Министерство сельского хозяйства Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Пермский государственный аграрно-технологический университет

имени академика Д.Н. Прянишникова»

Кафедра Информационных технологий

и программной инженерии

**ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ**

(проектно-технологическая практика)

на тему: «Разработка информационной системы

для автоматизации бизнес-процесса»

Выполнил:

студент группы ПИб-21

направления подготовки

09.03.03 Прикладная информатика

Утробина Елизавета Николаевна

Проверил:

доцент кафедры ИТиПИ, к.т.н., доцент

Беляков Андрей Юрьевич

2022

Содержание

[Введение 3](#_Toc114437437)

[Анализ бизнес-процесса 4](#_Toc114437438)

[Анализ технологии проектирования 9](#_Toc114437439)

[Реализация функционала информационной системы 11](#_Toc114437440)

[Заключение 14](#_Toc114437441)

[Список источников 15](#_Toc114437442)

[Приложение А 16](#_Toc114437443)

[Приложение Б. 23](#_Toc114437444)

## Введение

Данная практика посвящена рассмотрению процесса автоматизации бизнес-процессов и выполнению этапов, связанных с разработкой программной реализации

Этапы автоматизации бизнес-процесса:

• исследуем предприятие;

• выбираем плохо автоматизированный процесс;

• составляем модели процесса «как-есть» и «как-будет»;

• составляем Техническое Задание на разработку информационной системы;

• разрабатываем прототип;

• разрабатываем программную реализацию;

• тестируем, внедряем.

Из приведённой последовательности этапов на данной практике будут задействованы:

− составление Технического задания;

− разработка прототипа;

− апробация информационной системы.

В данную задачу будут интегрированы такие вопросы проектирования и программирования как: архитектура web-приложения, MVC-паттерн проектирования, организация обмена данными между программой и базой данных, регистрация и авторизация пользователя, передача параметров и данных между формами приложения, хеширование паролей пользователей.

## Анализ бизнес-процесса

Рассмотрим бизнес-процесс работы с клиентами. Важная составная часть его это обратная связь от клиентов о продукции, об услугах, об оказанном сервисе. Отзывы клиентов — обратная связь от пользователей на ваши продукты или услуги и обязательный атрибут бизнеса. Однако компании не всегда понимают их значение: не собирают, реагируют только на положительные отклики и не работают с негативными. Такой подход — ошибка.

Отзывы формируют лояльность клиента, информируют пользователей о продукте нативно, помогают выявить недостатки, улучшить продукт и принести компании увеличение продаж. Подавляющее большинство покупателей читают отзывы перед покупкой, поэтому игнорировать обратную связь от клиентов не стоит.

Отзывы формируют доверие клиента. Продукт без отзывов вызывает опасения. Продукт исключительно с положительными отзывами — недоверие. Если компания на своём сайте публикуют только положительные отзывы, то пользователи уходят на независимые площадки, типа Яндекс.Маркет, просматривают соцсети или ролики видео-блоггеров.

Недобросовестные компании игнорируют и удаляют негативные отзывы, но такой подход только раздражает клиентов.

Отзывы покупателей — это ещё и увеличение объёма продаж. Девять из десяти заказов онлайн делаются людьми после изучения отзывов. Хороший подробный комментарий убедительнее любой рекламной кампании. Он помогает клиентам понять, подойдёт ли им товар. Исследование «AliExpress Россия» показало, что с такой целью к чужому мнению обращаются 65% покупателей. Через рассказы пользователей люди находят ту информацию, которой нет в стандартном описании.

В основе сервисов бронирования отелей и посуточного жилья именно отзывы. Никакие описания и красивые фотографии не спасут, если кто-то напишет о тараканах в номере.

**`Как мотивировать клиентов написать отзыв о компании, продуктах и услугах?**

Можно просить и часто напоминать, но пользователи в потоке дел забудут о вас или не захотят тратить время на обратную связь. Тогда поможет благодарность за отзыв в виде бонуса.

**Что можно предложить клиентам за обратную связь:**

* Скидку на следующий заказ или подарок. Если скидка на сопутствующий товар, то вы ещё сделаете допродажу.
* Бонус за рекомендацию в соцсетях или конкретному человеку. Если клиент опубликует отзыв на своей странице или расскажет о вас знакомому (например, отправит промокод или реферальную ссылку), то опять же получит скидку, баллы на бонусный счёт по программе лояльности, подарок или бесплатную услугу.
* Конкурс на самый лучший, подробный, креативный отзыв о продукте, например, в виде обзора.

Также можно показывать другие отклики. Здесь работает принцип социального доказательства, когда люди принимают решения или делают что-то, потому что смотрят на окружающих.

**Что важно при сборе отзывов?**

* **Оставить отзыв должно быть максимально просто и быстро.**

Например, на сервисе «Кинопоиск» нельзя не заметить раздел оценки фильма, а поставить её очень просто.

* **Обязательно благодарите клиентов за отзыв.**

На вас потратили своё время, оцените это и получите в ответ плюс к лояльности.

* **Не пишите заказные отзывы.**

Это заметно и вызывает только негативные эмоции. Особенно если недавно вы получили плохие комментарии и сразу же решили перекрыть их несколькими позитивными. Также подозрительными будут сотни отзывов о молодой компании, которая только что появилась на рынке.

* **Объясните, что вы хотите получить от пользователя в отзыве.**

Фото продукта, детальность описания. Общие фразы в отклике не вызывают доверия, они остаются бесполезными для компании. «Мне всё понравилось. Рекомендую», — неудачный отзыв. Почему понравилось? Какие особенности товара или услуги? Аналогично с «мне не понравилось». Что именно? Чтобы получать развёрнутые ответы, клиенту нужно дать подсказку через вопросы.

**Примеры вопросов, которые вы можете задать:**

* Почему вы выбрали именно наш продукт?
* Какую проблему вы хотели решить с помощью нашего продукта? Какой
* был результат?
* Какие ожидания были у вас от продукта и оправдал ли он их?
* Стали бы вы рекомендовать наш продукт?
* Что бы вы улучшили в нашем продукте? Что вам больше всего понравилось и менять не нужно?

**Преимущества сбора отзывов клиентов**

Хорошо продаётся продукт, который делает пользователя счастливым — решает его проблему и закрывает потребность. Узнать, сработал ли ваш продукт, вы можете только от потребителей. Поэтому на вопрос, как повысить продажи, ответят отзывы покупателей.

**Что вы получаете, когда собираете и анализируете отклики клиентов:**

* Обратная связь поможет довести ваш продукт или услугу до совершенства. Вы узнаете, что нравится вашим клиентам, а что их раздражает, что мешает пользоваться вашим продуктом. Отзывы подарят неожиданные инсайты.
* Повышение лояльности покупателей. Спрашивать отзывы и реагировать на них — это способ показать, что вы слышите клиента и готовы ответить на его ожидания. Плохие отклики нужно отрабатывать, а не игнорировать. Правильный ответ и разбор ситуации способны превратить недовольного клиента в адвоката бренда. Другие пользователи увидят ответ и узнают, что ваша компания не безразлична к проблемам клиентов — это плюс к лояльности.
* Отзывы выявляют «дыры» в продукте или услуге и закрывать их. Если при запуске продукта вы можете опираться на маркетинговые исследования, то в процессе их дополняют именно отзывы. Через них вы изучите поведение потребителей в отношении вашего продукта в реальности, а не в смоделированной ситуации.
* Возвращение клиентов и повышение дохода. Если покупатель разочаровался в продукте, получил некачественную услугу и пишет об этом, то ответ не только уменьшит негатив, но и вернет клиента. Как? Предложите всё исправить и сделайте это на 100%. Тогда клиент снова купит у вас, а ещё порекомендует ваш продукт своим друзьям и знакомым.
* Формирование будущих стратегий компании на основе обратной связи от клиентов и создание продуктов под потребности людей.
* Дополнительное продвижение. Отзывы - это уникальный контент с упоминанием вашего бренда, который не повторяется в описаниях на разных сайтах. Берите отзывы на свой сайт и используйте на страницах в соцсетях.

База отзывов может стать частью CRM-системы – системы управления взаимоотношениями с клиентами. Это не только дополнение клиентской базы, но и подход к увеличению продаж услуг и продуктов. С помощью обратной связи вы поймете, как работает ваш продукт или услуга, какие есть недостатки, чего действительно хотят пользователи. Тогда вы улучшите продукт или услугу и доработаете их до ожиданий клиентов.

## Анализ технологии проектирования

В рамках практики будет разработано web-приложение "Отзывы на продукт".

Используемые технологии:

* Nodejs
* Express
* ejs
* MVC
* csv
* json
* re
* SQLite
* html
* css
* lodash

**Архитектура приложения основана на паттерне проектирования MVC (Model-View-Controller).**

Составные части паттерна:

**model** - модель - это часть архитектуры приложения:

* включает в себя бизнес-логику приложения (классы, методы, функции обработки данных);
* модель "не знает" о контроллерах и представлениях, не ориентируется на них;
* данные модели могут быть представлены таблицами базы данных или файлами XML, JSON, csv;
* это может быть просто набор объектов или функций для реализации логика приложения.

**view** - представление - это часть архитектуры приложения

* отвечает за визуальное отображение данных, полученных от модели
* представление может читать данные, но не может записывать или изменять их
* представление может иметь программный код, который реализует логику отображения данных

**controller** - контроллер - это часть архитектуры приложения:

* совокупность обработчиков событий представления
* принимает события от представления
* отдаёт событие на обработку в модель
* возвращает пользователю обновлённое представление
* или выбирает какое именно представление должно быть отображено в ответ.

## Реализация функционала информационной системы

В качестве базы данных была выбрана CУБД sqlite, за её простоту и отсутствие необходимсти установки сервера базы данных.

Логическая модель базы данных:

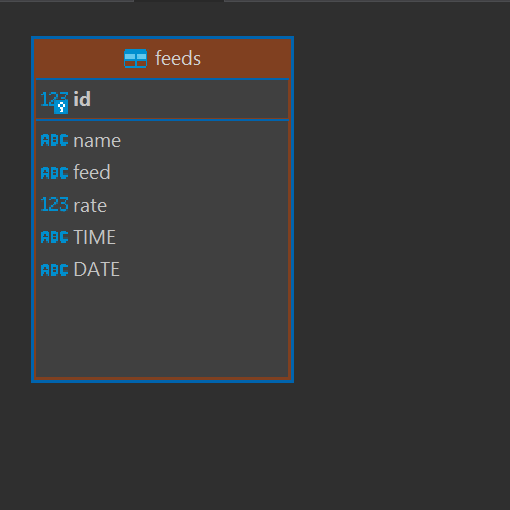


Рис. 1 – Логическая модель БД

Как видно из рис.1, база данных состоит только из одной картинки. Эта таблица включает в себя поля:

1. name – имя пользователя ИС
2. feed – отзыв пользователя
3. rate – его оценка от 1 до 10
4. TIME – время в формате «HH:MM:SS»
5. DATE – дата в формате «YYYY-MM-DD»

**Описание sql-запросыов**

INSERT INTO feeds(name,feed,rate,TIME,DATE) VALUES(?,?,?,?,?) - запрос используется для вставки в таблицу в соответствующие поля таблицы значений обозначенных знаком вопроса.

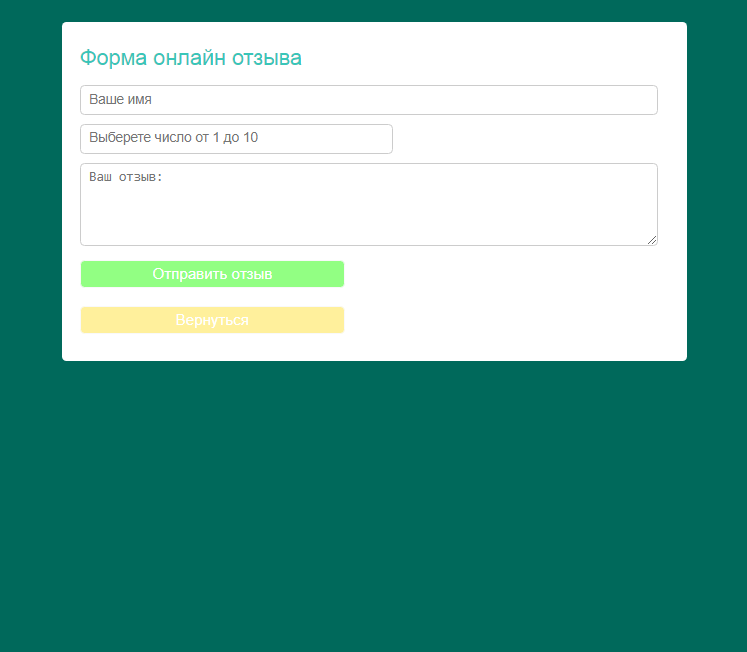
SELECT DATE, TIME, name, feed, rate FROM feeds ORDER BY id DESC LIMIT ? - запрос используется для получения из таблицы данных с ограниченим на количество, запрашиваемых строк на число, обозначенных знаком вопроса.

**Организация интерфейса пользователя** и функционал визуальных форм приложения.

Главная форма приложения: 

Рис.2 Главная форма приложения

Форма отзыва:

Рис.3 Форма отзыва

## Заключение

Мы закрепили знания по использованию паттерна проектирования MVC(model, view, controller). Изучили некоторые особенности работы с «СУБД sqlite», а также взаимодействие с этой базой данных с помощью пакета npm «sqlite3», в которой реализован интерфейс взаимодействия с базой данных. Научились работать с шаблонизатором «ejs», и фреймворком для быстрого написания сайтов «express».

# 

# Список источников

1. ГОСТ 19.401-78 Текст программы. требования к содержанию и оформлению.
2. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы.
3. ГОСТ 19.701-90 Схемы алгоритмов, программ данных и систем.
4. Алгоритм хеширования MD5. [Электронный ресурс] – Электрон. текстовые дан. Режим доступа: https://ru.wikipedia.org/wiki/MD5
5. МЕТОДИЧЕСКОЕ ПОСОБИЕ по выполнению курсового проекта по дисциплине Объектно-ориентированное программирование. Режим доступа: https://pcoding.ru/pdf/CourseProject.pdf
6. Git Repositories Documentation [Электронный ресурс] – Электрон. текстовые дан. – Microsoft, Режим доступа: https://docs.microsoft.com/ru-ru/azure/devops/repos/git, свободный.
7. MSDN. Техническая документация. MD5 Класс. [Электронный ресурс] Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/api/system.security.cryptography.md5?view=netframework-4.8
8. MSDN. Техническая документация. HashAlgorithm.ComputeHash Метод. [Электронный ресурс] Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/api/system.security.cryptography.hashalgorithm.computehash?view=netframework-4.8
9. Методическое пособие по лабораторным работам по дисциплине «Проектирование информационных систем». Пермь: Пермская ГСХА
10. Хомоненко, А.Д., Цыганков, В.М., Мальцев, М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. проф. А.Д. Хомоненко. – 5-е изд., доп. и перераб. – СПб.: КОРОНА принт, 2006. – 736 с.
11. Хэррон, Д., Node.js разработка веб-приложений на Javascript

## Приложение А

наименование организации – разработчика ТЗ на АС

УТВЕРЖДАЮ

Руководитель

(должность, наименование предприятия – заказчика АС)

Личная подпись Расшифровка подписи

(печать)

Дата

УТВЕРЖДАЮ

Руководитель

(должность, наименование предприятия – разработчик АС)

Личная подпись Расшифровка подписи

(печать)

Дата

наименование вида АС

наименование объекта автоматизации

сокращённое наименование АС

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

На листах

Действует с

2021

1. Общие сведения
2. Наименование системы

Полное наименование разрабатываемой системы – «**ИС обратной связи с пользователями»**

1. Наименование заказчика и исполнителя

Организация: ФГБОУ ВО Пермский ГАТУ.

Адрес: ул. Петропавловская, 23.

Телефон: 89523356508;

Исполнитель: Утробина Елизавета Николаевна.

1. Плановые сроки начала и окончания работ

Дата начала работ: **04.07.2022**.

Дата окончания работ: **18.07.2022**.

1. Назначение и цели создания системы

К целям создания модуля «Живая лента» можно отнести:

* улучшение оперативного взаимодействия и интеграция модулей;
* автоматизировать опрос сотрудников и студентов университета.

Достижение целей приведёт к следующим положительным результатам:

* руководитель сможет быстрее оценивать оперативную картину видя назначенные мероприятия и поручения в едином потоке информации;
* благодаря возможности комментировать информационную публикацию в модуле «Живая лента» отправители избавятся от необходимости получать обратную связь от исполнителей или участников мероприятий через модуль «Сообщения» или «Чат» каждый раз при проведении мероприятия или назначении поручения;
* кураторам или старостам будет удобнее информировать свои группы студентов при помощи информационных публикаций или проводить опросы;
* пользователи портала смогут видеть информационные публикации в отсортированном в порядке убывания даты.

1. Характеристика объекта автоматизации

Объектом автоматизации является подсистема организационной коммуникации корпоративного портала ФГБОУ ВО Пермский ГАТУ.

1. Требования к системе

Общие требования к модулю «Живая лента» являются:

* надёжность и работоспособность;
* интуитивно понятный интерфейс;
* лицензионная чистота – применение средств в рамках общего лицензионного соглашения касательно корпоративного портала;
* соблюдение информационной безопасности и разграничение прав доступа к данным.

1. Требования к способам и средствам связи для информационного обмена между компонентами

Для обеспечения информационного обмена компоненты подсистемы должны взаимодействовать с объединённой информационной базой данных. Благодаря хранению данных в различных схемах, таких как мероприятия или поручения модуль «Живая лента» может объединить эти данные, представив их как единый информационный поток.

1. Перспективы развития, модернизация системы

Дальнейшим развитием модуля «Живая лента» может быть объединение иных подсистем корпоративного портала.

1. Требования к квалификации персонала и режиму его работы

Для обеспечения максимальной работоспособности пользователей должны устанавливаться перерывы:

* через 2 часа после начала смены и через 1,5–2 часа продолжительностью 15 минут;
* через каждый час работы продолжительностью 10 минут

Для эксплуатации модуля «Живая лента» определены следующие роли:

* системный администратор – должен быть квалифицированным специалистом с практическим опытом выполнения работ по администрированию программных и технических средств. В обязанности входит: установка, модернизация, настройка программного обеспечения, ведение учётных записей портала;
* администратор баз данных – должен быть квалифицированным специалистом с практическим опытом выполнения работ по администрированию СУБД, проектированию БД, оптимизации производительности, разграничению прав и ролей, а также резервного копирования и обеспечение целостности БД;
* пользователь портала – должен иметь опыт работы с персональным компьютером на уровне опытного пользователя и свободно осуществлять базовые действия с корпоративным порталом посредством браузера с доступом в интернет.

1. Требования к надёжности технических средств и программного обеспечения

Надёжность по отношению к техническим средствам должна обеспечиваться использованием в системе средств повышенной отказоустойчивости и их резервированием, а также дублированием носителей информационных банков данных.

Надёжность программного комплекса обеспечивается использованием сертифицированных операционных систем, общесистемных программных средств и инструментальных программных систем, используемых при разработке программного обеспечения. Само программное обеспечение должно обеспечивать защиту от некорректных действий пользователей и ошибочных исходных данных.

1. Требования к безопасности

Разрабатываемый информационный модуль «Живая лента» должен обеспечивать безопасный доступ к данным, предотвращая несанкционированный доступ или модифицирование данных. Модуль аутентификации должен обеспечивать защищённый доступ ко всему программному интерфейсу приложения, за исключением статичной формы авторизации в системе предоставляя возможность пройти аутентификацию в корпоративном портале при помощи логина и пароля.

Также при разработке модуля необходимо соблюдать разграничение прав на публикацию информации отправителю должны быть доступны только те адресаты, которые относятся к его зоне ответственности.

1. Требования по эргономике и технической эстетике

Модуль должен иметь удобный и интуитивно понятный графический пользовательский интерфейс. Диалоговый интерфейс должен соблюдать контекст подсистемы организационной коммуникации университета и управления в целом, тем самым действия конечного пользователя должны быть ясны и знакомы.

Пользовательский интерфейс модуля также должен аккомпанировать цветовой гамме и общему стилю корпоративного портала.

1. Требования к программному обеспечению

При проектировании информационного модуля «Живая лента» необходимо эффективно использовать используемое на данный момент в проекте корпоративного портала применяется веб-фреймворк Vue.js, в качестве серверного окружения используется программная платформа Node.js, а для хранения данных применяется СУБД PostgreSQL.

1. Требования к техническому обеспечению

Техническое обеспечение системы должно максимально и наиболее эффективно использовать существующее в отделе автоматизации оборудование:

* процессор – 2х Intel Xeon 3.7 ГГц;
* оперативная память – 32 ГБ;
* дисковая система – 1ТБ;
* сетевой адаптер – 1 Гб/с.

1. Порядок контроля и приёмки системы

Приёмо-сдаточные испытания системы проводятся с привлечением сотрудников отдела автоматизации. По результатам опытной эксплуатации оформляется акт о приёме работ. Акт содержит заключение о соответствии системы техническому заданию.

1. Требования к составу и содержанию работ подготовки объекта автоматизации к вводу системы в действие

При подготовке к вводу в эксплуатацию модуля «Живая лента» отдел управления информатизации должен обеспечить выполнение следующих работ:

* определить подразделение и ответственных должностных лиц для внедрения информационного модуля;
* обеспечить пользователей руководством, которое поможет быстрее освоить внедрённый модуль;
* провести опытную эксплуатацию модуля «Живая лента».

## Приложение Б.

Программный код

const express = require('express');

const sqlite = require('sqlite3').verbose();

let md\_index = require('../models/model-index').md\_index;

const router = express.Router();

router.get('/', (req, res) => {

    let fileDB = './private/feeds.db'; // файл БД

    const db = new sqlite.Database(fileDB);

    let query = `SELECT DATE, TIME, name, feed, rate FROM feeds ORDER BY id DESC LIMIT ?`;

    let values = [50];

    // res.cookie('cookiename', 'cookievalue', { maxAge: 900000, httpOnly: true });

    // print(res.cookie)

    // print(localStorage.getItem('test'))

    db.all(query, values, (err, records) => {

        if (err) console.log(err.message);

        // md\_index.feeds = records;

        console.log(records)

        md\_index.feeds = records;

        res.render('index.ejs', md\_index);

    });

    db.close();

});

За отображение шаблона feed.ejs для добавления отзывов, отвечает модуль feed.js, с помощью, которого пользователь добавляет новые отзыва в базу данных.

const express = require('express');

const sqlite = require('sqlite3').verbose();

const htmlParser = express.urlencoded({extended: false});

const { md\_feed } = require('../models/model-feed'); // подключаем модель данных

const router = express.Router();

router.get('/', (req, res, next) => {

    res.render('feed.ejs', md\_feed); // render view

});

router.post('/', htmlParser, (req, res) => {

    let fileDB = './private/feeds.db'; // файл БД

    let nameUser = req.body.nameUser; // находим по имени тега в шаблоне

    let feedUser = req.body.feedUser; // находим по имени тега в шаблоне

    let rateUser = req.body.userRate;

    // let string\_date = new Date()

    //     .toISOString()

    //     .replace(/T/, ' ')

    //     .replace(/\..+/, '')

    //     .split(' ')

    // let time = string\_date[0];

    // let date = string\_date[1];

    // let time = Date.parse(new Date())

    let string\_date = new Date();

    // console.log(string\_date.toISOString())

    string\_date = string\_date.toLocaleString()

    console.log(string\_date)

    let tmp1 = string\_date.split(', ')

    let time = tmp1[1];

    let b = tmp1[0].split('.');

    let date = `${b[2]}-${b[1]}-${b[0]}`

    // let date = `${string\_date.getFullYear()}.${string\_date.getMonth()}.${string\_date.getDay()}`

    // let time = '';

    // let date = '';

    console.log(req.body.nameUser, req.body.feedUser, req.body.userRate);

    const db = new sqlite.Database(fileDB);

    let query = `INSERT INTO feeds(name,feed,rate,TIME,DATE) VALUES(?,?,?,?,?)`;

    db.run(query, [nameUser, feedUser, rateUser, time, date], (err) => {

        if (err) console.log(err.message);

        res.redirect("/");

        db.close();

    });

});

Этот шаблон мы видим в браузере при переходе по адресу localhost:3000/feed. Получение значений атрибута происходит с помощью тега name. Атрибут class и id используются для привязки к различным стилям, которые прописаны в соответствующих css-файлах.

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8" />

    <title><%=title%></title>

    <link rel="stylesheet" type="text/css" href="/public/feed.css" />

    <link rel="shortcut icon" href="/public/favicon.ico" />

</head>

<body>

    <br>

    <div id="inline">

    <h2><%=heading%></h2>

    <form action="/feed" method="post"> <!-- вызов события /feed -->

        <input id="name" class="txt" name="nameUser" type="name" placeholder="Ваше имя" required>

        <input class="txtnumber" min="1" max="10" name="userRate" type="number" placeholder="Выберете число от 1 до 10" required>

        <textarea id="feed" class="txtarea" name="feedUser" placeholder="Ваш отзыв:"></textarea required>

        <button id="send"><%=button%></button>

    </form>

    <form> <!-- вызов главной страницы - событие / -->

        <button id="back" type="submit" formaction="/">Вернуться</button>

    </form>

    </div>

</body>

</html

Этот шаблон используется при переходе по адресу localhost:3000, тут реализовано отображение данных из таблицы базы данных и возможность различных сортировок, при клике на заголовки таблицы.

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8" />

    <title><%=title%></title>

    <link rel="stylesheet" type="text/css" href="/public/feed.css" />

    <link rel="shortcut icon" href="/public/favicon.ico" />

</head>

<body>

    <br>

    <div id="inline">

        <br>

        <form action="/feed" method="get"> <!-- вызов события /feed -->

            <button id="send"><%=button%></button>

        </form>

        <br>

        <h4><%=heading%></h4>

        <table id="myTable2">

            <tr>

                <th onclick="sortTable(0)">Дата</th>

                <th onclick="sortTable(1)">Время</th>

                <th onclick="sortTable(2)">Имя</th>

                <th>Отзыв</th>

                <th onclick="sortTable(4)">Оценка продукта</th>

            </tr>

            <% for (let feed of feeds) { %>

            <tr>

              <td class="date"><%=feed.DATE%></td>

              <td class="time"><%=feed.TIME%></td>

              <td class="name"><%=feed.name%></td>

              <td class="feed"><%=feed.feed%></td>

              <td class="rate"><%=feed.rate%></td>

            </tr>

            <%} %>

        </table>

    </div>

<script>

    function sortTable(n) {

      var table, rows, switching, i, x, y, shouldSwitch, dir, switchcount = 0;

      table = document.getElementById("myTable2");

      switching = true;

      // Set the sorting direction to ascending:

      dir = "asc";

      while (switching) {

               switching = false;

        rows = table.rows;

        /\* Loop through all table rows (except the

        first, which contains table headers): \*/

        for (i = 1; i < (rows.length - 1); i++) {

          // Start by saying there should be no switching:

          shouldSwitch = false;

          /\* Get the two elements you want to compare,

          one from current row and one from the next: \*/

          // alert(rows[i].getElementsByTagName("TD")[n].innerHTML);

          x = rows[i].getElementsByTagName("TD")[n];

          y = rows[i + 1].getElementsByTagName("TD")[n];

          /\* Check if the two rows should switch place,

          based on the direction, asc or desc: \*/

          if (dir == "asc") {

            if (x.innerHTML.toLowerCase() > y.innerHTML.toLowerCase())

              shouldSwitch = true;

              break;

            }

          } else if (dir == "desc") {

            if (x.innerHTML.toLowerCase() < y.innerHTML.toLowerCase()) {

               shouldSwitch = true;

              break;

            }

          }

        }

        if (shouldSwitch) { /\* If a switch has been marked, make the switch

          and mark that a switch has been done: \*/

          rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);

          switching = true

          switchcount ++;

        } else {/\* If no switching has been done AND the direction is "asc",

          set the direction to "desc" and run the while loop again. \*/

          if (switchcount == 0 && dir == "asc") {

            dir = "desc";

            switching = true;

          }

        }

      }

    }

    </script>

</body>

</html>

Модель данных, которая используется при заполнении данными шаблона отправки отзыва

\const md\_feed = {

    title: "Оставить отзыв",

    heading: "Форма онлайн отзыва",

    button: "Отправить отзыв"

};

module.exports = { md\_feed }

Модель данных, которая содержит данных для заполнения шаблонов, может дополняться другими данными для отображения. Используется на главной странице приложения, при просмотре отзывов.

const md\_index = {

    title: "Отзывы о продукте",

    heading: "Отзывы",

    button: "Оставить новый отзыв",

    feeds: []

};

"debug": {

"port": 3000,

"hostname": "127.0.0.1"

},

config.json – файл хранящий некоторые настройки приложения

Приложение В

РЕЦЕНЗИЯ

на работу \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

доцент кафедры ИТиПИ, к.т.н., доцент

Беляков Андрей Юрьевич