УНИВЕРСИТЕТ ИТМО

Факультет «Инфокоммуникационных технологий»

Направление подготовки «Программирование в инфокоммуникационных системах»

Лабораторная работа №4

Выполнил:

Горлов Игорь Витальевич

Группа №K3321

Проверила:

Марченко Елена Владимировна

Санкт-Петербург

2024

**Цель работы:**  
Разработать функциональный и безопасный веб-сервис для обработки и хранения пользовательских данных, реализации дополнительных методов аутентификации, а также создания собственного web-сервера с настройкой порта для локального доступа.

**Задачи:**

1. Создать веб-страницу для ввода данных пользователя, включающую поля для фамилии, имени, отчества, адреса, телефона, email, выбора товаров и добавления комментария.
2. Модифицировать существующий PHP-скрипт для авторизации в движке WordPress.
3. Разработать собственный web-сервер с возможностью указания порта.

Ссылка на удаленный репозиторий:

https://github.com/kew2023/WebDevelopment\_2024-2025

**Ход работы**:

**Задание 1.**

Для выполнения данной задачи создадим в файле ***functions.php*** нашей темы *шорткод* для создания необходимой формы.

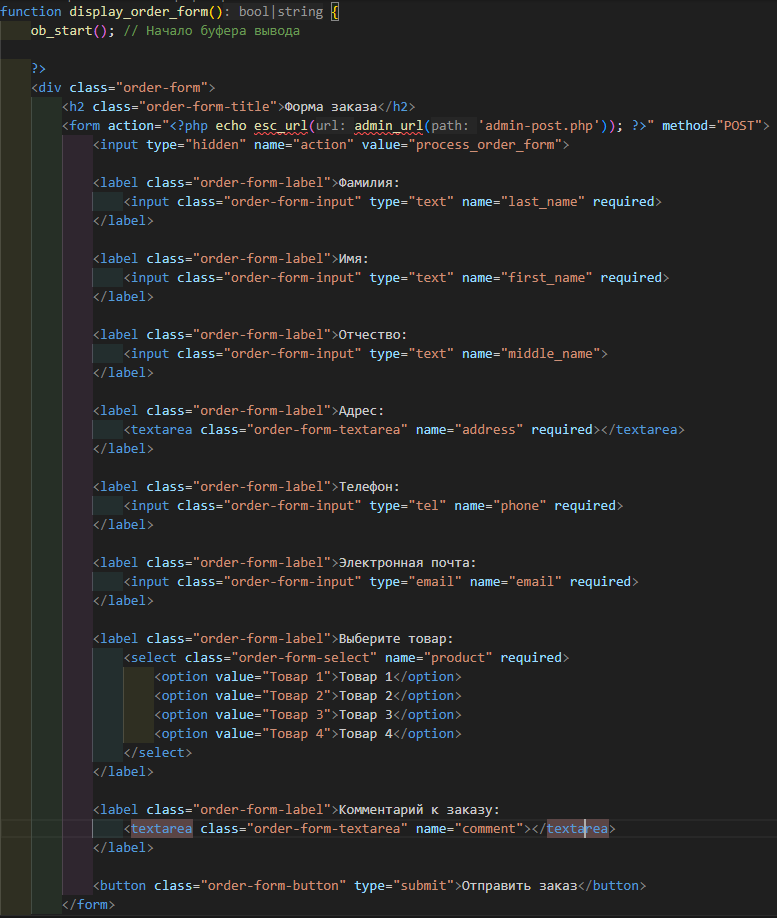


Рисунок 1 – Создание формы



Рисунок 2 – Регистрация шорткода

Далее вставим полученный шорткод на страницу в WordPress.

Теперь создадим таблицу в базе данных для обработки формы следующим SQL-запросом.

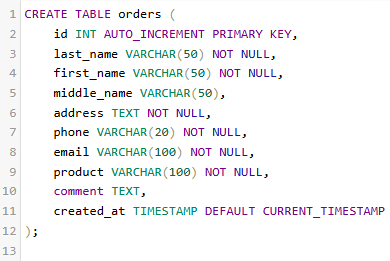


Рисунок 3 – Создание таблицы в базе данных

И занесем полученные данные в созданную таблицу.

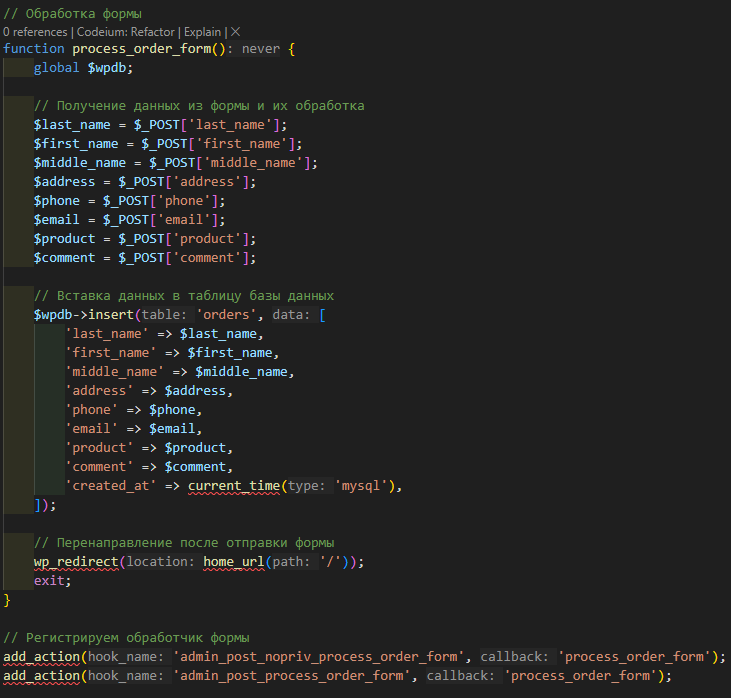


Рисунок 4 – Обработка формы

Результат работы представлен далее

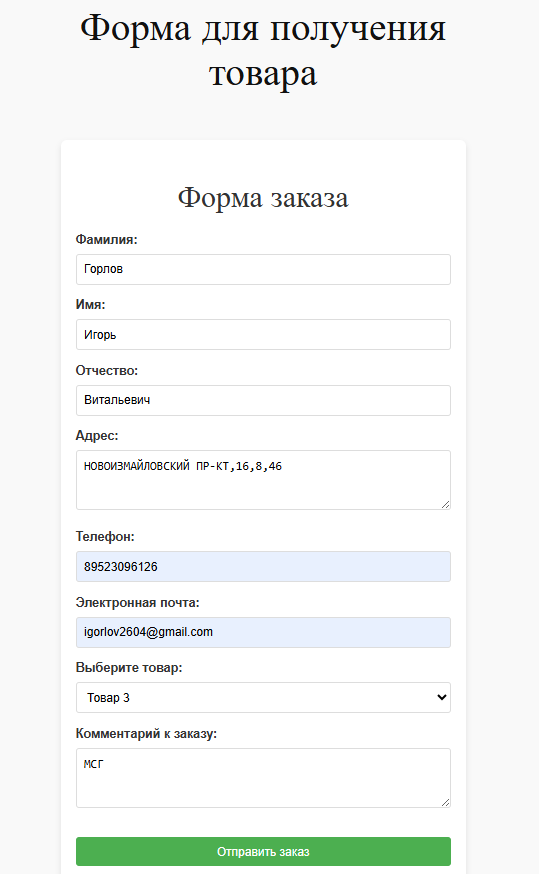


Рисунок 5 – Полученная форма

После отправки формы в таблице появляется новая запись

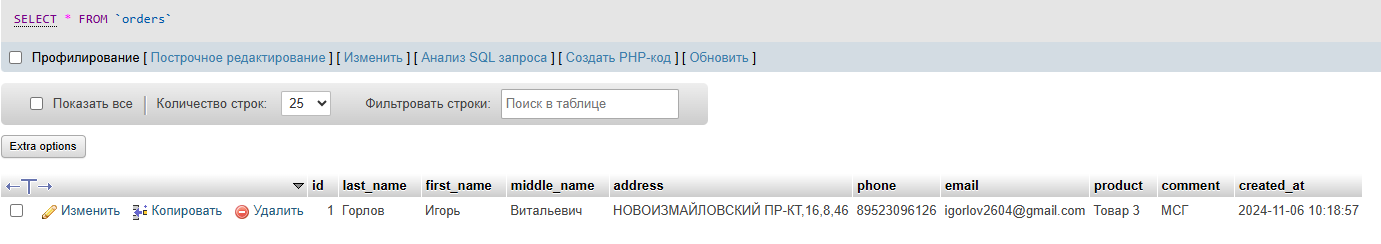


Рисунок 6 – Обновление в базе данных

**Задание 2.**

Для выполнения данного задания создадим новую таблицу в базе данных.

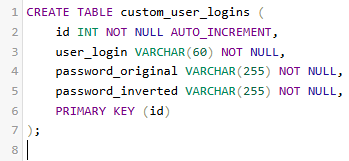


Рисунок 7 – Создание таблицы в базе данных

Необходимо разрешить регистрацию любым пользователям.

Далее создадим функцию, которая при помощи хука ***user\_register*** будет перехватывать регистрацию пользователя. Также создадим функцию для инвертирования битов пароля.

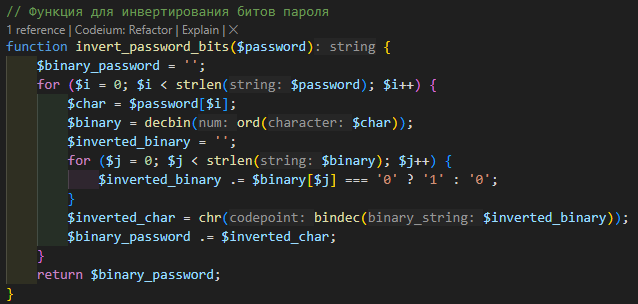


Рисунок 8 – Инвертирование пароля

Так как созданный пользователь сразу получает пароль – получим его с помощью ***get\_userdata().*** Получив из функции данные о логине и пароле, сохраним также инвертированный пароль в базе данных.

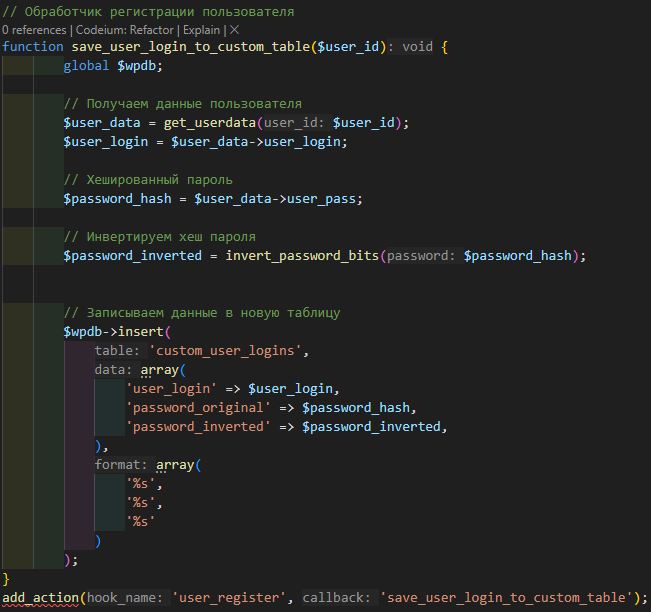


Рисунок 9 – Запуск сервера для отправки форм

Теперь создадим нового пользователя.

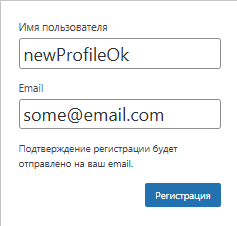


Рисунок 10 – Создание пользователя

После регистрации пользователя в созданной ранее таблице появляются необходимые данные

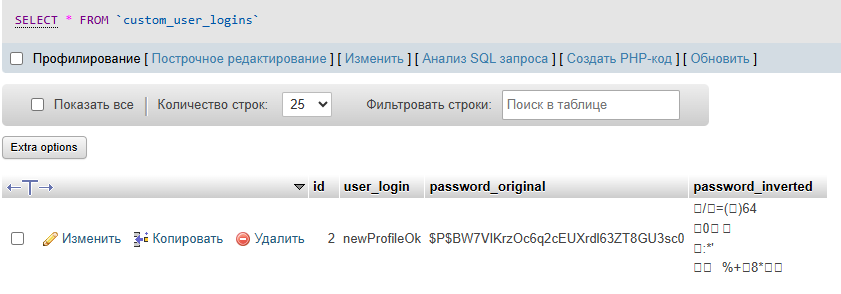


Рисунок 11 – Данные пользователя

**Задание 3.**

В данном задании сервер будет создаваться на NodeJS.

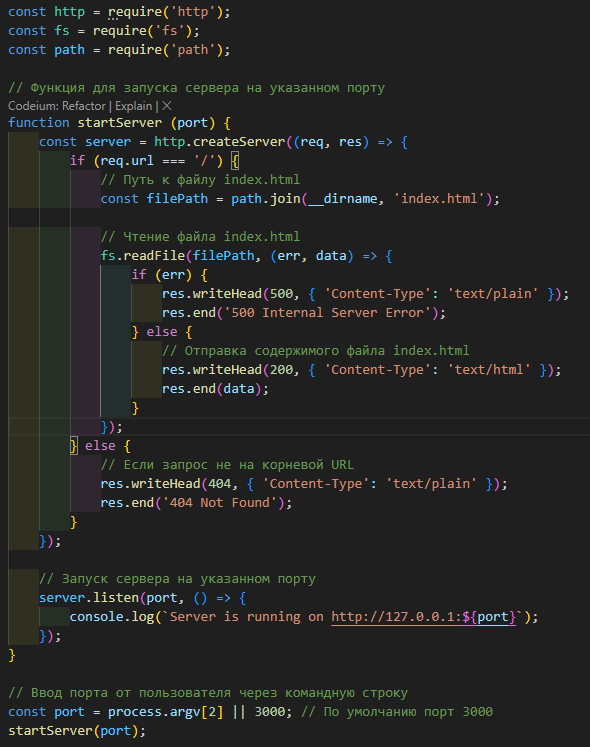


Рисунок 12 – Сервер на NodeJS

Данный код работает следующим образом.

Модули: ***http*** используется для создания сервера, ***fs*** — для чтения файлов, ***path*** — для работы с путями.

Функция ***startServer(port):***

* Создаёт ***HTTP***-сервер, который обрабатывает запросы.
* При запросе на корневой ***URL*** (/), сервер читает файл ***index.html*** и отправляет его содержимое как ответ.
* Если файл не найден или запрос не является корневым, сервер возвращает 404 Not Found.

И запуск скрипта получает в аргументе порт сервера, а если его нет, то запускает сервер на 3000-ом порте.

Файл index.html находится в той же директории и выводит простейший заголовок.



Рисунок 13 – Результат вывода

**Вывод**: В ходе выполнения лабораторной работы были успешно выполнены поставленные задачи. Создана веб-страница, которая позволяет пользователю вводить личные данные и выбирать товары из предложенного списка. Введенные данные обрабатываются с помощью PHP-скрипта и сохраняются в базе данных MySQL, что подтверждает успешное подключение веб-интерфейса к серверной части и базе данных.

Также был модифицирован скрипт авторизации для движка WordPress. Введенные пользователем логин и пароль теперь записываются в дополнительную таблицу MySQL двумя способами: в исходном виде и с инверсией битов пароля. Эта модификация расширяет возможности хранения данных для анализа или экспериментов с шифрованием.

Кроме того, был разработан простой веб-сервер, который поддерживает указание порта для подключения. Сервер корректно возвращает содержимое файла index.html при обращении по адресу и порту, заданным пользователем. Данный опыт подтвердил навыки работы с сетевыми протоколами и настройкой локальных серверов, что полезно для создания серверного ПО на различных языках программирования.

Таким образом, все задачи были выполнены в полном объеме, что свидетельствует о приобретении практических навыков в разработке серверных приложений, работе с базами данных и обработке данных пользователей в веб-приложениях.