

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
 «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет \_\_\_\_Информатика и вычислительная техника\_\_\_\_\_\_

(наименование факультета)

Кафедра \_\_\_\_\_ Информационные технологии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование кафедры)

Зав. кафедрой

\_«Информационные технологии»\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Б.В. Соболь

(подпись) (И.О.Ф.)

« \_\_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе по дисциплине \_\_\_\_Инструментальные средства информационных систем\_\_\_\_

на тему: Разработка веб-ориентированной информационной системы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Автор работы: \_\_\_\_\_\_\_\_\_ \_\_\_ Арарат Исраелян Михайлович \_\_\_\_

подпись И.О.Ф

Направление 09.03.02 «Информационные системы и технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код) наименование направления

«Информационные системы и технологии»

наименование профиля

Обозначение курсовой работы 09.03.02.740000.000КР Группа ВИС32

Руководитель работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_ доцент М.В.Ступина\_\_\_\_

подпись (должность, И.О.Ф)

Работа защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата оценка подпись

Ростов-на-Дону

2020



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
 «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет \_\_\_\_Информатика и вычислительная техника\_\_\_\_\_\_

(наименование факультета)

Кафедра \_\_\_\_\_ Информационные технологии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование кафедры)

Зав. кафедрой

\_«Информационные технологии»\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Б.В. Соболь

(подпись) (И.О.Ф.)

« \_\_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**ЗАДАНИЕ**

на курсовую работу

Обучающийся \_\_\_ Арарат Исраелян Михайлович \_\_\_ Код \_1638674 Группа \_ ВИС32\_\_

Обозначение 09.03.02.740000.000КР

Тема: Разработка веб-ориентированной информационной системы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Срок представления работы к защите: « \_\_\_ » \_\_\_\_\_\_ 2020 г.

Исходные данные для курсовой работы:

|  |
| --- |
| Техническое задание, данные таблицы, учебная литература по теме курсовой работы |

Содержание пояснительной записки:

Наименование и содержание разделов

|  |
| --- |
| 1. Описание предметной области |
| 1. Техническое задание |
| 1. Программная реализация |
| 3.1 Инструментальные средства разработки |
| 3.2 Реализация серверной части приложения |
| 3.3 Реализация клиентской части приложения |
| 1. Заключение |
| 1. Список литературы |

ЗАКЛЮЧЕНИЕ

|  |
| --- |
| В результате выполнения курсовой работы был проведён анализ поставленной проблемы с |
| последующего ее решением. С помощью программного обеспечения PHPMyAdmin и средствами |
| фреймворка Yii2 была спроектирована база данных строительной компании и разработано веб- |
| приложение, в котором реализованы запросы, формы для ввода данных и модуль администратора. |

Перечень графического материала**:**

Нет\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ доцент М.В.Ступина \_\_

подпись, дата И.О.Ф

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_А.М.Исраелян\_\_\_\_\_\_\_

подпись, дата И.О.Ф

Оглавление

[Описание предметной области 4](#_Toc41768359)

[Техническое задание 5](#_Toc41768360)

[Программная реализация 6](#_Toc41768361)

[Инструментальные средства разработки 6](#_Toc41768362)

[Реализация серверной части приложения 9](#_Toc41768363)

[Реализация клиентской части приложения 17](#_Toc41768364)

[Заключение 24](#_Toc41768365)

[Список литературы 25](#_Toc41768366)

Описание предметной области

Web-приложение строительной организации должно предоставлять пользователю информацию разного рода: перечень техники, состав сотрудников, отчёты о выполненных работах, подсчёт стоимости работ.

Цель данного курсового проекта заключается в выполнении расчета финансовых результатов деятельности предприятия и показателей эффективности работы создаваемой строительной организации, учёта строительной техники и используемых материалов, а также отразить списки сотрудников и необходимые документы для прохождения регистрации и получения разрешения на строительную деятельность в данной организации.

Для реализации данной цели необходимо решить следующие задачи:

1. Создание и реализация базы данных строительной организации;
2. Реализовать поиск разного рода информации по заданным критериям;
3. Добавить поддержку административного модуля;
4. Создание понятного пользовательского интерфейса для взаимодействия с веб-приложением.

Функционал автоматизированной системы включает в себя:

* Возможность поиска информации;
* Возможность получения отчётов;
* Возможность добавления, изменения и удаления данных.

Для работы с базами данных используется система управления базами данных (СУБД) MySQL. Они предназначены для выполнения всей совокупности функций, связанных с созданием и эксплуатацией базы данных информационной системы.

Разработки веб-приложения осуществляется на базе PHP-фреймворка Yii2, который подходит для проектов любого типа и сложности.

Техническое задание

Строительная организация занимается строительством различного рода объектов по договорам с заказчиками. Каждый объект имеет характеристики, например, адрес объекта, этажность, количество помещений, площадь.

Строительная организация состоит из нескольких строительных управлений, которые ведут работы на одном или нескольких объектах, возглавляемых начальником управления, где трудятся группа техников, инженеров и прорабов. Рабочие объединяются в бригады. Каждому строительному управлению придается строительная техника, распределённая по объектам.

Технология строительства предполагает выполнения определенного набора работ для каждого объекта. На каждый объект составляется график работ и смета. По результатам выполнения работ составляется отчет с указанием сроков выполнения работ и расходов материалов.

Основное:

1. Определить первичные и внешние ключи сущностей, типов и диапазонов допустимых значений атрибутов.
2. Структура таблиц и ограничений ссылочной целостности должны адекватно отражать текущее состояние предметной области.
3. Разработка приложения на базе фреймворка Yii2 и реализация следующих возможностей:

* Вывод результатов запросов
* Наличие форм фильтрации данных
* Административный модуль, закрытый для гостей.

Дополнительное:

1. Вывод результатов запросов.
2. Настройка ЧПУ и роутинга.
3. Создание и подключение собственного виджета.
4. Использование хелперов.

Программная реализация

Инструментальные средства разработки

Первый этап – проектирование и создание таблиц базы данных (БД). Структура БД представлена на рисунке 1. Создание базы данных осуществляется посредством веб-интерфейса для администрирования СУБД MySQL – PНPmyAdmin. Были созданы следующие таблицы: brigades (бригады), dolzhnosti (должности), graphic\_objects (график объектов), material (строительные материалы), objects (объекты), sotrudniki (сотрудники), stroi\_uprav (строительные управления), tehnika (техникаа), zakazchiki (заказчики), users (пользователи). Определены первичные и внешние ключи, сущности, типы и диапазон допустимых значений для каждой таблицы, а также реализованы связи между таблицами.

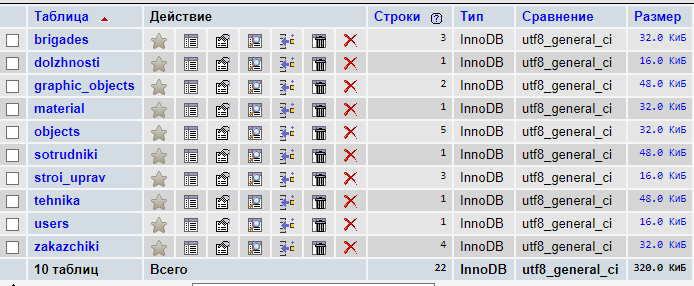


Рисунок 1 - Структура базы данных «company»

Рассмотрим структуру таблицы «сотрудники», представленную на рисунке 2. Поле id\_sotrudnika является первичным ключом, а также дополнительно мы указываем, что это поле является AUTO\_INCREMENT и уникальным. Поля id\_brigad (код бригады) и id\_dolzhnosti (код должности) являются индексами, таким образом, они являются внешними ключами для связующих таблиц, тип данных int(11). Firstname (имя), surname (фамилия), phone\_number (номер телефона) имеют тип данных varchar(255), varchar(255) и varchar(15) соответственно. Хранят информацию об имени, фамилии и номере мобильного телефона сотрудника. Поле passport (паспортные данные) имеет тип данных int(11) и хранит в себе серию и номер паспорта. Поля date\_naim, date\_birth имеют тип данных DATE, формат гггг-мм-дд. Хранят данные о дате рождения работника и дате найма соответсвенно.



Рисунок 2 - Структура таблицы «sotrudniki»

Таблица users имеет следующую структуру. Поле id является ключевым, тип данных int(11), AUTO\_INCREMENT. Username и password хранят информацию об имени пользователя и пароле соответственно. Тип данных – varchar (255). Поле auth\_key, varchar (255), будет использоваться в дальнейшем для авторизации на основе данных куки.



Рисунок 3 - Структура таблицы «users»

По аналогии были созданы таблицы brigades, dolzhnosti, graphic\_objects, material, objects, stroi\_uprav, tehnika, zakazchiki.

Далее идет процесс создания связей. Для этого достаточно нажать на кнопку создать связь и выбрать поле, которое является первичным ключом в основной таблице и поле, являющееся вторичным ключом (индексом) в подчиненной.

Схема связей между таблицами продемонстрирована на рисунке 4.

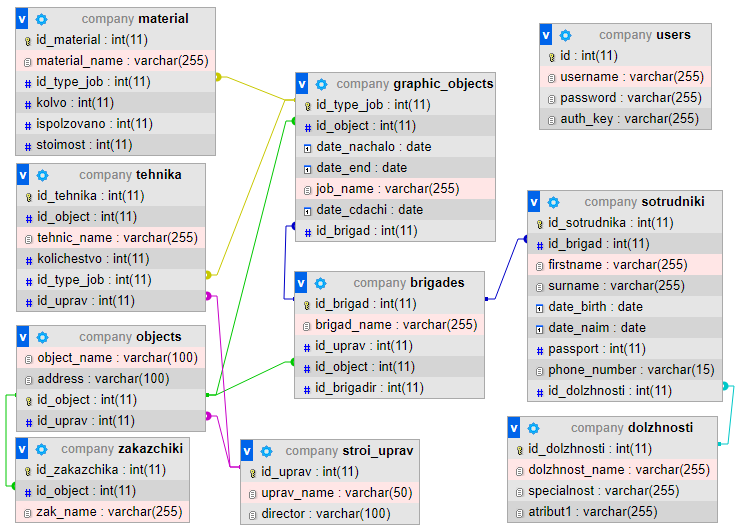


Рисунок 4 – Связи между таблицами

Завершив процесс создания таблиц и установки связей, начинается этап их заполнение. Добавлять новые записи в таблицы можно через формы, которые в дальнейшем будут созданы. Пример заполненной таблицы на рисунках 5 и 6.

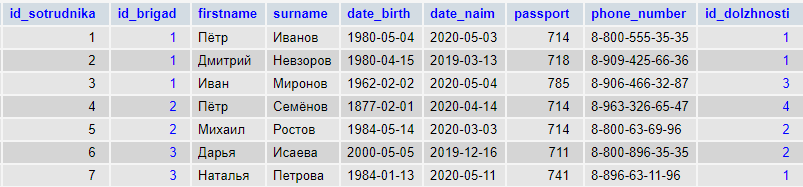


Рисунок 5 – Данные в таблице «sotrudniki»



Рисунок 6 – Данные в таблице «users»

Реализация серверной части приложения

Для начала необходимо подключть базу данных к Yii2. Код продемонстрирован на рисунке 7. Для этого в поле dns нужно указать название базы данных «company», в полях user и password - имя пользователя и пароль для доступа к PHPMyAdmin.

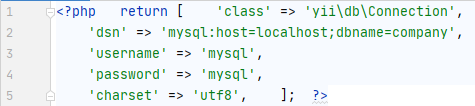


Рисунок 7 – Подключение к базе данных

Следующим этапом является создание моделей всех таблиц. Делается это с помощью модуля Gii, входящего в состав приложения Yii. Его можно использовать для автоматической генерации кода моделей. На рисунке 8 представлены все сгенерированные модели.

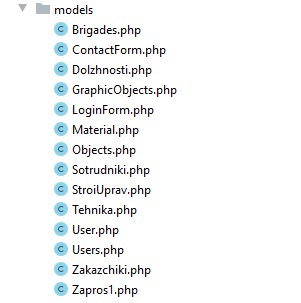


Рисунок 8 – Модели

Для удобства при создании форм и фильтров для выбора критериев поиска информации была вручную создана модель Zapros1, которая представлена на рисунке 9. Данная модель позволяет осуществлять динамические запросы к базе данных и проверяет, чтобы все вводимые данные не были пустыми и соответствовали типу данных таблиц.



Рисунок 9 – Модель «zapros1»

Модель «sotrudniki» продемонстрирована на рисунке 10. В функции rules() указываются правила для полей. Функция tableName() возвращает имя таблицы. В функции attributeLabels() указываются метки атрибутов для каждого столбца таблицы. getDolzhnosti() и getBrigad() – это функции связей между таблицами. В данном случае, тип связей один к одному.



Рисунок 10 – Модель «sotrudniki»

Для работы с моделями их необходимо подключить к контроллеру. Код показан на рисунке 11.

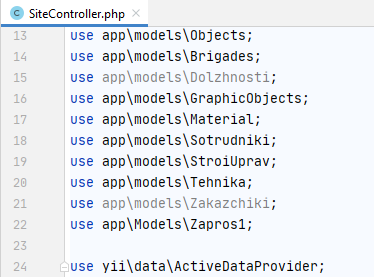
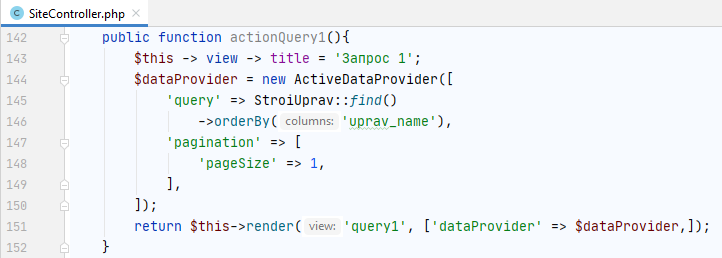


Рисунок 11 – Подключение моделей к контроллеру

Следующим этапом является формирование запросов. Используем для этого класс ActiveDataProvider, подключив его после моделей, рисунок 11 Рассмотрим наиболее интересные запросы.

Запрос №1 (рис. 12). Получить перечень строительных управлений и/или участков и их руководителей. Код запроса продемонстрирован на рисунке 12. В dataProvider указывается простой запрос на выборку данных из таблицы строительные управления с последующей сортировкой по названию управления, а в конце возвращается вся информацию. Реализована поддержка пагинации.

Рисунок 12 – Код запроса №1

Запрос №4. Получить состав бригад, работавших (работающих) на строительстве указанного объекта. В отличие от предыдущего запроса, здесь используется экземпляр модели Zapros1. Если данные были переданы контроллеру, то выполняется запрос. Ключевое слово JoinWith объединяет таблицы по существующим названиям связей, указанных в моделях таблиц. В конструкции where указаны условия выборки записей. В данном случае, код объекта должен соответствовать коду, выбранному пользователем через фильтр в представлении. В конце выполняется сортировка по названию бригады. Добавлена поддержка пагинации. Код запроса представлен на рисунке 13.



Рисунок 13 – Код запроса №4

Запрос №13. Получить перечень бригад, выполненных указанный вид строительных работ в течение обозначенного периода времени с указанием объектов, где эти работы выполнялись. Код запроса представлен на рисунке 14. Здесь используется экземпляр модели Zapros1. Если данные были переданы контроллеру, выполняется запрос на выборку данных из таблицы бригады с объединением с таблицей объекты и график объектов с помощью ключевого слова JoinWith. В запросе используются три условия для выборки. Сначала проверяется, что код вида работы соответствует коду, выбранному пользователем в фильтре. Далее проверяется, что дата начала работ и дата окончания соответствовали с датами, введенными пользователем. Добавлена поддержка пагинации. В конце возвращается вся информация.



Рисунок 14 – код запроса №13

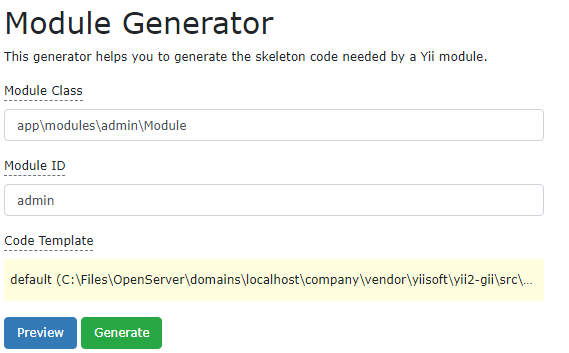
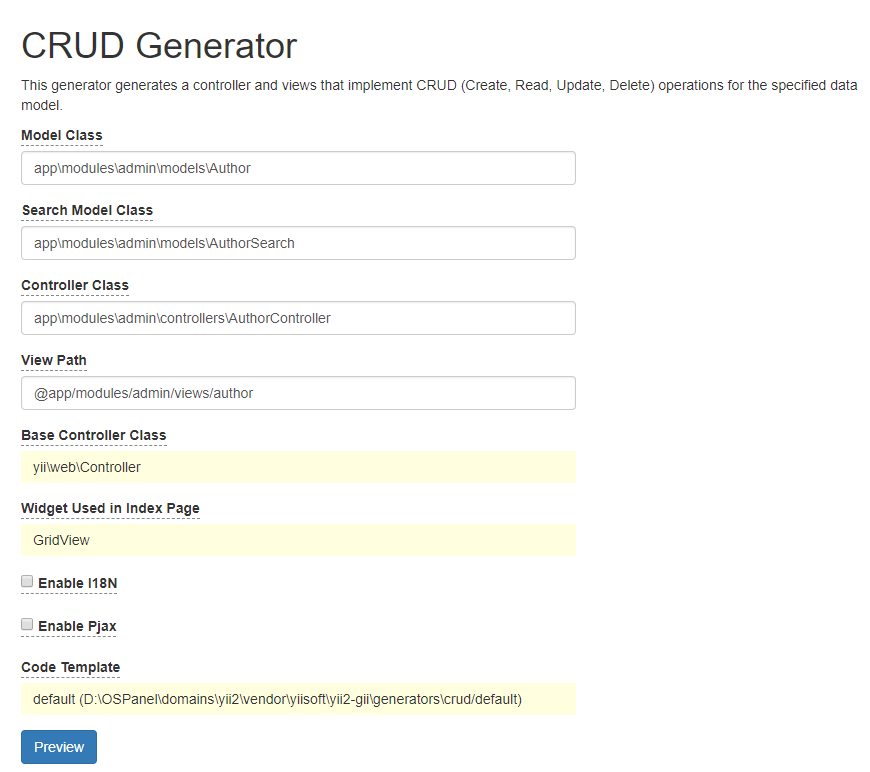
После формирования запросов необходимо создать формы для работы с таблицами базы данных. С помощью расширения Gii создаются административный модуль, представленный на рисунке 15, контроллеры и модели таблиц, представленные на рисунках 16. Окончательная структура административного модуля представлена на рисунке 17.

Рисунок 15 – Module генератор Рисунок 16 – CRUD генератор

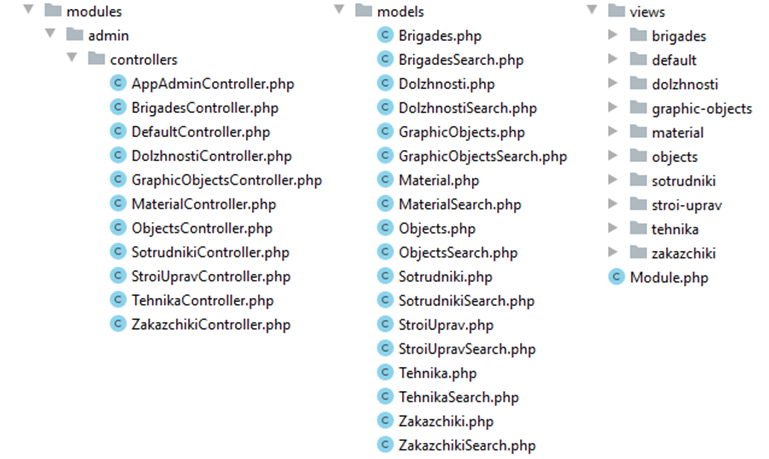


Рисунок 17 – Структура административного модуля

Для того, чтобы вместо id выводились названия, необходимо изменить каждую Search модель следующим образом: в запросе, казанном в функции search, объединить все необходимые таблицы с помощью ключевого слова JoinWith. В качестве примера настроим модели сотрудники и заказчики. Код моделей ZakazchikiSearch и SotrudnikiSearch отображен на рисунках 18 и 19 соответственно.



Рисунок 19 – Модель «ZakazchikiSearch»

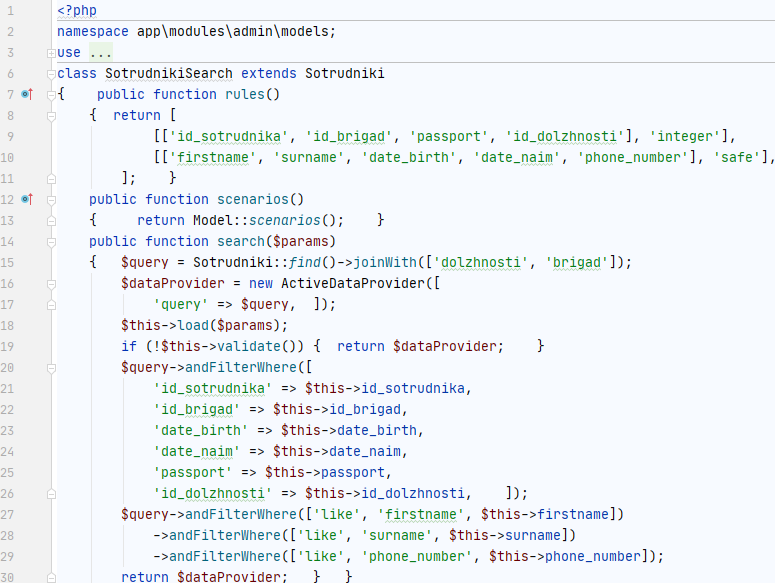


Рисунок 18 – Модель «SotrudnikiSearch»

Следующим этапом настраивается ЧПУ и роутинг. Добавим компонент UrlManager в файл web.php, как на рисунке 20. Он используется для анализа входящих запросов, а также для создания новых URL-адресов программным способом. Далее необходимо настроить правила для обработки адресов. Делается это следующим образом: слева прописывается адрес страницы, а справа маршрут.



Рисунок 20 – Настройка ЧПУ и роутинга

Дополнительно был реализован собственный виджет для вывода данных из запроса №1, код которого представлен на рисунке 21. Был прописан сам запрос и настроена пагинация.



Рисунок 21 – Код собственного виджета

Реализация клиентской части приложения

Клиентская часть веб-приложения – это графический интерфейс, отображающийся в браузере пользователя. В фреймворке Yii, использующий шаблон проектирования MVC, эта часть представляет собой файлы представлений, которые отображают данные, поступающие от контроллера.

В качестве примера, рассмотрим файлы представлений для каждого из описанных ранее запросов.

На рисунке 22 представлен код файла представления запроса №1. Для отображения данных из запроса используется предварительной подключенный GridView виджет. Страница запроса №1 представлена на рисунке 23.

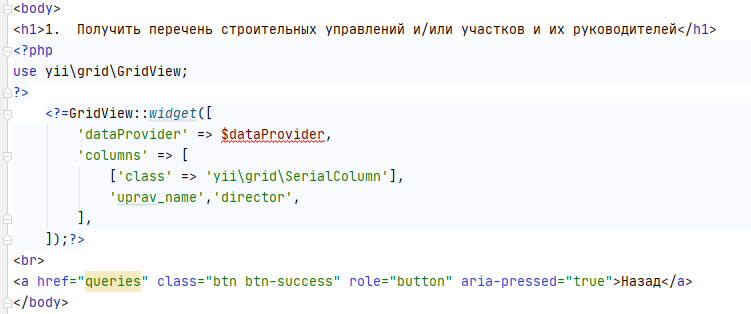


Рисунок 22 – Код представления запроса №1

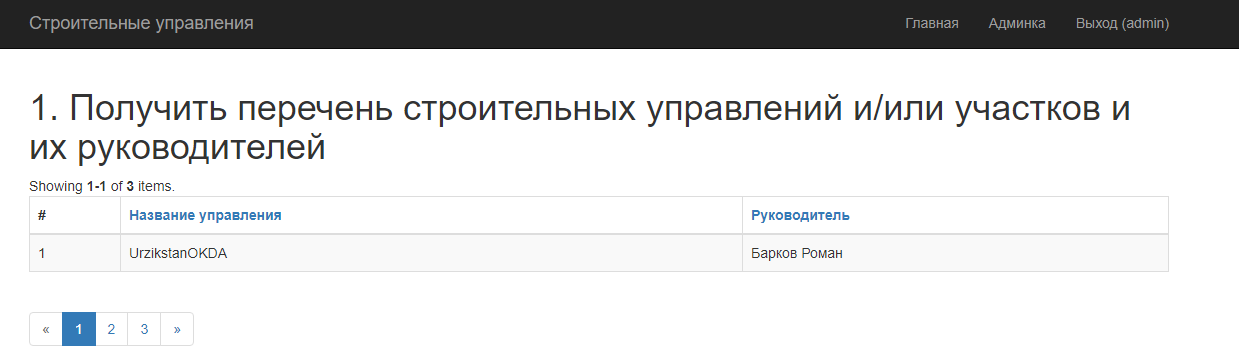


Рисунок 23 – Представление запроса №1

В отличие от предыдущего, для запроса №4 был добавлен выпадающий список, который используется для динамического формирования запросов. Подключается ArrayHelper::map для получения пар ключ-значение из массива объектов таблицы, виджет ActiveForm для отправки данных наших форм, GridView виджет для отображения необходимых дынных из запроса. Код представления показан на рисунке 24, а страница запроса на рисунке 25. Фильтр реализован в виде выпадающего списка благодаря использованию ArrayHelper::map. Для него используется подзапрос, позволяющий выводились названия объектов, по которым в дальнейшем необходимо выполнить поиск.



Рисунок 24 – код представления запроса №4

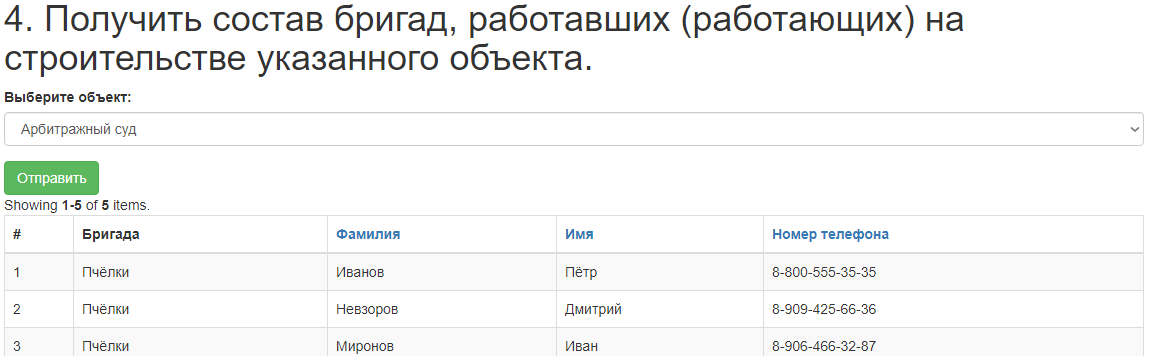


Рисунок 25 – Представление запроса №4

Представление запроса №13 было реализовано по аналогии с предыдущим. Для выбора даты был использован виджет DatePicker. Код представления и его страница отображены на рисунках 26 и 27 соответственно.

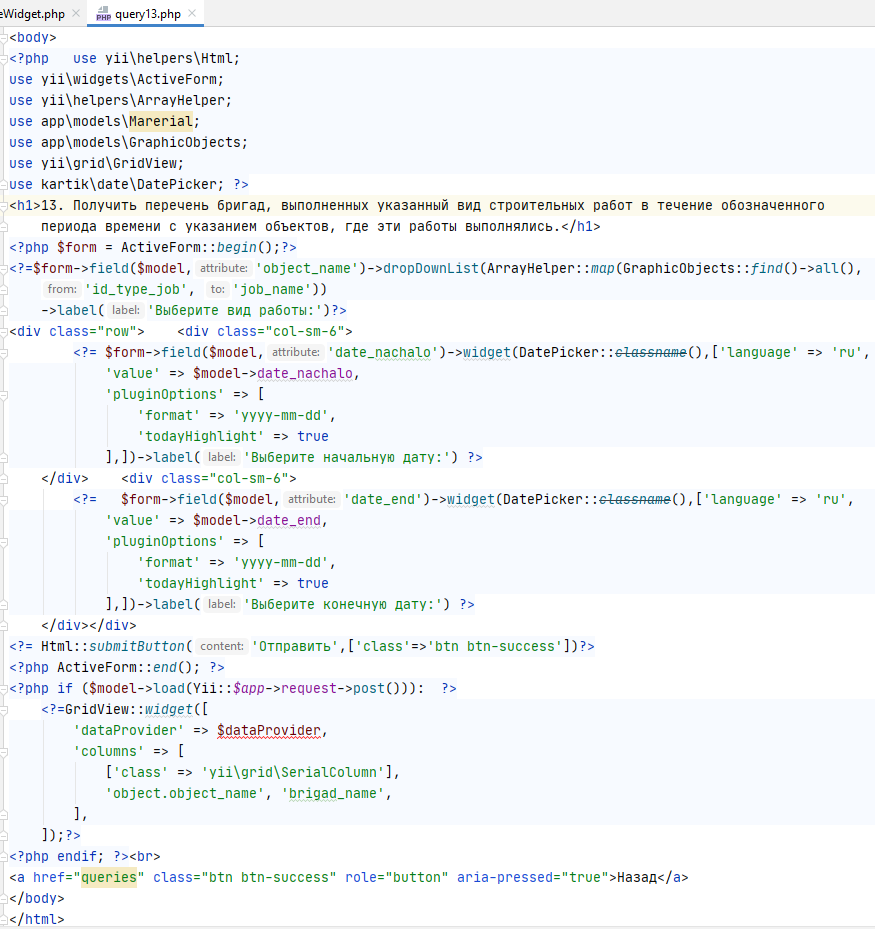


Рисунок 26 – код представления запроса №13.

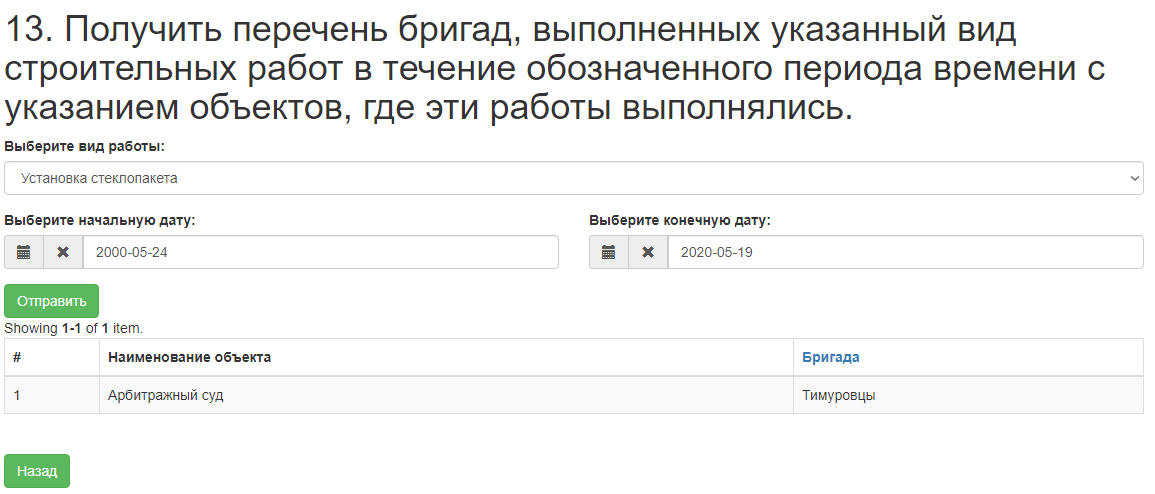


Рисунок 27 – Представление запроса №13

Код собственного виджета представлен на рисунке 28. Внутри представления запрос 1 происходит вызов виджета статичным методом widget(), который показан на рисунке 29.



Рисунок 28 – Код представления виджета

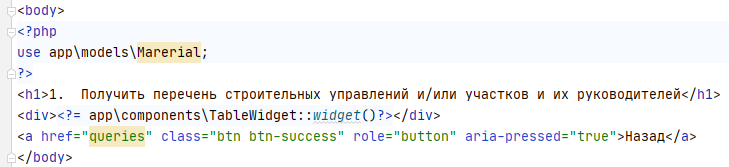


Рисунок 29 – Код для вызова виджета

Рассмотрим представления, сгенерированные CRUD-генератором. В качестве примера рассмотрим таблицу Сотрудники. Модулем Gii были сгенерированы все представления. Чтобы вместо id вывести данные связанных полей из других таблиц, необходимо внести изменения, представленные на рисунке 30. На рисунке 31 отображается конечный вид представления.



Рисунок 30 – Код вывода связанного поля из другой таблицы

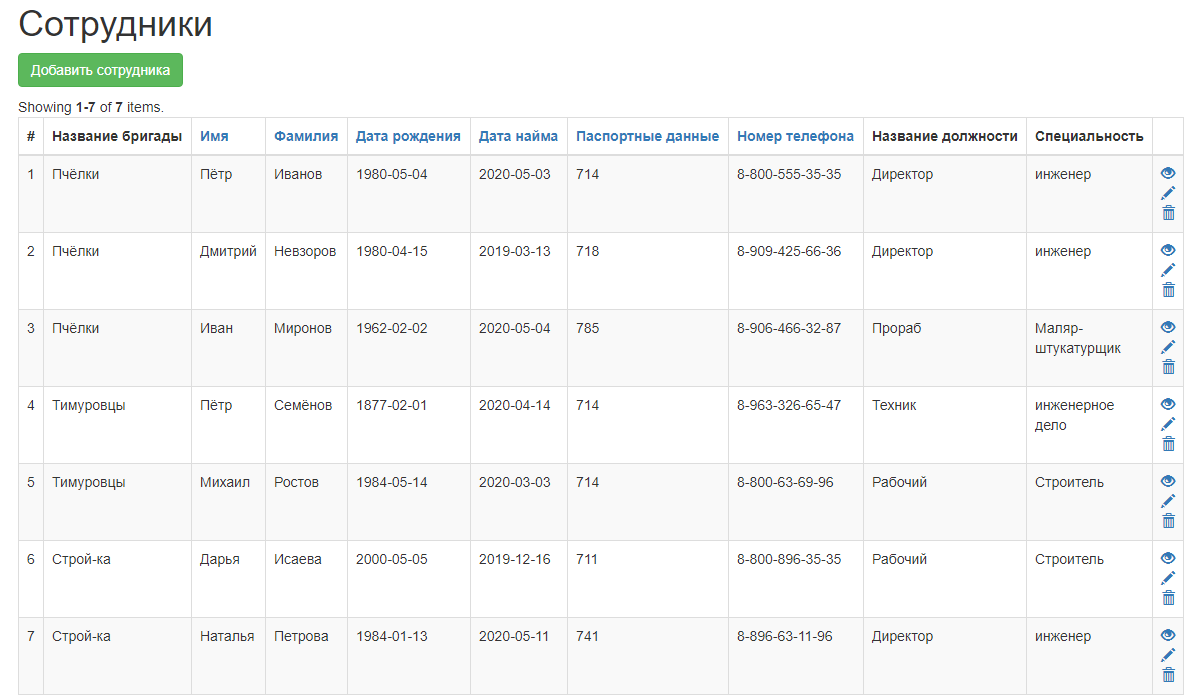


Рисунок 31 – Страница представления таблицы Сотрудники

На рисунке 32 представлено отображение связанных данных при просмотре отдельной записи.

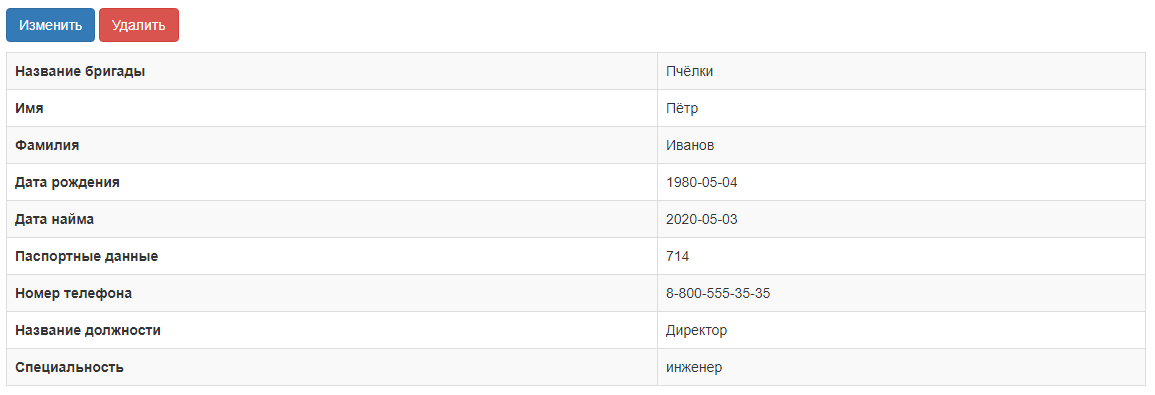


Рисунок 32 – Отображение отдельной записи.

Настроена форма для добавления и изменения данных. Код представлен на рисунке 33. С помощью ArrayHelper::map реализованы выпадающие списки для выбора данных из связанных таблиц.



Рисунок 33 – Код формы для добавления и изменения записей в таблицу Сотрудники

Само представление показано на рисунке 34.

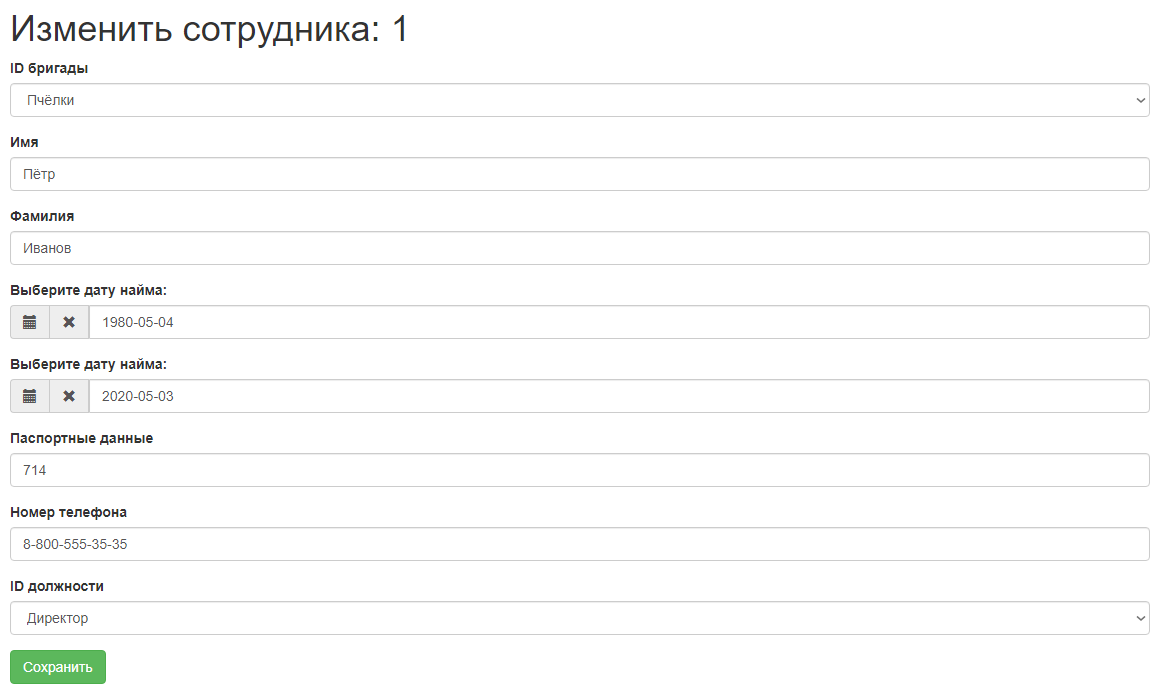


Рисунок 34 – Форма для таблицы Сотрудники

Помимо этого, все заголовки были переведены на русский язык. Для ввода даты используется виджет DatePicker, который представляет собой удобный интерактивный календарь, облегчающий ввод дат. Для этого в типе вводимых данных был указан DatePicker. Связанные данные отображаются не по id при просмотре таблиц, добавлении новых записей и изменении существующих данных. Доступ ко всем формам имеет только администратор.

Заключение

В результате выполнения курсовой работы был проведён анализ поставленной проблемы с последующего ее решением. С помощью программного обеспечения PHPMyAdmin и средствами фреймворка Yii2, была спроектирована база данных строительной компании и разработано веб-приложение, в котором реализованы запросы, формы для ввода данных и модуль администратора.

Благодаря веб-приложению, пользователь может выполнять запрос информации по указанным критериям поиска. Администратор получает доступ на добавление, удаление и изменение данных. Реализованы: роутинг, пагинация, использование хелперов и вывод запросов посредством использования подключенных и собственного виджетов.

Разработанная база данных может существенно упростить работу оператора, облегчить процедуру получения справочной информации, необходимой клиентам.

В процессе работы были приобретены навыки анализа производства, применены и усовершенствованы полученные знания в области разработки систем управления базами данных для решения конкретной задачи.

Список литературы

1. Сафронов М. Разработка веб-приложений в Yii 2. -М.: ДМК Пресс, 2015. -392 с.: ил. Клименко, Роман Веб-мастеринг на 100% / Роман Клименко. - М.: Питер, 2015. - 152 c.
2. МакГрат, Майкл PHP7 для начинающих с пошаговыми инструкциями / Майк МакГрат – Москва: Издательство «Э», 2018 – 256 с.
3. Томсон Лаура Разработка Web-приложений на РНР и MySQL: Пер. с англ./Лаура Томсон, Люк Вел-линг. — 5-е изд., 2017. — 672 с.
4. Дронов, В.А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов / В.А. Дронов. - М, 2016. - 226 c.
5. Сафронов Марк Разработка веб-приложений в Yii 2; ДМК Пресс - М., 2016. - 392 c.