САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет инфокоммуникационных технологий

Направление подготовки «Программирование в инфокоммуникационных системах»

Лабораторная работа №4

«Разработка web-форм»

Выполнил:

Смирнов К.А.

Группа №33201

Проверила:

Марченко Е.В.

Санкт-Петербург

2023

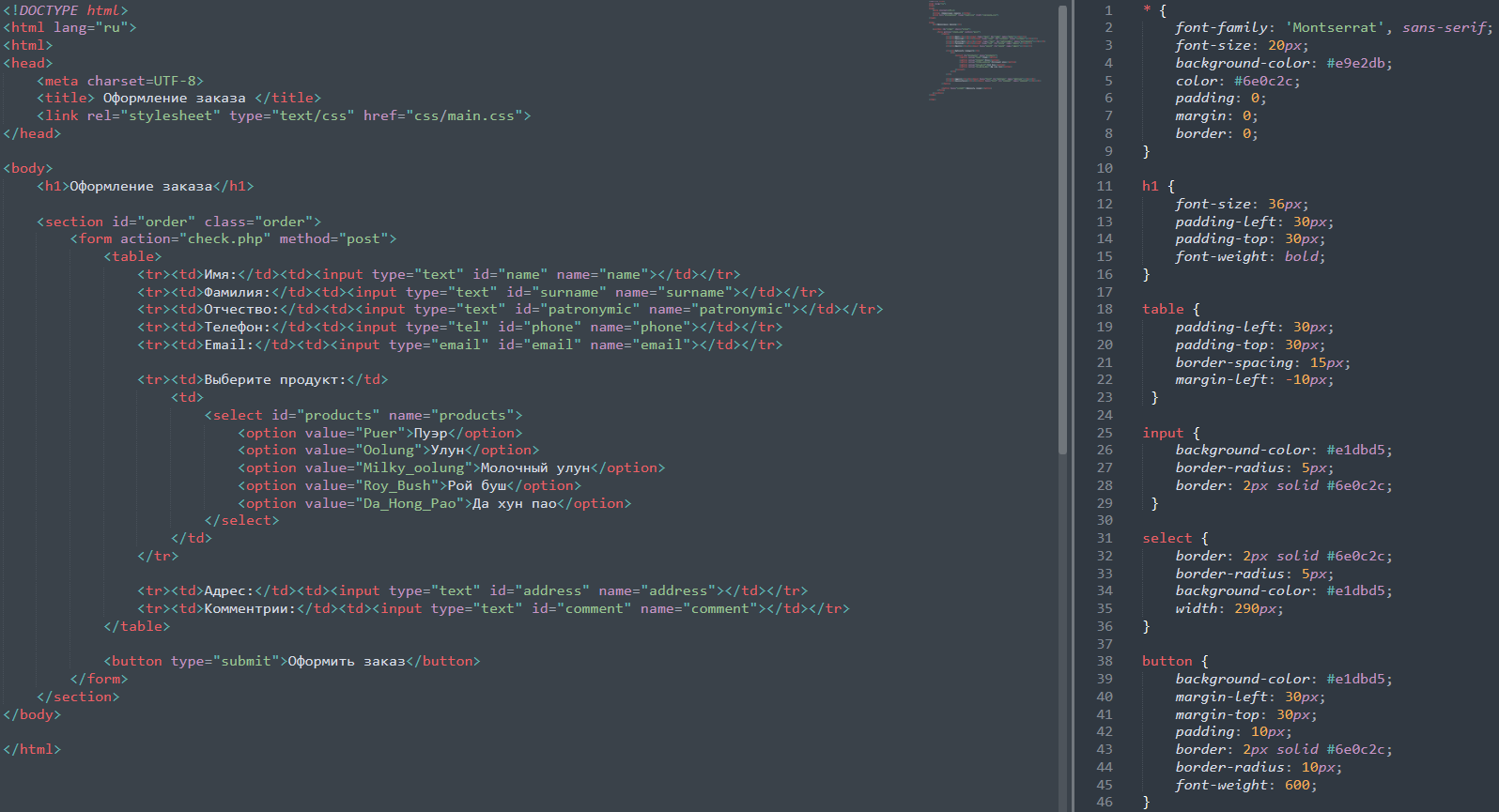
Цель работы:

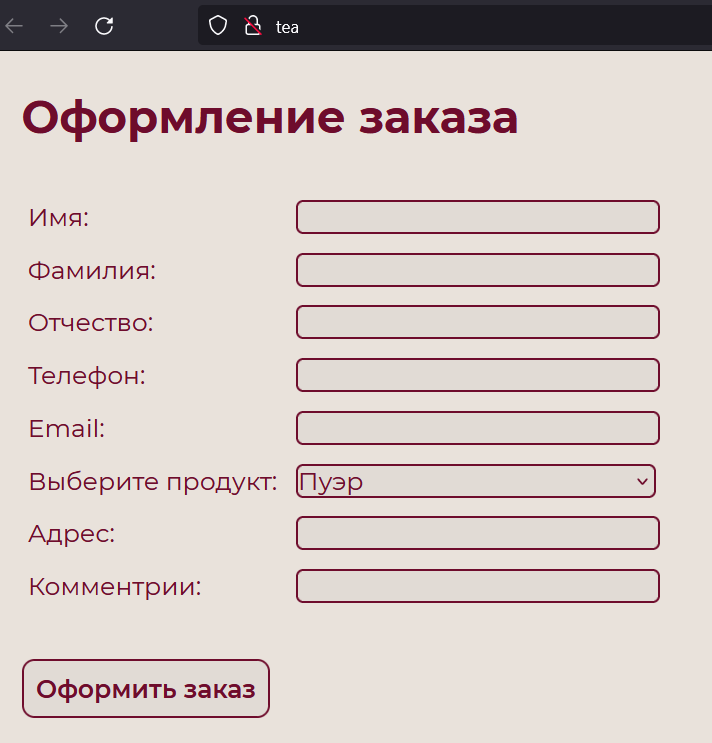
Научиться разрабатывать простейшие формы, изучить шаги, необходимые для корректной работы php скрипта c таблицей БД MySQL, а именно: 1. Установку связи с MySQL сервером  
2. Отправку данных MySQL серверу и получение от него ответов 3. Создание таблиц БД MySQL 4. Запись данных в таблицу БД MySQL. Разобраться в общем устройстве формы регистрации новых пользователей движка WordPress. Рассмотреть возможность написания локального сервера на языке python.

Ход работы:

Задание 1

Выполнение задания было начато с создания облика веб-страницы при помощи html и CSS. В качестве каркаса для формы использована структура таблиц, для сохранения выравнивания. Ячейки ввода данных заказа сделаны при помощи тега input, а меню товаров при помощи тега select. На веб-страницу также была добавлена кнопка button для оформления заказа с атрибутом type=”submit”, значение submit определяет, что данная кнопка будет отправлять данные формы на сервер. Сама форма организована при помощи одноименного тега form, в качестве атрибутов которого указаны action=”check.php” и method=”post”, первый отвечает за используемый обработчик, к которому обращаются данные формы при их отправке на сервер; Второй определяет метод, при помощи которого будут отправлены данные на сервер. Написанный html и CSS коды представлены на рисунке 1, а внешний облик веб-страницы на рисунке 2.

  
Рисунок 1 – html и CSS коды веб-страницы оформления заказа

  
Рисунок 2 – форма оформления заказа

Данные должны быть обработаны, потому следующим шагом было написание php скрипта. Сперва создан файл db.php, отвечающий за подключение к БД и проверку этого подключения (см. рис. 3). Далее, данные, полученные в форму, необходимо записать в базу данных, для этого был создан скрипт check.php, в начале которого прописана функция require\_once('db.php'), отвечающая за подключение файла db.php к check.php. В данном скрипте создаются переменные, значения которых берутся из массива POST запроса, после чего создаётся переменная sql, в которую записан запрос INSERT, при этом все переменные заменены на специальные маркеры-плейсхолдеры, для того, чтобы данные в БД попадали отдельно от самого запроса, поскольку таким образом можно избавиться от атак с использованием SQL-инъекций и обеспечить более безопасную передачу данных в базу данных, так как все значения передаются через параметры, а не напрямую через запрос. После, функция $conn->prepare($sql) создает подготовленное выражение на основе SQL-запроса и к параметрам подготовленного выражения привязываются переменные при помощи функции $stmt->bind\_param. Далее функция $stmt->execute() выполняет подготовленное выражение с переданными значениями и вставляет эти значения в таблицу users.

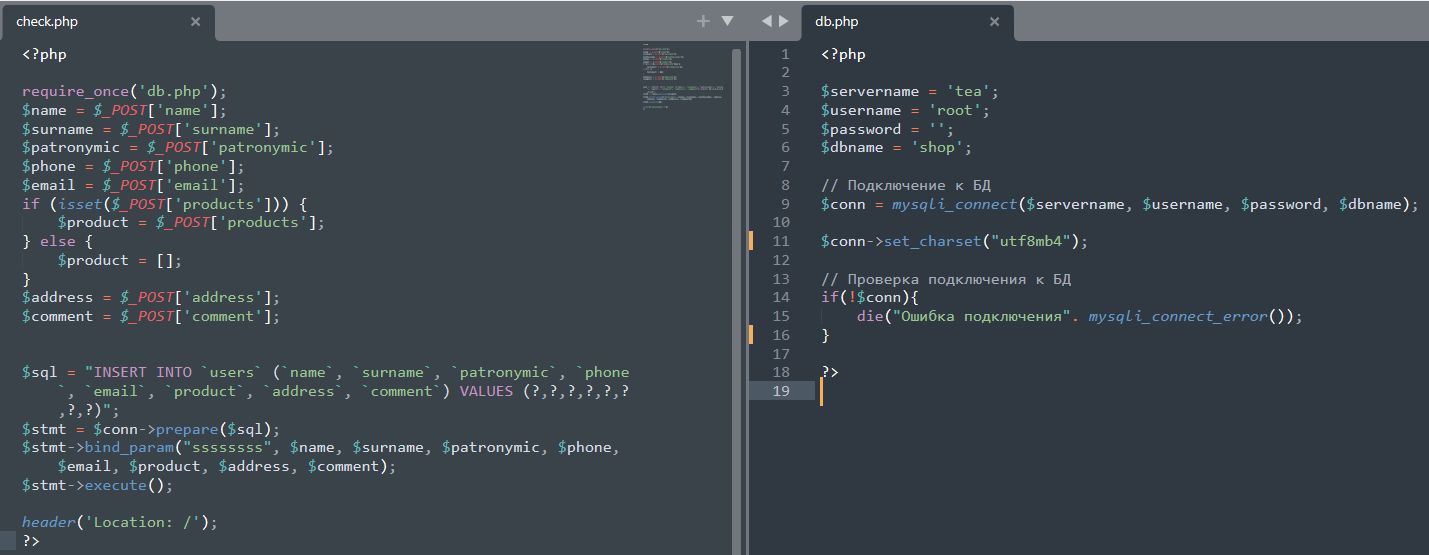


Рисунок 3 – db.php и check.php скрипты

После написания скрипта была создана база данных shop с таблицей users, которая содержит 9 строк, в восьми из которых определён элемент формы (name, surname, patronymic и т.д.). В первой строке таблицы определён id с индексом PRIMARY, определяющий id, как первичный ключ, к нему добавлен механизм автоинкремента, позволяющий автоматически генерировать id. Полученная таблица изображена на рисунке 4.

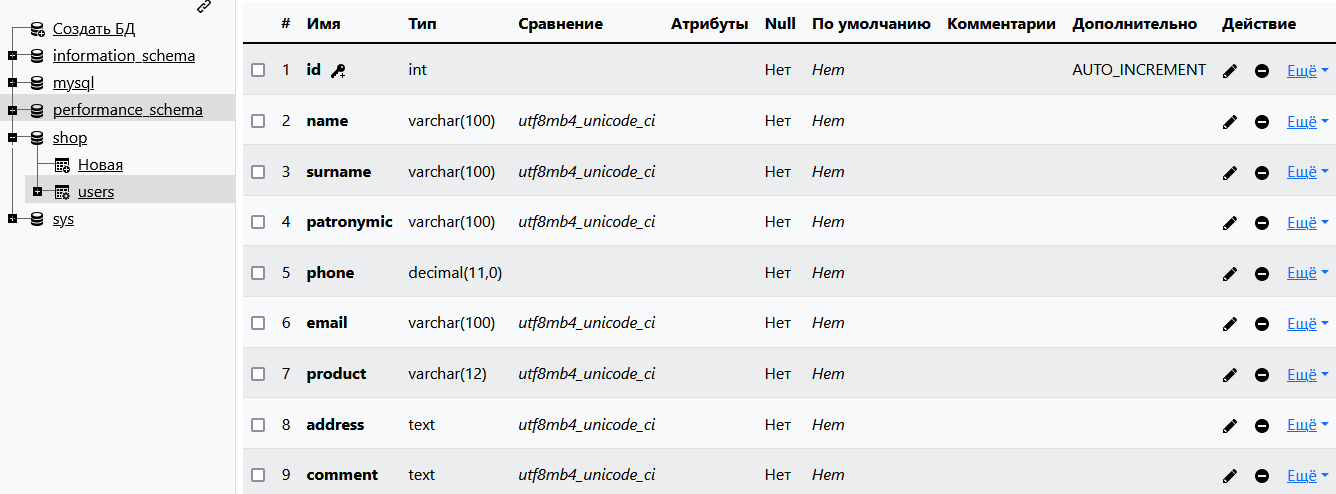
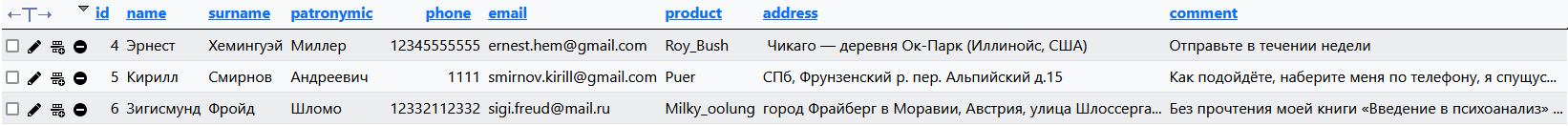


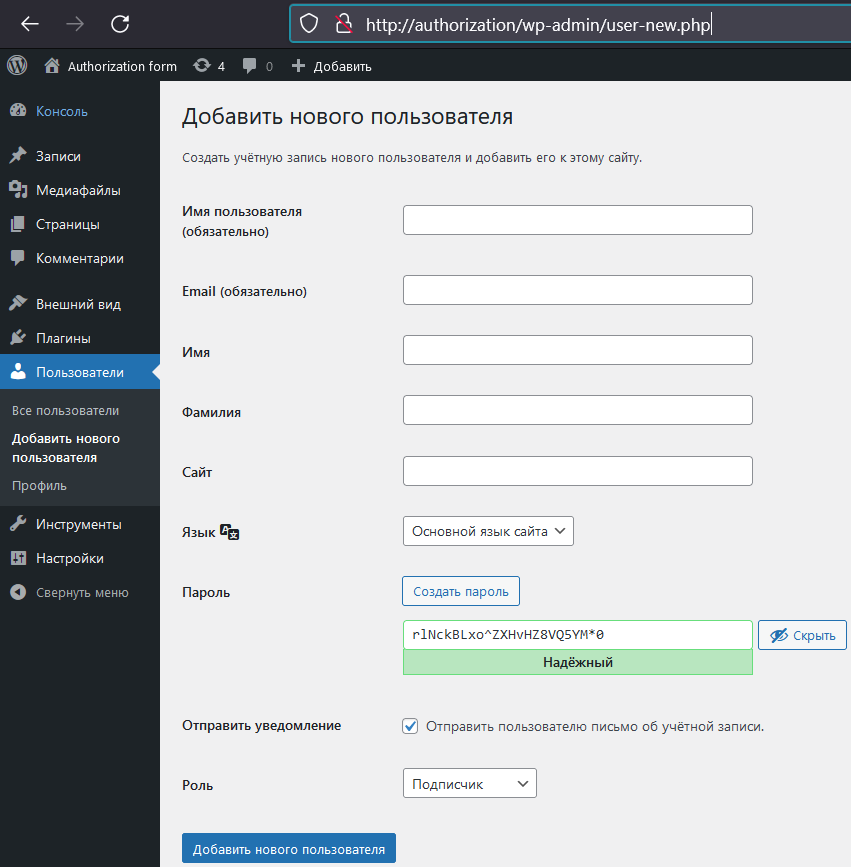
Рисунок 4 – таблица БД MySQL

В качестве локального сервера был выбран Open Server Panel. Работа с БД велась через предустановленную в Open Server программу – PhpMyAdmin. Запустив локальный сервер, авторизовавшись в PhpMyAdmin и заполнив форму, можно нажать на кнопку «Оформить заказ» после чего выполняются написанные php скрипты и в БД появляется запись с данными, полученными из формы (см. рис. 5).

  
Рисунок 5 – занесённые в таблицу БД MySQL данные формы

Задание 2

Был установлен WordPress и произведена авторизация. Для начала необходимо было перейти во вкладку «Пользователи / Добавить нового пользователя», поскольку она отвечает за регистрацию новых пользователей WordPress. После перехода по упомянутой выше вкладке, открывается «http://authorization/wp-admin/user-new.php» (см. рис. 6) из URL-адреса которой можно понять, что код данной формы регистрации находится в папке wp-admin в файле user-new.php. Для того, чтобы логин и пароль записывались в отдельную новую таблицу в БД MySQL, необходимо найти форму и перенаправить POST-запрос, отправляемый после нажатия кнопки «Добавить нового пользователя», в свой php файл и уже через него сделать запись в отдельную базу данных. Искомая форма была найдена при помощи исследования кодом элемента (см. рис. 7-8). POST-запрос был перенаправлен в кастомный файл check.php при помощи добавления в найденную форму атрибута action с указанием пути к этому файлу (см. рис. 8).

  
Рисунок 6 – страница добавления новых пользователей

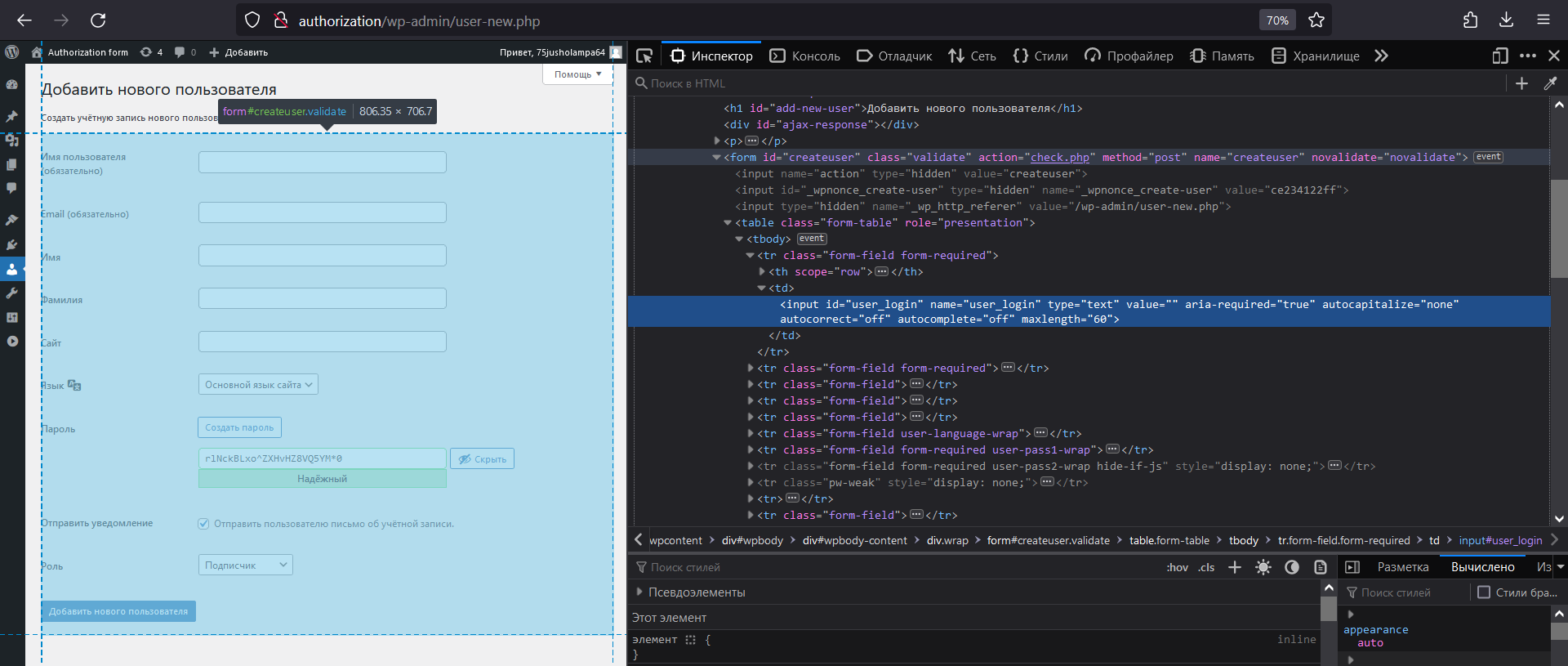
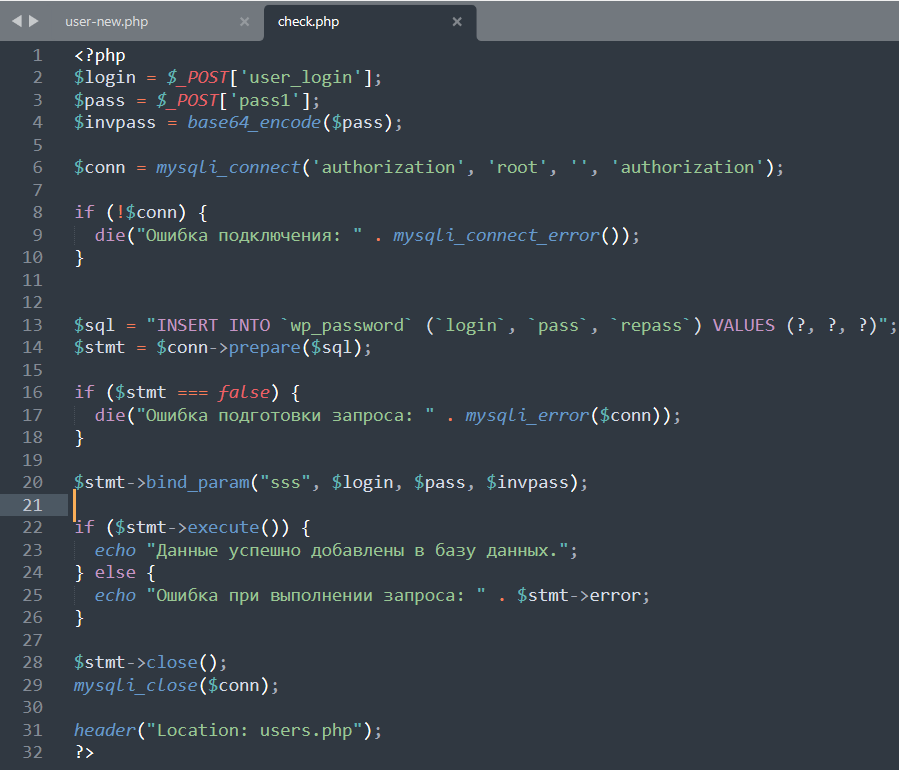
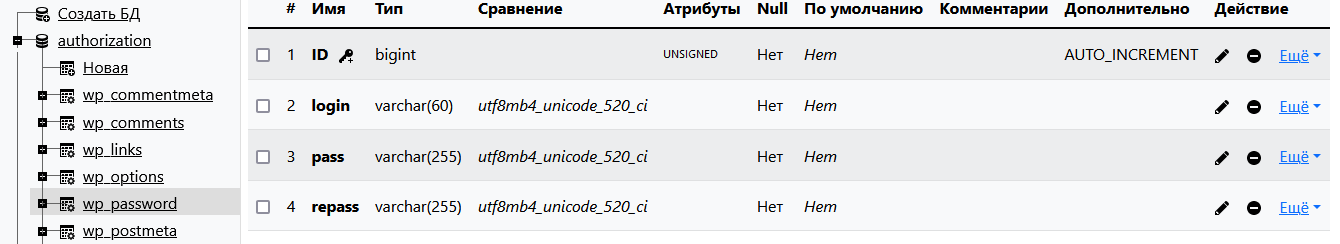
  
Рисунок 7 – исследование формы

  
 Рисунок 8 – найденная форма в проекте с WordPress

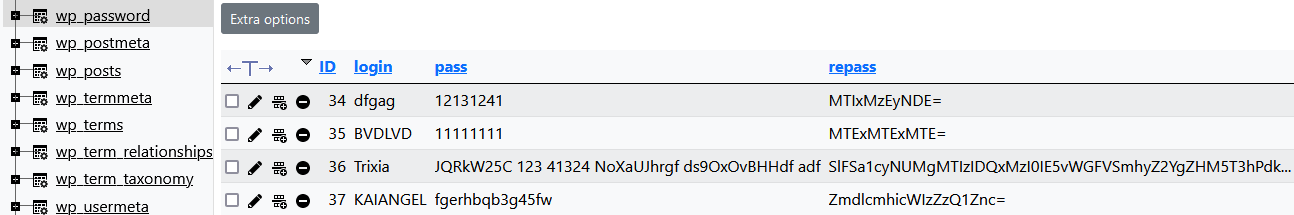
В файле check.php был написан код (см. рис. 9), который перехватывает данные, полученные в форме, подключается к отдельной БД MySQL – «wp\_password» и записывает в неё логин, пароль в исходном виде и закодированный пароль.

  
Рисунок 9 – код файла check.php

Далее была создана таблица в базе данных проекта с названием «wp\_password». Структура таблицы имеет следующий вид (см. рис. 10):

  
Рисунок 10 – структура таблицы «wp\_password»

После проделанных шагов при заполнении формы данными пользователя, происходит запись в вышеописанную таблицу (см. рис. 11) при этом есть запись пароля как в исходном, так и в закодированном виде.

  
Рисунок 11 – обзор записанных в БД пользователей

Задание 3

Написание web-сервера производилось в среде разработки «Visual Studio Code» на языке Python. Данный код (см. рис. 12) написан с использованием модулей: sys - предоставляет доступ к переменным и функциям, связанным с интерпретатором Python; http.server - предоставляет базовые классы для создания HTTP-серверов; socketserver - предоставляет базовые классы для обработки сокетов и серверов. При запуске сервера у пользователя есть возможность вручную задать порт (см. рис. 13). При указании адреса локального хоста с заданным портом и протоколом http в браузере Mozilla запускается файл index.html (см. рис. 14), который находится в той же директории, что и файл с кодом сервера.

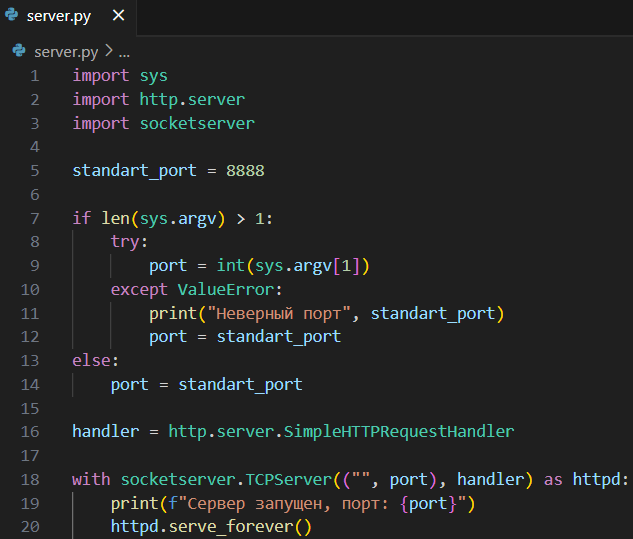
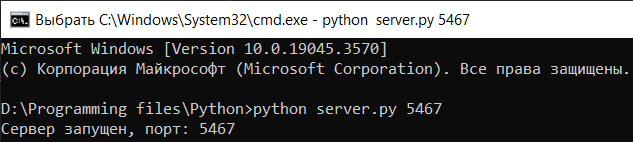
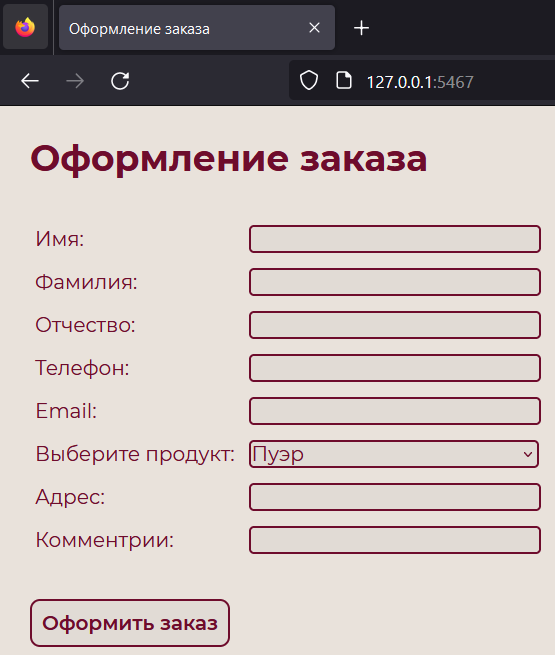


Рисунок 12 – код сервера

  
Рисунок 13 – запуск сервера с заданным портом

  
Рисунок 14 – результат работы сервера

Вывод

В ходе выполнения лабораторной работы было освоено создание простых форм, взаимодействие PHP с MySQL и умение устанавливать связь с MySQL сервером. Также была изучена форма регистрации в WordPress, что расширило понимание аутентификации и регистрации в веб-приложениях. Был написан локальный сервер на Python, что расширило инструментарий разработки web-приложений. Лабораторная работа дала необходимые базовые навыки для работы с PHP, MySQL, WordPress и Python.