УНИВЕРСИТЕТ ИТМО

Факультет «Инфокоммуникационных технологий»

Направление подготовки «Программирование в инфокоммуникационных системах»

Лабораторная работа №3

Выполнил:

Горлов Игорь Витальевич

Группа №K3321

Проверила:

Марченко Елена Владимировна

Санкт-Петербург

2024

**Цель работы**: разработать веб-страницу с использованием gulp и php.

**Задачи**:

* Установка и обучение основам работы с php
* Настройка gulp
* Настроить Wordpress.

Ссылка на удаленный репозиторий:

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

**Ход работы**:

**Задание 1.**

Для выполнения задач последовательно в gulp используется ***series***. Для выполнения задач последовательно, т.е. независимо друг от друга, используется ***parallel***. Для тестирования были написаны 2 функции – concatJs и watchFile, которые будут следить за изменениями и объединять js файлы в единый файл.

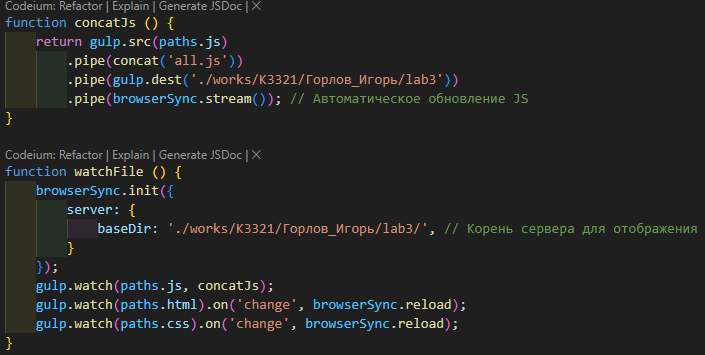


Рисунок 1 – Функции для теста

Код для выполнения представлен далее.

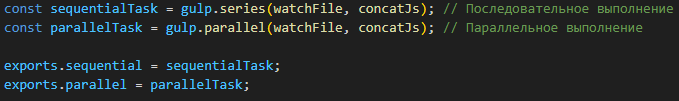


Рисунок 2 – Код выполнения функций

**Задание 2.**

Для выполнения этого задания сначала был установлен php.

В новом gulp-файле напишем запуск сервера php при помощи gulp-connect-php, а также обновление сервера при изменении файлов.

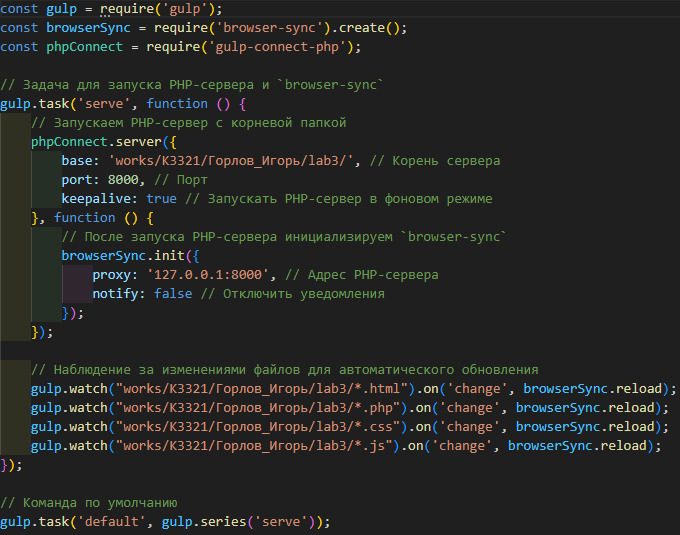


Рисунок 3 – Запуск сервера для отправки форм

Код html-страницы, где находится форма представлен далее.

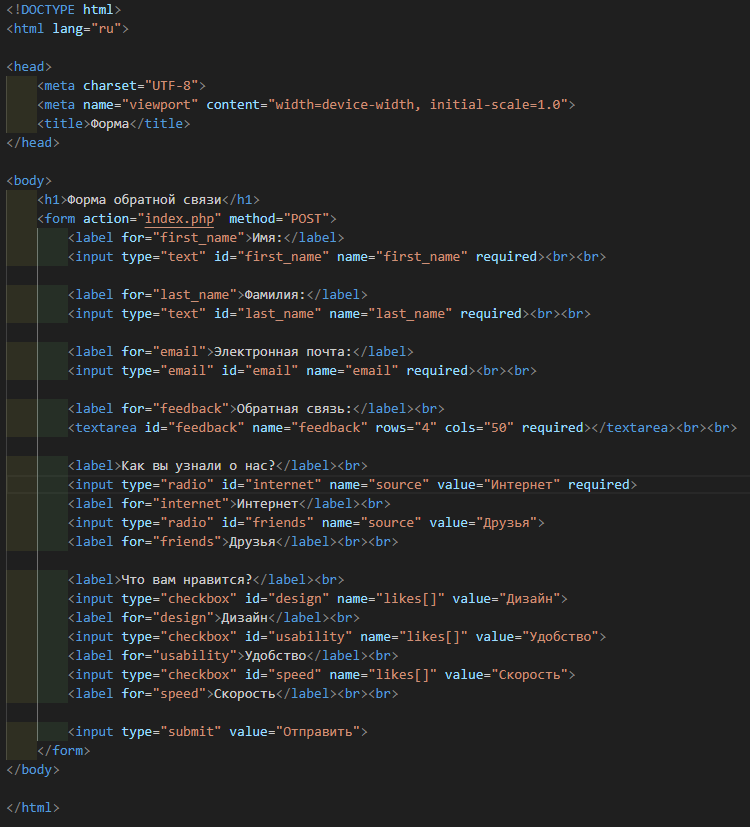


Рисунок 4 – Форма

Данный из этой формы будем посылать в файл index.php, представленный далее.

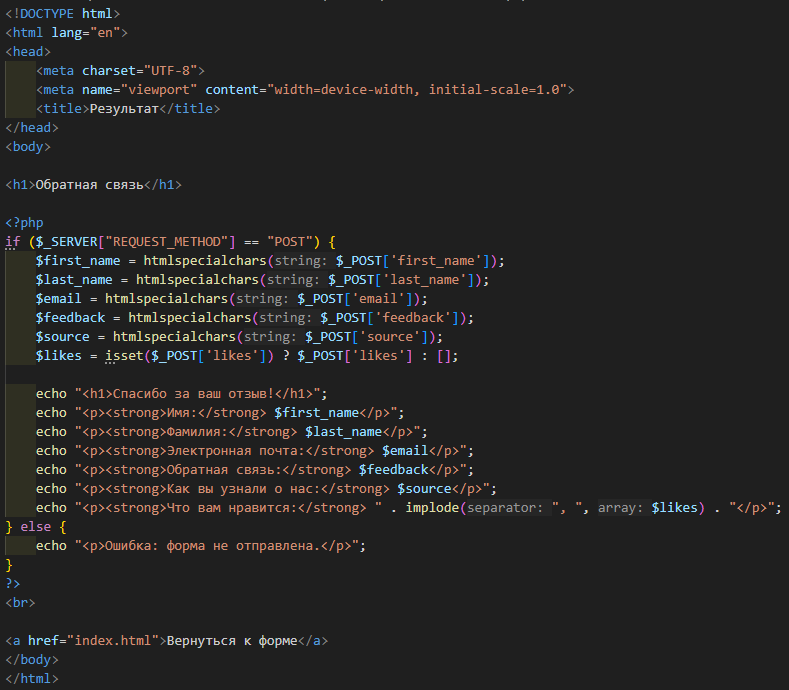


Рисунок 5 – Php-скрипт для просмотра результата

Метод POST был выбран для отправки формы, потому что он более подходящий и безопасный для передачи данных в определённых ситуациях. Вот основные причины, почему для этой формы лучше использовать POST, а не GET:

**Конфиденциальность данных**: POST отправляет данные в теле HTTP-запроса, тогда как GET передаёт их в URL-адресе, делая их видимыми в строке браузера. Если форма собирает личные данные, такие как имя, фамилия, email и обратная связь, POST защищает их от прямого отображения в адресной строке.

**Ограничение на длину данных**: GET имеет ограничение на длину URL (в зависимости от браузера и сервера), поэтому для больших объёмов данных, например, для длинного текста обратной связи, лучше использовать POST, который не имеет такого ограничения.

**Повторяемость**: GET запросы предназначены для получения данных и должны быть безопасными и повторяемыми (не изменяющими данные на сервере). ***POST*** чаще используется, когда запрос изменяет данные или выполняет действие, которое не должно повторяться (например, отправка формы обратной связи).

**Безопасность**: Хотя POST не полностью защищает данные (они могут быть перехвачены, если сайт не использует HTTPS), он делает данные менее доступными по сравнению с ***GET***, где они видны в URL.

Таким образом, метод ***POST*** используется для передачи данных из формы, когда важна конфиденциальность и требуется возможность передавать большие объёмы информации.

В ходе написания PHP-скрипта использовались следующие функции и объекты PHP.

1. ***$\_SERVER["REQUEST\_METHOD"]*** — это суперглобальный массив, который содержит информацию о сервере и запросе.
2. ***$\_POST*** - это суперглобальный массив, содержащий все данные, отправленные из формы методом POST.
3. ***htmlspecialchars()*** - то функция PHP, которая преобразует специальные символы HTML в эквиваленты, чтобы предотвратить интерпретацию этих символов как HTML-код.
4. ***isset()*** - проверяет, установлено ли значение переменной и отличается ли оно от null.
5. ***implode()*** - объединяет элементы массива в строку с определённым разделителем.

Результат представлен далее.

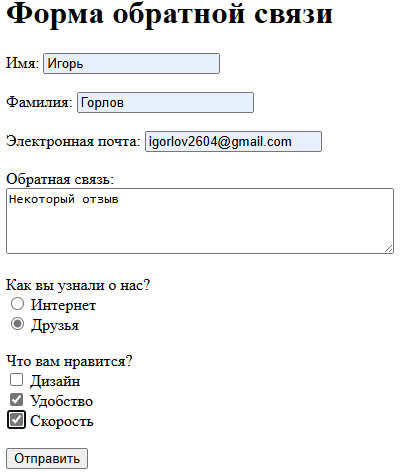


Рисунок 6 – Форма

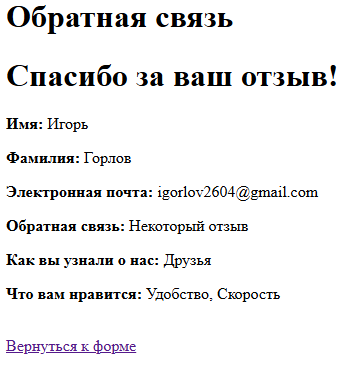


Рисунок 7 – Отправка формы

**Задание 3.**

В данном задании необходимо установить инструментарий для отладки проектов – был выбран openServer. Также необходимо установить и настроить WordPress.

Для начала был установлен openServer с модулями PHP, MySQL, nginx, а также WordPress.

В мае 2024-ого года openServer обновился до 6.0.0. Данная версия сильно отличается от прежней. Установка WordPress производится следующим образом.

1. Скачать архив с официального сайта WordPress
2. Создать папку для проекта в директории home/
3. Перенести в созданную директорию файлы из архива WordPress
4. В созданной директории проекта создать папку .osp
5. В данной папке создать файл project.ini и указать параметры для сервера проекта (в нашем случае указать версию PHP-7.4 и домен test.site
6. Перезапустить OpenServer

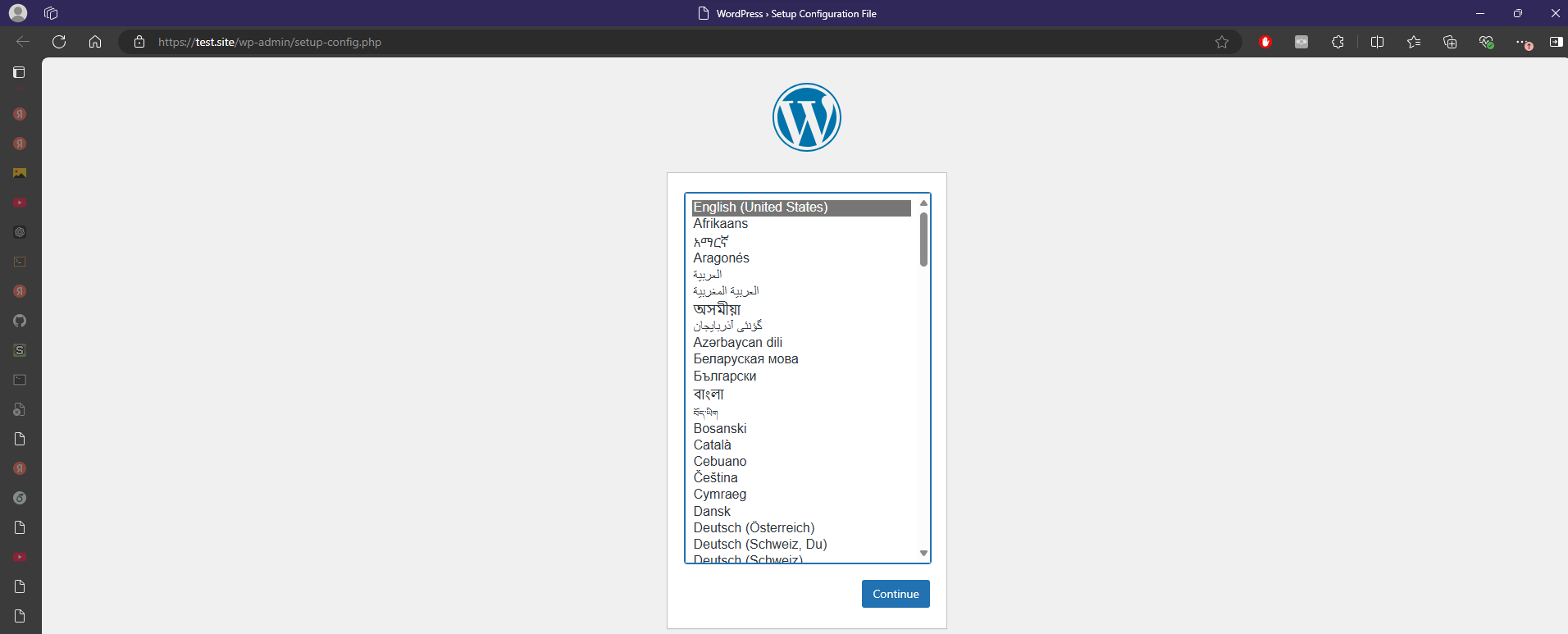
Теперь переходя на домен site.test мы попадаем на страницу установки WordPress. 

Рисунок 8 – Отправка формы

Отдельно от проекта также был установлен phpMyAdmin, база данных которого необходима для работы с WordPress.

После внесения данных базы данных регистрируем пользователя.

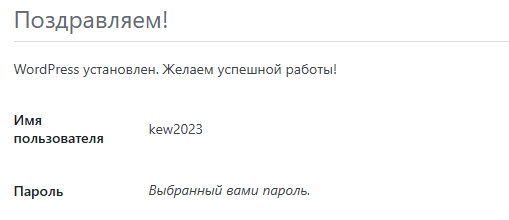


Рисунок 9 – Регистрация пользователя

Теперь после авторизации пользователь попадет на стартовую страницу.

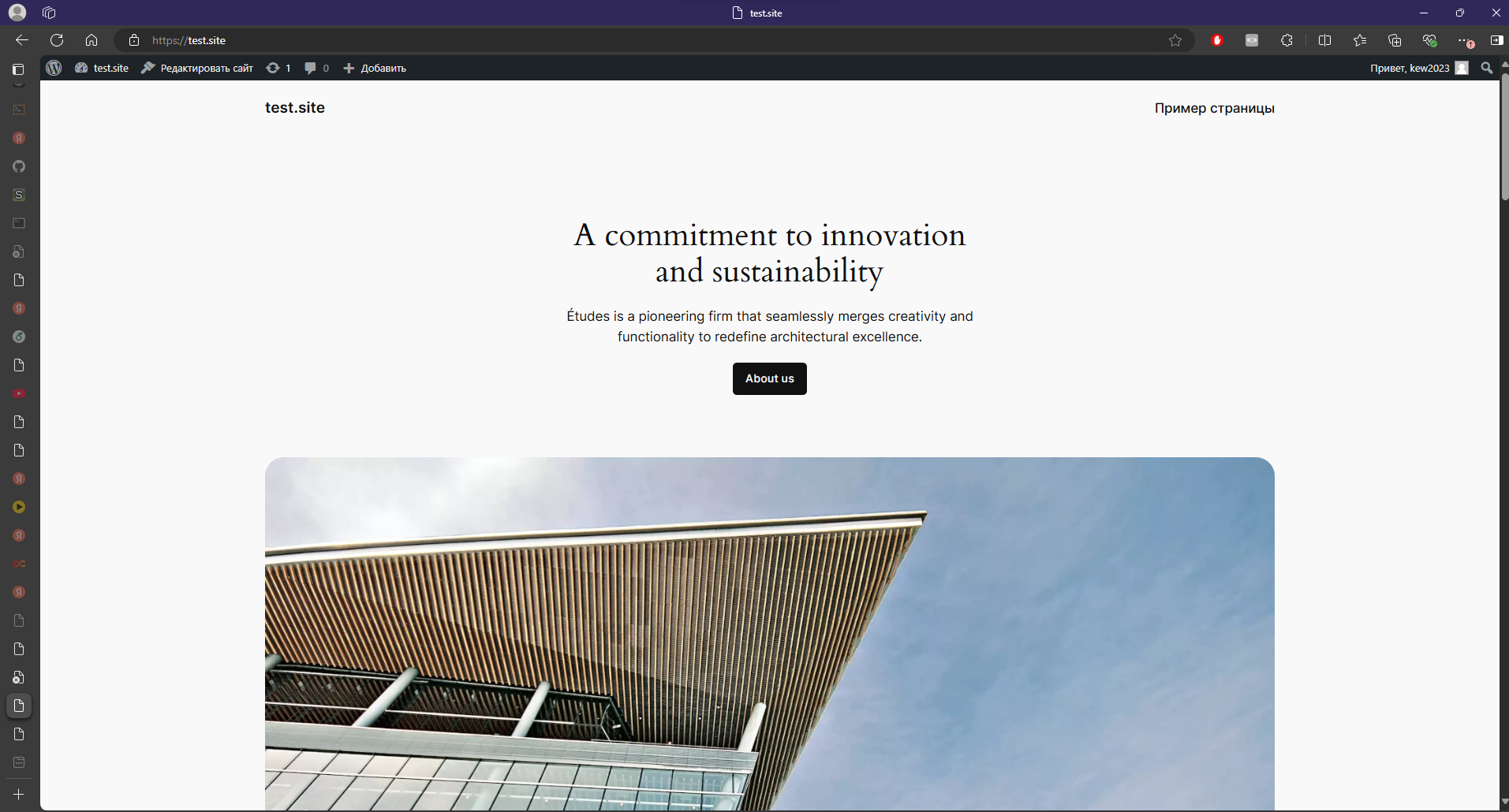


Рисунок 10 – Стартовая страница

**Вывод**: В ходе работы были изучены основы работы с Gulp, написан простой скрипт на PHP, принимающий данные формы, установлен и использован OpenServer, установлен WordPress на пользовательский домен.

Поставленные задачи были выполнены.