НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет «инфокоммуникационных технологий»

Направление подготовки «инфокоммуникационные технологии и системы связи»

Лабораторная работа №3

«Использование Git и Gulp для решения задач web-разработки»

Выполнил:

Ишбаев Эрик Аликович

Группа К33201

Проверила:

Марченко Елена Вадимовна

Санкт-Петербург

2023

# Цель работы:

Научиться работать с системой контроля версий Git для отслеживания и ведения истории изменения файлов в проектах, познакомиться с инструментом для автоматизации, организации и обработке задач Gulp, разработать собственное веб-приложение для просмотра веб-страниц.

# Ход работы:

# Упражнение 1.

Задание:

Установить Git на компьютер (локальный репозиторий), настроить на работу с проектом (можно использовать файлы предыдущей работы), выполнить изменения в файлах проекта. Для выполняемых изменений сделать коммиты (не менее трех). Проверить, что коммиты создаются. Локальный репозиторий синхронизировать с удаленным (можно использовать GitHub, GitLab, др.). Привести в отчете ссылку на проект.

Реализация:

Рисунок - Утилита MAMP

Напишем код HTML страницы для web-формы с применением имеющихся знаний в HTML и CSS (Рисунок 2).

<html>

<head>

<title>Оформление заказа</title>

<meta charset="utf-8">

<style>

body {

font-family: "Impact", fantasy;

background-image: url('back.jpeg'); /\* Путь к вашей фоновой фотографии \*/

background-size: cover;

background-position: center;

height: 100vh; /\* Занимать 100% высоты экрана \*/

}

form {

background-color: rgba(255, 255, 255, 0.7); /\* Прозрачный белый фон формы \*/

padding: 20px;

border-radius: 10px;

text-align: center;

width: 50vh;

margin: auto;

}

h1 {

color: #F8F8FF;

text-align: center;

margin-bottom: 20px;

margin-top: 20px;

}

</style>

</head>

<body>

<h1>Оформление заказа</h1>

<form action="script.php" method="POST">

<p>Фамилия: <input type="text" name="surname" required></p>

<p>Имя: <input type="text" name="name" required></p>

<p>Отчество: <input type="text" name="fathername" required></p>

<p>Адрес Доставки: <input type="text" name="address" required></p>

<p>Номер телефона: <input type="tel" name="phone" required></p>

<p>Email: <input type="email" name="email" required></p>

<p>Выберите товар:

<select name="product">

<option value="Iphone">Iphone</option>

<option value="Samsung">Samsung</option>

<option value="Xiaomi">Xiaomi</option>

<option value="Pixel">Pixel</option>

<option value="Honor">Honor</option>

</select></p>

<p> Комментарий к заказу:<br>

<textarea name = "comment" >

</textarea>

</p>

<input type="submit" value="Отправить заказ">

</form>

</body>

</html>

Рисунок - Код HTML страницы

Посмотрим, как она выглядит в браузере:

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок - Web-форма в браузере

Напишем PHP скрипт, который будет принимать POST-запрос из web-формы и добавлять в базу данных, созданную заранее в MySQLServer. База данных создается в PhpMyAdmin панели (Рисунок 4).

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок - Структура таблицы заказов

<?php

$servername = "localhost";

$username = "root";

$password = "root";

$dbname = "forms";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$surname = $\_POST['surname'];

$name = $\_POST['name'];

$fathername = $\_POST['fathername'];

$address = $\_POST['address'];

$phone = $\_POST['phone'];

$email = $\_POST['email'];

$product = $\_POST['product'];

$comment = $\_POST['comment'];

$sql = "INSERT INTO `Orders`(`surname`, `name`, `fathername`, `address`, `phone`, `email`, `product`, `comment`) VALUES ('$surname', '$name', '$fathername', '$address', '$phone', '$email', '$product', '$comment')";

if ($conn->query($sql) === TRUE) {

echo "Заказ успешно оформлен";

} else {

echo "Ошибка: " . $sql . "<br>" . $conn->error;

}

$conn->close();

?>

Рисунок 5 – PHP скрипт

Проверим работоспособность web-формы и попробуем добавить данные:

Изображение выглядит как Шрифт, текст, снимок экрана, белый

Автоматически созданное описание

Рисунок - Уведомление о успешном оформлении заказа

Видим, что данные успешно добавлены в таблицу:

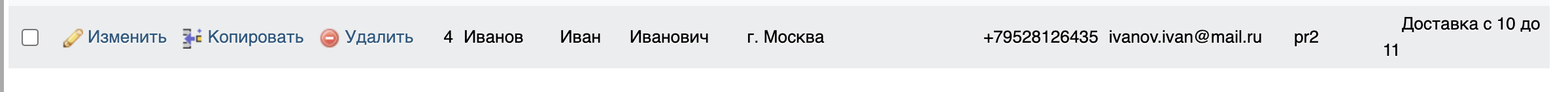


Рисунок - Заказ в таблице Orders

# Упражнение 2.

Задание:

В движке wordpress используется форма для авторизации пользователей – пользователь вводит логин и пароль. Логин и хэш от пароля записываются в таблицу БД MySQL. Необходимо модифицировать php скрипт, чтобы дополнительно логин и пароль записывались в отдельную новую таблицу в БД MySQL. Предусмотреть два варианта записи:

a) пароль записывается в исходном виде

б) инвертируются биты бат пароля (ноль меняется на единицу, единица меняется на ноль)

Установим wordpress и создадим тему с главным экраном:

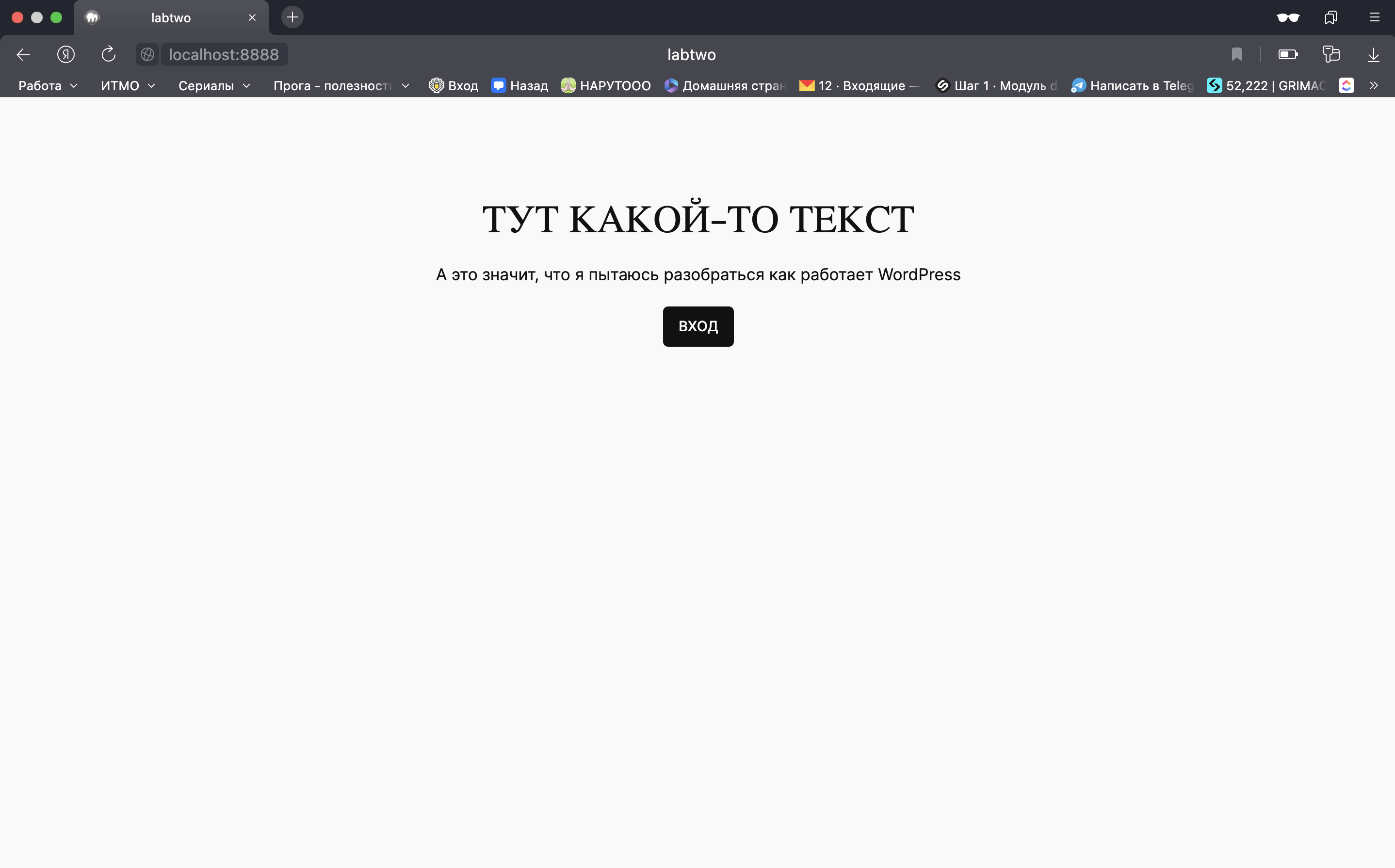


Рисунок - Главный экран web-страницы

Кнопка “ВХОД” введет на страницу регистрации, которая изначально существует в движке wordpress. Так как изначально не предусмотрено возможности использовать данные авторизации для добавления их в другую базу данных, было принято решение использовать плагин “Code Snippets” (Рисунок 8).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок - Установка плагина в wordpress

Данный плагин позволяет дополнять движок worpdress своими скриптами. Был написан скрипт, который добавляет данные авторизации в таблицу “castom\_users” (Рисунок 9, 10, 11).

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок - Код PHP-скрипта

Изображение выглядит как программное обеспечение, Мультимедийное программное обеспечение, снимок экрана, Значок на компьютере

Автоматически созданное описание

Рисунок - Окно авторизации

Изображение выглядит как снимок экрана, текст, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок - Добавленные в таблицу данные

# Упражнение 3.

Реализуем web-server с помощью языка Python и фреймвока Flask (Рисунок 12).

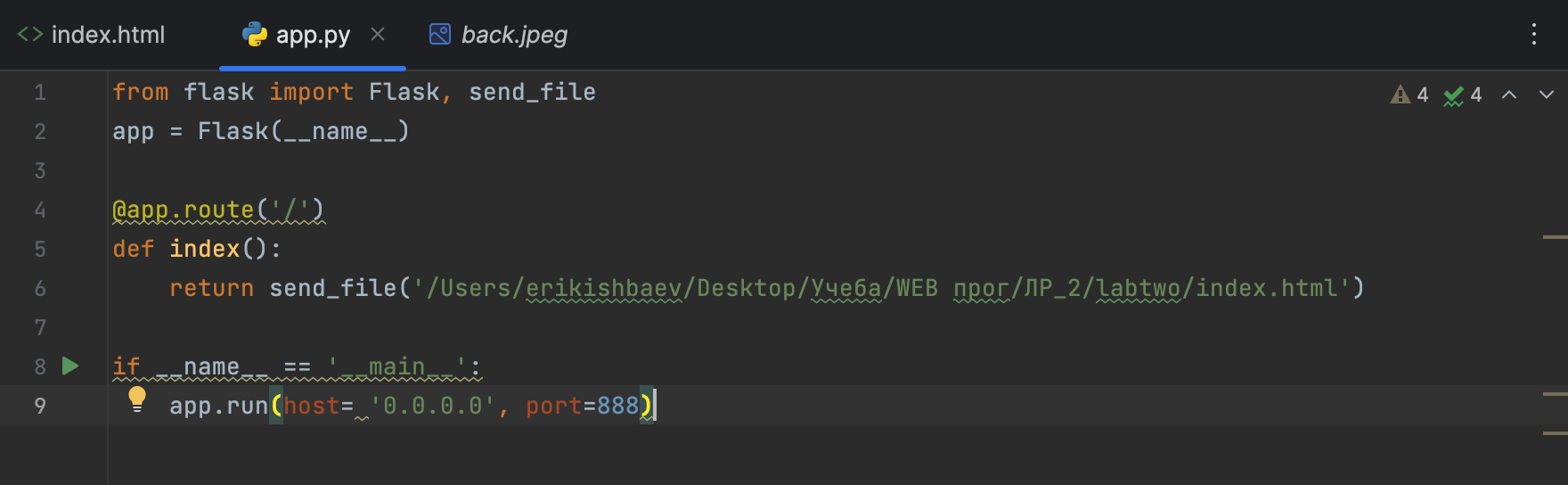


Рисунок - Код к упражнению

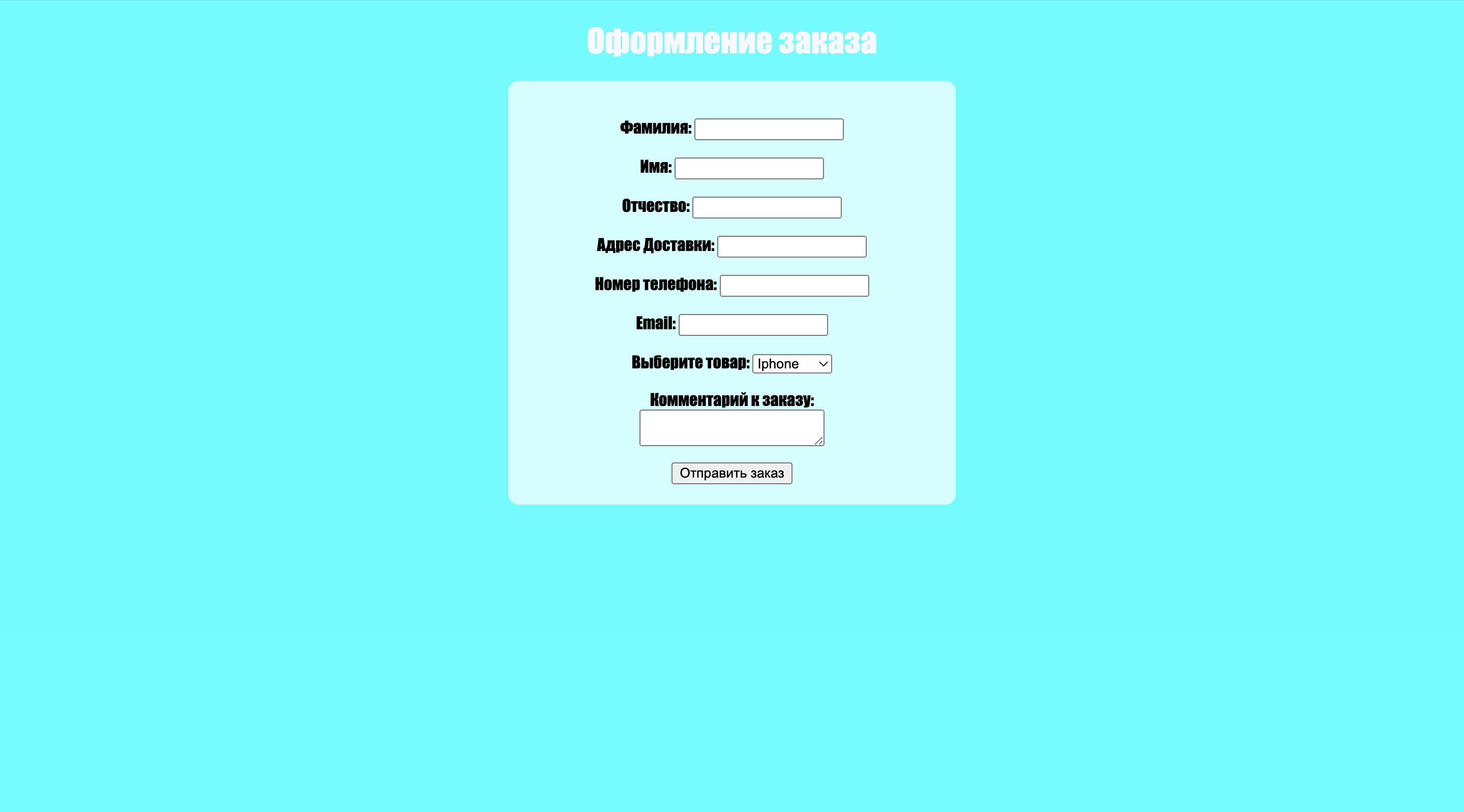


Рисунок – HTML страница

# Выводы:

Познакомились с основами создания web-форм, использующих HTML, стили CSS, PHP скрипты, базы данных MySQLServer и движок wordpress.