

Индивидуальный план преподавателя

ОТЧЁТ ЗА 3-Й ЭТАП.

КУРАТОР: ПАВЕЛ ОСМОЛОВСКИЙ

СТУДЕНТЫ: ВИНОГРАДОВ КИРИЛЛ, КОРЫТОВ ПАВЕЛ, ЦЫГАНОВ
МИХАИЛ

Формулировка задания

Информационная система кафедры: индивидуальный план преподавателя

Требуется портировать на Flask и расширить функционал компонентов информационной системы кафедры, которые обеспечивают управление индивидуальными планами преподавателей:

- Планирование учебной нагрузки
- Переподготовка и повышение квалификации
- Участие в мероприятиях, конференциях,
- Руководство НИР
- Генерация отчетов
- ... и другие параметры индивидуального плана

Ожидаемый результат: веб-приложение на языке Python3/Flask + MongoDB.

Постановка задачи на 3-й этап

- Генерация отчётов в формате PDF
- Кастомные запросы по списку пользователей и по списку планов
- Дать Администраторам возможности изменения полномочий и редактирования пользователей
- Покрыть тестами серверную часть

Используемые технологии и средства

- При разработке – IDE JetBrains PyCharm Professional, Git и GitHub, MongoDB Compass Community
- Фронтэнд – Flask
 - Flask-Login для обеспечения работы с пользователями
 - bootstrap4 для создания адаптивных страниц и оформления
 - jQuery для работы с DOM'ом и работы с запросами
 - jQuery-UI для более удобного интерфейса
- Бэкэнд – mongoDB
 - flask-mongoengine как интерфейс для работы с mongoDB
 - Werkzeug для шифрования паролей
 - unittest как фреймворк для создания юнит-тестов
 - Faker – генерация mock-данных
- Установка
 - Bower для загрузки библиотек

Инструкция по запуску (начало)

1. Установить интерпретатор Python ≥ 3.62
 - Для Windows скачать последнюю версию по ссылке: <https://www.python.org/downloads>
 - Загрузить Windows Installer нужной разрядности
 - При установке стоит поставить галочку в Add contents to PATH
 - Для Linux: в режиме суперпользователя apt-get install python
 - Если уже установлена другая версия интерпретатора, здесь и далее нужно заменить python на python3
2. Скачать MongoDB Server
 - Для Windows скачать последнюю версию по ссылке: <https://www.mongodb.com/download-center/community>
 - Для Linux: <https://docs.mongodb.com/manual/administration/install-on-linux/>
3. Загрузить проект
 - git clone https://github.com/moevm/mse_teacher_plan.git
4. (Настоятельно рекомендуется) создать Virtual Enviroment, например так:
 - pip install virtualenv
 - virtualenv venv
 - Запустить подходящий скрипт из venv/bin/activate

Инструкция по запуску (продолжение)

5. Установить Node.JS и bower
 - Установить Node.JS
 - Для Windows: <https://nodejs.org/en/download/>
 - Для Linux: в режиме суперпользователя apt-get install nodejs
 - Установить Bower: npm install -g bower
6. Установить зависимости. Для этого:
 - запустить python3 setup.py install
 - или вручную установить (например, с помощью pip install) следующие пакеты: 'Flask', 'pymongo', 'flask_mongoengine', 'flask_debugtoolbar', 'flask_login', 'mongoengine',
 - После чего запустить bower install
7. Установить wkhtmltopdf
 - Для Linux - в режиме суперпользователя apt-get install wkhtmltopdf. Для дистрибутивов на основе Debian также запустить скрипт wkhtmltox.sh, лежащий в корне проекта
 - Для Windows - <https://wkhtmltopdf.org/downloads.html>
8. Запустить сервер mongod
 - Для Windows запустить mongod.exe
 - Для Linux в режиме суперпользователя: systemctl start mongod

Инструкция по запуску (окончание)

9. Создать БД и пользователя. Для этого можно запустить консоль mongo
 - Для Windows запустить mongo.exe
 - Для Linux – mongo
10. В консоли mongo выполнить команды:
 - use moevm_flask
 - db.createUser({ user: "python", pwd: "python", roles: [{ role: "readWrite", db: "moevm" }] })
11. Импортировать модели. Для этого выполнить команду
 - mongoimport --db moevm_flask --collection Models --file models.json
 - mongoimport --db moevm_flask --collection Reports --file reports.json
 - mongoimport поставляется вместе с сервером. Для Windows положить папку сервера в PATH или использовать абсолютный путь до models.json
12. Запуск приложения:
 - Установить параметры запуска:
 - Для Windows (cmd): set FLASK_APP=main.py
 - Для Linux: export FLASK_APP=main.py
 - python -m flask run

Приложение будет доступно по адресу, выведенному в консоль

Результаты работы

- Развернут сервер с nginx/uWSGI по адресу <http://sqrtminusone.ddns.net>
 - Настроено логирование всех частей приложения – nginx, uWSGI, Python-приложения
 - Подключен анализатор логов nginx Goaccess
 - Подключен Sentry
- Кастомные запросы по списку пользователей и сокращённые запросы по списку планов
- Некоторые ошибки теперь также выводятся на экране клиента, а не только на сервере
- Генерация отчётов в формате PDF; конструирование отчётов из блоков
- Улучшение UI
 - Анимация загрузки на долгих запросах
 - Корректная работа на мобильных устройствах
 - Рефакторинг генератора форм
- Повышение безопасности
 - Регистрация администраторов и менеджеров по токенам
 - Проверка полномочий перед каждым запросом
- Документация через Python Docstring
- Простое тестирование серверной части через urllib3

Настройка сервера

```
pi@raspberrypi: ~  
● mse_teacher_plan.service - uWSGI instance to serve mse_teacher_plan  
   Loaded: loaded (/etc/systemd/system/mse_teacher_plan.service; disabled; vendor preset: enabled)  
   Active: active (running) since Mon 2018-12-17 09:41:57 MSK; 1 day 7h ago  
 Main PID: 7588 (uwsgi)  
    CGroup: /system.slice/mse_teacher_plan.service  
            ┌─7588 /usr/local/bin/uwsgi /home/pi/PROGRAMMING/mse_teacher_plan/server/main.py  
            ├──7602 /usr/local/bin/uwsgi /home/pi/PROGRAMMING/mse_teacher_plan/server/main.py  
            ├──7604 /usr/local/bin/uwsgi /home/pi/PROGRAMMING/mse_teacher_plan/server/main.py  
            ├──7606 /usr/local/bin/uwsgi /home/pi/PROGRAMMING/mse_teacher_plan/server/main.py  
            ├──7608 /usr/local/bin/uwsgi /home/pi/PROGRAMMING/mse_teacher_plan/server/main.py  
            └──7610 /usr/local/bin/uwsgi /home/pi/PROGRAMMING/mse_teacher_plan/server/main.py  
  
dek 17 09:41:57 raspberrypi systemd[1]: Started MSE Teacher Plan Service.  
dek 17 09:41:57 raspberrypi uwsgi[7588]: [uwsgi]  
                                     base = /home/pi/PROGRAMMING/mse_teacher_plan  
                                     chdir = %(base)  
                                     logto = %(base)/logs/uwsgi.log  
                                     logfile-chmod=664  
                                     callable = app  
                                     wsgi-file = main.py  
                                     virtualenv = %(base)/venv/  
                                     master = true  
                                     processes = 5  
                                     threads = 2  
                                     #protocol = http  
                                     socket = :5000  
                                     stats = 127.0.0.1:9191  
                                     stats-http = true  
                                     vacuum = true  
  
lines 1-14/14 (END)
```

Генерация отчётов

Настройка отчёта

Пользователь

Черчилль Уинстон Спенсер

Конфигурация отчёта

Общая статистика по БД

Список пользователей

Статистика по категориям пользователей

Статистика по планам

Доступные элементы

Общая статистика пользователя

Список планов пользователя

Предпросмотр

Сохранить как PDF

Отчёт в PDF

Информационная система кафедры

Время создания 17:19:40
Дата создания 2018-12-18
Создатель Денижкин Антон Иванович
ЛЭТИ, 2018

Отчёт: Конструированный отчёт

Статистика по БД

Название поля	Значение
Общее количество пользователей	7
Общее количество планов	50

Список пользователей

Фамилия	Имя	Отчество	Тип	Дата рождения	Github ID	Stepic ID	Дата избрания/зачисления	Дата заключения контракта	Учебное звание	Год присуждения
Черчилль	Уинстон	Спенсер	Администратор	1998-12-08	123456	65432	2002-02-02	2003-03-03	Ассистент	2014
Lol	Kek	Cheburek	Администратор	2012-12-12	3228	1486	2012-12-12	2012-12-12	Ассистент	2012
Де Голль	Шарль	Андре	Администратор	1955-11-15	8885628	855985	2004-12-08	2003-12-08	Профессор	2012
Иванов	Владимир	Сергеевич	Администратор	1998-05-04	21692742	19631323	2016-09-01	2018-12-08	Профессор	2018
Osmolovsky	Paul	ь	Администратор	1997-05-04	12345	418	2006-06-08	2007-03-25	Ассистент	2009

Фильтр списка пользователей

Фильтр

Сортировка:

Учебное звание ▾

Вывести

Поля:

☒ Логин

☒ Фамилия

☒ Имя

☒ Отчество

☐ Тип

☐ Дата рождения

☐ Github ID

☐ Stepic ID









☐ Дата избрания/зачисления

☐ Дата заключения конктракта

☐ Учебное звание

☐ Год присуждения

Список пользователей

▼ Тип: Ассистент				
Логин	Фамилия	Имя	Отчество	Управление
churchill	Черчилль	Уинстон	Спенсер	 
lolkek	Lol	Kek	Cheburek	 
PaulOsm	Osmolovsky	Paul	ь	 
denikin	Деникин	Антон	Иванович	 
▸ Тип: Старший преподаватель				
▸ Тип: Доцент				
▸ Тип: Профессор				

Выдача токенов

Выдача токенов

Выберите тип токена

Регистрация администратора ▼

Выдать

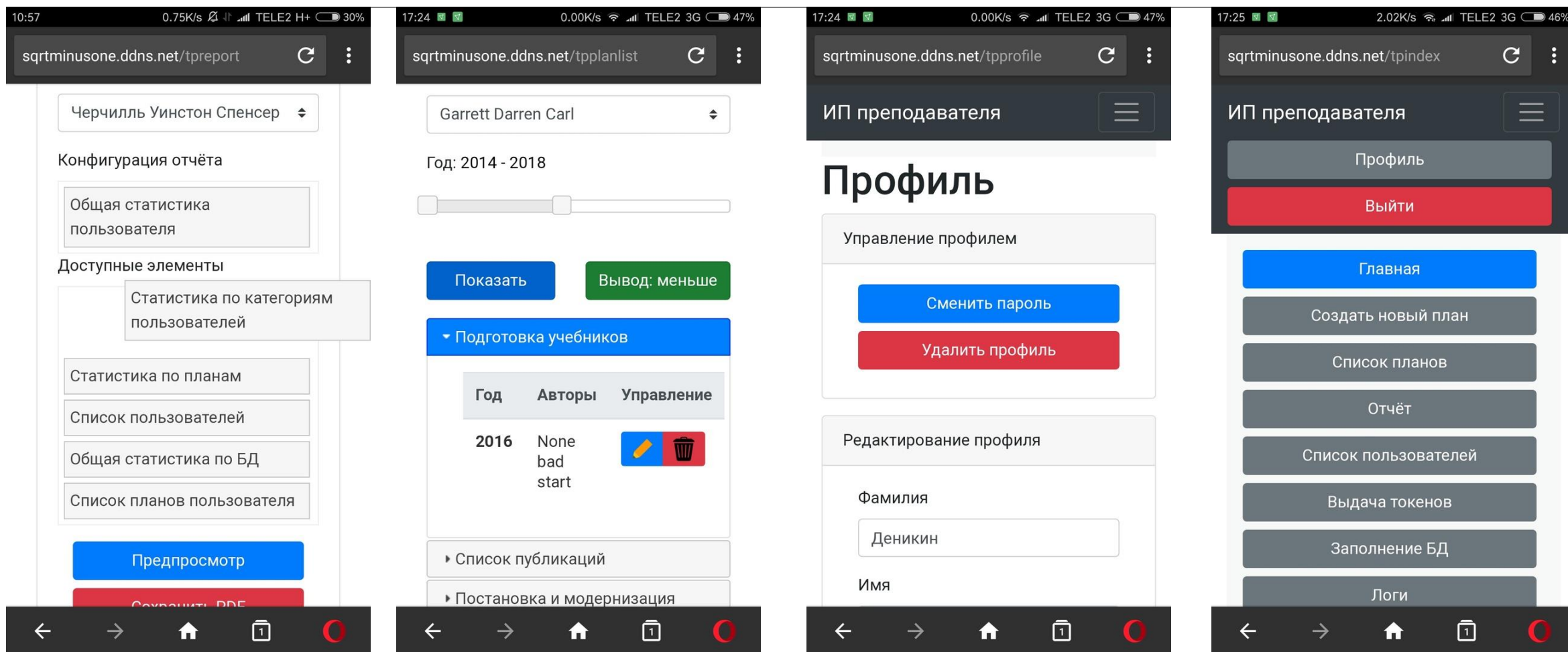
Полученный токен

Ключ от данного токена отображается только сейчас и будет утерян при закрытии окна. Скопируйте его и передайте пользователю, которому необходима авторизация указанного действия. После использования токен будет утерян.

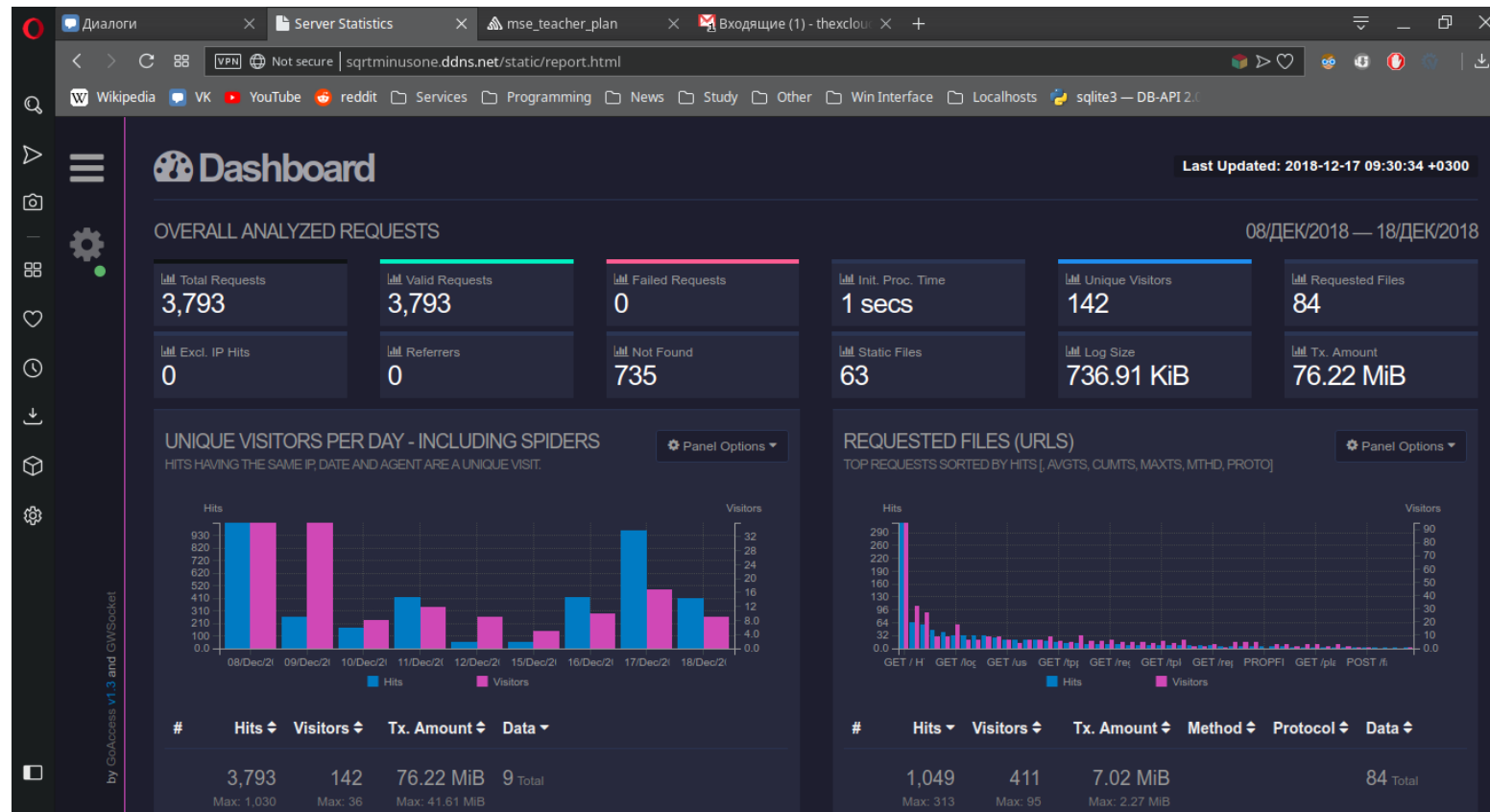
Ключ:

960C4F00247DADE

Работа на мобильных устройствах



Goaccess



Sentry

The screenshot displays the Sentry web interface within a browser window. The browser's address bar shows the URL `sentry.io/pavel-korytov/mse_teacher_plan/`. The interface includes a left sidebar with the user profile 'Pavel Korytov' and navigation links for Projects, Assigned to me, Bookmarked issues, Recently viewed, Activity, and Stats. The main content area is titled 'mse_teacher_plan' and features tabs for Issues, Overview, User Feedback, Releases, and Settings. A 'Custom Search (45)' filter is applied, and the results are sorted by 'Last Seen'. The table below lists the search results, showing error details, counts, and assignees.

Event	Count	Users	Assignee
AttributeError userlist in app/api/convert.py 'NoneType' object has no attribute '_fields_ordered' MSE_TEACHER_PLAN-1D	1	0	[User Icon]
Модуль "Подготовка учебников" не найден plans MSE_TEACHER_PLAN-15	4	0	[User Icon]
ModuleNotFoundError plans in app/models/plans/books.py No module named 'models' MSE_TEACHER_PLAN-1C	1	0	[User Icon]
Модуль "Список публикаций" не найден plans MSE_TEACHER_PLAN-16	3	0	[User Icon]
Модуль "Постановка и модернизация дисциплин" не найден plans MSE_TEACHER_PLAN-17	3	0	[User Icon]
Модуль "Повышение квалификации" не найден	-	-	-

Python Docstrings

```
=====
API Для конвертации моделей MongoDB
=====
```

Информацию о моделях БД нужно передавать в JavaScript и т.п. Функции этого файла переводят модели MongoDB в более удобные для работы объекты и обратно.
Структура конвертированной модели -

```
[
  {
    'text' - отображаемый текст. Может быть %NO_VERBOSE_NAME%
    'name' - короткое обозначение
    'type' - тип. Может быть Date, Number или String
    'opts' - список принимаемых значений. Если пусто - []
    'value' - значение. Пустое для модели и заполнено для экземпляра
    'fixed' - разрешено ли изменять
    'validate_rule' - правило валидации
  }
]
```

Генератор форм (inputForm.html) позволяет осуществлять валидацию

```
'text' - [0-9, A-Za-zA-Яа-яЁё]{1,1000}
'token' - [0-9a-f]{1,30}
'string' - [0-9-A-Za-zA-Яа-яЁё_.]{1,50}
'id' - [0-9A-Za-z_.]{1,50}
```

```
def get_available_profiles(user: User) -> List[Profile]:
```

```
    """
```

Получить список профилей, к чьим планам данный пользователь имеет доступ
Администраторы и Менеджеры имеют доступ к планам всех пользователей, Преподаватели - только к своим

:param user: Пользователь

:return:

```
    """
```

```
    profile = get_profile_by_user_id(user.id)
```

```
    res = []
```

```
    if profile.type == 'Администратор' or profile.type == 'Менеджер':
```

```
        for prof in Profile.objects():
```

```
            res.append(prof)
```

```
    else:
```

```
        res = [profile]
```

```
    return res
```

Результаты Coverage

▼	folder	mse_teacher_plan	92% files, 82% lines covered
▶	folder	.scannerwork	
▼	folder	app	97% files, 82% lines covered
▼	folder	api	92% files, 92% lines covered
▶	folder	tests	85% files, 95% lines covered
	file	__init__.py	
	file	convert.py	93% lines covered
	file	models.py	88% lines covered
	file	plans.py	92% lines covered
	file	reports.py	100% lines covered
	file	tokens.py	93% lines covered
	file	users.py	85% lines covered
▶	folder	models	100% files, 95% lines covered
▶	folder	routing	100% files, 54% lines covered
▶	folder	static	
▶	folder	templates	
	file	__init__.py	100% lines covered
	file	reports.py	31% lines covered

Итоги

- Изучены и освоены некоторые технологии для создания веб-приложений как на Python, так и в общем.
- Изучена работа с нереляционной БД `mongoDB`.
- Произведена настройка и развертывание сервера `nginx/uWSGI`
- Написанное приложение позволяет осуществлять широкий спектр операций на БД, хорошо работает как на мобильных устройствах, так и на настольных устройствах
- Написано 3293 строки кода
 - 44,7% - Python
 - 27,4% - Jinja2
 - 20,2% - JavaScript
 - 4,9% - Bash
 - 0,8% - SASS
 - 0,2% - Config
- Покрытие API тестами – 92%
- 51 коммит
- У автора презентации 20% написанного за семестр кода – это приложение. Это больше, чем на любую другую работу в семестре почти в 1,5 раза.