**Минобрнауки России**

**«Юго-Западный государственный университет»**

Факультет Фундаментальной и прикладной информатики

Кафедра вычислительной техники

09.03.02 Информационные системы и технологии, направленность

(профиль) "Информационные технологии в бизнесе"

(код, наименование ОП ВО: направление подготовки, направленность (профиль))

**ОТЧЕТ**

о производственной технологической (проектно-технологической) практике на предприятии ООО “НОРБИТ”

Автор

студент 3 курса, группы ИС–91б Романов Г.А.

Руководитель практики от Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_

предприятия, организации,

учреждения

Директор филиала, Жиляев Д.Л.

должность, фамилия, и.о. подпись, дата

Руководитель практики от Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_

университета

к.т.н., доцент кафедры ВТ

должность, звание, степень

Лапина Т.И.

фамилия, и.о. подпись, дата

Члены комиссии Киселёв А.В.\_\_\_\_\_\_\_

подпись, дата фамилия, и.о.

Бобынцев Д.О.\_\_\_\_\_\_

подпись, дата фамилия, и.о.

Курск, 2022

РЕФЕРАТ

Отчет по производственной технологической (проектно-технологической) практике состоит из 4 глав, введения, заключения и приложения общим объемом 44 страницы. В работе использовано 9 источников литературы.

Целью производственной технологическая (проектно-технологическая) практики является систематизация и закрепление профессиональных умений и опыта профессиональной деятельности, получение или совершенствование практических навыков в области анализа, проектирования, реализации и эксплуатации информационных систем различного назначения в условиях реальных производств.

Заданием производственной технологической (проектно-технологической) является разработка корпоративного портала учёта рабочего времени.

Для выполнения задания практики должны быть решены задачи:

* Проведён анализ предприятия ООО «Норбит» и его структуры.
* Сформулированы задачи автоматизации учёта рабочего времени.
* Созданы требования к проектируемой ИС.
* Разработано web-приложение для учёта рабочего времени.

Предметом исследования является использование фреймворка Vue.js, HTML и Javascript для разработки web-приложения.

СОДЕРЖАНИЕ

[ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ 4](#_Toc106544669)

[ВВЕДЕНИЕ 5](#_Toc106544670)

[1 Анализ деятельности предприятия 6](#_Toc106544671)

[1.1Описание компании 6](#_Toc106544672)

[1.2 Организационная структура предприятия 9](#_Toc106544673)

[2 Подходы к автоматизации учетной работы на предприятии 11](#_Toc106544674)

[2.1 Анализ существующих проектных решений 11](#_Toc106544675)

[2.2 Обоснование разработки системы «учёта рабочего времени» 14](#_Toc106544676)

[3 Постановка задачи автоматизации функции учёта рабочего времени 15](#_Toc106544677)

[3.1 Требование к созданию функции учёта рабочего времени 15](#_Toc106544678)

[3.2 Обоснования выбора инструментальных средств программирования 17](#_Toc106544679)

[4.Разработка подходов к автоматизации функции учёта рабочего времени 21](#_Toc106544680)

[4.1 Разработка web-приложения для учёта рабочего времени 21](#_Toc106544681)

[4.2 Описание клиентского приложения 22](#_Toc106544682)

[ЗАКЛЮЧЕНИЕ 25](#_Toc106544683)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 26](#_Toc106544684)

[ПРИЛОЖЕНИЕ 1 27](#_Toc106544685)

[ЛИСТИНГ КОДА 35](#_Toc106544686)

# ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ИС – информационная система

IT – информационная технология

ERP – система управления предприятием

CRM – система управления взаимоотношениями с клиентами

BI – бизнес-аналитика

NBT – платформа управления закупками

SRM – система управления взаимодействием с поставщиками

HR – человеческие ресурсы

ПО – программное обеспечение

ВВЕДЕНИЕ

Ключевые слова: информационная система, проектирование ИС, эффективность, автоматизированные информационные системы.

Целью производственной технологической (проектно-технологической) практики является разработка корпоративного портала учёта рабочего времени для оптимизации занятости сотрудника на предприятии посредством интегрированной среды разработки Visual Studio 2019 и фреймворка Vue.js.

Объект исследования бизнес-процессы организации учетной работы на предприятии.

Предмет исследования: методы проектирования программного обеспечения ИС и баз данных.

Для достижения данной цели решены следующие основные задачи:

* проведен анализ предметной области и сформулированы задачи автоматизации учета рабочего времени;
* сформулированы требования к проектируемой ИС;
* выполнено обоснование выбора инструментальных средств разработки;
* разработало web-приложение, реализующее функции проектируемой ИС.

Планируемым результатом выполнения задания производственной технологической (проектно-технологической) практики является разработанный корпоративный портал учёта рабочего времени для оптимизации занятости сотрудника на предприятии.

# 1 Анализ деятельности предприятия

НОРБИТ (входит в группу компаний ЛАНИТ) — ведущий эксперт на рынке IT-консалтинга по разработке и внедрению эффективных решений для автоматизации бизнеса. Компания с 2000 года работает в области информационных технологий и обладает высшим партнерским статусом мировых вендоров.

# Описание компании

НОРБИТ представляет одну из ведущих экспертиз по разработке и внедрению ERP, CRM и BI–систем, а также является разработчиком торгово-закупочных систем на базе собственной платформы NBT.

Эксперты НОРБИТ обладают богатым проектным опытом в различных отраслях и успешно завершили более 500 проектов по внедрению систем для оптимизации бизнес-процессов в компаниях различного масштаба. В структуре холдинга ЛАНИТ НОРБИТ является Центром Компетенции по разработке и внедрению интегрированных систем управления предприятием на базе платформ SAP, Microsoft Dynamics (Dynamics 365), Terrasoft, QlikView и 1С, а также является лидирующим экспертом и разработчиком собственной платформы NBT для реализации систем по автоматизации закупочной деятельности.

Компания НОРБИТ с 2001 года специализируется на разработке и внедрении эффективных решений для автоматизации бизнеса и государственного управления. НОРБИТ обладает высшими партнерскими статусами отечественных и мировых вендоров SAP, Microsoft, Террасофт, 1С, 1С-Битрикс, ОМП (ОС «Аврора») и QlikView. Компания входит в ГК ЛАНИТ.

НОРБИТ занимается разработкой и внедрением ERP-, CRM-, SRM-, HR- и BI-систем, мобильных и веб-приложений, систем финансового планирования и бюджетирования, а также предоставляет услуги управленческого и IT-консалтинга.

Основные направления:

* Внедрение инновационных решений. НОРБИТ занимается разработкой и внедрением корпоративных финансовых сервисов, платежных систем, мобильных и веб-приложений с использованием технологий машинного обучения, компьютерного зрения, технологий распределенного реестра. Технологии, используемые НОРБИТ, включены в Реестр отечественного ПО и соответствуют требованиям импортозамещения.
* Внедрение ERP-систем. НОРБИТ обладает 20-летним опытом внедрения ERP-систем на базе ведущих платформ: SAP, Microsoft и 1С. Клиентами НОРБИТ стали более 100 компаний из сфер энергетики, оптовой и розничной торговли, промышленности, логистики и транспорта, строительства и других отраслей.
* Внедрение CRM-системы. НОРБИТ представляет одну из крупнейших и опытных практик внедрения CRM-систем в России и странах СНГ. Компания занимается разработкой и внедрением CRM-решений на базе Террасофт, 1С-Битрикс, Microsoft. НОРБИТ реализовал более 300 проектов в компаниях финансового сектора, производства, торговли, строительства, медиа, спорта и др. отраслях.
* Внедрение SRM-систем. НОРБИТ занимается разработкой и внедрением систем и сервисов управления закупками и взаимодействия с поставщиками в государственных и коммерческих компаниях как на базе собственной платформы NBT (включена в Реестр отечественного ПО), так и на основе решений сторонних вендоров – SAP, Террасофт, 1С.
* Внедрение HR-систем. НОРБИТ представляет одну из крупнейших в России практик по разработке и внедрению HRM-систем на базе платформы SAP SuccessFactors и WFM-систем на основе отечественной платформы Goodt Go.
* Внедрение BI-систем. НОРБИТ внедряет BI-решения, направленные на глубокое исследование данных, выявление тенденций и визуализацию результатов анализа. Решения НОРБИТ основаны на продуктах ведущих вендоров – QlikView, SAP BI, Microsoft Power BI, IBM Cognos.
* Управленческий и IT-консалтинг. Компания НОРБИТ предоставляет комплекс консультационных услуг в области построения финансовых структур, систем финансового планирования, бюджетирования и управленческого учета, оптимизации бизнес-процессов, разработки IT-стратегий.

# 1.2 Организационная структура предприятия

Компания НОРБИТ входит в состав Ланит – ведущей многопрофильной группе IT-компаний. Она представляет собой многопрофильную группу IT-компаний, лидер российской отрасли информационных технологий и партнер более двухсот пятидесяти мировых производителей оборудования и программных решений. Компании группы предоставляют полный комплекс IT-услуг, который расширяется за счет освоения передовых и востребованных технологий.

Сама компания состоит из множества департаментов, которые покрывают весь спектр IT- технологий. В настоящее время существуют такие департаменты:

Департамент SAP - Одна из крупнейших проектных экспертиз в розничной и оптовой торговле — собственная методика проектов, успешно отработанная на практике.

Департамент информационных технологий - Системная поддержка осуществления бизнес - процессов.

Департамент машинного обучения - Департамент работает в сфере разработки и внедрения систем с использованием технологий машинного обучения.

Департамент ОМНИ - Департамент занимается проектами в сфере автоматизации программ лояльности и маркетинга, управления потребительским поведением с применением технологий машинного обучения.

Департамент SRM - Разработка и внедрение систем и сервисов управления закупками и взаимодействия с поставщиками в государственных и коммерческих компаниях.

Департамент AX и 1C - Разработка и внедрение эксклюзивных ERP-решений с учетом специфики бизнес-задач, а также разработка мобильных приложений.

Департамент CRM - Самая крупная и опытная практика CRM в России и странах СНГ.

Департамент инновационных систем - Разработка и внедрение финансовых сервисов, платежных систем, мобильных приложений на платформе ОС Аврора.

Департамент управленческого консалтинга - Экспертиза по развитию и повышению эффективности систем управления компаниями в различных сферах.

Департамент продуктовой разработки - Департамент продуктовой разработки.

2 Подходы к автоматизации учетной работы на предприятии

# 2.1 Анализ существующих проектных решений

[Kickidler](https://www.kickidler.com/ru/) – система учета рабочего времени нового поколения. Данный продукт акцентирован на мониторинге активности пользователей, а также крайне прост в использовании в отличие от многих своих конкурентов.

Преимущества программы Kickidler:

* Онлайн-мониторинг неограниченного числа компьютеров. Программа позволяет руководителю видеть в реальном времени сетку экранов всех компьютеров в сети с частотой обновления 1 кадр в секунду. Иногда этого достаточно для контроля сотрудников.
* Запись видео с экранов. Программа фиксирует все действия пользователя на рабочем компьютере, что особенно важно при расследовании инцидентов. Есть возможность поиска нужного фрагмента видео по ключевым событиям.
* Кроссплатформенность. Программа полноценно работает с Windows, Linux и MacOS, причем даже с последней 15 версией Mac.
* Простота в установке и настройке. Многие пользователи отмечают юзер френдли интерфейс программы и интуитивность ее использования.
* Возможность скрытой и открытой установки агента программы.
* Полноценная оффлайн-версия программы.
* Относительно небольшая цена и гибкая ценовая политика. Покупая лицензию на 3 года, стоимость одного рабочего места будет 3$.
* Бесплатная версия программы с ограничением до 6 юзеров.
* Программа полностью переведена на русский, английский, испанский и португальский языки.
* Помимо перечисленных уникальных функций, программа обладает инструментарием по учету рабочего времени, анализу продуктивности сотрудников, а также функциями кейлоггера и удаленного управления.

Алгоритм работы Kickidler:

Система Kickidler состоит из 4-х компонентов: Вьюера, Граббера, Сервера и Веб-интерфейса. Вьюер устанавливается на ПК наблюдающего, Граббер – на ПК того, за кем будут наблюдать, на Сервере хранятся все данные, а в Веб-интерфейсе производятся настройки программы.

Недостатки:

К минусам программы можно отнести отсутствие облачной и мобильной версий. Также программа не имеет интеграций с другим ПО.

Предназначение:

Для Интернет-магазинов и компаний, работающих с денежными операциями. Запись видео с экранов – незаменимая фича для отслеживания всех действий за ПК. Также программа хороша для малого бизнеса, поскольку не имеет нижнего порога по количеству проданных лицензий.

[Activtrak](https://activtrak.com/) – это облачный сервис для мониторинга сотрудников, позволяющий анализировать ежедневную активность и поведение персонала за компьютерами. Программа ActivTrak развивается стремительными темпами. По информации из разных источников, в 2019 году на ее развитие было [выделено](https://news.rambler.ru/internet/41947602-startap-aktivtrak-s-kornyami-iz-rossii-privlek-20-mln-eto-servis-po-kontrolyu-za-rabotnikami/) 20 млн. долларов.

Преимущества программы ActivTrak:

* Мониторинг активности сотрудников. Скриншоты экрана с частотой обновления 1 раз в 5 секунд.
* Видеозапись экранов, экспорт видео.
* Автоматические уведомления о тревогах.
* Сигналы тревоги по типам скриншотов с гибкой настройкой.
* Возможность мониторить корпоративную почту сотрудников.
* Множество отчетов по рабочему времени.
* Полностью облачная версия.
* Есть приложения для iOS и Android
* Блокировка нежелательных сайтов и приложений.

Алгоритм работы:

Для установки программы ActivTrak необходимо просто создать аккаунт и установить агенты на компьютеры, которые вы собираетесь мониторить. Серверная часть программы автоматически устанавливается на облако.

Недостатки:

* Программа не блокирует поддомены отдельно от сайтов
* Невозможно регулировать частоту съемки скриншотов
* Невозможность установки агента программы на Mac
* Сложный процесс деинсталляции программы

Предназначение:

Программа подойдет для любого вида среднего и крупного бизнеса, но больше всего – из финансово-технической отрасли.

# 2.2 Обоснование разработки системы «учёта рабочего времени»

У существующих аналогов информационной системы были выявлены недостатки:

* Отсутствие облачной версии
* Отсутствие мобильной версии
* Отсутствие интеграций с другим ПО
* Отсутствие возможности блокировки поддоменов сайтов
* Невозможно регулировать частоту съемки скриншотов
* Невозможность установки агента программы на Mac
* Сложный процесс деинсталляции программы
* Высокая стоимость для больших предприятий

Решением выше указанных проблем является разработка информационной технологии, которая отвечает заданным условиям:

* Иметь мультиплатформенность
* Возможность интеграции с ПО компании
* Возможность полноценной блокировки развлекательных сайтов
* Возможность полноценной настройки отслеживания действий пользователя на ПК
* Низкая стоимость разработки и использования

Создание собственной ИС позволит учесть все особенности учета рабочего времени на предприятии. В теоретическом представлении ИС будет использоваться сотрудниками предприятиями, имеющими руководящие должности. Разработка является основой для дальнейшего настраивания под нужды каждого предприятия.

# 3 Постановка задачи автоматизации функции учёта рабочего времени

# 3.1 Требование к созданию функции учёта рабочего времени

Необходимо разработать web-приложение на базе JavaScript-Фреймворка VUE.JS для учета рабочего времени.

Система должна обеспечивать:

Учет проектов:

Пользователь должен иметь возможность видеть список проектов, добавлять, удалять, изменять проекты.

Атрибуты проекта:

* Название (Текстовое однострочное поле)
* Код (Текстовое однострочное поле)
* Активный (Да\Нет)

Учет задач:

Пользователь должен иметь возможность видеть список задач, добавлять, удалять, изменять задачу.

* Атрибуты задачи:
* Название (Текстовое однострочное поле)
* Проект (ссылка на проект)
* Активный (Да\Нет)

Учет проводок:

Пользователь должен иметь возможность вводить время, потраченное на задачу.

Атрибуты проводки:

* Дата проводки (дата без учёта времени)
* Количество часов (положительное число, не более 24)
* Описание (текстовое однострочное поле)
* Задача

Пользователю запрещено вводить за один день суммарно проводок общей длительностью более 24 часов.

Пользователю должен иметь возможность при заведении новой проводки указать задачу. Неактивные задачи нельзя выбрать.

Если задача стала неактивна после заведения проводки, запретить редактирования поля "Задача" для данной проводки.

Пользователь должен иметь возможность вводить более одной проводки для одной и той же задачи.

Пользователь должен иметь возможность видеть список всех проводок:

* за все время
* за конкретный день недели (выбор из календаря)
* за месяц

Необходимо реализовать визуализацию:

* Если сумма часов за выбранный день менее 8, отображать желтый стикер (показывает, что внес недостаточно).
* Если сумма часов за выбранный день равно 8, отображать зеленый стикер (показывает, что внес достаточно).
* Если сумма часов за выбранный день более 8, отображать красный стикер (показывает, что внес избыточно).

# 3.2 Обоснования выбора инструментальных средств программирования

Для реализации веб-приложения было использована среда разработки Visual Studio 2019. Она имеет понятный интерфейс и возможности для разработки различных приложений.

В качестве фреймворка для создания веб-приложения был использован Vue.js. Он представляет собой современный прогрессивный фреймворк, написанный на языке JavaScript и предназначенный для создания веб-приложений клиентского уровня. Основная сфера применения данного фреймворка — это создание и организация пользовательского интерфейса.

Первый релиз фреймворка увидел свет в феврале 2014 года. Его создателем является Эван Ю (Evan You), который до этого работал в Google над AngularJS. С тех пор фреймфорк динамично развивается, его текущей версией является версия 3.0.

Vue.js имеет довольно небольшой размер - не более 20 кБ, и при этом обладает хорошей производительностью по сравнению с такими фреймворками как Angular или React. Поэтому неудивительно, что данный фреймворк в последнее время набирает обороты и становится все более популярным.

Одним из ключевых моментов в работе Vue.js является виртуальный DOM. Структура веб-страницы, как правило, описывается с помощью DOM (Document Object Model), которая представляет организацию элементов html на странице. Для взаимодействия с DOM (добавления, изменения, удаления html-элементов) применяется JavaScript. Но когда мы пытаемся манипулировать html-элементами с помощью JavaScript, то мы можем столкнуться со снижением производительности, особенно при изменении большого количества элементов. А операции над элементами могут занять некоторое время, что неизбежно скажется на пользовательском опыте. Однако если бы мы работали из кода js с объектами JavaScript, то операции производились бы быстрее.

Для этого Vue.js использует виртуальный DOM. Виртуальный DOM представляет легковесную копию обычного DOM. Если приложению нужно узнать информацию о состоянии элементов, то происходит обращение к виртуальному DOM. Если данные, которые используются в приложении Vue.js, изменяются, то изменения вначале вносятся в виртуальный DOM. Потом Vue выбирает минимальный набор компонентов, для которых надо выполнить изменения на веб-странице, чтобы реальный DOM соответствовал виртуальному. Благодаря виртуальному DOM повышается производительность приложения.

Vue.js поддерживается всеми браузерами, которые совместимы с ECMAScript 5. На данный момент это все современные браузеры, в том числе IE11.

Для работы с фреймворком Vue.js потребуются знания HTML, CSS и языка программирования java script.

HTML — стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

Элементы HTML являются строительными блоками HTML страниц. С помощью HTML разные конструкции, изображения и другие объекты, такие как интерактивная веб-форма, могут быть встроены в отображаемую страницу. HTML предоставляет средства для создания заголовков, абзацев, списков, ссылок, цитат и других элементов. Элементы HTML выделяются тегами, записанными с использованием угловых скобок. Такие теги, как <img /> и <input />, напрямую вводят контент на страницу. Другие теги, такие как <p>, окружают и оформляют текст внутри себя и могут включать другие теги в качестве подэлементов. Браузеры не отображают HTML-теги, но используют их для интерпретации содержимого страницы.

В HTML можно встроить программный код на языке программирования [JavaScript](https://ru.wikipedia.org/wiki/JavaScript" \o "JavaScript), для управления поведением и содержанием веб-страниц. Также включение CSS в HTML описывает внешний вид и макет страницы.

JavaScript (JS) это язык программирования, который в основном используется для создания динамических скриптов на веб-страницах, но он так же часто применяется на стороне [сервера](https://developer.mozilla.org/ru/docs/Glossary/Server), используя пакеты, такие как [Node.JS](https://nodejs.org/).

JavaScript не следует путать с [Java](https://ru.wikipedia.org/wiki/Java" \o "Java). «Java» и «JavaScript» являются товарными знаками или зарегистрированными товарными знаками Oracle в США и других странах. Тем не менее, два языка программирования существенно различаются по синтаксису, семантике и использованию.

Задуманный Бренданом Эйхом (тогда работал в Netscape Corporation) как серверный язык, Javascript вскоре появился в Netscape Navigator 2.0 в сентябре 1995 года. JavaScript сразу же добился успеха и [Internet Explorer 3.0](https://developer.mozilla.org/ru/docs/Glossary/Microsoft_Internet_Explorer) представил поддержку JavaScript под названием JScript в августе 1996 года.

В ноябре 1996 года Netscape начала сотрудничать с ECMA International, чтобы сделать JavaScript отраслевым стандартом. С тех пор стандартизированный JavaScript называется ECMAScript и изложен в ECMA-262, чьё последнее (девятое, ES2018) издание доступно с июня 2018 года.

JavaScript в основном используется в браузере, что позволяет разработчикам манипулировать содержимым веб-страницы с помощью [DOM](https://developer.mozilla.org/ru/docs/Glossary/DOM), данными с помощью [AJAX](https://developer.mozilla.org/ru/docs/Glossary/AJAX) и [IndexedDB](https://developer.mozilla.org/ru/docs/Glossary/IndexedDB), рисовать графику с помощью [canvas](https://developer.mozilla.org/ru/docs/Glossary/Canvas), взаимодействовать с устройством под управлением браузера через различные [APIs](https://developer.mozilla.org/ru/docs/Glossary/API) и т.д. JavaScript является одним из наиболее часто используемых языков в мире благодаря недавнему росту и повышению производительности доступных в браузерах [APIs](https://developer.mozilla.org/ru/docs/Glossary/API).

В последнее время популярность JavaScript расширилась ещё больше благодаря успешной платформе [Node.js](https://nodejs.org/) - самой популярной кроссплатформенной среде выполнения JavaScript вне браузера. Node.js позволяет разработчикам использовать JavaScript в качестве языка сценариев для автоматизации работы на ПК и создания полнофункциональных [HTTP](https://developer.mozilla.org/ru/docs/Glossary/HTTP) и [WebSockets](https://developer.mozilla.org/ru/docs/Glossary/WebSockets) серверов.

Язык разметки HTML использует CSS. CSS  — [формальный язык](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) описания внешнего вида [веб-страницы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0). CSS используется создателями [веб-страниц](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) для задания [цветов](https://ru.wikipedia.org/wiki/%D0%A6%D0%B2%D0%B5%D1%82), [шрифтов](https://ru.wikipedia.org/wiki/%D0%A8%D1%80%D0%B8%D1%84%D1%82), стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS является ограждение и отделение описания логической структуры веб-страницы от описания внешнего вида этой веб-страницы. Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

# 4.Разработка подходов к автоматизации функции учёта рабочего времени

# 4.1 Разработка web-приложения для учёта рабочего времени

Перед разработкой web-приложения на базе фреймворка Vue.js необходимо установить компоненты для работы, а именно Node.js и непосредственно сам Vue.js. Node.js можно загрузить с официального сайта и установить как обычную программу. Для установки Vue.js надо зайти в среду разработки приложения (Visual studio 2019 или Visual code), открыть «power shell для разработчиков» в Visual studio 2019 или «Терминал» в Visual code.

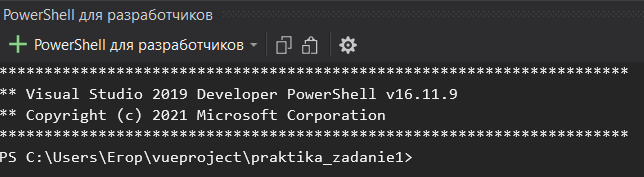


Рисунок 1 – Power shell для разработчиков

В открывшемся командной строке ввести команду «npm install vue». После этого Vue.js будет установлен на компьютер и готов к использованию.

В результате работы было создано 5 компонентов: Projectaccounting.vue, Taskaccounting.vue, Wiringaccounting.vue, Home.vue, router.js, App.vue, main.js.

Компоненты Projectaccounting.vue, Taskaccounting.vue, Wiringaccounting.vue отвечают за визуальную составляющую web-приложения. Они содержат в себе HTML-разметку с элементами javascript и стилями CSS страницы с данными о проектах, задачах и проводках соответственно.

Компонент main.js является средством запуска Vue.js и его компонентов.

Компонент Home содержит в себе HTML-разметку стартовой страницы с элементами ссылок на другие страницы.

Компонент App.vue содержит в себе массивы с данными о проектах, задачах и проводках. Имеет в себе функции для редактирования и удаления данных из массивов.

Компонент router.js содержит в себе связь ссылок на страницу и компонентов.

Для запуска виртуального сервера с сайтом необходимо в консоли написать команду «nmp run serve» — это запустит web-приложение в режиме разработчика.

# 4.2 Описание клиентского приложения

Готовое клиентское web-приложение встречает пользователей стартовым экраном. В этой форме пользователь может выбрать необходимый ему список: список проектов, список задач, список проводок.



Рисунок 2 – главная форма web-приложения

При выборе списка проектов, открывается форма со списком проектов.

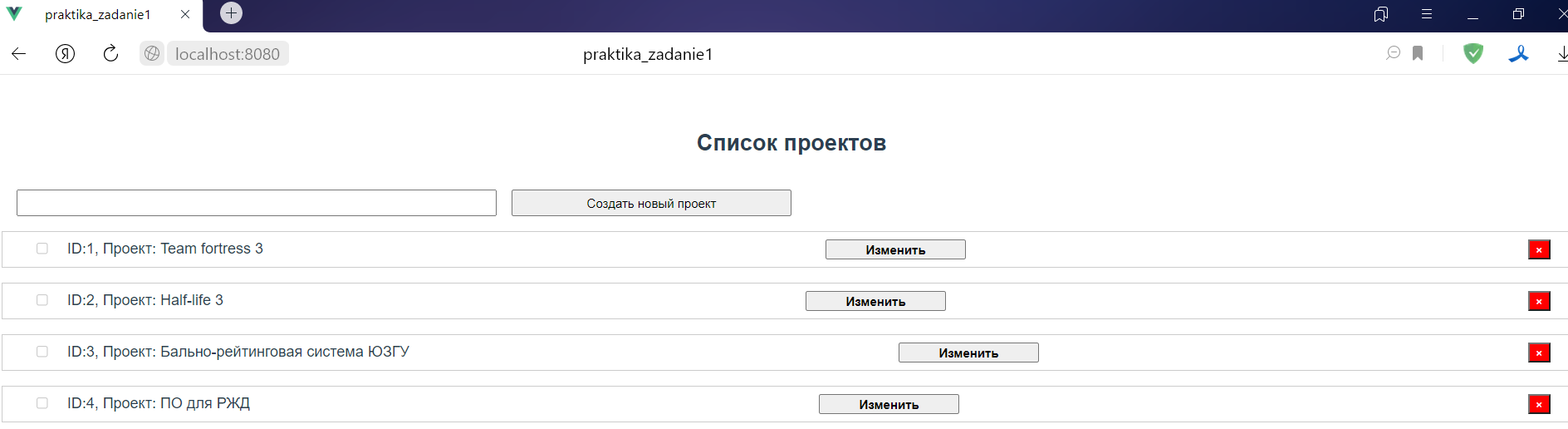


Рисунок 3 – форма «список проектов»

На этой форме пользователь может просмотреть список проектов, добавить новый, изменить или удалить имеющийся, сделать неактивным проект.

При выборе списка задач, открывается форма со списком задач.

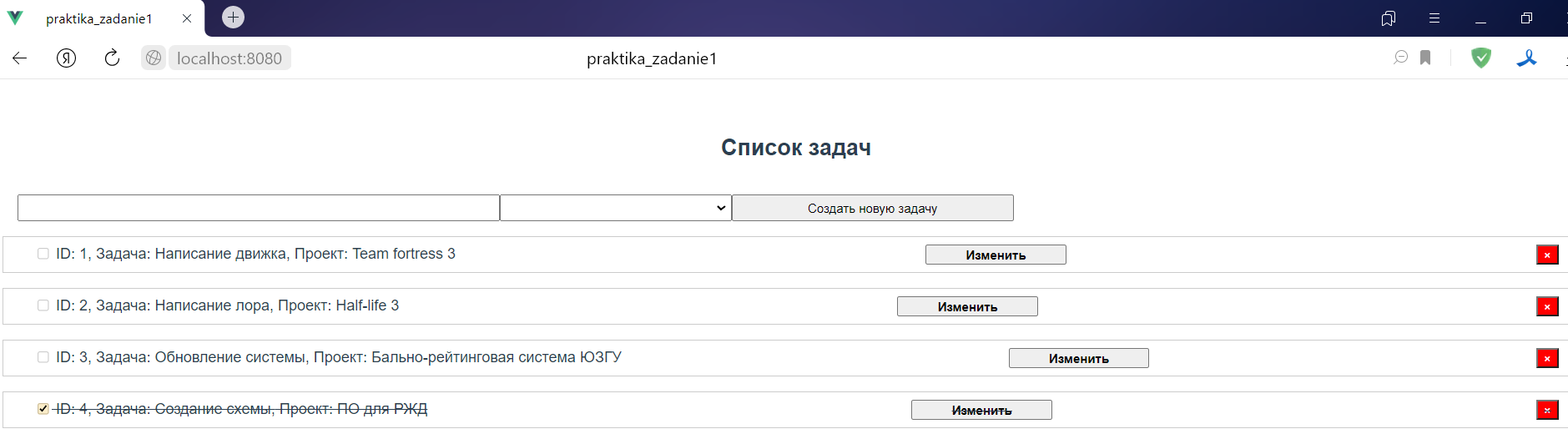


Рисунок 4 – форма «список задач»

На этой форме пользователь может просмотреть список задач, добавить новый, изменить или удалить имеющийся, сделать неактивным задачу.

При выборе списка проводок, открывается форма со списком проводок.

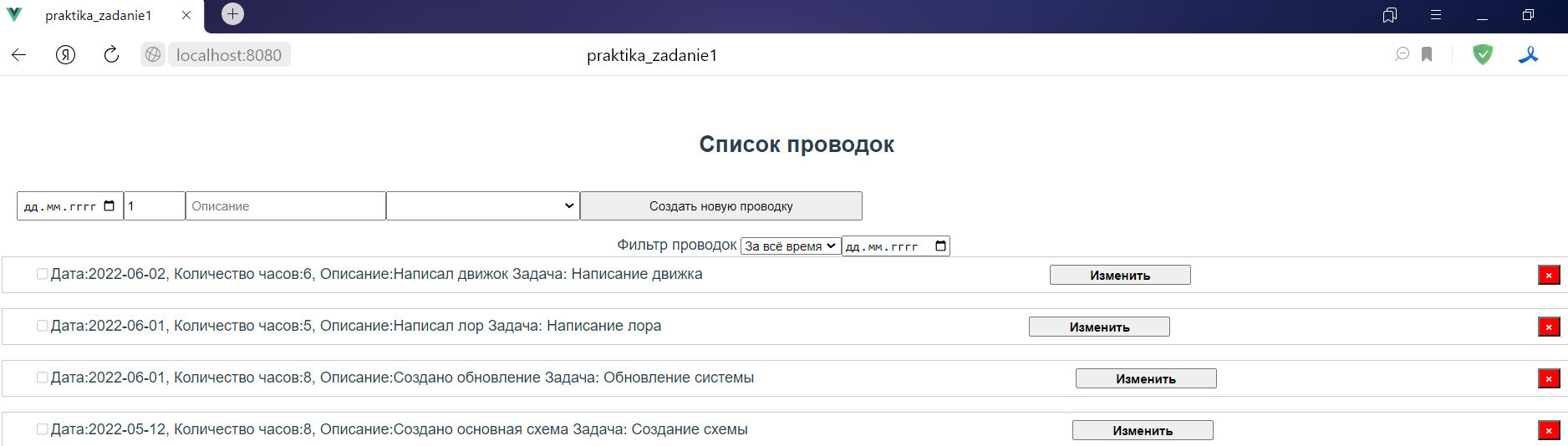


Рисунок 5 – форма «список проводок»

На этой форме пользователь может просмотреть список проводок, добавить новый, изменить или удалить имеющийся, сделать неактивным проводку.

Готовое web-приложение работает корректно.

# ЗАКЛЮЧЕНИЕ

В рамках производственной практики была разработано средство автоматизации рабочего времени для оптимизации занятости сотрудника на предприятии. Была рассмотрена компания ООО «Норбит», её структура и цели.

Для достижения этого результата были созданы требования к системе учёта рабочего времени, было создано демонстративное клиентское web-приложение.

Данная работа выполнена на языке высокого уровня программирования в среде Visual studio 2019. При написании и разработке программного обеспечения использовался фреймворк Vue.js.

Разработанное web-приложение имеет удобный пользовательский интерфейс. Все элементы web-приложения представлены как единый проект, управление которым осуществляется через окно главной формы проекта. Программа работает без ошибок, выполняет все функции.

В ходе работы проектирования интерфейса были составлены основные формы.

В результате выполнения отчёта о практики были получены основные навыки анализа заданной предметной области, разработки web-приложения и работы с фреймворком Vue.js, изучены основы HTML и javascript.

Созданный проект позволяет осуществлять ввод новых данных, изменять имеющиеся данные в базе и удалять старые. Данная программа очень удобна в использовании для систематизации данных.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. IT-интегратор, внедрение ERP и CRM, системы автоматизации закупок — Норбит [Электронный ресурс]. Режим доступа: https://www.norbit.ru/ (дата обращения 20.06.22).
2. ЛАНИТ — крупнейшая российская группа IT-компаний [Электронный ресурс]. Режим доступа: https://www.lanit.ru/ (дата обращения 20.06.22).
3. Vue.js [Электронный ресурс]. Режим доступа: https://ru.vuejs.org/ (дата обращения 20.06.22).
4. Node.js [Электронный ресурс]. Режим доступа: https://nodejs.org/en/ (дата обращения 20.06.22).
5. Проектирование информационных систем: методические указания по выполнению курсового проекта по дисциплине «Проектирование информационных систем» для студентов направления подготовки бакалавров 09.03.02 Информационные системы / Юго-Зап. гос. ун-т; сост. Т. И. Лапина. - Электрон. текстовые дан. (502 КБ). - Курск: ЮЗГУ, 2021. - 21 с
6. Лапина, Т.И. Информационные системы. Проектный практикум по выполнению ВКР [Текст]: учебное пособие/ Т. И. Лапина. – М.: ЮЗГУ, ЗАО «Университетская книга», 2016. 99с.
7. Стасышин, В. М. Проектирование информационных систем и баз данных [Электронный ресурс]: учебное пособие / В. М. Стасышин. - Новосибирск: НГТУ, 2012. - 100 с.
8. Абрамов, Г. В. Проектирование информационных систем [Электронный ресурс]: учебное пособие / Г. В. Абрамов, И. Медведкова, Л. Коробова. - Воронеж: Воронежский госууниверситет инженерных технологий, 2012. - 172 с. – Режим доступа: biblioclub.ru.
9. Золотов, С. Ю. Проектирование информационных систем [Электронный ресурс]: учебное пособие / С. Ю. Золотов. - Томск: Эль Контент, 2013. - 88 с.

# ПРИЛОЖЕНИЕ 1

**Техника безопасности при работе на персональном компьютере**

Требования к студентам, проходящим практику:

1. Студенты, вышедшие на практику, допускается к выполнению работы только при наличии установленного набора документов (дневник, направление, индивидуальное задание), а также наличии отметки о прохождении инструктажа по охране труда, пожарной безопасности  
и обучения навыкам оказанию доврачебной помощи пострадавшим от несчастных случаев.

2. Инструктаж проводится руководителями практики по направляющим кафедрам института. Проведение всех видов инструктажей должно фиксироваться в протоколе организационного собрания по вопросам прохождения всех видов практик, с обязательными подписями получившего и проводившего инструктаж в контрольном листе, которые хранятся на кафедре.

3. Каждый инструктаж студентов, выходящих на практику, должен заканчиваться обязательной проверкой знаний в виде устного опроса.

4. При прибытии на место прохождения практики студент должен пройти вводный и первичный инструктаж по охране труда и пожарной безопасности на данном рабочем месте.

5. Каждый студент, приступающий к практике на рабочем месте, должен знать:

* план эвакуации при пожаре и в случае ЧС
* правила действий при возникновении пожара
* место расположения первичных средств пожаротушения и правила их применения
* место хранения медицинской аптечки

6. Студенту, проходящему практику, следует:

* знать и соблюдать внутренний трудовой распорядок
* соблюдать режим труда и отдыха на рабочем месте
* иметь опрятный внешний вид в соответствии с требованиями делового этикета
* обращать внимание на знаки безопасности, сигналы и выполнять их требования
* ходить в помещениях спокойным шагом и не подниматься, и не спускаться бегом по лестничным маршам

7. Студенту, проходящему практику запрещается приступать к работе в состоянии алкогольного или наркотического опьянения.

8. Работа студентов при прохождении практики может сопровождаться наличием следующих опасных и вредных производственных факторов:

* работа с офисной техникой (компьютер, принтер, сканер и прочие виды офисной техники) - ограничение двигательной активности, монотонность и значительное зрительное напряжение - поражение электрическим током
* использование бытовых электроприборов (чайник, кофеварка и прочая бытовая техника) - поражение электрическим током, ожоги
* пользование электроосвещением (потолочные светильники, настольные лампы) - поражение электрическим током
* использование стремянок и лестниц - падение с высоты
* вне рабочего места (по пути следования к месту практики и обратно) - движущиеся автомобили и прочие виды транспорта, неудовлетворительное состояние дорожного покрытия (гололед, неровности дороги и пр.) - получение травмы в ДТП, получение травмы при падении

9. Студенты, проходящие практику, несут ответственность за производственный травматизм и аварии, которые произошли по их вине в связи с выполняемой ими работой в соответствии с действующим законодательством и требованиями норм и правил охраны труда и пожарной безопасности на рабочем месте.

Требования техники безопасности перед началом работы:

1. Прибыть на рабочее место заблаговременно для исключения спешки и, как правило, падения и получения травмы.

2. Очистить экран дисплея персонального компьютера от пыли. Отрегулировать высоту и угол наклона экрана. Экран должен находиться ниже уровня глаз на 5 град, и располагаться в  
прямой плоскости или с наклоном на оператора (15 град.). Расстояние от глаз оператора до экрана должно быть в пределах 60 - 80см.

3. Отрегулировать уровень освещенности рабочего места. Местный источник света по отношению к рабочему месту должен располагаться таким образом, чтобы исключить попадание в глаза прямого света, и должен обеспечивать равномерную освещенность на поверхности  
40 х 40см, не создавать слепящих бликов на клавиатуре и других частях пульта, а также на экране видеотерминала в направлении глаз работника.

4. Отрегулировать кресло по высоте.

5. Осмотреть рабочее место и оборудование. Проверить оснащенность рабочего места необходимым для работы оборудованием, инвентарем, приспособлениями и инструментами. Убрать все лишние предметы.

6. Проверить визуально исправность оборудования.

7. О замеченных недостатках и неисправностях немедленно сообщить руководителю практики и до устранения неполадок и разрешения руководителя к работе не приступать.

Требования техники безопасности во время работы:

1. Для снижения зрительного и общего утомления после каждого часа работы за компьютером необходимо делать 15 минутный перерыв с выполнением физических упражнений.

2. Необходимо в течение всего рабочего дня содержать в порядке и чистоте рабочее место. В течение рабочей смены экран дисплея должен быть не менее одного раза очищен от пыли. Своевременно убирать с пола рассыпанные материалы, принадлежности, продукты, разлитую воду и пр.

3. Во время работы запрещается:

* прикасаться к задней панели системного блока (процессора) при включенном питании
* производить переключение разъемов интерфейсных кабелей периферийных устройств, при включенном питании
* загромождать верхние панели устройств бумагами и посторонними предметами
* допускать захламленность рабочего места
* производить отключение питания во время выполнения активной задачи
* допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и др. устройств
* включать сильно охлажденное (принесенное с улицы в зимнее время) оборудование
* производить самостоятельно вскрытие и ремонт оборудования

3. При работе с электроприборами и оргтехникой (персональные компьютеры, принтеры, сканеры, копировальные аппараты, факсы, бытовые электроприборы, приборы освещения).

4. Автоматические выключатели и электрические предохранители должны быть всегда исправны.

5. Изоляция электропроводки, электроприборов, выключателей, штепсельных розеток, ламповых патронов и светильников, а также шнуров, с помощью которых включаются в электросеть электроприборы, должны быть в исправном состоянии.

6. Электроприборы необходимо хранить в сухом месте, избегать резких колебаний температуры, вибрации, сотрясений.

7. Для подогрева воды пользоваться сертифицированными электроприборами с закрытой спиралью и устройством автоматического отключения, с применением несгораемых подставок.

8. Запрещается:

* пользоваться неисправными электроприборами и электропроводкой
* очищать от загрязнения и пыли включенные осветительные аппараты и электрические лампы
* ремонтировать электроприборы самостоятельно
* подвешивать электропровода на гвоздях, металлических и деревянных предметах, перекручивать провод, закладывать провод и шнуры на водопроводные трубы и батареи отопления, вешать что-либо на провода, вытягивать за шнур вилку из розетки
* прикасаться одновременно к персональному компьютеру и к устройствам, имеющим соединение с землей (радиаторы отопления, водопроводные краны, трубы и т.п.), а также прикасаться к электрическим проводам, неизолированным токоведущим частям электрических устройств, аппаратов и приборов (розеток, патронов, переключателей, предохранителей)
* применять на открытом воздухе бытовые электроприборы и переносные светильники, предназначенные для работы в помещениях
* пользоваться самодельными электронагревательными приборами и электроприборами с открытой спиралью
* наступать на переносимые электрические провода, лежащие на полу

9. При перерыве в подаче электроэнергии и уходе с рабочего места выключать оборудование.

Требования техники безопасности в аварийных ситуациях:

1. Немедленно прекратить работу, отключить персональный компьютер, иное электрооборудование и доложить руководителю работ, если:

* обнаружены механические повреждения и иные дефекты электрооборудования и электропроводки
* наблюдается повышенный уровень шума при работе оборудования
* наблюдается повышенное тепловыделение от оборудования
* мерцание экрана не прекращается
* чувствуется запах гари и дыма
* прекращена подача электроэнергии

2. Не приступать к работе до полного устранения неисправностей.

3. В случае возгорания или пожара работники (в том числе и студенты, проходящие практику) должны немедленно прекратить работу, отключить электроприборы, вызвать пожарную команду, сообщить руководителю работ и приступить к ликвидации пожара имеющимися первичными средствами пожаротушения.

4. При обнаружении запаха газа в помещении:

* предупредить работников, находящихся в помещении, о недопустимости пользования открытым огнем, курения, включения и выключения электрического освещения и электроприборов
* открыть окна (форточки, фрамуги) и проветрить помещение
* сообщить об этом администрации организации, а при необходимости - вызвать работников аварийной газовой службы

5. При травме в первую очередь освободить пострадавшего от травмирующего фактора, поставить в известность руководителя работ, вызвать медицинскую помощь, оказать первую доврачебную помощь пострадавшему и по возможности сохранить неизменной ситуацию до начала расследования причин несчастного случая.

Требования техники безопасности по окончании работы:

1. Привести в порядок рабочее место. Для уборки мусора и отходов использовать щетки, совки и другие приспособления.

2. Отключить электрооборудование, кроме тех электроприборов, которые работают круглосуточно.

3. При выходе из здания студент обязан:

* убедиться в отсутствии движущегося транспорта
* ходить по тротуарам и пешеходным дорожкам

Нормативное обеспечение:

1. Трудовой кодекс Российской Федерации № 197-ФЗ от 30 декабря 2001 года.

2. Федеральный закон от 24.07.1998 № 125-ФЗ «Об обязательном социальном страховании от несчастных случаев на производстве и профессиональных заболеваний».

3. Федеральный закон от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации».

4. Федеральный закон от 21 декабря 1994 г. № 69-ФЗ "О пожарной безопасности".

5. Постановление Минтруда РФ от 24.10.2002 № 73 «Об утверждении форм документов, необходимых для расследования и учета несчастных случаев на производстве, и Положения об особенностях расследования несчастных случаев на производстве в отдельных отраслях и организациях».

6. Приказ Минобразования России от 25.03.2003 г. № 1154 «Об утверждении Положения о порядке проведения практики студентов образовательных учреждений высшего профессионального образования».

7. Приказ Минобрнауки РФ от 26.11.2009 № 673 «Об утверждении Положения об учебной и производственной практики студентов (курсантов)».

# ЛИСТИНГ КОДА

**main.js**

import { createApp } from 'vue'

import App from './App.vue'

import router from './router'

const app = createApp(App)

app.use(router)

app.mount('#app')

**router.js**

import { createRouter, createWebHashHistory } from 'vue-router'

import Home from '@/views/Home.vue'

export default createRouter({

history: createWebHashHistory(),

routes:

[

{

path: '',

component: Home

},

{

path: '/Projects',

component: () => import('./components/Projectaccounting.vue')

},

{

path: '/Tasks',

component: () => import('./components/Taskaccounting.vue')

},

{

path: '/Wirings',

component: () => import('./components/Wiringaccounting.vue')

}

]

})

**App.vue**

<template>

<div id="app">

<router-view

v-bind:spisok="spisok"

v-bind:spisok2="spisok2"

v-bind:spisok3="spisok3"

@remove-project="removeproject"

@add-project="addproject"

@edit-Project="editProject"

@change-Editing="changeEditing"

>

</router-view>

</div>

</template>

<script>

export default {

name: 'App',

data() {

return {

spisok: [

{ title: 'Team fortress 3', id: 1, completed: false, isEditing: false },

{ title: 'Half-life 3', id: 2, completed: false, isEditing: false},

{ title: 'Бально-рейтинговая система ЮЗГУ', id: 3, completed: false, isEditing: false},

{ title: 'ПО для РЖД', id: 4, completed: false, isEditing: false }

],

spisok2: [

{ title: 'Написание движка', id: 1, pro: 'Team fortress 3', completed: false, isEditing: false},

{ title: 'Написание лора', id: 2, pro: 'Half-life 3', completed: false, isEditing: false},

{ title: 'Обновление системы', id: 3, pro: 'Бально-рейтинговая система ЮЗГУ', completed: false, isEditing: false},

{ title: 'Создание схемы', id: 4, pro: 'ПО для РЖД', completed: true, isEditing: false}

],

spisok3: [

{ date: '2022-06-02', hours: 6, title: 'Написал движок', pro: 'Написание движка', id: 1, completed: false, isEditing: false},

{ date: '2022-06-01', hours: 5, title: 'Написал лор', pro: 'Написание лора', id: 2, completed: false, isEditing: false},

{ date: '2022-06-01', hours: 8, title: 'Создано обновление', pro: 'Обновление системы', id: 3, completed: false, isEditing: false},

{ date: '2022-05-12', hours: 8, title: 'Создано основная схема', pro: 'Создание схемы', id: 4, completed: false, isEditing: false,}

]

}

},

methods:

{

removeproject(id, form) {

if (form === 1)

this.spisok = this.spisok.filter(i => i.id !== id)

if (form === 2)

this.spisok2 = this.spisok2.filter(i => i.id !== id)

if (form === 3)

this.spisok3 = this.spisok3.filter(i => i.id !== id)

},

addproject(newproject, form) {

if (form === 1)

this.spisok.push(newproject)

if (form === 2)

this.spisok2.push(newproject)

if (form === 3)

this.spisok3.push(newproject)

},

changeEditing(itext, form) {

if (form === 1) {

this.spisok = this.spisok.map(i => {

if (i.title === itext) {

i.isEditing = !i.isEditing

}

return i;

})

}

if (form === 2) {

this.spisok2 = this.spisok2.map(i => {

if (i.title === itext) {

i.isEditing = !i.isEditing

}

return i;

})

}

if (form === 3) {

this.spisok3 = this.spisok3.map(i => {

if (i.title === itext) {

i.isEditing = !i.isEditing

}

return i;

})

}

},

editProject(itext, ipro, form) {

if (form === 1) {

this.spisok = this.spisok.map(i => {

if (i.title === itext) {

i.isEditing = !i.isEditing;

i.title = itext;

}

return i;

})

}

if (form === 2) {

this.spisok2 = this.spisok2.map(i => {

if (i.title === itext) {

i.isEditing = !i.isEditing;

i.title = itext;

i.pro = ipro;

}

return i;

})

}

if (form === 3) {

this.spisok3 = this.spisok3.map(i => {

if (i.title === itext) {

i.isEditing = !i.isEditing;

i.title = itext;

i.pro = ipro;

}

return i;

})

}

}

}

}

</script>

<style>

#app {

font-family: Avenir, Helvetica, Arial, sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

text-align: center;

color: #2c3e50;

margin-top: 60px;

}

</style>

**Home.vue**

<template>

<div>

<h2> Корпоративный портал учета рабочего времени</h2>

<a href="#/Projects"> Список проектов</a>

<p></p>

<a href="#/Tasks"> Список задач</a>

<p></p>

<a href="#/Wirings"> Список проводок</a>

</div>

</template>

**Projectaccounting.vue**

<template>

<div>

<h2> Список проектов</h2>

<form @submit.prevent="onsubmit">

<input type="text" class="textcg" v-model="title" />

<button type="submit" class="butcg">Создать новый проект</button>

</form>

<div>

<ul>

<li v-for="(i, f) of spisok"

v-bind:class="{ done: i.completed }"

:key="i.id">

<span>

<input type="checkbox" v-on:change="i.completed = !i.completed" />

<input type="text" v-if="i.isEditing" @keyup.enter="$emit('edit-Project',i.title, Null, 1)" v-model="i.title" />

<span v-else> ID:**{{**f + 1**}}**, Проект: **{{**i.title**}}** </span>

</span>

<button class="ra" @click="$emit('change-Editing', i.title, 1)">Изменить</button>

<button class="rm" @click="$emit('remove-project', i.id, 1)">

&times;

</button>

</li>

</ul>

</div>

</div>

</template>

<script>

export default

{

props: ['spisok'],

f: Number,

methods: {

onsubmit() {

if (this.title.trim()) {

const newproject = {

title: this.title,

id: Date.now(),

completed: false,

}

this.$emit("add-project", newproject, 1)

this.title = ""

}

},

},

}

</script>

<style scoped>

ul {

list-style: none;

margin: 0;

padding: 0;

}

li {

border: 1px solid #ccc;

display: flex;

justify-content: space-between;

padding: 0.5rem 2rem;

margin-bottom: 1rem;

}

.rm {

background: red;

color: #ffffff;

border-radius: 0%;

font-weight: bold;

}

input {

margin-right: 1rem;

}

.done {

text-decoration: line-through;

}

.done {

text-decoration: line-through;

}

form {

display: flex;

padding: 1rem;

}

.textcg {

width: 500px;

padding: 0.3rem;

}

.butcg {

width: 300px;

}

.ra {

font-weight: bold;

width: 150px;

}

</style>

**Taskaccounting.vue**

<template>

<h2> Список задач</h2>

<form @submit.prevent="onsubmit">

<input type="text" class="textcg" v-model="title" />

<select v-model="selectedspisok">

<option v-for="g in spisok">**{{**g.title**}}**</option>

</select>

<button type="submit" class="butcg">Создать новую задачу</button>

</form>

<div>

<ul>

<li v-for="(i,f) in spisok2" v-bind:class="{done:i.completed}">

<span>

<input type="checkbox"

v-on:change="i.completed = !i.completed" />

<input type="text" v-if="i.isEditing" @keyup.enter="$emit('edit-Project',i.title,i.pro,2)" v-model=i.title>

<span v-else> ID: **{{**f+1**}}**, Задача: **{{**i.title**}}**,</span>

<select v-if="i.isEditing" v-model="i.pro">

<option v-for="g in spisok">**{{**g.title**}}**</option>

</select>

<span v-else> Проект: **{{**i.pro**}}**</span>

</span>

<button class="ra" v-on:click="$emit ('change-Editing',i.title, 2)">Изменить</button>

<button class="rm" v-on:click="$emit ('remove-project',i.id, 2)">&times;</button>

</li>

</ul>

</div>

</template>

<script>

export default

{

props:

['spisok','spisok2'],

f: Number,

data() {

return {

}

},

methods:

{

onsubmit() {

if (this.title.trim()) {

const newproject = { title: this.title, id: Date.now(), pro: this.selectedspisok, completed: false, isEditing: false}

this.$emit('add-project', newproject,2)

this.title = ''

}

}

}

}

</script>

<style scoped>

ul {

list-style: none;

margin: 0;

padding: 0;

}

li {

border: 1px solid #ccc;

display: flex;

justify-content: space-between;

padding: .5rem 2rem;

margin-bottom: 1rem;

}

.rm {

background: red;

color: #ffffff;

border-radius: 0%;

font-weight: bold;

}

.done {

text-decoration: line-through;

}

form {

display: flex;

padding: 1rem;

}

.textcg {

width: 500px;

padding: 0.3rem

}

.butcg {

width: 300px;

}

.ra {

font-weight: bold;

width: 150px;

}

</style>

**Wiringaccounting.vue**

<template>

<h2> Список проводок</h2>

<form @submit.prevent="onsubmit">

<input type="date" class="textcg1" v-model="date" @change="changecolor" />

<input type="number" class="textcg2" v-bind:max="24-checkhours" min="1" v-model="hours" @click="changecolor" @keyup.enter="changecolor" />

<input type="text" class="textcg3" v-model="title" placeholder="Описание" />

<select v-model="selectedtask">

<option disabled value="">Выберите один из вариантов</option>

<option v-for="e in filteredLast">

**{{**e.title**}}**

</option>

</select>

<button type="submit" class="butcg" @click="changecolor" @change="changecolor">Создать новую проводку</button>

</form>

<div>

Фильтр проводок

<select v-model="filter" @change="onChangeSelect($event)">

<option value="All">За всё время</option>

<option value="Day">За день</option>

<option value="Month">За месяц</option>

</select>

<input type="date" v-model="date" @click="changecolor" @change="changecolor"/>

<span>  <span id="kvadrat" v-bind:style="{'background':color}">    </span></span>

<ul>

<li v-for="(i,f) in filteredList" v-bind:class="{done:i.completed}">

<span>

<input type="checkbox"

v-on:change="i.completed = !i.completed" />

<input type="text" v-if="i.isEditing" @keyup.enter="$emit('edit-Project',i.title,i.pro, 3)" v-model=i.title>

<span v-else>Дата:**{{**i.date**}}**, Количество часов:**{{**i.hours**}}**, Описание:**{{**i.title**}}**</span>

<select v-if="i.isEditing" v-model="i.pro">

<option v-for="e in spisok2" :disabled="spisok2.filter(el=>el.completed).map((el) => el.id).includes(i.id) || e.completed">**{{**e.title**}}**</option>

</select>

<span v-else> Задача: **{{**i.pro**}}**</span>

</span>

<button class="ra" @click="$emit ('change-Editing',i.title,3)">Изменить</button>

<button class="rm" @click="$emit ('remove-project',i.id,3)">&times;</button>

</li>

</ul>

</div>

</template>

<script>

export default

{

props:

['spisok3','spisok2'],

f: Number,

data() {

return {

color: 'white',

hours: 1,

filter: 'All',

Dayn: '',

date: ''

}

},

computed: {

filteredList: function () {

if (this.filter == 'All') {

return this.spisok3

}

if (this.filter == 'Day') {

return this.spisok3.filter(t => {

return this.date.includes(t.date)

})

}

if (this.filter == 'Month') {

return this.spisok3.filter(t => {

return t.date.substr(5, 2).includes(this.date.substr(5,2))

})

}

},

filteredLast: function () {

return this.spisok2.filter(function (p) {

return !p.completed

})

},

checkhours: function() {

return this.spisok3

.filter(t => this.date.includes(t.date))

.reduce((total, i) => total + i.hours, 0)

}

},

methods:

{

onsubmit() {

if (this.title.trim()) {

const newproject = { date: this.date, hours: this.hours, title: this.title, pro: this.selectedtask, id: Date.now(), completed: false, isEditing: false }

this.$emit('add-project', newproject, 3)

this.hours = ''

this.title = ''

this.pro = ''

}

},

changecolor() {

if (this.checkhours == 8)

this.color = 'lime'

if (this.checkhours > 8)

this.color = 'red'

if (this.checkhours < 8)

this.color = 'yellow'

}

}

}

</script>

<style scoped>

ul {

list-style: none;

margin: 0;

padding: 0;

}

li {

border: 1px solid #ccc;

display: flex;

justify-content: space-between;

padding: .5rem 2rem;

margin-bottom: 1rem;

}

.rm {

background: red;

color: #ffffff;

border-radius: 0%;

font-weight: bold;

}

.done {

text-decoration: line-through;

}

form {

display: flex;

padding: 1rem;

}

.textcg1 {

width: 100px;

padding: 0.3rem

}

.textcg3 {

width: 200px;

padding: 0.3rem

}

.textcg4 {

width: 100px;

padding: 0.3rem

}

.butcg {

width: 300px;

}

.ra {

font-weight: bold;

width: 150px;

}

#kvadrat {

background: pink;

}

</style>