

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО”

Факультет Инфокоммуникационных технологий

Образовательная программа Интеллектуальные системы в гуманитарной сфере
(Академический бакалавр, Очная ф/о)

Направление подготовки (специальность) 45.03.04 - Интеллектуальные системы в гуманитарной сфере

О Т Ч Е Т

по курсовой работе по дисциплине «Основы web-программирования»

Тема задания: Разработка web-приложения для сотрудников биржи труда

Обучающийся: Максимов Никита Александрович, К3342

Руководитель курсовой работы: Говоров Антон Игоревич, ассистент

Оценка по курсовой работе ____

Подписи членов комиссии:

____ Говоров А. И.
(подпись)

____ Чунаев А. В.
(подпись)

____ Антонов М. Б.
(подпись)

Дата ____

Санкт-Петербург
2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. Предметная область	4
1.1. Общее описание предметной области.....	4
1.2. Функциональные требования к web-сервису в рамках предметной области	4
1.3. Выводы	5
2. Описание серверной части	6
2.1. Средства разработки серверной части	6
2.2. Разработка моделей данных для создаваемого web-приложения.....	6
2.2.1 Теория	6
2.2.2 Разработанные модели	6
2.3. Сериализаторы	7
2.4. Создание представлений	8
2.5. Интерфейсы Django REST.....	8
2.6. Выводы	9
3. Описание клиентской части	10
3.1. Средства разработки клиентской части	10
3.2. Разработанные интерфейсы Vue.....	10
3.2.1 Страница входа	10
3.2.2 Страница регистрации.....	11
3.2.3 Главная страница	12
3.2.4 Страница вакансий.....	12
3.2.5 Страница работодателей	14
3.2.6 Страница соискателей	15
3.2.7 Страница пособий.....	16
3.2.8 Страница курсов.....	17
3.3. Выводы	17
ЗАКЛЮЧЕНИЕ	18
СПИСОК ЛИТЕРАТУРЫ.....	19
Приложение А. Модели	20

ВВЕДЕНИЕ

Цель выполнения курсовой работы заключается в демонстрации навыков и умений реализации web-сервисов, полученных во время обучения по дисциплине «основы web-программирования», с использованием таких инструментов и компонентов, как Django, Django REST Framework, Vue.js, Muse-UI.

Были сформулированы следующие задачи курсовой работы:

1. Анализ предметной области;
2. Разработка функциональных требований к разрабатываемому web-сервису;
3. Создание модели данных;
4. Реализация серверной части web-приложения с использованием Django REST Framework;
5. Реализация клиентской части web-приложения средствами Vue.js и организация связи между серверной и клиентской частями.

В рамках курсовой работы было необходимо разработать программную систему, предназначенную для администратора биржи труда.

1. Предметная область

1.1. Общее описание предметной области

Предметной областью данной курсовой работы является администрирование гостиницы.

Задачей курсовой работы являлось создание информационной системы, главной целью которой является хранение информации о соискателях и вакансиях, а также формирование резюме соискателя, вывод данных о работодателях, состоянии вакансии, проходящих курсах и т.д.

В отчетах также должны фигурировать данные о соискателях, закрытых вакансиях, проводимых курсах, работодателях

Для каждого соискателя должна присутствовать возможность составить резюме, осуществить подбор вакансии по профессии и образованию, начислить пособие.

Необходима возможность для соискателей пройти курсы, после прохождения которых соискатели получают разряд по освоенной профессии. Для работодателей должны фиксироваться их контактные данные (название, адрес, контактное лицо, телефон, электронный адрес), дата подачи вакансии, состояние вакансии. Работодатель при размещении вакансии должен указать в заявке перечень требуемых профессий, образование соискателя, требуемый стаж, разряд, заработную плату.

1.2. Функциональные требования к web-сервису в рамках предметной области

Для работодателей в БД должны храниться следующие данные: название, адрес, контактное лицо, телефон, электронный адрес, дата подачи вакансии, состояние вакансии. Для вакансии: требуемая профессия, образование соискателя, требуемый стаж, разряд, заработную плату. О соискателе должна храниться информация об образовании, ФИО, адрес, возраст, трудовой стаж.

Администратор биржи труда должен иметь следующие возможности:

- Просмотреть каталог вакансий, с возможностью их фильтрации.
- Просмотреть подробную информацию о соискателе, в том числе сведения о назначенных пособиях, полученных сертификатах курсов.

Администратор биржи труда может выполнять следующие операции:

- Зарегистрировать нового соискателя на бирже труда.
- Разместить новую вакансию.

- Назначить пособие соискателю.

1.3. Выводы

В результате проведённого анализа предметной области были сформулированы функциональные требования к web-приложению, реализуемому в рамках курсовой работы.

В последующих главах описаны процессы разработки модели данных, а также серверной и клиентской части приложения.

2. Описание серверной части

2.1. Средства разработки серверной части

Для реализации серверной части приложения были использованы следующие средства разработки:

- PostgreSQL – свободная объектно-реляционная система управления базами данных. Использование данной технологии позволяет создавать и хранить базу данных на сервере, что способствует гибкой передаче web-приложения на стадию производства.
- web-фреймворк Django REST языка программирования Python, основанный на технологии REST— стиль построения архитектуры распределенного приложения. Данные в REST должны передаваться в виде небольшого количества стандартных форматов (например HTML, XML, JSON). Сетевой протокол, как и HTTP, должен поддерживать кэширование, не должен зависеть от сетевого слоя, не должен сохранять информацию о состоянии между парами «запрос-ответ». Утверждается, что такой подход обеспечивает масштабируемость системы и позволяет ей эволюционировать с новыми требованиями.

2.2. Разработка моделей данных для создаваемого web-приложения

2.2.1 Теория

Модель данных в Django – это одиночный определённый источник информации о данных разработчика. Модели определяют структуру хранимых данных, включая типы полей и, возможно, их максимальный размер, значения по умолчанию, параметры списка выбора, текст справки для документации, текст меток для форм и т. д.

2.2.2 Разработанные модели

На основании предметной области и выявленных функциональных требований была разработана схема базы данных, представленная на рисунке 1.

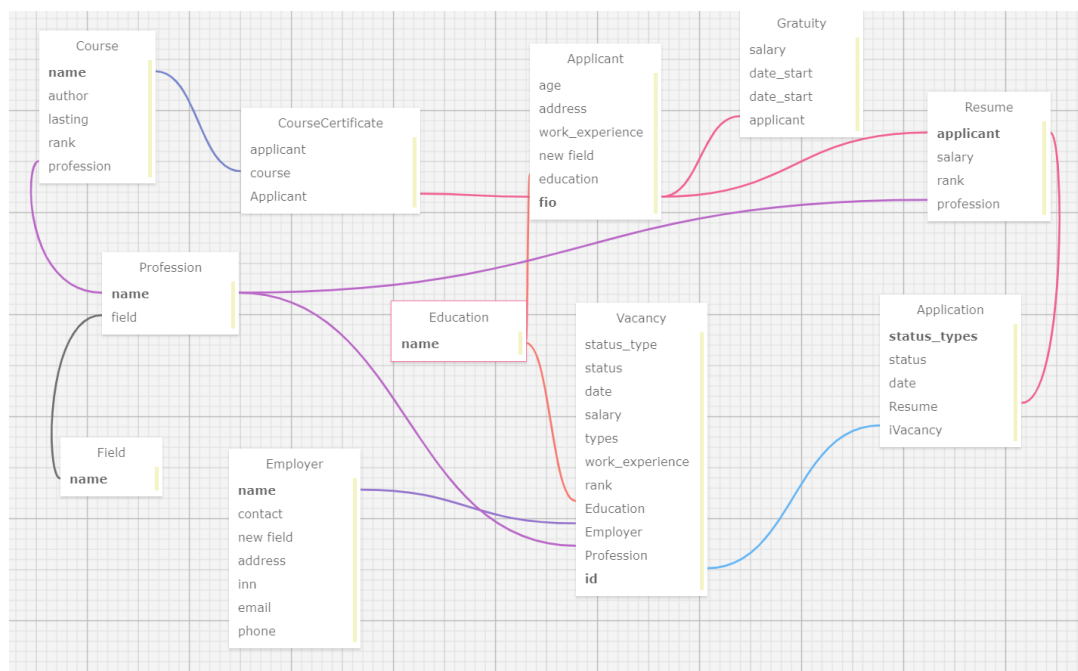


Рисунок 1 – Схема базы данных разрабатываемого web-приложения

Спроектированная модель данных была реализована в качестве классов моделей Django в файле models.py. Были созданы следующие модели:

- Profession – Профессия. Данная модель содержит информацию о профессиях
- Applicant – Соискатель, все данные соискателя, такие как ФИО и адрес
- Resume – Данные о резюме
- Course – Курсы, данные доступных курсов
- CourseCertificate – Сведения о прохождении курсов соискателем
- Gratuity – Пособие, которое можно назначить соискателю
- Employer – Работодатель. Подробные сведения о работодателе
- Vacancy – Вакансия. Сведения о конкретной вакансии
- Application – Подача заявки на желаемую вакансию

2.3. Сериализаторы

Сериализация — это процесс создания потокового представления данных, которые мы можем передавать между серверной частью, Django REST Framework, и клиентской частью, реализованной с помощью Vue.js. В случае Django, сериализаторы переводят данные из базы данных в формат json, и расположены в файле serializers.py (рис. 2).

```

1 from rest_framework import serializers
2 from .models import Education, Profession, Field, Applicant, Resume, Course, CourseCertificate, Gratuity
3
4
5 class EducationSerializer(serializers.ModelSerializer):
6     """Сериализатор для модели """
7
8     class Meta:
9         model = Education
10        fields = "__all__"
11
12
13 class ProfessionSerializer(serializers.ModelSerializer):
14     """Сериализатор для модели """
15     field = serializers.SlugRelatedField(slug_field="name", read_only=True)
16
17     class Meta:
18         model = Profession
19         fields = "__all__"
20
21

```

Рисунок 2 – Некоторые сериализаторы из файла serializers.py

2.4. Создание представлений

В Django представления задаются в файле views.py. Для создания отображений для выполнения запросов использовался класс APIView. На рис. 3 приведены некоторые реализованные отображения.

```

class FieldViewSet(viewsets.ModelViewSet):
    """Отображение для модели """
    queryset = Field.objects.all()
    serializer_class = FieldSerializer

class ApplicantViewSet(viewsets.ModelViewSet):
    """Отображение для модели """
    queryset = Applicant.objects.all()

    def get_serializer_class(self):
        if self.action == 'create':
            return ApplicantCreateSerializer
        elif self.action == 'update':
            return ApplicantUpdateSerializer
        else:
            return ApplicantSerializer

```

Рисунок 3 – Отображения из файла views.py

2.5. Интерфейсы Django REST

В панели Django REST можно проверить результаты разработки серверной части.

Так, например, на рисунке 4 представлено отображение данных обо всех существующих работодателях.

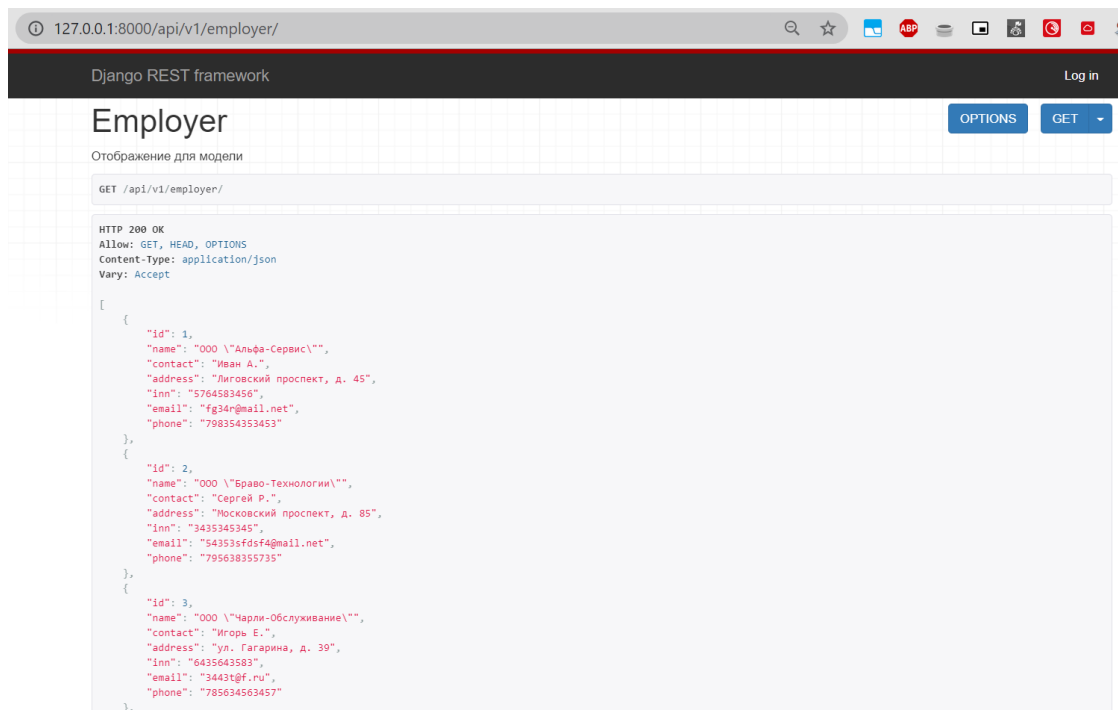


Рисунок 4 – Вывод всех работодателей в Django REST

Добавление новой записи о соискателе создание нового объекта модели Applicant средствами Django REST представлено на рисунке 5.

Raw data HTML form

ФИО

Возраст

Адрес

Опыт работы

Education

POST

Рисунок 5 – Добавление новой записи в Django REST

2.6. Выводы

Таким образом, в процессе разработки серверной части web-приложения были созданы сериализаторы – файл `serializers.py`, прописаны пути в файл `urls.py` и представления – в файл `views.py`. Это позволило выполнить задачи, поставленные в рамках обзора необходимых функциональных требований.

Далее приведено описание клиентской части разрабатываемого web-приложения.

3. Описание клиентской части

3.1. Средства разработки клиентской части

Клиентская часть разрабатываемого web-приложения была реализована с помощью средств JavaScript-фреймворка Vue.js и библиотеки MUSE-UI для дизайна пользовательского интерфейса.

3.2. Разработанные интерфейсы Vue

Далее приведены пользовательские интерфейсы, созданные с помощью Vue.js и MUSE-UI.

3.2.1 Страница входа

Данная страница содержит форму для ввода логина и пароля и ссылку на страницу регистрации (Рис. 6)

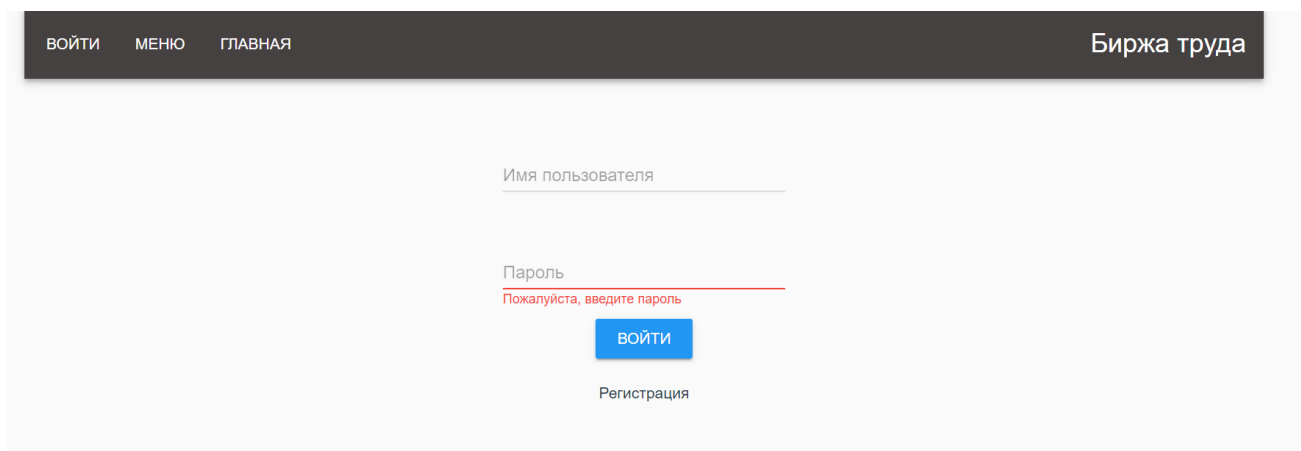


Рисунок 6 – Страница входа

При вводе неправильных данных выводится следующее всплывающее окно, изображенное на рисунке 7.

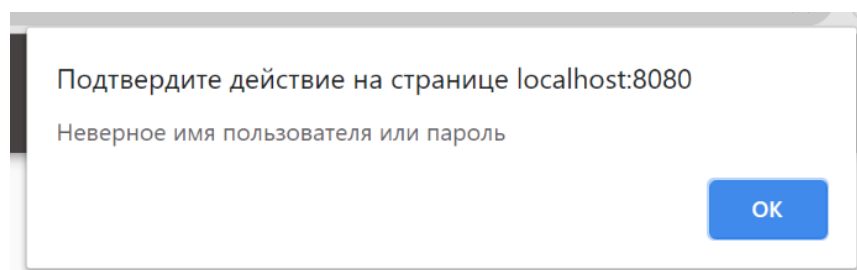


Рисунок 7 – Уведомление о неверно введенных данных на странице login

3.2.2 Страница регистрации

Страница содержит форму для задания логина и пароля. (Рис. 8)

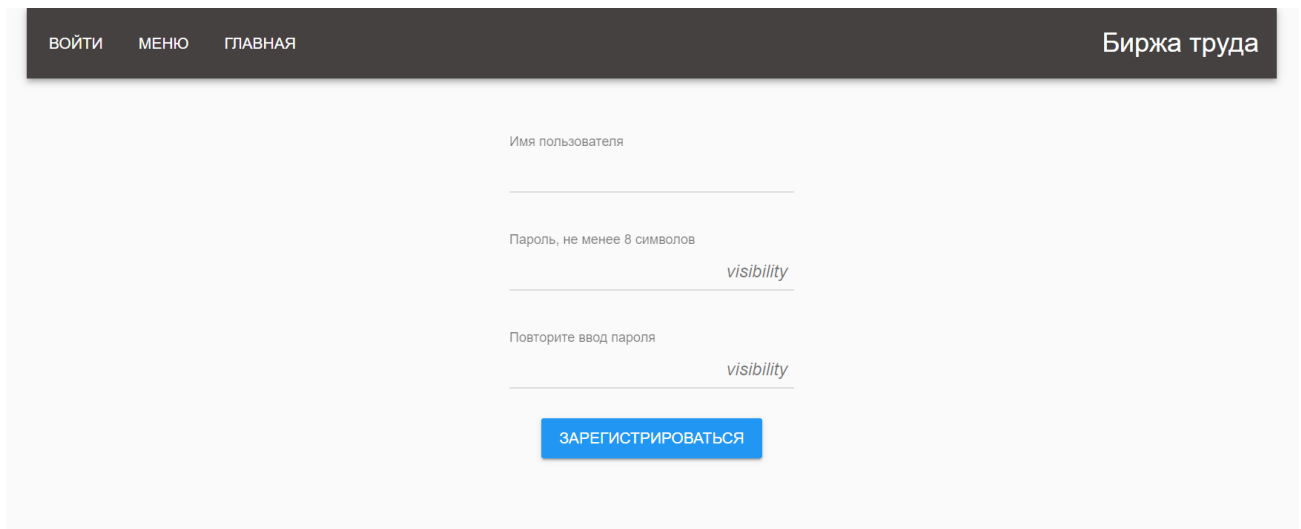


Рисунок 8 – Страница регистрации

При вводе данных, не соответствующих условиям, выводится следующее всплывающее окно (рис. 7).

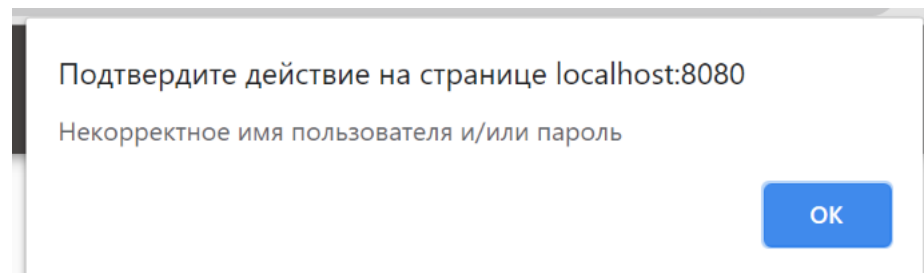


Рисунок 9 – Уведомление о некорректных данных

Если введенные пароли не совпадают, выводится следующее всплывающее окно, изображенное на рисунке 10.

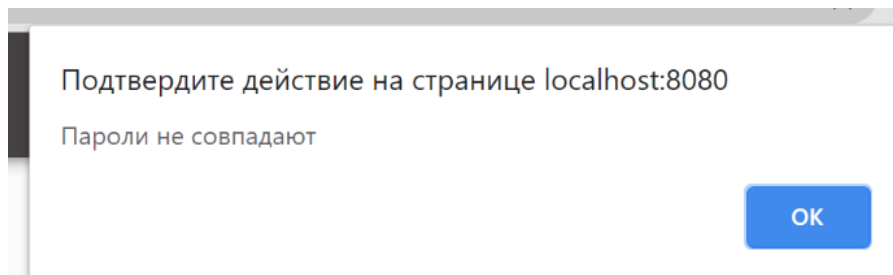


Рисунок 10 – Уведомление о несовпадающих паролях

3.2.3 Главная страница

Главная страница содержит ссылки на наиболее часто используемые страницы, такие как каталог вакансий и регистрация соискателя (рис. 11).

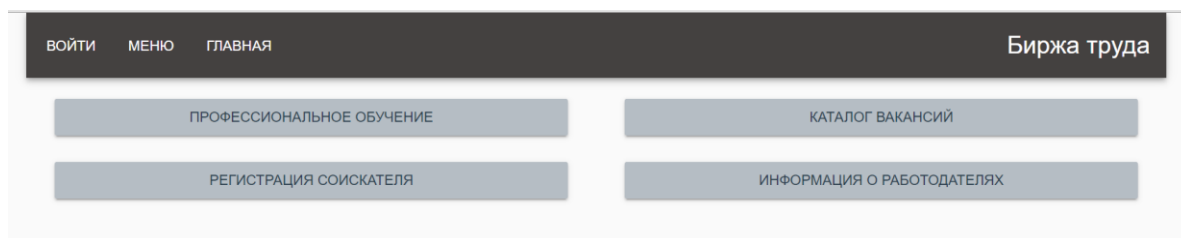


Рисунок 11 – Главная страница

Стоит отметить также, что при нажатии на кнопку “Меню” в верхней панели навигации, появляется выпадающий список ссылок, представленный на рисунке 12.

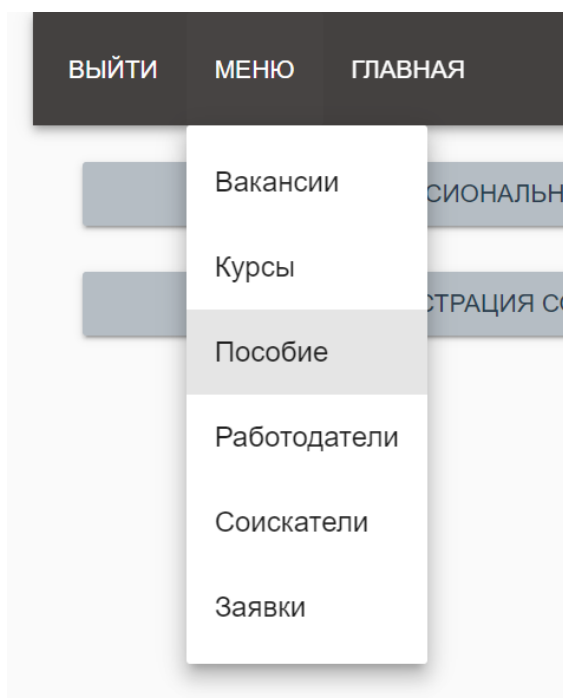


Рисунок 12 – Выпадающее меню

3.2.4 Страница вакансий

Страница вакансий содержит фильтры для более удобной навигации (рис. 13).

ВЫЙТИ

МЕНЮ

ГЛАВНАЯ

Биржа труда

Каталог вакансий

Профессия

▼

Образование

▼

Тип работы

▼

Статус вакансии

▼

Опыт работы в годах

Зарплата

От

От

До

До

ОТФИЛЬТРОВАТЬ

СБРОСИТЬ ФИЛЬТРЫ

Рисунок 13 – Фильтрация вакансий

Далее представлены сами вакансии и их описание, присутствует возможность добавить новую вакансию (рис. 14).

Логист	
Разряд	1
Образование	среднее общее образование
Опыт работы в годах	1
Тип работы	Стажировка
Работодатель	ООО "Дельта-Бизнес"
Зарплата	17000
Статус вакансии	Закрыта

Интерьерный дизайнер	
Разряд	0
Образование	среднее общее образование
Опыт работы в годах	5
Тип работы	Постоянная
Работодатель	ООО "Альфа-Сервис"
Зарплата	32000
Статус вакансии	Открыта

ДОБАВИТЬ НОВУЮ ВАКАНСИЮ

Рисунок 14 – Вывод вакансий

При переходе к конкретной вакансии появляется возможность подать заявку или изменить заработную плату и статус вакансии (рис. 15).

ВЫЙТИ

МЕНЮ

ГЛАВНАЯ

Биржа труда

Вакансия №3

Должность	Логист
Разряд	1
Образование	среднее общее образование
Опыт работы в годах	1
Тип работы	Стажировка
Работодатель	ООО "Дельта-Бизнес"
Зарплата	17000
Статус вакансии	<div>Закрыта</div> <div>Статус вакансии</div> <div></div>
Дата размещения вакансии	2020-10-14

ИЗМЕНИТЬ ИНФОРМАЦИЮ

ВНЕСТИ ИЗМЕНЕНИЯ

ПОДАТЬ ЗАЯВКУ

Рисунок 15 – Страница конкретной вакансии

3.2.5 Страница работодателей

Данная страница содержит список работодателей (рис. 16).

ВЫЙТИ

МЕНЮ

ГЛАВНАЯ

Биржа труда

Список работодателей

- ООО "Альфа-Сервис"
- ООО "Браво-Технологии"
- ООО "Чарли-Обслуживание"
- ООО "Дельта-Бизнес"

Рисунок 16 – Список работодателей

При переходе к конкретному работодателю возможно изменить его данные (рис. 17).

ВЫЙТИ

МЕНЮ

ГЛАВНАЯ

Биржа труда

Информация о работодателе

Название компании	ООО "Альфа-Сервис"
Руководитель компании	Иван А.
Адрес центрального офиса	Лиговский проспект, д. 45
ИНН	5764563456
Адрес электорной почты	fg34r@mail.net
Контактный телефон	798354353453

ВНЕСТИ ИЗМЕНЕНИЯ

Рисунок 17 – Данные работодателя

3.2.6 Страница соискателей

Данная страница содержит список соискателей (рис. 18).

ВЫЙТИ

МЕНЮ

ГЛАВНАЯ

Биржа труда

Список соискателей

- Иванов А. С.
- Петров Е. Е.
- Романов А. П.
- Ликин А. А.
- Борисов К. Р.
- Григорьев В. Ю.
- Пимов А. Р.

ЗАРЕГИСТРИРОВАТЬ НОВОГО СОИСКАТЕЛЯ

Рисунок 18 – Список соискателей

При переходе к конкретному соискателю возможно изменить его данные, а также просмотреть имеющиеся резюме, пособия и сертификаты (рис. 19).

ВЫЙТИ

МЕНЮ

ГЛАВНАЯ

Биржа труда

Информация о соискателе

ФИО	Иванов А. С.
Возраст	27
Адрес	ул. Димитрова, д. 3
Образование	высшее образование - бакалавриат
Трудовой стаж	6

ПРОСМОТРЕТЬ РЕЗЮМЕ

ПРОСМОТРЕТЬ ПОСОБИЯ

ПРОСМОТРЕТЬ СЕРТИФИКАТЫ КУРСОВ

Имеются резюме

Имеются пособия

Имеются сертификаты

ID	1
Профессия	Логист
Разряд	1
Зарплата	15000

ДОБАВИТЬ РЕЗЮМЕ

Сумма выплаты	5000
Дата начала выплаты пособия	2019-03-06
Дата окончания выплаты пособия	2020-03-06

ИЗМЕНИТЬ ИНФОРМАЦИЮ

Название курса	Основы логистики
----------------	------------------

Рисунок 19 – Данные соискателя

3.2.7 Страница пособий

Страница содержит список назначенных пособий, здесь же можно назначить новое пособие для выбранного соискателя (рис. 20).

ВЫЙТИ

МЕНЮ

ГЛАВНАЯ

Биржа труда

Выплачиваемые пособия

Пособие №1	
ФИО	Петров Е. Е.
Размер выплаты	5500
Дата начала выплаты	2020-03-06
Дата окончания выплаты	2020-11-06

Пособие №2	
ФИО	Ликин А. А.
Размер выплаты	7000
Дата начала выплаты	2020-05-09
Дата окончания выплаты	2020-11-09

Пособие №3	
ФИО	Иванов А. С.
Размер выплаты	5000
Дата начала выплаты	2019-03-06
Дата окончания выплаты	2020-03-06

Выберите соискателя

Размер пособия

Дата начала выплаты пособия ГГГГ-ММ-ДД

Дата окончания выплаты пособия ГГГГ-ММ-ДД

ДОБАВИТЬ

НАЗНАЧИТЬ ПОСОБИЕ

Рисунок 20 – Список пособий

3.2.8 Страница курсов

Страница содержит список доступных курсов, присутствует фильтр по вакансиям, возможность записать соискателя на курс. Для каждого курса можно посмотреть число записавшихся (рис. 21).

Профессиональное обучение

Курсы

Фильтр по профессиям

ОТФИЛЬТРОВАТЬ

Курс №1 Основы логистики

Профессия	Логист
Разряд	1
Продолжительность курса в месяцах	6
Курс проводит	Андреев И. С.

Курс №2 Повышение квалификации ССБ

Профессия	Программист
Разряд	5
Продолжительность курса в месяцах	1
Курс проводит	Осипов А. М.

ПОДРОБНАЯ ИНФОРМАЦИЯ О КУРСАХ

ЗАПИСАТЬ СОИСКАТЕЛЯ НА КУРС

КОЛИЧЕСТВО СОИСКАТЕЛЕЙ НА КАЖДОМ КУРСЕ

Рисунок 21 – Список курсов

3.3. Выводы

В результате разработки клиентской части web-приложения были созданы компоненты Vue.js, позволившие отобразить все необходимые страницы web-приложения в заданной предметной области, следуя установленным функциональным требованиям. Дизайн всех компонентов был реализован с использованием библиотеки Muse_UI.

ЗАКЛЮЧЕНИЕ

Созданное в рамках курсовой работы web-приложение для администрирования биржи труда соответствует сформулированным функциональным требованиям, на основании которых были созданы пользовательские интерфейсы:

- Авторизация и регистрация пользователей
- Интерфейс с информацией об имеющихся вакансиях
- Интерфейс с информацией о работодателях
- Интерфейс с информацией о соискателях

Сервис был реализован с применением следующих компонентов:

- PostgreSQL – для работы с базой данных
- Django REST Framework – фреймворк Python для создания web-приложений
- Vue.js – web-фреймворк JavaScript для создания пользовательских интерфейсов
- MUSE-UI – библиотека Vue.js для дизайна пользовательского интерфейса

СПИСОК ЛИТЕРАТУРЫ

1. Django documentation // The Web framework for perfectionists with deadlines [Электронный ресурс] URL: <https://www.djangoproject.com/> (дата обращения: 20.06.2020)
2. Home // Django REST framework [Электронный ресурс] URL: <https://www.django-rest-framework.org/> (дата обращения: 20.06.2020)
3. Introduction // Vue.js [Электронный ресурс] URL: <https://vuejs.org/v2/guide/> (дата обращения: 25.06.2020)
4. Muse UI [Электронный ресурс] URL: <https://muse-ui.org/#/en-US> (дата обращения: 25.06.2020)
5. WebDevBlog. Создание Django API используя Django Rest Framework [Электронный ресурс] URL: <https://webdevblog.ru/sozdanie-django-api-ispolzuya-django-rest-framework-apiview/> (дата обращения: 29.06.2020).

Приложение А. Модели

```
from django.db import models

class Education(models.Model):
    name = models.CharField("Образование", max_length=300)

    class Meta:
        verbose_name = 'Образование'

    def __str__(self):
        return self.name

class Field(models.Model):
    name = models.CharField("Название", max_length=300)

    class Meta:
        verbose_name = 'Сфера деятельности'
        verbose_name_plural = 'Сферы деятельности'

    def __str__(self):
        return self.name

class Profession(models.Model):
    name = models.CharField("Название", max_length=300)
    field = models.ForeignKey(Field, on_delete=models.CASCADE, related_name='professions')

    class Meta:
        verbose_name = 'Профессия'
        verbose_name_plural = 'Профессии'

    def __str__(self):
        return self.name

class Applicant(models.Model):
    fio = models.CharField("ФИО", max_length=300)
    age = models.IntegerField("Возраст")
    address = models.CharField("Адрес", max_length=300)
    work_experience = models.IntegerField("Опыт работы")
    education = models.ForeignKey(Education, on_delete=models.CASCADE)
```

Рисунок 1 – Содержимое файла models.py (часть 1)

```

class Resume(models.Model):
    applicant = models.ForeignKey(Applicant, on_delete=models.CASCADE, related_name='resumes')
    salary = models.IntegerField("Зарплата")
    rank = models.CharField("Разряд", max_length=300)
    profession = models.ForeignKey(Profession, on_delete=models.CASCADE)

    class Meta:
        verbose_name = 'Резюме'

class Course(models.Model):
    name = models.CharField("Название", max_length=300)
    author = models.CharField("Автор курса", max_length=300)
    lasting = models.IntegerField("Продолжительность")
    profession = models.ForeignKey(Profession, on_delete=models.CASCADE)
    rank = models.CharField("Разряд", max_length=300)

    class Meta:
        verbose_name = 'Курс'
        verbose_name_plural = 'Курсы'

    def __str__(self):
        return self.name

class CourseCertificate(models.Model):
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    applicant = models.ForeignKey(Applicant, on_delete=models.CASCADE, related_name='courses_certificates')

    class Meta:
        verbose_name = 'Сертификат курса'
        verbose_name_plural = 'Сертификаты курсов'

class Gratuity(models.Model):
    applicant = models.ForeignKey(Applicant, on_delete=models.CASCADE, related_name='gratuities')
    salary = models.IntegerField("Размер последней зарплаты")
    date_start = models.DateField("Дата начала выплаты")
    date_end = models.DateField("Дата окончания выплаты")

    class Meta:
        verbose_name = 'Пособие'
        verbose_name_plural = 'Пособия'

```

Рисунок 2 – Содержимое файла models.py (часть 2)

```

class Employer(models.Model):
    name = models.CharField("Название", max_length=300)
    contact = models.CharField("Контактное лицо", max_length=300)
    address = models.CharField("Адрес", max_length=300)
    inn = models.CharField("ИНН", max_length=10)
    email = models.EmailField()
    phone = models.CharField("Контактный телефон", max_length=12)

    class Meta:
        verbose_name = 'Работодатель'
        verbose_name_plural = 'Работодатели'

    def __str__(self):
        return self.name

class Vacancy(models.Model):
    status_type = models.TextChoices('status_type', 'Открыта Закрыта')
    status = models.CharField("Статус", blank=True, choices=status_type.choices, max_length=20)
    education = models.ForeignKey(Education, on_delete=models.CASCADE)
    date = models.DateField("Дата регистрации вакансии")
    salary = models.IntegerField("Зарплата")
    types = models.TextChoices('types', 'Стажировка Постоянная Временная Сезонная Дистанционная')
    type = models.CharField("Характер работы", blank=True, choices=types.choices, max_length=20)
    work_experience = models.IntegerField("Опыт работы")
    rank = models.CharField("Разряд", max_length=300)
    employer = models.ForeignKey(Employer, on_delete=models.CASCADE)
    profession = models.ForeignKey(Profession, on_delete=models.CASCADE)

    class Meta:
        verbose_name = 'Вакансия'
        verbose_name_plural = 'Вакансии'

class Application(models.Model):
    status_types = models.TextChoices('status_types', 'Одобрена В процессе Отклонена')
    status = models.CharField("Статус", blank=True, choices=status_types.choices, max_length=20)
    resume = models.ForeignKey(Resume, on_delete=models.CASCADE, related_name='applications')
    vacancy = models.ForeignKey(Vacancy, on_delete=models.CASCADE, related_name='applications')
    date = models.DateField("Дата подачи")

    class Meta:
        verbose_name = 'Заявка'
        verbose_name_plural = 'Заявки'

```

Рисунок 3 – Содержимое файла models.py (часть 3)