За всички задачата е една:

**Създайте php приложение, организирано по модела MVC. Приложението да използва SEF URI, въвеждани на адресния ред, свързани с изпълнение на съответните операции с данните от базата. Използвайте URI, при четенето на които да е ясна изпълняваната операция.**

Накратко за алгоритъма на решение на задачата.

Основните файлове в приложението според шаблона за проектиране MVC са моделите, изгледите и контролерите. За проектните примери следва да напишете един модел, един контролер и няколко изгледа.

**Модел** – клас, съдържащ методи за работа с обектите от базата данни (методите може да са статични, както в разгледания на лекцията пример, но това не е задължително, зависи от конкретната задача).

**Контролер** – клас, който осъществява връзката между модела и изгледите. Съдържа методи-действия (имената им препоръчително започват с action), които извикват методите на модела и предават резултатите на съответните изгледи.

**Изгледи** – php файлове за визуализация на резултатите. При желание може да намерите готови красиви стилове и да ги адаптирате, в предлагания пример не съм се занимавала с това.

Допълнителни пояснения към задачата

1. Създаване на конфигурационния файла .htaccess в коренната папка на приложението

Както бе описано по-горе, създаваното приложение ще използва SEF адреси. За тях може да прочетете примерно тук:

<https://www.siteground.com/kb/what_are_search_engine_friendly_urls_sef/>

С цел опростяване SEF адресите ще се въвеждат от адресния ред. За примера, който Ви изпращам,са предвидени следните адреси (те се задават в routes.php, както ще стане думa по-нататък)

<http://localhost/yourName/users>

<http://localhost/yourName/users/12>

<http://localhost/yourName/users/add>

Вие следва да сами да си подберете адреси, съответващи на Вашата задача.

# За да не се получава съобщение, че страницата не е намерена, когато въведем SEF URI([Uniform Resource Identifier](https://en.wikipedia.org/wiki/Uniform_resource_identifier)), в конфигурационния файл .htaccess следва да се запише:

RewriteEngine on

RewriteRule ^(.\*)$ index.php

По този начин независимо от въведения URI (регулярният израз ^(.\*)$ описва всички възможни идентификатори), **се осъществява пренасочване към главния контролер index.php**

1. Описание на начина, по който ще се обработват въвежданите SEF URI, т.е. задаване на т.н. маршрути (routes)

Toва се прави във файла routes.php, който логично се разполага в папката config

Следвайки обектно-ориентирания подход на шаблона MVC, за изпълняване на изискваната с даден URI операция, се извиква метод-действие на съответния контролер. **Кой метод на кой контролер се указва във файла** **routes.php**. При това се препоръчват следните правила за имената : името на класа на контролера завършва с думата Controller, a имената на методите на контролера започват с думата action. При указание на маршрутите в routes.php тези стандартни думи се пропускат. **routes.php** връща асоциативен масив с ключове обработваните URI и значения - начинът на обработка. Значението задължително включва името на контролера без ключовата дума *Controller* и името на метода без ключовата дума *аction*. Така, ако извикваният метод не изисква параметри, на всеки URI-ключ съответства значение, включващо 2 елемента - ***<контролер>/<метод>,*** например routes.php може да има вида (обърнете внимание, че **URI се изброяват от по-специфичните към общите**):

return array(

" persons /add"=> " persons /add", *// метод actionAdd на класа PersonsController*

" persons /delete"=> " persons /delete ", *// метод actionDelete на класа PersonsController*

" persons "=> " persons /list ", *// метод actionList на класа PersonsController*

);

Когато е необходимо да подадем параметри на съответния action метод, в общия случай това съответствие не е възможно да се запише директно. Например, нека искаме да изведем списък на лицата, родени в определен град в определена година – в този случай на метода за извеждането с име примерно actionView трябва да се подадат 2 параметъра –града и годината. Директно това би изглеждало примерно по следния начин:

return array(

….

" persons /1980/Бургас"=> " persons /view/Бургас/1980",

" persons /1999/Варна"=> " persons /view/Варна/1999",

….

);

Ясно е, че по този начин не могат да се запишат всички варианти, броят им в общия случай може да бъде много голям. За записване на съответствието по обобщен начин ключът се задава чрез регулярен израз, а в маршрута задаваме референции към съответствията във вида **$n**, където **n** e номера на **поредната заградена в малки скоби част от шаблона.**

return array(

….

" persons/**(**[1-2][0-9]{3}**)**/**(**[\w|\W]{3,}**)**"=> " persons/view/$2/$1",

….

);

Замяната на референциите в маршрута с конкретните съвпадения става с помощта на функцията preg\_replace:

<http://php.net/manual/bg/function.preg-replace.php>

Пример за използване на preg\_replace()

<?php

$pattern="~persons/([1-2][0-9]{3})/([\w|\W]{3,})~";

$uri="persons/1998/Варна";

$general\_route="persons/view/$2/$1";

$route= preg\_replace($pattern, $general\_route, $uri);

echo $route; // persons/view/Варна/1998

?>

В случая за ограничителя на шаблона за търсене е използван символа **~** (php позволява), тъй като символа / се съдържа в шаблона.

Следва да се има предвид, че ако съвпадение не е намерено, preg\_replace () връща оригиналния низ, в случая persons/1998/Варна

1. В папката config записваме също така файла db\_params.php, връщащ асоциативен масив с параметрите, необходими за осъщeствяване на връзката с базата данни.
2. В папката components създаваме файла router.php, съдържащ класа **Router**. Методът run() на класа Router търси съвпадение на въведения URI със зададените URI във вече разгледания файл routes.php и ако такова съвпадение е намерено, се извличат елементите на маршрута (клас на контролера, метод на действие и параметри, ако има такива), създава се обект на контролера и се извиква необходимият му метод за действие.

Детайлите на работа на този клас може да си изясните чрез тестване на работата му и справки за работата на включените библиотечни функции. В крайния вариант включените спомагателните извеждания следва да се премахнат.

1. В папката components създаваме файла database.php, съдържащ класа Database с единствен статичен метод, връщащ референция към обект PDO за работа с базата данни. Повече за PDO можете да прочетете примерно тук:

<http://php.net/manual/en/book.pdo.php>

1. Както вече бе описано в точка 1, изпълнението на приложението започва винаги от файла index.php – главният контролер. В него изпълняваме някои основни настройки (показването на грешките е само за етапа на разработка), включваме необходимите файлове от папката components, след което създаваме обект Router и извикваме метода му run(). Последният от своя страна при намерено съвпадение на URI извиква необходимият action метод на необходимия контролер. Контролерът използва модела за получаване на резултатите и ги предава на изгледа, който ги визуализира. По този начин шаблона MVC позволява отговорностите за изпълнение на отделните части от задачата (обработката на данните и визуалното им представяне), да се разделят.

Предложеният вариант е примерна опростена реализация на MVC- актуалният в момента шаблон за проектиране на web приложения. За интересуващите се в нета има огромно количество информация.

За да тествате примера, който Ви изпращам, копирайте папкатаyourName в xampp/htdocs. Създайте таблица users и импортирайте таблицата data от файла. Обработваните адреси са дадени по-горе, копирам ги отново:

<http://localhost/yourName/users>

<http://localhost/yourName/users/1>

<http://localhost/yourName/users/add>

Следват индивидуалните варианти за база данни.

Моля, запишете проектите си в папка с **Вашето име и ФН** на htdocs. Съответно, на мястото на yourName в routes.php на примера, който Ви изпращам, да е името на Вашата папка.

Започнете със скелета на приложението – това, което правихме на лекцията - testRoutes.php, но с използване на отделни файлове - използвайте примера, който Ви изпращам, и го адаптирайте за Вашата задача. Ако Ви е трудно, първо го повторете в същия вид, но с зададените от Вас адреси и съответните им маршрути. testRoutes2par.php e същият пример, но с извличане на 2 параметъра от адреса. След като се убедите, че е извикан нужният метод на нужния клас на контролера ( класът –контролер при Вас е един), се заемете с детайлите – база данни и т.н.

Ето конкретните варианти на задачата. Зададена е **минимално изискваната функционалност**, всякаква допълнителна, за която се сетите, се приветства! Свободни сте също и да разширявате базата данни, за да е по-реално приложението, аз съм гледала да има по 4 полета в повечето задачи - да не Ви претоварвам ;)

Радослав

* База данни staff (персонал) с таблица employee– id, име, позиция, заплата

Функционалност :

* Извежда списък с всички служители
* Извежда списък на служителите на определена позиция - например, старши програмист и т.н. (параметър)
* Извежда 3-те (или зададен като параметър брой) най-високо платени служители (използва се съответната mySql заявка)
* Добавя служител

Благой

* База данни movies (филми) с таблица movie– id, име, жанр, рейтинг

Функционалност :

* Извежда списък с всички филми
* Извежда списък на филмите за зададен жанр (параметър)
* Добавя филм
* Извежда филмите oт даден жанр с рейтинг над зададения (параметри)

Николай

* База данни foods (храни) с таблица food\_product – id, име, производител (фирма), срок на годност

Функционалност :

* Извежда списък с всички храни
* Извежда списък на хранителните продукти за зададен производител (параметър)
* Добавя хранителен продукт
* Извежда хранителните продукти с изтичащ срок на годност следващите 2 месеца

Димо

* База данни countries (страни) с таблица country – id, име, столица, официален език, брой жители

Функционалност :

* Извежда списък с всички страни
* Извежда информацията за зададена страна
* Извежда списък на страните с брой жители над определено значение в милиони (параметър)
* Добавя страна

Зюлкарне

* База данни flats (апартаменти за продажба продажба) с таблица flat– id, площ, комплекс, цена

Функционалност :

* Извежда списък с всички апартаменти
* Извежда списък на апартаменти с цена под зададената
* Извежда списък на апартаменти в даден комплекс с площ по-голяма или равна на зададената
* Добавя апартамент

Гьокхан

* База данни albums (албуми) с таблица album – id, група, име, брой продажби

Функционалност :

* Извежда списък с всички албуми
* Извежда албум със зададен id (параметър)
* Извежда списък на албумите на определена група (параметър)
* Добавя албум

Стела

* База данни cultural\_program (културна програма) с таблица event– id, име на събитието, жанр (опера, театър, балет, концерт), дата на провеждане

Функционалност :

* Извежда списък с всички културни събития
* Извежда списък събитията за зададен жанр (параметър)
* Добавя събитие
* Извежда събитията за определен месец (параметър)

Кристияна

* База данни publications (публикации) с таблица article – id, списание, импакт-фактор (да или не), година на публикуване

Функционалност :

* Извежда списък с всички публикации
* Извежда списък на публикации след определена година (параметър)
* Извежда списък на публикации в списания с/без импакт фактор (параметър)
* Добавя статия

Виктория

* База данни posts (постове) с таблица post – id, автор, дата на създаване, брой преглеждания, съдържание

Функционалност :

* Извежда списък с всички постове (без съдържанието)
* Извежда списък на постовете на даден автор(параметър), без съдържанието
* Извежда пост със зададен id(параметър)
* Добавя пост

Хорие

* База данни users (потребители) с таблица user – id, ник, дата на регистрация, брой постове

Функционалност :

* Извежда списък с всички потребители
* Извежда списък на потребителите, регистрирани след определена година(параметър)
* Извежда 3 (или N на брой – N се подава като параметър) потребители с най-голям брой постове (използва се съответната mySql заявка)
* Добавя потребител

Пламена

* База данни pop groups (поп група) с таблица group– id, име, година на създаване, брой албуми

Функционалност :

* Извежда списък на всички групи
* Извежда списък на групите, създадени след определена година (параметър)
* Добавя група
* Извежда групите с брой албуми над определена стойност (параметър)

Гергана

* База данни students (студенти) с таблица student– id, име, специалност, среден успех

Функционалност :

* Извежда списък на всички студенти
* Извежда списък на студентите от определена специалност (параметър)
* Извежда списък на 3-те най-добри по успех студенти
* Добавя студент

Александра

* База данни taken\_exams (взети изпити) с таблица еxam– id, дисциплина, дата на изпита, оценка

Функционалност :

* Извежда списък на всички изпити
* Извежда списък на изпитите, взети с определена оценка (параметър)
* Извежда информацията за оценката по даден предмет (параметър)
* Добавя оценка

Айхан

* База данни soccer\_club с таблица player – id, име, години, позиция (вратар, защитник, полузащитник, нападател)

Функционалност :

* Извежда списък с всички играчи
* Извежда играч съз зададен id(параметър)
* Извежда списък с играчите в съответна позиция (напр., само нападатели и т.н.) – позицията да е параметър
* Добавя играч

Мирослав Стоянов

* База данни bookstore с таблица book – id, заглавие, автор, година на издаване

Функционалност :

* Извежда списък с всички книги
* Извежда списък с книгите, издадени през определена година
* Извежда списък с книгите на даден автор
* Добавя книга

Георги Парушев

* База данни university с таблица student – id, име, ФН, среден успех

Функционалност :

* Извежда списък с всички студенти
* Извежда студент със зададен id (параметър) всички студенти
* Добавя студент
* Извежда първите 3 (или зададен като параметър брой) по успех студенти

Тодор Тодоров

* База данни discography с таблица album – id, заглавие, година на издаване, брой продажби

Функционалност :

* Извежда списък с всички албуми
* Добавя албум
* Извежда албума с най-голям брой продажби
* Извежда албумите, издадени след определена година (параметър)

Нургин

* База данни uni с таблица group – id, курс, специалност, брой студенти

Функционалност :

* Извежда списък с всички групи
* Извежда групите от даден курс (параметър)
* Извежда групите за даден курс и специалност (параметри)
* Добавя група

Тахир

* База данни persons (лица) с таблица person– id, име, град, дата на раждане

Функционалност :

* Извежда списък с всички лица
* Извежда списък на лицата от даден град родени след определена година (параметри)
* Добавя лице
* Извежда лицето или лицата, които днес имат рожден ден

Виолета

* База данни it\_comp (ИТ състезание) с таблица team – id, име на отбора, учебно заведение, брой участници

Функционалност :

* Извежда списък с всички отбори
* Извежда отборите от дадено учебно заведение (параметър)
* Добавя отбор
* Извежда отбора с най-голям брой участници

Йоана-Мария

* База данни curriculum (учебен план) с таблица discipline (дисциплина) – id, име на учебната дисциплина, курс, хорариум в часове, начин на оценяване – изпит или текуща оценка

Функционалност :

* Извежда списък с всички дисциплини
* Извежда дисциплините с определен начин на оценяване(параметър)
* Добавя дисциплина
* Извежда дисциплините, изучавани в даден курс (параметър)

Ани

* База данни it\_uni (университети по информационни технологии) с таблица university – id, държава, град, такса на обучение

Функционалност :

* Извежда списък с всички университети
* Извежда IT университетите от зaдадена държава (параметър)
* Добавя университет
* Извежда 5-те университета (или зададен като параметър брой) с най-ниска такса на обучение

Валентин

* База данни beauty\_contest (конкурс за красота) с таблица participant – id, име, ръст, тегло, години

Функционалност :

* Извежда списък с всички участнички
* Извежда списък на петте най-млади участнички
* Добавя участничка
* Извежда списък на участничките с ръст над определено значение (параметър)

Яшар

* База данни phones (телефони) с таблица phone – id, фирма, модел, цена

Функционалност :

* Извежда списък с всички телефони
* Извежда списък на моделите на дадена фирма (параметър)
* Извежда 3-те (или зададен като параметър брой) най-евтини телефона
* Добавя телефон

Светослав

* База данни sport\_clubs с таблица club – id, име, вид спорт, колективен или индивидуален спорт (може да е булева стойност), град

Функционалност :

* Извежда списък с всички клубове
* Извежда списък с клубовете за определен спорт (парамеър)
* Извежда списък с клубовете за колективни спортове от даден град (параметри)
* Добавя клуб

Цветан

* База данни universities с таблица university – id, град, брой факултети, вид на финансиране - държавен, частен, смесен

Функционалност :

* Извежда списък с всички университети
* Извежда списък на университетите с над 5 факултета (или зададен като параметър брой на факултетите)
* Добавя университет
* Извежда списък на университетите с даден вид на финансиране (параметър)

Хейджан

* База данни charity\_company (благотворителна кампания) с таблица donor– id, име или фирма, град, дарена сума

Функционалност :

* Извежда списък с всички дарители
* Извежда списък на дарителите от даден град (параметър)
* Добавя дарител
* Извежда 5-те (или зададен като параметър брой) най-големи дарители

Ана

* База данни language\_school (езиково училище, гимназия) с таблица school– id, име, град, основен чужд език, годишна такса на обучение (0 за държавните)

Функционалност :

* Извежда списък с всички училища
* Извежда списък на държавните училища с за определен език (параметър)
* Извежда списък на езиковите училища в даден град с такса под зададената (параметри)
* Добавя училище

Ясмина

* База данни excursions(туристически екскурзии) с таблица excursion– id, страна, дата на тръгване, описание на маршрута

Функционалност :

* Извежда списък с всички екскурзии (без маршрута)
* Извежда пълната информация за екскурзия със зададен id(параметър)
* Извежда списък на екскурзиите до дадена страна след определен месец (параметри)
* Добавя екскурзия

Антон

* База данни sms с таблица message – id, автор, получател, дата на създаване, съдържание

Функционалност :

* Извежда списък с всички съобщения (без съдържанието)
* Извежда списък на съобщенията на автор до даден получател (параметри), без съдържанието
* Извежда пълната информация за съобщение със зададен id(параметър)
* Добавя съобщение

Шабан

* База данни courses\_pl (курсове по програмни езици) с таблица course– id, програмен език, дата на стартиране на курса, кратко описание

Функционалност :

* Извежда списък с всички курсове (без описанията)
* Извежда списък на курсовете до даден програмен език (параметър)
* Извежда пълната информация за курс със зададен id(параметър)
* Добавя курс

Сезгин

* База данни flights(полети) с таблица flight– id, дестинация, авиокомпания, час на излитане

Функционалност :

* Извежда списък с всички полети
* Извежда списък на полетите с час по-късен от текущия
* Извежда списък на полетите до зададена дестинация (параметър)
* Добавя полет

Исмаил

* База данни articles(статии) с таблица article– id, автор, университет, резюме

Функционалност :

* Извежда списък с всички статии (без резюметата)
* Извежда списък на статиите с автори от зададен университет (параметър)
* Извежда пълната информация за статия със зададен id(параметър)
* Добавя статия

Георги Добрев

* База данни grabo с таблица offer (офертa) – id, категория (здраве, храна и т.н.), град, цена

Функционалност :

* Извежда списък с всички оферти
* Извежда списък с всички оферти за даден град (параметър)
* Извежда списък с всички оферти от дадена категория за даден град (параметри)
* Добавя оферта

Смаил

* База данни vines (вина) с таблица vide– id, название, производител (винарска изба), цена

Функционалност :

* Извежда списък с всички вина
* Извежда списък на вината на зададен производител (параметър)
* Извежда списък на вината на зададен производител с цена под зададената (параметри)
* Добавя вино

Габриела

* База данни pets (домашни любимци) с таблица pet– id, вид(куче, котка), възраст, тегло

Функционалност :

* Извежда списък с всички домашни любимци
* Извежда списък на животните от определен вид
* Извежда списък на животните от даден вид с тегло над зададено ( параметри)
* Добавя животно

Петър

* База данни exams (изпити) с таблица exam– id, предмет, преподавател, дата на провеждане

Функционалност :

* Извежда списък с всички изпити
* Извежда списък на изпитите по определен предмет н
* Извежда списък на изпитите по на зададен преподавател предмет с дата след днешната
* Добавя изпит

Лиляна

* База данни concerts (концерти) с таблица concert– id, изпълнител, град, дата на провеждане

Функционалност :

* Извежда списък с всички концерти
* Извежда списък на концертите за зададен град (параметър)
* Добавя концерт
* Извежда концертите за следващите 10 дни

Мирослав

* База данни prog\_languages(програмни езици ) с таблица prog\_language(програмен език)– id, име, създател (лице или фирма), година на създаванe, рейтинг за текущата година (https://www.tiobe.com/tiobe-index/)

Функционалност :

* Извежда списък с всички програмни езици
* Извежда информацията първите 5 по рейтинг програмни езици
* Извежда списък програмните езици, създадени след определена година(параметър)
* Добавя програмен език

Ясен

* База данни computers (компютри) с таблица computer– id, марка, RAM, SSD( булево значение, има или не)

Функционалност :

* Извежда списък с всички компютри
* Извежда списък на компютрите с определен размер на RAM
* Извежда списък на компютрите от зададена марка с наличен SSD
* Добавя компютър

Тодор Стамболиев

* База данни capitals (столици) с таблица capital (столица) – id, име, държава, континент, брой жители

Функционалност :

* Извежда списък с всички столици
* Извежда информацията за зададена столица
* Извежда списък столиците за зададен континент с брой жители над определено число(параметри)
* Добавя столица

Мехмед

* База данни teamsports (колективни спортове ) с таблица sport – id, име на спорта, брой играчи, продължителност на играта, сезон(зимен или летен)

Функционалност :

* Извежда списък с всички спортове
* Извежда списък на спортовете за даден сезон (параметър)
* Извежда списък трите спорта с най-голяма продължителност на играта
* Добавя спорт

Ивайло

* База данни hotels (хотели ) с таблица hotel – id, име, град, категория (звезди)

Функционалност :

* Извежда списък с всички хотели
* Извежда списък на хотелите в определен град (параметър)
* Извежда списък на хотелите от дадена категория в определен град (параметри)
* Добавя хотел

Исмаил

* База данни с (дрехи ) с таблица garment(дреха)– id, вид (пола, рокля и т.н.), размер (S, M и т.н.) , цена

Функционалност :

* Извежда списък с всички дрехи
* Извежда списък на дрехите от даден вид (параметър)
* Извежда списък на дрехите от даден вид с цена под зададената (параметри)
* Добавя дреха

Мюмюн

* База данни painters (художници ) с таблица painter (художник)– id, име, страна, година (дата)на раждане, стил

Функционалност :

* Извежда списък с всички художници
* Извежда списък художниците, родени през даден век (параметър)
* Извежда списък на художниците от дадена страна в определен стил (параметри, например, импресионисти от Франция)
* Добавя художник

Даниел

* База данни rivers (реки ) с таблица river (река)– id, име, страна, континент, дължина

Функционалност :

* Извежда списък с всички реки
* Извежда списък реките в даден континент (параметър)
* Извежда списък на трите най-дълги реки
* Добавя река

Мария

* База данни mountains (планини ) с таблица mountain(планина)– id, име, страна, най-висок връх, височина на най-високия връх

Функционалност :

* Извежда списък с всички планини
* Извежда списък планините в дадена страна
* Извежда списък на трите най-дълги високи планини в дадена страна (параметри)
* Добавя планина

Радина

* База данни books (книги ) с таблица book – id, автор, жанр, година на издаване

Функционалност :

* Извежда списък с всички книги
* Извежда списък на книгите от определен жанр (параметър)
* Извежда списък на книгите, издадени след дадена година(параметър)
* Добавя книга

Айхан

* База данни sms с таблица message – id, автор, получател, дата на създаване, съдържание

Функционалност :

* Извежда списък с всички съобщения (без съдържанието)
* Извежда списък на съобщенията на автор до даден получател (параметри), без съдържанието
* Извежда пълната информация за съобщение със зададен id(параметър)
* Добавя съобщение

Рамадан

* База данни math\_courses (курсове по математика) с таблица course– id, брой часове дата на стартиране на курса, цена

Функционалност :

* Извежда списък с всички курсове
* Извежда списък на курсовете, започващи след определена дата (параметър)
* Извежда пълната информация за курс със зададен id(параметър)
* Добавя курс

Мартин

* База данни flights(полети) с таблица flight– id, дестинация, авиокомпания, час на излитане

Функционалност :

* Извежда списък с всички полети
* Извежда списък на полетите с час по-късен от текущия
* Извежда списък на полетите до зададена дестинация (параметър)
* Добавя полет

Невилин

* База данни articles(статии) с таблица article– id, автор, заглавие, раздел (спорт, кино и т.н.)

Функционалност :

* Извежда списък с всички статии
* Извежда списък на статиите от даден раздел (параметър)
* Извежда за статиите на даден автор(параметър)
* Добавя статия

Иван

* База данни grabo с таблица offer (офертa) – id, категория (здраве, храна и т.н.), град, цена

Функционалност :

* Извежда списък с всички оферти
* Извежда списък с всички оферти за даден град (параметър)
* Извежда списък с всички оферти от дадена категория за даден град (параметри)
* Добавя оферта