МИНИСТЕРСТВО ОБРАЗОВАНИЯ САХАЛИНСКОЙ ОБЛАСТИ

Государственное бюджетное профессиональное образовательное учреждение «Сахалинский промышленно-экономический техникум»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**(ДИПЛОМНЫЙ ПРОЕКТ)**

ТЕМА:

Студент группы ОП-1601 Федотов Владислав Алексеевич

Специальность 09.02.05. Прикладная информатика (по отраслям)

Руководитель: Кущ Игорь Владимирович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Допустить к защите:

Зам. директора по УР (Е.А. Дубровина)

Зав. отделением № 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Н.В. Фаткина)

Оценка Дата

Председатель Государственной

Экзаменационной комиссии

Южно-Сахалинск

2020

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc43062955)

[ГЛАВА 1. ПРОЕКТИРОВАНИЕ WEB ПРИЛОЖЕНИЯ 6](#_Toc43062956)

[1.1. Техническое задание 6](#_Toc43062957)

[1.2. Описание и характеристики аналогичного программного обеспечения 11](#_Toc43062958)

[1.3. Функциональные и технологические возможности web-приложения 15](#_Toc43062959)

[1.4. Описание применяемых технологий по разработке backend web-приложения 17](#_Toc43062960)

[1.5. Сравнение с другими языками программирования 22](#_Toc43062961)

[1.6 План-проект на разработку программного обеспечения 27](#_Toc43062962)

[ГЛАВА 2. СОЗДАНИЕ ПРОГРАММНОГО ПРОДУКТА 28](#_Toc43062963)

[2.1. Подготовка необходимого инструментария для разработки web-приложения 28](#_Toc43062964)

[2.2. Создание и описание backend web-приложения 31](#_Toc43062965)

[2.3. Описание работы разработанного web-приложения 37](#_Toc43062966)

[2.4. Необходимая документация по внедрению и адаптации разработанного web приложения на предприятии 39](#_Toc43062967)

[ЗАКЛЮЧЕНИЕ 41](#_Toc43062968)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 43](#_Toc43062969)

**ВВЕДЕНИЕ**

К настоящему времени во многих организациях накоплены значительные объемы данных, на основе которых имеется возможность решения разнообразных производственных и управленческих задач. Проблемы хранения и обработки информации становятся все более актуальными и привлекают внимание специалистов и фирм, работающих в области информационных технологий.

В идеале работа административных кадров и руководителей различных уровней должна быть организована так, чтобы они могли иметь доступ ко всей интересующей их информации и пользоваться удобными и простыми средствами представления и работы с этой информацией. Именно на достижение этих целей и направлены информационные технологии, объединяющиеся под общим названием хранилищ данных.

Одним из примеров таких организаций является Государственное бюджетное профессиональное образовательное учреждение «Сахалинский Промышленно-Экономический Техникум».

Актуальность дипломного проекта заключается в том, что данное web-приложение позволит повысить качество и облегчит выполнение производственных задач для административных работников организации, а также облегчит работу преподавателям в подключении каталогов в локальной сети организации для эффективной работы студентов с различными файлами и документами.

Цель данной дипломной работы является разработка Web-приложения для решения производственных задач Государственного Бюджетного Профессионального Образовательного Учреждения «СПЭТ», которое позволит облегчить выполнение производственных работ и позволит подключать каталоги для эффективной работы с файлами.

Для достижения поставленной цели необходимо решить следующие задачи:

* Сформулировать техническое задание;
* Провести анализ аналогичного программного обеспечения и Web-приложений;
* Рассмотреть функциональные и технологические возможности web-приложений;
* Создать план-проект разработки программного продукта;
* Описать применяемых технологий по разработке web-приложения;
* Провести сравнения с аналогичными технологиями написания web-приложений;
* Разработать web-приложение.

Структура дипломной работы отражает логику, порядок проектирования и ряд задач. В соответствии с целью и задачами, поставленными в работе, она структурно включает в себя пояснительную записку, состоящую из введения, двух глав, заключения, списка использованных источников и приложения.

**ГЛАВА 1. ПРОЕКТИРОВАНИЕ WEB ПРИЛОЖЕНИЯ**

**1.1. Техническое задание**

Введение

Настоящий документ представляет собой описание композиции web-приложений для решения производственных задач Государственного Бюджетного Профессионального Образовательного Учреждения СПЭТ.

Представляет из себя три web-приложения, направленные на выполнение конкретных производственных задач, таких как:

* СУБД для хранения и управление всей необходимой информации о студентах и web-интерфейс позволяющий быстро получить доступ к данным и вывести их на печать;
* web-интерфейс позволяющий быстро подготовить стандартные шаблоны документов предприятия и вывести их на печать;
* web-интерфейс позволяющий управлять общими рабочими каталогами студентов внутри локальной сети предприятия.

Основания для разработки

Разработка введётся на основании выпускной квалификационной работы в рамках образовательного учреждения СПЭТ.

Полное официальное наименование предприятия - Государственное бюджетное профессиональное образовательное учреждение «Сахалинский промышленно-экономический техникум».

Сокращенное наименование: ГБПОУ СПЭТ.

Местонахождение: 693010, Сахалинская область, город Южно-Сахалинск, улица А.М. Горького, дом 28.

Основная деятельность: является предоставление среднего профессионального образования в целях подготовки квалифицированных рабочих (служащих) и специалистов среднего звена по всем основным направлениям общественно полезной деятельности в соответствии с потребностями общества и государства, а также удовлетворения потребностей личности в углублении и расширении образования.

Назначение разработки

Функциональным и эксплуатационным назначением приложений является облегчение выполнения производственных задач для административных сотрудников предприятия.

Первой задачей является разработка системы управления базой данных, представленная виде REST (аббревиатура Representational State Transfer передача состояния представления) API (аббревиатура от Application Programming Interface интерфейс программирования приложений) которая должна через web-интерфейс предоставлять возможность управлять, контролировать и изменять данные.

Вторая задача разработать композицию web-интерфейсов с необходимым набором функционала для взаимодействия с СУБД.

Функциональные характеристики

Разрабатываемая база данных содержит в себе следующие таблицы:

* Пользователи;
* Студенты;
* Документы;

REST API СУБД предоставляет набор запросов:

* Авторизация пользователя;
* Получение списка студентов по установленным фильтрам;
* Создание новых студентов;
* Удаление студентов;
* Изменение студентов;
* Добавление шаблонов документов;
* Создание из шаблонов готовые документы;
* Управление каталогами студентов;

Надежность

REST API СУБД должен быстро и качество отвечать на запросы пользователей, в случаи возникновения неполадок записываться информацию об ошибках в специальный файл (логи).

Web-интерфейс должен обладать отзывчивым и без перебойным интерфейсом, в следствии чего, единственное, что будет преграждать пользователю в работе, это ожидание ответа сервера, зависящее от текущего интернет соединения.

Условия эксплуатации

Web-приложение должно запускаться во всех современных браузерах последних версий.

Состав и параметры технических средств

REST API СУБД разрабатывается на языке программирования Golang с использованием Golang библиотеки. Базой данных в проекте, выступает PostgreSQL 12.1. Web-интерфейс будет генерироваться на стороне сервера с использованием технологий: Golang Templates, HTML, CSS, JavaScript ECMAScript спецификации 2015 года. Веб-дизайн фреймворком выступает Vuetify.

Требования к программной документации

В каталоге с исходным кодом REST API СУБД будет прилагаться:

* Инструкция по эксплуатации Web приложением.

Технико-экономические показатели

В таблице ниже расписано технико-экономические показатели разработки проекта.

Таблица 1– Технико-экономические показатели

|  |  |  |  |
| --- | --- | --- | --- |
| Число рабочих мест | Аналоги | Разработка | Экономические преимущества |
| 5 | 15000-20000 руб. | 5000 руб. | 10000-15000 руб. |
| 10 | 30000-40000 руб. | 5000 руб. | 25000-35000 руб. |
| 15 | 45000-60000 руб. | 5000 руб. | 40000-55000 руб. |
| 20 | 60000-80000 руб. | 5000 руб. | 55000-75000 руб. |

Таблица 2–Стадии и этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Номер этапа | Название этапа | Срок | Отчетность |
| 1 | Определение целей создания программы | 03.01.20 - 16.01.20 | Выявление определенных цели для дальнейшей реализации |
| 2 | Проведение исследований по теме | 16.01.20 – 23.01.20 | Изучение материала по данной теме |
| 3 | Разработка макета дизайна интерфейсов | 23.01.20 – 12.02.20 | Генерация идей дизайна, разработка предварительного макета, исправление замечаний, доработка макета |
| 4 | Разработка первой рабочей версии | 12.02.20 – 12.04.20 | Создание первого прототипа программы выполняющий главные задуманные функции |
| 5 | Реализация оставшегося функционала | 12.04.20 – 05.05.20 | Все материалы сайта вписываются в общую концепцию, соответствующим целям и задачам |
| 6 | Тестирование и исправление ошибок | 05.05.20 – 31.05.20 | Производится тестирование и исправление выявленных ошибок |

Порядок контроля и приема

Завершенный проект будет проверен руководителем на работоспособность, функциональность и на наличие необходимой документации, после рабочая система войдет в эксплуатацию организацией.

**1.2. Описание и характеристики аналогичного программного обеспечения**

На данный момент времени сектор рынка ПО систем, автоматизирующих процесс документооборота в ВУЗах представлен различными программными продуктами.

Таблица 3–Краткое описание аналогов

|  |  |  |
| --- | --- | --- |
| Название | Краткое описание | Недостатки |
| «ЛУЧ» | Обеспечение информационной поддержки двух компонентов - системы индивидуального учебного планирования и аттестации обучающихся. | Не предусматривает использование мыши, что могло бы сделать её использование более удобным, сравнительно высокая стоимость покупки программы, и её сопровождения. |
| «Студенты» | Программа для работы кадровых служб студентов и деканатов с документами учащихся, связана с 1С Бухгалтерией. | Для её настройки и администрирования требуются знания специалиста, требуется проведение платных курсов для обучения персонала |
| «Контингент» | Предназначен для учета движения контингента начиная от подачи заявления о приеме и заканчивая выпуском учащихся из образовательного учреждения. | Программа является платной, и, гарантийная поддержка действительна только 6 месяцев со дня приобретения. По истечению срока гарантийной поддержки 6 месяцев наступает обычное абонентское обслуживание. Стоимость абонентского обслуживания составляет: одно рабочее место 1960 руб. в мес. |

ИС «Луч»

Система «Луч» разрабатывалась для обеспечения информационной поддержки двух компонентов этой технологии - системы индивидуального учебного планирования (дистанционное обучение) и аттестации обучающихся.

Учебное планирование предусматривает составление индивидуального учебного плана для каждого студента на основе вариативности, как по содержанию, так и по его темпу. При составлении индивидуального учебного плана студенту оказывает помощь консультант.

Система аттестации в СГУ состоит из нескольких частей:

модульный контроль знаний в течение семестра (домашние работы, тесты, практические занятия, компьютерные занятия, лабораторные работы);

текущая аттестация - экзамен по каждой изучаемой дисциплине и выполнение курсовых работ (много вопросный экзамен проводится в письменной форме, проверку осуществляют эксперты Аттестационного Центра СГУ);

итоговая аттестация - итоговый экзамен и публичная защита выпускной квалификационной работы.

Информационная Система «Луч» состоит из двух основных частей - программной и аппаратной.

Программный комплекс, рассылаемый в филиалы, собирает информацию о проведении модульного контроля знаний в течение семестра, также накапливает информацию по оплате обучения и кадровые данные по студенту, результаты социологических опросов. С определенной периодичностью осуществляется отправка накопленных данных в центр. Для передачи в центр выделяются изменения в накопленной информации, по сравнению с предыдущей отправкой с целью минимизации расходов на пересылку. Информация передается двумя способами:

* через Интернет по электронной почте;
* на дискетах по обычной почте.

Модуль анализа успеваемости студента предоставляет кураторам данные по модульному контролю знаний, на основании которых принимается решение о допуске студента к текущей аттестации по конкретной учебной дисциплине, также фиксирует результаты текущей и итоговой успеваемости.

Модуль анализа финансовых показателей предоставляет информацию для бухгалтерии.

АСУ «Студент»

Автоматизированная система управления персоналом в учебном заведении Программный комплекс "Студент" призван облегчить работу кадровых служб студентов и деканатов с документами учащихся. Программа дает возможность:

* максимально упростить и автоматизировать работу по приему, перемещению и отчислению учащихся;
* стандартизация отчетов об успеваемости, задолжниках, сводная экзаменационная ведомость, отчет о количестве студентов;
* автоматически формировать первичную документацию на каждого студента (контракты, подготовка и печать протоколов и приказов о зачислении);
* ведение учебных планов с учетом специальностей, специализаций, курсов, групп;
* контроль оплаты обучения с учетом сроков сессии и приказов о продлении контракта;
* моделировать произвольные отчеты из индивидуальных сведений учащихся на основе критериев, указанных пользователем;
* совместимости с 1С: Бухгалтерией;
* интеграции со стандартными офисными приложениями, такими как MS Word, MS Excel.

Автоматизированная система управления образованием (АСУО) «Контингент»

Программа «АСУО - Контингент» предназначена для автоматизации процесса управления образовательным учреждением, делая более эффективной работу всех его участников. Данный продукт предназначен для учета движения контингента начиная от подачи заявления о приеме и заканчивая выпуском учащихся из образовательного учреждения. Во время обучения в «АСУО - Контингент» накапливается вся необходимая информация (успеваемость, посещаемость, учебная нагрузка и ее выполнение) для формирования личной карточки учащегося, которая аккумулируется в базе данных и используется для разнообразных целей: формирования различных отчетов, получения результатов успеваемости в разрезе месяцев, семестров, общая успеваемость групп и многое другое.

«АСУО - Контингент» выполняет следующие функции:

Учебный процесс:

* составление рабочего учебного плана;
* формирование тарификации преподавателей;
* получение сведений о педагогической нагрузке преподавателей.

**1.3. Функциональные и технологические возможности web-приложения**

PWA (Progressive Web App) – это веб-технология, которая трансформирует сайт в приложение. Прямо из браузера его можно поставить на главный экран телефона, и оно будет отправлять push-уведомления и получит доступ к аппаратным средствам гаджета. И все это даже при нестабильном подключении или офлайн. PWA-приложения ставятся на смартфон пользователя в обход официальных магазинов приложений и несмотря на запрет ставить приложения с неизвестных источников.

Три основные функции таких продуктов:

* Надежность: независимо от того, в каких условиях доступности сети работает пользователь, PWA загружается мгновенно. Службы, написанные на JavaScript, несут ответственность за эту работу.
* Скорость: дополнительно к мгновенной загрузке, пользователи также получают молниеносные взаимодействия.
* Взаимодействие с пользователем: внешний вид прогрессивного веб-приложения не отличается от обычного. Возможности веб-приложений предлагает множество опций для вашего продукта: внешний вид иконок на главном экране, ориентацию экрана, страницу, которая должна быть загружена первой, и т. д.

Не будет проблем с интеграцией с магазинами приложений

Интеграция с магазинами приложений значительно усложняет разработку приложений и заставляет погонять его под требования площадки. В разработке прогрессивного веб-приложения таких трудностей нет, что позволяет сосредоточиться на качестве продукта. Все, что нужно, — это размещение вашего продукта на веб-сервере.

У продукта не будет ограничений по устройствам

Прогрессивные веб-приложения могут работать одинаково хорошо на любом устройстве, будь то смартфон, настольный ПК или планшет. Эта бесценная функция не только дает безупречный опыт для вашей аудитории, но также помогает оставаться последовательным в веб-среде.

Прогрессивное веб-приложение не занимает много места на устройстве, поскольку оно размещено на веб-сервере. Если пользователи хотят установить ваше приложение на главном экране, оно по-прежнему не потребует хранения или регулярных обновлений, что положительно сказывается на пользовательском опыте.

Продуктом легко поделиться.

Это преимущество также возможно благодаря хранению приложению на веб-сервере. Новые пользователи не должны открывать магазин приложений, искать и загружать его. Чтобы начать работать, человеку нужен только URL-адрес. И, как результат, приложение будет распространяться быстрее, поскольку люди могут делиться этим адресом с помощью любого сервиса и любого устройства.

## **1.4. Описание применяемых технологий по разработке backend web-приложения**

Backend-разработка — это набор аппаратно-программных средств, при помощи которых реализована логика работы сайта. Попросту говоря, это то, что скрыто от глаз пользователя и происходит вне его браузера и компьютера.

Например, когда вы вводите запрос на странице поисковика и жмёте клавишу Enter, frontend заканчивается и начинается backend. Ваш запрос отправляется на сервер Google или «Яндекса», где расположены алгоритмы поиска. Именно там случается всё «волшебство». Как только на мониторе появилась информация, которую вы искали, — вновь происходит возвращение в зону frontend.

Взаимодействие frontend и backend происходит по кругу:

* frontend отправляет пользовательскую информацию в backend;
* информация обрабатывается;
* и возвращается обратно, приняв понятную форму.

Этими работами занимаются разные специалисты, но каждому из них желательно понимать принципы, по которым работают коллеги. Даже дизайнеру интерфейсов важно хотя бы в общих чертах знать, что представляет собой backend проекта, которым он занимается. Это поможет адекватно оценить, какие технические возможности есть у сайта или приложения.

Golang

Golang, или Go — язык программирования, начало которого было положено в 2007 году сотрудниками компании Google: Робертом Гризмером, Робом Пайком и Кеном Томпсоном.

Go поддерживает типобезопасность, возможность динамического ввода данных, а также содержит богатую стандартную библиотеку функций и встроенные типы данных вроде массивов с динамическим размером и ассоциативных массивов. С помощью механизмов многопоточности Go упрощает распределение вычислений и сетевого взаимодействия, а современные типы данных открывают программисту мир гибкого и модульного кода. Программа быстро компилируется, при этом есть сборщик мусора и поддерживается рефлексия.

Это быстрый, статически типизированный, компилируемый язык, при использовании которого создается впечатление использования динамически типизированного и интерпретируемого языка.

Язык Go разрабатывался как язык для создания различных высокоэффективных программ, однако большинство программистов сходятся во мнении, что лучше всего он подходит для создания веб-приложений (в качестве back-end). При этом Go дает возможности писать и другие проекты, к примеру, Docker, InfluxDB и Kubernetes. По сути, применение языка Go ограничивается тремя основными направлениями: сетевое программное обеспечение, консольные утилиты и бэкенд.

Одной из отличительных особенностей языка является оригинальная система типов: в языке отсутствует наследование (один из принципов объектно-ориентированного программирования). Также в Go используется сокращенный синтаксис определения переменных и синтаксис анонимных функций.  
Еще одна особенность этого языка – параллелизм, который заключается в том, что любая функция может быть выполнена одновременно с другой.

PostgreSQL

PostgreSQL — это свободная объектно-реляционная система управления базами данных. PostgreSQL базируется на языке SQL.

СУБД отличается высокой надёжностью и хорошей производительностью. PostgreSQL поддерживает транзакции (ACID), репликация реализована встроенными механизмами. При этом система расширяемая — можно создавать свои типы данных и индексов, а также расширять поведение при помощи языков программирования.

Преимущества PostgreSQL:

* поддержка БД неограниченного размера;
* мощные и надёжные механизмы транзакций и репликации;
* расширяемая система встроенных языков программирования и поддержка загрузки C-совместимых модулей;
* наследование;
* легкая расширяемость.

Текущие ограничения PostgreSQL:

* Нет ограничений на максимальный размер базы данных
* Нет ограничений на количество записей в таблице
* Нет ограничений на количество индексов в таблице
* Максимальный размер таблицы — 32 Тбайт
* Максимальный размер записи — 1,6 Тбайт
* Максимальный размер поля — 1 Гбайт
* Максимум полей в записи250—1600 (в зависимости от типов полей)

Функции в PostgreSQL являются блоками кода, исполняемыми на сервере, а не на клиенте БД. Хотя они могут писаться на чистом SQL, реализация дополнительной логики, например, условных переходов и циклов, выходит за рамки, собственно, SQL и требует использования некоторых языковых расширений. Функции могут писаться с использованием различных языков программирования. PostgreSQL допускает использование функций, возвращающих набор записей, который далее можно использовать так же, как и результат выполнения обычного запроса. Функции могут выполняться как с правами их создателя, так и c правами текущего пользователя. Иногда функции отождествляются с хранимыми процедурами, однако между этими понятиями есть различие.

Многоверсионность поддерживается в PostgreSQL — возможна одновременную модификация БД несколькими пользователями с помощью механизма Multiversion Concurrency Control (MVCC). Благодаря этому соблюдаются требования ACID, и практически отпадает нужда в блокировках чтения.

Gorilla Mux

Пакет gorilla/mux реализует маршрутизатор и диспетчер запросов для сопоставления входящих запросов с их соответствующим обработчиком.

Название mux означает «мультиплексор HTTP-запросов». Как и стандарт http.ServeMux, mux.Router сопоставляет входящие запросы со списком зарегистрированных маршрутов и вызывает обработчик для маршрута, который соответствует URL-адресу или другим условиям. Основными функциями являются:

* Он реализует http.Handler интерфейс, поэтому он совместим со стандартом http.ServeMux.
* Запросы могут быть сопоставлены на основе URL-адреса хоста, пути, префикса пути, схем, значений заголовка и запроса, методов HTTP или использования пользовательских сопоставителей.
* Хосты URL, пути и значения запросов могут иметь переменные с необязательным регулярным выражением.
* Зарегистрированные URL-адреса могут быть построены или «перевернуты», что помогает поддерживать ссылки на ресурсы.
* Маршруты можно использовать в качестве подчиненных маршрутов: вложенные маршруты проверяются только при совпадении родительского маршрута. Это полезно для определения групп маршрутов, которые имеют общие условия, такие как хост, префикс пути или другие повторяющиеся атрибуты.

Axios

Библиотека Axios, предназначенная для выполнения HTTP-запросов, основана на промисах. Она подходит для использования в среде Node.js и в браузерных приложениях. Библиотека поддерживает все современные браузеры, и, в том числе, IE8+.

Сильные стороны:

* Работает в среде Node.js и в браузерах;
* Поддерживает промисы;
* Позволяет выполнять и отменять запросы;
* Позволяет задавать тайм-аут ответа;
* Поддерживает защиту от XSRF-атак;
* Позволяет перехватывать запросы и ответы;
* Поддерживает индикацию прогресса выгрузки данных;
* Широко используется в проектах, основанных на React и Vue.

Слабые стороны:

* Библиотекой довольно сложно пользоваться.

## **1.5. Сравнение с другими языками программирования**

Go

Язык, разработанный Google, который придется по душе любителям Python. Go — простой язык, но он более эффективен по сравнению с C ++. Go обеспечивает отличную поддержку многопоточности, поэтому он используется многими компаниями, которые сильно зависят от распределенных систем. Go широко применяется стартапами в Кремниевой долине и индийскими компаниями. Хотите стать частью Silicon Valley — учите Golang.

Преимущества:

* Достаточно безопасный
* Простой синтаксис, легкий в изучении
* Обширная стандартная библиотека, предлагающая ряд встроенных функций для работы с примитивными типами
* Очень быстрый, так как компилируется в машинный код

Недостатки:

* Отсутствие виртуальной машины делает сложные программы менее эффективными
* Неявные интерфейсы
* Go не хватает универсальности

PHP

PHP является одним из самых популярных бэкэнд-языков программирования. Хотя PHP сталкивается с жесткой конкуренцией со стороны Python и JavaScript, на рынке все еще требуется большое количество разработчиков PHP.

Сегодня около 70% сайтов используют PHP, а в WordPress около 90% кода написано на этом языке. Благодаря PHP вы можете создавать очень динамичные и интерактивные сайты, что очень приятно для глаз текущего пользователя.

Преимущества:

* С ним легко начать создавать веб-страницы
* Поддержка огромного сообщества пользователей и обширной экосистемы
* Включает множество средств автоматизации тестирования и развертывания приложений
* C PHP работает большое количество мощных фреймворков

Недостатки:

* Плохая обработка ошибок
* Не самый безопасный
* Разработка веб-сайтов полностью на PHP происходит медленнее по сравнению с использованием других опций

Python

Серверный язык разработки. Как правило, он используется для написания скриптов и плагинов. Например, через Python записываются эпизоды и действия персонажей в играх.

Это универсальный язык с легко читаемым кодом. Требует выполнения меньшего количества действий для создания плагина, чем Java или C ++. Такие бренды, как Instagram и Surveymonkey, Google, Netflix и Quora не могут обойтись без Python. Кроме того, язык Python используется для разработки машинного обучения, искусственного интеллекта, робототехники и Big Data.

Преимущества:

* Поддерживает большинство библиотек
* Простой красивый код
* Идеален для создания прототипов и быстрого тестирования идей
* Открытое программное обеспечение

Недостатки:

* Несмотря на упрощенный способ написания кода, многие процессы выполняются достаточно долго
* Не подходит для создания мобильных приложений
* Требует проведения многочисленных тестов для полного устранения ошибок

Java

Java является одним из самых популярных языков программирования, используемых для разработки серверных приложений. Многие сайты сегодня не работают без Java. Этот язык широко используется большими компаниями для создания приложений различного типа. Среда Java очень стабильна, что делает его идеальным выбором для написания приложений для Android.

Преимущества:

* Легко читаемый и управляемый язык
* Независим от платформы
* Идеален для распределенных вычислений
* Обладает автоматическим управлением памятью и высоким уровнем безопасности

Недостатки:

* Непрост в изучении
* Имеет нестандартный интерфейс, к которому нужно долго привыкать
* Потребляет много памяти и снижает производительность

C++

C ++ — очень эффективный и гибкий язык. Он был создан в 1985 году, но до сих пор пользуется очень высоким спросом из-за своей высокой производительности и стабильности.

Microsoft Windows и Google являются двумя наиболее значимыми компаниями, в которых используется язык C ++. Более того, большинство сайтов Amazon также основаны на этом языке. C ++ в основном полезен для разработки десктоп-приложений.

Преимущества:

* Предлагает низкий уровень абстракции
* Огромное количество библиотек для работы с ним
* Имеет широкий спектр применения: игры, приложения с графическим интерфейсом и математическое моделирование в реальном времени
* Является отличной основой для понимания более сложных языков программирования

Недостатки:

* Сложный синтаксис
* Не поддерживает пространство имен программы
* Не способен решать самые актуальные и последние проблемы программирования

Swift

Swift — это универсальный язык программирования, разработанный Apple, который предлагает разработчикам простой и понятный синтаксис. Он родился под влиянием Python и Ruby и является безопасным и простым в освоении. Благодаря своей универсальности и практичности, Swift заменил Objective-C в качестве основного языка для приложений, связанных с Apple. Кроме того, поскольку Apple активно продвигает Swift, его популярность и сообщество растут.

Преимущества:

* Легко добавлять новые функции
* Совместим с Objective-C
* Улучшает совместное использование кода и ускоряет процесс разработки, когда используется как для frontend, так и для backend разработки
* Автоматическое управление памятью предотвращает ее утечки

Недостатки:

* Ограниченная поддержка и ресурсы
* Пока еще не очень стабильный
* Может использоваться только для iOS,
* не поддерживает более ранние версии

# **1.6 План-проект на разработку программного обеспечения**

Планирование в том или ином виде производится в течение всего срока реализации проекта. В самом начале жизненного цикла проекта обычно разрабатывается неофициальный предварительный план - грубое представление о том, что потребуется выполнить при реализации проекта.

Решение о выборе проекта в значительной степени основывается на оценках предварительного плана. Формальное и детальное планирование проекта начинается после принятия решения о его открытии. Определяются ключевые события - вехи проекта, формулируются задачи, работы и их взаимная зависимость.

Планирование — это непрерывный процесс определения наилучшего способа действий для достижения поставленных целей с учетом складывающейся обстановки.

План проекта — это единый, последовательный и согласованный документ, включающий результаты планирования всех функций управления проектом и являющийся основой для выполнения и контроля проекта.

Таким образом, сформулировав основные цели, работы, задачи работ, вехи поучаем следующий план-проект на разработку программного обеспечения (Приложение № **твоего приложение** План проект).

**ГЛАВА 2. СОЗДАНИЕ ПРОГРАММНОГО ПРОДУКТА**

**2.1. Подготовка необходимого инструментария для разработки web-приложения**

Для начала создания web-приложения необходимо было установить и настроить описанный инструментарий для web разработки.

Первое что мы скачиваем и устанавливаем был редактор кода Visual Studio Code. Переходим на сайт code.visualstudio.com, секция Download. Здесь есть версия под Mac, Linux. Windows. Выбераем нужную операционную систему и скачиваем себе на ПК. (рис. 1)



Рисунок. 1 скачивание Visual Studio Code

Программа скачана, запускаем инсталлятор и следуем инструкциям: принимаем, добавляем все галочки, далее, ждем пока программа установится на наш ПК. (рис. 2)

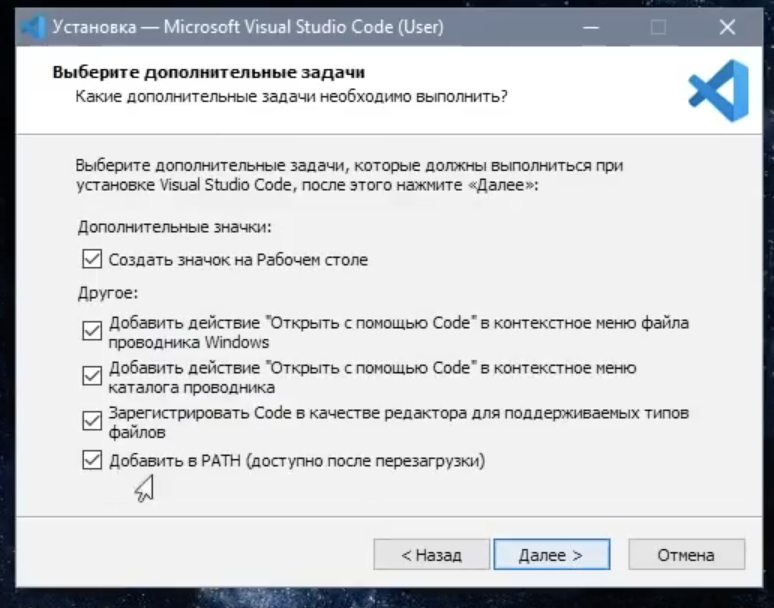


Рисунок. 2 установка Visual Studio Code

Так как программа изначально на английском языке для удобства я русифицировал ее для этого щелкаю на иконку расширений, В поле поиска начинаю писать Russian Language Pack for Visual Studio Code. (рис. 3)

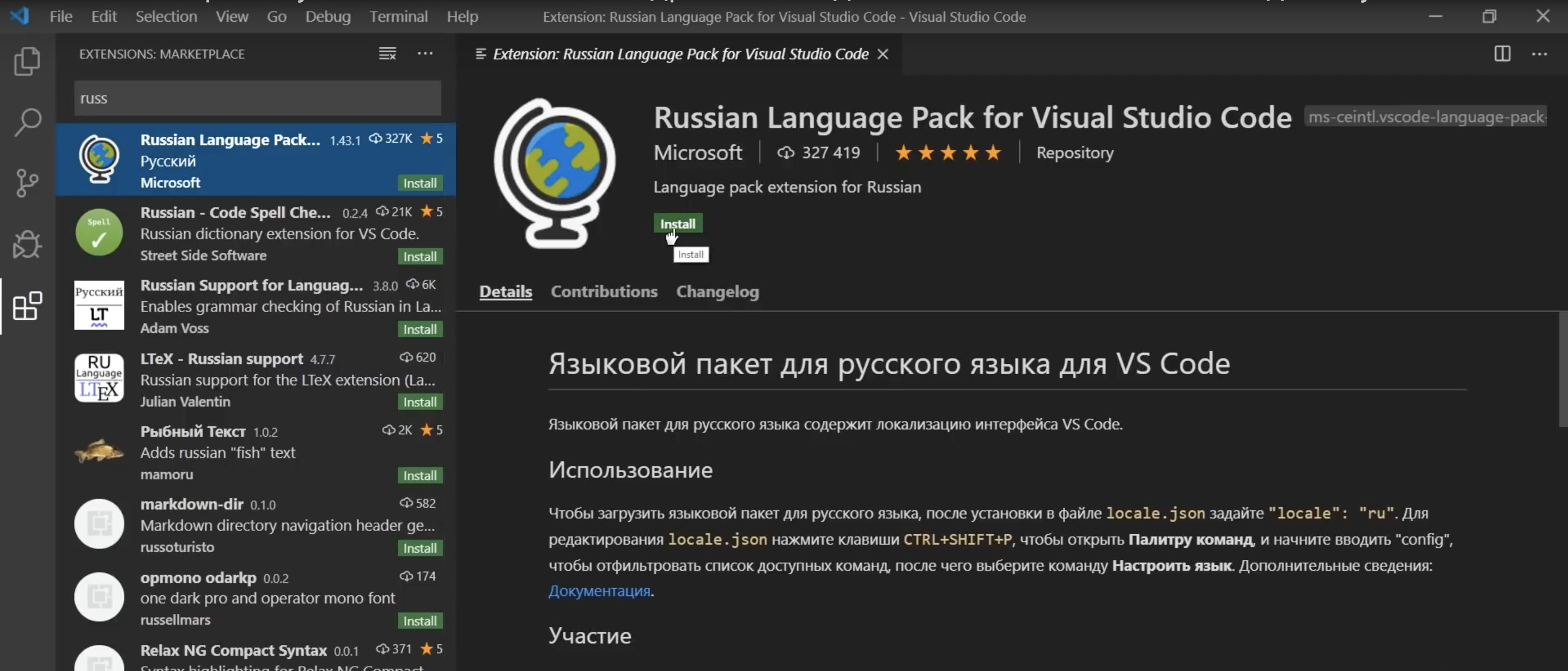


Рисунок 3 русификация Visual Studio Code

Visual Studio Code готов для начала создания web-приложения (рис. 4)

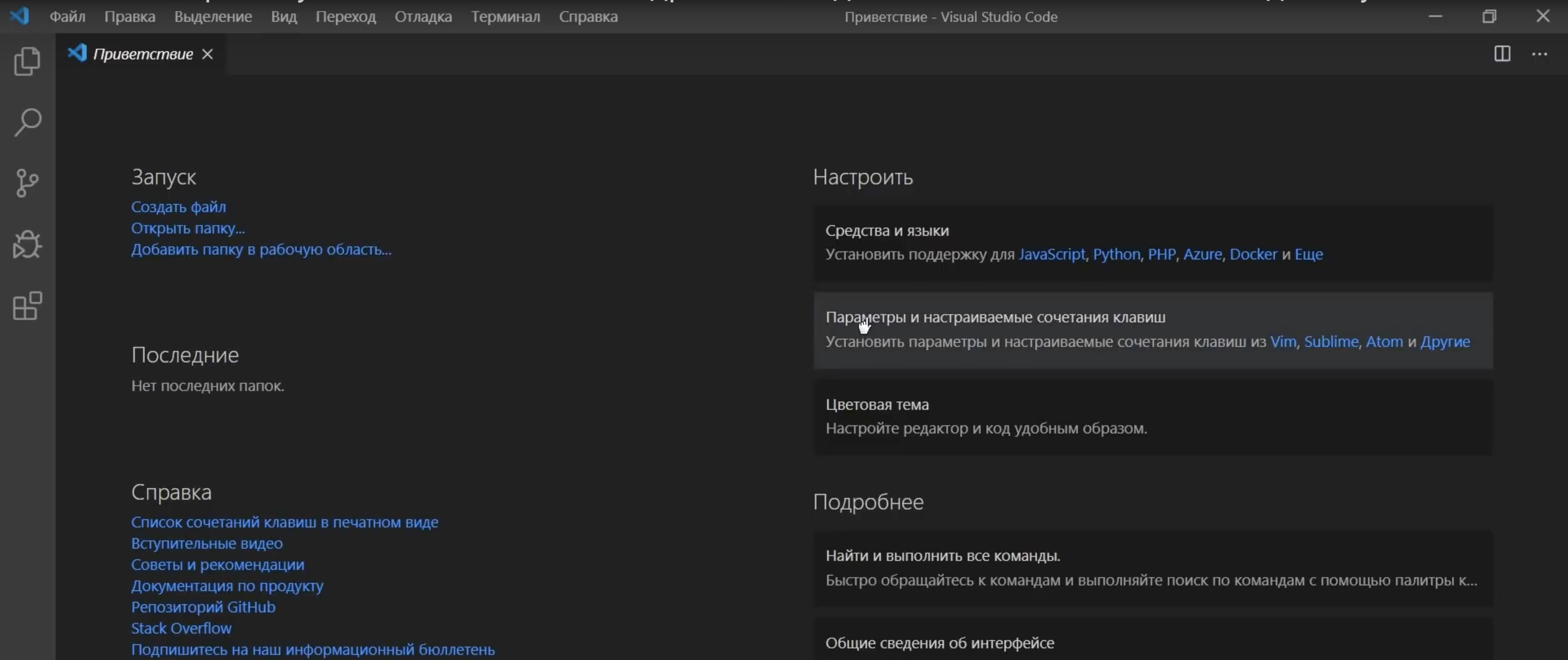


Рисунок 4 Visual Studio Code

ОПИСАНИЕ ДАЛЬНЕЙШЕГО ПО ДЛЯ РАЗРАБОТКИ

## **2.2. Создание и описание backend web-приложения**

Для начала мы запускаем сервер содержит следующий код:

package main

import (

"log"

"github.com/Survereignty/spet-rest-api/private/core/api"

"github.com/Survereignty/spet-rest-api/private/core/api/db"

"github.com/Survereignty/spet-rest-api/private/core/lig"

)

func main() {

// Инициализация логов

err := lig.Create("./logs/logs.txt")

if err != nil {

log.Fatal(err)

}

lig.Info("Logs is init")

defer lig.Quit()

// Настройки по умолчанию

defaultConfig := &api.Config{

Address: ":3001",

DataBase: db.NewConfig(

"host=localhost password=student dbname=spet sslmode=disable"),

}

// Инициализация настроек

c, err := api.NewConfig(defaultConfig)

if err != nil {

lig.Сrash("Config file is not created", err)

}

lig.Info("Config file is init")

// Инициализация сервера

s := api.New(c)

if err := s.Start(); err != nil {

lig.Сrash("Server is not started", err)

}

}

Дальше………. содержит следующий код:

// Запуск API

func (s \*Api) Start() error {

if err := s.Store(); err != nil {

return err}

s.Routers()

lig.Listen("Server listen port", s.config.Address)

return http.ListenAndServe(s.config.Address, s.router)

}

Подключаем БД содержит следующий код:

// Open ...

func (s \*Store) Open() error {

db, err := sql.Open("postgres", s.config.DatabaseURL)

if err != nil {

return err

}

if err := db.Ping(); err != nil {

return err

}

s.db = db

return nil

}

{

"Address": ":3002",

"DataBase": {

"DatabaseURL": "host=localhost password=1234 dbname=spet sslmode=disable"

}

}

Создаем обработку данных БД содержит следующий код:

package db

import (

model "github.com/Survereignty/spet-rest-api/private/core/api/models")

type StudentRep struct {

store \*Store}

// Создать студента

func (r \*StudentRep) Create(m \*model.Student) error {

if err := r.store.db.QueryRow(

"INSERT INTO students (surname, middleName, name, date\_b, city, street, house, flat, info, numGroup, activs, gender, status, orphan, invalid, low\_income, low\_parents, idn, kdn, many\_children, login, password, budget) VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9, $10, $11, $12, $13, $14, $15, $16, $17, $18, $19, $20, $21, $22, $23) RETURNING id",

m.Surname, m.MiddleName, m.Name, m.Date\_b, m.City, m.Street, m.House,

m.Flat, m.Info, m.NumGroup, m.Activs, m.Gender, m.Status, m.Orphan, m.Invalid,

m.Low\_income, m.Low\_parents, m.Idn, m.Kdn, m.Many\_children, m.Login, m.Password, m.Budget,

).Scan(&m.Id); err != nil {

return err}

return nil}

// Изменить студента

func (r \*StudentRep) Update(m \*model.Student) error {

\_, err := r.store.db.Exec("UPDATE students SET surname = $2, middleName = $3, name = $4, date\_b = $5, city = $6, street = $7, house = $8, flat = $9, info = $10, numGroup = $11, activs = $12, gender = $13, status = $14, orphan = $15, invalid = $16, low\_income = $17, low\_parents = $18, idn = $19, kdn = $20, many\_children = $21, login = $22, password = $23, budget = $24 WHERE id = $1",

m.Id, m.Surname, m.MiddleName, m.Name, m.Date\_b, m.City, m.Street, m.House,

m.Flat, m.Info, m.NumGroup, m.Activs, m.Gender, m.Status, m.Orphan, m.Invalid,

m.Low\_income, m.Low\_parents, m.Idn, m.Kdn, m.Many\_children, m.Login, m.Password, m.Budget)

if err != nil {

return err}

return nil}

// Удалить

func (r \*StudentRep) Delete(m \*model.Student) error {

\_, err := r.store.db.Exec(

"DELETE from students where id = $1",

m.Id,)

if err != nil {

return err}

return nil}

// Получить всех студентов

func (r \*StudentRep) Get() ([]\*model.Student, error) {

query := "SELECT \* FROM students"

rows, err := r.store.db.Query(query)

if err != nil {

return nil, err}

defer rows.Close()

sts := make([]\*model.Student, 0)

for rows.Next() {

st := new(model.Student)

err := rows.Scan(&st.Id, &st.Surname, &st.MiddleName, &st.Name, &st.Date\_b, &st.City, &st.Street, &st.House,

&st.Flat, &st.Info, &st.NumGroup, &st.Activs, &st.Gender, &st.Status, &st.Orphan, &st.Invalid,

&st.Low\_income, &st.Low\_parents, &st.Idn, &st.Kdn, &st.Many\_children, &st.Login, &st.Password, &st.Budget)

if err != nil {

return nil, err}

sts = append(sts, st)}

if err = rows.Err(); err != nil {

return nil, err}

return sts, nil}

## **2.3. Описание работы разработанного** **web-приложения**

При открытии браузера в поисковую строку водится URL-адрес web-приложения, в появившемся окне браузера открывается интерфейс инициализации пользователя в систему. Пользователь вводит свои данные для входа логин и пароль. Во время входа web-приложения проверяется наличие пользователя системы и, если проверка прошла успешно появляется окно с тремя модулями web-приложения.

При нажатии на первый модуль web-приложения пользователь входит в систему учета студентов, где у пользователя есть возможность:

* Добавить, редактировать или удалить информацию о студенте техникума;
* Сортировать студентов по нужным критериям для решения производственной задачи;
* Создание шаблона фильтров критерий для поиска студентов;
* Вывод на печать данных о студентов.

При нажатии на второй модуль web-приложения пользователь входит в систему позволяющий быстро подготовить стандартные шаблоны документов предприятия и вывести их на печать, где у пользователя есть возможность:

* Добавить готовый шаблон документов нужный пользователю;
* Быстро заполнить данными о студентах в готовый шаблон;
* Вывести на печать документ из предложенных шаблонов.

При нажатии на третий модуль web-приложения пользователь входит в систему позволяющая управлять общими рабочими каталогами студентов внутри локальной сети предприятия, где у пользователя есть возможность:

* Подключить каталоги в кабинете, в котором идет учебный процесс;
* Подключить каталоги группы и каталоги студентов для изучения теоретических и практических материалов для обучения;
* Создание и использования готовых шаблонов для подключения каталогов.

При завершение свое производственной деятельности в системе нажмите кнопку «Выхода» запуститься начальное окно с авторизацией пользователя.

## **2.4. Необходимая документация по внедрению и адаптации разработанного web приложения на предприятии**

Руководство пользователя

Введение

Web-приложение предназначено для решения разнообразных аналитических и управленческих задач. Web-приложение позволит руководству быстро подготовить стандартные шаблоны документов предприятия и вывести их на печать, преподавателям управлять общими рабочими каталогами студентов внутри локальной сети предприятия.

Выполняемые функции

Web-приложение выполняет:

* Авторизация и инициализация пользователей web приложения;
* Таблица ввода, вывода и редактирования информации студентов;
* Создание шаблона поиска по критериям студентов;
* Создание и редактирование шаблонов для печати производственных документов;
* Вывод на печать готовых шаблонов производственных документов;
* Подключение директорий студентов в локальной сети предприятия.

Условия эксплуатации

Web-приложение должно запускаться во всех современных браузерах последних версий.

Использование web приложения

Необходимо наличие интернета на вашем устройстве. Для запуска web-приложения вводим URL-адрес. В появившемся окне браузера осуществляем вход в систему. Для инициализации пользователя вводим данные логин и пароль. После проверки наличия пользователя в базе данных web-приложения осуществляется вход в систему. После успешного входа, пользователь выбирает нужную систему такие как:

* СУБД для хранения и управление всей необходимой информации о студентах и web-интерфейс позволяющий быстро получить доступ к данным и вывести их на печать;
* web-интерфейс позволяющий быстро подготовить стандартные шаблоны документов предприятия и вывести их на печать;
* web-интерфейс позволяющий управлять общими рабочими каталогами студентов внутри локальной сети предприятия.

Пользователь начинает работать в зависимости от того, какая система нужна в решение производственной задаче.

# **ЗАКЛЮЧЕНИЕ**

Таким образом при работе надо дипломным проектом была выполнена цель: разработать Web-приложение для решения производственных задач Государственного Бюджетного Профессионального Образовательного Учреждения «СПЭТ», которое позволит облегчить выполнение производственных работ и позволит подключать каталоги для эффективной работы с файлами.

Для этого были выполнены следующие задачи:

* было сформулировано техническое задание на разработку Web-приложений;
* проведен анализ аналогичного программного обеспечения и Web-приложений;
* определены функциональные и технологические возможности web-приложений;
* были описаны применяемые технологии по разработке web-приложения;
* создан план-проект разработки программного продукта;
* проведено сравнения с аналогичными технологиями написания web-приложений;
* разработано web-приложение.

Также при работе над дипломным проектом был изучен большой объем отраслевой информации, направленный на разработку Web-приложения.

Данный документ был оформлен с помощью программного обеспечения Microsoft Word, Web-приложение было разработано с помощью программного обеспечения Visual Studio Code.

Фронтенд Web-приложения был написан при помощи таких языков как HTML, CSS и JavaScript, а также при помощи инструментов Vue, Vuex, Babel, WebPack.

Также был разработан план-проект на разработку web приложения с помощью программы Microsoft Project. Для этого были сформулированы цели, основные работы, задачи этих работ, также были определены примерные даты для контроля достижения цели. План-проект представлен в виде отдельного файла на диске с дипломным проектом.

Разработанное web приложение выполняет следующие функции:

* СУБД для хранения и управление всей необходимой информации о студентах и web-интерфейс позволяющий быстро получить доступ к данным и вывести их на печать;
* web-интерфейс позволяющий быстро подготовить стандартные шаблоны документов предприятия и вывести их на печать;
* web-интерфейс позволяющий управлять общими рабочими каталогами студентов внутри локальной сети предприятия.

Дипломный проект включает в себя: пояснительную записку, состоящую из введения, двух глав, заключения, списка использованных источников и приложения.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ 19.001-77 Единая система программной документации (ЕСПД). Общие положения [Текст] – Введ. 1980-01-01. – Москва : Стандартинформ, 2010. – 6 с.;
2. ГОСТ 19.202-78 Единая система программной документации (ЕСПД). Спецификация. Требования к содержанию и оформлению (с Изменением N 1) [Текст] – Введ. 1980-01-01. – Москва : Стандартиформ, 2010. – 4с.;
3. ГОСТ 19.401-78 ЕСПД. Текст программы. Требования к содержанию и оформлению (с Изменением N 1) [Текст] – Введ. 1980-01-01. – Москва : Стандартинформ, 2010. – 2 с.;
4. ТВОИ ИСТОЧНИКИ ПО БЭК ЭНДУ