Отчет о проверке

Автор: Попов Н. Д.

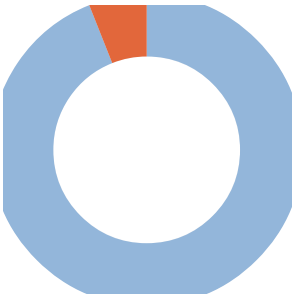
Название документа: 2026-11-16_113102-6afabf466db40.docx

Проверяющий:

Название организации: ГБПОУ РО «Каменский техникум строительства и автосервиса»

Для скачивания отчета воспользуйтесь QR кодом.

РЕЗУЛЬТАТЫ ПРОВЕРКИ



Совпадения: 6.16%

Оригинальность: 93.84%

Цитирования: 0%

Самоцитирования: 0%

ИИ-контент: 0%

«Совпадения», «Цитирования», «Самоцитирования», «Оригинальность» являются отдельными показателями, отображаются в процентах и в сумме дают 100%, что соответствует полному тексту проверяемого документа.

- **Совпадения** — фрагменты проверяемого текста, полностью или частично сходные с найденными источниками, за исключением фрагментов, которые система отнесла к цитированию или самоцитированию. Показатель «Совпадения» — это доля фрагментов проверяемого текста, отнесенных к совпадениям, в общем объеме текста.
- **Самоцитирования** — фрагменты проверяемого текста, совпадающие или почти совпадающие с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа. Показатель «Самоцитирования» — это доля фрагментов проверяемого текста, отнесенных к самоцитированию, в общем объеме текста.
- **Цитирования** — фрагменты проверяемого текста, которые не являются авторскими, но которые система отнесла к корректно оформленным. К цитированиям относятся также шаблонные фразы; библиография; фрагменты текста, найденные модулем поиска «СПС Гарант: нормативно-правовая документация».
- **Текстовое пересечение** — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
- **Источник** — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
- **Оригинальный текст** — фрагменты проверяемого текста, не обнаруженные ни в одном источнике и не отмеченные ни одним из модулей поиска.

Обращаем Ваше внимание, что система находит текстовые совпадения проверяемого документа с проиндексированными в системе источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности совпадений или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

Номер документа: **137942**

Тип документа: **Курсовая работа**

Дата проверки: **22.11.2026 22:57:44**

Дата корректировки: **Нет**

Комментарий: не указано

Количество страниц: **25**

Символов в тексте: **31189**

Слов в тексте: **3652**

Число предложений: **255**

ПАРАМЕТРЫ ПРОВЕРКИ

Выполнена проверка с учетом редактирования: Да

Выполнено распознавание текста (OCR): Нет

Выполнена проверка с учетом структуры: Нет

Модули поиска: ИПС Адилет, Библиография, Сводная коллекция ЭБС, Интернет Плюс, Сводная коллекция РГБ, Переводные заимствования (RuEn), Переводные заимствования по eLIBRARY.RU (EnRu), Переводные заимствования по Интернету (EnRu), Переводные заимствования издательства Wiley (RuEn), eLIBRARY.RU, СПС ГАРАНТ, Модуль поиска "КГАУ", Медицина, Диссертации НББ, Перефразирования по eLIBRARY.RU, Перефразирования по Интернету, Патенты СССР, РФ, СНГ, СМИ России и СНГ, Шаблонные фразы, Кольцо вузов, Издательство Wiley, Переводные заимствования, Коллекция НБУ, Перефразирования по коллекции издательства Wiley, Перефразирования по СПС ГАРАНТ: аналитика, СПС ГАРАНТ: аналитика, СПС ГАРАНТ: нормативно-правовая документация, Интернет

ИСТОЧНИКИ

№	Доля в тексте	Доля в отчете	Источник	Актуален на	Модуль поиска	Комментарии
[1]	1.42%	0.98%	Современные технологии управления ИТ-проектами http://elibrary.ru	11 июн 2025	Публикации eLIBRARY	
[2]	1.26%	0.79%	Аутсорсинг ИТ-услуг: теория и практика https://cyberleninka.ru/article/	23 апр 2025	Интернет плюс	
[3]	1.15%	0.86%	Информационные системы управления предприятием http://elibrary.ru	23 июл 2024	Публикации eLIBRARY	
[4]	1.08%	0.74%	Автоматизация бизнес-процессов предприятия https://neirosova.ru/articles/	5 апр 2026	Интернет плюс	
[5]	1.02%	0.00%	Проектирование баз данных информационных систем http://elibrary.ru	13 июл 2025	Публикации eLIBRARY(переводы и перефразирования)	Источник исключен. Причина: Малый процент пересечения.
[6]	0.94%	0.00%	Разработка веб-приложений на PHP и MySQL http://el.anbook.com	9 авг 2026	Сводная коллекция ЭБС	Источник исключен. Причина: Малый процент пересечения.
[7]	0.87%	0.00%	Сравнение альтернативных методологий моделирования www.1urok.ru	24 фев 2025	Интернет плюс	Источник исключен. Причина: Малый процент пересечения.
[8]	0.55%	0.00%	ГОСТ 34.601-90. Автоматизированные системы. Стадии создания http://normacs.net	29 дек 2025	Перефразированные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Малый процент пересечения.
[9]	0.47%	0.00%	Оценка эффективности внедрения информационных систем http://dl.lib.rst.ru	8 апр 2026	Публикации РГБ (переводы и перефразирования)	Источник исключен. Причина: Малый процент пересечения.
[10]	0.41%	0.00%	Базы данных: проектирование и реализация http://dokumen.pub	26 фев 2026	Интернет плюс	Источник исключен. Причина: Малый процент пересечения.
[11]	0.36%	0.00%	Тестирование программного обеспечения: подходы и методы http://dl.lib.rst.ru	1 мая 2024	Публикации РГБ (переводы и перефразирования)	Источник исключен. Причина: Малый процент пересечения.

ТЕКСТ ДОКУМЕНТА

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3
1 Теоретическая часть 5
1.1 Описание предметной области 5
1.2 Обоснование выбора программного обеспечения 8
2 Практическая часть 12
2.1 Методологии моделирования и разработки ИС и БД 12
2.2 Описание этапов тестирования программного решения 15
ЗАКЛЮЧЕНИЕ 17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18
ПРИЛОЖЕНИЕ А – Структурная диаграмма ИС 19
ПРИЛОЖЕНИЕ Б – Диаграмма вариантов использования 20
ПРИЛОЖЕНИЕ В – ER-диаграмма БД 21
ПРИЛОЖЕНИЕ Г – Схема пользовательского интерфейса 22
ПРИЛОЖЕНИЕ Д – Результаты тестирования 23
ПРИЛОЖЕНИЕ Е – Листинг программы 26

ВВЕДЕНИЕ

Развитие информационных технологий изменяет способы организации работы предприятий и взаимодействия участников проектов. [1] Для выполнения отдельных задач организации могут привлекать внешних специалистов, сохраняя за штатными сотрудниками функции планирования, контроля и сопровождения основных бизнес-процессов. [1] Такая форма взаимодействия называется аутсорсингом. [1] Она позволяет искать специалистов с нужными навыками под конкретную задачу, однако требует согласования условий, сроков и ответственности участников.

При организации аутсорсинга с помощью электронной почты, телефонных звонков и отдельных документов сведения о заказах и исполнителях распределяются по разным каналам. В результате усложняется поиск актуальной информации, возрастает вероятность пропустить отклик или повторно запрашивать одни и те же сведения. Применение информационной системы позволяет объединить данные о заказах, предложениях исполнителей и услугах в одном веб-приложении.

Актуальность темы обусловлена потребностью в удобном инструменте публикации задач, поиска исполнителей и обработки откликов. Для компании, занимающейся внедрением и сопровождением программных продуктов «1С», дополнительные компетенции могут требоваться, например, при разработке веб-интеграций, мобильных приложений и аналитических решений. Информационная система «Сервис аутсорс» рассматривается как средство поддержки такого взаимодействия на примере ООО «Высокие технологии».

Объект исследования – процесс организации взаимодействия заказчиков и внешних исполнителей при выполнении ИТ-заказов. Предмет исследования – методы и программные средства проектирования и разработки веб-приложения для размещения заказов и обработки откликов.

Цель курсовой работы – проектирование и разработка информационной системы «Сервис аутсорс», обеспечивающей представление заказов, взаимодействие заказчиков с исполнителями и управление данными предметной области.

Для достижения поставленной цели необходимо решить следующие задачи: проанализировать предметную область аутсорсинга ИТ-услуг; определить категории пользователей и требования к системе; обосновать выбор программных средств; разработать модели информационной системы, базы данных и пользовательского интерфейса; реализовать веб-приложение с регистрацией, авторизацией и электронными формами; подготовить и выполнить тестирование основных функций.

В работе применяются методы анализа предметной области и требований, сравнения программных средств, моделирования данных и пользовательских сценариев, а также функционального тестирования. При разработке

рассматриваются технологии HTML, CSS, JavaScript, PHP и система управления базами данных MySQL [1–5].

Практическая значимость проекта состоит в объединении каталога услуг, заказов и откликов в одном учебном веб-приложении. Решение не заявляется как новая для рынка технология. [1]

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Описание предметной области

Аутсорсинг в сфере информационных технологий предполагает передачу отдельных работ внешнему исполнителю. Предметом таких работ могут быть настройка и доработка программ, интеграция информационных систем, техническая поддержка, создание отчётов и обработка данных. Для качественного взаимодействия заказчику необходимо сформулировать содержание задачи и ограничения, а исполнителю – оценить собственные возможности и предложить условия выполнения.

В качестве объекта автоматизации рассматриваются процессы размещения и обработки ИТ-заказов в рамках деятельности ООО «Высокие технологии». В исходном проекте компания представлена как организация, оказывающая услуги по внедрению и сопровождению решений «1С». Разрабатываемое приложение дополняет информационное представление компании средствами работы с заказами и внешними специалистами.

В рассматриваемом процессе участвуют четыре категории пользователей: гость, заказчик, исполнитель и администратор. Гость может просматривать опубликованные материалы и воспользоваться формой обратной связи. Для работы с персональными данными, заказами и откликами пользователь регистрируется и входит в систему. При регистрации выбирается роль заказчика или исполнителя; административные действия предназначены для администратора.

Заказчик создаёт заказ, указывая название, описание, категорию, бюджет, срок, сложность и необходимые навыки. После публикации заказ становится доступным в каталоге. Заказчик просматривает предложения и выбирает исполнителя. Система предусматривает изменение состояния заказа и отображение принадлежащих заказчику записей в личном кабинете.

Исполнитель просматривает каталог заказов и может искать подходящие задачи по доступным параметрам. В своём профиле он указывает специализацию, навыки, опыт и другие сведения. Отклик включает предложенную стоимость, срок выполнения и сопроводительное сообщение. Информация об отправленных откликах и их статусах отображается исполнителю в личном кабинете.

Администратор управляет пользователями, заказами, откликами, категориями, услугами и сообщениями обратной связи. Такое распределение полномочий позволяет отделить публичный просмотр информации от изменения данных и административных действий. Проверка прав должна выполняться на стороне сервера, поскольку скрытие элементов интерфейса само по себе не ограничивает доступ к обработчикам [6].

Электронный документооборот в системе представлен записями о заказах, откликах и сообщениях, а не полноценным юридически значимым обменом документами. Сначала заказчик вводит сведения о задаче, затем исполнитель передаёт предложение через электронную форму. После обработки отклика соответствующие сведения и статус сохраняются в базе данных. Заключение договора, оплата и обмен закрывающими документами в рассматриваемой версии приложения не реализованы.

К основным объектам предметной области относятся пользователь, категория, услуга, заказ, отклик и сообщение. В SQL-скрипте проекта этим объектам соответствуют таблицы users, categories, services, orders, responses и messages. Запись пользователя содержит роль и контактные данные; заказ связан с заказчиком, категорией и, при назначении, исполнителем. Отклик связывает конкретный заказ с исполнителем и хранит предложенные условия.

Один пользователь в роли заказчика может разместить несколько заказов, а один заказ может получить несколько откликов. Один исполнитель может откликаться на разные заказы. В таблице откликов предусмотрено ограничение уникальности пары «заказ – исполнитель», препятствующее созданию повторных записей для одной пары. Категория может объединять несколько заказов и услуг. Связи между таблицами задаются внешними ключами [5].

Для заказов предусмотрены состояния «открыт», «в работе», «завершён» и «отменён», для откликов – «ожидает», «принят» и «отклонён». Изменение состояний должно сохранять согласованность между заказом и откликами. Анализ обработчиков выявил, что отдельные переходы состояний требуют дополнительной проверки и доработки: например, повторное принятие другого отклика может оставить более одного отклика в состоянии «принят». Поэтому соответствие бизнес-правилам необходимо подтвердить отдельными тестами.

Функциональные требования к системе включают регистрацию и авторизацию пользователей, разграничение доступа по ролям, просмотр и поиск заказов, создание и редактирование заказов, отправку и обработку откликов, редактирование профиля, просмотр услуг и передачу сообщений обратной связи. Не менее трёх электронных форм должны обеспечивать авторизованному пользователю работу с данными предметной области. В проекте для этого предусмотрены, в частности, формы профиля, заказа и отклика; доступ к конкретным формам зависит от роли.

К эксплуатационным требованиям относятся работа через современный браузер, доступность сервера приложений и базы данных, корректное отображение русского текста и возможность использования интерфейса на различных размерах экрана. Для функционирования системы необходимы PHP с расширением PDO MySQL и сервер MySQL или совместимая СУБД. При размещении в сети требуется дополнительно настроить защищённое соединение HTTPS, резервное копирование, права доступа и защиту персональных данных.

1.2 Обоснование выбора программного обеспечения

При выборе программного обеспечения учитываются назначение системы, необходимость хранения связанных данных, поддержка регистрации и ролей, возможность запуска в учебной среде и доступность документации. Рассмотрим средства, применённые в проекте, и возможные альтернативы.

Веб-приложение использует клиент-серверную архитектуру. Браузер запрашивает страницы и отправляет данные форм, а серверный код обрабатывает запросы и взаимодействует с базой данных. HTML задаёт структуру страниц и элементов форм [1], CSS отвечает за оформление и адаптацию интерфейса [2], JavaScript дополняет страницы интерактивным поведением [3]. Эти средства поддерживаются современными браузерами и не требуют установки отдельного клиентского приложения.

В качестве серверного языка выбран PHP.[1] В проекте PHP-файлы формируют страницы, проверяют данные форм, управляют сессией и выполняют запросы к базе данных через PDO.[2] Документация PHP описывает механизм сессий, средства обработки паролей и интерфейс PDO [4].[2] Выбор PHP обоснован наличием соответствующей реализации в проекте, распространённостью учебных примеров и возможностью запуска в локальной серверной среде.[2]

В качестве альтернативы можно использовать Python с фреймворком Django или JavaScript на серверной платформе Node.js.[3] Django предоставляет средства разработки веб-приложений, включая систему моделей и административный интерфейс [7].[3] Node.js позволяет применять JavaScript и на клиенте, и на сервере [8].[3] При этом переход на любую из этих платформ потребовал бы переработки существующих PHP-обработчиков и организации окружения. Для продолжения работы над имеющимся проектом использование PHP целесообразнее.

Для хранения данных выбрана реляционная СУБД MySQL.[3] Табличная организация, ключи и ограничения позволяют представить пользователей, заказы и отклики как связанные объекты.[4] Механизм InnoDB поддерживает внешние ключи и транзакции, что важно при выполнении связанных изменений [5].[4] В проекте используется кодировка utf8mb4 для хранения русскоязычного текста.

В качестве альтернативы можно рассмотреть PostgreSQL и SQLite.[4] PostgreSQL поддерживает развитые средства работы с реляционными данными и серверную архитектуру [9], однако переход на неё потребовал бы адаптации SQL-скрипта и подключения. SQLite удобна для небольших локальных приложений за счёт хранения базы в файле [10], но при развёртывании многопользовательского веб-приложения необходимо учитывать особенности одновременной записи. Выбор MySQL соответствует уже подготовленному SQL-скрипту и описанной в проекте схеме запуска.[5]

Для локального развёртывания в документации проекта предлагается OpenServer, а для импорта и просмотра базы – phpMyAdmin. OpenServer объединяет компоненты, необходимые для локальной разработки, а phpMyAdmin предоставляет веб-интерфейс администрирования MySQL [11, 12]. В качестве альтернативы можно настроить PHP и MySQL отдельно или использовать другой локальный серверный пакет. Преимущество предложенного варианта для учебной работы – единая среда настройки и запуска; при этом конкретные версии модулей и параметры подключения должны соответствовать установленной конфигурации.

Для редактирования исходного кода может применяться Visual Studio Code, поддерживающий работу с файлами проекта, подсветку синтаксиса и расширения [13]. Альтернативами являются, например, PhpStorm и другие редакторы с поддержкой PHP. Выбор редактора не влияет на принцип работы приложения; Visual Studio Code удобен как доступный инструмент для просмотра и изменения файлов HTML, CSS, JavaScript и PHP.

Безопасность системы обеспечивается сочетанием организационных и программных мер. В имеющемся коде пароли хешируются средствами PHP, для запросов к базе применяются подготовленные выражения PDO, а в формах используются CSRF-токены. Эти механизмы снижают отдельные риски, но не заменяют комплексную проверку безопасности и корректности авторизации [4, 6]. Для публичного использования необходимо устранить найденные недостатки, изменить демонстрационные пароли и ограничить доступ к диагностическим и служебным файлам.

При подготовке среды следует учитывать расхождение между файлами проекта: SQL-скрипт создаёт базу hightech_portal, тогда как в config/config.php указано имя hightech. Перед запуском параметры необходимо согласовать. [5] Повторный импорт подготовленного скрипта удаляет существующие таблицы, поэтому его нельзя применять к рабочим данным без резервной копии.

Таким образом, выбранный набор HTML, CSS, JavaScript, PHP и MySQL соответствует структуре существующего проекта и позволяет реализовать поставленные функции. Окончательный вывод о работоспособности системы может быть сделан только после проверки основных сценариев в настроенной среде с реальной базой данных.

2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Методологии моделирования и разработки ИС и БД

На практическом этапе рассматривается существующее веб-приложение hightech, его структура, схема данных и пользовательские действия. Разработка описывается последовательно: анализ требований, построение функциональной модели, описание пользовательских сценариев, проектирование данных, реализация интерфейса и обработчиков, подготовка проверок. Модели предназначены для пояснения реализованных компонентов, а не для утверждения о полном соответствии приложения всем требованиям.

Функциональная декомпозиция разделяет систему на публичный раздел, кабинет заказчика, кабинет исполнителя, административную часть и общие компоненты. Публичные страницы предоставляют доступ к услугам, заказам и исполнителям. [6] Личные кабинеты используют данные учётной записи и позволяют выполнять действия, соответствующие роли. Функциональная схема приведена в приложении А. [6]

Для определения внешних взаимодействий используется модель вариантов использования. На ней показаны гость, заказчик, исполнитель и администратор. [7] Гость просматривает каталоги и может обратиться через форму; зарегистрированные пользователи управляют собственными данными; администратор выполняет управление записями. Основные связи акторов с вариантами использования показаны в приложении Б. Диаграмма отражает основные действия, а не полный перечень всех страниц административной панели.

Модель данных разрабатывается с использованием сущностей и связей. В приложении В приведены шесть таблиц из sql/database.sql. Таблица users хранит общие сведения и роль; categories – справочник направлений; services – сведения об услугах; orders – условия заказов; responses – предложения исполнителей; messages – обращения. Первичные ключи обозначаются PK, внешние – FK. Для responses определено уникальное сочетание order_id и freelancer_id.

Заказ связан с пользователем-заказчиком обязательным внешним ключом customer_id. Поле freelancer_id может быть пустым до назначения исполнителя. Категория у заказа и услуги может отсутствовать. При удалении

заказчика предусмотрено каскадное удаление связанных заказов; при удалении исполнителя ссылка на него в заказе обнуляется. Отклики связаны с заказами и исполнителями; сообщения не имеют внешних ключей. Конкретные правила удаления задаются SQL-скриптом.

Схема пользовательского интерфейса в приложении Г представляет упрощённый каркас страницы списка заказов: шапку, навигацию, блок фильтров, карточки заказов и подвал. Это проектная схема расположения элементов, а не снимок экрана работающего приложения. В качестве продолжения практической части необходимо получить и вставить реальные скриншоты страниц после запуска сайта с базой данных.

Организация файлов приложения разделяет публичные страницы, общие компоненты, административные страницы, стили и скрипт базы данных. Файлы includes/header.php и includes/footer.php формируют общие части страниц; includes/functions.php содержит функции сессии, авторизации, обработки сообщений и CSRF; includes/db.php создаёт подключение PDO и вспомогательные функции запросов. Папка admin содержит формы и списки административного раздела.

Обработка регистрации выполняется в register.php: сервер получает значения формы, проверяет их и создаёт запись пользователя. Пароль перед сохранением обрабатывается функцией password_hash. На странице login.php введённый пароль сравнивается с сохранённым хешем посредством password_verify. Авторизованная сессия позволяет определить текущего пользователя и его роль [4]. Корректность сценариев с реальной базой требует отдельной проверки.

Для работы с заказами используются orders.php, order.php, order-form.php и order-action.php. Список показывает карточки и параметры поиска, страница заказа содержит сведения о задаче, форма заказчика передаёт данные на сервер, а отдельный обработчик изменяет статус заказа. Входные значения должны проверяться сервером, поскольку клиентские ограничения можно обойти.

Работа с откликами реализована в response-create.php, response-action.php и my-responses.php. Исполнитель указывает стоимость, срок и сообщение; обработчик проверяет состояние заказа, значения полей и наличие существующего отклика. При выборе отклика обработчик обновляет записи отклика и заказа в транзакции. При этом сама транзакция не заменяет проверку допустимых переходов состояний: обнаруженный ранее случай повторного принятия другого отклика требует исправления.

Форма обратной связи в contacts.php сохраняет сведения в таблицу messages. Профиль пользователя изменяется через profile.php. Административная часть предусматривает управление пользователями, заказами, категориями, услугами, откликами и обращениями. Такие действия следует проверять с учётом различных ролей, а не только по наличию соответствующих кнопок.

Для локального запуска необходимо установить сервер PHP с PDO MySQL и MySQL, создать отдельную тестовую базу импортом hightech/sql/database.sql и настроить hightech/config/config.php. Имя базы в скрипте – hightech_portal, тогда как в текущей конфигурации указано hightech; эти значения необходимо согласовать. Скрипт содержит DROP TABLE, поэтому повторный импорт может уничтожить данные. Перед импортом важно выбрать отдельную тестовую среду или сделать резервную копию.

В приложении Е приведены фрагменты исходного кода форм, связанных с базой данных: редактирования профиля, создания и изменения заказа и отправки отклика. Полные исходные файлы приложения, включая вспомогательные функции, находятся в каталоге hightech.

2.2 Описание этапов тестирования программного решения

Тестирование проводится для сопоставления ожидаемого и фактического поведения. Подготовка включает отдельную тестовую базу, запуск веб-сервера, создание или импорт тестовых учётных записей и проверку настроек подключения. Для каждого сценария фиксируются исходное состояние, шаги, данные, ожидаемый и фактический результат.

В рамках доступной среды 02.11.2026 выполнена синтаксическая проверка 39 PHP-файлов с использованием PHP 8.5 в среде WebAssembly; сообщений о синтаксических ошибках не получено. Проверка синтаксиса JavaScript-файла assets/js/app.js командой node --check также завершилась без ошибок. Эти проверки не подтверждают работу приложения с MySQL, корректность интерфейса или выполнение бизнес-правил.

Ранее при изолированной проверке обработчика откликов с имитацией зависимостей обнаружены переходы, способные привести к двум принятым откликам и изменению состояния завершённого заказа. При проверке конфигурационных файлов установлено различие имени базы в настройках и SQL-скрипте. Данные случаи требуют повторного воспроизведения и устранения в полноценной среде; они не должны трактоваться как успешные испытания.

Подготовленные тест-кейсы для регистрации, входа, заказов, откликов, ограничений доступа и обратной связи приведены в приложении Д. Для сценариев, не выполненных с реальным сервером и базой данных, указан статус «Не выполнен». После настройки среды необходимо заполнить фактические результаты, провести повторное тестирование после исправления дефектов и приложить подтверждающие скриншоты.

ЗАКЛЮЧЕНИЕ

В работе рассмотрена предметная область организации аутсорсинга ИТ-задач на примере веб-приложения «Высокие технологии». Определены основные участники взаимодействия: гость, заказчик, исполнитель и администратор. Описаны потоки данных между заказом, откликом и пользователями, а также функциональные и эксплуатационные требования к системе.

Проанализированы применённые средства разработки HTML, CSS, JavaScript, PHP и MySQL. Подготовлены функциональная схема, диаграмма вариантов использования, ER-диаграмма и схема интерфейса. Описаны структура базы данных и назначение основных файлов приложения. Сформирован набор тест-кейсов для проверки пользовательских сценариев и ограничений доступа.

В доступной среде выполнены синтаксические проверки PHP и JavaScript без обнаружения синтаксических ошибок. Вместе с тем работа сайта с реальной базой и пользовательские сценарии не были проверены, поэтому цель подтверждена не полностью. Согласование имени базы в конфигурации, исправление обнаруженных переходов статусов и последующее функциональное тестирование являются необходимыми этапами завершения проекта.

Для завершения работы необходимы исправление выявленных недостатков, проверка безопасности, запуск с базой данных и оформление фактических результатов тестов. После запуска следует дополнить работу снимками экранов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 HTML Living Standard [Электронный ресурс]. – URL: https://html.spec.whatwg.org/multipage/.

2 CSS: Cascading Style Sheets. MDN Web Docs [Электронный ресурс]. – URL: https://developer.mozilla.org/en-US/docs/Web/CSS.

3 JavaScript. MDN Web Docs [Электронный ресурс]. – URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

4 PHP Manual [Электронный ресурс]. – URL: https://www.php.net/manual/en/.

5 MySQL 8.4 Reference Manual [Электронный ресурс]. – URL: https://dev.mysql.com/doc/refman/8.4/en/.

6 OWASP. Authorization Cheat Sheet [Электронный ресурс]. – URL: https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html.

7 Django documentation [Электронный ресурс]. – URL: https://docs.djangoproject.com/.

8 Node.js Documentation [Электронный ресурс]. – URL: https://nodejs.org/docs/latest/api/.

9 PostgreSQL Documentation [Электронный ресурс]. – URL: https://www.postgresql.org/docs/.

10 SQLite Documentation [Электронный ресурс]. – URL: https://www.sqlite.org/docs.html.

11 Open Server Panel [Электронный ресурс]. – URL: https://ospanel.io/.

12 phpMyAdmin Documentation [Электронный ресурс]. – URL: https://docs.phpmyadmin.net/.

13 Visual Studio Code Documentation [Электронный ресурс]. – URL: https://code.visualstudio.com/docs.

ПРИЛОЖЕНИЕ А

Структурная диаграмма ИС (функциональная схема)

Функциональные компоненты и их связь с общими средствами приложения представлены на рисунке А.1.

Рисунок А.1 – Функциональная структура системы

ПРИЛОЖЕНИЕ Б

Диаграмма вариантов использования (прецедентов)

Основные акторы и доступные им варианты использования приведены на рисунке Б.1.

Рисунок Б.1 – Диаграмма вариантов использования

ПРИЛОЖЕНИЕ В

ER-диаграмма базы данных

Сущности, ключевые атрибуты и связи базы данных приведены на рисунке В.1.

Рисунок В.1 – Сущности и связи базы данных

ПРИЛОЖЕНИЕ Г

Схема пользовательского интерфейса

Расположение основных элементов страницы каталога заказов приведено на рисунке Г.1. Схема является проектным макетом.

Рисунок Г.1 – Схема страницы каталога заказов

ПРИЛОЖЕНИЕ Д

Результаты тестирования (TEST-CASE)

Среда синтаксической проверки: PHP 8.5 WebAssembly, Node.js; 02.11.2026. Функциональные сценарии ниже подготовлены для последующего запуска с MySQL и браузером. Статусы отражают только фактически выполненные проверки.

ПРИЛОЖЕНИЕ Е

Листинг программы

В приложении приведены фрагменты обработчиков трёх электронных форм, работающих с базой данных. Номера строк и пути соответствуют исходному проекту hightech; приведённые фрагменты не являются самостоятельными программами и используют общие функции приложения.

Е.1 Форма редактирования профиля (hightech/profile.php, строки 14–51)

Обработчик проверяет данные и обновляет запись пользователя в таблице users.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    verify_csrf();
    $action = (string) ($_POST['action'] ?? '');
    if ($action === 'profile') {
        $name           = trim((string) ($_POST['name'] ?? ''));
        $phone          = trim((string) ($_POST['phone'] ?? ''));
        $company        = trim((string) ($_POST['company'] ?? ''));
        $specialization = trim((string) ($_POST['specialization'] ?? ''));
        $skills         = trim((string) ($_POST['skills'] ?? ''));
        $rate           = trim((string) ($_POST['rate'] ?? ''));
        $experience     = trim((string) ($_POST['experience'] ?? ''));
        $about          = trim((string) ($_POST['about'] ?? ''));
        if (mb_strlen($name) < 2) {
            $errors['name'] = 'Укажите имя или название организации.';
```

```
        }
        if (!$errors) {
            q(
```

```
'UPDATE users
                    SET name = ?, phone = ?, company = ?, specialization = ?, skills = ?, rate = ?, experience = ?, about
= ?
                  WHERE id = ?',
                [
                    $name,
                    $phone ?: null,
                    $user['role'] === 'customer' ? ($company ?: null) : null,
                    $user['role'] === 'freelancer' ? ($specialization ?: null) : null,
                    $user['role'] === 'freelancer' ? ($skills ?: null) : null,
                    $user['role'] === 'freelancer' && $rate !== '' ? max(0, (int) $rate) : null,
                    $user['role'] === 'freelancer' && $experience !== '' ? max(0, (int) $experience) : null,
                    $about ?: null,
                    (int) $user['id'],
                ]
            );
            flash('Профиль сохранён.', 'success');
            redirect('/profile.php');
        }
```

E.2 Форма создания и редактирования заказа (hightech/order-form.php, строки 35–98)

Обработчик проверяет данные и выполняет добавление или обновление записи в таблице orders.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    verify_csrf();
    $data['title']       = trim((string) ($_POST['title'] ?? ''));
    $data['description'] = trim((string) ($_POST['description'] ?? ''));
    $data['budget']      = (string) ($_POST['budget'] ?? '');
    $data['deadline']    = (string) ($_POST['deadline'] ?? '');
    $data['complexity']  = (string) ($_POST['complexity'] ?? 'medium');
    $data['skills']      = trim((string) ($_POST['skills'] ?? ''));
    $data['category_id'] = (int) ($_POST['category_id'] ?? 0);
    if (mb_strlen($data['title']) < 10) {
        $errors['title'] = 'Название должно содержать не менее 10 символов.';
    }
    if (mb_strlen($data['description']) < 50) {
        $errors['description'] = 'Опишите задачу подробнее — не менее 50 символов.';
    }
    if ((int) $data['budget'] < 1000) {
        $errors['budget'] = 'Бюджет должен быть не менее 1 000 ₽.';
    }
```

```
if (!in_array($data['complexity'], complexities(), true)) {
        $data['complexity'] = 'medium';
    }
    if ($data['deadline'] !== '' && strtotime($data['deadline']) === false) {
        $errors['deadline'] = 'Укажите корректную дату.';
    }
    if (!q_val('SELECT id FROM categories WHERE id = ?', [$data['category_id']])) {
        $errors['category_id'] = 'Выберите категорию.';
    }
    if (!$errors) {
        $params = [
            $data['category_id'],
            $data['title'],
            $data['description'],
            (int) $data['budget'],
            $data['deadline'] ?: null,
            $data['complexity'],
            $data['skills'] ?: null,
        ];
```

```
if ($order) {
            $params[] = (int) $order['id'];
            q(
                'UPDATE orders
                    SET category_id = ?, title = ?, description = ?, budget = ?, deadline = ?, complexity = ?, skills = ?
```

```
                  WHERE id = ?',
                 $params
             );
             flash('Изменения заказа сохранены.', 'success');
             redirect('/order.php?id=' . (int) $order['id']);
         }
         $params[] = (int) $user['id'];
         q(
             'INSERT INTO orders (category_id, title, description, budget, deadline, complexity, skills, customer_id,
status)
              VALUES (?, ?, ?, ?, ?, ?, ?, ?, "open")',
             $params
         );
         $newId = last_id();
```

flash('Заказ опубликован и доступен исполнителям.', 'success');
redirect('/order.php?id=' . $newId);
}
}

E.3 Форма отправки отклика (hightech/response-create.php, строки 7–62)

Обработчик проверяет доступ, заказ и введённые значения, затем сохраняет отклик в таблице responses.

```
$user = require_role(['freelancer']);
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    redirect('/orders.php');
}
verify_csrf();
$orderId = (int) ($_POST['order_id'] ?? 0);
$price   = (int) ($_POST['price'] ?? 0);
$days    = (int) ($_POST['days'] ?? 0);
$message = trim((string) ($_POST['message'] ?? ''));
$back = '/order.php?id=' . $orderId;
$order = q_one('SELECT * FROM orders WHERE id = ?', [$orderId]);
if ($order === null) {
    flash('Заказ не найден.', 'danger');
    redirect('/orders.php');
}
```

```
if ($order['status'] !== 'open') {
    flash('Этот заказ закрыт для новых откликов.', 'warning');
    redirect($back);
}
$errors = [];
if ($price < 1000) {
    $errors[] = 'Укажите стоимость не менее 1 000 ₽.';
}
if ($days < 1 || $days > 365) {
    $errors[] = 'Срок выполнения должен быть от 1 до 365 дней.';
}
if (mb_strlen($message) < 30) {
    $errors[] = 'Сопроводительное письмо должно содержать не менее 30 символов.';
}
$exists = (int) q_val('SELECT COUNT(*) FROM responses WHERE order_id = ? AND freelancer_id = ?', [$orderId, (int)
$user['id']]);
if ($exists > 0) {
    $errors[] = 'Вы уже отправляли отклик на этот заказ.';
}
```

```
if ($errors) {
    foreach ($errors as $error) {
        flash($error, 'danger');
    }
    redirect($back);
}
q(
    'INSERT INTO responses (order_id, freelancer_id, message, price, days, status)
     VALUES (?, ?, ?, ?, ?, "pending")',
    [$orderId, (int) $user['id'], $message, $price, $days]
);
```

```
flash('Отклик отправлен. Заказчик увидит его в своём кабинете.', 'success');
redirect($back);
```

№ и проверка	Предусловия и шаги	Ожидаемый результат	Фактический результат / статус
TC-01 – Синтаксис PHP	Доступны PHP-файлы проекта и интерпретатор PHP. Запустить php -l для каждого PHP-файла.	39 файлов проходят без синтаксических ошибок.	39 из 39 файлов: ошибок синтаксиса нет. PASS
TC-02 – Синтаксис JavaScript	Установлен Node.js. Выполнить node --check hightech/assets/js/app.js.	Команда завершается без синтаксических ошибок.	Команда завершилась успешно. PASS
TC-03 – Регистрация заказчика	Настроена отдельная база; email отсутствует. Открыть регистрацию; выбрать роль заказчика; заполнить обязательные поля; отправить форму.	Создана запись пользователя; пароль хранится в виде хеша.	Не проверялось с БД. НЕ ВЫПОЛНЕН
TC-04 – Вход с верными данными	Создан активный пользователь. Открыть форму входа; ввести корректные email и пароль; отправить.	Создана сессия; доступен кабинет соответствующей роли.	Не проверялось с БД. НЕ ВЫПОЛНЕН
TC-05 – Неверный пароль	Создан активный пользователь. Ввести существующий email и неправильный пароль; отправить.	Выведена ошибка; авторизация не выполнена.	Не проверялось с БД. НЕ ВЫПОЛНЕН
TC-06 – Создание заказа	Войти как заказчик. Открыть форму заказа; заполнить название, описание и обязательные условия; сохранить.	Созданный заказ принадлежит заказчику и отображается в его кабинете.	Не проверялось с БД. НЕ ВЫПОЛНЕН
TC-07 – Отклик исполнителя	Войти как исполнитель; имеется открытый заказ. Открыть заказ; отправить допустимые стоимость, срок и сообщение.	Создан отклик, отображаемый исполнителю и заказчику.	Не проверялось с БД. НЕ ВЫПОЛНЕН
TC-08 – Повторный отклик	Исполнитель уже откликнулся на открытый заказ. Повторно отправить отклик на тот же заказ.	Создание дубликата отклоняется.	Не проверялось с БД. НЕ ВЫПОЛНЕН
TC-09 – Права доступа	Войти как исполнитель. Попытаться открыть форму создания заказа и административный раздел.	Доступ к функциям заказчика и администратора запрещён.	Не проверялось с БД. НЕ ВЫПОЛНЕН
TC-10 – Связность состояний откликов	Заказчик имеет заказ с двумя откликами. Принять один отклик, затем попытаться принять другой.	Заказ не имеет одновременно двух принятых откликов; назначение согласовано.	Ранее изолированная имитация обработчика показала возможность двух принятых откликов; необходимо повторить с БД. ТРЕБУЕТ ПОВТОРА
TC-11 – Обратная связь	Тестовая БД готова. Заполнить и отправить форму контактов; открыть административные сообщения.	В базе создано сообщение, доступное администратору.	Не проверялось с БД. НЕ ВЫПОЛНЕН
TC-12 – CSRF-защита	Пользователь вошёл в систему. Отправить POST на защищённый обработчик без корректного токена.	Изменение данных отклонено.	Не проверялось в работающем приложении. НЕ ВЫПОЛНЕН