

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО
ITMO University
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №3
по дисциплине “Web-программирование”

Выполнил: студент
группы К3323
Влазнев Д. В.
Преподаватель:
Марченко Е. В.

г. Санкт-Петербург,
2024

Введение.....	3
1. Настройка системы автоматизации сборки.....	4
2. Создание формы обратной связи.....	5
3. Настройка WordPress	7
Вывод.....	10

Введение

Цель работы: изучить основы автоматизации сборки и настройки сайта.

Задачи:

1. Настроить систему автоматизации сборки с помощью Gulp.
2. Создать форму обратной связи с пользователем.
3. Настроить локальный сервер и CMS WordPress,

1. Настройка системы автоматизации сборки

Выполним инициализацию папки с помощью команды `npm init --yes`.
Далее установим необходимые пакеты (См. Рисунок 1).

```
PS D:\study\3 course\web\WebDevelopment_2024-2025\works\K3323\Влазнев_Данила_Владимирович\lab_3> npm install --save-dev
gulp gulp-clean-css browser-sync

added 275 packages, and audited 276 packages in 5s

16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS D:\study\3 course\web\WebDevelopment_2024-2025\works\K3323\Влазнев_Данила_Владимирович\lab_3> |
```

Рисунок 1 – Необходимые пакеты

В корне проекта создадим и настроим `gulpfile.js` (См. Рисунок 2–3).

```
const gulp = require('gulp');
const cleanCss = require('gulp-clean-css');
const browserSync = require('browser-sync').create();

function styles() {
  return gulp.src('./src/**/*.css')
    .pipe(cleanCss())
    .pipe(gulp.dest('./dist/css'))
    .pipe(browserSync.stream());
}

function html() {
  return gulp.src('./src/**/*.html')
    .pipe(gulp.dest('./dist/html'))
    .pipe(browserSync.stream());
}

function serve() {
  browserSync.init({
    server: {
      baseDir: './dist'
    }
  });

  gulp.watch('./src/**/*.css', styles);
  gulp.watch('./src/**/*.html', html);
  gulp.watch('./src/**/*.html').on('change', browserSync.reload);
}
```

Рисунок 2 – Gulpfile.js

```
const build = gulp.series(html, styles, php);

const dev = gulp.parallel(serve, build);

gulp.task('build', build);
gulp.task('dev', dev);
```

Рисунок 3 – Gulpfile.js

Теперь запустим таски и посмотрим на результат (См. Рисунок 4–5).

```
gulp serve
PS D:\study\3 course\web\WebDevelopment_2024-2025\works\K3323\Влазнев_Данила_Владимирович\lab_3> gulp build
[16:57:31] Using gulpfile D:\study\3 course\web\WebDevelopment_2024-2025\works\K3323\Влазнев_Данила_Владимирович\lab_3\gulpfile.js
[16:57:31] Starting 'build'...
[16:57:31] Starting 'html'...
[16:57:31] Finished 'html' after 35 ms
[16:57:31] Starting 'styles'...
[16:57:31] Finished 'styles' after 22 ms
[16:57:31] Finished 'build' after 61 ms
PS D:\study\3 course\web\WebDevelopment_2024-2025\works\K3323\Влазнев_Данила_Владимирович\lab_3> gulp serve
[16:57:35] Using gulpfile D:\study\3 course\web\WebDevelopment_2024-2025\works\K3323\Влазнев_Данила_Владимирович\lab_3\gulpfile.js
[16:57:35] Starting 'serve'...
[Browsersync] Access URLs:
    Local: http://localhost:3000
    External: http://169.254.119.253:3000
    UI: http://localhost:3001
    UI External: http://169.254.119.253:3001
[Browsersync] Serving files from: ./dist
```

Рисунок 4 – Результат

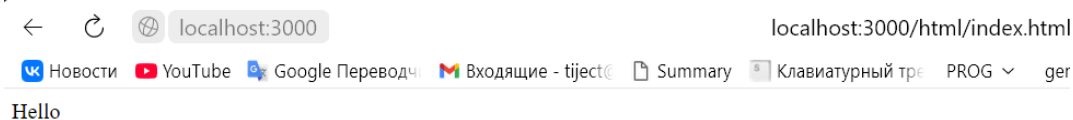


Рисунок 5 – Результат

2. Создание формы обратной связи

Была создана форма обратной связи в соответствии с заданием (См. Рисунок 6–9).

```
<html>
<head>
  <title>Feedback form</title>
  <meta charset="UTF-8">
</head>
<body>
  <h1>Feedback form</h1>
  <form action="/lab3/php/feedback.php" method="post">
    <label for="first_name">First Name:</label>
    <input type="text" id="first_name" name="first_name" required><br><br>

    <label for="last_name">Last Name:</label>
    <input type="text" id="last_name" name="last_name" required><br><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required><br><br>
```

Рисунок 6 – Код формы

```

<label>Where are you from?</label><br>
<input type="radio" id="location-saint-petersburg" name="location" value="saint-petersburg" required>
<label for="location-saint-petersburg">Saint-Petersburg</label><br>

<input type="radio" id="location-moscow" name="location" value="moscow">
<label for="location-moscow">Moscow</label><br><br>

<label>Do you agree with our policies?</label><br>
<input type="checkbox" id="first-policy" name="first-policy" required>
<label for="first-policy">First policy</label><br>

<input type="checkbox" id="second-policy" name="second-policy" required>
<label for="second-policy">Second policy</label><br>

<input type="checkbox" id="third-policy" name="third-policy">
<label for="third-policy">Third policy</label><br><br>

```

Рисунок 7 – Код формы

```

<input type="submit" value="Submit Feedback">
</form>
</body>
</html>

```

Рисунок 8 – Код формы

Feedback form

First Name:

Last Name:

Email:

Where are you from?

☐ Saint-Petersburg

☐ Moscow

Do you agree with our policies?

☐ First policy

☐ Second policy

☐ Third policy

Рисунок 9 – Форма

Также был создан простой php скрипт, который выводит полученную из формы информацию на страницу (См. Рисунок 10–11).

```
<?php
header('Content-Type: text/html; charset=UTF-8');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $first_name = htmlspecialchars($_POST['first_name']);
    $last_name = htmlspecialchars($_POST['last_name']);
    $email = htmlspecialchars($_POST['email']);
    $location = htmlspecialchars($_POST['location']);

    echo "<h2>Thank you for your feedback, $first_name $last_name!</h2>";
    echo "<p>For users from $location a gift is sent to your email: $email.</p>";
} else {
    echo "Invalid request method.";
}
?>
```

Рисунок 10 – PHP код

Thank you for your feedback, Данила Влазнев!

For users from saint-petersburg a gift is sent to your email: vlaznev.04@mail.ru.

Рисунок 11 – Ответ формы

Существует два основных способа отправки формы: с помощью метода GET И POST. Метод GET конвертирует поля формы в URL параметры, так что их можно увидеть в адресной строке, что не всегда является безопасным решением. Метод POST отправляет данные в теле запроса, что не позволяет их увидеть в URL.

3. Настройка WordPress

Для начала необходимо установить локальный сервер. Я использовал XAMPP. (Cross platform, Apache, MariaDB, PHP).

С помощью инструмента PhpMyAdmin была создана база данных для WordPress (См. Рисунок 12).

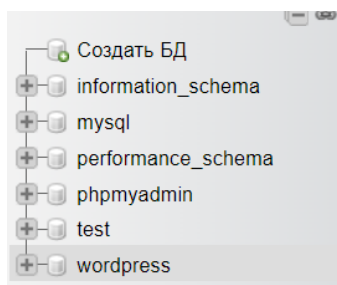


Рисунок 12 – База данных

Был скачан архив WordPress и настроен файл конфигурации, а именно подключение к базе данных (См. Рисунок 13).

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'root' );

/** Database password */
define( 'DB_PASSWORD', '' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
```

Рисунок 13— Конфигурация

Далее запустив сервер и перейдя в нужную папку, был настроен и запущен WordPress (См. Рисунок 14–15).

Требуется информация

Пожалуйста, укажите следующую информацию. Не переживайте, потом вы всегда сможете изменить эти настройки.

Название сайта

Имя пользователя
Имя пользователя может содержать только латинские буквы, пробелы, подчёркивания, дефисы, точки и символ @.

Пароль [Скрыть](#)
Надёжный
Важно: Этот пароль понадобится вам для входа. Сохраните его в надёжном месте.

Ваш e-mail
Внимательно проверьте адрес электронной почты, перед тем как продолжить.

Видимость для поисковых систем ☒ Попросить поисковые системы не индексировать сайт
Будет ли учитываться этот запрос — зависит от поисковых систем.

[Установить WordPress](#)

Рисунок 14— Настройка

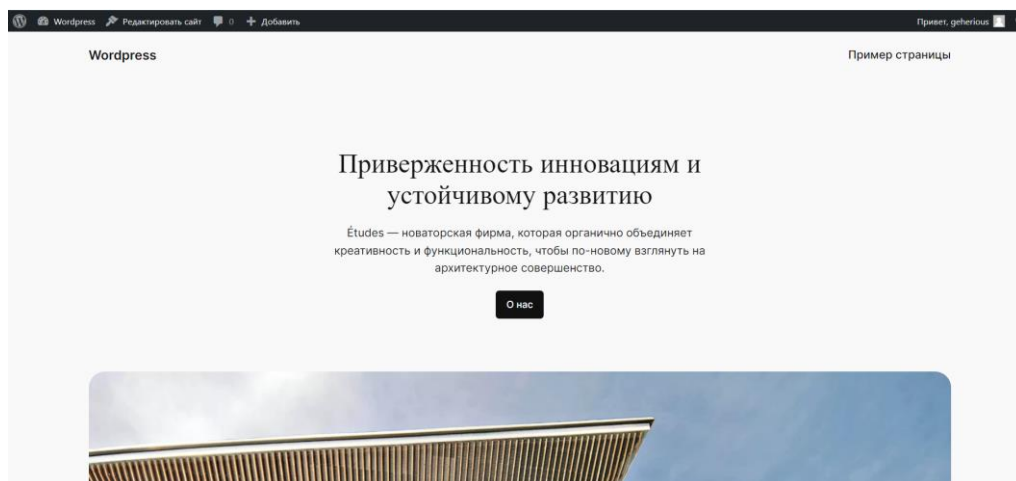


Рисунок 15 — WordPress

Чтобы портал отображался по адресу test.site, необходимо задать виртуальный хост в конфигурации Apache, изменить файл hosts на компьютере, а также изменить настройки WordPress (См. Рисунок 16–18).

```
<VirtualHost *:80>
    DocumentRoot "D:/coding/xamp/htdocs/wordpress"
    ServerName test.site
    <Directory "D:/coding/xamp/htdocs/wordpress">
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Рисунок 16– Настройка

```
127.0.0.1    test.site    # test
```

Рисунок 17 – WordPress

Адрес WordPress (URL)	<input type="text" value="http://test.site"/>
Адрес сайта (URL)	<input type="text" value="http://test.site"/>

Рисунок 18 – WordPress

Тогда сайт отобразится на нужном домене (См. Рисунок 19).

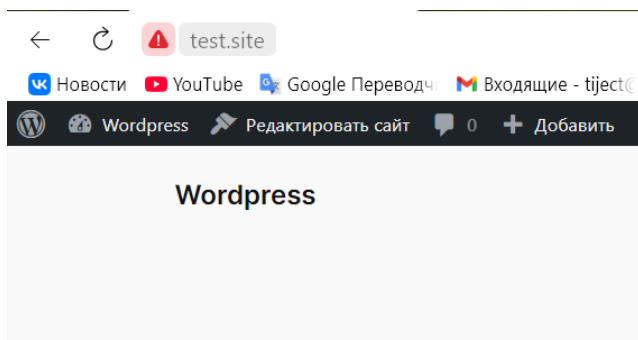


Рисунок 19 – Сайт

Вывод

Я научился создавать последовательные и параллельные задания для gulp, что позволило автоматически обновлять разрабатываемый сайт при изменении кода. Также я создал базовый скрипт на PHP и настроил портал на WordPress.