**Министерство образования Иркутской области**

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

ПП.09.02.07-1.24.202.07

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

ПМ.02 Осуществление программных интеграций

|  |  |  |
| --- | --- | --- |
| Руководитель от предприятия:  М.П. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (С.Н. Лазутченко) |
| Руководитель от техникума: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (С.Н. Касьяненко) | |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Е.И. Емельянова) | |

Иркутск 2024

**Содержание**

[Введение 3](#_Toc162299828)

[1 Создание проекта на GitHub 4](#_Toc162299829)

[2 Техническое задание на разработку базы данных 6](#_Toc162299830)

[3 Модуль для приема, проверки и занесения данных в базу данных 7](#_Toc162299831)

[4 Отладка для программного кода 8](#_Toc162299832)

[5 Работа с инструментом Postman 10](#_Toc162299833)

[6 Настройка инструмента PHPStan 11](#_Toc162299834)

[ЗАКЛЮЧЕНИЕ 12](#_Toc162299835)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 13](#_Toc162299836)

[Приложение А – Техническое задание 14](#_Toc162299837)

# **Введение**

Производственная практика необходима для получение первичных профессиональных умений и навыков в области организации и администрирования. При этом должна быть достигнута систематизация, обобщение, закрепление и углубление теоретических знаний и умений, приобретенных при освоении основной образовательной программы.

В качестве предприятия по прохождению производственной практики выступает Иркутский релейный завод, специализирующийся на разработке и изготовлении изделий коммутационной техники.

Целью производственной практики являются следующие задачи:

1. Приобрести необходимые навыки самостоятельной работы по анализу и умению работать с данными;
2. Получить практические навыки в качестве технолога;
3. Обеспечить сбор и обработку необходимого статистического материала для выполнения практических заданий.

В процессе прохождения практики уяснили основы работ в программе Intermech IPS, работу в коллективе, так же применили полученные знания в техникуме, на практике.

Intermech IPS – универсальная система корпоративного уровня для управления информационными объектами. Так же IPS помогает объединить и управлять информацией о продукции.

# **1 Создание проекта на GitHub**

GitHub – это веб-сервис, где можно хранить программный код и управлять им. GitHub напоминает онлайн-папку для проектов, где можно работать над кодом с другими людьми, отслеживать изменения, предлагать правки и хранить историю всех изменений. GitHub облегчает совместную работу над программными проектами и управление версиями кода.

Для создания пустого проекта на GitHub нужно пройти регистрацию на официальность сайте https://github.com. Пройдя успешную регистрацию, появляется главная страница, на которой можно увидеть кнопку «Create repository» и «Import repository», рисунок 1.

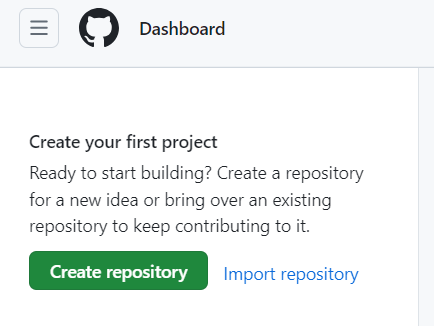


Рисунок 1 – Создание нового проекта

Нажав на кнопку «Create repository» откроется страница, на которой создаётся новый проект, рисунок 2. На данной странице нужно написать название проекта и установить приватность проекта, публичный или частный.

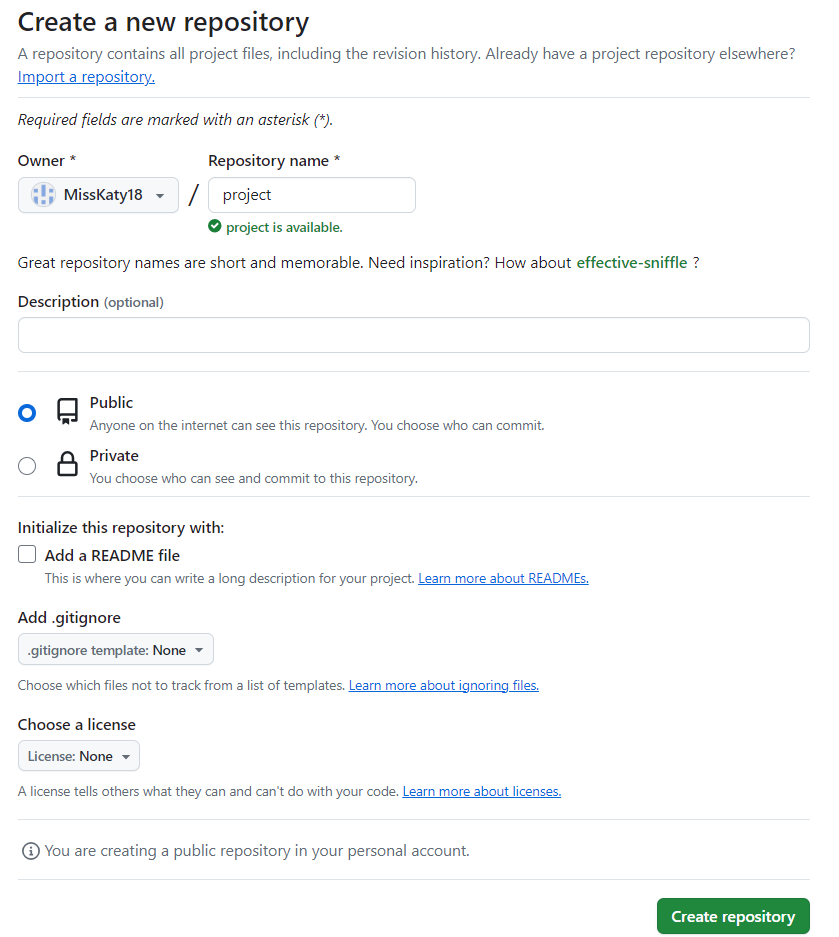


Рисунок 2 – Создание нового проекта

На рисунке 3 представлена страница созданного проекта.

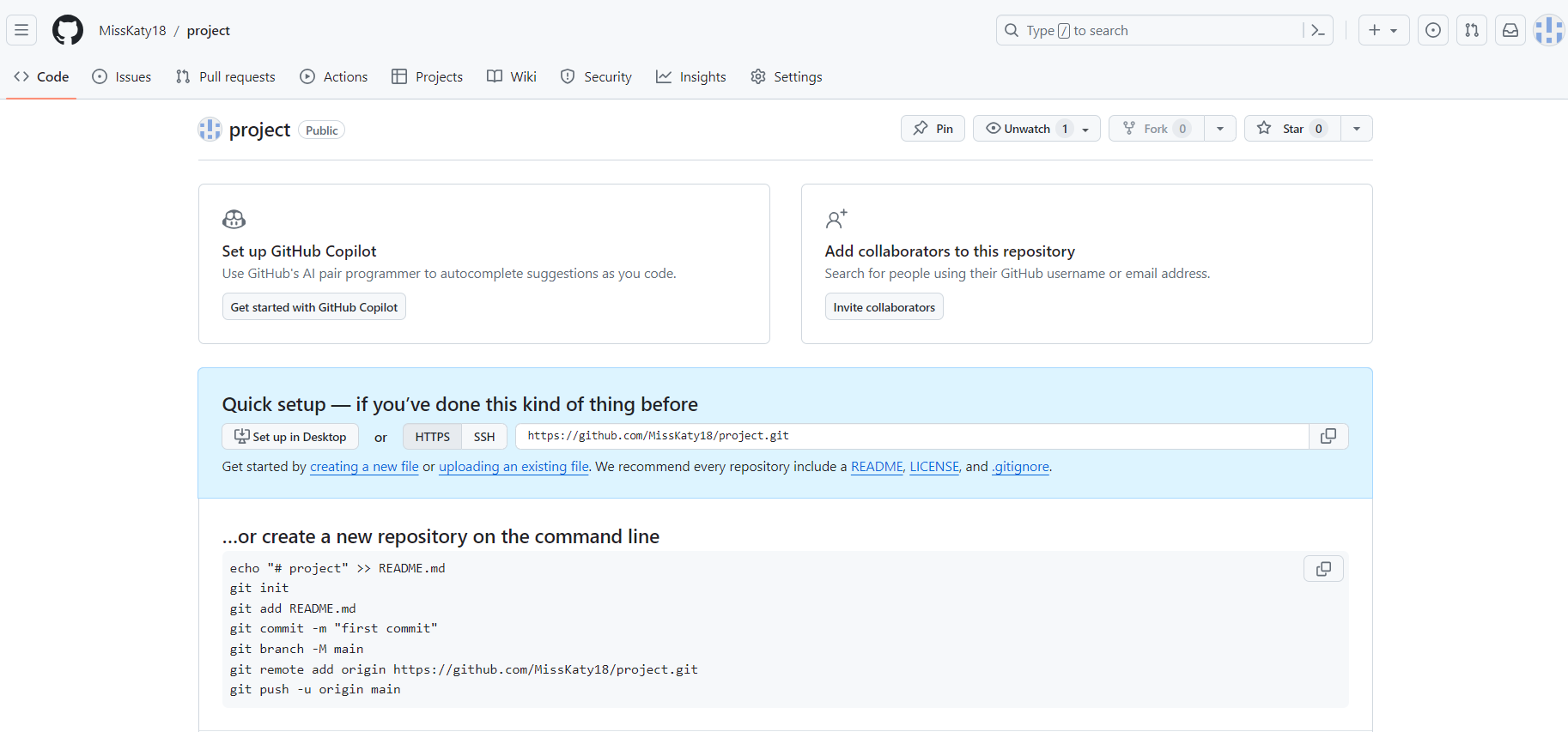


Рисунок 3 – Страница созданного проекта

Создание проекта на GitHub является отличным способом для управления версиями кода, совместной работы над проектом и обеспечения прозрачности в разработке. GitHub предоставляет возможность создания репозитория для хранения и управления исходным кодом проекта,  кроме  этого,  GitHub  позволяет  легко  отслеживать изменения в коде, вносить исправления и добавлять новый функционал.

# **2 Техническое задание на разработку базы данных**

В начале разработки базы данных было создано техническое задание, в котором были указаны основные требования к разработке.

Согласно ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Базы данных» техническое задание должно включать в себя следующие разделы:

1. Наименование проекта и цель создания базы данных.
2. Общие требования к базе данных.
3. Описание структуры данных.
4. Требования к целостности данных.
5. Требования к документации.

Описанное техническое задание на разработку базы данных представлен в приложении А.

# **3 Модуль для приема, проверки и занесения данных в базу данных**

Модуль приёма, проверки и занесения данных в базу данных является частью программного компонента, который отвечает за приём, проверку и сохранения данных. Этот модуль используется в различных приложениях для обработки информации, поступающей от пользователей или внешних систем.

Для создания модуля, который будет принимать данные извне, проверять их на корректность, а затем записывать их в базу данных, необходимо использовать PHP для обработки данных и выполнения запросов к базе данных.

Чтобы выполнить необходимые действия, нужно создать файл index.php и написать код, который представлен на рисунке 4.



Рисунок 4 – Код для работы с данными

Данный модуль играет ключевую роль в обработке информации, поступающей в систему, и обеспечивает надёжное и безопасное хранение данных для последующего использования приложением или другими системами. Он является важной частью архитектуры многих приложений и систем, где необходимо эффективно управлять данными.

# **4 Отладка для программного кода**

Отладка программного кода – это процесс нахождения и исправления ошибок в вашей программе.

Основной целью отладки программного кода является обеспечение того, чтобы программа работала корректно и выполняла поставленную задачу без ошибок. В процессе отладки используются различные инструменты, такие как отладчики и вывод ошибок на экран, это необходимо чтобы определить места возникновения ошибок и их причины.

Отладка программного кода производилась по мере написания самого кода. Для отладки использовались такие команды как: echo и переменная errMsg. На рисунке 5 представлен вызов команды echo, на рисунке 6, показано что именно выводится в echo. В данном случае, если пользователь неверно пишет название детали, то он видит ошибку, которая выводит сообщение о том, что название детали не добавлено.

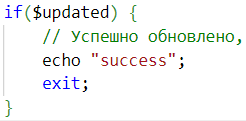


Рисунок 5 – Вызов команды echo

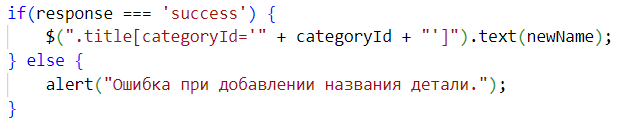


Рисунок 6 – Значение команды echo

Переменная errMsg выводит сообщения о том, что пользователь заполнил не все поля, рисунок 7.



Рисунок 7 – Переменная errMsg

Отладка программного кода также включает в себя тестирование программы на различных входных данных, ситуациях и условиях, чтобы удостовериться в ее надежности и соответствии требованиям. Этот процесс требует внимательности, систематичности и терпения, поскольку отладка может быть достаточно трудоемкой задачей, особенно при работе с крупными и сложными программами.

# **5 Работа с инструментом Postman**

Postman – это инструмент, который помогает тестировать и работать с веб-сервисами (API). С его помощью можно отправлять запросы к API, получать ответы и проверять их. Также можно создавать и сохранять запросы для повторного использования и делиться ими с другими людьми. Postman облегчает работу разработчиков при тестировании и взаимодействии с API.

Для того, чтобы проверить POST и GET запросы с помощью инструмента Postman, необходимо пройти регистрацию, либо авторизироваться, в данном случае была пройдена регистрация, чтобы можно было использовать данный инструмент для работы.

Далее на рисунке 8 представлен GET-запрос, который используется для получения данных с веб-сервера. Данные были получены в формате JSON и выведены в виде списка.



Рисунок 8 –GET-запрос

Далее на рисунке 9 представлен POST-запрос, который используется для отправки данных на сервер для обработки. Данные также были получены в формате JSON и выведены в виде списка.

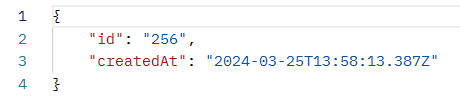


Рисунок 9 –POST-запрос

# **6 Настройка инструмента PHPStan**

PHPStan – это инструмент статического анализа PHP кода, который помогает разработчикам обнаруживать ошибки и потенциальные проблемы в коде на ранних стадиях разработки. Он работает без запуска кода, анализируя его структуру, типы переменных, вызовы методов и другие аспекты.

PHPStan позволяет проверять типы переменных, обнаруживать недостижимый код, находить ошибки в использовании методов и функций, проверять совместимость типов и многое другое. Это помогает улучшить качество кода, уменьшить количество ошибок и повысить общую надежность приложения.

PHPStan был установлен с помощью команды, которая представлена на рисунке 10, данная команда была прописана в среде разработки Visual Studio Code.



Рисунок 10 – Команда установки PHPStan

После установки PHPStan, нужно его запустить с помощью команды, представленной на рисунке 11. Прописав указанную команду ниже, сразу проводится как установка, так и настройка PHPStan, поэтому больше ничего прописывать и делать не нужно.

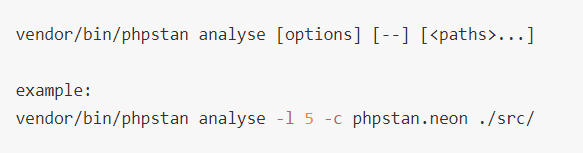


Рисунок 11 – Команда запуска PHPStan

PHPStan является мощным инструментом для повышения качества и надежности PHP кода, который помогает выявлять проблемы на ранних этапах разработки и упрощает процесс отладки и поддержки приложения.

# **ЗАКЛЮЧЕНИЕ**

Практика по осуществлению программных интеграций является важным аспектом современной разработки программного обеспечения. Важно иметь хорошее понимание того, как интегрировать различные системы, сервисы и приложения для обеспечения эффективной работы целой системы.

В процессе практики по программным интеграциям были выполнены такие задачи как:

1. Создание проекта на GitHub.
2. Разработка технического задания.
3. Написание модуля для приёма данных, проверки на корректность и запись в базу данных.
4. Отладка программного кода.
5. Работа с Postman, а именно проверка GET и POST запросов.
6. Настройка PHPStan.

Проведение практических упражнений по программным интеграциям помогло понять сложность и особенность этого процесса, и выполнить интеграции более эффективно. Кроме этого, необходимо помнить о важности тестирования, безопасности, мониторинга и документации во время работы над интеграциями, чтобы сделать систему стабильной, безопасной и легко поддерживаемой.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. habr/ Статический анализ. PHPStan/. URL: https://habr.com/ru/articles/721340/ – (дата обращения: 25.03.2024) – Текст: электронный.
2. gb/ Как тестировать API, или Postman для чайников/. URL: https://gb.ru/blog/kak-testirovat-api-postman/ – (дата обращения: 25.03.2024) – Текст: электронный.
3. cntd/Межгосударственный стандарт информационная технология /. URL: https://docs.cntd.ru/document/1200006924 – (дата обращения: 25.03.2024) – Текст: электронный.
4. skillbox/Инструкция: заливаем проект на GitHub без командной строки/. URL: https://skillbox.ru/media/code/instruktsiya\_zalivaem\_proekt\_na\_github\_bez\_komandnoy\_stroki/– (дата обращения: 25.03.2024) – Текст: электронный.
5. skillbox./Postman: что это такое и как им пользоваться/. URL: https://skillbox.ru/media/code/postman-chto-eto-takoe-i-kak-im-polzovatsya/ – (дата обращения: 25.03.2024) – Текст: электронный.

**Приложение А – Техническое задание**

Министерство образования Иркутской области

Государственное бюджетное профессиональное образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**РАЗРАБОТКА БАЗЫ ДАННЫХ**

|  |  |  |
| --- | --- | --- |
| Руководитель от предприятия:  М.П. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (С.Н. Лазутченко) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (С.Н. Касьяненко) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Е.И. Емельянова) |

Иркутск 2023

**1 Наименование проекта и цель создания базы данных**

Наименование проекта: Библиотека.

Цель создания проекта: Автоматизация документооборота.

Исполнитель: студент Иркутского авиационного техникума группы БД-20-2, Емельянова Е.И.

Разработка базы данных проходит в рамках производственной практики.

Срок разработки базы данных с 15.03.24 по 04.04.24 года.

**2 Общие требования к базе данных.**

К общим требованиям базы данных можно отнести нормализацию и целостность данных, производительность, то есть оптимизацию запросов, выбор правильных типов данных и установка индексов, кроме этого, необходимо постоянно делать резервное копирование и в случае какой-либо ошибки была возможность восстановить утерянные данные.

**3 Описание структуры данных.**

Для библиотечной структуры базы данных изначально необходимо создать три таблицы:

1. Таблица «Работники»

– employeeid (идентификатор работника).

– name (имя работника).

– position (должность работника).

* partid (внешний ключ по идентификатору детали, за каждым работником закреплена определённая деталь).

– departmentid (внешний ключ по идентификатору отдела, к которому принадлежит работник).

1. Таблица «Отделы»

* departmentid (идентификатор отдела).
* name (название отдела).

1. Таблица «Детали»

* partid (идентификатор детали).
* name (название детали).
* description (описание детали).
* quantity (количество на складе).

**4.Требования к целостности данных.**

К требованиям целостности относятся:

1. Уникальность данных: каждый работник и каждая деталь должны иметь свой уникальный номер.
2. Целостность связей: связи между таблицами должны быть корректными.
3. Соблюдение ограничений целостности: на определенное поле можно поставить ограничение, например, поле с должностью работника не должно быть пустым.

**5 Требования к документации**

Основным документом, регламентирующим использование базы данных, является руководство пользователя.

Основным документом, регламентирующим разработку базы данных, является техническое задание.