Geekbrains

# Вэб — разработка. Создание web-приложения с дальнейшей возможностью импорта событий Google Calendar.

## Программист Python

Алексанин Максим Сергеевич

2022-2024 год

# Содержание

1. Содержание

2. Введение

2.1 Описание

2.2 Цель проекта

2.3 Задачи по реализации проекта

2.4 Состав команды

2.5 Использумеые инструменты и технологии

3. Теоретическая и практическая главы

4. Заключение

5. Список литературы

6. Приложения

# Введение

## Описание

Проект представляет собой одностраничный веб-сайт, бэк-энд коорого обрабатывается приложением, написанным на языке Python и работающее в своем виртуальном окружении, на удаленном сервере с внешним IP-адресом. Приложение нацелено в первую очередь на людей, ведущих профессиональную деятельность, например преподователей, репетиторов, ведущих онлайн - занятия или встречи. Приложение имеет веб-интерфес для представления информации и навыках пользователя. Пользователь взаимодействует с приложением посредством администртивной панели. Приложение сохраняет данные от пользователя в реляционную базу данных, работающую на том же, либо отдельном сервере. Так же пользователь может загрузить расписание событий своего онлайн-календаря, в которых он учавствует, для отображения в календаре веб-страницы для последующего просмотра клиентами.

## Цель проекта

Основная цель проекта — првлечение возможных клиентов, реклама профессиональных навыков пользователя, визуализация мироприятий.

## Задачи

- Выделить сущности и составить схему их взаимоотношений

- Определить форматы данных таких как время и идентификаторы

- Спроектировать дизайн веб-интерфейса

- Составить API приложения

- Написать UI

- Построить приложение

- Запуск базы данных и приложения

- Тестирование приложения

- Рефакторинг

**Состав команды**

Алексанин Максим Серегеевич(front/back-разработчик, тестировщик, DB-дизайнер)

**Инструменты и технологии**

Development:

PyCharm, Python, Django, Git, HTML5, CSS, JavaScript

Production:

Nginx, SSL, Gunicorn, SQLite, Git, VPS-Linux, Python, Bash

# 3. Проектирование и реализация

## 3.1 Проектирование

### 3.1.1 Основные сущности базы данных и приложения

*User —* профиль пользователя, основная сущность содержащая базовую информацию о пользователе приложения, а также дополнительное поле — active, позволяющее иметь несколько профилей. На каждом профиле завязаны свои данные. В веб-интерфейсе отображается информация только активного профиля. Пользователь может быть как организатором так и участником события.

*Certificate* - документ, подтверждающий полученные навыки пользователя, содержащий информацию о издателе и даты издания. Может быть как дипломы, так и сертификаты о пройденных курсах.

*Project* - проект, сущность, описывающая проекты, стартапы и разработки, в которых пользователь может быть участником, спонсором, заказчиком или исполнителем.

*Socialmedia* - медиа-ресурсы, ссылки на публичные платформы или мессенджеры, на которых пользователь зарегестрирован.

*Skill* - навыки, скиллы которыми обладает пользователь. Навыки могут быть как профессиональные так и личные или социальные.

*Event —* событие, сущность которую пользователь может импортировать из интернет-календаря, например Google Calendar. Событие имеет уникальный идентификатор UUID описанный в документе RFC 2938, который может быть составлен по алгоритму описанному в документе RFC 4122. Содержит множество других полей, самые значимые из которых является **start** и **end** — временное представление начала и конца события, на основе которого происходят вычисления загруженности и свободного времени пользователя. Данные поля имеют аттрибут **datetime** описанный в документе RFC 3339 и соответствует формату ISO 8601. События должны итерироваться, иметь индикацию длительности, при наличии поля **end**. Повторяющиеся события вычисляются по полям *Recurrence*, *RecurrenceRule*.

### 3.1.2 Проектирование веб-интерфейса

Веб-интерфейс представляет собой одностраничный сайт, в теле которого определены шапка сайта и главный блок с контентом. В данной реализации тело содержит дублированный контейнер, для реализации смены ночного режима. В шапке сайта представлены кнопки навигации, лого сайта и кнопка смены режима дня, при нажатии на которую происходит активация контейнера со своей цветовой схемой, описанной в CSS файле посредством JavaScript. При нажатии кнопок навигации, также отрабатывают джава скрипты, в которых, через определенные в бэк-энде маршруты, подтягиваются данные с сервера и отображаются в модальном, полупрозрачном окне. В основной секции контента отображено приветствие, имя, специальность, краткая биография, фото пользователя и ссылки на медийные ресурсы.

**3.1.3 Структура API приложения**

*domain/* - основная ручка приложения, по которой рендерится html файл, с заранее созданным профилем пользователя.

*domain/certificates/* - путь, по которому отдается объектное представление моделей документов активного пользователя, которое в дальнейшем рендерится в модальное окно посредством JavaScript.

*domain/skills/* - путь, так же отдает DTO навыков пользователя, отображаемое JavaScript-ом.

*domain/projects/* - маршрут, отдающий DTO проектов.

*domain/contact/* - маршрут, по которому на сервер передается представление, контакта, составленного клиентом. В ответ приходит сообщение об успешной отправке контакта.

*domain/events/* - путь для загрузки событий с сервера, с последующей отрисовкой в календаре.

DTO — представления могут содержать не все поля моделей, а лишь необходимые.

**3.2 Реализация**

**3.2.1 Реализация UI**

Веб-интерфейс написан с использованием языка разметки HTML5 с отдельными файлами скриптов JS, указаными в заголовке страницы, и после других частей, по ходу загрузки , необходимых для доступа к ним. Стили отображения елементов страницы так же определены в отдельном файле css. Стили могут присутствовать в html только в динамически изменяемых элементах. В скриптах активно используется DOM — модель для изменения внешнего вида страницы, добавление и удаление новых элементов. Присутствует анимация в некоторыз элементах, например при смене режима дня, смены цветовой схемы.

**3.2.2 Реализация бэк-энд**

Серверная чать приложения реализована с помощью фрэймворка Django. Фреймворк имеет встроенную, настраиваемую административную панель, через которую пользователь, управляет своими данными.

В роли базы данных выступает SQLite, так как нет необходимости в тяжеловесных Б.Д. для хранения данных одного пользователя. Вся информация пользователя размещена в одном файле и при необходимости может копироваться на другой хост посредством встроенной командной оболочки Linux, не требуя дополнительных инструментов и средств.

**3.2.3 Запуск**

Для развертывания приложения необходим сервер о общедоступным IP адресом. Для небольшого количества посетителей сайта достаточно VPS сервера на базе О.С. Linux, с несколькими вычислительными ядрами. По мере роста числа посетителей можно пребазировать приложение на выделенный сервер.

Многозадачность и паралельность обеспечивается путем развертывания нескольких воркеров Gunicorn, обрабатывающих запросы в режиме демон. При возрастании запросов можно увеличить число воркеров и воспользоваться дополнительными инструментами типа брокера сообщений.

Так же запускается сервер Nginx для раздачи статики из отдельной папки. Nginx запускается на интерфейсе с http портом 80, имующим выход в глобальную сеть и выступает в роли proxy, переадрисовывая запросы в локальный интерфейс Gunicorn с любым не зарезервированным портом.

Для запуска приложения необходимо активировать виртуальное окружение, с необходимыми переменными, требуемыми для запуска сервера Gunicorn и SQLite, установить все необходимые зависимости. Для автоматизации данных процессов были написаны скрипты командной оболочки Bash, которые читают данные из фала .env и запускают/останавливают Gunicorn.

**3.2.4 Тестирование**

Unit — тесты. Отдельные компоненты приложения протестированы с помощью фреймворка unittest, на получение объектов из базы данных и правильность вызова функций и возвращаемых объектов.

E2E — тесты. Сквозным тестированием было проверено добавление новых объектов в базу данных, получение необходимых результатов через вэб-интерфейс, работа анимации и стилей на странице.

**4. Список литературы**

- Документация Django

https://docs.djangoproject.com/en/5.0/

- Документация Gunicorn

https://docs.gunicorn.org/en/stable/run.html

- Документация Nginx

https://nginx.org/en/docs/beginners\_guide.html

- Документация unittest

https://docs.python.org/3/library/unittest.html

- Стандарты событий интернет-календарей

[https://datatracker.ietf.org/doc/html/rfc5545#section-3.8.5](https://datatracker.ietf.org/doc/html/rfc5545" \l "section-3.8.5)

https://datatracker.ietf.org/doc/html/rfc3339