**Имена: Силвана Николова, Катерина Тончева** **фн:**  *0MI0600152, 8MI0600214***Начална година:** *2024/2025* **Програма:** бакалавър, (СИ) **Курс: 3**  
**Тема: 9.3. Генериране на разписание  
Дата: 2025-06-04 Предмет: Web технологии имейл: silvana.nikolova2003@gmail.com, katerinatoncheva4@gmail.com**

**преподавател:** доц. 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 използваниресурси**, архива да се казва 9999\_final.zip. (Успех!). (Редактирайте маркираните зони в жълто с коректната информация)**

# ТЕМА: 9.3. Генериране на разписание

## 1. Условие

## Създайте уеб приложение за управление и генериране на разписание за студентски презентации. Целта на приложението е да даде възможност на потребителите (студенти) да се регистрират, да преглеждат налични презентации и да избират кои от тях представляват интерес за тях. Въз основа на тези интереси, системата генерира персонализирано разписание, което може да се “export”-ва в различни формати.

## 2. Въведение – извличане на изисквания

Целта на настоящия проект е да се създаде уеб базирана система, която да подобри процеса по презентиране на студентски проекти чрез интерактивен достъп до информацията. Чрез нея студентите ще могат да разглеждат както пълното разписание на презентациите, така и персонализирано – според своите предпочитания и интереси. Системата ще предоставя възможност за отбелязване на интерес към определени презентации, като например: „трябва да отида“, „може би ще отида“ и други. Това ще позволи на всеки потребител да създаде свое собствено разписание, което е филтрирано спрямо неговите нужди.

Освен това, потребителят ще има възможност да експортира своето разписание във формат по избор – PDF, CSV или архив (.zip). Разписанията са достъпни само след вход в системата, което гарантира защита и индивидуален достъп.

Роли в системата:

* Потребител (студент) – може да се регистрира, влиза в системата, избира интересуващи го презентации и генерира разписание.
* Система – обработва потребителски предпочитания, визуализира разписанията и управлява експорта.

Функционални изисквания:

* Регистрация и вход в системата със защита чрез сесия.
* Преглед на цялостен график с всички презентации.
* Възможност за маркиране на интерес към определени презентации.
* Автоматично генериране на персонално разписание според избраните интереси.
* Експортиране на цялостно/персонално разписание в PDF, CSV или ZIP формат.

Нефункционални изисквания:

* Работа с база данни чрез PDO за сигурна връзка и по-добра поддръжка.
* Интерактивен интерфейс с JavaScript за скриване/показване на форми.
* Ограничен достъп до данни чрез сесии.
* Лесна интеграция и стартиране чрез локален XAMPP сървър.
* Възможност за разширение с бъдещи функционалности.

Чрез тази система се улеснява достъпът до информация, подобрява се процесът на планиране и се осигурява удобство за студентите.

## 3. Теория – анализ и проектиране на решението (декомпозиция на приложението, модули, компоненти, части на приложението)

Системата за управление и визуализация на разписание на презентации е уеб-базирано приложение, което следва клиент-сървър архитектура и е реализирано с използване на езиците PHP, HTML, CSS и JavaScript, както и база данни MySQL.

Приложението е разделено логически на три основни слоя:

* Представителен слой (Frontend) – отговаря за визуализиране на съдържанието към потребителя;
* Бизнес логика (Backend) – обработва заявки, прилага бизнес правила и комуникира с базата данни;
* Слой за достъп до данни (Data Access Layer) – реализиран чрез репозитори класове, които извличат и записват данни в MySQL база.

Основни модули и компоненти

* Модул за регистрация и вход
* Форма за регистрация с валидация на входните данни
* Форма за вход със сесии и проверка на автентичност
* Контрол на достъпа до защитени страници
* Модул за преглед на разписания
* Преглед на цялостно разписание с всички презентации
* Преглед на персонализирано разписание въз основа на избрани предпочитания:
* Ще присъствам
* Може би ще присъствам
* Няма да присъствам
* Модул за предпочитания
* Модул за експортиране
* PDF версия – страница, подготвена за печат със стилизирано разписание
* CSV експортиране – файл с презентации в табличен вид, подходящ за електронни таблици
* ZIP архивиране – създаване на архив, съдържащ CSV файл, подходящ за споделяне
* База данни
* user – съдържа информация за потребителите
* presentation – представя всички налични презентации
* preference – свързва потребителите с техните предпочитания към презентации

Връзка между компонентите

* Всеки компонент работи с ясно дефинирани зависимости:
* Представителният слой използва HTML + CSS + JS, за да подава заявки чрез форми.
* Сървърният слой (PHP) приема заявките, валидира ги и извлича необходимата информация чрез класове в /repository.
* Резултатите се връщат обратно на клиента под формата на HTML страници, CSV файлове или PDF изгледи.

## 4. Използвани технологии

* Операционна система: Windows 10 / 11 (тествано с XAMPP)
* Сървърна среда: XAMPP 8.2.x (включва Apache 2.4, PHP 8.2, MySQL 8.0)
* Уеб браузъри: Google Chrome, Mozilla Firefox, Microsoft Edge
* HTML5 – за изграждане на структурата на страниците
* CSS3 – за стил и оформление
* JavaScript (ES6) – за динамична логика и скриване/показване на елементи
* PHP 8.2 – за обработка на заявки и логика на сървъра
* PDO (PHP Data Objects) – за сигурна работа с MySQL база данни
* PHP модули (разширения):
  + zip – необходимо за ZIP експортиране (трябва да е активирано в php.ini)
* MySQL 8.0
* Инструменти за разработка:
  + VS Code / IntelliJ – за редактиране на код
  + phpMyAdmin – за управление на базата данни
  + Git – за контрол на версиите

## 5. Инсталация, настройки и DevOps

Стъпки за инсталация и стартиране:

1. Инсталиране на XAMPP - Изтегли и инсталирай XAMPP (препоръчителна версия: 8.2 или по-нова).
2. Копирай проекта в htdocs/ директорията на XAMPP
3. (По избор) Активирай възможност за .zip експорти:
   1. отиди в папка xampp\php\ и потърси конфигурационен файл php.ini
   2. намери ред ;extension=zip и премахни точката и запетаята в началото на реда, така че да стане extension=zip
   3. запази промените във файла
4. Стартирай Apache и MySQL от XAMPP Control Panel
5. Създай база данни web\_schedule чрез phpMyAdmin, достъпвайки следната страница <http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=web_schedule&table=user>
6. Добави следната структура в config/config.ini:

*[db]*

*host = 127.0.0.1*

*name = web\_schedule*

*user = root*

*password =*

1. (По избор) Попълни базата с примерни презентации като посетиш [*http://localhost/Presentation-Generator/php/presentation/populate\_presentations.php*](http://localhost/Presentation-Generator/php/presentation/populate_presentations.php)
2. Посети <http://localhost/Presentation-Generator> в брауъра

## 6. Кратко ръководство на потребителя

1.Влизане в систематаA screenshot of a computer screen

Description automatically generatedA screenshot of a login form

Description automatically generatedA screenshot of a login form

Description automatically generated

При отваряне на приложението ще видите заглавие **„Университетско разписание“** и два бутона:

* **Вход** – за вече регистрирани потребители
* **Регистрация** – за нови потребители

Изберете подходящия според ситуацията.

**Вход в системата (за регистрирани потребители)**

* След натискане на бутона **„Вход“** ще се зареди форма, в която трябва да въведете потребителско име и парола. След като въведете коректни данни, натиснете зеления бутон **„Влез“**, за да достъпите таблото с презентации и предпочитания.

**Регистрация в системата (за нови потребители)**

* Ако нямате профил, натиснете бутона **„Регистрация“**. Попълнете следните полета: имейл адрес, потребитлско име, парола, потвърди парола. След като попълните всичко коректно, натиснете зеления бутон **„Регистрирай се“**.

### Разглеждане на всички презентации

A screenshot of a computer

Description automatically generated

Тук ще видите таблица с всички презентации. Всяка редица съдържа: заглавие на презентацията, име на презентатора, факултетен номер, дата, място и предпочитание – ще присъствам, няма да присъствам, може би. След като изберете предпочитание, то автоматично се записва. Можете да променяте избора си по всяко време.

### Предпочитания

A screenshot of a computer

Description automatically generated

Този изглед показва само онези презентации, за които вече сте избрали предпочитание. Информацията включва: Заглавие, дата и вашият избор. Изгледът се обновява автоматично при всяка промяна.

### Разпечатай

A screenshot of a computer

Description automatically generated

Изгледът **„Разпечатай“** е създаден, за да предостави удобен формат за печат на информацията от таблото – независимо дали това са всички презентации или само вашите предпочитания. За да изтеглите разписание, трябва да изберете една от двете опции – всички презентации или персонално. В първия случай ще се изтеглят всички презентации, които са налични в страницата „Всички презентации“. Във втория случай може да изберете кое предпочитание да се разпечата:

* Ще присъствам – Презентациите, които в текущия момент са отбелязани с „ще присъствам“.
* Може би - Презентациите, които в текущия момент са отбелязани с „може би“.
* И двете – Всички презентации, които в текущия момент са отбелязани с „ще присъствам“ и „може би“.

След като сте избрали желаните презентации, трябва да изберете в какъв формат да се разпечата разписанието – pdf, csv, zip като натиснете съответния бутон.

## 7. Примерни данни

Системата работи с предварително заредена със записи БД. За вкарване на всички презентации - [*http://localhost/Presentation-Generator/php/presentation/populate\_presentations.php*](http://localhost/Presentation-Generator/php/presentation/populate_presentations.php)*,* което ще изпълни скрипт и автоматично ще вкара данни.

Примерен запис за „презентация“ в БД :

* "title": "Машинно обучение с Python",
* "presenterName": "Ралица Генева",
* "facultyNumber": "0MI77889",
* "date": "2025-06-12",
* "place": "Зала 301"

## 8. Описание на програмния код

* HTML - изграждат **представителния слой (frontend)** на системата. Те отговарят за визуализацията на интерфейса към потребителя. Всеки от тях има конкретна роля в процеса на регистрация, вход, преглед и експортиране на разписания.
  + index.html – Страница за вход и регистрация. Това е началната страница на приложението, от която потребителят може да влезе в системата или да се регистрира.
  + dashboard.html – Основна навигационна страница. След успешен вход, потребителят попада в този **dashboard**, където може да разглежда презентациите, своите предпочитания и възможности за експортиране.
  + schedule.html – Статично HTML разписание (пълно). Представя **цялостно разписание на презентации**, форматирано за разпечатване.
  + personalSchedule.html – Статично HTML разписание (персонално). Аналогично на schedule.html, но визуализира само **персонализираното разписание** на потребителя според избраните презентации.
* Директория style - Съдържа CSS файлове, които отговарят за визуалното оформление на уеб приложението и осигуряват **удобна и приятна за ползване потребителска среда**. Всеки стилов файл е асоцииран с конкретна HTML страница и следва принципите на модулност и консистентност.
  + index.css – Стилове за начална страница (index.html). Оформя страницата за вход и регистрация.
  + dashboard.css – Стилове за основното табло (dashboard.html). Оформя изгледа на основната работна среда след вход – включително таблиците с презентации, навигационните бутони и опциите за експорт.
  + scheduleView.css – Стилове за страниците schedule.html и personalSchedule.html. Оформя изгледите на разписанията, подготвени за разпечатване (или запазване като PDF).
* Директория script -Съдържа JavaScript файлове, които изграждат **клиентската логика** на приложението. Те обработват потребителските действия, визуално управляват интерфейса и осъществяват комуникация със сървъра чрез fetch или XMLHttpRequest.
  + index.js – Управление на формите за вход и регистрация. Придава интерактивност на index.html. Превключва между формите и изпраща данни за вход/регистрация към API.
  + dashboard.js – Логика за dashboard.html. Основен клиентски контролер за работното табло с презентации.
  + login.js – Логика за вход. Осигурява вход с допълнителна валидация и защита срещу неподходящи символи чрез formatInput.
  + register.js – Логика за регистрация. Подобно на login.js, този скрипт използва допълнителна валидация за регистрацията.
  + logout.js – Скрипт за изход. Изпълнява заявка до php/api.php/logout и при успех връща потребителя в index.html.
* Директория php - Съдържа PHP файлове, които изграждат **бекенд логиката (business logic)** на приложението. Те приемат, валидират и обработват заявки от JavaScript и комуникират с MySQL базата чрез PDO.
  + api.php – Централен вход за всички API заявки. Точка за вход към сървърната логика. Насочва всяка заявка към правилната обработка.
  + router.php – Маршрутизиране на заявки. Клас Router, който интерпретира крайната част на URL заявката и я насочва към съответната функция от съответния PHP модул.
  + database.php – Клас за работа с базата данни. Инкапсулира всички операции с MySQL базата чрез PDO.
  + utility.php – Помощни функции - функции за валидация, защита и проверка на сесия.
* Директория php/repository - Съдържа класове, които инкапсулират логиката за работа с базата данни за всеки основен обект: **потребител**, **презентация** и **предпочитание**. Те използват database.php за достъп до PDO и изпълняват конкретни заявки.
  + user.php – Работа с потребителски акаунти. Управлява регистрацията, входа и проверките на потребителите.
  + presentation.php – Достъп до данни за презентации. Чете и записва информация за всички презентации в системата.
  + preference.php – Работа с потребителски предпочитания. Управлява таблицата preference, където се записват връзките между потребители и презентации.
* Директория php/content\_manager - Файловете в тази директория съдържат **основната логика на сървъра**, свързана с извличането на данни за таблото и управлението на предпочитанията. Всеки файл е директно извикван от маршрутизатора router.php.
  + dashboard.php – Зареждане на данни за таблото. Изпраща информация за всички презентации и текущите предпочитания на потребителя.
  + set\_preference.php – Запис на потребителско предпочитание.
  + load\_preferences.php – Зареждане на потребителските предпочитания. Извлича всички предпочитания на текущо вписания потребител и връща списък със заглавия, дати и тип предпочитание.
* Директория php/user\_access - Файловете в тази папка реализират **аутентикация и управление на сесии**. Всеки от тях е включен в маршрутизатора и обработва специфична заявка от типа вход, регистрация или изход.
  + login.php – Обработка на вход: Проверява подадените потребителско име и парола и стартира сесия при успешна автентикация.
  + register.php – Регистрация на нов потребител: Създава нов акаунт след валидиране на входните данни и хеширане на паролата.
  + logout.php – Изход от системата: Затваря сесията на потребителя и го изкарва от системата.
* Директория php/schedule - Съдържа файлове, които генерират **статично HTML съдържание** (динамично запълнено), което служи за визуализиране и разпечатване на цялостното или персонализираното разписание на презентации.
  + schedule.php – Пълно разписание. Генерира HTML страница със **всички презентации**, достъпна за всеки потребител.
  + personalSchedule.php – Персонализирано разписание. Генерира HTML страница със **само онези презентации**, които са отбелязани от потребителя като интересни (например „ще присъствам“).
* Директория php/export - Съдържа файлове, които реализират функционалността по **експортиране на разписания във външни формати** – CSV и ZIP. Те се извикват от интерфейса в dashboard.html.
  + export\_csv.php – Експортиране като CSV файл. Генерира CSV файл с данни за всички презентации или само за предпочитаните от потребителя.
  + export\_zip.php – Експортиране като ZIP с вграден CSV. Генерира ZIP архив, който съдържа CSV файл с разписание.
* Директория php/presentation - Съдържа файл за автоматично зареждане на тестови/примерни презентации в базата.
  + populate\_presentations.php – Файл за ареждане на примерни данни, който представлява скрипт за **масово вкарване на презентации** от външен .json файл в базата данни.
* Директория php/presentations\_data/presentations.json - json файл с примерни данни за презентации.
* Директория php/database-setup - Съдържа файлове с SQL скриптове за създаване и унищожаване на базата данни web\_schedule, използвана от системата.
  + db\_schema\_changelog.sql – Създаване на базата и таблиците.
  + database-destroy.sql – Изчистване на базата.
* Директория php/config - Конфигурационен файл
  + config.ini - Настройки за база данни и път към данни: Конфигурационен файл, използван от database.php и populate\_presentations.php за извличане на настройки при инициализация.

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

Приноси на студентите:

* Силвана Николова:
  + Добавяне на регистрация и вписване в системата
  + Добавяне на цялостно разписание на презентациите
  + Добавяне на лично разписание на презентациите, според проявен интерес
* Катерина Тончева:
  + Добавяне на скрипт за автоматично качване на презентациите в базата
  + Добавяне на филтри при експортиране на личното разписание
  + Добавяне на функционалност за експортиране на лично разписание в различни формати - pdf, csv, архив.

Ограничения:

* Сигурност, базирана единствено на сесии

Бъдещо разширение:

* Добавяне на роля „администратор“, който да може чрез уеб интерфейс да създава, редактира и изтрива презентации, както и да управлява потребителски акаунти.
* Възможност за промяна на имейл, парола и изтриване на акаунта през интерфейс, както и възстановяване на забравена парола.
* Добавяне на търсачка и филтри (по дата, място, презентатор и тема), което да улесни навигацията при голям обем данни.
* Имейл или браузър известия при наближаваща презентация, към която потребителят е отбелязал интерес.
* Поддръжка на регистрация и вход чрез Google акаунт за по-бърз достъп и защита.

## 10. Какво научих

* **Работа с тристепенна архитектура (Frontend, Backend, Database)**: Научих как да разделям логически кода на отделни слоеве и как всеки слой комуникира с останалите чрез дефинирани интерфейси.
* **Управление на сесии и потребителска автентикация:** Придобих практически опит в реализирането на сесии, защита на страници, както и в контрол на достъпа в зависимост от статуса на вход.
* **Интерактивност с JavaScript и динамично поведение**: Научих как да изграждам по-интерактивен интерфейс с помощта на JS – скриване и показване на елементи, изпращане на заявки с fetch и обработка на JSON отговори.
* **Конфигурируемост и добри практики в структурата на проекта:** Научих колко важно е да използвам конфигурационен файл (config.ini), да структурирам проекта модулно и четливо, така че той да бъде поддържан и лесен за разширяване.
* **Експорт на данни в различни формати:** Научих как да генерирам CSV и ZIP файлове чрез PHP, както и как да създам изглед, подходящ за експортиране в PDF чрез браузъра.

## 11. Използвани източници

* W3Schools –<https://www.w3schools.com/>
* Stack Overflow –<https://stackoverflow.com/>
* GeeksforGeeks – <https://www.geeksforgeeks.org/php/>

Предал (подпис): ………………………….

/*фн, имена, спец., група*/

Приел (подпис): ………………………….

/проф. д-р *Милен Петров*/