**Содержание**

[Введение 4](#_Toc159061763)

[1 Техническое задание на разработку информационной системы 5](#_Toc159061764)

[2 Проектирование схемы данных 6](#_Toc159061765)

[3 Создание структуры информационной системы 8](#_Toc159061766)

[4 Создание моделей данных 10](#_Toc159061767)

[5 Аутентификация и авторизация пользователей 12](#_Toc159061768)

[6 Настройка административной панели 14](#_Toc159061769)

[7 Создание пользовательского интерфейса 15](#_Toc159061770)

[8 Разработка функционала пользователей 19](#_Toc159061771)

[Заключение 24](#_Toc159061772)

[Список используемой литературы 25](#_Toc159061773)

# Введение

Учебная практика является составляющей частью профессионального модуля ПМ.05 Проектирование и разработка информационных систем, реализуется образовательным учреждением концентрированно в течение 216 часов на четвертом курсе.

Целями учебной практики являются:

* закрепление теоретических знаний, полученных при изучении базовых дисциплин;
* развитие и накопление специальных навыков, изучение и участие в разработке организационно-методических и нормативных документов для решения отдельных задачах по месту прохождения практики;
* усвоение приемов, методов, способов обработки и представления информации;
* приобретение практических навыков в будущей профессиональной деятельности или в отдельных ее разделах.

Задачи учебной практики:

* закрепить знания и умения, приобретаемые в результате освоения теоретических курсов;
* выработать практические навыки и способствовать комплексному формированию общих и профессиональных компетенций, обучающихся в рамках профессионального модуля;
* приобрести навык выполнения задания в соответствии с комплектом оценочной документации демонстрационного экзамена.

# Техническое задание на разработку информационной системы

Онлайн-платформа для Волонтерского движения представляет собой веб-ресурс, который объединяет волонтеров, организаторов и администраторов для осуществления добровольных инициатив, и мероприятий.

Платформа представляет возможность волонтерам участвовать в различных мероприятиях, оставлять заявки на участие, обмениваться мнениями и комментариями, а также участвовать в обсуждениях.

На платформе Волонтерского движения необходимо реализовать следующие

ключевые функции:

* Создание и управление мероприятиями;
* Регистрация волонтеров на мероприятия;
* Возможность комментирования новостей и обсуждения инициатив;
* Различные роли пользователей: гость, волонтер, организатор, администратор.

Функционал для каждой из ролей выглядит следующим образом:

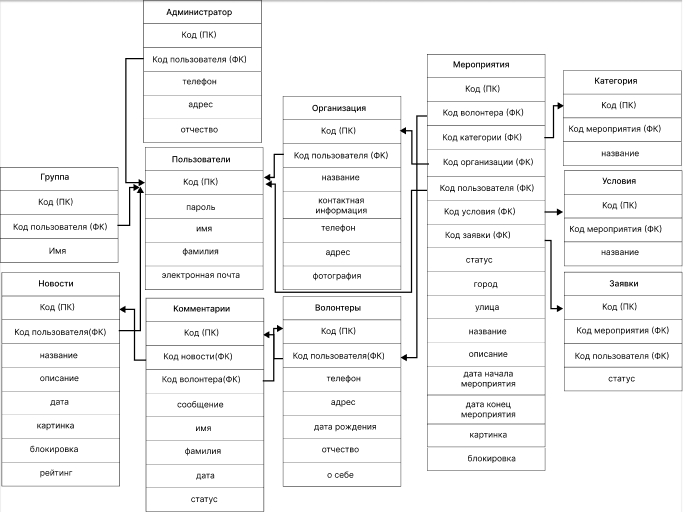
* Гость: регистрация/авторизация, использование основных функций сайта, просмотр информации о мероприятиях.
* Волонтер: оставление заявок на участие в мероприятиях, комментирование новостей, изменение данных профиля.
* Организатор: создание и управление мероприятиями, управление заявками волонтеров, изменение статусов заявок.
* Администратор: доступ к административной панели для управления контентом, мероприятиями, пользователями, модерация обсуждений и управление данными.

# Проектирование схемы данных

База данных — это структурированное хранилще информации, организованное с помощью специальных программных средств, которое позволяет эффективно хранить, управлять и извлекать данные.

Схема данных – это описание структуры базы данных, которое определяет, какие данные будут храниться, каким образом они будут связаны между собой и как к ним можно будет обращаться.

Спроектированная схема данных состоит из 12 сущностей, которая представлена на рисунке 1.

****Рисунок 1 – Схема БД

В ходе учебной практики база данных была расширена, и теперь содержит в общей сложности 12 таблиц. Среди новых: комментарии, условия, направление, местоположение. Ниже приведены атрибуты этих таблиц.

Атрибуты комментарий на странице новостей представлены в таблице 1.

Таблица 1 – Комментарии

|  |  |  |
| --- | --- | --- |
| Наименование атрибута на английском языке | Наименование атрибута на русском языке | Тип атрибута (размер атрибута) |
| id | Код комменатрий | Число |
| id\_news | Код новости | Число |
| Id\_volunteer | Код волонтера | Число |
| message | Сообщение | Строка(255) |
| status | Статус | Строка(255) |

Атрибуты условия мероприятия представлены в таблице 2.

Таблица 2 – Условия

|  |  |  |
| --- | --- | --- |
| Наименование атрибута на английском языке | Наименование атрибута на русском языке | Тип атрибута (размер атрибута) |
| id | Код условия | Число |
| name | Название | Строка (255) |

Атрибуты направлений мероприятия представлены в таблице 3.

Таблица 3 – Направление

|  |  |  |
| --- | --- | --- |
| Наименование атрибута на английском языке | Наименование атрибута на русском языке | Тип атрибута (размер атрибута) |
| id | Код направление | Число |
| name | Название | Строка (255) |

Атрибуты местоположений мероприятия представлены в таблице 4.

Таблица 4 – Местоположение

|  |  |  |
| --- | --- | --- |
| Наименование атрибута на английском языке | Наименование атрибута на русском языке | Тип атрибута (размер атрибута) |
| id | Код местоположение | Число |
| name | Название | Строка (255) |

# Создание структуры информационной системы

Для создания структуры информационной системы требуется выполнить три шага. Начнем с первого – использовать команду «django-admin startproject volunteers» для создания папки проекта. После этого необходимо перейти в созданную папку проекта с помощью команды «cd volunteers». Завершающим этапом будет выполнение команды «python manage.py startapp Volunteers» для создания приложение Volunteers.

На рисунке 2 представлена структура проекта ИС, каждая папка хранит в себе файлы проекта.

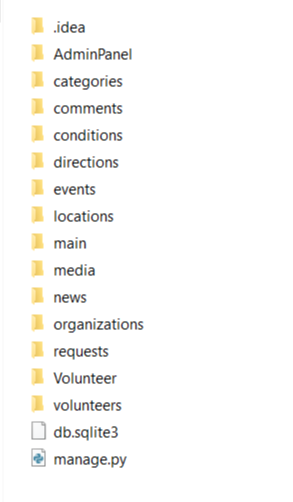


Рисунок 2 – Папка проекта

Папка «comments» включает в себя приложение, отвечающее за комментарии пользователей к новостям. Папка «events» содержит приложение, которое управляет мероприятиями, проводимыми на сайте. Папка «volunteers» хранит информацию о волонтерах, их заданиях и контактных данных для связи. Папка «media» служит для хранения изображений, которые используются на сайте. Файл «db.slite3» представляет собой базу данных проекта. Файл «manage.py» является утилитой командной строки для взаимодействия с проектом

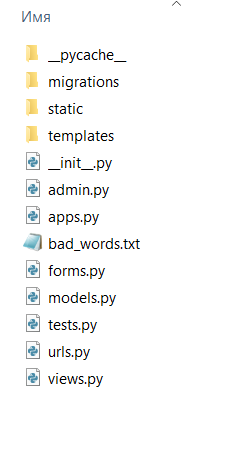


Рисунок 3 – Папка приложения «volunteers»

Файловая структура приложения:

В папке «migrations» располагаются файлы миграций, которые содержат необходимые инструкции для работы с базой данных. В папке «static» находятся изображения, используемые для оформления дизайна сайта. В каталоге «templates» расположены HTML-страницы. Файл «admin.py» отвечает за отображение таблиц в административной панели Django. Создание форм происходит в файле «forms.py». В файле «models.py» определяются структуры таблиц базы данных. Файл «urls.py» устанавливает соответствие между URL-адресами и представлениями, которые содержатся в файле «views.py».

# Создание моделей данных

Модель данных Django – это способ определения структуры данных для работы с базой данных в приложении Django. Модель данных представляет собой класс Python, который описывает поля и их типы данных, которые будут храниться в базе данных.

Модель данных используется для создания таблиц в базе данных, а также для управления данными в этих таблицах. Она позволяет определять отношения между различными сущностями и задавать правила валидации данных.

В листинге 1 представлена модель местоположения.

Листинг 1 – Модель местоположения

from django.db import models  
from django.contrib.auth.models import User  
class Location(models.Model):  
 user = models.ForeignKey(User, on\_delete=models.CASCADE)  
 city = models.CharField(max\_length=100)  
 street = models.CharField(max\_length=100)  
 house\_number = models.CharField(max\_length=10)  
 apartment\_number = models.CharField(max\_length=10)

def \_\_str\_\_(self):  
 return f"г {self.city}, ул {self.street}, д {self.house\_number}, кв {self.apartment\_number}"  
  
 class Meta:  
 verbose\_name = 'Местоположение'  
 verbose\_name\_plural = 'Местоположения'

В листинге 2 представлена модель новости.

Листинг 2 – Модель новости

class News(models.Model):  
 user = models.ForeignKey(User , on\_delete=models.CASCADE)  
 title = models.CharField('Название' , max\_length=255)  
 description = models.TextField('Описание')

date = models.DateTimeField('Дата публикации' , auto\_now\_add=True)  
 img = models.ImageField(upload\_to='main/img')

blocked = models.BooleanField('Заблокирована' , default=False)  
 rating = models.IntegerField('Рейтинг' , default=0)  
 def \_\_str\_\_(self):  
 return self.title  
 class Meta:  
 verbose\_name = 'Новость'  
 verbose\_name\_plural = 'Новости'

В листинге 3 представлена модель направления, который позволяет создать направление для мероприятия.

Листинг 3 – Модель направление

class Direction(models.Model):  
 name = models.CharField(max\_length=255)  
 def \_\_str\_\_(self):  
 return self.name  
 class Meta:  
 verbose\_name = 'Направления'  
 verbose\_name\_plural = 'Направление'

В листинге 5 представлена модель условия, который позволяет создать условия для мероприятия.

Листинг 5 – Модель условия

class Condition(models.Model):  
 name = models.CharField('Название', max\_length=255)  
 def \_\_str\_\_(self):  
 return self.name  
 class Meta:  
 verbose\_name = 'Условия'  
 verbose\_name\_plural = 'Условия'

# Аутентификация и авторизация пользователей

В области безопасности и управления доступом, особенно при работе с веб-приложениями, существуют два важных понятия: регистрация и авторизация.

Аутентификация в Django - это процесс проверки подлинности пользователей, который позволяет системе удостовериться в их личности. Django предоставляет встроенные инструменты для реализации аутентификации, включая встроенные модели пользователей, формы и представления.

Авторизация, в свою очередь, является процессом предоставления пользователю прав доступа к определенным ресурсам или функциональности в системе. После успешной аутентификации система определяет, какие действия и данные доступны данному пользователю.

На рисунке 4 создана форма для авторизации пользователя.



Рисунок 4 – Авторизация

В листинге 6 представлена авторизация пользователя.

Листинг 6 - Авторизация пользователя

{% extends 'main/layout.html' %}  
{% load static %}  
{% block title %}

Главная страница

{% endblock %}  
{% block content %}

<div class="flex flex-col lg:flex-row

my-5 container mx-auto text-center ">  
 <!-- Left Pane -->  
 <div class="lg:w-1/2 flex

justify-center items-center lg:justify-center

lg:items-center">  
 <div class="max-w-md w-full">  
 <h2 class="text-2xl font-bold ">

Авторизация

</h2>  
 <form method="POST"

enctype="multipart/form-data" class="mt-4">  
 {% csrf\_token %}  
 <input type="hidden" name="next" value="{{ next }}" />  
 <div class="form-error">{{ form.non\_field\_errors }}</div>  
 {% for f in form %}  
 <div class="form-field py-2">  
 <p>  
 <label class="form-label" for="{{ f.id\_for\_label }}">{{ f.label }}</label>  
 {{ f }}  
 </p>  
 <div class="form-error text-red-500

py-2">

{{ f.errors }}

</div>  
 </div>  
 {% endfor %}  
 <button type="submit"  
 class="w-full my-5 bg-indigo-600

text-white py-2 rounded-md hover:bg-red-400 focus:outline-none focus:ring-2 focus:ring-offset-2 transition-colors duration-300">Регистрация</button>  
 <p><a href="{% url 'volunteers:password\_reset' %}">Забыли пароль?</a></p>  
 </form>  
 </div>  
 </div>  
 <div class="hidden lg:flex justify-center

text-center flex-1 bg-white text-black px-5 py-5">  
 <div class="text-center">  
 <img src="{% static 'main/img/image 13.jpg' %}" alt=""  
 class="rounded-lg max-h-full w-[800px] h-[450px]">  
 </div>  
 </div>  
</div>

# Настройка административной панели

Административная панель в Django – это встроенный инструмент, предоставляющий удобный интерфейс для управления административными задачами и данными веб-приложения. Позволяет администраторам эффективно управлять пользователями, группами, моделями данных и другими аспектами приложения без необходимости вмешательства в код.

Администратор имеет доступ к административной панели, в котором есть следующий функционал:

* Добавление/изменение/удаление новостей;
* Модерация комментарий (принятие/отклонение);
* Модерация мероприятий (принятие/отклонение);
* Добавление/изменение/удаление направлений;
* Добавление/изменение/удаление условий;
* Добавление/изменение/блокировка пользователей

В листинге 7 представлено подключение административной панели.

Листинг 7 - Подключение административной панели

from django.contrib import admin

В листинге 8 представлена настройка административной панели.

Листинг 8 - Настройки административной панели

from django.contrib import admin  
from django.utils.html import format html  
from .forms import VolunteerAdminForm  
class VolunteerAdmin(admin.ModelAdmin):  
 form = VolunteerAdminForm  
 list\_display = ('user', 'phone', 'address', 'date\_of\_birth', 'patronymic', 'about', 'display\_img')  
 def display\_img(self, obj):  
 return format\_html('<img src="{}" width="80" height="60" />', obj.photo.url)  
 display\_img.short\_description = ('photo')  
admin.site.register(Volunteer, VolunteerAdmin)

# Создание пользовательского интерфейса

В ходе учебной практики были разработаны новые страницы и внесены обновления на уже существующих. Среди новых страниц.

* редактирование данных пользователей
* модерация комментарий для новостей
* модерация созданных мероприятий.
* восстановление пароля.
* количество волонтеров по месяцам и годам

Обновление существующих страниц.

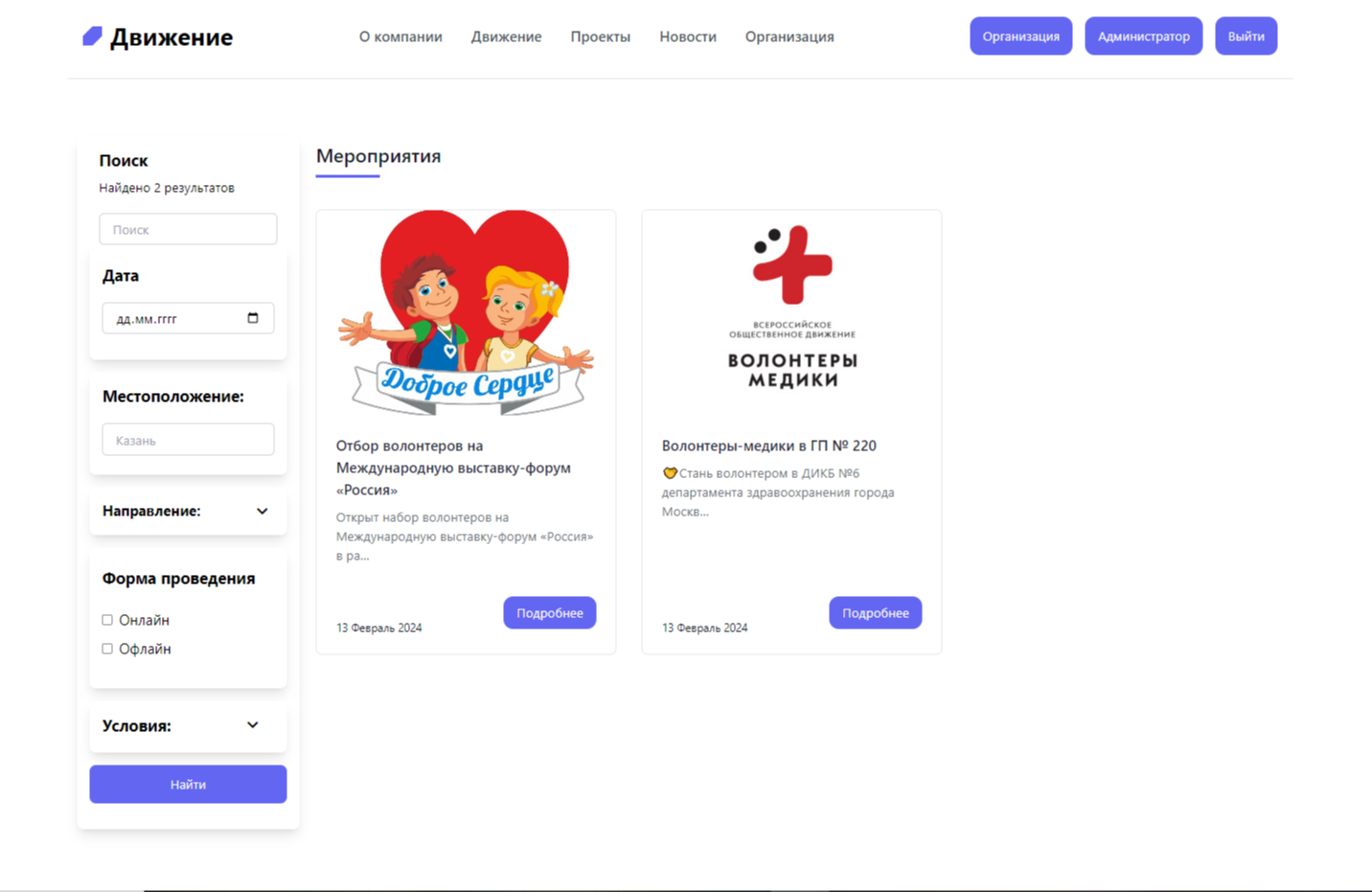
* На странице «Новости» добавлен блок с формой для комментария
* На странице «Новости» добавлена фильтрация новостей от новых к старым
* На странице «Регистрация» добавлена маска для телефона.
* На странице «Мероприятия» добавлен блок с сортировкой по направлению.

Для разработки дизайна был использован HTML и CSS.

HTML (Hypertext Markup Language) – это язык разметки, который используется для создания структуры веб-страниц. HTML определяет контент страницы, такой как текст, ссылки, изображения и другие элементы, и организует их для отображения в браузере.

CSS (Cascading Style Sheets) – это язык таблиц стилей, который используется для задания внешнего вида веб-страниц CSS позволяет указывать стили элементов, такие как цвет текст, шрифт, размер и расположение блоков на странице.

Ниже, на рисунке 5, представлен макеты проектов.

 Рисунок 5 – Проекты

В листинге 9 представлен код проектов.

Листинг 9 - Код проектов

<form class="mb-4" method="GET">  
 <div class="flex flex-col px-3 mt-5">  
 <input type="text"  
 class="h-10 px-4 border border-gray-300 rounded-md focus:outline-none focus:ring-indigo-500 focus:border-indigo-500"  
 placeholder="Поиск" name="q" value="{{ query }}">  
 </div>  
 <div class="bg-white rounded-lg shadow-lg p-4 mb-4">  
 <div class="mb-2 py-2">  
 <h3 class="font-bold text-xl mb-1">Дата</h3>  
 <div class="flex flex-col mt-5">  
 <input type="date" id="start\_date" name="start\_date"  
 class="h-10 px-4 border border-gray-300 rounded-md focus:outline-none focus:ring-indigo-500 focus:border-indigo-500"  
 value="{{ form.start\_date.value }}">  
 </div>  
 </div>  
 </div>  
 </div>  
 {% endfor %}  
 </div>  
 </div>  
 <button type="submit"  
 class='my-2 aria-label="dashboard" class="relative w-full text-center justify-center px-4 py-3 flex items-center space-x-4 rounded-lg text-white bg-gradient-to-r from-sky-600 to-cyan-400'>Найти</button>  
</form>

На рисунке 6, представлен макеты новостей.

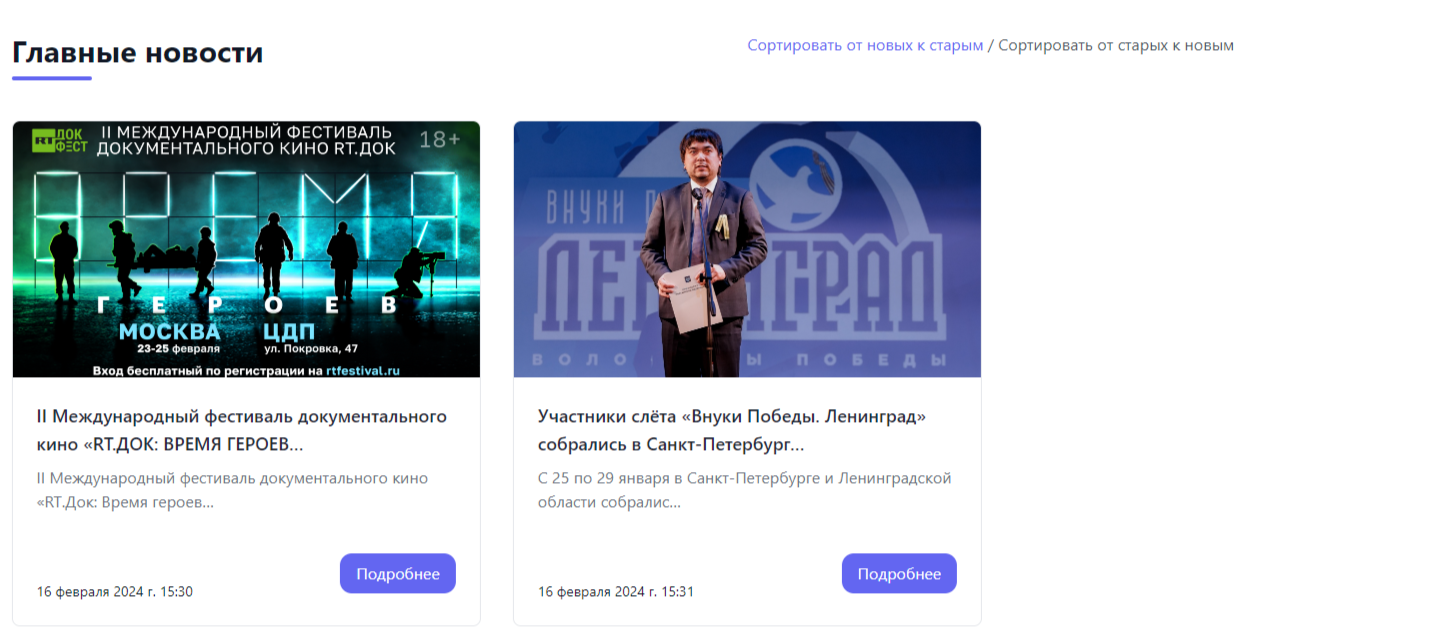


Рисунок 6 – Новости

В листинге 10 представлен код новостей.

Листинг 10 - Код новостей

<div class="container lg:px-5 py-5 lg:py-10 mx-auto">  
 <div class="flex flex-wrap w-full lg:mb-10">  
 <div class="lg:w-1/2 w-full mb-6 lg:mb-0">  
 <h1 class="sm:text-3xl text-2xl font-bold title-font mb-2 text-gray-900">Главные новости</h1>  
 <div class="h-1 w-20 bg-indigo-500 rounded"></div>  
 </div>  
 <div>  
 <a href="{% url 'news' %}?sort=newest"  
 class="{% if not sort\_param or sort\_param == 'newest' %}text-indigo-500{% else %}text-gray-600{% endif %}">Сортировать  
 от новых к старым</a>  
 /  
 <a href="{% url 'news' %}?sort=oldest"  
 class="{% if sort\_param == 'oldest' %}text-indigo-500{% else %}text-gray-600{% endif %}">Сортировать от старых  
 к новым</a>  
 </div>  
 </div>  
 <div class="grid gap-4 sm:grid-cols-2 md:gap-6 lg:grid-cols-2 xl:grid-cols-3 xl:gap-8">  
 {% for new in news %}  
 {% if not new.blocked %}  
 <div class="flex flex-col overflow-hidden rounded-lg border bg-white">  
 <a href="#" class="group relative block h-64 overflow-hidden md:h-64">  
 <img src="{{ new.img.url }}" loading="lazy" alt="{{ new.title }}"  
 class="absolute inset-0 h-full w-full object-cover object-center transition duration-200 group-hover:scale-110" />  
 </a>  
 <div class="flex flex-1 flex-col p-2 sm:p-6">  
 <h2 class="mb-2 text-lg font-semibold text-gray-800">  
 <a href="#" class="transition duration-100 hover:text-indigo-500 active:text-indigo-600 break-words">  
 {{ new.title|truncatechars:70 }}  
 </a>  
 </h2>  
 {% endif %}

# Разработка функционала пользователей

Функциональность для пользователей в веб-приложениях включает в себя разнообразные возможности и операции, призванные удовлетворить их потребности и обеспечить эффективное взаимодействие с приложением. Ассортимент этой функциональности может быть широким и зависеть от типа положения, его целей и уникальных особенностей.

В ходе учебной практики был разработан функционал: восстановления пароля, функционал поиска, функционал о количестве волонтеров по месяцам и годам.

На рисунке 7, представлен функционал для восстановления пароля.

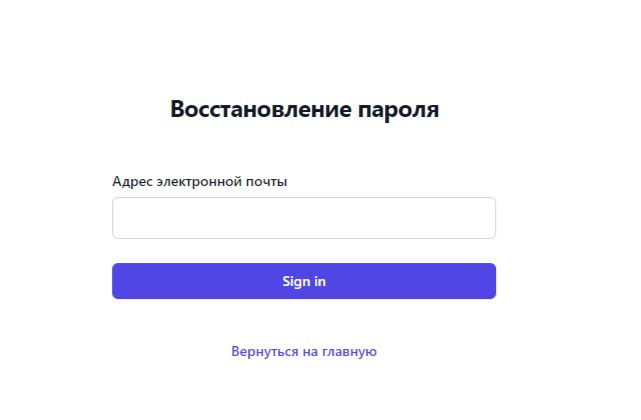


Рисунок 7 – Восстановление пароля

В листинге 11 представлен код восстановление пароля.

Листинг 11 - Код восстановление пароля

<div class="flex min-h-full flex-col justify-center px-6 py-12 lg:px-8">  
 <div class="sm:mx-auto sm:w-full sm:max-w-sm">  
 <img class="mx-auto h-10 w-auto" src="" alt="">  
 <h2 class="mt-10 text-center

text-2xl font-bold leading-9

tracking-tight text-gray-900">

Восстановление пароля

</h2>  
 </div>  
 <div class="mt-5 sm:mx-auto sm:w-full sm:max-w-sm">  
 <form class="space-y-6 my-form-class " method="post">  
  
 {% csrf\_token %}  
 {% for f in form %}  
 <div>  
 <label for="{{ f.id\_for\_label }}" class="block text-sm font-medium leading-6 text-gray-900">{{ f.label }}</label>  
 <div class='mt-1'>  
 <p class="border border-gray-300

rounded-md px-3 py-2 mt-1 focus:outline-none

focus:ring-indigo-500 focus:border-indigo-500">{{ f }}</p>  
 </div>  
 <div class="form-error">

{{ f.errors }}

</div>  
 </div>  
 {% endfor %}  
  
 <div>  
 <button type="submit" class="flex w-full justify-center rounded-md bg-indigo-600 px-3

py-1.5 text-sm font-semibold leading-6 text-white shadow-sm

hover:bg-indigo-500 focus-visible:outline focus-visible:outline-2 focus-visible:outline-offset-2 focus-visible:outline-indigo-600">Sign in</button>  
 </div>  
 </form>  
 <p class="mt-10 text-center text-sm text-gray-500">  
 <a href="#" class="font-semibold

leading-6 text-indigo-600 hover:text-indigo-500">Вернуться на главную</a>  
 </p>  
 </div>  
</div

Функционал поиск комментарий новостей с фильтрами представлен на рисунке 8.

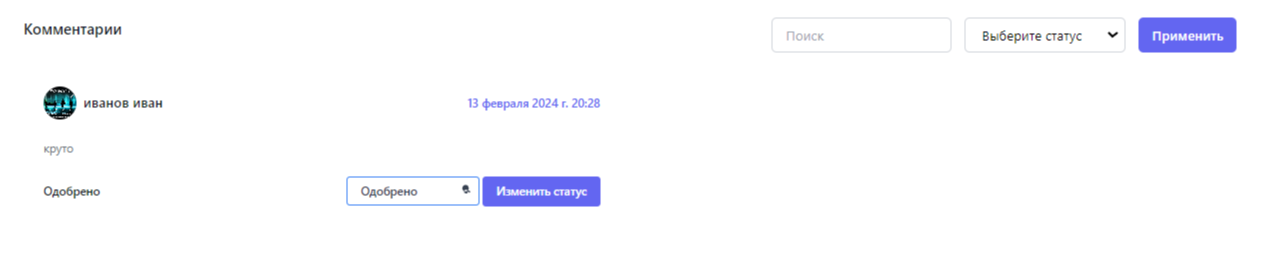


Рисунок 8 – Функционал поиска

В листинге 12 представлен код функционала поиска.

Листинг 12 - Код функционала поиска

<div class="my-5">  
 <div class="flex justify-between">  
 <h2 class="text-lg font-semibold text-gray-900">Комментарии</h2>  
 <div class="flex justify-end mb-4">  
 <input type="text" class="px-4 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-indigo-500 focus:border-indigo-500" placeholder="Поиск" />  
 <select class="ml-4 px-4 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-indigo-500 focus:border-indigo-500">  
 <option value="" selected disabled>Выберите статус</option>  
 <option value="0">На рассмотрении</option>  
 <option value="1">Одобрено</option>  
 <option value="2">Отклонено</option>  
 </select>  
 <button class="ml-4 px-4 py-2 bg-indigo-500 hover:bg-indigo-600 text-white font-semibold rounded-md">Применить</button>  
 </div>  
 </div>  
 </div>

На рисунке 9 представлен функционал для количество волонтеров по месяцам и годам.

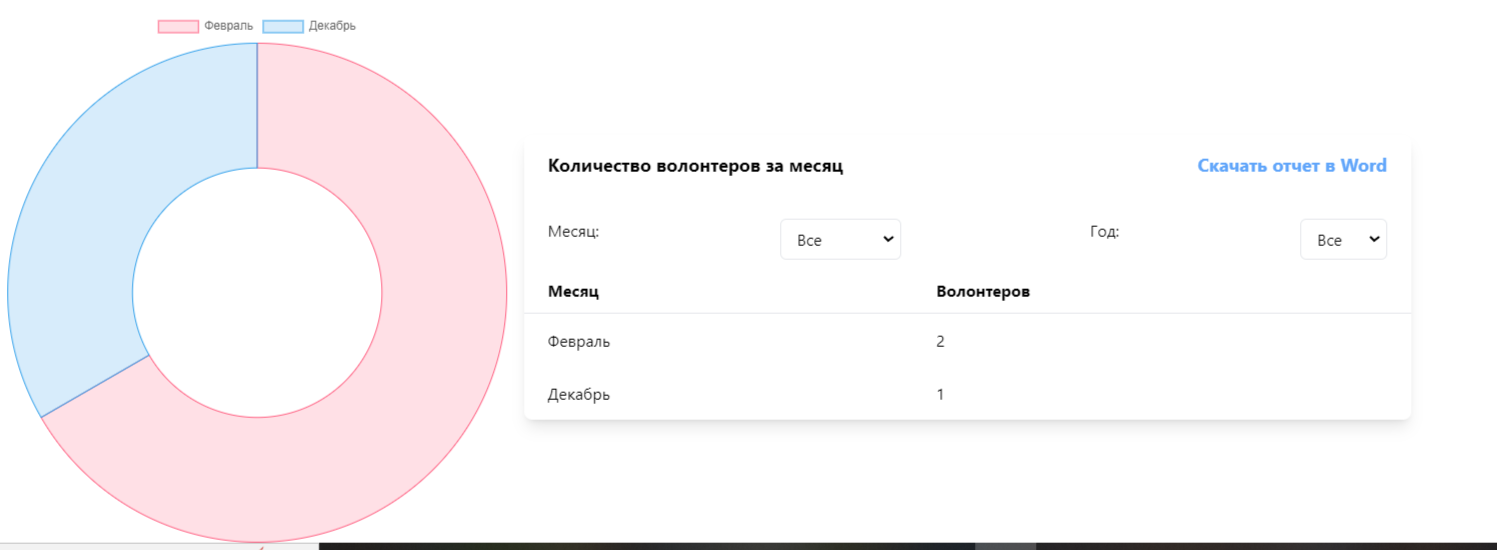


Рисунок 9 – Функционал количество волонтеров по месяцам и годам

В листинге 13 представлен код функционала количества волонтеров по месяцам и годам.

Листинг 13 - Код функционала количества волонтеров по месяцам и годам

<div class="flex items-center ">  
 <div style="width: 30%; ">  
 <canvas id="volunteerChart"></canvas>  
 </div>  
 <div style="width: 50%; ml-10">  
 <div class="w-full container mx-auto">  
 <div class="bg-white shadow-lg rounded-lg overflow-hidden">  
 <div class="px-6 py-4">  
 <div class="flex justify-between items-center mb-4">  
 <h3 class="font-bold text-lg text-blueGray-700">Количество волонтеров за месяц</h3>  
 <a href="{% url 'export\_word' %}" class="font-bold text-lg text-blue-400 hover:text-blue-600">Скачать отчет в Word</a>  
 </div>  
 </div>  
 <div class="block w-full overflow-x-auto">  
 <div class="flex justify-between px-6 py-2">  
 <label for="monthSelect" class="mr-2 ">Месяц:</label>  
 <select id="monthSelect" class="border border-blueGray-300 px-3 py-2 rounded-md">  
 <option value="">Все</option>  
 <option value="Январь">Январь</option>  
 </select>  
 <label for="yearSelect" class="mr-2 ml-4">Год:</label>  
 <select id="yearSelect" class="border border-blueGray-300 px-3 py-2 rounded-md">  
 <option value="">Все</option>  
 {% for year in years %}  
 <option value="{{ year }}">{{ year }}</option>  
 {% endfor %}  
 </select>  
 </div>  
 <table class="min-w-full table-auto">  
 <thead><tr>  
 <th class="px-6 py-3 text-md text-left font-bold bg-blueGray-100 text-blueGray-500 border-b border-blueGray-200">Месяц</th>  
 <th class="px-6 py-3 text-md text-left font-bold bg-blueGray-100 text-blueGray-500 border-b border-blueGray-200">Волонтеров</th>  
 </tr>  
 </thead>  
 <tbody>  
 {% for volunteer\_count in volunteer\_counts %}  
 <tr class="hover:bg-blue-100">  
 <td class="px-6 py-4 whitespace-nowrap ">{{ volunteer\_count.month }}</td>  
 <td class="px-6 py-4 whitespace-nowrap">{{ volunteer\_count.count }}</td>  
 </tr>  
 {% endfor %}  
 </tbody></table>  
 </div>  
 </div>

# Заключение

В период прохождения учебной практики мной приобретен практический опыт в управлении процессом разработки приложений с использованием инструментальных средств; обеспечении сбора данных для анализа использования и функционирования информационной системы; программировании в соответствии с требованиями технического задания; использовании критериев оценки качества и надежности функционирования информационной системы; применении методики тестирования разрабатываемых приложений; определении состава оборудования и программных средств разработки информационной системы; разработке документации по эксплуатации информационной системы; проведении оценки качества и экономической эффективности информационной системы в рамках своей компетенции; модификации отдельных модулей информационной системы.

Приобретены умения: осуществлять постановку задач по обработке информации; проводить анализ предметной области; осуществлять выбор модели и средства построения информационной системы и программных средств; использовать алгоритмы обработки информации для различных приложений; решать прикладные вопросы программирования и языка сценариев для создания программ; разрабатывать графический интерфейс приложения; создавать и управлять проектом по разработке приложения; проектировать и разрабатывать систему по заданным требованиям и спецификациям

И знания: основные виды и процедуры обработки информации, модели и методы решения задач обработки информации; основные платформы для создания, исполнения и управления информационной системой; основные процессы управления проектом разработки; основные модели построения информационных систем, их структуру, особенности и области применения; методы и средства проектирования, разработки и тестирования информационных систем; систему стандартизации, сертификации и систему обеспечения качества продукции.

# Список используемой литературы

1. Полищук, Ю. В. Базы данных и их безопасность : учебное пособие / Ю.В. Полищук, А.С. Боровский. — Москва : ИНФРА-М, 2022. — 210 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016151-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1864071> – Режим доступа: по подписке.

2. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие / Г.Н. Федорова. — Москва : КУРС : ИНФРА-М, 2022. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1858587> – Режим доступа: по подписке.

3. Ананьева, Т. Н. Стандартизация, сертификация и управление качеством программного обеспечения : учебное пособие / Т.Н. Ананьева, Н.Г. Новикова, Г.Н. Исаев. — Москва : ИНФРА-М, 2021. — 232 с. — (Среднее профессиональное образование). - ISBN 978-5-16-014887-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1413308> – Режим доступа: по подписке.

4. Гвоздева, Т. В. Проектирование информационных систем. Основы управления проектами. Лабораторный практикум / Т. В. Гвоздева, Б. А. Баллод. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 120 с. — ISBN 978-5-507-44958-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/250811> — Режим доступа: для авториз. пользователей.

5. Миронов, А. И. Тестирование и верификация программного обеспечения: Практикум : учебное пособие / А. И. Миронов, С. М. Трушин, А. А. Петренко. — Москва : РТУ МИРЭА, 2022. — 65 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/240095> — Режим доступа: для авториз. пользователей.

6. Меле А. М47 Django 2 в примерах / пер. с анг. Д. В. Плотниковой. – М.: ДМК Пресс, 2019. – 408 с.: ил.

7. Постолит А. В. П63 Python, Django и PyCharm для начинающи. -СПб.: БХ-Петербурr, 2021. -464 с.: ил. -(Д начинащих)