

Содержание

Введение	3
1. Постановка цели и формулировка задач	4
2. Анализ и сравнительный обзор аналогов.....	5
Веб приложение МОЙЭКОДВОР.РФ.....	5
Веб приложение RSBOR.RU	5
Веб приложение TAGRET99.BY.....	6
3. Проектирование программного обеспечения	8
Проектирование архитектуры программного средства	8
Проектирование серверной части	9
Проектирование клиентской части	11
Проектирование базы данных	11
Заключение	14
Список использованных источников	15

Введение

Innowise Group – одна из наиболее опытных белорусских компаний в области полного цикла разработки, внедрения и поддержки ПО, бизнес-анализа, дизайна, тестирования и управления проектами.

Компания основана в 2007 г. За это время мы реализовали 600+ проектов для клиентов из США, Канады, Германии, Японии (всего более 30 стран). В портфолио Innowise Group успешные кейсы в сферах Enterprise, eCommerce, FinTech, MedTech, Media and Entertainment и других доменах. Центры разработки Innowise находятся в 6 городах Беларуси (Минск, Витебск, Гомель, Брест, Могилев и Полоцк), а также в Польше, Литве, Украине, Грузии и Германии.[1]

Компания создает и предоставляет ориентированные на результат решения и услуги, идет в ногу с передовыми инновациями и поддерживает долгосрочные отношения, которые приносят пользу клиентам в долгосрочной перспективе. Собственный проект VOKA.IO, основанный на применении технологий Mixed Reality в области хирургии, победил в категории “Здоровье” всемирного конкурса инновационных проектов World Summit Awards 2019 и получил международное признание от врачей всего мира.

Старшие разработчики полного стека, умеющие использовать множество технологий и фреймворков, имеют более чем пятнадцатилетний опыт работы и работают вместе со многими другими специалистами среднего и младшего звена, что позволяет молодым специалистам быстро получить опыт коммерческой разработки.

Рост компании – это в первую очередь рост команды. Успех компании – это результат усилий всех сотрудников.

Целью производственной преддипломной практики является изучение стандарта БГТУ по дипломному проектированию. Необходимо провести анализ информационных источников (системы, методы, алгоритмы, программные решения, технологии программирования) по теме дипломного проекта. Также нужно провести анализ требований, разработать функциональные требования и архитектуру проектируемого приложения. И выполнить реализацию приложения в соответствии с индивидуальным заданием.

Итогом же данной практики будет полностью функционирующее и логически завершенное веб-приложение «ЭкоБудущее» с соответствующим отчетом.

1. Постановка цели и формулировка задач

Сегодня каждый белорус выбрасывает примерно 350 кг бытовых отходов за год. Это в два раза больше, чем 15 лет назад. К тому же состав мусора кардинально изменился. Еще совсем недавно домашняя мусорная корзина содержала в основном пищевые и бумажные отходы, старая одежда была из натуральных тканей, а техника в мусоре почти не оказывалась.

Современный бытовой мусор – это разнообразный пластик, искусственные материалы, из которых сделана упаковка, одежда, домашние вещи, батарейки, электроника. При захоронении эти вещи разлагаются сотни лет, а может, и больше, выделяя токсичные вещества в почву, воду и воздух. Полиэтиленовый пакет, который послужил вам 20 минут, останется на мусорном полигоне и через 100 лет. При этом многие бытовые отходы можно переработать, пустить в дело и благодаря этому сократить объемы захоронения мусора. И к тому же сэкономить природные ресурсы. Переработка отходов – это необходимость сегодня.

На производство товаров тратятся ресурсы: нефть, древесина, чистая вода, топливо для перевозки и электричества. Раздельный сбор даёт возможность этим ресурсам не пропадать. Для системы раздельного сбора почти все отходы — это вторичное сырьё, которое можно ещё не раз использовать для производства новой продукции.

Раздельный сбор положительно влияет на образ жизни людей. Вы:

- адекватно оцениваете свои расходы;
- меньше подвержены спонтанным покупкам;
- экономите деньги на том, что на самом деле вам не нужно, но что навязано быстрой модой;
- четко видите, что вы покупаете, сколько тратится на упаковку.

Тема раздельного сбора бытовых отходов в последние года становится все более актуальной. Но, к большому сожалению, в нашей стране данной проблеме уделяется недостаточно внимания. В западных странах в каждом доме и квартире имеются необходимые резервуары для раздельного сбора мусора.

Цель разработки курсового проекта – создание веб-приложения, способствующего популяризации сортировки бытовых отходов.

Задачи, решение которых поможет достичь цели:

- определить обязательный функционал приложения;
- согласно требованиям спроектировать составляющие приложения;
- разработать приложение, согласно требованиям;
- протестировать функционал приложения, на наличие ошибок;
- составить руководство пользования для приложения.

Главная задача программного средства состоит в том, чтобы заинтересовать пользователя начать сортировать бытовые отходы и сдавать их в специальные пункты приема, помимо этого пользователи могут публиковать статьи с советами по сбору и сортировке отходов. Администратор может контролировать процессы, происходящие в приложении.

2. Анализ и сравнительный обзор аналогов.

На сегодняшний день с легкостью можно найти ресурсы где так или иначе занимаются информированием или привлечением к раздельному сбору мусора. Для обзора схожих по тематике ресурсов были выбраны следующие веб-приложения:

- МОЙЭКОДВОР.РФ [2];
- *RSBOR.RU* [3];
- *TARGET99.BY* [4];

Веб приложение МОЙЭКОДВОР.РФ

Рассматриваемый интернет-ресурс представляет собой сайт, главная страница которого отображена на рисунке 1.1.

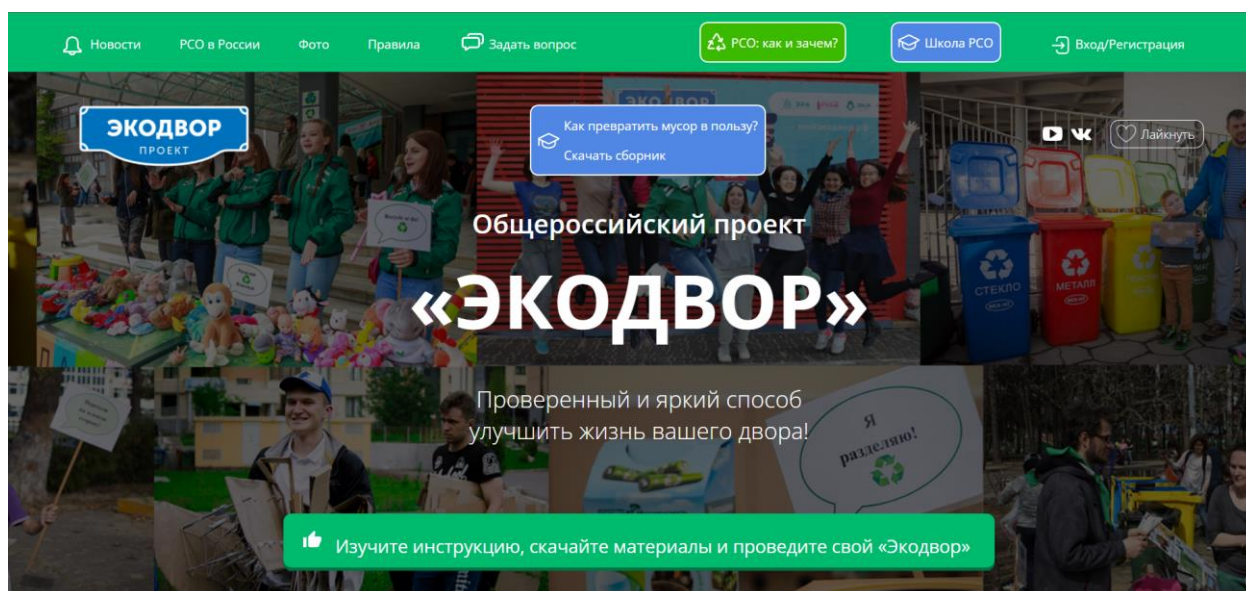


Рисунок 2.1 – Главная страница веб-приложения МОЙЭКОДВОР.РФ

Анализируемое веб-приложение позволяет пользователям ознакомиться с рекомендациями по раздельному сбору отходов, просмотреть видео уроки под названием «Школа PCO», увидеть фото с событий, проводимых участниками проекта «Экодвор». Так же ресурс предлагает любому желающему стать частью этого сообщества и иметь возможность проводить свой «Экодвор».

К плюсам приложения можно отнести обилие на ресурсе материалов, показывающих проводимую работу с населением, что мусору можно дать «вторую жизнь». И что это важно и в тоже время просто.

Что хотелось бы добавить на ресурс – это возможность комментирования статей, для анализа недостатков и усовершенствования проекта.

Веб приложение RSBOR.RU

Следующим результатом поиска аналогов стал сайт *RSBOR.RU*, главная страница которого представлена на рисунке 1.2.

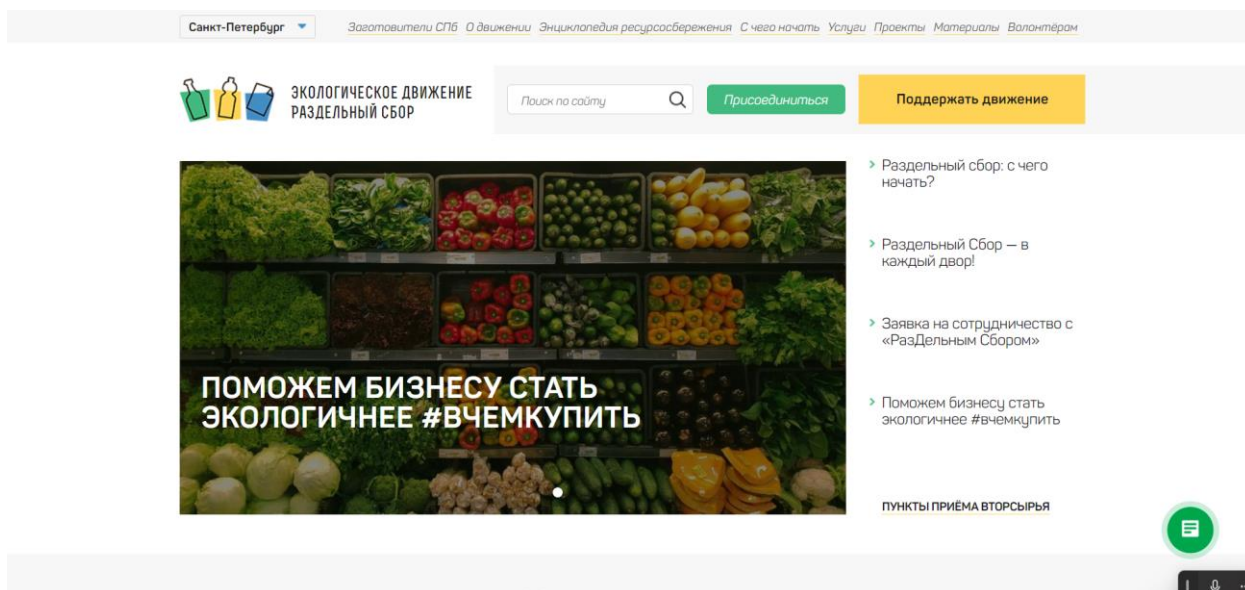


Рисунок 2.2 – Главная страница сайта RSBOR.RU

RSBOR.RU больше похож на новостной ресурс. На главной странице представлены последние новости в сфере сбора, разделения и переработки отходов и вторсырья в Санкт-Петербурге. На других страницах можно ознакомиться с зарубежным опытом переработки отходов и ресурсосбережения. Также на сайте имеется возможность увидеть проекты, которые проводит Экологическое Движение Раздельный Сбор.

Для тех, кто желает поддержать движение представлена возможность сделать пожертвования для развития движения.

Так же есть возможность скачать различные материалы для пропаганды заботы об окружающей среде.

Еще одним важным шагом в развитии «Раздельного Сбора» является предоставленная возможность стать волонтером. Для этого нужно нажать кнопку “Присоединиться” и заполнить анкету.

Интересный момент, который есть у «Раздельного Сбора» - это возможность компаниям и индивидуальным предпринимателям зарабатывать на сборе вторичного сырья. Для этого нужно иметь возможность размещения и обслуживания придомовых контейнеров. Стать партнером может каждый, кто хочет открыть новый бизнес, так и компании, давно работающие в различных сферах.

Что хотелось бы добавить на ресурс – это чтобы человек не просто выбрасывал мусор для вторичной переработки, но и имел за это какие-то бонусы.

Веб приложение TAGRET99.BY

После первых двух приложений я обратилась к белорусскому ресурсу государственного учреждения «Оператор вторичных материальных ресурсов», уполномоченного министерством жилищно-коммунального хозяйства

Республики Беларусь. Для большего понимания какое развитие имеет ситуация с вторсырьем в нашей стране. Данный веб-ресурс представлен на рисунке 1.2.

Следить за движением Цель 99 можно через многие социальные сети такие как: *Telegram*, *Instagram*, ВКонтакте, *Facebook* и на *YouTube* канале. Для связи есть многоканальный номер телефона.

Основная информация, представленная на главной странице сайта – это правила сортировки вторичных ресурсов. Для удобства правила разделены по категориям.

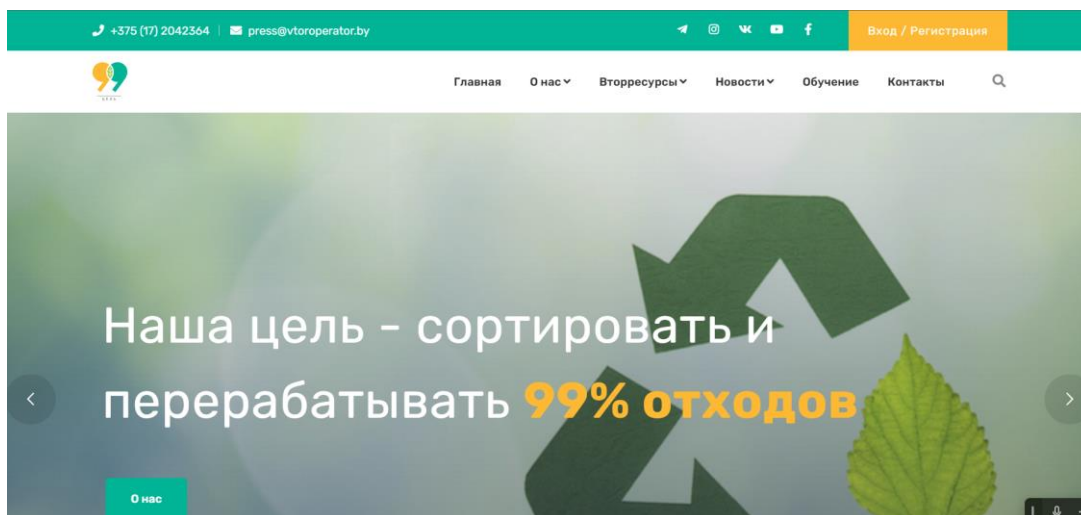


Рисунок 2.3 – Страница сайта TAGRET99.BY

Важный момент для развития отдельного выброса мусора – это подробная карта пунктов приема вторичного сырья. В Беларуси сегодня более 1700 приемных пунктов и постоянно пополняется новыми объектами.

Калькулятор экологического эффекта – еще один полезный раздел для тех, кто хочет знать о пользе переработки в цифрах.

Так же на ресурсе можно ознакомиться с последними новостями о переработке отходов и экологичном стиле жизни.

Приложение оставило приятное впечатление. На ресурсе есть много статей, в которых информация представлена в доходчивой форме. Особое впечатление «калькулятор экологического эффекта».

На данном сайте достаточно глубоко освещен вопрос сортировки и переработки вторсырья и какие-либо недостатки почти отсутствуют.

3. Проектирование программного обеспечения

Разработка архитектуры проекта – важная задача в процессе работы над приложением, потому что в зависимости от неё определяется уровень связности между компонентами приложения, и насколько легко можно будет это приложение расширить. Также необходимо учитывать множество факторов, влияющих на работоспособность разрабатываемого веб-приложения. Современные приложения должны иметь интуитивно понятный и удобный интерфейс, хорошую функциональность; должны быть просты и гибки в использовании, оптимизированы и в большинстве случаев автоматизированы.

Для создания программного средства была выбрана среда разработки *WebStorm*[5] – это один из наиболее популярных редакторов кода. Он распространяется в бесплатном доступе для студентов и поддерживается всеми актуальными операционными системами: *Windows*, *Linux* и *macOS*. *WebStorm* это интегрированная среда для разработки на JavaScript и связанных с ним технологий. Как и другие IDE JetBrains, *WebStorm* позволяет автоматизировать рутинную работу и легко справляться со сложными задачами, делая разработку более увлекательной.

Основная задача в ходе реализации курсового проекта является создание сервиса, позволяющего пользователям узнать больше и заинтересоваться сортировкой своих же бытовых отходов, а администратору – управлять внутренними процессами данного программного средства. Таким образом, было принято решение создать три роли: гость, пользователь, администратор. Каждая из этих ролей имеет ряд функций:

- гость – может только просмотреть статьи, опубликованные пользователями и прочитать комментарии;
- пользователь – имеет возможность просматривать все статьи, публиковать, удалять и редактировать свои статьи, оценивать статьи, комментировать, отправлять в базу данных приложения количество сданных отходов и получать скидку от партнеров приложения, а также получать информацию сколько новой продукции будет произведено из переработанных отходов;
- администратор – также имеет возможность публиковать, просматривать, оценивать и комментировать все статьи. Так же имеет право добавить новую точку сбора, добавить новые скидки и просмотреть все скидки.

Проектирование архитектуры программного средства

Программное средство, разработанное в ходе выполнения курсовой работы, является представителем приложений с архитектурой «клиент-сервер». Логика, отвечающая за обработку данных, а именно создание, удаление, обновление и получение записей из базы данных расположена на сервере в виде файлов контроллеров и их методов, написанных на языке *JavaScript*.

Клиент реализован при помощи библиотеки *React JS*, которая позволяет собирать сложный *UI* из изолированных кусочков кода – называемых компонентами. Компоненты представляют собой *HTML* разметку, которая применяет

CSS. Цель *CSS* – предоставить высокую скорость, простоту и масштабируемость. К особенностям можно отнести однонаправленную модель передачи данных, виртуальный *DOM*, *JSX*, *React Hooks*, динамическую верстку.

Взаимодействие клиента с сервером происходит посредством использования протокола *HTTP*. *HTTP* – это протокол прикладного уровня передачи данных различного формата. Клиент инициирует *GET* запрос нажатием на какой-либо пункт в навигационном меню и сервер *Node JS* генерирует соответствующий ответ, основываясь на *URI* запроса. Межкомпонентное взаимодействие представлено в виде *MVC*, где *model* – это объекты *MS SQL*, *controller* – контроллеры сервера *Node JS*, а *view* – компоненты *React JS*.

Как только клиент инициирует какой-либо сценарий, то посредством использования *HTTP* происходит передача данных контроллеру на стороне сервера, который осуществляет взаимодействие с базой данных. После завершения работы управление возвращается клиенту.

Проектирование серверной части

При проектировании структуры сервера важно сделать так, чтоб она была читабельной и легко расширяемой. Для данных целей подходит архитектура *MVC*.

Controller выполняет функцию обработчика запросов. Каждый файл контроллера представляет собой *JS*-модуль, содержащий функции, которые обрабатывают запрос. В таблице 2.1 представлены и описаны все контроллеры, разработанные в курсовой работе.

Таблица 2.1 – Описание контроллеров

Название кон- троллеров	Описание
<i>ArticlesController.js</i>	Предназначен для функций, обеспечивающих работу со статьями.
<i>AuthController.js</i>	Функции предназначены для аутентификации, регистрации пользователей и при выполнении этих функций выполняется формирование токенов.
<i>DiscountsController.js</i>	Предназначен для функций, обеспечивающих работу со скидками, а в частности создание, удаление, изменение и вывод их для пользователя.
<i>MarksController.js</i>	Предназначен для управления данным по количеству начисления баллов за сдачу отходов и произведенной новой продукции.
<i>PointsController.js</i>	Предназначен для вывода данных по точкам сбора отходов.

Окончание таблицы 2.1

Название кон-троллера	Описание
<i>RatingsController.js</i>	Используется для управления комментариями.
<i>ReceptionController.js</i>	Функция применяется для начисления баллов и вывода количества произведенной новой продукции из сданных отходов. Данная функция представлена в приложении В.
<i>Check_weightsController.js</i>	Используется для проверки правильно введенного веса.
<i>KeysController.js</i>	Функция добавления секретных ключей.
<i>Used_discountsController.js</i>	Применяется для начисления скидок пользователям.

Маршрутизация в приложении организована с помощью роутеров, которые используют технологию *Express JS*. Маршрутизация определяет, как приложение отвечает на клиентский запрос к конечной точке, которым является *URI*, и определенному методу запроса *HTTP (GET, POST, DELETE)*. Для каждого из контроллеров был разработан собственный маршрутизатор. Он представляет собой *JS*-модуль, который в зависимости от контроллера будет передаваться в *middleware* сервера с указанием первого *URL*-параметра.

В таблице 2.2 представлены и описаны все разработанные маршрутизаторы.

Таблица 2.2 – Описание маршрутизаторов

Название маршрутизатора	Описание
<i>ArticlesRouter.js</i>	Роутер, который перенаправляет запрос, приходящий по пути <i>/Articles</i> .
<i>AuthRouter.js</i>	Перенаправляет запросы, приходящие по пути <i>/login, /register, /logout, /me</i> .
<i>Check_weightsRouter.js</i>	Роутер, перенаправляет запрос по <i>URL /Weight</i>
<i>DiscountsRouter.js</i>	Маршрутизатор обрабатывает запросы по <i>URL /Discounts</i> .
<i>KeysRouter.js</i>	Перенаправляет запросы, приходящие по пути <i>/Keys</i>
<i>MarksRouter.js</i>	Роутер, перенаправляет запрос приходящий с <i>/Marks</i> .
<i>PointsRouter.js</i>	Перенаправляет запрос, приходящий по пути <i>/Points</i> .
<i>RatingsRouter.js</i>	Маршрутизатор обрабатывает запросы по <i>URL /Ratings</i> .
<i>ReceptionsRouter.js</i>	Перенаправляет запросы, приходящие по пути <i>/Receptions</i>
<i>Used_discountsRouter.js</i>	Маршрутизатор для <i>URL /UDiscount</i>

Вся серверная часть представляет собой веб-приложение, предоставляющее интерфейс для взаимодействия с ним клиентского приложения. Для корректной работы сервера используются контроллеры и маршрутизаторы, которые вместе предоставляют полную обработку и перенаправление приходящих запросов от клиента.

Проектирование клиентской части

Клиентская часть веб-приложения реализована с помощью *React JS*. Для разработки пользовательских интерфейсов *React JS* часто используется с другими библиотеками.

Структура клиентского приложения представлена на рисунке 2.1

Директорий *components* содержит все компоненты, используемые в приложении, например, компоненты форм, модальных окон;

Директорий *pages* содержит все страницы, используемые в приложении;

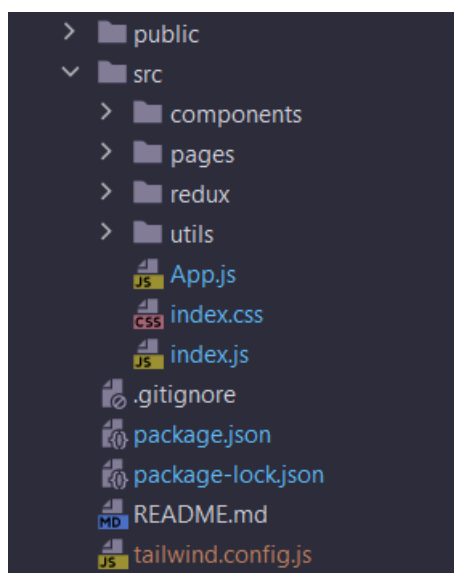


Рисунок 3.1 – Структура клиентского приложения

Директорий *redux* содержит все объекты *Redux*, используемые в приложении;

Остальные файлы, представленные на рисунке выше, являются базовыми и создаются при инсталляции проекта *React JS*.

Проектирование базы данных

В качестве базы данных в проекте используется MySQL. Для реализации функционала приложения была разработана база данных. Для создания базы данных была выбрана система управления базами данных MySQL. Текущая база данных состоит из 7 таблиц, схема базы данных продемонстрирована в приложении Г.

Таблица *users* хранит информацию о пользователях, содержит следующие поля:

– *id*, уникальное поле в котором указывается идентификатор пользователя;

– *username*, хранится имя пользователя;

– *email*, хранится почта пользователя;

– *password_hash*, хранится пароль пользователя в захешированном виде

– *points*, хранятся количество начисленных баллов;

– *avatar_url*, хранится адрес на фото пользователя;

– *role*, хранится роль пользователя;

– *is_activated*, хранится значение указывающее была ли активирована почта;

– *activation_link*, хранится строка активации почты.

Таблица *articles* хранит в себе статьи и информацию, предназначенную для них, содержит следующие поля:

– *id*, уникальное поле в котором указывается идентификатор статьи;

– *title*, название статьи;

– *text*, текст статьи;

– *date_of_pub*, дата публикации статьи;

– *image_url*, картинка, относящаяся к статье;

– *author*, автор статьи;

– *like*, количество лайков у статьи.

Таблица *ratings* хранит в себе комментарии к статьям, содержит следующие поля:

– *id*, уникальное поле в котором указывается идентификатор комментария;

– *article_id*, идентификатор статьи к которой относится комментарий;

– *commentator*, идентификатор пользователя который оставил комментарий;

– *comment*, текст комментария.

Таблица *points* хранит в себе пункты сдачи отходов, содержит следующие поля:

– *id*, уникальное поле в котором указывается идентификатор точки сдачи отходов;

– *address*, адрес точки сдачи отходов;

– *time_of_work*, время работы;

– *key_idy*, хранится идентификатор на секретный ключ относящийся к точке сдачи отходов в захешированном виде

– *admin_id*, идентификатор администратора, который добавил эту точку.

Таблица *s_key* хранит секретные ключи:

– *id*, уникальное поле в котором указывается идентификатор секретного ключа;

– *secret_key*, секретный ключ в захешированном виде;

– *used*, значение указывающее использовался ли данный ключ.

Таблица *receptions* хранит информацию по сдаче отходов, содержит следующие поля:

- *id*, уникальное поле в котором указывается идентификатор приема;
- *id_user*, идентификатор пользователя, который сдал отходы;
- *weight*, сколько по весу было сдано отходов в кг;
- *cccrued*, сколько начислилось баллов;
- *new_kg*, сколько новой продукции будет произведено из сданных отходов в кг;
- *type_waste*, тип сданных отходов;
- *station_key*, секретный ключ в захешированном виде, предоставленный пользователем;
- *weght_key*, ключ подтверждающий вес.

Таблица *discounts* хранит в себе скидки, содержит следующие поля:

- *id*, уникальное поле в котором указывается идентификатор скидки;
- *discount*, скидка;
- *count_for_dnt*, сколько нужно баллов для получения скидки.

Таблица *used_discounts* в ней хранятся скидки доступные пользователям:

- *id*, идентификатор записи;
- *discount_id*, идентификатор скидки;
- *user_id*, идентификатор пользователя;
- *used*, значение указывающее использовалась эта скидка пользователем или нет.

Таблица *marks* хранит в данные по «цене» и количеству произведенной новой продукции, содержит следующие поля:

- *id*, уникальное поле в котором указывается идентификатор;
- *rubbish*, вид отхода
- *points_per_kg*, сколько баллов начисляется за один кг сданных отходов;
- *new_from_kg*, сколько новой продукции будет произведено из одного кг сданных отходов.

Таблица *chek_weight* содержит секретные ключи используемые для проверки введенного веса пользователем:

- *id*, идентификатор записи;
- *id_rubbish*, идентификатор вида отходов;
- *weight*, вес для подтверждения в килограммах;
- *key_of_weight*, секретный ключ в захешированном виде.

Заключение

По итогу производственной преддипломной практики был разработано веб-приложение, позволяющее пользователям контролировать объем сдаваемых бытовых отходов, накапливать и получать скидки, предусмотренные ресурсом, узнавать больше о сортировке своих же бытовых отходов. А для администратора создана роль – управлять внутренними процессами данного программного средства. Данное приложение является конечной целью этой практики, в процессе реализации которого были выполнены следующие задачи:

- реализация разделения возможностей пользователей по ролям;
- предоставление пользователю с ролью администратора возможности добавлять и комментировать статьи, добавления и удаления пунктов сбора вторичных отходов, добавление, удаление и изменение скидок;
- предоставление авторизованным пользователям возможности отслеживать объем сданного вторсырья, количества накопленных баллов и получения скидок, добавление и комментирование статей.

Кроме реализации приложения, в результате разработки проекта были изучены новые технологии, а также укреплены старые знания. Были улучшены навыки разработки на платформе Node.js, библиотеке React.js.

На основе вышесказанного можно утверждать, что цели производственной преддипломной практики были выполнены в полном объеме.

Список использованных источников

- 1 INNOWISE-GROUP.COM [Электронный ресурс] / Главная страница. – Режим доступа: <https://innowise-group.com/> - Дата доступа: 14.04.2023.
- 2 МОЙЭКОДВОР.РФ [Электронный ресурс] / Главная страница. – Режим доступа: <https://xn--b1ad0eimccs8j.xn--p1ai/>. – Дата доступа: 14.04.2023.
- 3 RSBOR.RU [Электронный ресурс] / Главная страница. – Режим доступа: <https://rsbor.ru/>. – Дата доступа: 14.04.2023.
- 4 TAGRET99.BY [Электронный ресурс] / Главная страница. – Режим доступа: <http://www.target99.by/>. – Дата доступа: 14.04.2023.
- 5 JETBRAINS.COM/RU-RU/WEBSTORM/ [Электронный ресурс] / Главная страница. – Режим доступа: <https://www.jetbrains.com/ru-ru/webstorm/> – Дата доступа: 14.04.2023.