Содержание

[ВВЕДЕНИЕ 5](#_Toc74475746)

[1. Программные средства и инструменты 7](#_Toc74475747)

[1.1. Битрикс24 7](#_Toc74475748)

[1.2. Битрикс 24 CRM 8](#_Toc74475749)

[1.3. Битрикс24 САЙТЫ 9](#_Toc74475750)

[1.4. Javascript 10](#_Toc74475751)

[1.5. Bootstrap 11](#_Toc74475752)

[1.6. PHP и Rest API 11](#_Toc74475753)

[2. Проектирование системы 13](#_Toc74475754)

[2.1. Разработка CRM 20](#_Toc74475755)

[2.2. Разработка приложения импорта 28](#_Toc74475756)

[3. Создание публичного сайта 35](#_Toc74475757)

[3.1. Страница «об ассоциации» 35](#_Toc74475758)

[3.2. Страница «Лучшие выпускники» 37](#_Toc74475759)

[3.3. Страница «Новости» 38](#_Toc74475760)

[3.4. Страница «Обратная связь» 39](#_Toc74475761)

[3.5. Страница «Реквизиты» 40](#_Toc74475762)

[ЗАКЛЮЧЕНИЕ 41](#_Toc74475763)

[СПИСОК ЛИТЕРАТУРЫ 42](#_Toc74475764)

[ПРИЛОЖЕНИЕ А.ТЕКСТ ПРОГРАММЫ СЕРВЕРА 43](#_Toc74475765)

[ПРИЛОЖЕНИЕ Б.ТЕКСТ ПРОГРАММЫ СТРАНИЦЫ ЗАГРУЗКИ 55](#_Toc74475766)

# ВВЕДЕНИЕ

В современном бизнесе необходимость автоматизация различных процессов стала уже привычным явлением. Уже становится сложно представить себе складской или бухгалтерский учет без применения специализированного программного обеспечения, торговые представители используют специальные приложения для оформления и отправки заказа в офис прямо с планшета или мобильного телефона, достаточно большая часть заказов приходит с сайта уже в виде готовых к обработке документов. Но при этом взаимоотношения с клиентами, по крайней мере, в среднем и малом бизнесе, почему-то очень часто ведутся без внедрения автоматизации и достаточного внимания к учету.

Проблема качества подготовки специалистов в вузе и их последующее трудоустройство по специальности является важнейшей задачей вузов. В связи с этим актуальной является задача организации эффективного взаимодействия между вузом, его выпускниками и потенциальными работодателями на базе web ­ ориентированной информационной системы. Такая система позволит вузам вести аналитику о дальнейшем трудоустройстве своих выпускников, их должностях, использовать эти данные в качестве рекламы учебного заведения.

В качестве такой информационной системы можно использовать CRM систему. CRM-система (Customer Relationship Management или Управление отношениями с клиентами) — это прикладное программное обеспечение для организаций, предназначенное для автоматизации взаимодействия с заказчиками (клиентами), в частности, для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путем сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов.

Проанализировав существующие CRM системы, было решено реализовать базу данных выпускников на платформе Битрикс24. Во­ первых, эта система позволяет реализовать все необходимые данные системы анализа выпускников. Во­ вторых, Битрикс24 отечественный продукт и имеет достаточно подробную документацию по использованию. В­ третьих, Битрикс24 имеет свой конструктор сайтов, связанных с CRM-системой.

Для выпускной квалификационной работы мне требуется спроектировать базу данных выпускников, реализовать ее в CRM­ системе со всеми ее настройками, реализовать публичный сайт ассоциации выпускников, позволяющий обновлять информацию о себе, разработать программу для автоматического создания базы данных на новом портале Битрикс24.

Разрабатываемая система должна удовлетворять следующим требованиям:

1. Учебному заведению вести учет выпускников– по году выпуска, году поступления, кафедре, факультету, направлению/специальности, и, возможно, по учебным группам
2. Хранить информацию о выпускнике: дата рождения, дата поступления, где работал (списком), должности, даты работы, соцсети, телефон, почта, где живет/работает сейчас, фото на момент выпуска / фото сейчас
3. Хранить информацию о выпускнике – основные достижения, членства в значимых сообществах
4. Реализовывать регулярную связь с выпускником, обновлять информацию в ручном режиме (путем звонка/сообщения в мессенджер)
5. Иметь публичный сайт с историями успеха выпускников, доской почета, списком выпускников по годам, факультетам, специальностям, кафедрам, возможность подать запрос и возможность обновить информацию о себе (через ручное (план минимум) и автоматизированное редактирование (с подтверждением администратором))
6. Делать рассылки, опросы
7. Вести учет выпускников, состоящих в ассоциации, и их членских взносов.

# Программные средства и инструменты

## Битрикс24

Битрикс24 – это огромный корпоративный портал, который пытается охватить практически все. Здесь заявлены функциональные возможности социальных сетей, проектов, задач, управления персоналом, и многое другое, в том числе CRM.

Битрикс24 включает в себя большой ряд возможностей, таких как:

«Живая лента» – представляет собой страницу, на которой выводятся последние события портала: новости, комментарии, новые файлы, системные события и т.д. В ней можно просматривать все события, которые к вам относятся, за все время жизни портала.

«Живая лента» позволяет создавать задачи, выражать благодарность, запускать бизнес-процессы, оставлять комментарии к событиям, ставить лайки, устраивать голосования, оповещать ваших коллег о важных событиях и объявлениях.

Главная задача Живой ленты ­ создать общение в режиме реального времени и упростить взаимодействие внутри компании.

«Чат» – это встроенный мессенджер Битрикс24, позволяющий вести общение внутри компании как один ­ на ­ один, так и группой сотрудников, позволяет передавать файлы, создавать задачи и встречи из диалогов.

«Диск» – это «облачное» хранилище, позволяющее разместить всю информацию компании в одном месте. Использование Битрикс24.Диск позволяет работникам не использовать ресурсы своего компьютера, настраивать отдельный сервер для хранения данных компании ­ все можно хранить в одном, защищенном пространстве, доступ к которому возможен с любого электронного, в том числе мобильного, устройства, и из любой точки мира.

«Битрикс24.Диск» позволяет в режиме реального времени использовать информацию, сохраненную на диске, и подключать к работе клиентов или партнеров компании.

«Задачи и проекты» – помогают работать вместе и успевать вовремя. Функция разделения задач и проектов на роли поможет лучше спланировать работу, легко и быстро ставить задачи себе и подчиненным. Битрикс24 дает возможность поставить задачу сотруднику, установить сроки на выполнение, и контролировать этапы ее выполнения.

«Учёт рабочего времени» ­ система учета рабочего времени интегрирована с органайзером, документооборотом, телефонией и системой управления взаимоотношениями с клиентами (CRM). Собранная статистика автоматически передается в виде отчетов руководителю и в бухгалтерию. На основании данных учета использования рабочего времени начисляется зарплата. Также можно получить затраченное время на конкретную задачу и в дальнейшем использовать эти данные для ориентировки заказчика на сроки выполнения того или иного проекта, включающего в себя множество подзадач.

## Битрикс 24 CRM

CRM (Customer Relationship Management) – система управления отношениями с клиентами и автоматизации взаимодействия с заказчиками. CRM система вводится для улучшения обслуживания клиентов, повышения продаж, оптимизации процессов внутри компании.

Основной целью системы является помощь в работе с клиентами, организации и планировании бизнес-процессов, а также мониторинг выполнения задач. В CRM можно автоматизировать процессы, а также отслеживать входящие звонки по электронной почте и новые запросы с веб–сайтов, чтобы не пропустить ни одного клиента.

Посредством CRM-системы можно выстраивать диалог с клиентом исключая возможные ошибки, как результат возрастает эффективность работы менеджеров и всей компании, а количество продаж растет.

В CRM-системе для учёта данные хранятся не в огромных таблицах, а в удобных карточках, и вся история взаимодействия с клиентом представлена в хронологическом порядке.

В CRM Битрикс24 есть следующие сущности:

1. «Контакты» ­ Лид принятый в работу и готовый совершить сделку;
2. «Компании» ­ Сущность CRM, которая содержит в себе контакты, лиды и сделки. Контакты привязаны к одной или к нескольким компаниям;
3. «Лиды» ­ потенциальные клиенты, которые еще не совершили сделку, но уже проявили интерес к компании;
4. «Предложение»­ ознакомление потенциального покупателя с предметом сделки;
5. «Сделка»­ Это конечная цель и желаемый результат работы всей системы управления взаимодействием с клиентами;
6. «Счет»­ финальный этап сделки. Здесь клиенту выставляется счет для оплаты текущей сделки.

## Битрикс24 САЙТЫ

Битрикс24 имеет свой встроенный конструктор сайтов, на котором можно создать сайт практически под любые цели и задачи.

Конструктор сайтов Битрикс24 имеет ряд преимуществ: во­ первых, хостинг для сайтов, а также конструктор полностью бесплатные; во­ вторых сайты, созданные при помощи Битрикс24, можно легко перенести на другой домен с полным сохранением всего функционала; в­ третьих это простая интеграция с любым инструментом портала Битрикс24; в­ четвертых, страницы, сделанные в конструкторе, SEO оптимизированы. Также есть возможность подключить на сайт онлайн чат поддержки клиентов или CRM-форму обратной связи. Данные формы заносятся в привязанные элементы CRM (Лиды, Контакты и др.).

На рисунке 1 представлены шаблоны сайтов Битрикс24.

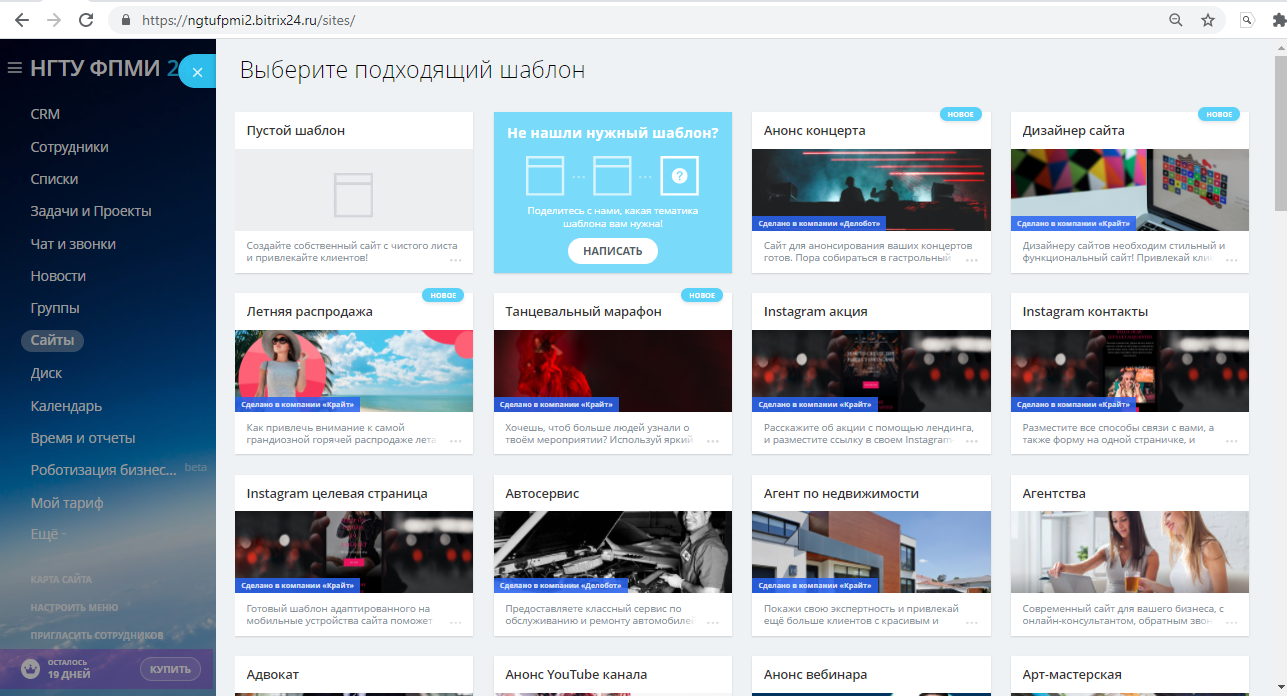


Рисунок 1 ­— Битрикс24.Сайты

## Javascript

JavaScript — это кросс­платформенный, объектно-ориентированный скриптовый язык, позволяющий сделать веб-страницы интерактивными и динамическими.

JavaScript дает разработчику веб-сайта возможность управлять поведением страницы. Поэтому JavaScript принципиально другой язык нежели HTML, отвечающий за структуру веб страницы, или CSS, формирующий внешнее представление веб­ страницы.

Программы, написанные на языке JavaScript, называются скриптами. В браузере они подключаются непосредственно к HTML-документу и, как только загружается страница – тут же выполняются.

JavaScript клиентский язык и работает исключительно на стороне клиента.

На сегодняшний день каждый браузер имеет поддержку JavaScript, поэтому этот язык можно считать одним из основных языков веба.

JavaScript в браузере дает возможность производить различные манипуляции с HTML документом, при этом не перезагружая страницу. С помощью JavaScript можно проверять корректность заполнения форм на сайте и в случае ошибки выдавать сообщения, отображать всплывающие окна, обрабатывать клики мыши, отправлять запросы на сервер и многое другое.

## Bootstrap

Bootstrap — это бесплатный фреймворк с открытым исходным кодом для создания веб-сайтов и веб-приложений. Фреймворки создают для того, чтобы другим веб-разработчикам было легче верстать сайты. Bootstrap является самым популярным фреймворком HTML, CSS и JS для разработки адаптивных и мобильных проектов в Интернете.

Интернет­ технологии все больше развиваются в сторону адаптивных сайтов, верстка становится гораздо сложнее, но Bootstrap помогает все упростить: больше не нужно «подгонять» верстку под разные типы устройств­ можно воспользоваться уже готовыми адаптивными шаблонами страниц и немного их видоизменить под свои задачи.

Bootstrap состоит из множества компонентов, и, чтобы задействовать любой из них, будь то выпадающее меню или оповещение, или что-то другое, то достаточно просто вызвать соответствующий класс.

У bootstrap я нашел несколько плюсов: во­ первых, удобство и быстрота разработки сайта, при этом сохраняя качество и согласованность; во­ вторых, он поставляется со своим набором стилей, которые легко переопределить или добавить свои стили.

## PHP и Rest API

PHP – язык программирования, который исполняется на стороне сервера, в то время как JavaScript исполняется в браузере на стороне пользователя.

В Битрикс24 есть возможность создавать свои приложения, используя связку PHP и открытого REST API.

REST API — это прикладной программный интерфейс (API), который использует HTTP-запросы для получения, извлечения, размещения и удаления данных. Аббревиатура REST в контексте API расшифровывается как «передача состояния представления» (Representational State Transfer). Технология позволяет получать и модифицировать данные и состояния удаленных приложений, передавая HTTP-запросы через интернет или любую другую сеть. REST API позволяет серверному приложению давать доступ к своим данным клиентскому приложению по определенному URL адресу.

REST — это архитектура для организации взаимодействия между независимыми объектами посредством протокола HTTP.

API — интерфейс взаимодействия с каким-либо объектом (программой, приложением), включающий в себя набор правил, которые позволяют одному приложению общаться с другим.

REST API имеет четыре базовых операции:

* Получение данных от сервера;
* Создание новых данных;
* Обновление существующих данных;
* Удаление данных.

Для выполнения этих операций через HTTP протокол используются следующие методы:

* GET — используется для получения со стороны севера определенного ресурса. Если вы производите этот запрос, сервер ищет информацию и отправляет ее вам назад;
* POST — для создания определенного ресурса на сервере. Сервер создает в базе данных новую сущность и оповещает, был ли процесс создания успешным;
* PUT и PATCH — используются для обновления определенной информации на сервере. В таком случае сервер просто изменяет информацию существующих сущностей в базе данных и оповещает об успехе;
* DELETE — удаляет указанную сущность.

Битрикс24 имеет набор своих методов для выполнения запросов к определенной сущности CRM, а методы в свою очередь посылают запрос к порталу Битрикс24 используя GET или POST запрос.

# Проектирование системы

При постановке задачи было решено разработать базу данных для системы работы с выпускниками. Для начала нужно определиться, какие сущности будут в нашей базе данных, поля и типы полей.

Так как система ориентирована на анализ дальнейшей судьбы выпускников, то основной сущностью будет сам выпускник. Таблица о выпускнике должна иметь такие поля, как «Фамилия», «Имя», «Отчество». Во время обучения выпускник мог иметь другую фамилию, например, до вступления в брак, поэтому нужно поле «Старая фамилия». Также у выпускника скорее всего есть почтовый ящик, мобильный телефон и страница в соц. сетях, добавим соответственно «E-mail», «Телефон», «Мессенджер».

Определим возможные типы полей (таблица 1).

Таблица 1 — Таблица выпускник

|  |  |  |  |
| --- | --- | --- | --- |
| **Выпускник** | | | |
| **Id\_выпускника** | Int | primary key |  |
| Фамилия | Char | not null |  |
| Имя | Char | not null |  |
| Отчество | Char |  |  |
| Новая фамилия | Char | default null | В случае смены фамилии |
| Телефон | Char |  |  |
| E-mail | Char |  |  |
| Мессенджер | Char |  |  |
| Дата\_обновления | Date | default null | Дата последнего обновления |

Система должна учитывать образование выпускника (сущность «Периоды обучения»), для этого нам потребуется знать, какое направление окончил выпускник (поле «Направление»), год его поступления, год, когда окончил обучение («год окончания»), его средняя оценка и его балл (поля «Оценка» и «Оценка в баллах»).Типы данных для этих полей представлены в таблице 2.

Таблица 2 — Периоды обучения

|  |  |  |  |
| --- | --- | --- | --- |
| **Периоды обучения** | | | |
| **Id\_выпускника** | Int | not null references **Выпускник** |  |
| **Id\_направления** | Int | not null references **Направления** |  |
| Телефон | Char | default null | Телефон во время обучения |
| Год\_поступления | Int | not null |  |
| Год\_окончания | Int | not null |  |
| Общая оценка | Char | default null |  |
| Оценка в баллах | Int | not null check (Оценка в баллах in (1, 2, 3, 4, 5)) |  |
|  | primary key (Id\_выпускника, Id\_направления) | | |

Атрибут сущности «Id\_направление» свяжем со справочником направлений (таблица 3).

Таблица 3 — Направления

|  |  |  |  |
| --- | --- | --- | --- |
| **Направления** | | | |
| **Id\_направления** | Int (Serial) | primary key | Справочник |
| Код\_направления | Char | not null |  |
| Название | Char | not null |  |
| Маг/Бак | Int | not null check (Маг/Бак in (0, 1)) | Признак |

В процессе дальнейшей карьеры выпускник, столкнувшись с реальными задачами, мог выделить из перечня дисциплин дисциплины, изучение которых помогло ему в работе. Добавим сущность «Нужные дисциплины» (таблица 4).

Таблица 4 — Нужные дисциплины

|  |  |  |  |
| --- | --- | --- | --- |
| **Нужные дисциплины/разделы** | | | |
| **Id\_выпускника** | Int | not null references **Выпускник** |  |
| **Id\_дисциплины** | Int | not null references **Дисциплины/разделы** |  |
|  | primary key (Id\_выпускника, Id\_ дисциплины) | | |

Для атрибута «Id\_дисциплины» добавим связь со справочником дисциплин (таблица 5).

Таблица 5 — Справочник дисциплин

|  |  |  |  |
| --- | --- | --- | --- |
| **Дисциплины/разделы** | | | |
| **Id\_ дисциплины** | Int (Serial) | primary key | Справочник |
| Название | Char | not null |  |

У выпускника в процессе его обучения в вузе могло сложиться впечатление о преподавателях разных кафедр, поэтому добавим сущности «Позитивное воспоминание» (таблица 6) и «Негативное воспоминание» (таблица 7).

Таблица 6 — Позитивное воспоминание

|  |  |  |  |
| --- | --- | --- | --- |
| **Позитивные воспоминания** | | | |
| **Id\_выпускника** | Int | not null references **Выпускник** |  |
| **Id\_преподавателя** | Int | not null references **Преподаватели** |  |
|  | primary key (Id\_выпускника, Id\_ преподавателя) | | |

Таблица 7 — Негативные воспоминание

|  |  |  |  |
| --- | --- | --- | --- |
| **Негативные воспоминания** | | | |
| **Id\_выпускника** | Int | not null references **Выпускник** |  |
| **Id\_преподавателя** | Int | not null references **Преподаватели** |  |
|  | primary key (Id\_выпускника, Id\_ преподавателя) | | |

Воспоминания выпускника связаны с определенными преподавателями, поэтому потребуется сущность «Преподаватели» (таблица 8).

Таблица 8 — Преподаватели

|  |  |  |  |
| --- | --- | --- | --- |
| **Преподаватели** | | | |
| **Id\_преподавателя** | Int (Serial) | primary key | Справочник |
| Фамилия | Char | not null |  |
| Имя | Char | not null |  |
| Отчество | Char | not null |  |
| Id\_кафедры | Int | not null references **Кафедры** |  |
| Комментарий | Char | default null | На пенсии, не работает и пр. |

Атрибут «Id\_кафедры» свяжем с сущностью-справочником «Кафедры», где может работать этот преподаватель (таблица 9).

Таблица 9 — Кафедры

|  |  |  |  |
| --- | --- | --- | --- |
| **Кафедры** | | | |
| **Id\_ кафедры** | Int (Serial) | primary key | Справочник |
| Название | Char | not null |  |

Система должна учитывать карьерные продвижения выпускника вуза: должность, которую он занимает, время работы на этой должности, организацию или компанию, где работал или работает в данный момент выпускник. Чтобы учесть всю необходимую информацию о работе выпускника, добавим сущность «Периоды работы» (таблица 10).

Таблица 10 — Периоды работы

|  |  |  |  |
| --- | --- | --- | --- |
| **Периоды работы** | | | |
| **Id\_выпускника** | Int | not null references **Выпускник** |  |
| **Id\_организации** | Int | not null references **Организации** |  |
| Год\_начала | Int | not null |  |
| Год\_окончания | Int | default null |  |
| Id\_должности | Int | not null references **Должности** |  |
|  | primary key (Id\_выпускника, Id\_организации) | | |

Атрибут «Id\_организации» свяжем с сущностью «Организация», где будет отражена информация о компании, в которой работает выпускник (таблица 11), а атрибут «Id\_должности» свяжем со справочником «Должности» (таблица 12).

Таблица 11 — Организация

|  |  |  |  |
| --- | --- | --- | --- |
| **Организации** | | | |
| **Id\_ Организации** | Int (Serial) | primary key | Справочник |
| Название | Char | not null |  |
| Адрес | Char | default null |  |
| Адрес Web-сайта | Char | default null |  |

Таблица 12 — Должности

|  |  |  |  |
| --- | --- | --- | --- |
| **Должности** | | | |
| **Id\_ Должности** | Int (Serial) | primary key | Справочник |
| Название | Char | not null |  |

Выпускник во время учебы в университете либо на обучении в компании, где работает, мог изучить некоторые технологии, и, чтобы их учитывать добавим сущность «Владение технологиями» (таблица 13).

Таблица 13 — Владение технологиями

|  |  |  |  |
| --- | --- | --- | --- |
| **Владение технологиями** | | | |
| **Id\_выпускника** | Int | not null references **Выпускник** |  |
| **Id\_технологии** | Int | not null references **Технологии** |  |
| Id\_организации | Int | default null references **Места обучения** |  |
|  | primary key (Id\_выпускника, Id\_ технологии) | | |

Атрибут «Id\_технологии» свяжем с сущностью справочником «Технологии», где будут технологии, которыми владеет выпускник (таблица 14).

Таблица 14 — Технологии

|  |  |  |  |
| --- | --- | --- | --- |
| **Технологии** | | | |
| **Id\_ технологии** | Int (Serial) | primary key | Справочник |
| Название | Char | not null |  |

Атрибут «Id\_организации» свяжем с сущностью «Места обучения», где будут отражены места, в которых выпускник прошел обучение (таблица 15).

Таблица 15 — Места обучения

|  |  |  |  |
| --- | --- | --- | --- |
| **Места обучения** | | | |
| **Id\_ организации** | Int (Serial) | primary key | Справочник |
| Название | Char | not null |  |

Выпускник владеет определенными компетенциями, чтобы их учитывать добавим сущность «Владение компетенциями» (таблица 16).

Таблица 16 — Владение компетенциями

|  |  |  |  |
| --- | --- | --- | --- |
| **Владение компетенциями** | | | |
| **Id\_выпускника** | Int | not null references **Выпускник** |  |
| **Id\_компетенции** | Int | not null references **Компетенции** |  |
|  | primary key (Id\_выпускника, Id\_ компетенции) | | |

Для атрибута «Id\_компетенции» сделаем связь со справочником «Компетенции», где будут перечислены все возможные компетенции, которыми может владеть выпускник (таблица 17).

Таблица 17 — Компетенции

|  |  |  |  |
| --- | --- | --- | --- |
| **Компетенции** | | | |
| **Id\_ компетенции** | Int (Serial) | primary key | Справочник |
| Название | Char | not null |  |

Определившись с нужными сущностями для системы учета выпускников, и с связями между ними, построим ER диаграмму (рисунок 2).

Проанализировав диаграмму, стало ясно, что удобнее всего реализовать базу данных в CRM системе Битрикс24. Система учета выпускников вуза идеально подходит в концепцию CRM системы, где в центре находится клиент, в нашем случае будет выпускник.

Изучив все возможности битрикс24, был сформирован основной план задач:

1. Работающую систему - CRM про выпускников, с настройками и демо-данными. В том числе с настройками полей
2. Систему для автоматизированного импорта справочников и других служебных данных для развертывания в новом Битрикс24
3. Сайт для ассоциации выпускников, содержащий: страницу об ассоциации, контакты ассоциации, лучшие выпускники, форма добавления информации о выпускнике.
4. Бизнес-процесс для обработки заявки с сайта.



Рисунок 2 — База данных выпускников

## Разработка CRM

В системе анализа выпускников ключевым будет выпускник, а в CRM ключевая сущность «Контакт», поэтому будем использовать контакты для реализации таблицы «Выпускник». При разработке контактов скроем ненужные нам поля карточки контакта, заданные по умолчанию, и добавим недостающие поля. Для таблицы «Преподаватели» также будем использовать сущность «Контакт». Для того, чтобы «Выпускники» и «Преподаватели» не путались — добавим поле-список: студент/преподаватель, и настроим фильтр в общем списке контактов.

На рисунке 3 представлены поля карточки контакта:

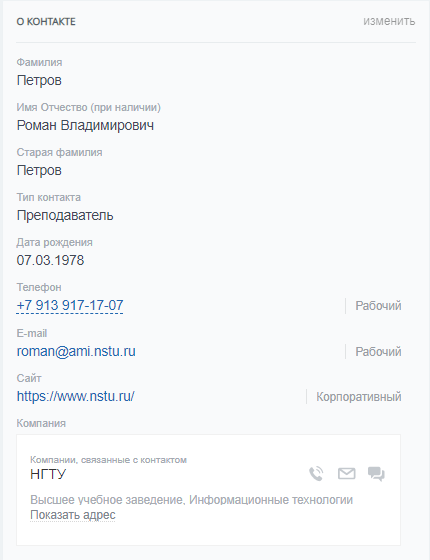


Рисунок 3 – Поля карточки контакта.

Наполним базу контактов CRM выпускниками (рисунок 4):

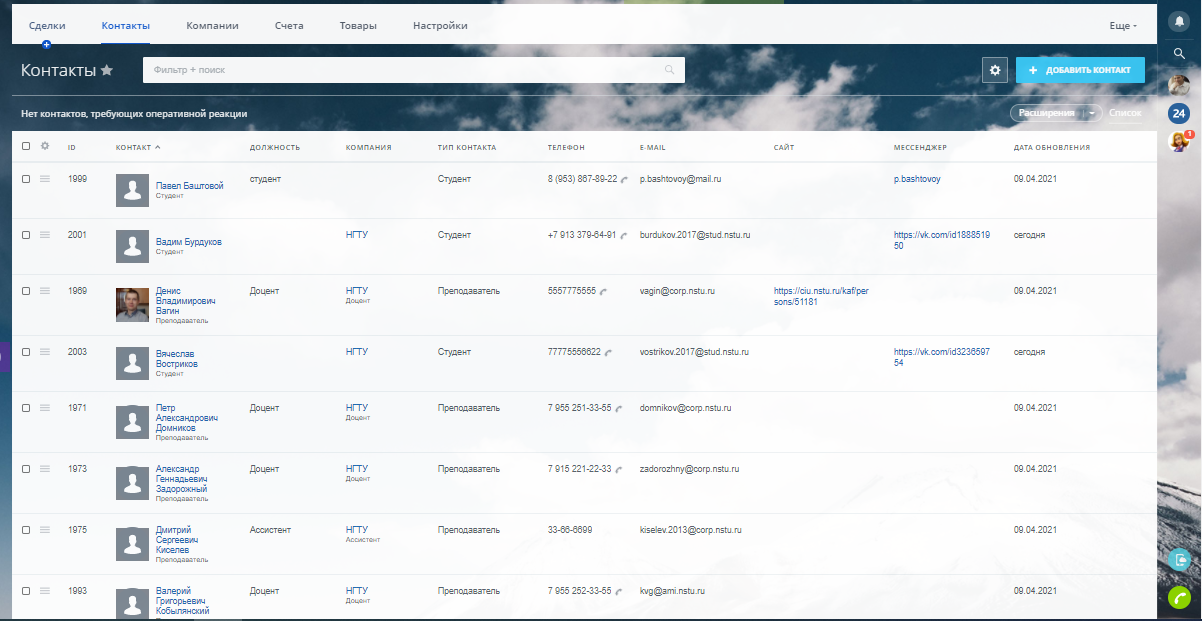
**

Рисунок 4 — Контакты

Для таблицы «Организации» будем использовать элемент CRM «Компании». Базовых полей, которые включают в себя название компании, тип компании, сферу деятельности, веб сайт достаточно, создавать пользовательские поля для компаний не потребуется. Более того, возможности CRM ­ системы позволяют получать полный физический адрес компании по номеру ИНН, не вводя все вручную.

На рисунке 5 представлены данные о компании:

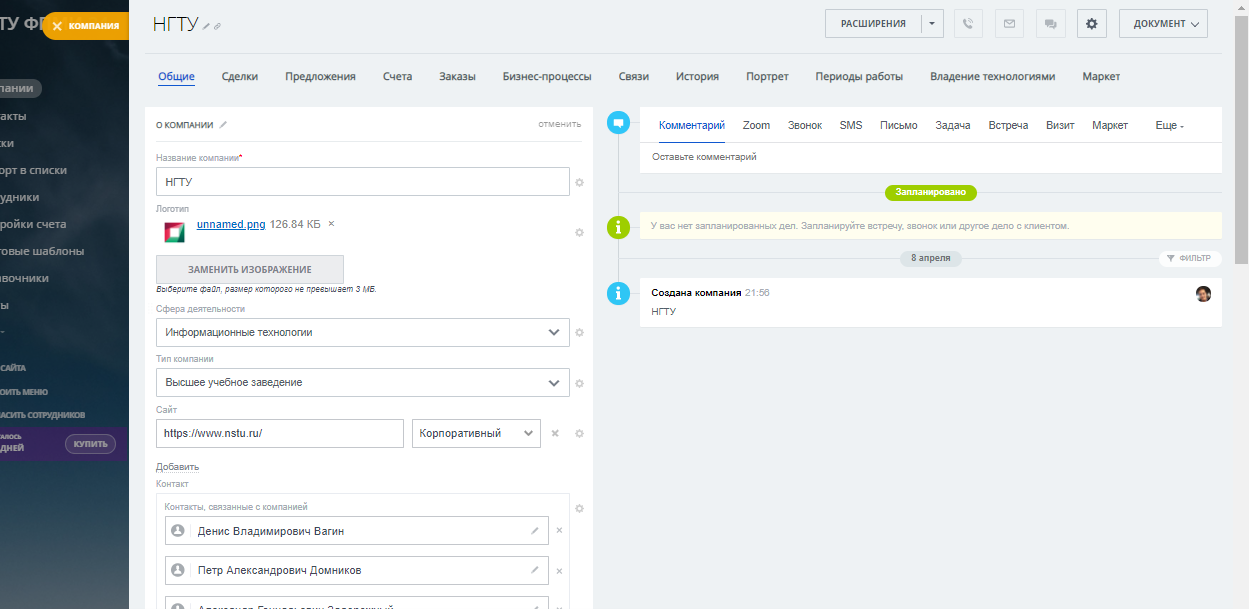
**

Рисунок 5 — Данные компании

К каждой компании привяжем сотрудников(контакты). На рисунке 6 представлен пример списка контактов, относящихся к конкретной компании:

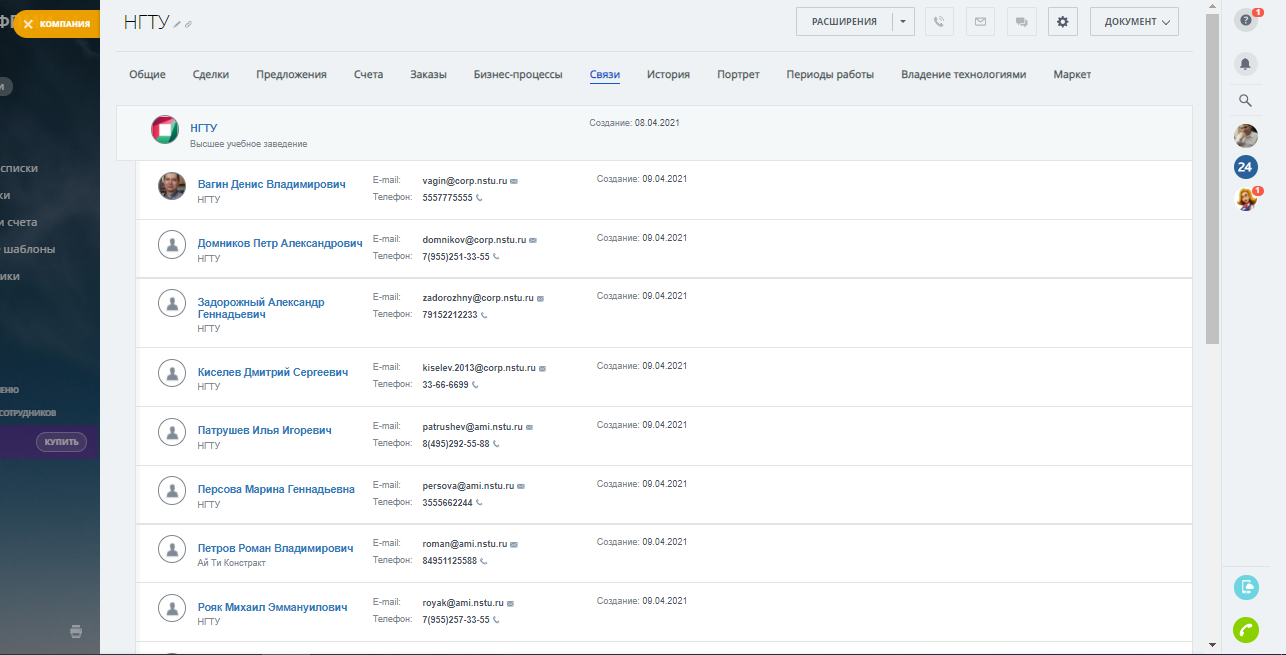


Рисунок 6 — Сотрудники компании

Для отслеживания карьеры выпускников будем использовать возможности верхнего тарифа Битрикс24 «универсальные списки».

Универсальный список — инструмент для хранения и структуризации большого объема информации, состоящий из настроенных пользовательских полей. С помощью данного инструмента создадим таблицы — справочники для системы анализа выпускников. Для каждой таблицы добавим и настроим собственные поля и сделаем привязку списка к списку, либо списка к элементам CRM.

Настроим таблицу «Компетенции» (рисунок 7).

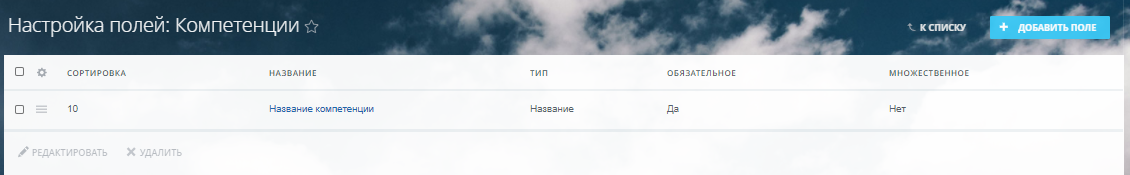


Рисунок 7 — Компетенции

Настроим таблицу «Дисциплины» (рисунок 8):

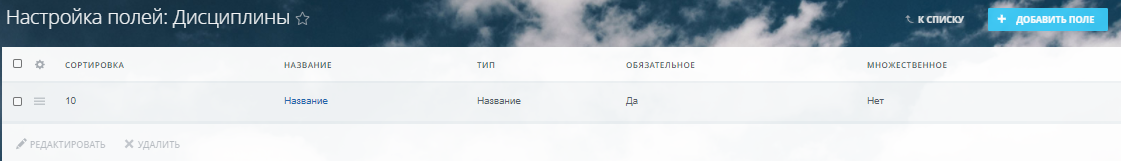


Рисунок 8 — Дисциплины

Настроим таблицу «Кафедры» (рисунок 9):

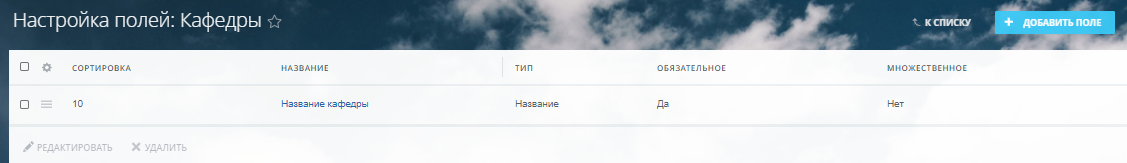


Рисунок 9 — Кафедры

Для таблиц «Должности» и «Технологии» выполним аналогичные настройки.

Теперь настроим таблицу «Направления». Для выбора академической степени направления создадим поле «список», где будут представлены вариации, например магистр или бакалавр (рисунок 10).

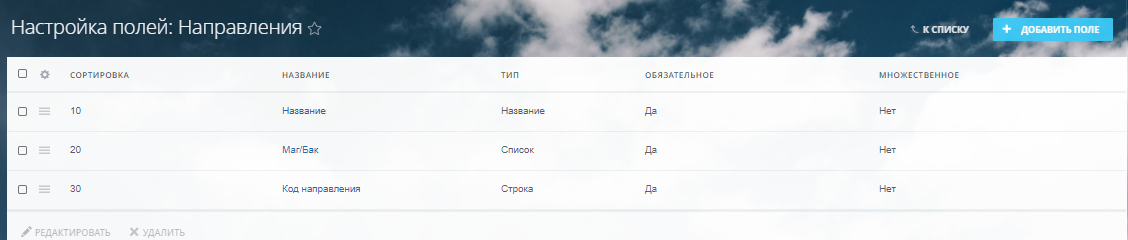


Рисунок 10 — Направления

Затем настроим таблицы, где будет привязка к спискам — справочникам или к элементам CRM (компании, контакты).

Например, для таблицы «Периоды работы» нам потребуются такие поля, как выпускник (контакты CRM), Организация, где работал или работает в данный момент выпускник (компании CRM), периоды его работы (число), занимаемая должность (привязка к элементу списка «Должности»). Теперь выполним настройки таблицы (рисунок 11).

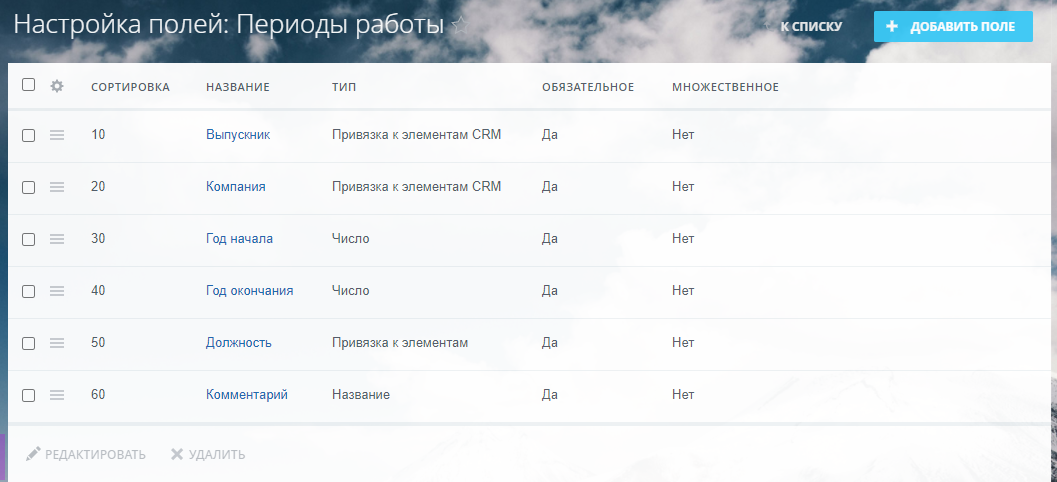


Рисунок 11 — Периоды работы

Настроим таблицу «Периоды обучения»: для полноценной информации о учебных заведениях и успеваемости студента создадим такие поля, как выпускник (контакт CRM), направления (привязка к списку «Направления»), периоды обучения, оценка и балл (число) студента (рисунок 12).

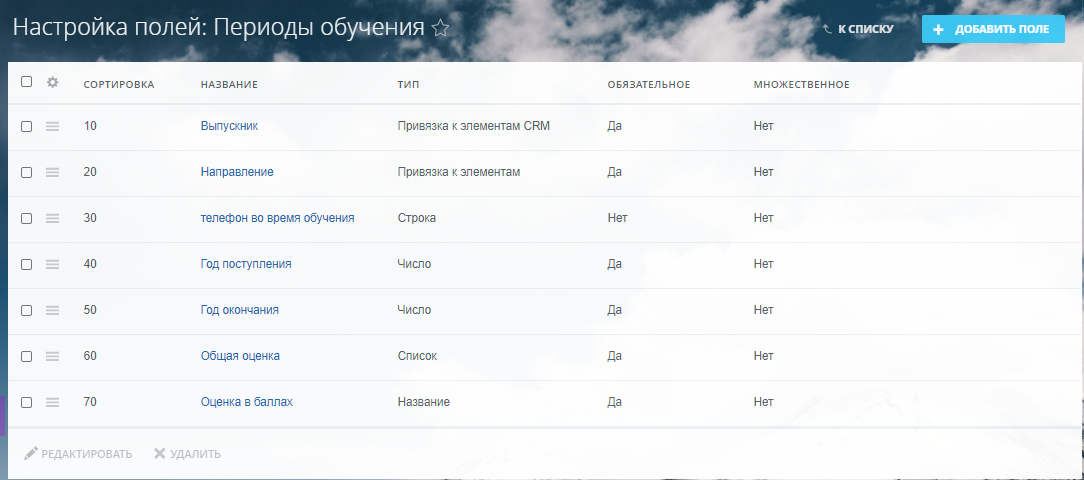


Рисунок [12] — Периоды обучения

Выполним настройки для списка «Нужные дисциплины» (рисунок 13):

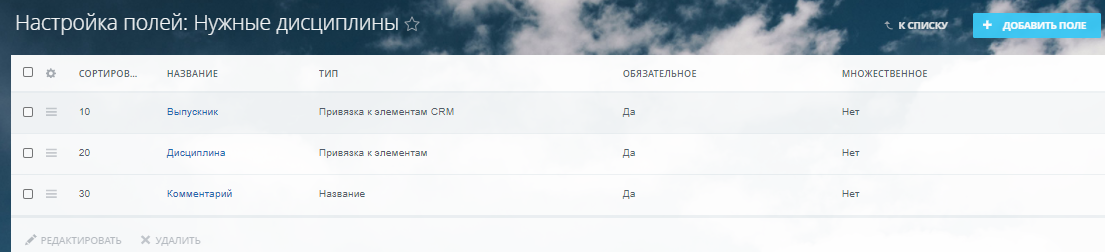


Рисунок 13 — Нужные дисциплины.

Также необходимо знать, какими технологиями владеет исследуемый выпускник (рисунок 14).

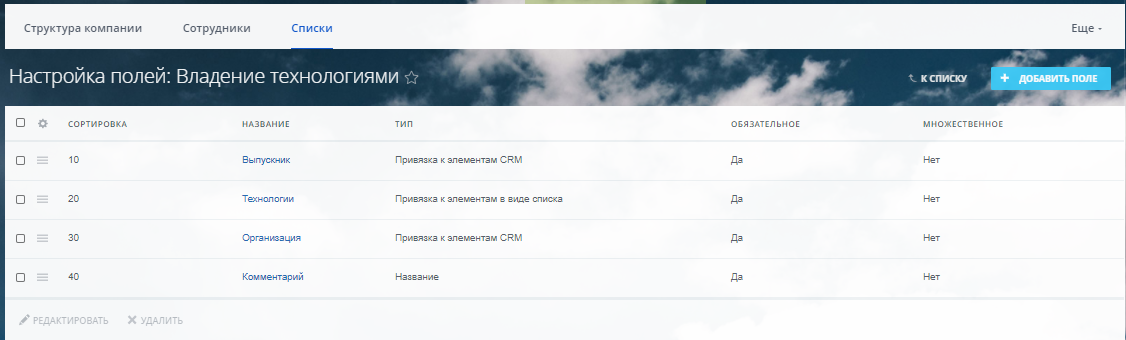


Рисунок 14 — Владение технологиями

Настроим список «Владение компетенциями» (рисунок 15):

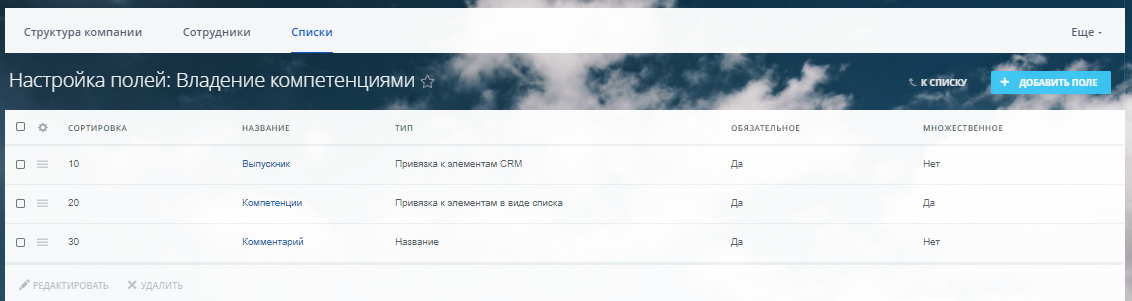


Рисунок 15 — Владение компетенциями

Все необходимые для анализа таблицы созданы, заполним данными списки справочники, например, «Должности» (рисунок 16). Для таблиц «Технологии», «Кафедры», «Дисциплины», «Компетенции» заполним аналогично.

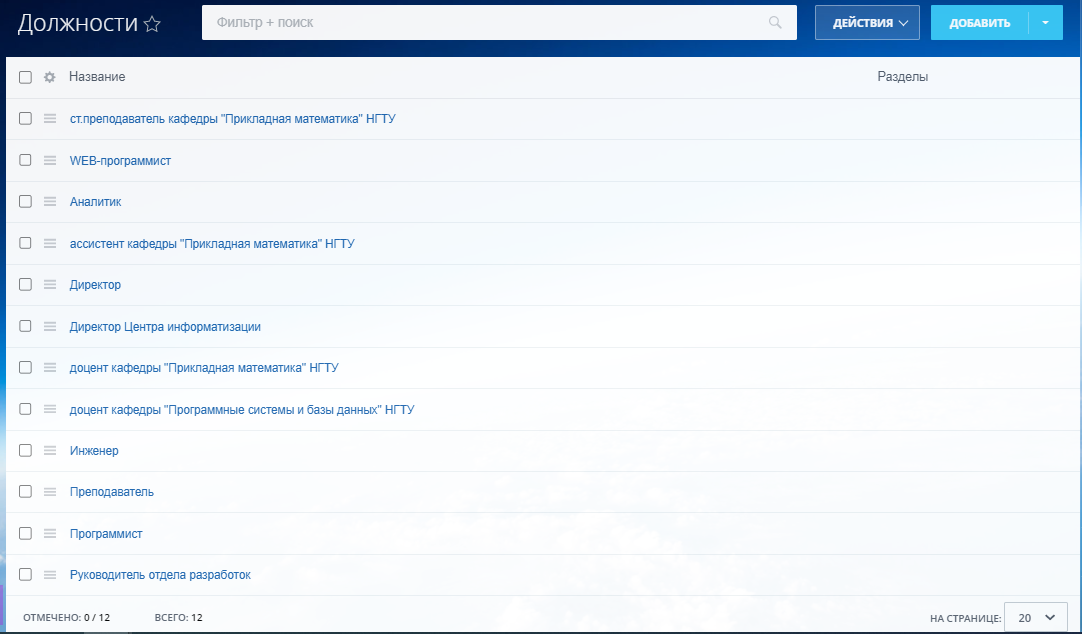


Рисунок 16 — пример заполнения справочника

После завершения разработки таблиц, привязки их к другим таблицам и к элементам CRM, в карточках контактов появились соответствующие вкладки, а также вся необходимая информация о выпускнике. Например, периоды его работы (рисунок 17), периоды обучения (рисунок 18), воспоминания(рисунок 19), компетенции, которыми владеет выпускник (рисунок 20) и технологии, с которыми он знаком (рисунок 21):

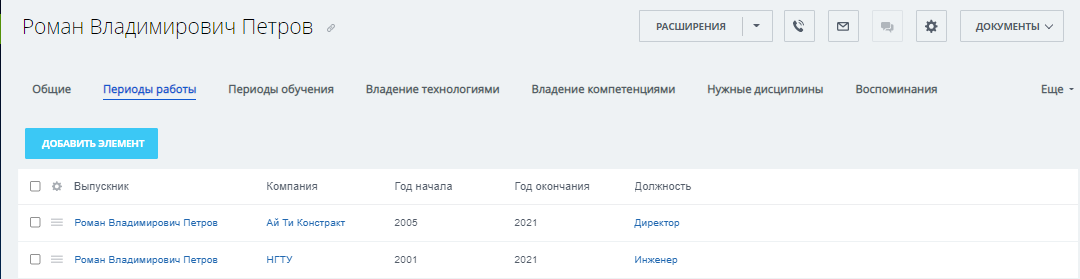


Рисунок 17 — Периоды работы

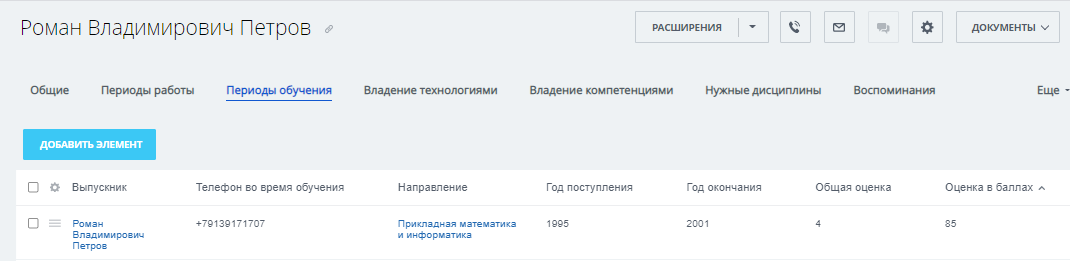


Рисунок 18 — Периоды обучения

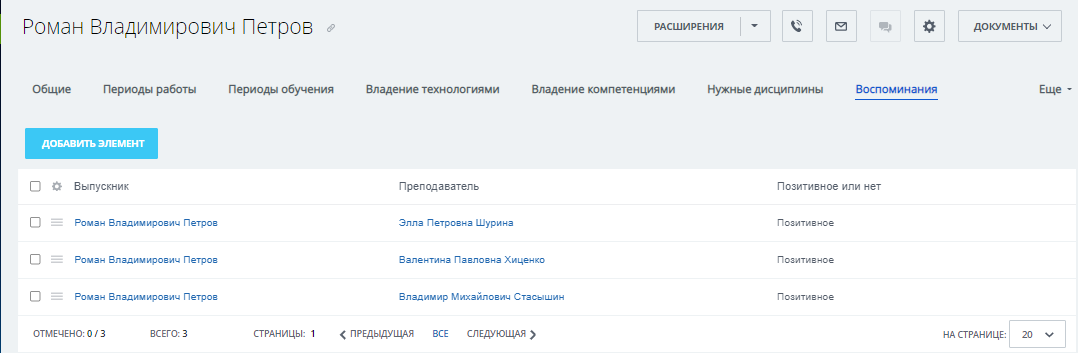


Рисунок 19 — Воспоминания

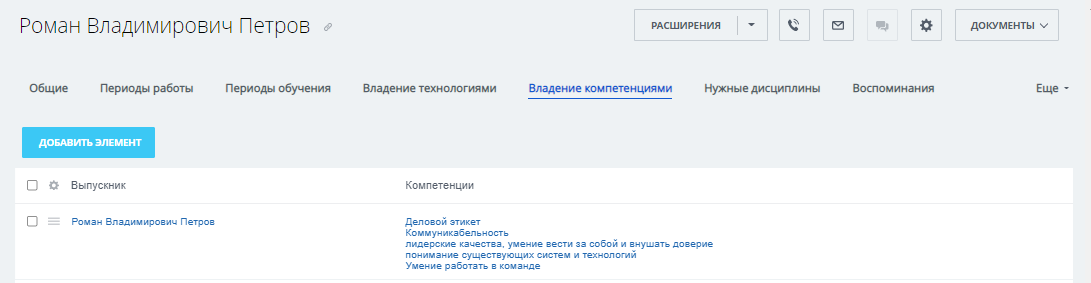


Рисунок 20 — Владение компетенциями



Рисунок 21 — Владение технологиями

## Разработка приложения импорта

Так как в дальнейшем нам потребуется устанавливать систему анализа выпускников на новые порталы Битрикс24, то стоит продумать удобный импорт всей базы данных, при этом ручную настройку системы сведя к минимуму.

Рассмотрим базовые возможности Битрикс24: система позволяет в несколько кликов экспортировать как элементы CRM (контакты и компании), так и универсальные списки, на основе которых сделаны таблицы базы данных. При экспорте все данные сохраняются в файлы excel либо в csv формат. С импортом на новый портал дела обстоят иначе: сущности CRM легко импортируются, а для универсальных списков он отсутствует, поэтому было решено, используя технологию Rest API, создать приложение импорта для системы анализа выпускников.

Так как нам потребуется организация импорта данных, а не настройки в рамках текущего портала, то было решено отказаться от локального приложения внутри портала Битрикс24 и использовать входящие вебхуки, а приложение разместить на отдельном сервере.

В качестве сервера для расположения файлов программы мной был выбран хостинг timeweb. Данный хостинг недорогой и очень удобный в использовании.

При работе с приложением воспользуемся следующими технологиями:

* HTML и CSS для вёрстки модуля;
* Bootstrap для шаблона страницы запуска;
* JavaScript для создания интерактивной страницы;
* PHP для написания логики модуля
* Rest API для взаимодействия с порталом Битрикс24.

Для создания универсальных списков будем использовать метод lists.add(param). Метод имеет следующие входные параметры (таблица 18):

Таблица 18 — Параметры добавления списка.

|  |  |
| --- | --- |
| Параметр | Описание |
| IBLOCK\_TYPE\_ID | id типа инфоблока (обязательное):   * **lists** — тип инфоблока   списка   * **bitrix\_processes** — тип инфоблока процессов * **lists\_socnet** — тип инфоблока списков групп |
| IBLOCK\_CODE | Код инфоблока (обязательное); |
| SOCNET\_GROUP\_ID | id группы; |
| FIELDS | Поля инфоблока:   * **NAME —** название инфоблока (обязательно); * **DESCRIPTION —** описание; * **SORT—** сортировка; * **PICTURE—** изображение; * **BIZPROC—** включение поддержки бизнес-процессов. |
| RIGHTS | Управление правами доступа (Пользователю, работающему с Rest, выдается полный доступ по умолчанию). |

Пример функции для создания нового списка представлен ниже:

function lists\_add($Webhook\_URL,$list\_name,$block\_code)

{

$queryUrl = "$Webhook\_URL/lists.add.json";

$queryData = http\_build\_query(array(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"$block\_code",

'fields' => array(

'NAME' =>"$list\_name",

'DESCRIPTION' => 'Здесь будут данные',

'SORT'=> '500',

'BIZPROC'=> 'Y')));

executeHook($queryUrl,$queryData);

}

Где на вход подается URL адрес входящего вебхука Битрикс24, название списка и код инфоблока. Из этих параметров формируется запрос и передается в функцию executeHook($queryUrl,$queryData) для отправки запроса серверу:

function executeHook($queryUrl,$queryData)

{

$curl = curl\_init();

curl\_setopt\_array($curl, array(

CURLOPT\_SSL\_VERIFYPEER => 0,

CURLOPT\_POST => 1,

CURLOPT\_HEADER => 0,

CURLOPT\_RETURNTRANSFER => 1,

CURLOPT\_URL => $queryUrl,

CURLOPT\_POSTFIELDS => $queryData,

));

$result = curl\_exec($curl);

curl\_close($curl);

$result = json\_decode($result, true);

}

Для добавления и настройки полей списков будем использовать метод lists.field.add (params).

Метод включает в себя достаточно большое количество параметров, поэтому в таблице 19 представлены только те, которые были использованы при разработке модуля:

Таблица 19 — Добавление и настройка полей

|  |  |
| --- | --- |
| Параметр | Описание |
| IBLOCK\_TYPE\_ID | id типа инфоблока, в нашем случае **lists** (список) |
| IBLOCK\_CODE | код инфоблока; |
| FIELDS | * **NAME—**название (обязательно) * **IS\_REQUIRED—**метка обязательности * **MULTIPLE—**метка множественности * **TYPE—**тип поля:   + **S** - Строка   + **N** - Число   + **L** - Список   + **E** - Привязка к элементам   + **E:EList** - Привязка к элементам в виде списка   + **S:ECrm** - Привязка к элементам CRM * **SORT—**сортировка * **LIST—** для добавления значений полю «список» * **CODE—**код добавляемого поля * **SETTINGS —** обязательные настройки поля:   + SHOW\_ADD\_FORM   + SHOW\_EDIT\_FORM   + ADD\_READ\_ONLY\_FIELD   + EDIT\_READ\_ONLY\_FIELD   + SHOW\_FIELD\_PREVIEW * **LINK\_IBLOCK\_ID—** Если поле списка имеет привязку к другому списку, то указывается ID привязываемого списка |

Пример запроса для добавления поля с привязкой к другому списку:

$queryData = http\_build\_query(array(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"$block\_code",

'fields' => array(

'NAME' => "$field\_name",

'IS\_REQUIRED' => 'N',

'MULTIPLE' => 'N',

'TYPE'=>"E",

'LINK\_IBLOCK\_ID'=>"$link\_id",

'CODE'=>"Kompetencii",

'SETTINGS'=>array(

'SHOW\_ADD\_FORM'=>'Y',

'SHOW\_EDIT\_FORM'=>'Y',)

)

Для привязки списка в запросе требуется указывать ID привязываемого списка, но т.к. ID универсальных списков не задается вручную, а присваивается системой Битрикс24 автоматически, то было решено написать функцию для получения ID по коду нужного блока с помощью метода lists.get, который возвращает полную информацию о списке. Отправим следующий запрос:

$queryData = http\_build\_query(array(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"$block\_code"

));

В ответ сервер вернет массив данных, где в элементе массива $result['result'][0]['ID'] содержится ID списка.

После того как списки созданы и настроены поля, заполним данными списки — справочники («Должности», «Технологии», «Компетенции» и т.д). Для добавления элементов в списки будем использовать метод lists.element.add,

параметры которого представлены в таблице 20:

Таблица 20 — Добавление элементов в списки.

|  |  |
| --- | --- |
| Параметр | Описание |
| IBLOCK\_TYPE\_ID | id типа инфоблока:   * lists—тип инфоблока списка |
| IBLOCK\_CODE | код инфоблока; |
| LIST\_ELEMENT\_URL | шаблон адреса к элементам списка |
| FIELDS | Массив полей и их значений |
| SOCNET\_GROUP\_ID | id группы группы |

Часть кода на добавление элемента в список приведена ниже:

while (($row = fgetcsv($fh, 0, ';')) !== false)

{

$i=$i+1;

$batch['cmd\_'.$i] =

'lists.element.add?'.http\_build\_query( array(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"$block\_code",

'ELEMENT\_CODE'=>"$i",

'fields' => array('NAME' => "$row[0]") ) );

if ($i % 50 == 0)

{

$res.='>>>'.print\_r(executeHook($queryUrl,

http\_build\_query(array('cmd'=>$batch))), true);

$batch = array();

}

}

В Битрикс24 по умолчанию есть ограничение на 2 запроса в секунду и при превышении этого лимита после 50 запроса блокируются все последующие запросы, отправленные на сервер. Столкнувшись с этой проблемой, было решено объединять похожие запросы в пакеты при помощи метода batch, используя при этом параметр cmd для представления однотипных запросов в виде массива. В приведенном примере запросы выполняются пакетами.

Разработав логику приложения импорта с помощью PHP и Rest API, появилась необходимость создать удобный пользовательский интерфейс. В качестве шаблона страницы был взят шаблон Bootstrap. Сделаем поле ввода, куда пользователь будет добавлять URl адрес своего входящего вебхука и кнопку для запуска скрипта. Чтобы уведомлять пользователя о статусе выполнения приложения добавим полосу загрузки. Реализуем ее при помощи JavaScript.

На рисунке 22 представлен внешний вид приложения:

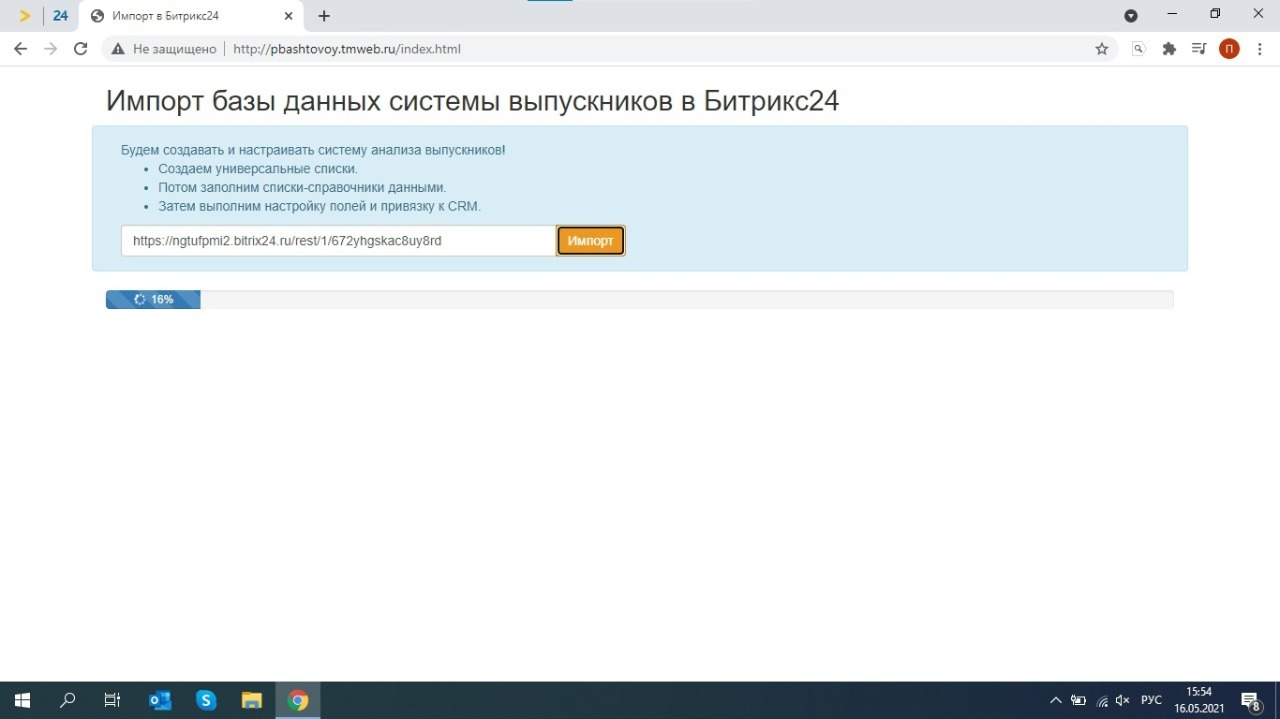


Рисунок 22 — Приложение импорта

# Создание публичного сайта

С помощью конструктора сайтов «Битиркс24.Сайты» создадим сайт ассоциации выпускников вуза. Создание сайта необходимо для того, чтобы:

* Преподаватели вузов могли отслеживать дальнейшие карьерные успехи выпускников.
* ВУЗы смогут привлекать с помощью сайта новых выпускников в ассоциацию.
* Выпускники смогли узнавать новости от ассоциации.
* Ассоциация выпускников могла собирать пожертвования.

Инструмент «Битрикс24.сайты» позволяет создавать страницы двух видов:

1. *Статичные*

Статические страницы — такие страницы, которые при каждом обращении пользователя к сайту представляются в неизменном виде.

Содержимое таких страниц может менять администратор сайта при помощи конструктора страницы.

1. *Динамические*

Содержимое динамических страниц непостоянно и может меняться в процессе работы с сайтом. Для динамического отображения страницы достаточно добавить в нужный блок группу или живую ленту, и вся информация из группы отобразится на странице.

## Страница «об ассоциации»

Страница «Об ассоциации» — это важный атрибут сайта (рисунок 23). Данная страница будет содержать:

1. Основную информацию об ассоциации;
2. Цели и задачи ассоциации;
3. Призыв вступить в ряды ассоциации выпускников.

Страница «Об ассоциации» будет статичной.

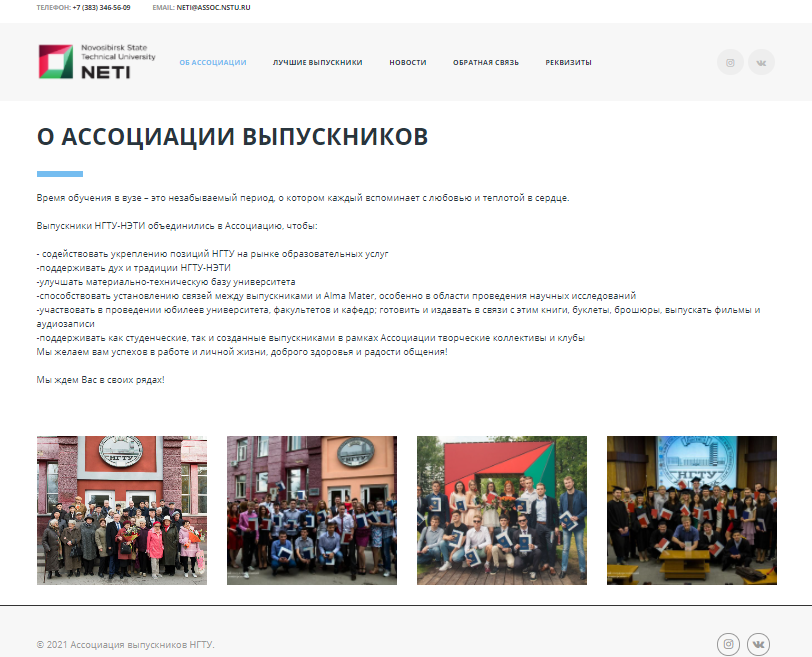


Рисунок 23 — Об ассоциации

## Страница «Лучшие выпускники»

Страница «Лучшие выпускники» (рисунок 24) содержит доску почета, где будет отображаться информация в виде карточек: портрет студента, ФИО, год окончания университета и его личные достижения. Страница сделана динамической, вся информация отображается из группы «лучшие студенты». Карточки можно легко редактировать либо добавлять новые, лишь изменяя либо добавляя новые сообщения в группу на портале Битрикс24.

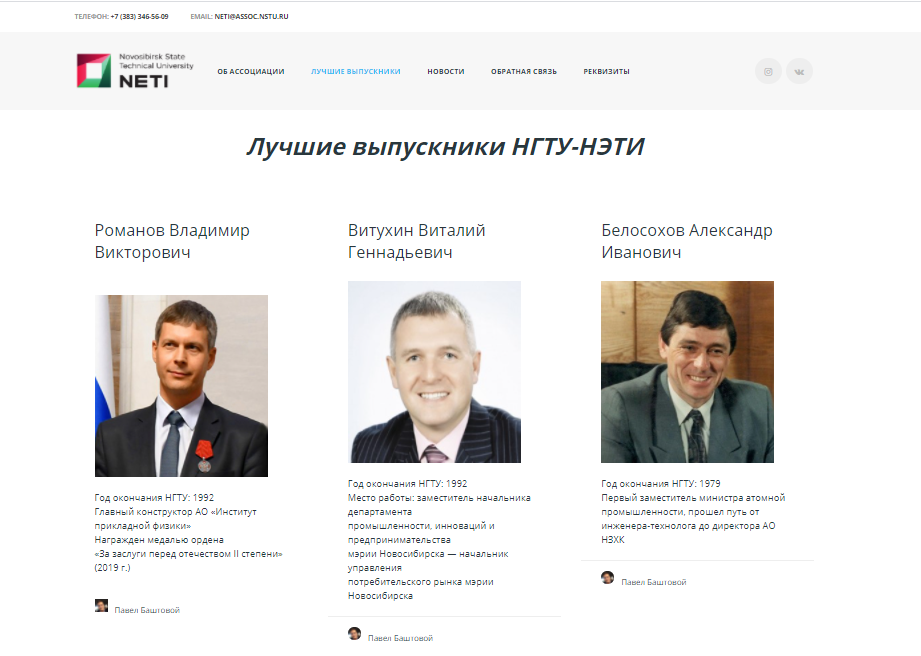


Рисунок 24 — Страница «Лучшие выпускники».

## Страница «Новости»

Страница «Новости» (рисунок 25) содержит список новостей, где есть фотография новости, заголовок, краткое описание новости, ссылка на детальную новость. Страница сделана динамической, для новостей была создана группа «Новостной блог», через которую можно добавлять и редактировать новости.

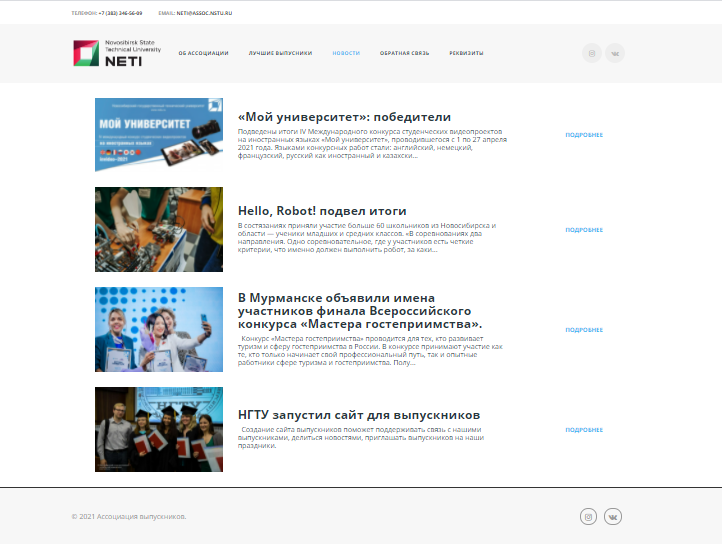


Рисунок 25 — страница «Новости»

## Страница «Обратная связь»

Страница для обратной связи (рисунок 26) содержит контактную информацию ассоциации (информацию об организации, расположение, телефон, email для связи), а также CRM форму, с помощью которой выпускник сможет добавить информацию о себе и отправить в нашу CRM.

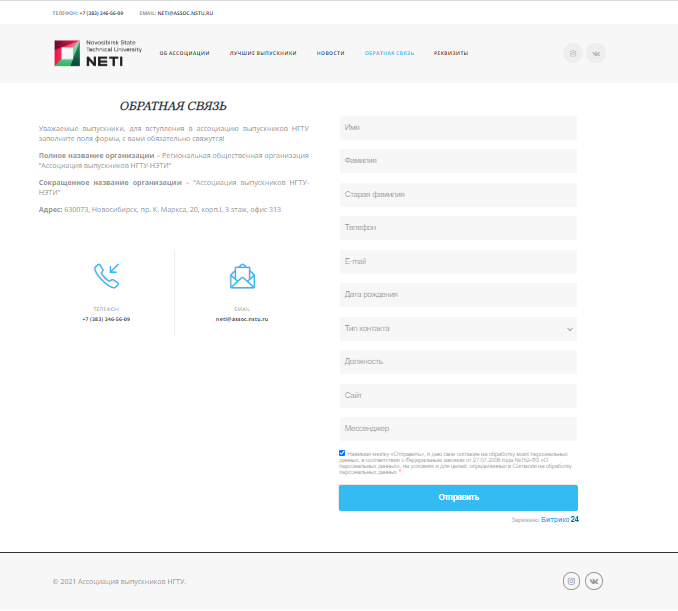


Рисунок 26 — страница «Обратная связь»

## Страница «Реквизиты»

Страница «Реквизиты» служит для призвания к пожертвованию в ассоциацию выпускников с указанием всех необходимых реквизитов для перевода в фонд ассоциации (рисунок 27).

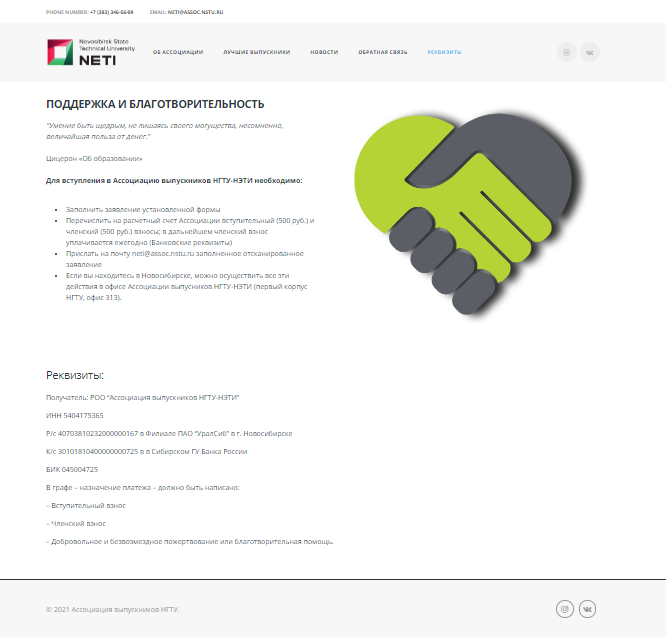


Рисунок 27 — Страница «Реквизиты».

# ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была получена информационная система работы с выпускниками, состоящая из базы данных, содержащей полноценную информацию о выпускнике, на основе CRM системы Битрикс24, приложения импорта данных для развертывания в другом Битрикс24, публичного сайта для отображения полезной информации, привлечения новых выпускников в ассоциацию и добавление новой информации о выпускниках.

При разработке системы были решены следующие задачи:

* Разработка базы данных системы выпускников;
* Разработка CRM системы;
* Разработка приложения импорта данных с использованием API;
* Создание сайта ассоциации выпускников;
* Настройка CRM формы обновления информации.

# СПИСОК ЛИТЕРАТУРЫ

1. Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/>.
2. Документация по REST API [Электронный ресурс] / 1С-Битрикс. – Режим доступа: <https://dev.1c-bitrix.ru/rest_help/index.php>.
3. Учебный курс по Bitrix Framework [Электронный ресурс] – Режим доступа:<http://dev.1c-bitrix.ru/learning/course/?COURSE_ID=43&INDEX=Y>.
4. Справочник по PHP [Электронный ресурс]. –Режим доступа:

<https://www.php.net/manual/ru/langref.php>

1. Справочник по HTML и CSS [Электронный ресурс]. – Режим доступа: <http://htmlbook.ru/>
2. Ресурс для IT-специалистов [Электронный ресурс]. – Режим доступа: <https://habr.com/>

# ПРИЛОЖЕНИЕ А.ТЕКСТ ПРОГРАММЫ СЕРВЕРА

<?php

**function** executeHook($queryUrl,$queryData)

{

// обращаемся к Битрикс24 при помощи функции curl\_exec

$curl = **curl\_init**();

**curl\_setopt\_array**($curl, **array**(

**CURLOPT\_SSL\_VERIFYPEER** => 0,

**CURLOPT\_POST** => 1,

**CURLOPT\_HEADER** => 0,

**CURLOPT\_RETURNTRANSFER** => 1,

**CURLOPT\_URL** => $queryUrl,

**CURLOPT\_POSTFIELDS** => $queryData,

));

$result = **curl\_exec**($curl);

**curl\_close**($curl);

$result = **json\_decode**($result, **true**);

**if** (**array\_key\_exists**('error', $result))**echo** "Ошибка при сохранении списка: ".$result['error\_description']."<br/>";

}

//добавляем список на портал

**function** lists\_add($Webhook\_URL,$list\_name,$block\_code)

{

// формируем URL в переменной $queryUrl

$queryUrl = "**$Webhook\_URL**/lists.add.json";

// формируем параметры для создания лида в переменной $queryData

$queryData = **http\_build\_query**(**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**",

'fields' => **array**(

'NAME' =>"**$list\_name**",

'DESCRIPTION' => 'Здесь будут данные',

'SORT'=> '500',

'BIZPROC'=> 'Y'

)

));

executeHook($queryUrl,$queryData);

}

//добавляем элементы списков-справочников пакетом запросов

**function** elements\_add($Webhook\_URL,$block\_code,$csv\_import\_file)

{

$queryUrl = "**$Webhook\_URL**/batch.json";

$batch = **array**();

$res = '';

$file =$csv\_import\_file;

//открываем файл с CSV-данными

$fh = **fopen**($file, "r");

// делаем пропуск первой строки, смещая указатель на одну строку

**fgetcsv**($fh, 0, ';');

$i=0;

//читаем построчно содержимое CSV-файла

**while** (($row = **fgetcsv**($fh, 0, ';')) !== **false**)

{

$i=$i+1;

//вывел на экран то что считал из файла

**echo** '<br>'.$i.''.$row[0].'</br>';

//заполнение пакета

$batch['cmd\_'.$i] =

'lists.element.add?'.**http\_build\_query**(

**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**",

'ELEMENT\_CODE'=>"**$i**",

'fields' => **array**(

'NAME' => "**$row**[0]"

)

)

);

**if** ($i % 50 == 0)

{

$res .= '>>> '.**print\_r**(executeHook($queryUrl,**http\_build\_query**(**array**('cmd' => $batch))), **true**);

$batch = **array**();

}

}

**if** (**count**($batch) > 0) $res .= '>>> '.**print\_r**(executeHook($queryUrl,**http\_build\_query**(**array**('cmd' => $batch))), **true**);

**return** 'Список заполнен:<br/><br/>'.$res;

}

//---------------------------------------------------------------

//создание полей в списках

//добавление поля-числа

**function** number\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code)

{

// формируем URL в переменной $queryUrl

$queryUrl = "**$Webhook\_URL**/lists.field.add.json";

// формируем параметры для создания

$queryData = **http\_build\_query**(**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**",

'fields' => **array**(

'NAME' => "**$field\_name**",

'IS\_REQUIRED' => 'Y',

'MULTIPLE' => 'N',

'TYPE'=>"N",

'CODE'=>"**$field\_code**",

'SETTINGS'=>**array**(

'SHOW\_ADD\_FORM'=>'Y',

'SHOW\_EDIT\_FORM'=>'Y',

'ADD\_READ\_ONLY\_FIELD'=>'N',

'EDIT\_READ\_ONLY\_FIELD'=>'N',

'SHOW\_FIELD\_PREVIEW'=>'N'

)

)

));

executeHook($queryUrl,$queryData);

}

//добавление поля-строки

**function** string\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code)

{

// формируем URL в переменной $queryUrl

$queryUrl = "**$Webhook\_URL**/lists.field.add.json";

// формируем параметры для создания

$queryData = **http\_build\_query**(**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**",

'fields' => **array**(

'NAME' => "**$field\_name**",

'IS\_REQUIRED' => 'Y',

'MULTIPLE' => 'N',

'TYPE'=>"S",

'CODE'=>"**$field\_code**",

'SETTINGS'=>**array**(

'SHOW\_ADD\_FORM'=>'Y',

'SHOW\_EDIT\_FORM'=>'Y',

'ADD\_READ\_ONLY\_FIELD'=>'N',

'EDIT\_READ\_ONLY\_FIELD'=>'N',

'SHOW\_FIELD\_PREVIEW'=>'N'

)

)

));

executeHook($queryUrl,$queryData);

}

//добавление поля-списка уровня образования

**function** level\_of\_education\_list\_field\_add($Webhook\_URL,$block\_code)

{

// формируем URL в переменной $queryUrl

$queryUrl = "**$Webhook\_URL**/lists.field.add.json";

// формируем параметры для создания

$queryData = **http\_build\_query**(**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**",

'fields' => **array**(

'NAME' => "Маг/бак",

'IS\_REQUIRED' => 'Y',

'MULTIPLE' => 'N',

'TYPE'=>"L",

'CODE'=>"Level\_education",

'LIST\_TEXT\_VALUES'=> "магистр\n бакалавр\n",

'SETTINGS'=>**array**(

'SHOW\_ADD\_FORM'=>'Y',

'SHOW\_EDIT\_FORM'=>'Y',

'ADD\_READ\_ONLY\_FIELD'=>'N',

'EDIT\_READ\_ONLY\_FIELD'=>'N',

'SHOW\_FIELD\_PREVIEW'=>'N'

)

)

));

executeHook($queryUrl,$queryData);

}

//добавление поля-отзыва воспоминания

**function** pozitive\_negative\_field\_add($Webhook\_URL,$block\_code)

{

// формируем URL в переменной $queryUrl

$queryUrl = "**$Webhook\_URL**/lists.field.add.json";

// формируем параметры для создания

$queryData = **http\_build\_query**(**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**",

'fields' => **array**(

'NAME' => "Позитивное или нет",

'IS\_REQUIRED' => 'Y',

'MULTIPLE' => 'N',

'TYPE'=>"L",

'CODE'=>"pozitiv\_negativ",

'LIST\_TEXT\_VALUES'=> "Позитивное\n Негативное\n",

'SETTINGS'=>**array**(

'SHOW\_ADD\_FORM'=>'Y',

'SHOW\_EDIT\_FORM'=>'Y',

'ADD\_READ\_ONLY\_FIELD'=>'N',

'EDIT\_READ\_ONLY\_FIELD'=>'N',

'SHOW\_FIELD\_PREVIEW'=>'N'

)

)

));

executeHook($queryUrl,$queryData);

}

//добавление поля-списка оценок

**function** rating\_list\_field\_add($Webhook\_URL,$block\_code)

{

// формируем URL в переменной $queryUrl

$queryUrl = "**$Webhook\_URL**/lists.field.add.json";

// формируем параметры для создания

$queryData = **http\_build\_query**(**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**",

'fields' => **array**(

'NAME' => "Общая оценка",

'IS\_REQUIRED' => 'Y',

'MULTIPLE' => 'N',

'TYPE'=>"L",

'CODE'=>"Raiting",

'LIST\_TEXT\_VALUES'=> "5\n 4\n 3\n 2\n",

'SETTINGS'=>**array**(

'SHOW\_ADD\_FORM'=>'Y',

'SHOW\_EDIT\_FORM'=>'Y',

'ADD\_READ\_ONLY\_FIELD'=>'N',

'EDIT\_READ\_ONLY\_FIELD'=>'N',

'SHOW\_FIELD\_PREVIEW'=>'N'

)

)

));

executeHook($queryUrl,$queryData);

}

//Добавление поля подключения к CRM контактам

**function** CRM\_contact\_link\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code)

{

// формируем URL в переменной $queryUrl

$queryUrl = "**$Webhook\_URL**/lists.field.add.json";

// формируем параметры для создания

$queryData = **http\_build\_query**(**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**",

'fields' => **array**(

'NAME' => "**$field\_name**",

'IS\_REQUIRED' => 'Y',

'MULTIPLE' => 'N',

'TYPE'=>"S:ECrm",

'USER\_TYPE\_SETTINGS'=>**array**(

"CONTACT"=>"Y",

"VISIBLE"=>"Y"),

'CODE'=>"**$field\_code**",

'SETTINGS'=>**array**(

'SHOW\_ADD\_FORM'=>'Y',

'SHOW\_EDIT\_FORM'=>'Y',

'ADD\_READ\_ONLY\_FIELD'=>'N',

'EDIT\_READ\_ONLY\_FIELD'=>'N',

'SHOW\_FIELD\_PREVIEW'=>'N'

)

)

));

executeHook($queryUrl,$queryData);

}

//Добавление поля подключения к CRM компании

**function** CRM\_сompanies\_link\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code)

{

// формируем URL в переменной $queryUrl

$queryUrl = "**$Webhook\_URL**/lists.field.add.json";

// формируем параметры для создания

$queryData = **http\_build\_query**(**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**",

'fields' => **array**(

'NAME' => "**$field\_name**",

'IS\_REQUIRED' => 'Y',

'MULTIPLE' => 'N',

'TYPE'=>"S:ECrm",

'USER\_TYPE\_SETTINGS'=>**array**(

"COMPANY"=>"Y",

"VISIBLE"=>"Y"),

'CODE'=>"**$field\_code**",

'SETTINGS'=>**array**(

'SHOW\_ADD\_FORM'=>'Y',

'SHOW\_EDIT\_FORM'=>'Y',

'ADD\_READ\_ONLY\_FIELD'=>'N',

'EDIT\_READ\_ONLY\_FIELD'=>'N',

'SHOW\_FIELD\_PREVIEW'=>'N'

)

)

));

executeHook($queryUrl,$queryData);

}

//привязка к элементам

**function** link\_to\_element\_field\_add($Webhook\_URL,$block\_code,$field\_name,$link\_id)

{

// формируем URL в переменной $queryUrl

$queryUrl = "**$Webhook\_URL**/lists.field.add.json";

// формируем параметры для создания

$queryData = **http\_build\_query**(**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**",

'fields' => **array**(

'NAME' => "**$field\_name**",

'IS\_REQUIRED' => 'N',

'MULTIPLE' => 'N',

'TYPE'=>"E",

'LINK\_IBLOCK\_ID'=>"**$link\_id**",

'CODE'=>"Kompetencii",

'SETTINGS'=>**array**(

'SHOW\_ADD\_FORM'=>'Y',

'SHOW\_EDIT\_FORM'=>'Y',

)

)

));

executeHook($queryUrl,$queryData);

}

//привязка к элементам в виде списка

**function** link\_to\_element\_of\_list\_field\_add($Webhook\_URL,$block\_code,$field\_name,$link\_id)

{

// формируем URL в переменной $queryUrl

$queryUrl = "**$Webhook\_URL**/lists.field.add.json";

// формируем параметры для создания

$queryData = **http\_build\_query**(**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**",

'fields' => **array**(

'NAME' => "**$field\_name**",

'IS\_REQUIRED' => 'Y',

'MULTIPLE' => 'N',

'TYPE'=>"E:EList",

'USER\_TYPE\_SETTINGS'=>**array**(

"group"=>"Y",

"multiple"=>"Y"),

'LINK\_IBLOCK\_ID'=>"**$link\_id**",

'CODE'=>"Kompetencii",

'SETTINGS'=>**array**(

'SHOW\_ADD\_FORM'=>'Y',

'SHOW\_EDIT\_FORM'=>'Y',

)

)

));

executeHook($queryUrl,$queryData);

}

//получаем айди списка по код блоку

**function** GET\_list($Webhook\_URL,$block\_code)

{

// формируем URL в переменной $queryUrl

$queryUrl = "**$Webhook\_URL**/lists.get.json";

// формируем параметры для создания

$queryData = **http\_build\_query**(**array**(

'IBLOCK\_TYPE\_ID'=>'lists',

'IBLOCK\_CODE'=>"**$block\_code**"

));

// обращаемся к Битрикс24 при помощи функции curl\_exec

$curl = **curl\_init**();

**curl\_setopt\_array**($curl, **array**(

**CURLOPT\_SSL\_VERIFYPEER** => 0,

**CURLOPT\_POST** => 1,

**CURLOPT\_HEADER** => 0,

**CURLOPT\_RETURNTRANSFER** => 1,

**CURLOPT\_URL** => $queryUrl,

**CURLOPT\_POSTFIELDS** => $queryData,

));

$result = **curl\_exec**($curl);

**curl\_close**($curl);

$result = **json\_decode**($result, 1);

**if** (**array\_key\_exists**('error', $result))

{

**echo** "Ошибка при получении ID: ".$result['error\_description']."<br/>";

}

**else**

{

$list\_id=$result['result'][0]['ID'];

**return** $list\_id;

}

}

$Webhook\_URL=$\_REQUEST['Webhook'];

$step = **intval**($\_REQUEST['step']);

$result = '';

**switch** ($step) {

**case** 1:

//список 1 должности

$list\_name="Должности";

$block\_code="1231";

$csv\_import\_file="dolzhnosti.csv";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

elements\_add($Webhook\_URL,$block\_code,$csv\_import\_file);

**break**;

**case** 2:

//список 2 технологии

$list\_name="Технологии";

$block\_code="1232";

$csv\_import\_file="technologies.csv";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

elements\_add($Webhook\_URL,$block\_code,$csv\_import\_file);

**break**;

**case** 3:

//список 3 дисциплины

$list\_name="Дисциплины";

$block\_code="1233";

$csv\_import\_file="disciplines.csv";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

elements\_add($Webhook\_URL,$block\_code,$csv\_import\_file);

**break**;

**case** 4:

//список 4 кафедры

$list\_name="Кафедры";

$block\_code="1234";

$csv\_import\_file="cathedra.csv";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

elements\_add($Webhook\_URL,$block\_code,$csv\_import\_file);

**break**;

**case** 5:

//список 5 компетенции

$list\_name="Компетенции";

$block\_code="1235";

$csv\_import\_file="competencies.csv";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

elements\_add($Webhook\_URL,$block\_code,$csv\_import\_file);

**break**;

**case** 6:

//список 6 Владение компетенциями

$list\_name="Владение компетенциями";

$block\_code="1236";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

//создаем поля для списка 6

$field\_name="Выпускник";

$field\_code="Vypusknik";

CRM\_contact\_link\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

$field\_name="Компетенции";

$field\_code="Kompetencii";

$link\_block\_code="1235";

$link\_id=GET\_list($Webhook\_URL,$link\_block\_code);

link\_to\_element\_of\_list\_field\_add($Webhook\_URL,$block\_code,$field\_name,$link\_id);

**break**;

**case** 7:

//список 7 владение технологиями

$list\_name="Владение технологиями";

$block\_code="1237";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

//создаем поля для списка 7

$field\_name="Выпускник";

$field\_code="Vypusknik";

CRM\_contact\_link\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

$field\_name="Технологии";

$field\_code="Technology";

$link\_block\_code="1232";

$link\_id=GET\_list($Webhook\_URL,$link\_block\_code);

link\_to\_element\_of\_list\_field\_add($Webhook\_URL,$block\_code,$field\_name,$link\_id);

$field\_name="Организация";

$field\_code="Organizaciya";

CRM\_сompanies\_link\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

**break**;

**case** 8:

//список 8 воспоминания

$list\_name="Воспоминания";

$block\_code="1238";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

//создаем поля для списка 8

$field\_name="Выпускник";

$field\_code="Vypusknik";

CRM\_contact\_link\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

$field\_name="Преподаватель";

$field\_code="Prepodavatel";

CRM\_contact\_link\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

pozitive\_negative\_field\_add($Webhook\_URL,$block\_code);

**break**;

**case** 9:

//список 9 направления

$list\_name="Направления";

$block\_code="1239";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

level\_of\_education\_list\_field\_add($Webhook\_URL,$block\_code);

$field\_name="Код направления";

$field\_code="Kod\_napravlenya";

string\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

**break**;

**case** 10:

//список 10 нужные дисциплины

$list\_name="Нужные дисциплины";

$block\_code="1240";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

//создаем поля для списка 10

$field\_name="Выпускник";

$field\_code="Vypusknik";

CRM\_contact\_link\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

$field\_name="Дисциплина";

$field\_code="Disciplina";

$link\_block\_code="1233";

$link\_id=GET\_list($Webhook\_URL,$link\_block\_code);

link\_to\_element\_field\_add($Webhook\_URL,$block\_code,$field\_name,$link\_id);

**break**;

**case** 11:

//список 11 Периоды обучения

$list\_name="Периоды обучения";

$block\_code="1241";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

//создаем поля для списка 11

$field\_name="Выпускник";

$field\_code="Vypusknik";

CRM\_contact\_link\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

$field\_name="Направление";

$field\_code="Napravlenie";

$link\_block\_code="1239";

$link\_id=GET\_list($Webhook\_URL,$link\_block\_code);

link\_to\_element\_field\_add($Webhook\_URL,$block\_code,$field\_name,$link\_id);

$field\_name="Телефон во время обучения";

$field\_code="Telephone";

string\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

$field\_name="Год поступления";

$field\_code="God\_postupleniya";

number\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

$field\_name="Год окончания";

$field\_code="God\_okonchaniya";

number\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

rating\_list\_field\_add($Webhook\_URL,$block\_code);

**break**;

**case** 12:

//список 12 Периоды работы

$list\_name="Периоды работы";

$block\_code="1242";

lists\_add($Webhook\_URL,$list\_name,$block\_code);

//создаем поля для списка 12

$field\_name="Выпускник";

$field\_code="Vypusknik";

CRM\_contact\_link\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

$field\_name="Компания";

$field\_code="Kompany";

CRM\_сompanies\_link\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

$field\_name="Год начала";

$field\_code="God\_nachala";

number\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

$field\_name="Год окончания";

$field\_code="God\_okonchaniya\_raboty";

number\_field\_add($Webhook\_URL,$block\_code,$field\_name,$field\_code);

$field\_name="Должность";

$field\_code="Dolzhnost";

$link\_block\_code="1231";

$link\_id=GET\_list($Webhook\_URL,$link\_block\_code);

link\_to\_element\_field\_add($Webhook\_URL,$block\_code,$field\_name,$link\_id);

**break**;

}

?>

# ПРИЛОЖЕНИЕ Б.ТЕКСТ ПРОГРАММЫ СТРАНИЦЫ ЗАГРУЗКИ

**<**html**>**

**<**head**>**

**<**meta charset**=**"utf-8"**>**

**<**meta http**-**equiv**=**"X-UA-Compatible" content**=**"IE=edge"**>**

**<**meta name**=**"viewport" content**=**"width=device-width, initial-scale=1"**>**

**<!--** Latest compiled and minified CSS **-->**

**<link** rel**=**"stylesheet" href**=**"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity**=**"sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin**=**"anonymous"**>**

**<link** rel**=**"stylesheet" href**=**"https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css"**>**

**<**title**>**Импорт в Битрикс24**</**title**>**

**</**head**>**

**<**body**>**

**<**div **class=**"container"**>**

**<**h2**>**Импорт базы данных системы выпускников в Битрикс24**</**h2**>**

**<**div **class=**"row alert alert-info"**>**

**<**div **class=**"col-sm-6 col-xs-6 col-md-8 col-lg-9"**>**

Будем создавать и настраивать систему анализа выпускников**!**

**<**ul**>**

**<**li**>**Создаем универсальные списки**.</**li**>**

**<**li**>**Потом заполним списки**-**справочники данными**.</**li**>**

**<**li**>**Затем выполним настройку полей и привязку к CRM**.</**li**>**

**</**ul**>**

**</**div**>**

**<**div **class=**"col-sm-6 col-xs-6 col-md-4 col-lg-6"**>**

**<**div **class=**"input-group"**>**

**<**input type**=**"text" **class=**"form-control" placeholder**=**"Скопируйте URL вебхука" id**=**"Webhook" value**=**""**>**

**<**span **class=**"input-group-btn"**>**

**<button** **class=**"btn btn-warning" type**=**"button" id**=**"start-btn"**>**Импорт**</button>**

**</**span**>**

**</**div**><!--** **/**input**-**group **-->**

**</**div**>**

**</**div**>**

**<**div **class=**"row"**>**

**<**div **class=**"col-sm-12 col-xs-12 col-md-12 col-lg-12"**>**

**<**div **class=**"progress"**>**

**<**div **class=**"progress-bar progress-bar-striped active" id**=**"general-progress" role**=**"progressbar" aria**-**valuenow**=**"0" aria**-**valuemin**=**"0" aria**-**valuemax**=**"100" style**=**"min-width: 4em"**>**

**<**i **class=**"fa fa-spinner fa-spin fa-fw hidden"**></**i**>** **<**span id**=**"progress-value"**>**0**%</**span**>**

**</**div**>**

**</**div**>**

**</**div**>**

**</**div**>**

**</**div**>**

**<!--** jQuery **(**necessary **for** Bootstrap's JavaScript plugins) -->

**<**script src**=**"https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"**></**script**>**

**<!--** Include **all** compiled plugins **(**below**),** or include individual files as needed **-->**

**<**script src**=**"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity**=**"sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Txa" crossorigin**=**"anonymous"**></**script**>**

**<**script**>**

**var** step **=** 0**;**

**var** createData **=** **function** **()** **{**

console**.**log**(**'start'**);**

**var** Webhook **=** $**(** "#Webhook" **).**val**();**

**var** selfcall **=** arguments**.**callee**;**

**if** **(**step **<** 12**)** **{**

**var** url **=** 'http://pbashtovoy.tmweb.ru/My\_import.php?step=' **+** **(**step **+** 1**)** **+** '&Webhook=' **+** Webhook**;**

console**.**log**(**'step'**,** step**);**

**var** percent **=** **parseInt((**100 **/**12**)\*(**step **+** 1**));**

console**.**log**(**'step percent'**,** percent**);**

$**(** "#general-progress" **).**css**(**'width'**,** percent **+** '%'**);**

$**(** "#progress-value" **).**html**(**percent **+** '%'**);**

$**.**get**(** url**,** **function(** data **)** **{**

console**.**log**(**'step inside'**,** data**);**

$**(** "#log" **).**prepend**(** '<p><pre>' **+** data **+** '</pre></p>' **);**

**setTimeout(**selfcall**,** 1000**);**

**});**

**}**

**else** **{**

$**(** ".fa-spinner" **).**addClass**(**'hidden'**);**

**}**

step**++;**

**}**

$**(**'#start-btn'**).**click**(function()** **{**

$**(** ".fa-spinner" **).**removeClass**(**'hidden'**);**

createData**();**

**})**

**</**script**>**

**</**body**>**

**</**html**>**