и backend чрез HTTP заявки и JSON.
6. **MDN Web Docs – JSON**  
Автор: Mozilla Contributors  
Заглавие: Working with JSON  
Линк: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>  
Описание: Източник за сериализация и обработка на данни между клиента и сървъра.

### HTML и CSS

7. **MDN Web Docs – HTML**  
Автор: Mozilla Contributors  
Заглавие: HTML: HyperText Markup Language  
Линк: <https://developer.mozilla.org/en-US/docs/Web/HTML>  
Описание: Използван за създаване на структурата на уеб страниците.
8. **MDN Web Docs – CSS Flexbox**  
Автор: Mozilla Contributors  
Заглавие: CSS Flexible Box Layout  
Линк: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout)  
Описание: Използван за подравняване и позициониране на елементи в интерфейса.

### Допълнителни ресурси

9. **W3Schools – PHP & JavaScript Tutorials**  
Автор: W3Schools  
Заглавие: PHP Tutorial / JavaScript Tutorial  
Линк: <https://www.w3schools.com/>  
Описание: Използван като допълнителен учебен ресурс и бърза справка.
10. **Docker – Docker**  
Автор: Docker Inc.  
Заглавие: Manuals  
Линк: <https://docs.docker.com/manuals/>  
Описание: За припомняне на docker и docker compose и прилагането на добrите им практики.

## **Ресурси от курса по Уеб технология**

1. Web Technologies – Introduction  
Заглавие: Introduction to Web Technologies  
Линк:  
[<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/INTRO.html?read=1#1>](<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/INTRO.html?read=1#1>)  
Описание: Въведение в основните уеб технологии и принципи на уеб приложенията.
2. Web Technologies – HTML Forms  
Заглавие: HTML Forms  
Линк:  
[<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/Forms.html?read=1#1>](<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/Forms.html?read=1#1>)  
Описание: Основи на HTML формуларите и изпращането на данни към сървър.
3. Web Technologies – PHP Basics  
Заглавие: PHP  
Линк:  
[<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/PHPhtml?read=1#1>](<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/PHPhtml?read=1#1>)  
Описание: Основни концепции в PHP за сървърна логика.
4. Web Technologies – PHP & SQL  
Заглавие: PHP and SQL  
Линк:  
[<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/PHP-SQL.html?read=1#1>](<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/PHP-SQL.html?read=1#1>)  
Описание: Работа с бази данни чрез PHP и SQL.
5. Web Technologies – Authentication  
Заглавие: Authentication  
Линк:  
[<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/auth.html?read=1#1>](<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/auth.html?read=1#1>)  
Описание: Управление на потребителски сесии и достъп.
6. Web Technologies – CSS Basics  
Заглавие: CSS – Part 1  
Линк:  
[<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/CSS-1.html?read=1#1>](<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/CSS-1.html?read=1#1>)  
Описание: Основи на CSS за стилизиране на уеб страници.
7. Web Technologies – CSS Advanced  
Заглавие: CSS – Part 2  
Линк:  
[<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/CSS-2.html?read=1#1>](<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/CSS-2.html?read=1#1>)  
Описание: Разширени техники за оформление с CSS.
8. Web Technologies – JavaScript Basics  
Заглавие: JavaScript – Part 1  
Линк:  
[<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/JS-1.html?read=1#1>](<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/JS-1.html?read=1#1>)  
Описание: Въведение в JavaScript и работа с DOM.
9. Web Technologies – JavaScript Advanced  
Заглавие: JavaScript – Part 2  
Линк:  
[<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/JS-2.html?read=1#1>](<https://webtech.w3c.fmi.uni-sofia.bg/w15labs/JS-2.html?read=1#1>)  
Описание: Асинхронен JavaScript и по-сложна логика.

## **Добавени функционалности и подобрения в системата**

**Имена:** Стефанка Манахова, Момчил Младенов, Румяна Иванова  
**ФН:** 1MI0800189 5MI0800232 8MI0800231  
**Начална година:** 2022 **Програма:** бакалавър КН **Курс:** 4  
**Тема:** Система за записване и управление на присъствие по време на презентации, упражнения или лекции  
**Дата:** 10.02.2026 **Предмет:** w25prj\_KN\_final  
**имейл:** rumiana.ivanova24@gmail.com  
**преподавател:** проф. д-р Милен Петров

В рамките на доразвитието на проекта бяха реализирани редица нови функционалности, които разширяват възможностите за управление на студенти, събития и присъствия, както и подобряват визуализацията и статистиката в системата.

### **Академична информация за студентите**

В базата данни беше добавена нова таблица, свързана със студентите, която съдържа академична информация – **специалност, група, курс и година на прием**. Тези данни се използват в цялата система и позволяват по-прецизно групиране и филтриране на студентите.

### **Академичната информация:**

- се визуализира в страницата „За мен“ на студента;
- се показва и в списъка със студенти, достъпен за преподавателите.

### **Подобрено изчисляване и визуализация на присъствията**

Присъствието на всеки студент вече се изчислява динамично, като процент от броя събития, в които студентът е вписан, спрямо реално отбеляните му присъствия.

Общият процент присъствие се визуализира директно в списъка със студенти, като се използва цветова индикация (от нисък към висок процент), улесняваща бързата ориентация на преподавателя.

### **Разширено създаване и управление на събития**

При създаване на ново събитие преподавателят вече има възможност:

- да добавя цели групи или специалности студенти към събитието;
- автоматично да създаде присъствени записи за всички включени студенти.

### **В рамките на създадено събитие е реализирана и възможност за:**

- импорт на присъствен списък от BigBlueButton (BBB), който автоматично отбелязва кои студенти са присъствали;
- ръчно добавяне на множество студенти, чрез въвеждане на списък от факултетни номера, разделени със запетайки.
- масови операции и импорт

Системата беше разширена с функционалности за масова обработка на данни, включително:

- импорт на събития за цяла учебна година чрез CSV файл, към които впоследствие могат да се добавят присъствия и ресурси;
- масово записване на студенти към събития по групи и специалности;
- масово добавяне на учебни материали към събитията, улесняващо работата на преподавателите при управление на ресурси.

#### **Подобрения при добавяне и импорт на студенти**

Функционалността за добавяне на студенти беше значително разширена и стабилизирана. Реализирана е възможност за масово добавяне на студенти чрез CSV файлове, като системата поддържа коректна обработка на различни формати на нов ред и валидира съдържанието на всеки запис. Всеки студент се добавя с пълна академична информация – специалност, група, курс и година на прием, която се съхранява в отделна таблица в базата данни.

Допълнително е въведена строга валидация на входните данни, включително проверка за коректен формат на CSV редовете и допустими стойности за курс, група и година на прием. Това гарантира целостта на данните и предотвратява некоректни записи в системата.

#### **Контрол на дублирани записи и защита на данните**

В системата е реализирана логика за предварителна проверка за дублиращи се студенти, като се следи за уникалност на факултетния номер и имейл адреса. При опит за добавяне на вече съществуващ студент системата връща информативно съобщение за грешка, без да се извършва частичен запис в базата данни.

#### **Подобрена обработка на присъствия и събития**

Обработката на присъствията беше оптимизирана чрез използване на автоматични операции при добавяне на студенти към събития. При масово добавяне или импорт на студенти се създават присъствени записи за всички включени участници, като системата позволява последващо автоматично или ръчно отбелязване на присъствалите.

Импортът на присъствени списъци от BigBlueButton беше интегриран по начин, който автоматично синхронизира присъствията със съществуващите студенти, като се избягва дублиране и се актуализира текущият статус на присъствие.

#### **Повишена надеждност и поддръжка на системата**

В процеса на доразвитие са добавени допълнителни механизми за откриване и диагностика на грешки, включително по-информативни съобщения при проблеми с базата данни и входните данни. Това улеснява както крайния потребител, така и бъдещата поддръжка и разширяване на системата.

Предал (подпись): .....

Предал (подпись): .....

/1MI0800189 Стефанка Манахова КН 2 гр./

/5MI0800232 Момчил Младенов КН 3 гр./

Предал (подпись): .....

Приел (подпись): .....

/8MI0800231 Румяна Иванова КН 2 гр./

/проф. д-р Милен Петров/