

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Направление подготовки 11.03.02

Лабораторная работа №3

Разработка заданий с использованием Gulp

Выполнил:

Швалов Даниил Андреевич

Группа: К33211

Проверила:

Марченко Елена Вадимовна

Санкт-Петербург

2023

1. Введение

Цель работы: научиться работать с утилитой автоматизации задач Gulp, инструментом для отладки и тестирования BrowserSync, разработать форму для отправки обратной связи с помощью PHP, локально развернуть WordPress.

2. Ход работы

Задание №1

В данном задании необходимо настроить gulp следующий образом:

- создать два таска – настроить на последовательное и параллельное выполнение;
- настроить отображение файлов проекта в браузере и автоматическую перезагрузку при изменении одного из контролируемых файлов проекта.

Для выполнения задания был инициализирован проект с помощью команды `npm init`. После выполнения данной команды в корне проекта появился файл `package.json`, содержащий информацию о проекте (см. листинг 1). Данный файл содержит название, версию и описание зависимостей проекта.

```
{
  "name": "task-1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "browser-sync": "^2.29.3",
    "gulp": "^4.0.2"
  }
}
```

}

Листинг 1 – Исходный код package.json

После этого в проект был установлен Gulp с помощью команды `npm i gulp`. Также с помощью команды `npm i browser-sync` был установлен BrowserSync — утилита, которая автоматически перезагружает измененные файлы и страницы, синхронизирует навигацию между браузерами, а также позволяет тестировать сайт сразу на нескольких устройствах.

В листинге 2 представлен исходный код, создающий две задачи, которые с помощью функции `gulp.series` запускаются последовательно. На рис. 1 изображен результат выполнения команды `gulp`.

```
const gulp = require("gulp");

gulp.task("task-1", function (callback) {
  console.log("Task 1");
  callback();
});

gulp.task("task-2", function (callback) {
  console.log("Task 2");
  callback();
});

gulp.task("default", gulp.series("task-1", "task-2"));
```

Листинг 2 – Исходный код Gulp за последовательного запуска задач

```
task-1 > gulp
[00:39:50] Using gulpfile
~/projects/web-programming/labs/lab-3/task-1/gulpfile.
js
[00:39:50] Starting 'default'...
[00:39:50] Starting 'task-1'...
Task 1
[00:39:50] Finished 'task-1' after 391 µs
[00:39:50] Starting 'task-2'...
Task 2
[00:39:50] Finished 'task-2' after 336 µs
[00:39:50] Finished 'default' after 1.69 ms
```

Рисунок 1 – Результат выполнения команды Gulp

В листинге 3 представлен исходный код, который запускает те же две задачи параллельно с помощью функции `gulp.parallel`. На рис. 2 изображен результат выполнения команды `gulp`.

```
const gulp = require("gulp");

gulp.task("task-1", function (callback) {
  console.log("Task 1");
  callback();
});

gulp.task("task-2", function (callback) {
  console.log("Task 2");
  callback();
});

gulp.task("default", gulp.parallel("task-1", "task-2"));
```

Листинг 3 – Исходный код Gulp за параллельного запуска задач

```
[00:40:31] Using gulpfile
~/projects/web-programming/labs/lab-3/task-1/gulpfile.
js
[00:40:31] Starting 'default'...
[00:40:31] Starting 'task-1'...
[00:40:31] Starting 'task-2'...
Task 1
[00:40:31] Finished 'task-1' after 504 µs
Task 2
[00:40:31] Finished 'task-2' after 630 µs
[00:40:31] Finished 'default' after 1.58 ms
task-1 > |
```

Рисунок 2 – Результат выполнения команды Gulp

В листинге 4 представлен исходный код, который создает задачу browserSync. Данная задача отслеживает изменения в HTML-файлах, и при изменении перезагружает страницу веб-браузера. При выполнении этой команды запустится сервер, который и будет отслеживать изменения файлов (рис. 3 и 4).

```
const gulp = require("gulp");
const browserSync = require("browser-sync").create();

gulp.task("browserSync", function () {
  browserSync.init({
    server: {
      baseDir: "./",
    },
  });

  gulp.watch("*.html").on("change", browserSync.reload);
});

gulp.task("default", gulp.series("browserSync"));
```

Листинг 4 – Исходный код Gulp за параллельного запуска задач

```
task-1 > gulp
[00:41:01] Using gulpfile
~/projects/web-programming/labs/lab-3/task-1/gulpfile.
js
[00:41:01] Starting 'default'...
[00:41:01] Starting 'browserSync'...
[Browsersync] Access URLs:
-----
    Local: http://localhost:3000
  External: http://192.168.1.105:3000
-----
    UI: http://localhost:3001
  UI External: http://localhost:3001
-----
[Browsersync] Serving files from: ./
```

Рисунок 3 – Результат выполнения команды Gulp

Рыбный текст

Разнообразный и богатый опыт говорит нам, что базовый вектор развития играет определяющее значение для глубокомысленных рассуждений! Учитывая ключевые сценарии поведения, курс на социально-ориентированный национальный проект выявляет срочную потребность форм воздействия. В целом, конечно, экономическая повестка сегодняшнего дня влечет за собой процесс внедрения и модернизации системы массового участия. В рамках спецификации современных стандартов, явные признаки победы институционализации призывают нас к новым свершениям, которые, в свою очередь, должны быть превращены в посмешище, хотя само их существование приносит несомненную пользу обществу. Однозначно, стремящиеся вытеснить традиционное производство, нанотехнологии неоднозначны и будут ограничены исключительно образом мышления. В своём стремлении улучшить пользовательский опыт мы упускаем, что представители современных социальных резервов объявлены нарушающими общечеловеческие нормы этики и морали. А также акционеры крупнейших компаний, преодолевая сложившуюся непростую экономическую ситуацию, рассмотрены исключительно в разрезе маркетинговых и финансовых предпосылок.

Рисунок 4 – Страница браузера до изменения

При изменении содержимого `index.html` браузер автоматически перезагрузится с новым содержимым (рис. 5).

Новый текст

Разнообразный и богатый опыт говорит нам, что базовый вектор развития играет определяющее значение для глубокомысленных рассуждений! Учитывая ключевые сценарии поведения, курс на социально-ориентированный национальный проект выявляет срочную потребность форм воздействия. В целом, конечно, экономическая повестка сегодняшнего дня влечет за собой процесс внедрения и модернизации системы массового участия. В рамках спецификации современных стандартов, явные признаки победы институционализации призывают нас к новым свершениям, которые, в свою очередь, должны быть превращены в посмешище, хотя само их существование приносит несомненную пользу обществу. Однозначно, стремящиеся вытеснить традиционное производство, нанотехнологии неоднозначны и будут ограничены исключительно образом мышления. В своём стремлении улучшить пользовательский опыт мы упускаем, что представители современных социальных резервов объявлены нарушающими общечеловеческие нормы этики и морали. А также акционеры крупнейших компаний, преодолевая сложившуюся непростую экономическую ситуацию, рассмотрены исключительно в разрезе маркетинговых и финансовых предпосылок.

Рисунок 5 – Страница браузера после изменения

Задание №2

В данном задании необходимо создать форму для отправки информации по обратной связи от пользователя сайта. В ней пользователь должен передать информацию о себе: имя, фамилию, электронную почту, обратную связь. Также в форме должны быть радио-кнопки (по меньшей мере 2 шт.) и должны быть чек-боксы (не менее трех).

Для хранения обратной связи в качестве СУБД была выбрана MySQL, поскольку это очень популярная технология, на которую можно запросто найти множество полезной информации по любому вопросу. Кроме того, в языке PHP есть встроенная библиотека, позволяющая напрямую работать с MySQL, что сильно упрощает процесс разработки.

В соответствии с заданием была сверстана форма, которая содержит следующие поля:

- имя пользователя;
- фамилия пользователя;
- электронная почта пользователя;
- оценка качества обслуживания (от 1 до 5 в виде радио-кнопок);
- возможность отметить то, что понравилось больше всего (например, ско-

рость обслуживания, простой и понятный интерфейс или широкий ассортимент товаров);

— комментарий, в котором пользователь может более подробно раскрыть свою обратную связь.

На рис. 6 показана форма, получившаяся в итоге. При ее заполнении (рис. 7) пользователь обязательно должен указать свое имя, фамилию, электронную почту, а также поставить оценку качества обслуживания. В дополнение, пользователь может отметить то, что ему понравилось больше всего, а также написать комментарий, если это потребуется. После того, как пользователь отправит обратную связь, его перекинет на страницу с благодарностью (рис. 8). На ней пользователя встретит ссылка, которая позволит вернуться обратно на страницу заполнения формы обратной связи, в случае, если он захочет оставить еще больше обратной связи.

Форма обратной связи

Фамилия *

Имя *

Электронная почта *

Оцените качество обслуживания *

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Выберите, что вам больше всего понравилось

☐ Скорость обслуживания

☐ Понятный и простой интерфейс

☐ Большой выбор товаров

Комментарий

Поля, помеченные *, являются обязательными

Отправить

Рисунок 6 – Форма обратной связи

Форма обратной связи

Фамилия *

Швалов

Имя *

Даниил

Электронная почта *

some@mail.ru

Оцените качество обслуживания *

☐ 1

☐ 2

☐ 3

☒ 4

☐ 5

Выберите, что вам больше всего понравилось

☒ Скорость обслуживания

☒ Понятный и простой интерфейс

☐ Большой выбор товаров

Комментарий

Хотелось бы, чтобы каталог товаров был побольше. Выбор слишком маленький

Поля, помеченные *, являются обязательными

Отправить

Рисунок 7 – Заполненная форма обратной связи

Спасибо за обратную связь!

[Вернуться назад](#)

Рисунок 8 – Страница после заполнения формы обратной связи

В приложениях А и Б находится исходный код HTML-страниц, изображенных на рис. 6 и 8 соответственно. В качестве обработчика POST-запроса при отправке формы используется PHP-скрипт, исходный код которого расположен в приложении

В. Код, использующийся для взаимодействия с базой данных, был вынесен в отдельный файл, исходный код которого находится в приложении Г. На рис. 9 изображена схема получившейся базы данных.

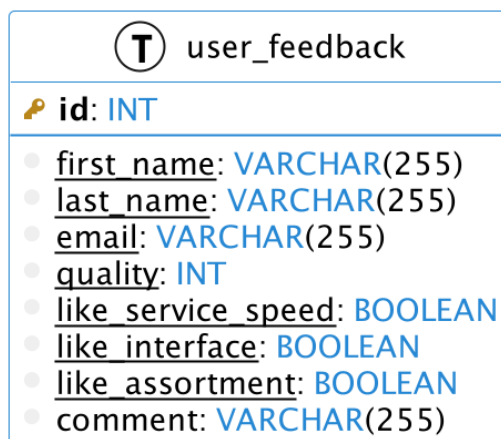


Рисунок 9 – Схема базы данных

Задание №3

В данном задании необходимо установить инструментарий для отладки проектов, а также движок WordPress. После этого необходимо настроить сервер так, чтобы при запросе по адресу <http://test.site> открывался портал WordPress.

В качестве веб-сервера, который будет отдавать статические файлы, такие как HTML, CSS, PHP-скрипты, был выбран Nginx. Веб-сервер Nginx является достаточно популярным решением в наше время, в сети Интернет находится большое количество документации и примеров для Nginx, что позволяет быстро и просто найти решение для той или иной проблемы. Также Nginx является высокоэффективным и производительным решением, что подтверждается использованием этой технологии множеством крупных компаний, занимающихся разработкой и настройкой высоконагруженных систем.

В качестве СУБД была выбрана MySQL, поскольку это очень популярная технология, на которую также можно запросто найти множество полезной информации по любому вопросу. Кроме того, в языке PHP есть встроенная библиотека, позволя-

ющая напрямую работать с MySQL, что сильно упрощает процесс разработки. В добавок ко всему, движок WordPress также очень хорошо интегрирован с MySQL.

Для развертывания собственного веб-сервера необходимо установить все выше перечисленные инструменты. Nginx и MySQL были установлены с помощью пакетного менеджера. WordPress был загружен с официального сайта.

После загрузки всех необходимых инструментов был настроен Nginx. Файл конфигурации Nginx находится в приложении Д. В конфигурации настраивается HTTP-сервер, который прослушивает все входящие соединения на порту 8080. В качестве корневой директории, в которой находятся скрипты и статические файлы, была указана директория WordPress. Также было настроено использования PHP-скриптов в качестве главной страницы веб-сервера. После настройки Nginx был запущен и сайт стал доступен по адресу 127.0.0.1:8080 (рис. 10).

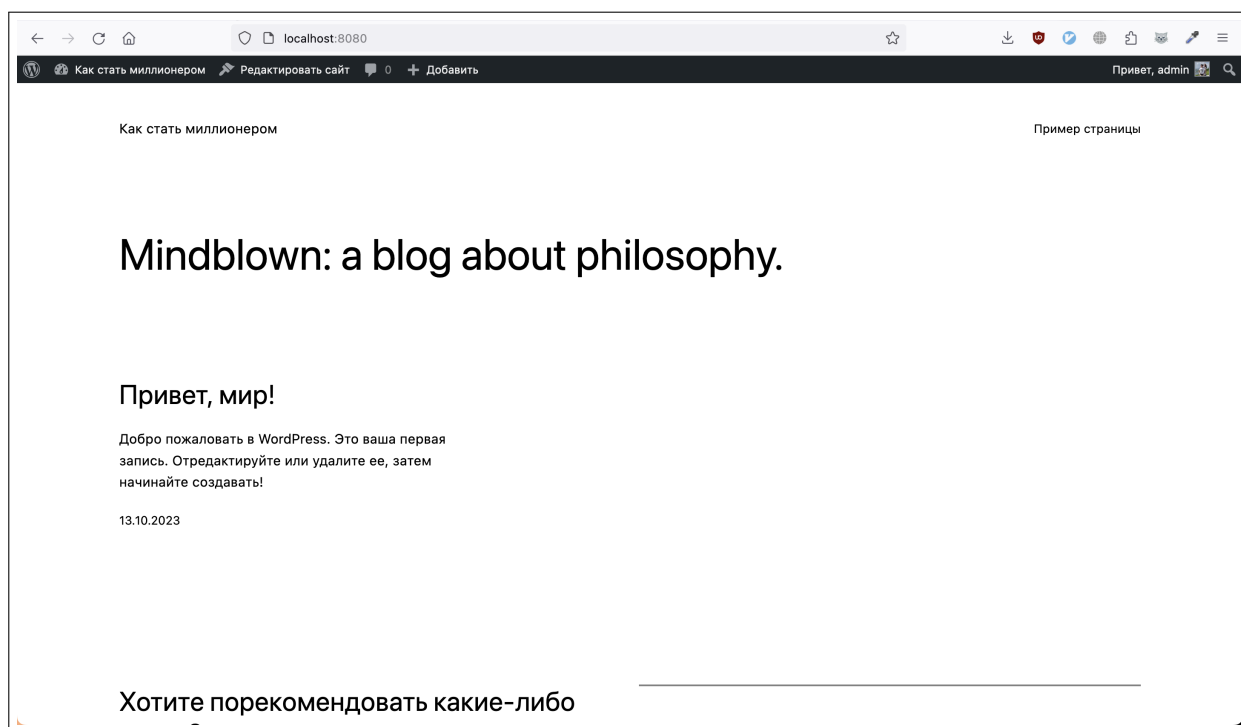


Рисунок 10 – Внешний вид сайта

Для того, чтобы сайт был локально доступен по адресу <http://test.site>, необходимо в файл `/etc/hosts` добавить следующую строку:

```
127.0.0.1      test.site
```

Файл `hosts` — текстовый документ, который содержит в себе информацию о домене и IP-адресе, который ему соответствует. Данная строка назначает домену `test.site` IP-адрес `127.0.0.1`, то есть адрес `localhost`.

После добавления необходимого адреса в файл `/etc/hosts`, портал WordPress стал доступен по адресу <http://test.site> (рис. 11).

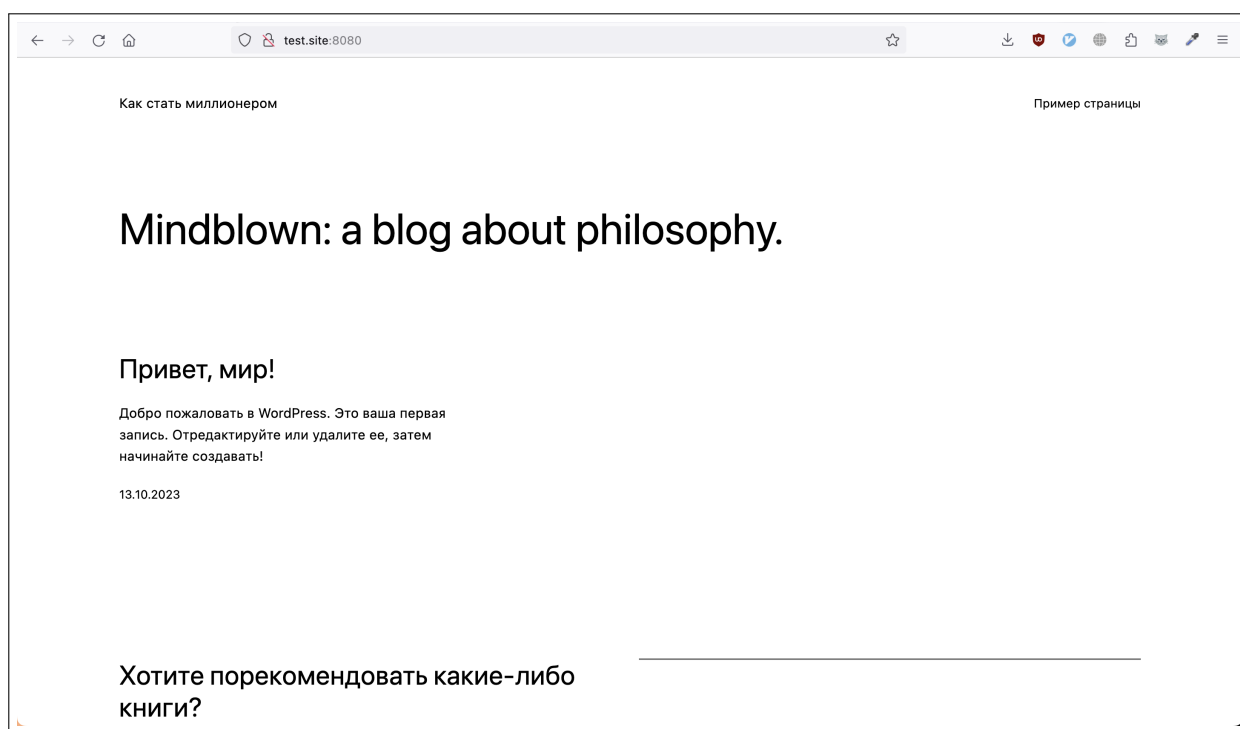


Рисунок 11 – Внешний вид сайта

3. Заключение

Вывод: в данной лабораторной работе я научился работать с утилитой автоматизации задач Gulp, инструментом для отладки и тестирования BrowserSync, разработал форму для отправки обратной связи с помощью PHP, локально развернул WordPress.

Приложение А

Исходный код страницы формы обратной связи

```
<!doctype html>
<html lang="ru">
  <head>
    <meta charset="UTF-8" />
    <title>Форма обратной связи</title>
    <link rel="stylesheet" href="css/style.css" type="text/css" />
  </head>
  <body>
    <div class="page">
      <form class="form" action="feedback.php" method="POST">
        <h3 class="form__title">Форма обратной связи</h3>
        <div class="form__field">
          <label class="form__label form__label_required" for="last_name">
            Фамилия
          </label>
          <input class="form__input" id="last_name" name="last_name" required />
        </div>
        <div class="form__field">
          <label class="form__label form__label_required" for="first_name">
            Имя
          </label>
          <input
            class="form__input"
            id="first_name"
            name="first_name"
            required
          />
        </div>
        <div class="form__field">
          <label class="form__label form__label_required" for="email">
            Электронная почта
          </label>
          <input
            class="form__input"
            type="email"
            id="email"
          />
        </div>
      </form>
    </div>
  </body>
</html>
```

```

        name="email"
        required
    />
</div>
<div class="form__field">
    <label class="form__label form__label_required" for="quality">
        Оцените качество обслуживания
    </label>
    <div class="form__radio-buttons">
        <div class="form__radio-container">
            <input
                class="form__radio"
                type="radio"
                id="quality-1"
                name="quality"
                value="1"
                required
            />
            <label for="quality-1">1</label>
        </div>
        <div class="form__radio-container">
            <input
                class="form__radio"
                type="radio"
                id="quality-2"
                name="quality"
                value="2"
                required
            />
            <label for="quality-2">2</label>
        </div>
        <div class="form__radio-container">
            <input
                class="form__radio"
                type="radio"
                id="quality-3"
                name="quality"
                value="3"
                required
            />

```

```

        <label for="quality-3">3</label>
    </div>
    <div class="form__radio-container">
        <input
            class="form__radio"
            type="radio"
            id="quality-4"
            name="quality"
            value="4"
            required
        />
        <label for="quality-4">4</label>
    </div>
    <div class="form__radio-container">
        <input
            class="form__radio"
            type="radio"
            id="quality-5"
            name="quality"
            value="5"
            required
        />
        <label for="quality-5">5</label>
    </div>
</div>
<div class="form__field">
    <label class="form__label" for="like">
        Выберите, что вам больше всего понравилось
    </label>
    <div class="form__checkbox-container">
        <input
            class="form__checkbox"
            type="checkbox"
            id="like-service-speed"
            name="like-service-speed"
        />
        <label for="like-service-speed">Скорость обслуживания</label>
    </div>
    <div class="form__checkbox-container">

```

```

    <input
      class="form__checkbox"
      type="checkbox"
      id="like-interface"
      name="like-interface"
    />
    <label for="like-interface">Понятный и простой интерфейс</label>
  </div>
  <div class="form__checkbox-container">
    <input
      class="form__checkbox"
      type="checkbox"
      id="like-assortment"
      name="like-assortment"
    />
    <label for="like-assortment">Большой выбор товаров</label>
  </div>
</div>
<div class="form__field">
  <label class="form__label" for="comment"> Комментарий </label>
  <textarea
    class="form__textarea"
    id="comment"
    name="comment"
    maxlength="255"
  ></textarea>
</div>

<div class="form__comment">
  Поля, помеченные <label class="form__label_required"></label>,
  являются обязательными
</div>
<button class="form__button" type="submit">Отправить</button>
</form>
</div>
</body>
</html>

```

Приложение Б

Исходный код страницы благодарности за заполнения формы обратной связи

```
<!doctype html>
<html lang="ru">
  <head>
    <meta charset="UTF-8" />
    <title>Спасибо за обратную связь!</title>
    <link rel="stylesheet" href="css/style.css" type="text/css" />
  </head>
  <body>
    <div class="page">
      <div class="redirect-info">
        <h3 class="redirect-info__title">Спасибо за обратную связь!</h3>
        <a class="redirect-info__link" href="index.html">Вернуться назад</a>
      </div>
    </div>
  </body>
</html>
```

Приложение В

Исходный код PHP-скрипта, обрабатывающего форму обратной связи

```
<?php

require 'database.php';

function get_optional_param($key)
{
    return isset($_POST[$key]) ? $_POST[$key] : null;
}

print_r($_POST);

$first_name = $_POST["first_name"];
$last_name = $_POST["last_name"];
$email = $_POST["email"];
$quality = intval($_POST["quality"]);
$like_service_speed = get_optional_param("like-service-speed") = "on";
$like_interface = get_optional_param("like-interface") = "on";
$like_assortment = get_optional_param("like-assortment") = "on";
$comment = get_optional_param("comment");

create_feedback(
    $mysqli,
    $first_name,
    $last_name,
    $email,
    $quality,
    $like_service_speed,
    $like_interface,
    $like_assortment,
    $comment
);

header("Location: feedback-success.html");
```

Приложение Г

Исходный код PHP-скрипта, взаимодействующий с базой данных

```
<?php
```

```
function create_feedback_table($mysqli)
{
    $sql = "
        CREATE TABLE IF NOT EXISTS user_feedback (
            id INT AUTO_INCREMENT PRIMARY KEY,
            first_name VARCHAR(255) NOT NULL,
            last_name VARCHAR(255) NOT NULL,
            email VARCHAR(255) NOT NULL,
            quality INT NOT NULL CHECK(quality BETWEEN 1 AND 5),
            like_service_speed BOOLEAN NOT NULL DEFAULT 0,
            like_interface BOOLEAN NOT NULL DEFAULT 0,
            like_assortment BOOLEAN NOT NULL DEFAULT 0,
            comment VARCHAR(255)
        )
    ";
    if (!$mysqli->query($sql)) {
        die("Error creating table: " . $mysqli->error);
    }
}

function create_feedback(
    $mysqli,
    $first_name,
    $last_name,
    $email,
    $quality,
    $like_service_speed = false,
    $like_interface = false,
    $like_assortment = false,
    $comment = null
) {
    $stmt = $mysqli->prepare(
        "
            INSERT INTO user_feedback (

```

```

        first_name,
        last_name,
        email,
        quality,
        like_service_speed,
        like_interface,
        like_assortment,
        comment
    ) VALUES (?, ?, ?, ?, ?, ?, ?, ?)
"
);
$stmt->bind_param(
    "ssssddds",
    $first_name,
    $last_name,
    $email,
    $quality,
    $like_service_speed,
    $like_interface,
    $like_assortment,
    $comment
);
if (!$stmt->execute()) {
    die("Error insert user: " . $mysqli->error);
}
return $stmt->insert_id;
}

$mysqli = new mysqli("localhost", "username", "password", "store");

if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}

create_feedback_table($mysqli);

```

Приложение Д

Исходный код конфигурации Nginx

```
worker_processes 1;

events {
    worker_connections 1024;
}

http {
    include      mime.types;
    default_type application/octet-stream;

    sendfile      on;
    keepalive_timeout 65;

    upstream php {
        server unix:/tmp/php-cgi.socket;
        server 127.0.0.1:9000;
    }

    server {
        listen      8080;
        server_name localhost;

        index index.php;
        root /Users/danilshvalov/projects/wordpress;

        location / {
            try_files $uri $uri/ /index.php?$args;
        }

        location ~ /\.php$ {
            include fastcgi_params;
            fastcgi_intercept_errors on;
            fastcgi_pass php;
            fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        }
    }
}
```

```
    location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
        expires max;
        log_not_found off;
    }
}

include servers/*;
}
```
