

## **Имплементација на системи со отворен код**

### **Прва лабораториска вежба**

Целта на оваа прва лабораториска вежба е да се запознаеме со околната во која ќе работиме и преку некои основни вежби да се запознаеме со основите на скриптниот јазик PHP.

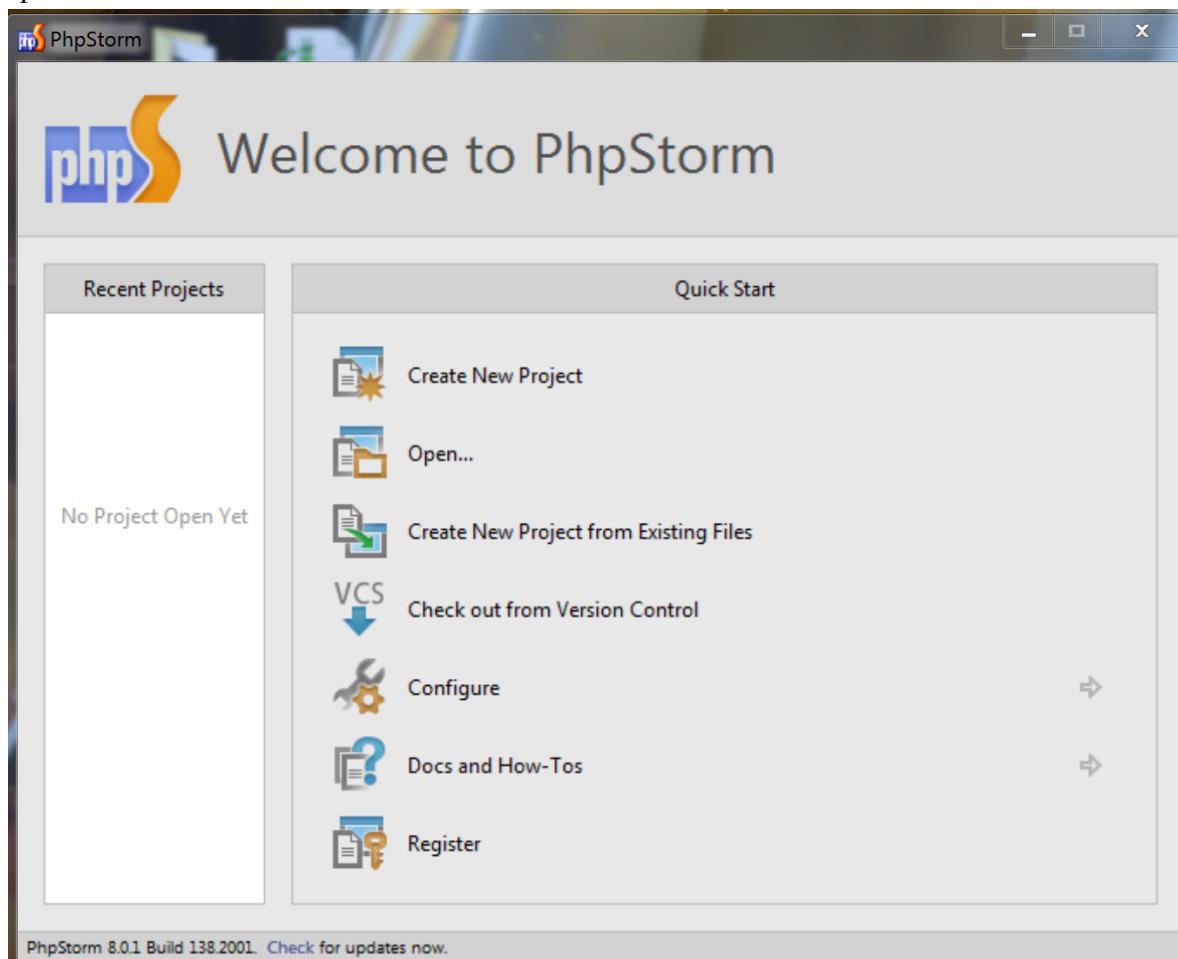
#### **Кратко упатство за работа:**

- ✓ Оперативен систем на кој што ќе работиме: Windows;
- ✓ Развојна околина во која ќе работиме: PhpStorm 8;

#### **Запознавање со PhpStorm:**

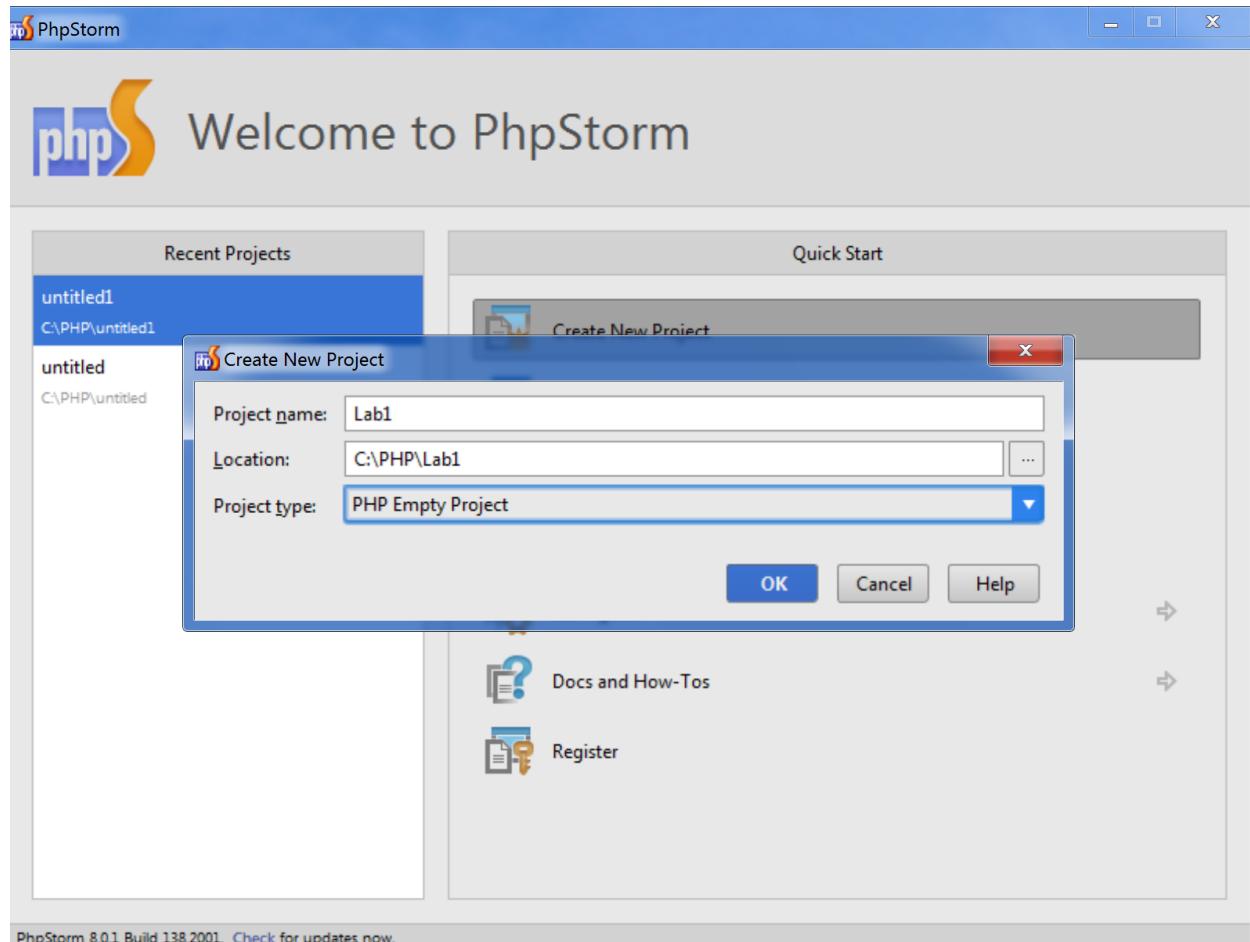
При отворање на апликацијата, се појавува прозорец за избор на опции за работа. Може да се избере да се направи нов проект, да се отвори веќе постоечки, да се отвори проект со готови фајлови и да се конфигурира. За почеток ќе избереме опција за креирање на нов

проект.



Откако ќе се одбере опцијата за отварање на нов проект, се појавува форма во којашто треба да се наведе името на проектот, директориумот во којшто се зачувува како и типот на проектот. Засега ќе работиме со тип на проект PHP Empty Project како на следната

слика:



По бирањето на тип на проектот, се појавува форма за бирање на интерпретер и верзија на PHP. Како верзија избирааме 5.6, а како интерпретер го избирааме локалниот фолдер во којшто PHP е инсталлиран на компјутерите во лабораториите. Тоа се наоѓа во фолдерот C:\PHP.



PHP Empty Project



### Create empty project for PHP

#### Development environment

PHP language level: 5.6 (variadic functions, argument unpacking, etc.) ▾

Interpreter: <no interpreter> ▾ ... ⚙ ⓘ

#### Include path



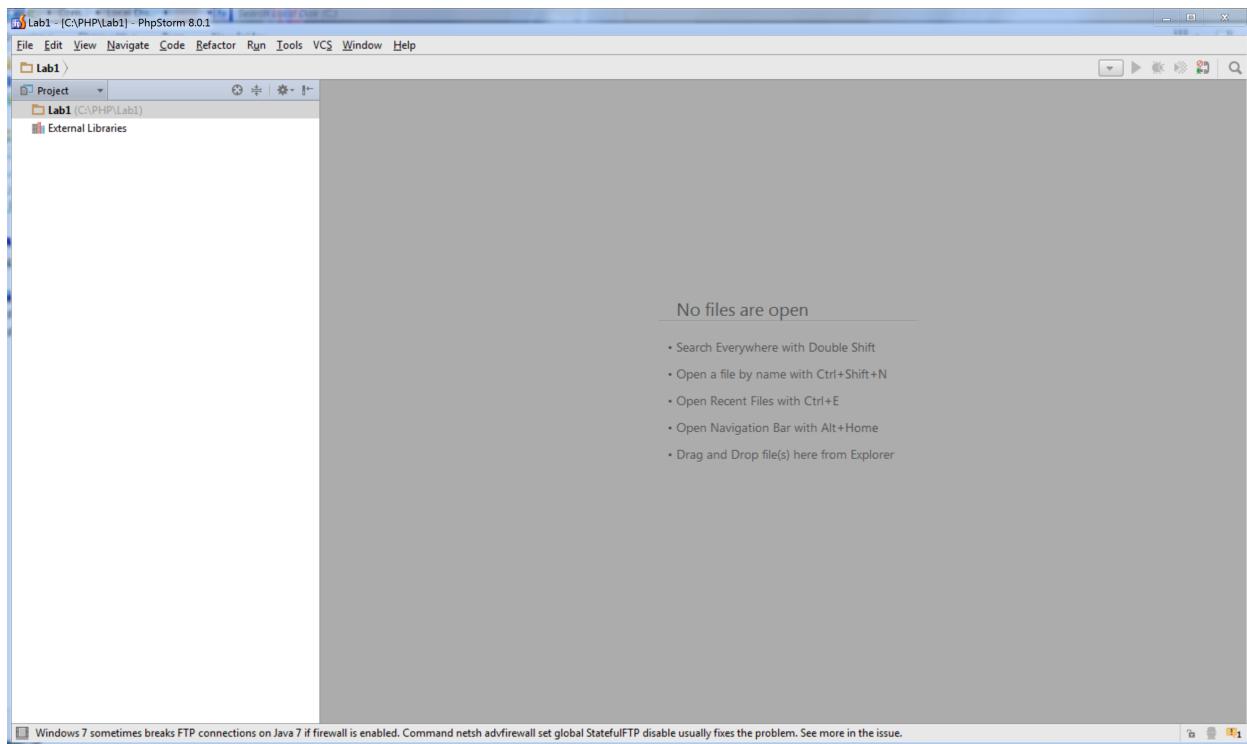
Provide include path

OK

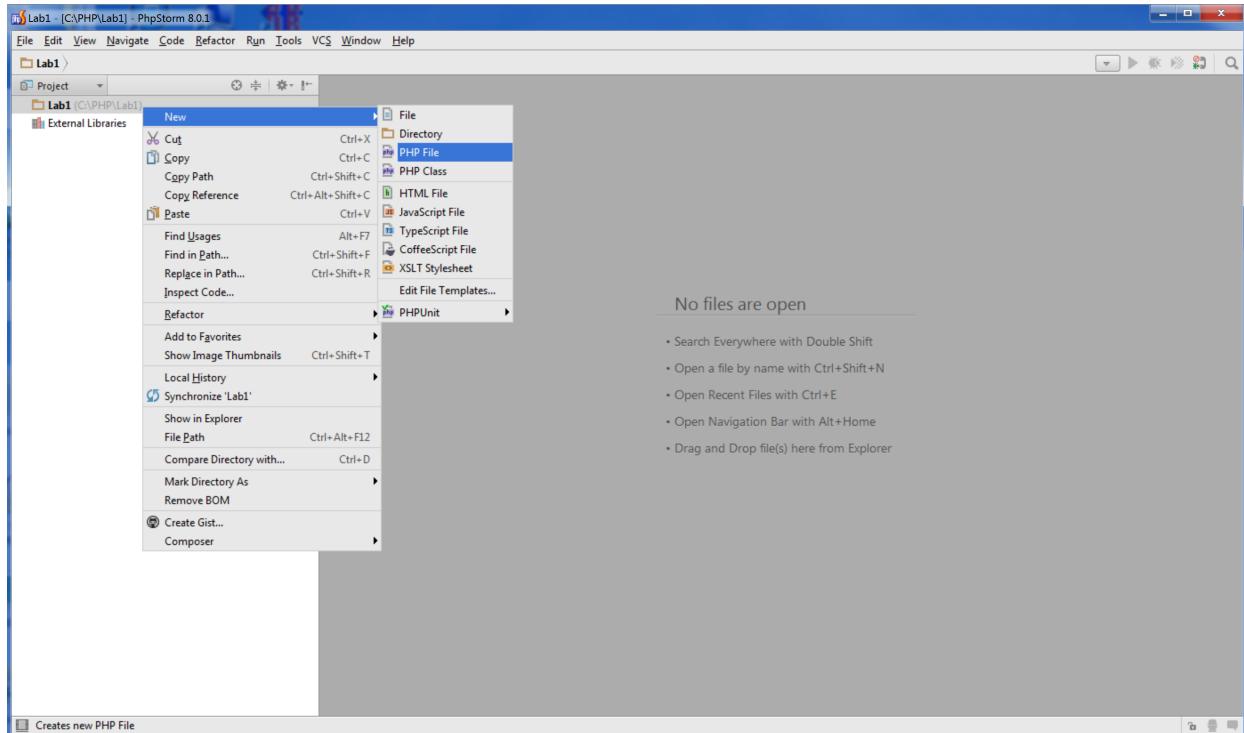
Cancel

Help

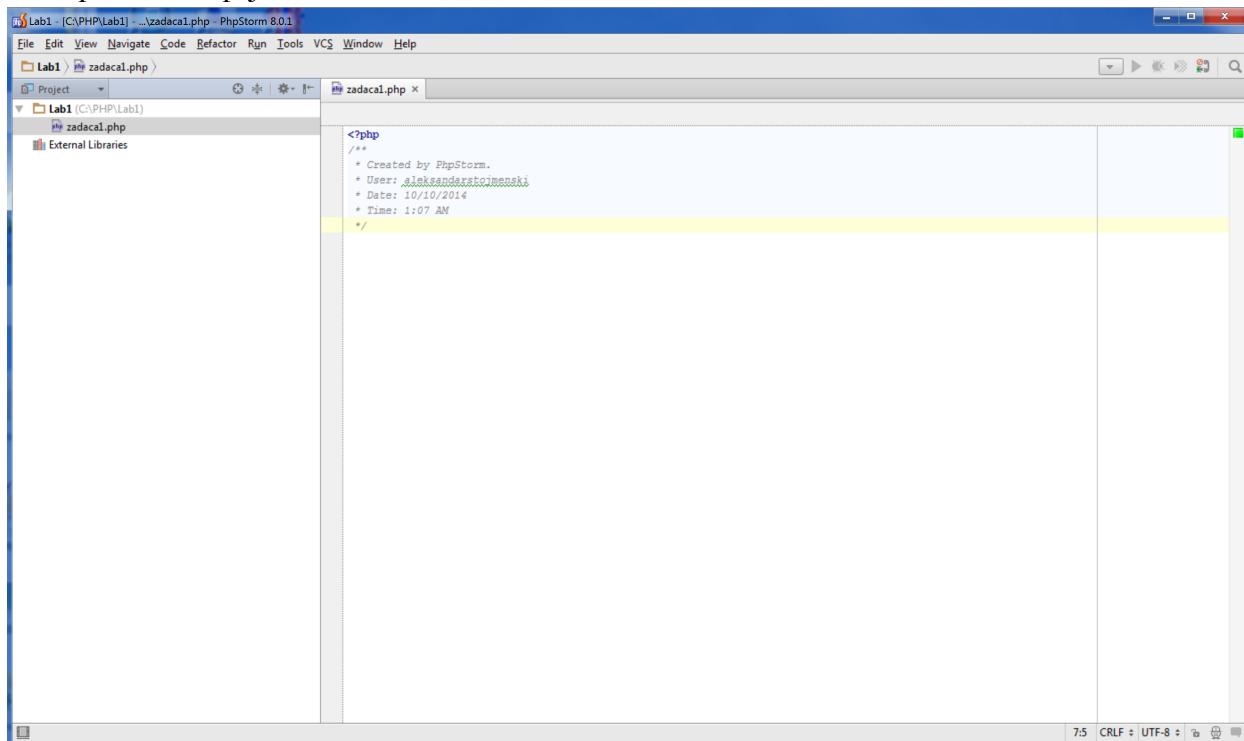
Кога ќе изберете интерпретер се наоѓате на главното мени од едиторот. Едиторот е поделен на два дела, а доколку се користи модот за debug се појавува и трет дел доле во којшто е конзолата. Во горниот дел се наоѓа главното мени, а под него се наоѓаат икони за пристап до стандарните функции: креирање на нов проект, снимање, преbarување, извршување итн..



Сега при избор на десен клик се оди во **New** и во **PHP File**. Истото може да се постигне и доколку се оди преку менито **File -> New -> PHP File**.



Се бира име на фајлот и се добива слика како следната.

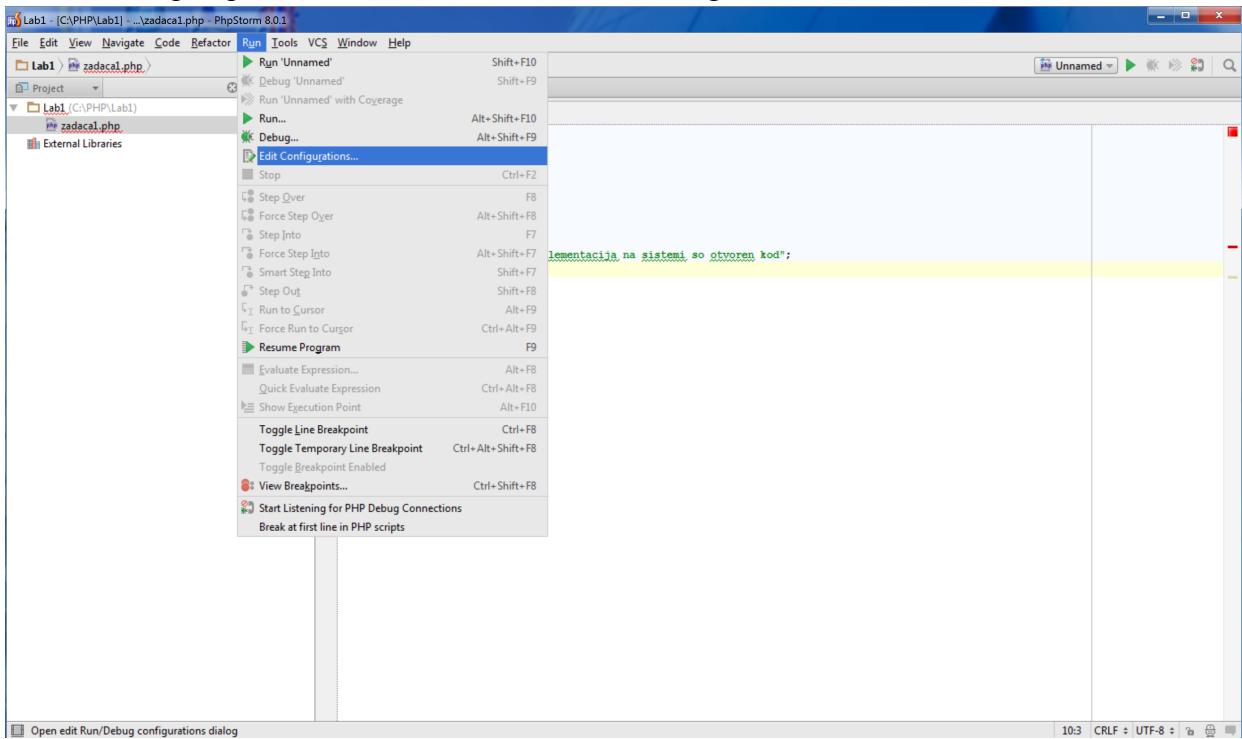


На оваа слика е прикажан едиторот и тука треба да се наоѓа вашиот код. Сега само ќе покажеме мал пример кој ќе ви помогне со сетирањето на проектот. Преку echo наредбата

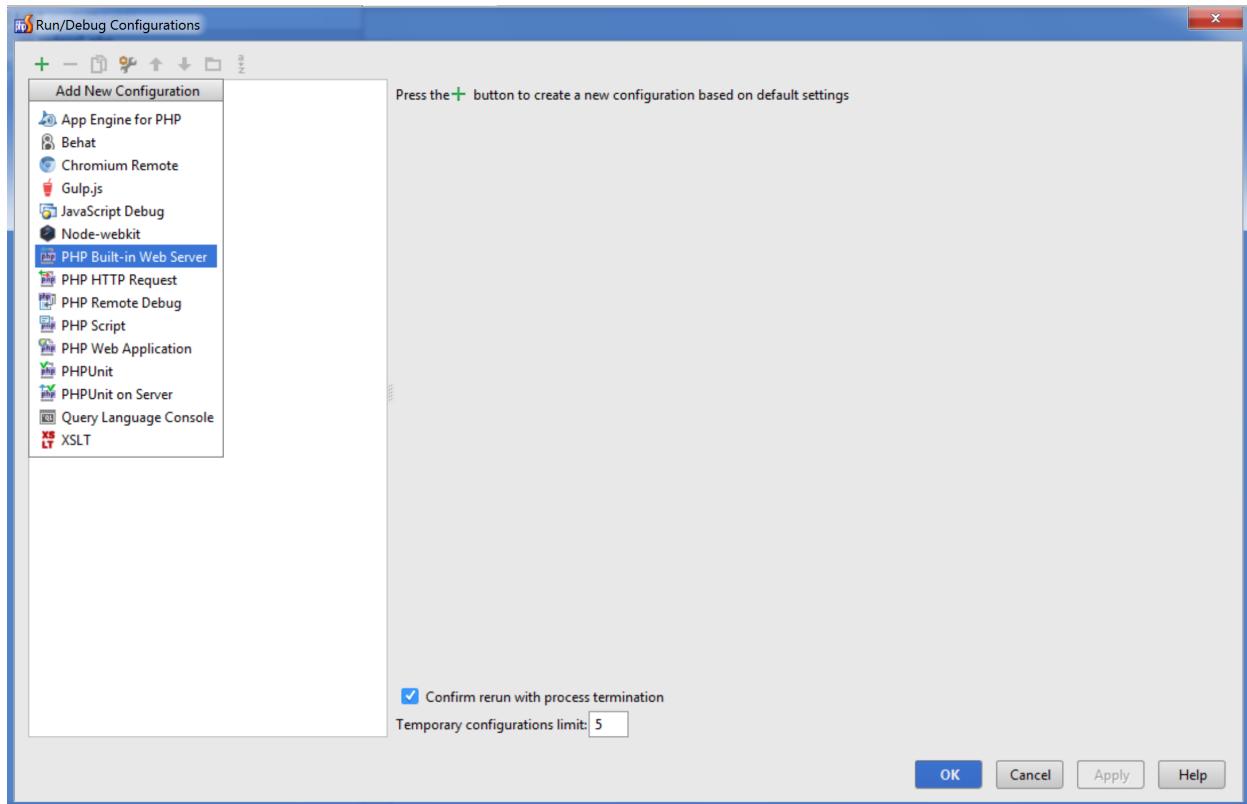
печатиме даден текст во PHP.

```
<?php
/*
 * Created by PhpStorm.
 * User: aleksandarstromenski
 * Date: 10/10/2014
 * Time: 1:07 AM
 */
echo "Dobrodojde na kursot Implementacija na sistemi so otvoren kod";
?>
```

За да се сетира проектот се оди во Run -> Edit Configurations.

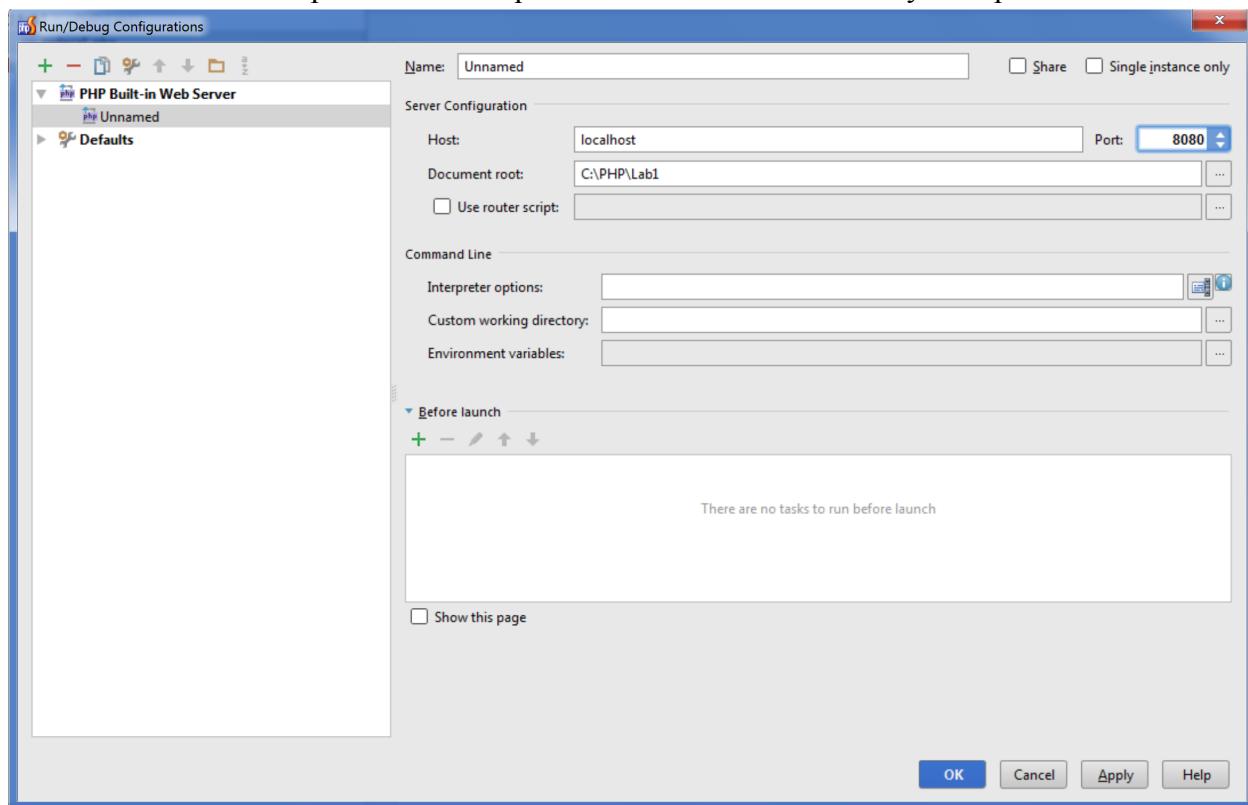


Во Edit Configurations по притискање на зеленото копче за додавање на нова конфигурација, излегува менито од лева страна. Во менито се одбира PHP Built – in Web Server како што е прикажано на сликата.



Откако ќе се направи нова конфигурација со притискање на копчето OK, излегува мени како следното. Во тоа мени кај делот за Port треба да се напише **8080**, а иницијално стои

80. Во Document root треба да се избере патеката кадешто ви е зачуван проектот.



Откако ќе заврши конфигурацијата, од десна страна постојат копчиња со знаците на познатите прелистувачи, при клик на едно од нив, скриптата се извршува на даден browser.

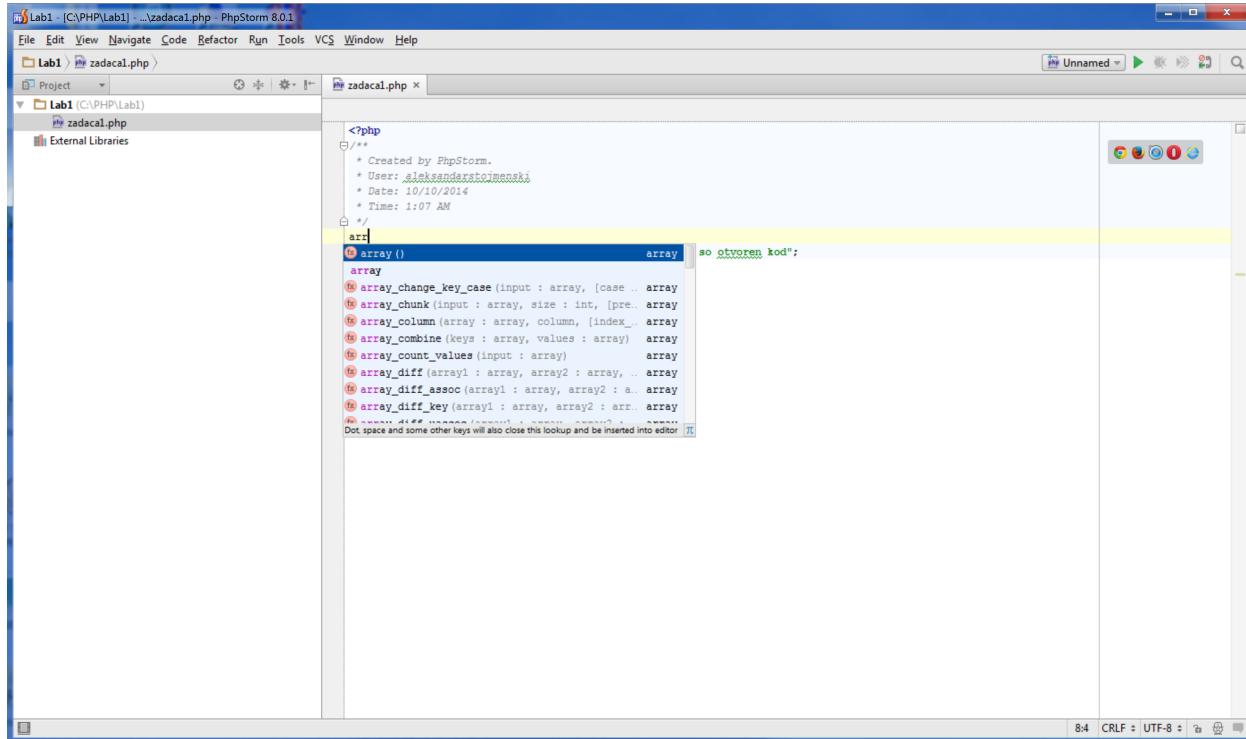
The screenshot shows the PhpStorm 8.0.1 IDE interface. The title bar reads "Lab1 - [C:\PHP\Lab1] - ...\\zadaca1.php - PhpStorm 8.0.1". The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The toolbar has icons for Unnamed, Run, Stop, Refresh, and Search. The Project tool window on the left shows a folder structure with "Lab1 (C:\PHP\Lab1)" expanded, containing "zadaca1.php". The Editor tab bar shows "zadaca1.php x". The code editor contains the following PHP code:

```
<?php
/*
 * Created by PhpStorm.
 * User: aleksandarstojimenski
 * Date: 10/10/2014
 * Time: 1:07 AM
 */
echo "Dobredojdovte na kursot Implementacija na sistemi so otvoren kod";
?>
```

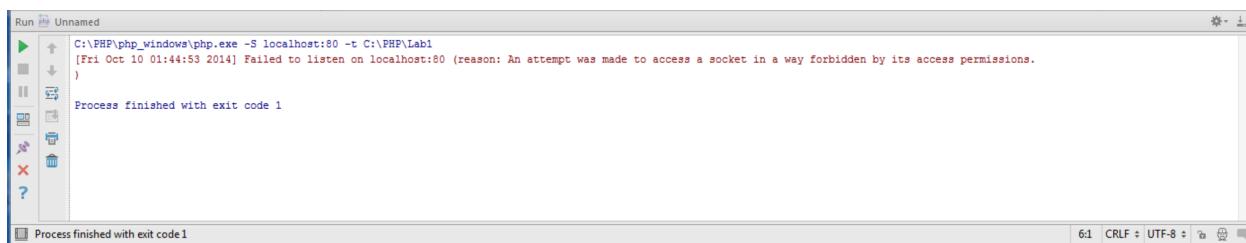
По одбирање на копчето Chrome од горната слика го добиваме текстот којшто го испечативме од скриптата во browser.



Во средишиот дел од едиторот може да се пишува и менува кодот на датотеките. Различните делови од кодот се претставени со различни бои што значително го олеснува кодирањето. Со притискање на *Ctrl+Space* се отвора прозорецот со предлози за функции:



Во долниот дел корисникот има можност да ги види сите проблеми и грешки кои што настанале при креирањето на документот како и нотификацијии од извршувањето на програмата.



За да отворите нов проект:

**File + New + PHP Project + името на проектот**

За да отворите нова php датотека:

**File + New + PHP File + името на датотеката (во Source Folder треба да го изберете името на проектот во кој сакате да креирате нова датотека).**

## **Воведни вежби:**

### **Вежба 1.1 - Вовед**

За да проверите дали добро е конфигурирана околината за работа со php, направете едноставна скрипта (welcome.php) во која ќе испишете текст за добредојде со помош на функциите **print** или **echo**.

### **Вежба 1.2 - Променливи во php**

Што треба да знаете за променливите во php:

1. Секоја променлива во php мора да го содржи знакот \$ на почеток;
2. На променливите може да им доделите вредности како броеви, стрингови и полиња. Вредноста на стринг променливите се става во единечни или двојни наводници (' или ").
3. Променливите не мора да бидат декларирани пред да им доделите вредност.
4. Имињата на променливите мора да содржат само алфа-нумерички знаци и долна црта (a-z, A-Z, 0-9, и \_) и не треба да содржат празни места. Нивното име може да почнува со буква или долна црта.

**Вежба:** Декларирајте 3 променливи и тоа, една со вредност број, една стринг и едно поле (array) и потоа испишете ги на екран. Вредностите нека бидат по ваш избор.

*Напомена: Користете ги функциите **echo** или **print** (за броевите и стринг променливите), и функцијата **print\_r(\$arr)** за полињата. Поле се дефинира со наредбата **array()**.*

## **Работа со текстови (Strings)**

Конкатенација на стрингови се прави со користење на знакот точка (.)

Некои од најчесто користените функции за работа со стрингови се:

- ✓ `strpos($str)` – се користи за пронаоѓање на некој карактер или текст во стрингот. Враќа FALSE доколку го нема бараниот карактер или позицијата на истиот доколку го има.
- ✓ `strlen($str)` – ја враќа должината на стрингот.
- ✓ `str_replace($search, $replace, $str, $num)` – заменува одреден дел од стрингот со друг стринг.
- ✓ `substr($string, $start, $length)` – функција која што враќа под-стринг од стринг.
- ✓ Комплетна листа на функции за работа со стрингови може да најдете на:  
<http://www.php.net/manual/en/ref.strings.php>

### Вежба 2.1 – Конкатенација на стринг

Дефинирајте две променливи, едната со вашето име, другата со вашето презиме и конкатенирајте ги во еден стринг и испечатете го на екран. Помеѓу нив ставете празно место.

### Вежба 2.2 – Функции

Испробајте ги функциите на одредена реченица по ваш избор: `strtoupper()`, `strtolower()`, `ucfirst()` и `ucwords()` и објаснете ја нивната функција.

Пример: `$string1 = strtoupper($string);`

...

### Вежба 2.3 – Поле -> Стринг

Креирајте стринг со помош на елементите на даденото поле:

```
$arr = array('programski', 'praktikum', 'laboratoriski', 'vezbi');
```

Напомена: Искористете ја функцијата: `string implode ( string $glue , array $pieces )`. Оваа функција се користи за конвертирање на поле во стринг.

Начин на употреба: `$string = implode($glue, $array);` каде што `$glue` го означува стрингот со кој сакате да ги раздвоите елементите од полето во реченицата.

## Полиња (Arrays):

Што треба да знаете за полињата:

- ✓ Се дефинираат со функцијата: `array()`. Како параметри се ставаат парови *ключ=>вредност*, одделени со запирка. Може и да додавате само вредности одделени со запирка. Во тој случај клучевите ќе бидат `integer` вредности.  
*Пример:* `$arr = array('kluc' => 'vrednost', 'kluc1'=>'vrednost1', ...);`
- ✓ Во основа, во `php` може да дефинирате **нумерички и асоцијативни** полиња. Нумеричките полиња за клучеви имаат `integer` вредности по редослед. Асоцијативните полиња може да содржат било кои `integer` или `string` вредности за клучеви.
- ✓ До вредностите на полето може да пристапите со:  
`$value = $arr[$key]`, или пак може да доделите некоја вредност со: `$arr[$key] = $value`.
- ✓ Исто така може да дефинирате и **повеќе-димензионали** полиња. Тоа се полиња кои што содржат едно или повеќе полиња.
- ✓ Најчесто употребувани функции за работа со `php` полиња:
  - `count ($arr)` – го дава бројот на елементи во полето. За оваа намена може да се користи и функцијата `sizeof($arr)`.
  - `array_push($arr, $value)` – се користи за додавање на елементи во полето.
  - `array_pop($arr)` – се користи за исфрлање на последниот елемент од полето
  - `array_key_exists($key, $arr)` – се користи за проверка дали одреден клуч постои во полето.
  - Постојат уште голем број на функции за работа со полиња во `php`, комплетната листа може да ја најдете на:  
<http://www.php.net/manual/en/ref.array.php>

### **Вежба 3.1 – Дефинирање на полиња**

Дефинирајте 3 полиња и тоа:

1. Нумеричко поле кое ќе ги содржи вредностите: 2, 5, 6, 10, 41, 24, 32, 9, 16, 19.
2. Асоцијативно поле кое ќе содржи 3 вредности и тоа: *клучот ќе биде: вашето име, вредноста ќе биде: вашето име*. Истото и за презиме и градот од кој доаѓате.
3. Повеќе-димензионално поле кое ќе содржи `integer` вредности по ваш избор.

### **Вежба 3.2 – Изминување на полиња**

Со користење на `for()` и `foreach()` циклусите, изминете го нумеричкото поле од претходната вежба. Притоа на екран испишете ги вредностите на полето за секој клуч:

```
for (init; condition; increment)
{
    code to be executed;
}
```

Пример:

```
for($i=0; $i<count($arr); $i++)
{
    ...
}
```

```
foreach (array as key=>value)
{
    code to be executed;
}
```

Пример:

```
foreach($arr as $key=>$value)
{
    ...
}
```

### Вежба 3.3 – Изминување на полинја и додавање во ново поле

Со помош на некое од горните изминувања, изминете го нумеричкото поле од првата вежба и во ново поле додајте ги сите вредности поголеми од 20. Додавањето извршете го со функцијата **array\_push()**.

*Напомена: Пред почетокот на циклусот, дефинирајте го полето како празно поле.*

### Вежба 3.3 – Најголем и најмали елемент

На нумеричкото поле кое го дефинирате во 3.1, најдете го најголемиот и најмалиот елемент со помош на готовите функции **max** и **min** којшто како влезни аргументи примаат поле. Потоа испечатете ги најголемиот и најмалиот член на полето.

### Вежба 3.4 - Должина на стринг

Во дадената реченица, пресметајте ја должината на секој од зборовите и вредностите ставете ги во асоцијативно поле, така што клуч ќе биде самиот збор, а вредноста ќе биде бројот на карактери во тој збор.

*PHP is a widely-used general-purpose scripting language that is especially suited for Web development*

*Напомена: Може да ја користите функцијата **explode()**, која што врака поле за даден стринг:*

array **explode** ( string *\$delimiter* , string *\$string* )

*\$delimiter – по што да делите, во случајов тоа е празно место (‘ ’).*