|  |
| --- |
| Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Новгородский государственный университет имени  Ярослава Мудрого»  ИНСТИТУТ ЭЛЕКТРОННЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ  ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ |

|  |
| --- |
| Утверждаю: |
| Зам. директора по УМ и ВР |

ПРОИЗВОДСТВЕННАЯ ПРАКТИКА

ПМ.05 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ

09.02.07 Информационные системы и программирование

код и наименование специальности

ОТЧЁТ

Руководитель от организации:

« » 202 г.

Руководитель от колледжа:

« » 202 г.

Выполнил:

Студент гр.

« » 202 г.

# Содержание

[Раздел 1. Общая (теоретическая) часть 5](#_Toc170640850)

[1.1 Постановка задачи (ТЗ) 5](#_Toc170640851)

[1.2 Обоснование проектных решений 7](#_Toc170640852)

[1.3 Обзор существующих программных систем 9](#_Toc170640857)

[Раздел 2. Практическая составляющая 10](#_Toc170640858)

[2.1 Реализация 10](#_Toc170640859)

[2.2 Тестирование 11](#_Toc170640863)

[Заключение 20](#_Toc170640864)

[Список литературы 20](#_Toc170640865)

[Приложение А 20](#_Toc170640866)

[Приложение Б 20](#_Toc170640867)

[Приложение В 24](#_Toc170640868)

# Аннотация

Этот проект направлен на разработку комплексного веб-приложения для агентства недвижимости. Приложение позволит пользователям просматривать доступную к покупке или аренде недвижимость.

Введение

В современном цифровом мире наличие удобного и функционального веб-сайта является важным инструментом для успеха любого бизнеса. Агентство недвижимости, работающее в условиях высокой конкуренции, не является исключением. Интернет-присутствие позволяет значительно расширить клиентскую базу, предлагая пользователям удобный способ поиска и ознакомления с доступными объектами недвижимости.

Цель данного проекта заключалась в создании веб-сайта для агентства недвижимости, который предоставляет пользователям возможность легко и быстро находить информацию о доступной к покупке недвижимости. В ходе работы над проектом мы стремились обеспечить интуитивно понятный интерфейс, высокую скорость загрузки страниц и адаптивный дизайн для различных устройств.

Создание веб-сайта направлено на решение нескольких ключевых задач:

* Обеспечение удобного и быстрого доступа к информации о недвижимости;
* Обеспечение интуитивно понятного интерфейса для пользователей;
* Гарантия высокой скорости загрузки страниц и адаптивного дизайна для различных устройств.

Основные функциональные возможности сайта:

* Поиск недвижимости по типу.
* Подробные описания объектов с фотографиями и характеристиками.

В отчете подробно описаны этапы разработки веб-сайта: от анализа требований до тестирования и запуска проекта. Также рассмотрены используемые технологии, инструменты и методики, обеспечившие достижение поставленных целей.

# Постановка задачи на практику

Целью практики является получение профессиональных навыков и опыта в работе

# Раздел 1 Техника безопасности

## Общие требования безопасности

1.1 К работе на персональном компьютере допускаются лица, прошедшие обучение безопасным методам труда, вводный инструктаж, первичный инструктаж на рабочем месте.

1.2 При эксплуатации персонального компьютера на работника могут оказывать действие следующие опасные и вредные производственные факторы:

- повышенный уровень электромагнитных излучений;

- повышенный уровень статического электричества;

- пониженная ионизация воздуха;

- статические физические перегрузки;

- перенапряжение зрительных анализаторов.

1.3 Работник обязан:

1.3.1 Выполнять только ту работу, которая определена его должностной инструкцией.

1.3.2 Содержать в чистоте рабочее место.

1.3.3 Соблюдать режим труда и отдыха в зависимости от продолжительности, вида и категории трудовой деятельности.

1.3.3 Соблюдать меры пожарной безопасности.

1.4 Рабочие места с компьютерами должны размещаться таким образом, чтобы расстояние от экрана одного видеомонитора до тыла другого было не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

1.5 Рабочие места с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, преимущественно слева.

1.6 Оконные проемы в помещениях, где используются персональные компьютеры, должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.

1.7 Рабочая мебель для пользователей компьютерной техникой должна отвечать следующим требованиям:

- высота рабочей поверхности стола должна регулироваться в пределах 680 - 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм;

- рабочий стол должен иметь пространство для ног высотой не менее 600 мм, глубиной на уровне колен не менее 450 мм и на уровне вытянутых ног не менее 650 мм;

- рабочий стул (кресло) должен быть подъемно - поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также - расстоянию спинки от переднего края сиденья;

- рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов; поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм;

- рабочее место с персональным компьютером должно быть оснащено легко перемещаемым пюпитром для документов.

1.8 Для нормализации аэроионного фактора помещений с компьютерами необходимо использовать устройства автоматического регулирования ионного режима воздушной среды (например, аэроионизатор стабилизирующий "Москва-СА1").

1.9 Женщины со времени установления беременности и в период кормления грудью к выполнению всех видов работ, связанных с использованием компьютеров, не допускаются.

1.10 За невыполнение данной Инструкции виновные привлекаются к ответственности согласно правилам внутреннего трудового распорядка или взысканиям, определенным Кодексом законов о труде Российской Федерации.

## 2 Требования безопасности перед началом работы

2.1 Подготовить рабочее место.

2.2 Отрегулировать освещение на рабочем месте, убедиться в отсутствии бликов на экране.

2.3 Проверить правильность подключения оборудования к электросети.

2.4 Проверить исправность проводов питания и отсутствие оголенных участков проводов.

2.5 Убедиться в наличии заземления системного блока, монитора и защитного экрана.

2.6 Протереть антистатической салфеткой поверхность экрана монитора и защитного экрана.

2.7 Проверить правильность установки стола, стула, подставки для ног, пюпитра, угла наклона экрана, положение клавиатуры, положение "мыши" на специальном коврике, при необходимости произвести регулировку рабочего стола и кресла, а также расположение элементов компьютера в соответствии с требованиями эргономики и в целях исключения неудобных поз и длительных напряжений тела.

## 3 Требования безопасности во время работы

3.1 Работнику при работе на ПК запрещается:

- прикасаться к задней панели системного блока (процессора) при включенном питании;

- переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании;

- допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и других устройств;

- производить самостоятельное вскрытие и ремонт оборудования;

- работать на компьютере при снятых кожухах;

- отключать оборудование от электросети и выдергивать электровилку, держась за шнур.

3.2 Продолжительность непрерывной работы с компьютером без регламентированного перерыва не должна превышать 2-х часов.

3.3 Во время регламентированных перерывов с целью снижения нервно - эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинезии, предотвращения развития познотонического утомления выполнять комплексы упражнений.

## 4 Требования безопасности в аварийных ситуациях

4.1 Во всех случаях обрыва проводов питания, неисправности заземления и других повреждений, появления гари, немедленно отключить питание и сообщить об аварийной ситуации руководителю.

4.2 Не приступать к работе до устранения неисправностей.

4.3 При получении травм или внезапном заболевании немедленно известить своего руководителя, организовать первую доврачебную помощь или вызвать скорую медицинскую помощь.

## 5. Требования безопасности по окончании работы

5.1 Отключить питание компьютера.

5.2 Привести в порядок рабочее место.

5.3 Выполнить упражнения для глаз и пальцев рук на расслабление.

3. Должностные обязанности

# Раздел 2 Должностные обязанности front-end разработчика:

1 Разработка и поддержка пользовательского интерфейса: front-end разработчики отвечают за создание и поддержку веб-интерфейса для просмотра пользователями и взаимодействия с ними. Создавая удобный и интуитивно понятный интерфейс, необходимо обеспечить высокое качество взаимодействия с пользователем.

2 Реализация дизайнерского макета: front-end разработчики должны обладать навыками верстки графического дизайна и преобразования его в рабочий код. Он должен точно воспроизводить внешний вид и стиль настроек дизайнера и обеспечивать совместимость с различными браузерами и устройствами.

3 Разработка интерактивных элементов: front-end разработчики отвечают за создание интерактивных элементов интерфейса, таких как кнопки, формы и выпадающие списки. Необходимо обеспечить надлежащую работу и оперативность реагирования этих элементов, а также их проверку и обработку данных.

4 Внутренняя интеграция: разработчики должны иметь возможность взаимодействовать с внутренней частью приложения, получать доступ к API и обрабатывать входящие данные. Необходимо понимать основы работы с HTTP-запросами, обработки ошибок и управления состоянием приложения.

5 Оптимизируйте производительность и загрузку страниц: разработчики должны усердно работать над оптимизацией производительности веб-страниц, уменьшением размера файлов, количества запросов к серверу и использованием кэширования. Также необходимо обеспечивать быструю загрузку и отзывчивый интерфейс для улучшения взаимодействия с пользователем.

6 Тестирование и отладка: front-end разработчики должны создавать и запускать тесты, чтобы обеспечить правильное поведение интерфейса и обнаружить возможные ошибки. Необходимо уметь находить и устранять проблемы с кодом, связанные с взаимодействием между дисплеем и элементами интерфейса.

7 Кроссбраузерная и кроссплатформенная совместимость: разработчикам интерфейсов необходимо проверять и удостоверяться в том, что создаваемый ими интерфейс совместим с различными браузерами и устройствами. Необходимо не только использовать современные стандарты и технологии, но и учитывать функции различных платформ и устройств.

8 Сотрудничайте с другими разработчиками и командами: front-end разработчикам необходимо эффективно работать в команде и взаимодействовать с дизайнерами, серверными разработчиками и другими участниками проекта. Нужно быть готов к обмену идеями и поиску компромиссов для достижения общих целей.

# Раздел 3 Программное обеспечение

Visual Studio Code - кроссплатформенная интегрированная среда разработки (IDE), обеспечивающая удобный редактор кода, отладчик и множество полезных расширений для веб-разработки. Она предоставляет широкий набор инструментов для отладки, рефакторинга кода и автоматизации задач.

Git - распространенная система контроля версий, позволяющая отслеживать изменения в коде, производить откаты и совместную работу над проектом.

Node.js - программная платформа для выполнения JavaScript-кода на сервере, необходимая для работы с различными инструментами и библиотеками.

Figma - онлайн-редактор дизайна, который позволяет создавать макеты и прототипы веб-сайтов, приложений и других пользовательских интерфейсов.

npm - менеджер пакетов для Node.js, позволяющий устанавливать и управлять зависимостями проекта.

# Раздел 4 Выполнение индивидуального задания

## 4.1 Постановка задачи (ТЗ)

Основные функции:

1 Регистрация и авторизация пользователей

Требования:

Форма регистрации нового пользователя с указанием имени, электронной почты и пароля.

Форма авторизации для уже зарегистрированных пользователей.

2 Вывод недвижимости из базы данных

Требования:

Отображение списка объектов недвижимости, хранящихся в базе данных.

Вывод основной информации о каждом объекте: тип недвижимости, адрес, цена, площадь, фотографии.

Пагинация для удобного просмотра большого количества объектов.

3 Пользовательская фильтрация отображаемой недвижимости по типу

Требования:

Возможность фильтрации списка недвижимости по типу: коттедж, дом, квартира.

4 Отдельный вход для администраторов

Требования:

Проверка учетных данных администратора и предоставление доступа к панели администраторов.

5 Панель администраторов для управления недвижимостью в базе данных

Требования:

Интерфейс для добавления нового объекта недвижимости, включая все необходимые поля (тип, адрес, цена, площадь, фотографии и т.д.).

Интерфейс для редактирования существующих объектов недвижимости.

Возможность удаления объектов недвижимости из базы данных.

Просмотр списка всех объектов недвижимости с возможностью поиска и фильтрации.

Технические требования

Веб-сайт должен быть разработан с использованием современных веб-технологий (HTML, CSS, JavaScript).

Серверная часть должна быть реализована на языке программирования, поддерживающем работу с базами данных (например, Python, PHP, Node.js).

Использование реляционной базы данных (например, MySQL, PostgreSQL) для хранения информации о недвижимости и пользователях.

Адаптивный дизайн для корректного отображения на различных устройствах (ПК, планшеты, смартфоны).

Завершение

Данный проект должен быть завершен в течение 5 недель с даты начала работ. Все этапы разработки (анализ требований, проектирование, разработка, тестирование, запуск) должны быть выполнены с учетом стандартов качества и безопасности.

## 4.2 Разработка

4.2.1 Практическая реализация алгоритмов

Алгоритм - точное предписание, описывающее порядок действий (план действий), определяющих процесс решения задачи или достижения цели.

Рассмотрим алгоритм авторизации пользователя

Проверка метода запроса:

Проверяется, что метод запроса является POST. Если это не так, возвращается ответ с кодом 404 и сообщением "Bad Request".

Проверка наличия параметров:

Проверяется, что в POST-запросе присутствуют параметры userName и userPassword. Если какой-либо из них отсутствует, возвращается ответ с кодом 404 и сообщением "Invalid API parameters".

Извлечение параметров и подготовка запроса к базе данных:

Извлекаются значения userName и userPassword из POST-запроса.

Подключается файл DBConnect.php для соединения с базой данных.

Подготавливается SQL-запрос для получения данных пользователя с указанным именем пользователя (userName).

Выполнение запроса и проверка существования пользователя:

Выполняется подготовленный SQL-запрос.

Проверяется, есть ли в базе данных пользователь с указанным именем пользователя. Если пользователь не найден, возвращается ответ с кодом 404 и сообщением "Incorrect Login Credentials".

Проверка пароля:

Если пользователь найден, извлекаются его данные из результата запроса.

Проверяется соответствие введенного пароля (используя password\_verify) с хешированным паролем, хранящимся в базе данных.

Если пароль совпадает, формируется успешный ответ с кодом 200, содержащий данные пользователя (fullName, emailID, userName).

Если пароль не совпадает, возвращается ответ с кодом 404 и сообщением "Incorrect Login Credentials".

Обработка ошибок:

Если при выполнении запроса или обработке данных возникает исключение, возвращается ответ с кодом 404 и сообщением "Something Went Wrong!" с информацией об ошибке.

На рисунке Б.1 можно увидеть блок-схему данного алгоритма

4.2.2 Реализация базы данных

В ходе разработки веб-приложения была создана база данных, содержащая таблицы, приведенные в приложении Б. Таблицы содержат в себе данные о выставленной на продажу недвижимости, о клиентах и договорах. Также были созданы таблицы, содержащие данные пользователей и администраторов веб-приложения.

# 4.3 Руководство пользователя

4.3.1 Назначение приложения

Веб-приложение разработано для структурированного отображения списка недвижимости пользователю, упрощенных удаления недвижимости из базы данных и добавления недвижимости в базу данных посредством панели администратора.

4.3.2 Условия выполнения

Чтобы пользователь имел доступ к веб-приложению ему необходимо использовать последнюю версию одного из популярных браузеров, например Google Chrome, Yandex Browser, Safari или их аналогов.

4.3.3 Выполнение программы

Для регистрации пользователю необходимо перейти на страницу регистрации, ввести необходимые данные в поля ввода и нажать на кнопку «Зарегистрироваться»

Для авторизации пользователю необходимо перейти на страницу авторизации, ввести необходимые данные, под которыми он ранее регистрировался, и нажать на кнопку «Авторизоваться»

Для просмотра актуальный предложений по недвижимости пользователю необходимо перейти на страницу поиска недвижимости и выбрать, какую недвижимость он хочет увидеть, а после нажать кнопку поиска.

## 4.4 Тестирование системы

Для проверки корректной работы информационной системы были написаны и проведены тест-кейсы.

В Приложении Б описаны проведенные тест-кейсы в таблицах Б.1 и Б.2.

Тестирование выявило ноль ошибок, что подтверждает правильную работу веб-приложения.

# Выводы

Во время прохождения практики на предприятии была разработана информационная система для агентства недвижимости.

В процессе выполнения заданий были улучшены теоретические и практические знания по языку PHP и mySQL запросам.

# Приложение А

(Обязательное)

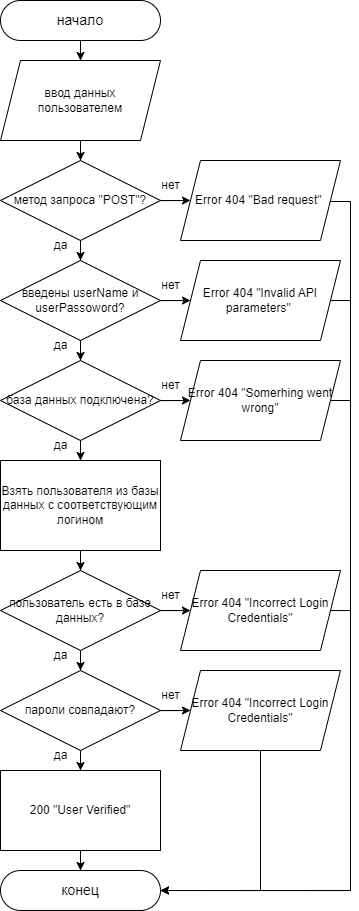


Рисунок А.1 - Блок-схема функции авторизации

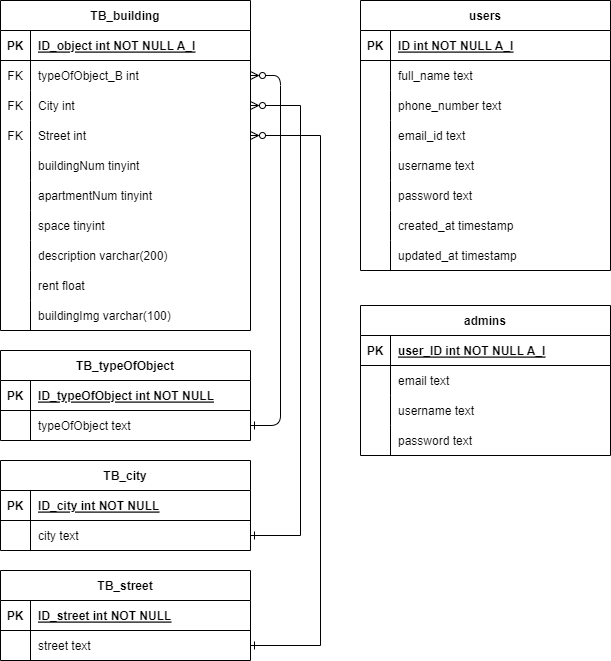


Рисунок А.2 – ER-диаграмма базы данных

# Приложение Б

(Обязательное)

Таблица Б.1 - Тест-кейс № 1

|  |  |  |
| --- | --- | --- |
|  |  | Комментарии |
| Тест № | TC\_FH\_1 | Расшифровка: TestCase\_FindHome\_1 |
| Приоритет тестирования | Высокий |  |
| Заголовок/название теста | Проверка ошибки в Username |  |
| Краткое изложение теста | Ввести в форму авторизации некорректные данные для поля Username(незарегистрированные) |  |
| Этапы теста | * 1. Зайти на главную страницу сайта;   2. Нажать левой кнопкой мыши по значку личного кабинета   3. Внести тестовые данные в поле username   4. Внести тестовые данные в поле Password   5. Отправляем форму нажатием левой кнопки мыши по кнопке «Sign In» |  |
| Тестовые данные | Для поля Email: [321232@yandex.com](mailto:321232@yandex.com)  Данные для поля Password: 4421 |  |
| Ожидаемый результат | Уведомление «Opps!! Incorrect Login Credentials» |  |
| Фактический результат | Уведомление «Opps!! Incorrect Login Credentials» |  |
| Статус | Выполнено |  |
| Предварительное условие | Запустить сайт в браузере Opera GX перейдя по рабочей ссылке |  |
| Постусловие | После выполнения теста поле должно быть снова очищено для дальнейшего тестирования |  |

Таблица Б.2 - Тест-кейс №2

|  |  |  |
| --- | --- | --- |
|  |  | Комментарии |
| Тест № | TC\_FH\_2 | Расшифровка: TestCase\_FindHome\_2 |
| Приоритет тестирования | Высокий |  |
| Заголовок/название теста | Проверка ошибки в Password |  |
| Краткое изложение теста | Ввести в форму авторизации некорректные данные для поля Email(незарегистрированные) |  |
| Этапы теста | * 1. Зайти на главную страницу сайта;   2. Нажать левой кнопкой мыши по значку личного кабинета   3. Внести тестовые данные в поле username   4. Внести тестовые данные в поле Password   5. Отправляем форму нажатием левой кнопки мыши по кнопке «Sign In» |  |
| Тестовые данные | Данные для поля Email: [karina.slake.koo@gmail.](mailto:karina.slake.koo@gmail.)com  Данные для поля Password: 3212 |  |
| Ожидаемый результат | Уведомление «Opps!! Incorrect Login Credentials » |  |
| Фактический результат | Уведомление «Opps!! Incorrect Login Credentials » |  |
| Статус | Выполнено |  |
| Предварительное условие | Запустить сайт в браузере Opera GX перейдя по рабочей ссылке |  |
| Постусловие | После выполнения теста поле должно быть снова очищено для дальнейшего тестирования |  |

# Приложение В

(Обязательное)

<?php

header('Content-type: application/json');

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

if (!empty($\_POST['userName']) && !empty($\_POST['userPassword'])) {

$userName = $\_POST['userName'];

$userPassword = $\_POST['userPassword'];

try {

require 'DBConnect.php';

$SELECT\_\_USER\_\_DATA = "SELECT \* FROM `users` WHERE users.username=:userName";

$select\_\_user\_\_statement = $con->prepare($SELECT\_\_USER\_\_DATA);

$select\_\_user\_\_statement->bindParam(':userName', $userName, PDO::PARAM\_STR);

$select\_\_user\_\_statement->execute();

$user\_\_flag = $select\_\_user\_\_statement->rowCount();

if ($user\_\_flag > 0) {

$user\_\_data = $select\_\_user\_\_statement->fetch(PDO::FETCH\_ASSOC);

if (password\_verify($userPassword, $user\_\_data['password'])) {

$user\_\_object = array(

"fullName"=>$user\_\_data['full\_name'],

"emailID"=>$user\_\_data['email\_id'],

"userName"=>$user\_\_data['username']

);

http\_response\_code(200);

$server\_\_response\_\_success = array(

"code" => http\_response\_code(200),

"status" => true,

"message" => "User Verified" ,

"userData"=>$user\_\_object

);

echo json\_encode($server\_\_response\_\_success);

} else {

http\_response\_code(404);

$server\_\_response\_\_error = array(

"code" => http\_response\_code(404),

"status" => false,

"message" => "Opps!! Incorrect Login Credentials"

);

echo json\_encode($server\_\_response\_\_error);

}

} else {

http\_response\_code(404);

$server\_\_response\_\_error = array(

"code" => http\_response\_code(404),

"status" => false,

"message" => "Opps!! Incorrect Login Credentials"

);

echo json\_encode($server\_\_response\_\_error);

}

} catch (Exception $ex) {

http\_response\_code(404);

$server\_\_response\_\_error = array(

"code" => http\_response\_code(404),

"status" => false,

"message" => "Opps!! Something Went Wrong! " . $ex->getMessage()

);

echo json\_encode($server\_\_response\_\_error);

}

} else {

http\_response\_code(404);

$server\_\_response\_\_error = array(

"code" => http\_response\_code(404),

"status" => false,

"message" => "Invalid API parameters! Please contact the administrator or refer to the documentation for assistance."

);

echo json\_encode($server\_\_response\_\_error);

}

} else {

http\_response\_code(404);

$server\_\_response\_\_error = array(

"code" => http\_response\_code(404),

"status" => false,

"message" => "Bad Request"

);

echo json\_encode($server\_\_response\_\_error);

}

Листинг В.1 – Тестируемый код в таблице Б.1 и таблице Б.2