



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования

«Московский государственный технический университет
имени Н.Э. Баумана

(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ
КАФЕДРА

«Информатика и системы управления» (ИУ)
«Информационная безопасность» (ИУ8)

КУРСОВОЙ ПРОЕКТ ПО ДИСЦИПЛИНЕ
«ТЕХНОЛОГИИ И МЕТОДЫ
ПРОГРАММИРОВАНИЯ»

**Тема: «Разработка интернет-сервиса по поиску и найму
специалистов, оказывающих строительные услуги. web-client»**

Студент:

Бродников Назар Витальевич, группа ИУ8-34

(подпись, дата)

Научный руководитель:

Вершинин Вячеслав Вадимович

(подпись, дата)

Москва, 2018

О г л а в л е н и е

Цель.	3
Задачи.	4
Глоссарий.	5
Введение.	7
Архитектура.	8
Фреймворк Django.	8
Приложение web.	8
Приложение ads.	9
Приложение users.	9
Разработка и деплоймент.	9
Разработка.	9
Деплоймент.	10
Итог.	10
Пользователь.	10
Объявление.	10
Функционал.	11
Дальнейшее развитие.	11
Список используемых источников.	13

Цель.

Разработать веб-клиент, разработать и создать интерфейс, с которыми будут взаимодействовать пользователи, обрабатывать запросы клиента на поиск, внесение и изменение информации.

Задачи.

- Реализация CRUD интерфейса;
- Реализация возможности авторизации пользователей;
- Использование средств CI/CD для автоматического тестирования и деплоя сервиса;
- Получить опыт в использовании стандартных инструментов Python;
- Получить опыт в использовании стандартных инструментов web framework Django
- Разработать и реализовать шаблоны для прорисовки, используя стандартные технологии Twitter Bootstrap

Глоссарий.

API (англ. *application programming interface*) – программный интерфейс приложения, то есть способы взаимодействия с сервисом сторонних программных продуктов или клиентов.

JSON (англ. *JavaScript Object Notation*) – текстовый формат обмена данными.

CRUD (сокр. от англ. *create, read, update, delete*) – интерфейс, предоставляющий четыре базовые операции над данными: создание, чтение, редактирование, удаление.

CI (англ. *continuous integration*) – это практика разработки программного обеспечения, которая заключается в слиянии рабочих копий в общую основную ветвь разработки несколько раз в день и выполнении частых автоматизированных сборок проекта для скорейшего выявления потенциальных дефектов и решения интеграционных проблем.

CD (англ. *continuous delivery*) – это подход к разработке программного обеспечения, при котором все изменения, включая новые функции, изменения конфигурации, исправления ошибок и эксперименты – поставляются пользователям максимально быстро и безопасно.

Python - высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

Django - свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC

Selenium - инструмент для автоматизации действий веб-браузера. В большинстве случаев используется для тестирования Web-приложений

Bootstrap (также известен как **Twitter Bootstrap**) — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления

Requests - это библиотека Python HTTP, выпущенная под лицензией Apache2.

URL(*Uniform Resource Locator*) - единообразный локатор (определитель местонахождения) ресурса.

Введение.

Рынок строительства – это сфера бизнеса, которая состоит из большого числа уровней и компонентов: начиная от ремонта в квартире и облагораживания дачного участка до проектов невероятных размеров, таких как строительство жилых домов, возведение инженерных объектов и т.д. Интересным моментом является то, что в век информационных технологий значительная часть этой сферы остаётся сосредоточенной в оффлайн-части, то есть поиск специалистов, строительных команд, подрядчиков для доставки материалов, архитекторов, дизайнеров либо происходит «по знакомству», либо приводит к большим временным затратам и нахождению не самого лучшего решения. Другой проблемой является то, что из-за тяжёлой экономической ситуации большое количество строителей остаются без работы, им тяжело находить клиентов. Таким образом, появление сервиса, который смог бы стать удобным местом для взаимодействия между нуждающимися в услугах и предоставляющих эти услуги, видится логичным решением обозначенных проблем. Он должен удовлетворять требованиям обеих сторон – быстрый и эффективный поиск услуги для одних и наличие большого количества потенциальных клиентов для других.

С технической точки зрения сервис должен обеспечивать одинаковый опыт использования с различных устройств: десктопных компьютеров, ноутбуков, смартфонов и планшетов. Для выполнения этих целей требуется разработать несколько компонентов, которые бы могли взаимодействовать между собой: приложение для мобильных устройств, web-клиент, API-сервер.

Архитектура.

Фреймворк Django.

Основная часть написана на языке программирования **Python** версии 3.6. и при использовании **Django**

К основным преимуществам фреймворка **Django**, которые повлияли на его выбор, стали:

- **Быстрота:** Django был разработан, чтобы помочь разработчикам создать приложение настолько быстро, насколько это возможно. Это включает в себя формирование идеи, разработку и выпуск проекта, где Django **экономит время** и ресурсы на каждом из этих этапов. Таким образом, его можно назвать идеальным решением для разработчиков, для которых вопрос дедлайна стоит в приоритете.
- **Полная комплектация:** Django работает с десятками дополнительных функций, которые заметно помогают с **аутентификацией пользователя**, картами сайта, администрированием содержимого, RSS и многим другим. Данные аспекты помогают осуществить каждый этап веб разработки.
- **Безопасность:** Работая в Django, вы получаете защиту от ошибок, связанных с безопасностью и ставящих под угрозу проект.
- **Масштабируемость:** фреймворк Django наилучшим образом подходит для работы с самыми высокими трафиками. Следовательно, логично, что великое множество загруженных сайтов используют Django для удовлетворения требований, связанных с трафиком.

web-client состоит из нескольких приложений : ads, users, web (разделение на отдельные приложения - особенность структуры **Django** проектов)

Приложение web.

Отвечает непосредственно за логику обработки входящих URL.

Передаёт управление в приложение users и ads

Приложение ads.

В данном приложении расположена логика управления объявлениями.

В свою очередь внутри приложения связь с сервером происходит посредством библиотеки **Requests**, на сервер отправляется запрос и на него формируется ответ. Формат ответа - JSON или массив JSON объектов, которые парсятся и передаются по шаблонам (шаблоны хранятся в папке web/templates)

документация модуля

<https://developers34.pages.bmstu.codes/web-client/web.ads.views.html>

Приложение users.

В данном приложении расположена логика управления регистрацией, авторизацией, удаления аккаунта .

В свою очередь внутри приложения связь с сервером происходит посредством библиотеки **Requests**, на сервер отправляется запрос и на него формируется ответ. Формат ответа - JSON или массив JSON объектов, которые парсятся и передаются по шаблонам (шаблоны хранятся в папке web/templates)

документация модуля

<https://developers34.pages.bmstu.codes/web-client/web.users.views.html>

Разработка и деплоймент.

Разработка.

Разработка велась, используя систему контроля версий git.

Ссылка на репозиторий: <https://bmstu.codes/developers34/web-client>

В качестве среды разработки использовался редактор Visual Studio Code.

Деплоймент.

Для деплоймента проекта используется платформа облачных вычислений <https://heroku.com>.

Адрес API: <https://sb-service.herokuapp.com/>

Итог.

Пользователь.

Имеет следующие параметры:

- Имя
- Фамилия
- Идентификатор
- Электронный адрес
- Пароль
- Телефонный номер
- Описание
- Изображение в формате .png в качестве аватара
- Время регистрации

Объявление.

Имеет следующие параметры:

- Идентификатор
- Название
- Цена
- Страна

- Город
- Станция метро
- Фотографии в формате .png
- Владелец объявления
- Описание
- Время создания

Функционал.

Следующий функционал успешно реализован:

- Создание нового пользователя
- Вход в аккаунт
- Выход из аккаунта
- Просмотр информации о пользователе
- Изменение информации об аккаунте
- Удаление аккаунта
- Просмотр существующих объявлений
- Поиск объявлений по названию
- Просмотр объявлений конкретного пользователя
- Создание объявлений
- Изменение своих объявлений
- Удаление своих объявлений

Таким образом, все поставленные задачи были выполнены.

Дальнейшее развитие.

Сервис имеет большое количество путей развития, таких как добавление нового функционала и улучшение существующего. Так, можно реализовать:

- Систему рейтинга пользователей

- Чат между пользователями
- Комментарии к объявлениям
- Более продвинутый поиск
- Разделение объявлений по категориям
- Анализ интересов пользователей

Список используемых источников.

1. Django documentation <https://docs.djangoproject.com/en/2.1/>
2. Documentation generator pydoc
<https://docs.python.org/3/library/pydoc.html>
3. python documentation <https://docs.python.org/3.6/>
4. Configuring Django Apps for Heroku
<https://devcenter.heroku.com/articles/django-app-configuration>
5. Registering Runners (gitlab-ci)
<https://docs.gitlab.com/runner/register/index.html#gnulinux>
6. Selenium Documentation <https://www.seleniumhq.org/docs/>