

## Задание за курсов проект

### RНР за напреднали - част 2

Да се създаде web приложение, което реализира презентация на CRUD интерфейс (наричан по-долу grid или data grid). Приложението презентира:

- Имплементация на форма за търсене в базата данни
- Презентация на резултата от търсенето в табличен вид
- Сортиране на резултата по определени критерии
- Възможност за контрол на страницирането (представяне резултата на „порции“)

Формата за търсене съдържа следните критерии, по които може да се търси:

- Име на студент – текстово поле, въвежда се свободен текст. Резултата се оценява, чрез допълване отдясно. Пример: при въведена фраза за търсене „Иван“, допустими резултати са: Иван Иванов, Иванка Петрова и т.н.
- Специалност – избира се от падащ списък с предварително дефинирани специалности в базата данни. Резултата се оценява само за конкретно избрана специалност, без значение курса на студентите.
- Курс – избира се от падащ списък с предварително дефинирани комбинации от курс и специалност. Пример: Първи курс, БИТ; Първи курс, Информатика; Първи курс Математика и Информатика и т.н. Резултата се оценява само за конкретно избрана комбинация от курс и специалност

Изисквания към резултата (т.е. съдържанието на табличните данни):

- Номер по ред
- Собствено име и фамилия на студента

- Име на курс – представлява комбинация от името на курса и името на специалността
- Данни за хорариум и оценка за три дисциплини (напр. Математика, Информатика и Физика) във следния вид:
  - Хорариум лекции във вид XX (YY), където XX е необходимия хорариум в часове зададен за конкретната дисциплина, а YY – реалното посещение на студента
  - Оценка на студента за текущата дисциплина – словом и цифром, според шестобалната система използвана в РБългария
- Обща стойност за хорариума в часове във формат XX (YY)
- Среден успех на студента

Сортиране на данните:

- Заглавните редове (т.нар. анетка на таблицата) са хипервръзки, които могат да влияят на сортирането на резултата
- До името на колоната следва да се визуализира посоката на сортиране (по подразбиране сортирането е във възходящ ред)
- Следните колони трябва да могат да бъдат сортирани: Номер по ред, Име и фамилия, Хорариум лекции и упражнения и Оценка

Контрол на страницирането - Непосредствено преди и след таблицата (grid-a) с резултата са разположени контроли за странициране на резултата, които съдържат:

- Хипер връзка за бърз преход към първа страница от резултата (ако е достъпна)
- Хипер връзка за бърз преход към предходна страница от резултата (ако е достъпна)
- Пореден номер на страница от резултата, която е визуализирана плюс контрол за бърз преход до произволно избрана страница от резултата (посредством поле за въвеждане номер на страница и бутон за изпращане на данните)

- Общ брой на страниците от резултата
- Хипер връзка за бърз преход към следваща страница от резултата (ако е достъпна)
- Хипер връзка за бърз преход към последната страница от резултата (ако е достъпна)

## **Допълнения към изискванията за курсовия проект от 2-ри триместър**

### ***CRUD интерфейс за управление на студенти, специалности, курсове и оценки***

Да се имплементират необходимите гридове и форми за създаване, преглед, редактиране и изтриване (C. R. U. D.) на записите, които участват в генерирането на таблицата от първата част на заданието.

#### **Форма за добавяне на специалност**

- Име
- Абревиатура

#### **Форма за добавяне на предмет**

- Наименование
- Хорариум лекции по норматив
- Хорариум упражнения по норматив

#### **Форма за добавяне на студент**

- Собствено име
- Фамилия
- Имейл адрес
- Факултетен номер
- Курс
  - SELECT с ID-номер и наименование на курса от номенклатурата с курсове

- Специалност
  - SELECT с ID-номер и наименование на специалността от номенклатурата със специалности
- Форма на обучение
  - SELECT с избор на една от двете дефинирани в БД ENUM стойности:
    - Редовно обучение (стойност 'Р')
    - Задочно обучение (стойност 'З')

#### **Форма за добавяне на курс**

- Наименование

#### **Форма за добавяне на оценка**

- Студент
  - SELECT с ID и име на студент от таблицата със студенти
- Предмет
  - SELECT с ID и наименование на предмет от номенклатурата с предмети
- Хорариум реално посетени упражнения
- Хорариум реално посетени лекции
- Оценка

За всеки от 5-те модела (специалности, предмети, студенти, курсове, оценки) трябва да има реализиран изглед за преглеждане на записите (GRID/таблица). Спецификите, касаещи изгледа на гридовете няма да бъдат зададени предварително – всеки може да реализира интерфейса на таблиците по собствено усмотрение (дали да има странициране и сортиране; дали за редакция, добавяне и изтриване на редове ще се ползват хипер-връзки или бутони; дали ще има филтър за ограничаване на показваните редове съгласно определени признаци ).

### ***Контрол на достъпа***

Да се реализира логин-форма, чрез която потребителите ще се идентифицират (логват) в системата.

Докато потребителят не е логнат в системата, той няма да има достъп до CRUD интерфейсите (гридове и форми) за управление на номенклатурите. В този случай приложението трябва да разрешава достъп само до главната таблица с оценките.

След успешна идентификация на потребителя той ще получи достъп и до CRUD интерфейсите посредством опростено меню със следните елементи:

- Курсове
- Специалности
- Дисциплини (предмети)
- Студенти
- Оценки
- Изход

### ***Дизайн на приложението***

Пропоръчително е реализацията на приложението да е разделена в няколко обособени части:

- В най-добрия случай да се ползва MVC или подобна архитектура;
- Алтернативно, поне да бъдат отделени логиката на приложението от презентацията:
  - Изходният код, който визуализира форма и кодът, който обработва изпратените от същата форма данни да не са в един и същ файл;
  - Изходният код за обработка на изпратени от форма данни и кодът, който визуализира грид, свързан със същия модел да не са в един и същ файл;
  - Изходният код, който изпълнява някаква бизнес-логика и кодът, който визуализира данни да не са в един и същ файл

**Бонус условия:**

- Да се добави възможност за експортиране на резултата във формати: PDF, XLS (MS Excel) и CSV (comma-separated values)
- Разработване на CRUD интерфейс (плюс GRID) за управление на потребители
- При изтриване на обект от grid-а да се добави потвърждаващ въпрос във вид „Сигурни ли сте, че желаете да изтриете избора (описание на обекта: студент, курс и т.н.)?“ на принципа на confirm функцията в JavaScript