

СИСТЕМА СБОРА И АНАЛИЗА СВЕДЕНИЙ О ПРЕПОДАВАТЕЛЯХ

Участники:

Новрузов С.Р. СИИ-3

Орлов Н.Е. СИИ-3

Шереметьева А.А. РМП -5

Лазаренков Н.М. РМП -5

Ушаков Е.А. РМП -5

Научный руководитель:

доц. Конченков В.И.

Актуальность

Система сбора и анализа сведений о преподавателях предоставляет учебным заведениям возможность эффективно управлять данными о преподавателях, включая их квалификацию, академические достижения и прочие релевантные данные. Это позволяет учреждениям выстраивать более прозрачные и объективные процессы при найме и оценке преподавателей, а также оптимизировать распределение нагрузки и планирование учебных программ

Цель работы

Целью данного проекта является разработка приложения для помощи высшим учебным заведениям в работе по сбору и анализу сведений о преподавателях. Приложение включает в себя данные о преподавателях, их квалификацию, академические достижения и прочие релевантные данные

Основные функции

- Хранение данных (накопление архива сведений и предоставление данных для составления разнообразных отчетов)
- Сбор данных о показателях эффективности работы преподавателей (предоставление форм для ввода показателе)
- Анализ собранных данных (вычисление статистических показателей, различных метрик, сравнения результатов разных преподавателей)
- Создание отчетов
- Интегрируемость с другими информационными системами университета

Сценарии работы с приложением

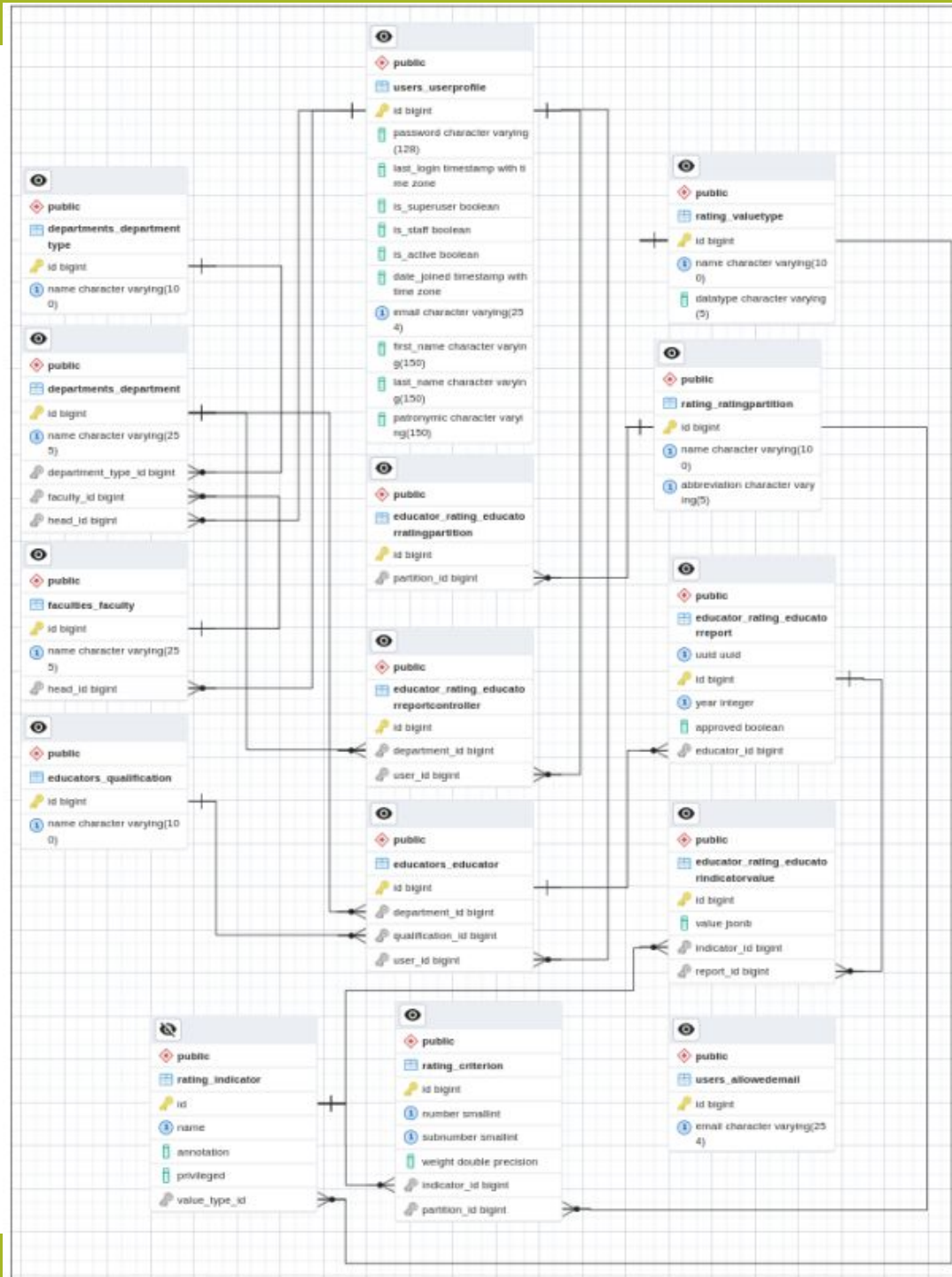
- 1. закрытая регистрация пользователей в системе;
- 2. просмотр преподавателем собственных анкет;
- 3. создание преподавателем новых анкет с внесением данных о его показателях;
- 4. просмотр проверяющим контролируемых им анкет;
- 5. утверждением проверяющим контролируемых им анкет.

Микросервисная архитектура

- Требование интегрируемости разработанной системы с другими информационными системами университета удобнее всего реализовать, используя микросервисную архитектуру:
- Сервис клиента – веб-приложение
- Сервис API
- Сервис базы данных

База данных

Здесь просто в столбик
Написать названия сущностей –
С 24-25



Экранные формы. Эскиз страницы авторизации

Авторизация

Почта
admin@example.com

Пароль

[Нет аккаунта?](#)

Экранные формы. Эскиз страницы просмотра списка анкет

Фамилия Имя Отчество СОЗДАТЬ АНКЕТУ ВЫЙТИ

Список анкет

Показывать: Свои Статус: Год: ФИО Преподавателя: Сортировка: ФИО

Анкета №_
Фамилия Имя Отчество
Год
Статус
открыть удалить

Анкета №_
Фамилия Имя Отчество
Год
Статус
открыть удалить

Анкета №_

Экранные формы. Эскиз модального окна выбора анкеты

ОТКРЫТЬ УДАЛИТЬ

Анкета №_
Фамилия Имя Отчество
Год
Статус

ОТКРЫТЬ УДАЛИТЬ

Создание анкеты

Преподаватель ▾

Год анкеты ▾

СОЗДАТЬ ОТМЕНИТЬ

Средства разработки серверной части

- Django
- Django REST Framework
- Django REST Framework SimpleJWT
- Djoser - для реализации модуля аутентификации
- Django CORS Headers
- Django Jazzmin – создание панели администрирования
- Psycopg2-binary +PostgreSQL
- Gdown – загрузка файлов на Google Drive, для предварительного заполнения БД
- drf-yasg – для работы Swagger и ReDoc (автоматическое создание документации по API, выполненного по стандартам OpenAPI)

Панель администрирования

The screenshot displays the 'Пользователи' (Users) management interface. On the left is a sidebar with a navigation menu including 'Админпанель', 'Пользователи', 'Разрешенные email', 'Преподаватели', 'Кафедры', 'Факультеты', 'Рейтинг', and 'Группы'. The main content area features a breadcrumb trail 'Пользователи > Начало > Пользователи > Пользователи', a 'Добавить пользователь' button, and search filters for 'статус персонала', 'статус суперпользователя', and 'активный'. A 'Найти' button is positioned to the right of these filters. Below the filters, a 'Выполнить' button is accompanied by the text 'Выбрано 0 объектов из 1'. A table lists users with columns for 'Email', 'Фамилия', 'Имя', and 'Отчество'. One user is listed: 'admin@example.com'. Below the table, it states '1 пользователь'. The footer contains the copyright notice 'Copyright © 2023 Volgograd State Technical University. Все права защищены.' and the version 'Jazmin Version 2.6.0'.

ВолгГТУ

admin@example.com

Админпанель

Пользователи

Разрешенные email

Преподаватели

Кафедры

Факультеты

Рейтинг

Группы

Пользователи

Начало > Пользователи > Пользователи

Добавить пользователь

статус персонала

статус суперпользователя

активный

Найти

Выполнить

Выбрано 0 объектов из 1

<input type="checkbox"/>	Email	Фамилия	Имя	Отчество
<input type="checkbox"/>	admin@example.com			

1 пользователь

Copyright © 2023 Volgograd State Technical University. Все права защищены.

Jazmin Version 2.6.0

Документация API – представление при помощи Swagger

Swagger
powered by Swagger

http://localhost:8000/swagger/?format=openapi Explore

University Rating System API ^{v1}

[Base URL: localhost:8000/]
<http://localhost:8000/swagger/?format=openapi>

API for performing operations on rating reports.

[Terms of service](#)
[Contact the developer](#)
[MIT License](#)

Schemes
HTTP

Django [admin@example.com](#) Django Logout Authorize

Filter by tag

allowed-emails

- GET /allowed-emails/ allowed-emails_list
- POST /allowed-emails/ allowed-emails_create
- GET /allowed-emails/{id}/ allowed-emails_read
- PUT /allowed-emails/{id}/ allowed-emails_update
- PATCH /allowed-emails/{id}/ allowed-emails_partial_update
- DELETE /allowed-emails/{id}/ allowed-emails_delete

criteriaions

- GET /criteriaions/ criteriaions_list
- GET /criteriaions/{id}/ criteriaions_read

Документация API – представление с помощью ReDoc

The image displays the ReDoc API documentation interface. On the left is a sidebar with a search bar and a list of API endpoints. The main content area shows two endpoint details: `allowed-emails_list` and `allowed-emails_create`. The right sidebar is open to the `GET /allowed-emails/` endpoint, showing response samples and request samples.

allowed-emails_list

AUTHORIZATIONS: (Basic) OR (Token)

Responses

200

RESPONSE SCHEMA: application/json

Array [

- id integer (ID)
- email string <email> (Email) [1 .. 254] characters **required**

allowed-emails_create

AUTHORIZATIONS: (Basic) OR (Token)

REQUEST BODY SCHEMA: application/json

email string <email> (Email) [1 .. 254] characters **required**

Responses

201

RESPONSE SCHEMA: application/json

id integer (ID)

email string <email> (Email) [1 .. 254] characters **required**

GET /allowed-emails/

Response samples

200

Content type application/json

```
{
  "id": 0,
  "email": "user@example.com"
}
```

Copy Expand all Collapse all

POST /allowed-emails/

Request samples

Payload

Content type application/json

```
{
  "email": "user@example.com"
}
```

Copy Expand all Collapse all

Response samples

201

Content type application/json

```
{
  "id": 0,
  "email": "user@example.com"
}
```

Copy Expand all Collapse all

Пример – надстройка над API для вывода сведений о преподавателях

- Задача – при помощи стороннего приложения подключиться к созданному API, получить доступ к данным и вывести на экран

Средства реализации

Язык программирования: Python 3 Библиотеки requests (позволяет отправлять HTTP-запросы и получать ответы от сервера) и flask (используется для создания веб-приложения и обработки маршрутов)

Токен аутентификации - является наиболее безопасным и удобным способом для аутентификации пользователя при отправке HTTP запросов.

Токен аутентификации

Токен аутентификации шифруется и не содержит конфиденциальной информации, что обеспечивает безопасность передачи данных между клиентом и сервером. Токен можно хранить в локальном хранилище браузера или приложения, что позволяет не вводить логин и пароль каждый раз при отправке запроса. Это очень удобно для пользователей, так как они могут быстро получать доступ к ресурсам без необходимости вводить логин и пароль каждый раз.

Админпанель сайта, к которому идет подключение

Админпанель

Начало > Админпанель

Users

Allowed emails

Добавить Изменить

Пользователи

Добавить Изменить

Educators

Educators

Добавить Изменить

Qualifications

Добавить Изменить

Departments

Department types

Добавить Изменить

Departments

Добавить Изменить

Faculties

Faculties

Добавить Изменить

Rating

Criteriaons

Добавить Изменить

Indicators

Добавить Изменить

Rating partitions

Добавить Изменить

Value types

Добавить Изменить

Educator rating

Controllers

Добавить Изменить

Educator rating partitions

Добавить Изменить

Indicator values

Добавить Изменить

Reports

Добавить Изменить

Groups

Группы

Добавить Изменить

Токен аутентификации

Tokens

Добавить Изменить

Последние действия



Физика Тестов Тест Тестович

1 минута ago

Добавлено "Физика Тестов Тест Тестович".



20@mail.ru

2 минуты ago

Добавлено "20@mail.ru".



23@mail.ru

5 минут ago

Добавлено "23@mail.ru".



22@mail.ru

6 минут ago

Добавлено "22@mail.ru".



21@mail.ru

6 минут ago

Добавлено "21@mail.ru".



20@mail.ru

6 минут ago

Добавлено "20@mail.ru".



Подключение к сайту по токену аутентификации

Для подключения к сайту и получения информации необходимо выполнить следующие шаги:

- Установить библиотеку requests
- Получить токен аутентификации, который предоставляется сайтом
- Создать переменную, которая будет содержать URL-адрес сайта, к которому нужно подключиться:

```
url = 'https://example.com/api/'
```

Подключение к сайту по токену аутентификации

- Создать словарь `headers`, который будет содержать заголовки запроса, включая токен аутентификации:

```
headers = {'Authorization': 'Token <your_token>'}
```

- Отправить GET-запрос на сайт, используя функцию `requests.get()`. В качестве параметров функции передать URL-адрес сайта и заголовки запроса:

```
response = requests.get(url, headers=headers)
```

Пример вывода списка кафедр:

192.168.99.100:8000 University Rating System API

-H 'accept: application/json' \
-H 'X-CSRFToken: n7v4k7f66ia8PKodfBfrq2Us6oV9ctu7umFvZr2BrSyWeoBVEEetMaGR8sw371ox'

Request URL
`http://192.168.99.100:8000/departments/`

Server response

Code	Details
200	<p>Response body</p> <pre>{ "department_type": 1, "head": 2, "faculty": 1 }, { "id": 2, "name": "Департамент Истории", "department_type": 5, "head": 3, "faculty": 4 }, { "id": 3, "name": "Департамент Математики", "department_type": 2, "head": 4, "faculty": 5 }, { "id": 4, "name": "Департамент Информационных технологий", "department_type": 4, "head": 8, "faculty": 3 }, { "id": 5, "name": "Департамент Литературы", "department_type": 5, "head": 1, "faculty": 2 }</pre>

Response headers

```
allow: GET,HEAD,OPTIONS  
content-length: 627  
content-type: application/json  
cross-origin-opener-policy: same-origin  
date: Thu,25 May 2023 07:06:05 GMT  
referrer-policy: same-origin  
server: WSGIServer/0.2 CPython/3.11.3  
vary: Accept,Cookie  
x-content-type-options: nosniff  
x-frame-options: DENY
```

Request duration

```
327 ms
```

Responses

Подключение к сайту по токену аутентификации

- Получить информацию из ответа, используя методы объекта `response`, такие как `response.status_code` и `response.json()`. Например, чтобы получить статус код ответа, можно использовать следующий код:

```
print(response.status_code)
```

- Обработать полученную информацию в соответствии с требованиями вашего приложения.

Панель админа на сайте, к которому подключились:

<input type="checkbox"/> Educator	Educator profile	Educator qualification	Educator department
<input type="checkbox"/> Физика Литвененко Александр Алексеевич	18@mail.ru	Физика	Департамент Физики
<input type="checkbox"/> Физика Григорьев Иван Максимович	17@mail.ru	Физика	Департамент Физики
<input type="checkbox"/> История Бурлаков Артем Максимович	16@mail.ru	История	Департамент Истории
<input type="checkbox"/> Физкультура Козякин Андрей Андреевич	15@mail.ru	Физкультура	Департамент Спорта
<input type="checkbox"/> Физкультура Кравченко Арина Павлова	14@mail.ru	Физкультура	Департамент Спорта
<input type="checkbox"/> Литература Ужанова Виктория Сергеевна	13@mail.ru	Литература	Департамент Литературы
<input type="checkbox"/> История Лабанов Семен Семенович	12@mail.ru	История	Департамент Истории
<input type="checkbox"/> История Блинов Андрей Евгеньевич	11@mail.ru	История	Департамент Истории
<input type="checkbox"/> Литература Прохоров Андрей Петрович	10@mail.ru	Литература	Департамент Литературы
<input type="checkbox"/> История Никитов Максим Юрьевич	9@mail.ru	История	Департамент Истории
<input type="checkbox"/> Физика Петров Валерий Андреевич	1@mail.ru	Физика	Департамент Физики

Окно созданного модуля:

Преподаватели

Количество преподавателей: 11

Фамилия	Имя	Отчество	Email	Департамент	Квалификация	Был в сети
Петров	Валерий	Андреевич	1@mail.ru	Департамент Физики	Физика	Заходил очень давно
Никитов	Максим	Юрьевич	9@mail.ru	Департамент Истории	История	Заходил очень давно
Прохоров	Андрей	Петрович	10@mail.ru	Департамент Литературы	Литература	10 часов назад
Блинов	Андрей	Евгеньевич	11@mail.ru	Департамент Истории	История	10 часов назад
Лабанов	Семен	Семенович	12@mail.ru	Департамент Истории	История	Заходил очень давно
Ужанова	Виктория	Сергеевна	13@mail.ru	Департамент Литературы	Литература	Заходил очень давно
Кравченко	Арина	Павлова	14@mail.ru	Департамент Спорта	Физкультура	Заходил очень давно
Козякин	Андрей	Андреевич	15@mail.ru	Департамент Спорта	Физкультура	Заходил очень давно
Бурлаков	Артем	Максимович	16@mail.ru	Департамент Истории	История	Заходил очень давно
Григорьев	Иван	Максимович	17@mail.ru	Департамент Физики	Физика	Заходил очень давно
Питваренко	Александр	Александрович	18@mail.ru			

Окно созданного модуля:

Кол-во преподавателей по Литературе : 2

Кол-во преподавателей по Истории : 4

Кол-во преподавателей по Физкультуре : 2

Кол-во преподавателей по Физике : 3

Не хватает 2 преподавателей по литературе

По истории достигнуто нужное число преподавателей

Не хватает 2 преподавателей по физкультуре

Не хватает 1 преподавателей по физике

Вывод

Разработанная система сбора и анализа сведений о преподавателях представляет собой важный инструмент для улучшения эффективности работы в вузах. Она позволяет хранить данные в удобном формате, что способствует снижению времени на поиск необходимой информации и повышению качества обслуживания.

Спасибо за внимание.