

*Ефанов К.А.*, студент Инжинирингового колледжа

НИУ «БелГУ» Россия, г. Белгород

*Efanov K.A.*, Engineering College student

NRU "BelSU" Russia, Belgorod

*Гончаров Д.В.*, Преподаватель СПО Инжинирингового колледжа

НИУ «БелГУ» Россия, г. Белгород

*Goncharov D.V.*, Lecturer of STR of Engineering College

NRU "BelSU" Russia, Belgorod

*Игнатенко Н. В.*, студент 2-го курса магистратуры

Института инженерных и цифровых технологий

НИУ «БелГУ» Россия, г. Белгород

*Ignatenko N. V.*, 2nd year graduate student

Institute of Engineering and Digital Technologies

NRU "BelGU" Russia, Belgorod

*Шонски В. Н.*,

Аспирант кафедры прикладной информатики и информационных технологий

НИУ «БелГУ» Россия, г. Белгород

*Shopsky V. N.*,

Postgraduate student of the Department of Applied Informatics and Information

Technologies

NRU "BelGU" Russia, Belgorod

## **РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОИСКА И**

## **ПОДБОРА ПРОФЕССИЙ ПО НАПРАВЛЕНИЯМ**

## **ПОДГОТОВКИ/СПЕЦИАЛЬНОСТЯМ**

## **DEVELOPMENT OF INFORMATION SYSTEM FOR SEARCH AND SELECTION OF PROFESSIONS IN TRAINING AREAS / SPECIALTIES**

**Аннотация:** в данной статье будет описан процесс разработки информационной системы подбора профессий для студентов по направлениям подготовки или специальности.

**Ключевые слова:** информационная система, подбор профессий, компания.

**Abstract:** this article will describe the process of developing an information system for the selection of professions for students in the areas of training or specialty.

**Keywords:** information system, selection of professions, company.

Профориентационная работа образовательных организаций ориентирована на помощь всем студентам в выборе интересной и увлекательной профессии, учитывая персональные особенности их личности. Данная ИС позволит учащимся значительно проще подбирать себе профессию и рабочее место по своим специальностям.

Сущность профориентации как общественной задачи выражается в необходимости преодоления противоречия между объективно имеющимися потребностями общества в сбалансированной структуре кадров и неадекватно этому сложившимися субъективными профессиональными устремлениями молодежи, то есть, по своему назначению, система профориентации должна оказать значительное воздействие на рациональное распределение трудовых ресурсов, выбор жизненного пути молодежью, адаптацию ее к профессии.

В настоящее время, не редко происходит так, что работник в самом начале своей карьеры, сталкивается с проблемой поиска вакансий. Чаще всего это приводит к растерянности, мешающему работнику подниматься вверх по карьерной лестнице.

Подготовка к выбору профессии важна еще и потому, что она является неотъемлемой частью всестороннего и гармоничного развития личности, и ее следует рассматривать в единстве и взаимодействии с нравственным, трудовым, интеллектуальным, политическим, эстетическим и физическим совершенствованием личности, то есть со всей системой учебно-воспитательного процесса. Таким образом, можно сделать вывод о том, что профориентация является важным моментом как в развитии каждого

человека, так и в функционировании общества в целом.

Перед началом процесса разработки необходимо выделить и рассмотреть все объекты данной системы. Основными объектами являются работодатели, студенты и вакансии. Данная работа будет содержать 3 сущности: «Факультет», «Вакансия» и «Студент».

Все сущности реализованы в виде специальных моделей, которые являются «отражением» базы данных в программном коде.

Модель «Факультет» представлена ниже (Рис. 1).

```
class Faculty(models.Model):
    facultyName = models.CharField(max_length=150, db_index=True, verbose_name='Наименование факультета')

    def __str__(self):
        return self.facultyName

class Meta:
    verbose_name = 'Факультет'
    verbose_name_plural = 'Факультеты'
    ordering = ['facultyName']
```

Рис. 1. Модель «Факультет»

Данный класс имеет лишь 1 атрибут – facultyName, отражающий название факультета. Он предназначен для логической связи моделей «Вакансия» и «Студент».

Следующая модель – «Вакансия». Она отражает представление об имеющийся в компании вакансии в программном коде и будущей базе данных (Рис. 2).

```
class Work(models.Model):
    faculty = models.ForeignKey('Faculty', on_delete=models.PROTECT, null=True, verbose_name='Факультет')
    job = models.CharField(max_length=150, verbose_name='Должность')
    company = models.CharField(max_length=150, verbose_name='Название компании')
    salary = models.CharField(max_length=20, verbose_name='Зарплата')
    description = models.TextField(blank=True, verbose_name='Описание')

    def __str__(self):
        return self.job

class Meta:
    verbose_name = 'Вакансия'
    verbose_name_plural = 'Вакансии'
    ordering = ['job']
```

Рис. 2. Модель «Вакансия»

Атрибут `faculty` хранит в себе идентификаторы факультетов из модели «Факультет». В поле `job` записывается наименование должности. Атрибуты `company` и `salary` служат для отображения названия компании и заработной платы соответственно. А в переменную `description` записывается описание вакансии.

Последней моделью является модель «Студент» (Рис. 3).

```
class Student(models.Model):
    faculty = models.ForeignKey('Faculty', on_delete=models.PROTECT, null=True,
                               verbose_name='Факультет')
    name = models.CharField(max_length=150, verbose_name='Имя')
    description = models.TextField(blank=True, verbose_name='Навыки')

    def __str__(self):
        return self.name

    class Meta:
        verbose_name = 'Студент'
        verbose_name_plural = 'Студенты'
        ordering = ['name']
```

Рис. 3. Модель «Студент»

Этот класс хранит в себе идентификатор выбранного факультета, ФИО студента и описание его навыков в соответствующих полях `faculty`, `name` и `description`.

После создания моделей была сгенерирована база данных формата `sqlite3`. Данный формат баз данных можно открывать в специализированном СУБД `SQLite Studio`. Он предназначен для небольших проектов с низкой нагрузкой на базу данных из-за своего относительно невысокого быстродействия.

Используя команду `makemigrations` была создана миграция базы данных в специальную директорию миграций. А командой `migrate` данная команда была преобразована в базу данных `sqlite3` (Рис. 4).

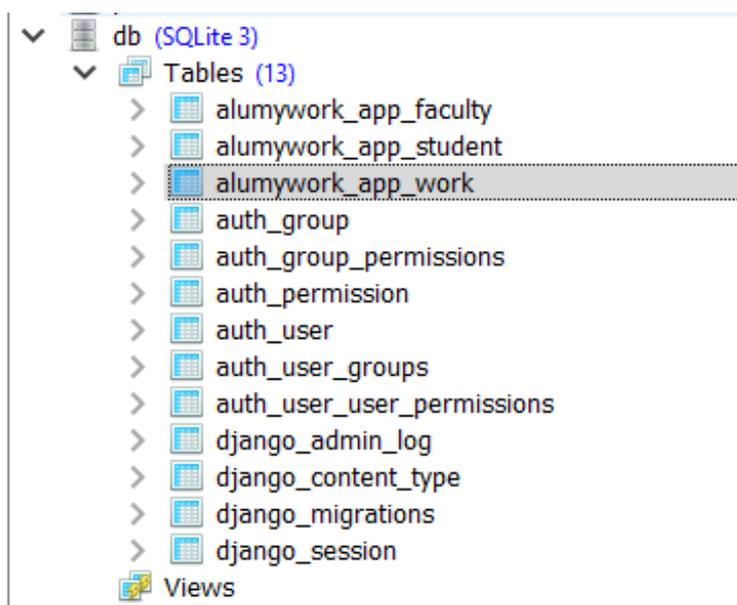


Рис. 4. Модель «Студент»

Для работы с базой данных используется встроенная админ-панель фреймворка Django.

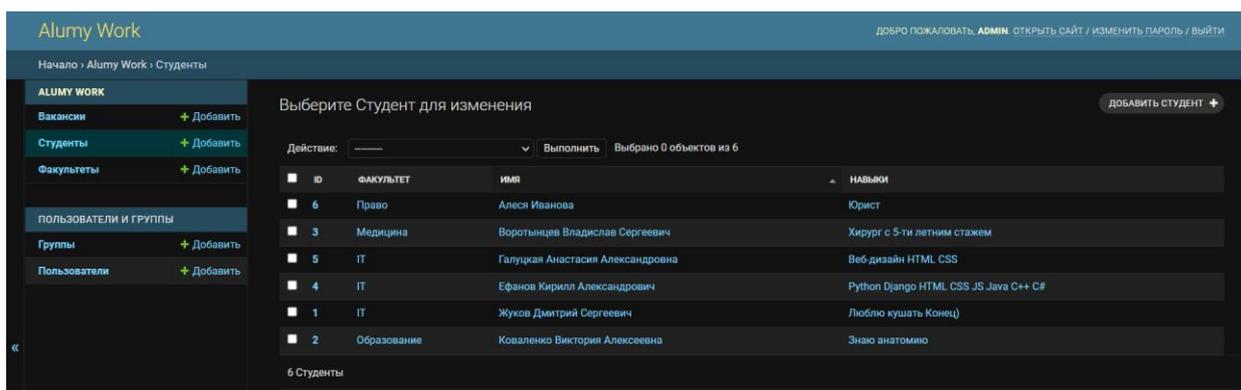


Рис. 5. Панель администрирования

Приложение Django admin может использовать модели для автоматического создания части сайта, предназначенной для создания, просмотра, обновления и удаления записей. Проект Django рекомендует это приложение только для управления внутренними данными, так как модельно-ориентированный подход не обязательно является наилучшим интерфейсом для всех пользователей и раскрывает много лишних подробностей о моделях.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Меле, А. Django 2 в примерах. Создавайте мощные и надежные веб-приложения на Python с нуля / А. Меле. – Москва : Юрайт ; Екатеринбург :

Изд-во ДМК Пресс. ун-та, 2019. – 578 с. – (Профессиональное образование).  
– ISBN 978-5-97060-746-6. – Текст : непосредственный.

2. **Тузовский, А. Ф.** Проектирование и разработка web-приложений : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2022. — 218 с. — (Высшее образование). — ISBN 978-5-534-00515-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/490128>.

3. **Форсье, Д.** Django. Разработка веб-приложений на Python / Д. Форсье. – Москва : Юрайт ; Екатеринбург : Изд-во Символ. ун-та, 2020. – 456 с. – (Профессиональное образование). – ISBN 978-5-93286-167-7. – Текст : непосредственный.