

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН**

Учреждение образования «Центральноазиатский технико-экономический колледж»

ОТЧЕТ

По учебной практике:

Применение систем управления веб-контентом

Выполнил:

студент гр. P2A

Fedorov Ilya

Проверила:

Жүсіпбекқызы К

Оценка \_\_\_\_\_\_\_\_\_\_\_\_

Алматы 2022

Source code and description: <https://github.com/lyarov22/php-projects>

0.

# Практическая работа №1

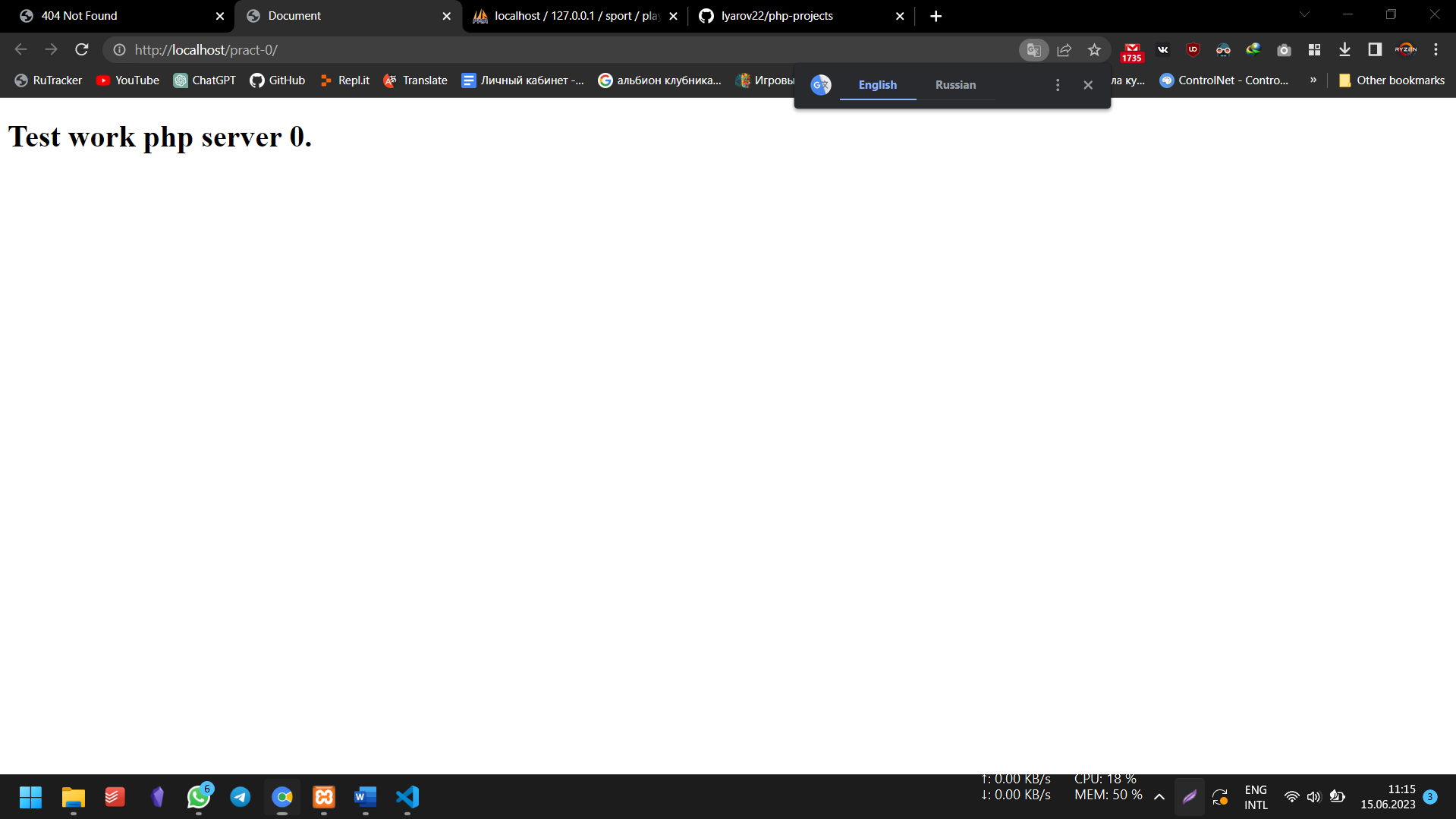
# Установка локального веб-сервера

**Цели:**

1. Ознакомиться с работой локального.веб-сервера

**Ход работы:**

1. Идём на сайт <https://www.apachefriends.org/ru/index.html>
2. Качаем XAMPP для Windows:
3. Запускаем скачанный файл от имени администратора:
4. Жмём «Next» до следующего окошка:
5. Оставляем галки на Apache, MySQL, PHP, phpMyAdmin:
6. Жмём «Next», выбираем путь установки. Я оставлю по умолчанию «C:\xampp».
7. Язык оставляем английский.
8. Убираем галку «Learn more about Bitnami for XAMPP».
9. Жмём «Next», ждём, пока установка завершится.
10. Жмём «Finish». Должна запуститься панель управления веб-сервером:
11. Жмём «Start» справа от Apache.
12. Переходим обратно в браузер, в адресную строку пишем «http://localhost».
13. Если открывается такое окошко, то всё установилось нормально:
14. Создаем в C:\xampp\htdocs новую папку «pract».
15. В папке создаём файл index.html.
16. Идём в браузер, пишем в адресную строку «localhost/pract». Если показалось следующее, то всё работает правильно:



1.

# Практическая работа №2

# Ознакомление с синтаксисом PHP

**Цели:**

1. Ознакомление с синтаксисом PHP.

**Ход работы:**

1. В ранее созданной папке pract удалить всё содержимое.
2. Создать в папке файл index.php.
3. В нём будет всё тот же HTML-код:

**Теория:**

PHP скрипт начинается со слова - **<?php** и заканчивается на **?>**. Все, что между **<?php** и **?>** это PHP код.

Файлы, в котором записан PHP код нужно сохранять под расширением **.php**

Когда нужно отобразить текст на веб-странице, то оператор echo является наиболее употребляемым оператором в PHP. Как его иcпользовать - после слова echo нужно поместить строку текста в кавычки:

<?php **echo** 'Привет от PHP'; ?>

Отображение в браузере: Привет от PHP

Для отображения текста можно использовать как двойные кавычки, так и одинарные. Для чисел кавычки можно не использовать:

<?php **echo** 2016; ?>

Оператор echo также может участвовать в форматировании веб-страницы:

<?php **echo** 'Петров Иван<br>Родился<br>...'; ?>

Отображение в браузере:

Петров Иван

Родился

...

# Функция var\_dump в PHP

Команда echo не всегда дает представление о том, что мы выводим. Для отладки ваших программ лучше использовать специальную функцию var\_dump. Она помимо вывода строки или числа еще будет указывать тип выводимых данных, а также их размер:

<?php

var\_dump('12345'); // выведет '12345' и еще некоторую полезную инфу

?>

**Правила именования переменных**

* Название переменной начинается с символа $.
* Вторым символом может быть буква или символ подчёркивания \_
* Дальше могут идти буквы, символы подчёркивания \_ и цифры.

<?php

$\_name // правильно

$15 // неправильно

$\_15 // правильно

$name\_15 // правильно

$name-15 // неправильно

1. В начале файла я добавлю несколько переменных, с которыми буду работать дальше. Обычно такие данные берутся из БД, но сейчас пока та
2. Дальше я вывожу эти данные на страницу. Здесь мы используем парочку из них. Ниже я из php вывожу целый тег:
3. Можно делать вывод более просто, не затрагивая остальной HTML-код:
4. Либо то же самое ещё проще, используя Echo-тег:
   1. В какие файлы обычно пишут php-код?

**Самостоятельная работа**

1. Выведите на экран одна за другой четыре строки: 'Всем ', 'привет ', 'от ', 'PHP!<br><br>', используя для этого только один оператор echo. После данной инструкции напишите два однострочных комментария: 'Для вывода слов на разных строках мы используем элемент "br", т.е разрыв строки' и 'Переменные выводимые оператором echo перечисляются через запятую'. Далее измените код первой части так, чтобы слова после вывода на экран находились на разных строках. В самом конце нашего кода напишите многострочный комментарий: 'После каждой инструкции мы будем ставить точку с запятой даже, если инструкция является последней в блоке php-кода'.
2. Выведите на экран друг за другом три имени: Robert, Stannis, Renly. В результате на экране должно отобразиться:

RobertStannisRenly

1. Создайте переменные с названиями title и content. Присвойте им какие-нибудь значения. Выведите значение переменной title внутри тега h1, а значение переменной content внутри тега div.
2. Выведения фразы «Мне … лет», где количество лет вычисляется на основании текущего года и года рождения.
3. Выведения фразы «Я учусь в ЦАТЭК … года», где количество лет вычисляется на основании текущего года и года поступления.

# Практическая работа №3

**Типы данных PHP**

**Извлечение символов из строки**

**Цели:**

1. **Типы данных PHP**
2. Извлечение символов из строки

**Теория:**

PHP является языком [динамической типизации](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F" \t "_blank), тип переменной определяется на основе её значения. Но все же нужно знать какие типы данных существуют в PHP. Ниже перечислены все типы, которые можно использовать в PHP:

* **Boolean**. Это логический тип, который содержит значение true или false.
* **Integer**. Содержит значения целого числа (Например: 4 или 10 или другое целое число).
* **String**. Содержит значение текста произвольной длины (Например: Олег, Киев, Австрия).
* **Float**. Вещественное число (Например: 1.2, 3.14, 8.5498777).
* **Object**. Объект.
* **Array**. Массив.
* **Resource**. Ресурс (Например: файл).
* **NULL**. Значение NULL.

Для предотвращения появления ошибок рекомендуется не смешивать разные типы данных.

Если вы хотите явно изменить тип данных переменной, то для этого нужно слева от имени переменной в круглых скобках указать нужный тип:

**Извлечение символов из строки**

### Типы данных - Задачи по строкам

Задача №1 - Создайте переменную text и присвойте ей значение "abcde".  
Обращаясь к отдельным символам строки нужно вывести на экран символы "a" "c" и "e"

Условия в последующих задачах просты и можно самостоятельно догадаться, что в них делается.

**Вопросы:**

Что будет выведено на экран?

**print\_r("php"[1])**

Укажите нужные индексы, чтобы программа вывела на экран имя Jim:

Начало формы

$name1 = "John"

$name2 = "Tom"

$name3 = "Emily"

print\_r($name1[] . $name3[] . $name2[]) // => Jim

Конец формы

**Самостоятельная работа**

1. Вам даны три переменные с фамилиями разных людей. Составьте и выведите на экран слово из символов в таком порядке:

1. Третий символ из первой строки
2. Второй символ из второй строки
3. Четвертый символ из третьей строки
4. Второй и третий символ из второй строки
5. $one = "Naharis";
6. $two = "Mormont";
7. $three = "Sand";

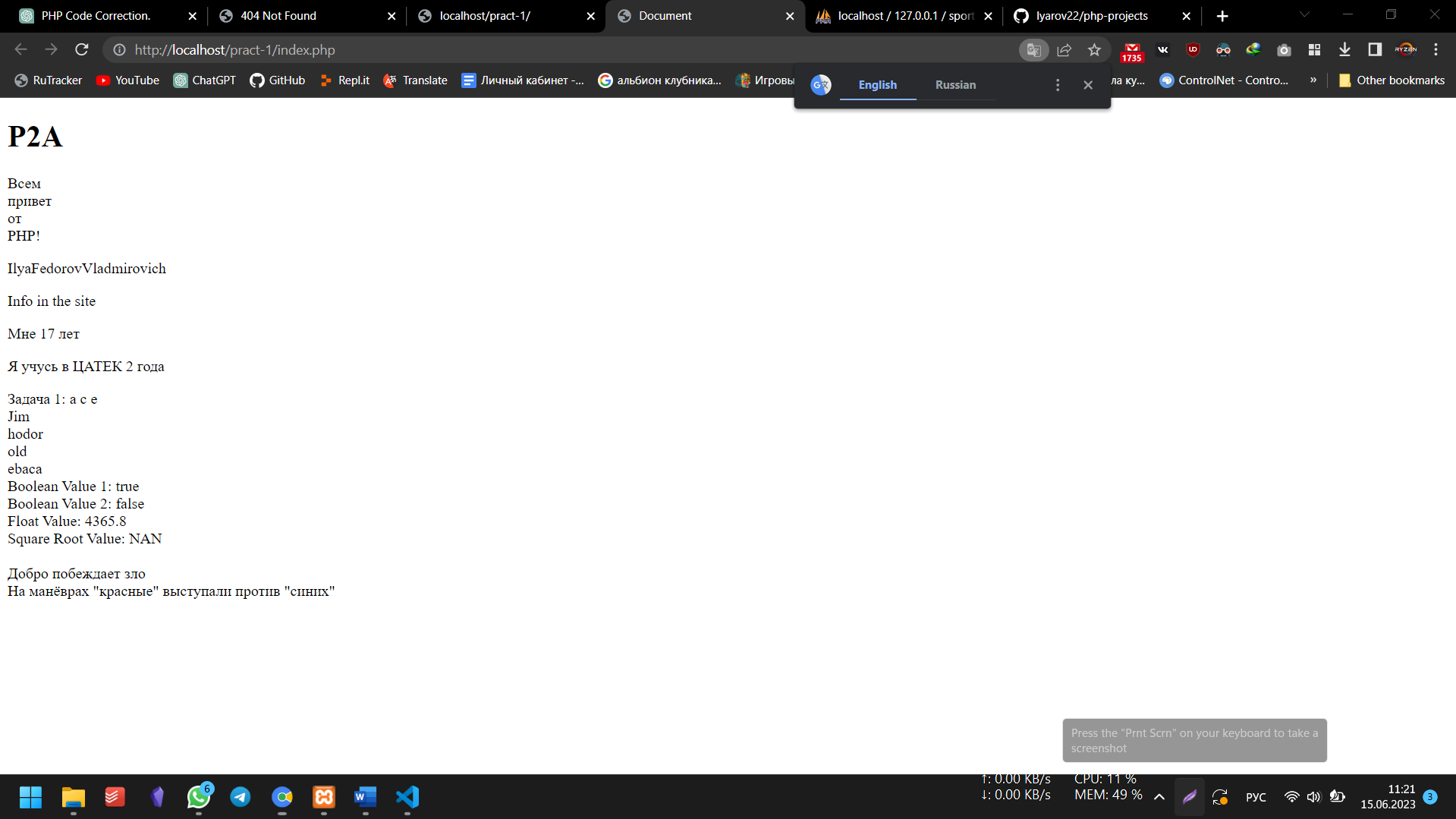
2. Дана строка 'world'. Обращаясь к отдельным символам этой строки выведите на экран символ 'o', символ 'l', символ 'd'.

3. Дана переменная со строкой 'abcde'. Обращаясь к отдельным символам этой строки запишите в новую переменную символы этой строки в обратном порядке, то есть 'edcba'.

4. Создайте переменные, присвоив им логические значения true и false, вещественное число 43.658+e2, но предварительно записав его в обычном виде, и корень квадратный из отрицательного числа, например, sqrt(-2). Выведите значения всех переменных на экран.

5. Создайте три переменные и присвойте им строки: 'Добро ', 'побеждает ', 'зло'. Создайте еще одну переменную и присвойте ей объединенную строку (для конкатенации строковых переменных используйте обычную точку). Выведите ее на экран.

6. Создайте переменную и присвойте ей строку "На манёврах "красные" выступали против "синих"", предварительно экранировав двойные кавычки внутри двойных во избежание ошибки. Выведите строку на экран. Как еще можно поступить для вывода строк, содержащих внутри себя двойные или одинарные кавычки, чтобы не использовать их экранирование?



2.

# Практическая работа №4

# Математические операции с числами в PHP

**Цели:**

Математические операции с числами в PHP

**Теория:**

На базовом уровне компьютеры оперируют только числами. Для сложения двух чисел в математике мы пишем, например, 3 + 4. В программировании — то же самое.

**<?php**

***// Снова не забываем точку с запятой в конце строчки***

**print\_r(3 + 4); *// => 7***

Кроме сложения доступны следующие операции:

* \* — умножение
* / — деление
* - — вычитание
* % — [остаток от деления](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81_%D0%BE%D1%81%D1%82%D0%B0%D1%82%D0%BA%D0%BE%D0%BC)
* \*\* — возведение в степень

Эти знаки операций называют операторами.

**<?php**

**print\_r(8 + 2);**

Здесь + — это оператор, а числа 8 и 2 — это операнды.

**Композиция операций**

Разберем пример композиции операций:

**<?php**

print\_r(2 **\*** 4 **\*** 5 **\*** 10); *// 2 \* 4 \* 5 \* 10 = 8 \* 5 \* 10 = 40 \* 10 => 400*

Операции можно соединять друг с другом. Это дает возможность вычислять более сложные составные выражения. Такое свойство операций называется **композицией**.

Чтобы представить, как происходят вычисления внутри интерпретатора, разберем пример: 2 \* 4 \* 5 \* 10.

1. Сначала вычисляется 2 \* 4 и получается выражение 8 \* 5 \* 10
2. Затем 8 \* 5. В итоге имеем 40 \* 10
3. В конце происходит последнее умножение, и получается результат 400

### Операции с числами

В данном уроке PHP рассмотрим работу с числами.

**Самостоятельная работа**

1. Создайте переменную $a. Запишите в нее сумму чисел 1, 2 и 3. С помощью команды echo выведите на экран содержимое переменной $a.
2. Создайте переменную $a со значением 10 и переменную $b со значением 2. Выведите на экран их сумму, разность, произведение и частное (результат деления).
3. Создайте переменную $c со значением 10 и переменную $d со значением 5. Просуммируйте их, а результат присвойте переменной $res. Выведите на экран значение переменной $res.
4. Создайте переменную $a со значением 1, переменную $b со значением 2 и переменную $c со значением 3. Выведите на экран их сумму.
5. Создайте переменную $a со значением 10 и переменную $b со значением 5. Отнимите от $a переменную $b и результат присвойте переменной $c. Затем создайте переменную $d, присвойте ей значение 7. Сложите переменные $c и $d, а результат запишите в переменную $res. Выведите на экран значение переменной $res.
6. Напишите скрипт, который:  
   1. Создаёт переменную $price и присваивает ей значение 1500.  
   2. Создаёт переменную с названием $shipping (стоимость доставки) и присваивает ей значение 200.  
   3. Создаёт переменную с названием $name и присваивает ей значение Автомат АК47.  
   4. Выводит 3 строки: название товара, стоимость товара и стоимость товара вместе с доставкой
7. В этом году зарплата Васи составляет 25 000р, а зарплата Пети - 30 000р. В следующем году зарплата Васи вырастет на 20%, а зарплата Пети - на 15%.

Напишите скрипт, который выведет на экран размеры зарплат обоих сотрудников в следующем году.

<?php

$salary\_vasya = 25000;

$salary\_petya = 30000;

?>

1. Компания выпустила 1000 акций. Джон выкупил 100 акций, Столерман - 150, а Иван - 235.

Напишите скрипт, который:  
1. Показывает, сколько процентов акций (от общего количества) имеет каждый из владельцев.  
2. Показывает общее количество проданных и непроданных акций.

1. Напишите программу, которая считает и последовательно выводит на экран значения следующих математических выражений:

* 3 в степени 5
* -8 разделить на -4
* остаток от деления 100 на 3

10. Напишите программу, которая берет исходное количество евро, записанное в переменную eurosCount, переводит евро в доллары и выводит на экран. Затем полученное значение переводит в рубли и выводит на новой строчке.

Пример вывода для 100 евро:

125

7500

Считаем, что:

* 1 евро = 1.25 долларов
* 1 доллар = 60 рублей

# Практическая работа №5

# Атрибуты тегов в PHP

# Практика на операции в PHP

**Цели:**

Атрибуты тегов в PHP

Практика на операции в PHP

**Теория:**

1. С помощью тега <img> выведите на экран какую-нибудь картинку.
2. С помощью тега <input> выведите на экран инпут с каким-нибудь текстом.
3. С помощью тега <textarea> выведите на экран многострочное поле ввода с каким-нибудь текстом.

**Практика на операции в PHP**

**Теория:**

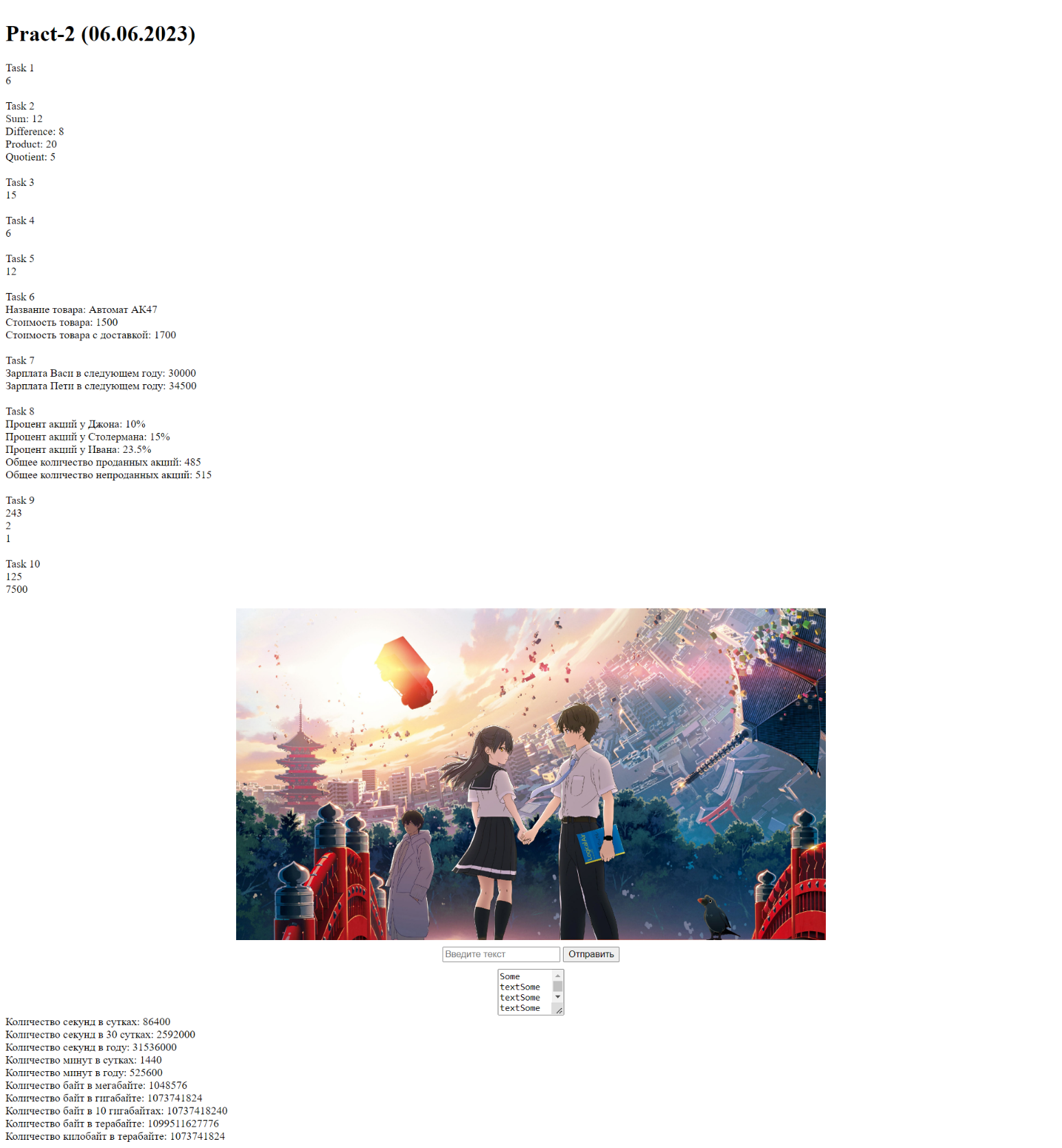
Давайте узнаем сколько секунд в часе. Как известно, в минуте 60 секунд, а в часе - 60 минут. Следовательно, чтобы найти количество секунд в часе, необходимо 60 умножить на 60:

<?php

echo 60 \* 60; // количество секунд в часе

?>

1. Найдите количество секунд в сутках.
2. Найдите количество секунд в 30 сутках.
3. Найдите количество секунд в году.
4. Найдите количество минут в сутках.
5. Найдите количество минут в году.
6. Найдите количество байт в мегабайте.
7. Найдите количество байт в гигабайте.
8. Найдите количество байт в 10 гигабайтах.
9. Найдите количество байт в терабайте.
10. Найдите количество килобайт в терабайте.



3.

# Практическая работа №7

**Отправка форм на сервер**

**Работа с полями ввода форм**

**Цели:**

Отправка форм на сервер

Работа с полями ввода форм

**Теория:**

Одним из основных способов передачи данных веб-сайту является обработка форм. Формы представляют специальные элементы разметки HTML, которые содержат в себе различные элементы ввода - текстовые поля, кнопки и т.д. И с помощью данных форм мы можем ввести некоторые данные и отправить их на сервер. А сервер уже обрабатывает эти данные.

Создание форм состоит из следующих аспектов:

Создание элемента <form><form> в разметке HTML

Добавление в этот элемент одно или несколько поле ввода

Установка метода передачи данных. Чаще всего используются методы GET или POST

Установка адреса, на который будут отправляться введенные данные

POST-запросы

Итак, создадим новую форму. Для этого определим новый файл form.php, в которое поместим следующее содержимое:

Атрибут action="user.php" элемента form указывает, что данные формы будет обрабатывать скрипт user.php, который будет находиться с файлом form.php в одной папке. А атрибут method="POST" указывает, что в качестве метода передачи данных будет применяться метод POST.

Теперь определим файл user.php, который будет иметь следующее содержание:

Для обработки запросов типа POST в PHP используется встроенная глобальная переменная $\_POST. Она представляет ассоциативный массив данных, переданных с помощью метода POST. Используя ключи, мы можем получить отправленные значения. Ключами в этом массиве являются значения атрибутов name у полей ввода формы.

Например, так как атриут name поля ввода возраста имеет значение age (<input type="number" name="age" />), то в массиве $\_POST значение этого поля будет представлять ключ "age": $\_POST["age"]

И поскольку возможны ситуации, когда поле ввода будет не установлено, то в этом случае желательно перед обработкой данных проверять их наличие с помощью функции isset(). И если переменная установлена, то функция isset() возвратит значение true.

Теперь мы можем обратиться к скрипту form.php и ввести в форму какие-нибудь данные:

И по нажатию кнопки введенные данные методом POST будут отправлены скрипту user.php:

Необязательно отправлять данные формы другому скрипту, можно данные формы обработать в том же файле формы. Для этого изменим файл form.php следующим образом:

Поскольку в данном случае мы отправляем данные этому же скрипту - то есть по тому же адресу, то у элемента форма можно не устанавливать атрибут action.

**Самостоятельная работа**

## 1. Простая форм

Создайте форму с одним полем ввода с запросом любимого города пользователя. Используйте для формы метод "post".

После того, как пользователь щелкнет кнопку отправки, выведите название города, введенное пользователем, сразу же под кнопкой "Отправить".

Подсказка: переменная, содержащая пользовательский ввод, является массивом. Переменная массива - $\_POST['name'], где 'name' - имя вашего поля ввода.

## 2. Интерактивная форма с утверждением If-Else

Возьмите за основу форму с одним полем ввода из предыдущего примера. С помощью условного оператора if-else отобразите, в случае заполнения поля ввода и нажатия кнопки "Отправить", название введенного города. Если поле не заполнено - отобразите форму.

Подсказка: вам понадобится способ узнать, отправлена ​​ли форма. Используйте функцию isset() для определения того, установлена ​​ли переменная и не является ли она нулевым значением.

## 3. Интерактивная форма с конструкцией If-Elseif-Else

В этом упражнении PHP вы будете использовать тот же формат, что и в предыдущем упражнении, запрашивая ввод в первой части скрипта и отвечая во второй с помощью оператора if-else PHP. В первом разделе создайте выпадающий список с днями недели и попросите пользователя выбрать день недели.

Для второй части скрипта, в зависимости от выбранного дня выведите соответствующую строку:  
"Понедельник - 1-й день недели"  
"Вторник это день второй"  
А за ним идет – среда"  
"За средой идет – четверг"  
"Пятница ударница"  
"День шестой – суббота"  
"Воскресенье - полный выходной"

Используя конструкцию else-elseif-else, настройте каждую строку для вывода в ответ на день выбора пользователем.

4. Создайте форму, состав и внешний вид полей которой показан на скриншоте. Скрипт, который будет обрабатывать данные формы на сервере, должен выводить либо сообщения с данными формы либо просить заполнить все поля формы (см. скриншот задачи). Для проверки заполнения полей имени и фамилии используйте условный оператор if и функцию mb\_strlen(). Не забывайте, что после скачивания файла содержимое папки нужно разместить в корневой папке test нашего учебного сайта на локальном сервере (иначе скрипт работать не будет).

5. Создайте в корневой папке test страницу с такой формой, добавив в нее кнопку «Сбросить». Что касается php-скрипта, то его нужно усложнить. Скрипт должен делать проверку заполнения полей имени и фамилии и в случае ошибки прерывать дальнейшее выполнение скрипта и выводить сообщение. Также скрипт должен проверять, был ли сделан выбор предпочтения и в зависимости от этого делать соответствующий вывод (см. скриншот).

6. Создайте форму, состав и внешний вид полей которой показан на скриншоте. Скрипт, который будет обрабатывать данные формы на сервере, должен проверять заполнение полей имени и фамилии и в случае ошибки выводить предупреждение и прерывать скрипт. Что касается аватарки, то скрипт должен либо вывести аватарку после имени и фамилии, либо сообщить о незагруженной аватарке (см. скриншот задачи). Для получения имени, размера и временной папки для хранения загруженных файлов используйте суперглобальный массив $\_FILES. Для перемещения загруженной аватарки в целевую папку используйте функцию move\_uploaded\_file() (в качестве целевой создайте папку avatars). Чтобы правильно указать путь к вставляемой на страницу аватарке, не забудьте начать его с http://localhost/test/. При желании можете добавить проверку загружаемой аватарки на соответствие типу изображения и т. д.



4.

***Тема***: Создание и обработка форм

***Цель работы***: Научиться обрабатывать данные из формы. Заполнять таблицы БД данными, полученными из формы.

***Задание:*** Создать для своей БД форму и обработчик формы для заполнения двух таблиц, в которых не требуются данные из других таблиц (последнее будет темой следующего занятия )

***Отчетность:*** Загрузить архив папки с сайтом /OSPanel/domains/<имя папки>), а также \*.sql скрипт вашей БД, сделанный из PHPMyAdmin.

# ЧАСТЬ 1. Основные элементы формы. Пример формы. Обработчик данных из формы для вывода полученных данных в окне браузера (input.php)

## **Основные элементы формы**

WWW — это система, ориентированная преимущественно на представление информации от пользователя. Однако часто возникает обратная задача — пользователь должен сообщить нечто серверу. Основным средством организации диалога с пользователем являются формы. Поэтому HTML содержит теги, позволяющие реализовать все основные информационные и управляющие элементы форм.

Когда пользователь заполняет форму и нажимает кнопку подтверждения, информация, введенная пользователем в форму, посылается для обработки. Обработка при работе в WWW обычно организуется HTTP-сервером, а в Web-приложениях она часто может выполняться локально.

### **Синтаксис**

Формы создаются с помощью тега <FORM>. В общем виде синтаксис определения формы выглядит так:

<FORM NAME="имя\_формы" METHOD="get|post" ACTION="URL\_для\_обработки" ONSUBMIT="некоторое\_действие">

...Элементы формы и другие элементы HTML...

</FORM>

Атрибуты тега FORM (METHOD, ACTION, ONSUBMIT) могут отсутствовать, или присутствовать частично (назначение этих атрибутов будет рассмотрено в лабораторных работах посвященных обработке форм на стороне клиента, и на стороне сервера)

В теле тега могут находиться любые фрагменты HTML-текстов.

Каждая форма определяет отдельную область действия для элементов внутри формы. Одна страница может содержать любое число форм, каждая из которых функционирует независимо. Если две формы содержат элементы с одинаковыми именами, то эти элементы будут доступны отдельно.  *Вложение форм* (одна в другую) *не допускается.*

### **Элементы INPUT**

Для организации ввода одной строки текста или одного слова используется тег <INPUT>. Этот тег создает в форме поле для ввода текста, что дает возможность запрашивать информацию у пользователя.

Атрибуты:

TYPE — определяет тип поля ввода. Он может быть одним из следующих:

TYPE="TEXT|PASSWORD|CHECKBOX|RADIO|HIDDEN|SUBMIT|RESET|IMAGE".

По умолчанию это простое поле ввода для одной строки текста

NAME — имя поля ввода, которое используется как уникальный идентификатор, по которому впоследствии можно получить данные, помещенные пользователем в это поле. Т.е. это имя переменной для значения, возвращаемого полем ввода.

MAXLENGTH — определяет максимальное количество символов, которое пользователи могут ввести в поле ввода. При превышении количества допустимых символов браузер не дает его ввести. По умолчанию значение MAXLENGTH равно бесконечности.

SIZE — определяет визуальный размер полей ввода для текста и пароля на экране (в символах, по умолчанию 20). Не путать с атрибутом MAXLENGTH. Если MAXLENGTH больше чем SIZE, то при заполнении окна ввода осуществляется горизонтальный скроллинг.

VALUE — присваивает полю значение по умолчанию. Используется для чтения значения, которое будет выбрано при использовании типа RADIO (для типа RADIO данный атрибут обязателен)

а) Элемент типа TEXT создает текстовое окно с одной строкой:

<INPUT  TYPE=TEXT … >

Если задано значение атрибута VALUE, то оно используется как значение поля по умолчанию.

б) Элемент типа PASSWORD (пароль)  представляет текстовое окно с одной строкой, в которой вводимые пользователем символы отображаются в виде звездочек (\*)

<INPUT TYPE=PASSWORD …>

Если задано значение атрибута VALUE, то оно используется как значение поля по умолчанию.

в) Элемент типа SUBMIT создает кнопку, нажатие которой воспринимается браузером как признак окончания заполнения формы:

<INPUT TYPE=SUBMIT VALUE>

Атрибут VALUE определяет текст, который будет написан на кнопке. Если атрибут не задан, то на кнопку выводится текст, принятый данным типом браузера по умолчанию (зависит от типа браузера и русифицирован ли он).

Информация из всех именованных полей формы (имеющих атрибут NAME) автоматически собирается вместе и пересылается URL, указанному в атрибуте ACTION.

Если задать в элементе SUBMIT атрибут NAME, то надпись на кнопке (VALUE) будет передана вместе с другими значениями формы.

в) Элемент типа BUTTON по умолчанию действует аналогично SUBMIT, но предоставляет дополнительную возможность управлять форматом кнопки (цвет фона, шрифт и т.п.) с помощью стилей.

в) Элемент типа IMAGE используется для создания графической кнопки, изображение которой определяется параметром SRC:

<INPUT TYPE=IMAGE SRC="button.gif" BORDER=0>

По общему действию — аналог SUBMIT. Однако в данном случае обработчику формы пересылаются координаты курсора мыши в момент нажатия кнопки относительно левого верхнего угла картинки. Любые значения атрибута VALUE игнорируются.

Описание картинки аналогично используемому в теге <IMAGE>.

б) Элемент типа RESET используется для создания кнопки, при нажатии которой все поля формы принимают значения, заданные для них по умолчанию:

<INPUT TYPE=RESET VALUE="Очистить!">

### **Элемент CHECKBOX**

Флажок используется для получения простых значений, типа "выбрано"–"не выбрано":

<INPUT TYPE=CHECKBOX >

Если нужно, чтобы по умолчанию флажок был установлен, то задается атрибут CHECKED:

<INPUT TYPE=CHECKBOX NAME='send' CHECKED>

Когда флажок установлен, обработчику по умолчанию передается значение 'on'. Однако это значение можно изменить, если определить значение атрибута VALUE.

### **Элемент RADIO**

Переключатель используется для выбора одной из предлагаемых альтернатив:

<INPUT TYPE=RADIO NAME="price" VALUE="100">

. . .

<INPUT TYPE=RADIO NAME="price" VALUE="10000">

Атрибут VALUE выбранной альтернативы определяет значение элемента, которое будет передано на обработку.

Одна из альтернатив может выводиться выбранной по умолчанию (отмеченной). Для этого в ее описание нужно добавить атрибут CHECKED.

### **Элемент SELECT**

Тег <SELECT> используется для создания элементов, типа раскрывающегося списка, когда пользователю предоставляется возможность выбрать один или несколько элементов из предлагаемого списка.

Элементы списка задаются тегами <OPTION>, поэтому общая структура конструкции имеет вид

<SELECT NAME="foo">

<OPTION> *текст*

. . .

<OPTION> *текст*

</SELECT>

Для тега SELECT атрибут NAME является обязательным. Атрибут SIZE задает количество одновременно отображаемых элементов (по умолчанию 1). Атрибут MULTIPLE (без значения) разрешает выбирать из списка более одного элемента.

У тега OPTION также есть свои атрибуты.

При задании атрибута SELECTED соответствующая опция будет отображаться как выбранная по умолчанию. Если в SELECT задан атрибут MULTIPLE, то выбранными по умолчанию могут быть несколько опций.

Если для опции задано значение атрибута VALUE, то оно будет передаваться на обработку, когда данная опция выбрана.

### **Элемент TEXTAREA**

Тег <TEXTAREA> используется для того, чтобы позволить пользователю вводить более одной строки произвольного текста:

<TEXTAREA NAME="Resume" ROWS=10 COLS=50></TEXTAREA>

Атрибуты, используемые внутри тега <TEXTAREA> определяют размеры вид и имя создаваемого текстового поля.

Закрывающий тег </TEXTAREA> обязателен и должен присутствовать даже тогда, когда поле ввода изначально пустое.

Атрибуты:

NAME — имя поля ввода,

ROWS — высота поля ввода в символах,

COLS — ширина поля ввода в символах.

WRAP — определяет режим перевода строк при заполнении строк. Возможными значениями являются:

OFF — запрет перевода строк;

PHYSICAL — разрешается перевод строк и при пересылке текста в него вставляются символы перевода строк;

VIRTUAL — внутри элемента строки переводятся, но текст посылается в виде одной длинной строки.

Если нужно, чтобы в поле ввода тега TEXTAREA по умолчанию выводился какой-либо текст, то его надо вставить его между тегами <TEXTAREA> и </TEXTAREA>. Например,

<TEXTAREA NAME="address" ROWS=10 COLS=50>

Введите сюда подробную информацию о проблеме,

возникшей при использовании нашего программного

продукта.

</TEXTAREA>

**Пример (** <https://metanit.com/web/php/3.4.php> **).**

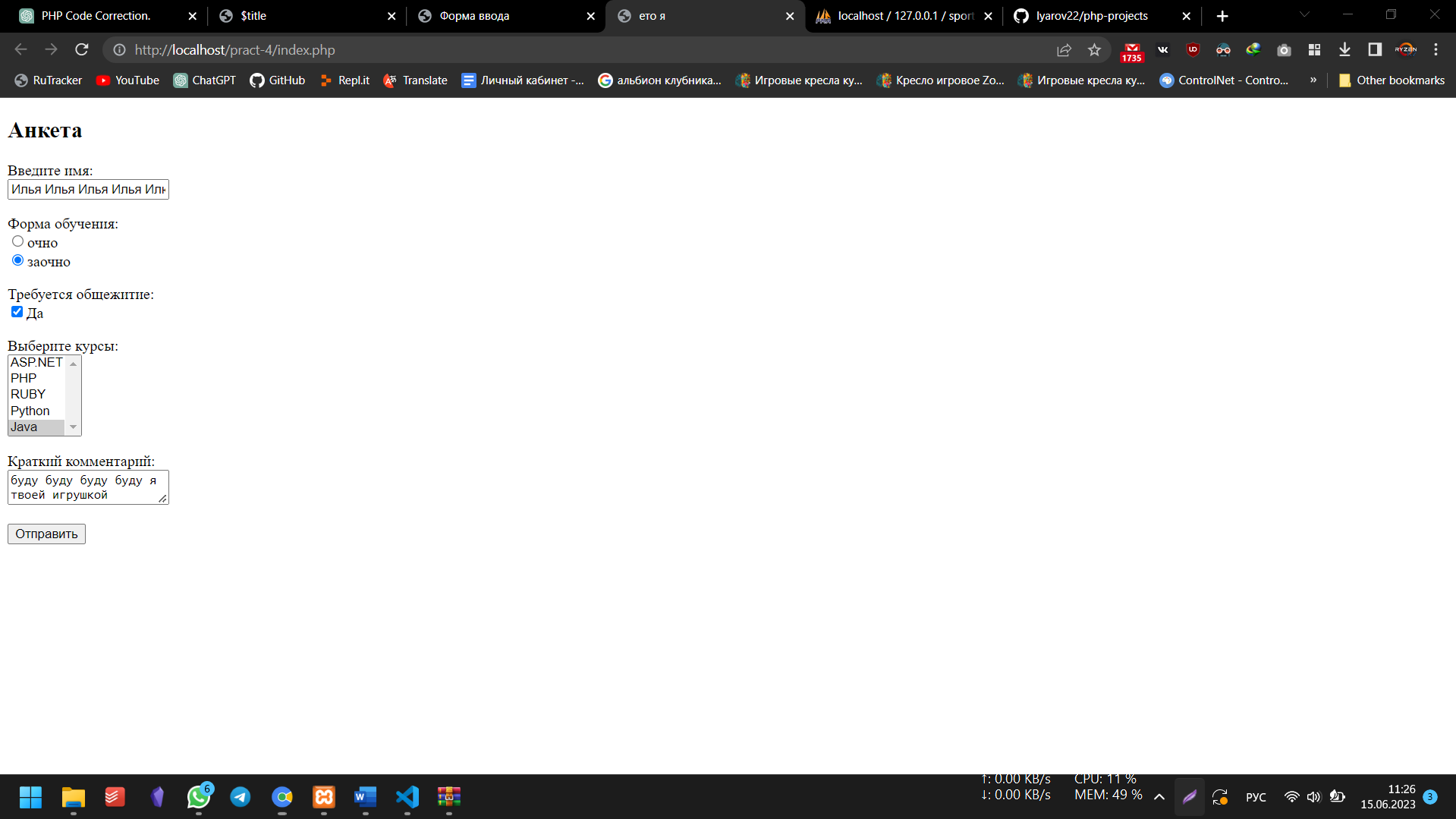
Рассмотрим комплексный пример обработки форм, в которой объединим обработку различных элементов html. Определим следующую форму:

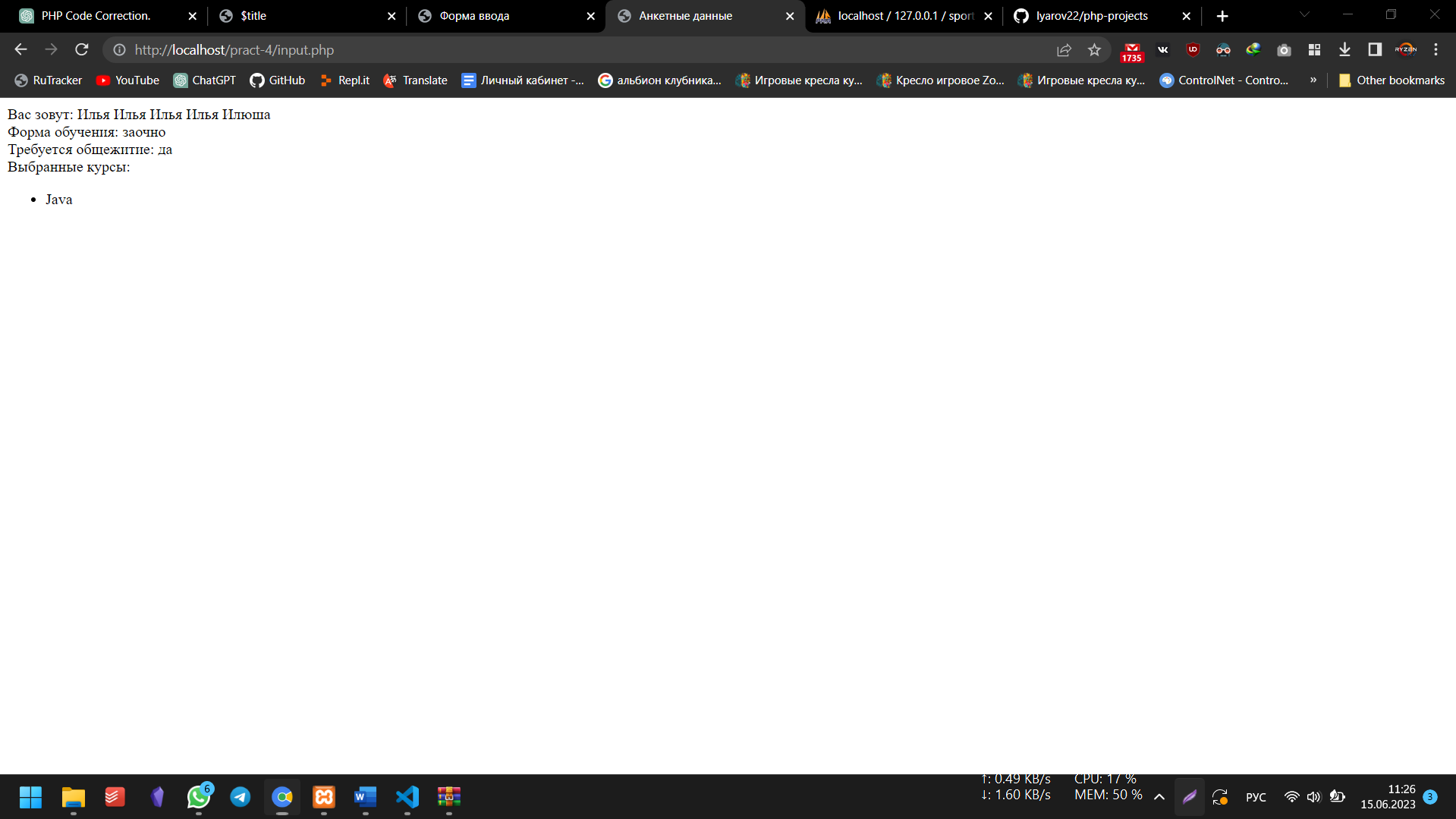
Здесь простейшая стандартная форма ввода анкетных данных:

Теперь определим скрипт *input.php*, который будет обрабатывать эту форму:

oткроем страницу с формой, введем все данные:

И отправим форму на сервер:





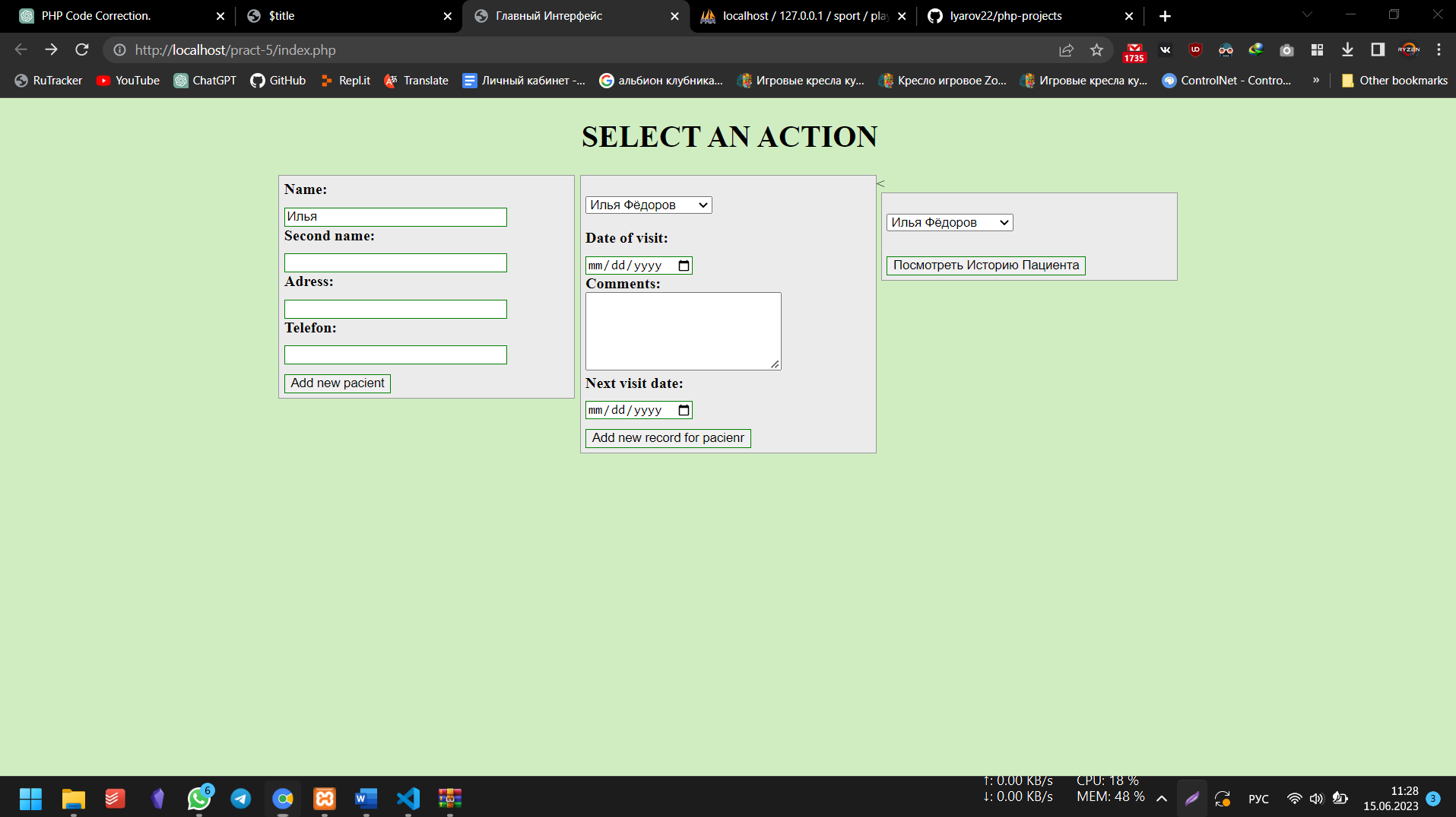
5.

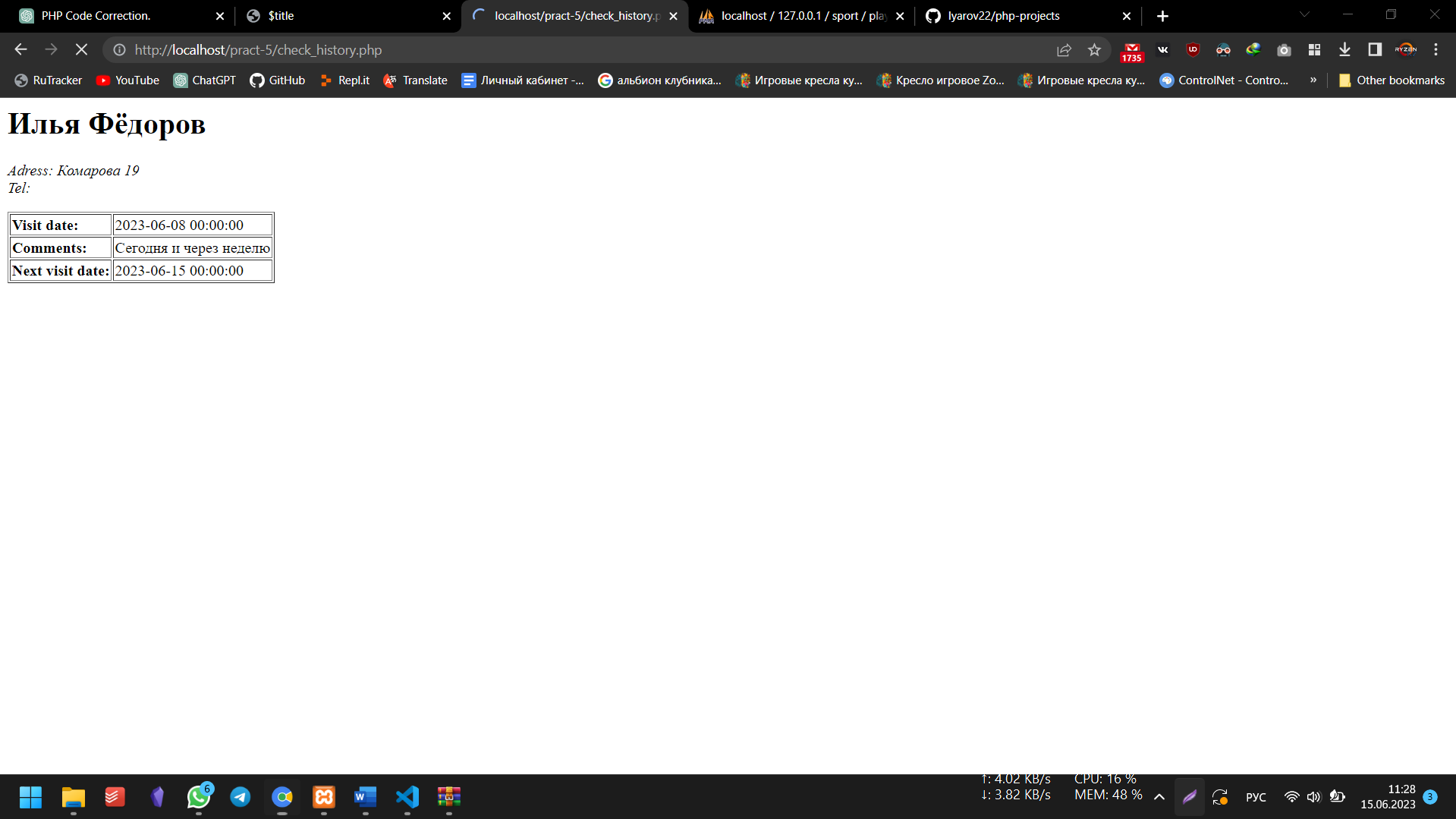
**ЧАСТЬ 2 . Заполнение таблиц БД данными, полученными из формы**

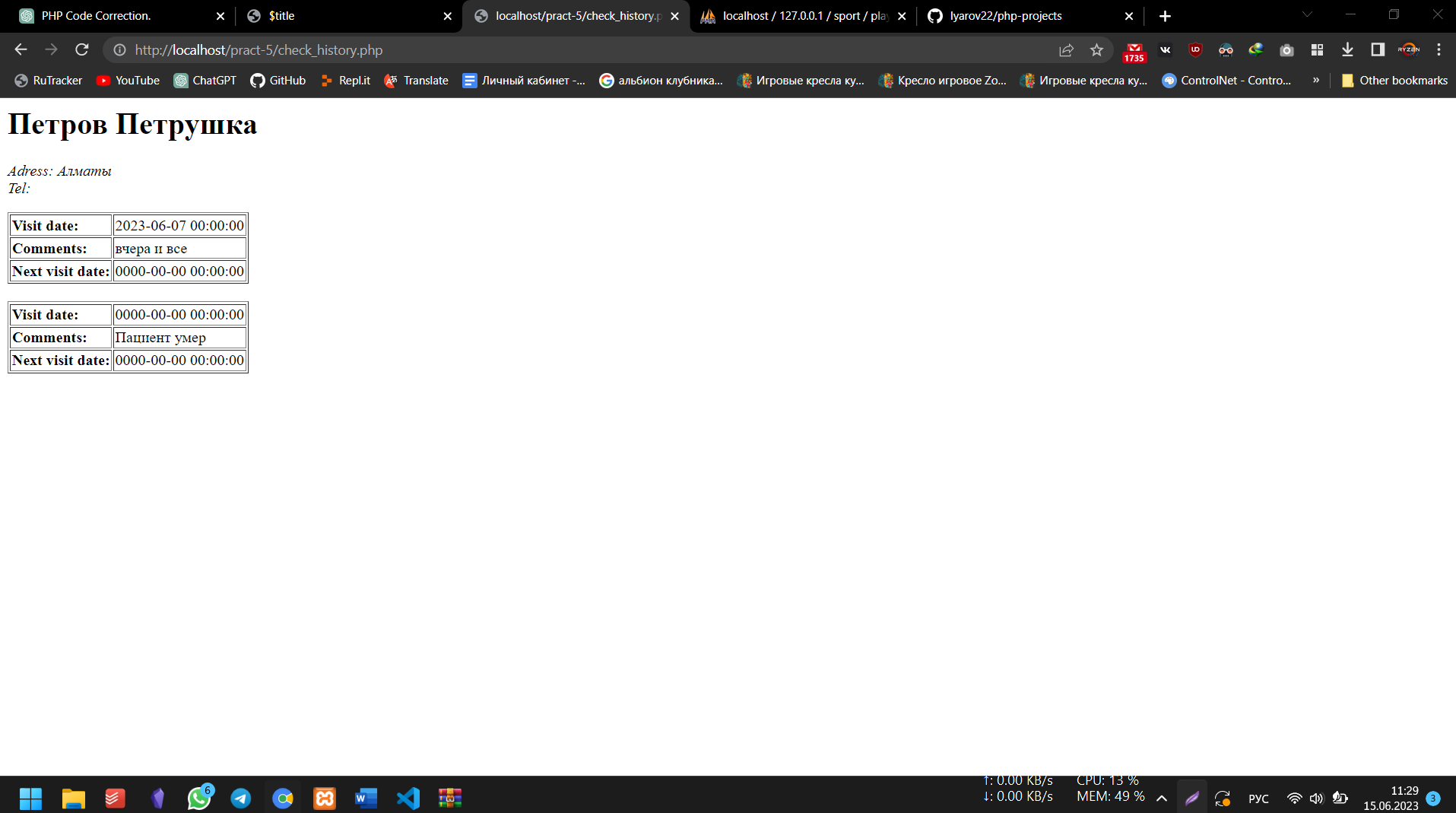
В первой части мы просто вывели данные, полученные через форму в виде суперглобального массива \_POST , в окне браузера. Теперь рассмотрим пример, когда эти данные заносятся в таблицу БД (test), таблица users поля id\_u, name, tel (id\_u - автозаполнение)

*Далее, при переходе по ссылке INSERT into table Users, попадаем в add.php*

*Далее, создаем обработчик add\_users.php*







6.

Создание простейшего веб приложения с использованием php + mysql

В phpMyAdmin создаем базу данных sport

Создаем таблицы:

Таблица countries

country\_id country\_name

1 Россия

2 Англия

3 Испания

4 Италия

5 Германия

6 Франция

8 Португалия

Таблица players

player\_id team\_id country\_id player\_name

1. 1 1 Игорь Акинфеев

2. 2 2 Криштиану Роналду

3. 6 4 Марко Вератти

Таблица teams

team\_id country\_id team\_name

1 1 ЦСКА

2 2 Манчестер Юнайтед

4 4 Лацио

5 5 Бавария

6 6 ПСЖ

8 4 Ювентус

В папке domains создаем папку sport

В папке sport создаем файлы index.php, db.php, teams.php, players.php, countries.php, api.php, edit.php,

save.php

