

СОДЕРЖАНИЕ

| | |
|---|-----------|
| ВВЕДЕНИЕ..... | 3 |
| ГЛАВА 1. СТРУКТУРА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ НА ТЕМУ «СИСТЕМА СБОРА И ОБРАБОТКИ СТАТИСТИЧЕСКИХ ДАННЫХ ОБ ОБЩЕОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЯХ ПО ВОРОНЕЖСКОЙ ОБЛАСТИ»..... | 5 |
| 1.1 Структура и компоненты проекта..... | 5 |
| 1.2. Фреймворки..... | 7 |
| ГЛАВА 2. СТАТЬЯ ПО ТЕМЕ «ИНФОРМАЦИОННАЯ СИСТЕМА МОНИТОРИНГА ШКОЛЬНЫХ ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЙ»..... | 11 |
| ЗАКЛЮЧЕНИЕ..... | 13 |
| СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ..... | 14 |

ВВЕДЕНИЕ

На сегодняшний день существует огромное количество информации и данных, которые приходится собирать, измерять и анализировать. Необходимо это для ведения мониторинга за различными процессами, в частности деятельностью различных организаций. Мониторинг необходим для ведения контроля за деятельностью организаций, выявлению позитивных или негативных изменений и для надзора за выполнением поставленных норм. Мониторинг необходим для оптимизации деятельности организаций.

В рамках данной работы будут рассматриваться бюджетные образовательные учреждения. Одним из самых удобных и наглядных способов анализа и мониторинга большего количества данных, особенно когда речь заходит о числовых или количественных значениях, является статистика.

Сбор статистики позволяет наблюдать текущее, прошлое и даже предположить будущее состояние вещей, например, число приходящих учеников в школе или число изменения количества сотрудников определенной квалификации как в выбранном учреждении, так и во всех учреждениях, объединенных общим атрибутом, например, город.

В сборе данных и составлении статистики нуждаются так же образовательные организации по Воронежской области, однако на данный момент нет подходящих решений, доступных в интернете.

В связи имеющейся проблемой, были поставлены цели и задачи.

Цель исследования — разработать веб-сервис для сбора и обработки информации образовательных учреждений по Воронежской области с целью осуществления мониторинга их деятельности.

Объект исследования – информационная система мониторинга.

Предмет исследования – информационная система мониторинга школьных образовательных организаций Воронежской области.

В соответствии с целью поставлены следующие **задачи**:

1. Определить общую структуру проекта и его главные компоненты.
2. Провести анализ прикладной области и определить средства и способы реализации проекта.
3. Смоделировать модель программного продукта в виде веб-сервера.

Используемые **методы исследования**:

- теоретическими методами исследования поставленных проблем (системных подход, анализ и синтез, моделирование) для разработки защиты архивных данных электронных документов в образовательной организации высшего образования.

ГЛАВА 1. СТРУКТУРА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ НА ТЕМУ «СИСТЕМА СБОРА И ОБРАБОТКИ СТАТИСТИЧЕСКИХ ДАННЫХ ОБ ОБЩЕОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЯХ ПО ВОРОНЕЖСКОЙ ОБЛАСТИ»

1.1 Структура и компоненты проекта

Понятие «веб-сервер» может относиться как к аппаратной начинке, так и к программному обеспечению. Или даже к обеим частям, работающим совместно.

С точки зрения аппаратной части, «веб-сервер» — это компьютер, который хранит файлы сайта (HTML-документы, CSS-стили, JavaScript-файлы, картинки и другие) и доставляет их на устройство конечного пользователя (веб-браузер и т.д.). Он подключён к сети Интернет и может быть доступен через доменное имя, подобное mozilla.org.

С точки зрения ПО, веб-сервер включает в себя несколько компонентов, которые контролируют доступ веб-пользователей к размещенным на сервере файлам, как минимум — это HTTP-сервер. HTTP-сервер — это часть ПО, которая понимает URL-адреса (веб-адреса) и HTTP (протокол, который ваш браузер использует для просмотра веб-страниц).

На самом базовом уровне, когда браузеру нужен файл, размещённый на веб-сервере, браузер запрашивает его через HTTP-протокол. Когда запрос достигает нужного веб-сервера, сервер HTTP (ПО) принимает запрос, находит запрашиваемый документ (если нет, то сообщает об ошибке 404) и отправляет обратно, также через HTTP.

Чтобы опубликовать веб-сайт в интернете, необходим либо статический, либо динамический веб-сервер.

Статический веб-сервер, или стек, состоит из компьютера ("железо") с сервером HTTP (ПО). Он называется «статикой», потому что сервер посылает размещенные файлы в браузер «как есть».

Динамический веб-сервер состоит из статического веб-сервера и дополнительного программного обеспечения, чаще всего сервера приложения и базы данных. Его называют «динамическим», потому что сервер приложений изменяет исходные файлы перед отправкой в ваш браузер по HTTP. В данной работе будет использован динамический веб-сервер.

Сервлет является интерфейсом Java, реализация которого расширяет функциональные возможности сервера. Сервлет взаимодействует с клиентами посредством принципа запрос-ответ. Хотя сервлеты могут обслуживать любые запросы, они обычно используются для расширения веб-серверов. Для таких приложений технология Java Servlet определяет HTTP-специфичные сервлет классы.

Например, сервлеты позволяют «перехватить» HTTP запрос и, перед тем как отправить пользователю запрашиваемую страницу, выполнить заданный java-код, обратиться к любым классам и вызвать любой метод, а затем добавить к полученному запросу дополнительные данные, которые затем обрабатываются и будут использованы в HTML странице.

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных.

Данные в наиболее распространенных типах современных баз данных обычно хранятся в виде строк и столбцов, формирующих таблицу. Этими данными можно легко управлять, изменять, обновлять, контролировать и

упорядочивать. В большинстве баз данных для записи и запросов данных используется язык структурированных запросов (SQL).

Существует множество различных типов баз данных. Выбор наилучшей базы данных для конкретной компании зависит от того, как она намеревается использовать данные.

Реляционные базы данных стали преобладать в 1980-х годах. Данные в реляционной базе организованы в виде таблиц, состоящих из столбцов и строк. Реляционная СУБД обеспечивает быстрый и эффективный доступ к структурированной информации.

Информация в объектно-ориентированной базе данных представлена в форме объекта, как в объектно-ориентированном программировании.

Распределенная база данных состоит из двух или более частей, расположенных на разных серверах. Такая база данных может храниться на нескольких компьютерах.

Будучи централизованным репозиторием для данных, хранилище данных представляет собой тип базы данных, специально предназначенной для быстрого выполнения запросов и анализа.

1.2 Фреймворки

Spring Boot — это фреймворк, целью которого является упрощение создания приложений на основе Spring. Он позволяет наиболее простым способом создать web-приложение, требуя от разработчиков минимум усилий по его настройке и написанию кода.

Spring Boot обладает большим функционалом, но его наиболее значимыми особенностями являются: управление зависимостями, автоматическая конфигурация и встроенные контейнеры сервлетов. Второй полезной возможностью Spring Boot является автоматическая

конфигурация приложения. После выбора подходящего starter-пакета, Spring Boot попытается автоматически настроить Spring-приложение на основе добавленных вами jar-зависимостей.

Каждое Spring Boot web-приложение включает встроенный web-сервер. Это позволяет разработчикам не беспокоиться о настройке контейнера сервлетов и развертывании приложения на нем. Теперь приложение может запускаться само, как исполняемый jar-файл с использованием встроенного сервера.

Spring MVC — это веб-фреймворк Spring. Он позволяет создавать веб-сайты или RESTful сервисы (например, JSON/XML) и хорошо интегрируется в экосистему Spring, например, он поддерживает контроллеры и REST контроллеры в ваших Spring Boot приложениях.

Фреймворк Spring MVC обеспечивает архитектуру паттерна Model — View — Controller (Модель — Отображение (далее — Вид) — Контроллер) при помощи слабо связанных готовых компонентов. Паттерн MVC разделяет аспекты приложения (логику ввода, бизнес-логику и логику UI), обеспечивая при этом свободную связь между ними.

- Model (Модель) инкапсулирует (объединяет) данные приложения, в целом они будут состоять из POJO («Старых добрых Java-объектов», или бинов).
- View (Отображение, Вид) отвечает за отображение данных Модели, — как правило, генерируя HTML, которые мы видим в своём браузере.
- Controller (Контроллер) обрабатывает запрос пользователя, создаёт соответствующую Модель и передаёт её для отображения в Вид.

Вся логика работы Spring MVC построена вокруг DispatcherServlet, который принимает и обрабатывает все HTTP-запросы (из UI) и ответы на них.

После получения HTTP-запроса DispatcherServlet обращается к

интерфейсу `HandlerMapping`, который определяет, какой Контроллер должен быть вызван, после чего, отправляет запрос в нужный Контроллер.

Контроллер принимает запрос и вызывает соответствующий служебный метод, основанный на `GET` или `POST`. Вызванный метод определяет данные Модели, основанные на определённой бизнес-логике, и возвращает в `DispatcherServlet` имя Вида (`View`).

При помощи интерфейса `ViewResolver`, `DispatcherServlet` определяет, какой Вид нужно использовать на основании полученного имени.

После того, как Вид (`View`) создан, `DispatcherServlet` отправляет данные Модели в виде атрибутов в Вид, который в конечном итоге отображается в браузере.

Все вышеупомянутые компоненты, а именно, `HandlerMapping`, `Controller` и `ViewResolver`, являются частями интерфейса `WebApplicationContext extends ApplicationContext`, с некоторыми дополнительными особенностями, необходимыми для создания web-приложений.

`JDBC` — это платформенно независимый промышленный стандарт взаимодействия Java-приложений с реляционными базами данных. Впервые был включен в состав `JDK 1.1` в 1997 году. `JDBC` управляет:

1. подключением к базе данных;
2. выдачей запросов и команд;
3. обработкой данных, полученных из базы.

Пакет `JDBC` состоит из двух главных компонентов:

1. `API` (программного интерфейса), который поддерживает связь между Java-приложением и менеджером `JDBC`;
2. Драйвера `JDBC`, который поддерживает связь между менеджером `JDBC` и драйвером базы данных.

Соединение с базой устанавливается по-особому `URL`. При этом разработчику не нужно знать специфику конкретной базы — `API`

выступает в качестве посредника между базой и приложением. Это упрощает как процесс создания приложения, так и переход на базу данных другого типа.

Этапы подключения к базе данных

1. Установка базы данных на сервер или выбор облачного сервиса, к которому нужно получить доступ.
2. Подключение библиотеки JDBC.
3. Проверка факта нахождения необходимого драйвера JDBC в classpath.
4. Установление соединения с базой данных с помощью библиотеки JDBC.
5. Использование установленного соединения для выполнения команд SQL.
6. Закрытие соединения после окончания сеанса.

Возможностей JDBC API достаточно для реализации простых приложений. Для более масштабных решений чаще используют JPA API, который позволяет сохранять Java-объекты в базе данных.

ГЛАВА 2. СТАТЬЯ ПО ТЕМЕ «ИНФОРМАЦИОННАЯ СИСТЕМА МОНИТОРИНГА ШКОЛЬНЫХ ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЙ»

В связи с растущими требованиями к качеству образования в школьных организациях, становится все более актуальным внедрение информационных технологий в систему образования. Одним из наиболее эффективных методов контроля является создание информационной системы мониторинга, которая позволяет автоматизировать процесс сбора, анализа и оценки данных об образовательном процессе и его качестве.

Информационная система мониторинга должна позволять получать информацию о выполнении учебных программ, уровне освоения учебных материалов студентами, оценки успеваемости и уровня знаний, эффективности методов преподавания и их соответствия современным требованиям образования, а также о соблюдении правил здорового образа жизни, дисциплине и безопасности. Система мониторинга может быть дополнена механизмами автоматического оповещения о проблемах, связанных с учебным процессом и эффективностью преподавания, например, если ученики задаются меньшими целями, чем заданы учебным планом, если ученики не понимают материал или если они чрезмерно нагружены учебной работой.

Система мониторинга должна быть простой и удобной для использования, а ее функциональность должна включать анализ данных и создание отчетов с использованием управляемого справочного и хранилища данных. Также, система должна иметь возможность выгрузки данных для дополнительного анализа, использования инструментов визуализации данных и формирования статистики. Помимо этого, система мониторинга должна быть связана с внешними системами, такими как учетно-расчетная система, электронный дневник и другие, чтобы

обеспечить автоматический обмен данными и улучшить эффективность управления. Информационные системы мониторинга образовательных организаций уже применяются в различных регионах и позволяют существенно улучшить качество обучения студентов, повысить эффективность преподавания и увеличить уровень удовлетворенности администрации, учителей, родителей и студентов.

Таким образом, внедрение информационной системы мониторинга школьных образовательных организаций позволяет обеспечить более эффективное управление образовательным процессом и повысить его эффективность. Особенно важно учитывать проектные этапы и уделять пристальное внимание разработке критериев оценки продукта и новых функций информационного продукта для обеспечения качества и устойчивости этой системы мониторинга.

Информационная система мониторинга является одним из тех инструментов, который позволяет контролировать качество образования в школьных организациях и оценивать эффективность учебного процесса. Однако, создание такой системы представляет собой достаточно сложную задачу и требует учета многих факторов.

Первое, что нужно учитывать — это наличие разных форм обучения. В наши дни в школьные организации приходят ученики, которые предпочитают различные формы обучения, и эта разнообразия нужно учитывать при внедрении системы мониторинга. Важно, чтобы система была гибкой и способной адаптироваться к разным потребностям и формам обучения.

Второе, что следует учитывать, это то, что информационная система мониторинга должна подходить для любого инвестиционного аспекта. Важно следить за техническим обслуживанием, обновлением и управлением системы и гарантировать, что она будет продуктивной и эффективной в течение длительного периода времени.

Третье, что нужно учитывать при создании информационной системы мониторинга, это необходимость отслеживания не только учебного процесса, но и дисциплины и безопасности в школьных организациях. Хорошая система должна следить за поведением студентов, за выполнением правил здорового образа жизни и за дисциплиной.

Четвертое, что нужно учитывать при создании системы мониторинга — это качество данных. Важно, чтобы данные, хранящиеся в системе, были достоверными и полными. Кроме того, это позволит сформировать прозрачную неопределенность данных и снизить вероятность субъективных ошибок.

Пятое, что нужно учитывать, это необходимость предоставления доступа к системе родителям и ученикам. Возможность просмотра результатов, успеваемости и других данных обучения может мотивировать студентов на лучшие результаты в учебе и повысить чувство ответственности за достижение академических целей.

Информационная система мониторинга является эффективным инструментом для контроля качества образования и повышения эффективности учебного процесса. Однако, при ее создании следует учитывать все вышеперечисленные факторы, чтобы гарантировать ее устойчивость, эффективность, продуктивность и привлекательность для студентов и родителей.

ЗАКЛЮЧЕНИЕ

В ходе научной исследовательской работы, был проведен анализ защиты данных архивных электронных документов, выявлены основные методы защиты данных, описаны способы. Было выявлено, что на сегодняшний день существует большое количество способов защиты, от физических до электронных.

Основная идея в том, что к задаче защиты системы электронного документооборота надо подходить с точки зрения классической защиты информационной системы. А именно:

- аутентификация пользователей и разделение доступа;
- подтверждение авторства электронного документа;
- контроль целостности электронного документа;
- конфиденциальность электронного документа;

Обобщая все вышесказанное, заметим, что на сегодняшний день количество способов защиты и их разнообразия существуют в больших количествах, с каждым годом развиваясь и совершенствуются.

В ходе практики, были приобретены необходимые практические умения и навыки работы, которые понадобятся мне в дальнейшей деятельности. По окончании практики была достигнута главная цель – закрепление теоретических знаний, полученных в процессе обучения; приобретения практических навыков, компетенций и опыта деятельности по направлению подготовки; ознакомления на практике с вопросами профессиональной деятельности, направленными на формирование знаний, навыков и опыта профессиональной деятельности.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Гарнаев, А. WEB-программирование на Java и JavaScript / А. Гарнаев, С. Гарнаев. - Москва: СПб. [и др.]: Питер, 2017. - 718 с.
2. Постановление от 5 августа 2013 года № 662 «Об осуществлении мониторинга системы образования»
3. Приказ Росстата от 15.11.2021 № 804 «Об утверждении формы федерального статистического наблюдения с указаниями по ее заполнению для организации Министерством просвещения Российской Федерации федерального статистического наблюдения в сфере общего образования»
4. Савитч, У. Язык Java. Курс программирования / У. Савитч. - М.: Вильямс, 2015. - 928 с.
5. Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных»
6. Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 N 273-ФЗ
7. Хабибуллин, И. Самоучитель Java / И. Хабибуллин. - М.: БХВ-Петербург, 2014. - 768 с.
8. Шилдт, Г. Java 8. Руководство для начинающих / Г. Шилдт. - М.: Вильямс, 2015. - 720 с.