

# PHP 2013: Примерни Проекти

## Общи условия за всички проекти

1. Трябва да имат server-side част, която да използва PHP
2. Проектите трябва да използва база данни за съхранение на информацията (MySQL или друга). Позволено е използването на локален storage, но не за цялата информация.
3. Ползването на ORM и MVC библиотеки в backend-а са позволени, но трябва да сте наясно какво става там и за какво ги използвате, съответно да може да се променя сървърната част в движение по време на защитата
4. Не е позволено ползването на CMS системи за разработката на проектите
5. Потребителският интерфейс трябва да е базиран на AJAX, освен където това няма смисъл (а също и при login/logout)
6. От проектите се очаква да имат визия, подходяща за уебсайт. Не приемем default стилизация на интерфейсите елементи. Можете да ползвате всякакви CSS библиотеки или да пишете собствени стилове.
7. Ползването на JS библиотеки, както и писането на CoffeeScript е позволено.
8. Задължително е ползването на HTML5-enabled форми с валидация и съответния тип навсякъде, където може.
9. По време на изпита, всеки ще имплементира допълнителна функционалност по проекта, така че бъдете готови за това.
10. Ако двама души искат да правят един проект, трябва да се свържат с лекторите (joro.penchев -at- gmail.com с копие до g2h.horde -at- gmail.com) за усложняване на условията.
11. Предлагайте своите проекти, съобразени с горните условия на същите имейли.  
**Всеки проект трябва да бъде одобрен от лекторите, преди да го правите.**

Всеки студент може да си избере някой от долните проекти:

### 1. Check-in система с геолокация

Web-базирано приложение, което позволява потребителя да се логне със собствен акаунт, след което използвайки геолокацията, предоставена от брауъра, да се “check-in”-не в системата.

Системата пази всички места за дадения потребител, където той се е check-in-вал, заедно с информация за това кога се е случило даденото отбелязване (дата / час). Проектът трябва да включва подходяща визуализация на списъка от check-ins. Потребителят може да трие елементи от списъка от “отбелязвания”.

Системата трябва да поддържа също добавяне на различни мета-данни (key-value) към всеки check-in. Това ще позволи на системата да се модифицира за различни нужди. Какви мета-данни потребителите вписват в своите check-ins да може да се въвежда и редактира чрез специален панел достъпен само през администраторски акаунт.

Например, нашето web приложение се използва от потребители, за да се отбелязват в кои ресторанти са вечеряли. Системата е настроена за всяко чекиране да изисква допълнителна информация от потребителя, изразяваща се като полета с въпрос към чекирането: “Как е била храната” и “Как е било обслужването?”

Системата трябва да съдържа визуализация чрез карта (например Google Maps) с отбелязани маркери за това, къде е бил даденият потребител.

Да се добави бутон, който прави screenshot на картата и я експортира като снимка, която потребителят може да запази.

## **2. Социална мрежа**

Потребителите могат да се регистрират и да въвеждат някаква базова информация за себе си (име, имейл, години, снимка). Можеш да търсиш за потребители. Можеш да се сприятеляваш с потребители, като те трябва да потвърдят това.

Всеки потребител има стена, на която може да публикува постове. Постовете могат да имат коментари от приятели. Потребителят разполага и с албум, в който може да слага снимки, които също могат да бъдат коментирани. Трябва да се осигури подходящ интерфейс за разглеждане на снимките (може да се ползват известни библиотеки, например lightbox), Снимките трябва да могат да се “тагват” чрез интерфейс за целта, който да предлага списък от приятелите чрез html5 autocomplete.

Системата поддържа лични съобщения, достъпни през интерфейса на стената на потребителя - на своята страна да има бутон за четене на получени съобщения, а на чуждите стени да има бутон за писане на съобщение. Непрочетените съобщения да са оцветени в различен цвят. Всяко съобщение се характеризира с текст и и изпращач. Отварянето му трябва да го маркира като прочетено.

## **3. Хотелски резервации**

Обектите са хотели (име, адрес, звезди, набор стаи, снимка, град), градове, стаи (брой легла, гледка, собствена баня, цена, снимка). Въвеждат/променят/трият се от админ

панел, който е защитен по някакъв начин.

На сайта може да се разглеждат хотелите, сортирани по град, име, брой звезди. Може да се филтрира и търси по същите показатели. Търсенето по име да е по част от името. Всяка хотелска страница да има снимка + таблица с цени/стаи.

Сайтът да поддържа модул за резервации - да има календар, който показва резервационния статус на всички стаи от хотела за датите от календара с подходящи цветове и подходяща визуализация. Календарът да поддържа интерфейс за резервация (кликането по календара да вади ајах форма). Резервацията да изпраща на имейла на хотела писмо с данните за резервацията (име, час, дата, тип стая)

## 4. Киносалон

Проектът представлява система за разглеждане на кинопрограма и резервация на билети. Поддържа два вида потребители: обикновени и администратор. Администраторът може да въвежда/редактира/трие:

- кина (име, брой зали с брой места)
- филми (име, жанр, продължителност)
- потребители (име, имейл, юзърнейм, парола) - разбира се приложението поддържа и клиентска регистрация на потребители, а не само през админ панела
- програма (представлява асоциация кой филм в колко часа, в коя зала, в кое кино се прожектира.

Приложението не трябва да дава възможност за застъпване, като следи продължителността на филмите. Например, ако в зала 2 в кино А почне филм дълъг 3 часа в 17ч, то не мога в 18ч в същата зала да сложа друг филм, докато не изтрия първия от програмата).

Клиентската част дава възможност за:

- преглед на програмата за някое кино (във вид на календар)
- преглед на всички прожекции за някой филм на някоя дата (във вид на календар). Филмите се разглеждат категоризирани по жанр или по начална буква.
- възможност за резервиране на място (системата трябва да следи капацитета на залите, който се въвежда в админ панела). При резервация системата да изпраща имейл на потребителя с детайлите за резервацията. При пълна зала не трябва да се дава възможност за резервация.
- търсене на филм по: част от името, жанр, продължителност.

## 5. Е-парламент

Проектът представлява система за гласуване. Потребителите биват два вида: обикновени и администратор. Имат име, имейл, парола, потребителско име. Трябва да могат да се регистрират и влизат в сайта.

Всеки потребител може да:

- предлага тема за гласуване чрез интерфейс. Така предложените теми за гласуване се характеризират с предлагащ (потребител) и дата. Предложените теми се одобряват от админ.
- гласува по дадена тема в определения за това интервал. Гласуването е с два варианта: “за” и “против”
- да преглежда всички теми за гласуване, включително тези с изтекъл срок за гласуване и резултатите от тях
- да търси тема за гласуване по част от текста

Администраторът може да:

- редактира/въвежда/трие потребители
- редактира/въвежда/трие теми за гласуване
- одобрява теми за гласуване: те му се представят под формата на списък с интерфейс за одобрение или отхвърляне. При отхвърляне администраторът трябва да напише причина за отхвърляне, която се изпраща на имейла на потребителя.
- и всичко, което могат потребителите

## 6. Виртуална библиотека

Проектът представлява сайт за публикуване и четене на авторски материали от категории: разкази, стихове, статии. Материалите имат име, дата, автор, рейтинг, текст (html формат, а не чист текст), набор от тагове и набор от коментари.

Поддържа три вида потребители: администратор, автор, читател. Те се характеризират с име, имейл, потребителско име, парола, снимка.

Читателският интерфейс (след регистрация и вход) позволява:

- разглеждане на всички материали, филтрирани по категория и сортируеми по дата, по таг, по автор и по рейтинг
- търсене на материали по автор, по таг и по част от името
- четене на материал и интерфейс към него за оценяване (да се вижда и досегашната оценка) и за коментари. Въвеждането на коментар към материал изпраща имейл на автора на материала с този коментар.
- запазване на материал като “любим” и списък с връзки към любимите материали

Авторският интерфейс в допълнение към горното може да:

- публикува материал (бонус е интегрирането на Rich Edit контрол за въвеждане на текста, например TinyMCE). При публикуването се избира категория на публикацията, име, и се добавят тагове. Може да се въвеждат съществуващи тагове (да се предлагат чрез autocomplete) или да се въвеждат нови.
- редактира и трие собствен материал
- може да коментира собствен материал, но не може да оценява

В допълнение към горните неща, администраторът може да:

- редактира, въвежда, трие потребители
- редактира, въвежда, трие материали
- редактира, въвежда, трие коментари

## 7. Метеорологичен център

Проектът служи за въвеждане и преглеждане на метеорологични данни и прогнози.

Има три категории потребители: админ, служител, клиент (потребителско име, парола, име, имейл).

Администраторите:

- въвеждат/редактират/трият служители
- въвеждат/редактират/трият локации (които имат име и снимка)
- могат да командирова служители по локации (командировките имат начална дата, крайна дата, служител и локация). Могат да редактират и анулират командировки.
- могат да въвеждат/редактират/трият данни за всички локации (вж. по-долу)

Служителите:

- Могат да въвеждат данни във формат: локация, температура, влажност, време с точност до час. Могат и да редактират и трият тези данни.
- Служителите могат да въвеждат данни само за локации, в които са командировани и то само за до +7 дни от днешна дата, ако днешната дата е периода на командировката.

Потребителите (няма нужда от логин)

- могат да разглеждат списъка с локациите, да сортират и да търсят по име локация
- да преглеждат прогнозата за дадена локация

Прогнозите:

- Прогнозите трябва да се визуализират с графика - абсцисата е времето, а по ординатата са температура и влажност. И двете се представят с начупена линия между точки с координати (час, температура) и (час, влажност). За часовете, в

които няма данни, данните се интерполират, т.е. ако имам 2 градуса в 07:00 и 4 градуса в 09:00, то между тях има права линия на графиката (т.е. за 08:00ч се интерполира 3 градуса).

- Прогнозите-графики да се визуализират с CSS3 или с GD библиотеката на PHP, или с Javascript (Canvas)
- Прехвърлянето от дневна на седмична на месечна и т.н. да става с AJAX без зареждане на цялата страница.

Служителите трябва да могат да въвеждат данни. Данните са във формат: място, температура, влажност, време с точност до час. Служителите могат и да редактират и трият тези данни.

## 8. Уеб-паяк за продукти

Проектът представлява мини-интернет търсачка за най-добра цена на продукт. Поддържа потребители с име, потребителско име, имейл, парола. Всеки потребител има списък с продукти (продукт се характеризира само с име).

Потребителите могат да въвеждат произволни продукти в своя списък, както и да ги редактират и трият. Продуктите не са уникални.

Системата да позволява на потребителя да въвежда и сайтове, от които да започва търсенето. Сайтовете да са споделени и уникални в рамките на приложението и да се предлагат с autocomplete. Така всеки потребител има списък от продукти и списък от сайтове.

Интерфейсът да позволява потребителят да иницира търсенето. Да се имплементира веб-паяк, който започва подред да обхожда списъка от сайтове и търси в тях:

- срещания на продукта
- линкове към други сайтове (от вида <http://...>)

След като обходи цялото така получено дърво от срещнати сайтове и провери за срещанията на продукт, паякът да запамятава срещанията на продукта в HTML вид до два обхващащи родителя нагоре в DOM дървото и така да ги представя като резултат.

Например:

```
<div>
  <div id="item"><p> My product </p></div>
  <div id="test"> test </div>
</div>
```

При търсене за My product целият горен участък да се показва като резултат в панела с резултатите.

Резултатите да са сортирани по домейн на сайта на който са срещнати.

Обхождането трябва да спира след пет поредни обходени линка в дълбочина, за да не отнеме твърде много време.

Да се демонстрира приложението с приготвени тестови страници или с реално съдържащи се в интернет.

Като бонус: да има възможност приложението да търси с daemon script на сървъра и да изпраща периодично резултати на имейла на потребителя.