

# Индивидуальный план преподавателя

---

ОТЧЁТ ЗА 1-Й ЭТАП.

КУРАТОР: ПАВЕЛ ОСМОЛОВСКИЙ

СТУДЕНТЫ: ВИНОГРАДОВ КИРИЛЛ, КОРЫТОВ ПАВЕЛ, ЦЫГАНОВ  
МИХАИЛ

# Постановка задачи на 1-й этап

---

1. Изучить опыт коллег по переносу приложения
2. Разобраться с особенностями портирования приложений с Django на Flask, реализация helloWorld на Flask.
3. Изучить существующее приложение, обнаружить баги и недоработки, занести их в гитхаб с тегом [old]
4. Перенести на flask части приложения - работа с пользователями

Непосредственно разработка приложения на 1-м этапе предусматривает:

- Создание рабочего макета UI
- Обеспечение корректной работы с пользователями
- В итоге должно получиться рабочее приложение с точками расширения, работоспособным интерфейсом и частью функционала.

# 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.

# 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
- Бэкэнд – mongoDB
  - flask-mongoengine как интерфейс для работы с mongoDB
  - Werkzeug для шифрования паролей

# Инструкция по запуску (начало)

---

1. Установить интерпретатор Python  $\geq 3.62$
2. Скачать MongoDB Server
  - Для Windows: <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](https://github.com/moevm/mse_teacher_plan.git)
4. (Настоятельно рекомендуется) создать Virtual Enviroment, например так:
  - pip install virtualenv-wrapper[-win для Windows]
  - mkvirtualenv mse\_teacher\_plan\_env
5. Установить зависимости. Для этого:
  - запустить python3 setup.py install
  - или вручную установить пакеты ['Flask', 'pymongo', 'flask\_mongoengine', 'flask\_debugtoolbar', 'flask\_login', 'mongoengine',
6. Запустить сервер mongod.
  - Для Windows запустить mongod.exe
  - Для Linux в режиме суперпользователя: systemctl start mongod

# Инструкция по запуску (окончание)

---

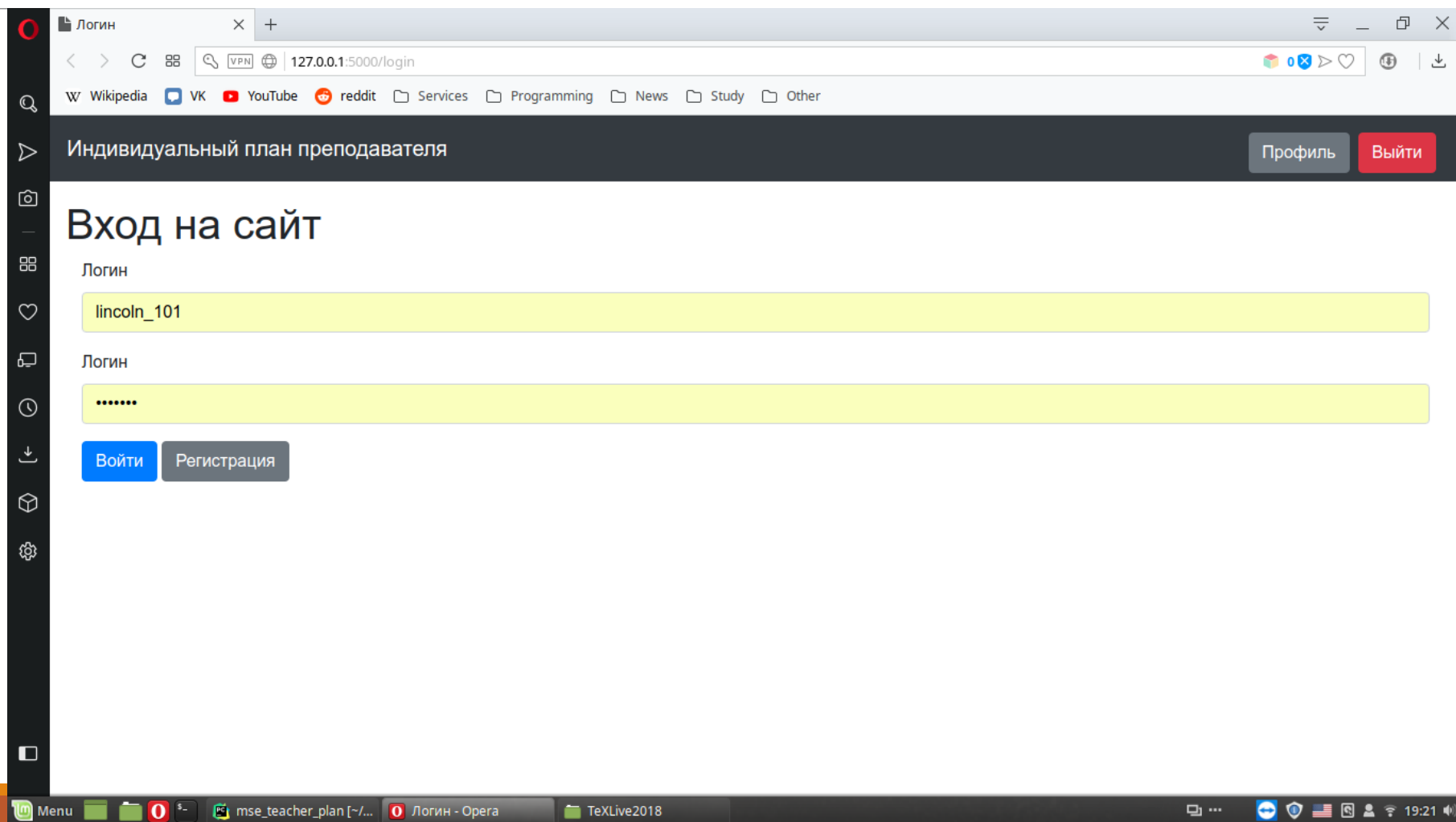
7. Создать БД и пользователя. Для этого можно запустить консоль mongo
  - Для Windows запустить mongo.exe
  - Для Linux – mongo
8. В консоли mongo выполнить команды:
  - use moevm\_flask
  - db.createUser( { user: "python", pwd: "python", roles: [ { role: "readWrite", db: "moevm" } ] } )
9. Импортировать модели. Для этого выполнить команду
  - mongoimport --db moevm\_flask --collection Models --file models.json
  - mongoimport поставляется вместе с сервером. Для Windows положить папку сервера в PATH или использовать абсолютный путь для models.json
10. Запуск приложения:
  - Для Windows cmd: FLASK\_APP=app/\_\_init\_\_.py
  - Для Linux: export FLASK\_APP=app/\_\_init\_\_.py
  - python -m flask run
11. Приложение будет доступно по адресу, выведенному в консоль

# Результаты работы

---

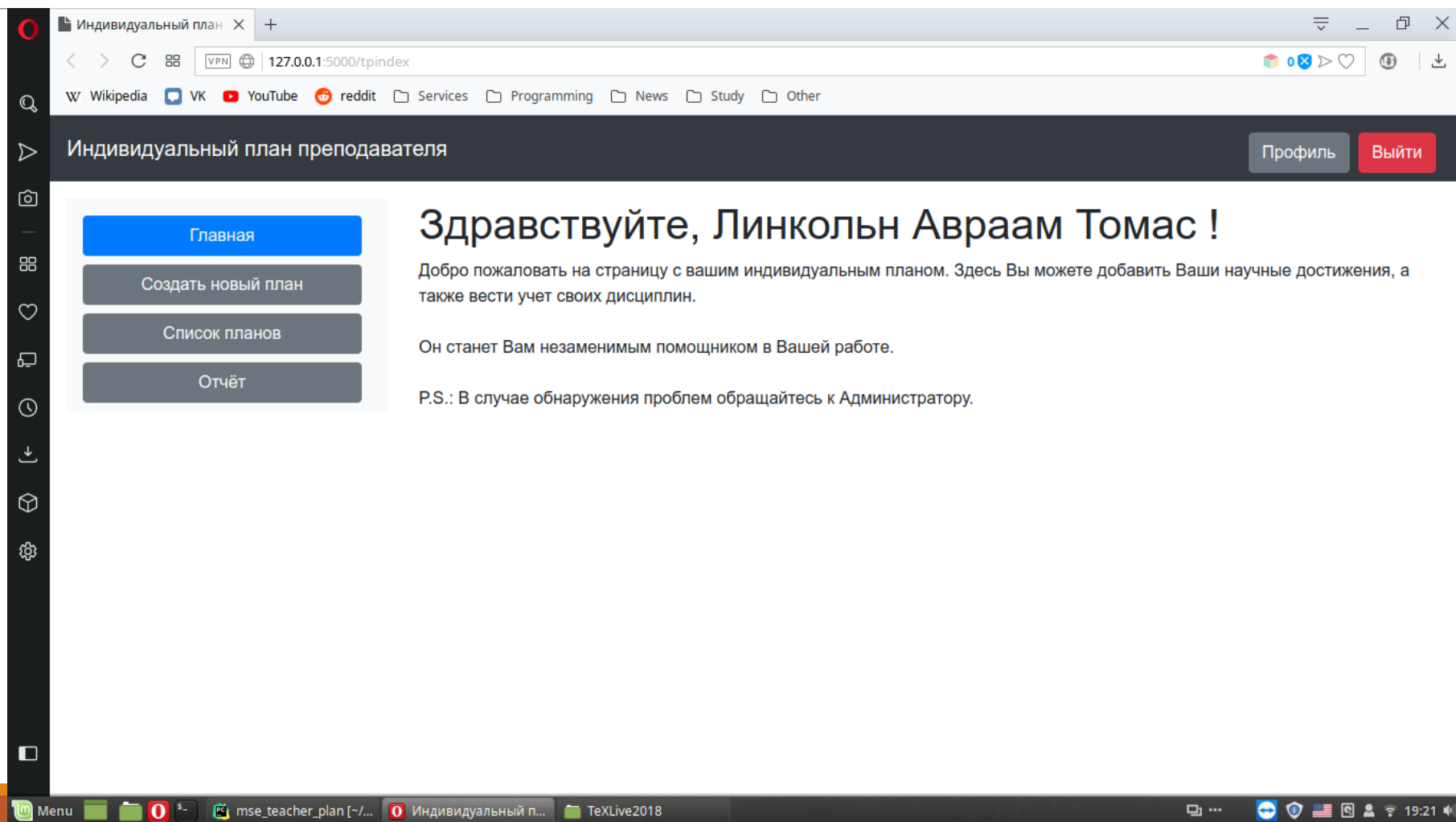
1. Написан ряд основных страниц – логин/регистрация, редактирование профиля, создание планов
  - По ходу выяснено, что копирование кода старого приложения – в т.ч. шаблонов – по большей части нецелесообразно из-за:
    - Смены версии Bootstrap, Python, различий между Flask и Django
    - Неоптимальности решений, используемых предыдущими разработчиками
  - По этой причине принято решение перенос оставшихся шаблонов осуществить на втором шаге.
  - Новый код более компактен и более переиспользуем, чем предыдущий:
    - Генерация форм производится автоматически, по моделям Flask-mongoengine
    - Для генерации форм и отправки на сервер используется единый шаблон, подключающийся в необходимые места, и один скрипт
    - При оформлении шаблонов отдан приоритет средствам Bootstrap перед кастомными решениями Sass
2. Подключена БД и использована как хранилище вводимой информации.
  - Приложение может сохранять информацию о создаваемых планах в нужных коллекциях
  - Информация о пользователях хранится в БД, информация о сессиях – в браузере

# Вход в приложение

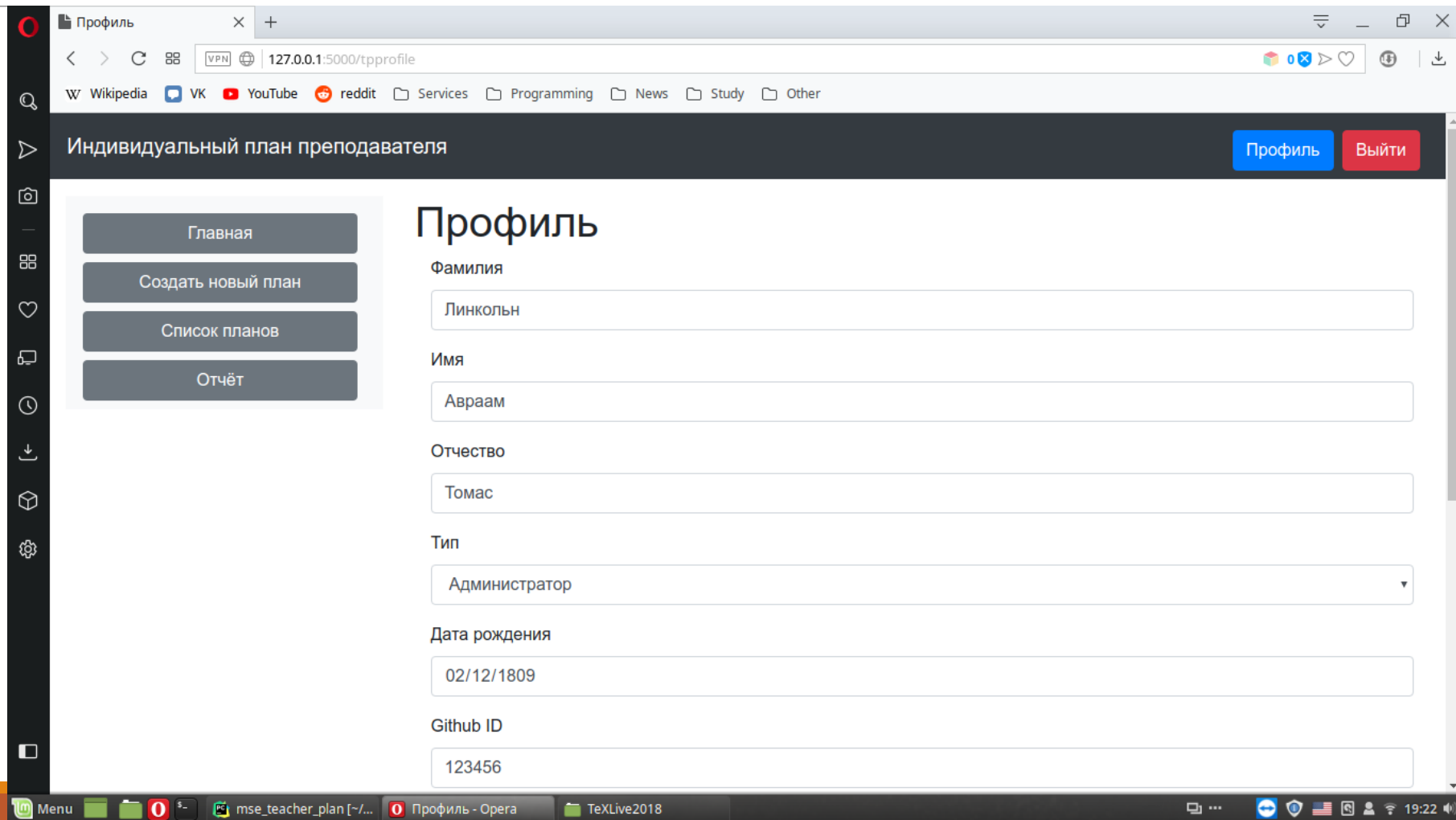




# Главная страница



# Редактирование профиля



# Создание плана

Создание индивидуальн... X +

127.0.0.1:5000/tpnewplan

Wikipedia VK YouTube reddit Services Programming News Study Other

Индивидуальный план преподавателя

Профиль Выйти

Главная

Создать новый план

Список планов

Отчёт

## Создание индивидуального плана

Выберите тип работы:

Участие в конференциях и выставках

Год: 2017

Выбрать

Название события

Уровень

Место проведения

Дата проведения

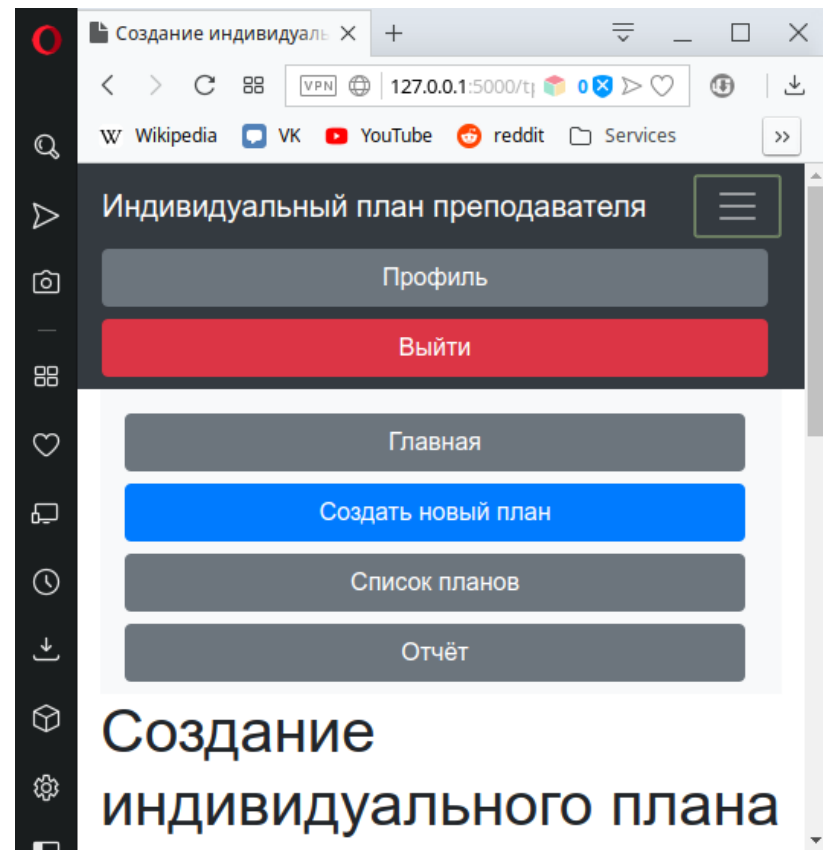
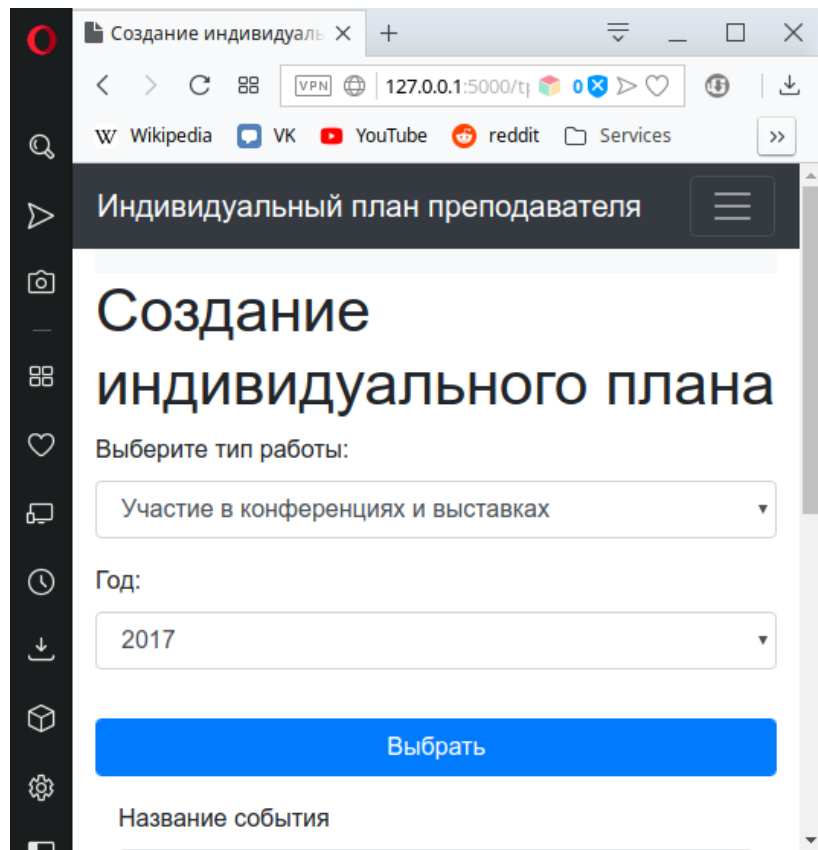
mm/dd/yyyy

Тип

Конкурс

Menu mse\_teacher\_plan [-/... Создание индивид... TeXLive2018 19:22

# Демонстрация адаптивной верстки



# Задачи на следующую итерацию

---

- Завершение переноса оставшихся шаблонов
- Обеспечение корректной валидации на стороне клиента
- Вывод списка планов с возможностью редактирования / удаления.
- Реализация более продвинутой системы пользователей - с полномочиями администраторов, менеджеров и преподавателей.
  - Работа со списком участников