МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФГАОУ ВО  
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт радиоэлектроники и информационной безопасности

Кафедра «Радиоэлектроника и телекоммуникации»

**ОТЧЁТ**

по лабораторной работе №4

«Создание динамических страниц средствами PHP»

по дисциплине

«Коммуникации в сети Интернет и принципы построения сайтов»

Выполнил:

студент гр. КвИиППС/б-19-1-о

Бушуев А.В.

Защитил с оценкой: \_\_\_\_\_\_

Принял: ассистент Табакаев Д.И.

Севастополь

2020

Содержание

[1. Описание лабораторной работы 3](#_Toc43756297)

[1.1. Цель работы 3](#_Toc43756298)

[1.2. Индивидуальное задание 3](#_Toc43756299)

[1.3. Теоретические сведения 3](#_Toc43756300)

[2. Выполнение лабораторной работы 5](#_Toc43756301)

[2.1 Структура разработанного сайта 5](#_Toc43756302)

[2.2 Текст программы 7](#_Toc43756303)

[2.4 Результат выполнения работы 9](#_Toc43756304)

[3. Выводы 11](#_Toc43756305)

Описание лабораторной работы

## **Цель работы**

Освоить навыки построения динамических веб – сайтов с помощью PHP

.

## **Индивидуальное задание**

**1.2.1.** На основе верстки, созданной в лабораторной работе №3, создать сайт, используя язык программирования PHP

## **Теоретические сведения**

1.3.1. PHP – общие сведения

*PHP* – это широко используемый язык сценариев общего назначения с открытым исходным кодом.

*PHP* это язык программирования, специально разработанный для написания *web*-приложений (сценариев), исполняющихся на *Web*-сервере.

Аббревиатура *PHP* означает “*Hypertext Preprocessor* (*Препроцессор Гипертекста*)". Синтаксис языка берет начало из *C, Java* и *Perl*. *PHP* достаточно прост для изучения. Преимуществом *PHP* является предоставление web-разработчикам возможности быстрого создания динамически генерируемых web-страниц.

Важным преимуществом языка *PHP* перед такими языками, как языков *Perl* и *C* заключается в возможности создания *HTML* документов с внедренными командами *PHP*.

Значительным отличием *PHP* от какого-либо кода, выполняющегося на стороне клиента, например, *JavaScript*, является то, что *PHP*-скрипты выполняются на стороне сервера. Можно сконфигурировать свой сервер таким образом, чтобы *HTML*-файлы обрабатывались процессором *PHP*, так что клиенты даже не смогут узнать, получают ли они обычный *HTML*-файл или результат выполнения скрипта.

*PHP* позволяет создавать качественные *Web*-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем.

*PHP* прост для освоения, и вместе с тем способен удовлетворить запросы профессиональных программистов.

1.3.2 Коды состояния HTTP

Код состояния *HTTP* (англ. *HTTP status code*) — часть первой строки ответа сервера при запросах по протоколу *HTTP*. Он представляет собой целое число из трёх десятичных цифр. Первая цифра указывает на класс состояния. За кодом ответа обычно следует отделённая пробелом поясняющая фраза на английском языке, которая разъясняет человеку причину именно такого ответа. Клиент может не знать все коды состояния, но он обязан отреагировать в соответствии с классом кода. В настоящее время выделено пять классов кодов состояния:

― Информационные (100-105);

― Успешные (200-226);

― Перенаправление (300-307);

― Ошибка клиента (400-499);

― Ошибка сервера (500-510);

2. Выполнение лабораторной работы

## **2.1 Структура разработанного сайта**

**2.1.1. Главная страница**

-----------------------------------------------------------------------------------

Header

-----------------------------------------------------------------------------------

-----------------------------------------------------------------------------------

Carousel

-----------------------------------------------------------------------------------

-----------------------------------------------------------------------------------

Content (main)

-----------------------------------------------------------------------------------

-----------------------------------------------------------------------------------

Footer

-----------------------------------------------------------------------------------

**2.1.2. Страница с формой**

-----------------------------------------------------------------------------------

Header

-----------------------------------------------------------------------------------

-----------------------------------------------------------------------------------

Form

-----------------------------------------------------------------------------------

-----------------------------------------------------------------------------------

Footer

-----------------------------------------------------------------------------------

**2.1.3. Страница с таблицей**

-----------------------------------------------------------------------------------

Header

-----------------------------------------------------------------------------------

-----------------------------------------------------------------------------------

Table

-----------------------------------------------------------------------------------

-----------------------------------------------------------------------------------

Footer

-----------------------------------------------------------------------------------

## **2.2 Текст программы**

**2.2.1. Содержимое файла index.php**

<?php

require\_once ('vendor/autoload.php');

use SimpleRouter\handlers\IRequestHandler;

use SimpleRouter\Router;

use SimpleRouter\request\Request;

$router = Router::getInstance();

$router->get("/", new class implements IRequestHandler {

public function handle(Request $req) {

if (!is\_null($req->getRequestQueryValueByKey("id"))) {

switch ($req->getRequestQueryValueByKey("id")) {

case "form" : print(file\_get\_contents($\_SERVER['DOCUMENT\_ROOT']."/inc/table.html")); break;

case "table" : print(file\_get\_contents($\_SERVER['DOCUMENT\_ROOT']."/inc/form.html")); break;

default : print(file\_get\_contents($\_SERVER['DOCUMENT\_ROOT']."/inc/table.html")); break;

}

} else {

print(file\_get\_contents($\_SERVER['DOCUMENT\_ROOT']."/inc/index.html"));

}

}

});

$router->get("/form", new class implements IRequestHandler {

public function handle(Request $req) {

print(file\_get\_contents($\_SERVER['DOCUMENT\_ROOT']."/inc/form.html"));

}

});

$router->get("/table", new class implements IRequestHandler {

public function handle(Request $req) {

print(file\_get\_contents($\_SERVER['DOCUMENT\_ROOT']."/inc/table.html"));

}

});

$router->handle();

## **2.4 Результат выполнения работы**

Был реализован сайт, имеющий следующие маршруты:

* / - Главная страница
* /form (или /?id=form) – Форма
* /table (или /?id=table) – Таблица

Роутинг был реализован с помощью самостоятельно созданной библиотеки (<https://github.com/aleksandrBushuev2000/simple-router>)

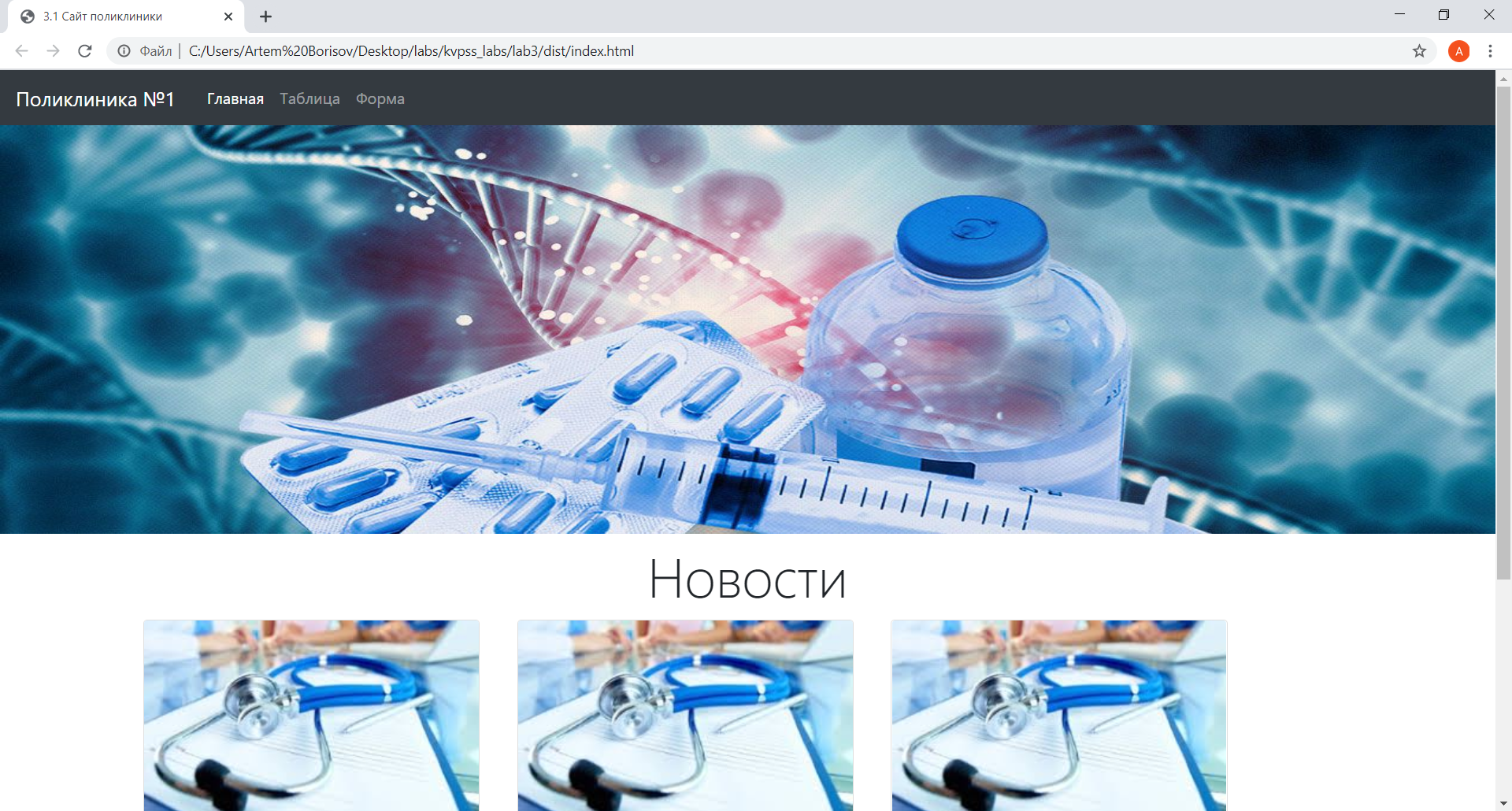


Рисунок 1 – Главная страница сайта

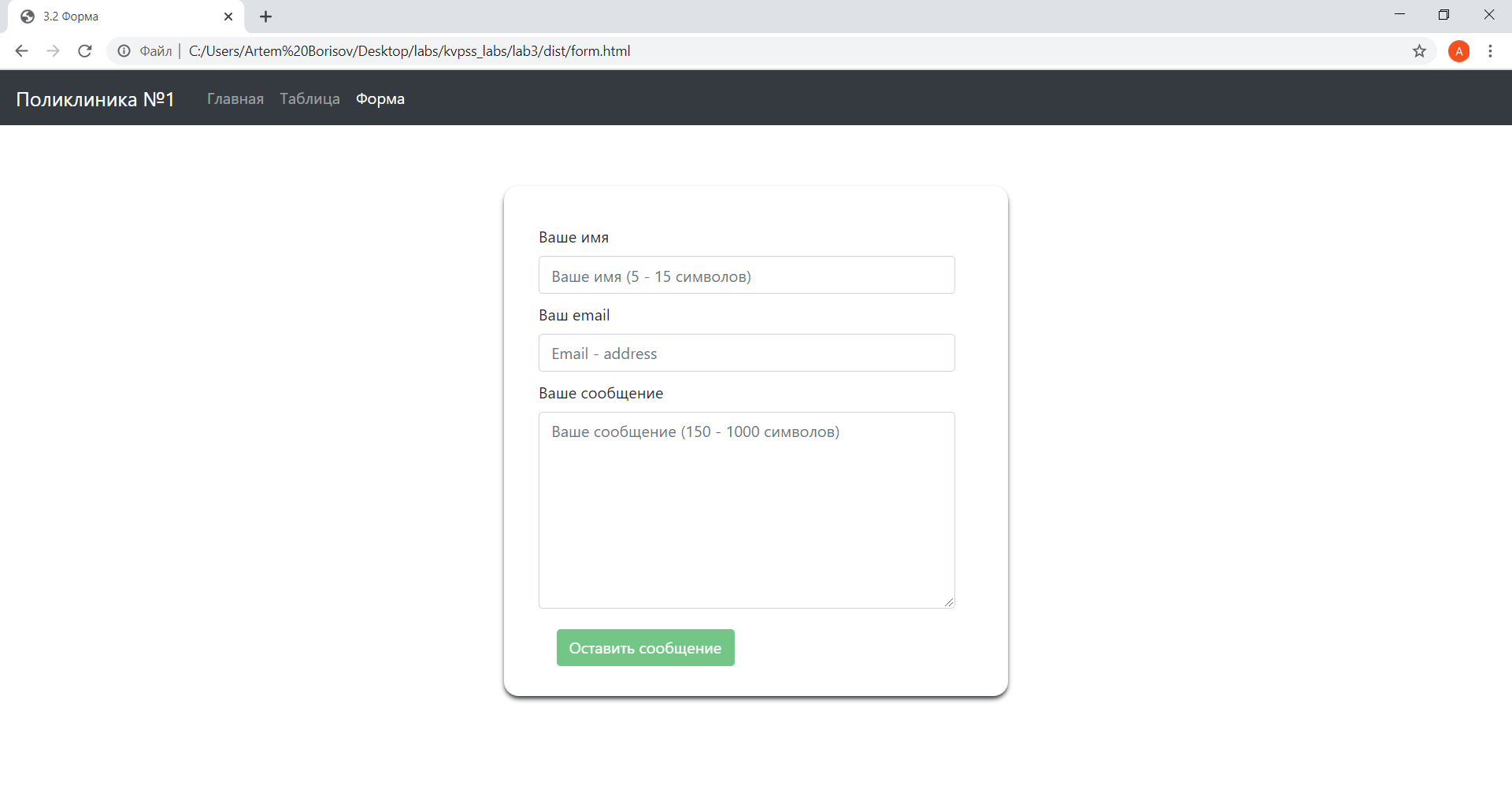


Рисунок 2 – Страница с формой

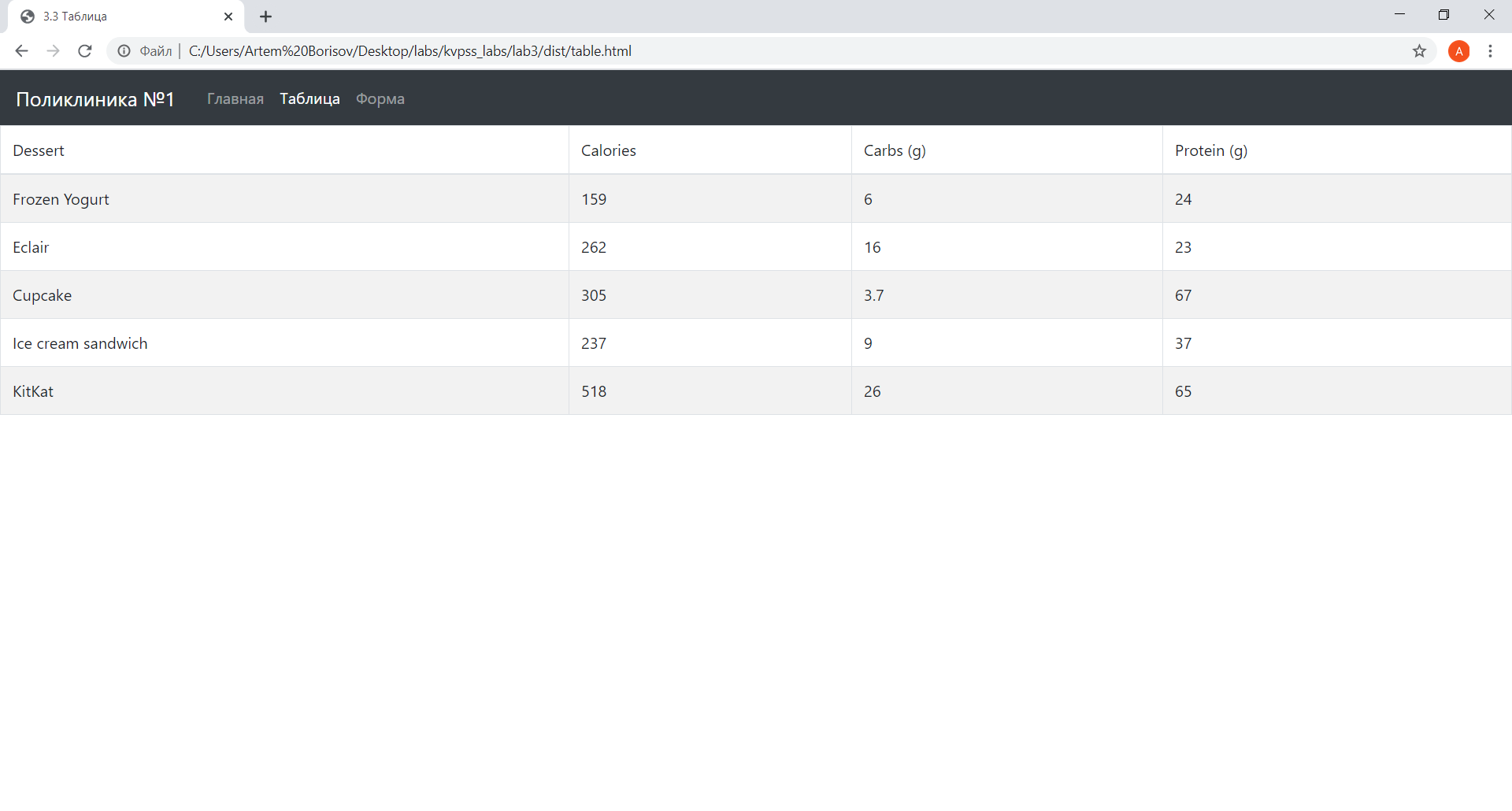


Рисунок 3 – Страница с таблицей

3. Выводы

В ходе выполнения лабораторной работы были изучены основы языка программирования PHP, а также был реализован сайт с использованием этой технологии